

Prediction of impurity spectrum function by deep learning algorithm

Ting Liu(刘婷), Rong-Sheng Han(韩榕生), and Liang Chen(陈亮)

Citation: Chin. Phys. B, 2024, 33 (**5**): 057102. DOI: 10.1088/1674-1056/ad3908
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Quantitative structure-plasticity relationship in metallic glass: A machine learning study

Yicheng Wu(吴义成), Bin Xu(徐斌), Yitao Sun(孙奕韬), and Pengfei Guan(管鹏飞)
Chin. Phys. B, 2021, 30 (**5**): 057103. DOI: 10.1088/1674-1056/abdda5

Resistivity minimum emerges in Anderson impurity model modified with Sachdev-Ye-Kitaev interaction

Lan Zhang(张欄), Yin Zhong(钟寅), and Hong-Gang Luo(罗洪刚)
Chin. Phys. B, 2021, 30 (**4**): 047106. DOI: 10.1088/1674-1056/abefcb

Controllable precision of the projective truncation approximation for Green's functions

Peng Fan(范鹏), Ning-Hua Tong(同宁华)
Chin. Phys. B, 2019, 28 (**4**): 047102. DOI: 10.1088/1674-1056/28/4/047102

The effects of combining alloying elements on the elastic properties of γ-Ni in Ni-based superalloy: High-throughput first-principles calculations

Baokun Lu(路宝坤), Chongyu Wang(王崇愚)
Chin. Phys. B, 2018, 27 (**7**): 077104. DOI: 10.1088/1674-1056/27/7/077104

Compton profiles of NiO and TiO_2 obtained from first principles GWA spectral function

S M Khidzir, M F M Halid, W A T Wan Abdullah
Chin. Phys. B, 2016, 25 (**6**): 067105. DOI: 10.1088/1674-1056/25/6/067105

DATA PAPER

Prediction of impurity spectrum function by deep learning algorithm

Ting Liu(刘婷)[1], Rong-Sheng Han(韩榕生)[1,2,3], and Liang Chen(陈亮)[1,2,3,†]

[1] *School of Mathematics and Physics, North China Electric Power University, Beijing* 102206, *China*
[2] *Institute of Condensed Matter Physics, North China Electric Power University, Beijing* 102206, *China*
[3] *Hebei Key Laboratory of Physics and Energy Technology, North China Electric Power University, Baoding* 071003, *China*

(Received 1 December 2023; revised manuscript received 9 March 2024; accepted manuscript online 29 March 2024)

By using the numerical renormalization group (NRG) method, we construct a large dataset with about one million spectral functions of the Anderson quantum impurity model. The dataset contains the density of states (DOS) of the host material, the strength of Coulomb interaction between on-site electrons (U), and the hybridization between the host material and the impurity site (Γ). The continued DOS and spectral functions are stored with Chebyshev coefficients and wavelet functions, respectively. From this dataset, we build seven different machine learning networks to predict the spectral function from the input data, DOS, U, and Γ. Three different evaluation indexes, mean absolute error (MAE), relative error (RE) and root mean square error (RMSE), are used to analyze the prediction abilities of different network models. Detailed analysis shows that, for the two kinds of widely used recurrent neural networks (RNNs), gate recurrent unit (GRU) has better performance than the long short term memory (LSTM) network. A combination of bidirectional GRU (BiGRU) and GRU has the best performance among GRU, BiGRU, LSTM, and BiLSTM. The MAE peak of BiGRU + GRU reaches 0.00037. We have also tested a one-dimensional convolutional neural network (1DCNN) with 20 hidden layers and a residual neural network (ResNet), we find that the 1DCNN has almost the same performance of the BiGRU + GRU network for the original dataset, while the robustness testing seems to be a little weak than BiGRU + GRU when we test all these models on two other independent datasets. The ResNet has the worst performance among all the seven network models. The datasets presented in this paper, including the large data set of the spectral function of Anderson quantum impurity model, are openly available at https://doi.org/10.57760/sciencedb.j00113.00192.

Keywords: machine learning, Anderson impurity model, spectral function

PACS: 71.10.–w, 71.27.+a, 75.20.Hr, 89.20.Ff **DOI:** 10.1088/1674-1056/ad3908

1. Introduction

The Anderson quantum impurity model (AQIM) is a very basic and important model in quantum many-body theory. It was first proposed in the 1960s to investigate the series of Kondo problems in diluted magnetic conductors.[1,2] The AQIM exhibits certain properties of strongly correlated physics, and plays an essential role in dynamical mean field theory.[3] Various theoretical and numerical methods are developed to solve this model, e.g., Bethe ansatz method,[4,5] boundary conformal field theory,[6,7] slave-boson mean field theory,[8,9] non-corssing[8] and one-crossing[10–12] approximations, numerical renormalization group (NRG),[13,14] Hirsch–Fye[15–17] and continuous-time[18] quantum Monte Carlo simulations, etc.

In recent years, machine learning methods have been widely used in different areas of physics,[19] i.e., reconstructing the image of black hole M87,[20] searching for new fundamental physics beyond the standard model,[21] dealing with large database in particle physics,[22] identifying different phases in condensed matter physics,[23–25] accelerating quantum Monte Carlo simulation,[26] first-principles calculations and material design,[27–31] to name just a few. For the AQIM in condensed matter physics, previous investigations show great power of machine learning in the prediction of spectral function and Green's function.[32–36] Arsenault *et al.*[32] construct for the first time the machine learning network with the Hubbard interaction, hybridization strength and impurity occupation as the input parameters. The typical constant density of states (DOS) for the host material is considered in this seminal work. The AQIM model is further investigated by Sturm *et al.*[33] by using neural networks and kernel ridge regression methods for very large databases, in which the Hubbard interaction, hybridization, impurity level, magnetic field, and temperature are designed as the input parameters. The band structure of host material beyond the constant DOS has been considered in Refs. [34,35]. In Ref. [34], the authors consider small clusters as the host material which is hybridized to the impurity. Different configurations of cluster energies are generated as input data for machine learning. Ren *et al.*[35] have tried for the first time using the DOS of the host material as the input data, and completed the solution of AQIM by using CNN and RNNs to predict spectral functions.

The quantum impurity solver is heavily used dynamical mean field theory. The development of generalized, fast and accurate quantum impurity solvers is an important research

†Corresponding author. E-mail: slchern@ncepu.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

direction in this field. The rapid development of machine learning methods in the last few years has brought several orders of magnitude improvement in the computational speed of impurity solvers.[33] However, in comparison with conventional methods, such as NRG and quantum Monte Carlo methods, the generality of machine learning models for impurity solving is still under development. In the initial works like Refs. [32,33], only a few discrete parameters are considered for the prediction of the spectral function. In Ref. [35], the DOS is introduced in the machine learning models. However, the general machine learning models, which have both continuous DOS and discrete parameters (such as on-site Columb intearction, hybridization, and impurity energy) as the input parameters, are not developed. In this work, we fill this gap, generalize the previous investigations by considering both the DOS of host material, the Columb interaction, and hybridization as the input data for machine learning to predict the spectral function. The prediction abilities of CNNs (including 1DCNN and ResNet) and RNNs are investigated in this work. The paper is organized as follows. In Section 2, we present the model Hamiltonian, the input data and NRG-calculation-generated standard output data. In Secion 3, we give a brief introduction about the network structures used in this work. We analyze the predition abilities of different networks in Section 4. A brief conclusion is given in Section 5.

2. Data set generation

The Hamiltonian of AQIM can be written as

$$\hat{H} = \sum_{\sigma} \varepsilon_d \hat{d}^{\dagger}_{\sigma} \hat{d}_{\sigma} + U \hat{n}_{d\uparrow} \hat{n}_{d\downarrow} + \sum_{\boldsymbol{k},\sigma} \varepsilon_{\boldsymbol{k}} \hat{c}^{\dagger}_{\boldsymbol{k}\sigma} \hat{c}_{\boldsymbol{k}\sigma} + \frac{1}{\sqrt{N}} \sum_{\boldsymbol{k},\sigma} V_{\boldsymbol{k}} (\hat{d}^{\dagger}_{\sigma} \hat{c}_{\boldsymbol{k}\sigma} + \hat{c}^{\dagger}_{\boldsymbol{k}\sigma} \hat{d}_{\sigma}), \quad (1)$$

where ε_d is the energy of the impurity, $\hat{d}^{\dagger}_{\sigma}$ and $\hat{d}_{\sigma}$ represent the creation and annihilation operators of the impurity states respectively, $\sigma = \uparrow, \downarrow$ represents the spin state, $\uparrow$ indicates that the spin state is upward, $\downarrow$ means downward spin. $U\hat{n}_{d\uparrow}\hat{n}_{d\downarrow}$ represents the Coulomb interaction of the magnetic impurity, $\hat{n}_{d\sigma} = \hat{d}^{\dagger}_{\sigma}\hat{d}_{\sigma}$ represents the particle population operator. $\varepsilon_{\boldsymbol{k}}$ represents the dispersion energy of the impurity material, $\hat{c}^{\dagger}_{\boldsymbol{k}\sigma}$ and $\hat{c}_{\boldsymbol{k}\sigma}$ represent the creation and annihilation operators of electron state with spin σ and wave vector $\boldsymbol{k}$, respectively. $V_{\boldsymbol{k}}$ describes the magnitude of the hybrid intensity, and N represents the total number of wave vectors. We use the NRG method, which solves different DOSs and corresponding spectral functions to construct input and output data sets for machine learning. In NRG calculations, the Hamiltonian can be rewritten in the following one-dimensional form:

$$\hat{H} = \sum_{\sigma} \varepsilon_d \hat{d}^{\dagger}_{\sigma} \hat{d}_{\sigma} + U \hat{n}_{d\uparrow} \hat{n}_{d\downarrow} + V \sum_{\sigma} (\hat{d}^{\dagger}_{\sigma} \hat{c}_{0\sigma} + \hat{c}^{\dagger}_{0\sigma} \hat{d}_{\sigma}) + \sum_{\sigma,n=0}^{\infty} [\varepsilon_n \hat{c}^{\dagger}_{n\sigma} \hat{c}_{n\sigma} + t_n (\hat{c}^{\dagger}_{n\sigma} \hat{c}_{n+1\sigma} + \hat{c}^{\dagger}_{n+1\sigma} \hat{c}_{n\sigma})], \quad (2)$$

where $V = \left(\sum_k |V_k|^2 / N\right)^{1/2}$ describes the hybridization of the impurity state and conduction electrons, $\hat{c}_{0\sigma} = \sum_k V_k \hat{c}_{k\sigma} / \left(\sqrt{N} V\right)$. $\hat{c}^{\dagger}_{n\sigma}$ and $\hat{c}_{n\sigma}$ $(n \geq 0)$ represent the creation and annihilation operators in the one-dimensional chain, ε_n and t_n are the coefficients of on-site energies and hopping terms in the chain. These coefficients can be obtained from the hybridization function $V_{\boldsymbol{k}}$ and the DOS of the conduction electrons after some proper discretization of the original model 1. In this work, we use the improved discretization scheme proposed in Ref. [37]. We choose the half-band-width of the conduction electron $D = 1$ as the unit of energy. The NRG discretization parameter Λ is set to 2. We set four discrete grids $N_z = 4$ to smooth the odd–even oscillation, $z = 0.25, 0.5, 0.75, 1$. In NRG calculations, at most 2000 states have been kept in each iteration.

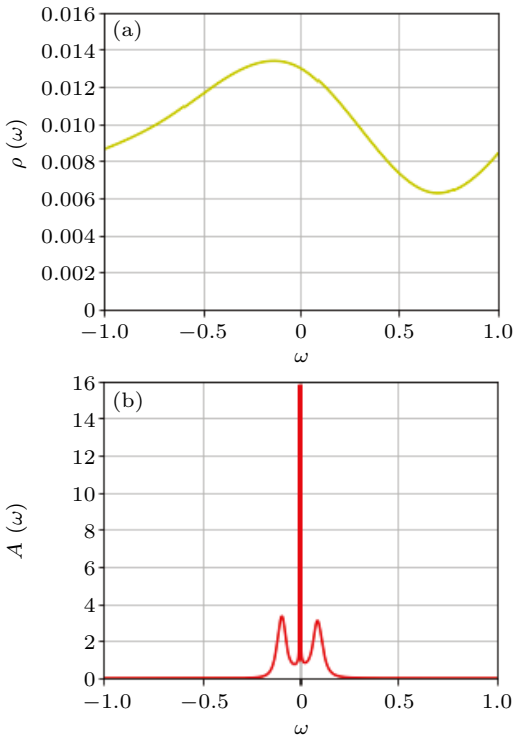


Fig. 1. NRG calculates the spectral function of a randomly selected sample. (a) DOS and two random variables. One random variable is the strength of Hubbard's repulsive force $U = -2\varepsilon_d$, where $\varepsilon_d = -0.20307$ is the impurity energy. Another random variable is the strength of the coupling between the impurity and the free electron $\Gamma = \pi V^2/(2D) = 0.06541$. (b) Spectral function. We choose half-width $D = 1$ as the unit of energy.

Here we give a detailed description of the dataset generated by NRG calculation. In our investigation, both discrete parameters and continuous DOS function are input data for machine learning. The impurity energy ε_d has been randomly distributed in the interval $[-D/2, 0]$, while the Coulomb interaction $U = -2\varepsilon_d$ is in the interval $[0, D]$. The hybridization strength in NRG calculation $\Gamma = \pi V^2/(2D)$ has been randomly distributed in the interval $[0, D/10]$. The DOS of the

host material has been generalized as follows. We set the interval $\omega \in [-2D, 2D]$, which is expressed as $\omega_j = (j-4)D/3$, $(j = 1, 2, \ldots, 7)$, corresponding to 7 grids. At the grid points between $[-2D, 2D]$, we generate 7 random numbers corresponding to them (in the range $[0, 1]$). In order to synthesize this set of scatter data into a smooth continuous function curve, we adopt the cubic spline interpolation method, which approximates the data in a small interval with low degree polynomials and uses the connection conditions of these polynomials to ensure the smoothness of the whole curve. We normalize this fitted function $f(\omega)$, to get the DOS, $\rho(\omega) = f(\omega)/\int_{-D}^{D} f(\omega)\,\mathrm{d}\omega$ for $\omega \in [-D, D]$, $\rho(\omega) = 0$ otherwise. Secondly, we use logarithmic equal-scale coordinates to calibrate the DOS for NRG calculation, 2918 grid points are constructed for zero-temperature calculation, $\omega_j^{\pm} = \pm 2\Lambda^{-j}$ $(j = 0, 1, 2, \ldots, 1458)$. Correspondingly, 2918 points of the spectral function with logarithmic coordinate are obtained by NRG calculation. Figure 1 shows an example of NRG calculation. One can find from Fig. 1(b) that the spectral function has three peaks, which correspond to the extended energy level of the impurity states and the Abrikosov–Suhl resonance peak, respectively.

In our investigation, the input data have two different types, discrete parameters and continuous function, while the output is essentially a function. For both the input DOS and output spectral function, we have 2918 points. Even the CNN is demanding for such a large system. Appropriate organization and pretreatment of the input and output data is essential for machine learning. A rough approach is to sample the functions with fewer points, as demonstrated in Ref. [35]. For general investigations, here we use the Chebyshev polynomial to express the input DOS function, while the output spectral function is approximated by wavelets (practical investigation shows that the Chebyshev polynomial is unstable to approximate the spectral function at the two ends).

We use Chebyshev polynomials of the second kind to approximate the input DOS,

$$\rho(\omega) = \sum_{i=0}^{N-1} c_i g_i U_i(\omega), \tag{3}$$

where N is the number of terms of the truncated polynomial,

$$U_i(\omega) = \frac{\sin[(i+1)\arccos(\omega/D)]}{\sin[\arccos(\omega/D)]} \tag{4}$$

is the Chebyshev polynomial defined in the interval $\omega \in [-D, D]$, g_i is a proper coefficient corresponding to a special kernel function which can be used to reduce the Gibbs oscillation. We find that the Jackson kernel[38] works well for our problem,

$$g_i = \frac{1}{N+1}\left[(N-i+1)\cos\frac{i\pi}{N+1} + \sin\frac{i\pi}{N+1}\cot\frac{\pi}{N+1}\right]. \tag{5}$$

c_i in Eq. (3) is the expansion coefficient. We find that $N = 80$ is sufficient to approximate the DOS (see Fig. 2 for typical fitting). Combining ε_d, Γ, and these coefficients $\{c_i\}$, we get the input data for machine learning,

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_{82}) = (\varepsilon_d, \Gamma, c_0, c_1, \ldots, c_{79}). \tag{6}$$

We perform discrete wavelet decomposition of spectral function to obtain output data. Its working principle is to carry out the inner product operation of the original image with the wavelet basis function and the scale function. The coefficients of 6-order wavelet decomposition are calculated, in which 150 low-frequency component coefficients CA_1 and 150 high-frequency component coefficients CD_1 of decomposition are saved respectively. The coefficients of the other higher-frequency components are omitted. The function $A(\omega)$ is obtained by wavelet reconstruction of each set of obtained coefficients. Figure 3 shows a typical spectral function reconstructed from the wavelet coefficients. One can find that the original spectral function $A(\omega)$ (black thick line) and the reconstructed function (red thin line) coincide with each other. The low-frequency coefficients CA_1 and high-frequency coefficients CD_1 are independent, so we can construct machine learning models for these datasets separately. We name $\boldsymbol{y}_\mathrm{a}$ and $\boldsymbol{y}_\mathrm{b}$ the low- and high-frequency coefficients,

$$\boldsymbol{y}_\mathrm{a} = (y_{\mathrm{a},1}, y_{\mathrm{a},2}, \ldots, y_{\mathrm{a},150}), \tag{7}$$

$$\boldsymbol{y}_\mathrm{b} = (y_{\mathrm{b},1}, y_{\mathrm{b},2}, \ldots, y_{\mathrm{b},150}). \tag{8}$$

The key problem is to build the machine learning models which work well and efficiently for the following maps:

$$\boldsymbol{x} \to \boldsymbol{y}_\mathrm{a}, \tag{9}$$

$$\boldsymbol{x} \to \boldsymbol{y}_\mathrm{b}. \tag{10}$$

We generate a very large dataset with 1 million data $(\boldsymbol{x}, \boldsymbol{y}_\mathrm{a}, \boldsymbol{y}_\mathrm{b})$. This model-training (validating) dataset is named τ_1, 80% of τ_1 is used as a training set and 20% as a validation set. Each group of input data has 82 data $(\boldsymbol{x})$, and each group of output data has 150 low-frequency data $(\boldsymbol{y}_\mathrm{a})$ and 150 high-frequency data $(\boldsymbol{y}_\mathrm{b})$. The two parts of output samples are trained separately. In addition, in order to unify the order of magnitude of data and avoid the problem that the model cannot converge due to some abnormal data values, the output data is normalized as follows: $y'_\mathrm{a} = y_\mathrm{a}/\max(y_\mathrm{a})$, $y'_\mathrm{b} = y_\mathrm{b}/\max(y_\mathrm{a})$, where y'_a and y'_b are the values after feature compression, as shown in Fig. 4. The normalization constant $\max(y_\mathrm{a})$ can be reobtained from the normalization condition of spectral function, $\int A(\omega)\,\mathrm{d}\omega = 1$.

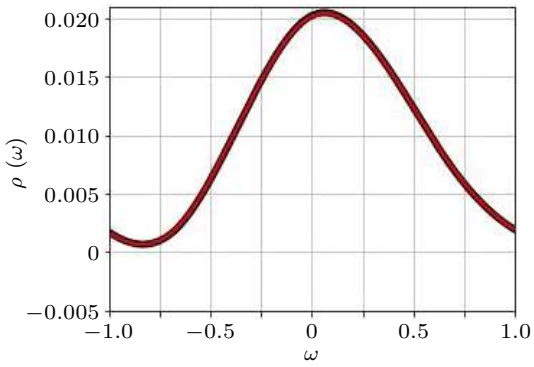


Fig. 2. Comparison between the DOS fitted by Chebyshev coefficient and the DOS calculated by NRG. The thick black line is the NRG calculation result, and the red line is the fitting result.

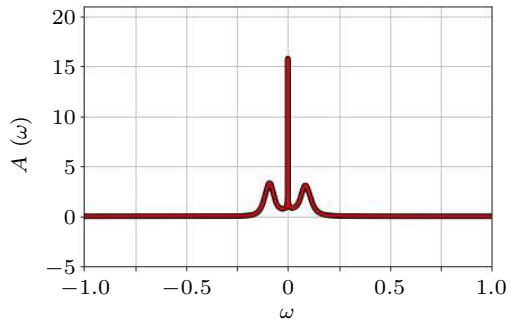


Fig. 3. Comparison of spectral function after wavelet decomposition and NRG calculation. The thick black line is the NRG calculation result, and the red line is the fitting result.

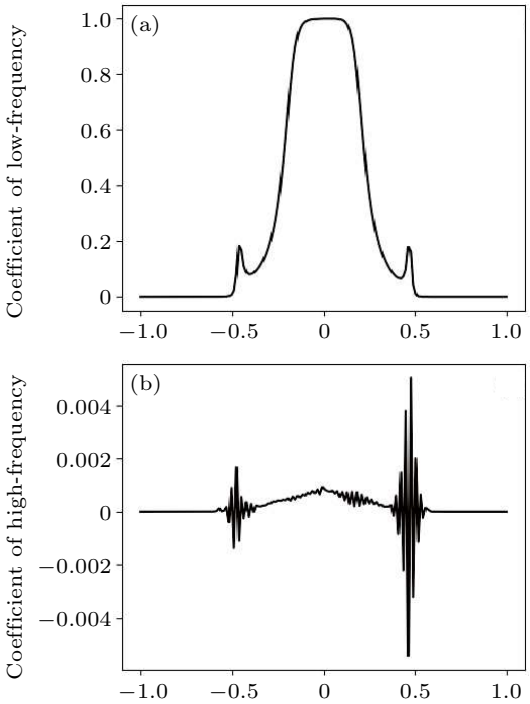


Fig. 4. The characteristic compression coefficient of spectral function after three-order wavelet decomposition. (a) Low-frequency coefficient and (b) high-frequency coefficient.

3. Neural network construction

Deep learning (DL) is a complex machine learning (ML) algorithm. With the explosion and popularity of deep learning, people have gradually separated this concept from machine learning, and its core is neural network, which is a kind of artificial intelligence technology. Based on the characteristics of different network architectures, we describe the neural networks used in this work. Recurrent neural network (RNN) is a neural network dedicated to processing data with sequence characteristics. It is very effective for mining sequence information and semantic information in data, because it has a storage memory unit inside, which can remember all sequence information and use this information for the output of the current sequence position. Convolutional neural network (CNN) generally refers to two-dimensional convolutional neural network, which is often used in the field of computer vision and is very good at processing image information efficiently. In addition, sequence prediction can also be achieved using a one-dimensional convolutional neural network (1DCNN), where the one-dimensional meaning is that the convolution kernel moves in the direction of only one dimension. Due to its relatively simple model structure and the characteristics of parameter sharing and parallel processing, 1DCNN's training speed is faster than that of the variant model based on RNN. In particular, for the 1DCNN model, adjusting the appropriate parameters can also achieve a better prediction effect. A complete 1DCNN combines five parts: input layer, convolution layer, ReLU layer, pooling layer and full connection layer.

3.1. LSTM layer

In the training process of long sequence prediction, RNN is prone to gradient dispersion and gradient explosion, which makes RNN only have short-term memory, resulting in information loss. In order to correct the shortcoming of RNN that cannot handle long sequence data, Hochreiter *et al.*[39] proposed the long short-term memory network (LSTM), which belongs to a more advanced RNN due to its particularity of internal structure. One of the key points of the LSTM structure is the gating system, which includes forgetting gate, input gate and output gate. In addition, cell states are introduced, unlike RNN which only have hidden states. Its structure is shown in Fig. 6, and the formal equations of the three gates can be obtained successively as follows:

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f), \tag{11}$$

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i), \tag{12}$$

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o), \tag{13}$$

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c), \tag{14}$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t, \tag{15}$$

$$H_t = O_t \odot \tanh C_t, \tag{16}$$

where t is the total time step, $t = 1, 2, \ldots, 82$ in this work, $\odot$ means element-wise multiplication. The three gates correspond to F_t, I_t, $O_t \in [0, 1]$ in turn, H_{t-1} is the cell output

value at $t-1$, X_t is the eigenvector at t, W and b are the weight matrix and bias matrix of the corresponding unit respectively, σ is the sigmoid activation function. The cell output value H_{t-1} at $t-1$ and the feature vector X_t at t first enter the forgetting gate F_t to determine whether to filter out the memory of the previous moment, and then select which signals need to be updated at the current moment through the input gate I_t, and cache the cell state at the current moment $\tilde{C}_t$, which is called the candidate memory state. The output gate O_t determines which part of the cell state needs to be outputted. Secondly, the cell state C_t processed by the tangent function is multiplied with the corresponding elements of the output part one by one to obtain the final cell output value H_t, thus solving the problem of gradient and achieving selective storage of information. There is a problem in using LSTM network modeling that the direction of encoded information can only be from front to back. Bidirectional long short-term memory (BiLSTM) network combines forward LSTM and backward LSTM, and combines the output vectors of two LSTM layers as the final calculation result to realize forward and backward processing sequences.

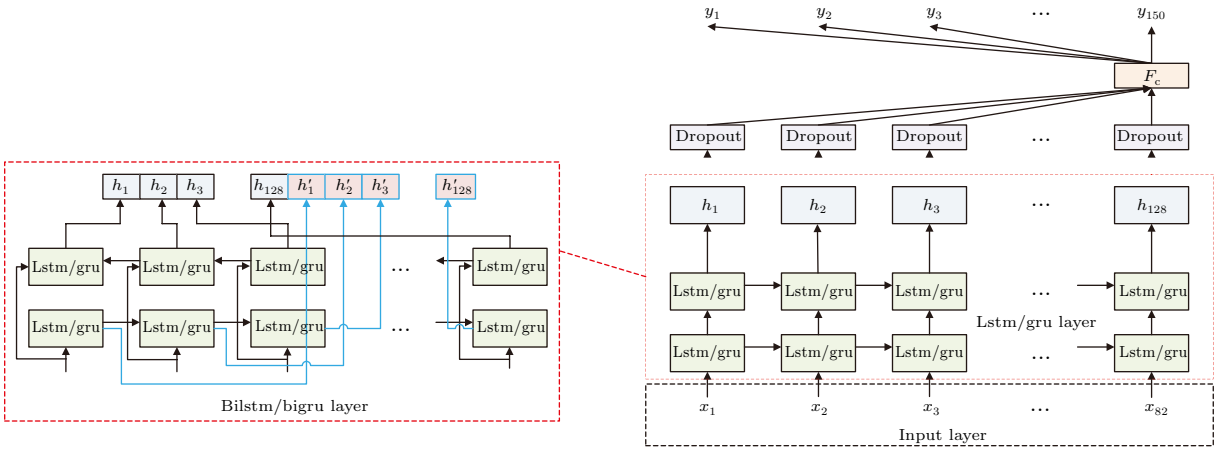


Fig. 5. RNN overall network structure.

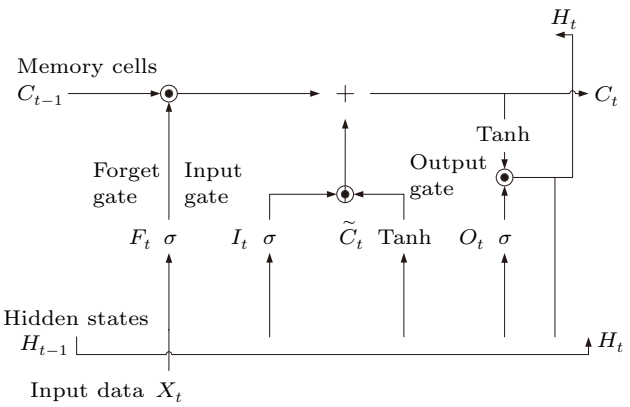


Fig. 6. LSTM working principle diagram. X_t is the eigenvector at time t, H_t and H_{t-1} are the output values at time t and $t-1$ respectively, C_t and C_{t-1} are the cell states at time t and $t-1$ respectively, $\tilde{C}_t$ is the candidate memory state, F_t, I_t and O_t are the forgetting door, the input door, and the output door, respectively, $\odot$ means element-wise multiplication, σ and Tanh are the sigmoid activation function.

3.2. GRU layer

Kyunghyun Cho *et al.*[40] proposed the gate recurrent unit (GRU), a simplified version of LSTM, which combines cell state and hidden state into one state and is controlled by only two gating structures, namely update gate Z_t and reset gate R_t. Its structure is shown in Fig. 7, and the calculation equations are as follows:

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z), \tag{17}$$

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r), \tag{18}$$

$$\tilde{H}_t = \tanh[X_t W_{xh} + W_{hh}(R_t \odot H_{t-1}) + b_h], \tag{19}$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t. \tag{20}$$

As can be seen from Eq. (20), the GRU model uses an update gate to realize the functions of both the forgetting gate and the input gate of the LSTM model, which greatly reduces the number of parameters and saves the computing cost of the computer, so it is a better choice. Forward and backward sequences can also be processed using the bidirectional gate recurrent unit (BiGRU).

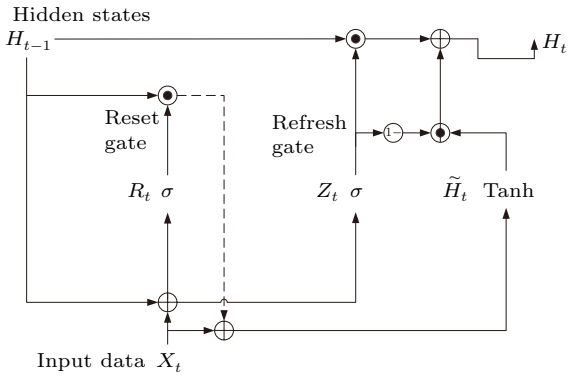


Fig. 7. Principle diagram of GRU network. X_t is the eigenvector at time t, H_t and H_{t-1} are the output values at time t and $t-1$ respectively, $\tilde{H}_t$ is the cell output after the matrix operation, Z_t and R_t are update doors and reset doors respectively. $\odot$ means element-wise multiplication, $\oplus$ is the matrix addition operation, σ and Tanh are the sigmoid activation function.

3.3. Convolution and pooling layers

The convolution layer uses convolution operation, which can also be understood as a cross-correlation operation in a non-strict sense, that is, the convolution kernel multiplies and then adds the corresponding part of the input matrix until the convolution kernel has scanned the entire input matrix, so as to achieve feature extraction. The equation is as follows:

$$\boldsymbol{O}(i,j) = (\boldsymbol{K} * \boldsymbol{I})(i,j) = \sum_m \sum_n \boldsymbol{I}(i+m, j+n)\boldsymbol{K}(m,n), \quad (21)$$

where $\boldsymbol{I}$ is the input matrix, i and j correspond to the rows and columns of the input matrix, $\boldsymbol{K}$ is the convolution kernel matrix with size $m \times n$, and $\boldsymbol{O}$ is the output matrix.

After extracting features from the convolution layer, the output matrix enters the pooling layer. We use maximum pooling, which preserves the features while reducing the number of parameters to reduce the computation.

3.4. Residual block

In the 1DCNN network, with the increase of the number of network layers, more features can be extracted, but it also causes the degradation problem, that is, with the deepening of the network, the error rate of the prediction results becomes higher. The core of the residual network (ResNet) is the introduction of CNN-based identity mapping, $X_{l+1} = F(X_l, W_l) + X_l$, where X_l is the input to the original l layer, X_{l+1} is the input to the $l+1$ layer, and $F(X_l, W_l)$ is the residual part. As shown in Fig. 8, the input is directly superimposed on the output of the non-linear layer neural network, when $F \to 0$, $X_{l+1} = X_l$, and the model degenerates into a shallow network.

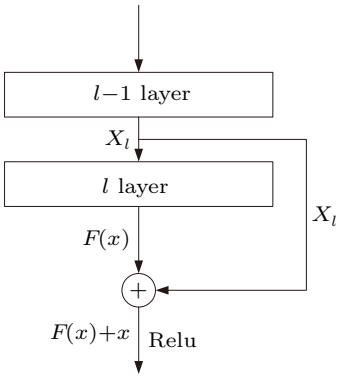


Fig. 8. Residual block schematic diagram, where l–1 and l are layer l–1 and layer l neural networks respectively, X_l is the input vector of layer l, $F(x)$ is the output of layer l network, $\oplus$ is the matrix addition operation, Relu is the activation function.

3.5. Dropout and fully connected layer

In order to prevent the problem of overfitting, Hinton *et al.*[41] proposed dropout. In the process of forward propagation, the activation value of a certain neuron is randomly discarded with a certain probability p, so that the output result will not be too dependent on some local features, thus increasing the generalization ability of the model. An illustration of dropout is given in Fig. 9.

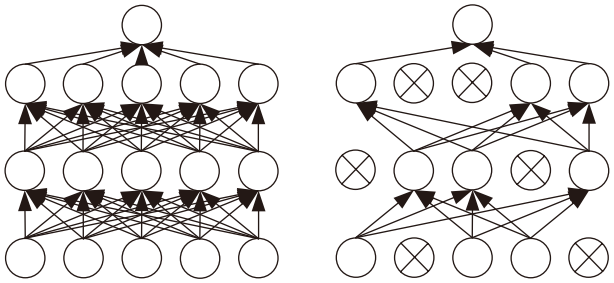


Fig. 9. Dropout schematic. The circles in the figure represent neurons, the left is a standard neural network, and the right is a neural network after killing some neurons with a certain probability.

In the last layer of the neural network, a fully connected layer is generally used, and the final prediction is realized after the output of the upper layer is nonlinearly transformed.

3.6. Optimizer and loss function

We build all our network models using the pytorch framework, which encapsulates many optimizers and loss functions. As an excellent deep learning optimizer with high efficiency and accuracy, Adam optimizer is widely used in the fields of natural language and computer vision, and it is the best choice for us to train network models. We choose L1 loss in pytorch as the loss function. This loss function is often used in regression tasks, and its explicit expression is

$$\text{Loss}(x,y) = \frac{1}{n}\sum_{i=1}^{n} |y_i - F(x_i)|, \quad (22)$$

where $F(x_i)$ is the predicted value and y_i is the truth-value.

4. Result analysis

We train LSTM, BiLSTM, GRU, BiGRU, BiGRU + GRU, 1DCNN and ResNet on a server with a graphics card of NVIDIA GeForce RTX 3090 and 23.7 GB of memory. The preparation of data is time-expensive, however, the prediction of spectral function from trained ML models is very quick. The average time for each set of data processed on the server is about 112 s if we use the standard NRG calculation. The ML model predictions are about six orders of magnitude faster, more specifically, the GRU model takes about 0.0014 s for the same data set, the BiGRU model takes also about 0.0014 s, the LSTM model takes about 0.0011 s, the BiLSTM model takes about 0.0014 s, the BiGRU + GRU model takes about 0.0022 s, the 1DCNN model takes about 0.0014 s, and the ResNet model takes about 0.0026 s. The speedup shows great advantage of ML investigation of the AQIM problem. Now we go into details about the performance of different ML models.

The overall structure diagrams of RNN and 1DCNN are shown in Figs. 5 and 10 respectively. The number of network layers for each ML model and the corresponding hyperparameters are shown in Table 1. The loss function curves obtained through training the dataset τ_1 with one million data are shown in Fig. 11. It can be observed that when the training times

reach 20 epochs, the loss functions of the low-frequency part (upper subfigure in Fig. 11) and the high-frequency part (the lower subfigure) of the spectral functions trained by all the seven models begin to converge. The further training up to 100 epochs shows that, all the models are not underfitting or overfitting. Furthermore, for each ML model, the difference between the loss functions of the training set and the verification set is very small. These results demonstrate that the training of the all ML models convergent. From the convergence of the loss functions, we can find that the model BiGRU + GRU has the best performance, i.e., this model has the smallest loss-value. 1DCNN and BiGRU are slightly inferior to BiGRU + GRU, while GRU and LSTM models have relatively poor convergence.

Table 1. Structure and hyperparameters of different models.

Model	Epochs	Batch size	Hidden layers	Dropout	Learning rate	Total params
GRU	100	100	layers 1, 2: GRU, layers 3: Linear	0.5	0.001	199830
BiGRU	90	128	layers 1, 2: BiGRU, layers 3: Linear	0.5	0.001	497814
LSTM	100	100	layers 1, 2: LSTM, layers 3: Linear	0.5	0.001	650902
BiLSTM	100	100	layers 1, 2: BiLSTM, layers 3: Linear	0.5	0.001	913046
BiGRU + GRU	100	100	layers 1, 2: BiGRU, layers 3, 4: GRU, layers 5: Linear	0.2	0.001	725910
1DCNN	100	100	layers 1, 4, 6, 8, 10, 13, 15, 17, 19: Conv1d, layers 2, 5, 7, 9, 11, 14, 16, 18, 20: ReLU, layers 12: MaxPool1d, layers 22: Linear	0.2	0.001	621462
ResNet	100	100	layers 1: RestNetBasicBlock, layers 2–4: RestNetDownBlock, RestNetBasicBlock layers 5: avgpool, layers 7: Linear	0.5	0.001	8877078

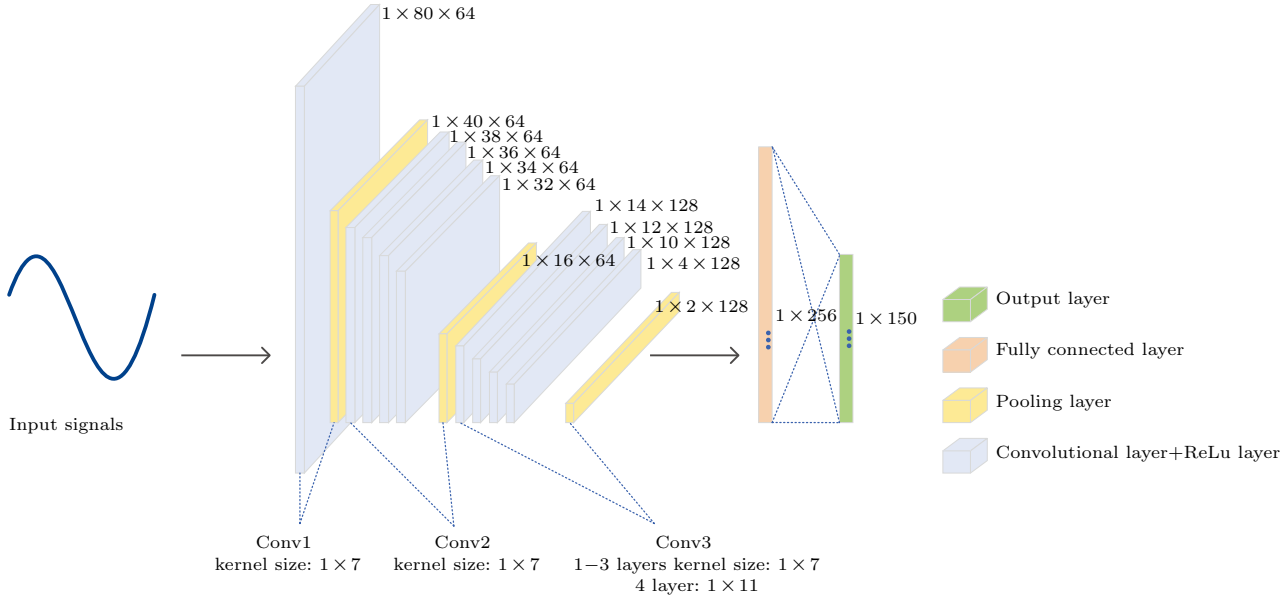


Fig. 10. The 1DCNN overall network structure.

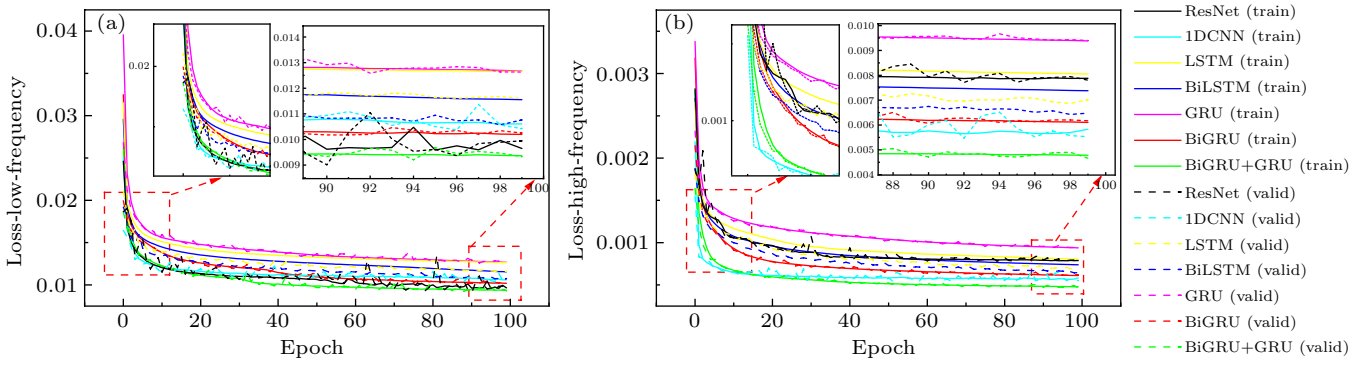


Fig. 11. The low-frequency loss-value curve (a) and the high-frequency loss-value curve (b) are trained by different neural networks on a dataset of 1 million samples. The horizontal coordinate is epoch = 100, the learning rate is 0.001, and the solid and dashed lines represent the losses in terms of the training set and the verification set, respectively. Different colors represent different neural networks.

Table 2. The τ_1 performance against the two baseline criteria of MAE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_1	91.0%	50.3%
BiGRU	τ_1	91.6%	68.8%
LSTM	τ_1	91.7%	74.1%
BiLSTM	τ_1	92.0%	75.0%
BiGRU + GRU	τ_1	92.0%	83.6%
1DCNN	τ_1	92.1%	75.2%
ResNet	τ_1	91.5%	43.4%

[a]Baseline proportion1 $= \tau_1\ (\mathrm{MAE} < 0.005)/\tau_1$,
[b]Baseline proportion2 $= \tau_1\ (\mathrm{MAE} < 0.0007)/\tau_1$.

Table 3. The τ_1 performance against the two baseline criteria of RE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_1	68.4%	2.5%
BiGRU	τ_1	78.8%	7.9%
LSTM	τ_1	80.2%	12.0%
BiLSTM	τ_1	85.3%	11.4%
BiGRU + GRU	τ_1	85.6%	17.3%
1DCNN	τ_1	86.2%	12.3%
ResNet	τ_1	60.6%	2.6%

[a]Baseline proportion1 $= \tau_1\ (\mathrm{RE} < 0.05)/\tau_1$,
[b]Baseline proportion2 $= \tau_1\ (\mathrm{RE} < 0.007)/\tau_1$.

Table 4. The τ_1 performance against the two baseline criteria of RESE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_1	87.6%	18.0%
BiGRU	τ_1	90.2%	45.0%
LSTM	τ_1	90.0%	43.0%
BiLSTM	τ_1	90.0%	55.3%
BiGRU + GRU	τ_1	91.5%	63.5%
1DCNN	τ_1	91.5%	55.5%
ResNet	τ_1	89.9%	4.3%

[a]Baseline proportion1 $= \tau_1\ (\mathrm{RESE} < 0.05)/\tau_1$,
[b]Baseline proportion2 $= \tau_1\ (\mathrm{RESE} < 0.007)/\tau_1$.

Table 5. The τ_2 performance against the two baseline criteria of MAE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_2	92.0%	55.8%
BiGRU	τ_2	92.9%	86.7%
LSTM	τ_2	92.7%	79.7%
BiLSTM	τ_2	93.0%	83.0%
BiGRU + GRU	τ_2	93.0%	88.2%
1DCNN	τ_2	92.9%	87.0%
ResNet	τ_2	93.0%	64.5%

[a]Baseline proportion1 $= \tau_2\ (\mathrm{MAE} < 0.005)/\tau_2$,
[b]Baseline proportion2 $= \tau_2\ (\mathrm{MAE} < 0.0007)/\tau_2$.

The low frequency and high frequency information of the spectral functions predicted by all the machine learning models is restored in the form of wavelet-coefficients. We can reconstruct the spectral functions from these coefficients. By comparing the predicted results and the truth-value, we can evaluate the accuracy of learning models. Figure 12 shows the prediction results for the same typical spectral function by using different models. From these figures, we can see the subtle differences in the predicted results of different models. The prediction curves of 1DCNN, BiLSTM and BiGRU + GRU models are smooth, while the other four models show weak edge oscillations. We use three error-evaluation indexes to evaluate the prediction ability of different models, i.e., the mean absolute error (MAE), the relative error (RE) and the root mean square error (RMSE),

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{23}$$

$$\mathrm{RE} = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n}|y_i|}, \tag{24}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{25}$$

Since the average error of each model cannot directly compare the distribution of the error rate of the data, we define two baseline standards respectively under different evaluation indicators. The baseline standard is the proportion of the data when the error is less than a certain specified value, and the error of baseline standard 2 is one order of magnitude smaller than that of baseline standard 1. The statistical results of MAE, RE and RMSE are shown in Tables 2–4.

According to the baseline criterion 1 of the MAE index, all models have a data proportion greater than 90%, with no significant difference in their performance. Following the baseline standard 2 of the MAE index, the MAE reaches a level of 10^{-4}, with the BiGRU + GRU model exhibiting the highest accuracy of 83.6%. The 1DCNN, BiLSTM, LSTM, and BiGRU models also have high accuracy rates, accounting for over 68% of the data. The performance of BiLSTM and BiGRU is stronger compared to LSTM and GRU, respectively, while Resnet has the poorest performance, with only 68% of its data meeting the criteria.

From the baseline criterion 1 of the RE index, we found that the 1DCNN performs the best, accounting for 86.2% of the data. The BiLSTM and BiGRU + GRU models showed similar proportions of high-accuracy data, around 85%. LSTM and BiGRU had moderate performance, with 80% of the data meeting the standards, while GRU and ResNet had lower effectiveness, with less than 70% of the data meeting the criteria. When considering the baseline standard 2 of the RE index, the RE values were on the order of 10^{-3}, and all models displayed poor performance, with BiGRU + GRU having the largest proportion of data.

Regarding the baseline criterion 1 of the RMSE index, all models performed well, with approximately 90% of the data meeting the criteria. Under the baseline standard 2 of the RMSE index, the RMSE values were on the order of 10^{-4},

with the BiGRU + GRU model having the highest proportion of high-accuracy data at 63.5%. The 1DCNN and BiLSTM models followed with around 55%, while the remaining models had less than 50% of the data meeting the criteria. GRU and ResNet exhibited weaker performance.

The results indicate that when all error indicators are reduced by one order of magnitude, the data proportion for each model significantly decreases. Under the baseline standard 2 of the RE and RMSE indicators, most data have higher errors compared to the corresponding evaluation standards. Figure 13 illustrates the prediction results of all the seven network models for some typical data with large errors. One can find that the spectral functions have three peaks, however, the middle Abrikosov–Suhl resonance peak is very sharp near $\omega = 0$, and the derivative is not continuous at this point. We suspect that this kind of abnormal spectral functions come from the NRG calculation settings. We have chosen a truncation value of energy 10^{-16} for NRG calculation. It seems that this truncation value is not low enough to present all the spectral functions.

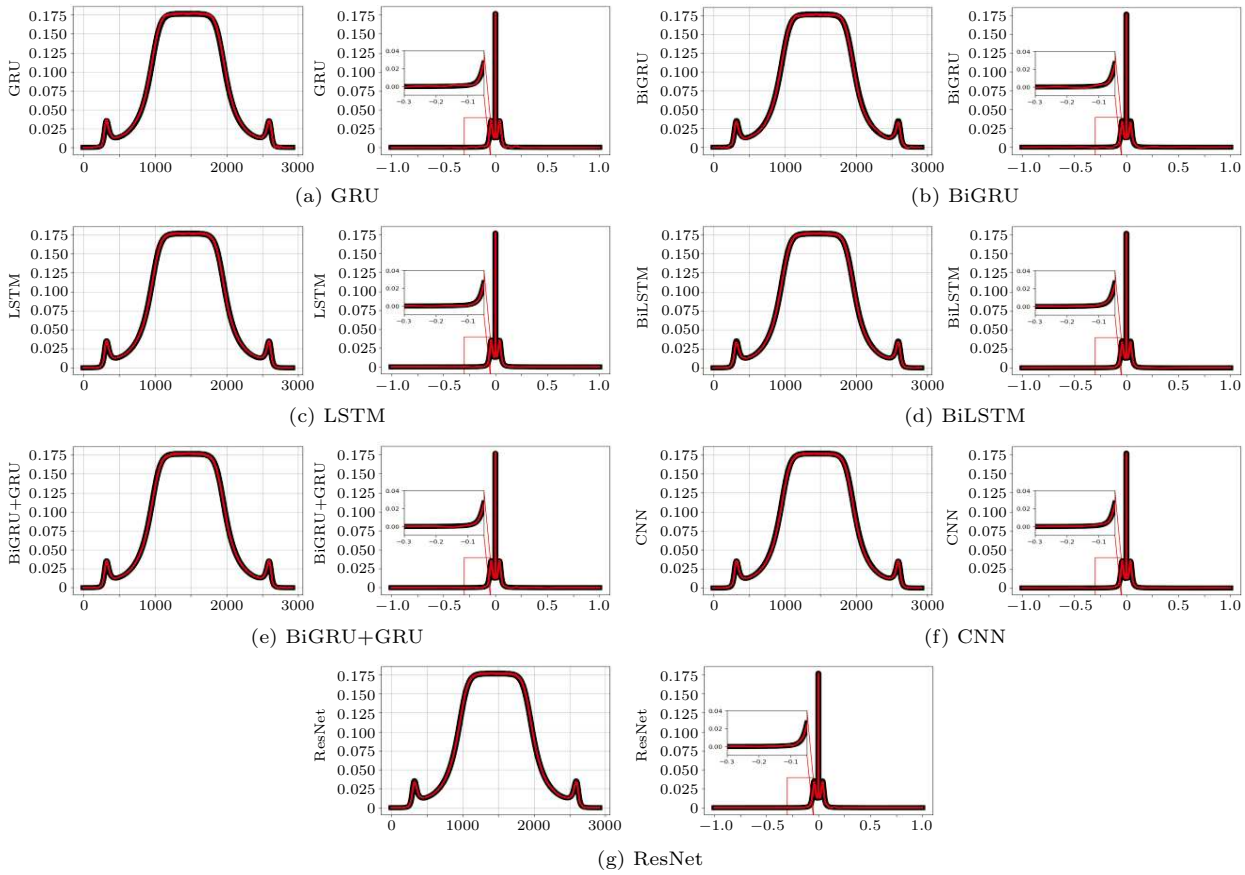


Fig. 12. The spectral function prediction results of a set of data under two coordinates of different neural networks. The left of the figure of each model is the uniform point coordinate, the right of the figure is the equal scale coordinate, the thick black line is the NRG calculation result (true value), and the red line is the prediction result. Declaration: the CNN in the figure refers to 1DCNN.

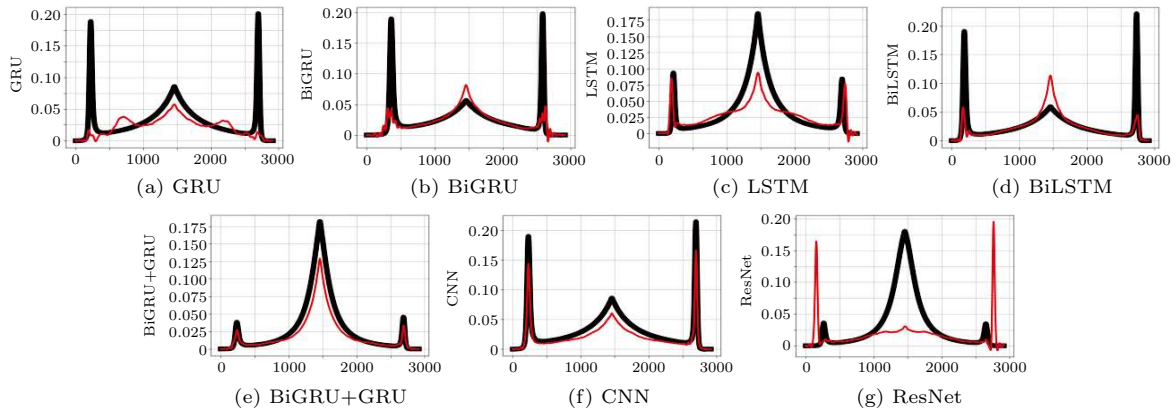


Fig. 13. The spectral function prediction results of the abnormal data caused by the system not entering the Kondo region within the temperature range studied, the thick black line is the NRG calculation result (true value), and the red line is the predicted value. Declaration: the CNN in the figure refers to 1DCNN.

Table 6. The τ_2 performance against the two baseline criteria of RE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_2	91.2%	42.5%
BiGRU	τ_2	92.6%	82.9%
LSTM	τ_2	92.3%	72.0%
BiLSTM	τ_2	92.5%	74.2%
BiGRU + GRU	τ_2	92.9%	78.0%
1DCNN	τ_2	92.6%	83.1%
ResNet	τ_2	92.8%	51.6%

[a]Baseline proportion1 $= \tau_2\ (\mathrm{RE} < 0.05)/\tau_2$,
[b]Baseline proportion2 $= \tau_2\ (\mathrm{RE} < 0.007)/\tau_2$.

Table 7. The τ_2 performance against the two baseline criteria of RMSE evaluation indicators.

Model	Dataset	Baseline proportion[a]	Baseline proportion[b]
GRU	τ_2	90.4%	25.1%
BiGRU	τ_2	92.1%	81.1%
LSTM	τ_2	91.9%	62.0%
BiLSTM	τ_2	92.0%	70.1%
BiGRU + GRU	τ_2	92.4%	80.5%
1DCNN	τ_2	92.1%	81.4%
ResNet	τ_2	92.4%	31.3%

[a]Baseline proportion1 $= \tau_2\ (\mathrm{RMSE} < 0.005)/\tau_2$,
[b]Baseline proportion2 $= \tau_2\ (\mathrm{RMSE} < 0.0007)/\tau_2$.

After removing all the abnormal data, a new database τ_2 is obtained, which has a size of 184000 and is used to retrain all the models. The statistics of different error rate indicators are presented in Tables 5–7. One can find that under the baseline criterion 1 of the MAE and RMSE, there is no significant difference in the performance between the original dataset, τ_1, and the dataset τ_2 after removing all the abnormal data. Under the baseline criterion 1 of the RE, τ_2 has better performances than τ_1 in all models. The performance improvement of τ_2 compared to τ_1 is approximately 10% for BiGRU, LSTM, BiLSTM, and BiGRU + GRU, about 20% for GRU, and about 30% for ResNet. The best-performing models are BiGRU + GRU and 1DCNN. Considering the baseline criterion 2 of the MAE index, τ_2 slightly outperforms τ_1, with BiGRU + GRU and 1DCNN being the top-performing models. However, under the baseline standard 2 of the RE, the accuracy of the τ_2 dataset in each model is approximately 40% to 75% higher than that of the τ_1 dataset when the error standard was reduced by one order of magnitude. This indicates a significant improvement in model accuracy after removing the abnormal data, with 1DCNN and BiGRU being the best performers, accounting for over 82% of the data. The performance of BiGRU + GRU is closed to 1DCNN and BiGRU, the accuracy of this model is about 78%. Under the baseline standard 2 of the RMSE index, τ_2 performs 7.1% to 36.1% better than τ_1 in all models, with 1DCNN, BiGRU, and BiGRU + GRU being the top performers, and their data proportions exceeding 80%. We find very interesting that, for ResNet and GRU, models that are less effective than others, the removal of abnormal data results in a significant increase in data proportion.

Figure 14 illustrates the statistical distribution of MAE for different models. It can be observed that the MAE statistics of the τ_1 database are not normally distributed, with a presence of large errors. In contrast, the τ_2 dataset effectively eliminates these large errors. On the τ_2 dataset, the peak MAE values are as follows: 0.0008 for GRU, 0.00046 for BiGRU and 1DCNN, 0.00055 for LSTM, and 0.00037 for BiLSTM and BiGRU + GRU. ResNet exhibits a peak MAE of 0.00064.

In order to test and improve the robustness of model performance, we add a Gaussian noise with mean zero and standard deviation 0.015 to the τ_2 dataset, and train all the network models again. The distributions of MAE for the retrained model are shown in Fig. 15. One can find that there are some small amounts of data with large errors, but the overall distribution of MAE is consistant with that before adding the Gaussian noise. The results show that adding some disturbance to the data can make the model have a certain anti-interference ability and a certain robustness to noise.

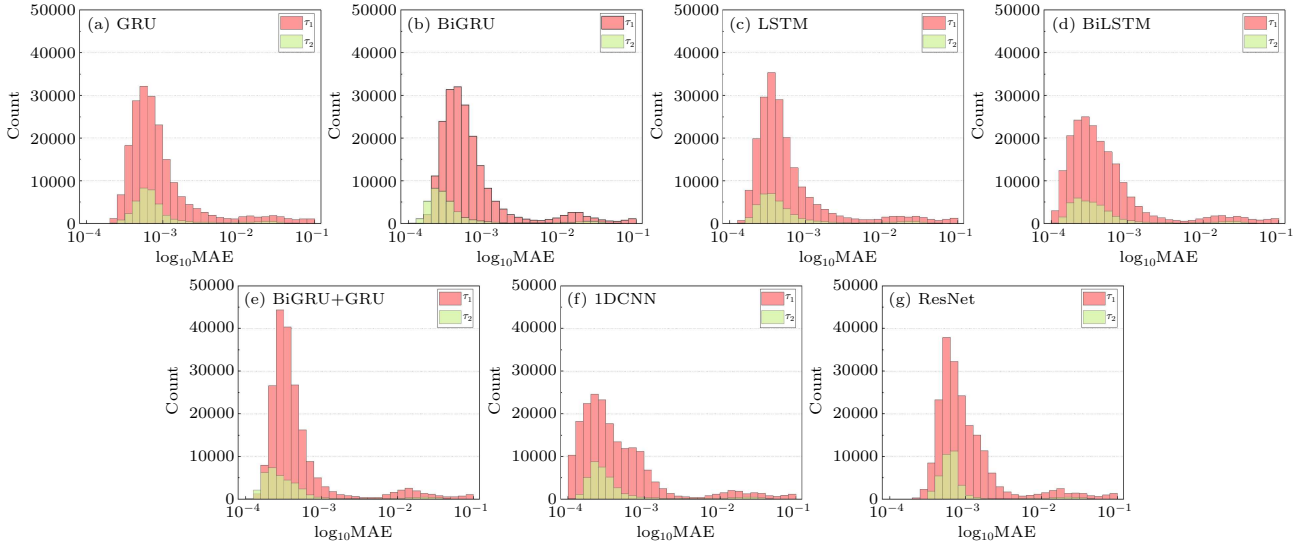


Fig. 14. Log-coordinate MAE statistical histograms of τ_1 and τ_2 data sets trained by different neural networks.

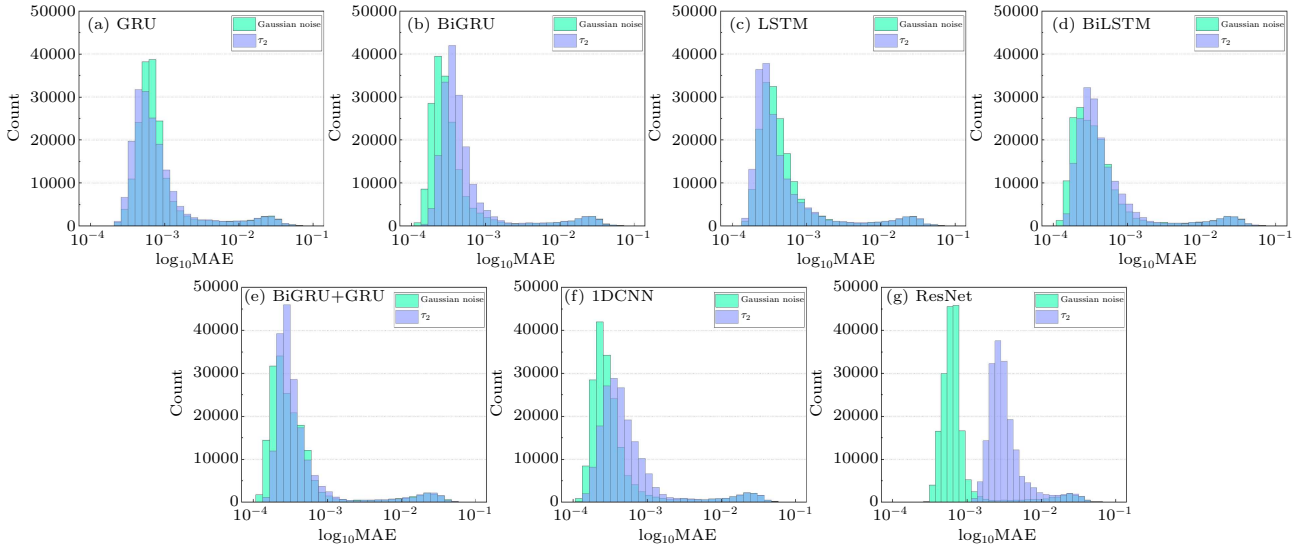


Fig. 15. Log-coordinate MAE statistical histogram trained by different neural networks after adding Gaussian noise to the τ_2.

The BiGRU + GRU and 1DCNN perform better than the other models, which is consistant with our previous investigation.[35] This can be understood as follows. Though the spectral function is dominated by the DOS near the Fermi surface, the relationship between spectral function and DOS has a long-term dependence rather than a transient correlation. This is dramatically different from natural language processing, for which LSTM has a good performance. Complex internal structure of building units, such as those used by LSTM and ResNet, makes the parameter-update to be tardy and the correlation to be short-ranged. 1DCNN and GRU have simpler internal structures, which makes the response and readjustment to be quick, long-term correlation can be detected in this case.

In comparison with previous works, our ML models have advantages. The best performance of accuracy in previous investigations is shown in Ref. [33], where the MAE is about 0.003. For the BiGRU + GRU model proposed in this work, we find that MAE for dataset τ_2 is close to 0.004, after a proper normalization of spectral functions, which shows great performance of accuracy and generality of our ML model. The input data for the ML model proposed in Ref. [33] contain only a few discrete parameters, while the input data for the ML model shown in Ref. [35] contain only the continuous DOS. In our work, both discrete parameters and continuous DOS are considered. Furthermore, in Ref. [35], both DOS and spectral functions are used by a rough sampling. Here we use Chebyshev polynomial and wavelet to describe DOS and spectral functions respectively, which is more accurate and rigorous.

5. Conclusion

For the prediction of spectral functions, we constructed a library of 1 million data samples and applied Chebyshev polynomial fitting and wavelet decomposition reconstruction methods to compress the data features. Using machine learning tools and the PyTorch framework, we trained seven different neural network models and compared their predictive performance. The accuracy of the models was evaluated using mean absolute error (MAE), relative error (RE), and root mean square error (RMSE). To test the robustness of the models, we also introduced noise into the database. Furthermore, we compared the calculation time of our approach with traditional impurity solver methods.

The results demonstrated that the methods of Chebyshev polynomial fitting and wavelet analysis reconstruction effectively reduced the number of data features and achieved a good fitting effect. These data processing techniques were essential for curve fitting. In terms of predicting past and future information, BiGRU and BiLSTM outperformed GRU and LSTM, respectively. Moreover, combining BiGRU and GRU in the BiGRU + GRU model produced even better results. Under the MAE index, the BiGRU + GRU and 1DCNN models exhibited the highest prediction accuracy. In terms of the RE and RMSE indices, 1DCNN, BiGRU, and BiGRU + GRU consistently demonstrated the best performance, with similar prediction abilities.

Additionally, the models showed a certain level of resistance to interference when Gaussian noise was added. Compared to traditional impurity solving methods, our machine learning-based approach is 10^5–10^6 orders of magnitude faster, highlighting its high efficiency in predicting magnetic impurity spectral function.

Future work will aim to improve the network's accuracy by incorporating combined networks such as CNN + LSTM and adding attention mechanisms. Additionally, we will explore the possibility of using other physical parameters, such as Hubbard interaction, impurity energy, and hybrid intensity, as input data for spectral function prediction. This research

will contribute to the development of dynamic mean field theory.

All the datasets presented in this paper, including the large data set of the spectral function of Anderson quantum impurity model, are compiled in the supplementary materials (SM) in DAT format.

Data availability statement

The data that support the findings of this study are openly available in Science Data Bank at the following link https://doi.org/10.57760/sciencedb.j00113.00192.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 12174101) and the Fundamental Research Funds for the Central Universities (Grant No. 2022MS051).

References

[1] Anderson P W 1961 *Phys. Rev.* **124** 41
[2] Anderson P W 1970 *J. Phys. C Solid State Phys.* **3** 2436
[3] Georges A, Kotliar G, Krauth W and Rozenberg M J 1996 *Rev. Mod. Phys.* **68** 13
[4] Wiegmann P B and Tsvelick A M 1983 *J. Phys. C Solid State Phys.* **16** 2281
[5] Tsvelick A M and Wiegmann P B 1983 *Adv. Phys.* **32** 453
[6] Affleck I 1990 *Nucl. Phys. B* **336** 517
[7] Affleck I 2009 arXiv:0809.3474v2 [cond-mat.str-el]
[8] Coleman P 1984 *Phys. Rev. B* **29** 3035
[9] Coleman P 1984 *Phys. Rev. B* **35** 5072
[10] Pruschke Z and Grewe N 1989 *Z. Phys. B: Condens. Matter* **74** 439
[11] Haule K, Kirchner S, Kroha J and Wölfle P 2001 *Phys. Rev. B* **64** 155111
[12] Tosi L, Roura-Bas P, Llois A M and Manuel L O 2011 *Phys. Rev. B* **83** 073301
[13] Wilson K G 1975 *Rev. Mod. Phys.* **47** 773
[14] Bulla R, Costi T A and Pruschke T 2008 *Rev. Mod. Phys.* **80** 395
[15] Hirsch J E 1983 *Phys. Rev. B* **28** 4059
[16] Hirsch J E and Fye R M 1986 *Phys. Rev. Lett.* **56** 2521
[17] Fye R M and Hirsch J E 1988 *Phys. Rev. B* **38** 433
[18] Rubtsov A N, Savkin V V and Lichtenstein A I 2005 *Phys. Rev. B* **72** 035122
[19] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 *Rev. Mod. Phys.* **91** 045002
[20] Medeiros L, Psaltis D, Lauer T R and Özel F 2023 *Astrophys. J. Lett.* **947** L7
[21] Karagiorgi G, Kasieczka G, Kravitz S, Nachman B P and Shih D 2021 *Nat. Rev. Phys.* **4** 399
[22] Boehnlein A, Diefenthaler M, Sato N, Schram M, Ziegler V, Fanelli C, Hjorth-Jensen M, Horn T, Kuchera M P, Lee D, Nazarewicz W, Ostroumov P, Orginos K, Poon A, Wang X N, Scheinker A, Smith M S and Pang L G 2022 *Rev. Mod. Phys.* **94** 031003
[23] Carrasquilla J and Melko R G 2017 *Nat. Phys.* **13** 431
[24] Rodriguez-Nieva J F and Scheurer M S 2019 *Nat. Phys.* **15** 790
[25] Lustig E, Yair O, Talmon R and Segev M 2020 *Phys. Rev. Lett.* **125** 127401
[26] Huang L and Wang L 2017 *Phys. Rev. B* **95** 035105
[27] Schleder G R, Padilha A C M, Acosta C M, Costa M and Fazzio A 2019 *J. Phys.: Mater.* **2** 032001
[28] Brockherde F, Vogt L, Li L, Tuckerman M E, Burke K and Müller K R 2017 *Nat. Commun.* **8** 872
[29] Butler K T, Davies D W, Cartwright H, Isayev O and Walsh A 2018 *Nature* **559** 547
[30] Schütt K T, Sauceda H E, Kindermans P J, Tkatchenko A and Müller K R 2018 *J. Chem. Phys.* **148** 241722
[31] Xie T and Grossman J C 2018 *Phys. Rev. Lett.* **120** 145301
[32] Arsenault L F, Lopez-Bezanilla A, von Lilienfeld O A and Millis A J 2014 *Phys. Rev. B* **90** 155136
[33] Sturm E J, Carbone M R, Lu D, Weichselbaum A and Konik R M 2021 *Phys. Rev. B* **103** 245118
[34] Walker N, Kellar S, Zhang Y, Tam K M and Moreno J 2022 *Crystals* **12** 1269
[35] Ren X Y, Han R S and Chen L 2021 *J. Phys. Condens. Matter* **33** 495601
[36] Miles C, Carbone M R, Sturm E J, Lu D, Weichselbaum A, Barros K and Konik R M 2021 *Phys. Rev. B* **104** 235111
[37] Žitko R and Pruschke T 2009 *Phys. Rev. B* **79** 085106
[38] Weiße A, Wellein G, Alvermann A and Fehske H 2006 *Rev. Mod. Phys.* **78** 275
[39] Hochreiter S and Schmidhuber J 1997 *Neural Comput.* **9** 1735
[40] Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H and Bengio Y 2014 *Proceedings of the* 2014 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1724–1734
[41] Hinton G E, Srivastava N, Krizhevsky A, Sutskever I and Salakhutdinov R R 2012 arXiv:1207.0580v1 [cs.NE]