



Dynamic sampling for scanned microscopy modalities

Zachary Varley, Gregory S. Rohrer, Marc De Graef*

Department of Materials Science and Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Keywords:

Image sampling
Variance
electron microscopy
Image scanning

ABSTRACT

We present an unsupervised heuristic-based batched dynamic algorithm for pixel-wise image sampling in a variety of scanned imaging modalities. Our algorithm offers improved performance over static Sobol sampling at a low computational cost. We evaluate the algorithm using three datasets to highlight potential savings in both two and three dimensional characterization contexts. Significantly, the implementation speed lends the approach runtime compatibility with electron microscopy. Reconstructed grain shape distributions for the sampled data sets are shown to be close to those of the original microstructures.

1. Introduction

Electron backscatter diffraction (EBSD) performed in a scanning electron microscope (SEM) offers a powerful and ubiquitous modality for the analysis of the microstructure of crystalline samples, generating rich and quantitative orientation data [1]. Chiefly, EBSD can probe a large mm-scale ($\leq 10^6 \mu\text{m}^2$) field of view at a resolution of 40–100 nm, well below common grain size ranges, allowing for grain-resolved data acquisition [2]. The EBSD modality makes use of the standard line scanning approach made available by SEM manufacturers; the diffraction patterns are acquired one line at a time, with user-defined step size, resulting in discrete data sets on a 2-D square or hexagonal grid of sampling points. Most SEMs, however, offer external scan controls, which enable the user to control the beam directly to create alternative sampling schemes. In this paper, we explore an unsupervised dynamic sampling scheme that can generate the same orientation data as the standard EBSD technique, but with a significantly reduced number of sampling points; our approach is not limited to EBSD and can be applied to any characterization technique that allows for point-wise sampling of a region of interest.

Previous research has leveraged the coarse structuring of EBSD data for data acquisition speed and accuracy; for instance, non-local pattern averaging and reindexing seeks to increase indexing noise-robustness by exploiting the spatial correlation of Kikuchi patterns through nonlocal

averaging during postprocessing [3]. Rapid EBSD is another approach that relies on forescatter electron imaging to segment one serial section of the microstructure for static sparse EBSD sampling [4]. Other efforts, such as the “supervised learning approach for dynamic sampling” (SLADS), implement a supervised learning approach for the pixel-wise dynamic sampling of a region of interest (ROI); this is applicable to general image sampling problems, including EBSD applications [5]. SLADS trains a regression model to predict the utility of the next sample pixel based on a number of calculated features of the local neighborhood of already measured pixels; the pixels that are chosen for acquisition are determined by maximizing the reduction of an error metric, the “expected reduction in distortion” or ERD, i.e., one attempts to select as the next sample, the pixel that will provide the maximum amount of information. The core model trained to accomplish this task was originally ordinary least squares regression, but neural network models have been subsequently employed with improved accuracy and similar run-time per pixel choice [6]. To our knowledge, apart from the SLADS related algorithms, no other immediately relevant work exists for pixel-wise dynamic sampling. SLADS’ offline training approach requires data that is similar, including identical ROI dimensions, to the expected input for dynamic sampling. The work presented here is an unsupervised dynamic sampling (UDS) method, with no need for offline training.

* Corresponding author.

E-mail addresses: zvarley@andrew.cmu.edu (Z. Varley), gr20@andrew.cmu.edu (G.S. Rohrer), degraef@cmu.edu (M. De Graef).

2. Material and methods

2.1. Motivation

This section outlines a method by which 2D images may be dynamically sampled and then infilled. The main motivation in design and implementation choices was runtime, to permit usage in electron microscopy applications. The average time needed to choose measurements should be faster than the typical electron beam dwell times; however, this is not strictly required to achieve an advantage over a full grid scan-line-based raster. To make a dynamic pixel-wise sampling algorithm, measurements acquired as the sampling progresses should be used to inform subsequent measurement choices. We have pursued a different approach than the SLADS method which selects samples by maximal expected reduction in distortion (ERD). The present work relies on a paradigm of scoring potential measurements by a product of the distance to the nearest measurement and the variance of the nearest K measurements. This heuristic-based viewpoint removes the need to train a model, and accelerates sampling dramatically.

2.2. Scoring and selection

To filter candidate measurements, a function `FILTER` described in pseudo-code in [algorithm 1](#) is defined. `FILTER` takes a set of pixels and returns a set of winning candidates for measurement. It uses a value function $V(p)$ to determine this winning set. This value function $V(p)$ of a pixel p is defined as the product of its minimum distance to a measured neighbor and the standard deviation of all K neighbor values. The value $V(p)$ of a measured pixel p is arbitrarily low, as no pixel measurement will be made twice. The window function `WINDOW(p)` facilitates non-maximal suppression by returning a set of nearby pixels, within a square window of size W , excluding the pixel p itself. In this approach,

pixels far from their closest measured neighbor, and possessing a local neighborhood of measured pixels with high variance are the most favorable measurements.

In [algorithm 2](#), `SAMPLE` takes in an underlying image, an initial sampling `SOBOL`, a stopping fraction `STOP`, and a filtering function `FILTER`. At the start of data acquisition, no pixels have been observed, so the set of measured pixels M is empty, and the set U holds all pixels in the image. Firstly, a Sobol sequence, one possible low discrepancy quasi-random sequence, is used to draw an initial exploratory sampling [7,8]. This choice is more favorable than fully random sampling due to

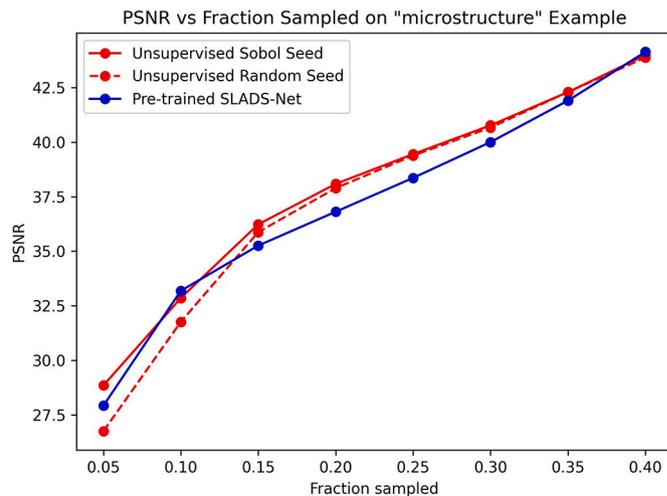


Fig. 2. Direct comparison of the PSNR for reconstruction of the same example micrograph dubbed “microstructure” in [6].

Table 1

Average runtime in μs (10 trials) across resolutions for random images.

Implementation	Image resolution				
	128×128	256×256	512×512	1024×1024	2048×2048
EDT GPU	8.58 ± 0.13	3.96 ± 0.06	2.42 ± 0.03	1.82 ± 0.02	1.72 ± 0.03
EDT CPU	11.24 ± 0.26	6.62 ± 0.09	5.33 ± 0.18	5.83 ± 0.21	8.75 ± 0.38
KeOps GPU	0.94 ± 0.02	0.60 ± 0.04	0.56 ± 0.01	1.90 ± 0.01	7.49 ± 0.09

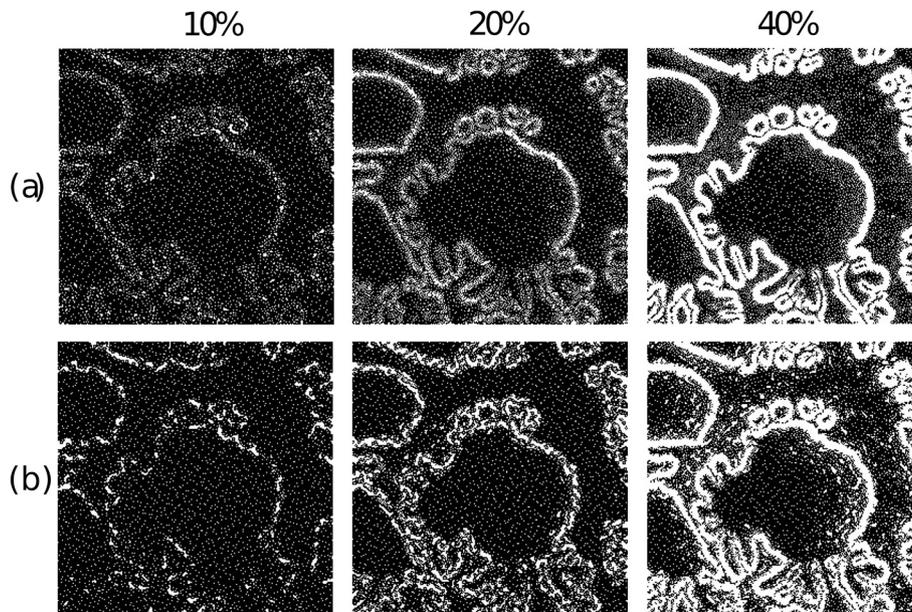


Fig. 1. For a candidate acceptance factors of (a) 0.01 and (b) 0.1 the resultant sampling masks for 10%, 20%, and 40% as is shown.

lower collision rates. These initial measurements are removed from the set of unmeasured pixels U and joined to the set of measured pixels M . Thereafter, the function FILTER continues to select pixels for measurement until a cutoff fraction STOP of the image is reached.

Algorithm 1. Scoring Function FILTER().

```

Require: unmeasured set: $U$ , value function: $V$ , fraction: $F$ , window function:
WINDOW
 $C^0 \leftarrow U$ 
 $W^0 \leftarrow \emptyset$ 
while  $\frac{|W|}{|C|} < F$  do
  for candidate pixel  $c \in C$  do
    if  $V(c) > V(n) \forall$  pixel  $n \in \text{WINDOW}(c)$  then
       $W^{i+1} \leftarrow W^i \cup \{c\}$ 
    end if
  end for
end while
return  $W^i$ 

```

Algorithm 2. Sampling routine SAMPLE().

```

Require: image set: IMG, pixel set: SOBOL, stop percent: STOP,
function:FILTER()
 $M^0 \leftarrow \emptyset$ 
 $U^0 \leftarrow \{\text{pixel } p \text{ for } p \text{ in IMG}\}$ 
 $M^{i+1} \leftarrow M^i \cup \text{SOBOL}$ 
 $U^{i+1} \leftarrow U^i \setminus \text{SOBOL}$ 
while  $\frac{|M|}{|M|+|U|} < \text{STOP}$  do
   $M^{i+1} \leftarrow M^i \cup \text{FILTER}(U)$ 
   $U^{i+1} \leftarrow U^i \setminus \text{FILTER}(U)$ 
end while
return  $M^i$ 

```

2.3. Implementation

2.3.1. Euclidean distance transform implementation

The sampling algorithm is built upon PyTorch, a versatile machine learning library known for its extensive tensor operations on both GPU and CPU, including just-in-time compilation, and for its user-friendly

interface and ease of installation [9]. One of the primary motivations for opting for PyTorch is its straightforward installation process, which typically requires just a few commands. For candidate scoring, our implementations assume a single-channel image or an image that can be mapped to a single channel. In cases where multichannel pixel values are involved, such as orientations in EBSD, these are filled in using their nearest measured values. The source code is available under a 3-clause

BSD license, consistent with that of PyTorch at <https://github.com/ZacharyVarley/DynamicSampling>.

The first component of the score in the dynamic sampling strategy we propose is the Euclidean distance to the nearest measured point. While it is possible to employ data structures like k-dimensional trees (k-d trees)

optimized for nearest neighbor searches in low dimensions, a more efficient alternative exists that exploits the discretized nature of the microscope grid. Crucially, our scoring mechanism doesn't necessitate identifying which specific measurement is the closest; we merely need the distance to the nearest measurement. Leveraging this insight, we utilize the Euclidean distance transformation (EDT). This approach

allows us to calculate the distance to the nearest measured point for all unmeasured points in linear time, $O(N)$ according to the number of image pixels, which significantly enhances the algorithm's efficiency. The repository "FastGeodis: Fast Generalised Geodesic Distance Transform" has convenience Python wrappers with implementations written in CUDA and OpenMP [10].

For the second part of the score, we use the windowed variance of the image, which can be effectively calculated using several Summed Area Tables (SATs). Specifically, SATs are computed for a mask of measurement locations, the sampled locations (with zeros where no measurements exist), and the square of the sampled locations (again with zeros where no measurements exist). These three SATs can then be used to calculate the variance within any window by the difference in the expectation and square expectation. Variance values for pixel windows that have fewer than three measurements within were set to unity so that only the Euclidean distance decides between them. SATs also have $O(N)$ construction keeping the overall runtime linear in the number of pixels.

The pixel averaged runtime in μs of this implementation on both the CPU and GPU is given in the first two rows of Table 1 below. It shows that the runtime per pixel when sampling a random image is approximately constant on both devices, even as the resolution increases dramatically. Further, the average time for most samplings is well below a modest 10 μs dwell time, crucial to the adoption in SEMs for electron backscatter microscopy.

2.3.2. KeOps implementation

While data structures and algorithm complexity analysis can indicate scaling, often it is important to benchmark against brute force approaches. Rapid brute-force k -nearest neighbor queries can be carried out with the library KeOps, which seamlessly integrates with PyTorch [11]. The KeOps Python bindings (PyKeOps) compile optimized CUDA kernels at runtime, and they are ideal for map-reduce type problems. Because the nearest k neighbors are returned, the window size for variance calculation is not needed, and it is replaced with the number of neighbors to consider. Table 1 below shows that the KeOps implementation permits dynamic sampling with dwell times below under 2 μs so long as the resolution does not exceed 1024×1024 .

Due to the improved performance, the KeOps implementation is the main implementation analyzed in the results. Parameter W is the window size of the filtering operation for picking local optima in the scores, F is the top fraction of local optima to accept, K is the number of neighbors used to calculate variance, and S is the Sobol coordinate seed size. If S is a fraction, it refers to the fraction of the total number of pixels in the image, while if it is an integer above 1 it refers to the exact number of pixels (useful notation for small ROIs). Default values around $W = 3$, $F = 0.25$, $K = 3$, and $S = 0.05$ are suitable for most applications.

2.4. Image completion

From the sampled pixels in the image, each unmeasured pixel is infilled as either the inverse distance weighted or unweighted mean of the values of its K -nearest neighbors. As averaging distinct crystalline orientations at grain boundaries falsely produces smoothed boundaries, a single nearest neighbor imputation was utilized for EBSD data.

2.5. Test datasets

We chose three test datasets from scanned image acquisition techniques, which conventionally raster across the ROI without sampling. By sampling an entire dataset, we want to show how image complexity can inform the ideal sampling time-accuracy trade-off.

The first test dataset is the MIDAS dataset provided by the Air Force Research Laboratories' Materials Directorate [12]. It consists of 900 consecutive slices of EBSD, optical micrographs, and backscatter electron (BSE) micrographs from an IN625 Ni-based superalloy sample. Data

acquisition required 40 successive days and generated 3 terabytes, originally motivating the development of a novel dynamic sampling approach to EBSD. The EBSD data was indexed using dictionary indexing in post-processing [13].

The second dataset is a compilation of 961 secondary electron (SE) micrographs of ultra-high carbon steel (UHCS) [14]. The primary microconstituents were spheroidite, carbide networks, and pearlite. All micrographs were provided at 645×484 pixels, after removing 38-pixel tall SEM banners at the bottom of each micrograph. The third test dataset is 230 images of the nodal heart cells from a Sprague-Dawley rat acquired via confocal laser scanning microscopy (CLSM) [15]. All images were of size 2425×2280 pixels, and they were resized to 605×564 pixels with bilinear interpolation before synthetic sampling experiments.

3. Calculation

3.1. Performance evaluation

For orientation data, the disorientation angle between predicted and ground truth crystalline orientations was used to examine imputation fidelity. For grayscale micrographs, the reconstruction peak signal-to-noise ratio (PSNR) in dB between a ground truth image and its predicted reconstruction is used to evaluate the quality of the reconstructed images. This metric is defined in eq. (1) between images A and B in terms of their mean squared error (MSE) and the maximum possible pixel value (MAX):

$$\begin{aligned} \text{PSNR}(A, B) &= 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{MSE}(A, B)} \right) \\ &= 20 \cdot \log_{10} \text{MAX} - 10 \cdot \log_{10} \text{MSE}(A, B) \end{aligned} \quad (1)$$

3.2. Image complexity

Several metrics have been developed to evaluate the complexity of images, which can inform the difficulty of their samplings and imputations. In this work, the complexity measure Q is compared with image histogram entropy [16]. Q is a function of the mean 2×2 windowed variance, V , and the image down-scaling factor S , as given in eq. (2). In eq. (3), s and v are the logarithm (Q is invariant of base choice) of S and V respectively.

$$Q = \frac{1}{s_{\max} - s_{\min}} \int_{s_{\min}}^{s_{\max}} \left[1 - \frac{1}{4} \left(\frac{dv}{ds} \right)^2 \right]_+ ds \quad (2)$$

$$\frac{dv}{ds} = \frac{S}{V} \frac{dV}{dS} \quad (3)$$

For this implementation of Q , all images were rescaled with bilinear interpolation to have their shortest length set to 512 pixels, and the images were further incrementally down-scaled by factors of $\frac{1}{\sqrt{2}}$ down to 8 pixels, so that every fourth rescaling corresponded to a halving: 512, 431, 362, 304, 256...8.

The image entropy based on the pixel intensity histogram is defined as follows:

$$H(X) = - \sum_{x \in X} p(x) \log p(x) \quad (4)$$

where X contains the allowed intensity values in the image. All calculations used 256 grayscale levels. Both measures of image complexity are compared as metrics to gauge sampling and imputation difficulty.

3.3. Microstructure comparison

In order to test any sampling method a metric for comparing microstructures must be established, and we have selected H , the Hellinger

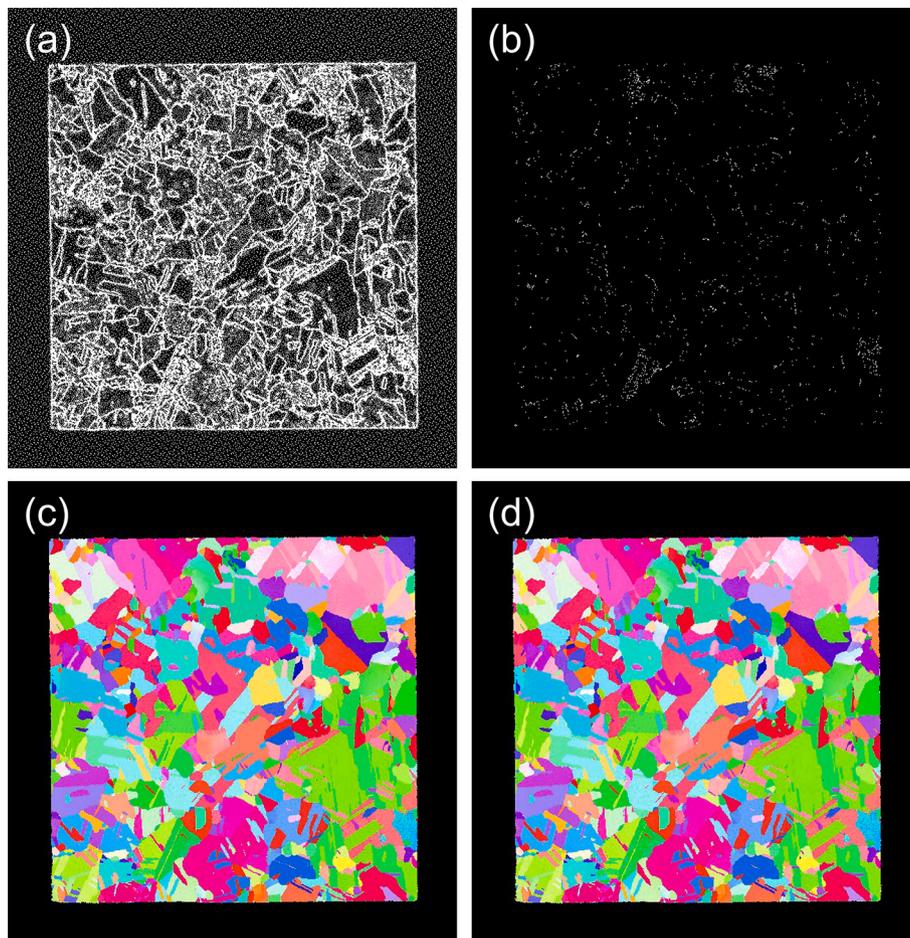


Fig. 3. (a) binary map indicating sampling locations (b) binary map indicating disorientation angle over 2° between (c) ground truth (d) nearest neighbor imputed orientation map (O_n IPF coloring).

metric [17]. For two discrete probability distributions P and Q , the Hellinger distance can also be defined in terms of the Bhattacharyya similarity measure, $\beta(P, Q)$ as is given by Eq. 5 and Eq. 6:

$$\beta(P, Q) = \sum_{i=1}^N \sqrt{P(i)Q(i)} \quad (5)$$

$$H(P, Q) = \sqrt{1 - \beta(P, Q)} \quad (6)$$

This distance has been identified as suitable for comparing microstructure-derived probability distributions such as feature volumes and the affine moment invariant Ω_3 , describing both grain size and shape respectively [18,19].

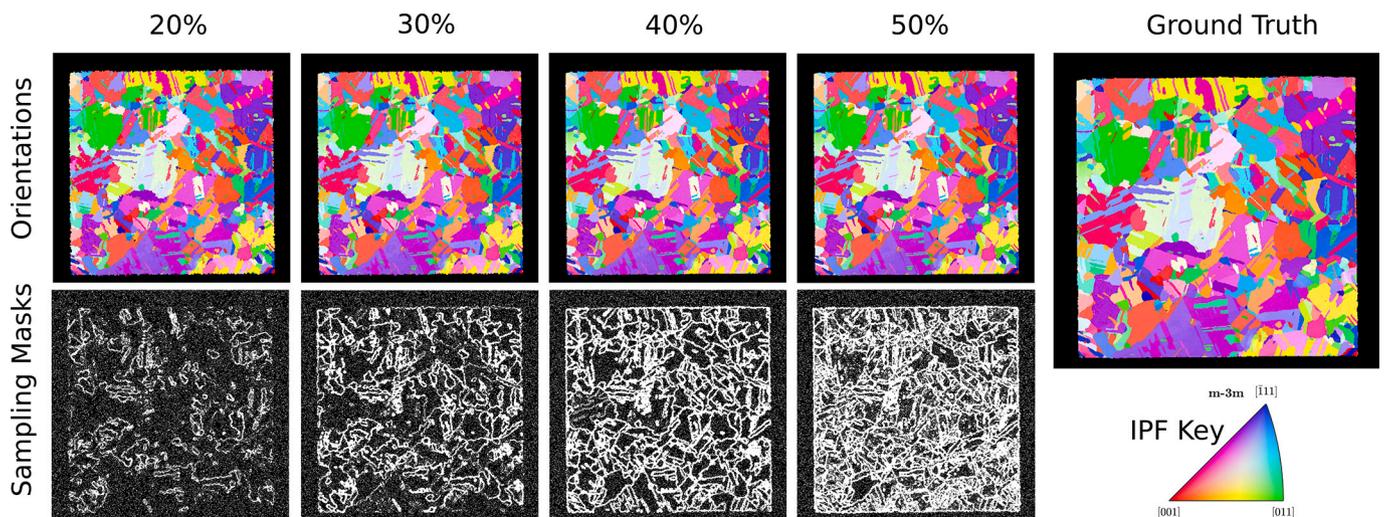


Fig. 4. (left) each row first contains the infilled IPF-Z coloring and sampling locations of the MIDAS dataset (right) the ground truth IPF-Z coloring's and the legend.

4. Results and discussion

4.1. Example samplings across datasets

4.1.1. SLADS comparison

While SLADS-Net is a supervised approach which can leverage patterns learned prior to sampling, it is useful to see where UDS stands in comparison. Further, a pretrained SLADS-Net model was proposed and tested in the original publication. We took the example microstructure image of size 256×256 from their publication and ran UDS on the same image. User parameters were $K = 4$, $W = 3$, $S = 0.05$, and $F = 0.1$ or $F = 0.01$. At most the runtime with KeOps on the GPU was 0.2 s for 40% sampling, or around $7.6 \mu\text{s}$ compute time averaged per pixel. This is approximately 1000 times faster than pretrained SLADS-Net. This runtime improvement could be the difference between method applicability and non-applicability in a variety of characterization contexts. Fig. 1 shows the sampling locations selected using UDS for $F = 0.1$ and $F = 0.01$. Further the resulting PSNR values are plotted for UDS with $F = 0.01$ against the SLADS-Net approach in 2. The sampling locations of UDS appear visually comparable to those of SLADS-Net when $F = 0.01$, but they are more clumped at borders with $F = 0.1$. Both random and Sobol seeding were used with UDS so that we know the improvement is not due to comparing UDS with Sobol seeding to SLADS-Net with random or uniform seeding. Surprisingly, there appears to be an accuracy improvement of UDS over SLADS-Net despite the latter being a supervised approach pretrained on images with a similar appearance. (See Fig. 2.)

4.1.2. MIDAS example slice

The performance of dynamic sampling is examined for an example EBSD serial section of the MIDAS dataset. This task represents the best-case scenario, as compared to the subsequent non-EBSD examples. Due to additional pixel-wise indexing costs, EBSD permits greater latitude in what constitutes a viable sampling dwell time as compared to BSE or CLSM. Converting the 626×610 inverse pole figure (IPF) color map to grayscale permitted dynamic sampling. This is not an injective mapping from orientations, although collision (two adjacent grains sharing indistinguishably-close grayscale values) did not occur in the entire image. Orientation values were imputed using the nearest measurements during post-processing. The sampling was simulated offline, with the ground truth orientation data as the source for pixel values. The user parameters set were $W = 3$, $F = 0.5$, $K = 3$, and $S = 8192$. Fig. 3 shows the result of the dynamic sampling after 33% of the pixels were sampled. For such coarse piece-wise constant images, dynamic sampling densely measures boundaries as shown in (a), and the errors in (b) are found mostly along grain boundaries. At 1 in 3 pixels sampled, the IPF maps of the ground truth and imputed orientation values in (c,d) are visually

indistinguishable. For this task, UDS required an average of $2.1 \mu\text{s}$ wall-time per pixel to choose each measurement. These measurements were selected with an average batch size of 10,000 pixels, yielding a mean batch time of approximately 21 ms.

4.1.3. MIDAS volume

To further demonstrate the capabilities of UDS beyond application to 2D EBSD maps, UDS was applied to the entire MIDAS dataset volume. The dataset was trimmed to exclude pixels which only had BSE data, and to exclude incomplete slices. While the principles behind UDS can be extended to use 3D coordinates, we decided to apply UDS slice-by-slice because there is not a clear way to know the transformation which brings the coordinates from the previous slice into the reference frame of the next slice. In other words, running the dynamic sampling algorithm on a volume successfully registered with the complete dataset would misrepresent the current capabilities of UDS. For this simulated sampling, user parameters set were $W = 3$, $F = 0.25$, $K = 4$, and $S = 0.05$ (image fraction). The initial sampling was done with a 2D Sobol sequence. The individual EBSD slices were then infilled using nearest neighbor imputation. For this example the real component of the fundamental zone quaternion was used as the non-unique mapping of orientations to grayscale values. This process was also repeated for static Random and Sobol sampling.

Fig. 4 shows the result of sampling to 20%, 30%, 40%, and 50%. The chosen locations indicated in the sampling masks demonstrate that UDS was able to quickly locate grain boundaries, and focus on exploring them. Few measurements were spent in data-lacking periphery regions where Kikuchi patterns were still acquired. Using DREAM3D [20], grains were segmented based on a misorientation threshold of 10° and a minimum voxel count of 1000. These grains were then analyzed for their Ω_3 values and the equivalent sphere diameters (ESD) were calculated. Fig. 5 and Fig. 6 show the Hellinger distance between discretized versions of the Ω_3 distributions, and their corresponding empirical cumulative distribution functions (ECDF). Fig. 6 also shows the grain size probability plots assuming a log-normal distribution of the ratio of ESD to mean ESD. The grain size distributions are essentially the same even with static random sampling at 20%. This implies that the grain size distributions might be easier to estimate from a sub-sampled volume than the Ω_3 distributions, although that might not extrapolate to other microstructures.

4.1.4. UHCS dataset

Despite lacking having more complexity, the already single-channel UHCS dataset images were more easily sampled and imputed. Fig. 7 shows that at a slightly lower final sampling fraction of 25%, the algorithm is better able to reconstruct the example micrograph. Fig. 7 (b) shows a majority of the erroneous pixel values are found in missed

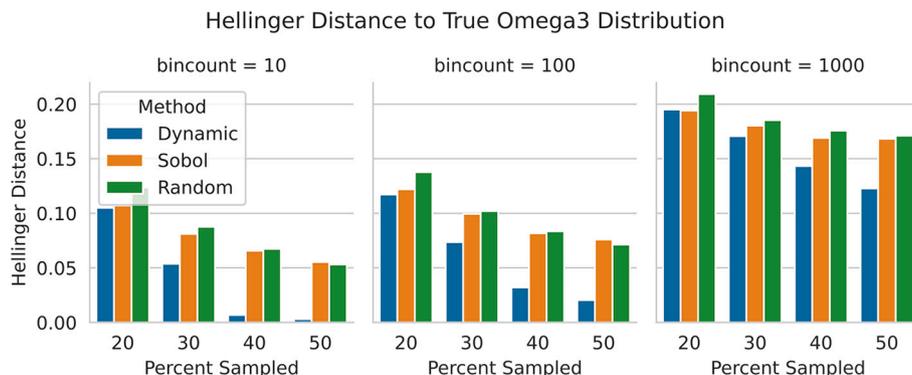


Fig. 5. For uniform binning with 10, 100, and 1000 bins, the Hellinger distances are shown between the Omega3 distributions computed from sampling and the corresponding distribution from the original dataset.

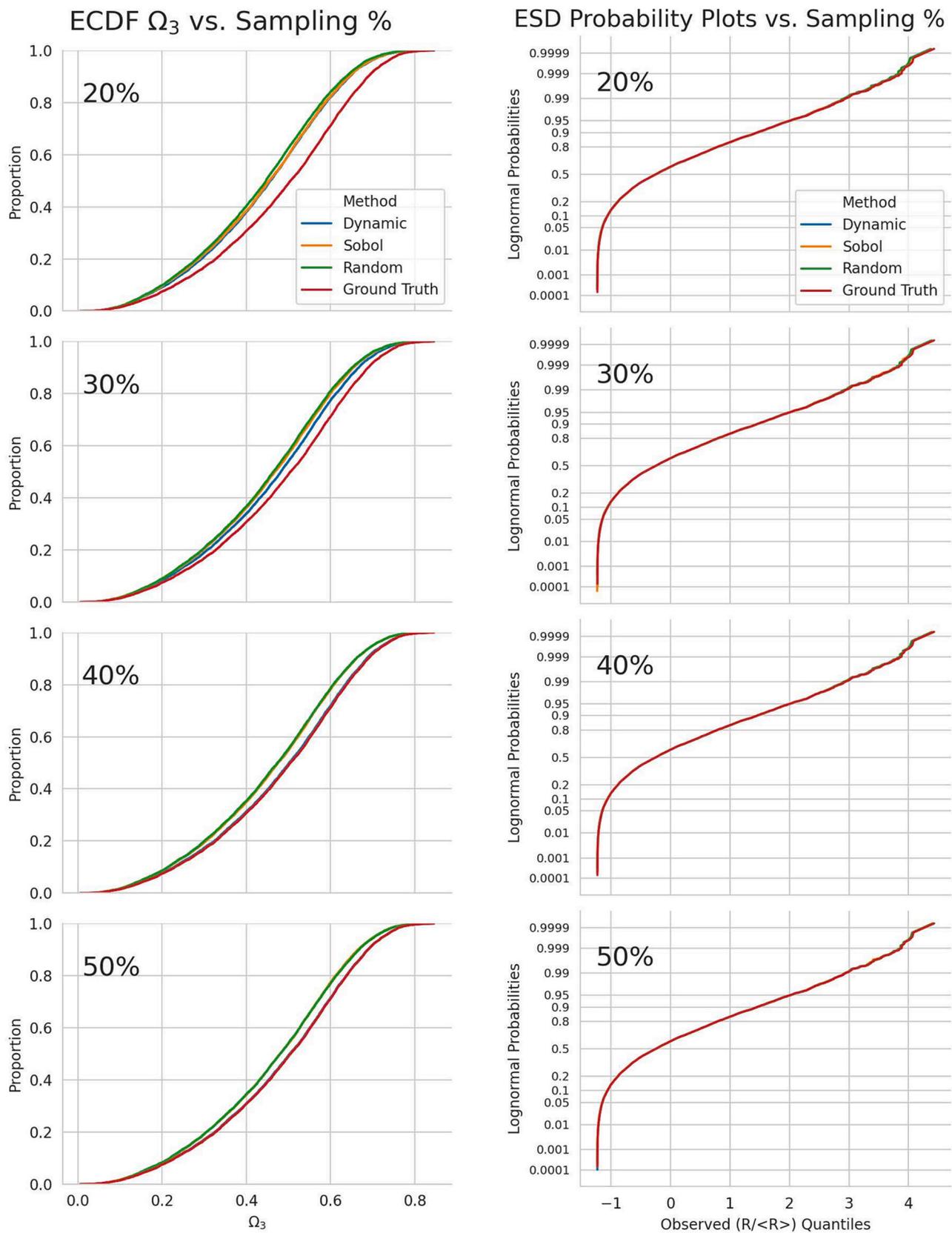


Fig. 6. Each row contains the (left) observed vs. log-normal quantiles plot and the (right) Omega3 empirical cumulative distribution function (ECDF) for the equivalent sphere diameter distributions extracted from the imputed volume. Identical x and y axis plotting ranges are used within columns.

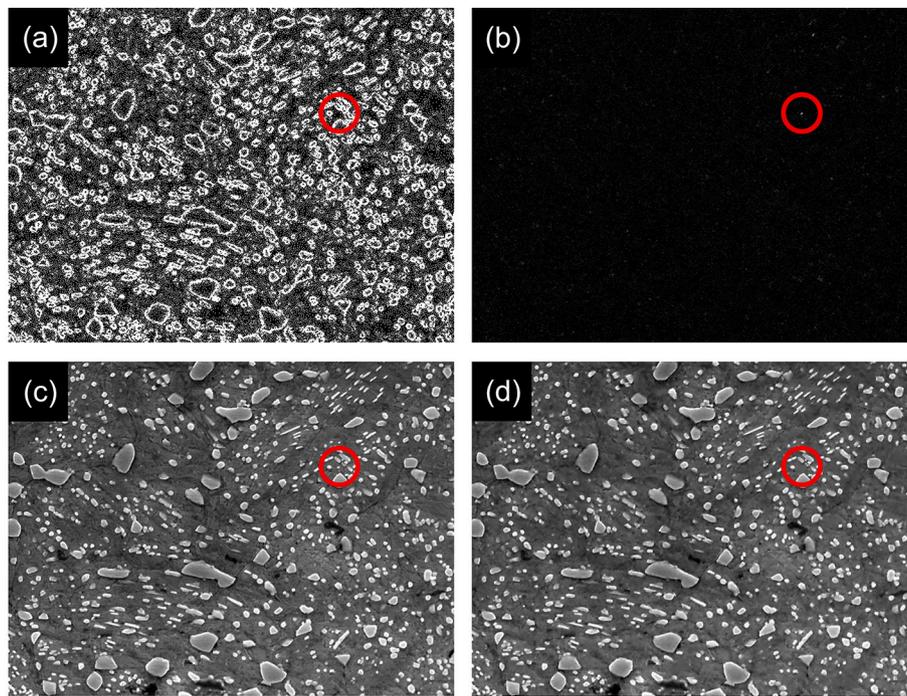


Fig. 7. (a) binary map indicating sampling locations (b) raw map of the absolute difference between (c) ground truth and (d) imputed micrograph.

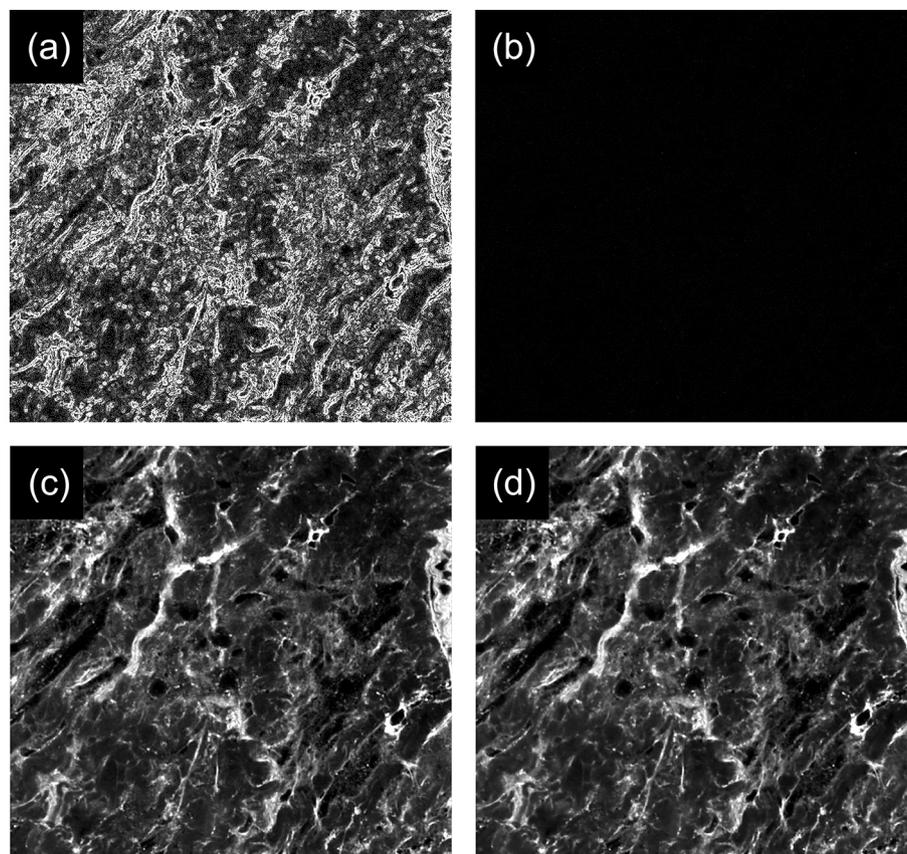


Fig. 8. (a) binary map indicating sampling locations (b) raw map of the absolute difference between (c) ground truth and (d) imputed micrograph.

micrograph features, one of which is circled in red. Upon close visual inspection, the rest of the error can be observed in the slightly blurred appearance of the micrograph in Fig. 7 (d). The user parameters were set to be $W = 3$, $F = 0.5$, $K = 3$, and $S = 8192$. Averaged over 100 runs, the

algorithm runtime was $1.74 \mu\text{s}$ averaged per pixel. The averaged computation time per pixel is comparable to the dwell time used for imaging in the SEM.

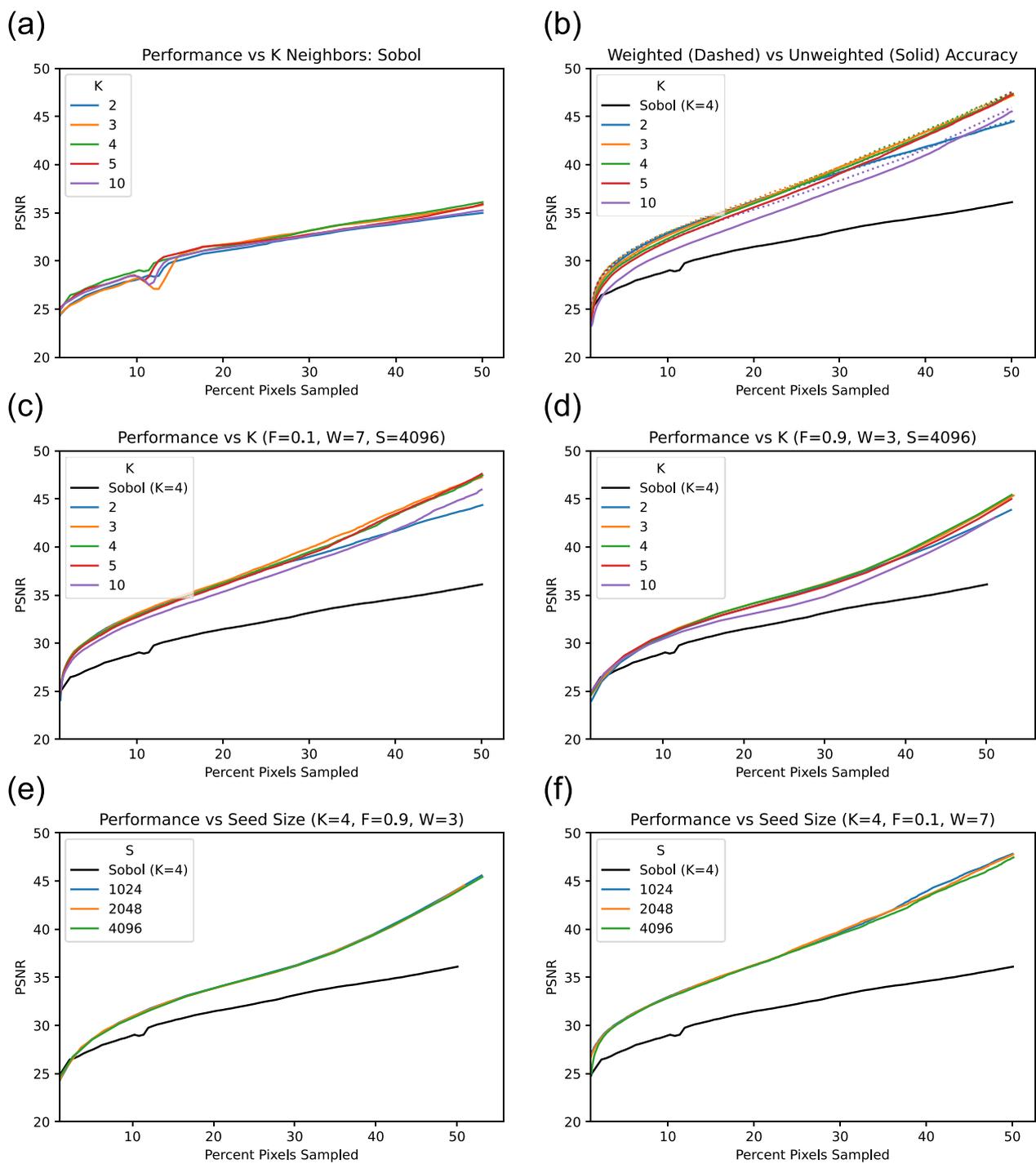


Fig. 9. Percent of pixels sampled up to 50.0% vs. PSNR plotted across K number of nearest neighbors and weighting.

4.1.5. CLSM dataset

For the CLSM dataset, the same user parameters were set to be $W = 3$, $F = 0.5$, $K = 3$, and $S = 8192$. Averaged over 100 runs, the algorithm runtime was similar at $1.64 \mu s$ averaged per pixel. As seen in Fig. 8 this micrograph sampling and imputation shows promising visual fidelity and a compatible runtime with the instrument at hand. For reference, according to its manual, the Zeiss laser scanning microscope (LSM) 5 Duo used to acquire these micrographs has a dwell time of $1.76 \mu s$ on a fast scan speed setting of 9, which is used to quickly adjust parameters before further imaging.

4.2. User parameter dependency

In order to explore the impact of various algorithm settings, a parametric study was conducted on an example BSE micrograph from the IN625 MIDAS dataset. The following subsections report on the importance of each of the user-configurable parameters.

All UDS sampling methods perform better than a Sobol sequence sampling when 3% or more of the images had been observed, as seen in Fig. 9(a). Fig. 9(b) shows that the inverse squared distance weighted mean infilling is consistently better than unweighted mean infilling across all values of K except for a few ranges of percent pixels sampled. For this reason, a weighted mean infilling is utilized henceforth. When

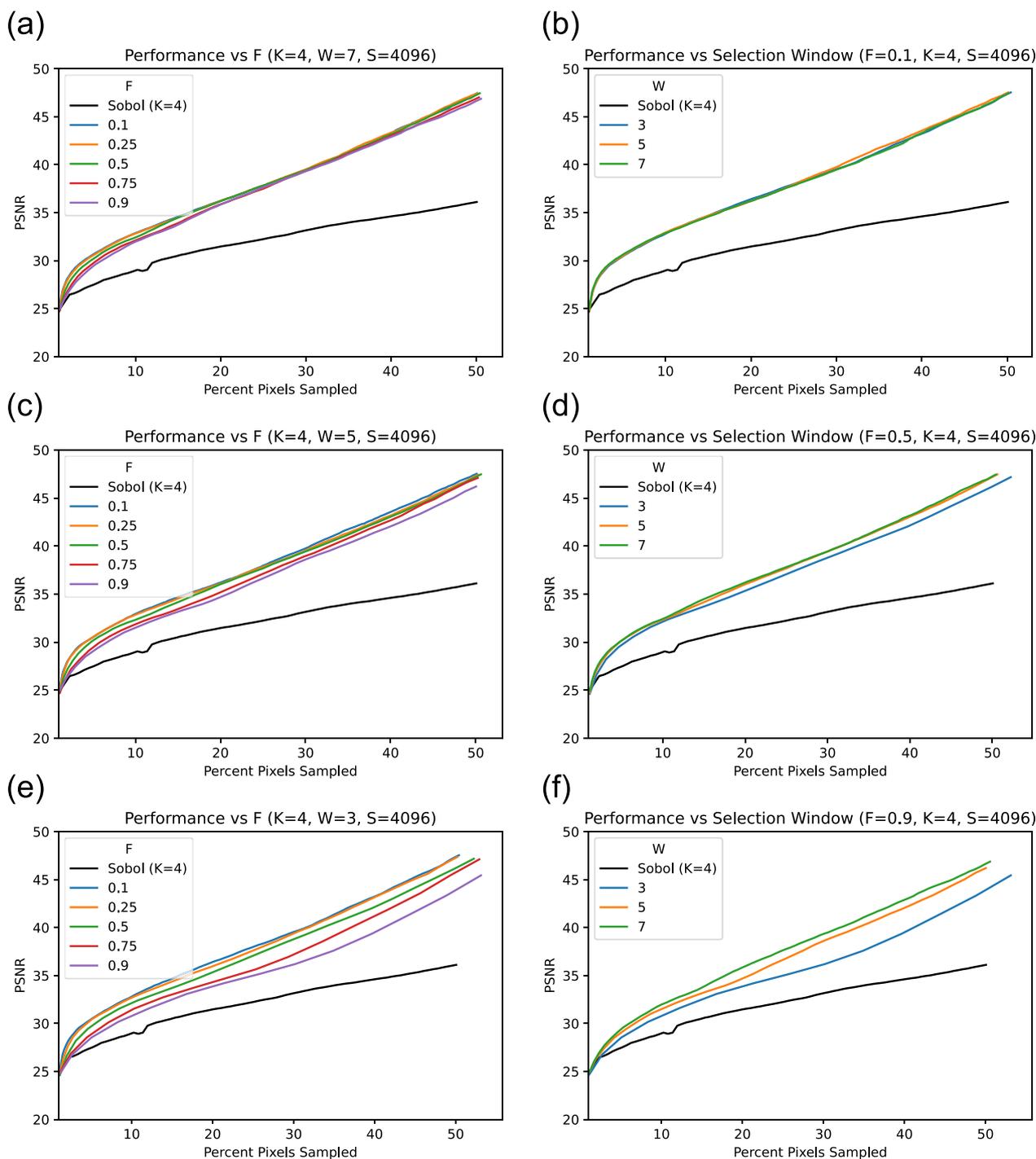


Fig. 10. Percent of pixels sampled up to 50.0% vs PSNR plotted across several user parameters.

comparing Fig. 9 (c) and (d), the number of neighbors K is pivotal to sampling performance at both extremes of F and W . However, comparing Fig. 9(e) and (f), the initial Sobol seed size has little impact on the sampling performance, except at excessive sampling rates above 40%.

The impact of window size W appears to be contingent on the value of F , and vice-versa. Sufficient rejection of poorly scoring local optima, whether by accepting fewer optima or by broadening the optima window, leads to high PSNR. Comparing Fig. 10 column (a,c,e) with column (b,d,f), we see that both acceptance fractions F and W are similar in their impact on dynamic sampling performance. Whether the window size is 3×3 or 7×7 , if F is set sufficiently low as in image (b), the sampling

performance is identical.

4.3. Sampling performance vs. image complexity

To extend these results, the algorithm’s performance was evaluated on the other two aforementioned datasets. These images are naturally heterogeneous in visual appearance, as they include varied steel micrographs and images from a 3D mouse heart cell scan. After sampling all images in the dataset utilizing parameters $W = 3, F = 0.3, S = 4096$, and $K = 3$, the resulting mean image complexity is plotted in Fig. 11 using image entropy and Q score previously defined. Each discrete heatmap box shows the mean image complexity that led to the

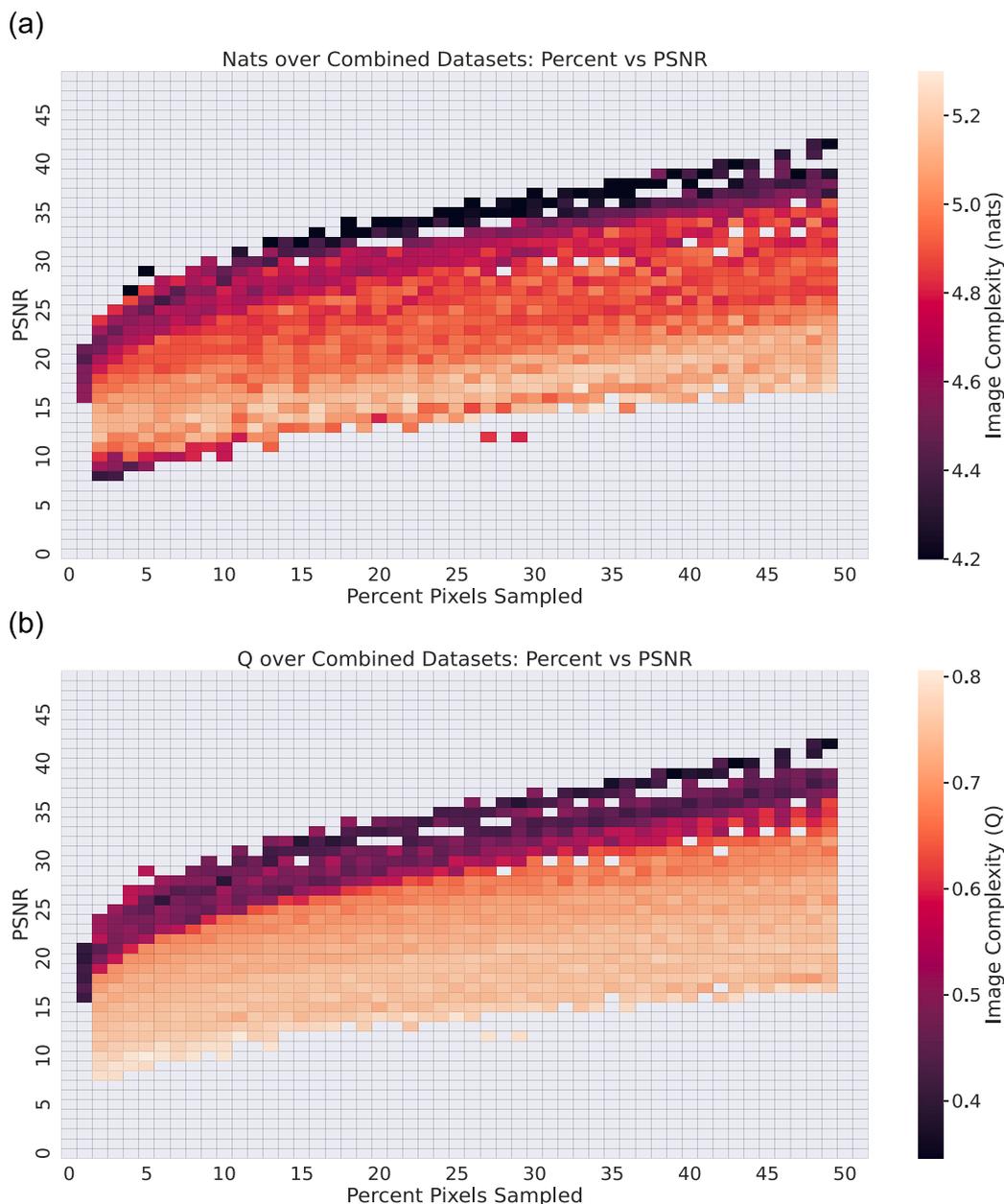


Fig. 11. Heatmap showing the average image complexity that led to a given PSNR at each percent sampled. Only mean values with more than three contributing data-points appear in the heatmap. UHCS and CLSM datasets are combined for both plots.

corresponding reconstruction PSNR at a given percent of pixels sampled. Both image histogram nats in (a) and Q score in (b) indicate that more complex images are more difficult to sample and impute. However, nats in (a) show a noisier relationship with image complexity, but the contrast is maintained across the range of image complexities. Complexity score Q is more binary over these two datasets but lacks clear outliers at 2–3% sampling and 8–10 PSNR.

5. Conclusion and summary

In summary, we have developed a UDS algorithm for 2D images which leverages the parallel processing power of the GPU. As this algorithm requires no prior training, it can be utilized for an arbitrarily shaped ROI. The current state of the art, pretrained SLADS-Net, operates approximately 1000 times slower and produces imputed micrographs with slightly lower PSNR than UDS. The provided Python implementation is easily adoptable and requires minimal to no tuning of default user

parameters. We have analyzed its performance as a function of two different image complexity measures across three different datasets. When executed on modest (laptop) hardware, the algorithm implementation runs faster than respective measurement acquisition times in a variety of imaging modalities. By using the closest measured neighbor alone, multi-channel data such as orientations can be imputed. For real experimental orientation data, measuring 1 in 3 voxels results in visually indistinguishable IPF maps. In terms of grain shapes as characterized by 3-D moment invariants, the reconstructed grains in the sampled case are very close to the grain shapes obtained in normal data acquisition. In the future, we aim to decrease this required sampling fraction with the incorporation of information from previous serial sections and the utilization of less costly modalities to further inform dynamic sampling in EBSD.

For EBSD applications in particular, this work has assumed that indexing can be performed in real time as measurements are gathered. This may be true with Hough or Radon transformation based

approaches; however, dictionary indexing might not be able to be performed in real time. A large hurdle to turning this slice-by-slice sampling into a truly 3D sampling technique is establishing a practical method to realign the sample surface to the same grid locations of the previous slice. Besides this, there are several other improvements planned for UDS. The first planned change for this algorithm is an automatic stopping condition. By periodically comparing the incremental changes in micrograph reconstruction as sampling proceeds, a threshold in PSNR or mean disorientation angle could be used to determine if sufficient sampling has been achieved. Further, electron microscope scan controls often allow arbitrary floating point precision in beam control across the image plane. By using non-discretized coordinates, reconstruction accuracy may improve sufficiently to justify the associated sampling runtime of not caching possible grid queries. Thirdly, an adaptive dwell time, where possible, could allow increased sampling speed and reconstruction accuracy. By briefly, but noisily, checking the center of undiscovered regions, small image features might be more easily discovered.

CRedit authorship contribution statement

Zachary Varley: Conceptualization, Formal analysis, Software, Validation, Writing – original draft. **Gregory S. Rohrer:** Funding acquisition, Supervision, Validation, Writing – review & editing. **Marc De Graef:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to acknowledge stimulating interactions with M. Uchic, M. Shah, S. Donegan, and G. Sparks of the Air Force Research Laboratory in Dayton, OH. This work was supported by the Air Force Research Laboratory under cooperative agreement number FA8650-19-2-5209, as well as the computational facilities of the Materials Characterization Facility at Carnegie Mellon University under grant # MCF-677785.

References

- [1] A.J. Schwartz, M. Kumar, B.L. Adams, D.P. Field (Eds.), *Electron Backscatter Diffraction in Materials Science*, Springer US, Boston, MA, 2009, <https://doi.org/10.1007/978-0-387-88136-2>.
- [2] A.J. Wilkinson, T.B. Britton, Strains, planes, and EBSD in materials science, *Mater. Today* 15 (9) (2012) 366–376, [https://doi.org/10.1016/S1369-7021\(12\)70163-3](https://doi.org/10.1016/S1369-7021(12)70163-3). URL <https://www.sciencedirect.com/science/article/pii/S1369702112701633>.
- [3] P.T. Brewick, S.I. Wright, D.J. Rowenhorst, NLPAR: non-local smoothing for enhanced EBSD pattern indexing, *Ultramicroscopy* 200 (2019) 50–61, <https://doi.org/10.1016/j.ultramic.2019.02.013>. URL <https://www.sciencedirect.com/science/article/pii/S0304399118304248>.
- [4] V.S. Tong, A.J. Knowles, D. Dye, T.B. Britton, Rapid electron backscatter diffraction mapping: painting by numbers, *Mater. Charact.* 147 (2019) 271–279, <https://doi.org/10.1016/j.matchar.2018.11.014>. URL <https://www.sciencedirect.com/science/article/pii/S1044580318326391>.
- [5] G.M.D. Godaliyadda, D.H. Ye, M.D. Uchic, M.A. Groeber, G.T. Buzzard, C. A. Bouman, D.H. Ye, M.D. Uchic, M.A. Groeber, G.T. Buzzard, C.A. Bouman, A supervised learning approach for dynamic sampling, *Elect. Imag.* 28 (2016) 1–8, publisher: Society for Imaging Science and Technology, <https://doi.org/10.2352/ISSN.2470-1173.2016.19.COMMG-153>. URL <https://library.imaging.org/ei/articles/28/19/art00020>.
- [6] Y. Zhang, G.M.D. Godaliyadda, N. Ferrier, E.B. Gulsoy, C.A. Bouman, C. Phatak, SLADS-Net: Supervised Learning Approach for Dynamic Sampling using Deep Neural Networks, arXiv:1803.02972 [eess], Mar. 2018, <https://doi.org/10.48550/arXiv.1803.02972>. URL <http://arxiv.org/abs/1803.02972>.
- [7] S. Joe, F.Y. Kuo, Constructing sobol sequences with better two-dimensional projections, *SIAM J. Sci. Comput.* 30 (5) (2008) 2635–2654, publisher: Society for Industrial and Applied Mathematics, <https://doi.org/10.1137/070709359>. URL <https://epubs.siam.org/doi/10.1137/070709359>.
- [8] I.M. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, *USSR Comput. Math. Math. Phys.* 7 (4) (1967) 86–112, [https://doi.org/10.1016/0041-5553\(67\)90144-9](https://doi.org/10.1016/0041-5553(67)90144-9). URL <https://www.sciencedirect.com/science/article/pii/0041555367901449>.
- [9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshain, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, arXiv:1912.01703 [cs, stat], Dec. 2019, <https://doi.org/10.48550/arXiv.1912.01703>. URL <http://arxiv.org/abs/1912.01703>.
- [10] M. Asad, R. Dorent, T. Vercauteren, FastGeodis: fast generalised geodesic distance transform, *J. Open Source Software* 7 (79) (2022) 4532, arXiv:2208.00001 [cs, eess].
- [11] B. Charlier, J. Feydy, J.A. Glaunès, F.-D. Collin, G. Durif, Kernel Operations on the GPU, with Autodiff, without Memory Overflows, arXiv:2004.11127 [cs], Apr. 2021, <https://doi.org/10.48550/arXiv.2004.11127>. URL <http://arxiv.org/abs/2004.11127>.
- [12] M.G. Chapman, M.D. Uchic, J.M. Scott, M.N. Shah, S.P. Donegan, P.A. Shade, W. D. Musinski, M. Obstalecki, M.A. Groeber, D. Menasche, M.E. Cox, E. J. Schwalbach, 3D reconstruction of an additive manufactured IN625 tensile sample using serial sectioning and multi-modal characterization, *Microsc. Microanal.* 25 (S2) (2019) 342–343, publisher: Cambridge University Press, <https://doi.org/10.1017/S1431927619002447>.
- [13] M.D. Graef, A dictionary indexing approach for EBSD, *IOP Conf. Series: Mater. Sci. Eng.* 891 (1) (2020) 012009, publisher: IOP Publishing, doi:10.1088/1757-899X/891/1/012009, <https://doi.org/10.1088/1757-899X/891/1/012009>.
- [14] B.L. DeCost, T. Francis, E.A. Holm, Exploring the Microstructure Manifold: Image Texture Representations Applied to Ultrahigh Carbon Steel Microstructures, arXiv:1702.01117 [cond-mat], Feb. 2017, <https://doi.org/10.48550/arXiv.1702.01117>. URL <http://arxiv.org/abs/1702.01117>.
- [15] F. Roels, CIL:48302, Sprague-Dawley rat. CIL. Dataset. URL <http://cellimagelibrary.org/images/48302>.
- [16] D.H. Zanette, Quantifying the complexity of black-and-white images, *PLoS One* 13 (11) (2018) e0207879, <https://doi.org/10.1371/journal.pone.0207879>. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6258117/>.
- [17] E. Hellinger, Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen, *J. für die reine angewandte Mathematik* 1909 (136) (1909) 210–271, <https://doi.org/10.1515/crll.1909.136.210>.
- [18] P.G. Callahan, J.P. Simmons, M.D. Graef, A quantitative description of the morphological aspects of materials structures suitable for quantitative comparisons of 3D microstructures, *Model. Simul. Mater. Sci. Eng.* 21 (1) (2012) 015003, publisher: IOP Publishing, <https://doi.org/10.1088/0965-0393/21/1/015003>.
- [19] P.G. Callahan, M. Groeber, M. De Graef, Towards a quantitative comparison between experimental and synthetic grain structures, *Acta Mater.* 111 (2016) 242–252, <https://doi.org/10.1016/j.actamat.2016.03.078>. URL <https://www.sciencedirect.com/science/article/pii/S135964541630252X>.
- [20] M.A. Groeber, M.A. Jackson, DREAM.3D: a digital representation environment for the analysis of microstructure in 3D, *Integr. Mater. Manuf. Innov.* 3 (1) (2014) 56–72, <https://doi.org/10.1186/2193-9772-3-5>.