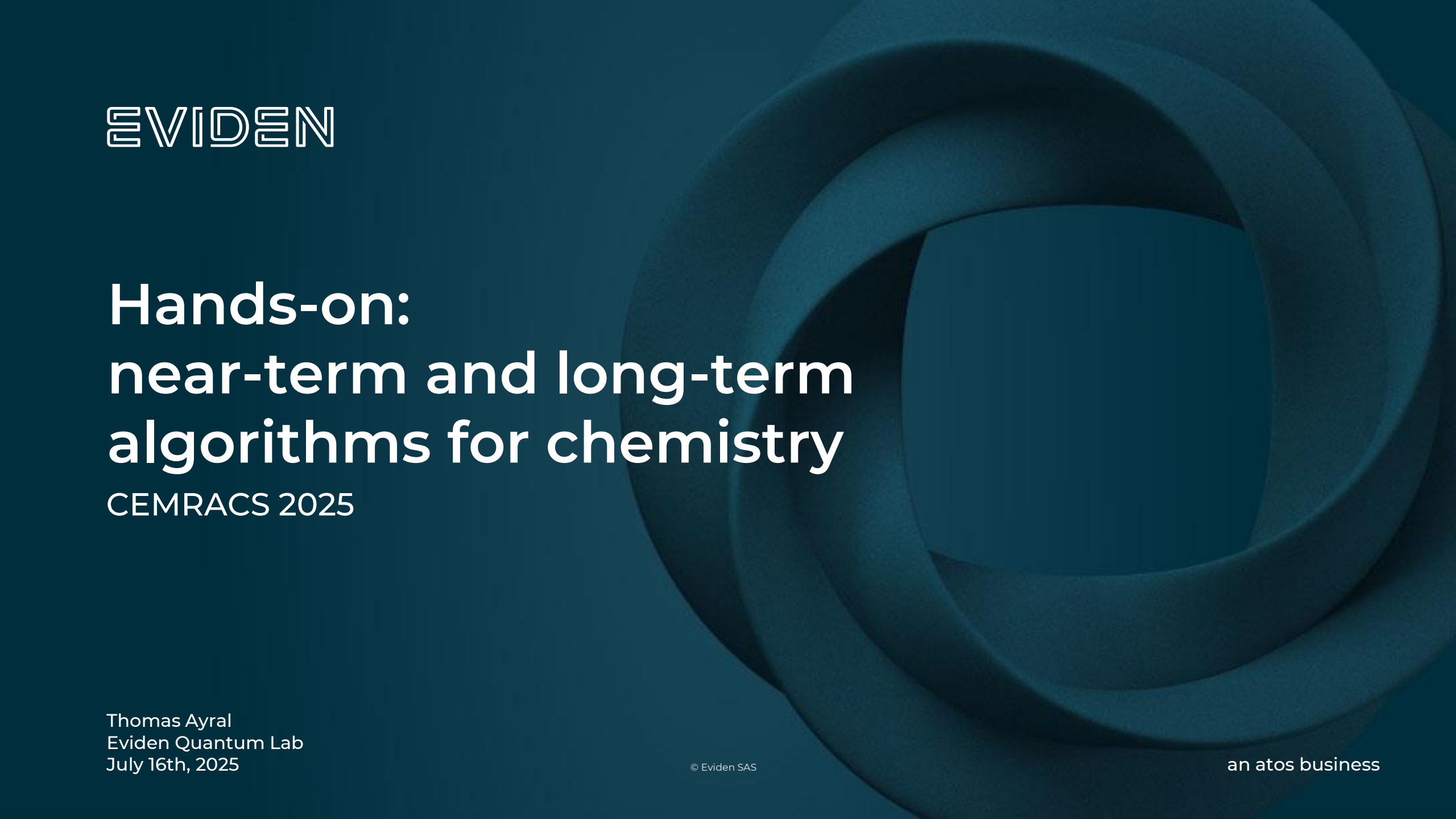


The background features a series of concentric circles in a dark teal color, with a solid teal square in the center. The circles are slightly offset, creating a 3D effect.

EVIDEN

Hands-on: near-term and long-term algorithms for chemistry

CEMRACS 2025

Thomas Ayrat
Eviden Quantum Lab
July 16th, 2025

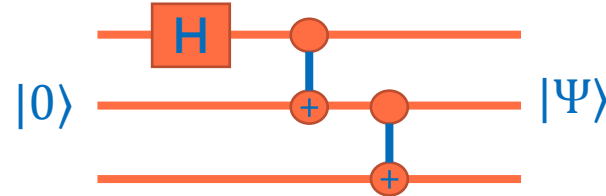
© Eviden SAS

an atos business

The QLM's key components

Jobs and QPUs

A basic quantum computation:

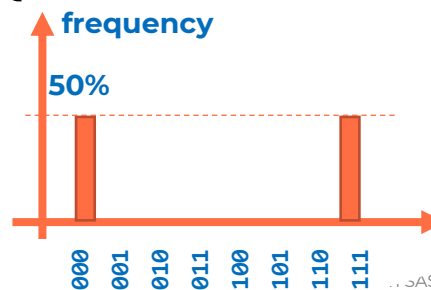


```
job = circ.to_job()
res = qpu.submit(job)
for sample in res:
    print(sample.state, sample.probability)
```

Job {Circuit}



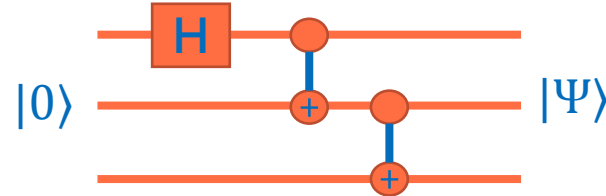
Result
{Histogram}



The QLM's key components

Jobs and QPUs

A basic quantum computation:

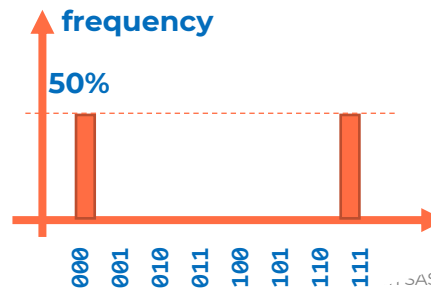


```
job = circ.to_job()
res = qpu.submit(job)
for sample in res:
    print(sample.state, sample.probability)
```

Job {Circuit}



Result
{Histogram}



What about more complex jobs?

More complex jobs (I)

Observables

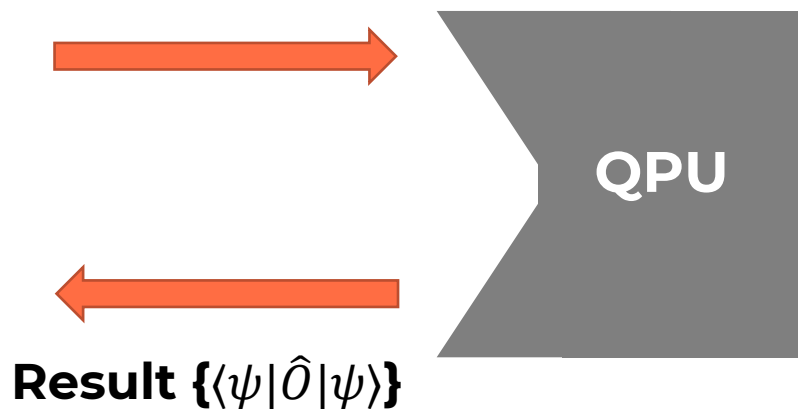
Most computations: computing/minimizing an observable:

- E.g combinatorial optimization,

$$\hat{O} = \sum_{ij} J_{ij} Z_i Z_j$$

- E.g chemistry: Hamiltonian $\hat{O} = \hat{H}$

Job {Circuit, Observable}



```
job = circ.to_job(observable=my_hamiltonian)
res = qpu.submit(job)
print(res.value) #<psi|H|psi>
```

More complex jobs (I)

Observables

Most computations: computing/minimizing an observable:

- E.g combinatorial optimization,

$$\hat{O} = \sum_{ij} J_{ij} Z_i Z_j$$

- E.g chemistry: Hamiltonian $\hat{O} = \hat{H}$

Job {Circuit, Observable}



Result $\{\langle\psi|\hat{O}|\psi\rangle\}$

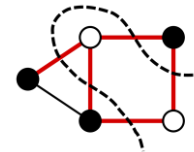


EVIDEN

Qaptiva: libraries to generate quantum jobs for many application areas

- Optimization libraries (MaxCut, etc)

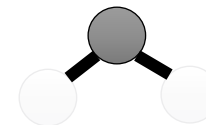
Graph



Job {Circuit, Observable}

- Chemistry, materials science libraries (myQLM-fermion)

Molecule



Job {Circuit, Observable}

```
job = circ.to_job(observable=my_hamiltonian)
res = qpu.submit(job)
print(res.value) #<psi|H|psi>
```

More complex jobs (I)

Observables

Most computations: computing/minimizing an observable:

- E.g combinatorial optimization,

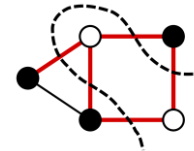
$$\hat{O} = \sum_{ij} J_{ij} Z_i Z_j$$

- E.g chemistry: Hamiltonian $\hat{O} = \hat{H}$

Qaptiva: libraries to generate quantum jobs for many application areas

- Optimization libraries (MaxCut, etc)

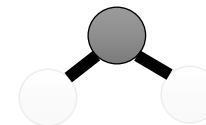
Graph



**Job {Circuit,
Observable}**

- Chemistry, materials science libraries (myQLM-fermion)

Molecule



**Job {Circuit,
Observable}**

Job {Circuit, Observable}



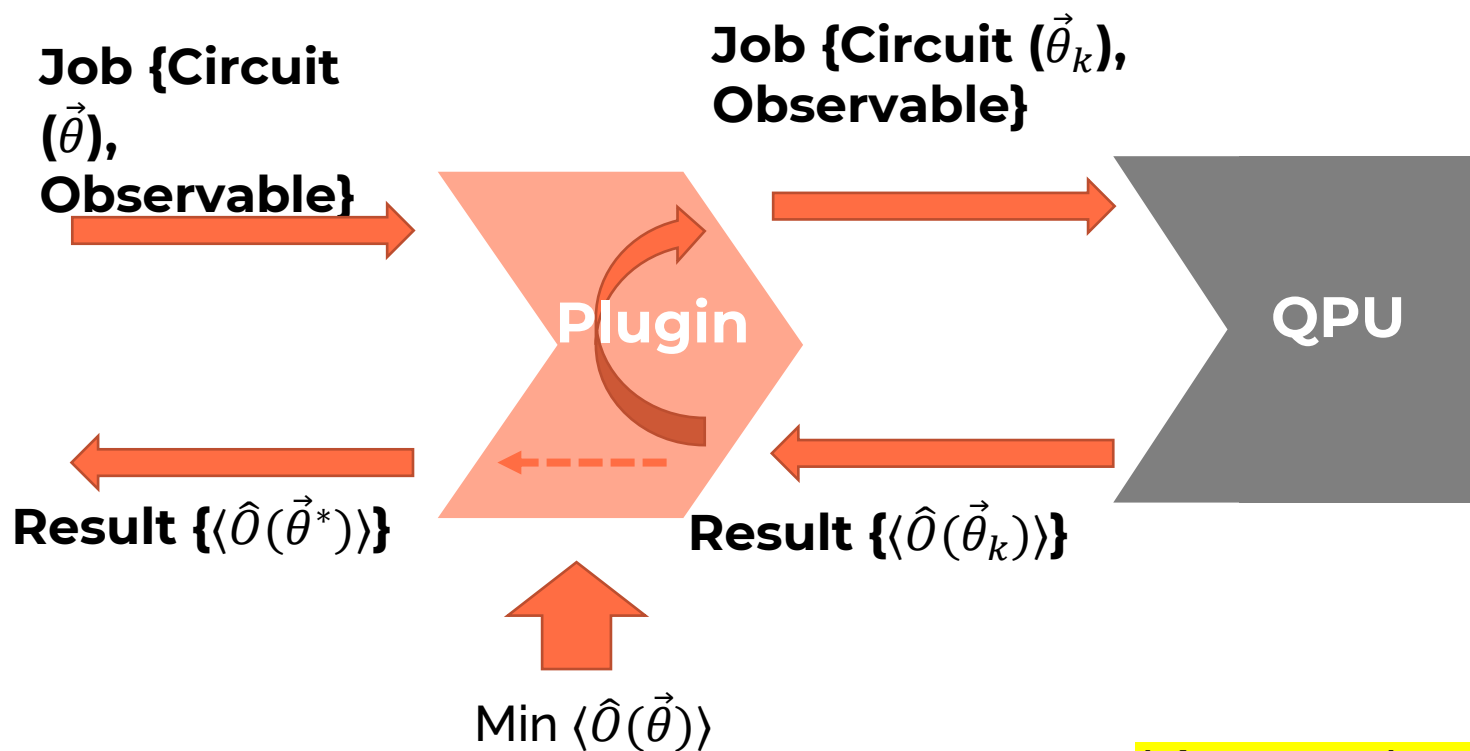
Result $\{\langle\psi|\hat{O}|\psi\rangle\}$

Minimization?

More complex jobs (II)

Minimizing observables

Many problems: minimize $\langle \psi(\vec{\theta}) | \hat{O} | \psi(\vec{\theta}) \rangle$ (VQE, QAOA...)

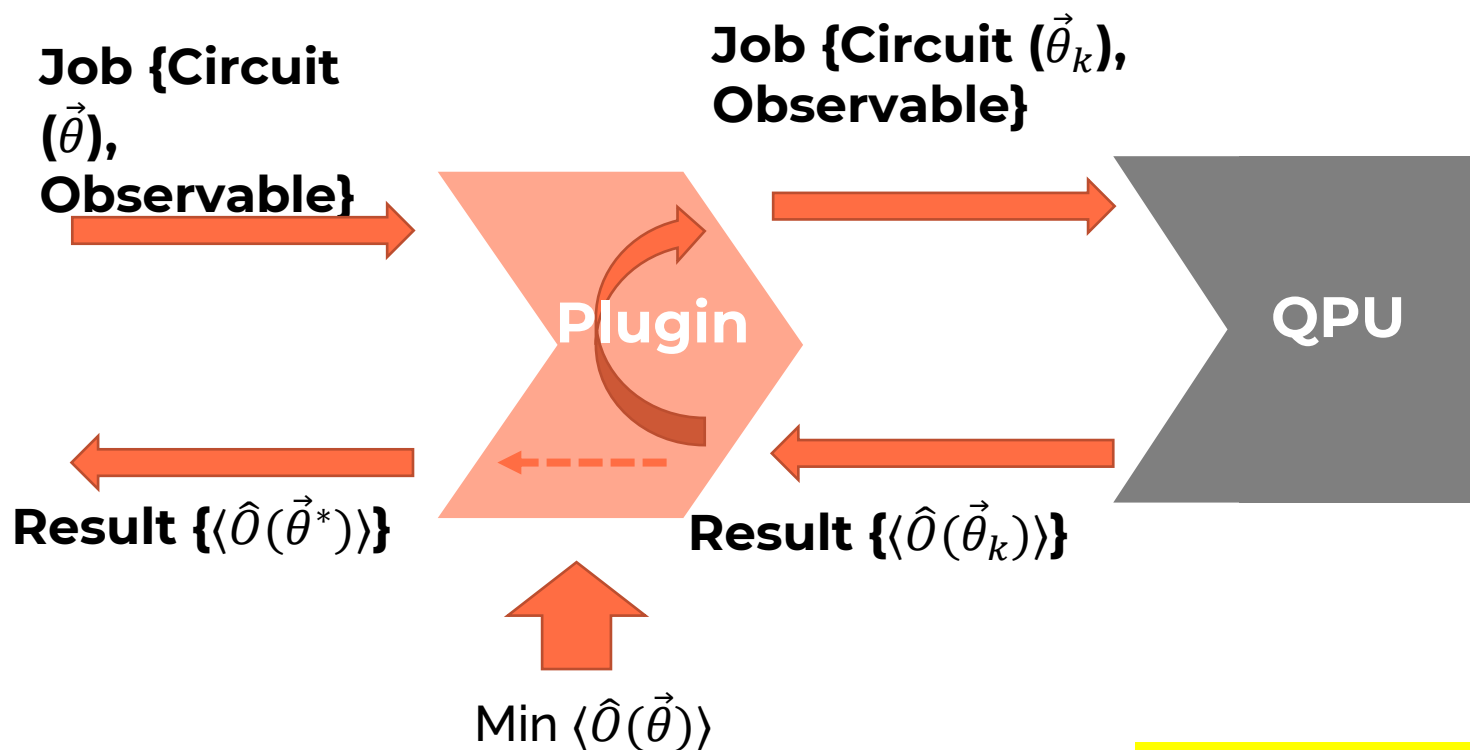


```
job = var_circ.to_job(observable=my_obs)
stack = plugin | qpu
res = stack.submit(job)
print(res.value) #<psi(theta*)|O|psi(theta*)>
```

More complex jobs (II)

Minimizing observables

Many problems: minimize $\langle \psi(\vec{\theta}) | \hat{O} | \psi(\vec{\theta}) \rangle$ (VQE, QAOA...)



Many classical optimization plugins:

- Scipy wrappers (COBYLA, BFGS)
- Gradient-free custom plugins: SPSA, Particle-Swarm Optimizer...
- Gradient-based optimization
- Natural gradient
- Etc.

```
job = var_circ.to_job(observable=my_obs)
stack = plugin | qpu
res = stack.submit(job)
print(res.value) #<psi(theta*)|O|psi(theta*)>
```

A Qaptiva computation in a nutshell

(1) Pick your application



A Qaptiva computation in a nutshell

(1) Pick your application



EVIDEN

(2) Pick your QPU

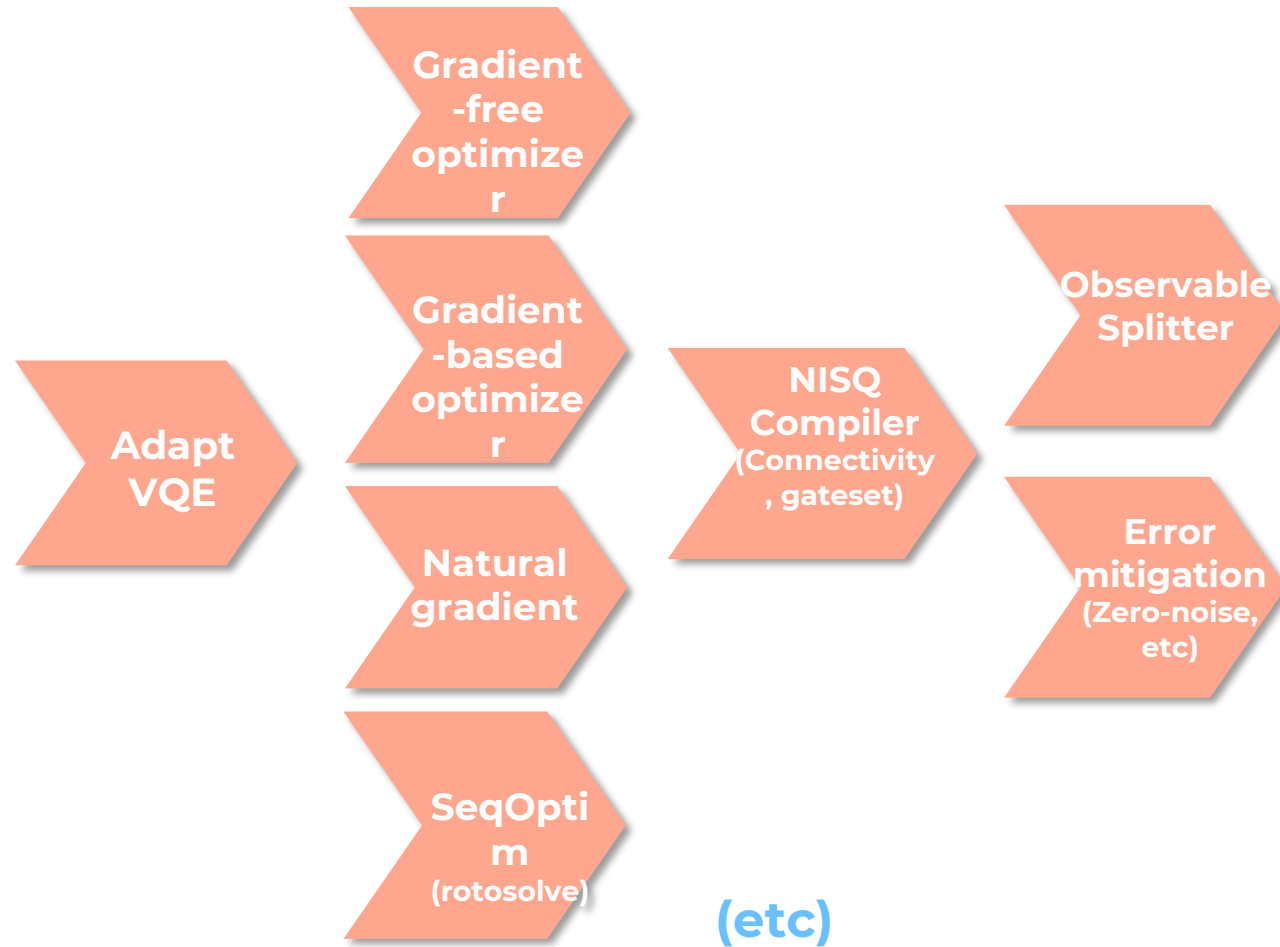


A Qaptiva computation in a nutshell

(1) Pick your application



(3) Pick your plugins



(2) Pick your QPU



EVIDEN

1.11.3

Search docs

Getting started

Installing myQLM

Writing quantum programs

Executing quantum programs

Variational algorithms with Qaptiva

User guide

API reference

Demos

Contributing to myQLM

Release notes

Variational algorithms with Qaptiva

Variational algorithms are believed to be well suited to Noisy, Intermediate-Scale Quantum (NISQ) processors as they do not necessarily require long circuits to nevertheless prepare powerful ansatz states.

In the code snippet below, we illustrate how this can be used to write such variational algorithms in a few lines of code: we first define the Hamiltonian H (here the antiferromagnetic Heisenberg Hamiltonian) whose ground-state energy we want to approximate. We then define the ansatz circuit, i.e a parametric circuit with parameters θ_i to be optimized. Finally, our quantum stack is composed of a QPU (here a simulator) and a so-called “plugin” that is going to perform the iterative optimization of the parameters given the ansatz circuit and the observable to be minimized.

Functional mode

Sequential mode

```
from qat.core import Observable as Obs, Term
from qat.lang import Program, RY, CNOT
from qat.qpus import get_default_qpu
from qat.plugins import ScipyMinimizePlugin

# we instantiate the Hamiltonian we want to approximate the ground state energy of
hamiltonian = (
    Obs.sigma_z(0) * Obs.sigma_z(1)
    + Obs.sigma_x(0) * Obs.sigma_x(1)
    + Obs.sigma_y(0) * Obs.sigma_y(1)
)
```

EVIDEN

myQLM-fermion

Quantum chemistry



- . Basis set + Geometry
- . Hartree-Fock $\rightarrow$ 1- and 2-body integrals h_{pq}, h_{pqrs} , orbital energies
- . [Beyond HF: 1-RDM D]

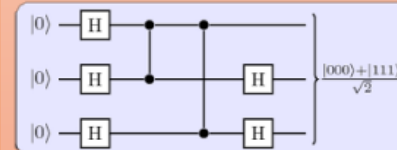
From fermions to qubits

- . Hamiltonian H
- . [Active space selection based on D]
- . Conversion to qubit representation H_{sp} (JW, BK, parity)

Ansatz construction

- . Construction of cluster operator $T(\vec{\theta})$
- . Initial guess via MP2 $\vec{\theta}_0$
- . Trotterization of $e^{T(\vec{\theta})} \rightarrow \text{circuit } C(\vec{\theta})$

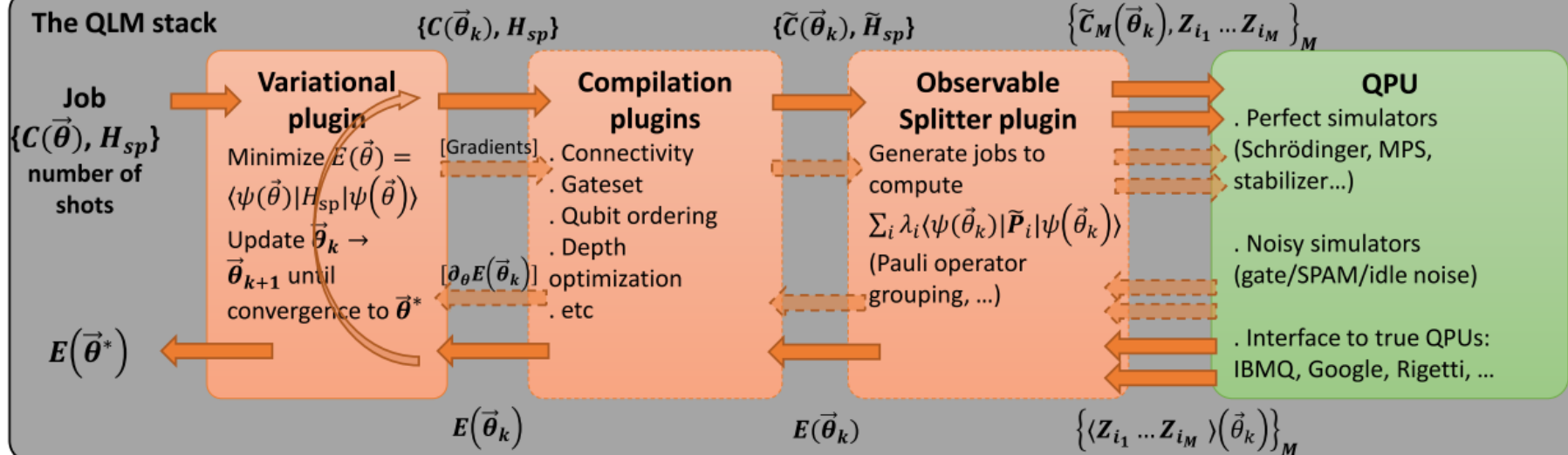
Quantum job definition



- . Variational circuit $C(\vec{\theta})$
- . Hamiltonian

$$H_{sp} = \sum_i \lambda_i P_i$$

The QLM stack





EVIDEN

Thank you

thomas.ayral@eviden.com

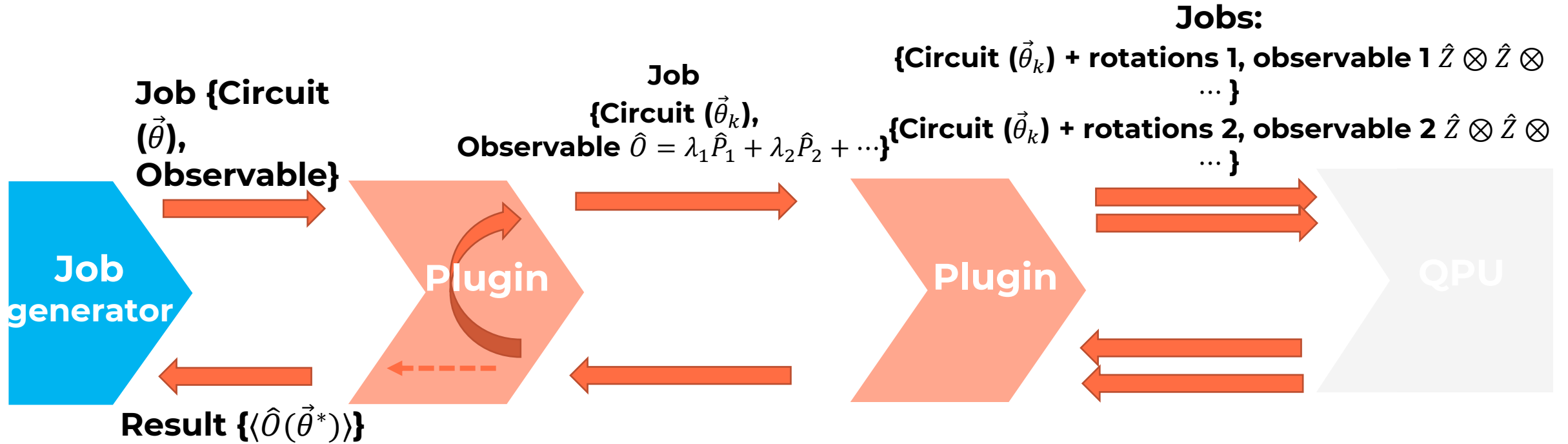
Confidential information owned by Eviden SAS, to be used by the recipient only.
This document, or any part of it, may not be reproduced, copied, circulated
and/or distributed nor quoted without prior written approval from Eviden SAS.

© Eviden SAS

Taking into account limited available observables

ObservableSplitter plugin

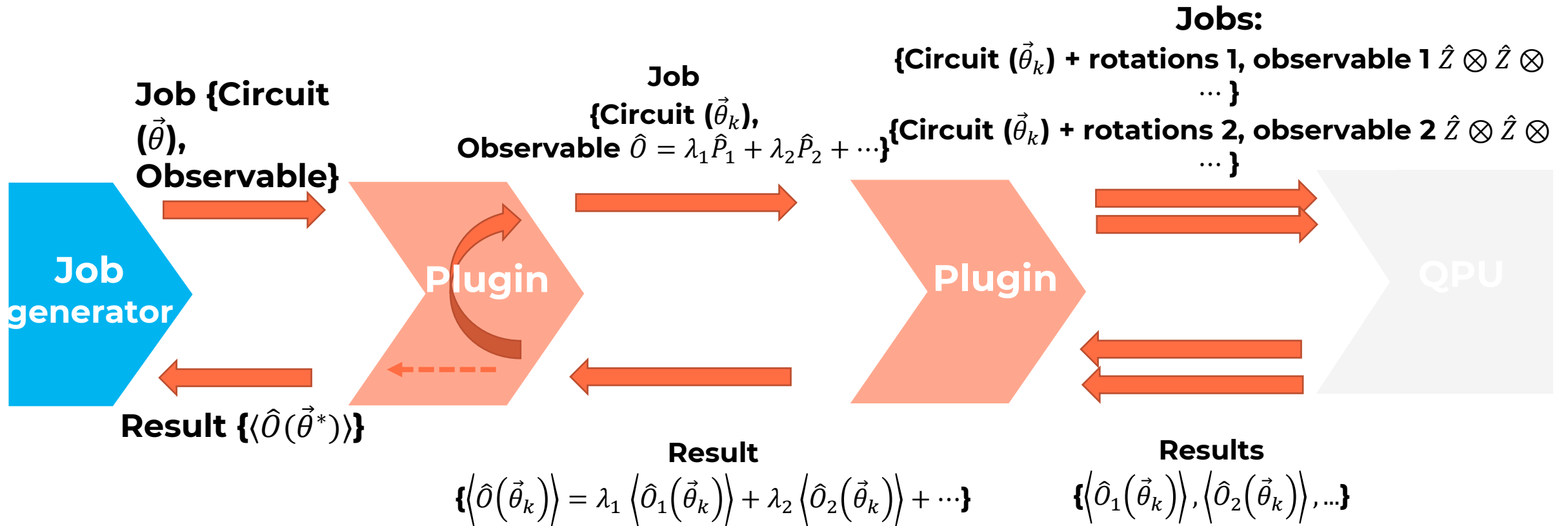
Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \dots$ observables...



Taking into account limited available observables

ObservableSplitter plugin

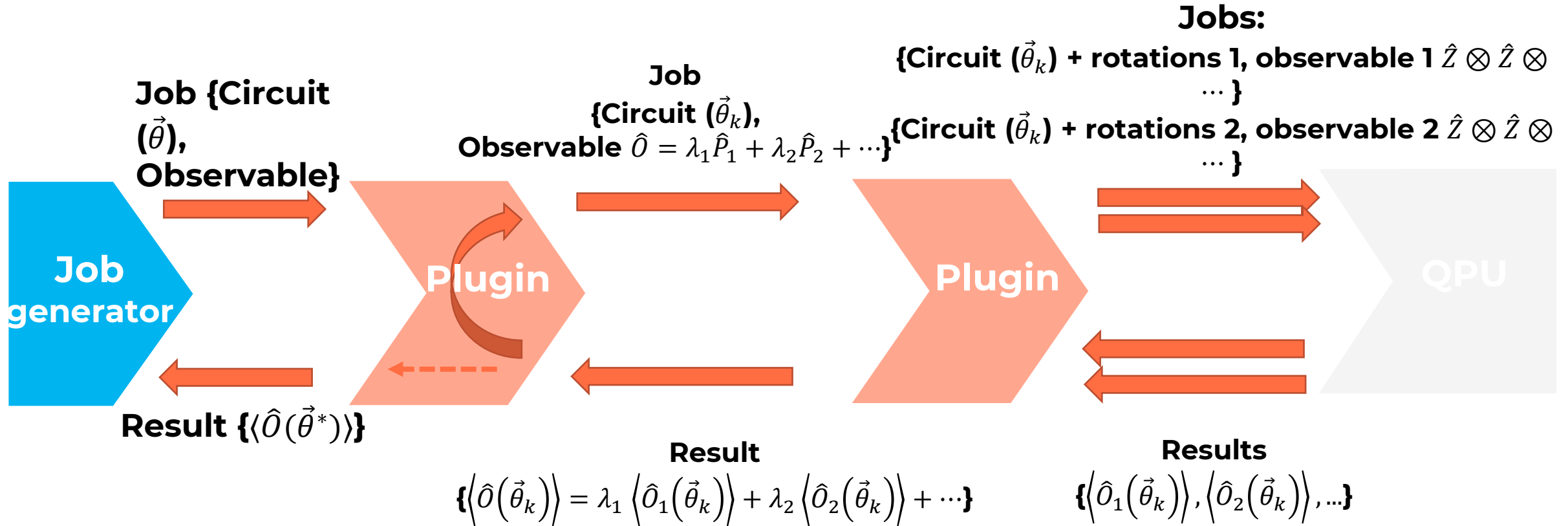
Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \dots$ observables...



Taking into account limited available observables

ObservableSplitter plugin

Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \dots$ observables...



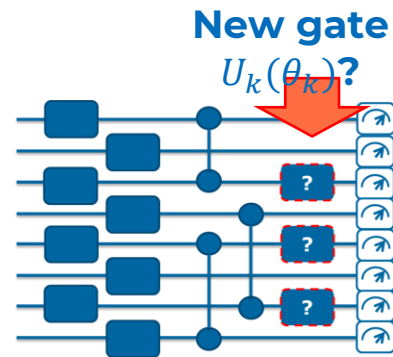
Adapt variational circuit iteratively?

AdaptVQE plugin

Build ansatz circuit iteratively:

At each step: pick gate $U(\theta)$

that maximizes gradient $\frac{d}{d\theta} \langle \psi | U^\dagger(\theta) H U(\theta) | \psi \rangle$



Adaptive construction of the variational circuit

Grimsley '19

AdaptVQE plugin

Build ansatz circuit iteratively:

At each step: pick gate $U(\theta)$

that maximizes gradient $\frac{d}{d\theta} \langle \psi | U^\dagger(\theta) H U(\theta) | \psi \rangle$

