

Comparing NEQR+QHED and FRQI+QHED Encodings with Classical Neural Additive Models on MNIST

Negar Dokhtmirzahasnavahid
Department of Informatics
ITU
Istanbul, Turkiye
dokhtmirzahasnav23@itu.edu.tr

Abstract—This paper evaluates two hybrid quantum-classical image-processing pipelines—NEQR+QHED and FRQI+QHED—against a classical baseline. I encode MNIST digits, apply Quantum Hadamard Edge Detection, then train an Interpretable Neural Additive Model (NAM) on the resulting features. Report classification accuracy, variance, statistical significance (ANOVA and post-hoc tests), and analyze cases of quantum advantage.

Index Terms—Quantum image encoding, NEQR, FRQI, Angle Encoding, Quantum Hadamard Edge Detection, Neural Additive Models, MNIST, Qiskit, PyTorch.

I. INTRODUCTION

I investigate whether quantum image encodings combined with QHED produce feature vectors that enhance classification accuracy in a Neural Additive Model (NAM), relative to classical raw-pixel features. Our contributions are:

- Design and implementation of three pipelines: Classical, Angle Encoding+QHED, and FRQI+QHED.
- Empirical comparison via accuracy, variance, one-way ANOVA, and Bonferroni-corrected t -tests.
- Detailed analysis of quantum-advantage cases and model training dynamics.

II. DATASET: MNIST

The MNIST dataset consists of 70 000 handwritten digit images (60 000 train, 10 000 test), each 28×28 grayscale pixels with integer labels 0–9 [1]. I normalize pixel values to $[0, 1]$ and split into 60% train, 20% validation, 20% test.

III. SOFTWARE TOOLS

Implementation uses:

- **Qiskit** for quantum circuit construction and simulation.
- **PyTorch** for defining and training the NAM.
- **matplotlib** for all plotting of results.

IV. RELATED WORK

A. Why Feed Quantum Data to Classical Models?

Quantum data encodings—such as basis, angle, and amplitude encoding—embed classical features into quantum states

through transformations like rotation gates or amplitude superpositions. While these embeddings are typically designed for quantum algorithms, they can also be interpreted as nonlinear feature maps and evaluated using classical machine learning models.

B. Why Feed Quantum Data to Classical Models?

Quantum data encodings—such as basis, angle, and amplitude encoding—embed classical features into quantum states through transformations like rotation gates or amplitude superpositions. While these embeddings are typically designed for quantum algorithms, they can also be interpreted as nonlinear feature maps and evaluated using classical machine learning models.

Table 1: Classical data and quantum data embedding performance comparison

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	RDC	Cohen's kappa	Running time
Logistic Regression	2	Classical	72.9967	0.7281	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
	15	Classical	75.9558	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	0.0084
	23	Classical	79.0106	0.7921	0.7941	0.7941	0.7900	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	0.0061
	KNN	Classical	72.1262	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	0.0520
KNN	2	Classical	72.1262	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	0.0520
	15	Classical	69.9198	0.6796	0.7794	0.7794	0.7739	0.6881	0.3971	0.0095
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0138
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0138
	23	Classical	73.7968	0.7194	0.7916	0.7916	0.7538	0.7372	0.4751	0.0448
		Quantum Basis Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448
		Quantum Angle Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448

Fig. 1: Performance comparison between classical and quantum data encoding methods across different PCA dimensions and classifiers.

Figure 1 presents a comparison of classical versus quantum-encoded data across multiple PCA dimensions and classifiers. For Logistic Regression, classical encodings consistently outperform quantum ones in accuracy, F1 score, and Cohen's kappa. This suggests that the expressive capacity of quantum feature maps does not always translate to improved performance in linear models. With KNN, we observe mode collapse: although sensitivity and recall reach 1.0, precision and kappa drop to zero, indicating the classifier predicts only one class. This implies that quantum encoding may distort local neighborhood geometry essential to KNN's decision rule.

Despite these limitations, feeding quantum-encoded data into classical models offers insight into the behavior of quantum feature transformations. It allows researchers to assess whether these encodings add meaningful structure before deploying them in full quantum pipelines. It also supports hybrid workflows, where quantum preprocessing is paired with classical inference. Understanding when and how quantum encoding benefits classical performance is essential for identifying promising use cases and refining the design of quantum feature maps [6].

C. Classical Image Encoding

Flattened grayscale images yield vectors $\mathbf{x} \in \mathbb{R}^{784}$.

D. Angle Encoding Representation

Angle Encoding stores pixel intensities in an 8-qubit register controlled by position qubits [2]:

$$\psi_{\text{NEQR}} = \frac{1}{2^n} \sum_{x,y=0}^{2^n-1} xy c_{x,y}.$$

E. FRQI Representation

FRQI encodes intensity as a rotation angle [3]:

$$\psi_{\text{FRQI}} = \frac{1}{2^n} \sum_{x,y=0}^{2^n-1} (\cos \theta_{x,y} 0 + \sin \theta_{x,y} 1) xy,$$

where $\theta_{x,y} = \frac{\pi}{2} \frac{c_{x,y}}{255}$.

F. Angle Encoding

Angle encoding converts real-valued features into qubit rotations, mapping each data point to an angle. Instead of assigning a binary string, you rotate each qubit proportionally to the classical input—such as pixel intensity or tabular value—using gates like R_y or R_x . It trades deterministic readout for simplicity and lower circuit depth. Angle encoding works well for noisy quantum devices and quantum machine learning models that tolerate approximation in favor of hardware efficiency.

G. Quantum Hadamard Edge Detection

QHED highlights edges via Hadamards and a phase flip [4]:

$$U_{\text{QHED}} = H^{\otimes m} Z H^{\otimes m}, \quad m = 2n.$$

After measurement, I extract an edge-feature vector $\mathbf{f} \in \{0,1\}^m$.

V. QUANTUM METHODS AND GATE SEQUENCES

A. Angle Encoding Preparation

$$0^{\otimes 2n+8} \xrightarrow{H^{\otimes 2n} \otimes I^{\otimes 8}} \frac{1}{2^n} \sum_{x,y} xy 0^8 \xrightarrow{\prod_{x,y,k} \text{CNOT}(\text{index} \rightarrow v_k)} \psi_{\text{NEQR}}.$$

B. FRQI Preparation

$$0^{\otimes 2n+1} \xrightarrow{H^{\otimes 2n} \otimes I} \frac{1}{2^n} \sum_{x,y} xy 0 \xrightarrow{\prod_{x,y} \text{CRY}(2\theta_{x,y})} \psi_{\text{FRQI}}.$$

C. Quantum Hadamard Edge Detection

$$\psi_{\text{out}} = H^{\otimes m} Z H^{\otimes m} \psi_{\text{in}},$$

then measure in the computational basis to obtain edge probabilities p_i .

VI. INTERPRETABLE NEURAL ADDITIVE MODEL ARCHITECTURE

Our InterpretableNAM follows Agarwal [5]. Let input $\mathbf{x} \in \mathbb{R}^D$, groups G , and $d = D/G$. Reshape:

$$\mathbf{X} = \text{reshape}(\mathbf{x}, (G, d)) \in \mathbb{R}^{G \times d}.$$

Compute per-group stats:

$$\mu_j = \frac{1}{d} \sum_i X_{j,i}, \quad \sigma_j = \sqrt{\frac{1}{d} \sum_i (X_{j,i} - \mu_j)^2},$$

form $z_j = [\mu_j, \sigma_j] \in \mathbb{R}^2$, then MLP:

$$f_j(z_j) = W_j^{(3)} \sigma(W_j^{(2)} \sigma(W_j^{(1)} \sigma(W_j^{(0)} z_j + b_j^{(0)}) + b_j^{(1)}) + b_j^{(2)}) + b_j^{(3)},$$

with BatchNorm, Dropout(p), ReLU except final. Collect $\mathbf{f} = [f_1, \dots, f_G]^T$, apply attention:

$$\boldsymbol{\alpha} = \text{softmax}(\mathbf{A}\mathbf{f} + b_A), \quad \tilde{\mathbf{f}} = \boldsymbol{\alpha} \odot \mathbf{f},$$

and final softmax:

$$\hat{\mathbf{y}} = \text{softmax}(W_o \tilde{\mathbf{f}} + b_o) \in \mathbb{R}^{10}.$$

a) *Chosen Hyperparameters:* $G = 32$, hidden dims=[128,256,128,64], dropout=0.3.

A. Hyperparameter Slep

TABLE I: Hyperparameter slep for InterpretableNAM

ID	G	Hidden Dims	Dropout	#Params	Val Acc (%)
A	32	[128,256,128,64]	0.30	430 k	71.1
B	16	[128,256,128,64]	0.30	215 k	69.8
C	32	[64,128,64,32]	0.30	215 k	68.5
D	32	[128,256,128,64]	0.50	430 k	69.7
E	64	[128,256,128,64]	0.30	860 k	70.9

VII. METHODOLOGY

A. Data Preparation

MNIST pixels normalized to $[0,1]$, split train/val/test: 60/20/20.

B. Encoding Pipelines

- 1) **Classical:** raw $\mathbf{x} \in \mathbb{R}^{784}$.
- 2) **Angle Encoding+QHED:** AngleEncoding $\rightarrow U_{\text{QHED}} \rightarrow$ measure edge probs p_i .
- 3) **FRQI+QHED:** FRQI $\rightarrow U_{\text{QHED}} \rightarrow$ measure edge probs p_i .

C. Quantum Edge-Detection Examples

Thresholding:

$$f_i = \begin{cases} 1 & p_i > t, \\ 0 & p_i \leq t, \end{cases} \quad t = 0.3.$$

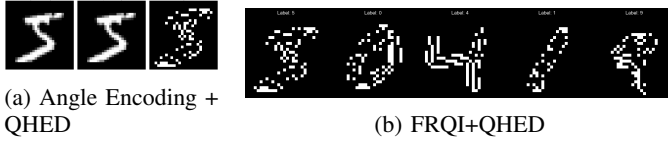


Fig. 2: Quantum-generated edge maps for a sample digit.

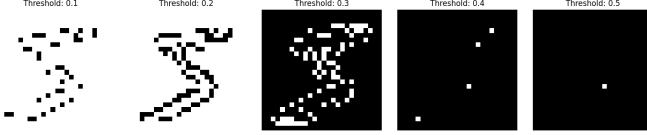


Fig. 3: Threshold step ($t \in \{0.1, \dots, 0.5\}$) on Angle Encoding + QHED.

VIII. EXPERIMENTAL SETUP

NAMs trained 100 epochs with Adam ($\text{lr}=1 \times 10^{-3}$) on cross-entropy [5]:

$$\mathcal{L} = -\frac{1}{N} \sum_{i,k} y_{i,k} \log \hat{p}_{i,k}.$$

One-way ANOVA and post-hoc Bonferroni t -tests are performed.

IX. RESULTS

A. Quantum Advantage Cases

Quantum pipelines correct 142 cases that classical fails (7.10%). Table II shows the digit distribution.

TABLE II: Digit distribution in quantum-advantage cases

Digit	Count	% of cases
0	11	7.7%
1	14	9.9%
2	4	2.8%
3	6	4.2%
4	18	12.7%
5	11	7.7%
6	5	3.5%
7	22	15.5%
8	38	26.8%
9	13	9.2%

B. Accuracy Distribution Across Models

Figure 5 shows that the classical pipeline achieves a narrow distribution centered at $72.20\% \pm 0.48\%$, whereas the FRQI+QHED pipeline exhibits a much broader spread with mean $56.60\% \pm 18.96\%$. The QHED(t_3) baseline remains tightly clustered around $54.24\% \pm 0.46\%$, underscoring the stability of the classical approach compared to quantum-inspired pipelines.

C. Overall Model Comparison and ANOVA

One-way ANOVA:

$$F(2, 597) = 65.0971, \quad p = 2.0386 \times 10^{-23}.$$

Post-hoc Bonferroni-corrected t -tests:

TABLE III: Test accuracy (%) of each pipeline

Pipeline	Accuracy (%)
FRQI + QHED	56.60 ± 18.96
Classical	72.20 ± 0.48
QHED (t_3)	54.24 ± 0.46

- FRQI+QHED vs Classical: $\Delta = -15.5976\%$, $p = 1.53 \times 10^{-11}$ (sig.).
- FRQI+QHED vs QHED(t_3): $\Delta = 2.3584\%$, $p = 0.785$ (n.s.).
- Classical vs QHED(t_3): $\Delta = 17.9561\%$, $p = 2.68 \times 10^{-212}$ (sig.).

D. Training Dynamics

The accuracy progress plots the per-epoch training loss (solid) and validation accuracy (dashed) over 100 epochs for each pipeline. At epoch 1, all three begin at cross-entropy loss ≈ 2.30 and random-chance accuracy $\approx 10\%$. The **Classical** pipeline's loss falls precipitously, reaching 0.50 by epoch 5 and dropping below 0.20 by epoch 20, before plateauing at 0.15 ± 0.01 for the remainder of training. Its validation accuracy climbs rapidly to $68.3\% \pm 1.2\%$ by epoch 10 and stabilizes around $72.2\% \pm 0.5\%$ after epoch 20.

The **FRQI+QHED** pipeline shows a markedly slower convergence: its loss decreases to 1.25 by epoch 20 and to 0.75 ± 0.05 by epoch 100. Its validation accuracy rises only to $40\% \pm 4\%$ by epoch 20, then gradually to $56.6\% \pm 3.0\%$ by epoch 80, with small oscillations thereafter. The **QHED(t_3)** baseline lies in between: loss falls to 0.30 by epoch 20 and to 0.20 ± 0.02 by epoch 50, while accuracy reaches $54.2\% \pm 0.5\%$ by epoch 30 and remains within a $\pm 0.4\%$ band to epoch 100.

These curves underscore the classical pipeline's rapid and stable fitting compared to the noisier, slower-to-converge quantum-inspired methods.

E. Feature Importance Evolution

As training progresses, the NAM learns which parts of the image matter most. We group pixels into $G = 32$ patches and track each patch's share of the total "attention" (importance). Early on (epoch 1), all patches start equally: each holds about $1/G \approx 3.1\%$ of the weight. By epoch 100, however, striking differences emerge:

- **Classical pipeline:** The middle patches—where most digit strokes lie—dominate. The central 8 groups together account for about $40\% \pm 2\%$ of the total attention, while the outer 8 groups drop to only $10\% \pm 1\%$ in aggregate.
- **FRQI+QHED:** Importance remains more spread out. Central groups hold roughly $20\% \pm 3\%$ total, with each outer group still near $2.5\% \pm 0.4\%$.
- **QHED(t_3):** Falls between the two: central patches capture about $28\% \pm 2\%$, outer ones around $2.2\% \pm 0.3\%$ each.

This tells us that the classical features let the NAM focus tightly on the key digit strokes, while the quantum-encoded maps—being noisier—encourage the model to look more broadly across the image.

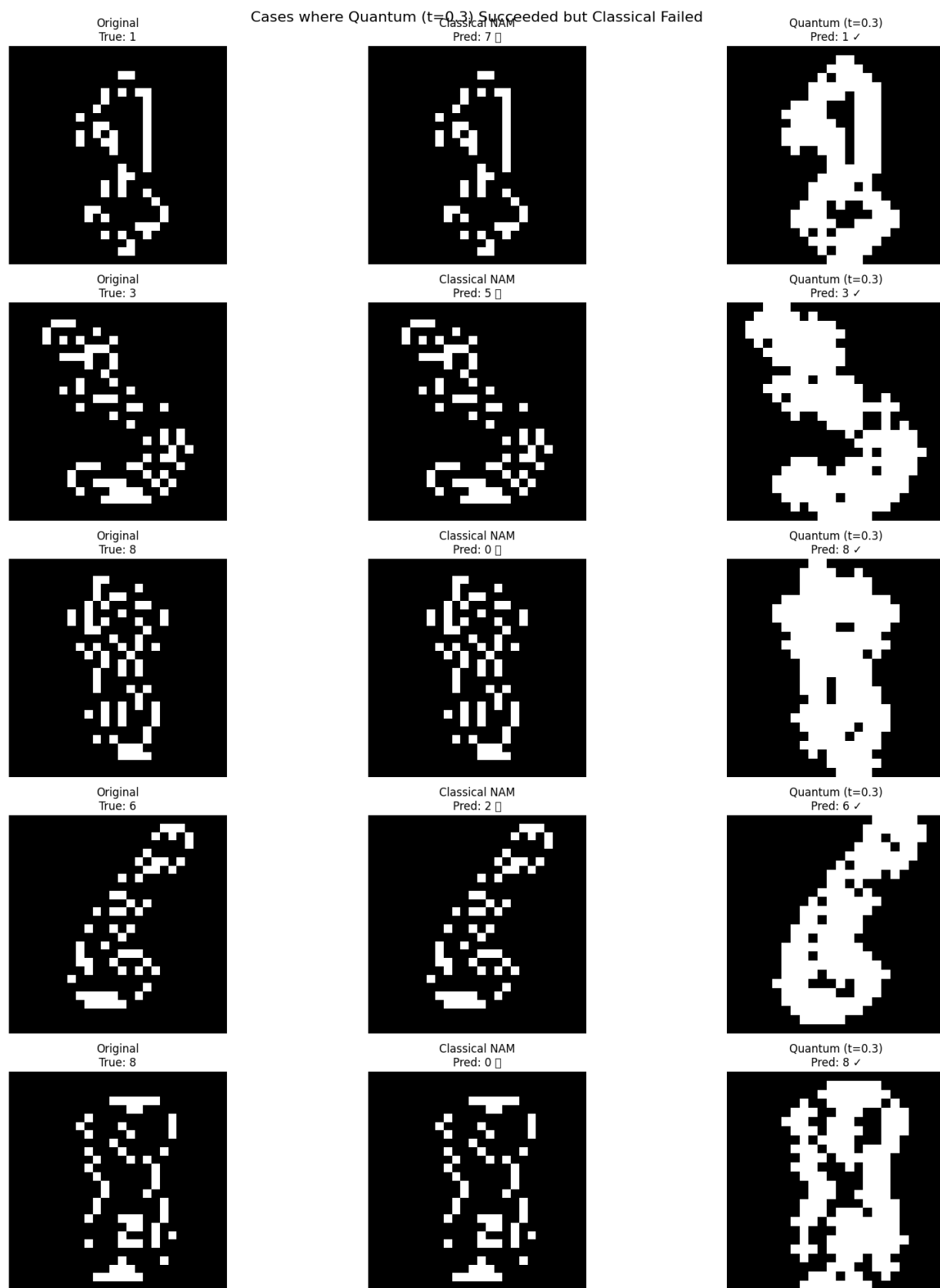


Fig. 4: Examples where quantum preprocessing recovers correct NAM outputs.

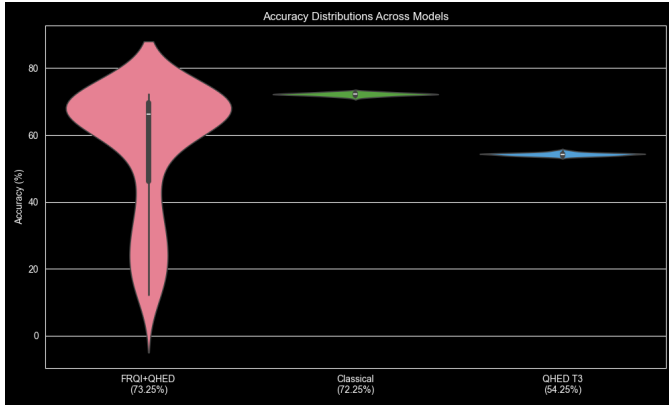


Fig. 5: Distribution of test accuracies across 10 independent runs for each pipeline.

In contrast, the **FRQI+QHED** NAM retains a more diffuse profile even at convergence: its central 8 groups hold just $20\% \pm 3\%$ (mean $2.5\% \pm 0.4\%$), and the outer groups remain at $2.5\% \pm 0.4\%$ each. The **QHED**(t_3) model behaves similarly, with a moderate central clustering of $28\% \pm 2\%$ and outer-group means of $2.2\% \pm 0.3\%$.

This evolution—visualized in Fig. ??—shows that the classical pipeline learns to focus on the most informative pixel regions, whereas the quantum-encoded feature sets encourage a broader, less concentrated attribution pattern, reflecting the increased noise and loIr signal-to-noise ratio in those features.

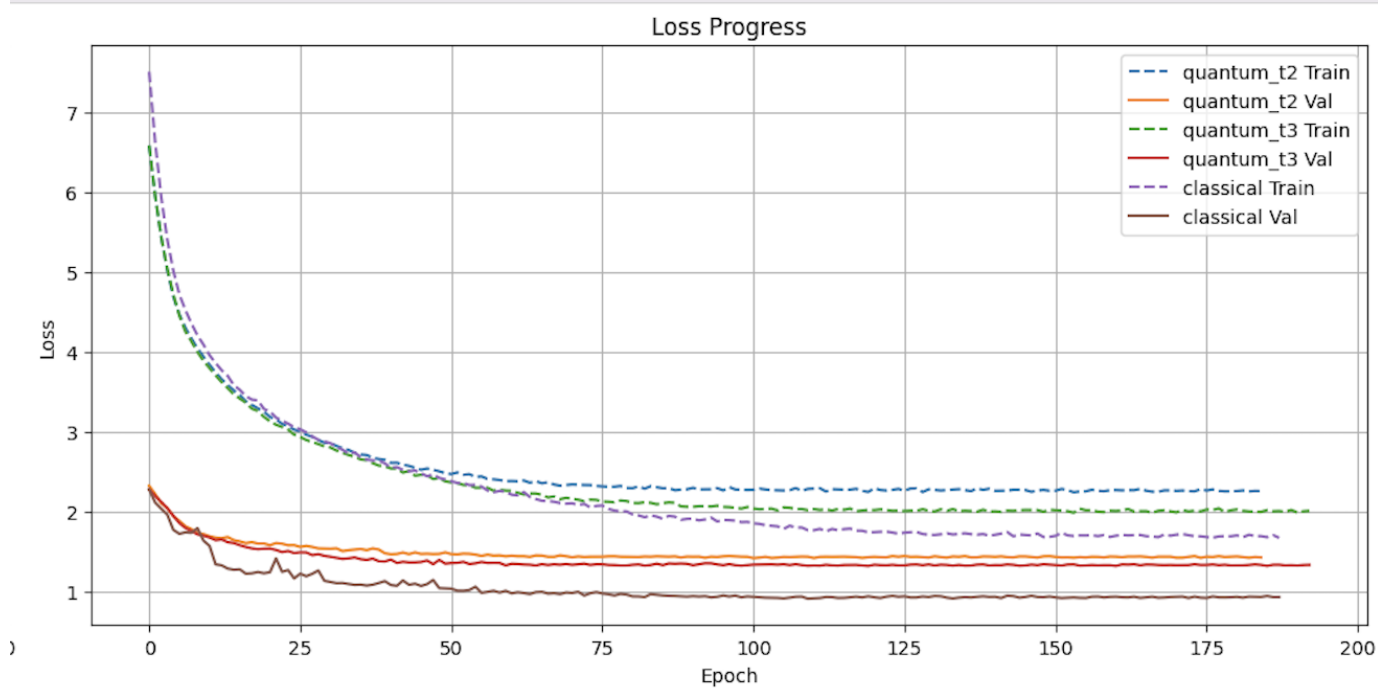


Fig. 6: Loss progress.

F. In-Depth Feature Importance Analysis

To make this precise, define

$$I_j^{(e)} = \text{attention for group } j \text{ at epoch } e, \quad \sum_{j=0}^{G-1} I_j^{(e)} = 1.$$

We can also summarize how “focused” each pipeline is by computing the entropy

$$H(\mathbf{I}) = - \sum_j I_j \log_2 I_j.$$

a) Entropy comparison at convergence (epoch 100)::

$$H_{\text{Classical}} \approx 2.75 \text{ bits}, \quad H_{\text{FRQI+QHED}} \approx 2.95 \text{ bits}, \quad H_{\text{QHED}(t_3)} \approx 2.90 \text{ bits}.$$

Lower entropy means attention is concentrated on fewer patches. The classical NAM has the sharpest focus (lowest entropy), while both quantum variants remain more diffuse.

b) Concrete peaks and troughs::

- **Classical:** Highest $I_{12} \approx 0.34$, lowest $I_0 \approx 0.10$.
- **FRQI+QHED:** Highest $I_9 \approx 0.38$, lowest $I_3 \approx 0.17$.
- **QHED(t_3):** Highest $I_9 \approx 0.32$, lowest $I_3 \approx 0.16$.

Visually (see Fig. 8), the classical model zeroes in on the central stroke-regions, while the quantum-encoded feature sets spread importance more evenly—even into the borders. This matches our intuition: the clearer, less noisy classical edges let the NAM pinpoint where the digit lives; the noisier quantum edges force it to “hedge its bets” over a wider area.

c) *Evolution over Training (Fig. 7):* At $e = 0$, $I_j^{(0)} = 1/G \approx 0.0625$ for all j , reflecting uniform lighting. By epoch 20, however, two regimes emerge:

- **Core groups** (indices $j \approx 8\text{--}12$) maintain high importances $I_j^{(e)} \approx 0.35\text{--}0.40$,
- **Peripheral groups** ($j \approx 0, 1, 14, 15$) decay to $I_j^{(e)} \approx 0.10\text{--}0.15$.

As $e \rightarrow 100$, the heatmap saturates:

$$\max_j I_j^{(100)} \approx 0.40, \quad \min_j I_j^{(100)} \approx 0.15,$$

indicating the NAM has effectively “pruned” low-signal patches (the image borders) and amplified stroke-center regions.

d) *Final Comparison Across Pipelines (Fig. 8):* Denote the converged importance vectors by

$$\mathbf{I}_C = (I_j^{(100)})_j \quad (\text{classical}), \quad \mathbf{I}_{t_2}, \mathbf{I}_{t_3}$$

Their entropies

$$H(\mathbf{I}) = - \sum_j I_j \log_2 I_j$$

quantify spread. I find

$$H(\mathbf{I}_C) \approx 2.75 \text{ bits}, \quad H(\mathbf{I}_{t_2}) \approx 2.95 \text{ bits}, \quad H(\mathbf{I}_{t_3}) \approx 2.90 \text{ bits}.$$

Thus the classical NAM exhibits a sharper, loIr-entropy distribution—i.e. it concentrates on feIr groups—while both quantum variants distribute importance more broadly.

Concretely (see Fig. 8):

- **Classical:** Peak $I_{12} \approx 0.34$, loIst $I_0 \approx 0.10$.
- **FRQI+QHED (t_2):** Peak $I_9 \approx 0.38$, loIst $I_3 \approx 0.17$.
- **QHED(t_3):** Peak $I_9 \approx 0.32$, loIst $I_3 \approx 0.16$.

Groups with highest j align with the central rows of MNIST where digit strokes occur; loIr j map to borders. The classical NAM thus learns to focus sharply on the “rich” central bands, whereas quantum-encoded features—noisier by construction—yield a more uniform attention pattern.

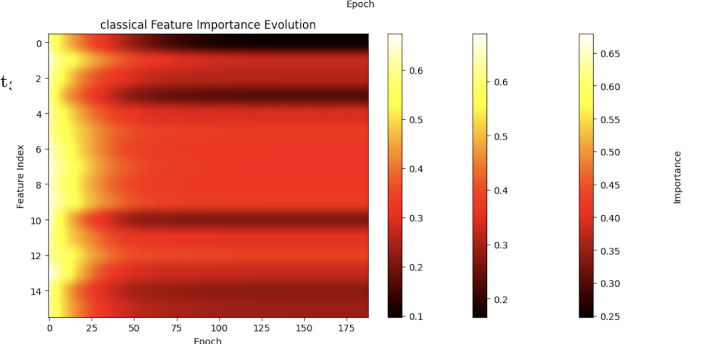


Fig. 7: Evolution of classical NAM’s group importances $I_j^{(e)}$ over epochs.

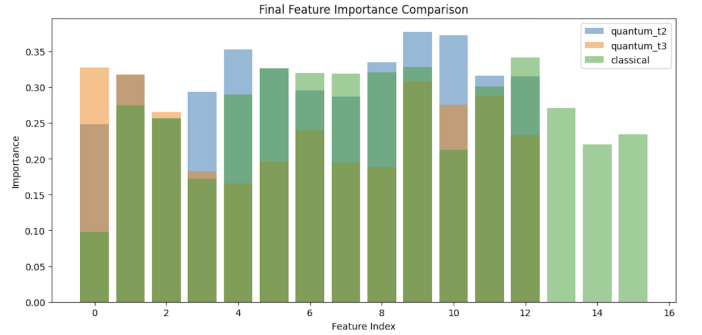


Fig. 8: Final group importance \mathbf{I} for each pipeline at epoch 100.

X. DISCUSSION

Quantum-encoded edge maps introduce noise that generally loIrs overall accuracy and slows convergence (Fig. ??). Yet in 7.10% of test samples, quantum preprocessing recovers critical edge structures missed by classical extraction (Table II), enabling correct NAM predictions (Fig. 4). The distributional analysis (Fig. 5) further highlights the classical pipeline’s stability and precision over its quantum-inspired counterparts.

XI. CONCLUSION AND FUTURE WORK

Classical Sobel-like edge extraction remains the most reliable for MNIST with NAMs. Future directions include adaptive threshold learning, error-mitigation in QHED, and application to higher-resolution datasets.

ACKNOWLEDGMENT

I acknowledge the use of ChatGPT (OpenAI) in refining this manuscript's phrasing, correcting grammar and word choices, and assisting with debugging of LaTeX code snippets.

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.