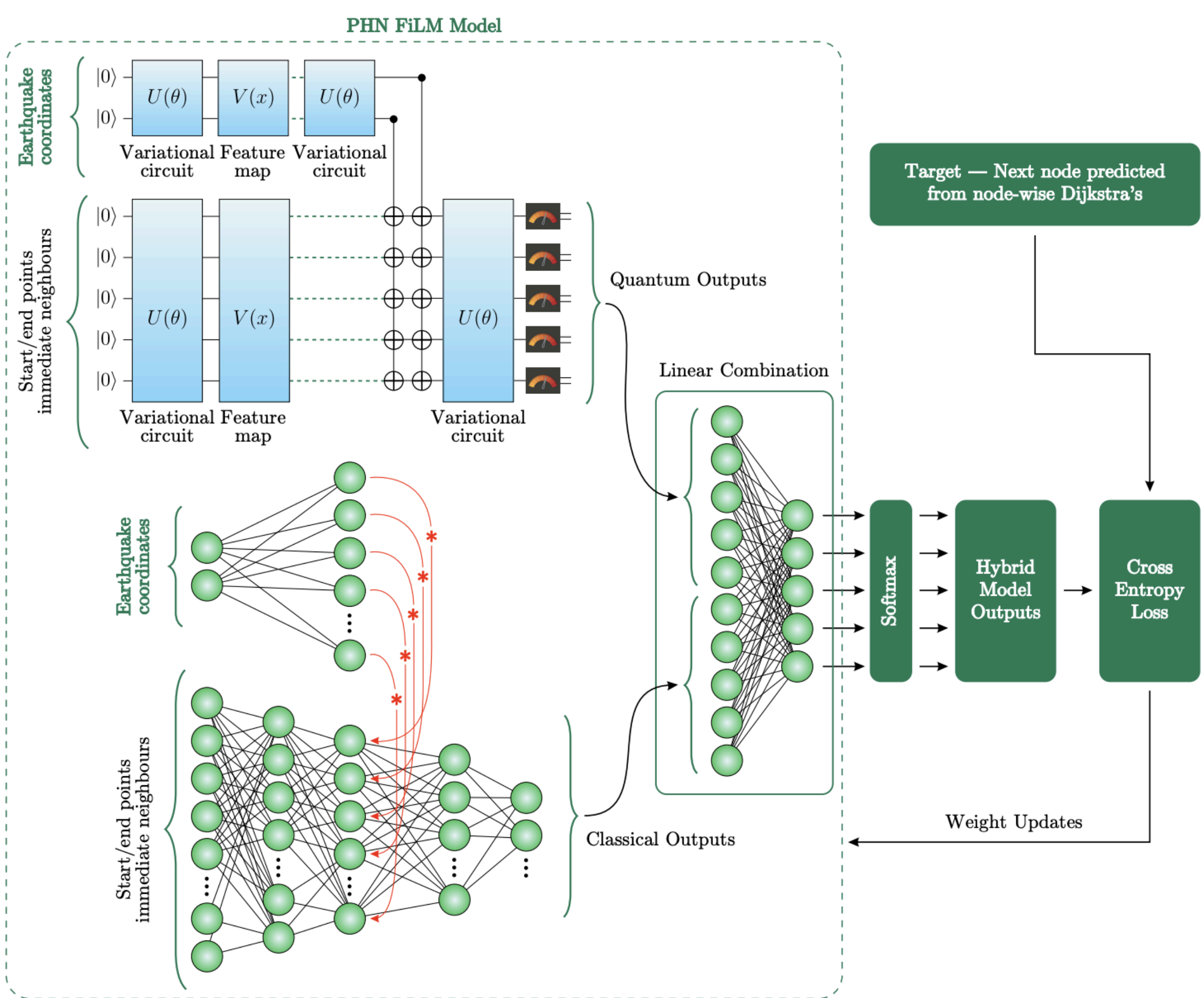
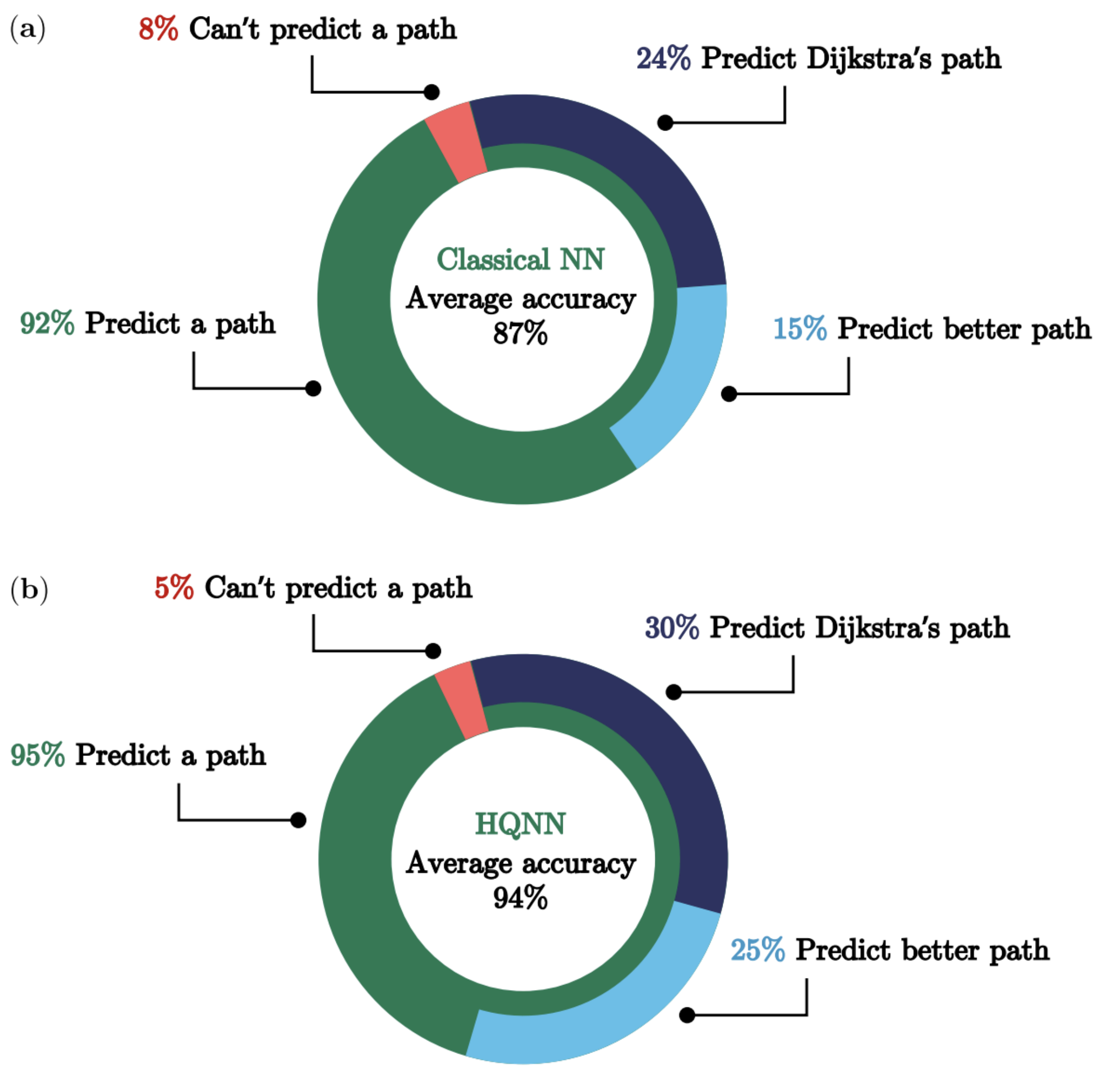
Notes:

1. Objective of the paper: To find how supervised hybrid QML can optimize emergency evacuation plans for cars during natural disasters.
2. The situation is modeled as a shortest-path problem on an **uncertain and dynamically evolving map.**
3. Novel feature used: Quantum feature-wise linear modulation(FiLM) neural network parallel to a classical FiLM network to imitate Dijkstra's node-wise shortest path algorithm on a **deterministic dynamic graph**.
4. Dynamic Effects include land deformation, collapsing buildings, or debris.
5. Comparison and Context
    1. **Optimality**:
        - **Dijkstra's Algorithm**: Always finds the shortest path, guaranteeing optimality without any conditions.
        - ***A* Algorithm***: This can offer faster performance by exploring fewer nodes but relies on the heuristic's properties to guarantee optimality. If the heuristic is not admissible and consistent, A* might not find the optimal path.
    2. **Performance**:
        - **Dijkstra's Algorithm**: Explores all possible paths in an expanding manner until the shortest path to the target is found. This can be computationally expensive, especially in large graphs.
        - ***A* Algorithm***: By using heuristics, A* can significantly reduce the search space, often resulting in faster performance. However, the speedup depends on the quality and accuracy of the heuristic. *(Heuristic function- typically an estimate of the distance from the current node to the goal)*
6. The paper introduces a hybrid qml approach that requires only local information, as opposed to the global information required by classical Dijkstra's algorithm, to mimic the node-wise Dijkstra's algorithm in terms of path quality on a dynamic graph.
7. Model efficiency is done through ZX Calculus, Fourier embedding analysis, and Fisher expressivity:
    a. **ZX Calculus**: A graphical language for quantum computing that simplifies the representation and manipulation of quantum operations.
    b. **Fourier Embedding Analysis**: A technique that uses Fourier transforms to represent sequences in the frequency domain, capturing periodic and structural properties.
    c. **Fisher Expressivity**: A measure of a model's expressive power based on Fisher information, indicating the model's sensitivity to changes in underlying parameters.
8. The study focuses on Earthquake emergencies.
9. The objective is to obtain a route for evacuating cars, which minimizes travel time.

10. Python OSMnx package is used to convert any selected region of a map into a graph.
11. Steps followed in the Hybrid Quantum Neural Network(HQNN) model:
    a. Dataset produced by simulating an earthquake at randomized coordinates in the city and then collecting routing data for each earthquake simulation.
    b. Both the classical and quantum neural networks and fed with the following input features:
        i. Earthquake Coordinates
        ii. Start/End points, immediate neighbors
    c. The output of classical FiLM and quantum neural networks are combined to produce the outcome which is fed to a trainable layer.
    d. The outcome of the trainable layer is 5 numbers that act as the logit(raw outputs) layer of the node classifier.
    e. Finally, the neighboring node corresponding to the highest number is chosen as the next node.



12. Results and Analysis: Results show that HQNN performs better than the Classical NN. The runtime of HQNN only depends on the number of nodes in one path which in the worst-case scenario includes all nodes $O(n_{nodes})$.



13. Practical Analysis:
    A.PHNs can suffer from primacy, where the network favors the output of either the MLP (Multilayer Perceptron) or the VQC (Variational Quantum Circuit), leading to suboptimal performance. To evaluate the contributions of each sub-network, the weights of the final layer (a 5×10 matrix) were analyzed. This analysis showed two key points: 1) the weights are similar between the VQC and MLP, indicating no primacy, and 2) the VQC weights show smoother transitions, while the MLP weights are more irregular. Relative contribution is quantified by the Frobenius norm
    B. Performance on QPU: On a 25 qubit ion-based QC, with 3 decision points, the tasks took UB 10 mins including queuing effects. Comparison between VQC and QPU outputs

qualitatively shows the high correlation between the simulator and actual hardware adjusting for shot noise and gate errors.

Questions/Suggestions:

A. *"Dijkstra's algorithm effectively finds the optimal path on a static graph, and while algorithms like A* [18] might offer faster alternatives, Dijkstra's is the only one with an optimality guarantee [19]. However, this algorithm struggles to find the shortest path in an evolving and uncertain situation."*

Is it worth looking into tradeoffs between optimality and faster computation? Does quantum computing speed up Dijkstra's algorithm enough to compete with algorithms like A* with classical computing?

B. *"The SL model is trained on a dataset with labels generated by node-wise Dijkstra's algorithm. This way, the HQNN approximates Dijkstra's algorithm while only accessing a limited portion of the map."*

The functionality of the neural network sounds similar to how the A* algorithm works. How much more efficient is the algorithm than A*? Does the model, once trained, predict escape routes without having to look at or negatively bias non-optimal routes?

C. Was the modeling based on earthquakes used for its geometrical simplicity? Would other disasters like floods or tsunamis require models with different parameters (representing the spread of the disaster effects using lines/cylindrical growth rather than circles and epicenters. Similar to electrical fields v/s magnetic fields), replacing the radial distance from epicenter to perpendicular distance from a Tsunami or flood.

D. Can the earthquake model work for disasters like sinkholes or volcanic eruptions whose areas of effects spread radially like an earthquake?