NovaQ- Phase 2 Submission NeuroQuantum Nexus

1. Neural Signal Target & Analysis Goal

We selected the electrophysiological spike train data (Buccino et al.) from the DANDI archive (Dandiset 000034, sub-P29-16-05-14-retina02-left) as our dataset of interest (given its 6GB size). We looked into the high-resolution Neuropixels recordings with up to 384 channels of electrical activity, extracted using SpikeInterface and PyNWB in Python. We originally attempted doing it with Lindi and then shifted to PyNWB.

Our target is to analyze spike synchrony and infer higher-order correlations that reflect collective firing events and network dynamics. These patterns, particularly synchronous population bursts across spatially distributed channels, are implicated in encoding information in sensory systems and are often missed by pairwise statistical models. We aim to detect and classify these collective events using a hybrid classical–quantum model that leverages quantum graph processing.

2. Quantum Model Structure & Theoretical Basis

Architecture Overview:
We propose a hybrid classical–quantum pipeline:

- CNN Module (Classical): Detects and encodes spatiotemporal spike features from multi-channel signals using 1D convolutional layers. Ideally, we would like to ensemble this with an LSTM (layers capture long-range dependencies and recurring motifs), but for now our strategy will remain a CNN.
- Graph Construction: Each electrode/channel becomes a node. Edges represent spike-time correlations or spatial proximity. Node features include spike waveform embeddings and spatial XYZ coordinates. We also considered for the calcium data set (which we are not using) feeding in parallel the location for a an entangled cluster creation, evaluating how to apply that here, likely will emerge once we work on it further and evaluate performance.
- QGNN Module (Quantum): Processes subgraphs of correlated channels using a shallow variational quantum circuit. Node features are angle-encoded into rotation gates. Edge structure defines entangling gate layout (e.g., CNOT or RZZ). We acknowledge the decision to encode with angles has it's implications, however for this strategy it seems better suited.

Though the Quantum Output could ideally predict whether the subgraph corresponds to a high-synchrony event (binary classification), it could be also used as an output quantum node embeddings for downstream clustering, acknowledging the historical performance limitations of QGNN's. So we may expand the pipeline.

Why QGNN:
Quantum Graph Neural Networks (QGNNs) can leverage entanglement and superposition to model non-local dependencies across the graph, potentially uncovering higher-order synchrony missed by classical GNNs. By using parametrized entangling gates guided by the adjacency matrix, our model emulates message-passing through quantum states.

This approach is NISQ-feasible by restricting the QGNN to small, pre-filtered subgraphs (e.g., 8–16 nodes) and operating with shallow-depth variational circuits. Our quantum layer will be trained in a hybrid loop using classical optimizers (e.g., SPSA or Adam).

3. Classical Benchmark Method for Comparison

Despite our research in spike forest for a benchmark that would be current we didn't seem to find one that we could just leverage without having to re-implement a new classical strategy, so we are considering the following as candidates – again given a is part of what we think will work best for the input of our hybrid pipeline we may need to go with b for a different baseline.
a. 1D CNN + LSTM Hybrid:
- CNN layers extract short-range temporal features from binned spike trains.
- LSTM layers capture long-range dependencies and recurring motifs.
- Dense output layer for event classification.

b. GNN (Classical):
- Graph built from spike correlations.
- Classical GNN (e.g., GraphSAGE or GCN) trained to predict synchrony or cluster membership.

We will evaluate classical vs. hybrid quantum performance in terms of:
- Accuracy and potentially F1 score (event classification),
- Interpretability (e.g., node-level attribution),
- Sensitivity to higher-order correlations (e.g., distinguishing pairwise vs. triadic synchrony).

4. Preprocessing Pipeline & Quantum Encoding

Tools:
- SpikeInterface: For multi-channel spike detection and sorting (all 9 steps).
- PyNWB: To load NWB-format data from DANDI.
- Optional: LINDI + Neurosift for data exploration and channel selection. We exported the Lindi file and

Pipeline:
1. Spike Extraction:
   - Bandpass filtering, noise removal.
   - Binning into 1–5 ms time windows.
2. Dimensionality Reduction:
   - Autoencoders (Neural Network-based) to compress across 384 channels to ~16 features per bin. Assumption here is we can train different versions for population spikes, per-neuron waveforms, etc.

3. Graph Construction:
   - Nodes = active channels; features = CNN + XYZ.
   - Edges = synchrony (e.g., Pearson r or bin-wise co-firing).

4. Quantum Encoding:
   - Angle encoding for node features (e.g., spike rate → RX/RY angles).
   - Entanglement applied via CNOT gates structured by adjacency matrix.
   - Optionally use adjacency encoding for topology-informed parameterization.
5. Training Loop:
   - Variational quantum circuit per subgraph.
   - Hybrid classical optimizer adjusts parameters for loss minimization.

5. Execution Platform, Resource Needs & Phase 3 Plan

Preferred Platform:
We will use IBM Quantum hardware, with simulations on Qiskit Aer and hardware execution on devices like:
- IBM Heron r1/r2 (up to 133 qubits),
- IBM Eagle r3 (for larger circuit simulation).

We also plan to utilize:
- Qiskit Machine Learning + TorchConnector for hybrid PyTorch integration, and if available Fire Opal (available via Qiskit's premium offerings) by Q-CTRL for AI-powered error mitigation and dynamic noise suppression to reduce circuit depth
- qiskit-metal (if needed for future circuit layout optimization).

Estimates:
- Qubits required: 8–16 per subgraph (we would ideally only be working on a subset of the data)
- Circuit depth: ≤ 20 layers
- Simulated training for our small hybrid cluster outputs, with real hardware validation on small test cases.

Phase 3 Readiness:
We will focus on expanding subgraph coverage, benchmarking, and pipeline refinement.

References (page 4)

- Yanhu Chen, Cen Wang, Hongxiang Guo, Xiong Gao, Jian Wu, Accelerating spiking neural networks using quantum algorithm with high success probability and high calculation accuracy, Neurocomputing, Volume 493, 2022, Pages 435-444, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2022.02.004.

- Petr Nejedly, Valentina Hrtonova, Martin Pail, Jan Cimbalnik, Pavel Daniel, Vojtech Travnicek, Irena Dolezalova, Filip Mivalt, Vaclav Kremen, Pavel Jurak, Gregory A Worrell, Birgit Frauscher, Petr Klimes, Milan Brazdil, Leveraging interictal multimodal features and graph neural networks for automated planning of epilepsy surgery, *Brain Communications*, Volume 7, Issue 3, 2025, fcaf140, https://doi.org/10.1093/braincomms/fcaf140

- **PyNWB Processign of Extracellular Electrophysiology Data** https://pynwb.readthedocs.io/en/stable/tutorials/domain/ecephys.html#sphx-glr-tutorials-domain-ecephys-py

Buccino, Alessio & Winter, Olivier & Bryant, David & Feng, David & Svoboda, Karel & Siegle, Joshua. (2023). Compression strategies for large-scale electrophysiology data. Journal of Neural Engineering. 20. 10.1088/1741-2552/acf5a4.
https://www.researchgate.net/publication/373576690_Compression_strategies_for_large-scale_electrophysiology_data#:~:text=Objective,and%20simulated%20recordings%20from%20Neuropixels