



OPEN

# Scalable parameterized quantum circuits classifier

Xiaodong Ding<sup>1,2</sup>, Zhihui Song<sup>1,2</sup>, Jinchun Xu<sup>1</sup>, Yifan Hou<sup>1</sup>, Tian Yang<sup>1</sup> & Zheng Shan<sup>1</sup>✉

As a generalized quantum machine learning model, parameterized quantum circuits (PQC) have been found to perform poorly in terms of classification accuracy and model scalability for multi-category classification tasks. To address this issue, we propose a scalable parameterized quantum circuits classifier (SPQCC), which performs per-channel PQC and combines the measurements as the output of the trainable parameters of the classifier. By minimizing the cross-entropy loss through optimizing the trainable parameters of PQC, SPQCC leads to a fast convergence of the classifier. The parallel execution of identical PQCs on different quantum machines with the same structure and scale reduces the complexity of classifier design. Classification simulations performed on the MNIST Dataset show that the accuracy of our proposed classifier far exceeds that of other quantum classification algorithms, achieving the state-of-the-art simulation result and surpassing/reaching classical classifiers with a considerable number of trainable parameters. Our classifier demonstrates excellent scalability and classification performance.

With the development of quantum computing technology<sup>1</sup>, quantum machine learning<sup>2–15</sup> has become a hot research field, and multi-category classification is one of the important tasks. Conventional multi-category classification algorithms are typically based on deep learning frameworks<sup>16–18</sup>, but these methods require large amounts of data and computational resource, and suffer from issues such as overfitting. Quantum machine learning, which combines quantum computing and machine learning, has the advantages of accelerating computation and reducing overfitting, making it widely applicable to various problems. However, for classification problems, most existing quantum machine learnings focus on binary classification tasks, and for multi-category classification problems, the research mainly focuses on deploying classical neural networks at the end of quantum algorithms<sup>19</sup>. Quantum machine learnings based on measurement of projection<sup>9</sup>, QF-hNet<sup>2</sup> and other methods<sup>20–22</sup> have also been proposed for solving the multi-category classification problem, but the classification effect and scalability are poor. It is worth looking forward to providing quantum multi-category classifiers with good scalability and excellent classification results. Therefore, in this paper, we propose SPQCC with PQC<sup>23,24</sup> as the core. This choice is made because PQC, as a generalized quantum machine learning model for quantum machine learning, has the properties of certain resilience to certain types of errors, coherence time, and more flexible operation attributed to the properties of quantum parallelism, superposition and entanglement, which has shown strong learning capability and is now used as a core module for algorithms such as QNN<sup>25–27</sup>, QCNN<sup>28,29</sup>, QLSTM<sup>30,31</sup>, and QGAN<sup>32,33</sup>. The number of parallel PQCs of this classifier is the same as the number of classes of samples, which has good scalability. Secondly, the multiple PQCs have the same structure and scale, and only needs to design one channel of PQC to complete the design of parallel multiple PQCs of the classifier, which makes the design of the classifier more convenient. Meanwhile, these PQCs are allowed to be executed in parallel, so the execution efficiency of the classifier is higher. Finally, the measurements of the parallel multiple PQCs are combined as the output of the final classifier, and the trainable parameters of all the PQCs are optimized by minimizing the cross-entropy loss function, which leads to fast convergence of the classifier. Additionally, in this paper we also emphasize on the design of PQC, the circuit measurement methods and parameters optimization.

## Results

The scalability and classification effectiveness of our proposed classifier are verified. The scalability of the model is primarily demonstrated in two aspects: the ability to handle datasets with varying numbers of categories and the ability to adjust the model's size. Scalability when dealing with datasets with different numbers of categories: When faced with datasets containing varying numbers of categories, the model must be capable of adapting flexibly while maintaining good performance. Specifically, the model can be scaled by simply adjusting the number of PQCs in the model based on the number of categories in the dataset. This approach ensures that the model is both robust and flexible when dealing with datasets containing varying numbers of categories. Scalability regarding

<sup>1</sup>Laboratory for Advanced Computing and Intelligence Engineering, Zhengzhou 450001, China. <sup>2</sup>These authors contributed equally: Xiaodong Ding and Zhihui Song. ✉email: shanzhengzz@163.com

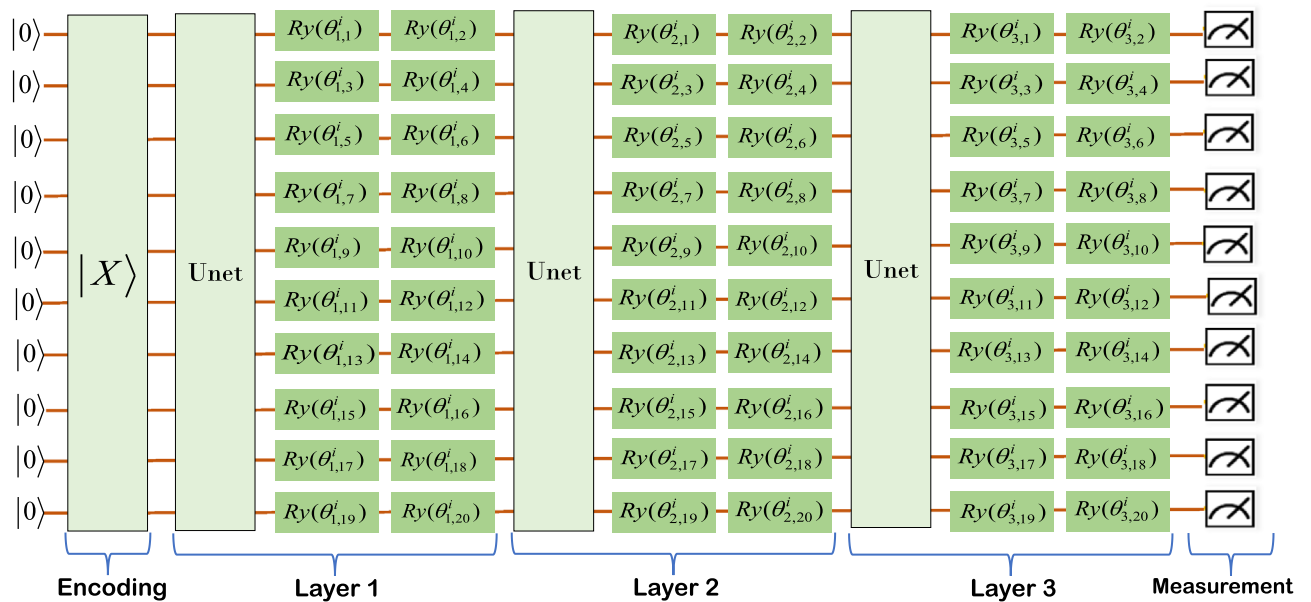
the size of the model: This involves not only an expansion in the number of PQCs, but also the adjustment of the number of layers in the quantum circuits. We chose the MNIST dataset for separate classification simulations on the quantum simulator TensorCircuit. This dataset was chosen because for more than a decade, researchers from the fields of Machine Learning, Machine Vision, Artificial Intelligence, and Deep Learning have used this dataset as one of the benchmarks for measuring classification algorithms<sup>34–38</sup>. We compare the classification accurate of our classifier with the classification accurate of other classifiers<sup>9,39</sup> on the MNIST dataset. To carry out the experimental validation, we have equipped the following hardware facilities: Processor: We used an Intel Core i7-8700K, a powerful processor with strong computing power and multi-threaded processing ability, to meet the experimental needs. Memory: The computer is equipped with 64GB of DDR4 RAM, ensuring sufficient memory resources to keep the experiment running efficiently when processing large amounts of data. In this experiment, we used the following hyperparameter configurations: Learning Rate: We chose 0.01 or the default 0.001 as the learning rate, depending on the specific dataset. Batch Size (Batch Size): We used 64 as the batch size. Iteration Count (Epochs): We set the iteration count to 50. Optimizer: We chose the Adam optimizer. Loss Function: We chose the CategoricalCrossentropy function. We perform data preprocessing by resize the input images from  $28 \times 28$  to  $32 \times 32$ , equivalent to the usage of 10-qubits system on the quantum hardware. In order to better demonstrate the advantages of our classifier. This experiment was used to validate the model's ability to handle datasets with varying numbers of categories. We perform 2,3,4,5 classification on the sub-datasets {1, 5}, {3, 6}, {3, 8}, {3, 9}, {0, 3, 6}, {1, 3, 6}, {0, 3, 6, 9}, {0, 1, 3, 6, 9}, {0, 1, 2, 3, 4} to compare the performance of our method on the training and testing datasets. Total Sample Count, Total Sample Count, and Total Sample Count vary depending on the classification problem of the dataset. Each dataset contains a specific classification problem and is trained and tested using the corresponding samples. These details are crucial for understanding the structure of the dataset, evaluating model performance, and comparing different approaches. We perform model evaluation on different classes of classification problems, and the details of the evaluation dataset are provided in Table 1. Set relative to BinMLP(C) w/o BN, BinMLP(C) w/BN, FFNN(Q) w/o BN, FFNN(Q) w/BN, MLP(C) w/o BN, MLP(C) w/BN, QF-pNet w/o BN, QF-pNet w/BN, QF-hNet w/o BN, QF-hNet w/BN and other algorithms for classification accuracy. For MNIST actual data features, we design PQC as shown in Fig. 1. For multi-category classification, our classifier only needs to replicate the circuit in Fig. 1 by configuring different trainable parameters for the corresponding number of times, without any additional procedures required. This approach can be extended to support  $N$  classes, indicating good scalability of our classifier. For the MNIST sub-datasets {1, 5}, {3, 6}, {3, 8}, {3, 9}, {0, 3, 6}, {1, 3, 6}, {0, 3, 6, 9}, {0, 1, 3, 6, 9}, {0, 1, 2, 3, 4} the classification accurate on the different algorithms are shown in Fig. 2. The validation results in Fig. 2 clearly indicate that the classification accuracy of our proposed classifier on various subsets of the MNIST dataset, both for training and testing, is noticeably better than that of other quantum classification approaches, establishing it as the current state-of-the-art simulation result and surpassing (or matching) classical classifiers (e.g., MLPs) with a significant number of trainable parameters.

To further validate the scalability and classification accuracy of our classifier, we extend the sub-dataset to the entire MNIST dataset. For our proposed classifier, it only requires parallelizing ten channels to complete the ten-category classification of the entire dataset. At the same time, we adjust the number of layers of the parameterized quantum circuit from three to four, thereby completing the experimental verification. This experiment is used to verify the ability of the model to adjust the size of the scale. After 50 epochs of training and testing, the classification accuracy and loss function evolution of the classifier are plotted in Fig. 3.

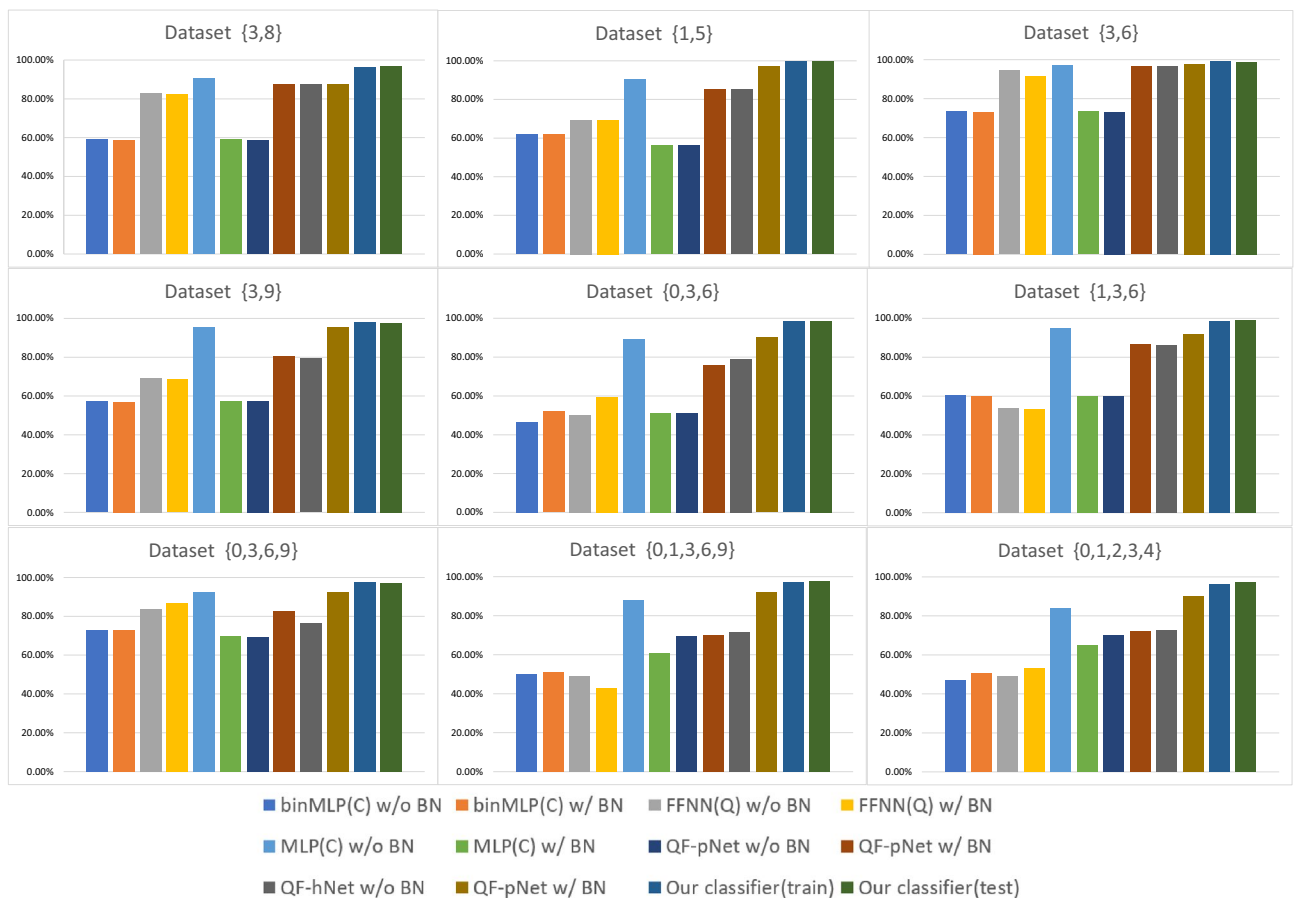
On the MNIST dataset, our classifier achieved a ten-category classification accuracy, where 50 epochs of iterative training were performed. For each epoch, the model was tested on the testing dataset, and the rate of change of classification accuracy and loss value on both the training and testing datasets throughout the iterations is plotted in Fig. 3. Our proposed classifier achieved a classification accuracy of 90% on both the training and testing datasets, and the classification accuracy of projection valued measure-based quantum machine learning for multi-category classification<sup>9</sup> was less than 80%. Our classifier was 10% higher in classification accuracy and showed good convergence by fast converging to the optimal model after 20 epochs. We have made the source code of all our experiments publicly available through the GitHub platform (<https://github.com/zhaoxing3/xiaodong/>), aiming to enable the general public to directly access, review, and validate the core aspects of our

Problem	Total sample count	Training sample count	Test sample count
2-class 1,5	13298	12163	2027
2-class 3,6	14017	12049	1968
2-class 3,8	13966	11982	1984
2-class 3,9	14099	12080	2019
3-class 0,3,6	20920	17972	2948
3-class 1,3,6	21894	18791	3103
4-class 0,3,6,9	27878	23921	3957
5-class 0,1,3,6,9	35755	30663	5092
5-class 0,1,2,3,4	35735	30596	5139
10-class 0,1,2,3,4,5,6,7,8,9	70000	60000	10000

**Table 1.** As far as the classification problem of the dataset is concerned, Total sample count, Training sample count and Test sample count vary and the table shows the detailed information for different datasets.

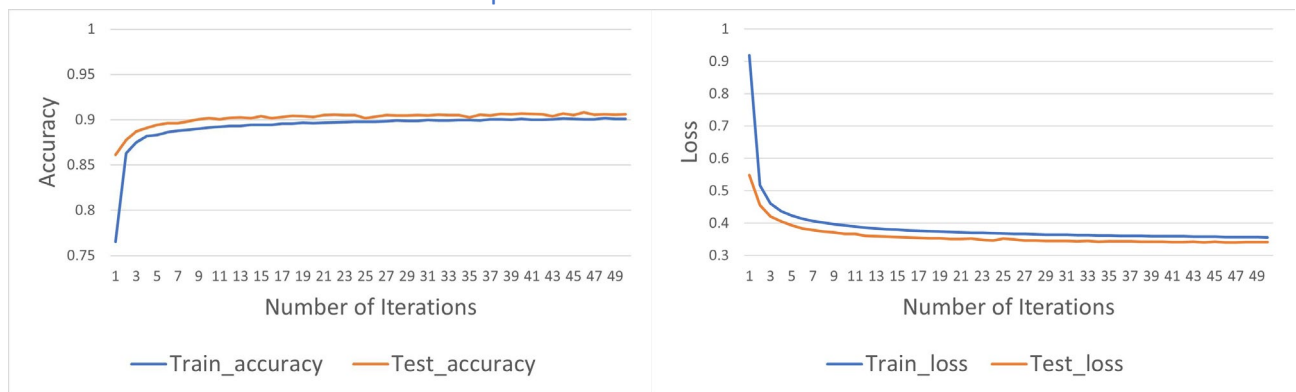


**Figure 1.** One channel PQC Design for Classifier over MNIST dataset, where  $\tilde{\theta}^i$  in the PQC could denote the weight matrix  $W$  in the traditional neural networks, PQC in the figure could be expressed by the following equation:  $W(\tilde{\theta}^i) = U_{net} U_l(\tilde{\theta}_3^i) U_{net} U_l(\tilde{\theta}_2^i) U_{net} U_l(\tilde{\theta}_1^i) U_{net} = \prod_{(i,j) \in E} CZ(i,j)$ ,  $U_l(\tilde{\theta}_j^i) = \otimes_{k=1}^{20,2} R_y(\tilde{\theta}_{j,k}^i) R_y(\tilde{\theta}_{j,k+1}^i)$ ,  $\otimes_{k=1}^{20,2}$  denote  $k$  from 1 to 20, increasing by 2 each time.



**Figure 2.** TBinMLP(C) w/o BN, BinMLP(C) w/ BN, FFNN(Q) w/o BN, FFNN(Q) w/ BN, MLP(C) w/o BN, MLP(C) w/ BN, QF-pNet w/o BN, QF-pNet w/ BN, QF-hNet w/o BN, QF-pNet w/ BN, Our classifier(train), Our classifier(test) Different algorithms on MNIST sub-datasets {1, 5}, {3, 6}, {3, 8}, {3, 9}, {0, 3, 6}, {1, 3, 6}, {0, 3, 6, 9}, {0, 1, 3, 6, 9}, {0, 1, 2, 3, 4} Classification Accuracy Display Graph.

## Classifier performance on the MNIST dataset



**Figure 3.** Through 50 epochs of training and testing, the classification accuracy and loss function variation of classifier. Classifier achieves 90% classification accuracy on both the training and testing Datasets. Classifier has good convergence by converging to the optimal model quickly after 20 epochs.

experimental environment configurations, data processing flow, and model training, thus ensuring the transparency of our research work and the reproducibility.

## Discussion

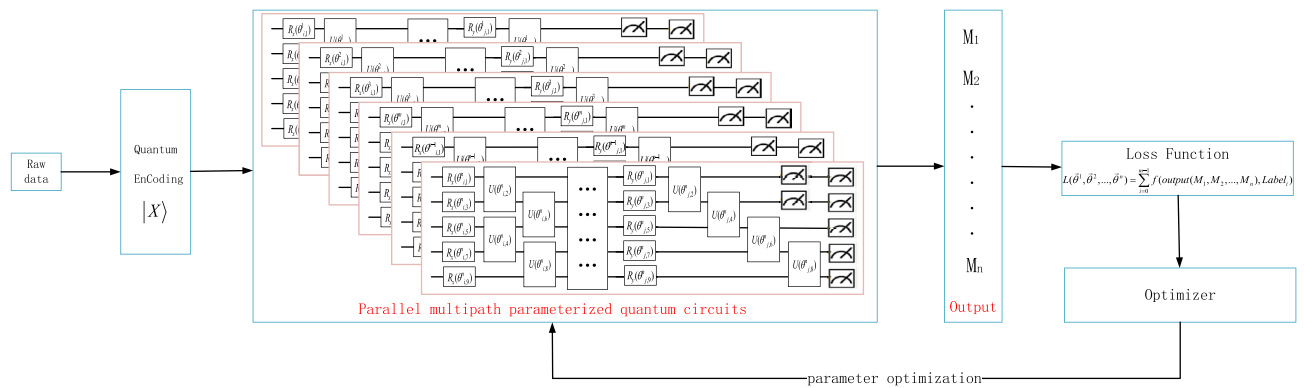
Multi-category classification is a crucial task in the field of machine learning. Conventional multi-category classification methods require significant amounts of data and computational resources, and suffer from issues such as overfitting. Existing quantum machine learnings mostly focus on binary classification problems, and the research on multi-category classification problems has poor classification accuracy and scalability. Therefore, SPQCC has better scalability while ensuring classifier performance. Our classifier requires PQC to have the same structure and scale, making it possible for the algorithm designer to design only one channel of PQC to complete the design of the whole classifier, greatly reducing the complexity of the design. Moreover, after designing a one-channel PQC, only the number of parallel PQCs equals to the classification class can be naturally extended to multi-category classifications, from which we could see that our proposed classifiers are more scalable. Additionally, our classifier realizes parallel execution of PQCs regardless of the number of classification classes, and the training time used is only related to the sample scale and the result and scale of one channel of PQC, but not the number of channels of PQCs executed, which has the same efficiency as that of the same scale of parameterized quantum circuit-based quantum machine learning algorithms. Our model employs the method of multiple PQCs. With each additional parameterized quantum circuit or increase in the number of layers of PQCs, the number of parameters in the whole model increases significantly, although the increase in the depth of the quantum circuit is not significant. Deeper circuits need to be designed to account for this additional parameter compared to the traditional single circuit model. However, designing deeper circuits poses a number of challenges. For instance, the coherence time of quantum bits, deeper circuit design may also introduce more errors and noise. This extended approach ensures the performance and stability of the model when dealing with more complex and larger datasets. Finally, our classifier is compared with existing classifiers in terms of classification effectiveness, and the experimental results illustrate that our proposed classifier exhibits excellent classification accuracy.

## Methods

### Classifier framework

Our proposed SPQCC belongs to a variant of quantum neural networks, which has four main components: quantum encoding, parallel multi-channel PQCs, quantum circuits measurement, and loss function minimization for parameters optimization. Its model is shown in Fig. 4.

**Quantum encoding:** Similar to other quantum machine learning algorithms, the implementation of SPQCC first requires mapping vectors to quantum states in the Hilbert space using features, which is generally achieved by quantum encoding for this process. The main encoding methods<sup>40</sup> at this stage are base encoding, amplitude encoding, repetitive amplitude encoding, rotational encoding, coherent state encoding, and so on. Amplitude encoding, approximate amplitude encoding, and rotational encoding are commonly used to map features from classical data to quantum states. Quantum amplitude coding is effective for certain problems, but implementing it can require a large number of quantum gate operations, posing computational complexity and scalability challenges. Approximate amplitude encoding<sup>41,42</sup>, as an important encoding method, is implemented by training shallow PQCs to encode given classical data into quantum circuits. Compared with quantum amplitude encoding, it uses fewer gates and shallower circuit depths. However, this encoding approach requires multiple training sessions to encode classical data, and thus also suffers from certain encoding efficiency issues. Rotational encoding is usually easier to understand and implement than other quantum encoding methods, and they are better resistant to noise and interference, making them more advantageous in quantum communication and quantum computing. But rotational encoding requires more quantum resources to accomplish computational tasks, potentially making them less practical in resource-limited systems. The choice of encoding method to implement feature



**Figure 4.** The SPQCC framework consists of four main components: quantum encoding, parallel multi-channel PQC, quantum circuits measurement, and loss function minimization for parameter optimization.

mapping usually depends on factors such as the model designer's experience, the characteristics of the original data, the number of bits in the quantum computer, and the decoherence time of the quantum system<sup>43</sup>. In order to meet the requirements of the validation experiment, we chose amplitude coding to realize feature mapping based on the characteristics of the experimental dataset and hardware conditions. In this paper, we focus on feature mapping using amplitude encoding, the core idea of which is to utilize the properties of quantum interference and quantum entanglement to encode vectors into the amplitude of a quantum state, which is processed in the form of a qubit, which has an exponential advantage in terms of memory. Amplitude encoding requires that the vectors are first normalized before encoding:  $x_{ij} = x_{ij} / \|\vec{x}_i\|$ , which has a general form:  $f(\vec{x}_i) = \sum_{j=1}^n x_{ij} |j\rangle$ . The main way to implement amplitude encoding is the iterative approach, where the basic process is that the encoding of new quantum states is accomplished by multiple control operations of the already encoded generated quantum states on those that need to be encoded until all the features of the vector are encoded.

**Parallel Multi-channel PQC:** Parallel multi-channel PQC serves as the core component of the classifier, each of which features the same structure and scale. The heart of PQC is built from trainable quantum gates containing parameters, which in turn constitutes modules of the unit layer through these parameter-bearing quantum gates. Based on actual needs, the modules of the unit layer are stacked to create PQC. For different application scenarios and problems, PQC of different structures and scales need to be designed according to the specific circumstances.

**Quantum Circuit Measurement:** Measurement is performed on each qubit in a quantum circuit<sup>44,45</sup>. In real quantum computers, the measurement is usually done through multiple iterations and the final results are tallied, which are presented in the form of vectors composed of quantum states and corresponding probabilities. Generally, to verify the correctness of the results of the method, quantum simulators could be leveraged, such as PennyLane<sup>46</sup>, Qiskit<sup>47</sup>, and TensorCircuit<sup>48</sup>.

**Parameters Optimization:** Similarly to quantum neural networks, SPQCC requires the definition of a loss function<sup>49</sup> to quantify the difference between the predicted and true values. Here, we choose the cross-entropy<sup>50</sup> loss function, and during training, the parameters in the classifier are iteratively updated by a gradient descent algorithm. In PQC models, the gradients of the computed parameters are typically estimated using traditional automatic differentiation methods, although they could also be calculated using the parameter-shift rule and gate decomposition<sup>51</sup> of quantum circuits.

In what follows, we will focus on the design of PQC, the quantum circuit measurement, loss function selection, and parameters optimization.

### Parallel multi-channel PQC

The PQCs in our proposed classifier all have the same structure and scale, making it possible to design only one channel PQC to meet the requirements of the classifier. This concept greatly simplifies and accelerates the design process of the classifier. The core of PQC is the trainable parameters contained in the Ansatz quantum gates, and the quantum gates containing trainable parameters in this paper are mainly composed of basic quantum gates of the form  $e^{-i\theta G/2}$  ( $G = \{X, Y\}$ ) and two-qubit gates  $U = e^{i\theta(Y \otimes Y)}$ ,  $U_1 = e^{i\theta(Z \otimes Z)}$ , where

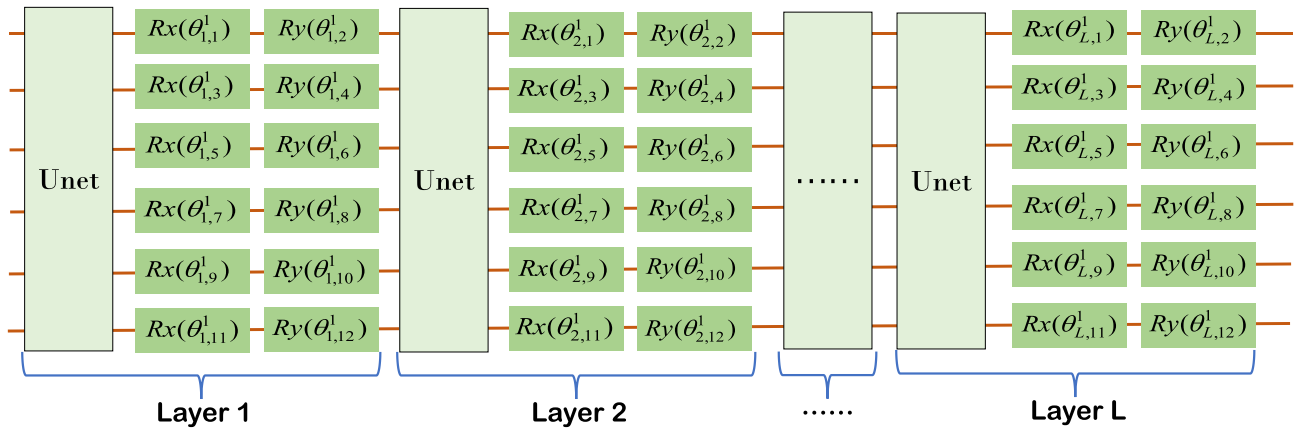
$$R_x(\theta) = e^{-i\theta X/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} X = \begin{bmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (1)$$

$$R_y(\theta) = e^{-i\theta Y/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} Y = \begin{bmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{bmatrix} \quad (2)$$

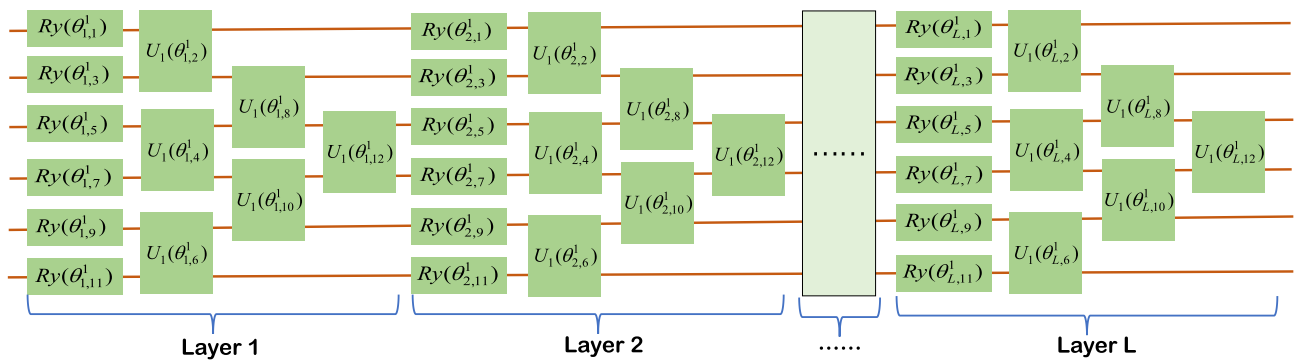
The matrices  $X, Y, Y \otimes Y, Z \otimes Z$  are obviously unitary matrices, which meet the requirement that the operations of a quantum system must be unitary matrices. PQC contains only  $R_x(\theta)$  and  $R_y(\theta)$ . The first step to do is to apply a *CNOT* gate between each pair of qubits to ensure that it generates quantum entanglement between qubits in Hilbert space, and the basic structure of the qubits in terms of six for example is shown in the Fig. 5. By including  $U(\theta)$  and  $U_1(\theta)$  in the basic layer, topological PQCs can realize quantum state entanglement without the need to apply *CNOT* gates between each pair of qubits. Illustrated here as an example using six qubits, two



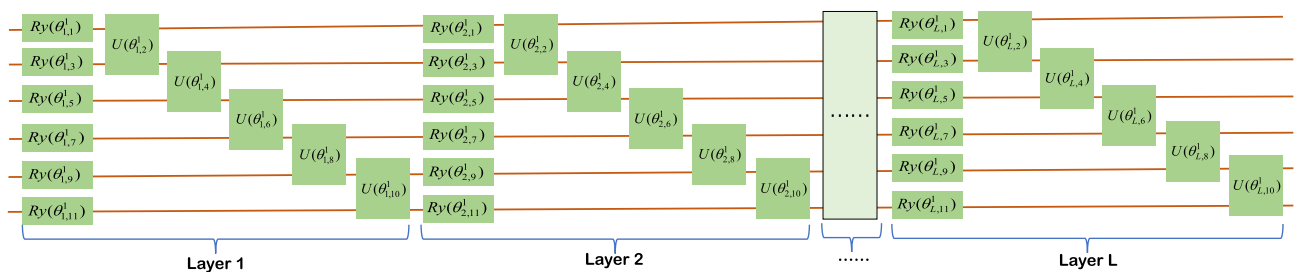
different topological basic structures are shown: the block structure in Fig. 6 and the ladder structure in Fig. 7. The trainable parameters  $\bar{\theta}$  in PQC are analogous to the adjustable weights  $W$  in conventional neural networks<sup>52</sup>, and the loss functions are constructed by measuring the expectation values of various observations on the PQCs. In PQCs, we initialize the parameters  $\bar{\theta} = (\bar{\theta}^1, \bar{\theta}^2, \dots, \bar{\theta}^n)$ , where  $n$  denotes the number of categories and  $\bar{\theta}^j$  represents the parameters of the  $j$ th channel of the PQC. In SPQCC  $\bar{\theta}^1, \bar{\theta}^2, \dots, \bar{\theta}^n$  are treated as a whole, and in the



**Figure 5.** The diagram of the first PQC, where  $\bar{\theta}^1$  in the PQC could denote the weight matrix  $W$  in the traditional neural networks, PQC in the figure could be expressed by the following equation:  $W(\bar{\theta}^1) = U_{net} U_L(\bar{\theta}^1_L) U_{net} U_{L-1}(\bar{\theta}^1_{L-1}) \dots U_{net} U_1(\bar{\theta}^1_2) U_{net} U_1(\bar{\theta}^1_1)$  ( $L$  is the number of layers),  $U_{net} = \prod_{(i,j) \in E} CZ(i,j)$ ,  $U_L(\bar{\theta}^1_L) = \otimes_{k=1}^{n/2} R_x(\bar{\theta}^1_{L,k}) R_y(\bar{\theta}^1_{L,k+1})$ ,  $\otimes_{k=1}^{n/2}$  denote  $k$  from 1 to  $n$ , increasing by 2 each time, and  $n$  is the number of parameters in each layer.



**Figure 6.** The diagram of the first PQC, where  $\bar{\theta}^1$  in the PQC could denote the weight matrix  $W$  in the traditional neural networks, PQC in the figure could be expressed by the following equation:  $W(\bar{\theta}^1) = \otimes_{j=1}^L U_L(\bar{\theta}^1_j)$  ( $L$  is the number of layers),  $U_L(\bar{\theta}^1_j) = \otimes_{k=1}^{n/2} R_y(\bar{\theta}^1_{j,k}) U_1(\bar{\theta}^1_{j,k+1})$ ,  $\otimes_{k=1}^{n/2}$  denote  $k$  from 1 to  $n$ , increasing by 2 each time, and  $n$  is the number of parameters in each layer.



**Figure 7.** The diagram of the first PQC, where  $\bar{\theta}^1$  in the PQC could denote the weight matrix  $W$  in the traditional neural networks, PQC in the figure could be expressed by the following equation:  $W(\bar{\theta}^1) = \otimes_{j=1}^L U_L(\bar{\theta}^1_j)$  ( $L$  is the number of layers),  $U_L(\bar{\theta}^1_j) = \otimes_{k=1}^{n/2} R_y(\bar{\theta}^1_{j,k}) U_1(\bar{\theta}^1_{j,k+1})$ ,  $\otimes_{k=1}^{n/2}$  denote  $k$  from 1 to  $n$ , increasing by 2 each time, and  $n$  is the number of parameters in each layer.

process of parameters optimization,  $\vec{\theta} = (\vec{\theta}^1, \vec{\theta}^2, \dots, \vec{\theta}^n)$  need to be updated for each channel for each optimization step. In summary, the design of PQC for a classifier allows us to determine the structure and scale of PQC according to the needs of different application scenarios, and then select appropriate quantum gates from the parameter-containing quantum gates to complete the design. This process is full of possibilities, and the diversity of PQCs designed are also an attraction of our classifier. Finally, assuming that the number of categories is  $m$ , the design of the classifier can be completed by parallelizing the  $m$  designed PQC.

### Quantum circuit measurement

Our measurement is performed on PQC where the qubit state is measured in a standard basis (Z-basis)<sup>53,54</sup>. The Z-basis is the basis used to measure whether a qubit is in the  $|0\rangle$  state or the  $|1\rangle$  state, which is determined by the state of the wavefunction of the qubit prior to the measurement. The process of performing a measurement is to project a qubit from the superposition state onto the standard basis states  $|0\rangle$  and  $|1\rangle$ , and the measurement causes the state of the qubit to undergo a collapse to a fixed standard basis state  $|0\rangle$  or  $|1\rangle$ , we record the probability of the state collapsing to  $|0\rangle$ , and then sum the probabilities of all the qubits collapsing to  $|0\rangle$  of each qubit of PQC as output, corresponding to the  $n$ -category classification, and the whole outputs of the classifier results in  $M_1, M_2, \dots, M_n$ .

The specific measurement is, for the measurement under the Z-basis, the two corresponding measurement operators are  $Z_0 = |0\rangle\langle 0|$ ,  $Z_1 = |1\rangle\langle 1|$ , which would be seen to be self-adjoint, i.e.,  $Z_0^\dagger = Z_0$ ,  $Z_1^\dagger = Z_1$ , and satisfy  $Z_0^2 = Z_0$ ,  $Z_1^2 = Z_1$ . Let the state of a qubit when it is measured be  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ , and the state in which the measurement result is 0 be  $p(0) = \langle\varphi|Z_0|\varphi\rangle = \langle\varphi|Z_0|\varphi\rangle = |\alpha|^2$ . Let one-channel PQC contain  $m$  qubits, then according to our quantum circuit measurement method, the output result of each quantum circuit is:  $M_i = \sum_{j=0}^m p_j(0)$ .

### Loss function design and parameters optimization

SPQCC, like quantum neural networks, requires the definition of loss function to measure the difference between the predicted value and the true value. As mentioned in parallel PQCs, the qubits of each PQC are measured at each qubit on Z. That is, each qubit is projected onto the ground state separately, and the probability of collapsing to  $|0\rangle$  is calculated as  $P_i$ . After adding all the  $P_i$ s of the PQC together and the sum  $M_i$  as the output, the outputs of the whole classifier are  $M_1, M_2, \dots, M_n$ . After combining the results and passing them through the SoftMax<sup>55,56</sup> as the final outputs. SoftMax mainly transforms the output value of the multi-category classification into a probability distribution in the range of  $[0,1]$  with sum 1. At the same time we use one-hot<sup>57,58</sup> to encode the sample labels, which provides the basis for generating the loss function for parameters optimization.

$\text{Soft max}(M_i) = \frac{e^{M_i}}{\sum_{c=1}^C e^{M_c}}$ ,  $M_i$  is the output of the  $i$ -th PQC, and  $C$  is the number of parallel PQCs (the number of classes to be classified). For the loss function we choose the cross-entropy, and the SoftMax maps the output of the classifier  $M_1, M_2, \dots, M_n$  to a vector  $\hat{y}$ ,  $\hat{y} = \text{Soft max}(M_1, M_2, \dots, M_n)$ , which we could consider as the estimated conditional probability of each class for an arbitrary sample  $x$ .

Let assume that the dataset  $\{X, Y\}$  has  $n$  samples, where the samples indexed as  $i$  are composed of a feature vector  $x_i$  and a corresponding vector  $y_i$  of one-hot label. Then for any  $x$  corresponding to the true label  $y$  and the result  $\hat{y}$  predicted by the classifier, we define the cross-entropy loss function as  $l(y, \hat{y}) = -\sum_{i=1}^n y_i \log \hat{y}_i$ . We use the one-hot labels for encoding, so in the vector  $y$ , only one component is 1 and the rest are 0. We could then write the loss function as:  $l(y, \hat{y}) = -\sum_{i=1}^n y_i \log \hat{y}_i$ . According to the definition of SoftMax, the loss function  $l(y, \hat{y}) = -\sum_{i=1}^n y_i \log \hat{y}_i$  could be expanded as:

$$l(y, \hat{y}) = -\sum_{i=1}^n y_i \log \frac{e^{o_i}}{\sum_{c=1}^n e^{o_c}} = \sum_{i=1}^n \log \sum_{c=1}^n e^{o_c} - \sum_{i=1}^n y_i o_i = \log \sum_{c=1}^n e^{o_c} - \sum_{i=1}^n y_i o_i \quad (3)$$

The loss function is derived for any prediction  $o_i$ :

$$\frac{\partial l(y, \hat{y})}{\partial o_i} = \frac{e^{o_i}}{\sum_{c=1}^n e^{o_c}} - y_i = \text{Soft max}(o)_i - y_i \quad (4)$$

From the above equation, the derivative is the difference between the output obtained by the multi-category classifier and the true value.

The parameters in the classifier are updated iteratively during training through a gradient descent algorithm. The gradient of the computed parameters for PQC is typically estimated using the traditional automatic differentiation method, or the parameter-shift rule of the quantum circuit can be applied. The PQC, with the phase in the quantum gate serving as the primary training parameters, causes the initial quantum state  $|\varphi_{in}\rangle$  to evolve into the desired quantum final state  $|\varphi_{out}\rangle$  through iterative training, i.e.,  $|\varphi_{out}\rangle = W(\vec{\theta})|\varphi_{in}\rangle$ , where  $W(\vec{\theta})$  represents the corresponding quantum state of the PQC. The optimization of the parameters focuses on reducing the deviation between the predicted and true values, quantified by the loss function  $l(y, \hat{y}) = -\sum_{i=1}^n y_i \log \hat{y}_i$ , where  $\hat{y}_i$  and  $y_i$  represent the predicted and true values corresponding to  $x_i$ , respectively. Through continuous training, the loss function is minimized, or the error of its loss function is brought within an acceptable range. For the simulations in this paper, we utilize the traditional automatic differentiation method to compute the gradient solutions.

Frontier research<sup>1,59-61</sup> has made it clear that quantum neural networks, through the superposition and entanglement properties of qubits, have demonstrated significant advantages compared to classical neural networks exhibit significant advantages, including fewer parameters, lower resource requirements, faster training, and lower risk of overfitting, which enable efficient representation of complex functional relationships. Our model

incorporates the core advantages of quantum neural networks. In addition, our model adopts a multiplexed PQC architecture, assuming that each channel contains  $m$  parameters and consists of  $c$  channels, resulting in a total number of  $m * c$  parameters. In terms of the use of qubits, the number of qubits required for amplitude encoding is  $c * \lceil \log n \rceil$  if a parallel structure is used, and only  $\lceil \log n \rceil$  qubits are required if a no-parallel structure is used, where  $n$  is the data dimension. In this case, the required resources are exponentially reduced compared to classical neural networks.

## Conclusion

PQC is one of the mainstream models of quantum machine learning, which mainly stacks a set of quantum gates containing parameters together to form a model. Optimization of these parameters through training is necessary to achieve the desired output. Multi-category classification has numerous applications and is a worthwhile research problem. SPQCC, which we propose, takes advantage of the parallelism of PQCs, merges the measurements of the model as the final output of the classifier, and minimizes the cross-entropy loss function for optimizing the classifier's parameters. This method allows for establishing the same number of PQCs corresponding to the number of classes according to the number of classes, using the same measurement method. The PQCs need only be designed once in the design process of the classifier, with the same design complexity as designing a single PQC. Additionally, the time complexity is equivalent to that of a single PQC classifier using parallel processing and multiple simulators. However, our designed classifier fully utilizes the advantages of quantum computing and has better scalability. We tested it on MNIST, and the classification accuracy is similar to that of traditional methods. Our findings provide new ideas and methods for solving multi-category classification problems using PQC, and contribute to the performance and efficiency of quantum algorithms for solving multi-category classification problems. In future work, we will build on this foundation to achieve multi-category classification using quantum perceptron machines.

## Data availability

All data generated or analysed during this study are included in this published article.

Received: 22 November 2023; Accepted: 1 July 2024

Published online: 10 July 2024

## References

1. Abbas, A. *et al.* The power of quantum neural networks. *Nat. Comput. Sci.* **1**, 403–409. <https://doi.org/10.1038/s43588-021-00084-1> (2021).
2. Jiang, W., Xiong, J. & Shi, Y. A co-design framework of neural networks and quantum circuits towards quantum advantage. *Nat. Commun.* **12**, 579 (2021).
3. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001. <https://doi.org/10.1088/2058-9565/ab4eb5> (2019).
4. Bokhan, D., Mastiukova, A. S., Boev, A. S., Trubnikov, D. N. & Fedorov, A. K. Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. *Front. Phys.* **10**, 1069985. <https://doi.org/10.3389/fphy.2022.1069985> (2022).
5. Avinash Chalamuri, R. K. & Manoj, B. S. A hybrid classical-quantum approach for multi-class classification. *Quantum Inf. Process.* **20**, 119. <https://doi.org/10.1007/s1128-021-03029-9> (2021).
6. Huang, K. *et al.* Quantum generative adversarial networks with multiple superconducting qubits. *NPJ Quantum Inf.* **7**, 165. <https://doi.org/10.1038/s41534-021-00503-1> (2021).
7. Tak Hur, L. K. & Park, D. K. Quantum convolutional neural network for classical data classification. *Quantum Mach. Intell.* <https://doi.org/10.1007/s42484-021-00061-x> (2022).
8. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors (2018). [arXiv:1802.06002](https://arxiv.org/abs/1802.06002).
9. Yun, W. J., Baek, H. & Kim, J. Projection valued measure-based quantum machine learning for multi-class classification (2022). [arXiv:2210.16731](https://arxiv.org/abs/2210.16731).
10. Havlíček, V. & Córcoles, A. D. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212. <https://doi.org/10.1038/s41586-019-0980-2> (2019).
11. Tang, E. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, (ACM, 2019). <https://doi.org/10.1145/3313276.3316310>
12. Linke, N. M. *et al.* Quantum machine learning. *Nat. Nanotechnol.* **15**, 607. <https://doi.org/10.1038/nnano.2019.242> (2017).
13. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J. & Killoran, N. Quantum embeddings for machine learning (2020). [arXiv:2001.03622](https://arxiv.org/abs/2001.03622).
14. Nguyen, N. & Chen, K.-C. Quantum embedding search for quantum machine learning. *IEEE Access* **10**, 41444–41456. <https://doi.org/10.1109/access.2022.3167398> (2022).
15. Nguyen, N. & Chen, K.-C. Bayesian quantum neural networks. *IEEE Access* **10**, 54110–54122. <https://doi.org/10.1109/ACCESS.2022.3168675> (2022).
16. Bandhu, A. & Roy, S. S. Classifying multi-category images using deep learning : A convolutional neural network model. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 915–919. <https://doi.org/10.1109/RTEICT.2017.8256731> (2017).
17. de la Torre, J., Puig, D. & Valls, A. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognit. Lett.* **105**, 144–154. <https://doi.org/10.1016/j.patrec.2017.05.018> (2018).
18. Meng, N., Lam, E. Y., Tsia, K. K. & So, H. K.-H. Large-scale multi-class image-based cell classification with deep learning. *IEEE J. Biomed. Health Inform.* **23**, 2091–2098. <https://doi.org/10.1109/JBHI.2018.2878878> (2019).
19. Mari, A., Bromley, T. R., Izaac, J., Schuld, M. & Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **4**, 340. <https://doi.org/10.22331/q-2020-10-09-340> (2020).
20. Schuld, P. M. Quantum ensembles of quantum classifiers. *Sci. Rep.* **8**, 2772 (2018).
21. Li, X. *et al.* An all-pair quantum svm approach for big data multiclass classification. *Knowl. Based Syst.* <https://doi.org/10.1007/s1128-021-02519-2> (2022).
22. Li, W. & Deng, D.-L. Recent advances for quantum classifiers. *Sci. China Phys. Mech. Math* **65**, 220301. <https://doi.org/10.1007/s11433-021-1793-6> (2021).



23. Liu, Y. *et al.* Parameterized quantum circuits as machine learning models. *bioRxiv* <https://doi.org/10.1101/2021.11.17.427859> (2021).
24. Du, Y., Hsieh, M.-H., Liu, T. & Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* <https://doi.org/10.1103/physrevresearch.2.033125> (2020).
25. Carr, A. *et al.* Neural-network quantum state tomography. *Phys. Rev. Res.* **3**, 033057. <https://doi.org/10.1103/PhysRevResearch.3.033057> (2021).
26. Kwak, Y., Yun, W. J., Jung, S. & Kim, J. Quantum neural networks: Concepts, applications, and challenges (2021). [arXiv:2108.01468](https://arxiv.org/abs/2108.01468).
27. Jarrod, R., McClean, V. & Sergio Boixo, N. S. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 1–9 (2018).
28. Iris Cong, S. C. & Lukin, M. D. Quantum convolutional neural networks. *Nat. Phys.* **15**, 1273–1278 (2019).
29. Di, S. *et al.* Amplitude transformed quantum convolutional neural network. *Appl. Intell.* **53**, 20863–20873. <https://doi.org/10.1007/s10489-023-04581-w> (2023).
30. Chen, S. Y.-C., Yoo, S. & Fang, Y.-L. L. Quantum long short-term memory (2020). [arXiv:2009.01783](https://arxiv.org/abs/2009.01783).
31. Cao, Y. *et al.* Linear-layer-enhanced quantum long short-term memory for carbon price forecasting. *Quantum Mach. Intell.* **5**, 26. <https://doi.org/10.1007/s42484-023-00115-2> (2023).
32. Huang, K. *et al.* Quantum generative adversarial networks with multiple superconducting qubits. *NPJ Quantum Inf.* **7**, 165. <https://doi.org/10.1038/s41534-021-00503-1> (2021).
33. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *NPJ Quantum Inf.* **5**, 103. <https://doi.org/10.1038/s41534-019-0223-2> (2019).
34. Shamsuddin, M., Abdul-Rahman, S. & Mohamed, A. Exploratory analysis of mnist handwritten digit for machine learning modeling. In Yap, B., Mohamed, A. & Berry, M. (eds.) *Soft Computing in Data Science. SCDS 2018. Communications in Computer and Information Science*, vol. 937 (Springer, Singapore, 2019).
35. Mu, N. & Gilmer, J. Mnist-c: A robustness benchmark for computer vision (2019). [arXiv:1906.02337](https://arxiv.org/abs/1906.02337).
36. Koonce, B. Mnist: 1d neural network. In *Convolutional Neural Networks with Swift for Tensorflow*, (Apress, Berkeley, CA, 2021).
37. Deng, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Process. Mag.* **29**, 141–142 (2012).
38. Schott, L., Rauber, J., Bethge, M. & Brendel, W. Towards the first adversarially robust neural network model on mnist (2018). [arXiv:1805.09190](https://arxiv.org/abs/1805.09190).
39. Jiang, W. & Xiong, J. S. A co-design framework of neural networks and quantum circuits towards quantum advantage. *Nat. Commun.* **12**, 576. <https://doi.org/10.1038/s41467-020-20729-5> (2021).
40. Schuld, M. Supervised quantum machine learning models are kernel methods (2021). [arXiv:2101.11020](https://arxiv.org/abs/2101.11020).
41. Nakaji, K. *et al.* Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Phys. Rev. Res.* **4**, 023136. <https://doi.org/10.1103/physrevresearch.4.023136> (2022).
42. Mitsuda, N. *et al.* Approximate complex amplitude encoding algorithm and its application to data classification problems. *Phys. Rev. A* **109**, 052423 (2023).
43. Schuld, M., Sweke, R. & Meyer, J. J. The effect of data encoding on the expressive power of variational quantum machine learning models. *ArXiv* (2020).
44. Briegel, H., Browne, D. & Dür, W. Measurement-based quantum computation. *Nat. Phys.* **5**, 19–26. <https://doi.org/10.1038/nphys1157> (2009).
45. Ware, B. & Vasseur, R. Measurements make the phase. *Nat. Phys.* **17**, 298–299. <https://doi.org/10.1038/s41567-020-01131-w> (2021).
46. Bergholm, V. *et al.* PennyLane: Automatic differentiation of hybrid quantum-classical computations (2022). [arXiv:1811.04968](https://arxiv.org/abs/1811.04968).
47. Gadi Aleksandrowicz, P. B. Thomas Alexander. Qiskit: An open-source framework for quantum computing (2019).
48. Zhang, S.-X., Allcock, J. & Wan, Z.-Q. TensorCircuit: A quantum software framework for the NISQ era. *Quantum* **7**, 912. <https://doi.org/10.22331/q-2023-02-02-912> (2023).
49. Barron, J. T. A general and adaptive robust loss function (2019). [arXiv:1701.03077](https://arxiv.org/abs/1701.03077).
50. Boer, P. T. D., Kroese, D. P., Mannor, S. & Rubinstein, R. Y. A tutorial on the cross-entropy method. *Ann. Oper. Res.* **134**, 19–67 (2005).
51. Crooks, G. E. Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition (2019). [arXiv:1905.13311](https://arxiv.org/abs/1905.13311).
52. Clothiaux, B. *Neural Networks and Their Applications*, vol. 29 of *The GeoJournal Library*, Chap. two, 11–18 (Springer, Dordrecht, 1994).
53. Takeuchi, Y., Morimae, T. & Hayashi, M. Quantum computational universality of hypergraph states with pauli-x and z basis measurements. *Sci. Rep.* **9**, 13585 (2019). [arXiv:1809.07552](https://arxiv.org/abs/1809.07552).
54. James, D. F. V., Kwiat, P. G., Munro, W. J. & White, A. G. Measurement of qubits. *Phys. Rev. A* **64**, 052312. <https://doi.org/10.1103/PhysRevA.64.052312> (2001).
55. Yao, Y. & Wang, H. Optimal subsampling for softmax regression. *Stat. Papers* <https://doi.org/10.1007/s00362-018-01068-6> (2019).
56. Zhang, H., Wang, X. & He, Z. Weighted softmax loss for face recognition via cosine distance (2018).
57. Alaya, M. Z., Bussy, S., Gaïffas, S. & Guilloux, A. Binarisity: A penalization for one-hot encoded features in linear supervised learning (2019). [arXiv:1703.08619](https://arxiv.org/abs/1703.08619).
58. Rodríguez, P., Bautista, M. A., González, J. & Escalera, S. Beyond one-hot encoding: Lower dimensional target embedding. *Image Vis. Comput.* **75**, 21–31. <https://doi.org/10.1016/j.imavis.2018.04.004> (2018).
59. Huang, H. *et al.* Power of data in quantum machine learning. *Nat. Commun.* **12**, 2631. <https://doi.org/10.1038/s41467-021-22539-9> (2021).
60. Preskill, J. Quantum computing in the nisq era and beyond. *Quantum* **2**, 79 (2018).
61. Coles, P. Seeking quantum advantage for neural networks. *Nat. Comput. Sci.* **1**, 389–390. <https://doi.org/10.1038/s43588-021-00088-x> (2021).

# Acknowledgements

The authors acknowledge the financial support from Major Science and Technology Projects in Henan Province, China, Grant No.: 221100210600.

# Author contributions

X.D.: Formal Analysis, Writing—Original Draft, Writing—Review and Editing; Z.S.: Formal Analysis, Writing—Original Draft, Writing—Review and Editing; J.X.: Methodology, Supervision; Y.H.: Visualization, Validation, Investigation; T.Y.: Data curation, Validation, Visualization; Z.S.: Funding acquisition, Methodology, Supervision.

### Competing interests

The authors declare no competing interests.

### Additional information

**Correspondence** and requests for materials should be addressed to Z.S.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024