Scanning tunneling state recognition with multi-class neural network ensembles

O. Gordon, P. d'Hondt, L. Knijff, S. E. Freeney, F. Junqueira, P. Moriarty, I. Swart

Review of Scientific Instruments

Scanning tunneling state recognition with multi-class neural network ensembles

Cite as: Rev. Sci. Instrum. **90**, 103704 (2019); https://doi.org/10.1063/1.5099590
Submitted: 11 April 2019 • Accepted: 06 September 2019 • Published Online: 11 October 2019

O. Gordon, P. D'Hondt, L. Knijff, et al.

Review of Scientific Instruments ARTICLE scitation.org/journal/rsi

Scanning tunneling state recognition with multi-class neural network ensembles

Cite as: Rev. Sci. Instrum. 90, 103704 (2019); doi: 10.1063/1.5099590
Submitted: 11 April 2019 • Accepted: 6 September 2019 •
Published Online: 11 October 2019

O. Gordon,[1,a)] P. D'Hondt,[1,2] L. Knijff,[3] S. E. Freeney,[3] F. Junqueira,[1] P. Moriarty,[1] and I. Swart[3]

AFFILIATIONS

[1]School of Physics & Astronomy, The University of Nottingham, University Park, Nottingham, NG7 2RD, United Kingdom
[2]IEMN – Laboratoire Central De L'Institut, Cité Scientifique, Avenue Henri Poincaré, CS 60069, 59 652 Villeneuve d'Ascq Cedex, France
[3]Debye Institute for Nanomaterials Science, Utrecht University, Utrecht 3584 CC, The Netherlands

[a)]**Electronic mail:** oliver.gordon@nottingham.ac.uk

ABSTRACT

One of the largest obstacles facing scanning probe microscopy is the constant need to correct flaws in the scanning probe *in situ*. This is currently a manual, time-consuming process that would benefit greatly from automation. Here, we introduce a convolutional neural network protocol that enables automated recognition of a variety of desirable and undesirable scanning tunneling tip states on both metal and nonmetal surfaces. By combining the best performing models into majority voting ensembles, we find that the desirable states of H:Si(100) can be distinguished with a mean precision of 0.89 and an average receiver-operator-characteristic curve area of 0.95. More generally, high and low-quality tips can be distinguished with a mean precision of 0.96 and near perfect area-under-curve of 0.98. With trivial modifications, we also successfully automatically identify undesirable, non-surface-specific states on surfaces of Au(111) and Cu(111). In these cases, we find mean precisions of 0.95 and 0.75 and area-under-curves of 0.98 and 0.94, respectively. Provided that training data are available, these ensembles therefore enable fully autonomous scanning tunneling state recognition for a wide range of typical scanning conditions.

I. INTRODUCTION

While scanning tunneling microscopy (STM) has allowed researchers to make observations at the atomic level for decades,[1–3] success is highly reliant on the production of atomically sharp scanning tips. Although sharp tips are readily created *ex situ*,[4] imperfections in the tip apex including the presence of "double" or multiple tips mean that image artifacts often appear spontaneously during experimental sessions. To maintain resolution, apex flaws must be repeatedly corrected *in situ* through a repeated combination of controlled voltage pulsing and/or tip crashing.

Despite the manual, time-consuming nature of tip correction, there have been surprisingly few attempts to date to automate the process[5–8] and allow for the setup and collection of large amounts of data in the absence of a microscopist. Of these attempts, a variety of pitfalls have been identified, ranging from low accuracy and high computational cost to faltering when multiple tip flaws are present. They also often require a degree of manual input, are invariant to scale and rotation, or fail when the tip spontaneously changes the visible resolution midimage. Convolutional neural networks (CNNs) are highly promising candidates for this task which routinely achieve high accuracy in complex vision tasks such as medical, satellite, and digit recognition.[9–11] Despite this, in the context of STM, only Rashidi and Wolkow[8] have to the best of our knowledge used CNNs for tip-conditioning and only while scanning the H:Si(100) surface.

In this paper, we broaden the approach of Rashidi and Wolkow to a method that can reliably assess the state of an STM tip while scanning on both metallic and semiconducting surfaces. This is achieved via majority voting from an ensemble of multiple CNNs. We also increase the number of distinct recognizable states and allow for desirable and nondesirable tip state classifications at non-fixed length scales and rotations. We ultimately present ensemble CNNs capable of classifying multiple tip states with humanlike performance and weighted accuracies in excess of 80% (and 90% in some cases).

II. METHODS

When assessing the quality of an STM image, not all features are considered equally. For example, STM tips with the highest possible image resolving power are not necessarily the most suitable for nonimaging tasks such as atomic and molecular manipulation.[12] Furthermore, an operator may want to observe desirable features but actively avoid undesirable artifacts. While Rashidi and Wolkow[8] distinguished two key tip states when imaging H:Si(100) for CNN-driven automated STM, a much wider set of classifications is possible. For this surface, these include[13] "atoms" (for the sharpest tips), "dimers," "asymmetries," and "rows." Example images are shown in Fig. 1. Although H:Si(100) is a substrate that underpins many advances in single atom technologies,[12,14,15] these classifications are surface-specific. "Double tips," "tip changes," "step edges," "impurities," and image corruption "defects" are all undesirable artifacts that could apply to any surface. To this end, and to demonstrate the general applicability of our CNN protocol, we also study two other commonly studied surfaces:[16,17] Cu(111) and Au(111).

To train the CNNs, 13 789 images were first obtained. H:Si(100) images were acquired at room temperature between March 2014 and November 2015 on a Scienta Omicron variable temperature STM at various rotations, length scales between 3×3 nm^2 and 80×80 nm^2, and resolutions up to 512×512 pixels. The Au(111) and Cu(111) images were acquired similarly on an Omicron LT but at a fixed scan size (30×30 nm), resolution (150×150 pixels), and temperature (4.5 K). The images of H:Si(100) were manually classified into the four tip states listed above (i.e., atoms, dimers, asymmetries, and rows) and two other categories, tip changes and generic defects. Similarly, Au(111) and Cu(111) images were classified into five categories of undesirable defects and the one desirable state of sharp resolution.[18] To prevent overestimation of performance, a random selection of images were withheld as holdout data for analysis. Train and test data were then created by randomly splitting the remaining images with an 80:20 ratio.

Although in practice STM images are multilabel (in which images can belong to multiple categories), we classified and discarded data such that we only trained with multiclass (in which images can belong to only one category). This was beneficial as CNNs learn from the relationship between categories and so did not have to learn to ignore relations that did not exist. It is also known that although a CNN can learn with ambiguous or misleading training labels, performance is reduced.[19–21] However, because undesirable tip changes can occur even when observing a desirable tip state, these were not excluded. Instead, tip changes were trained in a separate binary yes/no CNN for H:Si(100), and the remaining Si images were trained in a four-class CNN. Tip change separation was not applied to the Cu and Au datasets as the aim was to explore the relations between undesirable defects.

There was also a great deal of variety between classifications despite the consistent classification scheme and limited number of classifiers. While we did not train on images that the human classifiers did not agree on, the large degree of ambiguity in classification meant that many ambiguous images remained. In the absence of a perfect classification system and greater number of classifiers, these imperfect human classifications formed the training data that the network had to learn from. As such, no CNN could achieve 100% accuracy without overfitting. For example, given that 78% of the silicon dataset was agreed upon, it could be tentatively argued that a humanlike CNN would score similarly. (A poll carried out in our group which involved the manual classification of a small subset of the Si dataset by nine scanning probe microscopists, similarly found only 73% agreement.) Ultimately, 3386 H:Si(100), 3600 Cu(111), and 2470 Au(111) images were used for training/testing and 431, 1120, and 432 images for verification, respectively.

To improve training performance, the training and testing data were repeated and augmented. Expanding on the simple vertical and horizontal flips used by Rashidi, we also applied rotations from 0° to 360°, and cropped, panned, and added random amounts of Gaussian noise. For the tip change categories, only horizontal flips and Gaussian noise were applied as in our case tip changes were horizontal shears, and zooming in might crop off the discontinuity. These steps improve performance by reducing overfitting, in which a CNN uses random trends to correctly classify training data at the expense of misclassifying unseen data.[22] Additional random trends created by

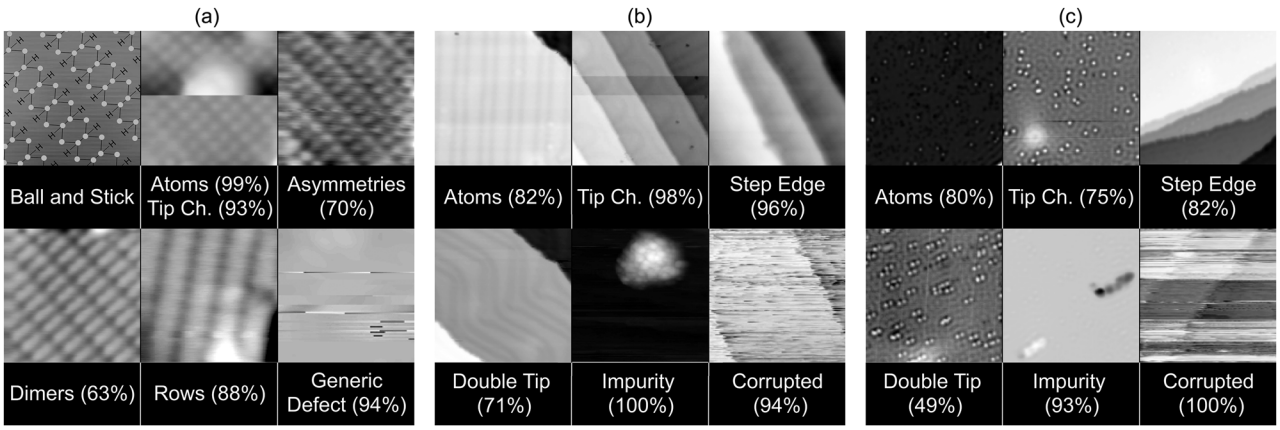

FIG. 1. Selection of images demonstrating key tip states for STM imaging of (a) H:Si(100), (b) Au(111), and (c) Cu(111), and the confidence thresholds of convolutional neural networks used to classify them. We note that in many examples, features can appear to strongly blend between images, such as with asymmetries and dimerlike modulation in rows in (a). Because creating unambiguous training data was impractical, we therefore combined these classes.

the physical scan environment were negated with minimal processing[23] by flattening data on a line-by-line basis along the x axis. Data were also scaled to the order of −1 to 1.

CNNs also had to be prevented from overfitting by learning about the differing number of images in each class. For example, 5.6% of the images in the H:Si(100) filtered dataset were atoms, compared to 41.9% generic defects. A large variety was also observed in the Cu(111) and Au(111) sets. We therefore weighted each category by the reciprocal of the percentage of each class present, used a weighted accuracy metric[24] (where the reciprocal of the number of classes is defined as guessing), and randomly shuffled data. Without these steps, a CNN could rapidly take an example dataset containing nine good images and one bad, and be 90% accurate by guessing all images as good. It is for this reason that other authors warn against using solely accuracy to judge the performance of weighted datasets.[8,25]

Furthermore, our priority when establishing a CNN protocol was not to maximize the ratio of true to false classifications (i.e., accuracy) but to maximize true *positive* classifications. This was observed using the metric of precision (defined as the ratio of true positive classifications to total positive classifications). By increasing the confidence threshold required to make a positive classification, precision was increased at the cost of increased false negatives and therefore decreased accuracy. This is visible in receiver operator characteristic (ROC) and precision-recall (PR) curves. All-around performance is given by the area under ROC (AUROC), in which a perfect classifier has an AUROC of 1 and guessing 0.5. These metrics are also not affected by class imbalance[25] and were therefore superior to accuracy. (We therefore also made the traditional distinction between accuracy and precision, rather than using the terms interchangeably.[8])

In addition to the network described by Rashidi and Wolkow[8] (RW), we tested models similar to the popular visual geometry group (VGG) network with and without batch normalization. We also tested a model highly similar to Google's SqueezeNet.[26] This network had ten back-to-back convolutional layers, filters increasing in number from 32 to 1024, strides alternating between 1 and 2, and 3 × 3 convolutions. Between layers, batch normalization and the elu[27] activation function were applied. The loss rate was also gradually reduced during training to reduce overfitting further.

We note that although multiclass networks are typically trained with a sigmoid activation function and categorical cross-entropy loss function, we did not use these. Because future data would be multilabel, we instead opted for the multilabel standard of softmax and binary cross-entropy.[28] This made the confidence prediction of each category 0–1 independent of each other, instead of mathematically linking all the predictions for each category to sum to 1.[29]

To determine an optimal ensemble CNN, a variety of model structures, optimizers[30–33] and learning rates were trained and analyzed. In all cases, training was done at a batch size of 128 and image sizes of 128 × 128 pixels. At higher sizes, training time massively increased but with little to no improvement in performance. A more traditional Random Forest Classifier (RFC) was also implemented for comparison. The top performing models were then combined to create a majority voting ensemble, which have been shown to further improve performance.[34] For ambiguous data, this was also more analogous to a majority human vote with different models having different preferences. Training and analysis was performed with Python 3.6.3, Keras[35] 2.2.2, Tensorflow 1.11.0, and an Nvidia Titan Xp.

III. RESULTS

First, the individual models were compared. Table I displays the best results obtained for all the desirable/undesirable multiclass models. Although all networks performed significantly better than RFC and weighted random guessing, the RW CNN performed poorly and similar to the more traditional RFC. Furthermore, at the 32 × 32 image size described by Rashidi and Wolkow,[8] RW performed comparable to random guessing, indicating the high difficulty of this task.

We also found a wide variety in performance between different surfaces, indicating that *CNN architectures respond differently to different surfaces*. For example, while SqueezeNet was the best performer for H:Si(100), only VGG like networks were suitable for Au and Cu. Furthermore, batch normalization improved performance on H:Si(100) while negatively impacting Au(111) and Cu(111). This variance is understandable, given the current lack of consensus on how network structure relates to performance on a given data set.[36]

The best performing networks were then taken and turned into an ensemble. Three were chosen as this gave a good balance between performance and memory usage. As expected, small performance improvements were seen when moving to ensembles. For H:Si(100), the top performer was an ensemble of two SqueezeNets and one batch-normalized VGG, with adam, sgd, and rmsprop optimizers, and learning rates of 0.001, 0.0001, and 0.0001, respectively. However, our ensemble structure did not train well with Cu(111) and Au(111) [65% balanced accuracy, 0.64 precision, 0.89 AUROC on Cu(111)]. This is likely because of the low performance of the component networks on these surfaces. As such, ensembles for

TABLE I. Table to compare the performance of a variety of machine learning methods for classifying desirable and undesirable tip states for six classes of Au(111) and Cu(111), and four classes of H:Si(100). The SqueezeNet, VGG, Rashidi-Wolkow (RW), and ensemble networks are examples of convolutional neural networks. These all performed significantly better than the more traditional Random Forest Classifier (RFC) with 5000 trees, and random guessing, which performed as expected.

	Ensemble			SqueezeNet			VGG (Batchnorm)			VGG			RW			RFC			Random		
	Si	Au	Cu	Si	Au	Cu	Si	Au	Cu	Si	Au	Cu	Si	Au	Cu	Si	Au	Cu	Si	Au	Cu
AUROC	0.95	0.98	0.94	0.94	0.95	0.88	0.92	0.93	0.85	0.91	0.98	0.93	0.87	0.82	0.77	0.79	0.88	0.83	0.50	0.50	0.50
Bal. Acc.	0.78	0.86	0.80	0.77	0.71	0.67	0.74	0.74	0.59	0.72	0.86	0.76	0.62	0.55	0.50	0.46	0.53	0.52	0.25	0.16	0.16
Precision	0.89	0.95	0.75	0.88	0.82	0.67	0.82	0.77	0.57	0.82	0.92	0.72	0.71	0.54	0.47	0.57	0.62	0.51	0.25	0.18	0.17

Au(111) and Cu(111) were therefore created from multiple repeats of the VGG like network.

Although Table I indicated strong overall performance, this was likely underestimated. Considering Fig. 1, there was a high degree of feature overlap (such as Si dimer images with bright, asymmetric edges) which made the classification task subjective. While these categories were eventually combined as they were routinely misclassified together,[37] the filtered multiclass images still contained acceptable multilabel answers. Because we only allowed one correct classification for any image, the network was often punished despite producing a sensible distribution. This would have been avoided were significantly more classifiers employed.

Despite this, Fig. 2 shows that the AUROC for all categories and surfaces was very high. This indicated that the classifier had a low false positive rate but at the cost of a high false negative rate. Although decreasing accuracy, this characteristic is not detrimental to areas such as ours when only positive predictions are to be acted upon. Furthermore, ambiguous classifications often had confidences <0.5, increasing false negative count and reducing accuracy further still. Unambiguous cases, such as corruptions, impurities and individual atoms of Au(111) and Cu(111), and generic defects of H:Si(100), were otherwise classified extremely well with near perfect AUROC.

Furthermore, misclassifications were often between sets of desirable/undesirable states, rather than with desirable states being misclassified as undesirable and vice versa. To demonstrate this, the four class H:Si(100) and Au(111) ensembles were simplified into "good/bad" classifiers. They then achieved improved balanced

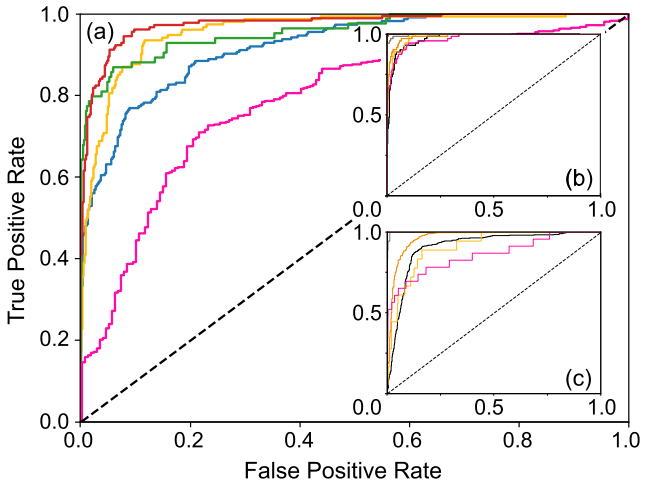

FIG. 2. Receiver operator characteristic graphs demonstrating the overall performance and area a under curve as the confidence threshold required to make a positive prediction is varied for CNN ensembles. Classification performance is compared for scanning tunneling images of (a) H:Si(100), (b) Au(111), and (c) Cu(111). A perfect classifier has an area under a curve of 1, with guessing 0.50 (black dashed line, theoretical). For (a), we find asymmetry/dimer = 0.92 (blue), individual atoms = 0.96 (yellow), rows = 0.95 (green), tip change = 0.79 (pink), and generic defect = 0.98 (red). For (b) and (c), respectively, we find impurities = 1.00, 1.00 (gray), double tip = 0.98, 0.91 (black), corruption = 1.00, 1.00 (brown), individual atoms = 0.98, 0.91 (yellow), step edges = 0.99, 0.97 (orange), and tip change = 0.97, 0.86 (pink).

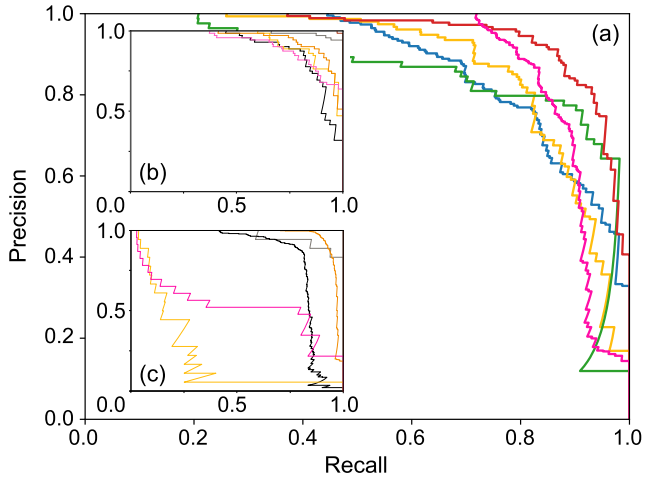

FIG. 3. Precision-recall graphs to demonstrate the overall performance of ensemble CNNs when classifying the known tip states for images of (a) H:Si(100) and (b) Au(111) (c) Cu(111) as the confidence threshold required to make a positive classification was varied. Precision is the percentage of true positives compared to total positive classification, while recall is the percentage of positive classifications that have been correctly identified as positive. Some tip states are desirable and surface specific, such as asymmetry/dimer (blue), individual atoms (yellow), and rows (green). However, tip changes (pink), impurities (gray), double tips (black), corrupted (brown), step edges (orange), and generic defect (red) are undesirable. Performance is strong, except for individual atoms and tip change in (c).

accuracies of 93% and 91%, mean precisions of 0.96 and 0.97, and AUROCs of 0.98 and 0.98, respectively. Cu(111) did not improve owing to poor PR of individual atoms and tip changes, as visible in Fig. 3(c).

However, although tip changes were classified respectably with Au(111) and Cu(111), this was not the case with H:Si(100). When including the separate binary network to cover all classes for H:Si(100), performance was significantly poorer, with a balanced accuracy of 77%, mean precision of 0.88, and average AUROC of 0.92. This is particularly visible in Fig. 2, with the tip change category having an ROC line below the other categories and AU of 0.80. This is likely because when augmentations were limited to simple flips and noise, the network rapidly overfit, and learning had to be stopped earlier. Regardless, few false positives were made for tip changes when increasing confidence thresholds. This is because precision was only seen to decrease at high values of recall, as visible in Fig. 3. As such, tip states could still be distinguished with a low false positive rate by requiring a high confidence threshold.

IV. CONCLUSION

We have successfully trained CNNs capable of classifying numerous desirable and undesirable STM tip states for multiple surfaces. We achieve significantly greater all-around performance than other supervised learning techniques and an even stronger ability to differentiate good and bad tip apices. The protocol is also likely applicable to a broad range of other SPM techniques, given the relative similarity of images produced by these methods.

However, there are a number of limitations to the approach which limit its general applicability. Importantly, each trained

ensemble is only applicable to a single surface (and in turn requires large amounts of training data of each surface). We also find that without significantly expanded datasets not all surfaces are equally suitable for CNN classification. New datasets must therefore be manually created for each surface studied. This not only makes practical implementations of automatic recognition on other surfaces time-consuming and inconvenient but also requires the surface to be well studied in advance. It would therefore be nontrivial to use this protocol to explore STM tip states of previously unexplored surfaces. Furthermore, the low number of human classifiers was also problematic. Were more human classifiers available, the networks should have been trained on the entire multilabel dataset and then scored based on a cross-entropy of average classifications. Performance could also be improved further with the addition of more training data and potentially combined with time-dependent data to allow for real-time classification and tip enhancement during scanning.

Regardless, the CNN protocol in its current state will enable a fully autonomous *in situ* approach to selecting and observing a variety of tip states during imaging, spectroscopic, and atomic manipulation experiments in STM. Although such an approach would still require using the unreliable and time consuming tip correction methods used today, future work aims to also automate this aspect by combining CNN ensembles with other machine learning methods.

SUPPLEMENTARY MATERIAL

Included in the supplementary material is a pictorial demonstration of the SqueezeNet network for H:Si(100). After inputting an image manually classified to have atomic resolution, shown are the output for each convolutional layer and the final network confidence for each resolution category.

ACKNOWLEDGMENTS

O.G., F.J., P.D'H., and P.M. acknowledge funding from the Engineering and Physical Sciences Research Council via Grant No. EP/N02379X/1. I.S. acknowledges funding from NWO via Grant No. 16PR3245. O.G. and P.M. also thank Bob Wolkow, Mohammad Rashidi, and Jeremy Croshaw of the University of Alberta, and Ken Gordon, President and CEO of Quantum Silicon, Inc., for sharing data and a series of very helpful personal communications.

REFERENCES

[1] G. Binnig, H. Rohrer, C. Gerber, and E. Weibel, Phys. Rev. Lett. **49**, 57 (1982).
[2] A. A. Gewirth and B. K. Niece, Chem. Rev. **97**, 1129 (1997).
[3] B. Voigtländer, *Scanning Probe Microscopy* (Springer, 2016).
[4] M. Rezeq, J. Pitters, and R. Wolkow, J. Chem. Phys. **124**, 204716 (2006).
[5] J. C. Straton, T. T. Bilyeu, B. Moon, and P. Moeck, Cryst. Res. Technol. **49**, 663 (2014).
[6] Y. Wang, J. I. Kilpatrick, S. P. Jarvis, F. M. F. Boland, A. Kokaram, and D. Corrigan, IEEE Trans. Image Process. **25**, 2774 (2016).
[7] R. A. J. Woolley, J. Stirling, A. Radocea, N. Krasnogor, and P. Moriarty, Appl. Phys. Lett. **98**, 253104 (2011).
[8] M. Rashidi and R. A. Wolkow, ACS Nano **12**, 5185 (2018).
[9] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, in *International Conference on Medical Image Computing and Computer-Assisted Intervention* (Springer, 2013), pp. 411–418.
[10] M. Lochner, J. D. McEwen, H. V. Peiris, O. Lahav, and M. K. Winter, Astrophys. J. Suppl. Ser. **225**, 31 (2016).
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Curran Associates, Inc., 2012), pp. 1097–1105.
[12] M. Møller, S. P. Jarvis, L. Guérinet, P. Sharp, R. Woolley, P. Rahe, and P. Moriarty, Nanotechnology **28**, 075302 (2017).
[13] A. Sweetman, S. Jarvis, R. Danza, and P. Moriarty, Beilstein J. Nanotechnol. **3**, 25 (2012).
[14] M. Fuechsle, J. A. Miwa, S. Mahapatra, H. Ryu, S. Lee, O. Warschkow, L. C. Hollenberg, G. Klimeck, and M. Y. Simmons, Nat. Nanotechnol. **7**, 242 (2012).
[15] G. Lopinski, D. Wayner, and R. Wolkow, Nature **406**, 48 (2000).
[16] Z. Sun, M. P. Boneschanscher, I. Swart, D. Vanmaekelbergh, P. Liljeroth *et al.*, Phys. Rev. Lett. **106**, 046104 (2011).
[17] P. H. Jacobse, A. van den Hoogenband, M.-E. Moret, R. J. Klein Gebbink, and I. Swart, Angew. Chem., Int. Ed. **55**, 13052 (2016).
[18] The Au(111) and Cu(111) data were acquired by the Utrecht group, whereas the H:Si(100) images were obtained at Nottingham and classified separately and by different researchers.
[19] I. Jindal, M. Nokleby, and X. Chen, in *2016 IEEE 16th International Conference on Data Mining (ICDM)* (IEEE, 2016), pp. 967–972.
[20] B. Frénay and M. Verleysen, IEEE Trans. Neural Networks Learn. Syst. **25**, 845 (2014).
[21] X. Zhu and X. Wu, Artif. Intell. Rev. **22**, 177 (2004).
[22] C. E. Rasmussen, *Advanced Lectures on Machine Learning* (Springer, 2004), pp. 63–71.
[23] J. Stirling, R. A. Woolley, and P. Moriarty, Rev. Sci. Instrum. **84**, 113701 (2013).
[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, J. Mach. Learn. Res. **12**, 2825 (2011).
[25] T. Saito and M. Rehmsmeier, PLoS one **10**, e0118432 (2015).
[26] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, and K. Keutzer, Squeezenet v1. 1 model, 2017.
[27] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *CoRR* (2015), Vol. abs/1511.07289.
[28] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, in *International Workshop on Multiple Classifier Systems* (Springer, 2003), pp. 125–134.
[29] We also note that although Rashidi and Wolkow used sigmoid and categorical cross-entropy functions,[8] and they were not the standard choices for their binary classification scheme because the confidence of both the positive and negative classifications could be high, degrading performance.
[30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *CoRR* (2014), Vol. abs/1412.6980.
[31] M. D. Zeiler, e-print arXiv:1212.5701 (2012).
[32] J. Duchi, E. Hazan, and Y. Singer, J. Mach. Learn. Res. **12**, 2121 (2011).
[33] S. Ruder, e-print arXiv:1609.04747.
[34] T. G. Dietterich, in *International Workshop on Multiple Classifier Systems* (Springer, 2000), pp. 1–15.
[35] F. Chollet *et al.*, Keras, https://keras.io, 2015.
[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2015), pp. 1–9.
[37] We note that although it seems like asymmetries/atoms should be grouped because they are visually similar, the end goal was to observe atoms.