

Design of a novel hybrid quantum deep neural network in INEQR images classification

Shuang Wang(王爽), Ke-Han Wang(王柯涵), Tao Cheng(程涛), Run-Sheng Zhao(赵润盛), Hong-Yang Ma(马鸿洋), Shuai Guo(郭帅)

Citation: Chin. Phys. B, 2024, 33 (**6**): 060310. DOI: 10.1088/1674-1056/ad342e
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

An anti-aliasing filtering of quantum images in spatial domain using a pyramid structure

Kai Wu(吴凯), Rigui Zhou(周日贵), and Jia Luo(罗佳)
Chin. Phys. B, 2024, 33 (**5**): 050305. DOI: 10.1088/1674-1056/ad2504

Recurrent neural network decoding of rotated surface codes based on distributed strategy

Fan Li(李帆), Ao-Qing Li(李熬庆), Qi-Di Gan(甘启迪), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2024, 33 (**4**): 040307. DOI: 10.1088/1674-1056/ad2bef

Simulation of optimal work extraction for quantum systems with work storage

Peng-Fei Song(宋鹏飞) and Dan-Bo Zhang(张旦波)
Chin. Phys. B, 2024, 33 (**2**): 020312. DOI: 10.1088/1674-1056/ad09d0

Quantum algorithm for minimum dominating set problem with circuit design

Haoying Zhang(张皓颖), Shaoxuan Wang(王绍轩), Xinjian Liu(刘新建), Yingtong Shen(沈颖童), and Yukun Wang(王玉坤)
Chin. Phys. B, 2024, 33 (**2**): 020310. DOI: 10.1088/1674-1056/ad02e5

Threshold-independent method for single-shot readout of spin qubits in semiconductor quantum dots

Rui-Zi Hu(胡睿梓), Sheng-Kai Zhu(祝圣凯), Xin Zhang(张鑫), Yuan Zhou(周圆), Ming Ni(倪铭), Rong-Long Ma(马荣龙), Gang Luo(罗刚), Zhen-Zhen Kong(孔真真), Gui-Lei Wang(王桂磊), Gang Cao(曹刚), Hai-Ou Li(李海欧), and Guo-Ping Guo(郭国平)
Chin. Phys. B, 2024, 33 (**1**): 010304. DOI: 10.1088/1674-1056/ace3a9

Design of a novel hybrid quantum deep neural network in INEQR images classification

Shuang Wang(王爽)[1], Ke-Han Wang(王柯涵)[2], Tao Cheng(程涛)[1], Run-Sheng Zhao(赵润盛)[2], Hong-Yang Ma(马鸿洋)[2], and Shuai Guo(郭帅)[2,†]

[1] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao* 266033, *China*
[2] *School of Sciences, Qingdao University of Technology, Qingdao* 266033, *China*

(Received 16 January 2024; revised manuscript received 23 February 2024; accepted manuscript online 15 March 2024)

We redesign the parameterized quantum circuit in the quantum deep neural network, construct a three-layer structure as the hidden layer, and then use classical optimization algorithms to train the parameterized quantum circuit, thereby propose a novel hybrid quantum deep neural network (HQDNN) used for image classification. After bilinear interpolation reduces the original image to a suitable size, an improved novel enhanced quantum representation (INEQR) is used to encode it into quantum states as the input of the HQDNN. Multi-layer parameterized quantum circuits are used as the main structure to implement feature extraction and classification. The output results of parameterized quantum circuits are converted into classical data through quantum measurements and then optimized on a classical computer. To verify the performance of the HQDNN, we conduct binary classification and three classification experiments on the MNIST (Modified National Institute of Standards and Technology) data set. In the first binary classification, the accuracy of 0 and 4 exceeds 98%. Then we compare the performance of three classification with other algorithms, the results on two datasets show that the classification accuracy is higher than that of quantum deep neural network and general quantum convolutional neural network.

Keywords: quantum computing, image classification, quantum–classical hybrid neural network, quantum image representation, interpolation

PACS: 03.67.Ac, 03.67.Lx, 03.67.–a **DOI:** 10.1088/1674-1056/ad342e

1. Introduction

Recently, the rapid development in quantum computing indicates that quantum computer has immense potential, which has brought about a ground-breaking transformation in the field of computer science.[1–3] The three principles of quantum physics: parallelism, interference and entanglement, have greatly improved the computing speed of quantum computer compared with classical computers.[4] However, qubits are extremely vulnerable to environmental impact, resulting in bit flipping or phase flipping errors. Due to the decoherence and gate infidelity, near-term quantum computer can only implement shallow quantum circuit depth, which is named as noisy intermediate-scale quantum (NISQ).[5,6] A quantum–classical hybrid algorithm has been proposed to adapt to NISQ devices, which refers to train a parameterized quantum circuit with a classical optimizer and can be used to maximize the quantum advantages.[7,8] The key element in quantum–classical algorithm is parameterized quantum circuits (PQCs), which can be considered as the neural networks in classical machine learning and are often called quantum neural networks.[9,10]

Currently, most quantum algorithms adopt this hybrid structure, such as quantum convolutional neural networks,[11] quantum approximation optimization algorithm,[12] and quantum support vector machines.[13] In Ref. [14], the author used variational quantum circuit as neurons and designed a hybrid quantum–classical architecture, demonstrating that its performance in binary classification is 10% higher than that of a single variational quantum circuit. A hybrid quantum–classical convolutional neural network was designed in Ref. [15], to conduct experiments on different earth observation benchmarks, proving that it is more general and has faster convolution operations than the classical counterpart. Additionally, in Ref. [16] the authors designed a hybrid structure and gave a method to automatically calculate the gradient of the loss function. Quantum neural networks have also been applied in multiple fields of machine vision, such as image segmentation,[17] classification,[18] recognition,[19] and encryption.[20] A hot topic in quantum computing is the design of efficient parameterized quantum circuits for image processing, particularly in image classification. In 2022, Li *et al.* designed a hybrid quantum–classical convolutional neural network to realize image classification, in which a quantum convolutional layer was employed to carry out linear unitary transformations on quantum states.[21] In Ref. [22], Mishra and Tsai designed two different efficient parameterized quantum circuits for image classification tasks and achieved good results on the MNIST (Modified National Institute of Standards and Technology) and CIFAR-10 datasets. In Ref. [23] Alam *et al.* demonstrated that using trainable quantum filters in quan-

†Corresponding author. E-mail: guoshuai@qut.edu.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

tum convolutional neural networks can yield more accurate classification results than untrained filters. A quantum deep neural network (QDNN) was formed by stacking several layers of quantum neural networks, and the design of each layer adopted a hybrid quantum–classical algorithm architecture. It has been demonstrated that QDNN can consistently approximate arbitrary continuous functions and has a greater ability to represent data than classical deep neural network.[24,25] On the other hand, multi-layer network structures can achieve good training effects, making quantum machine learning during the NISQ period a good choice.

The premise for calculating in parameterized quantum circuits is that classical data must be encoded as quantum data. Two main methods are prevalently used to encode images into quantum states. One is to transform the classical data representation of an image into quantum states, such as angle encoding. The other is to directly represent an image using quantum states, known as quantum image representation. Currently, flexible representation of quantum images (FRQI)[26,27] and the novel enhanced quantum representation (NEQR)[28,29] are the most prevalent methods. However, in existing image processing tasks using quantum neural networks, ground-state encoding methods instead of quantum image representations are used.[30–33] FRQI encodes the color and its corresponding position information of the image into a normalized quantum state.[34] NEQR is an improved method of FRQI, the difference is that multiple qubits are utilized to represent the color information of the image.[35]

Different from previous research, in this paper we design a quantum–classical hybrid quantum deep neural network (HQDNN) and construct a three-layer parameterized quantum circuit as the hidden layer. In addition, we implement image classification with images represented by an improved novel enhanced quantum representation (INEQR), which preserves more of the features of the image, thereby achieving more accurate classification results. Furthermore, in order to better adapt to NISQ machines, we use bilinear interpolation to reduce the dimensionality of the images before encoding,[36–39] so that fewer qubits are used to represent the image, thereby reducing the circuit depths of PQC. In the following we summarize the main contributions of this article: (1) The parameterized quantum circuit in the quantum deep neural network is redesigned to make the quantum encoding independent of the quantum neural network layer, realizing the use of quantum circuits to process quantum images. (2) Introducing quantum image representation into quantum deep neural networks. Quantum data retains the pixels and geometric relationships of the image, achieving better results in image processing. INEQR can encode $2^m \times 2^n$ images instead of just square ones. (3) Bilinear interpolation is employed for reducing the scale of the image, which can reduce the overhead of quantum circuits and is more suitable for current NISQ equipment.

The structure of this paper is arranged in the following manner. Section 2 introduces the encoding method of INEQR, as well as the structure of parameterized quantum circuits. Section 3 mainly includes the steps of classification and the design of quantum deep neural network. In Section 4, the feasibility of the method is proved by experiments. Section 5 summarizes the main work of this paper.

2. Preliminaries

2.1. Deep neural network

The classic deep neural network (DNN) consists of input layer, multiple hidden layers, and output layer, in which the input layer and output layer can be composed of multiple neurons, and all the layers are fully connected. DNN uses a forward propagation algorithm to calculate layer by layer until the output result is obtained. Multi-layer affine transformation and nonlinear activation functions make DNN have strong representation capabilities. Therefore, DNN is widely used in many fields of machine learning. Assuming that the affine transformation in DNN is written as $z = \sum w_i x_i + b$, the activation function is $\sigma(z)$, the output of the j neuron in the l layer p_j^l can be written as

$$p_j^l = \sigma(z_j^l) = \sigma\Big(\sum_{k=1}^{n} w_{jk}^l p_k^{l-1} + b_j^l\Big), \tag{1}$$

where n denotes the number of neurons in $l-1$ layer. Then, a loss function is selected, and the comparison between the output results of the neural network and the actual results is fed back to the previous layer of network through the back propagation method, thereby adjusting the parameter values to achieve training of the neural network.

2.2. Parameterized quantum circuit

Quantum circuits are the representation of quantum algorithms and are composed of quantum gates. The combination of different quantum gates forms quantum circuits that can fulfill various algorithm requirements.[40] Mathematically, unitary matrices can represent the quantum gates that perform transformation operations on qubits. According to the general approximation theorem in neural networks, in an any small error range, we can always find a quantum circuit to represent the objective function. In parameterized quantum circuit (PQC), some quantum gates contain variable parameters that can be trained and updated like parameters in neural networks. A typical PQC is hardware-efficient circuit,[41] which consists of several layers of elements, each comprising a layer of entanglement and a sequence of one-qubit rotation gates. The entanglement layer consists of multiple controlled NOT gates, and the rotation gate is a simple transformation of the

unitary matrix. For example, if a rotation gate is written as $U_j(\theta_i) = \exp(-\mathrm{i}\frac{\theta_j}{2}H_j)$, $H_j^2 = I$, where H_j is one of Pauli-X, Pauli-Y, and Pauli-Z matrices, $U_j(\theta_i)$ denotes the single qubit rotation gates R_x, R_y, and R_z. Common parameterized quantum gates include

$$RZ(\theta) = \begin{pmatrix} \mathrm{e}^{\frac{-\mathrm{i}\theta}{2}} & 0 \\ 0 & \mathrm{e}^{\frac{\mathrm{i}\theta}{2}} \end{pmatrix}, \quad RY(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix},$$

$$RX(\theta) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -\mathrm{i}\sin(\frac{\theta}{2}) \\ \mathrm{i}\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}, CRZ(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{e}^{\frac{-\mathrm{i}\theta}{2}} & 0 \\ 0 & 0 & 0 & \mathrm{e}^{\frac{\mathrm{i}\theta}{2}} \end{pmatrix}. \tag{2}$$

3. Image preprocessing

At NISQ era, quantum computer is affected by the limitations of the number of qubit and quantum circuit depth. At present, unrestricted quantum computing cannot be constructed, making quantum computation difficult. To reduce the processing difficulty and further reduce the depth of the quantum circuit, we use bilinear interpolation to reduce the scale of the image before encoding. Furthermore, in order to be more suitable for image processing tasks, we use quantum image representation to encode the image into quantum states.

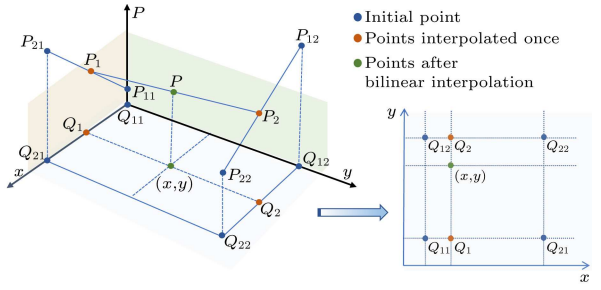


Fig. 1. The steps of bilinear interpolation. Among them, x and y donate position information of this image, and p represents the pixel value. Conduct a linear interpolation in the x direction firstly, then perform the linear interpolation again in the y direction, and finally calculate the pixel value. The image on the right shows pixels at any position.

3.1. Image scale reduction

Bilinear interpolation is often used to transform low-resolution images into high-resolution images, but if the points around the original image are ignored while interpolating the target pixels, the purpose of reducing the scale of images can be achieved. The algorithm calculates the value of the target pixel based on linear interpolation in two latitudes, involving a grid of four neighboring points. If we want to get the corresponding pixel value at coordinates (x,y), we need to find the four nearest pixels around it, and interpolate according to their weights. The weight here is determined by the distance of the coordinates and the gray value of the corresponding position. The steps of interpolation can be seen visually from Fig. 1.

$$f(Q_1) = f(x,y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}), \tag{3}$$

$$f(Q_2) = f(x,y_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}), \tag{4}$$

$$f(x,y) \approx \frac{y_2 - y}{y_2 - y_1} f(Q_1) + \frac{y - y_1}{y_2 - y_1} f(Q_2). \tag{5}$$

Given the position coordinates of four points $Q_{11}(x_1,y_1)$, $Q_{12}(x_1,y_2)$, $Q_{21}(x_2,y_1)$, $Q_{22}(x_2,y_2)$ and the corresponding pixel values P_{11}, P_{12}, P_{21}, P_{22} the interpolation point to be solved is point Q and its corresponding pixel P. First calculate the x-coordinate values Q_1 and Q_2 according to Eqs. (3) and (4), and then calculate the y-coordinate through Eq. (5) to obtain the coordinates of point Q. The same steps are used for solving the pixel P, and details are not repeated here. After dimensionality reduction preprocessing of the images, we obtain the image sets that need to be classified.

3.2. Image encoding

Next, in order to enable the images to be processed on the quantum circuits, we use INEQR to encode them into quantum images. Given a general gray-scale image, supposing the size of image is $2^m \times 2^n$, and the gray-scale value between 0 and $2^q - 1$. The encoding idea of INEQR is similar to that of NEQR, but when encoding position information, different number of qubits are used for two directions of X and Y axes.[39] The formula is expressed as

$$\begin{aligned} |I\rangle &= \frac{1}{2^{\frac{m+n}{2}}} \sum_{Y=0}^{2^m-1} \sum_{X=0}^{2^n-1} |f(Y,X)\rangle |YX\rangle \\ &= \frac{1}{2^{\frac{m+n}{2}}} \sum_{Y=0}^{2^m-1} \sum_{X=0}^{2^n-1} \otimes_{i=0}^{q-1} |C_{YX}^i\rangle |YX\rangle, \end{aligned} \tag{6}$$

in which $Y_t, X_t \in \{0,1\}$, $C_{YX}^i \in \{0,1\}$, $i = q-1, q-2, \ldots, 0$, $|f(Y,X)\rangle = |C_{YX}^{q-1} C_{YX}^{q-2} \cdots C_{YX}^1 C_{YX}^0\rangle$, $f(Y,X) \in [0, 2^q - 1]$ indicates the gray-scale values of the image represented in quantum state. $|YX\rangle = |Y\rangle|X\rangle$ denotes the position information of the pixel in the quantum image.

INEQR uses m qubits to encode the X coordinate, n qubits to encode the Y coordinate, that is to say, $m+n$ qubits are used for position information of the quantum image. In addition, q qubits are used to encode the gray value. In this way, for an image with $m \times n$ pixel values between $[0, 2^q - 1]$, a total of $m+n+q$ qubits are needed to represent it for INEQR. The encoding method is shown in Fig. 2.

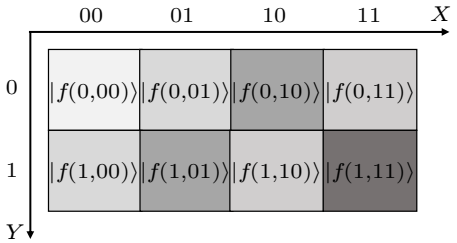


Fig. 2. The encode method of INEQR. For images with different lengths and widths, two coordinates are encoded using a different number of quantum bits. As is shown it is a 4×2 image, with 2 bits on the X axis and 1 bit on the Y axis to represent its position information.

4. Hybrid quantum deep neural network

Different from classic DNN, the network layer of quantum deep neural network is composed of PQC. We design three layers of PQC as hidden layers and use classical optimization algorithms to train the parameters of PQC, thereby proposing a hybrid quantum deep neural network (HQDNN). Different from the deep neural network in Ref. [24], our encoder is set before the neural network and uses quantum image representation, which preserves more geometric features than traditional angle encoding. The specific flow chart of the algorithm is shown in Fig. 3.

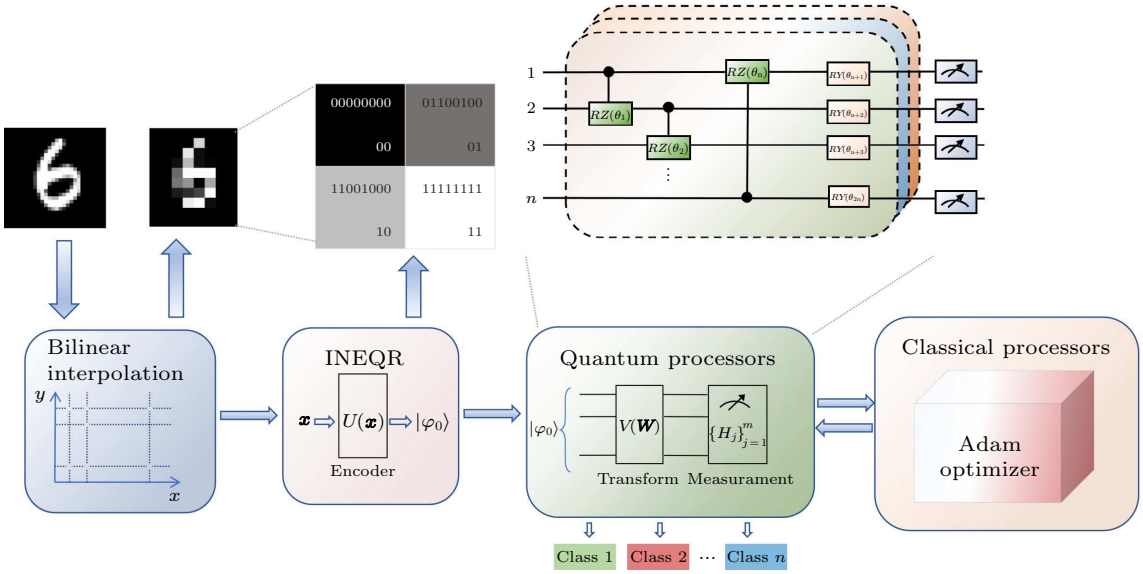


Fig. 3. Specific steps of the HQDNN algorithm. For the original image, bilinear interpolation is first used to reduce its size, then the INEQR is used to encode the picture represented by classical data into quantum states, and finally the classification is achieved using a quantum–classical hybrid structure. The quantum circuits in the quantum processor are responsible for extracting image features and implementing classification, while the classical processor serves as an auxiliary to optimize the parameters in the quantum circuit.

4.1. Hybrid quantum–classical structure

The quantum–classical hybrid algorithm is proposed to adapt to NISQ devices. Its main design idea is to combine quantum computers with classical computers to solve problems, so as to break through the limitations of current quantum computers that can only implement finite qubit calculations. The structure of the quantum classical hybrid algorithm is very similar to the machine learning model, but the classical optimizer is used to optimize a quantum circuit instead of a neural network, as shown in Fig. 4.

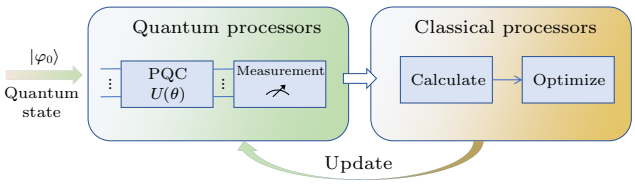


Fig. 4. Hybrid quantum classical structure, training quantum neural networks with classical optimizer.

HQDNN is composed of multiple quantum neural network layers (QNNLs). The design of QNNLs adopts the structure of a hybrid quantum–classical algorithm, using parameterized quantum circuits (PQCs) to prepare and process quantum states in quantum processors, and optimizing parameters in classical computers.

A PQC can be expressed as

$$U(\boldsymbol{\theta}) = \prod_{j=1}^{l} U_j(\theta_j), \tag{7}$$

in which $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_l)$ indicates parameters that can be optimized, $U_j(\theta_j)$ represent the rotation gates. Given a PQC $U(\boldsymbol{\theta})$ and a Hamiltonian H, where H is an observable projection operator, commonly used is Pauli Z, the loss function can be defined as the form $L = \langle 0|U^{\dagger}(\boldsymbol{\theta})HU(\boldsymbol{\theta})|0\rangle$. Then the optimization of loss function L can be achieved by updating parameters $\boldsymbol{\theta}$ using gradient-based algorithms. Adam optimizer is used in the subsequent experiments of this article.

4.2. Quantum deep neural network layer

Once the image has been represented as quantum states, it can then be processed by the quantum circuit. In this section, we mainly introduce the quantum circuit structure for classification tasks. Each layer in a quantum neural network is equivalent to a nonlinear function. However, since quantum gates are unitary matrices and their transformations are linear, it is crucial to introduce nonlinearity in the quantum neural network layer. Otherwise, the network layer will behave like an affine transformation, making no difference. In previous research, a PQC is employed as an encoder to transform input classical data into nonlinear quantum states. Here

$|\psi\rangle(\boldsymbol{x}) = U(\boldsymbol{x})|\psi_0\rangle$, with $\boldsymbol{x}$ being the input, and $|\psi_0\rangle$ representing the initial state. The transformation also designed as a PQC, which is analog to linear transformation designed in classical DNNs. For the output, fixed Hamiltonians are used as measurement, the quantum state is converted into classical data in this step. In the HQDNN of this article, the quantum encoding adopts the method of quantum image representation, therefore, it is an part independent of QNNL.

As shown in Fig. 5, we construct a parameterized quantum circuit that represents a QNNL, the HQDNN is composed of three network layers. The circuit mainly consists of a sequence of controlled rotating Z-gates and rotating Y-gates, the results are obtained through quantum measurement.

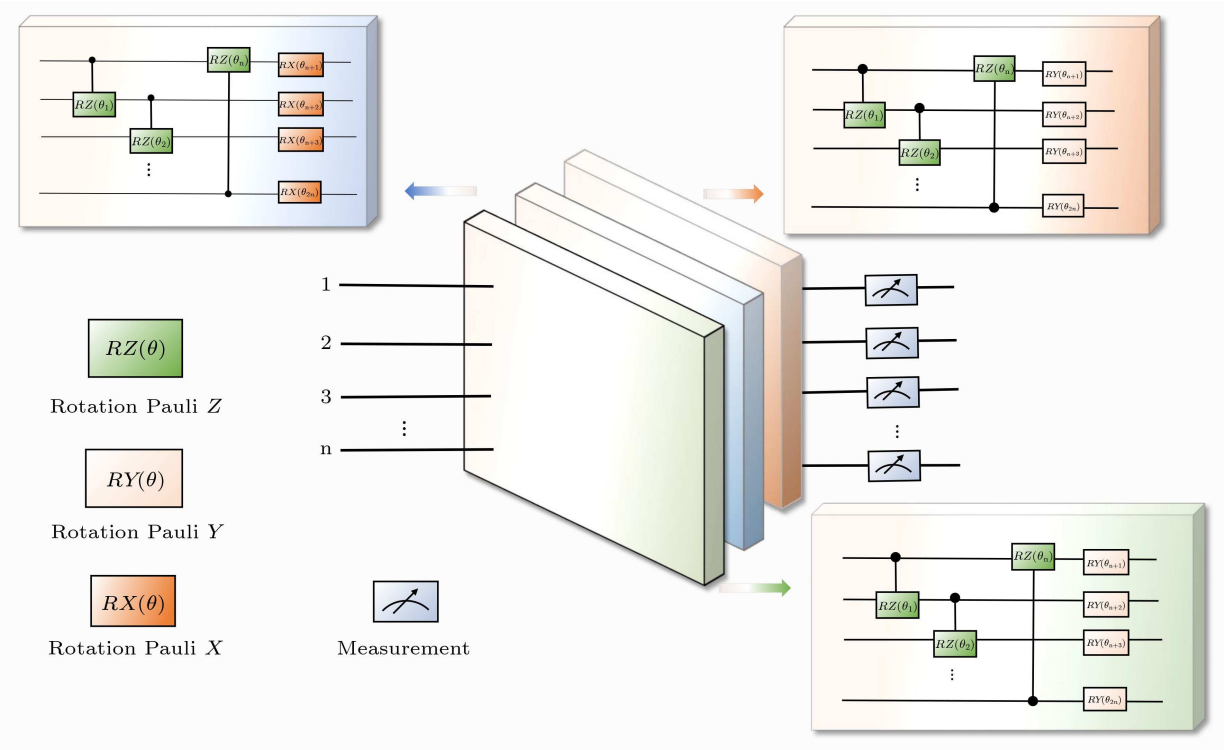


Fig. 5. The ansatz circuit of the transformation which is composed of multiple layers, and each layer consists of controlled RZ and RX, RY rotation gates, where $RY(\theta)$ and $RX(\theta)$ can cause amplitude changes, $RZ(\theta)$ can generate phase changes.

Algorithm 1 Design of quantum circuit

```
Define function circuit(inputs, weights)
initialize PQC as [RZ,X] * (n/2)
for qub in range n_qubits do
  apply Hadamard gate to qub
  for i in range n do
    if (qub * n + i) < len(inputs)
      execute PQC[i] gate with parameter inputs[qub * n + i] on qub
    else
      pass
    end if
  end for
  for l in range n_layers do
    for i in range n_qubits do
      apply CRZ gate with parameter weights [l, i] on [i, (i+1)%n_qubits]
    end for
    for j in range n_qubits to 2 * n_qubits do
      apply RY gate with parameter weights [l, j] on j % n_qubits
    end for
    for i in range (n_qubits)] do
      Initialize expectations as [qml.expval (qml.PauliZ(i))
    end for
end for
return expectations
```

It is not difficult to find that in QDNN, the quantum deep neural network layer's input and output are classical data. After introducing INEQR, our input becomes a quantum state, which is equivalent to replacing the encoder with a quantum image representation. The design process of parameterized quantum circuits is shown in Algorithm 1.

4.3. Circuit learning

Gradient-based optimization algorithm is currently most commonly used and effective in neural network training. To calculate the gradient of the parameters in the circuit, suppose that the ansatz is expressed in the following form:

$$U(\boldsymbol{\theta}) = \prod_{j=1}^{l} U_j(\theta_j),\ \ U_j(\theta_j) = \exp\Big(-\mathrm{i}\frac{\theta_j}{2}\widehat{H_j}\Big), \tag{8}$$

therefore

$$\nabla_{\theta_i} U_i(\theta_i) = -\frac{\theta_i}{2}\widehat{H_j}U_i(\theta_i). \tag{9}$$

Notably, $U_{j:k}$ represents $U_j \cdots U_l$, therefore we have the expected value of a measurement as follows:

$$\langle B(\boldsymbol{\theta})\rangle = \mathrm{Tr}(BU_{l:1}\rho_{\mathrm{input}}U_{l:1}^{\dagger}). \tag{10}$$

Taking the derivative of θ_j, we can get

$$\partial_{\theta_j}\langle B(\boldsymbol{\theta})\rangle = -\frac{\mathrm{i}}{2}\mathrm{Tr}(BU_{l:1}[H_j, U_{j-1:1}\rho_{\mathrm{input}}U_{j-1:1}^{\dagger}]U_{l:j}^{\dagger}), \tag{11}$$

and because

$$\begin{aligned} U_j\Big(\frac{\pi}{2}\Big) &= \exp\Big(-\mathrm{i}\frac{\pi}{4}P_j\Big) = \cos\frac{\pi}{4}I - \mathrm{i}\sin\frac{\pi}{4}H_j, \\ U_j^{\dagger}\Big(\frac{\pi}{2}\Big) &= U_j\Big(-\frac{\pi}{2}\Big), \end{aligned} \tag{12}$$

we can get

$$[H_j, \rho] = \mathrm{i}\Big[U_j\Big(\frac{\pi}{2}\Big)\rho U_j^{\dagger}\Big(\frac{\pi}{2}\Big) - U_j\Big(\frac{\pi}{2}\Big)\rho U_j^{\dagger}\Big(\frac{\pi}{2}\Big)\Big]. \tag{13}$$

Replace θ_j here with $\frac{\pi}{2}$ and $-\frac{\pi}{2}$ to get the gradient of θ_j

$$\begin{aligned} \partial_{\theta_j}\langle B(\boldsymbol{\theta})\rangle = &\frac{\pi}{2}\mathrm{Tr}\Big(BU_{l:j+1}U_j\Big(\frac{\pi}{2}\Big)\rho_j U^{\dagger}\Big(\frac{\pi}{2}\Big)U_{l:j+1}^{\dagger}\Big) \\ &-\frac{1}{2}\mathrm{Tr}\Big(BU_{l:j+1}U_j\Big(-\frac{\pi}{2}\Big)\rho_j U^{\dagger}\Big(-\frac{\pi}{2}\Big)U_{l:j+1}^{\dagger}\Big), \end{aligned} \tag{14}$$

where $\rho_j = U_{j:1}\rho_{\mathrm{input}}U_{j:1}^{\dagger}$.

In this way, we obtain the gradient values of the circuit parameters, so that gradient-based optimization algorithms can be used to update and optimize them.

5. Numerical experiments and analysis

Two evaluations will be conducted in this section. To adjust the size of the images, bilinear interpolation is used in the first evaluation. In the second evaluation, the classification performance of the HQDNN is compared to that of two other algorithms.

5.1. Bilinear interpolation scale reduction

Since only shallow quantum circuits can currently be implemented, we first reduce the image size before classifying the image. We select the MNIST data set, whose original pictures include 28×28 numbers 0–9. We notice the characteristics of the handwritten digits dataset, in which most of the images are located in the middle and are elongated. Therefore, we use bilinear interpolation to convert it to a size of 8 width and 10 height, which is more consistent with the characteristics of the data set and saves circuit overhead compared to the 28×28 size in literature. In addition, in some previous works, the image was reduced to 8×8, but due to the characteristics of the numbers in the MNIST dataset, more qubits should be needed in y-direction than that in x-direction to make it closer to the original data. Therefore, we believe that setting it as a square is unreasonable.

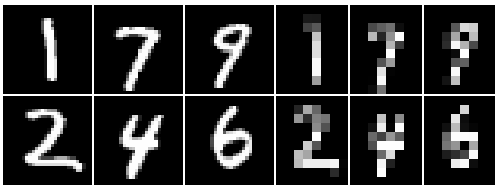


Fig. 6. The ansatz circuit of the transformation which is composed of multiple layers, and each layer is composed of controlled RZ and RX, RY rotation gates, where $RY(\theta)$ and $RX(\theta)$ can cause amplitude changes, $RZ(\theta)$ can generate phase changes.

Figure 6 shows the comparison before and after interpolation. It can be seen that the density of effective information in the picture accounts for a larger proportion. Due to $2^4 < 28 < 2^5$, for the 28×28 images, the position information of the image must be encoded using 10 qubits. Similarly, an 8×8 image requires 6 qubits to encode positional information, while an 8×10 image requires 7 qubits. Compared to using NEQR to encode images into squares, INEQR only uses an additional quantum bit to retain more image information, and higher resolution images clearly achieve better performance in subsequent processing.

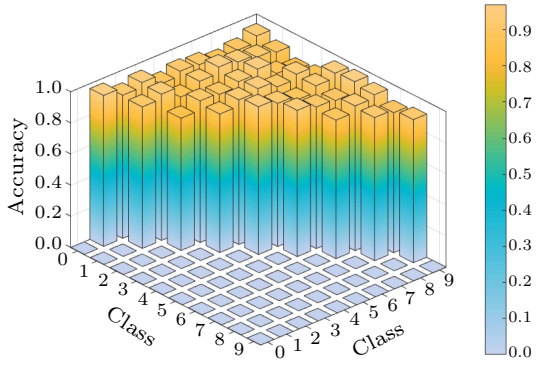


Fig. 7. Result of binary classification accuracy using BIQDNN. The two horizontal axes represent the numbers 0–9, and the vertical axis represents the classification accuracy value. The accuracy value at the intersection of x and y is the classification result of them.

5.2. Classification performance

5.2.1. Binary classification

To confirm the classification effect of our HQDNN which improved quantum deep neural network using bilinear interpolation and INEQR encoding, we first conducted a binary classification experiment on the numbers 0–9 in the MNIST dataset, classify it pairwise. The results are shown in the 3-dimensional histogram below. It is evident that the average classification accuracy is above 0.9, which is at a high level. The classification accuracy of 0 and 4 exceeds 0.98, indicating that the HQDNN algorithm can achieve high accuracy in image classification.

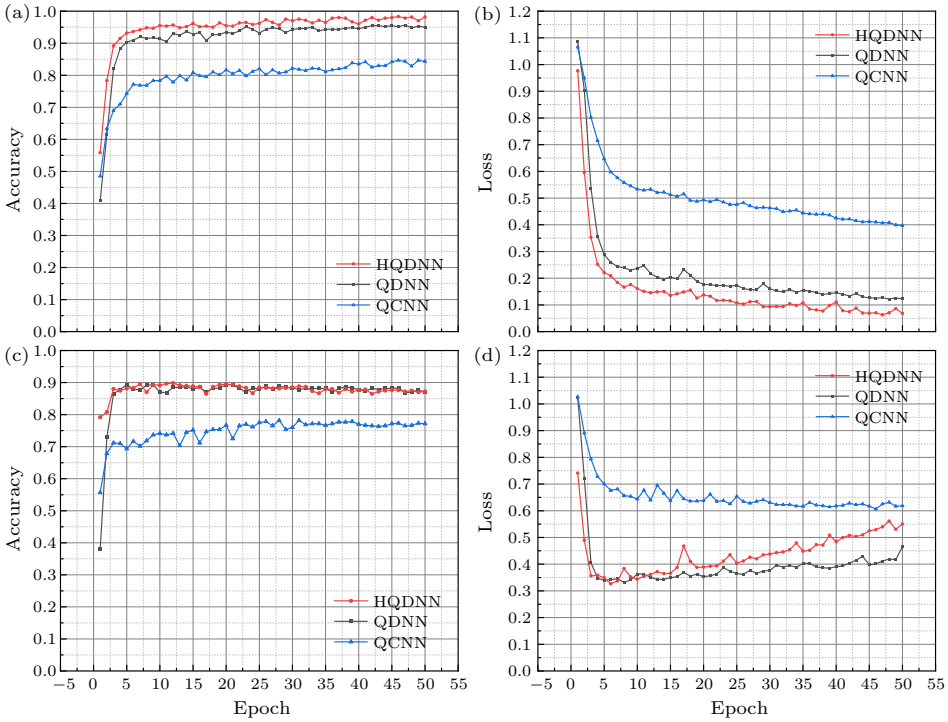


Fig. 8. Accuracy and loss value results of three classifications: (a) the accuracy of the 1, 7, and 9 training datasets, (b) the loss value of the 1, 7, and 9 training datasets, (c) the accuracy of the 1, 7, and 9 validation datasets, (d) the loss value of 1, 7, and 9 validation datasets.

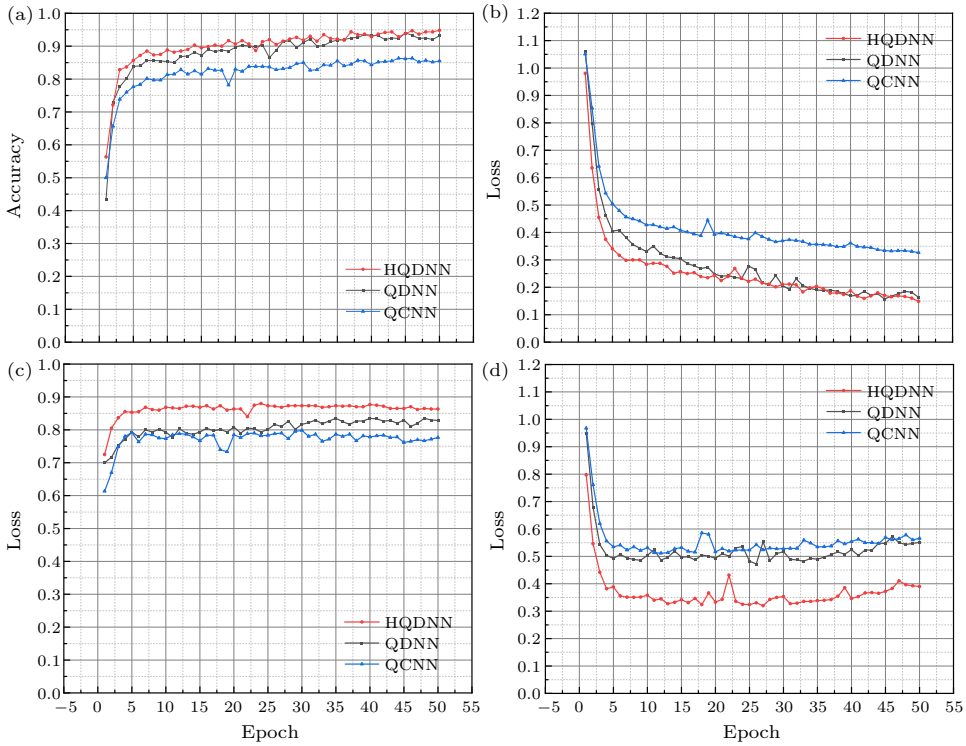


Fig. 9. Accuracy and loss value results of three classifications: (a) the accuracy of the 2, 4, and 6 training datasets, (b) the loss value of the 2, 4, and 6 training datasets, (c) the accuracy of the 2, 4, and 6 validation datasets, (d) the loss value of 2, 4, and 6 validation datasets.

5.2.2. Comparative experiment

Then, we compared the HQDNN with the angle-encoded quantum deep neural network[24] and the single-layer quantum convolutional neural network, selecting two subsets in the MNIST data set, namely, $1,7,9$ and $2,4,6$. Each small data set contains 1200 images. In order to make the comparison effect more convincing, the HQDNN and QDNN both have a three-layer structure, but they use different encoding methods, while the HQDNN and QCNN both use INEQR encoding, but their network structures are different. In addition, we all used the Adam optimizer in these three networks, set the learning rate to 0.01, and repeated 50 epochs. Figure 8 shows the result of classifying 1, 7, and 9. From the graph, it is quite obvious that the classification accuracy of the HQDNN is slightly superior than that of QDNN using simple angle encoding, and both of them have higher classification performance than single-layer quantum convolutional neural networks. The HQDNN can achieve an accuracy of 0.983 in the training set, with a loss value of 0.063. The effects of validation set is generally lower than that in training set, with a convergence accuracy of around 0.898 and a loss value of around 0.326. The training accuracy of QDNN is around 0.956, with a loss value around 0.122. The validation set accuracy is about 0.893, and the loss value is about 0.332. The classification performance of single-layer quantum convolutional neural networks is significantly lower than that of quantum deep neural networks. The training set's accuracy is 0.846, and it has a loss value of 0.397. The validation set has a accuracy and loss value of 0.781 and 0.606, respectively. Similarly, Fig. 9 is the result of classifying 2, 4, and 6. In order to see a more accurate comparison, we have compiled the results of the three classification methods on two datasets into Table 1.

Table 1. Classification result statistics.

Method	Dataset 246				Dataset 179			
	tr_acc	tr_loss	va_acc	va_loss	tr_acc	tr_loss	va_acc	va_loss
BIQDNN	0.983	0.063	0.898	0.326	0.948	0.148	0.880	0.320
QDNN	0.956	0.122	0.893	0.331	0.940	0.157	0.835	0.472
QCNN	0.846	0.397	0.781	0.606	0.863	0.326	0.798	0.511

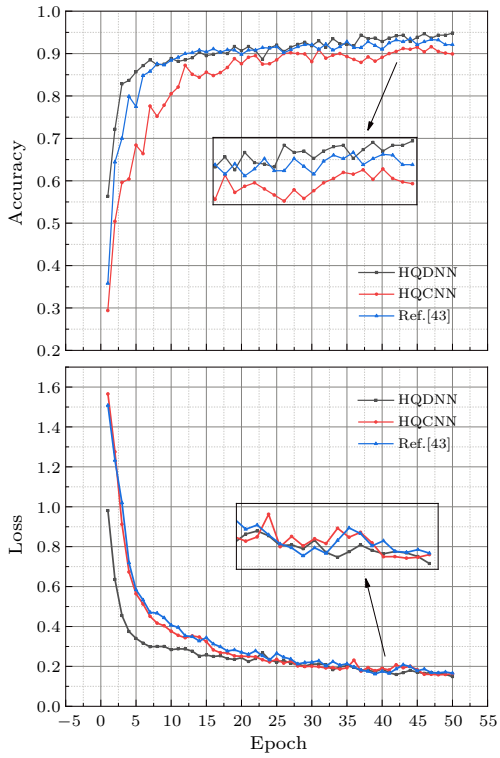


Fig. 10. Classification performance comparison of the HQDNN with the HQCNN in Ref. [42] and differential convolutional neural networks in Ref. [43].

Observing the overall trend of the curves in Figs. 8 and 9, it is not difficult to find that the accuracy of the training set for both sets of data has been continuously increasing and gradually approaching the highest value, while the accuracy convergence speed of the validation set is faster than that of the training set. Additionally, the validation accuracy value of 2, 4, 6 datasets quickly tends to stabilize, but the validation accuracy value of the 1, 7, 9 datasets has a slight downward trend after increasing. However, we believe that this does not affect the evaluation of the model's classification performance, because different datasets may exhibit differences in model performance due to their data characteristics.

Finally, in order to manifest the performance superiority of our designed model, we compare its classification performance with the hybrid quantum convolutional neural network (HQCNN) in Ref. [42] and the differential convolutional neural network in Ref. [43] using datasets 1, 7, 9 as an example. The results are shown in Fig. 10, the classification performance of the HQDNN is generally better than that of HQCNN and differential neural networks. In addition, it can be seen that the HQCNN has a faster convergence speed, but the final convergence accuracy is lower than the HQDNN.

6. Conclusions

In the NISQ era, classical quantum hybrid algorithms have become the main method to save quantum resources. The network layer in quantum deep neural networks adopts this structure and achieves good performance in image classification tasks. This article proposes a novel hybrid quantum deep neural network using a three-layer parameterized quantum circuit as the hidden layer. INEQR encodes classical images into quantum images as input to the HQDNN, the data is comprised

of not just pixel values, but also the geometric relationships of the images. Before encoding, this article uses bilinear interpolation to change the size of the image, encoding the image with fewer bits and preserving image information as much as possible. The experimental results in MNIST demonstrate that the improved method has higher classification performance than quantum deep neural networks using simple angle encoding and other models. In future work, quantum image representation methods can be applied to various fields of machine vision, such as image segmentation, and object recognition. It can also be combined with different neural networks, such as ensemble learning.

Acknowledgments

Project supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2021MF049) and the Joint Fund of Natural Science Foundation of Shandong Province (Grant Nos. ZR2022LLZ012 and ZR2021LLZ001).

References

[1] DiVincenzo D P 1995 *Science* **270** 255
[2] Steane A 1998 *Rep. Prog. Phys.* **270** 117
[3] Caraiman S and Manta V I 2015 *Quantum Inf. Process.* **14** 1693
[4] Nagy M and Akl S G 2006 *Int. J. Parallel, Emerg. Distrib. Systems* **21** 1
[5] Preskill J 2018 *Quantum* **2** 79
[6] Lau J, Wei Z, Lim K H, Shrotriya H and Kwek L C 2022 *AAPPS Bull.* **32** 27
[7] Cerezo M, Sharma K, Arrasmith A and Coles P J 2022 *npj Quantum Inf.* **8** 113
[8] Cerezo M, Arrasmith A, Babbush R, Benjamin S C, Endo S, Fujii K, McClean J R, Mitarai K, Yuan X, Cincio L and Coles P J 2021 *Nat. Rev. Phys.* **3** 625
[9] Benedetti M, Lloyd E, Sack S and Fiorentini M 2019 *Quantum Sci. Technol.* **4** 043001
[10] Beer K, Bondarenko D, Farrelly T, Osborne T J, Salzmann R, Scheiermann D and Wolf R 2020 *Nat. Commun.* **11** 808
[11] Wei S J, Chen Y H, Zhou Z R and Long G L 2022 *AAPPS Bull.* **32** 2
[12] Farhi E, Goldstone J and Gutmann S 2014 arXiv:1411.4028 [quant-ph]
[13] Lin J, Zhang D B, Zhang S, Li T, Wang X and Bao W S 2020 *Phys. Lett. A* **384** 126590
[14] Arthur D 2022 arXiv:2201.01820[cs.LG]
[15] Fan F, Shi Y, Guggemos T, *et al.* 2023 *IEEE Transactions on Neural Networks and Learning Systems* p. 1
[16] Liu J, Lim K H, Wood K L, *et al.* 2021 *Sci. China Phys. Mech. Astron.* **64** 290311
[17] Pal P, Bhattacharyya S, Platos J and Snasel V 2023 *Int. J. Hybrid Intellig.* **2** 102
[18] Gawron P and Lewiński S 2020 *IEEE International Geoscience and Remote Sensing Symposium* September 26–October 2, 2020, Waikoloa, HI, USA, p. 3513
[19] Li Y C, Zhou R G, Xu R Q, Luo J and Hu W W 2020 *Quantum Sci. Technol.* **5** 044003
[20] Shi J J, Chen S H, Chen T, Zhao T G, Tang J Q, Li Q, Yu C L and Shi H Y 2022 *Quantum Inf. Process.* **21** 214
[21] Li W, Chu P C, Liu G Z, Tian Y B, Qiu T H and Wang S M 2022 *Quantum Engineering* **2022** 5701479
[22] Mishra S and Tsai C Y 2022 *14th International Conference on Computer and Automation Engineering (ICCAE)* March 25–27, 2022, Brisbane, Australia p. 70
[23] Alam M, Kundu S, Topaloglu R O and Ghosh S 2022 *International Conference on Computer Aided Design* November 1–4, 2022, Munich, Germany, p. 70
[24] Zhao C and Gao X S 2021 *Quantum Mach. Intell.* **3** 15
[25] LeCun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436
[26] Zhou R G, Tan C Y and Ian H 2017 *Int. J. Theor. Phys.* **56** 1382
[27] Zhou R G and Tan C Y and Fan P 2017 *Mod. Phys. Lett. B* **31** 1750184
[28] Chetia R, Boruah S M B and Sahu P P 2021 *Quantum Inf. Process.* **20** 21
[29] Hao W T, Zhang T S, Chen X Y and Zhou X Y 2023 *Signal Process.* **205** 108890
[30] Farhi E and Neven H 2018 arXiv:1802.06002 [quant-ph]
[31] Alrikabi H T H S, Aljazaery I A, Qateef J S, and Alaidi A H M and Roa'a M 2022 *Int. J. Interactive Mobile Technol.* **16** 35
[32] Otgonbaatar S and Datcu M 2021 *IEEE Geosci. Remote Sens. Lett.* **19** 1
[33] Li Y C, Zhou, R G, Xu R Q, Luo J and Hu W W 2020 *Quantum Sci. Technol.* **5** 044003
[34] Le P Q, Dong F and Hirota K 2011 *Quantum Inf. Process.* **10** 63
[35] Zhang Y, Lu K, Gao Y H and Wang M 2013 *Quantum Inf. Process.* **12** 2833
[36] Kirkland E J 2010 *Bilinear Interpolation*. In: *Advanced Computing in Electron Microscopy* (Boston: Springer) p. 261
[37] Smith P R 1981 *Ultramicroscopy* **6** 201
[38] Mastyło M 2013 *J. Funct. Anal.* **265** 185
[39] Zhou R G, Hu W W, Luo G F, *et al.* 2018 *Quamtum Inf. Process.* **17** 166
[40] Huang Y C and Chen X2015 *Phys. Rev. B* **91** 195143
[41] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 *Nature* **549** 242
[42] Christol Y 2023 GitHub
[43] Sarıgül M, Ozyildirim B M and Avci M 2019 *Neural Netw.* **116** 279