

Machine learning number of Dirac cones with neural networks

JunAng Wang

August 9, 2023

Recent, machine learning has emerged as a powerful tool in various scientific disciplines. Particularly, some researchers have begun to utilize machine learning to solve many-body physics problems in the recent year [1–3], e.g. classifying phases[4–7], representing many-body wave function[8, 9], speeding up many-body simulations[10–13] et al. In our previous work titled "Topological Quantization of Superfluid Stiffness in Dirac Materials"[14], we demonstrated the mechanism of the quantization of the superfluid stiffness in Dirac materials. Our findings revealed that the superfluid stiffness in Dirac materials remains quantized as long as the number of Berry monopoles is conserved in the synthetic space. Specifically, in the case of charge-neutral and Zeeman field neutral conditions, the superfluid stiffness(D_{total}) is equal to $\frac{\Delta}{\pi}$ per cone (we will derive a more general D_{total} later). Therefore the quantity of superfluid stiffness is determined by the number of Dirac cones present. This raises a intriguing question: can we establish a mapping from the value of superfluid stiffness to the number of Dirac cones? Such a mapping would have practical implication allowing experimentalists to determine the number of Dirac cones by measuring the superfluid stiffness of real materials, like graphene.

In this note, we aim to address above question through a supervised machine learning approach. We train our neural network by feeding superfluid stiffness data obtained from the Dirac model that may encompass higher order Dirac cones, enabling it to predict the Hamiltonian's total absolute vorticity. The total absolute vorticity is equal to the number of Dirac cones times their corresponding order. After fully training our network, we test our model's performance using superfluid stiffness data for graphene which is a 2D material. Our model achieve a high accuracy, which validates the effectiveness of our method.

1 Dirac model and higher order band touching points

In this section, we will introduce the Dirac model and derive the corresponding superfluid stiffness. It's important to note that we utilize Dirac model's superfluid stiffness as input for our neural network during training process.

The Hamiltonian of the Dirac model in momentum space can be written as

$$\hat{h} = (ak_x + ibk_y)^n \hat{\sigma}_+ + (ak_x - ibk_y)^n \hat{\sigma}_- + \mu_0 \quad (1)$$

$$= |\tilde{k}|^n \cos(n\theta) \hat{\sigma}_x - |\tilde{k}|^n \sin(n\theta) \hat{\sigma}_y + \mu_0 \quad (2)$$

$$= |\tilde{k}|^n e^{in\theta \hat{\sigma}_z} \hat{\sigma}_x + \mu_0, \quad (3)$$

where μ is chemical potential, n is a positive integer, $\hat{\sigma}$ denote Pauli matrices in sub-lattice space, $|\tilde{k}| = \sqrt{(ak_x)^2 + (bk_y)^2}$ and $\tan(n\theta) = \frac{(ak_x - ibk_y)^n}{(ak_x + ibk_y)^n}$. The eigen-vector of the Hamiltonian is read as

$$|u_{\pm}\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm e^{-in\theta} \end{pmatrix}, \quad (4)$$

whose corresponding eigen-value is presented as $\epsilon_{\pm} = \pm |\tilde{k}|^n + \mu_0$.

Now let's consider that superconductivity is introduced into the system. The super current flow in the direction of gradient of superconducting phase ϕ proportional to superfluid stiffness, such as $J_i = D_{ij} \nabla_j \phi$, where superfluid stiffness D is a symmetric matrix

$$D = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{pmatrix} \quad (5)$$

The element of superfluid stiffness can be read as:

$$D_{ij} = 2\Delta^2 \iint \frac{dk_x}{2\pi} \frac{dk_y}{2\pi} \sum_{\alpha, \beta = \pm} \frac{\langle u_\alpha | \partial_i \hat{h} | u_\beta \rangle \langle u_\beta | \partial_j \hat{h} | u_\alpha \rangle}{E_\alpha^2 - E_\beta^2} \left(\frac{f(E_\beta)}{E_\beta} - \frac{f(E_\alpha)}{E_\alpha} \right), \quad (6)$$

where $i, j \in \{x, y\}$, $f(E) = \Theta(B + E) - \Theta(B - E) = \Theta(E - |B|)$ and $E_\pm = \sqrt{\epsilon_\pm^2 + \Delta^2}$. Utilizing Eq. (6), we immediately obtain the superfluid stiffness of the Dirac model,

$$D_{xx} = \frac{a}{b} (D_{\text{intra}} + D_{\text{inter}}), \quad (7)$$

$$D_{yy} = \frac{b}{a} (D_{\text{intra}} + D_{\text{inter}}), \quad (8)$$

$$D_{xy} = D_{yx} = 0, \quad (9)$$

where

$$D_{\text{intra}} = n \frac{\sqrt{\mu_0^2 + \Delta^2}}{2\pi} \left\{ \Theta(|\Delta| - |B|) + \Theta(|B| - |\Delta|) \Theta(\sqrt{\mu_0^2 + \Delta^2} - |B|) \left[1 - \frac{|\mu_0|}{\sqrt{\mu_0^2 + \Delta^2}} \frac{|B|}{\sqrt{B^2 - \Delta^2}} \right] \right\},$$

$$D_{\text{inter}} = \frac{n}{|\mu_0|} \frac{\Delta^2}{2\pi} \left\{ \Theta(|\Delta| - |B|) \ln \left(\frac{\sqrt{\mu_0^2 + \Delta^2} + |\mu_0|}{|\Delta|} \right) + \Theta(|B| - |\Delta|) \Theta(\sqrt{\mu_0^2 + \Delta^2} - |B|) \ln \left(\frac{\sqrt{\mu_0^2 + \Delta^2} + |\mu_0|}{\sqrt{B^2 - \Delta^2} + |B|} \right) \right\}.$$

when $\mu_0 \rightarrow 0$, $D_{\text{inter}} = n \frac{|\Delta|}{2\pi} \Theta(|\Delta| - |B|)$. For zero Zeeman field and chemical potential, $D_{xx} = D_{yy} = n \frac{|\Delta|}{\pi}$ which aligns in our initial statement.

In order to eliminate the effect of anisotropy, we are interested at the determinant of superfluid stiffness

$$|D| = D_{xx}D_{yy} - D_{xy}D_{yx} \quad (10)$$

$$= D_{\text{total}}^2, \quad (11)$$

which does not include a, b . Here $D_{\text{total}} = D_{\text{intra}} + D_{\text{inter}}$. By considering the determinant, we can effectively capture the overall behavior and characteristics of the superfluid stiffness, while cancels the influence of the inherent anisotropy present in the system. It is worth to notice that $|D| = D_{\text{total}}^2$ appears because of $D_{xy} = D_{yz} = 0$ in Dirac model, for more general case we need to calculate Eq. (10). For convenience, in the following discussion we set $\Delta = 1$. Figure. (1) illuminates the amplitude of $|D|$ with μ_0 and B varying in the range of $(-3, 3)$. Notably, one may observe that $|D|$ changes discontinuously at $|B| = \Delta$, specifically when $\mu_0 = 0$ $|D|$ experiences an abrupt drop $\frac{1}{\pi^2}$ (see inset figure). This discontinuity may serves as an crucial signal in identifying the so called total absolute vorticity.

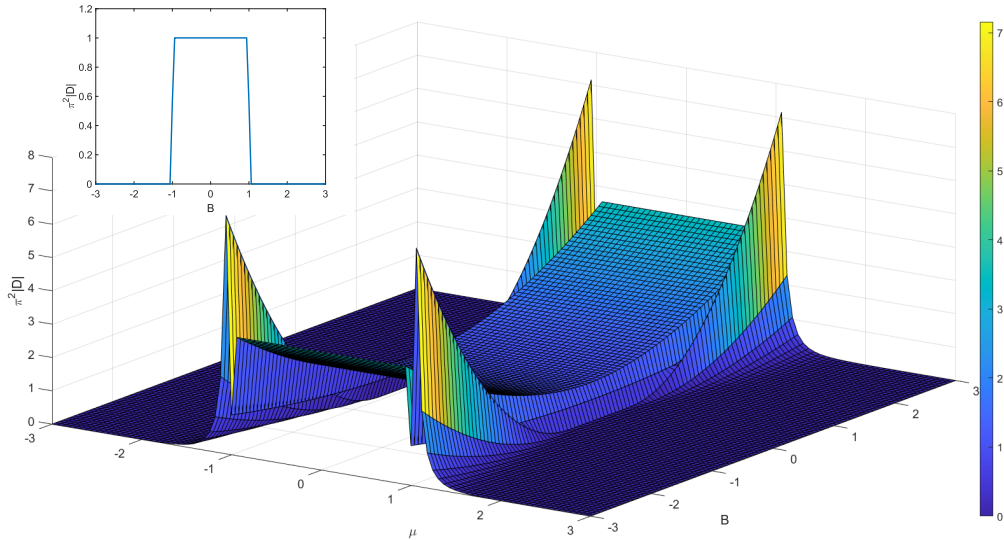


Figure 1: Amplitude of $|D|$ with μ_0 and B varying in the range of $(-3, 3)$. Inset figure shows variation of $|D|$ with B varying and $\mu_0 = 0$.

1.1 Total absolute vorticity

The total absolute vorticity is equal to the number of Dirac cones times their corresponding order, which is our neural network's main prediction. One can rewrite above Dirac model Hamiltonian as a vector form

$$\hat{h} = \mathbf{H} \cdot \boldsymbol{\sigma}, \quad (12)$$

where $\mathbf{H} = (|\tilde{k}|^n \cos(n\theta), |\tilde{k}|^n \sin(n\theta))$. The angle of vector \mathbf{H} is denoted by $\phi = \arctan(-\tan(n\theta)) = -n\theta$. The total absolute vorticity is written as

$$w = \frac{1}{2\pi} \left| \oint d\phi \right| = n. \quad (13)$$

2 Parabolic model

In previous section, we compute the contribution of a single Dirac cone to superfluid stiffness. In this section, we study the contribution of a trivial crossing point of the Fermi level to the superfluid stiffness. Without loss generality, we study the parabolic model whose Hamiltonian can be written as

$$\hat{h}_p = [(ak_x)^2 + (bk_y)^2] \hat{\sigma}_x + \mu_0 \quad (14)$$

whose eigen-vectors are

$$u_{\pm}^p = \begin{pmatrix} 1 \\ \pm 1 \end{pmatrix} \quad (15)$$

corresponds to eigen-value $\epsilon_{\pm}^p = \pm |\tilde{k}|^2 + \mu_0$. Applying Eq. (6), the contribution to the superfluid stiffness is

$$D_{xx}^p = \frac{a}{b} D_{\text{intra}}, \quad (16)$$

$$D_{yy}^p = \frac{b}{a} D_{\text{intra}}, \quad (17)$$

$$D_{xy}^p = D_{yx}^p = 0. \quad (18)$$

For trivial case, the interband part of superfluid stiffness is equal to zero.

Since the parabolic model's Hamiltonian only have contribution to $\hat{\sigma}_x$, then the corresponding vorticity is equal to zero.

3 Training data set

Before constructing the input data set, it's important to emphasize the primary objective of our neural network. Our neural network aims to predict two key aspects: firstly, whether the crossing point of the Fermi level corresponds to a Dirac cone. Secondly, the number of Dirac cones within a small window as the parameters B and μ_0 vary.

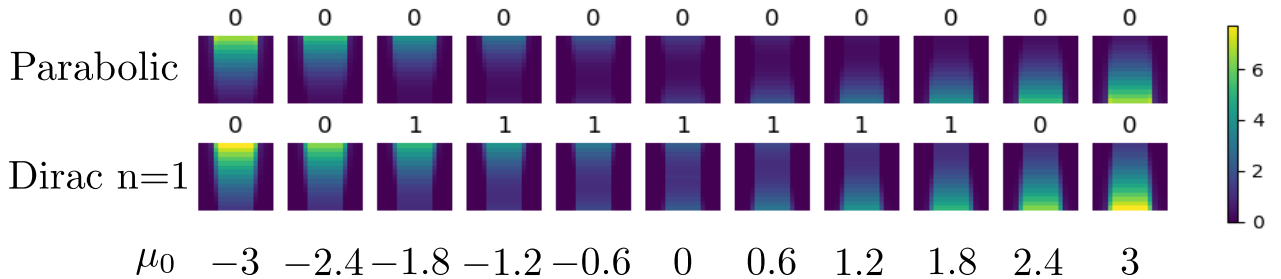


Figure 2: Training data set example. The first row illustrates the determinant of superfluid stiffness for a parabolic model with μ_0 varying in the range of $(-3, 3)$, $\mu_{\text{off}} = B_{\text{off}} = 2$. The second row represents the determinant for a Dirac model ($n = 1$) where μ_0 varies in the same range. The corresponding vorticity is shown above the matrix.

4 Neural network

In this note, we have chosen the convolutional neural network(CNN)[15–17] to classify the total absolute vorticity due to the crucial information concentrated along the $B = \Delta$ edge. The convolutional neural network is particularly effective at detecting and learning oriented edges, making it well suited for capturing the relevant features in our data. The neural network work flow and the structure is shown in Fig. (3).

The architecture of neural network structure is a simple and standard CNN classifier. One input the determinant of superfluid stiffness to the neural network which output the total absolute vorticity. The neural network encompass two convolutional layers with 32 channels of kernel size 3×3 and 2 channels of kernel size 3×3 , followed by two fully connected layers with the first one containing 1024 neurons and the second 512 neurons. At last, the neural network output the scores for 8 categories. The first 7 categories identify the total absolute vorticity from 0 to 6 and the last one match the total absolute vorticity above 6. By applying softmax function to normalize and convert the scores to possibilities. Consequently, the neural network provide the most probable total absolute vorticity as its output. All the hidden layers in the network are equipped with the rectified linear units(ReLU) $f(x) = \max\{0, x\}$ as activation function and the output layer utilizes linear activation $f(x) = x$. The learning rate and regularization strength is set to 10^{-3} and 0, respectively.

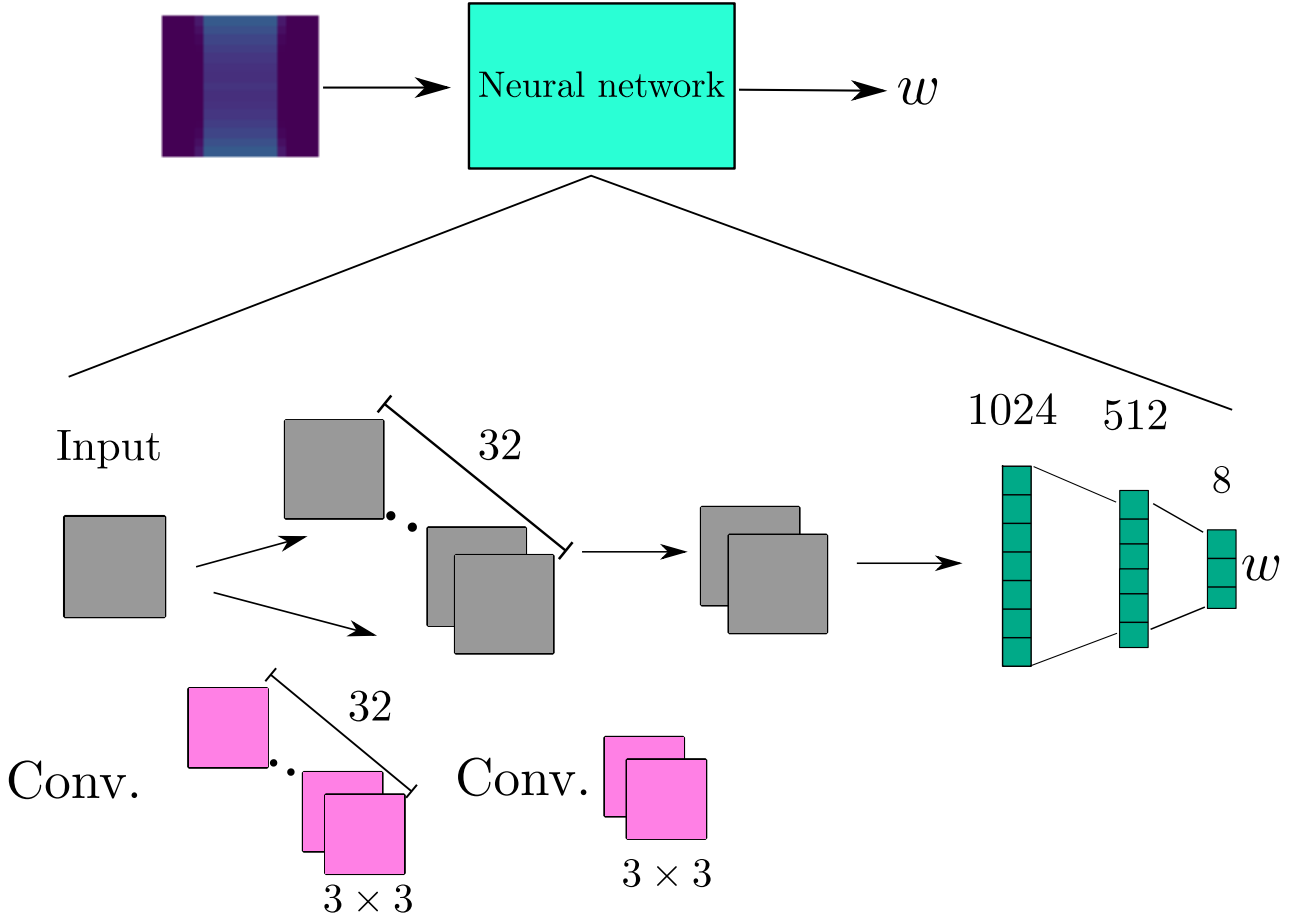


Figure 3: Schematic of the machine learning workflow and the structure of the convolutional neural network.

We generate Dirac model data set by using the approach discussed earlier. We split the Dirac and parabolic model data set into 2 part, 4×10^3 determinant samples with the total absolute vorticity $w \in (0, 7)$, randomly choose 90% of them for training data set and the remainder as (i) the first validation data set which was not used during training process; 10^3 samples with the total absolute vorticity $w \in (8, 9)$ (ii) as the second validation data set whose vorticity is unseen in training process. And set mu_0 ranges in $(-5, 5)$. During training, our neural network minimize the cross-entropy loss[18] to learn the patterns in the data by the adam optimizer[19].

After training, we evaluated the performance of our model by validation data set and achieved an accuracy of 99.60% for data (i) and 99.15% for data (ii). This high accuracy indicates that our neural network effectively identify and classify the total absolute vorticity of ideal Dirac model. In order to guarantee our method is suitable for real materials, in the following we evaluate our model by graphene data. The network construction and training were performed using PyTorch platform[20] .

5 Graphene model

In this note, we validate our neural network by testing a real material model, the graphene model. Graphene is made out of carbon atoms arranged in hexagon structure which can be viewed as two atoms per unit cell. Thus graphene possesses the so called sublattice symmetry (chiral symmetry). The Hamiltonian of Graphene in momentum space can be written as

$$\begin{aligned} \hat{h}_g = & -t \left[\cos(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y) + \cos(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y) + \cos(-k_x) \right] \hat{\sigma}_x \\ & - t \left[\sin(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y) + \sin(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y) + \sin(-k_x) \right] \hat{\sigma}_y + \mu, \end{aligned} \quad (19)$$

where $t \approx 2.8eV$ is hopping amplitude. $\hat{\sigma}$ are Pauli matrices acting on sub-lattice space.

The eigen-energy have formula $\epsilon_{\pm}^g = \pm t \sqrt{3 + 2 \cos(\sqrt{3}k_y) + 4 \cos(\frac{\sqrt{3}}{2}k_y) \cos(\frac{3}{2}k_x)} + \mu$ and its corresponding eigen-vector can be read as

$$u_{\pm}^g = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ \pm i e^{i\theta_g} \end{pmatrix}, \quad (20)$$

where

$$\tan(\theta_g) = \frac{\sin(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y) + \sin(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y) + \sin(-k_x)}{\cos(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y) + \cos(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y) + \cos(-k_x)} \quad (21)$$

Applying Eq. (6), one obtain the determinant of graphene's superfluid stiffness.

5.1 Total absolute vorticity of graphene

Since the hopping amplitude is much larger than the induced superconductivity in graphene $\Delta \approx 1meV$, the Hamiltonian can be approximately described by two Dirac cones

$$\hat{h}_g \approx v_F (q_x \hat{\sigma}_x + q_y \hat{\rho}_z \hat{\sigma}_y) + \mu_0, \quad (22)$$

where \mathbf{q} is the momentum measured relatively to the Dirac points, v_F is the Fermi velocity, given by $v_F = \frac{3}{2}t$, $\hat{\rho}$ are Pauli matrices in valley space.

The total absolute vorticity is the sum of absolute vorticity of two containing Dirac cones, i.e. $w_g \approx 2$. The determinant of graphene would be labeled as 2 if Dirac cones present within the defined window, else would be labeled as 0. By varying hopping amplitude and central chemical potential, one gather a set of test date.

5.2 The performance of neural network

(iii) We generate determinant samples from graphene with $t \in (3000, 4000)$ as test data set. Figure (4) illustrates the performance of network plot versus the number of training samples. The green and blue dots depict the accuracy of the first (i) and the second (ii) validation data set, respectively, reaching an accuracy 99.60% and 99.16% with training on 3500 samples. Meanwhile the red dots represent the accuracy achieved on the (iii) graphene data, which attains an accuracy of 95.00%. Notably, as the number of training data increases, the accuracies of all mentioned data improve indicating the beneficial impact of augmenting the training data size. The observation reflects the network's capability to enhance its predictive capability and as it learns from larger and more diverse training data. The test result is presented in Table (2).

The sketch (Figure 4(b)) shows the graphene spectrum in momentum space projecting on the k_y axis, where two Dirac points are located at the chemical potential $\mu_0 = 0$. The green and red boxes represent two examples of the previously discussed scanning windows concentrated at $\mu_0 = 0$ and $\mu_0 = 4.2$, ranging in $(\mu_0 - 2, \mu_0 + 2)$. The green box predicts the total absolute vorticity w to be equal to 2, while the red box predicts $w = 0$. The figure on the right-hand side demonstrates our network’s performance in 15 energy bins. Our results achieve high accuracy of around 100%, except for a drop to approximately 69% in the bins $1.67 < |\mu_0| < 2.33$. The reason for the accuracy drop is that the Dirac points are located around the edge of the scanning window, leading to ambiguity and harmful impact on predictions

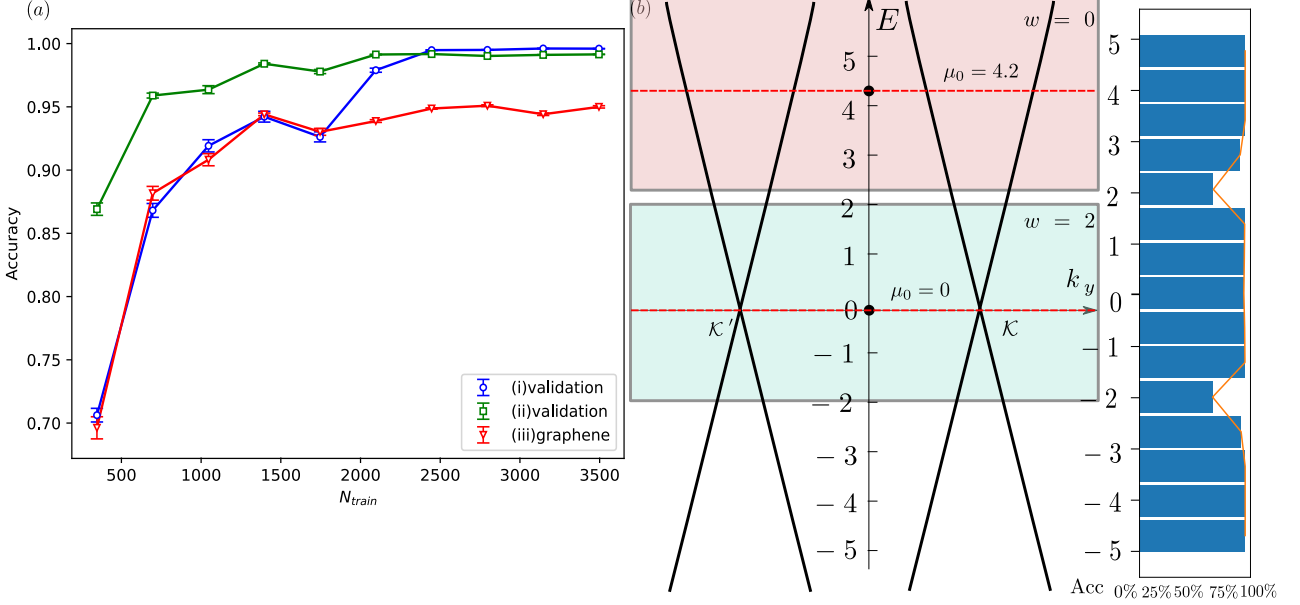


Figure 4: (a) The performance of the network versus the number of training samples. The green, blue and red dots represent the accuracy of the first (i) and the second (ii) validation data set and (iii) graphene data, respectively, reaching an accuracy 99.60%, 99.16% and 95.00% with training on 3.5×10^3 samples. One can observe that the accuracies of all data sets increase with more training data. (b) Sketch of graphene spectrum in momentum space, projecting on k_y axis, with two Dirac points locate at $\mu_0 = 0$. The green and red box depict two examples of previous discussed scanning windows concentrated at $\mu_0 = 0$ and $\mu_0 = 4.2$, range in $(\mu_0 - 2, \mu_0 + 2)$. The green box predicts the total absolute vorticity w equal to 2 while the red box predicts $w = 0$. The figure on the right hand side demonstrates the our network’s performance in 15 energy bins, achieving high accuracy around 100%, except for a drop to approximately 69% in the bins $1.67 < |\mu_0| < 2.33$. The accuracy drop is due to ambiguity caused by the proximity of the Dirac points to the edge of the scanning windows.

	Validation(i)	Validation(ii)	Test(iii)
Accuracy	99.60%	99.15%	95.00%

Table 1: Performance of the neural network training by $N_{train} = 3500$ with respect to different validation and test data sets.

6 Chiral symmetry broken model

In the real world, materials may or may not possess chiral symmetry. To demonstrate the effectiveness of our neural network in broader applications, we evaluate it by a chiral symmetry broken model

$$\begin{aligned} \hat{h}_g = & -t \left[\cos\left(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y\right) + \cos\left(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y\right) + \cos(-k_x) \right] \hat{\sigma}_x \\ & - t \left[\sin\left(\frac{1}{2}k_x + \frac{\sqrt{3}}{2}k_y\right) + \sin\left(\frac{1}{2}k_x - \frac{\sqrt{3}}{2}k_y\right) + \sin(-k_x) \right] \hat{\sigma}_y + \mu_0 + \text{Sign}(k_y)\delta\mu. \end{aligned} \quad (23)$$

This model involves the graphene Hamiltonian, which is modified by a chemical potential $\delta\mu$ depending on the sign of k_y in the first Brillouin zone. As a result, the two Dirac cones in the model are not longer aligned at the same energy level but shifted by $\delta\mu$, the low energy effective model can be expressed as

$$\hat{h}_g = v_F (q_x \hat{\sigma}_x + q_y \hat{\rho}_z \hat{\sigma}_y) + \delta\mu \hat{\rho}_z + \mu_0. \quad (24)$$

6.1 Train the network

We utilize the previously discussed data set and compute the determinant superfluid stiffness of a Dirac model containing two Dirac points permuting in the full spectrum ($|D_1 + D_2|$). For better comparison, we set the maximum vorticity of every individual Dirac point to $\max(|D_1|) = \max(|D_2|) = 9$ and the maximum of total vorticity of the full system is also set to $\max(|D_1 + D_2|) = 9$. This approach allows us to generate a new data set with $\mu_0 \in (-5, 5)$, $\delta\mu \in (-5, 5)$, $w \in (0, 9)$ containing around a total of 5×10^4 samples. We split the data set into 2 part: the first part include 4.5×10^4 samples with $w \in (0, 7)$ and the second part contains 0.5×10^4 samples with $w \in (8, 9)$ forming the unseen w validation data set (v). Moreover, 90% of the first part samples used as new training data set and the remainder forming (iv) the validation data set which remained unseen during training process. Additionally, we generate the determinant $|D|$ for modified graphene model Eq. (23) as test data set (vi). Sketch (5(b)) depicts the modified graphene low energy effective spectrum, similarly to sketch (4(b)) the system containing two Dirac cones however the cones shifted by chemical potential $\delta\mu$. Three examples of scanning windows are shown in the sketch, unlike only predicting $w = 0$ and $w = 2$ in sketch (4(b)), sketch (5) predict cases $w = 1$ due to nonzero $\delta\mu$, seen in the blue box where $\mu_0 = -2.2$. Additionally, the accuracy density plot illuminates the accuracy remain relative high expect the ambiguous situation discussed earlier.

After training our network on the training data, we validate the performance of our network by using data (iv), (v) and (vi). We achieve an accuracy of 98.26% for (iv), 98.18% for (v) and 90.54% for (vi). The high accuracy demonstrate the effectiveness of our method for chiral symmetry broken material, further proving the versatility and robustness of our approach. Figure (5(a)) depicts the performance of the network to chiral symmetric broken two-Dirac-points systems versus the training scale. The accuracies improve as the training scale increasing.

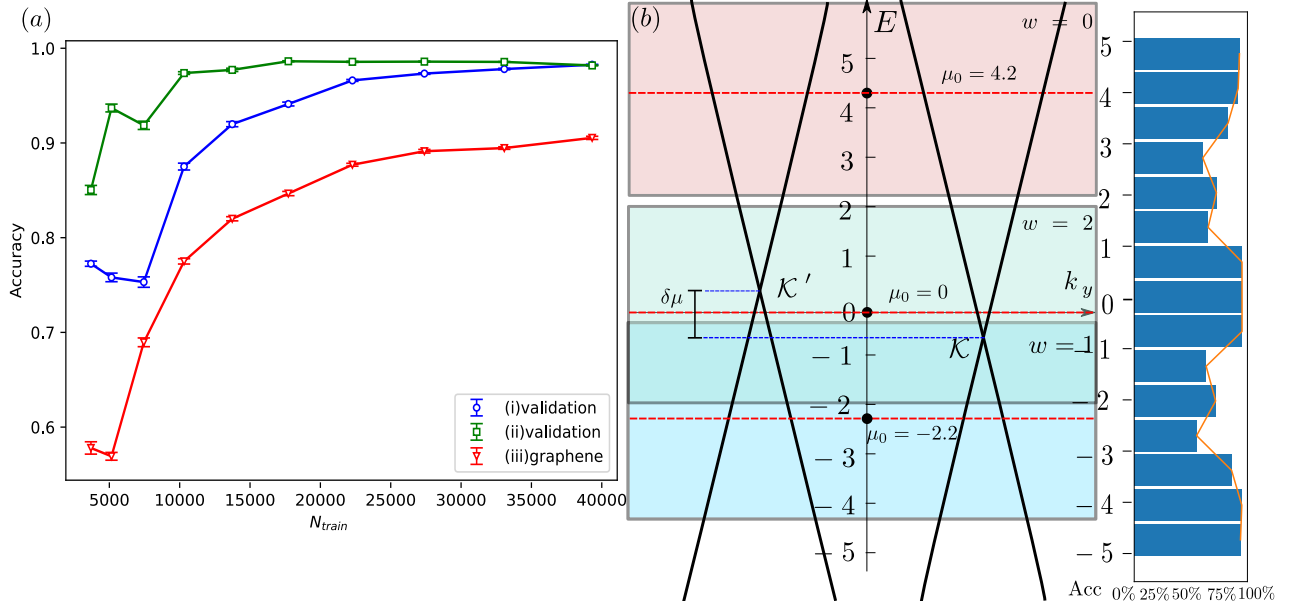


Figure 5: (a) The performance of the network to chiral symmetric broken systems versus the number of training samples. The green, blue and red dots represent the accuracy of the first (iv) and the second (v) validation data set and (vi) modified graphene data, respectively, reaching an accuracy 98.26%, 98.18% and 90.54% with training on 4×10^4 samples. The accuracies of all data sets increase with more training data. (b) Sketch of graphene spectrum in momentum space, projecting on k_y axis, with two Dirac points with $\mu_0 = 0$, $\delta\mu = 1.2$. The green, red and blue boxes depict two examples of previous discussed scanning windows concentrated at $\mu_0 = 0$, $\mu_0 = 4.2$ and $\mu_0 = -2.2$, range in $(\mu_0 - 2, \mu_0 + 2)$. The green box predicts the total absolute vorticity w equal to 2 while the red and blue box predicts $w = 0$ and $w = 1$, respectively. The figure on the right hand side demonstrates the our network's performance in 15 energy bins, achieving high accuracy around 100%, except for a drop in the bins $1 < |\mu_0| < 3$. The accuracy drop is due to ambiguity caused by the proximity of the Dirac points to the edge of the scanning windows.

	Validation(iv)	Validation(v)	Test(vi)
Accuracy	98.26%	98.18%	90.54%

Table 2: Performance of the neural network training by $N_{train} = 4 \times 10^4$ with respect to different validation and test data sets.

References

- [1] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.
- [2] Juan Carrasquilla. Machine learning for quantum matter. *Advances in Physics: X*, 5(1):1797528, 2020.
- [3] Steven Johnston, Ehsan Khatami, and Richard Scalettar. A perspective on machine learning and data science for strongly correlated electron problems. *Carbon Trends*, 9:100231, 2022.
- [4] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.
- [5] Lei Wang. Discovering phase transitions with unsupervised learning. *Physical Review B*, 94(19):195105, 2016.

- [6] Akinori Tanaka and Akio Tomiya. Detection of phase transition via convolutional neural networks. *Journal of the Physical Society of Japan*, 86(6):063001, 2017.
- [7] Pengfei Zhang, Huitao Shen, and Hui Zhai. Machine learning topological invariants with neural networks. *Physical review letters*, 120(6):066401, 2018.
- [8] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.
- [9] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Physical review letters*, 124(2):020503, 2020.
- [10] Chuang Chen, Xiao Yan Xu, Junwei Liu, George Batrouni, Richard Scalettar, and Zi Yang Meng. Symmetry-enforced self-learning monte carlo method applied to the holstein model. *Physical Review B*, 98(4):041102, 2018.
- [11] Dian Wu, Lei Wang, and Pan Zhang. Solving statistical mechanics using variational autoregressive networks. *Physical review letters*, 122(8):080602, 2019.
- [12] Yuki Nagai, Huitao Shen, Yang Qi, Junwei Liu, and Liang Fu. Self-learning monte carlo method: Continuous-time algorithm. *Physical Review B*, 96(16):161102, 2017.
- [13] Junwei Liu, Huitao Shen, Yang Qi, Zi Yang Meng, and Liang Fu. Self-learning monte carlo method and cumulative update in fermion systems. *Physical Review B*, 95(24):241104, 2017.
- [14] Jun-Ang Wang, Mohamed Assili, and Panagiotis Kotetes. Topological quantization of superfluid stiffness in dirac materials. *arXiv preprint arXiv:2211.07676*, 2022.
- [15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [18] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The” wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.