doi:10.1017/S1431927622011217

Approaches Taken to Streamline and Consolidate Large Dataset Processing Techniques, with a Focus on Ptychography

Thomas C. Pekin[1*], Marcel Schloz[1], Pablo Fernandez Robledo[1], Anton Gladyshev[1], Sherjeel Shabih[1], Benedikt Haas[1], Christoph T. Koch[1]

[1]Humboldt Universität zu Berlin, Institut für Physik & IRIS, Adlershof, Berlin, Germany
[*] Corresponding author: tcpekin@physik.hu-berlin.de

In the past half decade, the relative explosion of fast pixelated detectors optimized for four-dimensional scanning transmission electron microscopy (4D-STEM) experiments has revolutionized the type and amount of data microscopists are able to obtain [1]. It has raised familiar questions across and between institutes, like "How is the data processed and stored? And what does one do with all these hard drives?"

As the amount of data that is acquired grows, the techniques that we use to deal with it must change. Ideally these techniques would be robust – they would include concepts of data history, or provenance [2, 3], and they would be applicable across a wide range of computational approaches, be that a single workstation or a large computing cluster. One way in which this is done is through the use of generalizable data analysis workflows (DAWs), which satisfy the prior requirements and provide a framework in which repeatable, large scale data processing can be performed. In this talk I will present the current and upcoming experimental, infrastructural and computational approaches we have used to develop an adaptable 4D-STEM workflow with these constraints in mind, with a focus on ptychography [4]. Figure 1 shows an overview of the approaches we have chosen and how they integrate with each other.

The first step in our pipeline is experimental data acquisition and is also where we must make our first choice. The choice is what type of data to acquire – full diffraction patterns, or data that is compressed in some manner, be that simply through binning or through a more complex process. Using compression at this step can save hundreds of gigabytes of data that needs to be saved, transferred and further processed [5-7]. This can also allow for many more diffraction patterns to be acquired before data transfer needs to occur, widening the field of view (FOV) or the number of 4D-STEM datasets acquired, for example, during an *in situ* experiment, while still allowing for a ptychographic reconstruction.

The data is then moved to a small server maintained by our group, which uses Slurm to manage jobs and allocate resources such as graphical processing units (GPUs) to jobs which need them [8]. On the server, the data can be remotely accessed by or shared with multiple people to process the data away from the microscope, and importantly, also remotely from their own personal computers, obviating the need for bulky and expensive personal workstations and cumbersome installation of dedicated software, whose primary design goal is not typically platform independence or ease of installation. This processing has usually consisted of a number of preprocessing steps on the data itself, followed by an iterative, gradient based ptychographic reconstruction algorithm (regularized optimization for ptychography – ROP [7]) to return the sample's phase. In order to streamline the preprocessing steps while increasing the traceability and reproducibility of our preprocessing pipeline, we have implemented Nextflow as our DAW framework [9]. Nextflow is used to manage the preprocessing steps, ensuring each piece of created data has a history and workflow associated with it, and also allows for portability across a variety of computational platforms (and supports Slurm, containers, etc.).

This leads us to the final step, iterative ptychographic reconstruction. As previously mentioned, we have traditionally used ROP, which is a derivative based ptychographic reconstruction algorithm written in-house in C++ and CUDA to take advantage of the inherent speed increases of a compiled language and the utilization of GPUs. In parallel, for rapid prototyping and testing new ideas, we additionally use the open-source Python code originally meant for X-rays, Automatic Differentiation-based Object Reconstruction with dYnaMical Scattering (ADORYM) [10]. This has proven very successful – it utilizes the same gradient based approach to reconstruction as ROP, it is written in Python using popular packages for automatic differentiation (Autograd and PyTorch are supported), can utilize (multiple) GPUs, is relatively simple to extend, in part due to the automatic calculation of the derivatives for the forward scattering process, and has excellent documentation for onboarding of new users.

In conclusion, we show here a number of design choices both at the microscope and away from it for the purpose of streamlining the processing of 4D-STEM data into a data workflow. These improvements allow us to reduce the amount of data acquired, safeguard and track how the data is processed, allow for multiple users to share computational resources, and finally, more easily onboard people who want to perform (and improve) ptychography.

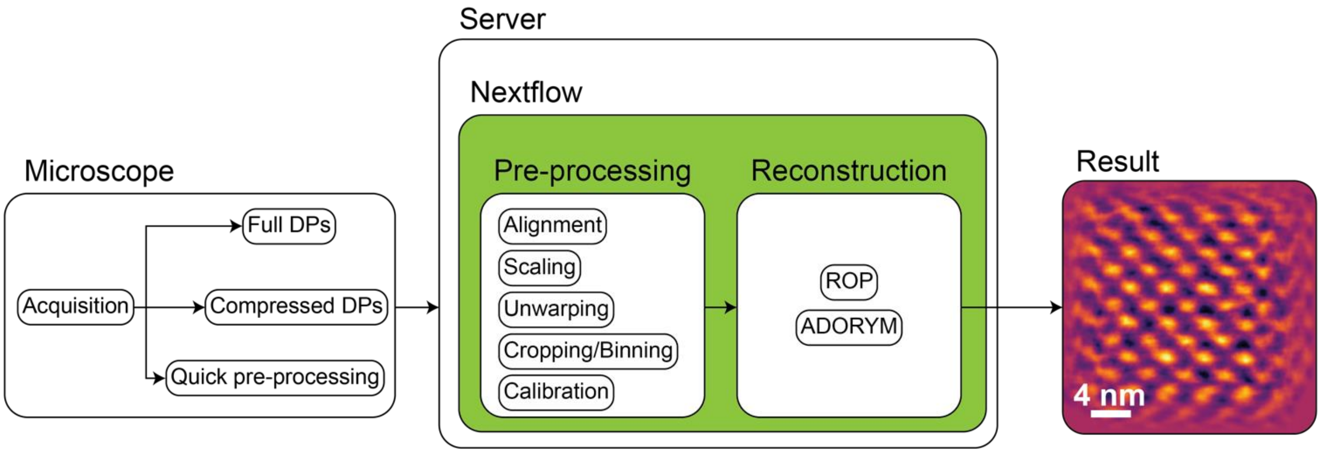

Figure 1. Overview of the 4D-STEM workflow as designed for ptychography. The first step shows the option of taking full or compressed 4D-STEM datasets, or doing some pre-processing at the microscope, followed by the movement of the data to a central server, and control of the DAW via Nextflow, in green. Nextflow allows for the tracing of workflow steps and metadata to ensure data provenance and processing repeatability, and can process the various steps asynchronously as computational resources become available.

References:
[1] C Ophus, Micros. Microanal. **25** (2019), p. 563. doi:10.1017/S1431927619000497
[2] YL Simmhan, B Plale and D Gannon, ACM SIGMOD Record **34** (2005), p. 31. doi:10.1145/1084805.1084812
[3] C Draxl and M Scheffler, MRS Bulletin **43** (2018), p. 676. doi:10.1557/mrs.2018.208
[4] MJ Humphry et al., Nat. Comm. **3** (2012), p.1. doi:10.1038/ncomms1733
[5] A Mittelberger et al., Microsc. Microanal. **27** (Suppl 1) (2021), p. 1064. doi:10.1017/S1431927621004013
[6] B Haas et al., Microsc. Microanal. **27** (Suppl 1) (2021), p. 994. doi:10.1017/S1431927621003779

https://doi.org/10.1017/S1431927622011217 Published online by Cambridge University Press

[7] M Schloz et al., Opt. Express **28** (2020), p. 28306. doi:10.1364/OE.396925
[8] AB Yoo, MA Jette, and M Grondona in “Job Scheduling Strategies for Parallel Processing”, ed. D Feitelson, L Rudolph, and U Schwiegelshohn, (Springer, Berlin) p.44.
[9] P Di Tommaso et al., Nat. Biotechnol. **35** (2017), p. 316. doi:10.1038/nbt.3820
[10] M Du et al., Opt. Express **29** (2021), p. 10000. doi:10.1364/OE.418296
[11] The authors acknowledge financial support from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-IDs 414984028 (SFB 1404), 182087777 (SFB 951), and 460197019 (FAIRmat).

https://doi.org/10.1017/S1431927622011217 Published online by Cambridge University Press