Introducing Qibo

from quantum circuits to machine learning arXiv:2009.01845

Stefano Carrazza

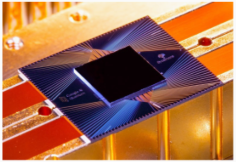
6th October 2021

PyHEP topical meeting, Università degli Studi di Milano

Introduction

Introduction

From a practical point of view, we are moving towards new technologies, in particular hardware accelerators:

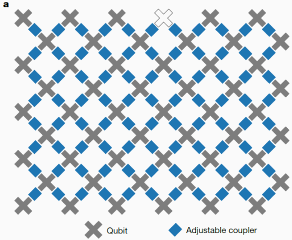
CPU	GPU	FPGA/ASIC	Quantum chip
			

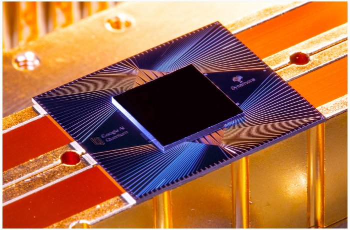
Moving from general purpose devices $\Rightarrow$ application specific

For example, in HEP we are transitioning from **CPU to GPU**.

Quantum advantage

First quantum computation that can not be reproduced on a classical supercomputer from Google, Nature 574, 505-510(2019):





53 qubits (86 qubit-couplers) $\rightarrow$ Task of sampling the output of a pseudo-random quantum circuit (extract probability distribution).

Classically the probability distribution is exponentially more difficult.

Qubits

What is a qubit?

Let us consider a two-dimensional Hilbert space, we define the computational basis:

$$|0\rangle \to \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle \to \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

A quantum bit (**qubit**) is the basic unit of quantum information and it written as:

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

where $\alpha, \beta \in \mathbb{C}$ and the state is normalized, i.e. $|\alpha|^2 + |\beta|^2 = 1$.

All quantum mechanics rules are preserved: state measurement is probabilistic, wave-function collapse after measurement, no-cloning theorem, etc.

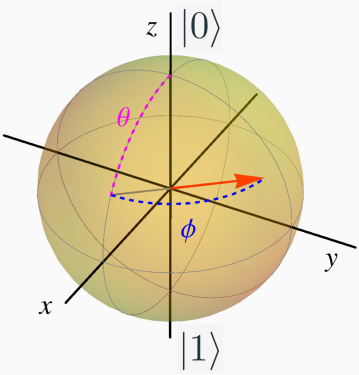
The Bloch sphere

Qubit states can be graphically represented in the Bloch sphere, by defining ϕ and θ angles and associating to the state coefficients:

$$\alpha = \cos\frac{\theta}{2}, \quad \text{and} \quad \beta = e^{i\phi}\sin\frac{\theta}{2}, \quad \text{with} \quad \theta \in [0, \pi], \phi \in [0, 2\pi].$$

We can use a 3D vector representation as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix}$$



In particular:

- $|0\rangle = (0, 0, 1)$, $|1\rangle = (0, 0, -1)$
- $(|0\rangle + i\,|1\rangle)/\sqrt{2}$ equator of the sphere

Multiple qubits states

A system with n qubits lives in 2^n-dimensional Hilbert space, defining the basis:

$$|0\rangle_n = |00\ldots00\rangle\,,\ |1\rangle_n = |00\ldots01\rangle\,,\ |2\rangle_n = |00\ldots10\rangle\,,\ \ldots, |2^n-1\rangle_n = |11\ldots1\rangle$$

therefore a generic n qubits state is defined as

$$|\psi_n\rangle = \sum_{i=0}^{2^n-1} \alpha_i\,|i\rangle_n \quad \text{with} \quad \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$$

i.e. a superposition state vector in 2^n dimensional Hilbert space.

Quantum operators

As any other quantum state defined in Hilbert space, qubits are subject to:

- time evolution via Schrödinger equation: $H(t)\,|\psi(t)\rangle = i\hbar\partial_t\,|\psi(t)\rangle$
- quantum operators/gates, in particular unitary operators (reversible computing):

$$UU^\dagger = U^\dagger U = I$$

- entanglement state, e.g. supposing $|\psi_A\rangle\,|\phi_B\rangle$, e.g. Bell's states:

$$|\psi^+\rangle = \frac{|0_A\rangle\,|0_B\rangle + |1_A\rangle\,|1_B\rangle}{\sqrt{2}}, \quad |\psi^-\rangle = \frac{|0_A\rangle\,|0_B\rangle - |1_A\rangle\,|1_B\rangle}{\sqrt{2}}$$

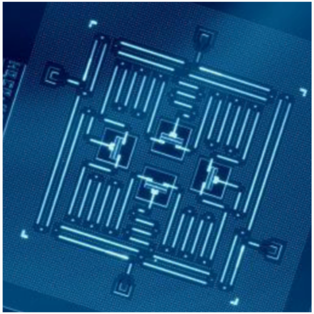
$$|\phi^+\rangle = \frac{|1_A\rangle\,|0_B\rangle + |0_A\rangle\,|1_B\rangle}{\sqrt{2}}, \quad |\phi^-\rangle = \frac{|1_A\rangle\,|0_B\rangle - |0_A\rangle\,|1_B\rangle}{\sqrt{2}}$$

Quantum technologies

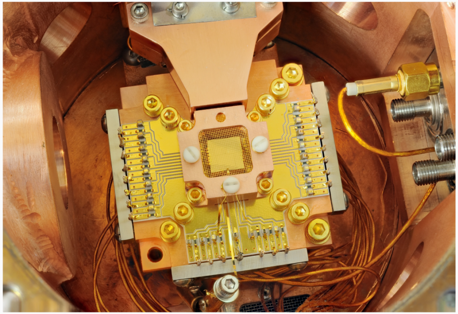
Some popular quantum technologies available today

	Superconducting loops	Trapped ions	Silicon quantum dots	Topological qubits	Diamond vacancies
	A resistance-free current oscillates back and forth around a circuit loop. An injected microwave signal excites the current into super-position states.	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.	These “artificial atoms” are made by adding an electron to a small piece of pure silicon. Microwaves control the electron’s quantum state.	Quasiparticles can be seen in the behavior of electrons channeled through semi-conductor structures. Their braided paths can encode quantum information.	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.
Number entangled	9	14	2	N/A	6
Company support	Google, IBM, Quantum Circuits	ionQ	Intel	Microsoft, Bell Labs	Quantum Diamond Technologies
Pros	Fast working. Build on existing semiconductor industry.	Very stable. Highest achieved gate fidelities.	Stable. Build on existing semiconductor industry.	Greatly reduce errors.	Can operate at room temperature.
Cons	Collapse easily and must be kept cold.	Slow operation. Many lasers are needed.	Only a few entangled. Must be kept cold.	Existence not yet confirmed.	Difficult to entangle.

Quantum chips



(a) Superconducting device assembled by IBM



(b) Chip based on trapped ions techology

Superconducting labs

The current Quantum era

NISQ era

$\Rightarrow$ **We are in a Noisy Intermediate-Scale Quantum era** $\Leftarrow$

(i.e. hardware with few noisy qubits)

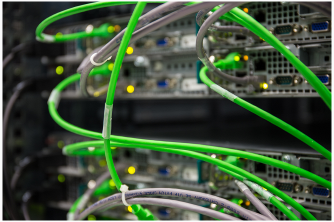
How can we contribute?

- Develop new algorithms
  $\Rightarrow$ using classical simulation of quantum algorithms
- Adapt problems and strategies for current hardware
  $\Rightarrow$ hybrid classical-quantum computation

Challenges

However, there are several challenges:

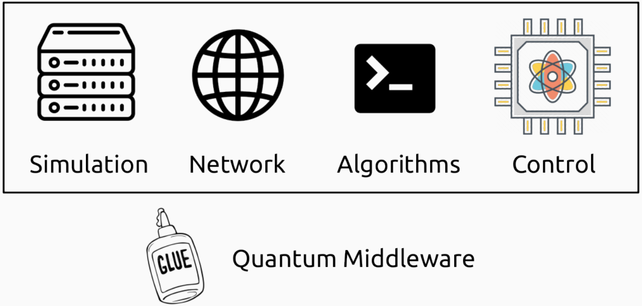
- simulate efficiently algorithms on classical hardware for QPU?
- control, send and retrieve results from the QPU?
- error mitigation, keep noise and decoherence under control?



How can we interact with QPU?

Solution:

Construct a Quantum Middleware:



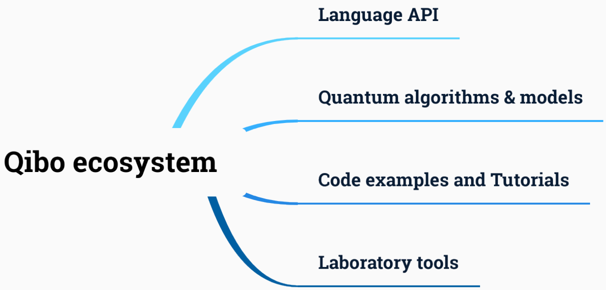
Introducing Qibo

Introducing Qibo

Qibo is an open-source full stack **API** for quantum simulation and hardware control.

It is platform **agnostic** and supports **multiple backends**.

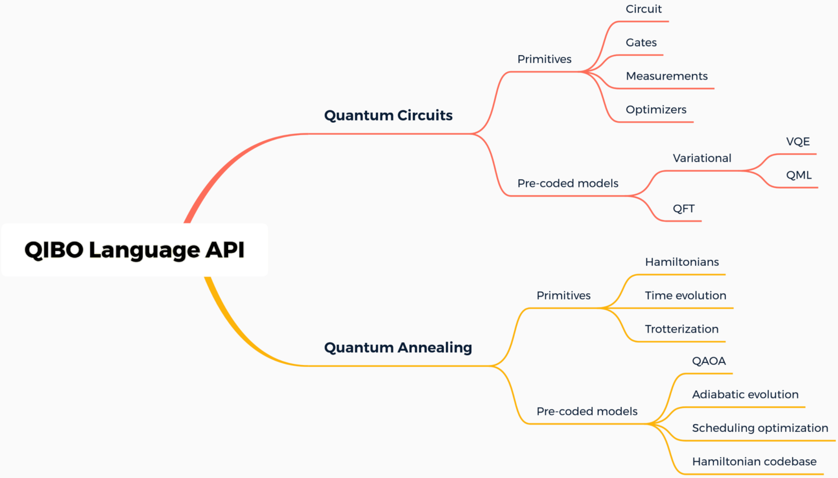
https://github.com/qiboteam/qibo https://arxiv.org/abs/2009.01845



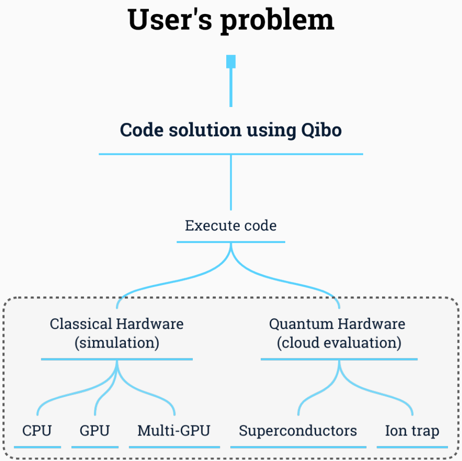
Qibo provides the level of flexibility required in HEP applications.

Computational models in Qibo

Example of models included in **Qibo**:



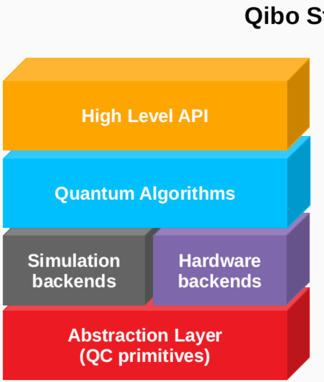
Abstractions in Qibo



- Single piece of code
- Automatic deployment on simulators and quantum devices
- Plugin backends mechanism


Los Alamos
NATIONAL LABORATORY
EST. 1943
QILIMANJARO
QUANTUM · TECH
IFAE
EXCELENCIA SEVERO OCHOA
Institut de Física d'Altes Energies
UNIVERSITAT DE BARCELONA
BSC
Barcelona Supercomputing Center
Centro Nacional de Supercomputación
CERN
UNIVERSITÀ DEGLI STUDI DI MILANO
INFN
Istituto Nazionale di Fisica Nucleare
TII
Technology Innovation Institute
CQT
Centre for Quantum Technologies
National University of Singapore
SG
NSCC
SINGAPORE
National Supercomputing Centre
Agency for Science, Technology and Research
SINGAPORE
Universidade Federal do Rio de Janeiro
UBA
Universidad de Buenos Aires

Abstractions in Qibo



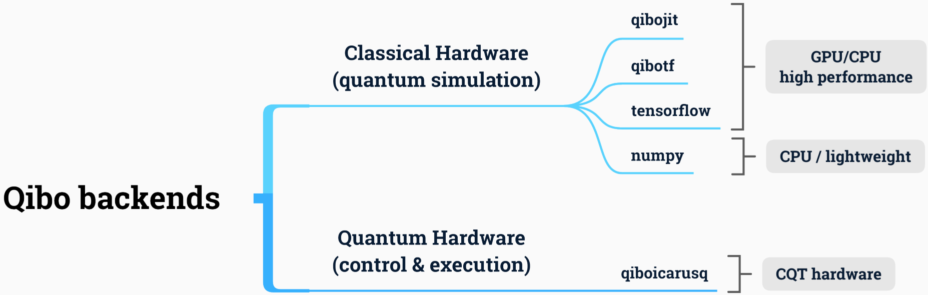
Interface for users: model definition and execution.

Implementation of algorithms based on quantum operations.

Backend specialization for classical and quantum hardware.

Code abstraction for circuit and gates representation.

Backends in Qibo



This layout opens the possibility to develop application specific projects.

⇒ *e.g.* parton distribution function determination (PRD arXiv:2011.13934).

numpy

pip install qibo

Simulator based on tensordot and linear algebra operations.

Features:

- Cross-architecture (x86, arm64, etc).
- Cross-platform.
- Fast for single-threaded operations.

tensorflow

pip install tensorflow

Simulator based on tensorflow primitives (einsum, matmul).

Features:

- Multithreading CPU.
- Single GPU.
- Gradient descent on quantum circuits.

qibotf

pip install qibotf

Simulator based on tensorflow custom operators in C++ and CUDA.

Features:

- Excellent single node performance.
- Multithreading CPU.
- Multi-GPU.
- Low memory footprint.

qibojit NEW

pip install qibojit

Simulator based on numba and cupy operations.

Features:

- Excellent single node performance.
- Multithreading CPU, single GPU and multi-GPU
- Cross-platform (just-in-time compilation)
- Works on NVIDIA and AMD GPUs.

Quantum computing with qubits

Quantum circuits

The quantum circuit model considers a sequence of unitary quantum gates:

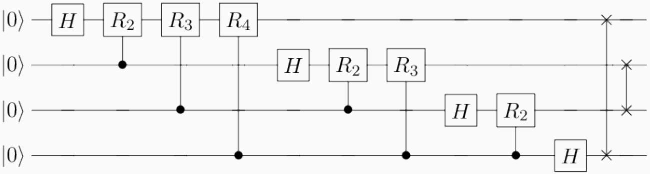
$$\left|\psi'\right\rangle = U_2 U_1 \left|\psi\right\rangle \quad \rightarrow$$

Quantum circuits

The quantum circuit model considers a sequence of unitary quantum gates:

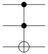
$$|\psi'\rangle = U_2 U_1 |\psi\rangle \quad \rightarrow$$

For example a Quantum Fourier Transform with 4 qubits is represented by



Quantum gates

- **Single-qubit gates**
  - Pauli gates
  - Hadamard gate
  - Phase shift gate
  - Rotation gates
- **Two-qubit gates**
  - Conditional gates
  - Swap gate
  - fSim gate
- Special gates: Toffoli

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

Quantum circuit simulation

Classical simulation of quantum circuits uses dense complex state vectors $\psi(\sigma_1, \sigma_2, \ldots, \sigma_N) \in \mathbb{C}$ in the computational basis where $\sigma_i \in \{0, 1\}$ and N is the total number of qubits in the circuit.

The final state of circuit evaluation is given by:

$$\psi'(\boldsymbol{\sigma}) = \sum_{\boldsymbol{\sigma'}} G(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) \psi(\sigma_1, \ldots \sigma'_{i_1}, \ldots, \sigma'_{i_{N_{\text{targets}}}}, \ldots, \sigma_N),$$

where the sum runs over qubits targeted by the gate.

- $G(\boldsymbol{\sigma}, \boldsymbol{\sigma'})$ is a gate matrix which acts on the state vector.
- $\psi(\boldsymbol{\sigma})$ from a simulation point of view is bounded by memory.

Pauli gates

X gate

The X gate acts like the classical NOT gate, it is represented by the σ_x matrix,

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

therefore

$$|0\rangle \; -\boxed{X}- \; |1\rangle$$

$$|1\rangle \; -\boxed{X}- \; |0\rangle$$

Z gate

The Z gate flips the sign of $|1\rangle$, it is represented by the σ_z matrix,

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

therefore

$$|0\rangle \; -\boxed{Z}- \; |0\rangle$$

$$|1\rangle \; -\boxed{Z}- \; -|1\rangle$$

Hadamard gate

The Hadamard gate (H gate) is defined as

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

Therefore it creates a superposition of states

$$|0\rangle \; -\boxed{H}- \quad \frac{|0\rangle + |1\rangle}{\sqrt{2}} \equiv |+\rangle$$

$$|1\rangle \; -\boxed{H}- \quad \frac{|0\rangle - |1\rangle}{\sqrt{2}} \equiv |-\rangle$$

The rotation gates

Rotations gates (Bloch sphere) are defined as

$$R_X(\theta) = e^{-i\frac{\theta}{2}\sigma_x} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \qquad R_Y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

Note that $R_X(\pi) \equiv X, R_Y(\pi) \equiv Y, R_Z(\pi) \equiv Z$.

The rotation gates

Rotations gates (Bloch sphere) are defined as

$$R_X(\theta) = e^{-i\frac{\theta}{2}\sigma_x} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}, \qquad R_Y(\theta) = e^{-i\frac{\theta}{2}\sigma_y} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

$$R_Z(\theta) = e^{-i\frac{\theta}{2}\sigma_z} = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$$

Note that $R_X(\pi) \equiv X, R_Y(\pi) \equiv Y, R_Z(\pi) \equiv Z$.

Every unitary transformation as decomposed in rotations around the y and z axis:

$$U \equiv R_Z(\theta_1)R_Y(\theta_2)R_Z(\theta_3),$$

for a fixed set of angles θ_1, θ_2 and θ_3.

Two-qubit gates

The controlled-NOT (CNOT) gate is a conditional gate defined as

$$\text{CNOT} = \begin{pmatrix} 1 & 0 \\ 0 & \sigma_X \end{pmatrix}$$

We define a control qubit which if at $|1\rangle$ applies X to a target qubit.

Supposing the first qubit is the control and the second qubit the target:

$$|00\rangle \to |00\rangle \qquad |01\rangle \to |01\rangle$$

$$|10\rangle \to |1\mathbf{1}\rangle \qquad |11\rangle \to |1\mathbf{0}\rangle$$

Two-qubit gates

The controlled-NOT (CNOT) gate is a conditional gate defined as

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 \\ 0 & \sigma_X \end{pmatrix}$$

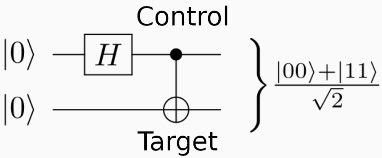
We define a control qubit which if at $|1\rangle$ applies X to a target qubit.

Supposing the first qubit is the control and the second qubit the target:

$$|00\rangle \to |00\rangle \qquad |01\rangle \to |01\rangle$$

$$|10\rangle \to |11\rangle \qquad |11\rangle \to |10\rangle$$

CNOT allows entangled states, e.g.:



Measurements

So far we have simulated quantum circuits using wave-function propagation.

In real experiments we perform measurements with a preselected number of shots.

Shots contribute to the reconstruction of the underlying wave-function distribution.

Measurement (M) gate:

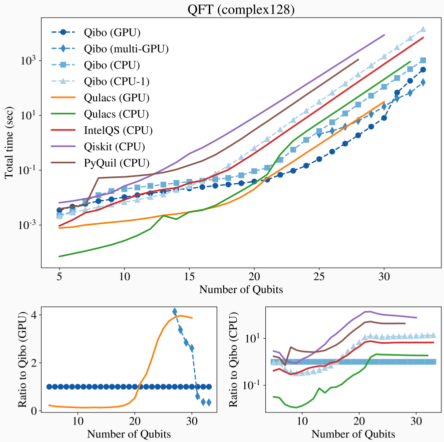
Lets consider the following circuit:

The analytic final state is:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

When measuring the final state we obtain 0 or 1 each with 50% probability.

Benchmarks

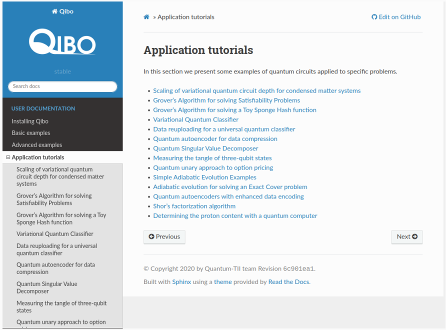


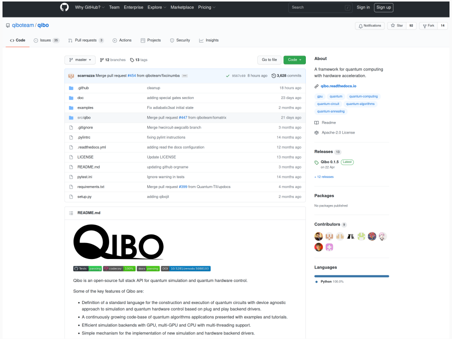
Quantum Fourier Transform performance on the left. Variational circuit simulation performance comparison in single precision (right).

Tutorial

Hands on tutorial

We will use `Qibo` for a practical demonstration:





Documentation: https://qibo.readthedocs.io

GitHub: https://github.com/qiboteam/qibo

Hands on tutorial

Visit the tutorial:

https://colab.research.google.com/drive/1M4HV1RroiHtxh4uZdrSGASv51Tjpe6dT?usp=sharing

Variational Quantum Circuits

Variational Quantum Circuits

Getting inspiration from **AI**:

- Supervised Learning $\Rightarrow$ Regression and classification
- Unsupervised Learning $\Rightarrow$ Generative models, autoencoders
- Reinforcement Learning $\Rightarrow$ Quantum RL / Q-learning

Variational Quantum Circuits

Getting inspiration from **AI**:

- Supervised Learning $\Rightarrow$ Regression and classification
- Unsupervised Learning $\Rightarrow$ Generative models, autoencoders
- Reinforcement Learning $\Rightarrow$ Quantum RL / Q-learning

Define new parametric model architectures for quantum hardware:

$\Rightarrow$ **Variational Quantum Circuits / Quantum Machine Learning**

Why Quantum Machine Learning?

Why QML?

1. Proof-of-concept, study new architectures.
2. Obtain a hardware representation (analogy with GPU and FPGA).
3. Lower power consumption.

Why Quantum Machine Learning?

Why QML?

1. Proof-of-concept, study new architectures.
2. Obtain a hardware representation (analogy with GPU and FPGA).
3. Lower power consumption.

NISQ era Warning...

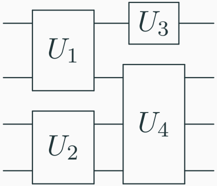
- Quantum devices implement few qubits, noise is a bottleneck.
- We can simulate quantum computation on classical hardware.

Rational

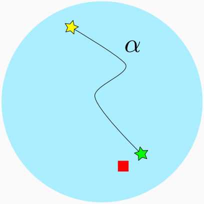
Rational:

Deliver variational quantum states $\rightarrow$ explore a large Hilbert space.

$$U(\vec{\alpha}) = U_n \dots U_2 U_1$$



Near optimal solution

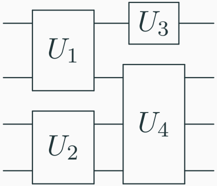


Rational for Variational Quantum Circuits

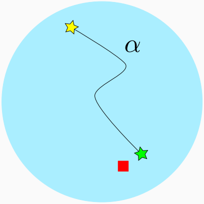
Rational:

Deliver variational quantum states $\rightarrow$ explore a large Hilbert space.

$$U(\vec{\alpha}) = U_n \dots U_2 U_1$$



Near optimal solution



Idea:

Quantum Computer is a machine that generates variational states.

$\Rightarrow$ **Variational Quantum Computer!**

Solovay-Kitaev Theorem

Let $\{U_i\}$ be a dense set of unitaries.

Define a circuit approximation to V:

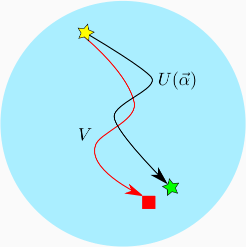
$$|U_k \dots U_2 U_1 - V| < \delta$$

Scaling to best approximation

$$k \sim \mathcal{O}\left(\log^c \frac{1}{\delta}\right)$$

where $c < 4$.

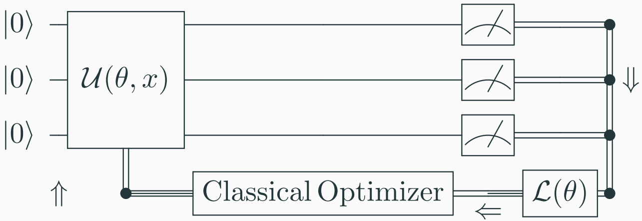
Optimal solution



$\Rightarrow$ The approximation is efficient and requires a finite number of gates.

How do we parametrize models using a quantum computer?

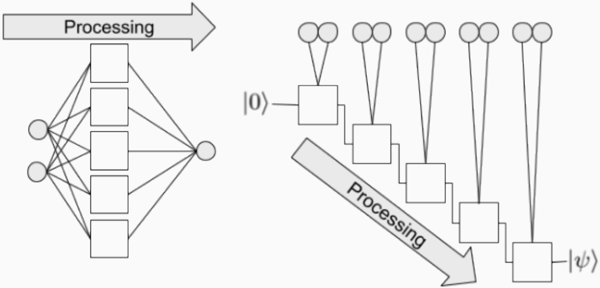
Using variational quantum circuits and data re-uploading algorithms:



Data re-uploading strategy

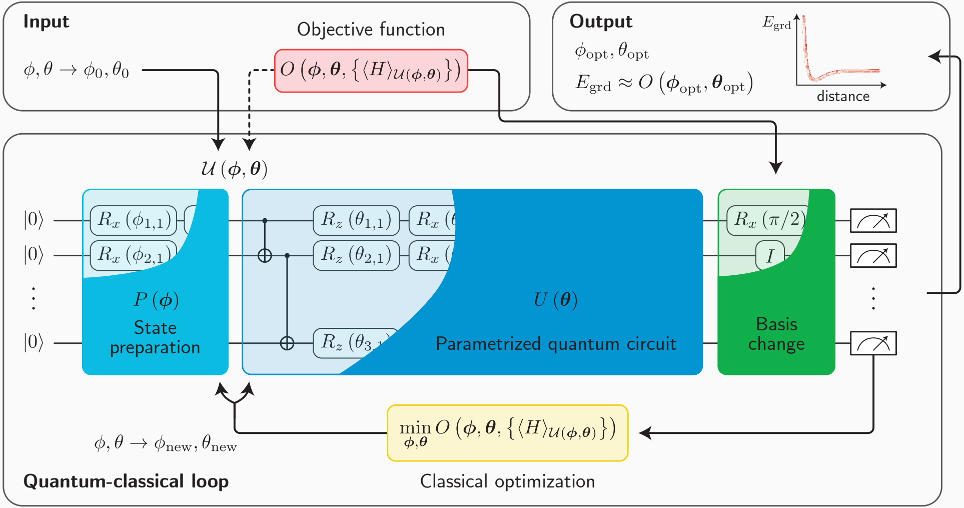
Pérez-Salinas et al. [arXiv:1907.02085]

Encode data directly "inside" circuit parameters:



(a) Neural network (b) Quantum classifier

Variational quantum algorithm



Hands on tutorial

Visit the tutorial:

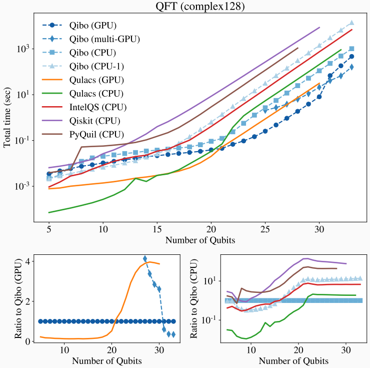
https://colab.research.google.com/drive/1M4HV1RroiHtxh4uZdrSGASv51Tjpe6dT?usp=sharing

Outlook

Outlook

Qibo is currently a framework for research:

1. publicly available as an open-source code:
   https://github.com/qiboteam/qibo
2. Designed with several abstraction layers.
3. For fast prototyping of quantum algorithms.



Qibo in the next months will:

- Support multiple quantum devices.
- Support further simulators in particular for clusters.
- Provide simple and intuitive access to remote users.

Thank you for your attention.