Chinese Physics B

PAPER

Domain adaptation method inspired by quantum convolutional neural network

To cite this article: Chunhui Wu *et al* 2025 *Chinese Phys. B* **34** 070302

Domain adaptation method inspired by quantum convolutional neural network

Chunhui Wu(武春辉)[1,†], Junhao Pei(裴骏豪)[1], Yihua Wu(吴逸华)[1], Anqi Zhang(张安琪)[1], and Shengmei Zhao(赵生妹)[1,2,3,‡]

[1] *Institute of Signal Processing and Transmission, Nanjing University of Posts and Telecommunications (NUPT), Nanjing* 210003, *China*
[2] *Key Laboratory of Broadband Wireless Communication and Sensor Network Technology, Ministry of Education, Nanjing* 210003, *China*
[3] *National Laboratory of Solid State Microstructures, Nanjing University, Nanjing* 210093, *China*

(Received 17 November 2024; revised manuscript received 15 February 2025; accepted manuscript online 2 April 2025)

Quantum machine learning is an important application of quantum computing in the era of noisy intermediate-scale quantum devices. Domain adaptation (DA) is an effective method for addressing the distribution discrepancy problem between the training data and the real data when the neural network model is deployed. In this paper, we propose a variational quantum domain adaptation method inspired by the quantum convolutional neural network, named variational quantum domain adaptation (VQDA). The data are first uploaded by a 'quantum coding module', then the feature information is extracted by several 'quantum convolution layers' and 'quantum pooling layers', which is named 'Feature Extractor'. Subsequently, the labels and the domains of the samples are obtained by the 'quantum fully connected layer'. With a gradient reversal module, the trained 'Feature Extractor' can extract the features that cannot be distinguished from the source and target domains. The simulations on the local computer and IBM Quantum Experience (IBM Q) platform by Qiskit show the effectiveness of the proposed method. The results show that VQDA (with 8 quantum bits) has 91.46% average classification accuracy for DA task between MNIST→USPS (USPS→ MNIST), achieves 91.16% average classification accuracy for gray-scale and color images (with 10 quantum bits), and has 69.25% average classification accuracy on the DA task for color images (also with 10 quantum bits). VQDA achieves a 9.14% improvement in average classification accuracy compared to its corresponding classical domain adaptation method with the same parameter scale for different DA tasks. Simultaneously, the parameters scale is reduced to 43% by using VQDA when both quantum and classical DA methods have similar classification accuracies.

Keywords: quantum image processing, domain adaptation, quantum convolutional neural network, IBM quantum experience

PACS: 03.67.–a, 03.67.Ac, 03.67.Bg, 03.67.Lx

DOI: 10.1088/1674-1056/adc7ed **CSTR:** 32038.14.CPB.adc7ed

1. Introduction

In computer vision, the model's labeled training dataset (source domain) often has distributional differences from the real data (target domain), so the application of the model is limited. It is also a time-consuming and laborious task to label each real data to construct a useful training set. Therefore, domain adaptation (DA), an effective method for addressing the distribution discrepancy problem between the training data and the real data, makes the trained neural network models from the labeled source domain data effectively used in the target domain. Currently, there are four categories of DA methods,[1] including the discrepancy-based DA, the reconstruction-based DA, the instances-based DA, and the adversarial-based DA.[2,3] Depending on the criterion used to measure the distributional differences, the discrepancy-based DA can be further categorized into DA based on statistical criterion,[4] DA based on structural criterion,[5] DA based on prevalence criterion,[6] and DA based on graph criterion.[7] Reconstruction-based DA is mainly implemented by extracting domain-invariant features through a self-encoder.[8] Instances-based methods first generate the labeled target domain samples from the labeled source domain samples, establish the relationship between the source domain samples and the generated samples, and then use the generated samples to train the network suitable for the target domain.[9] While the adversarial-based DA is to introduce the game idea of generative adversarial network[10] (GAN) into the network training. For example, Ganin *et al.* first proposed a domain adversarial neural network[11] (DANN) by introducing a domain classifier and a gradient reversal layer in the feed-forward network.

Machine learning is one of the important applications of quantum computing,[12] and the resultant quantum machine learning[13–15] (QML) provides polynomial or exponential acceleration for learning tasks. Based on the superposition and entanglement of quantum bits, QML is expected to overcome the current problems in DA on its large data set and slow train-

[†] Corresponding author. E-mail: wch13295016367@163.com

[‡] Corresponding author. E-mail: zhaosm@njupt.edu.cn

ing process.[16–19] In the context of the noisy intermediate-scale quantum (NISQ) era, the variational quantum algorithm (VQA) was proposed to provide a general framework for the implementation of QML.[20] VQA has three steps, (i) design the objective function and the corresponding variational quantum circuit (VQC) according to the learning task, (ii) solve the expectation value of the objective function by using VQC, and (iii) optimize the parameters in VQC by classical computation and find the optimal parameters to satisfy the objective function.[21,22] VQA can greatly reduce the number of quantum bits, quantum gates, and the depth of required circuits by combining classical computing and quantum computing, it has become an effective way to realize quantum superiority[23] nowadays.

Quantum convolutional neural network (QCNN)[24] was used to accurately identify the quantum states, and it is one of the VQC models in VQA. Later, Hur *et al.* discussed the performance of various QCNN models in terms of the structure of VQC, quantum data coding methods, classical data preprocessing methods, and loss functions.[25] Lü *et al.* extended the application of QCNN from quantum data to classical image data by implementing a binary classification task in MNIST dataset.[26] Yang *et al.* proposed federal learning in the context of a decentralized feature extraction method based on QCNN to solve the privacy preservation problem in speech recognition.[27] Wei *et al.* also applied QCNN in image processing, such as image smoothing, image sharpening, and edge detection.[28]

Inspired by QCNN, we propose a variational quantum domain adaptation method, named VQDA, in which a new QCNN model is designed by using VQC, which has a similar hierarchical structure to a classical convolutional neural network (CNN) model, that is, the new QCNN model has the quantum convolutional layer, the quantum pooling layer, and the quantum fully connected (QFC) layer. The operations on quantum data, such as convolution, pooling, and full connection, can be implemented by the quantum convolutional layer, the quantum pooling layer and the quantum fully connected layer. Finally, VQDA is achieved by introducing an additional quantum fully connected layer for classifying domains in the feed-forward network, and a gradient reversal module (GRM) in the back-propagation. The simulations are performed on the local computer and IBM Quantum Experience (IBM Q) platform by Qiskit to validate the proposed method. We further discuss the effectiveness of the proposed VQDA on the Digit-5 dataset for 'gray images' domain adaptation, 'color and gray images' domain adaptation, and 'color images' domain adaptation.

The main contribution of the paper can be summarized as follows. (i) The proposed scheme incorporates quantum computing into domain adaptation to improve the average classification accuracy of DA task with only a few qubits, say 8(10) qubits. (ii) VQDA promotes the emergence of two features: one is discriminative for the main learning task on the source domain and another is indiscriminate with respect to the shift between the domains. (iii) The proposed scheme utilizes the advantages of quantum convolutional neural networks to achieve domain adaptation with 43% parameters scale reduced in comparison with its classical counterpart for the similar classification accuracy. (iv) The proposed scheme does not contain any quantum black boxes and has been verified on the IBM Quantum Experience (IBM Q) platform.

2. Variational quantum domain adaptation

The quantum circuit of the proposed VQDA is shown in Fig. 1. The classical information (such as images) is firstly encoded into their corresponding quantum states through a 'Quantum Coding Module', and then the generated quantum states are extracted from the sample features by a 'Feature Extractor', composed of a hierarchical structure with the 'quantum convolution layer' (QCL) and the 'quantum pooling layer' (QPL). Here, the 'quantum convolution layer' is comprised of several two-quantum bit unitary transformations U_i by invariant translation, and the 'quantum pooling layer' consists of some measurement-control circuits that determine whether or not to apply a single-quantum bit unitary transformation V_i to its adjacent quantum bit. QCL is used to extract the feature of the input quantum state, while QPL is used to reduce the feature mapping dimensions, achieve nonlinearity, and prevent the overfitting. With the alternate applications of QCL and QPL, the remaining quantum bits are measured to obtain the extracted features when the system size is small enough. By measuring the 'quantum fully connected layer' (named QFC1), the labels of the input feature prediction samples are obtained after a unitary on the remaining qubits, where the unitary θ_{QFC1} is learned. The 'Feature Extractor' and QFC1 construct a feed-forward network model, composing of a label prediction part ('Label predictor'). The output of QFC1 is used to predict the labels of the samples. At the same time, an additional 'quantum fully connected layer' (named QFC2) is added to VQDA for predicting the features whether from the source domain or the target domain. The 'Feature Extractor' and QFC2 are composed of 'Domain classifier', and the output is used to predict the domains of the samples. In particular, the gradient of QFC2 is multiplied by -1 through the GRM. QFC2 and GRM are used to ensure that the trained 'Feature Extractor' can extract the features that cannot be distinguished from the two domains (i.e., the common features of the two domains).

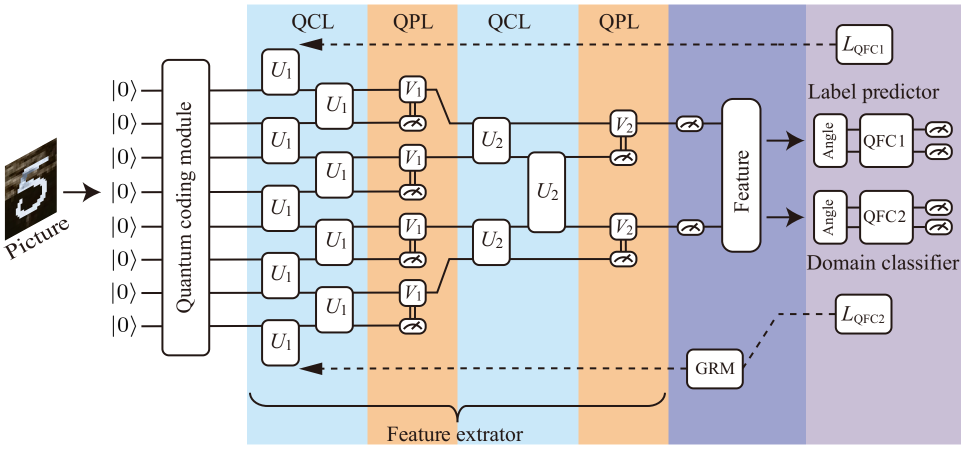

Fig. 1. The architecture diagram of the proposed VQDA. U is the two-quantum bit, and V is the one-quantum bit unitary transformation. QCL, QPL, QFC, and GRM represent the quantum convolution layer, the quantum pooling layer, the quantum fully-connected layer and the gradient inversion module, respectively. L_{QFC1} and L_{QFC2} are the loss functions for QFC1 and QFC2.

In the training phase, the source domain image participates in label training and domain training, while the target domain image only participates in domain training. The gradient of QFC2 is inverted by GRM during back-propagation, and θ_{cp} and θ_{QFC1} are learned by minimizing the loss of the 'Label predictor', while θ_{cp} and θ_{QFC2} are learned by maximizing the loss of the 'Domain classifier'.

The unsupervised DA is achieved by adding 'Domain classifier', who connects to the feature extractor via a gradient reversal layer that multiplies the gradient by -1 during the backpropagation-based training. Gradient reversal ensures that the feature distributions over the two domains are made similar, thus resulting in the domain-invariant features, satisfying with the criterion for effective domain transfer, that is, the predictions must be made based on features that cannot discriminate between the training (source) and test (target) domains. VQDA can promote the emergence of features that are discriminative for the learning task on the source domain and indiscriminate with respect to the shift between the domains.

The first step of VQDA is preparing the quantum state of the input classical information (images). Here, 'Quantum Coding Module' is implemented by a specific VQC to obtain the required quantum state based on the amplitude state[29] preparation method proposed in Ref. [30].

In the following, we describe each part in detail.

2.1. Quantum convolutional layer

The quantum convolutional layer is designed to extract the feature of the input quantum state. It is comprised of some two quantum bits gate U_i, where i denotes the convolutional layer in which it is located. Here, U_i operates on the two neighboring quantum bits to demonstrate the local connectivity, as the convolutional kernel in the convolutional neural network (CNN). At the same time, U_i is sequentially operated on each quantum bit in a translationally-invariant manner to show the parameters sharing property. To extract more features, the amplitude of the input quantum state should be flexibly adjusted by U_i. Thus, U_i is a universal two-quantum bit circuit that can realize any unitary transformation. According to the decomposition fashion in Ref. [31], we decompose U_i into a combination of a collection of basic quantum gates $\{\mathrm{CNOT}, R_Y, R_Z\}$, whose circuit structure is shown in Fig. 2. The expression of the proposed circuit is

$$U_i = (A \otimes B) C_1^2 (R_Z \otimes R_Y) C_2^1 C_1^2, \tag{1}$$

where A and B are the general single quantum bit unitary transformation (R_Z, R_Y, R_Z), C_j^i denotes a CNOT gate with i as the control bit and j as the controlled bit. As shown in Fig. 2, a U_i contains 8 quantum rotation gates as well as 3 CNOT gates.

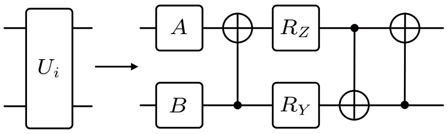

Fig. 2. The circuit structure of U_i.

2.2. Quantum pooling layer

In CNN, a pooling layer is usually added to the adjacent convolutional layers to reduce the feature mapping dimensions, to achieve nonlinearity, which in turn speeds up the computation, as well as prevents overfitting. Here, the quantum pooling layer is archived by quantum measurement on some quantum bits, followed by a single quantum bit unitary transformation V_i operated on each measured neighboring quantum bit, where i denotes the quantum pooling layer in which it is located.

According to the principle of deferred measurement,[32] a quantum measurement can be moved to the end of a quantum circuit when it is an intermediate step in the quantum circuit and the measurement result is a condition for controlling subsequent quantum gates. Furthermore, the controlling bit should have an arbitrary control state, while the controlled bit should apply an arbitrary single quantum bit unitary transformation based on the measured results.[23] Therefore, the measurement-control circuit for the quantum pooling layer can be designed as Fig. 3. At first, the equivalent circuit for the measurement-control circuit in Fig. 3(a) can be described as that in Fig. 3(b) based on the principle of the delayed measurement, then the implementation of the equivalent circuit of the measurement-control can be realized as that in Fig. 3(c).

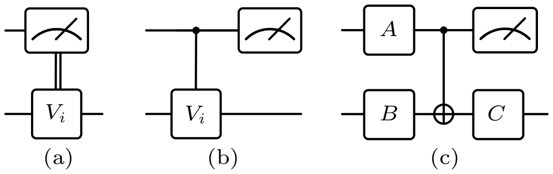

Fig. 3. (a) The structure of the measurement-control circuit. (b) The equivalent circuit is based on the principle of delayed measurement. (c) The implementation of the equivalent circuit of the measurement-control circuit, where A and B are all the universal single quantum bit unitary transformations. C consists of R_Z and R_Y gates in sequence.

2.3. Quantum fully connected layer

It is shown that the quantum circuit has reduced quantum bits after applying several quantum convolutional layers and quantum pooling layers, alternatively. When the system size is relatively small, a quantum fully connected (QFC) layer can be applied on the remaining qubits to perform classification on the extracted features. Here, the strongly entangling circuits proposed in Ref. [33] is adopted as the initial quantum circuit for QFC, who is comprised of multiple quantum circuit units, and each quantum circuit unit consists of several quantum rotation gates and CNOT gates. After removing the redundant parameters, figure 4 demonstrates one quantum circuit unit for QFC, who has 4 quantum rotation gates and 2 CNOT gates.

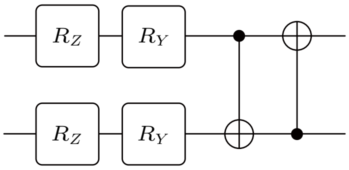

Fig. 4. The structure of a quantum circuit for the quantum fully connected (QFC) layer.

In a quantum circuit unit, each single qubit is typically composed of a sequence of rotations, represented by

$$G_j = R_y(\beta)R_z(\gamma), \tag{2}$$

where β, γ are the parameterized angles that are optimized during training. G_j are then defined as

$$G_j(\beta,\gamma) = \begin{bmatrix} \exp(-\mathrm{i}\frac{\gamma}{2})\cos\left(\frac{\beta}{2}\right) & -\exp(\mathrm{i}\frac{\gamma}{2})\sin\left(\frac{\beta}{2}\right) \\ \exp\left(-\mathrm{i}\frac{\gamma}{2}\right)\sin(\frac{\beta}{2}) & \exp(\mathrm{i}\frac{\gamma}{2})\cos\left(\frac{\beta}{2}\right) \end{bmatrix}. \tag{3}$$

With the followed CNOT gate, the two-qubit product states can then be transformed into an entangled state. For a two-qubit gate C (G), where G is a single-qubit gate applied to the target qubit b, the action between control qubit a and target qubit b can be written as

$$C_a(G_b)\left|x\right\rangle\left|y\right\rangle = \left|x\right\rangle \otimes G_b^x \left|y\right\rangle. \tag{4}$$

Then, one quantum circuit unit, denoted as B, can be expressed as

$$B = \prod_{k=0}^{n-1} R_k^Y C_{ck}(P_{tk}) \prod_{j=0}^{n-1} G_j, \tag{5}$$

where R_k^Y is a Y-axis rotation on qubit $\boldsymbol{k}$, $C_{ck}(P_{tk})$ is a controlled phase gate with a phase shift P_{tk}, and G_j is a single-qubit gate represented by the sequence $R_y(\beta)R_z(\gamma)$. It is observed that such quantum circuit unit is capable of unentangling almost all the inside qubits, so that one can measure the values of the qubits.

With the measurement, one can obtain the expectation value, and then get the classification result after the processing on the expectation value. For example, if the classification is a binary task and the expectation values p_1 and p_2 are obtained by the quantum measurements, then the classification result y can be described as

$$y = \begin{cases} 0, & p_1 \geq p_2, \\ 1, & p_1 < p_2. \end{cases} \tag{6}$$

If the classification is an M-ary task, and the expectation values p_1, p_2, ..., p_M are obtained via quantum measurements, then the classification outcome y can be described as

$$y = \begin{cases} 0, & p_1 = \max(p_1, p_2, \ldots, p_M), \\ 1, & p_2 = \max(p_1, p_2, \ldots, p_M), \\ \ldots, & \\ M-1, & p_M = \max(p_1, p_2, \ldots, p_M). \end{cases} \tag{7}$$

2.4. Optimization in VQDA

Since the unsupervised DA is implemented in the proposed VQDA, the labeled source domain training samples go through QFC1 and QFC2 to get the predicted labels and the predicted domains during the forward propagation, while the unlabelled target domain training samples only go through QFC2 to get the predicted domain. At the same time, the parameters in the quantum circuit are optimized by using the optimization method proposed in Ref. [11]. That is, the gradient from QFC2 should be subtracted, instead of being added, from the feature extractor during the gradient backward propagation, so that the GRM is designed between the last quantum pooling layer and QFC2. The gradient is multiplied by -1 when the gradient from QFC2 passes through this module. Then, the optimization of VQDA can be described as

$$\begin{aligned} & E\left(\theta_{\mathrm{cp}}, \theta_{\mathrm{QFC1}}, \theta_{\mathrm{QFC2}}\right) \\ = & \frac{1}{n}\sum_{i=1}^{n} L_{\mathrm{QFC1}}^{i}\left(\theta_{\mathrm{cp}}, \theta_{\mathrm{QFC1}}\right) \\ & -\frac{\lambda}{n}\sum_{i=1}^{n} L_{\mathrm{QFC2}}^{i}\left(\theta_{\mathrm{cp}}, \theta_{\mathrm{QFC2}}\right) \\ & -\frac{\lambda}{N-n}\sum_{i=n+1}^{N} L_{\mathrm{QFC2}}^{i}\left(\theta_{\mathrm{cp}}, \theta_{\mathrm{QFC2}}\right), \end{aligned} \tag{8}$$

where n samples are from the source domain, $N-n$ samples are from the target domain, L^i_{QFC1} denotes the i-th sample loss on QFC1, L^i_{QFC2} denotes the i-th sample loss on QFC2, θ_{QFC1}, θ_{QFC2}, and θ_{cp} are the parameters in QFC1, QFC2, the quantum convolution layers, and the quantum pooling layers, respectively, λ is the domain adaptation factor.

Moreover, the rules for updating the parameters in the quantum circuit are

$$\begin{aligned}
\theta_{\mathrm{cp}} &\leftarrow \theta_{\mathrm{cp}} - \mu\left(\frac{\partial L^i_{\mathrm{QFC1}}}{\partial\theta_{\mathrm{cp}}} - \lambda\frac{\partial L^i_{\mathrm{QFC2}}}{\partial\theta_{\mathrm{cp}}}\right),\\
\theta_{\mathrm{QFC1}} &\leftarrow \theta_{\mathrm{QFC1}} - \mu\frac{\partial L^i_{\mathrm{QFC1}}}{\partial\theta_{\mathrm{QFC1}}},\\
\theta_{\mathrm{QFC2}} &\leftarrow \theta_{\mathrm{QFC2}} - \lambda\mu\frac{\partial L^i_{\mathrm{QFC2}}}{\partial\theta_{\mathrm{QFC2}}},
\end{aligned} \tag{9}$$

where μ is the learning rate, and the gradient of the parameters can be obtained by the parameter shifting method.[33] Algorithm 1 shows the pseudo-codes of the optimization method in VQDA.

It is shown that the 'Feature Extractor' plays an important role in the designed VQDA, which is comprised of several 'Quantum Convolutional Layers' and 'Quantum Pooling Layers' in hierarchy. Importantly, for any used gates U_i and V_i, they can be decomposed by a collection of the basic quantum gates $\{\mathrm{CNOT}, R_X, R_Z\}$. Therefore, optimizing the quantum circuit for the 'Feature Extractor' is necessary. That is, the redundant basic quantum gates can be removed from the 'Feature Extractor' and 'QFC' while keeping its expressive power.

The redundant quantum gates in PQC can be removed by using Algorithm 2 (see details in Appendix A). It is observed that the parameters with non-zero amplitude in the eigenvector can be deleted from the PQC without changing its expression ability. The algorithm removes one redundant gate and removes the corresponding entries in the quantum Fisher information metric (QFI), and then recalculates the eigenvector of the QFI. These steps are repeated until no redundant gates are left. Here, three optimization methods are designed for Algorithm 2, named 'Entire', 'Partial Active', and 'Partial Inactive'. The 'Entire' is an overall optimization to optimize the whole quantum circuit, the 'Partial Active' involves optimizing modular convolutional kernels and pooling kernels separately and removing quantum gates, the expressive power is affected by random parameters of the quantum initial state, and the 'Partial Inactive' preserves potential quantum gates with expressive power.

Algorithm 1: The optimization in VQDA

```
input:  Source domain sample $S=\{(|x_i\rangle, y_i, d_s)\}_{i=1}^{n}$, target domain sample $T=\{(|x_j\rangle, d_T)\}_{j=1}^{n'}$;
        iteration number $E$; batch size $s$; learning rate $\mu$; domain adaptation factor $\left\{\lambda(t)_{t=0}^{E-1}\right\}$; the
        loss function for the labelled features $L_{\mathrm{QFC1}}$; the loss function for the domain $L_{\mathrm{QFC2}}$
output: Optimized parameters in VQDA $\theta^* = \{\theta_{\mathrm{cp}}, \theta_{\mathrm{QFC1}}, \theta_{\mathrm{QFC2}}\}$

Randomly initialize the parameter $\theta^*$ in $[0, 2\pi]$;
Initialize data length $l = \min(\lfloor n/s \rfloor, \lfloor n'/s \rfloor)$;
for $t$ from 0 to $E-1$ do
    for $a$ from 1 to $l$ do
        Randomly select $s$ samples in $S$ to input into the circuit;
        Calculate the gradient $\nabla^S_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC1}}$, $\nabla^S_{\theta^{(t)}_{\mathrm{QFC1}}} L_{\mathrm{QFC1}}$, $\nabla^S_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC2}}$ and $\nabla^S_{\theta^{(t)}_{\mathrm{QFC2}}} L_{\mathrm{QFC2}}$;
        Randomly select $s$ samples in $T$ to input into the circuit only through QFC2;
        Calculate the gradient $\nabla^T_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC2}}$ and $\nabla^T_{\theta^{(t)}_{\mathrm{QFC2}}} L_{\mathrm{QFC2}}$;
        Update the parameters:
        $\theta^{(t+1)}_{\mathrm{cp}} \leftarrow \theta^{(t)}_{\mathrm{cp}} - \mu\left(\nabla^S_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC1}} - \lambda(t)\left(\nabla^S_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC2}} + \nabla^T_{\theta^{(t)}_{\mathrm{cp}}} L_{\mathrm{QFC2}}\right)\right)$
        $\theta^{(t+1)}_{\mathrm{QFC1}} \leftarrow \theta^{(t)}_{\mathrm{QFC1}} - \mu\nabla^S_{\theta^{(t)}_{\mathrm{QFC1}}} L_{\mathrm{QFC1}}$
        $\theta^{(t+1)}_{\mathrm{QFC2}} \leftarrow \theta^{(t)}_{\mathrm{QFC2}} - \mu\lambda(t)\left(\nabla^S_{\theta^{(t)}_{\mathrm{QFC2}}} L_{\mathrm{QFC2}} + \nabla^T_{\theta^{(t)}_{\mathrm{QFC2}}} L_{\mathrm{QFC2}}\right)$;
    end
end
Output the parameters $\theta^*$;
```

It is noticed that Algorithm 2 is optimized for the fixed parameters of quantum initial state, however, in the proposed VQDA, the initial state i.e. image data is alterable during the training phase, therefore, Algorithm 2 is modified to be suitable for initial adjustable parameters. Simultaneously, a random quantum initial state is set at first for the optimization algorithm and each type of the optimization method is employed for many times. All quantum circuits in the simulations are the optimization results by using the 'Partial Inactive' optimization method.

3. Numerical simulations and demonstrations on IBM Q platform

In this section, we verify the performance of the proposed VQDA through numerical simulations and demonstrations on the IBM Q platform for DA tasks in the Digit-5 dataset. For DA in gray images, MNIST has 5000 and 1600 samples for the training set and test set, respectively; while USPS has 1600 and 600 samples for the training set and test set, the samples from both domains are downsampled to a size of 16 × 16 to match 8 quantum bits VQDA model. For DA in color images, MNIST-M, SYNDigits, and SVHN all contain about 5000 training sets and 1200 test sets, and all the samples are downsampled to a size of 3-channels 16×16 to match the inputs of the 10 quantum bits VQDA model. The cross-entropy is chosen as the loss function for both QFC1 and QFC2 and the optimizer Adam is adopted. The IBM Q quantum device is accessed through Qiskit. The hardware environment used for the numerical simulations is an Intel i9-13900HX @5.40 GHz CPU, an NVIDIA GeForce RTX 4070 (8 GB) GPU, and 32 G@5600 MHz of RAM. All the simulation source codes are developed using Python and the Pennylane software package.

The samples in MNIST, MNIST-M, USPS, SYNDigits, and SVHN are the digit numbers from 0 to 9. In the simulation, they are categorized into five classes, such as '0' and '9', '1' and '8', '2' and '7', '3' and '6', and '4' and '5'. The proposed VQDA can be considered as the quantum improved version of DA discussed in Ref. [11]. Therefore, we list the results with the proposed VQDA, together with that of DANN in Ref. [11] for comparisons.

First, we present the demonstration of the proposed VQDA on MNIST → USPS, where the images in both MNIST and USPS datasets are without backgrounds. The VQDA model used here contains two quantum convolutional layers and two quantum pooling layers, and there is a total of 96 adjustable parameters in VQDA.

Figure 5 shows the probability histogram of V_2 and QFC1 in Fig. 1 for '3' and '6' in the USPS database on the IBM Q platform with 10000 shots, where figure 5(a) shows the measurement results after V_2 for '3' and '6', figure 5(b) presents the measurement results after QFC1 for '3' and '6'. In this case, there are two quantum measurements both at V_2 and QFC1, so that the measurement results are both 00, 01, 10, and 11. Table 1 further lists the expectation values after V_2 and QFC1 in Fig. 5, where the expectation value is calculated by $P_{|0\rangle} - P_{|1\rangle}$, $P_{|0\rangle}$ represents the probability of measurement outcome $|0\rangle$, while $P_{|1\rangle}$ is for the outcome $|1\rangle$. According to the strategy in Eq. (6), the results show that '3' and '6' are predicted correctly by the proposed VQDA in the '3' and '6' classifications.

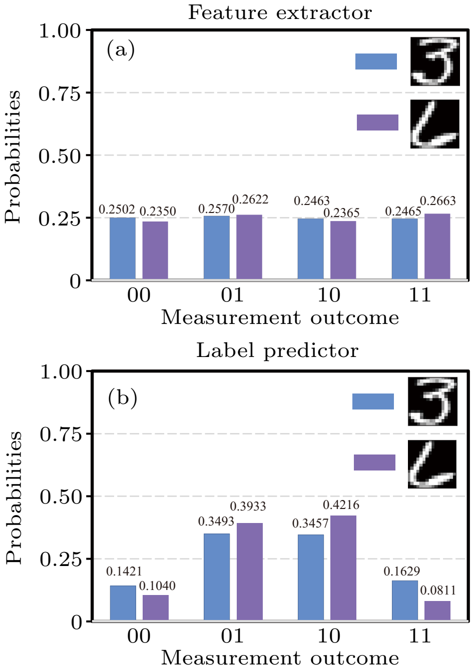

Fig. 5. The probability histogram after V_2 and QFC1 in Fig. 1 in USPS database on the IBM Q platform with 10000 shots: (a) measurement results after V_2 for '3' and '6', (b) measurement results after QFC1 for '3' and '6'.

Table 1. The measured expectation values after V_2 and QFC1 in Fig. 1 on the IBM Q platform in USPS.

Image	E_{V_2}	E_{QCF1}	Result
'3'	0.0144 –0.0070	–0.0172 –0.0244	3
'6'	–0.0056 –0.0570	–0.0054 0.0512	6

Note that, for the output of 'Feature Extractor', the expectation values are evaluated through Z measurement, and the obtained expectation values are then input to QFC1 in the form of angles, while the expectation values are computed through X measurement for the output of 'Label Predictor'. When X measurement is selected, the Hadamard gate should be added before the corresponding quantum bit measurement operation.

Figure 6 shows the probability histogram of V_2 and QFC1 in Fig. 1 for '3' and '6' in the USPS database on the IBM Q platform with 10000 shots for six samples, where figure 6(a) shows the measurement results after V_2 for the three '3', figure 6(b) shows the measurement results after V_2 for the three '6', figure 6(c) presents the measurement results after QFC1 for the three '3', and figure 6(d) gives the measurement results after QFC1 for the three ‘6’. The three images ‘3’ and ‘6’, labeled as ‘Blue’, ‘Orange’, and ‘Gray’, are randomly selected from the USPS database.

It is noted that the measurement outcomes labeled here are listed from right to left. Figure 6(a) (figure 6(b)) shows that for different three ‘3’ (‘6’), the probability distributions after the ‘Feature Extractor’ are almost the same against the measurement outcomes. It hints that the features can be extracted by the designed ‘Feature extractor’. Furthermore, the results in Fig. 6(c) (Fig. 6(d)) demonstrate the predicted labels for the three ‘3’ (‘6’) after QFC1. For the three ‘3’, the probability of 01 outcomes is 0.8074, 0.7899, and 0.8824, respectively, while the probabilities of 10 outcomes are 0.8144, 0.8405, and 0.8312, respectively, for the three ‘6’. It is shown that the labels for the three ‘3’ (‘6’) are predicted correctly by the proposed VQDA. Table 2 lists the expectation values after V_2 and QFC1 in Fig. 6. It is shown that all three ‘3’ (‘6’) are predicted correctly by the proposed VQDA in the ‘3’ and ‘6’ classifications.

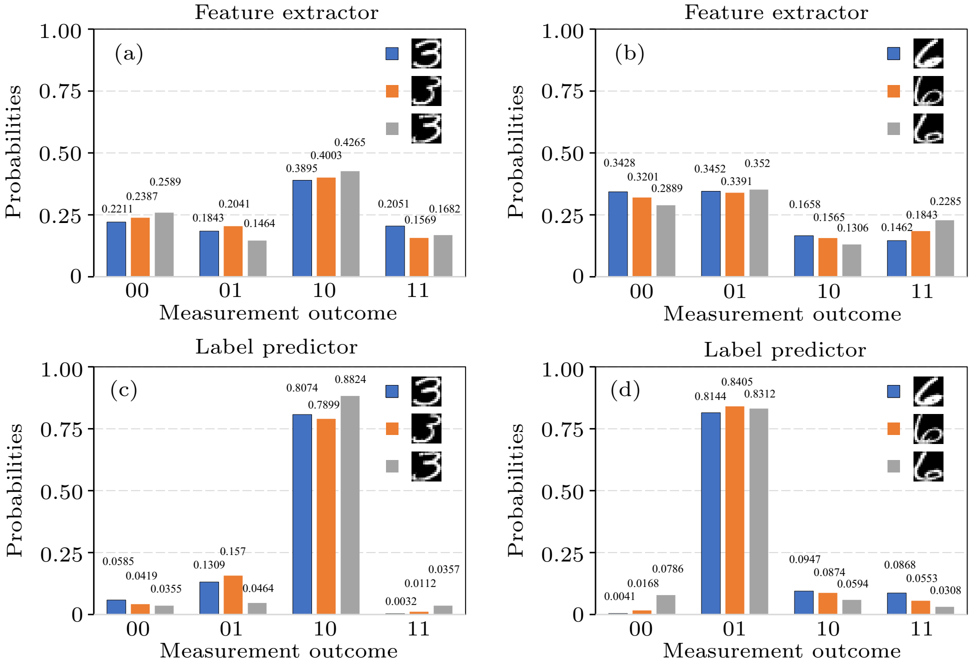

Fig. 6. The probability histogram after V_2 and QFC1 in Fig. 1 in USPS database on the IBM Q platform with 10000 shots: (a) measurement results after V_2 for ‘3’, (b) measurement results after V_2 for ‘6’, (c) measurement results after QFC1 for ‘3’, (d) measurement results after QFC1 for ‘6’.

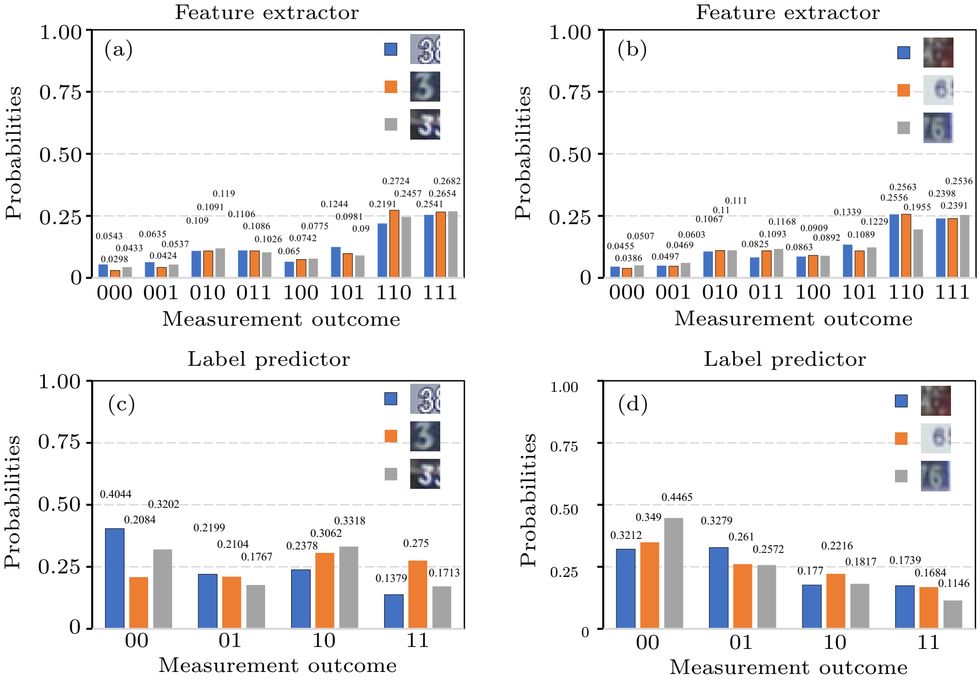

Fig. 7. The probability histogram of V_2 for ‘3’ and ‘6’ images in the SVHN database on the IBM Q platform (VQDA1) with 10000 shots: (a) measurement results after V_2 for ‘3’, (b) measurement results after V_2 for ‘6’, (c) measurement results after QFC1 for ‘3’, (d) measurement results after QFC1 for ‘6’.

Table 2. The measured expectation values after V_2 and QFC1 in Fig. 1 on the IBM Q platform in USPS.

Image	E_{V_2}-'3'	E_{QCF1}-'3'	Result	E_{V_2}-'6'	E_{QCF1}-'6'	Result
'Blue'	0.2212 –0.1892	0.7318 –0.6212	3	0.0172 0.3760	–0.8024 0.6370	6
'Orange'	0.2780 –0.1144	0.6636 –0.6022	3	–0.0468 0.3184	–0.7916 0.7146	6
'Gray'	0.3708 –0.1894	0.8358 –0.8362	3	–0.1610 0.2818	–0.7240 0.8196	6

Lastly, we demonstrate VQDA on SYNDigits → SVHN, where the images in both domains are with color backgrounds. VQDA uses 10 quantum bits to encode the sample information, and containing two quantum convolutional layers and two quantum pooling layers. There are 120 adjustable parameters inside.

Similarly, figure 7 demonstrates the probability histogram by using VQDA for the 3–6 classification task in SVHN on the IBM Q platform with 10000 shots, where the image '3' and '6' are with a color background. For the 10 quantum bits circuit, the 'Feature extractor' has 3 quantum measurements. Figures 7(a) and 7(b) show the measurement outcome probability distributions, while figures 7(c) and 7(d) show the predicted labels for the three '3' ('6'). In the same way, the expectation values of each quantum bit after V_2 and QFC1 are listed in Table 3. It is indicated that all three colored backgrounds '3' ('6') are classified correctly by the proposed VQDA in the 3–6 classification task.

Furthermore, we compare the performance of the proposed VQDA with its classical counterpart DANN. At first, we test them between the gray-scale images and the results are shown in Table 4, where VQDA/VQDAo represents VQDA with/without the quantum circuits optimization, DANN1 is the classical counterpart DANN with the same parameter size of VQDA, while DANN2 is the classical counterpart DANN with a similar average classification accuracy, 'N' denotes the parameters used in DA, and 'Average' is the average of the classification accuracy of the domain adaptation task. The results show that VQDA obtains almost all higher classification accuracy for the ten tasks. For the domain adaptation task between the MNIST dataset and the USPS dataset, under the same parameter size, the average classification accuracy is 91.46% by using VQDA, which is 15% higher than that of DANN1. When the accuracy is similar to DANN2 (90.87%), the parameter size is reduced to 0.393 of DANN2. In comparison with VQDAo, VQDA improves 3% in average classification accuracy and reduces 27.3% parameters. It hints that the proposed VQDA scheme is feasible.

Furthermore, we test the proposed VQDA scheme between gray-scale images and color images, and the results are shown in Table 5. The results show that the proposed VQDA has all higher classification accuracy except 1–8 DA tasks in MNIST-M → MNIST. For the domain adaptation task between

Table 3. The measured expectation values after V_2 and QFC1 in Fig. 1 on the IBM Q platform in SVHN.

Image	E_{V_2}-'3'	E_{QCF1}-'3'	Result	E_{V_2}-'6'	E_{QCF1}-'6'	Result
'Blue'	–0.1052 –0.3856 –0.3252	0.2844 0.2486	3	–0.0118 –0.3692 –0.4312	–0.0036 0.2982	6
'Orange'	–0.0290 –0.5110 –0.4202	0.0292 –0.1624	3	–0.0084 –0.4294 –0.3904	0.1412 0.2200	6
'Gray'	–0.0290 –0.4710 –0.3628	0.3040 –0.0062	3	–0.1072 –0.3538 –0.3224	0.2564 0.4074	6

Table 4. The classification accuracy (%) by the proposed VQDA and its classical counterpart DANN for test dataset in MNIST → USPS and USPS → MNIST (gray images).

Model	N	MNIST → USPS					USPS → MNIST					Average
		0–9	1–8	2–7	3–6	4–5	0–9	1–8	2–7	3–6	4–5	
VQDA	**96**	**92.83**	**82.83**	**98.17**	**97.83**	**88.67**	**97.63**	**81.19**	**96.06**	**98.06**	81.38	**91.46**
DANN1	102	73.50	76.50	69.00	68.17	78.67	87.38	79.94	76.38	66.06	**91.25**	76.68
VQDAo	132	83.67	57.50	97.83	99.17	92.83	90.88	78.63	95.63	98.56	88.56	88.33
DANN2	244	95.00	83.50	93.67	76.00	84.17	94.19	97.81	92.06	95.38	96.94	90.87

Table 5. The classification accuracy (%) by VQDA and DANN for test dataset in MNIST → MNIST-M and MNIST-M → MNIST (gray images /color images).

Model	*N*	MNIST → MNIST-M					MNIST-M → MNIST					Average
		0–9	1–8	2–7	3–6	4–5	0–9	1–8	2–7	3–6	4–5	
VQDA	**120**	**88.88**	**88.08**	**85.11**	**83.62**	**88.98**	**95.75**	94.56	**95.19**	**95.38**	**96.06**	**91.16**
DANN1	127	74.63	66.72	74.72	67.46	68.59	92.81	**97.56**	90.06	90.88	95.06	81.85
VQDAo	165	90.00	88.45	84.41	87.41	88.98	96.00	96.44	95.06	96.56	95.63	91.89
DANN2	1249	77.63	83.07	81.48	80.67	85.27	99.25	99.63	97.63	99.69	99.94	90.42

Table 6. The classification accuracy (%) by using VQDA and DANN for the test dataset in color images.

Model	*N*	SYNDigits → SVHN					SVHN → SYNDigits					Average
		0–9	1–8	2–7	3–6	4–5	0–9	1–8	2–7	3–6	4–5	
VQDA	**120**	66.64	73.82	71.26	**74.49**	75.76	72.46	**56.53**	52.14	**67.90**	82.51	69.35
DANN1	127	**74.41**	**78.71**	**74.23**	62.49	**77.10**	**76.72**	54.47	**53.48**	61.98	**83.13**	**69.67**
VQDAo	165	69.59	90.39	76.64	74.13	79.26	69.71	56.31	61.83	70.30	81.61	72.98
DANN2	146	67.15	83.12	77.16	72.52	66.71	78.63	53.66	57.76	60.02	81.39	69.81
Model	*N*	SYNDigits → MNIST-M					MNIST-M → SYNDigits					Average
		0–9	1–8	2–7	3–6	4–5	0–9	1–8	2–7	3–6	4–5	
VQDA	**120**	**85.06**	**71.10**	59.77	**76.30**	**73.95**	59.73	64.77	55.87	63.87	**82.06**	**69.25**
DANN1	127	70.75	69.99	**69.79**	67.23	63.70	**63.38**	**66.23**	**64.11**	**63.99**	59.98	65.91
VQDAo	165	79.38	73.10	69.52	78.40	77.78	61.92	69.86	57.71	64.49	75.50	70.77
DANN2	188	71.81	77.53	72.77	67.23	72.42	62.42	69.11	81.30	64.94	56.61	69.61
Model	*N*	SVHN → MNIST-M					MNIST-M→SVHN					Average
		0–9	1–8	2–7	3–6	4–5	0–9	1–8	2–7	3–6	4–5	
VQDA	**120**	**60.19**	**66.61**	**50.62**	**63.10**	**77.49**	59.29	**74.38**	64.32	**62.01**	**64.19**	**64.22**
DANN1	127	56.94	60.18	50.51	51.81	54.98	**59.46**	54.09	**67.72**	61.89	55.14	57.27
VQDAo	165	68.24	61.06	76.64	53.00	75.79	63.34	74.25	66.26	64.70	63.03	66.63
DANN2	168	68.13	71.36	65.84	66.21	68.65	60.56	60.62	51.82	59.68	63.08	63.59

MNIST and MNIST-M, under the same parameter size, the average classification accuracy of using VQDA is 91.16%, which is 9.31% higher than that of DANN1 (classification accuracy 81.85%), whereas DANN2 has 90.42% average classification accuracy, needs 10.4 times the parameters. In comparison with VQDAo, VQDA has reduced 27.3% parameters while keeping the same classification accuracy.

Finally, we test the proposed VQDA between color images and the results are shown in Table 6. There are three kinds of domain adaptation here, SYNDigits → SVHN (SVHN→ SYNDigits), SYNDigits → MNIST-M(MNIST-M→ SYNDigits), and SVHN→MNIST-M(MNIST-M →SVHN). The results show that VQDA and DANN1 can obtain approximate classification accuracies between SYNDigits and SVHN. At the same time, when the classification accuracy in DANN2 is 69.81%, the parameter scale is increased by 17.8%. For the domain adaptation task between SYNDigits and MNIST-M, the average classification accuracy under the same parameters is 69.25% by using VQDA, which is 3.34% higher than that of DANN1 (65.91%), whereas DANN2 requires 1.55 times the parameters. Compared to VQDAo, VQDA reduces the parameters by 27.3% while maintaining similar classification accuracy. For the domain adaptation task between SVHN and MNIST-M, the average classification accuracy by using VQDA is 64.22%, which is 6.95% higher than that of DANN1 (57.27%). The parameters are 0.714 of DANN2 (63.59%) when the classification accuracy is similar. Compared to VQDAo, VQDA reduces the parameters by 27.3% while keeping the approximate classification accuracy.

4. Conclusion

In the paper, we propose a novel QCNN-based DA, named VQDA, to address the discrepancy problem between the source domain and the target domain. The availability is validated by Qiskit on IBM Q platform and numerical simulations on the local computer in Digit-5 dataset. To compare with its classical counterpart and VQDA without quantum circuit optimization, the results by using DANN1, DANN2, and VQDAo are also presented. The simulation results show that VQDA with 8 quantum bits has 91.46% average classification accuracy for DA task between MNIST → USPS (USPS →MNIST), which is 14.78% improvement over DANN1, 3% over VQDAo, and has reduced 60.7% parameters over DANN2, 27.3% over VQDAo. For gray-scale and color images, VQDA with 10 quantum bits has achieved 91.16% average classification accuracy for DA task, which is 9.31% improvement in comparison with DANN1, and has reduced 90.4% parameters over DANN2, 27.3% over VQDAo. For color images, VQDA and DANN1 perform closely in DA task between SYNDigits→ SVHN (SVHN→ SYNDigits). It has an average classification accuracy of 69.25% on the DA task between SYNDigits →MNIST-M (MNIST-M→ SYNDigits), which is 3.34% better than DANN1 and 36.17% less parameters than DANN2, and 27.3% less than VQDAo. For the DA task between SVHN → MNIST-M (MNIST-M → SVHN), VQDA has an average classification accuracy of 64.22%, which is a 6.95% improvement over DANN1 and a reduced parameterizations of 28.57% over DANN2 and 27.3% over VQDAo. For different types of classification tasks, VQDA can achieve a 9.14% improvement in average recognition accuracy compared to its classical counterpart with the same parameter scale and can reduce 43% of parameters when both quantum and classical DA methods have similar classification accuracies. It hints that the proposed VQDA is an effective domain adaption, and it is a promising way for the application of QML in the NISQ era.

Appendix A: Optimizing quantum circuits

The effective quantum dimension G_C and the parameter dimension D_C are the two quantitative measures of parameterized quantum circuit (PQC) capacity, where G_C is a local measurement of the quantized state space, which can be accessed by locally perturbing the parameters of the PQC, while D_C is a measurement of the total number of independent parameters that can be expressed by the quantum state defined by PQC. Both of them can be derived from the quantum geometry of PQC through the quantum Fisher information metric (QFI) $\mathcal{F}(\theta)$.

Algorithm 2: Prune PQC of redundant parameters

```
input:  PQC $C$, QFI $\mathcal{F}^C(\theta_{\text{random}})$, number of parameters $N^C_{\text{param}}$, $D_C < N^C_{\text{param}}$, empty set $\mathcal{K} = \{\}$
output: Pruned PQC $C_{\text{pruned}}$ with $N^{C_{\text{pruned}}}_{\text{parameters}} \approx D_C$

do
    Get eigenvalues $\lambda^{(i)}$ of $\mathcal{F}^C(\theta_{\text{random}})$ sorted in ascending order, eigenvectors $\alpha^{(i)}$ and rank $r$;
    Calculate $\beta_j = \sum_{i=1}^{N_{\text{param}}-r} \left|\alpha_j^{(i)}\right|^2$ where $\alpha^{(i)}$ denotes the $i$th eigenvector with corresponding
      eigenvalue $\lambda^{(i)} = 0$ ;
    Pick largest index $k$ such that $\beta_k \neq 0$;
    Update $\mathcal{F}^C(\theta_{\text{random}})$ by removing row $k$ and column $k$;
    Add $k$ to set $\mathcal{K}$;
while $\mathcal{F}^C(\theta_{\text{random}})$ has nonzero eigenvalues;
Removing parameters corresponding to set $\mathcal{K}$ from $C$ gives pruned PQC $C_{\text{pruned}}$;
```

Acknowledgement

Project supported by the National Natural Science Foundation of China (Grant Nos. 62375140 and 61871234).

References

[1] Fan C, Liu P, Xiao T, Zhao W and Tang X L 2020 *Acta Autom. Sin.* **46** 515

[2] Wang M and Deng W 2018 *Neurocomputing* **312** 135

[3] Tan C, Sun F, Kong T, Zhang W, Yang C and Liu C 2018 *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks*, October 4–7, 2018, Rhodes, Greece, p. 270

[4] Long M, Cao Y, Wang J and Jordan M 2015 *Proceedings of the 32nd International Conference on Machine Learning*, July 7–9, 2015, Lille, France, p. 97

[5] Rozantsev A, Salzmann M and Fua P 2018 *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41** 801

[6] Gong R, Li W, Chen Y and Gool L V 2019 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 15–20, 2019, Long Beach, USA, p. 2477

[7] Ma X, Zhang T and Xu C 2019 2019 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 15–20, 2019, Long Beach, USA, p. 8258

[8] Zhuang F, Cheng X, Luo P, Pan S J and He Q 2015 *Proceedings of the 24th International Conference on Artificial Intelligence*, July 25, 2015, Buenos Aires, Argentina, p. 4119

[9] Kim T, Cha M, Kim H, Lee J K and Kim J 2017 *Proceedings of the 34th International Conference on Machine Learning - Volume* 70, August 6, 2017, Sydney, NSW, Australia, p. 1857

[10] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2020 *Commun. ACM* **63** 139

[11] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, March M and Lempitsky V 2016 *The Journal of Machine Learning Research* **17** 2096

[12] Pan S J, Wan L C, Liu H L, Wu Y S, Qin S J, Wen Q Y and Gao F 2022 *Chin. Phys. B* **31** 060304

[13] He X, Zhang A and Zhao S 2022 *Quantum Inf. Process.* **21** 86

[14] Cerezo M, Verdon G, Huang H Y, Cincio L and Coles P J 2022 *Nat. Comput. Sci.* **2** 567

[15] Hou Y Y, Li J, Chen X B and Tian Y 2022 *Chin. Phys. B* **31** 030304

[16] Zhang A, He X and Zhao S 2022 *Quantum Inf. Process.* **21** 358

[17] Schuld M and Killoran N 2019 *Phys. Rev. Lett.* **122** 040504

[18] Wu Y, Wu C, Zhang A and Zhao S 2023 *Quantum Inf. Process.* **22** 261

[19] Guo X Y, Li S S, Xiao X, Xiang Z C, Ge Z Y, Li H K, Song P T, Peng Y, Wang Z, Xu K and Zhang P 2023 *Chin. Phys. B* **32** 010307

[20] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O?brien J L 2014 *Nat. Commun.* **5** 4213

[21] Chen R Y L, Zhao B C, Song Z X, Zhao X Q, Wang K and Wang X 2021 *Acta Phys. Sin.* **70** 210302 (in Chinese)

[22] Zhang A, Wang K, Wu Y and Zhao S M 2023 *Chin. Phys. B* **32** 100308

[23] Zhang A and Zhao S 2023 *Quantum Inf. Process.* **22** 283

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

[25] Hur T, Kim L and Park D K 2022 *Quantum Machine Intelligence* **4** 3

[26] Lü Y, Gao Q, Lü J, Ogorzałek M and Zheng J 2021 2021 40*th Chinese Control Conference (CCC)*, July 26–28, 2021, Shanghai, China, p. 6329

[27] Yang C H H, Qi J, Chen S Y C, Chen P Y, Siniscalchi S M, Ma X and Lee C H 2021 *ICASSP* 2021-2021 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, June 6–11, 2021, Toronto, ON, Canada, p. 6523

[28] Wei S, Chen Y, Zhou Z and Long G 2022 *AAPPS Bull.* **32** 1

[29] Yao X W, Wang H, Liao Z, Chen M C, Pan J, Li J, Zhang K, Lin X, Wang Z, Luo Z and Zheng W 2017 *Phys. Rev. X* **7** 031041

[30] Zhang K, Hsieh M H, Liu L and Tao D 2020 arXiv:2011.06258 [quant-ph]

[31] Shende V V, Markov I L and Bullock S S 2004 *Phys. Rev. A* **69** 062321

[32] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information*, 10th Anniversary Edition. (Cambridge: Cambridge University Press) pp. 186–187

[33] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 *Phys. Rev. A* **101** 032308

[34] Amari S *2016 Information Geometry and Its Applications* (Springer Tokyo) Vol. 194