

PAPER • OPEN ACCESS

## Embedding human heuristics in machine-learning-enabled probe microscopy

To cite this article: Oliver M Gordon *et al* 2020 *Mach. Learn.: Sci. Technol.* **1** 015001

View the [article online](#) for updates and enhancements.

You may also like

- [Theoretical characterization of uncertainty in high-dimensional linear classification](#)  
Lucas Clarté, Bruno Loureiro, Florent Krzakala et al.
- [Data-driven discovery of Koopman eigenfunctions for control](#)  
Eurika Kaiser, J Nathan Kutz and Steven L Brunton
- [Spectrally adapted physics-informed neural networks for solving unbounded domain problems](#)  
Mingtao Xia, Lucas Böttcher and Tom Chou



## PAPER



## Embedding human heuristics in machine-learning-enabled probe microscopy

## OPEN ACCESS

RECEIVED  
31 July 2019REVISED  
5 September 2019ACCEPTED FOR PUBLICATION  
10 September 2019PUBLISHED  
4 February 2020

Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Oliver M Gordon<sup>1</sup> , Filipe L Q Junqueira and Philip J Moriarty 

School of Physics &amp; Astronomy, The University of Nottingham, University Park, Nottingham, NG7 2RD, United Kingdom

<sup>1</sup> Author to whom correspondence should be addressed.E-mail: [oliver.gordon@nottingham.ac.uk](mailto:oliver.gordon@nottingham.ac.uk), [filipe.junqueira@nottingham.ac.uk](mailto:filipe.junqueira@nottingham.ac.uk) and [philip.moriarty@nottingham.ac.uk](mailto:philip.moriarty@nottingham.ac.uk)**Keywords:** STM, SPM, automated STM, convolutional neural networks, real time machine learning, STM tip stateSupplementary material for this article is available [online](#)**Abstract**

Scanning probe microscopists generally do not rely on complete images to assess the quality of data acquired during a scan. Instead, assessments of the state of the tip apex, which not only determines the resolution in any scanning probe technique, but can also generate a wide array of frustrating artefacts, are carried out in real time on the basis of a few lines of an image (and, typically, their associated line profiles.) The very small number of machine learning approaches to probe microscopy published to date, however, involve classifications based on full images. Given that data acquisition is the most time-consuming task during routine tip conditioning, automated methods are thus currently extremely slow in comparison to the tried-and-trusted strategies and heuristics used routinely by probe microscopists. Here, we explore various strategies by which different STM image classes (arising from changes in the tip state) can be correctly identified from partial scans. By employing a secondary temporal network and a rolling window of a small group of individual scanlines, we find that tip assessment is possible with a small fraction of a complete image. We achieve this with little-to-no performance penalty—or, indeed, markedly improved performance in some cases—and introduce a protocol to detect the state of the tip apex in real time.

**1. Introduction**

One of the major challenges in the drive to fully automate the scanning probe microscope is the need to constantly maintain the integrity of the tip [1, 2]. During an experimental session, interactions with the surface can cause the tip to spontaneously and randomly change shape, modifying the interactions and therefore changing the data acquired in a highly nonlinear fashion. This frequently results in inconsistent scans containing visual artefacts, often making data unusable or, at best, problematic to interpret. Furthermore, it is becoming *de rigueur* in state-of-the-art SPM to functionalise tips by deliberately picking up adsorbed molecules or atoms from the surface [3], vastly improving resolution [4], enabling direct measurement of intermolecular pair potentials [5, 6], and/or modifying the capability of the probe, for better or worse, to manipulate and position single adsorbates [7].

Indeed, SPM experimentation is now at the point where not only is single atom/molecule termination of the tip apex required, but fine control and detailed understanding of its atomic/molecular orbital structure is often essential. Gross *et al* [8] provided a particularly elegant example of the importance of ‘orbital engineering’ of this type by demonstrating the significant enhancement of submolecular resolution in scanning tunnelling microscopy (STM) images of pentacene and naphthalocyanine molecules via tunnelling through *p*-wave orbitals, as the tunnelling matrix element for these states is proportional not to the sample wavefunction itself but its spatial derivatives. The spatial distribution and orientation of electron density at the tip apex also plays a central role in single atom manipulation [9]. Controlling and maintaining the atomistic and orbital structure of the tip apex is therefore a vital part of state-of-the-art SPM operation. Currently, this requires a protracted and

repetitive routine of voltage pulsing, ‘gentle’ (or not-so-gentle) indenting of the tip into the surface, scanning at relatively high voltages and currents, and/or attempts to pick up adsorbates. This is at present a high-effort, time-consuming and manual process involving only simple sub-processes, making it ideal to automate.

Whilst convolutional neural networks (CNNs) have been shown to be capable of assessing SPM tips [10–12], and, most recently, of extracting ‘hidden order’ from STM datasets [13], CNN methods to-date have been trained exclusively with complete images. Partial scans comprising a small number of scanlines therefore simply do not provide the information upon which the network mathematically depends and so current methods of CNN image assessment require complete scans. This method of CNN assessment *after complete* scans compares extremely poorly to human-based assessment, in which SPM operators routinely perform accurate assessment *during in-process* scans by observing individual line profiles as the image is acquired. Indeed, as little as 1%–2% of a full scan may be required to correctly assess a particularly poor tip. Furthermore, because the majority of time spent maintaining the tip is spent acquiring the data to assess, manual maintenance by a human is beyond an order-of-magnitude faster than any current CNN protocol. Given that manual maintenance can take several hours as-is, automated tip assessment with full-scan CNN protocols may be unable to keep up with the demands of SPM experimentalists unless an alternative strategy is introduced. In this paper we outline such a strategy and demonstrate that it performs extremely well against current methods based on complete images.

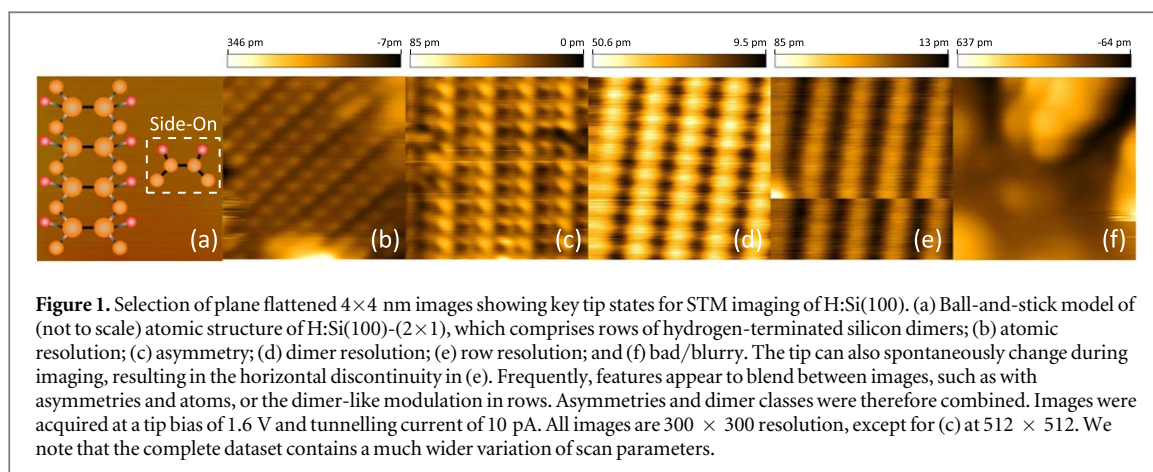
Furthermore, there is a wealth of data embedded in SPM scans, which can be exploited with neural network structures. For example, Burzawa *et al* [14] have very recently shown that single layer neural networks can be used to extract meaning in Ising model images, which are otherwise difficult for humans to interpret. A key difference in our case is that we focus on accurately observing atomic resolution, which is lost (or at best aliased) for larger scan areas. This is not the case for Burzawa *et al*’s work, where close to a critical point the correlation length becomes effectively ‘system spanning’. These patterns at criticality therefore are described by power law behaviour. Indeed, our group has previously examined this type of power law behaviour and the associated structural correlations for nanoparticle assemblies [15, 16].

Beginning with a dataset of 6167 scans of the H:Si(100) surface, we extend our previous study of CNN tip detection [12] to explore and compare a variety of methods by which the state of the tip can be determined using incomplete, partial scans. In addition to the simple, common method of ‘padding’ incomplete scan frames with an arbitrary marker value, we also discuss training the network to recognise individual linescans instead of entire images. Optimal performance is seen when classifying a ‘window’ consisting of a small group of linescans, and using a second temporal network to determine tip state as the window is ‘rolled’ over the course of a scan. This method remarkably produces better-than-complete-image performance with only a fraction of the data. By combining several of these methods in a ‘hybrid’ approach, it is possible to accurately assess scanning probe (in this case, STM) data by at least an order of magnitude faster than current CNN protocols [10–12].

## 2. H:Si(100) dataset

As discussed in the Introduction, SPM images often contain multiple features because of the scanning probe apex changing during a single scan. These tip changes also regularly and immediately result in discontinuities perpendicular to the direction of the scan. After the tip changes shape, multiple, more complex visual artefacts can also appear [17–19]. For example, features can appear to ‘ghost’ due to the presence of multiple tip apices [10, 20], or large blurs may appear due to impurities on the probe itself. Whilst these particular features can be seen when scanning any surface, others are specific to the surface being investigated [21]. For example, for the H:Si(100) surface, four different, distinct tip states of ‘individual atoms’ (for the sharpest tips), ‘dimers’, ‘asymmetries’, and ‘rows’ have been observed and discussed in the literature [18, 22, 23]. Typically, an operator will want to coerce the tip into producing images with one of these atomistic resolutions visible. (It is also worth noting that the tip apex capable of the highest resolution may not be best suited to other tasks, including, in particular, single atom manipulation [24].) Uncontrolled, and sometimes controlled, tip changes, however, mean that it is possible to produce images of H:Si(100) showing a combination of any of these four states, tip change shears, and other defects. Examples for each state are shown in figure 1, along with a diagram of the H:Si(100)-(2 × 1) surface reconstruction.

Besides its distinctive surface features, H:Si(100) is an ideal test-bed for developing CNN automation techniques. In addition to the relative simplicity of its reconstruction and a wealth of previous literature [25], H:Si(100) is a well understood substrate that has been used in many important advances in single atom technology and atomically precise materials engineering [24, 26–31]. Furthermore, because it has been previously studied in the context of machine-learning-enabled SPM [10–12], a good comparison can be formed with existing machine learning approaches based on full scans. As such, we used our existing dataset of 6167 complete images of H:Si(100) [12]. These images were acquired on a Omicron variable-temperature STM between March 2014 and November 2015, and at varying scan sizes and voltage biases of  $3 \times 3 \text{ nm}^2$  to



$80 \times 80 \text{ nm}^2$  and  $-2 \text{ V}$  to  $+2 \text{ V}$ , respectively. They were then hand-classified into the four categories listed above, as well as ‘tip changes’ and ‘generic defects’. Specific defects were not considered, as tip conditioning is performed based on the presence of any defect, and not the specific defect itself. As such, combining all defects into one category simplified the classification task, improving CNN performance.

From here, images were then randomly assigned into a training/testing set for training the network. Performance was then calculated with a separate, blind holdout set for verification. After random shuffling, 4987 of the images were assigned into the training/testing datasets in an 80/20 split, and 1180 into the holdout set. Data was then filtered to remove ambiguous images that were classified in multiple categories and/or the human classifiers did not perfectly agree upon [12]. This left 3386 images for training/testing, and 648 for blind verification. Because of the relatively small number of images available and to further improve performance, all data were then pre-processed using identical methods as in Gordon *et al* [12] (namely flattening and scaling linescans to have mean of 0 and standard deviation of 1). The training/testing sets were also augmented with vertical and horizontal flips, random rotations from  $0^\circ$  to  $360^\circ$ , crops, pans, and random Gaussian noise. This step was needed to prevent the network from rapidly overfitting. (This is where a CNN learns about random noise in the training set [32], performing extremely well during training, but poorly with testing/verification data unseen during training.) To allow for the best approximation of real-world performance on unseen data, the verification set was not augmented.

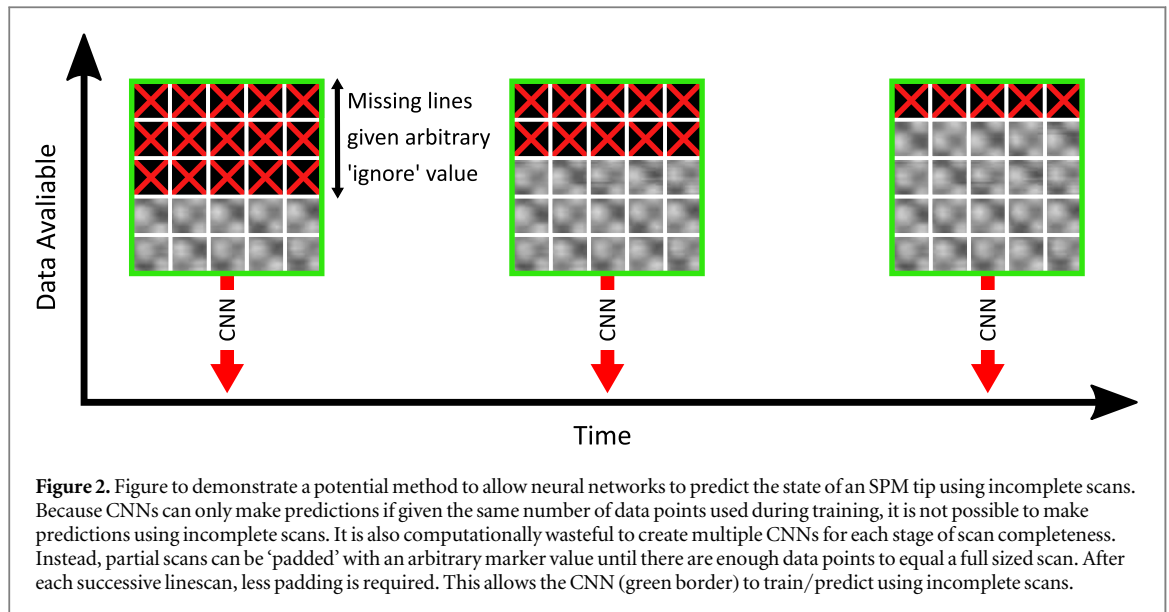
Furthermore, the data were also downscaled, which reduced both training time and overfitting further still [32]. Previously, optimal performance was found when reducing full-scans from size  $512 \times 512$  to  $128 \times 128$  [12]. Panning and cropping augmentations were applied in such a way that  $128 \times 128$  regions of the images were taken, allowing for downscaling without interpolation of data. Because the holdout data were not augmented, these images were downscaled in the more traditional sense.

Because an operator may desire the presence of some tip states (e.g. ‘individual atoms’), but desire the absence of others (e.g. ‘blurry’/defects), there are different implications to predicting the tip to be (or not to be) in different states. This makes use of the fact that CNNs do not make binary yes/no predictions, but instead output confidence ratings between 0 and 1 for each category. A decision is then made by rounding each number to 0 or 1. This rounding can be altered, and true positive/false positive rates then compared to demonstrate the overall performance of the classifier for each category. This forms the receiver-operator characteristic curve (ROC) [33, 34], which is then easily quantified by calculating the area under (AU) the curve. The AUROC for all categories can then be averaged to give an average AUROC for the classifier as a whole. A perfect classifier has  $\text{AUROC} = 1$ , while a network that operates purely by guessing has  $\text{AUROC} = 0.5$ . To the same end, we also calculate the precision-recall curve, in which the average precision is the weighted AU this curve. These metrics are independent of the number of images in each class, which drastically skews a pure accuracy value. As a result of this class imbalance, we therefore also calculate weighted accuracy [34] instead of pure accuracy. This was of particular importance in our case, as the filtered dataset was highly imbalanced, with only 5.6% images classified in the most ideal ‘atomic resolution’ class. Certain states were also more likely to appear than others, with 4.0% classified as asymmetries, 32.2% dimers, 16.4% rows, and 41.9% generic defect.

### 3. Results and discussion

#### 3.1. Data padding/masking

In many neural network applications, data are often of varying length. For example, in natural language processing [35], some words and sentences are inevitably longer than others. In these cases, shorter pieces of data



are lengthened by ‘padding’ them with a marker value [35] until they are as long as the longest piece of data. The marker value is chosen such that it cannot naturally appear in the real data. Training and testing then continues as normal, as the network learns to ignore the marker value. In the context of SPM, we can exploit the fact that images are sequentially generated one linescan at a time, and that completed images contain the same number of linescans, regardless of scan parameters. During an incomplete scan, the missing linescans can therefore be replaced with a marker value to allow the network to produce an output. Figure 2 demonstrates how data could be padded during scanning to form a full sized image.

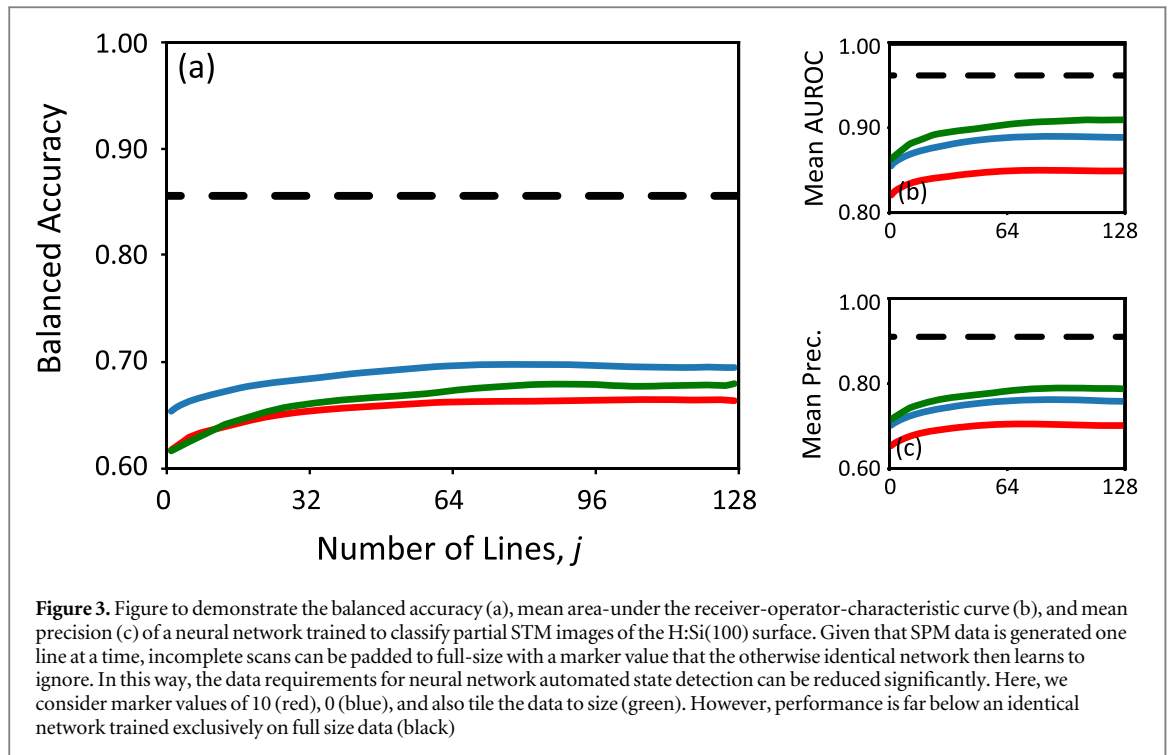
As such, it is possible simulate and test partial scans with the original dataset of complete scans. To do this, a random number of linescans from the end of the scan were ‘masked’ during training by replacing the real data with the marker value. To do this, we let

$$\begin{bmatrix} \mathbf{A}_j^i \\ \mathbf{A}_{j+1}^i \\ \mathbf{A}_{j+2}^i \\ \vdots \\ \mathbf{A}_N^i \end{bmatrix} = M, \quad (1)$$

where  $N$  is the total number of lines in a full image, and  $M$  the marker value. This produces an array,  $\mathbf{A}_j^i$ , for the  $i$ th image of a dataset, in which only  $j$  linescans appear to have been produced. To improve performance, data is further augmented by repeating  $\mathbf{A}^i$  multiple times, but with randomly assigned  $j$ .

Although this method is simple and can easily be applied to existing protocols, the use of a marker value is of course highly problematic. In the context of SPM, data can theoretically contain any positive or negative value within the operating range of the acquisition hardware. As such, no marker value exists that could not show up in the actual dataset, without being so large as to make the actual data miniscule by comparison and negatively impacting learning. As such, the network will likely become insensitive to some of the actual data. Given that each line was pre-processed to have mean of 0 and standard deviation of 1, we therefore consider arbitrary marker values of  $M = 0$  and  $M = 10$ . As an alternative, we also consider ‘tiling’ by repeating  $\mathbf{A}_j^i$  to full scan-size. This avoids the need to fill with an arbitrary marker value.

The CNN structure was chosen to be VGG-like [36] after strong all-round performance was previously found for H:Si(100) using a similar structure [12]. This network [36] begins with two 2D convolutional layers of 32 output filters,  $3 \times 3$  convolutional filters, and  $3 \times 3$  strides. This is followed by a third max pooling layer with  $2 \times 2$  convolutional filters and  $2 \times 2$  strides. This three layer block is then repeated, but with output filters of 64 and then 128 layers, respectively. The very first convolutional layer in the model was then altered to have  $7 \times 7$  convolutional filters and  $2 \times 2$  strides. This structure was then trained three separate times to create a majority voting ensemble. Not only does this allow for the performance benefits seen when taking a majority vote of a subjective task, but also reduces variance in CNN performance which was found to vary by about 1% between repeats. A schematic of this structure is provided as supplementary material available online at [stacks.iop.org/MLST/1/015001/mmedia](https://stacks.iop.org/MLST/1/015001/mmedia).

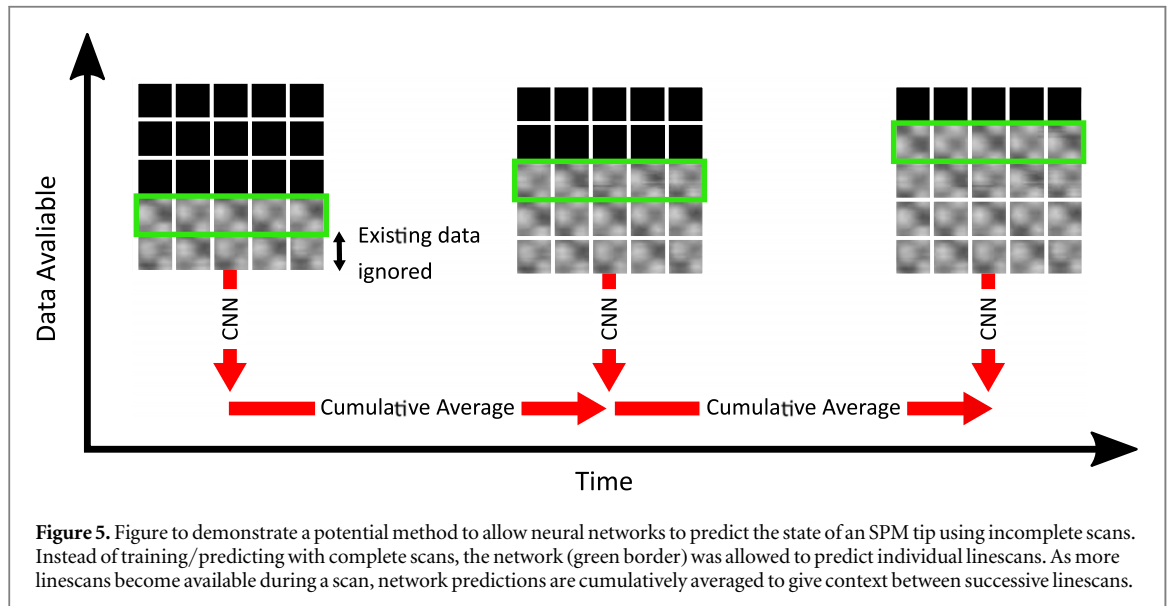
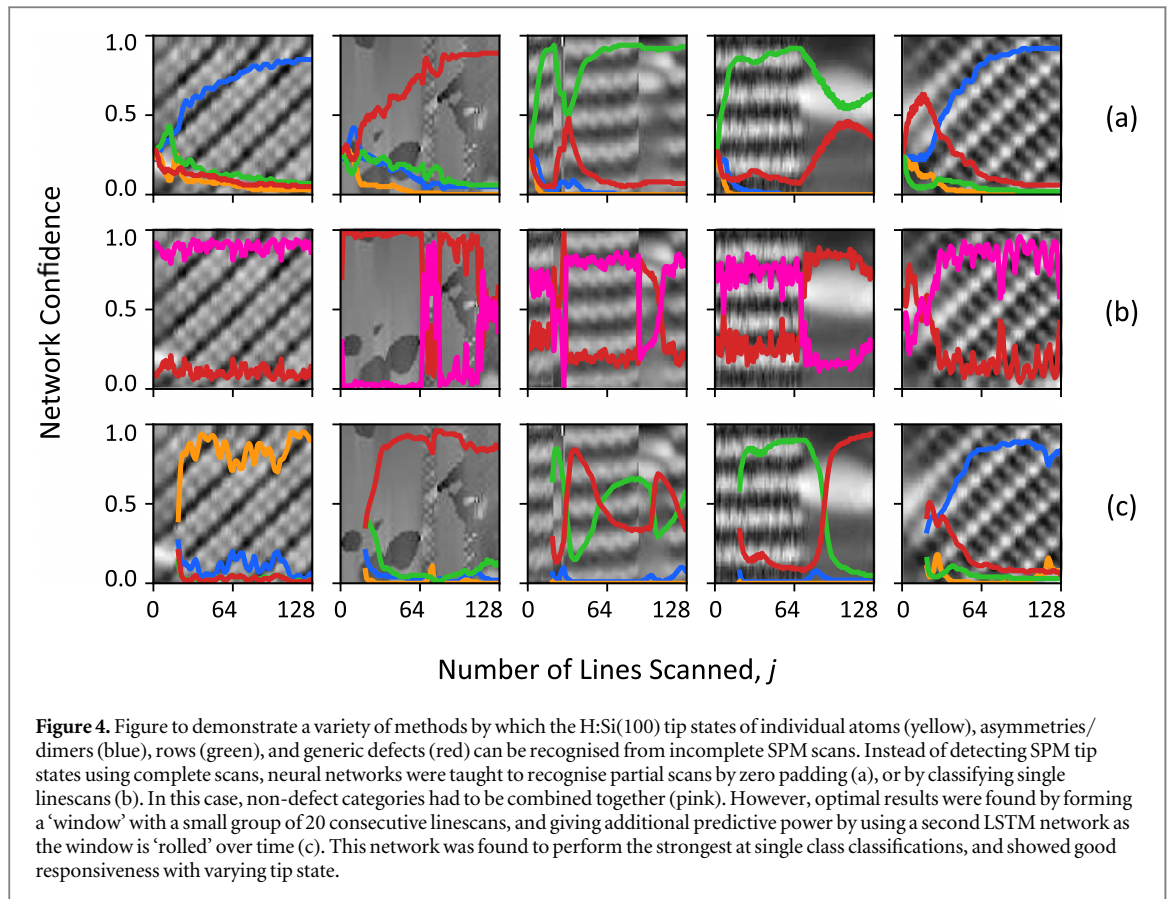


To test this method, the performance of the CNN ensemble was calculated as one additional line was unmasked at a time. We do this by masking from the  $j$ th line of  $A_j^i$  using equation (1) for all 648 images in the verification dataset. The CNN ensemble was then used to predict the tip state,  $P(A_j^i)$ , from  $j = 2$  to  $j = N$ . By assuming the human prediction to be perfectly correct, performance was calculated by comparing CNN predictions to the corresponding human predictions. Performance is shown as a function of  $j$  in figure 3.

From these figures, it can clearly be seen that for all types of padding, the padding-enabled-CNNs successfully learnt to make correct observations with limited data. Furthermore, when comparing the performance difference of small amounts of data with  $j = 2$  to full scans with  $j = N$ , the performance of all padding types only decreased by an average of  $4\% \pm 1\%$ ,  $8\% \pm 6\%$ , and  $7\% \pm 2\%$  for mean AUROC, mean precision, and balanced accuracy, respectively. Given that the balanced accuracy, mean precision and AUROC values are significantly better than the 0.25, 0.25 and 0.50 of guessing, respectively, it is entirely possible to assess SPM tip state with only a small number of linescans.

However, at  $j = N$  the padding-enabled-CNNs performed significantly worse than an identical ensemble trained without padding. Here, padding reduced full size performance by up to 12%, 23% and 22% for the mean AUROC, mean precision, and balanced accuracy, respectively, when compared to the worst performing padding methods. Giving CNNs the ability to classify partial scans therefore significantly harms performance, reducing the real-world effectiveness of such systems. We also note that this architecture also performed better than the ensembles presented in Gordon *et al* [12]. The large initial convolutional window may have caused this. Besides the reduced maximum performance, there was also a large computational inefficiency due to training the CNNs to perform (and subsequently ignore) a large number of expensive calculations on meaningless data.

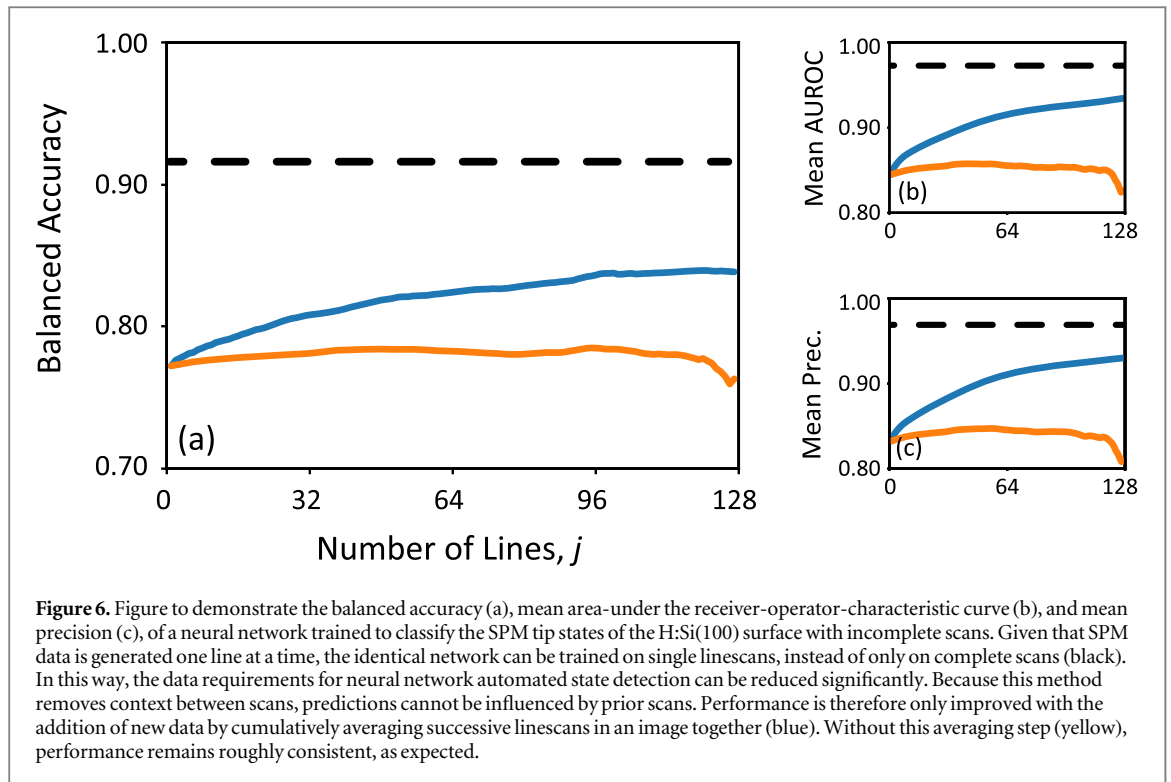
One advantage of partial-scan methods is that tip changes can be instantly detected by looking for changes and impulses in CNN output, as visible in figure 4. This is a significant improvement on previous full-scan methods which require a secondary ‘tip change’ network [12]. We note that without manual labelling of all tip change locations on all images, a quantitative analysis of tip-change detection is not possible. However, the imperfect ignoring of the marker values meant that some of the horizontal shears due to tip changes caused little-to-no-change in network output. The change in prediction to reflect a new tip state was also often small, and tended to ‘drift’ rather than instantly ‘snap’ to the new value. This was particularly problematic for tip changes later on in a scan. One explanation is that the network learnt to heavily rely on earlier scanlines because training images often had early scanlines present, but later scanlines did so increasingly rarely. It was also impossible to detect tip changes using the ‘tile’ method of data padding, which created a horizontal shear (visually identical to a tip change shear) between every tile. As such, padding should only be employed early on in scans and when the tip state is likely stable.



### 3.2. Individual linescan windows and cumulative averages

One alternative to padding incomplete scans is to train so as to classify the individual linescans that form an image, rather an image in its entirety. As new lines are scanned, they could immediately be predicted. This negates much of the insensitivity and computational wastefulness caused as a result of padding, and is demonstrated in figure 5.

However, one consequence of basing predictions on individual linescans is that each linescan is stripped of its context to the rest of the scan. Acquiring more linescans should therefore not improve network performance. As such, a small amount of context can be applied to the other scanlines in the image by applying an additional layer to cumulatively average the network predictions using the equation



$$\mathbf{P}(A_j^i) = \frac{\sum_{k=1}^j \mathbf{P}(A_k^i)}{j}, \quad (2)$$

where  $\mathbf{P}(A_j^i)$  is the cumulatively averaged vector describing the predictions of the  $j$ th linescan of the  $i$ th image in the dataset. A prediction for the entire image is therefore found when the condition  $j = N$  is met.

We also note that although the cumulative averaging provided context to the predictions, the actual predictive part of the network was unaware of the surrounding linescans. Whilst this averaging therefore served to reward consistent single-class output, it should be expected to have poor responsiveness to scans where the tip constantly changes shape. Furthermore, the network had little-to-no ability to distinguish between features that cannot be distinguished at the 1D level. For example, a single linescan of ‘atoms’ or ‘rows’ features in figure 1 would appear identical with a half-rectified sinusoid. The varying scan areas of the dataset required to make predictions invariant to scan area then prevent the network from learning any spatial information to distinguish between the two states. As such, the number of tip states was simplified to just two - ‘generic defect’ and ‘visible resolution’.

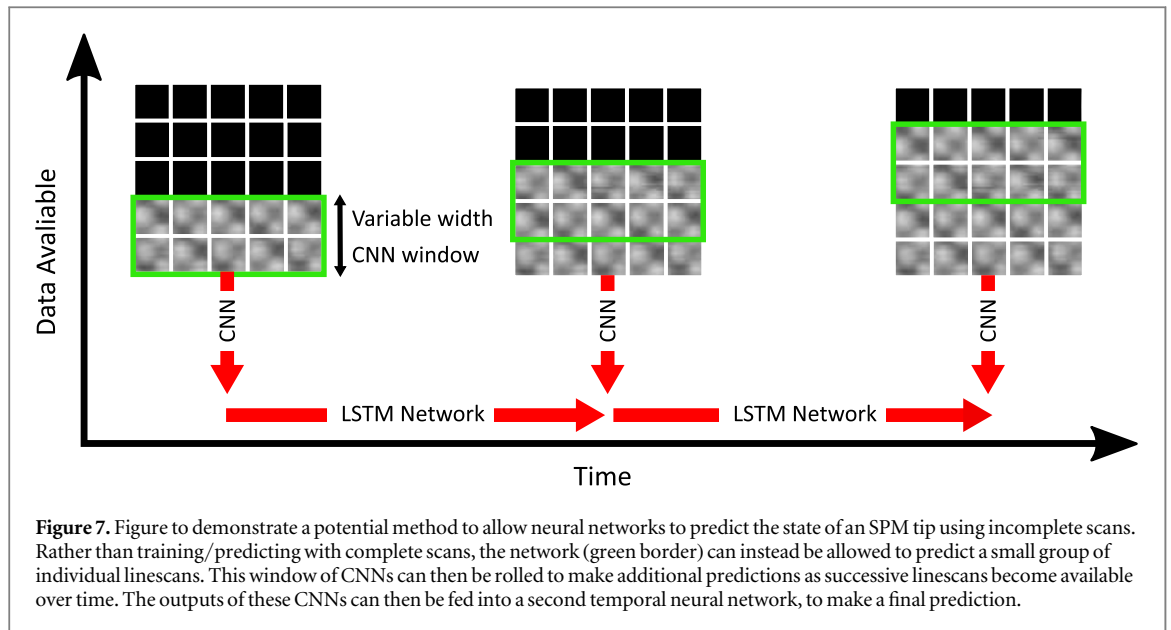
Adaptions also had to be made to the network architecture. Because 2D convolutions cannot be performed on 1D data, the 2D layers of the network were replaced with their one-dimensional counterparts to provide the closest possible comparison between the protocols. Furthermore, because successive lines were often highly similar, only 1 in every 30 lines of each image were used during training to prevent improper training and decrease training time.

As before, performance was verified by iteratively predicting additional lines of the  $i$  images in the holdout set and calculating the cumulative predictions using equation (2). This is shown in figure 6. To compare with full-sized performance, the 1D convolutions were replaced with their 2D equivalents (as used in section 3.1), and trained to recognise only the two simplified states.

Without the cumulative averaging layer, the low standard deviation demonstrated that performance remained near constant as expected, with AUROC of  $0.853 \pm 0.006$ , mean precision of  $0.841 \pm 0.007$ , and balanced accuracy of  $0.780 \pm 0.004$ . Un-averaged individual linescans therefore provide an effective means of making a basic, but accurate, assessment of the tip. Further, despite forcing the simplification of classes recognised, the decoupling of the lines meant that the network was highly sensitive to tip changes. This is visible when looking at the unaveraged output in figure 4(b). As this network was clearly more responsive to state changes than padding, it is possible to use the single linescan network, (along with its low computational cost), solely for the purpose of detecting tip changes by looking for sharp peaks and changes in network output.

Regardless, even stronger performance was seen with single-class images after cumulatively averaging. After including the layer, performance began to improve as expected, with AUROC substantially improving by 13.2%





relative to the average, mean precision by 11.8%, and balanced accuracy increasing by 9.9%. This resulted in an AUROC of over 0.9, thus demonstrating highly effective ability when full data is available. This was also found to hold true for the padding strategy with all 128 linescans. It should, however, be stressed, that relative to training only with complete scans, peak performance is still reduced. In this case, when training the 2D CNN solely with complete scans and the two simplified categories, AUROC performance was near perfect, at 0.973. Further, cumulative averaging caused predictions to be significantly less sensitive to tip changes, as expected.

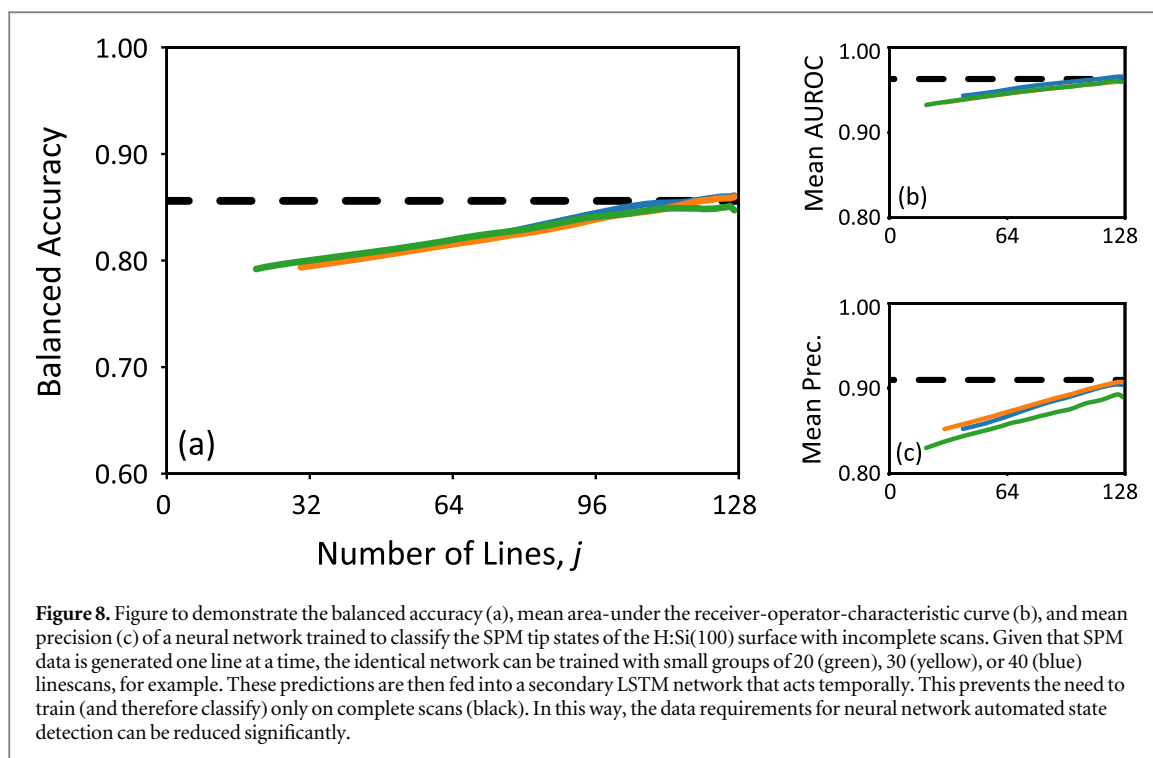
### 3.3. Multiple linescan windows and LSTM

Whilst single linescans provide an effective method to make a basic assessment of the tip, the inability to assess the complete range of states makes it of limited use. To overcome the lack of context between linescans, a CNN could instead be trained to recognise a small ‘window’ consisting of a fixed number,  $W$ , of linescans. As new data becomes available, the window could then be ‘rolled’ to consist of the new line and the  $(W - 1)$  linescans preceding it. This window could then be iteratively rolled while an image is being generated. We therefore modify equation (2), and use cumulative averaging to make predictions after each successive linescan from  $j = W + 1$  to  $j = N$

$$\mathbf{P}(A_j^i) = \frac{\sum_{k=W+1}^j \mathbf{P}(A_{(k-W):k}^i)}{j}. \quad (3)$$

However, whilst effective at improving single-state classification performance and rewarding tip consistency, cumulative averaging does not make the predictive part of the neural network aware of the lines surrounding each window, resulting in decreased responsiveness. One recent advance in the area of video content recognition is the long-term recurrent convolutional network (LRCN) [37], which has been shown to be highly effective at this task. Here, a second network is placed just before the final (dense) CNN layer (which reduces the output to a size equal to the number of classification categories). This second network is typically a long-short-term-memory (LSTM) network [38], which is often used for 1D sequence classification. The LSTM network then acts on the temporal domain of the data, giving context to the single CNNs which have no knowledge of how the video frames link together. This can be made analogous to SPM, where each sub-image of width  $W$  becomes a video frame. The temporal element is seen as the window rolls when  $j$  increments over time with new data. We therefore replace the cumulative averaging layer with an LSTM network with 256 hidden layers, and calculate predictions,  $\mathbf{P}(A_{(j-W):j}^i)$ , from  $j = W + 1$  to  $j = N$  as before, with increasing  $j$  chosen as the temporal axis. For consistency, we employ the same 2D CNN architecture as before. The resulting protocol is shown in figure 7.

One consequence of this method is that  $W$  linescans must first be accumulated before any predictions can be made. As such, whilst larger  $W$  will give the network more data with which to make predictions, a larger number of linescans are required to be scanned before the window can be fully filled. For example, for a window of  $W = 20$ , predictions can only be made after the 20th, 21st, 22nd linescans, and so on. We also note that the size and number of convolutions used meant that predictions with  $W < 20$  were not possible with the CNN structure used. Furthermore, all but one linescan of data is repeated with each additional window, multiplying memory usage by  $N - W + 1$  times.



As can be seen in figure 8, the inclusion of additional linescans once again resulted in improved performance, demonstrating that the LSTM component did indeed learn from the temporal evolution of the scans. Performance was also very strong regardless of  $j$ . For example, full scans with  $W = 20$  yielded a near-perfect AUROC of 0.960, mean precision of 0.890, and a balanced accuracy of 0.847. This is almost identical to the AUROC, mean precision and balanced accuracy of 0.963, 0.910 and 0.856, respectively, calculated when training the CNN component only on full-sized images. The wider LRCN networks were even able to exceed full-size performance, *despite using less data*. This is understandable, given that a human operator will often look not only at the scanlines, but also at how they evolve over time. Only the LRCN network takes advantage of this temporal context. It can therefore be concluded that by adding LSTM to an existing network and retraining on partial scans of fixed size, a full set of STM image classes/tip states can be correctly and accurately assessed with negligible performance impact despite using a fraction of the data. However, increasing  $W$  beyond  $W = 30$  did not always improve performance. Although wider windows provided more opportunities to observe trends in the 2D convolutional domain, leading to near-baseline precision of 0.908 for  $W = 30$  smaller windows provided more temporal elements for the LSTM layer to use.

The benefit of using temporal information can also be seen by comparing LRCN to cumulative averaging. For the same  $W = 20$  window, full-scan performance using cumulative averaging was calculated to have AUROC of 0.880, mean precision of 0.862 and balanced accuracy of 0.620. Not only was this slightly worse than the padding method of figure 3, but also significantly poorer than LRCN, which scored 9.10% higher for AUROC, 3.24% for mean precision, and 36.7% for balanced accuracy. This performance disparity held true regardless of values of  $W$  and  $j$ , or when classifying variable state images. As viewable in the supplementary material, cumulative averaging was often unresponsive to both sudden changes in state. Moreover, LRCN was more able to correctly distinguish between atoms and asymmetries, and was less likely to mistakenly see rotated surfaces as a 'generic defect' compared to the baseline of full scan classification. Whilst it would seem obvious to combine both LRCN and cumulative averaging, the issues with decreased responsiveness later in a scan remain. This resulted in a small performance penalty which increased as more linescans were simulated (on the order of 1% at  $j = N$ ). Whilst cumulative averaging was still better than guessing and is therefore another potential method for speeding up tip state recognition, LRCN is superior.

Furthermore, whilst the state of the tip was still successfully observed with  $W = 20$ , the size and number of convolutions used meant that window sizes below  $W = 20$  were not possible to test. This meant that  $j = 20$  lines must first be acquired before predictions can be made. To reduce the number of linescans further, larger images could instead be considered (which in this case would be achieved by downscaling from  $512 \times 512$  to a size larger than  $128 \times 128$ ). For example, simulating  $W = 20$  with 256 points per linescan would be equivalent to 128 points per linescan with  $W = 10$ . However, the same number of data-points would need to be acquired before predictions could be made. There would therefore be no improvement to tip assessment speed in

practise. Although the network parameters could be decreased to allow for smaller  $W$ , this would result in a different network that could not be fairly compared in this study. To allow for predictions at any  $j$ , it is trivial to create a ‘hybrid’ network ensemble in which a basic assessment is made using the linescan/padding methods for low  $j$ , and then LRCN for the remainder of the scan.

## 4. Conclusion

By comparing a variety of methods based around a common VGG network, we have successfully demonstrated that STM images of the H:Si(100) surface can be accurately assessed using partial scans. As such, only a few lines from a typical  $128 \times 128$  scan are now required to assess the tip, which is a fraction of the data required by previous CNN assessment protocols. Given that the majority of the time spent maintaining SPM tips is spent acquiring data, a ‘hybrid’ approach combining individual linescans and LRCN prediction would speed up CNN routines by approximately 100 times. This allows for state recognition in a time similar to that of current manual means, thus making it practical for everyday use. However, given that the states considered only apply to the H:Si(100) surface, new datasets and networks must be manually created and trained for each surface, making this strategy non-applicable to poorly understood surfaces.

Relative to a full-size network, we find that similar or better performance can be achieved with less data by creating a small window of multiple linescans, and adding an LSTM layer to make predictions as the window is rolled over time. Furthermore, we qualitatively demonstrate that the use of partial linescans allows tip changes to be detected without the need for a secondary network. We also show that this method allows for the detection of images in which tip changes cause multiple tip states to be present, alongside their relative position in the image. However, the low number of human classifiers and lack of manual labelling of these positions during data collection meant that only single tip-state images were quantitatively assessed. Furthermore, none of these approaches overcome the limitation of only being able to automate assessment of a single, already known surface reconstruction after a lengthy data collection process.

In future, we aim to assess SPM tips with a ‘hybrid’ approach combining multiple protocols of predicting with padded full-scans, individual linescans, and temporally connected partial scans of fixed width. Ultimately, this will enable seamless, automatic and constant maintenance of SPM tip integrity as part of routine experimental sessions. Unsupervised learning is the next, obvious, protocol to adopt in order to make machine learning strategies sample-independent.

## Acknowledgments

The authors gratefully acknowledge funding by the Engineering and Physical Sciences Research Council via grant EP/N02379X/1. We also thank I Swart, L Knijff, S E Freney, and S Zevenhuizen of the Debye Institute for Nanomaterials Science, at Utrecht University for their continued assistance and advice (including support for the invaluable MATE-for-Dummies and access2TheMatrix Python packages). We gratefully acknowledge helpful discussions with Bob Wolkow, John Randall, Morten Moller (who also provided the dataset of H:Si(100) images used in this work), Nicole Landon and Richard Woolley.

## Data availability statement

The data that support the findings of this study are openly available at <https://doi.org/10.25412/iop.9767282.v1>.

## ORCID iDs

Oliver M Gordon  <https://orcid.org/0000-0001-8733-7500>

Philip J Moriarty  <https://orcid.org/0000-0002-9926-9004>

## References

- [1] Tajaddodianfar F, Moheimani S O R, Owen J and Randall J N 2018 On the effect of local barrier height in scanning tunneling microscopy: measurement methods and control implications *Rev. Sci. Instrum.* **89** 013701
- [2] Tewari S, Bastiaans K M, Allan M P and van Ruitenbeek J M 2017 Robust procedure for creating and characterizing the atomic structure of scanning tunneling microscope tips *Beil. J. Nanotechnol.* **8** 2389–95
- [3] Giessibl F J 2019 The qPlus sensor, a powerful core for the atomic force microscope *Rev. Sci. Instrum.* **90** 011101

- [4] Gross L, Mohn F, Moll N, Liljeroth P and Meyer G 2009 The chemical structure of a molecule resolved by atomic force microscopy *Science* **325** 1110–4
- [5] Sun Z, Boneschanscher M P, Swart I, Vanmaekelbergh D and Liljeroth P 2011 Quantitative atomic force microscopy with carbon monoxide terminated tips *Phys. Rev. Lett.* **106** 046104
- [6] Chiutu C, Sweetman A M, Lakin A J, Stannard A, Jarvis S, Kantorovich L, Dunn J L and Moriarty P 2012 Precise orientation of a single C-60 molecule on the tip of a scanning probe microscope *Phys. Rev. Lett.* **108** 268302
- [7] Meyer G, Bartels L and Rieder K 2001 Atom manipulation with the STM: nanostructuring, tip functionalization, and femtochemistry, *Comput. Mater. Sci.* **20** 443–50
- [8] Gross L, Moll N, Mohn F, Curioni A, Meyer G, Hanke F and Persson M 2011 High-resolution molecular orbital imaging using a p-Wave STM tip *Phys. Rev. Lett.* **107** 086101
- [9] Jarvis S, Sweetman A, Bamidele J, Kantorovich L and Moriarty P 2012 Role of orbital overlap in atomic manipulation *Phys. Rev. B* **85** 235305
- [10] Rashidi M and Wolkow R A 2018 Autonomous scanning probe microscopy *in situ* tip conditioning through machine learning *ACS Nano* **12** 5185–9
- [11] Rashidi M, Croshaw J, Mastel K, Tamura M, Hosseinzadeh H and Wolkow R A 2019 Autonomous atomic scale manufacturing through machine learning arXiv:1902.08818
- [12] Gordon O, D'Hondt P, Knijff L, Freeney S, Junqueira F, Moriarty P and Swart I 2019 Scanning probe state recognition with multi-class neural network ensembles *Rev. Sci. Int.* **90** 103704
- [13] Zhang Y et al 2019 Machine learning in electronic-quantum-matter imaging experiments *Nature* **570** 484
- [14] Burzawa L, Liu S and Carlson E 2019 Classifying surface probe images in strongly correlated electronic systems via machine learning *Phys. Rev. Mater.* **3** 033805
- [15] Blunt M, Martin C, Ahola-Tuomi M, Pauliac-Vaujour E, Sharp P, Nativo P, Brust M and Moriarty P 2007 Coerced mechanical coarsening of nanoparticle assemblies *Nat. Nanotechnol.* **2** 167
- [16] Siepmann P, Martin C P, Vancea I, Moriarty P J and Krasnogor N 2007 A genetic algorithm approach to probing the evolution of self-organized nanostructured systems *Nano Lett.* **7** 1985–90
- [17] Straton J C, Bilyeu T T, Moon B and Moeck P 2014 Double-tip effects on scanning tunneling microscopy imaging of 2D periodic objects: unambiguous detection and limits of their removal by crystallographic averaging in the spatial frequency domain *Cryst. Res. Technol.* **49** 663–80
- [18] Woolley R A J, Stirling J, Radocea A, Krasnogor N and Moriarty P 2011 Automated probe microscopy via evolutionary optimization at the atomic scale *Appl. Phys. Lett.* **98** 253104
- [19] Stirling J, Woolley R A and Moriarty P 2013 Scanning probe image wizard: a toolbox for automated scanning probe microscopy data analysis *Rev. Sci. Instrum.* **84** 113701
- [20] Wang Y, Kilpatrick J L, Jarvis S P, Boland F M F, Kokaram A and Corrigan D 2016 Double-tip artifact removal from atomic force microscopy images *IEEE Trans. Image Process.* **25** 2774–88
- [21] Wolkow R A 1992 Direct observation of an increase in buckled dimers on Si(001) at low temperature *Phys. Rev. Lett.* **68** 2636
- [22] Sweetman A, Stirling J, Jarvis S P, Rahe P and Moriarty P 2016 Measuring the reactivity of a silicon-terminated probe *Phys. Rev. B* **94** 115440
- [23] Sweetman A, Jarvis S, Danza R and Moriarty P 2012 Effect of the tip state during qPlus noncontact atomic force microscopy of Si(100) at 5 K: Probing the probe *Beil. J. Nanotechnol.* **3** 25
- [24] Møller M, Jarvis S P, Guérinet L, Sharp P, Woolley R, Rahe P and Moriarty P 2017 Automated extraction of single H atoms with STM: tip state dependency *Nanotechnology* **28** 075302
- [25] Walsh M A and Hersam M C 2009 Atomic-scale templates patterned by ultrahigh vacuum scanning tunneling microscopy on silicon *Annu. Rev. Phys. Chem.* **60** 193–216
- [26] Shen T, Wang C, Abeln G, Tucker J, Lyding J, Avouris P and Walkup R 1995 Atomic-scale desorption through electronic and vibrational-excitation mechanisms *Science* **268** 1590–2
- [27] Lopinski G, Wayner D and Wolkow R 2000 Self-directed growth of molecular nanostructures on silicon *Nature* **406** 48
- [28] Fuechsle M, Miwa J A, Mahapatra S, Ryu H, Lee S, Warschkow O, Hollenberg L C, Klimeck G and Simmons M Y 2012 A single-atom transistor *Nat. Nanotechnol.* **7** 242
- [29] Weber B et al 2012 Ohm's law survives to the atomic scale *Science* **335** 64–7
- [30] Achal R, Rashidi M, Croshaw J, Churchill D, Taucer M, Huff T, Cloutier M, Pitters J and Wolkow R A 2018 Lithography for robust and editable atomic-scale silicon devices and memories *Nat. Commun.* **9** 2778
- [31] Huff T, Labidi H, Rashidi M, Livadaru L, Dienel T, Achal R, Vine W, Pitters J and Wolkow R A 2018 Binary atomic silicon logic *Nat. Electron.* **1** 636–43
- [32] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
- [33] Fawcett T 2006 An introduction to ROC analysis *Pattern Recognit. Lett.* **27** 861–74
- [34] Pedregosa F et al 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
- [35] Young T, Hazarika D, Poria S and Cambria E 2018 Recent trends in deep learning based natural language processing *IEEE Comput. Intell. Mag.* **13** 55–75
- [36] Simonyan K and Zisserman A 2014 Very deep convolutional networks for large-scale image recognition arXiv:1409.1556
- [37] Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K and Darrell T 2015 Long-term recurrent convolutional networks for visual recognition and description *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 2625–34
- [38] Hochreiter S and Schmidhuber J 1997 Long short-term memory *Neural Comput.* **9** 1735–80