ADP: Automatic Differentiation Ptychography

Sushobhan Ghosh
sushobhan@u.northwestern.edu

Youssef S. G. Nashed
ynashed@northwestern.edu

Oliver Cossairt
ollie@eecs.northwestern.edu

Aggelos Katsaggelos
agkk@eecs.northwestern.edu

Northwestern University

Abstract

Ptychography is an imaging technique which aims to recover the complex-valued exit wavefront of an object from a set of its diffraction pattern magnitudes. Ptychography is one of the most popular techniques for sub-30 nanometer imaging as it does not suffer from the limitations of typical lens based imaging techniques. The object can be reconstructed from the captured diffraction patterns using iterative phase retrieval algorithms. Over time many algorithms have been proposed for iterative reconstruction of the object based on manually derived update rules. In this paper, we adapt automatic differentiation framework to solve practical and complex ptychographic phase retrieval problems and demonstrate its advantages in terms of speed, accuracy, adaptability and generalizability across different scanning techniques.

1. Introduction

Nano-scale imaging is central to many research as well as industrial applications. While there are numerous ways to achieve nanometer resolution, X-ray ptychography [1, 2] remains one of the most versatile techniques. This is because electron microscopy is limited to imaging through samples that are no thicker than 1 μm, X-ray microscopy requires lenses, whose aperture limits resolution, and single shot Coherent Diffration Imaging (CDI) [3] requires the samples to have limited physical extents. In X-ray ptychography, far-field coherent diffraction patterns are recorded at the detector, as a finite coherent beam is scanned across the sample with overlap in the spots scanned, as shown in Figure 1.

The diffraction patterns recorded at the detector are the intensities of the propagated object exit wavefront. If the detector lies in the far-field, this exit wavefront undergoes Fraunhofer diffraction, and the propagation function is the Fourier transform. Since the phase of these diffraction patterns is lost, one cannot recover the object wavefront by simply using the inverse Fourier transform. Mathematically, the phase retrieval problem is ill-posed and hence the solution is non-unique [4]. In order to recover the phase, phase retrieval algorithms make use of priors and constraints specific to the problem at hand. In ptychography, the constraints are imposed due to redundancy in the captured data resulting from the partial overlap of neighboring scan points.

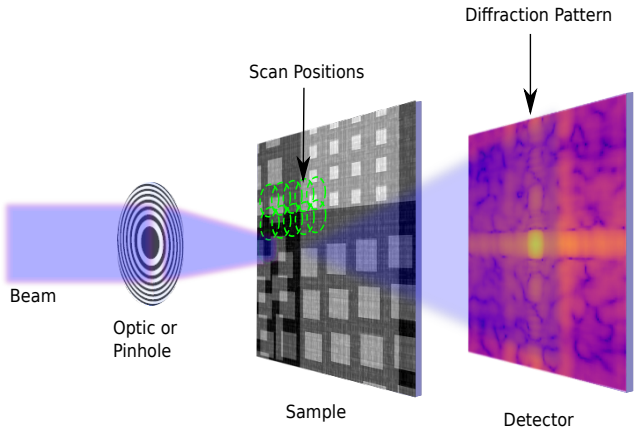


Figure 1: **Schematic of a ptychography setup.** A coherent X-ray beam is focused onto a spot on the sample. A detector is used to record the far field diffracted intensity as the focused beam is scanned across the sample. Ptychography recovers the phase and absorption contrast at a resolution well below the size of the focused beam (e.g., Figure 6 shows 23 nm resolution from a 100 nm probe beam).

In order to recover the lost phase, several phase retrieval algorithms have been proposed. The most popular techniques in phase retrieval use an alternating projections scheme [5], where a randomly initialized complex-valued object wavefront function is iteratively updated by replac-

ing the magnitudes of its diffraction pattern by the measured diffraction magnitudes. Variants of this approach for ptychography are Ptychographical Iterative Engine (PIE) [2] and Difference Map [6].

An alternative to this approach is to formulate the phase retrieval problem as an optimization problem [7]. Here, we establish a cost function that evaluates the error between our measurement and the one that our forward model predicts from our estimate of the complex object field. The cost function is then differentiated with respect to the object field to perform gradient descent optimization. The advantage of gradient based methods is that, based on the constraints of the problem, we can use out of the box gradient optimization techniques like Gauss-Newton [8], conjugate gradient [7], quasi-Newton [9], etc.

A major drawback of this formulation is that the gradient expression for the backward model needs to be explicitly derived every time the cost function is modified. In practice, when performing ptychography experiments, it is often necessary to adapt the cost function to accommodate changes in the experimental setup.

Automatic differentiation (AD) [10], also known as algorithmic differentiation, offers a more general solution to the optimization problem. The key idea is to formulate the cost function using only basic mathematical functions and operations for which analytic derivative functions are known. An AD framework can then be used to automatically find the analytic derivative of the cost function with respect to the independent variables, which in turn allows us to perform optimization using gradient descent. As we show in this paper, Automatic Differentiation Ptychography (ADP) offers a flexible framework to express the forward model for complex ptychography setups. Furthermore, no explicit algorithm to invert the model is necessary since gradient descent optimization is performed automatically using algorithmic differentiation.

The application of algorithmic differentiation to phase retrieval was highlighted by Jurling and Fienup who derived the complex valued elementary operations specific to phase retrieval algorithms by hand [11]. More recently, Nashed et al. successfully applied AD to ptychographic phase retrieval with step-scan and a known probe [12] and highlighted its advantages in terms of convergence and scalabitily to distributed multi-GPU computing architectures.

In this paper, we show that AD can be used as a unifying framework to evaluate how changes in the forward model affect reconstruction performance. Here we focus our study on the effect of unknown probe functions (i.e. probe retrieval) and continuous probe motion (i.e., fly-scan mode) with AD as a general framework to solve these complex ptychographic problems, which we collectively refer to as ADP. Thus, the main contributions of this paper are:

- Develop ADP as a generalized framework (in Tensorflow) to solve for challenging ptychographic models such as simultaneous probe retrieval and flyscan sampling by extending existing AD techniques for ptychography
- Demonstrate the advantages of ADP in terms of speed and accuracy by using optimized minibatch size and learning rate
- Show experimental results of reconstructions from flyscan X-ray ptychography diffraction patterns captured at the Argonne Advanced Photon Source facility

It is important to note that, in this paper, we use the backpropagation learning algorithm to solve an inverse problem, and do not train a deep neural network for a classification or regression task. ADP is implemented using Tensorflow framework, which makes it scalable and easy to use.

2. Background

2.1. Phase Retrieval

Phase retrieval is the process of computationally solving for the phase $\phi(k)$ of a complex signal $A(k)e^{j\phi(k)}$, given only its magnitude $A(k)$. In lensless CDI techniques, the far-field diffraction pattern (Fourier pattern) is recorded at the detector plane. Since detectors can only detect intensity, phase information of the wavefront is lost. Without this phase information, we cannot recover the original image by employing a simple inverse FFT operation. Phase retrieval algorithms are iterative computational techniques which aim to recover this lost phase by imposing some priors or constraints to the problem. Many algorithms have been proposed with different constraints like the Gerchberg-Saxton Error Reduction (ER) algorithm [13] which assumes phase only objects, and the Hybrid input-output algorithm (HIO) [14, 5, 15], which assumes that the object has finite spatial extent. Ptychography is a technique where constraints are imposed by the overlapping scan areas in the object space.

2.2. Ptychography

Ptychography was proposed by Hoppe [1] as a technique to determine the relative phase between different crystalline planes and hence the crystal structure. With the development of advanced detectors, stable X-ray sources and powerful computers, ptychography was adopted as a technique for X-ray microscopy. In X-ray ptychography, a limited area of an extended object is scanned by a probe beam producing a diffraction pattern at the far field detector plane. Figure 1 is a simplified schematic of a ptychography experiment. Each adjacent scan point has to have an overlap in the object space. This data redundancy (each point in object space contributes to multiple diffraction patterns) imposes the constraints required by phase retrieval algorithms to work, effectively turning an under-determined non-linear

system into a determined non-linear system by oversampling.

The forward model for ptychography can be written as:

$$\psi_i(r) = P(r - r_i) \circ O(r). \tag{1}$$

Where complex value object $O(r)$ interacts with a complex valued probe beam $P(r)$ at position r_i to produce a complex valued wavefront $\psi_i(r)$ with $\circ$ denoting multiplication. This wavefront propagates to the far field detector plane, approximated by a Fourier transform, where only the intensity I_i of the propagated wavefront is recorded, such that

$$I_i(k) = |\mathscr{F}[\psi_i(r)]|^2, \tag{2}$$

where $\mathscr{F}$ denotes the Fourier transform from real space r to reciprocal space k. Here r and k are treated as scalar variables, but generalization to 2D image coordinates is straightforward.

2.3. Scanning

In ptychography, a probe beam is scanned across the object and the corresponding diffraction patterns are recorded at the detector. Among the known ptychographic scanning techniques, step-scan and fly-scan are the two popular ones used in practice.

2.3.1 Step-scan

Step scan represents the traditional acquisition mode for ptychography. In step scan, the probe moves in a move-settle-acquire sequence, where the detector does not acquire data while the probe is moving or settling. Since the move-settle time is typically larger compared to the acquire time, especially for small steps, the detector is effectively idle for most of the time. This limits the area that can be scanned in a given period of time and thus step-scan takes a long to image the complete area under investigation.

2.3.2 Fly-scan

In fly-scan [16, 17], the probe and the detector move at a constant velocity along the scan direction relative to the object. The data is captured over the exposure time while the probe is in motion, resulting in an incoherent sum of the diffraction patterns of the scanned area. Since the detector does not have a settle time, fly-scanning is much faster compared to step-scan.

2.4. Current Methods

PIE: Algorithms such as Ptychographical Iterative Engine (PIE) [2] were developed to make use of redundancy in ptychography datasets to iteratively reconstruct the complex object exit wavefront. PIE assumes that the complex illumination function of the probe used for scanning is known. The object is randomly initialized as $O^0(r)$ where $O^n(r)$ is the estimated object after n iterations. $\psi_i^n(r)$ is calculated from $O^n(r)$ using equation 1. Replacing the magnitude of $\mathscr{F}[\psi_i^n(r)]$ by the magnitude of the measured diffraction pattern (given by equation 2) defines:

$$\Psi_i^n(k) = \sqrt{I_i(k)} \frac{\mathscr{F}[\psi_i^n(r)]}{|\mathscr{F}[\psi_i^n(r)]|}. \tag{3}$$

A new $\psi_i^{'n}(r)$ is calculated from $\Psi_i^n(k)$ using the inverse Fourier transform operation, such that

$$\psi_i^{'n}(r) = \mathscr{F}^{-1}[\Psi_i^n(k)]. \tag{4}$$

The residual $(\psi_i^{'n}(r) - \psi_i^n(r))$ is then used to iteratively update the object $O^n(r)$ as follows:

$$O^{n+1}(r) = O^n(r) + \alpha \frac{P_i^*(r - r_i)}{|P_i^*(r - r_i)|^2_{\max}} (\psi_i^{'n}(r) - \psi_i^n(r)), \tag{5}$$

where $*$ is the complex conjugate operator, and α a scalar $\in [0, 1]$ to control the contribution of the residual step.

ePIE: In a ptychography experiment, it is technically challenging to determine the exact complex beam profile of the probe. PIE algorithm assumes that the probe is known, and hence can only be used in situations when a previous estimate of the illumination function is given. A variant of the PIE algorithm, extended PIE or ePIE was developed [18] to simultaneously reconstruct the probe and the object. In ePIE, at each iteration, the probe is updated as

$$P^{n+1}(r) = P^n(r) + \alpha \frac{O_i^{*n}(r + r_i)}{|O_i^{*n}(r + r_i)|^2_{max}} (\psi_i^{'n}(r) - \psi_i^n(r)) \tag{6}$$

Thus, at every iteration, each pixel in the object gets updated as many times as the number of diffraction patterns to which it contributes to, and the probe gets updated as many times as the number of scan points. It is import to note that ePIE requires a good initialization of the probe for good convergence.

Multi-probe ePIE: All CDI phase retrieval methods depend on the quality of the high-resolution speckle information in a diffraction pattern. Because of the beam partial coherence properties, setup stability issues, signal to noise ratio, among other reasons, the measured signal can suffer from reduced fringe contrast due to the incoherent superposition of diffraction patterns. This is often the reason why fly-scan ptychography reconstruction algorithms face challenges in achieving the required resolution. They assume that the probe is focused on a single spot when the diffraction pattern is captured. This problem can be addressed by

employing multiple probe functions, or modes, in the forward model [19]. This forward model is given by the following equations:

$$\psi_{ij}(r) = P_j(r - r_i) \circ O(r) \tag{7}$$

$$I_i(k) = \sum_{j=1}^{Q} |\mathscr{F}[\psi_{ij}(r)]|^2 \tag{8}$$

Where, j is the probe mode index and Q is the number of probe modes.

Multi-probe mode decomposition produces better reconstructions than single-probe for fly-scan datasets [17] at the cost of increasing the number of unknowns by $Q \times sizeof(P)$. It does not, however, constrain the probe estimates to physical constraints imposed by the experimental setup, such as non-Cartesian scan trajectories and fly scan acquisition.

3. Automatic Differentiation

Automatic Differentiation (AD) automatically calculates analytic partial derivatives of any function with respect to the independent variables. Most AD techniques use a *computational graph* to calculate the analytic derivatives, which is a direct application of the chain rule for derivatives in calculus. The key idea in AD is that if the analytic functional form for the derivatives of elementary functions are known, we can use the chain rule, the product of these partial derivatives, to find the analytic derivative function for any complex composition of these functions.

In the last few years, there has been a growing interest in AD, both in academia and industry [20], especially because of its application to the backpropagation algorithm in deep-learning. For many deep-learning applications, it is necessary to calculate the gradients of the cost function across multiple layers (of the order of 100) of mathematical operations. Any numerical differentiation method would quickly run into numerical errors in calculating gradients across multiple operations due to finite floating point precision and the computational load. AD alleviates the problem by calculating the analytic derivative using the chain rule.

In this paper, we use Tensorflow [21], a Python based library developed by Google to find the analytic gradients for complex deep neural network architectures.

3.1. Computational Graph

For calculating the analytic derivative of a composition of functions, the forward model for the function is implemented as a directed, acyclic graph known as *computational graph*, where operators and variables (as well as their derivatives) are contained in the vertices or nodes of the graph. The roots of graph are the independent variables or constants, while the leaves of the graph are the dependent

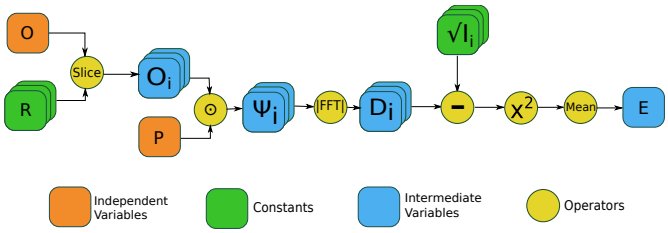


Figure 2: **Computational Graph for ADP with probe retrieval.** Traversal of the data flow graph from left to right expresses calculation of the objective function from the unknown object function $O(r)$ to the objective expressed by $\mathbb{E}$ in Eq. 9. This traversal is equivalent to functional composition of elemental operators (shown in yellow) whose analytic derivatives relative to the independent variables are known. Gradient descent optimization is performed by applying the chain rule to backpropagate partial derivatives through the network.

variables. To find the gradients for the independent variables, the computational graph is traversed in reverse order (from leaves to root), accumulating the gradients at each node according to the chain rule, and eventually updating the root variables using the accumulated gradients. Figure 2 shows the computational graph for step-scan ADP with simultaneous probe retrieval. Here, the probe and object are the independent variables (shown in orange) which are equivalent to weights in neural networks. The diffraction intensities and relative scan shifts are constants (input data) shown in green, while all other variables are intermediate variables. It is important to note that although we use the deep-learning framework, ADP is not a machine learning algorithm for two reasons: 1) it does not need training data, 2) the learned variables do not generalize to other datasets; they define a best-fit forward model to the measured data.

3.2. Automatic Differentiation Ptychography (ADP)

As mentioned in Section 1, one approach for ptychographic reconstruction is to formulate it as an optimization problem, where we minimize a cost function. An important distinction between conventional gradient descent and ADP algorithms is that, for ADP, no explicit knowledge of the gradient is required because the gradient is computed automatically via the chain rule using the computational graph. For ptychography, we formulate the cost function as described in equation 9 for step-scan and equation 11 for fly-scan. The AD framework then calculates the analytic expression for the gradient of the cost function with respect to the independent variables and evaluates it using the values of these variables at an instant. An optimizer then calculates the gradients and iteratively updates the independent variables, here the object $O(r)$ and the probe $P(r)$.

3.3. Optimizer

An optimizer is a function that takes the gradients and/or higher order derivatives and generates the values of updates to be made to the corresponding variables. Over time, many variants of gradient descent have been proposed such as Stochastic Gradient Descent (SGD), AdaGrad, RMSProp, etc. In this paper we use Adam (Adaptive Moment Estimation) [22], which is a modification of the RMSProp optimizer, using running averages on both, gradient and second moment of the gradients.

Every optimizer takes a set of parameters, called hyperparameters, which need to be chosen by the user. This offers the user flexibility and control over convergence in terms of speed and accuracy.

3.4. Minibatch

In ePIE and ADP, at each iteration, updates are calculated for each scan point independently. In ePIE, an update is made to the object after calculating the gradients for each scan point. This allows subsequent gradient calculations to make use of the updates made to the object within the same iteration. Alternatively, we can calculate the updates for multiple scan points in parallel and make a collective update to the object once. Minibatch size (m) for a gradient update is thus the number of scan points used for making a single update.

Using smaller minibatch sizes corresponds to larger number of updates per iteration, and hence requires less number of iterations to converge. On the other hand, each iteration takes longer since updates are made sequentially. Thus, minibatch size is a parameter that trades-off speed for accuracy at each iteration.

It has been shown that the update rules for ePIE can be obtained from minimization of the cost function in equation 9 [23]. Thus, ePIE is a special case of the ADP formulation for ptychography where the minibatch size, $m = 1$.

4. Algorithm

In this section, we describe the forward ADP model for both step-scan and fly-scan. Since ADP is formulated as an optimization problem, we need a cost function corresponding to the forward model.

4.1. Step-scan ADP

Step-scan, as mentioned in Section 2.3.1, is a scanning mode for ptychography, where a beam is focused at one spot while the diffraction pattern is captured. To recover the object $O(r)$, we formulate the cost function as:

$$\mathbb{E} = \frac{1}{m} \sum_{i \in M} \{|\mathscr{F}[\psi_i(r)]| - \sqrt{I_i(k)}\}^2 \tag{9}$$

where $\psi_i(r)$ is defined by equation 1, M contains the indices of a minibatch of size m and $\mathscr{F}$ denotes the Fourier transform.

4.2. Single-mode Fly-scan ADP

Fly-scan, as mentioned in Section 2.3.2, is a scanning mode for ptychography, where a focused probe beam is scanned across the object at constant velocity while a diffraction pattern is captured. This modification in the scanning technique can me modelled by calculating $\psi_{ij}(r)$ according to equation:

$$\psi_{ij}(r) = P(r - (r_i + j\delta)) \circ O(r) \tag{10}$$

where r_i corresponds to the beginning position of the fly-scan corresponding to I_i, $j \in \{1, 2, ...S\}$, where S is the number of diffraction patterns to average over to model fly-scan, $\delta = d/S$, and d is the fly-distance. The cost function for fly-scan is:

$$\mathbb{E} = \frac{1}{m} \sum_{i \in M} \left\{ \sqrt{\sum_{j=1}^{S} |\mathscr{F}[\psi_{ij}(r)]|^2} - \sqrt{I_i(k)} \right\}^2 \tag{11}$$

where $\psi_{ij}(r)$ is defined by equation 10, M contains the indices of a minibatch of size m and $\mathscr{F}$ denotes Fourier Transform.

By comparing the cost functions of equations 1 and 7, we can see that Fly-scan ADP depends on just a single probe P that is shifted to S different locations while multi-mode ePIE uses S different probes. This more accurate forward model reduces the amount of unknowns that must be estimated, and hence produces higher reconstruction quality than multi-probe ePIE.

4.3. Probe Retrieval

As mentioned in Section 2.4, in a ptychography experiment, it is difficult to know the exact complex beam profile of the probe. Thus, we need to simultaneously solve for both, the object and the probe. While the object is initialized randomly, a random initialization of the probe leads to a slow convergence and lower accuracy. In ePIE, this limitation is overcome by making a good initial guess of the probe according to some heuristics. In this paper, we make no assumptions about the probe. The probe is initialized as $P^0(r)$ given by the equation:

$$P^0(r) = \mathscr{F}^{-1}\Big[\sum_{i=1}^{N} \sqrt{I_i(k)}\Big], \tag{12}$$

where N is the number of measured diffraction patterns. We show that this initialization results in better overall performance.

Another practical difficulty in determining the probe and the object simultaneously is that since initially the object is randomly initialized, the gradient updates to the probe in the first few iterations is random. This results in a poor probe estimation, which in-turn affects object reconstruction, leading to slower convergence and lower accuracy. To alleviate this problem, we start updating the probe only after a few updates to the object. This ensures that the probe is updated only after we have a rough estimate of the object, leading to a faster convergence. We have experimented with different starting points for the probe update, and the observation was that the convergence PSNR did not vary much, so long as the probe retrieval starts after convergence without probe retrieval.

5. Experiments

In this section, we perform experiments with both, simulated and real experimental data.

5.1. Metrics

Speed: To compare the computational speed of ADP and ePIE, we run both algorithms on a Nvidia TITAN X GPU, for fixed time.

Accuracy: To compare the quality of reconstruction, we calculate (Peak Signal to Noise Ratio) PSNR between the reconstruction and ground truth images.

For a fair comparison, we use GPU optimized ePIE implementation in Ptycholib [24].

5.2. Simulation

5.2.1 Data

Step-scan: For step-scan simulation, integrated-chip images of size 512×512 are used as object ground truth, with beam diameter of 35×35 and probe size of 128×128. The step size along both scan direction was set to 8 pixels, which corresponds to ptychographic oversampling of 77%.

Fly-scan: For fly-scan simulation, integrated-chip images of size 256×256 are used as object ground truth, with beam diameter of 35×35 and probe size of 64×64.The step size along both direction was set to 8 pixels, which corresponds to ptychographic oversampling of 77%. The object was scanned from left to right, hence the horizontal direction is fly-direction.

5.2.2 Optimization parameters

ADP offers flexibility to choose hyper-parameters which can be used to optimize convergence and trade speed for accuracy. In this paper we focus on 2 hyper-parameters: minibatch and learning rate.

Minibatch: Minibatch offers a trade-off between speed and accuracy. Increasing the minibatch size m results in slower convergence in terms of iteration, but each iteration is relatively fast.

Learning Rate: Learning rate is the length of the step taken at each update. A higher learning rate corresponds to faster convergence, but converges to a lower accuracy.

Figure 4a shows the 2D color contour of PSNR for different values of minibatch m and learning rate lr for step-scan and fly-scan respectively.

Since PIE and its variants (ePIE and multi-probe ePIE) rely on explicit update rules calculated by hand, it can only accommodate special case where $m = 1$. Figure 4b shows that $m = 1$ is sub-optimal. Since ADP is a cost function optimization at each step, we can easily vary the minibatch size by changing the number of scan-points that contributes to the cost function in each update.

5.2.3 Probe retrieval: ePIE vs ADP

In this section we compare the performance of ePIE and ADP with and without probe retrieval. Figure 3a shows the results for ePIE and ADP on step-scan dataset and Figure 3b shows the results for multi-probe ePIE and flyscan ADP on fly-scan dataset. We can see that the reconstruction improves significantly if we simultaneously update the probe and object, especially with ADP (indicated by the sharp jump at 50s and 25s for step-scan and fly-scan respectively). We also observe that without probe updates, the step-scan ADP begins to diverge. It is important to notice that even without probe retrieval, APD outperforms ePIE on both step-scan and fly-scan in terms of PSNR after 200s.

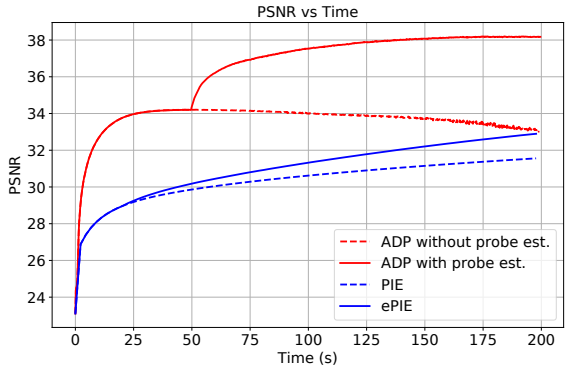
5.2.4 ePIE vs Step-scan ADP

Here, we compare ADP against ePIE on step-scan data of simulated chip images. Both algorithms were run on the same computational resource for 200s. Based on Figure 4a, we choose $m = 64$ and $lr = 0.0045$ for step-scan ADP. Figure 4b shows the PSNR plot for ePIE and step-scan ADP with optimized parameters, as well as with $m = 1$ and $m = N$ (complete dataset) to emphasize the importance of minibatch optimization.

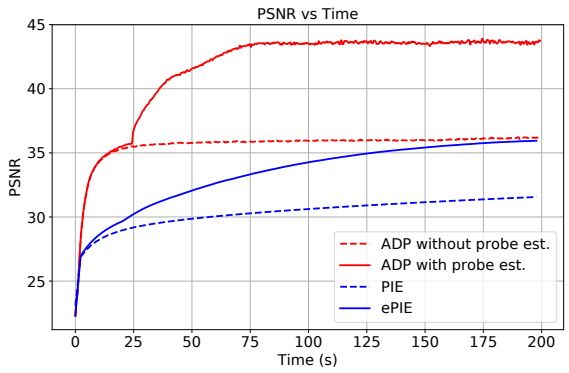
From Figure 4b, we can see that step-scan ADP results in faster convergence as well as 5db higher PSNR at the end of 200s.

5.2.5 Multi-probe ePIE vs single probe fly-scan ADP

In this section, we make a comparative analysis of the Multi-probe ePIE algorithm with single-probe ADP on simulated fly-scan data. Both algorithms were run on the same computational resource for 200s. We choose $m = 16$ and $lr = 0.009$ based on Figure 4a. Figure 4b shows the PSNR plot for multi-probe ePIE and single-probe fly-scan ADP with optimized parameters, as well as with $m = 1$ and $m = N$ (complete dataset).



(a) ADP and ePIE with and without probe estimation.



(b) Single-probe ADP and Multi-probe ePIE with and without probe estimation.

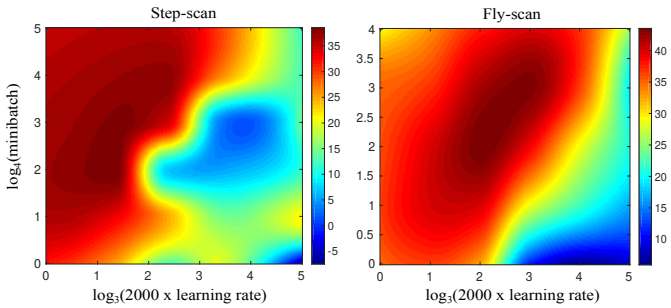
Figure 3: **Comparative PSNR plots for ADP and ePIE on step-scan and fly-scan data.** Solid lines represent with probe retrieval and dashed lines represent without probe retrieval. (a) The comparisons are made on simulated step-scan data. The probe estimation for step-scan ADP starts after 50s. (b) The comparisons are made on simulated fly-scan data. The probe estimation for single-probe fly-scan ADP starts after 25s.

Here we see a clear advantage of ADP as it not only converges faster, but leads to almost 8db higher convergence compared to multi-probe ePIE.

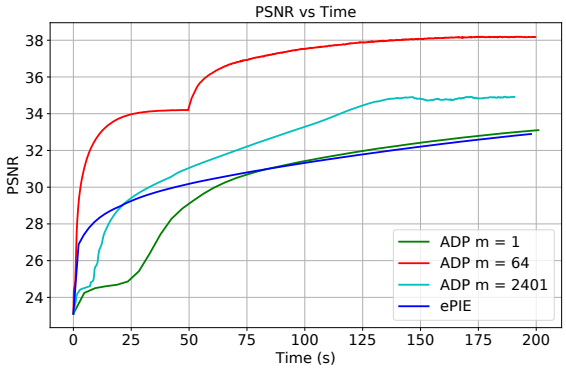
To quantify the performance, Figure 5d and 5e shows the horizontal (fly-direction) and vertical (non-fly direction) line errors (distance from ground truth) for multi-probe ePIE and single-probe ADP. We can clearly see that the error multi-probe ePIE is much higher than that of ADP, justifying the performance.

5.3. Experimental Results

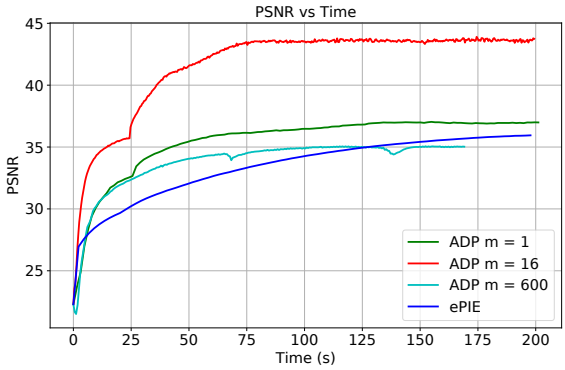
In order to validate our algorithm's performance, we tested it on real data acquired by the Bionanoprobe at the 21-ID-D beamline for the Advanced Photon Source (APS).



(a) 2D PSNR contours for minibatch size vs learning rate for step-scan and flyscan ADP



(b) PSNR vs time for step-scan ADP and ePIE



(c) PSNR vs time for fly-scan ADP and multi-probe ePIE

Figure 4: **Optimal parameter selection criteria for ADP.** (a) PSNR contours for learning rate and minibatch size shows that we can achieve much higher accuracy by tuning the minibatch and learning rate. (b) Optimal minibatch size of 64 converges much faster than ePIE. We also show m=1 and m=2401 (full dataset) for ADP for comparison. (c) Optimal minibatch size of 16 converges much faster than multi-probe ePIE. We also show m=1 and m=600 (full dataset) for flyscan ADP. We see that ADP step-scan shows a 5db improvement, whereas ADP fly-scan shows 8db improvement over ePIE and milti-probe ePIE

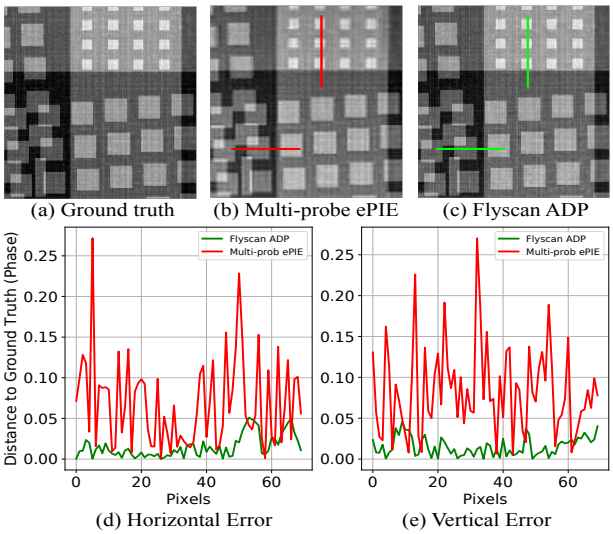


Figure 5: **Qualitative analysis of fly-scan ADP and multi-probe ePIE.** (a) Simulated ground truth chip image (b) multi-probe ePIE reconstruction (c) single-probe ADP reconstruction (d) Horizontal and (e) vertical line plot for distance from ground truth phase in radians

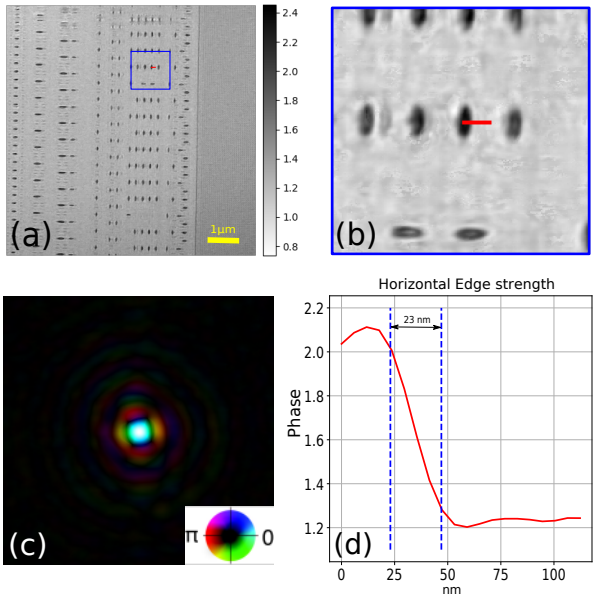


Figure 6: **Experimental results for Hynix DRAM integrated circuit.** (a) Phase reconstruction for experimental diffraction data collected at APS, Argonne. (b) Zoomed region for region in blue box in (a). (c) Reconstructed probe. Phase shown in color while amplitude shown as magnitude. (d) Edge response (10%-90%) of 23 nm, suggesting a resolution of about $\Delta x = 23$ nm.

It is a sample of a 8-Gbit SK Hynix DRAM chip which was fabricated on a 32 nm technology node with an initial wafer thickness of 130 μm. It is a 135×135 Cartesian grid fly-scan data covering 7.9 μm×7.9 μm area. A 10 KeV (1.24 Å) X-ray source with probe beam of diameter 100 nm and a step-size of 60 nm was used, resulting in 40% overlap. A detector with a pixel-size of 172 μm was placed at a distance of 2.1 meters downstream to collect the diffraction patterns. The central 256×256 pixels of the detector data was used for the reconstruction. Such large datasets usually contain experimental noise resulting from finite photon count, positional inaccuracies from scanning stage, beam intensity fluctuations due to air, scattering, changes in temperature and bad detector pixel.

Figure 6a shows the reconstructed IC image using single-probe Fly-scan ADP algorithm. The reconstruction consists of 2.6 Mega-pixels, with each pixel corresponding to 5.91 nm in physical dimensions. We only show the phase reconstruction because amplitude contrast is low due to strong absorption of X-rays. Figure 6d shows that a 23 nm resolution was achieved despite using a 100 nm probe beam.

6. Conclusion

In this paper we presented a generalized ptychographic reconstruction algorithm based on Automatic Differentiation (ADP). We show that ADP has advantages over current state of the art algorithms like PIE (ePIE and multi-probe ePIE), both in terms of speed and accuracy. We also show that using ADP, a more accurate forward model for flyscan can be used to reconstruct sharp features while state-of-the-art reconstruction algorithms like multi-probe ePIE can not. We demonstrate the flexibility of the ADP framework by modifying the forward model, implementing it as a computational graph, and using the same AD algorithm to solve the inverse problem. We also showed that ADP is much easier to adapt to modifications to the forward model, since the solution to the inverse problem is automatically calculated by the computational graph. The algorithms were evaluated on both, synthetic and real experimental data to establish their superior convergence properties, as well as evaluated on a GPU to demonstrate their scaling capabilities with parallelization. Lastly, we showed the performance of ADP with different minibatch sizes to emphasize its flexibility in terms of speed/accuracy trade-off.

7. Future Work

Thick samples require larger flux, which may be achieved by reducing the spatial and temporal coherence of the beam. In the future, we plan to extend ADP to incorporate the partial spatial and temporal coherence of the probe. We also plan to have a comparative analysis of the differ-

ent Ptychography algorithms at different oversampling ratios (diffraction oversampling and ptychographic oversampling) as well as the effect of priors and regularization. Additionally, we also plan to extend ADP to 3D ptychographic reconstruction of a volume, eliminating the need for explicit reconstruction of 2D projections when performing 3D imaging.

8. Acknowledgement

We are thankful to USC for providing access to the virtual testbed to generate ptychography simulation for chip designs. We also gratefully acknowledge the Advanced Photon Source at Argonne National Laboratory for the experimental chip data. The authors of this work were funded by IARPA RAVEN grant /No. 86101156 // FA8650-17-C-9112; National Science Foundation (NSF) CAREER grant IIS-1453192; Defense Advanced Research Projects Agency (DARPA) (REVEAL HR0011-16-C-0028); and Office of Naval Research (ONR) grant N00014-15-1-2735. The Bionanoprobe is funded by NIH/NCRR High End Instrumentation (HEI) grant (1S10RR029272-01) as part of the American Recovery and Reinvestment Act (ARRA).

References

[1] W. Hoppe, "Beugung im inhomogenen Primärstrahlwellenfeld. I. Prinzip einer Phasenmessung von Elektronenbeungungsinterferenzen," *Acta Crystallographica Section A*, vol. 25, pp. 495–501, Jul 1969.

[2] J. M. Rodenburg and H. M. Faulkner, "A phase retrieval algorithm for shifting illumination," *Applied physics letters*, vol. 85, no. 20, pp. 4795–4797, 2004.

[3] J. Miao, P. Charalambous, J. Kirz, and D. S. yre, "An extension of the methods of x-ray crystallogra phy to allow imaging of micron-size non-crystalline specimens," *Nature*, vol. 400, pp. 342–344, 1999.

[4] A. M. J. Huiser and P. van Toorn, "Ambiguity of the phase-reconstruction problem," *Opt. Lett.*, vol. 5, pp. 499–501, Nov 1980.

[5] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.*, vol. 21, pp. 2758–2769, Aug 1982.

[6] P. Thibault, M. Dierolf, A. Menzel, O. Bunk, C. David, and F. Pfeiffer, "High-resolution scanning x-ray diffraction microscopy," *Science*, vol. 321, no. 5887, pp. 379–382, 2008.

[7] M. Guizar-Sicairos and J. R. Fienup, "Phase retrieval with transverse translation diversity: a nonlinear optimization approach," *Opt. Express*, vol. 16, pp. 7264–7278, May 2008.

[8] J. Zhong, L. Tian, P. Varma, and L. Waller, "Nonlinear optimization algorithm for partially coherent phase retrieval and source recovery," *IEEE Transactions on Computational Imaging*, vol. 2, no. 3, pp. 310–322, 2016.

[9] J. Li and T. Zhou, "Numerical optimization algorithm of wavefront phase retrieval from multiple measurements," *arXiv preprint arXiv:1607.01861*, 2016.

[10] L. B. Rall, "Automatic differentiation: Techniques and applications," 1981.

[11] A. S. Jurling and J. R. Fienup, "Applications of algorithmic differentiation to phase retrieval algorithms," *JOSA A*, vol. 31, no. 7, pp. 1348–1359, 2014.

[12] Y. S. Nashed, T. Peterka, J. Deng, and C. Jacobsen, "Distributed automatic differentiation for ptychography," *Procedia Computer Science*, vol. 108, pp. 404–414, 2017.

[13] R. W. Gerchberg and W. O. Saxton, "Practical algorithm for determination of phase from image and diffraction plane pictures," *OPTIK*, no. 2, pp. 237–&.

[14] J. R. Fienup, "Reconstruction of an object from the modulus of its fourier transform," *Opt. Lett.*, vol. 3, pp. 27–29, Jul 1978.

[15] J. R. Fienup, "Reconstruction of a complex-valued object from the modulus of its fourier transform using a support constraint," *J. Opt. Soc. Am. A*, vol. 4, pp. 118–123, Jan 1987.

[16] J. N. Clark, X. Huang, R. J. Harder, and I. K. Robinson, "Continuous scanning mode for ptychography," *Optics letters*, vol. 39, no. 20, 2014.

[17] J. Deng, Y. S. Nashed, S. Chen, N. W. Phillips, T. Peterka, R. Ross, S. Vogt, C. Jacobsen, and D. J. Vine, "Continuous motion scan ptychography: characterization for increased speed in coherent x-ray imaging," *Optics express*, vol. 23, no. 5, pp. 5438–5451, 2015.

[18] A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," *Ultramicroscopy*, vol. 109, no. 10, pp. 1256–1262, 2009.

[19] P. Thibault and A. Menzel, "Reconstructing state mixtures from diffraction measurements," *Nature*, vol. 494, no. 7435, pp. 68–71, 2013.

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[21] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] C. Yang, J. Qian, A. Schirotzek, F. Maia, and S. Marchesini, "Iterative Algorithms for Ptychographic Phase Retrieval," *ArXiv e-prints*, May 2011.

[24] Y. S. Nashed, D. J. Vine, T. Peterka, J. Deng, R. Ross, and C. Jacobsen, "Parallel Ptychographic Reconstruction," *Optics Express*, vol. 22, no. 26, pp. 32082–32097, 2014.