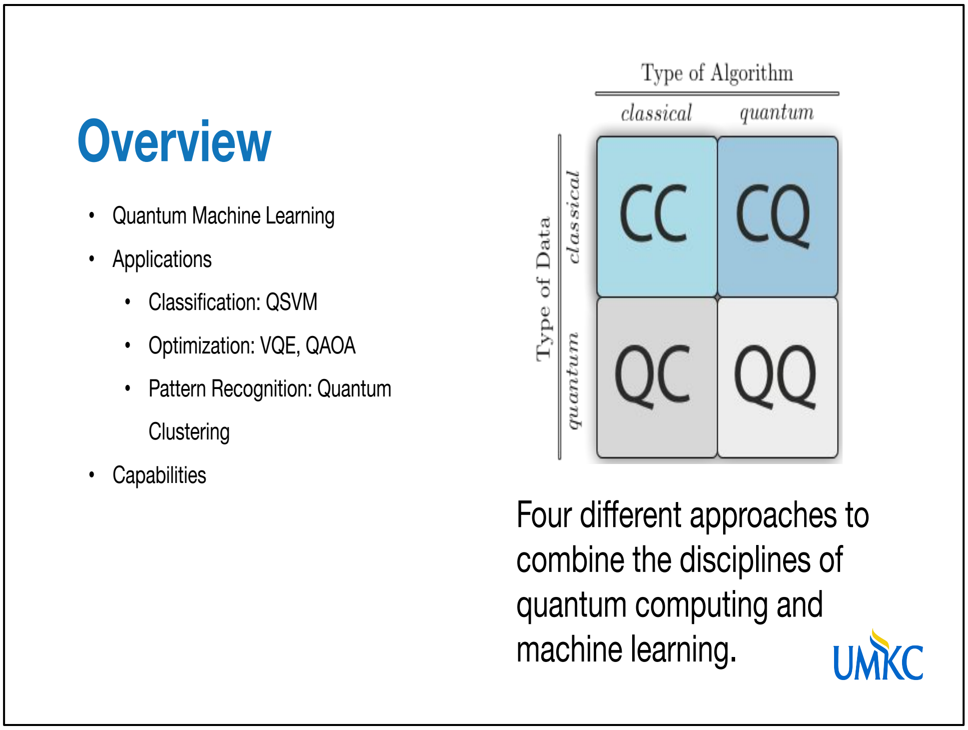
UMKC

Quantum AI Mini Seminar Series:

Session 2: Quantum Machine Learning

- Overview
- Variational Quantum Eigensolver
- Quantum Approximate Optimization Algorithm (QAOA)
- Data Encoding in Quantum Systems
- Challenges in Quantum AI



Definition of QML:

- QML is the integration of **quantum mechanics** into **machine learning models** to solve complex problems with enhanced speed and efficiency.

Applications in ML Tasks:

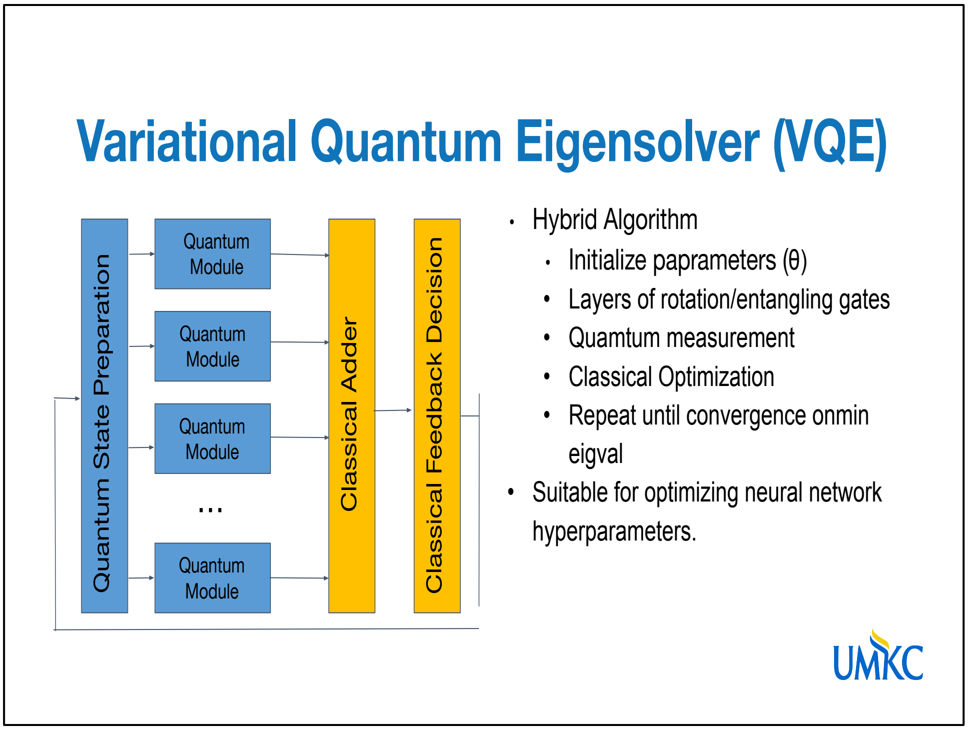
- **Classification**: Quantum Support Vector Machines (QSVM).
- **Optimization**: Algorithms like **VQE** and **QAOA** reduce steps in cost function minimization.
- **Clustering and Pattern Recognition**: Quantum-based clustering shows promise for high-dimensional data analysis.

Potential Impact on Machine Learning:

- **Enhanced processing power** for complex models.
- **Parallelism** allows evaluation of multiple states at once.
- May reduce computation times for **optimization, data processing, and training** tasks.

Session Focus:

- Key QML algorithms: **Variational Quantum Eigensolver (VQE)**, **Quantum Approximate Optimization Algorithm (QAOA)**.
- **Data encoding** methods in quantum systems.
- Current **challenges** limiting quantum computing in AI.



VQE Overview:

- **Hybrid Algorithm**: Combines **quantum processing** and **classical optimization** to solve for the lowest eigenvalue of a Hamiltonian.
- Used in **optimization** problems, analogous to finding the minimum of a **cost function** in AI.

How VQE Works:

1. **Parameterized Quantum Circuit**:
   - Initialize parameters θ={θ1,θ2,…,θn}\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}θ={θ1,θ2,…,θn}.
   - Circuit structure: Layers of **rotation gates** and **entangling gates**.
2. **Quantum Measurement**:
   - Evaluate ⟨ψ(θ)∣H∣ψ(θ)⟩\langle \psi(\theta) | H | \psi(\theta) \rangle⟨ψ(θ)∣H∣ψ(θ)⟩ for cost function.
3. **Classical Optimization**:
   - Adjust parameters θ\thetaθ to minimize the cost function with classical optimizers (e.g., **gradient descent**).
   - Repeat until convergence on **minimum eigenvalue**.

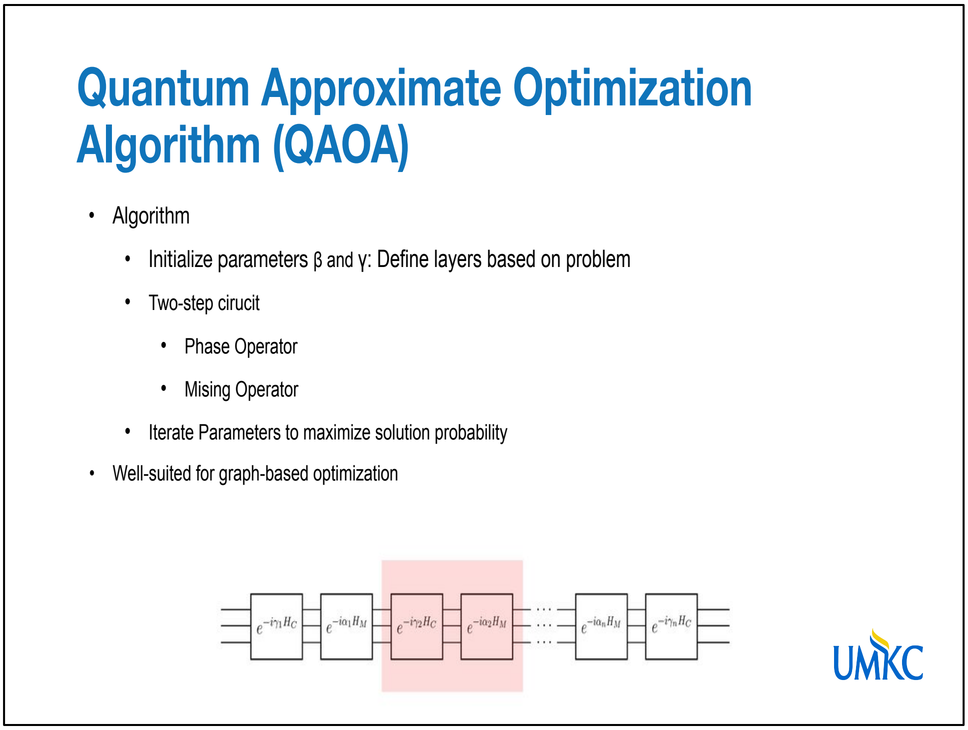
Circuit Example:

- **Circuit Layout**: Alternating layers of **single-qubit rotation gates** and **CNOT gates** for entanglement.
- **Equation for Expectation**: E(θ)=⟨ψ(θ)∣H∣ψ(θ)⟩E(\theta) = \langle \psi(\theta) |

- H | \psi(\theta) \rangleE(θ)=⟨ψ(θ)∣H∣ψ(θ)⟩
- Goal: Minimize E(θ)E(\theta)E(θ) to approximate the ground state energy.

AI Application:

- VQE can be applied to **optimize neural networks** by encoding network parameters in the quantum circuit.
- Uses fewer steps for **cost function minimization** than classical methods, potentially speeding up training times.



QAOA Overview:

- **Objective**: Approximate solutions for **combinatorial optimization problems**.
- Uses a **parameterized quantum circuit** to iteratively adjust solution probabilities.

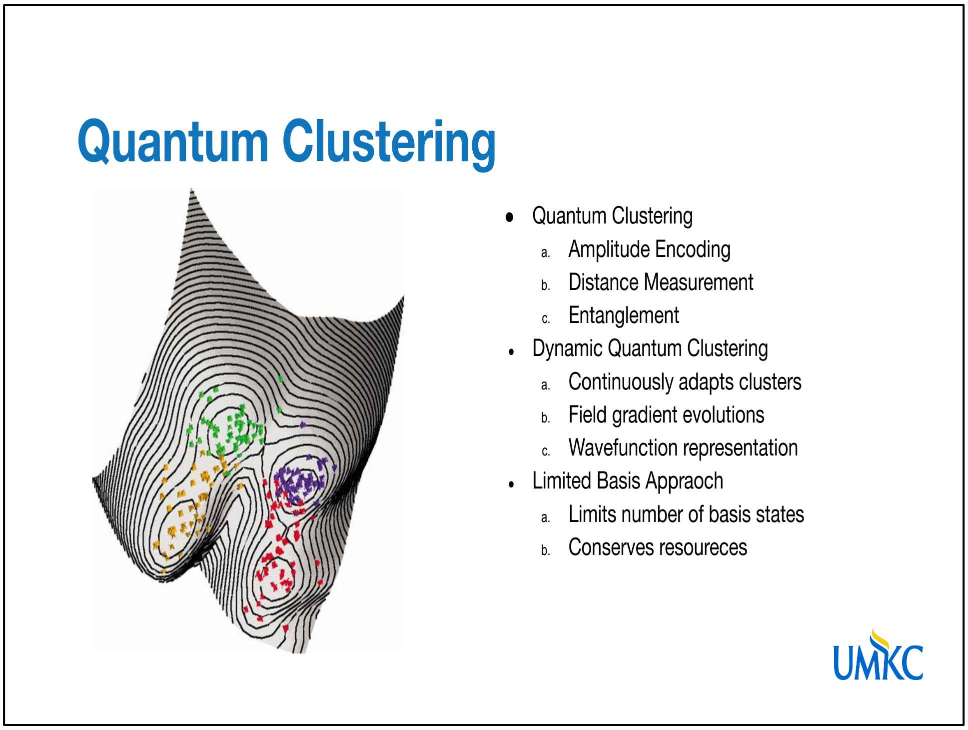
Algorithm Steps:

1. **Initialize Parameters** β\betaβ and γ\gammaγ:
   - Define layers based on the optimization problem.
2. **Two-Step Circuit**:
   - **Phase Operator**: Applies constraints of the problem, rotating by γ\gammaγ.
   - **Mixing Operator**: Rotates state with parameter β\betaβ, exploring other potential solutions.
3. **Iterate Parameters**:
   - Adjust β\betaβ and γ\gammaγ to maximize probability of finding optimal solution.

Circuit Structure:

- **Phase Separation Layer**: Applies **e−iγHe^{-i \gamma H}e−iγH** to encode constraints.
- **Mixing Layer**: Applies **e−iβBe^{-i \beta B}e−iβB** to mix states, encouraging exploration.

- QAOA is well-suited for **graph-based optimization** in AI, including tasks like **graph partitioning**, **scheduling**, and **route optimization**.
- Faster than classical algorithms for problems with complex constraints.



Quantum Clustering Overview:

- Quantum clustering involves mapping data points into **quantum states** to perform clustering with quantum principles like **superposition** and **entanglement**.
- Each data point is represented as a quantum state ∣x⟩|x\rangle∣x⟩ in a high-dimensional **Hilbert space**. Clusters are then identified by measuring the similarities between these states.
- Quantum clustering algorithms are effective for high-dimensional data and **complex clustering problems** where traditional clustering methods are computationally expensive.

How Quantum Clustering Works:

- **Amplitude Encoding**: Encodes each data point as an amplitude vector in a quantum state: ∣x⟩=∑i=0N−1xi∣i⟩|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle∣x⟩ =i=0∑N−1xi∣i⟩
- **Distance Measurement**: The overlap (or fidelity) between quantum states can be used as a measure of similarity. The closer two states are in the Hilbert space, the more likely they belong to the same cluster.
- **Entanglement for Multi-dimensional Relations**: Entanglement can be used to create relationships across multiple features, making clustering based on **multi-dimensional correlations** more effective.

Dynamic Quantum Clustering (DQC):

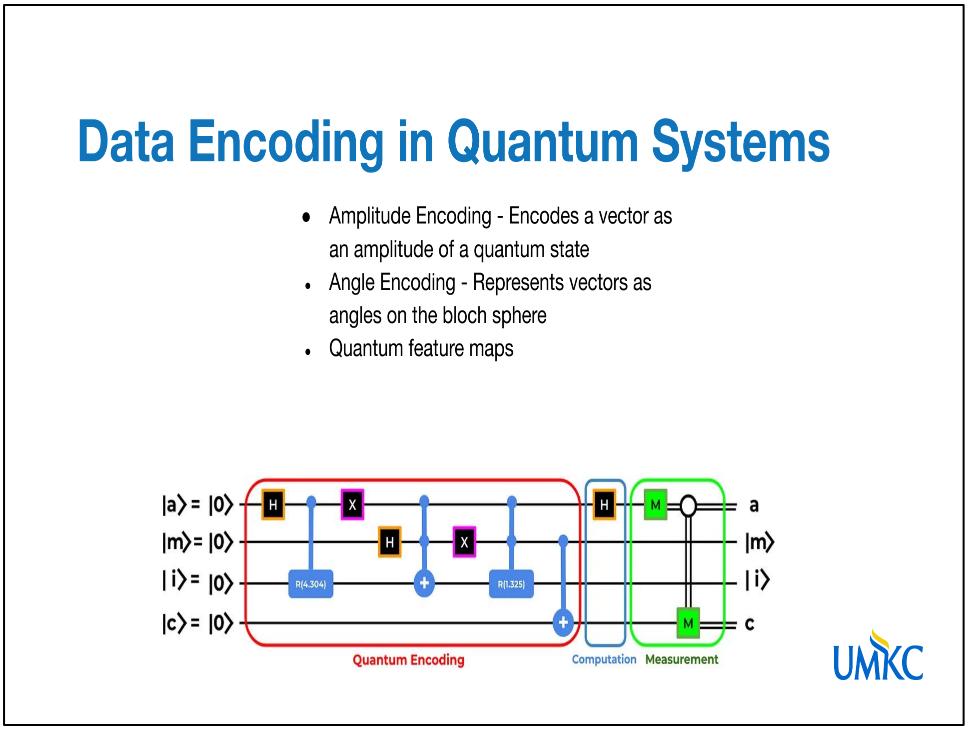
- **Adaptability**: DQC continuously adapts clusters based on **real-time data updates**. It's effective for applications where data points are constantly changing, such as real-time sensor data or financial markets.
- **Wavefunction Evolution**: DQC uses a **Schrödinger equation** approach to simulate data point movements over time in a potential field that represents the cluster structure.
- **Algorithm Mechanics**:
    - Each data point is represented by a **quantum state** in a potential field. The field is defined so that points within the same cluster experience a similar potential.
    - Over time, data points evolve according to the field's gradient, allowing for dynamic adjustments to cluster memberships.
    - The time evolution of each point's wavefunction can be represented by: iℏ∂ψ(x,t)∂t=Hψ(x,t)i \hbar \frac{\partial \psi(x, t)}{\partial t} = H \psi(x, t)iℏ∂t∂ψ(x,t)=Hψ(x,t) where HHH is the Hamiltonian representing the potential landscape of the clusters.

Limited Basis Approaches:

- **Basis Restriction**: Uses a limited number of basis states for data representation, reducing the **computational resources** required.
- In practice, only a subset of possible quantum states is used to approximate clusters. This approach is useful when quantum resources are constrained or when data points are sparse.
- **Example Application**: For simple binary clustering, a limited basis could reduce the space to two primary states, representing distinct classes or clusters.

Applications in AI:

- **Dynamic Environments**: DQC is particularly suited for clustering in environments where data changes frequently (e.g., financial trading, dynamic sensor networks).
- **Large-Scale Data**: Limited basis approaches make it feasible to apply quantum clustering techniques to **large datasets** by focusing on key basis states.
- **Pattern Recognition**: Quantum clustering methods, especially with dynamic and limited basis approaches, excel in **pattern recognition** and **anomaly detection** in high-dimensional datasets, where traditional methods are less efficient.



Data Encoding Challenge:

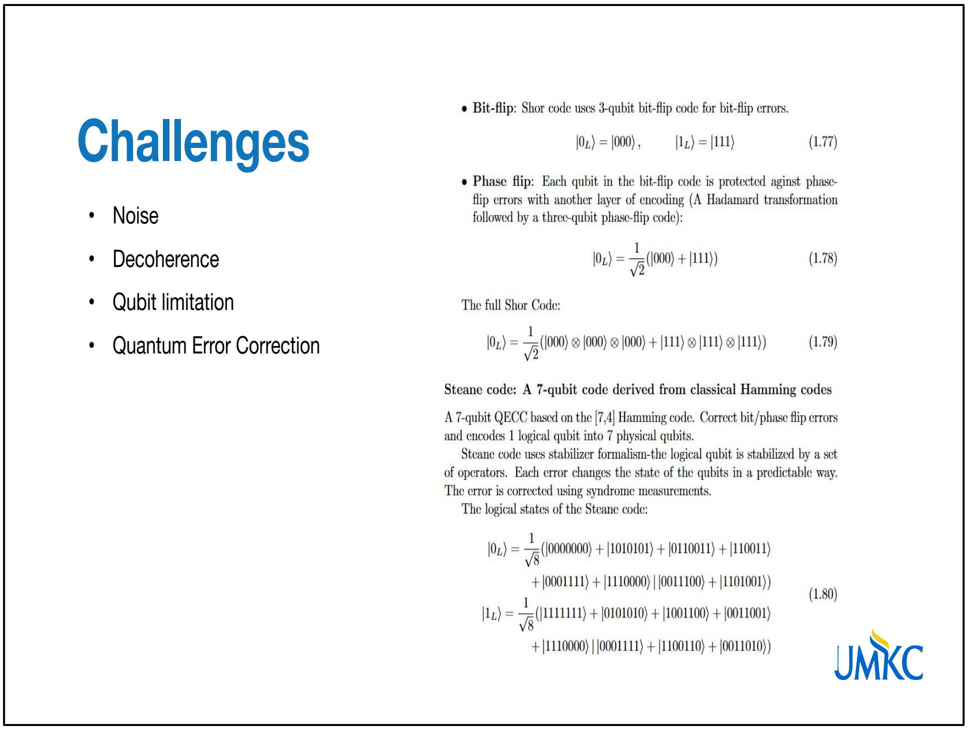
- Quantum systems require classical data to be encoded as **quantum states** for QML algorithms to process.

Common Encoding Methods:

- **Amplitude Encoding**:
  - Encodes a vector of data into the amplitudes of a quantum state.
  - Example: A vector x=[x0,x1,x2,…,xN−1]x = [x_0, x_1, x_2, \ldots, x_{N-1}]x=[x0,x1,x2,…,xN−1] is mapped to: ∣x⟩=∑i=0N−1xi∣i⟩|x\rangle = \sum_{i=0}^{N-1} x_i |i\rangle∣x⟩=i=0∑N−1xi∣i⟩
- **Angle Encoding**:
  - Represents data points as angles on the Bloch sphere for each qubit.
  - Data point xix_ixi is mapped as: ∣x⟩=cos(xi)∣0⟩+sin(xi)∣1⟩|x\rangle = \cos(x_i) |0\rangle + \sin(x_i) |1\rangle∣x⟩=cos(xi)∣0⟩+sin(xi)∣1⟩

Encoding Example in ML:

- **Quantum Feature Map**: Each feature in a dataset can be mapped onto a qubit state using amplitude or angle encoding, allowing the **quantum model** to operate on it directly.
- In clustering tasks, for example, encoding high-dimensional features into qubit states can enable rapid, complex pattern recognition.



Noise and Decoherence:

- **Noise**: Quantum gates are sensitive to external interference, leading to errors in calculations.
- **Decoherence**: Quantum states lose information over time due to interaction with the environment.
- Effect: Reduces the accuracy and reliability of quantum computations, limiting circuit depth and runtime.

Scalability:

- **Qubit Limitations**: Current quantum computers have few qubits, restricting the size of data and models they can handle.
- **Quantum Error Correction**: Required to build larger, fault-tolerant quantum systems, but significantly increases hardware requirements.
- **Implications for AI**: Limited qubit numbers and noisy environments make it challenging to apply quantum algorithms at scale for AI tasks involving large datasets.

Current Research Directions:

- **Hardware Improvements**: Developing more stable qubits to reduce noise.
- **Error Correction Techniques**: Quantum error correction codes that increase qubit fidelity.
- **NISQ Era (Noisy Intermediate-Scale Quantum)**: Emphasis on algorithms designed for today’s limited-capacity quantum systems, like hybrid approaches

- with classical preprocessing.

Session 3 Summary

- Grover's, QFT, QAOA, VQE, and QSVM applications in AI
- Speedups in search, optimization, signal processing, and classification
- Quantum Walks, and QVSM

Key Concepts Recap:

- **Quantum Machine Learning (QML)**: Application of quantum principles in ML.
- **VQE**: Quantum algorithm for **cost function minimization**.
- **QAOA**: Quantum algorithm for **combinatorial optimization**.
- **Data Encoding**: Methods to represent classical data as quantum states.

Next Session Preview:

- **Quantum Graph Neural Networks (QGNNs)**: Focusing on applying quantum computing to graph-based AI tasks, commonly seen in social networks, biological systems, and recommendation engines.

UMKC