# Quantum-Assisted Hierarchical Fuzzy Neural Network for Image Classification

Shengyao Wu, Runze Li, Yanqi Song, Sujuan Qin, Qiaoyan Wen, and Fei Gao

***Abstract*—Deep learning is a powerful technique for data-driven learning in the era of Big Data. However, most deep learning models are deterministic models that ignore the uncertainty of data. Fuzzy neural networks are proposed to tackle this type of problem. In this article, we proposed a novel quantum assisted hierarchical fuzzy neural network (QA-HFNN). Different from classical fuzzy neural networks, QA-HFNN uses quantum neural networks (QNNs) to learn fuzzy membership functions. The model is a multifeature fusion learning algorithm with a parallel structural design that integrates quantum and classical neural networks. The classical network is used to capture high-dimensional neural features, the QNNs are designed to capture fuzzy logic features of the data, then, the two features are fused to form the final features to be classified. The experiment is performed on a classical computer, and the quantum circuit is built through a simulated quantum environment. The results indicate that the accuracy of QA-HFNN can equal to or even surpass classical methods in image classification tasks. The quantum circuit utilizes only a single qubit which is easy to implement. In addition, the fidelity of quantum circuit in a quantum noise environment is assessed, demonstrating that QA-HFNN has strong robustness. The time and computational complexity of QNNs was analyzed, further proving the effectiveness of the model.**

***Index Terms*—Fuzzy member function, fuzzy neural network, image classification, quantum neural network (QNN).**

## I. Introduction

In THE era of Big Data, a diverse and vast amount of data are generated. One concern is that the data contain unpredictable uncertainties [1]. These uncertainties may arise from a variety of factors, including data ambiguity, incompleteness, noise, and redundancy. In the actual data analysis process, uncertainty may make it difficult to capture the real patterns in the data, which brings great challenges to data classification tasks [2].

Most of the existing learning models are deterministic algorithms, which face difficulties when dealing with uncertain data. To solve this problem, fuzzy learning has been proposed and applied in image processing [3], financial analysis [4], and control systems [5]. Fuzzy systems can automatically learn fuzzy membership functions from a large amount of training data and derive fuzzy rules accordingly. Following the determination of fuzzy membership functions, fuzzy logic values for the data can be obtained. Subsequently, these fuzzy logic values are effectively defuzzified through linear combination to form features for the final classification task. Compared with traditional deterministic representation, the automated learning process of fuzzy logic representation enables it to effectively represent the uncertainty data.

Fuzzy neural networks often integrate deep learning methods, combining deep neural networks (DNNs) and fuzzy logic methods to solve various real-world problems. Common fuzzy network structures include sequential [6], parallel [7], and hybrid [8]. In sequential structure networks, data are processed sequentially within the fuzzy system and DNN. Parallel structure networks employ feature fusion techniques to derive information from both fuzzy systems and DNN. Hybrid structure networks first convert input vectors into fuzzy values through fuzzy inference, then, process them through multiple layers of DNN, and finally, defuzzify the fuzzy values into crisp output. Fuzzy neural networks with different structures have shown their advantages in practical tasks. However, current methodologies still suffer to effectively represent uncertain data. Quantum neural networks (QNNs) offer promising advantages in expressivity over classical neural networks [9]. Therefore, QNNs have the potential to better extract fuzzy features of data. With the help of QNNs, the fuzzy features of uncertain data can be better learned, thereby, enhancing the performance of fuzzy neural networks.

In this article, we introduce a novel quantum assisted hierarchical fuzzy neural network (QA-HFNN) which uses QNNs to learn membership functions in fuzzy neural network. The proposed model simultaneously extracts the high-dimensional neural features of classical DNNs and the fuzzy features processed by QNNs. The features extracted by the two models are fused to produce the final data representation for data classification. We explore the possibility of learning fuzzy logic features using QNNs. In details, learning fuzzy membership functions using QNNs, and extracting the learned fuzzy features using fuzzy rules. The proposed model is suitable for difficult tasks containing data ambiguity and noise. The performance of QA-HFNN is verified on different image datasets. Experimental results indicate that QA-HFNN can outperform existing classical methods in accuracy.

Received 8 June 2024; revised 20 July 2024; accepted 22 July 2024. Date of publication 19 August 2024; date of current version 3 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62372048, Grant 62371069, and Grant 62272056, in part by the Beijing Natural Science Foundation under Grant 4222031 and in part by the 111 Project under Grant B21049. Recommended by Associate Editor W. Wei. *(Corresponding author: Fei Gao.)*

The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: wusybn@bupt.edu.cn; rzli@bupt.edu.cn; songyanqi_417@163.com; qsujuan@bupt.edu.cn; wqy@bupt.edu.cn; gaof@bupt.edu.cn).

Our code is available at https://github.com/wodaka/QA-HFNNhttps://github.com/wodaka/QA-HFNN.

Digital Object Identifier 10.1109/TFUZZ.2024.3435792

Note that a study has been proposed that combines QNN, classical networks and fuzzy logic to introduce quantum fuzzy neural network (QFNN) [10]. The model is proposed for sentiment and sarcasm detection. QFNN is a multitask learning algorithm with Sqe2Seq structure, and the model use QNN in defuzzifier to obtain predictions. The method and objective of QFNN are both different from our model.

We list the main contributions as follows.

1) A novel network QA-HFNN is proposed. QNNs are used to learn fuzzy membership functions in the fuzzy neural network. Multimodal learning is applied to fuse the fuzzy logic representation obtained by the QNNs and the neural representation obtained by the classical network to increase accuracy in image classification tasks.
2) Circuit structures of QNNs are designed to extract fuzzy logic representation from data. The proposed circuit structure utilizes only a single qubit and constructed by a small number of single-qubit rotation gates, making it easy to implement.
3) Experiments to validate the performance of QA-HFNN have been conducted on different image datasets. Compared with existing classic methods, it achieves better accuracy. The experiments simulated the performance of QNNs in noisy environments, the results demonstrate the strong robustness of QA-HFNN. The complexity of QNNs is analyzed, further demonstrating the effectiveness of the model.

The rest of this article is organized as follows. Section II reviews related work, including image classification, quantum machine learning (QML), and quantum fuzzy logic. Section III details the main structure and algorithm process of QA-HFNN. In Section IV, experimental setup is described, and the experimental results are presented to illustrate the capability of the model. Finally, Section V concludes this article.

## II. Related Work

### A. Image Classification

Image classification holds great significance in computer vision and machine learning. Early image classification approaches relied on handcrafted feature extraction algorithms, such as, SIFT and HOG [11], and then used support vector machines (SVMs) or k nearest neighbors (KNNs) for classification. This type of method only works well for simple images and small-scale datasets. As image scenes become complex and data size increases, this type of method is no longer applicable.

The emergence of convolutional neural network (CNN) has made a major breakthrough in image classification tasks. CNN extracts the feature representation of the image through convolution and pooling operations. Subsequently, a large number of CNN-based models were proposed to improve the accuracy of image classification. In 1998, LeCun et al. [12] proposed LeNet. Its core idea is to gradually extract the image features by employing convolution and pooling operations, and then, classify the extracted features to different categories through a fully connected layer. Krizhevsky et al. [13] proposed AlexNet in 2012, which uses more convolutional layers and parameters, and uses technologies including ReLU activation functions, data enhancement and dropout, which extremely improves the performance and generalization capabilities of the network. In 2014, Simonyan [14] proposed VGGNet. Its core idea is to gradually extract the features from the image by stacking multiple small-sized convolution kernels and pooling layers, with a simple and unified network structure. It uses a parameter sharing strategy to decrease the parameter size and improve training efficiency and generalization ability. In 2015, He et al. [15] proposed ResNet, which introduced the concept of residual learning, solved the problem of gradient disappearance and degradation in deep network training. In 2023, Wang et al. [16] proposed InternImage, whose core operator is deformable convolution. Under the similar parameters and computational costs, experimental results show that the classification performance of InternImage on ImageNet is comparable or even better than state-of-the-art CNN-based models.

In addition, image classification algorithms based on attention mechanisms have also appeared [17]. By learning the key areas in the image, the degree of attention to important areas is increased, thereby, improving the classification performance. In 2021, Liu et al. [18] proposed a Swin transformer, a method designed to handle visual tasks by using a shifted window to calculate hierarchical transformer representations. In 2023, Roy et al. [19] introduced a new multimodal fusion transformer network, which uses a multihead cross-patch attention to integrate multimodal features for hyper spectral image classification.

Despite the diversity and complexity of image classification methods, image classification remains a research hotspot in the field. In the Big Data era, the amount and complexity of image data are constantly increasing, necessitating further investigation into methods to address the challenges. However, the combination of new technologies, such as, fuzzy logic and quantum computing, to handle tasks in this scenario has not been fully explored.

### B. Classical Fuzzy Logic

In 1965, Zadeh [20] introduced the idea of fuzzy logic, which showed great advantages in solving practical problems. There are many models based on classic fuzzy logic. Deng et al. [7] constructed an FDNN which used fuzzy representation of data and neural network representation for information derivation, and achieved good performance in scene image classification and brain MRI segmentation tasks. In 2019, Sarabakha et al. [21] used antecedent fuzzification to learn the control of nonlinear systems. The proposed fuzzy neural network is a sequence structure. In 2021, Połap [22] proposed fuzzy consensus to solve the problem of how to make decisions when not all the information is available. The proposed method shows advantages for medical purposes compared to state-of-the-art algorithms. In 2022, Zhang et al. [23] proposed using the federated fuzzy neural network (FedFNN) learned by evolutionary rules to solve nonIID and uncertain data. The proposed evolutionary rule learning method improves the generalization and personalization of the network, and the experimental results

express the superiority of FedFNN. Wang et al. [24] proposed a novel method that can extract the input image into the fuzzy domain. Compared with existing super-resolution methods, the method performs well in high resolution medical image reconstruction, and reduces model parameters and analysis time. In 2023, Yazdinejad et al. [25] proposed an OFDL model. OFDL employs NSGA-II to obtain the Pareto optimal solutions for multiobjective optimization based on the objective functions, thereby, optimizing fuzzy membership functions.

In general, fuzzy logic has shown its unique advantages in industrial control, image processing, and other fields, and is especially suitable for tasks with uncertainty in data. Quantum fuzzy logic, which combines quantum computing and fuzzy logic, is an emerging field, and its research on image classification tasks is still in its nascent stages.

### C. Quantum Machine Learning

Quantum machine learning (QML) [26] is a promising field that explores the advantages of quantum computing in the filed of machine learning. QML includes quantum support vector machines (QSVM), quantum linear regression, quantum clustering and QNNs, etc. QML uses the parallel characteristics of quantum computing to accelerate certain classical algorithms. In addition, QNNs use the quantum properties of entanglement and superposition to allow data to be better represented [27].

In 1995, Kak et al. [28] proposed quantum neural computing, laying a theoretical foundation for QNNs. In 2003, Anguita et al. [29] proposed a QSVM algorithm, which uses Grover search to accelerate the training process of the SVM data classification algorithm. In 2012, Wiebe et al. [30] introduced a quantum linear regression algorithm. This algorithm demonstrates exponential acceleration to its classical counterpart when the data matrix is sparse and has a very low condition number. In 2014, Lloyd et al. [31] proposed a distance-based classification algorithm, which can achieve exponential acceleration. In 2019, Kerenidis et al. [32] proposed the $q$-means clustering algorithm. Compared to the classic $k$-means algorithm, the $q$-means algorithm provides exponential speedups. However, the above algorithm needs a large-scale quantum computer that is tolerant to noise. In the context of noisy intermediate-scale quantum computing, variational quantum algorithms which uses parameterized quantum circuits (PQCs) to evaluate loss functions and optimize parameters using classical optimizer [33] have been proposed and widely studied [34]. In 2018, Mitarai et al. [35] proposed a QML framework based on PQCs, called quantum circuit learning (QCL), and proposed the idea of using unitary operator gates to approximate nonlinear functions. In 2018, Lloyd [36] first proposed quantum generative adversarial network and analyzed the potential quantum advantages from a theoretical perspective. In 2019, Cong et al. [37] proposed a quantum convolutional neural network (QCNN) based on variational circuits, which constructed a convolution layer and a pooling layer with parameterized quantum gates, and proved it on two types of problems as follows: 1) quantum phase identification and 2) quantum error correction. In 2020, Schuld et al. [38] proposed a low-depth PQC as a binary classifier. In 2022, Qu et al. [39] proposed quantum graph convolutional neural network, which was utilized for traffic congestion prediction. In 2023, Skolik et al. [40] investigated the performance of variational quantum reinforcement learning under shot noise, coherent, and incoherent error noise sources. They also proposed methods to reduce the measurements required for training Q-learning agent. In 2024, Song et al. [41] proposed a quantum federated learning model for classical clients, addressing the issue of limited quantum computing resources for classical clients. In the same year, they proposed a computationally resource-efficient QCNN model [42], demonstrating a significant reduction in computing resources compared to the classical CNN model, while achieving high accuracy on multiclass classification tasks.

In summary, QML models have been involved in all aspects of different research fields. QNNs have shown classification advantages over classical neural networks in specific datasets. In addition, because of the entanglement and superposition properties, models trained with QNNs have better performance and generalization capabilities.

### D. Quantum Fuzzy Logic

Quantum fuzzy logic is a new theory that combines quantum theory and fuzzy logic. In 2007, Menichenko et al. [43] proposed that quantum logic can also be regarded as a kind of fuzzy logic. This lays the theoretical foundation for quantum fuzzy logic. In 2023, Tiwari et al. [10] proposed a quantum fuzzy logic and applied it to multimodal sentiment and sarcasm detection. In 2024, Qu et al. [44] proposed a quantum fuzzy federated learning algorithm named QFFL, which utilizes quantum fuzzy neural networks on local nodes and quantum federated inference on the global model. This approach achieved faster training speed and higher training accuracy on the COVID-19 and MNIST datasets. Currently, there are few studies on quantum fuzzy logic, which is a field worthy of further exploration.

## III. Proposed Method

The QA-HFNN is composed of four main network parts as follows: 1) quantum fuzzy logic representation, 2) DNN representation, 3) fusion layer, and 4) classifier layer. The design of QA-HFNN is depicted in Fig. 1. The input data will be input into the quantum fuzzy logic representation network and the DNN to generate fuzzy logic representation and neural network representation of the data, respectively. Then, the features of the two parts are fused through the fusion layer. Finally, the fused features are input to a classifier layer, which classifies the data into different categories.

*Quantum fuzzy logic representation:* The input data will be processed by membership functions, and each input variable is assigned a membership degree for different fuzzy sets. Generally speaking, fuzzy membership functions calculate the extent to which an input node belongs to a specific fuzzy set. For the $i$th fuzzy membership function, this is a mapping $u_i : R \to [0, 1]$. In previous work [45], the membership function is a Gaussian function with a mean of $\mu$ and a variance of $\sigma^2$

$$u_i(x_k) = e^{-(x_k-\mu_i)^2/{\sigma_i}^2} \tag{1}$$

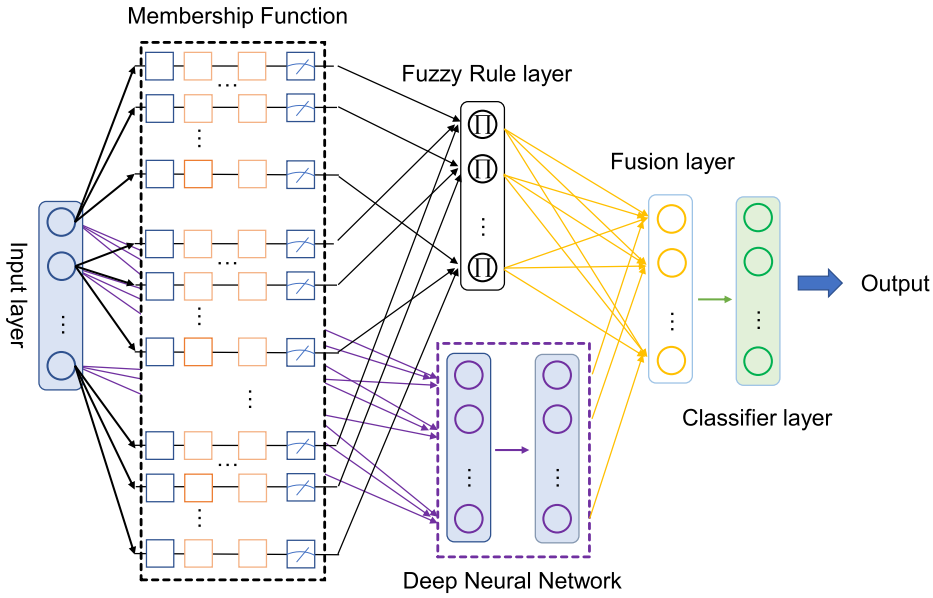


Fig. 1. Structure of QA-HFNN. It comprises four parts: quantum fuzzy logic representation (black), DNN representation (purple), fusion layer (yellow), and classifier layer (green).

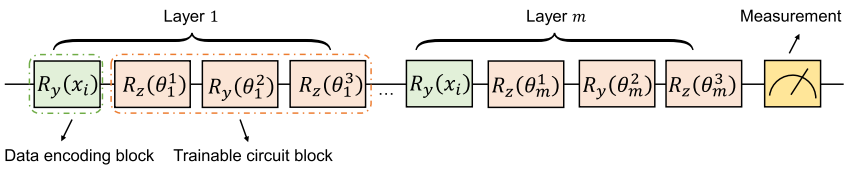


Fig. 2. Quantum circuit with single-qubit for quantum membership function.

$x_k$ is the $k$th element of the input $x$. This article proposes to use the QNN as shown in Fig. 2 as the membership function.

Fuzzy sets were first proposed by Zadeh [20] in 1995. Fuzzy sets are described as

$$A = \{(x_k, \mu_A(x_k)) \,|\, x_k \in X\} \tag{2}$$

$\mu_A$ represents the membership function. $X$ is the domain, representing the set of input variables. In the representation of a QNN (usually implemented with a PMC), the quantum membership function can be defined as

$$\begin{aligned}\mu_A(x_k) &= f_{\boldsymbol{\theta}}(x_k) \\ &= \left(\langle 0|^{\otimes n} U^{\dagger}(x_k, \boldsymbol{\theta}) M U(x_k, \boldsymbol{\theta}) |0\rangle^{\otimes n} + 1\right)/2\end{aligned} \tag{3}$$

where $|0\rangle^{\otimes n}$ represents the initial state

$$|0\rangle^{\otimes n} = \begin{bmatrix}1\\0\end{bmatrix}^{\otimes n} \tag{4}$$

$\otimes$ denotes Kronecker product, $\langle 0|^{\otimes n} = |0\rangle^{\otimes n\dagger}$, $n$ is the number of qubits. $U(x_k, \boldsymbol{\theta})$ is unitary matrix which represents the PQC determined by $x_k$ and $\boldsymbol{\theta}$, $\boldsymbol{\theta}$ is adjustable parameter vector. $M$ is an observable. Here, we choose Pauli operators $P^{\otimes n}, P \in \{\sigma_x, \sigma_y, \sigma_z, I\}$ as observable because they have eigenvalues of 1 and $-1$.

$$\begin{aligned}\sigma_x &= \begin{pmatrix}0 & 1\\1 & 0\end{pmatrix}, \sigma_y = \begin{pmatrix}0 & -i\\i & 0\end{pmatrix}, \\ \sigma_z &= \begin{pmatrix}1 & 0\\0 & -1\end{pmatrix}, I = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}\end{aligned} \tag{5}$$

$\sigma_x$, $\sigma_y$, and $\sigma_z$ denote the Pauli-$X$, Pauli-$Y$, and Pauli-$Z$ operators, respectively. $f_{\boldsymbol{\theta}}(x_k)$ can be estimated by measuring the quantum circuit multiple times. Since $\langle 0|U^{\dagger}(x_k, \boldsymbol{\theta}) M U(x_k, \boldsymbol{\theta})|0\rangle$ is the expectation under the observable $M$, then, $\langle 0|U^{\dagger}(x_k, \boldsymbol{\theta}) M U(x_k, \boldsymbol{\theta})|0\rangle \in [-1, 1]$. Therefore, $f_{\boldsymbol{\theta}}(x_k) \in [0, 1]$, which can be regarded as a membership function.

Here, we choose the quantum circuit structure as shown in Fig. 2 as $U(x_k, \boldsymbol{\theta})$ which adopts the data reuploading [34] method and consists of multiple layers of data encoding circuits and trainable circuit blocks. Fig. 2 shows a quantum circuit with only a single qubit, which is composed of multiple layers of circuit blocks, each circuit block containing data encoding circuit and trainable circuit block. Among them, the data encoding utilized the angle encoding method, that is, the $R_y$ gate is used to encode input data into quantum state

$$|x_k\rangle = R_y(x_k)|0\rangle \tag{6}$$

where

$$R_y(x_k) = \begin{bmatrix} \cos\frac{x_k}{2} & -\sin\frac{x_k}{2} \\ \sin\frac{x_k}{2} & \cos\frac{x_k}{2} \end{bmatrix}. \tag{7}$$

It is known that any single qubit gate $U$, which can be discomposed into three rotation gates

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta) \tag{8}$$

where, $e^{i\alpha}$ is a global phase and $R_z$ is the rotation gate around the Z-axis

$$R_z(r) = \begin{bmatrix} e^{-ir/2} & 0 \\ 0 & e^{ir/2} \end{bmatrix} \tag{9}$$

$r$ denotes the rotation angle.

Since the global phase cannot change the state of a quantum state, any single qubit state $|\varphi\rangle$ can be obtained from any initial state $|init\rangle$ by operating the $R_z, R_y$, and $R_z$ gate sequentially with proper angles

$$|\varphi\rangle = R_z(\alpha) R_y(\beta) R_z(\gamma) |init\rangle. \tag{10}$$

Then, the single qubit quantum circuit can be expressed as

$$U(x_k, \boldsymbol{\theta}) = \prod_l \left( R_z\left(\theta_l^3\right) R_y\left(\theta_l^2\right) R_z\left(\theta_l^1\right) R_y(x_k) \right) \tag{11}$$

$\theta = \{\theta_1^1, \theta_1^2, \theta_1^3, \ldots, \theta_L^1, \theta_L^2, \theta_L^3\}$, $l$ indicates the layer index of the circuit, $l \in \{1, 2, \ldots, L\}$. The above quantum circuit only uses a single qubit and does not use a two-qubit gate. Therefore, it is suitable for the current scenario where quantum computing resources are limited.

This layer maps the $k$th node of input to fuzzy degree

$$h_{i,k}{}^{(l)} = f_\theta{}^i(x_k), \forall i \tag{12}$$

in which, $h_{i,k}{}^{(l)}$ is the calculation result of the $i$th membership function. After fuzzy membership function processing, we enter the fuzzy rule layer, which is represented by 'AND' fuzzy logic, that is $h_i{}^{(l)} = \prod_k h_{i,k}{}^{(l-1)}$. Then, the input is converted into fuzzy degree.

*DNN representation:* The neural network representation block will obtain the neural representation. Here, we generally use a fully connected neural network to extract neural representation of the input. After the $l$th layer of fully connected neural network, we can get

$$ht_i{}^{(l)} = w_i{}^{(l)} h^{(l-1)} + b_i{}^{(l)} \tag{13}$$

in which, $w_i{}^{(l)}$ and $b_i{}^{(l)}$ represent the parameters from the $(l-1)$th layer node to the $l$th layer $i$th node. Then, ReLU activation function will be applied.

$$h_i{}^{(l)} = ReLU\left(ht_i{}^{(l)}\right) = \begin{cases} ht_i{}^{(l)}, & \text{if } ht_i{}^{(l)} > 0 \\ 0, & \text{other} \end{cases} \tag{14}$$

$h_i{}^{(l)}$ is the output of the $l$th layer neural network. To improve performance, the dropout strategy can be used in each layer of the fully connected network to prevent model over-fitting. It should be pointed out that for different tasks, specific neural network structures can be selected for replacement. For example, for image data, a CNN can be used to replace a fully connected neural network to obtain better classification results.

*Fusion layer:* The feature fusion is based on the idea of multimodal fusion [7], [46], which merging outputs of multiple networks to capture different features of the input data to form neural features, and finally, classifying them through the classification layer. We use the same aggregation method as in [7] to aggregate the outputs of the neural representation module and quantum fuzzy logic representation. The two representations are responsible for characterizing the fuzzy features and neural features of the input data. First, we combine the two features and obtain

$$h_{fus} = fuse(h_{fuz}, h_{neu}) \tag{15}$$

where fuse is a fusion function.

Then, a fully connected neural network is utilized to obtain

$$h_{fuz}{}^{(l)} = ReLU\left(w^{(l)} h_{fuz}{}^{(l-1)} + b^{(l)}\right) \tag{16}$$

in which, ReLU is activation function, $w^{(l)}$ and $b^{(l)}$ are the parameters from the $(l-1)$th layer node to the $l$th layer node. After multilayer fully connected network, the output consists of deeply fused fuzzy features and neural features.

*Classifier layer:* The classifier layer processes the fused features and classifies them into corresponding categories. Assuming that there are k categories of data labels in total, through the classifier layer, for the $i$th input sample, a $k$-dimensional output vector will be obtained. Then, a Softmax function is utilized to derive the predicted label

$$\widehat{y_i} = \text{Soft}\max(\widetilde{y_i}) = \frac{e^{\widetilde{y_i}}}{\sum_k e^{\widetilde{y_{ik}}}} \tag{17}$$

in which, $\widehat{y_i} = [\widehat{y_{i1}}, \widehat{y_{i2}}, \ldots, \widehat{y_{ik}}]$. In this article, the cross-entropy function is chosen as loss function. For $m$ training data, the loss is calculated as follows:

$$C = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{k} -y_{ij} \log \widehat{y_{ij}} - (1 - y_{ij}) \log\left(1 - \widehat{y_{ij}}\right) \tag{18}$$

In the model training phase, the initialization of model parameters and parameter optimization method need to be considered. Parameter initialization is very important in DNN learning. A good parameter initialization strategy can help the model converge to a better local minimum point. For the parameters of the classifier layer, we followed the strategy from [47] for initialization. The bias of each layer is initialized to 0, and the weights are initialized following the rules:

$$w_{ij}{}^{(l)} \sim U\left[-\frac{1}{\sqrt{n^{(l-1)}}}, \frac{1}{\sqrt{n^{(l-1)}}}\right] \tag{19}$$

in which, $U$ represents uniform distribution, $n^{(l-1)}$ represents the nodes number in the $(l-1)$th layer. According to different learning tasks, we will fine-tune the value of $n^{(l-1)}$ and do not completely follow the above rules.

During the parameter optimization process, we first need to calculate the derivative for each parameter, and then, use the classic optimization algorithm to update the parameters. For the QNN part, i.e., the membership function layer, the gradient can

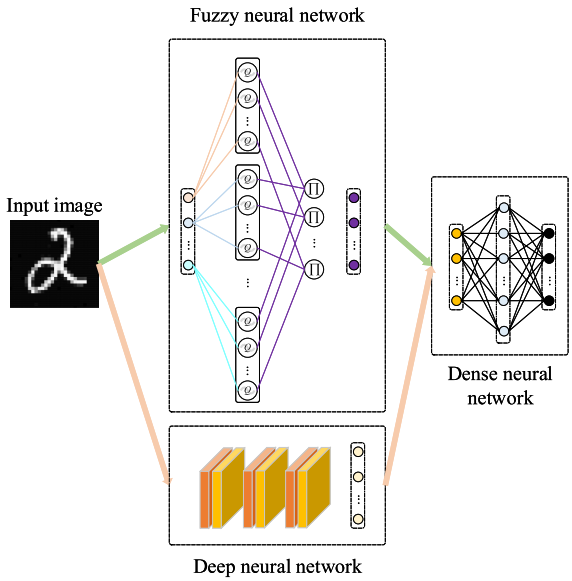


Fig. 3. Visualization process of the QA-HFNN model. $Q$ denotes quantum membership function.

be obtained using the parameter shift method

$$\frac{\partial f_\theta(x)}{\partial \theta_i} = \frac{2}{\pi}\left(f_{\theta_i+\frac{\pi}{2}}(x) - f_{\theta_i-\frac{\pi}{2}}(x)\right) \tag{20}$$

in which, $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$ and $\theta_i \pm \frac{\pi}{2} = [\theta_1, \theta_2, \ldots, \theta_{i\pm\frac{\pi}{2}}, \ldots, \theta_n]$. The classic neural network in the model uses the backpropagation algorithm to obtain the gradient of each parameter. Then, the parameters is updated by an SGD algorithm. The learning rate is set to 0.01, which will be gradually decreased by a decay strategy. The decay multiplication factor is set to 0.1. It is proposed that there are many nodes in the neural network layer in the model, and the model can easily be overfitting in training phase. In order to alleviate the phenomenon of over-fitting, we use the dropout strategy, that is, for a certain layer of neural network, some nodes do not participate in the gradient update according to a certain percentage. Dropout probability is set to 40%. The specific visualization process of using the proposed model to process image data is shown in the Fig. 3. The steps of the entire training are shown in Algorithm 1.

**Algorithm 1:** QA-HFNN Model Training Steps.

**Input:** training samples and labels $(x, y)$, learning rate $\alpha$, number of categories $k$, hidden layer feature dimension $h$, number of training epochs $N$.

**Components:**

Quantum fuzzy logic representation: $FuzMem(Inp\_dim, k \times Inp\_dim)$, $Inp\_dim$ indicates the dimension of input;

Deep neural network representation: $NeurNet(Inp\_dim, h)$;

Fusion layer: $Linear(k, h)$;

Classifier layer: $ClassifierL(h, k)$.

**for** $e = 1, \ldots, N$ **do**

$FuzFea = FuzMem(x)$

$NeurFea = NeurNet(x)$

$FuzRulFea = PROD(FuzFea, \dim = 1)$

$FusFea = ADD(Linear(FuzRulFea), NeurFea)$

$\widehat{y} = \text{Softmax}(ClassifierL(FusFea))$

Calculate the loss function $C$ according to $\widehat{y}$ and $y$;

Calculating gradients using the backpropagation algorithm and updating all parameters using the SGD algorithm

**end for**

**Output:** The well-trained model.

## IV. Experiments

### A. Experiment Settings

*1) Experiment Environment:* We conducted the experiment on the Linux platform. PyTorch is used to construct classical neural networks, while the quantum circuit part is implemented using the Torch Quantum Library [48].

*2) Datasets:* In this article, Dirty-MNIST, 15-Scene, Japanese Female Facial Expression (JAFFE), Fasion MNIST, and CIFAR-10 datasets are selected to evaluate the model.

Dirty-MNIST is a combination of MNIST and AmbigouslyMNIST, with 60 000 sample-label pairs in the training set [49]. AmbigeousMNIST comprises generated ambiguous samples with varying entropy: 6000 unique samples, each having 10 labels. By default, the dataset is being normalized and Gaussian noise is introduced to each sample with standard deviation 0.05.

The 15-Scene dataset includes 15 different natural and urban scenes [50], such as, beaches, city streets, forests, offices, kitchens, living rooms, etc. Since it includes a variety of different types of scenes, from natural landscapes to indoor environments, this dataset provides greater challenges for algorithms, especially in handling the diversity and complexity of the scenes. This dataset comprises over 4500 natural scenery images. A total of 100 images from each category are selected as the training set and the remaining images are selected as the testing set.

The JAFFE dataset consists of 213 expression pictures depicting 7 different expressions of 10 Japanese women [51], [52]. The expressions include 6 facial expressions and 1 neutral expression. The size of each expression picture is 256×256. We preprocess this dataset and extract the facial expression area.

The Fashion MNIST dataset consists of images from 10 distinct categories [53], each category represents a distinct type of clothing. Each image is a grayscale image with 28×28 pixels. The dataset contains 60 000 training samples 10 000 testing samples.

The CIFAR-10 dataset [54] is widely used in image processing. It consists of 60 000 color images, each with dimensions of 32 × 32 pixels, divided into 50 000 images for training and 10 000 images for testing. The dataset comprises 10 categories, with 6000 images in each category.

*3) Hyperparameters:* In this experiment, the hyperparameters include settings in classical neural networks and QNNs. The hyperparameter settings of QA-HFNN are shown in Table II.

TABLE I
PERFORMANCE OF MODELS ON DIRTY-MNIST

| Model | Accuracy | Recall | Precision | F1-Score |
|---|---|---|---|---|
| FDNN | 0.838 | 0.85 | 0.855 | 0.852 |
| QA-HFNN | **0.84** | **0.851** | **0.857** | **0.853** |
| ResNet18 | 0.839 | 0.85 | 0.856 | 0.852 |
| ResNet50 | 0.834 | 0.844 | 0.849 | 0.846 |
| Fuzzy pattern [55] | 0.71 | 0.89 | 0.79 | 0.76 |
| Fuzzy reduction rule [56] | 0.80 | 0.89 | 0.88 | 0.91 |
| Fuzzy pattern tree top down [57] | 0.83 | 0.84 | 0.87 | 0.90 |
| Multimodal evolutionary [58] | 0.79 | 0.89 | 0.8 | 0.72 |
| Fuzzy pattern classifier GA [59] | 0.82 | 0.81 | 0.82 | 0.89 |

The bold values represent the metrics of our proposed model.

TABLE II
HYPERPARAMETER SETTINGS IN THE MODEL

| Hyper-parameter | Value |
|---|---|
| Batch size | 128 |
| Qubits (each QNN) | 1 |
| Learning rate | 0.01 |
| Activation function | ReLU |
| Measurement | Pauli-$Z$ |
| Quantum encoding gate | $R_y$ |

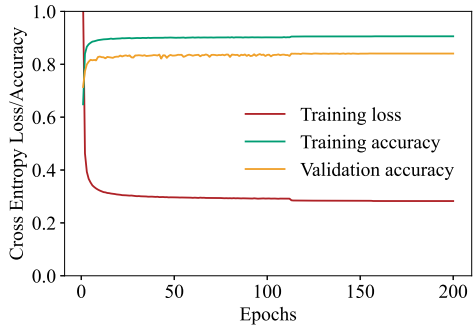


Fig. 4. Convergence analysis of the training on Dirty-MNIST.

*4) Evaluation Criteria:* In binary classification tasks, the metrics include accuracy, precision, recall and F1 score which can evaluate how the model performs in different aspects. In multiclass classification tasks, metrics like macroprecision, macrorecall, and macro-F1 are commonly utilized [10].

## B. Performance Evaluation

For the Dirty-MNIST dataset, FDNN is chosen as one of the benchmarks. The DNN representation part of the proposed model is set to CNN. The CNN comprises two layers of convolutional and pooling layers. The convolution kernel size of the first layer is set to $5 \times 5$, with a stride of 1, and it yields 10 output channels. For the second layer, the kernel size is also $5 \times 5$, with a stride of 1, and it yields 20 output channels. In addition, each pooling layer utilizes a pooling kernel of size 2 with a stride equal to the pooling kernel size. After the processing is completed, the output is flattened into a 1-D vector. Subsequently, the data are passed through a fully connected neural network, mapping it to 128-D features. The abovementioned CNN network is the DNN representation part of FDNN or QA-HFNN. If utilized as an independent CNN model, a classification layer must be added to the network to map the 128-D features to 10-D features, thereby, classifying the hidden layer features into respective categories.

In order to observe the convergence of QA-HFNN, we provide training details depicted in Fig. 4. In the first few epochs, the loss drops rapidly, and the training accuracy and validation accuracy increase rapidly. At the fifth epoch, the loss dropped to about 0.35, the training accuracy reached about 88.3%, and the validation accuracy reached about 81.5%. As the epoch increases, the loss slowly decreases. Until epoch reaches 116, where the learning rate begins to decay, the loss experiences a significant drop. Subsequently, the loss curve gradually flattens out, indicating that the model has converged.

The performance of QA-HFNN on the Dirty-MNIST dataset is given in Table I. In addition to FDNN, we selected classifiers such as, fuzzy pattern classifier, fuzzy reduction rule classifier, fuzzy pattern tree top down classifier, multimodal evolutionary classifier, and fuzzy pattern classifier GA as benchmarks. Fuzzy pattern classifier uses fuzzy aggregation functions and fuzzy sets to represent data, fuzzy reduction rule uses min/max membership functions to learn fuzzy representation of data, fuzzy pattern tree top down uses the top-down method to build fuzzy pattern trees, and multimodal evolutionary uses genetic algorithms to learn fuzzy rules, fuzzy pattern classifier GA uses genetic algorithms to optimize membership functions at a global scale. These classifiers can be implemented using fylearn [55] which is based on SciKit-Learn. Table I shows that, compared with FDNN, QA-HFNN can achieve the performance of FDNN and be slightly better in each metric. The QA-HFNN performs better than other fuzzy classifiers under the accuracy metric. The results means that QNNs can be used to learn membership functions and achieve better performance.

The performance of QA-HFNN is also evaluated on other datasets, i.e., 15-Scene, JAFFE, Fashion MNIST, and CIFAR-10. DNN and FDNN are chosen as benchmarks. We first preprocess the 15-Scene data, and we adopt the same method as in [7]. Specifically, the dense SIFT features of each image

TABLE III
ACCURACY COMPARISON OF PROPOSED MODEL ON OTHER DATASETS

| Model | 15-Scene | JAFFE | Fashion MNIST | CIFAR-10 |
|---|---|---|---|---|
| DNN | 0.737 | 0.891 | 0.905 | 0.953 |
| FDNN | 0.74 | 0.924 | 0.907 | 0.954 |
| QA-HFNN | **0.742** | **0.933** | **0.907** | **0.956** |

The bold values represent the metrics of our proposed model.

are extracted. Then, the local features are clustered to 200 codewords. Finally, the SIFT features are allocated to 200-bit codewords through the kernel algorithm. We choose a 3-layer $256 \times 256$ fully connected neural network for the hidden layer of DNN. A ReLU function is applied to each layer of the network, and the dropout probability is set to 40%. The DNN in the proposed model and FDNN model adopts the same DNN network. For the JAFFE dataset, we first preprocess the data, convert the image to grayscale and extract regions containing only facial expressions. Then, we performed data augmentation on the dataset, including random horizontal flipping and random rotation by a certain angle. As the dataset is relatively small, the k-fold cross-validation method is used, where $k = 8$. For DNN, we chose CNN with residual block. Specifically, DNN contains 6 convolutional layers and 2 residual layers. The number of output channels of the first convolutional layer is 32, the kernel size is 5, and the stride is 2. In the second convolutional layer, kernel size and stride is set to 3 and 1, respectively, and the number of output channels is 64. Each two convolutional networks are followed by a residual layer. The residual layer consists of 2 convolutional layers and a skip connection. In the first residual layer, the number of input and output channels is 64, and the input and output channels of the second residual layer is both 128. The third convolutional layer has output channels of 128 with a stride of 2 and a kernel size of 3. The subsequent convolutional layers only adjust the number of output channels, and the number of output channels of the last convolutional layer is 512. Finally, the flattened features are then, fed into a fully connected neural network. The DNN part of both FDNN and QA-HFNN adopts the abovementioned DNN. For the Fashion MNIST dataset, we use the same DNN as the Dirty-MNIST dataset for the DNN part.

The performance of QA-HFNN is shown in Table III. It illustrates that the accuracy of QA-HFNN reaches 74.2%. On the 15-Scene dataset, the proposed model achieves slightly better performance than DNN and FDNN. On the Fashion MNIST dataset, QA-FHNN does not show a significant improvement compared to DNN and FDNN. For the JAFFE dataset, the accuracy of QA-HFNN reaches 93.3%, which improves 4.2% and 0.9% compared with DNN and FDNN, respectively. On the CIFAR-10 dataset, the accuracy of QA-HFNN reaches 95.6%, which performs better than DNN and FDNN. The results indicate that the proposed model can effectively extract the uncertainty of the datasets, achieving significant classification accuracy.

For fuzzy membership functions, this article attempts to use QNNs to construct fuzzy membership functions. The proposed QNN uses only single qubit. A layer of QNN contains a $R_y$ gate encoding input data and adjustable parameter quantum gates $R_z$, $R_y$, $R_z$. It is easy to find that a one-layer QNN contains three adjustable parameters. With increasing layers, the parameters of this part of the network show a linear growth trend. Compared with the Gaussian membership function, which only has two adjustable parameters, mean and variance. As the layers of the QNN increase, the training difficulty of the proposed model also increases, but the advantage is that the QNN can learn more suitable fuzzy membership function. Therefore, for different problems, the proposed membership functions may be more suitable for the task be learned. It is believed that the proposed model may also perform better in classification tasks on other image datasets.

Regarding that quantum part may be affected by quantum noise, the quantum circuit built by the proposed fuzzy membership function only utilizes single qubit, does not involve two-qubit CNOT gates [60]. Most of the noise in quantum circuits comes from the CNOT gate which requires a long operation time. Therefore, the quantum part is easy to implement and less affected by noise. A detailed discussion of quantum noise will be addressed in the following section.

### C. Robustness of Quantum Circuits

During the execution of quantum circuits, the quantum system will be affected by environmental noise. If the quantum circuit has weak resistance to environmental noise, it will seriously affect the accuracy of QNN. Therefore, if environmental noise has a small impact on the quantum circuit, then, we say that the quantum circuit has good robustness. In this article, we mainly analyze quantum circuits with single qubits. We will simulate four types of quantum noise in quantum circuits, which are the following: 1) amplitude damping (AD), 2) depolarization (DP) noise, 3) bit flip (BF), and 4) phase flip (PF).

1) AD:

$$E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix} \quad E_1 = \begin{pmatrix} 1 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}. \tag{21}$$

2) DP:

$$E_0 = \sqrt{1-\gamma}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad E_1 = \sqrt{\frac{\gamma}{3}}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$$E_2 = \sqrt{\frac{\gamma}{3}}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad E_3 = \sqrt{\frac{\gamma}{3}}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}. \tag{22}$$

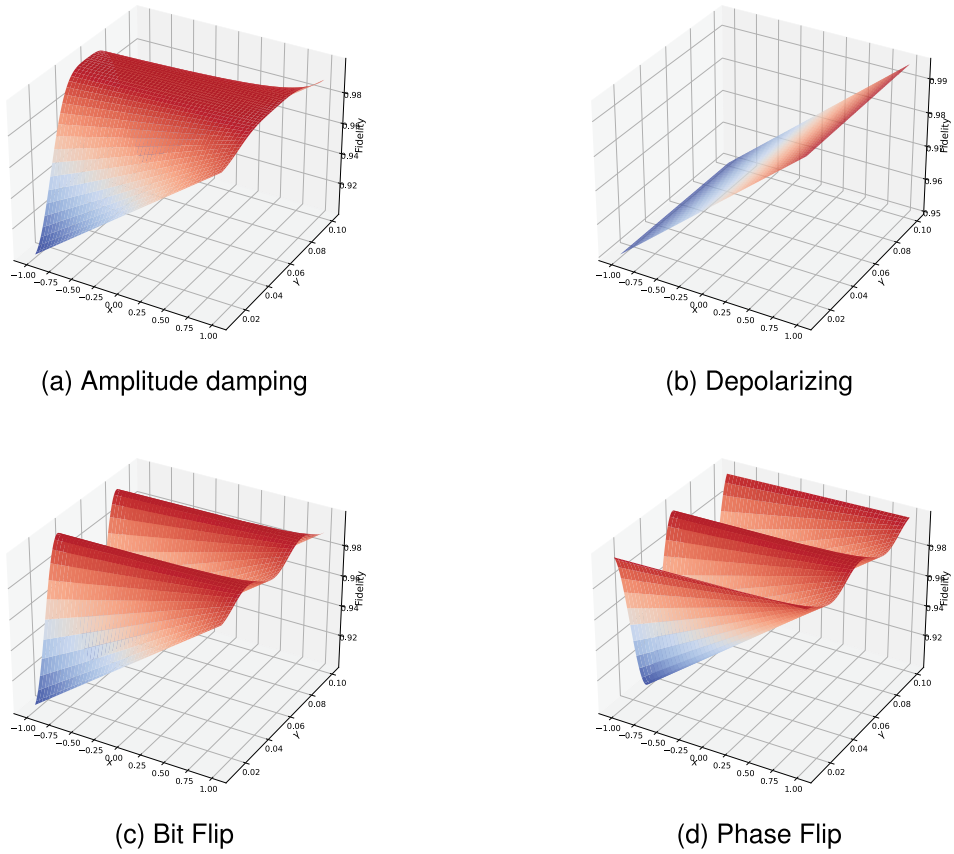


Fig. 5. Fidelity of output quantum state of quantum circuit changes with input $x$ and noise probability $\gamma$. (a) AD. (b) DP. (c) BF. (d) PF.

3) BF:

$$E_0 = \sqrt{1-\gamma}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \quad E_1 = \sqrt{\gamma}\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}. \tag{23}$$

4) PF:

$$E_0 = \sqrt{1-\gamma}\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \quad E_1 = \sqrt{\gamma}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix} \tag{24}$$

Amplitude damping describes the energy dissipation of quantum systems. Depolarization noise describes the transformation of a quantum state into a maximum mixed state $I/2$ with a certain probability. $\gamma$ is the probability of noise acting on the quantum state. Bit flip and phase flip, respectively, mean flipping the qubit or adding a relative phase to the qubit with a certain probability.

The evolution process of noise acting on quantum systems is described by operator-sum

$$\varepsilon(\rho) = \sum_k E_k \rho {E_k}^{\dagger} \tag{25}$$

in which, $\{E_k\}$ is called the Kraus operator, satisfies $\sum_k {E_k}^{\dagger} E_k = I$, and $\varepsilon(\rho)$ is the evolved quantum system. We apply a quantum noise behind each single qubit gate to complete the simulation of noisy environment.

For quantum circuits containing noise, we use fidelity to quantify the tolerance of quantum circuits to noise. Fidelity represents the degree of similarity of two arbitrary quantum states

$$F(\rho,\sigma) = Tr\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2. \tag{26}$$

Here, we calculate the fidelity between the quantum state generated by the quantum circuit without quantum noise and the quantum state affected by quantum noise. The closer the fidelity is to 1, the better the quantum circuit's resistance to noise.

The input $x$ of the quantum circuit is a real number in $[-1,1]$. We sampled 100 input values in $[-1,1]$ and analyzed the change of the noise probability from 0.01 to 0.1 in each input case. The fidelity of output quantum state of quantum circuit changes with input and noise probability as shown in Fig. 5. Subsequently, we averaged the fidelity of 100 inputs under the same noise to obtain the quantum circuit fidelity. We show the quantum circuit fidelity when the noise probability is 0.01, 0.03, 0.05, 0.07, and 0.1 in Table IV. It shows that as the noise probability increases, the fidelity of the quantum circuit gradually decreases. Although noise probability becomes 0.1, the fidelity remains as high as 0.9481. Thus, it illustrates that the proposed QA-HFNN exhibits robustness against quantum noise.

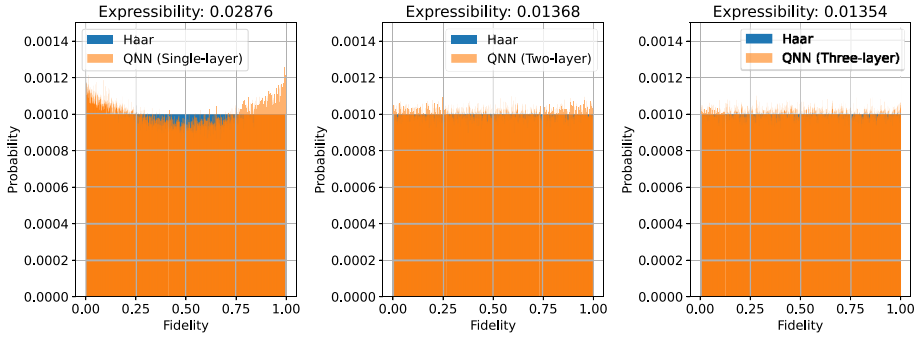


Fig. 6. Expression for QNN in QA-HFNN.

TABLE IV
FIDELITY OF NOISY QUANTUM CIRCUIT

| $\gamma$ | AD | DP | BF | PF |
|---|---|---|---|---|
| 0.01 | 0.9964 | 0.995 | 0.9951 | 0.9948 |
| 0.03 | 0.9894 | 0.985 | 0.9855 | 0.9844 |
| 0.05 | 0.9823 | 0.975 | 0.9759 | 0.9741 |
| 0.07 | 0.9751 | 0.965 | 0.9662 | 0.9637 |
| 0.1 | 0.9644 | 0.95 | 0.9518 | 0.9481 |

TABLE V
EXPRESSIBILITY AND ENTANGLING CAPABILITY OF QNN

| QNNs | Expressibility | Entangling capability |
|---|---|---|
| QNN (Single-layer) | 0.02876 | 0 |
| QNN (Two-layer) | 0.01368 | 0 |
| QNN (Three-layer) | 0.01354 | 0 |

### D. Expressibility and Entangling Capability of QNN

In order to measure whether PQCs can effectively represent the solution space, Sim et al. [61] proposed a technique to calculate the expressibility and entangling capability of PQCs. This method first calculates the fidelity distribution between the state generated by the target QNN and the Haar random state. The Haar random state is considered to be the state with the greatest expressive power. Then, the Kullback–Leibler (KL) divergence between the estimated fidelity distribution and the Haar distribution is calculated. Expressibility can be calculated as follows:

$$Expr = D_{\mathrm{KL}}\left(\hat{P}_{\mathrm{QNN}}(F;\theta) \parallel P_{\mathrm{Haar}}(F)\right) \tag{27}$$

where $F$ represents fidelity and $\theta$ represents the adjustable parameters in QNN. $\hat{P}_{\mathrm{QNN}}$ is the estimated fidelity probability distribution obtained from the quantum states in the sampled QNN, and $P_{\mathrm{Haar}}$ represents the Haar state fidelity probability distribution. The smaller the value of the expressibility, the closer the quantum state prepared by the QNN is to the Haar random state, and the stronger the expressibility of the QNN.

For estimating the entangling capability, the operation of Meyer–Wallach measurement is proposed in [61]. An estimate of the entangling capability can be calculated by

$$Ent = \frac{1}{|S|}\sum_{\theta_i \in S} Q\left(|\psi_{\theta_i}\rangle\right) \tag{28}$$

where $Q$ represents the Meyer–Wallach entanglement measurement, $S$ represents the parameter set of the QNN, and $|\psi_{\theta_i}\rangle$ represents the output quantum state of the QNN with parameters $\theta_i$. If $Ent$ is closer to 1, it means that the circuit has produced a highly entangled quantum state.

We discussed three QNN settings in QA-HFNN, corresponding to QNNs with single-layer, two-layer, and three-layer circuit blocks, respectively. The comparison of expressibility of QNNs is shown in Fig. 6. When the QNN is single-layer, it already has a high expressibility. The fidelity probability distribution of the quantum state generated by the QNN basically covers the fidelity probability distribution of the Haar state. This may be because of the $R_z$, $R_y$, and $R_z$ gates. It can represent any unitary operation on a single qubit. When the circuit is increased to two layers, the expressibility increases by 52.43% compared to single-layer. When it is increased to three layers, the expressibility of QNN has not increased significantly. QA-HFNN uses QNNs containing three layers of circuit blocks, which has considerable expressive capabilities.

Regarding the entangling capability, since the QNNs used in QA-HFNN only utilizes a single qubit gate and does not contain entanglement resources in the circuit, the entanglement capability is 0. Even without entanglement capability, QA-HFNN still performs well. A comparison of expressibility and entangling capability has been given in Table V.

### E. Hyperparameter Analysis

To explore the impact of hyperparameter fine-tuning on model performance, this article analyzes hyperparameter sensitivity. Specifically, hyperparameters such as learning rate (LR), activation function (AF), number of training epochs (TE), batch size (BS), and the number of circuit layers (CL) are considered for testing. In this test, we adjust only one parameter at a time while keeping the other parameters constant and compare the average change in model performance resulting from the hyperparameter adjustment. A single qubit is used in the QNN, we examine the sensitivity of the number of circuit layers. For this study, as a

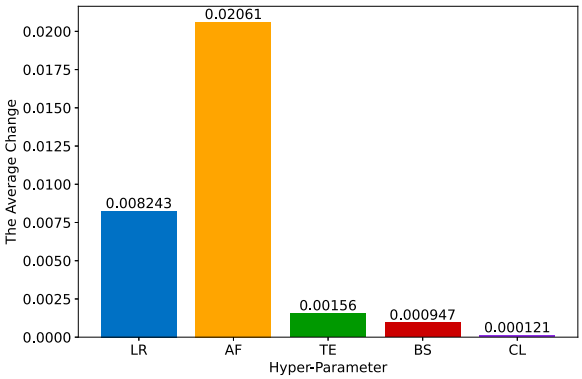


Fig. 7. Hyperparameter analysis.

QNN can exhibit good expressibility with just a single layer of circuits, we consider the circuit layers of 1, 2, and 3.

The average change in model performance caused by each hyperparameter adjustment is shown in Fig. 7. It is evident that the choice of activation function has the most significant impact on QA-HFNN, with an average change rate of 2.06% in model performance. The second most influential hyperparameter is the learning rate, with an average change of 0.8%. Following is the training epoch, with an average change rate of 0.15%. The selection of batch size also has a relatively small average impact on model performance, which is 0.09%. The number of circuit layers has the smallest impact on model performance, with an average change rate of only 0.012%. Therefore, the adjustment of hyperparameters should focus more on adjusting the activation function and learning rate.

### F. Complexity Analysis

Complexity analysis involves two aspects as follows: 1) quantum circuit complexity and 2) computational complexity. For quantum circuit complexity, we can calculate it by the number of quantum gates used in QA-HFNN. The number of qubits is equal to the input data dimension $N$, the circuit layer is $l$, each circuit layer contains 4 quantum gates, the total number of quantum gates is $4lN$. Since $l$ is constant, the quantum circuit complexity is $O(N)$.

Computational complexity mainly includes space complexity and time complexity. Space complexity primarily depends on the number of qubits used. Since we use $N$ qubits, the space complexity is $O(N)$. Regarding time complexity, given that the circuit depth of the QNNs is $4l$ and the circuit contains only single-qubit rotation gates, if the execution time of a single-qubit rotation gate is $T$, the total circuit execution time is $4lT$. Since both $l$ and $T$ are constants, the time complexity of the QNNs is $O(1)$.

## V. Conclusion

This article proposed a QA-HFNN that utilizes QNNs to learn membership functions in classical fuzzy neural network. The proposed model aggregates the fuzzy logic representation extracted by QNNs and the representation of classical neural networks. We use QNNs to learn fuzzy membership functions. After the data are processed by the fuzzy membership functions, we use the logic rule layer to obtain fuzzy logic representations. The fusion of fuzzy logic representation and features extracted by DNNs can effectively represent noisy or ambiguous data. The proposed model achieves better classification accuracy than existing classical methods. The fidelity is tested in noisy environments, which indicates QA-HFNN has strong robustness against quantum noise. However, the performance of QA-HFNN may be compromised in physical quantum computing environment due to the current quantum devices limitations. In addition, utilizing only a single qubit to construct a QNN will inherently restrict the network's capacity to handle more complex or larger datasets. In the future, more qubits are considered to construct the QNN. Moreover, the integration of entanglement resources is expected to further augment the model's efficacy. As quantum computers mature, the feasibility of QA-HFNN can be further verified and it is hoped that more efficient and practical models can be developed with the aid of QNNs.

## References

[1] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *2013 IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 7398–7402.
[2] J. Gawlikowski et al., "A survey of uncertainty in deep neural networks," 2021, *arXiv:2107.03342*.
[3] K. Bhalla, D. Koundal, B. Sharma, Y.-C. Hu, and A. Zaguia, "A fuzzy convolutional neural network for enhancing multi-focus image fusion," *J. Vis. Commun. Image Representation*, vol. 84, 2022, Art. no. 103485.
[4] R. S. Lee, "Chaotic type-2 transient-fuzzy deep neuro-oscillatory network (CT2TFDNN) for worldwide financial prediction," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 4, pp. 731–745, Apr. 2020.
[5] S. Hou, J. Fei, C. Chen, and Y. Chu, "Finite-time adaptive fuzzy-neural-network control of active power filter," *IEEE Trans. Power Electron.*, vol. 34, no. 10, pp. 10298–10313, Oct. 2019.
[6] S. Zhou, Q. Chen, and X. Wang, "Fuzzy deep belief networks for semi-supervised sentiment classification," *Neurocomputing*, vol. 131, pp. 312–322, 2014.
[7] Y. Deng, Z. Ren, Y. Kong, F. Bao, and Q. Dai, "A hierarchical fused fuzzy deep neural network for data classification," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 4, pp. 1006–1012, Aug. 2017.
[8] M. Yeganejou, S. Dick, and J. Miller, "Interpretable deep convolutional fuzzy classifier," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1407–1419, Jul. 2020.
[9] S. Mensa, E. Sahin, F. Tacchino, P. Kl Barkoutsos, and I. Tavernelli, "Quantum machine learning framework for virtual screening in drug discovery: A prospective quantum advantage," *Mach. Learn.-Sci. Technol.*, vol. 4, no. 1, 2023, Art. no. 015023.
[10] P. Tiwari, L. Zhang, Z. Qu, and G. Muhammad, "Quantum fuzzy neural network for multimodal sentiment and sarcasm detection," *Inf. Fusion*, vol. 103, 2024, Art. no. 102085.
[11] C. Van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel codebooks for scene categorization," in *Proc. Comput. Vis.–2008: 10th Eur. Conf. Comput. Vis.* Marseille, France, Oct. 2008, pp. 696–709.
[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.
[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[16] W. Wang et al., "InternImage: Exploring large-scale vision foundation models with deformable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 14408–14419.

[17] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "CoatNet: Marrying convolution and attention for all data sizes," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 3965–3977.
[18] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[19] S. K. Roy, A. Deria, D. Hong, B. Rasti, A. Plaza, and J. Chanussot, "Multimodal fusion transformer for remote sensing image classification," *IEEE Trans. Grosci. Remote Sens.*, vol. 61, pp. 1–20, 2023.
[20] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965.
[21] A. Sarabakha and E. Kayacan, "Online deep fuzzy learning for control of nonlinear systems using expert knowledge," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 7, pp. 1492–1503, Jul. 2020.
[22] D. Połap, "Fuzzy consensus with federated learning method in medical systems," *IEEE Access*, vol. 9, pp. 150383–150392, 2021.
[23] L. Zhang, Y. Shi, Y.-C. Chang, and C.-T. Lin, "Federated fuzzy neural network with evolutionary rule learning," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 5, pp. 1653–1664, May 2023.
[24] C. Wang, X. Lv, M. Shao, Y. Qian, and Y. Zhang, "A novel fuzzy hierarchical fusion attention convolution neural network for medical image super-resolution reconstruction," *Inf. Sci.*, vol. 622, pp. 424–436, 2023.
[25] A. Yazdinejad, A. Dehghantanha, R. M. Parizi, and G. Epiphaniou, "An optimized fuzzy deep learning model for data classification based on NSGA-II," *Neurocomputing*, vol. 522, pp. 116–128, 2023.
[26] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
[27] G. L. Long, "Enhanced universality in quantum neural networks with fewer qubits and reduced circuit depth," *Sci. China Phys. Mech.*, vol. 66, no. 7, 2023, Art no. 270361.
[28] S. C. Kak, "Quantum neural computing," *Adv. Imag. Electron Phys.*, vol. 94, pp. 259–313, 1995.
[29] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, "Quantum optimization for training support vector machines," *Neural Netw.*, vol. 16, no. 5–6, pp. 763–770, 2003.
[30] N. Wiebe, D. Braun, and S. Lloyd, "Quantum algorithm for data fitting," *Phys. Rev. Lett.*, vol. 109, no. 5, 2012, Art. no. 050505.
[31] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," 2013, *arXiv:1307.0411*.
[32] I. Kerenidis, J. Landman, A. Luongo, and A. Prakash, "Q-means: A quantum algorithm for unsupervised machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 4134–4144.
[33] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
[34] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum-Astria*, vol. 4, 2020, Art. no. 226.
[35] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, no. 3, 2018, Art. no. 032309.
[36] S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," *Phys. Rev. Lett.*, vol. 121, no. 4, 2018, Art. no. 040502.
[37] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Phys.*, vol. 15, no. 12, pp. 1273–1278, 2019.
[38] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Phys. Rev. A*, vol. 101, no. 3, 2020, Art. no. 032308.
[39] Z. Qu, X. Liu, and M. Zheng, "Temporal-spatial quantum graph convolutional neural network based on schrödinger approach for traffic congestion prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8677-8686, Aug. 2023.
[40] A. Skolik, S. Mangini, T. Bäck, C. Macchiavello, and V. Dunjko, "Robustness of quantum reinforcement learning under hardware errors," *EPJ Quantum Technol.*, vol. 10, no. 1, pp. 1–43, 2023.
[41] Y. Song et al., "A quantum federated learning framework for classical clients," *Sci. China Phys. Mech.*, vol. 67, no. 5, 2024, Art. no. 250311.
[42] Y. Song, J. Li, Y. Wu, S. Qin, Q. Wen, and F. Gao, "A resource-efficient quantum convolutional neural network," *Front. Phys.*, vol. 12, 2024, Art. no. 1362690.
[43] G. Melnichenko, "Quantum decision, quantum logic, and fuzzy sets," 2007, *arXiv:0711.1437*.
[44] Z. Qu, L. Zhang, and P. Tiwari, "Quantum fuzzy federated learning for privacy protection in intelligent information processing," *IEEE Trans. Fuzzy Syst.*, 2024, to be published, doi: 10.1109/TFUZZ.2024.3419559.
[45] C. T. Lin, C. M. Yeh, S. F. Liang, J. F. Chung, and N. Kumar, "Support-vector-based fuzzy neural network for pattern classification," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 1, pp. 31–41, Feb. 2006.
[46] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 689–696.
[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[48] H. Wang et al., "Quantumnas: Noise-adaptive search for robust quantum circuits," in *2022 IEEE Int. Symp. High- Perform. Comput. Architecture*, 2022, pp. 692–708.
[49] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. Torr, and Y. Gal, "Deterministic neural networks with appropriate inductive biases capture epistemic and aleatoric uncertainty," 2021, *arXiv:2102.11582*.
[50] "Scene-15 dataset," 2020. [Online]. Available: https://www.kaggle.com/yiklunchow/scene15
[51] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Proc. 3rd IEEE Int. Conf. Autom. Face Gesture Recognit.*, 1998, pp. 200–205.
[52] M. J. Lyons, "Excavating AI" re-excavated: Debunking a fallacious account of the JAFFE dataset," 2021, *arXiv:2107.13998*.
[53] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[54] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," Toronto, ON, Canada, 2009.
[55] "fylearn, Fuzzy machine learning algorithms," 2020. [Online]. Available: https://github.com/sorend/fylearn
[56] J. Casillas, O. Cordón, M. J. Del Jesus, and F. Herrera, "Genetic tuning of fuzzy rule deep structures preserving interpretability and its interaction with fuzzy rule set reduction," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 1, pp. 13–29, Feb. 2005.
[57] R. Senge and E. Hüllermeier, "Top-down induction of fuzzy pattern trees," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 241–252, Apr. 2011.
[58] Y. Liu, G. G. Yen, and D. Gong, "A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 660–674, Aug. 2019.
[59] K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi, and K. Vijayakumar, "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks," *Concurrency Comput.-Pract. Experience*, vol. 33, no. 3, 2021, Art. no. e5242.
[60] Z. Li et al., "Novel quantum circuit implementation of advanced encryption standard with low costs," *Sci. China Phys. Mech.*, vol. 65, no. 9, 2022, Art. no. 290311.
[61] S. Sim, P. D. Johnson, and A. Aspuru-Guzik, "Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms," *Adv. Quantum Technol.*, vol. 2, no. 12, 2019, Art. no. 1900070.