Preserving Data Locality in Multidimensional Variational Quantum Classification

Mingyoung Jeng, Md. Alvir Islam Nobel, Vinayak Jha, David Levy, Dylan Kneidel, Manu Chaudhary, SM Ishraq Ul Islam, and Esam El-Araby

Department of Electrical Engineering and Computer Science (EECS), University of Kansas (KU)

Introduction and Motivation

- Classical convolutional neural networks (CNNs) are effective at exploiting data locality in machine learning applications such as image classification.
  - Preserving data locality allows CNN models to reduce the number of training parameters, and hence their training time, while achieving high classification accuracy.
- Existing quantum machine learning (QML) methods don't effectively leverage data locality in multidimensional features.
- We propose a variational quantum classification technique which facilitates:
  - Multidimensional quantum convolution while preserving data locality.
  - Quantum pooling based on quantum Haar transform (QHT).
- We experimentally demonstrate the advantage of our method in comparison to existing classical and quantum techniques for image classification in staple multidimensional datasets using state-of-the-art quantum simulations.

Background

Fundamentals of Quantum Computing

- Quantum computers leverage superposition and entanglement of quantum states for advantage over classical computers in certain workloads.
- Near-term noisy-intermediate-scale-quantum (NISQ) hardware possesses strict decoherence constraints where quantum states break down after a certain amount of time.
- Representation of an n-qubit quantum statevector:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i|i\rangle = \begin{bmatrix} c_0 \\ \vdots \\ c_{2^n-1} \end{bmatrix} \qquad \langle\psi|\psi\rangle = \sum_{i=0}^{2^n-1} |c_i|^2 = 1 \qquad c_i \in \mathbb{C}$$

- Quantum operations act on quantum states and can be represented as unitary matrices or quantum "gates".
- All quantum gates can be decomposed into fundamental single-qubit rotation and two-qubit CNOT gates.
- Quantum circuits must optimize circuit depth and gate count due to decoherence and gate errors.

Amplitude Encoding via Arbitrary State Synthesis

- Data values can be encoded into the probability amplitudes of the statevector, with the basis state representing positional information.
- Data must be normalized to generate a valid statevector.
- Multidimensional data can be mapped to the 1-D statevector using column-major ordering.
- If the size of a dimension is not a power of 2, it must be padded with zeroes to the next largest power of 2.
- An arbitrary state synthesis operation, e.g., classical-to-quantum (C2Q), can be used for data encoding from the ground state.

$$U_{C2Q}|0\rangle^{\otimes n} = |\psi_0\rangle \qquad U_{C2Q} = \begin{bmatrix} |\psi_0\rangle & |\times\rangle & \cdots & |\times\rangle \end{bmatrix} \quad |\times\rangle = \text{"don't care"}$$

Quantum Machine Learning with Variational Algorithms

- Quantum optimization is not feasible due to decoherence constraints.
- Current state of the art is quantum-classical hybrid algorithms, such as variational algorithms.
  - Static circuit gate layout is parameterized by rotation gates.
  - Circuit parameters are trained with classical optimization methods, e.g., gradient descent.

Related Work

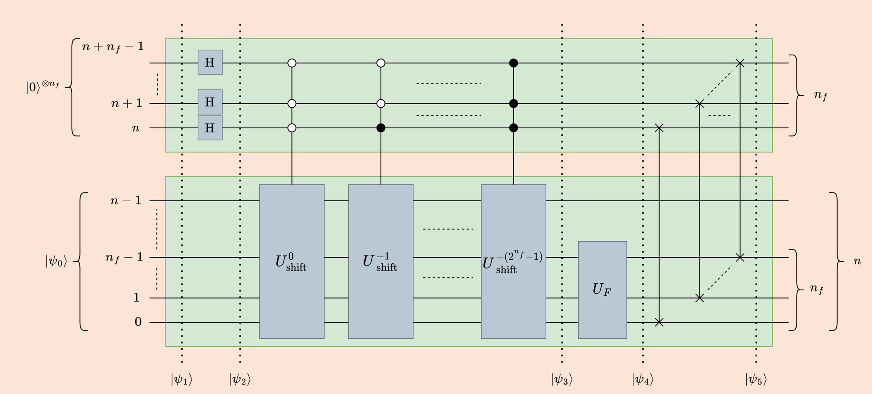
Quantum Convolutional Neural Networks (QCNNs) [1]

- Structure inspired by CNNs, with "convolutional," "pooling," and "fully-connected" layers.
- "Convolution" layers based on locality of logical virtual qubits.
  - No emphasis on data locality or the convolution operation.

Quanvolutional Neural Networks [5]

- Leverages classical preprocessing to divide multidimensional input data into data-local windows.
- Forgoes significant quantum advantage from parallel processing.
- Introduces significant overhead from preprocessing and large number of quantum circuit iterations.
  - Classical-to-quantum (C2Q) data encoding and quantum-to-classical (Q2C) data decoding are slow operations.
- Inherently a hybrid technique intended to accelerate CNNs, not for independent QML.

One-Dimensional Quantum Convolution

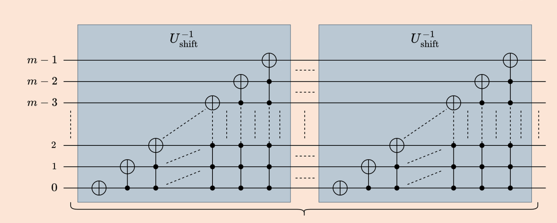


1-D Quantum Convolution Circuit

The general convolution operation can be broken down into the following basic operations.

Shift:

- Generates shifted (unity strided) replicas of the input data
- Uses additional "filter" qubits and controlled quantum decrementers



Quantum Decrementer by k

$$U^k_{\text{shift}} = \prod_{i=0}^{k-1} U^{-1}_{\text{shift}}$$

Multiply-and-accumulate:

- Filter operation is applied to all data replicas in parallel.
- For filters of $N_f = 2^{n_f}$ terms, filter is composed of N_f coupled filters (row vectors) with $N_f - 1$ degrees of freedom.

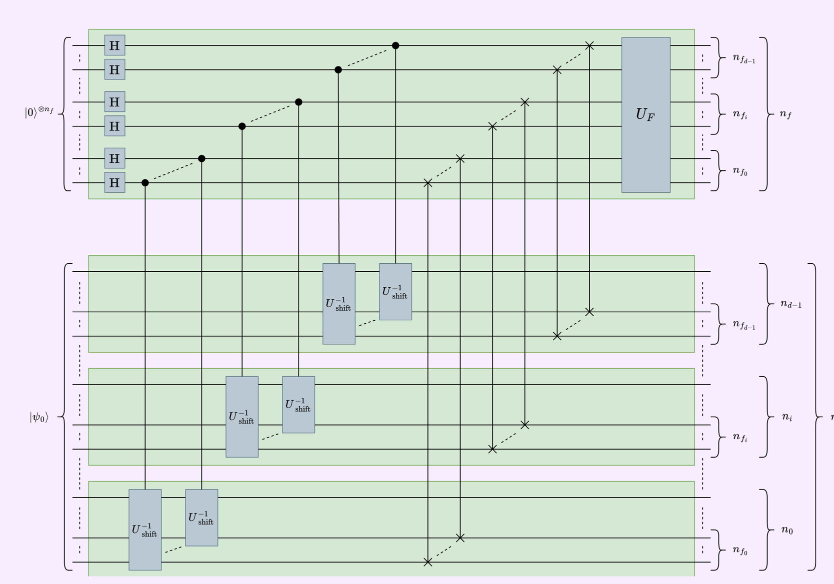
$$U_F = \begin{bmatrix} \langle F_0| \\ \vdots \\ \langle F_{2^{n_f}-1}| \end{bmatrix}$$

- Define a specific filter using inverse arbitrary state synthesis and classical-to-quantum (C2Q) encoding

Data Rearrangement:

- Groups fragmented data into one contiguous output data
- Uses quantum permutations (SWAP gates)

Multi-Dimensional Quantum Convolution



Multi-Dimensional Quantum Convolution Circuit

- With multidimensional data, *shift* and *data rearrangement* operations can be "stacked" and performed in parallel.
- Figure reverses *multiply-and-accumulate* and *data rearrangement* steps with an identity for visual clarity.
- Using a pyramidal cascade of quantum decrementers instead of multiplexed quantum decrementers reduces number of U^{-1}_{shift} operations from $\sum_{j=0}^{2^{n_f}-1} j = 2^{n_f-1}(2^{n_f}-1)$ to n_f.
- Overall, the circuit depth complexity is $\mathcal{O}(n_{f_{\max}} n^2_{\max} + 2^{n_f})$, where $n_{\max}$ qubits represent the largest data dimension and $n_{f_{\max}}$ qubits represent the corresponding filter dimension.

Simulation of Quantum Convolution on (512×512×3) Images

Original Image

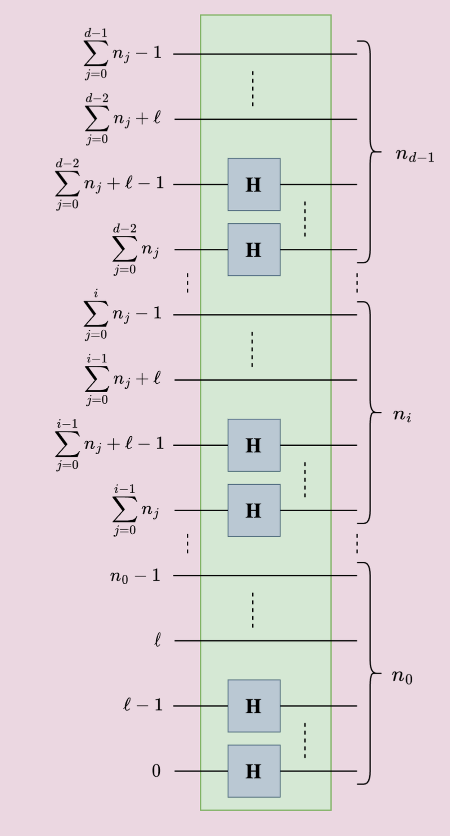
(3×3) Sobel-X

(3×3) Sobel-Y

Quantum Pooling via Quantum Haar Transform

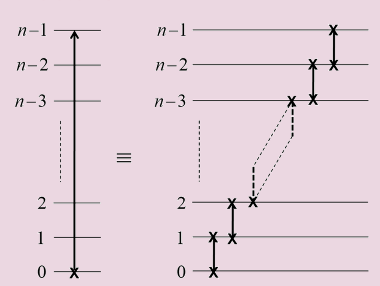
Overview of the Quantum Haar Transform (QHT)

- Wavelet transforms decompose data into spatio-temporal components while preserving data locality.
- Commonly used in image processing for dimension reduction.
- We leverage the pyramidal variant of the Haar transform, the first and simplest discrete wavelet transform.
- The quantum equivalent of the classical Haar transform, the quantum Haar transform (QHT) decomposes amplitude-encoded data into low- and high-frequency components.

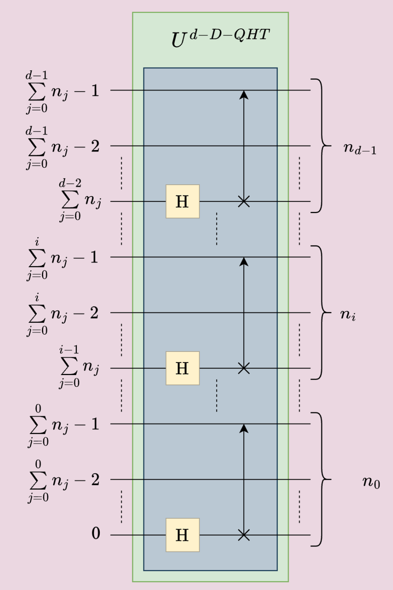


Multi-level Multidimensional Quantum Haar Transform Circuit (No Rearrangement)

- Applying Hadamard gates to the least-significant qubits representing a particular dimension of data performs QHT decomposition of components.
- One level of decomposition for each Hadamard gate per dimension.
- Each level of decomposition reduces the size of the corresponding dimension by a factor of 2.



Rotate-Right (RoR) Gate

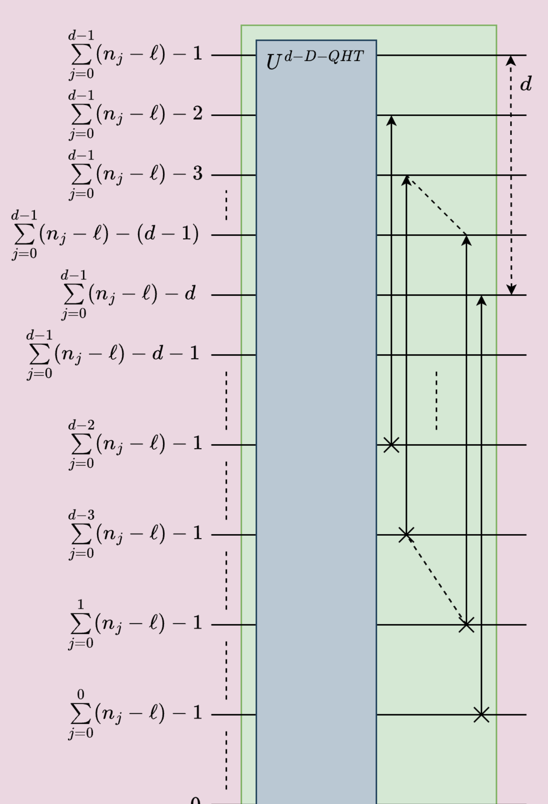


1-level Multidimensional Quantum Haar Transform Circuit (With Rearrangement)

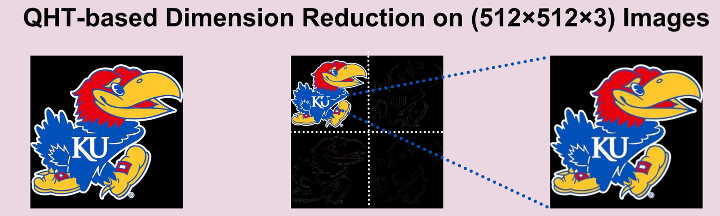
- Data rearrangement groups the fragmented low- and high-frequency components.
- In the quantum circuit, rearrangement can be done with rotate-right (RoR) gates constructed from SWAP gates.
- Can also be done by classical register remapping/indexing during quantum-to-classical (Q2C) data decoding to conserve depth.

Pyramidal Structure of Multi-level QHT-based Dimension Reduction

- Adding / extending the data rearrangements with interleved permutations can be used to create a pyramidal structure.
- Allows clearer representation of quantum operations applied to dimension-reduced data.
- With SWAP optimizations, entire operation requires constant depth.



QHT-based Dimension Reduction on (512×512×3) Images

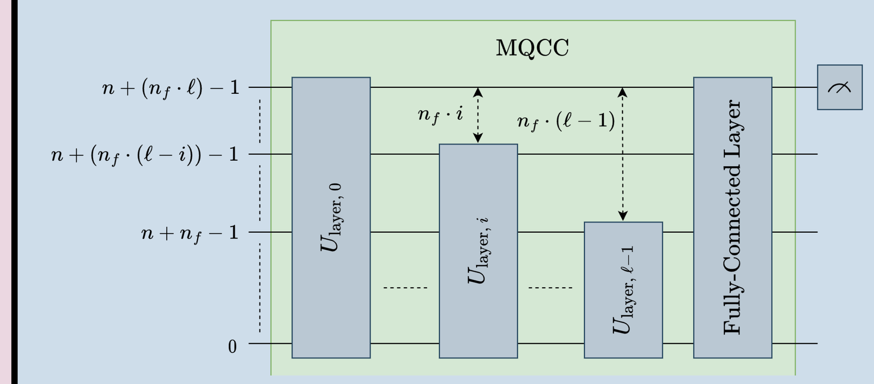


Original Image

1-Level Decomposition

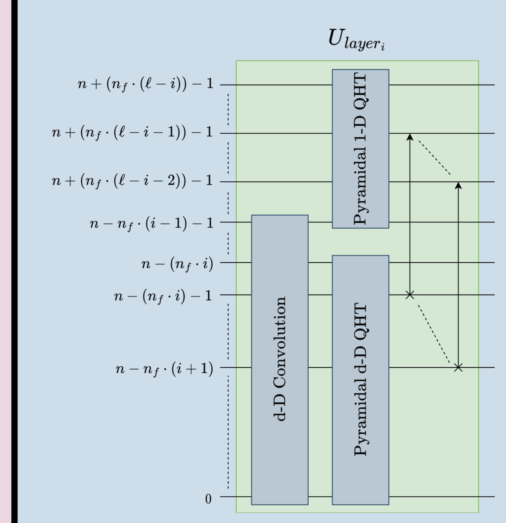
Low-frequency Component

Multidimensional Quantum Convolutional Classifier



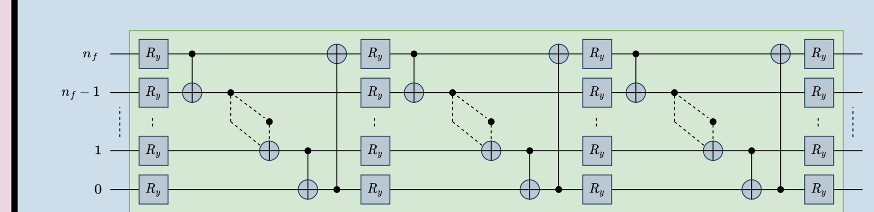
High-level Overview of MQCC Structure

- Like a classical CNN, the multidimensional quantum convolutional classifier (MQCC) has ℓ layers and a fully-connected layer.



Layer Structure

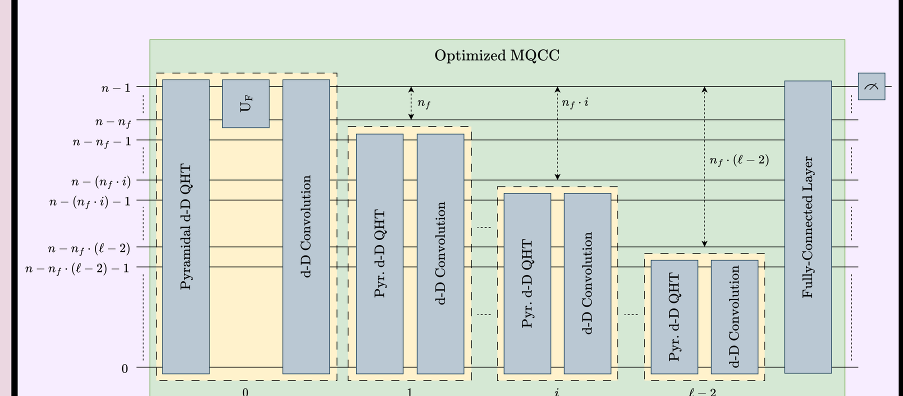
- Each layer made of a trainable quantum convolution operation and a quantum pooling operation.
- Note that RoR permutations are shown in the figure for visual clarity, but do not actually contribute to circuit depth.
- Fully connected layer is a pyramidal cascade of multiplexed rotation (R_y) rotation gates that condense quantum state to a single qubit — like pooling layers in QCNN [1].



Parameterized Filter Operation for Quantum Machine Learning

- To adapt the filter operation U_F for QML, we replaced the inverse arbitrary state synthesis operation with a parameterized ansatz.
- Implementing U_F using inverse arbitrary state synthesis is necessary to implement a pre-defined arbitrary filter, but trainable filters have more flexibility.
- Above parameterized ansatz offers:
  - High parameter density: $4 \cdot n_f$
  - Low circuit depth complexity: $3 \cdot n_f + 4$
  - High qubit entanglement from the ring of CNOT gates

Width-Optimized MQCC



Multidimensional Quantum Convolutional Classifier (Optimized)

- Each layer of quantum convolution introduces n_f additional qubits.
- Over multiple convolution layers, MQCC accumulates a relatively large circuit width of $n + (n_f \cdot \ell)$.
- To conserve circuit width and eliminate the filter qubits, the pooling layers can be placed before their corresponding convolution layers.
- The qubits used to represent high-frequency terms can be used in-place of adding additional qubits.
- Since high-frequency qubits do not start at their ground state, the width optimization introduces some error to the MQCC structure.
- An additional trainable convolution filter is added after the first pooling operation to correct for error.
- Since the filter qubits are eliminated, marginal circuit depth conservation is seen from the fully-connected layer.

Experimental Setup

Techniques:

- Quantum simulation was performed using *Pennylane* [4].
- Classical optimization was performed using *PyTorch* [3].
- Multi-Dimensional Quantum Convolutional Classifiers (MQCCs)
  - Tested with kernel size of 2 and unity stride.
  - Experiments included both MQCC and the width-optimized MQCC.
- Convolutional Neural Network (CNN) [8]
  - Kernel size, stride, circular padding, and feature count from MQCC.
  - Max pooling was used instead of average pooling.
- Quanvolutional Neural Networks [5]
  - Replaced first convolution / pooling layer in CNN with quanvolution layer using stride=2.
- Quantum Convolutional Neural Network (QCNN) [1, 6]
  - Original implementation didn't encode states correctly when dimensions were not powers of 2.
  - Tested with original data encoding and corrected data encoding.

Metrics (average over 10 trials)

- Log Loss
- Accuracy
- Training Time
- Testing Time
- Number of Parameters
- Circuit Depth *(quantum only)*
- Gate Count *(quantum only)*

Datasets

We performed binary classification with the following datasets at their original resolution (see below) and down-sampled to (16×16).

- MNIST [2]: (28×28).
- FashionMNIST [10]: (28×28).
- CIFAR-10 [9]: (32×32×3).

Hardware Specifications:

- Intel Xeon Gold 6342 CPU
  - 48 Cores
  - Base frequency @ 2.8GHz
- 3× NVIDIA A100 80GB GPUs
- 256GB DDR4 RAM @ 3200MHz
- PCIe 4.0 connectivity

Machine Learning Parameters:

- Optimizer: Adam
- Loss Function: Log
- Learning Rate: 0.001
- Training Batch Size: 8
- Testing Batch Size: 1000
- Epoch: 1

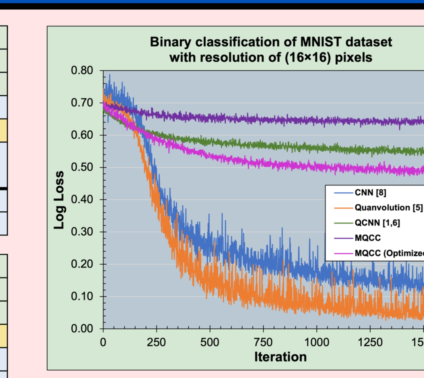
Results and Analysis

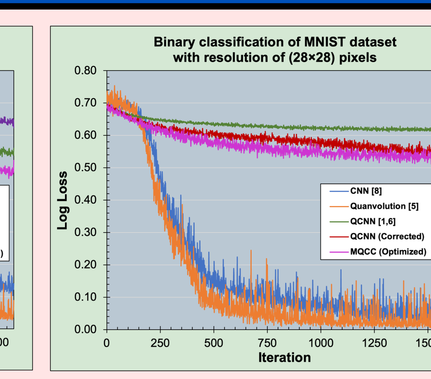
Method	Accuracy					
	MNIST		FashionMNIST		CIFAR-10	
	16×16	28×28	16×16	28×28	16×16×3	32×32×3
CNN [8]	94.62% ± 14.39%	99.42% ± 0.3%	77.16% ± 23.39%	82.11% ± 22.2%	72.69% ± 8.36%	75.02% ± 9.08%
Quanvolution [5]	99.47% ± 0.13%	99.67% ± 0.12%	91.26% ± 14.53%	91.51% ± 14.63%	78.33% ± 1.96%	77.14% ± 9.79%
QCNN [1,6]	99.16% ± 0.87%	99.22% ± 0.45%	93.16% ± 0.77%	84.33% ± 10.29%	51.96% ± 3.25%	58.91% ± 7.63%
QCNN (Corrected)		97.2% ± 1.91%		92.53% ± 1.55%		
MQCC	89.61% ± 0.63%	Out of VRAM	86.06% ± 0.98%	Out of VRAM	55.13% ± 3.77%	Out of VRAM
MQCC (Optimized)	99.18% ± 0.63%	97% ± 2.02%	93.21% ± 0.98%	94.15% ± 1.21%	58.49% ± 3.77%	58.73% ± 4.7%

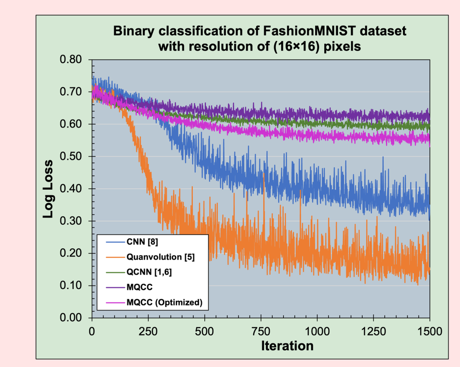
Method	Training Time (seconds)					
	MNIST		FashionMNIST		CIFAR-10	
	16×16	28×28	16×16	28×28	16×16×3	32×32×3
CNN† [8]	5.417 ± 0.854	6.274 ± 0.806	5.859 ± 2.85	5.924 ± 0.963	4.505 ± 0.69	5.055 ± 0.692
Quanvolution†‡ [5]	33.817 ± 0.659	34.858 ± 1.131	30.77 ± 2.722	33.783 ± 2.598	30.644 ± 1.067	64.322 ± 9.82
QCNN‡ [1,6]	191.358 ± 8.808	285.516 ± 3.371	179.82 ± 3.901	270.591 ± 6.715	219.463 ± 1.54	314.658 ± 3.077
QCNN‡ (Corrected)		290.376 ± 7.811		280.049 ± 15.621		
MQCC‡	743 ± 1.712	Out of VRAM	704.671 ± 2.338	Out of VRAM	229.737 ± 1.193	Out of VRAM
MQCC‡ (Optimized)	103.799 ± 1.712	150.577 ± 13.391	105.35 ± 2.338	140.052 ± 0.626	95.567 ± 1.193	139.237 ± 4.279

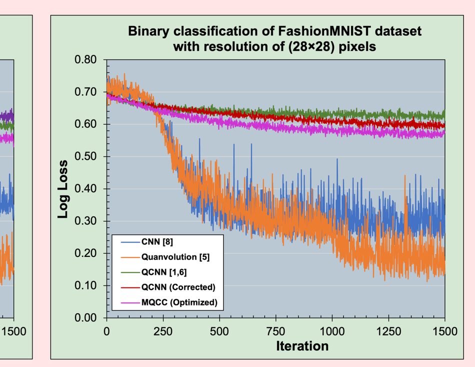
Method	Testing Time (seconds)					
	MNIST		FashionMNIST		CIFAR-10	
	16×16	28×28	16×16	28×28	16×16×3	32×32×3
CNN† [8]	0.169 ± 0.002	0.166 ± 0.003	0.16 ± 0.003	0.155 ± 0.004	0.2 ± 0.001	0.196 ± 0.004
Quanvolution†‡ [5]	0.212 ± 0.002	0.217 ± 0.004	0.193 ± 0.006	0.199 ± 0.008	0.251 ± 0.003	0.322 ± 0.01
QCNN‡ [1,6]	0.703 ± 0.002	0.941 ± 0.007	0.625 ± 0.002	0.826 ± 0.015	0.844 ± 0.032	1.019 ± 0.005
QCNN‡ (Corrected)		0.873 ± 0.014		1.885 ± 3.454		
MQCC‡	1.829 ± 0.004	Out of VRAM	1.736 ± 0.008	Out of VRAM	5.662 ± 0.004	Out of VRAM
MQCC‡ (Optimized)	0.646 ± 0.004	0.801 ± 0.02	0.593 ± 0.008	0.715 ± 0.006	0.758 ± 0.004	0.856 ± 0.006

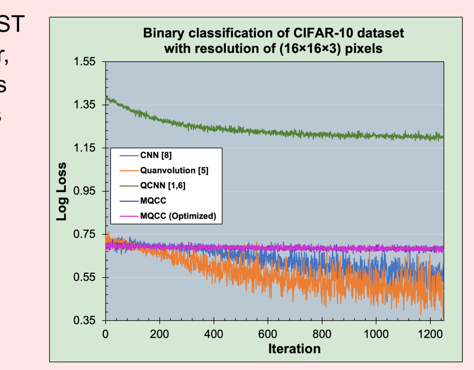
†Includes actual execution time of neural network layers. ‡Includes execution time of *simulated* quantum circuits

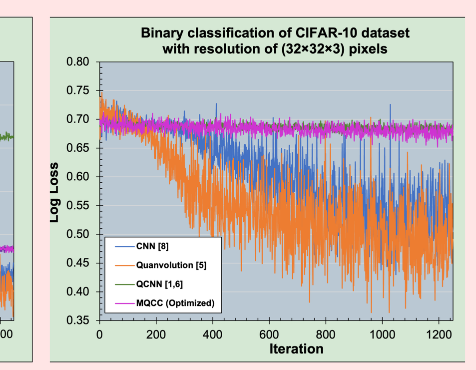












- **Accuracy:** MQCC (Optimized) shows the highest accuracy for the FashionMNIST dataset, while CNN outperforms all other models for CIFAR10 dataset. However, CNNs exhibit high accuracy variance (high maximum, low minimum) owing to its model complexity (number of parameters). Furthermore, Quanvolution performs relatively better than other models for MNIST dataset.
- **Training Time and Testing Time:** The classical and the hybrid model require lower time for training and testing compared to the quantum models. The implementation of MQCC is challenging for larger images and requires comparatively more time. However, MQCC (Optimized) requires less time compared to QCNN (Corrected).
- **Gate Count:** From our basic gate decomposition analysis, both MQCC models have lower gate counts regardless of the data size.
- **Circuit Depth:** Both MQCC models offer lower circuit depth for smaller datasets compared to QCNN.
- **Number of Parameters:** MQCC (Optimized) requires the lowest number of training parameters. Overall, the quantum convolution models require fewer training parameters compared to the classical and hybrid model.
- **Training Log-Loss:**
  - Both MQCC models show consistent behavior towards the log-loss measure as the number of iterations increases.
  - Classical and Quanvolution require a relatively higher number of training iterations for reaching a "saturation" point.
  - Classical and Quanvolution exhibit minimal loss for simple datasets, i.e., MNIST, which does not necessarily result in achieving higher accuracy. Such behavior suggests the model is overfitting to training data.

Method	Number of Parameters			
	16×16	16×16×3	28×28	32×32×3
CNN [8]	234	266	302	334
Quanvolution [5]	230	358	298	426
QCNN [1,6]	51	51	68	68
MQCC	62	78	66	82
MQCC (Optimized)	46	50	58	62

Method	Circuit Depth			
	16×16	16×16×3	28×28	32×32×3
QCNN [1,6]	135	188	188	189
MQCC	111	115	242	242
MQCC (Optimized)	107	111	238	238

Method	Gate Count			
	16×16	16×16×3	28×28	32×32×3
QCNN [1,6]	415	593	593	693
MQCC	183	191	352	360
MQCC (Optimized)	151	159	312	320

Conclusion and Future Work

We proposed a variational quantum classification technique that preserves locality in multidimensional data using quantum convolution and QHT-based pooling.

- Compared to QCNNs, our optimized MQCC achieved improvements in log loss, accuracy, training/testing time, and gate count due to the preservation of data-locality.
- Compared to CNNs and quanvolutional neural networks, MQCC demonstrated faster convergence, higher average accuracy in the MNIST and FashionMNIST datasets while substantially reducing the number of training parameters.
- Future Work:
  - Optimizations and extensions to multiclass classification
  - Investigate scalability using real-world datasets
  - Implementation on physical quantum hardware

Acknowledgements: This research used resources of the *Oak Ridge Leadership Computing Facility*, which is a DOE Office of Science User Facility supported under Contract DE-AC05-00OR22725.

References

[1] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. 2019. Quantum convolutional neural networks. Nature Physics 15, 12 (01 Dec 2019), 1273–1278. https://doi.org/10.1038/s41567-019-0648-8

[2] Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine 29, 6 (2012), 141–142.

[3] Adam Paszke et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[4] Ville Bergholm et al. 2022. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968 [quant-ph] https://pennylane.ai/

[5] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook.2020. Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence 2, 1 (2020), 2.

[6] Tak Hur, Leeseok Kim, and Daniel K. Park. 2022. Quantum convolutional neural network for classical data classification. Quantum Machine Intelligence 4, 1 (10 Feb 2022), 3. https://doi.org/10.1007/s42484-021-00061-x

[7] Mingyoung Jeng et al. 2023. Improving quantum-to-classical data decoding using optimized quantum wavelet transform. The Journal of Supercomputing (16 Jun 2023). https://doi.org/10.1007/s11227-023-05433-7

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. Nature 521, 7553 (2015), 436–444.

[9] Learning Multiple Layers of Features from Tiny Images. 2009. Alex Krizhevsky. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[10] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv:1708.07747 [cs.LG]