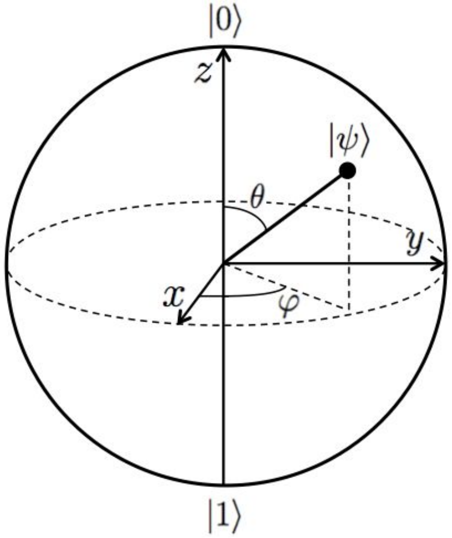
Quantum Computing and Track Reconstruction

Miriam Lucio Martínez

Quantum Computing in a nutshell

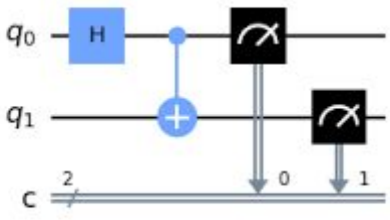
- Instead of **bits** we use **qubits**, the fundamental units of quantum information
  - Not 0 **or** 1, but a two-state quantum system → coherent superposition of both
  - They can be **measured** → probabilistic results
- There are **quantum logic gates** that operate on these qubits
  - Unitary transformations
  - Quantum gates can be **single** or **multiple**



$$|0\rangle = \binom{1}{0},\ |1\rangle = \binom{0}{1}$$
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

Quantum Computing in a nutshell

A sequence of gates acting on a register of qubits is called a quantum circuit



Some computational problems can profit from Quantum Computing using the principles of superposition and interference.

https://quantumalgorithmzoo.org/

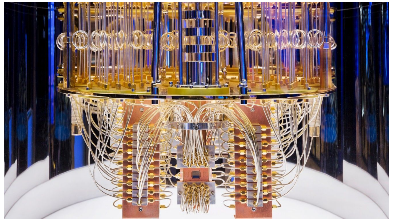
Quantum Computing - Hardware

Several technologies are being explored as physical qubits:

Superconducting

IBM Google

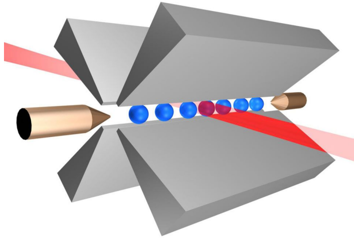
Superconducting electric circuits at 10mK behave as quantum systems with discrete energy levels



Trapped ions

IONQ QUANTINUUM

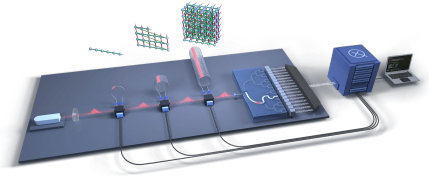
Charged atoms constrained in electromagnetic traps and manipulated with laser



Optical

XANADU

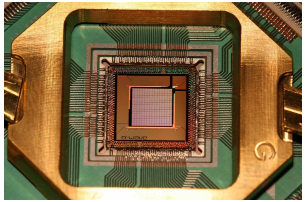
Linear optics devices using photons as information carriers



Annealing

D-Wave
The Quantum Computing Company™

Ising-chain qubits interacting with a customizable Hamiltonian

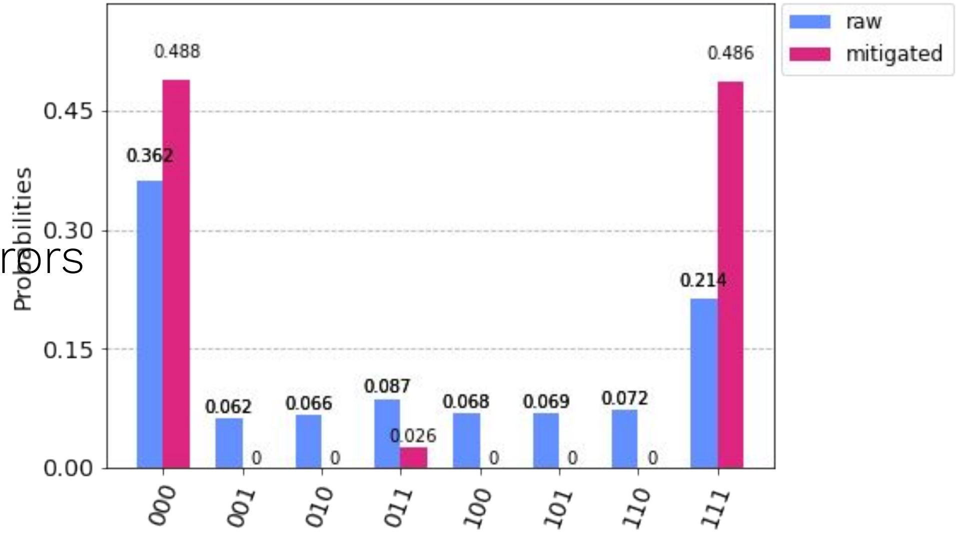


Quantum Computing - Noise

All the previous technologies are far from being perfect. Current qubits are **noisy**:

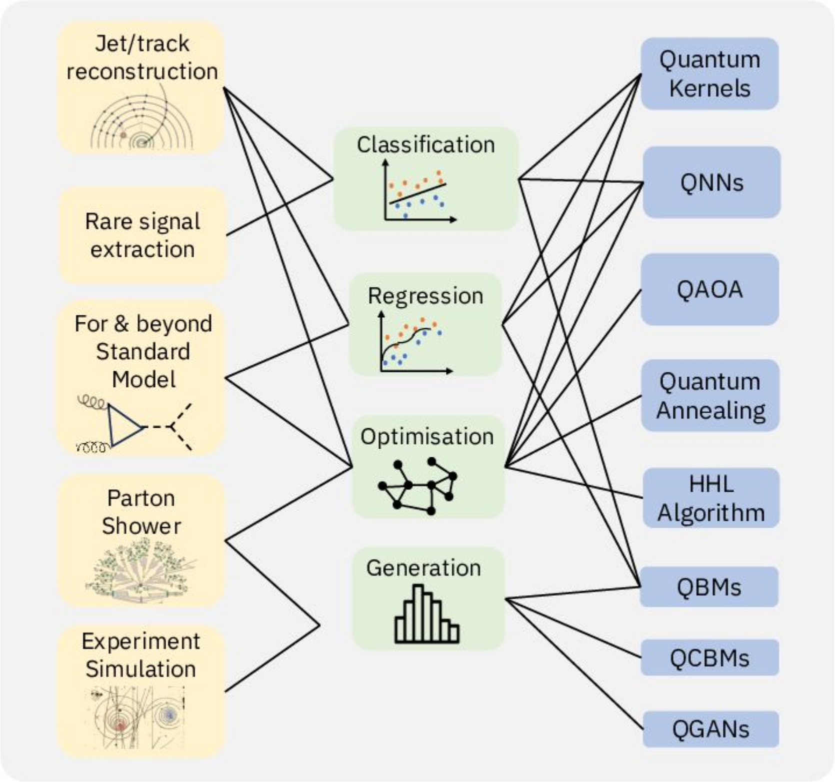
- Measurement errors
- 1-qubit and 2-qubit gates fidelities
- T1 and T2 decoherence time
- Calibration errors

→ *Noise Error Mitigation*



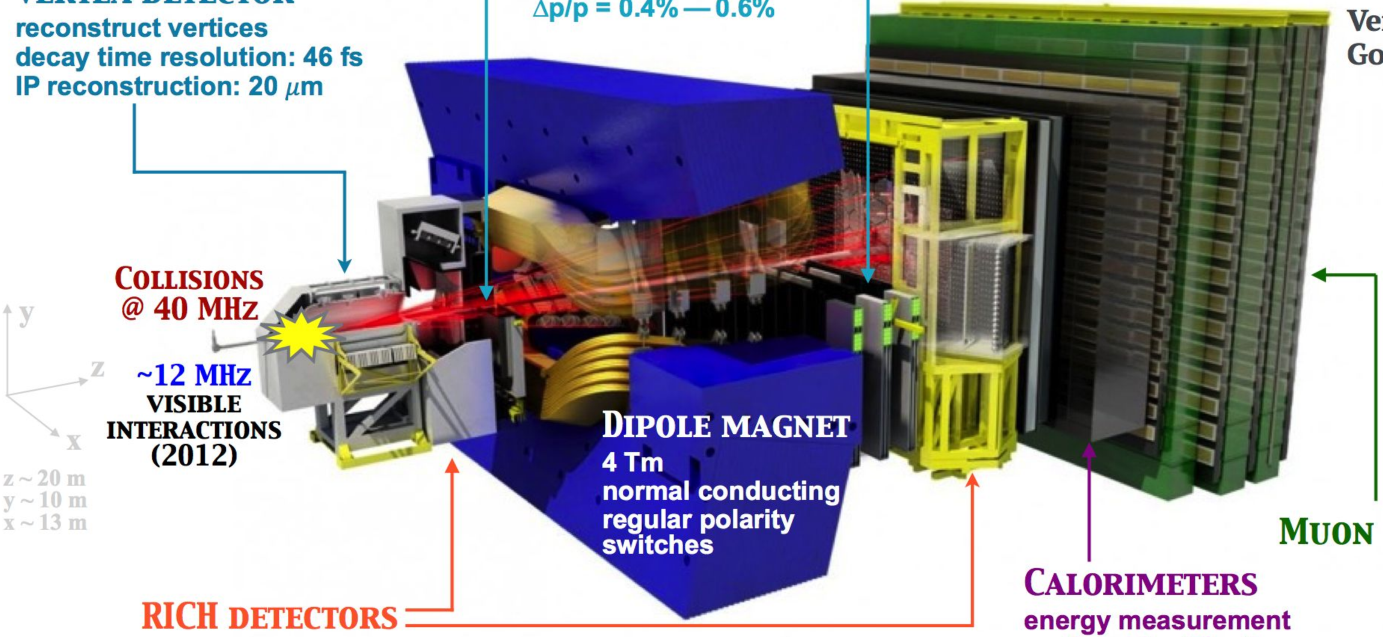
HEP use-cases

- Summary of the QC4HEP WG
- Focused mostly in projects concerning experimental particle physics at LHC and LHCb
- Events are **quantum** in nature, but measurements are **classical**
- Quantum sensing not covered in this talk



The LHCb detector

Single forward-arm spectrometer



- **TWO-LEVEL TRIGGER:**
  - L0 hardware (12 → 1 MHz)
  - HLT software (1 → 0.005 MHz)

Very good $\varepsilon(\mu)$
Good $\varepsilon(h)$

VERTEX DETECTOR
reconstruct vertices
decay time resolution: 46 fs
IP reconstruction: 20 μm

TRACKING SYSTEM
momentum resolution
$\Delta p/p = 0.4\% — 0.6\%$

COLLISIONS
@ 40 MHz

~12 MHz
VISIBLE
INTERACTIONS
(2012)

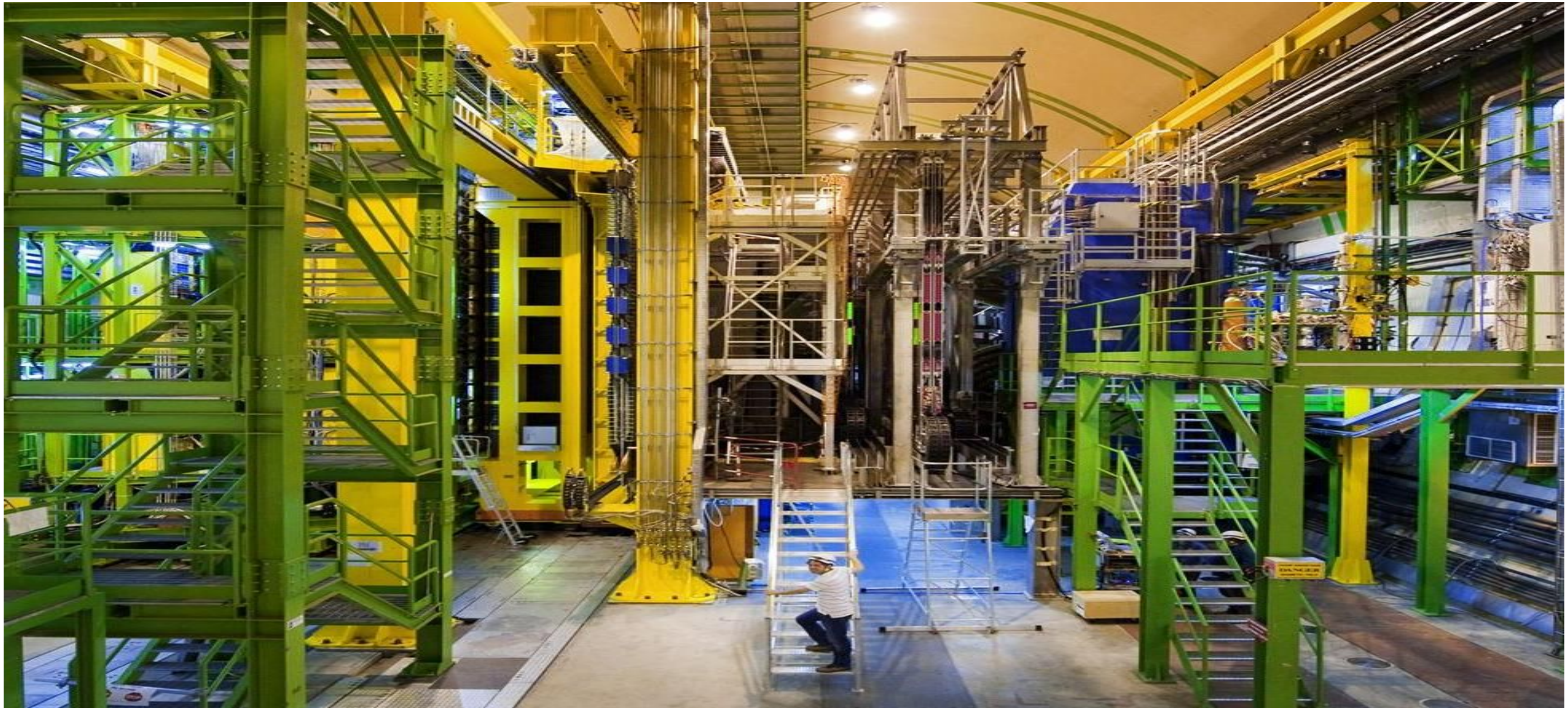
DIPOLE MAGNET
4 Tm
normal conducting
regular polarity switches

RICH DETECTORS
K/π/p separation

CALORIMETERS
energy measurement
particle identification

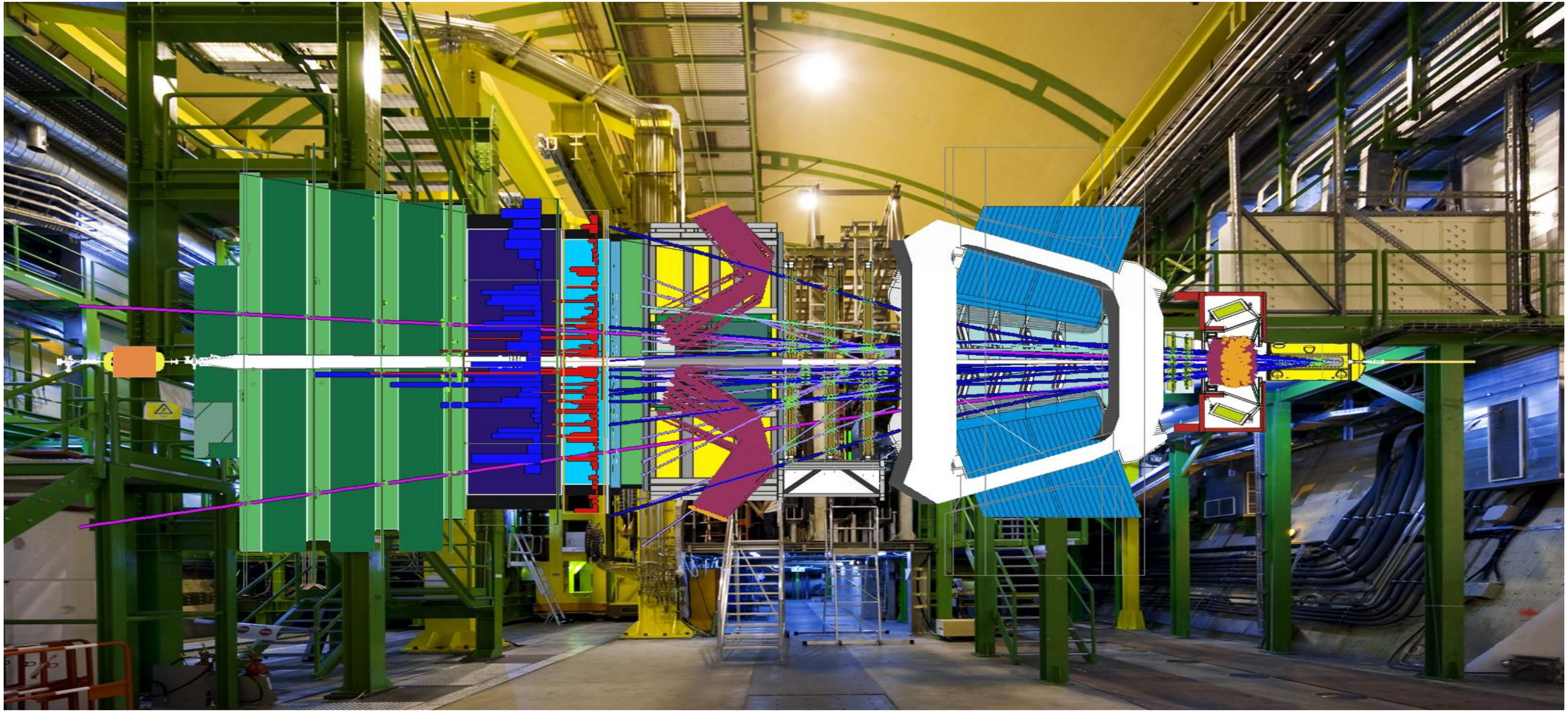
MUON SYSTEM

How does an event look like?



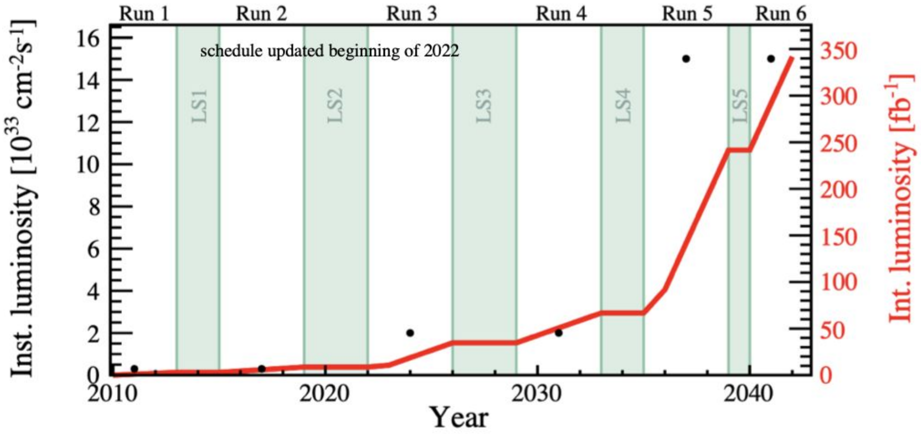
How does an event look like?

Reconstruct events **40 Million times per second**.

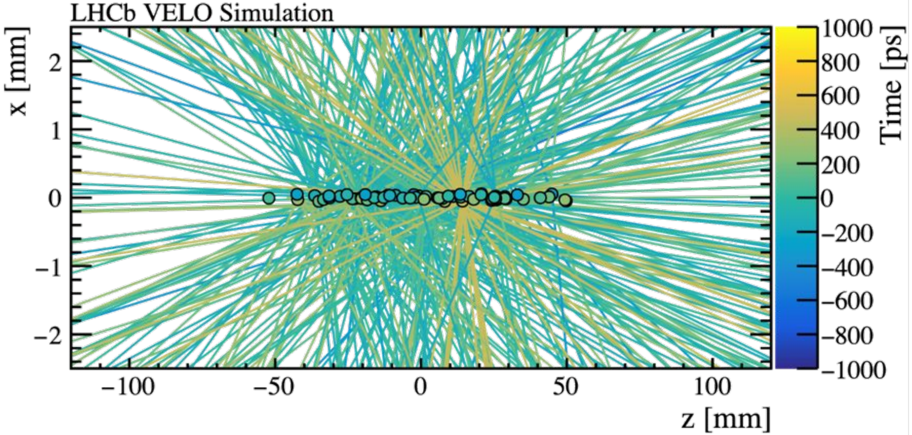


Motivation for QC

- New algorithms and architectures needed to deal with the increased luminosity & limited bandwidth @ **HL-LHC**



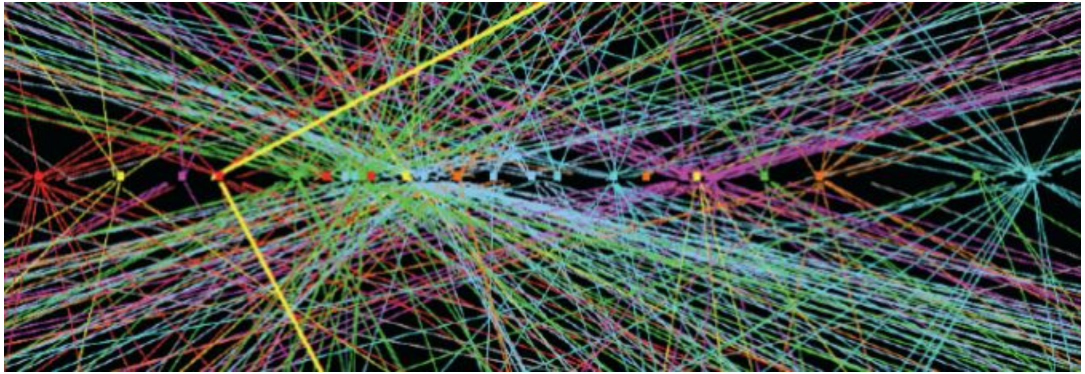
ECFA



Courtesy of Robbert Geertsema

Track reconstruction

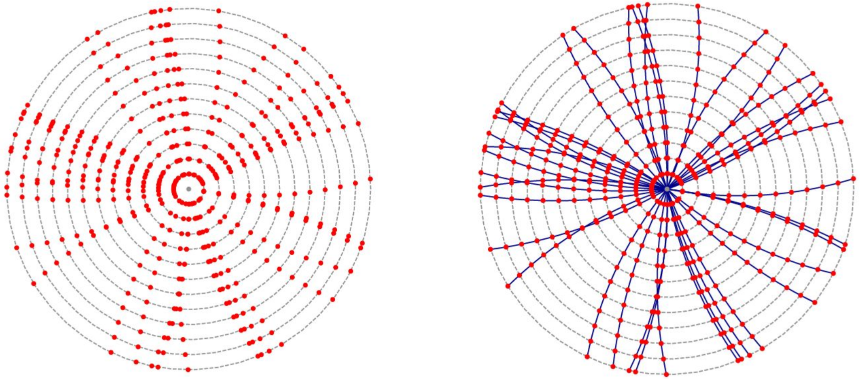
- Recover the original trajectories from signals left by charged particles
    - Signals are converted into 3D points called hits
    - Need efficient distinction between the combinations of hits that are of interest and those that aren't
- A typical HEP event contains a large number of tracks
- Tracks are modelled by a collection of segments



Track Reconstruction

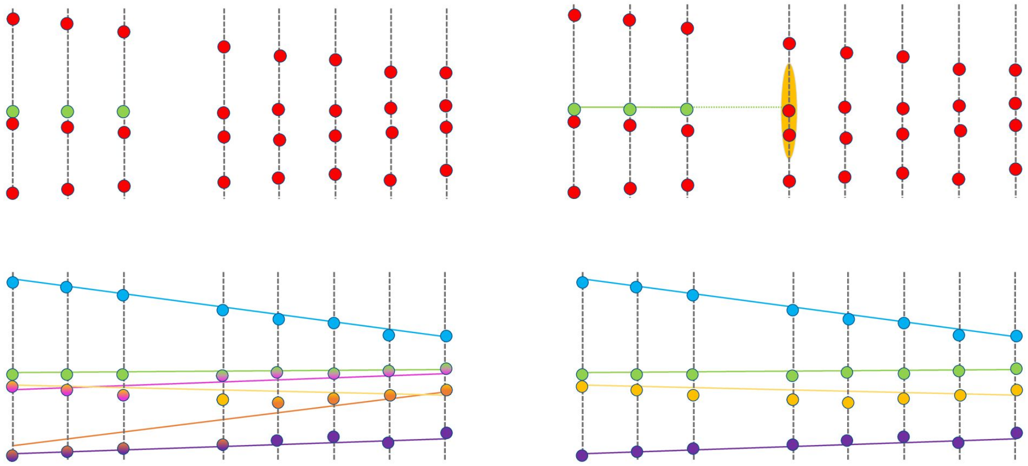
- **Local tracking methods**: steps are performed sequentially. Some studies exist on QC for local tracking methods [arXiv:2104.11583]
- **Global tracking methods**: all hits are processed by the algorithm in the same way. Global algorithms are **clustering** algorithms. E.g.: QAOA, quantum annealing, Hopfield Networks, Hough transform

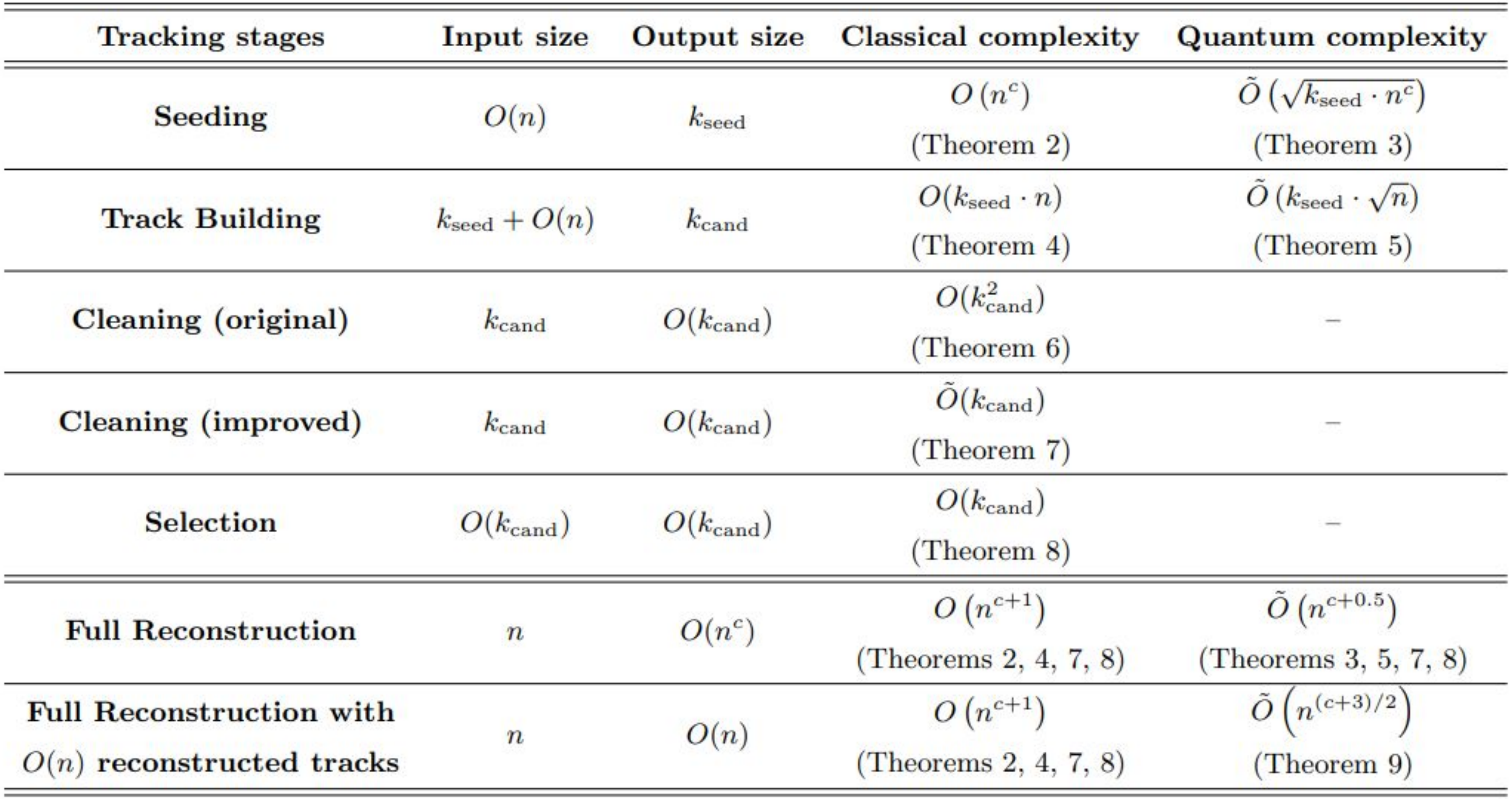
→ Focus of this talk: *global* algorithms



Local tracking methods [arXiv:2104.11583]

1. Seeding
2. Track building
3. Cleaning
4. Selection





Tracking stages	**Input size**	**Output size**	**Classical complexity**	**Quantum complexity**
Seeding	$O(n)$	k_{seed}	$O\left(n^c\right)$ (Theorem 2)	$\tilde{O}\left(\sqrt{k_{\text{seed}}} \cdot n^c\right)$ (Theorem 3)
Track Building	$k_{\text{seed}} + O(n)$	k_{cand}	$O(k_{\text{seed}} \cdot n)$ (Theorem 4)	$\tilde{O}\left(k_{\text{seed}} \cdot \sqrt{n}\right)$ (Theorem 5)
Cleaning (original)	k_{cand}	$O(k_{\text{cand}})$	$O(k_{\text{cand}}^2)$ (Theorem 6)	–
Cleaning (improved)	k_{cand}	$O(k_{\text{cand}})$	$\tilde{O}(k_{\text{cand}})$ (Theorem 7)	–
Selection	$O(k_{\text{cand}})$	$O(k_{\text{cand}})$	$O(k_{\text{cand}})$ (Theorem 8)	–
Full Reconstruction	n	$O(n^c)$	$O\left(n^{c+1}\right)$ (Theorems 2, 4, 7, 8)	$\tilde{O}\left(n^{c+0.5}\right)$ (Theorems 3, 5, 7, 8)
Full Reconstruction with $O(n)$ reconstructed tracks	n	$O(n)$	$O\left(n^{c+1}\right)$ (Theorems 2, 4, 7, 8)	$\tilde{O}\left(n^{(c+3)/2}\right)$ (Theorem 9)

n: number of particles, c: number of hits, k_{seed}: total number of generated seeds, k_{cand}: number of track candidates

QC for Track Reconstruction

- QC has very interesting prospects of improvements in algorithm **complexity/timing**
- This talk: two track reconstruction algorithms
- Define **Ising-like** $H^{\text{TrackReco}}$(hits):

$$H = -\frac{1}{2}\sum_{ij}\omega_{ij}\sigma_z^i\sigma_z^j - \sum_i \omega_i \sigma_z^i$$

→ $\mathbf{H_{min}^{TrackReco}}$ == solution with the correct reconstructed tracks

HHL for Track Reconstruction [arXiv:2308.00619]

Differentiable Hamiltonian:

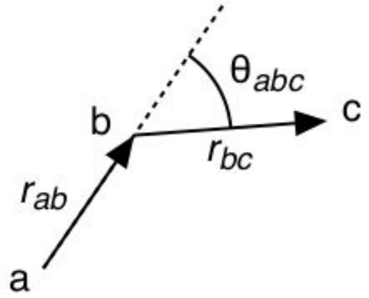
$$\nabla\mathcal{H} = 0 \Rightarrow A\mathbf{S} = \mathbf{b}$$

HHL: QC algorithm to solve the **system of linear equations**

Segment [S_{ab}]: combination of hit *a* and hit *b*

→ in consecutive layers - for now

Hamiltonian accounts for **all** possible segments



HHL for Track Reconstruction [arXiv:2308.00619]

$$\mathcal{H}(\mathbf{S}) = -\frac{1}{2}\left[\sum_{abc} f(\theta_{abc},\varepsilon)S_{ab}S_{bc} + \gamma\sum_{ab} S_{ab}^2 + \delta\sum_{ab}(1-2S_{ab})^2\right]$$

angular term (a) (b)

$$f(\theta_{abc},\varepsilon) = \begin{cases} 1 & \text{if } \cos\theta_{abc} \geq 1-\varepsilon \\ 0 & \text{otherwise} \end{cases}$$

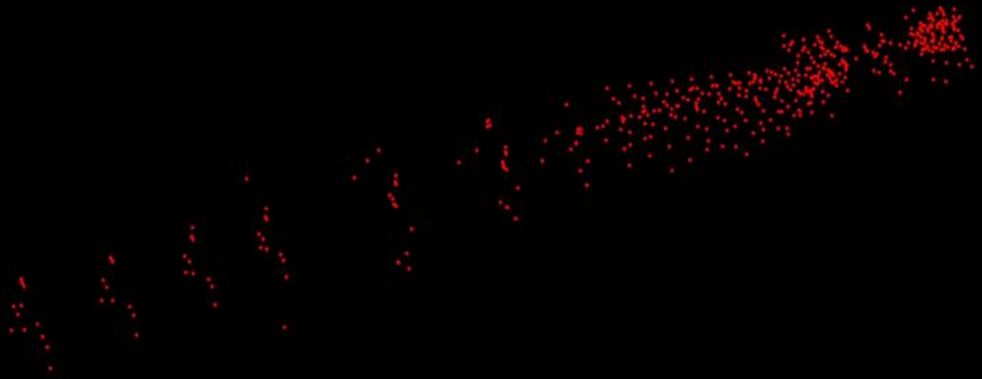
- **(a) regularization term**: makes the spectrum of A positive
- **(b) gap term**: ensures gap in the solution spectrum

Validation with a classical linear solver

LHCb MC event $B_s \rightarrow \phi\phi$

1 collision event

Half of the VELO

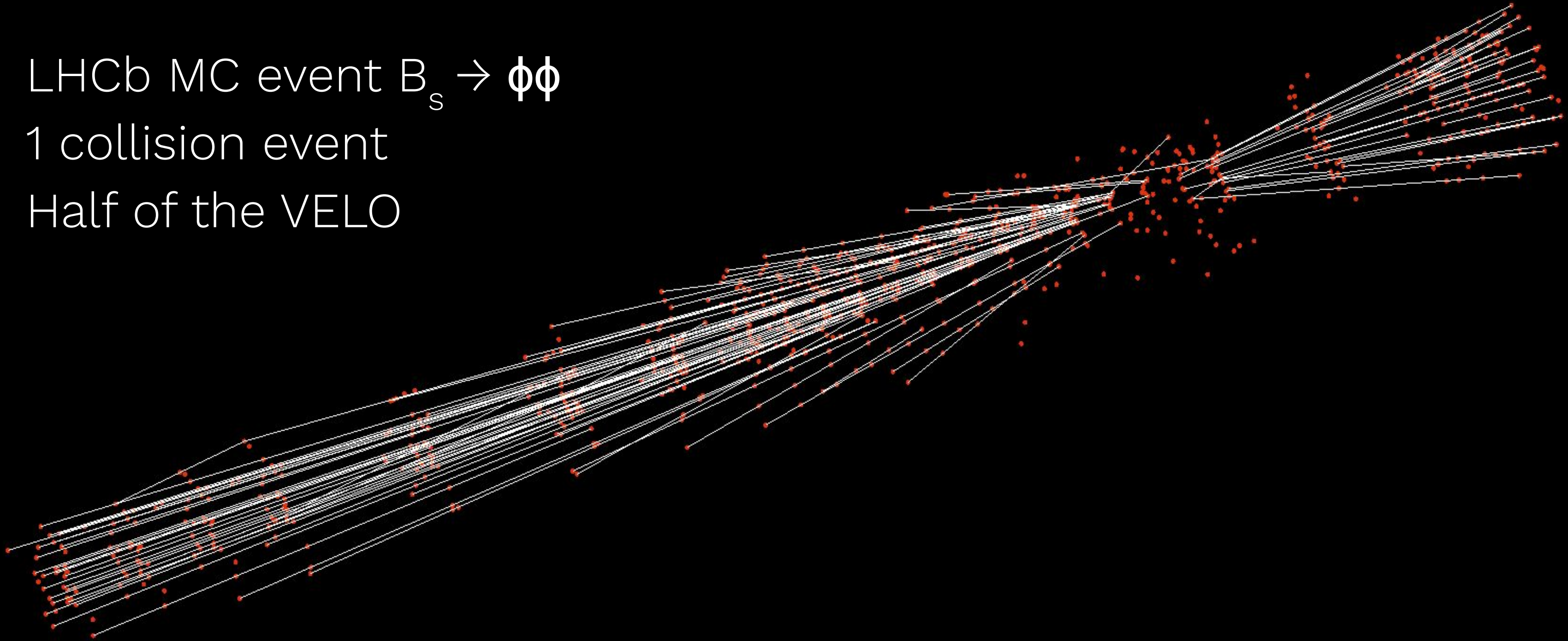


Validation with a classical linear solver

LHCb MC event $B_s \to \phi\phi$

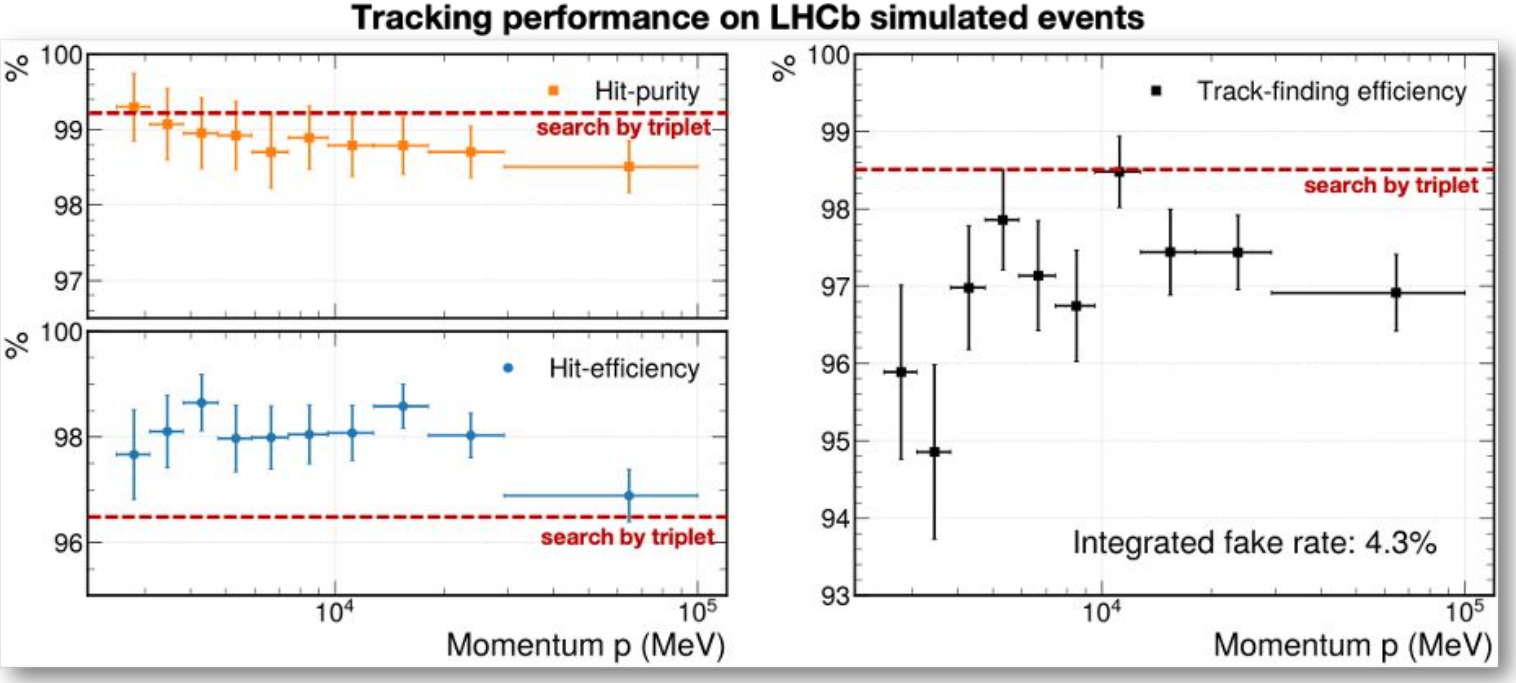
1 collision event

Half of the VELO



Tracking performances with classical solver

- Very good performance **with LHCb MC**, but **high** circuit depth.



HHL on a quantum simulator

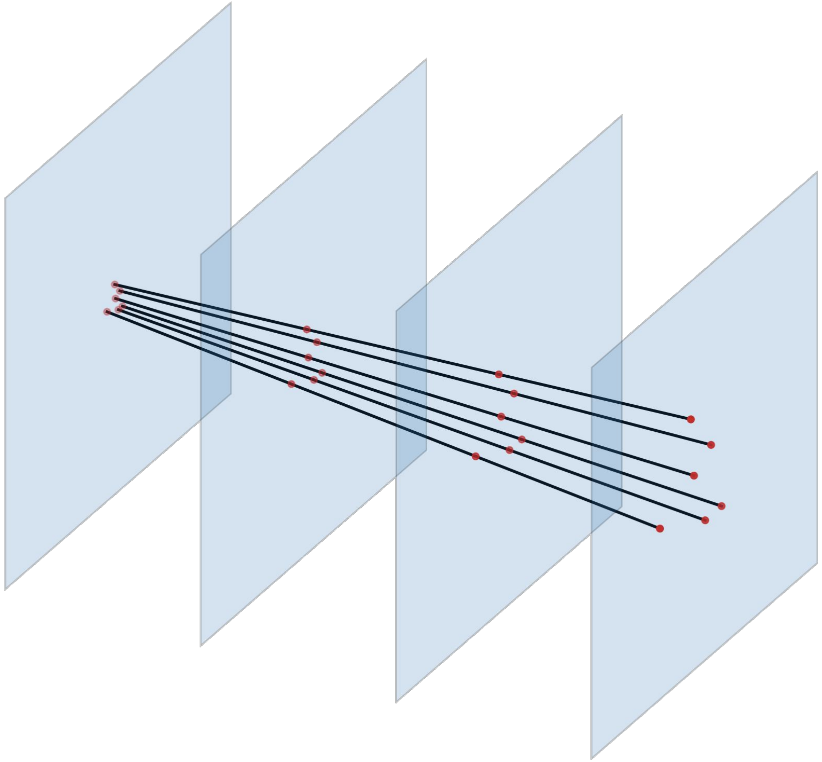
Qiskit

6 particles, 4 detector layers

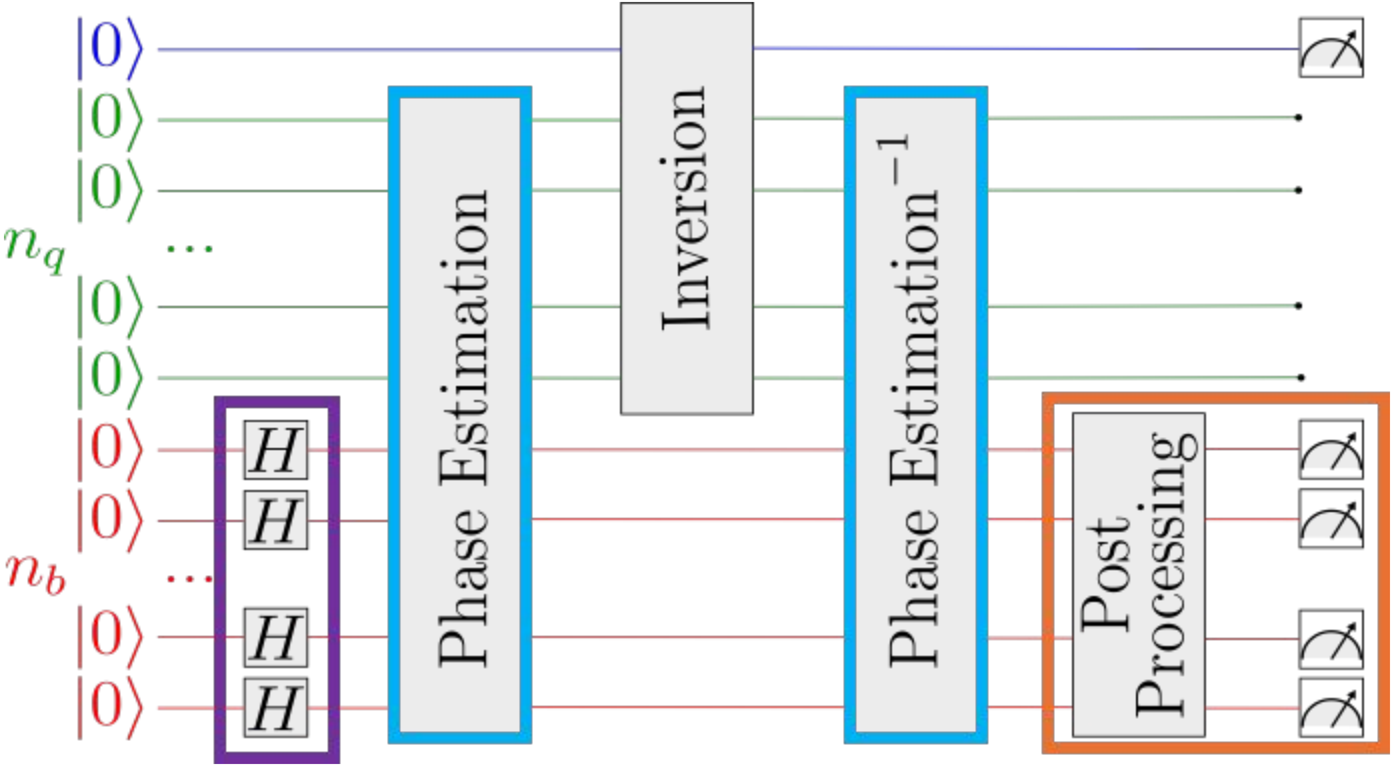
→ very complex, deep circuit

→ 2 days to complete

- Validate with classical baseline ✅
- Toy simulation on qiskit ✅
- Integrate within Allen ✅ (*)
- Scalability & Hamiltonian simulation **ongoing**



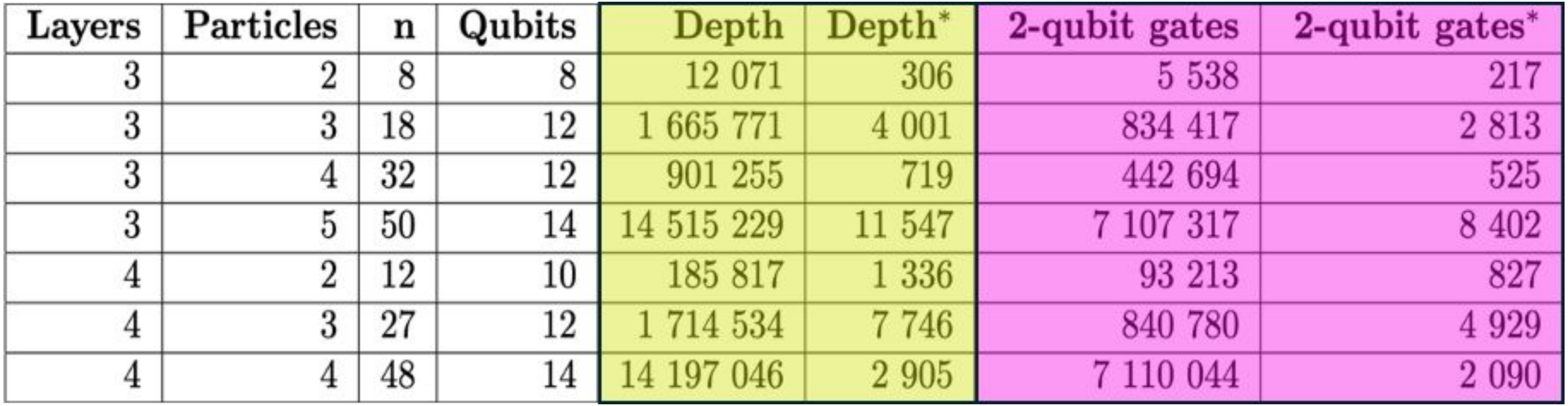
The Quantum Circuit for HHL



Updates since publication

Work ongoing to improve circuit depth **(Xenofon)**

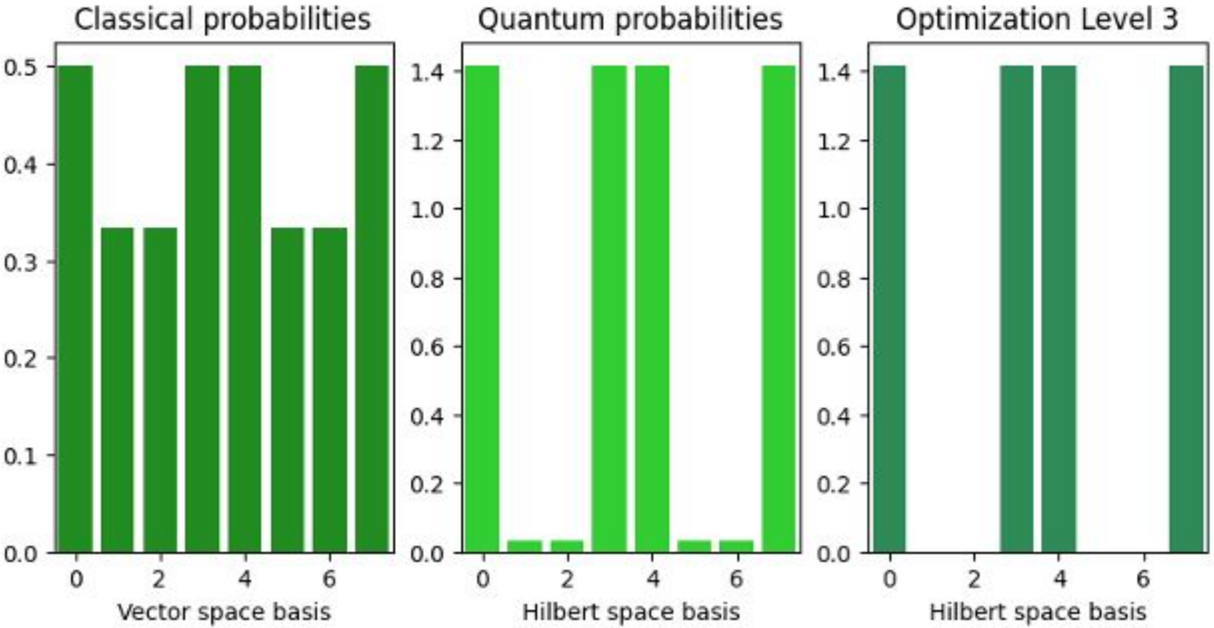
- original → 1-bit phase estimation + Suzuki-trotter decomposition



Layers	Particles	n	Qubits	Depth	Depth*	2-qubit gates	2-qubit gates*
3	2	8	8	12 071	306	5 538	217
3	3	18	12	1 665 771	4 001	834 417	2 813
3	4	32	12	901 255	719	442 694	525
3	5	50	14	14 515 229	11 547	7 107 317	8 402
4	2	12	10	185 817	1 336	93 213	827
4	3	27	12	1 714 534	7 746	840 780	4 929
4	4	48	14	14 197 046	2 905	7 110 044	2 090

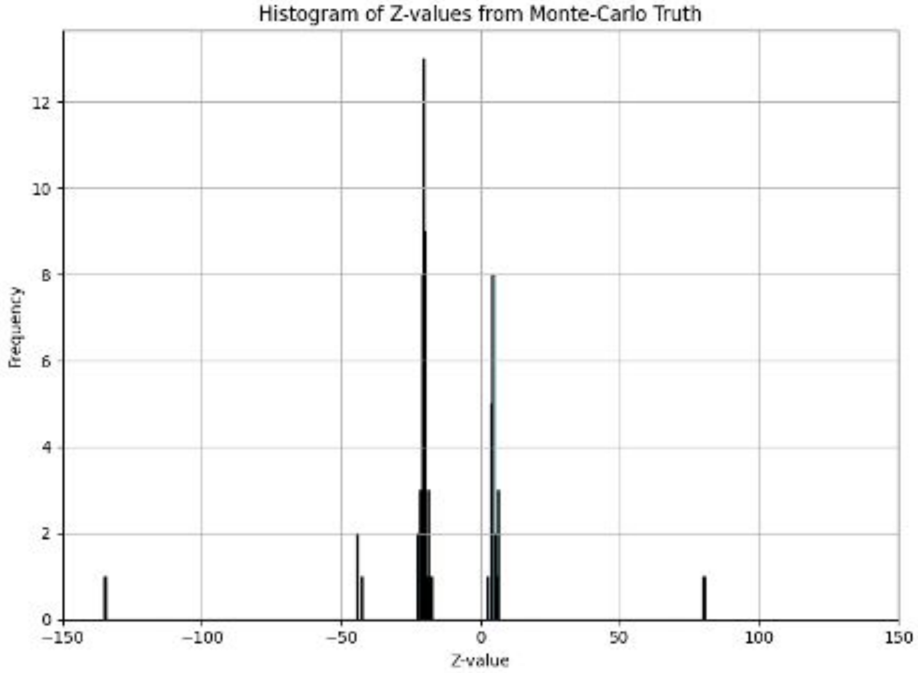
Updates since publication

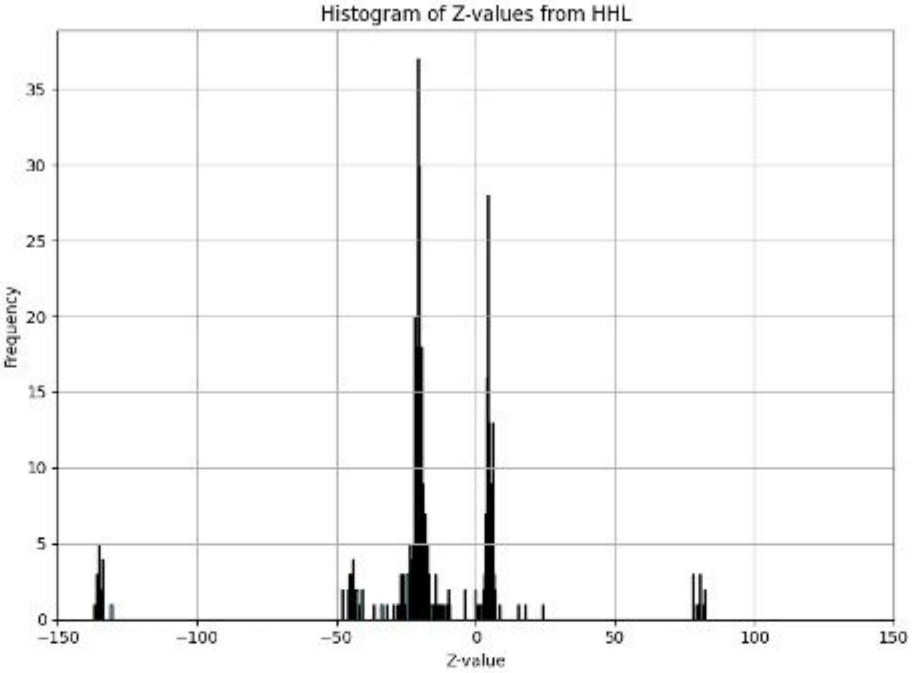
Talk with current results accepted at **CHEP**



Updates since publication

PV location using track segments from HHL:





Track reconstruction with QAOA

- Quantum Approximate Optimization Algorithm [arXiv:1411.4028, tutorial]

$$\mathcal{H} = -\frac{1}{2}\left[\left(\sum_{a,b,c}\frac{\cos^{\lambda}(\theta_{abc})}{r_{ab}+r_{bc}}s_{ab}s_{bc}\right) - \alpha\left(\sum_{b\neq c}s_{ab}s_{ac}+\sum_{a\neq c}s_{ab}s_{cb}\right) - \beta\left(\sum_{a,b}s_{ab}-N\right)^{2}\right]$$

(1) (2) (3)

(1) main term: favours aligned, short segments
(2) 1st penalty term: forbids segments that share head/tail from belonging to the same track
(3) 2nd penalty term: keeps the number of active segments equal to #hits

QAOA for Track Reconstruction

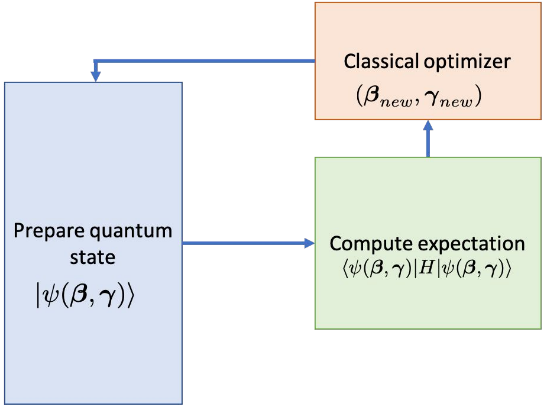
A **variational** algorithm ideal to solve combinatorial optimization problems, e.g. Max-Cut problem

- *‘Finding an optimal object out of a finite set of objects’*

$$|\psi(\beta,\gamma)\rangle = U(\beta)U(\gamma)...U(\beta)U(\gamma)\,|\psi_0\rangle$$

$$U(\beta) = e^{-i\beta H_B},\ U(\gamma) = e^{-i\gamma H_P}$$

- H_B: mixing Hamiltonian, H_P: **problem** Hamiltonian
- **Goal:** find optimal parameters **$(\square_{opt},\gamma_{opt})$** such that the quantum state encodes the solution to the problem

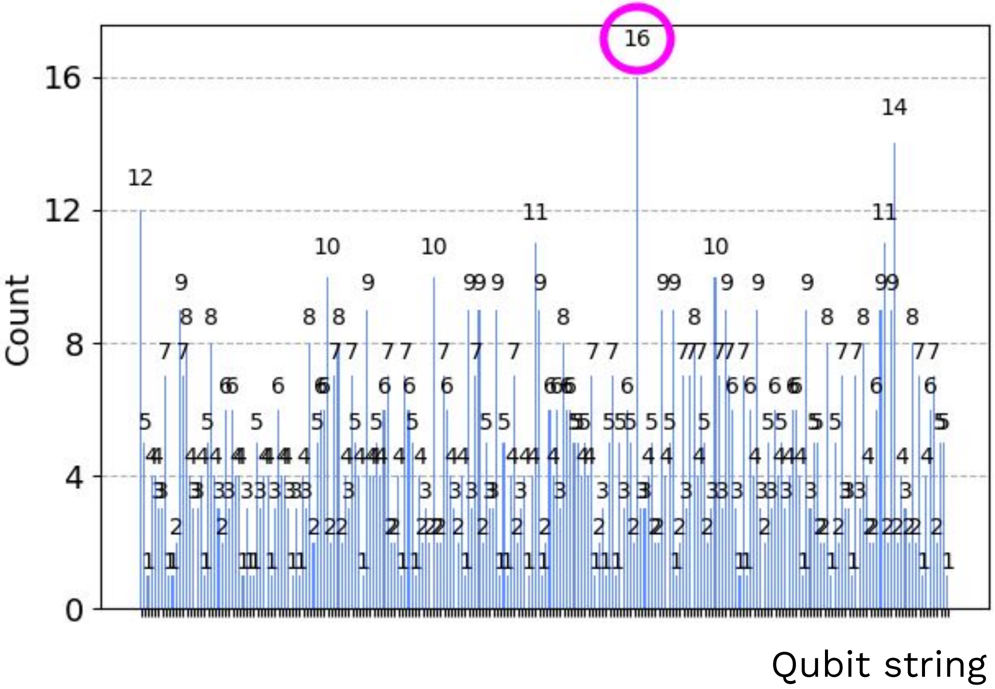


Classical optimizer
$(\boldsymbol{\beta}_{new}, \boldsymbol{\gamma}_{new})$

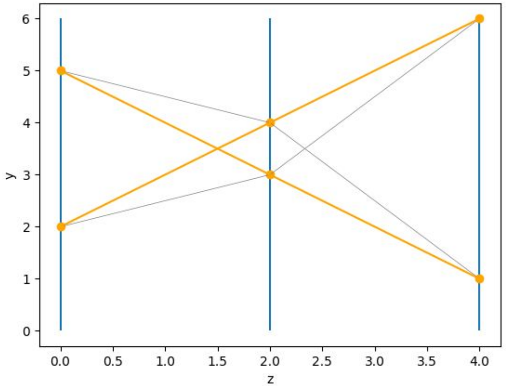
Prepare quantum state
$|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$

Compute expectation
$\langle\psi(\boldsymbol{\beta},\boldsymbol{\gamma})|H|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$

Initial results



- Study with simulated straight tracks: 2 tracks, 3 detector layers
- Working on the generalized case

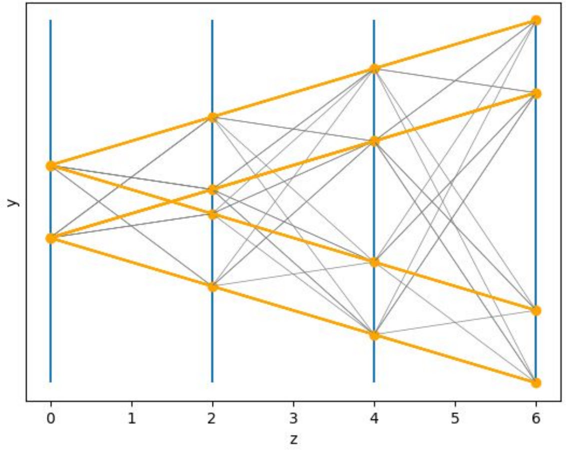


Results and ongoing work

Memory issues to simulate 27 qubits:

- Fix 6, run 21 in batches
- sub-QUBO approach

# tracks	# layers	#qubits (segments)	Number of Z and ZZ gates
2	3	8	512
3	3	18	2664
3	4	27	5940

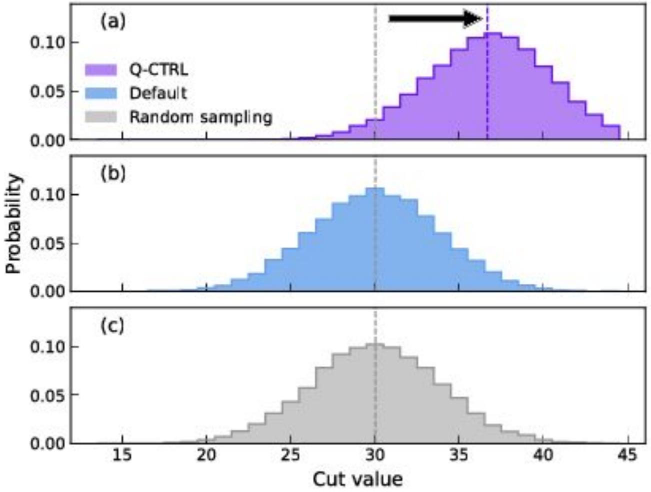


\+ Translation to triplets
\+ Transpilation studies

Fresh from the oven

Figures of Merit à la Q-CTRL [arXiv:2406.01743v1]

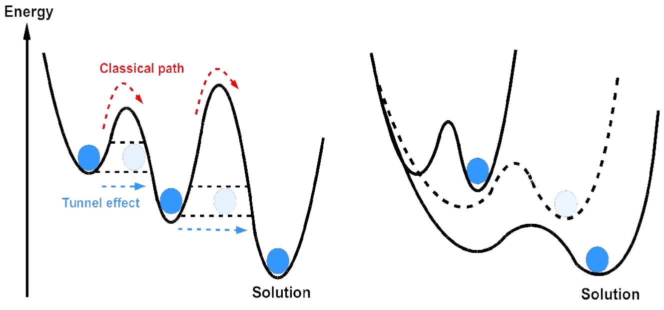
- ○ approximation ratio
- ○ raw likelihood that the top solution found by the solver is returned
- Modified QAOA



Quantum Annealers

The Quantum Computing Company™

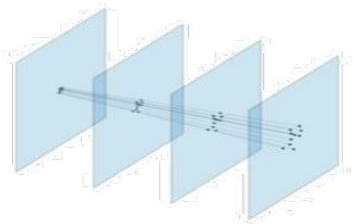
- Different hardware, not gate-based
- Optimal for minimizing Ising-like Hamiltonians



Quantum Tunnelling **Adiabatic evolution**

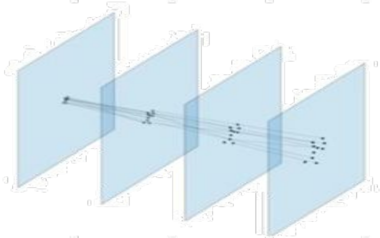
SIMULATED ANNEALING

- Low energy state: -40
- Time: 1.5 hours



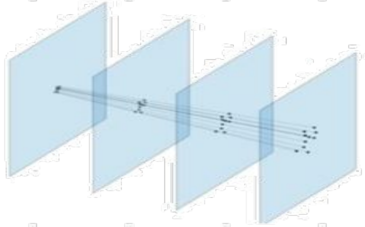
QUANTUM ANNEALING

- Low energy state: 2
- Time: few minutes



LEAP HYBRID SOLVER

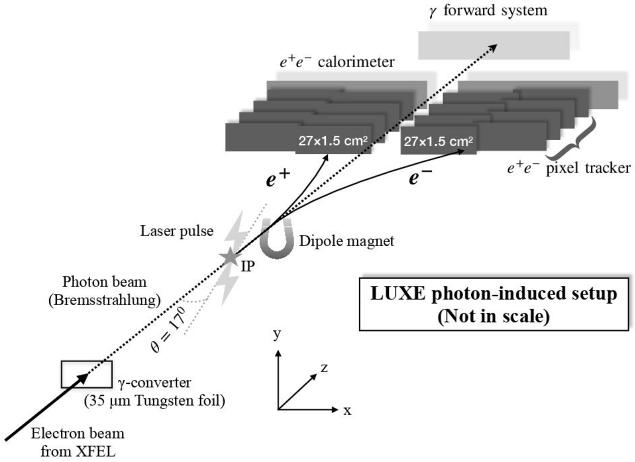
- Low energy state: -40
- QPU access time: 38.993 milliseconds,
- Run time: 3000.198 milliseconds.



Related work [arXiv:2210.13021]

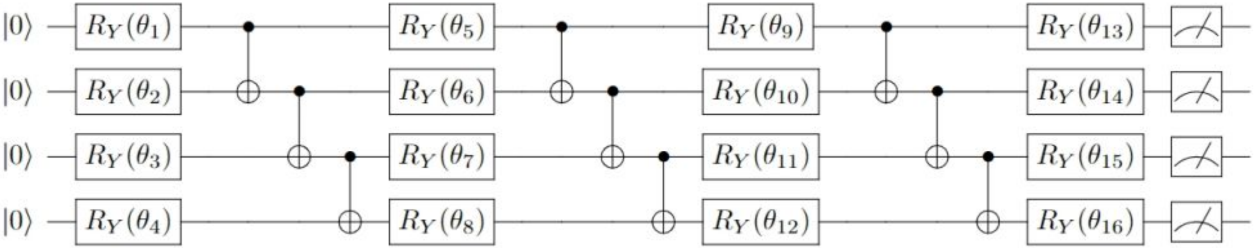
- **LUXE** experiment @ **DESY** to study QED in the strong-field regime
- Tracking of positrons traversing 4 layers of tracking detectors
- Classical methods:
    - Combinatorial Kalman Filter using triplets of hits
    - GNN where each hit is a node

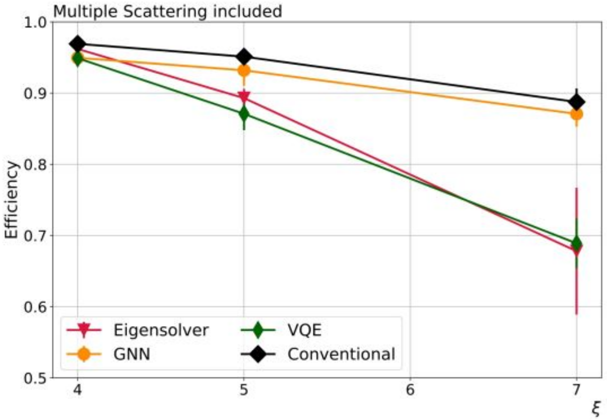
$$O = \sum_{i}^{N} \sum_{j<i} b_{ij} T_i T_j + \sum_{i=1}^{N} a_i T_i \quad T_i, T_j \in \{0,1\}$$

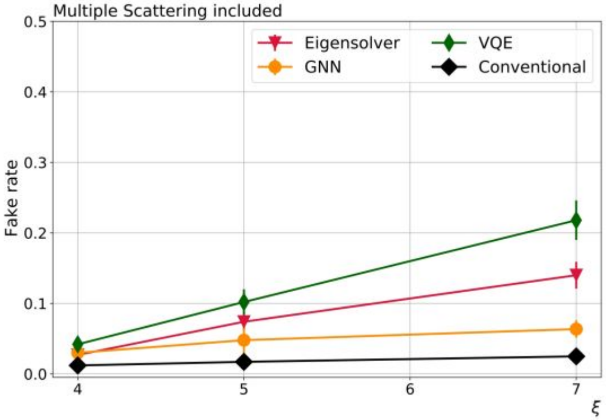


Related work [arXiv:2210.13021]

Variational Quantum Eigensolver: hybrid quantum-classical algorithm







Outlook

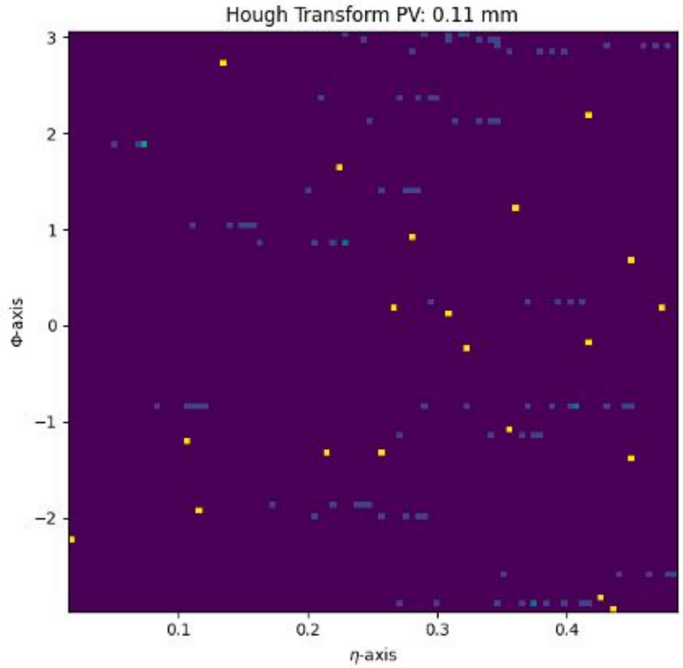
Several well-known caveats affect virtually all the approaches:

- Scalability of input
- Circuit depth

Efficient **output retrieval**:

- Ongoing studies using **Hough transform**

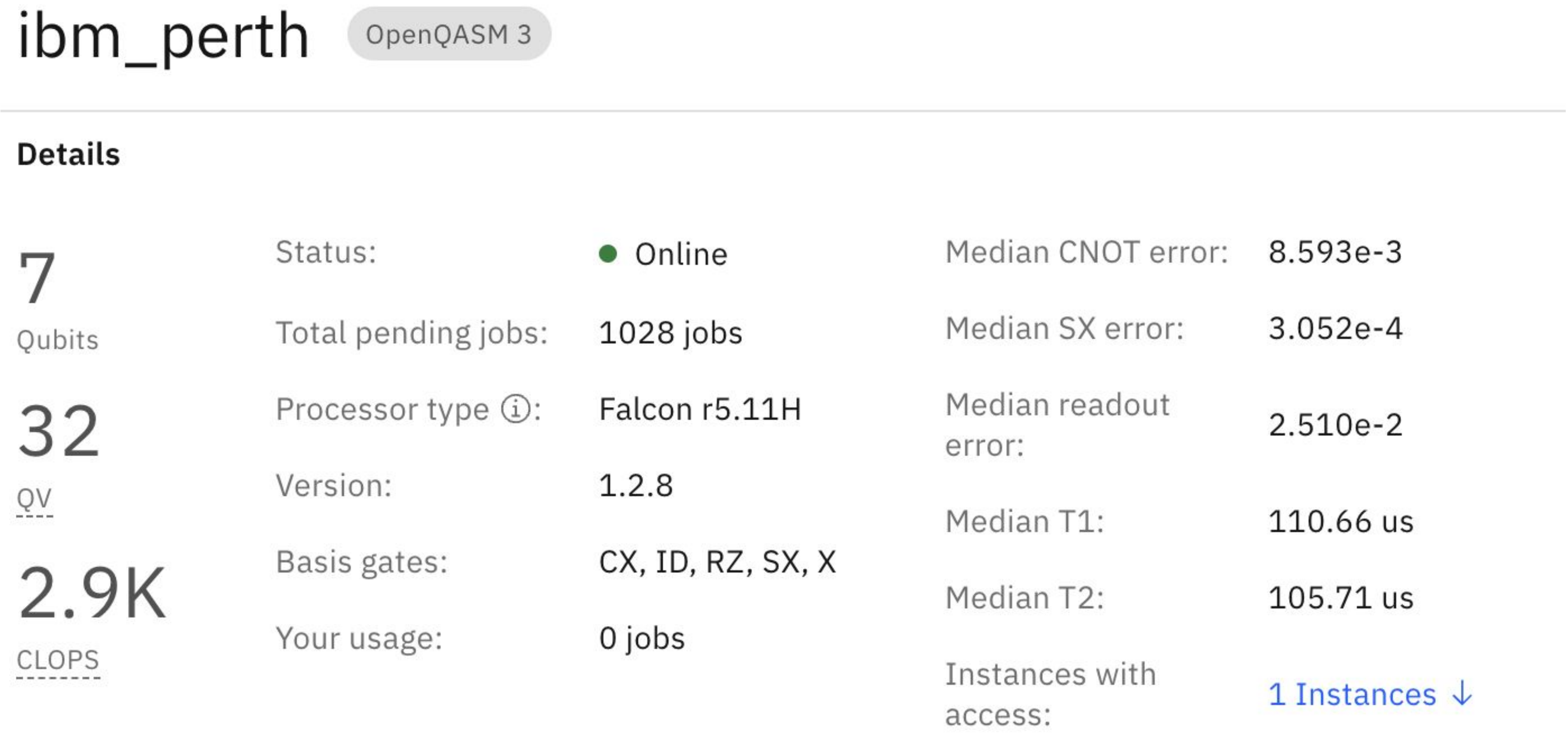
VQLS and **ansatz search** under investigation



Moitas grazas!

Porting to hardware

IBM **Quantum**



ibm_perth

OpenQASM 3

Details

7
Qubits

32
QV

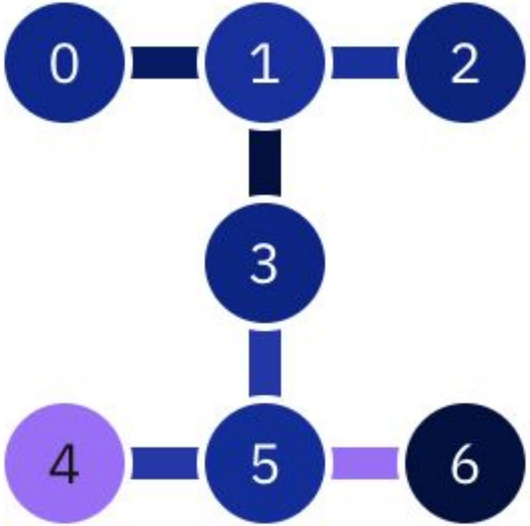
2.9K
CLOPS

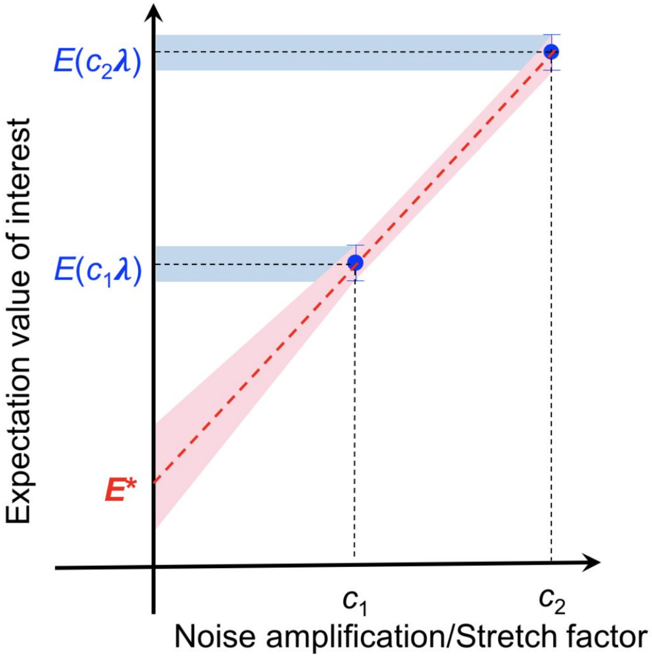
Status: ● Online
Total pending jobs: 1028 jobs
Processor type ⓘ: Falcon r5.11H
Version: 1.2.8
Basis gates: CX, ID, RZ, SX, X
Your usage: 0 jobs

Median CNOT error: 8.593e-3
Median SX error: 3.052e-4
Median readout error: 2.510e-2
Median T1: 110.66 us
Median T2: 105.71 us
Instances with access: 1 Instances ↓

Porting to hardware

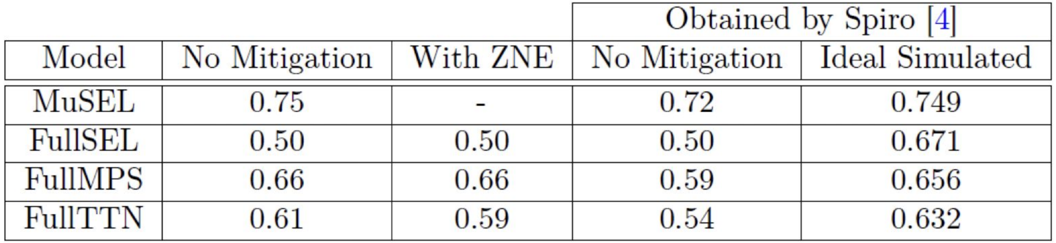
IBM **Quantum**



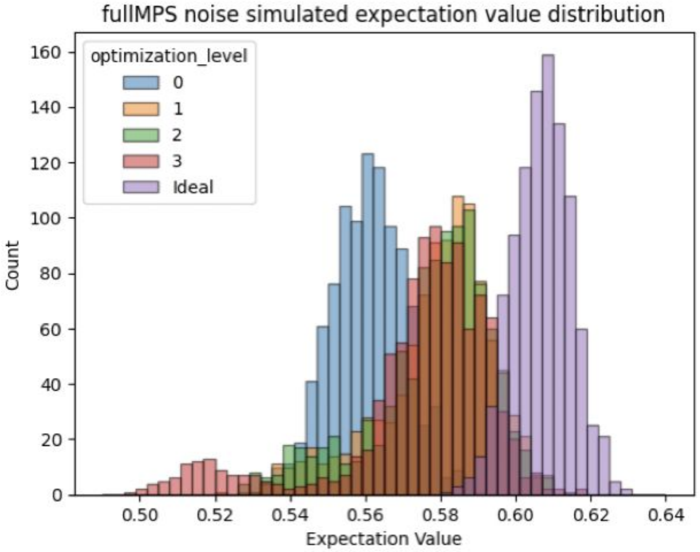


Noise Error Mitigation

- Study of transpilation optimisation levels (4000 transpilations)
- Tried Zero Noise Extrapolation, Probabilistic Error Correction:
- Need to further investigate



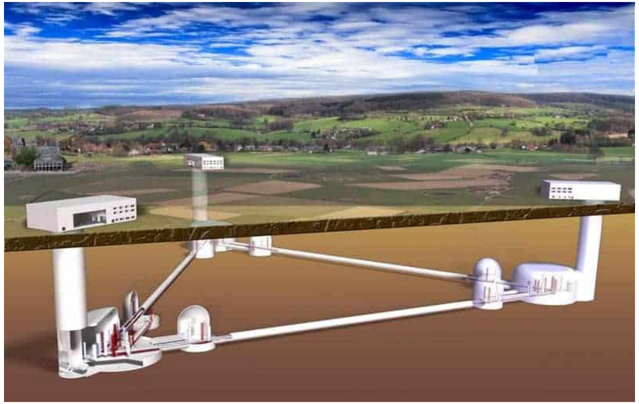
			Obtained by Spiro [4]	
Model	No Mitigation	With ZNE	No Mitigation	Ideal Simulated
MuSEL	0.75	-	0.72	0.749
FullSEL	0.50	0.50	0.50	0.671
FullMPS	0.66	0.66	0.59	0.656
FullTTN	0.61	0.59	0.54	0.632



QC & Gravitational Waves

Next generation of GW detectors: increased **bandwidth** and **sensitivity.**
→ new techniques are needed on top classical template matching

Grover search: for template matching.
Theoretical studies ongoing on the feasibility of this for GW detection.



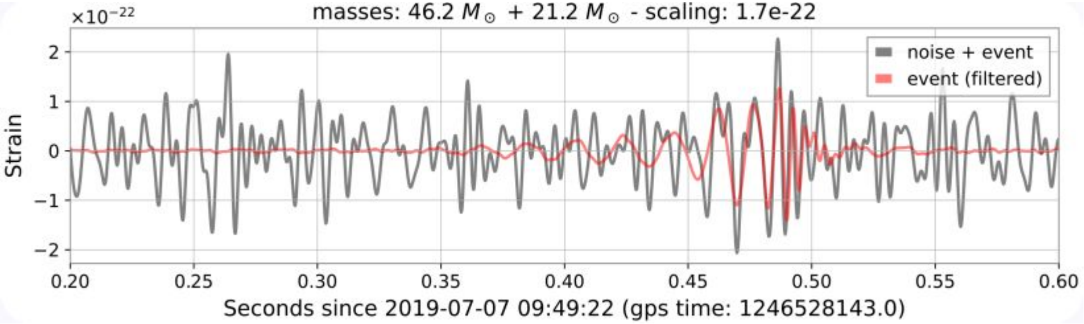
Solving Einstein Field Equations:

- The GW signals need to be calculated by solving the set of non-linear equations of the EFE.
- A proof of principle using the algorithms proposed by [2011.10395] to solve a simplified model has been implemented.

QC & Gravitational Waves

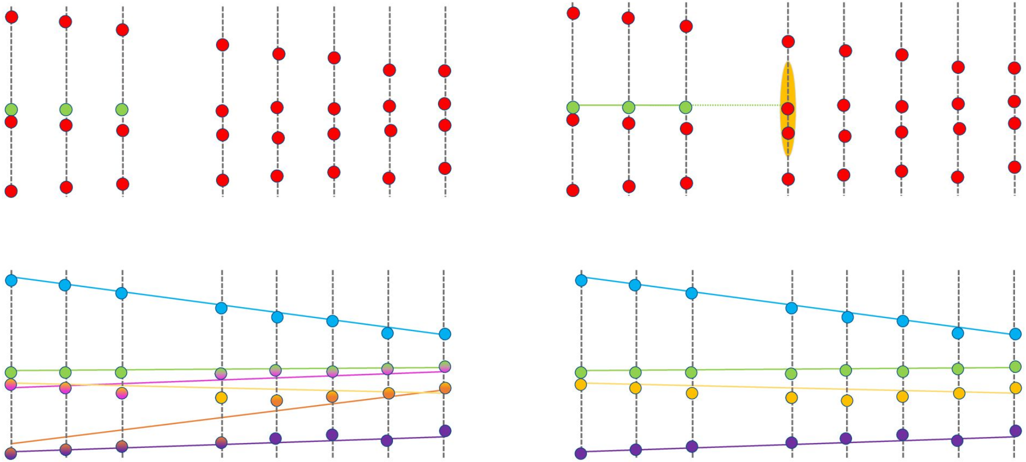
Quantum-enhanced Feature Spaces:

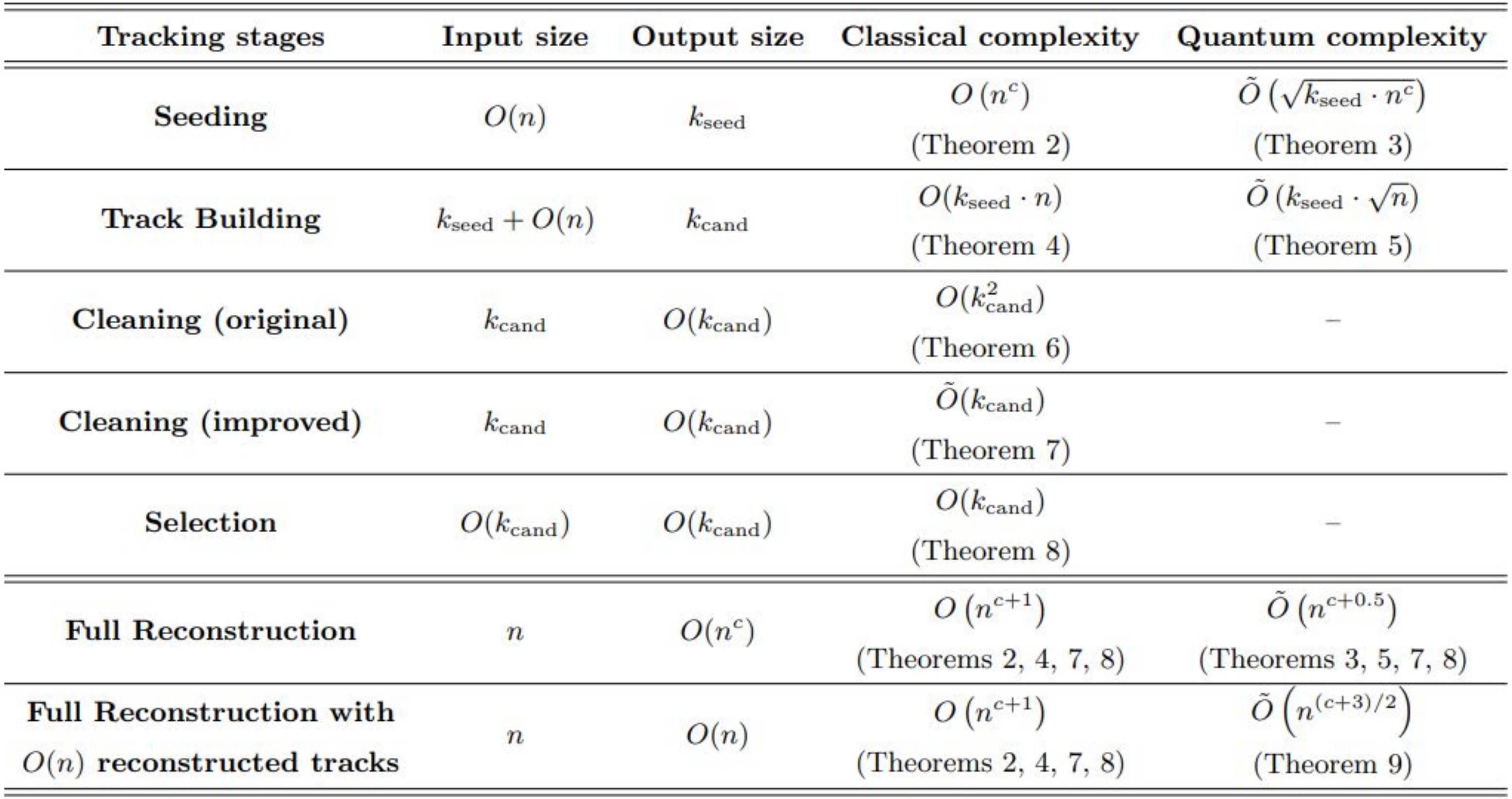
- Data is too noisy and large to be used directly by a QML algorithm.
- The number of events is too small for proper training.
- Real noise samples and a simulated event signal are used asc a *signal database* → a set of time-series features is extracted to create the training dataset.
- **Detection:** kernel method. **Characterisation:** support vector machine.



Local tracking methods [arXiv:2104.11583]

1. Seeding
2. Track building
3. Cleaning
4. Selection





Tracking stages	**Input size**	**Output size**	**Classical complexity**	**Quantum complexity**
Seeding	$O(n)$	k_{seed}	$O(n^c)$ (Theorem 2)	$\tilde{O}\left(\sqrt{k_{\text{seed}} \cdot n^c}\right)$ (Theorem 3)
Track Building	$k_{\text{seed}} + O(n)$	k_{cand}	$O(k_{\text{seed}} \cdot n)$ (Theorem 4)	$\tilde{O}\left(k_{\text{seed}} \cdot \sqrt{n}\right)$ (Theorem 5)
Cleaning (original)	k_{cand}	$O(k_{\text{cand}})$	$O(k_{\text{cand}}^2)$ (Theorem 6)	–
Cleaning (improved)	k_{cand}	$O(k_{\text{cand}})$	$\tilde{O}(k_{\text{cand}})$ (Theorem 7)	–
Selection	$O(k_{\text{cand}})$	$O(k_{\text{cand}})$	$O(k_{\text{cand}})$ (Theorem 8)	–
Full Reconstruction	n	$O(n^c)$	$O\left(n^{c+1}\right)$ (Theorems 2, 4, 7, 8)	$\tilde{O}\left(n^{c+0.5}\right)$ (Theorems 3, 5, 7, 8)
Full Reconstruction with $O(n)$ reconstructed tracks	n	$O(n)$	$O\left(n^{c+1}\right)$ (Theorems 2, 4, 7, 8)	$\tilde{O}\left(n^{(c+3)/2}\right)$ (Theorem 9)

n: number of particles, c: number of hits, k_{seed}: total number of generated seeds, k_{cand}: number of track candidates

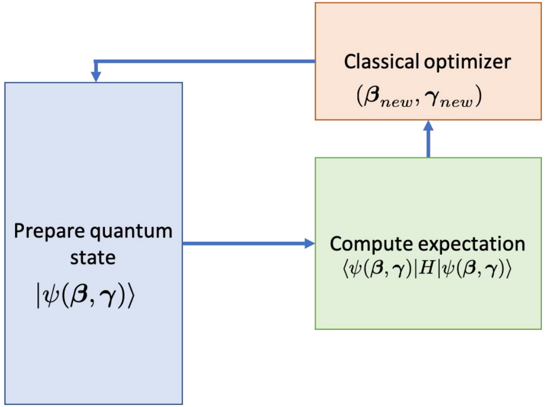
QAOA for Track Reconstruction

- Quantum Approximate Optimization Algorithm [arXiv:1411.4028, tutorial]
- A **variational** algorithm ideal to solve combinatorial optimization problems, e.g. Max-Cut problem
    - *'Finding an optimal object out of a finite set of objects'*

$$|\psi(\beta,\gamma)\rangle = U(\beta)U(\gamma)...U(\beta)U(\gamma)\,|\psi_0\rangle$$

$$U(\beta) = e^{-i\beta H_B},\; U(\gamma) = e^{-i\gamma H_P}$$

- H_B: mixing Hamiltonian, H_P: **problem** Hamiltonian
- **Goal:** find optimal parameters $(\square_{opt},\gamma_{opt})$ such that the quantum state encodes the solution to the problem


Classical optimizer
$(\boldsymbol{\beta}_{new}, \boldsymbol{\gamma}_{new})$

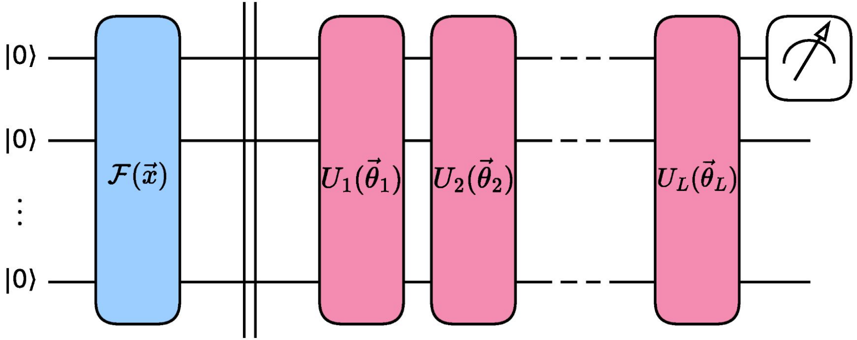
Prepare quantum state
$|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$

Compute expectation
$\langle\psi(\boldsymbol{\beta},\boldsymbol{\gamma})|H|\psi(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle$

Entropy studies

Study of the Entropy production within a Variational Quantum Circuit during its training phase:

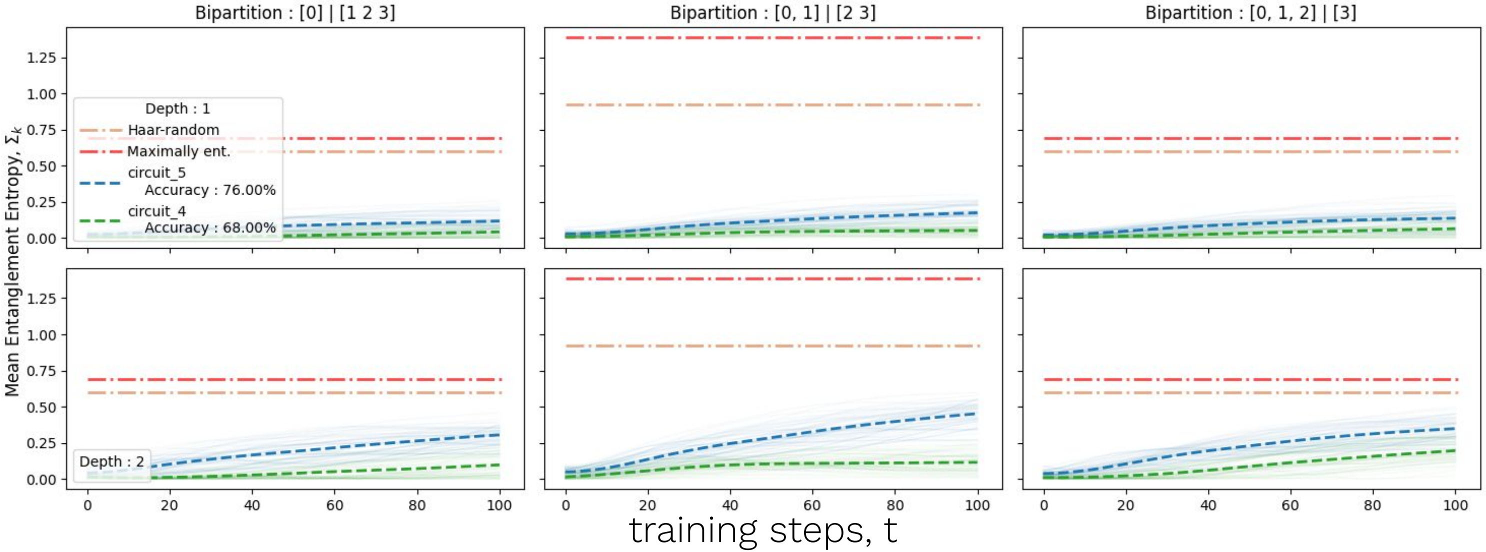
Goal: Use the information of the entropy values to enhance the training performance for the task of jet-tagging (b vs c)



Values of Entropy were inspected

- For each training step “t”
    - At each “depth” of the circuit:
        - depth 0 : $\mathcal{F}(\vec{x})\left|0^{\otimes N}\right\rangle$
        - depth 1 : $U_1(\vec{\theta}_1)\mathcal{F}(\vec{x})\left|0^{\otimes N}\right\rangle$
        - depth L : $U_L(\vec{\theta}_L)\ldots U_1(\vec{\theta}_1)\mathcal{F}(\vec{x})\left|0^{\otimes N}\right\rangle$ (output state)

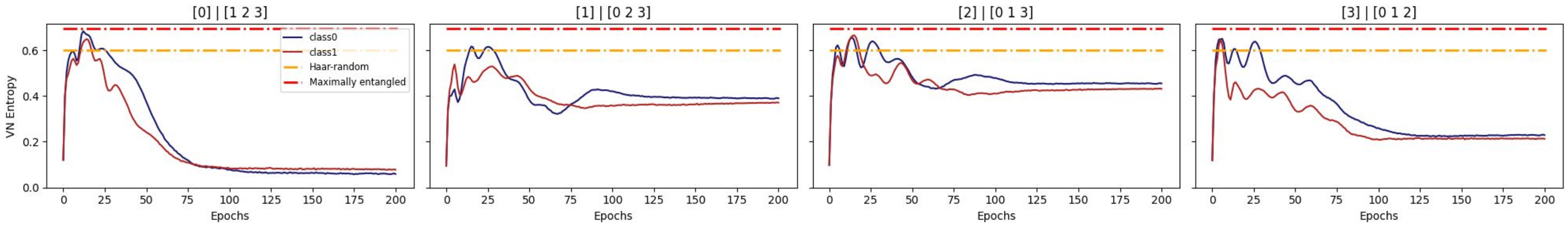
Study of the Entropy production within a Variational Quantum Circuit



- different circuits
- different datasets (b vs c jet-tagging and IRIS)
- different parameters initializations (Gaussian vs Uniform)
- different loss functions

Study of the Entropy production within a Variational Quantum Circuit

Feature importance from Entropy values



More results coming soon!

Optimization for hardware

IBM **Quantum**

- Ported from quantum simulations to *real* quantum computers ✅
- Tested and optimised several architectures ✅
    - Different advantages in terms of robustness against ***noise*** from hardware imperfections
- Currently trying *noise error mitigation* techniques 🚧

quantum simulator
quantum hardware