EVIDEN

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

CEMRACS 2025

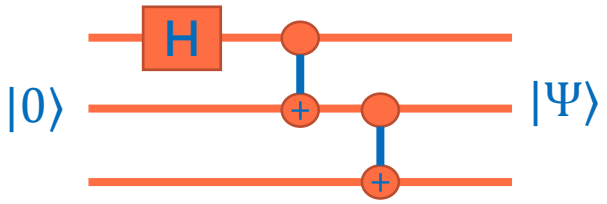
Thomas Ayral
Eviden Quantum Lab
July 16th, 2025

an atos business

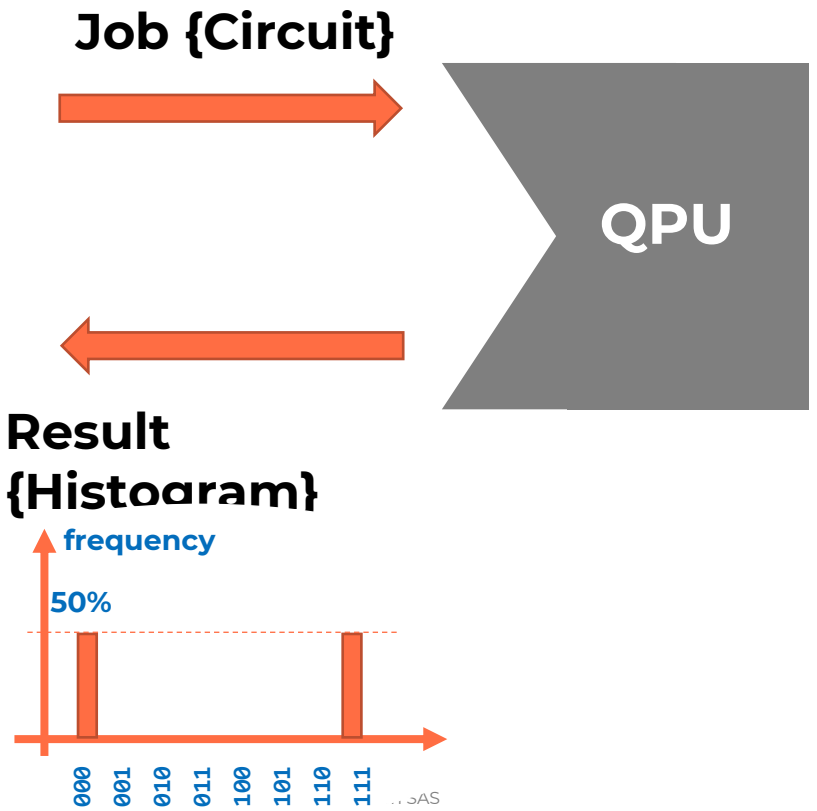
The QLM's key components

Jobs and QPUs

A basic quantum computation:



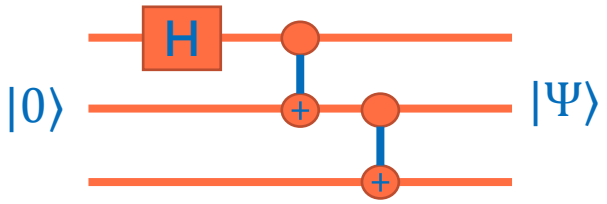

```
job = circ.to_job()
res = qpu.submit(job)
for sample in res:
    print(sample.state, sample.probability)
```



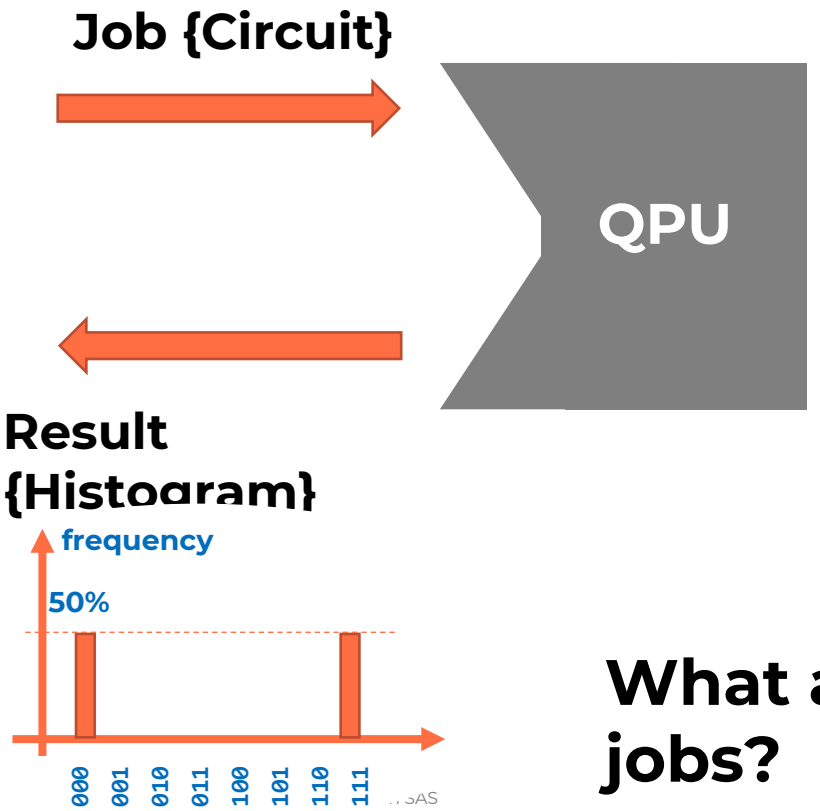
The QLM's key components

Jobs and QPUs

A basic quantum computation:




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



What about more complex jobs?

More complex jobs (I)

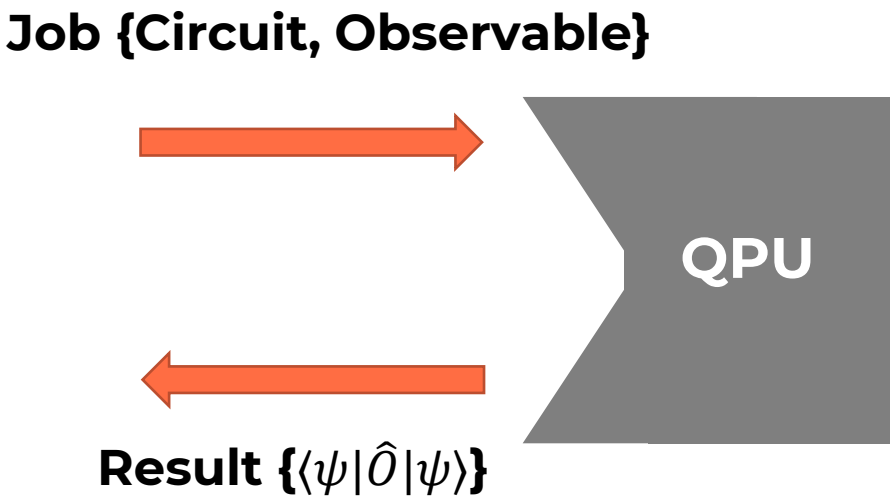
Observables

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

EVIDEN

More complex jobs (I)

Observables

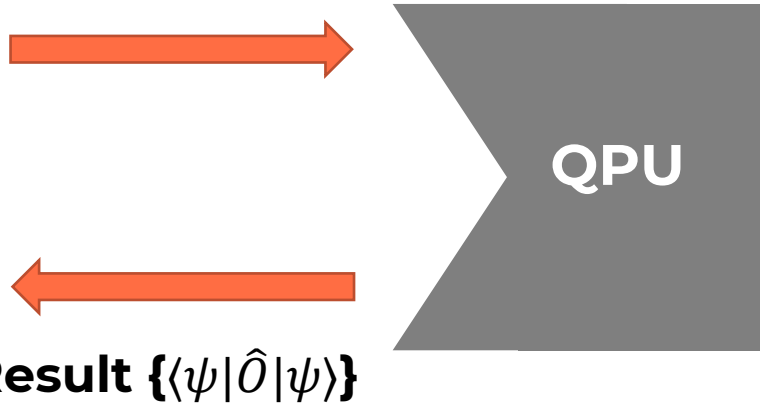
Most computations: computing/minimizing an observable:

- 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}

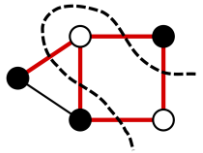


Result $\{\langle\psi|\hat{O}|\psi\rangle\}$

Qaptiva: libraries to generate quantum jobs for many application areas

- **Optimization libraries (MaxCut, etc)**

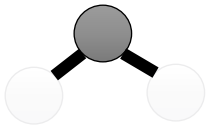
Graph



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

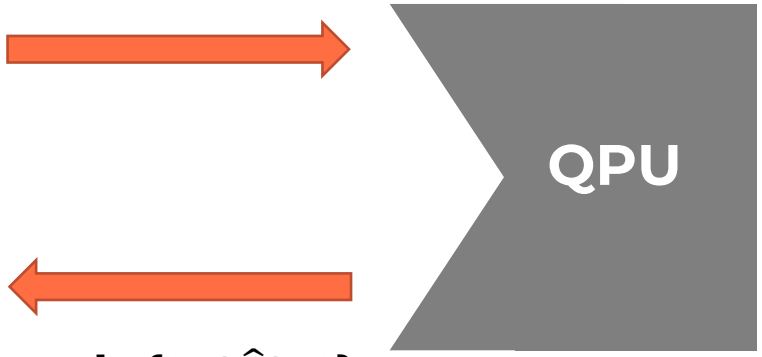
Most computations: computing/minimizing an observable:

- E.g combinatorial optimization,

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

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

Job {Circuit, Observable}

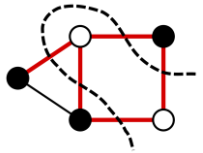


Result {$\langle\psi|\hat{O}|\psi\rangle$}

Qaptiva: libraries to generate quantum jobs for many application areas

- **Optimization libraries (MaxCut, etc)**

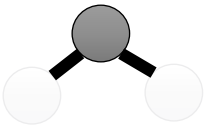
Graph



Job {Circuit, Observable}

- **Chemistry, materials science libraries (myQLM-fermion)**

Molecule



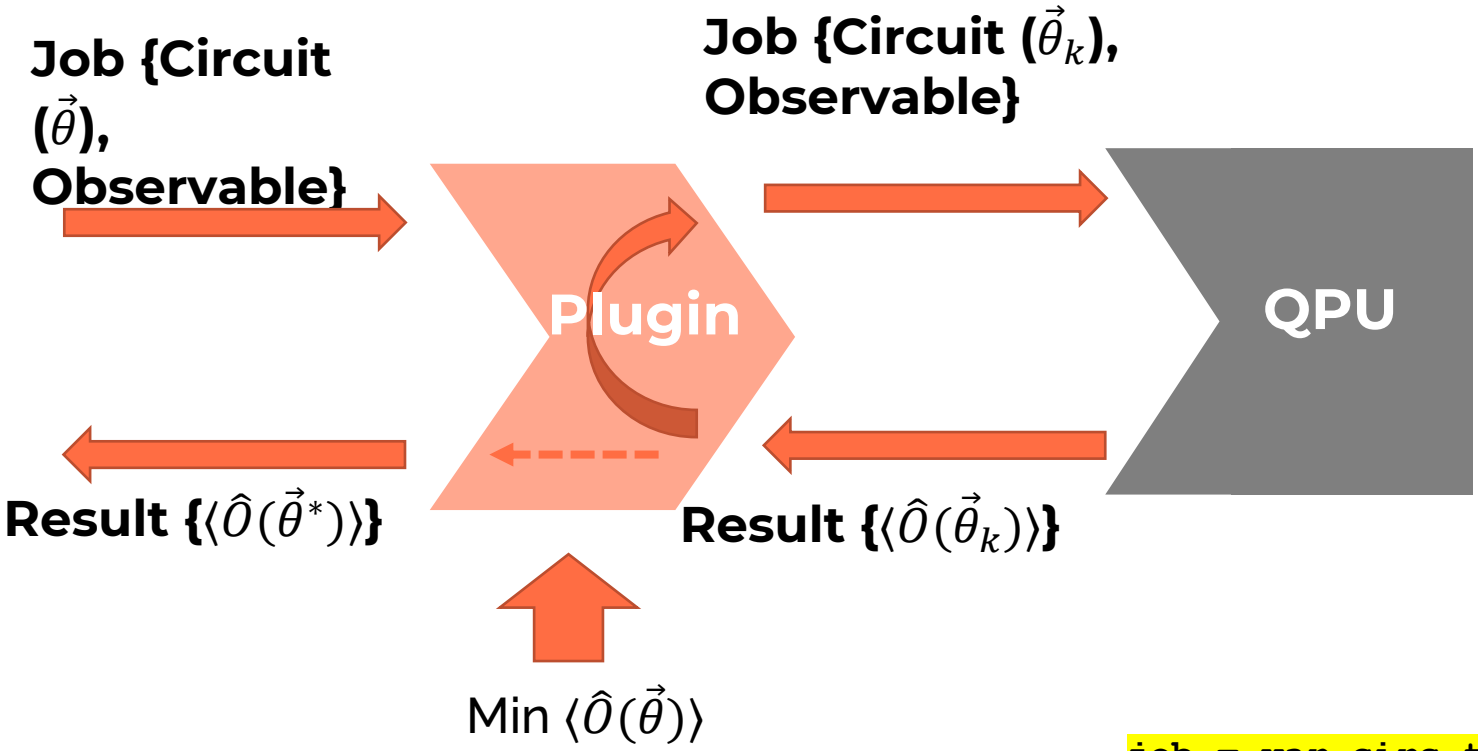
Job {Circuit, Observable}

Minimization?

More complex jobs (II)

Minimizing observables

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

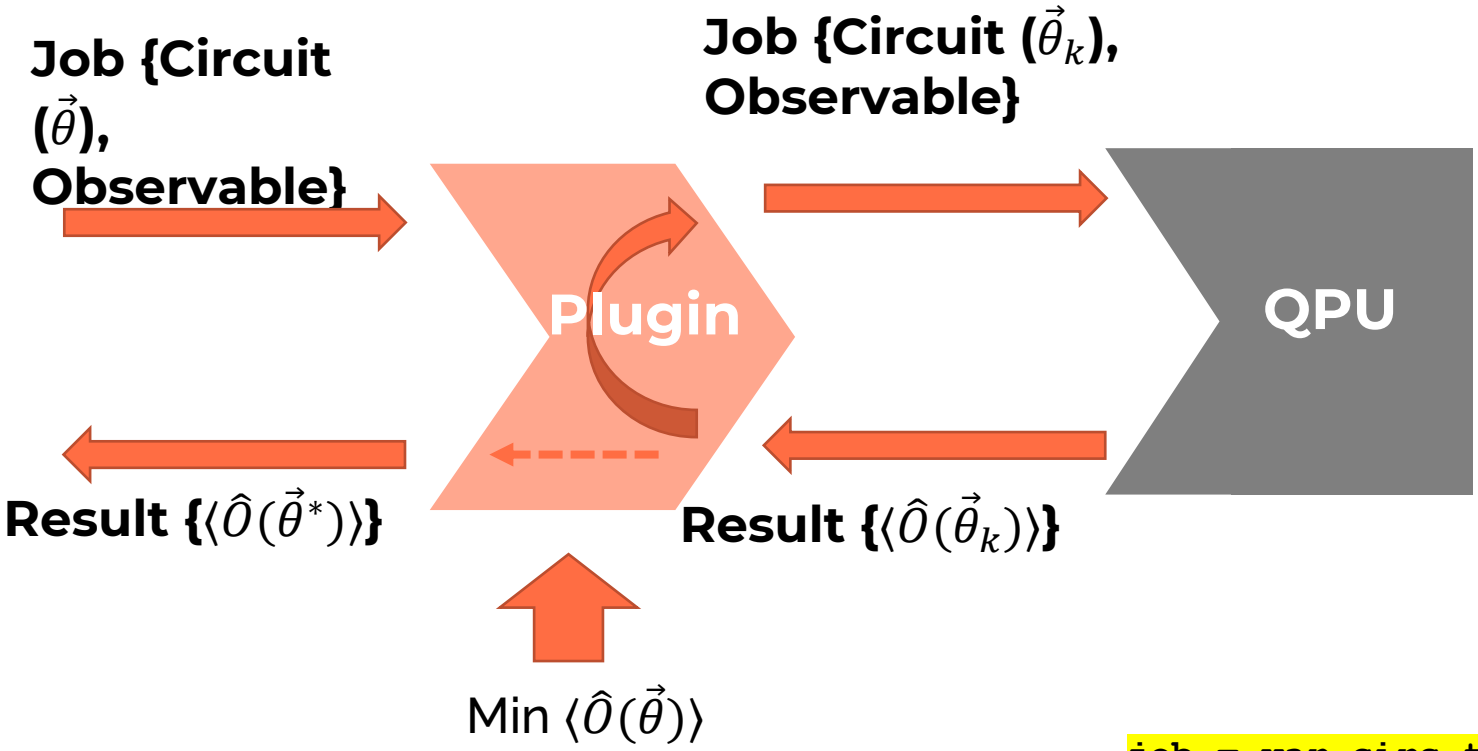



```
job = var_circ.to_job(observable=my_obs)
stack = plugin | qpu
res = stack.submit(job)
print(res.value) #<psi(theta*)|O|psi(theta*)>
```

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:

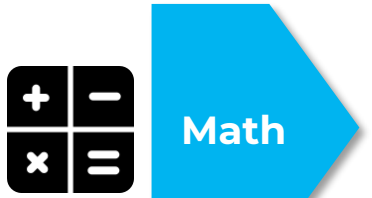
- Scipy wrappers (COBYLA, BFGS)
- Gradient-free custom plugins: SPSA, Particle-Swarm Optimizer…
- Gradient-based optimization
- Natural gradient
- Etc.

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

EVIDEN

A Qaptiva computation in a nutshell

(1) Pick your application



Math

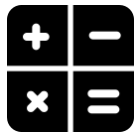
Combinatorial optimization

Chemistry

(etc)

A Qaptiva computation in a nutshell

(1) Pick your application



Math

Combinatorial optimization

Chemistry

(etc)

(2) Pick your QPU

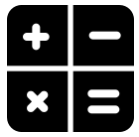
Perfect simulator (LinAlg, MPS...)

Noisy simulator (deterministic, stochastic...)

Actual QPU (IBM, IQM, Pasqal...)

A Qaptiva computation in a nutshell

(1) Pick your application



Math

Combinatorial optimization

Chemistry

(etc)

(3) Pick your plugins

Adapt VQE

Gradient-free optimizer

Gradient-based optimizer

Natural gradient

SeqOptim (rotosolve)

NISQ Compiler (Connectivity, gateset)

Observable Splitter

Error mitigation (Zero-noise, etc)

(etc)

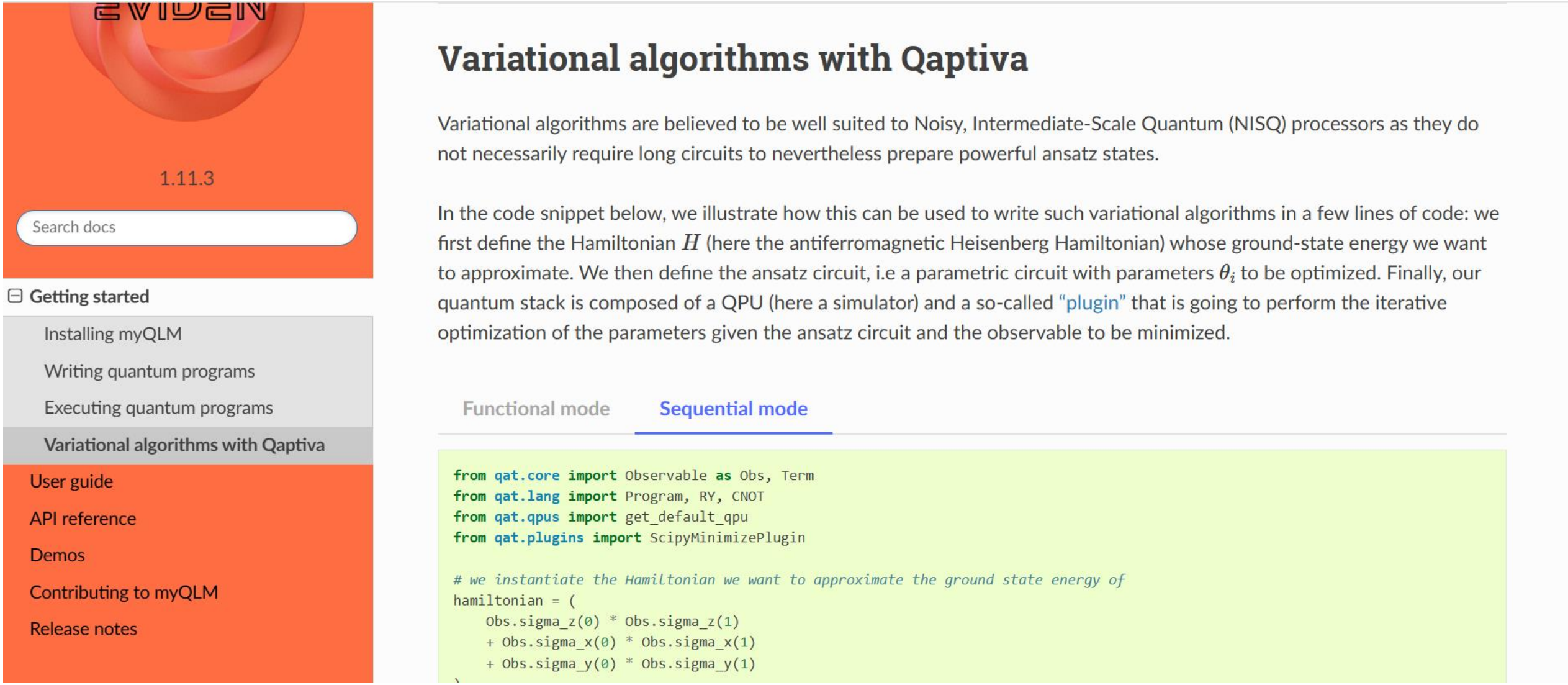
(2) Pick your QPU

Perfect simulator (LinAlg, MPS...)

Noisy simulator (deterministic, stochastic...)

Actual QPU (IBM, IQM, Pasqal...)

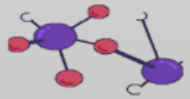
Documentation: https://myqlm.github.io/



EVIDEN

myQLM-fermion

Quantum chemistry



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

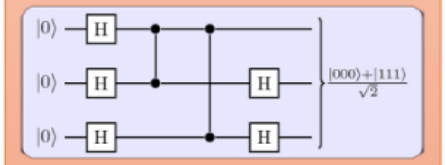
From fermions to qubits

. 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 $e^{T(\vec{\theta})}$ → circuit $C(\vec{\theta})$

Quantum job definition



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

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

The QLM stack

Job $\{C(\vec{\theta}), H_{sp}\}$ number of shots

$E(\vec{\theta}^*)$

Variational plugin
Minimize $E(\vec{\theta}) = \langle\psi(\vec{\theta})|H_{\mathrm{sp}}|\psi(\vec{\theta})\rangle$
Update $\vec{\theta}_k \rightarrow \vec{\theta}_{k+1}$ until convergence to $\vec{\theta}^*$

$\{C(\vec{\theta}_k), H_{sp}\}$

[Gradients] $[\partial_\theta E(\vec{\theta}_k)]$

Compilation plugins
. Connectivity
. Gateset
. Qubit ordering
. Depth optimization
. etc

$E(\vec{\theta}_k)$

$\{\widetilde{C}(\vec{\theta}_k), \widetilde{H}_{sp}\}$

Observable Splitter plugin
Generate jobs to compute $\sum_i \lambda_i \langle\psi(\vec{\theta}_k)|\widetilde{P}_i|\psi(\vec{\theta}_k)\rangle$ (Pauli operator grouping, …)

$E(\vec{\theta}_k)$

$\{\widetilde{C}_M(\vec{\theta}_k), Z_{i_1} \ldots Z_{i_M}\}_M$

QPU
. Perfect simulators (Schrödinger, MPS, stabilizer…)
. Noisy simulators (gate/SPAM/idle noise)
. Interface to true QPUs: IBMQ, Google, Rigetti, …

$\{\langle Z_{i_1} \ldots Z_{i_M}\rangle(\vec{\theta}_k)\}_M$

EVIDEN

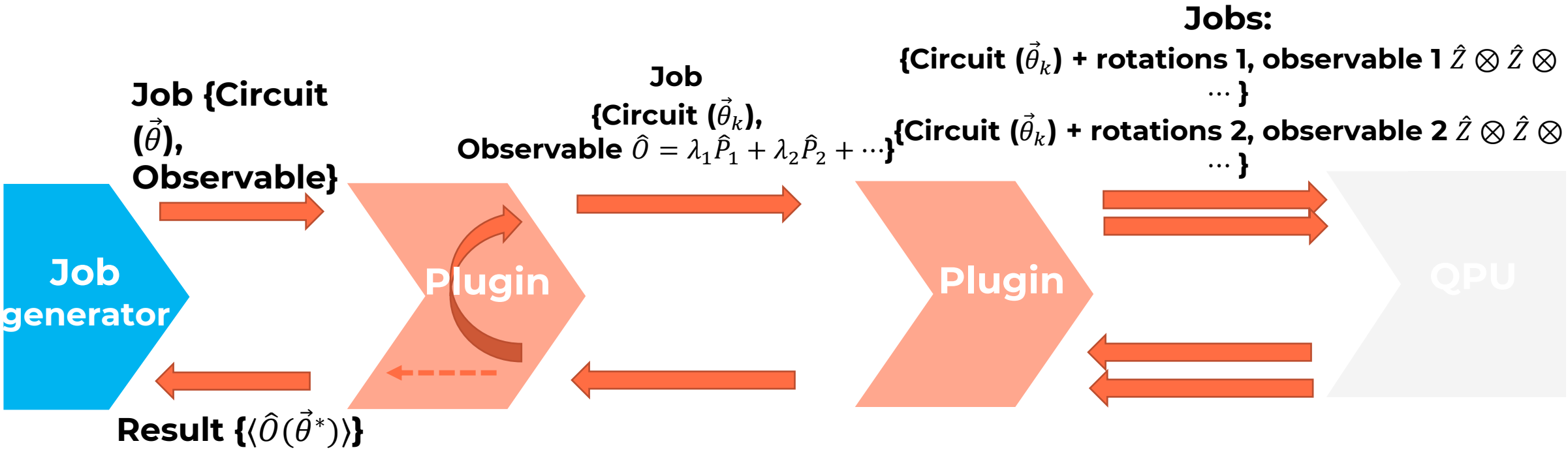
Thank you

thomas.ayral@eviden.com

Taking into account limited available observables

ObservableSplitter plugin

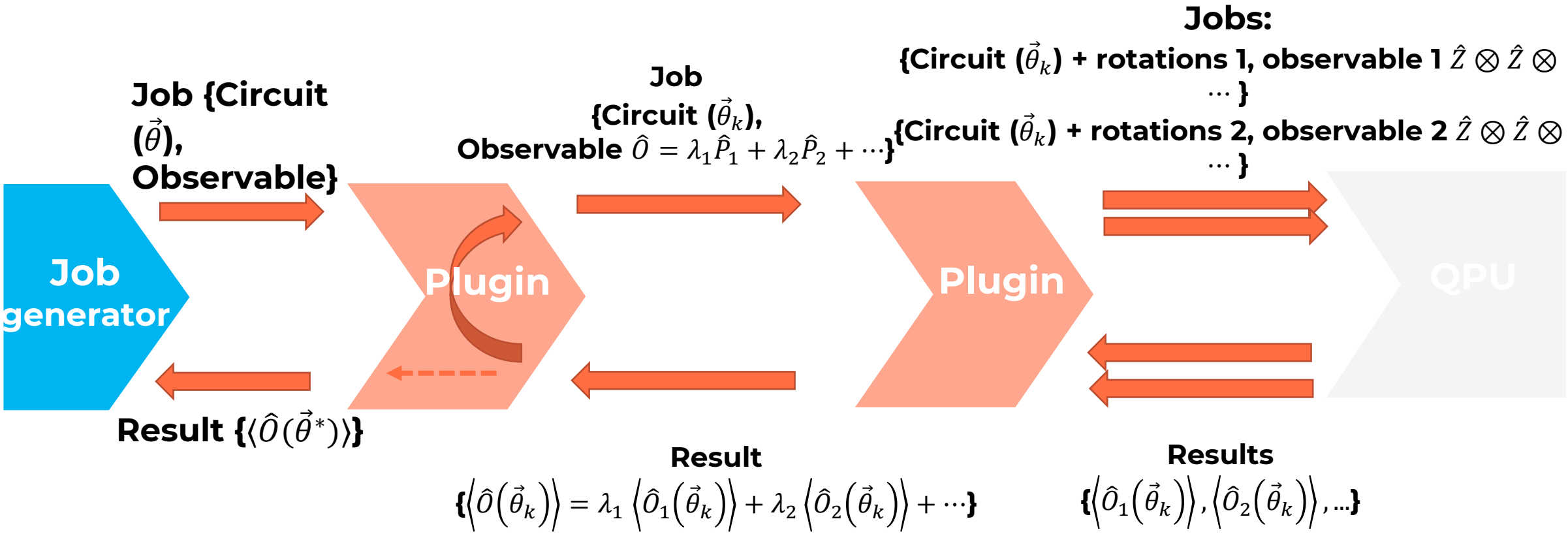
Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \cdots$ observables...



Taking into account limited available observables

ObservableSplitter plugin

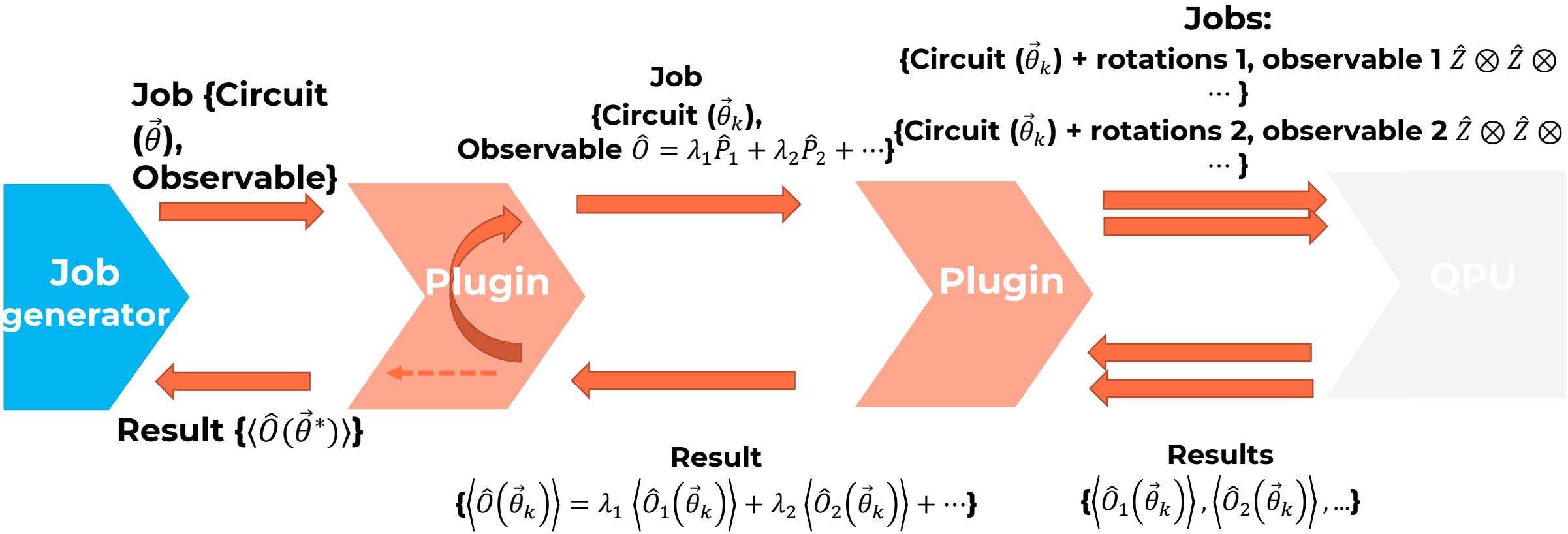
Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \cdots$ observables...



Taking into account limited available observables

ObservableSplitter plugin

Most QPUs: can measure only $\hat{Z} \otimes \hat{Z} \otimes \cdots$ observables…



Adapt variational circuit iteratively?

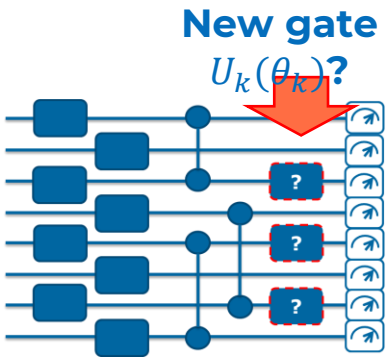
EVIDEN

Adaptive construction of the variational circuit

Grimsley ‘19

AdaptVQE plugin

Build ansatz circuit iteratively:
At each step: pick gate $U(\theta)$
that maximizes gradient $\frac{d}{d\theta}\langle\psi|U^\dagger(\theta)HU(\theta)|\psi\rangle$

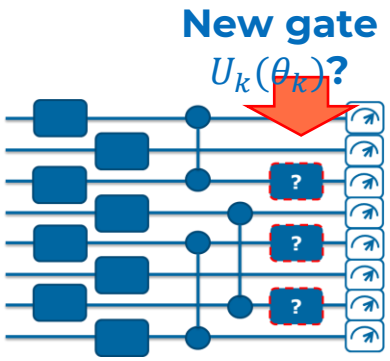


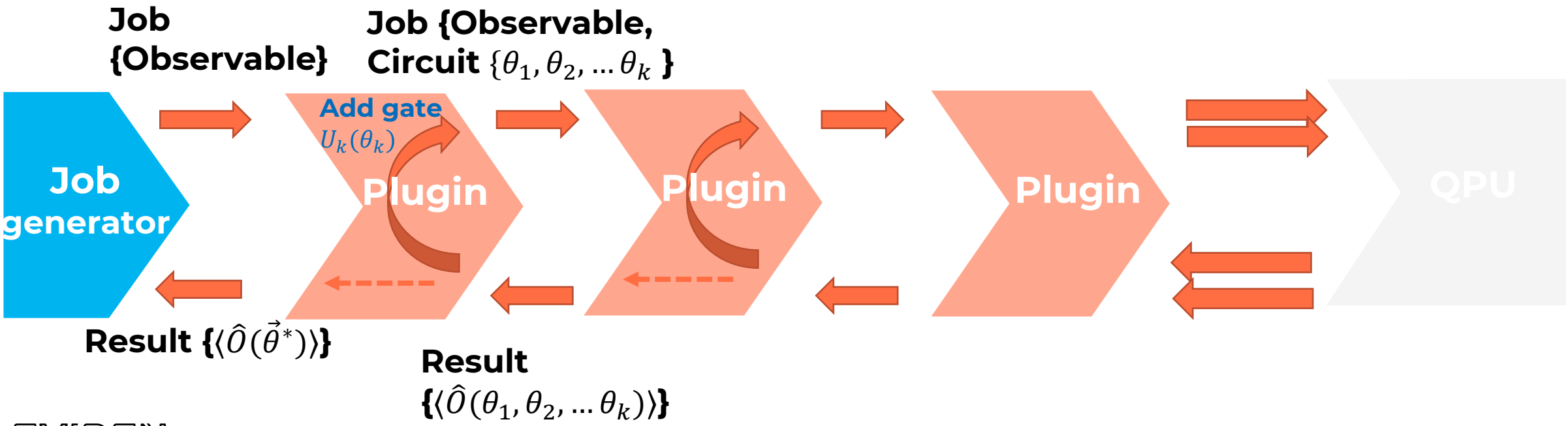
Adaptive construction of the variational circuit

Grimsley '19

AdaptVQE plugin

Build ansatz circuit iteratively:
At each step: pick gate $U(\theta)$
that maximizes gradient $\frac{d}{d\theta}\langle\psi|U^\dagger(\theta)HU(\theta)|\psi\rangle$





EVIDEN