

## Letter



# Design of a graphical user interface for few-shot machine learning classification of electron microscopy data

Christina Doty<sup>a,1</sup>, Shaun Gallagher<sup>b,1</sup>, Wenqi Cui<sup>c,1</sup>, Wenya Chen<sup>d,1</sup>, Shweta Bhushan<sup>a,1</sup>, Marjolein Oostrom<sup>e</sup>, Sarah Akers<sup>e</sup>, Steven R. Spurgeon<sup>f,\*</sup>

<sup>a</sup> Department of Materials Science and Engineering, University of Washington, Seattle, Washington 98195, United States of America

<sup>b</sup> Department of Chemistry, University of Washington, Seattle, Washington 98195, United States of America

<sup>c</sup> Department of Electrical and Computer Engineering, University of Washington, Seattle, Washington 98195, United States of America

<sup>d</sup> Department of Chemical Engineering, University of Washington, Seattle, Washington 98195, United States of America

<sup>e</sup> National Security Directorate, Pacific Northwest National Laboratory, Richland, Washington 99352, United States of America

<sup>f</sup> Energy and Environment Directorate, Pacific Northwest National Laboratory, Richland, Washington 99352, United States of America

## ARTICLE INFO

## Keywords:

Transmission electron microscopy  
Machine learning  
Sparse data analytics  
Few-shot  
Segmentation  
Graphical user interface

## ABSTRACT

The recent growth in data volumes produced by modern electron microscopes requires rapid, scalable, and flexible approaches to image segmentation and analysis. Few-shot machine learning, which can richly classify images from a handful of user-provided examples, is a promising route to high-throughput analysis. However, current command-line implementations of such approaches can be slow and unintuitive to use, lacking the real-time feedback necessary to perform effective classification. Here we report on the development of a Python-based graphical user interface that enables end users to easily conduct and visualize the output of few-shot learning models. This interface is lightweight and can be hosted locally or on the web, providing the opportunity to reproducibly conduct, share, and crowd-source few-shot analyses.

## 1. Introduction

High-resolution characterization of materials using electron microscopy is an essential part of advancing technology in various fields, such as clean energy, catalysis, biomedicine, and quantum computing [1–5]. The techniques of scanning electron microscopy (SEM) and (scanning) transmission electron microscopy ((S)TEM) have provided fundamental new insight into materials structure and chemistry, with the latter capable of routine imaging and spectroscopy at the atomic scale [6]. (S)TEM in particular is widely used to study phase distributions, microstructures, and crystallographic defects in nanomaterials to build more accurate structure–property models [7]. Traditional methods of image analysis involve an expert user manually observing and analyzing individual images for specific microstructural features; not only is this approach tedious, it can also be error-prone, is often not reproducible, and is difficult to scale to large volumes of data [8,9]. With the introduction of high-speed detectors and spectrometers [10–13], today's instruments now generate data at scales (up to ~ 200 Tb/hr) far beyond the range of direct human comprehension, which has spurred efforts to integrate high-throughput data science tools into the microscopy workflow [14–17].

One of the most common and useful tasks in electron microscopy is that of semantic segmentation and phase identification [18]. In this task, the abundance and spatial location of relevant microstructural features or motifs are classified, with the goal to derive physically meaningful descriptors for a dataset [19,20]. While small datasets may be amenable to manual analysis, emerging automated systems [21,22] require approaches capable of analyzing tens to hundreds of thousands of images containing dense, noisy, or rare features of interest at high speed. Moreover, these approaches must be scalable to multiple imaging and spectroscopic modalities to harness the full potential of modern instrumentation and derive more unique structure–property solutions [14]. To address this challenge, a range of commercial [23, 24] and open-source [22,25–27] packages have been developed for full and semi-automated image analysis, particularly in the biological sciences [28,29]. While each has its strengths and weaknesses, these packages are often require extensive manual tuning, particularly in cases where limited prior labeled data is available.

Over the past several years, machine learning (ML)-based approaches have been developed for segmentation, classification, and other tasks in electron microscopy [15,30–37]. Put simply, these methods train a neural network to learn the mapping between the latent

\* Corresponding author.

E-mail address: [steven.spurgeon@pnl.gov](mailto:steven.spurgeon@pnl.gov) (S.R. Spurgeon).

<sup>1</sup> These authors contributed equally.

space representation of image features and their microstructural classification. However, most ML frameworks require a large support set of labeled images for training (e.g., containing  $> 100$  examples). Since labeling can require time-consuming manual identification and annotation of image features [38], preparing large training sets is typically not practical, especially when considering that feature types, imaging artifacts, and data quality can vary greatly from experiment to experiment [39,40].

Few-shot ML has been proposed as a solution to the problem of limited training data [41,42]. In this approach, a small, expert-selected training dataset is examined by a meta-learner that extracts a latent space prototype vector for each expert-provided label. These labels are based on the expert's *a priori* knowledge of the image or past ground truth examples. Classification is then performed by computing Euclidean distances to the prototype representations. Few-shot learning of TEM images has recently been successfully demonstrated for a variety of microstructures [43] and has been shown to efficiently generalize to different labeling tasks. However, the actual usage of the few-shot code requires multiple steps of image processing, support set selection, and few-shot classification. This process can be difficult for those not familiar with ML and programming, posing a barrier to real-world use.

Here we describe the design and implementation of a graphical user interface (GUI) for few-shot ML-based segmentation of electron microscopy data. We present an intuitive Python Flask-based application (pyChip) to upload image data, select desired features, perform classification, and visualize statistics on feature distributions. The user-friendly interface provides dynamic, real-time feedback that allows the expert to tune their analysis for different microstructural features, classification tasks, length scales, and image types. While we present TEM example data, we emphasize that this application can be extended to other types of imaging, such as SEM. In addition, the application is portable to both local and web-based hosts and can be easily updated with different back-end codes for few-shot learning. It is also lightweight and can be run on an average laptop computer, which will facilitate its adoption by the microscopy community.

## 2. Material and methods

The application integrates Python, D3, JavaScript, HTML/CSS, and Vega-lite with Flask, a Python web framework. Flask allows the back-end python scripts to easily pass information to the front-end interactive visualization [44].

The front-end interactive visualization was created with JavaScript and HTML/CSS. JavaScript can manipulate Document Object Model (DOM) objects such as paragraphs and images [45]. JavaScript was used to allow users to (1) view the image chips, (2) modify the names of support sets, (3) assign image chips to a specified support set, and (4) view the current collection of selected chips for each support set. Parts of the JavaScript code utilized D3, an open-source JavaScript library capable of building customized visualization from code [46]. D3 uses data as input to create visualizations with Scalable Vector Graphics (SVG) elements such as lines, rectangles and circles. In this application, D3 was used to dynamically display grid lines over the user's image based on a user-inputted grid-size. The code for the grid lines was adapted from code written by user "Chuck Grimm" on [bl.ock](#) [47]. We also used Vega-lite, a JSON syntax to display the results of the few-shot model in the form of a bar chart [48].

The back-end scripts are written in Python. The first function chips the images into squares matching the user-selected grid size. The second function implements the few-shot ML model with the selected support sets as input. The few-shot function utilizes a pre-trained ResNet available from PyTorch.

The Flask Framework allows the inputs from the front-end user interaction to be passed as input to the Python scripts on the back-end. The user selected grid-size on the front-end is used as an input for the image chipping Python function on the back-end. Likewise, the support

set selected on the front-end is used as an input for the few-shot Python functions on the back-end. In turn, outputs from the back-end Python scripts, such as the image chips and the few-shot model's results are displayed in the front-end web page. The Flask framework mediates these front-end and back-end interactions.

## 3. Results and discussion

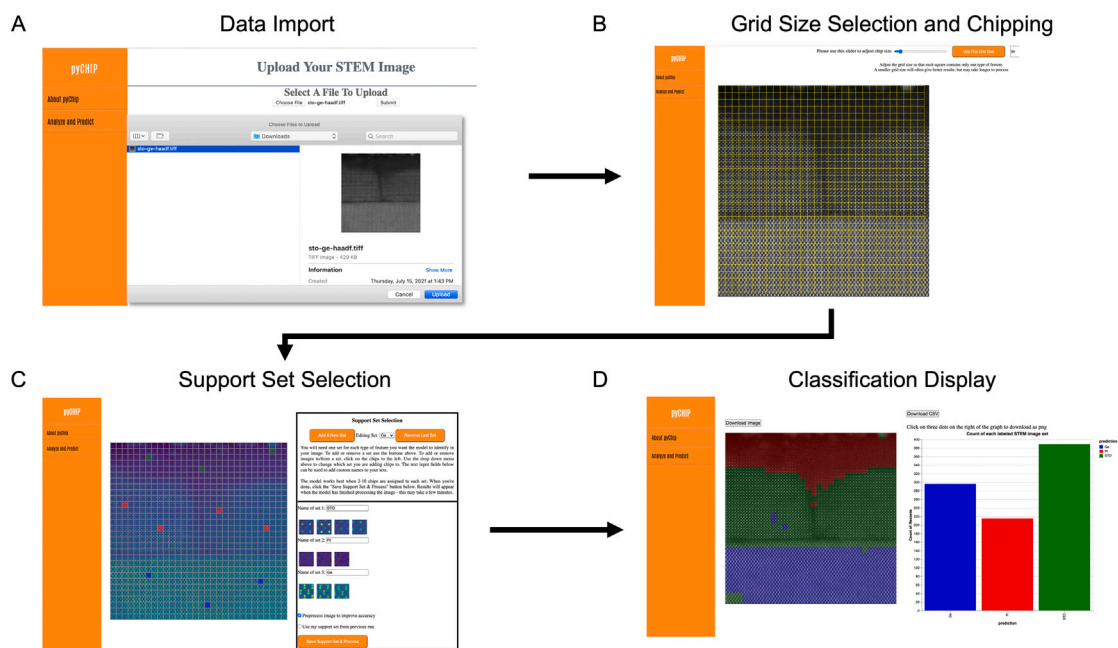
An overview of the pyChip application workflow is shown in Fig. 1. The pyCHIP user interface contains a homepage and components for image data import, grid selection, support set selection, feature identification, and classification display. Each component is assigned a respective HTML page, serviced by a unique route. The tab on the left is used for navigation. The following sections describe the steps in the analysis workflow and practical considerations for their use. These sections are followed by selected examples of the analysis of different material microstructures. Finally, we conclude with an outlook on potential future developments and the application's usage in emerging automated analysis workflows.

### 3.1. Data import

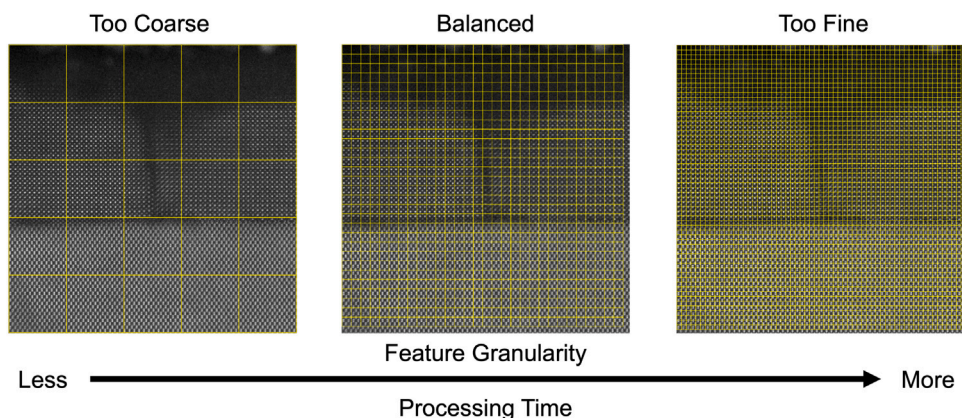
As shown in Fig. 1.A, the pyCHIP application initially presents the user with an upload prompt to select images for analysis. Currently allowed data formats are `jpg`, `png`, `tiff`, and `dm4`. The first three are common image formats, while the last is a proprietary container format used by the Gatan Microscopy Suite [49], an industry-standard TEM analysis application. Images should be imported without any visible scale bar or other annotation. The `dm4` format is loaded using a library developed by the National Center for Electron Microscopy [50] and pixel values are created by scaling the embedded image data to a range between 0 and 255. If no image is uploaded before submitting, an error message indicates that no appropriate file was selected. While in principle any image resolution can be used, the processing time does increase significantly with larger resolution. For best performance, it is recommended to downsample large images to resolution of  $512 \times 512$  or  $1024 \times 1024$  pixels. However, it is important to avoid aliasing artifacts, which can occur in high-resolution micrographs, particularly those containing lattice features.

### 3.2. Grid size selection and chipping

After the selected data has been imported, the user must perform the important task of grid size selection and image chipping, as shown in Fig. 1.B. The user selects an appropriate sampling grid relative to their features of interest, which is used to chip the image in the backend few-shot code. To aid in this process, the user can drag a slider bar to dynamically adjust a superimposed grid atop their image. An ideal support set for the few-shot pyCHIP model is built from a grid with each square containing a unique microstructural feature or motif. The range of allowed grid sizes is dynamic, scaling from a minimum 50 pixels up to the dimensions of the uploaded image. The default lower limit is set to 50 pixels for visibility, but can be as small as the single pixel level. Too coarse a grid size will not capture relevant features with sufficient fidelity, while too fine a grid will greatly increase processing time, so the user should test different sizes to strike an optimal balance, as shown in Fig. 2. Once an appropriate size is determined, the user clicks the "Use This Grid Size" button and the desired grid size is passed to a function that crops the edges of the full size image to the total mesh, sub-divides the image, and saves the resulting collection of support set "chips." Both the raw chips and support sets are saved for future recall.



**Fig. 1.** (A–D) Overview of the pyChip application workflow for few-shot classification of electron microscopy data, consisting of data import, grid size selection and chipping, support set selection, and classification display, respectively.



**Fig. 2.** Grid size selection relative to atomic-scale motifs for effective support set selection, illustrating tradeoff in processing time vs. feature granularity.

### 3.3. Support set selection

After chipping, the user must assign support set examples as features of interest, as shown in Fig. 1.C. Building on the concept of the ubiquitous CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) security feature [51] on websites, we have designed an intuitive interface for labeling. The chips created in the previous step now appear on the dashboard as interactive objects, as shown in detail in Fig. 3. The user is provided with an empty starting support set in a drop-down menu, which can be renamed. This menu controls the specific support set(s) that is/are available for modification. Any image chips that the user clicks on will be added to the currently staged support set and appear in the right panel. These selected chips will appear highlighted in colors based on the user-designated support set. In cases where the user clicks the wrong chip and would like to remove it from the support set, they may click the highlighted chip again. Additional support sets can be added or removed by clicking the appropriate buttons at the top of the panel.

In practice, the user is advised to assign a minimum of 2 chips or maximum of 10 chips to each labeled set [43]. These chips should be selected from different parts of the image to capture some of the

variability in the data. Since the selected chips will become the support set for the few-shot ML model, all suspected feature types should be selected—sets can easily be adjusted by reprocessing. After the chip selection and labeling, the user may click “Save Support Set & Process” to update the training set and begin training the model. The trained model will then predict features against the selected support sets.

### 3.4. Classification display

After training the model using the selected chips, the model predicts the support set label for all the chips in the original image and returns a colored segmented image, as shown in the left of Fig. 1.D. The right side presents the count for each user-selected support set within the image. The segmented image and associated statistics tell the user critical information regarding features in the image. The user can export the colored image (png format), the bar chart (png or svg formats), and the raw data for the bar chart (csv format).

### 3.5. Usage examples

Here we discuss two specific material use cases to illustrate the process of support set selection and corresponding outputs. Nanomaterials

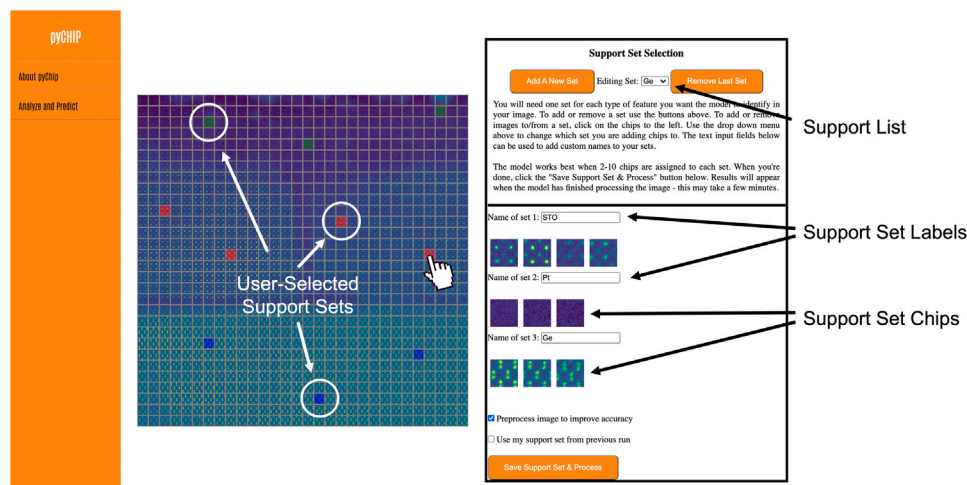


Fig. 3. Interactive support set selection and labeling of user-defined features of interest.

are a natural subject for study via TEM, since they exhibit properties distinct from bulk materials that are strongly governed by atomic-to-nanoscale structure and defects. Quantifying the distribution of these features, such as the abundance of particular phases, defects, and morphology, is a common but challenging and time-consuming task [15, 52]. We consider two very different examples: a cross-sectional thin film heterostructure of SrTiO<sub>3</sub> (STO) / Ge and nanoparticles of MoO<sub>3</sub>. The former sample represents a potential material for next-generation electronics [53,54] and contains features, such as interfaces and crystal motifs, that are commonly encountered in atomic-scale images (> 6 Mx magnification). The latter sample is an important precursor for organic photovoltaic synthesis [55] and takes the form of a particle dispersion with varying morphology that is evident at low magnification (30 – 80 kx). This example is particularly relevant, since many important systems manifest in nanoparticle form. As shown Fig. 4.A, the atomic-resolution STO/Ge image is divided into square chips, each 99 pixels wide relative to a total image width of 2970 pixels. By selecting just three or fewer examples from each region of interest (STO, Ge, and vacuum), the user is able to quickly perform a segmentation that traces the boundaries of the interface and the sample surface. Moreover, the user can generate statistics on the abundance (pixel fraction) of each feature type. In the case of the lower-magnification MoO<sub>3</sub> nanoparticle sample, shown in Fig. 4.B, the image is divided into square chips, each 18 pixels wide relative to a total image width of 504 pixels. The user provides three examples each of two different particle morphologies and the carbon background, to quickly distinguish features and produce statistics. It is important to note that both the labeling task and image type can be easily changed *without* time-consuming retraining of the network, which is a strength of the few-shot approach [43]. The pyCHIP application provides rapid feedback, so the user can adjust the grid size and select different support sets if necessary. This approach enables both single image analysis and systematic comparisons of multiple images of similar type.

#### 4. Conclusions

We have described the design and implementation of a flexible, multi-platform GUI application for data import, support set selection, classification, and visualization of a few-shot ML code. This interface will aid in development of rapid, reproducible, and scalable analyses of electron microscopy images. We envision that this tool will eventually become part of a standard workflow for data classification, particularly in cases where there are limited prior labeled examples.

Moving forward, it will be desirable to implement both usability and model improvements; these include the implementation of user

accounts and recall of images, parameters, and support sets. In addition, more automatic guidance on chipping and support set selection is necessary, as well as recommendations for appropriate parameters. Although the colorized segmentation and bar graph provide a clear summary of the location and abundance of each feature, future developments should include more informative statistics, such as error bars or other confidence measures. While this task alone is a significant challenge and ongoing area of research in ML, such information will provide the user with more complete, insightful semantics for their analysis.

Finally, it will be important to expand the application to provide detailed management of metadata and curation of the labeling process, particularly for multimodal datasets [56]. These procedures can eventually be automated, enabling more repeatable image analysis and leading to improved reproducibility in materials characterization [8]. Batch processing is also needed for the analysis of time-series data, but this will require further optimization of the few-shot code and graphics processing unit (GPU) acceleration for practical implementation. We envision eventual on-the-fly integration of the few-shot approach to conduct analyses and track the dynamic evolution of systems in real-time, which will be a key step toward automated microscopy.

#### Code availability

The GUI code used in this study is available on Github at [https://github.com/pnnl/pychip\\_gui](https://github.com/pnnl/pychip_gui). The core few-shot codebase is proprietary, but the Prototypical Network code is available on Github at <https://github.com/jakesnell/prototypical-networks>.

#### CRediT authorship contribution statement

**Christina Doty:** Conceptualization, Methodology, Software, Writing – original draft. **Shaun Gallagher:** Conceptualization, Methodology, Software, Writing – original draft. **Wenqi Cui:** Conceptualization, Methodology, Software, Writing – original draft. **Wenya Chen:** Conceptualization, Methodology, Software, Writing – original draft. **Shweta Bhushan:** Conceptualization, Methodology, Software, Writing – original draft. **Marjolein Oostrom:** Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing, Supervision. **Sarah Akers:** Conceptualization, Methodology, Software, Writing – review & editing, Supervision. **Steven R. Spurgeon:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

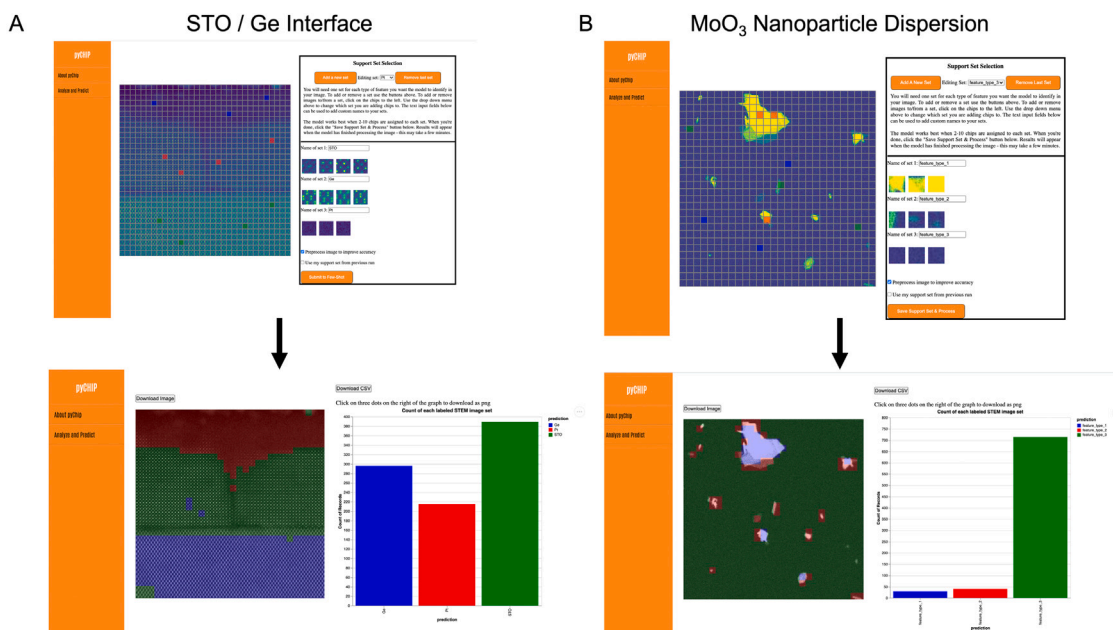


Fig. 4. (A–B) Support set selection and output visualization for a cross-sectional STO/Ge interface and a dispersion of MoO<sub>3</sub> nanoparticles on a carbon support, respectively.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The authors would like to thank Drs. Jenna Pope, Elizabeth Kautz, and Matthew Olszta for reviewing the manuscript. M.O., S.A. and S.R.S. were supported by the I3T Commercialization Laboratory Directed Research and Development (LDRD) program at Pacific Northwest National Laboratory (PNNL). PNNL is a multiprogram national laboratory operated for the U.S. Department of Energy (DOE) by Battelle Memorial Institute under Contract No. DE-AC05-76RL0-1830. Experimental sample preparation was performed at the Environmental Molecular Sciences Laboratory (EMSL), a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at PNNL. TEM data was collected in the Radiological Microscopy Suite (RMS), located in the Radiochemical Processing Laboratory (RPL) at PNNL. C.D., S.G., W.C., W.Q. and S.B. acknowledge support from the University of Washington Clean Energy Institute and the National Science Foundation Research Traineeship under Award NSF DGE-1633216.

## References

- [1] Raghav S. Rai, Swaminathan Subramanian, Role of transmission electron microscopy in the semiconductor industry for process development and failure analysis, *Prog. Cryst. Growth Charact. Mater.* 55 (3–4) (2009) 63–97, <http://dx.doi.org/10.1016/j.pcrysgrow.2009.09.002>.
- [2] Chao Zhang, Konstantin L. Firestein, Joseph F.S. Fernando, Dumindu Siriwardena, Joel E. Treifeldt, Dmitri Golberg, Recent progress of In Situ transmission electron microscopy for energy materials, *Adv. Mater.* 1904094 (2019) 1904094, <http://dx.doi.org/10.1002/adma.201904094>, URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201904094>.
- [3] Stephen J. Pennycook, The impact of STEM aberration correction on materials science, *Ultramicroscopy* 180 (2017) 22–33, <http://dx.doi.org/10.1016/j.ultramic.2017.03.020>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304399117301213>.
- [4] Furqan A. Shah, Krisztina Ruscsák, Anders Palmquist, 50 Years of scanning electron microscopy of bone—a comprehensive overview of the important discoveries made and insights gained into bone material properties in health, disease, and taphonomy, *Bone Res.* 7 (1) (2019) 1–15, <http://dx.doi.org/10.1038/s41413-019-0053-z>.
- [5] S.J. Pennycook, M. Varela, C.J.D. Hetherington, A.I. Kirkland, Materials advances through aberration-corrected electron microscopy, *MRS Bull.* 31 (01) (2006) 36–43, <http://dx.doi.org/10.1557/mrs2006.4>, URL: <http://www.journals.cambridge.org/abstract/S0883769400009519>.
- [6] Ian MacLaren, Quentin M. Ramasse, Aberration-corrected scanning transmission electron microscopy for atomic-resolution studies of functional oxides, *Int. Mater. Rev.* 59 (3) (2014) 115–131, <http://dx.doi.org/10.1179/1743280413Y.0000000026>, URL: <http://www.maneyonline.com/doi/abs/10.1179/1743280413Y.0000000026>.
- [7] David T. Fullwood, Stephen R. Niezgoda, Brent L. Adams, Surya R. Kalidindi, Microstructure sensitive design for performance optimization, *Prog. Mater. Sci.* 55 (6) (2010) 477–562, <http://dx.doi.org/10.1016/j.pmatsci.2009.08.002>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0079642509000760>.
- [8] Alexander Tropsha, Karmann C. Mills, Anthony J. Hickey, Reproducibility, sharing and progress in nanomaterial databases, *Nat. Nanotechnol.* 12 (12) (2017) 1111–1114, <http://dx.doi.org/10.1038/nnano.2017.233>, URL: <http://www.nature.com/articles/nnano.2017.233>.
- [9] Anne L. Plant, Laurie E. Locascio, Willie E. May, Patrick D. Gallagher, Improved reproducibility by assuring confidence in measurements in biomedical research, *Nature Methods* 11 (9) (2014) 895–898, <http://dx.doi.org/10.1038/nmeth.3076>, URL: <http://www.nature.com/articles/nmeth.3076>.
- [10] Mark W. Tate, Prafull Purohit, Darol Chamberlain, Kayla X. Nguyen, Robert Hovden, Celesta S. Chang, Pratiti Deb, Emrah Turgut, John T. Heron, Darrell G. Schlom, Daniel C. Ralph, Gregory D. Fuchs, Katherine S. Shanks, Hugh T. Philipp, David A. Muller, Sol M. Gruner, High dynamic range pixel array detector for scanning transmission electron microscopy, *Microsc. Microanal.* 22 (1) (2016) 237–249, <http://dx.doi.org/10.1017/S1431927615015664>, URL: [https://www.cambridge.org/core/product/identifier/S1431927615015664/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S1431927615015664/type/journal_article).
- [11] J.A. Mir, R. Clough, R. MacInnes, C. Gough, R. Plackett, I. Shipsey, H. Sawada, I. MacLaren, R. Ballabriga, D. Maneuski, V. O'Shea, D. McGrouther, A.I. Kirkland, Characterisation of the Medipix3 detector for 60 and 80 keV electrons, *Ultramicroscopy* 182 (2017) 44–53, <http://dx.doi.org/10.1016/j.ultramic.2017.06.010>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304399116303989>.
- [12] Benjamin Plotkin-Swing, George J. Corbin, Sacha De Carlo, Niklas Dellby, Christoph Hoernemann, Matthew V. Hoffman, Tracy C. Lovejoy, Chris E. Meyer, Andreas Mittelberger, Radosav Pantelic, Luca Piazza, Ondrej L. Krivanek, Hybrid pixel direct detector for electron energy loss spectroscopy, *Ultramicroscopy* (2020) 113067, <http://dx.doi.org/10.1016/j.ultramic.2020.113067>, <https://linkinghub.elsevier.com/retrieve/pii/S0304399120302187>.
- [13] Ian MacLaren, Thomas A. Macgregor, Christopher S. Allen, Angus I. Kirkland, Detectors—The ongoing revolution in scanning transmission electron microscopy and why this important to material characterization, *APL Mater.* 8 (11) (2020) 110901, <http://dx.doi.org/10.1063/5.0026992>, URL: <http://aip.scitation.org/doi/10.1063/5.0026992>.
- [14] Steven R. Spurgeon, Colin Ophus, Lewys Jones, Amanda Petford-Long, Sergei V. Kalinin, Matthew J. Olszta, Rafael E. Dunin-Borkowski, Norman Salmon, Khalid Hattar, Wei chang D Yang, Renu Sharma, Yingge Du, Ann Chiamaronti, Haimei Zheng, Edgar C. Buck, Libor Kovarik, R. Lee Penn, Dongsheng Li, Xin Zhang,

- Mitsuhiro Murayama, Mitra L. Taheri, Towards data-driven next-generation transmission electron microscopy, *Nature Mater.* 20 (3) (2021) 274–279, <http://dx.doi.org/10.1038/s41563-020-00833-z>, URL: <http://www.nature.com/articles/s41563-020-00833-z>.
- [15] Jeffrey M. Ede., Deep learning in electron microscopy, *Mach. Learn. Sci. Technol.* 2 (1) (2021) 011004, <http://dx.doi.org/10.1088/2632-2153/abd614>, URL: <https://iopscience.iop.org/article/10.1088/2632-2153/abd614>, arXiv:2009.08328.
- [16] Sergei V. Kalinin, Andrew R. Lupini, Ondrej Dyck, Stephen Jesse, Maxim Ziatdinov, Rama K. Vasudevan, Lab on a beam—Big data and artificial intelligence in scanning transmission electron microscopy, *MRS Bull.* 44 (7) (2019) 565–575, <http://dx.doi.org/10.1557/mrs.2019.159>, URL: [https://www.cambridge.org/core/product/identifier/S0883769419001593/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0883769419001593/type/journal_article).
- [17] Alex Belianinov, Rama Vasudevan, Evgheni Strelcov, Chad Steed, Sang Mo Yang, Alexander Tselev, Stephen Jesse, Michael Biegalski, Galen Shipman, Christopher Symons, Albina Borisevich, Rick Archibald, Sergei Kalinin, Big data and deep data in scanning and electron microscopies: deriving functionality from multidimensional data sets, *Adv. Struct. Chem. Imaging* 1 (1) (2015) 6, <http://dx.doi.org/10.1186/s40679-015-0006-6>, URL: <http://www.ascimaging.com/content/1/1/6>.
- [18] Brian L. DeCost, Bo Lei, Toby Francis, Elizabeth A. Holm, High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel, *Microsc. Microanal.* 25 (1) (2019) 21–29, <http://dx.doi.org/10.1017/S1431927618015635>, arXiv:1805.08693.
- [19] Lukas Vlcek, Artem Makhov, Minghuo Pan, Rama K. Vasudevan, Sergei V. Kalinin, Knowledge extraction from atomically resolved images, *ACS Nano* 11 (10) (2017) 10313–10320, <http://dx.doi.org/10.1021/acsnano.7b05036>, URL: <http://pubs.acs.org/doi/10.1021/acsnano.7b05036>.
- [20] Rama K. Vasudevan, Maxim Ziatdinov, Stephen Jesse, Sergei V. Kalinin, Phases and interfaces from real space atomically resolved data: Physics-based deep data image analysis, *Nano Lett.* 16 (9) (2016) 5574–5581, <http://dx.doi.org/10.1021/acsnanolett.6b02130>, URL: <http://pubs.acs.org/doi/abs/10.1021/acsnanolett.6b02130>.
- [21] Matthew. Olszta, Derek. Hopkins, Kevin R. Fiedler, Marjolein. Oostrom, Sarah. Akers, Steven R. Spurgeon, An automated scanning transmission electron microscope guided by sparse data analytics, 2021, URL: arXiv:2109.14772.
- [22] Martin Schorb, Isabella Haberbosch, Wim J.H. Hagen, Yannick Schwab, David N. Mastrorade., Software tools for automated transmission electron microscopy, *Nature Methods* 16 (6) (2019) 471–477, <http://dx.doi.org/10.1038/s41592-019-0396-9>.
- [23] MIPAR, MIPAR. URL: <https://www.mipar.us>.
- [24] TFAvizo, Thermo Fisher Avizo. URL: <https://www.thermofisher.com/us/en/home/industrial/electron-microscopy/electron-microscopy-instruments-workflow-solutions/3d-visualization-analysis-software.html>.
- [25] CellProfiler, CellProfiler. URL: <https://cellprofiler.org>.
- [26] ImageJ, ImageJ. URL: <https://imagej.nih.gov/ij/>.
- [27] Martin Schorb, Py-EM, 2021, URL: <https://git.embl.de/schorb/pyem>.
- [28] Ignacio Arganda-Carreras, Verena Kaynig, Curtis Rueden, Johannes Schindelin Kevin W Eliceiri, Albert Cardona, H Sebastian Seung, Trainable weka segmentation: a machine learning tool for microscopy pixel classification, *Bioinformatics* 33 (15) (2017) 2424–2426, <http://dx.doi.org/10.1093/bioinformatics/btx180>, URL: <https://academic.oup.com/bioinformatics/article/33/15/2424/3092362>.
- [29] Dan Ciresan, Alessandro Giusti, Luca Gambardella, Jürgen Schmidhuber, Deep neural networks segment neuronal membranes in electron microscopy images, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), *Adv. Neural Inf. Process. Syst.*, Vol. 25, Curran Associates, Inc., 2012, URL: <https://proceedings.neurips.cc/paper/2012/file/459a4ddb586f24efd9395aa7662bc7c-Paper.pdf>.
- [30] Afshin. Khadangi, Thomas. Boudier, Vijay. Rajagopal, EM-net: Deep learning for electron microscopy image segmentation, in: 2020 25th Int. Conf. Pattern Recognit., IEEE, 2021, pp. 31–38, <http://dx.doi.org/10.1109/ICPR48806.2021.9413098>, URL: <https://ieeexplore.ieee.org/document/9413098/>.
- [31] M. Ge, F. Su, Z. Zhao, D. Su, Deep learning analysis on microscopic imaging in materials science, *Mater. Today Nano* 11 (2020) 100087, <http://dx.doi.org/10.1016/j.mtnano.2020.100087>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S258884202030016X>.
- [32] J.A. Aguiar, M.L. Gong, R.R. Unocic, T. Tasdizen, B.D. Miller, Decoding crystallography from high-resolution electron imaging and diffraction datasets with deep learning, *Sci. Adv.* 5 (10) (2019) eaaw1949, <http://dx.doi.org/10.1126/sciadv.aaw1949>, URL: <http://advances.sciencemag.org/lookup/doi/10.1126/sciadv.aaw1949>.
- [33] R. Kannan, A.V. Ievlev, N. Laanait, M.A. Ziatdinov, R.K. Vasudevan, S. Jesse, S.V. Kalinin, Deep data analysis via physically constrained linear unmixing: universal framework, domain examples, and a community-wide platform, *Adv. Struct. Chem. Imaging* 4 (1) (2018) 6, <http://dx.doi.org/10.1186/s40679-018-0055-8>, URL: <https://ascimaging.springeropen.com/articles/10.1186/s40679-018-0055-8>.
- [34] Rama K. Vasudevana, Nouamane Laanait, Erik M. Ferraguta, Kai Wang, David B. Geohegan, Kai Xiao, Maxim Ziatdinov, Stephen Jesse, Ondrej Dyck, Sergei V. Kalinin, Mapping mesoscopic phase evolution during E-beam induced transformations via deep learning of atomically resolved images, *Npj Comput. Mater.* 4 (1) (2018) 30, <http://dx.doi.org/10.1038/s41524-018-0086-7>, URL: <http://www.nature.com/articles/s41524-018-0086-7>.
- [35] Maxim Ziatdinov, Artem Makhov, Sergei V. Kalinin, Learning surface molecular structures via machine vision, *Npj Comput. Mater.* 3 (1) (2017) 31, <http://dx.doi.org/10.1038/s41524-017-0038-7>, URL: <http://www.nature.com/articles/s41524-017-0038-7>.
- [36] Paul M. Voyles, Informatics and data science in materials microscopy, *Curr. Opin. Solid State Mater. Sci.* 21 (3) (2017) 141–158, <http://dx.doi.org/10.1016/j.cossms.2016.10.001>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S135902861630095X>.
- [37] Brian L. DeCost, Elizabeth A. Holm, A computer vision approach for automated analysis and classification of microstructural image data, *Comput. Mater. Sci.* 110 (2015) 126–133, <http://dx.doi.org/10.1016/j.commatsci.2015.08.011>, URL: <https://linkinghub.elsevier.com/retrieve/pii/S0927025615005066>.
- [38] Deniz Iren, Marc Ackermann, Julian Gorfer, Gaurav Pujar, Sebastian Weselmecking, Ulrich Krupp, Stefano Bromuri, Aachen-heerlen annotated steel microstructure dataset, *Sci. Data* 8 (1) (2021) 140, <http://dx.doi.org/10.1038/s41597-021-00926-7>, URL: <http://www.nature.com/articles/s41597-021-00926-7>.
- [39] Khalid Hattar, Katherine L. Jungjohann, Possibility of an integrated transmission electron microscope: enabling complex in-situ experiments, *J. Mater. Sci.* 56 (9) (2021) 5309–5320, <http://dx.doi.org/10.1007/s10853-020-05598-z>.
- [40] Mitra L. Taheri, Eric A. Stach, Ilke Arslan, P.A. Crozier, Bernd C. Kabius, Thomas LaGrange, Andrew M. Minor, Seiji Takeda, Mihaela Tanase, Jakob B. Wagner, Renu Sharma, Current status and future directions for in situ transmission electron microscopy, *Ultramicroscopy* 170 (2016) 86–95, <http://dx.doi.org/10.1016/j.ultramicro.2016.08.007>.
- [41] Nathan Hilliard, Lawrence Phillips, Scott Howland, Artëm Yankov, Courtney D. Corley, Nathan O. Hodas, Few-Shot Learning with Metric-Agnostic Conditional Embeddings, Feb 2018, URL: <http://arxiv.org/abs/1802.04376>.
- [42] Chelsea Finn, Pieter Abbeel, Sergey Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: 34th Int. Conf. Mach. Learn., Vol. 3, ICML 2017, 2017, pp. 1856–1868, URL: arXiv:1703.03400.
- [43] Sarah Akers, Elizabeth Kautz, Andrea Trevino-Gavito, Matthew Olszta, Bethany E. Matthews, Le Wang, Yingge Du, Steven R. Spurgeon, Rapid and flexible segmentation of electron microscopy data using few-shot machine learning, *npj Comput. Mater.* 7 (1) (2021) 187, <http://dx.doi.org/10.1038/s41524-021-00652-z>.
- [44] Miguel Grinberg, Flask Web Development: Developing Web Applications with Python, O'Reilly Media, Inc., 2018.
- [45] David. Flanagan, Gregor.M. Novak, JavaScript: The Definitive Guide, 1998.
- [46] Michael Bostock, Vadim. Ogievetsky, Jeffrey Heer, D<sup>3</sup> data-driven documents, *IEEE Trans. Vis. Comput. Graphics* 17 (12) (2011) 2301–2309.
- [47] Chuck Grimmer, Let's make a grid with D3.js, URL: <https://bl.ocks.org/cagrimmer/07f8c8daea00946b9e704e3efcbcd5739>.
- [48] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, Jeffrey Heer, Vega-lite: A grammar of interactive graphics, *IEEE Trans. Vis. Comput. Graphics* 23 (1) (2016) 341–350.
- [49] Gatan, Gatan Microscopy Suite, URL: <https://www.gatan.com/products/tem-analysis/gatan-microscopy-suite-software>.
- [50] NCEM, NCEM, URL: <https://github.com/ercius/openNCEM>.
- [51] Luis von Ahn, Manuel Blum, John Langford, Telling humans and computers apart automatically, *Commun. ACM* 47 (2) (2004) 56–60, <http://dx.doi.org/10.1145/966389.966390>, URL: <https://dl.acm.org/doi/10.1145/966389.966390>.
- [52] Karl Pazdernik, Nicole L. LaHaye, Conor M. Artman, Yuanyuan Zhu, Microstructural classification of unirradiated LiAlO<sub>2</sub> pellets by deep learning methods, *Comput. Mater. Sci.* 181 (May) (2020) 109728, <http://dx.doi.org/10.1016/j.commatsci.2020.109728>.
- [53] Scott A Chambers, Yingge Du, Ryan B Comes, Steven R Spurgeon, Peter V Sushko, The effects of core-level broadening in determining band alignment at the epitaxial SrTiO<sub>3</sub> (001)/ p-Ge (001) heterojunction, *Appl. Phys. Lett.* 110 (2017) 082104, <http://dx.doi.org/10.1063/1.4977422>.
- [54] Mantu K. Hudait, Michael Clavel, Yan Zhu, Patrick S. Goley, Souvik Kundu, Deepam Maurya, Shashank Priya, Integration of SrTiO<sub>3</sub> on crystallographically oriented epitaxial germanium for low-power device applications, *ACS Appl. Mater. Interfaces* 7 (9) (2015) 5471–5479, <http://dx.doi.org/10.1021/am5091726>, URL: <http://pubs.acs.org/doi/abs/10.1021/am5091726>.
- [55] Yongshuai Gong, Yiman Dong, Biao Zhao, Runnan Yu, Siqian Hu, Zhan'ao Tan, Diverse applications of MoO<sub>3</sub> for high performance organic photovoltaics: fundamentals, processes and optimization strategies, *J. Mater. Chem. A* 8 (3) (2020) 978–1009, <http://dx.doi.org/10.1039/C9TA12005J>, URL: <http://xlink.rsc.org/?DOI=C9TA12005J>.
- [56] Philip.J. Withers, Timothy.L. Burnett, Rich multi-dimensional correlative imaging, *IOP Conf. Ser. Mater. Sci. Eng.* 580 (2019) 012014, <http://dx.doi.org/10.1088/1757-899X/580/1/012014>, URL: <https://iopscience.iop.org/article/10.1088/1757-899X/580/1/012014>.