Quantum Machine Intelligence (2025) 7:51
https://doi.org/10.1007/s42484-025-00279-z

RESEARCH ARTICLE

LHQNN: sequential and non-sequential layered hybrid quantum neural networks for image classification

Monika Kabir[1] · Mohammed Kaosar[1] · Hamid Laga[1] · Ferdous Sohel[1]

Received: 14 December 2024 / Accepted: 2 April 2025

Abstract

Quantum neural networks have emerged as a promising approach to solving complex problems across various domains, especially when integrated with classical methods. Several hybrid quantum-classical architectures have been developed to leverage the potential of quantum advantages for image classification tasks. The design of the quantum layer plays an important role in exploiting quantum properties such as superposition and entanglement. In this research, we propose hybrid quantum neural networks with multiple quantum layers, utilizing sequential circuits for enhanced feature representation through structured depth, and non-sequential circuits to reduce complexity and improve performance. Our experimental results demonstrate that stacking multiple layers in the quantum circuit enhances performance significantly. Furthermore, the results indicate that the optimal range of 6–10 qubits achieves the best trade-off between accuracy and computational efficiency. The results also show that amplitude embedding consistently outperformed angle embedding for image classification tasks. Notably, our proposed hybrid sequential model with amplitude embedding outperforms traditional convolutional neural networks on MNIST and Fashion-MNIST datasets, while requiring fewer parameters. These findings provide valuable insights for advancing quantum machine learning in real-world applications.

Keywords Hybrid quantum neural networks · Quantum embedding · Image classification · Quantum machine learning

1 Introduction

Quantum computers are considered more powerful than classical computers due to the potentiality of parallel calculations and exponential speed-ups. Currently, in the noisy intermediate-scale quantum (NISQ) era, quantum devices face several challenges, such as a limited number of qubits and high error rates (Preskill 2018). Due to these hardware constraints, replacing classical computing entirely with quantum is not feasible. A key challenge is harnessing the properties of quantum computing while integrating them with existing classical approaches.

Quantum computing offers novel approaches for various research problems integrated with traditional machine learning concepts (Biamonte et al. 2017). In addition to quantum machine learning approaches (Rebentrost et al. 2013; Lloyd et al. 2014; Amin et al. 2018), deep learning models have also gained quantum advantages for image classification (Sagingalieva et al. 2025; Mari et al. 2020). One of the widely accepted deep learning approaches in image processing is the convolution neural network (CNN) (Krizhevsky et al. 2012). Inspired by the capability of this architecture, various quantum-classical hybrid classification models have been developed (Warrier et al. 2024; Wang et al. 2024).

One of the prominent hybrid approaches is quantum CNN (QCNN) (Cong et al. 2019), where filters are modified with quantum-based quanvolution filters for feature extraction (Henderson et al. 2019; Vu et al. 2024). Due to the hardware limitations of quantum, these approaches often lead to lower accuracy compared to classical convolution filters in practical applications. Different optimization techniques have been applied to design parameterized quantum circuits

✉ Ferdous Sohel
F.Sohel@murdoch.edu.au

Monika Kabir
35009367@student.murdoch.edu.au

Mohammed Kaosar
mohammed.kaosar@murdoch.edu.au

Hamid Laga
H.Laga@murdoch.edu.au

[1] School of Information Technology, Murdoch University, 90 South Street, Perth 6150, WA, Australia

Published online: 09 April 2025

(PQCs) and leverage the quantum properties with classical architecture (Liao et al. 2024). Optimizing these PQCs plays a crucial role in enhancing the performance of quantum circuits in quantum neural networks (QNN) (Zhang et al. 2024). The integration of QNN with the classical models, known as HQNN, has shown promising advances in image classification tasks (Senokosov et al. 2024; Ling et al. 2024).

In this paper, we propose layered hybrid quantum neural networks for multi-class image classification. In the proposed model, both sequential and non-sequential quantum layer (QL) configurations are designed, with varying the depth of the circuit. Rather than optimizing the quantum circuit, our motivation is to study the deeper quantum network to utilize the quantum properties, i.e., superposition and entanglement, for image classification. The performance of the proposed architectures is evaluated with respect to several variables, such as the optimal range of qubits, embedding techniques, and the structure of the quantum circuit. The key contributions of the proposed research are as follows:

- We propose sequential and non-sequential layered hybrid quantum neural networks (LHQNN) for image classification problems. Multiple layers in these hybrid architectures harness quantum properties effectively, with sequential models providing structured depth and non-sequential models minimizing circuit length for improved efficiency. Our proposed models explore quantum circuits with varying numbers of stacked quantum layers that outperform traditional approaches.
- We investigate how the number of qubits and embedding techniques, such as amplitude and angle encoding, affect the performance of LHQNN models, focusing on their impact on accuracy and computational efficiency.
- We analyze the effects of the complexity and the size of quantum circuits on models performance, highlighting the role of circuit depth in leveraging quantum properties. This research provides valuable insights into designing optimal resource configuration models and enhanced accuracy.

The rest of the paper is organized as follows: In Section 2, we summarize the basic preliminaries for quantum computing. The proposed LHQNN architectures are introduced in Section 3. The experimental setup and analysis of results are presented in Sections 4 and 5, respectively. Finally, Section 6 concludes with a summary and future direction of the research.

2 Preliminaries

This section presents a brief background and preliminaries related to this paper. It focuses on basic quantum computing concepts, such as quantum bits, quantum encoding techniques, quantum circuits, and different types of quantum gates. These concepts and notations will be useful in comprehending the terminology of the proposed LHQNN architectures in this paper.

2.1 Quantum bits

Quantum bit or qubit is the fundamental component of information in quantum computing. In classical computing, bits hold two states- 0 and 1 for storage and processing information. On the other hand, qubits can have a mixed state of 0 and 1 (Fig. 1) (Team 2024). Based on quantum the-

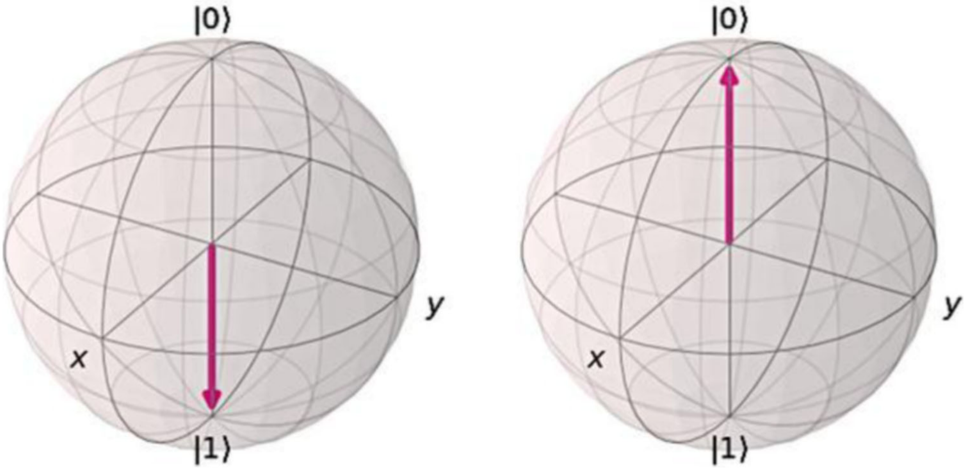


Fig. 1 The figure shows two Bloch spheres representing the quantum bit in classical states. The left and right spheres illustrate the qubit in state $|1\rangle$ and $|0\rangle$, respectively

ory, a qubit exhibits some properties such as superposition, entanglement, and interference, which make it distinguishable from classical bits (Nielsen and Chuang 2010).
In quantum computing, to represent the quantum states of a qubit, a mathematical notation is used, which is called Dirac notation or bra-ket notation. According to Dirac notation, mainly two types of symbols, ket ($|\Psi\rangle$) and bra ($\langle\Psi|$) are used to describe any instant of a qubit. The computational basis state of a qubit can be represented by column vectors in multidimensional Hilbert Space using these notations (Griffiths and Schroeter 2018). The state of vector a using ket notation can be described as,

$$|a\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{1}$$

Mainly, two computational basis states $|0\rangle$ and $|1\rangle$ are used for quantum and can be expressed as a matrix as follows,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{2}$$

2.2 Quantum gates and circuits

In quantum computing, quantum gates and circuits are fundamental components for designing quantum models. Mainly, quantum gates are mathematical operations that act on one or more qubits to manipulate the quantum state (Mermin 2007). The Hadamard, CNOT, Rotation, and Pauli gates are the most frequently used gates in quantum computing.

Hadamard gate The Hadamard gate is used to create superposition on qubits. It performs transformation from state $|0\rangle$ or $|1\rangle$ into a superposition of $|0\rangle$ and $|1\rangle$. Mathematically, it can be represented as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{3}$$

CNOT gate CNOT gate is an effective two-qubit gate that establishes entanglement between qubits. These two qubits are known as control and target qubits. With this gate, depending on the value of the control qubits, the state target qubits change.

$$\mathrm{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

Rotation gates Rotation gates are single-qubit gates with parameterize of angle θ that perform rotation around the X, Y, or Z axes of the Bloch sphere.

$$R_x(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{5}$$

$$R_y(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{6}$$

$$R_z(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Z = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{7}$$

Pauli gates Pauli gates manipulate the quantum state of a qubit by rotation of the X, Y, or Z axes. With these gates, bit, phase, or both states of a quantum bit can be modified with Pauli X, Y, and Z gates, respectively.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{8}$$

Quantum circuit A quantum circuit is a set of quantum gates that are used in quantum algorithms to manipulate the state of qubits. In quantum circuits, superposition and entanglement are created with one and two-qubit gates, respectively. At the end of the circuit, the measurement gate measures the quantum state to extract the classical information.

2.3 Quantum encoding

In quantum computing, classical data are encoded into quantum bits to leverage the principles of quantum mechanics (Lloyd et al. 2020). In this research, two of the most popular embedding techniques- amplitude encoding and angle encoding are used.

2.3.1 Amplitude encoding

In amplitude encoding, classical information is represented by the probability amplitudes of a quantum state. To represent information using quantum superposition principles, the states are observed to determine the outcome from probability amplitude in quantum superposition (Schuld and Petruccione 2018). The main advantage of this technique is that a large amount of data can be represented with a limited number of qubits. Specifically, a classical dataset of M inputs, each having N number of features, can be encoded with $n = log N M$ number of qubits.
To encode classical data into quantum, the classical vectors are normalized using the Euclidean norm. If X is a classical vector of dataset $x = (x_1, x_2, ..., x_n)$, the corresponding

quantum state $|\psi\rangle$ can be represented as,

$$|\psi\rangle = \frac{1}{\|x\|} \sum_{i=0}^{N-1} x_i |i\rangle \tag{9}$$

Again, with the computational basis state $|i\rangle$ and the i-th component of classical data x_i, normalization can be calculated as,

$$\|x\| = \sqrt{\sum_{i=0}^{N-1} x_i^2} \tag{10}$$

Depending on the number of classical data vectors, the initial state $|0\rangle^{\otimes n}$ is prepared, where n determines the number of qubits with all-zero initial states. Afterward, a series of different unitary gates, such as rotation gates, controlled gates, etc., are applied to the initial state to transform the initial state into the encoded quantum state.

2.3.2 Angle encoding

Angle encoding is a widely used embedding technique in quantum computing to encode classical data into quantum states. In angle embedding, classical data are encoded into quantum states by a rotation about X, Y, or Z axes by a specific angle.

A classical dataset x having n number of features $x = (x_1, x_2, ..., x_n)$, can be encoded with rotation about Y axis such as:

$$|\psi(x)\rangle = R_y(x_1) \otimes R_y(x_2) \otimes \cdots \otimes R_y(x_n)|0\rangle^{\otimes n} \tag{11}$$

where $|0\rangle$ is the initial state, and each feature is encoded with an individual angle ($\otimes$) to generate a quantum state $|\psi\rangle$. In general, any classical dataset having n number of features requires the same amount of qubits. For large and complex datasets with a high number of features, it becomes impractical to use angle encoding for hardware limitations due to quantum computers.

3 Proposed methodology

This section describes the architecture of the proposed layered hybrid quantum neural networks. The generalized architecture of the proposed models is depicted in Fig. 2a. The convolution block of CNN extracts significant features from input images using convolution layers. It typically includes an activation function, Rectified Linear Unit (ReLU) for non-linearity, and a pooling layer to reduce spatial dimensions. These components worked together to capture meaningful patterns for dense layers. In the dense layers, the quantum layer is integrated with a fully connected layer for classification. Notably, the design incorporates a single fully connected layer to maximize the utilization of quantum properties. Further details of each component of LHQNN architectures are presented in the subsequent sections.

3.1 Convolution and pooling layers

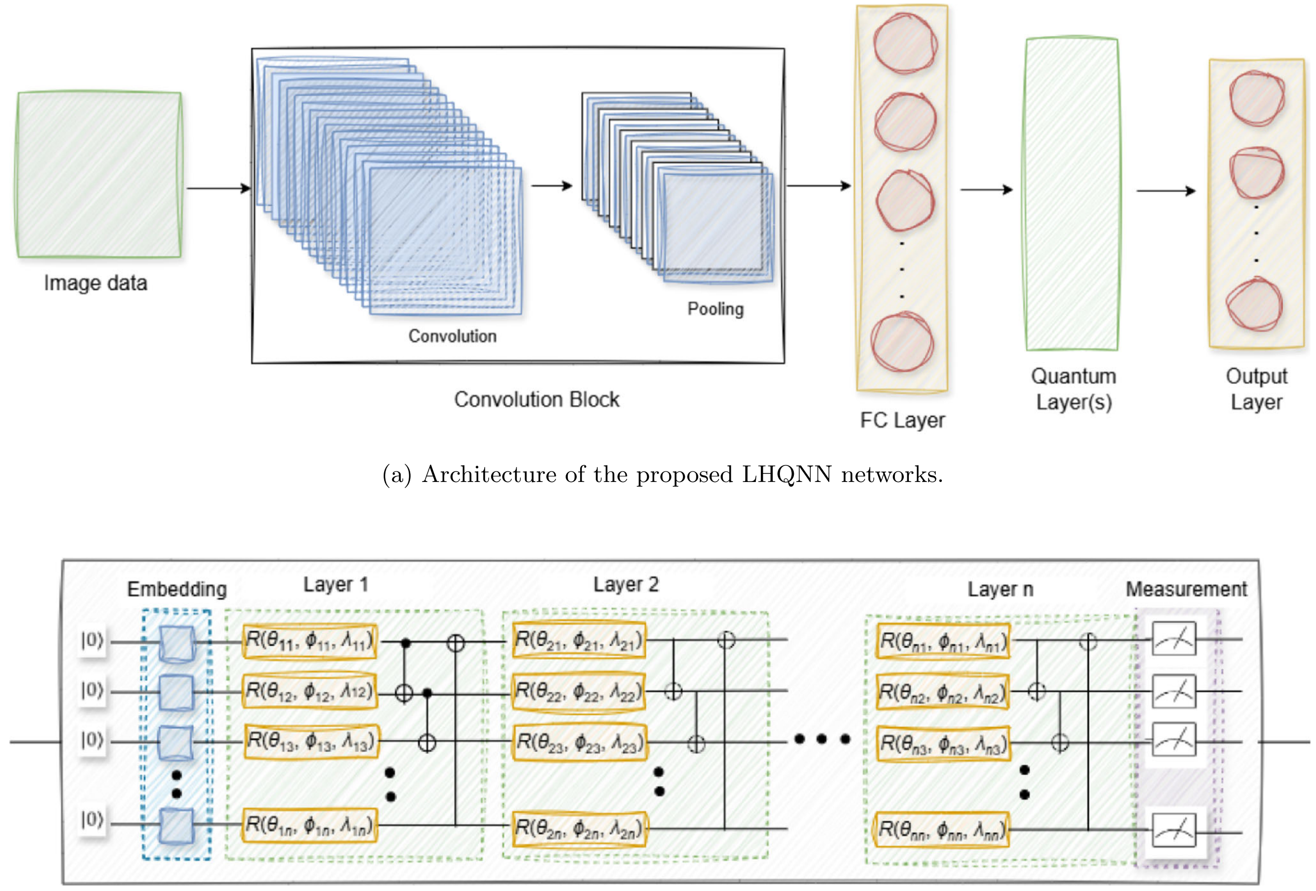
The first module of LHQNN is classical convolution, which is used to identify significant patterns and extract features from images. In the first convolutional layer, with 16 output channels, a square kernel of dimension 3×3 is employed. Smaller kernels can effectively capture these fine-grained details, which is crucial for our datasets. In addition, a single-pixel stride and padding are utilized for the input data to preserve the spatial dimensions. This layer is followed by a pooling layer for down-sampling the input with an activation function ReLU. In this layer, max-pooling with a kernel size of two pixels, stride with two pixels, and zero padding is applied, resulting in a feature map having dimensions of 16×14×14 pixels.

The second convolutional layer has 16 input channels and 32 output channels, preserving the stride and padding configuration from the first layer. The output is passed through a pooling layer, which reduces the spatial dimensions while retaining crucial features. The output of this layer generates a feature map with dimensions 32×7×7. This low-dimensional feature map is subsequently fed into the next layers for further processing.

3.2 Flatten and fully connected layer

In the flattened layer, the multidimensional feature space generated from the previous layers is converted into a one-dimensional vector. Mainly, 3D tensors of images (height, width, and channels) are flattened into a 1D vector by combining their dimensions. From the final feature map of the convolutional module, a vector of dimension 1568 is generated. As the research focuses on leveraging the strength of quantum properties, only one fully connected layer is utilized before the quantum layer.

The vector in the flattened layer is fed into the first fully connected layer. The patterns of the extracted features by the convolution layer are learned by the model in this layer. An activation function, ReLU, is applied here to identify the non-linear relationship among the features and pass it into the quantum layer (Agarap 2019). In the proposed LHQNN architecture, different numbers and shapes of quantum layers are infused between fully connected layers. Finally, the last fully connected layer performs the classification depending on the number of classes in the dataset. Notably, since the research aims to harness the power of quantum properties,



(a) Architecture of the proposed LHQNN networks.

(b) Detailed architecture of the quantum layer.

Fig. 2 **a** Architecture of the proposed LHQNN networks. A series of convolution and pooling layers are applied to input data to extract significant features and reduce the spatial dimensions. The outputs are flattened into a single vector and passed to a hybrid dense layer that combines classical neural network components. The final output layer is used to classify the input image into predefined categories. **b** Detailed architecture of quantum layer. Features are encoded and distributed in a quantum circuit. A parameterized quantum circuit is implemented using CNOT and entangling gates across n layers to capture and learn complex feature spaces. The resultant quantum states are measured, and the outputs are fed into the final classification layer

only a single fully connected layer is employed before the quantum layer.

3.3 Quantum layer

In the proposed LHQNN models, the quantum layer plays a key role in leveraging quantum properties for enhanced image classification. Quantum layers mainly consist of three components- embedding, quantum circuit, and measurement. A detailed architecture of the quantum layer is depicted in Fig. 2b.

The fully connected layer connects all input features into n output features, where n represents the number of qubits. Classical data are encoded into quantum states using a primary embedding technique. This research investigates the performance of both amplitude and angle embedding to assess their effectiveness in image classification tasks. A unitary transformation is applied in the encoding operation E to map the classical data vector x onto a quantum state, transforming the initial state into a quantum representation. If initial states are represented by $|0\rangle$, for n qubits, the encoded state can be expressed as follows.

$$|\psi_{\text{encoded}}\rangle = E(x)|0\rangle^{\otimes n} \tag{12}$$

The encoded quantum states are fed into the quantum circuit, where strongly entangling layers (SEL) architecture is utilized as a quantum circuit (Schuld et al. 2020). SEL increases the number of superposed quantum states and entanglement between qubits, which is crucial for achieving quantum advantages. This circuit consists of a combination of single-qubit rotations followed by entangling CNOT gates between adjacent qubits. For single i-th qubit rotation, with the angle of $\theta_i^{(l)}$, $\phi_i^{(l)}$, and $\lambda_i^{(l)}$ in the l-th layer can be defined as,

$$R(\theta_i^{(l)}, \phi_i^{(l)}, \lambda_i^{(l)}) = R_z(\phi_i^{(l)}) R_y(\theta_i^{(l)}) R_z(\lambda_i^{(l)}) \tag{13}$$

and,

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right), \quad R_z(\lambda) = \exp\left(-i\frac{\lambda}{2}Z\right) \tag{14}$$

where Y and Z are Pauli matrices.
For entanglement between neighboring qubits of total n qubits, CNOT gates are applied.

$$U_{\text{entangle}}^{(l)} = \prod_{i=1}^{n-1} \text{CNOT}(i, i+1) \tag{15}$$

With the combination of rotation and entanglement, The unitary operation is performed by the quantum circuit SEL.

$$U_{\text{SEL}}^{(l)} = \prod_{i=1}^{n-1} \text{CNOT}(i, i+1) \cdot \left(\bigotimes_{i=1}^{n} R(\theta_i^{(l)}, \phi_i^{(l)}, \lambda_i^{(l)})\right) \tag{16}$$

After applying the SEL to the encoded data, state $|\psi_i'\rangle$ is obtained, which can be presented by the following expression.

$$|\psi_i'\rangle = U_{\text{SEL}}^{(l)} |\psi_{\text{encoded}}\rangle \tag{17}$$

In the quantum layer, the final states $|\psi_{\text{final}}\rangle$ are measured by applying Pauli Z operations.

$$|\psi_{\text{final}}\rangle = Z|\psi_i'\rangle \tag{18}$$

Lastly, the measured classical data are fed into the final output layer for classification.

3.4 Architectures of LHQNN

We design two variations of the LHQNN model- sequential and non-sequential LHQNN based on the consecutive number of qubits. While in sequential LHQNN, the quantum circuit is constructed with a total of n qubits, in non-sequential LHQNN, the qubits are divided into smaller groups in the quantum layer. The diverse quantum architectures in LHQNN can significantly influence the performance, depending on how the data is manipulated within the quantum layers. In the NISQ era, the quantum circuit has several hardware limitations when utilizing quantum resources.

3.4.1 Quantum layers in sequential LHQNN

In sequential LHQNN, the quantum circuit is designed with a total of n qubits that perform simultaneously. Multiple layers of SEL can increase the quantum expressivity to represent more complex quantum states. In this architecture, the complete unitary transformation for L layers can be presented as follows.

$$U_{\text{entangle}}^{(l)} = \prod_{i=1}^{n-1} \text{CNOT}(i, i+1) \tag{19}$$

$$U_{\text{SEL}} = \prod_{l=1}^{L} \left(U_{\text{entangle}}^{(l)} \cdot \left(\bigotimes_{i=1}^{n} R(\theta_i^{(l)}, \phi_i^{(l)}, \lambda_i^{(l)}) \right) \right) \tag{20}$$

Alternatively, the expression can be expanded to provide a more detailed representation.

$$U_{\text{SEL}} = U_{\text{entangle}}^{(L)} \cdot \left(\bigotimes_{i=1}^{n} R(\theta_i^{(L)}, \phi_i^{(L)}, \lambda_i^{(L)}) \right) \cdot \ldots \cdot$$
$$U_{\text{entangle}}^{(1)} \cdot \left(\bigotimes_{i=1}^{n} R(\theta_i^{(1)}, \phi_i^{(1)}, \lambda_i^{(1)}) \right) \tag{21}$$

3.4.2 Quantum layers in non-sequential LHQNN

In non-sequential LHQNN, parallel operations are performed on encoding, circuit, and decoding, where qubits are arranged in groups. These smaller groups of qubits can be processed independently or in parallel, enabling a more dynamic approach to quantum information processing. This approach minimizes the number of consecutive quantum operations, directly reducing the overall length. The detailed non-sequential LHQNN architecture is illustrated in Fig. 3. The non-sequential LHQNN model is designed to minimize the circuit complexity, which is important in reducing noise and decoherence in NISQ computing.
Consider the classical input data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, where n is a multiple of k (i.e., $n = k \times m$), and the data is divided into m groups, each containing k values $(x_{k(i-1)+1}, x_{k(i-1)+2}, \ldots, x_{ki})$ having $i = 1, 2, \ldots, m$. Each of these group is encoded into a k-qubit quantum state using an encoding function E_i:

$$|\psi_i\rangle = E_i(x_{k(i-1)+1}, x_{k(i-1)+2}, \ldots, x_{ki}) \tag{22}$$

resulting in m parallel k-qubit states: $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_m\rangle$.
After encoding, a strongly entangling layer is applied to each group individually. The SEL introduces entanglement within each k-qubit group in parallel, applying CNOT gates

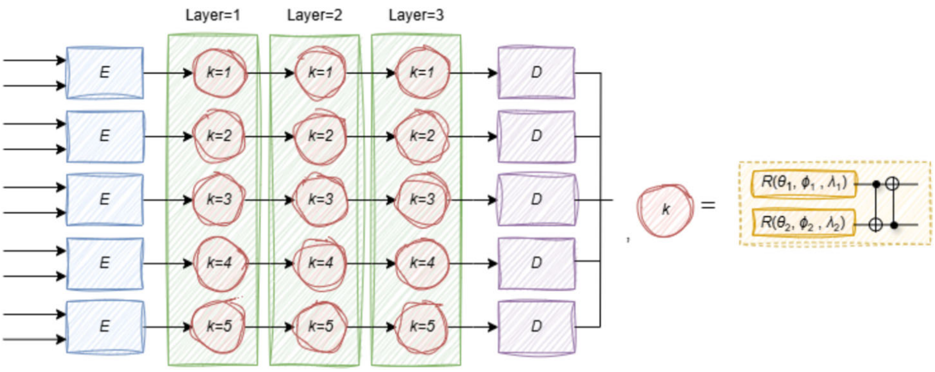

Layer=1
Layer=2
Layer=3
E
E
E
E
E
k=1
k=2
k=3
k=4
k=5
D
D
D
D
D
k
=
$R(\theta_1, \phi_1, \lambda_1)$
$R(\theta_2, \phi_2, \lambda_2)$

Fig. 3 Sample architecture of the non-sequential LHQNN networks. For ten qubits, five groups are created, each having two qubits. These groups can perform encoding, SEL, and decoding independently. k represents each group constructed with three rotations and CNOT of SEL. The decoded outputs are merged together to be fed into the final classification layer

in sequence within each group. For each layer l, the entangling unitary $^{(l)}_{\text{entangle}}$ is defined as:

$$U^{(l)}_{\text{entangle}} = \prod_{j=1}^{k-1} \text{CNOT}(j, j+1) \tag{23}$$

Across all layers, the complete SEL operator U_{SEL} applies entanglement and rotation gates in parallel for each group as follows:

$$U_{\text{SEL}} = \prod_{l=1}^{L} \left(\bigotimes_{i=1}^{m} \left(U^{(l)}_{\text{entangle}} \cdot \left(\bigotimes_{j=1}^{k} R(\theta^{(l)}_{ij}, \phi^{(l)}_{ij}, \lambda^{(l)}_{ij}) \right) \right) \right) \tag{24}$$

where $R(\theta^{(l)}_{ij}, \phi^{(l)}_{ij}, \lambda^{(l)}_{ij})$ represents a rotation applied to qubit j in group i at layer l.

Alternatively, the sequence of operations within each group can be presented explicitly as:

$$U_{\text{SEL}} = \left(\bigotimes_{i=1}^{m} U^{(L)}_{\text{entangle}} \cdot \bigotimes_{j=1}^{k} R(\theta^{(L)}_{ij}, \phi^{(L)}_{ij}, \lambda^{(L)}_{ij}) \right) \cdots$$
$$\times \left(\bigotimes_{i=1}^{m} U^{(1)}_{\text{entangle}} \cdot \bigotimes_{j=1}^{k} R(\theta^{(1)}_{ij}, \phi^{(1)}_{ij}, \lambda^{(1)}_{ij}) \right) \tag{25}$$

After applying SEL to the quantum state $|\psi_i\rangle$, the transformed state $|\psi'_i\rangle = U_{\text{SEL}}(|\psi_i\rangle)$ is obtained, and the Pauli-Z expectation values $\langle Z \rangle_i$ are measured for each qubit. Finally, the measured expectation values are concatenated into a single vector to serve as input to the classical classification layer:

$$|\Phi_{\text{concat}}\rangle = \text{concat}\,(\langle Z \rangle_1, \langle Z \rangle_2, \ldots, \langle Z \rangle_m) \tag{26}$$

where $\langle Z \rangle_i = \langle \psi'_i | Z | \psi'_i \rangle$ represents the Pauli-Z measurement of the i-th qubit, and $|\Phi_{\text{concat}}\rangle$ is the concatenated vector of measurement results that is input into the classical classification layer for further processing.

4 Experimental setup and performance evaluation

This section describes the experimental setup, datasets, and metrics used to evaluate the performance. It includes detailed information about the hardware and software environment, training configurations, and metrics used to validate the results.

4.1 Environment details

The research experiments were conducted on a desktop computer with an Intel(R) Core(TM) i9-14900 and a NVIDIA GeForce RTX 4090 GPU. The software environment was based on Windows 11, with Python 3.10 as the programming language. All models were implemented using PyTorch 1.13.1 and CUDA 11.8 with CuDNN 8.6 for GPU acceleration. Pennylane simulation is used to design and run the quantum sections (Bergholm et al. 2022).

In these experiments, a random seed of 42 is set for reproducibility. Batch sizes 64 and 128 are considered for training and testing, respectively. Datasets were normalized before training (Bishop and Nasrabadi 2006).

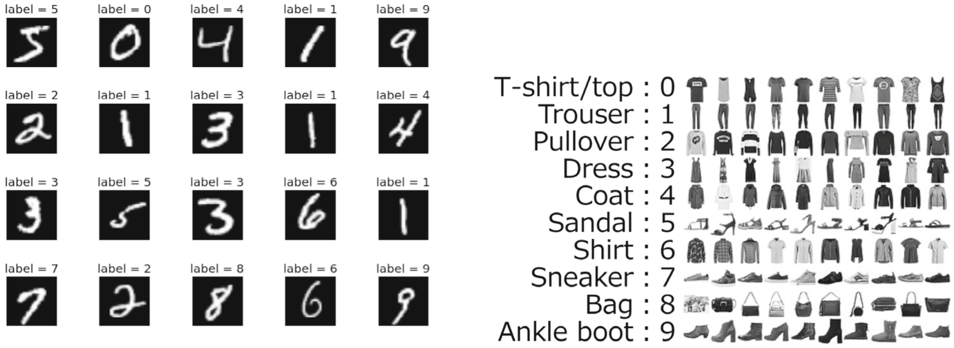
The Adam optimizer is used with a learning rate of 0.001. The loss function employed for training was the cross-entropy loss. To avoid over-fitting, an early stopping method is applied where patience=5 with delta=0.0001.

4.2 Datasets

In this experiment, two publicly available datasets are used to evaluate performance. Description of these datasets are given below:

MNIST dataset: MNIST handwritten digit recognition is a widely use dataset for image classification (Fig. 4a) (Deng 2012). It has ten classes, which represent the digits from 0 to 9. Mainly, there are a total of 70,000 grayscale images of 28×28 dimensions each. Among them, 60,000 images are training data, and the rest are testing data.

Fig. 4 Samples from MNIST and Fashion-MNIST dataset



(a) MNIST dataset

(b) Fashion-MNIST dataset

Fashion-MNIST dataset: Fashion MNIST is a substitute for MNIST data consisting of images of 10 fashion categories (Fig. 4b) (Xiao et al. 2017). It has 60,000 training and 10,000 test images for classification. Each image has a dimension of 28×28, ranging the pixel value from 0–255.

4.3 Evaluation criteria

Accuracy, recall, precision, and f-score are used as metrics to evaluate the performance of this experiment (Bishop and Nasrabadi 2006). These evaluation metrics are described below:

Accuracy: Accuracy quantifies a model's performance by measuring the correct number of predictions. It is the ratio of true identified instances and the total number of instances. Mathematically,

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{27}$$

Precision: Precision measures how often a model can correctly predict positive instances. Mathematically,

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{28}$$

Recall: The recall metric quantifies the percentage of actual positive cases that the model can detect accurately.

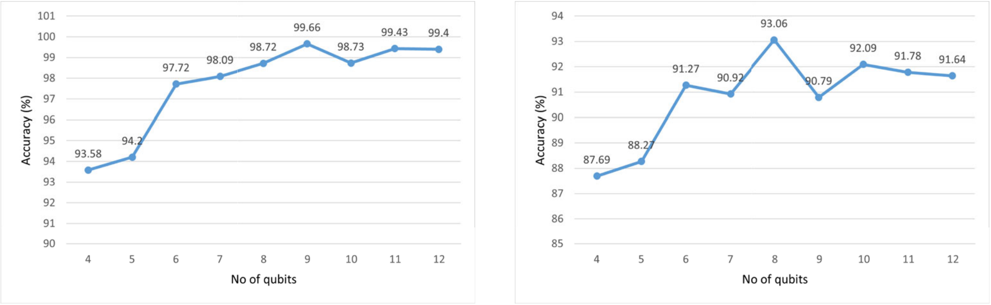
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{29}$$

F1 score: The F1 score represents the harmonic mean of recall and precision.

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{30}$$

5 Results and discussion

We thoroughly evaluate the proposed sequential and non-sequential LHQNN models on MNIST and Fashion-MNIST datasets. To understand the different properties of quantum regarding image classification, diverse architectures of quantum layers are investigated, maintaining minimum classical layers. Two popular encoding techniques- amplitude and angle embedding are implemented separately on each model with 10 qubits to understand their significance in image classification tasks. These experiments employed 5-fold cross-validation to ensure robustness, with the average results being presented for evaluation.



(a) Accuracy curve for different no of qubits with amplitude embedding

(b) Accuracy curve for different no of qubits with angle embedding

Fig. 5 Accuracy curve for different numbers of qubits on Fashion-MNIST data with **a** amplitude and **b** angle embedding

Table 1 Performance metrics of sequential LHQNN models on MNIST data (best results are shown in boldface)

Encoding techniques	No of QL	Accuracy (%)	Precision (%)	Sensitivity (%)	F-score (%)
Amplitude	1	44.98	47.26	43.87	35.59
	2	98.75	98.76	98.74	98.74
	3	99.18	99.17	99.17	99.17
	4	99.20	99.20	99.19	99.19
	5	**99.26**	**99.26**	**99.26**	**99.26**
Angle	1	95.82	95.88	95.79	95.81
	2	95.46	95.52	95.39	95.42
	3	**97.49**	**97.49**	**97.48**	**97.45**
	4	97.30	97.31	97.27	97.28
	5	97.30	97.30	97.28	97.29

5.1 Performance analysis with different qubit numbers

The performance of Sequential LHQNN by varying the number of qubits is presented here. The accuracy achieved with three quantum layers is illustrated in Fig. 5a for amplitude embedding and in Fig. 5b for angle encoding.

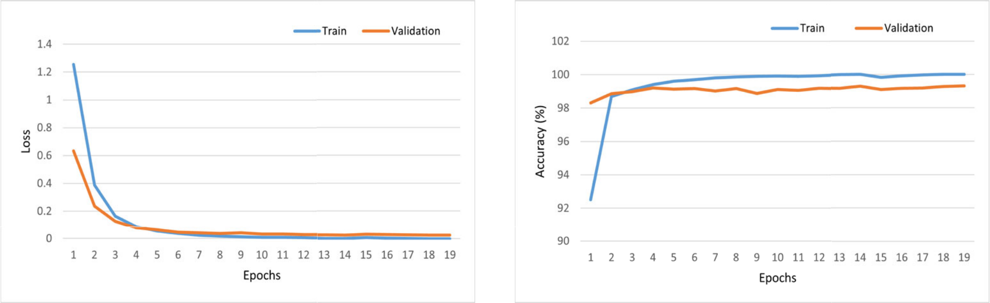
Due to the noisy and error-prone nature of quantum, there is no consistent improvement in accuracy with an increasing number of qubits and fluctuations in the results. However, the accuracy significantly improves with 6 qubits and maintains strong performance beyond that point. While specific qubit numbers cannot be precisely determined, the range of 6 to 10 qubits appears optimal, balancing accuracy and the number of parameters effectively.

5.2 Performance evaluation of sequential LHQNN

The experimental results of sequential LHQNN models on MNIST and Fashion-MNIST datasets are demonstrated here. In this experiment, amplitude and angle encoding are utilized, varying the stacked quantum layer from 1 to 5.

Table 1 represent the detailed experimental results of the sequential LHQNN model on the MNIST dataset. The highest accuracy achieved by this model is 99.26%, using amplitude embedding with 5 quantum layers. With angle embedding, the best accuracy achieved is 97.49% with 3 layers, which is significantly lower compared to amplitude encoding. It is prominent that the accuracy consistently improves as the number of quantum layers increases. For both embedding techniques, the accuracy increases considerably up to 3 quantum layers. However, after the third layer, there is marginal to no improvement in accuracy, indicating that additional layers do not significantly enhance the model's performance.

To evaluate the generalization and robustness of sequential LHQNN, loss, and accuracy curves are depicted in Fig. 6a and b respectively. The best results among the 5-fold cross-validation are considered to demonstrate these figures. The training and validation loss consistently decreases over time, reaching an optimal point after 19 epochs and five early



(a) Training and validation loss curve

(b) Training and validation accuracy curve

Fig. 6 Performance plot for sequential LHQNN model on MNIST data with amplitude embedding. Graph **a** shows training and validation loss, and graph **b** illustrates training and testing accuracy

Table 2 Performance metrics of sequential LHQNN models on Fashion-MNIST data (best results are shown in boldface)

Encoding techniques	No of QL	Accuracy (%)	Precision (%)	Sensitivity (%)	F-score (%)
Amplitude	1	86.76	86.52	86.76	86.06
	2	90.64	90.63	90.64	90.62
	3	91.36	91.34	91.36	91.29
	4	**91.54**	**91.54**	**91.54**	**91.51**
	5	**91.54**	**91.54**	**91.54**	**91.51**
Angle	1	84.40	84.82	84.40	84.45
	2	88.70	88.94	88.76	88.78
	3	88.64	88.83	88.64	88.68
	4	89.04	89.14	89.04	89.05
	5	**89.51**	**89.54**	**89.51**	**89.48**

stops, with final values of 0.0005 for training loss and 0.0255 for validation loss. While the training accuracy has a sharp edge, reaching 100%, the validation accuracy shows a steady improvement, reaching 99.32% in the accuracy curve.

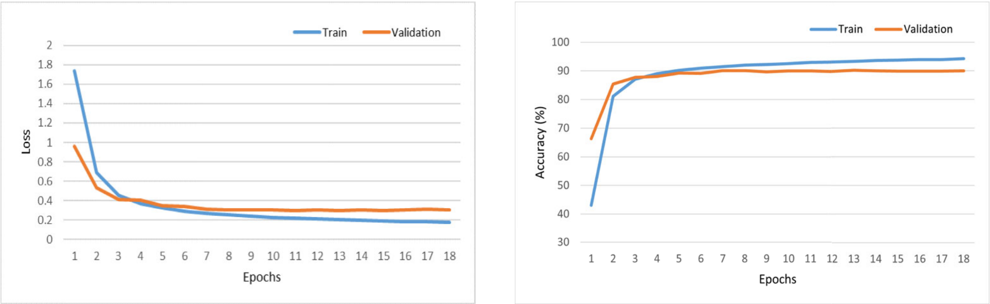
Table 2 shows the results of sequential LHQNN on the Fashion-MNIST dataset, which exhibit a trend similar to that observed with MNIST. The results indicate that performance improves with the addition of more quantum layers, which highlights the advantage of deeper architectures. Using amplitude embedding, this model achieved the best accuracy of 91.54% with 5 quantum layers, outperforming angle embedding, which achieved an accuracy of 89.51% with the same number of layers.

Fig. 7a illustrates the loss curves for both training and validation data, showing a steady decline as the model trains. The training loss decreased to 0.174, while the validation loss stabilized at 0.3034 after 18 epochs. Figure 7b depicts the corresponding accuracy curves with the training accuracy reaching 94.25% and the validation accuracy following at 89.93%. The final epoch represents the optimal performance point, demonstrating the model's ability to reduce loss and achieve high accuracy.

5.3 Performance evaluation of non-sequential LHQNN

This section presents the results of non-sequential LHQNN models using amplitude and angle encoding, with architectures varying from 1 to 5 stacked quantum layers.

The results of non-sequential LHQNN architecture on the MNIST dataset are presented in Table 3. There is a minimal change in performance as the number of quantum layers changes. The best accuracy with amplitude embedding is 98.99% using 4 quantum layers, whereas angle embedding shows an accuracy of 98.11% with 5 quantum layers. Amplitude embedding consistently outperforms angle embedding



(a) Training and validation loss curve

(b) Training and validation accuracy curve

Fig. 7 Performance plot for sequential LHQNN model on Fashion-MNIST data with angle embedding. Graph **a** shows training and validation loss, and graph **b** illustrates training and testing accuracy

Table 3 Performance metrics of non-sequential LHQNN model on MNIST data (best results are shown in boldface)

Encoding techniques	No of QL	Accuracy (%)	Precision (%)	Sensitivity (%)	F-score (%)
Amplitude	1	10.03	1.00	1.00	1.00
	2	98.82	98.81	98.82	98.81
	3	98.95	98.94	98.95	98.94
	4	**98.99**	**99.00**	**98.98**	**98.99**
	5	98.61	98.61	98.61	98.60
Angle	1	98.03	98.02	98.01	98.01
	2	98.00	98.01	97.97	97.99
	3	98.08	98.07	98.07	98.06
	4	98.08	98.07	98.06	98.06
	5	**98.11**	**98.11**	**98.09**	**98.10**

in all configurations. However, with just 1 layer, it encounters issues that prevent the model from effectively learning and training on this dataset.

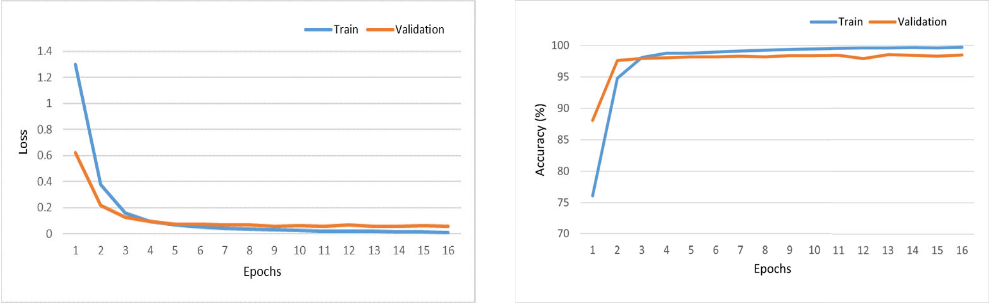
In non-sequential LHQNN architectures, all the models show consistency to converge and achieve the best accuracy. Using angle embedding, the highest training accuracy is 99.72%, while the validation accuracy reached a maximum of 98.48%, which is shown in Fig. 8b. Figure 8a demonstrates that after 16 epochs, the training and validation losses are minimum, having values of 0.0117 and 0.0558, respectively.

In non-sequential LHQNN with FMNIST data, the detailed evaluation is presented in Table 4. For amplitude embedding, these are fluctuations with the changing number of quantum layers having the maximum result of 90.97% with 2 quantum layers. However, angle embedding exhibits a marginally lower performance compared to amplitude embedding, achieving a best accuracy of 89.82% with 5 quantum layers.

The performance analysis curve for non-sequential LHQNN with amplitude embedding is illustrated in Fig. 9a and b. The validation loss reached a minimum of 0.2592 after eight epochs. The accuracy also shows a continuous improvement, achieving an accuracy of 91.43%. Similarly, the training loss steadily decreased to 0.147, while the training accuracy increased to 94.85%, further indicating effective model training.

5.4 Comparison with traditional and quantum approaches

A detailed comparative analysis of classical CNN and proposed LHQNN models is presented in this subsection. The graphical representation is illustrated in Fig. 10a for MNIST dataset and Fig. 10b for Fashion-MNIST dataset. The best accuracy across all cross-validation folds is presented for comparison.



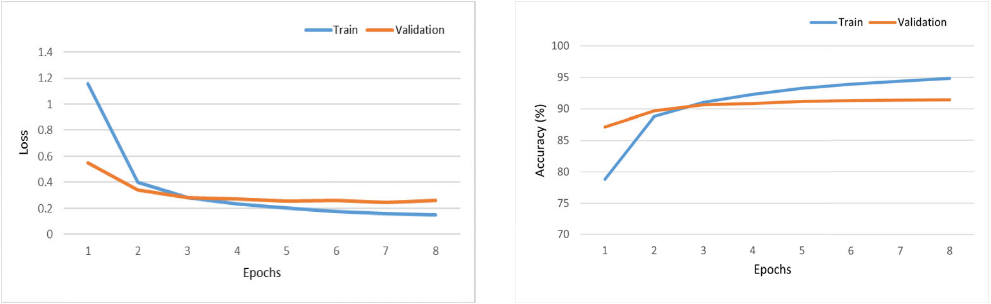
(a) Training and validation loss curve

(b) Training and validation accuracy curve

Fig. 8 Performance plot for non-sequential LHQNN model on MNIST data with angle embedding. Graph **a** illustrates training and validation loss, and graph **b** depicts training and testing accuracy

Table 4 Performance metrics of non-sequential LHQNN models on Fashion-MNIST data (best results are shown in boldface)

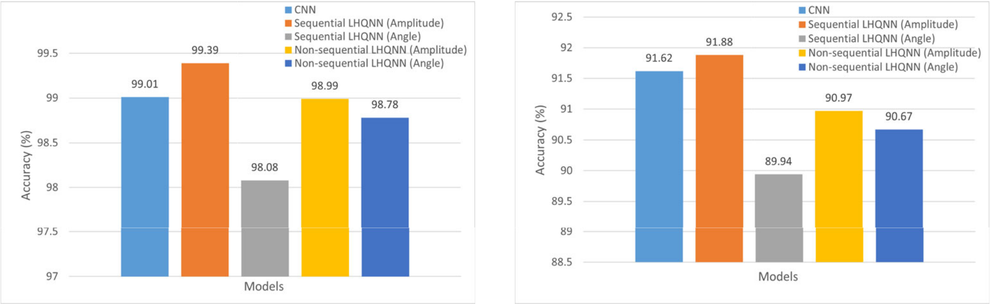
Encoding techniques	No of QL	Accuracy (%)	Precision (%)	Sensitivity (%)	F-score (%)
Amplitude	1	90.56	90.61	90.56	90.55
	2	**90.97**	**90.93**	**90.97**	**90.93**
	3	90.38	90.34	90.38	90.28
	4	90.85	91.01	90.85	90.89
	5	90.87	90.91	90.87	90.79
Angle	1	89.61	89.61	89.61	89.58
	2	89.80	89.76	89.80	89.75
	3	89.75	89.75	89.75	89.72
	4	89.61	89.54	89.61	89.53
	5	**89.82**	**89.91**	**89.82**	**89.84**



(a) Training and validation loss curve

(b) Training and validation accuracy curve

Fig. 9 Performance plot for non-sequential LHQNN model on Fashion-MNIST data with amplitude embedding. Graph **a** depicts training and validation loss, and graph **b** shows training and testing accuracy



(a) Accuracy comparison of different models on MNIST data

(b) Accuracy comparison of different models on Fashion-MNIST data

Fig. 10 Performance comparison of traditional CNN with proposed LHQNN models on **a** MNIST and **b** Fashion-MNIST data

Table 5 Performance comparison of the proposed approach with existing quantum approaches on MNIST data (best results are shown in boldface)

Study	Model	Accuracy(%)
Li et al. (2020)	QDCNN	98.97
Zeng et al. (2022)	HQNN	89.06
Riaz et al. (2023)	Classical NN	92.00
	QuanvNN	93.00
	NNQE	93.80
Gong et al. (2024)	CNN	96.04
	QCNN(VQC)	98.40
Hassan et al. (2024)	QCNN	97.50
	ResNet(50)	98.90
	MQCNN	99.60
Proposed (LHQNN)	**LHQNN**	**99.92**

For both datasets, sequential LHQNN with amplitude embedding achieved the best result, outperforming traditional CNN. The non-sequential LHQNN with amplitude embedding shows slightly lower performance than the classical CNN. With angle embedding, the non-sequential LHQNN outperforms the sequential LHQNN significantly.

Table 5 presents a comparative analysis of existing quantum approaches with LHQNN. The proposed model was evaluated over multiple independent runs on the MNIST dataset, with the best result selected. The results demonstrate that our proposed LHQNN outperforms existing quantum techniques for image classification.

5.5 Analysis of parameter numbers

The number of parameters plays a crucial role in the model's complexity and performance. In this section, we analyze the parameter count for LHQNN models with amplitude and angle encoding having one quantum layer.

The total number of parameters using amplitude embedding is depicted in Fig. 11a. The graph shows an exponential growth of parameters with the increase in qubit numbers. On the other hand, in Fig. 11b, LHQNN with angle encoding demonstrates a linear upward trend varying the number of qubits. It can be concluded that amplitude embedding contains a more complex architecture than angle embedding for image classification due to its capability to handle a high number of features.

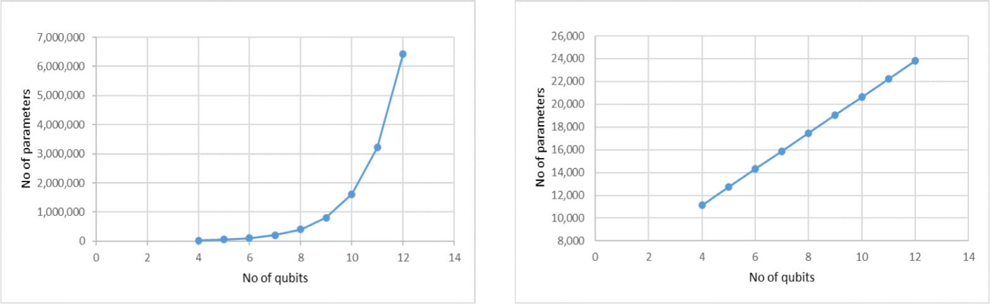
Table 6 Number of parameters of LHQNN with amplitude and angle encoding varying quantum layers

LHQNN encoding	QL 1	QL 2	QL 3	QL 4	QL 5
Amplitude	1,611,596	1,611,626	1,611,656	1,611,686	1,611,716
Angle	20,630	20,660	20,690	20,720	20,750

Table 6 presents the total number of parameters for LHQNN architectures with different numbers of quantum layers. Here, both amplitude and angle encoding are implemented using 10 qubits. Based on the number of qubits and circuits, each layer adds a few more parameters to the quantum layer. Specifically, in this architecture, the SEL circuit includes 3 rotations, and with 10 qubits, this results in a total parameter increase of 30 per layer. However, the proposed LHQNN achieves superior performance while using significantly fewer parameters than CNN, which consists of 1,743,946 parameters.

5.6 Discussion

This study explores the performance of LHQNNs on benchmark datasets such as MNIST and Fashion-MNIST. Compared to angle embedding, amplitude embedding yields a better result across all configurations of quantum layers. For sequential LHQNN, accuracy shows a consistent improvement with an increasing number of quantum layers up to a specific limit. However, non-sequential LHQNN does not have a significant impact on the change of the quantum layer. Due to the smaller circuit sizes in non-sequential architectures, angle embedding demonstrates lower errors and



(a) Parameter count vs number of qubits (b) Parameter count vs number of qubits

Fig. 11 Total number of parameters on LHQNN varying qubit numbers with **a** amplitude and **b** angle embedding

performs more effectively than in sequential LHQNN. In contrast, with amplitude embedding, feature space is significantly reduced in non-sequential LHQNN, which limits the model's ability to perform effectively. Among these architectures, sequential LHQNN yields the best results, outperforming classical approaches. It can be concluded that although a deeper quantum circuit adds a few more parameters to the model, it significantly enhances the capacity to capture complex patterns within the data.

6 Conclusion

In this research, we propose layered hybrid neural network architectures using sequential and non-sequential approaches for image classification. The integration of quantum layers with CNN architecture leverages the quantum advantages in the models. Our proposed sequential LHQNN with amplitude embedding achieves the highest accuracy compared to other models and outperforms the traditional CNN.

Moreover, our research focuses on the fundamentals of quantum layer architecture, including the number of qubits, embedding techniques, and circuit design, specifically in image classification problems. Our results indicate that amplitude embedding outperforms angle embedding on both the MNIST and Fashion-MNIST datasets since it has the capacity to handle a large number of features. For quantum circuit design, non-sequential architecture provides better performance with angle embedding primarily due to its ability to minimize errors. In contrast, amplitude embedding does not perform as well in non-sequential architectures, as no specific feature selection techniques are employed to handle the prominent features. It is also observed that to capture the patterns in the data effectively, the number of qubits must exceed 6. Finally, the performance and complexity of stacked quantum layers are analyzed with different embedding techniques. Evaluating how parameter counts change with different configurations provides valuable insights for designing hybrid models and optimizing resource utilization.

In the NISQ era, the limitations of quantum hardware impose significant constraints on the size and complexity of quantum layers. Despite these challenges, our proposed LHQNN models demonstrate the potential to leverage the strengths of classical and quantum paradigms effectively. The architectures we present exhibit scalability with respect to the number of quantum layers, opening avenues for further research to design quantum circuits. These findings lay the foundation for exploring more efficient quantum-classical hybrid models, which could unlock new possibilities in quantum machine learning.

Acknowledgements Monika Kabir was funded by a Murdoch University PhD scholarship. Ferdous Sohel was partly funded by a strategic small grant from the School of Information Technology, Murdoch University.

Author contribution M.K.: conceptualization, methodology, software, validation, formal analysis, data curation, investigation, writing—original draft, writing—review and editing. Mo.K.: methodology, writing—review and editing, supervision. H.L.: methodology, writing—review and editing, supervision. F.S.: conceptualization, methodology, writing—original draft, writing—review and editing, supervision, project administration. All authors reviewed the manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability No new dataset was generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

Agarap AF (2019) Deep learning using rectified linear units (ReLU). Available from: arXiv:1803.08375

Amin MH, Andriyash E, Rolfe J, Kulchytskyy B, Melko R (2018) Quantum Boltzmann machine. Phys Rev X 8(2). https://doi.org/10.1103/physrevx.8.021050

Bergholm V, Izaac J, Schuld M, Gogolin C, Ahmed S, Ajith V et al (2022) PennyLane: automatic differentiation of hybrid quantum-classical computations. Available from: arXiv:1811.04968

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature 549(7671):195–202. https://doi.org/10.1038/nature23474

Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning. vol. 4. Springer

Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15(12):1273–1278. https://doi.org/10.1038/s41567-019-0648-8

Deng L (2012) The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Proc Mag 29(6):141–142. https://doi.org/10.1109/MSP.2012.2211477

Gong LH, Pei JJ, Zhang TF, Zhou NR (2024) Quantum convolutional neural network based on variational quantum circuits. Opt Commun 550:129993. https://doi.org/10.1016/j.optcom.2023.129993

Griffiths DJ, Schroeter DF (2018) Introduction to quantum mechanics. 3rd ed. Cambridge University Press

Hassan E, Hossain MS, Saber A, Elmougy S, Ghoneim A, Muhammad G (2024) A quantum convolutional network and ResNet (50)-based classification architecture for the MNIST medical dataset. Biomed Signal Process Control 87:105560. https://doi.org/10.1016/j.bspc.2023.105560

Henderson MP, Shakya S, Pradhan S, Cook T (2019) Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Mach Intell 2

Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJ, Bottou L, Weinberger KQ (eds) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc.. Available from: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Li Y, Zhou RG, Xu R, Luo J, Hu W (2020) A quantum deep convolutional neural network for image recognition. Quantum Sci Technol 5(4):044003. https://doi.org/10.1088/2058-9565/ab9f93

Liao Y, Hsieh MH, Ferrie C (2024) Quantum optimization for training quantum neural networks. Quantum Mach Intell 6(33). https://doi.org/10.1007/s42484-024-00169-w

Ling YQ, Zhang JH, Zhang LH, Li YR, Huang HL (2024) Image classification using hybrid classical-quantum neutral networks. Int J Theor Phys 63(125). https://doi.org/10.1007/s10773-024-05669-w

Lloyd S, Mohseni M, Rebentrost P (2014) Quantum principal component analysis. Nat Phys 10(9):631–633. https://doi.org/10.1038/nphys3029

Lloyd S, Schuld M, Ijaz A, Izaac J, Killoran N (2020) Quantum embeddings for machine learning. Available from: arXiv:2001.03622

Mari A, Bromley TR, Izaac J, Schuld M, Killoran N (2020) Transfer learning in hybrid classical-quantum neural networks. Quantum 4:340. https://doi.org/10.22331/q-2020-10-09-340

Mermin ND (2007) Quantum computer science: an introduction. Cambridge University Press

Nielsen MA, Chuang IL (2010) In: introduction and overview. Cambridge University Press. p. 1–59

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

Rebentrost P, Mohseni M, Lloyd S (2013) Quantum support vector machine for big feature and big data classification. Physical Rev Lett 113(13):130503

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(5). https://doi.org/10.3390/s23052753

Sagingalieva A, Kordzanganeh M, Kurkin A, Melnikov A, Kuhmistrov D, Perelshtein M et al (2025) Hybrid quantum ResNet for car classification and its hyperparameter optimization. Quantum Mach Intell 5(2). https://doi.org/10.1007/s42484-023-00123-2

Schuld M, Bocharov A, Svore KM, Wiebe N (2020) Circuit-centric quantum classifiers. Phys Rev A 101(3). https://doi.org/10.1103/physreva.101.032308

Schuld M, Petruccione F (2018) Supervised Learning with quantum computers. Quantum Science and Technology. Springer International Publishing. Available from: https://books.google.com.au/books?id=1zpsDwAAQBAJ

Senokosov A, Sedykh A, Sagingalieva A, Kyriacou B, Melnikov A (2024) Quantum machine learning for image classification. Mach-Learn Sci Technol 3 5(1):015040. https://doi.org/10.1088/2632-2153/ad2aef

Team IQ (2024) Qiskit documentation: plot_bloch_vector. Available from: https://docs.quantum.ibm.com/api/qiskit/qiskit.visualization.plot_bloch_vector. Accessed 13 Dec 2024

Vu TH, Le LH, Pham TB (2024) Exploring the features of quanvolutional neural networks for improved image classification. Quantum Mach Intell 6(29). https://doi.org/10.1007/s42484-024-00166-z

Wang A, Hu J, Zhang S, Li L (2024) Shallow hybrid quantum-classical convolutional neural network model for image classification. Quantum Inf Process 23(17). https://doi.org/10.1007/s11128-023-04217-5

Warrier SR, Reddy DSH, Bada S, Achampeta R, Uppapalli S, Dontabhaktuni J (2024) On-board classification of underwater images using hybrid classical-quantum CNN-based method. Quantum Mach Intell 6(70). https://doi.org/10.1007/s42484-024-00206-8

Xiao H, Rasul K, Vollgraf R (2017) Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. Available from: arXiv:1708.07747

Zeng Y, Wang H, He J, Huang Q, Chang S (2022) A multi-classification hybrid quantum neural network using an all-qubit multi-observable measurement strategy. Entropy 24(3). https://doi.org/10.3390/e24030394

Zhang S, Wang A, Li L (2024) Quantum-convolution-based hybrid neural network model for arrhythmia detection. Quantum Mach Intell 6(75). https://doi.org/10.1007/s42484-024-00207-7