IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 November 2024, accepted 29 January 2025, date of publication 17 February 2025, date of current version 3 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3543164

RESEARCH ARTICLE

An Improved Convolutional Neural Networks: Quantum Pseudo-Transposed Convolutional Neural Networks

LI HAI[1], CHEN LIANG[1], HAO YAMING[1], YU WENLI[2], AND SHI FENGQUAN[1]
[1]School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai 264005, China
[2]School of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China

Corresponding author: Li Hai (shenghuo2003@126.com)

This work was supported by Yantai Science and Technology Innovation Development Program under Grant 2022JCYJ044.

ABSTRACT Recent advancements in quantum machine learning have spurred the development of hybrid quantum-classical convolutional neural networks (HQCCNNs), which have demonstrated promising potential for image classification tasks. Building on the operational principles of classical transposed convolutional neural networks (CNNs), we introduce a novel quantum variant: the Quantum Pseudo-Transposed Convolutional Neural Network (QPTCNN). The QPTCNN adapts the concept of classical transposed CNNs to the quantum domain, leveraging a hybrid quantum-classical framework that combines a quantum convolutional layer with a classical fully connected layer. In the QPTCNN, the quantum convolutional layer emulates a transposed convolution operation, ensuring that the output feature map retains the same dimensions as the input image. This is accomplished using rotational angle encoding and a ring-structured quantum circuit, interconnected by two-qubit control gates such as *CNOT* and *CRY* gates, facilitating efficient quantum convolution. We evaluated the performance of the QPTCNN on the MNIST and Fashion-MNIST datasets, with two distinct versions of the model: Model A, which utilizes a *CNOT*-gate entanglement circuit, and Model B, which employs a *CRY*-gate entanglement circuit. The results demonstrate that both Model A and Model B achieve strong performance across the datasets. However, Model A outperforms Model B, achieving higher classification accuracy and lower loss compared to earlier models. These findings suggest that the QPTCNN is highly capable of learning and extracting relevant feature information from input images, making it well-suited for high-performance image classification tasks. This work represents a significant advancement in quantum-enhanced image classification.

INDEX TERMS Convolutional neural networks, quantum pseudo-transposed convolutional neural networks, quantum circuits, transposed convolution.

I. INTRODUCTION

The convolutional neural network (CNN), first proposed by LeCun et al. [1] in 1989, is one of the most influential deep learning algorithms. Its structural features, including local connectivity, weight sharing, and downsampling, have enabled its use in image processing tasks. With advancements in deep learning theory and numerical computation technology, CNNs have undergone rapid development. Consequently, several new CNN models have been introduced, such as AlexNet [2], VGGNet [3], and GoogLeNet et al. [4], which have demonstrated impressive performance across various aspects of computer vision, including image classification [5], [6], [7], object detection [8], [9], [10], and semantic segmentation [11], [12], [13].

Achieving the highest possible data processing speed has always been a primary objective for researchers and engineers. In addition to these algorithms, the hardware of computing systems plays a crucial role in information processing. The rapid advancement of nano-integration technology has significantly contributed to the miniaturization of integrated circuits, thereby enhancing the operational speed

The associate editor coordinating the review of this manuscript and approving it for publication was Tony Thomas.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

of electronic computers. However, constrained by Moore's Law [14], the operating speed of electronic computers is expected to peak within the next few years [15]. Fortunately, the advent of quantum computers that leverage quantum phenomena such as entanglement and coherence holds the promise of exponentially accelerating information processing [16], [17]. This emerging technology is a powerful tool for tackling complex problems and managing vast datasets. Although fully universal quantum computers have not yet been realized, existing quantum computer models, such as Google's "Sycamore" superconducting quantum computer [18] and the "Jiuzhang-3" optical quantum computers [19], have already demonstrated quantum supremacy in specific tasks, including random circuit sampling and Gaussian boson sampling. The distinct computational advantages of quantum computing offer new possibilities in machine learning. Consequently, exploring methods for implementing machine learning algorithms on quantum computers, as well as effective integration strategies between quantum and classical computing, has become a prominent research area for advancing efficient information processing.

Recent studies have demonstrated that it is feasible and beneficial to replace certain components of conventional machine learning systems with their quantum counterparts [20], [21], [22], [23], such as substituting classical convolutional kernels with quantum circuits [24], [25], [26], [27], [28]. Therefore, the construction of quantum versions of CNNs has drawn considerable attention. Recently, some significant effort has been made to explore quantum CNN (QCNN). For instance, Cong et al. [24] proposed a full QCNN including convolutional, pooling, and fully connected layers. In addition, the QCNN can accurately identify quantum states associated with topological phases protected by symmetry, as evidenced by the training results for near-term quantum devices. However, this QCNN model is limited to the processing of quantum data and cannot directly handle classical datasets. To address this limitation, Henderson et al. [29] replaced the classical convolutional layers in CNNs with quantum convolutional layers, which use random quantum circuits for local transformations to perform convolutional operations. These quantum layers generate feature maps analogous to those produced by classical convolutions and can be highly effective for image classification tasks. Building on Henderson's work, integrating the principles of CNNs into QCNNs has become a promising research direction. For instance, Monnet et al. [30] developed a quantum version of a pooling layer to enhance network expressiveness. Recently, Chen [31] applied the concept of dilated convolution in classical CNNs to quantum convolutional layers to optimize QCNNs. This scheme allows the quantum convolution operation to capture more features and information by expanding the receptive field of the convolutional kernel with a fixed kernel size. Based on the quantum cavity convolution method [22], Ovalle-Magallanes et al. [32] proposed a learnable rotation angle encoding scheme implemented by multiple quantum convolutional circuits, which also led to improved classification accuracy on the MNIST dataset.

It is well known that the classification accuracy of convolutional neural networks (CNNs) in image classification tasks fundamentally depends on the capacity of CNNs to extract the feature information of an image, which in turn augments its expressiveness. Classical transposed convolution techniques facilitate the expansion of low-dimensional feature maps into higher-dimensional spaces, which can lead to more feature information of the input image captured by the transposed CNNs [33]. Inspired by classical transposed CNNs, we extend the concept of transposed convolution to QCNNs and assess their possible potential in image classification tasks. However, similar studies have not been reported.

Compared to traditional CNNs, the obvious feature of classical transposed CNNs is that the output feature map matches the input image (i.e., the dimensions of the input image and its feature map are identical or approximate). Such a feature is determined by the special structure of the convolutional layers of transposed CNNs, that is, standard convolution layers plus a transposed convolution layer, which is the duty to implement the inverse-like convolutional operation. Drawing on the characteristics of classical transposed CNN, we propose a quantum variant. Specifically, we replace the standard convolution operation in classical transposed CNNs with a scaling operation on the input image by invoking a scaling function to generate a scaled-down image with the same size as that obtained by the standard convolution operation. The transposed convolution layer in a classical transposed CNN is instead a quantum convolution circuit that performs a transposed-like convolution operation. The architectures of QCNNs in [24] and [30] are similar to our model, which has a hybrid quantum–classical CNN framework, where a classical fully connected layer is placed at the end of the network. Because the quantum variant emulates the classic transposed CNN to behave similarly (feature map matching with the input image), we refer to it as a quantum pseudo-transposed CNN (QPTCNN). Based on the MNIST and Fashion-MNIST datasets, we evaluated the performance of the QPTCNN, in which the entanglement circuits with the *CNOT*-gates ring connection structure and the *CRY*-gates ring connection structure are individually considered. For simplicity, the QPTCNN is designed with the *CNOT*-gate entanglement circuit as Model A and that with *CRY*-gates as Model B. The results indicate that under specific learning rates, Model A achieves optimal classification accuracies of 0.91 on the MNIST dataset and 0.835 on the Fashion-MNIST dataset, which surpasses those of the optimal QCNN model in [32] for the MNIST dataset by 1.11% and in [31] on the Fashion-MNIST dataset by 3.75%. Notably, although Model B performs slightly weaker than Model A on the Fashion-MNIST dataset, it still maintains significant advantages in terms of accuracy and loss values compared with those in [31] and [32]. This suggests that the QPTCNN possesses strong

expressivity, ensuring that it is well-suited for high-accuracy image classification tasks.

II. BACKGROUND OF QUANTUM COMPUTING

A. QUANTUM STATES AND QUANTUM GATES

Quantum bits are the fundamental units of information processing in quantum computers. Unlike classical bits, which only represent two different states commonly denoted as 0 or 1, qubits can not only represent states of 0 and 1 but also remain in superposition states of these two states, such as

$$|\psi\rangle = \alpha\ |0\rangle + \beta\ |1\rangle\,, \tag{1}$$

where $|\psi\rangle$ represents the superposition state of qubit, $|0\rangle = (1, 0)$and $|1\rangle = (0, 1)$ are quantum basis states in Dirac notation, respectively, α and β individually correspond to the probability amplitudes of qubit at states $|0\rangle$ and $|1\rangle$, and $|\alpha|^2 + |\beta|^2 = 1$ holds. Similarly, for an N-qubit system spanned by an 2^N-dimensional Hilbert space, the general superposition state can be expressed as

$$|\psi\rangle = \sum_{n=1}^{2^N} \alpha_n\ |n\rangle\,, \tag{2}$$

where $|n\rangle$and α_n represent the n-th basis vector of the N-qubit system, and the corresponding probability amplitude $\alpha_n$satisfies the normalization condition $\sum_{n=1}^{2^N} |\alpha_n|^2 = 1$. Quantum gates are the basic units for implementing quantum computing. Common quantum gates include the Hadamard gate (H-gate), *CNOT*-gates, and rotation-gates. These gates have different functions in the transition of quantum states. By applying a single quantum gate or composite gate to a quantum system, the expected quantum state (final state) can be obtained from the initial state of the quantum system. The features of the common gates, including the math presentation and map properties, are given in Appendix A.

B. QUANTUM CIRCUITS

Quantum computing is implemented using quantum circuits, which typically consist of many quantum gates with adjustable parameters. By acting on qubits in a particular order, a quantum circuit can perform various operations and realize certain quantum algorithms. The general circuit architecture of quantum computing includes three modules, as shown in Figure FIGURE 1: encoding, entanglement, and measurement layers. The specific features of each module are as follows:

1) ENCODING

A quantum circuit is responsible for processing quantum information. Therefore, classical data must be transformed into quantum states (i.e., to encode classical information into quantum information) when classical datasets are used within a quantum circuit.

A general framework for the encoding process of a classical dataset $X = \{x_i\}$ into a quantum state can be described as follows:

$$X \rightarrow |\psi_X\rangle = E\,(X)|\ \psi_0\rangle\,, \tag{3}$$

where $|\psi_0\rangle$ and $|\psi_X\rangle$ denote the initial and final quantum states of the circuit, respectively, prior to and following the data encoding. The operator $E\,(X)$ represents the encoding operation, and its specific form depends on the encoding schemes employed, such as basis encoding [34], amplitude encoding [35], and angle encoding [36], which correspond to different implementations.

Notably, a novel encoding method known as learnable rotation angle encoding (LRAE) was reported in [32], where classical information can be encoded into the quantum state of the qubit by the rotation gate. Usually, adopting the LRAE scheme allows for a significant reduction in the circuit depth, thereby lowering the computational complexity. Therefore, it was also used in this study. For a quantum circuit with M-input channels (qubits) with initial state $|\psi_0\rangle$, the encoded quantum state $|\psi_X\rangle$ using the LRAE with the Ry-gate can be expressed as:

$$|\psi_x\rangle = \otimes_{m=1}^{M} R_Y\ (x_m \cdot \theta_m)\ |\psi_0\rangle\,, \tag{4}$$

where $R_Y(x_m \cdot \theta_m)$ represents the encoding map of a Ry-gate acting on the m-th input qubit to encode the normalized data x_m in the classical dataset Xinto quantum state $|\psi_X\rangle$and θ_m is the trainable rotation angle parameter in the m-th quantum channel. These components are integrated into the encoding quantum circuit, allowing the effective transformation of classical information into quantum states.

2) ENTANGLEMENT EVOLUTION

After the encoding process, all the qubits across different wires in a quantum circuit remain independent and uncorrelated. However, upon the implementation of the quantum gate circuit, that is, the unitary operations are performed on different wires, the encoded quantum bits become entangled, enabling information sharing among them. The circuit that facilitates the entanglement of different channels is referred to as ansatz [37]. As highlighted in [38], four connection topologies, including star, chain, ring, and full connection topologies, can be employed to construct the ansatz. Each topology offers distinct advantages and affects the efficiency and performance of the entanglement process.

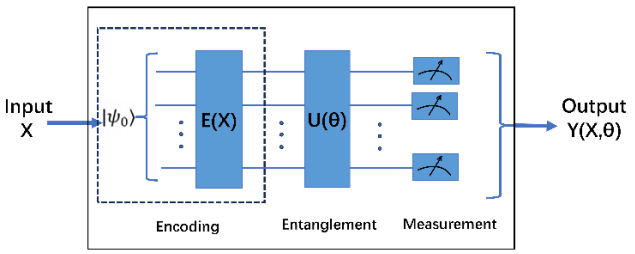

FIGURE 1. Diagram of general circuit structure in quantum computing. It usually includes three key components: the encoding layer, the entanglement evolution layer, and the measurement layer.

To denote $|\psi_x\rangle$ as the quantum state of the encoded qubits, the quantum state after executing the ansatz circuit can be expressed as

$$|\psi_x\rangle \rightarrow |\psi_y(x,\theta)\rangle = U(\theta)\,|\psi_x\rangle, \tag{5}$$

where $U(\theta)$ represents a sequence of unitary operations performed by quantum gates of a specific form

$$U(\theta) = \prod_{m=1}^{N} U_m(\theta_m), \tag{6}$$

where $U_m(\theta_m)$, $m = 1, 2, 3, \ldots, N$, represents the unitary operation of the m-th quantum gate (which may be either a single-qubit rotation gate or a two-qubit controlled gate) in Ansatz, and $\theta_m$represents the trainable parameter corresponding to the rotation angle used in the encoding process. To emphasize that two-qubit controlled gates, such as the *CNOT*-gates and *CR*-gates, are vital for generating quantum entanglement in the Ansatz circuit. In addition, the interaction between these gates and their associated parameters determines the quantum dynamics and computational capabilities of quantum circuits.

3) QUANTUM MEASUREMENT

Following the entanglement evolution, in which the encoded information becomes delocalized throughout the quantum circuit, a quantum measurement is performed on the local observable $O^{\otimes m}$ (e.g., Pauli operators $\sigma_i^{\otimes m}$, $i = \{x, y, z\}$with $m \leq q$, and q denotes the total quantum channels (or qubits) of the quantum circuit) within the entangled state $|\psi_y(x,\theta)\rangle$ generated by the entanglement circuit. The expected value of the observable $O^{\otimes m}$ after repeated measurements is given by

$$Y(x,\theta) = \langle |\psi_y(x,\theta)\rangle \,|\, O^{\otimes m} \,|\, |\psi_y(x,\theta)\rangle \rangle = \frac{1}{m}\sum_{k=1}^{m} O_k, \tag{7}$$

where m represents the number of measurement repetitions and O_k denotes the outcome of the k-th measurement. The quantum measurement process corresponds to the decoding process in which classical information is retrieved in the form of the output vector $Y(x,\theta)$ (i.e., the expected value of an observable O). This classical information can subsequently be processed using a classical computer for further analysis and application. This integration of quantum and classical information underscores the essential role of measurements in quantum computations and information processing.

III. METHODS

A. CONVOLUTION OPERATION

The convolutional layer in CNNs performs a critical operation known as ''convolution,'' which is essential for feature extraction from input images and the subsequent generation of feature maps. This operation involves a linear computation characterized by a weighted summation of the input data. During the convolution operation, a feature extractor, referred to as a convolution kernel or filter, was applied to the localized regions of the input image. Mathematically, this filter is represented as a two-dimensional (2D) array of learnable parameters that are adjusted during the training process to optimize feature extraction. The projection region of the filter on the input image is also known as the receptive field, which can be described as a 2D array (or vectors) with pixel elements in the receptive field. The convolution operation corresponds to the dot product of the two 2D vectors of the filter and the receptive field to generate an output feature map. To denote$H = \{h_{m,n}\}$ and $X = \{x_{i,j}\}$ as the vectors of the filter and receptive field with size $M \times N$, respectively, the output map labeled as $Y = \{y_{i,j}\}$ can be expressed as

$$y_{i,j} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} h_{m,n}x_{i+m,j+n}, \tag{8}$$

where$y_{i,j}$ represents the value of the output feature map at position(i, j), $x_{i+m,j+n}$, and $h_{m,n}$ individually represent the pixel value of the input image at position $(i + m, j + n)$ and the corresponding weight $h_{m,n}$ assigned by the filter. By repeatedly implementing the above convolution operation as the filter shifts with a certain stride (or step size) to scan the entire input image, the network can capture various features, such as edges, textures, and more complex patterns.

We denote l_i and w_i $(i = in, out)$ as the length and width of the input image $(i = in)$ and the output feature map $(i = out)$, respectively, and the length and width of the output feature map under a stride $s$of the filter can be expressed as

$$l_{out} = \left\lfloor \frac{l_{in} - M + 2P}{s} + 1 \right\rfloor, \tag{9}$$

$$w_{out} = \left\lfloor \frac{w_{in} - N + 2P}{s} + 1 \right\rfloor, \tag{10}$$

where p denotes the padding hyperparameter. According to (9) and (10) for $p = 0$ (no zero-padding to the input image), the size of the feature map generated through the convolution operation is always smaller than that of the input image, that is, $l_{out} < l_{in}$ and $w_{out} < w_{in}$ with $p = 0$.

B. TRANSPOSED CONVOLUTION NERUAL NETWORK

The transposed convolution introduced by Zeiler et al. [39] serves as an inverse-like operation to standard convolution, which can increase the spatial size of feature maps to recover low-resolution prediction maps. The primary distinction between the standard convolution and transposed convolution is that the backward propagation update protocol has been adopted in the former and backward propagation for the latter. Two approaches exist for implementing transposed convolution [40]. First, the transposed convolution operation is realized by the transposed convolution kernel (directly transposing the weight matrix of the convolution kernel). Second, the transposed convolution kernel was simulated using the standard convolution operation. However, for the latter, the input feature map must be preprocessed before the convolution operation begins. Specifically, we first insert

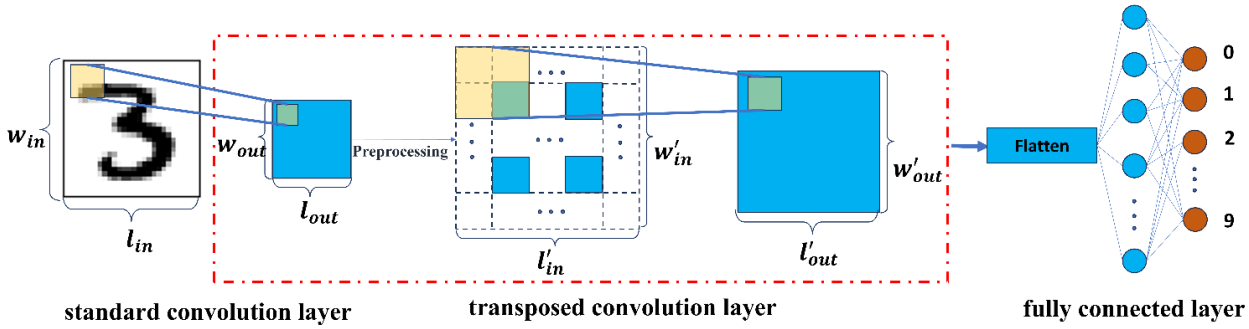

FIGURE 2. Architecture of a transposed convolutional neural network (CNN) with simulation scheme of transposed convolution. It is composed of a standard convolutional layer followed a transposed convolutional layer, and a fully connected layer.

$s-1$ rows and $s-1$ line zeros into the input image and then add $N-P-1$ rounds of zero values around the boundary of the image (p, N, and s are the parameters in (9) and (10)). The post-processed input image ensures that the final feature map generated across the standard convolution operation matches the input image.

In FigureFIGURE **2**, the general architecture of a transposed CNN adopting a simulation scheme is presented. This architecture comprises three main components: a standard convolutional layer, a transposed convolutional layer, and a fully connected layer. During the image processing, the CNN first extracts features from the input image with no preprocessing, such as ''digit 3,'' via the standard convolutional layer, obtaining a smaller feature map than the input image. Then, to preprocess the small feature map using the zero-padding scheme in the simulation transposed strategy and input the post-processing image into the following quantum convolutional circuit, the transposed convolutional layer can produce a final feature map that closely matches the dimensions of the original image. In addition, a classical fully connected layer was set at the end of the network to output the predicted results.

C. QUANTUM PSEUDO-TRANSPOSED CONVOLUTION NERUAL NETWORK

In this section, in the spirit of classical transposed CNN, as illustrated in Figure FIGURE 2, we propose a quantum variant known as QPTCNN. As depicted in FigureFIGURE **3**, a uniform scaling operation in our model acting on the input image was used to replace the standard convolution layer in Figure FIGURE 2. In addition, we constructed quantum convolutional circuits to substitute for the classical convolutional network. In our model, the last classical fully connected layer remained unchanged. The specific features of each module in the QPTCNN and the image-processing process are illustrated below.

For clarity, one takes an example of the input image ''digital 3'' with length l_{in} and width w_{in} to display the execution of QPTCNN. As shown in Figure FIGURE3, the scaling function was first utilized:

$$S(x) = resizeS(x, kl_{in}, kw_{in}), \tag{11}$$

with the scaling factor k ($k > 0$) to generate a resized picture with length $l_{out} = kl_{in}$ and width $w_{out} = kw_{in}$, which are the same as those given in (9) and (10), respectively, produced by the standard convolutional layer in Figure FIGURE2. Here, the resize function has been integrated into libraries such as OpenCV and PIL and can be invoked directly. To preprocess the resized image via the zero-padding technology as in the classical transposed CNN, one can obtain the image of size $l'_{in} \times w'_{in}$ with $l'_{in} = 2(M-p-1) + (l_{out}-1)(s-1)$ and $w'_{in} = 2(N-p-1) + (w_{out}-1)(s-1)$, which serves as the input of the quantum convolutional circuit (QCC). Here, each classical datum (i.e., the pixel value of the image) in the receptive field is encoded into a qubit (or quantum channel) via an H-gate followed by an Ry-gate performing the LRAE scheme. Thus, to save quantum resources (the number of qubits), a filter of size 2×2 is usually adopted. Subsequently, the quantum entanglement circuit denoted as $U(\theta)$ is constructed using single or two-qubit rotation gates. In our subsequent experiments, the ring configuration with the *CNOT*-gates (or *CRY*-gates) was adopted (see Figure FIGURE 4).

The *CNOT*-gates and *CRY*-gates connections between neighboring quantum channels of the quantum circuit can enable the generation of the entire entanglement state, meaning that the input classical data (information) encoded by each quantum channel is delocalized into the entire QCC (i.e., completing the information share). That is, unitary evolution $U(\theta)$ produces a global quantum entanglement state. Then, the quantum measurement onto the observable Pauli operators σ_i and $i = \{x, y, z\}$ is implemented (σ_z is selected in our experiments in subsections) to decode the information encoded in the entanglement state into classical information (extract the feature information of the input image from the entanglement state) to form the final feature map of size $l'_{out} \times w'_{out}$. Notably, the feature map matches the input image, that is, $l'_{out} = l_{in}$ and $w'_{out} = w_{in}$. At the end of the QPTCNN, a classical fully connected layer, followed by a softmax activation function, is employed to finalize the classification of the input image. For brevity, the QPTCNN incorporating the *CNOT*-gates entanglement circuit shown in Figure FIGURE4 (a) is designated as Model A, and that utilizing the *CRY*-gates entanglement circuit shown in

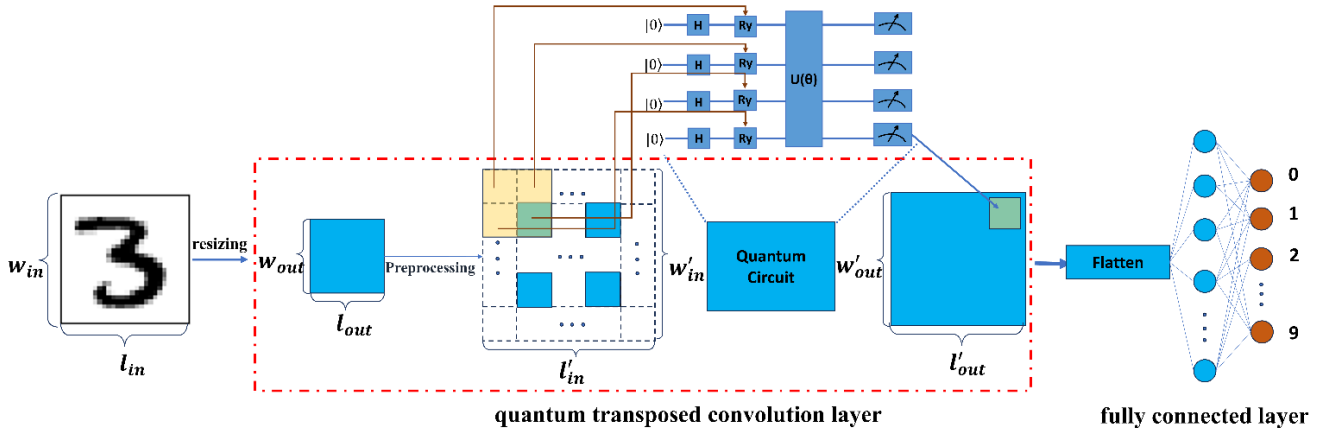

FIGURE 3. Diagram of QPTCNN implementing with an input image "digital 3" as an example. It involves the resizing operation with scaling function on the input image instead of the standard convolutional process in classical transposed CNN shown in Figure FIGURE 2, and a quantum transposed convolutional layer and a classical fully connected layer are followed to accomplish image recognition.

Figure FIGURE4 (b) is referred to as Model B. Here, it is noted that except for the structural differences of quantum entanglement circuits in Models A and B, another primary difference is the number of trainable parameters in the two models. For the square filter of size $N \times N$, there exist N^2 trainable rotation angle parameters from the *RY*-gates acting on the N^2 input qubits in the encoding layer, and no parameter for the entanglement layer with *CNOT*-gates connection in Model A. However, for Model B, it is twice that in Model A, owing to N^2 *CRY*-gates being involved in Model B.

IV. EXPERIMENTS

In our experiments, we mainly focus on the performance of QPTCNNs with different quantum circuit layers, that is, Models A and B. Specifically, based on the MNIST and Fashion-MNIST datasets, we estimate the test accuracy and loss value of our models with different stride s, $s = 1, 2, 3$, fixed filter size $M = N = 2$, and parameter $p = 0$. Before achieving the feature map through the QPTCNNs, the input image undergoes the following processes. First, in terms of (9) and (10), each input image with $l_{in=}w_{in} = 28$ in the MNIST and Fashion-MNIST datasets is scaled to $l_{out}(s) = w_{out} = \left\lfloor {}^{26}/_{s} + 1 \right\rfloor$ with $s = \{1, 2, 3\}$ by the resize function, that is, setting the scaling factor k in (12) as $k_s = {}^{l_{out}(s)}/_{l_{in}}$. Then, the resized image of size $l_{out} \times w_{out}$ via the zero-padding technology is transformed into the same size, $l'_{in}(s) = w'_{in}(s) = 29$ with $s = \{1, 2, 3\}$. Following the quantum pseudo-transposed convolutional layer with 2×2 filters, a feature map of the same size as that of the input image, that is, $l_{out} = w_{out} = 28$, can be generated.

A. DATASETS

The MNIST [41] and Fashion-MNIST [42] datasets are two datasets, that are widely applied in the field of computer vision. The MNIST dataset consists of images of handwritten digits, and the Fashion-MNIST dataset is composed of images of fashion items across ten categories, such as shirts, pants, and shoes. Each image in both datasets was a grayscale image with the dimensions of 28×28 pixels. Both MNIST and Fashion-MNIST contain 60,000 training samples and 10,000 testing samples. In our experiments, we randomly selected 1,200 samples for the training set and 200 samples from the same set for the testing set. Such an approach to sample selection allows for quicker iterations during the training process, facilitating rapid experimentation and tuning of the model parameters. Random sampling from the dataset helps mitigate potential biases that may arise from using specific subsets, ensuring a more generalized evaluation of the model's performance. By this method, we can maintain a balance between the computational efficiency and robustness of our performance assessment.

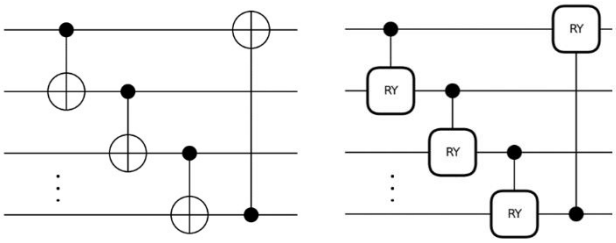

FIGURE 4. The ring-structure circuits interconnected with the CNOT-gates in (a) and the CRY-gates in (b).

B. EXPERIMENTAL ENVIRONMENT

We conducted the experiments on a local computer equipped with an Intel Core i5-7300HQ processor and NVIDIA GeForce RTX 1050Ti with 4GB of dedicated GPU memory. The quantum convolutional circuit in QPTCNN was constructed using TensorCircuit [43], an open-source, high-performance quantum computing software developed by Tencent Quantum Lab. TensorCircuit is an advanced tensor

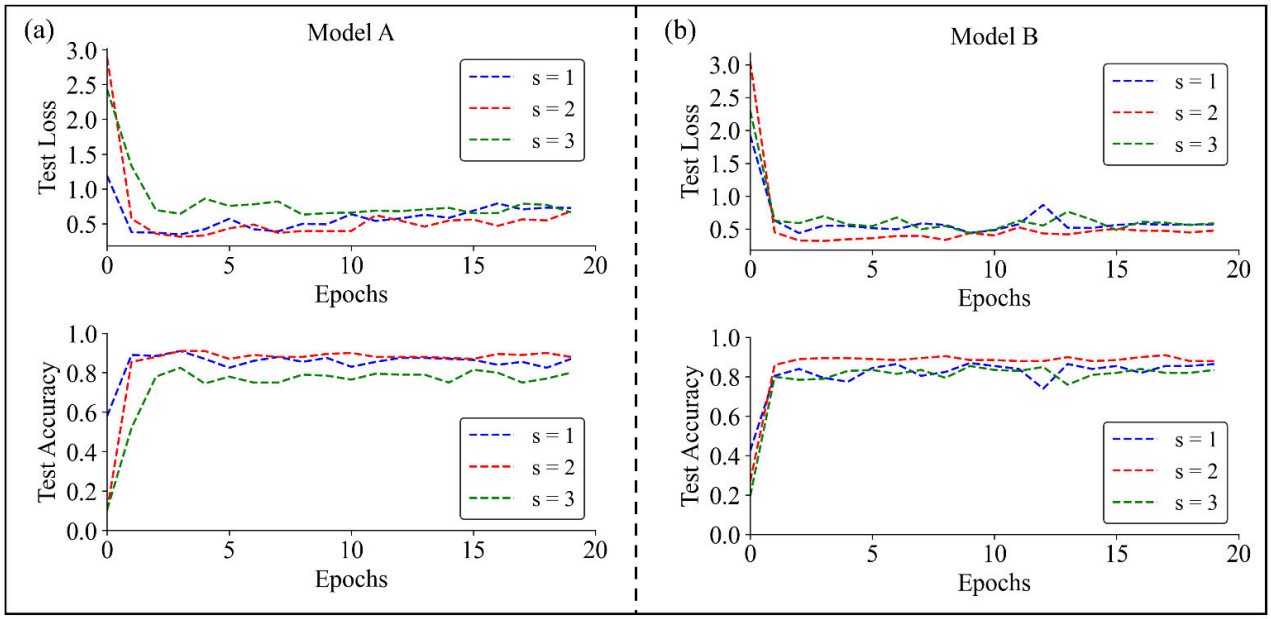

FIGURE 5. Variations of test accuracy and loss value as epoch times on the MNIST dataset for Model A in (a) and Model B in (b) when the different stride s, $s = 1, 2, 3$, are taken.

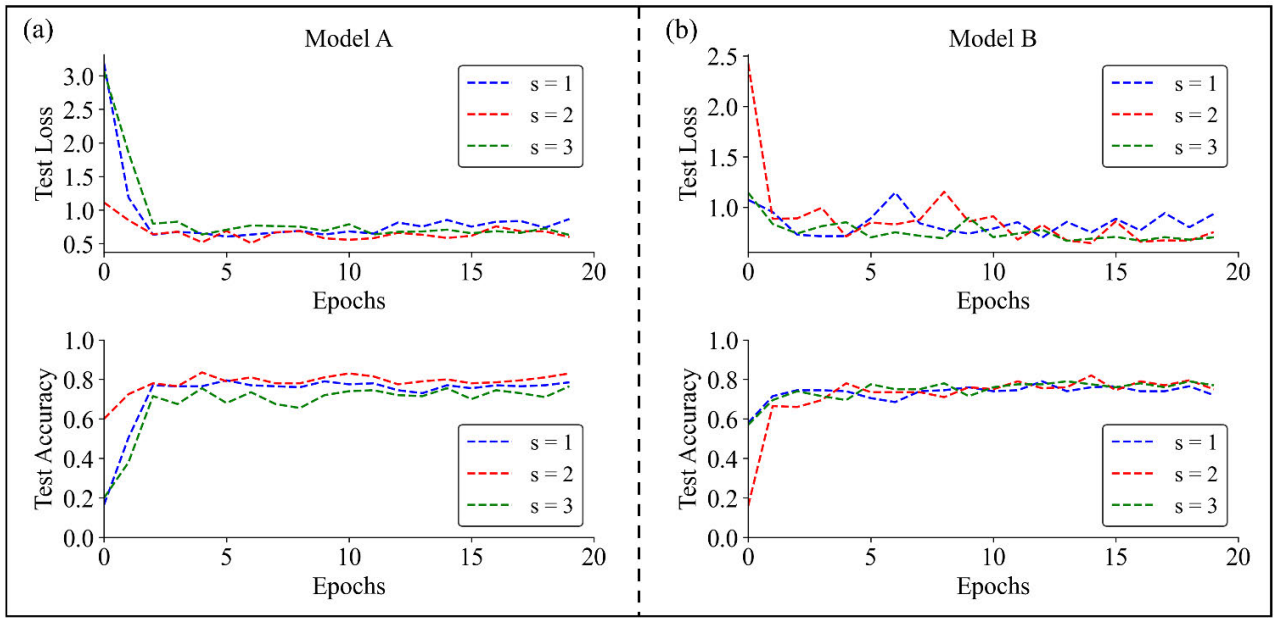

FIGURE 6. Variations of test accuracy and loss value as epoch times on the Fashion-MNIST dataset for Model A in (a) and Model B in (b) when the different stride s, $s = 1, 2, 3$, are taken.

network model [44] built on industry-standard machine learning frameworks such as PyTorch [45], TensorFlow [46], and JAX [47] It seamlessly integrates deep learning interfaces and engineering paradigms, enabling features such as automatic differentiation, just-in-time compilation, vector parallelization, and GPU acceleration. This makes TensorCircuit particularly suitable for building and optimizing quantum circuits, as well as executing quantum algorithms. Furthermore, the gradient updating process in QPTCNN is managed using PyTorch.

V. RESULT

Based on models A and B, we trained and tested them on the MNIST and Fashion-MNIST datasets, respectively. Figure FIGURE5 shows the variations in the loss and accuracy of models A and B as epochs on the MNIST dataset when different strides ($s = 1, 2, 3$) are selected, and Figure FIGURE6 shows the Fashion-MNIST dataset. As shown in Figure FIGURE5, Models A and B exhibit rapid convergence to a relatively high classification accuracy and low loss value on the MNIST dataset. Subsequently,

TABLE 1. Results of Model A and Model B on the MNIST and Fashion-MNIST datasets with different strides *s*.

Model	Dataset	s	Test acc	Test loss	Epoch Time [s]
Model A	MNIST	1	0.9100	0.3440	≈790
		2	**0.9100**	**0.3119**	≈**725**
		3	0.8250	0.6418	≈782
	Fashion-MNIST	1	0.7950	0.6054	≈734
		2	**0.8350**	**0.5197**	≈**834**
		3	0.7650	0.6294	≈907
Model B	MNIST	1	0.8700	0.4413	≈711
		2	**0.9100**	**0.4728**	≈**775**
		3	0.8550	0.4382	≈804
	Fashion-MNIST	1	0.7900	0.7007	≈759
		2	**0.8200**	**0.6447**	≈**726**
		3	0.7900	0.6672	≈837

TABLE 2. Comparison of optimal results for different QCNNs on the MNIST and Fashion-MNIST datasets.

Dataset	Model	Test acc	Test loss	Epoch time [s]
MNIST	Liu et al. [43]	0.8650	0.5341	1386
	Chen [31]	0.8950	0.3646	1095
	Ovalle-Magallanes et al. [32]	0.9000	0.3441	4256
	Proposed	**0.9100**	**0.3119**	**725**
Fashion-MNIST	Liu et al. [43]	0.7950	0.8923	1392
	Chen [31]	0.8050	0.8837	1191
	Ovalle-Magallanes et al. [32]	0.7850	0.7306	1191
	Proposed	**0.8350**	**0.5197**	**726**

the performance remained stable, with minor fluctuations in subsequent epochs. A similar pattern is observed in Figure FIGURE6 for the Fashion-MNIST dataset. These results demonstrate that our QPTCNN model possesses strong learning capability and relatively stable output. Additionally, Figure FIGURE5 and Figure FIGURE**6** show that the performance of the model (in terms of classification accuracy and loss value) is influenced by the stride parameter s. when $s = 2$ was used, both Models A and B achieved their best performance.

In Table 1, we compare the optimal performance of the two models on the MNIST and Fashion-MNIST datasets across different stride s. It was found that both Models A and B had the highest accuracy and lowest loss for stride $s = 2$, and they have the same optimal accuracy (0.9100) on the MNIST dataset, but a lower loss value (0.335) appears in Model A than in Model B with 0.4728. On the Fashion-MNIST dataset, although Model A required a slightly longer epoch time than Model B, it consistently outperformed Model B in terms of both accuracy and loss. Thus, QPTCNN with fewer parameters (Model A with four parameters) outperformed the situation with more parameters (Model B with eight parameters) in the classification tasks. In addition, comparisons of the optimal performance of our QPTCNNs and other QCNNs on the MNIST and Fashion-MNIST datasets are presented in Table 2. It can be observed that the optimal QPTCNN (i.e., Model A stride $s = 2$) also outperforms the existing QCNN models [31], [32], and [43] with a similar hybrid quantum-classical CNN framework. Specifically, on the MNIST dataset, the optimal QPTCNN achieved an accuracy improvement of 1.11% and a reduction of 9.29% in loss compared to the best model previously proposed in [32]. Similarly, on the Fashion-MNIST dataset, our model achieved an accuracy of 0.8350 and a loss of 0.5197, surpassing the best model from [31] by 3.73% accuracy and reducing the loss by 41.29%. The above results suggest that our model with the quantum pseudo-transposed convolution framework has strong learning capability and can capture richer feature information.

VI. CONCLUSION

Inspired by the operational characteristics of classical transposed convolutional neural networks (CNNs), we have developed a quantum variant, i.e., the QPTCNN model. The QPTCNN employs a hybrid quantum-classical framework, combining a quantum convolutional layer with a classical fully connected layer. By incorporating image preprocessing steps, the quantum circuit layer in the QPTCNN mimics a transposed convolution operation, ensuring that the output feature map retains the same size as the input image. In the QPTCNN model, we utilized single-qubit rotation gates and Hadamard gates to construct an encoding layer based on the rotational angle encoding protocol. The quantum convolutional layer is designed using a ring-structure quantum circuit interconnected by two-qubit control gates, such as *CNOT* and *CRY* gates. This structure allows the model to perform the quantum convolution operation effectively. To optimize quantum resource utilization, a 2×2 filter was applied in our experiments, reducing the number of qubits required. In terms of the MNIST and Fashion-MNIST datasets, we evaluated the performance of two versions of the QPTCNN: Model A utilizing a *CNOT*-gate entanglement circuit and Model B with the *CRY*-gates entanglement circuit. The results demonstrate that the QPTCNN is a high-performance CNN model with significant learning power and an enhanced ability to capture important feature information. Notably, Model A can exhibit superior classification accuracy and lower loss across both datasets compared to similar existing models.

This study makes a step forward in improving image classification accuracy, highlighting the potential for quantum computing to enhance high-precision image recognition tasks. It might also offer significant insights for optimizing other neural network architectures, advancing the integration of quantum computing with deep learning.

APPENDIX A

A. MATH REPRESENTATION AND OPERATION OF QUANTUM GATES

The general quantum gates applied in a single qubit include the Hadamard gate (H-gate), rotation-gates, and phase gate (P -gate). The matrix representations are given as

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}(H-\text{gate}), \tag{A1}$$

$$R_x(\theta) = \begin{bmatrix} & -isin\left(\frac{\theta}{2}\right) \\ -isin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}(RX-\text{gate}), \tag{A2}$$

$$R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -sin\left(\frac{\theta}{2}\right) \\ sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}(RY-\text{gate}), \tag{A3}$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}(RZ-\text{gate}), \tag{A4}$$

$$P(\phi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}(P-\text{gate}), \tag{A5}$$

where $R_i(\theta)$, $(i = x, y, z)$ represents the rotation operation around the i-axis by an angle of θ in the Bloch representation of the qubit. Here, the mapping operations of the H-gate and P-gates can be written as

$$|0\rangle \to H\ |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \tag{A6}$$

$$|1\rangle \to H\ |1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle), \tag{A7}$$

$$|\psi\rangle \to P(\phi)\ |\psi\rangle = \alpha\ |0\rangle + \beta\ e^{i\phi}|1\rangle\Big\rangle, \tag{A8}$$

where $|\psi\rangle = \alpha\ |0\rangle + \beta\ e^{i\phi}|1\rangle$ is a general supposition state of qubit based $|0\rangle$ and $|1\rangle$. For the two qubits, the $CNOT$-gate and the control rotational gate (CR-gate) are usually applied and can be expressed as

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}(CNOT-\text{gate}), \tag{A9}$$

$$CR_i(\theta) = \begin{bmatrix} I_{2\times 2} & 0 \\ 0 & R_i(\theta) \end{bmatrix}, i = x, y, z\,(CR_i-\text{gate}). \tag{A10}$$

Let $|\varphi\rangle = \alpha\ |10\rangle + \beta\ |01\rangle$, then

$$CNOT\ |\varphi\rangle = \alpha\ |10\rangle + \beta\ |00\rangle, \tag{A11}$$

$$CR_i(\theta)\ |\varphi\rangle = \alpha\ |10\rangle + \beta\ |0\rangle R_i(\theta)\ |1\rangle, \tag{A12}$$

(A11) and (A12) show that the $CNOT$-gates and $CR_i(\theta)$-gate $(i = x, y, z)$ independently perform the flipping operation $|0\rangle \leftrightarrow |1\rangle$and the rotational operation $|k\rangle \leftrightarrow R_i(\theta)|k\rangle$ $(k = 0, 1)$ of the target qubit (second qubit) only for the control qubit (first qubit) at the state $|1\rangle$.

It is noted that the operations of quantum gates are unitary satisfying $UU^\dagger = I$, with I being the identity matrix and $U^\dagger$ being the transpose conjugate operator of U.

REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, vol. 31, no. 1, pp. 4278–4284.
[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf.*, vol. 18, Munich, Germany. Cham, Switzerland: Springer, Oct. 2015, pp. 234–241.
[7] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 833–851.
[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[9] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast R-CNN for pedestrian detection," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 985–996, Apr. 2018.
[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[11] F. Chen, X. Wang, Y. Zhao, S. Lv, and X. Niu, "Visual object tracking: A survey," *Comput. Vis. Image Understand.*, vol. 222, Jul. 2022, Art. no. 103508.
[12] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1451–1460.
[13] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1520–1528.
[14] G. E. Moore, "Lithography and the future of Moore's law," *Proc. SPIE*, vol. 2439, no. 3, pp. 2–17, May 1995.
[15] S. S. Gill et al., "AI for next generation computing: Emerging trends and future directions," *Internet Things*, vol. 19, Aug. 2022, Art. no. 100514.
[16] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Rev. Modern Phys.*, vol. 81, no. 2, pp. 865–942, Jun. 2009.
[17] J. Grabowski, M. Kuś, and G. Marmo, "Geometry of quantum systems: Density states and entanglement," *J. Phys. A, Math. Gen.*, vol. 38, no. 47, pp. 10217–10244, Nov. 2005.
[18] J. Bardin, "Beyond-classical computing using superconducting quantum processors," in *Proc. IEEE Int. Solid- State Circuits Conf. (ISSCC)*, vol. 65, Feb. 2022, pp. 422–424.
[19] Y.-H. Deng et al., "Gaussian boson sampling with pseudo-photon-number-resolving detectors and quantum computational advantage," *Phys. Rev. Lett.*, vol. 131, no. 15, Oct. 2023, Art. no. 150601.
[20] O. Simeone, "An introduction to quantum machine learning for engineers," *Found. Trends Signal Process.*, vol. 16, nos. 1–2, pp. 1–223, 2022.
[21] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemp. Phys.*, vol. 56, no. 2, pp. 172–185, 2015.
[22] Y. Zhang and Q. Ni, "Recent advances in quantum machine learning," *Quantum Eng.*, vol. 2, no. 1, p. 34, Mar. 2020.
[23] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature Comput. Sci.*, vol. 2, no. 9, pp. 567–576, Sep. 2022.

[24] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural net-works," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, 2019.
[25] S. Oh, J. Choi, and J. Kim, "A tutorial on quantum convolutional neural networks (QCNN)," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 236–239.
[26] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," *Phys. Rev. Res.*, vol. 4, no. 1, Mar. 2022, Art. no. 013231.
[27] A. Senokosov, A. Sedykh, A. Sagingalieva, B. Kyriacou, and A. Melnikov, "Quantum machine learning for image classification," *Mach. Learn., Sci. Technol.*, vol. 5, no. 1, Mar. 2024, Art. no. 015040.
[28] Z. Hong, J. Wang, X. Qu, X. Zhu, J. Liu, and J. Xiao, "Quantum convolutional neural network on protein distance prediction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
[29] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, "Quanvolutional neural networks: Powering image recognition with quantum circuits," *Quantum Mach. Intell.*, vol. 2, no. 1, p. 2, Jun. 2020.
[30] M. Monnet, H. Gebran, A. Matic-Flierl, F. Kiwit, B. Schachtner, A. Bentellis, and J. M. Lorenz, "Pooling techniques in hybrid quantum-classical convolutional neural networks," in *Proc. IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, Sep. 2023, pp. 601–610.
[31] Y. Chen, "Quantum dilated convolutional neural networks," *IEEE Access*, vol. 10, pp. 20240–20246, 2022.
[32] E. Ovalle-Magallanes, D. E. Alvarado-Carrillo, J. G. Avina-Cervantes, I. Cruz-Aceves, and J. Ruiz-Pinales, "Quantum angle encoding with learnable rotation applied to quantum–classical convolutional neural networks," *Appl. Soft Comput.*, vol. 141, Jul. 2023, Art. no. 110307.
[33] M. Zeiler, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, vol. 1311, 2014, pp. 818–833.
[34] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Data encoding patterns for quantum computing," in *Proc. 27th Conf. Pattern Lang. Programs*, 2020, pp. 1–11.
[35] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Phys. Rev. A, Gen. Phys.*, vol. 102, no. 3, Sep. 2020, Art. no. 032420.
[36] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4806–4814.
[37] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, Feb. 2019.
[38] M. Benedetti, "Quantum-classical generative models for machine learning," Ph.D. thesis, Dept. Comput. Sci., Univ. College London, 2019.
[39] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
[40] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016, *arXiv:1603.07285*.
[41] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[42] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[43] S.-X. Zhang, J. Allcock, Z.-Q. Wan, S. Liu, J. Sun, H. Yu, X.-H. Yang, J. Qiu, Z. Ye, Y.-Q. Chen, C.-K. Lee, Y.-C. Zheng, S.-K. Jian, H. Yao, C.-Y. Hsieh, and S. Zhang, "TensorCircuit: A quantum software framework for the NISQ era," *Quantum*, vol. 7, p. 912, Feb. 2023.
[44] M. Niedermeier, J. L. Lado, and C. Flindt, "Simulating the quantum Fourier transform, Grover's algorithm, and the quantum counting algorithm with limited entanglement using tensor-networks," 2023, *arXiv:2304.01751*.
[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.
[46] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[47] S. Duffield, G. Matos, and M. Johannsen, "Qujax: Simulating quantum circuits with JAX," *J. Open Source Softw.*, vol. 8, no. 89, p. 5504, Sep. 2023.

LI HAI received the bachelor's degree in physics education from Shanxi Normal University, Linfen, China, in 2003, the master's degree in curriculum and pedagogy from Xihua Normal University, Nanchong, China, in 2006, and the Ph.D. degree in condensed matter physics from Beijing Institute of Technology, Beijing, China, in 2014.

From 2016 to 2018, he conducted Postdoctoral Research with the Institute of Theoretical Physics, Chinese Academy of Sciences, China. He has been an Associate Professor with Shandong Technology and Business University, since 2016. His research interests include the fundamental study of quantum information and quantum computing, machine learning, energy storage and conversion in micro/nano quantum systems, and quantum statistics and thermodynamics.

Dr. Li is a Committee Member of the Quantum Science and Technology Professional Committee, Shandong Institute of Physics, China.

CHEN LIANG received the B.S. degree in electronic information from Qiqihar University, Qiqihar, China, in 2022. He is currently pursuing the M.S. degree in information and electronic engineering with Shandong Technology and Business University, Yantai, China, with research focusing on quantum computing and artificial intelligence.

HAO YAMING received the B.S. degree in physics from Jining University, Qufu, China, in 2013, and the Ph.D. degree in materials science and engineering from the East China University of Science and Technology, Shanghai, China, in 2019.

Since 2019, she has been a Lecturer with the School of Information and Electronic Engineering, Shandong Technology and Business University, Yantai, China. Her research interests include utilizing a composite system of Rydberg atoms and cavity QED for quantum information processing, quantum batteries, and quantum neural networks.

YU WENLI received the B.S. degree in economic information management from Shandong University of Economics, Jinan, China, in 1998, and the M.S. degree in computer applications technology from Xihua Normal University, Nanchong, China, in 2006.

Since 2009, she has been a Lecturer with the School of Computer Science and Technology, Shandong Technology and Business University, Yantai, China. Her interests include machine learning, artificial intelligence, and algorithms.

SHI FENGQUAN received the bachelor's degree in electronic physics, and the master's and Ph.D. degrees in quantum radiation from Osaka Prefectural University, Japan, in 2014, 2016, and 2020, respectively. From 2014 to 2016, he participated in the follow-up research and development of a mid-infrared sensor using type-II InAs/GaSb superlattice on InP substrates at Sumitomo Electric Industries. From 2016 to 2020, he an assisted Professor Ryoichi Taniguchi in conducting research and development projects sponsored by the Japan Society for the Promotion of Science and Technology (JSPS) and the Science Research Funding Assistance Agency (KAKEN).