Machine Learning of Chiral Symmetric Topological Insulator with Transformer-based Geometric Tomography

Ziyu Wang(王子雨)[1,2,*], Zeming Fang(方泽明)[1,2,*] and Ying Hu(胡颖)[1,2,†]

[1]State Key Laboratory of Quantum Optics Technologies and Devices,
Institute of Laser Spectroscopy, Shanxi University, Taiyuan 030006, China

[2]Collaborative Innovation Center of Extreme Optics, Shanxi University, Taiyuan 030006, China

2026 年 8 月 13 日

Abstract

Machine learning is emerging as an important tool in the study of topological states of matter, whose existence is intrinsically linked to the geometric structure of quantum states. While many machine learning methods can efficiently predict topological invariants of band insulators, they often act like "black boxes", failing to simultaneously reveal the underlying geometric features of the band structure. In this work, we design a neural network based on the Transformer architecture to characterize the topological properties of one-dimensional chiral symmetric topological insulators in the AIII symmetry class. After suitable training, the network not only accurately predicts topological invariants, but its attention map also autonomously highlights band regions with the maximal phase gradient as being topologically relevant. Remarkably, the network achieves this simultaneous identification even for the Hamiltonians whose topological numbers are not included in the training set. Moreover, we investigate how the number of sampling points and the choice of loss function affect the network training efficiency. This work provides an efficient and partially interpretable approach for characterizing topological states of matter.

Keywords: neural networks; topological invariant; attention mechanism

PACS: 07.05.Mh, 03.65.Vf

1. Introduction

In recent years, machine learning, particularly deep neural networks, has emerged as a powerful data-driven approach in artificial intelligence, demonstrating immense potential across many fields [1–6]. With the advancement of efficient training algorithms [7–11], neural networks have become indispensable, finding widespread applications in fields ranging from physical chemistry [12], medical imaging [13], structural defect detection [14] and accelerating Monte Carlo algorithms [15]. In particular, the ability of neural networks

*These authors contributed equally to this work.

†Corresponding author. E-mail:huying@sxu.edu.cn

to extract physical laws directly from raw data without relying on explicit analytical solutions offers promising prospects in areas such as optical device design [16,17], entanglement detection [18,19], quantum control [20,21], quantum simulation [22], and phase classification [23,24].

The exploration of topological states of matter represents an important forefront in modern physics [25,26]. Recently, many machine learning approaches have been developed that can identify topological phases with high accuracy. These include supervised learning methods that learn the nonlocal topological invariants from local data [27–29], and unsupervised learning methods that classify topological phases without data labels [30–33]. Moreover, numerous studies have extracted topological invariants directly from experimental data [34–36]. However, these networks often behave as "black boxes", and it remains a challenge to simultaneously probe the nonlocal topological invariants and the relevant local geometric features.

Recently, an approach based on the Transformer [7] has been demonstrated for characterizing the braiding topology of non-Hermitian bands [37], where the self-attention mechanism has been leveraged to diagnose the band-crossing points as the key spectral regions relevant to the braiding topology. This work convincingly shows that the self-attention mechanism endows the Transformer a two-fold capability that makes it particularly suitable for addressing the challenge of detecting the topology-geometry interplay. On the one hand, since the self-attention mechanism directly establishes correlations among all momentum-space tokens, it is well suited to learn the topological invariant which is intrinsically nonlocal. On the other hand, although the network is trained solely for high-confidence classification of topological phases (i.e., no local geometric quantity is used as an additional label or included in the loss function), the attention maps autonomously identify geometrical features important for the topological classification. Such a dual-capability is a key advantage of the Transformer over other neural networks in characterizing the topological phases.

Building on this study, we design a Transformer-based neural network (TNN) to explore the interplay between topology and geometry in one-dimensional topological insulators with chiral symmetry, which belong to the AIII symmetry class and are characterized by the topological winding number. Using a large number of randomly generated Hamiltonians, we train the neural network, optimizing it solely for predicting topological number with high accuracy. The trained network not only identifies topological invariants efficiently, but its attention map explicitly and autonomously identify regions of the Brillouin zone (BZ) exhibiting minimal energy gaps as being crucially relevant to the topology. Remarkably, this is achieved even for Hamiltonians whose winding numbers are larger than the training set. This dual predictive capability distinguishes our study from prior work [27–29,34–36] using, e.g., convolutional neural networks (CNNs) and fully-connected neural networks (FNNs). In addition, we introduced structural modifications to investigate how the number of sampling points and the choice of loss function affect the training. We show that increasing the number of sampling points does not indefinitely improve predictive capability; instead, the performance begins to decline after reaching a certain threshold due to overfitting. We also compared the effect of the Mean Squared Error (MSE) and Huber loss functions on the network's performance. Our results indicate that using MSE loss function yields higher confidence, whereas Huber

loss function offers greater stability. This findings deepened our understanding of neural network behavior, and pave the way for developing more generalizable and efficient networks for characterizing topological phases.

This paper is organized as follows. In Section 2, we present the physical model used in this study, along with the methodology for generating the datasets for training and testing the neural network, as well as the corresponding dataset structure. We also detail the architecture of TNN we constructed and describe the method used to explore the attention distribution of the network. Section 3 reports the training results, including learning curves, test results, and the attention weight distributions in the Brillouin zone (BZ), alongside our preliminary findings on network optimization. Finally, Section 4 provides a comprehensive conclusion summarizing the main achievements of this research.

2. Input dataset and network architecture

In this section, we introduce our data composition and network architecture in two parts. Section 2.1 details the physical model used for data generation, as well as the composition of the training, validation and test sets. Section 2.2 describes the architecture of the TNN we constructed and elucidates its core self-attention mechanism.

2.1. Topological invariant and data generation

We consider a topological band insulator in AIII symmetry class [25,38–40], characterized by a Hamiltonian $\boldsymbol{H}(k) = \boldsymbol{h}(k) \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} = (\boldsymbol{\sigma}_x, \boldsymbol{\sigma}_y, \boldsymbol{\sigma}_z)$ is the vector of Pauli matrices, and k denotes the momentum. The chiral symmetry of the AIII symmetry class implies that the Hamiltonian satisfies $\boldsymbol{S}\boldsymbol{H}(k)\boldsymbol{S}^{-1} = -\boldsymbol{H}(k)$, where $\boldsymbol{S} = \boldsymbol{\sigma}_z$ is the matrix form of the chiral symmetry operator.

Before feeding the data into the neural network, the system's Hamiltonian is preprocessed following the method described in Ref. [28]. We deform the Hamiltonian as $\tilde{\boldsymbol{H}}(k) = \tilde{h}_x(k)\boldsymbol{\sigma}_x + \tilde{h}_y(k)\boldsymbol{\sigma}_y$ and uniformly pick L points in the BZ. Here $\tilde{h}_i(k) = h_i(k)/|\boldsymbol{h}(k)|$, $i = x, y$. This results in a matrix of size $(L+1) \times 2$, which takes the form:

$$\begin{pmatrix} \tilde{h}_x(0) & \tilde{h}_x(2\pi/L) & \tilde{h}_x(4\pi/L) & \cdots & \tilde{h}_x(2\pi) \\ \tilde{h}_y(0) & \tilde{h}_y(2\pi/L) & \tilde{h}_y(4\pi/L) & \cdots & \tilde{h}_y(2\pi) \end{pmatrix}^T, \tag{1}$$

with $L = 128$ in this manuscript.

For the topological band insulators of the AIII symmetry class, the topological invariant is the winding number

$$w = -\frac{i}{2\pi} \oint_0^{2\pi} U^*(k)\partial_k U(k)\mathrm{d}k, \tag{2}$$

where $U(k) = \tilde{h}_x(k) + i\tilde{h}_y(k)$ for the Hamiltonian given above. Here, the winding number w is an integer. Its magnitude represents the number of times $U(k)$ winds around the origin as k varies from 0 to 2π,

while its sign indicates the winding direction, with $w > 0$ corresponding to counter-clockwise winding and $w < 0$ to clockwise winding. For the discrete $U(k)$, the winding number can be expressed as

$$w = \frac{1}{2\pi} \sum_{n=1}^{L} \Delta\theta(n), \tag{3}$$

where $\Delta\theta(n) = \theta(n) - \theta(n-1)$ is the phase difference and $\theta(n) = -i \ln[U(2\pi n/L)]$ is the phase.

To enable the TNN to learn the winding number calculation as shown in Eq. (3), a training set was generated for the learning process. Meanwhile, a validation set was generated to monitor and prevent overfitting during learning, and a test set was generated to evaluate the performance. Considering that using simple physical systems leads to poor generalization ability as observed in Ref. [28], we generate the training data using a general one-dimensional Hamiltonian with chiral symmetry, given by

$$\boldsymbol{H}(k) = h_x(k)\boldsymbol{\sigma}_x + h_y(k)\boldsymbol{\sigma}_y, \tag{4}$$

where

$$h_i(k) = \sum_{n=0}^{c} [a_i^{(n)} \cos(nk) + b_i^{(n)} \sin(nk)], \tag{5}$$

$i = x, y$, and $c = 4$ is a cutoff, meaning that the maximum possible winding number of the Hamiltonian is 4. $a_i^{(n)}$ and $b_i^{(n)}$ are random coefficients uniformly distributed in $[-1, 1]$. Subsequently, we can generate inputs in the form of Eq. (1) and use the corresponding winding numbers calculated via Eq. (3) to serve as the labels for these inputs.

The training set was generated with 10^6 Hamiltonians, comprising 37% with the winding number of $w = 0$, 50% with $w = \pm 1$, and 13% with $w = \pm 2$. The validation set consists of 10^4 Hamiltonians with $w = 0, \pm 1, \pm 2$ that are distinct from the training data. For the test set, four different datasets were generated to evaluate the network's performance on various types of input data. Test I comprises 10^4 Hamiltonians with $w = 0, \pm 1, \pm 2$ that are not included in the training or validation sets. Test II and Test III consist of 10^4 Hamiltonians with winding numbers $w = \pm 3$ and $w = \pm 4$, respectively. The data for Test IV consist of 10^4 Hamiltonians generated by the following form:

$$\boldsymbol{H}_{\mathrm{SSH}}(k) = (t_1 + t_2 \cos k)\boldsymbol{\sigma}_x + (t_2 \sin k)\boldsymbol{\sigma}_y. \tag{6}$$

This Hamiltonian corresponds to the Su-Schrieffer-Heeger (SSH) model [41], which is the simplest and most accessible model belonging to the AIII symmetry class. Here, t_1 and t_2 are random numbers uniformly distributed within $[-10, 10]$. In the SSH model, only two winding numbers are supported, where $w = 0$ when $t_1 > t_2$ and $w = 1$ when $t_1 < t_2$.

2.2. Architecture of Transformer-based neural network

After generating the datasets described in Section 2.1, the input data from the training set are fed into the TNN shown in Fig. 1(a). The preprocessed input given by Eq. (1) is first processed by an embedding layer, which expands the feature dimensionality of each point in the BZ from 2 to 128. A learnable

128-dimensional classification (CLS) token is prepended to the $\boldsymbol{H}(0)$ token to capture global topological features and trained jointly with the embedded inputs. Subsequently, the concatenated data undergoes positional encoding and is fed into a Transformer encoder as shown in Fig. 1(b). Within the encoder, the input data learns the relationships between points via a multi-head attention layer (the number of heads is 8 in this manuscript). The data is then passed to a feed forward layer with 512 neurons through residual connections and layer normalization to further refine the features. The refined features are output after passing through residual connections and layer normalization once more. Finally, the CLS token, which has learned the global topological features, is fed into a hidden layer with 64 neurons, ultimately outputting the winding number identified by the network. The hidden layer uses the Gaussian Error Linear Unit as the activation function in the TNN.

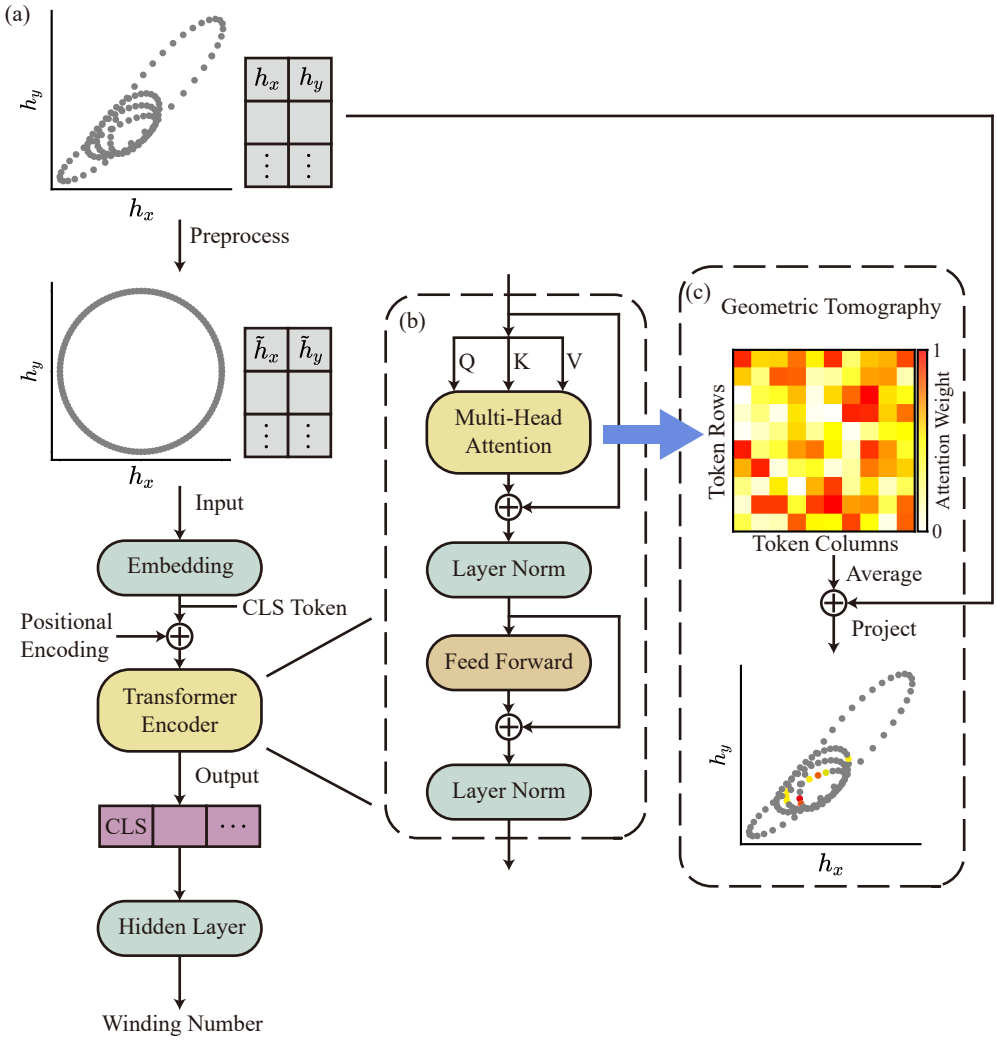


Fig. 1. Schematic of TNN and the self-attention mechanism. (a) TNN architecture. The preprocessed input is expanded through a linear embedding layer, concatenated with a CLS token, positionally encoded, and processed by an encoder block. The CLS token, which has learned the global features, is fed into a hidden layer with 64 neurons to predict the corresponding winding number. (b) The structure of the Transformer encoder. Inputs of the encoder are transformed into query($\boldsymbol{Q}$), key($\boldsymbol{K}$), and value($\boldsymbol{V}$) matrices for multi-head attention computation. The output of the multi-head attention module subsequently undergoes a residual connection and layer normalization, followed by a feed-forward network, and finally another residual connection with layer normalization before propagation to subsequent layers. (c) Schematic of the working principle of the geometric tomography. By leveraging the unique self-attention mechanism of the Transformer, the network generates attention weight matrices. After averaging the attention weights of the CLS token over other tokens across all heads, the matrix is projected onto the input data to visualize the network's focus on different regions.

In the Transformer encoder shown in Fig. 1(b), the output of the self-attention mechanism is given by [7]

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_k}})\boldsymbol{V} \equiv \boldsymbol{\Gamma}\boldsymbol{V} \tag{7}$$

where d_k denotes the feature dimension, and $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are the query, key, and value matrices obtained by linearly transforming the input with weight matrices $\boldsymbol{W}_Q$, $\boldsymbol{W}_K$, and $\boldsymbol{W}_V$, respectively. In Eq. (7), the dot product of $\boldsymbol{Q}$ and $\boldsymbol{K}$, after scaling and the softmax operation, yields the attention weight matrix $\boldsymbol{\Gamma}$, which is then used to compute a weighted sum of the value matrix. To enhance feature extraction, matrices $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ are projected into multiple subspaces where each head operates independently, and the results are finally concatenated to form the output.

Unlike traditional neural networks such as CNNs and FNNs, the attention weights matrix $\boldsymbol{\Gamma}$ encodes the learned correlations between all pairs of tokens corresponding to different k-points of a given Hamiltonian. The elements of $\boldsymbol{\Gamma}$ quantify the degree of correlation between $\boldsymbol{H}(k_\alpha)$ and $\boldsymbol{H}(k_\beta)$ (here α and β are integers), thereby enabling their geometric visualization and physical interpretation. As shown in Fig. 1(c), by averaging the attention weights of the CLS token over other tokens across all heads, we obtain an attention score vector with dimension $L+1$, highlighting the key input tokens essential for learning the winding number.

3. Results

In this section, we use the training set generated in Section 2.1 to train the TNN to learn the topological features. In Section 3.1, we present the learning curves and evaluate the network's performance using the test sets generated in Section 2.1. In Section 3.2, we analyze the learning process by extracting attention weights from the multi-head attention layers and visualizing their distribution. Finally, in Section 3.3, we investigate the impact of the number of sampling points and the choice of loss function on the network training process.

3.1. Learning curve

We feed the input data from the generated training set into the TNN, adding Gaussian noise with a standard deviation of 0.05 to prevent overfitting. The network compares its predicted values with the

ground truth and performs backpropagation, updating the parameters of each module via gradient descent to learn the data features. The learning rate is linearly increased to 10^{-3} during the first five epochs, followed by a cosine annealing schedule down to 10^{-8}. Here, we employ the MSE loss function and the AdamW optimizer, without introducing any other regularization. To monitor and prevent overfitting during learning, we introduce validation data every 10 epochs to evaluate the network's generalization ability on unseen data. The validation data is not involved in the parameter update process.

The learning curves after 100 epochs are shown in Fig. 2. Fig. 2(a) displays the loss curves for both the training and validation sets, illustrating the evolution of the loss function during the learning process. Fig. 2(b) presents the prediction accuracy curves for both datasets, where accuracy is defined as the percentage of predictions that match the ground truth after rounding. After 100 epochs, the training loss value and accuracy reach 0.0441 and 94.98%, respectively, while the validation set achieves 0.0149 and 98.7%. Due to the added Gaussian noise for regularization, the training accuracy is slightly lower than the validation accuracy, a common phenomenon in noisy training scenarios.

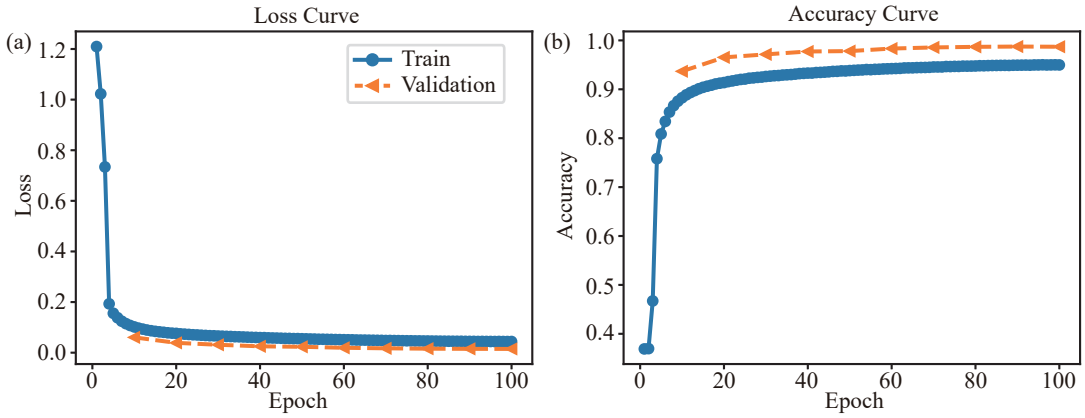


Fig. 2. The network (a) loss and (b) accuracy curves. The blue solid line represents the training loss(accuracy) curve, while the orange dashed line represents the validation loss(accuracy) curve.

To compare the performance of TNN with other neural networks in learning topological invariants, we additionally trained a CNN and an FNN using the same training dataset and identical training procedures. For the CNN, a convolutional layer with a 5×5 kernel replaces the multi-head attention layer of the TNN. Additionally, a pooling layer is added after the feed-forward layer to ensure its output dimension matches that of the TNN encoder module, while the rest of the architecture remains consistent with the TNN. The FNN consists of three hidden layers with 128, 64, and 32 neurons, respectively.

We then fed the input data from the four test sets into the trained TNN, CNN and FNN. The results are presented in Table 1. The results indicate that for Test I, which has a composition similar to the training set, the TNN successfully identifies the topological features and achieves predictive performance comparable to the CNN. The TNN and CNN exhibit robust performance on Test II and III datasets, demonstrating that the TNN and CNN maintain excellent discriminative power even when faced with topological structures absent from the training set. In stark contrast, the FNN's accuracy collapses to 30.34% and 6.62%, highlighting its inability to generalize beyond the training distribution. Meanwhile, these networks all achieve high accuracy on Test IV that consists of Hamiltonians generated by the SSH

model.

Table 1. Accuracy of the neural networks in the test set.

Network	Test I	Test II	Test III	Test IV
TNN	98.56%	99.10%	97.40%	100%
CNN	99.84%	99.26%	98.84%	100%
FNN	94.27%	30.34%	6.62%	100%

3.2. Heat map and attention weight

To investigate the focus of the TNN during the learning process, we extracted the attention weight matrix $\boldsymbol{\Gamma}$ from Eq. (7) and then obtained the attention score vector. Subsequently, as shown in Figs. 3(a) and (b), we normalized this vector and projected it into the locus of (h_x, h_y) as k runs over the BZ, generating a heatmap of the attention weight distribution that allows us to observe the network's attention distribution over the input data.

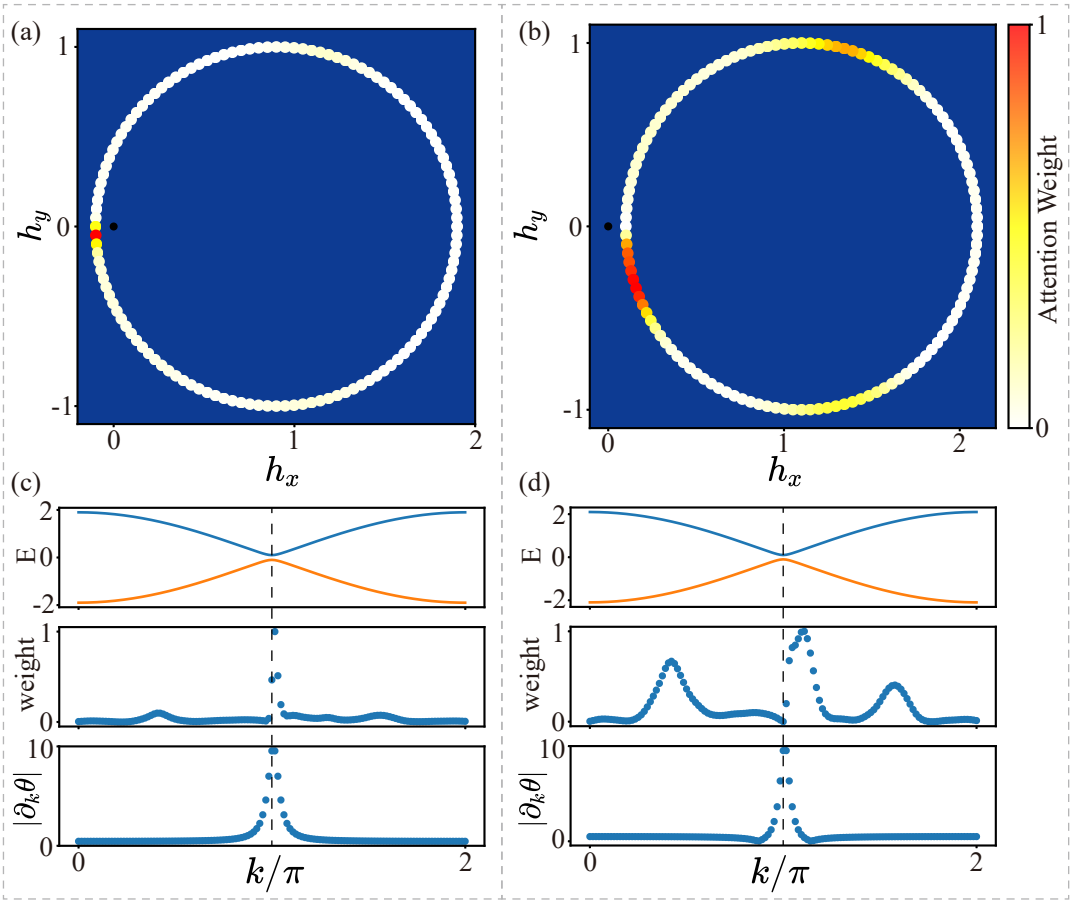


Fig. 3. Attention-based geometric tomography of Hamiltonian of the SSH system. (a, b) Distribution of attention weights shown in the locus of (h_x, h_y) as k runs over the Brillouin zone, where (a) $t_1 = 0.9$ for $w = 1$ and (b) $t_1 = 1.1$ for $w = 0$. Here the black point is the origin of coordinates. (c, d) Corresponding energy band, distribution of attention weights in the BZ and theoretically calculated distribution of the phase gradient $|\partial_k\theta|$ in momentum space. Here, the other parameter of the Hamiltonian is $t_2 = 1$.

Subsequently, we systematically generated a large number of Hamiltonians of the SSH system using Eq. (6) and obtained their corresponding attention heatmaps. For these Hamiltonians, we set $t_1 \in [-2, 2]$ and $t_2 = 1$. We selected the datasets for $t_1 = 0.9$ and $t_1 = 1.1$ to plot the attention heatmaps shown in Figs.

3(a) and (b). These two datasets have winding numbers of 1 and 0, respectively. From Figs. 3(a) and (b), we observe that the network's attention weights are concentrated in the vicinity of $h_y = 0$ (indicated by red dots). Figs. 3(c) and (d) present the corresponding energy band, distribution of attention weights in the BZ and theoretically calculated distribution of $|\partial_k \theta(n)|$ in momentum space. Here, $|\partial_k \theta(n)| = |\Delta\theta(n)/(2\pi/L)|$ represents the phase gradient of discrete $U(k)$. From Figs. 3(c) and (d), we find that the maximal attention weights appear in the vicinity of the minimal energy gaps and the maximal phase gradient, which coincide with the regions contributing most significantly to the topological invariants. This indicates that the trained TNN can simultaneously identify both the topological invariants and the maximal phase gradient as the underlying geometric origins.

To extend this finding to more complex systems, we use Eq. (4) to generate some general one-dimensional Hamiltonians. Here, we select the datasets for $a_x^{(4)} = -0.23$ and $a_x^{(4)} = 1.37$ to plot the attention heatmaps shown in Figs. 4(a) and (b), respectively, while keeping $b_y^{(4)} = 1$ and all other parameters set to 0.5. These two datasets have winding numbers of -2 and 4, respectively. The corresponding energy bands, distribution of attention weights and theoretically calculated distribution of $|\partial_k \theta|$ are shown in Fig. 4.

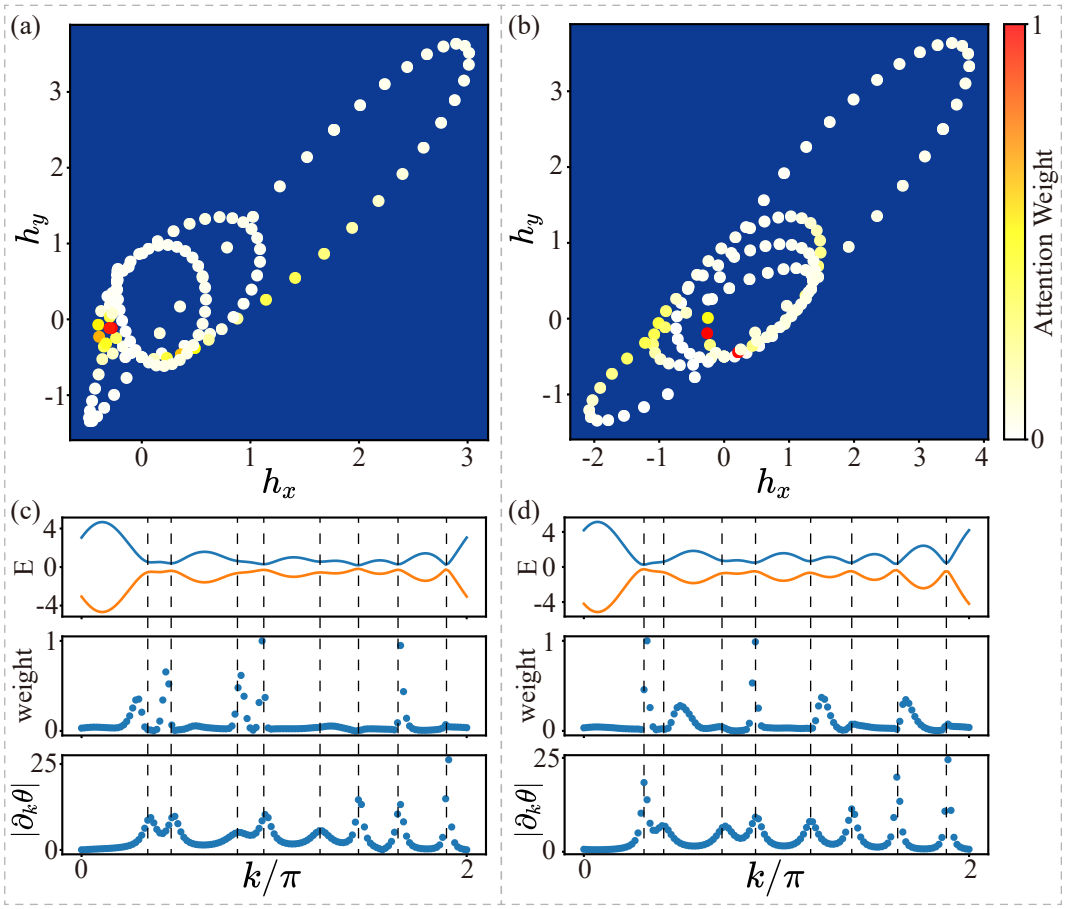


Fig. 4. Attention-based geometric tomography of the Hamiltonian of the general one-dimensional topological system. (a, b) Distribution of attention weights shown in the locus of (h_x, h_y) as k runs over the BZ, where (a) $a_x^{(4)} = -0.23$ for $w = -2$ and (b) $a_x^{(4)} = 1.37$ for $w = 4$. (c, d) Corresponding energy band, distribution of attention weights in the BZ and theoretically calculated distribution of the phase gradient $|\partial_k \theta|$ in momentum space. Here, $b_y^{(4)} = 1$ and the other parameters of the Hamiltonian are 0.5.

It can be seen that even for complex systems with larger winding numbers, TNN can still identify

the regions near $h_x = 0$ or $h_y = 0$, as shown in Figs. 4 (a) and (b). Furthermore, as shown in Fig. 4(c) and (d), these locations are also situated near the local minimal energy gaps and local maximal phase gradients. This demonstrates that even for winding numbers absent from the training set, the TNN can still achieve this simultaneous identification.

3.3. Analysis of network optimization

In this section, we additionally compare the impact of changing the number of sampling points in the BZ and the loss function on the network's predictive performance. In Section 3.3.1, we change the dimensionality of the input data by changing the number of sampling points selected within the BZ. In Section 3.3.2, we introduce a different loss function, the Huber loss function, to compare it with the MSE loss function.

3.3.1. The number of points

Theoretically, when identifying the winding number of a closed curve in momentum space, a higher density of sampling points generally facilitates easier recognition. Based on this rationale, we attempted to change the number of sampling points selected within the BZ. Following the data generation method described in Section 2.1, we regenerated the training set for different numbers of sampling points L, along with the corresponding four test sets. To reduce training costs, these training sets contain 10^5 Hamiltonians each. The training results are presented in Fig. 5.

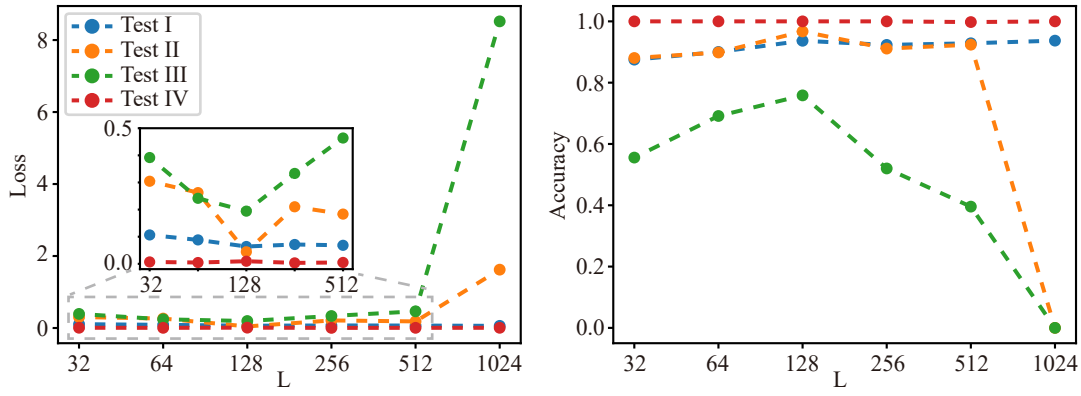


Fig. 5. Loss (left) and accuracy (right) of the networks for different numbers of sampling points L in the BZ. The blue dashed lines denote the loss value and prediction accuracy for Test I, while the orange, green, and red dashed lines correspond to Test II, Test III, and Test IV, respectively.

By comparing these networks, we observed that the network's predictive capability indeed improved as the number of sampling points L increased from 32 to 128. However, with a further increase in L, the performance began to degrade, particularly in Test II and III. When $L = 1024$, the network completely failed to identify winding numbers that were absent from the training set, recognizing only those encountered during the training process. This phenomenon is attributed to the fact that the increased number of sampling points results in excessive input information, which prevents the network from accurately extracting the key topological features that determine the winding number. This implies that when training

neural networks, it is necessary to appropriately control the dimension of input information, reducing its negative impact on network performance.

3.3.2. Loss function

Generally, the MSE loss function is a standard choice for neural networks and works well for most tasks. However, because of the squaring operation, it is very sensitive to outliers. Therefore, testing different loss functions is an important step in optimization. Given that the Mean Absolute Error (MAE) loss function leads to substantially longer training times, we replaced the MSE loss function with the Huber loss function, which incorporates the strengths of both approaches. The Huber loss function has the following functional form [42]

$$\text{Loss}_{\text{Huber}} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(\hat{y}_i - y_i)^2, & \text{if } |\hat{y}_i - y_i| \leq \delta, \\ \frac{1}{n}\sum_{i=1}^{n}\delta|\hat{y}_i - y_i| - \frac{1}{2}\delta^2, & \text{otherwise,} \end{cases} \tag{8}$$

where n is the batch size, $\hat{y}_i$ is the predicted value, y_i is the ground truth, and δ is a tunable hyperparameter. This loss function continues to use the MSE loss function when the error is small, allowing for faster convergence of the predicted value to the true value. However, for large errors, it switches to the MAE loss function, making it more robust to outliers and preventing gradient explosion caused by their presence. In this study, we set $\delta = 0.5$. This implies that for samples where the prediction is correct after rounding, the MSE loss function is used to accelerate parameter updates. Conversely, for samples where the deviation exceeds 0.5, the MAE loss function is applied to ensure more stable parameter updates. The training results are presented in Table 2. Considering that the loss values of the test set do not participate in parameter updates, and to make comparisons easier, the loss values for the test set are still computed using the MSE loss function.

Table 2. Loss and accuracy of the networks when the loss functions we used in the training process are MSE and Huber loss functions.

	Test I	Test II	Test III	Test IV
Loss for MSE loss	0.0160	0.0109	0.0605	0.0019
Accuracy for MSE loss	98.56%	99.10%	97.40%	100%
Loss for Huber loss	0.0158	0.0123	0.0506	0.0010
Accuracy for Huber loss	98.30%	98.81%	96.37%	100%

By comparing the training results of the two networks, we observed that although the network using the Huber loss function exhibited slightly lower accuracy, it achieved lower loss function values in Test I, III, and IV compared to the one using MSE loss function. This implies that with the Huber loss function, the overall deviation between predicted and true values is smaller, demonstrating higher stability. Therefore, for the task of predicting topological invariants, if high confidence is the priority, the MSE loss function is preferable; however, if the stability of predictions is paramount, the Huber loss function serves as a better alternative.

4. Conclusion

In summary, this study designed the TNN to identify the topological invariants of one-dimensional topological band insulators in AIII symmetry class. The network demonstrates the ability to recognize topological invariants that were absent from the training set, exhibiting a certain degree of generalization capability. By comparing the performance of TNN with that of CNN and FNN, it is evident that TNN significantly outperforms FNN and performs on par with CNN. Furthermore, in terms of neural network interpretability, the multi-head attention layer within the TNN encoder can quantify the correlations among all pairs of input tokens. This provides a solid foundation for investigating the network's learning process regarding topological features.

By extracting the attention weight matrices from the multi-head attention layers, averaging and normalizing them, and projecting them onto the corresponding momentum space, we obtained heatmaps of the attention weight distribution. We observed that the network focuses on the regions near $h_x = 0$ or $h_y = 0$ during identification. Comparing these with the energy bands and the theoretical distribution of the phase gradient, we found that these locations correspond to the vicinities of local minimal energy gaps and the phase gradient peaks. This implies that the network observes the phase gradient peaks—specifically, the regions that contribute most significantly to the topological invariants. Meanwhile, we also observed that even for topological structures absent from the training set, TNN accurately captures the local minimal energy gaps and the phase gradient peaks. This implies that the network does not merely establish a simple mapping based on the training data; instead, it learns the underlying correlations among the input elements and derives the corresponding results based on these relationships, thereby achieving the simultaneous identification of topological invariants and their underlying geometric origins. Crucially, this dual capability is achieved through a training solely optimized for high confidence prediction of the winding number. Namely, the network is trained solely with winding-number labels. No local geometric quantity, such as the band-crossing location, or phase gradient, is used as an additional label or included in the loss function. Nevertheless, after training for accurate topological classification, the attention maps autonomously identify momenta near local-minimum energy gaps as key regions important for the topological classification. Thus, the identification of these geometric features is not an independently imposed task; it emerges autonomously from the optimization for topological classification. This phenomenon deepens our understanding of the learning process of the neural network and offers a new way that is both efficient and interpretable for identifying topological states of matter.

Additionally, regarding precision, the physically relevant criterion here is reliable assignment to the correct integer topological sector. For example, outputs of 0.97 and 0.98 both correspond to the same topological class, $w = 1$, after assignment to the nearest integer. This does not mean that numerical accuracy is irrelevant; rather, once the correct topological sector is robustly identified, the additional scientific value of our framework is that it reveals how the network extracts the topological information from the underlying band geometry. The attention map is therefore intended as a semi-quantitative and interpretable diagnostic, rather than as an exact point-by-point reconstruction of the band geometry.

Finally, during optimizing our neural network, we explored two strategies: changing the number of sampling points and employing a more robust loss function. Training results revealed that increasing the number of sampling points to 1024 did not improve performance; instead, it caused the network to lose its generalization ability, rendering it unable to identify topological invariants not present in the training set. On the other hand, switching the loss function to Huber loss yielded a different trade-off: while it provided lower loss values in some tests, its overall performance gain was marginal for this specific task. Therefore, the standard MSE loss function remains a sufficiently effective choice.

Notably, this work leverages machine learning to directly capture the topological-geometric coupling, presenting numerous promising research avenues. Since the TNN is capable of simultaneously identifying topological invariants and their underlying geometric origins, this framework is expected to generalize well to higher-dimensional topological systems and other symmetry classes, demonstrating broad application potential. Furthermore, in the ongoing effort to demystify the "black box" nature of neural networks, recent studies have explored various approaches. For instance, chain-of-thought reasoning has been shown to enhance model performance by generating a sequence of intermediate reasoning steps prior to producing the final output [43,44]. These directions represent highly promising avenues for our future research.

Acknowledgment

This research is funded by the Key Project of the National Natural Science Foundation of China Joint Funds (No. U25A20197), and the National Natural Science Foundation of China (No. 12374246). We acknowledge helpful suggestions from Xin Zhang and Yang Yue for network optimization. We are also grateful to Yue Sun, Nan Li, Zhiyong Liu and Xuzhen Cao for reading and helpful comments on the manuscript.

参考文献

[1] Ahmed W W, Farhat M, Staliunas K, Zhang X and Wu Y 2023 *Commun. Phys.* **6** 2

[2] Wu Y, Zhang P, Shen H and Zhai H 2018 *Phys. Rev. A* **98** 010701

[3] Ch'ng K, Carrasquilla J, Melko R G and Khatami E 2017 *Phys. Rev. X* **7** 031038

[4] Zhang Y and Kim E A 2017 *Phys. Rev. Lett.* **118** 216401

[5] Scheurer M S and Slager R J 2020 *Phys. Rev. Lett.* **124** 226401

[6] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431

[7] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 *the 31st Annual Conference on Neural Information Processing Systems*, December 4-9, 2017, Long Beach, USA, p. 5998

[8] He K, Zhang X, Ren S and Sun J 2016 *2016 IEEE Conference on Computer Vision and Pattern Recognition*, June 27-30, 2016, Las Vegas, USA, p. 770

[9] Luong T, Pham H and Manning C D 2015 *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, September 17-21, 2015, Lisbon, Portugal, p. 1412

[10] Bahdanau D, Cho K and Bengio Y 2015 *3rd International Conference on Learning Representations*, May 7-9, 2015, San Diego, USA

[11] Sutskever I, Vinyals O and Le Q V 2014 *the 28th Annual Conference on Neural Information Processing Systems*, December 8-13, 2014, Montreal, Canada, p. 3104

[12] Shang H, Guo C, Wu Y, Li Z and Yang J 2025 *Nat. Commun.* **16** 8464

[13] Rahman M I 2025 *J. Transform. Technol. Sustain. Dev.* **9** 8

[14] Cubuk E D, Schoenholz S S, Rieser J M, Malone B D, Rottler J, Durian D J, Kaxiras E and Liu A J 2015 *Phys. Rev. Lett.* **114** 108001

[15] Huang L and Wang L 2017 *Phys. Rev. B* **95** 035105

[16] Fu P, Du S, Lan W, Hu L, Wu Y, Li Z, Huang X, Guo Y, Zhu W, Li J, Liu B and Gu C 2023 *Commun. Phys.* **6** 254

[17] Meng X, Zhang Y, Zhang X, Jin S, Wang T, Jiang L, Xiao L, Jia S and Xiao Y 2023 *Nat. Commun.* **14** 6105

[18] Lu S, Huang S, Li K, Li J, Chen J, Lu D, Ji Z, Shen Y, Zhou D and Zeng B 2018 *Phys. Rev. A* **98** 012315

[19] Cao J H, Chen F, Liu Q, Mao T W, Xu W X, Wu L N and You L 2023 *Phys. Rev. Lett.* **131** 073201

[20] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 *Phys. Rev. X* **8** 031086

[21] Palittapongarnpim P, Wittek P, Zahedinejad E, Vedaie S and Sanders B C 2017 *Neurocomputing* **268** 116

[22] Solinas M, Barton B, Zhang Y, Nys J and Carrasquilla J 2026 *Phys. Rev. Lett.* **136** 126501

[23] Gao J, Qiao L F, Jiao Z Q, Ma Y C, Hu C Q, Ren R J, Yang A L, Tang H, Yung M H and Jin X M 2018 *Phys. Rev. Lett.* **120** 240501

[24] van Nieuwenburg E P L, Liu Y H and Huber S D 2017 *Nat. Phys.* **13** 435

[25] Chiu C K, Teo J C Y, Schnyder A P and Ryu S 2016 *Rev. Mod. Phys.* **88** 035005

[26] Shen S Q 2017 *Topological Insulators: Dirac Equation in Condensed Matter*, 2nd edn. (Singapore: Springer) pp. 51-79

[27] Tanaka A and Tomiya A 2017 *J. Phys. Soc. Jpn* **86** 063001

[28] Zhang P F, Shen H T, and Zhai H 2018 *Phys. Rev. Lett.* **120** 066401

[29] Cong I, Choi S and Lukin M D 2019 *Nat. Phys.* **15** 1273

[30] Chen J, Wang Z, Tan Y T, Wang C and Ren J 2024 *Commun. Phys.* **7** 209

[31] Rodriguez-Nieva J F and Scheurer M S 2019 *Nat. Phys.* **15** 790

[32] Lin M R, Li W J and Huang S M 2023 *Mach. Learn.: Sci. Technol.* **4** 015032

[33] Long Y, Xue H and Zhang B 2024 *Nat. Mach. Intell.* **6** 904

[34] Zhao E, Mak T H, He C, Ren Z, Pak K K, Liu Y J and Jo G B 2022 *Opt. Express* **30** 37786

[35] Lian W, Wang S T, Lu S, Huang Y, Wang F, Yuan X, Zhang W, Ouyang X, Wang X, Huang X, He L, Chang X, Deng D L and Duan L 2019 *Phys. Rev. Lett.* **122** 210503

[36] Rem B S, Käming N, Tarnowski M, Asteria L, Fläschner N, Becker C, Sengstock K and Weitenberg C 2019 *Nat. Phys.* **15** 917

[37] Yue Y, Li N, Zhang X, Wang C, Fang Z, Ji Z, Xiao L, Jia S, Zhao Y, Bai L and Hu Y 2026 *Nat. Commun.* **17** 3539

[38] Schnyder A P, Ryu S, Furusaki A and Ludwig A W W 2008 *Phys. Rev. B* **78** 195125

[39] Moore J E and Balents L 2007 *Phys. Rev. B* **75** 121306

[40] Kane C L and Mele E J 2005 *Phys. Rev. Lett.* **95** 146802

[41] Su W P, Schrieffer J R and Heeger A J 1979 *Phys. Rev. Lett.* **42** 1698

[42] Huber P J 1964 *Ann. Math. Statist.* **35** 73

[43] Zhang Z, Zhang A, Li M, Zhao H, Karypis G and Smola A 2023 arXiv:2302.00923 [cs.CL]

[44] Zhang Z, Zhang A, Li M and Smola A 2023 *The Eleventh International Conference on Learning Representations*, May 1-5, 2023, Kigali, Rwanda