2024 IEEE International Conference on Communications; Selected Areas in Communications: Quantum Communications and Information Technology

Hybrid Quantum Classical Machine Learning with Knowledge Distillation

Mingze Li[1,2], Lei Fan[1,2], Aaron Cummings[3], Xinyue Zhang[3], Miao Pan[1], and Zhu Han[1]
Dept. of Electrical and Computer Engineering[1], Engineering Technology[2], University of Houston, TX, USA
[3]Department of Computer Science, Kennesaw State University, Marietta, GA, USA

***Abstract*—The rapid advancement of machine learning (ML) and the growing need for computational power have led to the exploration of quantum computing, which offers significant potential for faster complex calculations. However, Quantum Machine Learning (QML) faces challenges due to the limited number of qubits and noise of quantum circuits, particularly with Noisy Intermediate-Scale Quantum (NISQ) devices. These challenges severely limit the current capacity to train accurate and stable Quantum Machine Learning Models. In this paper, we propose a novel framework for QML that employs the knowledge distillation method to harness the power of well-trained classical machine learning (CML) models and enhance the training performance of QML models. In this framework, we utilize the well-trained CML as a teacher model to assist the training of the student QML model using the knowledge distillation method. By distilling knowledge from the robust CML model, our framework can potentially address the problem of the barren plateau which hinders effective model training. Knowledge distillation is well suited for this framework through the transfer of knowledge without parameter sharing. Through empirical tests, our framework has demonstrated not only an increase in the accuracy of QML models but also a notable improvement in training stability.**

***Index Terms*—Quantum Computing, Machine Learning, Quantum Machine Learning, Knowledge Distillation.**

I. Introduction

Recent advancements in machine learning can largely be attributed to the increasing size of models to leverage the increasing volume of data across many domains of applications [1]. The increase in model size rapidly necessitates advancements over traditional paradigms in computing to compensate for the computational power necessary for handling immense data volume [2]. This necessity is particularly evident in the realm of modern communication and networking systems. In this context, quantum computing has emerged as a promising alternative to classical computing [3]. Quantum computers leverage quantum phenomena, such as superposition and entanglement, which can potentially accelerate the training and inference speed of machine learning models, especially for complex problems and large datasets [4]–[6]. Quantum machine learning (QML), a multidisciplinary field that combines quantum computing and machine learning, aims to harness the power of quantum computing to solve complex problems efficiently [7], [8]. Variational quantum circuits (VQC) can be employed to construct QML models. VQC has been applied in hybrid quantum-classical models that utilize current classical optimization techniques such as stochastic gradient descent, while taking advantage of the quantum speed-up provided in utilizing a parameterized quantum circuit. Quantum circuits are constructed using quantum gates. These quantum circuits' parameters can be optimized using an objective function, in a fashion like the training of model parameters in classical models. Recently, QML has emerged in various applications, including understanding nanoparticles, crafting novel materials through molecular and atomic mapping, molecular modeling for drug discovery and medical research, and probing the deeper structures of the human body [9], [10]. Additionally, QML plays a crucial role in enhancing communication networks by enabling advanced data processing and decision-making capabilities at unprecedented speeds and efficiencies. In [11], the authors envision 6G wireless networks as a transformative leap in telecommunications, driven by the integration of machine learning, quantum computing, and quantum machine learning, to enable highly adaptive, efficient, and responsive network orchestration and management.

However, current quantum computers, often referred to as Noisy Intermediate-Scale Quantum (NISQ) devices [12], [13], are still in the early stage and face challenges of a limited number of qubits and inherent noise in quantum circuits. These limitations not only hinder the development of large-scale Quantum Machine Learning (QML) models but also impact the performance and efficiency of these models. The constraints of available quantum hardware, coupled with the inherent noise, limit the scalability of QML models. Scaling up to a larger number of qubits is challenging due to physical constraints and increasing complexity in error correction and control. The inherent noise in quantum circuits, which scales with the depth of QML models, can significantly affect the performance and applicability of QML algorithms. One method to mitigate this issue is to use a relatively shallow model [14]. However, it will lead to a decrease in final performance.

Given the limitations in quantum computing capacity, particularly in scaling up QML models, there's a pressing need to find efficient training methods for QML. Current quantum computing capacity necessitates measures to reduce training efforts required to construct large-scale QML models. Therefore, we explore the concept of transferring the advanced capabilities of current classical machine learning models into the quantum domain, in order to enhance the efficiency of the QML model. The current classical machine learning (CML) models for vision tasks, such as DenseNet [15] and ResNet [16], are state-of-the-art for image classification tasks.

On the CIFAR-10 dataset, ResNet-110 has achieved 6.61 percent error and DensNet-190 has achieved 3.46 percent error. A commonality between these networks was breakthroughs in model design to allow training at larger model depths to be effective. With the abundance of these well-trained, well-performing models, we propose a novel framework that utilizes the knowledge distillation (KD) approach to distilling the knowledge of the CML model into the QML model to improve the performance of QML. KD is originally a model compression technique in which a larger, cumbersome model is used to train a smaller, lightweight model [17]. This is performed by matching output logits between a "teacher" and a "student" model. The teacher model is pre-trained before the distillation, allowing for the ideal performance of the teacher model to guide the student in the learning process. While KD is normally used for model compression, it may be used as a method for transferring learning between models [18]. Therefore, in this work, a well-trained CML model may be used as a teacher to transfer knowledge using KD to a student QML model. Our contributions are listed as follows.

- We propose a novel knowledge distillation-based QNN framework by leveraging the knowledge of CML models to enhance the training efficiency of QNNs. This innovative approach represents a significant advancement in integrating classical and quantum computing methodologies
- By harnessing the strength of the CML models, our approach can not only enhance the performance of the inherently resource-constrained QNN but also increase the stability of the training process.
- Through extensive simulations, we demonstrate that the proposed framework can significantly accelerate the convergence and improve the training performance.

The rest of the paper is organized as follows. In Section II, we introduce the preliminaries of quantum computing basics and quantum neural networks. In Section III, we illustrate the structure of the quantum neural network used in this work. We present the proposed knowledge distillation based QNN framework in IV. We evaluate the performance of the proposed framework in Section V. Finally, we conclude our paper in Section VI.

II. Preliminaries

A. Qubits, Quantum Gates, Measurements & Quantum Circuits

In contrast to the binary nature of traditional bits, a qubit can simultaneously occupy a superposition of the $|0\rangle$ and $|1\rangle$ states. There are multiple technological methods for producing qubits, including superconducting qubits and trapped-ions, among others. In quantum computing, quantum gates play a pivotal role, analogous to classical logic gates in conventional computing. While classical gates operate on bits that take either a 0 or 1 value, quantum gates act on qubits, which can exist in a superposition denoted as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Quantum gates guide the evolution of these qubit states, facilitating quantum operations. Simple gates, such as the Pauli-X (X), Pauli-Y (Y), and Pauli-Z (Z) gates, manipulate individual qubits. In contrast, multi-qubit gates, like the Controlled-NOT (CNOT) or general controlled-U gates, interact with multiple qubits, harnessing the entanglement property unique to quantum mechanics. Multiple gate operations are encapsulated within a quantum circuit. To discern the end state of a quantum algorithm, qubits are measured. Quantum measurement in quantum computing involves collapsing the superposition of a qubit into a definite state, either $|0\rangle$ or $|1\rangle$ [19]. This process is probabilistic, based on the qubit's superposition before measurement. Measuring a qubit fundamentally alters its state, a principle of quantum mechanics. Additionally, measuring one qubit can instantly affect another entangled qubit, a property used in quantum information transfer. Quantum measurement is essential for translating quantum information into classical data, concluding the quantum computation process.

B. Quantum Neural Network

Quantum Neural Networks (QNNs) leverage quantum mechanics to enhance computational capabilities, centering around a Parameterized Quantum Circuit (PQC). This architecture includes a data encoding circuit for translating classical information into quantum states, the PQC itself, and measurement operations to extract outcomes. In data encoding, a common method is angle encoding, which converts classical features into quantum rotations, often with each feature requiring its own qubit. However, multiple features can be encoded on a single qubit through sequential rotations, noting that qubit rotations are periodic with a 2π interval.

The PQC comprises layers of quantum entanglements and parametric single-qubit rotations, known as parametric layers (PLs). Entangling operations create quantum correlations between qubits, and the rotations explore potential solutions. The challenge in QNN design mirrors that in classical computing, involving finding the optimal number of PLs for a given task and balancing computational efficiency and noise resilience.

III. Quantum Neural Networks

Machine learning is a powerful technique used to identify and extract hidden patterns from vast sets of data. This is achieved by adjusting a set of parameters within a mathematical framework to ensure that the predictions or outputs it produces align closely with the actual data. The specific mathematical framework or method used to identify these patterns is termed a "machine learning algorithm."

For a clearer picture, consider a machine learning algorithm as a recipe. This recipe, when followed with a certain set of ingredients (parameters), will produce a specific dish (prediction). Now, to make the dish taste its best (i.e., have the most accurate prediction), one needs to adjust and refine the ingredients' quantity and quality (tuning the parameters). Once we've fine-tuned these ingredients to perfection, our recipe, now tailor-made, is termed a "model".

Taking this a step further into the realm of quantum computing, there's a burgeoning field known as QML. QML leverages the principles of quantum mechanics to potentially speed up complex computations and offer efficient algorithms. With the rapid advancements in quantum hardware and the increasing availability of quantum computers and simulators on cloud platforms, the execution of QML algorithms is becoming more practical and feasible. This opens up a new frontier in machine learning, offering the potential for breakthroughs in processing speeds and algorithmic capabilities.

Neural networks are a specialized category within the vast landscape of machine learning algorithms. At their core, neural networks are composed of multiple layers that process data in a sequential manner, where each layer contributes to refining the input data for the subsequent layer.

Visualize each layer in a neural network as a series of computations. The fundamental computation within each layer involves multiplying the input (often represented as 'x') by a weight matrix (denoted as 'W'), and then adding a bias term (represented as 'b'). This resultant value is then passed through a non-linear function called an activation function (symbolized as '$\phi(\cdot)$'). Mathematically, this operation can be expressed as:

$$f(x) = \phi(Wx + b). \tag{1}$$

The beauty of a neural network lies in its layered structure. The output from one layer doesn't just end there; it serves as the input for the next layer. This chain of transformations, from the first layer to the last, can be expressed as a composition of functions:

$$f(x) = L_m \circ L_{n-1} \circ \ldots \circ L_1(x). \tag{2}$$

This structure allows neural networks to capture complex patterns and relationships in the data. However, for this system to work effectively, it's crucial that the weight matrix 'W' and the bias vector 'b' for each layer are set correctly. These are not set manually but are "learned" from data.

In a quantum context, the idea is to transform this classical expression into its quantum analog:

$$f(|x\rangle) = |\phi(Wx + b)\rangle.$$

There are several key components and challenges:

1) **Data Encoding**: This step involves translating classical data x into quantum states $|\psi(x)\rangle$. Typically, data encoding methods use amplitude encoding or basis encoding to represent classical data in quantum states.
2) **Affine Transformation**: Applying the weight matrix W on the quantum data $|\psi(x)\rangle$ and adding the bias $|b\rangle$ is non-trivial in a quantum setup. While quantum gates can implement various transformations on quantum states, direct implementation of bias addition isn't straightforward, as quantum gates are unitary (and thus linear).
3) **Non-linear Activation Function**: The non-linear activation function $\phi(|\cdot\rangle)$ is even more challenging in the quantum realm. All operations in the qubit model based on available unitary gates are inherently linear, making it a challenge to directly port classical non-linear functions.

In our study, we draw upon the continuous-variable model outlined in the referenced paper [20]. The affine transformation, represented by $Wx+b$, is achieved through the sequential application of the operations $D \circ U_2 \circ S \circ U_1$. Here, each U_k stands for the k^{th} interferometer. The set S consists of m squeezers, while D represents a collection of m displacement gates. The nonlinear activation function, symbolized by $\phi(\cdot)$, is realized using a series of Kerr gates. When the composite operation $\phi \circ D \circ U_2 \circ S \circ U_1$ is applied to a quantum state $|x\rangle$, it yields the target state.

Given these challenges, direct implementations of classical neural networks in quantum circuits aren't simple translations. Quantum neural networks (QNNs) require novel architectures and paradigms that align with quantum mechanics principles.

For instance, non-linearity in QNNs can be introduced in several ways, including:

- Using measurements, which are inherently non-linear, to induce non-linearity in the system.
- Adapting non-linear quantum phenomena, like the behavior of certain quantum systems at specific energy levels.

Utilizing the Pennylane Tensorflow plug-in [21], quantum circuits can be seamlessly integrated as Keras layers alongside classical layers. This integration permits the leveraging of Keras' native loss functions and optimization algorithms for the parameter update process. Predominantly, the models employ the Categorical Crossentropy as the loss function and utilize the Stochastic Gradient Descent for optimization. The Categorical Crossentropy loss function is formulated as:

$$\mathcal{L}^C(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i), \tag{3}$$

where y represents the true categorical labels and $\hat{y}$ denotes the predicted probabilities.

The integration of quantum principles with classical neural networks is an active research area, with constant developments and innovations to overcome these challenges and harness the potential power of quantum computing for machine learning tasks.

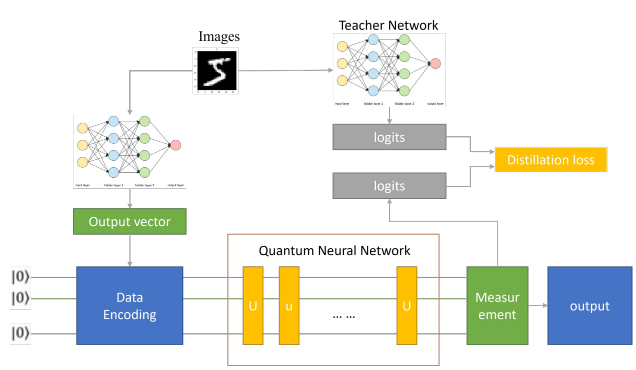

Fig. 1. The flow chart of knowledge distillation based QNN.

IV. Knowledge Distillation Based Quantum Machine Learning Framework

We propose to leverage knowledge distillation (KD) [17], which is a model compression method to transfer the learning capabilities from a complex "teacher" model to a typically simpler "student" model. The overview of our framework is illustrated in Figure 1. In our proposed framework, we assume that the student model is our QML model, and the teacher model is the pre-trained CML model before the distillation, allowing for the ideal performance of the teacher model to guide the student in the learning process. The goal of KD is to force a student QML model to mimic the output of the teacher CML model by utilizing knowledge distilled from the teacher. During KD, the student QML model is trained by using a loss function in which loss is calculated by matching the logits between the student QML model and the teacher CML model. Logits are the nuanced probability distributions generated by the teacher model. Unlike hard targets that only offer label information, logits or soft targets disclose the predicted probabilities across all classes. The logits of i-th class can be represented as $z_i = W_i x + b$, where W_i is the weight corresponding to the i-th class and b is the bias vector. The softmax layer converts logits into the output probabilities for each class as follows,

$$p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}. \tag{4}$$

Hence, to prevent the loss of valuable information, it is often advantageous to employ logits z rather than the predicted probabilities p during the training of the student model.

Given any data sample x in the training dataset, let z_T^i and z_S^i be the logits on i-th class, which are the inputs to the softmax layer, of the teacher CML model and the student QML model, respectively. The goal of our proposed framework is to minimize the difference between the softmax results of the teacher $p_T^i = \text{softmax}(z_T^i/\tau)$ and the student $p_S^i = \text{softmax}(z_S^i/\tau)$ using KL divergence, where τ is the temperature to control the probability distribution over classes, to essentially smooth the probability distribution, thereby capturing the nuanced relationships between different classes as learned by the teacher model. Then, the loss function for KD on the QML student model is

$$\mathcal{L}_g^{KD} = -\tau^2 \sum_i KL(p_T^i, p_S^i). \tag{5}$$

With this loss function, the logits information is transferred from the teacher to the student. Currently, in the field of QML, supervised training with hard targets has proven to be effective. By leveraging the soft targets of a pre-trained CML model as well as the hard targets of a classical dataset, the training performance of a QML model can be further improved.

V. Performance Evaluation

A. Simulation Settings

In the simulation, we utilize the PennyLane and TensorFlow packages for training the QNN models, employing Qiskit for circuit compilations and noise simulations, and leveraging the SciPy package for optimization tasks. All numerical experiments are run on the Intel Xeon W-2195 CPU with 256GB of RAM.

TABLE I
Parameters of Classical Layers

	Sequential Model				
	Flatten	Dense	Dense	Dense	Dense
MNIST	28*28	128	64	32	14

To build the proposed framework, we utilize the CNN as the teacher model. Specifically, the architecture comprises two convolutional layers and two pooling layers, followed by two fully connected layers, The data flow of the framework is structured as follows:

- **Classical network:** Composed of 2 hidden layers, each with 10 neurons using Exponential Linear Units (ELU) as the activation function. The output layer consists of 14 neurons.
- **Data encoding:** The output vector from the classical network is transformed into a quantum state by the circuit, utilizing components such as squeezers, interferometers, displacement gates, and Kerr gates.
- **Quantum network:** Consists of 4 layers of QNN as shown in Fig. 2.
- **Measurement:** The expectation value of the Pauli-X gate, $\langle\phi_k|X|\phi_k\rangle$, is evaluated for each qmode state $|\phi_k\rangle$ for the k^{th} qmode.

After measurement of the quantum circuit, we get the soft target of the training set and can calculate the distillation loss by Eq. (5). Then we combine the loss by:

$$\mathcal{L} = \alpha\mathcal{L}^C + (1-\alpha)\mathcal{L}_g^{KD}, \tag{6}$$

where α is a balance parameter.

B. Training Dataset and Classical Data Encoding

The teacher models are trained for 20 epochs and the student models are trained for 20 epochs using the Adam optimizer. We utilize the MNIST and Fashion-MNIST datasets, both of which contain 60,000 training samples and 10,000 test samples distributed across 10 distinct classes. Each sample in these datasets features 28×28 attributes.

Classical feed-forward neural networks are employed for the preprocessing of image data, primarily to condense the size of the 28×28 image matrices to more manageable, smaller-sized vectors. This resizing aligns with the availability of parameters designated for data encoding. During this process, the image matrices are reshaped into vectors with a size of 28×28 = 784. Subsequently, through the utilization of dense layer operations in Keras, featuring the "ELU" activation function, these vectors are then diminished to even smaller sizes.

Following this reduction, the resultant output vectors undergo encoding as quantum states within the data encoding quantum circuit. This approach ensures that the preprocessed image data is in an optimal state and size for the subsequent

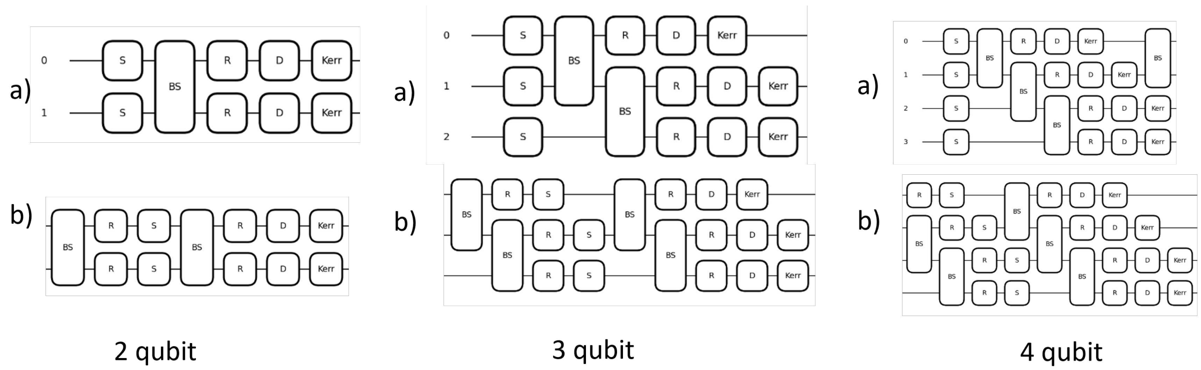

Fig. 2. The structure of variational circuits. a)Encoder b)Layer of quantum circuit

encoding and processing steps in a quantum computing environment.

The output vectors produced by the classical neural network exist in classical states. Subsequently, the quantum data encoding circuit transforms these classical states into quantum states. During the data encoding process, a variety of quantum gates are utilized, including squeezers, interferometers, displacement gates, and Kerr gates. The entries of a classical vector are used as the parameters for these respective parameterized quantum gates, facilitating the precise conversion from classical to quantum states. This conversion is essential for harnessing the advanced computational capabilities of quantum processing in handling data representations and transformations.

C. Analysis on the Performance of the Proposed Framework

The performance of the proposed framework is shown in Fig. 3. In this figure, we compare the model performance of the original QNN model as the blue curve and the proposed knowledge distillation-based QNN model as the green curve on the MNIST and Fashion-MNIST datasets. The performance is evaluated with two sample sizes, 10k and 60k. We find two primary insights from the simulation results. Firstly, there is an enhanced model accuracy, as the plots of accuracy and loss clearly show that the models trained "With Teacher" via knowledge distillation consistently outperform those trained "Without Teacher". This demonstrates a pronounced boost in prediction precision. Secondly, the proposed framework can accelerate the convergence and increase the training stability. The plots on the second row, which show loss metrics, not only highlight the declining "Student Loss" and "Distillation Loss" trends across epochs, thereby attesting to the effectiveness of knowledge distillation, but also exhibit the stability of our models. Their loss descent is consistent and smooth, indicating the robustness and dependable convergence of our approach. In essence, the simulation results demonstrate the effectiveness of the proposed framework in improving accuracy and ensuring robust, stable model performance across diverse datasets.

We also conducted simulations on the MNIST600 dataset to evaluate the impact of different quantum circuit configurations on model performance. The results shown in Fig. 4 provide a comparison of the model accuracy among different amounts of qubit counts and circuit layer complexities. Each sub-figure shows the model accuracy for a specific number of layers in the quantum circuit. It is obvious that with a deeper QML model, the model accuracy increases. However, we also note that using more qubits to encode the training dataset does not always lead to higher accuracy. There is a significant increase in model performance when increasing the number of qubits from two to three. Despite that, further increases do not lead to better accuracy and are comparable to the results with three qubits. Therefore, it is necessary to determine how many qubits should be used to encode the classical data in QML. This figure also demonstrates the performance of the proposed knowledge distillation-based framework in different quantum circuit configurations. The results clearly represent that employing knowledge distillation consistently leads to performance improvements across all tested configurations. This signifies the robustness and effectiveness of the proposed framework, highlighting its potential to enhance performance in various quantum circuit setups.

VI. Conclusion

In this paper, we propose a novel application of knowledge distillation for training QML models. Our framework effectively enhances model accuracy and training stability across various quantum circuit configurations. The experiment results show that utilizing well-trained CML models as teachers is significantly more effective than relying solely on the QML models, since the QML models often suffer from issues such as barren plateaus that have not been adequately addressed. This work bridges the gap between translating the classical model performance to a quantum computing paradigm, thereby enabling faster and more complex computations compared to classical computers.

Acknowledgement

This work is partial supported by U.S. NSF EECS-2045978, CNS-2107216, CNS-2128368, CMMI-2222810, ECCS-2302469, CNS-2107057 and CNS-2318664.

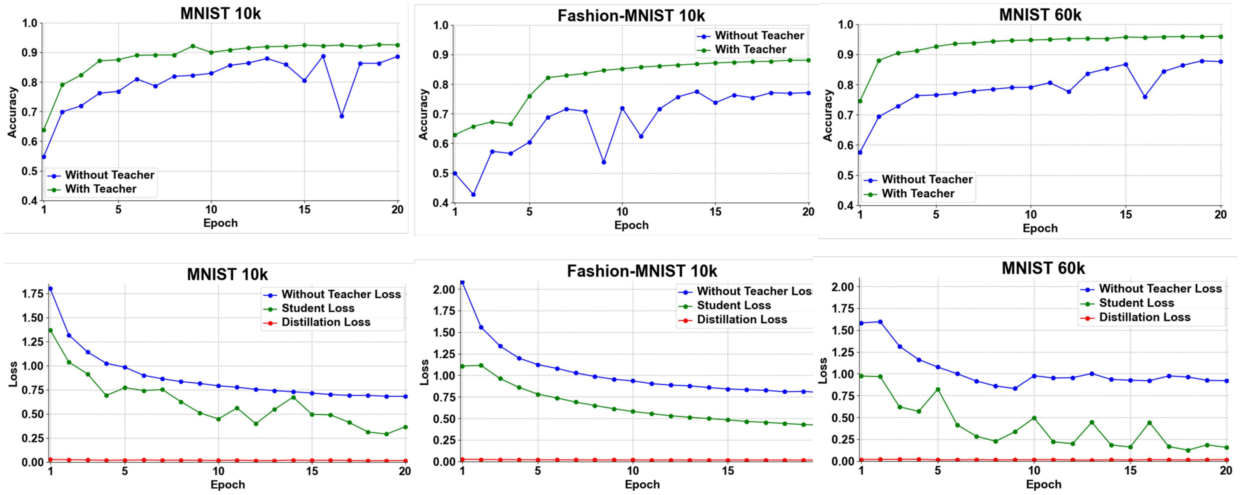

Fig. 3. Comparative analysis of model performance: QNN model with teacher vs. QNN model without teacher on the MNIST dataset.

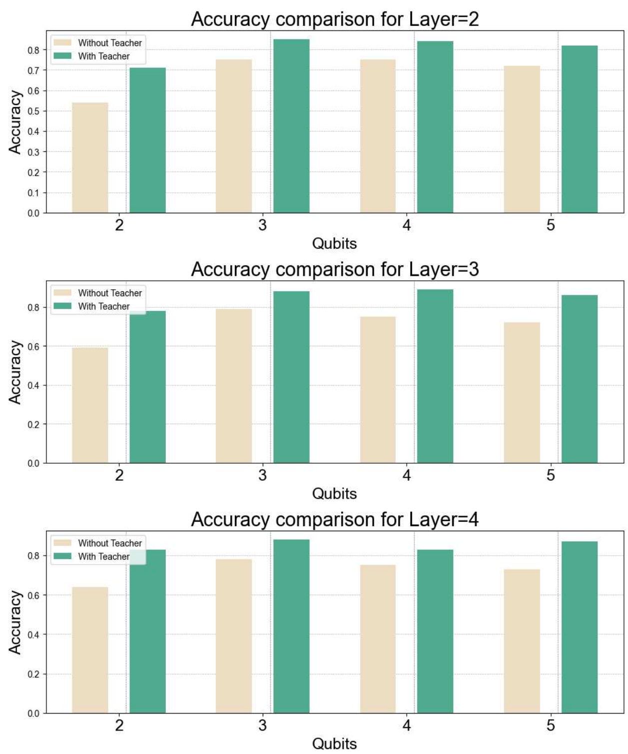

Fig. 4. The impact of different qubits and layers on model performance.

References

[1] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Apr. 2021.

[2] Y. Zhang and Q. Ni, "Recent advances in quantum machine learning," *Quantum Engineering*, vol. 2, no. 1, p. e34, Mar. 2020.

[3] L. Gyongyosi and S. Imre, "A survey on quantum computing technology," *Computer Science Review*, vol. 31, pp. 51–71, Feb. 2019.

[4] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," *Nature Physics*, vol. 17, no. 9, pp. 1013–1017, Jul. 2021.

[5] S. Ubaru, I. Y. Akhalwaya, M. S. Squillante *et al.*, "Quantum topological data analysis with linear depth and exponential speedup," *arXiv preprint arXiv:2108.02811*, Aug. 2021.

[6] M. Cerezo, G. Verdon, H.-Y. Huang *et al.*, "Challenges and opportunities in quantum machine learning," *Nature Computational Science*, vol. 2, no. 9, pp. 567–576, Sep. 2022.

[7] J. Biamonte, P. Wittek, N. Pancotti *et al.*, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[8] H.-Y. Huang, M. Broughton, J. Cotler *et al.*, "Quantum advantage in learning from experiments," *Science*, vol. 376, no. 6598, pp. 1182–1186, Jun. 2022.

[9] W. Guan, G. Perdue, A. Pesah *et al.*, "Quantum machine learning in high energy physics," *Machine Learning: Science and Technology*, vol. 2, no. 1, p. 011003, Mar. 2021.

[10] E. A. Cherrat, I. Kerenidis, N. Mathur *et al.*, "Quantum vision transformers," *arXiv preprint arXiv:2209.08167*, Sep. 2022.

[11] S. J. Nawaz, S. K. Sharma, S. Wyne *et al.*, "Quantum machine learning for 6g communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46 317–46 350, Apr. 2019.

[12] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[13] K. Bharti, A. Cervera-Lierta, T. Kyaw *et al.*, "Noisy intermediate-scale quantum (nisq) algorithms," *arXiv preprint arXiv:2101.08448*, Feb. 2021.

[14] M. Cerezo, G. Verdon, H.-Y. Huang *et al.*, "Challenges and opportunities in quantum machine learning," *Nature Computational Science*, vol. 2, no. 9, pp. 567–576, Sep. 2022.

[15] G. Huang, Z. Liu, L. V. D. Maaten *et al.*, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2017, pp. 2261–2269.

[16] H. Kaiming, Z. Xiangyu, R. Shaoqing *et al.*, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, Nevada, USA: IEEE, Jun. 2016, p. 770.

[17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, Mar. 2015.

[18] M. Phuong and C. Lampert, "Towards understanding knowledge distillation," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 5142–5151.

[19] A. Broadbent and E. Kashefi, "Parallelizing quantum circuits," *Theoretical Computer Science*, vol. 410, no. 26, pp. 2489–2510, Jun. 2009.

[20] S. Choe and M. Perkowski, "Continuous variable quantum mnist classifiers—classical-quantum hybrid quantum neural networks," *Journal of Quantum Information Science*, vol. 12, pp. 37–51, Dec. 2022.

[21] V. Bergholm *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, Nov. 2018.