

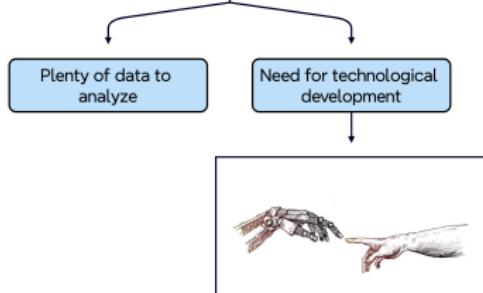
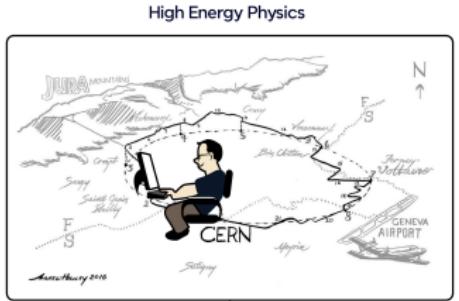
Quantum Machine Learning in HEP with Qibo

Matteo Robbiati

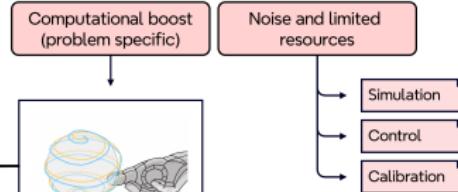
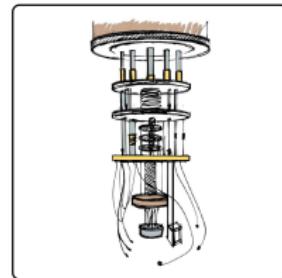
3 July 2023



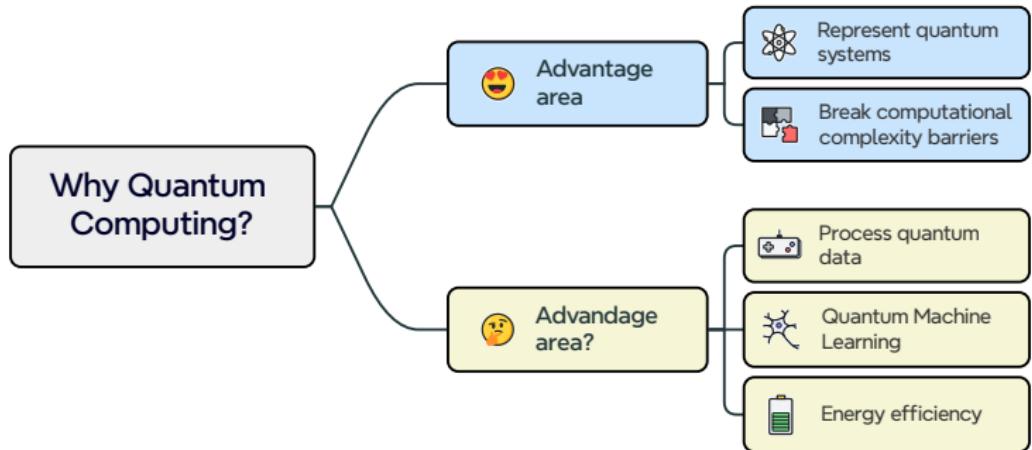
Quantum Computing



Quantum computers

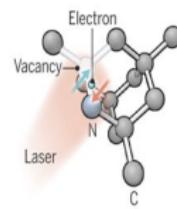
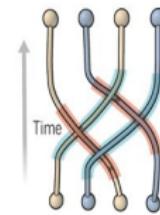
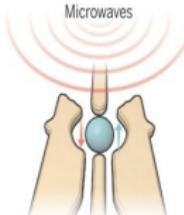
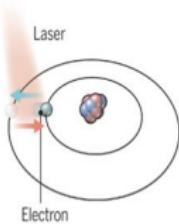
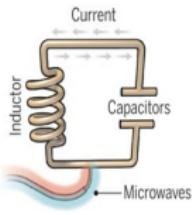


a more general “why?”



Qubits

1. classical bits are replaced by **qubits** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (quantum states).



Superconducting loops

A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into superposition states.

Trapped ions

Electrically charged atoms, or ions, have quantum energies that depend on the location of electrons. Tuned lasers cool and trap the ions, and put them in superposition states.

Silicon quantum dots

These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.

Topological qubits

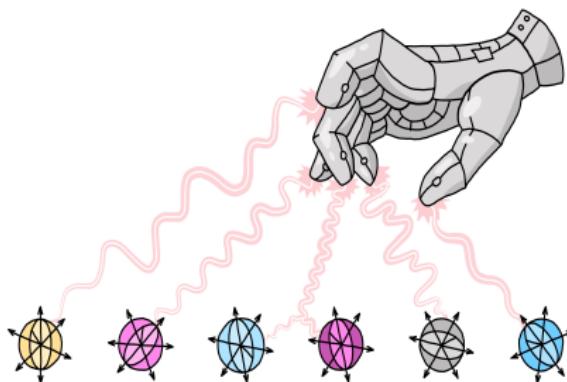
Quasiparticles can be seen in the behavior of electrons channeled through semiconductor structures. Their braided paths can encode quantum information.

Diamond vacancies

A nitrogen atom and a vacancy add an electron to a diamond lattice. Its quantum spin state, along with those of nearby carbon nuclei, can be controlled with light.

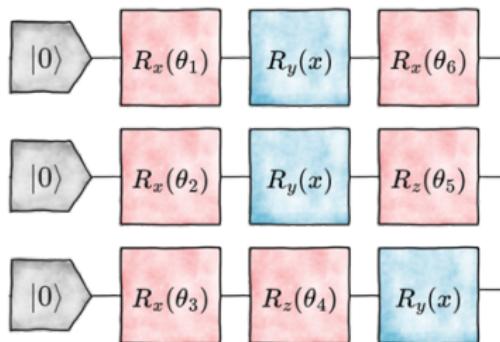
Qubits

1. classical bits are replaced by **qubits** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (quantum states).
 2. we can manipulate the qubit state applying **gates**: $|\psi'\rangle = \mathcal{U}(\theta)|\psi\rangle$.
- Typically we use 1-qubit and 2-qubits gates!



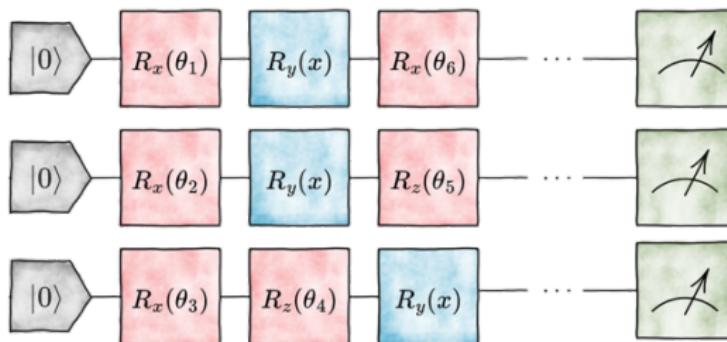
Qubits

1. classical bits are replaced by **qubits** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (quantum states).
2. we can manipulate the qubit state applying **gates**: $|\psi'\rangle = \mathcal{U}(\theta)|\psi\rangle$.
Typically we use 1-qubit and 2-qubits gates!
3. combine together gates to build **quantum circuits**;



Qubits

1. classical bits are replaced by **qubits** $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ (quantum states).
2. we can manipulate the qubit state applying **gates**: $|\psi'\rangle = \mathcal{U}(\theta)|\psi\rangle$.
Typically we use 1-qubit and 2-qubits gates!
3. combine together gates to build **quantum circuits**;
4. to access the information we need to measure the system.



Example 1: preparing entangled states

Example 1: preparing entangled states

With quantum computing, we introduce new tools.

- ☞ prepare a quantum state in the computational zero $|0\rangle$;
- ≡ we can prepare superposition:

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

Example 1: preparing entangled states

With quantum computing, we introduce new tools.

- 👉 prepare a quantum state in the computational zero $|0\rangle$;
- ➡ we can prepare superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{with} \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

- ⌚ let's apply a controlled-NOT (CNOT) gate on a second qubit prepared in $|0\rangle$:

$$\text{CNOT}\left(\underbrace{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)}_{\text{control}} \otimes |0\rangle\right) = \frac{1}{\sqrt{2}}(|00\rangle + \text{NOT}_{\text{targ}}|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

Example 1: preparing entangled states

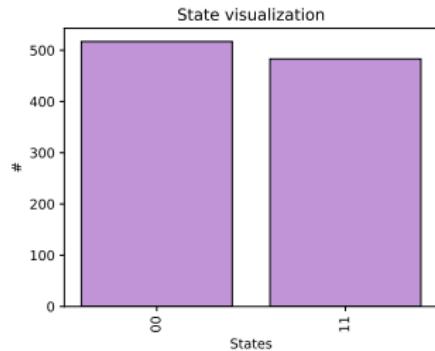
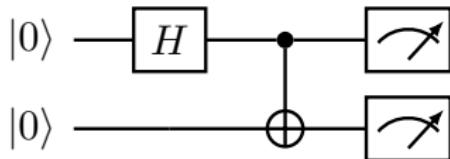
With quantum computing, we introduce new tools.

- 👉 prepare a quantum state in the computational zero $|0\rangle$;
- 👉 we can prepare superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{with} \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

- 👉 let's apply a controlled-NOT (CNOT) gate on a second qubit prepared in $|0\rangle$:

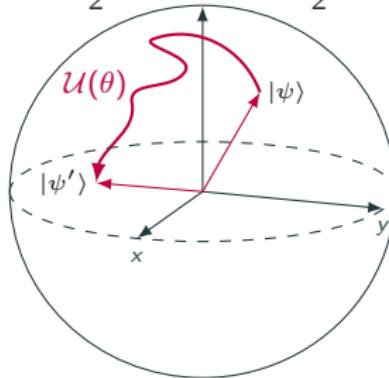
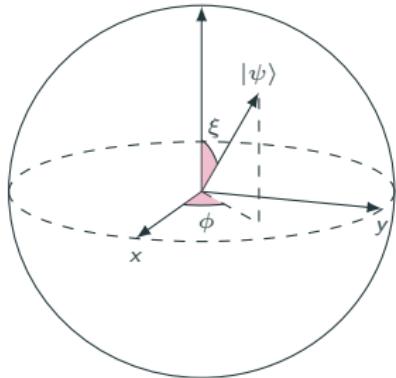
$$\text{CNOT}\left(\underbrace{\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)}_{\text{control}} \otimes |0\rangle\right) = \frac{1}{\sqrt{2}}(|00\rangle + \text{NOT}_{\text{targ}}|10\rangle) = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$



Parametric gates prepare variational quantum states

- Among the gates, parametric ones can be useful!
- Let's consider a single qubit system:

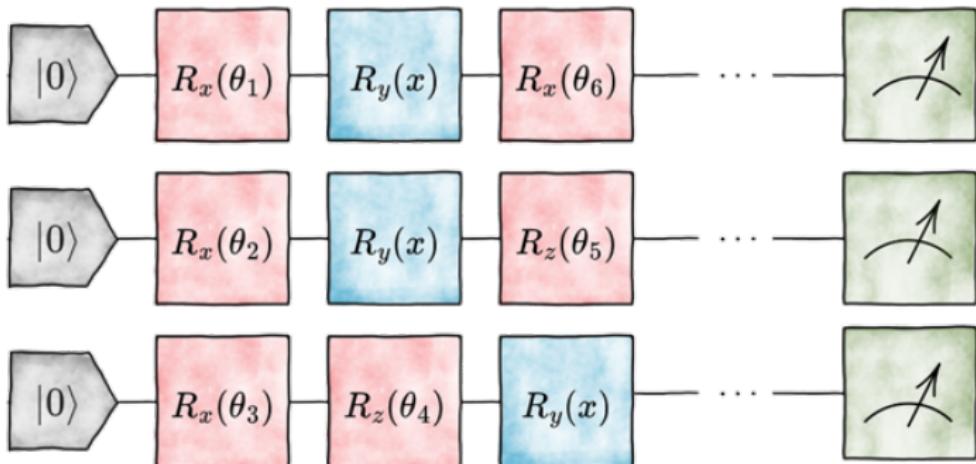
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \text{with} \quad \alpha = \cos \frac{\theta}{2}, \quad \beta = e^{i\phi} \sin \frac{\theta}{2}.$$



We can use as parametric gates the rotation around the axis of the block sphere:

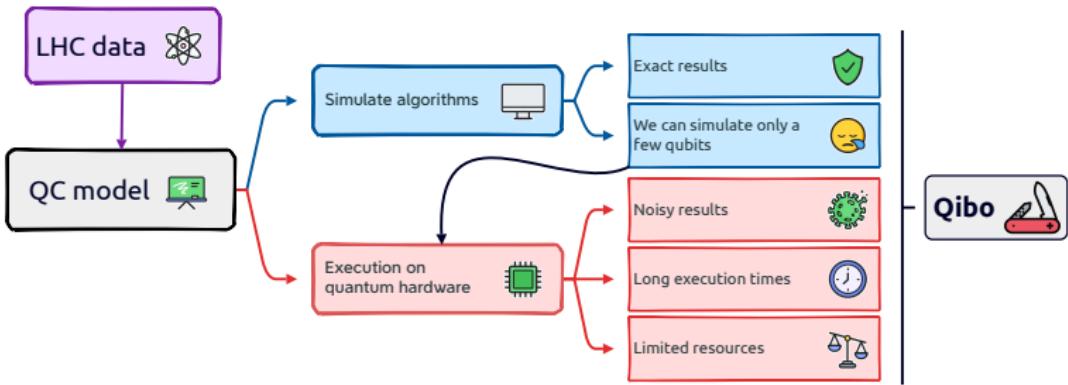
$$R_k(\theta) = \exp[-i\theta\sigma_k], \quad \text{with} \quad \sigma_k \in \{I, \sigma_x, \sigma_y, \sigma_z\}.$$

Parametric quantum circuits

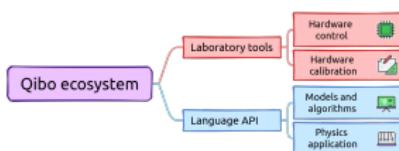


Qibo 0.2.8

What is needed for doing quantum computing?



Qibo is an **open-source** hybrid quantum operating system for self-hosted quantum computers.

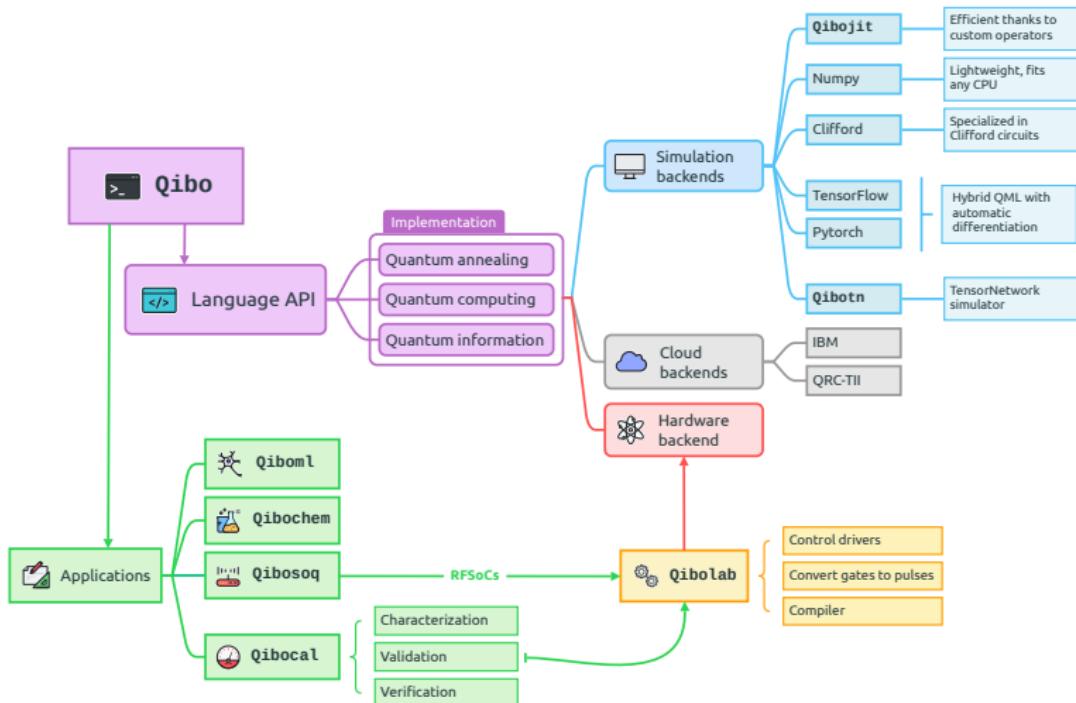


1. Fully open-source and community driven.
2. Modular layout design with possibility of adding:
 - new backends for simulation,
 - new platforms for hardware control,
 - new drivers for control electronics.
3. Supported by documentation and tests/CI on quantum hardware.



<https://qibo.science>

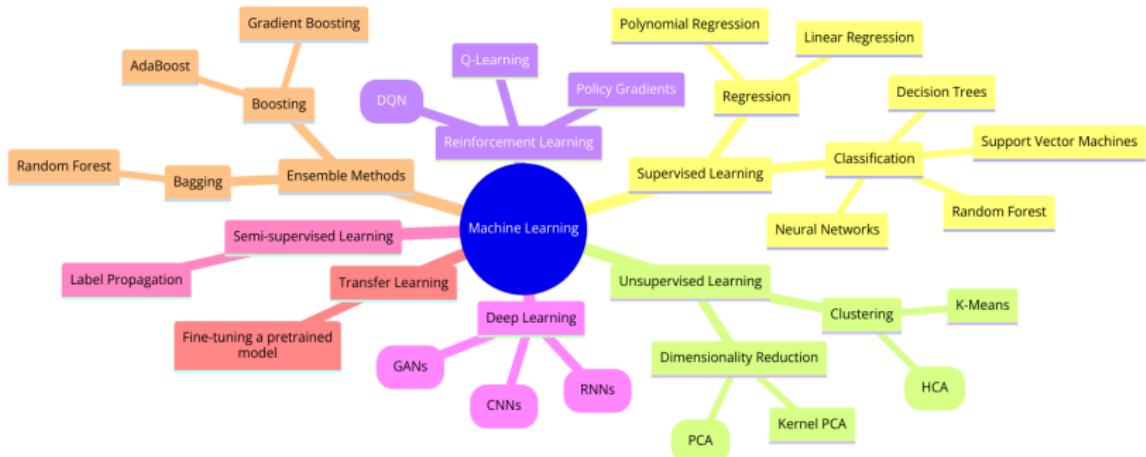
An open source and full-stack framework for quantum computing



Machine Learning

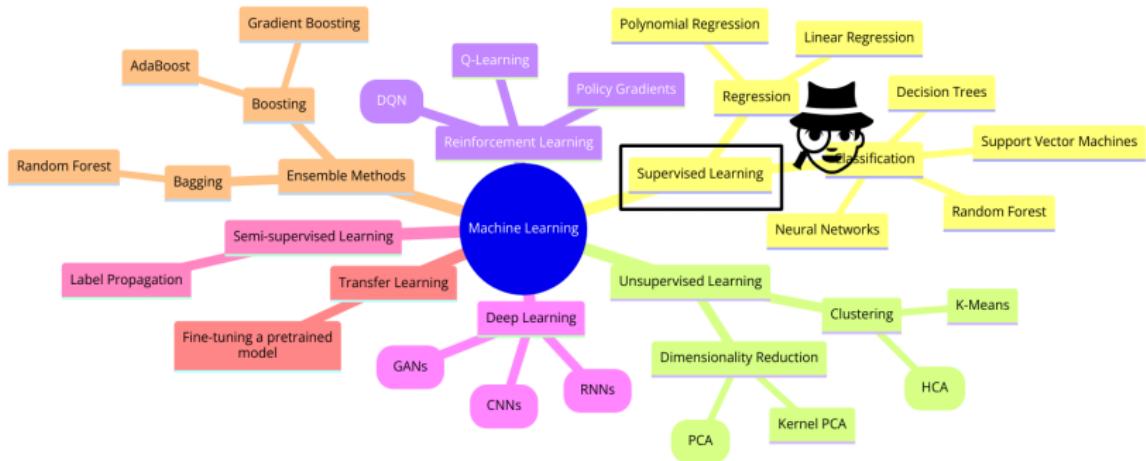
Classical Machine Learning

I asked ChatGPT to give me a comprehensive diagram of Machine Learning (ML) models.



Classical Machine Learning

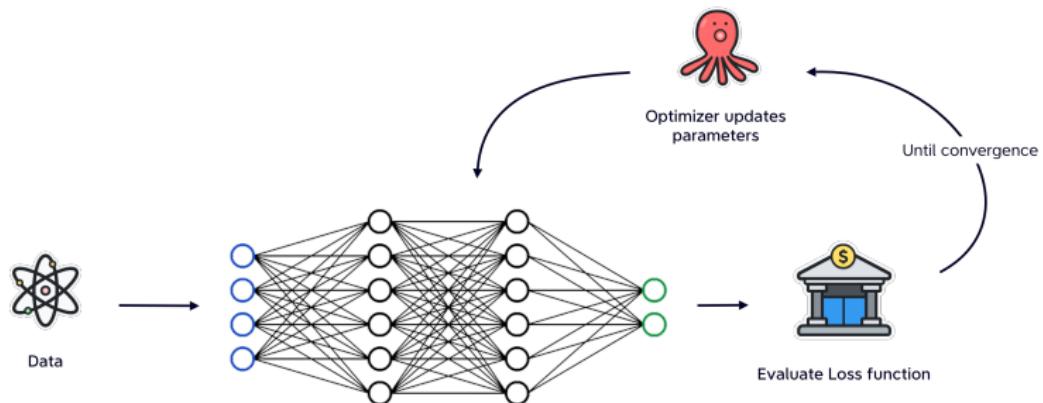
I asked ChatGPT to give me a comprehensive diagram of Machine Learning (ML) models.



Focusing on the supervised ML!

- ❖ we aim to know some hidden law between two variables: $y = f(x)$;
- 📊 we define a parameteric model which returns $\hat{y}_{\text{est}} = f_{\text{est}}(x; \theta)$;
- 🔭 we define an optimizer, which task is to compute $\operatorname{argmin}_{\theta} [J(y_{\text{meas}}, \hat{y}_{\text{est}})]$.

Classical Machine Learning



Quantum Machine Learning

Machine Learning

\mathcal{M} : model;

\mathcal{O} : optimizer;

\mathcal{J} : loss function.

(x, y) : data

Quantum Computation

\mathcal{Q} : qubits;

\mathcal{S} : superposition;

\mathcal{E} : entanglement.

Quantum Machine Learning - operating on qubits

Machine Learning

\mathcal{M} : model;

\mathcal{O} : optimizer;

\mathcal{J} : loss function.

(x, y) : data

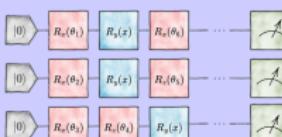
Quantum Computation

\mathcal{Q} : qubits;

\mathcal{S} : superposition;

\mathcal{E} : entanglement.

Circuit execution



Machine Learning

\mathcal{M} : model;

\mathcal{O} : optimizer;

\mathcal{J} : loss function.

(x, y) : data

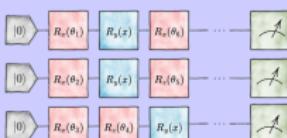
Quantum Computation

\mathcal{Q} : qubits;

\mathcal{S} : superposition;

\mathcal{E} : entanglement.

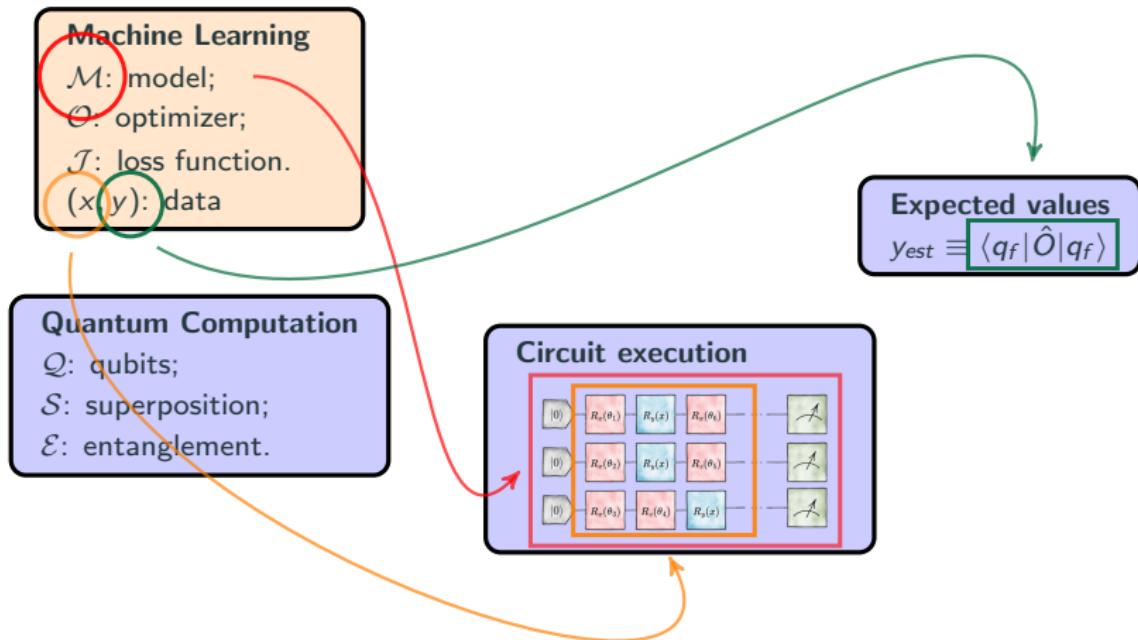
Circuit execution



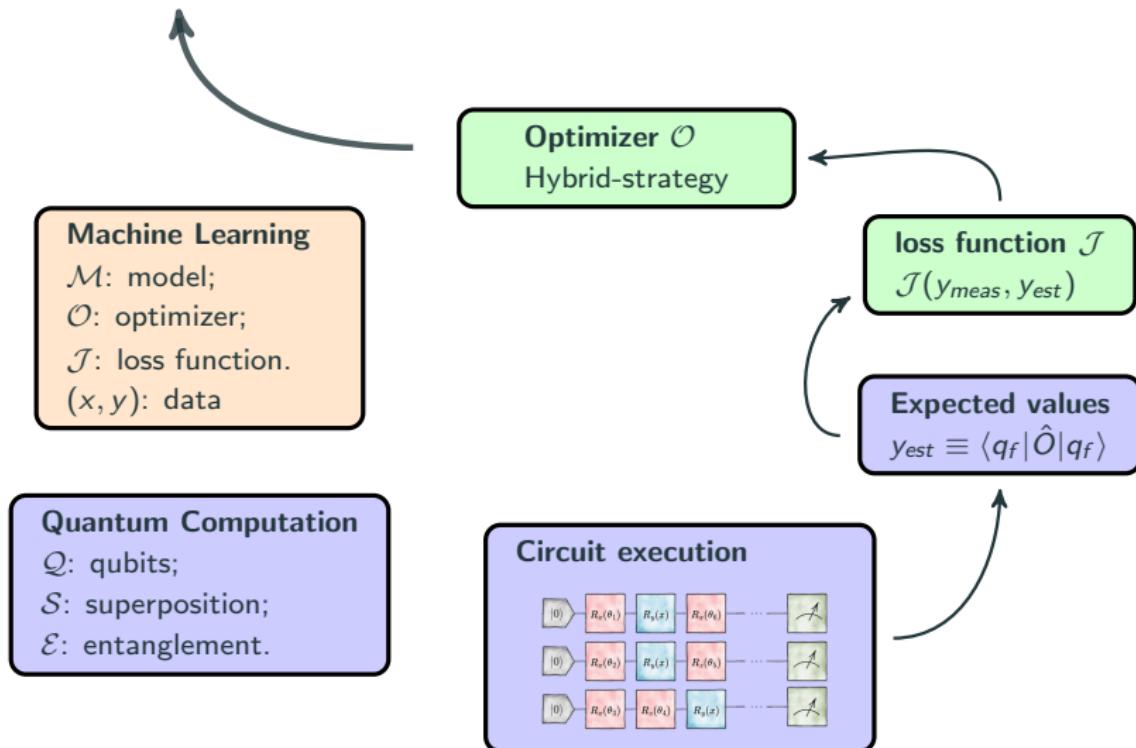
Expected values

$$y_{est} \equiv \langle q_f | \hat{O} | q_f \rangle$$

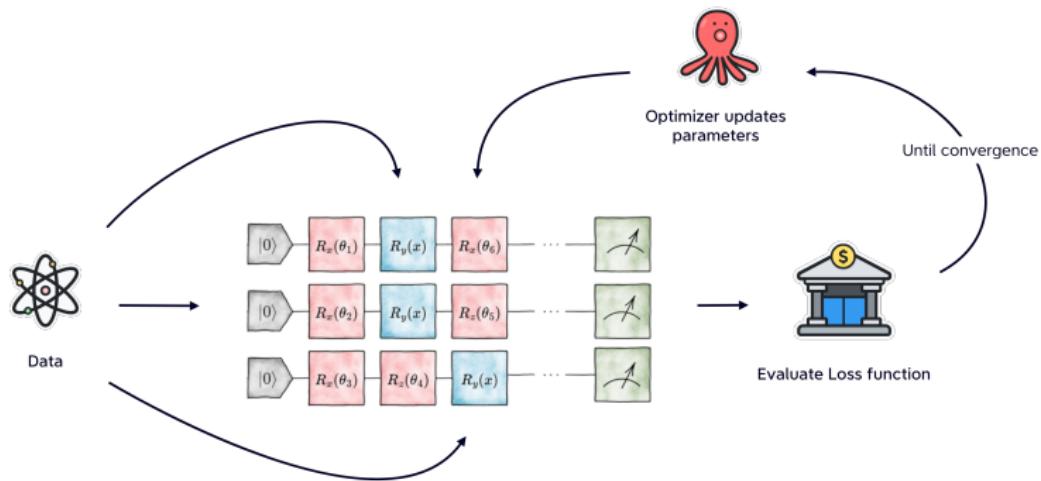
Quantum Machine Learning - encoding the problem



Quantum Machine Learning!



From ML to QML



We parametrize **Parton Distribution Functions** with multi-qubit variational quantum circuits:

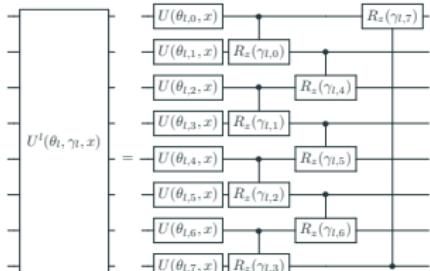
1. Define a quantum circuit:

$$\mathcal{U}(\theta, x)|0\rangle^{\otimes n} = |\psi(\theta, x)\rangle$$

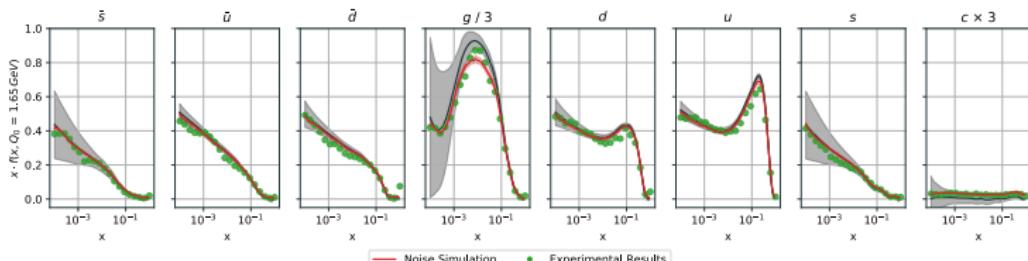
2. $U_w(\alpha, x) = R_z(\alpha_3 \log(x) + \alpha_4)R_y(\alpha_1 \log(x) + \alpha_2)$

3. Using $z_i(\theta, x) = \langle \psi(\theta, x) | Z_i | \psi(\theta, x) \rangle$:

$$\text{qPDF}_i(x, Q_0, \theta) = \frac{1 - z_i(\theta, x)}{1 + z_i(\theta, x)}.$$



Results from **classical quantum simulation and hardware execution** (IBM) are promising:



High level API: Qibo

- </> define **prototypes** and **models**;
- </> simulate training and noise.

Calibration: Qibocal

- ❖ calibrate qubits;
- ❖ generate **platform configuration**;

Execution: Qibolab

- ⚙️ allocate **calibrated** platform;
- ⚙️ **compile** and **transpile** circuits;
- ⚙️ execute and return **results**.

High level API: Qibo

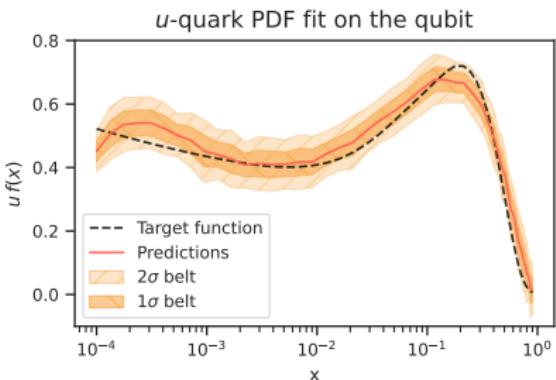
- </> define **prototypes** and models;
- </> simulate training and noise.

Calibration: Qibocal

- ❖ calibrate qubits;
- ❖ generate **platform configuration**;

Execution: Qibolab

- ⚙️ allocate **calibrated** platform;
- ⚙️ compile and **transpile** circuits;
- ⚙️ execute and return **results**.



Parameter	Value
N_{data}	50
N_{shots}	500
MSE	$\sim 10^{-3}$
Electronics	Xilinx ZCU216
Training time	$\sim 2\text{h}$

Some applications

- Multi-variable integration using the qPDF ansatz
arXiv:2303.11346;
- Real-time quantum error mitigation on superconducting devices to improve trainability,
arXiv:2303.11346;

