Using Supervised Learning for Strain Analysis of 4DSTEM Datasets

Miti Shah

Abstract—Understanding and visualizing strain in materials is important for determining sources of failure, analyzing material properties, and improving semiconductor design. Transmission electron microscopy (TEM) allows for high resolution imaging of these devices, and recent algorithmic and instrument improvements allow for improved strain mapping. However, existing methods face limitations in precision and resolution due to noise and distortion in 4DSTEM datasets. Cross-correlation-based Bragg disk detection methods are susceptible to intensity fluctuations and Poisson noise, while cepstral transform approaches struggle with sub-pixel peak localization accuracy required for precise strain measurements. Recent machine learning approaches have shown promise in reducing noise in diffraction patterns, but focus on reconstructing dynamical diffraction patterns rather than directly facilitating strain analysis. In this work, we propose a supervised learning approach using a U-Net architecture that takes both magnitude and phase of Fourier-transformed convergent beam electron diffraction (CBED) patterns as input to predict atomic position maps. We introduce a custom loss function combining binary cross-entropy and mean squared error to balance peak localization with intensity preservation. On simulated datasets, Our method demonstrates accurate peak position prediction with low absolute error. However, translation to experimental datasets reveals challenges in maintaining sub-pixel accuracy needed for reliable strain mapping, highlighting the need for more diverse training data and refinement of the peak detection pipeline. This work provides insights into the challenges of applying supervised learning to experimental 4DSTEM strain analysis and identifies specific directions for future improvement.

Index Terms—Computational Imaging, Machine Learning, Strain, Transmission Electron Microscopy

1 Introduction

Transmission electron microscopy (TEM) is a technique that allows atomic resolution imaging of materials and devices. 4D scanning transmission electron microscopy (4DSTEM) uses a 2D pixel grid sensor to take a convergent beam electron diffraction pattern (CBED) at each pixel position in a scanned grid. Figure 1 illustrates the scanning structure of the microscope and the resulting grid of CBED patterns. This material characterization technique gives local crystallographic information of the material and can be used to reconstruct real-space images of strain, crystal orientation, or crystal structure [1]. The ability to map these properties at high spatial resolution is crucial for understanding material behavior. For instance, strain mapping in semiconductor devices shows how interface quality impacts device performance, while in battery materials, local strain fields can predict degradation pathways and cycle life. Despite these capabilities, improved algorithms are required to analyze 4DSTEM datasets in order to extract this information accurately and efficiently.

Strain analysis can be conducted by measuring sub-pixel shifts in Bragg disk positions (the bright disks in the CBED patterns) for each pixel position from a given reference using cross-correlation. Specifically, for each pattern the basis vectors are determined using the disk positions. The change in these basis vectors from the reference gives the strain for that pixel position. However, obtaining accurate strain maps is complicated by multiple sources of noise and distortion in the input data. Most CBED patterns are susceptible to Poisson noise inherent to electron microscopy, Gaussian noise, and intensity fluctuations due to sample thickness variations and local tilt [1]. These effects make it difficult to precisely determine peak positions, which in turn limits the accuracy of the resulting strain maps. Since strain measurements often require sub-pixel precision, even small errors in peak detection can translate to significant artifacts in the final strain analysis. In this work, we develop a supervised learning approach that uses both magnitude and phase of Fourier-transformed CBED patterns to predict atomic position maps for strain analysis. We demonstrate accurate results on simulated datasets and identify key challenges in extending the approach to experimental data.

2 Related Work

Various algorithms have been developed to calculate strain maps from 4DSTEM datasets, each with distinct advantages and limitations.

The bragg disk detection method uses bragg disks in the CBED pattern to determine the lattice vectors. The positions of each Bragg disk (the circular bright spots in the CBED pattern) are identified using cross correlation using a bragg disk in measurement in vacuum or a template bragg disk. Using these Bragg disk positions, the lattice vector is determined and refined using an optimization procedure based on Bragg disk indexing. This process is repeated for each CBED pattern at each pixel where a measurement was taken in the grid. These lattice vectors are then compared to a predetermined unstrained reference in order to calculate the magnitude of strain at each pixel position, which in turn gives the strain map [2]. Under ideal conditions, this method can provide strain maps with precision on the order

• *M. Shah is with the Department of Electrical Engineering, Stanford University, Stanford, CA.*
E-mail: mitis@stanford.edu.

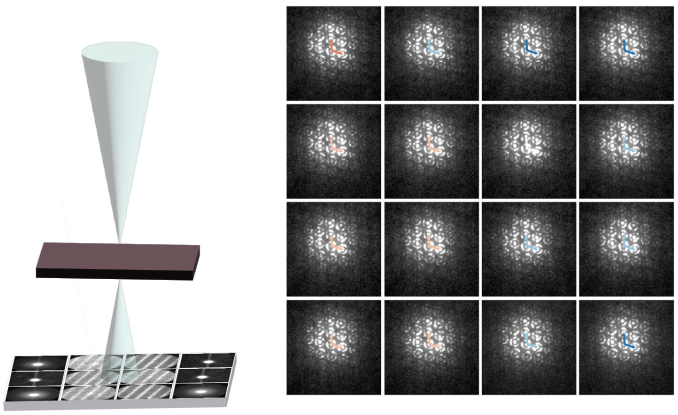

Fig. 1. Example diagram of 4DSTEM experiment and resulting dataset. Left: Incoming electron beam onto sample (Brown square) gives a diffraction pattern at each pixel. Right: Resulting dataset showing the changes in bragg disk spacing using arrows.

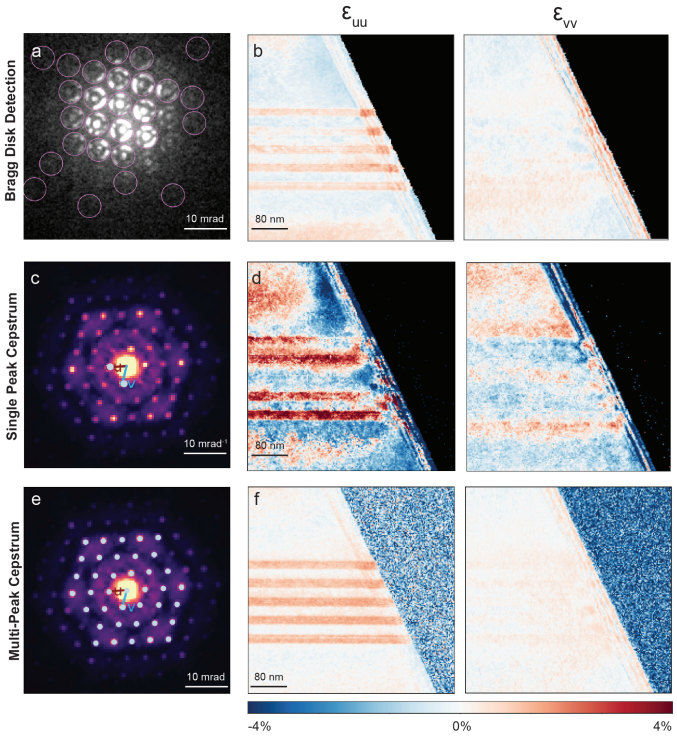

Fig. 2. Experimental results of strain maps generated using current algorithms. Dataset was obtained from an SiGe crystal [4] a) Bragg disk detection method showing the disks used to map lattice spacing to. b) Strain map generated using Bragg disk detection. c) Cepstral transform method showing how the lattice vectors can be obtained using two peaks in the cepstral transform. c) Strain maps generated using the two peak vector refinement from the cepstral transform. d) Cepstral transform method with multi peak refinement of the lattice vectors. e) Strain maps generated using the multi-peak cepstral transform method.

of 0.1 % strain. However, the method faces significant challenges with accurate disk detection. Since cross-correlation is the primary mechanism for identifying disk positions, the results are highly susceptible to errors from intensity fluctuation, Poisson noise, and Gaussian noise. Intensity fluctuations pose the largest problem because they arise from unavoidable physical effects, primarily sample thickness variations and local crystal tilt, which cause individual disks to vary significantly in brightness and shape. When a disk's appearance deviates substantially from the template, cross-correlation fails to reliably identify its position, leading to either missing disks or incorrectly localized peaks.

To address the cross-correlation instabilities inherent in Bragg disk methods, an alternative approach uses the cepstral transform. The cepstral transform is the logarithm of the magnitude of the Fourier transform of the CBED patterns: $CepstralTransform = log(|\mathcal{F}(I_{CBED})|)$. Since the CBED pattern itself is the Fourier transform of the real-space structure, the cepstral transform effectively returns to real space and directly gives the atomic lattice positions. Therefore, identifying peak positions in the cepstral transform allows for strain calculation similar to the Bragg disk detection method, but operating in real space rather than diffraction space. This method reduces the instability caused by cross correlation of Bragg disks since the problem is simplified to peak detection of the cepstral transform [3]. However, it still has sources of error since the peak detection must have sub-pixel accuracy in order to obtain sufficiently precise lattice vectors for accurate strain mapping. Achieving this sub-pixel accuracy is difficult because the peaks in the cepstral transform are broadened by noise in the CBED pattern, making exact positioning challenging. To address this, a multi-peak refinement approach allows for more accurate lattice vector determination by simultaneously fitting multiple peaks and enforcing indexing constraints. Another improvement to the algorithm is to use the square root instead of logarithm to calculate the cepstral transform: $CepstralTransform = \sqrt{(|\mathcal{F}(I_{CBED})|)}$. The use of square root better preserves the relative intensities of weak and strong peaks and reduces numerical instabilities from taking the log of near-zero values improving peak visibility for detection. Similar to the Bragg disk detection method, once the lattice vectors are found for each pixel position, a strain map can be calculated by comparing the measured vectors to a reference lattice vector from an unstrained region.

Figure 2 shows the strain maps obtained from these algorithms for an experimental dataset of SiGe. The expected strain map should show bands of strain along one lattice vector direction, and a relatively uniform map along the orthogonal vector. The quality of each algorithm can be assessed by the visibility of these bands and the absence of sharp gradients between adjacent pixels, which are unrealistic and due to algorithmic failure.

Machine Learning Approaches. More recently, machine learning has been applied to improve strain analysis from 4DSTEM datasets. Munshi et al. [5] developed a neural network trained to reduce noise and simplify CBED patterns, making subsequent strain analysis more robust. Their approach was successful in reducing the main sources of error associated with Bragg disk detection, including noise and intensity variations. The network was trained on datasets where the inputs were various CBED patterns and the outputs were clean maps of disk positions including information about their structure factors. This greatly simplifies peak detection, allowing for more accurate strain reconstruction since there is less error in the lattice vector determination. Their results showed strong performance on

simulated datasets compared to traditional cross-correlation methods. However, the algorithm focused primarily on generating cleaned dynamical diffraction patterns (denoised versions of the original CBED patterns) and were not robust to any experimental datasets that did not have the appropriate preprocessing. We believe that working directly with the cepstral transform or a related real-space representation will prove more effective for strain mapping applications, since it more directly targets the quantity of interest (atomic positions) and avoids the intermediate step of interpreting diffraction patterns.

3 Theory and Proposed Method

We propose our supervised learning approach for strain calculation from 4DSTEM datasets. The method builds on existing computational techniques while introducing a neural network architecture designed to directly predict atomic position maps from CBED patterns.

3.1 Data Generation

The datasets used to train and validate this algorithm must be generated in a way that accurately reflects the physics of electron microscopy imaging. In this project, I used abTEM to generate the input and label pairs used to train the network [6]. To start, I selected several representative materials from the MaterialsProject database, choosing structures that span common crystal symmetries (FCC, BCC, and hexagonal) and include both single-element and compound materials. For each base structure, I generated multiple variants with different thicknesses ranging from 5-50 nm and rotations varying by 5-degree increments around major zone axes. These variations were important because the CBED pattern appearance changes significantly with thickness and orientation, and the network needs to be robust to these variations. Using the abTEM software, I generated CBED patterns for each configuration along with the corresponding labels. The CBED patterns generated at this stage do not have noise or other augmentations, those are added later during training. The current dataset size used for algorithm testing is 10,000 different patterns, though this is relatively small compared to typical deep learning datasets and likely contributes to the generalization challenges discussed later. Algorithm 21 shows the general structure of the data generation process using functions from abTEM.

In this algorithm the multislice propagation (line 9), simulates the electron beam passing through the sample and scattering from atoms, and the autocorrelation calculation (line 11), generates the label representing atomic positions in real space. The frozen phonon configurations (line 15) account for thermal vibrations of atoms, which is important for generating realistic diffraction patterns that match experimental data.

3.2 ML Model

Following data generation, the CBED patterns undergo augmentation and preprocessing before being input into the neural network. The model architecture and loss function were designed based on the specific requirements of strain mapping applications.

Algorithm 1 Generate Input-Label Pairs

```
1:  Input: Crystal structure, rotation, thickness
2:  Output: Diffraction patterns, labels, lattice vectors
3:
4:  Build crystal cell with given rotation and dimensions
5:  Create atomic potential with thermal vibrations
6:
7:  Define aperture (circular or bullseye pattern)
8:  Generate electron probe
9:
10: Simulate electron multislice propagation
11: Compute diffraction patterns from exit waves
12: Compute autocorrelation of projected potential
13:
14: Crop all arrays to central region
15: Normalize label
16:
17: for each frozen phonon configuration do
18:     Store diffraction pattern, label, lattice vectors, probe
19: end for
20:
21: return diffraction patterns, labels, lattice vectors, probes
```

3.2.1 *Data Preprocessing*

Before being input into the model, the data is processed to match the model requirements and improve robustness. The specific augmentations applied and their order were determined based on what is common in electron microscopy. First, a Gaussian background and DC offset of random magnitude are applied to simulate inelastic scattering contributions and camera error. Then the image is shifted and distorted with ellipticity to replicate lens aberrations and sample drift that occur during experimental acquisition. The noise is added last, in the form of shot noise (Poisson noise from discrete electron counting), salt and pepper noise (from detector artifacts), and Gaussian blur (from the point spread function of the detector). Following these augmentations, a Hanning window is applied to the CBED pattern to ensure the edges go to zero. Then the shifted Fourier transform is taken and the magnitude and phase of the result make up the two input channels that go into the algorithm. Algorithm **??** summarizes the overall structure of the data augmentation pipeline that occurs before an image is input into the model.

Figure 3 shows image outputs of the data preprocessing pipeline prior to inputting data for training. It starts with the original CBED pattern simulated using abTEM and its augmented version shown in Figure 3a,b. The amplitude and phase of the augmented image's Fourier transform are shown in Figure 3c,d. The label, or target output of the algorithm is shown in Figure 3e,f. In the label, the peaks indicate the atom positions, and therefore can be used to determine the lattice vectors. This is demonstrated in the zoomed-in version of the label (Fig. 3f) where the points indicating lattice vectors for the given crystal are highlighted in blue. It can be seen that the blue points coincide with the peaks in the label. The variation in intensity in the label occurs due to differences in atomic number of the different atoms in the

Algorithm 2 Image Augmentation Pipeline

Input: Diffraction pattern I
Output: Augmented diffraction pattern I'

$I' \leftarrow I$

if background augmentation enabled **then**
  $I' \leftarrow \text{AddBackground}(I')$
end if

if geometric augmentation enabled **then**
  $I' \leftarrow \text{ApplyElasticTransform}(I')$
  $I' \leftarrow \text{ApplyScaleTransform}(I')$
  $I' \leftarrow \text{ApplyShiftTransform}(I')$
end if

if shot noise enabled **then**
  $I' \leftarrow \text{AddPoissonNoise}(I')$
end if

if blur enabled **then**
  $I' \leftarrow \text{ApplyBlur}(I')$
end if

if salt and pepper noise enabled **then**
  $I' \leftarrow \text{AddSaltPepperNoise}(I')$
end if

$I' \leftarrow W_{\text{Hann}}(I')$ {Setting edges to 0}
$\mathcal{F} \leftarrow \text{fftshift}(\text{fft2d}(I'))$
$X \leftarrow [|\mathcal{F}|, \angle(\mathcal{F})]$

return I'

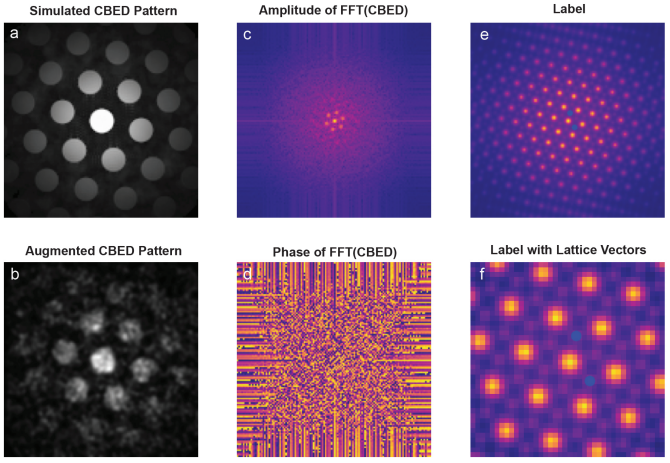

Fig. 3. The various datasets in the Data preprocessing pipeline. a) Original CBED pattern simulated with abTEM. b) Augmented CBED pattern adding shift, noise and distortion. c) Amplitude of the Fourier transform of the augmented CBED pattern which makes up one of the two channel inputs. d) Phase of the Fourier transform of the augmented CBED pattern which is the second of the two channel input. e) Expected output label of the model. f) Zoomed in version of the output label with the expected lattice vectors outlined by blue points from the center.

crystal as well as the structure factor of the crystal.

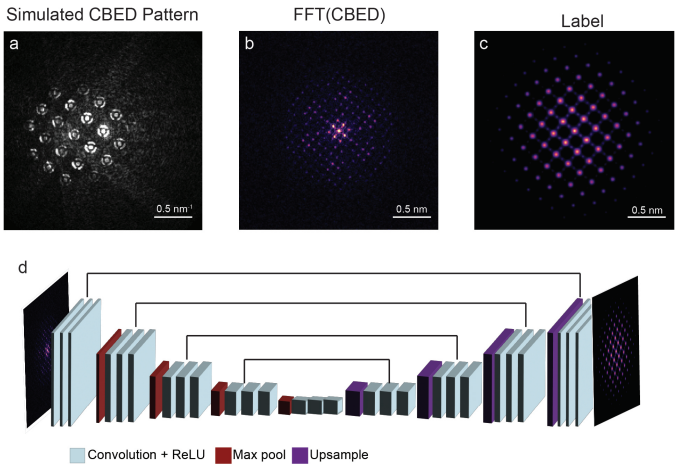

Fig. 4. Diagram of model architecture. a) Example augmented CBED pattern. b) Amplitude of the fourier transform of the CBED pattern. c) Expected label which is a simulated representation of the fourier transform of the CBED pattern. d) Current model architecture.

3.2.2 *Model Architecture and Loss Function*

The model uses a U-Net architecture to ensure a large receptive field, which is necessary because much of the information relevant to accurate peak localization is spread across the entire diffraction pattern rather than confined to a few pixels near each peak. Specifically, the spacing and arrangement of Bragg disks in the CBED pattern directly determines the real-space lattice parameters. Therefore, in order to build an accurate representation that combines these features, the receptive field must be large enough to encompass multiple neighboring peaks simultaneously. Figure 4 shows the current model architecture. It is a four layer CNN with three filters per layer. The reason for this is, it ensures that the UNet is not shallow while allowing for some feature extraction such as peak locations.

Another important part of the model architecture is the loss function and optimizer choice. Originally, the loss function was set to a L2 loss loss since it greatly penalizes error. However, using this loss resulted in the model failing to find peak positions which was crucial for strain mapping. With L2 loss, the model generally output a uniform intensity map which minimized error but did not provide any peak information. Switching to a binary cross entropy (BCE) loss allowed for improved peak localization but looses intensity information since it only allows binary inputs. In the end, a custom loss function combining both approaches allows for sharp peak localization while preserving the relative intensity information that reflects atomic structure. Equation 1 shows the current loss function used for the algorithm.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}}(b_p, b_t) + (\mathcal{L}_{\text{MSE}}(p, t))^{\gamma} \quad (1)$$

In equation 1, p and t denote the predicted and target values respectively, and the binary masks are defined as:

$$b_p = PredictedImage_{p>0}, \quad b_t = TargetImage_{t>0} \quad (2)$$

Here, $\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy loss, $\mathcal{L}_{\text{MSE}}$ is the mean squared error loss, and γ is a hyperparameter which tunes the importance of peak position over accurate intensity mapping. The current gamma value is set at 0.1.

Equations 3 and 4 show the full form of the BCE and MSE loss.

$$\mathcal{L}_{\text{BCE}}(b_p, b_t) = -\frac{1}{N}\sum_{i=1}^{N}[b_{t,i}\log(b_{p,i}) + (1-b_{t,i})\log(1-b_{p,i})] \tag{3}$$

$$\mathcal{L}_{\text{MSE}}(p, t) = \frac{1}{N}\sum_{i=1}^{N}(p_i - t_i)^2 \tag{4}$$

For the optimizer, AdamW was used rather than standard Adam. AdamW improves generalization to data outside the training distribution which is important given the goal of applying the trained network to experimental datasets that differ from the simulated training data.

4 Experimental Results

The described algorithm was tested on both simulated and experimental datasets to evaluate its performance and identify limitations.

The simulated test dataset was generated in the same manner as the training dataset but using different material structures and orientations to ensure the network was evaluated on new data. Using crystals that share structural properties with those in the training dataset, CBED patterns were generated and augmented. The Fourier transform of each augmented image was split into its phase and amplitude components and input into the trained network.

Figure 5 shows representative results from this test of the model. The first column shows the amplitude channel of the test inputs. The second column shows the target model output, which was generated with the input. The third column shows the predicted model output. The final column shows the difference between the target and predicted outputs, which is important for checking whether peak positions match. A difference value of zero at a peak location means the predicted peak position matches the target, while non-zero values indicate a peak offset between the target and predicted patterns. Using the difference matrices, the absolute error was calculated between the target and predicted model outputs for each test image according to Equation 5, and Table 1 summarizes these results.

$$\text{Error}_{\text{absolute}} = |(p - t)| \tag{5}$$

Dataset Number	Absolute Error
Dataset 1	30.637291
Dataset 2	43.568565
Dataset 3	78.75087
Dataset 4	75.59518

TABLE 1
List of absolute error values for ML model output of simulated data corresponding to Figure 5.

In addition to testing the model on simulated data, an experimental dataset was used to determine whether the approach could produce valid strain maps from real microscopy data. The experimental dataset used is the same

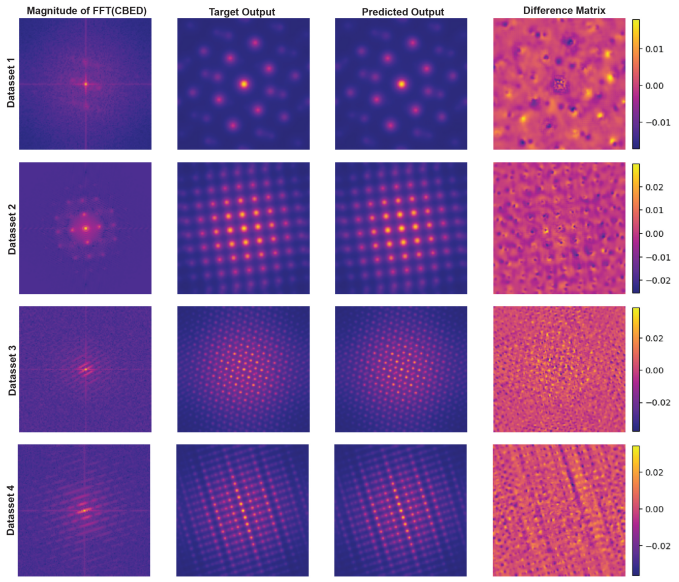

Fig. 5. ML algorithm output for simulated example datasets. The first column shows the channel 1 input of the algorithm (the magnitude of the Fourier transform of the CBED pattern). The second column shows the target output for the given input dataset. The third column is the actual output of the machine learning algorithm. The fourth column shows the difference between the predicted and target output. The color-bar corresponds to the difference matrix.

SiGe sample shown in Figure 2 for comparison with existing methods. For each CBED pattern in the experimental dataset, the Fourier transform was computed and split into phase and amplitude channels to input into the model. The resulting model outputs were then used to calculate lattice vectors at each pixel position, which were compared to a reference unstrained region to generate the strain map. Figure 6a shows a few example model outputs from different pixel positions in the experimental dataset. The left example shows a relatively clean output with well-defined peaks, similar to what is seen in simulated data. However, the right example shows a noisy output with poorly defined peaks. Figure 6b shows the strain maps calculated based on the model outputs across the entire scanned region.

The results from the experimental dataset can be compared with the current methods. Using the multi-peak cepstral as the standard and checking the other methods against it, the absolute error from each strain map to the multi-peak cepstral can be calculated. Table 2 summarizes the absolute error from each method when compared to the strain from multi-peak cepstral.

Method	Absolute Error euu	Absolute Error evv
Single Peak Cepstral	78.072	75.211
Bragg Disk Detection	49.477	19.756
ML Model	177.919	100.388

TABLE 2
List of absolute error values comparing strain maps using different methods on experimental datasets.

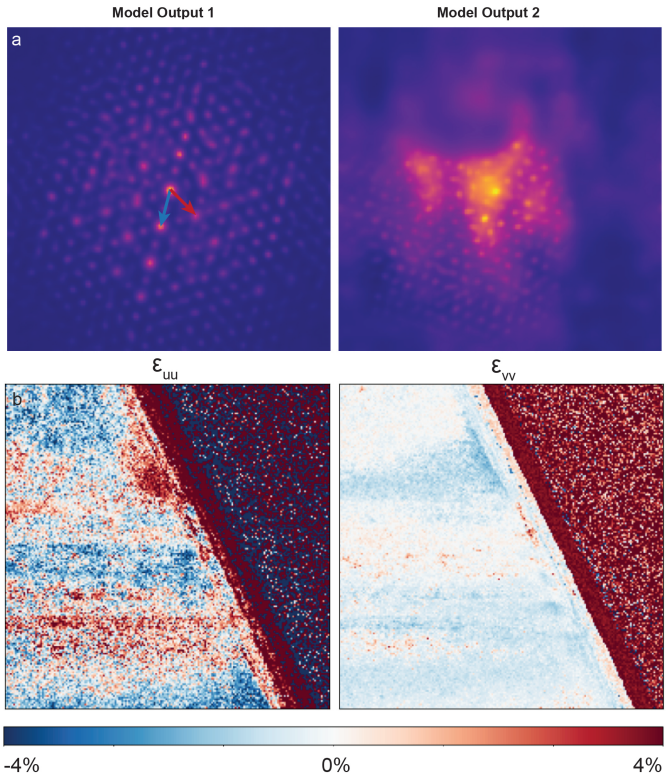

Fig. 6. ML algorithm output for the experimental dataset. a) Example model outputs from the experimental datasets used to determine lattice vectors at that pixel position. b) Strain maps calculated based on the model output images.

4.1 Discussion

The results demonstrate different levels of success for simulated versus experimental data, revealing important limitations of the current approach.

It can be seen the model output matches the input for the simulated datasets. The difference matrix and error are low even though it was trained for 150 epochs and with a relatively smaller dataset. The difference matrices help indicate if the peaks are predicted to be in the right position. In Figure 5, it can be seen that the difference matrices are close to zero at the positions of the peaks. This indicates that the predicted peak positions are the same as the target peak positions. However, for Datasets 3 and 4 where there are multiple close peaks, the exact positioning of the peaks is less accurate as shown by the fewer zero points in the difference matrix. This discrepancy is similarly seen in table where the absolute error is significantly higher for Datasets 3 and 4. The reason for this difference could be when many peaks are present, the network must localize all of them accurately, which is a much harder task and may require a more complex network.

However, when translated to experimental dataset, it is harder to discern the accuracy of the label since there is no precise way to verify against an target output. While some individual model outputs (shown in Figure 6a) appear reasonable, when these outputs are used to calculate strain maps, the results (shown in Figure 6b) are much noisier than those produced by existing methods (shown in Figure 2). The strain maps lack the stripes of strain expected, and instead have random variations that are physically unrealistic. The error in Table 2 confirms this since the error for the ML model is significantly greater than the other two current methods.

There are several potential causes for this. First, the current peak-finding algorithm used in post-processing to extract lattice vectors from the model output may not be accurate enough to generate strain maps. Even if the model output is qualitatively correct,errors in peak localization would translate to strain errors, which would appear as the noise seen in the strain maps. Second, while the model outputs look similar to the training labels, the peaks may not be positioned correctly at the sub-pixel level required for strain analysis. Third, the simulated training data might not fully capture all the variations and artifacts present in experimental data. Real CBED patterns contain many distortions from lens aberrations, sample contamination, and detector nonlinearities that are not included in the current augmentation pipeline.

Several improvements could address these limitations. Generating a larger and more diverse training dataset—including more materials, orientations, and particularly more examples of complex multi-peak patterns—would likely improve performance on both simulated and experimental data. Incorporating experimental CBED patterns into the training data, even without fully accurate labels could help the network learn to handle real experimental artifacts. Finally, developing a more sophisticated peak-finding post-processing algorithm would help improve strain maps. Alternatively, the network could be modified to directly output lattice vectors rather than position maps, potentially improving accuracy. Another network modification could be to have the entire 4DSTEM dataset as an input to the algorithm. Since each pattern is inherently related in structure, having the entire dataset as the input might allow the algorithm to better learn the patterns between them.

5 Conclusion

In this paper, I presented a supervised learning approach for generating atomic position maps from convergent beam electron diffraction patterns in 4DSTEM datasets. The purpose of this algorithm is to enable improved and potentially faster strain mapping compared to existing methods like cross-correlation-based Bragg disk detection and the cepstral transform.

The method uses a U-Net architecture with the phase and amplitude of the Fourier transform of CBED patterns as the two input cahnnels, and atomic position maps as the output. A custom loss function combining binary cross-entropy and mean squared error was developed to balance peak localization accuracy with intensity maps. Training data was generated using physics-based multislice electron scattering simulations with realistic augmentations to match experimental noise and distortions.

The results showed that the algorithm works well for simulated datasets; however, it is not as accurate for patterns with many closely-spaced peaks. Additionally, when used on experimental datasets, the strain maps are significantly noisier than those produced by existing methods.

Moving forward, the model can be improved by generating larger and more diverse training datasets, potentially incorporating real experimental data, and either improving the peak-finding post-processing or redesigning the network to directly output lattice vectors. Additionally, more sophisticated data augmentation that better captures the full range of experimental artifacts would likely improve generalization.

Acknowledgments

The author would like to thank Arthur McCray and Colin Ophus.

References

[1] C. Ophus, "Four-dimensional scanning transmission electron microscopy (4d-stem): From scanning nanodiffraction to ptychography and beyond," *Microscopy and Microanalysis*, vol. 25, no. 3, pp. 563–582, 2019.

[2] T. C. Pekin, C. Gammer, J. Ciston, A. M. Minor, and C. Ophus, "Optimizing disk registration algorithms for nanobeam electron diffraction strain mapping," *Ultramicroscopy*, vol. 176, pp. 170–176, 2017.

[3] E. Padgett, M. E. Holtz, P. Cueva, Y.-T. Shao, E. Langenberg, D. A. Muller, and J. E. Spanier, "The exit-wave power-cepstrum transform for scanning nanobeam electron diffraction: robust strain mapping at subnanometer resolution and subpicometer precision," *Ultramicroscopy*, vol. 214, p. 112994, 2020.

[4] S. M. Ribet, T. Gebhard, B. D. Thackray, E. Hoffmann, T. J. Pennycook, V. P. Dravid, D. G. Stroppa, and K. A. Paton, "Multi-angle precession electron diffraction," *Microscopy and Microanalysis*, vol. 31, no. 6, p. ozaf103, 2025.

[5] J. Munshi, A. Mazilkin, V. Uglov, D. Krapf, C. Ophus, B. Sarac, M. Herbig, P. Kumar, D. Raabe, and B. Gault, "Disentangling multiple scattering with deep learning: application to strain mapping from electron diffraction patterns," *npj Computational Materials*, vol. 8, no. 1, p. 254, 2022.

[6] J. Madsen and T. Susi, "The abtem code: transmission electron microscopy from first principles," *Microscopy and Microanalysis*, vol. 1, p. 24, 2021.