



Reduced electron exposure for energy-dispersive spectroscopy using dynamic sampling



Yan Zhang^a, G. M. Dilshan Godaliyadda^{a,b}, Nicola Ferrier^c, Emine B. Gulsoy^d, Charles A. Bouman^b, Charudatta Phatak^{a,*}

^a Materials Science Division, Argonne National Laboratory, 9700 Cass Ave, Lemont, IL 60439 United States

^b ECE Department, Purdue University, 465 Northwestern Ave, West Lafayette, IN 47907 United States

^c Mathematics and Computer Science Division, Argonne National Laboratory, 9700 Cass Ave, Lemont, IL 60439 United States

^d Department of Materials Science and Engineering, Northwestern University, 2220 Campus Drive, Evanston, IL 60208 United States

ARTICLE INFO

Article history:

Received 23 June 2017

Revised 17 October 2017

Accepted 20 October 2017

Available online 23 October 2017

Keywords:

Scanning electron microscopy (SEM)

Energy dispersive spectroscopy (EDS)

Dynamic sampling

SLADS

Neural networks

Dose reduction

ABSTRACT

Analytical electron microscopy and spectroscopy of biological specimens, polymers, and other beam sensitive materials has been a challenging area due to irradiation damage. There is a pressing need to develop novel imaging and spectroscopic imaging methods that will minimize such sample damage as well as reduce the data acquisition time. The latter is useful for high-throughput analysis of materials structure and chemistry. In this work, we present a novel machine learning based method for dynamic sparse sampling of EDS data using a scanning electron microscope. Our method, based on the supervised learning approach for dynamic sampling algorithm and neural networks based classification of EDS data, allows a dramatic reduction in the total sampling of up to 90%, while maintaining the fidelity of the reconstructed elemental maps and spectroscopic data. We believe this approach will enable imaging and elemental mapping of materials that would otherwise be inaccessible to these analysis techniques.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Analytical electron microscopy based on energy dispersive X-ray spectroscopy (EDS) is a very versatile and successful technique for exploring elemental composition in microanalysis from the sub-nanometer scale to the micron scale [1–3]. Modern scanning electron microscopes (SEM) equipped with EDS detectors are routinely used for qualitative, semi-quantitative or quantitative elemental mapping of various materials ranging from inorganic to organic, and including biological specimens. Although EDS allows us to identify the elemental composition at a given location with high accuracy, each spot measurement can take anywhere from 0.1 to 10 s to acquire. As a result, if one wants to acquire EDS maps on a rectilinear grid with 256×256 grid points, the total imaging time could be on the order of tens to hundreds of hours. Furthermore, during the acquisition process, the sample gets exposed to a highly focused electron beam that can result in unwanted radiation damage such as knock-on damage, radiolysis, sample charging or heating. Organic and biological specimens are more prone to such damage due to electrostatic charging. Therefore minimizing

the total radiation exposure of the sample is also of critical importance. One approach to solve this problem is to sample the rectilinear grid sparsely. However, it is critical that elemental composition maps reconstructed from these samples are accurate. Hence the selection of the measurement locations is of critical importance.

Sparse Sampling techniques in the literature fall into two main categories – Static Sampling and Dynamic Sampling (DS). In Static Sampling the measurement locations are predetermined. Such methods include object independent static sampling methods such as Random Sampling strategies [4] and Low-discrepancy Sampling strategies [5], and sampling methods based on a model of the object being sampled such as those described in [6,7]. In Dynamic Sampling, previous measurements are used to determine the next measurement or measurements. Hence, DS methods have the potential to find a sparse set of measurements that will allow for a high-fidelity reconstruction of the underlying sample. DS methods in the literature include dynamic compressive sensing methods [8,9] which are meant for unconstrained measurements, application specific DS methods [10–12], and point-wise DS methods [13–15]. In this paper, we use the dynamic sampling method described in [15], Supervised Learning Approach for Dynamic Sampling (SLADS). SLADS is designed for point-wise measurement schemes, and is both fast and accurate, making it an ideal candidate for EDS mapping.

* Corresponding author.

E-mail address: cd@anl.gov (C. Phatak).

In SLADS, each measurement is assumed to be scalar valued, but each EDS measurement, or spectrum, is a vector, containing the electron counts for different energies. Therefore, in order to apply SLADS for EDS, we need to extend SLADS to vector quantities or convert the EDS spectra into scalar values. In particular, we need to classify every measured spectrum as pure noise or as one of L different phases. To determine whether a spectrum is pure noise, we use a Neural Network Regression (NNR) Model [16]. For the classification step we use Convolutional Neural Networks (CNNs).

Classification is a classical and popular machine learning problem in computer science for which many well-established models and algorithms are available. Examples include logistic regression and Support Vector Machines (SVM) which have been proven very accurate for binary classification [17]. Artificial neural networks, previously known as multilayer perceptron, have recently gained popularity for multi-class classification particularly because of CNNs [18,19] that introduced the concept of deep learning. The CNNs architecture has convolution layers and sub-sampling layers that extract features from input data before they reach fully connected layers, which are identical to traditional neural networks. CNNs-based classification has shown impressive results for natural images, such as those in the ImageNet challenge dataset [20], the handwritten digits (MNIST) dataset [21] and the CIFAR-10 dataset [22]. CNNs are also becoming popular in scientific and medical research, in areas such as tomography, magnetic resonance imaging, genomics, protein structure prediction etc. [23–26]. It is because of the proven success of CNNs that we chose to use one for EDS classification.

In this paper, we first introduce the theory for SLADS and for detection and classification of EDS spectra. Then, we show results from four SLADS experiments performed on EDS data. In particular, we show experiments on a 2-phase sample measured at two different resolutions and experiments on a 4-phase sample measured at two different resolutions. We also evaluate the performance of our classifier.

2. Theoretical methods

In this section we introduce the theory behind dynamic sampling as well as how we adapt it for EDS.

2.1. SLADS Dynamic sampling

Supervised learning approach for Dynamic Sampling (SLADS) was developed by Godaliyadda et al. [15,27,28]. The goal of dynamic sampling, in general, is to find the measurement which, when added to the existing dataset, has the greatest effect on the expected reduction in distortion (ERD). It is important to note that in this section we assume, as in the SLADS framework, that every measurement is a scalar quantity. We later elaborate how we generalize SLADS for EDS, where measurements are vectors.

First, we define the image of the underlying object we wish to measure as $X \in \mathbb{R}^N$, and the value of location s as X_s . Now assume we have already measured k pixels from this image. Then we can construct a measurement vector,

$$Y^{(k)} = \begin{bmatrix} s^{(1)}, X_{s^{(1)}} \\ \vdots \\ s^{(k)}, X_{s^{(k)}} \end{bmatrix}.$$

Using $Y^{(k)}$ we can then reconstruct an image $\hat{X}^{(k)}$.

Second, we define the distortion between the ground-truth X and the reconstruction $\hat{X}^{(k)}$ as $D(X, \hat{X}^{(k)})$. Here $D(X, \hat{X}^{(k)})$ can be any metric that accurately quantifies the difference between X and $\hat{X}^{(k)}$. For example, if we have a labeled image, where each label corresponds to a different phase, then,

$$D(X, \hat{X}^{(k)}) = \sum_{i=1}^N I(X_i, \hat{X}_i^{(k)}), \tag{1}$$

where I is an indicator function defined as

$$I(X_i, \hat{X}_i^{(k)}) = \begin{cases} 0 & X_i = \hat{X}_i^{(k)} \\ 1 & X_i \neq \hat{X}_i^{(k)} \end{cases}. \tag{2}$$

Assume we measure pixel location s , where $s \in \{\Omega \setminus \mathcal{S}\}$, where Ω is the set containing indices of all pixels, and \mathcal{S} is the set containing pixel locations of all measured pixels. Then we can define the reduction in distortion (RD) that results from measuring s as,

$$R^{(k;s)} = D(X, \hat{X}^{(k)}) - D(X, \hat{X}^{(k;s)}). \tag{3}$$

Ideally we would like to take the next measurement at the pixel that maximizes the RD. However, because we do not know X , i.e. the ground-truth, the pixel that maximizes the expected reduction in distortion (ERD) is measured in the SLADS framework instead. The ERD is defined as,

$$\bar{R}^{(k;s)} = \mathbb{E}[R^{(k;s)} | Y^{(k)}]. \tag{4}$$

Hence, in SLADS the goal is to measure the location,

$$s^{(k+1)} = \arg \max_{s \in \Omega} \{\bar{R}^{(k;s)}\}. \tag{5}$$

In SLADS the relationship between the measurements and the ERD for any unmeasured location s is assumed to be given by,

$$\mathbb{E}[R^{(k;s)} | Y^{(k)}] = \hat{\theta} V_s^{(k)}. \tag{6}$$

Here, $V_s^{(k)}$ is a $t \times 1$ feature vector extracted for location s and $\hat{\theta}$ is $1 \times t$ vector that is computed in training.

To compute $\hat{\theta}$ we use the procedure described in [15,27]. First, we select M images that are similar to the image of the object we intend to measure. Then, for an image $m \in M$, we first select u_1 number of pixels at random as measurements, and designate the remaining pixels as unmeasured pixels. Then, for every unmeasured pixel s , we can extract a feature vector V_s and compute the RD in Eq. (3) by computing the reconstruction before, $\hat{X}^{(k)}$, and reconstruction after, $\hat{X}^{(k;s)}$, pixel s is included in the measurements. However, with this method, we need to compute 2 reconstructions for each unmeasured pixel, and for the $N - u$ unmeasured pixels, we need to compute $N - u + 1$ reconstructions. To address this problem, an approximation to the RD is introduced in [15,27], which reduces the number of reconstructions to 1 per training image m and measurement selection u .

$$\tilde{R}^{(s)} = \sum_{r \in \Omega} h_{s,r} D(X_r, \hat{X}_r), \tag{7}$$

where

$$h_{s,r} = \exp \left\{ -\frac{1}{2\sigma_s^2} \|r - s\|^2 \right\} \tag{8}$$

and $\|r - s\|$ is the Euclidean distance between pixels r and s , and σ_s is given by

$$\sigma_s = \frac{\min_{t \in \mathcal{S}} \|s - t\|}{c}, \tag{9}$$

where \mathcal{S} is the set of measured locations. It is important to note that we have removed the superscript k in these and the proceeding equations because when building the training database, we extract entries with different initial measurements i.e. different values of k . The procedures for estimating this parameter an empirical validation of this approximation are also detailed in [15,27]. Now, from different random selections u_1, u_2, \dots, u_h for each image $m \in M$, we extract a features vector and the corresponding RD for every unmeasured pixel, to form,

$$\mathbf{R} = \begin{bmatrix} \tilde{R}^{(s_1)} \\ \vdots \\ \tilde{R}^{(s_n)} \end{bmatrix}, \mathbf{V} = \begin{bmatrix} V_{s_1} \\ \vdots \\ V_{s_n} \end{bmatrix}. \tag{10}$$

It is important that u_1, u_2, \dots, u_h are selected so that high and low measurement densities are represented in the training database. Then, using \mathbf{R} and \mathbf{V} we estimate $\hat{\theta}$ by,

$$\hat{\theta} = \arg \min_{\theta \in \mathbb{R}^t} \|\mathbf{R} - \mathbf{V}\theta\|^2, \quad (11)$$

2.2. Adapting SLADS for energy-dispersive spectroscopy

In SLADS it is assumed that a measurement is a scalar value. However, in EDS, the measurement spectrum is a $p \times 1$ vector. So to use SLADS for EDS we either need to redefine the distortion metric, or convert the $p \times 1$ vector spectra to a labeled discrete class of scalar values. In this paper, we use the latter approach.

In order to make a meaningful conversion, the scalar value should be descriptive of the measured energy spectrum, and ultimately allow us to obtain a complete understanding of the underlying object. The objective in this work is to identify the distribution of different phases in the underlying object. Note that a phase is defined as the set of all locations in the image that have the same EDS spectrum. So if we can classify the measured spectra into one of L classes, where the L classes correspond to the L different phases, and hence L different spectra, then we can readily adapt SLADS for EDS. Fig. 1(a) shows this adaptation in more details. The method we used for classifying spectra is detailed in the next section.

2.3. Classifying energy dispersive spectra

Assume the EDS measurement from a location s is given by $Z_s \in \mathbb{R}^p$, where $p > 1$. Hence we need to convert Z_s to a discrete integer class X_s , where X_s is a label that corresponds to the elemental composition (phase) at location s . Also, let us assume that the sample we are measuring has L different phases, and therefore, $X_s \in \{0, 1, \dots, L\}$, where 0 corresponds to an *ill-spectrum*. An *ill-spectrum* can be caused by a sample defect, equipment noise or other undesired phenomena, and therefore is not from one of the L phases that are known.

To classify spectra, there are many approaches that could be used. One such widely used approach, is a third-party software named “Hyperspy” [29], which finds the elemental composition of a spectrum by identifying the peak locations of the spectrum. For our implementation, we attempted to use Hyperspy, but found the misclassification rate to be too high. Another popular classification method is Support Vector Machines [17], which we did not use because this method is a binary classifier, and we would need to train multiple binary classifiers for a multi-class problem.

In this paper, to classify Z_s into one of $L + 1$ classes we use a two step approach as shown in Fig. 1(b). In the first step, we determine if the measured spectrum is an *ill-spectrum*. We call this step the *detection step*. If we determine that Z_s is an *ill-spectrum* then we let $X_s = 0$. If not we move on to the second step of determining which of the L phases Z_s belongs to and assign that label to X_s . We call this second step the *classification step*. We chose to use a CNNs based approach because of its proven accuracy and because we observed the classification accuracy was very high with CNNs. In the next two sections we will explain the algorithms we used for the Detection and Classification steps.

2.3.1. Detection using neural network regression

In the detection step, we use a Neural Network Regression (NNR) model to detect the *ill-spectrum* class [16]. The NNR model we use has Q neurons in each hidden layers as well as the output layer.

Assume that we have M training spectra for each of the L phases. The goal of training is to find a function $\hat{f}(\cdot)$ that minimizes the Loss function and project training spectra onto a pre-set

straight line f , where:

$$\text{Loss} = \frac{1}{2} \sum_{r \in \{1, 2, \dots, LM\}} \|f - \hat{f}(Z_s)\|^2 \quad (12)$$

Here, LM is the total number of training spectra and Z_s where $s \in \{1, 2, \dots, LM\}$ denotes one training spectrum. It is important to note that since this is a neural network architecture, by saying we find $\hat{f}(\cdot)$, it is understood that we find the weights of the neural networks, that correspond to $\hat{f}(\cdot)$.

To determine if a spectrum Z_s is an *ill-spectrum* or one which belongs to one of L phases, we first compute,

$$g_s = |f - \hat{f}(Z_s)|. \quad (13)$$

where, $g_s \in \mathbb{R}^Q$. Then we compute the variance metric,

$$\sigma^2(Z_s) = \frac{1}{Q} \sum_{i=1}^Q [g_{s,i} - \mu_s]^2 \quad (14)$$

where, $g_{s,i}$ is the i th element of the vector g_s and

$$\mu_s = \frac{1}{Q} \sum_{i=1}^Q g_{s,i} \quad (15)$$

Then a pre-set threshold is applied to the variance metric to decide whether the Z_s is an *ill-spectrum* i.e.

$$\hat{X}_s = \begin{cases} 0, & \sigma^2(Z_s) > T \\ \{1, 2, \dots, L\}, & \sigma^2(Z_s) \leq T. \end{cases} \quad (16)$$

2.3.2. Classification using convolutional neural networks

The next task at hand is to classify the spectrum according to one of the L labels, given that we found a spectrum Z_s for some location s , which is not an *ill-spectrum*. For this classification problem we use the convolutional neural networks (CNNs) as described in [20] and implement using tensorflow [30].

The CNNs we use in this paper has two convolution layers, each followed by a max-pooling layer followed by three fully connected layers, shown in Fig. 2. The first convolution layer has u_1 $1 \times k$ kernels sliding across the input spectrum with a stride v to extract u_1 features, each of size $1 \times n_1$. The max-pooling layer that follows this layer again operates with the same stride and kernel size, resulting in u_1 features, each of size $1 \times m_1$. The second convolution layer increases the number of features from u_1 to u_2 , where, $u_2 \bmod u_1 = 0$, by the application of u_2 kernels of size $1 \times k$ at the same stride (v) to the output of the first max-pooling layer. The max pooling layer that follows is identical to the previous max-pooling layer.

After the convolution and max-pooling layers, all feature values are stacked into a single vector known as a *flat layer*. The flat layer is the transition into the fully connected layers that follow. These layers have the same architecture as typical neural networks. In the fully connected layers, the number of neurons in each layer is reduced in our implementation.

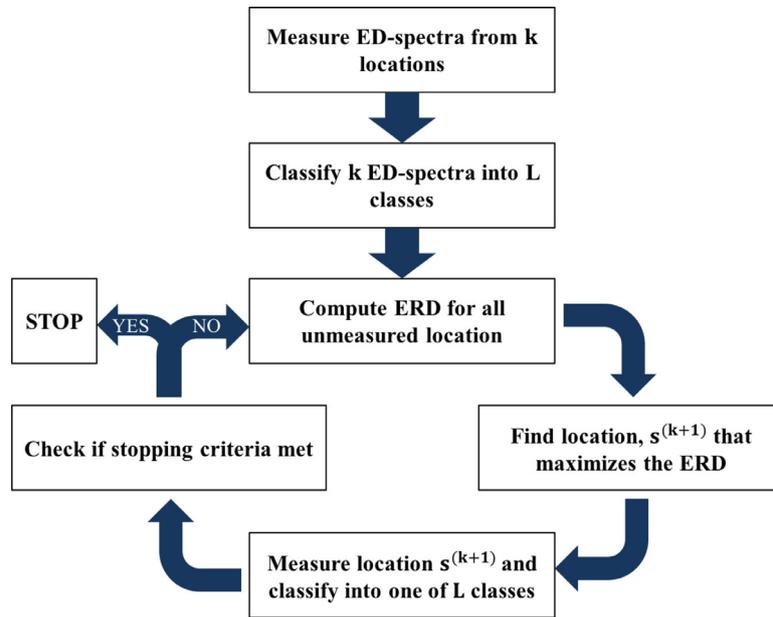
The output of the fully connected layers is a $1 \times L$ vector, X_s^1 . Each entry of this vector is then sent through a SoftMax function to create again a $1 \times L$ vector, which we will denote as X_s^2 , for a location s .

$$X_{s,i}^2 = \frac{\exp(X_{s,i}^1)}{\sum_{j=1}^L \exp(X_{s,j}^1)}. \quad (17)$$

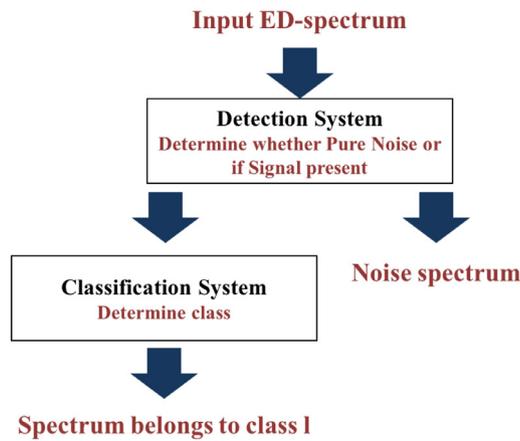
Here, $X_{s,j}^1$ corresponds to the j th component of X_s^1 .

Now assume we have the same training examples as in the previous section i.e. M spectra from the L phases. When training the CNNs we minimize the Cross-Entropy, defined as,

$$CE = - \sum_{r=1}^{LM} X_s^{\text{one-hot}} \log X_s^2 \quad (18)$$



(a)



(b)

Fig. 1. (a) The SLADS algorithm adapted for ED-spectra and (b) the two-tiered structure of our EDS classification system.

where, LM is the total number of training samples, and $X_s^{one-hot}$ is the “one-hot” representation of X_s . Here, the “one-hot” notation of label m , when L labels are available, is a $L \times 1$ dimensional vector with 1 at location m and zeros everywhere else.

3. Experimental methods and results

In this section, we will first describe the experimental methods used in the simulated experiments in Section 3.1 and then present the results for the simulated experiments in Section 3.2.

3.1. Experimental methods

Here we first present how we generate images for training and simulated objects to perform SLADS on and finally how we train the classifier we described in Section 2.3. For all the experiments we used a Phenom ProX Desktop SEM. The acceleration voltage of the microscope was set to 15 kV. To acquire spectra for the experi-

ments, we used the EDS detector in *spot mode* with an acquisition time of 10 s.

3.1.1. Constructing segmented images for training

We first acquired representative images from the object with L different phases using the back scattered detector on the Phenom. Then we segmented this image so that each label would correspond to a different phase. Finally we denoised the image using an appropriate denoising scheme to create a clean image.

3.1.2. Constructing a simulated object

In this paper, we want to dynamically measure an object in the EDS mode. This means that if the beam is moved to a location s , the measurement extracted, Z_s , is a p dimensional vector. So we can think of the problem as sampling an object with dimensions $N \times N \times p$, where we can only sample sparsely in the spatial dimension, i.e. the dimension with $N \times N$ points. This hypothetical object is what we call here *asimulated object*.

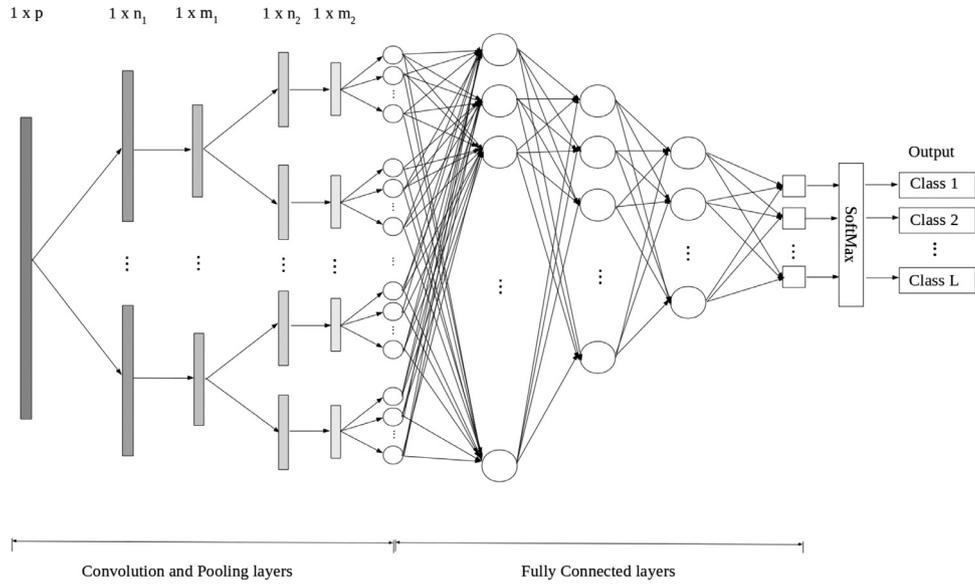


Fig. 2. The CNNs architecture for EDS spectra classification.

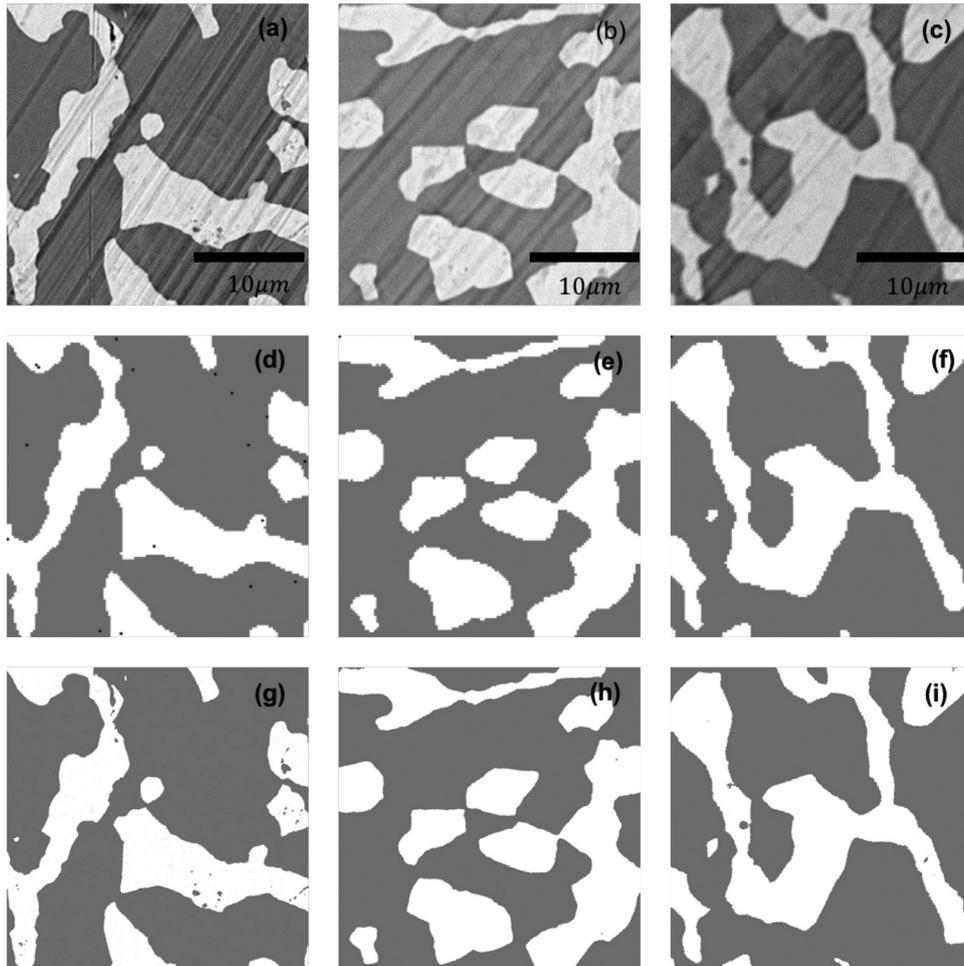


Fig. 3. Testing and training images for the 2-phase SLADS experiments. (a)–(c): Original SEM images; (d)–(f): 128×128 images with labels 0,1, and 2; (g)–(i): 1024×1024 images with labels 0,1, and 2; the images in the first column are the ones used for testing and the others are the ones used for training.

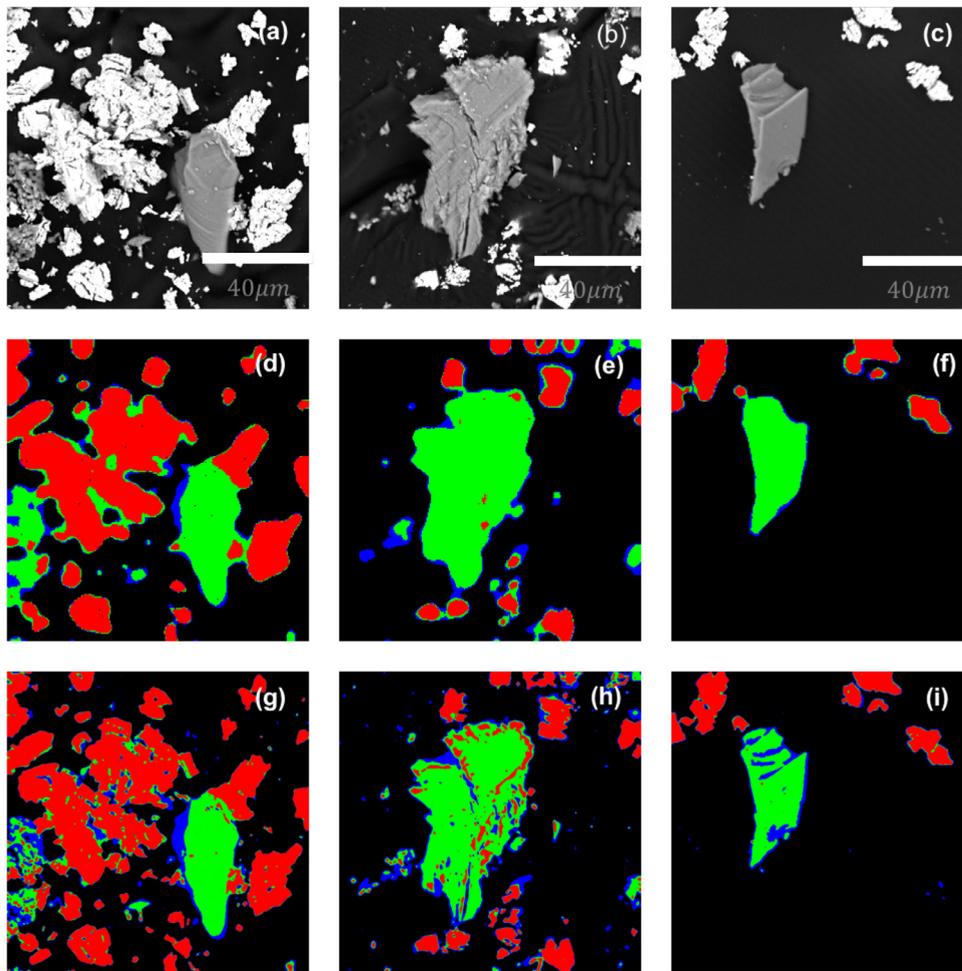


Fig. 4. Testing and training images for the 4-phase SLADS experiments. (a)–(c): Original SEM images; (d)–(f): 256×256 images with labels 0,1,2,3, and 4; (g)–(i): 1024×1024 images with labels 0,1,2,3, and 4; the images in the first column are the ones used for testing and the others are the ones used for training.

To create the simulated object, we first acquire an SEM image, and segment it just as in the previous section. We then add noise to this image by assigning the label 0 to a randomly selected set of the pixels. Then we experimentally collect M different spectra from each of the L different phases using the EDS detector in *spot mode*. Finally we raster through the segmented and noise added image and assign a spectrum to each pixel location in the following manner: If the value read at a pixel location s is 0 we assign a pure noise spectrum to that location. If the value read is $l \in \{1, 2, \dots, L\}$, we randomly pick one of the M spectra we acquired previously for phase l and add Poisson noise to it. Then we assign the noise added spectrum to location s . It is important to note that the noise added to each location is independent of location and is different for each pixel location. So now we have an object of size $N \times N \times p$ to use in our SLADS experiment.

3.1.3. Specification of the neural networks to classify EDS spectra

In order to train and validate the detection and classification neural networks, we again collected M^{train} spectra for each phase. Then we added Poisson noise to each spectrum and then used half of the spectra to train the NNR and the CNNs, and used the other half to validate, before using it in SLADS.

The NNR network we used has 5 fully connected hidden layers, each with 100 neurons. The CNNs classification system we used has kernel size $k = 10$ and a stride of 2 for all convolution and max-pooling layers. The number of features in the first and second convolution layers are 8 and 16 respectively. The flat layer

stacks all features from second max-pooling layer into a 2048 dimensional vector. The number of neurons for the following 3 fully connected layers are 100, 32 and 8.

3.2. Results

In this section, we will present results from SLADS experiments performed on 4 different simulated objects. To quantify the performance of SLADS, we will use the total distortion (TD) metric. The TD after k measurements are made is defined as,

$$TD_k = \frac{1}{|\Omega|} D(X, \hat{X}^{(k)}). \quad (19)$$

We will also evaluate the accuracy of the classification by computing the misclassification rate.

3.2.1. Experiment on simulated 2-phase object with Pb–Sn alloy

Here we will present results from sampling two 2-phase simulated objects, one with dimensions $128 \times 128 \times p$, and the other with dimensions $1024 \times 1024 \times p$ created using spectra and SEM images acquired from a Pb–Sn eutectic alloy sample [31]. It is important to note that the dimension p here corresponds to the dimension of the spectrum, i.e. in the simulated object at each pixel location we have a p dimensional spectrum. One of the phases has Pb and Sn, and the other only Sn.

Both these objects, as well as the training data for SLADS, were created using SEM images taken at 1024×1024 resolution. We created 128×128 images by down-sampling the original 1024×1024

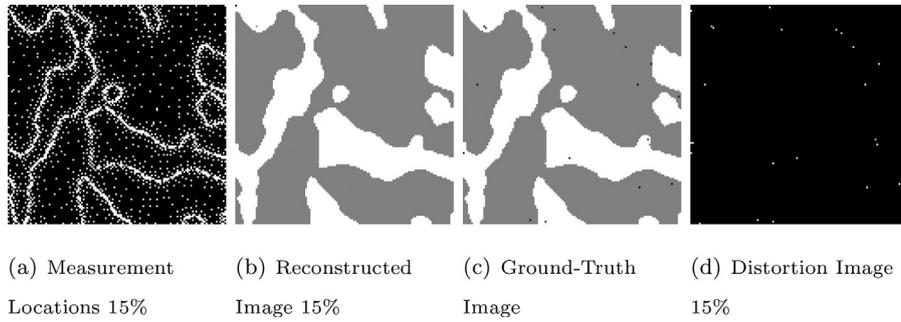


Fig. 5. Results of EDS-SLADS experiment performed on the 2-phase simulated object of size $128 \times 128 \times 2040$. Here we have acquired 15% of available measurements.

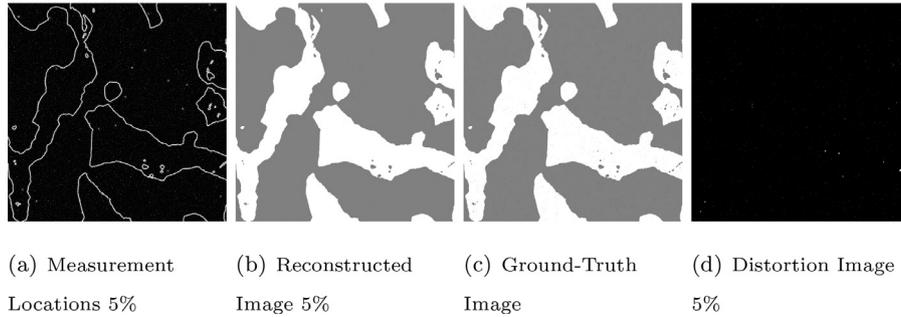


Fig. 6. Results of EDS-SLADS experiment performed on the 2-phase simulated object of size $1024 \times 1024 \times 2040$. Here we have acquired 5% of available measurements.

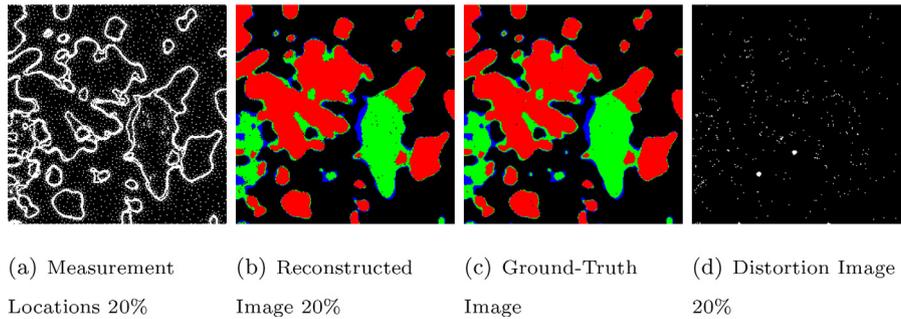


Fig. 7. Results of EDS-SLADS experiment performed on the 4-phase simulated object of size $256 \times 256 \times 2040$. Here we have acquired 20% of available measurements.

images. Then we used a simple thresholding scheme to segment all the SEM images. Then we added noise only to the testing images. The images we used for testing and training are shown in Fig. 3.

To train and validate the neural networks we acquired 24 spectra from each phase. To create the simulated testing object we acquired and used 12 (different) spectra for each phase. The noise added to the spectra while creating the simulated object is Poisson noise with $\lambda = 2$. The ill-spectrum we generated were also Poisson random vectors, with independent elements and $\lambda = 20$.

The results after 15% of samples were collected from the 128×128 image is shown in Fig. 5. The TD with 15% of samples was 0.0015. From this figure it is clear that we can achieve near perfect reconstruction with just 15% of samples. The misclassification rate of the detection and classification system was computed to be 0.0002, which tells us that the detection and classification system is also very accurate.

The results after 5% of samples were collected from the 1024×1024 image is shown in Fig. 6. The TD with 5% of samples was 0.0013. Here we see that even for the same object, if we sample at a higher spatial resolution we can achieve similar results with just 5% of measurements. In this experiment the misclassification rate of the detection and classification system was computed to be 0.

3.2.2. Experiment on simulated 4-phase object

For this experiment we will again sample two simulated objects, one with dimensions $256 \times 256 \times p$, and the other with dimensions $1024 \times 1024 \times p$ created using spectra and SEM images from a micro-powder mixture with 4 phases i.e. CaO, LaO, Si and C.

The testing and training images, shown in Fig. 4 were created in exactly same manner as in the previous experiment. The testing object was again created with 12 spectra from each phase. The neural networks were also trained validated just as before once more using 24 spectra from each phase.

The results after 20% of samples were collected from the 256×256 image is shown in Fig. 7. The TD with 20% of samples was 0.006. Again we see that we can achieve near perfect reconstruction with 20% of samples. The misclassification rate of the detection and classification system was computed to be 0.005, which again tells us that the detection and classification system is very accurate. However, we do note that this is not as accurate as in the 2-phase case.

The results after 5% of samples were collected from the 1024×1024 image is shown in Fig. 8. The TD with 5% of samples was 0.02. In this experiment the misclassification rate of the detection and classification system was computed to be 0.0009.

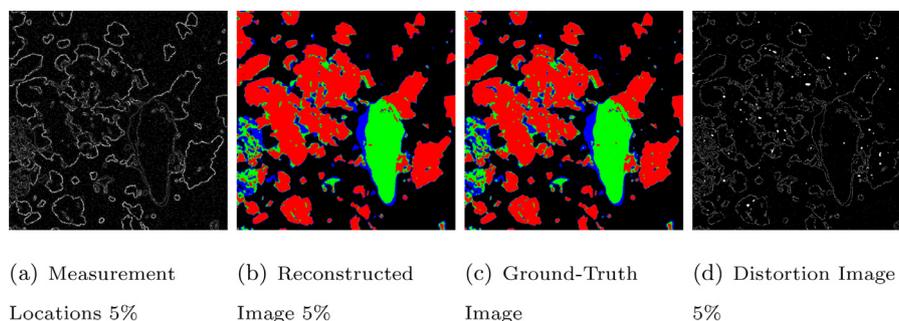


Fig. 8. Results of EDS-SLADS experiment performed on the 4-phase simulated object of size $1024 \times 1024 \times 2040$. Here we have acquired 5% of available measurements.

4. Discussion

It is clear that by using SLADS to determine the sampling locations, we can reduce the overall exposure to anywhere between 5–20%, and still obtain a near perfect reconstruction. These results also show that the SLADS method is better suited for higher pixel resolution mapping which maximizes the resolution capability of the instrument and detector. This method would be useful for investigation of biological and beam-sensitive samples, such as live cell imaging, as well as for high-throughput imaging of large samples, such as fabrication by additive manufacturing and defects metrology in chemical and structural study.

5. Conclusion

In conclusion, we have shown that integrating dynamic sampling (SLADS) with EDS classification (CNNs) offers significant advantage in terms of dose reduction and the overall data acquisition time. We have shown that when imaging at lower pixel resolution i.e. 128×128 or 256×256 we can achieve a high-fidelity reconstructions with approximately 20% samples and when imaging at higher pixel resolution i.e. 1024×1024 we can achieve a high-fidelity reconstruction with just 5% samples. We have also shown that our classification algorithm performs remarkably well in all the experiments. For future work, we will expand our EDS training database by including analytically simulated EDS data for more commonly used elements. We will use pure simulated EDS data to train the classification system, which enables more automated, high-throughput acquisition and characterization across different microscopic and spectroscopic platforms.

Acknowledgment

This material is based upon work supported by Laboratory Directed Research and Development (LDRD) funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under contract no. DE-AC02-06CH11357.

References

- [1] J. Goldstein, D.E. Newbury, P. Echlin, D.C. Joy, A.D. Romig Jr, C.E. Lyman, C. Fiori, E. Lifshin, *Scanning Electron Microscopy and X-ray Microanalysis: A Text for Biologists, Materials Scientists, and Geologists*, Springer Science & Business Media, 2012.
- [2] A.J. D'Alfonso, B. Freitag, D. Klenov, L.J. Allen, Atomic-resolution chemical mapping using energy-dispersive x-ray spectroscopy, *Phys. Rev. B* 81 (2010) 100101, doi:10.1103/PhysRevB.81.100101.
- [3] T.C. Lovejoy, Q.M. Ramasse, M. Falke, A. Kaeppl, R. Terborg, R. Zan, N. Dellby, O.L. Krivanek, Single atom identification by energy dispersive x-ray spectroscopy, *Appl. Phys. Lett.* 100 (15) (2012) 154101, doi:10.1063/1.3701598.
- [4] H.S. Anderson, J. Ilic-Helms, B. Rohrer, J. Wheeler, K. Larson, *Sparse imaging for fast electron microscopy*, in: IS&T/SPIE Electronic Imaging, 2013, p. 86570C.
- [5] R. Ohbuchi, M. Aono, Quasi-monte carlo rendering with adaptive sampling (1996).
- [6] K. Mueller, Selection of optimal views for computed tomography reconstruction, 2011.
- [7] Z. Wang, G.R. Arce, Variable density compressed image sampling, *Image Process. IEEE Trans.* 19 (1) (2010) 264–270.
- [8] M.W. Seeger, H. Nickisch, Compressed sensing and Bayesian experimental design, in: *Proceedings of the 25th International Conference on Machine Learning*, ACM, 2008, pp. 912–919.
- [9] W.R. Carson, M. Chen, M.R.D. Rodrigues, R. Calderbank, L. Carin, Communications-inspired projection design with application to compressive sensing, *SIAM J. Imaging Sci.* 5 (4) (2012) 1185–1212.
- [10] M. Seeger, H. Nickisch, R. Pohmann, B. Schölkopf, Optimization of k-space trajectories for compressed sensing by Bayesian experimental design, *Magn. Reson. Med.* 63 (1) (2010) 116–126.
- [11] K. Joost Batenburg, W.J. Palenstijn, P. Balázs, J. Sijbers, Dynamic angle selection in binary tomography, *Comput. Vision Image Understanding* (2012).
- [12] J. Vanlier, C.A. Tiemann, P.A.J. Hilbers, N.A.W. van Riel, A Bayesian approach to targeted experiment design, *Bioinformatics* 28 (8) (2012) 1136–1142.
- [13] T.E. Merryman, J. Kovacevic, An adaptive multirate algorithm for acquisition of fluorescence microscopy data sets, *IEEE Trans. Image Process.* 14 (9) (2005) 1246–1253.
- [14] G.M.D. Godaliyadda, G.T. Buzzard, C.A. Bouman, A model-based framework for fast dynamic image sampling, in: *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, 2014, pp. 1822–1826.
- [15] G.D. Godaliyadda, D. Hye Ye, M.A. Uchic, M.A. Groeber, G.T. Buzzard, C.A. Bouman, A supervised learning approach for dynamic sampling, *IS&T Imaging*, International Society for Optics and Photonics, 2016.
- [16] D.F. Specht, A general regression neural network, *IEEE Trans. Neural Netw.* 2 (6) (1991) 568–576.
- [17] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, *Neural Process. Lett.* 9 (3) (1999) 293–300.
- [18] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science* 313 (5786) (2006) 504–507.
- [19] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [20] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* (2012) 1097–1105.
- [21] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2, IEEE, 2006, pp. 1735–1742.
- [22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [23] K.-L. Hua, C.-H. Hsu, S.C. Hidayati, W.-H. Cheng, Y.-J. Chen, Computer-aided classification of lung nodules on computed tomography images via deep learning technique., *Oncotargets Ther.* 8 (2014) 2015–2022.
- [24] G. Wang, A perspective on deep imaging, *IEEE Access* 4 (2016) 8914–8924.
- [25] Y. Park, M. Kellis, Deep learning for regulatory genomics, *Nat. Biotechnol.* 33 (8) (2015) 825–826.
- [26] J. Zhou, O. Troyanskaya, Deep supervised and convolutional generative stochastic network for protein secondary structure prediction, in: *International Conference on Machine Learning*, 2014, pp. 745–753.
- [27] G.D. Godaliyadda, D. Hye Ye, M.A. Uchic, M.A. Groeber, G.T. Buzzard, C.A. Bouman, A framework for dynamic image sampling based on supervised learning (slads), *ARXIV*, 2017.
- [28] N.M. Scarborough, G.M.D.P. Godaliyadda, D.H. Ye, D.J. Kissick, S. Zhang, J.A. Newman, M.J. Sheedlo, A.U. Chowdhury, R.F. Fischetti, C. Das, G.T. Buzzard, C.A. Bouman, G.J. Simpson, Dynamic x-ray diffraction sampling for protein crystal positioning, *J. Synchrotron Radiat.* 24 (1) (2017) 188–195, doi:10.1107/S160057751601612X.
- [29] F. de la Peña, P. Burdet, T. Ostasevicius, M. Sarahan, M. Nord, V.T. Fauske, J. Taillon, A. Eljarrat, S. Mazzucco, G. Donval, et al., Hyperspy: hyperspy 0.8, 2015.
- [30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).
- [31] D. Rowenhorst, J. Kuang, K. Thornton, P. Voorhees, Three-dimensional analysis of particle coarsening in high volume fraction solid-liquid mixtures, *Acta Mater.* 54 (8) (2006) 2027–2039.