Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 3043

"Twisting" the data: a universal machine-learning approach to classify single-molecule curves and beyond

C. Roldán-Piñero, *[ae] M. Teresa González, [b] Pablo M. Olmos,[c] Linda A. Zotti [ade] and Edmund Leary *[c]

We present a new automated supervised procedure trained to classify both conductance-voltage ($G(V)$) curves and conductance-distance ($G(z)$) traces generated in single-molecule junctions to a high degree of confidence. Compared to unsupervised methods, our approach, involving a convolutional neural network (CNN), is vastly superior as it allows core shapes to be recognised by ignoring differences in scale and is relatively insensitive to conductance jumps. A key aspect is the transformation of curves into a spiral image map, which allows us to separate various fundamental $G(V)$ and $G(z)$ shapes from datasets containing tens of thousands of curves. Moreover, by using transfer learning, training requires little input data compared to other approaches. This is extremely advantageous as it reduces training time by many orders of magnitude and means the model can be trained on user-selected shapes, including rare types. This contrasts with arbitrary class-assignment, instead basing classification on a sound physical understanding of the system. Furthermore, as there is no minimum class population requirement, our method can be used to find rare events with a high degree of confidence. As an example, we used our procedure to find, with a minimum 66% confidence level, a class of $G(V)$ curves which are parabolic at low bias but flat at high bias. Such curves make up just 4% of the total, and would be very difficult to detect cleanly with unsupervised methods. This gives insights into the electron transport behaviour at high-bias because we can now easily quantify the types of curves present. Thanks to its universality, this opens up new possibilities in general signal processing and the identification of rare and important events.

Received 19th May 2025
Accepted 18th August 2025

DOI: 10.1039/d5dd00207a

rsc.li/digitaldiscovery

Introduction

Huge datasets are becoming commonplace in research, especially within molecular electronics. This calls for automated classification procedures which can sort the data into groups based on meaningful criteria. Without such methods, overall results simply express average behaviour, with a loss of nuance and detail. Rare events, which may highlight interesting and insightful behaviour, will be overlooked. Machine-learning (ML) offers a way to deal with large volumes of data by sorting individual measurements into subsets based on particular shapes or patterns. ML is becoming increasingly important in many scientific fields such as particle physics, neuroscience and brain imaging, engineering, materials science, electrochemistry and battery research.[1–7] In molecular electronics, and more specifically in single-molecule electronics, large datasets are essential for exploring the many configurations of a molecular junction. The potential geometryspace of a single-molecule junction (represented in Fig. 1a) formed by repeatedly making and breaking metal–metal contacts in the presence of an adsorbed species (known as the breakjunction, BJ, method) is enormous, and thousands/tens of thousands of individual junctions must be created and analysed.[8–10] There has been significant effort to categorise conductance *versus* distance ($G(z)$) curves using various "hands-off" approaches. In this method, as the electrodes are separated, the conductance is monitored as a function of the distance, and plateaus in G appear when a molecule (like that shown in Fig. 1b) bridges the gap. Examples of $G(z)$ curves are shown in Fig. 1c. We have previously demonstrated a simple routine (a plateau-separation algorithm, PSA) which searches for conductance plateaus and can separate exponential background tunnelling from the molecular plateau events.[11,12]

More recently, ML techniques have been applied to classify $G(z)$ traces which look at their overall shape rather than isolated segments. These techniques can be divided into two broad

[a]*Departamento de Física Teórica de la Materia Condensada, Universidad Autónoma de Madrid, E-28049 Madrid, Spain. E-mail: carlos.roldanp@uam.es*
[b]*Fundación IMDEA Nanociencia, E-28049 Madrid, Spain. E-mail: edmund.leary@imdea.org*
[c]*Departamento de Teoría de la Señal, Universidad Carlos III de Madrid, Madrid, Spain*
[d]*Instituto Nicolás Cabrera (INC), Universidad Autónoma de Madrid, E-28049 Madrid, Spain*
[e]*Condensed Matter Physics Center (IFIMAC), Universidad Autónoma de Madrid, E-28049 Madrid, Spain*

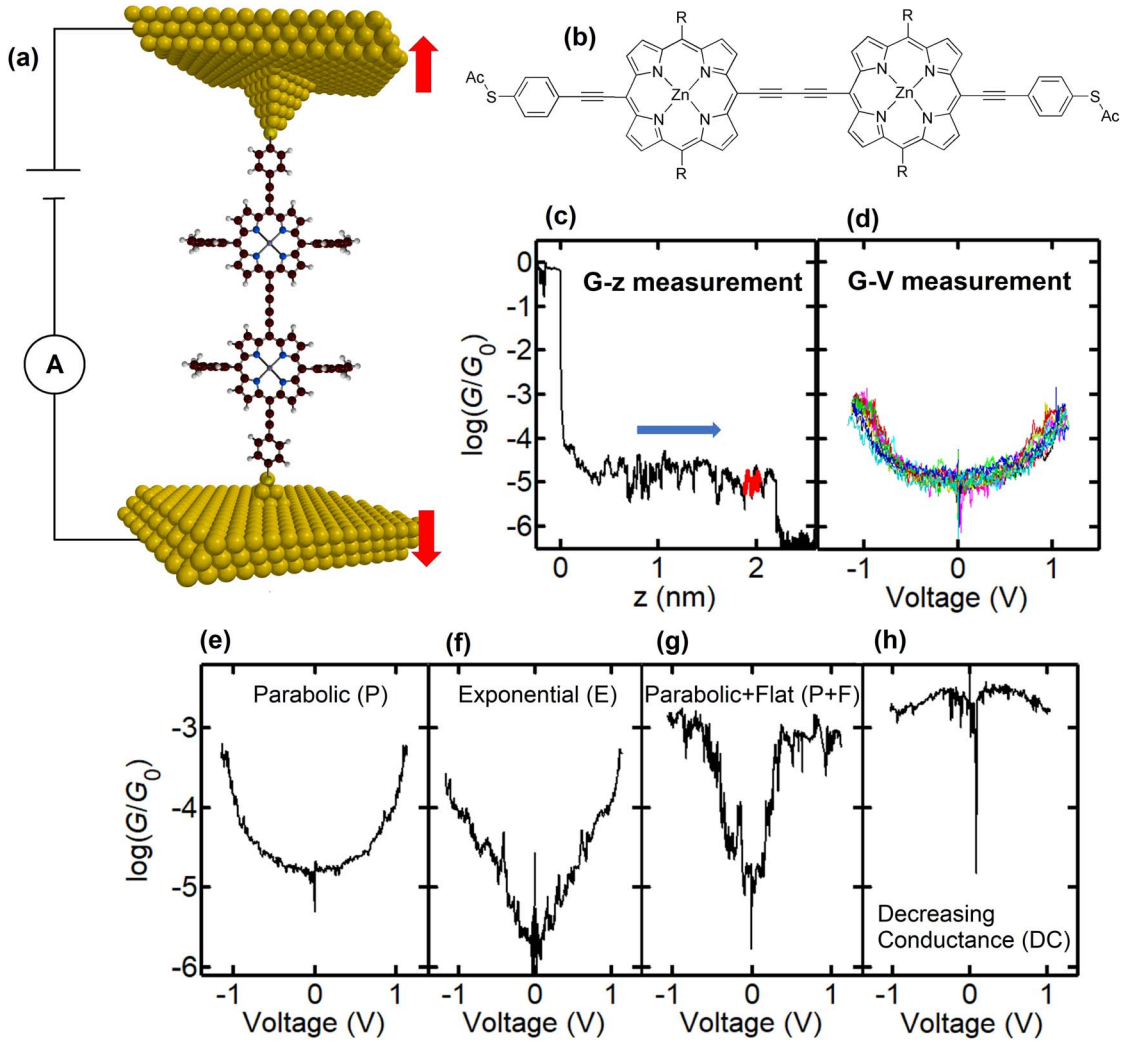

Fig. 1 (a) Schematic of a Au–P2–Au molecular junction. Red arrows indicate the direction of the electrodes during the conductance-distance measurement. (b) Chemical structure of P2. (c) An example of a $G(z)$ curve for P2 in which $G(V)$ curves are recorded. The highlighted red region shows the positions at which the $G(V)$ curves shown in (d) where recorded. Note, G_0 is the quantum of conductance, equal to 77.5 μS. (d) $G(V)$ curves recorded between $V = \pm 1.1$ V. (e–h) Examples of different $\log(G/G_0)$ profiles observed when high-bias voltages are applied. (e) Parabolic (P), (f) exponential (E), (g) decreasing-conductance (DC) and (h) parabolic + flat (P + F).

categories: supervised and unsupervised machine learning (SML and UML respectively). SML involves training a model using preclassified data, *i.e.* a control group, and then tries to infer for new data based on similarities with the control group. Typically, this requires a large amount of training data. Alternatively, UML aims not to impose any preconceived notions about the data, and does not require training. Instead, it looks for commonalities throughout a large-enough dataset, grouping the data based on different metrics.

UML approaches, like *k*-means, have been successfully applied to $G(z)$ traces, identifying physically meaningful groups.[13–17] This includes the presence of different conductance pathways through porphyrin molecular junctions,[18] and different conformations of amino acid molecular junctions have also been distinguished this way.[14] There are, however, significant limitations of this approach as, in particular, it fails to group the smaller population shapes due to the uniform effect.[19,20] The problem of choosing the optimal number of cluster groups is especially tricky as, *a priori*, no physical relevance is assigned to the groups. This raises the possibility of forming groups where the physical significance is unclear and, moreover, rarer events will be ignored.

SML algorithms have started being used to identify $G(z)$ traces, but only in a very limited way. Several groups have implemented models that can reliably identify molecule-free junctions displaying a well-defined shape in which the conductance decreases exponentially with electrode separation.[21,22] As far as we are aware, however, no SML algorithms have been directed towards identifying specific shapes of $G(z)$ traces, which is a more complex task. Furthermore, training requires thousands of individual traces, making manual labelling a cumbersome task. It also means that rare events cannot be analysed.

Apart from $G(z)$ traces, the voltage dependence of the conductance of molecular junctions is an equally important characterisation tool.[10,23,24] Despite this, $G(V)$ spectroscopy has, unlike $G(z)$ measurements, received no attention in terms of "hands-off" classification approaches. $G(V)$ curves are obtained by fixing the inter-electrode distance and varying the applied voltage (examples of such curves are shown in Fig. 1d). This gives valuable information regarding the transport mechanism and the energetic position of the closest molecular energy level(s) with respect to the Fermi level. The technique is applied both to single-molecule as well as monolayer devices.[25,26] The shape of $G(V)$ curves depends on the structure of the compound, but is also influenced by the contacts and the temperature. Moreover, the shape can give clues as to the nature of the charge transport mechanism.[27–29]

Various shapes can appear, which have been related to redox events (gain or loss of electrons) for molecules with small HOMO–LUMO gaps (*i.e.* the energetic difference between the highest-occupied and lowest unoccupied molecular orbitals).[8,30,31] In the charged state, the shape of the $G(V)$ curves differs significantly compared to the neutral state. For a neutral junction, G(V) curves are typically approximately parabolic. In the charged state, however, different shapes appear. When plotted as $\log(G(V))$, typical shapes include: parabolic (P), exponential (E), parabolic with saturation at high-bias (parabolic + flat, P + F) and also flat/slightly decreasing conductance (DC).[30] Examples are shown in Fig. 1e–h. Combinations of these shapes also appear, probably due to asymmetries at the molecule–electrode interface, as well as stochastic switching between different shapes. There are also curves that have no clearly-definable shape.

These factors, along with the intrinsic conductance variability (due to different metal-molecule coupling strengths) make it difficult for UML algorithms to perform a meaningful classification of $G(V)$ curves. Such algorithms classify curves based on their numerical similarity, which can work for $G(z)$ traces, but, in the case of $G(V)$ curves, it would be more meaningful to group curves based on their generic shape, which relates directly to the mechanism of conductance.

In this article, we solve some of the major limitations of typical SML models applied to data analysis by bringing together two separate techniques: spiral image mapping and transfer learning (TL). The former is a technique which converts 1D curves/traces into 2D images, which are more suitable for use with CNNs trained on images. The latter concept, TL, is based around the idea that pattern/image recognition tasks are transferable, and previously deeply-trained networks can be applied to seemingly unrelated recognition tasks without the need for significant retraining. The optimal use of previously deeply-trained CNNs in this case is thus enabled by the 1D to 2D conversion of data *via* spiralization. This is a major leap for a number of reasons. One, it hugely reduces training times, which opens the door to rare trace identification. Secondly, it means it is possible to focus on identifying the core shapes in 1D signals rather than looking for absolute similarities. This is very important as it is the shape which often carries mechanistic information regarding the transport process, not the absolute conductance values.

Methods

Using CNNs on x–y traces

Initially, in order to address this issue, we were inspired by Bro-Jørgensen *et al.*[32] to try two different techniques, namely parametrization and histogram-based approaches which are reported in Sections S4 and S5 of the SI, respectively. After testing both of these methods, our conclusion was that in order to achieve good performance, these models need to be highly tailored to the specific task, which would be inefficient for our aim of being able to identify both $G(V)$ curves and $G(z)$ traces. We thus turned to pre-configured neural networks as an approach which is efficient and can be applied without modifications to different conductance data classification tasks.

As discussed, the use of neural networks, more specifically convolutional neural networks (CNNs)[32] and recursive neural networks,[33] in $G(z)$ trace classification has already been reported, yielding promising results. In this paper, we focus on the use of CNNs, as they are known to be optimal for the identification of images,[34] which better suits the task of classifying the shape (a visual property) of $G(V)$/$G(z)$ curves/traces. However, they typically require a huge number of traces in order to be trained properly, which precludes manual sample labelling for training purposes. Being able to label samples manually is advantageous in that it allows a human to decide if a particular shape has a physical significance or not. Moreover, we wanted to reduce computational demand in training so that a basic desktop computer may perform the training and subsequent classification as part of the standard data analysis workflow. As such, we sought an approach that could reduce the amount of training data required. All these factors could be resolved by using transfer learning (TL, see Section S2 in the SI)[35] on very deep pretrained CNNs, trained for the identification of general objects in images.[36] For the choice of the final classification step, we tested logistic regression, support vector machines, stochastic gradient descent, k-neighbours, decision trees, random forest and a single fully connected layer; for all but the last we performed hyperparameter optimization (see Section S6 of the SI). Although all displayed good performance, we found that a single fully-connected layer offered the best overall accuracy. We used the Adam algorithm for optimization[37] with a constant learning rate of 0.001. For further detailed information on our approach, please see Section S2 of the SI.

In order to generate images from curves, we took inspiration from[38] where they showed that U-nets[39] for noise reduction work well with a greyscale spiral representation of the curves. To illustrate this, Fig. 2a shows an artificially-generated $G(V)$ curve (solid brown line) which has been divided into bins whose heights depend on the conductance. To generate an image, we assign a shade of grey to each data bin so that the minimum G value corresponds to a black pixel and the highest to a white one. Starting from 0 V, these pixels are plotted in the image starting from the centre and moving outwards in an anticlockwise fashion. The central pixel thus corresponds to the lowest

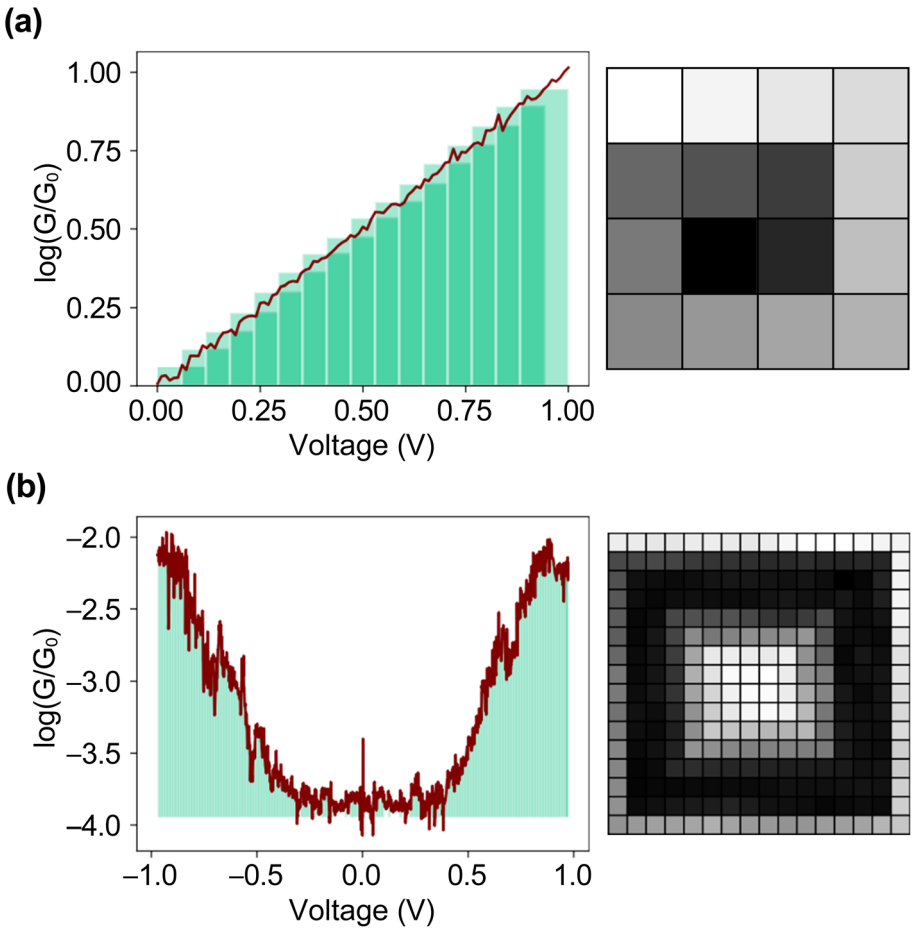

Fig. 2 (a) Example of an artificial $G(V)$ curve (left) with the corresponding spiral image mapping (right). Superimposed is a bar plot representing the average value of each interval. These values are subsequently used for the image generation. The black lines between the pixels are drawn for clarity and do not feature in the final image. (b) U-shaped $G(V)$ curve (left) with its corresponding spiral image mapping (right).

voltage, and each subsequent pixel corresponds to a higher voltage. Fig. 2b shows an example of spiral image generation for a real $G(V)$ curve. In this case, the first pixel is generated from the most negative voltage, with subsequent pixels for progressively more positive voltages (please see Section S2 of the SI for a pseudocode implementation of the spiral mapping). We will show that this mapping works well for both $G(V)$ and $G(z)$ curve/trace classification. It also helps reduce the impact of high-frequency noise in individual curves. Following this, we fed these images to our pretrained network and fitted the last layer.

We have tested various available pretrained CNNs in the PyTorch API.[40] We include a comparison of these in Section S3 of the SI. All performed similarly. For our results, we selected ConvNext Tiny[41] for its balance between high accuracy and low feature extraction time. It gives a total of 768 features.

We note that CNN-based models operate largely as "black boxes", making it difficult to explain their individual decisions in a physically meaningful way. Techniques such as Grad-CAM, saliency maps, and related methods may offer partial insight into which image regions or features influence a given prediction, but applying these approaches rigorously would require a dedicated study in of itself. For now, our priority is to ensure that the model produces accurate and consistent results overall. Future studies may be performed to try and interpret individual classifications at a deeper level.

In this work, we focus on data obtained using the STM breakjunction technique on the previously reported compound **P2** (shown in Fig. 1b).[42] Other compounds used for training and further analysis are described in Section S1.1 of the SI.

Benchmarking the models

To provide an estimation of the performance of the model we used a train-test split approach. We first split the data into two groups containing eighty percent and twenty percent respectively of the total number of samples in the training data. All data splits were done preserving the relative abundances of each curve type, *i.e.* in a stratified manner. Then, the model was trained with the larger group which was then used to predict the shapes of the smaller group. This way we benchmark how good the model will perform against previously unseen curves.

To convey the quality of the model visually, we built confusion matrices (CM), defined so that the element in the ith row and jth column corresponds to the number of times an instance with target variable of type i (the true shape) is classified as type j (the shape determined by the model). The more the matrix

approaches a diagonal matrix, the higher the classifier accuracy. Note that, because we manually selected curves for the training data, these are likely fairly ideal examples, and so we would expect slightly lower accuracy when we come to analyse unseen data compared to what we estimate based on this initial testing.

Lastly, we used the whole set to train a final model was subsequently used to predict the shape of a large set of unclassified curves.

Results and discussion

G(*V*) curve classification

To train the model for the classification of *G*(*V*) curves, we used 728 manually pre-classified curves from several porphyrin compounds (see SI Fig. S1 for 2D histogram representations of the data used). The data were measured with voltage windows of either ±1.0 V or ±1.2 V. This includes both ring-fused (**fP2** and **fP3**) and alkynyl-coupled (**P2**) compounds with thioacetate anchoring groups. The structure of **P2** is shown in Fig. 1b, whilst the structures of **fP2/3** have been published elsewhere.[42] The use of different compounds helps to make the model less compound specific. The training set was initially broken down into 335 P, 170 E, 80 DC and 145 P + F curves (representing 80% of the training data for each shape). These data were used to train the model, whilst the remaining 20% were used for testing. The results are reported in Fig. 3a where we find the model accurately identifies the majority of the shapes, with the exception of DC which is often mistaken with P + F. The model

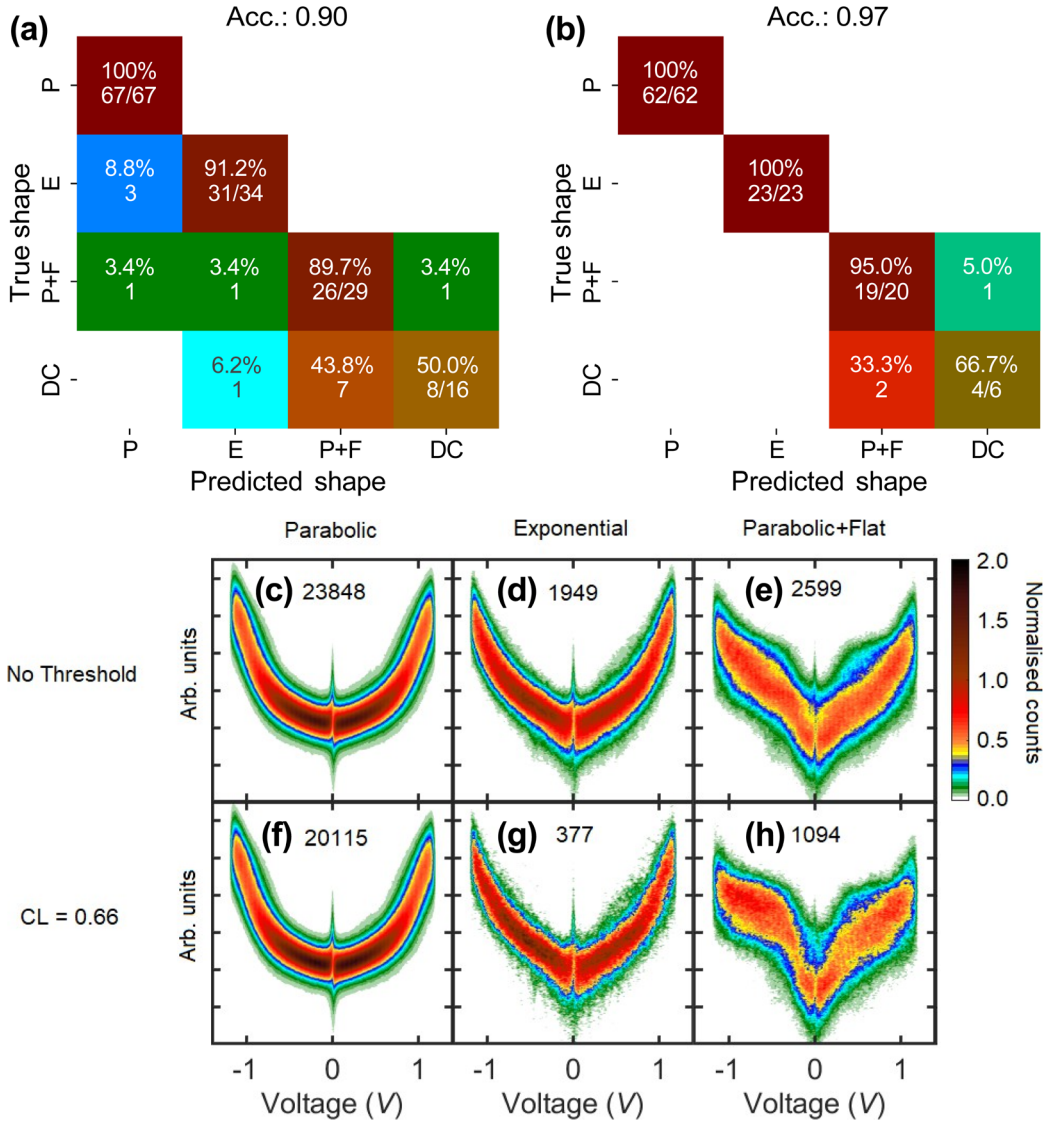

Fig. 3 (a) Confusion matrix of the model without confidence level (CL) threshold imposed. Four categories were considered: parabolic + flat (P + F), decreasing conductance (DC), parabolic (P) and exponential (E). The accuracy is defined as the number of correct matches divided by the total number of curves. Out of 728 curves, 80% (582) were used to train the model and 20% (146) were used to test it. (b) Confusion matrix of the model with 66% confidence threshold imposed. The same four categories were considered (c–e) Histograms of 28 396 curves classified by the model without any imposed CL threshold and (f–h) with a 66% threshold applied. The data have been standard scaled for visualisation purposes. A total of 21 586 curves out of 28 396 passed the filter. Note, the DC group output has been ignored.

correctly identified 100% of P curves, 91% of E curves and 90% of P + F curves. The model managed only to identify, however, 50% of the DC curves, which is still much better than random, where it would be expected to get about 25% correct. For DC, the performance is probably a result of the very low number of curves used for training compared to the other shapes, which is due to the rarity of this shape in the original data pool. This results in a slightly under-representative sample, highlighting the lower limit for training.

A useful property of using a dense layer as the classification model is that the results at the final layer may be normalised and interpreted as a confidence level (CL) between 0 and 1 for the classification. This will give an indication of how sure the model is that the analysed curve belongs to the predicted group. Each curve will be assigned a CL for every shape being assessed. For example, in the case of the four shapes considered here, if the model assigns a 0.85 CL to a curve being P, then the sum of the CLs for the curve being E, P + F or DC would come to 0.15. Note that this is not an absolute measure of confidence, rather it is more a relative estimation within the bounds of the shapes considered. As such, care should be taken when a trace is a mixture of two or more general shapes, as similar CLs will be assigned for each relevant group, bringing the highest confidence down. This does not necessarily mean that the model does not recognise the curve's features, rather that the curve may be classified in more than one group (*vide infra*). Nonetheless, a high enough CL should mean that the shape truly falls into the predicted category. In order to apply the CL, one must define a threshold. The choice of the threshold is, however, rather arbitrary. Here, we found that a threshold of 0.66 gives good balance between increasing accuracy (from 0.90 to 0.97) and not rejecting too many curves (see Section S7 in the SI).

In Fig. 3b we show the associated CM for the classification results on the training data with a threshold applied. The fractions expressed along the diagonal refer to the number of correctly assigned curves (numerator) and the total number assigned to the particular shape (denominator). After applying this threshold, the accuracy increases to 100% for E curves and 95% for P + F (*n.b* it was already 100% for the P shape before applying the CL). The number of classified curves decreases for all categories, from 146 before the threshold to 111. This means that 35 curves were assigned CLs less than 0.66. For the DC group, there is still some confusion between it and the P + F shape. The accuracy increased to 67%, which is an improvement, but despite this, we decided not to proceed further with the identification of DC curves as we felt the model would require more data for optimal training. Instead, we focus on the classification of the three shapes with the most training data (P, E and P + F). Based on the training results, we can see that good training of $G(V)$ curves requires a minimum of roughly 150 curves, which is still extremely low, but the DC shape, with just 80 curves, falls just short of this level.

After testing the performance of our model, we proceeded by applying the final model (trained with the whole training set) to a much larger dataset containing 33 498 curves of unknown shape. We still trained the model with the four shapes, but we ignored the classification output for the DC shape henceforth. Panels c-e of Fig. 3 show the two-dimensional histograms built from the resulting classified curves. With no CL applied, the model assigns each curve to one of the predefined groups. This means that even curves with relatively low confidence will be assigned to one of the groups. Note that for representational purposes the curves were standard-scaled before plotting in histograms, *i.e.* each of the curves was shifted and scaled in the y axis (after classification) to have zero mean and unit standard deviation. Despite the fact that we are "forcing" the model to place all curves within the predefined groups, all three histograms represent their respective expected shapes quite well. This shows that the dataset can, in fact, be fairly well represented by these three groups (where the DC type of curve most likely contributes with a tiny percentage). It is clear that the P curves have the best-defined histogram, closely followed by the E class. The P + F group is the only group in which the shape in the 2D histogram is not quite in line with the anticipated shape. This is not too surprising, however, given that this shape was harder for the model to classify during training than P and E shapes.

After this, we applied a 0.66 CL threshold, and the corresponding histograms are shown in panels f–h of Fig. 3. Totally, 21 586 curves passed the filter (76%). For the P group, only about 16% of the initially selected curves were eliminated, showing the ease to which the model can recognise this shape. For the P + F group, the number of rejected curves increased to 58%, and the 2D histogram now much more clearly resembles the anticipated shape with a central parabolic part and flatter regions towards higher bias. As we initially force the model to choose to put the curves into one of several specified groups, it makes sense that many are rejected after applying the CL, because we do not expect all curves to have one of these shapes. For the E group, such a visual improvement in the 2D histogram is not as obvious as for the P + F group, despite the elimination of many curves (81%). Looking at the rejected curves (individual examples shown in SI Fig. S10) there are many that have a shape somewhat intermediate between parabolic and exponential. We decided to run an unsupervised clustering algorithm on these rejected curves to look for specific trends. Fig. S11 shows the results of the clustering, where we found five groups (Groups 1–5) with varying degrees of asymmetry and parabolic/exponential character. Such curves can be described as intermediate between exponential and parabolic, either having an overall average between the two extremes, or containing both shapes on different sides of 0 V. As the CL values are relative, then if a curve is intermediate between one shape and another, the CLs for the trace being of either type will be reasonably close. This means the model will assign a CL close to 0.5 for the curve being P or E, which is why, when we choose a CL = 0.66, such curves will be rejected. Furthermore, as the model was trained on symmetric curves, it again makes sense that the model assigns a lower CL to the partially-exponential curves compared to more symmetric ones. This is an interesting results as, despite the model not being trained on asymmetric curves, it can still effectively separate those with partial E character from purely P and P + F shapes. In the future, we envisage training the model

with asymmetric curves, which would allow the model to identify these curves more confidently.

What this shows is that applying a CL is extremely useful in the data classification process. For curves with E character, this ability allows us initially to separate curves with any level of E character, which can be then be refined by applying a CL. In this sense, by applying a CL, we obtain only the curves with well defined E character on both sides of 0 V due to the symmetry of the curves used in training. The rejected curves can be said to have, therefore, partial E character.

Previously, we have shown that applying a bias voltage of 1.2 V to a fused porphyrin trimer (**fP3**) resulted in most of the $G(V)$ curves having non-parabolic shapes. This was straight forward to assess *via* a visual inspection of the data. Here, by using our CNN approach, we have shown that under the same conditions, a porphyrin dimer, with rings connected by butadiyne groups, displays around 30% non-parabolic $G(V)$ curves (the percentage of P curves is about 70% directly out of the model). Of these nonparabolic curves, we show that about 6% have some level of exponential character, and about 8% have parabolic + flat character. The remaining 16% remain, as yet, unclassified. Applying a CL to the data, we can refine the E group further by focussing on the most-symmetric curves. This allows us to identify 377 symmetric E curves, which represents just 1% of the total. Such a low percentage would be extremely difficult to classify either manually due to the size of the dataset, or by UML algorithms due to the small population size. It would also be difficult without the application of CLs. The result makes physical sense considering that **fP3** has a much smaller HOMO–LUMO gap (about 0.8 eV) compared to **P2** (1.7 eV) making redox events more likely. For **P2**, if we assume that the Fermi level sits close to the middle of the HOMO–LUMO gap, we can infer that a bias voltage of roughly 1.7 V would be required to align the closest molecular level fully with the gold chemical potential. In the measurement, the bias voltage was ramped between $V = \pm 1.1$ V, which is significantly lower, explaining why the majority of curves are parabolic. On the other hand, the presence of about 30% non-parabolic curves implies several possibilities. One is that the Fermi level may sit closer to one of the frontier molecular-based levels than the other (meaning less than 1.7 V would be required to inject charge). It may also point to significant fluctuations in level alignment (on the order of 0.5–1.0 eV) so that, occasionally, a molecular level is brought into resonance. Further work is required using a varying voltage window to analyse these possibilities further.

$G(z)$ trace classification

Now we turn to the task of classifying $G(z)$ traces using our supervised methodology. As mentioned earlier, this problem has been tackled by several groups using SML and UML techniques. A pioneering example is that of Lemmer *et al.*[16] who used a clustering approach (unsupervised) to find similarities between $G(z)$ traces and identify sub-populations contained within large datasets. SML techniques like ours cannot be used to group unseen data without prior training, but when the shape of the $G(z)$ traces are generally known, they should have a distinct advantage. Plateau-free junctions (*i.e.* those in which there is no molecular junction) typically display a characteristic exponential conductance decrease with distance, often termed "tunnelling" traces (see SI Fig. S2a). Their identification and removal is, thus, well-suited to a supervised approach.[22] $G(z)$ traces containing molecular plateaus also often fall into one of two categories, "continuous" and "broken-plateau" (see SI Fig. S2b and c). "Continuous-plateau" traces correspond to molecular junctions in which the molecule remains attached during the entire molecular junction elongation process, until the electrode separation is larger than the molecular length, and the conductance along the plateau remains fairly constant. "Broken-plateau" traces, on the other hand, correspond to junctions in which, for a period, no molecule forms a bridge, either initially or during its evolution, which causes the conductance to drop periodically well below the value of the plateau.[12] Such generic shapes should be well suited to a SML approach. The reason why some molecular junctions break and reform whilst others do not is often unclear, making it important to have reliable ways to separate these classes of trace.

To train the model for $G(z)$ identification, we hand-selected a small set of 242 traces belonging to three categories we then extracted 80% from each type: broken (49), continuous (74) and tunnelling (71). We stress this is an extremely low number, which we could do thanks to our use of TL. For comparison, 100 000 traces had to be used in a previous study.[22] This allowed us to select the training data manually, as opposed to the clustering approach used in the study by van Veen *et al.*[22] We trained the model as previously, using the train-test split approach. Fig. 4a shows the CM for the trained model. It is important to underline that even with such a low number of training traces we are able to obtain almost perfect scoring (*i.e.* a near-perfectly diagonal CM). Only two broken traces were misclassified as continuous (see Section S8 of the SI for more details on the training of the model and applying CLs). Remarkably, compared to the training for $G(V)$ traces, we were able to train the model well with even fewer $G(z)$ traces.

Next, we ran our model against a large dataset containing 10 807 unclassified $G(z)$ traces. Initially, we did not impose any CL threshold so that we could compare the raw performance of the model against a custom plateau-separation algorithm (PSA) which is a simple program that sorts $G(z)$ traces into the same three groups based on a few parameters[11] (details given in Section S10 of the SI). For this purpose we plotted the histograms of the CNN method and the PSA in panels b–d and e–g of Fig. 4, respectively (the values in each panel are the number of traces in each histogram).

Generally, there is good agreement between the two procedures. Both agree the majority of traces correspond to tunnelling, with the next largest group being continuous-plateaus and finally broken-plateaus. A visual inspection, however, shows that although the tunnelling and the broken groups are quite similar, the continuous plateau histogram differs in the region below the main conductance cloud (*i.e.* below $\log(G/G_0) = -5$ down to the noise). The histogram from the CNN method contains few data points in this region, which is to be expected for continuous junctions when the molecule remains attached

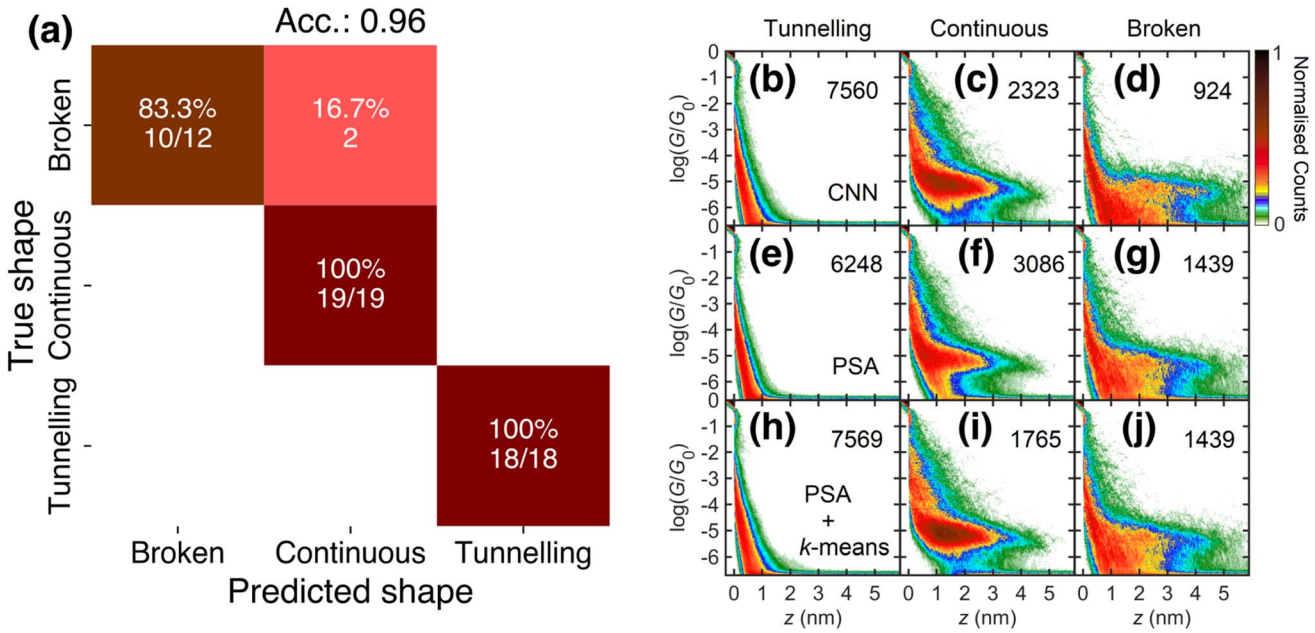

Fig. 4 (a) Confusion Matrix showing the results of the model training phase. Three categories were considered: Broken, Continuous and Tunneling. Of the total 242 training traces 80% (194) were used for training and 20% (48) for benchmarking. (b–d) Histograms of the traces classified by our CNN model on a large dataset containing 10 807 traces. (e–g) Histogram of the same traces classified by the PSA model. (h–j) Histogram of traces classified by the PSA model with a subsequent *k*-means step applied to the continuous-plateau traces.

until final break-down. On the other hand, the PSA-generated histogram shows a clear extension of the background tunnelling slope below the molecular cloud. This suggests that some traces containing tunnelling signal below the main plateau level are being considered as plateau traces by the PSA (either tunnelling-only traces or broken-plateau traces with an initial tunnelling part). This is consistent with the lower number of traces classified as tunnelling by the PSA, which is possible because a relatively short length parameter is used to identify plateaus, which is necessary due to the fluctuations along typical $G(z)$ traces. In turn, this leads to traces with slight variations from exponential decay often fitting the criteria for a continuous plateau. We generally find this is unavoidable regardless of the parameters used. To demonstrate this, we passed the continuous classified traces through a *k*-means clustering algorithm, which should be able to distinguish traces with small deviations from pure exponential behaviour from true molecular plateaus (Section S10 of the SI for further details).

The updated tunnelling/continuous-plateau/broken-plateau histograms after performing this second step are shown in panels h-j in Fig. 4. Now, the visual agreement between all groups is very good. A slight discrepancy remains, however, in the number of traces in the broken/continuous groups. Specifically, the CNN classifies about 30% more traces as continuous plateaus than the PSA, and about 30% fewer as broken plateaus. We do not expect perfect agreement, particularly as there is no concrete dividing line between continuous and broken plateaus. For example, the conductance may drop close to the noise threshold without the molecule actually detaching. The PSA may consider such traces as broken as the minimum conductance threshold must be above the instrument noise level. Our CNN is less sensitive to this criteria. Part of the difference may also be ascribed to the greater sensitivity of the PSA towards very small "breaks" in plateaus, which would correspond to a tiny change in the contrast of a single pixel in the images generated in the CNN method.

Overall, the CNN and PSA perform similarly, showing that both are capable of confidently identifying different $G(z)$ trace types. The CNN, however, outperforms the PSA in correctly identifying the tunnelling traces, whereas the PSA required the use of a second clustering step to clean up the traces misclassified as continuous plateaus. This is a clear advantage of the CNN model. In the future, the CNN model could be further trained to detect other generic shapes (such as plateaus with either a positive or negative slope) and we envisage a library of shapes could be constructed over time which could be used to screen datasets. This is something ideally suited to a neural network because, as we mentioned previously, UML algorithms struggle with relatively small subpopulations. Our current CNN may be limited in terms of the resolution of detail it can identify, but this may be minimised through the use of further training and/or a finer pixel array in the 2D map. Further, we imagine that once a library of shapes has been built up, this may be used as a way of searching for unusual/exotic behaviours *via* a process of elimination (similar to what could also be done with $G(V)$ curves). We also believe that combining SML models with UML algorithms may unlock new potential for discovering as yet unknown behaviours. A symbiotic approach, combining the best of both approaches, could be used to identify known shapes, and then focus in on much smaller subpopulations that are simultaneously unknown and rare. This is something which

neither technique could do independently, and which may unlock new insights from large datasets that would otherwise be beyond reach.

Conclusions

In summary, we showcase the ability of a convolutional neural network, initially trained to identify and discern elements in images, to classify both molecular conductance-voltage ($G(V)$) and conductance-distance ($G(z)$) traces in a highly efficient and accurate manner with low computational cost. This is the first time that a universal molecular conductance classification approach has been developed and successfully demonstrated. Thanks to the use of transfer learning, we have been able to reduce the training time to negligible levels and, moreover, we could reduce the number of training samples to a very low number whilst still achieving excellent classification performance. Our strategy involves recognising the core shapes of traces whilst ignoring differences in scale and variations due to noise. Our approach also allows a confidence level to be assigned to the classified traces, which can be used to improve accuracy further and yield information about traces with partial characteristics. We used our method to classify $G(V)$ curves from a large dataset recorded with a porphyrin dimer molecular wire. The results show that by applying a bias voltage which opens an energy window corresponding to a significant fraction of the molecular HOMO–LUMO gap, about 70% of curves correspond to neutral junctions, whilst at least 14% have a shape associated with charged junctions. The ability to quantify this behaviour will allow future analysis of redox process in molecular junction with unprecedented detail.

The large difference in shapes we have been able to classify shows that this model is well suited to analysing multi-component datasets. Further still, our model is able to distinguish subtly different shapes, recognisable by eye, but difficult to separate purely numerically. Though focussed on molecular conductance traces, our approach should be transferable to other research areas where the mapping of traces into 2D images is possible, exploiting the rich technology developed for the treatment of images.

Author contributions

EL and LZ conceived the project. CRP developed the methodology and wrote the machine learning code. CRP, MTG and EL performed analysis. EL and CRP wrote the manuscript, with contributions from all authors.

Conflicts of interest

There are no conflicts to declare.

Data availability

The raw data for the conductance *vs.* voltage curves and conductance *vs.* displacement traces obtained *via* the STM break junction technique, as well as a pseudocode for trace spiralization, have been archived on the Zenodo repository: **https://doi.org/10.5281/zenodo.15332767**. Other codes can also be found on Zenodo: **https://doi.org/10.5281/zenodo.17048167**.

Supplementary information: including examples of training data, details of the Deep Learning approach and comparison with other classification models. See DOI: **https://doi.org/10.1039/d5dd00207a**.

Acknowledgements

This work received financial support from grant PID2021-127964NB-C21 funded by MICIU/AEI/10.13039/5011000-11033. C. R.-P. acknowledges the FPI studentship associated with the CEX2018-000805-M-20-1 grant. M. T. G. acknowledges the grant DECOSMOL (EIGConcertJapan, PCI2023-143389). E. L acknowledges the grant CNS2023-145464 funded by MICIU/AEI/10.13039/501100011033 and by EU NextGenerationEU/PRTR. L. A. Z thanks the Spanish MICIU/AEI/10.13039/501100011033 for the grants PID2021-125604NB-I00, CNS2024-154593 and the "María de Maeztu" Programme for Units of Excellence in R&D (CEX2023-001316-M). IMDEA Nanociencia acknowledges support from the 'Severo Ochoa' Programme for Center of Excellence in R&D (CEX2020-001039-S). We thank Prof. H. L. Anderson (Oxford University) for providing the compounds used in this study.

Notes and references

1 T. Albrecht, G. Slabaugh, E. Alonso and S. M. R. Al-Arif, *Nanotechnology*, 2017, **28**, 423001.
2 Z. Lyu, Y. Wang, A. Sciazko, H. Li, Y. Komatsu, Z. Sun, K. Sun, N. Shikazono and M. Han, *J. Energy Chem.*, 2023, **87**, 32–41.
3 Z. Yao, Y. Lum, A. Johnston, L. M. Mejia-Mendoza, X. Zhou, Y. Wen, A. Aspuru-Guzik, E. H. Sargent and Z. W. Seh, *Nat. Rev. Mater.*, 2023, **8**, 202–215.
4 Y. Komoto, T. Ohshiro, Y. Notsu and M. Taniguchi, *RSC Adv.*, 2024, **14**, 31740–31744.
5 B. G. del Rio, B. Phan and R. Ramprasad, *npj Comput. Mater.*, 2023, **9**, 1–9.
6 A. Thelen, X. Huan, N. Paulson, S. Onori, Z. Hu and C. Hu, *npj Mater. Sustain.*, 2024, **2**, 14.
7 M. A. Hannan, D. N. How, M. H. Lipu, M. Mansor, P. J. Ker, Z. Dong, K. Sahari, S. K. Tiong, K. M. Muttaqi, T. I. Mahlia, *et al.*, *Sci. Rep.*, 2021, **11**, 19541.
8 J.-R. Deng, M. T. González, H. Zhu, H. L. Anderson and E. Leary, *J. Am. Chem. Soc.*, 2024, **146**, 3651–3659.
9 M. Kamenetska, S. Y. Quek, A. C. Whalley, M. L. Steigerwald, H. J. Choi, S. G. Louie, C. Nuckolls, M. S. Hybertsen, J. B. Neaton and L. Venkataraman, *J. Am. Chem. Soc.*, 2010, **132**, 6817–6821.
10 S. Guo, J. Hihath, I. Díez-Pérez and N. Tao, *J. Am. Chem. Soc.*, 2011, **133**, 19189–19197.
11 M. T. González, E. Leary, R. García, P. Verma, M. Á. Herranz, G. Rubio-Bollinger, N. Martín and N. Agraït, *J. Phys. Chem. C*, 2011, **115**, 17973–17978.

12 M. T. González, A. Díaz, E. Leary, R. García, M. Á. Herranz, G. Rubio-Bollinger, N. Martín and N. Agraït, *J. Am. Chem. Soc.*, 2013, **135**, 5420–5426.
13 D. Cabosart, M. El Abbassi, D. Stefani, R. Frisenda, M. Calame, H. S. J. van der Zant and M. L. Perrin, *Appl. Phys. Lett.*, 2019, **114**, 143102.
14 L. A. Zotti, B. Bednarz, J. Hurtado-Gallego, D. Cabosart, G. Rubio-Bollinger, N. Agraït and H. S. J. van der Zant, *Biomolecules*, 2019, **9**, 580.
15 J. M. Hamill, X. T. Zhao, G. Mészáros, M. R. Bryce and M. Arenz, *Phys. Rev. Lett.*, 2018, **120**, 016601.
16 M. Lemmer, M. S. Inkpen, K. Kornysheva, N. J. Long and T. Albrecht, *Nat. Commun.*, 2016, **7**, 12922.
17 J. Hurtado-Gallego, S. Sangtarash, R. Davidson, L. RincónGarcía, A. Daaoub, G. Rubio-Bollinger, C. J. Lambert, V. S. Oganesyan, M. R. Bryce, N. Agraït and H. Sadeghi, *Nano Lett.*, 2022, **22**, 948–953.
18 M. E. Abbassi, P. Zwick, A. Rates, D. Stefani, A. Prescimone, M. Mayor, H. S. J. van der Zant and D. Dulíc, *Chem. Sci.*, 2019, **10**, 8299–8305.
19 K. Zhou and S. Yang, *Pattern Anal. Appl.*, 2020, **23**, 455–466.
20 G. Pellicer and C. Sabater, *Processes*, 2025, **13**, 2061.
21 W. Bro-Jørgensen, J. M. Hamill, G. Mezei, B. Lawson, U. Rashid, A. Halbritter, M. Kamenetska, V. Kaliginedi and G. C. Solomon, *ACS Nanosci. Au*, 2024, **4**, 250–262.
22 F. van Veen, L. Ornago, H. S. van der Zant and M. El Abbassi, *J. Mater. Chem. C*, 2023, **11**, 15564–15570.
23 L. A. Zotti, T. Kirchner, J.-C. Cuevas, F. Pauly, T. Huhn, E. Scheer and A. Erbe, *Small*, 2010, **6**, 1529–1535.
24 P. Darancet, J. R. Widawsky, H. J. Choi, L. Venkataraman and J. B. Neaton, *Nano Lett.*, 2012, **12**, 6250–6254.
25 S. H. Choi, B. Kim and C. D. Frisbie, *Science*, 2008, **320**, 1482–1486.
26 A. Vilan, D. Aswal and D. Cahen, *Chem. Rev.*, 2017, **117**, 4248–4286.
27 J. C. Cuevas and E. Scheer, *Molecular Electronics: An Introduction To Theory And Experiment*, WSPC, New Jersey, 2nd edn, 2017.
28 M. Paulsson and S. Datta, *Phys. Rev. B:Condens. Matter Mater. Phys.*, 2003, **67**, 241403.
29 G. D. Scott and D. Natelson, *ACS Nano*, 2010, **4**, 3560–3579.
30 E. Leary, G. Kastlunger, B. Limburg, L. Rincón-García, J. Hurtado-Gallego, M. T. González, G. R. Bollinger, N. Agraït, S. J. Higgins, H. L. Anderson, R. Stadler and R. J. Nichols, *Nanoscale Horiz.*, 2021, **6**, 49–58.
31 F. Schwarz, G. Kastlunger, F. Lissel, C. Egler-Lucas, S. N. Semenov, K. Venkatesan, H. Berke, R. Stadler and E. Lörtscher, *Nat. Nanotechnol.*, 2016, **11**, 170–176.
32 W. Bro-Jørgensen, J. M. Hamill, R. Bro and G. C. Solomon, *Chem. Soc. Rev.*, 2022, **51**, 6875–6892.
33 K. P. Lauritzen, A. Magyarkuti, Z. Balogh, A. Halbritter and G. C. Solomon, *J. Chem. Phys.*, 2018, **148**, 084111.
34 J. Carracedo-Cosme, P. Hapala and R. Pérez, *Mach. Learn.: Sci. Technol.*, 2024, **5**, 025025.
35 D. S. Maitra, U. Bhattacharya and S. K. Parui, *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 1021–1025.
36 O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, *Int. J. Comput. Vis.*, 2015, **115**, 211–252.
37 D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2017.
38 L. Ming, I. Zabala-Gutierrez, P. Rodríguez-Sevilla, J. R. Retama, D. Jaque, R. Marin and E. Ximendes, *Adv. Mater.*, 2023, **35**, 2306606.
39 O. Ronneberger, P. Fischer and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.
40 J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. K. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, S. Zhang, M. Suo, P. Tillet, X. Zhao, E. Wang, K. Zhou, R. Zou, X. Wang, A. Mathews, W. Wen, G. Chanan, P. Wu and S. Chintala, *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, vol. 2, La Jolla CA USA, 2024, pp. 929–947.
41 Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell and S. Xie, *A ConvNet for the 2020s*, 2022.
42 E. Leary, B. Limburg, A. Alanazy, S. Sangtarash, I. Grace, K. Swada, L. J. Esdaile, M. Noori, M. T. González, G. RubioBollinger, H. Sadeghi, A. Hodgson, N. Agraït, S. J. Higgins, C. J. Lambert, H. L. Anderson and R. J. Nichols, *J. Am. Chem. Soc.*, 2018, **140**, 12877–12883.