Using Deep Learning to Identify Molecular Junction Characteristics

Tianren Fu[1], Yaping Zang[1], Qi Zou[1,2], Colin Nuckolls[1], Latha Venkataraman*[1,3]

[1] Department of Chemistry, Columbia University, New York, New York 10027, United States
[2] Shanghai Key Laboratory of Materials Protection and Advanced Materials in Electric Power, Shanghai University of Electric Power, Shanghai 200090, China
[3] Department of Applied Physics and Applied Mathematics, Columbia University, New York, New York 10027, United States

Abstract

The scanning tunneling microscope-based break junction (STM-BJ) is used widely to create and characterize single metal-molecule-metal junctions. In this technique, conductance is continuously recorded as a metal point-contact is broken in a solution of molecules. Conductance plateaus are seen when stable molecular junctions are formed. Typically, thousands of junctions are created and measured, yielding thousands of distinct conductance versus extension traces. However, such traces are rarely analyzed individually to recognize the types of junctions formed. Here, we present a deep learning-based method to identify molecular junctions and show that it performs better than several commonly used and recently reported techniques. We demonstrate molecular junction identification from mixed solution measurements with accuracies as high as 97%. We also apply this model to an *in situ* electric-field driven isomerization reaction of a [3]cumulene to follow the reaction over time. Furthermore, we demonstrate that our model can remain accurate even when a key parameter, the average junction conductance, is eliminated from the analysis, showing that our model goes beyond conventional analysis in existing methods.

Main Text

Break junction techniques, such as the scanning tunneling microscope-based break junction (STM-BJ)[1, 2] and mechanically controlled-break junction (MC-BJ),[3, 4] are robust and powerful methods to create and characterize well-defined single Au-molecule-Au junctions. In break junction experiments, the electronic properties of these junctions are typically recorded although in addition mechanical, thermoelectric and flicker noise characteristics can also be measured and analyzed.[5-9] Most frequently, conductance data from these measurements are analyzed by looking at averages through histograms. However, a single break-junction measurement with multiple possible junction types requires a junction-by-junction analysis. This is especially true in STM-BJ measurements where *in situ* chemical reactions involve different molecules participating or created during the course of the measurement in one experiment.[10-13] Recently, machine learning methods have been applied to STM-BJ data.[14-17] However, these methods still rely on averaging some aspects of the measurements, which results in a loss of information during the data preprocessing and analysis.

Deep learning is a powerful but more complicated machine learning technique which is capable of representing and analyzing multiple aspects of measured data. Recently, deep learning-based analysis have been applied to STM measurements[18] and nano-gap conductance data.[19] For break junction-related data, Lauritzen and coworkers study the rupture process of Au-Au contact using recurrent neural network[20] and Huang and coworkers develop a clustering method on conductance traces with deep auto-encoder[21] techniques. Among deep learning techniques, convolutional neural network (CNN) is a particularly powerful and popular method for image recognition.[22] Since STM-BJ data, which records conductance as a function of distance (or equivalently time), can be regarded as a 1D image, CNN can, in principle, be applied to such data. In this study, we develop a CNN-based model that can be applied to single-molecule conductance data collected using an STM-BJ setup and demonstrate its higher accuracy and robustness compared to non-deep learning models. Importantly, we show how this method can be used to characterize junctions where we remove a key parameter, its average conductance, highlighting the rich information available in conductance-time traces beyond what is analyzed using histograms. We note however that this performance gain has its cost. When compared with non-deep learning methods,[14-17] our method requires more time to train the model. Additionally, our method is a supervised method, i.e., we need to feed the model reference measurements to sort the data, while unsupervised methods such as clustering,[14, 15, 21] do not require reference measurements.

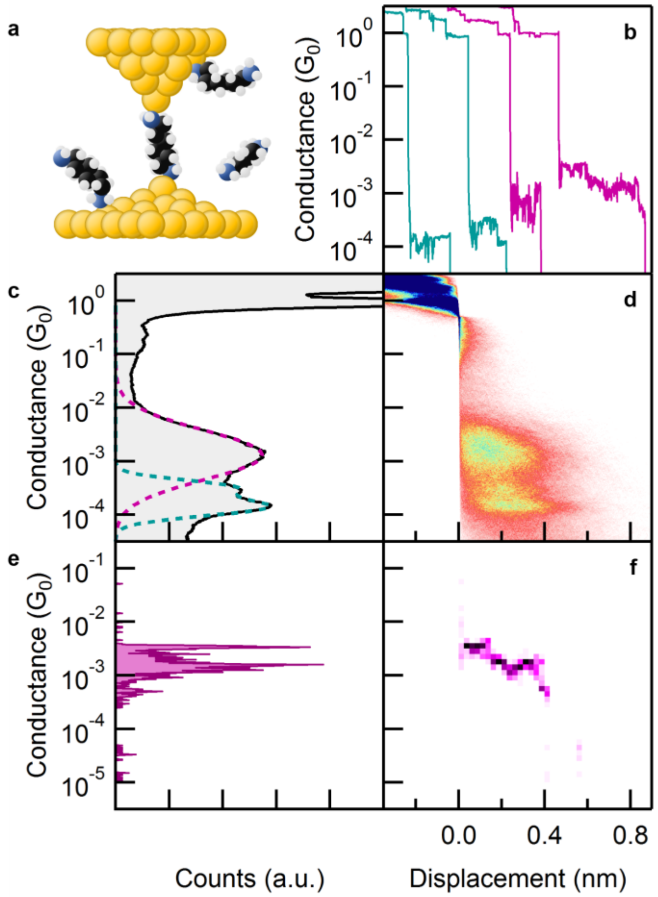


Figure 1. (a) Illustration of a molecular junction formed with STM-BJ. (b) Typical STM-BJ traces. (c) The 1D and (d) 2D histograms of a measurement of a mixed solution with1,6-diaminohexane and 4,4’-bis(methylthiol)biphenyl. (e) The 1D and (f) 2D histograms of the rightmost trace (single trace) shown in (b) showing only the molecular conductance region.

In a single break junction measurement, two gold electrodes start in contact and are gradually pulled apart in a molecular solution, forming molecular junctions as shown in Figure 1a. Conductance is recorded as a function of the electrode separation. Plateaus at or above 1 G_0 (G_0 =

$2e^2/h$, the quantum of conductance) correspond to atomic size gold contacts and plateaus below 1 G_0 are attributed to a molecule bridging the gap between the two electrodes. Figure 1b shows several example conductance-versus-displacement traces measured in the presences of a mixture of two molecules. Typically, conductance traces are analyzed by creating 1-dimensional (1D) conductance and 2-dimensional (2D) conductance-displacement histograms from all measured traces, as shown in Figure 1c and 1d. From these histograms we can obtain the average junction conductance and the average junction elongation length.

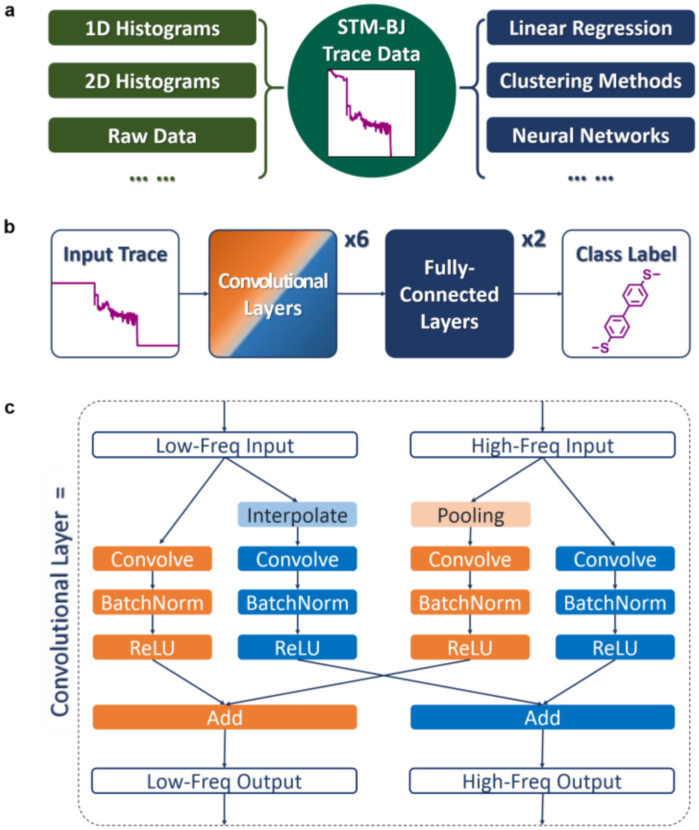


Figure 2. (a) Illustration of STM-BJ data analysis methods. On the left are the methods used for data preprocessing to generate an input from original trace. On the right are the models that can be applied to analyze STM-BJ data. (b) A simplified chart showing the flow of data in the CNN model used here. (c) The illustration of one convolutional layer shown in (b).

Single traces can also be converted to individual 1D and 2D histograms (see Figure 1e and 1f) and then analyzed using machine learning methods. For example, Hamillet *et al*[15] have used the principal component analysis (PCA) method on single-trace 1D histograms (denoted as PC_1/1DH), while Cabosart *et al*[14] have applied a KMeans++ clustering algorithm[23, 24] on single-trace 2D histograms (denoted as KMeans/2DH) to categorize STM-BJ data. However, both these methods lose information that is present in the raw conductance-versus-displacement traces. For example, focusing on the molecular conductance plateau (Figure 1b), we see that small fluctuations and oscillations are lost when these are converted into single-trace histograms (Figure 1e and 1f).

Here, we analyze the original STM-BJ conductance trace, i.e. a 1D array of conductance values. In Figure 2a, we summarize some common data analysis methods and show how traces are processed on the left and the classification algorithms used on the right. Among these, keeping all the raw data are likely the best, and this is easiest using a CNN-based analysis method. We therefore then design a CNN-based model as illustrated in Figure 2b. In this model, a clipped STM-BJ trace that excludes the gold point contact (data points with a conductance greater than 0.1 G_0) and noise floor (lower than 10^{-5} G_0) is taken as input. This focuses the analysis on the molecular conductance region. After processing the data with 6 convolutional layers and 2 fully-connected layers, the model generates a class label as output, identifying the molecular junction type. The fully-connected layer here has the same structure as a layer in a regular multilayer perceptron, where in each fully-connected layer, the input data matrix is multiplied by a weight matrix and offset by a bias matrix. The result from each of these multiplications undergoes a non-linear activation to break the linearity; here we use a rectified linear unit (ReLU), where the negative values are simply flattened to zero.[25] Dropout is then applied to provide extra robustness by randomly discarding outputs of some neurons during training; this prevents the network from relying on very few neurons.[26] The convolutional layers used in this model are of the octave convolution (OctConv) style,[27] as illustrated in Figure 2c. Compared to vanilla convolution, OctConv recognizes data shapes better and remains invariant under scaling (by introducing the low-frequency section in Figure 2c). Each of the four columns in Figure 2c represent a vanilla convolutional layer, with a 1D convolution operation, batch normalization (BatchNorm)[28] and ReLU. An OctConv layer is broken into four columns of convolutions providing the cross-processing within and between the high-frequency branch and low-frequency branch to keep information shared between the two spatial scales. Nearest neighbor interpolation and average pooling are used to double or half the size of data to match the different data sizes. The structure of OctConv layers is described in detail in the Section 2 of Supporting Information (SI).

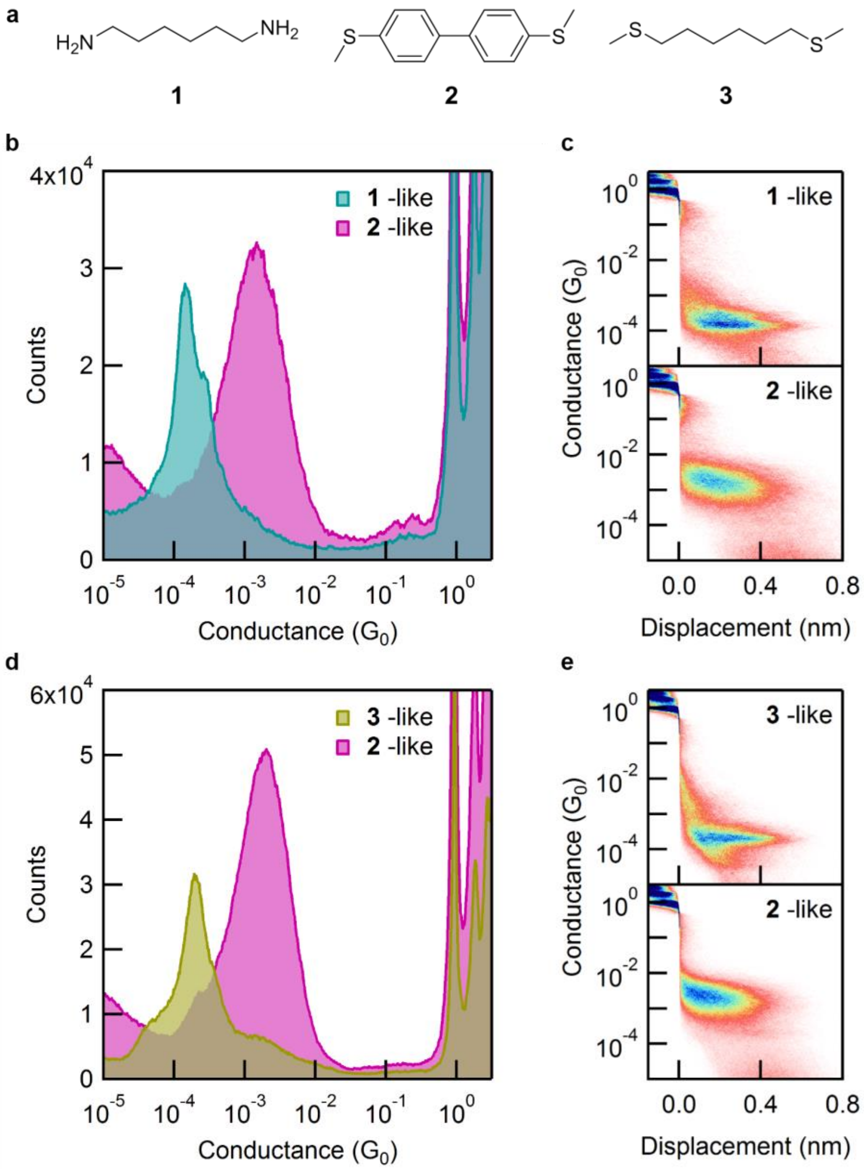


Figure 3. (a) Chemical structures **1**, **2**, **3**. (b) The 1D and (c) 2D histograms of the traces judged to be **1**-like (3406 traces) or **2**-like (4876 traces) by the CNN model from mixed solution measurements. The histograms of all traces are shown in Figure 1 (b) and (c), and histograms of measurements on pure solutions are shown in SI Figure S1. (d) The 1D and (e) 2D histograms of the traces judged to be **2**-like (7678 traces) or **3**-like (4098 traces) by the CNN model from mixed solution measurements.

To demonstrate the capabilities of this CNN model in classifying break-junction measurements trace-by-trace as well as those that have been used in the literature, we collect STM-BJ data using three commercial compounds: 1,6-diaminohexane (**1**), 4,4'-bis(methylthiol)biphenyl (**2**) and 1,6-bis(methylthiol)hexane (**3**) (structures shown in Figure 3a). We measure each molecule individually and as mixed solutions (**1** with **2**, and **2** with **3)** in 1,2,4-trichlorobenzene (TCB). The 1D and 2D histograms of the **1**/**2** mixture are shown in Figure 1c and 1d (and those of the **2**/**3** mixture are shown in SI Figure S2c and S2d). As an example, we train this CNN model on data obtained from measurements of pure **1** and pure **2**, and an accuracy of 97.6% is achieve on this test dataset (based on analyzing 10% traces that were not used in training). We use this trained model to label the traces from mixed **1**/**2** solution measurement and plot the 1D histogram of all the traces classified to be **1**- and **2**-like by the model in Figure 3b. Figure 3c shows the corresponding 2D histograms. These histograms are very much like those measured on pure **1** and pure **2** (shown in SI Figure S1a-S1d). We do not see a peak at the conductance value corresponding to **2** in the **1**-like traces and vice versa indicating that the model is highly accurate. The corresponding classification result using model designs reported by others are shown in SI Figure S3; the accuracies of these models on pure molecule-test datasets are significantly lower (Table 1). We also train this model in the same way on the **2**/**3** data, and obtain a 95.9% accuracy on the pure molecule test dataset. The 1D and 2D histograms of the algorithm-labeled traces from mixed **2**/**3** solution measurements are shown in Figure 3d and 3e. We can see this CNN model performs extremely well in sorting data corresponding to molecules that have different backbone structures (alkane versus phenylenes). For molecule pairs with the same backbones (for example two alkanes such as the **1**/**3** pair, shown in Table 1), the classification accuracy is lower (89.6% on the test dataset). This indicates that the deep learning algorithm picks out features in the conductance traces that are likely related to the molecular backbone rather than the linker. It is possibly that the backbone contributes more to the trace properties such as the conductance value and plateau length.

We next apply our CNN model to characterize conductance data measured with [3]cumulene derivatives **4** and **5** (structures shown in Figure 4a). We recently discovered and reported that the electric field in STM-BJ setup can isomerize the *cis*-isomer **4** to the *trans*-isomer **5** *in situ*.[13] In this experiment, we recorded more than 100,000 conductance traces over a period of 30-hour. By training the CNN model on measurements of pure **4** and **5** (achieving an 88.4% accuracy on the test dataset) and then applying it to the large data set, we determine the ratio of the *cis*-isomer **4** to the *trans*-isomer **5** as a function of time. Figure 4b shows this ratio determined from sets of 1,000 traces. From Figure 4b, we can observe the transformation of **4** to **5** during the timescale of the measurement. To demonstrate the performance this classification, we show the 1D and 2D histograms of the algorithm-labeled traces from a set of 10,000 traces measured at about 22 hrs after the start of the measurement in Figure 4c and 4d. We can see that these histograms have a very similar appearance comparing to the histograms of pure *cis*-isomer **4** and *trans*-isomer **5** (SI Figure S1g-S1j), highlighting the accuracy of our model.

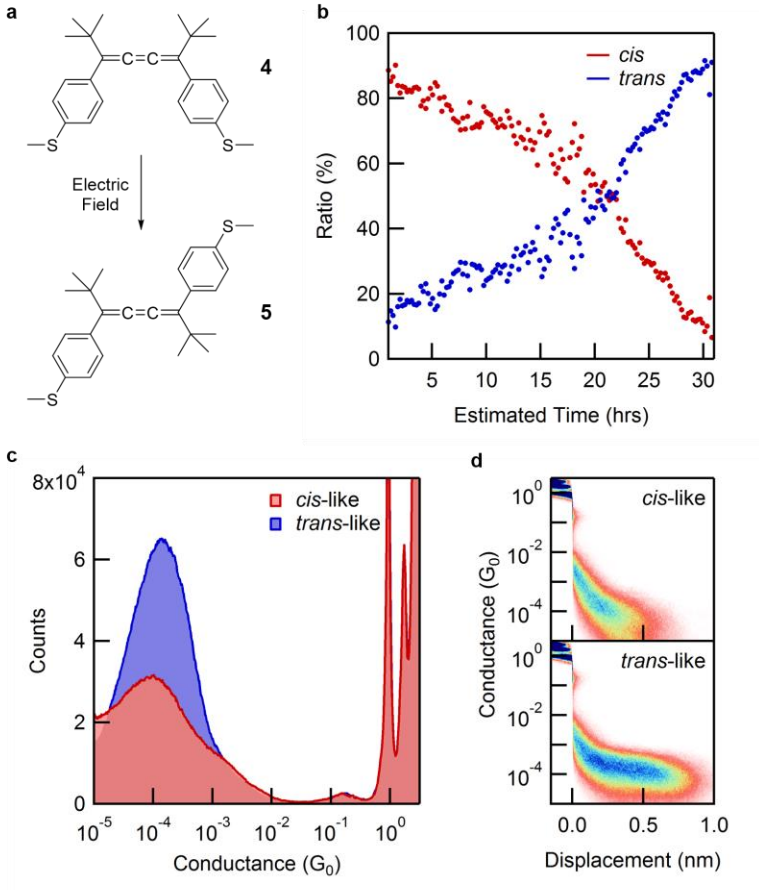
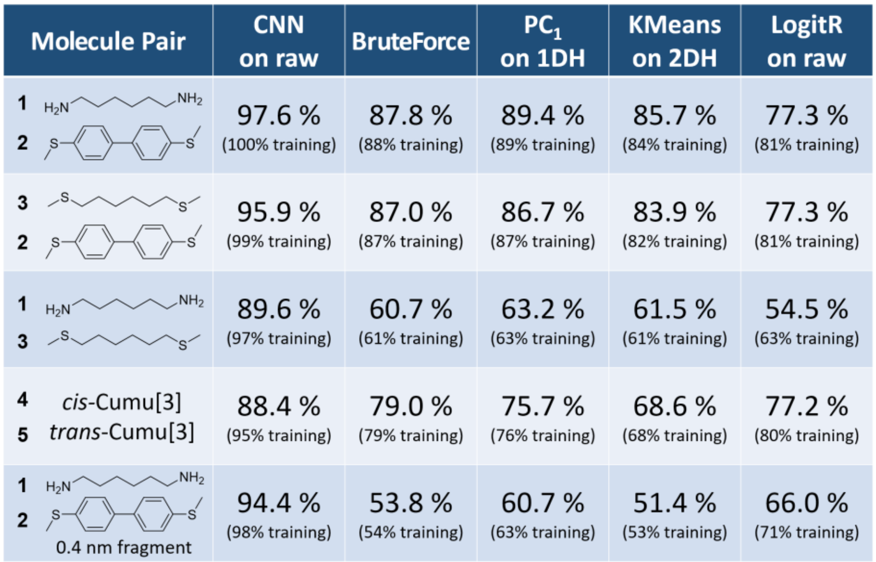


Figure 4. (a) Chemical structures of the [3]cumulene derivatives. Under electric field, the *cis*-isomer (**4**) transforms into the *trans*-isomer (**5**). (b) The percentage **4** (red dots) and **5** (blue dots)

as a function of time as determined by the CNN model. (c) The 1D and (d) 2D histograms of the traces judged to be **4**-like (4997 traces) or **5**-like (4994 traces) by the CNN model from the 10000 traces measured 22 hrs after starting with a pure **4** solution.

In Table 1, we show results from applying the alternative models to sort different conductance data. We test the PC_1/1DH and KMeans/2DH models (taken from the literatures[14, 15]) and also introduce two additional ones. The first is a "*brute force*" method, which uses individual trace conditional histogram, and then sorts data based on the number of counts within different conductance regions.[29] The second is a naïve logistic regression (LogitR), which does a logistic regression on the raw clipped conductance trace as a series of independent variables; this method is a simple linear model using the same input as the CNN model introduced in this work. We can see from the first column of Table 1 that the CNN model performs significantly better than all these simpler models for the mixed **1**/**2** molecule pair. Thus, although CNN needs more computational power for the training step, its extra complexity yields higher classification accuracy. In addition to these comparisons, we also apply a reported recurrent neural network-based model aims for classification on rupture process of Au-Au contact[20] for reference, and find it performs lower accuracy on this molecule recognition problem as detailed in the SI Section 6.



Molecule Pair	CNN on raw	BruteForce	PC_1 on 1DH	KMeans on 2DH	LogitR on raw
1 2	97.6 % (100% training)	87.8 % (88% training)	89.4 % (89% training)	85.7 % (84% training)	77.3 % (81% training)
3 2	95.9 % (99% training)	87.0 % (87% training)	86.7 % (87% training)	83.9 % (82% training)	77.3 % (81% training)
1 3	89.6 % (97% training)	60.7 % (61% training)	63.2 % (63% training)	61.5 % (61% training)	54.5 % (63% training)
4 *cis*-Cumu[3] 5 *trans*-Cumu[3]	88.4 % (95% training)	79.0 % (79% training)	75.7 % (76% training)	68.6 % (68% training)	77.2 % (80% training)
1 2 0.4 nm fragment	94.4 % (98% training)	53.8 % (54% training)	60.7 % (63% training)	51.4 % (53% training)	66.0 % (71% training)

Table 1. The comparison among the reported and proposed models described in the main text. The accuracies of different models on different molecule pairs are shown in the table. For each experiment shown in each cell, 90% of the labeled dataset are used to train the model, while the

remaining 10% are used for testing. In each cell, the accuracy on the test dataset is shown in the center, and the accuracy on the training dataset given in parenthesis.

The accuracy of these four models on all other systems considered here are also shown in Table 1. For sorting the **1/3** mixture (the 3rd row) where the backbones are the same and the individual molecular conductances are also similar (both at ~2×10^{-4} G_0), the accuracy is lower for all models when compared to the **1/2** and **2/3** mixtures. However, the drop in accuracy for the CNN model sorting the **1/3** mixture is much smaller than for other models. This implies that the CNN model can identify trace characteristics beyond simply the conductance value. To test if this is indeed the case, we design a reference analysis where the average plateau conductance information is removed (5th row of Table 1). Instead of using the clipped conductance trace as input, we use a randomly selected 0.4 nm-long fragment of molecular conductance plateau from the clipped trace and then subtract the average conductance value of this segment from this data, in order to remove the influence from conductance value as well as plateau length. We then test all models using this new input. The accuracies of all the models decrease, but for the CNN model, the accuracy remains reasonably high (94.4% on the test dataset).

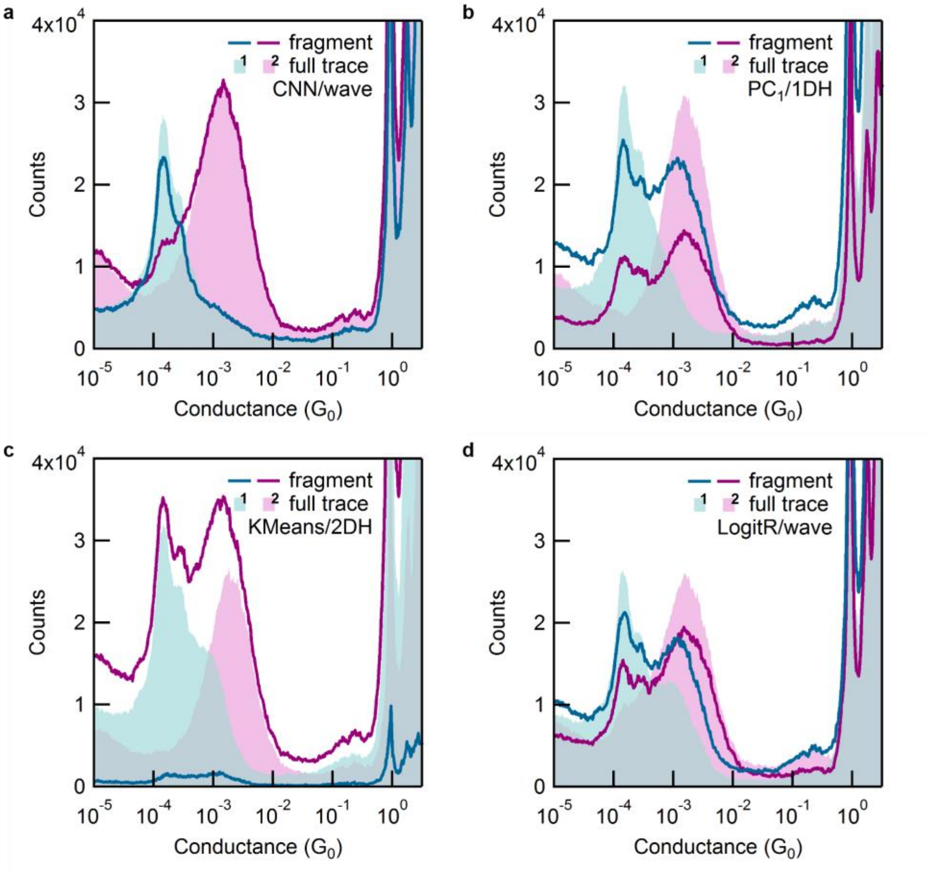


Figure 5. The 1D histograms of **1**-like (the light blue) or **2**-like (the magenta) traces sorted by different models from mixed solution measurements. The classification results based on using 0.4 nm fragments as inputs are shown as solid lines. As a reference, classification results using the clipped trace as input, are reproduced here as shaded regions. (a) The CNN model applied to 0.4 nm fragments yield 3066 **1**-like and 5216 **2**-like traces (compared with 3406 **1**-like and 4876 **2**-like traces when using the full trace). (b) The PC$_1$/1DH model applied to 0.4 nm fragments yield 6053 **1**-like and 2229 **2**-like traces (compared with 4397 **1**-like and 3901 **2**-like traces when using full trace). (c) The KMeans/2DH model applied to 0.4 nm fragments yield 392 **1**-like and 7890 **2**-like traces (compared with 5260 **1**-like and 3022 **2**-like traces when using full trace). (d) Logistic regression model applied to 0.4 nm fragments yield 4730 **1**-like and 3553 **2**-like traces (compared with 4569 **1**-like and 3713 **2**-like traces when using full trace).

We next demonstrate the classifications of traces excluding the average conductance information on the mixture solution of **1** and **2** in Figure 5. The significant result here, shown in Figure 5a, is that discarding the average conductance information does not yield very different results when using the CNN model, showing its robustness against the elimination of average conductance information. For the other models, discarding conductance information produces a sorting that is more random. This indicates that these models rely strongly on the average conductance information.

In conclusion, we have demonstrated a new deep learning-based model to recognize molecular junction measurements performed with the STM-BJ technique that enables an accurate classification and characterization of molecular types. Comparing our model to some widely used and recently reported ones, we show that the CNN-based method achieves a much higher accuracy and importantly is able to sort traces without relying on the average conductance information, a critical innovation of this work. We demonstrate the application of this model to measurements of mixtures of molecules and also apply it to monitor an *in situ* chemical reaction that is driven by the electric field during STM-BJ experiment. The excellent performance and robustness of this model makes it a favorable algorithm for analyzing such data. Its high-accuracy will enable more detailed investigations on systems with mixture of different kinds of molecular junctions, including, for example *in situ* reaction and surface chemistry.

ASSOCIATED CONTENT

Supporting Information

Additional figures, synthetic details and model details. The Supporting Information is available free of charge on the ACS Publications website.

AUTHOR INFORMATION

Corresponding Author

lv2117@columbia.edu

Notes

The authors declare no competing financial interests.

ACKNOWLEDGMENT

We acknowledge financial support from the National Science Foundation under award grant CHE-1764256 and DMR-1807580. Q.Z. acknowledges financial support from Shanghai University of Electric Power, the Science and Technology Commission of Shanghai Municipality (14DZ2261000).

Reference

1. Xu, B.; Tao, N. J. *Science* **2003,** 301, (5637), 1221-3.

2. Venkataraman, L.; Klare, J. E.; Nuckolls, C.; Hybertsen, M. S.; Steigerwald, M. L. *Nature* **2006,** 442, (7105), 904-7.

3. Reed, M. A. *Science* **1997,** 278, (5336), 252-254.

4. Smit, R. H.; Noat, Y.; Untiedt, C.; Lang, N. D.; van Hemert, M. C.; van Ruitenbeek, J. M. *Nature* **2002,** 419, (6910), 906-9.

5. Xu, B. Q.; Xiao, X. Y.; Tao, N. J. *Journal of the American Chemical Society* **2003,** 125, (52), 16164-16165.

6. Reddy, P.; Jang, S. Y.; Segalman, R. A.; Majumdar, A. *Science* **2007,** 315, (5818), 1568-1571.

7. Frei, M.; Aradhya, S. V.; Koentopp, M.; Hybertsen, M. S.; Venkataraman, L. *Nano Lett* **2011,** 11, (4), 1518-23.

8. Widawsky, J. R.; Darancet, P.; Neaton, J. B.; Venkataraman, L. *Nano Lett* **2012,** 12, (1), 354-8.

9. Adak, O.; Rosenthal, E.; Meisner, J.; Andrade, E. F.; Pasupathy, A. N.; Nuckolls, C.; Hybertsen,

M. S.; Venkataraman, L. *Nano Lett* **2015,** 15, (6), 4143-9.

10. Aragones, A. C.; Haworth, N. L.; Darwish, N.; Ciampi, S.; Bloomfield, N. J.; Wallace, G. G.; Diez-Perez, I.; Coote, M. L. *Nature* **2016,** 531, (7592), 88-91.

11. Huang, X.; Tang, C.; Li, J.; Chen, L. C.; Zheng, J.; Zhang, P.; Le, J.; Li, R.; Li, X.; Liu, J.; Yang, Y.; Shi, J.; Chen, Z.; Bai, M.; Zhang, H. L.; Xia, H.; Cheng, J.; Tian, Z. Q.; Hong, W. *Sci Adv* **2019,** 5, (6), eaaw3072.

12. Zang, Y.; Pinkard, A.; Liu, Z. F.; Neaton, J. B.; Steigerwald, M. L.; Roy, X.; Venkataraman, L. *J Am Chem Soc* **2017,** 139, (42), 14845-14848.

13. Zang, Y.; Zou, Q.; Fu, T.; Ng, F.; Fowler, B.; Yang, J.; Li, H.; Steigerwald, M. L.; Nuckolls, C.; Venkataraman, L. *Nat Commun* **2019,** 10, (1), 4482.

14. Cabosart, D.; El Abbassi, M.; Stefani, D.; Frisenda, R.; Calame, M.; van der Zant, H. S. J.; Perrin, M. L. *Applied Physics Letters* **2019,** 114, (14).

15. Hamill, J. M.; Zhao, X. T.; Meszaros, G.; Bryce, M. R.; Arenz, M. *Phys Rev Lett* **2018,** 120, (1), 016601.

16. Inkpen, M. S.; Lemmer, M.; Fitzpatrick, N.; Milan, D. C.; Nichols, R. J.; Long, N. J.; Albrecht, T. *J Am Chem Soc* **2015,** 137, (31), 9971-81.

17. Magyarkuti, A.; Balogh, N.; Balogh, Z.; Venkataraman, L.; Halbritter, A., Unsupervised feature recognition in single molecule break junction data. 2020, *arXiv* https://arxiv.org/abs/2001.03006 (accessed March 23, 2020).

18. Albrecht, T.; Slabaugh, G.; Alonso, E.; Al-Arif, S. *Nanotechnology* **2017,** 28, (42), 423001.

19. Korol, R.; Segal, D. *J Phys Chem B* **2019,** 123, (13), 2801-2811.

20. Lauritzen, K. P.; Magyarkuti, A.; Balogh, Z.; Halbritter, A.; Solomon, G. C. *J Chem Phys* **2018,** 148, (8), 084111.

21. Huang, F.; Li, R.; Wang, G.; Zheng, J.; Tang, Y.; Liu, J.; Yang, Y.; Yao, Y.; Shi, J.; Hong, W. *Phys Chem Chem Phys* **2020**.

22. Rawat, W.; Wang, Z. *Neural Comput* **2017,** 29, (9), 2352-2449.

23. Arthur, D.; Vassilvitskii, S., k-means++: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics: New Orleans, Louisiana, 2007; pp 1027-1035.

24. Lloyd, S. *IEEE Transactions on Information Theory* **1982,** 28, (2), 129-137.

25. Nair, V.; Hinton, G. E., Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress: Haifa, Israel, 2010; pp 807-814.

26. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. *J. Mach. Learn. Res.* **2014,** 15, (1), 1929-1958.

27. Chen, Y.; Fan, H.; Xu, B.; Yan, Z.; Kalantidis, Y.; Rohrbach, M.; Yan, S.; Feng, J., Drop an

Octave: Reducing Spatial Redundancy in Convolutional Neural Networks with Octave Convolution. 2019, *arXiv* https://arxiv.org/abs/1904.05049 (accessed March 23, 2020).

28. Ioffe, S.; Szegedy, C., Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. 2015, *arXiv* https://arxiv.org/abs/1502.03167 (accessed March 23, 2020).

29. Aradhya, S. V.; Frei, M.; Halbritter, A.; Venkataraman, L. *ACS Nano* **2013,** 7, (4), 3706-12.