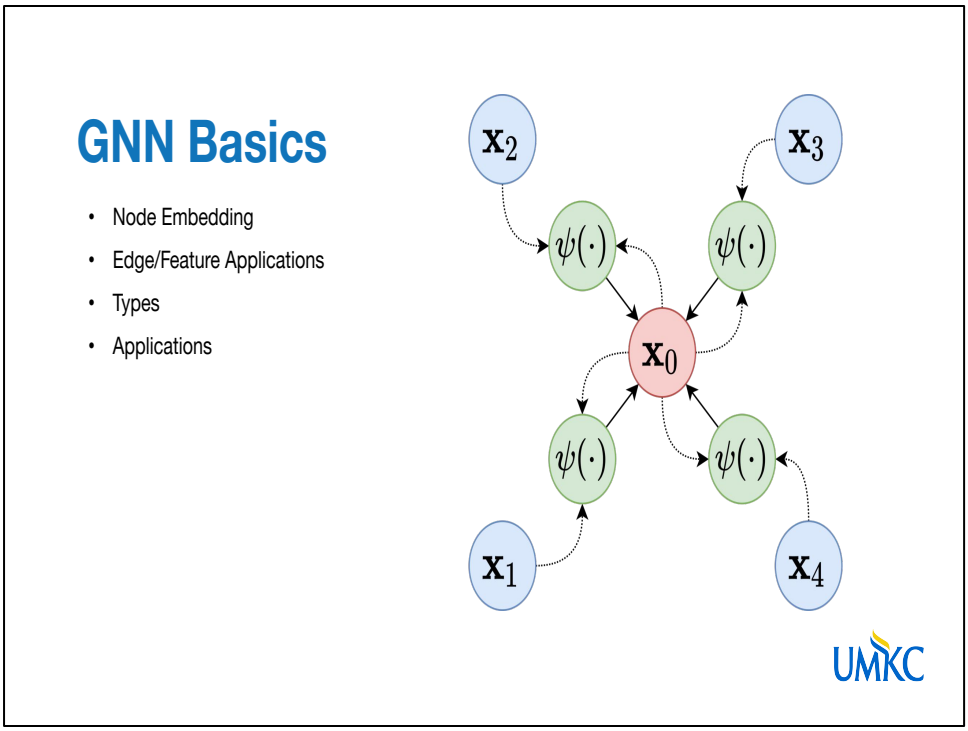
UMKC

Quantum AI Mini Seminar Series:

Session 4: Quantum Machine Learning

- Introduction to Graph Neural Networks (GNNs)
- Quantum Parallelism for Graph-Structured Data
- Quantum Graph Neural Networks (QGNNs)
- Quantum Algorithms for Graph Processing in QGNNs
- Challenges and Future Directions for QGNNs



Node Embedding

- Node embeddings map nodes in a graph into a lower-dimensional vector space, capturing **topological structure** and **node attributes**.
- Mathematically, embeddings are often learned through **minimizing a loss function** that encourages connected nodes to have similar embeddings, represented as: L=∑(i,j)∈Ef(hi,hj)\mathcal{L} = \sum_{(i, j) \in \mathcal{E}} f(\mathbf{h}_i, \mathbf{h}_j)L=(i,j)∈E∑f(hi,hj) where hi\mathbf{h}_ihi and hj\mathbf{h}_jhj are embeddings of connected nodes iii and jjj, and fff is a similarity function, such as the dot product or cosine similarity.

Edge/Feature Applications

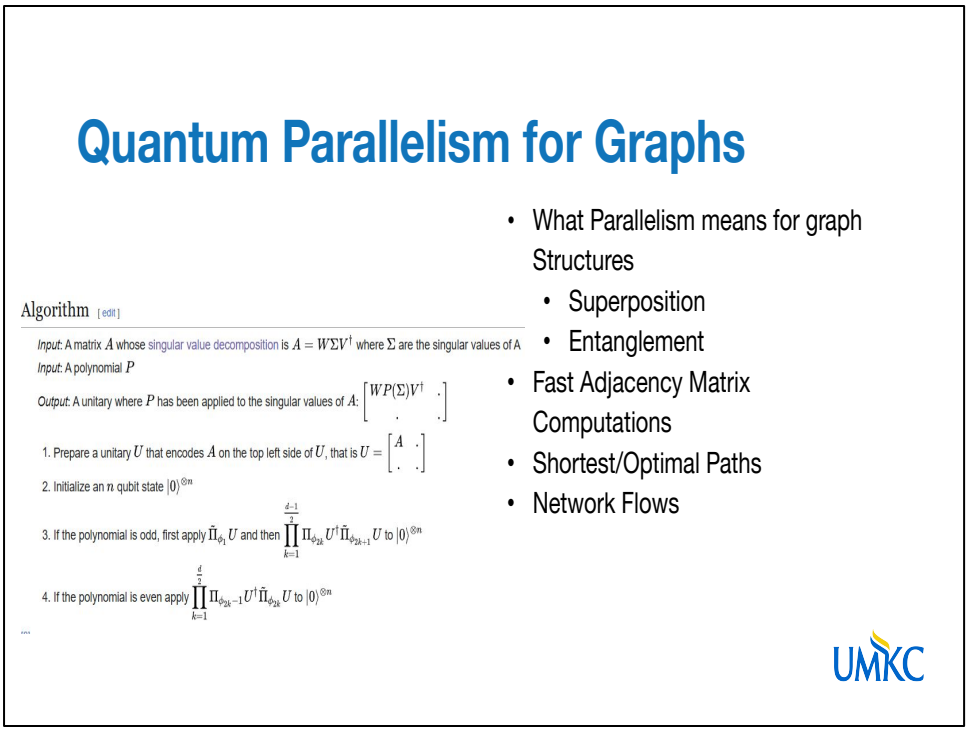
- GNNs leverage edge features to weigh relationships between nodes, enabling tasks such as **link prediction**, **edge classification**, and **graph classification**.
- Edge weights modify the aggregation function, e.g., in a **Graph Convolutional Network (GCN)**: hi(k+1)=σ(∑j∈N(i)1didjW(k)hj(k))\mathbf{h}_i^{(k+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \frac{1}{\sqrt{d_i d_j}} \mathbf{W}^{(k)} \mathbf{h}_j^{(k)} \right)hi(k+1)=σj∈N(i)∑didj1W(k)hj(k) where did_idi and djd_jdj are degrees of nodes iii and jjj, respectively, and W(k)\mathbf{W}^{(k)}W(k) is a layer-specific learnable weight matrix.

Types of GNNs

- **(GAT)**, and **Message Passing Neural Networks (MPNN)**. Each type differs in how it aggregates information from neighboring nodes, with **attention mechanisms** in GATs allowing for differentiated weights across neighbors.

Applications

- GNNs are applied in diverse domains such as **social networks**, **recommendation systems**, **drug discovery**, and **biological network analysis**, leveraging their ability to handle data with **non-Euclidean structures**.



What Parallelism Means for Graph Structures

- Quantum parallelism enables the simultaneous evaluation of multiple states due to **superposition**. For graphs, this means that all possible paths, connections, or configurations can be processed concurrently, enhancing the efficiency of combinatorial graph problems.

Superposition

- Superposition in qubits allows for encoding a combination of multiple graph states. For a graph with nodes represented as states $|0\rangle$|0\rangle$|0\rangle$ and $|1\rangle$|1\rangle$|1\rangle$, superposition enables representing all nodes simultaneously: $|\psi\rangle$=α$|0\rangle$+β$|1\rangle$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$|\psi\rangle$=α$|0\rangle$+β$|1\rangle$ where $|\alpha|2+|\beta|2=1$|\alpha|^2 + |\beta|^2 = 1$|\alpha|2+|\beta|2=1$, allowing for simultaneous exploration of paths and adjacency relations.

Entanglement

- Entanglement enables correlation between qubits representing different nodes or edges, which can represent dependencies in graph structures. For example, two entangled qubits, $|\psi\rangle=12(|00\rangle+|11\rangle)$|\psi\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$|\psi\rangle$=21($|00\rangle$+$|11\rangle$), capture a **mutual dependence**—useful for encoding linked nodes or edge properties.

Fast Adjacency Matrix Computations

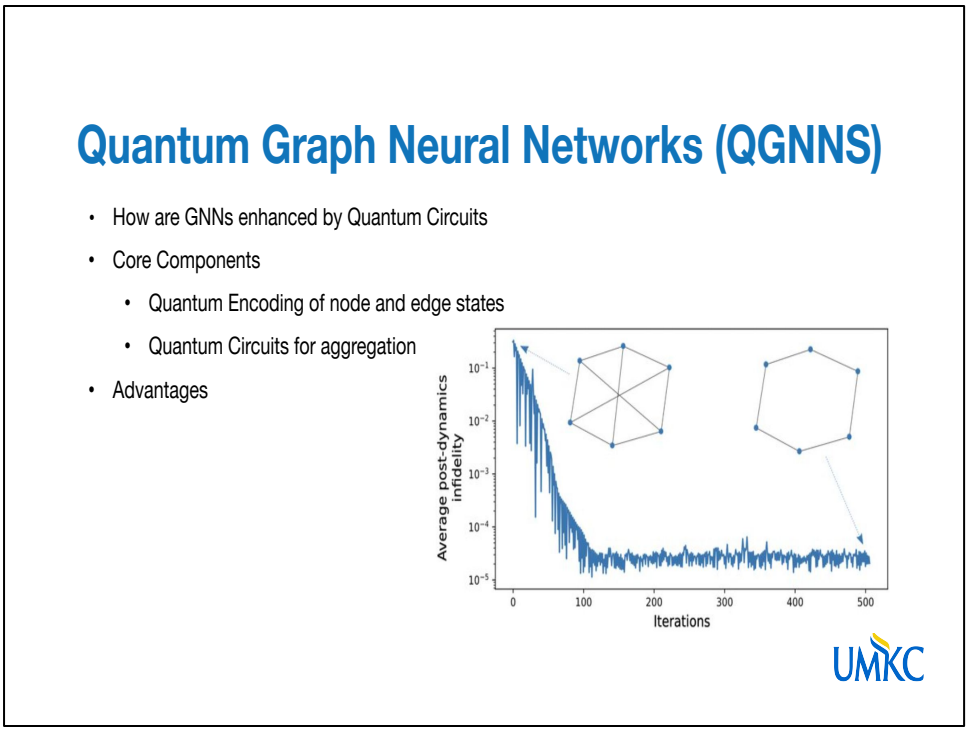
- classically. Quantum algorithms, such as the **Quantum Fourier Transform (QFT)**, enable faster calculations by operating in O(logn)O(\log n)O(logn) time.

Shortest/Optimal Paths

- Quantum algorithms, such as **quantum walks** and **Grover's search**, allow for efficient discovery of shortest paths by amplifying the probability of optimal solutions through amplitude amplification: Uω=I−2∣ω⟩⟨ω∣U_{\omega} = I - 2|\omega\rangle \langle \omega|Uω=I−2∣ω⟩⟨ω∣

Network Flows

- Quantum parallelism facilitates network flow calculations, potentially solving **max-flow** or **min-cost flow problems** more efficiently by using entangled states to evaluate multiple flow configurations simultaneously.



How are GNNs Enhanced by Quantum Circuits

- Quantum circuits allow GNNs to evaluate multiple paths and relationships simultaneously via **quantum superposition** and **entanglement**, thereby increasing the **scalability** and **speed** of GNNs on large, complex graphs.

Core Components

1. **Quantum Encoding**: Maps nodes and edges to quantum states, enabling parallel evaluation.
2. **Quantum Walks**: Explore multiple paths in the graph simultaneously, with amplitude amplification focusing on the most promising routes.
3. **Quantum Circuit Aggregation**: Uses quantum gates for efficient information aggregation from neighboring nodes.

Quantum Encoding of Node and Edge States

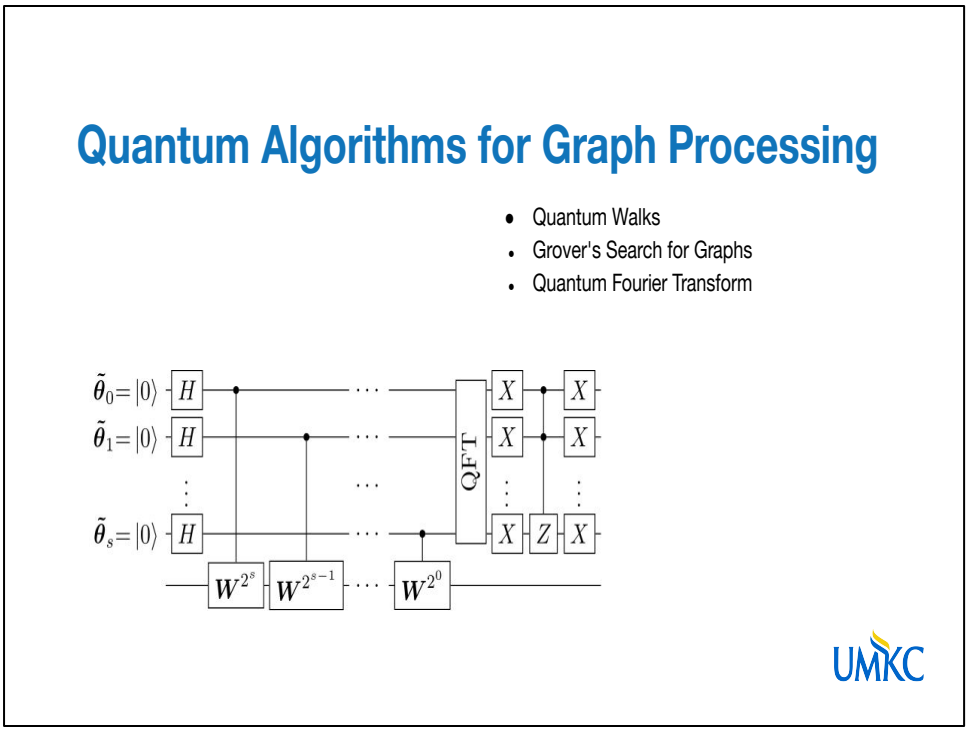
- Node and edge features are encoded into quantum states (qubits), allowing properties like node importance (degree centrality) or edge weights to influence the quantum walk's behavior.

Quantum Circuits for Aggregation

- Quantum circuits replace classical aggregation layers, such as **weighted sums** in GCNs, with quantum operations: U(θ)=e−iθHU(\theta) = e^{-i \theta

Advantages

- **Speed**: Parallel processing of graph components enables faster computation, especially on large-scale graphs.
- **Scalability**: Quantum GNNs can handle larger graphs by leveraging superposition and entanglement, potentially scaling with a reduced qubit count per node/edge pair compared to classical methods.



Quantum Walks

- Quantum walks, the quantum analog of classical random walks, are used for **graph traversal** and **node/edge discovery**. They operate on the probability distribution of states, exponentially accelerating convergence for certain path-finding tasks.

Grover's Search for Graphs

- Grover's algorithm offers a quadratic speedup, reducing search times from O(n)O(n)O(n) to O(n)O(\sqrt{n})O(n). This is particularly useful in finding specific nodes or substructures within large graphs.

Quantum Fourier Transform (QFT)

- The QFT transforms states between computational and Fourier bases, useful in applications like **graph spectrum analysis** where eigenvalues provide insights into graph structure and connectivity. The QFT has complexity O(logn)O(\log n)O(logn), significantly faster than classical FFT for large graphs.

Additional Algorithms

1. **Amplitude Amplification**: Builds on Grover's algorithm to increase the probability of desired outcomes in search and optimization tasks.

- Quantum approaches to MST leverage **quantum walks** and **Grover's search** to find minimum spanning trees more efficiently than classical methods by parallel evaluation of edge weights.

2. **Quantum Hamiltonian Simulation**:
   - Used for simulating physical systems and solving combinatorial optimization problems on graphs by encoding the graph structure into a Hamiltonian and evolving it over time:
3. $U(t)=e^{-iHt}$ U(t) = e^{-i H t}U(t)=e−iHt
   where HHH encodes graph connectivity and edge weights.
4. **Quantum Clustering**:
   - Quantum clustering algorithms use **density matrices** to find clusters within graphs, allowing for fast community detection. Dynamic quantum clustering extends this by identifying clusters that evolve over time, useful in network flow or dynamic social networks.

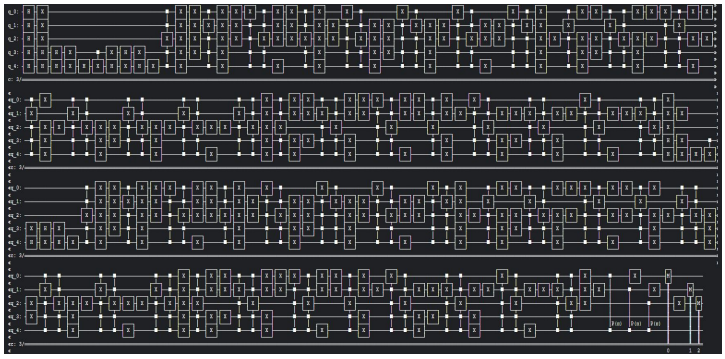
Practical Example: Using Quantum Walks for Disaster Response

- We prepare for a major typhoon to hit the Maldives.
- The major port, Male, loses all capacity.
- What is the most effective method of distributing resources from nearby islands?

UMKC

Quantum Walks

- Encoded qubits denoting demand, capacity, and distance
- Initialized a coin operator, Developed a shift operator, Applied Grover's Search Algorithm



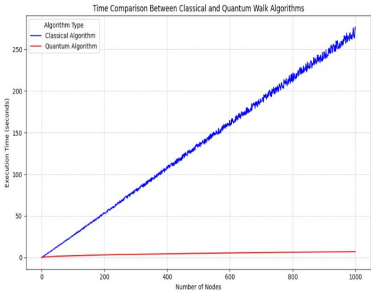
UMKC

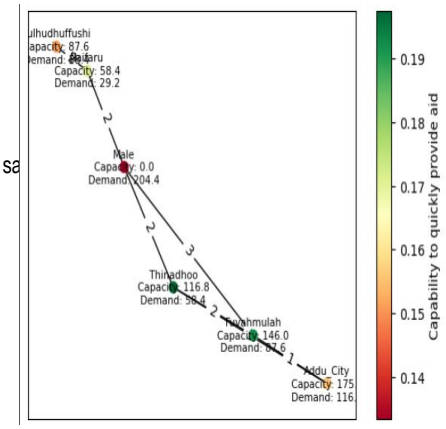
Time Efficiency

Quadratic Speedup.

Time vs best classical algorithms providing sa

level of results.





UMKC

Session 4 Summary

- Introduction to Graph Neural Networks (GNNs)
- Quantum Parallelism for Graph-Structured Data
- Quantum Graph Neural Networks (QGNNs)
- Quantum Algorithms for Graph Processing in QGNNs
- Challenges and Future Directions for QGNNs

UMKC


UMKC