Quantum Machine Intelligence (2023) 5:41
https://doi.org/10.1007/s42484-023-00130-3

RESEARCH ARTICLE

Quantum convolutional neural networks for multi-channel supervised learning

Anthony M. Smaldone[1] · **Gregory W. Kyro**[1] · **Victor S. Batista**[1]

Received: 30 May 2023 / Accepted: 31 August 2023

Abstract
As the rapidly evolving field of machine learning continues to produce incredibly useful tools and models, the potential for quantum computing to provide speed up for machine learning algorithms is becoming increasingly desirable. In particular, quantum circuits in place of classical convolutional filters for image detection-based tasks are being investigated for the ability to exploit quantum advantage. However, these attempts, referred to as quantum convolutional neural networks (QCNNs), lack the ability to efficiently process data with multiple channels and, therefore, are limited to relatively simple inputs. In this work, we present a variety of hardware-adaptable quantum circuit ansatzes for use as convolutional kernels, and demonstrate that the quantum neural networks we report outperform existing QCNNs on classification tasks involving multi-channel data. We envision that the ability of these implementations to effectively learn inter-channel information will allow quantum machine learning methods to operate with more complex data.

Keywords Quantum machine learning · Convolutional neural networks · Multi-channel data · Image classification · Supervised learning

1 Introduction

Quantum computers are devices that utilize quantum mechanical phenomena such as superposition and entanglement to solve problems that are infeasible with timescales provided by classical computers. From the time quantum computers were first pro- posed in 1982 (Feynman 1982), researchers have been working to develop algorithms that provide quantum advantage over classical algorithms (Grover 1996; Shor 1994; Deutsch and Jozsa 1992; Cleve et al. 1998; Bernstein and Vazirani 1997; Simon 1997; Kitaev 1995). The main application challenges are related to noise, high gate errors and short decoherence times (Preskill 2018), necessitating the development of more efficient quantum circuit ansatzes. In recent years, devices have demonstrated quantum supremacy such as those introduced by Google AI Quantum (2019) (Arute et al. 2019), the University of Science and Technology of China (2020) (Zhong et al. 2020), and IBM (2021) (Chow et al. 2021).

Machine learning (ML) has been gaining significant attention in recent years for strongly influencing many important fields of study and sectors of society. Image recognition (Pak and Kim 2017; Wu and Chen 2015; Liu et al. 2020), natural language processing (Otter et al. 2020; Young et al. 2018; Li 2018), self-driving vehicles (Fujiyoshi et al. 2019; Gupta et al. 2021; Rao and Frtunikj 2018; Rao and Frtunikj 2017; Mozaffari et al. 2020), robotics (Pierson and Gashler 2017; Wang and Siau 2019; Kim et al. 2021; Lesort et al. 2020; Kleeberger et al. 2020), and molecular biochemistry (Choi et al. 2019; Bonetta and Valentino 2020; Ballester and Mitchell 2010; Sino et al. 2021) are just some of the fields that are being revolutionized by ML. The most prominent example would be ChatGPT, a chatbot built on top of OpenAI's generative pre-trained transformer (GPT)−3.5 and GPT-4 large language models that have been fine-tuned for conversation using both supervised and reinforcement learning from human feedback techniques (OpenAI 2023a, b). In the natural sciences, Deepmind's Alphafold is able to accurately predict 3D models of pro-

✉ Victor S. Batista
victor.batista@yale.edu

Anthony M. Smaldone
anthony.smaldone@yale.edu

Gregory W. Kyro
gregory.kyro@yale.edu

[1] Department of Chemistry, Yale University, 225 Prospect Street, New Haven 06511, CT, USA

Published online: 09 October 2023

tein structures, and is accelerating research in nearly every field of molecular biology and biochemistry (Jumper et al. 2021).

The advances in both quantum computing and ML inspire the development of quantum ML (QML) methods to exploit the speed of quantum computations and the predictive capabilities of ML. There has been recent work demonstrating the feasibility and advantages of substituting components of classical ML architectures with quantum analogs (Schuld et al. 2015; Biamonte et al. 2017; Zhang and Ni 2020; Cerezo et al. 2022), such as quantum circuits in place of classical convolutional kernels in convolutional neural networks (CNNs) (Cong et al. 2019; Henderson et al. 2019; Oh et al. 2020; Chen et al. 2022; Hong et al. 2021; Jing et al. 2022; Mishra and Tsai 2023; Mari 2021; Hur et al. 2022). Classical CNNs are state-of-the-art for image, video, and sound recognition tasks (Rawat and Wang 2017; LeCun et al. 2015) and also have applications in the natural sciences (Jumper et al. 2021; Gao et al. 2021; Wei and Chen 2019; Chen et al. 2018; Kyro et al. 2023; Casey et al. 2020; Senior et al. 2020). CNNs that incorporate quantum circuits to function as kernels, referred to as Quantum CNNs (QCNNs), have performed well on classification tasks involving simple data such as the MNIST dataset of handwritten digits (Hur et al. 2022; Oh et al. 2020), as well as multi-channel data such as the CIFAR-10 dataset (Jing et al. 2022; Riaz et al. 2023).

The QCNN was introduced in 2019 by Cong et al. (2019), and was applied to quantum phase recognition and quantum error correction optimization. Since then, QCNNs have been applied in areas ranging from high energy physics (Chen et al. 2022) to biochemistry (Hong et al. 2021). However, current methods either do not effectively capture inter-channel information, or require more qubits than are currently permissible, and therefore lack the ability to efficiently process more complex data with multiple channels.

With current QCNNs, the number of required qubits scales linearly with the length of the channel dimension of the input data. This is a feasible approach for simple data that can be modeled with small filters. For instance, Jing et al. demonstrated the ability to use a 12- and 18-qubit circuit to function as a 2×2 and 3×3 convolutional filter, respectively, on low resolution red-green-blue (RGB) images (three channels, one for each color) (Jing et al. 2022). However, extension of this approach to tasks involving data with more channels is prohibited by current hardware limitations. As an attempt to overcome this challenge, there has been much recent work that performs a measurement on each channel individually, collapsing the wavefunction after measuring a given channel of the data and storing the measurement classically (Chen et al. 2022; Hong et al. 2021; Oh et al. 2020; Mishra and Tsai 2023). Although the hardware requirements have no dependence on the number of channels when using this method, much of the inter-channel information is lost, which is valuable for accurately modeling the data. In this work, we propose several methods for operating with multi-channel data that preserve inter-channel information and require a number of qubits that is independent of the length of the channel dimension of the input data.

2 Quantum computers for convolutional neural networks

2.1 Basics of quantum computing

Qubits are two-level systems used in computations to exploit the quantum mechanical phenomena of superposition and entanglement. The state of a single qubit can be represented by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \text{such that} |\alpha|^2 + |\beta|^2 = 1 \text{ and } \alpha, \beta \in \mathbb{C}. \tag{1}$$

The state of an n-qubit system is represented by the superposition of all possible n-bit strings:

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle \tag{2}$$

These qubit states are transformed via unitary operations. For any single qubit unitary,

$$U = \begin{bmatrix} u_{00} & u_{01} \\ u_{10} & u_{11} \end{bmatrix}, \tag{3}$$

the matrix of a two-qubit controlled operation may be expressed as

$$CU = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{00} & u_{01} \\ 0 & 0 & u_{10} & u_{11} \end{bmatrix}. \tag{4}$$

In a two-qubit controlled gate, if the control qubit is in the $|1\rangle$ state, the unitary operation is applied to the target qubit. If the control qubit is in the $|0\rangle$ state, the target qubit is unaffected. Measurements on a quantum state are performed by taking the expectation value of a Pauli operator, σ as shown in Eq. 5. Quantum circuits model the sequential manipulation and measurement of qubits.

$$\mathcal{M} = |\psi\rangle\sigma_{x,y,z}|\psi\rangle \tag{5}$$

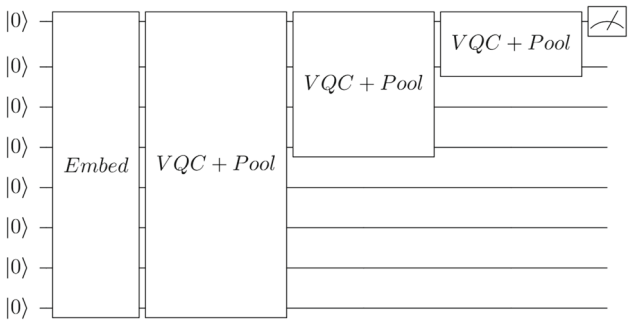


Fig. 1 A general circuit for a quantum convolutional neural network with 8 input qubits. The classical data is embedded, followed by variational quantum circuits and pooling layers, concluded with a measurement

2.2 The structure of quantum convolutional neural networks

The traditional structure of QCNNs consist of embedding classical data in a quantum mechanical state, having $O(\log(n))$ convolutional and pooling layers for n initial qubits, a measurement layer, followed by a classical fully-connected layer. The "convolutional" component in this quantum circuit is a variational quantum circuit (VQC). A VQC is a series of gates such that they have learnable parameters that are updated via backpropagation. The circuit architecture for a traditional QCNN is shown in Fig. 1. While not a hard requirement, this hierarchical structure of halving the number of qubits with each layer (analogous to classical pooling in CNNs) circumvents the "barren plateau" phenomenon and thus guarantees trainability. Oh et al. (2020) provides a simple QCNN tutorial for a 2×2 filter, utilizing a single VQC and pooling layer shown in Fig. 2. This structure was implemented (Menborong 2020) and is used as the comparative "control" model with respect to this work.

2.3 Multi-channel data with quantum convolutional neural networks

Previous works (Oh et al. 2020; Chen et al. 2022; Hong et al. 2021; Mishra and Tsai 2023) have performed VQCs on each channel, while storing this information classically between channel convolutions and summing the result. Collapsing the wavefunction with a measurement between channels breaks any entanglement between them, hindering the model's ability to learn inter-channel patterns effectively. To our knowledge, the only attempt at including all channel data into a quantum circuit for convolutions was performed by Jing et al. (2022). Jing takes an intuitive approach and increases the number of qubits proportionally to the number of pixels covered by the filter. Thus, a 2×2 filter acting on an image containing three color channels requires 12 qubits in this approach, as shown in Fig. 3.

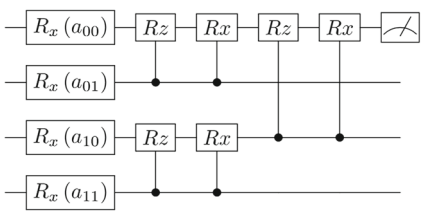


Fig. 2 The simple circuit for a quantum convolutional neural network demonstrated by Oh et al. (2020) which inspired the Control model (Menborong 2020)

3 Extending QCNNs for multi-channel data

3.1 The channel overwrite method

The proposed Channel Overwrite (CO) method takes advantage of a controlled phase gate applied to a single ancilla qubit to entangle information between channels. In our case, a single channel of classical data is encoded into a quantum state and passed through a learnable set of unitary operations. Quantum information from the working qubits is exchanged onto the ancilla qubit via a controlled phase gate, where the phase shift angle is a learnable parameter. The classical data for the next channel is angle encoded onto the working qubits, passed though the learnable set of unitaries, and has information further exchanged with the ancilla using a controlled phase gate. This process is repeated for each channel. After processing all channels of the data, the ancilla qubit is measured, and the expectation value is used as the output of that quantum convolution.

3.1.1 State preparation

The CO-QCNN method removes the dependency of the number of necessary qubits on the length of the channel dimension of the data, and requires only $F^2 + 1$ qubits, where F is the length of a single dimension of the square filter. The working qubits perform a convolution by executing a set number of learnable unitary gates over one channel of an input signal.

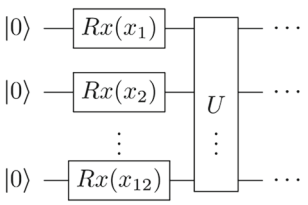


Fig. 3 The embedding and convolutional scheme for Jing et al. (2022) implementation of a 2×2 filter on a color image. The twelve pixels covered by the filter are flattened and angle encoded via a rotation about the Pauli X axis, and then passed to a variational unitary block U

The resulting state undergoes a phase shift controlled by the ancilla qubit. The uninitialized quantum state and classical input can be represented as follows:

$$|\Psi_0\rangle = |0\rangle \otimes |\psi_0\rangle = |0\rangle \otimes |0\rangle^{\otimes F^2}, \mathbf{x}_{l,w,c} \in \mathbb{R}^{N-1} \tag{6}$$

where l, w, c are the initial indices of an input 3D-tensor of total length L, width W, and channels C, respectively, $|\Psi_0\rangle$ is the state of the total quantum system, and $|\psi_0\rangle$ is the state of the working qubits. $\mathbf{x}_{l,w,c}$ is a vector of the flattened normalized input data covered by the filter for a single channel, described by the expression:

$$\mathbf{x}_{l,w,c} = \Big\{x_{l,w,c}, x_{l,w+1,c}, \cdots, x_{l,w',c}, x_{l+1,w,c}, x_{l+1,w+1,c}, \cdots, x_{l+1,w',c}, \cdots, x_{l',w',c}\Big\} \tag{7}$$

where l' and w' represent the final index covered by the filter of the length and width dimensions.

$$l' = l + F - 1, w' = w + F - 1 \tag{8}$$

In this work, the input data is encoded into a quantum state with angle encoding, as this method requires only a single gate per qubit and permits a low circuit depth compared to other encoding schemes. We encode all normalized data $\mathbf{x}$ with a rotation about the Pauli-X axis by πx_i radians. Additionally, we prepare the ancilla qubit by placing it into a state of maximal superposition with a Hadamard transformation. The matrix representation of the gates for a rotation about the Pauli-X axis and a Hadamard transformation are described in Eqs. 9 and 10, respectively.

$$\mathbf{R}_{\mathbf{x}}(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ -\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{9}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{10}$$

Letting $U_{RX,c}$ be the unitary block that embeds the classical data of channel c, the prepared quantum state is represented as follows:

$$\begin{aligned} |\Psi_1\rangle &= |+\rangle \otimes |\psi_1\rangle = \mathbf{H}|0\rangle \otimes \bigotimes_{i=1}^{F^2} \mathbf{R}_{\mathbf{x}}(x_i \cdot \pi) |\psi_0\rangle \\ &= \mathbf{H}|0\rangle \otimes U_{RX,1}|\psi_0\rangle \end{aligned} \tag{11}$$

3.1.2 Circuit construction

Let $\mathbf{U}$ be the set of γ chosen unitaries to perform the quantum convolution.

$$\mathbf{U} = \prod_{i=1}^{\gamma} U_i \tag{12}$$

After the classical data has been encoded and the deposition qubit has been placed into a state of maximal superposition, $\mathbf{U}$ is applied to the working qubits, transforming the total state according to:

$$|\Psi_2\rangle = |+\rangle \otimes \mathbf{U}|\psi_1\rangle = |+\rangle \otimes |\psi_2\rangle. \tag{13}$$

Following this learnable unitary block, we parametrically exchange phasic information from the entangled working register and the prepared ancilla qubit. This is achieved via a controlled phase gate, CP:

$$|\Psi_3\rangle = CP\left(|+\rangle \otimes |\psi_2\rangle\right), \tag{14}$$

where the matrix representation of the single qubit phase gate is as follows:

$$P(\theta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \tag{15}$$

It should be noted that this exchange of phasic information with the ancilla qubit is ideally applied to each qubit in the working register to retain as much information as possible. However, connecting all qubits to a single particular qubit can be challenging depending on the available quantum hardware. Unless otherwise stated, all results shown in this work restrict the controlled operations to the ancilla qubit and the first qubit of a given working register.

The working qubits are then overwritten with data from the next channel, and the circuit proceeds as before. This procedure of angle-encoding channel data and executing a controlled phase gate with the ancilla is repeated for each channel of the input data. Therefore, the final state $|\Psi_f\rangle$ of the circuit is described by Eq. 16, and output of this circuit is obtain via a measurement (Eq. 17) of the ancilla qubit in the Hadamard basis. A sample circuit is shown in Fig. 4.

$$|\Psi_f\rangle = \left(\prod_{c=2}^{C} CP \cdot \mathbf{U} \cdot U_{RX,c}\right) |\Psi_3\rangle \tag{16}$$

$$\mathcal{M} = \langle \mathbf{H}\Psi_f | \sigma_z | \mathbf{H}\Psi_f \rangle \tag{17}$$

3.2 Parallel channel overwrite method

The Parallel Channel Overwrite-QCNN (PCO-QCNN) method decreases the circuit depth compared to that which is used in the CO-QCNN method. The PCO-QCNN operates simultaneously on R parallelized channels, where R is the number of specified working registers. Information from these working registers is exchanged with the ancilla and are all overwritten with the next set of channel data as in the CO-QCNN method. This method therefore requires $RF^2 + 1$ qubits.

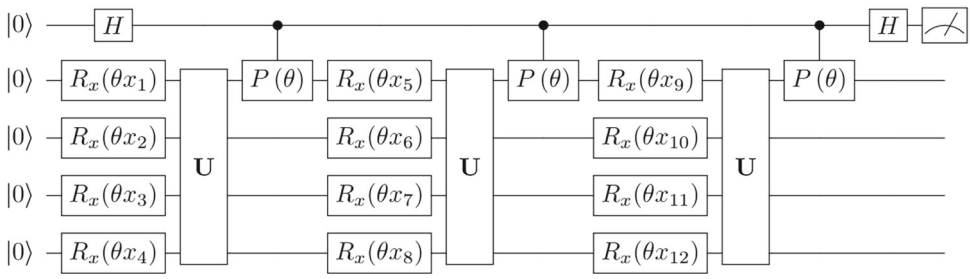


Fig. 4 Channel Overwrite (CO) circuit using a 2×2 filter applied to three channels

3.2.1 State preparation

The states of the working qubits in the PCO-QCNN method are prepared similarly to the CO-QCNN method, with the caveat of each register being encoded with a different channel of the data The overall quantum state is initialized as done in Eq. 18, where $|\psi_{0,r}\rangle$ is the initial substate of qubits in a given register. The data covered by the filter of R channels is angle encoded as done in Eq. 19, and the ancilla qubit is placed in a uniform superposition.

3.2.2 Circuit construction

The method of applying a controlled phase gate and overwriting is extended to R working qubit registers in the PCO-QCNN. First intra-channel data is entangled followed by the entanglement of inter-channel data. This hierarchical method of first entangling intra-channel data (Eq. 20) and then inter-channel data (Eq. 21) is similar to the method used in the HQConv circuit (Jing et al. 2022). Controlled phase gates are performed between the ancilla qubit and the first qubit of each working register in Eq. 22, and a measurement is taken as previously done in Eq. 17. The state evolution of the PCO-QCNN circuit is sequentially presented in Eqs. 18-23.

$$|\Psi_0\rangle = |0\rangle \otimes \bigotimes_{r=1}^{R} |\psi_{0,r}\rangle = |0\rangle \otimes |0\rangle^{\otimes RF^2} \tag{18}$$

$$|\Psi_1\rangle = |+\rangle \otimes \bigotimes_{r=1}^{R} |\psi_{1,r}\rangle = \mathbf{H}|0\rangle \otimes \bigotimes_{r=1}^{R} \left(U_{RX,r}\,(x_r \cdot \pi)\,|\psi_{0,r}\rangle\right) \tag{19}$$

$$|\Psi_2\rangle = |+\rangle \otimes \bigotimes_{r=1}^{R} \mathbf{U}|\psi_{1,r}\rangle = |+\rangle \otimes \bigotimes_{r=1}^{R} |\psi_{2,r}\rangle \tag{20}$$

$$|\Psi_3\rangle = |+\rangle \otimes \mathbf{U}_c \bigotimes_{r=1}^{R} |\psi_{2,r}\rangle = |+\rangle \otimes |\psi_3\rangle \tag{21}$$

$$|\Psi_4\rangle = CP^R|\Psi_3\rangle \tag{22}$$

$$|\Psi_f\rangle = \left(\prod_{l=1}^{C/R} CP^R \cdot \mathbf{U}_c \cdot \bigotimes_{r=l}^{l+R} \left(\mathbf{U} \cdot U_{RX,r}\right)\right) |\Psi_4\rangle \tag{23}$$

This circuit is visualized in Fig. 5. Although more qubits are being used than in the CO-QCNN, the number of gates and circuit depth have been reduced.

3.3 Parallel channel overwrite - Topologically Considerate Method

The circuit construction put forth in Sect. 3.2.2 is generalizable to C channels and R registers; however, it becomes difficult to control each register from the same ancilla qubit as R becomes large. To account for this difficulty, we present an additional method that mitigates the topological challenges of the hardware being used by introducing additional ancilla qubits. In this method, the desired number of controlled phase gates can be applied to each ancilla qubit. With this Parallel Channel Overwrite - Topologically Considerate (PCO-T-QCNN) method, the ancilla qubits are entangled with learnable unitaries at the conclusion of all controlled phase gates. The full circuit is pictured in Fig. 6. The total number of qubits in this method is $RF^2 + A$, where A is the number of ancilla qubits chosen to decrease the degree of required connectivity.

3.4 Weighted expectation value method

The Weighted Expectation Value (WEV)- QCNN method adds a hybrid component to classically learn inter-channel features. The quantum convolutions are performed in the traditional sense of passing the filter over each channel individually. After the expectation values are acquired, we apply a classical weight and bias before summing each value to generate the corresponding output:

$$\tilde{x}_{i',j'} = \sum_{c=1}^{C} \langle \Psi_{i,j,c}|\sigma_z|\Psi_{i,j,c}\rangle \cdot w_{i,j,c} + b_{i,j,c} \tag{24}$$

$\tilde{x}_{i',j'}$ is a single classical output data point given by the quantum convolution. $\Psi_{i,j}$ is the wavefunction after the quantum convolution. This method is ideal for few qubits, few gates, and shallow circuit depth.

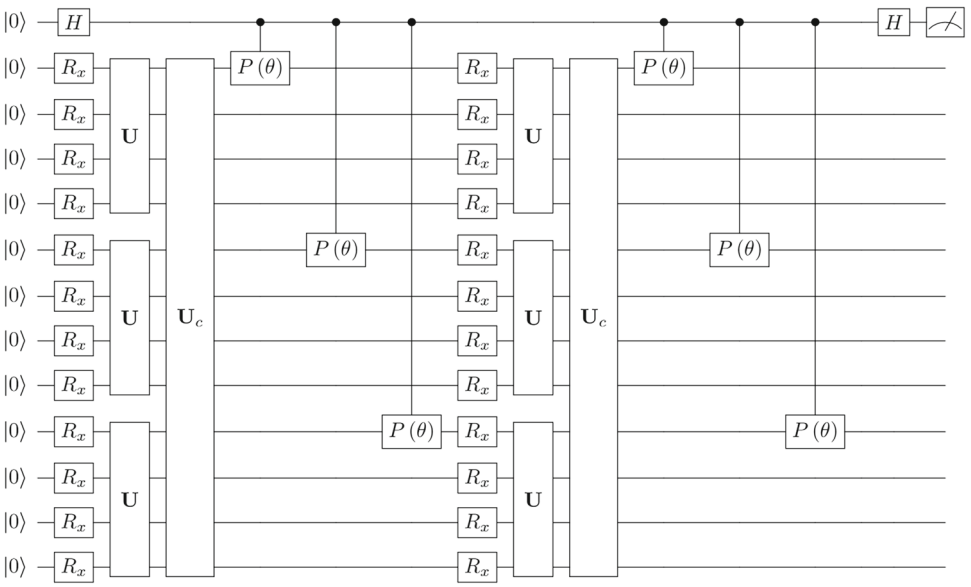


Fig. 5 Parallel Channel Overwrite (PCO) circuit with three registers using a 2×2 filter applied to six channels. $\mathbf{U}_c$ (Appendix Fig. 19) is a block of unitary gates to entangle the different working qubit channel registers

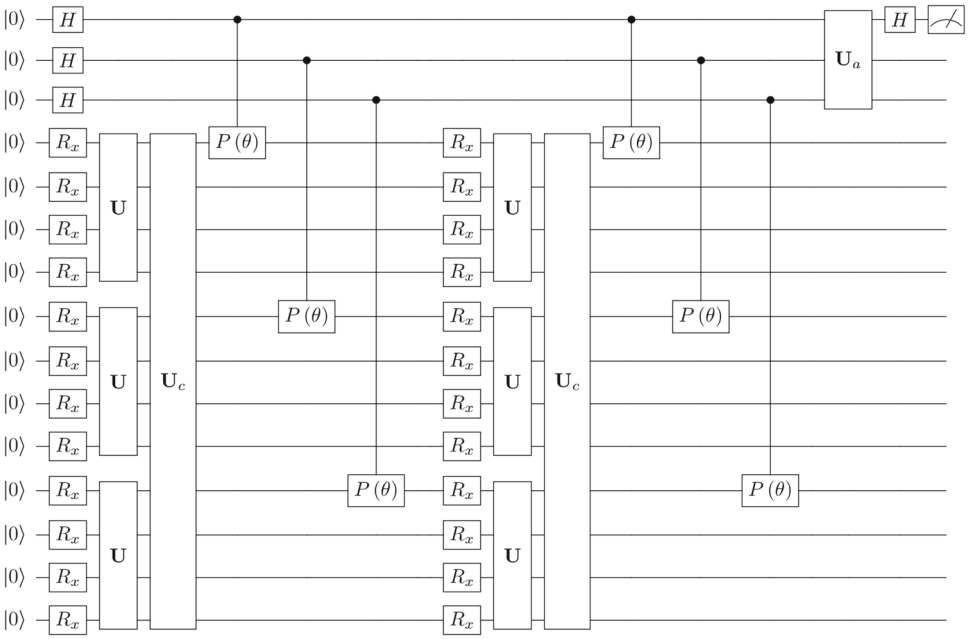


Fig. 6 Parallel Channel Overwrite - Topologically Considerate (PCO-T) circuit with three registers using a 2×2 filter applied to six channels. $\mathbf{U}_a$ (Appendix Fig. 20) is a block of unitary gates to entangle the ancilla qubits

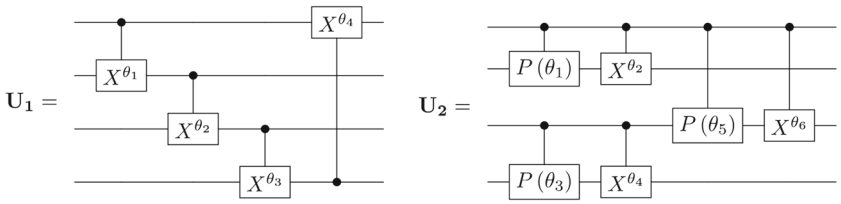


Fig. 7 The $\mathbf{U_1}$ and $\mathbf{U_2}$ circuits used in this work, visualized for a 2×2 filter

3.5 Convolutional unitary blocks

The unitary blocks used to perform the convolutions are designed to demonstrate the functionality of the proposed methods. Unitary blocks $\mathbf{U}_1$ and $\mathbf{U}_2$ are shown in Fig. 7, and are based on Schuld et al. (2020) and Oh et al. (2020), respectively. The gate X^θ represents a power of an X gate, and is described by Eq. 25. The entangling unitary blocks $\mathbf{U}_c$ and $\mathbf{U}_a$ used in the PCO and PCO-T circuits are detailed in the Appendix.

$$X^\theta = \begin{bmatrix} e^{\frac{i\pi\theta}{2}}\cos\left(\frac{\pi\theta}{2}\right) & -ie^{\frac{i\pi\theta}{2}}\sin\left(\frac{\pi\theta}{2}\right) \\ -ie^{\frac{i\pi\theta}{2}}\sin\left(\frac{\pi\theta}{2}\right) & e^{\frac{i\pi\theta}{2}}\cos\left(\frac{\pi\theta}{2}\right) \end{bmatrix} \tag{25}$$

4 Methods

4.1 Hardware

All scripting was done in Python 3.9.13. The quantum simulation package (Cirq 2021) version 0.13.1 was used, as well as Tensorflow 2.7.0 (Abadi et al. 2015) and Tensorflow Quantum 0.7.2 (Broughton et al. 2020). An Intel i7-13700KF CPU, 12GB Nvidia GeForce RTX 3080Ti GPU, and 64GB of 3600MHz CL18 RAM were used for all computations.

4.2 Datasets

The CIFAR-10 dataset (Sect. 5.2.1) was loaded with Tensorflow Keras, and the pixel values were normalized between 0 and 1. The training and testing data were segregated by class, and the first 500 data points in each class of the training data were used as the training set. The first 100 data points in each class of the test data were used as the test set. Both sets were then shuffled using scikit-learn (Pedregosa et al. 2011). After building the training and test sets, all images were downsized from $32 \times 32 \times 3$ to $10 \times 10 \times 3$ using bilinear interpolation.

The *noisy color dataset* (Sect. 5.2.2) was created by making a colored 10×10 - pixel square in Microsoft Paint using the RGB values specified in Table 1 and then images were saved as PNG files. For each color, 400 replicas of the images were loaded with Tensorflow, and 20% of the pixels were randomly corrupted in each by setting the pixel value to 0, 0, 0. The *noisy colors with shapes dataset* (Sect. 5.2.3) was prepared analogously, where the shapes were drawn by setting the pixel values to white (255, 255, 255). Each channel in both of these RGB synthetic datasets was normalized to $\frac{c}{C}$, where c is the index of the channel and C is the total number of channels. This allows the model to better distinguish between channels in the Bloch sphere.

Table 1 Noisy colors RGB values

Color	RGB value
Blue	0, 0, 255
Green	0, 255, 0
Red	255, 0, 0
Cyan	0, 255,255
Magenta	255, 0, 255
Yellow	255, 255, 0
Light cyan	128, 255, 255
Pink	255, 128, 255
Light yellow	255, 255, 128

The *synthetic 12-channel dataset* (Sect. 5.2.4) was created by initializing a $10 \times 10 \times 12$ tensor, where all elements were populated with uniformly distributed random values between 0 and 1. 0.5 was added to every element in the three specified channels that comprised a class, where i class is defined by adding 0.5 to channels i to $i + 2$. 100 replicas of each class were generated to serve as training data, and 20 replicas of each class were generated to serve as test data.

4.3 Machine learning

The learnable parameters in the quantum circuits were initialized using the Xavier method (Glorot and Bengio 2010). For the CIFAR-10 and synthetic RGB datasets (all but the *synthetic 12-channel dataset*), the classical weights and biases associated with the expectation values in the WEV-QCNN were initialized with a random normal distribution centered at 1.0 and 0.0, respectively, with both distributions having a standard deviation of 0.1. Xavier initialization was also used for the classical weights and biases when training on the *synthetic 12-channel dataset*. Categorical cross-entropy was used as the loss function, and Adam optimizer (Kingma and Ba 2014) was used to update the weights. A constant learning rate of 0.001 and was used for all datasets, except for the *synthetic 12-channel dataset* which used a learning rate of 0.01. The exponential decay rate for the first and second moment estimations was set to 0.9 and 0.999, respectively for all training. Epsilon was set to 1.0×10^{-7}. The output of each hidden layer in the model architecture was activated with ReLU, and Softmax was applied to the final output layer to obtain probabilities. The classical architecture was the same for all models. Accuracy was calculated as the ratio of images the model classified correctly to the number of total images the model attempted to classify.

5 Results and discussion

5.1 Architecture

We evaluate the proposed quantum circuits as components of a hybrid quantum-classical machine learning architecture

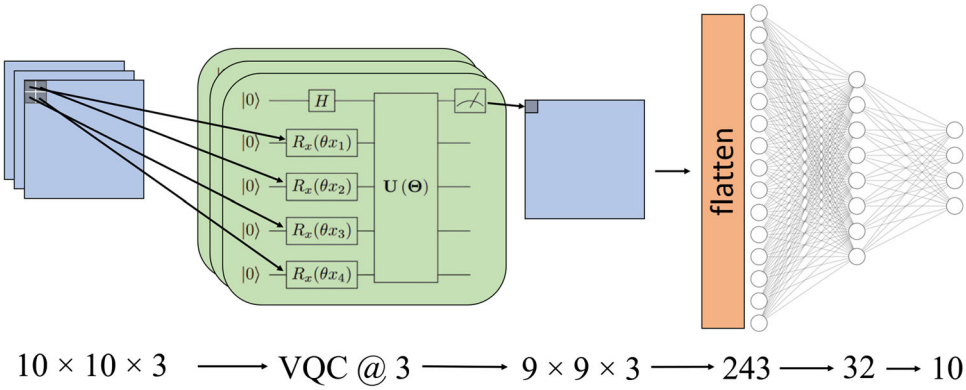
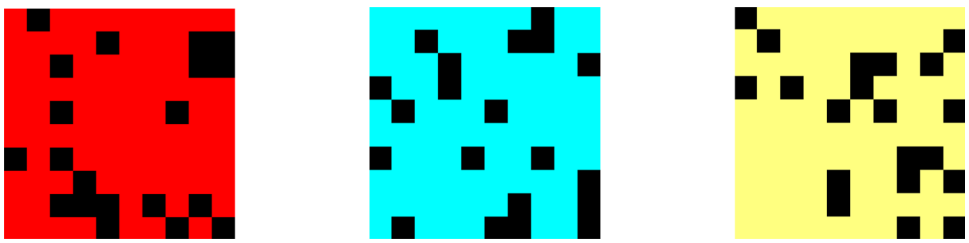


Fig. 8 Hybrid quantum-classical machine learning architecture used to evaluate each of the proposed circuits

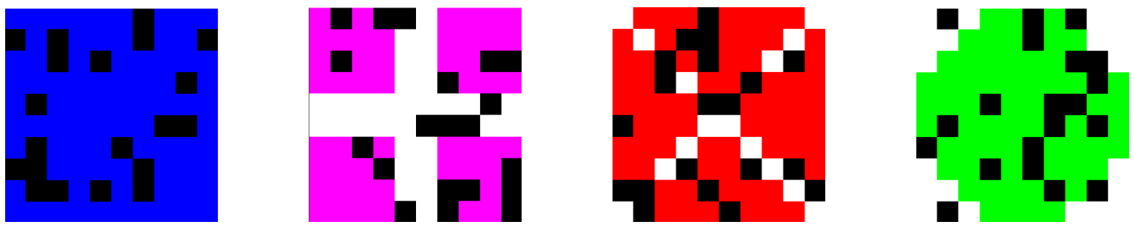
(a) Cat (b) Frog (c) Ship

Fig. 9 Three representative full-resolution (32 × 32 pixel) images from the CIFAR-10 dataset



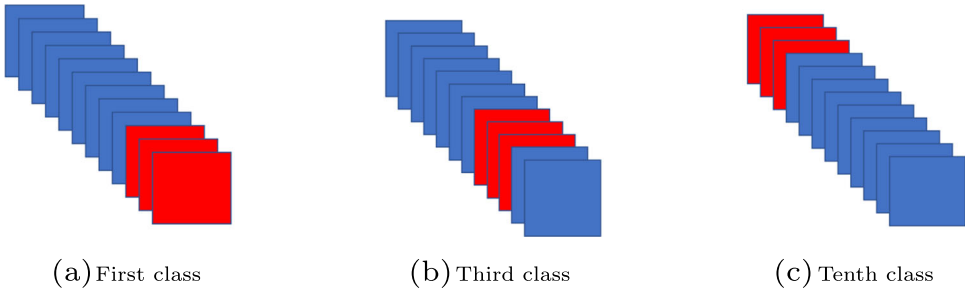
(a) Red (b) Cyan (c) Light Yellow

Fig. 10 Three representative noisy color data points from the data set. Each color in the figure has a different number of populated channels: red (255, 0, 0), cyan (0, 255, 255), and light yellow (255, 255, 128)



(a) Blue, No design (b) Magenta, Cross (c) Red, X (d) Green, Rounded

Fig. 11 Each design shape of patterned noisy color from the dataset. The designs are created using white (255, 255, 255), and then subsequently subjected to the random corruption



(a) First class (b) Third class (c) Tenth class

Fig. 12 Synthetic high-channel data. Each element in the blue colored channels are random numbers between 0 and 1. Each element in the red colored channels are random numbers between 0.5 and 1.5

for image classification. A quantum circuit is used to perform the convolution over the embedded input pixels of the image. This builds the output feature map, and a classical two-layer feed-forward network is used to obtain the probabilities for each possible class. This architecture is portrayed in Fig. 8.

5.2 Datasets

We utilize the CIFAR-10 dataset, as well as three sets of synthetic data designed to isolate the ability of our circuits to learn inter-channel information. All models are trained for 20 epochs and the hyperparameters used for each dataset are not changed between models.

5.2.1 CIFAR-10

The Canadian Institute for Advanced Research, 10 classes (CIFAR-10) (Krizhevsky 2009) is a dataset of images that has become a prominent benchmark for image recognition models. This dataset consists of 50,000 training and 10,000 testing RGB images of ten different classes: airplane, automobile, bird, cat, dog, deer, frog, horse, ship, and truck. Each image is 32×32 pixels (Fig. 9) (in this work, we rescale them to 10×10 pixels). Unless otherwise specified, we train on 500 randomly selected images from each class, and test on 100 randomly selected images from each class that do not appear in the training set.

5.2.2 Synthetic noisy colors

10×10 pixel images of nine colors are created (blue, green, red, cyan, magenta, yellow, light cyan, pink, and light yellow), and 20% of these pixels are randomly selected and corrupted by setting all channel values of the corrupted pixel to 0. Three representative images are shown in Fig. 10. In total, there are 400 images for each color (3600 total images), where 80% of these images were used for training and 20% are used for testing.

5.2.3 Synthetic noisy patterned colors

To demonstrate that this method also captures intra-channel features, four different design shapes are drawn on the synthetic colors in white before corruption. In this dataset six base colors (blue, green, red, cyan, magenta, and yellow) are used, for a total of 24 classes, four of which are shown in Fig. 11. We create 400 10×10 training images of each class, and use 80% for training and 20% for testing.

5.2.4 Synthetic high-channel data

We create a 12-channel dataset consisting of tensors of shape $10 \times 10 \times 12$ that are each populated with a uniform random distribution of numbers $\in [0, 1)$. Each of the 10 classes is defined by the addition of 0.5 to three distinct channels of the data, and is shown in Fig. 12.

5.3 CIFAR-10 dataset results

All four proposed methods CO-QCNN, PCO-QCNN, PCO-T-QCNN, and WEV-QCNN are tested against a control QCNN. This control QCNN operates by passing the quantum filter over each channel individually, acquiring an expectation value, and summing the expectation values to produce the final output.

To comprehensively evaluate the proposed QCNNs on the CIFAR-10 dataset, each model is evaluated on an n-member classification task, where n (two - ten) classes (Table 2) are contained in the training and test sets. We begin with ten classes and iteratively remove the remaining class that the CO-QCNN classifies with the least accuracy, perform-

Table 2 Classes used in CIFAR-10

Number of classes	Class that is added
2	frog, ship
3	2 + automobile
4	3 + truck
5	4 + airplane
6	5 + bird
7	6 + cat
8	7 + horse
9	8 + dog
10	9 + deer

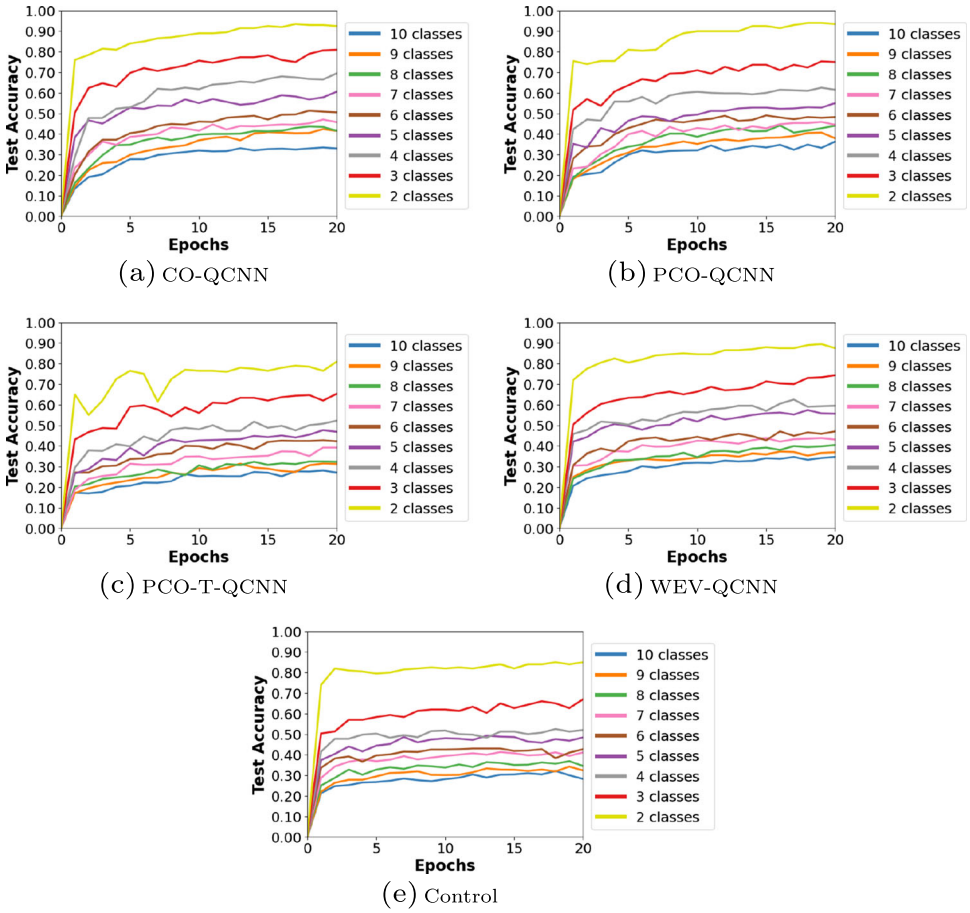


Fig. 13 Test set accuracy as a function of training epochs for the CIFAR dataset. Channel Overwrite (CO) quantum convolutional neural network (QCNN), Parallel CO (PCO)-QCNN, PCO - topologically considerate (PCO-T)-QCNN, weighted expectation value (WEV)-QCNN, and a control QCNN are evaluated for two to ten classes on the CIFAR dataset. Quantum convolutions are performed with $\mathbf{U}_1$

ing a classification at each iteration. The PCO-QCNN using $\mathbf{U}_2$ achieves the highest binary accuracy of 94.5%, compared to 85.0% with the control QCNN. When trained on all ten classes, the CO-QCNN was able to achieve 34.9% while the control QCNN produced 28.2% accuracy using $\mathbf{U}_1$. The performance of each method on variable numbers of classes is shown in Fig. 13. A comparison between each circuit is displayed for each type of classification in Fig. 14. For all classifications, the CO-QCNN, PCO-QCNN, and WEV-QCNN outperformed the control QCNN and achieve state-of-the-art accuracy compared to current quantum neural networks (Table 3).

5.4 Synthetic dataset results

In an attempt to demonstrate that the increased accuracy of the CO-QCNN, PCO-QCNN, and WEV-QCNN models can be attributed to their ability to extract features across the channel dimension, all proposed methods and the control QCNN are evaluated with synthetic datasets that emphasize inter-channel relationships. These synthetic datasets place all patterns to be learned in the channel dimension only, rather than both the spatial and channel dimension of the data. In an RGB image, the color is explicitly defined by the value of each channel. In the synthetic 12-channel dataset, the classes

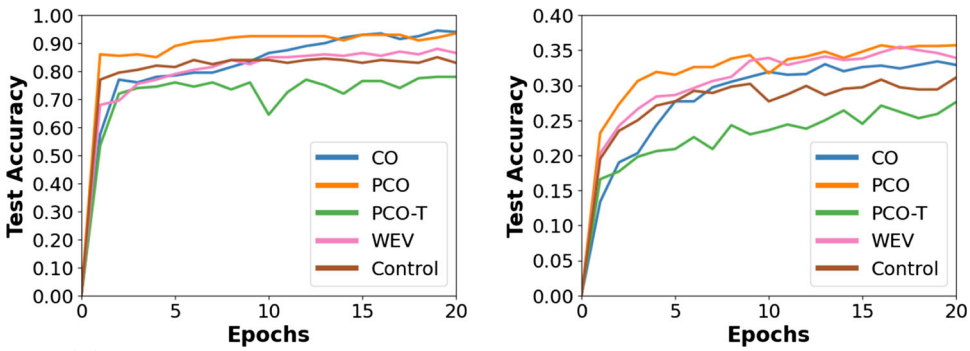


Fig. 14 Test set accuracy as a function of training epochs for the Channel Overwrite (CO) quantum convolutional neural network (QCNN), Parallel CO (PCO)-QCNN, PCO topologically considerate (PCO-T)-QCNN, weighted expectation value (WEV)-QCNN, and a control QCNN for binary and ten-member classification. Quantum convolutions shown are performed with $\mathbf{U}_2$

Table 3 Full 60,000 images CIFAR-10 image recognition

Quantum model	Test accuracy
Quanvolutional Neural Network[1] Mari (2021); Riaz et al. (2023)	34.9%
Neural Network with Quantum Entanglement [1] Riaz et al. (2023)	36.0%
Flat Quantum Convolutional Ansatz [2] Jing et al. (2022)	41.8%
CO-QCNN (U1/U2)	43.1/45.1%
PCO-QCNN (U1/U2)	44.9/44.4%
WEV-QCNN (U1/U2)	44.0/44.3%
Classical Model	Test Accuracy
CNN [3]	64.6%
CNN-P [3]	51.2%

[1] Models make no attempt to learn inter-channel information, and instead converts the images to grayscale
[2] Model is tested with images from the training data. This accuracy is a reproduction trained and tested on the full CIFAR-10 data using their exact architecture and circuit
[3] Both architectures are shown in Figs. 17 and 18 in the Appendix. CNN-P is a CNN with the same number of parameters as our work

are determined by which channels contains higher values than the others. By relegating all important learnable patterns to the channel dimension, the proposed models' abilities to extract inter-channel features are demonstrated.

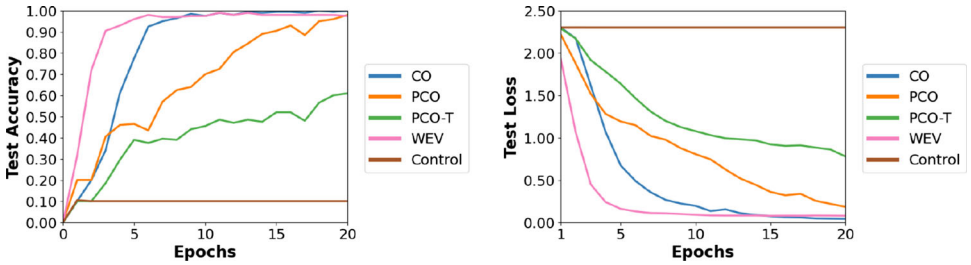
All proposed methods except PCO-T-QCNN are able to achieve 98% accuracy, with CO-QCNN achieving 100% accuracy on the *synthetic 12-channel dataset*. When predicting the color of noisy RGB images, the proposed models demonstrate the ability to learn, with the CO-QCNN and PCO-QCNN achieving 100% accuracy. The control QCNN is shown to perform much worse. The results for the *noisy colors with shapes dataset* are similar to those of the *noisy color dataset* (Figs. 15 and 16).

6 Conclusions

The proposed quantum circuits allow our QCNNs to effectively learn inter-channel information, as shown through evaluation of the models on synthetic data that holds important patterns in the channel dimension. Moreover, the CO-QCNN, PCO-QCNN, and WEV-QCNN methods achieve state-of-the-art performance on the CIFAR-10 dataset compared to current quantum neural networks. In particular, the CO-QCNN and WEV-QCNN methods achieve a greater accuracy while being computationally cheaper (Table 4) in both circuit depth and circuit width compared to this work's predecessor (Jing et al. 2022).

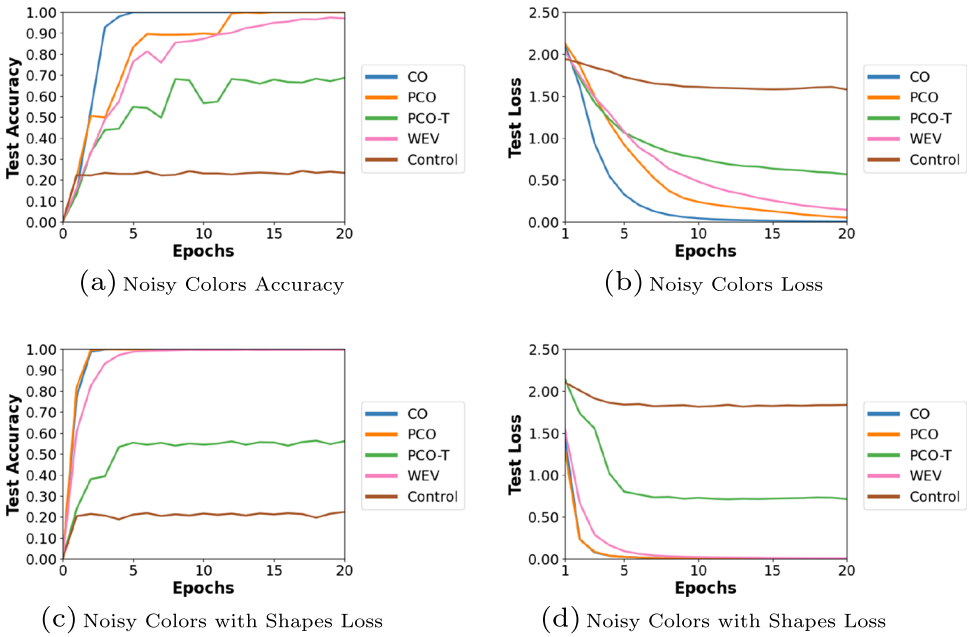
The poorer performance of the proposed PCO-T-QCNN suggests better efforts must be made to effectively correlate the multiple ancilla qubits. It is our hope that soon this method will lose relevance, as fully connected quantum devices are beginning to emerge, most recently Honeywell's 32-qubit trapped-ion quantum processor (Moses et al. 2023). As these models possess a heightened ability to learn inter-channel patterns, we wish to apply these to deeper layers in quantum-hybrid neural networks. For the CO-QCNN, PCO-QCNN, and PCO-T-QCNN, a controlled phase gate with only the first qubit of each working register and the ancilla qubit(s) is used. It is theorized higher accuracy may be achieved by performing controlled phase gates targeting all qubits in the working register. However as this work aims to create hardware considerate circuits, this is left to future projects.

Since all expectation values are weighted in the WEV-QCNN method, it is proposed that as this method is paral-



(a) 12-Channel Dataset Accuracy (b) 12-Channel Dataset Loss

Fig. 15 Test set accuracy as a function of training epochs for the Channel Overwrite (CO) quantum convolutional neural network (QCNN), Parallel CO (PCO)-QCNN, PCO topologically considerate (PCO-T)-QCNN, weighted expectation value (WEV)-QCNN, and a control QCNN for the classification of *synthetic 12-channel data*. Quantum convolutions shown are performed with $\mathbf{U}_2$



(a) Noisy Colors Accuracy (b) Noisy Colors Loss

(c) Noisy Colors with Shapes Loss (d) Noisy Colors with Shapes Loss

Fig. 16 Test set accuracy as a function of training epochs for the Channel Overwrite (CO) quantum convolutional neural network (QCNN), Parallel CO (PCO)-QCNN, PCO topologically considerate (PCO-T)-QCNN, weighted expectation value (WEV)-QCNN, and a control QCNN for the classification of synthetic RGB data. Quantum convolutions shown are performed with $\mathbf{U}_1$

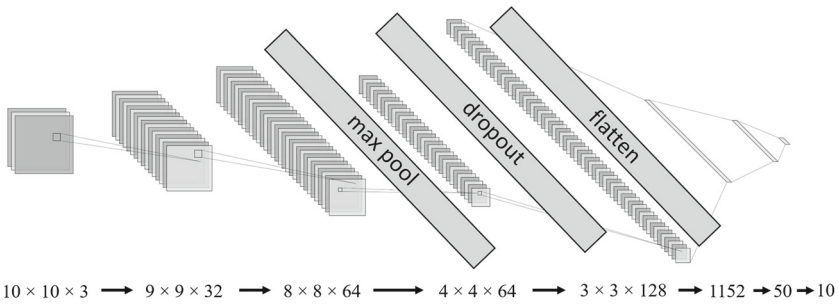
Table 4 Proposed quantum circuit complexities

Quantum model	Dataset	Circuit depth	Circuit width
FQConv (Jing et al. 2022)	CIFAR	25	12
CO-QCNN	RGB images	19	5
	High channel	73	5
PCO-QCNN	RGB images	12	13
	High channel	42	13
PCO-T-QCNN	RGB images	13	15
	High channel	40	15
WEV-QCNN	RGB images	5	4
	High channel	5	4

lelized across a quantum processing unit the classical weights will learn which qubits are better performing than others and weight them less in the final output. This could serve as a quantum error corrective technique inherent to the WEV-QCNN method. Running this method on real quantum hardware would test this theory. Beyond image recognition, we envision that these methods can be applied to problems in the natural and physical sciences.

Appendix

Fig. 17 CNN Architecture



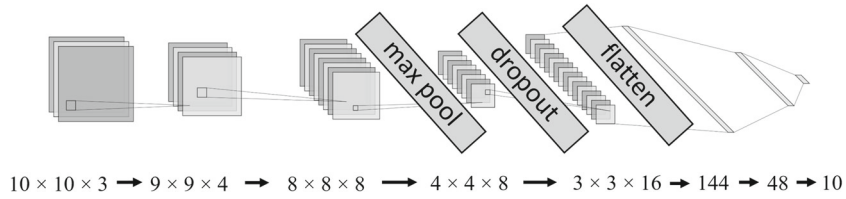


Fig. 18 CNN-P Architecture

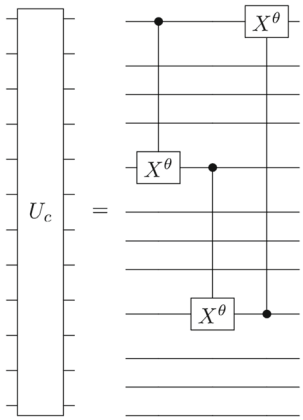


Fig. 19 Unitary block used to entangle the working registers in the DRPC and DRPC-T methods

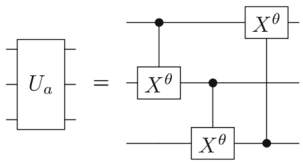


Fig. 20 Unitary block used to entangle the ancilla register in the RPCT method

Acknowledgements We acknowledge the support provided by the NSF CCI grant (Award Number 2124511).

Author contribution AMS, GWK designed research; AMS, GWK, developed software; AMS, GWK performed research; AMS, GWK, VSB analyzed data; and AMS, GWK wrote the paper. All authors have given approval to the final version of the manuscript.

Data availability All of our software including that to generate the synthetic datasets and reproduce the results from this work are available as open source at https://github.com/anthonysmaldone/QCNN-Multi-Channel-Supervised-Learning

Declarations

Competing interests The authors declare no competing financial interests.

References

Abadi M, Agarwal A, Barham P, Brevdo E,Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org. https://www.tensorflow.org/

Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, Biswas R, Boixo S, Brandao FGSL, Buell DA, Burkett B, Chen Y, Chen Z, Chiaro B, Collins R, Courtney W, Dunsworth A, Farhi E, Foxen B, Fowler A, Gidney C, Giustina M, Graff R, Guerin K, Habegger S, Harrigan MP, Hartmann MJ, Ho A, Hoffmann M, Huang T, Humble TS, Isakov SV, Jeffrey E, Jiang Z, Kafri D, Kechedzhi K, Kelly J, Klimov PV, Knysh S, Korotkov A, Kostritsa F, Landhuis D, Lindmark M, Lucero E, Lyakh D, Mandrá S, McClean JR, McEwen M, Megrant A, Mi X, Michielsen K, Mohseni M, Mutus J, Naaman O, Neeley M, Neill C, Niu MY, Ostby E, Petukhov A, Platt JC, Quintana C, Rieffel EG, Roushan P, Rubin NC, Sank D, Satzinger KJ, Smelyanskiy V, Sung KJ, Trevithick MD, Vainsencher A, Villalonga B, White T, Yao ZJ, Yeh P, Zalcman A, Neven H, Martinis JM (2019) Quantum supremacy using a programmable superconducting processor. Nature 574:505–510. https://doi.org/10.1038/s41586-019-1666-5

Ballester PJ, Mitchell JBO (2010) A machine learning approach to predicting protein-ligand binding affinity with applications to molecular docking. Bioinformatics 26(9):1169–1175. https://doi.org/10.1093/bioinformatics/btq112 https://academic.oup.com/bioinformatics/article-pdf/26/9/1169/48856826/bioinformatics_26_9_1169.pdf

Bernstein E, Vazirani U (1997) Quantum complexity theory. SIAM J Comput 26:1411–1473. https://doi.org/10.1137/S0097539796300921

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature 549(7671):195–202

Bonetta R, Valentino G (2020) Machine learning techniques for protein function prediction. Proteins: structure, function, and bioinformatics 88(3):397–413. https://doi.org/10.1002/prot.25832https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.25832

Broughton M, Verdon G, McCourt T, Martinez AJ, Yoo JH, Isakov SV, Massey P, Halavati R, Niu MY, Zlokapa A, Peters E, Lockwood O, Skolik A, Jerbi S, Dunjko V, Leib M, Streif M, Dollen DV, Chen H, Cao S, Wiersema R, Huang H-Y, McClean JR, Babbush R, Boixo S, Bacon D, Ho AK, Neven H, Mohseni M (2020) Tensorflow quantum: a software framework for quantum machine learning

Casey AD, Son SF, Bilionis I, Barnes BC (2020) Prediction of energetic material properties from electronic structure using 3d convolutional neural networks. J Chem Inf Model 60:4457–4473. https://doi.org/10.1021/acs.jcim.0c00259

Cerezo M, Verdon G, Huang H-Y, Cincio L, Coles PJ (2022) Challenges and opportunities in quantum machine learning. Nature Comput Sci 2:567–576. https://doi.org/10.1038/s43588-022-00311-3

Chen X, Jørgensen MS, Li J, Hammer B (2018) Atomic energies from a convolutional neural network. J Chem Theory Comput 14:3933–3942. https://doi.org/10.1021/acs.jctc.8b00149

Chen SY-C, Wei T-C, Zhang C, Yu H, Yoo S (2022) Quantum convolutional neural networks for high energy physics data analysis. Phys

Rev Res 4:013231. https://doi.org/10.1103/PhysRevResearch.4.013231
Choi H, Kang H, Chung K-C, Park H (2019) Development and application of a comprehensive machine learning program for predicting molecular biochemical and pharmacological properties. Phys Chem Chem Phys 21:5189–5199. https://doi.org/10.1039/C8CP07002D
Chow J, Dial O, Gambetta J (2021) Ibm quantum breaks the 100-qubit processor barrier. IBM Research Blog
Cirq (2021) A python framework for creating, editing, and invoking noisy intermediate scale quantum (nisq) circuits. https://doi.org/10.5281/zenodo.5182845, https://github.com/quantumlib/Cirq
Cleve R, Ekert A, Macchiavello C, Mosca M (1998) Quantum algorithms revisited. Proceedings of the royal society of London. Series A: Mathematical, Physical and Engineering Sciences 454:339–354. https://doi.org/10.1098/rspa.1998.0164
Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15:1273–1278. https://doi.org/10.1038/s41567-019-0648-8
Daily M, Medasani S, Behringer R, Trivedi M (2017) Self-driving cars. Computer 50(12):18–23
Deutsch D, Jozsa R (1992) Rapid solution of problems by quantum computation. Proceedings of the royal society of London. Series A: Mathematical and Physical Sciences 439: 553–558. https://doi.org/10.1098/rspa.1992.0167
Feynman RP (1982) Simulating physics with computers. Int J Theor Phys 21:467–488. https://doi.org/10.1007/BF02650179
Fujiyoshi H, Hirakawa T, Yamashita T (2019) Deep learning-based image recognition for autonomous driving. IATSS Research 43(4):244–252. https://doi.org/10.1016/j.iatssr.2019.11.008
Gao H, Sun L, Wang J-X (2021) Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. J Comput Phys 428:110079. https://doi.org/10.1016/j.jcp.2020.110079
Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. J Mach Learn Res 9(Jun): 249–256
Grover LK (1996) A fast quantum mechanical algorithm for database search, pp 212–219. ACM Press, New York. https://doi.org/10.1145/237814.237866
Gupta A, Anpalagan A, Guan L, Khwaja AS (2021) Deep learning for object detection and scene perception in self-driving cars: survey, challenges, and open issues. Array 10
Henderson M, Shakya S, Pradhan S, Cook T (2019) Quanvolutional neural networks: powering image recognition with quantum circuits
Hong Z, Wang J, Qu X, Zhu X, Liu J, Xiao J (2021) Quantum convolutional neural network on protein distance prediction, pp 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533405
Hur T, Kim L, Park DK (2022) Quantum convolutional neural network for classical data classification. Quantum Mach Intell 4:3. https://doi.org/10.1007/s42484-021-00061-x
Jing Y, Li X, Yang Y, Wu C, Fu W, Hu W, Li Y, Xu H (2022) Rgb image classification with quantum convolutional ansatz. Quantum Inf Process 21:101. https://doi.org/10.1007/s11128-022-03442-8
Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A, Bridgland A, Meyer C, Kohl SAA, Ballard AJ, Cowie A, Romera-Paredes B, Nikolov S, Jain R, Adler J, Back T, Petersen S, Reiman D, Clancy E, Zielinski M, Steinegger M, Pacholska M, Berghammer T, Bodenstein S, Silver D, Vinyals O, Senior AW, Kavukcuoglu K, Kohli P, Hassabis D (2021) Highly accurate protein structure prediction with alphafold. Nature 596:583–589. https://doi.org/10.1038/s41586-021-03819-2
Kim D, Kim S-H, Kim T, Kang BB, Lee M, Park W, Ku S, Kim D, Kwon J, Lee H et al (2021) Review of machine learning methods in soft robotics. Plos one 16(2):0246102
Kingma DP, Ba J (2014) Adam: A method for stochastic optimization
Kitaev AY (1995) Quantum measurements and the abelian stabilizer problem
Kleeberger K, Bormann R, Kraus W, Huber MF (2020) A survey on learning-based robotic grasping. Current Robotics Reports 1:239–249
Krizhevsky A (2009) Learning multiple layers of features from tiny images
Kyro GW, Brent RI, Batista VS (2023) Hac-net: a hybrid attention-based convolutional neural network for highly accurate protein-ligand binding affinity prediction. J Chem Inf Model 63:1947–1960. https://doi.org/10.1021/acs.jcim.3c00251
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
Lesort T, Lomonaco V, Stoian A, Maltoni D, Filliat D, Díaz-Rodríguez N (2020) Continual learning for robotics: definition, framework, learning strategies, opportunities and challenges. Information Fusion 58:52–68
Li H (2018) Deep learning for natural language processing: advantages and challenges. National Sci Rev 5(1):24–26
Liu L, Wang Y, Chi W (2020) Image recognition technology based on machine learning. IEEE Access 1–1. https://doi.org/10.1109/ACCESS.2020.3021590
Mari A (2021) Quanvolutional neural networks - Pennylane. https://pennylane.ai/qml/demos/tutorial_quanvolution.html
Menborong (2020) Simple-QCNN. GitHub. https://github.com/Menborong/Simple-QCNN
Mishra S, Tsai C-Y (2023) Qsurfnet: a hybrid quantum convolutional neural network for surface defect recognition. Quantum Inf Process 22:179. https://doi.org/10.1007/s11128-023-03930-5
Moses SA, Baldwin CH, Allman MS, Ancona R, Ascarrunz L, Barnes C, Bartolotta J, Bjork B, Blanchard P, Bohn M, Bohnet JG, Brown NC, Burdick NQ, Burton WC, Campbell SL, Campora JP, Carron C, Chambers J, Chan JW, Chen YH, Chernoguzov A, Chertkov E, Colina J, Curtis JP, Daniel R, DeCross M, Deen D, Delaney C, Dreiling JM, Ertsgaard CT, Esposito J, Estey B, Fabrikant M, Figgatt C, Foltz C, Foss-Feig M, Francois D, Gaebler JP, Gatterman TM, Gilbreth CN, Giles J, Glynn E, Hall A, Hankin AM, Hansen A, Hayes D, Higashi B, Hoffman IM, Horning B, Hout JJ , Jacobs R, Johansen J, Jones L, Karcz J, Klein T, Lauria P, Lee P, Liefer D, Lytle C, Lu ST, Lucchetti D, Malm A, Matheny M, Mathewson B, Mayer K, Miller DB, Mills M, Neyenhuis B, Nugent L, Olson S, Parks J, Price GN, Price Z, Pugh M, Ransford A, Reed AP, Roman C, Rowe M, Ryan-Anderson C, Sanders S, Sedlacek J, Shevchuk P, Siegfried P, Skripka T, Spaun B, Sprenkle RT, Stutz RP, Swallows, M, Tobey RI, Tran A, Tran T, Vogt E, Volin C, Walker J, Zolot AM, Pino JM (2023) A race track trapped-ion quantum processor
Mozaffari S, Al-Jarrah OY, Dianati M, Jennings P, Mouzakitis A (2020) Deep learning-based vehicle behavior prediction for autonomous driving applications: a review. IEEE Trans Intell Transp Syst 23(1):33–47
Oh S, Choi J, Kim J (2020) A tutorial on quantum convolutional neural networks (qcnn)
OpenAI (2023) Gpt-4 technical report
OpenAI (2023) New gpt-3 capabilities: edit & insert
Otter DW, Medina JR, Kalita JK (2020) A survey of the usages of deep learning for natural language processing. IEEE Trans Neural Netw Learn Syst 32(2):604–624
Pak M, Kim S (2017) A review of deep learning in image recognition. In: 2017 4th international conference on computer applications and information processing technology (CAIPT), pp 1–3. https://doi.org/10.1109/CAIPT.2017.8320684
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E

(2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Pierson HA, Gashler MS (2017) Deep learning in robotics: a review of recent research. Adv Robot 31:821–835. https://doi.org/10.1080/01691864.2017.1365009

Preskill J (2018) Quantum computing in the nisq era and beyond. Quantum 2:79. https://doi.org/10.22331/q-2018-08-06-79

Rao Q, Frtunikj J (2018) Deep learning for self-driving cars: chances and challenges. In: Proceedings of the 1st international workshop on software engineering for AI in autonomous systems, pp 35–38

Rawat W, Wang Z (2017) Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 29:2352–2449. https://doi.org/10.1162/neco_a_00990

Riaz F, Abdulla S, Suzuki H, Ganguly S, Deo RC, Hopkins S (2023) Accurate image multi-class classification neural network model with quantum entanglement approach. Sensors 23:2753. https://doi.org/10.3390/s23052753

Schuld M, Sinayskiy I, Petruccione F (2015) An introduction to quantum machine learning. Contemp Phys 56(2):172–185

Schuld M, Bocharov A, Svore KM, Wiebe N (2020) Circuit-centric quantum classifiers. Phys Rev A 101(3):032308

Senior AW, Evans R, Jumper J, Kirkpatrick J, Sifre L, Green T, Qin C, Žídek A, Nelson AWR, Bridgland A, Penedones H, Petersen S, Simonyan K, Crossan S, Kohli P, Jones DT, Silver D, Kavukcuoglu K, Hassabis D (2020) Improved protein structure prediction using potentials from deep learning. Nature 577:706–710. https://doi.org/10.1038/s41586-019-1923-7

Shor PW (1994) Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science, pp 124–134. https://doi.org/10.1109/SFCS.1994.365700

Simon DR (1997) On the power of quantum computation. SIAM J Comput 26:1474–1483. https://doi.org/10.1137/S0097539796298637

Sino NI, Farhan RN (1804) Seno ME (2021) Review of deep learning algorithms in computational biochemistry. J Phys D Appl Phys 1:012135. https://doi.org/10.1088/1742-6596/1804/1/012135

Wang W, Siau K (2019) Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: a review and research agenda. J Database Manag (JDM) 30(1):61–79

Wei Z, Chen X (2019) Physics-inspired convolutional neural network for solving full-wave inverse scattering problems. IEEE Trans Antennas Propag 67:6138–6148. https://doi.org/10.1109/TAP.2019.2922779

Wu M, Chen L (2015) Image recognition based on deep learning. In: 2015 Chinese automation congress (CAC), pp 542–546. https://doi.org/10.1109/CAC.2015.7382560

Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. IEEE Comput Intell Mag 13(3):55–75

Zhang Y, Ni Q (2020) Recent advances in quantum machine learning. Quantum Engineering 2(1):34

Zhong H-S, Wang H, Deng Y-H, Chen M-C, Peng L-C, Luo Y-H, Qin J, Wu D, Ding X, Hu Y, Hu P, Yang X-Y, Zhang W-J, Li H, Li Y, Jiang X, Gan L, Yang G, You L, Wang Z, Li L, Liu N-L, Lu C-Y, Pan J-W (2020) Quantum computational advantage using photons. Science 370:1460–1463. https://doi.org/10.1126/science.abe8770