A Comparative Analysis of Hybrid-Quantum Classical Neural Networks

Kamila Zaman[1,2,*], Tasnim Ahmed[1,2,*], Muhammad Abdullah Hanif[1,2], Alberto Marchisio[1,2], and Muhammad Shafique[1,2],
[1]eBrain Lab, Division of Engineering, New York University Abu Dhabi (NYUAD), Abu Dhabi, UAE
[2]Center for Quantum and Topological Systems (CQTS), NYUAD Research Institute, NYUAD, Abu Dhabi, UAE
{kz2137,tasnim.ahmed,mh6117,alberto.marchisio,muhammad.shafique}@nyu.edu

ABSTRACT

Hybrid Quantum-Classical Machine Learning (ML) is an emerging field, amalgamating the strengths of both classical neural networks and quantum variational circuits on the current noisy intermediate-scale quantum devices [13]. This paper performs an extensive comparative analysis between different hybrid quantum-classical machine learning algorithms, namely Quantum Convolution Neural Network, Quanvolutional Neural Network and Quantum ResNet, for image classification. The experiments designed in this paper focus on different Quantum ML (QML) algorithms to better understand the accuracy variation across the different quantum architectures by implementing interchangeable quantum circuit layers, varying the repetition of such layers and their efficient placement. Such variations enable us to compare the accuracy across different architectural permutations of a given hybrid QML algorithm. The performance comparison of the hybrid models, based on the accuracy, provides us with an understanding of hybrid quantum-classical convergence in correlation with the quantum layer count and the qubit count variations in the circuit.

KEYWORDS

Quantum Machine Learning, Hybrid Quantum-Classical Neural Networks, Quantum Convolutional Neural Networks, Quantum ResNet, Quanvolutional Neural Networks

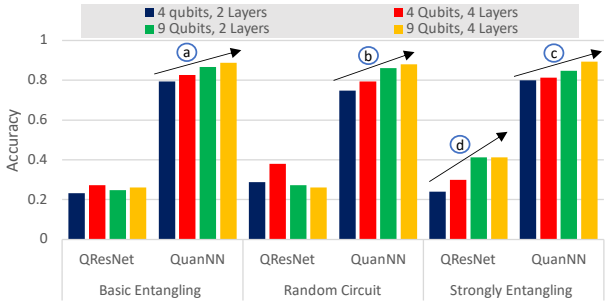

Figure 1: Penny-lane: The arrows indicate an observed pattern suggesting a potential positive correlation of accuracy with the number of layers & qubits in hybrid quantum-classical architectures. Specifically, (a), (b), and (c) consistently exhibit this behavior for the QuanNN across all entangling circuits, while (d) displays a similar effect in the QResNet. While the QuanNN responds to both layer and qubit count, the QResNet demonstrates improved accuracy primarily with an increase in qubits only. Hence, observation at (d), raises questions for further studies on whether the QResNet's accuracy significantly improves with a substantial increase in qubits further or not.

1 INTRODUCTION

Merging Quantum Computing (QC) with Machine Learning (ML) forms the emerging Quantum Machine Learning (QML) paradigm [7, 8, 11, 12]. It represents an excellent opportunity for researchers and industries to make phenomenal discoveries and unravel efficient ways to solve complex real-world problems with significant direction towards practicality and improved accuracy compared to classical systems. QML opens new avenues for the community to discover, build, and align their designs to different levels of the quantum stack. However, the currently developed Noisy-Intermediate Scale Quantum (NISQ) devices [9] have a limited number of qubits, with small-scale resilience to noise, therefore, making it difficult to develop and practically realize the potential of standalone quantum machine learning algorithms. The limitation of NISQ devices has motivated the development of hybrid quantum-classical machine learning algorithms (HQML), which are NISQ-compatible algorithms.

HQML algorithms have emerged as an important paradigm that amalgamates the power of both classical and quantum processing for machine learning tasks, opening new avenues for algorithm and architecture exploration [1]. The most renowned HQML algorithms employ variational quantum circuits, featuring parameterized quantum gates optimized by classical computers to achieve specific goals. These variational quantum circuits in QML, known as Quantum Neural Networks (QNNs), hold significant promise due to their expressiveness and reduced trainable parameters, garnering considerable development interest.

Given the recent breakthroughs of classical algorithms such as Convolutional Neural Networks (CNNs) for image classification tasks, numerous QNN architectures have been developed for classification tasks. However, those architectures are either implemented as basic building blocks or lack the verification of their usability against a benchmark model for classification tasks. *The observations in Figure 1, based on our implementation of permutations of two different algorithms, show that different architectures of the same algorithms have different impacts on classification tasks.* Thus, it is imperative to explore the architectures, applications, and usefulness of QNN algorithms and evaluate their accuracy to identify efficient model configurations that can be deemed as reference benchmarks for future research.

*These authors contributed equally to this work.

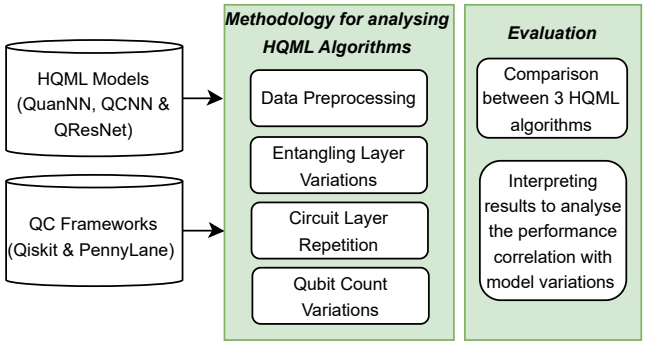

Figure 2: Overview of our novel contributions.

In this paper, we explore three different QNN algorithms amongst the pool of hybrid algorithms, namely, Quanvolutional Neural Networks [5], Quantum Convolutional Neural Networks [2], and Quantum ResNet [6] and perform an extensive comparative analysis. Our methodology first involves identifying efficient architectures amongst the commonly used QNNs and understanding their practical utility for classification tasks. Then we assess how variations of such algorithms impact their accuracy and robustness under architectural permutations of each algorithm. The architectural permutations are based on the implementation of interchangeable variational circuit layers over different qubit counts, varying repetition of the layers, and considering their optimal placement. By implementing the varied models, we evaluate the performance of QNNs based on the accuracy of the training process, which provides us with an understanding of hybrid quantum-classical convergence in correlation with our experimental approach. Such a comprehensive analysis is necessary to establish an understanding of the correlation between circuit architectures, their robustness, and utility in QML.

1.1 Our Novel Contributions

An overview of our novel contributions is shown in Figure 2. Their brief descriptions with key features is presented below.

- We propose a methodology to investigate QNN models' circuit permutations and their effect on the accuracy for classification tasks. (**Section 3**)
- We analyze the effect of different circuit layer variations[1], namely, Random Circuit, Basic Entangling, and Strongly Entangling. (**Section 4.2**)
- We investigate the repetition of different layers to analyze the effect of circuit depth and complexity. (**Section 4.3**)
- We test the scalability of the algorithms by varying qubit counts (**Section 4.4**)

2 SELECTED HYBRID QML ALGORITHMS

An overview of the hybrid QML algorithms is shown in Figure 3. A brief description of the key features of each algorithm and their implementations is presented in the following paragraphs.

[1]The circuit variations are studied only for QNN models that support different types of entanglement.

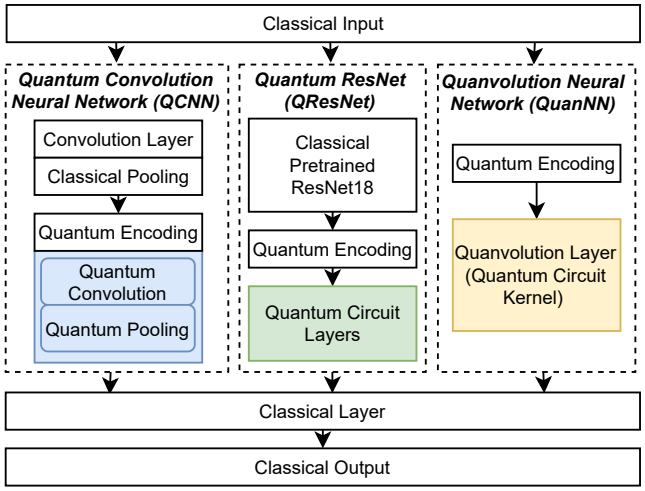

Figure 3: Pipeline of the implemented hybrid QML algorithms. Each model utilizes a classical fully connected layer to transform quantum circuit measurement into classification probabilities. In the QCNN, classical convolutional and pooling layers are used for image downsizing to match the qubit count of a circuit.

2.1 Quanvolutional Neural Networks

The Quanvolutional Neural Network (QuanNN) is an innovative hybrid quantum-classical architecture developed in [5], which enhances the capabilities of classical CNNs by harnessing the potential of quantum computation. This architecture introduces a new type of transformation layer called the quanvolutional layer, akin to classical convolutional layers, composed of multiple quanvolutional filters which locally transform input data, extracting valuable features for classification. These filters correspond to a certain circuit design, which can either be generated randomly or based on a specific entanglement, namely Basic Entangling or Strongly Entangling. The reason we chose the QuanNN as one of our benchmarking models is because of its generalizability which can be achieved by adhering to the following conditions: specifying an arbitrary integer number of quanvolutional filters in each layer, stacking multiple quanvolutional layers in the network, defining layer-specific parameters like encoding method, entanglement, and average quantum gates per qubit in the quantum circuit. To formalize classical data transformation using quanvolutional filters:

(1) Start with a single filter q operating on subsections ux of dataset images.
(2) Encode ux into an initialized state ix using an encoding function e.
(3) Apply the quantum circuit to ix, producing an output quantum state ox.
(4) Decode ox to ensure consistent outputs, resulting in the final decoded state fx.
(5) This entire process, denoted as the "quanvolutional filter transformation" Q, is $fx = Q(ux, e, q, d)$.

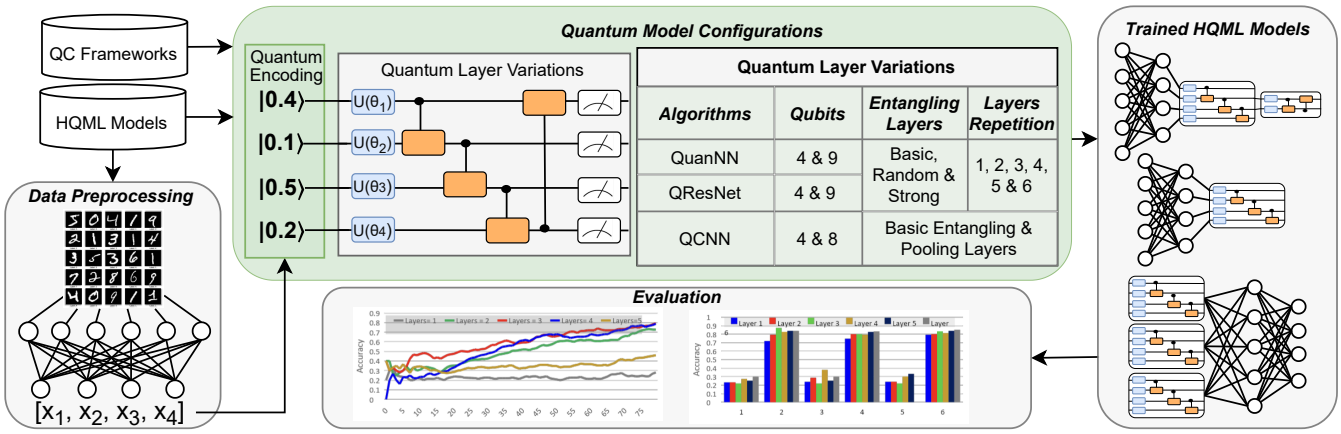

Figure 4: Overview of our comparative analysis methodology.

2.2 Quantum Convolutional Neural Networks

Similar to the QuanNN, the Quantum Convolutional Neural Network (QCNN) is a QML algorithm inspired by CNNs that was introduced in [2]. Unlike the QuanNN, the QCNN does not have room for lots of circuit design variations. The QCNN has a fully quantum implementation of convolutional and pooling layers. We chose this architecture in our experiments because it is one of the state-of-the-art QML models and the classical counterpart represents the state-of-the-art in classical image recognition. Moreover, it is important to note that the QCNN presented in [2] makes use of only $O(log(N))$ variational parameters for input sizes of N qubits. This allows for its efficient training and implementation on realistic, near-term quantum devices.

The basic structure of the QCNN includes an input encoding circuit layer, a convolutional circuit layer, a pooling circuit layer, and a circuit measurement layer, each composed of parametric quantum gates. It is categorized as an HQML algorithm because it uses classical optimization techniques to update the parameterized gate weights. In the quantum convolutional and pooling layers, the interactions between quantum bits can effectively extract features from the input data based on the types of gates and their placement in each layer. Given the current NISQ devices, the QCNN is limited in terms of scalability and can only be implemented with a small number of qubits. Therefore, it requires classical layers for downsizing large inputs to match the qubit size. In our implementation, we employ single classical convolution and pooling layers for input downsizing, without loss of key features.

2.3 Quantum ResNet

Our decision to employ the Quantum ResNet (QResNet) algorithm was inspired by [6], which introduced a hybrid quantum-classical strategy for deep residual learning. The primary challenge in this approach is to establish a connection between the residual block structure and the quantum processing layers. In our specific context, our focus lies in experimenting with various quantum layer types. We delve into an analysis of potential methods and variations that combine residual learning with quantum machine learning. Notably, the work in [4] emphasizes that experimental outcomes demonstrate the QResNet's superior capability of learning an unknown unitary transformation and its enhanced robustness with noisy data, compared to state-of-the-art methods. These findings motivate us to further explore and fine-tune this promising architecture. Our choice of the QResNet stems from our desire to test this HQML architecture and its impact on pre-trained classical models.

3 COMPARATIVE ANALYSIS METHODOLOGY

With the aim of understanding hybrid-quantum classical neural networks in-depth, we divide our problem into interdependent experimental sections as shown in Figure 4. The sectioning allows us to identify areas of possible efficient implementation and improvements for the HQML algorithms discussed in Section 2.

3.1 Overview of Analyses

Our experimental sections are summarized in the *Quantum Layer Variations* table in the *Quantum Model Configurations* group of Figure 4. For each algorithm, we analyze the impact of different architectural permutations based on:

- ***Entanglement variation of quantum circuit***: The predefined circuit structure of QuanNN and QResNet enables us to change the entanglement type of the circuit. Each circuit has its own orientation of CNOT gates and parameterized corresponding to the strength of their entanglement. Whereas for QCNN, there is no room for changing the entanglement type of the circuit, since it follows the structure defined in [2].
- ***Layer count variation***: For each algorithm, a circuit can be applied multiple times to an input, which corresponds to understanding how the varying depth of a quantum circuit affects the model accuracy.
- ***Qubit count variation***: The strength and capability of a circuit depends on the number of qubits it has. Hence, in our experiments, we vary the qubit counts in the architecture to analyze how the variation correlates with the models' learning curve and accuracy.

3.2 Comparison Metrics

We intentionally made our experimental setup simple and precise, to gain more understanding about how different independent parameters of a circuit's design of our selection algorithms can influence their accuracy. To understand the contribution and convergence behaviours of the aforementioned variations, we focus on analyzing the correlation between a model's accuracy by studying their classical accuracies in relation to the learning curve attained over the training progress.

4 EVALUATION AND DISCUSSION

4.1 Experimental Setup

This paper investigates the design variations of the quantum components of HQML architectures. Hence, to ensure a fair comparison, we use a uniform classical optimization environment for both the classical and the quantum layers as specified in Table 1. In the HQML architectures of our experiment, all the algorithms have a classical layer after the quantum layer to convert the quantum measurement values into classical probabilities. However, the QCNN has a classical layer before the input encoding for image downsize. As for the quantum layers, PennyLane and Qiskit frameworks provide PyTorch integration modules, which convert a quantum layer into PyTorch trainable layers and perform classical optimization of the gates based on the training hyperparameters that we specified in Table 1. In our experiments, to ensure that the output of a circuit corresponds to the minimum number of qubits of our qubit count pool, we use a subset of the MNIST dataset [3] consisting of only 4 classes, [0, 1, 2, 3]. Thus, the last classical layer of our models consists of only 4 neurons.

For analyzing the HQML components of our architecture, three different kinds of variation are applied to the quantum layer of an algorithm, as shown in Figure 4:

- ***Entanglement variation of quantum circuit***: For algorithms that allow different entanglement orientation, we vary their architectures with Random Circuit (RC), Basic Entangling (BE), or Strongly Entangling (SE) circuit.
- ***Layer count variation***: Each quantum layer can have numerous repetitions of a circuit. In our implementation, we repeat the circuit from 1 to 6 times. *Note: the QCNN has a fewer layer variation compared to the QuanNN and QResNet because number of layer* $= \sqrt{qubits}$.
- ***Qubit count variation***: QuanNN and QResNet circuits follow a square filter-like structure where the qubits must be a square number. Hence, we experiment with 4 and 9 qubits. On the other hand, the circuit of the QCNN must be an even number that can be reduced in half at each pooling step. Hence, we experiment with 4 and 8 qubits.

Note: We implemented all these pipelines in Qiskit [10] & PennyLane [1] to enable the support for both QML frameworks. They are purposely presented separately (and can be identified in boldened figure captions) to avoid cross-tool comparison which is out of the scope of this paper.

Table 1: Training Environment Specifications

Algorithm	Experiment Name
Software FrameWork	`PennyLane(PL), Qiskit(QK)`
Back-End Simulator	`lightning.qubit(PL), qasm_simulator(QK)`
Back-End Machine	`NVIDIA RTX 6000 Ada`
Deep-Learning Interface	`Pytorch`
Data-set	`MNIST` [3]
Training Samples, Testing Samples	PL: (`100`, `100`), QK: (`500`, `100`)
Epoch, Batch-Size, LR	`5`, `5`, `0.01`

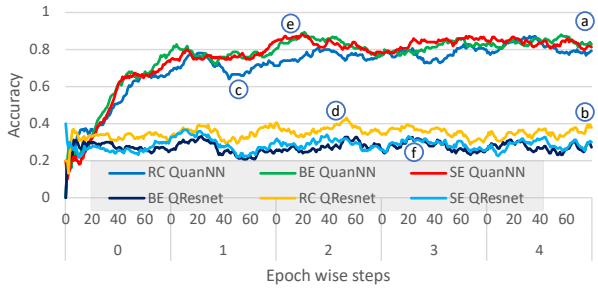

Figure 5: *PennyLane:* RC: Random Circuit, SE: Strongly Entangling, BE: Basic Entangling. Circuit Variation across QuanNN and QResNet.

4.2 Results: Entangling Circuit Variations

Figure 5 shows the accuracy evolution over training epochs for QuanNN and QResNet at different entanglement settings (RC, BE & SE) for PennyLane implementation. As demonstrated by the accuracy and convergence gap between the two algorithms, it is evident that QuanNN learns significantly better than QResNet (see labels a and b).

Further analyses show that the RC QuanNN (label c) acquires lower accuracy than the other entanglement settings. Whereas, for QResNet, the RC model (label d) achieves the highest accuracy compared to the SE and BE settings. Towards the end of the training, we can see that the accuracy of the QuanNN, for all three entanglements, converges to around 80%, with the SE QuanNN having the highest accuracy. Following the traces for BE and SE circuits, for both QuanNN and QResNet (labels e and f, respectively), it is noticeable how the traces are similar in each algorithm. On the other hand, RC varies in learning compared to SE and BE in both algorithms. In the RC QuanNN (label c), the accuracy improves at a slower pace compared to SE and BE, but soon converges closer towards the end. As for the RC QResNet, the rate of learning is similar to BE and SE, but with a higher accuracy throughout.

Figure 6 shows the same evolution graphs as Figure 5, but for Qiskit models. It is obvious that even for Qiskit implementations the accuracy gap between QuanNN and QResNet models are same as PennyLane (labels a and b). It is quite evident, that QuanNN models in Qiskit have different learning curve, with a great accuracy jump after the first epoch (label c).

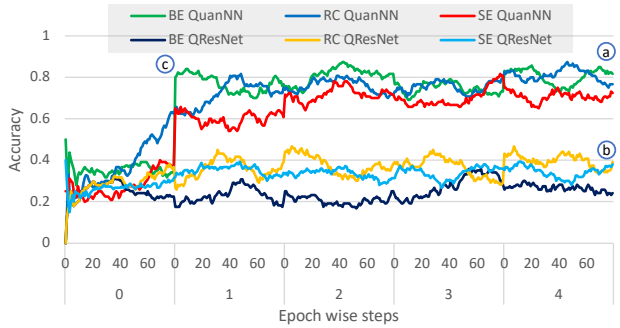

Figure 6: *Qiskit:* RC: Random Circuit, SE: Strongly Entangling, BE: Basic Entangling. Circuit Variation across QuanNN and QResNet.

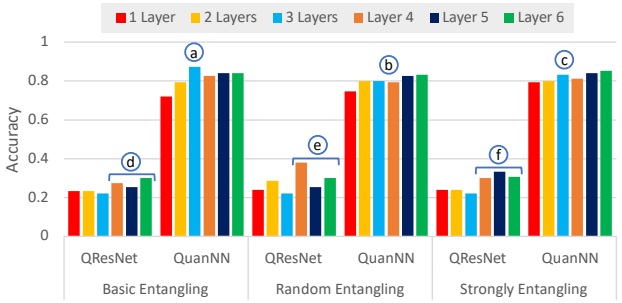

Figure 7: *PennyLane:* Layer Variation across different Entangling Circuits for QuanNN and QResNet. Asides minor perturbation, an overall trend of improving accuracy is observed with the increasing number of layers, for every type of circuit.

Note: the QCNN is not presented in this comparison because only the Basic Entangling circuit can be implemented in this architecture.

4.3 Results: Layer Count Variations

Figure 7 shows the accuracy of the QResNet and QuanNN models with varying number of layers implemented in PennyLane. From the results, we can gauge that there is minor difference between 2, 3, and 4 layers for QResNet models (see labels b and c). Moreover, adding more than 4 layers to the QuanNN does not always contribute to increasing the accuracy (see labels a and e). For the QResNet, it can be observed that the accuracy increases quite significantly for the QResNet models with 4, 5, and 6 layers (see d, e, and f), more specifically for circuit with 4 layers. Based on this observation, we conducted the remaining experiments with 4 layers in places where no layer variation was involved. Even though the accuracy of the QResNet is lower than the QuanNN, the algorithm has some potential because it can improve the accuracy for increasing the number of layers (see labels d and f).

In Figure 8, we can observe that the QuanNN and QResNet models implemented in Qiskit do not have an overall trend with respect to increasing the number of layers, which indicates that

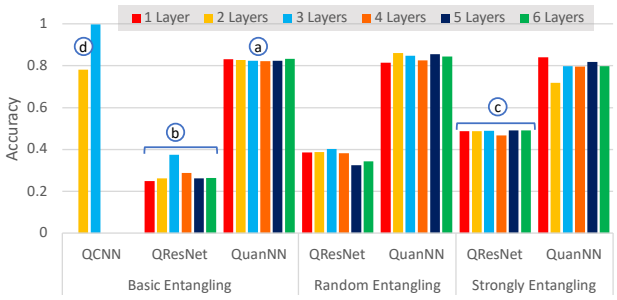

Figure 8: *Qiskit:* Layer Variation across different Entangling Circuits for QuanNN and QResNet. No overall trend observed with increasing number of layers.

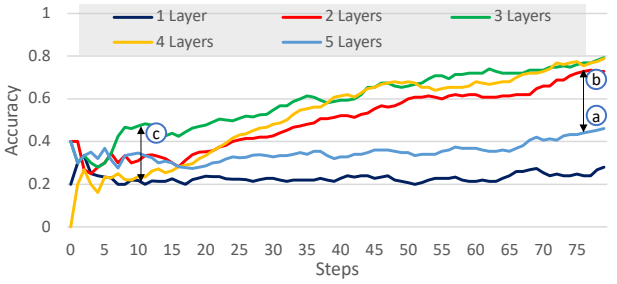

Figure 9: *PennyLane:* Initial learning behavior of Layer Variations, for the QuanNN with Basic Entangling. Having 3 layers shows a clear advantage over the other configurations in the first epochs, but the accuracy of the 4-layer QuanNN, despite being slow at first, catches up quickly with the 3-layer counterpart and converges to better results in the majority of the experiments.

the circuit depth does not have enough impact on these models' accuracy (see labels a, b and c). However, it is quite evident for QCNN models that increasing the number of layers has a positive correlation with the accuracy (see label d).

Figure 9 shows the learning curve for the QuanNN with Basic Entangling for different layers implemented in PennyLane. For 1 layer, the learning curve is constant throughout and little improvement is observed (see label c). For more than one layer, we can observe that the learning improves with increasing the number of layers. However, the optimal peak is until 4 layers (see label b), because beyond that the learning curve drops to being close to the 1-layer curve (see label a). Nonetheless, a steady and gradual improvement in accuracy is observed for more than one layer.

For the models implemented in Qiskit, as shown in Figure 10, we can observe that QCNN models with 3 layers have the highest accuracy towards the end of the first epoch (see label a), however, with a very steadily increasing learning curve (see label c). As for the QuanNN models, although the initial accuracy is lower than the QCNN 3L, the models have a rapidly increasing learning curve. Moreover, we can see that the learning curve for the QuanNN models peak only until 5 layers (QuanNN 5L), because beyond that,

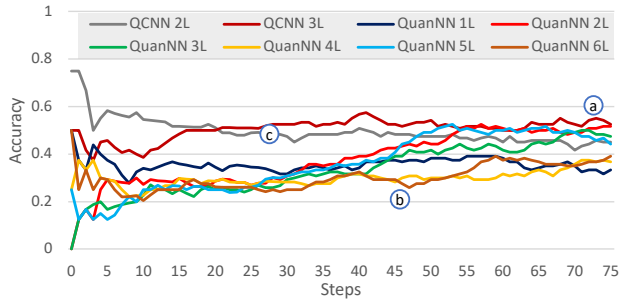

Figure 10: Qiskit: Initial learning behaviour of different Basic Entangling layer variation counts.

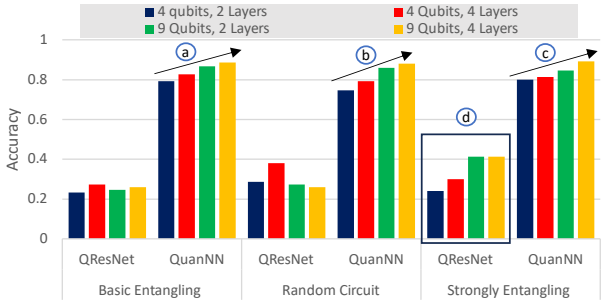

Figure 11: *PennyLane:* Qubits Count Variation across different quantum entangling circuits for QuanNN and QResNet. It can be noted that there a positive correlation between number of qubits in QuanNN model and its accuracy.

for 6 layers, a very slow and steady improvement is observed (see label b).

4.4 Results: Qubit Count Variations

Figure 11 illustrates the results for the QuanNN and QResNet models with different qubits counts, implemented in PennyLane. A clear pattern can be observed for the QuanNN models, where there is a consistent positive correlation between the number of qubits and model accuracy, with increasing layer counts (see labels a, b, and c), indicating that the models learn better with increasing number of qubits. On the other hand, the QResNet models with Basic Entangling circuit and Random Circuit do not have any significant trend. However, the QResNet with Strongly Entangling circuit shows an increase in accuracy with increasing its qubit count (see label d), which might lead one to derive that QResNet with Strongly Entangled circuit has a positive learning capability with an increased number of qubits.

Figure 12 depicts the variation of models implemented in Qiskit with regards to varying qubit count. Firstly, we can notice that the QCNN accuracy improves with increasing the number of qubits, which indicates that the QCNN learns better with more qubits (see label a). A similar trend is observed for the QuanNN and QResNet, but only for the models with Random Circuit (see labels b and c). From this analysis, we can gauge that, changing the number

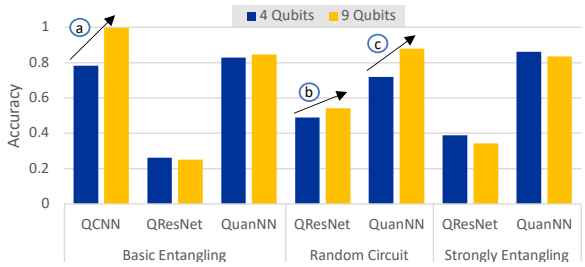

Figure 12: *Qiskit:* Qubit Count variations across different entangling circuits for QuanNN, QResNet, and QCNN. The QCNN accuracy positively correlates with the increasing number of qubits.

of qubits in Qiskit models have a varying impact for different entagnling circuits.

4.5 Result Discussion & Findings

We observed from the results that different variations combined together have a compounding effect on the model accuracy, as it is not trivial to determine the best set of permutations for the given HQML algorithm. For instance, the QResNet had a stagnant improvement across most models. On the other hand, if implemented with Strongly Entangling and 9 qubits, then the accuracy improves drastically. However, in most cases, the accuracy gain achieved when increasing the number of qubits comes with the cost of increased execution time.

In addition, varying the number of layers with different entangling circuits can impact the accuracy differently, as we observed in our results above. In some cases, the increasing accuracy with increasing depth indicates that there is potential in applying more quantum layers at the initial stages of HQML models instead of classical convolutional layers. However, as observed in Figure 11, we can see that the accuracy of HQML models keeps increasing as the model becomes more complex (i.e., with a higher number of layers and qubits). However, the more complex a model becomes, the more time it takes to execute.

All of our findings prove that the best circuit configuration cannot be identified easily for a given algorithm when keeping in mind all the constraints.

5 CONCLUSION

In our work, we studied the accuracy variation of 3 different algorithms, QuanNN, QCNN, and QResNet, by configuring the quantum architecture of each circuit by varying the layer count, qubit count, and the type of entangling circuit. These 3 categories of variation enabled us to design different permutations of circuits for a given algorithm and study their effect on their accuracy.

This study has granted us valuable insights into the establishment and exploration of avenues for optimizing hybrid quantum-classical neural network architectures. We have effectively advanced our pipelines toward practical improvements, using foundational structures for experimentation. The detailed deconstruction of the architecture offers transparency into the parameters that influence the performance of quantum circuits

within neural networks. Such an in-depth analysis serves as the building block for designing efficient HQML architectures and paves the way to develop robust HQML algorithms applicable in the NISQ era, while also considering design aspects that can smoothly transition into Fault Tolerant Quantum Computers.

ACKNOWLEDGMENTS

This work was supported in part by the NYUAD Center for Quantum and Topological Systems (CQTS), funded by Tamkeen under the NYUAD Research Institute grant CG008.

REFERENCES

[1] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, Juan Miguel Arrazola, Utkarsh Azad, Sam Banning, Carsten Blank, Thomas R Bromley, Benjamin A. Cordier, Jack Ceroni, Alain Delgado, Olivia Di Matteo, Amintor Dusko, Tanya Garg, Diego Guala, Anthony Hayes, Ryan Hill, Aroosa Ijaz, Theodor Isacsson, David Ittah, Soran Jahangiri, Prateek Jain, Edward Jiang, Ankit Khandelwal, Korbinian Kottmann, et al. 2022. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968

[2] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. 2019. Quantum convolutional neural networks. *Nature Physics* (2019).

[3] Li Deng. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* (2012).

[4] E. Ghasemian and M. K. Tavassoly. 2022. Hybrid classical-quantum machine learning based on dissipative two-qubit channels. *Scientific Reports* (2022).

[5] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. 2019. Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits. arXiv:1904.04767

[6] Yanying Liang, Wei Peng, Zhu-Jun Zheng, Olli Silvén, and Guoying Zhao. 2021. A hybrid quantum-classical neural network with deep residual learning. arXiv:2012.07772

[7] Stefano Markidis. 2023. Programming Quantum Neural Networks on NISQ Systems: An Overview of Technologies and Methodologies. *Entropy* (2023).

[8] Fabio Valerio Massoli, Lucia Vadicamo, Giuseppe Amato, and Fabrizio Falchi. 2022. A Leap among Quantum Computing and Quantum Neural Networks: A Survey. arXiv:2107.03313

[9] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* (2018).

[10] Qiskit contributors. 2023. Qiskit: An Open-source Framework for Quantum Computing.

[11] N Schetakis, D Aghamalyan, P Griffin, and M Boguslavsky. 2022. Review of some existing QML frameworks and novel hybrid classical-quantum neural networks realising binary classification for the noisy datasets. *Sci. Rep.* (2022).

[12] Maria Schuld and Nathan Killoran. 2019. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters* (2019).

[13] Kamila Zaman, Alberto Marchisio, Muhammad Abdullah Hanif, and Muhammad Shafique. 2023. A Survey on Quantum Machine Learning: Current Trends, Challenges, Opportunities, and the Road Ahead. arXiv:2310.10315