**Optimizing Quantum CNN Architectures for Efficient Image Classification**

**FYP– I REPORT**

**BS(CS/SE/CYS/AI) Fall 2024**

Name:

21K-3153

Name:

21K-3293

Name:

21K-4564



**Supervisor:**

**Ms Sumaiyah Zahid**

**Co-supervisor:**

**Dr Muhammad Rafi**

**Department of Computer Science**

**FAST-National University of Computer & Emerging Sciences, Karachi**

Contents

Introduction..................................................................................................................... 4 1. Related Work............................................................................................................... 5 2. Methodology................................................................................................................8 3. Testing and Results.................................................................................................. 12 4. System Diagram....................................................................................................... 15 5. Goals For FYP-II....................................................................................................... 15 6. Conclusion................................................................................................................ 15 References.................................................................................................................... 16

1. Introduction

Quantum Convolutional Neural Networks (QCNNs) have emerged as a promising intersection of quantum computing and artificial intelligence, offering the potential to enhance classical machine learning tasks. These hybrid models integrate quantum circuits into neural architectures to capture intricate patterns in data, potentially outperforming classical methods in areas such as image recognition. The concept of Quanvolutional Neural Networks [5], which leverage quantum feature extraction for improved classification accuracy, marked a significant step forward in quantum-assisted learning systems.

One of the foundational challenges in Quantum Machine Learning (QML) lies in encoding classical data into quantum systems. Several encoding techniques, including Basis Embedding, Angle Embedding, and Amplitude Embedding, have been developed to address this challenge. Among these, refinements such as the Flexible Representation of Quantum Images (FRQI) preserve spatial information crucial for image-based tasks [10]. More recently, Hamiltonian embedding has gained attention for its ability to map classical data to quantum states through system Hamiltonians, enabling enriched feature extraction.

1. Related Work / (SRS/SDS)

Related Work:

The integration of Hamiltonian embedding with variational circuits, as proposed in [22], has shown considerable promise in classification tasks. For example, [18] introduced a model based on the data reuploading circuit with quantum Hamiltonian data embedding, demonstrating significant improvements in classification accuracy compared to baseline QCNN models. This underscores Hamiltonian embedding's potential to redefine state-of-the-art quantum-assisted machine learning.

Despite these advancements, Quantum Neural Networks (QNNs) face the significant challenge of barren plateaus (BP), where gradients diminish exponentially with increasing problem size, hindering parameter optimization [3]. This issue is particularly severe for randomly initialized deep circuits [12]. BPs have also been linked to entanglement and noise in quantum systems [13, 19]. Strategies to mitigate BPs, such as targeted parameter initialization techniques, have shown promise in alleviating these challenges [4, 20, 23]. Among these, initialization strategies have emerged as a crucial factor in addressing barren plateaus and enhancing QNN trainability.

Recent studies, such as [6], have highlighted the critical role of initialization in QNNs, with beta initialization showing improved performance under specific conditions. Earlier conjectures [9] suggested its potential advantages; however, subsequent findings indicate that the observed improvements may stem more from the range of initialization values rather than the distribution itself. Despite these findings, the role of initialization strategies in hybrid quantum-classical architectures, particularly those utilizing Hamiltonian embedding, remains underexplored.

1. Methodology

The Hamiltonian embedding method, as detailed in [18], offers a robust approach to encoding classical image data into quantum circuits while preserving spatial structures. This methodology transforms a grayscale image matrix M into a Hermitian matrix HM , computed as:

[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=H_M%20%3D%20%5Cfrac%7BM%20%2B%20M%5ET%7D%7B2%7D.#0)

The embedding is achieved through a data encoding unitary transformation:

[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=W(t%3B%20M)%20%3D%20e%5E%7B-i%20H_M%20t%20%2F%202%7D#0)

where t serves as a trainable parameter. This embedding process applies a matrix polynomial function on the image, ensuring richer feature extraction capabilities compared to conventional embedding techniques.

**2.1. Data Reuploading Circuit:**

Following Hamiltonian embedding, a data reuploading variational quantum circuit [15] is employed, enhancing the expressive power of Quantum Neural Networks (QNNs) by reintroducing input data at each layer:

[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%7C%5CPsi(x%3B%20%5Cvec%7B%5Comega%7D)%5Crangle%20%3D%20%5Cprod_%7Bi%3D1%7D%5EL%20%5Cleft%5B%20V(%5Comega_i)%20U_%7B%5Cphi%7D(x)%20%5Cright%5D%20%7C0%5Crangle%5E%7B%5Cotimes%20n%7D.#0)

In this definition, the data encoding unitary [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=U_%7B%5Cphi%7D(x)#0) is repeated [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=L#0) times along with the parameterized layer [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=V#0). While the data encoding unitary remains the same across repetitions, the parameters for [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=V#0) vary for each layer, represented by [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Cvec%7B%5Comega%7D%20%3D%20%5C%7B%5Comega_1%2C%20%5Comega_2%2C%20%5Cdots%2C%20%5Comega_L%5C%7D#0).

A diagram of mathematical equations

Description automatically generated

This iterative structure enables the quantum circuit to learn intricate patterns by repeatedly exposing it to the same data encoded via Hamiltonian embedding, leveraging the nonlinear transformations inherent in quantum systems.

**2.2. Proposed Model:**

The final quantum classifier model integrates Hamiltonian embedding with data reuploading circuits. The probability of a given class *i* is calculated by measuring the projection operator **[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5C(%20P_i%20%3D%20%7Ci%5Crangle%20%5Clangle%20i%7C%20%5C)#0):**

**[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%7C%5Cvarphi(t%2C%20%5Cvec%7B%5Comega%7D%3B%20M)%5Crangle%20%3D%20%5Cprod_%7Bi%3D1%7D%5EL%20%5Cleft%5B%20V(%5Comega_i)%20W(t_i%3B%20M)%20%5Cright%5D%20%7C%2B%5Crangle%5E%7B%5Cotimes%20n%7D.%20#0)**

where *I* is the identity operator on the remaining qubits. The model is trained using the cross-entropy loss function, ensuring effective classification:

[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Ctext%7BCross-Entropy%20Loss%7D(M)%20%3D%20-%5Csum_%7Bi%7D%20y_i%20%5Clog_2%20p(M%3B%20i)#0)

with  *[A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5C%20y_i%20%5C#0)* as the one-hot-encoded true label.

The circuit is initialized in an equal superposition of all basis states, represented as [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%7C%2B%5Crangle%5E%7B%5Cotimes%20n%7D#0).

By combining the Hamiltonian embedding’s ability to preserve image structures with the flexibility of data reuploading circuits, this model aims to outperform traditional QCNNs. The effectiveness of this approach is supported by prior results on benchmark datasets, demonstrating improved performance across various classification tasks [18].

**2.3. Initialization Strategies:**

Beta initialization has been demonstrated to significantly enhance the trainability of Quantum Neural Networks (QNNs) by addressing barren plateaus, a critical challenge in quantum machine learning [3]. However, the potential for refining initialization methods and their specific impact on optimization dynamics remain open areas of research. To further explore this, we evaluate and compare various initialization strategies to assess their effect on convergence rates and model performance.

1. **Beta Initialization with Positive Skew**: This method employs a beta distribution, as proposed in [9], with a right-skew ( [A black background with a black square

   Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)< β) to initialize parameters. The goal is to mitigate barren plateaus by concentrating most parameter values near zero, while maintaining sufficient diversity across the initialization range to ensure effective gradient flow and model expressiveness.
2. **Uniform Random Initialization**: Parameters are initialized uniformly at random within the predefined range [0,1]. This widely used method serves as a baseline for comparison with more advanced initialization strategies.
3. **Ensemble Weighted Average Initialization**: This method combines the outputs of models initialized with different strategies by applying an ensemble weighted average. We take the models initialized using Beta Initialization (with positive skew) and Uniform Random Initialization, and average their predictions, weighting each model based on its initial performance. The goal is to determine whether combining the results from both initialization methods yields better performance and convergence compared to using a single initialization strategy. A block diagram outlining this approach can be seen in Figure [1](file:///C:\Users\bilal\Downloads\Evaluating%20Initialization%20Strategies%20in%20Hamiltonian-Embedded%20Quantum%20Neural%20Networks(1).docx#fdleaarietca).

These initialization strategies are applied to investigate their influence on optimization dynamics, mitigation of barren plateaus, and classification accuracy across diverse datasets.

**2.4. Advantages of Beta Initialization:**

The beta initialization method offers distinct advantages for training Quantum Neural Networks (QNNs), as detailed in [6]:

* **Mitigation of Barren Plateaus**: Beta initialization effectively avoids regions with vanishing gradients, thereby addressing barren plateaus that severely hinder optimization in QNNs.
* **Improved optimization trajectories**: By initializing parameters in regions with higher gradient magnitudes, beta initialization enables faster and more efficient convergence during training.
* **Consistent Empirical Performance**: Across various datasets and experimental settings, beta initialization achieves superior accuracy and F1 scores, demonstrating robustness in both noiseless and noisy environments.
* **Range Effectiveness**: The observed performance improvements are primarily attributed to the advantageous value ranges of the beta distribution, emphasizing the importance of initialization ranges over specific distribution shapes.

While these benefits highlight the effectiveness of beta initialization, it remains uncertain whether the observed improvements stem solely from advantageous initial parameter positions. Further exploration into the underlying mechanisms of these benefits could offer valuable insights for developing more refined initialization strategies.

In summary, beta initialization improves QNN trainability and performance by addressing key challenges such as barren plateaus and gradient scaling issues. However, its theoretical workings warrant further exploration, as highlighted by [6].

**2.5. Datasets and Evaluation**

We evaluate our methodology on four datasets, each split into training, validation, and testing sets as follows:

* **Kaggle CT Scans**: This is a subset of images from [1], sourced from the Kaggle website [16]. It contains 100 CT images with binary labels indicating the presence or absence of contrast. The original images, which had a resolution of 512x512 pixels, were resized to 32x32 pixels using OpenCV [7] to reduce computational costs. The resized images were then split into a training set of **75 images**, a validation set of **5 images**, and a testing set of **20 images**.
* **SKlearn Digits**: This dataset was sourced from the machine learning package Scikit-learn [14]. The images have a resolution of 8x8 pixels. For this study, only samples with labels ranging from 0 to 7 (a total of eight classes) were used. The data was split into a training set of **1040 images**, a validation set of **130 images** and a testing set of **130 images** [2].
* **MNIST**: The dataset was obtained using the data loading module in Torchvision [17]. For this experiment, only images with labels ranging from 0 to 7 (a total of eight classes) were selected. This resulted in training datasets containing **45,790 images**, validation sets containing **2,410 images** and testing datasets with **8,017 images** [11]. The original MNIST images, with dimensions of 28x28 pixels, were padded with zeros to a size of 32x32 pixels.
* **FashionMNIST**: This dataset was obtained using the data loading module in Torchvision [17]. For this work, only samples with labels ranging from 0 to 7 (a total of eight classes) were selected. This resulted in training datasets containing **45,600 images**, validation datasets containing **2,400 images** and testing datasets with **8,000 images** [21]. The original dimensions of the FashionMNIST images were 28x28 pixels, and they were padded with zeros to reach a size of 32x32 pixels.

For each dataset, we measure classification accuracy and training loss over multiple trials to assess the impact of different initialization methods. The goal is to improve the performance of the Hamiltonian embedding model by experimenting with various initialization techniques, such as beta initialization with positive skew and an ensemble weighted average approach.

These enhancements aim to optimize the model’s convergence and accuracy, building upon the existing Hamiltonian embedding framework.

1. Testing and Results

**4.1. Implementation Details**

The majority of the code used in this study was adapted from the work presented in [18], with modifications made to optimize initialization strategies and evaluation procedures for Quantum Neural Networks (QNNs). We built upon the Hamiltonian embedding technique and data reuploading circuit from [18], integrating our changes to implement Beta and Random initialization strategies, as well as an ensemble approach. These modifications were specifically designed to assess the impact of different initialization methods across various datasets. Furthermore, we utilized the same datasets as in [18], including Kaggle CT Medical Images, SKlearn Digits, MNIST, and FashionMNIST, to evaluate the effectiveness of these initialization strategies.

**4.2 Random and Beta Initialization Training**

The Hamiltonian Embedding model was trained as follows:

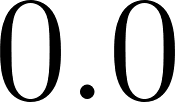
* **Datasets:**

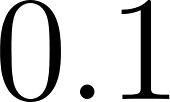
1. Kaggle CT Medical Images.
2. A subset of the SKlearn Digits dataset.
3. A subset of the MNIST dataset.
4. A subset of the FashionMNIST dataset.

* **Training details for the Kaggle CT Medical Images:**
  1. 20 random parameter initializations and 20 beta parameter initializations.
  2. Each trained for 500 iterations using the Adam optimizer [8].
  3. Loss and accuracy were averaged over the 20 initializations.
* **Training details for Sklearn Digits:**
  1. 7 random parameter initializations and 7 beta parameter initializations.
  2. Each trained for 500 iterations using the Adam optimizer [8].
  3. Loss and accuracy were averaged over the 7 initializations.
* **Training details for MNIST and FashionMNIST:**
  1. 5 random parameter initializations and 5 beta parameter initializations.
  2. Each trained for 500 iterations using the Adam optimizer [8].
  3. Loss and accuracy were averaged over the 5 initializations.
* **Results:** The averaged evaluation metrics (loss and accuracy) for the final iteration of each dataset are summarized in Table [1](file:///C:\Users\bilal\Downloads\Evaluating%20Initialization%20Strategies%20in%20Hamiltonian-Embedded%20Quantum%20Neural%20Networks(1).docx#yoezfu8elkex). The curve plot of the training and validation accuracies and losses can be found in Figure [2](file:///C:\Users\bilal\Downloads\Evaluating%20Initialization%20Strategies%20in%20Hamiltonian-Embedded%20Quantum%20Neural%20Networks(1).docx#126tnd1r7s63).

**4.3. Weighted Ensemble Approach**

For the weighted ensemble approach, the outputs from models trained with Random and Beta initializations were combined using a weighted average, where the weights were denoted as [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) (for Beta initialization) and [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=1-%5Calpha#0) (for Random initialization). The value of [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) was varied from [](https://www.codecogs.com/eqnedit.php?latex=0.0#0) to [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=1.0#0) in increments of [](https://www.codecogs.com/eqnedit.php?latex=0.1#0) to observe its effect on model performance. This approach was evaluated on the test split of all the datasets, which was kept separate from the training and validation sets. The test sets represent a portion of the data that the model has never encountered during training, providing an unbiased evaluation of the model’s generalization ability. By using the test splits of the datasets, we ensured that the performance of the weighted ensemble was assessed on data not influenced by the initializations during training or validation. This step is crucial for determining the true effectiveness of the ensemble approach, especially when comparing it to models trained with individual initialization methods.

For each dataset, the ensemble's accuracy and cost were computed for different [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) values, and the results were averaged over multiple runs. The weighted ensemble aimed to assess whether combining these initialization methods could yield better performance compared to individual models. The averaged evaluation metrics (loss and accuracy), for the weighted ensemble approach are presented in Table [2](file:///C:\Users\bilal\Downloads\Evaluating%20Initialization%20Strategies%20in%20Hamiltonian-Embedded%20Quantum%20Neural%20Networks(1).docx#lhtvimdojmu2). Additionally, Figure [3](file:///C:\Users\bilal\Downloads\Evaluating%20Initialization%20Strategies%20in%20Hamiltonian-Embedded%20Quantum%20Neural%20Networks(1).docx#k1eyoka4l3zs) provides a detailed visualization of the accuracy and loss trends for each [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) value across all datasets.

Results

**4.1. Performance of Beta vs Random Initialization**

Table [1](#yoezfu8elkex) and Figure [2](#126tnd1r7s63) provide the training and validation performance metrics for the four datasets, comparing Random and Beta initialization strategies.

Additionally, Table [2](#lhtvimdojmu2) and Figure [3](#k1eyoka4l3zs) showcase the ensemble approach, where [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0) = 0.0 corresponds to full Random initialization and [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 1.0 represents full Beta initialization, highlighting the performance of both strategies across the test splits of all datasets.  
  
The results reveal that Beta initialization consistently outperformed or matched Random initialization across training, validation, and testing phases. The performance gap between the two initialization methods is clearly visible in certain datasets. Specifically, the gap in training and validation accuracies is prominent in MNIST and FashionMNIST (Figure [2](#126tnd1r7s63)). Figure 3demonstrates that Beta initialization outperforms Random initialization on three out of the four datasets, while matching Random initialization on the remaining dataset.

The dataset-specific findings are as follows:   
  
**• Kaggle CT Medical Images:** Beta initialization achieved better convergence during training, with higher training accuracy than Random initialization. However, its validation accuracy was slightly lower, likely due to the limited size of the validation set (only 5 images). In terms of testing, Beta initialization exhibited better generalization, achieving higher test accuracy and lower test loss. This underscores the suitability of Beta initialization for complex medical imaging tasks despite challenges posed by small datasets.   
**• SKlearn Digits:** Beta initialization demonstrated more efficient learning during training, reflected in higher training accuracy and slightly lower training loss compared to Random initialization. Both initialization methods achieved comparable validation and test accuracies, indicating strong generalization for both. Beta’s ability to achieve marginally better optimization highlights its advantage in scenarios where Random initialization already performs well.   
**• MNIST:** On this larger and more structured dataset, Beta initialization consistently outperformed Random initialization in training, validation, and testing metrics. The improved convergence during training and higher validation and test accuracies suggest that Beta initialization enhances the model’s ability to generalize effectively in datasets with well-defined patterns.   
**• FashionMNIST:** Beta initialization delivered the most noticeable improvements on this dataset, achieving higher training, validation, and test accuracies compared to Random initialization. These results highlight Beta’s adaptability to datasets with overlapping class distributions and greater intra-class variability. The reduced losses across all phases further reinforce its effectiveness in handling challenging datasets.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Initialization** | **Train loss** | **Val loss** | **Test Loss** | **Train Acc** | **Val Acc** | **Test Acc** |
| **Kaggle CT Medical Images** | Random Beta | 0.5535 0.5450 | 0.6533 0.6788 | 0.6131 0.6047 | 82.60% 85.33% | 80.00% 60.00% | 65%  75% |
| **SKlearn Digits** | Random Beta | 1.5469 1.5202 | 1.5674 1.5512 | 1.5560 1.5366 | 95.38% 96.73% | 96.15% 96.15% | 95.60%  95.60% |
| **MNIST** | Random Beta | 1.5694 1.5534 | 1.5671 1.5544 | 1.5893 1.5492 | 89.32% 90.39% | 88.24% 89.9% | 89.04%  90.78% |
| **FashionMNIST** | Random Beta | 1.6188 1.5920 | 1.6144 1.5939 | 1.6212 1.5949 | 77.75% 79.64% | 78.07%  80.20% | 77.32%  79.46% |

Table 1. Compression of Results Across Datasets with Different Initializations

**4.2. Results and Analysis of Ensemble Approach**

Table [2](#lhtvimdojmu2) and Figure [3](#k1eyoka4l3zs) presents the performance metrics of the ensemble approach, where varying values of alpha represent different weightages between Beta and Random initialization. As alpha increases, more weight is given to Beta initialization, with [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 1.0 representing full reliance on Beta initialization, and [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 0.0 indicating reliance solely on Random initialization.  
  
The results demonstrate that increasing the weightage of Beta initialization generally leads to improved performance across three out of the four datasets. As [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0) increases accuracy improves while loss decreases, with the most notable improvements observed in CT Medical Images, MNIST, and FashionMNIST. In these datasets, the performance metrics consistently improve, as the accuracy rises and the loss decreases with higher [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0), indicating that Beta initialization helps refine model convergence and generalization.  
  
However, in the SKLearn Digits dataset, a peak is observed at [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 0.4 in terms of accuracy and loss, with further increases in [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)resulting in a slight drop in performance. This suggests that while Beta initialization provides benefits in most cases, there may be diminishing returns or an optimal balance for certain datasets like SKLearn Digits. The plateau and subsequent slight decline could be attributed to the nature of the dataset itself, which might benefit from a balance between Random and Beta initialization rather than relying too heavily on Beta. This suggests that the impact of Beta initialization might vary depending on the dataset’s characteristics, particularly its inherent complexity and the level of overlap in class distributions.  
  
In summary, increasing the weightage of Beta initialization leads to improvements in both accuracy and loss in three out of the four datasets, with particularly noticeable gains in CT Medical Images, MNIST, and FashionMNIST. The improvements in loss across these datasets suggest that Beta initialization contributes to better convergence and generalization, particularly in more complex datasets. In contrast, SKLearn Digits shows smaller gains, but a peak being observed at [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0) = 0.4 suggests that an ensemble approach can be useful when applied to certain datasets.

**4.3. Impact of Beta Initialization and Ensemble Approach**

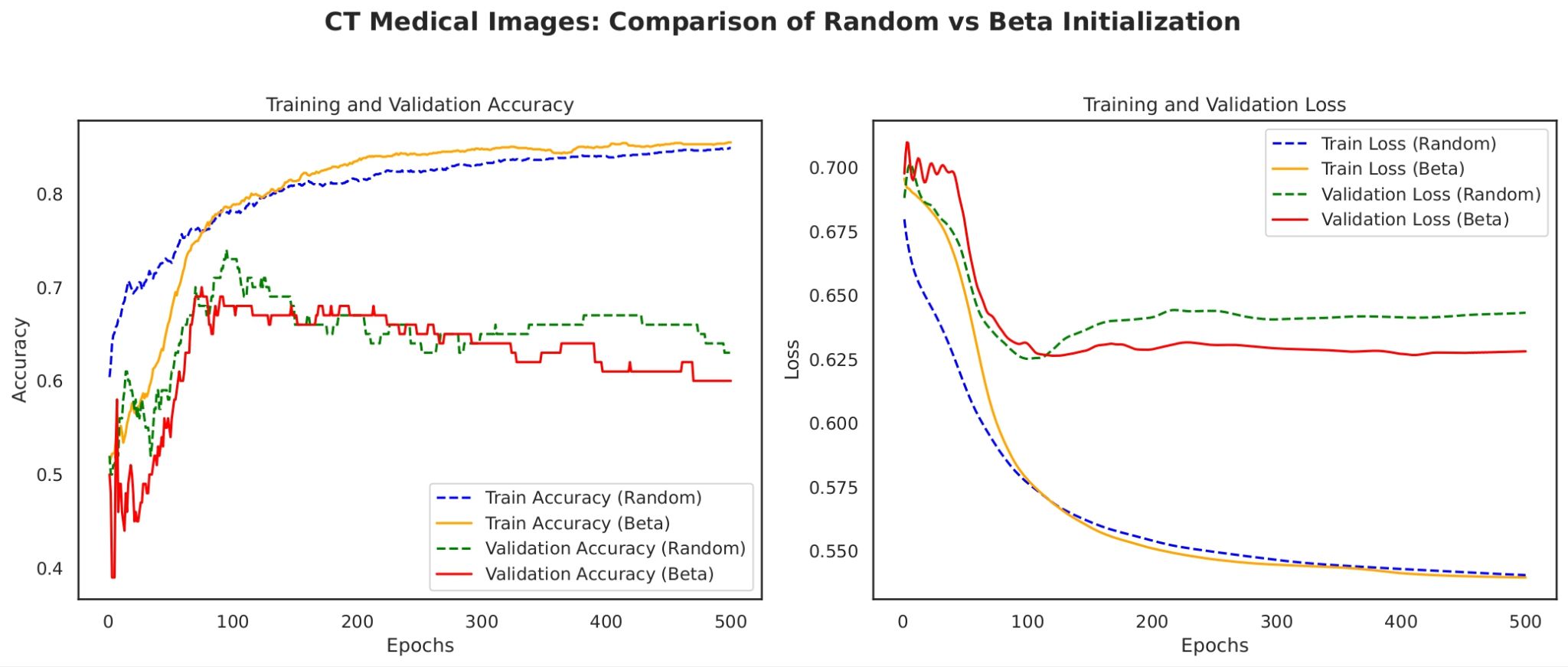
The results consistently demonstrate that Beta initialization provides a structured starting point for model parameters, facilitating better optimization and generalization across datasets. Beta initialization’s superior performance is particularly evident in tasks involving complex or larger datasets, where it reduces losses and improves accuracy across training, validation, and testing phases.  
  
While Random initialization performs adequately in simpler scenarios, Beta initialization offers clear advantages, especially in improving convergence and handling challenging data distributions. Overall, these findings validate the effectiveness of Beta initialization in optimizing QNNs, as reflected in both the tabulated results and graphical trends.  
  
The weighted ensemble approach shows that increasing the weight given to Beta initialization ( [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)) generally improves accuracy and reduces loss across three out of four datasets (CT Medical Images, MNIST, and FashionMNIST). This suggests that Beta initialization enhances model performance, particularly for complex datasets. However, in the SKLearn Digits dataset, performance peaks at [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 0.4, after which accuracy slightly declines. This indicates that while Beta initialization is effective, the optimal weighting may vary depending on the dataset’s complexity. Overall, the weighted ensemble approach offers a flexible strategy that adapts to different datasets, improving performance through appropriate weighting of Beta initialization.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Alpha** | **CT Medical Images** | | **SKLearnDigits** | | **MNIST** | | **FashionMNIST** | |
| **Avg Cost** | **Avg Acc** | **Avg Cost** | **Avg Acc** | **Avg Cost** | **Avg Acc** | **Avg Cost** | **Avg Acc** |
| **0.0** | 0.6131 | 65% | 1.5560 | 95.60% | 1.5893 | 89.05% | 1.6218 | 77.33% |
| **0.1** | 0.6117 | 65% | 1.5538 | 96.33% | 1.5852 | 89.36% | 1.6183 | 77.54% |
| **0.2** | 0.6105 | 65% | 1.5517 | 95.97% | 1.5811 | 89.71% | 1.6148 | 77.80% |
| **0.3** | 0.6093 | 65% | 1.5496 | 96.70% | 1.5771 | 90.00% | 1.6113 | 78.15% |
| **0.4** | 0.6083 | 70% | 1.5476 | 97.44% | 1.5730 | 90.20% | 1.6078 | 78.39% |
| **0.5** | 0.6073 | 70% | 1.5456 | 96.70% | 1.5690 | 90.47% | 1.6044 | 78.69% |
| **0.6** | 0.6066 | 70% | 1.5437 | 96.34% | 1.5650 | 90.62% | 1.6009 | 78.84% |
| **0.7** | 0.6059 | 70% | 1.5418 | 95.97% | 1.5610 | 90.63% | 1.5975 | 79.09% |
| **0.8** | 0.6054 | 75% | 1.5400 | 95.60% | 1.5570 | 90.74% | 1.5942 | 79.29% |
| **0.9** | 0.6050 | 75% | 1.5383 | 95.97% | 1.5531 | 90.71% | 1.5908 | 79.43% |
| **1.0** | 0.6047 | 75% | 1.5366 | 95.60% | 1.5492 | 90.78% | 1.5875 | 79.46% |

Table 2. Comparison of Results As Alpha Increases

A graph of different colored lines

Description automatically generatedA comparison of a graph

Description automatically generated

**MNIST: Comparison of Random vs Beta Initialization**

A graph showing the number of numbers and the number of objects

Description automatically generated with medium confidenceA graph of a training and validation curve

Description automatically generated

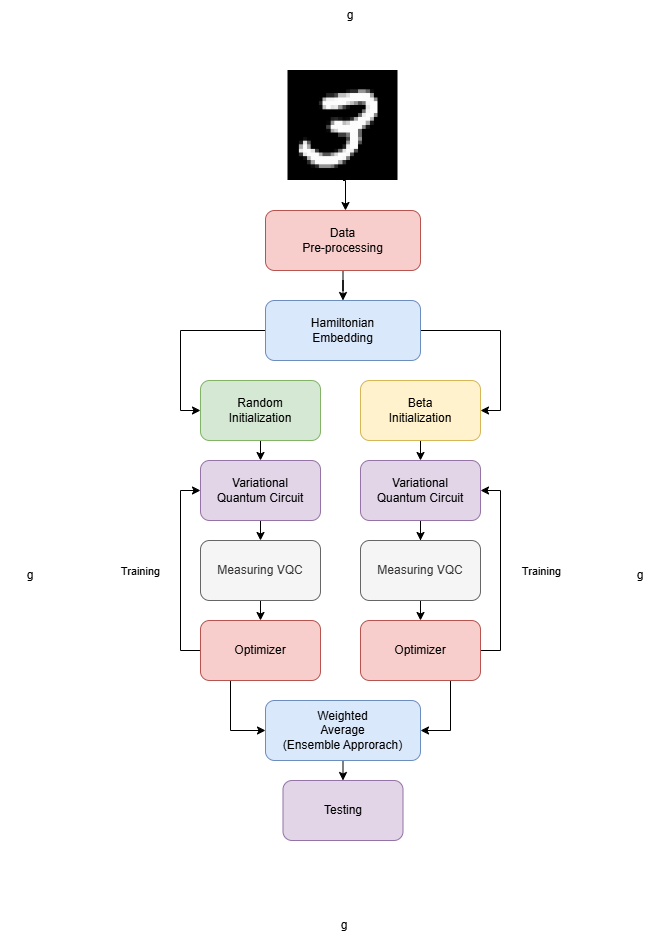
A graph with different colored lines

Description automatically generatedA graph of a graph

Description automatically generated with medium confidence

Figure 3: **Top**: Accuracies for the ensemble approach across all 4 datasets. **Bottom**: Losses for the ensemble approach across all 4 datasets.

1. System Diagram



1. Goals For FYP-II

**Future Work**

Building on the demonstrated advantages of Hamiltonian embedding combined with ancillary qubits, future research can explore several directions to further enhance the effectiveness and scalability of quantum-assisted learning systems:

1. **Advanced Hamiltonian Embedding Techniques**
   * Investigate optimized Hamiltonian formulations that capture even richer high-dimensional features and further improve the expressivity of quantum models.
   * Develop adaptive embedding methods that dynamically adjust Hamiltonian parameters based on input data characteristics.
2. **Enhanced Ancillary Qubit Integration**
   * Explore methods for incorporating multiple ancillary qubits to scale the representational capacity of QCNNs for complex datasets.
   * Analyze the impact of ancillary qubits on mitigating barren plateaus in deeper quantum circuits and extend these findings to hybrid quantum-classical architectures.
3. **Synergistic Effects in Multi-Qubit Systems**
   * Study the interplay between Hamiltonian embedding and ancillary qubits to design more robust nonlinear decision boundaries in high-dimensional Hilbert spaces.
   * Quantify the contribution of ancillary qubits to improving fidelity under varying levels of quantum noise and decoherence.
4. **Application to Diverse Datasets**
   * Apply this synergistic framework to more complex real-world datasets, such as high-resolution medical imaging, to evaluate its generalizability and scalability.
   * Investigate its utility in other domains, such as quantum-enhanced natural language processing or financial modeling.
5. **Optimization of Quantum Circuits**
   * Focus on reducing the circuit depth and computational overhead while maintaining the performance benefits of Hamiltonian embedding and ancillary qubits.
   * Explore variational techniques to fine-tune both embedding parameters and ancillary qubit configurations for task-specific objectives.

By addressing these areas, future work can further establish Hamiltonian embedding and ancillary qubit integration as cornerstone techniques in quantum machine learning, paving the way for more precise and efficient quantum-assisted AI systems.

1. Conclusion

**Funding**: The authors received no outside funding during this project.

**Data/Code availability**: Code and data are available on [GitHub](https://github.com/ClawSwipe/QMLFYP).

In this study, we explored the potential of Hamiltonian embedding within Quantum Neural Networks (QNNs) by comparing Random and Beta initialization strategies across four diverse datasets: Kaggle CT Medical Images, SKlearn Digits, MNIST, and FashionMNIST. Our findings consistently demonstrated that Beta initialization provides a clear advantage, enhancing model performance by mitigating barren plateaus and enabling better optimization. Beta initialization not only outperformed Random initialization in most cases but also displayed superior generalization capabilities, particularly in complex datasets with intricate data distributions. However, it remains unclear whether the observed improvements are solely due to favorable initial parameter positions or if there exists a deeper connection between the distribution of data and the parameters involved in quantum circuits, which remains an open question [6].  
  
The ensemble approach, which combined the outputs of Random and Beta initialization models, further highlighted the flexibility of leveraging complementary strategies. Notably, the SKlearn Digits dataset benefitted from a balanced weighting of Random and Beta initialization, achieving optimal performance at [A black background with a black square

Description automatically generated with medium confidence](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5C#0)= 0.4. This suggests that for certain datasets, a hybrid approach might outperform pure initialization methods. Future work could explore dynamic ensemble weighting techniques, where the weighting is adapted based on dataset-specific characteristics during training.

References

[1] B. Albertina, M. Watson, C. Holback, R. Jarosz, S. Kirk, Y. Lee, K. Rieger-Christ, and J. Lemmerman. The cancer genome atlas lung adenocarcinoma collection (tcga-luad).

[https://www.cancerimagingarchive.net/collection/tcgaluad, 2016](http://cancerimagingarchive.net)

[2] E. Alpaydin and C. Kaynak. Optical recognition of handwritten digits. UCI Machine Learning Repository, 1998.

[3] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C.Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean,Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J.Coles. Variational quantum algorithms. Nature Reviews

Physics, August 2021. 1, 3

[4] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. Quantum,December 2019. 2

[5] Maxwell Henderson, Samriddhi Shakya, Shashindra Pradhan, and Tristan Cook. Quanvolutional neural networks: Powering image recognition with quantum circuits.

[6] Sabrina Herbst, Vincenzo De Maio, and Ivona Brandic. On optimizing hyperparameters for quantum neural networks.Applied Intelligence, 2024. 2, 3, 9

[7] Itseez. Open source computer vision library. [https://github.com/itseez/opencv](http://github.com/itseez/opencv), 2015. Accessed:November 11, 2024. 3

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980,2017. Published in 2014, updated in 2017. 4

[9] Ankit Kulshrestha and Ilya Safro. Beinit: Avoiding barren plateaus in variational quantum algorithms. In Proceedings of the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2022. 2, 3

[10] Phuc Q. Le, Fangyan Dong, and Kaoru Hirota. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. Quantum Information Processing, 2011. 1

[11] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. [http://yann.lecun.com/exdb/mnist,](http://yann.lecun.com/exdb/mnist)2010. Accessed: November 11, 2024. 3

[12] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy,Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. Nature Communications, November 2018. 1

[13] Carlos Ortiz Marrero, M´aria Kieferov´a, and Nathan Wiebe.Entanglement induced barren plateaus. PRX Quantum, October 2021. 1

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B.Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss,V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M.Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. Journal of Machine Learning Research,2011. 3

[15] Adri´an P´erez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster,

and Jos´e I. Latorre. Data re-uploading for a universal quantum classifier. Quantum, 2020.

[16] Kevin Scott Mader. Ct medical images.

<https://www.kaggle.com/datasets/kmader/siim-medical-images>, 2017. Accessed: November 11, 2024.

[17] TorchVision maintainers and contributors. Torchvision: Pytorch’s computer vision library. [https://github.com/pytorch/vision, 2016. Accessed: November 11, 2024](https://github.com/pytorch/vision)

[18] Peiyong Wang, Casey R. Myers, Lloyd C. L. Hollenberg,and Udaya Parampalli. Quantum hamiltonian embedding of images for data reuploading classifiers. arXiv preprint arXiv:2402.02401, 2024. 1, 2, 4

[19] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma,Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise induced barren plateaus in variational quantum algorithms.Nature Communications, November 2021. 1

[20] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville,Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen.Exploring entanglement and optimization within the hamiltonian variational ansatz. PRX Quantum, December 2020.2

[21] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion MNIST: A novel image dataset for benchmarking machine learning algorithms[.](https://github.com/zalando-)

<https://github.com/zalandoresearch/fashion-mnist>, 2017. Accessed: November 11, 2024. 3

[22] Rui Yang, Samuel Bosch, Bobak Kiani, Seth Lloyd, and Adrian Lupascu. Analog quantum variational embedding classifier. arXiv preprint arXiv:2303.00045, 2023. 1

[23] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. Phys. Rev. X, June 2020. 2