Implementing a Quantum Image Processing Technique with Classical Neural Additive Models on MNIST

Negar Dokhtmirzahasanvahid
Advanced Topics in Image Processing

Outline

Literature Review

- Survey the most influential papers in Quantum Image Processing.

Interpretable AI Foundations

- Introduce Neural Additive Models (NAMs) and other interpretable ML techniques.

QHED Pipeline

- Train a NAM on MNIST using angle-encoded QHED features.

- Compare performance to find the best threshold

FRQI + QHED Pipeline

- Train a NAM on MNIST using FRQI-encoded QHED features.
- Compare performance with a classical baseline.
- Perform ANOVA testing to evaluate statistical significance.

Motivation

Can Quantum Encodings of Pictures be Used in Interpretable Classical Models?

Quantum Computers

- Qubits are the basic unit of quantum information.
- Quantum bits can be 0, 1, or both at once.
- Quantum computers operate using these quantum bits or qubits.

Quantum Speed-up

- Shor’s Factoring Algorithm: Finds the prime factors of a big number.
- Grover’s Search Algorithm: Finds the “right” item in a list faster than classical search.
- HHL: Solves systems of linear equations using quantum circuits.

Quantum Speed-up

Algorithm	Problem	Classical Complexity	Quantum Complexity	Speedup Type
Shor's Algorithm	Integer factoring	$O(e^{(\log N)^{1/3}(\log\log N)^{2/3}})$ (sub-exponential)	$O((\log N)^3)$	**Exponential**
Grover's Algorithm	Unstructured search (size N)	$O(N)$	$O(\sqrt{N})$	**Quadratic**
HHL Algorithm	Solve $A\vec{x} = \vec{b}$	$O(N\log N)$ to $O(N^3)$	$O(\log N)$ *	**Exponential***

Biggest Challenge: NOISE

Why Do Qubit Operations Go Wrong?

- Qubits are extremely sensitive to their surroundings.
- Even tiny interactions with the environment, like **thermal noise or stray electromagnetic fields**, disturb their state.
- Over time, this leads to loss of coherence or unwanted entanglement with external systems.
- Two-qubit gates are more vulnerable:

  Any mismatch in frequency, timing, or control pulses introduces additional noise.

As a result, quantum gates often drift from their ideal behavior, causing computational errors.

Literature Review

Why Quantum Data?

Introducing the Most Successful Quantum Image Encoding Algorithms

Novel Enhanced Quantum Representation (NEQR)

NEQR stores each pixel value directly in a block of qubits, bit-by-bit, alongside the qubits that mark its row and column. Instead of encoding brightness as a rotation, you simply write the binary digits of the intensity into the quantum register.

Deterministic readout: You can recover each pixel exactly in one go, without repeated sampling.
Higher qubit cost: Because you allocate one qubit per bit of intensity, large bit-depth images demand many qubits.**Circuit depth**: Writing all those bits can require deeper circuits, which may amplify gate errors on noisy devices.

Best for

Applications where exact pixel values matter (e.g., lossless pattern matching, quantum templates).

Why Quantum Encoding Data for Classical Models?

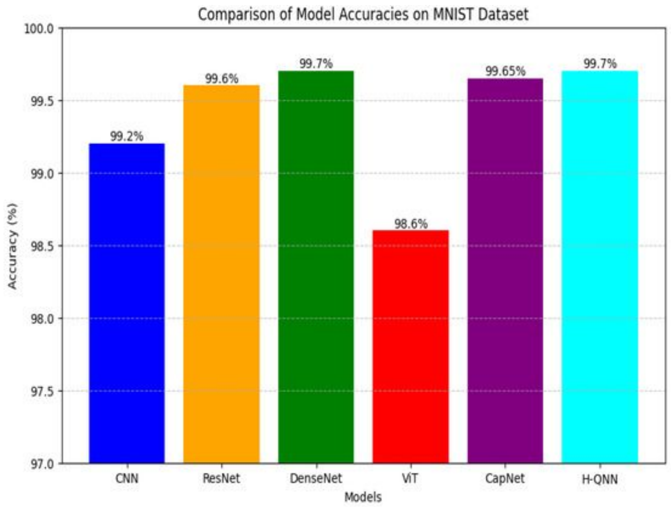
Table 1 *(Continued)*

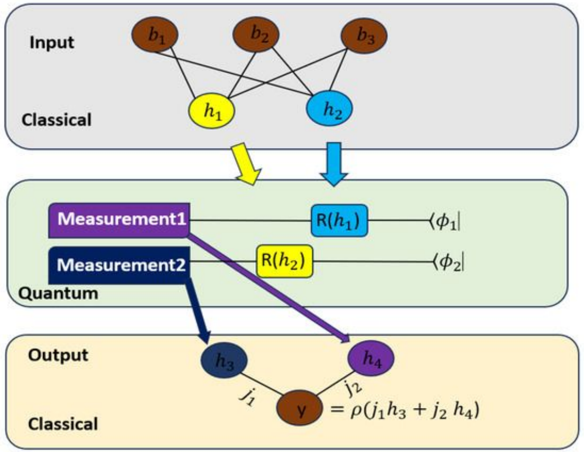
Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
Gradient Boosting	2	Classical	74.8663	0.7370	0.7836	0.7836	0.7596	0.7482	0.4968	0.6142
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1080
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1088
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1088
	15	Classical	74.3316	0.7221	0.8021	0.8021	0.7600	0.7425	0.4858	1.7732
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1905
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0955
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0955
	23	Classical	78.4759	0.7884	0.7863	0.7863	0.7873	0.7847	0.5695	2.6064
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1531
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1027
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1027
XGBoost	2	Classical	74.3316	0.7379	0.7652	0.7652	0.7513	0.7430	0.4863	0.6346
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0298
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0290
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0290
	15	Classical	73.3957	0.7239	0.7678	0.7678	0.7452	0.7335	0.4674	8.6842
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0433
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0255
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0255
	23	Classical	74.0642	0.7493	0.7335	0.7335	0.7413	0.7407	0.4813	5.7666
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1155
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0253
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0253

Screenshot

Rath, M., & Date, H. (2024). Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology, 11*(72). https://doi.org/10.1140/epjqt/s40507-024-00285-3

Why Quantum Encoding Data for Classical Models?





The H-QNN achieved a classification accuracy of 99.7% on the binary MNIST task, demonstrating superior performance compared to traditional CNNs and QCNNs.
Hybrid Quantum–Classical Neural Networks for Efficient MNIST Binary Image Classification

Ranga, D., Prajapat, S., Akhtar, Z., Kumar, P., & Vasilakos, A. V. (2024). Hybrid quantum–classical neural networks for efficient MNIST binary image classification. Mathematics, 12(23), 3684. https://doi.org/10.3390/math12233684

Quantum Hadamard Edge Detection (QHED)

QHED finds edges by using the Hadamard transform to compare neighboring pixel values in superposition. You prepare a quantum image, apply a layer of Hadamard gates that effectively compute local differences, and then measure to highlight where intensity changes sharply.

Linear circuit depth: Typically one layer of Hadamards plus the image-preparation routine, so it's shallow enough for current devices.

Quantum advantage potential: For very large images, the superposition lets you probe all pixel transitions in one shot, in principle faster than classical loop, even though noise can erode that gain.

Sensitivity to noise: Since edge detection amplifies differences, even small gate errors can introduce spurious "edges."

Best for

- Binary or grayscale images where detecting outlines is the goal (e.g., feature extraction in medical scans).

Angle Encoding

Angle Encoding encodes classical data into the phase of quantum states by rotating qubits around a chosen axis on the Bloch sphere. Rather than storing a binary string directly, it embeds information into how much each qubit is rotated.

Performance:

- **Compact and expressive**: You can embed continuous values using just one qubit per feature, making it highly efficient for datasets with many features.
- **Noise resilience**: Shallow circuits with fewer gates can lead to reduced decoherence, which makes this method appealing for NISQ-era devices.
- **No exact reversibility**: Unlike NEQR, you don't get exact pixel values or bit patterns back, only statistical expectations from measurement outcomes.

Flexible Representation of Quantum Images (FRQI)

FRQI encodes a grayscale picture by separating “where” each pixel lives from “how bright” it is.

You use one group of qubits to label positions (the rows and columns) and a single extra qubit whose orientation represents intensity.

To read any pixel, you look at that position label and then gently measure the intensity qubit.

Compactness: Only one qubit per pixel plus a small overhead, so it scales well when you just need a rough sketch of intensities.

Probabilistic retrieval: Measuring to get a pixel’s value is inherently noisy, so you often repeat the process many times to estimate intensities accurately.

Noise sensitivity: If your hardware has significant decoherence, the rotations on that one intensity qubit can be blurred, making subtle shades harder to recover

Best for: Grayscale images where you value a low qubit count over perfect precision (e.g., simple patterns, silhouettes).

Summary

QHED (Quantum Hadamard Edge Detection)

1. RY(2 * arcsin(sqrt(p))) → encodes pixel intensity
2. H → highlights contrast (edges)
3. Measure → reads edge info

Summary

Angle Coding

1. RX(x_i) or RY(x_i) → encodes each feature as a rotation
2. Measure → gets feature amplitudes

Summary

FRQI (Flexible Representation of Quantum Images)

1. H on index qubits → superposition over pixel positions
2. Multi-controlled RY($2\theta_i$) → encodes pixel brightness per position
3. Measure → reads position–intensity relation

Why Quantum Encoding Data for Classical Models?

Table 1 Classical data and quantum data embedding performance comparison

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
Logistic Regression	2	Classical	72.9947	0.7241	0.7546	0.7546	0.7390	0.7296	0.4595	0.0052
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0059
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0052
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0052
	15	Classical	75.9358	0.7519	0.7836	0.7836	0.7674	0.7590	0.5184	0.0130
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0084
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0051
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0051
	23	Classical	79.0106	0.7921	0.7941	0.7941	0.793	0.7900	0.5801	0.0095
		Quantum Basis Encoding	66.8449	0.6512	0.7440	0.7440	0.6945	0.6674	0.3354	0.0061
		Quantum Angle Encoding	66.8449	0.6512	0.7440	0.7440	0.6945	0.6674	0.3354	0.0029
		Quantum Amplitude Encoding	66.8449	0.6512	0.7440	0.74406	0.6945	0.6674	0.3354	0.0029
KNN	2	Classical	72.3262	0.7172	0.7493	0.7493	0.7329	0.7229	0.4461	0.0437
		Quantum Basis Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0520
		Quantum Angle Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0859
		Quantum Amplitude Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0859
	15	Classical	69.9198	0.6766	0.7784	0.7784	0.7239	0.6981	0.3971	0.0995
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1338
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0493
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0493
	23	Classical	73.7968	0.7194	0.7916	0.7916	0.7538	0.7372	0.4751	0.0848
		Quantum Basis Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0884
		Quantum Angle Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448
		Quantum Amplitude Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448

Rath, M., & Date, H. (2024). Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology, 11*(72). https://doi.org/10.1140/epjqt/s40507-024-00285-3

Why Quantum Encoding Data for Classical Models?

Table 1 (*Continued*)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
SVM Linear	2	Classical	73.1283	0.7247	0.7573	0.7573	0.7406	0.7309	0.4621	0.2453
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3365
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3376
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3376
	15	Classical	75.0000	0.7388	0.7836	0.7836	0.7606	0.7495	0.4995	0.5831
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5701
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2929
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2929
	23	Classical	77.5401	0.7482	0.8391	0.8391	0.7910	0.7745	0.5500	0.6471
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5428
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1568
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1568
SVM Poly	2	Classical	71.7914	0.6714	0.8681	0.8681	0.7572	0.7159	0.4335	0.3055
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.4763
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3603
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3603
	15	Classical	72.1925	0.7031	0.7810	0.7810	0.7400	0.7211	0.4429	0.3926
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4803
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2430
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2430
	23	Classical	74.0642	0.7273	0.7810	0.7810	0.7532	0.7401	0.4807	0.4581
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.7717
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1801
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1801

Rath, M., & Date, H. (2024). Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy. *EPJ Quantum Technology, 11*(72). https://doi.org/10.1140/epjqt/s40507-024-00285-3

Quantum Data in Interpretable ML Models

Main Question: Can we store data using quantum encodings and get an improvement using Interpretable ML Models? What is the baseline before error mitigation?

Interpretable Machine Learning Models

- Introducing the XAI Field
- Exploring NAMs

Interpretable and Explainable AI

Interpretable and Explainable AI (often called XAI) is a field focused on making machine learning models more understandable to humans.

- **Interpretability** = *Built-in clarity*
  The model is understandable by design (e.g. linear models, decision trees, NAMs).

- **Explainability** = *Post-hoc insight*
  External tools explain how a complex model made a decision (e.g. SHAP, LIME).

Why Interpretable & Explainable AI Matters

You need not just accuracy, you need trust.

In medicine and drug design, clinicians and researchers must understand model decisions before acting on them.

Life-critical decisions need transparency.

Models help with diagnosis, treatment, and molecule selection — but black boxes risk undetected errors.

Neural Additive Models (NAMs)

NAMs blend classical interpretability with neural power.

They build on the idea of Generalized Additive Models (GAMs), where each feature contributes independently and additively to the prediction. But now, each contribution is learned by a neural network.

Features:

- **Interpretability you can actually plot.**
  NAMs learn a separate function for each feature, which means you can visualize the shape of how each input affects the model's output. No need for post-hoc explanations.
- **Nonlinear patterns, still easy to explain.**
  Unlike linear models, NAMs can model complex curves and thresholds. While still keeping the structure simple enough for human inspection.
- **You know what the model is focusing on.**
  Each feature has its own dedicated subnetwork. You can inspect the function it learned, and even turn it off to see how prediction changes.

Neural Additive Models (NAMs)

1. Feature Grouping

Instead of feeding all features to one big network, you:

- Divide the input features into GGG groups (e.g., 676 pixels → 26 groups of 26 pixels)
- Each group is processed by its own MLP

2. Attention

To control how much to "trust" each subnetwork's output, NAMs can include a **feature attention mechanism**

3. Output Layer

The outputs of all subnetworks (optionally weighted) are summed and passed through a residual head or classification layer. This often includes:

- **Normalization (e.g., LayerNorm)**
- **ReLU and dropout**
- **Final linear layers to produce logits**

Idea:

Implementing FRQI with QHED and feeding ot to a classical NAM Model

Set Up: MNIST Dataset

Set Up: Software

- Trained and handled dataset with PyTorch
- Preprocessed the data classically using TorchVision
- Performed ANOVA analysis using SciPy
- Generated all the plots by Matplotlib
- Preprocessed the quantum data using Qiskit’s QASM Simulator

Set Up

For training the models, number of features: 10000 (20/20/60) split

For finding the best threshold with Angle Encoding: 2000 features
2000 (20/20/60) split

Set Up: NAM

Feature networks
A list of small neural nets (SpatialFeatureNet), each assigned to a different group of input features. This setup processes each part of the input separately, maintaining interpretability.

Attention mechanism
A neural layer that learns to assign weights to each feature group. This tells the model how much to trust each group when making a prediction, which gives built-in feature importance.

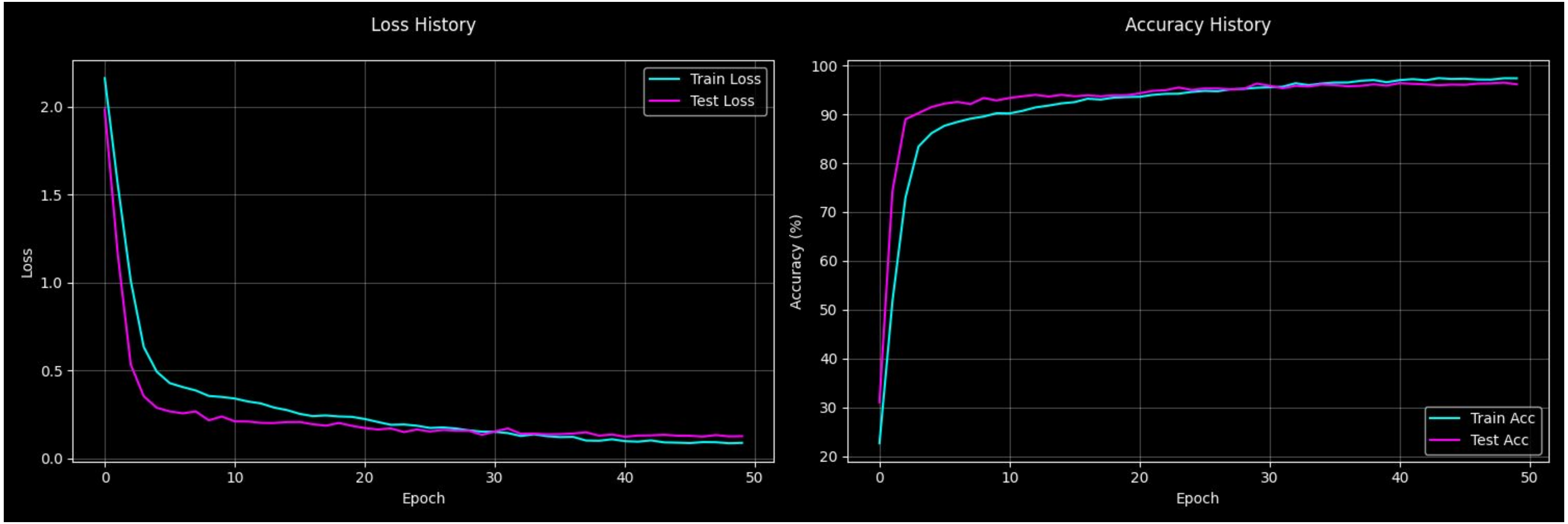
Output layer
Takes the attention-weighted features and maps them to the final class scores. It includes a residual-style layout to help stability and generalization. Outputs 10 logits for classification.

```
class InterpretableNAM(nn.Module):
  ...
        # Create feature networks with spatial awareness
        self.feature_nets = nn.ModuleList([
            SpatialFeatureNet(self.features_per_group, hidden_dims, dropout_rate)
            for _ in range(feature_groups)
        ])

        # Attention mechanism for feature importance
        self.feature_attention = nn.Sequential(
            nn.Linear(feature_groups, feature_groups),
            nn.LayerNorm(feature_groups),
            nn.ReLU(),
            nn.Dropout(dropout_rate),
            nn.Linear(feature_groups, feature_groups),
            nn.Softmax(dim=1)
        )

        # Output layer with residual connection
        self.output_layer = nn.Sequential(
            nn.Linear(feature_groups, 32),
            nn.LayerNorm(32),
            nn.ReLU(),
            nn.Dropout(dropout_rate),
            nn.Linear(32, 10)
        )

    []
...
```

Set Up: Baseline NAM

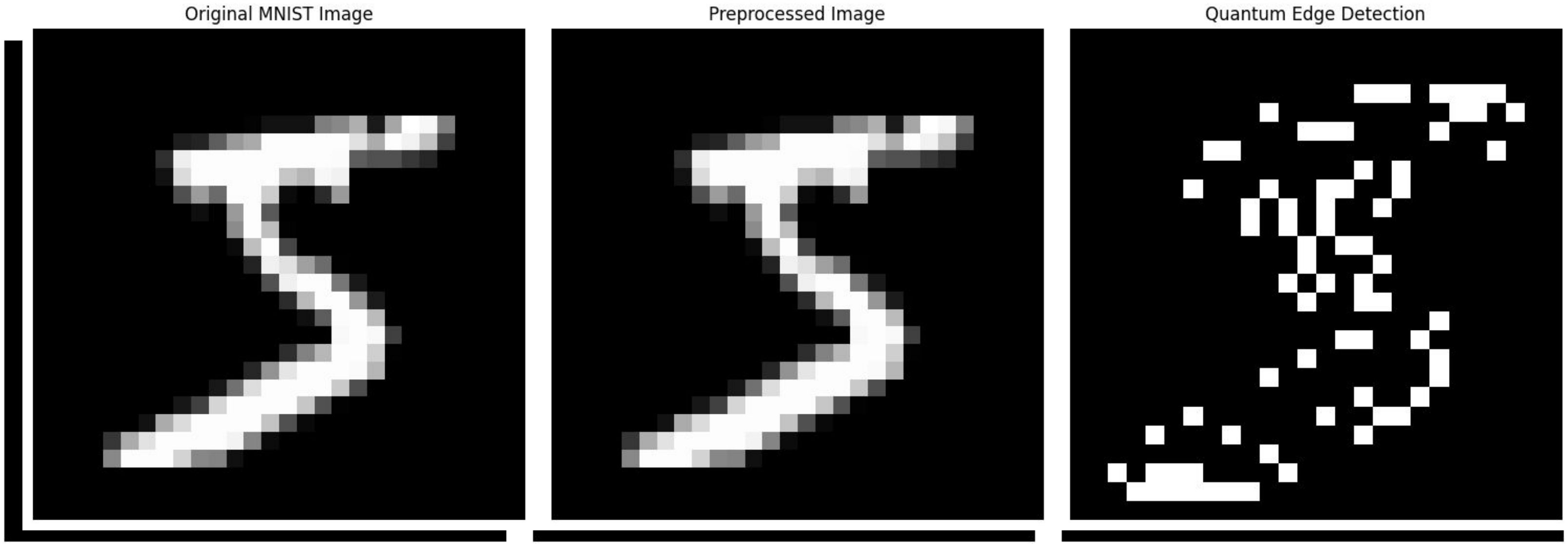
ID	G	Hidden Dims	Dropout	#Params	Val Acc (%)
A	**32**	[128,256,128,64]	**0.30**	430 k	**96.1**
B	16	[128,256,128,64]	0.30	215 k	87.8
C	32	[64,128,64,32]	0.30	215 k	78.5
D	32	[128,256,128,64]	0.50	430 k	90.7
E	64	[128,256,128,64]	0.30	860 k	89.9

Baseline NAM

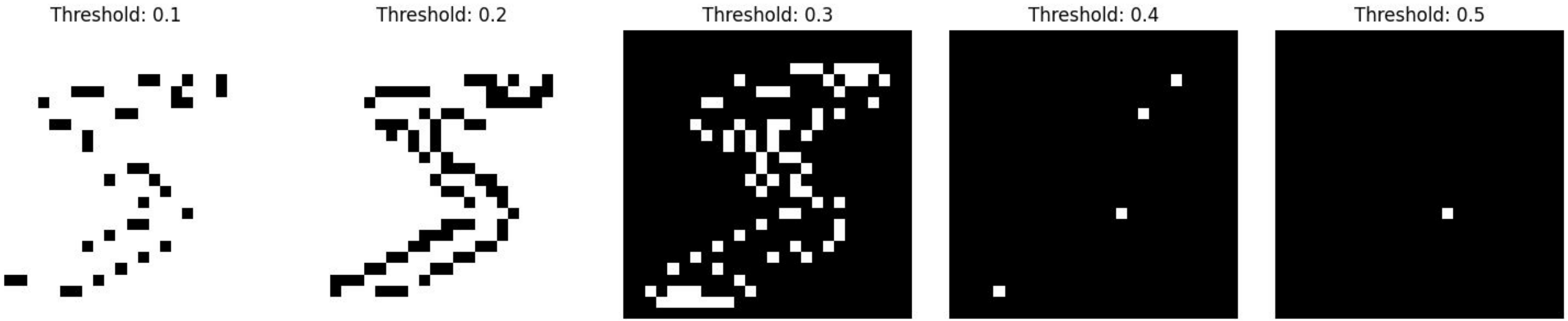


QHED with Angle Encoding

QHED with Angle Encoding: Demo



QHED with Angle Encoding: Different QHED Thresholds



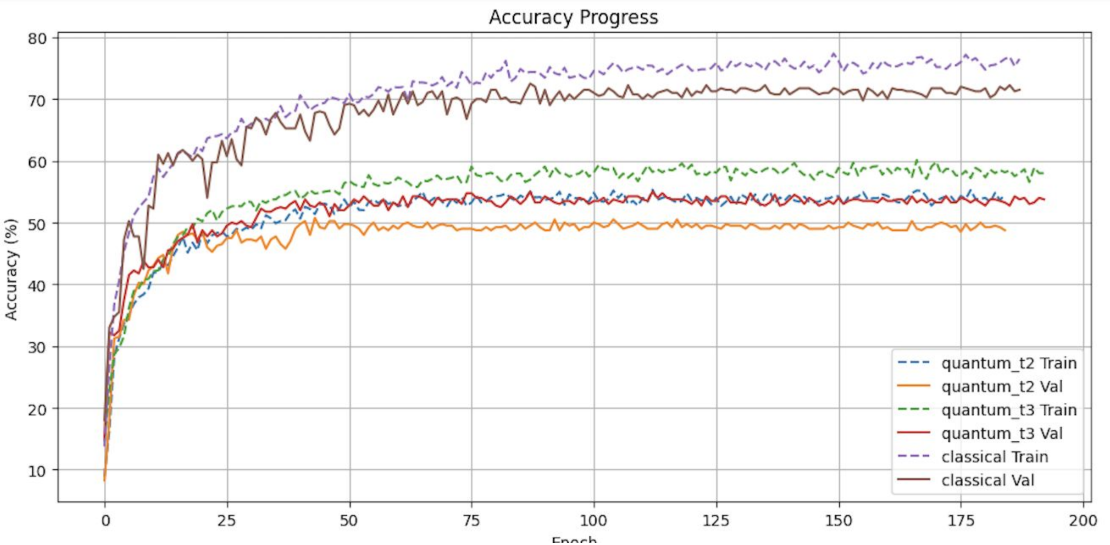
- QHED applies edge detection using quantum interference patterns.
- Thresholds control how strong an edge must be to appear in the output.
- Lower thresholds (0.1, 0.2) capture more detail but introduce noise.
- Higher thresholds (0.4, 0.5) filter weak edges, keeping only the most prominent ones.

QHED with Angle Encoding

Classical model reaches the highest accuracy, both on training and validation sets.

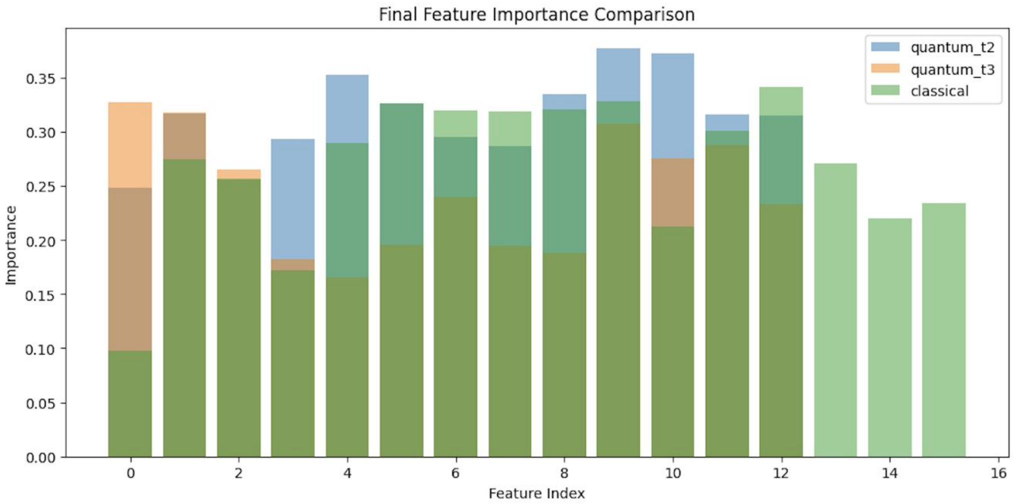
Quantum models improve over time but plateau earlier and lower.

Quantum_t3 slightly outperforms quantum_t2, but both lag behind classical.

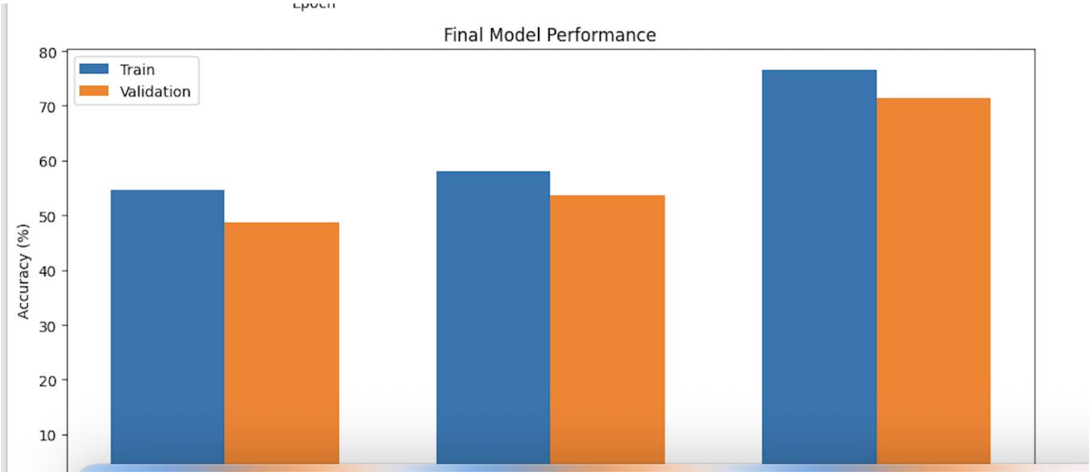


QHED with Angle Encoding

- The classical model spreads attention pretty evenly across features. It's not just relying on a few; it pulls signal from a wide range of the input.
- Quantum_t2 is much more selective. It spikes on a few features like index 4 and 10, and mostly ignores the rest.
- Quantum_t3 sits somewhere in between. It's more balanced than t2 but still doesn't match the classical model's distribution.
- So even though all models saw the same MNIST data, they end up focusing on different parts. That's not just a performance gap. It's a difference in how they represent the data.



QHED with Angle Encoding



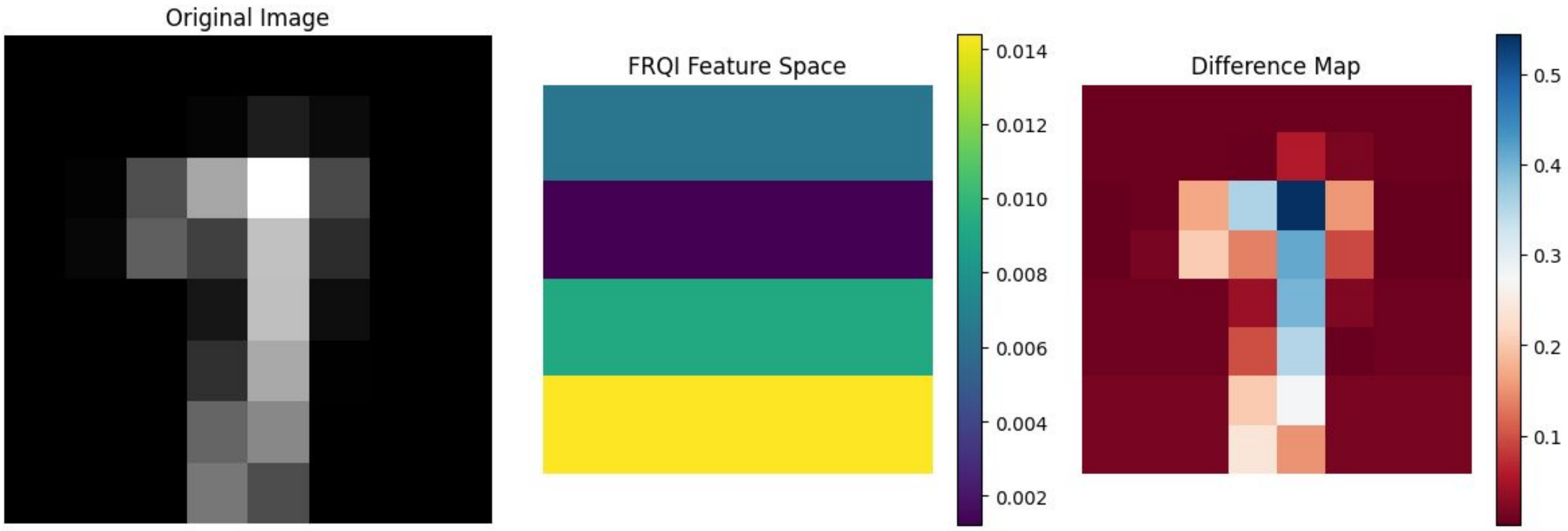
Classical model ends up with the highest accuracy overall, both on training and validation.

Quantum_t3 performs better than quantum_t2.

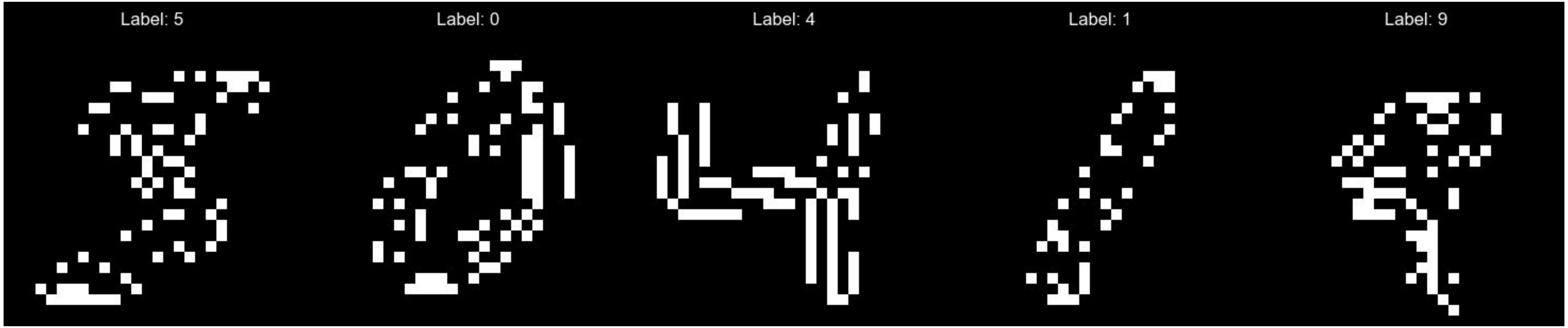
The gap between training and validation is small for all models — so no clear overfitting.

QHED with FRQI

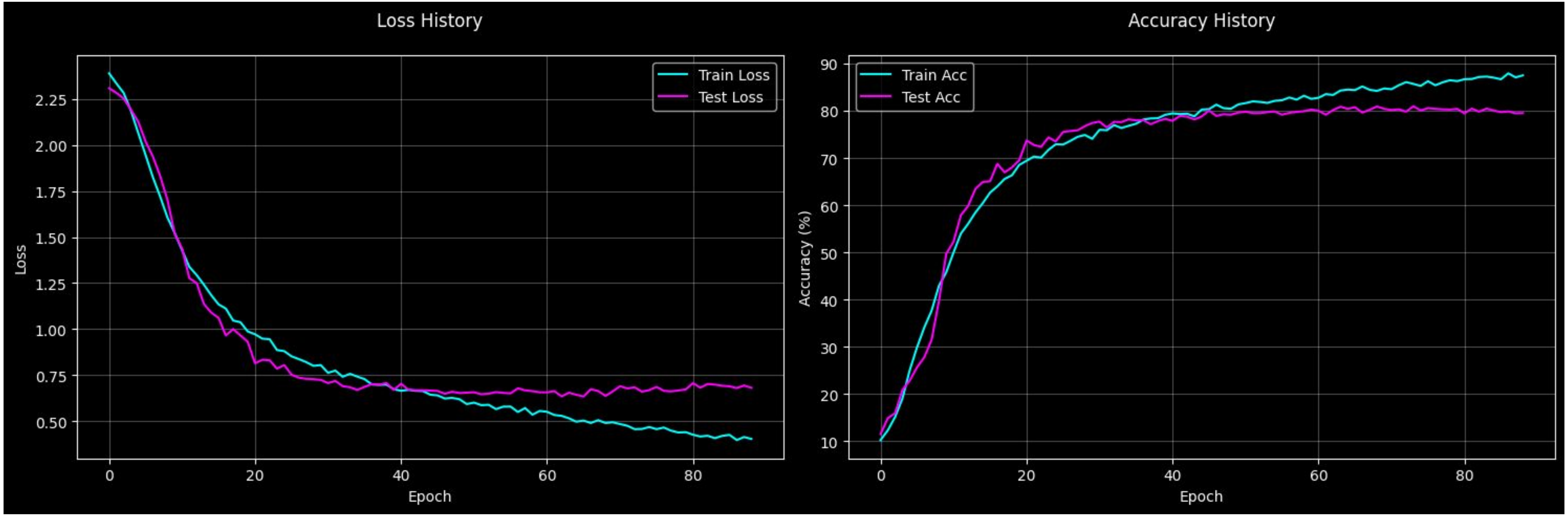
QHED with FRQI: Demo



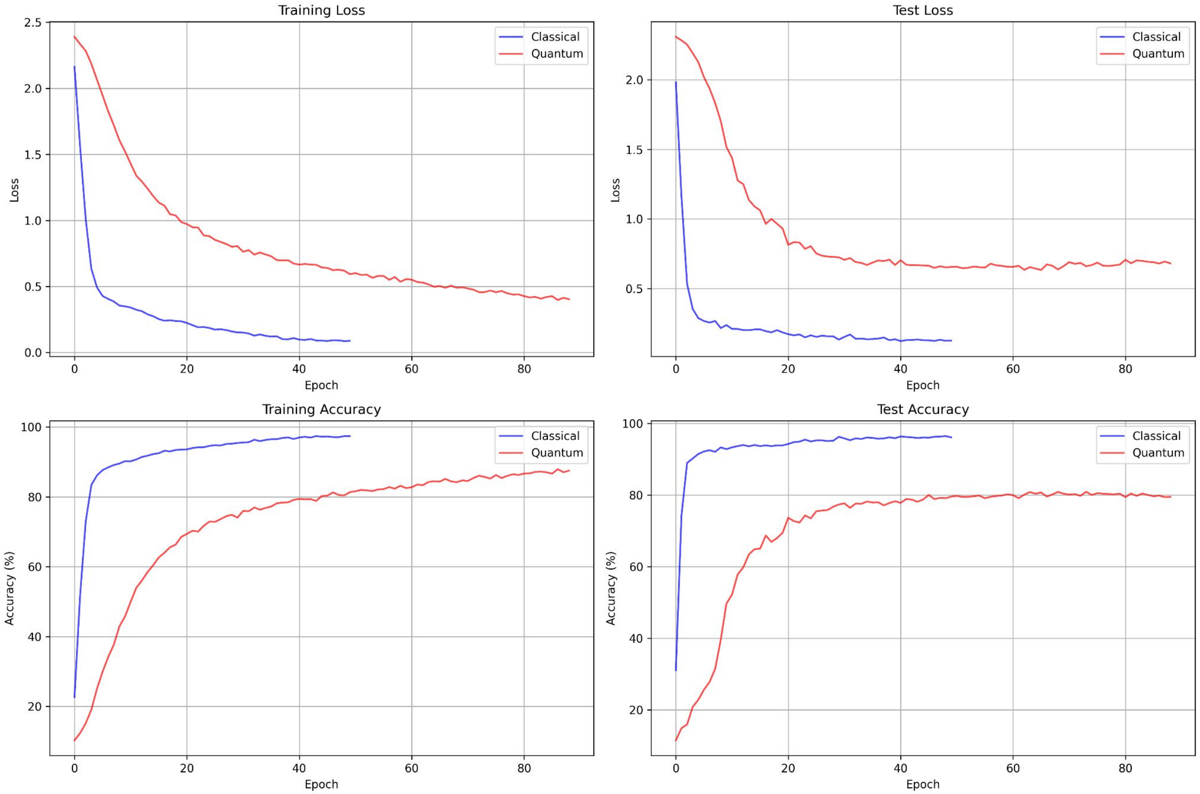
QHED with FRQI: Demo



QHED with FRQI

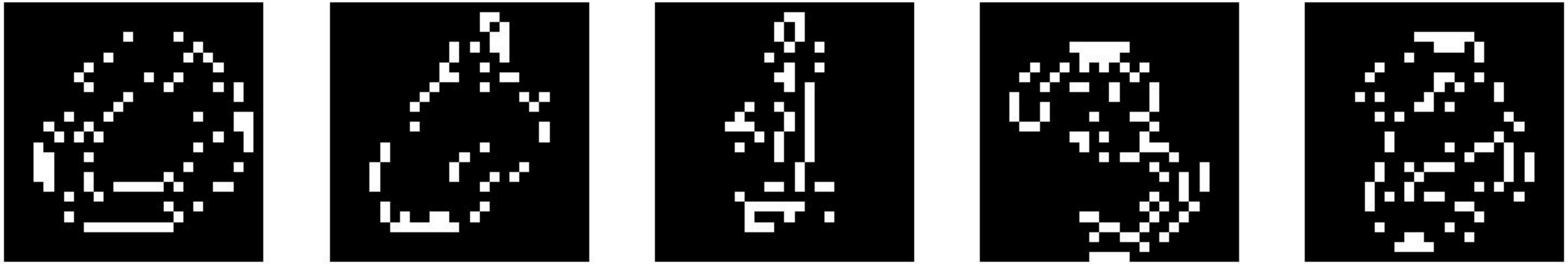


Comparing The Models

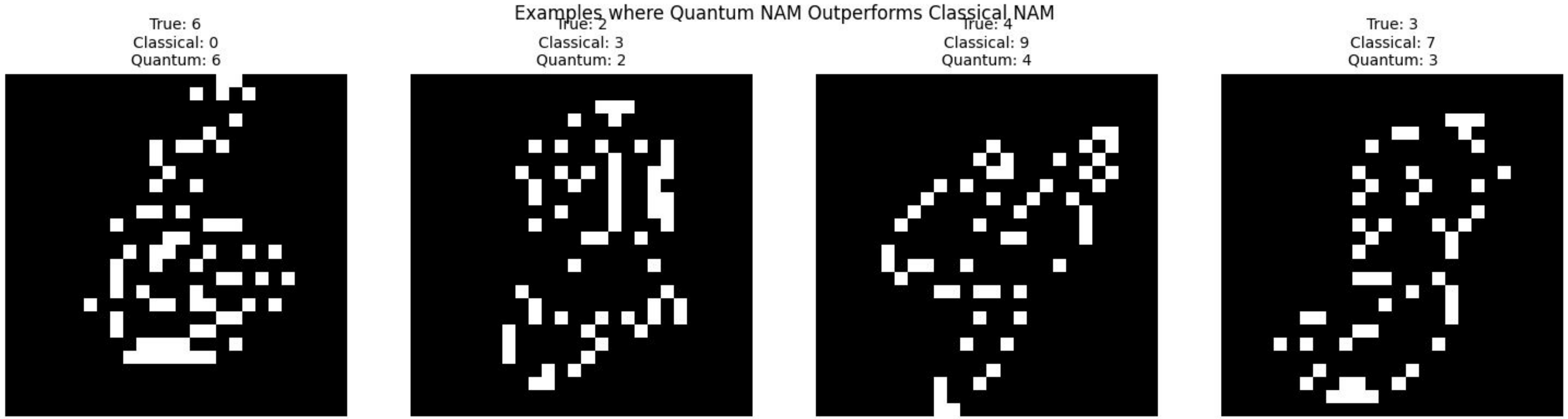


Quantum Advantage?

Examples where Models Disagree



Quantum Advantage?



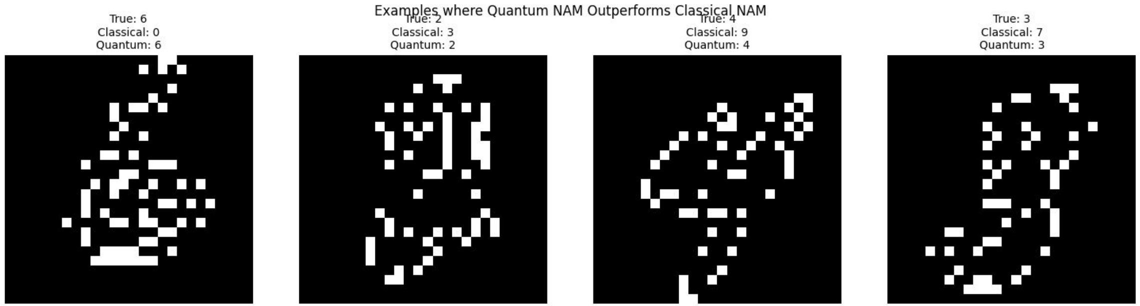
Quantum Advantage?

The quantum model shows particularly strong improvements over the classical model for certain digits:

Digit 1: 79.19% improvement

Digit 0: 61.26% improvement

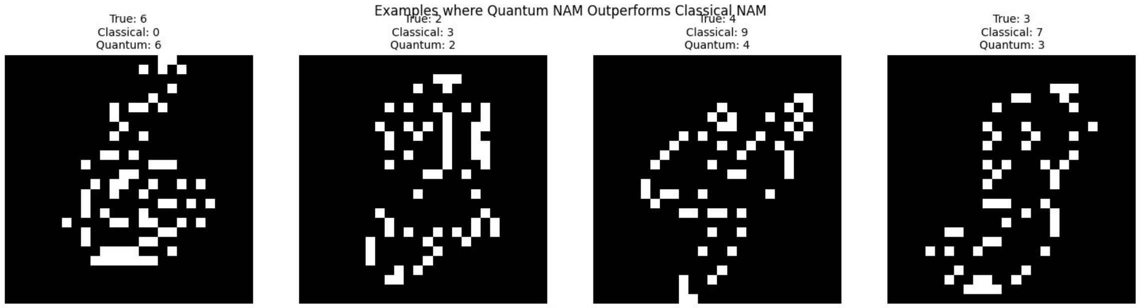
Digit 6: 58.99% improvement



Quantum Advantage?

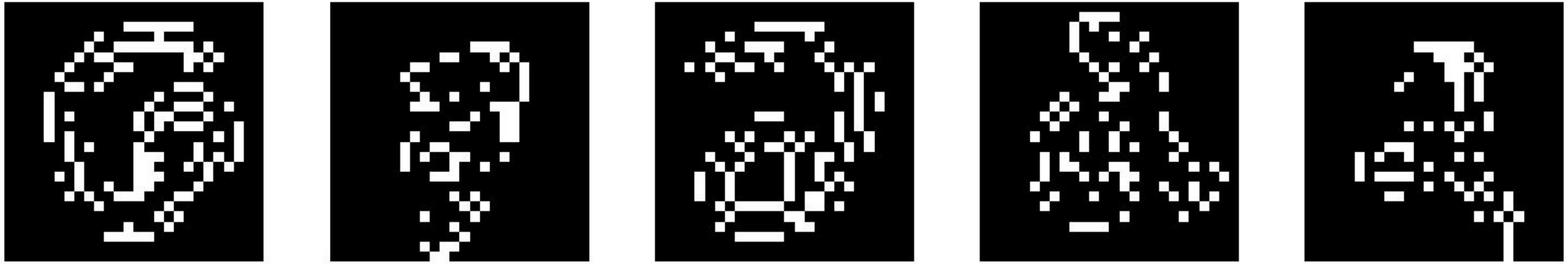
Digit 7 shows the smallest quantum advantage and the highest classical advantage

The quantum model shows significant advantages in recognizing certain digits, particularly digit 1 where it outperforms the classical model in 78.17% of cases

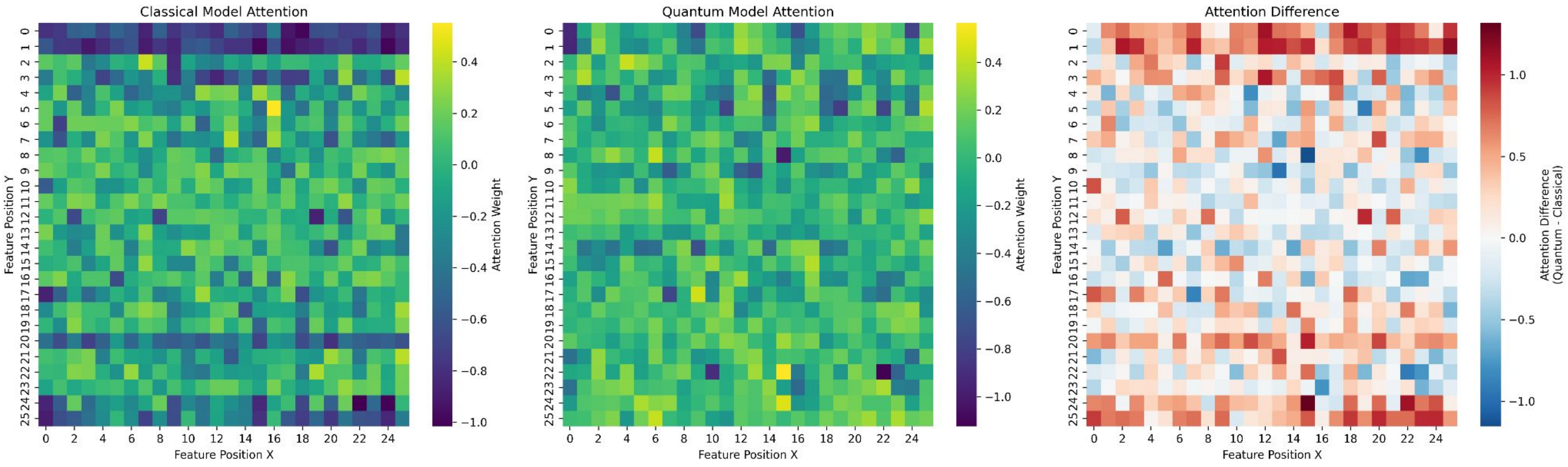


Quantum Advantage?

Examples where Classical NAM is Correct but Quantum NAM is Wrong



Comparing The Models



Comparing The Models

- Classical Model: More concentrated attention (-0.1396 ± 0.3219)
- Quantum Model: More distributed attention (-0.0262 ± 0.2402)
- Both models have similar ranges but different distribution.
- Classical Model: Focuses on central regions (positions like [7,13], [21,25])
- Quantum Model: More spread out attention (positions like [25,6], [2,4])
- The models show distinct attention patterns with minimal overlap

QHED with FRQI vs Classical Baseline

The significant F-statistic and p-value indicate a statistically significant difference in training loss between classical and quantum models. The classical model exhibits a lower mean training loss, suggesting better performance during training.

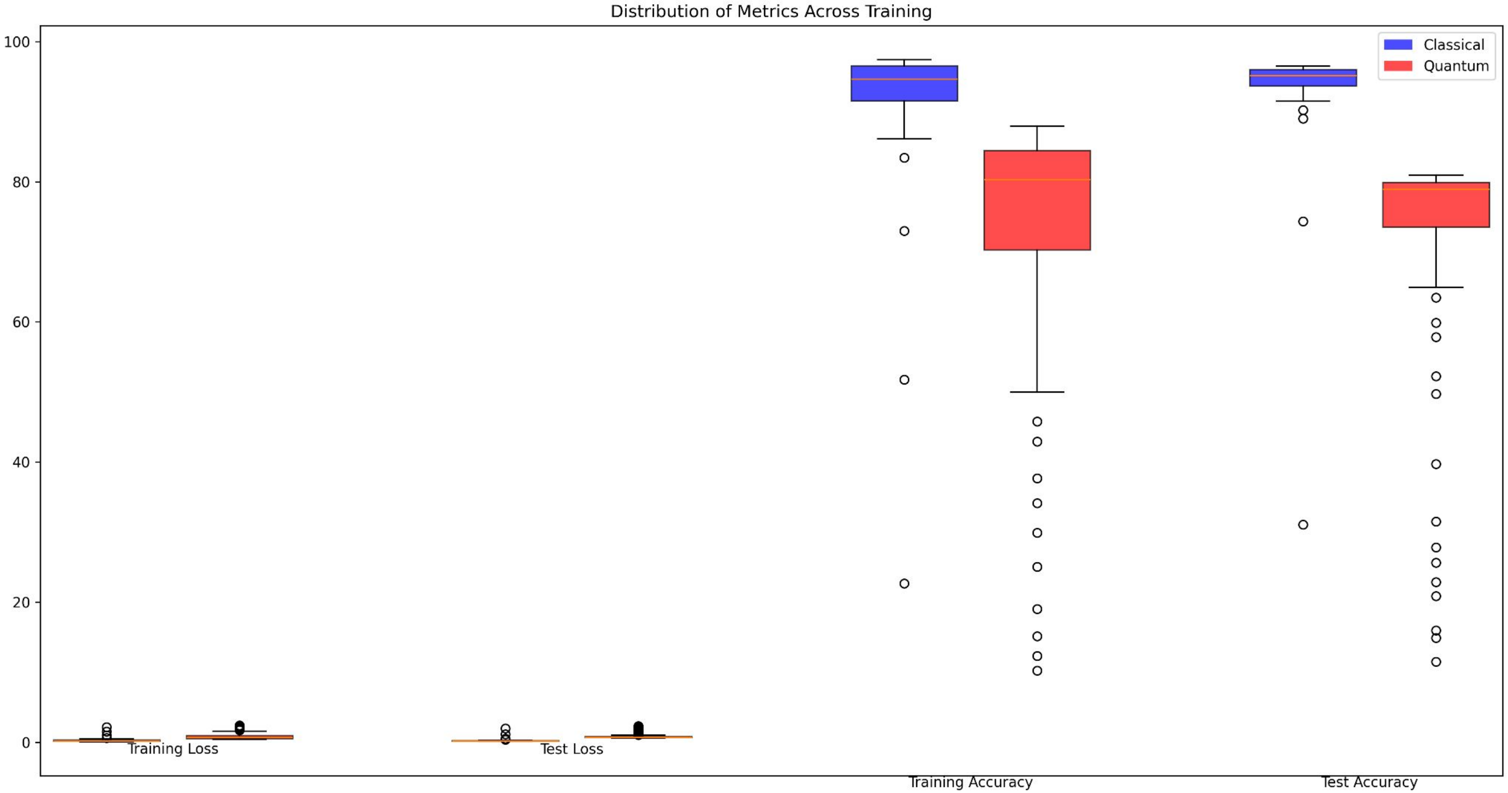
The significant F-statistic and p-value suggest a meaningful difference in test loss, with classical models outperforming quantum models by achieving lower loss on unseen data.

The classical model demonstrates higher training accuracy, and the significant p-value confirms that this difference is statistically significant.

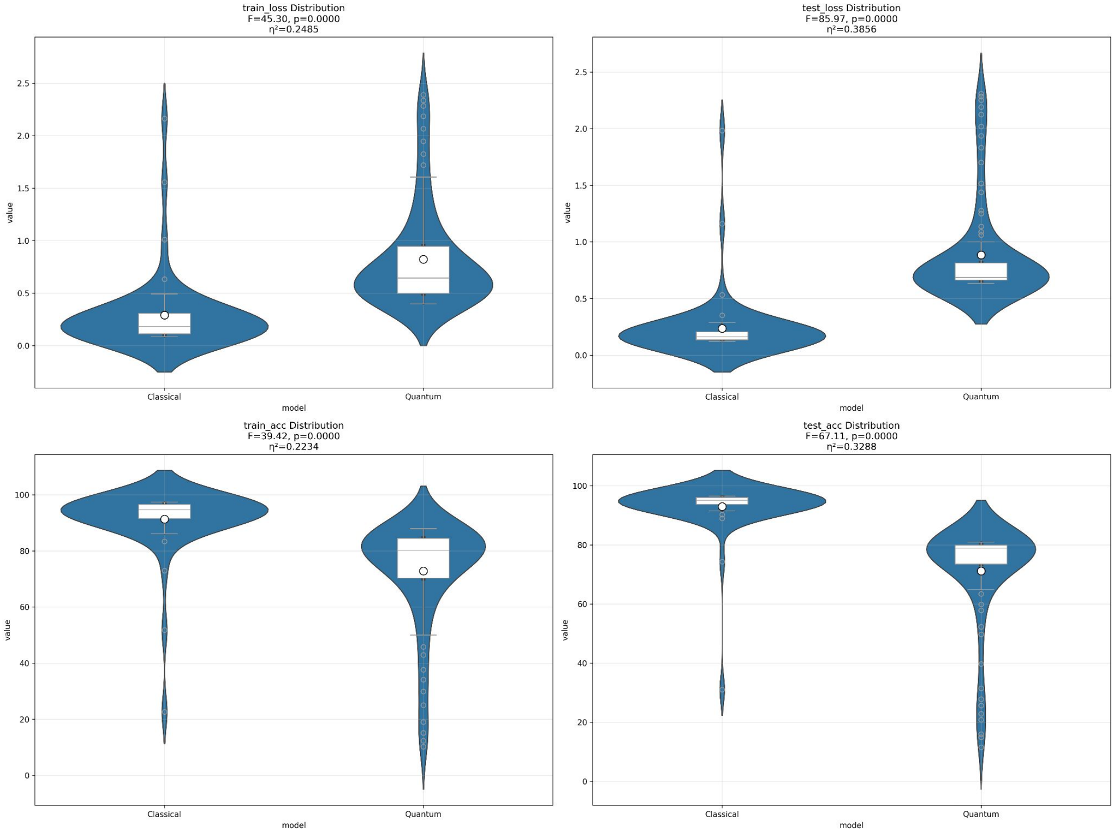
The classical model also achieves higher test accuracy, with the significant p-value indicating that this difference is not due to random chance.

```
Train Loss ANOVA Results:
F-statistic: 45.3020
p-value: 0.0000
Classical: 0.2900 ± 0.3646
Quantum: 0.8230 ± 0.4841

Test Loss ANOVA Results:
F-statistic: 85.9706
p-value: 0.0000
Classical: 0.2368 ± 0.2931
Quantum: 0.8839 ± 0.4379

Train Accuracy ANOVA Results:
F-statistic: 39.4179
p-value: 0.0000
Classical: 91.3112 ± 12.2197
Quantum: 72.8258 ± 18.5289

Test Accuracy ANOVA Results:
F-statistic: 67.1139
p-value: 0.0000
Classical: 92.9890 ± 9.4393
Quantum: 71.1483 ± 17.3265
```

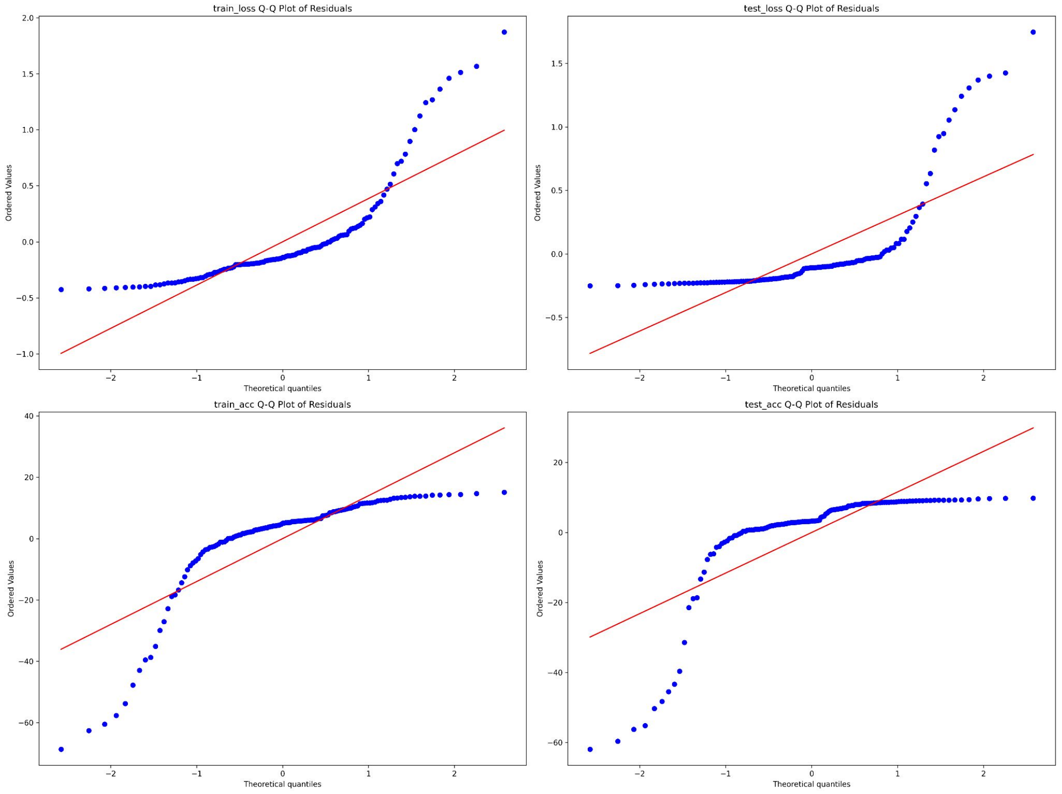
Comparing The Models



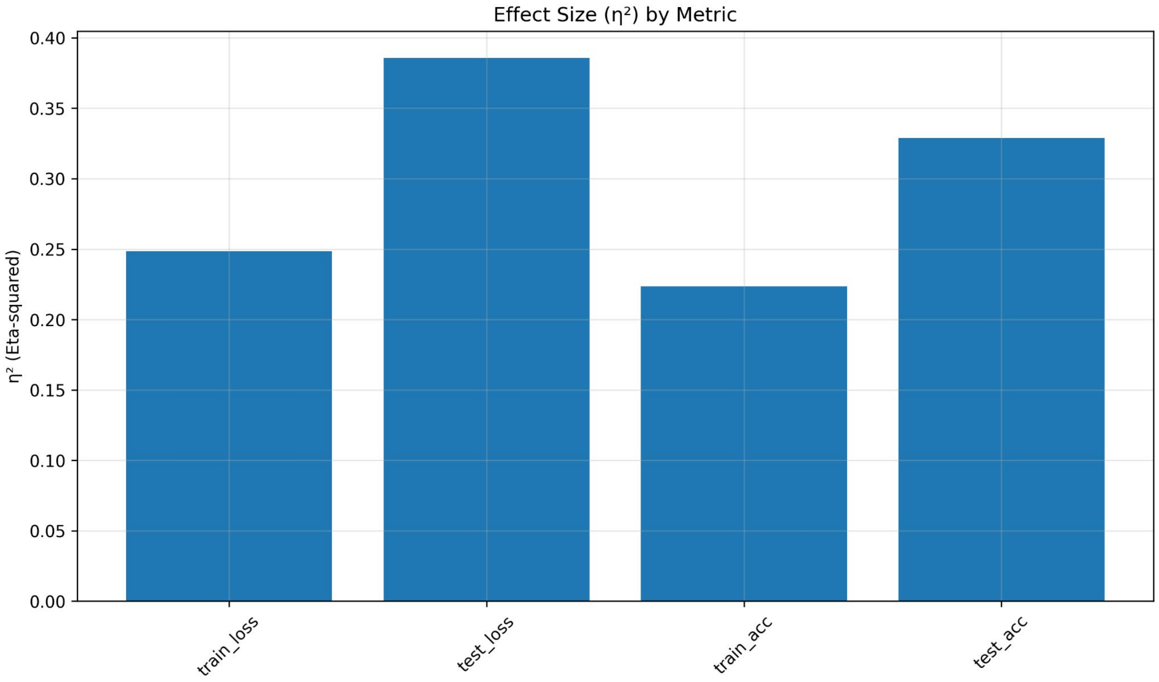
Comparing The Models



Comparing The Models



Comparing The Models



Conclusion

Conclusion

Classical NAM consistently outperformed quantum pipelines in accuracy, stability, and reliability.

QHED with FRQI was less stable than the classical baseline, with large performance variance across runs. This suggests it's **sensitive to noise**, likely due to its amplitude-based encoding.

In some of cases, specially with the digit 1, the quantum model succeeded where classical failed — mostly on noisy or ambiguous digits. This hints at **niche strengths** in certain edge-based cases.

Overall, quantum models using QHED are **not yet competitive in general performance**, but they might capture patterns that classical models overlook.

The challenge ahead is **reducing quantum encoding noise** and finding ways to combine quantum selectivity with classical robustness.

Next Steps

- Implement noise mitigation to FRQI + QHED
- Play with the NAM architecture: Does more non-linearity make quantum encodings perform better?
- Perform Gate Optimization on FRQI
- Perform a comprehensive XAI analysis to understand what the quantum features focus on compared to the classical ones.

References

[1] 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.
[2] Y. Zhang, M. Li, and Q. Su, "NEQR: A novel enhanced quantum representation of digital images," Quantum Inf. Process., vol. 12, no. 8, pp. 2833–2860, 2013.
[3] P. Q. Le, F. Dong, and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," Quantum Inf. Process., vol. 10, no. 1, pp. 63–84, 2011.
[4] Y. Li and J. Li, "Quantum Hadamard Edge Detection of Digital Images," arXiv:1703.04040, 2017.
[5] A. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton, "Neural Additive Models: Interpretable Machine Learning with Neural Nets," in Adv. Neural Inf. Process. Syst., vol. 33, 2020.
[6] M. Rath and H. Date, "Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," EPJ Quantum Technology, vol. 11, no. 72, 2024.
[7] Ranga, D., Prajapat, S., Akhtar, Z., Kumar, P., & Vasilakos, A. V. (2024). Hybrid quantum–classical neural networks for efficient MNIST binary image classification. *Mathematics, 12*(23), 3684.
[8] Tomal, S. M. Y. I., Shafin, A. A., Afaf, A., & Bhattacharjee, D. (2024). Quantum convolutional neural network: A hybrid quantum-classical approach for Iris dataset classification. arXiv preprint arXiv:2410.16344
[9] Chen, S. Y.-C., Huang, C.-M., Hsing, C.-W., & Kao, Y.-J. (2021). An end-to-end trainable hybrid classical-quantum classifier. Machine Learning: Science and Technology, 2(3), 035014.

Thank you!