Supplementary information

Identifying charge density and dielectric environment of graphene using Raman spectroscopy and deep learning

Zhuofa Chen, Yousif Khaireddin, Anna Swan

Department of Electrical and Computer Engineering, Boston University, Boston

E-mail: zfchen@bu.edu

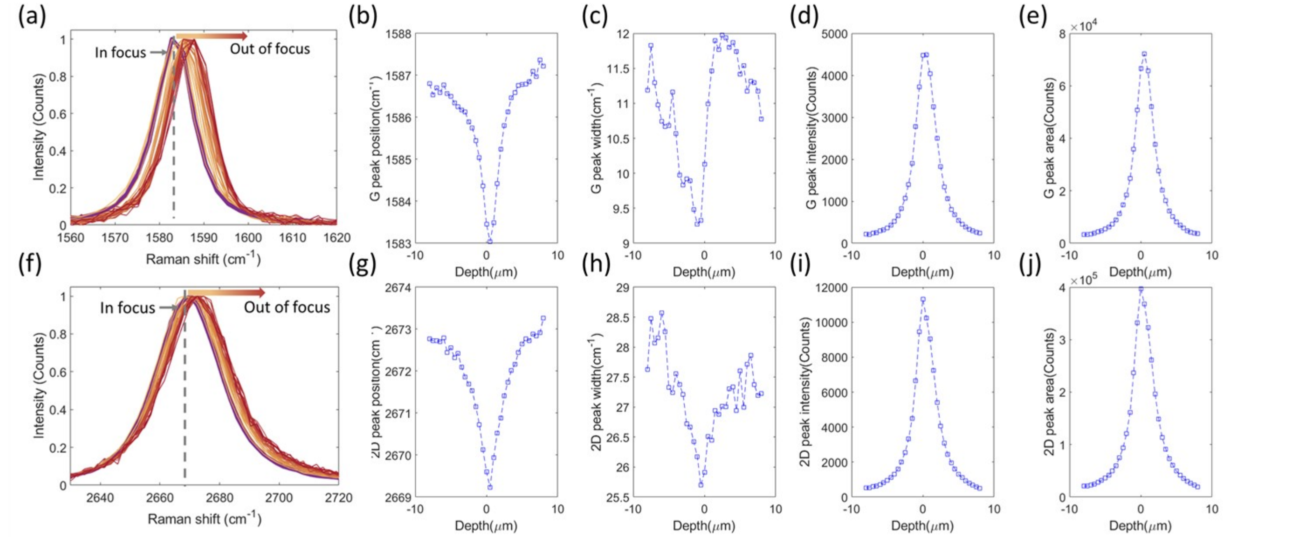

Figure S 1 Effect of focal plane on Raman spectra and the analysis results by curve fitting. (a) Raman G peak of graphene at different focus planes. (b-e) The curve fitting results of G peak position, G peak FWHM, G peak intensity, G peak area. (f) Raman 2D peak of graphene at different focus planes. (g-j) The fitting results of 2D peak position, 2D peak FWHM, 2D peak intensity, 2D peak area. The Raman spectra have different peak information at different focus planes even though we maintain all the other experimental conditions constant.

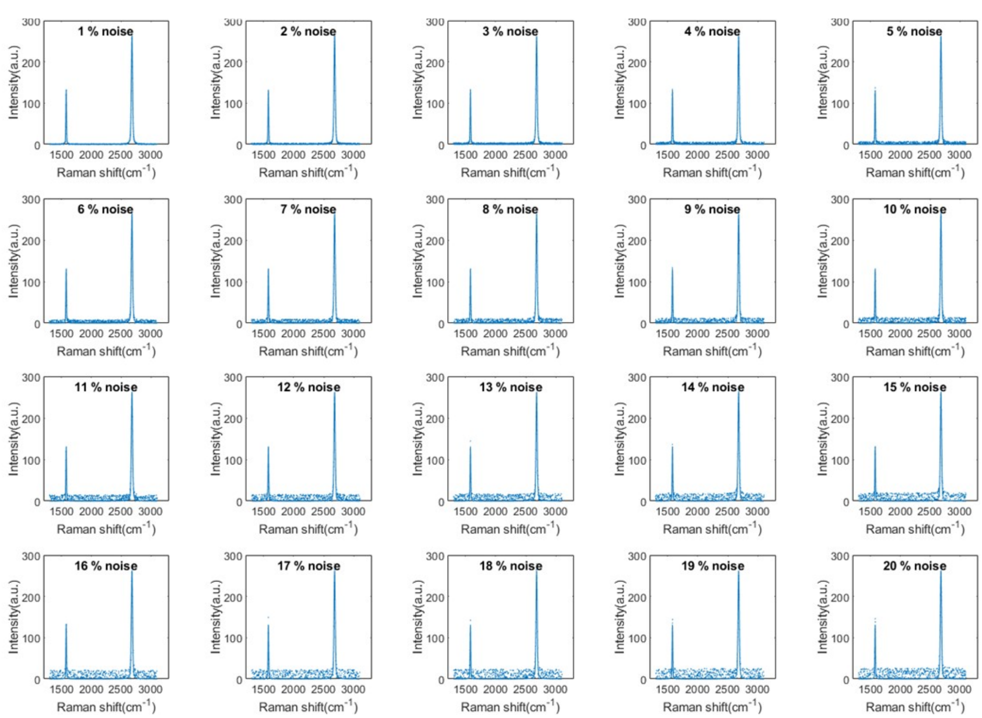

Figure S2 Raman spectra with different noise levels from 1 % to 20 %. Noise levels is defined as the noise fluatuation range over the maximum intensity of the spectra. Noise levels are generated by adding random distributed white noise to the spectra.

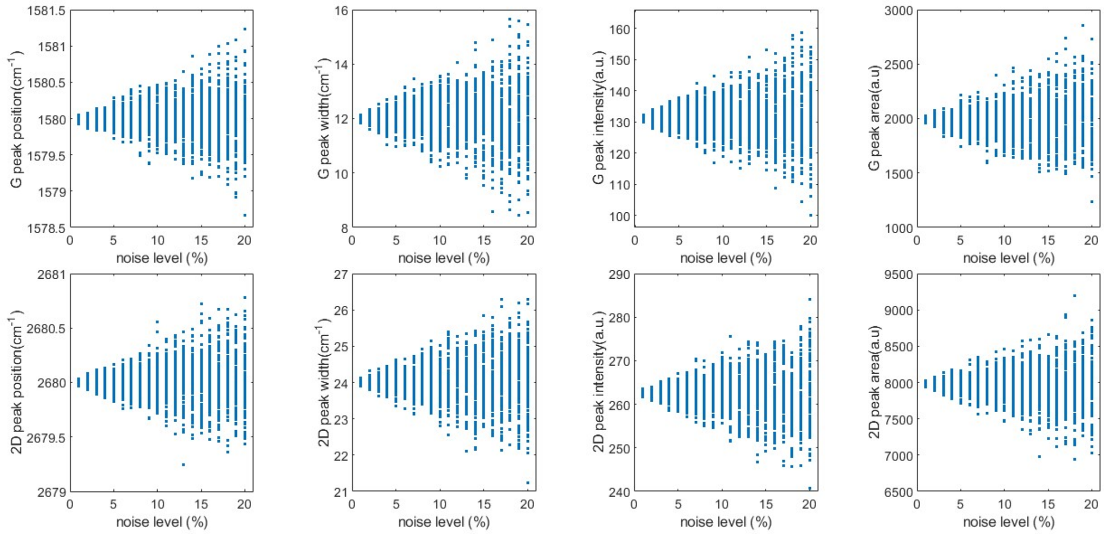

Figure S3 The effect on noise level on the curve fitting results. The top row is the variation of G peak position, G peak FWHM, G peak intensity, G peak area, respectively. The bottom row is the variation of 2D peak position, 2D peak FWHM, 2D peak intensity, 2D peak area, respectively. At each noise level, we artificially generate 100 spectra and the spectra are fitted with Voigt profile. The results show that the extracted parameters are highly dependent on the noise levels of the collected spectra.

Table 1 Charge density and correlated parameters extracted from transport measurement.

	CNP (V)	n_0 ($\times 10^{11} cm^{-2}$)	Δn($\times 10^{11} cm^{-2}$)	Mobility (cm^2/Vs)	Charge range : n0 ± Δn) ($\times 10^{11} cm^{-2}$)
C4	1.5	3.6	2.2	17k	1.4 to 5.8
C3	4.5	10.9	7.2	5.2k	3.7 to 18.1
C2	9.25	22.4	5.7	2k	16.7 to 28.1
C1	---	---	---	---	24 to 43

To study graphene with different charge densities, we fabricated a graphene field-effect transistor (GFET) on a 90 nm SiO_2 substrate. The charge neutrality point (CNP), average accidental doping (n_0), charge fluctuation (Δn), mobility, and charge range are shown in Table 1. C1 is graphene on SiO2, which has been studied extensively and the charge doping level ranges from 24 to $43 \times 10^{11} cm^{-2}$. C2, C3, and C4 are from graphene fabricated on OTMS treated substrate.

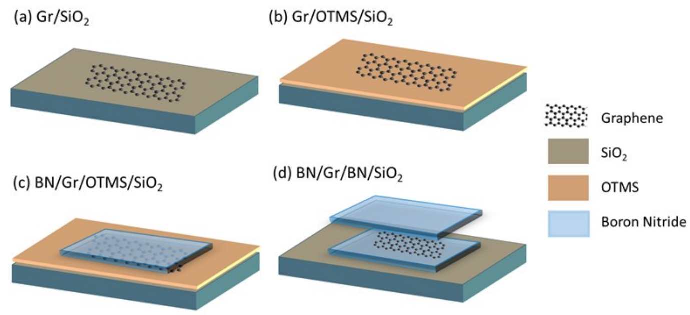

Figure S4 Schematic of graphene in different dielectric environments. (a) graphene on SiO2 substrate. (b) graphene on OTMS-treated SiO2 substrate. (c) Boron nitride/Graphene on OTMS treated SiO2 substrate. (d) graphene encapsulated between boron nitride.

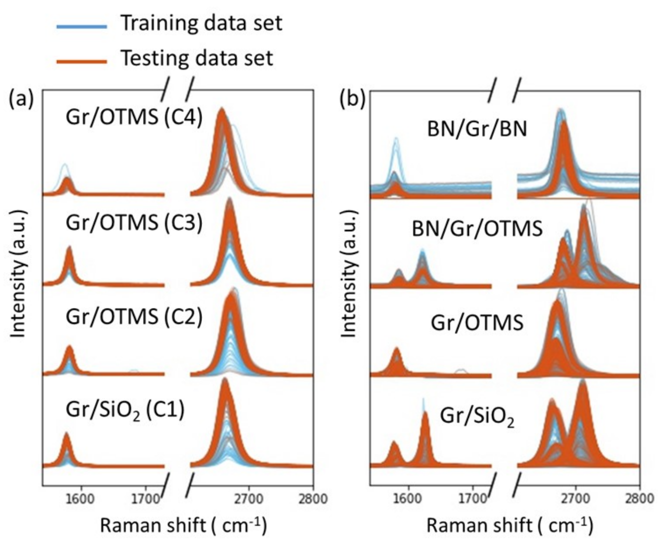

Figure S5 Raman spectra plots of different classes. (a) 4 different classes of Raman spectra in the graphene charge variation dataset (GCV dataset). (b) 4 different classes of Raman spectra in the graphene dielectric variation dataset (GCV dataset). The Raman spectra of each class are gathered from different graphene samples. The blue lines are the training set and the red lines are the testing set.

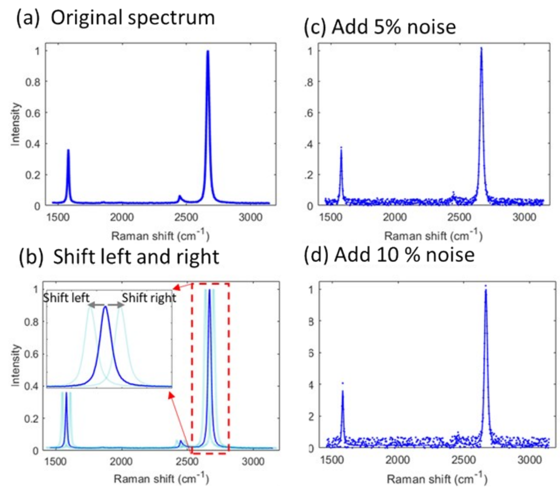

Figure S 6 Illustration of data augmentation. (a) A representative Raman spectrum of graphene. (b) Left shift and right shift of the Raman peak. (c-d) add 5% and 10 % noise level to the original spectrum. The noise level is defined as the percentage of the noise fluctuation to the maximum intensity of the 2D peak in the Raman spectrum. The noise is generated by a random function that uniformly generates a random number up to the specified noise level.

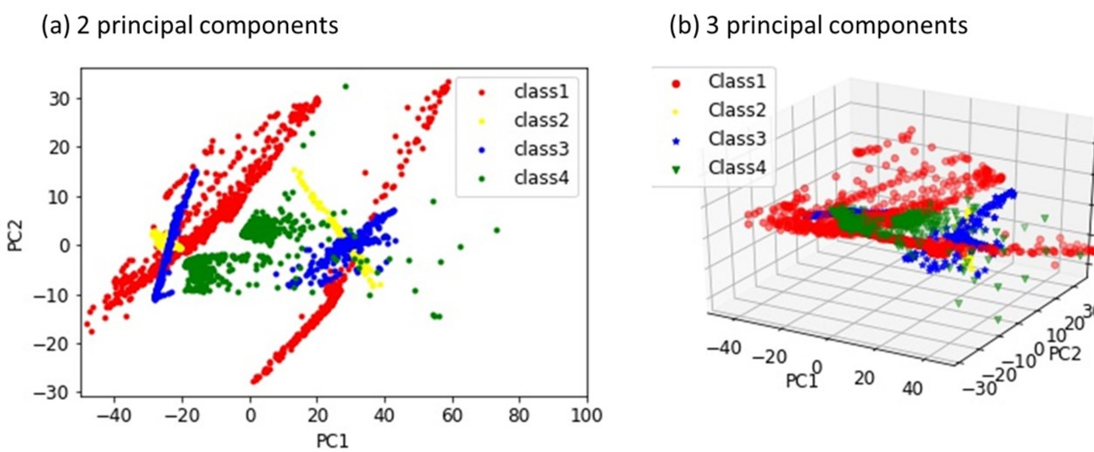

Figure S7 Principal component analysis of the Raman spectra data. (a) Two pricipal component. (b) Three principal component.

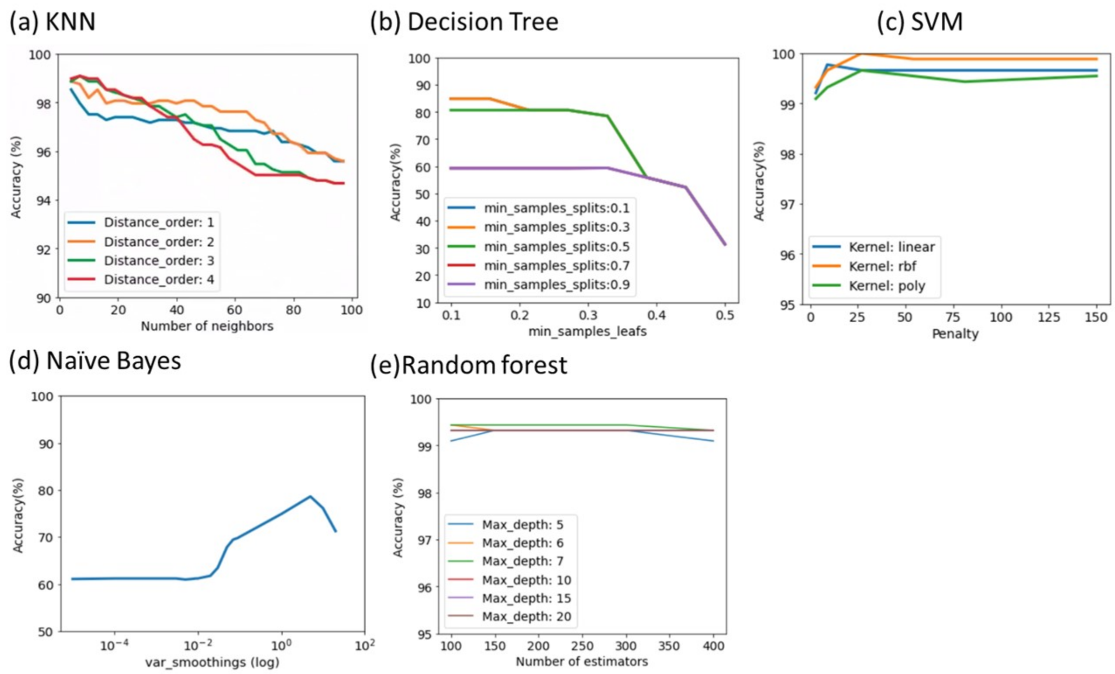

Figure S8 Machine learning algorithms hyperparameter tuning. (a) KNN, the number of neighbors and the distance order vs. classification accuracy. (b) Decision tree, the minimum leaf size and and minimum split samples vs. accuracy. (c) SVM, the penalty and Kernal type vs. accuracy. (d) Naïve Bayes, the distribution variance parameter vs. accuracy. (e) Random forest, the number of estimators and maximum tree depth vs. accuracy.

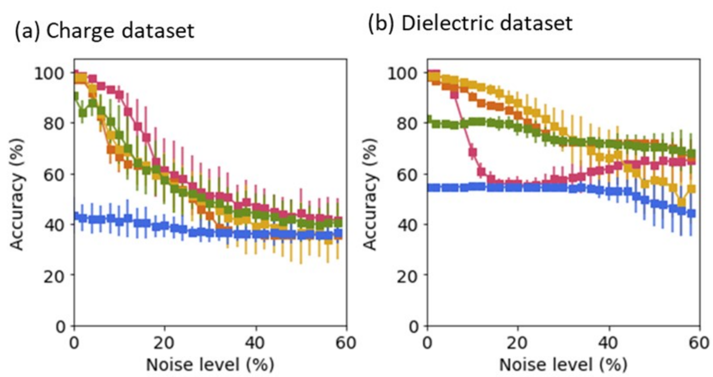

Figure S9 (a) Testing performance as a function of noise on the charge dataset. (b) Testing performance as a function of noise on the dielectric dataset. Note that except adding noise, peak shift of spectra was also applied to data augmentation.

Model Summaries

Throughout this project, we experimented with four different deep network architectures CNN, FullCNN, MultiHead CNN, and Fully Connected neural network, all of which are detailed below.

Fully Connected

This network is by far the simplest of the four architectures. It is 7 back-to-back linear layers. Each of the layers is followed by a relu activation and an appropriately sized batchnorm layer. The final output is then passed through a softmax function.

Table 2 Architecture summary of Fully connected neural network.

Fully Connected	
Layer Type	**Filter Size**
FC	1024
FC	512
FC	256
FC	128
FC	64
FC	16
FC	4

CNN

This model uses 5 one dimensional convolutions after each other. The output is then flattened and passed to 2 fully connected layers. The exact details of the layers can be seen in the figure below. It is important to note that each of the 1D Conv layers is followed by a relu activation, batchnorm of the appropriate size, and then a 1D average pooling of size 2. Similarly, the first fully connected layer is also followed by a relu activation and a batchnorm. Finally, the last fully connected layer is followed by a relu activation and a softmax.

Table 3 Architecture summary of CNN

CNN		
Layer Type	**Kernel Shape**	**Filter Count**
1D Conv	9x1	2
1D Conv	7x1	2
1D Conv	7x1	4
1D Conv	5x1	8
1D Conv	3x1	12
Flatten	-	-
FC	128	1
FC	4	1

FullCNN

This model is very closely related to the CNN model described above. It also starts with 5 1D convolution layers exactly as detailed before. However, after the fifth convolution, the output is then passed to a 1x1 convolution layer with a filter count of four. The output is this layer is then averaged across each filter yielding a single number for each filter, this is known as global average pooling. Once again, each convolution is followed by a relu activation, a batchnorm, and a 1d average pooling of size 2, and the last layer is also followed by a softmax function.

Table 4 Architecture summary of Full CNN.

Full CNN		
Layer Type	**Kernel Shape**	**Filter Count**
1D Conv	9x1	2
1D Conv	7x1	2
1D Conv	7x1	4
1D Conv	5x1	8
1D Conv	3x1	12
1D Conv	1x1	4
Pooling	1x19	1

Multihead CNN

This model is another variant of the CNN architecture. It passes the input into 3 different paths simultaneously, each of which containing its own, differently shaped, pair of 1d convolutional layers. All three outputs are then flattened and concatenated before being passed into the final fully connected layer. Each convolutional layer is followed by a relu activation, a batch normalization, and 1d max-pooling of size 2. The final layer is also followed by a softmax.

Table 5 Architecture summary of MultiHead CNN.

MultiHead CNN								
Layer	**Kernel Shape**	**Filter Count**	**Layer**	**Kernel Shape**	**Filter Count**	**Layer**	**Kernel Shape**	**Filter Count**
1D Conv	3x1	16	1D Conv	5x1	16	1D Conv	7x1	16
1D Conv	3x1	4	1D Conv	5x1	4	1D Conv	7x1	4
Flatten	-	-	Flatten	-	-	Flatten	-	-

Layer	Size
FC	4