Quantum Machine Learning challenges with `Qibo`

Second year PhD school workshop

Matteo Robbiati[†] on behalf of the Qibo team[‡]

9 September 2024

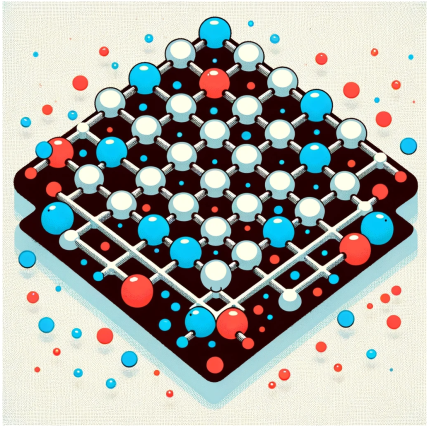
[†] PhD candidate, University of Milan, Italy and CERN, Switzerland.
matteo.robbiati@cern.ch

[‡] *https : / / qibo . science /*

Compute quantum mechanics

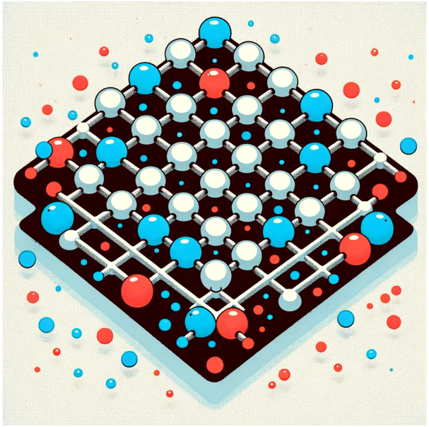
Compute quantum mechanics

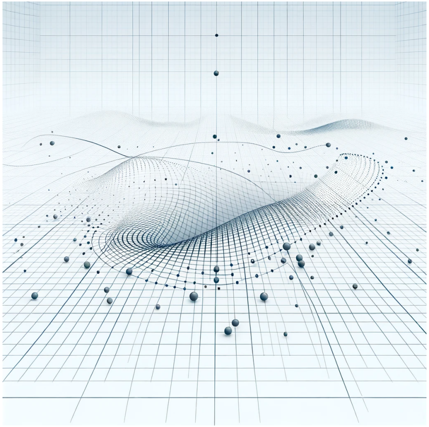
- Representing N particles is difficult;



Compute quantum mechanics

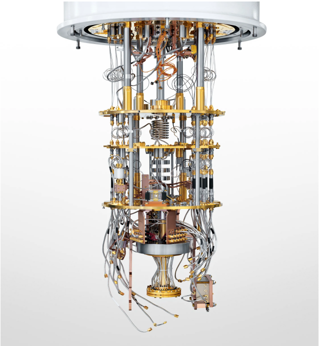
- Representing N particles is difficult;
- considering N spins $(\uparrow, \downarrow)$, we deal with a 2^N dimensional Hilbert space!





What can we do?

1. we can try to use classical methods to represent the system;
2. we can build a quantum computer.

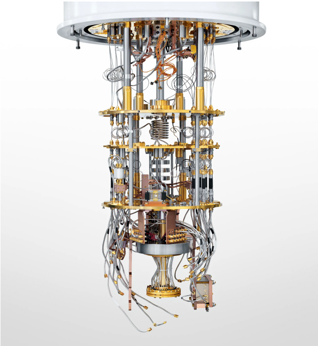


Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

Richard Feynman, 1982, Simulating Physics with Computers

What can we do?

1. **we can try to use classical methods to represent the system**;
2. we can build a quantum computer.

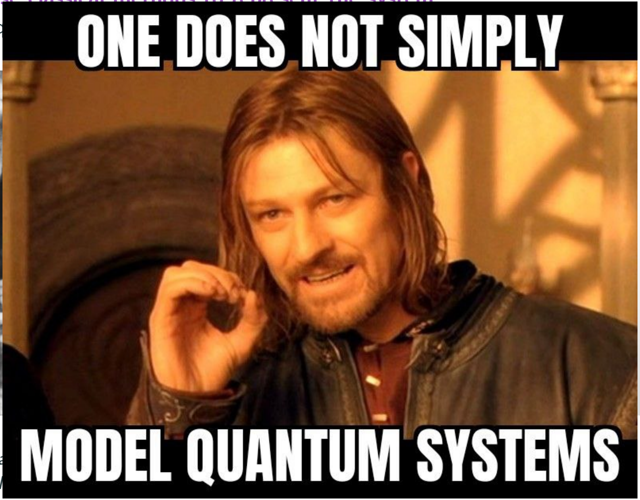


Nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

Richard Feynman, 1982, Simulating Physics with Computers

What can we do?

1. **we can try to use classical methods to represent the system;**
2. we can build a c



Nature isn't classical, da *quantum mechanical, and by golly it's a wonderful*

Richard Feynman, 1982, Simulating Physics with Computers

Can we represent a state with a classical computer?

Let's suppose we want to represent a system of **qubits** $(\uparrow, \downarrow)$.

1. Each `float 64` requires 8 bytes of memory to be stored;
2. each `complex 128` requires 16 bytes of memory;
3. let's take a nice 32Gb of RAM: it can store up to 2 billions of `complex 128`.
4. a 30 qubits state requires ~ 1 billion of complex numbers;
5. a 31 qubits state cannot be represented by my PC;

Can we represent a state with a classical computer?

Let's suppose we want to represent a system of **qubits** $(\uparrow, \downarrow)$.

1. Each `float 64` requires 8 bytes of memory to be stored;
2. each `complex 128` requires 16 bytes of memory;
3. let's take a nice 32Gb of RAM: it can store up to 2 billions of `complex 128`.
4. a 30 qubits state requires ~ 1 billion of complex numbers;
5. a 31 qubits state cannot be represented by my PC;
6. no problem. Let's get serious: `Fugaku`!

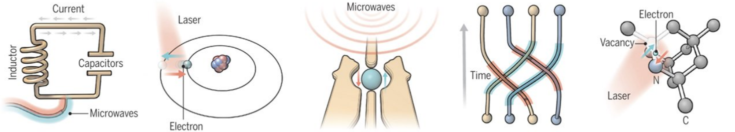
Can we represent a state with a classical computer?

Let's suppose we want to represent a system of **qubits** $(\uparrow, \downarrow)$.

1. Each `float 64` requires 8 bytes of memory to be stored;
2. each `complex 128` requires 16 bytes of memory;
3. let's take a nice 32Gb of RAM: it can store up to 2 billions of `complex 128`.
4. a 30 qubits state requires ~ 1 billion of complex numbers;
5. a 31 qubits state cannot be represented by my PC;
6. no problem. Let's get serious: `Fugaku`!

Gate based quantum computing

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.

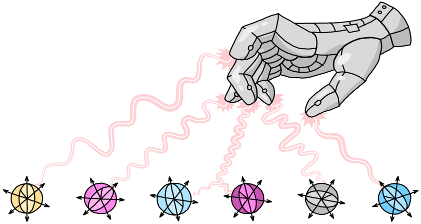
Silicon quantum dots
These "artificial atoms" are made by adding an electron to a small piece of pure silicon. Microwaves control the electron's quantum state.

Topological qubits
Quasiparticles can be seen in the behavior of electrons channeled through semi-conductor 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.

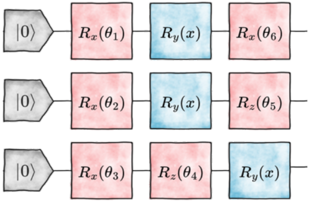
Gate based quantum computing

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**: $|\psi'\rangle = \mathcal{U}(\boldsymbol{\theta})|\psi\rangle$.
   Typically we use 1-qubit and 2-qubits gates!



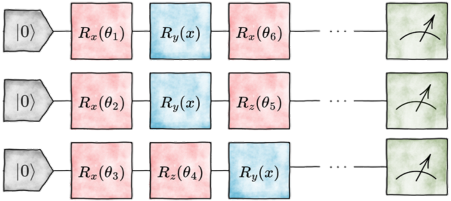
Gate based quantum computing

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**: $|\psi'\rangle = \mathcal{U}(\boldsymbol{\theta})\,|\psi\rangle$.
   Typically we use 1-qubit and 2-qubits gates!
3. combine together gates to build **quantum circuits**;



Gate based quantum computing

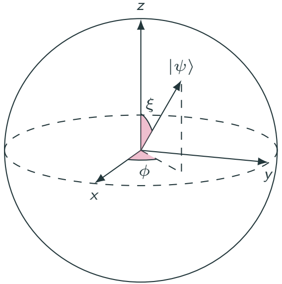
1. classical bits are replaced by **qubits** $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ (quantum states).
2. we can manipulate the qubit state applying **gates**: $|\psi'\rangle = \mathcal{U}(\boldsymbol{\theta})\,|\psi\rangle$.
   Typically we use 1-qubit and 2-qubits gates!
3. combine together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

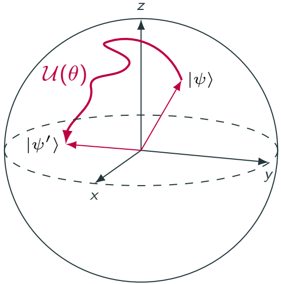


Parametric gates prepare variational quantum states

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

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



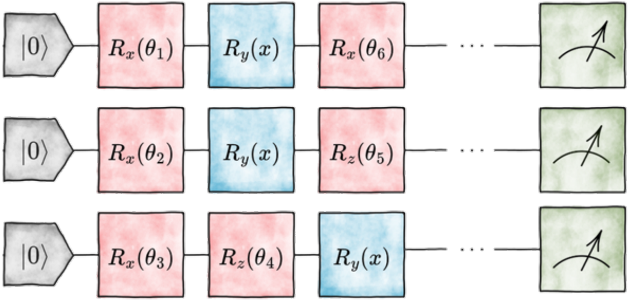


We can use as parametric gates the rotation around the axis of the block sphere:

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

Parametric quantum circuits

Parametric gates can be used to build parametric quantum circuits.



Superconducting qubits in a nutshell

We start from a simple LC circuit

Let's consider an LC circuit.

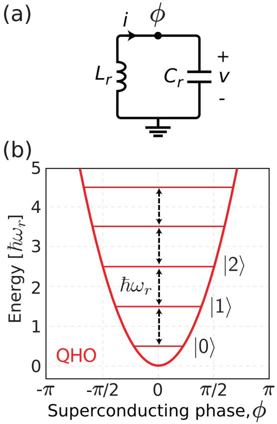
1. The system energy is accumulated in C and L:

$$H = \frac{1}{2}CV^2 + \frac{1}{2}LI^2$$

2. and since $Q = CV$, $\Phi = LI$ and $\omega_0 = 1/\sqrt{LC}$:

$$H = \frac{Q^2}{2C} + \frac{1}{2}C\omega_0^2\Phi^2 \quad \leftrightarrow \quad H = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2.$$

3. writing Q and Φ in terms of their quantum operators, we construct a quantum harmonic oscillator (QHO) which satisfies $H_{LC}\left|n\right\rangle = E_n\left|n\right\rangle$ with $E_n = \hbar\omega_0\left(n + 1/2\right)$ and $\Delta E = E_n - E_{n_1} = \hbar\omega_0$ for each n.



Having equispaced energies can be a problem if we aim to isolate **only two levels**.

Adding nonlinearity to our circuit

We aim to break the linearity and introduce anharmonicity in the system ($\omega_{01} \neq \omega_{12}$).

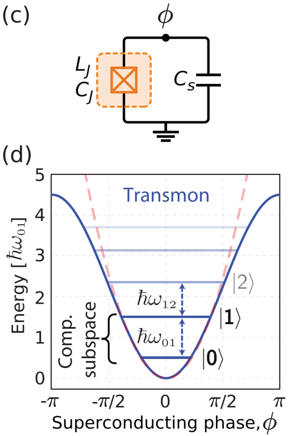
1. L is replaced with a **Josephson junction**: a tiny insulator layer between the superconducting wires.
2. J generates a non linear potential:

$$H_J = \frac{Q^2}{2C_J} - E_J \cos \Phi.$$

3. Q is usually expressed in terms of excess of Cooper pairs in the junction (superconductivity).
4. If we finally expand the potential in power series:

$$E_J \cos \Phi = \frac{1}{2} E_J \Phi^2 - \frac{1}{24} E_J \Phi^4 + O(\Phi^6),$$

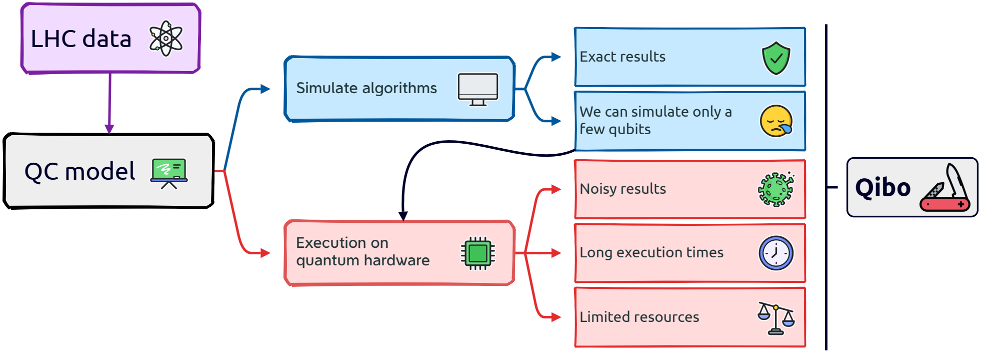
where the quartic term implies anharmonicity.



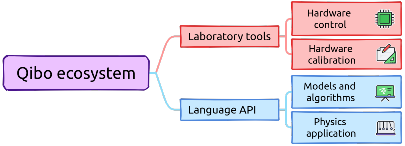
The anharmonicity cannot be arbitrarly large because of the sign of the quartic term: $\alpha = \omega_{12} - \omega_{01} < 0$.

Qibo 0.2.9

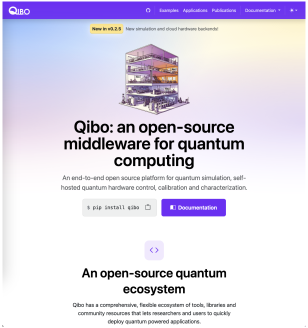
What is needed for doing quantum computing?



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

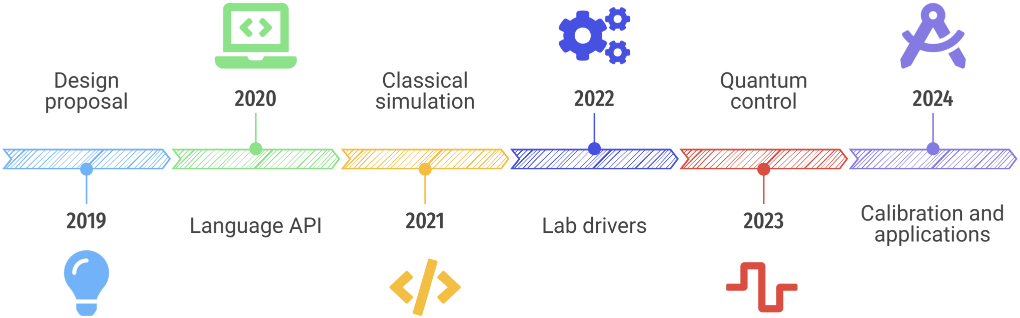


1. Fully open-source and community driven.
2. Modular layout design with possibility of adding:
   - new backends for simulation,
   - new platforms for hardware control,
   - new drivers for control electronics.
3. Supported by documentation and tests/CI on quantum hardware.



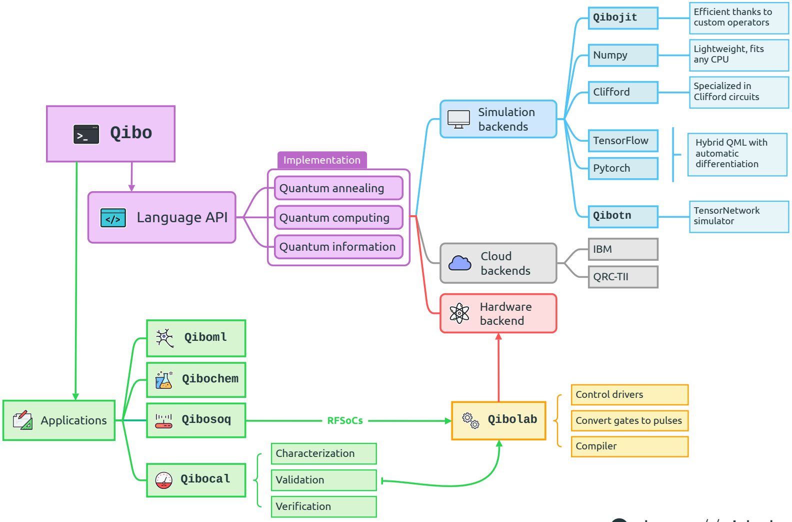
https://qibo.science

The Qibo timeline



An open source and full-stack framework for QC

arXiv:2308.06313



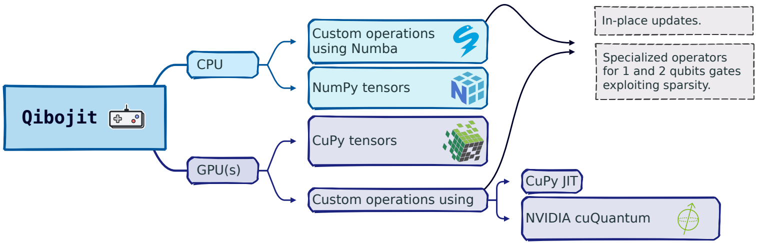
https://github.com/qiboteam

State vector simulation solves:

$$\psi'(\sigma_1, \dots, \sigma_n) = \sum_{\boldsymbol{\tau}'} G(\boldsymbol{\tau}, \boldsymbol{\tau}')\psi(\sigma_1, \dots, \boldsymbol{\tau}', \dots, \sigma_n)$$

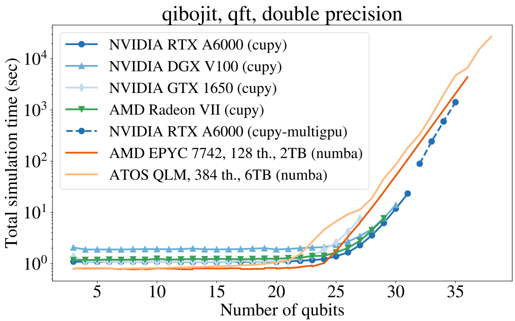
The number of operations scales exponentially with the number of qubits.

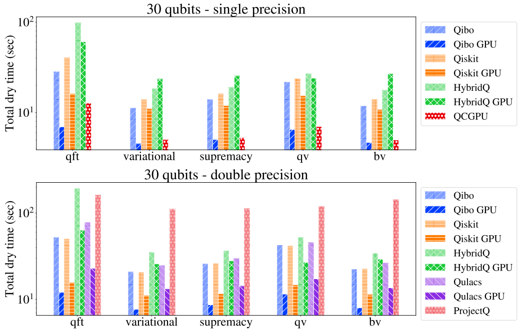
Qibo uses just-in-time technology and hardware acceleration:



A focus on classical simulation performances

Through its modularity, `Qibo` allows execution of the same high level language onto different classical hardware accellerators.

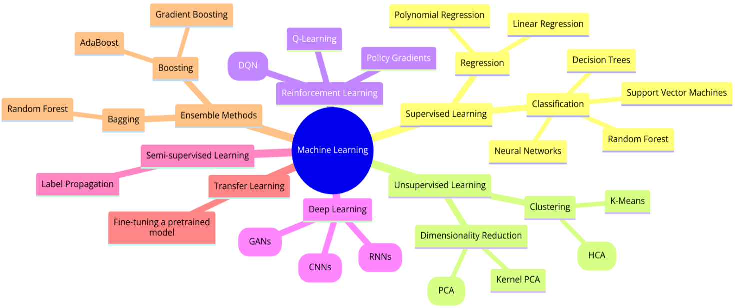




We reach satisfying performances thanks to custom operators and in-place updates of the statevector.

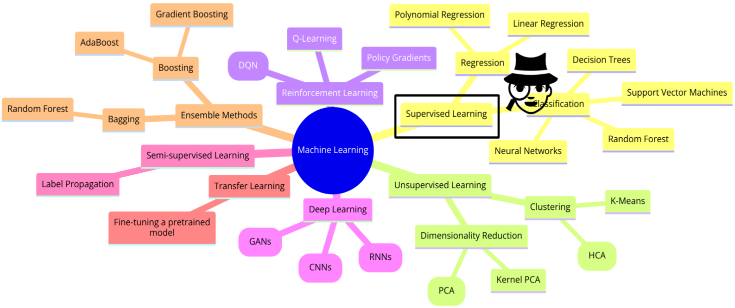
Quantum Machine Learning

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



Classical Machine Learning

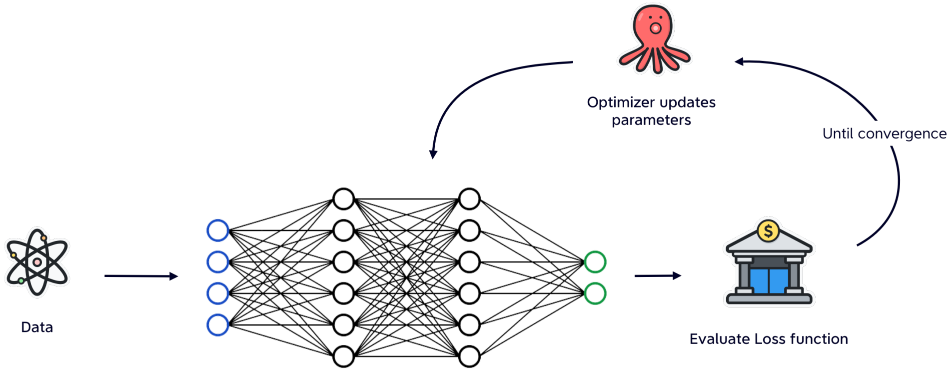
I asked ChatGPT to give me a comprehensive diagram of Machine Learning (ML) models.



Focusing on the supervised ML!

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

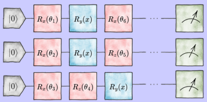
Classical Machine Learning



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.

Circuit execution



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

Machine Learning

$\mathcal{M}$: model;

$\mathcal{O}$: optimizer;

$\mathcal{J}$: loss function.

(x, y): data

Quantum Computation

$\mathcal{Q}$: qubits;

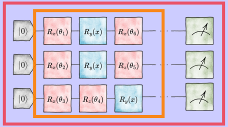
$\mathcal{S}$: superposition;

$\mathcal{E}$: entanglement.

Expected values

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

Circuit execution



Quantum Machine Learning!

Optimizer $\mathcal{O}$
Hybrid-strategy

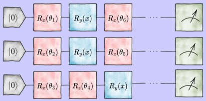
Machine Learning
$\mathcal{M}$: model;
$\mathcal{O}$: optimizer;
$\mathcal{J}$: loss function.
(x, y): data

loss function $\mathcal{J}$
$\mathcal{J}(y_{meas}, y_{est})$

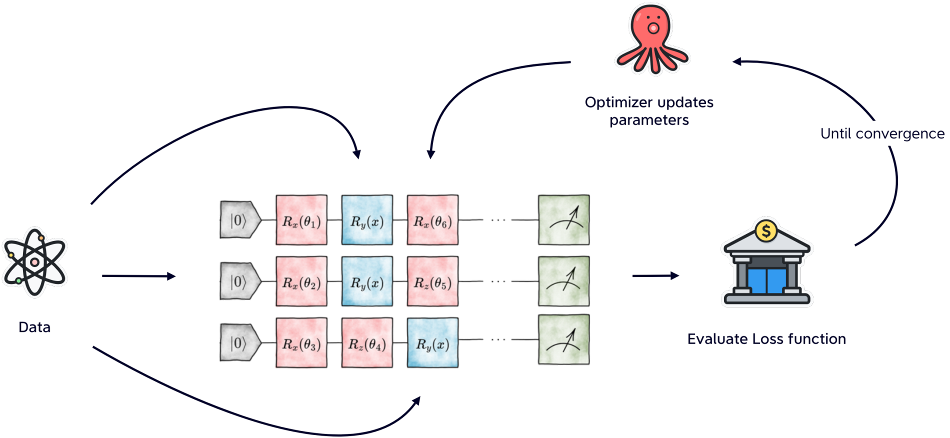
Expected values
$y_{est} \equiv \langle q_f | \hat{O} | q_f \rangle$

Quantum Computation
$\mathcal{Q}$: qubits;
$\mathcal{S}$: superposition;
$\mathcal{E}$: entanglement.

Circuit execution



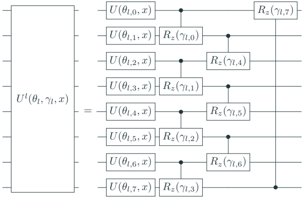
From ML to QML



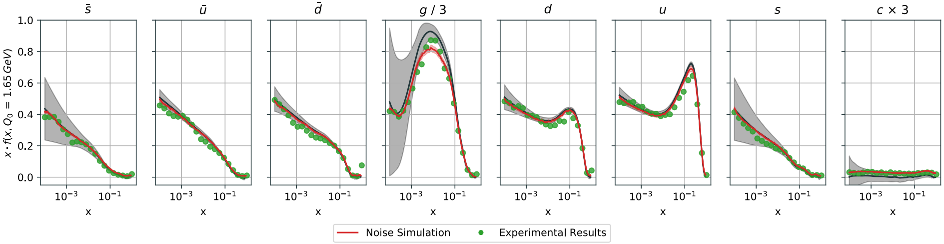
We parametrize **Parton Distribution Functions** with multi-qubit variational quantum circuits:

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

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



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



High level API: Qibo

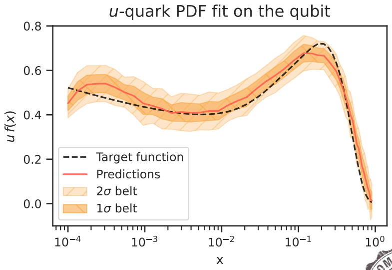
- define **prototypes** and models;
- **simulate** training and noise.

Calibration: Qibocal

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

Execution: Qibolab

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

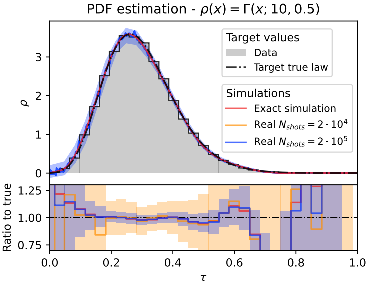


Parameter	Value
N_{data}	50
N_{shots}	500
MSE	$\sim 10^{-3}$
Electronics	Xilinx ZCU216
Training time	$\sim$ 2h

Some more applications

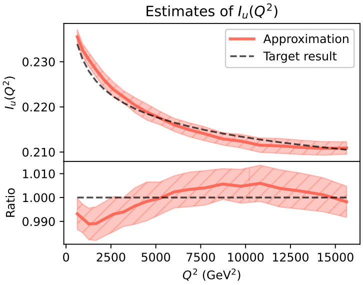
Some more applications

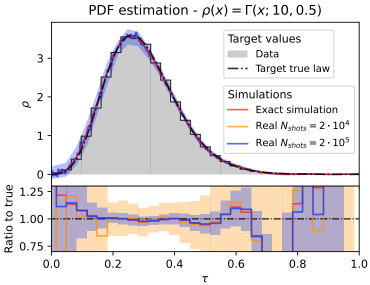
- Density estimation via Adiabatic QML,
  arXiv:2303.11346;



Some more applications

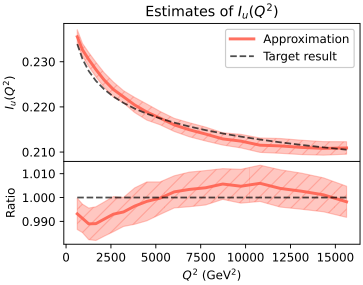
- Density estimation via Adiabatic QML, arXiv:2303.11346;
- Multi-dimensional integration using Variational Quantum Circuits, arXiv:2308.05657;

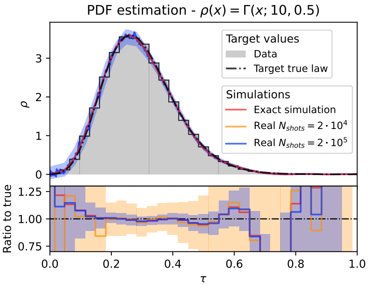


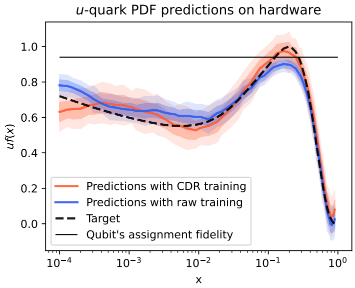


Some more applications

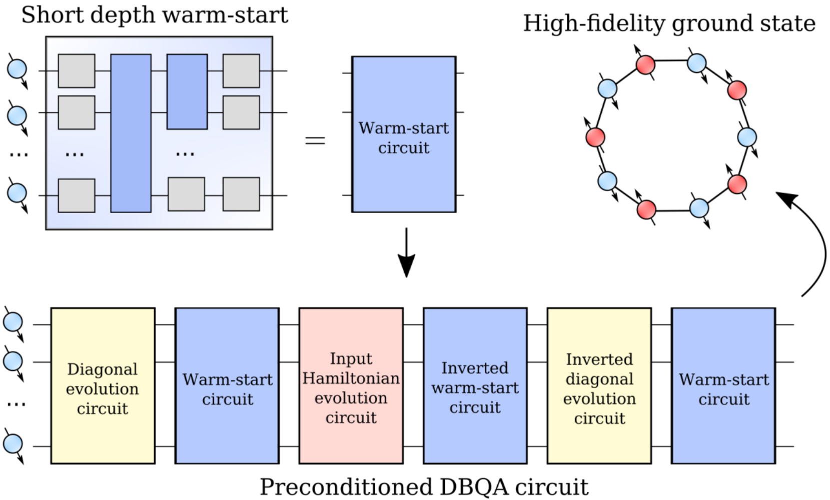
- Density estimation via Adiabatic QML, 📘 arXiv:2303.11346;
- Multi-dimensional integration using Variational Quantum Circuits, 📘 arXiv:2308.05657;
- Real-Time Quantum Error Mitigation in QML, 📘 arXiv:2311.05680.







Bonus: preparing high-fidelity ground states



Bonus: preparing high-fidelity ground states