ARTICLE

Using weakly supervised deep learning to classify and segment single-molecule break-junction conductance traces

Dongying Lin,[a] Zhihao Zhao,[b,c] Haoyang Pan,[a] Shi Li,[a] Yongfeng Wang,[a] Dong Wang,[b,c]

Stefano Sanvito,[d] Shimin Hou *[a]

[a] D. Lin, H. Pan, S. Li, Dr. Y. Wang, Prof. S. Hou
Department of Electronics, Peking University
Center for Nanoscale Science and Technology
Key Laboratory for the Physics and Chemistry of Nanodevices
Beijing 100871 (China)
E-mail: smhou@pku.edu.cn

[b] Z. Zhao, Dr. D. Wang
Chinese Academy of Sciences
CAS Key Laboratory of Molecular Nanostructure and Nanotechnology
CAS Research/Education Center for Excellence in Molecular Sciences
Beijing National Laboratory for Molecular Science (BNLMS)
Beijing 100190 (China)

[c] Z. Zhao, Dr. D. Wang
University of Chinese Academy of Sciences
Beijing 100049 (China)

[d] Prof. S. Sanvito
School of Physics, Trinity College
AMBER and CRANN Institute
Dublin 2, Ireland

Supporting information for this article is given via a link at the end of the document.

Abstract: In order to design molecular electronic devices with high performance and stability, it is crucial to understand their structure-to-property relationships. Single-molecule break junction measurements yield a large number of conductance-distance traces, which are inherently highly stochastic. Here we propose a weakly supervised deep learning algorithm to classify and segment these conductance traces, a method that is mainly based on transfer learning with the pretrain-finetune technique. By exploiting the powerful feature extraction capabilities of the pretrained VGG-16 network, our convolutional neural network model not only achieves high accuracy in the classification of the conductance traces, but also segments precisely the conductance plateau from an entire trace with very few manually labeled traces. Thus, we can produce a more reliable estimation of the junction conductance and quantify the junction stability. These findings show that our model has achieved a better accuracy-to-manpower efficiency balance, opening up the possibility of using weakly supervised deep learning approaches in the studies of single-molecule junctions.

Introduction

Scanning tunneling microscope-based break junction (STM-BJ) [1-5] and mechanically controlled-break junction (MCBJ) [5-9] are two among the most popular techniques employed to fabricate single-molecule junctions and to measure their conductance. Both of them are statistical methods and create a large quantity of conductance-distance traces. Due to the stochastic nature of the breaking process, the variable possible binding configurations between the molecule and the electrodes, molecular conformations, noise and environmental fluctuations, these conductance-distance traces exhibit diverse features. When there is no molecule bound in the junction, the conductance decays exponentially with distance, as expected from direct tunneling between the two electrodes. In contrast, when one or a few molecules are found in the junction, a conductance plateau always appears in the distance trace, indicating the formation of a molecular junction. However, the variation in the conductance values of the plateaus associated to the same molecule can be as large as one order of magnitude. Furthermore, such plateaus may break off and re-form within the same trace due to the fracture and reconnection of the molecular junctions. Therefore, the conductance value, together with the shape and length of these plateaus, are vital factors defining the electrical characteristics and the device stability of molecular junctions. It is highly desirable to develop efficient and effective approaches to classify the conductance-distance traces and extract meaningful features of the conductance plateaus.

Recently machine learning has been introduced into molecular electronics,[10-27] as a powerful tool to analyze break junction data. Supervised and unsupervised learning are the two main classes of machine learning algorithms, their main difference being whether or not a manually labeled training set is needed. Due to limited labor costs and the potential advantage of avoiding predefined bias in solving specific problems, many previous studies prefer unsupervised learning methods,[10-22] such as multi-parameter vector-based classification (MPVC),[10] deep auto-encoder K-means (DAK),[11] K-means++,[12] principal component analysis (PCA),[13] Alexnet-enhanced autoencoder[14] and spectral clustering.[19] In general, the conductance traces collected for a certain molecule or different molecules are grouped by their mutual similarities to uncover the underlying features. However, most of these unsupervised learning methods focus on clustering and/or dimensionality reduction tasks. In contrast, when we face a definite purpose, such as sorting the traces according to some specific localized features, unsupervised

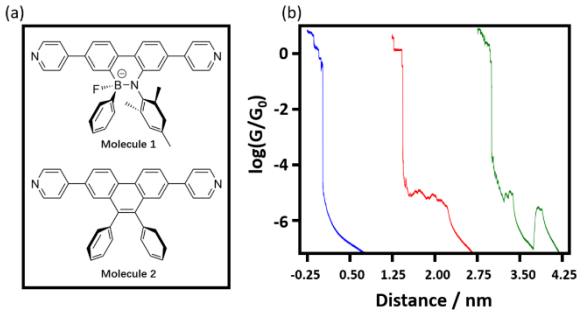


Figure 1. (a) chemical structures of the two molecules we investigate in this work. (b) typical STM-BJ traces collected for molecule 1, offset by 1.5 nm for clarity. Left, blue: conductance trace when there is no molecular junction formation; middle, red: conductance trace showing a distinct and smooth conductance plateau; right, green: conductance trace with one valley appearing in the middle of the plateau.

learning algorithms may sometimes provide unsatisfactory results.

At the same time, supervised learning has also been applied to the classification of conductance traces.[23,24] For example, Lauritzen et al. trained a recurrent neutral network for classifying experimental conductance curves of gold break junctions.[23] Moreover, a convolutional neural networks (CNN)-based method has been demonstrated to achieve a much higher accuracy in the identification of molecular junctions and a striking ability to sort conductance traces without relying on average conductance information.[24] Unfortunately, an obvious weakness of traditional supervised learning methods is that they need a large amount of manually labeled training data, leading to excessive manpower consumption. So the question we are asking is: can we implement supervised deep-learning methods with a few labeled data? Weakly supervised learning is a promising approach to handling this problem. Finetuning a pretrained neural network such as VGG-16[28] has been proved to be an effective strategy for image processing tasks, because the first several layers of neural networks always extract general features that can be used in other datasets.[29] We hope that this strategy may also work well in the classification of single-molecule conductance traces. Furthermore, we here try to extend the weakly supervised learning method to segment the conductance plateau from an entire trace, so to make a more accurate analysis of the junction conductance and the plateau lengths. In previous studies this is often realized through a threshold,[4] but its usefulness and accuracy are very limited due to large variations in the shape and length of the conductance plateaus.

Here we develop a weakly supervised learning algorithm that is based on a transfer learning methodology combined with convolutional neural network. In what follows, we describe our computation methodology, including our data pretreatment technique, and then present two case studies using our own STM-BJ data. These illustrate the applicability and advantages of a weakly supervised deep learning strategy. In the first case study, we show that our model can properly classify the conductance traces of the same molecule and segment the conductance plateau from an entire trace. As a result, the junction conductance can be extracted in a more reliable way and the junction stability can be quantified. In the second case study, we use a known data mixture to demonstrate that our model can sort conductance traces from different molecules with overlapping conductance distributions.

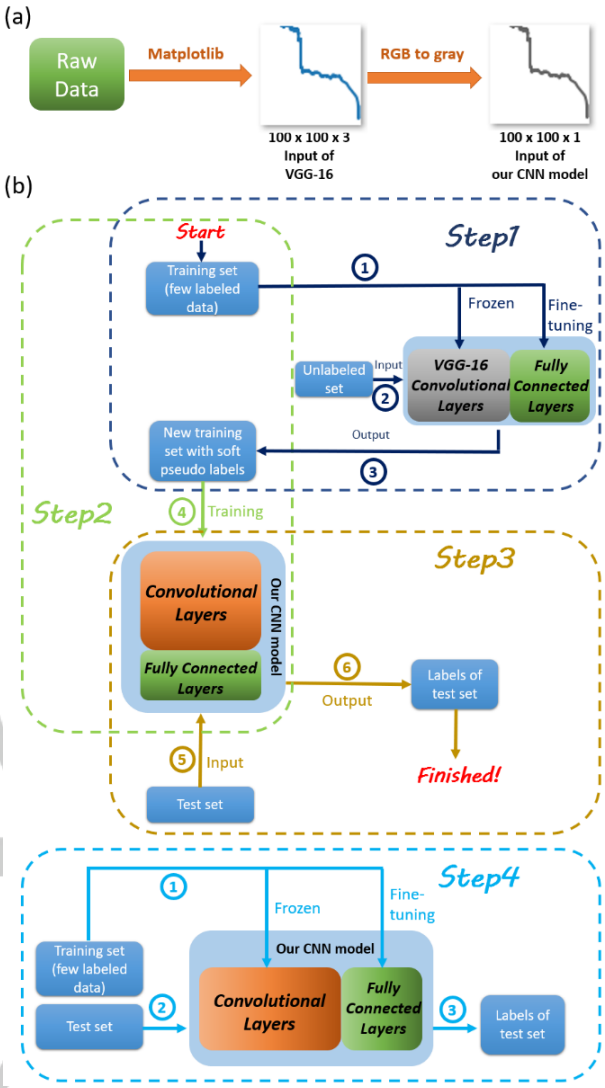


Figure 2. (a) General workflow used for the data pretreatment, where the matplotlib library transforms the raw traces into (100,100,3) RGB-images. These are then input in the VGG-16 network. For our CNN model, the traces are converted into (100,100,1) gray-scale images. The first two values encode the number of pixels of the images and the last one represents the number of color channels. In (b) we show the full algorithm flow chart.

Experimental Section

Data collection

The chemical structures of the two molecules investigated are shown in Figure 1(a). Molecules 1 and 2 have similar backbones, and their two ends are modified with pyridyl groups to anchor to the Au atoms of the substrate and the STM tip. We took measurements for the two molecules under the same experimental conditions. A solution (10 μl) containing 0.5 mM target molecules in the mixture of tetrahydrofuran (THF) and mesitylene (TMB, vTHF/vTMB=1:4) was dropped on the gold substrate to form the molecular junctions with the gold STM tip. With the STM-BJ technique, we obtain 1082 and 627 conductance-distance traces for molecules 1 and 2, respectively. Since the experimental details and the physical interpretation of the experimental results have been reported in reference [4], here we mainly focus on the data analysis aspects.

Data pretreatment

Data pretreatment is an important step in our method, whose entire workflow is shown in Figure 2(a). Each conductance trace is converted into a two-dimensional (2D) image as required by the VGG-16 network. This has many advantages including that all the images have the same size, independent from the number of data points in each conductance trace and that the classification accuracy is also much improved compared to the direct use of the raw data.[12,14,18] One of the key parameters used to construct such images is the number of pixels, which is selected by considering the accuracy, the computational cost and the suitability for the specific task. In our datasets, each conductance-distance trace contains more than 3000 data points. In order to fit these in the VGG-16 input shape dimension, we first plot the traces as (224, 224, 3) RGB images, the standard input size of VGG-16, but such representation is heavily time-consuming and is also inconvenient for the segmentation task. Then we plot them as (100,100,3) images [12] and find they can provide satisfied results for both the classification and segmentation tasks. Note that the fully connected (FC) layers of the original VGG-16 network are dropped so that it does not require the original input shape (3 color channels are still required). As for the input of our CNN model, we plot the traces as (100,100,1) gray-scale images, because the color information is redundant and such image color conversion is also very convenient.

Algorithm

Our STM-BJ conductance-distance traces have two specific characteristics. The first one is simplicity. We find that, after data pretreatment, a shallow neural network is enough for the classification and segmentation tasks, achieving a high accuracy. The second characteristic is similarity. Usually experiments for different molecules are conducted under similar experimental conditions, so that the measured traces have some similarities and it is not necessary to retrain the network completely.

We find that finetuning the VGG-16 network can give us high accuracy. However, due to the above-mentioned specific characteristics, the direct use of deep pretrained networks, like VGG-16, will extract unnecessary features such as color, and will require massive computing resources, thus leading to low efficiency. Therefore, we propose the algorithm shown in Figure 2(b): steps 1-3 are for a new experiment and the last step (blue dashed box) is for a similar task on different molecules. In our algorithm, the pretrained VGG-16 network works as a teacher model and helps us to train our CNN model (a student model) with few labeled data. The original VGG-16 network is designed for 1000-classification so that its output layer contains 1000 neurons. Therefore, the FC layers must be redesigned.

For the classification task, we use the 13 convolutional layers of VGG-16 with the weight of Imagenet and add 3 redesigned FC layers (3 neurons in the output layer) to its end. The complete model of this simplified VGG-16 network is shown in Figure S1 of the supplementary information (SI). In contrast, our CNN model is composed of 4 convolutional layers and 2 FC layers, and the activation functions of the last layer and the intermediate layers are, respectively, Soft-Max and ReLU. The optimizers and the loss functions in both the simplified VGG-16 network and our CNN model are Adam [30] and Categorical Cross Entropy. The complete model of our CNN is shown in Figure S2. Clearly, this is much simpler and shallower than VGG-16 (number of parameters: 212k vs 15,313k, see Figures S1 and S2) so that it has a much higher efficiency.

When looking at the segmentation task, the structure of the simplified VGG-16 network and our CNN model are the same as those used for classification, except for the last FC layers (now they both contain only 1 neuron). Moreover, the activation function of the last layer and the loss function are changed to Sigmoid and Mean Square Error (MSE), respectively.

Going into more detail, in step 1, we finetune the FC layers of the simplified VGG-16 network with a few labeled traces and use it to judge a large quantity of unlabeled traces. These are then added together in forming a new training set (with soft pseudo labels given by the simplified VGG-16 network). Hinton et al. suggested that the soft labels given by the teacher model always have a higher entropy and carry more information, such as the similarity of the different classes.[31] Thus, soft labels improve the performance of the student model. In step 2, we train our CNN with the so-constructed new training set, so that this can learn the "knowledge" reached by VGG-16. In step 3, we use our CNN model to label the traces contained in the test set. Step 4 is finally used either to classify the traces or to segment the conductance plateaus, of a new molecule dataset measured with similar experimental settings. The convolutional layers of our CNN model can still work well. However, we will need to finetune the FC layers with a few labeled traces specific of the new molecule. This is a rather numerically cheap operation, which adds little to the global computational overheads.

All the deep learning models are built using the Keras Python library running on top of TensorFlow-2.0.0rc2, while other methods except for GAL and UMAP are from the Scikit-Learn-0.23.2 Python library. The GAL method is from a Matlab toolbox and the UMAP method is from umap Python library. Data pretreatment is implemented with the matplotlib and OpenCV libraries.

Results and Discussion

Classification of traces collected for the same molecules. We start our investigation on the implementation of steps 1-3 on the classification of the conductance traces collected for molecule 1. As shown in Figure 1(b), there are 3 kinds of typical conductance traces, so that this is a tri-classification task: traces without the formation of any molecular junctions (class 0), traces with a clear and smooth molecular conductance plateau (class 1) and traces with one or more valleys appearing within the molecular conductance plateau (class 2). The 1082 traces available are plotted as (100,100,3) RGB and (100,100,1) gray-scale images and are divided into 3 different sets: the training set (20 traces with labels, ~2%), the unlabeled set (762 traces without labels, ~70%) and the test set (300 traces without labels, ~28%). We finetune the simplified VGG-16 network with the training set and use it to classify all the 782 traces contained in the combined unlabeled and the training sets. This latter ensemble forms a new training set with pseudo soft labels. After our CNN model is trained on such new training set, its performance is evaluated against the test set, delivering an overall accuracy of 90.3%. Here the classification accuracy is defined as the proportion of correctly

Table 1. Comparison of the classification accuracies of different machine learning methods

	Directly trained CNN	Label Propagation	UMAP with GMM	t-SNE with GAL	DAK	PCA with K-means++	Spectral Clustering	Our method
Tri-classification	72.3%	65.3%	64.3%	62.0%	61.3%	58.0%	63.7%	90.3%
Dual-classification	91.3%	89.7%	80.0%	84.3%	80.0%	84.3%	82.0%	94.7%

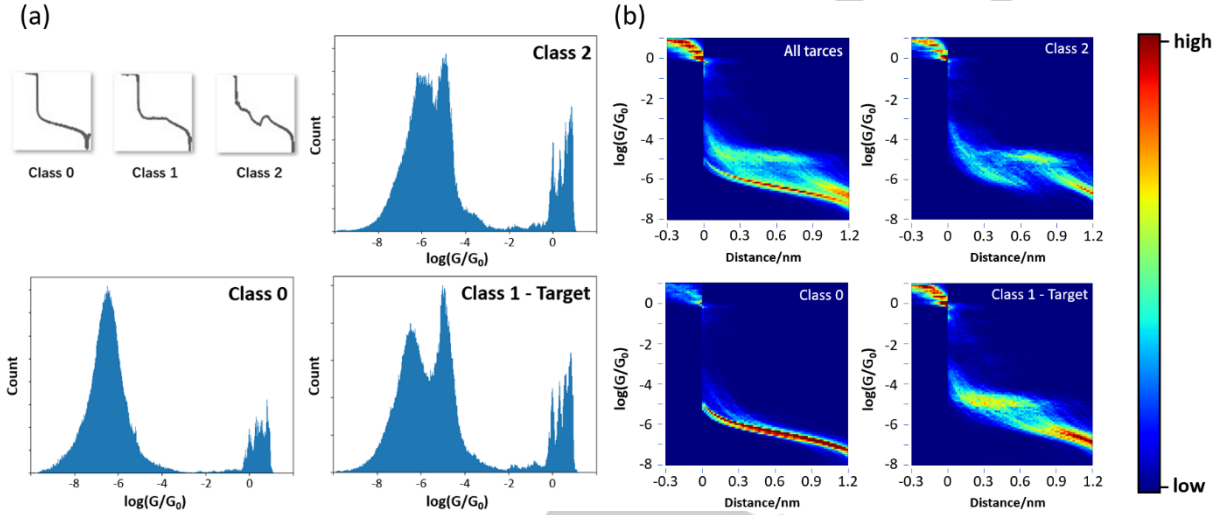


Figure 3. (a) typical inputs from classes 0, 1 and 2 collected with molecule 1 and the corresponding 1D histograms associated to these classes. (b) the 2D histograms of all the traces collected with molecule 1 (upper left) and traces in classes 0 (lower left), 1 (lower right) and 2 (upper right).

classified traces in the test set (by comparing the algorithm-generated class label of each conductance trace against its manual label). For comparison, we also apply alternative models to sort the conductance traces in the test set, including five unsupervised methods [t-distributed stochastic neighbor embedding (t-SNE) combined with graph average linkage (GAL),[32-34,18] uniform manifold approximation and projection (UMAP) combined with Gaussian mixed model (GMM),[35,36,18] DAK,[11] PCA combined with K-means++ [12,13] and spectral clustering[19]], one supervised method (directly trained CNN) and one semi-supervised method (label propagation[37]). For the former two unsupervised methods, the cosine (cos.) distance measure approach is used. As listed in Table 1, our method performs significantly better than all the others in our task. Since the most marked difference between the directly trained CNN model and our method is that the former is not combined with the simplified VGG-16 network and is trained only on the 20 labeled traces included in the training set, the much higher classification accuracy of our method with only 20 labeled traces is impressive and can be ascribed to the strong feature extraction capabilities of the convolutional layers of VGG-16. This is employed to generate a larger training set with pseudo soft labels. The VGG-16 network is trained with the Imagenet dataset and is very successful in image recognition.[28] Therefore, the incorporation of the VGG-16 network with the transfer learning technique greatly enhances the capability of our CNN model and thus significantly relaxes the requirement of obtaining large amounts of manually labeled traces for network training. In contrast, the directly trained CNN model overfits because of the insufficient number of labeled traces in the training set. When the training set for the directly trained CNN model is increased to 300 labeled traces, its classification accuracy on the same test set can be improved to 85.3%; however, this is still lower than the accuracy of our method.

In order to discover the reasons behind the poor performance of the unsupervised methods in the tri-classification task, we plot in Figure S3 the reduced feature vector distribution as well as the classification results obtained using PCA and K-means++ (see SI). As we can see, it is rather difficult to separate traces between class 1 and class 2 and those between class 1 and class 0. This means that unsupervised methods may not be suitable for such specific task, even when considering the best approach (t-SNE combined with GAL) proposed by El Abbassi et al.[18] However, in the much simpler dual-classification task where the traces are classified according to whether or not one or more molecular conductance plateaus appear, we find that all of these methods can provide an accuracy higher than 80%, while our method still provides the highest accuracy of 94.7%.

Figures 3a and 3b show one-dimensional (1D) conductance and 2D conductance-distance histograms constructed with all the unlabeled traces classified by our method. As we can see, there is a long distinct conductance plateau in the 2D histogram of class 1 and the boundary between the molecular conductance plateau and the background tail (the break-off from the end of a conductance plateau) is also very clear. Note that there is still a significant data density just below the main plateau feature, which is caused by some of the smooth plateaus being just significantly

Table 2. The number of traces in different classes and the junction stability

	Molecule 1	Molecule 2
Class 0	288	1
Class 1	441	473
Class 2	353	153
Connection	794	626
Stable-connection	297	461
Connection rate	73.4%	99.8%
Stable-connection rate	27.4%	73.5%

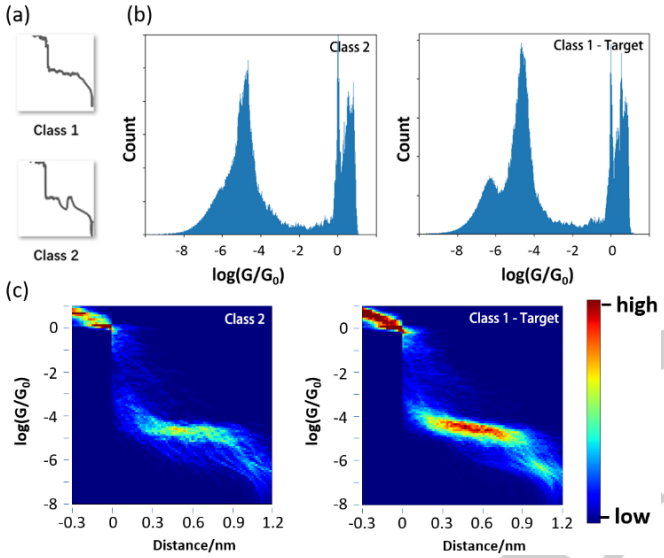


Figure 4. Typical inputs (a) and the corresponding 1D (b) and 2D (c) histograms of classes 1 and 2 collected for molecule 2.

shorter than others and producing plateau break-off features. Correspondingly, the conductance peak in the 1D histogram is very sharp and well separated from the background. In contrast,the conductance plateau (the flat section of a conductance trace) in the 2D histogram of class 2 is much shorter. In addition to the conductance plateau and the background tail, there is also an additional feature due to the fracture and reconnection of the molecular junctions. Finally, we do not see a peak and a plateau at the conductance value of molecule 1 in the 1D and 2D histograms of class 0. At the same time, only very faint characteristics of class 0 appears in the 2D histograms of classes 1 and 2, indicating that our method is highly accurate. When we construct the 2D histogram with all these conductance traces, the conductance plateau becomes very faint and the boundary between the conductance plateau and the background tail is also blurred by the contributions from traces in classes 0 and 2, signifying the importance of the tri-classification. It should be noted that only 20 labeled traces are included in the training set so that they must be selected with distinguished features of these three classes (see Figure S4). However, the classification results are not strongly dependent on specific traces. This is illustrated with other three different training sets, each of which is composed of 20 labeled traces, and the corresponding 2D histograms of classes 0, 1 and 2 are shown in Figures S7-S9. In contrast, the 1D and 2D histograms of the classification results obtained by using other methods demonstrate that the accuracies of these models are significantly lower (Figures S10-S16).

Next, we apply our CNN model to characterize the conductance data of molecule 2. Since our CNN model has been trained over molecule 1 at the steps 1-3, we freeze its convolutional layers and only finetune its FC layers with 20 labeled traces. It should be noted that there are very few traces in class 0 for molecule 2 (see Table 2) so that the training set is composed of 4 labeled traces in class 0 from molecule 1 and 16 labeled traces in classes 1 and 2 from molecule 2. Then we use the trained CNN model to process 611 unlabeled traces (the test set) and generate a class label as output, identifying the molecular junction type. The overall classification accuracy is 89.5%, only marginally lower than that of molecule 1. Here, we also classify the traces of molecule 2 into 3 classes but only show the 1D and 2D histograms of classes 1 and 2 in Figures 4b and 4c, because the number of the traces in class 0 is too small to construct appropriate histograms.

For class 1, both the conductance peak and the background peak in the 1D histogram, and the conductance plateau and the background tail in the 2D histogram are clearly distinguished from each other. In contrast, although a conductance plateau can be seen in the 2D histogram of class 2, the boundary between the conductance plateau and the background tail is unclear. Correspondingly, in the 1D histogram the background peak merges into the conductance peak and makes it heavily broadened at the left-hand side. The marked differences in both the 1D and 2D histograms between class 1 and class 2 demonstrate the effectiveness and accuracy of our method. Moreover, we can easily find that the traces of molecule 2 always show a longer conductance plateau than those of molecule 1, although these two molecules have identical linker groups and similar backbones. This indicates that molecule 2 has a higher junction stability, which will be quantitatively discussed in the next section.

Plateau segmentation. As shown in Figures 3 and 4, the conductance peaks of molecules 1 and 2 still overlap with the peak coming from the background tail. Therefore, it is highly desirable to separate the plateau from an entire conductance trace so that both the junction conductance and the plateau length can be determined accurately.[17] Certainly, this cannot be realized using the traditional threshold method, due to large variations in the plateau shape and length. Considering the powerful pattern recognition ability of the CNN, we implement steps 1-3 on the class-1 traces for molecule 1, with the task of performing plateau segmentation. Since it is easy to locate the starting point of a plateau by setting a threshold or by detecting a discontinuity in the trace (see Figure S17), only data points after the starting point of the plateau are plotted as a (100,100,3) RGB or (100,100,1) gray-scale image in order to simplify the task. Then, 20 traces with labels and 100 unlabeled traces are respectively used as the training set and the test set, while the remaining 321 traces form the unlabeled set. In this case the end position of a plateau is the only output of the model. Since the Sigmoid function outputs values in the (0,1) range, all data should be normalized. A typical result of the plateau segmentation is shown in Inset 1 of Figure 5(a). We then compare the plateau lengths of the 100

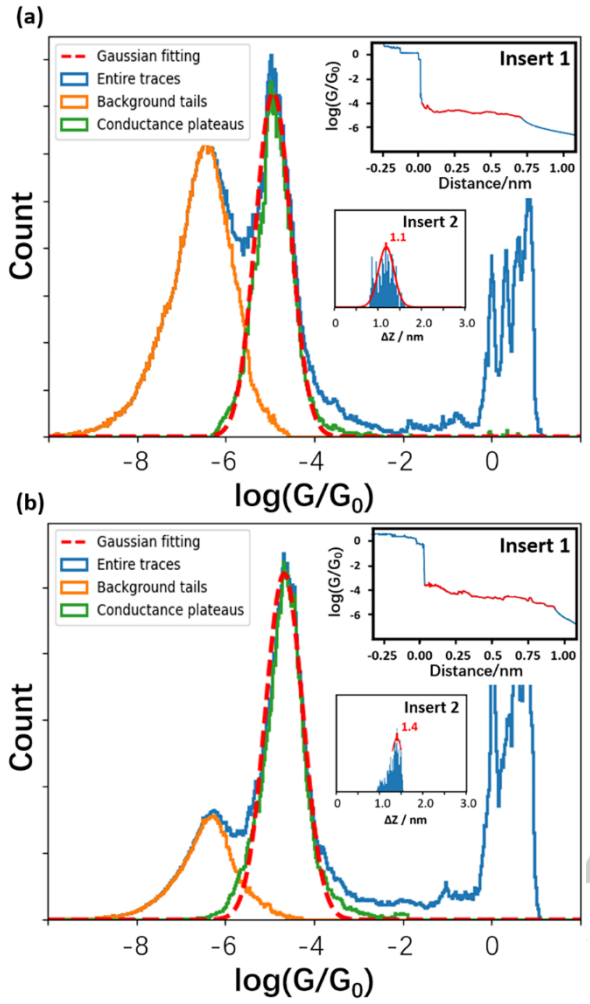


Figure 5. The 1D conductance histograms constructed with traces of class 1 of molecule 1 (a) and molecule 2 (b). Blue solid line: data points of the entire traces; green solid line: data points of the conductance plateaus; Orange solid line: data points of the background tails; Red dashed line: Gaussian fitting. Insert 1: typical result of the conductance plateau (labeled in RED) segmented with our CNN model. Insert 2: the 1D histogram of the lengths of the segmented conductance plateaus.

traces in the test set determined by the algorithm with those determined manually and find that the mean absolute error is 0.05 nm. The same task is also applied for the class-1 conductance traces of molecule 2, but only by performing step 4. Now 20 traces with labels and 453 traces without labels are respectively used as the training set and the test set, and the mean absolute error is calculated to be 0.04 nm. The 1D conductance histograms constructed with the data points from the extracted conductance plateaus and the background tails of the unlabeled traces of molecules 1 and 2 are shown in Figures 5a and 5b, respectively. Clearly, the peaks of the molecular conductance and the background tail are now well separated and there is also no significant reduction of the peak height compared to the raw data. This demonstrates that our segmentation method has extracted nearly all of the molecular signatures contained in the conductance traces. We find that the first 0.1 nm segment of the conductance plateaus is greatly affected by the shrink of the gold STM tip or the gold substrate when the single-atom point contact is broken, so that these data points are not included in the 1D conductance histogram.

After the tri-classification of the conductance traces and the plateau segmentation, the 1D histograms of these two molecules show conductance peaks having a simple shape, which can be fitted with an unrestricted single Gaussian. Furthermore, the most probable junction conductance [log (G/G_0)] of molecule 1 is determined to be -4.9±1.5, while that of molecule 2 is -4.7±1.2. Here $G_0=2e^2/h$ is the conductance quantum. Now, both the peak value and the confidence can be obtained from unrestricted Gaussian fitting. The peak value of molecule 1 is slightly larger (-5.0) than that obtained from fitting the raw data to a restricted Gaussian, while it is the same for molecule 2.[4] Moreover, there is another potential advantage of our method: if other junction configurations exist, we can easily find them in the 1D conductance histogram.

The numbers of the conductance traces in the three classes of molecules 1 and 2 are listed in Table 2. Now we can quantitatively estimate the connection rate and the stable connection rate of the molecular junctions. The connection rate, or junction formation probability, is defined as the ratio between the sum of the numbers of traces belonging to classes 1 and 2 and the total number of traces collected for one type of molecule. When the conductance plateau of one class-1 trace is long enough, this molecular junction can be considered as a stable connection. Since the lengths of these two molecules are both ~1.5 nm, the lower limit of the plateau length of a stable connection is chosen to be 1.0 nm. Thus, the stable connection rate is defined as the ratio of the number of stable connections to the total number of all traces. As we can see in Table 2, both the connection rate and the stable connection rate of molecule 2 are much higher than those of molecule 1. The reduction in junction formation probability and the stable connection rate of molecule 1 may be attributed to the increasing steric hindrance due to the F-anion coordinated to the B atom of the B-N bond. Therefore, compared to molecule 1, molecule 2 can easily bind to the gold electrodes and form stable molecular junctions.

Moving to the lengths of the conductance plateaus of the two molecules, our statistical results are shown in Insert 2 of Figures 5(a) and (b). It should be noted that the plateau lengths have been increased by 0.5 nm to take into account the snap-back distance (0.5±0.1 nm) caused by the abrupt Au-Au rupture.[4] The plateau-length distribution of molecule 1 shows a Gaussian-like profile. Indeed, Gaussian fitting is often used to analyze the lengths of molecular conductance plateaus in most of the previous studies.[3-5,11,13] In contrast, the plateau-length histogram of molecule 2 significantly deviates from a Gaussian distribution, showing a rapid decline at the right-hand side of the peak. Considering that the lengths of these two molecules are very close due to their similar backbones, this striking difference is remarkable and can be ascribed to the quite different stabilities of these two types of Au-molecule-Au junctions. The fracture of a molecular junction can be categorized into two types: 1) random fractures happening during the pulling process, 2) molecular junctions extending to the maximum length of the molecule and breaking off at the gold-pyridine interfaces. As listed in Table 2, junctions made of molecule 2 are much more stable than those formed with molecule 1. For molecule 1, due to the lower stability, most of the molecular junctions fracture during the pulling process and cannot reach the maximum length (see Figure 3b). Furthermore, the stochastic nature of these fractures leads to a Gaussian-like

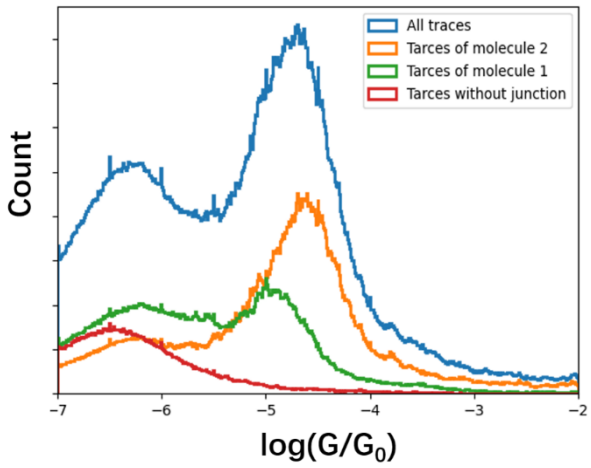


Figure 6. The 1D histogram of the conductance traces of the mixed data. blue: all traces collected for molecules 1 and 2; Orange: the traces judged to be molecule 2; Green: the traces judged to be molecule1; Red: the traces judged to be direct tunneling (without the formation of any molecular junctions)

distribution with a peak value of 1.1 nm. In contrast, junctions made of molecule 2 have a much higher stability and many of them can approach the maximum length (see Figure 4c), so that the length distribution of the conductance plateaus has a peak with a higher value of 1.4 nm and a rapid decay at the right side of the peak.

Classification of traces collected for different molecules. Now we demonstrate that our method can also work well in the classification of the conductance traces collected for different molecules, investigated previously in many studies.[11,13,14,17,21] Now, we merge the conductance traces collected for molecules 1 and 2 and train our CNN model to classify these traces with the steps 1-3. All the procedures are the same as those in the tri-classification task of molecule 1. There are still 3 classes of traces in the dataset, namely molecules 1, 2 and those without the formation of any molecular junctions. A total of 1709 traces are divided into 3 different sets: the training set (20 traces with labels, ~1%), the unlabeled set (980 traces without labels, ~57%) and the test set (709 traces without labels, ~42%). The overall accuracy of the model here is as high as 92.9% on the test set. The 1D histogram constructed with all of the unlabeled conductance traces is shown in Figure 6. In the raw data, there is only one broad peak corresponding to the molecular conductance, since the conductance values of these two types of molecular junctions are too close and heavily overlapping with each other. However, with the help of our network, these three classes of traces can be accurately separated. Moreover, the conductance peaks of molecules 1 (green) and 2 (orange) cannot be fitted with an unrestricted single Gaussian like the one shown in Figures 5a and 5b for class 1, highlighting the necessity of the tri-classification for conductance traces of the same type of molecule.

We also observe another interesting phenomenon in Figure 6. Compared to the traces corresponding to junctions not presenting a molecular bridge, the background tails of the traces of actual molecular junctions show a peak with a higher conductance value. One possible explanation for this observation is that the molecule may still bind to the STM tip or the gold substrate after the molecular junction breaks off, leading to the observed large conductance.

Conclusion

We have developed a weakly supervised deep learning method, mainly based on transfer learning from the pretrain-finetune technique. With the help of the VGG16 network, our CNN model can perform classification of the conductance traces collected with the STM-BJ technique and the segmentation of the conductance plateaus. These tasks are delivered with a very high accuracy, and require only very few manually labeled data. As a result, we can make a more reliable estimation of the junction conductance after removing the complex background signatures originating from tunneling and from junction rupture and re-formation (broken plateaus in the conductance trace). In addition we can extract information concerning the junction formation probability and stability, properties that are highly correlated to the distribution of the lengths of the conductance plateaus. Our findings demonstrate that weakly supervised deep learning methods can be used to efficiently analyze break-junction data and thus are helpful for the establishment of structure-property relationships in single-molecule electronic devices.

Acknowledgements

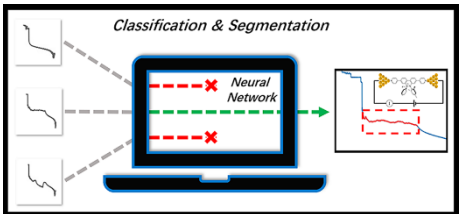
This work was supported by National Natural Science Foundation of China (Grant Nos. 21933002, 21803061 and 21725306), the National Key R&D Program of China (Grant Nos. 2016YFA0201901 and 2018YFA0306003) and High-performance Computing Platform of Peking University. SS thanks Science Foundation Ireland (AMBER Center grant 12/RC/2278_P2) for financial support. We thank Prof. Ge Li for comments on the deep learning algorithms.

Keywords: single-molecule junction; conductance-distance trace; weakly supervised deep learning; transfer learning; the pretrain-finetune technique

[1]. B. Xu, N. J. Tao, *Science* **2003**, 301, 1221-1223.
[2]. L. Venkataraman, J. E. Klare,C. Nuckolls, M. S. Hybertsen, M. L. Steigerwald, *Nature* **2006**, 442, 904–907.
[3]. V. Kaliginedi, P. Moreno-García, H. Valkenier, W. Hong, V. M. García-suarez, P. Buiter, J. L. H. Otten, J. C. Hummelen,´C. J. Lambert, T. Wandlowski, *J. Am. Chem. Soc.* **2012**, 134, 5262-5275.
[4]. Z. Zhao, L. Wang, S. Li, W. Zhang, G. He, D. Wang, S. Hou, L. Wan, *J. Am. Chem. Soc.* **2020**, 142, 8068-8073.
[5]. C. Huang, A. V. Rudnev, W. Hong, T. Wandlowski, *Chem. Soc. Rev.* **2015**, 44, 889-901.
[6]. M. A. Reed, C. Zhou, C. J. Muller, T. P. Burgin, J. M. Tour, *Science* **1997**, 278, 252-254.
[7]. P. Gehring, J. M. Thijssen, H. S. J. van der Zant, *Nat. Rev. Phys.* **2019**, 1, 381–396.
[8]. R. H. M. Smit, Y. Noat, C. Untiedt, N. D. Lang, M. C. van Hemert, J. M. van Ruitenbeek, *Nature* **2002**, 419, 906-909.
[9]. C. A. Martin, D. Ding, H. S. J. van der Zant, J. M. van Ruitenbeek, *New Journal of Physics* **2008**, 10, 065008.
[10]. M. Lemmer, M. S. Inkpen, K. Kornysheva, N. J. Long, T. Albrecht, *Nat. Commun.* **2016**, 7, 12922.
[11]. F. Huang, R. Li, G. Wang, J. Zheng, Y. Tang, J. Liu, Y. Yang, Y. Yao, J. Shi, W. Hong, *Phys. Chem. Chem. Phys.* **2020**, 22, 1674-1681.
[12]. D. Cabosart, M. El Abbassi, D. Stefani, R. Frisenda, M. Calame, H. S. J. van der Zant, M. L. Perrin, *Appl. Phys. Lett.* **2019**, 114, 143102.
[13]. J. M. Hamill, X. Zhao, G. Meszaros, M. R. Bryce, M. Arenz, *Phys. Rev. Lett.* **2018**, 120, 016601.

[14]. A. Vladyka, T. Albrecht, *Mach. Learn.: Sci. Technol.* **2020**, 1, 035013.
[15]. M. S. Inkpen, M. Lemmer, N. Fitzpatrick, D. C. Milan, R. J. Nichols, N. J. Long, T. Albrecht, *J. Am. Chem. Soc.* **2015**, 137, 9971-81.
[16]. A. Magyarkuti, N. Balogh, Z. Balogh, L. Venkataraman, A. Halbritter, *Nanoscale* **2020**, 12, 8355-8363.
[17]. N. D. Bamberger, J. A. Ivie, K. N. Parida, D. V. McGrath, O. L. A. Monti, *J. Phys. Chem. C* **2020**, 124, 18302-18315.
[18]. M. El Abbassi, J. Overbeck, O. Braun, M. Calame, M. L. Perrin, H. S. J. van der Zant, *Commun. Phys.* **2021**, 4, 50.
[19]. L. Lin, C. Tang, G. Dong, Z. Chen, Z. Pan, J. Liu, Y. Yang, J. Shi, R. Ji, W. Hong, *J. Phys. Chem. C* **2021**, 125, 3623–3630.
[20]. B. H. Wu, J. A. Ivie; T. K. Johnson;O. L. A. Monti, *J. Chem. Phys.* **2017**, 146, 092321
[21]. Q. Zhang, C. Liu, S. Tao, R. Yi, W. Su, C. Zhao, C. Zhao, Y. J. Dappe, R. J. Nichols, L. Yang, *Nanotechnology* **2018**, 29, 325701.
[22]. B. Liu, S. Murayama, Y. Komoto, M. Tsutsui, M. Taniguchi, *J. Phys. Chem. Lett.* **2020**, 11, 6567-6572.
[23]. K. P. Lauritzen, A. Magyarkuti, Z. Balogh, A. Halbritter, G. C. Solomon, *J. Chem. Phys.* **2018**, 148, 084111.
[24]. T. Fu, Y. Zang, Q. Zou, C. Nuckolls, L. Venkataraman, *Nano Lett.* **2020**, 20, 3320-3325.
[25]. T. Albrecht, G. Slabaugh, E. Alonso, S. Al-Arif, *Nanotechnology* **2017**, 28, 423001.
[26]. R. Korol, D. Segal, *J. Phys. Chem. B* **2019**, 123, 2801-2811.
[27]. A. Akihide, T.Makusu, W.Tasashi, B. Yoshinobu, K.Tomoji, *J. Phys. Chem. C* **2021**, 93, 215-227.
[28]. K. Simonyan, A. Zisserman, *In International Conference on Learning Representations* **2015**.
[29]. J. Yosinski, J. Clune, Y. Bengio, H. Lipson, *In Advances in Neural Information Processing Systems* **2014**, 27, 3320–3328.
[30]. D. P. Kingma, J. Ba, *In International Conference on Learning Representations* **2015**.
[31]. G. Hinton, O. Vinyals, J. Dean, *In NIPS* **2014**.
[32]. G. Hinton, S.T. Roweis, *In Advances in Neural Information Processing Systems, **2003**,* 857-964
[33]. L. Van Der Maaten, G. Hinton, *J. Mach. Learn. Res.* **2008**, 9, 2579-2605
[34]. W. Zhang, D. Zhao, X. Wang, *Pattern Recognit.* **2013**, 46, 3056-3065
[35]. L. McInnes, J. Healy, N. Saul, L. Groβberger, *J. Open Source Softw.* **2018**, 3, 861
[36]. C.K.I. Williams, C.E. Rasmussen, *IN: Advances in Neural Information Processing Systems 8*. **1996**, 514-520
[37]. X. Zhu, Z. Ghahramani, *Technical Report CMU-CALD-02-107, Carnegie Mellon University* **2002**

Entry for the Table of Contents



By classifying and segmenting the conductance-distance traces with a weakly supervised deep learning algorithm, we can produce a more reliable estimation of the junction conductance and quantify the junction stability.