Team Name: Quantum HQ
Team Participants: Anusha Agarwal, Neha Chandran, Sophia Hou, Surbhi Singla (contact emails linked)
Academic Cohort: Thomas Jefferson High School Research QLab

High Level Overview: Our solution aims to develop a streamlined quantum machine learning (QML) model to predict phenotypes from the calcium signals dataset. By applying a post-variational optimization approach to the input state vectors, we can generate an enhanced set of feature vectors. These optimized features will then be used to train a classical dense neural network (DNN), achieving accurate predictions with fewer training epochs.

1 Neural Problem Framing

The calcium imaging dataset we will be using is taken from *Bowen et al*. This data is structured such that each row represents a sample collected over a short window, with entries corresponding to the normalized calcium fluorescent values for every neuron. Specifically, we will consider `allPlanesVariables27-Feb-2021.mat` as the input file from the zip folder.

2 Quantum Modeling Strategy

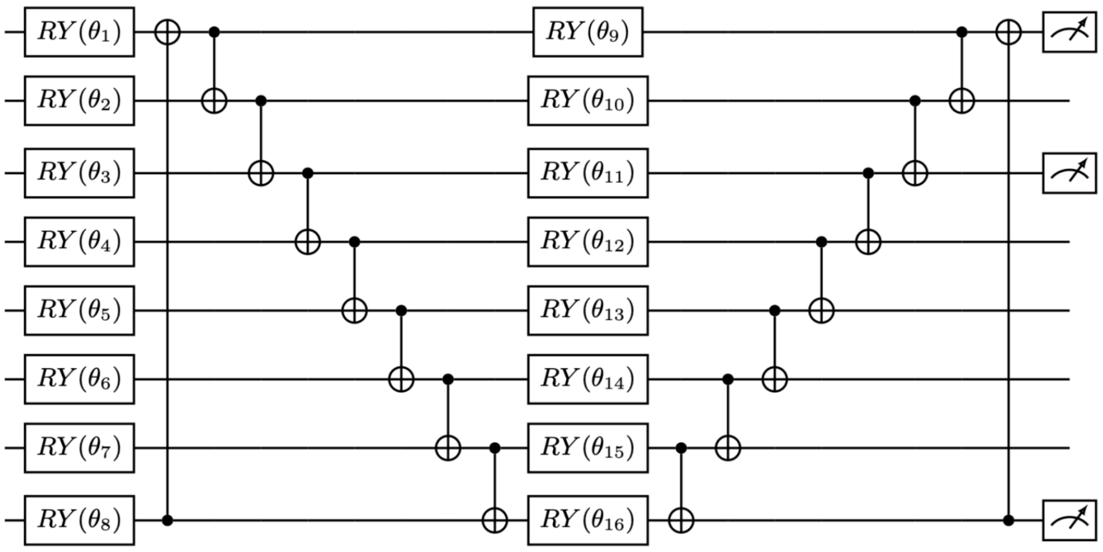
Our approach implements a hybrid quantum-classical architecture that augments classical learning with quantum-optimized feature embeddings. The model harnesses a parameterized quantum circuit to transform raw calcium imaging traces into expressive quantum feature vectors, capturing entanglement-style dependencies that classical models often overlook.

Each calcium trace (a vector of normalized $\Delta F/F$ values) is encoded into a quantum state via amplitude-based or angle-based rotation encoding. The feature map applies alternating RZ and RX gates, where each neuron's fluorescence value determines the rotation angle θ.

We then apply a variational ansatz composed of:

- RY rotations for parameterized learnability
- Two full layers of CNOT gates, cyclically connecting every pair of qubits in both directions

This structure maximizes entanglement and state space exploration, allowing the quantum model to capture higher-order, non-linear dependencies across neuron activity patterns. See the circuit below.



The full quantum circuit will then be fed to an optimization loop, which will iteratively invoke a classifier, loss function, and cost function to generate enhanced feature vectors. These optimized vectors will ultimately serve as inputs to a classical dense neural network (DNN) with an $N - 2N - 1$ architecture, where N corresponds to the dimensionality of each input vector.

Classical DNNs often struggle to learn dependencies that involve combinatorially entangled neurons or non-linear correlations across sparse time-series windows. The quantum layer enables entanglement-style mixing, learning co-activations across neuron groups that don't appear linearly. Furthermore, feature space warping via variational optimization yields embeddings more separable by a classical classifier. In effect, we can improve generalization with fewer parameters and data samples using a quantum neural network as opposed to a classical one.

3 Classical vs. Quantum Comparison Plan

To evaluate the impact of quantum-enhanced learning, we will compare our hybrid QML model against state-of-the-art classical baselines using both unsupervised and supervised learning metrics.

We define the following classical techniques for baseline comparison:

- **PCA (Principal Component Analysis):** For linear dimensionality reduction, capturing the directions of maximal variance in neural activity.
- **ICA (Independent Component Analysis):** To extract statistically independent motifs, often used in neuroscience to uncover latent activity sources.
- **k-means Clustering:** Applied to PCA/ICA-reduced features to identify activity motifs and neuron assemblies.

We will run both pipelines (classical and quantum) on identical training and test splits of the calcium dataset. Post-encoding, we will train identical DNN classifiers on both sets of feature vectors and compare performance over multiple metrics and random seeds. Our hypothesis is that the variationally optimized quantum embeddings will yield higher accuracy and robustness with fewer training samples, especially in capturing biologically meaningful activity patterns.

4 Data Preprocessing & Encoding Pipeline

To prepare the calcium imaging data for downstream analysis, we construct a preprocessing pipeline that transforms raw fluorescence traces into a compact, noise-reduced representation suitable for modeling or visualization.

From the zDFF variable in the MATLAB files, we extract calcium fluorescence traces for neurons in planes 2 through 6 (excluding planes 1 and 7 as per the README). Each plane contains an array of shape 541 x 6100 *(n_neurons, t_timepoints)*. These arrays are concatenated vertically across all valid planes to form a unified 2D matrix.

Then, we will employ the following data preprocessing techniques:

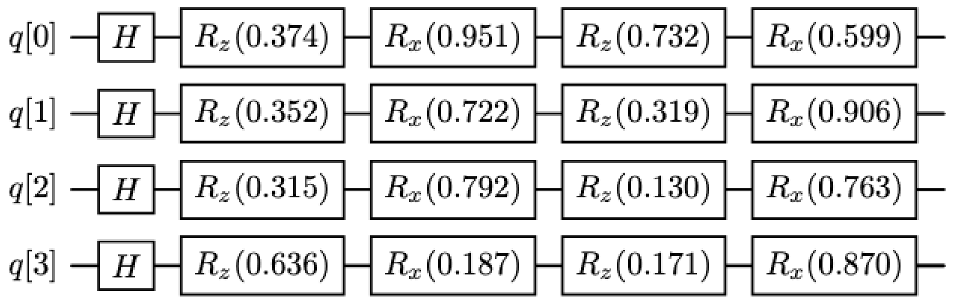
a. Noise Filtering & Normalization
Fluorescence traces are denoised using a low-pass or moving average filter, then normalized (e.g., z-score or min-max scaling) to ensure comparability across neurons.

b. Spike Binning & Thresholding
Optional binning aggregates activity over fixed time windows. Thresholding identifies putative spikes by marking values exceeding a set fluorescence change, producing binary activity patterns.

c. Dimensionality Reduction
We apply PCA or t-SNE to reduce the high-dimensional activity data into a lower-dimensional feature space, capturing key temporal or spatial activity patterns.

In order to construct our feature map using quantum circuits, we must take a smaller partitioning of the matrix due to hardware constraints of quantum technology. For this challenge, we will implement a 127 x 500 matrix, where each entry corresponds to the normalized calcium fluorescence intensity. We will encode these calcium trace values as parameterized θ angles, which determine the rotations applied through a sequence of alternating RZ and RX gates.

See below for a 4 x 4 sample visualization of our feature map.



5 Resource Requirements & Execution Plan

We will first build our model using IBM's Qiskit and collect data from IBM Eagle r3 hardware, a 127-qubit device which matches with the dimension of our circuit. To verify, we will use the qBraid QIR simulator. This model has a circuit depth of 8 – Hadamard layer, RZ layer, RX layer, RY layer, 2 cyclic CNOT layers, second RY layer, and another 2 cyclic CNOT layers.