Qibo: an open-source hybrid quantum operating system

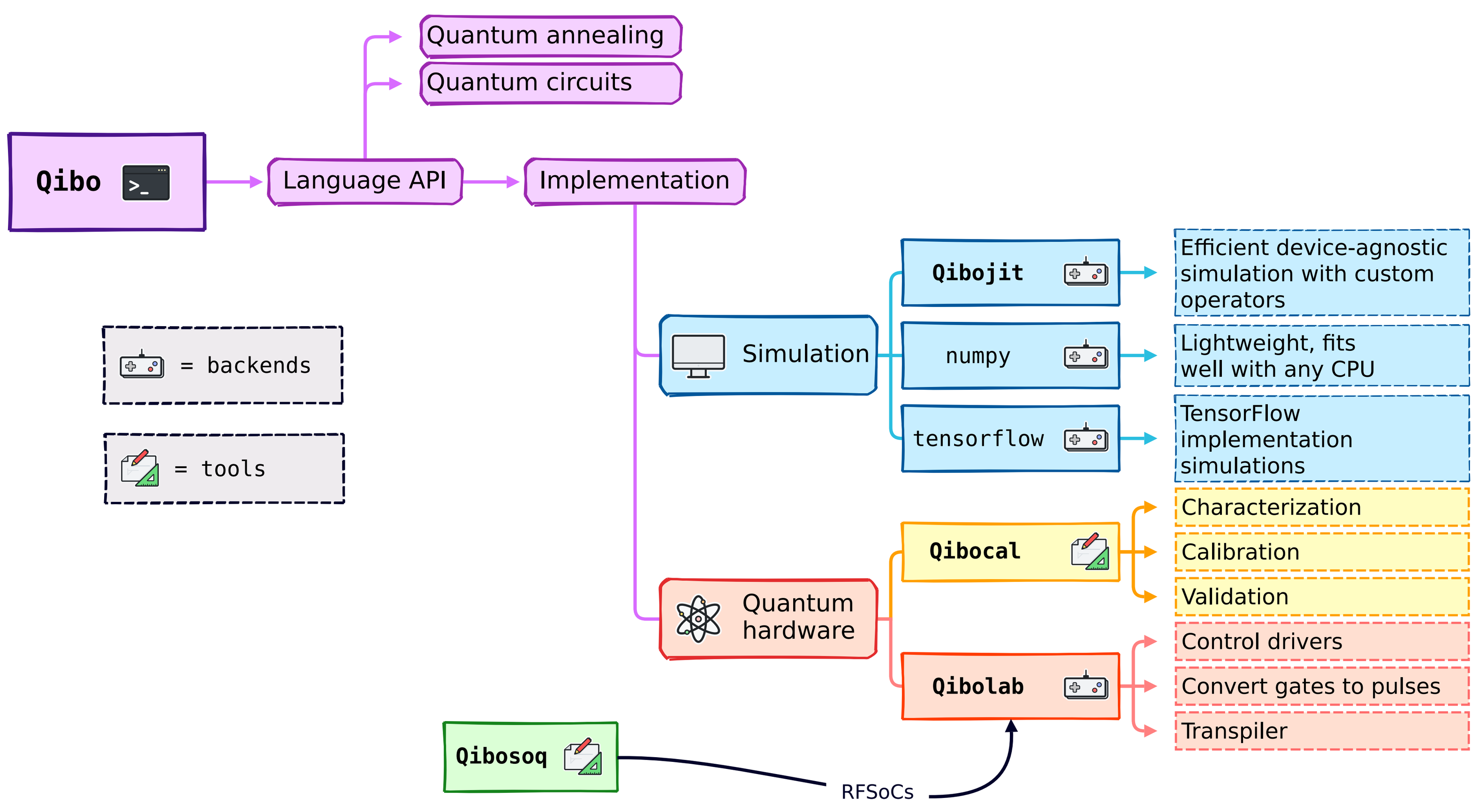
Stavros Efthymiou, Alvaro Orgaz-Fuertes, Rodolfo Carobene, Juan Cereijo, Andrea Pasquale, Sergi Ramos-Calderer, Simone Bordoni, David Fuentes-Ruiz, Alessandro Candido, Edoardo Pedicillo, Matteo Robbiati, Yuanzheng Paul Tan, Jadwiga Wilkens, Ingo Roth, José Ignacio Latorre, Stefano Carrazza.

Abstract

We present `qibo` [1], a full-stack and open source framework which can be used to:

- efficiently **simulate** quantum computation routines, both in circuit notation and in the context of adiabatic computation;
- **control self-hosted** quantum devices with an hardware-agnostic approach;
- performing quantum **characterization**, **calibration** and **verification** routines on the hosted qubits.

OVERVIEW: the qibo ecosystem



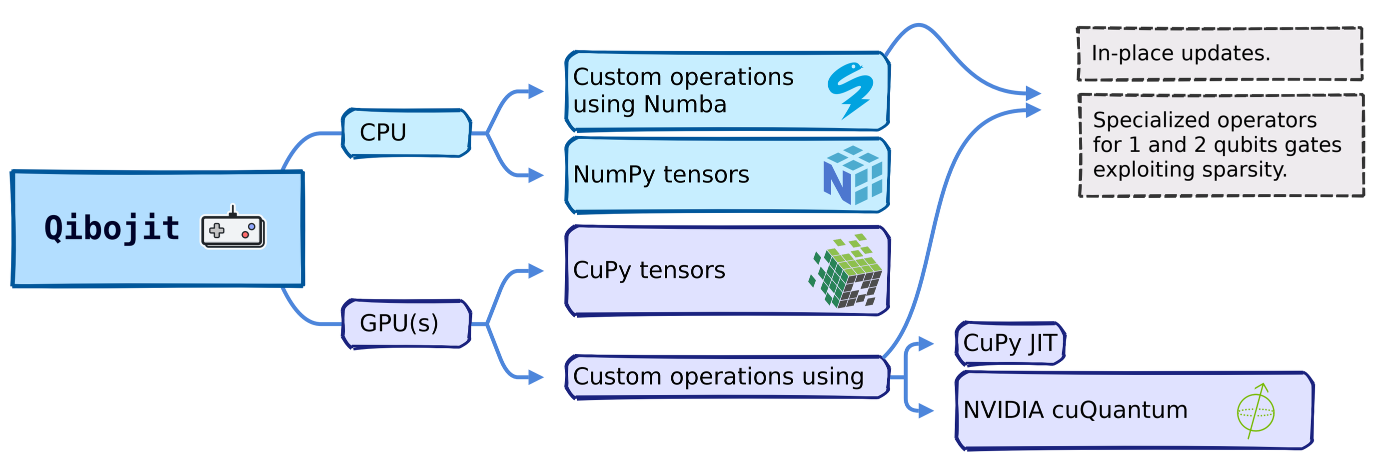
SIMULATION: qibojit

We do state vector simulation of a system of qubits $\{\sigma_j\}$, which solves:

$$\psi'(\sigma_1, ..., \sigma_n) = \sum_{\tau'} G(\tau, \tau')\psi(\sigma_1, ..., \tau', ..., \sigma_n). \tag{1}$$

The number of operations scales exponentially with N_{qubits}, thus we built the `qibojit` back-end [2].

Custom operators and **in-place updating** of matrices and vectors are exploited to improve state vector simulation of quantum systems.

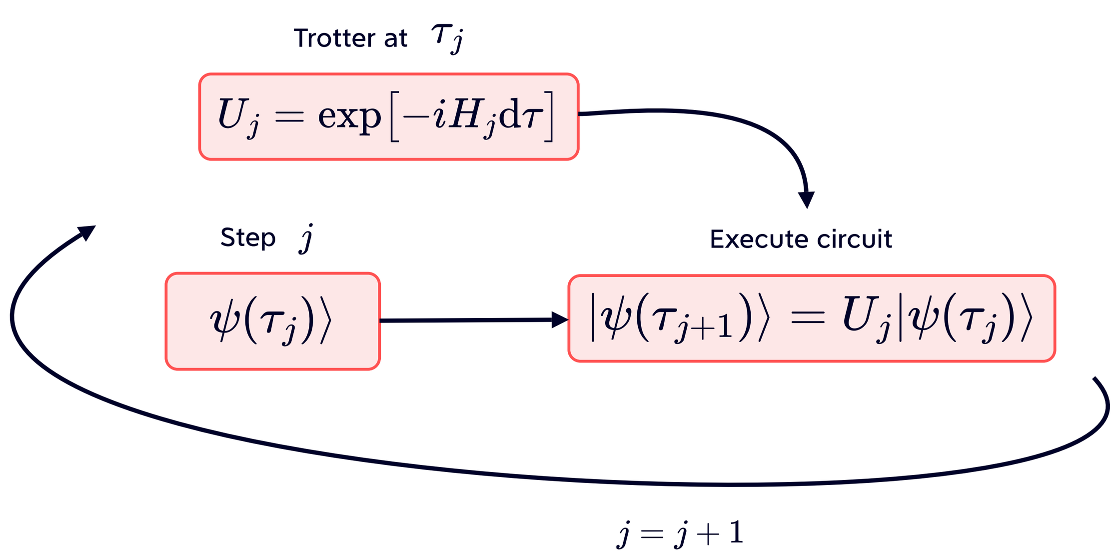


SIMULATION: adiabatic computation

In `qibo` symbolic hamiltonians can be defined and used to perform adiabatic computation:

$$H_{\text{ad}}(\tau; \boldsymbol{\theta}) = \left[1 - s(\tau; \boldsymbol{\theta})\right] H_0 + s(\tau; \boldsymbol{\theta}) H_1, \tag{2}$$

where the scheduling function $s(\tau, \boldsymbol{\theta})$ can be set to depend on variational parameters.



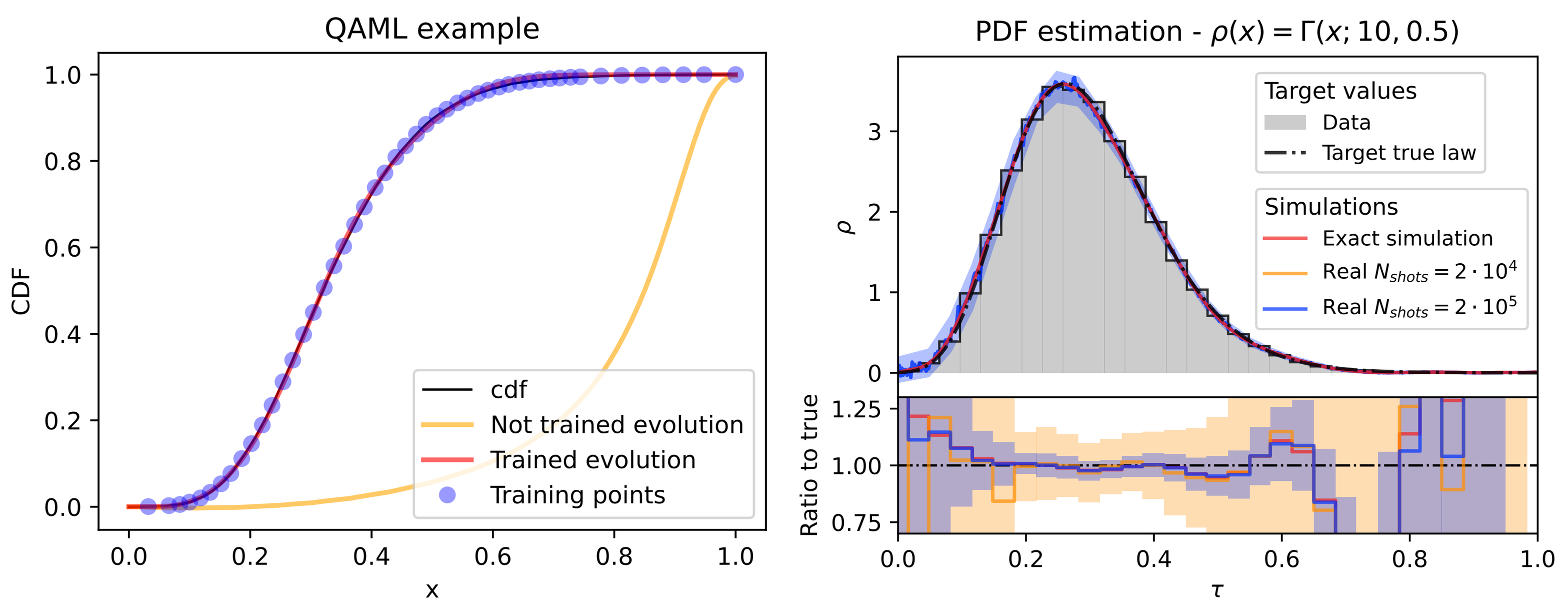
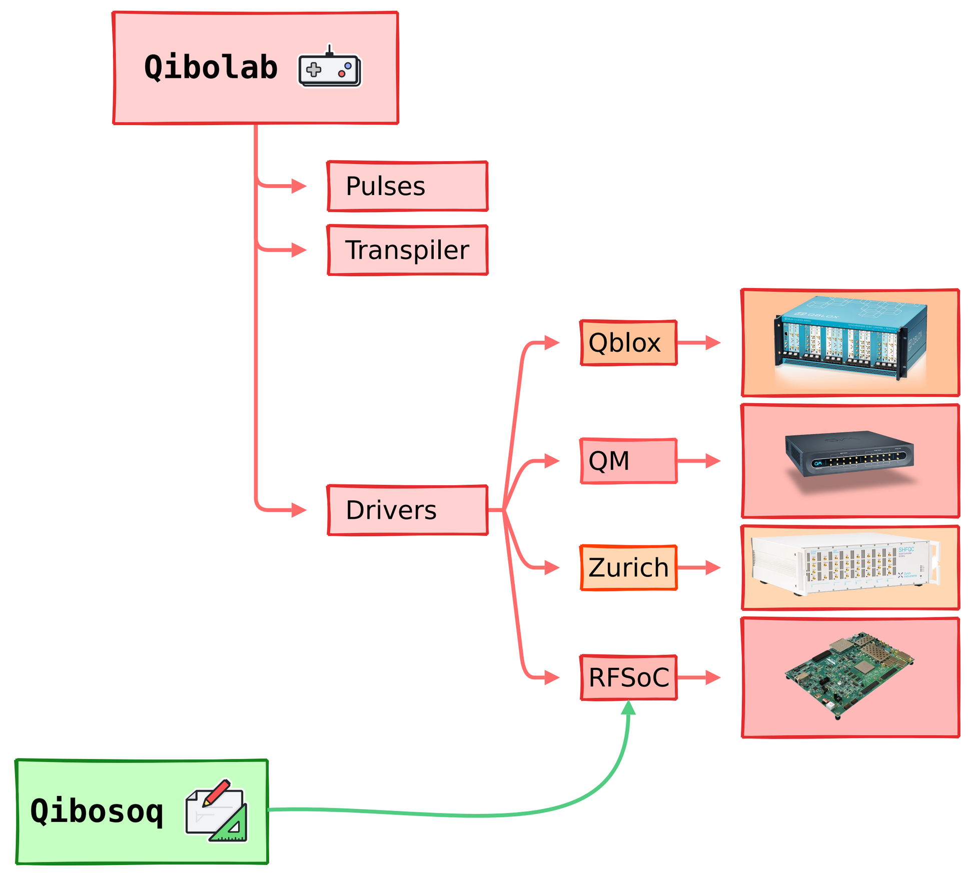


Figure 1. Both the adiabatic evolution and the circuit computation approaches are exploited within the same Quantum Adiabatic Machine Learning (QAML) algorithm to fit probability densities [3].

CONTROL: qibolab

The full-stack framework is hardware agnostic!

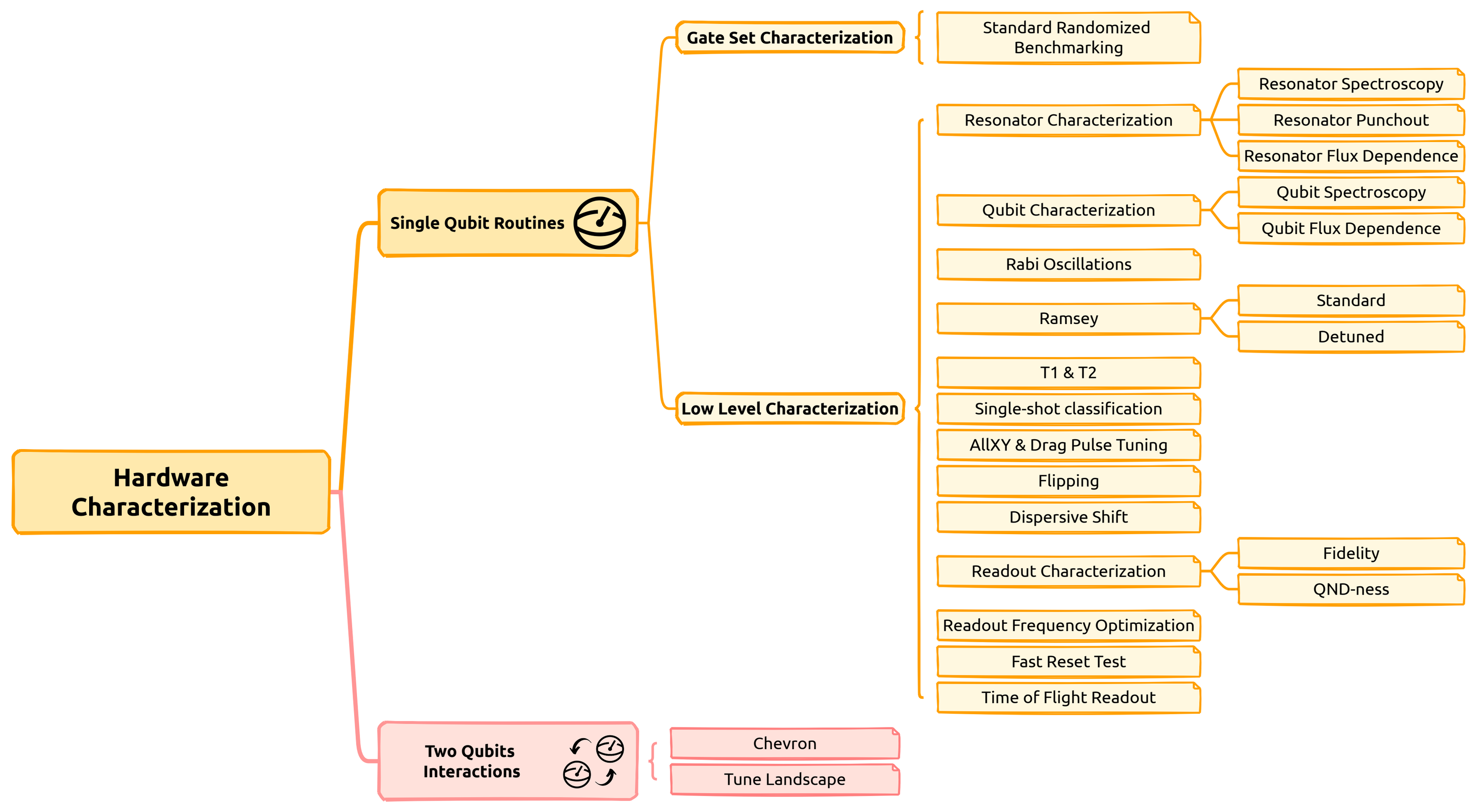


The `qibo`'s high level language can be deployed on any quantum hardware technology by defining a `platform` object:

1. a quantum computation algorithm can be written with `qibo`;
2. define `custom_platform` object for a self-hosted device;
3. the hardware backend can be set via `qibo.set_backend('qibolab', 'custom_platform')`.

CALIBRATION: qibocal

Each qubit needs characterization, calibration and verification [4].



Full-stack gradient descent using qibo

We train a quantum circuit to fit the u-quark Parton Distribution Function. The full gradient descent is performed on the device controlled via RFSoC [5].

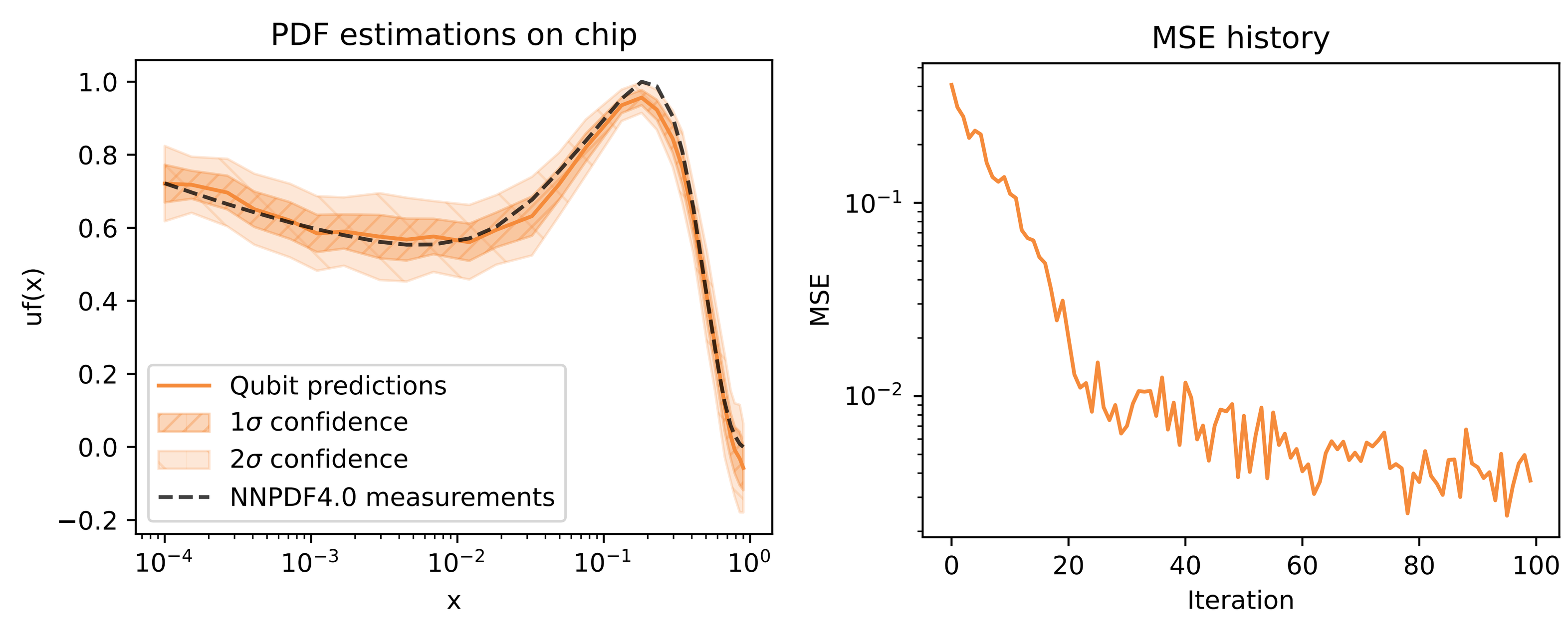


Figure 2. Fit of the u-quark Parton Distribution Function values performed using a parametric quantum circuit trained with gradient descent on a self-hosted single-qubit superconducting device.

References

[1] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, "Qibo: a framework for quantum simulation with hardware acceleration," *Quantum Science and Technology*, vol. 7, p. 015018, dec 2021.

[2] S. Efthymiou, M. Lazzarin, A. Pasquale, and S. Carrazza, "Quantum simulation with just-in-time compilation," *Quantum*, vol. 6, p. 814, sep 2022.

[3] M. Robbiati, J. M. Cruz-Martinez, and S. Carrazza, "Determining probability density functions with adiabatic quantum computing," 2023.

[4] A. Pasquale, S. Efthymiou, S. Ramos-Calderer, J. Wilkens, I. Roth, and S. Carrazza, "Towards an open-source framework to perform quantum calibration and characterization," 2023.

[5] S. Efthymiou, A. Orgaz-Fuertes, R. Carobene, J. Cereijo, A. Pasquale, S. Ramos-Calderer, S. Bordoni, D. Fuentes-Ruiz, A. Candido, E. Pedicillo, M. Robbiati, Y. P. Tan, J. Wilkens, I. Roth, J. I. Latorre, and S. Carrazza, "Qibolab: an open-source hybrid quantum operating system," 2023.