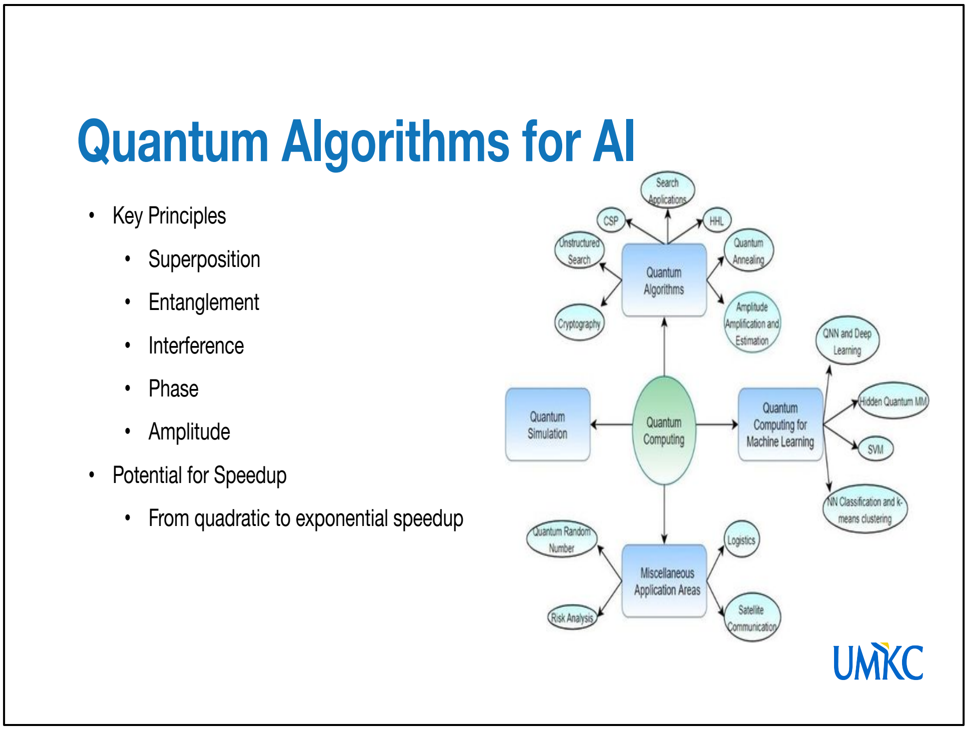
Quantum AI Mini Seminar Series:

Session 2: Quantum Algorithms and their Potential for AI

- Quantum Algorithms for AI
- Grover's Search Algorithm
- Quantum Fourier Transform
- Quantum Speedup
- Other Quantum Algorithms
- AI application of Quantum Algorithms



Key Principles

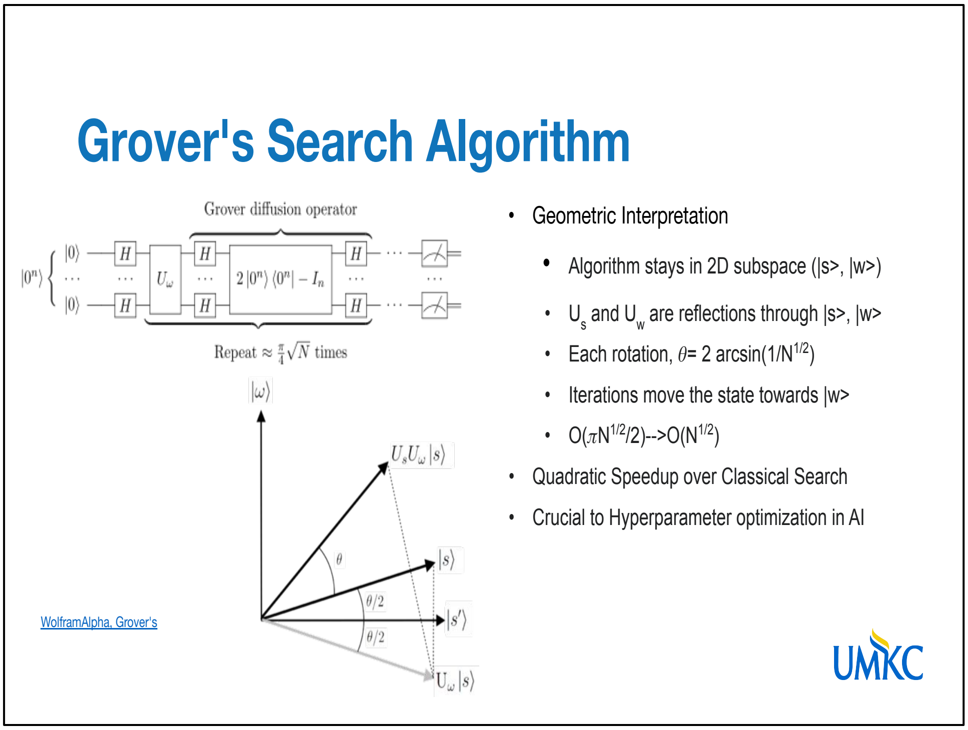
- **Superposition**
  - **Description**: In classical computing, bits can be in one of two definite states, 0 or 1. In quantum computing, **qubits** can exist in a combination of both states simultaneously, known as **superposition**.
  - **Mathematics**: A qubit in superposition is represented as: ∣ψ⟩=α∣0⟩+β∣1⟩|\psi\rangle = \alpha |0\rangle + \beta |1\rangle∣ψ⟩=α∣0⟩+β∣1⟩ where α\alphaα and β\betaβ are complex probability amplitudes, with the condition that ∣α∣2+∣β∣2=1|\alpha|^2 + |\beta|^2 = 1∣α∣2+∣β∣2=1. The qubit is in a linear combination of the 0 and 1 states, and when measured, it collapses to either 0 or 1 based on these probabilities.
  - **Relevance**: Superposition enables quantum algorithms to **explore multiple possibilities** in parallel, drastically increasing computational efficiency for certain tasks, such as searching or optimization in AI.
- **Entanglement**
  - **Description**: **Entanglement** is a unique quantum phenomenon where the state of one qubit becomes dependent on the state of another, regardless of the distance between them.
  - **Mathematics**: If two qubits are entangled, the state of the system is described by a joint wavefunction: ∣ψ⟩=12(∣00⟩+∣11⟩)|\psi\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle \right)∣ψ⟩=21(∣00⟩+∣11⟩)

    - Measuring one qubit immediately determines the state of the other, no matter how far apart they are. This is not possible in classical systems.
    - **Relevance**: Entanglement allows for **correlations** across qubits that classical bits cannot achieve, enabling faster information sharing and **quantum parallelism**. It is key in quantum communication and some quantum algorithms, such as Grover's and Shor's.
- **Interference**
    - **Description**: In quantum systems, **interference** occurs when quantum states interact, either amplifying or canceling each other out. This is analogous to how waves interfere.
    - **Mathematics**: Quantum interference is governed by the amplitudes of the wavefunction. Constructive interference increases the probability of measuring correct solutions, while destructive interference reduces the probability of incorrect solutions.
    - **Relevance**: Interference is crucial in quantum algorithms. For instance, in **Grover's algorithm**, interference is used to **amplify the probability** of finding the correct solution and suppress incorrect ones, leading to a faster search process.
- **Phase**
    - **Description**: **Phase** is a critical property in quantum mechanics, representing the angle of the wavefunction. It is not just the magnitude of the quantum state that matters but also its phase relative to other states.
    - **Mathematics**: The phase of a quantum state can be written as eiθe^{i\theta}eiθ, where θ\thetaθ represents the phase angle. The total wavefunction is: ∣ψ⟩=αeiθ0∣0⟩+βeiθ1∣1⟩|\psi\rangle = \alpha e^{i\theta_0} |0\rangle + \beta e^{i\theta_1} |1\rangle∣ψ⟩=αeiθ0∣0⟩+βeiθ1∣1⟩ Phase differences between qubits influence how they interfere with one another.
    - **Relevance**: **Phase manipulation** is crucial for algorithms like the **Quantum Fourier Transform (QFT)** and **Quantum Phase Estimation**, which are used in AI for **optimization** and **signal processing** tasks.
- **Amplitude**
    - **Description**: **Amplitude** in quantum mechanics represents the likelihood of a quantum state. When squared, it gives the **probability** of measuring a particular state.
    - **Mathematics**: For a state ∣ψ⟩=α∣0⟩+β∣1⟩|\psi\rangle = \alpha |0\rangle + \beta |1\rangle∣ψ⟩=α∣0⟩+β∣1⟩, the probability of measuring ∣0⟩|0\rangle∣0⟩ is ∣α∣2|\alpha|^2∣α∣2, and the probability of measuring

- ∣1⟩|1\rangle∣1⟩ is ∣β∣2|\beta|^2∣β∣2.
- **Relevance**: Quantum algorithms often aim to **increase the amplitude** of desired states through **amplitude amplification**. For example, in Grover's algorithm, the amplitude of the correct solution grows with each iteration, improving the probability of success.

Potential for Speedup

- **Quadratic Speedup**:
  - **Description**: Quantum algorithms can offer quadratic speedup compared to classical algorithms. A prime example is **Grover's algorithm**, which reduces the time complexity of an unstructured search from O(N)O(N)O(N) in classical systems to O(N)O(\sqrt{N})O(N) in quantum systems.
  - **Relevance in AI**: This quadratic speedup is particularly useful in AI tasks like **hyperparameter tuning**, **combinatorial search**, and **data mining**, where large solution spaces need to be explored efficiently.
- **Exponential Speedup**:
  - **Description**: Some quantum algorithms, such as **Shor's algorithm** for integer factorization, offer **exponential speedup** over classical counterparts. Shor's algorithm runs in O((logN)3)O((\log N)^3)O((logN)3) time, compared to the best-known classical algorithm, which runs in **super-polynomial time**.
  - **Relevance in AI**: Exponential speedups can dramatically impact AI areas that require solving problems like **cryptography**, **optimization**, and **machine learning** model training. For example, tasks that would take millions of years to solve classically could be completed in seconds on a sufficiently powerful quantum computer.
  - **Quantum Machine Learning (QML)**: **Quantum Speedup** also extends to potential **exponential improvements** in specific machine learning tasks, such as **quantum-enhanced support vector machines (QSVM)** and **quantum neural networks**, which could outperform their classical counterparts by learning and processing data more efficiently.



Classical Search Problem:

- In a classical search, finding a specific element in an **unstructured database** of NNN elements requires an average of O(N)O(N)O(N) queries.
- This is because, without any structure, the best classical algorithm can only **check one element at a time**. This linear search requires, on average, N/2N/2N/2 queries to find the target element.

Grover's Quadratic Speedup:

- Grover's algorithm reduces the search complexity to **O(√N)** by exploiting **quantum superposition** and **amplitude amplification**. This is a **quadratic speedup** over classical search algorithms, which is significant when NNN is large.
- Grover's algorithm starts with a uniform superposition of all possible states: ∣s⟩=1N∑x=0N−1∣x⟩|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle∣s⟩=N1x=0∑N−1∣x⟩
    1. In this state, each element in the database is equally likely to be the solution.
- The algorithm uses two key quantum operations:
    1. **Oracle UωU_{\omega}Uω**: Marks the correct solution ∣ω⟩|\omega\rangle∣ω⟩ by flipping its phase. It is essentially a reflection around the hyperplane orthogonal to the solution state.
    2. **Diffusion Operator UsU_sUs**: Reflects the current state about the

1. average amplitude of all states, amplifying the amplitude of the correct solution ∣ω⟩|\omega\rangle∣ω⟩.

Geometric Interpretation:

- Grover's algorithm can be visualized geometrically in a 2D subspace spanned by ∣s⟩|s\rangle∣s⟩ (the initial state) and ∣ω⟩|\omega\rangle∣ω⟩ (the solution state).
- Each iteration of Grover's algorithm **rotates the quantum state** toward ∣ω⟩|\omega\rangle∣ω⟩ by an angle θ\thetaθ, where: θ=2arcsin(1N)\theta = 2 \arcsin \left(\frac{1}{\sqrt{N}} \right)θ=2arcsin(N1)
- The number of iterations rrr required to reach the solution is approximately r≈π4Nr \approx \frac{\pi}{4} \sqrt{N}r≈4πN. After this, the quantum state is close enough to ∣ω⟩|\omega\rangle∣ω⟩ that measuring the state yields the correct solution with high probability.

Quantum Operators in Grover's Algorithm:

- The **oracle** UωU_{\omega}Uω is a reflection operator: Uω=I−2∣ω⟩⟨ω∣U_{\omega} = I - 2 |\omega\rangle \langle \omega|Uω=I−2∣ω⟩⟨ω∣
  - This operation flips the phase of the state ∣ω⟩|\omega\rangle∣ω⟩, marking it as the correct solution.
- The **diffusion operator** UsU_sUs reflects around the average amplitude: Us=2∣s⟩⟨s∣−IU_s = 2 |s\rangle \langle s| - IUs=2∣s⟩⟨s∣−I
  - This increases the amplitude of the state ∣ω⟩|\omega\rangle∣ω⟩ and decreases the amplitude of the other states.

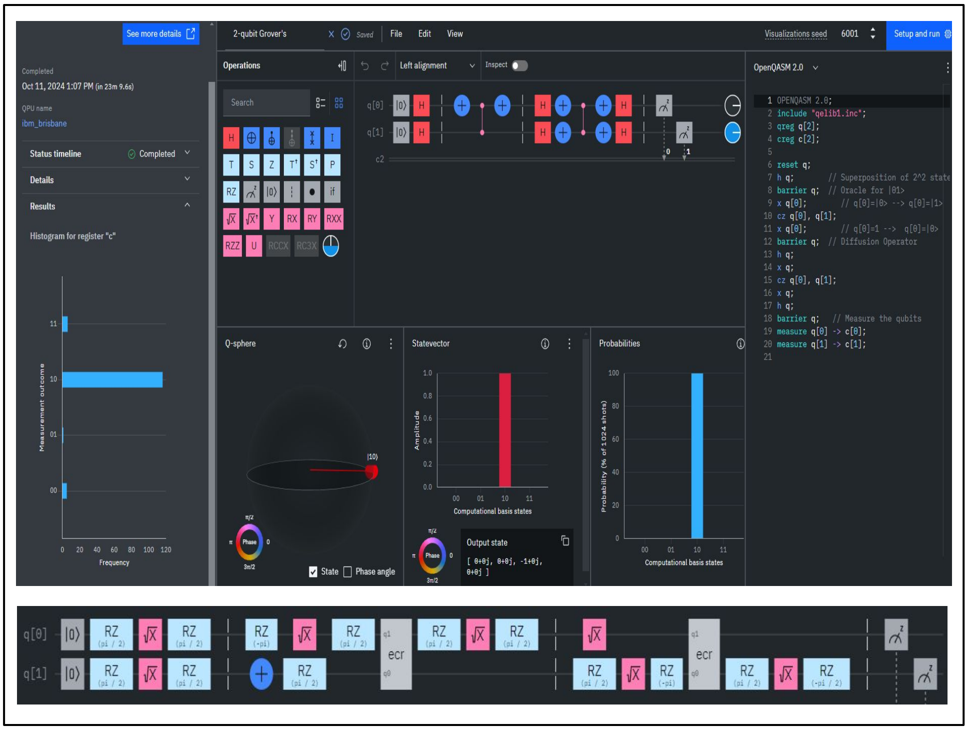
Application in AI:

- In AI, Grover's algorithm can be used to **speed up search tasks** that are computationally expensive, such as **hyperparameter tuning** in machine learning models or **combinatorial search problems**.
- For instance, in a large search space of potential hyperparameters, Grover's algorithm can reduce the number of evaluations needed to find the optimal set of parameters.
- Similarly, it can be applied in **data mining** or **pattern recognition** tasks, where searching through a large, unstructured dataset is required.

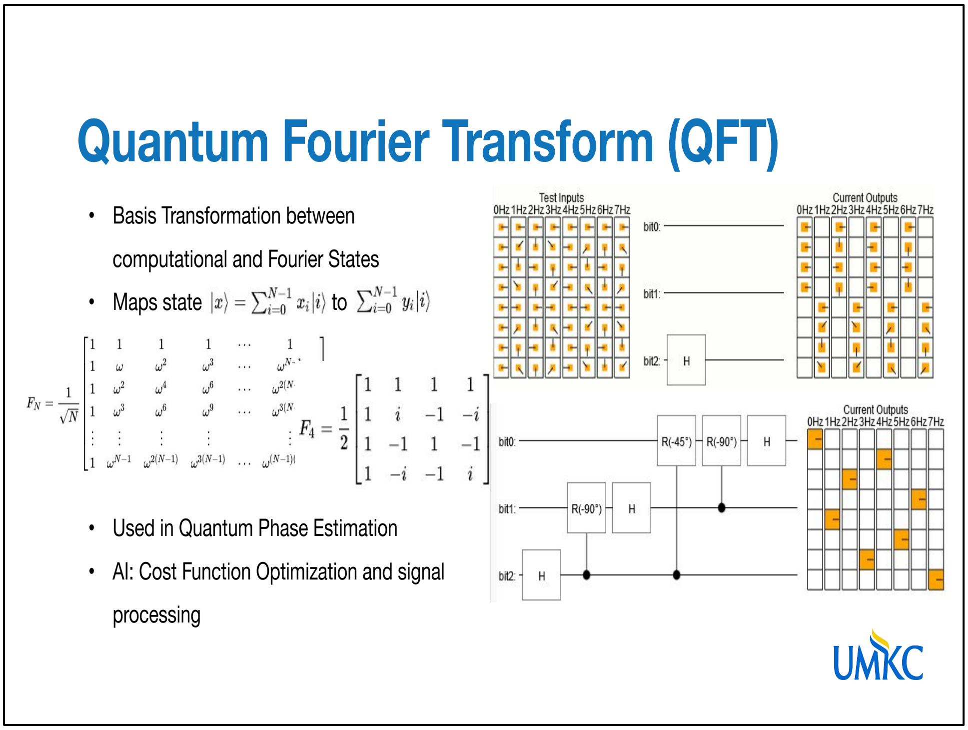
Probability of Success:

- After rrr iterations, the state vector is rotated close to ∣ω⟩|\omega\rangle∣ω⟩, and the probability of measuring the correct answer is maximized.
- The probability of success is given by: sin2((r+12)θ)\sin^2 \left(\left(r + \frac{1}{2} \right) \theta \right)sin2((r+21)θ)
- To maximize the probability of success, the number of iterations should be

- While Grover's algorithm provides a quadratic speedup, it is not exponential. It is optimal for **unstructured search** problems but does not outperform classical algorithms in problems that have inherent structure.
- However, in cases where **brute-force search** is the only viable classical approach, Grover's algorithm can provide a **significant advantage**, especially for large NNN.
-



See more details
Completed
Oct 11, 2024 1:07 PM (in 23m 9.6s)
QPU name
ibm_brisbane
Status timeline
Completed
Details
Results
Histogram for register "c"
Measurement outcome
Frequency
2-qubit Grover's
Saved
File
Edit
View
Visualizations seed
6001
Setup and run
Operations
Left alignment
Inspect
Search
OpenQASM 2.0
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
creg c[2];
reset q;
h q;   // Superposition of 2^2 state
barrier q; // Oracle for |01>
x q[0];   // q[0]=|0> --> q[0]=|1>
cz q[0], q[1];
x q[0];   // q[0]=1 --> q[0]=|0>
barrier q; // Diffusion Operator
h q;
x q;
cz q[0], q[1];
x q;
h q;
barrier q;   // Measure the qubits
measure q[0] -> c[0];
measure q[1] -> c[1];
Q-sphere
Statevector
Amplitude
Computational basis states
Output state
[0+0j, 0+0j, -1+0j, 0+0j]
State
Phase angle
Probabilities
Probability (% of 1024 shots)
Computational basis states



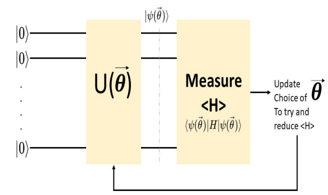
Speaker's Notes:

- **Basis Transformation**: The **Quantum Fourier Transform (QFT)** is a quantum analog of the classical Fourier transform. It transforms quantum states from the **computational basis** (where qubits are represented by binary numbers) to the **Fourier basis**, which is useful in many quantum algorithms. The key feature of QFT is that it maps a quantum state ∣x⟩|x\rangle∣x⟩ to a superposition of states in the Fourier basis.
- **Mapping States**:
  - The QFT maps a state ∣x⟩|x\rangle∣x⟩ into a superposition of Fourier basis states as follows: ∣x⟩→1N∑k=0N−1e2πixk/N∣k⟩|x\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ixk/N} |k\rangle∣x⟩→N1k=0∑N−1e2πixk/N∣k⟩
  - This transformation is pivotal in algorithms where **periodicity** or **phase information** is important. You can show this with a **visualization of the state mapping**, showing how the computational state is spread across the Fourier basis after the transformation.
- **Quantum Phase Estimation**:
  - One of the most important applications of the QFT is in **Quantum Phase Estimation**, an algorithm that determines the eigenvalue (phase) associated with an eigenvector of a unitary operator. The QFT is used at the final step of this algorithm to extract the phase

    - information from the quantum state.
    - **Show an equation** that highlights how QFT is used to measure the phase: $|\psi\rangle = \sum_{k=0}^{N-1} e^{2\pi i\theta k}|k\rangle$
    - In **AI**, phase estimation can assist in solving optimization problems where periodicity or phase information is needed, especially in systems that model complex interactions or cost functions.
- **AI Applications**:
    - The QFT can be applied in **cost function optimization** in machine learning algorithms. For instance, in certain AI models where the cost function is periodic or non-linear, the QFT can help in identifying the optimal parameters by transforming the problem into the frequency domain.
    - Another application is in **signal processing** for AI systems. For example, AI models that deal with time-series data, audio analysis, or sensor data could use QFT to filter noise or extract relevant signal frequencies more efficiently than classical Fourier transforms.

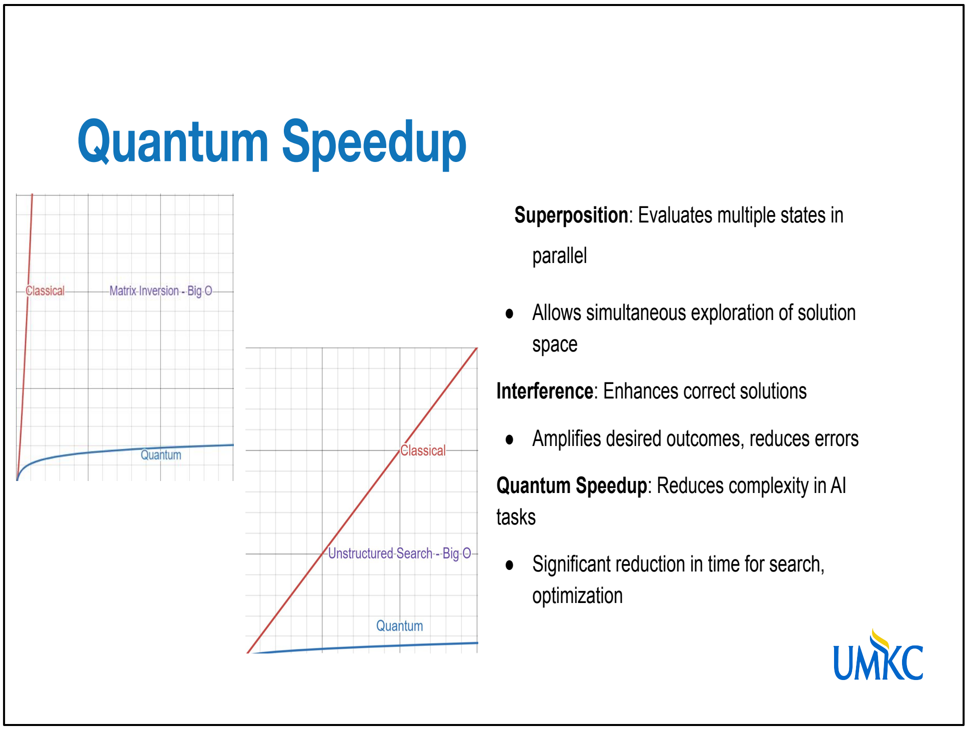
More Quantum Algorithms for AI

- Grover's algorithm: Speedup in search and optimization
- QFT: Enhances signal processing and pattern recognition
- QAOA: Combinatorial optimization for logistics, scheduling
- VQE: Non-convex optimization for AI models
- Quantum Walks: Faster graph traversal and node ranking
- QSVM: Speedup in classification and machine learning tasks



UMKC

Grover's algorithm is highly effective for reducing the time required for search and optimization problems in AI, which are common in model training and data mining.
QFT improves **signal processing** tasks in AI, particularly in systems that rely on time-series analysis or pattern recognition.
QAOA (Quantum Approximate Optimization Algorithm) addresses **combinatorial optimization problems** seen in logistics or scheduling, providing faster solutions than classical methods.
VQE (Variational Quantum Eigensolver) is useful for **non-convex optimization**, applicable to AI models that require minimizing complex loss functions.
Quantum Walks provide faster methods for **graph traversal**, essential for AI applications that work with network-based data like social graphs or recommendation systems.
QSVM (Quantum Support Vector Machine) enhances traditional SVMs by using quantum kernels to accelerate **classification tasks**.



Superposition: Quantum computers leverage **superposition**, where qubits can exist in a combination of states (0 and 1), enabling the system to evaluate many possible solutions **in parallel**. This allows for a more efficient exploration of the **solution space** compared to classical algorithms.

Interference: Through **quantum interference**, the algorithm amplifies the probabilities of **correct solutions** while suppressing incorrect ones. This constructive and destructive interference is a key advantage of quantum systems over classical systems, which don't have this mechanism to accelerate convergence toward correct answers.

Quantum Speedup: The combined effect of superposition and interference leads to **speedups** in various AI tasks, especially in **optimization** and **search problems**. For example, Grover's algorithm achieves quadratic speedup ($O(\sqrt{N})$) in search tasks, while other algorithms like QAOA and VQE optimize large solution spaces more efficiently than classical methods

Session 2 Summary

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

UMKC

UMKC