

Deep learning framework for time series classification based on multiple imaging and hybrid quantum neural networks

Jianshe Xie(谢建设) and Yumin Dong(董玉民)
Citation: Chin. Phys. B, 2023, 32 (**12**): 120302. DOI: 10.1088/1674-1056/accb45
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

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

A backdoor attack against quantum neural networks with limited information

Chen-Yi Huang(黄晨猗) and Shi-Bin Zhang(张仕斌)
Chin. Phys. B, 2023, 32 (**10**): 100306. DOI: 10.1088/1674-1056/acd8ab

Thermodynamic limit of the *XXZ* central spin model with an arbitrary central magnetic field

Fa-Kai Wen(温发楷) and Kun Hao(郝昆)
Chin. Phys. B, 2023, 32 (**9**): 090307. DOI: 10.1088/1674-1056/acb426

Commensurate and incommensurate Haldane phases for a spin-1 bilinear-biquadratic model

Yan-Wei Dai(代艳伟), Ai-Min Chen(陈爱民), Xi-Jing Liu(刘希婧), and Yao-Heng Su(苏耀恒)
Chin. Phys. B, 2023, 32 (**9**): 090304. DOI: 10.1088/1674-1056/acac0c

Degenerate polarization entangled photon source based on a single Ti-diffusion lithium niobate waveguide in a polarization Sagnac interferometer

Yu Sun(孙宇), Chang-Wei Sun(孙昌伟), Wei Zhou(周唯), Ran Yang(杨然), Jia-Chen Duan(端家晨), Yan-Xiao Gong(龚彦晓), Ping Xu(徐平), and Shi-Ning Zhu(祝世宁)
Chin. Phys. B, 2023, 32 (**8**): 080308. DOI: 10.1088/1674-1056/acd8b3

Algorithm for evaluating distance-based entanglement measures

Yixuan Hu(胡奕轩), Ye-Chao Liu(刘烨超), and Jiangwei Shang(尚江伟)
Chin. Phys. B, 2023, 32 (**8**): 080307. DOI: 10.1088/1674-1056/acd5c5

Deep learning framework for time series classification based on multiple imaging and hybrid quantum neural networks

Jianshe Xie(谢建设) and Yumin Dong(董玉民)†

College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, *China*

(Received 20 January 2023; revised manuscript received 19 March 2023; accepted manuscript online 7 April 2023)

Time series classification (TSC) has attracted a lot of attention for time series data mining tasks and has been applied in various fields. With the success of deep learning (DL) in computer vision recognition, people are starting to use deep learning to tackle TSC tasks. Quantum neural networks (QNN) have recently demonstrated their superiority over traditional machine learning in methods such as image processing and natural language processing, but research using quantum neural networks to handle TSC tasks has not received enough attention. Therefore, we proposed a learning framework based on multiple imaging and hybrid QNN (MIHQNN) for TSC tasks. We investigate the possibility of converting 1D time series to 2D images and classifying the converted images using hybrid QNN. We explored the differences between MIHQNN based on single time series imaging and MIHQNN based on the fusion of multiple time series imaging. Four quantum circuits were also selected and designed to study the impact of quantum circuits on TSC tasks. We tested our method on several standard datasets and achieved significant results compared to several current TSC methods, demonstrating the effectiveness of MIHQNN. This research highlights the potential of applying quantum computing to TSC and provides the theoretical and experimental background for future research.

Keywords: quantum neural networks, time series classification, time-series images, feature fusion

PACS: 03.67.–a, 03.67.Ac, 03.67.Lx **DOI:** 10.1088/1674-1056/accb45

1. Introduction

A time series is a series of data points (measurements) with a natural time order. Many important real-world pattern recognition tasks involve time series analysis. Time series are prevalent in various fields, such as biology, astronomy, meteorology, finance, medicine, engineering, and others.[1] People use techniques such as classification, segmentation, and prediction of time series to exploit their potential value as much as possible. TSC, as a branch of time series analysis, has attracted a lot of attention in the field of data mining.

Time series data are highly volatile and uncertain, and traditional methods such as expert empirical consultation and multiple models perform poorly in classification. In recent years, a large number of algorithms have been developed for TSC, which can be specifically summarized as distance-based time series classification algorithms and feature-based classification algorithms. The most popular distance-based algorithms are the nearest neighbor (NN) classifier and distance function,[2] the k-nearest neighbor (k-NN)[3] and dynamic time warping (DTW).[4] In the feature-based algorithm, Nanopoulos uses mean, standard deviation, skewness, and kurtosis to represent and classify time series,[5] and Morchen *et al*. used features derived from wavelet and Fourier transforms of a series of time series datasets to classify time series.[6] Subsequently, Lin *et al*.[7] proposed the bag-of-words (BOW) method, which quantifies the extracted feature BOW and feeds it into the classifier as a feature attribute.

Recently, DL models have achieved high recognition rates in computer vision recognition. DL is practical and effective in generating features of raw data, which provides clues to improve the accuracy of TSC methods. The idea of converting time series to images and using them as input for deep learning has received a lot of attention. Convolutional neural networks (CNN), one of the most successful DL models, have been applied to solve complex problems in many fields. A CNN unifies the feature learning and classification parts in one model and learns them jointly. As a result, CNN has achieved remarkable success in image and video recognition tasks.

Quantum machine learning (QML)[8] combines classical machine learning techniques with quantum computing and is emerging as a powerful approach that allows quantum acceleration and improvement of classical machine learning algorithms. QML mainly uses parametric quantum circuits (PQC) for prediction and classification. Quantum circuits for machine learning problems are similar to classical neural networks (NN), so they are sometimes called QNN. The input layer of a QNN is called the embedded circuit and the hidden layer is called the variable quantum circuit layer, which together form the PQC. So far, several QML algorithms have been proposed and demonstrated, including quantum support vector machines,[9,10] quantum deep learning,[11–17] quantum Boltzmann machines (QBM),[18,19] and quantum generative adversarial learning.[20–22] Exploration of quantum computing has shown that quantum effects, such as superposition and entanglement, can improve the efficiency of training when solving certain problems.

Inspired by recent achievements in computer vision techniques and quantum machine learning, we propose a frame-

†Corresponding author. E-mail: dym@cqnu.edu.cn

© 2023 Chinese Physical Society and IOP Publishing Ltd

work for learning based on multiple imaging and hybrid quantum neural networks. Specifically, the time series datasets are first divided into two categories, one with a sufficient number of training samples and the other with an insufficient number of samples. For the former, we use recurrence plots (RP)[36] to convert the raw temporal data into 2D images. For the latter, we will use four time series imaging methods, RP, Markov transition field (MTF),[38] Gramian angular summation field (GASF),[25] and Gramian angular difference field (GADF),[25] to convert the time series into four 2D images. Then, the four images are fused into one image to strengthen the features and reduce the probability of overfitting in the training process caused by insufficient sample quantity. The resulting images will be trained and classified by combining QNN and classical neural networks. ResNet[24] is used as the feature extractor, and QNN is used as the classifier. In order to evaluate the influence of different quantum variational circuits on classification, we will use four kinds of quantum circuits as classifiers. The results show that compared with the three most recent best classification methods, our approach achieves the best performance on 9 of the 10 standard datasets in the UCR archive.[39] In datasets with a small number of samples, experiments showed that the method using four time series imaged and fused achieved the best performance on four of the standard datasets from the UCR archive compared to the method using a single time series imaged.

The main contributions of this paper are summarized in the following three points. First, for the TSC task, a model combining time-series imaging and hybrid quantum neural networks is proposed. Second, considering the insufficient sample size of the training data, four time-series imaging fusion methods are used to enhance the sample features and reduce the risk of overfitting. Third, to explore the impact of quantum circuits, four quantum circuits were designed and investigated for the TSC task.

The rest of this paper is organized as follows. In Section 2, we discuss the background and related work. We present our framework in Section 3, and give the experimental results of the proposed method on a standard dataset in Section 4. Finally, Section 5 presents the conclusion and perspectives.

2. Related work

In recent years, DL has made impressive progress in many areas, including speech recognition, image recognition, and natural language processing, which has motivated researchers to study DL in the TSC domain. It is only recently that deep learning methods such as multi-scale convolutional neural networks (MCNN),[23] fully convolutional networks (FCN),[24] and ResNet[24] have started to appear in TSC. Wang[25] proposed converting time series into images using GAF and MTF and applying CNN models for time series classification (GAF-MTF) in 2015. Subsequently, Hatami[26] *et al*. proposed the use of RP to convert time series into 2D texture images and then perform recognition operations using a deep CNN classifier (RPCNN). Chen[27] *et al*. investigated a relative position matrix to convert time series data into 2D images and construct an improved CNN architecture to classify the data (RPMCNN). Karim,[28] on the other hand, proposed solving the TSC problem with long short-term memory fully convolutional networks (LSTM-FCN). These approaches can be broadly classified into two categories: one relies on modified traditional CNN architectures and uses 1D time series signals as input, such as LSTM-FCN, while the other first converts the original time series into 2D images and then applies them to deep learning models, such as GAF-MTF, RPCNN, and RPMCNN.

Quantum computing (QC) on the other hand, as a new computing paradigm, is expected to be applied in several fields, including machine learning. The field of QC has demonstrated its important role in problems that are difficult to solve by their classical counterparts through quantum superiority.[29,30] The quantum algorithm for deep convolutional neural networks proposed by Kerenidis[31] implements a new frontier in image recognition and numerical simulations of MNIST dataset classification, demonstrating the efficiency of quantum computing. In Ref. [32], a hybrid quantum convolution neural network model is introduced to predict COVID-19 images. It fully exploits the advantages of quantum convolutional layers and CNN and demonstrates the possibility of hybrid quantum models for new crown image classification. Mari[33] proposed a migration learning model in hybrid classical–quantum neural networks to demonstrate the possibility of quantum migration learning. Mari used the ResNet deep learning model to process the samples classically and then used a variational quantum circuit to process the features. There are several classical feature extraction models to choose from, such as AlexNet,[34] ResNet,[24] DENSENET,[35] and VGGNet.[36] The feature extraction in our proposed model is also based on the ResNet model to solve the TSC task.

However, there are few studies based on using time-series imaging and hybrid quantum neural networks to solve TSC tasks. Most of the existing hybrid quantum neural network models have mostly focused on solving predictive classification tasks where the original data are images and have not explored the possibilities of hybrid quantum models for processing secondary images. Our proposed model proves its possibility and, in experiments on the UCR archive, it is found that the hybrid model can achieve an accuracy equal to or even better than the classical model for the classification of time-series imaging data compared to several best models.

3. Methodology

3.1. MIHQNN framework

The overall architecture of the MIHQNN is shown in Fig. 1, which consists of three sequential phases: the imag-

ing phase, feature extraction phase, and classification phase.

1. In the imaging stage, we classify the dataset into two types, one with sufficient sample data and the other with insufficient sample data. The former applies the RP time series imaging method alone to transform a one-dimensional time series into a 2D image, and the latter uses four time series imaging methods and fuses them into a single image.

2. In the feature extraction phase, we will use a modified ResNet model. This includes several residual blocks and fully connected layers to output the extracted features.

3. In the classification phase, we designed four classifiers based on quantum circuits for each dataset, and trained and tested them for 50 iterations.

See Subsections 3.2–3.4 for more details.

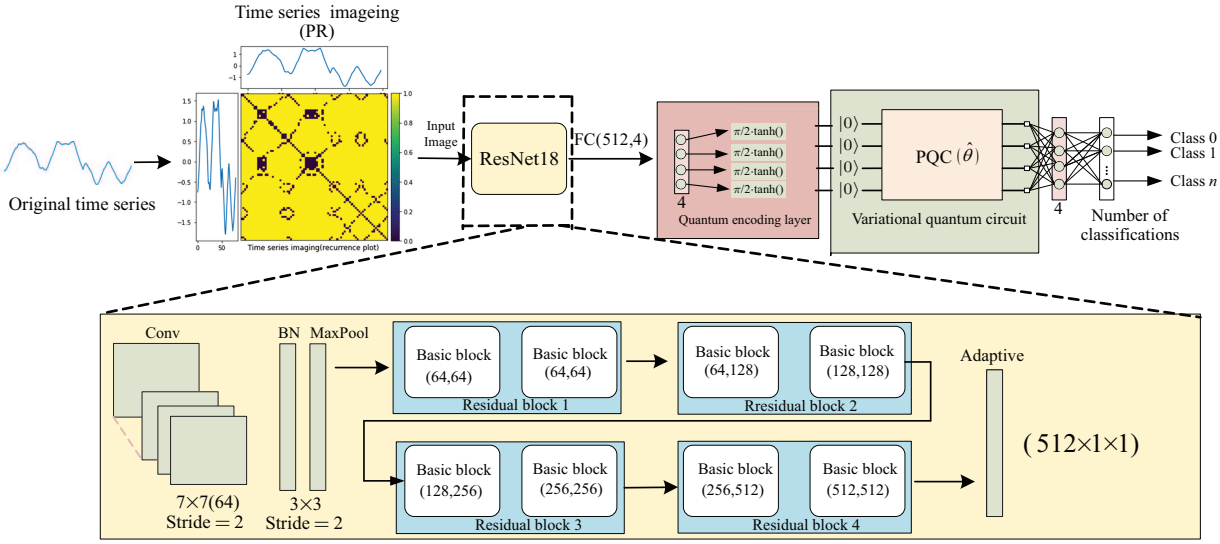


Fig. 1. MIHQNN model based on RP time series imaging.

3.2. Imaging stage

For datasets with a sufficient number of samples, we use a single RP imaging, because with a sufficient number of samples, the MIHQNN model rarely over-fits during training. The use of multiple imaging fusions may add additional consumption, including time and memory consumption. However, for datasets with insufficient sample size, the need to use multiple imaging becomes apparent. The edges and diagonals of time series images contain richer classification features. Using the feature fusion method, the four sets of feature vectors are combined into a joint vector, thus extending the feature vector space of the original image, which is beneficial to the deep learning framework for its classification and thus improves the classification rate. Multiple imaging and fusion, specifically four imaging methods, RP, MTF, GASF, and GADF, are used to convert each time series into four images. And an image is formed by RP, MTF, GASF, and GADF in a counterclockwise direction, which can be seen in Fig. 2. The principles of the four time series imaging are described in detail below.

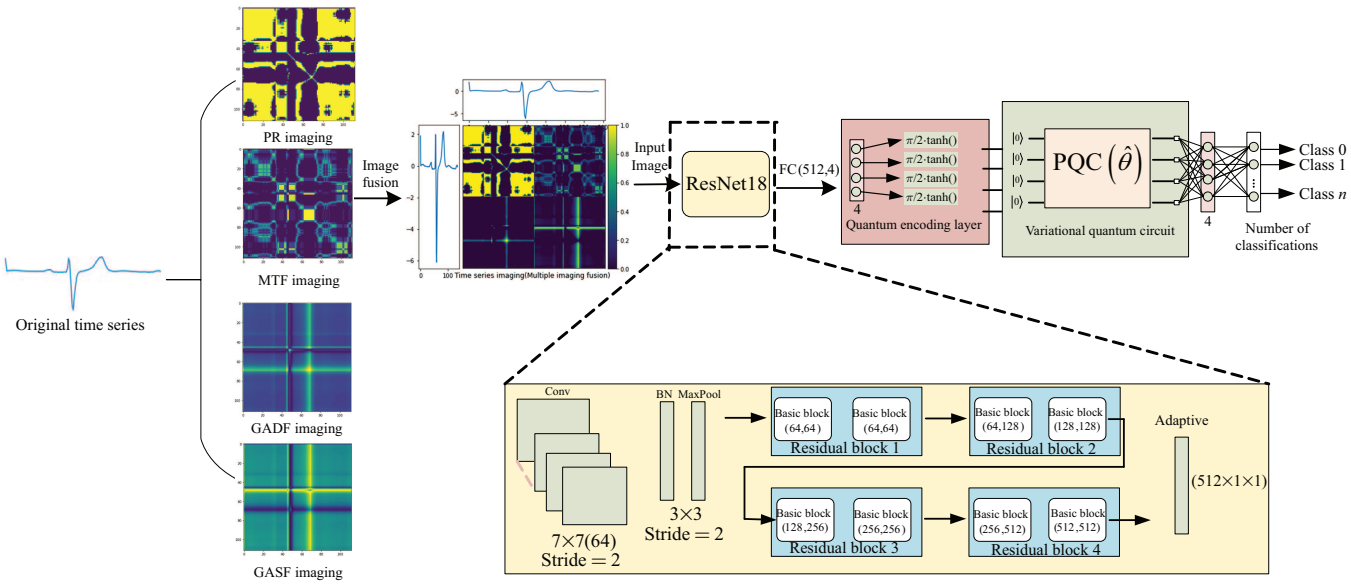


Fig. 2. MIHQNN model based on multiple time series imaging fusion.

3.2.1. RP imaging

RP is an important method for analyzing the periodicity, chaos, and non-stationarity of time series. It can be used to reveal the internal structure of time series, giving a priori knowledge about similarity, informativeness, and predictability. RP is particularly suitable for short time series data, and can test the smoothness and intrinsic similarity of time series. RP is an image that represents the distance between trajectories extracted from the original time series, as follows.

Given the time series $(x_1,\ldots,x_n)$, the result of reconstructing the time series is

$$\begin{aligned} &\boldsymbol{x}_i = \left(x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau}\right), \\ &\forall i \in \{1,\ldots,n-(m-1)\tau\}, \end{aligned} \tag{1}$$

where m denotes the embedding dimension and τ denotes the delay time.

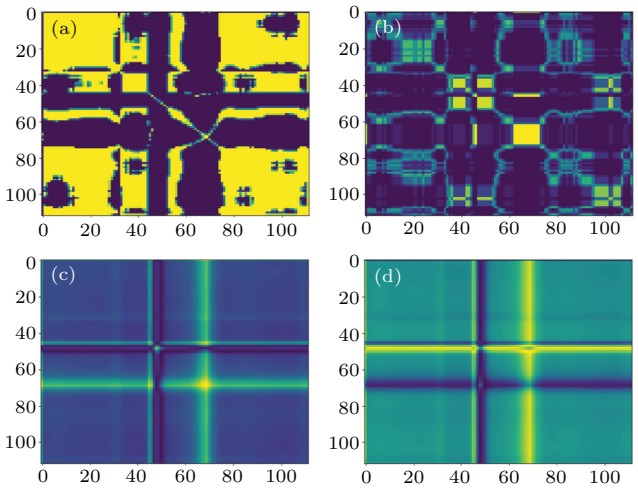


Fig. 3. Applying RP, MTF, GASF, and GADF to convert time series to images.

The distance between point i, $\boldsymbol{x}_i$, and point j, $\boldsymbol{x}_j$, in the reconstructed phase space is

$$\boldsymbol{S}_{ij} = \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|, \quad \forall i,j \in \{1,\ldots,n-(m-1)\tau\}. \tag{2}$$

The recursive value can then be expressed as

$$R_{ij} = \theta\left(\varepsilon - \boldsymbol{S}_{ij}\right), \quad \forall i,j \in \{1,\ldots,n-(m-1)\tau\}, \tag{3}$$

where θ represents the Heaviside function, a segmentation function that is commonly used in recursive graphs to represent the stopping conditions of recursion. ε is a threshold; if "none", the graph is repeated without binarization. If "point", the threshold is calculated, e.g., the percentage of points is less than the threshold. If "distance", then the threshold is calculated as the maximum distance. This is set to "none" when plotting RP in this article.

Figure 3(a) shows the image after applying RP to a time series transformation, where $\tau = 0.1$ and $m = 1$.

3.2.2. MTF imaging

MTF is proposed based on the first-order Markov chain transformation because Markov transfer fields are insensitive to the time dependence of the sequences, so MTF is based on the relationship of time positions.

The process of constructing the MTF is as follows.

Step 1: First divide the time series data into Q bins. Each data point i corresponds to a unique bin, i.e., each data point has only one identifier q_i, $i \in \{1,2,\ldots,Q\}$.

Step 2: Construct a Markov state transfer matrix

$$A_{ij} = P\left(S_t = j \mid S_{t-1} = i\right), \tag{4}$$

where A_{ij} represents the transfer probability of state i transforming to state j. The transfer probability is generally estimated using the maximum likelihood method, where the matrix size is $[Q, Q]$.

Step 3: Construct the Markov variation field $\boldsymbol{M}$. $\boldsymbol{M}$ is an $N \times N$ matrix and N is the length of the time series,

$$M_{kl} = A_{q_k q_l}, \tag{5}$$

where q_k is the bin of x_k, q_l is the bin of x_l, and x is the time series data.

The shape of the MTF is as follows:

$$\begin{bmatrix} \omega_{i,j} \mid x_1 \in q_i, x_1 \in q_j & \omega_{i,j} \mid x_1 \in q_i, x_2 \in q_j & \cdots & \omega_{i,j} \mid x_1 \in q_i, x_n \in q_j \\ \omega_{i,j} \mid x_2 \in q_i, x_1 \in q_j & \omega_{i,j} \mid x_2 \in q_i, x_2 \in q_j & \cdots & \omega_{i,j} \mid x_2 \in q_i, x_n \in q_j \\ \cdots & \cdots & \cdots & \cdots \\ \omega_{i,j} \mid x_n \in q_i, x_1 \in q_j & \omega_{i,j} \mid x_n \in q_i, x_2 \in q_j & \cdots & \omega_{i,j} \mid x_n \in q_i, x_n \in q_j \end{bmatrix}. \tag{6}$$

Figure 3(b) shows the image after applying MTF to a time series transformation, where bins = 5.

3.2.3. GAF imaging

A GAF image is an image obtained from a time series, mainly based on some temporal correlation between each pair of values starting at a given time. There are two methods: GASF and GADF. The difference between these two methods is that, when the scaled time series data are converted from a right angle coordinate system to a polar coordinate system, GASF considers the sum of the angles between different points as a criterion for identifying different points in time and GADF

considers the difference in angle between different points as a criterion for identifying different time points. The steps of the GAF implementation are as follows.

Step 1: Scale the data to $[-1,1]$ (in this paper we scale the data to $[-1,1]$; we can also scale the data to $[0,1]$). The formula for scaling is

$$\tilde{x}_{-1}^{i} = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)}, \tag{7}$$

where X represents a time series data.

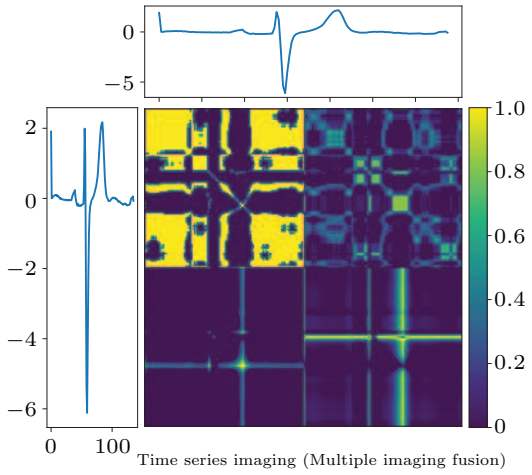


Fig. 4. Feature fusion of combined images. In order to solve the problem of single image feature loss during MIHQNN training and classification, the combined image algorithm converts the time series into four images and combines them into a square image for feature-level fusion. The combined images are RP, MTF, GASF, and GADF from top left to bottom right.

Step 2: Convert the scaled sequence data to polar coordinates, i.e., the value $\widetilde{X}$ of the time series is regarded as the angle cosine and the time stamp is regarded as the radius,

$$\left\{ \begin{array}{l} \theta = \arccos(\tilde{x}_i), \quad -1 \le x_i \le 1, \ \ \tilde{x}_i \in X \\ r = \frac{t_i}{N}, \quad t_i \in N \end{array} \right\}. \tag{8}$$

Step 3: Calculate the GASF and GADF respectively according to the different criteria of identification

$$\mathrm{GASF} = [\cos(\phi_i + \phi_j)] = \widetilde{X}' \cdot \widetilde{X} - \sqrt{I - \widetilde{X}^2}' \cdot \sqrt{I - \widetilde{X}^2}, \tag{9}$$

$$\mathrm{GADF} = [\sin(\phi_i - \phi_j)] = \sqrt{I - \widetilde{X}^2} \cdot \widetilde{X} - \widetilde{X}' \cdot \sqrt{I - \widetilde{X}^2}. \tag{10}$$

Figures 3(c) and 3(d) show the images after applying the GASF and GADF to transform the time series. Figure 4 shows the shape after fusing the four images.

3.3. Feature extraction stage

After the raw time series data have been converted into images, CNN models can be trained to perform feature extraction on these images, and the feature extraction model in this paper is based on ResNet for improvement. The main purposes of the modification are: 1) Some mature deep learning models, such as ResNet, are commonly used for RGB (three-channel) color image classification tasks. However, after the processing in Subsection 3.2, the obtained image type is single channel, and the input channel of the first convolution layer needs to be modified to single channel. 2) The output of the feature extraction model must be consistent with the input requirements of the later classification stage. In this paper, the trainable parametric variational quantum circuit used as the classification stage is 4 qubits in number, so it is necessary to set the number of features of the output of the feature extraction model to 4.

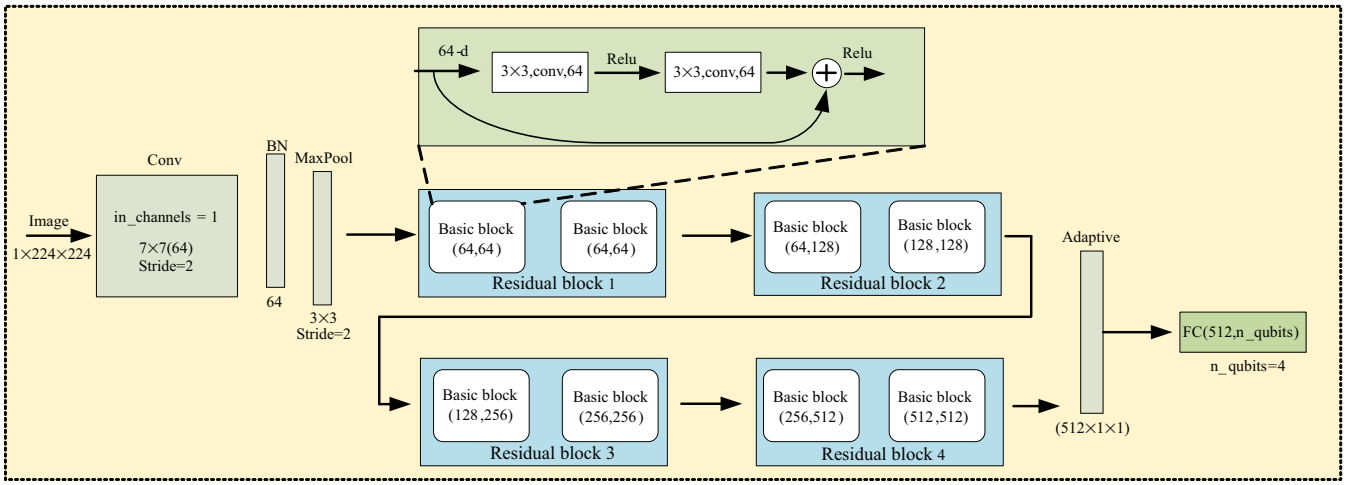


Fig. 5. Time series data feature extraction model.

As shown in Fig. 5, the feature extraction model contains three main components. The composition of the first part: a 1-channel input of size 224×224 with 64 output channels, a 7×7 convolutional layer (conv) of step size 2 followed by a batch normalization layer (BN), followed by a 3×3 maximum pooling layer (MaxPool) of step size 2. BN is mainly used to solve the gradient disappearance and gradient explosion problems. The composition of the second part: it mainly contains four residual blocks, each with two 3×3 convolutional layers with the same number of output channels inside. Each con-

volutional layer is followed by a batch planning layer and a ReLU activation function. The input also needs to be added directly before the final ReLU activation function in each residual block. Such a design can satisfy the requirement that the outputs of the 2 convolutional layers have the same shape as the inputs, so that they can be added together. The composition of the third part: finally, the features extracted from the residual layer are followed by a global average pooling, and the global average pooling is added to suppress the overfitting phenomenon. Then a fully connected layer is added to match the input dimension of the subsequent quantum neural network classifier.

3.4. Classification stage

In the classification phase, we focus on exploring the correlation between the possibility of using quantum classifiers instead of classical classifiers and the classification accuracy of different quantum circuits. The classification phase has two main components: 1) The features obtained after the feature extraction stage are classical states, which cannot be directly embedded in the quantum circuit at this point, and must go through an encoding section where the extracted features are fed into the quantum variational circuit. 2) Quantum circuits are the basis of quantum neural networks. In quantum neural networks, quantum circuits are used to build neurons and layers, while the training and optimization of neural networks are achieved by adjusting the parameters of quantum gates. Both the design and optimization of quantum circuits affect the efficiency and performance of quantum neural networks. Once the features were input, we used four quantum circuits as four classifiers each to explore the effect of different quantum circuits on the classification performance.

3.4.1. Encoding stage

In order to construct hybrid quantum models, it is crucial to convert classical data to a high-dimensional quantum representation. Ground state coding,[43] amplitude coding,[43] angle coding,[43] IQP coding,[44] and Hamiltonian evolution coding[45] have been proposed. The encoding approach we take is to convert the previously obtained eigenvalues into angles, as in Fig. 1, by multiplying each of the four eigenvalues by $(\pi/2)$, and then applying them to the quantum circuit. As in Fig. 6(a), the Hadamard gate (H) can be applied first to act on the initial quantum state, placing the initial quantum state in the superposition state. The RY gate is then applied to the qubit, and the control angle of the RY gate is the angle previously converted by the eigenvalue, which enables the transfer of the obtained classical information to the subsequent quantum circuit.

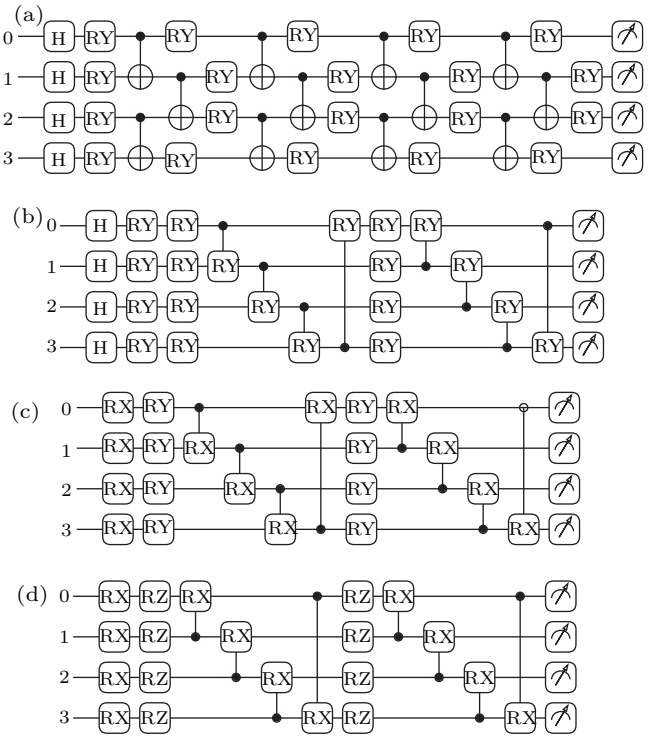


Fig. 6. Parametric quantum circuits. (a) Circuit 1, four qubits and four layer configuration. (b) Circuit 2, four qubits and single layer configuration. Mainly composed of the RY gate. (c) Circuit 3, four qubits and single layer configuration. Mainly composed of the RX gate. (d) Circuit 4, four qubits and two layer configuration.

3.4.2. Quantum circuit classification stage

To create MIHQNN, parametric quantum circuits need to be used as the hidden layer of the neural network, and four types of circuits are selected and designed in this paper. A description of the circuit employed will help provide insight into how the quantum gate affects the final result. Figure 6(a) related configurations are described here and presents more information about the quantum circuit.[40]

In Fig. 6(a), each qubit first gets a superposition state through the H gate and then through the RY gate, the parameters of which are the outputs of the classical network. Then CNOT gates are used to entangle with each other, where the RY gates that pass through are constant rotation, i.e., $R_y(\theta_i) = I$, $i = 0, 1, 2, 3$. Then the state before the CNOT gate is

$$\begin{aligned} |\psi_1\rangle &= \left(\bigotimes_{i=0}^{i=3} H\right)|0000\rangle = \bigotimes_{i=0}^{i=3}(H|0\rangle) \\ &= \left(\frac{1}{\sqrt{2}}\right)^4 \bigotimes_{i=0}^{i=3}(|0\rangle + |1\rangle). \end{aligned} \tag{11}$$

Because the RY gate is assumed to be a constant rotation here, the state $|\psi_1\rangle$ remains unchanged after passing through the CNOT gate, i.e., $|\psi_2\rangle = |\psi_1\rangle$. The effect of RY gates in the real situation is not constant rotation, which reflects the importance of encoding classical information into quantum information and feeding it into the quantum circuit. Then the

state after passing through the revolving door four times is

$$|\psi_3\rangle = \bigotimes_{j=0}^{j=3} R_y(\theta_j)\,|\psi_2\rangle\,. \tag{12}$$

The depth of circuit 1 is 4. The combination of the CNOT gate and the RY gate is repeated a total of four times. During training the parameter θ_j is also called quantum weight, similar to the weights in neural networks, and θ_j is trainable. The principles of circuit 2, circuit 3, and circuit 4 are the same as those of circuit 1, except that the combination and depth of the quantum gates are different, in analogy to the different network structures in neural networks. After the superposition and entanglement of the quantum circuit, etc., the final measurement needs to be obtained, and we use the Pauli Z-gate at four qubits. The circuit needs to repeat the measurement, and the number of repetitions set in this paper is 1000. The number of measurements obtained is 4. It is also necessary to link a fully connected layer after the quantum circuit; the input of the fully connected layer is 4 and the output is n, where n denotes the number of targets for classification. The follow-up experiments focus on testing datasets with a target number of 2 or 3 in the UCR archive.

With the whole MIHQNN framework established, we conducted comprehensive experiments, and the results are shown in Section 4.

4. Experiment

4.1. Experimental setup

We evaluated the performance of MIHQNN on a dataset of UCR time series classification archives, of which 14 datasets were selected, with the number of classifications of 2 and 3.

In the subsequent sections, a number of experiments are performed. 1) Experiments on four different quantum circuits. 2) Experiments on three classical deep learning models (ResNet, DENSENET, VGGNet) for comparison with MIHQNN. 3) For MIHQNN, RP time series imaging is compared with multiple time series imaging fusions.

Our proposed MIHQNN is implemented based on PyTorch and pennylane and runs on an NVIDIA GeForce GTX 1650 graphics card with 896 cores and 4 GB of global memory. The hyperparameters of the model are $\{\text{batch_size} = 10, \text{lr} = 0.0007, \text{step_size} = 10, \text{gamma} = 0.1\}$, which denote the batch size, learning rate, learning rate adjustment period, and multiplication factor for updating the learning rate, respectively. The learning rate adjustment period and the learning rate multiplication factor indicate that the learning rate is adjusted to lr$\times$gamma every 10 epochs. For the four MIHQNN classifiers, the Adam optimizer was used to train 50 cycles, regardless of whether the quantum circuits were different, and cross-entropy was chosen as the loss function. The three deep learning models are in the same conditions and environment as MIHQNN.

In this paper, the performance of all TSC methods is evaluated by the classification accuracy, which is defined as

$$\text{acc} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{13}$$

where TP indicates the number of predicted results that agree with the true results, and FN indicates the number of predicted results that do not agree with the true results.

4.2. Comparison with the classical model

To evaluate the performance of MIHQNN, we chose three models that have excelled in processing images and time series in the last five years, ResNet18, DENSENET121, and VGGNet11. They are all three deep learning models based on PyTorch implementation. The number of parameters of the quantum neural network used as the classifier in the MIHQNN model is 2062, while the number of parameters of the classifiers of the three classical models ResNet18, DENSENET121, and VGGNet11 are 513000, 1025000, and 1236428, respectively. In the comparison with the classical model, 10 standard datasets from the UCR archive were selected. The raw time series data are first transformed into 2D images based on RP imaging, and then fed into our model and comparison model for training and testing, respectively. After 50 cycles of training, the best classification accuracy that each model can achieve is collected.

Table 1 shows the accuracy of our proposed method and other TSC methods. It also shows the descriptive information of each dataset $\{c, N_{\text{tr}}, N_{\text{te}}, L\}$, which indicate the number of categories, the number of training sets, the number of test sets, and the length of the time series of the dataset, respectively, and where MIHQNN1 represents the MIHQNN model implemented based on circuit 1.

The results in Table 1 show that the MIHQNN model can achieve the same or even better accuracy than the classical model when faced with the TSC task. For example, the MIHQNN model based on quantum circuit 1 achieves the best performance on 3 of the 10 standard datasets archived at UCR. The MIHQNN model based on quantum circuit 2 achieves the best performance on 2 of the 10 standard datasets archived at UCR. In addition, in the experiments on the dataset SmallKitchenAppliances, the accuracy of MIHQNN based on four different quantum circuits is 0.874, 0.832, 0.834, and 0.840, respectively, while the classical models (ResNet, DENSENET, VGGNet) have accuracies of 0.813, 0.792, and 0.781. In the case where the number of trainable parameters of the MIHQNN model is much less than the three classical models mentioned above, the MIHQNN model can achieve the same

or even better accuracy than the classical model when faced with the TSC task. This fully demonstrates the feasibility and effectiveness of the hybrid quantum model to deal with time series data, proving its high theoretical and practical value, and it is expected to become one of the tools for time series data analysis in the future.

Table 1. Performance (in terms of accuracy) of the proposed method on 10 selected datasets from the UCR archive compared to the prior art TSC algorithm.

Dataset	c	N_{tr}	N_{te}	L	MIHQNN1	MIHQNN2	MIHQNN3	MIHQNN4	ResNet	DENSENET	VGGNet
FordA	2	3601	1320	500	0.844	0.872	0.809	0.815	0.820	0.866	0.718
FordB	2	3606	810	500	0.861	0.872	0.869	0.874	0.867	0.893	0.811
CBF	3	900	30	128	0.992	0.997	0.997	0.995	0.992	0.994	1.000
ECGFiveDays	2	861	23	136	0.969	0.976	0.977	1.000	0.996	0.990	0.997
HandOutlines	2	1000	370	2709	0.927	0.913	0.916	0.914	0.918	0.910	0.871
ItalyPowerDemand	2	1029	67	24	0.949	0.944	0.948	0.946	0.961	0.959	0.972
MoteStrain	2	1252	20	84	0.903	0.925	0.948	0.933	0.935	0.937	0.938
PhalangesOutlinesCorrect	2	1800	858	80	0.777	0.845	0.822	0.838	0.832	0.833	0.702
ProximalPhalanxOutlineCorrect	2	600	291	80	0.903	0.888	0.875	0.871	0.883	0.880	0.750
SmallKitchenAppliances	3	375	375	720	0.874	0.832	0.834	0.840	0.813	0.792	0.781

4.3. Comparison between four different quantum circuits

For this experiment, we selected four variational quantum circuits as classifiers, each with the same encoding stage and the same initial parameters for each type of revolving gate. As shown in Fig. 6, all circuits are one layer, except for circuit 1, which has a four layer structure. circuit 1 and circuit 2 each incorporate four H gates between the initial state and the embedded classical information, while circuit 3 and circuit 4 embed the classical information directly after the initial state. By examining the data in Table 1, it is found that the model based on quantum circuit 1 achieves the best performance on three datasets, HandOutlines, ProximalPhalanxOutlineCorrect and SmallKitchenAppliances. The model based on quantum circuit 2 achieves the best performance on two datasets, FordA and PhalangesOutlinesCorrect. The models based on quantum circuits 3 and 4 both achieve the best performance on only one dataset. This suggests that quantum circuits 1 and 2 may be better suited to handling the TSC task. The reason for this different performance may be that the first two circuits insert a layer of H gates between the classical data embedding layer and the initial state. The H-gate can transform the initial state into a superposition state. A 4 qubit initial state passing through four H-gates will form a system with $4^2 = 16$ states existing simultaneously. So the overall effect of the classification is better compared to the quantum circuit without H-gate processing. It also shows that the design and selection of quantum circuits are important when using hybrid quantum neural networks to handle TSC tasks. The layout of qubits in the quantum circuit, the choice of quantum gates, and the optimization will affect the efficiency and performance of the circuit, which has been studied extensively by several researchers in the past.[41,42]

4.4. Comparison between different time series imaging methods

limited by the number of qubits, the number of classifications in the classification datasets targeted in the experiments of this paper is between 1 and 4. Some of the datasets in the UCR archive are too sparse in the number of training samples, causing the selected datasets to be prone to overfitting during the training process. For datasets with sparse data sample sizes, we propose the use of multiple time series imaging fusion methods instead of single RP time series imaging. A time sequence will be converted into a 56×56 2D image by RP, MTF, GASF, and GADF respectively, and then the four images will be fused into a 224×224 image. To validate the possibility of the idea, we selected four datasets in the UCR archive and tested the accuracy of RP-based MIHQNN and MIHQNN based on multiple imaging fusions, as shown in Tables 2 and 3.

Table 2. Accuracy of RP time-series imaging-based MIHQNN models for datasets with few samples.

Dataset	c	N_{tr}	N_{te}	L	MIHQNN1	MIHQNN2	MIHQNN3	MIHQNN4
computer	2	250	250	720	0.800	0.800	0.812	0.832
ScreenType (ST)	3	375	375	720	0.658	0.621	0.637	0.618
herring	2	64	64	512	0.781	0.718	0.843	0.703
ham	2	109	105	431	0.895	0.876	0.885	0.847

Table 3. Accuracy of MIHQNN models based on multiple time series imaging and fusion for datasets with a small number of samples.

Dataset	c	N_{tr}	N_{te}	L	MIHQNN1	MIHQNN2	MIHQNN3	MIHQNN4
computer	2	250	250	720	0.848	0.800	0.820	0.840
ScreenType (ST)	3	375	375	720	0.746	0.749	0.674	0.733
herring	2	64	64	512	0.796	0.812	0.781	0.781
ham	2	109	105	431	0.904	0.923	0.866	0.885

Via observing Figs. 7–10, Tables 2 and 3, for the four datasets with sparse sample sizes, the improvement in classification accuracy of MIHQNN based on multiple time series imaging and fusion can be found to be significant compared to MIHQNN based on single RP time series imaging. This is predictable because using multiple imaging and fusion will result in more features, especially at the diagonal bringing together the features of the four images. Multiple imaging and fusion are not necessary, and for sufficiently large sample sizes, using multiple imaging adds additional overhead, such as memory consumption and longer training times. But the idea of multiple imaging and fusion is feasible and valid for particular datasets. And in Figs. 7–10, the performance of the multiple imaging fusion-based model is smoother on the test set, which can indicate that the multiple imaging fusion model has better ability to generalize and mitigate the overfitting phenomenon.

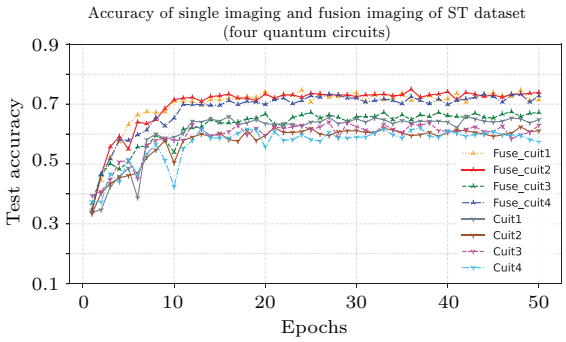


Fig. 7. Classification accuracy of four MIHQNN models based on RP and multiple imaging for dataset ST.

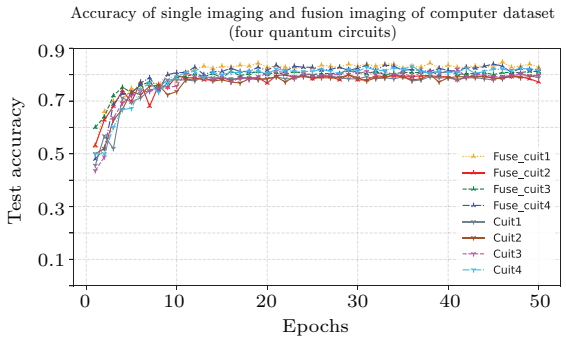


Fig. 8. Classification accuracy of four MIHQNN models based on RP and multiple imaging for dataset computer.

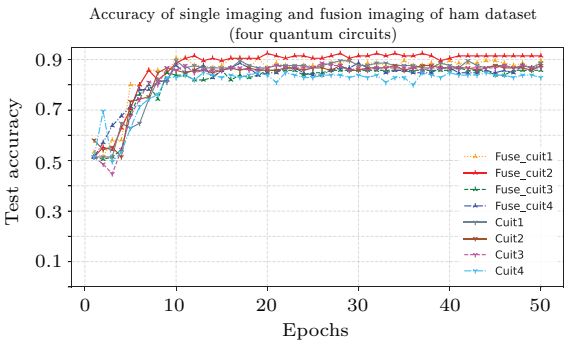


Fig. 9. Classification accuracy of four MIHQNN models based on RP and multiple imaging for dataset ham.

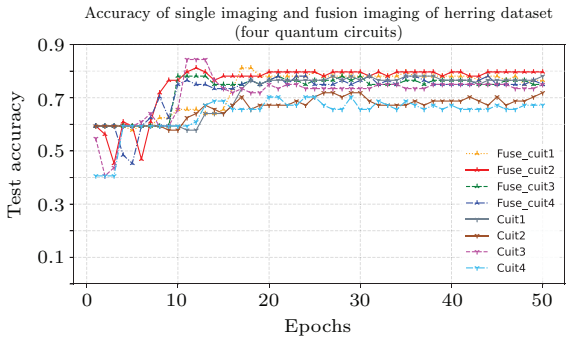


Fig. 10. Classification accuracy of four MIHQNN models based on RP and multiple imaging for dataset herring.

Through the above combined comparison, we validate the usability of the approaches based on time series imaging and hybrid quantum neural networks. In particular, the proposed new framework, MIHQNN, was tested in the UCR archive, demonstrating its remarkable performance as well as proving the possibility of the new framework handling TSC problems.

5. Conclusions

We propose a new framework MIHQNN for the TSC task, using a hybrid quantum neural network architecture to recognize 2D images transformed from time series data. Images are converted according to the size of the training samples in the dataset; large samples will be converted by a single RP imaging, and those with small sample sizes will be converted by multiple imaging and fusion. Converting time series to 2D images makes it easier to see similarities in categories from the converted images. In particular, DL has recently been effective in image recognition, and we use a combination of deep learning frameworks and quantum neural networks in order to achieve the best classification results. A number of datasets from the UCR archive were tested and compared with several of the best recent TSC methods. The experimental results prove that MIHQNN performs better overall.

We also separately examined the effect of different quantum circuits on the classification of MIHQNN, as well as the effect of single RP imaging and multiple imaging fusions on the classification of MIHQNN. This research highlights the potential of applying quantum computing to TSC and provides the theoretical and experimental background for future research.

Acknowledgements

Project supported by the National Natural Science Foundation of China (Grant Nos. 61772295 and 61572270), the PHD foundation of Chongqing Normal University (Grant No. 19XLB003), and Chongqing Technology Foresight and Institutional Innovation Project (Grant No. cstc2021jsyj-yzysbAX0011).

References

[1] Paparrizos J and Gravano L 2017 *ACM Transactions on Database Systems (TODS)* **42** 1

[2] Lines J and Bagnall A 2015 *Data Mining and Knowledge Discovery***29** 565

[3] Xiao X and Lu Y 2019 *6th Asia-Pacific Conference on Synthetic Aperture Radar* November 26-29, 2019, Xiamen, China, **1**

[4] Geler Z, Kurbalija V, Ivanovi M and Radovanovi M 2020 *International Conference on Innovations in Intelligent Systems and Applications (INISTA)* August 24–26, 2020, Novi Sad, Serbia, 1

[5] Buza K, Nanopoulos A and Schmidt-Thieme L 2010 *13th IEEE International Conference on Computational Science and Engineering* December 11–13, Hong Kong, China, 48

[6] Morchen F, Ultsch A, Thies M and Lohken I 2005 *IEEE Transactions on Audio, Speech, and Language Processing* **14** 81

[7] Lin J, Keogh E, Wei L and Lonardi S 2007 *Data Mining and Knowledge Discovery* **15** 107

[8] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195

[9] Rebentrost P, Mohseni M and Lloyd S 2014 *Phys. Rev. Lett.* **113** 130503

[10] Li Z, Liu X, Xu N and Du J 2015 *Phys. Rev. Lett.* **114** 140504

[11] Wiebe N, Kapoor A and Svore K M 2014 arXiv:1412.3489

[12] Rebentrost P, Bromley T R, Weedbrook C and Lloyd S 2018 *Phys. Rev. A* **98** 042308

[13] Benedetti M, Realpe-Gómez J and Perdomo-Ortiz A 2018 *Quantum Science and Technology* **3** 034007

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

[15] Steinbrecher G R, Olson J P, Englund D and Carolan J 2019 *npj Quantum Information* **5** 60

[16] Levine Y, Sharir O, Cohen N and Shashua A 2019 *Phys. Rev. Lett.* **122** 065301

[17] Yang Z and Zhang X 2020 *New J. Phys.* **22** 033041

[18] Amin M H, Andriyash E, Rolfe J Kulchytskyy B and Melko R 2018 *Phys. Rev. X* **8** 021050

[19] Kieferová M and Wiebe N 2017 *Phys. Rev. A* **96** 062327

[20] Lloyd S and Weedbrook C 2018 *Phys. Rev. Lett.* **121** 040502

[21] Hu L, Wu S H, Cai W, Ma Y, Mu X, Xu Y, Wang H, Song Y, Deng D L and Zou C L 2019 *Sci. Adv.* **5** eaav2761

[22] Dallaire-Demers P L and Killoran N 2018 *Phys. Rev. A* **98** 012324

[23] Cui Z, Chen W and Chen Y 2016 arXiv:1603.06995

[24] Wang Z, Yan W and Oates T 2017 *International Joint Conference on Neural Networks May 14-19, 2017, Anchorage, Alaska, USA* p. 1578

[25] Wang Z and Oates T 2015 arXiv:1506.00327

[26] Hatami N, Gavet Y and Debayle J 2017 *Tenth International Conference on Machine Vision*, November 13–15, 2017, Vienna, Austria **10696** 242

[27] Chen W and Shi K 2019 *Neurocomputing* **359** 384

[28] Karim F, Majumdar S, Darabi H and Chen S 2017 *IEEE Access* **6** 1661

[29] Arute F, Arya K, Babbush R, *et al.* 2019 *Nature* **574** 505

[30] Boixo S, Isakov S V, Smelyanskiy V N, *et al*. 2018 *Nat. Phys.* **14** 595

[31] Kerenidis I, Landman J and Prakash A 2019 arXiv:1911.01117

[32] Houssein E H, Abohashima Z, Elhoseny M and Mohamed W M 2022 *Journal of Computational Design and Engineering* **9** 343

[33] Mari A, Bromley T R, Izaac J, Schuld M and Killoran N 2020 *Quantum* **4** 340

[34] Krizhevsky A, Sutskever I and Hinton G E 2017 *Communications of the ACM* **60** 84

[35] Huang G, Liu S, Van der Maaten L and Weinberger K Q 2018 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 18–21, 2018, Salt Lake City, USA, p. 2752

[36] Simonyan K and Zisserman A 2014 arXiv:1409.1556

[37] Eckmann J P, Kamphorst S O and Ruelle D 1995 *World Scientific Series on Nonlinear Science Series A* **16** 441

[38] Wang Z and Oates T 2015 *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 25–30, 2015, Austin, Texas USA **6** 1661

[39] Dau H A, Bagnall A, Kamgar K, *et al* 2019 *IEEE/CAA Journal of Automatica Sinica* **6** 1293

[40] Nielsen M A and Chuang I 2002 *Quantum Am. J. Phys.* **70** 558

[41] Du Y, Huang T, You S, Hsieh M H and Tao D 2022 *npj Quantum Information* **8** 62

[42] Hubregtsen T, Pichlmeier J, Stecher P and Bertels K 2021 *Quantum Machine Intelligence* **3** 1

[43] Schuld M 2021 arXiv:2101.11020

[44] Havlíek V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209

[45] Huang H Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H, and McClean J R 2021 *Nat. Commun.* **12** 2631