



**INAF**

ISTITUTO NAZIONALE DI ASTROFISICA  
OSSERVATORIO ASTROFISICO DI CATANIA



Centro Nazionale di Ricerca in HPC,  
Big Data and Quantum Computing

# Quantum Machine Learning

**USC-C General assembly, Trieste, 11/03/2026**

---

**Farida Farsian**

farida.farsian@inaf.it

# OUTLINES

---

## Why Quantum Machine Learning?

- Limitations of classical ML
- What makes ML quantum?
- Data Encoding

## Types of Quantum ML Approaches

- Variational Quantum Circuits
- Quantum Kernels

## What Quantum ML can do today?

- Optimisation problem
- What Works Today on Real Hardware?
- Future direction

# QUICK INTRODUCTION OF MACHINE LEARNING (ML)

---

## Machine Learning Definition:

- ▶ Using data to answer questions, Machine learning uses statistical techniques to give computers the ability to “learn” with data, without being explicitly programmed

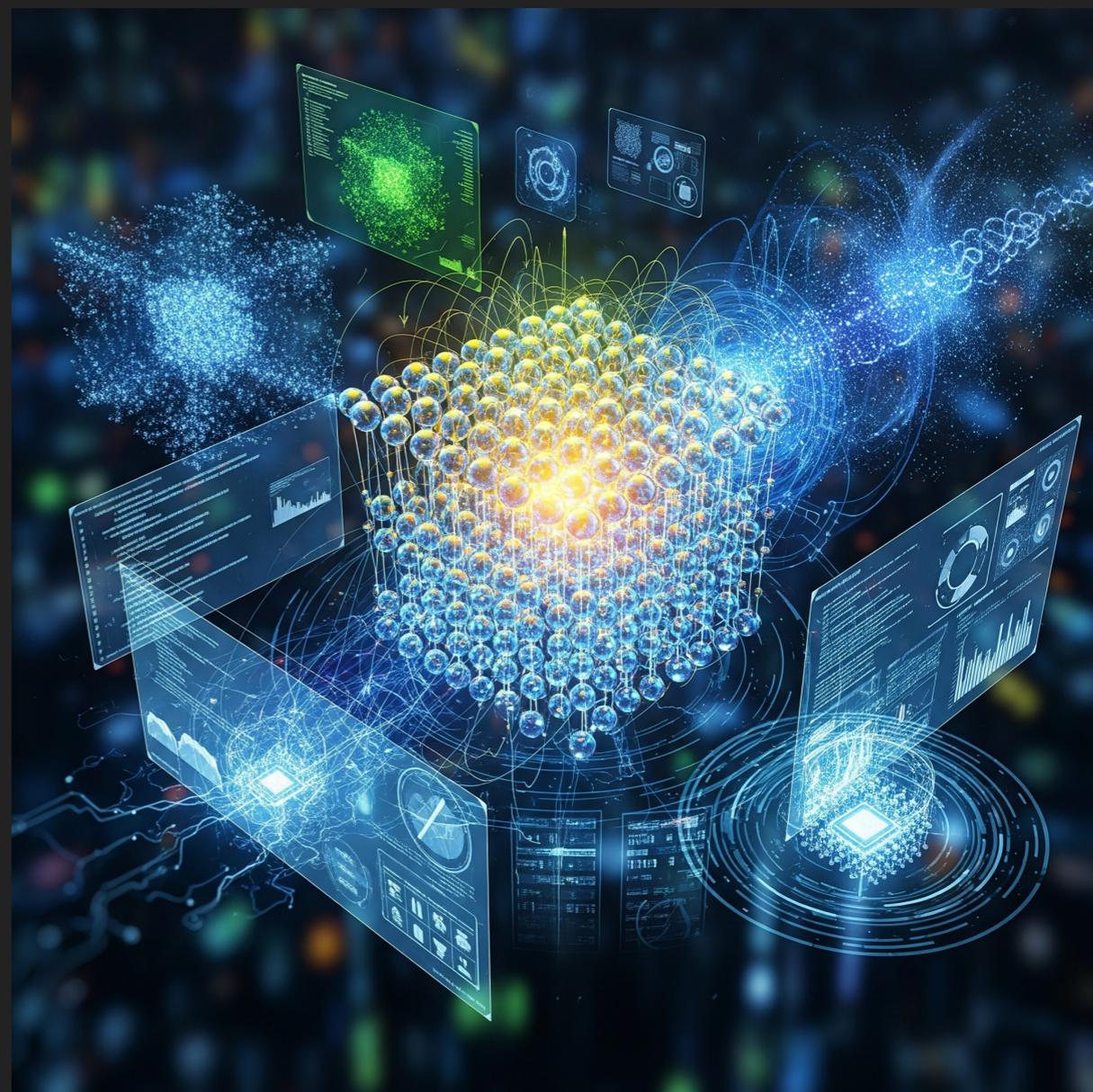
## Different Types of Machine learning:

- ▶ Supervised: The system tries to learn from the previous examples that are given
  1. Classification
  2. Regression
- ▶ Unsupervised: The algorithms are left to themselves to discover interesting structures in the data.
  3. Association
  4. Clustering



# WHY QUANTUM MACHINE LEARNING?

- ▶ Classical ML struggles with some classes of high-dimensional data
- ▶ Quantum systems naturally represent  **$2^n$ -dimensional vectors** with  $n$  qubits
- ▶ QML explores whether **quantum feature spaces** can capture patterns difficult for classical models
- ▶ Hybrid quantum-classical methods are promising for the **NISQ era**
- ▶ QML is not about replacing ML, but about finding tasks where quantum circuits offer expressive power



# THE IMPORTANT PLAYERS IN QML

- ▶ **Qubit** : The unit of information for QC

$$|\psi\rangle = \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = c_0 |0\rangle + c_1 |1\rangle$$

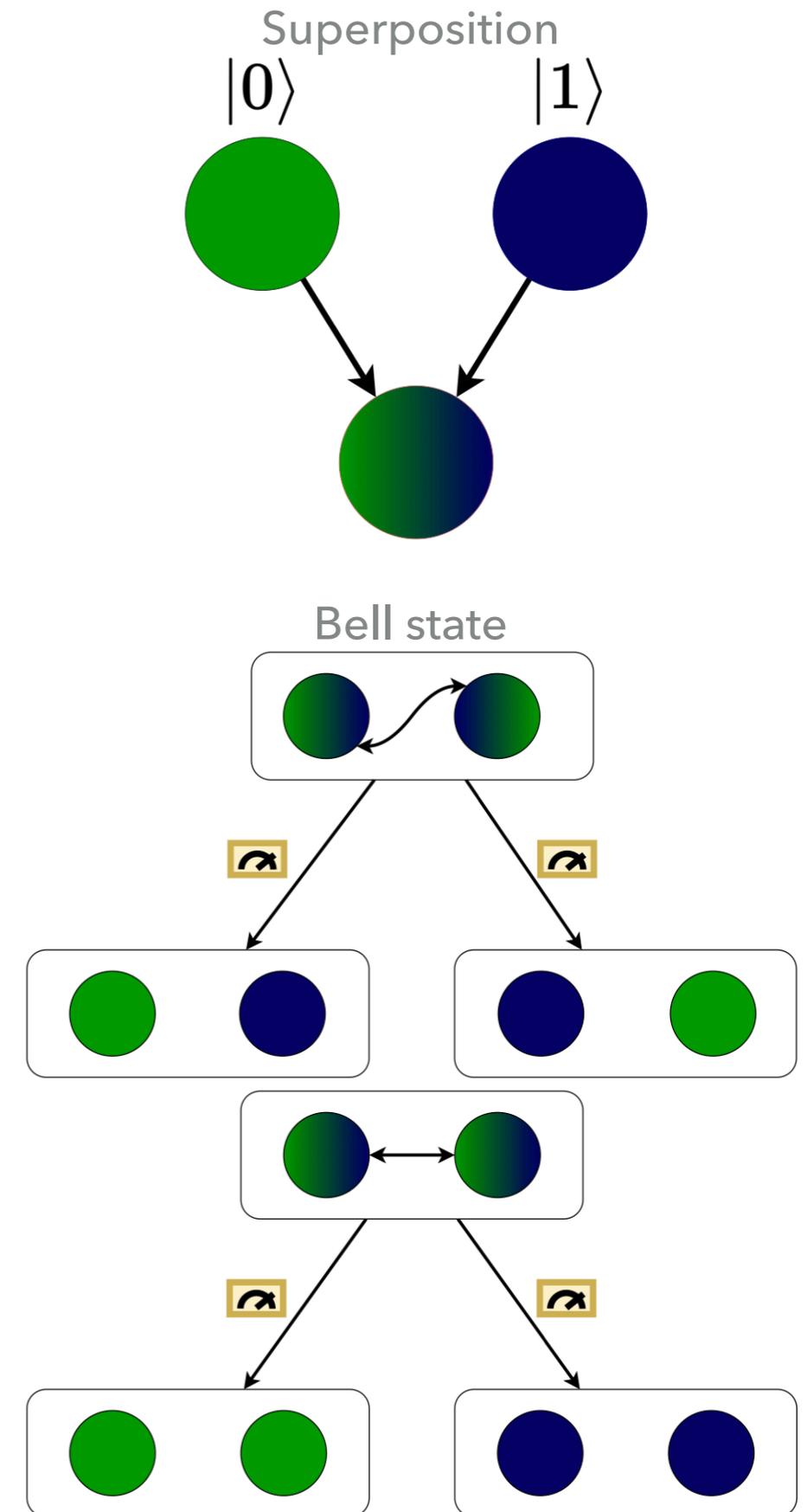
- ▶ **Measurement**: an operation that alters the system and is a non-deterministic process (unlike classical computation).

- ▶ The basis of Quantum computing:

**Superposition**: the state with no-null probability of being in both the state  $|0\rangle$  and  $|1\rangle$ .

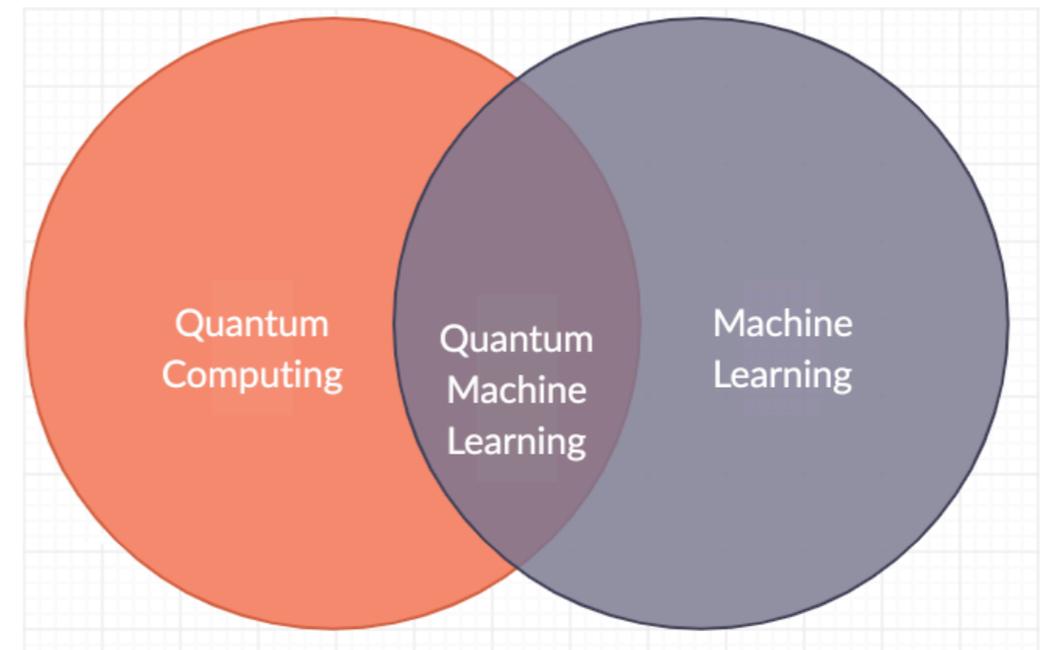
**Entanglement**: the correlation of two qubits.

- ▶ **Quantum gate**: transformations (matrices) which can be used to manipulate the qubits. They should have these properties: Linearity, Unitarity and Reversibility

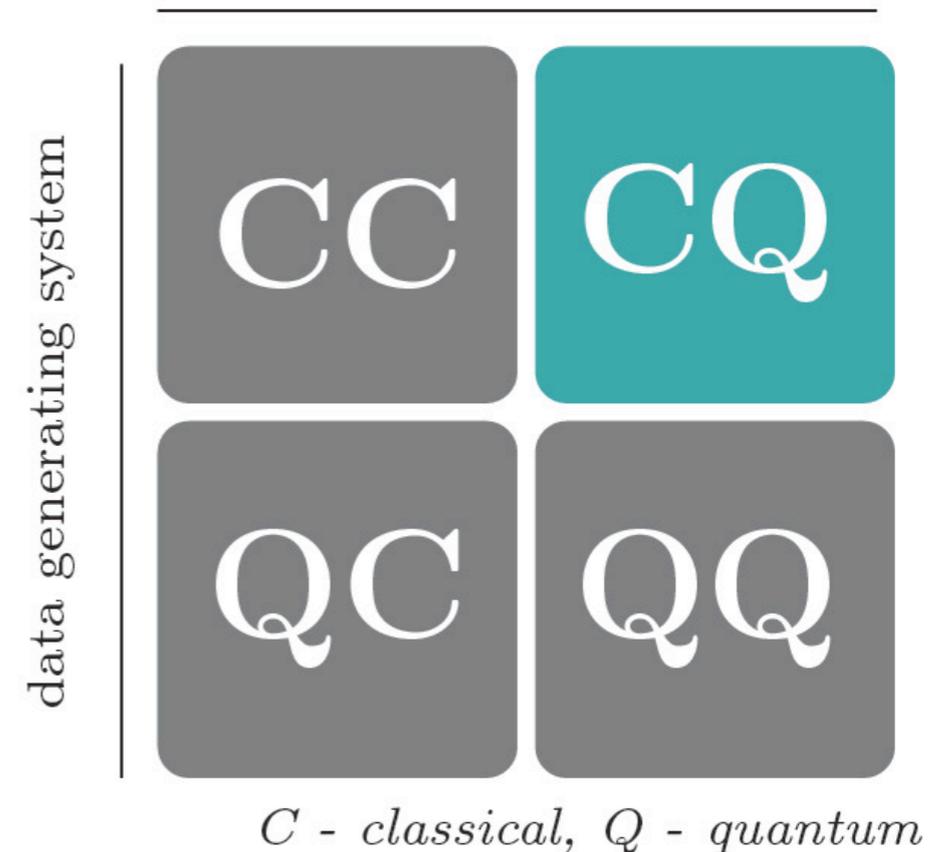


# QUANTUM MACHINE LEARNING (QML)

- ▶ Combines concepts from **quantum computing** and **machine learning** to develop algorithms capable of exploiting quantum phenomena to enhance learning tasks.
- ▶ Utilizes **Quantum Properties** like superposition and entanglement to process high-dimensional data spaces and explore multiple solutions simultaneously.
- ▶ This could potentially lead to **exponential speedups** for certain problems.
- ▶ QML faces **challenges** like fault-tolerant hardware, efficient algorithms, and integrating quantum with classical systems.



data processing device



# WHAT MAKES ML “QUANTUM”?

---

**QML USES QUANTUM  
CIRCUITS AS LEARNABLE  
MODELS**

**DATA IS ENCODED INTO  
QUANTUM STATES  $|\Psi(x)\rangle$**

**CIRCUITS RUN  
PARAMETERIZED GATES  
 $U(\theta)$**

**MEASUREMENT  
GENERATES CLASSICAL  
OUTPUTS FOR A LOSS  
FUNCTION**

**TRAINING LOOP UPDATES  
 $\theta$  VIA CLASSICAL  
OPTIMIZATION**

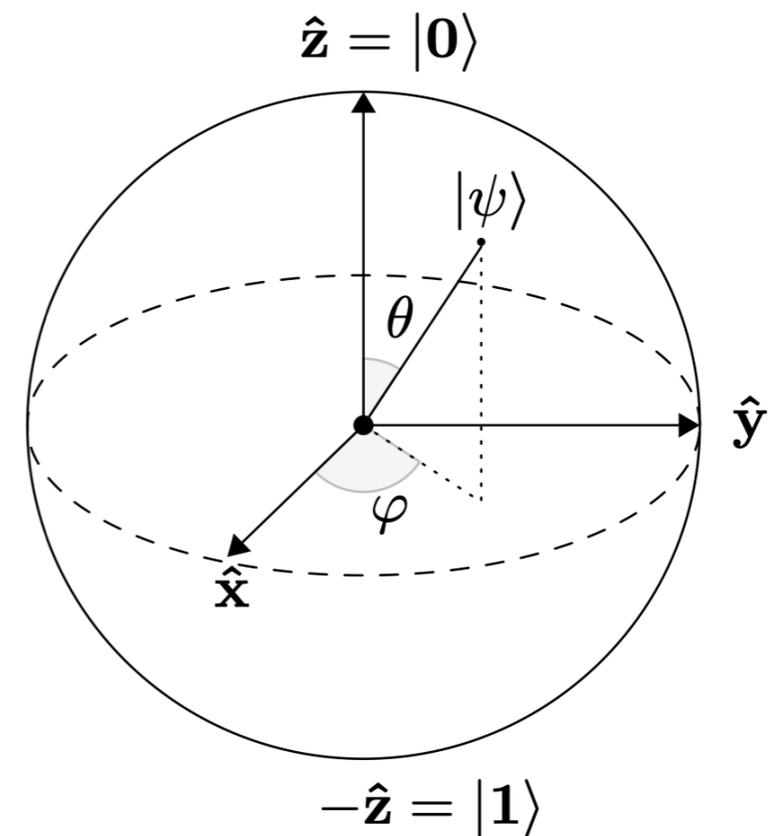
# DATA ENCODING METHODS

---

For encoding (embedding), we take a classical data point,  $x$ , and encode it by applying a set of gate parameters in the quantum circuit.

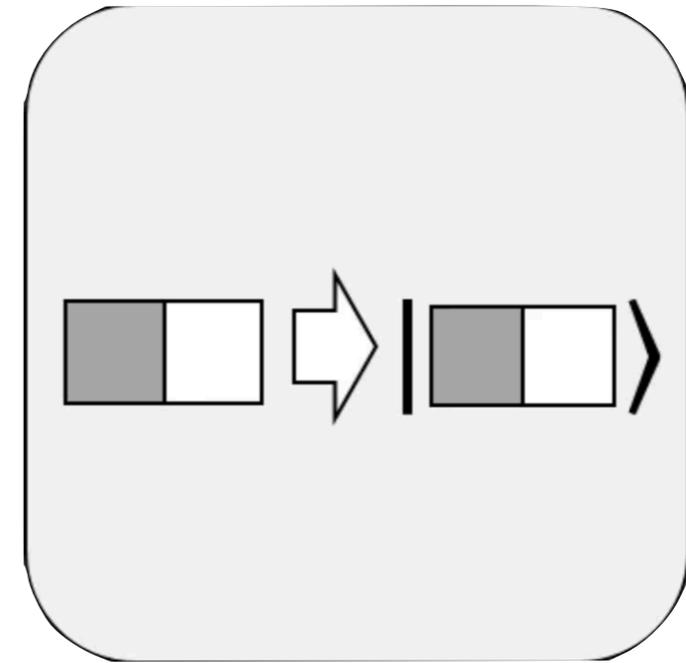
There are different types of encoding the data:

- ▶ **Basic encoding**
- ▶ **Angle encoding**
- ▶ **Amplitude encoding**
- ▶ Data Reuploading
- ▶ QuAM (Quantum Associated Memory)
- ▶ QRAM (Quantum Random Access Memory)
- ▶ ...



# BASIC ENCODING

- ▶ This encoding represents real numbers as binary numbers and then transforms them into a quantum state on a computational basis.
- ▶ Is not efficient in terms of the required number of qubits but is good for arithmetic operation.

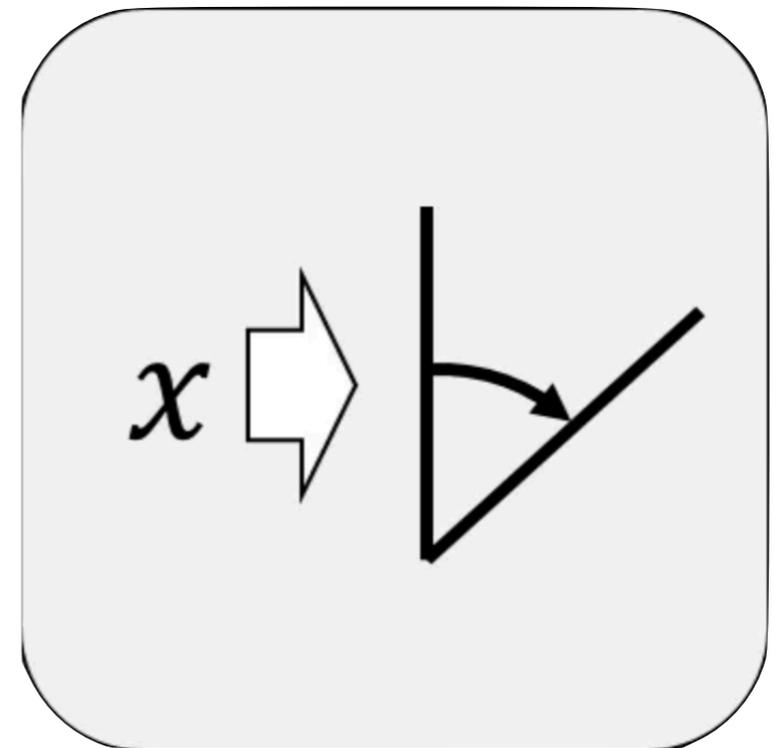


Input Data			Pre-processing to convert to Binary form		Basis encoded state	
Input Sample	Feature x1	Feature x2	Binary(x1)	Binary(x2)	Basis encoded Quantum state	Amplitude vector
$x^1$	5	6	101	110	$ 101110\rangle$	$\frac{1}{\sqrt{2}} 101110\rangle$
$x^2$	4	1	100	001	$ 100001\rangle$	$\frac{1}{\sqrt{2}} 100001\rangle$

# ANGLE ENCODING

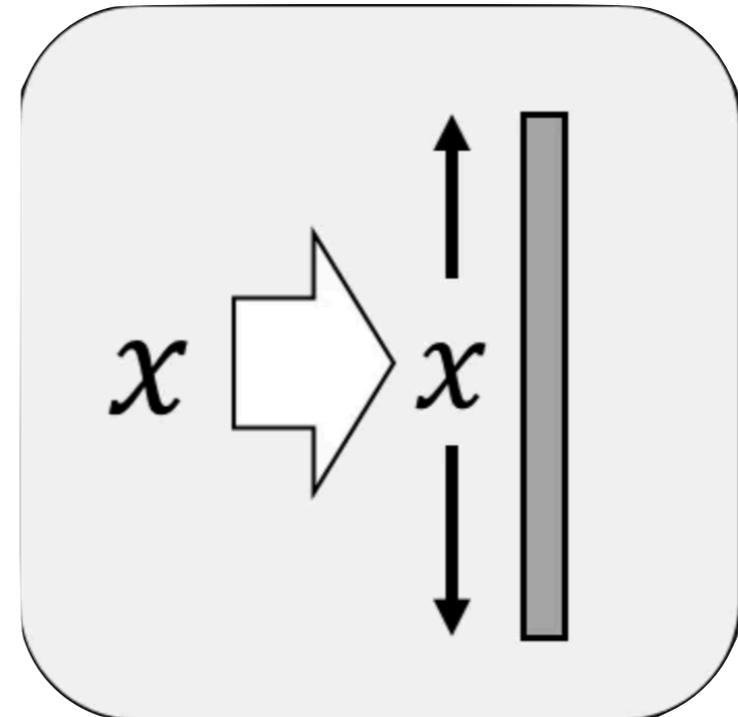
---

- ▶ The  $n$  classical features are encoded into the rotation angle of the  $n$  qubit.
- ▶ It requires  **$n$  qubits** to represent  $n$ -dimensional data but is cheaper to prepare in complexity: it requires one rotation on each qubit.



# AMPLITUDE ENCODING

- ▶ The data is encoded into the amplitudes of a quantum state.
- ▶ This encoding requires  **$\log_2(n)$  qubits** to represent an  $n$ -dimensional data point.



	x1	x2	x3	x4
Input Classical Data	6	-12.5	11.15	7
Normalization	$\frac{6}{19.12}$	$\frac{-12.5}{19.12}$	$\frac{11.15}{19.12}$	$\frac{7}{19.12}$
Amplitude encoding	$\frac{6}{19.12}  00\rangle - \frac{12.5}{19.12}  01\rangle + \frac{11.15}{19.12}  10\rangle + \frac{7}{19.12}  11\rangle$			

# QISKIT EXAMPLE

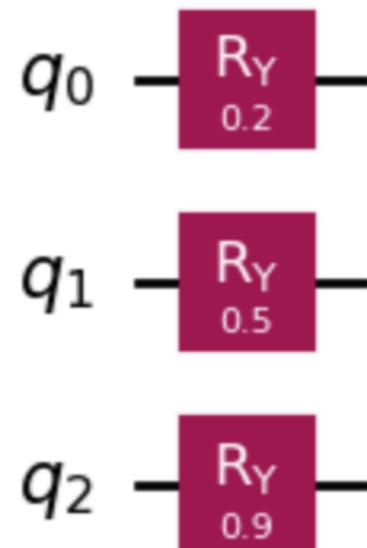
```
▶ from qiskit import QuantumCircuit
  from qiskit.visualization import circuit_drawer

x = [0.2, 0.5, 0.9]

# Create a quantum circuit with 3 qubits
qc = QuantumCircuit(3)

# Apply RY gates with the random values for angle encoding
for i, value in enumerate(x):
    qc.ry(value, i)

# Draw the circuit using Matplotlib
circuit_drawer(qc, output='mpl')
```



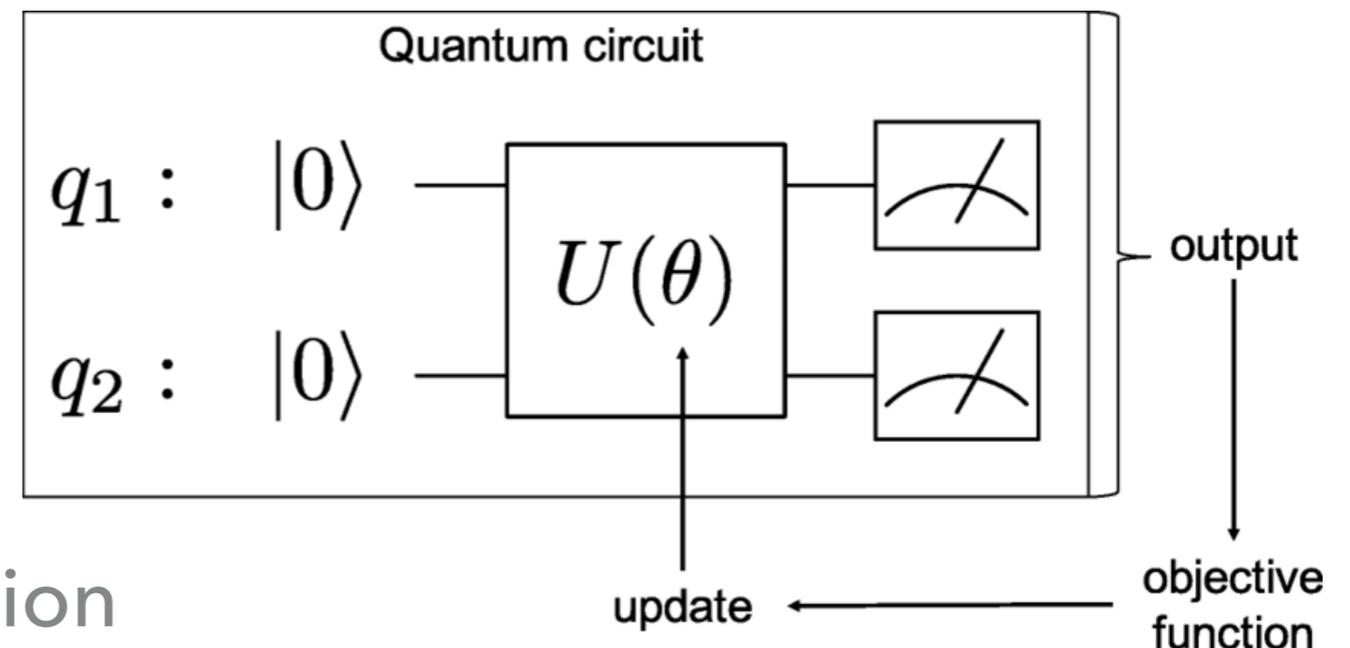


# VARIATIONAL QUANTUM CIRCUIT (VQC)

- A **VQC** is a quantum circuit with tunable parameters  $\theta$
- Parameters are optimized to minimize a classical loss

- Typical structure:

- Encoding layer ( $\Phi(x)$ )
- Learnable ansatz ( $U(\theta)$ )
- Measurement  $\rightarrow$  prediction



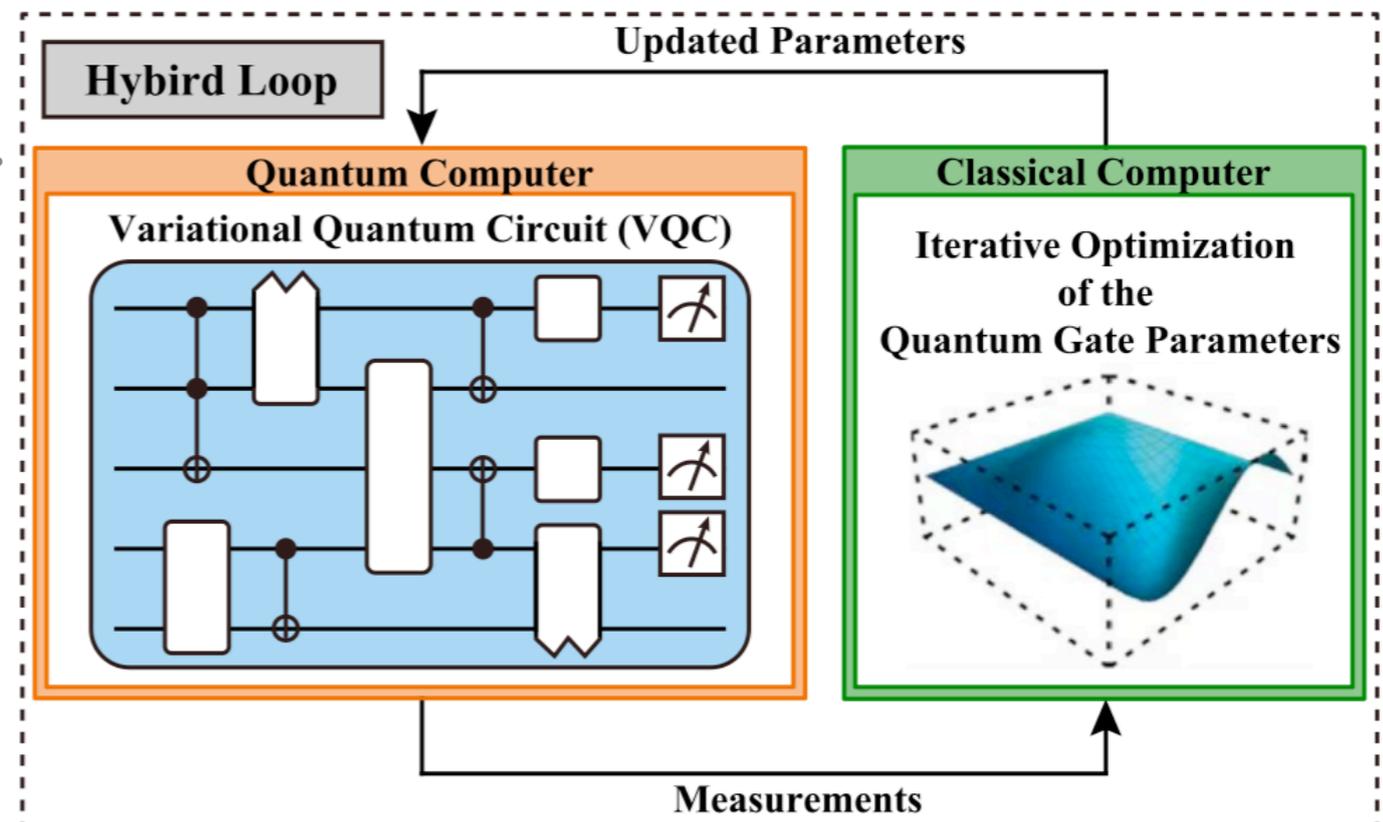
VQCs are the quantum analogue of classical neural networks.

# THE HYBRID OPTIMIZATION LOOP

Pipeline:

1. Encode classical data  $\rightarrow |\psi(x)\rangle$
2. Run parameterized quantum circuit  $U(\theta)$
3. Measure qubits  $\rightarrow$  classical outcomes
4. Compute loss function (cross-entropy, MSE, etc.)
5. Classical optimizer updates  $\theta$

This loop repeats until convergence.



# VARIATIONAL ANSATZ STRUCTURE

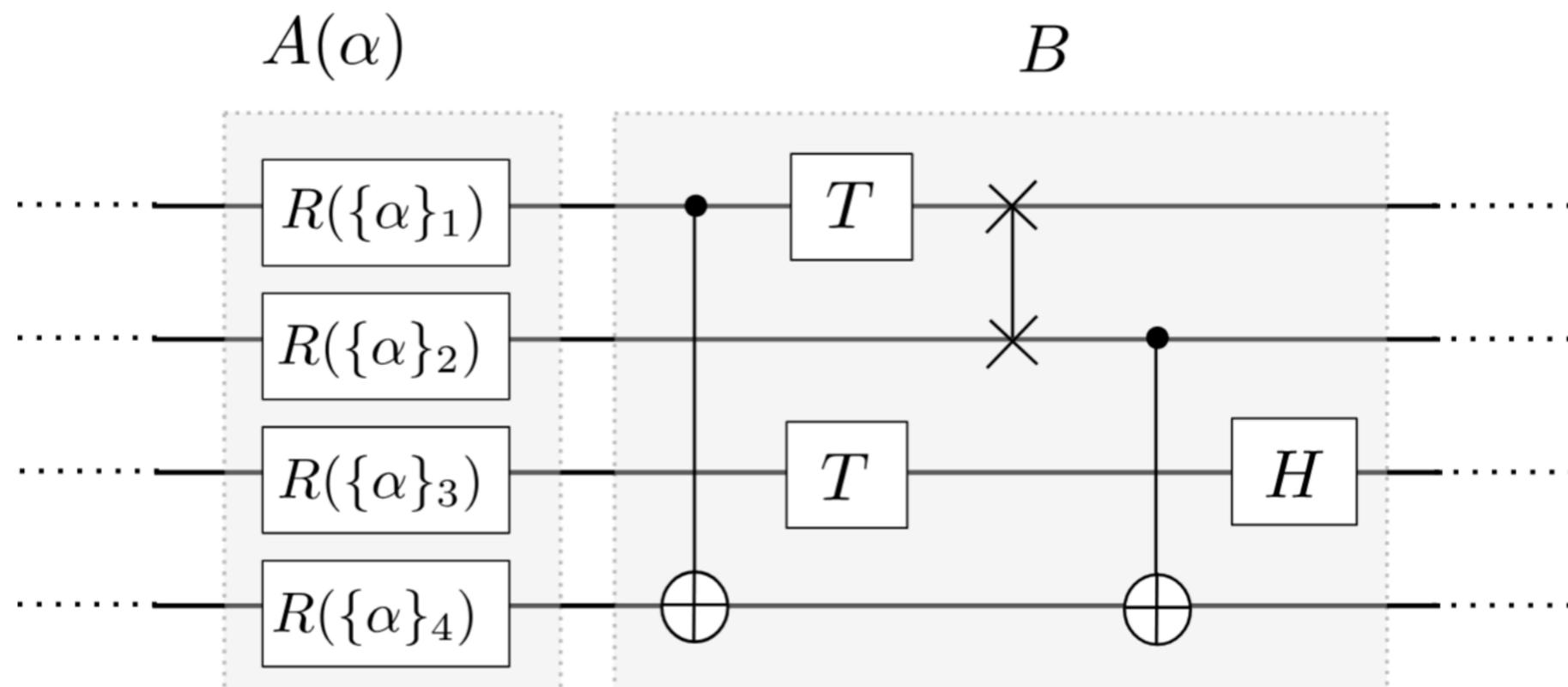
---

A generic variational layer consists of:

- **Single-qubit rotations:**  $R_x(\theta)$ ,  $R_y(\theta)$ ,  $R_z(\theta)$
- **Entanglement block:** CNOT ring or full connectivity
- **Multiple repetitions** (reps) increase expressivity

**Trade-off:**

Deeper circuits  $\rightarrow$  more expressive  $\rightarrow$  but harder to train (barren plateaus).



# QISKIT EXAMPLE

```
from qiskit import QuantumCircuit
from qiskit.circuit.library import RealAmplitudes, ZZFeatureMap
from qiskit.visualization import circuit_drawer
from qiskit.circuit import ParameterVector

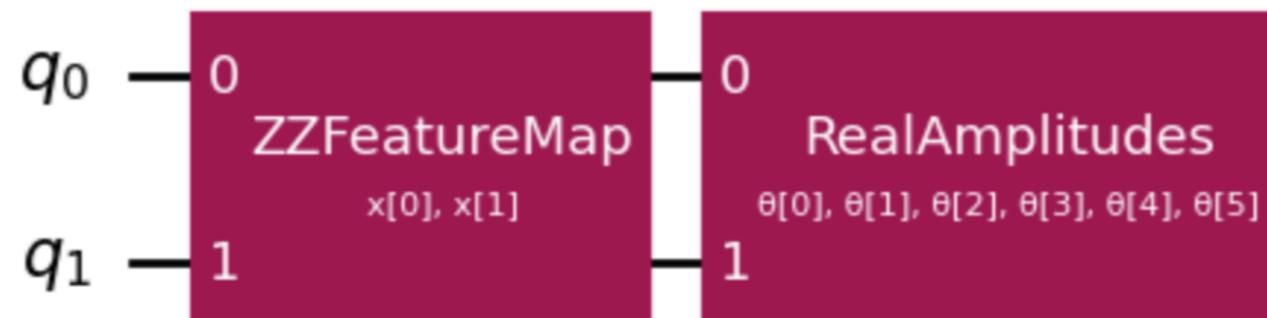
# Define the number of qubits
num_qubits = 2

# Define parameters for the feature map (input data)
input_params = ParameterVector('x', num_qubits)

# Create a simple feature map (e.g., ZZFeatureMap)
feature_map = ZZFeatureMap(num_qubits, reps=1)

# Define the ansatz (RealAmplitudes)
ansatz = RealAmplitudes(num_qubits, reps=2)

# Combine the feature map and ansatz to create the VQC circuit
# The ansatz parameters (ansatz.parameters) are the trainable weights
vqc_circuit = feature_map.compose(ansatz)
|
circuit_drawer(vqc_circuit, output='mpl', idle_wires=False)
```



# VARIATIONAL MODELS: PROS & CONS

## Advantages

Flexible and expressive

Work on real hardware today

Integrate naturally into hybrid ML pipelines

Good for small data and simple patterns

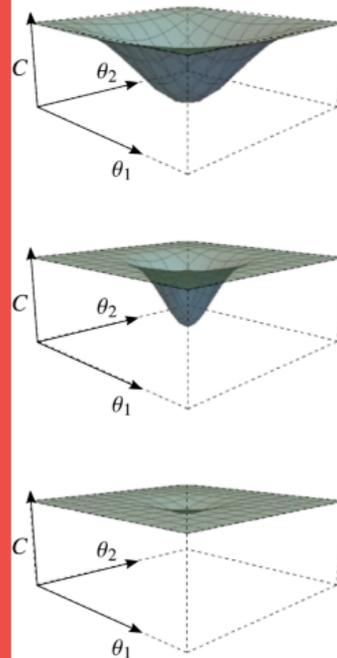
## Challenges

Suffer from barren plateaus

Sensitive to noise

Encoding cost can dominate

Hard to scale beyond ~6 qubits on hardware



# QUANTUM FEATURE MAPS

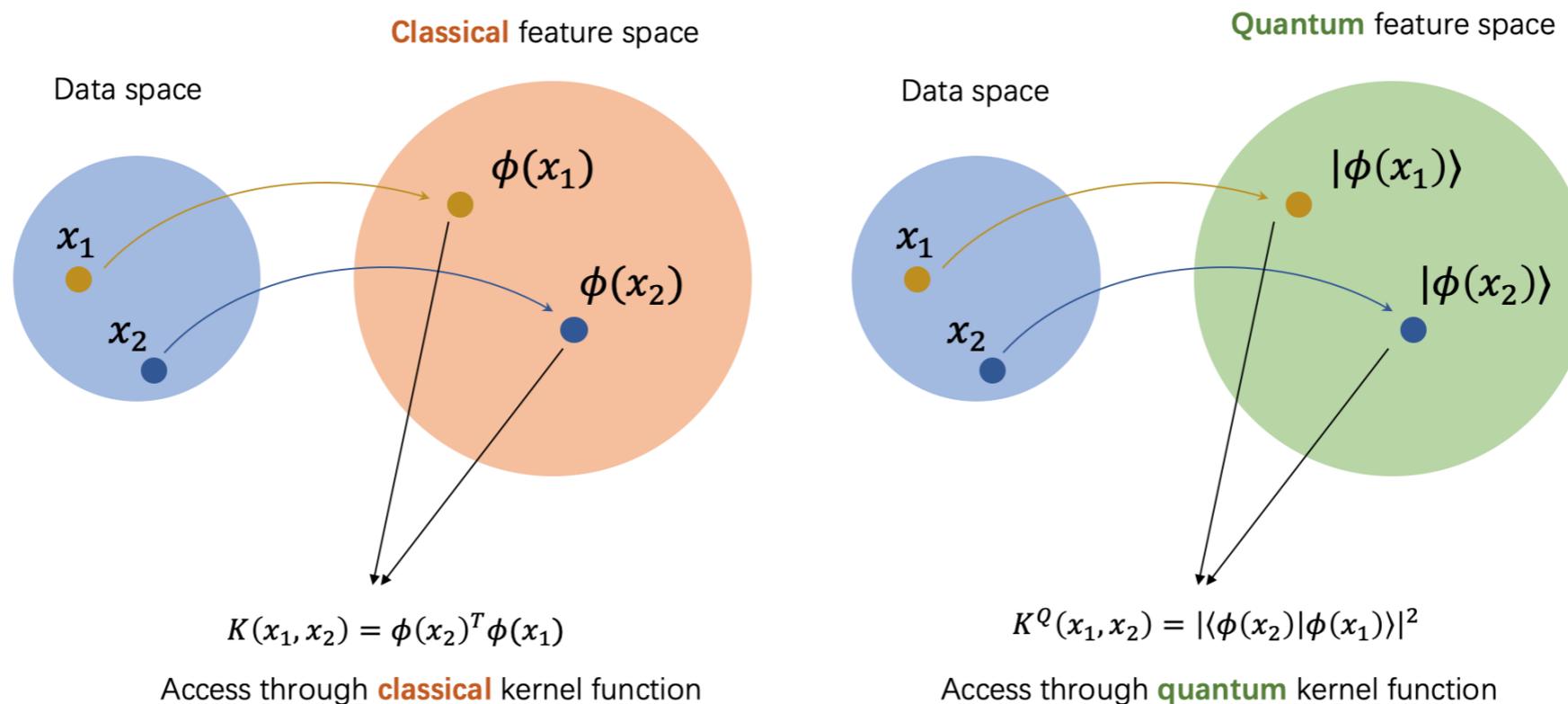
A quantum feature map  $\Phi(x)$  encodes classical data into a quantum state:

$$|\psi(x)\rangle = U_\phi(x) |0\rangle$$

Important properties:

- Introduces complex nonlinear transformations
- Might be classically hard to simulate
- Can be used with classical SVMs

Quantum-enhanced kernels  $\rightarrow$  potential for early quantum advantage.



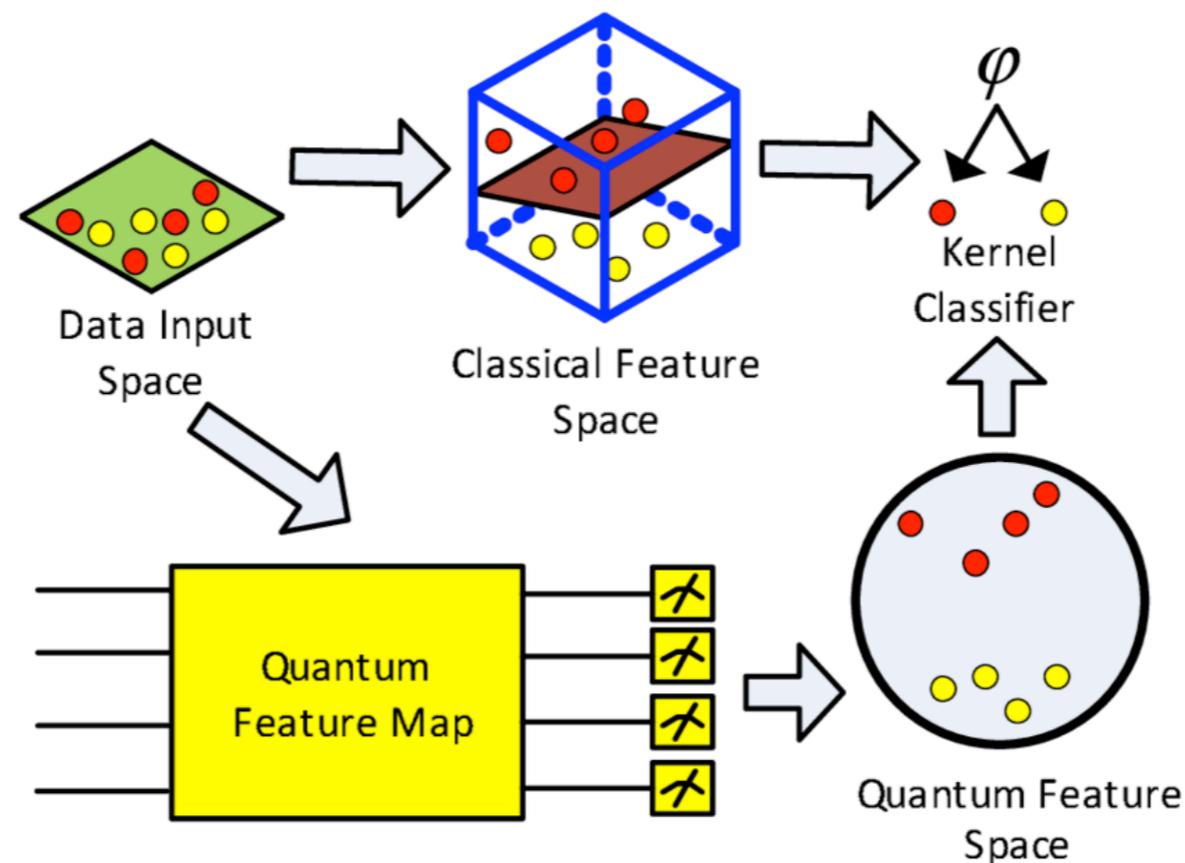
# QUANTUM KERNEL METHODS

We compute the kernel:

$$K(x, x') = |\langle \psi(x) | \psi(x') \rangle|^2$$

- Quantum computer estimates **state overlap**
- Classical SVM uses this matrix to classify data
- Useful for small datasets ( $n < 1,000$ )

**Idea:** quantum circuits generate richer kernels than classical ones.



# QISKIT EXAMPLE

```
▶ import numpy as np
from qiskit import QuantumCircuit
from qiskit.circuit.library import ZZFeatureMap
from qiskit.primitives import Sampler

# 1. Define a simple feature map (using 2 qubits for simplicity)
num_features = 2
feature_map = ZZFeatureMap(feature_dimension=num_features, reps=1)

# 2. Define two simple data points (vectors)
data_x = np.array([0.1, 0.2])
data_y = np.array([0.3, 0.4])

# 3. Create the quantum circuit to compute the overlap  $\langle \phi(x) | \phi(y) \rangle$ 
qc_y = feature_map.assign_parameters(data_y)
qc_overlap = feature_map.assign_parameters(data_x).inverse().compose(qc_y)

# 4. Use the Sampler primitive to run the circuit and get measurement probabilities
sampler = Sampler()

# Run the circuit with the sampler
job = sampler.run(qc_overlap, shots=1000) # Using 1000 shots for demonstration
result = job.result()

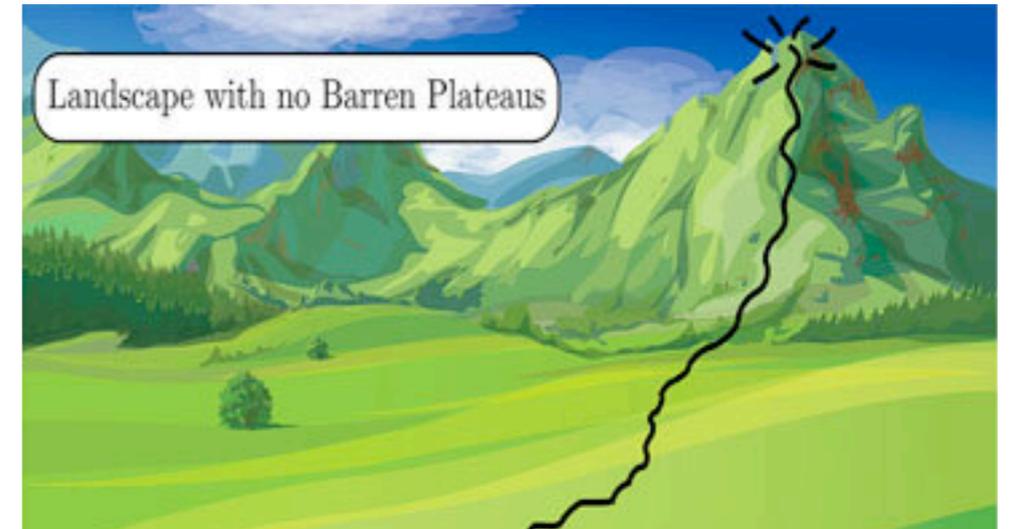
# Get the measurement probabilities
probabilities = result.get_counts(0)

# 5. Extract the probability of the all-zeros state ( $|00\rangle$ )
prob_zero = probabilities.get('0' * num_features, 0)
```

# OPTIMISATION CHALLENGE

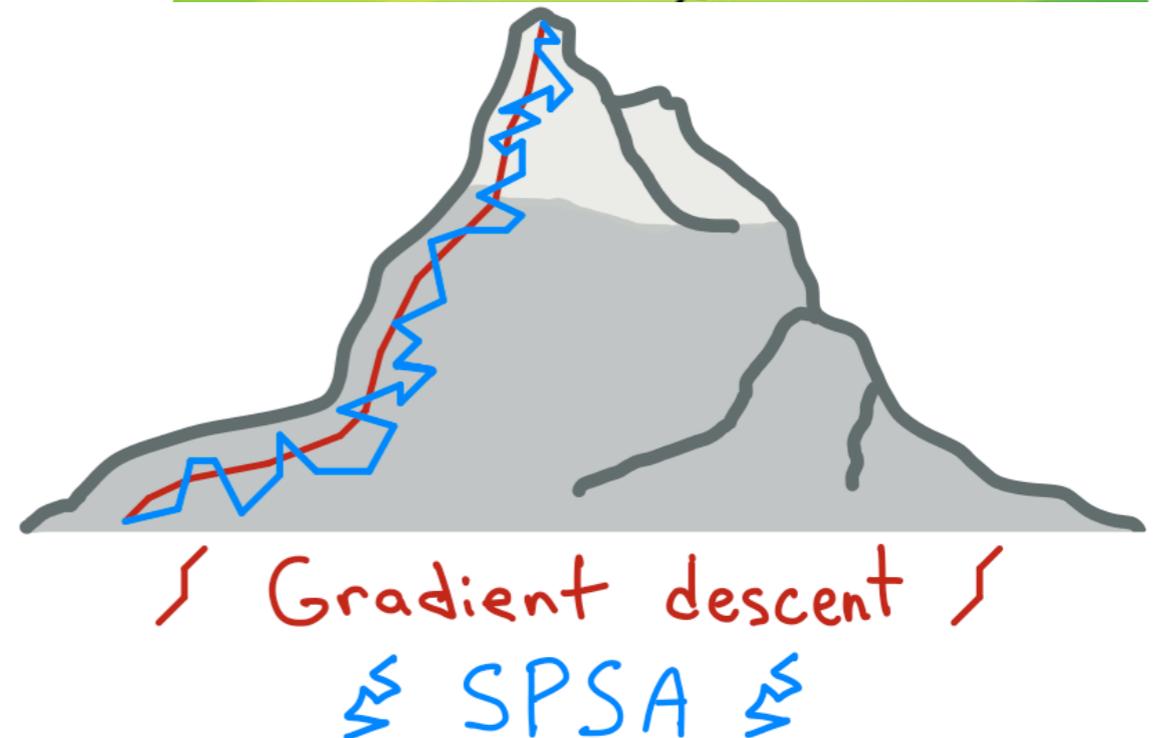
Variational models face issues:

- Vanishing gradients (“Barren plateaus”)
- Highly non-convex optimization landscapes
- Hardware noise makes gradients unreliable



Techniques:

- SPSA optimizer
- Problem-inspired ansatz
- Shallow circuits
- Parameter initialization strategies

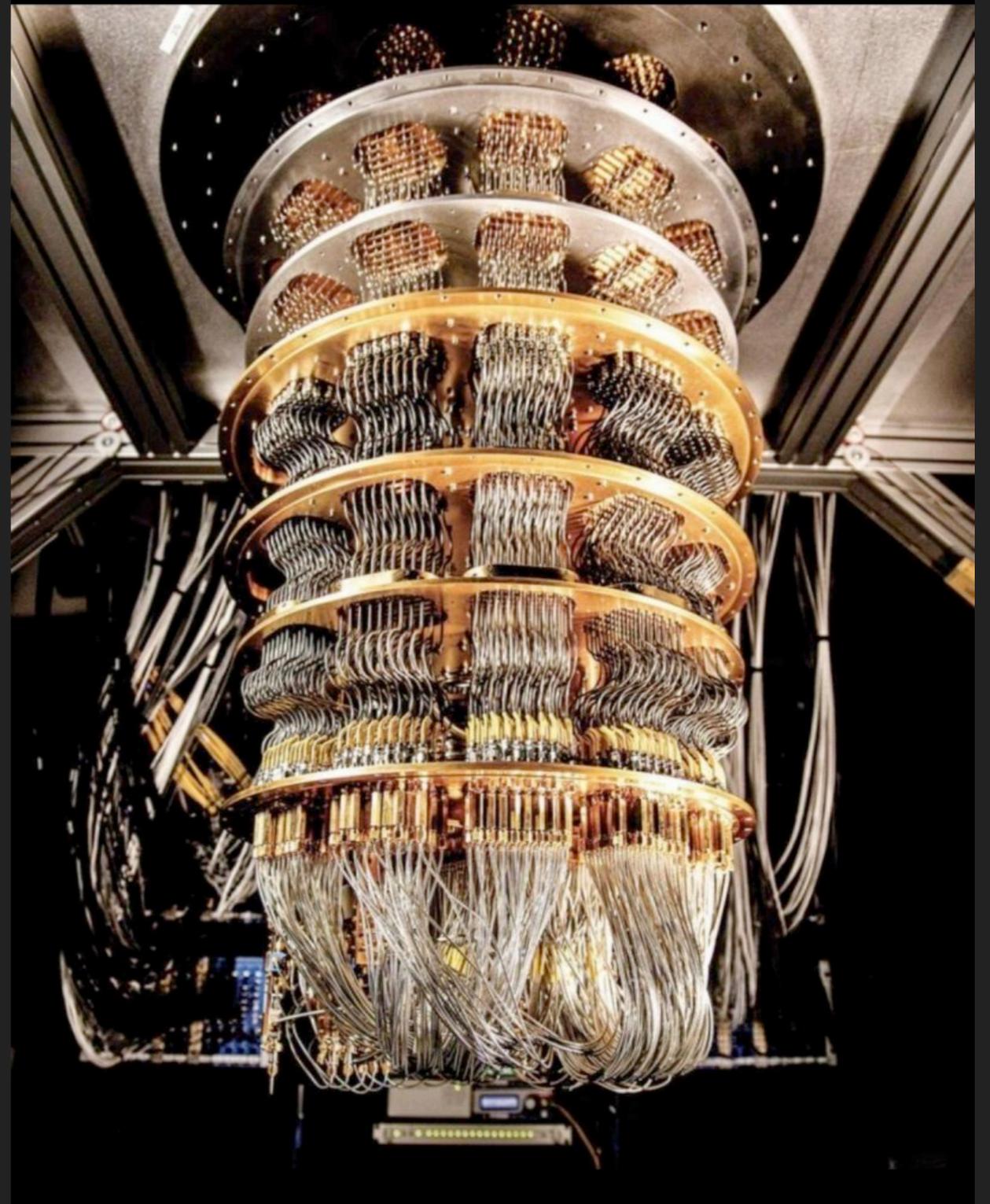


# WHAT WORKS TODAY ON REAL HARDWARE?

- ▶ Shallow variational circuits ( $< 6$  qubits)
- ▶ Quantum kernels with 2-4 qubits
- ▶ Hybrid pipelines using Sampler/Estimator
- ▶ Noisy simulations for prototyping

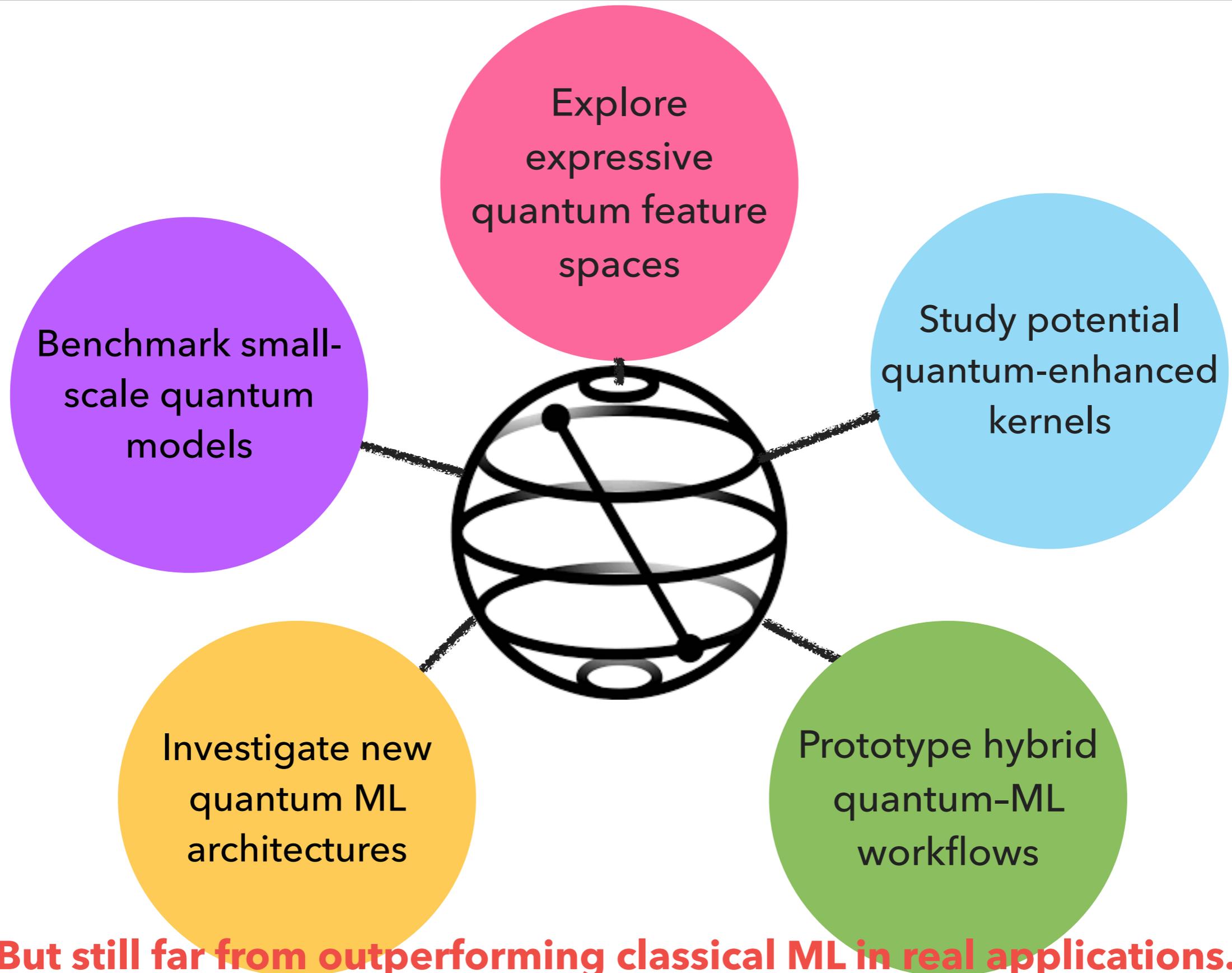
## Not yet feasible:

- Deep networks
- Large datasets
- Quantum advantage in real-world tasks (still research)



# WHAT QML CAN DO TODAY?

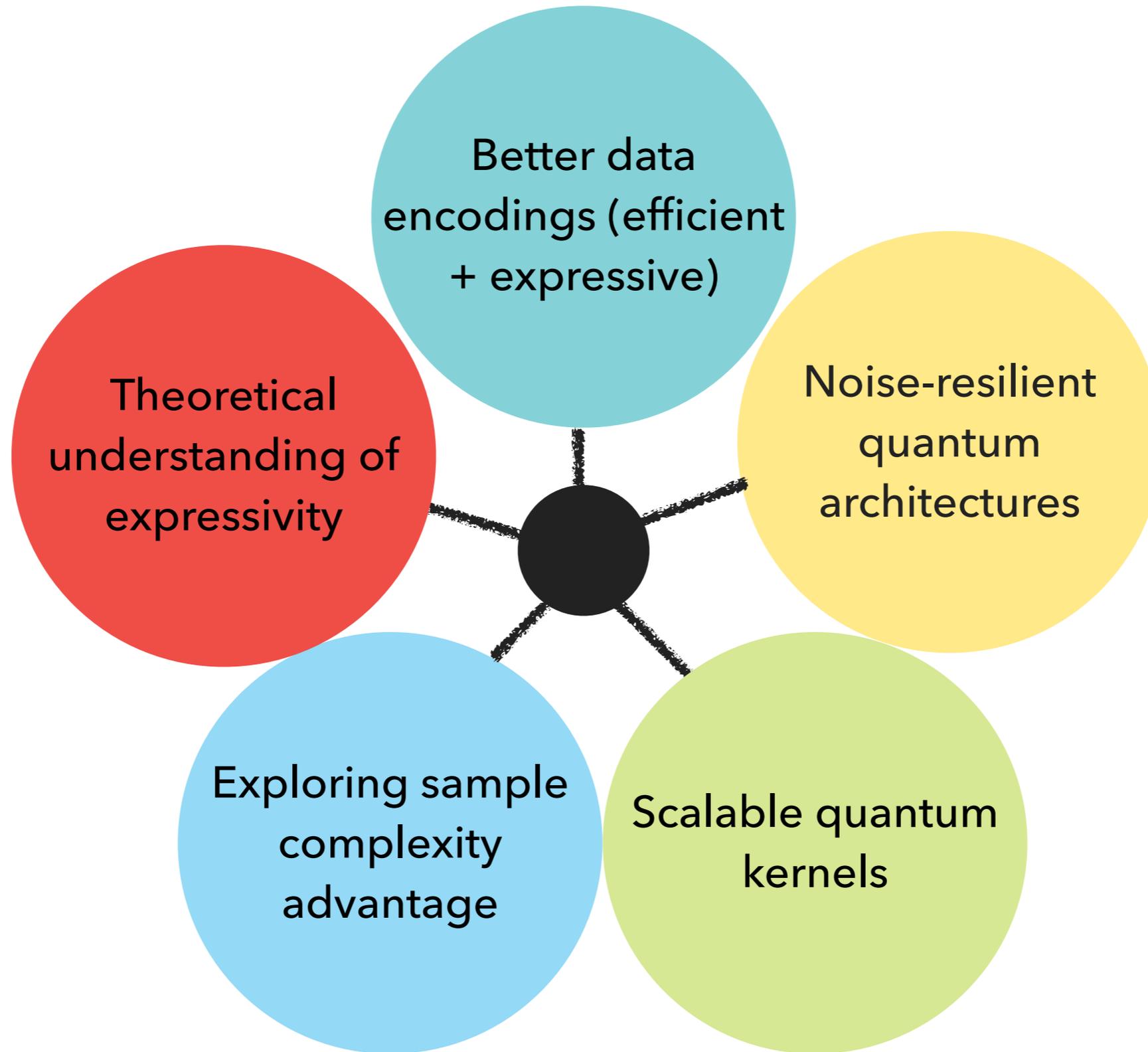
---



**But still far from outperforming classical ML in real applications.**

# FUTURE DIRECTIONS

---



# SUMMARY AND TAKE HOME MESSAGES

---

QML explores quantum circuits as ML models

Two main paradigms:

1. Variational models
2. Quantum kernel methods

Hybrid models are today's best option

Encoding is both powerful and challenging

Encoding is the bottleneck

Qiskit provides state-of-the-art primitives

QML is promising but in early research stage

# QISKIT RESOURCES

<https://quantum.cloud.ibm.com/>

The image displays the IBM Quantum Platform website interface, which is organized into several sections:

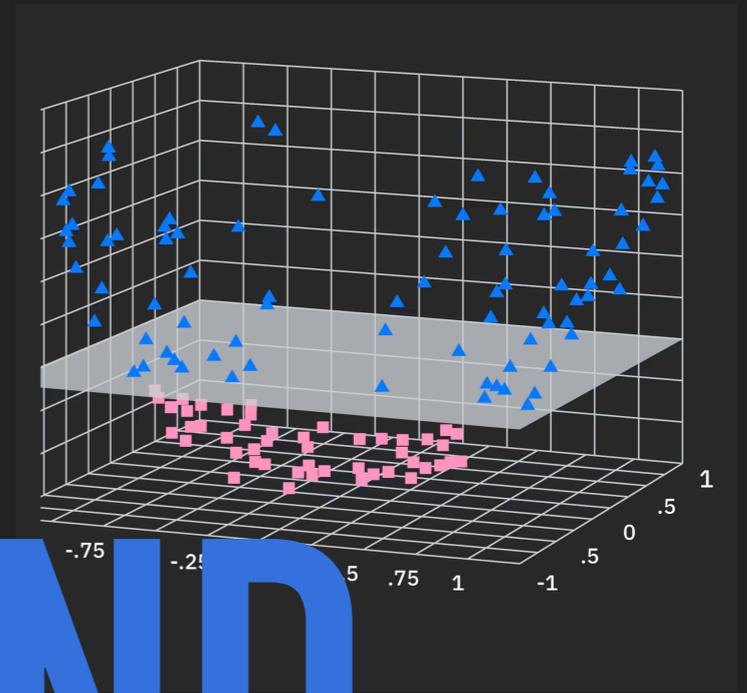
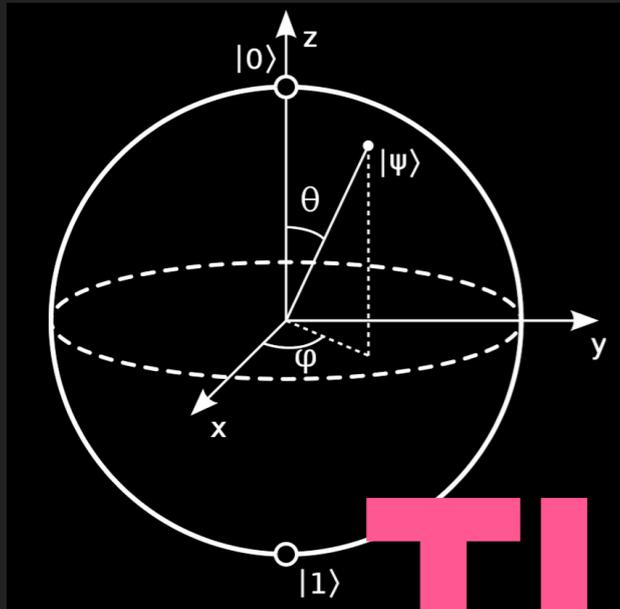
- Top Navigation:** Includes the IBM Quantum Platform logo, a search bar, and a "Sign in" button.
- Learning Paths:** A sidebar on the left lists "Quantum machine learning" and "Overview". The main content area features a large purple graphic with a book and a pencil, accompanied by the text: "This course explores how quantum computing can play an exciting role in machine learning. The course will offer some review of classical machine learning approaches, but will also assume some familiarity with these methods. The primary focus will be how quantum computing can fit into existing machine learning workflows. Topics include data encoding, quantum kernels, and..."
- Course Cards:** A grid of course cards is displayed, including:
  - Quantum Computing in Practice:** Learn potential use cases and best practices for experimenting with quantum processors having 100+ qubits. (5 hours)
  - Quantum Diagonalization Algorithms:** Explore quantum approaches to matrix diagonalization, including VQE, QKD, SKD and variations of these. (5 hours)
  - Quantum Machine Learning:** Learn to leverage the power of quantum computing in machine learning methods. (5 hours)
  - Variational Algorithm Design:** An overview of variational algorithms: hybrid classical quantum algorithms. (5 hours)
  - Quantum Chemistry with VQE:** (5 hours)
  - Utility-scale quantum computing:** (5 hours)
- Additional Resources:** A row of four cards covers "Superposition", "Uncertainty", "Quantum teleportation", and "Quantum key distribution".
- Build circuits with Composer:** A section with a visual representation of a quantum circuit and a link to "Explore composer".
- Additional resources:** Two cards for "Qiskit YouTube" (Browse lectures, tips and tricks, tutorials, community updates, and more) and "Tutorials" (Explore utility-grade examples that leverage IBM Quantum and Qiskit technologies).
- Quantum Circuit Simulator:** A detailed view of a quantum circuit with:
  - Operations Panel:** A grid of quantum gates including H, T, S, Z, T†, S†, P, RZ, |0⟩, |1⟩, if, √X, √X†, Y, RX, RY, RXX, RZZ, U, RCCX, and RC3X.
  - Qubit Lines:** Labeled q[0], q[1], q[2], q[3], and c4.
  - Code Editor:** Shows OpenQASM 2.0 code: 

```
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[4];
5 creg c[4];
6
7
```
  - Probabilities Panel:** A bar chart showing the probability distribution for various bit strings (0000 to 1111).
  - Q-sphere Panel:** A Bloch sphere visualization with a state vector and phase angle controls.

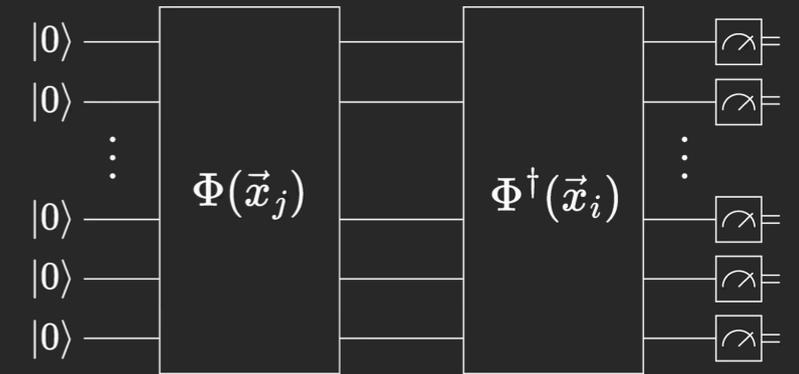
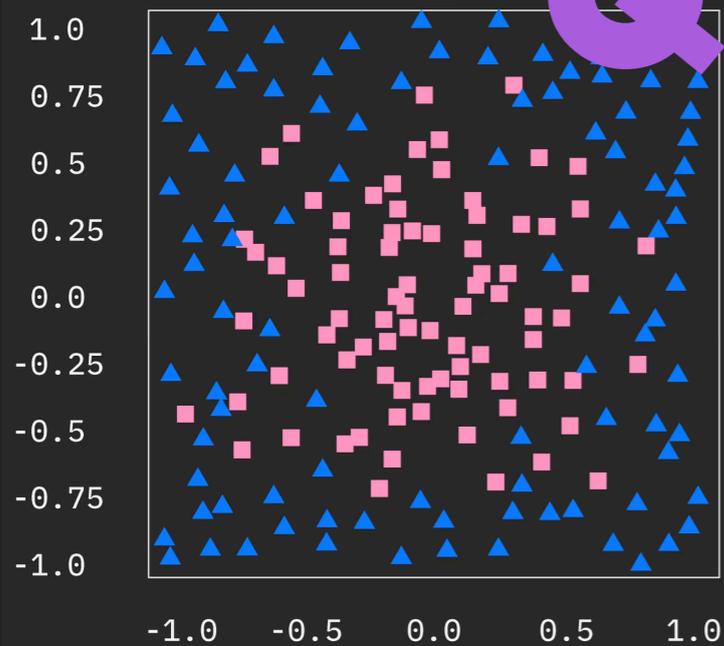
# APPLICATION OF QML IN ASTROPHYSICS AND COSMOLOGY

---

- ▶ "Exploring the potential of Quantum Graph Neural Networks in analysing large scale structure of Universe", **F.Farsian et al.**, Accepted for publication in Astronomy and Computing journal.
- ▶ "Comparing Quantum Machine Learning Approaches in Astrophysical Signal Detection", M. Ziatdinov, **F.Farsian et al.**, Accepted for publication to IEEE QSW 2025, [arXiv:2507.19505](https://arxiv.org/abs/2507.19505).
- ▶ "Benchmarking Quantum Convolutional Neural Networks for Signal Classification in Simulated Gamma-Ray Burst Detection", **F.Farsian et al.**, [arXiv:2501.17041](https://arxiv.org/abs/2501.17041).
- ▶ "The Application of Quantum Fourier Transform in Cosmic Microwave Background Data Analysis", **F.Farsian et al.**, Accepted for publication in IEEE QSW 2025, [arXiv:2505.15855](https://arxiv.org/abs/2505.15855).
- ▶ "Quantum Markov Chain Monte Carlo for Cosmological Functions", G. Sarracino et al, including F.Farsian, accepted for publication in IEEE International Conference on Quantum Artificial Intelligence 2025, [arXiv:2509.09395](https://arxiv.org/abs/2509.09395).
- ▶ "Comparing Classical and Quantum Deep Learning Techniques for Anomaly Detection of GRBs", A. Rizzo et al, including F.Farsian, Accepted for publication in Astronomy and Computing journal.
- ▶ "Numerical limits in the integration of Vlasov-Poisson equation for Cold Dark Matter", L. Cappelli et al, including F.Farsian, [arXiv:2503.22842](https://arxiv.org/abs/2503.22842).
- ▶ "Quantum Convolutional Neural Networks for the detection of Gamma-Ray Bursts in the AGILE space mission data", A. Rizzo et al, including F.Farsian, [arXiv:2404.14133](https://arxiv.org/abs/2404.14133).
- ▶ "A Quantum Genetic Algorithm for Cosmological Parameters Estimation", G. Sarracino et al, including F.Farsian, Accepted for publication in Astronomy & Computing journal.



# THANKS AND QUESTIONS



$$P_{|0\rangle} = |\langle 0 | \Phi^\dagger(\vec{x}_i) \Phi(\vec{x}_j) | 0 \rangle|^2$$

# PARAMETRISED QUANTUM CIRCUIT (PQC)

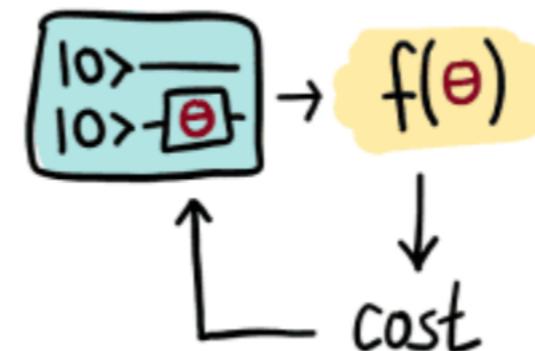
---

PQC bridge quantum and classical computing: the quantum computer estimates a quantity, while the classical computer optimizes the parameters. This process iterates, continually refining the quantum state.

They consist of three ingredients:

- ▶ Preparation of a fixed **initial state** (e.g., the vacuum state or the zero state).
- ▶ A quantum circuit  $U(\theta)$ , parameterized by a set of free parameters  $\theta$
- ▶ **Measurement** of an observable  $\hat{B}$  at the output.

trained by a classical optimization algorithm, by querying to the quantum device.



$$f(\theta) = \langle 0|U^\dagger(\theta)\hat{B}U(\theta)|0\rangle$$