

# PtyRAD: A High-performance and Flexible Ptychographic Reconstruction Framework with Automatic Differentiation

Chia-Hao Lee<sup>1,\*</sup> Steven E. Zeltmann<sup>2</sup> Dasol Yoon<sup>1,3</sup> Desheng Ma<sup>1</sup>  
and David A. Muller<sup>1,4,\*</sup>

<sup>1</sup>School of Applied and Engineering Physics, Cornell University, Ithaca, New York, USA, <sup>2</sup>Platform for the Accelerated Realization, Analysis, and Discovery of Interface Materials, Cornell University, Ithaca, New York, USA, <sup>3</sup>Department of Materials Science and Engineering, Cornell University, Ithaca, New York, USA and

<sup>4</sup>Kavli Institute at Cornell for Nanoscale Science, Cornell University, Ithaca, New York, USA

\*Corresponding authors. [chia-hao.lee@cornell.edu](mailto:chia-hao.lee@cornell.edu), [david.a.muller@cornell.edu](mailto:david.a.muller@cornell.edu)

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

## Abstract

Electron ptychography has recently achieved unprecedented resolution, offering valuable insights across diverse material systems, including in three dimensions. However, high-quality ptychographic reconstruction is computationally expensive and time consuming, requiring a significant amount of manually tuning even for experts. Additionally, essential tools for ptychographic analysis are often scattered across multiple software packages, with some advanced features available only in costly commercial software like MATLAB. To address these challenges, we introduce PtyRAD (Ptychographic Reconstruction with Automatic Differentiation), an open-source software framework offers a comprehensive, flexible, and computationally efficient solution for electron ptychography. PtyRAD provides seamless optimization of multiple parameters—such as sample thickness, local tilts, probe positions, and mixed probe and object modes—using gradient-based methods with automatic differentiation (AD). By utilizing PyTorch’s highly optimized tensor operations, PtyRAD achieves up to a 24× speedup in reconstruction time compared to existing packages without compromising image quality. In addition, we propose a real-space depth regularization, which avoids wrap-around artifacts and can be useful for twisted two-dimensional (2D) material datasets and vertical heterostructures. Moreover, PtyRAD integrates a Bayesian optimization workflow that streamlines hyperparameter selection. We hope the open-source nature of PtyRAD will foster reproducibility and community-driven development for future advances in ptychographic imaging.

**Key words:** PtyRAD, ptychography, automatic differentiation, phase retrieval, open source

## Introduction

Ptychography has emerged as a high-resolution, dose-efficient phase retrieval technique by computationally reconstructing both the object and the illumination from a series of diffraction patterns (Rodenburg and Maiden (2019)). Originally proposed by Hoppe (1969) primarily to solve the phase problem in crystallography using overlapping microdiffraction patterns, its ability to retrieve phase information has been recognized as a potential approach to overcome the lens aberrations that limit the resolution of electron microscopes (Rodenburg et al. (1993); Nellist et al. (1995); Humphry et al. (2012); Putkunz et al. (2012)) and found widespread application for lensless imaging with X-rays (Chapman and

Nugent (2010); Rodenburg et al. (2007)). For a comprehensive review of the early state of the field, we refer readers to Rodenburg and Maiden (2019).

Recently, electron ptychography has achieved resolution well beyond the diffraction limit, reaching sub-0.5 Å in atomically thin 2D materials (Jiang et al. (2018); Nguyen et al. (2024)). Such advances and a renewed interest in the method were driven by high-dynamic-range and high-speed direct electron detectors (Tate et al. (2016); Philipp et al. (2022); Zambon et al. (2023); Ercius et al. (2024)) that can simultaneously resolve bright direct beams alongside weak high-angle scattering while overcoming drift and instrument instability. Another key development is multislice electron ptychography (MEP), which adopts the multislice formalism

(Maiden et al. (2012b); Van den Broek and Koch (2013); Godden et al. (2014); Tsai et al. (2016)) to relax the specimen thickness limitation. By explicitly modeling the multiple scattering and the evolution of the probe as it propagates through the specimen, MEP achieves lateral resolution primarily limited by atomic vibrations rather than the imaging system (Chen et al. (2021)), while also enabling depth-resolved information to be extracted from a single scan (Gao et al. (2017); Chen et al. (2021); Dong et al. (2024)). Most importantly, such three-dimensional (3D) reconstructions are less susceptible to channeling artifacts, which often hinder defect localization in thick crystals when using other optical sectioning techniques with annular darkfield (Kimoto et al. (2010)), annular brightfield (Gao et al. (2018)), or differential phase contrast (Close et al. (2015)) imaging.

Reconstruction algorithms are essential parts of ptychography and can be broadly classified into direct and iterative approaches. Direct methods provide fast reconstructions for live imaging and beam sensitive applications (Strauch et al. (2021); O’Leary et al. (2020)), but these methods do not incorporate scattering outside the objective aperture so the resolution is limited to twice the illumination angle, nor do they correct for channeling in the sample (Rodenburg and Bates (1992); Rodenburg et al. (1993); Pennycook et al. (2015)). Recently, machine-learning-based techniques have been explored as alternative direct solvers (Cherukara et al. (2020); Friedrich et al. (2022); Chang et al. (2023)). While these methods offer significant speed advantages over iterative approaches, their performance is inherently tied to the training data, often limiting their applicability to a narrower range of experimental conditions and sample types.

In contrast, iterative algorithms refine the object and probe estimates through successive steps until the model agrees with the measured diffraction data. This approach provides the flexibility to incorporate more complex scattering processes and enables the reconstruction of details beyond the diffraction limit. Various iterative ptychographic algorithms are inspired by or adapted from classical phase retrieval methods (Gerchberg and Saxton (1972); Fienup (1978, 1982); Luke (2004); Elser (2003)), with the extended ptychographic iterative engine (ePIE) (Maiden and Rodenburg (2009)) and the gradient-based least-squares maximum likelihood framework (LSQML) (Odstrcil et al. (2018)) emerging as the most widely used approaches today.

One particular advantage of iterative methods is their ability to incorporate essential features for modeling experimental imperfections and complex physical models—such as position correction (Maiden et al. (2012a)), mixed-state probe retrieval (Thibault et al. (2009); Thibault and Menzel (2013); Odstrcil et al. (2016)), and multislice object modeling (Tsai et al. (2016)). These capabilities have made iterative approaches the dominant choice for high-resolution ptychography, and several open-source iterative ptychography packages (Enders and Thibault (2016); Wakonig et al. (2020); Jiang (2020); Savitzky et al. (2021); Loetgering et al. (2023); Varnavides et al. (2023)) have been developed, each implementing different features and tailored to specific use cases. A more comprehensive list of open-source ptychography packages is given in SI Table S1.

Despite their strengths, a key limitation of the existing iterative packages is that efficient reconstructions are only possible when analytical update rules are available for the forward model. Introducing new optimizable parameters often require either manually deriving the update steps, estimating the gradients in a computationally expensive finite-difference manner, or refining them

as hyperparameters for the entire reconstruction task. Therefore, many recent ptychography implementations (Kandel et al. (2019); Du et al. (2020, 2021); Seifert et al. (2021); Guzzi et al. (2022); Diederichs et al. (2024); Wu et al. (2024)), primarily developed in the X-ray community, have adopted automatic differentiation (AD) (Rall (1981)) for gradient computation. The AD algorithm systematically applies the chain rule to calculate gradients from any composition of differentiable operations, making it an ideal framework for rapid prototyping new forward models or building on existing models.

Here, we introduce PtyRAD (Ptychographic Reconstruction with Automatic Differentiation), an iterative gradient descent ptychography package that harnesses AD to provide a flexible and efficient approach to ptychographic reconstructions. PtyRAD is an open-source framework developed with the following guiding principles:

1. Accessibility: Fully implemented in Python, with comprehensive documentation.
2. Performance: Efficient reconstruction pipelines, supporting single and multi-GPU acceleration on various platforms.
3. Flexibility: A modular architecture allowing easy customization of forward models, optimization schemes, and physical constraints.
4. Comprehensive Workflow: A complete suite of tools covering preprocessing, reconstruction, and hyperparameter tuning, streamlining the entire reconstruction pipeline.

We first briefly discuss the major components of PtyRAD and our implementation of the mixed-state multislice ptychography model. We then provide detailed benchmarks on published datasets, comparing computational efficiency and reconstruction quality with other commonly-used electron ptychography packages. We further demonstrate how incorporating constraints and regularizations improves the reconstruction quality, including an enhanced depth regularization approach for non-periodic structures. Additionally, we introduce a fast and effective method for hyperparameter selection. Lastly, we summarize key contributions and provide potential future directions.

## Implementation of PtyRAD

PtyRAD is implemented in Python, with its core AD engine powered by PyTorch (Paszke et al. (2019)), leveraging efficient GPU-accelerated tensor operations. Figure 1 illustrates major components of PtyRAD, including a unified forward model supporting mixed probe and object states with multislice; AD-powered gradient-based optimization of probe positions, object misorientations, and slice thickness; and integrated physical constraints to regularize reconstructions. PtyRAD provides an end-to-end pipeline, from initial data preprocessing to final reconstruction with efficient hyperparameter tuning, enabling researchers to rapidly prototype new algorithms, benchmark existing approaches, and foster further developments. See SI Figure S1 for a more detailed workflow diagram including the hyperparameter tuning.

## Physical model

PtyRAD implements the multislice (Tsai et al. (2016)) and mixed-state ptychography formalisms (Thibault and Menzel (2013)) with an adaptive multislice propagator (Kirkland (2010); Sha et al. (2022)). The forward model  $f$  describes the physical process of how

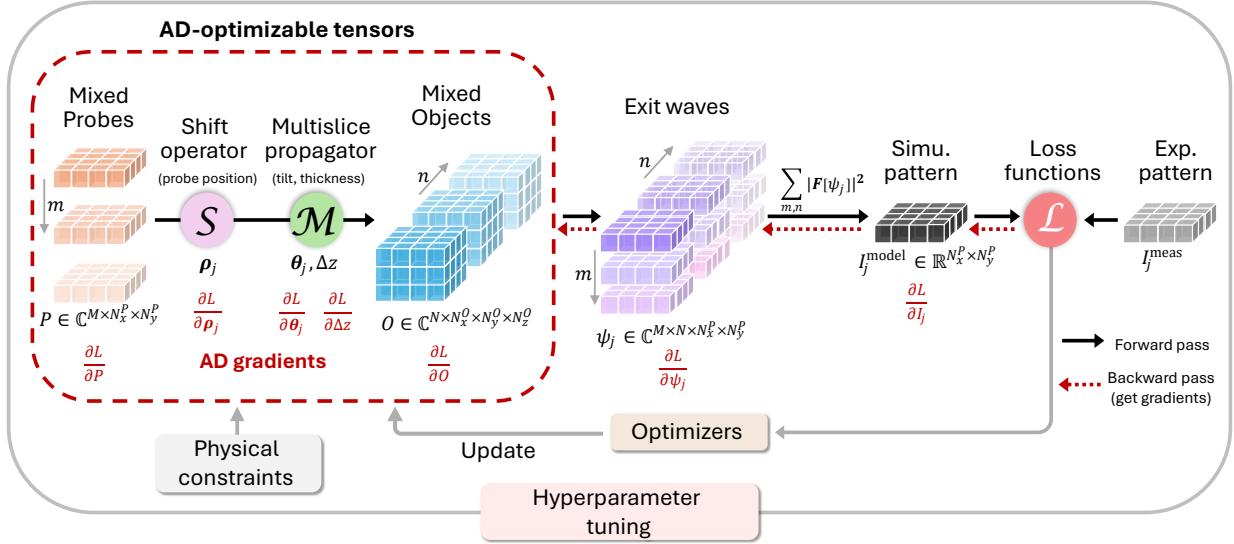


Fig. 1: Schematic of the PtyRAD framework for ptychographic reconstruction using automatic differentiation (AD). The forward model simulates ptychographic measurements at each scan position  $\rho_j$  by first applying the shift operator to the probe ( $S_j P = P_j$ ), and then propagate the shifted probe  $P_j$  through the multislice object patch  $O_j$  using the adaptive multislice propagator  $\mathcal{M}$ , parameterized by local tilt ( $\theta_j$ ) and slice thickness ( $\Delta z$ ).  $j$  is the labeling index used to distinguish different probe positions and associating diffraction patterns. The probe and object are both represented in mixed-state formulation with  $M$  probe modes and  $N$  object modes, resulting in a set of 4D mixed-state exit waves  $\psi_j \in \mathbb{C}^{M \times N \times N_x^P \times N_y^P}$ , which are subsequently reduced to a simulated diffraction pattern  $I_j^{\text{model}} \in \mathbb{R}^{N_x^P \times N_y^P}$  by incoherently summing all states. The simulated pattern is then compared with the experimental pattern  $I_j^{\text{meas}}$  through loss functions  $\mathcal{L}$ . PtyRAD leverages AD to compute gradients (marked in red text) via the backward pass across all participating tensors, enabling the simultaneous optimization of mixed probes, mixed objects, probe positions, tilts, and thickness using advanced PyTorch optimizers while enforcing physical constraints at each iteration. Additionally, PtyRAD seamlessly integrates a hyperparameter tuning workflow to search for promising initial parameters for each reconstruction trial.

the probe  $P$ , object  $O$ , probe position  $\rho_j$ , object misorientation  $\theta_j$ , and slice thickness  $\Delta z$ <sup>1</sup> interact with each other to generate the modeled diffraction intensities  $I_j^{\text{model}}$ .

$$I_j^{\text{model}}(\mathbf{k}) = f(P, O, \rho_j, \theta_j, \Delta z) \quad (1)$$

The index  $j$  is used to label each probe position and the associated diffraction pattern. The variables carrying the subscript can have distinct values for each index.  $j$  ranges from 1 to  $N_{\text{tot}}$ , where  $N_{\text{tot}}$  is the total number of probe positions. All of the model parameters in Equation 1 are AD-optimizable.

Following the mixed state formalism (Thibault and Menzel (2013)), the probe  $P$  and object  $O$  arrays are represented as mixtures of mutually incoherent states (Figure 1). Specifically, the probe is given by  $P \in \mathbb{C}^{M \times N_x^P \times N_y^P}$ , with individual states denoted as  $P^{(1)}, P^{(2)}, \dots, P^{(M)}$  for  $M$  probe modes. Similarly, the object is represented as  $O \in \mathbb{C}^{N \times N_x^O \times N_y^O \times N_z^O}$ , with its modes denoted as  $O^{(1)}, O^{(2)}, \dots, O^{(N)}$  for  $N$  object states. Here,  $N_x$ ,  $N_y$ , and  $N_z$  along with their superscripts, denote the number of pixels along each dimension for the probe and object arrays<sup>2</sup>. Since each state

is assumed incoherent with the others, the total diffraction intensity for pattern  $j$ , denoted as  $I_j^{\text{model}}$ , is simply the summation of diffraction intensities produced from each probe and object mode combination  $I_j^{(m)(n)}$ . Individual intensity contribution  $I_j^{(m)(n)}$  is modeled by the squared modulus of the Fourier-transformed exit waves  $|\mathcal{F}[\psi_j^{(m)(n)}]|^2$ , where  $m \in \{1, \dots, M\}$  and  $n \in \{1, \dots, N\}$ .

$$\begin{aligned} I_j^{\text{model}}(\mathbf{k}) &= \sum_{m=1}^M \sum_{n=1}^N I_j^{(m)(n)} \\ &= \sum_{m=1}^M \sum_{n=1}^N |\mathcal{F}[\psi_j^{(m)(n)}(\mathbf{r})]|^2 \end{aligned} \quad (2)$$

The Fourier transform ( $\mathcal{F}$ ) describes the far-field diffraction as the real-space exit wave propagates to the electron detector, which records the diffraction pattern in reciprocal space.  $\mathbf{r} = (r_x, r_y)$  and  $\mathbf{k} = (k_x, k_y)$  represent the real and reciprocal space coordinates, respectively.

Each exit wave  $\psi_j^{(m)(n)}$  is computed using the standard multislice algorithm detailed in Chapter 6 of Kirkland (2010). For each probe position  $\rho_j$ , an object patch  $O_j^{(n)}$ , centered on  $\rho_j$ , is cropped from the full object array  $O^{(n)}$ . Since object cropping is limited to integer pixel positions, we apply a shift operator to account for sub-px shifts of probe relative to the object patch. This is expressed as  $S_{\Delta \mathbf{r}_j} P^{(m)} = P_j^{(m)}$ , where  $S_{\Delta \mathbf{r}_j}$  is the shift operator given

<sup>1</sup> While  $\Delta z$  is often colloquially referred as “slice thickness,” the multislice approach treats the slices as infinitely thin and  $\Delta z$  is the separation distance between slices.

<sup>2</sup>  $O$  and  $P$  may have different sizes, as the size of  $P$  is most conveniently fixed by the size of the diffraction patterns, while the size of  $O$  is governed by the field of view of the scan.

$\Delta\mathbf{r}_j$ , and  $\rho_j - \text{round}(\rho_j) = \Delta\mathbf{r}_j$  represents the sub-px shift<sup>3</sup>. This ensures that  $O_j^{(n)}$  and  $P_j^{(m)}$  share the same real-space sampling and lateral extent of  $N_x^P \times N_y^P$ , while  $O_j^{(n)} \in \mathbb{C}^{N_x^P \times N_y^P \times N_z^O}$  has an additional depth dimension with  $N_z^O$  slices. Here,  $O_{j,1}^{(n)}$  denotes the first slice of the object, while  $O_{j,N_z^O}^{(n)}$  denotes the last slice.

The multislice calculation is then done by sequentially transmitting through object slices  $O_{j,1}^{(n)}, O_{j,2}^{(n)}, \dots, O_{j,N_z^O}^{(n)}$ , while propagating with the multislice propagator  $\mathcal{M}_{\theta_j, \Delta z}$  between each slice.

$$\psi_j^{(m)(n)} = O_{j,N_z^O}^{(n)} \cdots \mathcal{F}^{-1} [\mathcal{M}_{\theta_j, \Delta z} \mathcal{F} [O_{j,1}^{(n)} P_j^{(m)}]] \quad (3)$$

The multislice propagator depends on the slice thickness  $\Delta z$  and includes the local mistilt  $\theta_j = (\theta_{j,x}, \theta_{j,y})$ , following equation 6.99 in Kirkland (2010).

$$\mathcal{M}_{\theta_j, \Delta z}(\mathbf{k}) = \exp [-i\pi\lambda|\mathbf{k}|^2\Delta z + 2\pi i\Delta z(k_x \tan \theta_{j,x} + k_y \tan \theta_{j,y})] \quad (4)$$

### Parameter specification

Reconstruction parameters of PtyRAD are defined via a configuration file, which can be provided in .py, .yaml, or .json formats, or reconstructions can be run interactively via a Jupyter notebook. This flexibility accommodates both scripted and exploratory workflows, allowing users to easily modify reconstruction settings and experiment with different configurations while ensuring reproducibility.

### Data import

PtyRAD supports raw datasets with multiple formats, including .raw, .hdf5, .mat, .tif, and .npy, ensuring broad compatibility with different experimental setups. Additionally, reconstructed models (object, probe, and probe positions) from other ptychographic reconstruction packages, such as PtychoShelves/fold\_slice (Wakonig et al. (2020); Jiang (2020)) and py4DSTEM (Savitzky et al. (2021); Varnavides et al. (2023)), can be seamlessly imported as well. This feature facilitates cross-comparisons and iterative refinement, making PtyRAD suitable for integrating with existing workflows.

### Preprocessing and initialization

PtyRAD provides a wide variety of preprocessing functions for the imported 4D-STEM data, such as permutation, reshaping, flipping, transposing, cropping, padding, and resampling. Particularly, the padding and resampling can be done in an “on-the-fly” manner, which greatly reduce the required GPU memory. Additionally, PtyRAD also implements features to conveniently apply Poisson noise, detector blur, and partial spatial coherence on perfect simulated data to quickly explore different experimental conditions without generating and storing redundant datasets. By default, the object is initialized with unit amplitude and small random phase perturbations uniformly sampled from the range  $[0, 10^{-8}]$ . This produces an object with flat amplitude and near-zero phase, which serves as a neutral starting point for optimization.

<sup>3</sup> Position correction is achieved by optimizing  $\Delta\mathbf{r}_j$  through the shift operator because rounding and cropping operations are not differentiable.

### PyTorch model and AD optimization

Once the data is prepared, PtyRAD constructs a PyTorch model and imports the data as PyTorch tensors. We choose PyTorch because it provides a fully integrated framework with GPU acceleration, automatic differentiation (AD), a wide range of optimization algorithms, and an extensive toolkit for data processing, all backed by a large and active community. Its ease of use and flexibility have led to its wide adoption in both machine learning and scientific computing, enabling rapid development and excellent extensibility.

A key motivation for the choice of PyTorch for ptychographic reconstruction is its support for automatic differentiation (AD) (Rall (1981)), which efficiently computes exact gradients by decomposing complex operations into elementary components and systematically applying the chain rule, making AD the backbone of backpropagation in modern machine learning. AD allows PtyRAD to flexibly incorporate sophisticated physics models, such as multiple scattering and partial coherence in a simple and unified way. Unlike conventional methods that require manually deriving update steps, AD eliminates this burden entirely: one only needs to define the forward model and the loss function, and all gradient computations follow automatically. This makes AD an ideal framework for incorporating new optimizable parameters and rapidly prototyping novel reconstruction algorithms.

### AD-optimizable parameters

Currently, PtyRAD implements six AD-optimizable parameters (Equation 1), including object amplitude, object phase, object tilt, probe, probe position, and slice thickness. Note that PtyRAD represents the complex object function using two independent real-valued arrays for amplitude and phase, which are recombined into a complex transmission function during the forward pass, while allowing their learning rates to be specified separately. This design allows independent control over the learning rate and onset—i.e., the iteration at which specific parameters begin optimization—enabling precise tuning of the reconstruction process. For example, one may fix the object amplitude at 1 for a pure phase object approximation or delay probe and position optimization until a rough object structure is retrieved. To improve convergence and stability, it is often beneficial to introduce optimizable parameters gradually rather than all at once. Since the computational cost of AD scales with the number of active parameters, starting with a smaller set and progressively refining the reconstruction in a pyramidal approach balances efficiency and accuracy.

### Optimizers and loss functions

PtyRAD supports all 14 gradient-based optimization algorithms provided by PyTorch, with Adam (Kingma and Ba (2014)) as the default choice due to its adaptive learning rate and robustness in handling noisy gradients.

In addition to optimizer selection, PtyRAD permits flexibility in the construction of the loss function. We primarily utilize the negative log likelihood functions,  $\mathcal{L}_{\text{Gaussian}}$  and  $\mathcal{L}_{\text{Poisson}}$ , which are adapted from the original derivation in Thibault and Guizar-Sicairos (2012), assuming Gaussian or Poisson noise statistics for the measured data, respectively.

$$\mathcal{L}_{\text{Gaussian}} = \frac{\sqrt{\left\langle (I_{\text{model}}^p - I_{\text{meas}}^p)^2 \right\rangle_{\mathcal{D}, \mathcal{B}}}}{\langle I_{\text{meas}}^p \rangle_{\mathcal{D}, \mathcal{B}}} \quad (5)$$

$$\mathcal{L}_{\text{Poisson}} = - \frac{\left\langle I_{\text{meas}}^p \log(I_{\text{model}}^p + \epsilon) - I_{\text{model}}^p \right\rangle_{\mathcal{D}, \mathcal{B}}}{\langle I_{\text{meas}}^p \rangle_{\mathcal{D}, \mathcal{B}}} \quad (6)$$

Here,  $I_{\text{model}}$  and  $I_{\text{meas}}$  denote the modeled and measured diffraction pattern. The patterns are raised to a chosen power  $p$  when computing the loss, with default choices of  $p = 0.5$  for  $\mathcal{L}_{\text{Gaussian}}$  and  $p = 1$  for  $\mathcal{L}_{\text{Poisson}}$ . The  $\langle \cdot \rangle$ s represents averaging over a certain dimension  $\mathcal{S}$ , where  $\mathcal{D}$  denotes the detector dimension, and  $\mathcal{B}$  denotes the batch<sup>4</sup>, and  $\mathcal{R}$  denotes the spatial dimension. We added  $\epsilon = 10^{-6}$  into the  $\mathcal{L}_{\text{Poisson}}$  formula for numerical stability.

Additionally, we have implemented a PACBED loss, which promotes consistency with the Position-Averaged Convergent Beam Electron Diffraction (PACBED) pattern (LeBeau et al. (2010)), and an explicit sparsity-promoting regularization  $\mathcal{L}_{\text{sparse}}$  containing the  $L^p$ -norm of the object phase (Candes and Tao (2006); Tibshirani (1996)). By default, we use  $p = 1$ , corresponding to L1 regularization. We are also interested in extending PtyRAD with L0-type sparsity constraints using Fourier-domain thresholding for low dose ptychography (Moshtaghpour et al. (2025)) in future versions.

$$\mathcal{L}_{\text{PACBED}} = \sqrt{\left\langle \left( \langle I_{\text{model}} \rangle_{\mathcal{B}}^p - \langle I_{\text{meas}} \rangle_{\mathcal{B}}^p \right)^2 \right\rangle_{\mathcal{D}}} \quad (7)$$

$$\mathcal{L}_{\text{sparse}} = \langle |O_p|^p \rangle_{\mathcal{R}, \mathcal{B}}^{\frac{1}{p}} \quad (8)$$

The objective function for optimization is given by a weighted combination of the above terms, with user-controlled weights  $w$

$$\mathcal{L}_{\text{total}} = w_1 \mathcal{L}_{\text{Gaussian}} + w_2 \mathcal{L}_{\text{Poisson}} + w_3 \mathcal{L}_{\text{PACBED}} + w_4 \mathcal{L}_{\text{sparse}} \quad (9)$$

### Physical constraints

Inverse imaging problems are inherently ill-posed, making it essential to incorporate physical constraints to help regularize the optimization process, mitigate artifacts, and enforce physically meaningful reconstructions. As shown in Figure 1, these constraints are applied to the probe, object, and tilt arrays at each iteration after taking the gradient descent step. PtyRAD provides a diverse set of physical constraints, including orthogonalization of the mixed-state probes (Thibault and Menzel (2013)); cutoff mask to constrain the probe in Fourier space; object blurring in real and reciprocal space to ensure stability in multislice reconstruction; and a series of thresholding, positivity, and complex relation (Clark et al. (2010); Xu et al. (2024)) that regularize the object amplitude and phase to enforce physically reasonable solutions with improved interpretability, particularly in multislice reconstructions. Our framework allows each constraint to be applied at set intervals during reconstruction and with relaxation to permit partial deviation from the regularization condition.

### Hyperparameter tuning

Convergence in iterative optimization problems is highly sensitive to algorithmic parameters such as batch sizes, learning rates, and other configurational settings which are generally referred to as hyperparameters (Bergstra et al. (2011)). Automatic parameter selection based on Bayesian optimization (Cao et al. (2022))

has been shown to provide faster and more optimal choice of these parameters as compared to hand-tuning by human experts. Moreover, experimental parameters such as specimen thickness and crystal tilts are often imprecisely known, while others—such as the semi-convergence angle, probe aberrations, and scan distortion—are challenging to optimize via gradient descent simultaneously with the object reconstruction. Bayesian optimization provides a better balance of exploration and exploitation for parameters that have many local minima or complex coupling to the object reconstruction trajectory, and so has been widely implemented for optimization of this class of parameters. PtyRAD uses Optuna (Akiba et al. (2019)), a versatile hyperparameter tuning framework that offers a wide range of algorithms beyond Bayesian optimization. In addition, Optuna provides efficient pruning algorithm and distributed optimization capability to search the parameter space even more efficiently. The high-level and non-AD-optimizable parameters of the reconstruction are refined by the hyperparameter tuning process which encloses the AD reconstruction loop. SI Table S2 summarizes the tunable hyperparameters currently implemented in PtyRAD, including batch size, learning rates, defocus, and more.

## Results and Discussion

In order to demonstrate the performance of PtyRAD, we have performed a series of benchmarking, speed, and convergence tests using publicly available ptychography datasets. We then explore the impact of the regularizations and physical constraints available for improving reconstruction quality as well as the performance of the hyperparameter tuning workflow for rapidly determining optimal reconstruction parameters.

### Benchmarking on published datasets

In order to benchmark the performance of PtyRAD, we performed multislice ptychography reconstructions from a selection of publicly-available datasets from the literature (Li et al. (2025); Zhang et al. (2023); Nguyen et al. (2024); Chen et al. (2021)), spanning a wide range of electron doses and sample thicknesses. The resulting depth-summed phase images are shown in Figure 2a–d with their corresponding fast Fourier transform (FFT) power spectra in Figures 2e–h. See SI Table S3 for the major experimental and reconstruction parameters. Full details on the experimental conditions for each dataset are available in the original references, and PtyRAD input files to reproduce these reconstructions are provided in our Zenodo record.

To demonstrate reconstruction of very low-dose data from radiation sensitive materials, we show in Figure 2a a reconstruction of the dataset taken from a metal-organic framework MOSS-6 at  $100 \text{ e}^-/\text{\AA}^2$  reported by Li et al. (2025). At such low electron dose, the resolution is essentially dose-limited, and we observe the same information limit as reported in the original paper and are similarly able to resolve individual atoms in the linker nodes. The ability to reconstruct this dataset demonstrates the robustness of the AD approach to experimental data with severe shot noise. A dataset taken from the zeolite ZSM-5 at a higher electron dose of  $3500 \text{ e}^-/\text{\AA}^2$  is shown in Figure 2b, using the original data taken from Zhang et al. (2023). Here the resolution is also dose-limited, but the higher dose permits an information limit of 86 pm, consistent with the resolution reported in the original work.

<sup>4</sup> Iterative ptychographic reconstructions often group diffraction patterns as  $I_B \in \mathbb{R}^{B \times N_x \times N_y}$  for efficient GPU utilization.  $B$  is often called the “batch size.”

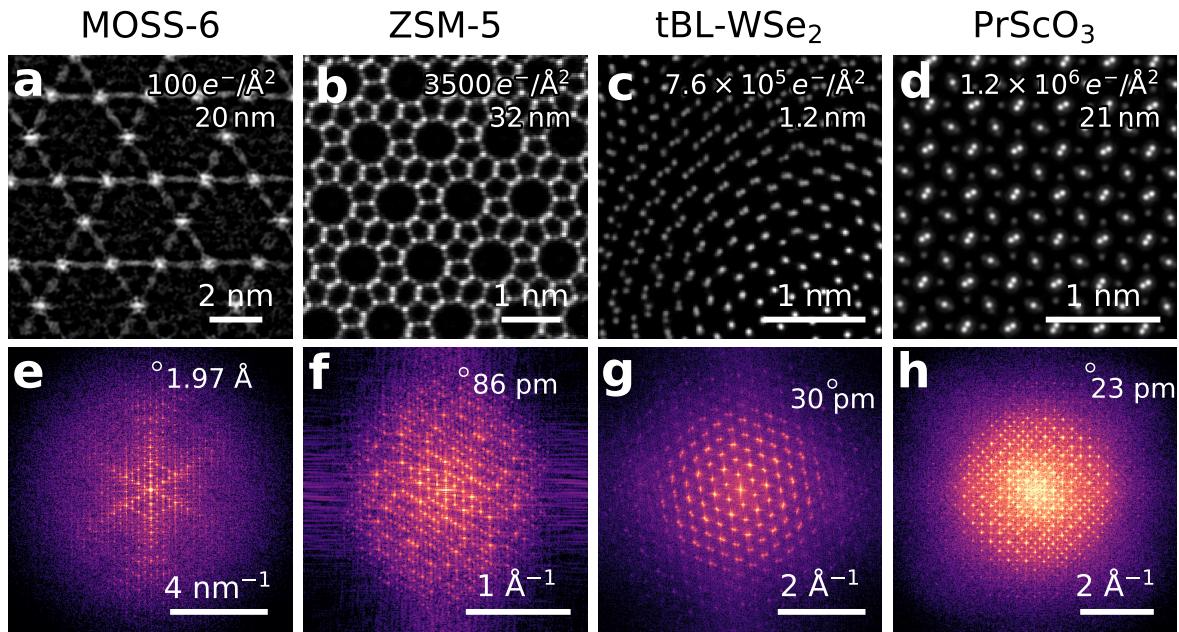


Fig. 2: (a–d) Ptychographic phase images reconstructed using PtyRAD. These examples are from publicly available datasets acquired under various experimental conditions and material systems, including a metal-organic framework (MOSS-6), a zeolite (ZSM-5), a twisted bilayer transition metal dichalcogenide (tBL-WSe<sub>2</sub>), and a rare-earth oxide (PrScO<sub>3</sub>). The sample thickness and electron doses are indicated in the top right corner of each panel. (e–h) Corresponding fast Fourier transform (FFT) power spectra of (a–d), demonstrating high information transfer of the reconstructed images. These results highlight PtyRAD’s capability to successfully reconstruct datasets across a wide range of doses and specimen thicknesses.

We also demonstrate sub-0.5 Å resolution reconstructions from radiation-hard samples imaged at high electron doses. Figure 2c shows a twisted bilayer of WSe<sub>2</sub> using the dataset from Nguyen et al. (2024) at a dose of  $7.6 \times 10^5$  e<sup>-</sup>/Å<sup>2</sup> and using a probe-corrected instrument. Using PtyRAD for the multislice electron ptychographic reconstruction, we observe an information limit of 30 pm, superior to the 41 pm limit reported by the original work using the single slice model. Finally, Fig 2d shows a reconstruction of the PrScO<sub>3</sub> dataset reported by Chen et al. (2021) at a dose of approximately 10<sup>6</sup> e<sup>-</sup>/Å<sup>2</sup>. Using PtyRAD, we achieve good information transfer down to 23 pm, similar to that reported in the original work. That work attributed the observed width of the atomic columns primarily to random thermal displacement of the atoms at room temperature, rather than limitation of the imaging method, and estimated an instrument blur of approximately 16 pm.

These four benchmarking reconstructions, which cover samples from bilayer to 32 nm in thickness and span four orders of magnitude of dose, demonstrate the applicability of PtyRAD and AD-based ptychography to a wide range of relevant experimental parameters. We successfully reconstruct data ranging from sparse, low dose patterns dominated by shot noise, up to data with very high signal to noise, strong dynamical scattering effects, and substantial darkfield information. While thorough comparisons of reconstructions on experimental data are difficult, we show high-quality phase images with equivalent or better information limit in each example.

#### Feature comparison with other open-source packages

To clarify the unique capabilities of PtyRAD, we compare PtyRAD with two widely used open-source packages, PtychoShelves/fold\_slice (Wakonig et al. (2020); Jiang (2020)) and py4DSTEM (Savitzky et al. (2021); Varnavides et al. (2023)) in Table 1, which we select based on their current wide adoption in electron ptychography research. PtychoShelves originated from the X-ray ptychography community and has been extended by multiple groups to implement additional features relevant for electron ptychography, including the fold\_slice fork (Jiang (2020)) and its descendants (LeBeau (2022)). It provides a powerful toolbox for ptychographic reconstruction, particularly for X-ray applications, supporting advanced features such as fly-scan, ptycho-tomography, and various ptychographic algorithms including difference map and ePIE. Py4DSTEM is a full-featured 4D-STEM analysis suite providing comprehensive tools from pre-processing to numerous virtual imaging techniques and supporting a wide range of structural characterization methods (Savitzky et al. (2021); Ophus et al. (2022); Donohue et al. (2021)). The phase contrast imaging tools in py4DSTEM include DPC, direct ptychography, mixed-state multislice ptychography using gradient descent and other projection-based phase retrieval algorithms, as well as ptycho-tomography and novel algorithms for recovering the magnetic vector potential. In the following sections, we compare the performance of these packages against PtyRAD using their gradient-descent mixed-state multislice implementations. A more comprehensive list of available open-source ptychography packages, covering a wider range of algorithms, is provided in SI

**Table 1.** Feature comparison of different iterative ptychography packages.

Package	Multi slice	Mixed Probe	Mixed Object	Position Correction	Tilt Correction	Ptycho Tomo	Hyperparam Tuning	Language	GPU
PtyRAD (ours)	✓	✓	✓	✓	✓ <sup>c</sup>		✓	Python	Multi
PtychoShelves <sup>a</sup>	✓	✓		✓	Fixed value <sup>b</sup>	✓	✓	MATLAB	Single
py4DSTEM	✓	✓		✓	Fixed value	✓	✓	Python	Single

<sup>a</sup> Refers to `fold_slice`, the modified electron ptychography version maintained by Dr. Yi Jiang ([Jiang \(2020\)](#)).

<sup>b</sup> This feature is implemented by another GitHub fork (variant) of `fold_slice` maintained by LeBeau group ([LeBeau \(2022\)](#)).

<sup>c</sup> PtyRAD can correct either global or position-dependent tilts during the ptychographic reconstruction.

Table S1. It is important to note that the information in Table 1 reflects the current state of these packages, but as open-source projects under current development, they are continuously evolving, and new features may be integrated over time.

All three packages share essential functionalities, including multislice algorithms ([Tsai et al. \(2016\)](#)), mixed probe modes ([Thibault and Menzel \(2013\)](#)), and probe position correction ([Maiden et al. \(2012a\)](#); [Thibault and Guizar-Sicairos \(2012\)](#)). PtyRAD expands on these capabilities by introducing support for mixed states in the object function. Reconstruction of mixed object states have been demonstrated in optical ptychography ([Thibault and Menzel \(2013\)](#); [Li et al. \(2016\)](#)), and recent studies have investigated its potential for reconstructing phonon modes ([Gladyshev et al. \(2023\)](#)) and improving reconstruction quality ([Schloz et al. \(2024\)](#)) in electron ptychography. We note that while PtyRAD includes full support for mixed object states, all the reconstructions shown in this manuscript use a single object state. A thorough exploration of the advantages and implications of mixed objects will be presented in a follow-up study.

Taking advantage of automatic differentiation (AD), PtyRAD also introduces position-dependent specimen tilt correction, with the option to switch between local and global tilt corrections. This feature addresses practical challenges in experiments, as unavoidable mistilts or bending in the specimen can impact reconstruction accuracy. Local tilt correction is particularly valuable for large field-of-view datasets ([KP et al. \(2025\)](#)), where a global tilt correction may be insufficient. Although [Sha et al. \(2022\)](#) have reported iterative tilt and thickness correction, their implementation is not freely available. PtychoShelves and py4DSTEM support fixed global tilt corrections which are not updated during iteration, requiring the use of the hyperparameter tuner for optimization.

Accessibility is another key consideration. PtychoShelves is primarily implemented in MATLAB, which introduces licensing costs that can be a barrier to broader adoption, especially outside academia. PtyRAD and py4DSTEM are implemented in Python using only open-source dependencies, lowering the entry barrier for new users and fostering collaboration. While all three packages support GPU acceleration, PtyRAD uniquely supports multi-GPU acceleration ([Gugger et al. \(2022\)](#)), enabling reconstruction of large datasets that may not fit within a single GPU’s memory and faster reconstruction times when using large batch sizes (SI Figure S2).

### Computation speed and convergence

To assess the computational performance of PtyRAD, we performed reconstructions with each package on identical datasets and using the same GPU hardware (a 20 GB MIG slice from an 80 GB NVIDIA A100). In this comparison, we choose Adam as the

optimizer for PtyRAD, LSQML for PtychoShelves, and the “gradient descent (GD)” option for py4DSTEM. For each package, we recorded the iteration time excluding the package initialization and result-saving time to focus on the main computational work, and one iteration is defined as a full pass of all diffraction patterns in the dataset. We average all recorded iteration times for each package, ensuring that the warm-up effect from early iterations is negligible. Input files for each package to reproduce these reconstructions are provided in our Zenodo record, and we have made our best efforts to optimize the reconstruction settings for each package. See Supplementary Information for more benchmarking details.

To measure the convergence versus computation time of each package, we reconstruct a simulated dataset of twisted bilayer WSe<sub>2</sub> and compare the structural similarity index measure (SSIM) ([Wang et al. \(2004\)](#)) of the reconstruction with the ground truth phase image at each iteration. This metric is preferable over the data error or loss function as it measures the perceptual similarity in terms of structure, brightness, and contrast of the reconstruction image. In addition, this is independent of the objective function used in the optimization and is not able to be arbitrarily lowered by overfitting. The simulated dataset was generated using abTEM ([Madsen and Susi \(2021\)](#)) with parameters comparable to the experimental tBL-WSe<sub>2</sub> dataset, including phonon, partial coherence, and Poisson noise at a dose of  $1 \times 10^6$  e<sup>-</sup>/Å<sup>2</sup> (SI Table S4). The ground truth 3D object phase is generated by first applying the strong phase approximation  $t(\mathbf{x}) = \exp(i\sigma v_z(\mathbf{x})$  ([Kirkland \(2010\)](#)) to the abTEM-simulated 3D potential, and then taking the phase angle of the complex transmission function. The SSIM is calculated from the depth-summed 2D ground truth and reconstructed phase images. Note that we subtract the minimum value from each reconstructed phase image to remove any arbitrary global phase offset in the reconstruction, but phase differences are compared on an absolute scale, so failures to quantitatively reproduce phase contrast will be penalized. The simulated dataset is available in our Zenodo record.

Figure 3a compares the progression of the SSIM with the accumulated reconstruction time over 200 iterations. The corresponding final phase images and their FFTs are shown in SI Figure S3, while intermediate reconstructions and the ground truth potential are provided in SI Figure S4. Note that the accumulated reconstruction time is not the “wall time”; it is defined as the previously mentioned average iteration time multiplied by the iteration number. PtyRAD completes one iteration of reconstruction in 26 sec, a 6–12× speedup over the other packages (166 and 310 sec per iteration). This improvement stems primarily from the use of PyTorch—a highly efficient GPU computing

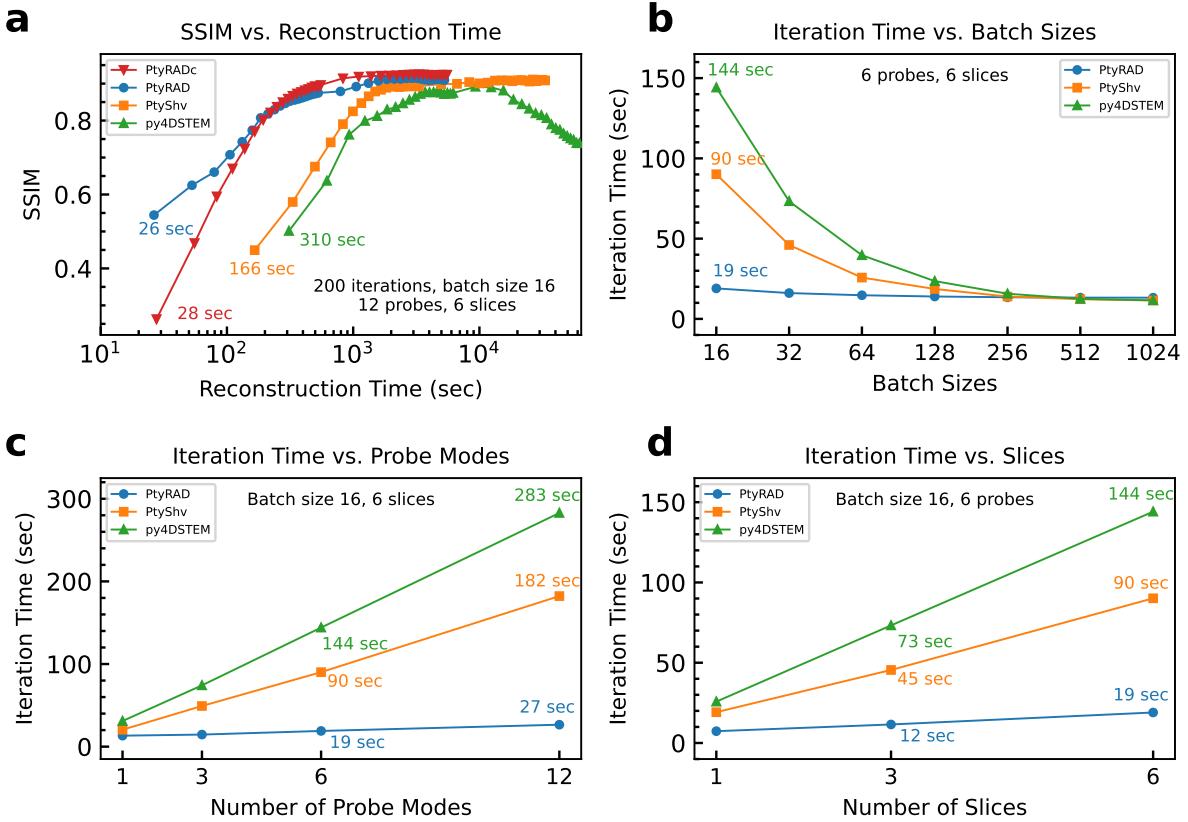


Fig. 3: Computation efficiency benchmark of PtyRAD against PtyShelves (labeled as “PtyShv”) and py4DSTEM using identical GPU-accelerated hardware and algorithm settings. (a) Reconstruction quality assessed by the structural similarity index measure (SSIM) comparing the reconstructed object phase with the ground truth using a simulated dataset. Higher SSIM corresponds to a better match with the ground truth. PtyRAD run with additional positivity constraint and sparsity is labeled as “PtyRADc”. Scaling behavior of iteration times under different (b) batch sizes, (c) number of probe modes, and (d) number of object slices. The iteration time roughly scales inversely proportional with batch sizes for PtyShelves and py4DSTEM, especially for small batch sizes, while PtyRAD only shows mild increase in the iteration time. Additionally, although the iteration time typically scales linearly with model complexities such as the number of probe modes and object slices, PtyRAD shows a much gentler slope due to PyTorch’s optimized tensor operations and hardware acceleration, resulting the excellent computational efficiency of PtyRAD.

framework—and a vectorized implementation of probe mode computations, which will be discussed later in more details. We present two reconstructions using PtyRAD: the blue curve (“PtyRAD”) uses comparable settings to py4DSTEM and PtyShelves, while the red curve (“PtyRADc”) includes our additional positivity and sparsity regularizations that will be further discussed in Figure 4 and 5. Notably, these constraints do not significantly affect iteration time, and both PtyRAD runs achieve comparable or higher SSIM than the other packages while requiring much less time. It is important to note that due to the large space of possible algorithmic settings and the inherent variability between datasets, the convergence behavior of each package will vary in practice, and so these results may not apply universally across all applications.

Figures 3b–d illustrate how iteration time varies with key reconstruction parameters: the batch size (the number of diffraction patterns grouped together for processing per gradient update), the

number of mixed-state probe modes, and the number of object slices, respectively<sup>5</sup>. The full table of iteration times is provided as SI Figure S5.

Figure 3b shows the iteration time as a function of the batch size, also known as grouping size or mini-batch size. The batch size does not change the total amount of computation required per iteration, but it sets an upper bound on the achievable level of parallelism, as patterns within a batch can be computed in parallel while individual batches must be processed sequentially. Smaller batch sizes generally lead to longer iteration times due to lower GPU utilization, while increasing the batch size reduces the iteration time until the parallel computing pipeline of the GPU is fully utilized and no further improvement is observed. For very large batch sizes, with parallel computation of each pattern in the batch,

<sup>5</sup> The timings here are measured using the experimental tBL-WSe<sub>2</sub> dataset as in Figure 2c. This choice does not affect the comparison, as the timings are primarily dependent on the dataset size, rather than its content.

the GPU memory will eventually become fully occupied—in this case, such large batch sizes cannot be processed by py4DSTEM or PtychoShelves. PtyRAD supports processing larger batches by either spreading the workload across multiple GPUs (SI Figure S2), or by splitting the batch into sub-batches, which are computed serially and accumulate their gradients before applying the update step<sup>6</sup>. In practice, very large batch sizes are not commonly used with gradient descent ptychography as reconstruction quality tends to decrease significantly with increasing batch sizes (SI Figure S6). Smaller batch sizes have been observed to yield better reconstruction quality, likely because they update the reconstruction much more frequently and introduce noisier gradients, which can help the optimizer to escape shallow local minima and improve convergence in non-convex problems. Conversely, reconstructions with larger batch sizes tend to stagnate or over-smooth fine features. However, we anticipate that optimizations for large batches will be important as high-speed detectors with greater pixel count become available, as the memory required per batch increases sharply with the detector pixel count.

Next, we investigate the performance impact of the model complexity, specifically the number of probe modes and object slices, for all three packages. Figure 3c and d show that the iteration time scales linearly with increasing model complexity. The linearity is expected because the iteration time is primarily dominated by the forward and backward passes, and the needed computation increases proportionally with the number of probe modes and object slices as outlined in [Physical model](#). Note that the curves in Figure 3c and d do not pass through the origin due to unavoidable GPU overheads<sup>7</sup>.

It is also worth mentioning that both PtychoShelves and py4DSTEM process probe modes sequentially, whereas PtyRAD employs a fully vectorized approach that performs the forward and update steps for all probe modes in parallel. Since each probe mode contributes independently to the total diffraction intensity, PtyRAD vectorizes the computation by broadcasting the object tensor across probe modes as illustrated in Figure 1. Parallelization of the probe mode calculations allows PtyRAD to more effectively fill the parallel computing pipeline of the GPU, leading to higher performance. In contrast, computation of each slice is necessarily serial and can not be vectorized for all three packages.

Additionally, the underlying GPU frameworks differ among these packages—PtyRAD leverages PyTorch, py4DSTEM utilizes CuPy, while PtychoShelves uses the MATLAB Parallel Computing toolbox. These GPU frameworks play a critical role in performance as behind-the-scenes scheduling and dispatching of computational tasks to the GPU cores can significantly impact the GPU utilization. Since each GPU framework employs different strategies for resource scheduling, kernel launching, and data caching, PtyRAD achieves the best performance at small batch sizes, while the advantage diminishes with increasing batch sizes as shown in Figure 3b and SI Figure S5. We also report iteration times and GPU metrics for each package in SI Table S5, and observe that PtyRAD consistently achieves higher GPU and memory utilization rates, highlighting the impact of the underlying GPU framework on compute efficiency.

<sup>6</sup> A common technique in the machine learning community called “gradient accumulation”.

<sup>7</sup> Common GPU overheads include data transfer, device synchronization, kernel launch, and other operations that incur time costs without performing actual computation.

While PtyRAD shows faster iteration times across the range of parameters shown here, we note that the observed convergence speed in Figure 3a is determined by both the computation time and the improvement per iteration. Since both factors are highly sensitive to the chosen reconstruction parameters and properties of the dataset, the speed advantage of PtyRAD may not always translate into overall higher reconstruction quality when compared to other major packages.

In our testing we have also found that the performance varies greatly with different GPU hardware, even when keeping the same reconstruction parameters. To understand which hardware specification has the most influence on reconstruction speed, we also benchmarked PtyRAD on different GPUs and find that the performance is roughly proportional to the GPU memory bandwidth under typical settings (SI Figure S7), and relatively less sensitive to the peak computing power measured in floating point operations per second (FLOPS). This indicates that PtyRAD is primarily memory bandwidth-limited, meaning its throughput is ultimately constrained by how quickly data can be moved between GPU memory and compute cores. This limited bandwidth helps explain the linear scaling of PtyRAD observed in Figure 3c despite the probe modes are calculated in parallel. It also accounts for the saturation behavior seen in Figure 3b, where iteration times plateau at larger batch sizes across all three packages.

Figure 4 compares the final reconstructed phase images, with their corresponding FFT power spectra, produced with each package from the experimental tBL-WSe<sub>2</sub> dataset. For all packages we used the same condition as in Figure 3a—12 incoherent probe modes, 6 object slices, a batch size of 16 patterns, and ran for 200 iterations. Note that a batch size of 16 yields the best reconstruction quality across packages under comparable conditions for this dataset (SI Figure S6). The intermediate and final phase images are shown in SI Figure S8, while the reconstructed probe modes are presented in SI Figure S9. The timings indicated on each panel for the total reconstruction time were measured using a full NVIDIA A100 80GB GPU. Despite using the same number of iterations, PtyRAD achieves better contrast and higher information transfer while completing the reconstruction in 17–24× less time than the other packages given the test condition.

## Object constraints and regularizations

Ptychography reconstructions are large-scale optimization problems involving millions of free parameters. While ptychography experiments are designed to include multiple redundant measurements for each unknown parameter (i.e. to have a sufficient overlap ratio as discussed by [Bunk et al. \(2008\)](#); [Edo et al. \(2013\)](#); [da Silva and Menzel \(2015\)](#)) and thus to overdetermine the solution ([Schlöz et al. \(2020\)](#); [Gilgenbach et al. \(2023\)](#)), the reconstruction is nevertheless ill-conditioned due to inherent ambiguities ([Rodenburg and Maiden \(2019\)](#); [Li et al. \(2016\)](#)), imprecisely known experiment parameters, and noisy data. Therefore, physical constraints and regularizations are indispensable for ptychographic reconstructions to ensure a physically meaningful solution. Generally, regularization refers to penalty terms added to the loss function (e.g., sparsity regularization in Equation (8)), whereas constraints correspond to operations that directly modify the solution (e.g., positivity constraint). While we follow this distinction, we use the term “depth regularization” to be consistent with existing literature, even though it acts more like a soft constraint.

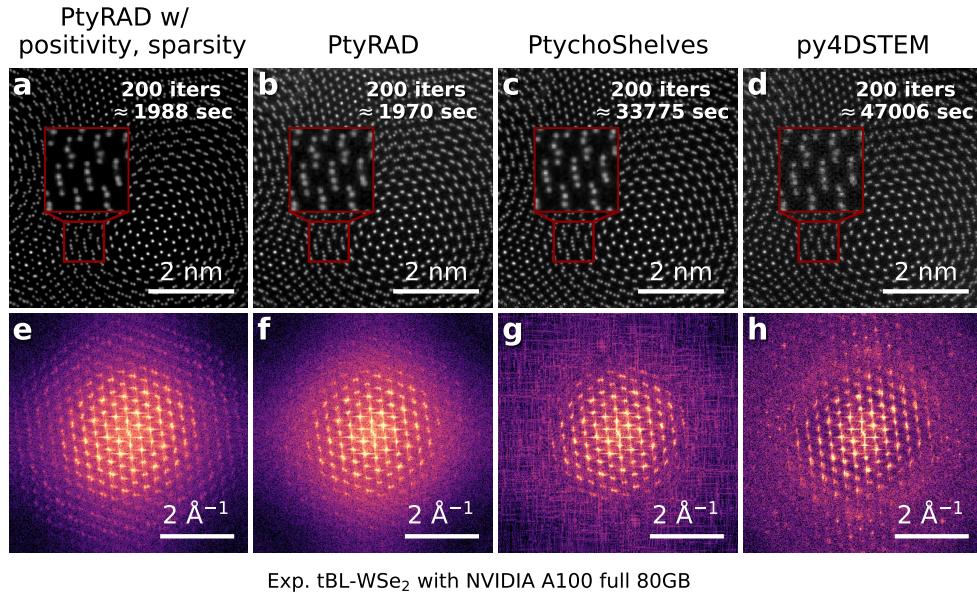


Fig. 4: Reconstructed phase images of the experimental twisted bilayer WSe<sub>2</sub> dataset after 200 iterations using (a) PtyRAD with positivity and sparsity regularization, (b) PtyRAD, (c) PtychoShelves, and (d) py4DSTEM and zoom-in insets. (e–h) Corresponding fast Fourier transform (FFT) power spectra of (a–d). Total reconstruction times taken for 12 probe modes, 6 object slices, batch size 16, and 200 iterations are labeled in the top right corners of each panel. PtyRAD achieves higher information transfer given the same number of iterations and a shorter reconstruction time.

Figure 5 demonstrates the impact of various constraint and regularization techniques on the reconstruction of the experimental tBL-WSe<sub>2</sub> dataset. The reconstructions are all performed using 12 probe modes, 12 1-Å object slices, and a batch size of 16, for 200 iterations. The model contains 9,473,888 optimizable parameters, including 9,047,904 for the object, 393,216 for the probe, and 32,768 for the probe positions. Meanwhile, the 4D-STEM dataset contains  $128^4 = 268,435,456$  measured values, yielding an overdetermination ratio of 28.33. Despite the seemingly sufficient overdetermination of the problem, with no regularization applied to the reconstruction, the phase image in Figure 5a shows poor contrast and no clear separation between the two layers of the heterostructure along the depth direction.

#### Fourier-space depth regularization

A fundamental challenge in MEP is the limited transfer of information along the beam propagation direction (Terzoudis-Lumsden et al. (2023)) combined with the need for thin object slices in order to correctly model multiple scattering of electrons. As a result, it is very difficult to reconstruct three-dimensional information without imposing regularizations on the object along the beam direction. A common approach is the use of a low pass filter function in Fourier space, which downweights spatial frequencies along the  $k_z$  direction for which there is minimal information transfer. PtychoShelves, py4DSTEM, and PtyRAD all implement such  $k_z$ , or “missing wedge” filter (Wakonig et al. (2020); Chen et al. (2021)) with slight differences. The missing wedge filter  $W(\mathbf{k})$  is given by

$$W(\mathbf{k}) = 1 - \frac{2}{\pi} \tan^{-1} \left( \frac{\beta^2 |k_z|^2}{k_x^2 + k_y^2 + \epsilon} \right) \quad (10)$$

where  $\beta$  is the strength parameter, typically in the range 0.1–1, and  $\epsilon = 10^{-3}$  is a small constant added for numerical stability. In PtychoShelves and PtyRAD,  $W$  is further modified into  $W_a$  by applying a lateral Gaussian blur with strength parameter  $\alpha$ , while py4DSTEM directly uses the original  $W$  as the final filter function.

$$W_a(\mathbf{k}) = W \exp \left( -\alpha(k_x^2 + k_y^2) \right) \quad (11)$$

The object function is then Fourier-filtered with the given filter function in reciprocal space, producing the modified object function  $O'$ .

$$O'(\mathbf{r}) = \mathcal{F}_{3D}^{-1} [W_a(\mathbf{k}) \cdot \mathcal{F}_{3D} [O(\mathbf{r})]] \quad (12)$$

Figure 5b shows that the contrast in the depth-summed phase image improves significantly by applying such  $k_z$  filter with strength parameter  $\beta = 1$ , but the separation between the top and bottom slices remains poor.

#### Positivity and sparsity

Arbitrary offsets of the object phase are also ambiguous, so we additionally apply an optional positivity constraint to ensure the object phase is strictly nonnegative in Figure 5c. By clipping negative phase values after each iteration, the reconstructed image shows a higher contrast level and lower background intensity. We have also tested enforcing the positivity constraint by subtracting the minimum value at each iteration and found that clipping negative values provides much better reconstruction quality.

Since we expect phase images to contain bright, discrete atoms for atomic-resolution applications, for such applications we can also incorporate a sparsity-promoting regularization term (Equation (8)) as shown in Figure 5d. This additional sparsity term

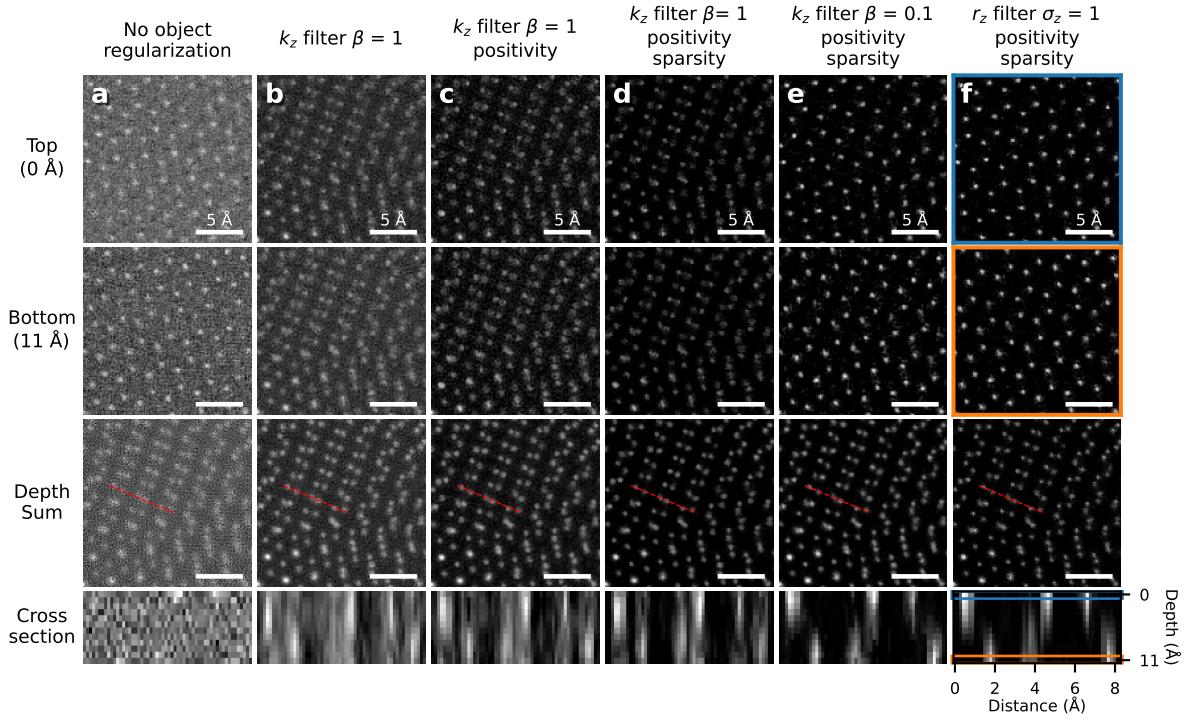


Fig. 5: Impact of object constraints and regularizations on multislice electron ptychography (MEP) tested with the experimental tBL-WSe<sub>2</sub> dataset. The two layers are stacked in an anti-parallel configuration with an interlayer twist angle is 3 °. We tested different constraint and regularization settings, including (a) no object regularization, (b) multislice  $k_z$  filter  $\beta = 1$ , (c)  $k_z$  filter  $\beta = 1$  with object positivity constraint, (d)  $k_z$  filter  $\beta = 1$  with positivity and sparsity regularization, (e)  $k_z$  filter  $\beta = 0.1$  with positivity and sparsity, and (f) multislice  $r_z$  filter  $\sigma_z = 1$  with positivity and sparsity. The multislice  $k_z$  and  $r_z$  filters are necessary for stable MEP reconstructions, while the object positivity constraint and sparsity regularization improves the overall contrast and suppress noise. Each column displays the reconstructed top slice (0 Å), bottom slice (11 Å), the depth sum projection of all 12 slices, and a cross-sectional view along the red dashed lines labeled in the depth sum images. The blue and orange boxes highlight that the top and bottom slices reconstructed using the  $r_z$  filter are well decoupled, showing reduced wrap-around artifact with a cleaner reconstruction quality compared to other regularization approaches. By properly combining the multislice  $r_z$  filter, positivity, and sparsity regularizations, PtyRAD achieves much improved MEP reconstruction quality.

promotes “atomicity” (Sayre (1952); Van den Broek and Koch (2012, 2013)) by adding a weighted L1 penalty term of the object phase into the loss function during reconstruction (Equation (9)). The default weighting parameter of the sparsity term is set at 0.1. By simply suppressing small phase values, sparsity regularization improves visual quality with reduced background noises, enhanced contrast and layer separation, and extended information transfer (SI Figure S10).

The sparsity promoting regularization appears to assist in deconvolution of the probe from the object by penalizing the “ringing” artifact around the atoms that occurs when the probe and object are not separated correctly. This also differs from simple intensity clipping, which can be used to artificially sharpen atomic resolution STEM images (Yu et al. (2003)); in that case, intensity between atoms caused by the tails of the probe are artificially discarded. In ptychography, the probe is deconvolved from the image, and the addition of this regularization term simply assists in correctly determining the probe function. Therefore, positivity and sparsity regularization may enhance atom localization accuracy by suppressing background noise and reducing irregular phase

variations around atomic columns, which often arise from probe-object intermixing. However, care must be taken as the selective penalization of small values can potentially distort the relative phase contrast between different atomic species. This is particularly concerning when imaging heterogeneous materials, as weaker atomic columns or isolated defects might be disproportionately affected or potentially eliminated by aggressive regularization. Therefore, the regularization strength must be carefully selected and checked to prevent excessive suppression of weak signals and potential artifacts.

#### Regularization strength and artifacts

The strength parameter  $\beta$  of the  $k_z$  filter presents a similar trade-off between the reconstruction stability and depth resolution. This  $k_z$  filter is designed to downweight noise in the “missing wedge” of k-space, which stabilizes the multislice reconstruction. However, increasing  $\beta$  also suppresses more  $k_z$  information within the region, yielding a reconstruction with less details along the depth dimension. As shown in Figure 5e, reducing the  $k_z$  filter strength from  $\beta = 1$  to 0.1 improves the separation of the top and bottom layers, as seen in both individual slices and cross-sectional views.

This improved layer separation primarily stems from preserving more  $k_z$  information with a weaker filter, enabling recovery of finer structural features in depth. At the same time, the weaker filtering also reduces the level of wrap-around artifacts, which can be particularly problematic in specimens lacking depth periodicity—such as twisted 2D materials or those with uncorrected crystal tilt, where the top and bottom slices do not perfectly align. The wrap-around artifact “transfers” surfaces intensities to the other end of the volume via the periodic boundary condition inherent in Fourier space, effectively imprinting features from one surface onto the other. Although this effect is largely mitigated by reducing  $\beta$ , some residual wrap-around artifacts remain visible in the top and bottom slices of Figure 5e. In addition, we have observed that a  $k_z$  filter with excessive regularization strength can introduce incorrect local intensity variations (SI Figure S11), highlighting the need for careful parameter selection and mindful interpretation.

Beyond reducing the strength parameter  $\beta$ , another common approach to mitigating the wrap-around artifact of the  $k_z$  filter is to set the extent of the object array thicker than the actual specimen. Conceptually, this introduces vacuum layers above and below the specimen. However, this method presents several challenges in practice. First, increasing the total number of slices adds computational overhead, further prolonging the already time-consuming MEP reconstructions. Second, it introduces ambiguity in the specimen’s vertical position within the object array, as this position can be freely adjusted by altering the probe defocus during reconstruction. Third, there is no guarantee that the reconstructed object will form well-defined vacuum layers. In our observations, reconstructions with additional slices often produce specimens that either concentrate at one of the surfaces or become evenly “stretched” vertically to fill the entire object array, eliminating any vacuum regions. While py4DSTEM offers an option to pad the object array before applying the  $k_z$  filter and then crop it afterward, we find that this approach significantly reduces the reconstructed phase values at the surfaces for the tBL-WSe<sub>2</sub> dataset.

### Real-space depth regularization

In contrast to the Fourier-space  $k_z$  filter, PtyRAD uniquely implements a real-space depth regularization,  $r_z$  filter, by applying a 1D Gaussian blur along the object depth dimension.

$$O'(\mathbf{r}) = O(\mathbf{r}) * G_{\sigma_z}(r_z) \quad (13)$$

Here,  $G_{\sigma_z}$  is the 1D Gaussian kernel with a standard deviation  $\sigma_z$ , and the modified object  $O'$  is obtained by convolving the original object  $O(\mathbf{r})$  with the Gaussian kernel  $G_{\sigma_z}(r_z)$  in real space.

Unlike the  $k_z$  filter (Figure 5d–e), which operates in Fourier space and introduces the wrap-around artifact, the  $r_z$  filter (Figure 5f) stabilizes the multislice reconstruction without intermixing the top and bottom slices. Given that the interlayer distance of bulk WSe<sub>2</sub> is approximately 6–7 Å, our results (Figure 5e–f) indicate a comparable depth discrimination as shown in their cross-sectional views. To better estimate the depth resolution, we reconstructed with additional vacuum slices for a more accurate evaluation. We measured the full width at half maximum (FWHM) of 34 single W atoms along the depth direction across different sample regions (SI Figure S12). This analysis yields a mean depth resolution of 7.5 Å, with the best value reaching 6.6 Å. These

results highlight the importance of appropriate regularization strategies and careful parameter selection in achieving high-quality 3D reconstructions.

### Hyperparameter tuning

Beyond the millions of model parameters optimized through gradient descent, ptychographic reconstructions also depend on dozens of algorithmic and experimental parameters that critically influence reconstruction quality but are difficult to optimize directly via gradient-based methods. These parameters are fixed during the reconstruction and are often referred to as hyperparameters (Bergstra et al. (2011)). To identify optimal choices for these hyperparameters, all three packages implement a Bayesian optimizer that runs separate trial reconstructions with different input values. Both PtychoShelves (specifically the `fold_slice` forks) and py4DSTEM utilize Bayesian optimization with Gaussian process (BO-GP) for their hyperparameter tuning. PtyRAD uses Optuna (Akiba et al. (2019)), a widely-used open-source framework for distributed hyperparameter search, which supports a variety of probability models and optimization strategies. Note that the required accuracy of initial experimental parameters is strongly dependent on the dataset quality, especially the signal-to-noise ratio of the diffraction patterns. For example, an initial guess within  $\pm 5$  nm for the probe defocus is usually acceptable for dose-sufficient datasets like the tBL-WSe<sub>2</sub> and PSO. In contrast, low dose datasets like MOSS-6 and ZSM-5 are much harder to recover the probe, so the initial defocus value becomes critical and requires hyperparameter tuning to optimize the probe defocus.

Figure 6 demonstrates the use of Bayesian optimization (BO) for tuning the algorithmic and experimental parameters for a reconstruction of the ZSM-5 dataset (Zhang et al. (2023)) using a full NVIDIA A100 80GB GPU. Figure 6a compares different hyperparameter search strategies based on their performance over time. The optimization targets two critical parameters affecting reconstruction quality: total sample thickness and crystal misorientation (tilts). While PtyRAD supports AD-based optimization of these parameters, they are typically optimized as hyperparameters in other packages, making them a natural benchmark for evaluating search strategies. We compare random search with the Tree-structured Parzen Estimator (TPE) (Bergstra et al. (2011)), a nonparametric BO algorithm that adaptively refines the sampling distribution by distinguishing between well-performing and poorly performing configurations. Notably, TPE has been reported to outperform GP (Bergstra et al. (2011)), and has been widely adopted in modern hyperparameter optimization frameworks developed by the machine learning community (Watanabe (2023)). TPE begins with random sampling for the first 10 trials to establish an initial estimate of the model response. Following these initial trials, the TPE method chooses superior hyperparameters which produce reconstructions with improved optimization objective.

To improve the convergence speed of the BO optimization, Optuna supports pruning, which monitors the objective function improvement of each trial reconstruction during iteration and terminates underperforming trials early. Figure 6a shows the convergence of both random and TPE search with and without use of the Hyperband (Li et al. (2018)) pruner. Each convergence curve is averaged over 20 independent runs, while each run is limited to a 4-hour time budget. Each trial has a maximum of

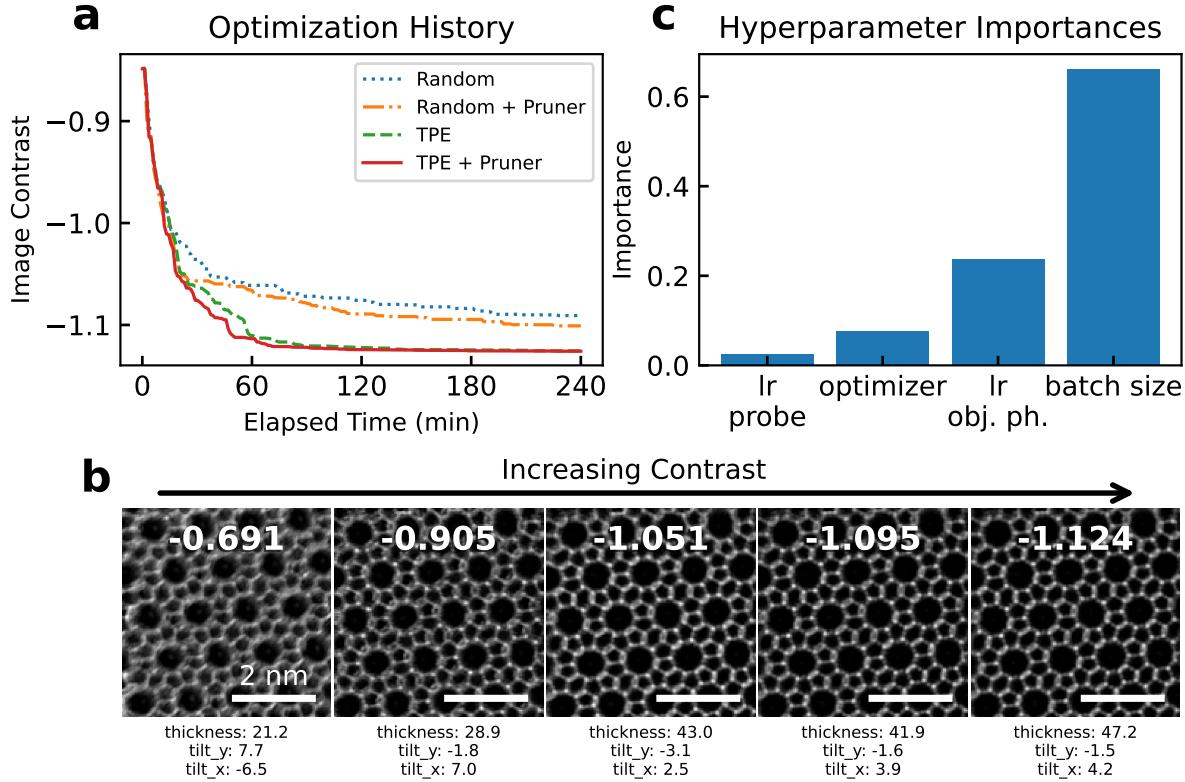


Fig. 6: Hyperparameter tuning using Bayesian optimization (BO) with pruning for ptychographic reconstruction on the experimental ZSM-5 dataset. (a) Convergence of image contrast optimization using different search methods, including random sampling and BO with Tree-structured Parzen Estimator (TPE). Each method is evaluated with and without the Hyperband pruner. TPE combined with pruning demonstrates the fastest convergence and best performance. (b) Representative reconstructed phase images with image contrast values labeled on top, demonstrating the impact of hyperparameter tuning on image quality. The BO-suggested specimen thickness ( $\text{\AA}$ ) and tilts (mrad) are labeled at the bottom. Note that a negative sign is added to the image contrast for the minimization direction. (c) Hyperparameter importances calculated using PED-ANOVA. Batch size is identified as the most influential hyperparameter, followed by the learning rate for the object phase (labeled as “lr obj. ph.”), while the choice of optimizer and learning rate for the probe (denoted as “lr probe”) show relatively low impact on the optimization objective.

20 iterations if not pruned. To ensure fair comparisons, all strategies share the same 10 initial trials in each run, with random seeds assigned based on the run index. This guarantees that every strategy starts from an identical initialization before exploring further. The TPE model with pruning (red curve) shows the fastest improvement of the image contrast and results in the most optimal solution in the allotted time. On average, TPE with pruning explores 156 trials, of which 76 of them are pruned during the process. Without pruning, only 125 trials were evaluated in the time limit, indicating the efficiency of pruning algorithm. Although Figure 6a suggests that on average, the image contrast converges within 120 minutes using the TPE algorithm with pruning, the actual time required naturally depends greatly on the initial accuracy of the experimental parameters and the signal-to-noise ratio of the data.

One critical challenge in hyperparameter tuning for ptychographic reconstruction is selecting an appropriate quality metric. Conventional data error metrics computed from diffraction patterns do not always correlate with perceptual image quality (SI Figure S13) as overfitting can lead to solutions with lower error but nonphysical artifacts. SSIM provides an objective measure

of reconstruction quality but requires a ground truth reference for comparison, making it irrelevant for experimental data. To address this, we use the image contrast, defined as the standard deviation ( $\sigma$ ) divided by the mean intensity ( $\mu$ ), as our objective metric for hyperparameter tuning. This metric provides a simple, normalized measure of image variation that is linearly related to standard deviation, making it applicable for reconstructed phase images regardless of their absolute scales. It has long been used for auto-focusing in electron microscopy (Kirkland (1990, 2018)) and is commonly referred to as the normalized variance (Pattison et al. (2024)), coefficient of variation (CV), normalized root-mean-square deviation (NRMSD), and relative standard deviation (RSD) (Everitt and Skrondal (2010)).

$$\text{Optimization Objective} = -\frac{\sigma(O_p)}{\mu(O_p)} \quad (14)$$

Here,  $O_p$  denotes the reconstructed object phase. For optimization purposes, we apply a negative sign to the contrast metric, ensuring a consistent minimization objective. Figure 6b shows the representative phase images reconstructed during the BO process,

demonstrating a good correlation between image quality with the image contrast value.

To better understand the influence of other general hyperparameter on the optimization objective, Figure 6c calculates the importance of different hyperparameters, including batch size, learning rates, and choice of optimizers using PED-ANOVA (Watanabe et al. (2023)), a statistical analysis that quantifies the relative importance of each hyperparameter by measuring its contribution to the total variance in optimization outcomes. The parameter search space for importance analysis is listed in SI Table S6. The analysis reveals that batch size has the greatest impact, which is consistent with our previous observation (SI Figure S6). The learning rate for the object phase is also critical, as it sets the update step size during optimization and directly impacts the resulting image contrast. In contrast, the choice of optimizer and learning rate for the probe have relatively low influence compared to batch size and object learning rate. Although the importance values might vary with datasets, optimization setting, search spaces, and reconstruction configurations, this analysis provides valuable guideline for the overall workflow and is readily provided by Optuna and PtyRAD.

## Conclusion

In this paper, we presented PtyRAD, an open-source software framework for iterative ptychographic reconstructions. By leveraging automatic differentiation and PyTorch’s optimized tensor operations, PtyRAD achieves up to a 24 $\times$  speedup over existing packages. Our benchmarking results demonstrate that this performance improvement does not come at the expense of reconstruction quality, making PtyRAD a practical solution for high resolution and high throughput applications. Furthermore, we introduce a real-space depth regularization technique to mitigate wrap-around artifacts, which can be particularly useful for twisted 2D materials and vertical heterostructures. By combining different regularization techniques, we demonstrate a mean depth resolution of 7.5 Å on an experimental tBL-WSe<sub>2</sub> dataset with the best value approaching 6.6 Å. In addition, PtyRAD’s integrated Bayesian optimization workflow streamlines hyperparameter selection, improving reconstruction robustness across diverse experimental conditions.

Looking forward, we aim to expand PtyRAD’s capabilities to support other 3D imaging modalities, including ptychotomography (Ding et al. (2022); Romanov et al. (2024)) and tilt-coupled multislice electron ptychography (TCMEP) (Dong et al. (2025)). Further computational performance can potentially be improved by alternative GPU frameworks with just-in-time (JIT) compilation capability, such as Jax (Bradbury et al. (2018)). Most critically, we seek to establish more quantitative and reproducible ptychographic reconstruction strategies by exploring new metrics, optimization techniques, and regularization methods. By continuing to refine and extend PtyRAD, we hope to advance the field of ptychography and facilitate high-quality, interpretable reconstructions across a wide range of applications in electron ptychography and beyond.

## Data and code availability

For review purposes, a minimal working example of the code and datasets are available at: <https://bit.ly/42HtQyD>

The PtyRAD package, along with the data, input parameter files, and code required to reproduce all figures, will be made publicly available at:

- GitHub repository: <https://github.com/chiaha03/ptyrad/>
- Zenodo record: <https://doi.org/10.5281/zenodo.1527317>

## Supporting information

To view supplementary material for this article, please visit [LINK].

## Competing interests

No competing interest is declared.

## Acknowledgments

The authors thank Dr. Yi Jiang, Dr. Xiangyu Yin, Dr. Amey Luktu, and Dr. Ming Du for the support of X-ray compatibilities and valuable discussions. We also thank the members of the Muller group, particularly Dr. Guanxing Li, Harikrishnan KP, Lopa Bhatt, Noah Schnitzer, Clara Chung, Shake Karapetyan, Naomi Pieczulewski, Schuyler Zixiao Shi, and Zhaslan Baraissov for providing experimental datasets, sending feature requests, giving useful feedback during the development stage of PtyRAD. This project is supported by the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship, a program of Schmidt Sciences, LLC. The authors acknowledge the use of PARADIM computing resources under cooperative agreement number DMR-2039380.

## References

- Akiba, T., Sano, S., Yanase, T., Ohata, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631.
- Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs. <http://github.com/jax-ml/jax>.
- Bunk, O., Dierolf, M., Kynde, S., Johnson, I., Marti, O., and Pfeiffer, F. (2008). Influence of the overlap parameter on the convergence of the ptychographical iterative engine. *Ultramicroscopy*, 108(5):481–487.
- Candes, E. J. and Tao, T. (2006). Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425.
- Cao, M. C., Chen, Z., Jiang, Y., and Han, Y. (2022). Automatic parameter selection for electron ptychography via bayesian optimization. *Scientific Reports*, 12(1):12284.
- Chang, D. J., O’Leary, C. M., Su, C., Jacobs, D. A., Kahn, S., Zettl, A., Ciston, J., Ercius, P., and Miao, J. (2023). Deep-learning electron diffractive imaging. *Physical review letters*, 130(1):016101.
- Chapman, H. N. and Nugent, K. A. (2010). Coherent lensless x-ray imaging. *Nature photonics*, 4(12):833–839.

- Chen, Z., Jiang, Y., Shao, Y.-T., Holtz, M. E., Odstrčil, M., Guizar-Sicairos, M., Hanke, I., Ganschow, S., Schlom, D. G., and Muller, D. A. (2021). Electron ptychography achieves atomic-resolution limits set by lattice vibrations. *Science*, 372(6544):826–831.
- Cherukara, M. J., Zhou, T., Nashed, Y., Enfedaque, P., Hexemer, A., Harder, R. J., and Holt, M. V. (2020). Ai-enabled high-resolution scanning coherent diffraction imaging. *Applied Physics Letters*, 117(4).
- Clark, J. N., Putkunz, C., Pfeifer, M. A., Peele, A., Williams, G., Chen, B., Nugent, K. A., Hall, C., Fullagar, W., Kim, S., et al. (2010). Use of a complex constraint in coherent diffractive imaging. *Optics express*, 18(3):1981–1993.
- Close, R., Chen, Z., Shibata, N., and Findlay, S. (2015). Towards quantitative, atomic-resolution reconstruction of the electrostatic potential via differential phase contrast using electrons. *Ultramicroscopy*, 159:124–137.
- da Silva, J. C. and Menzel, A. (2015). Elementary signals in ptychography. *Optics express*, 23(26):33812–33821.
- Diederichs, B., Herdegen, Z., Strauch, A., Filbir, F., and Müller-Caspary, K. (2024). Exact inversion of partially coherent dynamical electron scattering for picometric structure retrieval. *Nature Communications*, 15(1):101.
- Ding, Z., Gao, S., Fang, W., Huang, C., Zhou, L., Pei, X., Liu, X., Pan, X., Fan, C., Kirkland, A. I., et al. (2022). Three-dimensional electron ptychography of organic–inorganic hybrid nanostructures. *Nature Communications*, 13(1):4787.
- Dong, Z., Huo, M., Li, J., Li, J., Li, P., Sun, H., Gu, L., Lu, Y., Wang, M., Wang, Y., and Chen, Z. (2024). Visualization of oxygen vacancies and self-doped ligand holes in La<sub>3</sub>Ni<sub>2</sub>O<sub>7</sub>·δ. *Nature*, 630(8018):847–852.
- Dong, Z., Zhang, Y., Chiu, C.-C., Lu, S., Zhang, J., Liu, Y.-C., Liu, S., Yang, J.-C., Yu, P., Wang, Y., et al. (2025). Sub-nanometer depth resolution and single dopant visualization achieved by tilt-coupled multislice electron ptychography. *Nature Communications*, 16(1):1219.
- Donohue, J., Bustillo, K. C., Zeltmann, S. E., Ophus, C., Savitzky, B., Jones, M. A., Meyers, G. F., and Minor, A. (2021). 4D-STEM analysis of an amorphous-crystalline polymer blend: Combined nanocrystalline and RDF mapping. *Microscopy and Microanalysis*, 27(S1):1798–1800.
- Du, M., Kandel, S., Deng, J., Huang, X., Demortiere, A., Nguyen, T. T., Tucoulou, R., De Andrade, V., Jin, Q., and Jacobsen, C. (2021). Adorym: A multi-platform generic x-ray image reconstruction framework based on automatic differentiation. *Optics express*, 29(7):10000–10035.
- Du, M., Nashed, Y. S., Kandel, S., Gürsoy, D., and Jacobsen, C. (2020). Three dimensions, two microscopes, one code: Automatic differentiation for x-ray nanotomography beyond the depth of focus limit. *Science advances*, 6(13):eaay3700.
- Edo, T., Batey, D., Maiden, A., Rau, C., Wagner, U., Pešić, Z., Waigh, T., and Rodenburg, J. (2013). Sampling in x-ray ptychography. *Physical Review A—Atomic, Molecular, and Optical Physics*, 87(5):053850.
- Elser, V. (2003). Phase retrieval by iterated projections. *Journal of the Optical Society of America A*, 20(1):40–55.
- Enders, B. and Thibault, P. (2016). A computational framework for ptychographic reconstructions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2196):20160640.
- Ercius, P., Johnson, I. J., Pelz, P., Savitzky, B. H., Hughes, L., Brown, H. G., Zeltmann, S. E., Hsu, S.-L., Pedroso, C. C., Cohen, B. E., et al. (2024). The 4d camera: an 87 khz direct electron detector for scanning/transmission electron microscopy. *Microscopy and Microanalysis*, 30(5):903–912.
- Everitt, B. S. and Skrondal, A. (2010). *The Cambridge dictionary of statistics*, volume 4. Cambridge university press Cambridge, UK.
- Fienup, J. R. (1978). Reconstruction of an object from the modulus of its fourier transform. *Optics letters*, 3(1):27–29.
- Fienup, J. R. (1982). Phase retrieval algorithms: a comparison. *Applied optics*, 21(15):2758–2769.
- Friedrich, T., Yu, C., Verbeeck, J., and Van Aert, S. (2022). Phase object reconstruction for 4d-stem using deep learning,(4d-stem example data). URL <https://doi.org/10.5281/zenodo.7034879>.
- Gao, P., Kumamoto, A., Ishikawa, R., Lugg, N., Shibata, N., and Ikuhara, Y. (2018). Picometer-scale atom position analysis in annular bright-field stem imaging. *Ultramicroscopy*, 184:177–187.
- Gao, S., Wang, P., Zhang, F., Martinez, G. T., Nellist, P. D., Pan, X., and Kirkland, A. I. (2017). Electron ptychographic microscopy for three-dimensional imaging. *Nature communications*, 8(1):163.
- Gerchberg, R. and Saxton, W. (1972). A practical algorithm for the determination of phase from image and diffraction plane picture. *Optik*, 35(2):237–246.
- Gilgenbach, C., Chen, X., and LeBeau, J. M. (2023). Sampling metrics for robust reconstructions in multislice ptychography: Theory and experiment.
- Gladyshev, A., Haas, B., Boland, T. M., Rez, P., and Koch, C. T. (2023). Reconstructing lattice vibrations of crystals with electron ptychography. *arXiv preprint arXiv:2309.12017*.
- Godden, T. M., Suman, R., Humphry, M. J., Rodenburg, J. M., and Maiden, A. M. (2014). Ptychographic microscope for three-dimensional imaging. *Optics Express*, 22(10):12513–12523.
- Gugger, S., Debut, L., Wolf, T., Schmid, P., Mueller, Z., Mangrulkar, S., Sun, M., and Bossan, B. (2022). Accelerate: Training and inference at scale made simple, efficient and adaptable. <https://github.com/huggingface/accelerate>.
- Guzzi, F., Kourousias, G., Billè, F., Pugliese, R., Gianoncelli, A., and Carrato, S. (2022). A modular software framework for the design and implementation of ptychography algorithms. *PeerJ Computer Science*, 8:e1036.
- Hoppe, W. (1969). Beugung im inhomogenen Primärstrahlwellenfeld. I. Prinzip einer Phasenmessung von Elektronenbeugungsinterferenzen. *Acta Crystallographica Section A*, 25(4):495–501.
- Humphry, M., Kraus, B., Hurst, A., Maiden, A., and Rodenburg, J. (2012). Ptychographic electron microscopy using high-angle dark-field scattering for sub-nanometre resolution imaging. *Nature communications*, 3(1):730.
- Jiang, Y. (2020). fold slice. [https://github.com/yijiang1/fold\\_slice](https://github.com/yijiang1/fold_slice).
- Jiang, Y., Chen, Z., Han, Y., Deb, P., Gao, H., Xie, S., Purohit, P., Tate, M. W., Park, J., Gruner, S. M., et al. (2018). Electron ptychography of 2d materials to deep sub-ångström resolution. *Nature*, 559(7714):343–349.
- Kandel, S., Maddali, S., Allain, M., Hruszkewycz, S. O., Jacobsen, C., and Nashed, Y. S. (2019). Using automatic differentiation as a general framework for ptychographic reconstruction. *Optics express*, 27(13):18653–18672.

- Kimoto, K., Asaka, T., Yu, X., Nagai, T., Matsui, Y., and Ishizuka, K. (2010). Local crystal structure analysis with several picometer precision using scanning transmission electron microscopy. *Ultramicroscopy*, 110(7):778–782.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kirkland, E. J. (1990). An image and spectrum acquisition system for a vg hb501 stem using a color graphics workstation. *Ultramicroscopy*, 32(4):349–364.
- Kirkland, E. J. (2010). *Advanced Computing in Electron Microscopy*. Springer US, Boston, MA, 2 edition.
- Kirkland, E. J. (2018). Fine tuning an aberration corrected adf-stem. *Ultramicroscopy*, 186:62–65.
- KP, H., Harbola, V., Choi, J., Crust, K. J., Shao, Y.-T., Lee, C.-H., Yoon, D., Lee, Y., Fuchs, G. D., Dreyer, C. E., Hwang, H. Y., and Muller, D. A. (2025). Microscopic mechanisms of flexoelectricity in oxide membranes.
- LeBeau, J. (2022). fold slice. [https://github.com/LeBeauGroup/fold\\_slice](https://github.com/LeBeauGroup/fold_slice).
- LeBeau, J. M., Findlay, S. D., Allen, L. J., and Stemmer, S. (2010). Position averaged convergent beam electron diffraction: Theory and applications. *Ultramicroscopy*, 110(2):118–125.
- Li, G., Xu, M., Tang, W.-Q., Liu, Y., Chen, C., Zhang, D., Liu, L., Ning, S., Zhang, H., Gu, Z.-Y., Lai, Z., Muller, D. A., and Han, Y. (2025). Atomically resolved imaging of radiation-sensitive metal-organic frameworks via electron ptychography. *Nature Communications*, 16(1):914.
- Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52.
- Li, P., Edo, T., Batey, D., Rodenburg, J., and Maiden, A. (2016). Breaking ambiguities in mixed state ptychography. *Optics express*, 24(8):9038–9052.
- Loetgering, L., Du, M., Boonzajer Flaes, D., Aidukas, T., Wechsler, F., Penagos Molina, D. S., Rose, M., Pelekanidis, A., Eschen, W., Hess, J., et al. (2023). Ptylab. m/py/jl: a cross-platform, open-source inverse modeling toolbox for conventional and fourier ptychography. *Optics Express*, 31(9):13763–13797.
- Luke, D. R. (2004). Relaxed averaged alternating reflections for diffraction imaging. *Inverse problems*, 21(1):37.
- Madsen, J. and Susi, T. (2021). The abtem code: transmission electron microscopy from first principles. *Open Research Europe*, 1:24.
- Maiden, A., Humphry, M., Sarahan, M., Kraus, B., and Rodenburg, J. (2012a). An annealing algorithm to correct positioning errors in ptychography. *Ultramicroscopy*, 120:64–72.
- Maiden, A. M., Humphry, M. J., and Rodenburg, J. M. (2012b). Ptychographic transmission microscopy in three dimensions using a multi-slice approach. *JOSA A*, 29(8):1606–1614.
- Maiden, A. M. and Rodenburg, J. M. (2009). An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy*, 109(10):1256–1262.
- Moshtaghpour, A., Velazco-Torrejon, A., Robinson, A. W., Browning, N. D., and Kirkland, A. I. (2025). Lorepie: L0 regularized extended ptychographical iterative engine for low-dose and fast electron ptychography. *Optics Express*, 33(5):9357–9368.
- Nellist, P., McCallum, B., and Rodenburg, J. M. (1995). Resolution beyond the ‘information limit’ in transmission electron microscopy. *nature*, 374(6523):630–632.
- Nguyen, K. X., Jiang, Y., Lee, C.-H., Kharel, P., Zhang, Y., van der Zande, A. M., and Huang, P. Y. (2024). Achieving sub-0.5-angstrom-resolution ptychography in an uncorrected electron microscope. *Science*, 383(6685):865–870.
- Odstrcil, M., Baksh, P., Boden, S., Card, R., Chad, J., Frey, J., and Brocklesby, W. (2016). Ptychographic coherent diffractive imaging with orthogonal probe relaxation. *Optics express*, 24(8):8360–8369.
- Odstrcil, M., Menzel, A., and Guizar-Sicairos, M. (2018). Iterative least-squares solver for generalized maximum-likelihood ptychography. *Optics express*, 26(3):3108–3123.
- O’Leary, C., Allen, C., Huang, C., Kim, J., Liberti, E., Nellist, P., and Kirkland, A. (2020). Phase reconstruction using fast binary 4d stem data. *Applied Physics Letters*, 116(12).
- Ophus, C., Zeltmann, S. E., Bruefach, A., Rakowski, A., Savitzky, B. H., Minor, A. M., and Scott, M. C. (2022). Automated crystal orientation mapping in py4DSTEM using sparse correlation matching. *Microscopy and Microanalysis*, 28(2):390–403.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- Pattison, A. J., Ribet, S. M., Noack, M. M., Varnavides, G., Park, K., Kirkland, E., Park, J., Ophus, C., and Ercius, P. (2024). Beacon-automated aberration correction for scanning transmission electron microscopy using bayesian optimization. *arXiv preprint arXiv:2410.14873*.
- Pennycook, T. J., Lupini, A. R., Yang, H., Murfitt, M. F., Jones, L., and Nellist, P. D. (2015). Efficient phase contrast imaging in stem using a pixelated detector. part 1: Experimental demonstration at atomic resolution. *Ultramicroscopy*, 151:160–167.
- Philipp, H. T., Tate, M. W., Shanks, K. S., Mele, L., Peemen, M., Dona, P., Hartong, R., van Veen, G., Shao, Y.-T., Chen, Z., et al. (2022). Very-high dynamic range, 10,000 frames/second pixel array detector for electron microscopy. *Microscopy and Microanalysis*, 28(2):425–440.
- Putkunz, C. T., D’Alfonso, A. J., Morgan, A. J., Weyland, M., Dwyer, C., Bourgeois, L., Etheridge, J., Roberts, A., Scholten, R. E., Nugent, K. A., et al. (2012). Atom-scale ptychographic electron diffractive imaging of boron nitride cones. *Physical Review Letters*, 108(7):073901.
- Rall, L. B. (1981). *Automatic differentiation: Techniques and applications*. Springer.
- Rodenburg, J. and Bates, R. (1992). The theory of super-resolution electron microscopy via wigner-distribution deconvolution. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 339(1655):521–553.
- Rodenburg, J. and Maiden, A. (2019). Ptychography. *Springer Handbook of Microscopy*, pages 819–904.
- Rodenburg, J., McCallum, B., and Nellist, P. (1993). Experimental tests on double-resolution coherent imaging via stem. *Ultramicroscopy*, 48(3):304–314.
- Rodenburg, J. M., Hurst, A., Cullis, A. G., Dobson, B. R., Pfeiffer, F., Bunk, O., David, C., Jefimovs, . f. K., and Johnson, I. (2007). Hard-x-ray lensless imaging of extended objects. *Physical review letters*, 98(3):034801.
- Romanov, A., Cho, M. G., Scott, M. C., and Pelz, P. (2024). Multi-slice electron ptychographic tomography for three-dimensional phase-contrast microscopy beyond the depth of

- focus limits. *Journal of Physics: Materials*, 8(1):015005.
- Savitzky, B. H., Zeltmann, S. E., Hughes, L. A., Brown, H. G., Zhao, S., Pelz, P. M., Pekin, T. C., Barnard, E. S., Donohue, J., Rangel DaCosta, L., Kennedy, E., Xie, Y., Janish, M. T., Schneider, M. M., Herring, P., Gopal, C., Anapolsky, A., Dhall, R., Bustillo, K. C., Ercius, P., Scott, M. C., Ciston, J., Minor, A. M., and Ophus, C. (2021). py4dstem: A software package for four-dimensional scanning transmission electron microscopy data analysis. *Microscopy and Microanalysis*, 27(4):712–743.
- Sayre, D. (1952). The squaring method: a new method for phase determination. *Acta Crystallographica*, 5(1):60–65.
- Schloz, M., Pekin, T. C., Brown, H. G., Byrne, D. O., Esser, B. D., Terzoudis-Lumsden, E., Taniguchi, T., Watanabe, K., Findlay, S. D., Haas, B., et al. (2024). Improved three-dimensional reconstructions in electron ptychography through defocus series measurements. *arXiv preprint arXiv:2406.01141*.
- Schloz, M., Pekin, T. C., Chen, Z., Van den Broek, W., Muller, D. A., and Koch, C. T. (2020). Overcoming information reduced data and experimentally uncertain parameters in ptychography with regularized optimization. *Optics Express*, 28(19):28306–28323.
- Seifert, J., Bouchet, D., Loetgering, L., and Mosk, A. P. (2021). Efficient and flexible approach to ptychography using an optimization framework based on automatic differentiation. *OSA Continuum*, 4(1):121–128.
- Sha, H., Cui, J., and Yu, R. (2022). Deep sub-angstrom resolution imaging by electron ptychography with misorientation correction. *Science Advances*, 8(19):eabn2275.
- Strauch, A., Weber, D., Clausen, A., Lesnichaia, A., Bangun, A., März, B., Lyu, F. J., Chen, Q., Rosenauer, A., Dunin-Borkowski, R., et al. (2021). Live processing of momentum-resolved stem data for first moment imaging and ptychography. *Microscopy and Microanalysis*, 27(5):1078–1092.
- Tate, M. W., Purohit, P., Chamberlain, D., Nguyen, K. X., Hovden, R., Chang, C. S., Deb, P., Turgut, E., Heron, J. T., Schlom, D. G., et al. (2016). High dynamic range pixel array detector for scanning transmission electron microscopy. *Microscopy and Microanalysis*, 22(1):237–249.
- Terzoudis-Lumsden, E., Petersen, T., Brown, H. G., Pelz, P., Ophus, C., and Findlay, S. D. (2023). Resolution of virtual depth sectioning from four-dimensional scanning transmission electron microscopy. *Microscopy and Microanalysis*, 29(4):1409–1421.
- Thibault, P., Dierolf, M., Bunk, O., Menzel, A., and Pfeiffer, F. (2009). Probe retrieval in ptychographic coherent diffractive imaging. *Ultramicroscopy*, 109(4):338–343.
- Thibault, P. and Guizar-Sicairos, M. (2012). Maximum-likelihood refinement for coherent diffractive imaging. *New Journal of Physics*, 14(6):063004.
- Thibault, P. and Menzel, A. (2013). Reconstructing state mixtures from diffraction measurements. *Nature*, 494(7435):68–71.
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.
- Tsai, E. H. R., Usov, I., Diaz, A., Menzel, A., and Guizar-Sicairos, M. (2016). X-ray ptychography with extended depth of field. *Opt. Express*, 24(25):29089–29108.
- Van den Broek, W. and Koch, C. T. (2012). Method for retrieval of the three-dimensional object potential by inversion of dynamical electron scattering. *Physical review letters*, 109(24):245502.
- Van den Broek, W. and Koch, C. T. (2013). General framework for quantitative three-dimensional reconstruction from arbitrary detection geometries in tem. *Physical Review B—Condensed Matter and Materials Physics*, 87(18):184108.
- Varnavides, G., Ribet, S. M., Zeltmann, S. E., Yu, Y., Savitzky, B. H., Byrne, D. O., Allen, F. I., Dravid, V. P., Scott, M. C., and Ophus, C. (2023). Iterative phase retrieval algorithms for scanning transmission electron microscopy. *arXiv preprint arXiv:2309.05250*.
- Wakonig, K., Stadler, H.-C., Odstrčil, M., Tsai, E. H. R., Diaz, A., Holler, M., Usov, I., Raabe, J., Menzel, A., and Guizar-Sicairos, M. (2020). *PtychoShelves*, a versatile high-level framework for high-performance analysis of ptychographic data. *Journal of Applied Crystallography*, 53(2):574–586.
- Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612.
- Watanabe, S. (2023). Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*.
- Watanabe, S., Bansal, A., and Hutter, F. (2023). Ped-anova: efficiently quantifying hyperparameter importance in arbitrary subspaces. *arXiv preprint arXiv:2304.10255*.
- Wu, L., Yoo, S., Chu, Y. S., Huang, X., and Robinson, I. K. (2024). Dose-efficient automatic differentiation for ptychographic reconstruction. *Optica*, 11(6):821–830.
- Xu, W., Ning, S., Sheng, P., Lin, H., Kirkland, A. I., Peng, Y., and Zhang, F. (2024). A high-performance reconstruction method for partially coherent ptychography. *Ultramicroscopy*, 267:114068.
- Yu, Z., Batson, P. E., and Silcox, J. (2003). Artifacts in aberration-corrected adf-stem imaging. *Ultramicroscopy*, 96(3–4):275–284.
- Zambon, P., Bottinelli, S., Schnyder, R., Musarra, D., Boye, D., Dudina, A., Lehmann, N., De Carlo, S., Rissi, M., Schulze-Briese, C., et al. (2023). Kite: high frame rate, high count rate pixelated electron counting asic for 4d stem applications featuring high-z sensor. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1048:167888.
- Zhang, H., Li, G., Zhang, J., Zhang, D., Chen, Z., Liu, X., Guo, P., Zhu, Y., Chen, C., Liu, L., Guo, X., and Han, Y. (2023). Three-dimensional inhomogeneity of zeolite structure and composition revealed by electron ptychography. *Science*, 380(6645):633–638.

## **Supplementary Information**

---

# **PtyRAD: A High-performance and Flexible Ptychographic Reconstruction Framework with Automatic Differentiation**

Chia-Hao Lee,<sup>\*,†</sup> Steven E. Zeltmann,<sup>‡</sup> Dasol Yoon,<sup>¶</sup> Desheng Ma,<sup>†</sup> and David A. Muller<sup>\*,†,§</sup>

<sup>†</sup>*School of Applied and Engineering Physics, Cornell University, Ithaca, New York 14850,  
United States*

<sup>‡</sup>*Platform for the Accelerated Realization, Analysis, and Discovery of Interface Materials,  
Cornell University, Ithaca, New York 14850, United States*

<sup>¶</sup>*Department of Materials Science and Engineering, Cornell University, Ithaca, New York  
14850, United States*

<sup>§</sup>*Kavli Institute at Cornell for Nanoscale Science, Cornell University, Ithaca, New York  
14850, United States*

E-mail: [chia-hao.lee@cornell.edu](mailto:chia-hao.lee@cornell.edu); [david.a.muller@cornell.edu](mailto:david.a.muller@cornell.edu)

# Benchmarking Details

## Hardware and OS Specifications

All benchmarking was performed on a private cluster hosted by the Cornell University Center for Advanced Computing (CAC). The cluster consists of a head node running OpenHPC 2.3 with Rocky Linux 8.4. Computations were carried out on four identical compute nodes, each equipped with dual 64-core AMD EPYC 7713 processors (with hyperthreading enabled, yielding 256 logical CPUs), 1TB of RAM, and four NVIDIA A100 (80 GB) GPUs.

For each reconstruction task shown in Figure 3, all ptychographic reconstruction packages were benchmarked using the same hardware configuration to ensure a fair comparison. Specifically, each job was assigned a 20 GB MIG (Multi-Instance GPU) slice of an NVIDIA A100 80 GB GPU and four CPU cores. MIG is a feature of NVIDIA Ampere GPUs that partitions a single physical GPU into multiple isolated instances, each with dedicated compute cores and memory, allowing multiple jobs to run concurrently without interference. A 20 GB MIG slice corresponds to approximately 2/7 (or 28.6%) of the total GPU memory bandwidth and streaming multiprocessors (SMs) of the full A100 GPU. The number of CPU cores and total CPU memory had negligible impact on the reconstruction time, as all packages fully load the diffraction patterns into GPU memory at the start of computation. Resource allocation, including memory management, was handled automatically by the Slurm job scheduler.

## Software Version and Environments

- **PtyRAD** was run using Python 3.11.9, with PyTorch 2.1.2 and CUDA 11.8. The full Python environment specification is provided in our GitHub repository <https://github.com/chiaha03/ptyrad>.
- **py4DSTEM** was benchmarked using version 0.14.18,<sup>1</sup> specifically the `baf9e30` com-

mit from the original “`dev`” branch. The environment was built with Python 3.11.10, CuPy 13.3.0, and CUDA 11.8. We introduced several lightweight modifications to the package—such as results saving, timing utilities, and wrapper functions—to facilitate systematic reconstruction from input files. These changes do not modify the core algorithms or affect performance timing. Our modified version is openly available at <https://github.com/chiahao3/py4DSTEM/tree/benchmark>, and an installation guide is included.

- **PtychoShelves** was tested using the `fold_slice` fork,<sup>2</sup> specifically the `d9a1204` commit from the original “`main`” branch, executed with MATLAB 2021a.

## Package configurations

While all reconstructions in Figures 3 and 4 use the same physical model complexity (e.g., number of object slices, probe modes) and batch sizes across packages, each software package implements its own optimization algorithms and regularization strategies. We summarize the most relevant settings below. Full parameter files for each reconstruction are provided in our Zenodo record for full transparency and reproducibility.

- **PtyRAD**: Uses the Adam optimizer with optimizable parameters including the object amplitude and phase, probe, and probe positions. The base learning rate is set to  $5 \times 10^{-4}$ , with a separate learning rate of  $1 \times 10^{-4}$  for the probe. The loss function includes a normalized root-mean-square error (NRMSE) under a Gaussian noise model, along with an L1 sparsity regularization term weighted by 0.1. Additional regularization techniques include the probe mode orthogonalization, a real-space  $r_z$  filter with  $\sigma_z = 1$  for multislice reconstruction, positivity constraint for object phase, and a thresholding within [0.98, 1.02] for the object amplitude.
- **PtychoShelves**: Configured with the `GPU_MS` engine using the `LSQML` algorithm and the `MLs` option, which corresponds to an L1 likelihood model consistent with

the Gaussian noise assumption. The multislice regularization weight  $\beta$  is set to 0.1. Optimizable parameters include the complex object, probe, and probe positions. No variable probe is used in these reconstructions.

- **py4DSTEM:** Uses the “gradient descent” solver with default update step size of 0.5. Although we find that setting the step size to 0.1 significantly improves numerical stability. Optimizable parameters include the complex object, probe, and probe positions. The multislice regularization parameter  $\beta$  is also set to 0.1, implemented as the `kz_regularization_gamma` parameter. Note that the multislice  $k_z$  filters are implemented differently across packages so the regularization strength may not be directly comparable.

Supplementary Table S1: Publicly available ptychography software packages.

Year	Reference	Supported Algorithms	Language	Notes
2015	(py)ptychoSTEM <sup>3-5</sup>	SSB,WDD	MATLAB, Python	Direct ptychographic phase reconstructions with 4D STEM data.
2016	PtyPy <sup>6</sup>	DM, RAAR, ePIE, ML	Python	Support of on-the-fly reconstructions (data being acquired); mixed probe and object
2018	NSLS-II (Ptycho.gui) <sup>7</sup>	DM	Python	Deployed on NSLS-II beamline machines with GUI
2019	ptychoSampling <sup>8</sup>	AD	Python	Generalized forward models: near-field ptychography and 3D Bragg projection ptychography
2020	PtychoShelves <sup>9</sup>	ePIE, DM, LSQML	MATLAB	MATLAB-based GPU-accelerated engine
2020	PyNX <sup>10</sup>	ER, RAAR, DM, ML	Python	CDI, ptychography, wavefront propagation, scattering calculations, etc.
2020	abTEM <sup>11</sup>	PIE	Python	A flexible package for simulating TEM experiments
2021	PtychoNN <sup>12</sup>	Neural Network-based	Python	Predicts sample amplitude and phase from input diffraction data alone
2021	Ptychopy <sup>13</sup>	ePIE, DM, LSQML	Python	CUDA C++ backend
2021	py4DSTEM <sup>1</sup>	SSB, WDD, DM, RAAR, GD	Python	A complete toolbox for 4D-STEM data processing beyond ptychography
2021	Ptychography 4.0 <sup>14</sup>	SSB	Python	Support of live processing of data (SSB method)
2021	Adorym <sup>15</sup>	AD	Python	HPC AD package for 2D/3D ptychography, CDI, holography, and tomography
2021	PtychoKeras <sup>16</sup>	AD	Python	AD-based using TensorFlow and Keras
2022	Tike <sup>17</sup>	ePIE, LSQML	Python	Laminography, tomography from ptychographic reconstructions
2022	SciComPy <sup>18</sup>	AD	Python	A modular framework for the design of new ptychography algorithms with simulations
2022	Airpi <sup>19</sup>	Pretrained NN	Python	CNN recovered complex electron wave function from CBEDs
2023	Deep-CDI <sup>20</sup>	Pretrained NN	Python	CNN predicted convergent diffraction imaging (CDI)
2023	PtyLab <sup>21</sup>	ePIE family	MATLAB, Python, Julia	Fourier and conventional ptychography; variants of ePIE: mPIE, zPIE, aPIE, pcPIE, e3PIE
2024	torchslice <sup>22</sup>	AD,	Python	Optimize discrete atomic models to incorporate thermal diffuse scattering
2024	PtychoFormer <sup>23</sup>	Pretrained NN	Python	Predicts and stitches local patches from sparse scans with optional integration of ePIE

ML: Maximum Likelihood

WDD: Wigner Distribution Deconvolution

SSB: Single Side Band

DM: Difference Map

ePIE: extended Ptychographical Iterative Engine

ER: Error Reduction

LSQML: Least-Squares Maximum-Likelihood

RAAR: Relaxed-Averaged Alternating Reflections

GD: Gradient Descent

AD: Automatic Differentiation

NN: Neural Network

Supplementary Table S2: Hyperparameters implemented for automatic tuning in PtyRAD.

Hyperparameters	Description
Optimizer	Optimization algorithm for gradient descent updates
Batch size	Number of diffraction patterns processed per update step
Learning rate (probe) <sup>a</sup>	Scaling factor for gradient updates to probe
Learning rate (object amplitude) <sup>a</sup>	Scaling factor for gradient updates to object amplitude
Learning rate (object phase) <sup>a</sup>	Scaling factor for gradient updates to object phase
Learning rate (probe position) <sup>a</sup>	Scaling factor for gradient updates to probe positions
Learning rate (object tilts) <sup>a</sup>	Scaling factor for gradient updates to object tilt angles
Learning rate (slice thickness) <sup>a</sup>	Scaling factor for gradient updates to slice thickness
Real space px size <sup>b</sup>	Pixel size for probe and object arrays
Number of probe modes	Number of incoherent probe modes for mixed states
Convergence angle	Probe convergence semi-angle
Defocus	Defocus in the simulated initial probe
C3	Third-order spherical aberration
C5	Fifth-order spherical aberration
Number of object layers	Number of depth slices for the multislice object
Slice thickness <sup>c</sup>	Propagation distance between object slice
Scan affine (scale)	Global scale factor for scan grid coordinates
Scan affine (asymmetry)	Relative scaling along fast vs. slow scan axes
Scan affine (rotation)	In-plane rotation of the scan grid
Scan affine (shear)	Shear distortion of the scan pattern
Object tilt (y) <sup>d</sup>	Tilt of the object along the <i>y</i> direction
Object tilt (x) <sup>d</sup>	Tilt of the object along the <i>x</i> direction

<sup>a</sup> The applied update is the gradient scaled by the learning rate.

<sup>b</sup> The pixel size is determined by the diffraction pattern calibration of  $k_{\max}$ .

<sup>c</sup> Slice thickness can be optimized with AD during reconstruction, or tuned as a hyperparameter.

<sup>d</sup> Object tilt can be optimized with AD during reconstruction, or tuned as a hyperparameter.

Supplementary Table S3: Acquisition and reconstruction parameters for each experimental dataset used in Figure 2.

Dataset	MOSS-6 <sup>24</sup>	ZSM-5 <sup>25</sup>	tBL-WSe <sub>2</sub> <sup>26</sup>	PrScO <sub>3</sub> <sup>27</sup>
Acceleration voltage (kV)	300	300	80	300
Convergence angle (mrad)	10	15	24.9	21.4
Defocus (Å) <sup>a</sup>	885	350	0	0
Real space px size (Å)	0.2962	0.3591	0.1494	0.0934
Scan pattern	256×256	256×256	128×128	64×64 <sup>b</sup>
Scan step size (Å)	1.051	0.3989	0.429	0.41
Collection angle (mrad)	33.2	27.4	139.7	105.1
$k_{\max}$ (Å <sup>-1</sup> )	1.69	1.39	3.35	5.35
Detector pixel	128	128	128	256 <sup>c</sup>
Dose (e <sup>-</sup> /Å <sup>2</sup> )	100	3500	7.55E+05	1.22E+06
Slice thickness (Å)	40	40	2	10
Number of slices	5	8	6	21
Probe modes	2	6	6	8
Batch size	512	32	32	32
Iteration	20	2000	4000	4000

<sup>a</sup> Positive defocus corresponds to underfocus condition.

<sup>b</sup> The available dataset includes only a subset of scan positions used in Chen et al.<sup>27</sup>.

<sup>c</sup> Diffraction patterns were padded from 128 to 256 pixels; the available dataset was already zero-padded.

Supplementary Table S4: Parameters for the simulated tBL-WSe<sub>2</sub> dataset used in Figure 3a

Dataset	Simulated tBL-WSe <sub>2</sub>
Package	abTEM <sup>11</sup>
Version	1.0.6 (pypi)
Interlayer twist (°)	3
Se vacancy density (%)	2
Supercell size (x, y, z) (Å)	(85.72, 85.66, 14.75)
Potential shape (x, y, z) (px)	(861, 861, 15)
Real space px size (Å)	0.0996 <sup>a</sup>
Slice thickness (Å)	1
Frozen phonon configurations	25
Phonon perturbation std. (Å)	0.1
Acceleration voltage (kV)	80
Convergence angle (mrad)	24.9
Defocus (Å)	0
$C_3$ (nm)	500
$C_c$ (nm)	1000
Energy spread std. (eV)	0.35 <sup>b</sup>
Focal spread std. (Å)	43.75
Number of defoci	5
Source size std. (Å)	0.34 <sup>c</sup>
Scan pattern	128×128
Scan step size (Å)	0.429
Collection angle (mrad)	139.7 <sup>d</sup>
$k_{\max}$ (Å <sup>-1</sup> )	3.35 <sup>d</sup>
Simulated CBED size (px)	574×574
Final CBED size (px)	128×128 <sup>e</sup>
Dose (e <sup>-</sup> /Å <sup>2</sup> )	1.0E+06

<sup>a</sup> abTEM crops the diffraction pattern to 2/3  $k_{\max}$  for antialiasing at the end so the final pixel size is  $0.0996 \times 1.5 = 0.1494$ .

<sup>b</sup> Equivalent to an energy spread FWHM of 0.82 eV.

<sup>c</sup> Equivalent to a source size FWHM of 0.80 Å.

<sup>d</sup> Both collection angle and  $k_{\max}$  show the final value after antialiasing cropping.

<sup>e</sup> The diffraction patterns are resampled to match the k-space sampling of the experimental dataset.

Supplementary Table S5: Iteration times and GPU metrics from `nvidia-smi` for various packages and batch sizes. The reconstruction was conducted on the experimental tBL-WSe<sub>2</sub> dataset using a full NVIDIA 80GB A100 GPU with 12 probe modes and 6 object slices. The best value for iteration time and utilization metrics are bolded. Reported values are averaged over a 9-minute window sampled at 1-second intervals, beginning after a 1-minute warmup period to ensure steady-state performance.

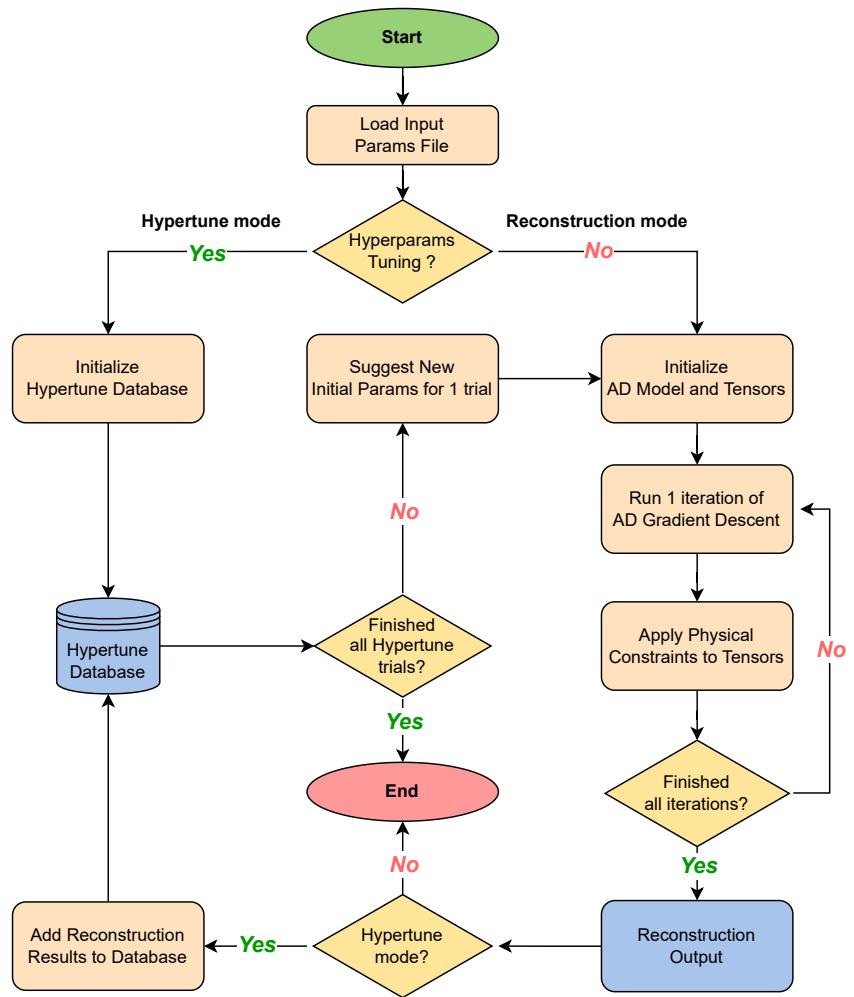
Batch Size	Package	Iter. Time (sec)	GPU Util. (%) <sup>a</sup>	Mem. Util. (%) <sup>b</sup>	Mem. Used (MB)	Temp. (°C)	Power (W)
16	ptyrad	<b>9.83</b>	<b>76.07</b>	<b>41.77</b>	3103	52	216
	ptyshv	165.00	11.12	0.97	2376	38	76
	py4dstem	250.87	13.52	1.43	1730	39	78
64	ptyrad	<b>6.95</b>	<b>91.88</b>	<b>62.70</b>	4065	55	252
	ptyshv	46.80	16.06	3.48	4062	39	84
	py4dstem	67.79	20.33	4.94	2896	40	91
256	ptyrad	<b>5.85</b>	<b>97.92</b>	<b>76.88</b>	7645	61	300
	ptyshv	13.50	41.28	28.96	7844	47	163
	py4dstem	19.33	39.83	23.69	8767	45	146
1024	ptyrad	<b>5.69</b>	<b>99.34</b>	<b>78.79</b>	22333	59	289
	ptyshv	7.64	70.53	42.11	21777	51	216
	py4dstem	7.91	76.34	58.70	32575	62	266

<sup>a</sup> Percent of time over the past sampled period during which one or more kernels was executing on the GPU

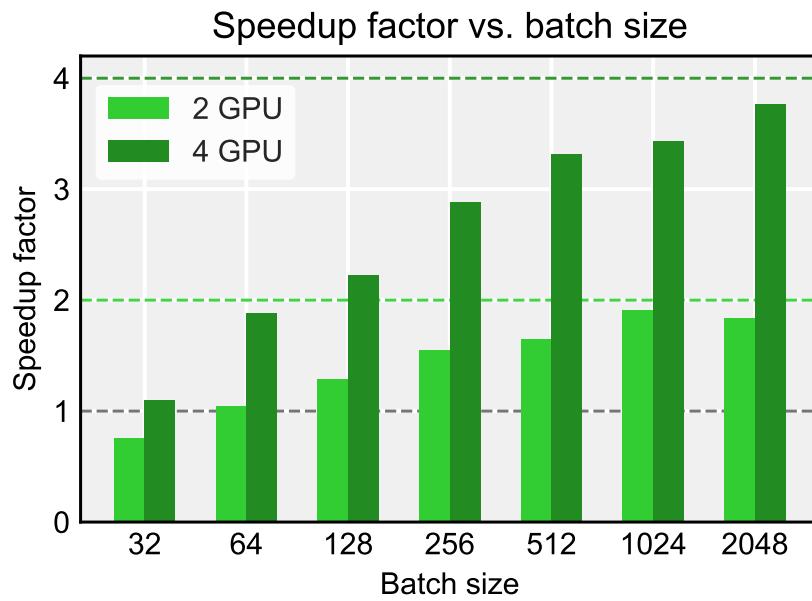
<sup>b</sup> Percent of time over the past sampled period during which GPU memory was being read or written

Supplementary Table S6: Search space used for hyperparameter tuning and importance analysis in Figure 6c. The optimization and importance analysis are performed in a 4-dimensional discrete space defined by the tunable hyperparameters listed in the table. The object amplitude learning rate is fixed at 5.0e-4, while the other learning rates and batch size are selected from the specified candidate values.

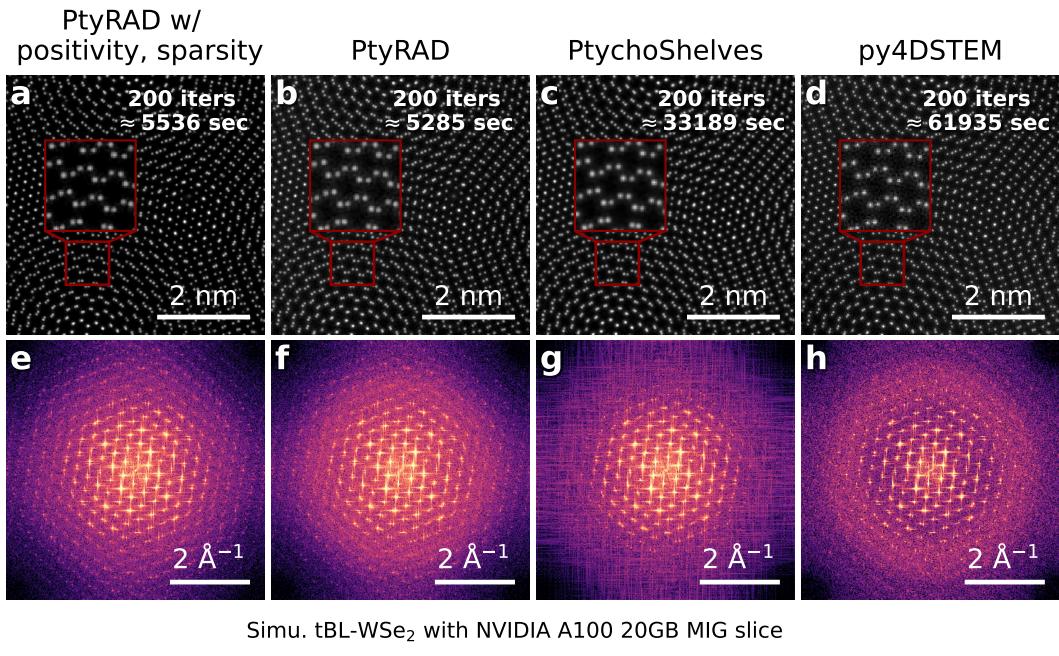
Hyperparameters	Search space
Batch size	16, 32, 64, 128, 256, 512, 1024
Learning rate (probe)	3.0e-2, 1.0e-2, 3.0e-3, 1.0e-3, 3.0e-4, 1.0e-4
Learning rate (object phase)	3.0e-2, 1.0e-2, 3.0e-3, 1.0e-3, 3.0e-4, 1.0e-4
Optimizer	Adam, AdamW, RMSprop, SGD



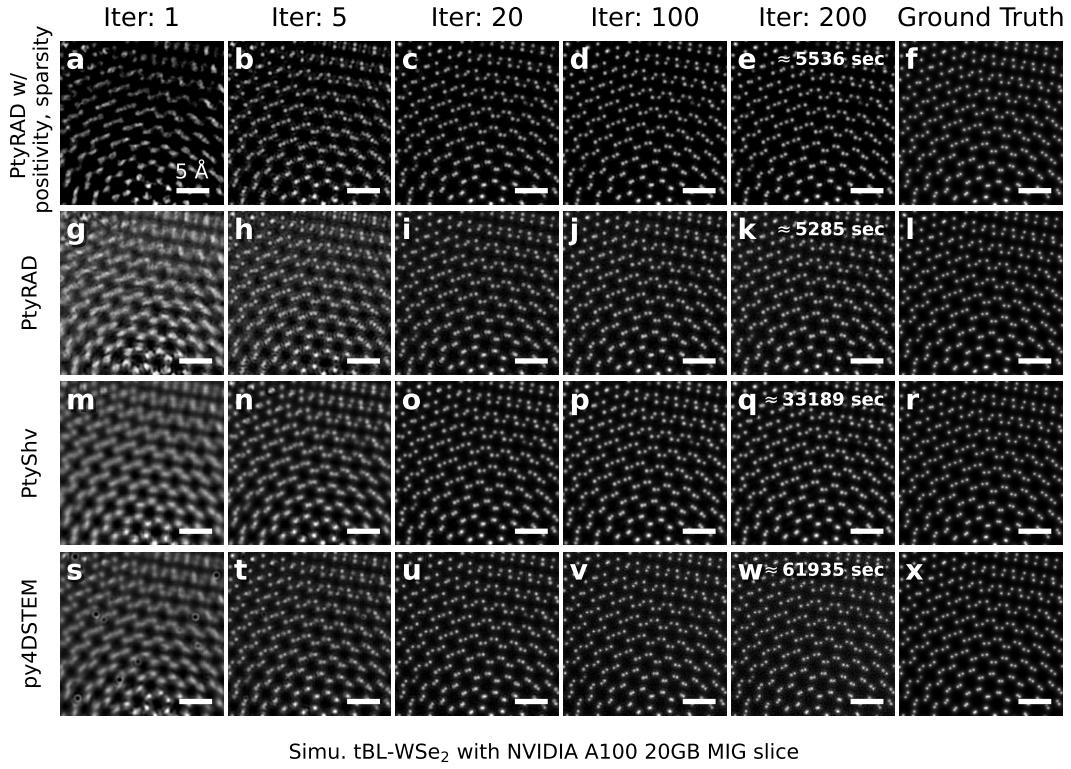
Supplementary Figure S1: Flowchart diagram of PtyRAD for both reconstruction and hyperparameter tuning (denoted as “hypertune”) modes.



Supplementary Figure S2: Speedup factors for PtyRAD reconstructions on the tBL-WSe<sub>2</sub> dataset using 2 and 4 NVIDIA A100 GPUs with varying batch sizes, normalized to single GPU performance. The dashed lines represent the ideal linear speedup for 2 and 4 GPUs. Performance scales sub-linearly with increasing GPU count, particularly for smaller batch sizes, where overhead and inter-GPU communication reduce speedup and may even cause slowdowns.



Supplementary Figure S3: Reconstructed phase images of the simulated twisted bilayer WSe<sub>2</sub> dataset after 200 iterations using (a) PtyRAD with positivity and sparsity regularization, (b) PtyRAD, (c) PtychoShelves, and (d) py4DSTEM and zoom-in insets. (e–h) Corresponding fast Fourier transform (FFT) power spectra of (a–d). Total reconstruction times taken for 12 probe modes, 6 object slices, batch size 16, and 200 iterations are labeled in the top right corners of each panel. All the reconstructions are conducted using the same hardware (a 20 GB MIG slice from an 80 GB NVIDIA A100). PtyRAD achieves higher information transfer given the same number of iterations and a shorter reconstruction time.



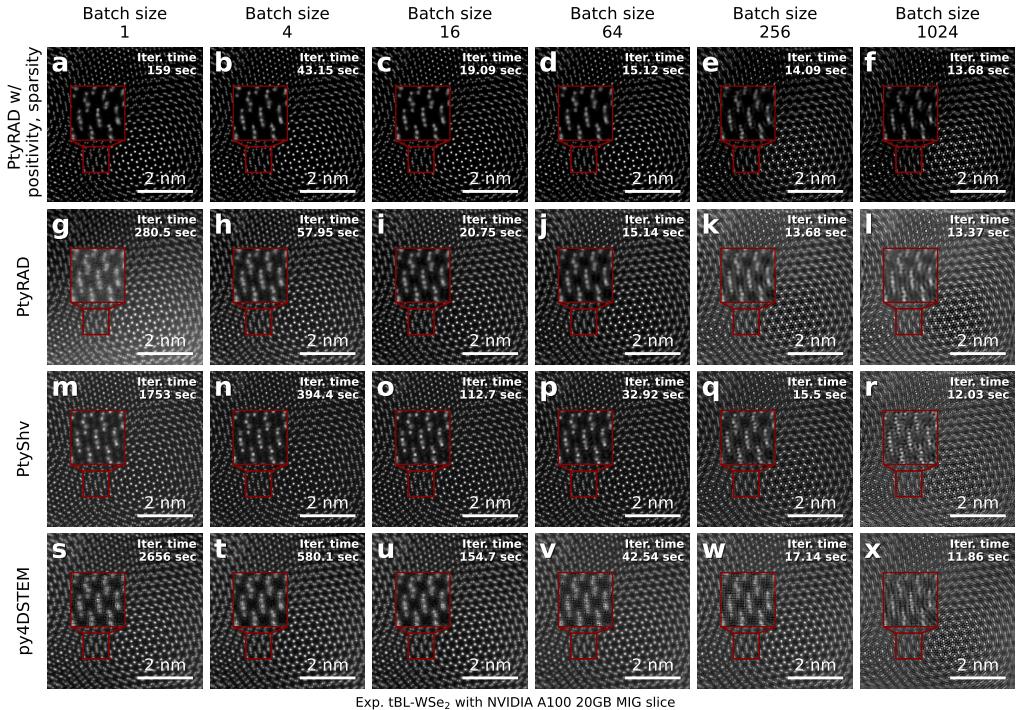
Supplementary Figure S4: Convergence comparison of different ptychographic reconstruction packages: (a–e) PtyRAD with positivity and sparsity regularization, (g–k) PtyRAD, (m–q) PtychoShelves (labeled as “PtyShv”), and (s–w) py4DSTEM. The benchmark was conducted on the simulated tBL-WSe<sub>2</sub> dataset using the same hardware (a 20 GB MIG slice from an 80 GB NVIDIA A100) with 12 probe modes, 6 object slices, and a batch size of 16 for all packages. Reconstructions are shown at selected iterations (1, 5, 20, 100, and 200) to illustrate the progression toward convergence. The total reconstruction time for 200 iterations (excluding initialization and result-saving time) is indicated in the corresponding column. The ground truth images (f, l, r, x) are shown in the last column for comparison. PtyRAD completes 200 iterations significantly faster than PtychoShelves and py4DSTEM given the tested condition, achieving a 6× to 12× speedup without compromising reconstruction quality. Scale bars are consistent across all panels.

### Iteration Time (sec) Benchmark

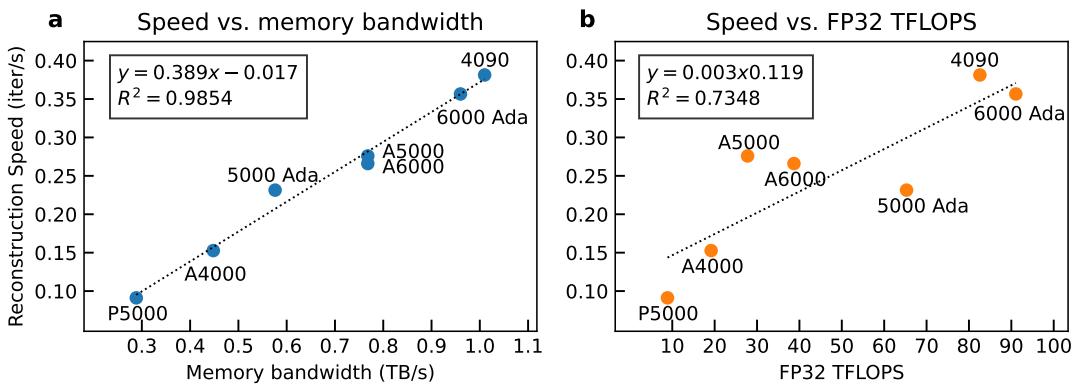
Measured on a 20 GB MG slice from an 80 GB Nvidia A100.

		1 slice						3 slices						6 slices					
		batch 16	batch 32	batch 64	batch 128	batch 256	batch 512	batch 16	batch 32	batch 64	batch 128	batch 256	batch 512	batch 16	batch 32	batch 64	batch 1024		
phyrad	<b>6.32 ± 0.2</b>	<b>6.34 ± 0.05</b>	<b>7.35 ± 0.09</b>	<b>9.98 ± 0.07</b>	<b>8.62 ± 0.1</b>	<b>9.02 ± 0.08</b>	<b>11.5 ± 0.09</b>	<b>16.1 ± 0.09</b>	<b>14.6 ± 0.1</b>	<b>14.6 ± 0.1</b>	<b>19 ± 0.09</b>	<b>26.6 ± 0.09</b>	<b>13.3 ± 0.07</b>	<b>13.3 ± 0.07</b>	<b>14.6 ± 0.1</b>	<b>26.6 ± 0.09</b>			
physrv	<b>6.87 ± 0.4</b>	<b>11.8 ± 0.7</b>	<b>19.2 ± 0.9</b>	<b>34.2 ± 2</b>	<b>12 ± 0.6</b>	<b>25 ± 1</b>	<b>45.5 ± 2</b>	<b>85 ± 3</b>	<b>20.5 ± 1</b>	<b>49.2 ± 2</b>	<b>90.1 ± 5</b>	<b>182 ± 7</b>	<b>11.3 ± 0.07</b>	<b>11.3 ± 0.07</b>	<b>14.6 ± 0.1</b>	<b>26.6 ± 0.09</b>			
py4dstem	<b>7.02 ± 0.3</b>	<b>14.6 ± 0.9</b>	<b>25.8 ± 1</b>	<b>49.1 ± 2</b>	<b>16.6 ± 0.8</b>	<b>39.2 ± 2</b>	<b>73.3 ± 3</b>	<b>142 ± 5</b>	<b>31 ± 2</b>	<b>74.2 ± 3</b>	<b>144 ± 7</b>	<b>283 ± 8</b>	<b>10.8 ± 0.6</b>	<b>25.4 ± 1</b>	<b>46 ± 2</b>	<b>90.2 ± 4</b>			
phyrad	<b>3.37 ± 0.1</b>	<b>4.03 ± 0.04</b>	<b>5.35 ± 0.1</b>	<b>7.9 ± 0.1</b>	<b>5.38 ± 0.09</b>	<b>6.97 ± 0.06</b>	<b>9.3 ± 0.08</b>	<b>13.8 ± 0.1</b>	<b>9.47 ± 0.1</b>	<b>12.4 ± 0.1</b>	<b>16.1 ± 0.1</b>	<b>23.6 ± 0.09</b>	<b>10.8 ± 0.6</b>	<b>13.9 ± 0.8</b>	<b>14.6 ± 0.1</b>	<b>23.6 ± 0.09</b>			
physrv	<b>3.75 ± 0.2</b>	<b>6.35 ± 0.4</b>	<b>10.2 ± 0.5</b>	<b>18.2 ± 0.8</b>	<b>6.16 ± 0.4</b>	<b>13.1 ± 0.6</b>	<b>23 ± 1</b>	<b>43.7 ± 2</b>	<b>15.5 ± 0.8</b>	<b>38.7 ± 1</b>	<b>73.3 ± 3</b>	<b>147 ± 5</b>	<b>8.37 ± 0.5</b>	<b>20 ± 1</b>	<b>37.6 ± 2</b>	<b>73.3 ± 3</b>			
py4dstem	<b>3.6 ± 0.2</b>	<b>7.37 ± 0.4</b>	<b>13.3 ± 0.5</b>	<b>25.4 ± 1</b>	<b>4.24 ± 0.1</b>	<b>10.5 ± 0.4</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>	<b>7.91 ± 0.3</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>	<b>78.9 ± 4</b>	<b>4.24 ± 0.1</b>	<b>10.5 ± 0.4</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>			
phyrad	<b>2.28 ± 0.06</b>	<b>3.19 ± 0.05</b>	<b>4.44 ± 0.04</b>	<b>7 ± 0.06</b>	<b>5.96 ± 0.06</b>	<b>8.19 ± 0.05</b>	<b>12.8 ± 0.1</b>	<b>12.8 ± 0.1</b>	<b>8.22 ± 0.05</b>	<b>11 ± 0.05</b>	<b>14.7 ± 0.08</b>	<b>22.3 ± 0.1</b>	<b>4.3 ± 0.06</b>	<b>7.48 ± 0.3</b>	<b>24.6 ± 1</b>	<b>25.7 ± 1</b>			
physrv	<b>2.23 ± 0.2</b>	<b>3.76 ± 0.2</b>	<b>6.02 ± 0.4</b>	<b>10.1 ± 0.8</b>	<b>3.6 ± 0.2</b>	<b>7.48 ± 0.3</b>	<b>13.2 ± 0.8</b>	<b>13.2 ± 0.8</b>	<b>13.9 ± 0.8</b>	<b>13.9 ± 0.8</b>	<b>14.7 ± 0.08</b>	<b>25.7 ± 1</b>	<b>4.0 ± 0.2</b>	<b>7.48 ± 0.3</b>	<b>24.6 ± 1</b>	<b>49.6 ± 2</b>			
py4dstem	<b>1.84 ± 0.07</b>	<b>3.92 ± 0.2</b>	<b>7.22 ± 0.3</b>	<b>13.8 ± 0.6</b>	<b>4.24 ± 0.1</b>	<b>10.5 ± 0.4</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>	<b>7.91 ± 0.3</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>	<b>78.9 ± 4</b>	<b>4.24 ± 0.1</b>	<b>10.5 ± 0.4</b>	<b>20.3 ± 0.8</b>	<b>39.7 ± 2</b>			
phyrad	<b>1.83 ± 0.03</b>	<b>2.7 ± 0.04</b>	<b>3.97 ± 0.03</b>	<b>6.32 ± 0.02</b>	<b>3.78 ± 0.03</b>	<b>5.33 ± 0.05</b>	<b>7.63 ± 0.05</b>	<b>12.1 ± 0.04</b>	<b>7.57 ± 0.04</b>	<b>10.2 ± 0.05</b>	<b>14 ± 0.05</b>	<b>21.4 ± 0.06</b>	<b>4.3 ± 0.03</b>	<b>6.51 ± 0.2</b>	<b>9.51 ± 0.3</b>	<b>17.5 ± 0.8</b>			
physrv	<b>1.6 ± 0.05</b>	<b>2.65 ± 0.09</b>	<b>4 ± 0.08</b>	<b>6.72 ± 0.2</b>	<b>2.62 ± 0.02</b>	<b>5.51 ± 0.2</b>	<b>9.51 ± 0.3</b>	<b>17.5 ± 0.8</b>	<b>13.41 ± 0.1</b>	<b>18.7 ± 0.3</b>	<b>18.7 ± 0.3</b>	<b>34.9 ± 1</b>	<b>6.3 ± 0.05</b>	<b>9.95 ± 0.2</b>	<b>13.9 ± 0.4</b>	<b>26.5 ± 0.3</b>			
py4dstem	<b>1.15 ± 0.05</b>	<b>2.36 ± 0.08</b>	<b>4.28 ± 0.1</b>	<b>8.18 ± 0.2</b>	<b>2.54 ± 0.1</b>	<b>6.3 ± 0.2</b>	<b>12.1 ± 0.4</b>	<b>23.4 ± 0.7</b>	<b>4.59 ± 0.2</b>	<b>12 ± 0.4</b>	<b>23.5 ± 0.8</b>	<b>46.2 ± 2</b>	<b>4.24 ± 0.1</b>	<b>6.3 ± 0.2</b>	<b>12.1 ± 0.4</b>	<b>23.5 ± 0.8</b>			
phyrad	<b>1.59 ± 0.06</b>	<b>2.46 ± 0.06</b>	<b>3.72 ± 0.01</b>	<b>6.26 ± 0.02</b>	<b>3.78 ± 0.03</b>	<b>5.04 ± 0.02</b>	<b>7.29 ± 0.02</b>	<b>11.8 ± 0.02</b>	<b>7.14 ± 0.01</b>	<b>10.2 ± 0.02</b>	<b>13.5 ± 0.02</b>	<b>21 ± 0.03</b>	<b>3.47 ± 0.06</b>	<b>4.17 ± 0.1</b>	<b>7.2 ± 0.3</b>	<b>13.4 ± 0.2</b>			
physrv	<b>1.33 ± 0.06</b>	<b>2 ± 0.07</b>	<b>2.98 ± 0.1</b>	<b>5.51 ± 0.2</b>	<b>2.15 ± 0.07</b>	<b>4.17 ± 0.1</b>	<b>7.2 ± 0.3</b>	<b>13.4 ± 0.2</b>	<b>3.41 ± 0.1</b>	<b>18.7 ± 0.3</b>	<b>18.7 ± 0.3</b>	<b>34.9 ± 1</b>	<b>3.47 ± 0.05</b>	<b>4.39 ± 0.05</b>	<b>7.54 ± 0.2</b>	<b>13.9 ± 0.4</b>			
py4dstem	<b>0.852 ± 0.03</b>	<b>1.7 ± 0.05</b>	<b>2.97 ± 0.09</b>	<b>5.51 ± 0.2</b>	<b>1.77 ± 0.03</b>	<b>4.33 ± 0.1</b>	<b>8.11 ± 0.2</b>	<b>15.6 ± 0.4</b>	<b>3.11 ± 0.08</b>	<b>8.23 ± 0.3</b>	<b>15.7 ± 0.4</b>	<b>30.7 ± 1</b>	<b>4.24 ± 0.1</b>	<b>6.3 ± 0.2</b>	<b>12.1 ± 0.4</b>	<b>23.5 ± 0.8</b>			
phyrad	<b>1.47 ± 0.06</b>	<b>2.34 ± 0.01</b>	<b>3.61 ± 0.05</b>	<b>6.15 ± 0.01</b>	<b>3.33 ± 0.01</b>	<b>4.87 ± 0.01</b>	<b>7.13 ± 0.009</b>	<b>11.7 ± 0.004</b>	<b>6.97 ± 0.01</b>	<b>9.52 ± 0.01</b>	<b>13.3 ± 0.01</b>	<b>26.9 ± 0.01</b>	<b>2.01 ± 0.06</b>	<b>3.86 ± 0.09</b>	<b>6.32 ± 0.2</b>	<b>13.7 ± 0.008</b>			
physrv	<b>1.22 ± 0.03</b>	<b>1.81 ± 0.04</b>	<b>2.61 ± 0.09</b>	<b>4.26 ± 0.2</b>	<b>2.01 ± 0.06</b>	<b>3.86 ± 0.09</b>	<b>6.32 ± 0.2</b>	<b>11.8 ± 0.2</b>	<b>3.15 ± 0.09</b>	<b>6.86 ± 0.08</b>	<b>12.3 ± 0.2</b>	<b>23.5 ± 0.4</b>	<b>2.01 ± 0.06</b>	<b>3.58 ± 0.1</b>	<b>6.58 ± 0.1</b>	<b>12.5 ± 0.3</b>			
py4dstem	<b>0.813 ± 0.05</b>	<b>1.46 ± 0.06</b>	<b>2.43 ± 0.05</b>	<b>4.43 ± 0.1</b>	<b>1.58 ± 0.06</b>	<b>3.58 ± 0.1</b>	<b>6.49 ± 0.1</b>	<b>12.5 ± 0.3</b>	<b>3.11 ± 0.08</b>	<b>8.23 ± 0.3</b>	<b>15.7 ± 0.4</b>	<b>30.7 ± 1</b>	<b>4.24 ± 0.1</b>	<b>6.3 ± 0.2</b>	<b>12.1 ± 0.4</b>	<b>23.5 ± 0.8</b>			
batch 512	phyrad	<b>1 probe</b>	3 probes	6 probes	12 probes	1 probe	3 probes	6 probes	12 probes	1 probe	3 probes	6 probes	1 probe	3 probes	6 probes	12 probes			
batch 1024	phyrad	<b>1 probe</b>	3 probes	6 probes	12 probes	1 probe	3 probes	6 probes	12 probes	1 probe	3 probes	6 probes	1 probe	3 probes	6 probes	12 probes			

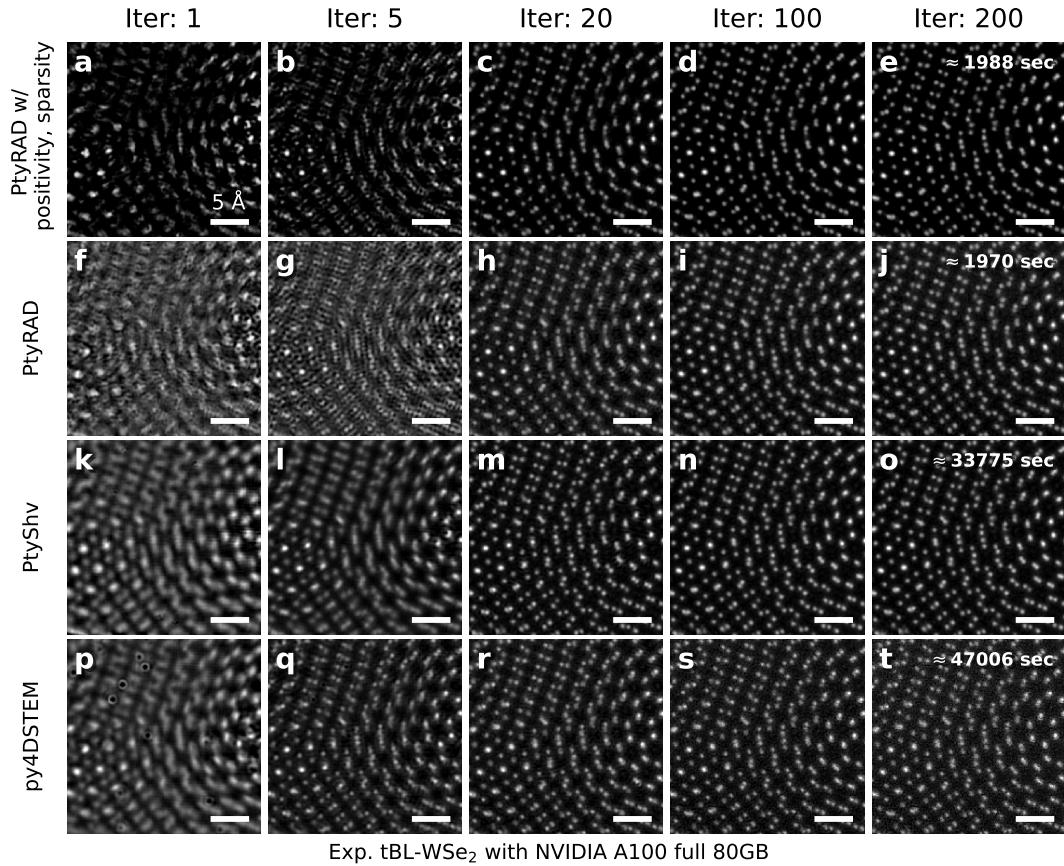
Supplementary Figure S5: Iteration time benchmark (mean  $\pm$  standard deviation over 5 runs) across varying batch sizes, probe modes, and object slices for each ptychographic reconstruction packages. Bolded entries indicate best performance in each column. Background shading identifies reconstruction package.



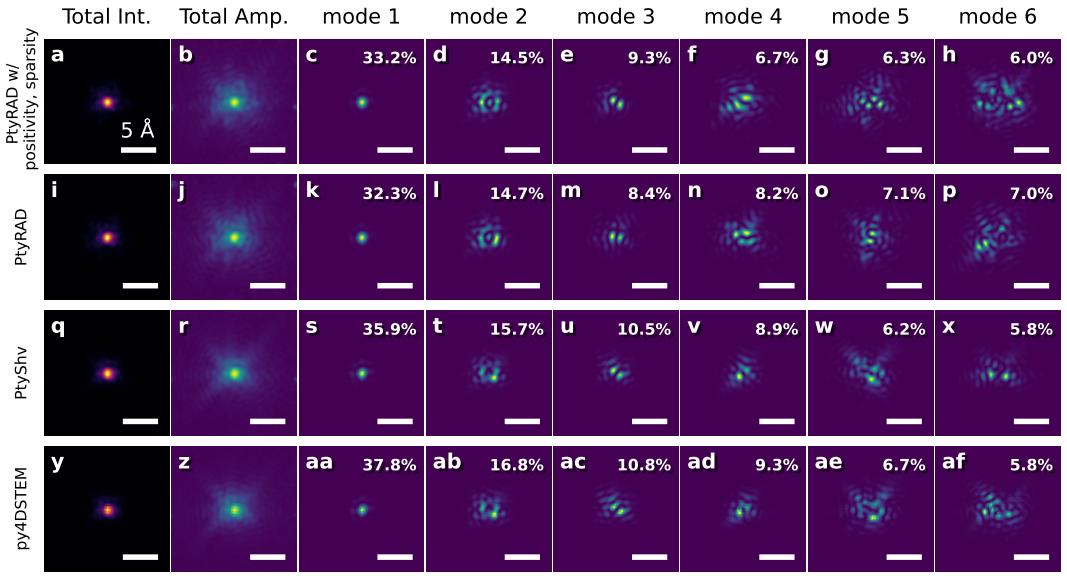
Supplementary Figure S6: Influence of batch size on reconstruction quality. Rows correspond to different reconstruction packages, including (a–f) PtyRAD with positivity and sparsity regularization, (g–l) PtychoShelves, and (m–r) py4DSTEM, respectively. Columns correspond to batch sizes of 4 to 1024. All the reconstructions are done on the experimental tBL-WSe<sub>2</sub> dataset with 128 by 128 diffraction patterns using 6 probe modes, 6 object slices, and 100 iterations. Insets highlight a magnified region to facilitate visual comparison of structural details. Smaller batch sizes (e.g., 16 and 64) generally yield sharper and more detailed reconstructions compared to larger batch sizes, which exhibit blurring and loss of contrast. Iteration times reconstructed with a 20 GB MIG slice of NVIDIA A100 are labeled on top-right corners. Note that PtychoShelves gives stronger but incorrect inter-atomic contrasts for these reconstructions, likely due to the excessive  $k_z$  regularization.



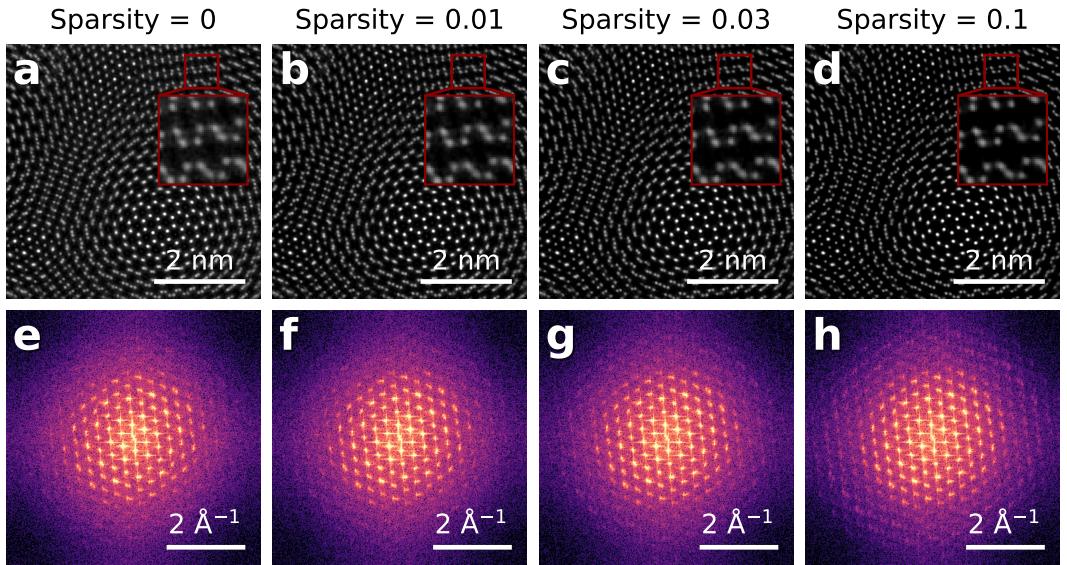
Supplementary Figure S7: Comparison of different GPUs for reconstruction speed using PtyRAD. Reconstruction speed (iterations per second) for PtyRAD is tested on the experimental tBL-WSe<sub>2</sub> dataset with 1 slice, 12 probes, and batch size of 256, plotted against (a) the memory bandwidth and (b) the floating point 32 TFLOPS of various NVIDIA GPUs rented via vast.ai. The GPUs tested include P5000, A4000, A5000, A6000, 5000 Ada, 6000 Ada, and 4090. We observe a strong linear correlation ( $R^2 = 0.9854$ ) between GPU memory bandwidth and reconstruction speed, suggesting that PtyRAD is more memory-bandwidth bound than compute-bound.



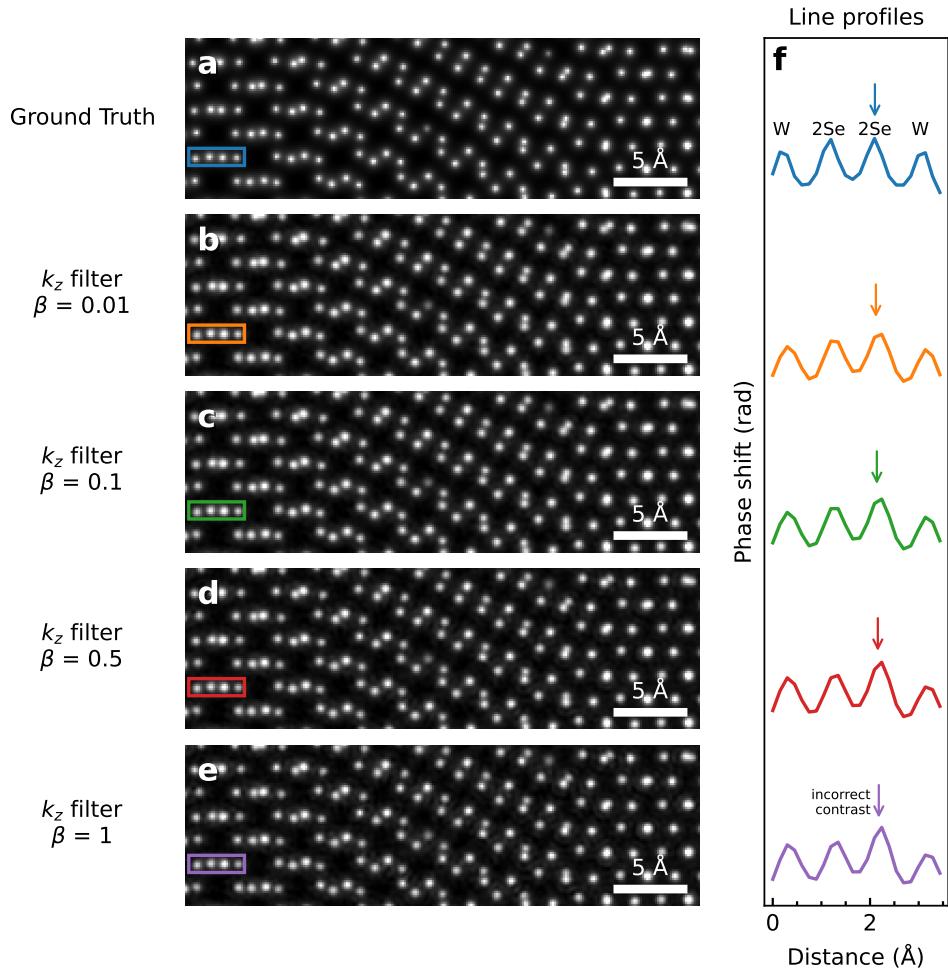
Supplementary Figure S8: Convergence comparison of different ptychographic reconstruction packages: (a–e) PtyRAD with positivity and sparsity regularization, (f–j) PtyRAD, (k–o) PtychoShelves (labeled as “PtyShv”), and (p–t) py4DSTEM. The benchmark was conducted on the experimental tBL-WSe<sub>2</sub> dataset using the same hardware (a full 80 GB NVIDIA A100) with 12 probe modes, 6 object slices, and a batch size of 16 for all packages. Reconstructions are shown at selected iterations (1, 5, 20, 100, and 200) to illustrate the progression toward convergence. The total reconstruction time for 200 iterations (excluding initialization and result-saving time) is indicated in the last column. PtyRAD completes 200 iterations significantly faster than PtychoShelves and py4DSTEM given the tested condition, achieving a 17× to 24× speedup without compromising reconstruction quality. Scale bars are consistent across all panels.



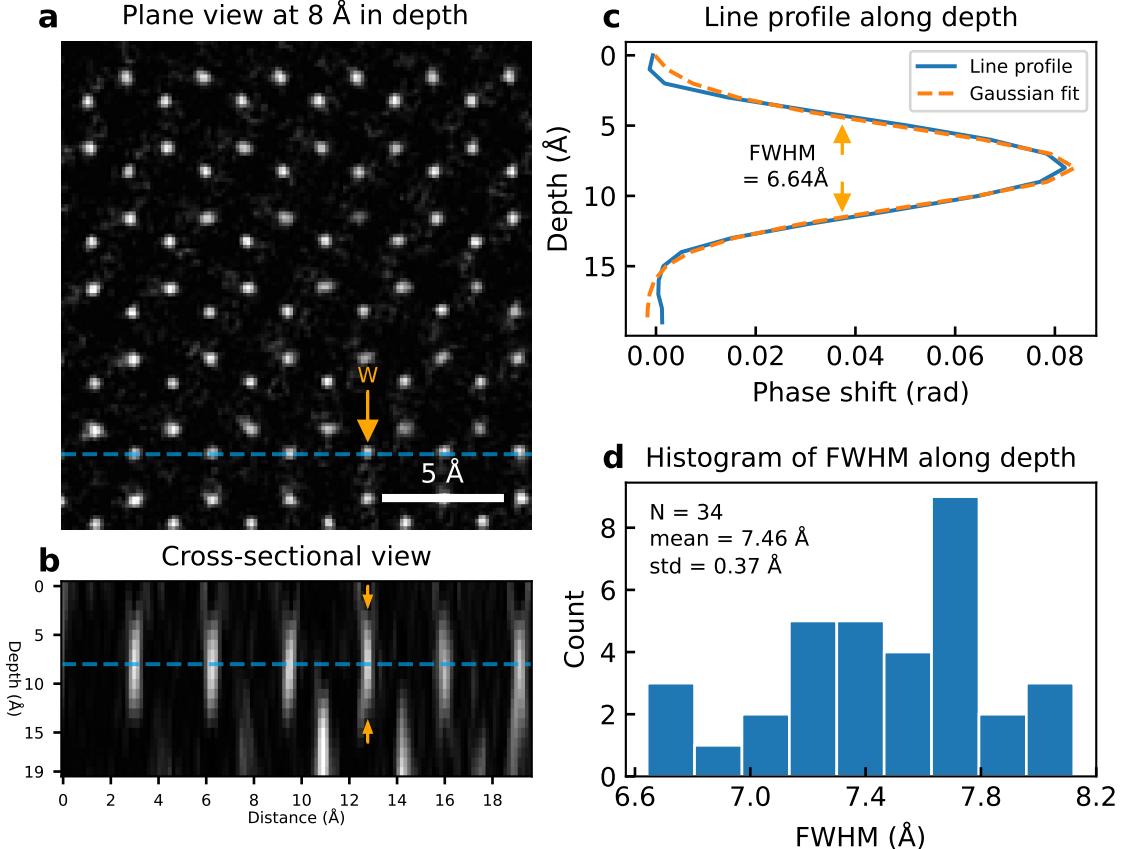
Supplementary Figure S9: Reconstructed mixed-state probes for different ptychographic reconstruction packages. All reconstructions were conducted on the experimental tBL-WSe<sub>2</sub> dataset using the same hardware (a full 80 GB NVIDIA A100) with 12 probe modes, 6 object slices, and a batch size of 16 for 200 iterations. For clarity, only the first six modes are shown.



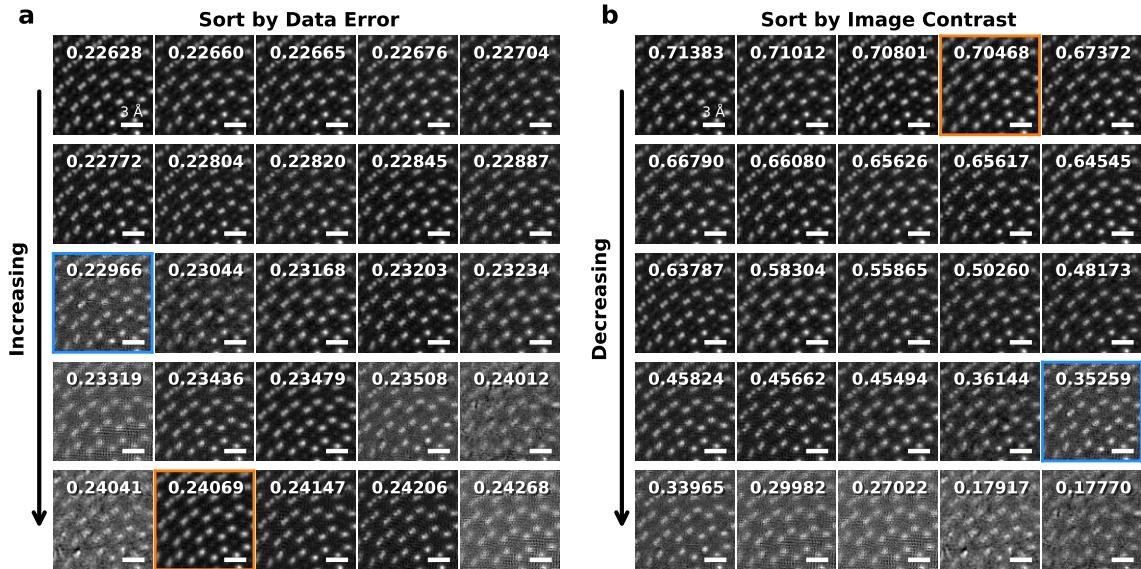
Supplementary Figure S10: Effect of sparsity regularization on ptychographic reconstruction of tBL-WSe<sub>2</sub>. (a-d) Reconstructed phase images with increasing sparsity weights of 0, 0.01, 0.03, and 0.1, respectively. Red insets show magnified regions highlighting visual details. (e-h) Corresponding FFT power spectra showing the reciprocal space information transfer increase with stronger sparsity regularization.



Supplementary Figure S11: Effect of  $k_z$  regularization on ptychographic reconstruction of simulated tBL-WSe<sub>2</sub> using PtychoShelves. (a) Ground truth atomic potential. (b–e) Reconstructed phase images with increasing  $k_z$  regularization parameters  $\beta$  of 0.01, 0.1, 0.5, and 1, respectively. (f) Line profiles extracted from the marked regions in (a–e), corresponding to the W-2Se-2Se-W atomic columns. The arrows indicate the 2Se atomic site for each  $k_z$  regularization value. As regularization increases, the local phase contrast becomes less accurate, highlighting the importance of selecting proper regularization parameters. 2Se corresponds to the atomic column with 2 overlapping Se atoms.



Supplementary Figure S12: Depth resolution analysis of multislice electron ptychography (MEP) reconstruction using the experimental tBL-WSe<sub>2</sub> dataset. The reconstruction was performed using PtyRAD with 12 probe modes, 20 object slices spaced by 1 Å, a batch size of 16, and regularization strategies including a positivity constraint, sparsity weight of 0.1, and an  $r_z$  filter with  $\sigma_z = 1$ , over 1000 iterations. (a) Phase slice at 8 Å depth from the reconstructed 3D object stack, showing a top-down view of the top WSe<sub>2</sub> layer. The blue dashed line highlights a row of isolated W atoms used for depth analysis. (b) Cross-sectional view along the dashed line in (a), visualizing the reconstructed depth structure. The W atoms are vertically elongated due to the limited depth resolution of MEP. (c) Depth profile of a single W atom at the location marked by orange arrows in (a) and (b), with a Gaussian fit yielding a full width at half maximum (FWHM) of 6.64 Å. (d) Histogram of fitted FWHM values from 34 W atoms across the field-of-view (FOV) in (a), resulting a mean depth resolution of 7.46 Å with a standard deviation of 0.37 Å.



Supplementary Figure S13: Comparison of different error metrics for hyperparameter optimization. Collection of 25 reconstructed phase images sorted by (a) data error calculated with diffraction patterns in reciprocal space, and (b) image contrast calculated with the phase image. The images in (a) reveal a weak correlation between visual reconstruction quality and data error, indicating that lower data error does not consistently correspond to better visual quality. The blue box marks a low quality image with low data error, while the orange box marks a high quality reconstruction with higher data error. In contrast, the two images are sorted correctly with image contrast, and (b) shows a much stronger association between image contrast and perceived quality, suggesting that image contrast may serve as a more reliable metric for evaluating reconstruction quality, and hence more suitable for hyperparameter optimization.

## References

- (1) Savitzky, B. H. et al. py4DSTEM: A Software Package for Four-Dimensional Scanning Transmission Electron Microscopy Data Analysis. *Microscopy and Microanalysis* **2021**, 27, 712–743.
- (2) Jiang, Y. fold slice. [https://github.com/yijiang1/fold\\_slice](https://github.com/yijiang1/fold_slice), 2020; [https://github.com/yijiang1/fold\\_slice](https://github.com/yijiang1/fold_slice).
- (3) Pennycook, T. J.; Lupini, A. R.; Yang, H.; Murfitt, M. F.; Jones, L.; Nellist, P. D. Efficient phase contrast imaging in STEM using a pixelated detector. Part 1: Experimental demonstration at atomic resolution. *Ultramicroscopy* **2015**, 151, 160–167.
- (4) Yang, H.; Pennycook, T. J.; Nellist, P. D. Efficient phase contrast imaging in STEM using a pixelated detector. Part II: Optimisation of imaging conditions. *Ultramicroscopy* **2015**, 151, 232–239.
- (5) Yang, H.; Rutte, R.; Jones, L.; Simson, M.; Sagawa, R.; Ryll, H.; Huth, M.; Pennycook, T.; Green, M.; Soltau, H., et al. Simultaneous atomic-resolution electron ptychography and Z-contrast imaging of light and heavy elements in complex nanostructures. *Nature Communications* **2016**, 7, 12532.
- (6) Enders, B.; Thibault, P. A computational framework for ptychographic reconstructions. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2016**, 472, 20160640.
- (7) Dong, Z.; Fang, Y.-L. L.; Huang, X.; Yan, H.; Ha, S.; Xu, W.; Chu, Y. S.; Campbell, S. I.; Lin, M. High-performance multi-mode ptychography reconstruction on distributed GPUs. 2018 New York Scientific Data Summit (NYSDS). 2018; pp 1–5.
- (8) Kandel, S.; Maddali, S.; Allain, M.; Hruszkewycz, S. O.; Jacobsen, C.; Nashed, Y. S. Us-

- ing automatic differentiation as a general framework for ptychographic reconstruction. *Optics express* **2019**, 27, 18653–18672.
- (9) Wakonig, K.; Stadler, H.-C.; Odstrčil, M.; Tsai, E. H. R.; Diaz, A.; Holler, M.; Usov, I.; Raabe, J.; Menzel, A.; Guizar-Sicairos, M. *PtychoShelves*, a versatile high-level framework for high-performance analysis of ptychographic data. *Journal of Applied Crystallography* **2020**, 53, 574–586.
- (10) Favre-Nicolin, V.; Girard, G.; Leake, S.; Carnis, J.; Chushkin, Y.; Kieffer, J.; Paleo, P.; Richard, M.-I. PyNX: high-performance computing toolkit for coherent X-ray imaging based on operators. *Journal of Applied Crystallography* **2020**, 53, 1404–1413.
- (11) Madsen, J.; Susi, T. The abTEM code: transmission electron microscopy from first principles. *Open Research Europe* **2021**, 1, 24.
- (12) Cherukara, M. J.; Zhou, T.; Nashed, Y.; Enfedaque, P.; Hexemer, A.; Harder, R. J.; Holt, M. V. AI-enabled high-resolution scanning coherent diffraction imaging. *Applied Physics Letters* **2020**, 117.
- (13) Yue, K.; Deng, J.; Jiang, Y.; Nashed, Y.; Vine, D. Ptychopy: GPU framework for ptychographic data analysis.
- (14) Weber, D.; Lesnichaia, A.; Strauch, A.; Clausen, A.; Bangun, A.; Melnyk, O.; Meissner, H.; Ehrig, S.; Wendt, R.; Sukumaran, M.; et al., Ptychography 4.0: 0.1.0. <https://zenodo.org/records/5055127>.
- (15) Du, M.; Kandel, S.; Deng, J.; Huang, X.; Demortiere, A.; Nguyen, T. T.; Tucoulou, R.; De Andrade, V.; Jin, Q.; Jacobsen, C. Adorym: A multi-platform generic X-ray image reconstruction framework based on automatic differentiation. *Optics express* **2021**, 29, 10000–10035.

- (16) Seifert, J.; Bouchet, D.; Loetgering, L.; Mosk, A. P. Efficient and flexible approach to ptychography using an optimization framework based on automatic differentiation. *OSA Continuum* **2021**, 4, 121–128.
- (17) Gursoy, D.; Ching, D. J. *Tike*; 2022.
- (18) Guzzi, F.; Kourousias, G.; Billè, F.; Pugliese, R.; Gianoncelli, A.; Carrato, S. A modular software framework for the design and implementation of ptychography algorithms. *PeerJ Computer Science* **2022**, 8, e1036.
- (19) Friedrich, T.; Yu, C.; Verbeeck, J.; Van Aert, S. Phase Object Reconstruction for 4D-STEM using Deep Learning,(4D-STEM Example Data). [URL https://doi.org/10.5281/zenodo.2022.7034879](https://doi.org/10.5281/zenodo.2022.7034879)
- (20) Chang, D. J.; O’Leary, C. M.; Su, C.; Jacobs, D. A.; Kahn, S.; Zettl, A.; Ciston, J.; Ercius, P.; Miao, J. Deep-learning electron diffractive imaging. *Physical review letters* **2023**, 130, 016101.
- (21) Loetgering, L.; Du, M.; Boonzajer Flaes, D.; Aidukas, T.; Wechsler, F.; Penagos Molina, D. S.; Rose, M.; Pelekanidis, A.; Eschen, W.; Hess, J., et al. PtyLab. m/py/jl: a cross-platform, open-source inverse modeling toolbox for conventional and Fourier ptychography. *Optics Express* **2023**, 31, 13763–13797.
- (22) Diederichs, B.; Herdegen, Z.; Strauch, A.; Filbir, F.; Müller-Caspary, K. Exact inversion of partially coherent dynamical electron scattering for picometric structure retrieval. *Nature Communications* **2024**, 15, 101.
- (23) Nakahata, R.; Zaman, S.; Zhang, M.; Lu, F.; Chiu, K. PtychoFormer: A Transformer-based Model for Ptychographic Phase Retrieval. [arXiv preprint arXiv:2410.17377](https://arxiv.org/abs/2410.17377) **2024**,
- (24) Zhang, H.; Li, G.; Zhang, J.; Zhang, D.; Chen, Z.; Liu, X.; Guo, P.; Zhu, Y.; Chen, C.;

- Liu, L.; Guo, X.; Han, Y. Three-Dimensional Inhomogeneity of Zeolite Structure and Composition Revealed by Electron Ptychography. Science **2023**, 380, 633–638.
- (25) Li, G.; Xu, M.; Tang, W.-Q.; Liu, Y.; Chen, C.; Zhang, D.; Liu, L.; Ning, S.; Zhang, H.; Gu, Z.-Y.; Lai, Z.; Muller, D. A.; Han, Y. Atomically Resolved Imaging of Radiation-Sensitive Metal-Organic Frameworks via Electron Ptychography. Nature Communications **2025**, 16, 914.
- (26) Nguyen, K. X.; Jiang, Y.; Lee, C.-H.; Kharel, P.; Zhang, Y.; van der Zande, A. M.; Huang, P. Y. Achieving sub-0.5-angstrom-resolution ptychography in an uncorrected electron microscope. Science **2024**, 383, 865–870.
- (27) Chen, Z.; Jiang, Y.; Shao, Y.-T.; Holtz, M. E.; Odstrčil, M.; Guizar-Sicairos, M.; Hanke, I.; Ganschow, S.; Schlom, D. G.; Muller, D. A. Electron ptychography achieves atomic-resolution limits set by lattice vibrations. Science **2021**, 372, 826–831.