



FAST PARTIAL FOURIER TRANSFORMS FOR LARGE-SCALE PTYCHOGRAPHY

RICARDO PARADA✉^{*1}, SAMY WU FUNG✉² AND STANLEY OSHER✉¹

¹Department of Mathematics, University of California, Los Angeles,
United States of America

²Department of Applied Mathematics and Statistics, Colorado School of Mines,
United States of America

(Communicated by Hui Ji)

ABSTRACT. Ptychography is a popular imaging technique that combines diffractive imaging with scanning microscopy. The technique consists of a coherent beam that is scanned across an object in a series of overlapping positions, leading to reliable and improved reconstructions. Ptychographic microscopes allow for large fields to be imaged at high resolution at additional computational expense. In this work, we explore the use of the fast Partial Fourier Transforms (PFTs), which efficiently compute Fourier coefficients corresponding to low frequencies. The core idea is to use the PFT in a plug-and-play manner to warm-start existing ptychography algorithms such as the ptychographic iterative engine (PIE). This approach reduces the computational budget required to solve the ptychography problem. Our numerical results show that our scheme accelerates the convergence of traditional solvers without sacrificing quality of reconstruction.

1. Introduction. Ptychography is a coherent diffraction imaging (CDI) technique used across various fields including materials science [12, 13, 24], biology [18, 29], and x-ray crystallography [7]. Originally developed to enhance resolution in electron or x-ray microscopy, ptychography replaces single-element detectors with two-dimensional array detectors and integrates diffractive imaging with scanning microscopy. In this process, a coherent beam is systematically moved over an object in a pattern of overlapping positions, linking information across successive diffraction patterns (refer to Fig. 1). In conventional single-pattern CDI, the application of appropriate finite support constraints is essential for the effectiveness of standard algorithms such as Error Reduction [10], Hybrid Input Output (HIO) [8], gradient-based algorithms [5], Relaxed Averaged Alternating Reflections [15], and Saddle Point Optimization [17, 32]. In ptychography, prior knowledge of scanning positions inherently provides these constraints, resulting in methods that are quicker and more reliable than those used in single-pattern CDI.

The extensive use of ptychography has spurred significant research into its reconstruction techniques. The extended Ptychographic Iterative Engine (ePIE) [16] is one of the most favored methods, involving alternating projections onto non-convex

2020 *Mathematics Subject Classification.* Primary: 15A29, 49N45; Secondary: 65T50.

Key words and phrases. Partial fast Fourier transform, Fourier transform, large-scale, ptychography, inverse problem.

*Corresponding author: Ricardo Parada.

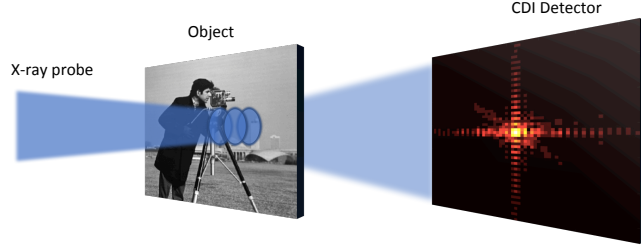


FIGURE 1. Illustration of the ptychography experiment with three overlapping scans.

modulus constraint sets, and has become increasingly popular in the optics field. Mathematically, PIE functions as a projected steepest descent algorithm applied to a specific objective function. Other established methods include conventional gradient-based techniques and the Wirtinger Flow [5, 37], which employs a spectral method for initial setup. Another well-known method, called PhaseLift [4], reformulates the phase retrieval challenge as a convex optimization problem, but requires solving for a significantly larger number of unknowns, becoming impractical for large-scale applications. Despite considerable efforts, solving large-scale ptychographic phase retrieval remains a formidable challenge due to the complexity of handling high resolution, high overlaps, and small scanning beams, all of which contribute to vast amounts of intricate data.

1.1. Our contribution. In this work, we focus on large scale ptychography problems where even applying the FFT is considered computationally taxing. In particular, we explore the use of recent work on fast partial Fourier transforms (PFTs) [22] as a warm up mechanism within the ePIE algorithm to accelerate convergence and time-to-solution of large-scale ptychographic phase retrieval problems. See Figure 2 for an illustration of the PFT and Section 3.1 for more details. The core idea is to let the PFT-based ePIE algorithm capture the large features from the low frequencies in the initial iterations of ePIE, followed by standard FFT-based ePIE to capture the fine details of the reconstruction. Our experiments show that including the PFT within existing algorithms such as the ePIE 1) does not reduce the quality of the reconstruction, and 2) helps accelerate convergence by improving runtimes for large-scale problems. Importantly, in order to be able to use the PFT in gradient-based algorithms, it is necessary to be able to differentiate through the PFT operator. To this end, we also provide a PyTorch [23] implementation of the PFT in order for users to be able to differentiate through the operator using automatic differentiation (AD) [33].

2. Ptychography background. Let $z \in \mathbb{C}^n$ be the object of interest and $d_j \in \mathbb{R}^m$ be the observed data (or intensities) measured from the j^{th} probe, where n and m are the dimensions of the vectorized object and data resolution images, respectively. A ptychography experiment is modeled by

$$d_j = |\mathcal{F}(\omega \odot Q_j z)|, \quad j = 1, \dots, N, \quad (1)$$

where $\mathcal{F} \in \mathbb{C}^{m \times m}$ is the discrete Fourier operator, $\omega \in \mathbb{R}^m$ is the localized probe, $\odot$ represents the Hadamard (or element-wise) product, and $Q_j \in \mathbb{R}^{m \times n}$ is a matrix with binary elements extracting a patch (with the index j and size m) from the

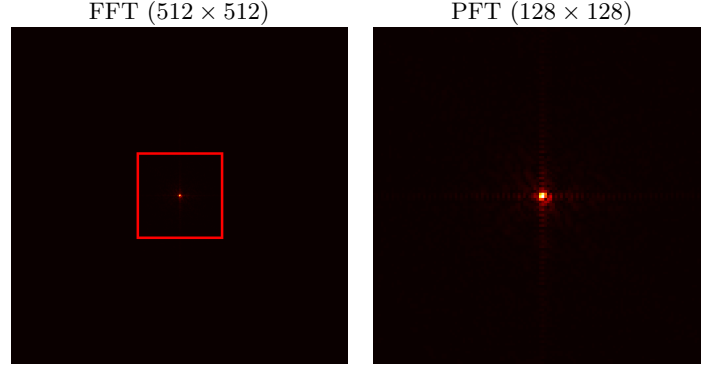


FIGURE 2. Illustration of coefficients computed by the PFT. On the left, we show the full FFT applied to an image of size 512×512 . The red square shows the frequencies that the PFT computes on the right *without* requiring one to take the FFT of the original image and then cropping as one might naively attempt.

entire sample. The blind ptychography problem (BPP) is an inverse problem and can be stated as:

$$\text{Find } \omega \text{ and } z \text{ such that } d_j = |\mathcal{F}(\omega \odot Q_j z)| \quad \text{for } j = 1, \dots, N. \quad (2)$$

In practice, the observations d_j can contain noise, usually Poisson, as is common in most inverse problems. In our experiments, we will also consider the (non-blind) ptychography problem where the probes are known [16]. This model can be realistic in some settings where domain scientists know the probing mechanism a priori. We also note that the above is a specific type of ptychography based on CDI; however other forms of ptychography exist, e.g., Fourier Ptychography [38], and frequency-resolved optical gating (FROG) [31]. While the techniques discussed in this work can be applied to all of them, we will only discuss CDI-based ptychography for ease of presentation.

2.1. The ptychographic iterative engine. As previously stated, the ePIE algorithm is perhaps the most widely used algorithm in practice due to its simplicity. It can be viewed as an alternating projection algorithm onto non-convex modulus constraint sets that solve following feasibility problem. Let the j^{th} measurement constraint set and its corresponding projection operator be denoted by

$$\begin{aligned} \mathcal{M}_j^\omega &= \{z \in \mathbb{C}^n : |\mathcal{F}(\omega \odot Q_j z)| = d_j\} \quad \text{and} \\ \mathcal{P}_{\mathcal{M}_j^\omega}(z) &= \mathcal{F}^{-1} [d_j \odot \exp(i\theta(\mathcal{F}(\omega \odot Q_j z)))] , \end{aligned} \quad (3)$$

where $\theta: \mathbb{C} \rightarrow [-\pi, \pi]$ returns the argument of a complex number and is applied element-wise. The ePIE algorithm generates an approximation to problem by solving the following optimization problem

$$\min_{\omega, z} \Phi(\omega, z) = \frac{1}{N} \sum_{j=1}^N \left\| \omega \odot Q_j z - \mathcal{P}_{\mathcal{M}_j^\omega}(z) \right\|^2. \quad (4)$$

Letting $z_j = Q_j z$, the ePIE iterates are given by

$$z_j^{k+1} = z_j^k - \beta \bar{\omega}^k \left(\omega^k \odot z_j^k - \mathcal{P}_{\mathcal{M}_j^{\omega^k}}(z_j^k) \right), \quad (5)$$

$$\omega^{k+1} = \omega^k - \gamma \bar{z}_j^{k+1} \left(\omega^k \odot z_j^{k+1} - \mathcal{P}_{\mathcal{M}_j^{\omega^k}}(z_j^{k+1}) \right), \quad j = 1, \dots, N, \quad (6)$$

where $\beta, \gamma > 0$ are positive scalars corresponding to step sizes generally chosen to be small [27, 30] and $\bar{\omega}$ is the complex conjugate of ω . The above iterates correspond to performing gradient descent updates on Φ with respect to z_j and ω in an alternating manner [36]. Importantly, we note that to recover z from the latest $z_j, j = 1, \dots, N$ iterates, the pixels extracted (or illuminated) by Q_j are updated in z . Finally, we remark that when the probe ω is known, then the problem is known as a non-blind ptychographic retrieval problem, and the corresponding algorithm simply uses (5) and is known as PIE.

3. A hybrid ePIE algorithm. In this section, we describe the PFT and present our proposed hybrid ePIE algorithm.

3.1. The fast partial Fourier transform (PFT). As seen in Section 2, the key ingredient in the blind (and non-blind) ptychography problem is the discrete fast Fourier transform (FFT), $\mathcal{F}$. Although the discrete FFT is fast and can be applied in $\mathcal{O}(n \log n)$ complexity, it can be the primary bottleneck of reconstruction algorithms such as ePIE when the size of the image is extremely high ($n \gg 1$).

Recently, an algorithm for approximating the FFT, called the fast partial Fourier transform (PFT) [22], was introduced to speed up the computation of the FFT with applications to time series. The primary motivation for the PFT arises from applications where the resulting data from the FFT in the frequency domain is sparse i.e., the Fourier coefficients corresponding to high frequencies are predominantly small or equal to zero, and not all Fourier coefficients are necessary for the task at hand [22]. Indeed, computing the FFT and then cropping the necessary coefficients still requires a cost of $\mathcal{O}(n \log n)$. To this end, the core idea of the PFT is to have a fast mechanism that truncates the high frequencies from the FFT in an efficient manner; in particular, the application of the PFT can be done at a cost of $\mathcal{O}(n + \tilde{m} \log \tilde{m})$ where $\tilde{m} \ll n$.

For ease of presentation, we describe the one-dimensional PFT; however, a two-dimensional version is a straightforward extension and its description can be found in Appendix A. To implement the PFT, there are two phases, an offline phase, which is used to pre-compute the polynomial approximation (see Algorithm 1), and then an online phase which allows one to apply the PFT on the fly (see Algorithm 2). As we are only concerned with ptychographic reconstructions, we are primarily concerned with the online phase of the PFT application and also leave a more thorough derivation of the offline phase in Appendix A.

3.1.1. PFT offline configuration phase: Approximating twiddle factors. While a thorough derivation and description of the PFT is presented in [22], we present parts of the derivation in this work for completeness. Recall that the Discrete Fourier Transform (DFT) is given by:

$$\hat{z}_t = \sum_{k \in [n]} z_k e^{(-2\pi i t k / n)} \quad (7)$$

where $z \in \mathbb{C}^n$ is an n -dimensional complex data vector and $[n] = \{0, 1, \dots, n-1\}$. We assume that n is a composite integer so that there exist $p, q > 1$ such that

$n = pq$. The Cooley-Tukey algorithm [6] rearranges the above expression as:

$$\hat{z}_t = \sum_{k \in [p]} \sum_{j \in [q]} z_{qk+j} e^{-2\pi i t(qk+j)/n} = \sum_{k \in [p]} \sum_{j \in [q]} z_{qk+j} e^{(-2\pi i t j/n)} \cdot e^{(-2\pi i t k/p)}. \quad (8)$$

Further modification of the above expression yields

$$\begin{aligned} \hat{z}_t &= \sum_{k \in [n]} z_k e^{-2\pi i t(k-q/2)/n} \cdot e^{-\pi i t/p} \\ &= \sum_{k \in [p]} \sum_{j \in [q]} z_{qk+j} e^{-2\pi i t(j-q/2)/n} \cdot e^{-2\pi i t k/p} \cdot e^{-\pi i t/p}. \end{aligned} \quad (9)$$

Here, $[q]$ corresponds to the set $\{0, 1, \dots, q-1\}$. The key idea behind the PFT is to use a polynomial to approximate the exponential $e^{\pi i x}$. Afterwards, one can re-scale the polynomials and use exponent laws to get an approximation of each of the *twiddle factors* in the collection $\{e^{-2\pi i t(j-q/2)/n}\}_{j=0}^{q-1}$.

To choose the approximating polynomial of the exponential function $e^{\pi i x}$, we consider the choice of hyper-parameter p given n and desired output size $\tilde{m}$. Let $\|\cdot\|_R$ be the uniform norm (supremum norm) restricted to a set $R \subseteq \mathbb{R}$, that is, $\|f\|_R = \sup \{|f(x)| : x \in R\}$. Then, given non-negative integer α , and non-zero real number ξ , we define the polynomial $\mathcal{P}_{\alpha,\xi}$ as the best approximation to $e^{\pi i x}$ out of the space P_α of polynomials on $\mathbb{R}$ of degree at most α under the restriction $|x| \leq |\xi|$ as:

$$\mathcal{P}_{\alpha,\xi} = \operatorname{argmin}_{P \in P_\alpha} \|P(x) - e^{\pi i x}\|_{|x| \leq |\xi|}. \quad (10)$$

We remark that such polynomials exist and are unique and there are minimax algorithms that can be used to compute them [9]. The goal is to choose $\xi \in \mathbb{R}$ such that $|\tilde{m}| \leq |\xi|$. Given a tolerance $\varepsilon > 0$ and a positive integer $r \geq 1$, we define $\xi(\varepsilon, r)$ to be the scope about the origin such that the exponential function $e^{\pi i x}$ can be approximated by a polynomial of degree less than r with approximation bound ε :

$$\xi(\varepsilon, r) := \sup \left\{ \xi \geq 0 : \|\mathcal{P}_{r-1,\xi}(x) - e^{\pi i x}\|_{|x| \leq \xi} \leq \varepsilon \right\}, \quad (11)$$

where we express the corresponding polynomial as $\mathcal{P}_{r-1,\xi}(x) = \sum_{j \in [r]} w_{\varepsilon,r,j} x^j$. Using a minimax approximation algorithm [9], precompute $\xi(\varepsilon, r)$ and $\{w_{\varepsilon,r,j}\}_{j \in [r]}$ for several tolerance ε 's (e.g. $10^{-1}, 10^{-2}, \dots$) and positive integer r 's (the authors of [22] choose values in the range $1 \leq r \leq 25$). When $n, \tilde{m}, p$, and ε are given, we choose the minimum r satisfying $\xi(\varepsilon, r) \geq \tilde{m}/p$. Following the preceding argument, one can then show that the re-scaled polynomial $\mathcal{P}_{r-1,\xi(\varepsilon,r)}(-2t(j-q/2)/n)$ approximates $e^{-2\pi i t(j-q/2)/n}$ on $|t| \leq \left| \frac{n}{2(j-q/2)} \cdot \frac{\tilde{m}}{p} \right|$ for each $j \in [q]$ [22]. Noting that $\left| \frac{n}{2(j-q/2)} \cdot \frac{\tilde{m}}{p} \right| = \left| \frac{q}{2j-q} \cdot \tilde{m} \right| \geq \tilde{m}$ for all $j \in [q]$, we have a polynomial approximation on $|t| \leq \tilde{m}$ for each twiddle factor in the collection $\{e^{-2\pi i t(j-q/2)/n}\}_{j=0}^{q-1}$, namely the re-scaled polynomials $\{\mathcal{P}_{r-1,\xi(\varepsilon,r)}(-2t(j-q/2)/n)\}_{j=0}^{q-1}$. Algorithm 1 shows how to build the polynomial approximation¹. The polynomial coefficients $w_{\varepsilon,r-1,j}$ are precomputed, and we obtain them from the code database of [22].

¹The algorithm written here centers the PFT at 0 - this is all we need for our purposes. However, with a minor modification, one may also compute the PFT centered at another coordinate [22]

Algorithm 1 : Configuration (Offline) Phase of 1D PFT

Input: Input size $n \in \mathbb{N}$, crop size $\tilde{m} \in \mathbb{N}$, divisor $p \in \mathbb{N}$, and tolerance ε
Output: Matrices $B \in \mathbb{C}^{q \times r}$, $W \in \mathbb{C}^{(2\tilde{m}+1) \times r}$, divisor p , and number of rows and columns, q and r

```

1:  $q = n/p$ 
2:  $r = \min\{r \in \mathbb{N} : \xi(\varepsilon, r) \geq \tilde{m}/p\}$   $\triangleright$  degree of polynomial  $\mathcal{P}$  approximating  $e^{\pi i x}$  within tolerance  $\varepsilon$ 
3: for  $(l, j) \in [q] \times [r]$  do
4:    $x = (1 - 2l/q)$ 
5:    $B[l, j] = w_{\varepsilon, r-1, j} \cdot x^j$   $\triangleright$  Using precomputed  $w_{\varepsilon, r-1, j}$ 
6: end for
7: for  $k = -\tilde{m}, -\tilde{m} + 1, \dots, \tilde{m} - 1, \tilde{m}$  do
8:    $W[k, j] = (k/p)^j \cdot e^{-\pi i k/p}$   $\triangleright$  Precompute remaining terms
9: end for

```

3.1.2. Online PFT computation. Substituting the approximating polynomial described in Section 3.1.1 for the twiddle factors in (9) and performing some algebraic manipulations, we can represent the summation as a matrix-matrix multiplication $C = Z \times B$ where $Z \in \mathbb{C}^{p \times q}$ is the reshaped data vector z and $B \in \mathbb{C}^{q \times r}$ is defined in line 5 of Algorithm 1 followed by a series of FFT computations applied to the columns of the resulting matrix C . Worth noting, the primary cost in Algorithm 2 arises from a) the matrix-matrix multiplication between a matrix $Z \in \mathbb{C}^{p \times q}$ and $B \in \mathbb{C}^{q \times r}$ in line 2 and b) the application of the FFT on a vector of size $\mathbb{C}^p$, which is shown to have complexity $\mathcal{O}(n + \tilde{m} \log \tilde{m})$ [22, Theorem 3]. For thorough details, we refer the reader to [22, Section 3.4]. Finally, we note that % in Algorithm 2 represents the standard integer modulus.

A hyper-parameter that needs to be chosen for the PFT computation is the integer p such that $n = pq$. In our experiments, we are interested in the setting where $\tilde{m} \ll n$ as we wish to economically capture low frequency features when deploying the PFT. In this case, the primary cost in Algorithm 2 occurs in the matrix-matrix multiplication in line 2, which has complexity $\mathcal{O}(nr)$. Thus, it is recommended that $\tilde{m}/p$ be small so that the number r of approximating polynomial terms decreases [22, Section 4.3]. As previously mentioned, a 2D implementation of the PFT is a straightforward extension of the 1D PFT, where given an image of size $n_1 \times n_2$, we have to decide on $\tilde{m}_1, \tilde{m}_2, p_1, p_2$. In our experiments, we find that choosing $p_1 = p_2 = \tilde{m}_1 = \tilde{m}_2 = 64$ is adequate for improved performance. In this case, the resulting cropped image, has size 128×128 since the cropping is performed from $[-\tilde{m}_1, \tilde{m}_1] \times [-\tilde{m}_2, \tilde{m}_2]$ from the center; we note that an FFT on an image of size 128×128 is trivial to compute.

3.2. Hybrid ePIE algorithm. We propose a hybrid ePIE algorithm that consists of using a PFT-based ePIE as a “warm up” followed by the standard FFT-based ePIE. The idea is to use the PFT-based ePIE to capture large features arising from the low frequencies in a cheap manner. Indeed, while one may use the PFT in a plug-and-play manner, we empirically observe that it is better to initially capture the large features early on, and let the standard FFT-based ePIE algorithm capture the fine details in later iterations.

To run the PFT-based ePIE, we convert the observed data d_j , $j = 1, \dots, N$ to match the output dimension of $\mathcal{F}_{\text{PFT}}$. Considering the reshaped version of d_j as a

Algorithm 2 : Computation (Online) Phase of 1D PFT

Input: Vector $z \in \mathbb{C}^n$, crop size $\tilde{m}$, and configuration results
 $B \in \mathbb{C}^{q \times r}$, $W \in \mathbb{C}^{2\tilde{m}+1 \times r}$, p, q, r

Output: Vector $\hat{z}_{\text{PFT}}$ of estimated Fourier coefficients of z in range $[-\tilde{m}, \tilde{m}]$

- 1: $Z = z.\text{reshape}(p, q)$ ▷ reshape z into $p \times q$ matrix
- 2: $C = Z \times B$ ▷ matrix multiply Z by B
- 3: **for** $j \in [r]$ **do**
- 4: $\hat{C}[:, j] = \text{FFT}(C[:, j])$ ▷ apply FFT to each column of C
- 5: **end for**
- 6: **for** $k = -\tilde{m}, -\tilde{m} + 1, \dots, \tilde{m} - 1, \tilde{m}$ **do**
- 7: $\hat{z}_{\text{PFT}}[k] = \sum_{j=0}^{r-1} \left(\hat{C}[k \% p, j] \cdot W[k, j] \right)$ ▷ sum values of $\hat{C}$ (modulus p) times W
- 8: **end for**

matrix, this can be easily done by cropping the centers of $d_j \in \mathbb{R}^{m_1 \times m_2}$ to generate $d_j^{\text{crop}} \in \mathbb{R}^{\tilde{m}_1 \times \tilde{m}_2}$.

Algorithm 3 : Hybrid ePIE Algorithm

Input: Initial Guess $z \in \mathbb{C}^{n_1 n_2 \times 1}$, crop size parameters $\tilde{m}_1, \tilde{m}_2$, observed data $d_j \in \mathbb{R}^{m_1 \times m_2}$ for $j = 1, \dots, N$, stopping tolerances $\epsilon_{\text{PFT}}, \epsilon$, maximum number of iterations $n_{\text{maxiters, PFT}}, n_{\text{maxiters}}$

Output: Solution z_{opt}

- 1: Construct cropped data $d_j^{\text{crop}} \in \mathbb{R}^{\tilde{m}_1 \tilde{m}_2 \times 1}$ from d_j ▷ crop centers of d_j , $j = 1 \dots, N$
- 2: run PFT-based ePIE iterates (5) - (6) using d_j^{crop} , $j = 1, \dots, N$ until convergence to obtain z_{PFT}
- 3: run FFT-based ePIE iterates (5) - (6) using d_j , $j = 1, \dots, N$ and z_{PFT} as an initial condition until convergence to obtain z_{opt} .

4. Numerical experiments. We consider the non-blind and the blind ptychography problem. For each problem type, we investigate the quality of reconstructions on a small experiment with image size 512×512 and study the quality of local minima. Indeed, the PFT is only worth employing when the runtime over the FFT is reduced - this is not as obvious in the small experiments. To this end, we demonstrate the computational benefits of using the PFT, and in particular the proposed hybrid PIE, on a large experiment with image size 16384×16384 . For all experiments, we assume $m = n$, i.e. square images. We work with two images, the baboon and cameraman. We set the baboon image to be the magnitude and the cameraman to be the phase of the ground truth, as shown in Figure 3. Here, the magnitude is chosen to have range $[0, 1]$ and the phase is chosen to have range $[0, \pi/2]$. As explained in Section 3.1, one wants p_1 and p_2 such that $\tilde{m}_1/p_1$ and $\tilde{m}_2/p_2$ is small; in our case, we find that choosing $\tilde{m} = p = 64$ is good enough for all of our experiments. More details on the choice of hyperparameter p can be found in [22, Section 4.3]. Finally, the stopping criteria considered in all algorithms is $\|z^{k+1} - z^k\|/\|z^k\|$. Following [22], we choose a tolerance of 10^{-7} in the computation of the approximating polynomial in Algorithm 4. Our numerical results are also based on a PyTorch implementation of the PFT that was translated from that presented in [22]. The small-scale experiments (Sections 4.1.2 and 4.2.2) are

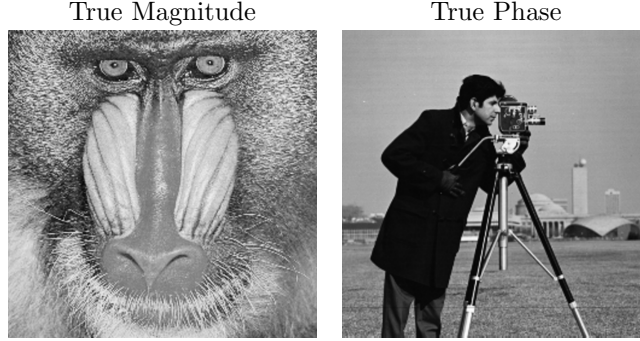


FIGURE 3. The ground truth used to simulate data in numerical experiments. The baboon image is used as the magnitude and the cameraman image is used as the phase of the object of interest.



FIGURE 4. Scanning positions generated by the illumination windows described in Section 4.1.2. For the small problem where $n_1 = n_2 = 512$ (Section 4.1.1), each illumination window Q_i has size 256×256 and shifts 128 pixels at a time. Similarly, for the large-scale problem where $n_1 = n_2 = 16384$ (Section 4.1.4, each illumination window Q_i has size 8192×8192 and shifts 4096 pixels at a time. In both setups, there is a 50% overlap between consecutive probes.

run on a machine equipped with Intel Core i5-9400 CPU, 2.90GHz and 16GB of RAM while large-scale experiments (Sections 4.1.4 and 4.2.3) are run on a Lambda Vector machine with a AMD Threadripper Pro 3955WX processor containing 16 cores, 3.90 GHz, 64 MB cache, PCIe 4.0.

4.1. Non-blind ptychography.

4.1.1. *Experimental setup.* As explained in Section 2, the non-blind ptychography experimental setup assumes ω is known. In this case, we use probes that resembles an identity operator, that is, $\omega \odot Q_k z = Q_k z, k = 1, \dots, N$. Moreover, each probe illuminates $\frac{n}{2} \times \frac{n}{2}$ pixels. That is, the illumination $Q_k \in \mathbb{R}^{n \times n}$ is a matrix that satisfies

$$[Q_k]_{i,j} = \begin{cases} 1 & \text{if pixel } (i,j) \text{ is illuminated} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The illumination window is shifted $\frac{n}{4}$ pixels at a time starting from the top-left corner downward until it reaches the bottom-left corner. The probe is then shifted upwards and then to the right until it reaches the top edge again. This scanning procedure is continued until we have covered the entire image, and results in a

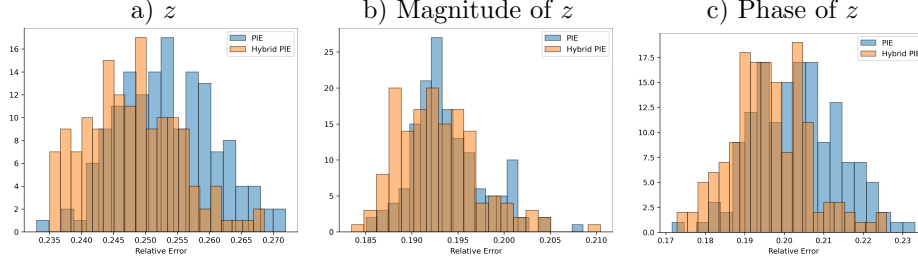


FIGURE 5. Histogram of the final reconstruction relative errors. The blue histogram shows the relative error frequency for PIE and the orange histogram shows the relative error frequency for hybrid PIE. a) shows the relative error of the reconstructed object, b) shows relative errors of only the magnitude of the object, and c) shows the relative errors of the phase of the object.

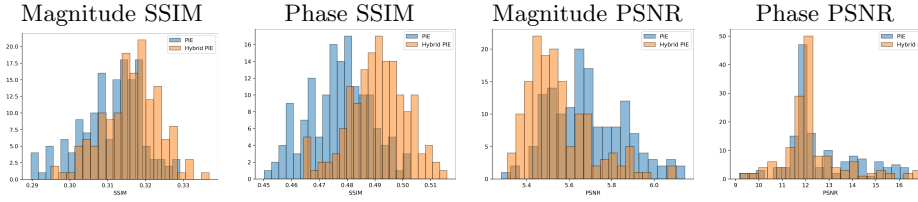


FIGURE 6. Histogram of the final SSIM and PSNR values for the reconstructed images. The blue histogram shows the SSIM/PSNR for PIE and the orange histogram shows the SSIM/PSNR for hybrid PIE.

total of 9 probes with 50% overlap between adjacent probes (see Figure 4 for an illustration).

4.1.2. Distribution of reconstruction quality. To test whether the proposed algorithm 3 leads to improved reconstructions, we compare the distribution of relative errors of the reconstructed images between the PIE and the proposed hybrid PIE algorithms on small-scale images of size 512×512 . We generate 150 reconstructions using both algorithms with uniformly random initial guesses (here, we choose uniformly random in $[0, 1]$ for the magnitude and $[0, \pi/2]$ for the phase). We then compute the relative error of the final reconstructions

$$\frac{\|z - z_{\text{true}}\|}{\|z_{\text{true}}\|} \quad (13)$$

where z is our current reconstruction estimate and z_{true} is the ground truth. Similarly, we also compute the relative errors, peak signal-to-noise ratio (PSNR) [21], and structural similarity index SSIM [34] of the magnitude and phase to better understand the quality of reconstructions.

Figure 5 shows the relative errors between a) the full object, b) the magnitude of the object, and c) the phase of the object. The results show that the hybrid PIE approach leads to a similar, if not slightly better, distribution of reconstructions

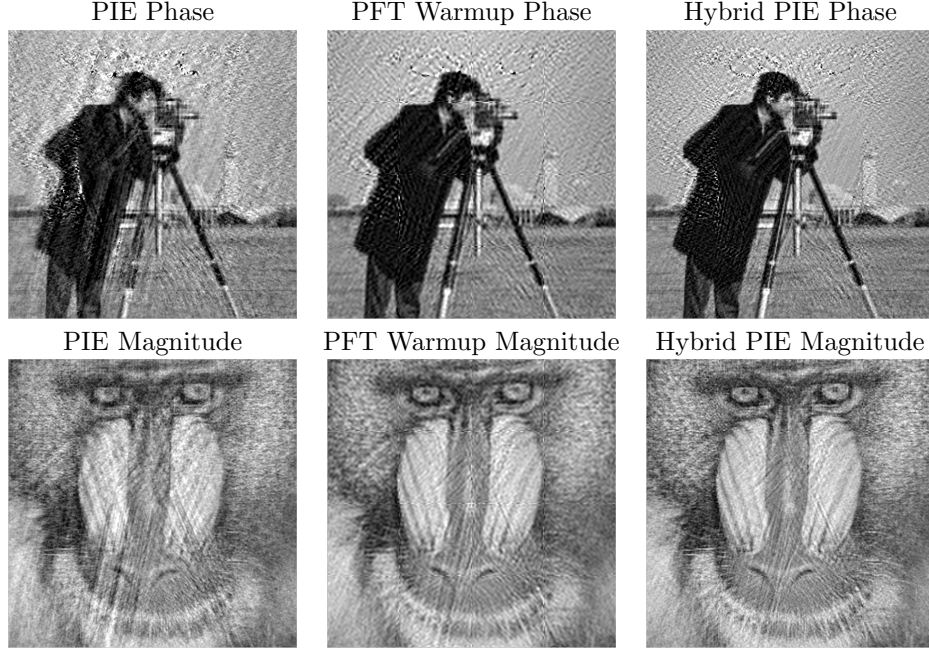


FIGURE 7. Large-scale phase and magnitude reconstructions of the FFT-based PIE (first column), PFT-based PIE warmup (second column), and Hybrid PIE (third column). These reconstructions are for the large-scale ptychography problem where $n_1 = n_2 = 16384$.

as the distribution of the traditional PIE. Similarly, Figure 6 shows a comparisons of the distributions of magnitude and phase SSIM and PSNRs (note for this figure that higher SSIM and PSNR is better).

4.1.3. Investigating overlap. We also investigate the effect of probe overlap on our proposed hybrid approach, generating results using 25%, 50%, and 75% overlap (see Appendix C). For these experiments, we use more realistic cell images to compare four algorithms: PIE, Hybrid PIE, LBFGS [20], and Hybrid LBFGS. We include LBFGS and its hybrid variant to demonstrate that our hybrid approach generalizes beyond PIE alone. As expected, higher overlap leads to improved results, with both hybrid approaches demonstrating superior reconstructions across all overlap percentages. Notably, the hybrid LBFGS approach also improves the quality distribution of reconstructions, further validating the effectiveness of our hybrid methodology (see Appendix C.2).

4.1.4. Large-scale reconstruction. To demonstrate the computational benefits of using a PFT-based PIE in the proposed hybrid algorithm, we run a larger experiment where $n_1 = n_2 = m_1 = m_2 = 16384$, and $\tilde{m}_1 = \tilde{m}_2 = p_1 = p_2 = 64$, leading to a cropped image of size 128×128 . The degree of the approximating polynomial in this setting is given by degrees $r_1 = r_2 = 13$, which is obtained from the precomputed values $\xi(\varepsilon, r)$ defined in Section 3.1.1. Here, we run the PFT-based PIE algorithm until a tolerance of $\epsilon_{\text{pft}} = 10^{-2}$ or a maximum number of 50 iterations is reached.

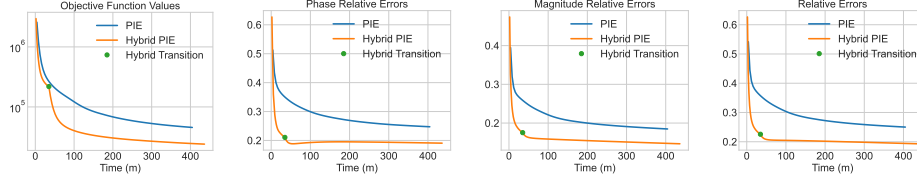


FIGURE 8. Objective function values (first column), phase relative errors (second column), magnitude relative errors (third column), and relative errors (fourth column) for a large-scale ptychographic reconstruction explained in Section 4.1.1. The x-axis represents time in minutes. Green dot represents the transition from the PFT-based PIE to the FFT-based PIE in the hybrid algorithm.

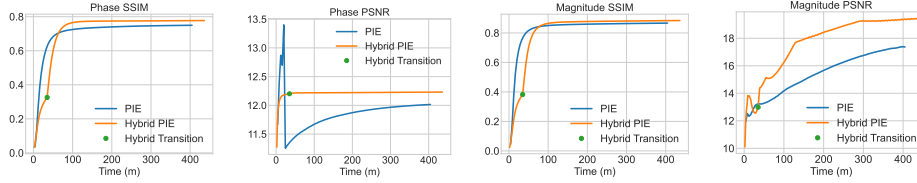


FIGURE 9. SSIM and PSNR for PIE (blue) and Hybrid PIE (orange) for the phase and magnitude over time (in minutes). The green dot represents the transition from the PFT-based PIE to the FFT-based PIE in the hybrid algorithm.

Afterwards, we run the standard FFT-based PIE until a tolerance of $\epsilon = 5 \times 10^{-4}$ or a maximum number of 100 iterations is reached. Here, we choose step sizes $\beta = 10$ for the FFT-based PIE and $\beta_{PFT} = 10^{-3}$ for the PFT-based PIE. These were chosen tuned using a logarithmic gridsearch over the set $\{10^{-6}, 10^{-5}, \dots, 10^2, 10^3\}$, and afterwards using a standard gridsearch starting from the currently chosen parameter, e.g., 10^{-2} , to the next order of magnitude, e.g., 10^{-1} . The intuition for choosing a higher tolerance for the PFT-based PIE is based on the fact that the PFT-based PIE will primarily capture large features, whereas the FFT-based PIE needs more iterations to capture fine details. For all reconstructions, we use total variation regularization [28] with parameters $\lambda = 10^{-6}$ for the FFT-based PIE and $\lambda = 10^2$ for the PFT-based PIE. These are the parameters that led to the lowest relative error and were found over a logarithmic gridsearch over the set $\{10^{-6}, 10^{-5}, \dots, 10^2, 10^3\}$. Interestingly, the PFT-based PIE benefits from larger total variation.

In Figure 7, we show the reconstructions of the magnitude and phase generated by PIE, the initial PFT-based PIE, and the proposed hybrid PIE. In Figure 8, we show the objective function values of the full reconstruction (first column), the phase (second column), the magnitude (third column), and the relative errors (fourth column). Here, the x-axis represents time in minutes. Similar results are shown in Figure 9, where we only show the phase and magnitude structural similarity index (SSIM) [34] and peak signal to noise ratio (PSNR) [26]. We observe that using the PFT-based PIE as warm up leads to improved SSIM and PSNR on the phase and magnitudes of the reconstruction.

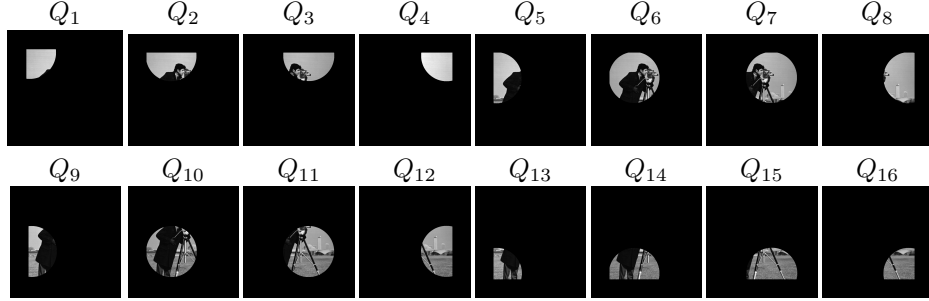


FIGURE 10. Scanning positions of the illumination window described in Section 4.2.1. Here, each probe illuminates a circle of radius 175 pixels and shifts 150 pixels at a time, leading to roughly 50% overlap between consecutive probes.

4.2. Blind ptychographic retrieval.

4.2.1. *Experimental setup.* For the blind experiments, we follow a very similar setup to that in [25]. We use the same images of the baboon and cameraman where the baboon image is set to be the magnitude and the cameraman is set to be the phase of the ground truth. We also set the range of the true phase to be $[0, \frac{\pi}{2}]$ as before.

Following [25], we use circular probes for the blind ptychography problem and the probe, ω is also to be reconstructed. Note this is a very different experimental setup to that the nonblind setting. For a specified radius of pixels r , each illumination window $Q_k \in \mathbb{R}^{n \times n}$ for $k = 1, \dots, N$ is a matrix that satisfies

$$[Q_k]_{i,j} = \begin{cases} 1 & \text{if pixel } (i,j) \text{ lies within circle of radius } r \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

Moreover, the true probe function in this experiment is given by

$$\omega(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \quad (15)$$

where $\sigma = 10^6$. The probe is shifted $\frac{150n}{768}$ pixels at a time starting from the top-left corner to the right until it reaches the top-right corner. The probe is then shifted downwards and then to the left until it reaches the left edge again. This scanning procedure is continued until we have covered the entire image, and results in a total of 16 probes with roughly 50% overlap between adjacent probes. An illustration of the illumination window is shown in Figure 10 for the small-scale experiment (Section 4.2.2) and in Figure 16 for the large-scale experiment (Section 4.2.3). We remark, unlike in the non-blind case, there are nuances in the experimental setup with circular illumination windows that do not allow for the exact same settings when moving up to the large-scale setting. Moreover, following [25] due to the circular nature of the probes, images need to be padded so that the entire image can be covered by the probes. This leads to reconstructions that include the padding (see Figure 13).

Finally, we remark that in this implementation of ePIE (5), (6), we follow the techniques proposed in [25] where the probes, i.e., the value of j , are chosen uniformly at random with no replacement; that is, rather than choose j sequentially in each iteration (as was done in the non-blind case), we choose j randomly until

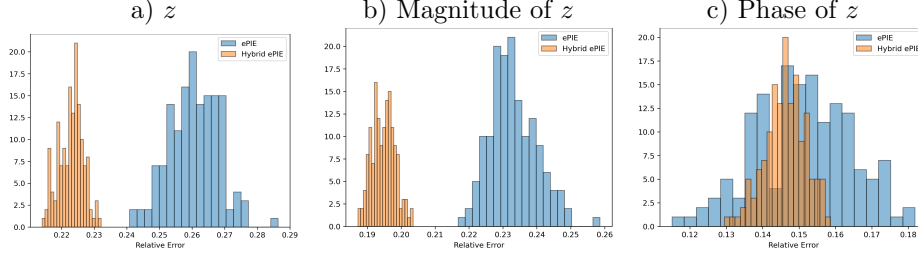


FIGURE 11. Histogram of the final reconstruction relative errors. The blue histogram shows the relative error frequency for ePIE and the orange histogram shows the relative error frequency for hybrid ePIE. a) shows the relative error of the reconstructed object, b) shows relative errors of only the magnitude of the object, and c) shows the relative errors of the phase of the object.

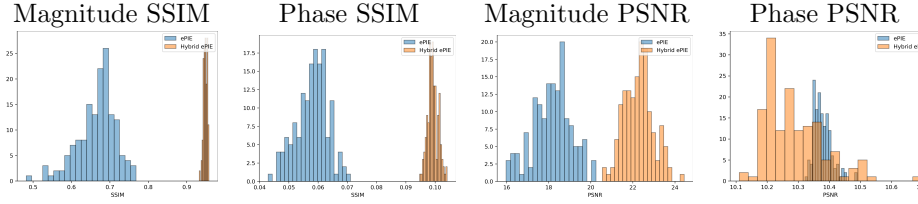


FIGURE 12. Histogram of the final SSIM and PSNR values for the reconstructed images. The blue histogram shows the SSIM/PSNR for ePIE and the orange histogram shows the SSIM/PSNR for hybrid ePIE.

all N probes have been utilized. This random selection is chosen following, where it is observed that this randomness helps avoid local minima.

4.2.2. Distribution of reconstruction quality. We now consider the distribution of the relative errors for the reconstructed images from the ePIE and warm-started ePIE algorithms on small-scale images of size 512×512 . As with the non-blind ptychography experiments, we generate a sample of 150 guess images where the guess magnitude is initialized with uniform random values in the range $[0, 1]$ and $[0, \frac{\pi}{2}]$ for the guess phase. We then compute the relative error of the final reconstructions using (13), as well as the PSNR, and SSIM of the magnitude and phase, respectively, to further analyze the quality of the reconstructions.

Figure 11 shows the relative errors between a) the full object, b) the magnitude of the object, and c) the phase of the object. The results show that hybrid ePIE tends to find a better distribution of reconstructions compared to the distribution of vanilla ePIE. In particular, we see that the relative errors of the magnitude are lower in the hybrid approach. Moreover, we observe that the relative errors of the phase are similar in value (but with lower variance) between hybrid ePIE and traditional ePIE. We also compare the distribution of the final SSIM and PSNR values of the reconstructed image from ePIE and hybrid ePIE shown in Figure 22. We observe that the distribution of the SSIM values are overall higher in the hybrid approach.

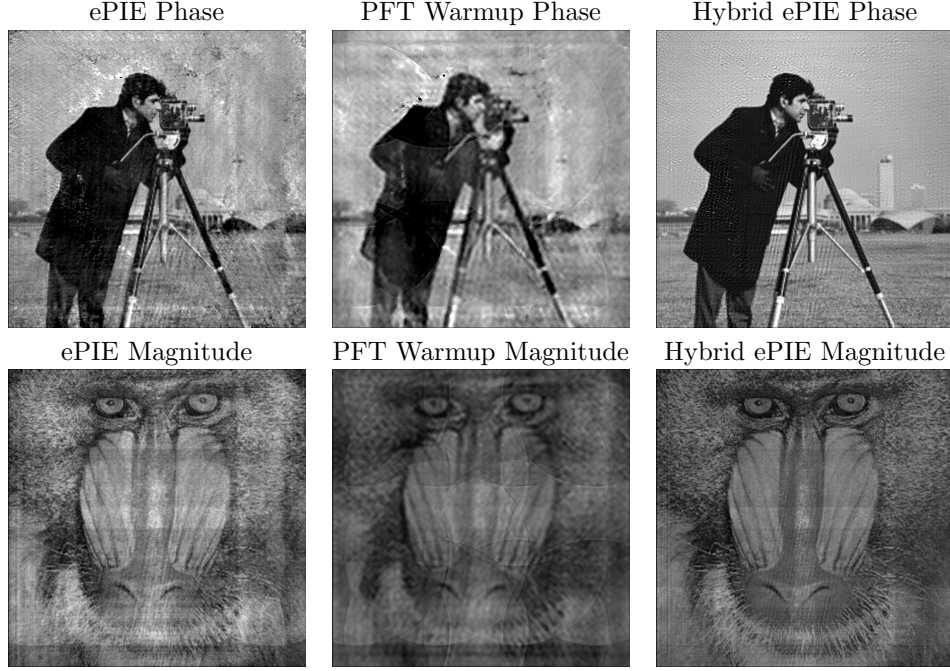


FIGURE 13. Large-scale phase and magnitude reconstructions of the FFT-based ePIE (first column), PFT-based ePIE warmup (second column), and Hybrid ePIE (third column). These reconstructions are for the large-scale ptychography problem where $n_1 = n_2 = 8200$.

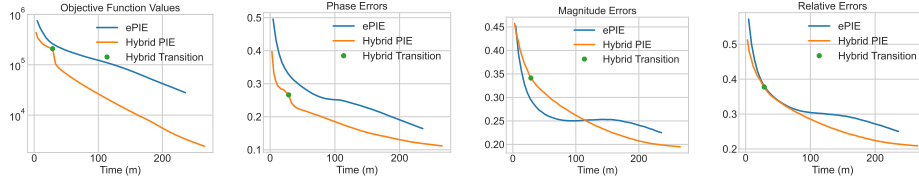


FIGURE 14. Objective values on full reconstruction (first column), phase relative errors (second column), magnitude relative errors (third column), and relative errors (fourth column) for a large-scale ptychographic reconstruction explained in Section 4.2.1. The x-axis represents time in minutes. Green dot represents the transition from the PFT-based ePIE to the FFT-based ePIE in the hybrid algorithm.

However, the hybrid ePIE magnitude reconstructions achieve higher PSNR values and overall slightly lower phase reconstruction PSNR values than vanilla ePIE. In other words, the PSNR histograms support the result of the relative errors, demonstrating the reconstruction of the magnitude makes up for the reconstruction of the phase in the hybrid algorithm.

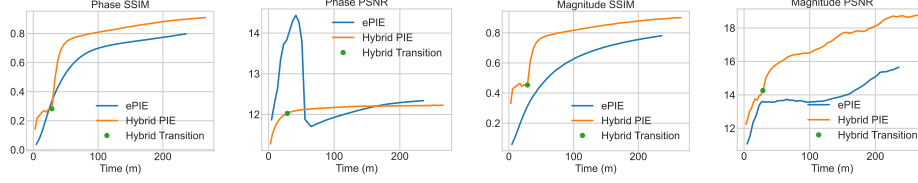


FIGURE 15. SSIM and PSNR for ePIE (blue) and Hybrid ePIE (orange) for the phase and magnitude over time (in minutes). The green dot represents the transition from the PFT-based ePIE to the FFT-based ePIE in the hybrid algorithm.

4.2.3. Large-scale reconstruction. As in the nonblind case, we demonstrate the computational benefits of using a PFT-based ePIE in the proposed algorithm. Here, $n_1 = n_2 = m_1 = m_2 = 8200$, and $\tilde{m}_1 = \tilde{m}_2 = p_1 = p_2 = 64$, leading to a cropped image of size 128×128 . The degree of the approximating polynomial in this setting is given by degrees $r_1 = r_2 = 13$, which is obtained from the precomputed values $\xi(\varepsilon, r)$ defined in Section 3.1.1. Here, we run the PFT-based ePIE algorithm until a tolerance of $\epsilon_{\text{PFT}} = 10^{-2}$ or a maximum number of 10 iterations is reached. Afterwards, we run the standard FFT-based ePIE until a tolerance of $\epsilon = 5 \times 10^{-4}$ or a maximum number of 50 iterations is reached. Here, we choose step sizes $\gamma = \beta = 10$ for the FFT-based ePIE and $\gamma_{\text{PFT}} = \beta_{\text{PFT}} = 2 \times 10^{-3}$ for the PFT-based ePIE. These were chosen tuned using a logarithmic gridsearch over the set $\{10^{-6}, 10^{-5}, \dots, 10^2, 10^3\}$, and afterwards using a standard gridsearch starting from the currently chosen parameter, e.g., 10^{-2} , to the next order of magnitude, e.g., 10^{-1} . The intuition for choosing a higher tolerance for the PFT-based ePIE is based on the fact that the PFT-based ePIE will primarily capture large features, whereas the FFT-based ePIE needs more iterations to capture fine details. For all reconstructions, we use total variation regularization [28] with parameters $\lambda = 10^{-6}$ for the FFT-based ePIE and $\lambda = 10^3$ for the PFT-based ePIE. These are the parameters that led to the lowest relative error and were found over a logarithmic gridsearch over the set $\{10^{-6}, 10^{-5}, \dots, 10^2, 10^3\}$.

In Figure 14, we show the objective values of the full reconstruction (first column), the phase (second column), the magnitude (third column), and the relative errors (fourth column). Here, the x-axis represents time in minutes. Similar results are shown in Figure 15, where we only show the phase and magnitude structural similarity index (SSIM) [34] and peak signal to noise ratio (PSNR) [26]. We observe that while using the PFT-based ePIE as warm up leads to overall improved reconstructions, there are some trade-offs, this can be seen in, e.g., the hybrid ePIE magnitude reconstruction. However, for this particular large-scale run, we obtain a very good quality phase reconstruction as seen in Figure 13. This is appealing as the primary object of interest in ptychography (also known as ptychographic phase retrieval).

Note there is a nontrivial padding element for this ptychographic experiment, where the image recovered consists of the reconstruction and corresponding noisy padding. To address this, we use the built-in function `match_template` algorithm from `scikit-image` [14] to generate Figure 13.

5. Discussion. We present a new hybrid algorithm for ptychographic phase retrieval based on the fast partial Fourier transform (PFT), which only computes

the coefficients corresponding to low frequencies. The PFT is utilized within the ptychographic iterative engine (PIE). The hybrid algorithm consists of using PFT-based PIE in early iterations as a warm up followed by the standard FFT-based PIE algorithm. The core idea is to let the PFT-based PIE iterations capture large features corresponding to low frequencies whereas the FFT-based PIE iterations capture fine details in the reconstruction. Our numerical results demonstrate that the proposed hybrid PIE algorithm accelerates convergence by reducing the time-to-solution. This work also provides a PyTorch implementation of the PFT with automatic differentiation capabilities, enabling ease of use of the PFT within, e.g., deep learning architectures and other optimization algorithms. Future works include exploring a differentiable optimization methodology [1, 11, 19] of the proposed hybrid algorithm as well as distributed methods such as alternating direction method of multipliers [2, 3, 35]. Our code can be found in <https://github.com/mines-opt-ml/pft-for-ptycho>.

Acknowledgments. Samy Wu Fung was partially funded by National Science Foundation award DMS-2110745. Stanley Osher was partially funded by Air Force Office of Scientific Research (AFOSR) MURI FA9550-18-502, Office of Naval Research (ONR) N00014-20-1-2787, and STROBE: a National Science Foundation Science and Technology Center under Grant No. DMR-1548924. We thank Yong-chan Park for his help on setting up the PyTorch-based PFT code.

REFERENCES

- [1] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond and J. Z. Kolter, Differentiable convex optimization layers, *Advances in Neural Information Processing Systems*, **32**.
- [2] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, [Distributed optimization and statistical learning via the alternating direction method of multipliers](#), *Foundations and Trends® in Machine Learning*, **3** (2011), 1-122.
- [3] K. Bui and Z. Di, [A stochastic ADMM algorithm for large-scale ptychography with weighted difference of anisotropic and isotropic total variation](#), *Inverse Problems*, **40** (2024), Paper No. 055006, 39 pp.
- [4] E. J. Candes, Y. C. Eldar, T. Strohmer and V. Voroninski, [Phase retrieval via matrix completion](#), *SIAM Review*, **57** (2015), 225-251.
- [5] E. J. Candes, X. Li and M. Soltanolkotabi, [Phase retrieval via Wirtinger flow: Theory and algorithms](#), *IEEE Transactions on Information Theory*, **61** (2015), 1985-2007.
- [6] J. W. Cooley and J. W. Tukey, [An algorithm for the machine calculation of complex Fourier series](#), *Mathematics of Computation*, **19** (1965), 297-301.
- [7] L. De Caro, D. Altamura, M. Arciniegas, D. Siliqi, M. R. Kim, T. Sibillano, L. Manna and C. Giannini, [Ptychographic imaging of branched colloidal nanocrystals embedded in free-standing thick polystyrene films](#), *Scientific Reports*, **6** (2016), 19397.
- [8] J. R. Fienup, [Phase retrieval algorithms: A comparison](#), *Applied Optics*, **21** (1982), 2758-2769.
- [9] W. Fraser, [A survey of methods of computing minimax and near-minimax polynomial approximations for functions of a single independent variable](#), *Journal of the ACM (JACM)*, **12** (1965), 295-314.
- [10] R. W. Gerchberg, A practical algorithm for the determination of phase from image and diffraction plane pictures, *Optik*, **35** (1972), 237-246.
- [11] H. Heaton and S. Wu Fung, [Explainable AI via learning to optimize](#), *Scientific Reports*, **13** (2023), 10103.
- [12] M. Holler, M. Guizar-Sicairos, E. H. Tsai, R. Dinapoli, E. Müller, O. Bunk, J. Raabe and G. Aeppli, [High-resolution non-destructive three-dimensional imaging of integrated circuits](#), *Nature*, **543** (2017), 402-406.
- [13] R. Hoppe, J. Reinhardt, G. Hofmann, J. Patommel, J.-D. Grunwaldt, C. D. Damsgaard, G. Wellenreuther, G. Falkenberg and C. Schroer, [High-resolution chemical imaging of gold nanoparticles using hard x-ray ptychography](#), *Applied Physics Letters*, **102** (2013), 203104.
- [14] J. P. Lewis, Fast normalized cross-correlation, in *Vision Interface*, **10** (1995), 120-123.

- [15] D. R. Luke, [Relaxed averaged alternating reflections for diffraction imaging](#), *Inverse Problems*, **21** (2005), 37-50.
- [16] A. M. Maiden and J. M. Rodenburg, [An improved ptychographical phase retrieval algorithm for diffractive imaging](#), *Ultramicroscopy*, **109** (2009), 1256-1262.
- [17] S. Marchesini, [Phase retrieval and saddle-point optimization](#), *JOSA A*, **24** (2007), 3289-3296.
- [18] J. Marrison, L. Rätty, P. Marriott and P. O'toole, [Ptychography—a label free, high-contrast imaging technique for live cells using quantitative phase information](#), *Scientific Reports*, **3** (2013), 2369.
- [19] D. McKenzie, S. Wu Fung and H. Heaton, Differentiating through integer linear programs with quadratic regularization and davis-yin splitting, *Transactions on Machine Learning Research*.
- [20] J. Nocedal and S. J. Wright, *Numerical Optimization, 2nd edition*, Springer, New York City, 2006.
- [21] G. Padmavathi, P. Subashini and P. Lavanya, Performance evaluation of the various edge detectors and filters for the noisy ir images, in *2nd International Conference on Sensors, Signals, Visualization, Imaging, Simulation and Material, Baltimore, USA*, (2009), 199-203.
- [22] Y.-c. Park, J.-G. Jang and U. Kang, Fast and accurate partial fourier transform for time series data, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, (2021), 1309-1318.
- [23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, **32**.
- [24] P. M. Pelz, M. Guizar-Sicairos, P. Thibault, I. Johnson, M. Holler and A. Menzel, [On-the-fly scans for X-ray ptychography](#), *Applied Physics Letters*, **105** (2014), 251101.
- [25] M. Pham, A. Rana, J. Miao and S. Osher, [Semi-implicit relaxed Douglas-Rachford algorithm \(sDR\) for ptychography](#), *Optics Express*, **27** (2019), 31246-31260.
- [26] D. Poobathy and R. M. Chezian, [Edge detection operators: Peak signal to noise ratio based comparison](#), *International Journal of Image, Graphics and Signal Processing*, **10** (2014), 55-61.
- [27] J. M. Rodenburg, [Ptychography and related diffractive imaging methods](#), *Advances in Imaging and Electron Physics*, **150** (2008), 87-184.
- [28] L. I. Rudin, S. Osher and E. Fatemi, [Nonlinear total variation based noise removal algorithms](#), *Physica D: Nonlinear Phenomena*, **60** (1992), 259-268.
- [29] A. Suzuki, K. Shimomura, M. Hirose, N. Burdet and Y. Takahashi, [Dark-field X-ray ptychography: Towards high-resolution imaging of thick and unstained biological specimens](#), *Scientific Reports*, **6** (2016), 35060.
- [30] P. Thibault, M. Dierolf, O. Bunk, A. Menzel and F. Pfeiffer, [Probe retrieval in ptychographic coherent diffractive imaging](#), *Ultramicroscopy*, **109** (2009), 338-343.
- [31] R. Trebino, *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses*, Springer Science & Business Media, 2012.
- [32] A. Tripathi, S. Leyffer, T. Munson and S. M. Wild, [Visualizing and improving the robustness of phase retrieval algorithms](#), *Procedia Computer Science*, **51** (2015), 815-824.
- [33] B. Van Merriënboer, O. Breuleux, A. Bergeron and P. Lamblin, Automatic differentiation in ml: Where we are and where we should be going, *Advances in Neural Information Processing Systems*, **31**.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, [Image quality assessment: From error visibility to structural similarity](#), *IEEE Transactions on Image Processing*, **13** (2004), 600-612.
- [35] S. Wu Fung and L. Ruthotto, [An uncertainty-weighted asynchronous ADMM method for parallel PDE parameter estimation](#), *SIAM Journal on Scientific Computing*, **41** (2019), S129-S148.
- [36] S. Wu Fung and Z. Di, [Multigrid optimization for large-scale ptychographic phase retrieval](#), *SIAM Journal on Imaging Sciences*, **13** (2020), 214-233.
- [37] 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, 1806.05546.
- [38] G. Zheng, C. Shen, S. Jiang, P. Song and C. Yang, [Concept, implementations and applications of Fourier ptychography](#), *Nature Reviews Physics*, **3** (2021), 207-223.

Received August 2024; revised April 2025; early access May 2025.

Appendix A. 2D PFT algorithm. We present the two-dimensional (2D) extension of the PFT algorithm. Analogous to the 1D case, we begin by recalling the 2D DFT given by:

$$\hat{z}_{t_1, t_2} = \sum_{(k_1, k_2) \in [n_1] \times [n_2]} z_{k_1, k_2} e^{-2\pi i t_1 k_1 / n_1} e^{-2\pi i t_2 k_2 / n_2} \quad (16)$$

where now $z \in \mathbb{C}^{n_1 \times n_2}$ is a complex-valued matrix of size $n_1 \times n_2$. We assume that $n_1 = p_1 q_1$ and $n_2 = p_2 q_2$ are composite integers where $p_1, p_2, q_1, q_2 > 1$. Then rearranging the above expression, we get:

$$\hat{z}_{t_1, t_2} = \sum_{k_1, k_2, j_1, j_2} z_{q_1 k_1 + j_1, q_2 k_2 + j_2} \prod_{\nu} e^{-2\pi i t_1 (j_{\nu} - q_{\nu}/2)/n_{\nu}} \cdot e^{-2\pi i t_2 j_{\nu}/p_{\nu}} \cdot e^{-\pi i t_{\nu}/p_{\nu}} \quad (17)$$

where $k_1 \in [p_1], k_2 \in [p_2], j_1 \in [q_1], j_2 \in [q_2]$, and $\nu = 1, 2$. As before, we want to use polynomial approximations for the exponential $e^{\pi i x}$. Afterwards, by re-scaling the polynomials and using exponent laws, one can get an approximation of the twiddle factors in the collection $\{e^{-2\pi i t_{\nu} (j_{\nu} - q_{\nu}/2)/n_{\nu}}\}_{j_{\nu}=0}^{q_{\nu}-1}$. Using the same definitions and notation as in Section 3.1.1, choose the minimum r_{ν} satisfying $\xi(\varepsilon, r_{\nu}) \geq \tilde{m}_{\nu}/p_{\nu}$ to get the re-scaled polynomial approximations $\{\mathcal{P}_{r_{\nu}-1, \xi(\varepsilon, r_{\nu})}(-2t_{\nu}(j_{\nu} - q_{\nu}/2)/n_{\nu})\}_{j_{\nu}=0}^{q_{\nu}-1}$. Algorithm 4 shows how to build the polynomial approximation. The polynomial coefficients $w_{\varepsilon, r_{\nu}-1, j}$ are precomputed, and we obtain them from the code database of [22].

Algorithm 4 : Configuration (Offline) Phase of 2D PFT

Input: Input size $(n_1, n_2) \in \mathbb{N}^2$, crop size $(\tilde{m}_1, \tilde{m}_2) \in \mathbb{N}^2$, divisors $(p_1, p_2) \in \mathbb{N}^2$, and tolerance ε

Output: Matrices $B_1 \in \mathbb{C}^{q_1 \times r_1}, B_2 \in \mathbb{C}^{q_2 \times r_2}$, tensor $W \in \mathbb{C}^{(2\tilde{m}_1+1) \times (2\tilde{m}_2+1) \times r_1 \times r_2}$, configuration results $p_1, p_2, q_1, q_2, r_1, r_2$

- 1: **for** $\nu = 1, 2$ **do**
- 2: $q_{\nu} = n_{\nu}/p_{\nu}$
- 3: $r_{\nu} = \min\{r_{\nu} \in \mathbb{N} : \xi(\varepsilon, r_{\nu}) \geq \tilde{m}_{\nu}/p_{\nu}\} \triangleright$ degree of polynomial $\mathcal{P}$ approximating $e^{\pi i x}$ within tolerance ε
- 4: **for** $l \in [q_{\nu}], j \in [r_{\nu}]$ **do**
- 5: $x = (1 - 2l/q_{\nu})$
- 6: $B_{\nu} = w_{\varepsilon, r_{\nu}-1, j} \cdot x^j \quad \triangleright$ Using precomputed $w_{\varepsilon, r_{\nu}-1, j}$
- 7: **end for**
- 8: **end for**
- 9: **for** $(t_1, t_2) = ([-\tilde{m}_1, \dots, \tilde{m}_1] \times [-\tilde{m}_2, \dots, \tilde{m}_2])$ **do**
- 10: $W[t_1, t_2, j_1, j_2] = (t_1/p_1)^{j_1} e^{-\pi i t_1/p_1} \cdot (t_2/p_2)^{j_2} e^{-\pi i t_2/p_2} \quad \triangleright$ Precompute remaining terms
- 11: **end for**

Substituting the approximating polynomials for the twiddle factors in (17) and performing some algebraic manipulations (in particular, swapping and rewriting summations), we represent the summations as matrix-matrix multiplications between $C^{(k_1, k_2)} = B_1^{\top} \times Z^{(k_1, k_2)} \times B_2$ where $Z^{(k_1, k_2)} \in \mathbb{C}^{q_1 \times q_2}$ are slices of the data matrix z , the matrices $B_1 \in \mathbb{C}^{q_1 \times r_1}$ and $B_2 \in \mathbb{C}^{q_2 \times r_2}$ are defined in line 6 of Algorithm 4 followed by a series of 2D FFT computations on C for each

$(k_1, k_2) \in [p_1] \times [p_2]$. As in the 1D PFT, we note that the primary cost of Algorithm 5 lies in the matrix multiplications between $Z^{(k_1, k_2)}$, B_1 , and B_2 and the 2D FFT computations on each of the matrices $C^{(k_1, k_2)} \in \mathbb{C}^{r_1 \times r_2}$. This gives us time complexity $\mathcal{O}(n + \tilde{m} \log \tilde{m})$, where $n = n_1 n_2$ and $\tilde{m} = \tilde{m}_1 \tilde{m}_2$. For thorough details, we refer the reader to [22, Appendix B].

Algorithm 5 : Computation (Online) Phase of 2D PFT

Input: 2D array Z of size $n_1 \times n_2$, crop size $(\tilde{m}_1, \tilde{m}_2)$, tensor $W \in \mathbb{C}^{(2\tilde{m}_1+1) \times (2\tilde{m}_2+1) \times r_1 \times r_2}$, configuration results $B_1 \in \mathbb{C}^{q_1 \times r_1}, B_2 \in \mathbb{C}^{q_2 \times r_2}, p_1, p_2, q_1, q_2, r_1, r_2$

Output: 2D array $\hat{Z}_{PFT}$ of estimated Fourier coefficients of Z

- 1: $Z = Z.\text{reshape}(p_1, q_1, p_2, q_2)$
- 2: $Z = Z.\text{permute}(0, 2, 1, 3).\text{contiguous}()$
- 3: $Z = Z.\text{reshape}(p_1, p_2, q_1, q_2)$ $\triangleright$ reshape z into $p_1 \times p_2 \times q_1 \times q_2$ tensor
- 4: **for** $(k_1, k_2) \in [p_1] \times [p_2]$ **do**
- 5: $C[k_1, k_2, :, :] = B_1^\top \times Z[k_1, k_2, :, :] \times B_2$ $\triangleright$ matrix multiply $B_1^\top$ by $Z[k_1, k_2, :, :]$ by B_2
- 6: **end for**
- 7: **for** $(j_1, j_2) \in [r_1] \times [r_2]$ **do**
- 8: $\hat{C}[:, :, j_1, j_2] = \text{FFT2}(C[:, :, j_1, j_2])$ $\triangleright$ apply 2D FFT to matrices $C[:, :, j_1, j_2]$
- 9: **end for**
- 10: **for** $(t_1, t_2) = ([-\tilde{m}_1, \dots, \tilde{m}_1] \times [-\tilde{m}_2, \dots, \tilde{m}_2])$ **do**
- 11: $\hat{Z}_{PFT}[t_1, t_2] = \sum_{j_1 \in [r_1], j_2 \in [r_2]} \hat{C}[t_1 \% p_1, t_2 \% p_2, j_1, j_2] \cdot W[t_1, t_2, j_1, j_2]$ $\triangleright$ Hadamard product
- 12: **end for**

Appendix B. Illumination windows for large-scale blind ptychography.

We provide an illustration of the illumination window for the large-scale experiment. The size of the image is 8200×8200 , each circular probe has a radius of 2553 pixels, and shifts 2188 pixels at a time.

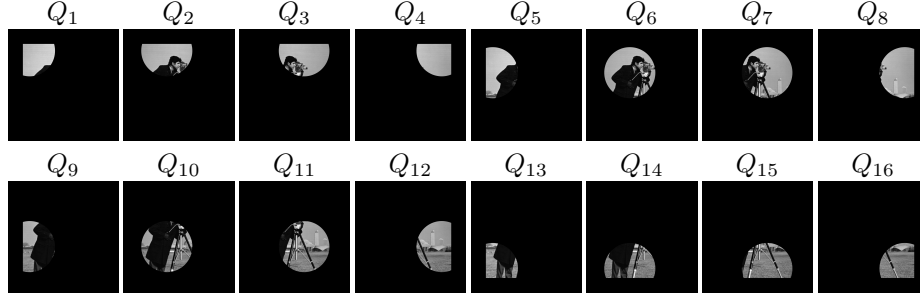


FIGURE 16. Scanning positions of the illumination window described in Section 4.2.1. Here, each probe illuminates a circle of radius 2553 pixels and shifts 2188 pixels at a time, leading to roughly 50% overlap between consecutive probes.

Appendix C. Additional experiments. We perform additional experiments using more realistic cell images and investigate the effect of different overlaps in the PIE experimental setup.

C.1. Additional PIE experiments.

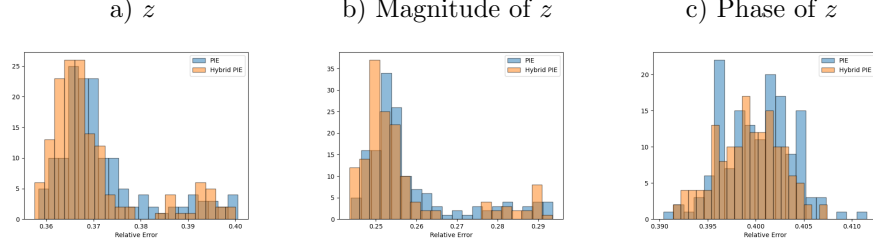


FIGURE 17. Histogram of the final reconstruction relative errors for the cell images. The blue histogram shows the relative error frequency for PIE and the orange histogram shows the relative error frequency for hybrid PIE. a) shows the relative error of the reconstructed object, b) shows relative errors of only the magnitude of the object, and c) shows the relative errors of the phase of the object.

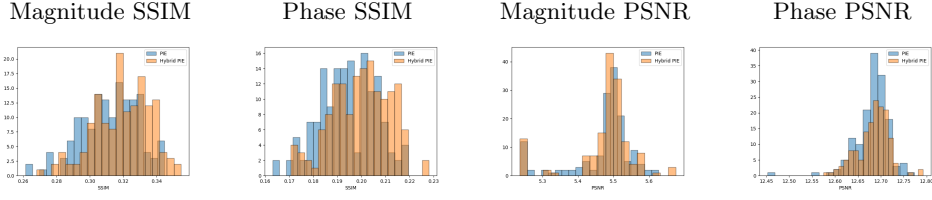


FIGURE 18. Histogram of the final SSIM and PSNR values for the reconstructed cell images. The blue histogram shows the SSIM/PSNR for PIE and the orange histogram shows the SSIM/PSNR for hybrid PIE.

C.2. Additional LBFGS experiments.

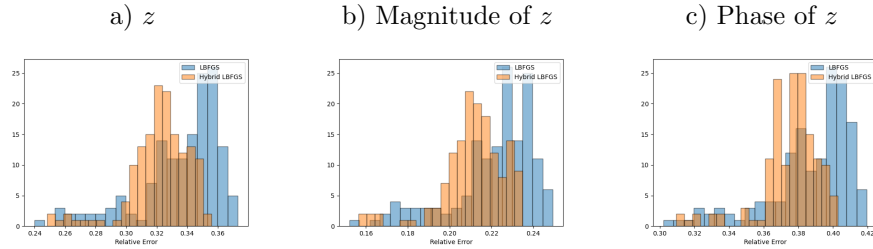


FIGURE 19. Histogram of the final reconstruction relative errors for the cell images. The blue histogram shows the relative error frequency for LBFGS and the orange histogram shows the relative error frequency for hybrid LBFGS. a) shows the relative error of the reconstructed object, b) shows relative errors of only the magnitude of the object, and c) shows the relative errors of the phase of the object.

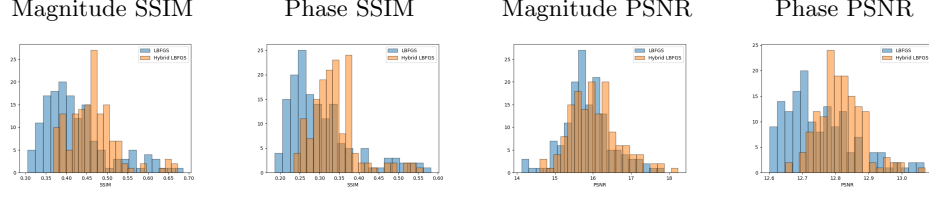


FIGURE 20. Histogram of the final SSIM and PSNR values for the reconstructed cell images. The blue histogram shows the SSIM/PSNR for LBFGS and the orange histogram shows the SSIM/PSNR for hybrid LBFGS.

C.3. Additional ePIE experiments.

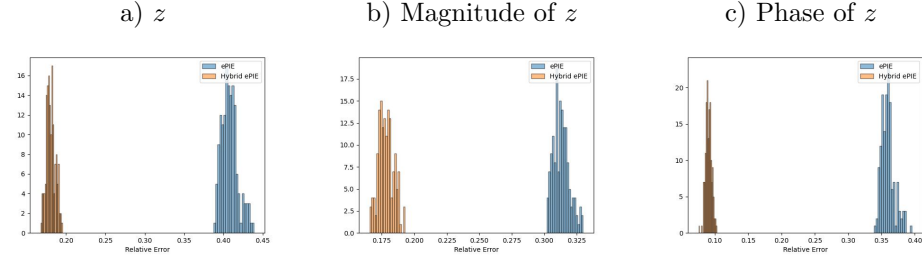


FIGURE 21. Histogram of the final reconstruction relative errors for the cell images. The blue histogram shows the relative error frequency for ePIE and the orange histogram shows the relative error frequency for hybrid ePIE. a) shows the relative error of the reconstructed object, b) shows relative errors of only the magnitude of the object, and c) shows the relative errors of the phase of the object.

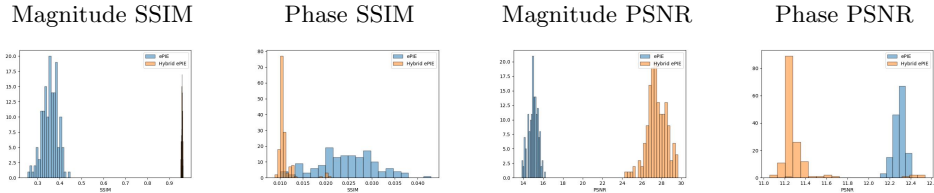


FIGURE 22. Histogram of the final SSIM and PSNR values for the reconstructed cell images. The blue histogram shows the SSIM/PSNR for ePIE and the orange histogram shows the SSIM/PSNR for hybrid ePIE.

C.4. Investigating overlap. In the following experiments, we examine the effects on performance by changing the overlap percentage of the illumination windows. In particular, we investigate 25%, 50%, and 75% overlap for both PIE, LBFGS, and their hybrid counterparts.

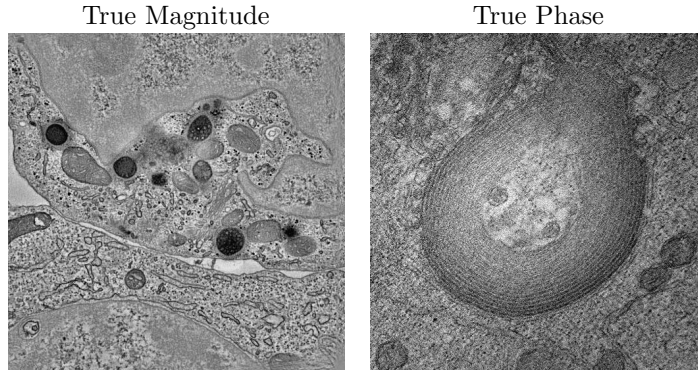


FIGURE 23. The ground truth used to simulate data in numerical experiments. Two distinct cell images are used as the magnitude and phase of the object of interest.

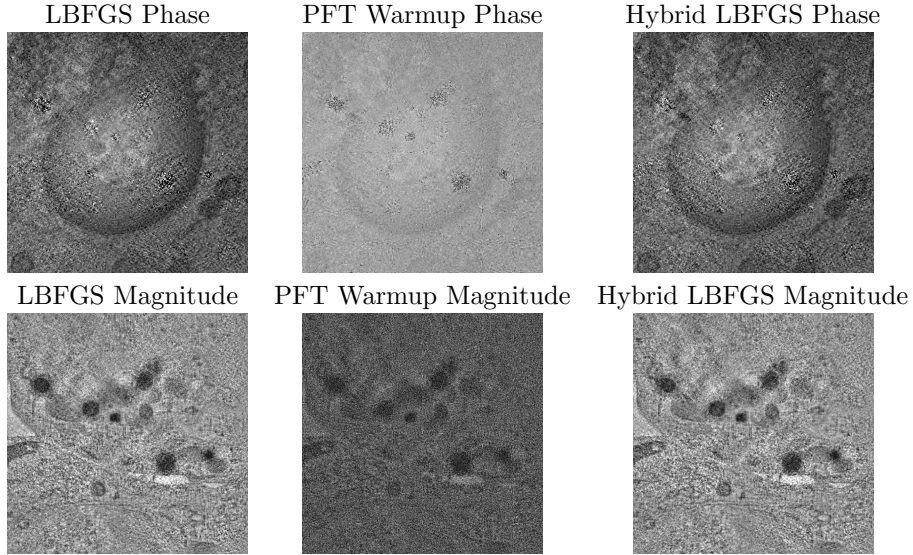


FIGURE 24. Phase and magnitude reconstructions of LBFGS (first column), PFT-based LBFGS warmup (second column), and hybrid LBFGS (third column). These reconstructions are done on illuminations windows with 25% overlap.

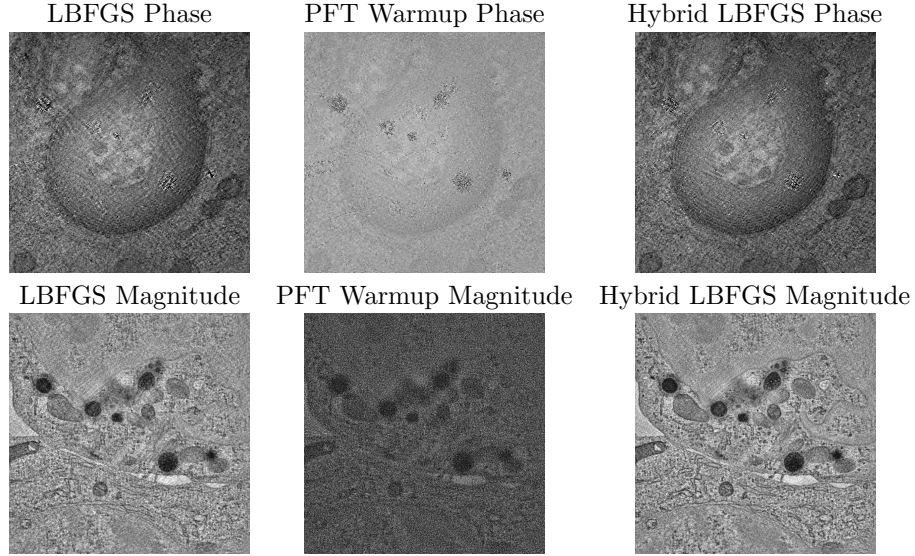


FIGURE 25. Phase and magnitude reconstructions of LBFGS (first column), PFT-based LBFGS warmup (second column), and hybrid LBFGS (third column). These reconstructions are done on illuminations windows with 50% overlap.

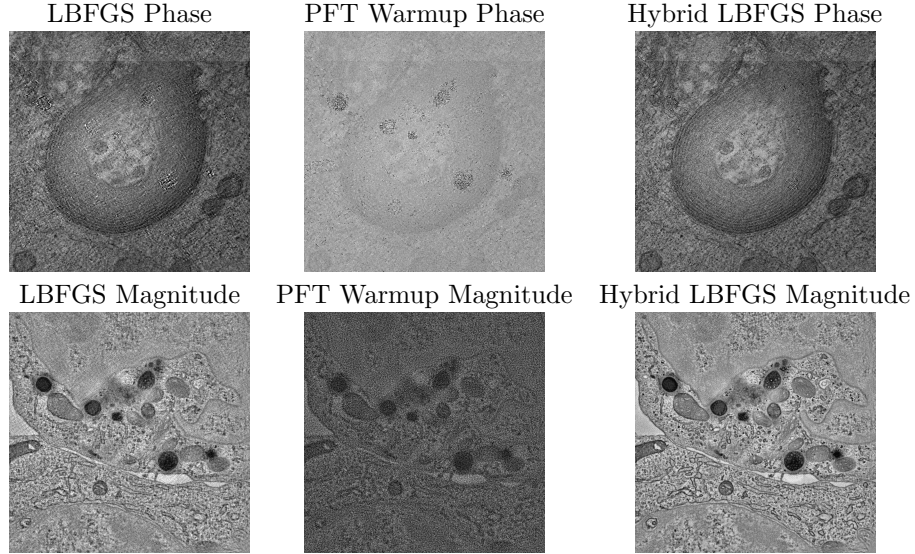


FIGURE 26. Phase and magnitude reconstructions of LBFGS (first column), PFT-based LBFGS warmup (second column), and hybrid LBFGS (third column). These reconstructions are done on illuminations windows with 75% overlap.

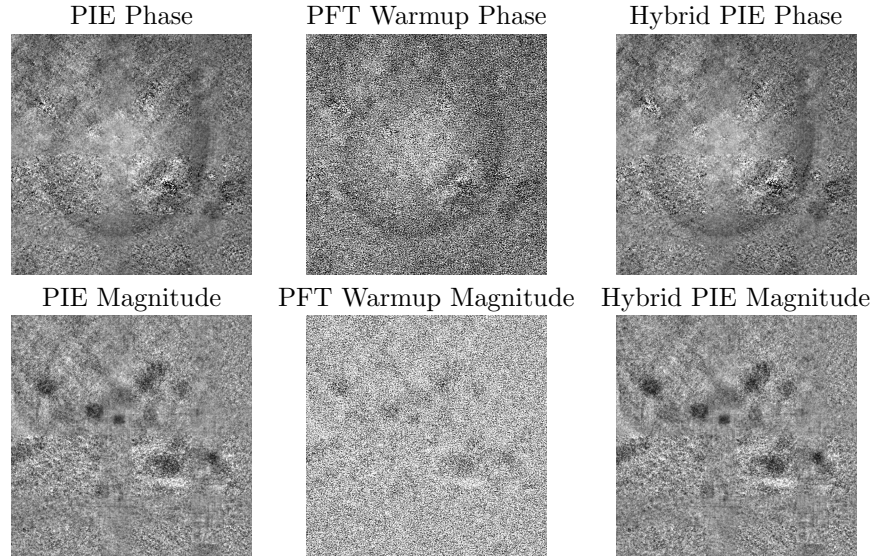


FIGURE 27. Phase and magnitude reconstructions of PIE (first column), PFT-based PIE warmup (second column), and hybrid PIE (third column). These reconstructions are done on illuminations windows with 25% overlap.

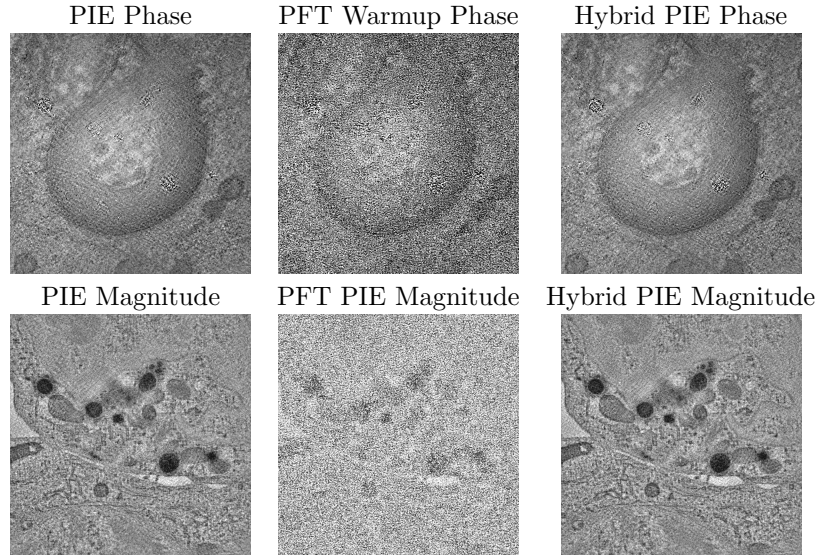


FIGURE 28. Phase and magnitude reconstructions of PIE (first column), PFT-based PIE warmup (second column), and hybrid PIE (third column). These reconstructions are done on illuminations windows with 50% overlap.

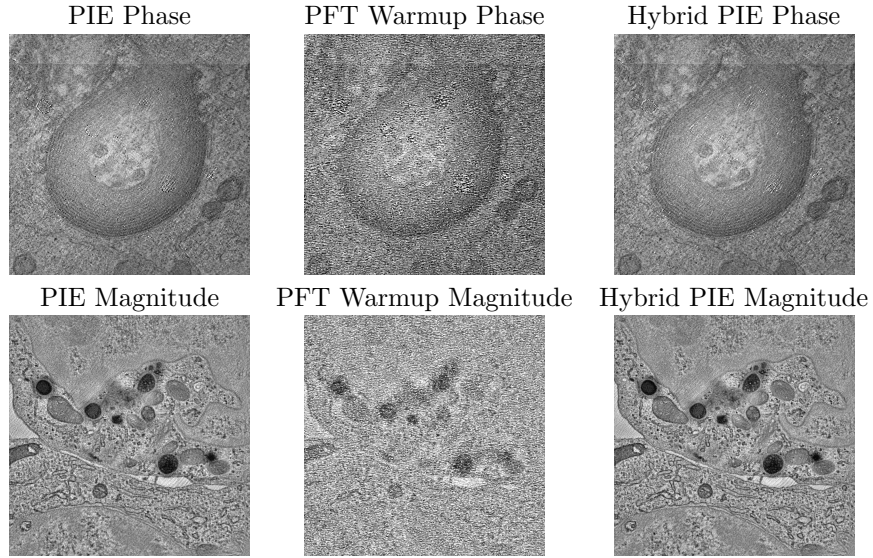


FIGURE 29. Phase and magnitude reconstructions of PIE (first column), PFT-based PIE warmup (second column), and hybrid PIE (third column). These reconstructions are done on illuminations windows with 75% overlap.