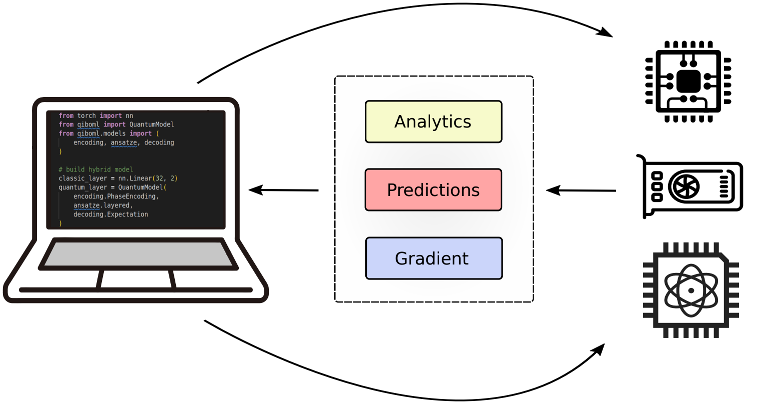
# A few slides on full-stack QML

November 21, 2024

# Quantum Machine Learning challenges

We aim to involve quantum process units into Machine Learning pipelines.

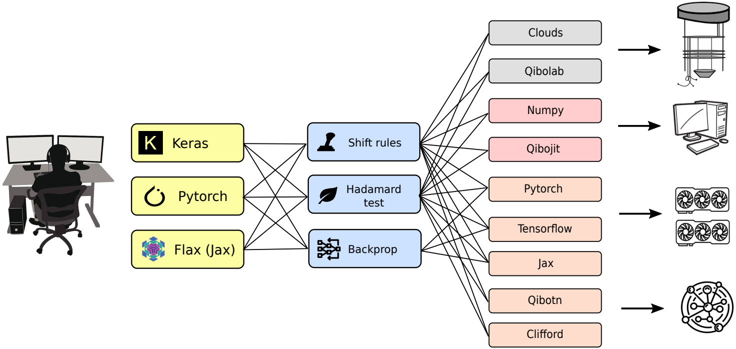



Classical and quantum components have to be executable on CPUs, GPUs and QPUs to fullfill the whole potential of an hybrid quantum-classical cluster.

# Qibo as a modular playground

To do so, `Qibo` stands as an intriguing playgrond thanks to its modularity.

Once a favourite ML framework is chosen, a quantum circuit can be built with Qibo and included into the pipeline.



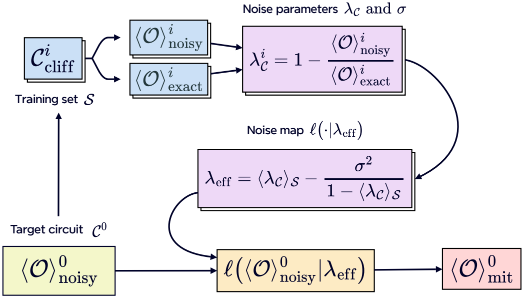

The circuit can then be executed onto the desired Qibo backend (quantum or classical).

We want to learn the Parton Distribution Function of the *up* quark in the proton with a superconducting qubit.

In doing so, we want to mitigate the noise of the quantum device using a learning-based error mitigation technique.

The method requires:

1. efficient classical simulation tools;
2. easy access to the quantum hardware;
3. ability to easily make quantum and classical outputs interact;
4. access to GPUs clusters if the process is part of a ML pipeline (optionally).



## High level API: Qibo

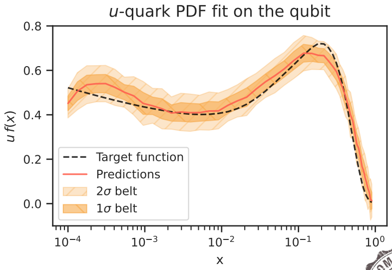
- define **prototypes** and models;
- **simulate** training and noise.

## Calibration: Qibocal

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

## Execution: Qibolab

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

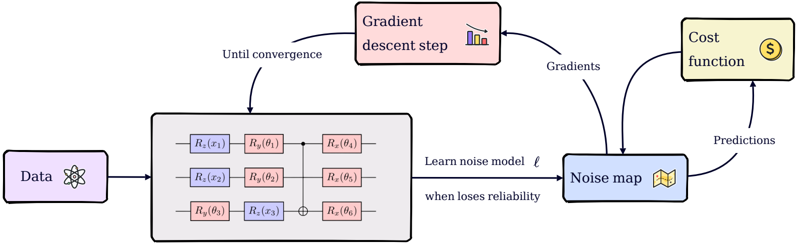




| Parameter | Value |
|---|---|
| $N_{\text{data}}$ | 50 |
| $N_{\text{shots}}$ | 500 |
| MSE | $\sim 10^{-3}$ |
| Electronics | Xilinx ZCU216 |
| Training time | $\sim$ 2h |

# Real-time quantum error mitigation

We define a Real-Time Quantum Error Mitigation (RTQEM) procedure.



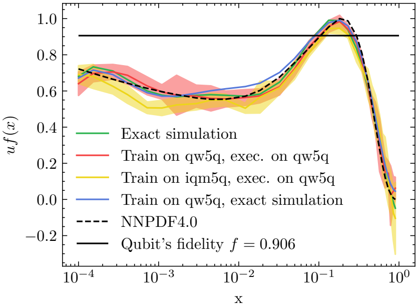

1. consider a Variational Quantum Algorithm trained with gradient descent;
2. learn the noise map $\ell$ every time is needed over the procedure;
3. use $\ell$ to clean up both predictions and gradients.

This procedure requires real-time interaction between quantum and classical devices: the former returns the output of the quantum system, the latter mitigate the noise.

# RTQEM in action

We perform a gradient descent on two different quantum devices (and noises!)

| **Parameter** | $N_{\mathrm{train}}$ | $N_{\mathrm{params}}$ | $N_{\mathrm{shots}}$ | Epochs | Optimizer | Learning rate |
|---|---|---|---|---|---|---|
| **Value** | 15 | 16 | 500 | 100 | Adam | 0.1 |




- `qw5q` from QuantWare and controlled using Qblox instruments;
- `iqm5q` from IQM and controlled using Zurich Instruments.

| **Train.** | **Epochs** | **Pred.** | **Config.** | MSE |
|---|---|---|---|---|
| `qw5q` | 100 | `qw5q` | RTQEM | 0.0013 |
| `iqm5q` | 100 | `qw5q` | RTQEM | 0.0037 |
| `qw5q` | 100 | `sim` | RTQEM | 0.0016 |

All the hardware results are obtained deploying the $\boldsymbol{\theta}_{\mathrm{best}}$ on `qw5q`.