Quantum Computing & High Energy Physics

4th COMCHA School

Zaragoza, 8 – 15 April 2026

CAPA COMCHA Computing challenges

Miriam Lucio Martínez

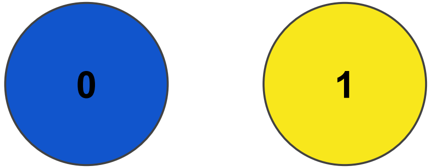
CSIC
Consejo Superior de Investigaciones Científicas
Vniversitat de València
IFIC Institut de Física Corpuscular
Excelencia Severo Ochoa

Before we start

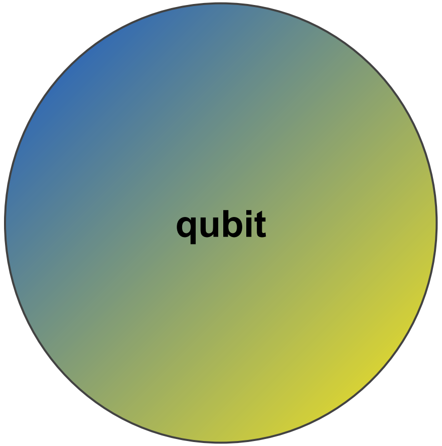
Questions are **very** welcome!!

What is Quantum Computing?

Classic computer: bits



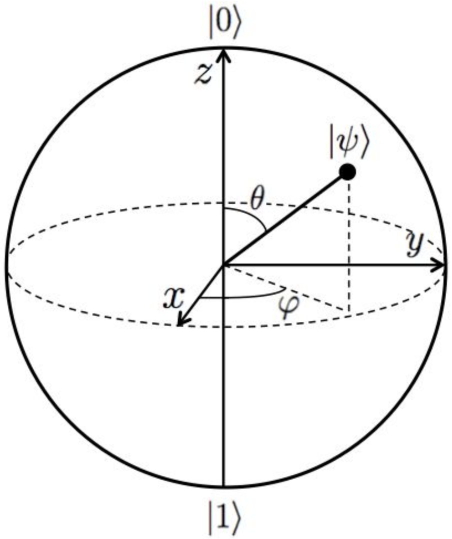
Quantum Computer: qubits



Quantum Computing in a nutshell

- **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

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



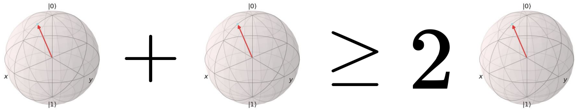
Bloch sphere

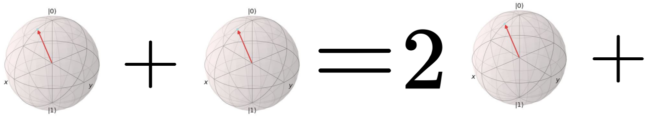
Quantum Computing Cheat Sheet - Qubits

Multi-qubit states are not just a collection of Bloch spheres...

In fact what we are missing are quantum correlations between the qubits: **entanglement**

The number of coefficients needed to describe a multi-qubit state scales exponentially with the number of qubits





Quantum Correlations

$$2^{N_{qubit}}$$

HUGE!
Can it be used for computing?

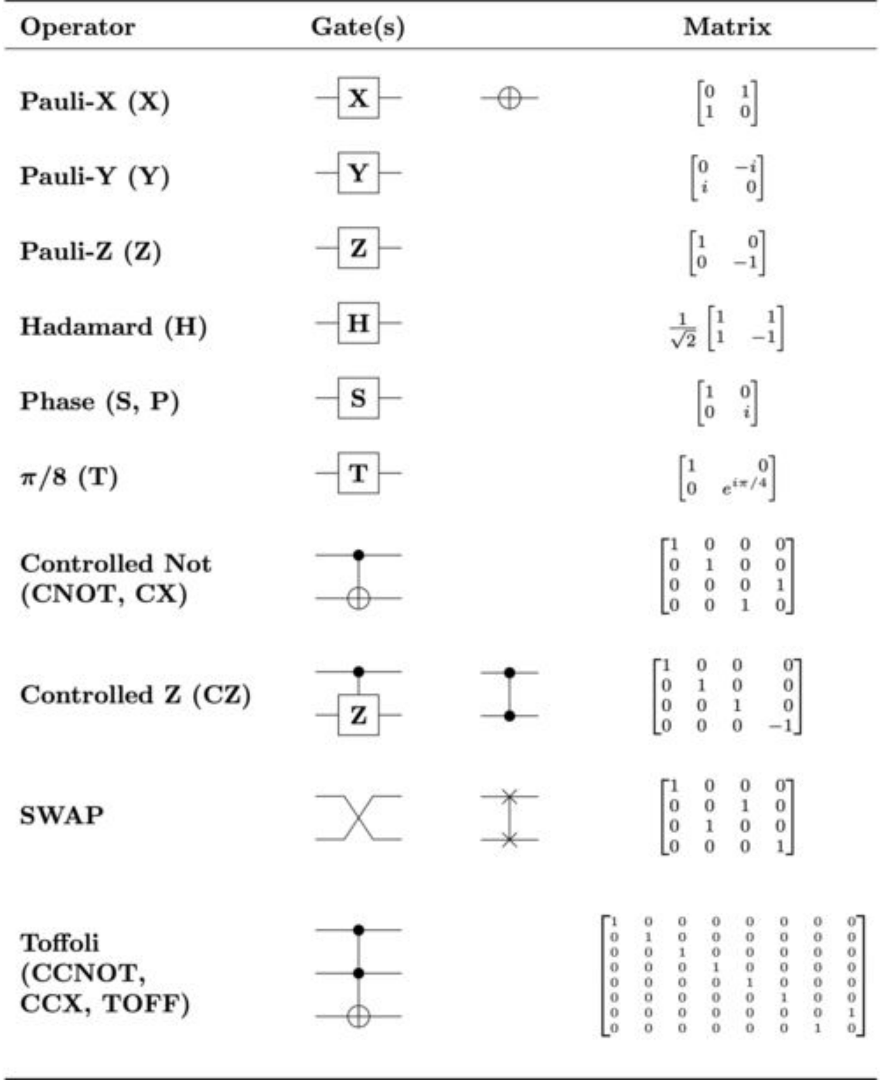
Quantum gates

- Unitary transformations → *reversible*
- Can be **single** or **multiple**

Universal set

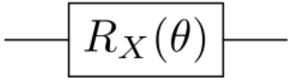
- Finite set of gates that can be used to approximate any unitary transformation on any number of qubits to within any desired level of precision (*)
- QC's *alphabet*, not unique
- E.g.: {H, S, CNOT, T}

(*) Solovay-Kitaev Theorem

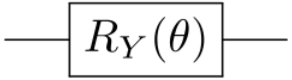


Operator	Gate(s)	Matrix
Pauli-X (X)	X ⊕	$\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$
Pauli-Y (Y)	Y	$\begin{bmatrix}0 & -i\\i & 0\end{bmatrix}$
Pauli-Z (Z)	Z	$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$
Hadamard (H)	H	$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}$
Phase (S, P)	S	$\begin{bmatrix}1 & 0\\0 & i\end{bmatrix}$
$\pi/8$ (T)	T	$\begin{bmatrix}1 & 0\\0 & e^{i\pi/4}\end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$
Controlled Z (CZ)	Z	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}$
SWAP		$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix}1&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0\\0&0&1&0&0&0&0&0\\0&0&0&1&0&0&0&0\\0&0&0&0&1&0&0&0\\0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&1\\0&0&0&0&0&0&1&0\end{bmatrix}$

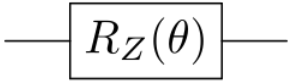
Rotation gates



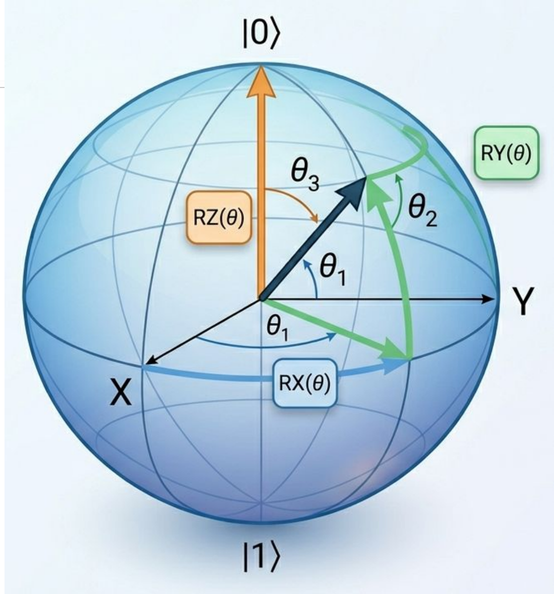
$$R_x(\theta) = e^{-i\frac{\theta}{2}X} = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -i\sin\left(\frac{\theta}{2}\right) \\ -i\sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$



$$R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}$$

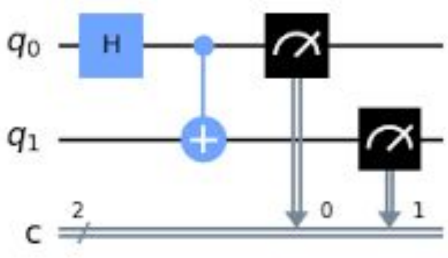


$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix}$$



Quantum circuits

Qubits + quantum gates + (some) measurements = **Quantum circuit**

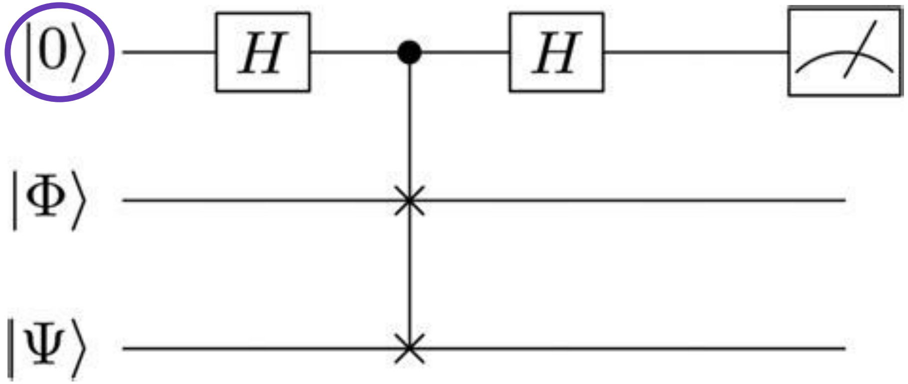


Quantum algorithms are built with quantum circuits

SWAP Test

- **Goal**: measure the similarity between quantum states in a *nondestructive* way [quant-ph/0102001]
- Very important for e.g. pattern recognition, clustering

Ancilla qubit



$$P(0) = \frac{1}{2} + \frac{1}{2}|\langle\phi|\psi\rangle|^2$$

$$\text{If } |\phi\rangle = |\psi\rangle, \text{ then } |\langle\phi|\psi\rangle|^2 = 1 \implies P(0) = 1$$

$$\text{If } |\phi\rangle \perp |\psi\rangle, \text{ then } |\langle\phi|\psi\rangle|^2 = 0 \implies P(0) = 0.5$$

Quantum algorithms

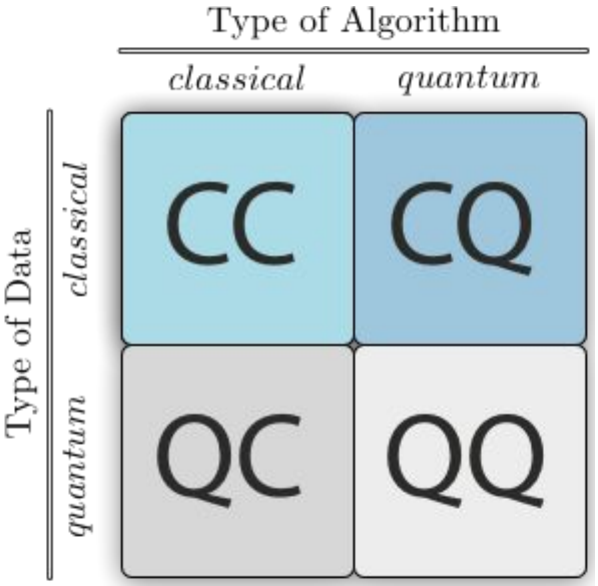
The idea is to leverage superposition and interference from QC for:

- Improved computational complexity
- Better handling of (quantum) data
- Faster algorithms

However:

- A lot of approaches are **hybrid**: classical[HPC]-quantum
- Example: variational algorithms

https://quantumalgorithmzoo.org/



Quantum Computing Algorithms

Designing quantum algorithms is hard:

- # of known algorithms **~50**
- # of actually useful algorithms **~10**
- # of useful algorithms that we can run today **0**

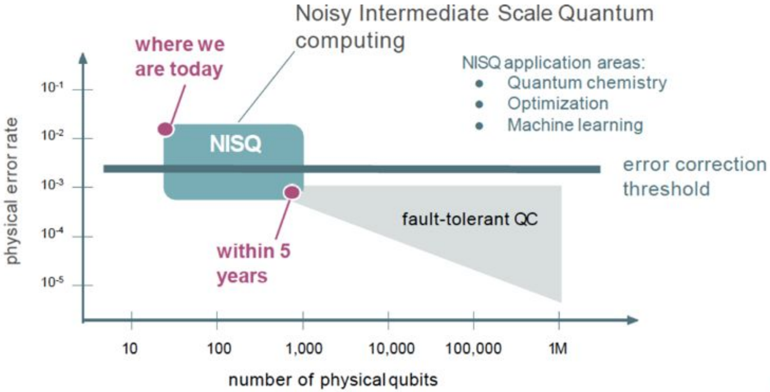
Shor's algorithm (factorization)

Grover's algorithm (search)

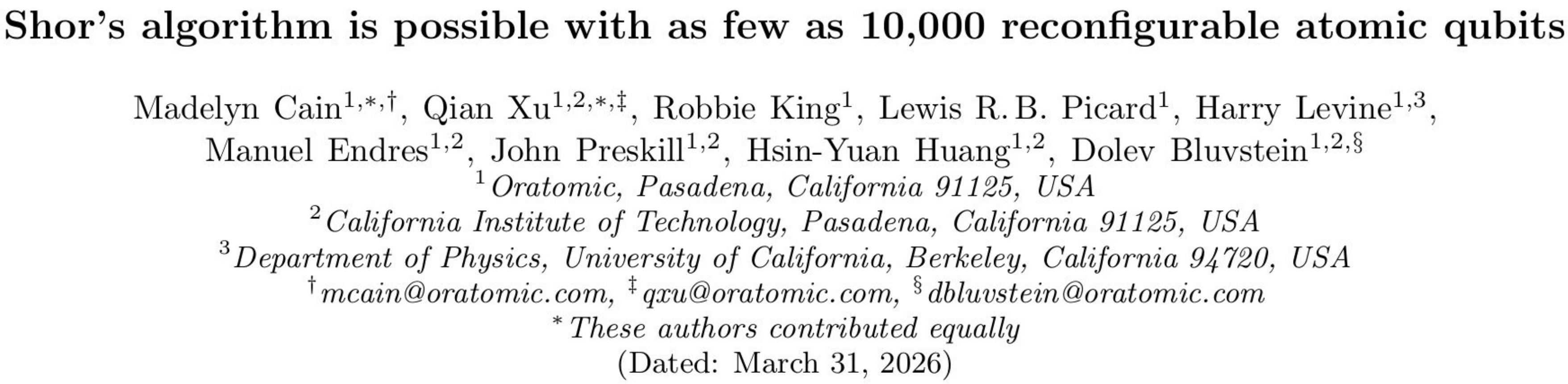
HHL algorithm (linear algebra)

Why?

- Classical computing knowledge does not help
- Current quantum hardware is **noisy**
- Large circuits are extremely hard to simulate



A note about this



Shor's algorithm is possible with as few as 10,000 reconfigurable atomic qubits

Madelyn Cain[1,*,†], Qian Xu[1,2,*,‡], Robbie King[1], Lewis R. B. Picard[1], Harry Levine[1,3],
Manuel Endres[1,2], John Preskill[1,2], Hsin-Yuan Huang[1,2], Dolev Bluvstein[1,2,§]
[1] *Oratomic, Pasadena, California 91125, USA*
[2] *California Institute of Technology, Pasadena, California 91125, USA*
[3] *Department of Physics, University of California, Berkeley, California 94720, USA*
[†] *mcain@oratomic.com*, [‡] *qxu@oratomic.com*, [§] *dbluvstein@oratomic.com*
[*] *These authors contributed equally*
(Dated: March 31, 2026)

Before: order of **million** qubits!!

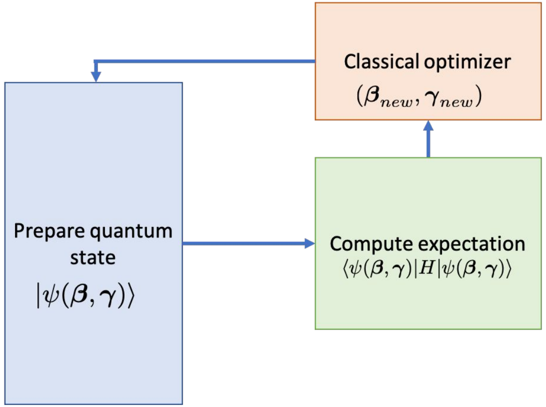
An example: QAOA

- 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

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

Encoding of classical data

Basis encoding

- Classical n-bit string (e.g., 0011) → computational basis state of an n-qubit system (e.g., |ψ⟩ = |0011⟩).

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |\boldsymbol{x}^{(m)}\rangle$$

$$D = \{2, 4, 6, 8\} \rightarrow |D\rangle = \frac{1}{2}\left(|0010\rangle + |0100\rangle + |0110\rangle + |1000\rangle\right)$$

Encoding of classical data

Amplitude encoding

- Classical data → amplitudes of quantum state
- Classical value → amplitude of the corresponding computational basis state.

$$|D\rangle = \frac{1}{\sqrt{\sum_{m=1}^{M} |x^{(m)}|^2}} \sum_{m=1}^{M} x^{(m)} |x^{(m)}\rangle$$

$$D = \{2, 4, 6, 8\} \rightarrow |D\rangle = \frac{1}{\sqrt{120}} (2|00\rangle + 4|01\rangle + 6|10\rangle + 8|11\rangle))$$

Encoding of classical data

Angle encoding

- Classical data → angles of a quantum rotations
- Each qubit codifies one angle

$$|D\rangle = \bigotimes_{m=1}^{M} R_x(\theta_m)|0\rangle = \bigotimes_{m=1}^{M} \left[\cos\left(\frac{\theta_m}{2}\right)|0\rangle - i\sin\left(\frac{\theta_m}{2}\right)|1\rangle\right]$$

$$D = \{2, 4, 6, 8\} :$$

$$|D\rangle = (\cos(1)|0\rangle - i\sin(1)|1\rangle) \otimes (\cos(2)|0\rangle - i\sin(2)|1\rangle) \otimes (\cos(3)|0\rangle - i\sin(3)|1\rangle) \otimes (\cos(4)|0\rangle - i\sin(4)|1\rangle)$$

Encoding of classical data

Angle encoding

WARNING!

The choice of encoding method impacts efficiency, expressiveness

✅ encoding → less qubits

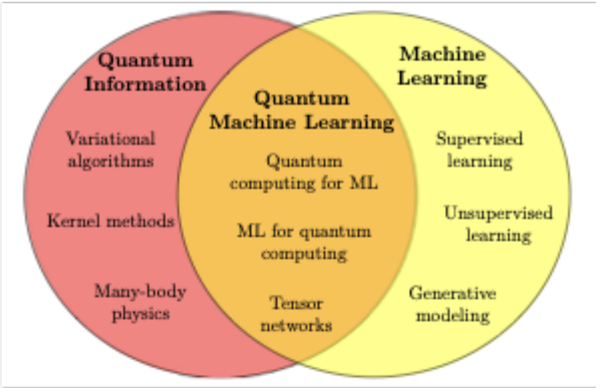
❌ encoding → decoding → bigger circuit → higher risk of noise

$$|D\rangle = (\cos(1)|0\rangle - i\sin(1)|1\rangle) \otimes (\cos(2)|0\rangle - i\sin(2)|1\rangle) \otimes (\cos(3)|0\rangle - i\sin(3)|1\rangle) \otimes (\cos(4)|0\rangle - i\sin(4)|1\rangle)$$

What about QML?

Plethora of algorithms:

- Hybrid CQ/QC
    - Variational algorithms, QGans, etc
    - Classical part is usually the **optimizing** part
- Quantum-inspired classical ML
    - Tensor networks, Simulated annealing
- Fully quantum algorithms
    - Binary classifiers

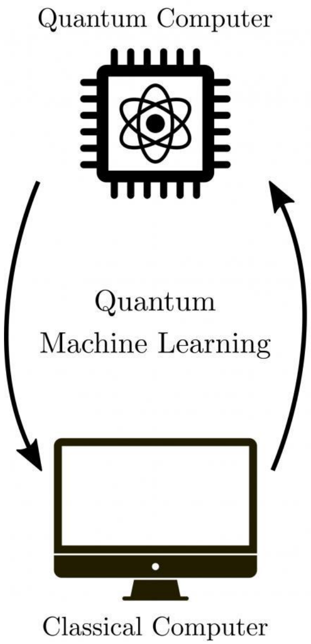


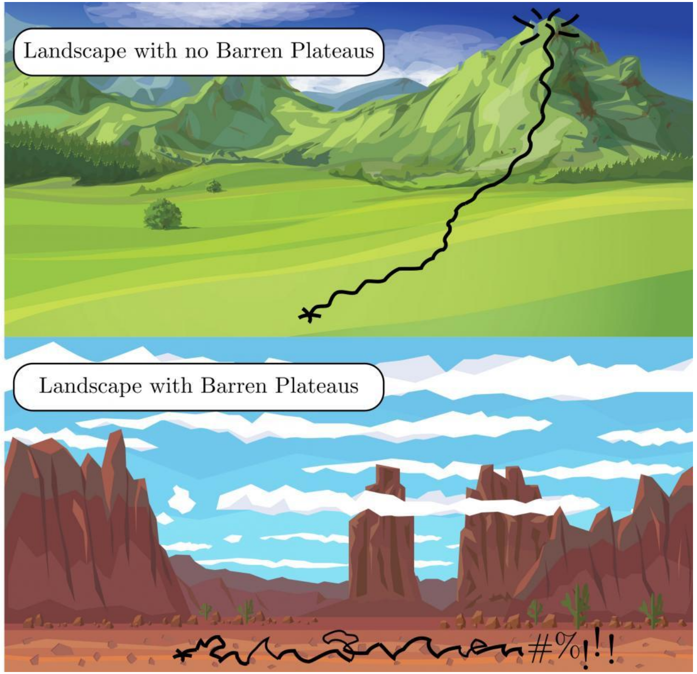
From G. Martínez de Lejarza Samper’s PhD tehsis

What about QML?

Common **challenges**:

- Local minima
- Barren plateaus
- Data encoding and signal post-processing





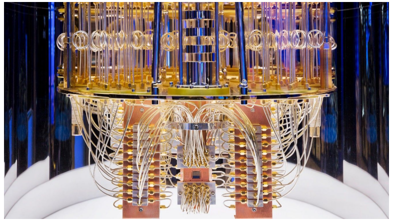
Quantum Computing - Hardware

Several technologies are being explored as physical qubits:

Superconducting

IQM IBM Google

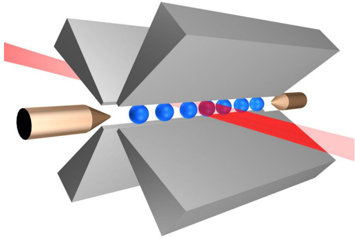
Superconducting electric circuits at 10mK behave as quantum systems with discrete energy levels



Trapped ions

IONQ QUANTINUUM

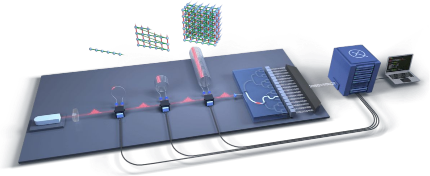
Charged atoms constrained in electromagnetic traps and manipulated with laser



Optical

XANADU

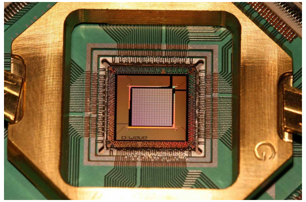
Linear optics devices using photons as information carriers



Annealing

D-Wave The Quantum Computing Company™ QILIMANJARO QUANTUM-TECH

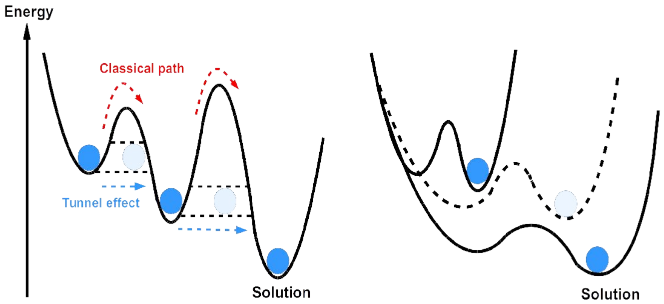
Ising-chain qubits interacting with a customizable Hamiltonian



Quantum Annealers

D:Wave
The Quantum Computing Company™

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

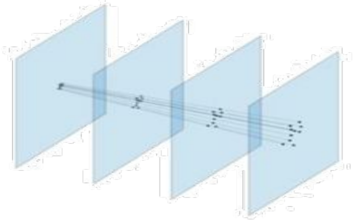


Quantum Tunnelling

Adiabatic evolution

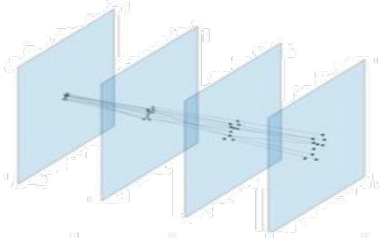
SIMULATED ANNEALING

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



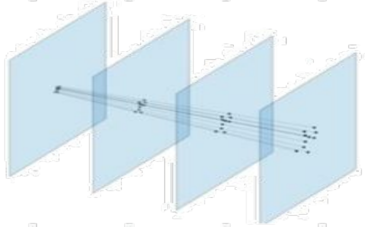
QUANTUM ANNEALING

- 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.



Development Roadmap

IBM **Quantum**

	2016–2019 ✓	2020 ✓	2021 ✓	2022 ✓	2023 ✓	2024	2025	2026	2027	2028	2029	2033+
	Ran quantum circuits on the IBM Quantum Platform	Released multi-dimensional roadmap publicly with initial aim focused on scaling	Enhanced quantum execution speed by 100x with Qiskit Runtime	Brought dynamic circuits to unlock more computations	Enhanced quantum execution speed by 5x with Quantum Serverless and execution modes	Improve quantum circuit quality and speed to allow 5K gates with parametric circuits	Enhance quantum execution speed and parallelization with partitioning and quantum modularity	Improve quantum circuit quality to allow 7.5K gates	Improve quantum circuit quality to allow 10K gates	Improve quantum circuit quality to allow 15K gates	Improve quantum circuit quality to allow 100M gates	Beyond 2033, quantum-centric supercomputers will include 1000's of logical qubits unlocking the full power of quantum computing
Data scientists						Platform						
						Code assistant ⟳	Functions	Mapping collections	Specific libraries			General purpose QC libraries
Researchers					Middleware							
					Quantum Serverless ✓	Transpiler service ⟳	Resource management	Circuit knitting x p	Intelligent orchestration			Circuit libraries
Quantum physicists			Qiskit Runtime									
	IBM Quantum Experience ✓		QASM 3 ✓	Dynamic circuits ✓	Execution modes ✓	Heron (5K) ⟳	Flamingo (5K)	Flamingo (7.5K)	Flamingo (10K)	Flamingo (15K)	Starling (100M)	Blue Jay (1B)
	Early ✓ Canary 5 qubits Albatross 16 qubits Penguin 20 qubits Prototype 53 qubits	Falcon ✓ Benchmarking 27 qubits		Eagle ✓ Benchmarking 127 qubits		Error mitigation 5k gates 133 qubits Classical modular Up to 133x3 = 399 qubits	Error mitigation 5k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 7.5k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 10k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error mitigation 15k gates 156 qubits Quantum modular Up to 156x7 = 1092 qubits	Error correction 100M gates 200 qubits Error corrected modularity	Error correction 1B gates 2000 qubits Error corrected modularity

✓ Executed by IBM

⟳ On target

Development roadmap

QUANTINUUM

	2020	2023	2025	2027	2029
SYSTEMS:	SYSTEM MODEL **H1**	SYSTEM MODEL **H2**	QUANTINUUM **HELIOS**	QUANTINUUM **SOL**	QUANTINUUM **APOLLO**
PHYSICAL QUBITS:	20	56	96	192	1000's
PHYSICAL 2-QUBIT GATE ERROR:	1×10^{-3}	1×10^{-3}	$< 5 \times 10^{-4}$	$< 2 \times 10^{-4}$	1×10^{-4}
LOGICAL QUBITS:		> 12	~ 50	~ 100	100's
LOGICAL ERROR RATES:		1×10^{-3}	$< 10^{-4}$	$\sim 10^{-5}$	1×10^{-5} to 1×10^{-10}*

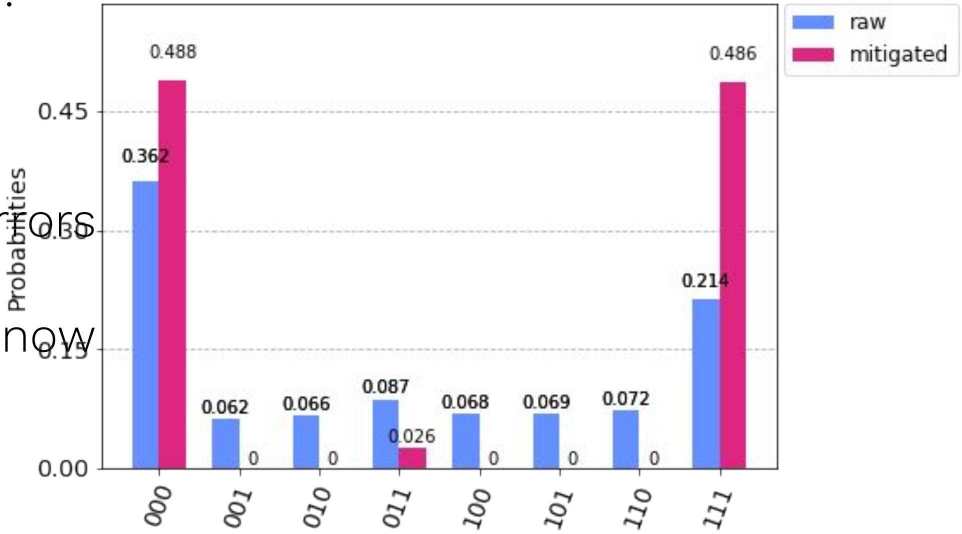
*analysis based on recent literature in new, novel error correcting codes predict that error could be as low as 1E-10 in Apollo (ref: arXiv:2403.16054, arXiv:2308.07915)

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

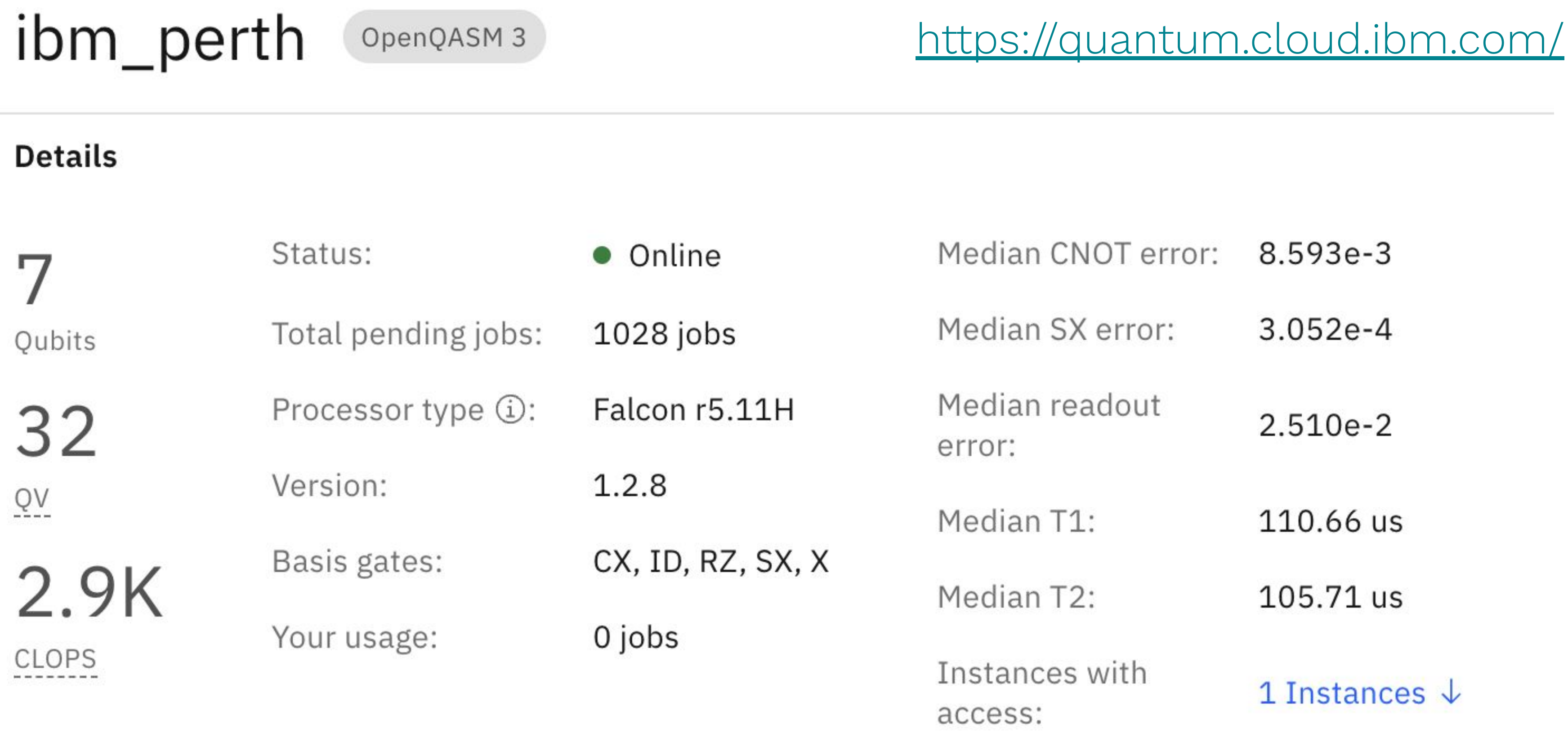
→ *Noise Error Mitigation* now
→ *Noise Error Correction* in the future



Porting to hardware

IBM **Quantum**

https://quantum.cloud.ibm.com/



ibm_perth

OpenQASM 3

Details

7
Qubits

32
QV

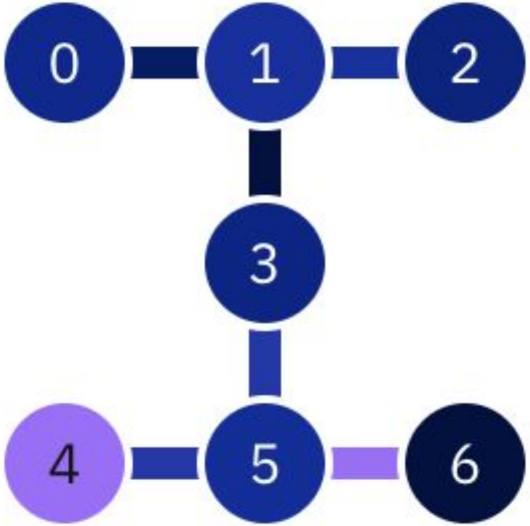
2.9K
CLOPS

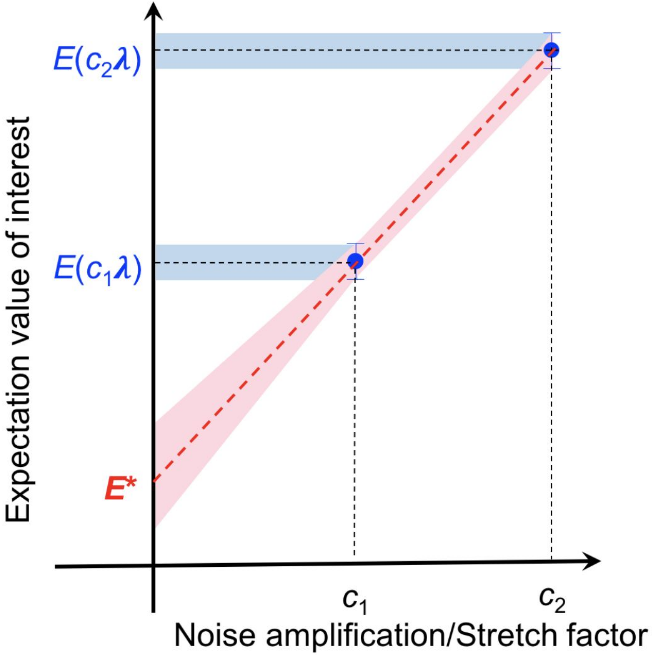
Status:	● Online
Total pending jobs:	1028 jobs
Processor type ⓘ:	Falcon r5.11H
Version:	1.2.8
Basis gates:	CX, ID, RZ, SX, X
Your usage:	0 jobs

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**





QC and High Energy Physics

Seems like the ⭐perfect match⭐ **but**:

- Data from experiments is usually classical – unless we work in quantum sensing and/or are optimistic about quantum RAMs

Approaches:

- Quantum-to-quantum matching

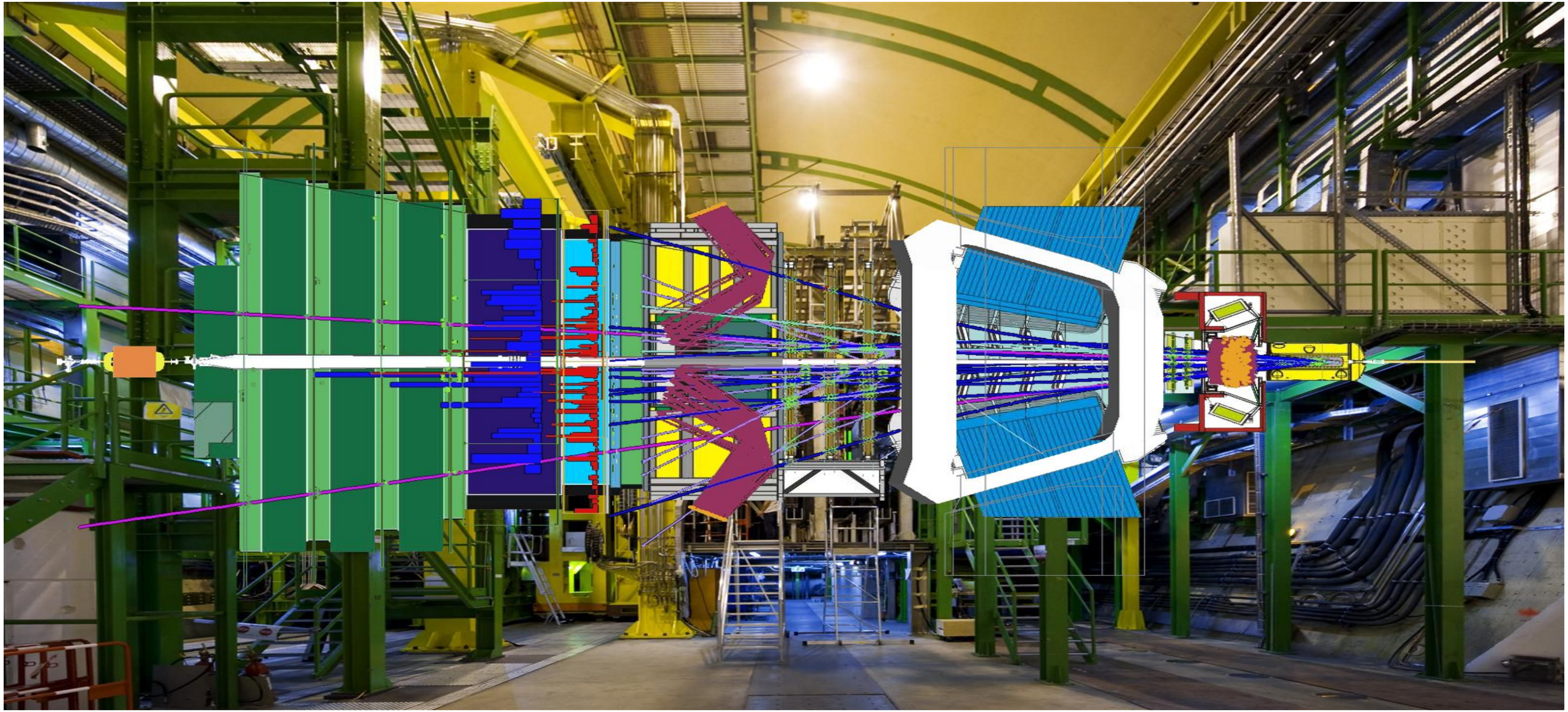
For simulation, theory, etc
Gold example: lattice QCD

- Quantum as a computational tool

Approach very computationally demanding tasks

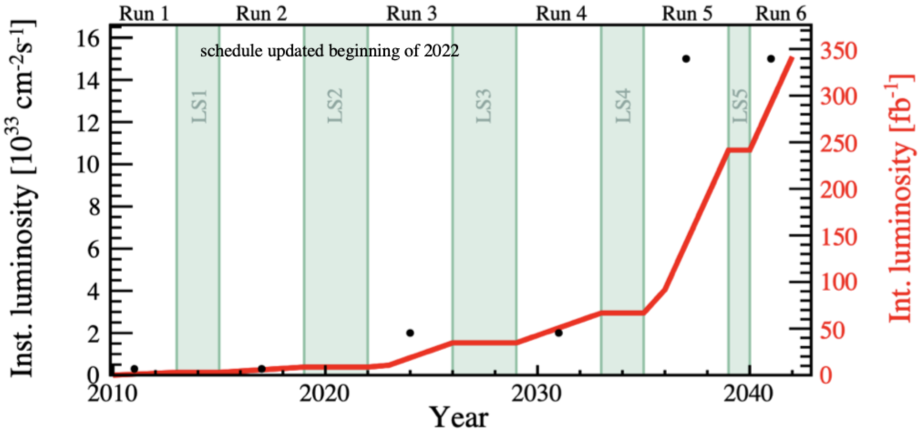
Example: LHCb event

Reconstruct events 40 Million times per second.

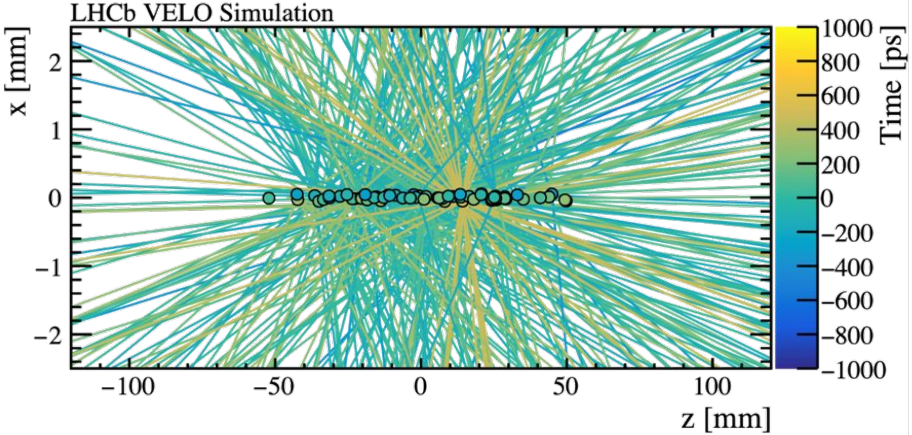


Motivation for QC

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

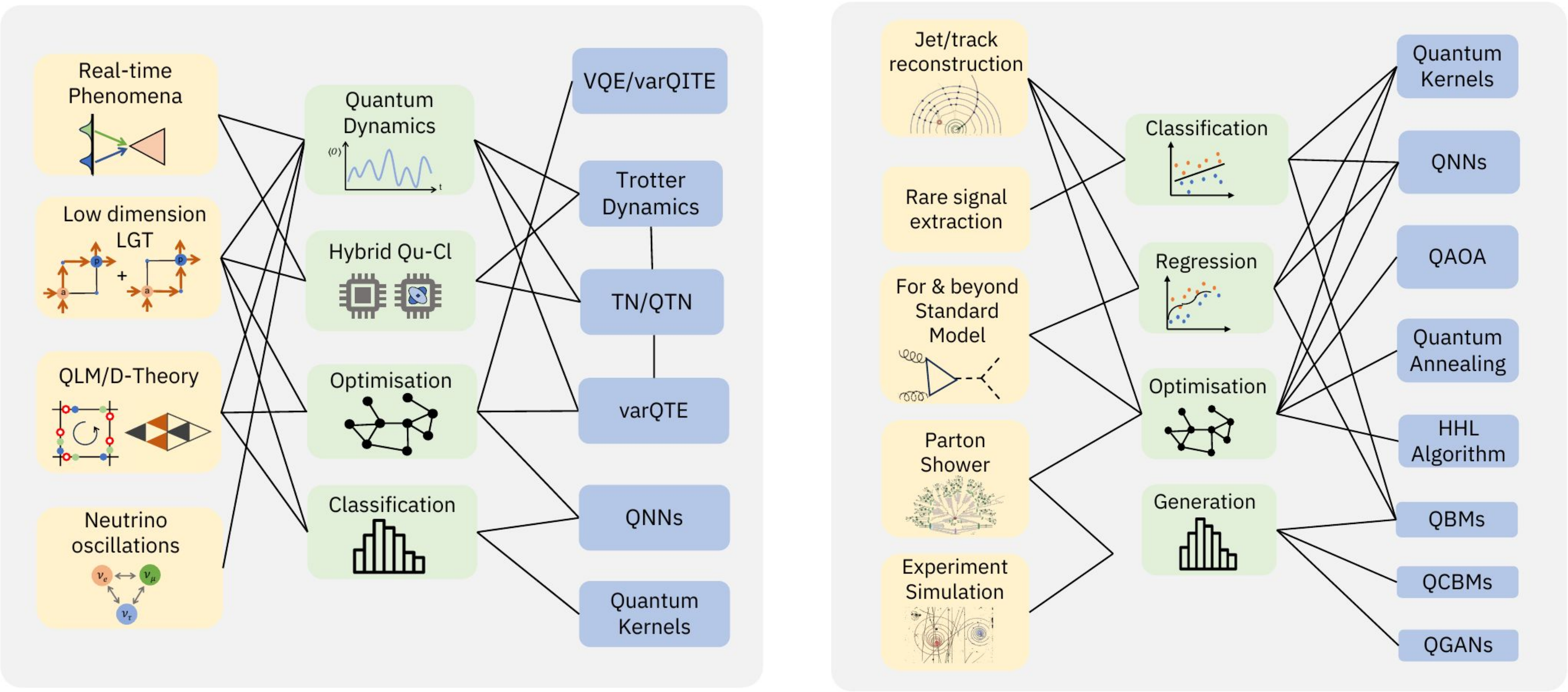


ECFA



Courtesy of Robbert Geertsema

HEP: Theory and experiment



Real-time Phenomena

Low dimension LGT

QLM/D-Theory

Neutrino oscillations

Quantum Dynamics

Hybrid Qu-Cl

Optimisation

Classification

VQE/varQITE

Trotter Dynamics

TN/QTN

varQTE

QNNs

Quantum Kernels

Jet/track reconstruction

Rare signal extraction

For & beyond Standard Model

Parton Shower

Experiment Simulation

Classification

Regression

Optimisation

Generation

Quantum Kernels

QNNs

QAOA

Quantum Annealing

HHL Algorithm

QBMs

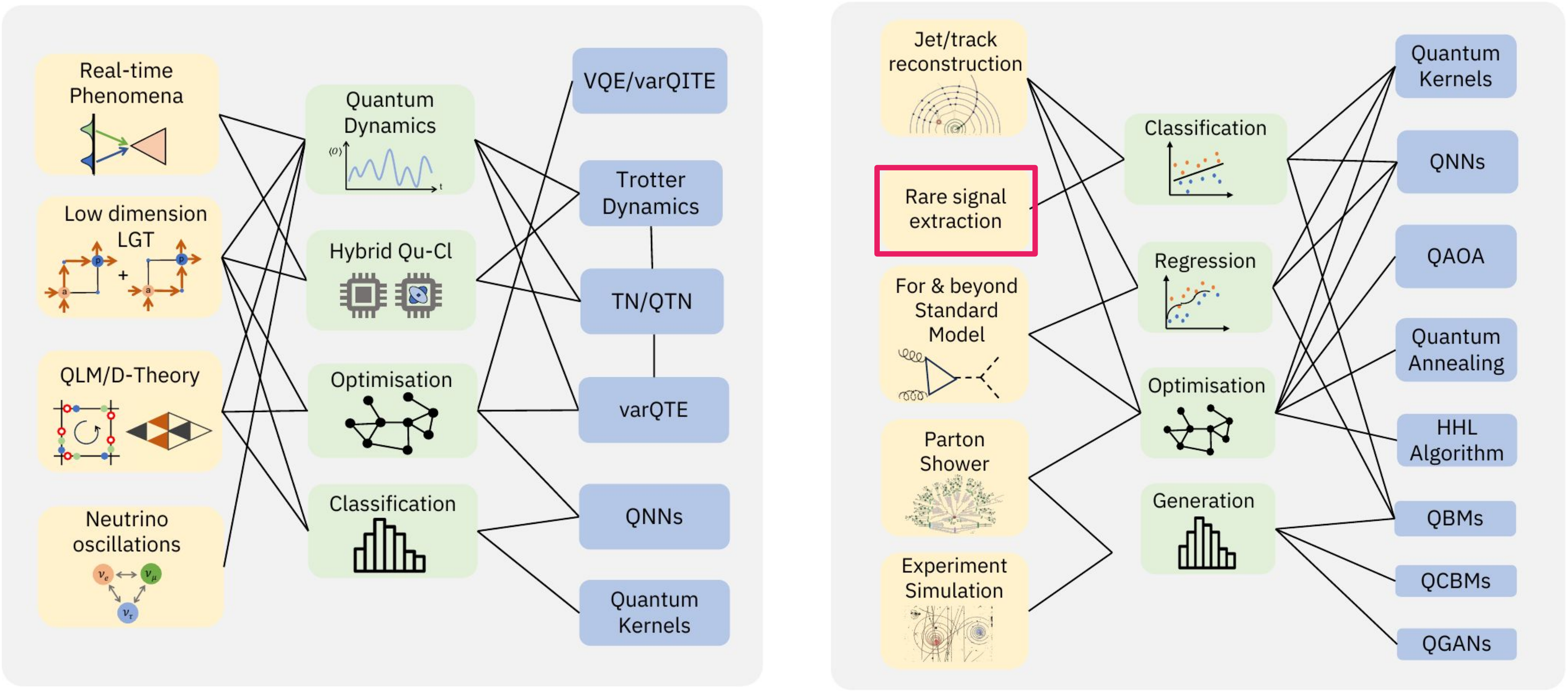
QCBMs

QGANs

Summary of the QC4HEP WG

Hands on (with M. Carou Laiño)

HEP: Theory and experiment



Real-time Phenomena

Low dimension LGT

QLM/D-Theory

Neutrino oscillations

Quantum Dynamics

Hybrid Qu-Cl

Optimisation

Classification

VQE/varQITE

Trotter Dynamics

TN/QTN

varQTE

QNNs

Quantum Kernels

Jet/track reconstruction

Rare signal extraction

For & beyond Standard Model

Parton Shower

Experiment Simulation

Classification

Regression

Optimisation

Generation

Quantum Kernels

QNNs

QAOA

Quantum Annealing

HHL Algorithm

QBMs

QCBMs

QGANs

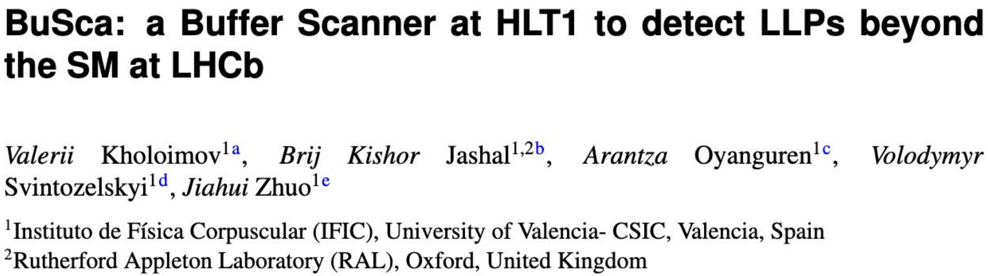
[Summary of the QC4HEP WG]

Anomaly detection

Train QC algorithm on SM (background) data

'Weird' (BSM) event → **anomaly**!!

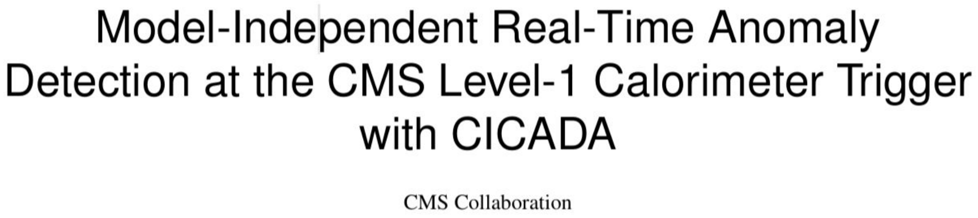
- May find things we are not looking for
- Quantum approach:
    - Entanglement
    - Algorithmic speedup
    - Needs less data for comparable performance



BuSca: a Buffer Scanner at HLT1 to detect LLPs beyond the SM at LHCb

Valerii Kholoimov[1a], *Brij Kishor* Jashal[1,2b], *Arantza* Oyanguren[1c], *Volodymyr* Svintozelskyi[1d], *Jiahui* Zhuo[1e]

[1]Instituto de Física Corpuscular (IFIC), University of Valencia- CSIC, Valencia, Spain
[2]Rutherford Appleton Laboratory (RAL), Oxford, United Kingdom



Model-Independent Real-Time Anomaly Detection at the CMS Level-1 Calorimeter Trigger with CICADA

CMS Collaboration

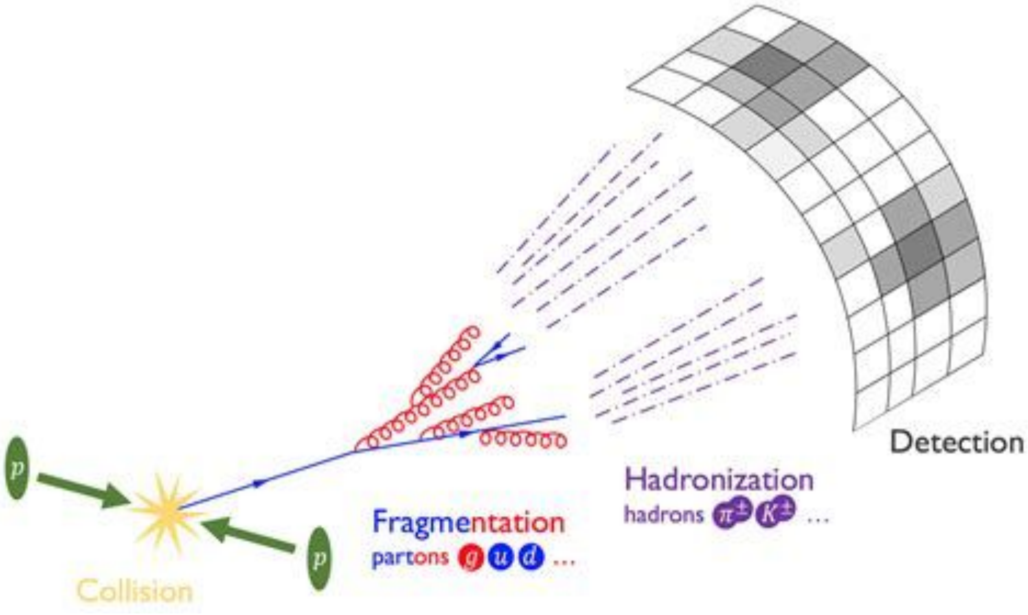
1 Particle 1 Qubit

Unsupervised anomaly detection with Quantum AutoEncoders (QAE) [2502.17301]

- Each **particle** is assigned to an individual **qubit**
- **Signal**: jets from processes e.g. H → bbar
- **Background:** QCD jets + CMS data

$$
\begin{aligned}
f \cdot \frac{p_{\mathrm{T}}}{p_{\mathrm{T,jet}}} \cdot (\eta - \eta_{\mathrm{jet}}) &\rightarrow \theta & (1)\\
f \cdot \frac{p_{\mathrm{T}}}{p_{\mathrm{T,jet}}} \cdot (\phi - \phi_{\mathrm{jet}}) &\rightarrow \varphi & (2)\\
(p_{\mathrm{T}}, \eta, \phi) &\rightarrow |\psi\rangle = R_X(\varphi) R_Y(\theta) |0\rangle & \\
&= \alpha(\theta, \varphi) |0\rangle + \beta(\theta, \varphi) |0\rangle & (3)\\
f &\rightarrow 1 + \frac{2\pi}{1 + e^{-w}} & (4)
\end{aligned}
$$

Intermezzo: jets



Variables:

- Transverse momentum

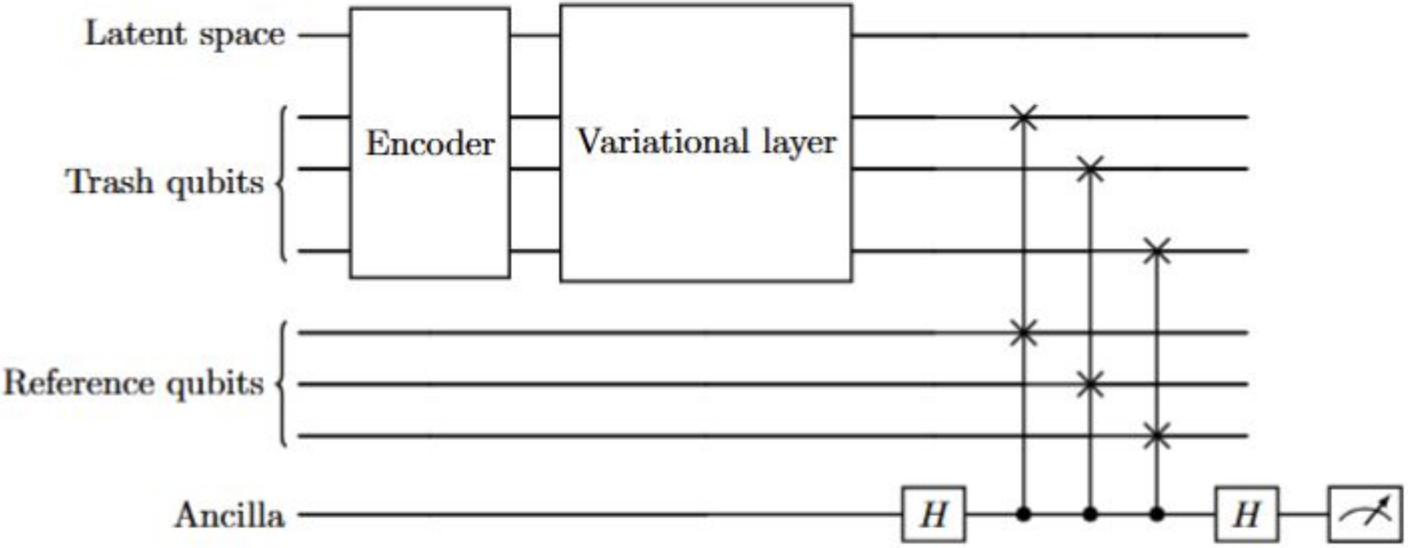
$$p_{\mathrm{T}} = \sqrt{p_x^2 + p_y^2}$$

- Pseudorapidity

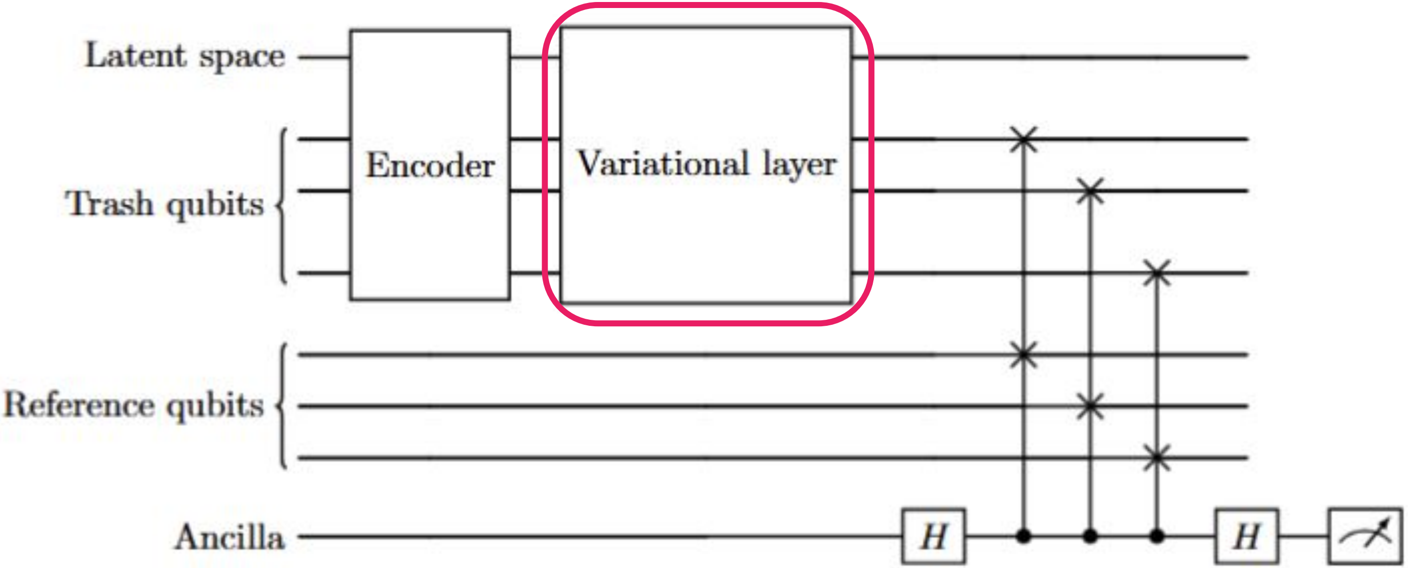
$$\eta = -\ln \tan(\theta/2)$$

- Azimuthal angle ϕ

The quantum circuit



Variational layer

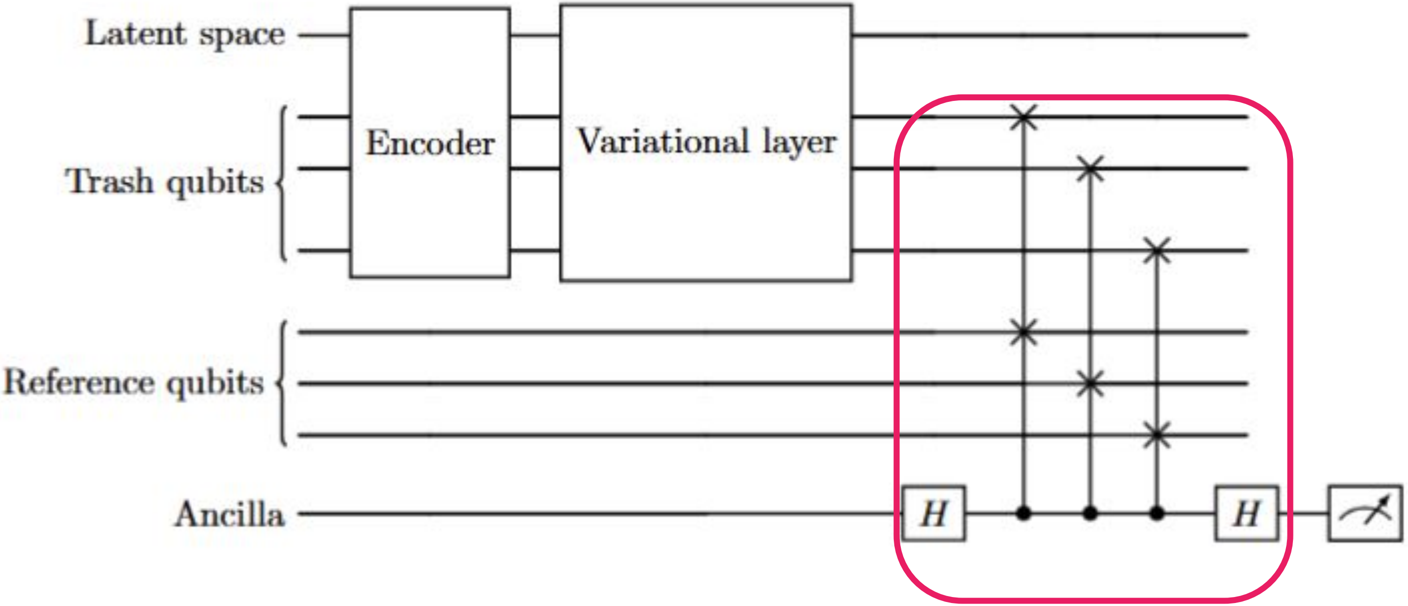


Variational layer

$$U(\Theta) = \boxed{\left(\bigotimes_{i=1}^{N} R_X(\phi_i) R_Y(\theta_i) R_Z(\omega_i)\right)} \otimes \bigotimes_{1 \leq i < j \leq N} C_{ij}$$

- Optimization of angles is done **classically**
- Minimize a cost-function

SWAP test


Latent space
Trash qubits
Reference qubits
Ancilla
Encoder
Variational layer
H
H

Hands-on

Task 1: apply rotation gates in variational layer

Task 2: implement SWAP test

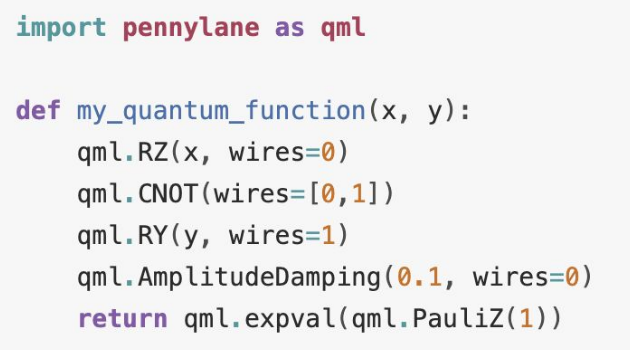
Task 3: compute AUC scores for signal and background

Metrics: fidelity and AUC scores

Question: how can we interpret the fidelity plots?

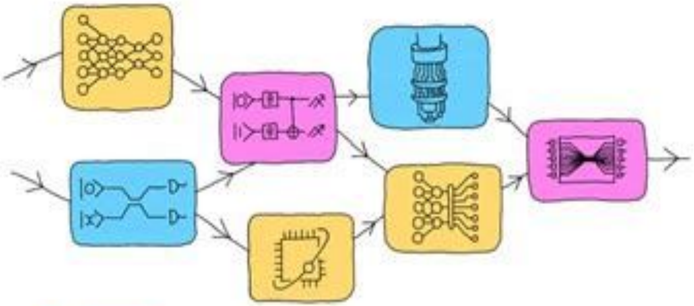
The simulator

- Python-based simulator
- Install using instructions from [https://pennylane.ai/] or running in google_collab



```
import pennylane as qml

def my_quantum_function(x, y):
    qml.RZ(x, wires=0)
    qml.CNOT(wires=[0,1])
    qml.RY(y, wires=1)
    qml.AmplitudeDamping(0.1, wires=0)
    return qml.expval(qml.PauliZ(1))
```



[https://docs.pennylane.ai/en/stable/introduction/operations.html]

Already finished?

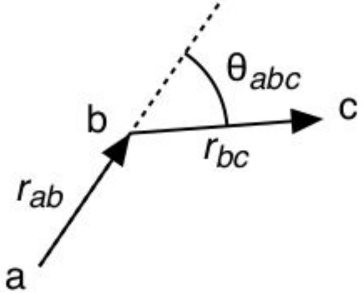
Let’s do tracking

Ising-like Hamiltonian:

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

Tracks, modelled by a collection of **segments**

Segment [S_{ab}]: combination of hit *a* and hit *b*
→ in consecutive layers - for now



Hamiltonian accounts for all possible segments

Hands-on

Question: how many *possible* segments can we expect for 2 tracks, 3 detector layers?

Task 1: define number of qubits

Task 2: implement mixing Hamiltonian with Pauli X gates in all the qubits

Task 3: define cost layers, mixing layers

Useful references

- *Quantum Computation and Quantum Information*, M. A. Nielsen & I. L. Chuang
- *Machine Learning with Quantum Computers*, M. Schuld, F. Petruccione
- Youtube lectures:
  - Pennylane: link
  - Quantum Spain: link
  - CERN: link
- Simulator demos: qiskit, qibo, etc

Thanks for your attention!

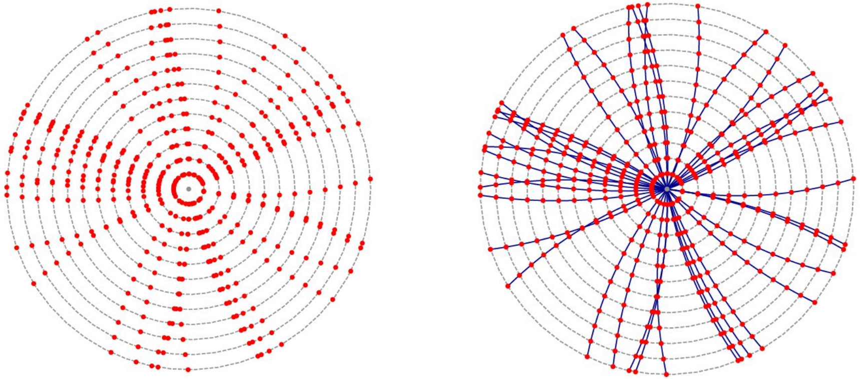
Happy World Quantum Day!!

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

→ LHCb's current method of search by triplet

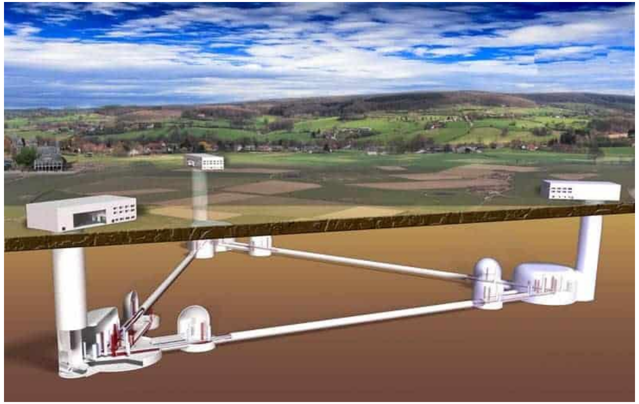
→ Focus of this talk: *global* algorithms



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.