

ATP: Adaptive Threshold Pruning for Efficient Data Encoding in Quantum Neural Networks

Mohamed Afane^{1,*} Gabrielle Ebbrecht¹ Ying Wang² Juntao Chen^{1,*} Junaid Farooq³
¹Fordham University ²Stevens Institute of Technology ³University of Michigan-Dearborn

Abstract

Quantum Neural Networks (QNNs) offer promising capabilities for complex data tasks, but are often constrained by limited qubit resources and high entanglement, which can hinder scalability and efficiency. In this paper, we introduce Adaptive Threshold Pruning (ATP), an encoding method that reduces entanglement and optimizes data complexity for efficient computations in QNNs. ATP dynamically prunes non-essential features in the data based on adaptive thresholds, effectively reducing quantum circuit requirements while preserving high performance. Extensive experiments across multiple datasets demonstrate that ATP reduces entanglement entropy and improves adversarial robustness when combined with adversarial training methods like FGSM. Our results highlight ATP's ability to balance computational efficiency and model resilience, achieving significant performance improvements with fewer resources, which will help make QNNs more feasible in practical, resource-constrained settings.

1. Introduction

Quantum machine learning (QML) has gained attention for its ability to solve problems that are difficult for classical models by using the unique properties of quantum systems [1]. It has shown practical benefits in fields such as chemistry [32], optimization [13], and others involving high-dimensional or structured data [7, 37]. However, quantum algorithms are still constrained by the availability of qubits, noise, and decoherence in hardware [19, 29]. Because of quantum hardware's sensitivity to the environment, qubits are prone to premature collapse that degrades the fidelity of computation. Qubits may also be disturbed by signals intended to alter the state of another qubit in close proximity [4] which reduces stability in hardware with many qubits and complicates the training of quantum models. As a result, shallower Quantum Neural Networks (QNNs) are often more practical. [9, 38].

*Corresponding authors: {mafane, jchen504}@fordham.edu

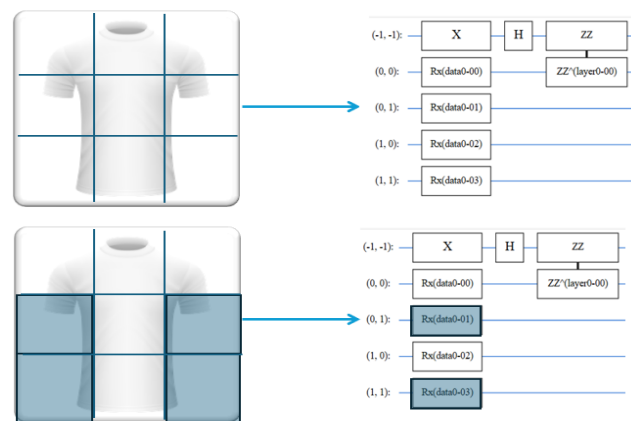


Figure 1. Demonstration of the adaptive threshold pruning framework. The original image (top) is split into a 3x3 grid, with each section assessed for information density. The pruned image (bottom) shows filtered regions that fall below a defined intensity threshold in blue indicating areas that do not contribute significantly to the classification task, effectively freeing qubits and optimizing resources by focusing only on key areas of high relevance.

Efficient data encoding into quantum states remains a crucial bottleneck for achieving scalability and accuracy in QML applications [30, 41]. For instance, limiting qubit interactions can reduce crosstalk and enhance stability, which is essential for noise resilience [2]. While using larger and deeper QNNs can enhance a model's ability to capture complex data relationships and potentially improve performance, increasing encoding depth introduces new challenges. As the depth of the encoding circuit grows, encoded quantum states tend to converge towards a maximally mixed state at an exponential rate, becoming increasingly indistinguishable from random noise [21]. This concentration effect leads to a lack of state diversity and the emergence of barren plateaus, where gradients vanish, significantly hindering the model's learning process [26]. These effects are especially problematic when encoding is not adapted to the structure or scale of the data. Thus, QNNs must balance expressibility with computational demands and sensitivity to barren plateaus [25].

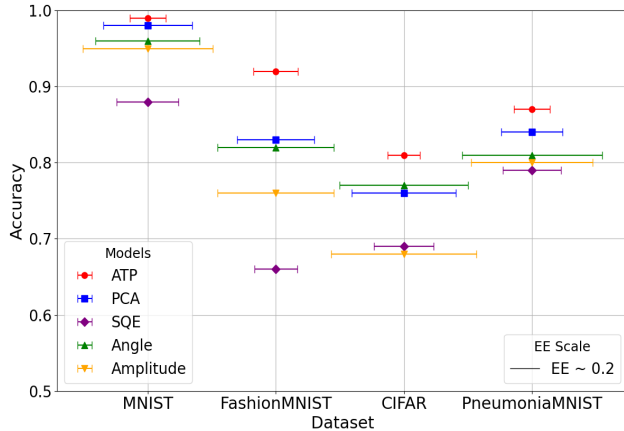


Figure 2. Average accuracy across four datasets, comparing the performance of various encoding methods: Adaptive Threshold Pruning (ATP), Principal Component Analysis (PCA), Single Qubit Encoding (SQE), Angle, and Amplitude encoding. Horizontal error bars represent entanglement entropy (EE), with longer bars indicating higher entanglement. ATP generally achieves the highest accuracy with lower EE.

While efficient data encoding can improve performance in classical models, such as by reducing training time or resource use, it is far more critical in QML. In quantum models, encoding directly impacts whether meaningful patterns can be learned in the first place, since poor encoding can waste limited qubit resources or amplify noise to the point that the model fails entirely [22]. For instance, some methods only encode states with significant contributions to the model, allowing qubit resources to focus on meaningful data and enabling more effective scaling [34]. Amplitude encoding is one such approach, representing data compactly to reduce the number of required qubits, though it limits the types of data and operations that can be used [33]. Other strategies, such as Qubit Lattice, a direct encoding method with low circuit depth, and FRQI, a compact scheme that uses fewer qubits, demonstrate different tradeoffs between qubit efficiency and processing flexibility [22]. In general, encoding strategies that reduce resource overhead can optimize computation and limit potential errors by requiring fewer qubits to operate.

In this context, entanglement entropy (EE) serves as a key metric for assessing a model’s encoding efficiency. EE measures the degree of correlation between different qubits in a quantum system by quantifying the information shared between subsystems, typically calculated as the von Neumann entropy of a reduced density matrix of one part of a bipartite system. Higher EE indicates a greater level of interdependence between qubits, which can be advantageous for capturing complex data structures, but excessive entanglement may increase computational complexity and lead

to overfitting. Anagiannis and Cheng, for instance, found that as EE rises in q-convolutional neural networks, the model’s cost function decreases, suggesting a link between structured entanglement and efficient learning [3]. Similarly, Martyn et al. demonstrated that while certain levels of entanglement improve training accuracy in quantum models, over-entangling the qubits can lead to diminished generalization and higher resource demands [24]. Thus, EE not only provides insight into the information distribution across qubits but also helps balance model complexity and computational efficiency in quantum learning.

To address these challenges, we introduce Adaptive Threshold Pruning (ATP), a technique that optimizes the encoding process by adaptively pruning non-essential features, thereby reducing both qubit usage and entanglement in the circuit, as visualized by Figure 1. ATP sets dynamic thresholds for feature pruning, selectively removing low-variance data to maintain model performance while lowering EE. Figure 2 summarizes ATP’s performance compared to other encoding methods, demonstrating improvements in both accuracy and entanglement efficiency.

2. Related Works

Masking: In classical machine learning, masking strategies are commonly used in self-supervised learning by training models to reconstruct missing parts of the input from partial observations. Masked autoencoding, as used in the Masked Autoencoders (MAE) framework, achieves this by randomly hiding image patches and requiring the model to reconstruct them, encouraging the learning of meaningful representations without labels [14]. Although the primary objective is representation learning, MAE improves efficiency by processing only visible patches through a lightweight encoder-decoder setup. This reduction in input complexity helps scale models to large datasets with fewer computational demands. Similar to how MAE filters input through random masking, our ATP approach uses threshold-based pruning to discard low-variance regions before encoding, allowing quantum models to focus on the most relevant parts of the data while reducing resource usage [40].

Pruning: In contrast, pruning in QML has primarily been applied at the circuit level to reduce model complexity, addressing challenges such as noise and limited quantum resources. These techniques streamline QNNs by selectively removing parameters with minimal impact on accuracy, thereby enhancing computational efficiency and scalability. Hu et al. [16] propose an architecture compression method for quantum circuits, achieving reduced circuit depth with minimal accuracy loss, while Wang et al. [39] demonstrate how pruning supports scalability in quantum neural networks. Circuit-level pruning methods, such as those by Sim et al. [36] and Kulshrestha et al. [20], adaptively eliminate non-essential parameters, optimizing quan-

tum resources without compromising model performance. While architecture pruning addresses circuit complexity, efficient data encoding remains critical in quantum contexts, directly impacting qubit usage and entanglement management.

Data encoding: Several encoding methods have been developed to optimize qubit usage while preserving essential data features in quantum neural networks. For example, Single-Qubit Encoding (SQE) [10] minimizes resource requirements by encoding data into a single qubit rather than multiple qubits. This technique uses a series of Z-Y-Z axis rotations on the Bloch sphere, parameterized by input data, to capture spatial relationships with minimal qubit and parameter usage. Classical methods like Principal Component Analysis (PCA) [18] and autoencoding [11] also improve quantum model efficiency by reducing data dimensionality while retaining key features [17]. PCA identifies principal components with the highest variance, focusing on informative aspects of the data, while autoencoders learn compact representations through non-linear dimensionality reduction.

The choice of encoding strategy directly impacts the effectiveness of quantum methods by influencing qubit efficiency and circuit complexity. Angle encoding, commonly used for classification tasks, maps classical data to quantum gate rotations due to its simplicity [31]. Amplitude encoding, which embeds data as probability amplitudes, allows quantum representation of exponentially large datasets but presents scalability challenges. Basis encoding represents binary data as computational basis states, suitable for binary data but less practical for complex datasets. Hybrid methods, such as amplitude-angle encoding, aim to combine strengths of individual techniques but often increase circuit depth, posing issues on noisy hardware with limited coherence times [6]. These limitations emphasize the need for adaptable encoding methods like ATP that optimize encoding based on data relevance and variability.

Our approach: ATP distinguishes itself by dynamically pruning with adaptive thresholds. Unlike PCA, which applies linear transformations to capture data variance, ATP zeroes out less informative data points based on their relevance, adapting to each dataset’s unique structure. Unlike SQE, which minimizes resources by encoding data into a single qubit, ATP selectively retains expressive data features with minimal resource overhead. By concentrating quantum resources on essential data, ATP balances computational efficiency and entanglement requirements, offering a scalable solution tailored to the specific constraints and advantages of quantum systems.

To our knowledge, no prior work has addressed adaptive feature pruning within quantum data encoding. ATP thus fills this gap by offering a systematic approach to reduce quantum overhead while preserving model accuracy.

3. Preliminaries

3.1. Quantum Computing Fundamentals

Quantum computing exploits principles of quantum mechanics, providing a framework that differs profoundly from classical systems. In classical computing, data is represented by bits in distinct states of 0 or 1. Quantum computing, however, relies on *qubits*, which can exist in linear combinations of 0 and 1 states, represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. This property, known as *superposition*, allows qubits to perform concurrent calculations, making quantum algorithms inherently parallel [15, 35].

3.1.1 Entanglement and Measurement

Entanglement is a quantum phenomenon in which qubits become correlated in such a way that the state of one qubit directly influences the state of another, regardless of physical distance [28]. For two entangled qubits, the state can be expressed as:

$$|\psi\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle), \quad (2)$$

forming a maximally entangled Bell state. Entanglement is a crucial resource in QNNs, as it enables intricate data encoding and interaction patterns that cannot be replicated classically. However, the control and preservation of entanglement are challenging due to decoherence and noise, making efficient encoding schemes essential for managing entanglement and optimizing qubit usage [12].

3.2. Quantum Neural Networks

QNNs use quantum circuits with parameterized gates, known as PQC, to process data. A QNN layer can be defined by a series of parameterized unitary transformations [43]:

$$U(\theta) = \prod_j U_j(\theta_j), \quad (3)$$

where $\theta = \{\theta_j\}$ are tunable parameters. Each U_j represents a quantum gate that rotates or entangles qubits based on input data. These rotations enable the QNN to encode input features, transforming them into high-dimensional Hilbert space representations. During training, gradients of these parameters are optimized to minimize a loss function, analogous to classical neural networks, but constrained by quantum hardware limitations and noise sensitivity [1].

3.3. Data Encoding in Quantum Neural Networks

Efficient data encoding is critical for the practicality of QNNs. In this work, we apply ATP before encoding data into quantum states. Primarily, we use *angle encoding*, where each pixel value $x_{i,j}$ of an image is mapped onto a rotation angle $\theta_{i,j}$ for the RX gate:

$$\theta_{i,j} = \pi \cdot x_{i,j}, \quad (4)$$

resulting in the transformation $RX(\theta_{i,j})$ applied to each qubit [27]. This method straightforwardly represents pixel intensities but can introduce redundancy for higher-dimensional data, as unnecessary values in qubit states increase entanglement and computational complexity.

We also conduct experiments using *amplitude encoding*, which encodes normalized data values as amplitudes of a quantum state. For an input vector \mathbf{x} with dimension 2^n (for n qubits), amplitude encoding transforms it to:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} x_i |i\rangle, \quad (5)$$

allowing all features to be encoded simultaneously in a single quantum state. While amplitude encoding can be efficient for certain data dimensions, it faces scalability challenges for large datasets.

Additionally, alternative encoding strategies exist, such as dense encoding techniques that represent multiple data features using fewer qubits by encoding them into different parameters of multi-qubit gates. These methods aim to balance representational power with resource constraints, but each encoding approach carries unique trade-offs in complexity and scalability. Across these various encoding schemes, ATP helps reduce redundancy by pruning non-essential features before encoding, leading to more efficient and robust QNN performance.

3.4. Entanglement Entropy

EE is a metric from quantum information theory that quantifies the correlation between different parts of a quantum system, typically measured using the von Neumann entropy of a reduced density matrix [3]. In QML, EE reflects a model’s capacity to capture intricate data relationships. Higher EE often indicates stronger interdependence between qubits, which can increase the model’s expressive power but also adds computational complexity and risks overfitting [5]. Conversely, lower EE may represent efficient resource usage and reduced complexity, which is advantageous for simpler datasets [23]. Balancing EE is thus crucial, as it influences both learning potential and robustness in noisy environments. Therefore, achieving high performance with lower EE, or reduced complexity, is an important aim of this study.

4. Methods

To select effective threshold values for data pruning, we formalize threshold selection as a constrained optimization problem tailored to adapt to each dataset’s unique characteristics. This approach employs the Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm with box constraints (L-BFGS-B), a quasi-Newton method that efficiently handles high-dimensional optimization by leveraging gradient-based updates to refine threshold levels. By using this optimization framework, we systematically balance accuracy with computational efficiency, ensuring that thresholds are adjusted to retain essential information while minimizing redundancy. The following sections present the mathematical formulation and algorithmic steps that drive this threshold-tuning process for enhanced QNN performance.

4.1. Pruning Function Definition

To define the threshold τ for filtering, we calculate the average pixel intensity matrices \bar{x}_0 and \bar{x}_1 across all training samples in each binary class. Specifically, $\bar{x}_0(i, j)$ and $\bar{x}_1(i, j)$ represent the average intensity values at position (i, j) for classes 0 and 1, respectively. Using these averages, we prune data by setting values to zero at positions (i, j) where both class averages fall below the threshold τ , as follows:

$$x_\tau(i, j) = \begin{cases} 0, & \text{if } \bar{x}_0(i, j) < \tau \text{ and } \bar{x}_1(i, j) < \tau, \\ x(i, j), & \text{otherwise,} \end{cases} \quad (6)$$

where $x(i, j)$ represents the original pixel intensity at position (i, j) . This operation generates a pruned dataset \mathcal{X}_τ that retains only grid positions with sufficient intensity to contribute effectively to classification.

To determine the optimal threshold τ^* , we maximize the test accuracy $\text{Acc}_{\text{test}}(\mathcal{X}_\tau)$:

$$\tau^* = \arg \max_{\tau \in [0, \tau_{\max}]} \text{Acc}_{\text{test}}(\mathcal{X}_\tau), \quad (7)$$

where τ_{\max} is the upper bound on τ . This optimization is equivalent to minimizing the negative accuracy $-\text{Acc}_{\text{test}}(\mathcal{X}_\tau)$ over the interval $[0, \tau_{\max}]$.

4.2. Gradient-Based Optimization via L-BFGS-B

To solve the threshold optimization problem, we use the L-BFGS-B algorithm [44], which efficiently manages high-dimensional optimization with limited memory. The algorithm iteratively adjusts τ by minimizing the negative accuracy:

$$f(\tau) = -\text{Acc}_{\text{test}}(\mathcal{X}_\tau), \quad (8)$$

updating τ at each step according to:

$$\tau_{k+1} = \tau_k - \alpha_k H_k \nabla f(\tau_k), \quad (9)$$

where α_k is the step size, H_k is an approximation of the inverse Hessian matrix, and $\nabla f(\tau_k)$ is the gradient of $f(\tau_k)$ with respect to τ . The gradient $\nabla f(\tau_k)$ is computed as:

$$\nabla f(\tau_k) = \sum_{(i,j)} \frac{\partial f}{\partial x_\tau(i,j)} \cdot \frac{\partial x_\tau(i,j)}{\partial \tau}. \quad (10)$$

This approach, as detailed in Algorithm 1, leverages gradient information to efficiently converge on the optimal threshold τ^* , ensuring a refined selection of relevant data regions.

Algorithm 1 Bi-Level Threshold Optimization for QNN

Require: Binary class pair (c_0, c_1) , dataset (x, y) , encoding function f_{enc} , grid size s , epochs, threshold range

Ensure: Optimized threshold τ^* and classification accuracy

- 1: Filter and resize data for selected class pair (c_0, c_1) , converting labels to binary
 - 2: Calculate average values for each class and set pixels to zero where averages are below τ
 - 3: Convert processed images to quantum circuits using f_{enc}
 - 4: Initialize QNN model with angle encoding
 - 5: Define bi-level optimization:
 - 6: Inner level: apply threshold, filter data, train QNN on $(x_{\text{train}}, y_{\text{train}})$
 - 7: Outer level: maximize test accuracy on $(x_{\text{test}}, y_{\text{test}})$
 - 8: Optimize over τ within range using L-BFGS-B
 - 9: Calculate entanglement entropy of final model on test set
 - 10: **return** optimal threshold τ^* , test accuracy, and entanglement entropy
-

4.3. Constraint Handling and Convergence Criteria

The box constraints $0 \leq \tau \leq \tau_{\text{max}}$ are enforced within L-BFGS-B by projecting any out-of-bounds updates back to the feasible region, ensuring that the threshold remains within the desired range throughout the optimization. Convergence is achieved when the norm of the projected gradient $\|\nabla f(\tau_k)\|_\infty$ falls below a predefined tolerance, ϵ , or when the maximum number of iterations is reached:

$$\|\nabla f(\tau_k)\|_\infty < \epsilon. \quad (11)$$

Additionally, the accuracy is evaluated on a validation set \mathcal{X}_{val} after each iteration to ensure that the optimization process generalizes effectively to unseen data, mitigating overfitting to the training set.

4.4. Approximation of the Inverse Hessian

To achieve computational efficiency, L-BFGS-B approximates the Hessian matrix rather than computing it directly

[44]. Using a limited history of m gradient and position vectors $\{s_i, y_i\}_{i=k-m}^{k-1}$, where $s_i = \tau_{i+1} - \tau_i$ and $y_i = \nabla f(\tau_{i+1}) - \nabla f(\tau_i)$, the inverse Hessian approximation H_k is updated using the recursive formula:

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T, \quad (12)$$

where $\rho_k = (y_k^T s_k)^{-1}$ and $V_k = I - \rho_k y_k s_k^T$. This recursive update captures the essential curvature information while minimizing memory overhead, making the optimization feasible for high-dimensional datasets.

After identifying the optimal threshold τ^* , we generate the filtered datasets \mathcal{X}_{τ^*} for training and testing, where irrelevant positions have been pruned based on τ^* . The QNN is then trained on \mathcal{X}_{τ^*} , leveraging the optimized threshold to focus on critical data regions while minimizing computational load and enhancing classification performance. Test accuracy $\text{Acc}_{\text{test}}(\mathcal{X}_{\tau^*})$ is subsequently evaluated to confirm the effectiveness of threshold optimization.

5. Experiments

5.1. Setup

To evaluate our approach, we conducted binary classification experiments on multiple benchmark datasets: MNIST, FashionMNIST, CIFAR, and PneumoniaMNIST. These datasets were chosen to assess the model's adaptability across varied image types and complexities. The experiments were implemented using a compact three-layer QNN model, which serves as the base architecture for all encoding techniques tested. The QNN model applies a parameterized quantum circuit built with data qubits and a readout qubit, initialized with an X gate followed by a Hadamard gate to prepare it in superposition. The circuit incorporates both entangling XX and ZZ gates between the data qubits and the readout qubit, with learnable parameters to capture complex dependencies within the input data. A final Hadamard gate is applied to the readout qubit before measurement, completing the encoding and entanglement process for each layer.

5.1.1 Encoding and Preprocessing Techniques

For encoding, Angle and Amplitude methods were applied directly to the data. In contrast, ATP, SQE, and PCA were used beforehand, each applying a different approach to prepare the data before encoding. Specifically, ATP and PCA refine data structure before encoding, while SQE focuses on single-qubit encoding. The results presented here are from experiments where these preparatory methods were followed by Angle encoding. Additional experiments with direct Amplitude encoding, as well as further analyses, are included in the supplementary materials and ablation study section.

5.1.2 Realistic Noise Conditions

To evaluate robustness in realistic settings, depolarizing noise was introduced at intensities of 3%, 5%, and 10%, allowing for a comparison of baseline performance on clean data and the model’s resilience under these quantum noise levels. This setup enabled a thorough assessment of each encoding method’s effectiveness, revealing how the QNN model performs across both ideal and noisy environments.

5.2. Manual Threshold Pruning

In this part, we examine the effect of pruning with different thresholds on QNN performance. Figure 3 illustrates the data distribution for each position within a 3x3 division of the image, where the values represent average intensities across samples from a specific class. As shown, positions with lower information content tend to have closely overlapping values between classes, which can make training more challenging by introducing less distinctive information into the QNN.

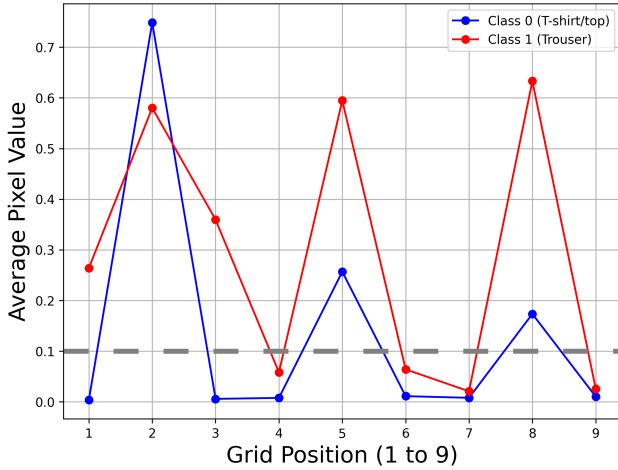


Figure 3. Data distribution across positions in FashionMNIST for two classes (T-shirt/top and Trouser), with dashed lines marking threshold levels. Positions with lower variance are pruned to streamline training and focus on more informative regions.

To further investigate the impact of these thresholds on performance, we apply a range of manual thresholds across different class pairs in both MNIST and FashionMNIST datasets. Figures 4 and 5 show that moderate pruning generally leads to improved accuracy by eliminating non-essential features while preserving critical distinctions. In Figure 4, results for MNIST indicate that moderate thresholds enhance accuracy, while higher thresholds reduce it by removing valuable details. Similarly, for FashionMNIST (Figure 5), the optimal thresholds cluster between 0.1 and 0.3, but specific values vary with data distribution characteristics.

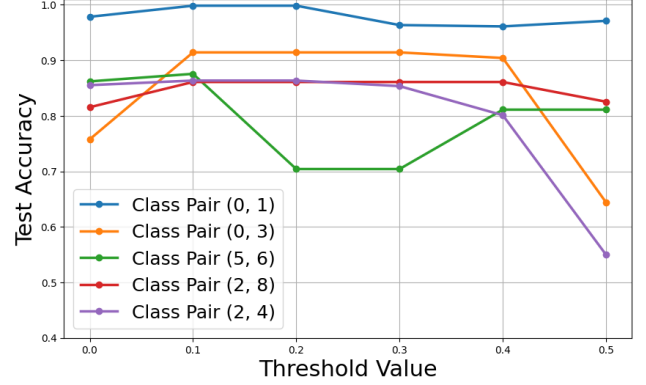


Figure 4. Test accuracy for MNIST class pairs with varying pruning thresholds. Moderate thresholds improve accuracy, while higher thresholds may exclude key information.

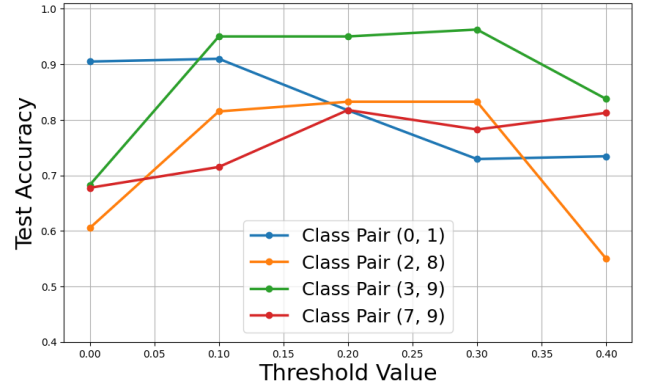


Figure 5. Test accuracy for FashionMNIST class pairs with varying thresholds. Similar to MNIST, a threshold around 0.3 generally provides optimal performance.

These results demonstrate that the optimal pruning level is influenced by the data distribution, motivating the need for an adaptive thresholding approach. The following section details the results from the bi-level optimization method used in our framework to automate threshold selection based on each dataset’s characteristics.

5.3. Performance Results

Tables 1 and 2 present the classification accuracy and EE, respectively, for different encoding techniques across the four datasets used in the QNN binary classification tasks. For MNIST and FashionMNIST, multiple class pairs were chosen to evaluate the model’s capability in distinguishing various subsets, providing a broader assessment of encoding effectiveness. Across the majority of datasets and class pairs, ATP achieved the highest accuracy. ATP also consistently minimized EE compared to other methods, indicating a more efficient use of quantum resources by lowering com-

Table 1. Performance of encoding techniques on various classes for binary classifications (Accuracy).

Classes	Angle	Amplitude	ATP	PCA	SQE
MNIST					
(0,1)	96.0	95.5	99.0	99.0	88.0
(0,3)	89.0	88.5	91.0	88.0	86.0
(2,4)	85.0	84.0	86.0	84.5	82.0
(5,6)	86.0	85.5	87.0	85.0	83.5
(2,8)	81.0	79.5	83.0	86.0	78.5
Fashion MNIST					
(0,1)	88.5	88.0	91.5	88.5	86.0
(2,8)	86.0	84.5	86.0	86.0	83.0
(3,9)	94.0	87.0	94.0	93.0	91.0
(7,9)	82.0	78.0	83.0	79.0	77.0
CIFAR					
(0,1)	70.0	68.5	74.2	68.0	66.0
PneumoniaMNIST					
(0,1)	81.0	68.5	87.0	80.0	75.5

Table 2. Entanglement entropy of encoding techniques on various classes for binary classifications.

Classes	Angle	Amplitude	ATP	PCA	SQE
MNIST					
(0,1)	0.67	0.52	0.39	0.60	0.45
(0,3)	0.59	0.53	0.34	0.55	0.44
(2,4)	0.77	0.64	0.45	0.56	0.41
(5,6)	0.82	0.53	0.41	0.61	0.39
(2,8)	0.63	0.56	0.34	0.36	0.34
Fashion MNIST					
(0,1)	0.56	0.47	0.35	0.58	0.39
(2,8)	0.52	0.50	0.35	0.41	0.38
(3,9)	0.54	0.62	0.38	0.55	0.42
(7,9)	0.59	0.58	0.41	0.64	0.43
CIFAR					
(0,1)	0.63	0.51	0.46	0.65	0.43
PneumoniaMNIST					
(0,1)	0.88	0.79	0.37	0.59	0.42

plexity without compromising accuracy.

To assess the encoding methods’ resilience to noise, we examined model performance under various levels of depolarizing noise (3-10%). Table 3 highlights that encoding techniques with lower entanglement entropy, such as ATP and SQE, showed stronger robustness with accuracy reductions typically limited to 3–8 points. In contrast, Angle, Amplitude, and PCA experienced more substantial drops in accuracy of 4–17 points. These results emphasize ATP and SQE’s effectiveness in preserving model accuracy under challenging noise conditions, establishing a baseline for QNN stability in practical, noisy environments.

Table 3. Performance of encoding techniques on various classes for binary classifications under depolarizing noise (Accuracy).

Classes	Angle	Amplitude	ATP	PCA	SQE
MNIST					
(0,1)	87.0	86.4	89.0	89.0	88.2
(0,3)	82.4	80.5	83.0	78.9	81.9
(2,4)	80.2	68.8	71.0	79.7	79.8
(5,6)	70.8	69.4	73.0	68.5	74.0
(2,8)	68.8	57.4	76.5	74.9	75.2
Fashion MNIST					
(0,1)	80.5	79.0	84.9	84.9	82.1
(2,8)	76.1	75.4	79.3	78.6	79.5
(3,9)	84.3	78.2	87.9	86.5	86.7
(7,9)	70.0	66.2	68.4	73.2	74.5
CIFAR					
(0,1)	60.4	59.0	61.0	57.8	58.5
PneumoniaMNIST					
(0,1)	70.5	63.4	76.0	67.3	68.7

5.4. Adversarial Robustness

To evaluate the resilience of each encoding method against adversarial attacks, we applied Fast Gradient Sign Method (FGSM) with an attack strength of $\epsilon = 0.3$ across the datasets. Table 4 provides accuracy results for binary classification tasks under these adversarial conditions. In most cases, direct encoding methods like Angle and Amplitude encoding were generally more susceptible to the attacks, displaying substantial accuracy reductions. ATP, PCA, and SQE exhibited a moderate level of robustness, although performance varied across datasets. Notably, ATP and PCA maintained comparable accuracy levels in several cases, but in certain tasks (PneumoniaMNIST and CIFAR), Angle encoding unexpectedly achieved the highest accuracy, suggesting that additional adversarial training methods may be required to further improve model resilience.

The encoding techniques used in this study, including ATP, lack inherent robustness against adversarial attacks, as shown in prior evaluations. To address this, adversarial training was applied as an additional defense. Figure 6 presents post-training accuracy for each encoding method under FGSM attacks, highlighting ATP and PCA’s enhanced performance due to their filtering of non-informative features, which decreases sensitivity to irrelevant perturbations. Additional results for other attack models are provided in the supplementary material.

As shown in Figure 6, adversarial training enhances resilience across most encoding methods, particularly at lower ϵ values. ATP consistently achieves higher accuracy across attack strengths, maintaining a moderate lead over PCA and SQE despite performance declines as ϵ increases. Amplitude and Angle encoding remain more sus-

Table 4. Performance of encoding techniques on various classes for binary classifications under FGSM attack with $\epsilon = 0.3$ (Accuracy).

Classes	Angle	Amplitude	ATP	PCA	SQE
MNIST					
(0,1)	62.0	58.4	67.2	66.5	65.0
(0,3)	63.2	61.0	67.3	69.0	63.7
(2,4)	55.2	54.7	60.0	61.5	59.0
(5,6)	64.3	61.5	66.7	65.0	62.5
(2,8)	48.6	50.5	56.0	57.8	54.0
Fashion MNIST					
(0,1)	68.4	66.0	72.5	73.0	70.2
(2,8)	66.2	65.5	70.3	71.0	69.0
(3,9)	77.5	74.2	76.0	75.8	73.5
(7,9)	60.0	59.2	64.8	63.5	62.3
CIFAR					
(0,1)	55.0	51.4	59.5	61.2	58.0
PneumoniaMNIST					
(0,1)	66.5	62.0	64.0	65.2	63.4

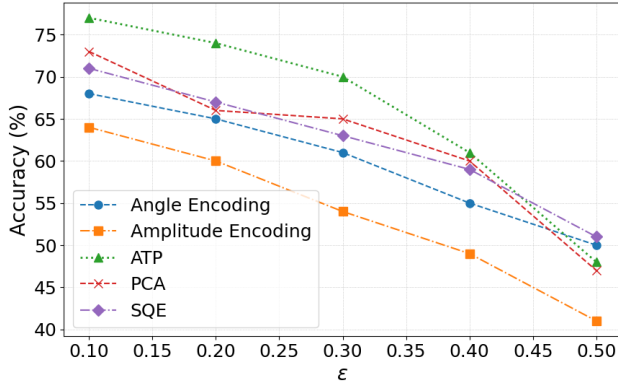


Figure 6. Effectiveness of different encoding methods against adversarial attacks after adversarial training, evaluated at varying attack strengths (measured by ϵ) for (0,1) classification tasks on MNIST. ATP shows higher robustness across most ϵ values, though all models experience performance declines as ϵ increases.

ceptible to attacks, especially at higher ϵ values, indicating lower robustness. These results underscore ATP’s advantage in accuracy preservation, though further measures may be needed for robustness under stronger adversarial conditions.

5.5. Experiments on IBM Quantum Hardware

To evaluate ATP’s robustness on real quantum hardware, we extended our experiments to the IBM Sherbrooke backend using Qiskit’s Runtime Service. We implemented the proposed framework with Qiskit’s native `EstimatorQNN` interface, using default settings for error mitigation and backend resilience. The model was trained with a COBYLA op-

timizer for 30 iterations. Across five MNIST class subsets, ATP demonstrated an average performance improvement of 7% compared to the direct encoding approach, with notable accuracy gains on the (2,4) and (2,8) classification tasks, the latter showing the highest improvement.

6. Discussion and Limitations

While ATP achieved notable performance gains across datasets, some limitations remain. First, we focus on binary classification for two main reasons: comparability with existing literature and the efficiency of quantum measurement processes. Comparability is important, as most prior works predominantly use binary classification to evaluate state-of-the-art QNN implementations. Moreover, binary classification simplifies quantum measurement, as the class label can be inferred from a single-qubit von Neumann measurement—such as the sign of the expectation value of the σ_z observable. This approach, including measuring only the last qubit to determine the output label with minimal circuit complexity, has been well-documented in prior studies [17, 42]. In contrast, most multi-class classification methods require a hybrid between classical and quantum models [8], making binary classification the preferred choice both in the literature and in our study. Additionally, ATP’s optimization process incurs moderate computational overhead, with threshold selection and pruning taking approximately 15% longer than direct angle encoding (compared to 12% for PCA and 4% for SQE). While this may affect efficiency in larger applications, further optimizations in the thresholding algorithm could help reduce these demands.

Future work could also explore ATP’s adaptability to datasets with varied underlying distributions, as current evaluations focus on standard QNN benchmarks. Investigating performance on more diverse datasets may offer insights for improving ATP’s robustness in broader real-world scenarios.

7. Conclusion

We presented Adaptive Threshold Pruning (ATP) to address the resource constraints of quantum systems by removing non-essential input features before encoding, enabling more efficient use of qubits. Rather than reducing circuit complexity directly, ATP lowers input redundancy, which leads to reduced entanglement and improved computational efficiency. Across multiple datasets, ATP consistently outperforms other encoding methods by achieving higher accuracy with lower entanglement entropy. By adapting pruning thresholds to the variance structure of the data, ATP offers a flexible and scalable encoding strategy that enhances QNN performance in resource-limited settings.

References

- [1] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.
- [2] Mahabubul Alam and Swaroop Ghosh. Qnet: A scalable and noise-resilient quantum neural network architecture for noisy intermediate-scale quantum computers. *Frontiers in physics*, 9:755139, 2022.
- [3] Vassilis Anagiannis and Miranda CN Cheng. Entangled q-convolutional neural nets. *Machine Learning: Science and Technology*, 2(4):045026, 2021.
- [4] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. Experimental characterization, modeling, and analysis of crosstalk in a quantum computer. *IEEE Transactions on Quantum Engineering*, 1:1–6, 2020.
- [5] Sheng-Chen Bai, Yi-Cheng Tang, and Shi-Ju Ran. Unsupervised recognition of informative features via tensor network machine learning and quantum entanglement variations. *Chinese Physics Letters*, 39:100701, 2022.
- [6] Bhattaraprot Bhabhatsatam and Sucha Smanchat. Hybrid quantum encoding: Combining amplitude and basis encoding for enhanced data storage and processing in quantum computing. In *2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 512–516. IEEE, 2023.
- [7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [8] Denis Bokhan, Alena S Mastiukova, Aleksey S Boev, Dmitrii N Trubnikov, and Aleksey K Fedorov. Multiclass classification using quantum convolutional neural networks with hybrid quantum-classical learning. *Frontiers in Physics*, 10:1069985, 2022.
- [9] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022.
- [10] Philip Easom-McCaldin, Ahmed Bouridane, Ammar Belatreche, Richard Jiang, and Somaya Al-Maadeed. Efficient quantum image classification using single qubit encoding. *IEEE Transactions on Neural Networks and Learning Systems*, 35(2):1472–1486, 2022.
- [11] Ian Goodfellow. Deep learning, 2016.
- [12] Mayank Gupta and Manisha J Nene. Quantum computing: An entanglement measurement. In *2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI)*, pages 1–6. IEEE, 2020.
- [13] Stuart Hadfield, Zhihui Wang, Bryan O’gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.
- [14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.
- [15] Mika Hirvensalo. *Quantum computing*. Springer Science & Business Media, 2013.
- [16] Zhirui Hu, Peiyan Dong, Zhepeng Wang, Youzuo Lin, Yanzhi Wang, and Weiwen Jiang. Quantum neural network compression. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pages 1–9, 2022.
- [17] Tak Hur, Leeseok Kim, and Daniel K Park. Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, 4(1):3, 2022.
- [18] Ian T Jolliffe and BJT Morgan. Principal component analysis and exploratory factor analysis. *Statistical methods in medical research*, 1(1):69–95, 1992.
- [19] Emanuel Knill. Quantum computing with realistically noisy devices. *Nature*, 434(7029):39–44, 2005.
- [20] Ankit Kulshrestha, Xiaoyuan Liu, Hayato Ushijima-Mwesigwa, Bao Bach, and Ilya Safro. Qadaprune: Adaptive parameter pruning for training variational quantum circuits. *arXiv preprint arXiv:2408.13352*, 2024.
- [21] Guangxi Li, Ruilin Ye, Xuanqiang Zhao, and Xin Wang. Concentration of data encoding in parameterized quantum circuits. *Advances in Neural Information Processing Systems*, 35:19456–19469, 2022.
- [22] Marina Lisnichenko and Stanislav Protasov. Quantum image representation: A review. *Quantum Machine Intelligence*, 5(1):2, 2023.
- [23] Yuhan Liu, Wen-Jun Li, Xiao Zhang, Maciej Lewenstein, Gang Su, and Shi-Ju Ran. Entanglement-based feature extraction by tensor network machine learning. *Frontiers in Applied Mathematics and Statistics*, 7:716044, 2021.
- [24] John Martyn, Guifre Vidal, Chase Roberts, and Stefan Leichenauer. Entanglement and tensor networks for supervised image classification. *arXiv preprint arXiv:2007.06082*, 2020.
- [25] Tuyen Nguyen, Incheon Paik, Yutaka Watanobe, and Truong Cong Thang. An evaluation of hardware-efficient quantum neural networks for image data classification. *Electronics*, 11(3):437, 2022.
- [26] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2(4):040316, 2021.
- [27] Emmanuel Ovalle-Magallanes, Dora E Alvarado-Carrillo, Juan Gabriel Avina-Cervantes, Ivan Cruz-Aceves, and Jose Ruiz-Pinales. Quantum angle encoding with learnable rotation applied to quantum-classical convolutional neural networks. *Applied Soft Computing*, 141:110307, 2023.
- [28] John Preskill. Quantum computing and the entanglement frontier. *arXiv preprint arXiv:1203.5813*, 2012.
- [29] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [30] Deepak Ranga, Aryan Rana, Sunil Prajapat, Pankaj Kumar, Kranti Kumar, and Athanasios V Vasilakos. Quantum machine learning: Exploring the role of data encoding techniques, challenges, and future directions. *Mathematics*, 12(21):3318, 2024.
- [31] Minati Rath et al. Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques

- and their impact on machine learning accuracy. *EPJ Quantum Technology*, 11(1):1–22, 2024.
- [32] Manas Sajjan, Junxu Li, Raja Selvarajan, Shree Hari Sureshbabu, Sumit Suresh Kale, Rishabh Gupta, Vinit Singh, and Sabre Kais. Quantum machine learning for chemistry and physics. *Chemical Society Reviews*, 51(15):6475–6573, 2022.
 - [33] Maria Schuld and Francesco Petruccione. *Supervised learning with quantum computers*. Springer, 2018.
 - [34] Yu Shee, Pei-Kai Tsai, Cheng-Lin Hong, Hao-Chung Cheng, and Hsi-Sheng Goan. Qubit-efficient encoding scheme for quantum simulations of electronic structure. *Physical Review Research*, 4(2):023154, 2022.
 - [35] Peter W Shor. Quantum computing. *Documenta Mathematica*, 1(1000):467–486, 1998.
 - [36] Sukin Sim, Jonathan Romero, Jérôme F Gonthier, and Alexander A Kunitsa. Adaptive pruning-based optimization of parameterized quantum circuits. *Quantum Science and Technology*, 6(2):025019, 2021.
 - [37] Jinkai Tian, Xiaoyu Sun, Yuxuan Du, Shanshan Zhao, Qing Liu, Kaining Zhang, Wei Yi, Wanrong Huang, Chaoyue Wang, Xingyao Wu, et al. Recent advances for quantum neural networks in generative learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12321–12340, 2023.
 - [38] Aijuan Wang, Jianglong Hu, Shiyue Zhang, and Lusi Li. Shallow hybrid quantum-classical convolutional neural network model for image classification. *Quantum Information Processing*, 23(1):17, 2024.
 - [39] Xinbiao Wang, Junyu Liu, Tongliang Liu, Yong Luo, Yuxuan Du, and Dacheng Tao. Symmetric pruning in quantum neural networks. *arXiv preprint arXiv:2208.14057*, 2022.
 - [40] Zeyu Wang, Xianhang Li, Hongru Zhu, and Cihang Xie. Revisiting adversarial training at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24675–24685, 2024.
 - [41] Manuela Weigold, Johanna Barzen, Frank Leymann, and Marie Salm. Data encoding patterns for quantum computing. In *Proceedings of the 27th Conference on Pattern Languages of Programs*, pages 1–11, 2020.
 - [42] Wenjie Wu, Ge Yan, Xudong Lu, Kaisen Pan, and Junchi Yan. Quantumdarts: differentiable quantum architecture search for variational quantum algorithms. In *International Conference on Machine Learning*, pages 37745–37764. PMLR, 2023.
 - [43] Renxin Zhao and Shi Wang. A review of quantum neural networks: methods, models, dilemma. *arXiv preprint arXiv:2109.01840*, 2021.
 - [44] Ciyao Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.