Supporting Information

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

Email: lv2117@columbia.edu

Table of Contents:

1. STM-BJ Experiments in Detail

All the scanning tunneling microscope-break junction (STM-BJ) experiments in this work are conducted in ambient conditions.[1, 2] A gold tip and a gold-coated substrate are used as the two electrodes. To create molecular junctions, a piezo actuator is used to drive the tip and it is moved in and out of contact of the substrate at a rate of 20 $nm \cdot s^{-1}$. During a measurement, the voltage (V) and current (I) across the junction are continuously recorded, and conductance G is calculated as $G = I/V$. These values are measured and recorded in a sampling frequency of 40 kHz. Hence, a 1 nm displacement corresponds to 2,000 data points.

To form junctions with molecules, a dilute solution of analytes is added on the substrate and measurements are made within this solution environment. For measurements with compounds **1**, **2** and **3**, we use 1,2,4-trichlorobenzene (TCB) as the solvent and apply a bias of 250 mV across the junction. For measurements with compounds **4** and **5**, we use *n*-tetradecane (TD) as the solvent and apply a bias of 100 mV across the junction. All the pure molecule measurements (shown in Figure S1) are done with solution concentration of 0.1 mM. The measurement of mixtures of **1** and **2** shown in Figure 3b, 3c, S2a and S2b have ~0.05 mM of **1** and **2**. The measurement of mixtures of **2** and **3** shown in Figure 3d, 3e, S2c and S2d have 0.01 mM of **2** and 0.0025 mM of **3**. The *in-situ* isomerization experiment of **4** starts with a 0.1 mM TD solution of **4**. The histograms shown in Figure 4c, 4d, S2e and S2f are from 10000 traces measured ~22 hours after the experiment was started.

The compound **1** and **2** are obtained from Aldrich, and **3** from Alfa Aesar. These are used without further purification. The synthesis of compound **4** and **5** is reported in our previous work.[3]

2. Structure of OctConv Layers.

As briefly discussed in the main text, we apply OctConv-style convolutional layers.[4] As shown in Figure 2c in the main text, one OctConv layer has a pair of inputs and outputs (a high-frequency and a low-frequency branch). The high-frequency branch uses the original input data, and the low-frequency branch has half the number of data points. Without the low-frequency input and output, the OctConv layer is the same as the vanilla convolutional layer.

Each column in Figure 2c is a vanilla convolutional layer structure, where the input matrix goes through a 1D convolution operation, and then ReLU. Batch normalization (BatchNorm) technique is applied to achieve a better and faster training, by normalizing the output values of

convolution into zero mean-unit standard deviation during the training.[5] As illustrated in Figure 2c, in one OctConv layer there are four of such vanilla convolutional columns; low-to-low, low-to-high, high-to-low and high-to-high. Among them, the low-to-low and high-to-high do not involve any transformation because the lengths of input and output are the same. For low-to-high, a nearest neighbor interpolation is used to double the data length by duplicating every data point. For high-to-low, an average pooling operation, which replaces two neighboring values by their mean is used to half the length.

The OctConv layers have two pairs of input and output, so when connecting two of them, we connect low-frequency to low-frequency, high-frequency to high-frequency. The other components of the network, however, have only one pair of input and output. As the high-frequency branch is the main stream, it is always retained. The low-frequency input/output is not used when connected to other components of the network. This means that two of the four columns of convolution (low-to-low and either low-to-high or high-to-low) are discarded because the low-frequency is not provided or is not generated.

3. CNN Model.

The convolutional neural network (CNN) model described in the main text is an CNN-based model taking conductance traces as input and determining a class label as output. The model takes a 2000-point-long 1D vector (segment of the conductance trace) as input. This corresponds to 1 nm of displacement in the measurement. As described in the main text (and illustrated in Figure 2b and 2c), the input is first processed by 6 OctConv layers. The underlying convolution operations use 1D kernel sizes of 7, 7, 7, 9, 9, 9, respectively for each layer. The numbers of channels are 32, 32, 32, 64, 128, 256, respectively. If an OctConv layer produces both high-frequency and low-frequency outputs, half of the channel number is distributed to either of them. For example, the third OctConv layer has 16 high-frequency channels and 16 low-frequency channels, giving a total of 32 channels. After all convolutional layers, the data is flattened from 2D (spatial × channels) into 1D vectors, so that they can be further fed into fully-connected layers. The following two fully-connected layers have size of 1024 and 16, respectively. The dropout[5] rate applied on the fully-connected layers is 20%. Finally, after going through a sigmoid function, the output is generated as 1-bit probability of 0 or 1 providing a probability of the trace having the first label (0) or second label (1).

This model is trained on the TensorFlow platform.[6] For model training, a batch size of 32 and learning rate of 3×10^{-6} are used, with an Adam optimizer[7]. A weight decay[8, 9] of 10% learning rate is applied to all the trainable variable in the whole model (except for BatchNorm) to provide extra regularization. These hyperparameters were not deliberately tuned in this work; fine tuning or improvement on the model is left for future work.

Preprocessing of conductance traces.

In order to focus on the molecular plateaus, we remove the gold conductance region of a trace (conductance > 10^{-1} G_0) before feeding it to the model. We also remove the noise floor (conductance < 10^{-5} G_0). In addition, all traces are aligned at close to the point when atomic Au-Au contact breaks (chosen to be 0.5 G_0), as is used in creating 2D conductance-displacement histograms. Finally, we feed only the first 2000 points of data to the model, which represent the first nanometer of conductance data after Au-Au contact has ruptured.

Modifications analyzing data without including conductance.

In the main text and Figure 5, we discuss a reference analysis when average plateau conductance is removed from the input during the training process. As we described in the main text, here, we use an 800-point-long (0.4 nm) segment of conductance plateau. These segments are randomly cut from molecular conductance plateaus in traces. We first take the logarithm of the values and then subtract the segment average. The input size of the CNN model is set to 800 points. We also change the flatten operation into a global mean operation, where for each channel, it returns the average value. Compared to flatten, global mean does not keep spatial information, and hence more appropriate for this type of input as the absolute values of displacement do not have physical meaning here. While for the training process the segments are randomly cut from plateaus, for the recognition process, the only first 800 points after rupture of Au-Au contact is used to ensure that only one segment is generated from each trace.

4. Additional STM-BJ Histograms

STM-BJ histograms of pure molecular solutions.

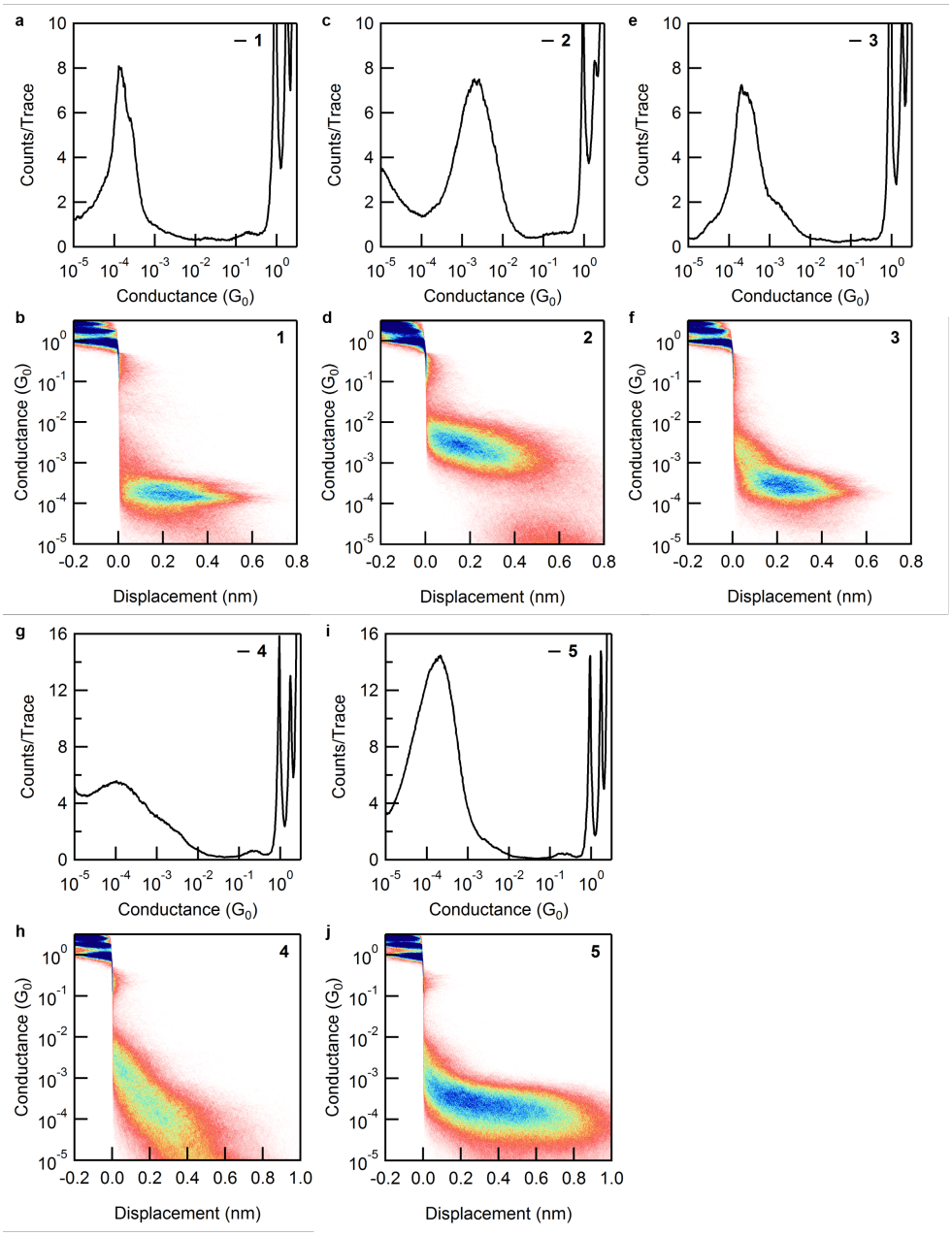

Figure S1. (a) 1D and (b) 2D conductance histograms of 1,6-diaminohexane (**1**). (c) 1D and (d) 2D conductance histograms of 4, 4'-bis(methylthiol)biphenyl (**2**). (e) 1D and (f) 2D conductance histograms of 1,6-bis(methylthiol)hexane (**3**). (g) 1D and (h) 2D conductance histograms of **4**. (i) 1D and (j) 2D conductance histograms of **5**.

STM-BJ Histograms of Mixtures.

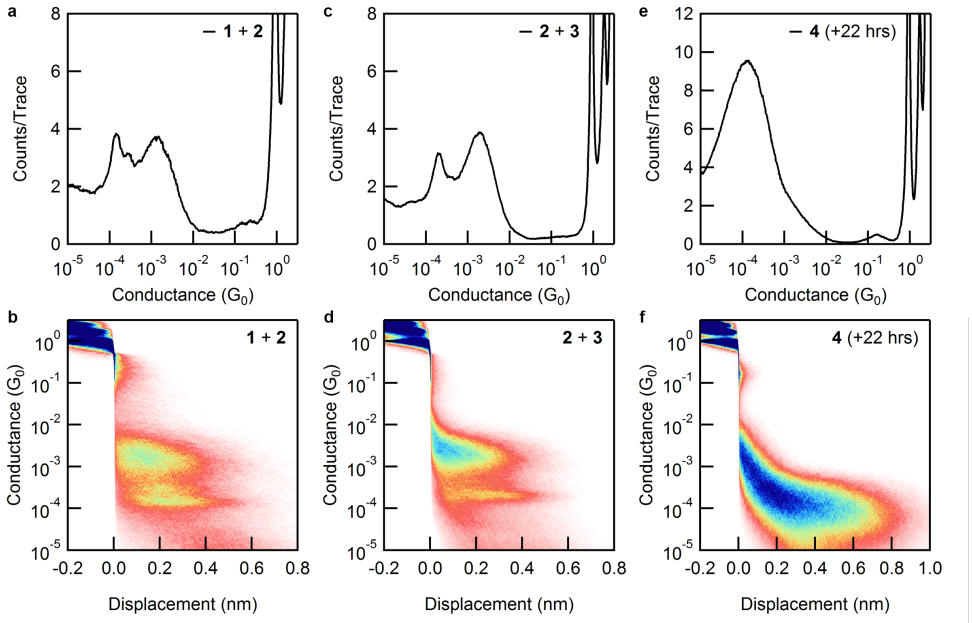

Figure S2. (a) 1D and (b) 2D conductance histograms of the mixture of **1** and **2**. These are the same histograms as Figure 1c and 1d except that they are rotated. (c) 1D and (d) 2D conductance histograms of the mixture of **2** and **3**. (e) 1D and (f) 2D conductance histograms of the traces measured 22 hours after the experiment starting with pure **4**.

5. Classification Results of All Models

Results from different models with different molecule pairs shown in Table 1.

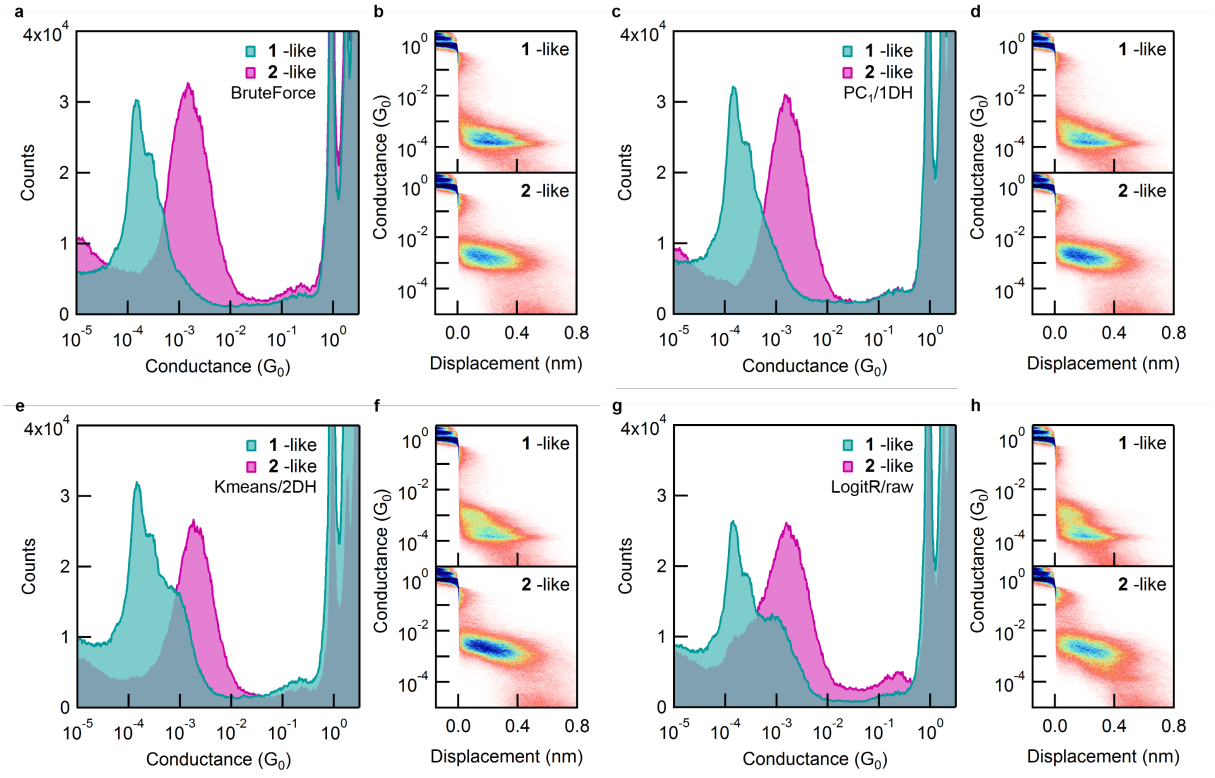

Figure S3. Histograms of the traces judged to be **1**-like or **2**-like from measurements of a mixed solution. (a) The 1D and (b) 2D histograms classified by the *brute force* model: there are 3782 **1**-like traces and 4516 **2**-like traces. (c) The 1D and (d) 2D histograms classified by the PC_1/1DH model: there are 4397 **1**-like traces and 3901 **2**-like traces. (e) The 1D and (f) 2D histograms classified by the KMeans/2DH model: there are 5260 **1**-like traces and 3022 **2**-like traces. (g) The 1D and (h) 2D histograms classified by the logistic regression model on raw traces: there are 4569 **1**-like traces and 3901 **2**-like traces.

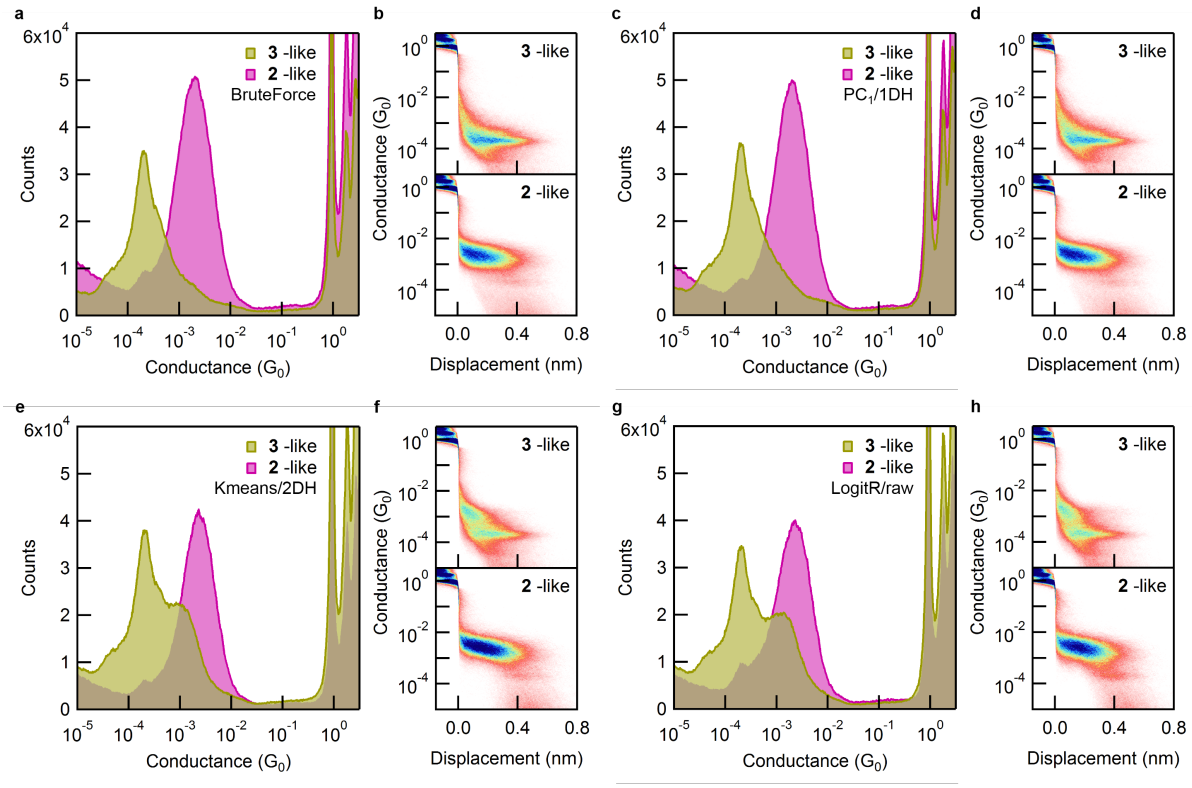

Figure S4. Histograms of the traces judged to be **2**-like or **3**-like from measurements of a mixed solution. (a) The 1D and (b) 2D histograms classified by the *brute force* model: there are 7062 **2**-like traces and 4737 **3**-like traces. (c) The 1D and (d) 2D histograms classified by the PC_1/1DH model: there are 6549 **2**-like traces and 5250 **3**-like traces. (e) The 1D and (f) 2D histograms classified by the KMeans/2DH model: there are 4625 **2**-like traces and 7151 **3**-like traces. (g) The 1D and (h) 2D histograms classified by the logistic regression model on raw traces: there are 4959 **2**-like traces and 6817 **3**-like traces.

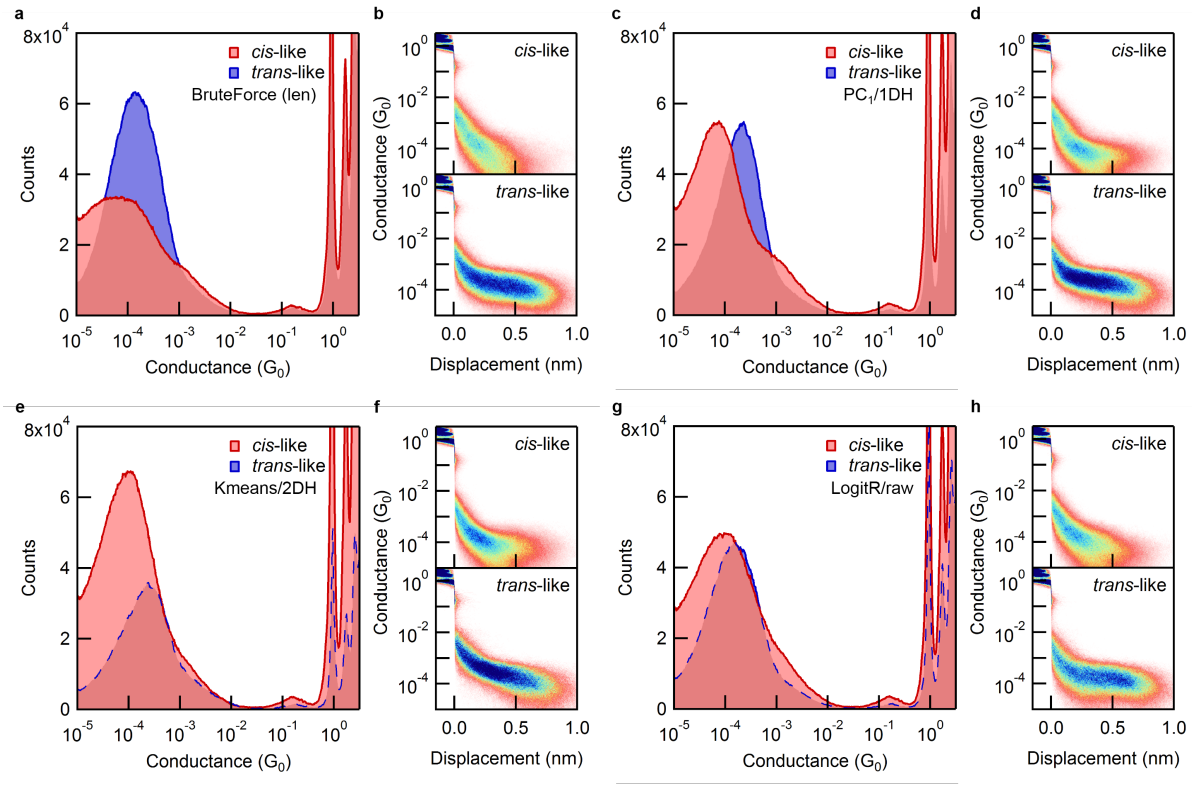

Figure S5. Histograms of the traces judged to be **4**-like or **5**-like from the traces measured after ~22 hrs starting with a pure **4** solution. (a) The 1D and (b) 2D histograms classified by the *brute force* model: there are 5653 *cis*-like (**4**-like) traces and 4338 *trans*-like (**5**-like) traces; here the *brute force* model judges based on counts of points in the molecular conductance region. (c) The 1D and (d) 2D histograms classified by the PC_1/1DH model: there are 6577 *cis*-like traces and 3414 *trans*-like traces. (e) The 1D and (f) 2D histograms classified by the KMeans/2DH model: there are 7552 *cis*-like traces and 2439 *trans*-like traces. (g) The 1D and (h) 2D histograms classified by the logistic regression model on raw traces: there are 6691 *cis*-like traces and 3300 *trans*-like traces.

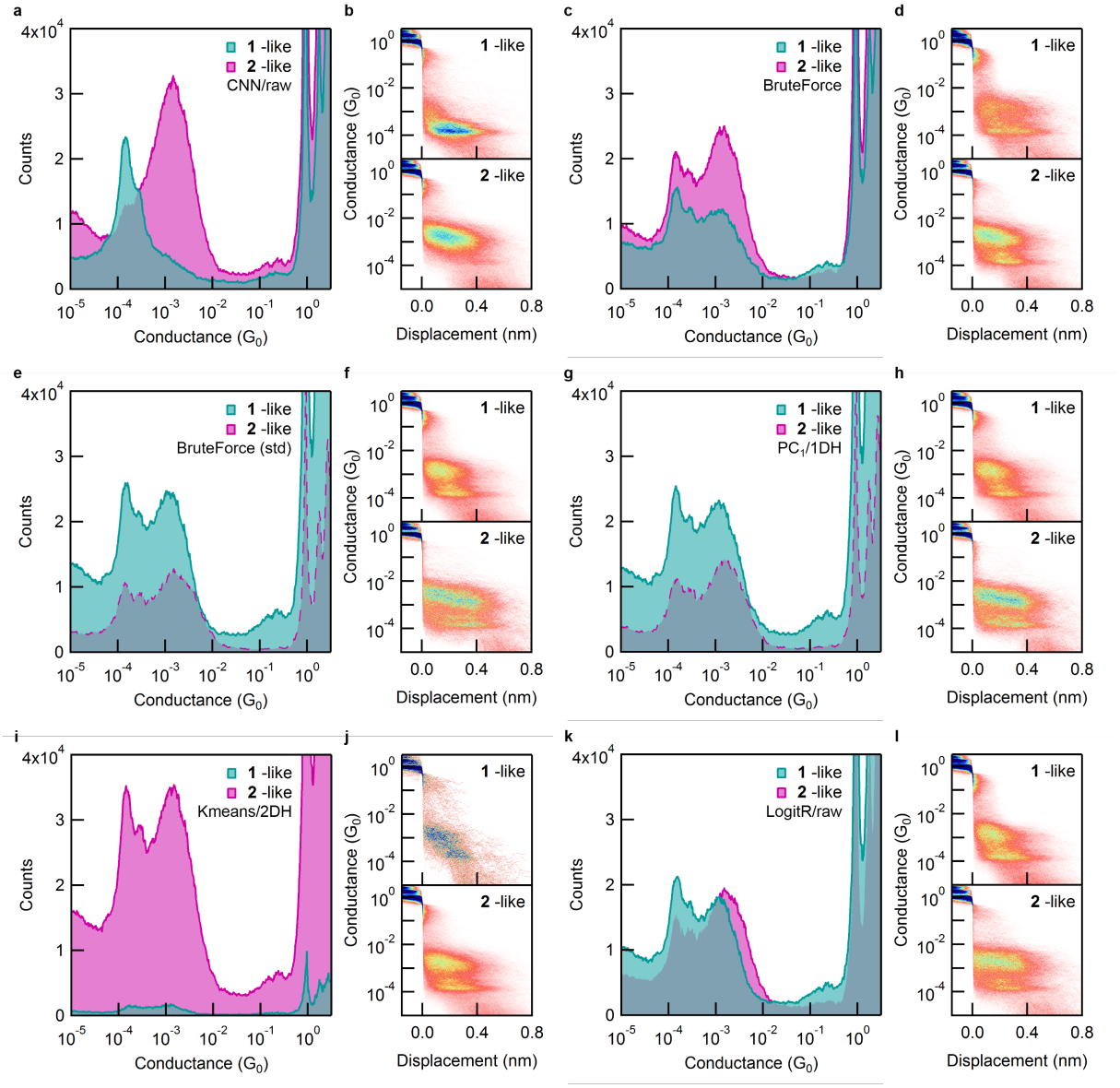

Figure S6. Histograms of the traces judged to be **1**-like or **2**-like with average plateau conductance removed from measurements a mixed solution. (a) The 1D and (b) 2D histograms classified by the CNN model with 3066 **1**-like and 5216 **2**-like traces. (c) The 1D and (d) 2D histograms classified by the *brute force* model with 3245 **1**-like and 5037 **2**-like traces. (e) The 1D and (f) 2D histograms classified by modified *brute force* model which uses the standard deviation of the sections with 6331 **1**-like and 1951 **2**-like traces; (g) The 1D and (h) 2D histograms classified by the PC_1/1DH model with 6053 **1**-like and 2229 **2**-like traces. (i) The 1D and (j) 2D histograms classified by the KMeans/2DH model with are 392 **1**-like and 7890 **2**-like traces. (k) The 1D and (l) 2D histograms classified by the logistic regression model on raw traces with 4730 **1**-like and 3552 **2**-like traces.

Comparing performance of models in Figure 5 with random selection.

As shown in Figure 5, after changing the input from whole trace into segments of molecular conductance plateau with average plateau conductance removed, the performance of models other than the CNN model drops significantly. This reflects that these other models rely on the average conductance information. Figure S7 compares histograms made with randomly selected traces with those shown in Figure 5. We can see the PC_1/1DH model and KMeans/2DH model are close to a random selection.

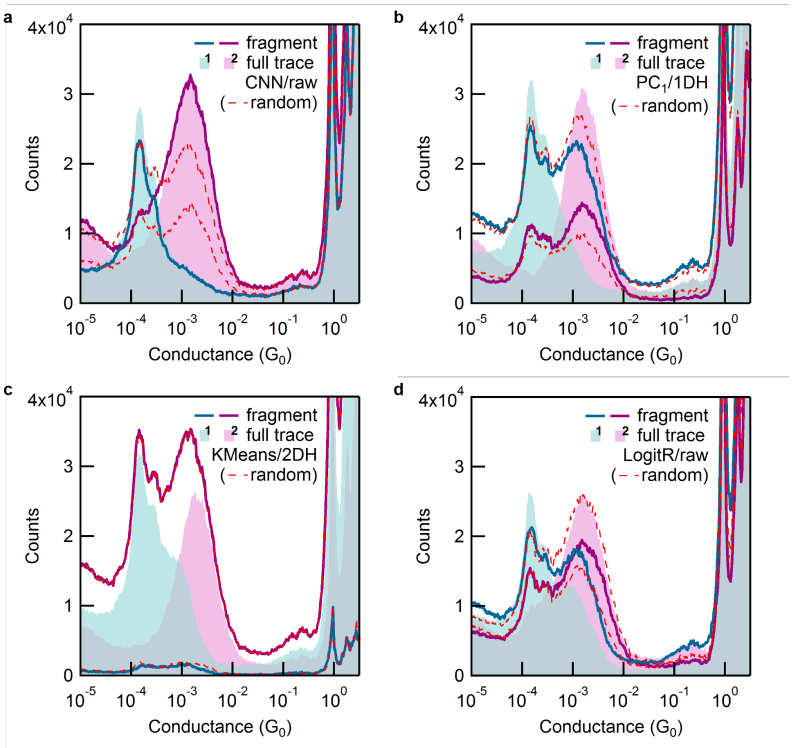

Figure S7. We compare the histograms shown in Figure 5 with ones generated from selecting the trace class randomly while keeping a similar number of traces as the sorted histograms to maintain a similar histogram height (red dashed lines).

6. Comparison with LSTM.

Long short-term memory (LSTM) is a popular design of recurrent neural network (RNN). In the work of Lauritzen and co-authors,[10] an LSTM-based supervised classification method is developed to classify conductance traces measured during the rupture of gold point contacts. In these traces, the conductance decreases in steps with each step having a clear physical significance, i.e. narrowing of the gold neck. We however focus on the molecular conductance region below $1G_0$ where the conductance feature is closer to a plateau and with small conductance fluctuations. We believe that the RNN method is not ideally suited for these types of features, nonetheless, we have applied it and summarize our findings below.[10]

When training with a conductance trace data set measured with a one type of molecule, the LSTM model reaches a 89.6% accuracy on the test dataset (95% on training dataset). When training with segments of traces with average conductance information removed, the LSTM model reaches a 59.3% accuracy on testing dataset (80% on training dataset). This result shows that our CNN-base design is probably better at recognizing molecular conductance features.

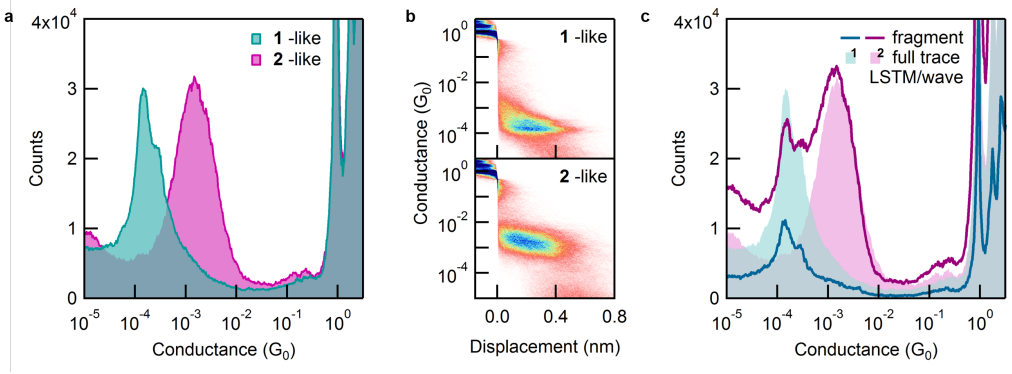

Figure S8. The histograms of the traces judged by the LSTM model. (a) 1D and (b) 2D histograms of the traces judged to be **1**-like (4124 traces) or **2**-like (4181 traces) with full conductance traces as input. (c) Solid lines are 1D histograms of the traces judged to be **1**-like (1586 traces) or **2**-like (6987 traces) with average conductance information removed from the input; the shade regions are the histograms copied from (a) as a reference.

References

(1) Xu, B.; Tao, N. J., Measurement of single-molecule resistance by repeated formation of molecular junctions, *Science,* **2003,** 301, 1221-3.

(2) Venkataraman, L.; Klare, J. E.; Nuckolls, C.; Hybertsen, M. S.; Steigerwald, M. L., Dependence of single-molecule junction conductance on molecular conformation, *Nature,* **2006,** 442, 904-7.

(3) Zang, Y., et al., Directing isomerization reactions of cumulenes with electric fields, *Nat. Commun.,* **2019,** 10, 4482.

(4) 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. In *arXiv* *https://arxiv.org/abs/1904.05049* *(accessed March 23, 2020)*, 2019.

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

(6) Abadi, M. n., et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. In *arXiv e-prints*, 2016.

(7) Kingma, D. P.; Ba, J., Adam: A Method for Stochastic Optimization. In *arXiv* *https://arxiv.org/abs/1412.6980* *(accessed March 23, 2020).* 2014.

(8) Krogh, A.; Hertz, J. A., A simple weight decay can improve generalization. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, Morgan Kaufmann Publishers Inc.: Denver, Colorado, 1991; pp 950-957.

(9) Loshchilov, I.; Hutter, F., Decoupled Weight Decay Regularization. In *arXiv* *https://arxiv.org/abs/1603.04467* *(accessed March 23, 2020)*, 2017.

(10) Lauritzen, K. P.; Magyarkuti, A.; Balogh, Z.; Halbritter, A.; Solomon, G. C., Classification of conductance traces with recurrent neural networks, *J. Chem. Phys.,* **2018,** 148, 084111.