Introduction to Quantum Machine Learning

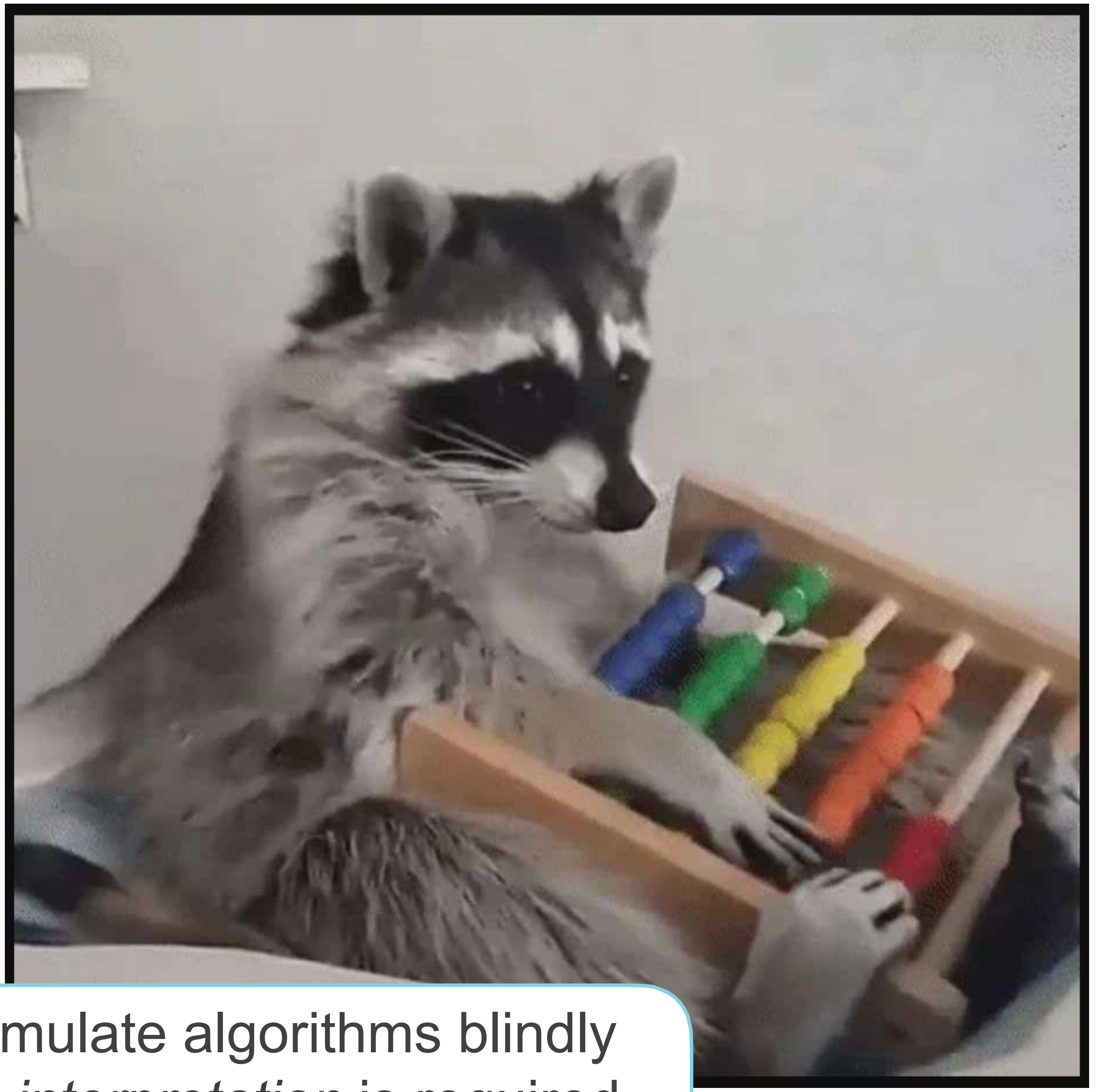
Krishna Pratap Singh, PhD

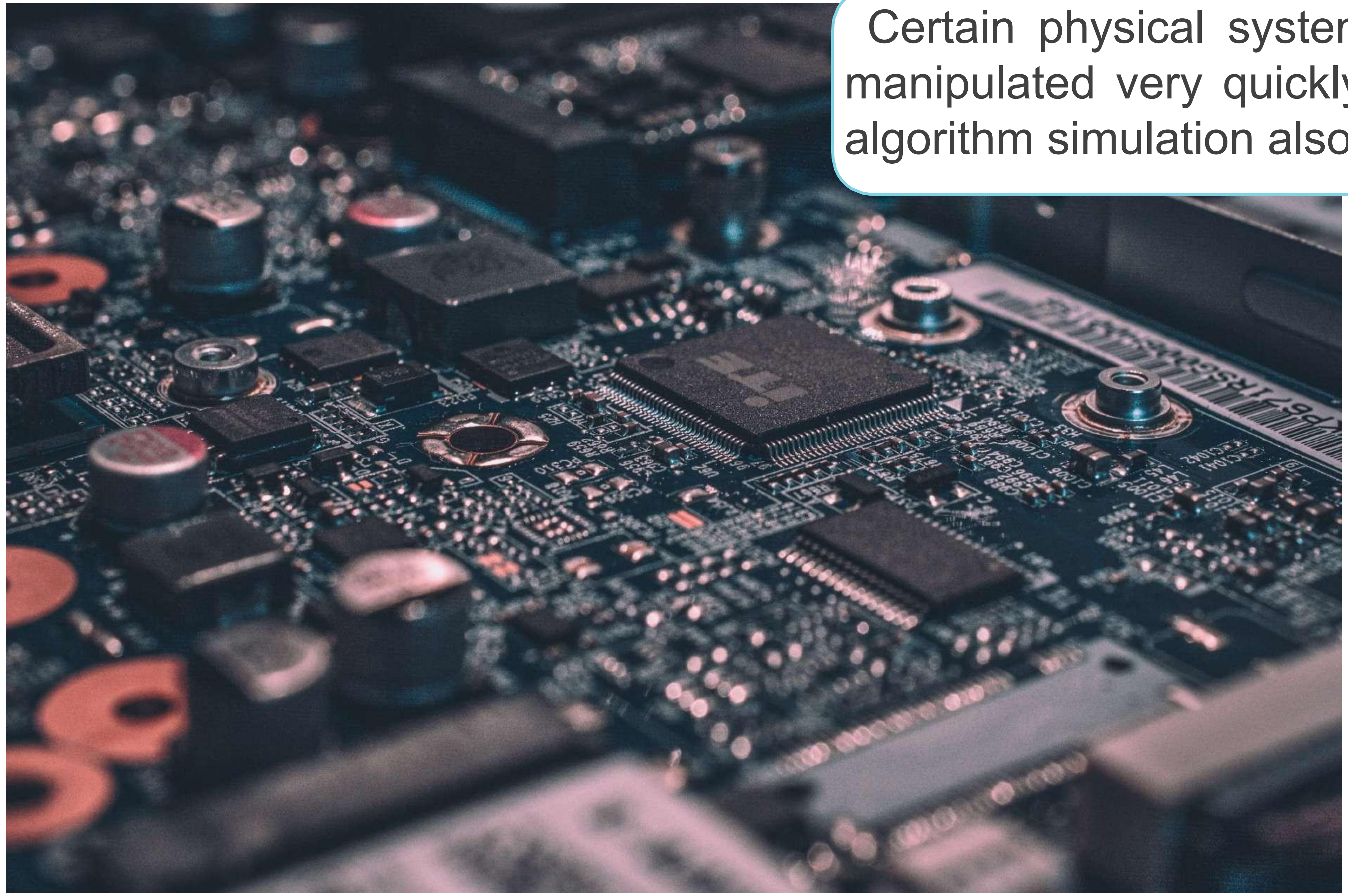
Overview

- Computing… what is it?
- Quantum computing… what is it?
- Quantum machine learning
- Quantum genetic Algorithms (QAG)

What is computing?

- First, what is computing? One perspective - it is ***physical simulation of algorithms coupled to interpretation***. We manipulate a physical system according to rules. A metaphysical tower of concepts then allows us to *interpret* the results.

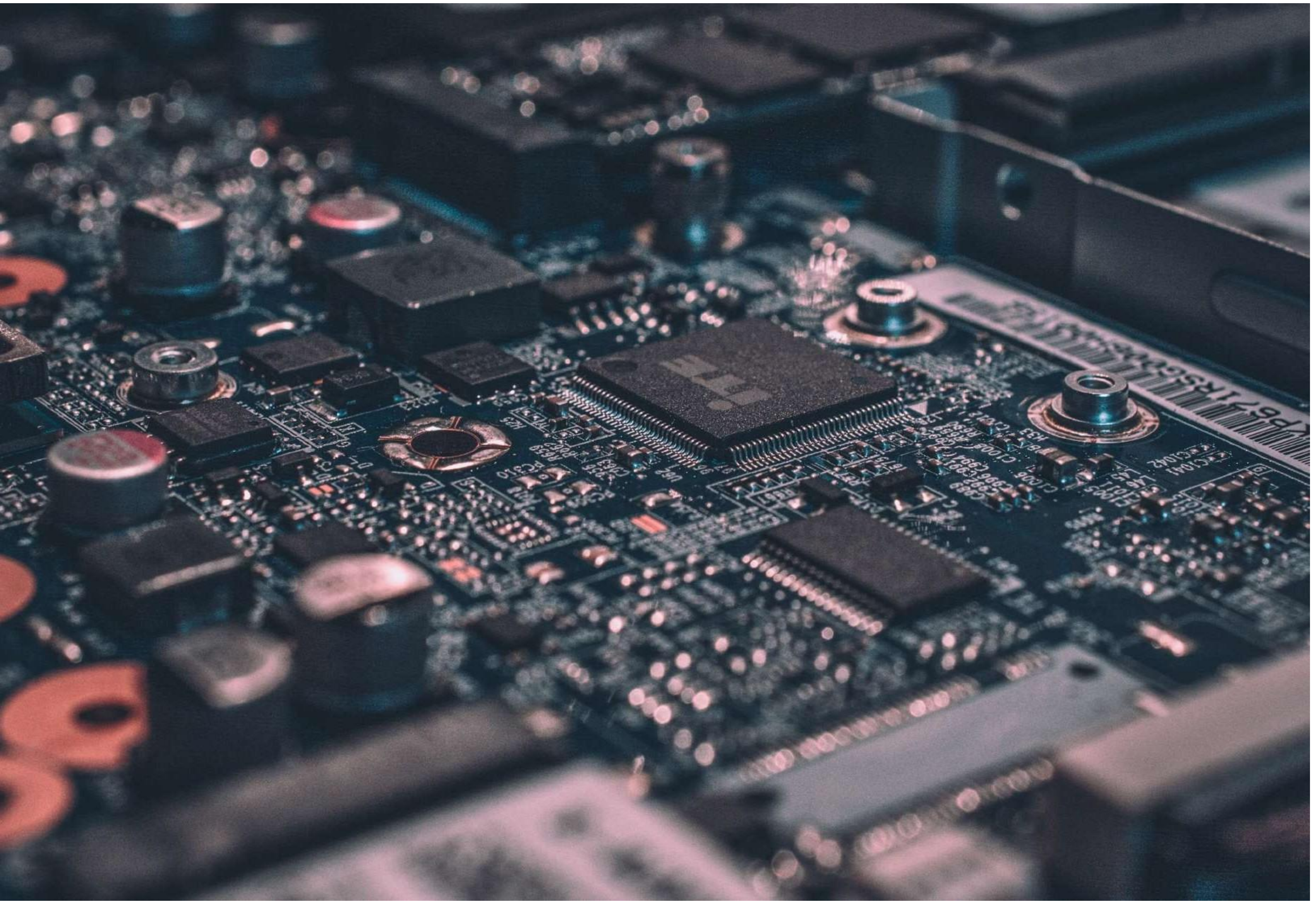




Certain physical systems can be manipulated very quickly - making algorithm simulation also very fast.

We can simulate algorithms blindly - ultimately *interpretation* is required.

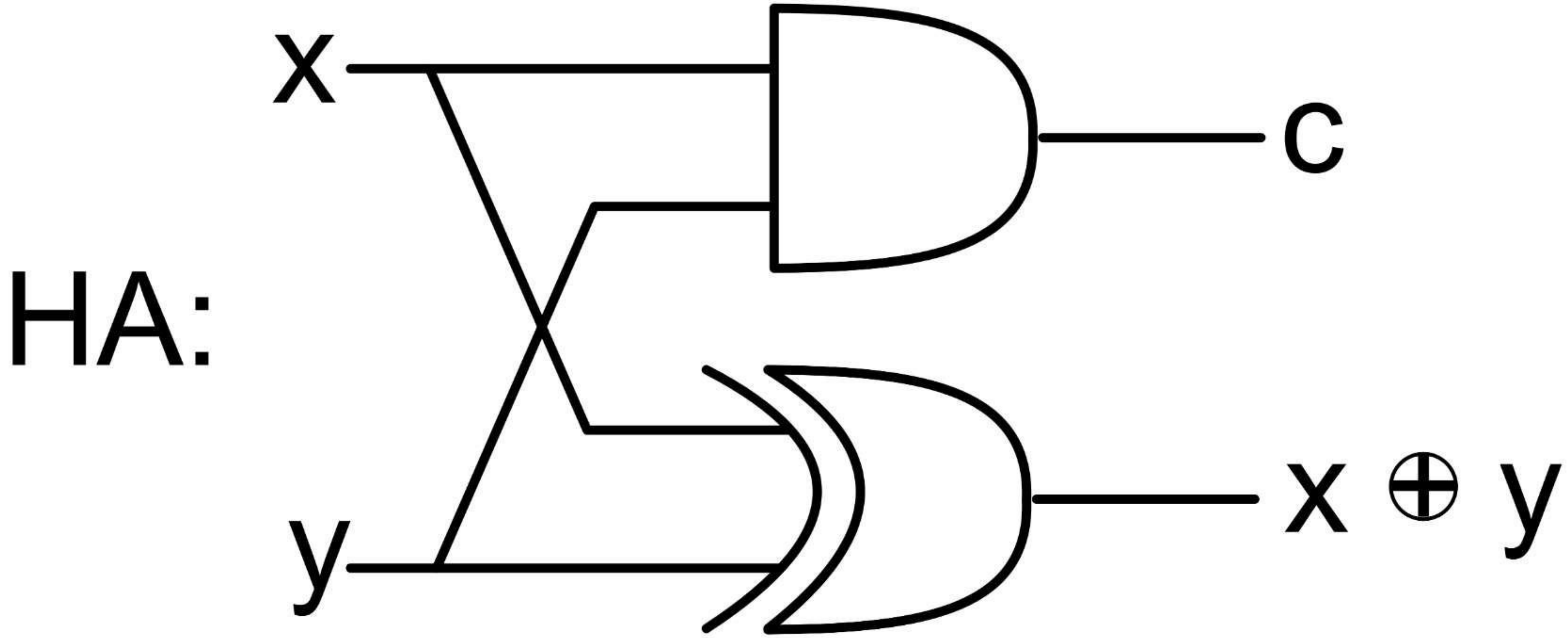
What is classical computing?

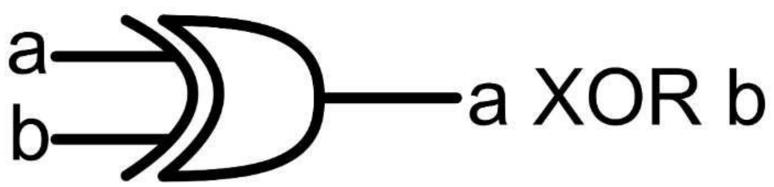


You can see how you would implement a table like this one with logic gates:

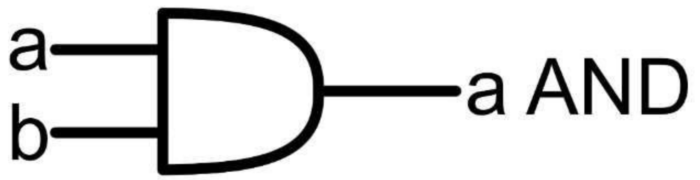
```
 0     0     1     1
+0    +1    +0    +1
---   ---   ---   ---
00    01    01    10
```

You need two inputs and two outputs. This function is called a *Half Adder*:





	b = 0	b = 1
a = 0	0	1
a = 1	1	0



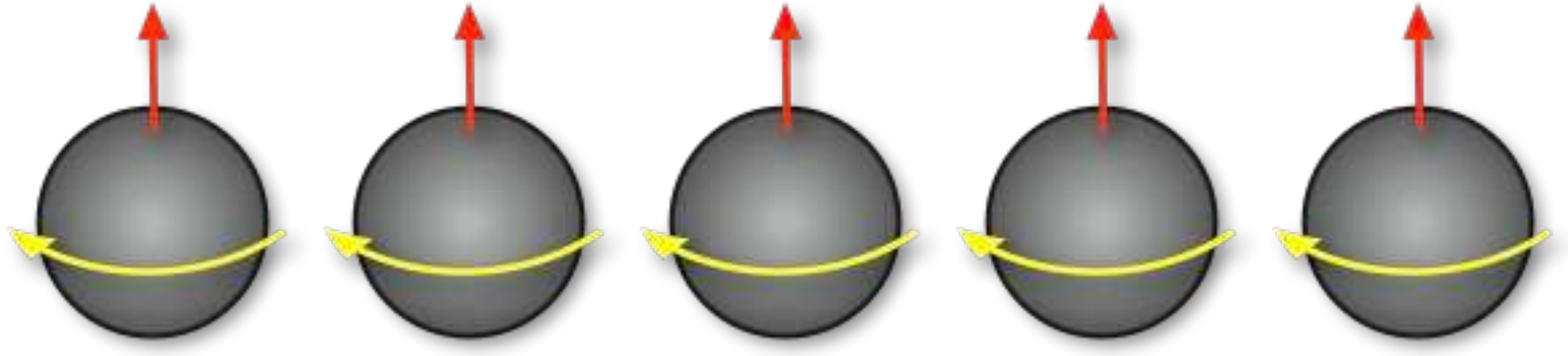
	b = 0	b = 1
a = 0	0	0
a = 1	0	1

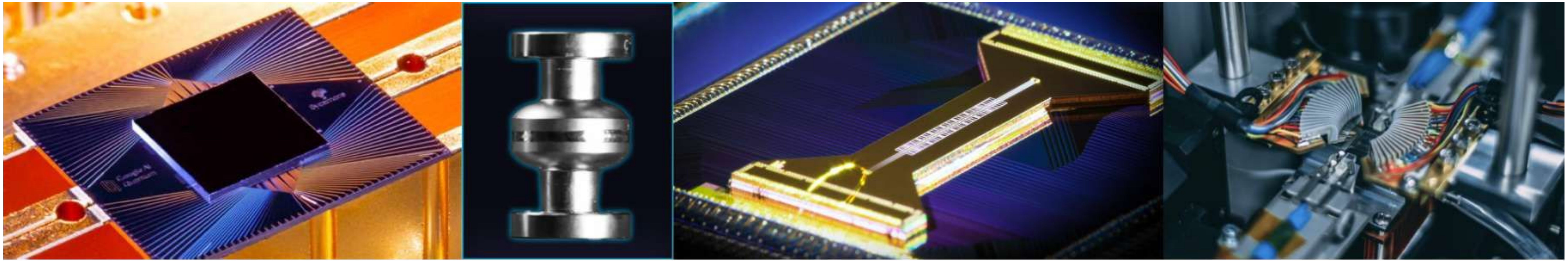
```
 0     0     1     1      x
+0    +1    +0    +1     +y
---   ---   ---   ---   ---
00    01    01    10   c(x⊕y)
```

What is *quantum* computing?

- Quantum computing is using quantum systems to simulate our algorithms.
- Challenges are rooted in the fact that quantum systems are *delicate*. And algorithms are non-obvious.
- Multiple, “competing” platforms for quantum computation exist. The ultimate goals are *scale* and *quantum error correction*.

There are many ways to leverage quantum systems to simulate an algorithm. Features of quantum measurement mean the calculations are probabilistic.



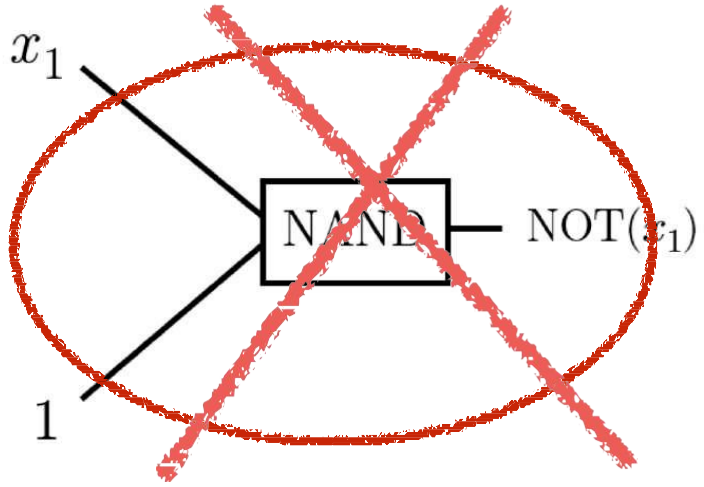


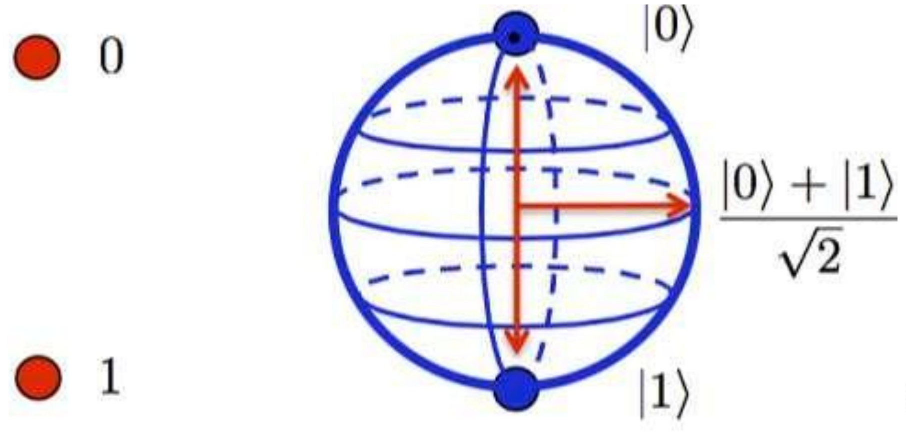
https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.htm l https://sqms.fnal.gov/research/

https://www.honeywell.com/en-us/company/quantu m https://www.xanadu.ai/hardware

What *is* quantum computing?

- At heart, quantum computing is unitary evolution of quantum states.
- It is distinguished by the following features:
  - Entanglement
  - Unitary evolution
  - Superposition of states
  - Reversible computation
  - Probabilistic computation
  - Exponential Hilbert spaces
  - Challenges with state coherence





Classical Bit | Qubit

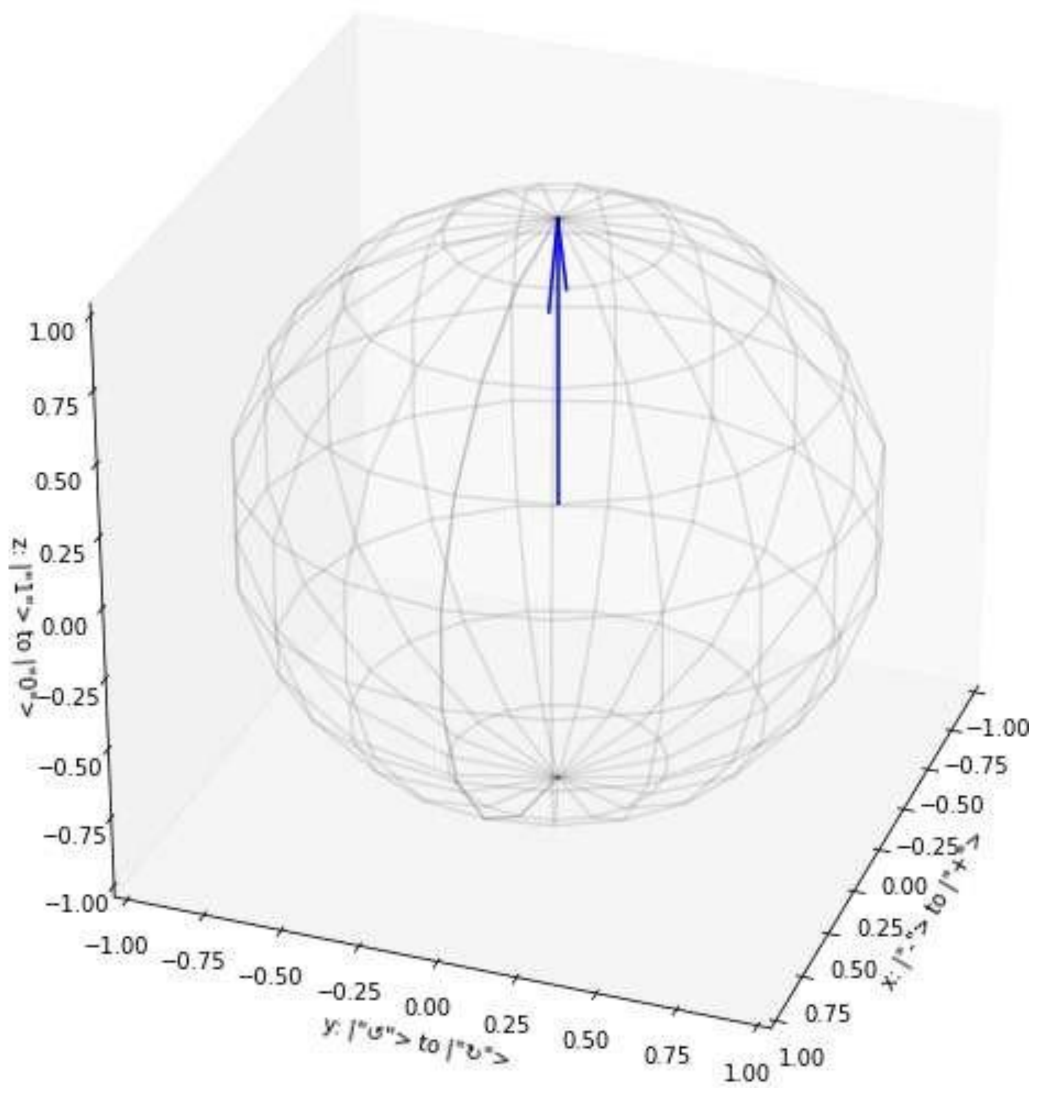
Quantum computing power* scales exponentially with qubits
N bits can exactly simulate log N qubits

This compute unit....	 Commodore 64	 AWS M4 Instance 1 Million x Commodore 64	 Entire Global Cloud 1 Billion x (1 Million x Commodore 64)
can exactly simulate:	**10 Qubits**	**30 Qubits**	**60 Qubits**

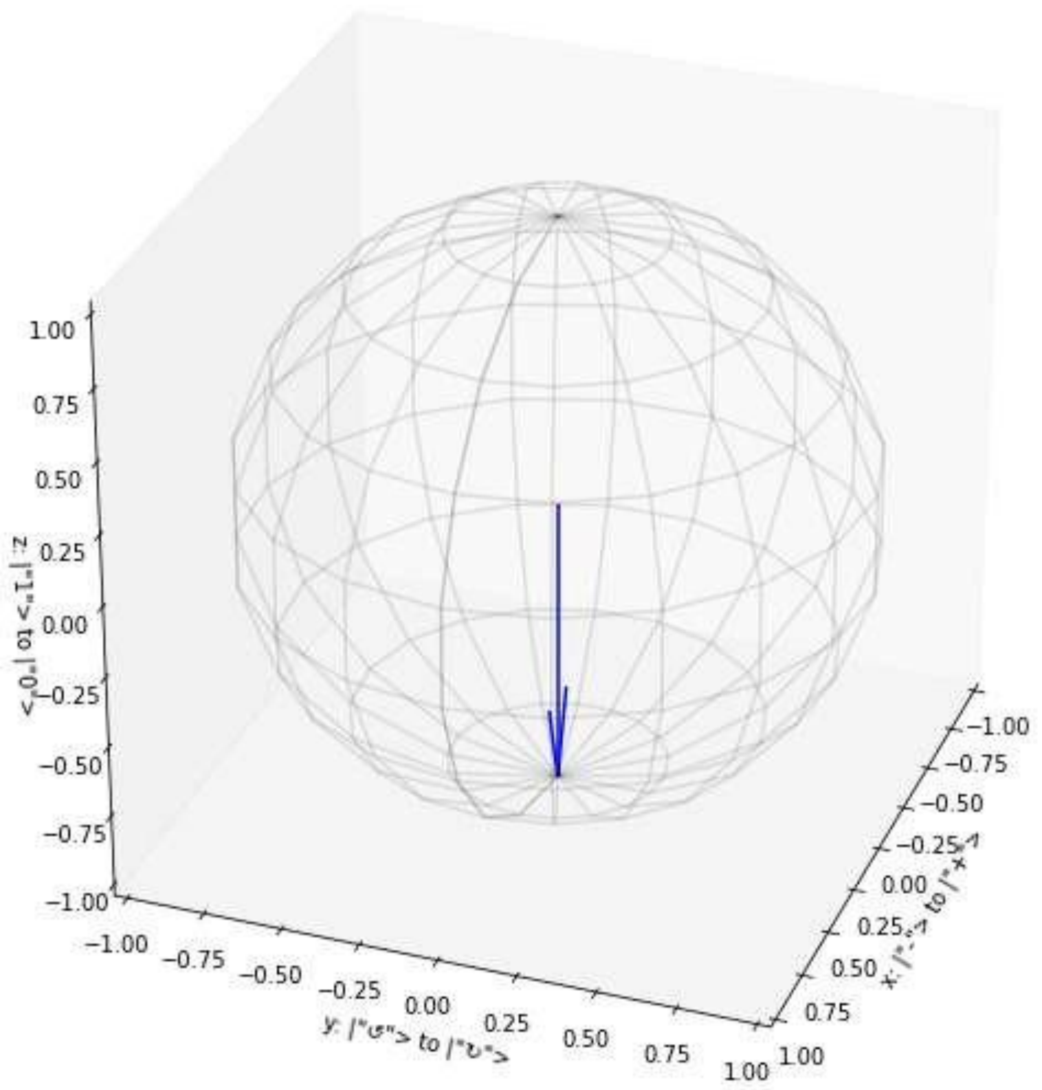
Qubits

$$|\psi\rangle = \begin{pmatrix}\alpha \\ \beta\end{pmatrix} = \alpha \times \begin{pmatrix}1 \\ 0\end{pmatrix} + \beta \times \begin{pmatrix}0 \\ 1\end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle$$

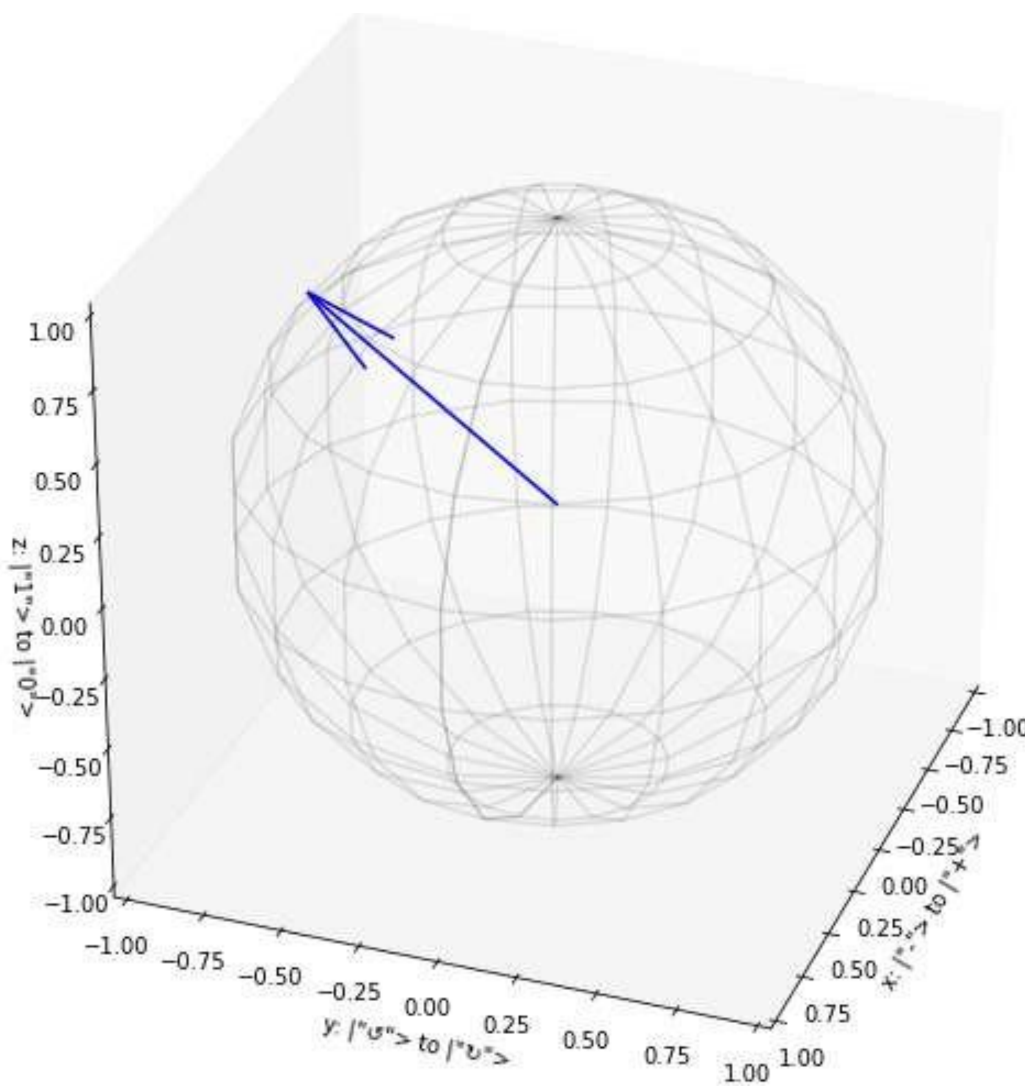
Quantum operators rotate the vector's direction.



$|0\rangle$



$|1\rangle$



$\sqrt{0.2}|0\rangle + \sqrt{0.8}|1\rangle$

What *is* quantum computing?

$$|\psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} = \alpha \times \begin{pmatrix}1\\0\end{pmatrix} + \beta \times \begin{pmatrix}0\\1\end{pmatrix} \equiv \alpha|0\rangle + \beta|1\rangle$$

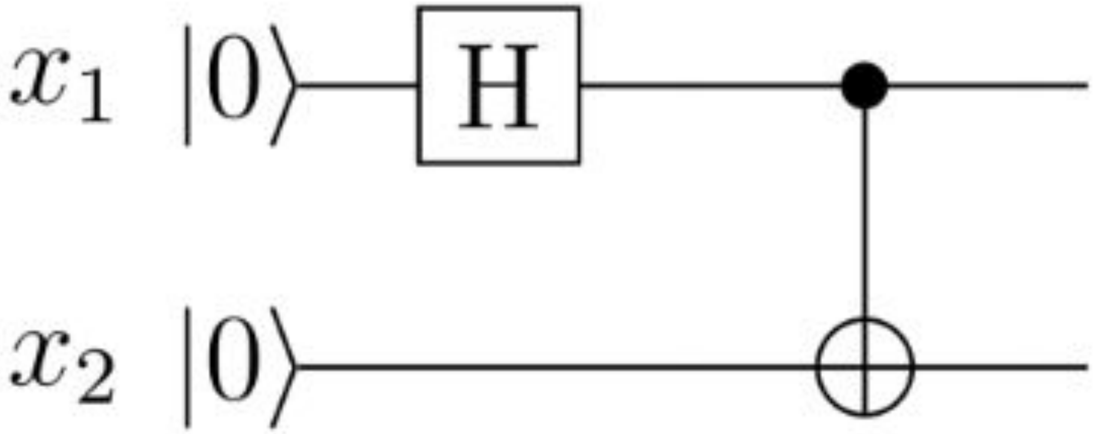
$$|0\rangle|0\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}1\begin{pmatrix}1\\0\end{pmatrix}\\ 0\begin{pmatrix}1\\0\end{pmatrix}\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} = |00\rangle$$

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \qquad X = \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$$

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle$$

Superposition!



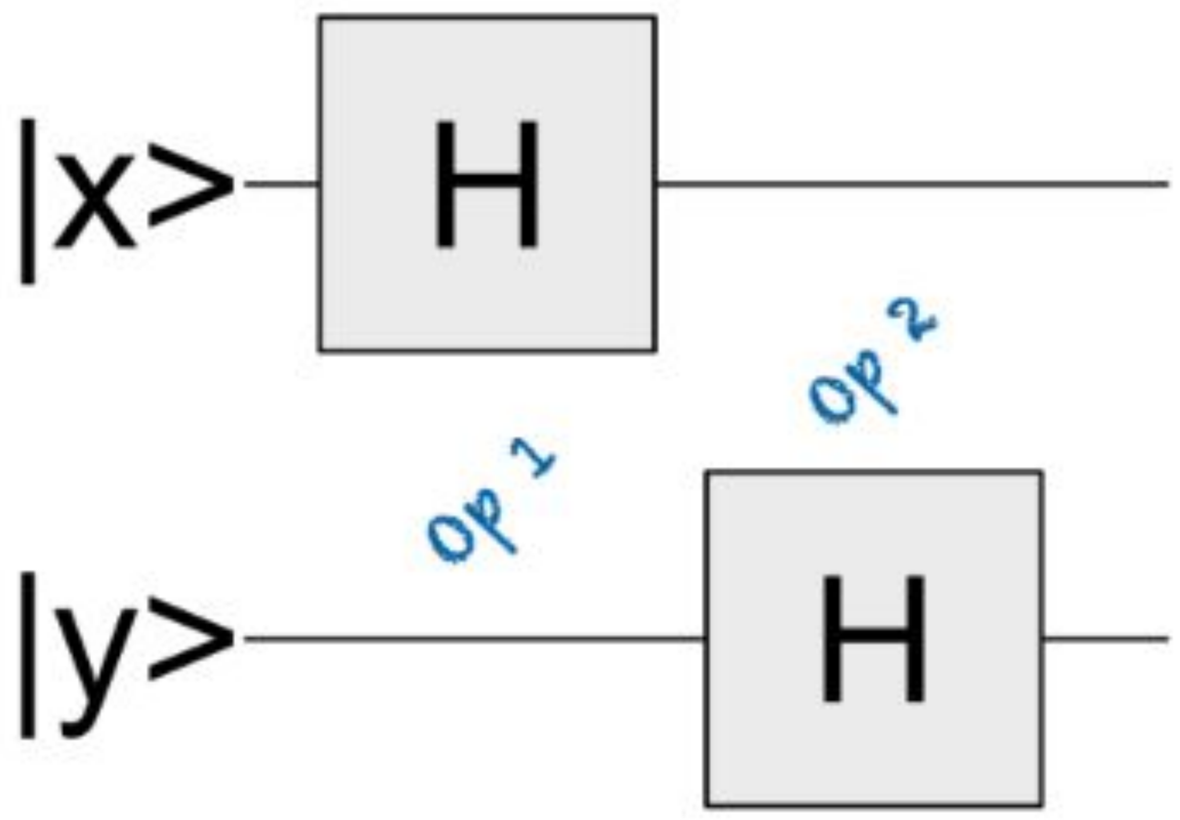
$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{pmatrix} \quad \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{pmatrix}$$

Entanglement!

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\left(\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Circuit composer: https://quantum-computing.ibm.com/

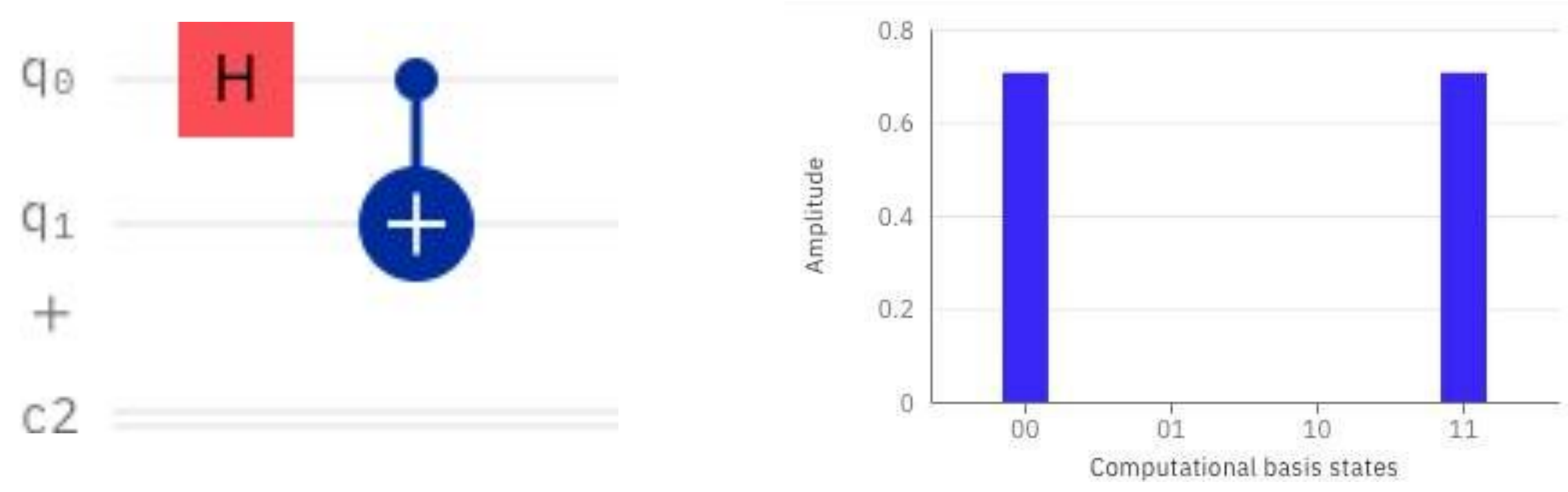
What *is* quantum computing?

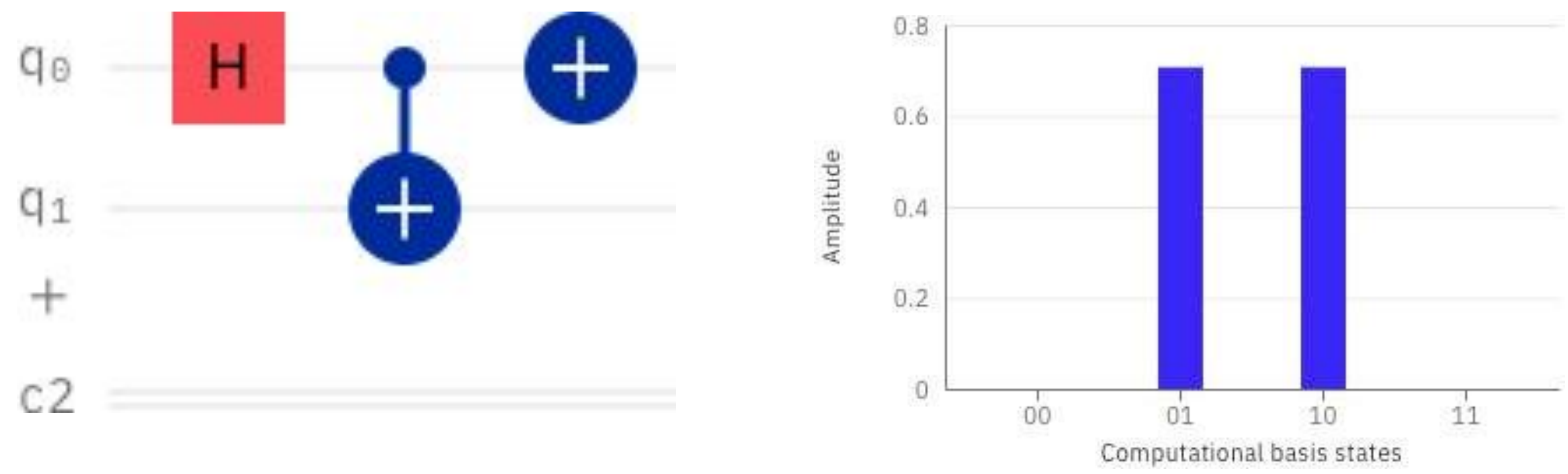


$$H \otimes I \qquad\qquad I \otimes H$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1&0&1&0\\0&1&0&1\\1&0&-1&0\\0&1&0&-1\end{pmatrix}\frac{1}{\sqrt{2}}\begin{pmatrix}1&1&0&0\\1&-1&0&0\\0&0&1&1\\0&0&1&-1\end{pmatrix}$$

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix} \rightarrow \frac{1}{2}\begin{pmatrix}1\\1\\1\\1\end{pmatrix} = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

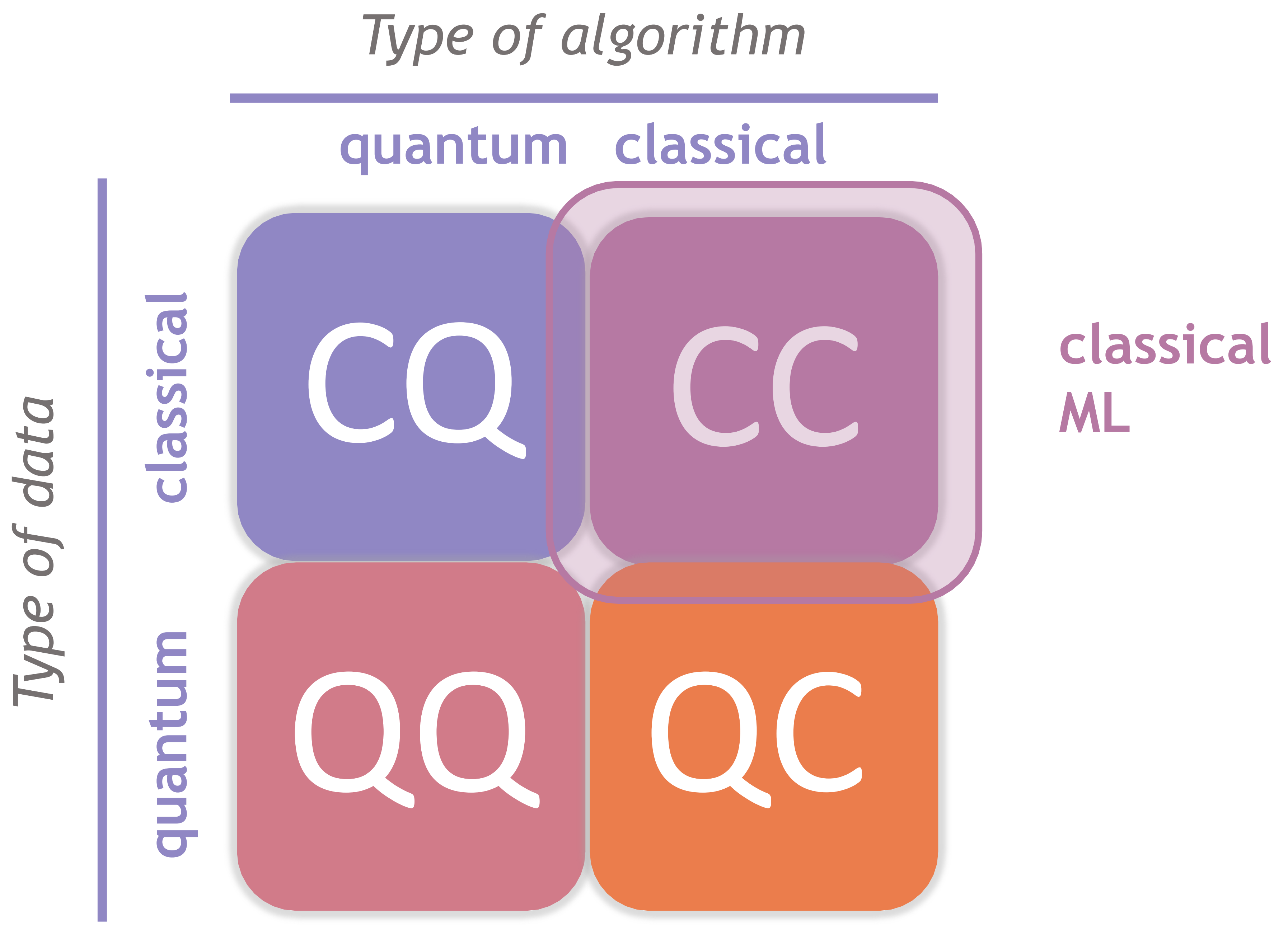




Super hand-wavy “quantum advantages”

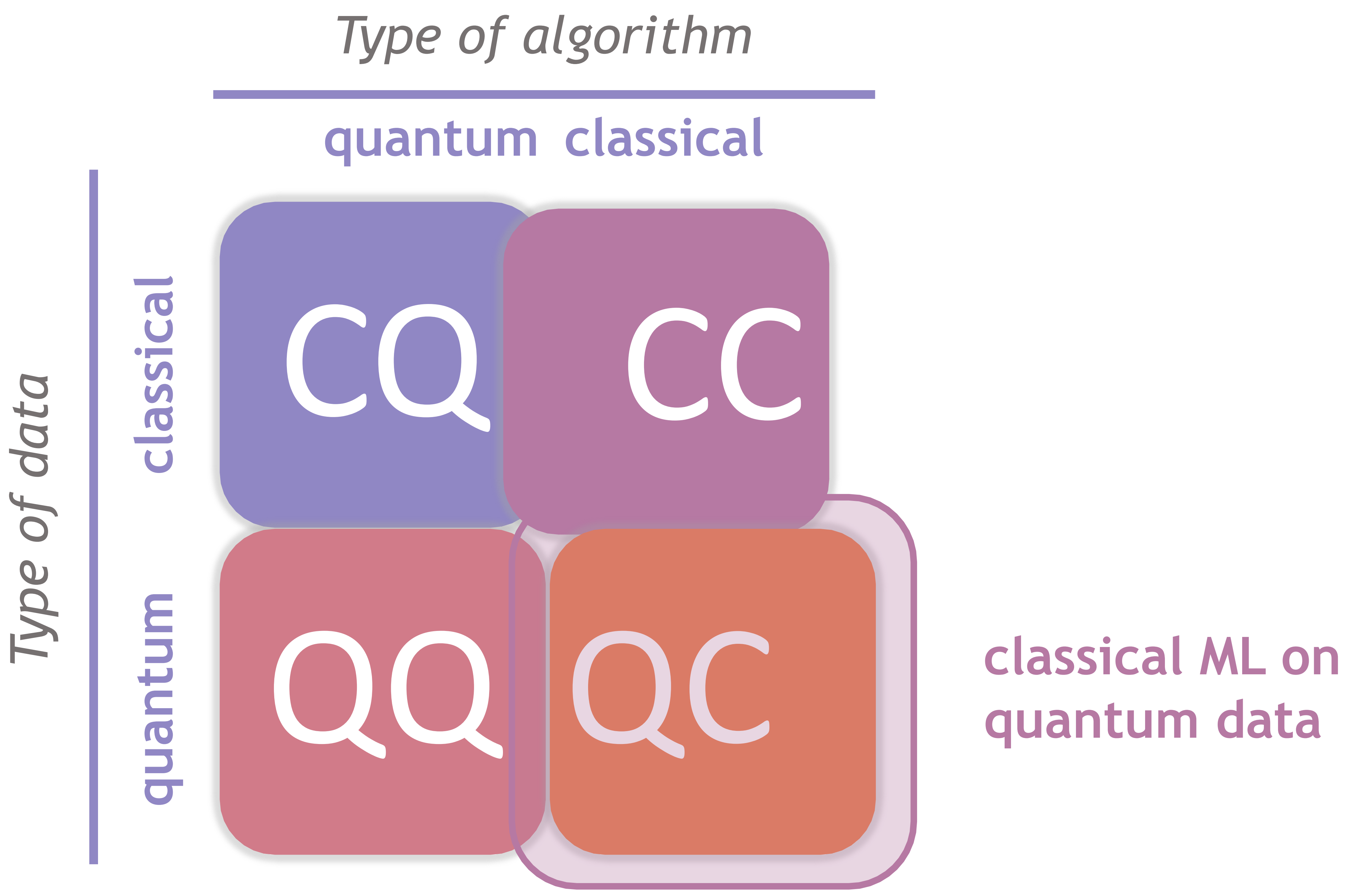
- Superposition lets us create a sum state with two operations instead of four.
- Entanglement means we can manipulate the entire state vector with one operation.
- *Exploiting* these operations with *provable* speedup is actually pretty hard! (Consider measurement if nothing else…)

Quantum machine learning



The intersection of quantum computing and ML is rich!

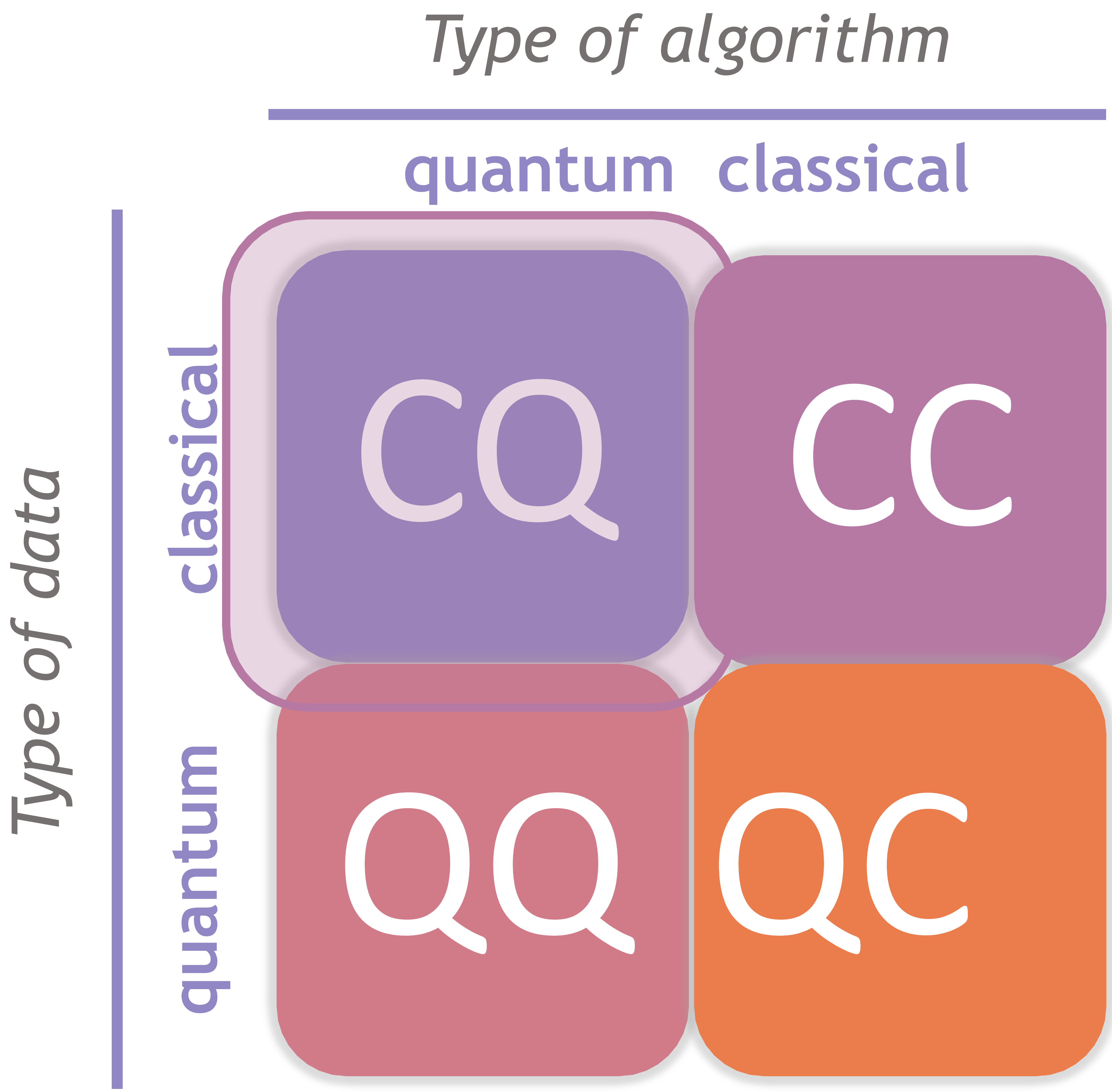
Quantum machine learning



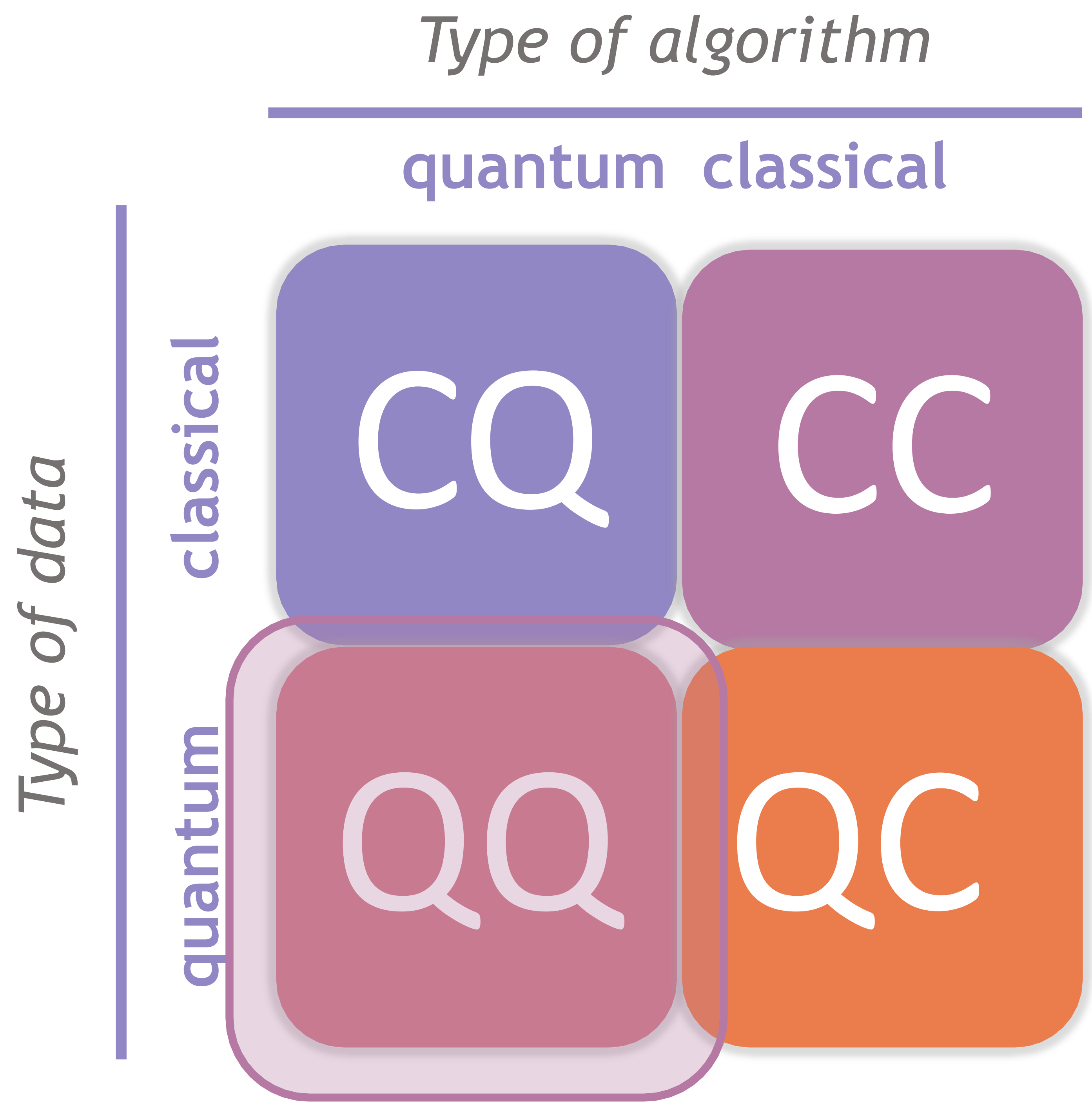
The intersection of quantum computing and ML is rich!

Quantum machine learning

Most HEP applications to date



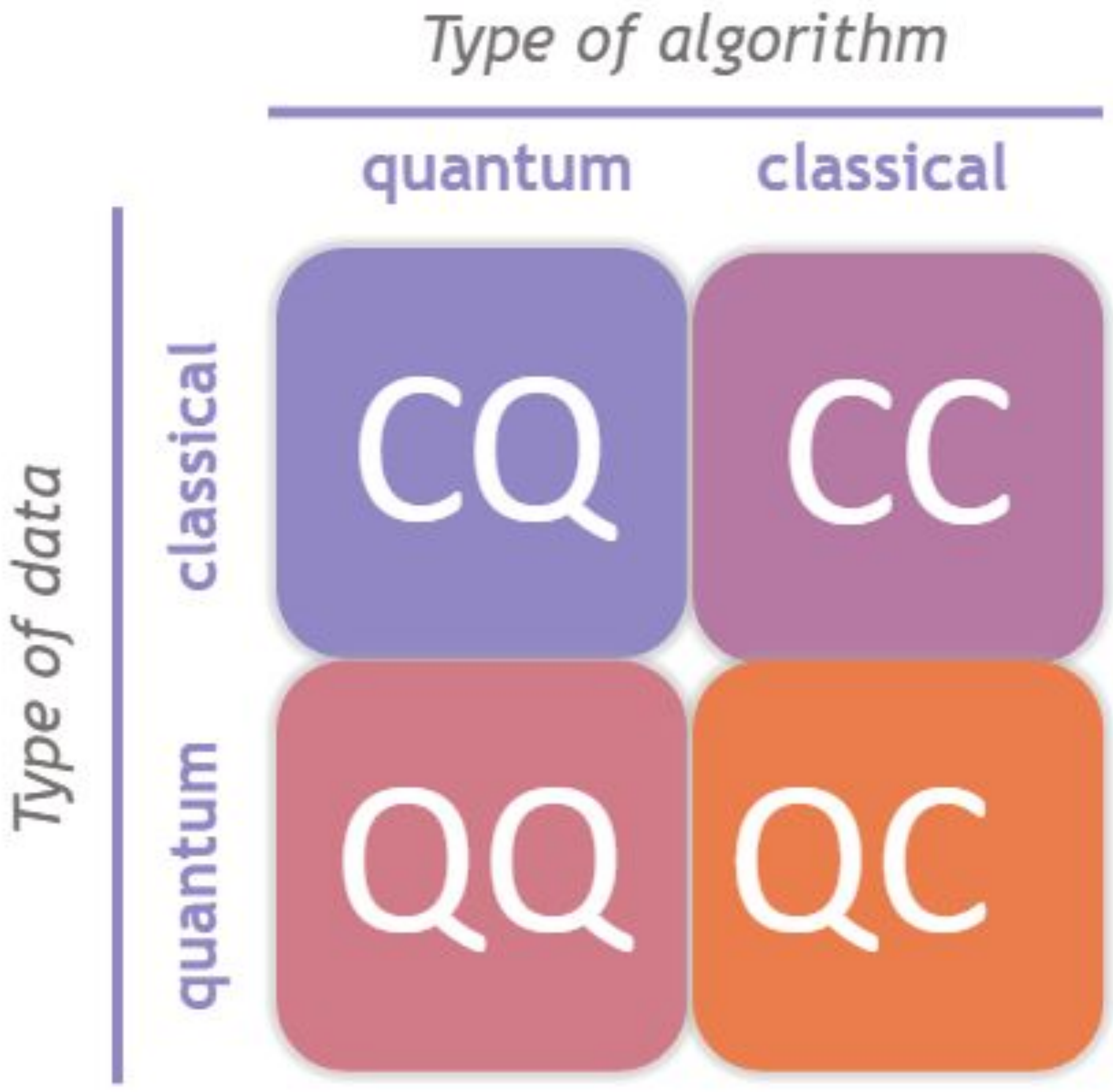
Quantum machine learning



- Chemical simulation
- Quantum matter simulation
- Quantum control
- Quantum networks
- Quantum metrology

Quantum machine learning: The power of Data

- ★ Very **unlikely that QML will beat ML** performance on classical data.
- ★ Data generated by a quantum circuit that is hard to simulate classically is *not necessarily hard to learn for a classical model.*
- ★ Datasets that are hard for classical models and easy for quantum models to learn do exist.



Understanding when a QC can help in a ML task depends not only on the task, but also on the data available, and a complete understanding of this must include both [*].

Classical Machine Learning Overview

Machine learning involves deriving patterns from data to interpret new inputs, crucial for tasks like image recognition and strategy optimization. It processes vast amounts of information, adapting to human needs without explicit programming.

Supervised Learning

Infers mappings from labeled training data, primarily for pattern classification.

Unsupervised Learning

Discovers patterns in data without prior examples, focusing on tasks like data clustering.

Reinforcement Learning

Optimizes strategies based on reward functions, common in intelligent agents and games.

Quantum Machine Learning Fundamentals

Quantum computing manipulates quantum systems to process information, leveraging superposition for computational speedup. Quantum machine learning adapts classical algorithms to run on quantum computers, aiming for greater efficiency.

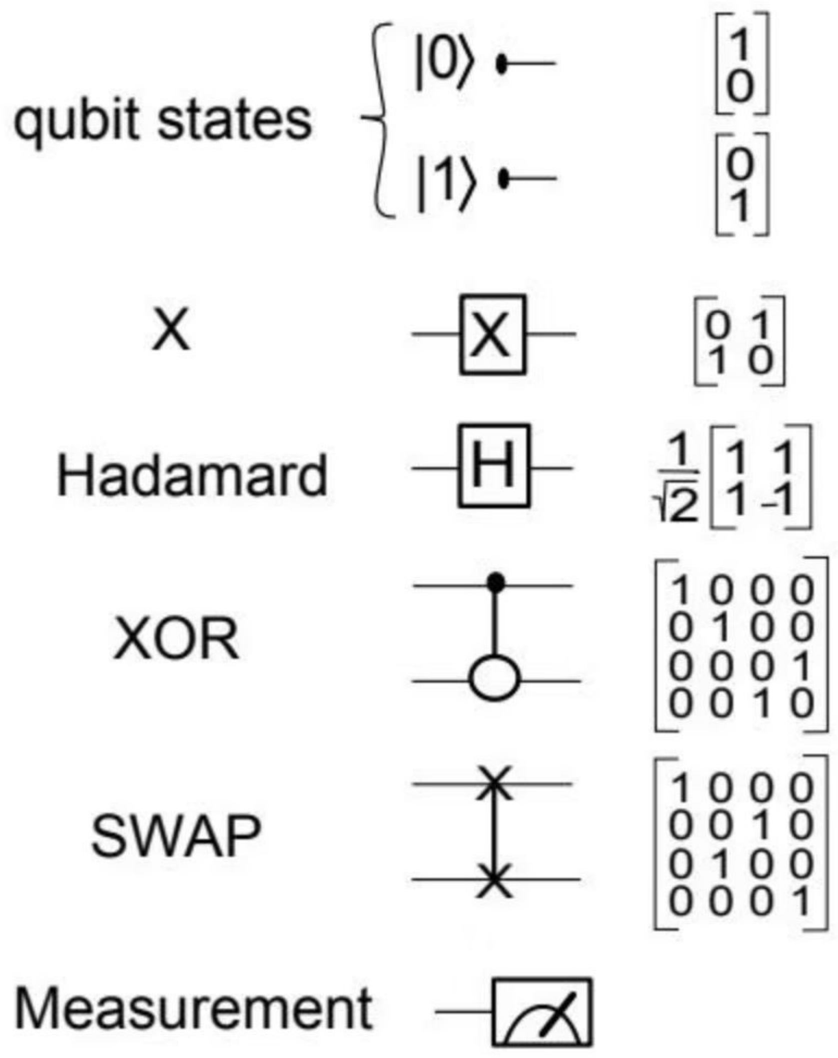
Qubit States

The basic unit is the qubit, $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ allowing operations on multiple states simultaneously.

Quantum Gates

Unitary transformations like XOR and SWAP gates manipulate qubit states, expressed as matrices.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$



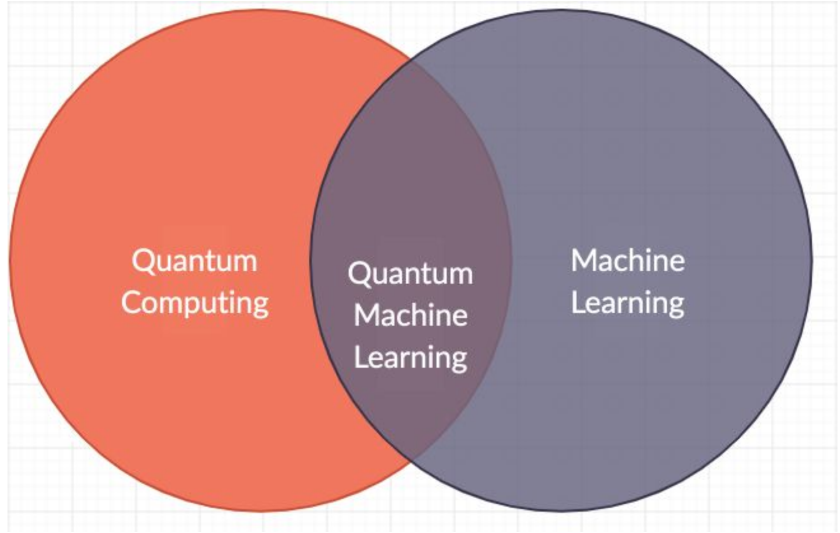


Figure 3: Representation of qubit states, unitary gates and measurements in the quantum circuit model and in the matrix formalism.

Quantum k-Nearest Neighbor Methods

The k-nearest neighbor algorithm classifies new inputs based on the majority class of their 'k' closest neighbors in a training set. Quantum versions focus on efficiently evaluating classical distances using quantum algorithms.

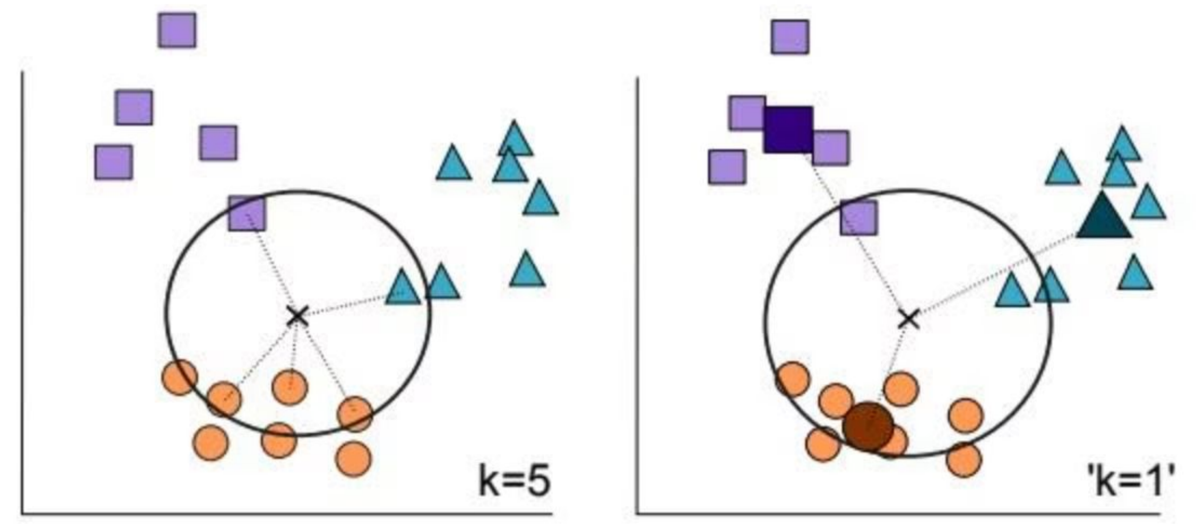


Figure 4: Illustration of the kNN method of pattern classification.

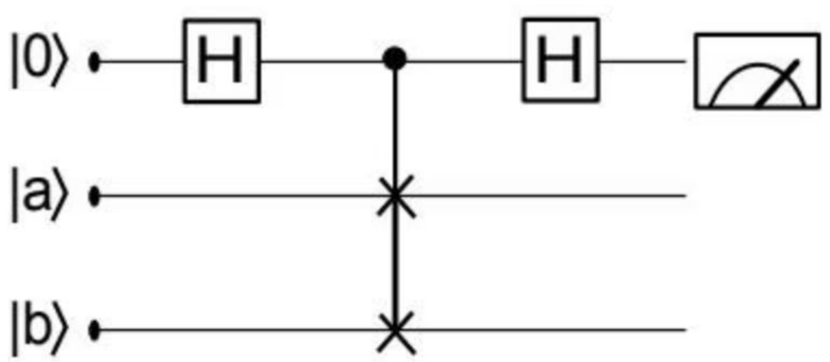


Figure 5: Quantum circuit representation of a swap test routine.

Fidelity-Based Distance (Swap Test)

Let $|\psi a\rangle$ and $|\psi b\rangle$ be quantum states corresponding to vectors $\vec{a}$ and $\vec{b}$. The **swap test** is used to compute **fidelity** (i.e., overlap):

$$P(0_{\text{anc}}) = \frac{1}{2} + \frac{1}{2}|\langle\psi_a|\psi_b\rangle|^2$$

- If states are **orthogonal**: $|\langle\psi_a|\psi_b\rangle| = 0 \Rightarrow P = 0.5$
- If states are **identical**: $|\langle\psi_a|\psi_b\rangle| = 1 \Rightarrow P = 1$

Quantum Support Vector Machines (SVM)

SVMs find an optimal hyperplane to discriminate between two class regions, serving as a decision boundary. Quantum computing aims to speed up the computationally expensive kernel calculations.

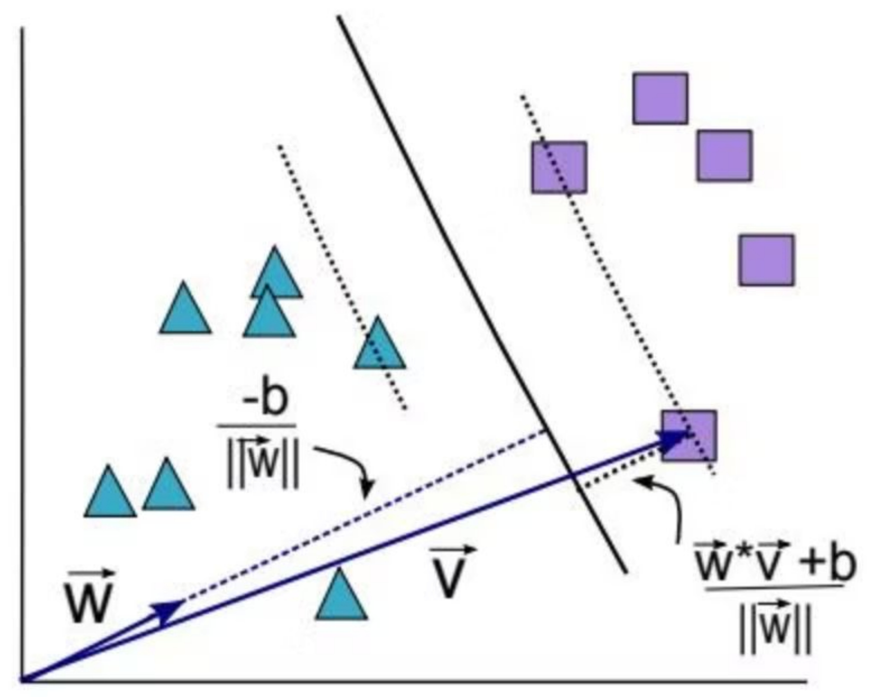


Figure 6: A support vector machine finds a hyperplane (here a line) with maximum margin to the closest vectors.

Optimization Problem

The goal is to maximize the margin between the hyperplane and the closest data points (support vectors), subject to classification constraints:

$$\vec{w}\vec{v}_i + b \geq 1, \text{ when } c_i = 1,$$
$$\vec{w}\vec{v}_i + b \leq -1, \text{ when } c_i = -1,$$

Quantum approaches claim faster inner product evaluation, crucial for kernel methods.

Kernel Method and Kernel Matrix

When data is not linearly separable in original space, map it to a **higher-dimensional space** using a kernel K :

➤ **Kernel matrix:**

$$K_{pq} = \phi(\vec{v}_p) \cdot \phi(\vec{v}_q)$$

or if we avoid explicit feature maps:

$$K_{pq} = K(\vec{v}_p, \vec{v}_q) = \vec{v}_p \cdot \vec{v}_q$$

But **computing this kernel matrix is expensive**:

> **Time complexity:** $O((Nn)^3)$, where N = number of samples, n = dimensions

Quantum Clustering Algorithms

Clustering divides unlabeled feature vectors into subsets. Quantum algorithms for clustering, like k-means, leverage quantum properties to find optimal cluster assignments and centroids.

- **Quantum Approaches to Clustering**

Several **quantum routines** aim to speed up or improve clustering, especially via **distance computation** and **optimization**.

1. **Quantum k-Median**

➤ **Procedure:**

Quantum distance oracle: Computes total distance from one state $|\psi_i\rangle$ to others in the cluster

$$D_i = \sum_{j=1}^{|C|} \mathrm{Dist}(|\psi_i\rangle, |\psi_j\rangle)$$

Use **quantum minimum finding algorithm** to find:

$$\arg\min_i D_i$$

The winner becomes the **new median** of the cluster.

- But note: Oracle implementation is not fully specified in the original proposal.

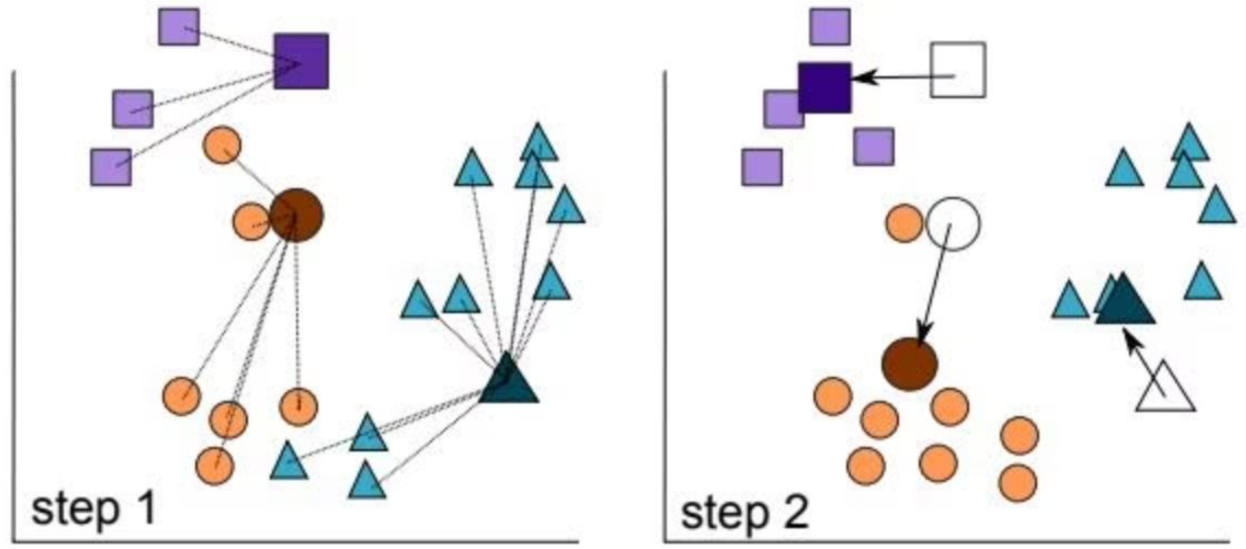


Figure 7: The alternating steps of a k-means algorithm.

Quantum Neural Networks

Researchers are exploring quantum versions of neural networks and decision trees, aiming to leverage quantum principles for improved pattern classification and learning schemes.

Feed-forward networks use sigmoid activation functions to classify patterns. Quantum versions explore integrating quantum mechanisms, though a fully efficient method is still sought.

$$x_l = \mathrm{sgm}\left(\sum_{m=1}^{N} w_{ml} x_m ; \kappa\right)$$

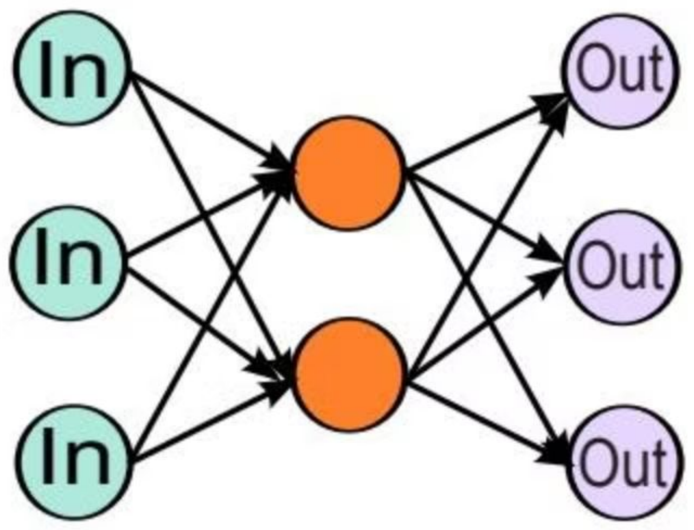


Figure 8: Illustration of a feed-forward neural network with a sigmoid activation function for each neuron.

Quantum Decision Trees

Classifiers that use a tree structure with decision functions at each node to classify inputs. Quantum decision trees propose using quantum feature states and von Neumann entropy for graph partitioning.

Quantum Feature State:

- Encode feature vector $\vec{v}_p = (v_{p1}, v_{p2}, \ldots, v_{pn})$ into quantum state:

$$|\vec{v}_p\rangle = v_{p1}|1\rangle + v_{p2}|2\rangle + \cdots + v_{pn}|n\rangle$$

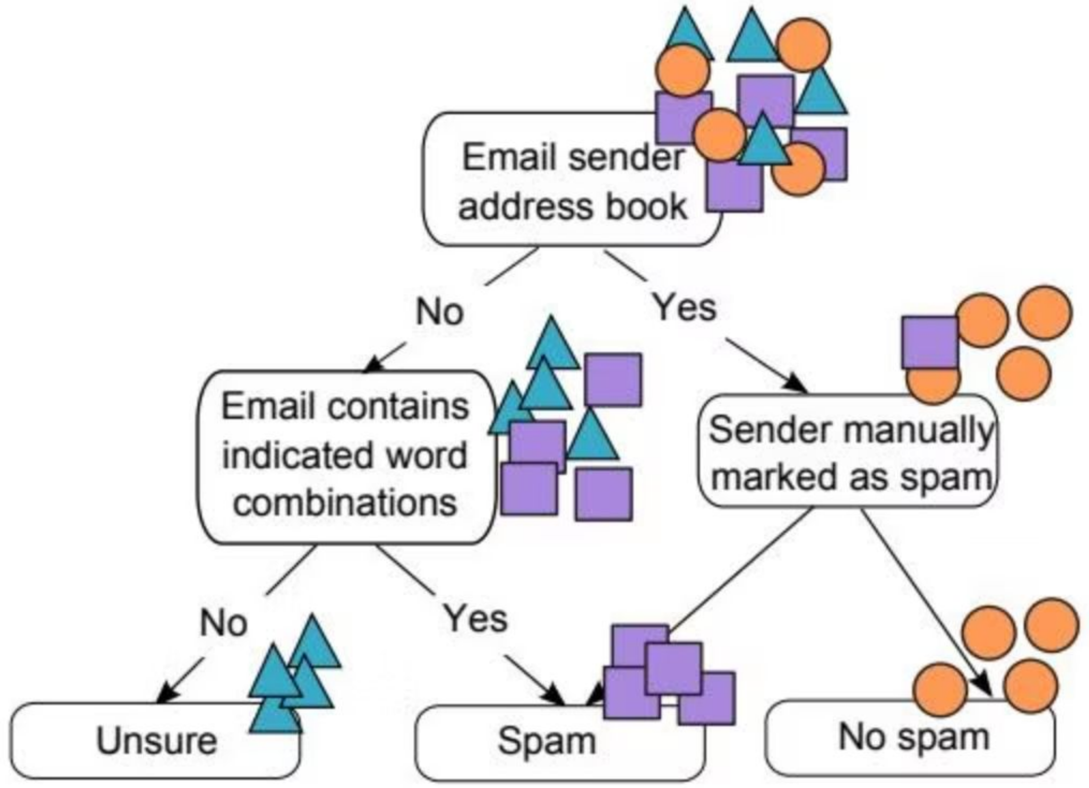


Figure 9: A simple example of a decision tree for the classification of emails.

Node Operation:

At each node: **measure an attribute** (quantum observable).
This divides the quantum training states into subsets.

Partition Evaluation:
Use **von Neumann Entropy** instead of Shannon entropy:

$$S(\rho) = -\mathrm{Tr}(\rho \log \rho)$$

- Where ρ is the density matrix of the mixed quantum state.

Limitation:
The **exact mechanism** of splitting quantum states at nodes is **not fully defined**.
Conceptual framework is introduced, but **implementation is incomplete**.

Bayesian Methods and Hidden Quantum Markov Models

Stochastic methods like Bayesian decision theory and Hidden Markov Models are being translated into quantum physics, offering new approaches for pattern classification and quantum state discrimination.

Bayesian Theory

Calculates the probability of an input belonging to a certain class using Bayes' formula:

$$p(c|\vec{x}) = \frac{p(c)p(\vec{x}|c)}{p(\vec{x})}$$

Used for quantum state classification, discriminating between quantum states from an unknown source.

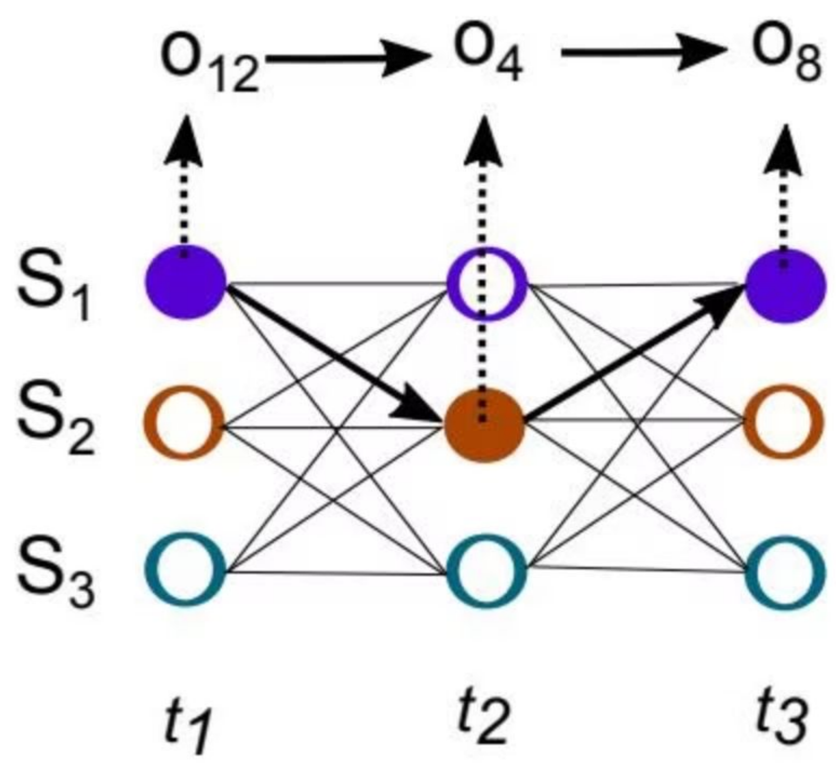


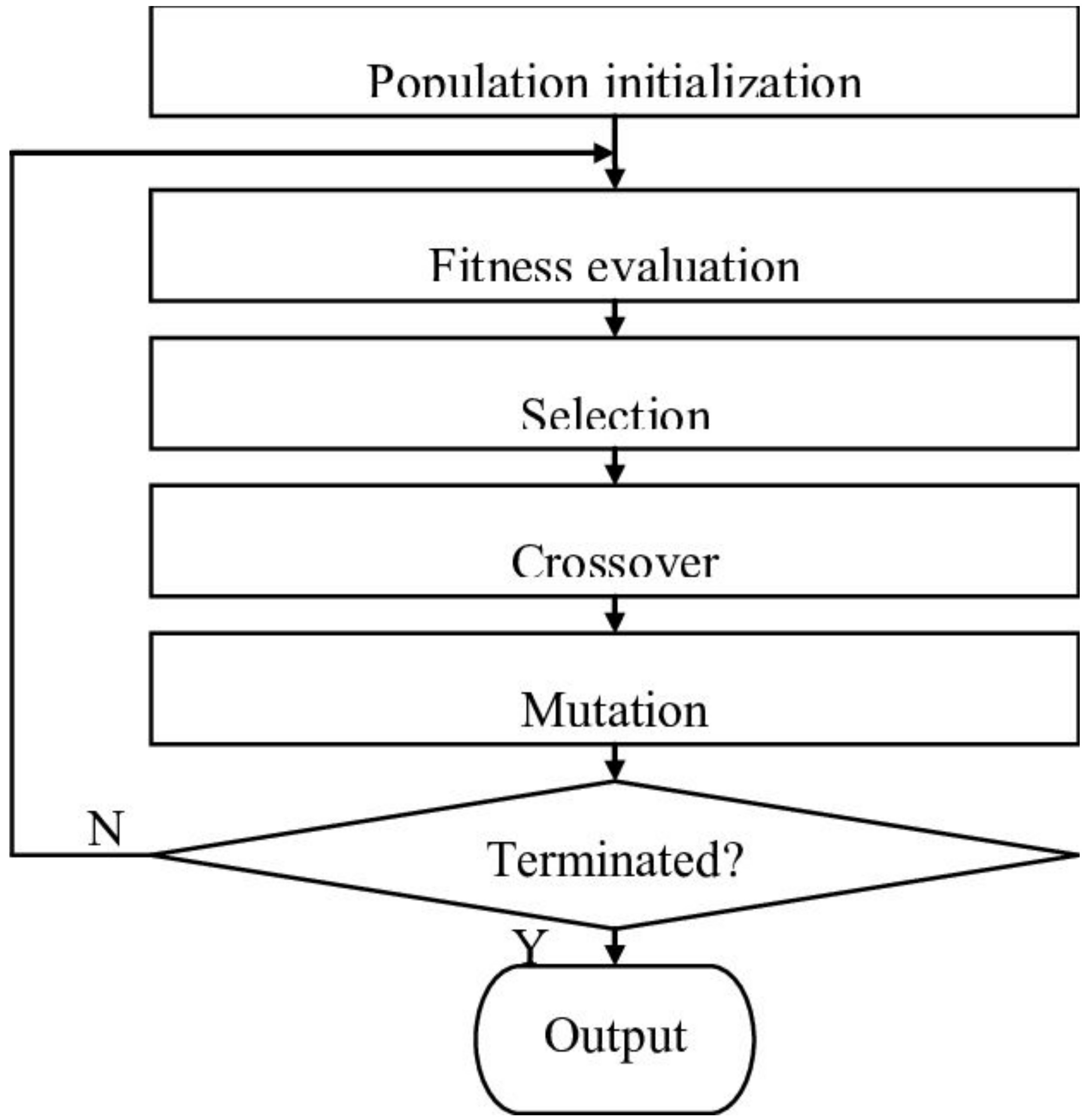
Figure 10: A hidden Markov model is a stochastic process of state transitions.

Hidden Markov Models

Markov processes where system states are accessible only through observations. Hidden quantum Markov models generalize classical models, offering richer dynamics and potential for quantum simulation.

$$\rho' = A^i \rho = \sum_k \mathcal{K}_k^i \rho \mathcal{K}_k^{i\dagger}$$

Classical Genetic Algorithm (CGA)



Why Quantum Genetic Algorithm (QGA)

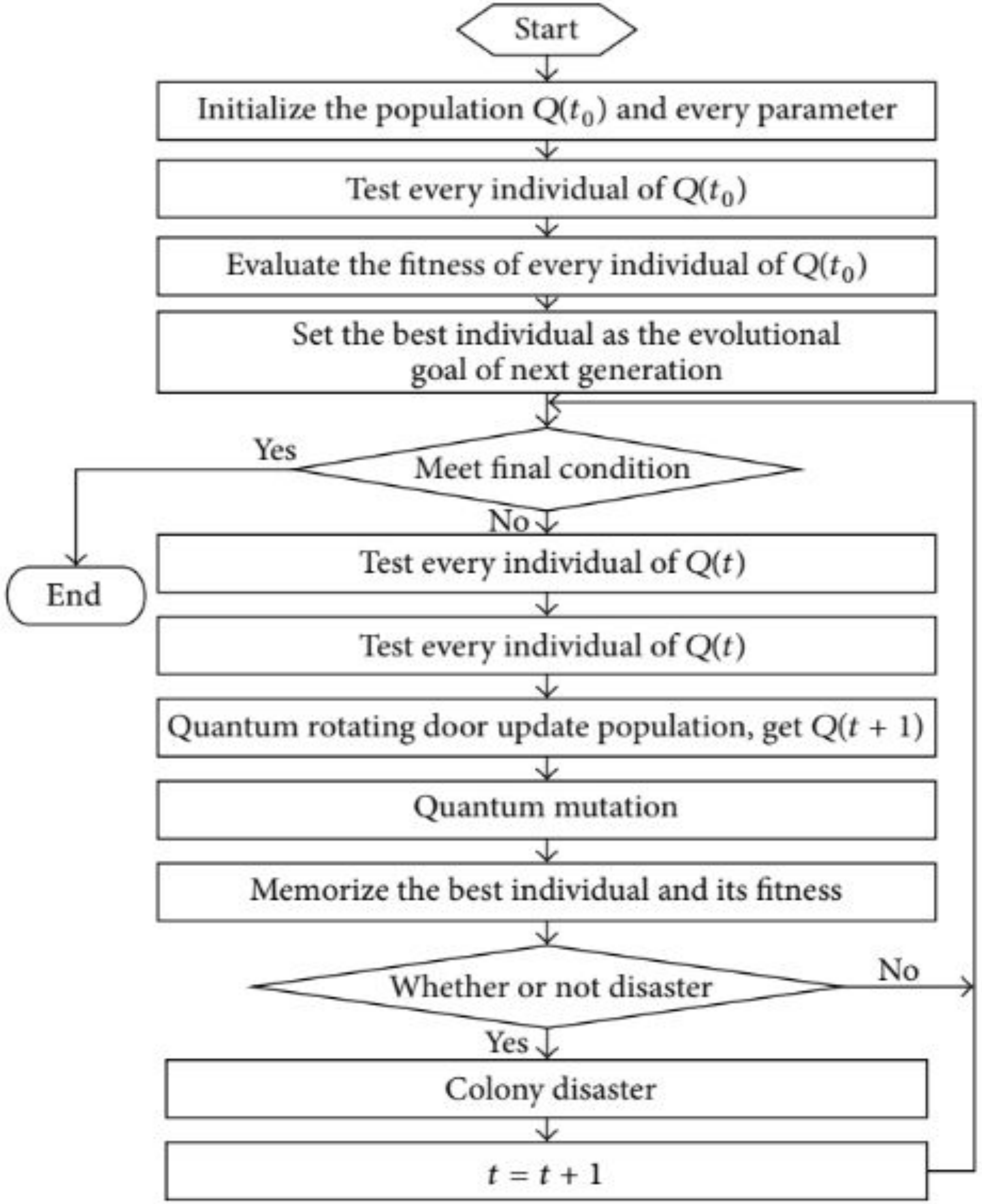
- Classical Genetic Algorithms (GAs) face issues like:

    Slow convergence

    Local optima

- Quantum Computing provides new tools: **Qubit**, **Rotation Gate**, **Superposition**
- Objective: **Improve QGA performance** in terms of speed and accuracy

Quantum Genetic Algorithm (QGA)



- QGA = Genetic Algorithm + Quantum Computing Concepts
- Inspired by quantum mechanics: *superposition*, *probability amplitudes*, and *rotation operators*
- Each individual (chromosome) is encoded as **qubits** instead of classical bits
- Enhances **diversity**, **parallelism**, and **global search**

Example:

Instead of representing a gene as 0 or 1, QGA uses:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{with } |\alpha|^2 + |\beta|^2 = 1$$

- α = amplitude of state **0**
- β = amplitude of state **1**

Quantum Rotation Gate

- Used to update qubit states towards the better solution.
- Rotation Gate matrix:

$$U(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

- Update rule:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = U(\theta) \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

Rotation Angle Direction Using Determinant

- Determines rotation direction without lookup table.
- Determinant used: $$A = \begin{vmatrix} \alpha_b & \alpha_i \\ \beta_b & \beta_i \end{vmatrix} = \sin(\theta_i - \theta_b)$$
- If A>0 $\longrightarrow$ rotate clockwise
- If A<0 $\longrightarrow$ rotate counterclockwise

Self-Adaptive Rotation Angle

- Rotation angle decreases with generations to balance exploration and exploitation.
- Formula: $\theta_i = \theta_{max} - \left(\frac{\theta_{max} - \theta_{min}}{it_{max}}\right) \times iter$
- Larger angle in early stages (diversification).
- Smaller angle in later stages (intensification).

Quantum Mutation

- With small probability (e.g., 0.001), **swap** α and β.
- Helps maintain population diversity.
- Mutation prevents premature convergence to local optima.

Quantum Disaster Operator

- Applied when the algorithm stagnates (no improvement for many generations).
- Randomly reinitializes part of the population (except the best individual).
- Helps the algorithm escape from local optima traps.

Fitness Function Example

- Example used in the research paper:

$$f(x, y) = x \cdot \sin(4\pi x) + y \cdot \sin(20\pi y)$$

- The goal is to maximize this function using QGA.

References

- Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). An introduction to quantum machine learning. Contemporary Physics, 56(2), 172–185. https://doi.org/10.1080/00107514.2014.964942
- Wang, H., Liu, J., Zhi, J., & Fu, C. (2013). The improvement of Quantum genetic Algorithm and its application on function optimization. Mathematical Problems in Engineering, 2013, 1–10. https://doi.org/10.1155/2013/730749
- Gupta, S., & Zia, R. K. P. (2001). Quantum neural networks. In Journal of Computer and System Sciences (Vols. 63–383) [Journal-article]. https://doi.org/10.1006/jcss.2001.1769

Thanks