QUANTUM COMPUTING AND HIGH-ENERGY PHYSICS

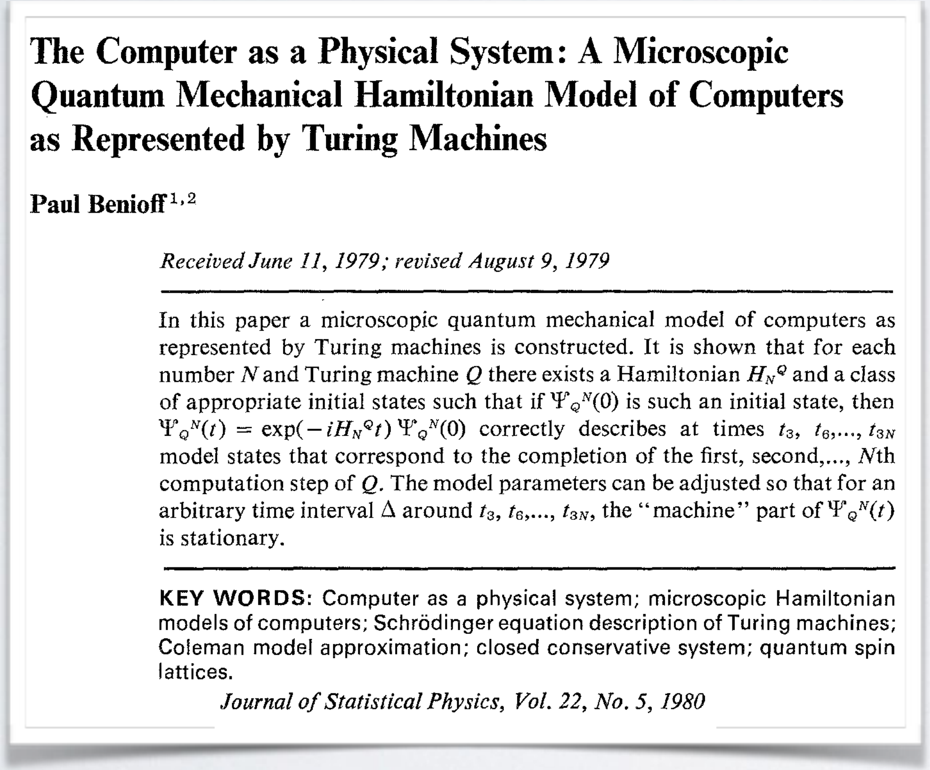
Heather M. Gray
UC Berkeley/LBNL

Berkeley
UNIVERSITY OF CALIFORNIA

BERKELEY LAB

U.S. DEPARTMENT OF
ENERGY
Office of Science

INITIAL IDEAS OF QUANTUM COMPUTING



The Computer as a Physical System: A Microscopic Quantum Mechanical Hamiltonian Model of Computers as Represented by Turing Machines

Paul Benioff[1,2]

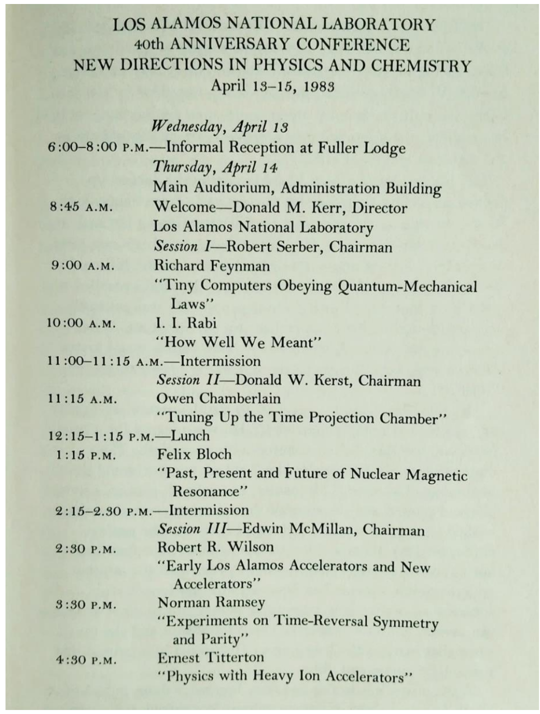
Received June 11, 1979; revised August 9, 1979

In this paper a microscopic quantum mechanical model of computers as represented by Turing machines is constructed. It is shown that for each number N and Turing machine Q there exists a Hamiltonian $H_N{}^Q$ and a class of appropriate initial states such that if $\Psi_Q{}^N(0)$ is such an initial state, then $\Psi_Q{}^N(t) = \exp(-iH_N{}^Q t)\,\Psi_Q{}^N(0)$ correctly describes at times $t_3, t_6, \ldots, t_{3N}$ model states that correspond to the completion of the first, second,..., Nth computation step of Q. The model parameters can be adjusted so that for an arbitrary time interval Δ around $t_3, t_6, \ldots, t_{3N}$, the "machine" part of $\Psi_Q{}^N(t)$ is stationary.

KEY WORDS: Computer as a physical system; microscopic Hamiltonian models of computers; Schrödinger equation description of Turing machines; Coleman model approximation; closed conservative system; quantum spin lattices.

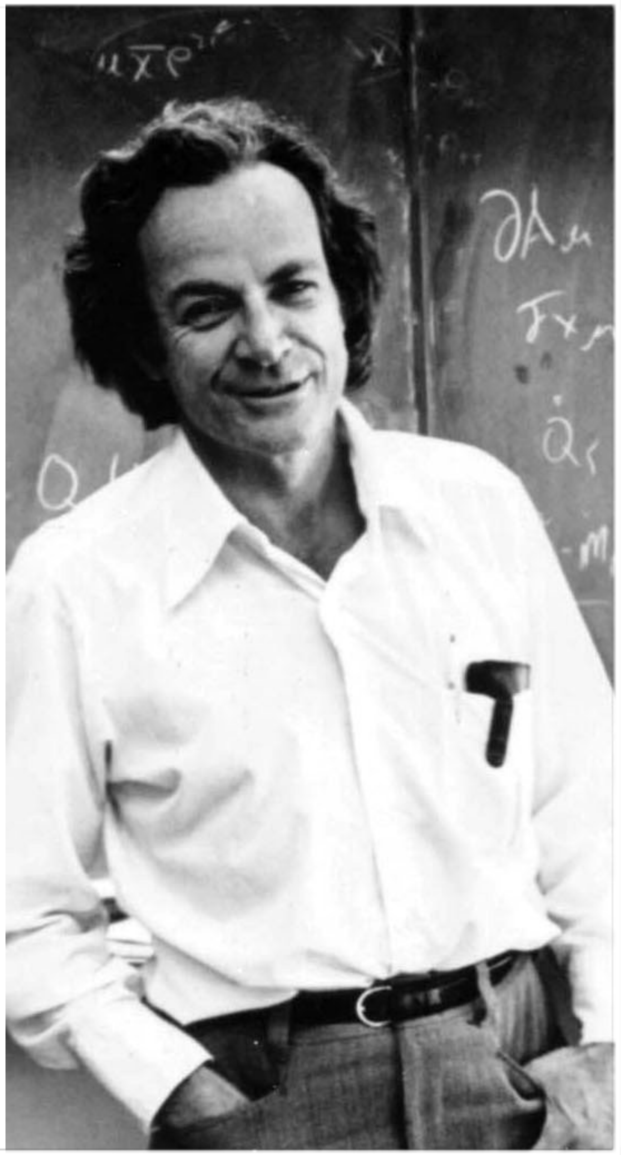
Journal of Statistical Physics, Vol. 22, No. 5, 1980

> *"Let the computer itself be built of quantum mechanical elements which obey quantum mechanical laws."*



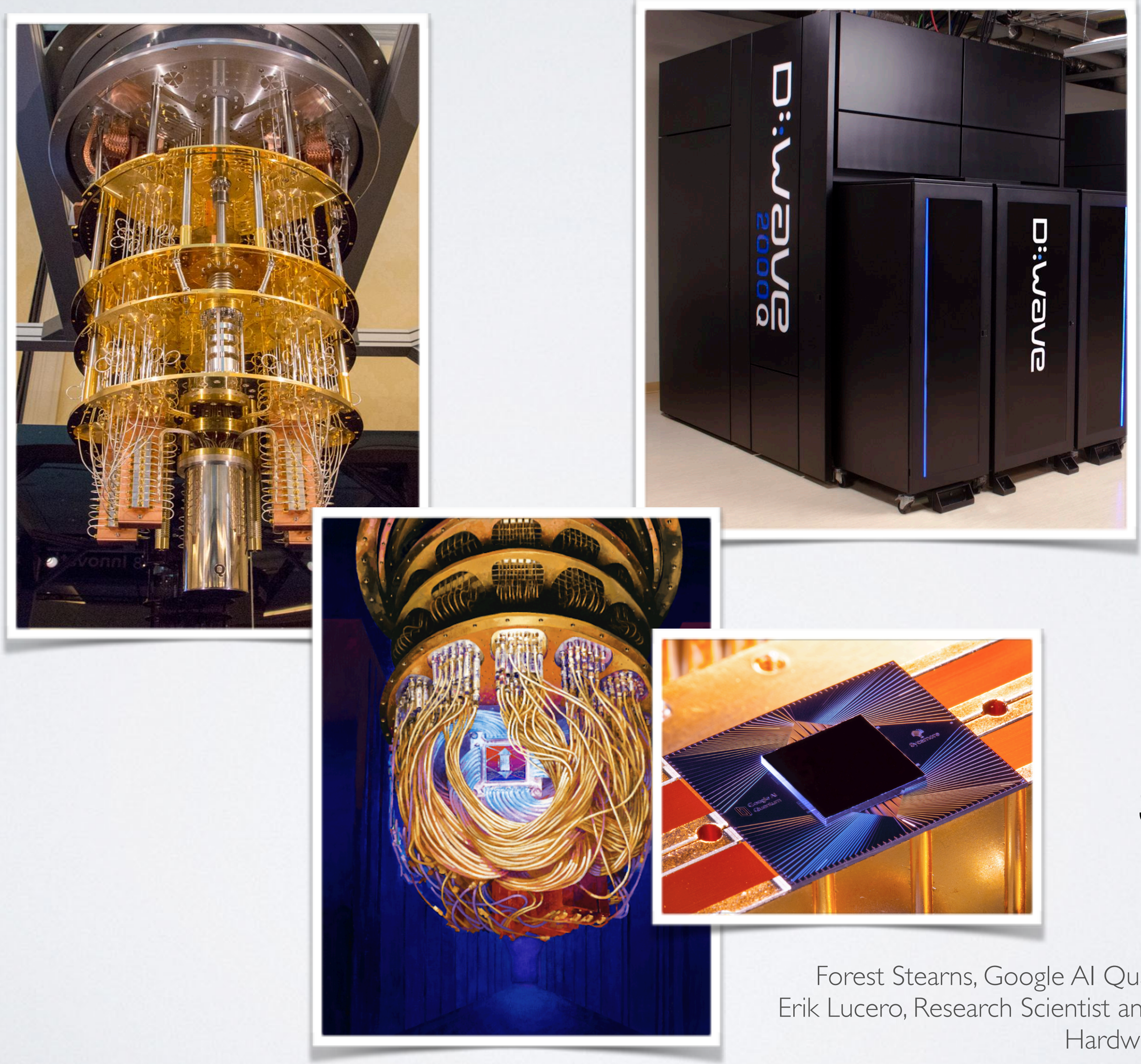
LOS ALAMOS NATIONAL LABORATORY
40th ANNIVERSARY CONFERENCE
NEW DIRECTIONS IN PHYSICS AND CHEMISTRY
April 13–15, 1983

Wednesday, April 13
6:00–8:00 P.M.—Informal Reception at Fuller Lodge
Thursday, April 14
Main Auditorium, Administration Building
8:45 A.M. Welcome—Donald M. Kerr, Director
Los Alamos National Laboratory
Session I—Robert Serber, Chairman
9:00 A.M. Richard Feynman
"Tiny Computers Obeying Quantum-Mechanical Laws"
10:00 A.M. I. I. Rabi
"How Well We Meant"
11:00–11:15 A.M.—Intermission
Session II—Donald W. Kerst, Chairman
11:15 A.M. Owen Chamberlain
"Tuning Up the Time Projection Chamber"
12:15–1:15 P.M.—Lunch
1:15 P.M. Felix Bloch
"Past, Present and Future of Nuclear Magnetic Resonance"
2:15–2.30 P.M.—Intermission
Session III—Edwin McMillan, Chairman
2:30 P.M. Robert R. Wilson
"Early Los Alamos Accelerators and New Accelerators"
3:30 P.M. Norman Ramsey
"Experiments on Time-Reversal Symmetry and Parity"
4:30 P.M. Ernest Titterton
"Physics with Heavy Ion Accelerators"



RICHARD FEYNMAN (1982)

ALMOST 40 YEARS LATER



IBM
20Q
Tokyo
chip

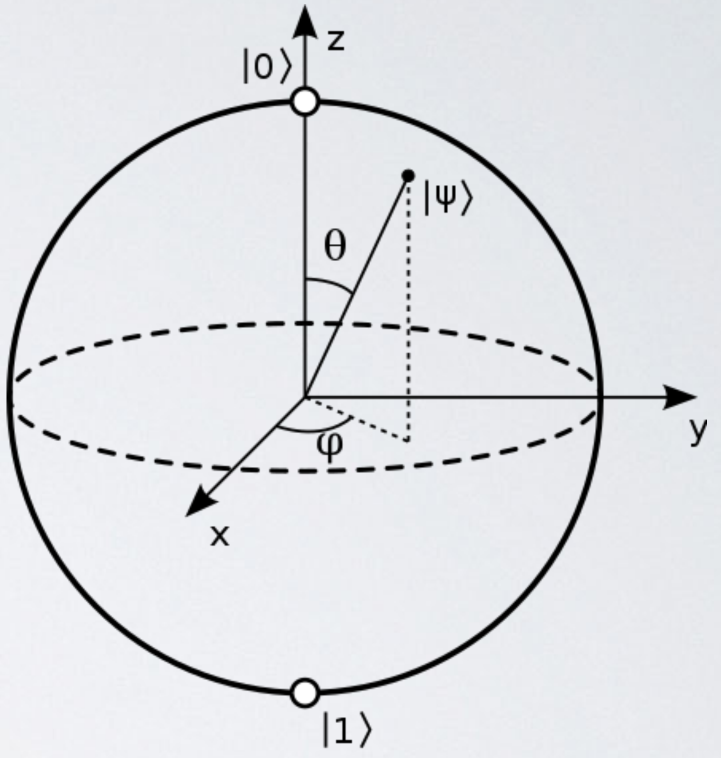
D Wave
2000Q

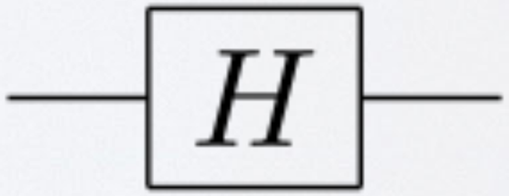
Google
Sycamore

Forest Stearns, Google AI Quantum Artist in Residence
Erik Lucero, Research Scientist and Lead Production Quantum Hardware

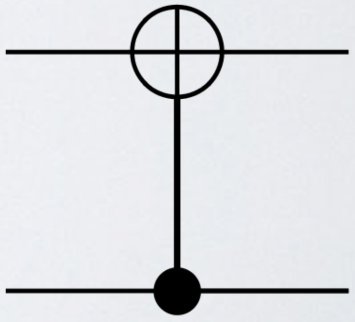
WHAT IS A (UNIVERSAL) QUANTUM COMPUTER?

- Bits → qubits
- Exploit quantum properties: superposition, entanglement, interference
- Quantum logic gates
- Obey unitarity → reversible computing



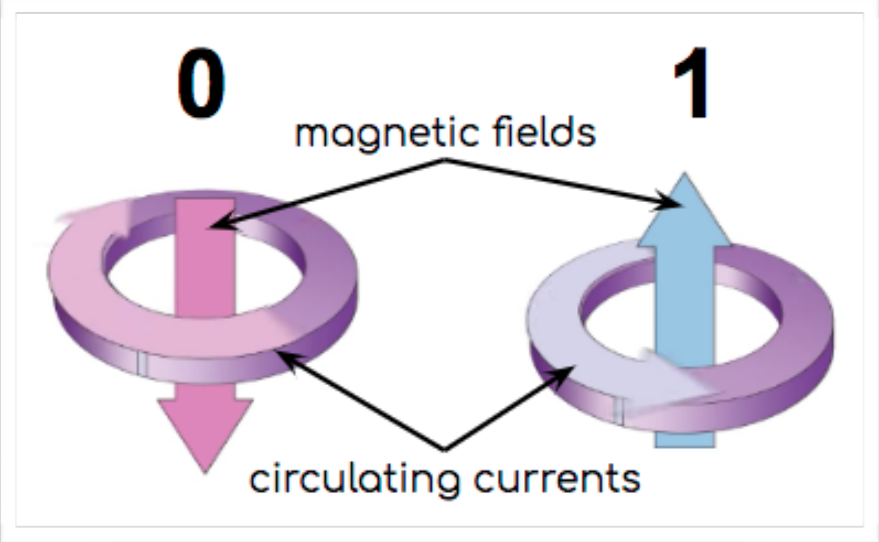


Hadamard

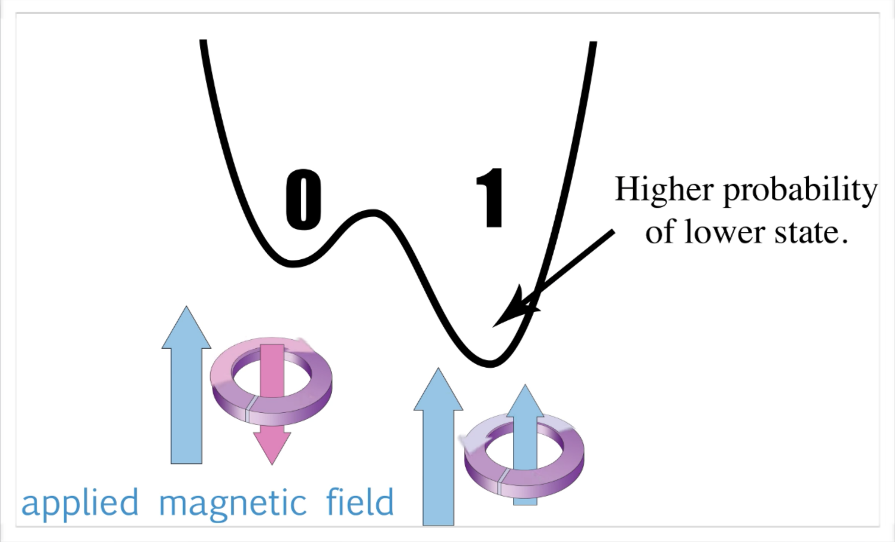


CNOT

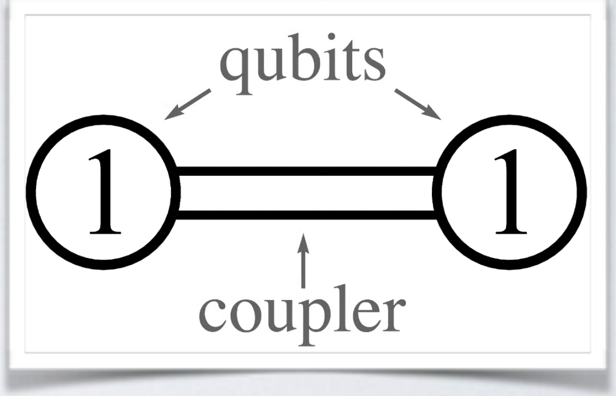
WHAT IS A QUANTUM ANNEALER?



qubits ⇒ q_i



bias weights ⇒ a_i

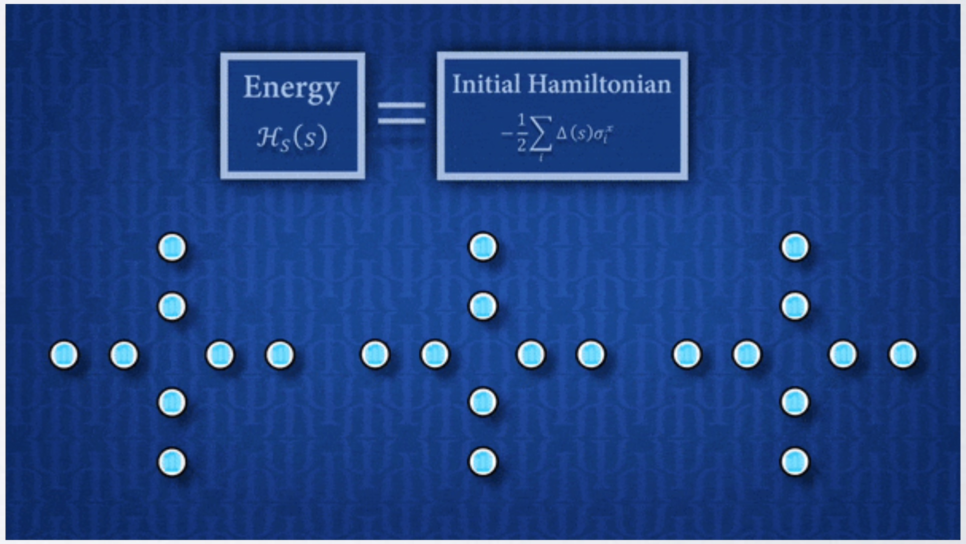


coupling strength ⇒ b_{ij}

$$O(a;b;q) = \sum_{i=1}^{N} a_i q_i + \sum_{i}^{N}\sum_{j}^{N} b_{ij} q_i q_j \qquad q_i \in \{0,1\}$$

QUBO

Quadratic
Unconstrained
Binary
Optimisation



annealing time
~20μs

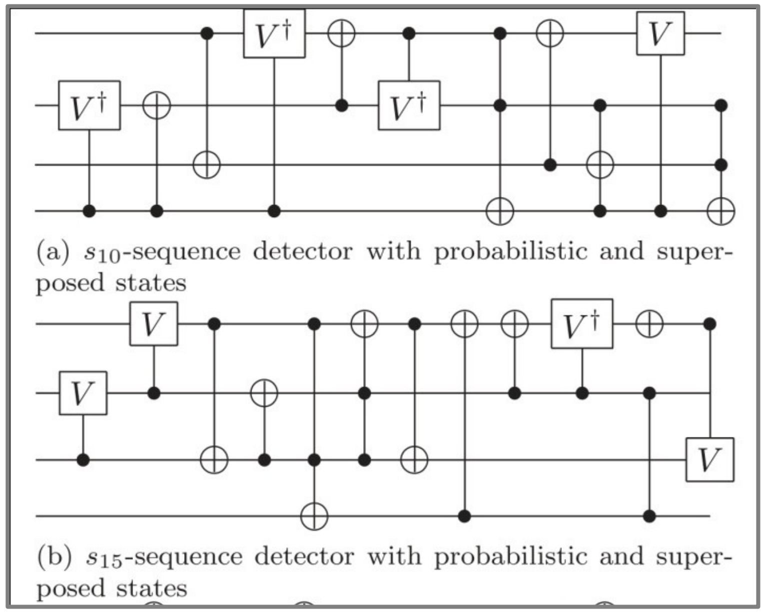
Kadowaki and Nishimori, PRE58 5355, 1998

Glover et al, arXiv:1811.11538

source: dwavesys on YouTube

Slide credit: L. Linder

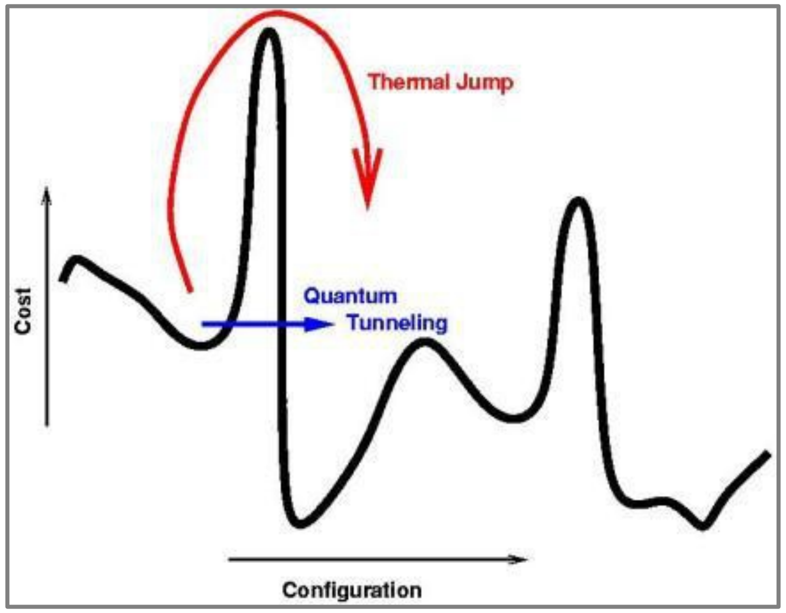
QUBIT AND QUBIT



(a) s_{10}-sequence detector with probabilistic and superposed states

(b) s_{15}-sequence detector with probabilistic and superposed states

<u>Quantum Circuits</u>
Series of quantum gates operating on a set of quantum states.



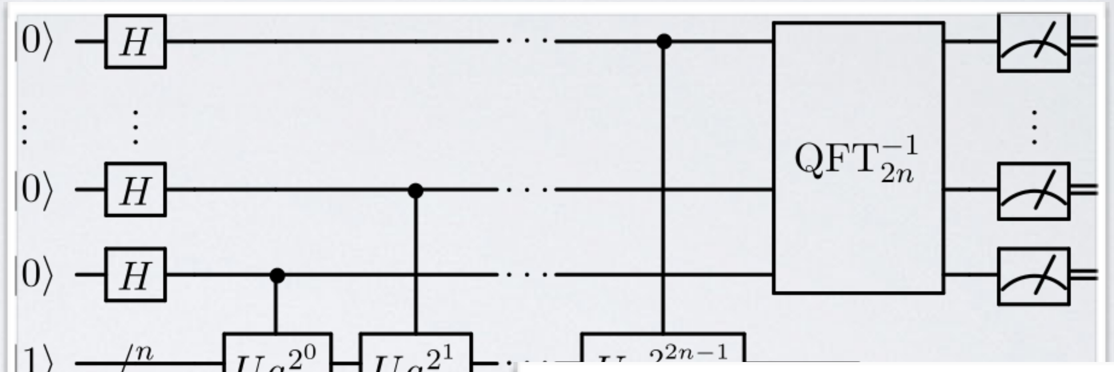
<u>Quantum Annealing</u>
Evolution of a quantum system to a low T Gibbs state
That's D-Wave !

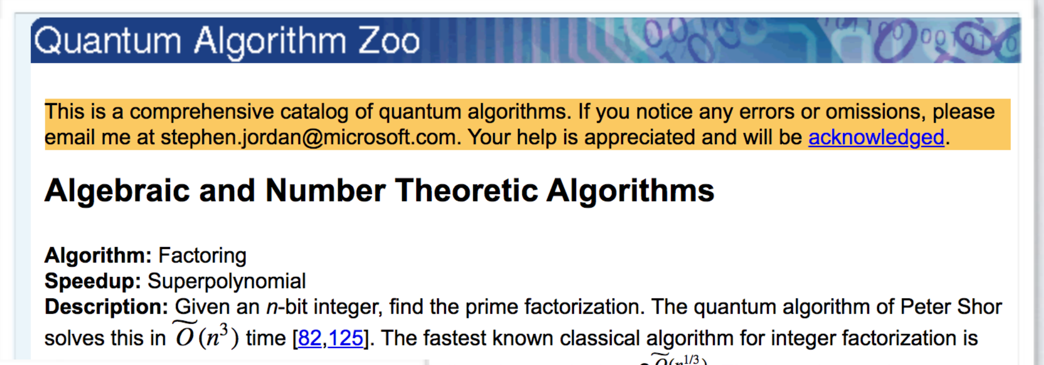
Slide Credit: J.R. Vlimant

WHY ARE PEOPLE EXCITED?

- Quantum cryptography
    - Shor's algorithm
- Quantum simulation
- Quantum search
    - Grover's algorithm
- Huge information capacity
- Quantum machine learning
- Quantum supremacy

Circuit from Shor's Algorithm



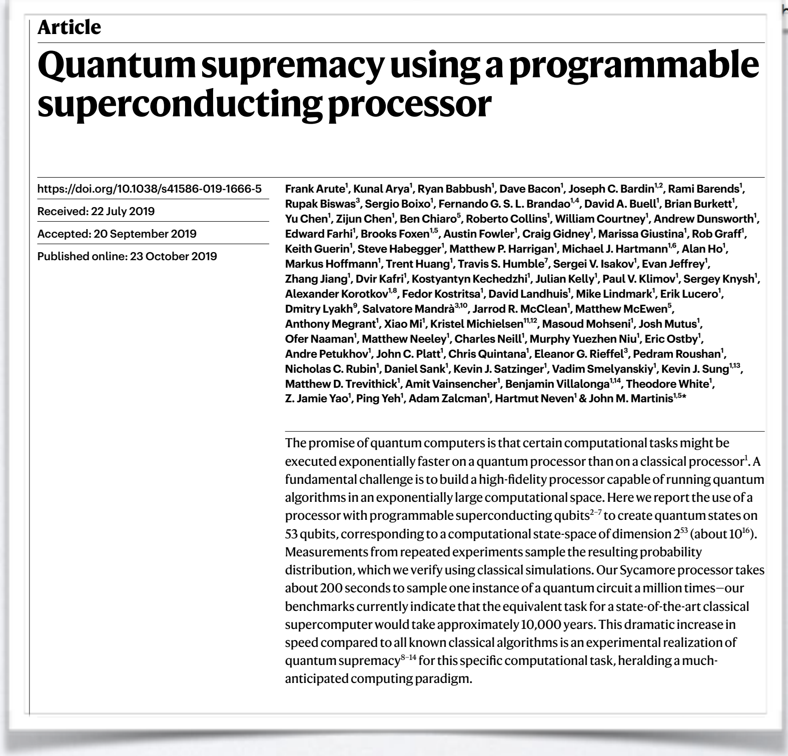


Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@microsoft.com. Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an *n*-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven



Article

Quantum supremacy using a programmable superconducting processor

https://doi.org/10.1038/s41586-019-1666-5

Received: 22 July 2019

Accepted: 20 September 2019

Published online: 23 October 2019

Frank Arute[1], Kunal Arya[1], Ryan Babbush[1], Dave Bacon[1], Joseph C. Bardin[1,2], Rami Barends[1], Rupak Biswas[3], Sergio Boixo[1], Fernando G. S. L. Brandao[1,4], David A. Buell[1], Brian Burkett[1], Yu Chen[1], Zijun Chen[1], Ben Chiaro[5], Roberto Collins[1], William Courtney[1], Andrew Dunsworth[1], Edward Farhi[1], Brooks Foxen[1,5], Austin Fowler[1], Craig Gidney[1], Marissa Giustina[1], Rob Graff[1], Keith Guerin[1], Steve Habegger[1], Matthew P. Harrigan[1], Michael J. Hartmann[1,6], Alan Ho[1], Markus Hoffmann[1], Trent Huang[1], Travis S. Humble[7], Sergei V. Isakov[1], Evan Jeffrey[1], Zhang Jiang[1], Dvir Kafri[1], Kostyantyn Kechedzhi[1], Julian Kelly[1], Paul V. Klimov[1], Sergey Knysh[1], Alexander Korotkov[1,8], Fedor Kostritsa[1], David Landhuis[1], Mike Lindmark[1], Erik Lucero[1], Dmitry Lyakh[9], Salvatore Mandrà[3,10], Jarrod R. McClean[1], Matthew McEwen[5], Anthony Megrant[1], Xiao Mi[1], Kristel Michielsen[11,12], Masoud Mohseni[1], Josh Mutus[1], Ofer Naaman[1], Matthew Neeley[1], Charles Neill[1], Murphy Yuezhen Niu[1], Eric Ostby[1], Andre Petukhov[1], John C. Platt[1], Chris Quintana[1], Eleanor G. Rieffel[3], Pedram Roushan[1], Nicholas C. Rubin[1], Daniel Sank[1], Kevin J. Satzinger[1], Vadim Smelyanskiy[1], Kevin J. Sung[1,13], Matthew D. Trevithick[1], Amit Vainsencher[1], Benjamin Villalonga[1,14], Theodore White[1], Z. Jamie Yao[1], Ping Yeh[1], Adam Zalcman[1], Hartmut Neven[1] & John M. Martinis[1,5*]

The promise of quantum computers is that certain computational tasks might be executed exponentially faster on a quantum processor than on a classical processor[1]. A fundamental challenge is to build a high-fidelity processor capable of running quantum algorithms in an exponentially large computational space. Here we report the use of a processor with programmable superconducting qubits[2–7] to create quantum states on 53 qubits, corresponding to a computational state-space of dimension 2^{53} (about 10^{16}). Measurements from repeated experiments sample the resulting probability distribution, which we verify using classical simulations. Our Sycamore processor takes about 200 seconds to sample one instance of a quantum circuit a million times—our benchmarks currently indicate that the equivalent task for a state-of-the-art classical supercomputer would take approximately 10,000 years. This dramatic increase in speed compared to all known classical algorithms is an experimental realization of quantum supremacy[8–14] for this specific computational task, heralding a much-anticipated computing paradigm.

Quantum zoo

> *"the point when quantum computers can do things that classical computers can't"*

John Preskill, Caltech

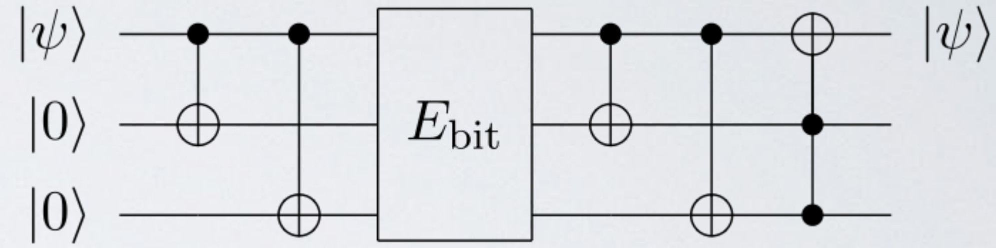
ASIDE: QUANTUM SUPREMACY

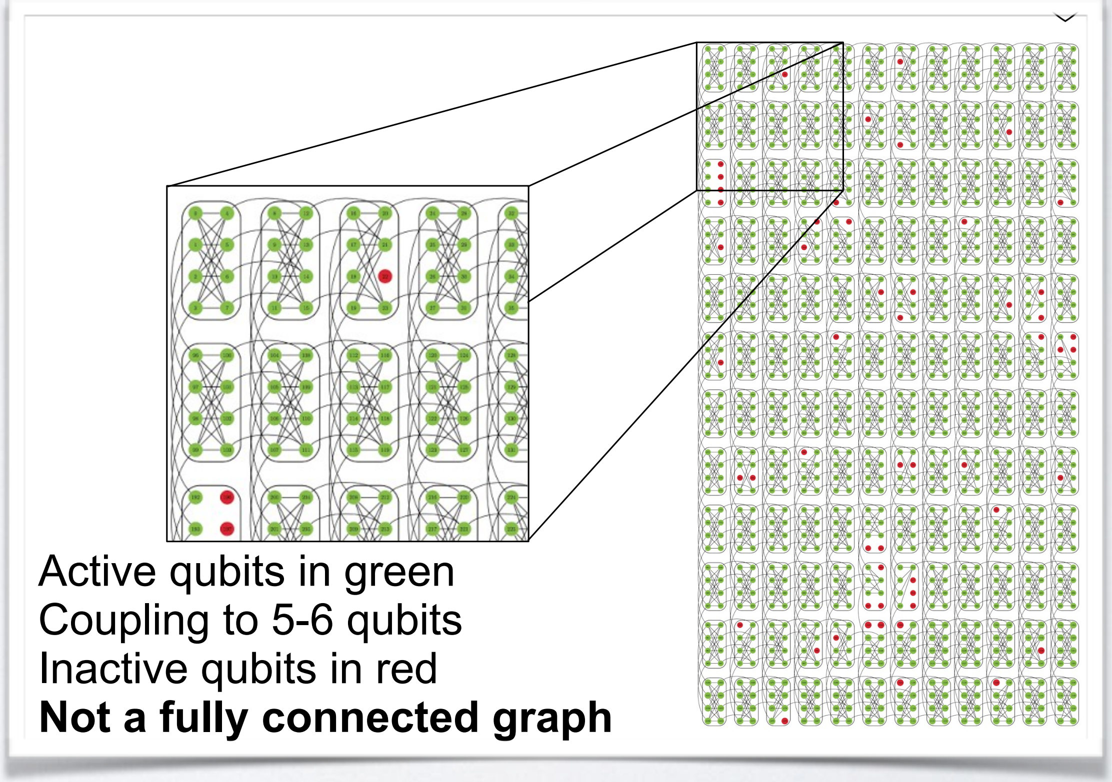
- In Oct 2019, Google published a paper in Nature claiming they had achieved quantum supremacy by solving a problem in 200s on Sycamore that would take Summit 10k years
    - The problem: sampling numbers from a pseudo-random quantum circuit
- Response from IBM: "*We argue that an ideal simulation of the same task can be performed on a classical system in 2.5 days and with far greater fidelity.* "
    - *They argue that Google had neglected to account for disk space*

WHAT ARE THE PROBLEMS?

- Quantum decoherence
- Quantum noise
  - Quantum error correcting codes
- Scalability (typically O(10s) qubits)
- Connectivity

Quantum error correcting code





Active qubits in green
Coupling to 5-6 qubits
Inactive qubits in red
Not a fully connected graph

Credit: J.R. Vlimant

Dyakanov, The Case Against Quantum Computing

UNIVERSAL QUANTUM COMPUTERS

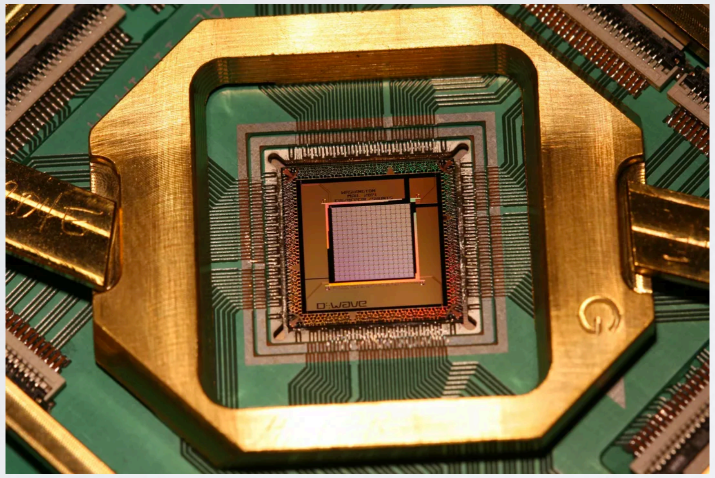
Circuit-based quantum processors [edit]

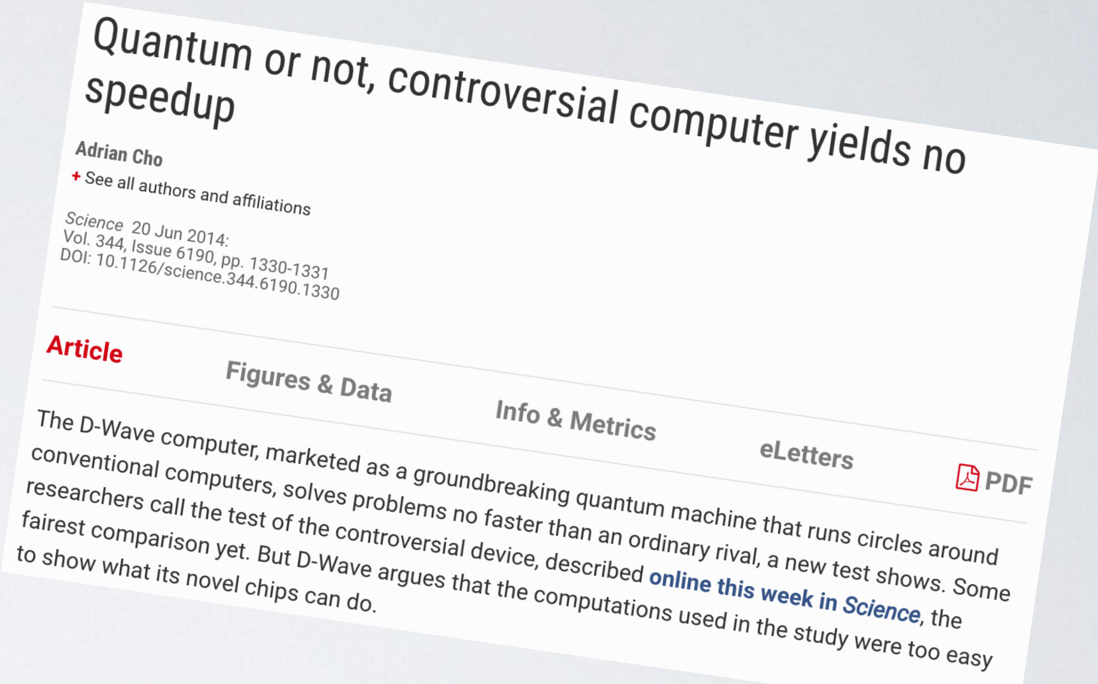
These QPUs are based on the quantum circuit and quantum logic gate-based model of computing.

Manufacturer	Name/Codename/Designation	Architecture	Layout	Socket	Fidelity	Qubits	Release date
Google	Bristlecone	Superconducting	6×12 lattice	N/A	99% (readout) 99.9% (1 qubit) 99.4% (2 qubits)	72 qb[3][4]	5 March 2018
Google	Sycamore	Nonlinear superconducting resonator	N/A	N/A	N/A	54 transmon qb 53 qb effective	2019
IBM	IBM Q 53	Superconducting	N/A	N/A	N/A	53 qb	October 2019
IBM	IBM Q 50 prototype	Superconducting	N/A	N/A	N/A	50 qb[7]	
Google	N/A	Superconducting	7×7 lattice	N/A	99.7%[1]	49 qb[2]	Q4 2017 (planned)
Intel	Tangle Lake	Superconducting	N/A	108-pin cross gap	N/A	49 qb [10]	9 January 2018
Google	N/A	Superconducting	N/A	N/A	99.5%[1]	20 qb	2017
IBM	IBM Q 20 Tokyo	Superconducting	5x4 lattice	N/A	99.812% (average gate) 93.21% (readout)	20 qb[7]	10 November 2017

Quantum processors on wikipedia

QUANTUM ANNEALERS





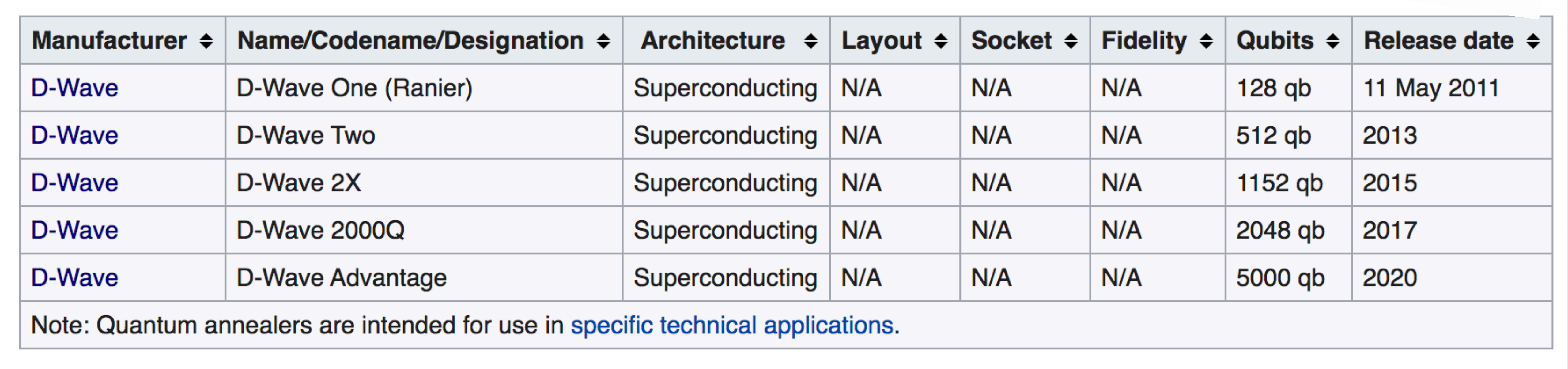
Quantum or not, controversial computer yields no speedup

Adrian Cho

+ See all authors and affiliations

Science 20 Jun 2014:
Vol. 344, Issue 6190, pp. 1330-1331
DOI: 10.1126/science.344.6190.1330

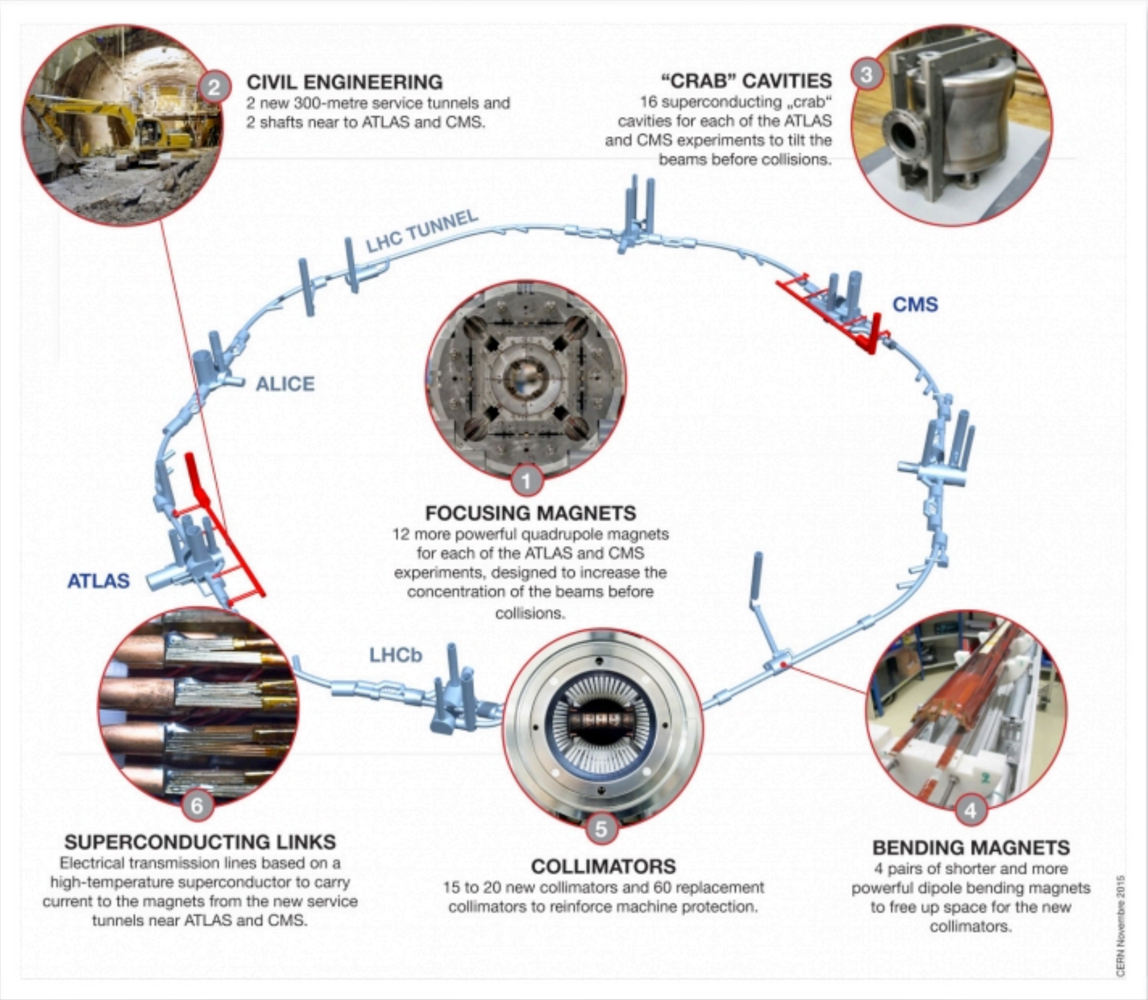
The D-Wave computer, marketed as a groundbreaking quantum machine that runs circles around conventional computers, solves problems no faster than an ordinary rival, a new test shows. Some researchers call the test of the controversial device, described online this week in *Science*, the fairest comparison yet. But D-Wave argues that the computations used in the study were too easy to show what its novel chips can do.

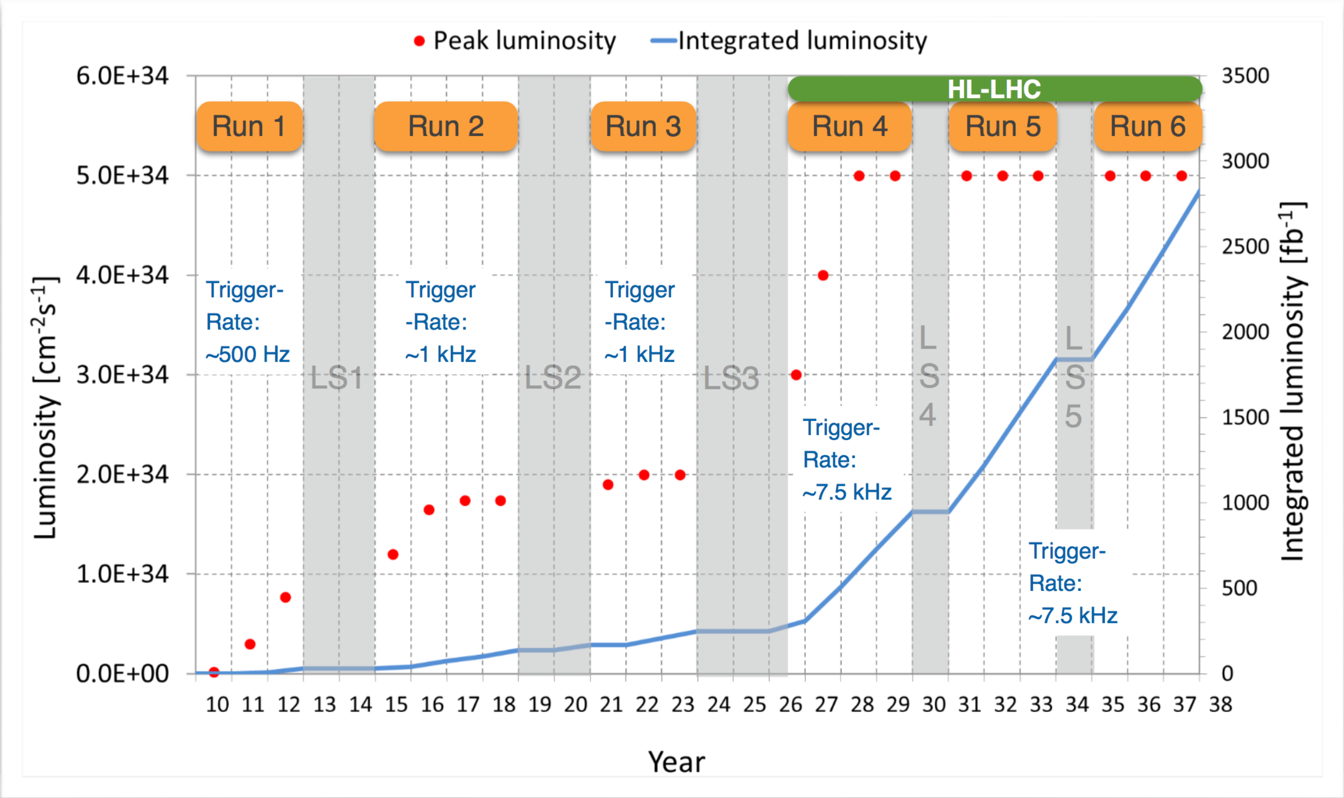


Manufacturer	Name/Codename/Designation	Architecture	Layout	Socket	Fidelity	Qubits	Release date
D-Wave	D-Wave One (Ranier)	Superconducting	N/A	N/A	N/A	128 qb	11 May 2011
D-Wave	D-Wave Two	Superconducting	N/A	N/A	N/A	512 qb	2013
D-Wave	D-Wave 2X	Superconducting	N/A	N/A	N/A	1152 qb	2015
D-Wave	D-Wave 2000Q	Superconducting	N/A	N/A	N/A	2048 qb	2017
D-Wave	D-Wave Advantage	Superconducting	N/A	N/A	N/A	5000 qb	2020

Note: Quantum annealers are intended for use in specific technical applications.

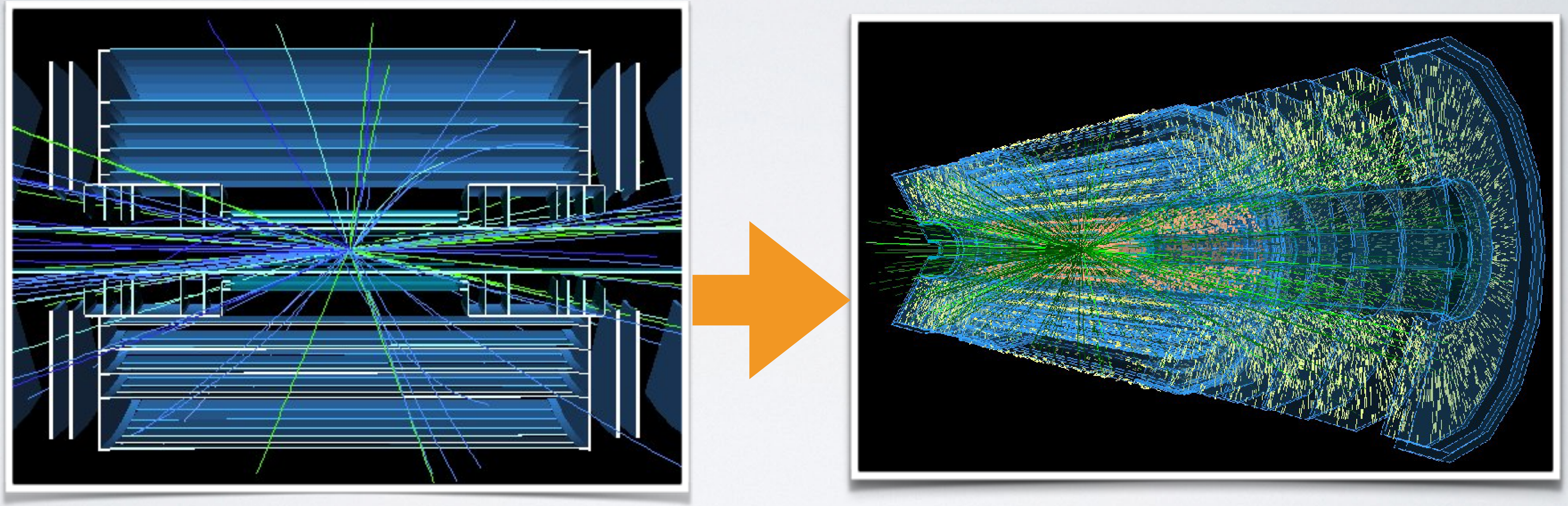
UPGRADE ALERT: HL-LHC





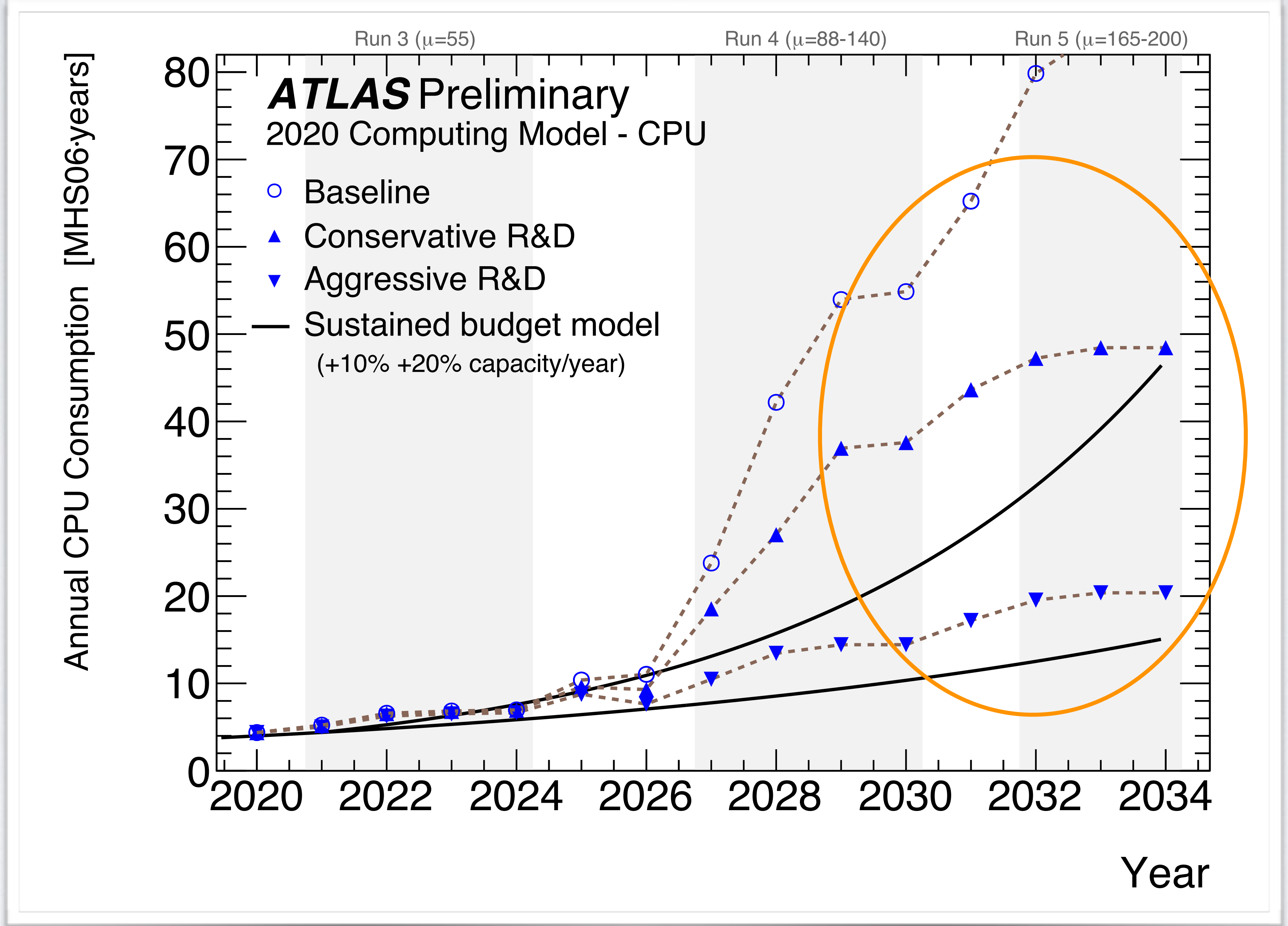
Great for physics … but a challenge for computing

HL-LHC COMPUTING CHALLENGE



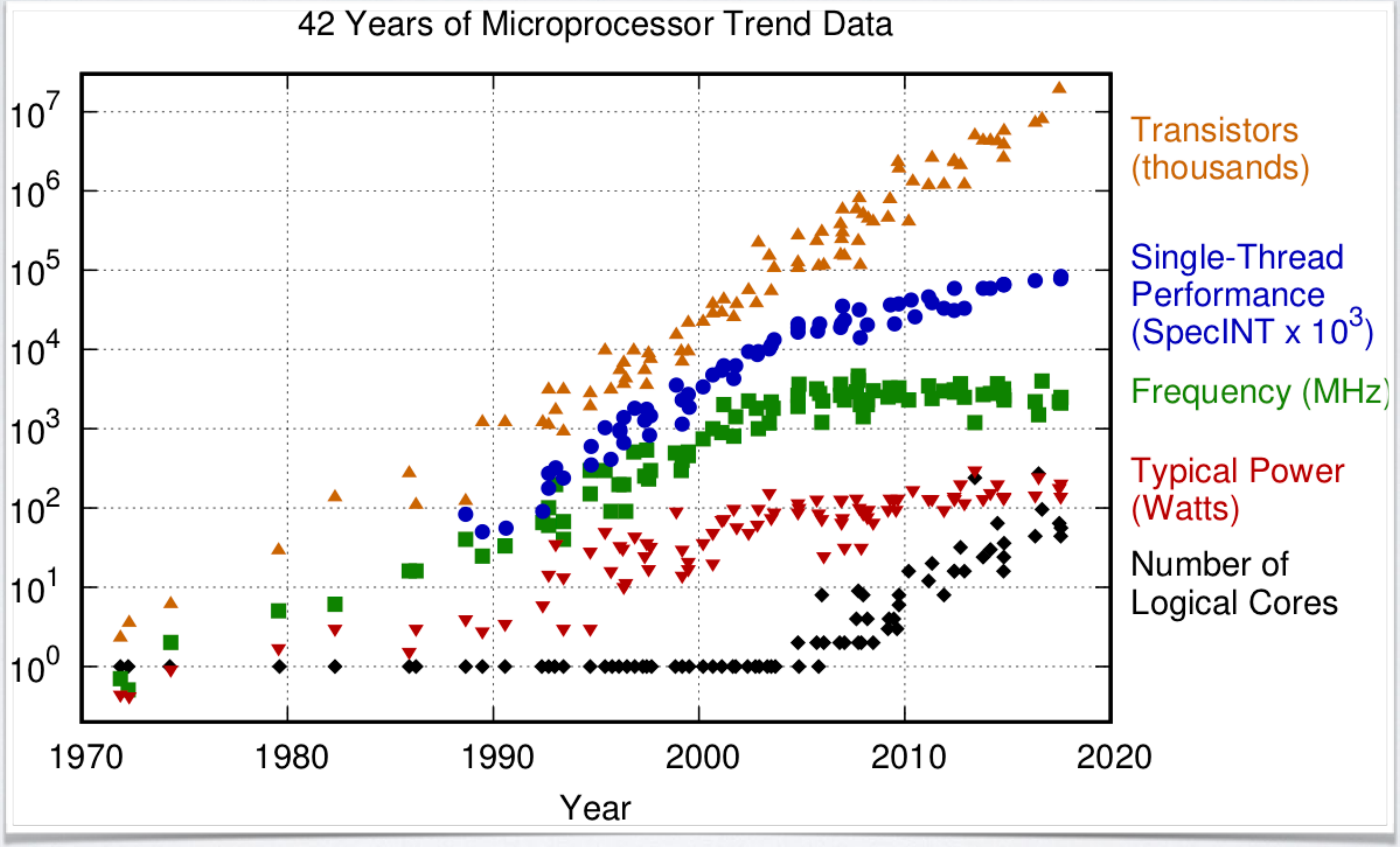
Combinatoric explosion that naively scales as n!

TRACKING: KEY FOR HL-LHC

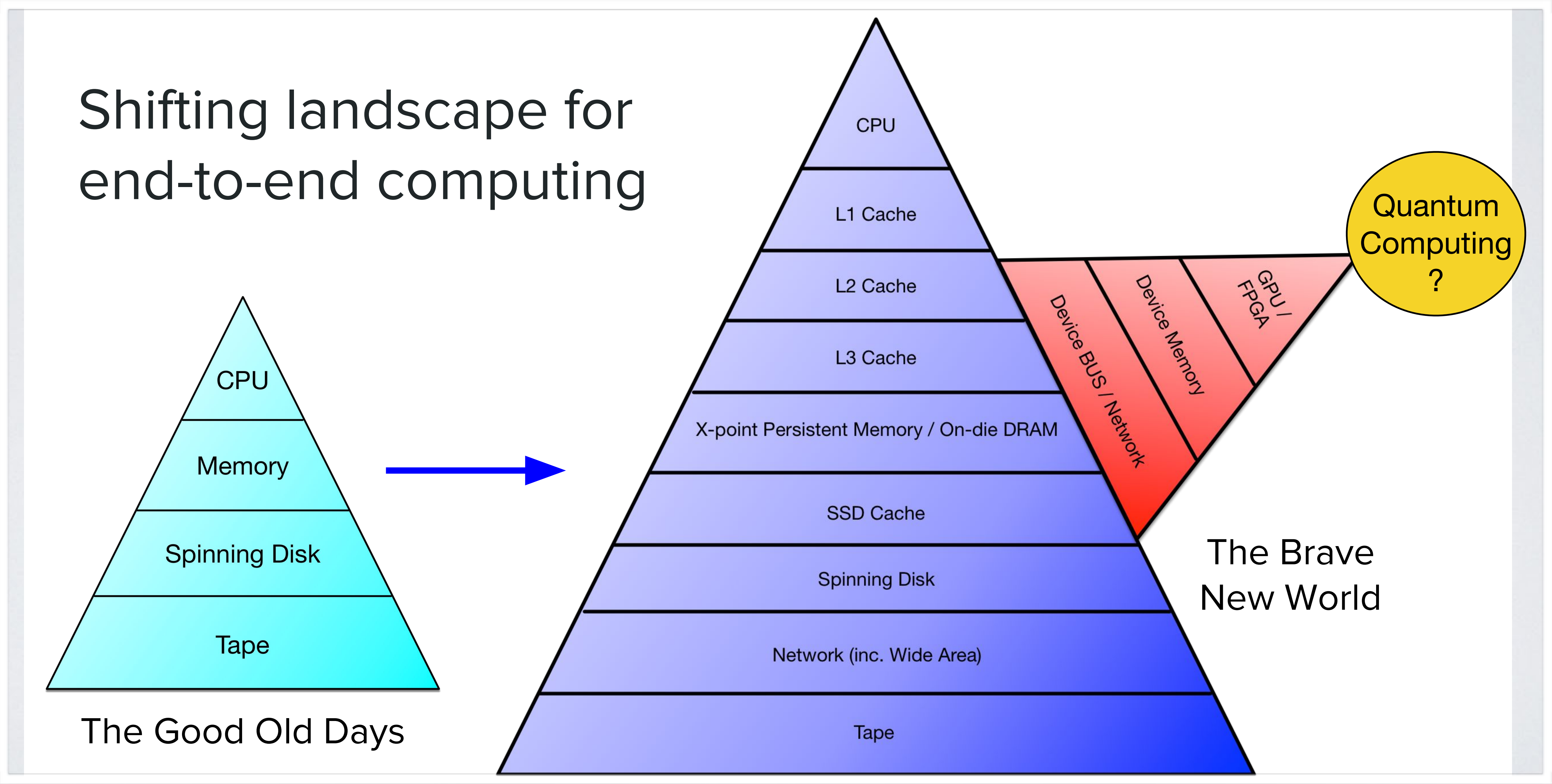


ATLAS S&C twiki

A SECOND PROBLEM: TECHNOLOGY



SHIFTING COMPUTING LANDSCAPE

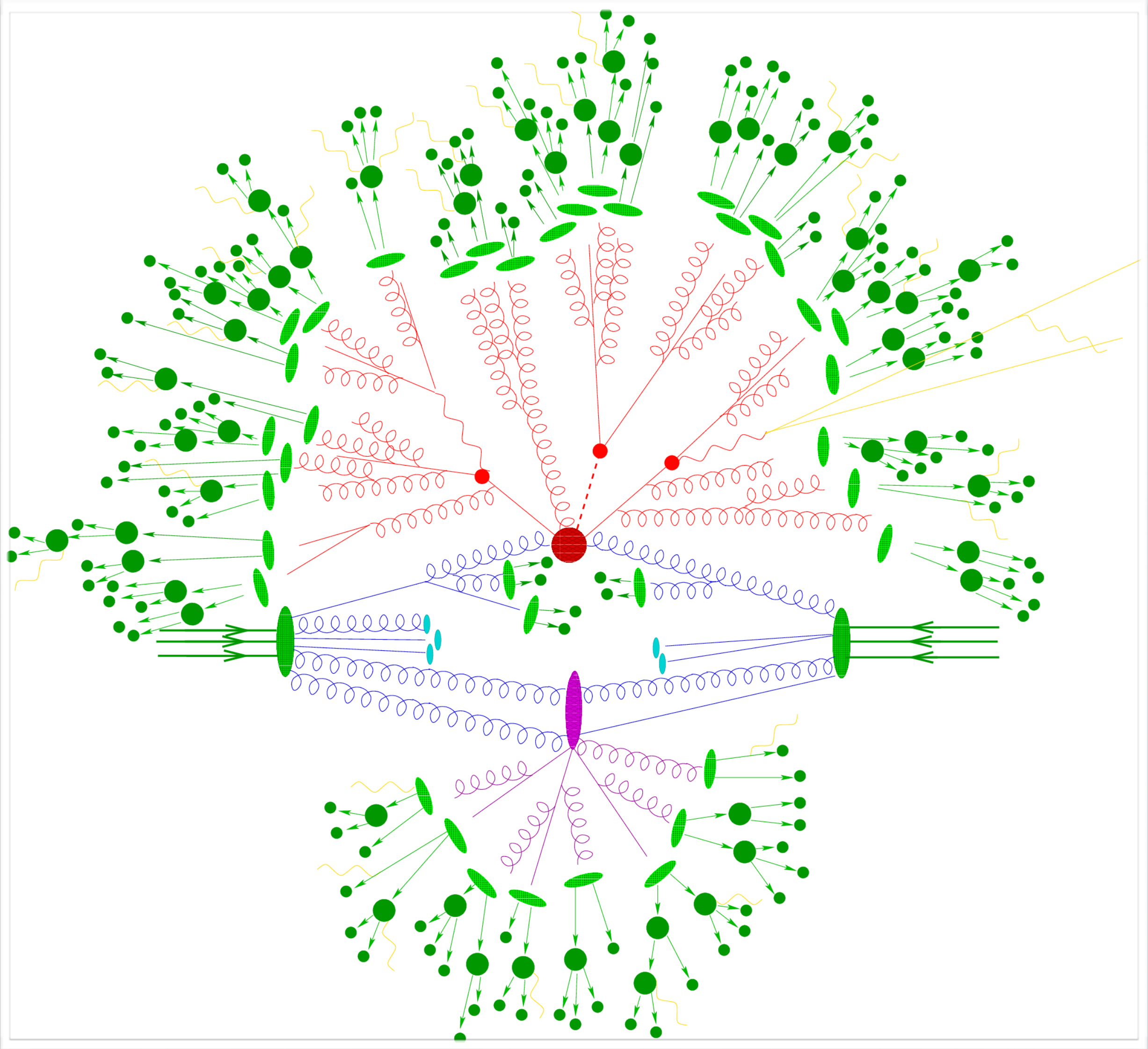


Slide Credit: G. Stewart

COULD QUANTUM COMPUTING BE USEFUL FOR PARTICLE PHYSICS?

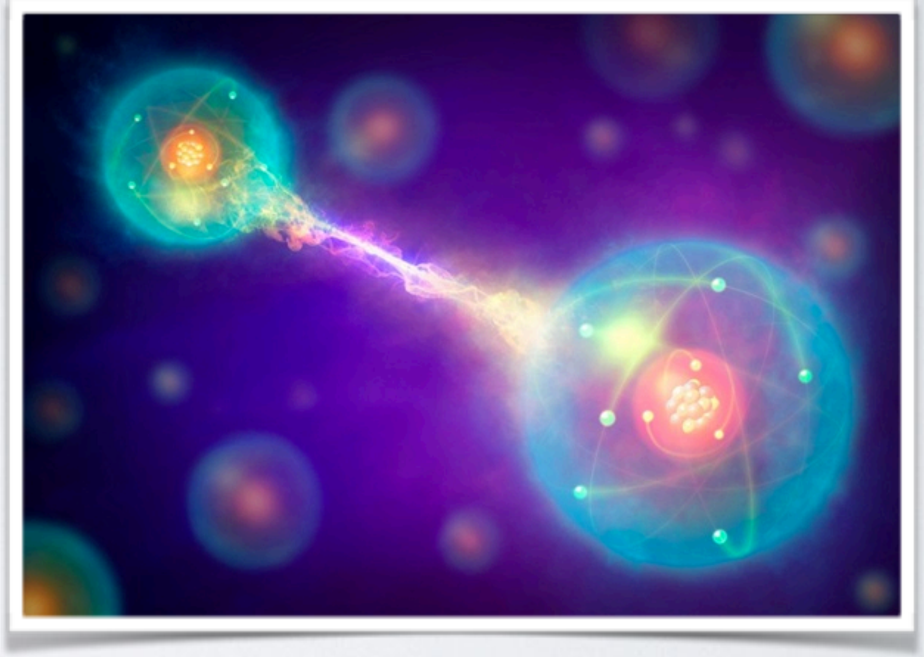
SIMULATING CORRELATIONS

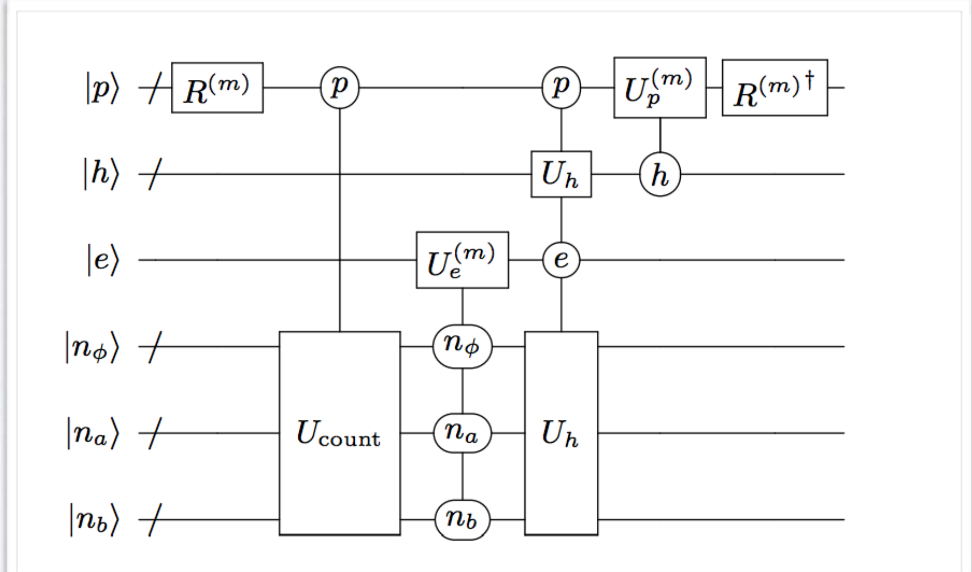
Currently simulate events assuming the evolution of each particle is independent



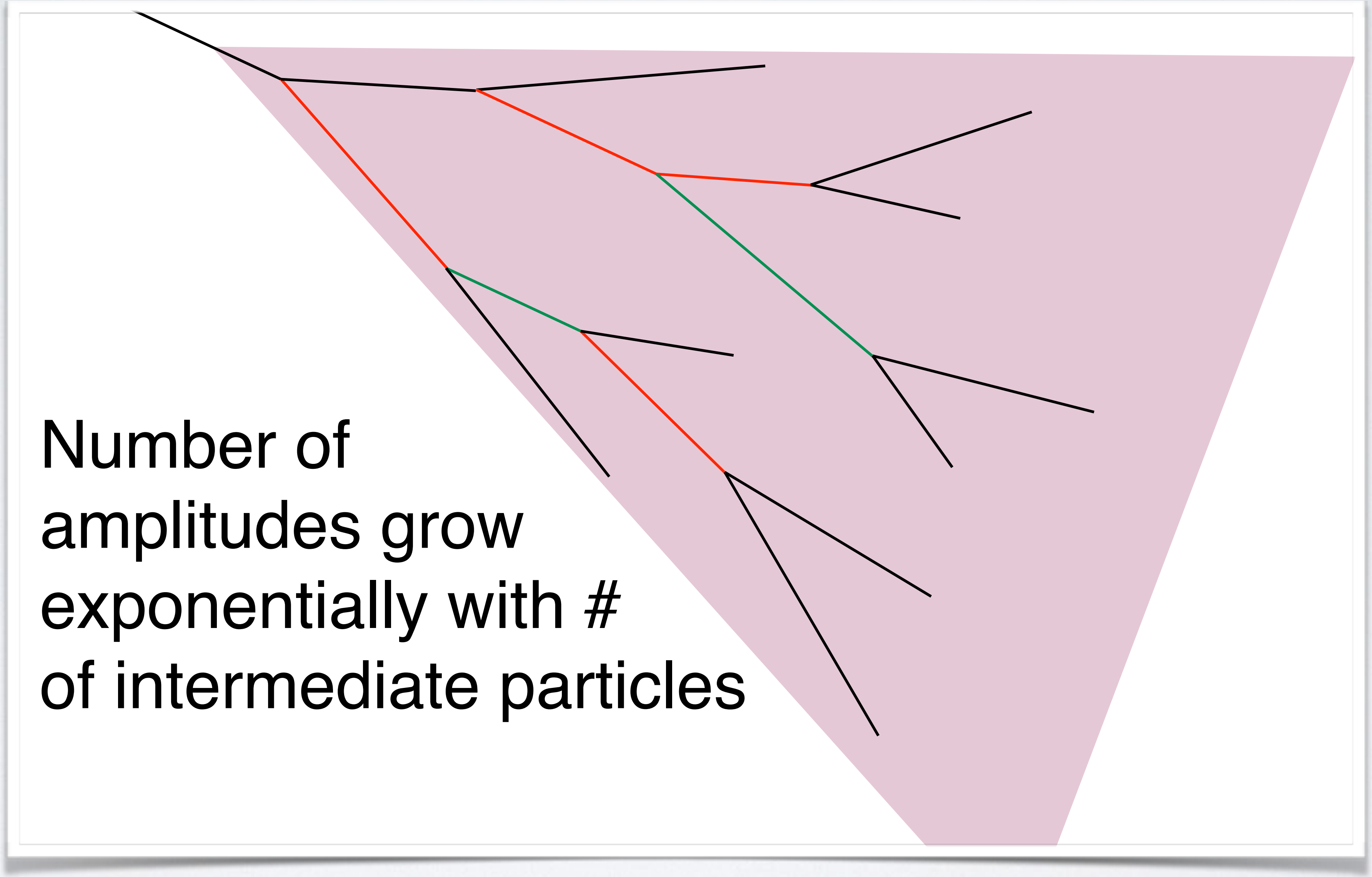
ENTANGLEMENT

- This isn't the full picture
  - Particles are quantum mechanical objects
  - Correlations exist between them
- Idea: exploit entanglement between qubits on a quantum computer to improve the description of the parton shower





CLASSICALLY



Number of amplitudes grow exponentially with # of intermediate particles

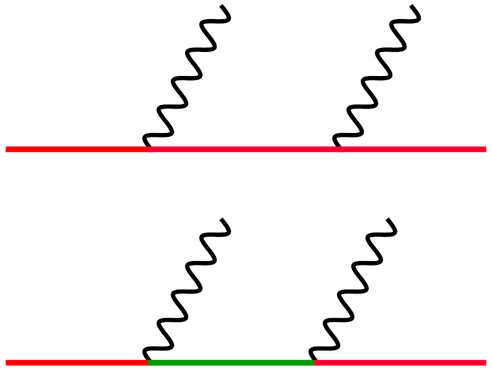
Slide Credit: C. Bauer

TOY MODEL

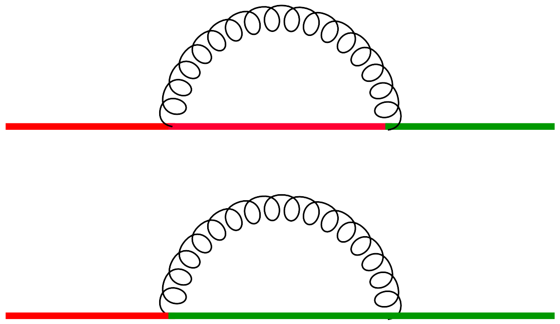
$$\begin{aligned}\mathcal{L} = &\bar{f}_1(i\partial\!\!\!/ + m_1)f_1 + \bar{f}_2(i\partial\!\!\!/ + m_2)f_2 + (\partial_\mu\phi)^2 \\ &+ g_1\bar{f}_1 f_1\phi + g_2\bar{f}_2 f_2\phi + g_{12}\left[\bar{f}_1 f_2 + \bar{f}_2 f_1\right]\phi\end{aligned}$$

The mixing g_{12} gives several interesting effects

Different real emission amplitudes give rise to interference



Virtual diagrams give rise to flavor change without radiation

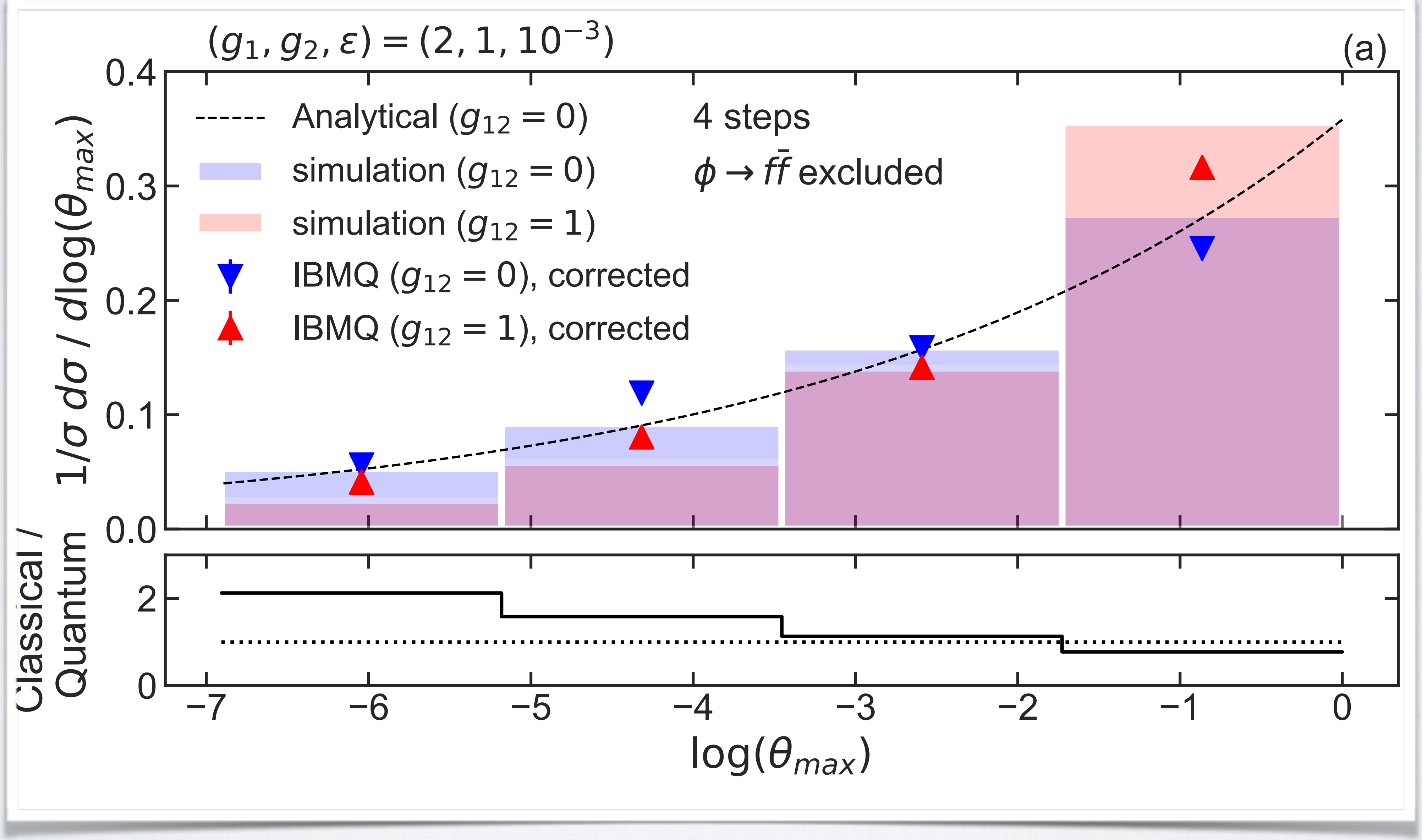


Need to correct both real and virtual effects

Similar to including subleading color

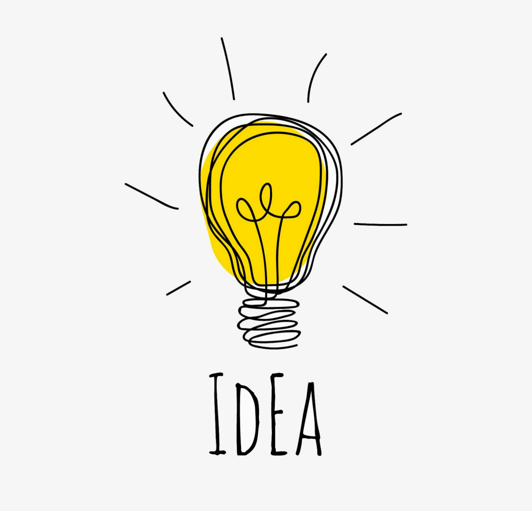
Slide credit: C. Bauer

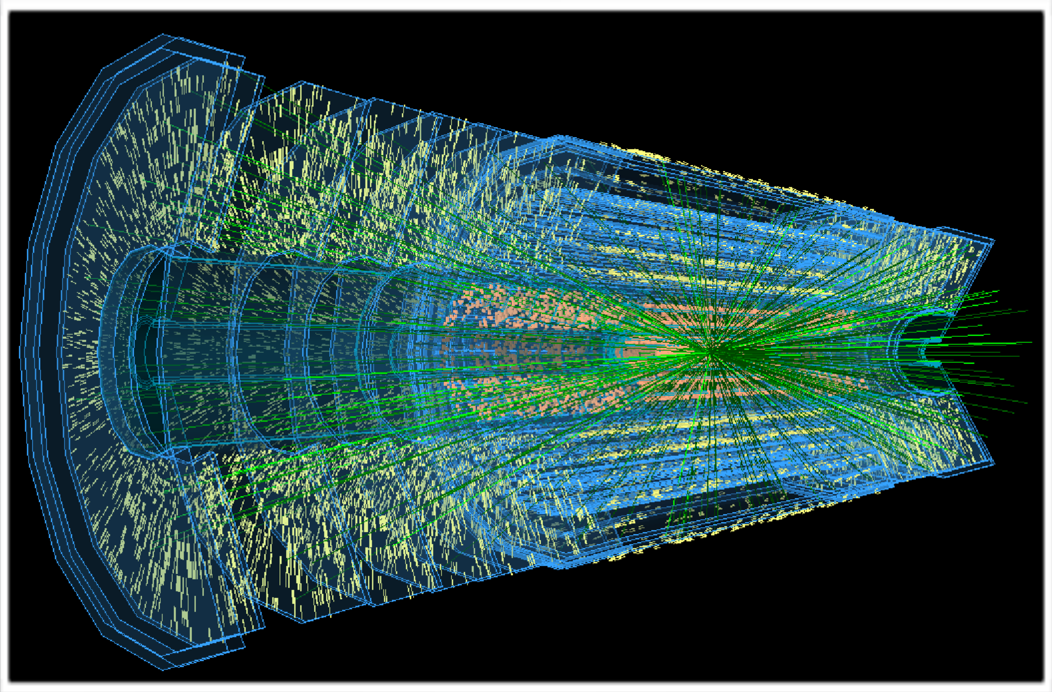
EARLY RESULTS

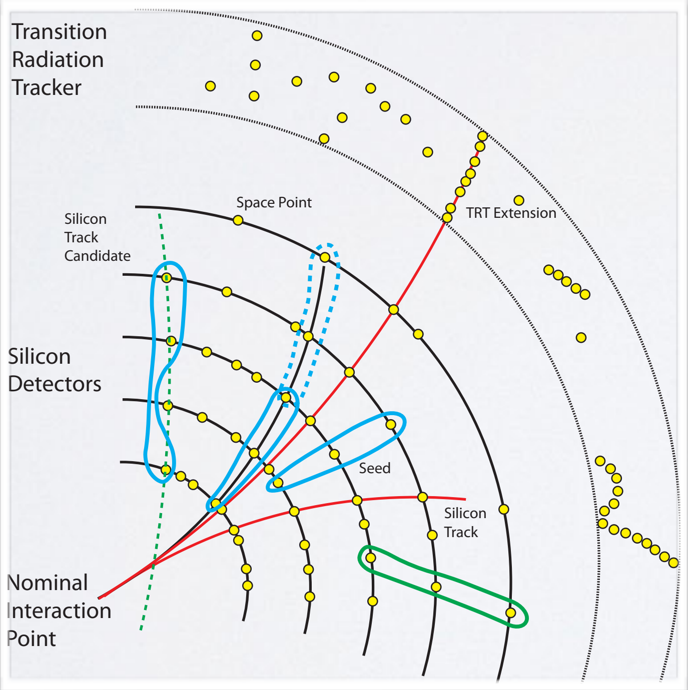


arxiv:1904.03196

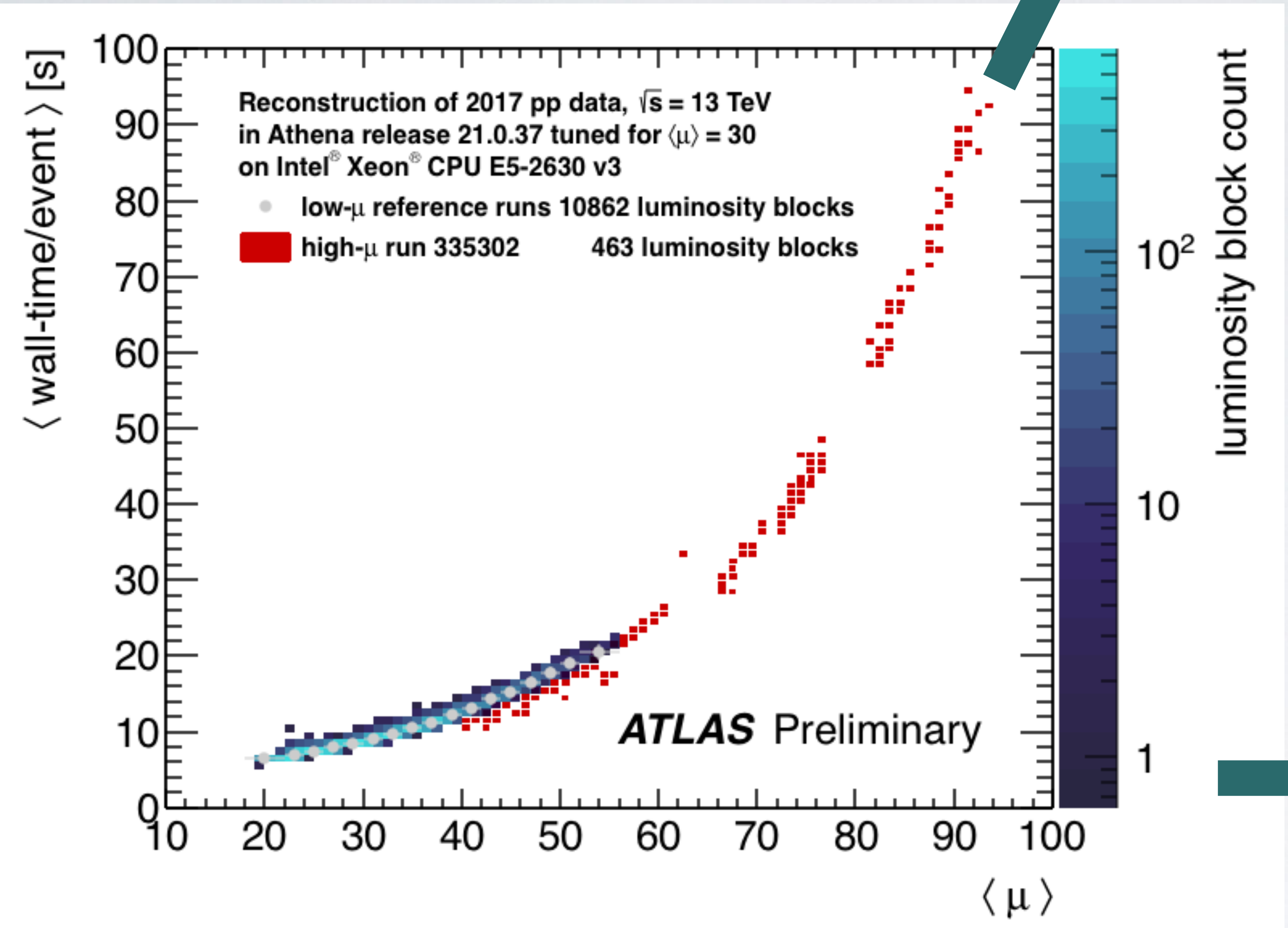
RECONSTRUCTING TRACKS







https://sites.google.com/lbl.gov/hep-qpr



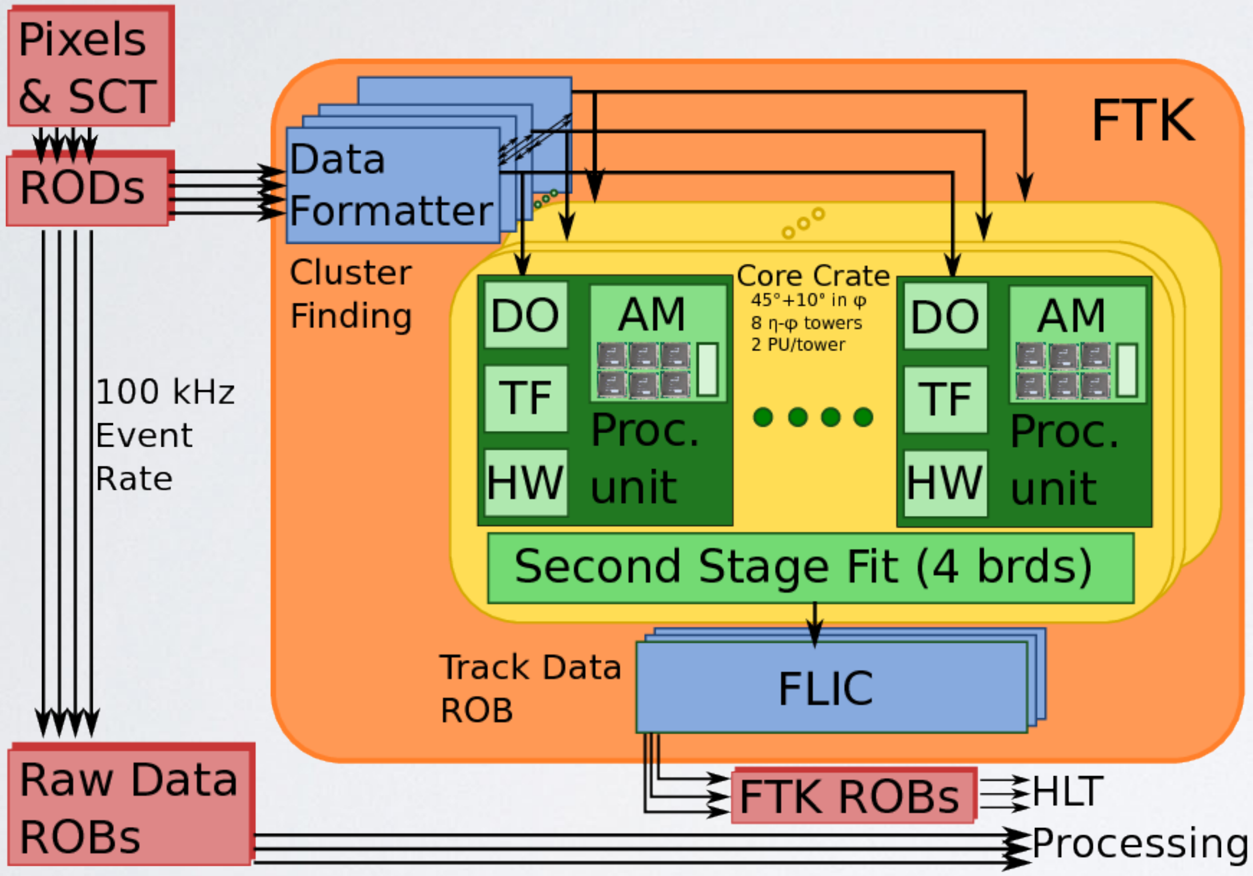
Track reconstruction is expected to have the largest CPU burden at the HL-LHC

HL-LHC: μ= 140-200

Storage is also (probably more significant) problem

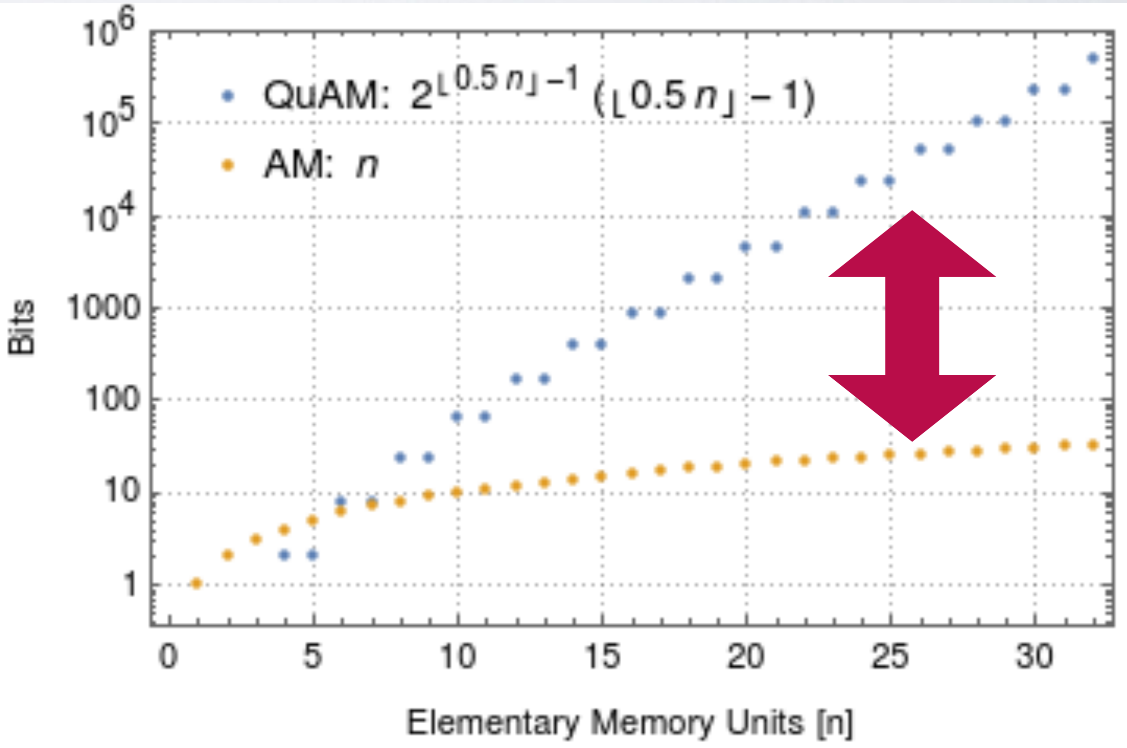
ATLAS S&C twiki

DIFFERENT ALGORITHMS: ASSOCIATIVE MEMORY



Inspired by ideas for FTK

Memory required scales far more slowly with the number of tracks



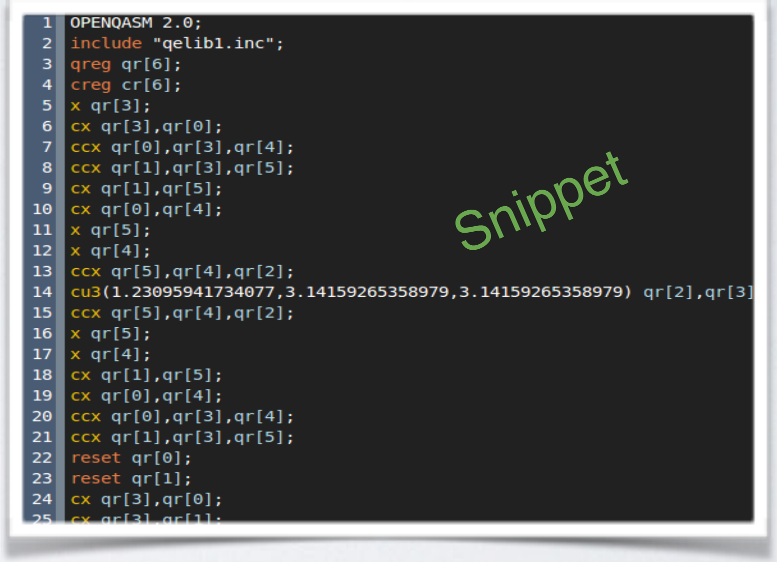
arXiv:1902.00498

Slide credit: I. Shapoval

IMPLEMENTATION (I)

- We developed QuAM circuit generators implementing the Trugenberger's initialization and generalized Grover's algorithms.
  - use open-source quantum computing platform, Qiskit
- Supported backends
  - IBM QE cloud-based quantum chips [5Q Yorktown/Tenerife, 14Q Melbourne, 20Q Tokyo]
  - Local/remote noisy simulators

Storage QuAM



```
OPENQASM 2.0;
include "qelib1.inc";
qreg qr[6];
creg cr[6];
x qr[3];
cx qr[3],qr[0];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
cx qr[1],qr[5];
cx qr[0],qr[4];
x qr[5];
x qr[4];
ccx qr[5],qr[4],qr[2];
cu3(1.23095941734077,3.14159265358979,3.14159265358979) qr[2],qr[3]
ccx qr[5],qr[4],qr[2];
x qr[5];
x qr[4];
cx qr[1],qr[5];
cx qr[0],qr[4];
ccx qr[0],qr[3],qr[4];
ccx qr[1],qr[3],qr[5];
reset qr[0];
reset qr[1];
cx qr[3],qr[0];
cx qr[3],qr[1];
```

Retrieval QuAM

```
s qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
s qr[5];
h qr[4];
h qr[5];
x qr[4];
x qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
x qr[4];
x qr[5];
h qr[4];
h qr[5];
h qr[5];
cx qr[4],qr[5];
h qr[5];
```

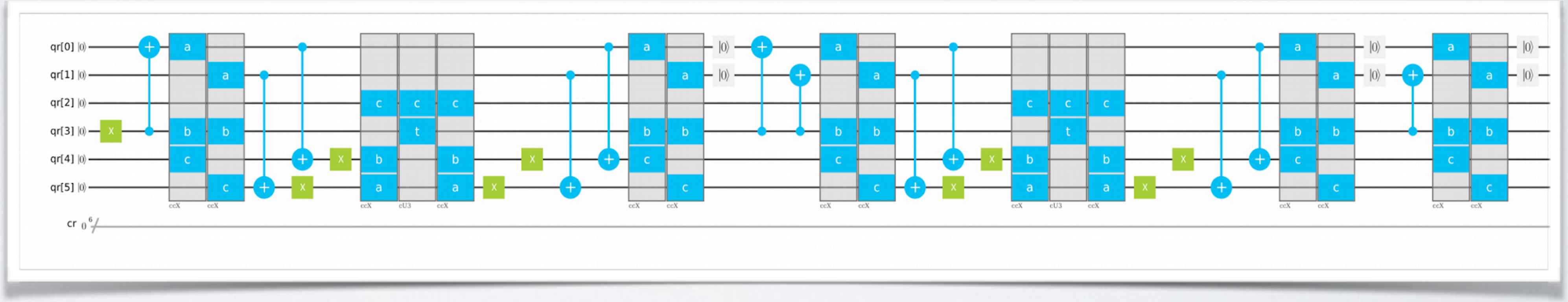
arXiv:1902.00498

Slide credit: I. Shapoval

QUAM DEMONSTRATION ON IBM-Q

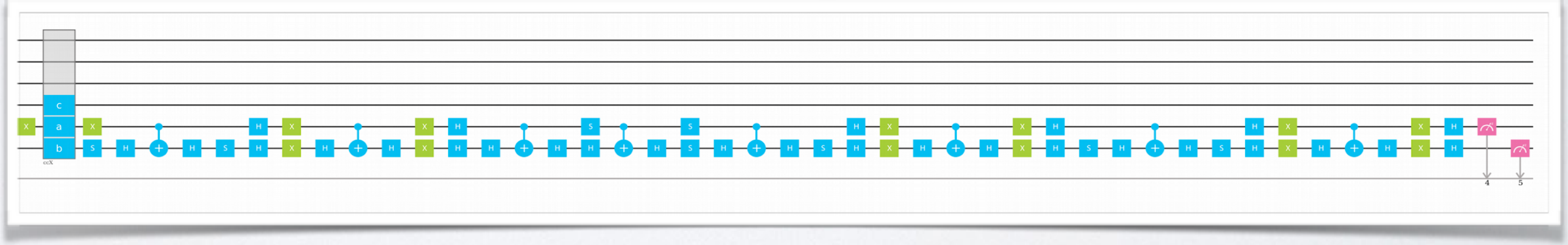
QuAM storage circuit generator

Ex.: complete circuit for retrieving one 2-bit pattern



QuAM retrieval circuit generator

Ex.: complete circuit for retrieving one 2-bit pattern

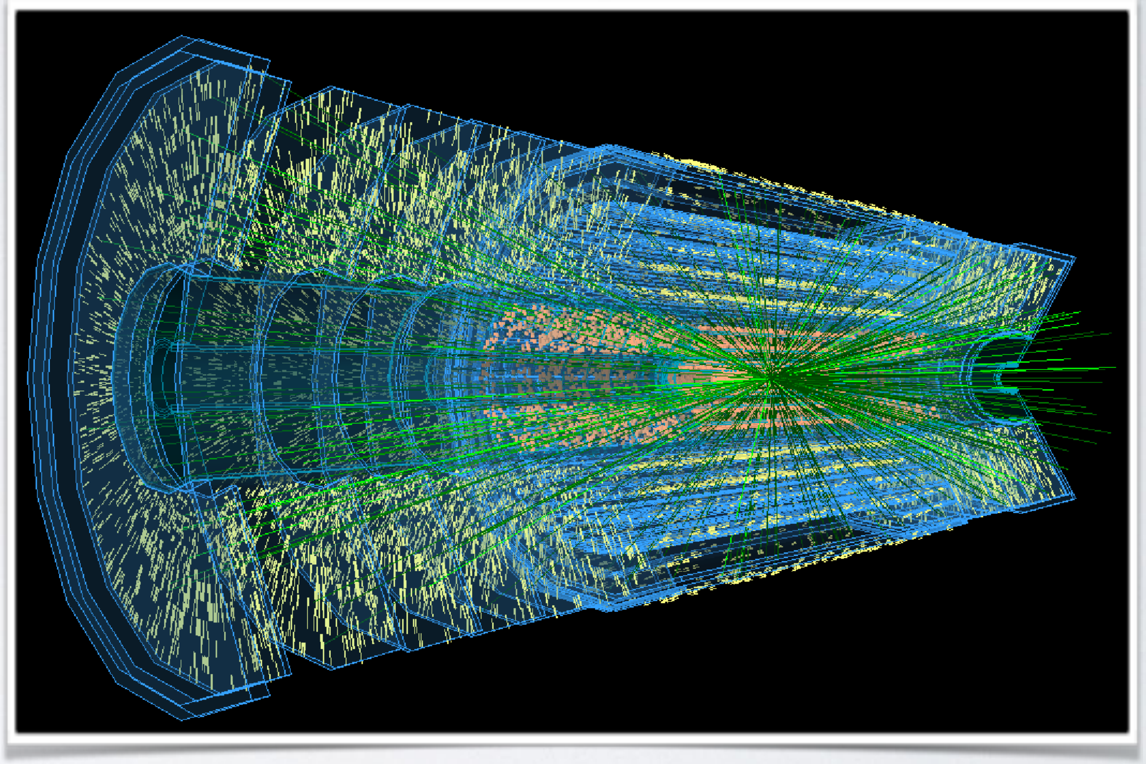


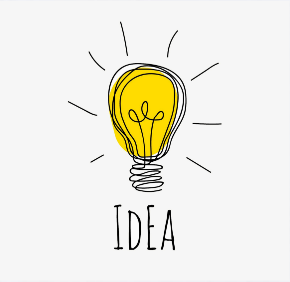
arXiv:1902.00498

Slide credit: I. Shapoval

DIFFERENT ALGORITHMS: QUANTUM ANNEALING

- Reformulate track reconstruction as an energy minimisation problem
  - Solve using the D-Wave quantum annealer
- Solution time won't scale with number of tracks

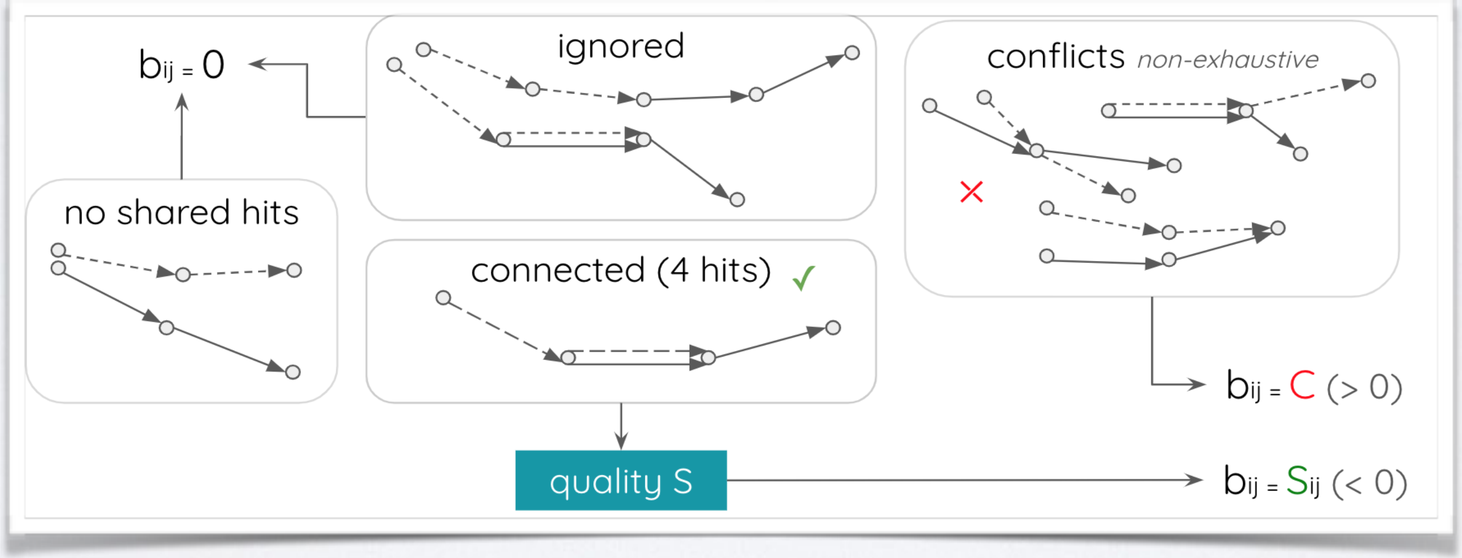




arXiv:1902.08324

QUANTUM ANNEALING

- Implement QUBO minimisation on D-Wave and study scaling with track multiplicity
- Inspired from *, but use triplets (3 hits) as the qubits
- Encode the quality of the triplets based on physics properties. Pair-wise connections b act as constraints (>0) or incentives (<0):



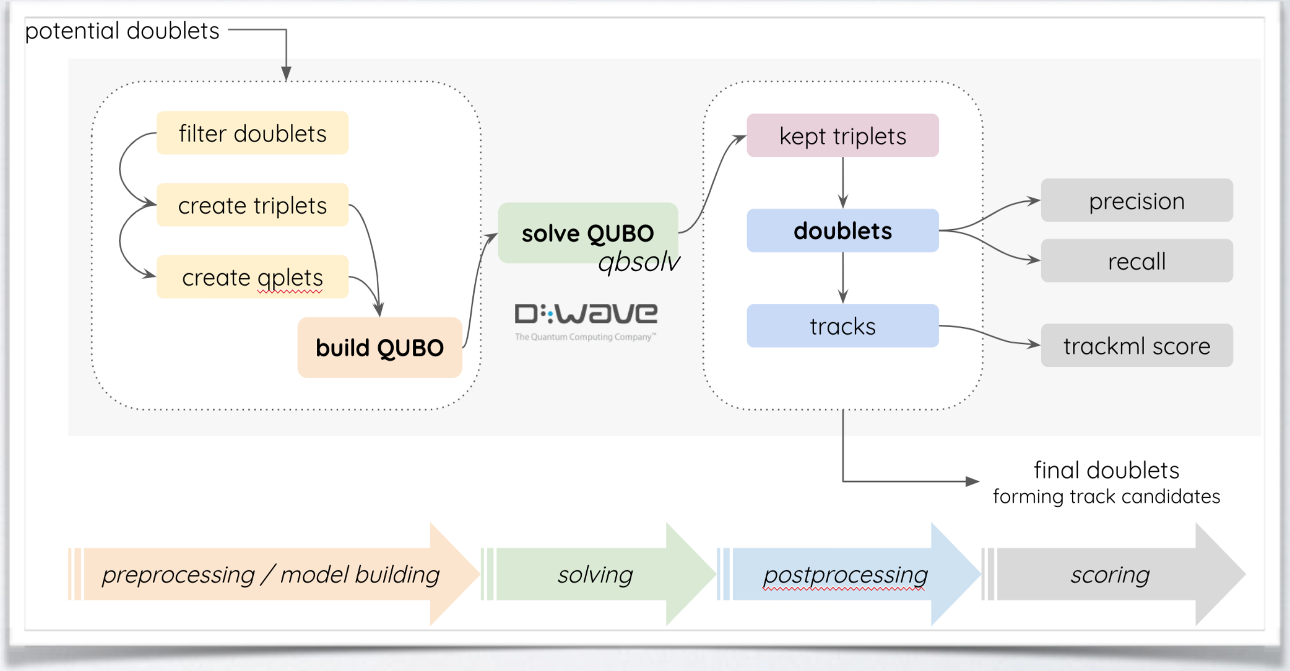
- Minimizing O means selecting the best triplets to form track candidates.

*Stimpfl-Abele & Garrido, Fast track finding with neural networks

Slide credit: L. Linder

IMPLEMENTATION

- Dataset: simplified HL-LHC-like* dataset (focus on barrel, 1+ GeV, 5+ hits)
    - Toy dataset, but representative of expected conditions at the HL-LHC
- QUBO solvers: qbsolv (D-Wave + simulation), neal (simulation)
- D-Wave 2X (1152 qubits), D-Wave 2000Q (2048 qubits)

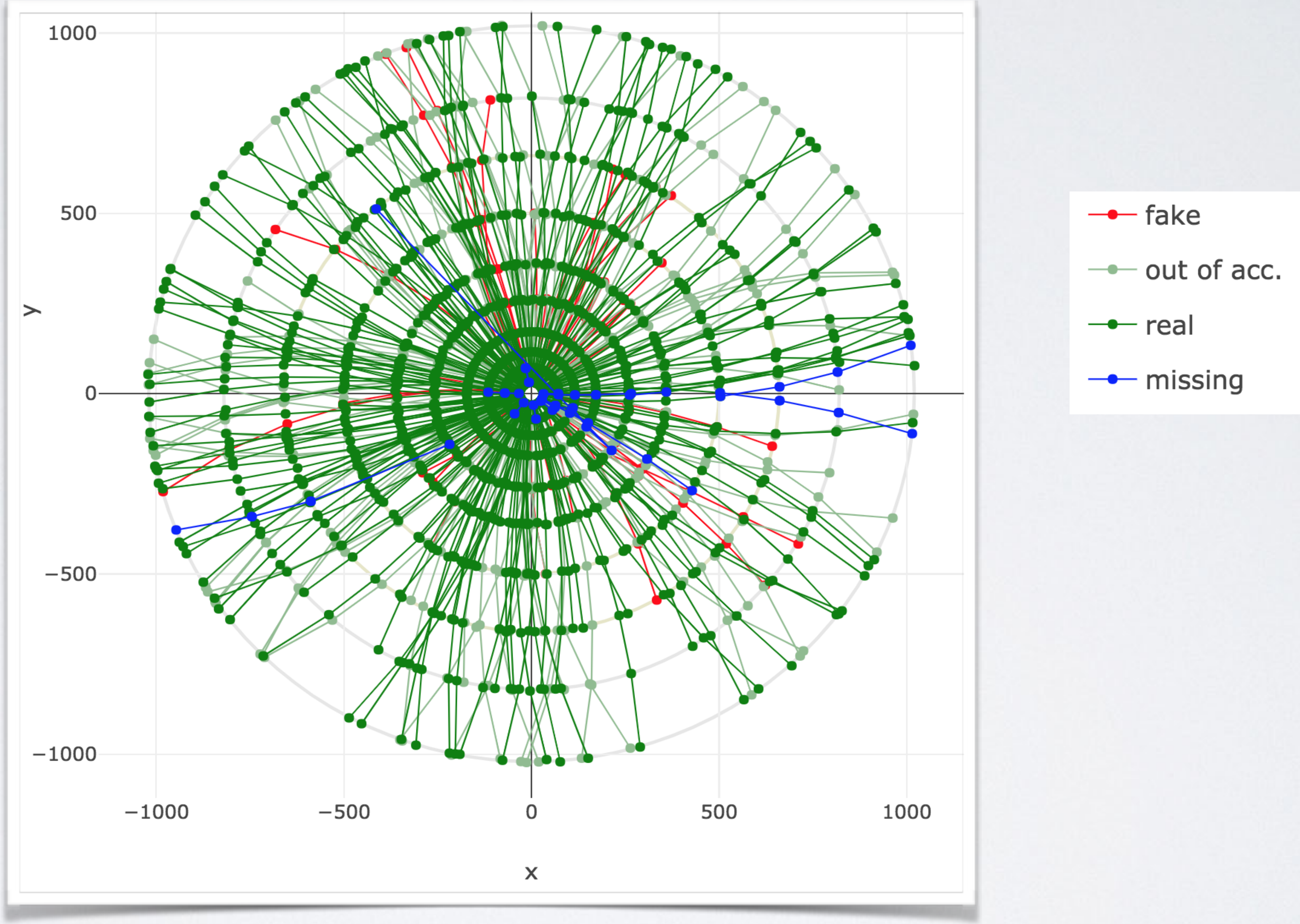


***trackml**

Slide credit: L. Linder

PERFORMANCE

Doublets for a dataset of 2456 particles and 16855 hits



arXiv:1902.08324

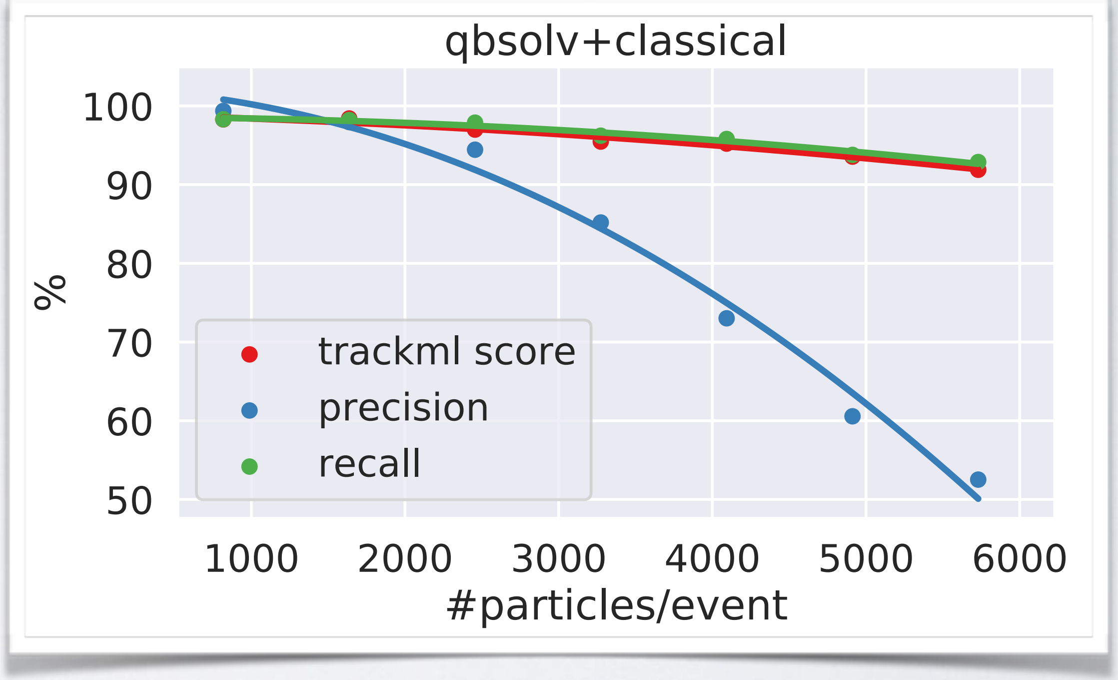
Slide credit: L. Linder

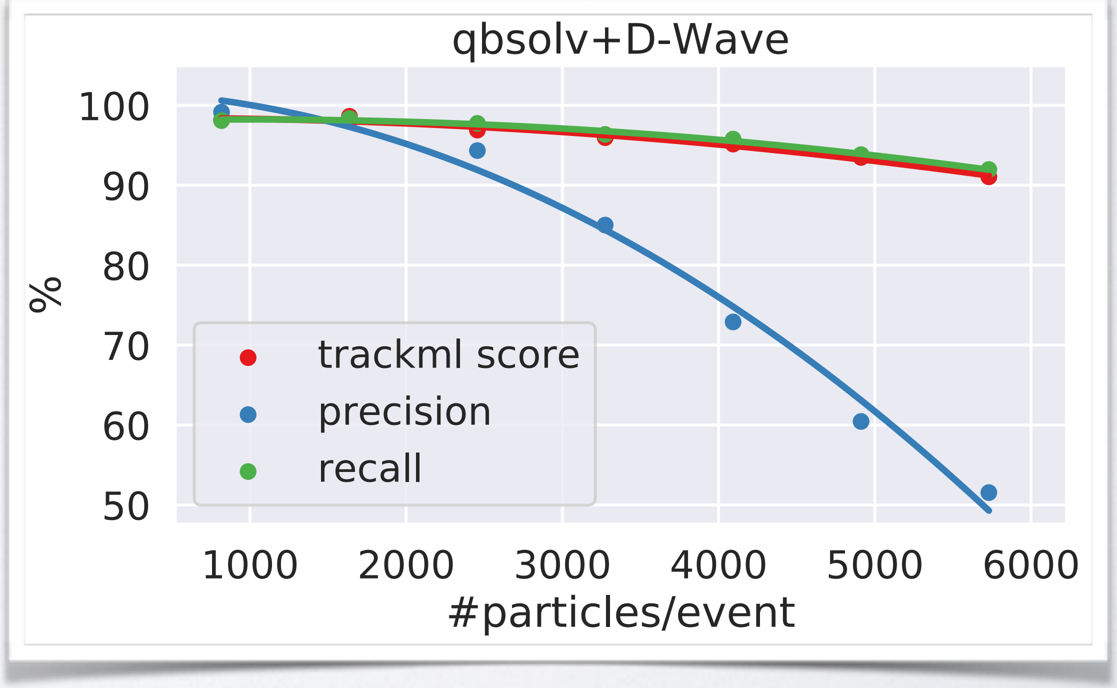
PERFORMANCE (2)

Physics performance as a function of occupancy using a D-Wave 2X (qbsolv).

Timing building: 0-20 min | solving: 0-12s (sim), 0-56 min (D-Wave)

D-Wave | sim. Same physics, important time overhead with D-Wave

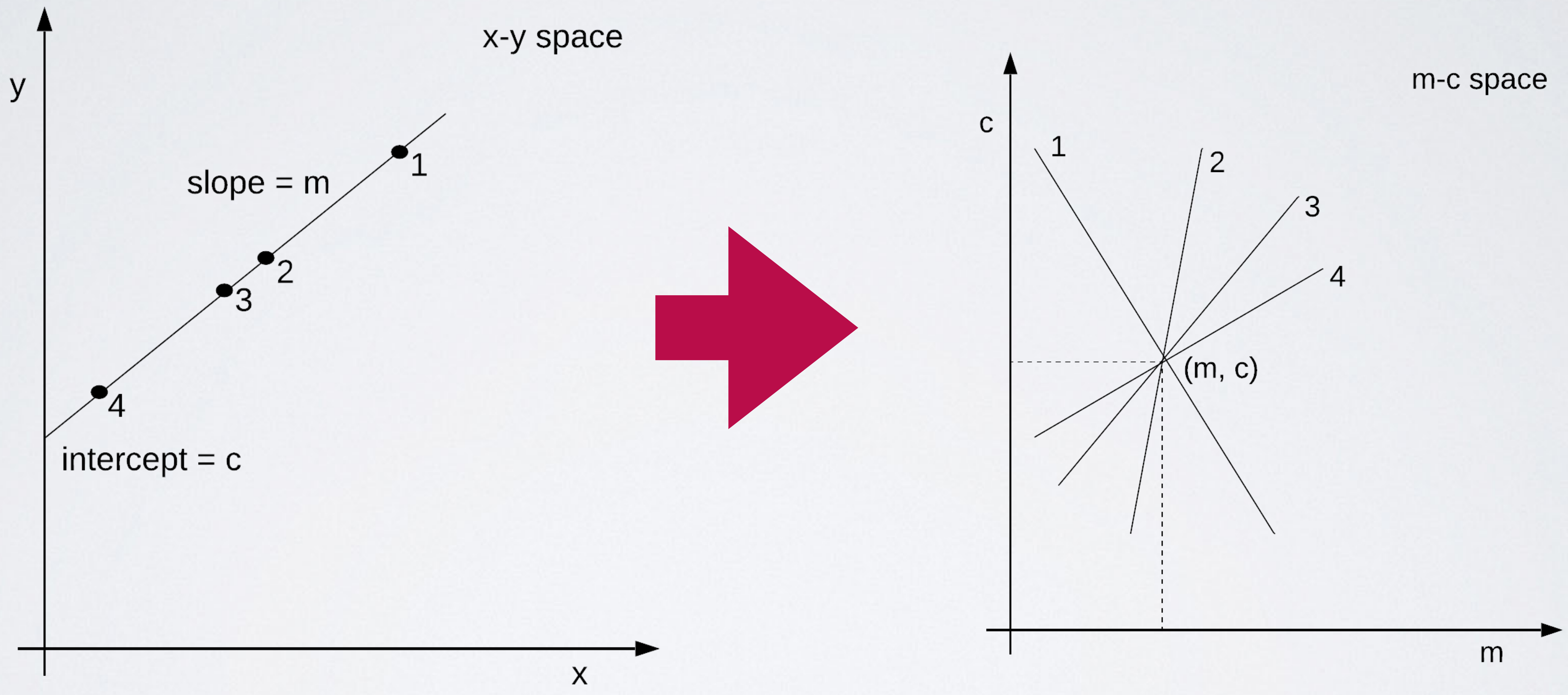




arXiv:1902.08324

Slide credit: L. Linder

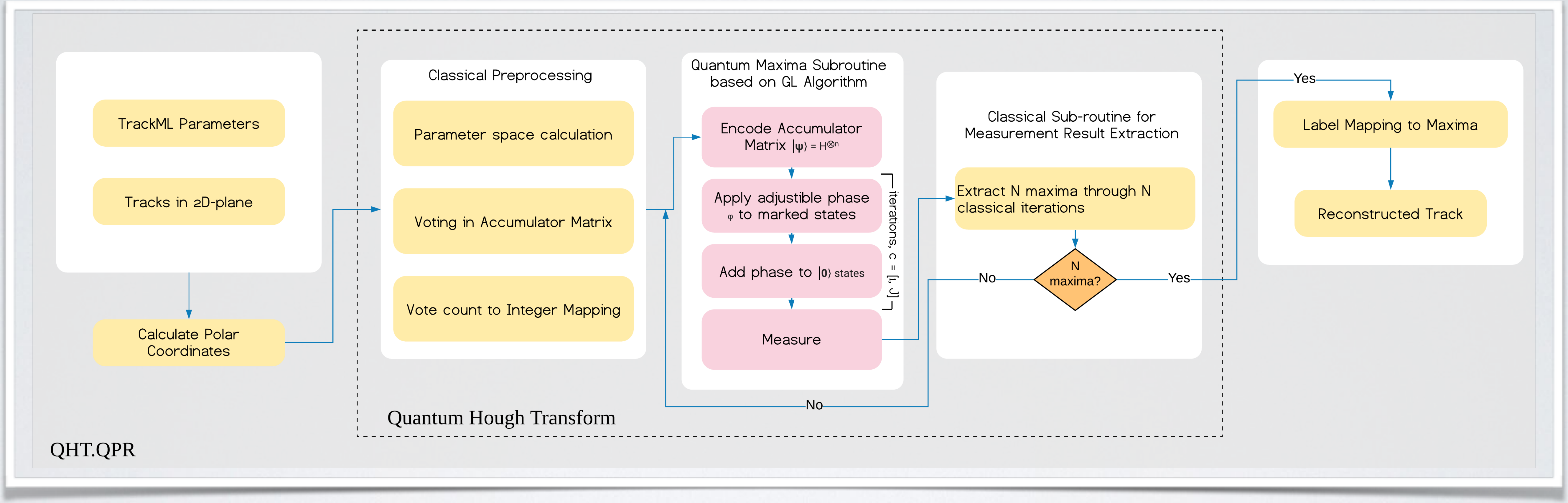
DIFFERENT ALGORITHMS: QUANTUM HOUGH TRANSFORM



P.V.C. Hough (1962), R.O. Dude, P.E. Hart (1972), D.H. Ballard (1980)

Slide Credit: A. Yadav

IMPLEMENTATION



TrackML Parameters

Tracks in 2D-plane

Calculate Polar Coordinates

Quantum Hough Transform

Classical Preprocessing

Parameter space calculation

Voting in Accumulator Matrix

Vote count to Integer Mapping

Quantum Maxima Subroutine based on GL Algorithm

Encode Accumulator Matrix $|\psi\rangle = H^{\otimes n}$

Apply adjustible phase φ to marked states

Add phase to $|0\rangle$ states

Measure

iterations, $c = [1, J]$

Classical Sub-routine for Measurement Result Extraction

Extract N maxima through N classical iterations

N maxima?

Label Mapping to Maxima

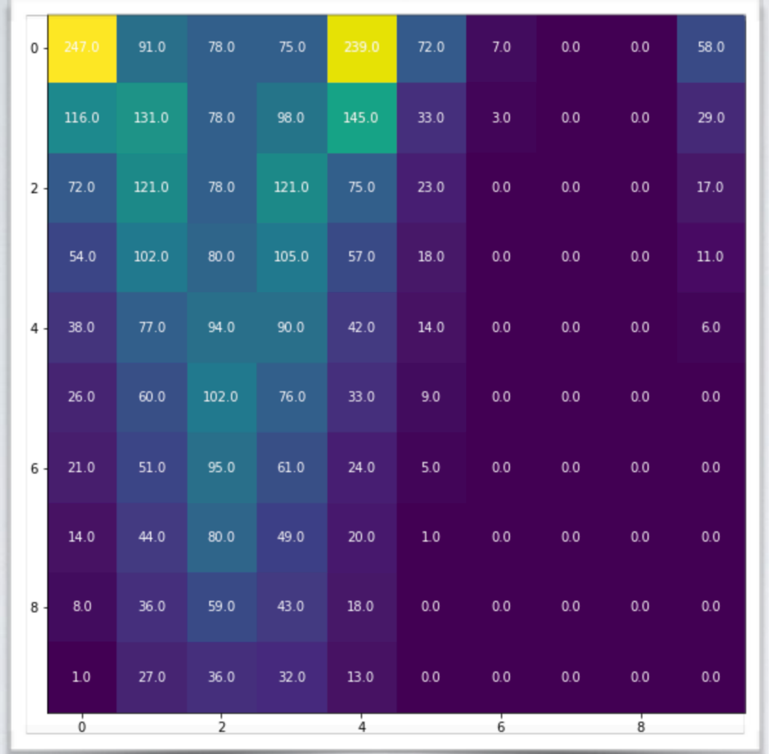
Reconstructed Track

QHT.QPR

An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits.

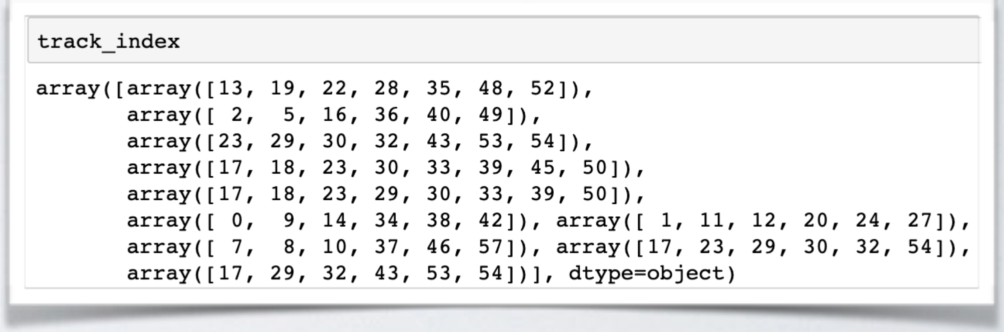
Slide Credit: A. Yadav

RESULTS: QUANTUM MAXIMA FINDING



247.0	91.0	78.0	75.0	239.0	72.0	7.0	0.0	0.0	58.0
116.0	131.0	78.0	98.0	145.0	33.0	3.0	0.0	0.0	29.0
72.0	121.0	78.0	121.0	75.0	23.0	0.0	0.0	0.0	17.0
54.0	102.0	80.0	105.0	57.0	18.0	0.0	0.0	0.0	11.0
38.0	77.0	94.0	90.0	42.0	14.0	0.0	0.0	0.0	6.0
26.0	60.0	102.0	76.0	33.0	9.0	0.0	0.0	0.0	0.0
21.0	51.0	95.0	61.0	24.0	5.0	0.0	0.0	0.0	0.0
14.0	44.0	80.0	49.0	20.0	1.0	0.0	0.0	0.0	0.0
8.0	36.0	59.0	43.0	18.0	0.0	0.0	0.0	0.0	0.0
1.0	27.0	36.0	32.0	13.0	0.0	0.0	0.0	0.0	0.0

Accumulator Space
for 8 tracks

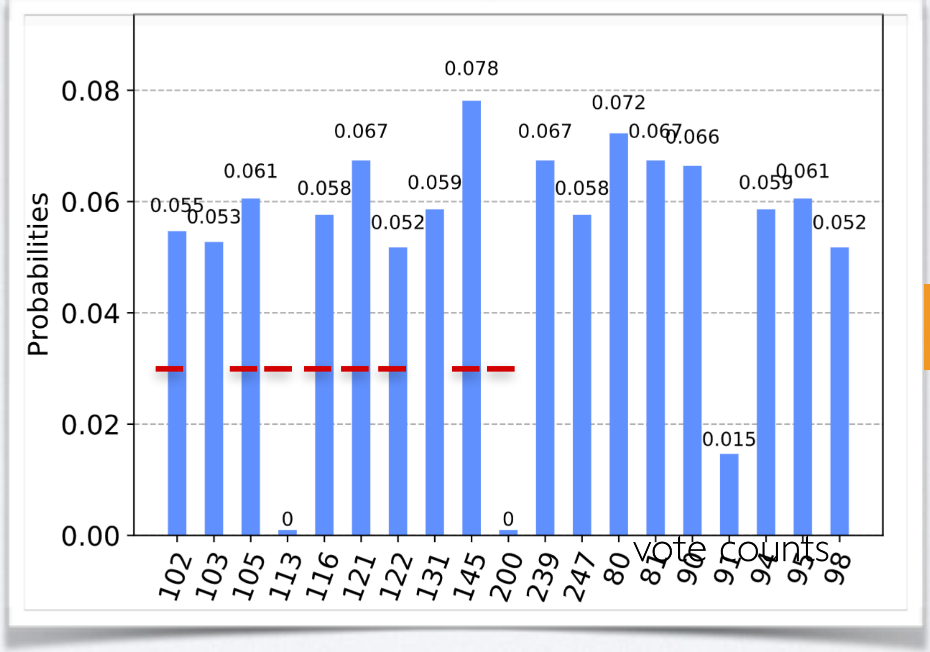


```
track_index

array([array([13, 19, 22, 28, 35, 48, 52]),
       array([ 2,  5, 16, 36, 40, 49]),
       array([23, 29, 30, 32, 43, 53, 54]),
       array([17, 18, 23, 30, 33, 39, 45, 50]),
       array([17, 18, 23, 29, 30, 33, 39, 50]),
       array([ 0,  9, 14, 34, 38, 42]), array([ 1, 11, 12, 20, 24, 27]),
       array([ 7,  8, 10, 37, 46, 57]), array([17, 23, 29, 30, 32, 54]),
       array([17, 29, 32, 43, 53, 54])], dtype=object)
```

8 maxima values

Quantum
Superposition



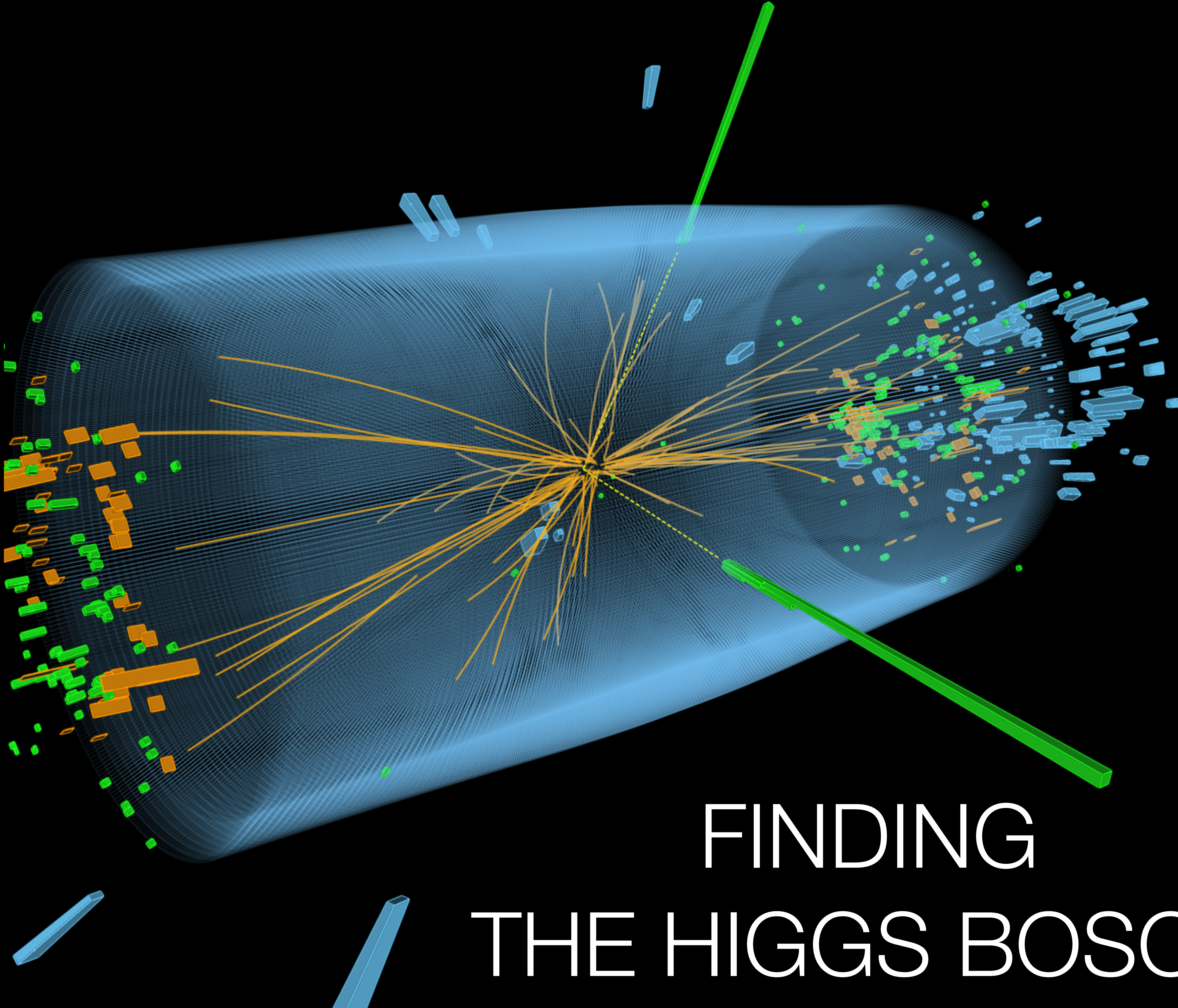
8-qubit simulation
#tracks = 4, #bins up to $2^8 = 256$, time = 58.6s

Local Maxima Detection using Grover-Long Algorithm

An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits.

Slide Credit: A. Yadav

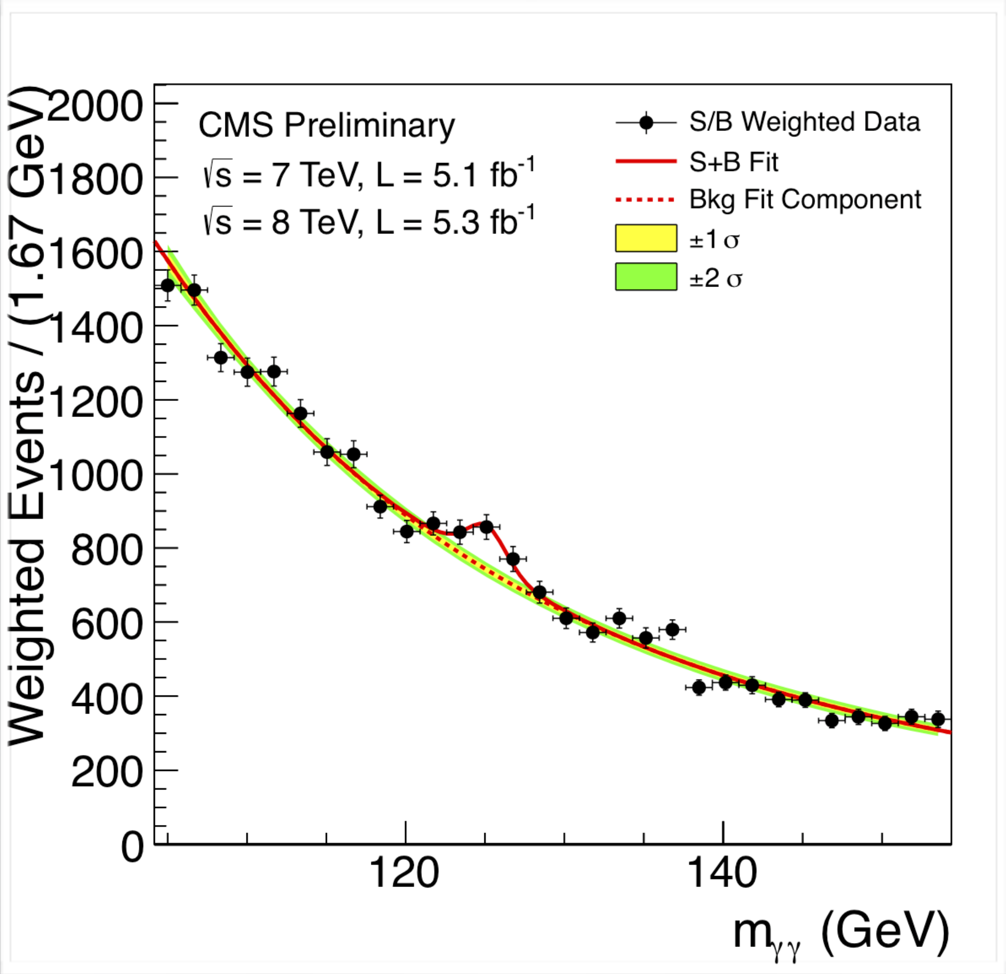
ANALYSIS



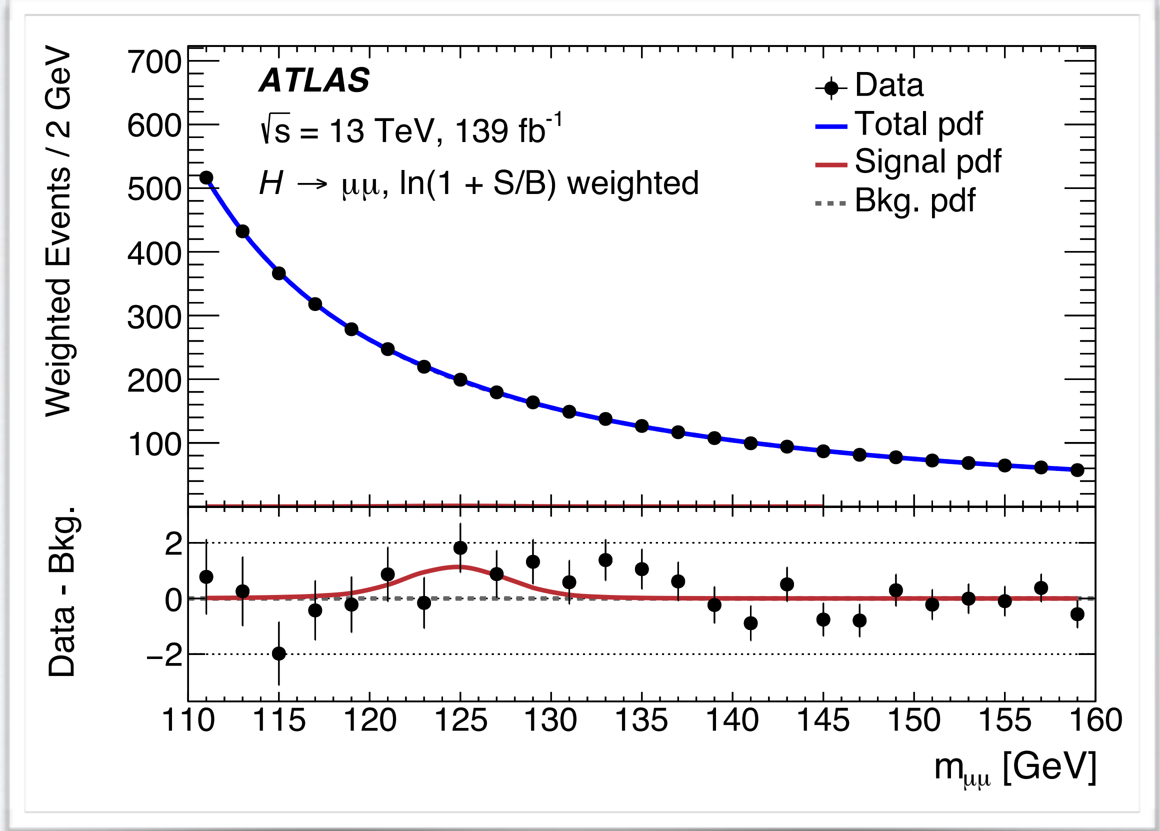
FINDING THE HIGGS BOSON

THE $H \to \gamma\gamma$ AND $H \to \mu\mu$ CHANNELS

Something old and something new



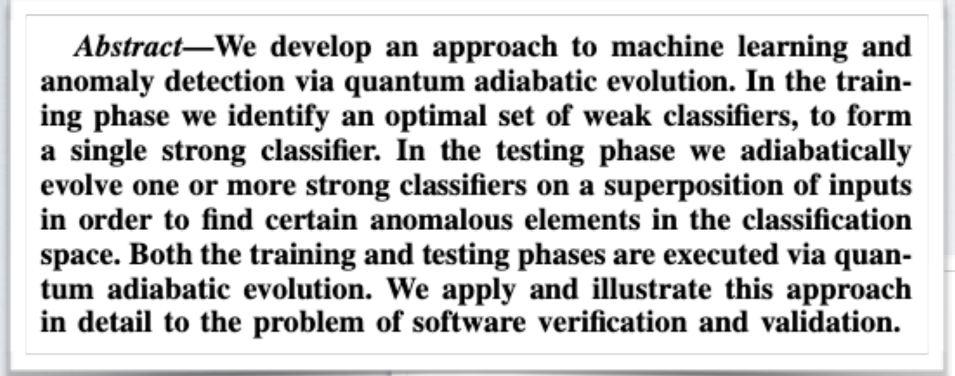
CMS, PLB 716, 30-61



ATLAS, arXiv:2007.07830

QAML CLASSIFIERS

(Quantum Adiabatic Machine Learning)



Abstract—We develop an approach to machine learning and anomaly detection via quantum adiabatic evolution. In the training phase we identify an optimal set of weak classifiers, to form a single strong classifier. In the testing phase we adiabatically evolve one or more strong classifiers on a superposition of inputs in order to find certain anomalous elements in the classification space. Both the training and testing phases are executed via quantum adiabatic evolution. We apply and illustrate this approach in detail to the problem of software verification and validation.

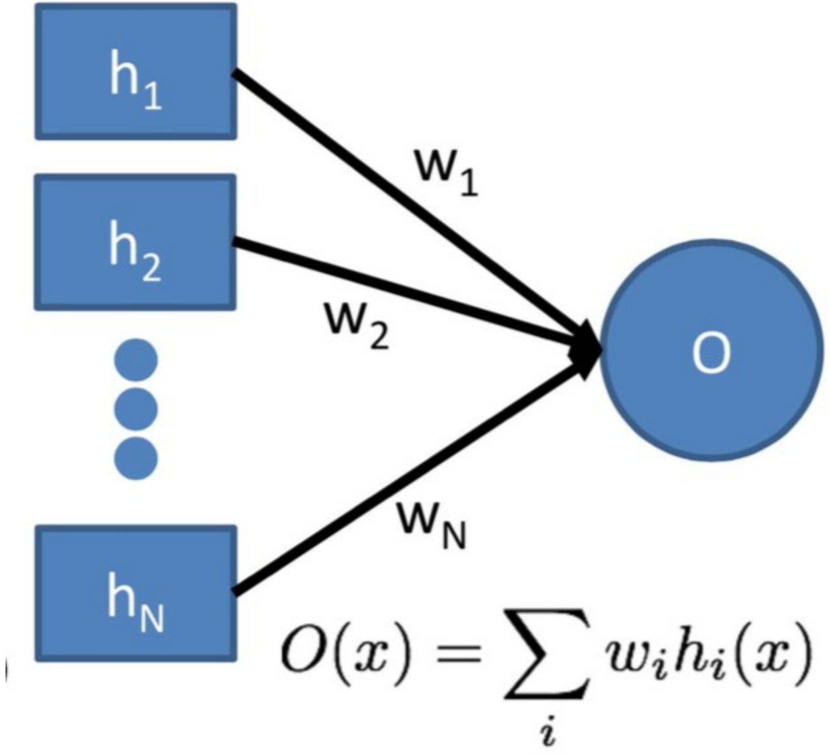
Pudenz and Lidar, arXiv:1109.0325

Define functions **h**$_i$ of the input variables into [-1,1] such that

- P(signal|h>0) > P(bkg|h>0)
- P(bkg|h<0) > P(signal|h<0)

i.e. Most signal on h>0, most bkg on h<0

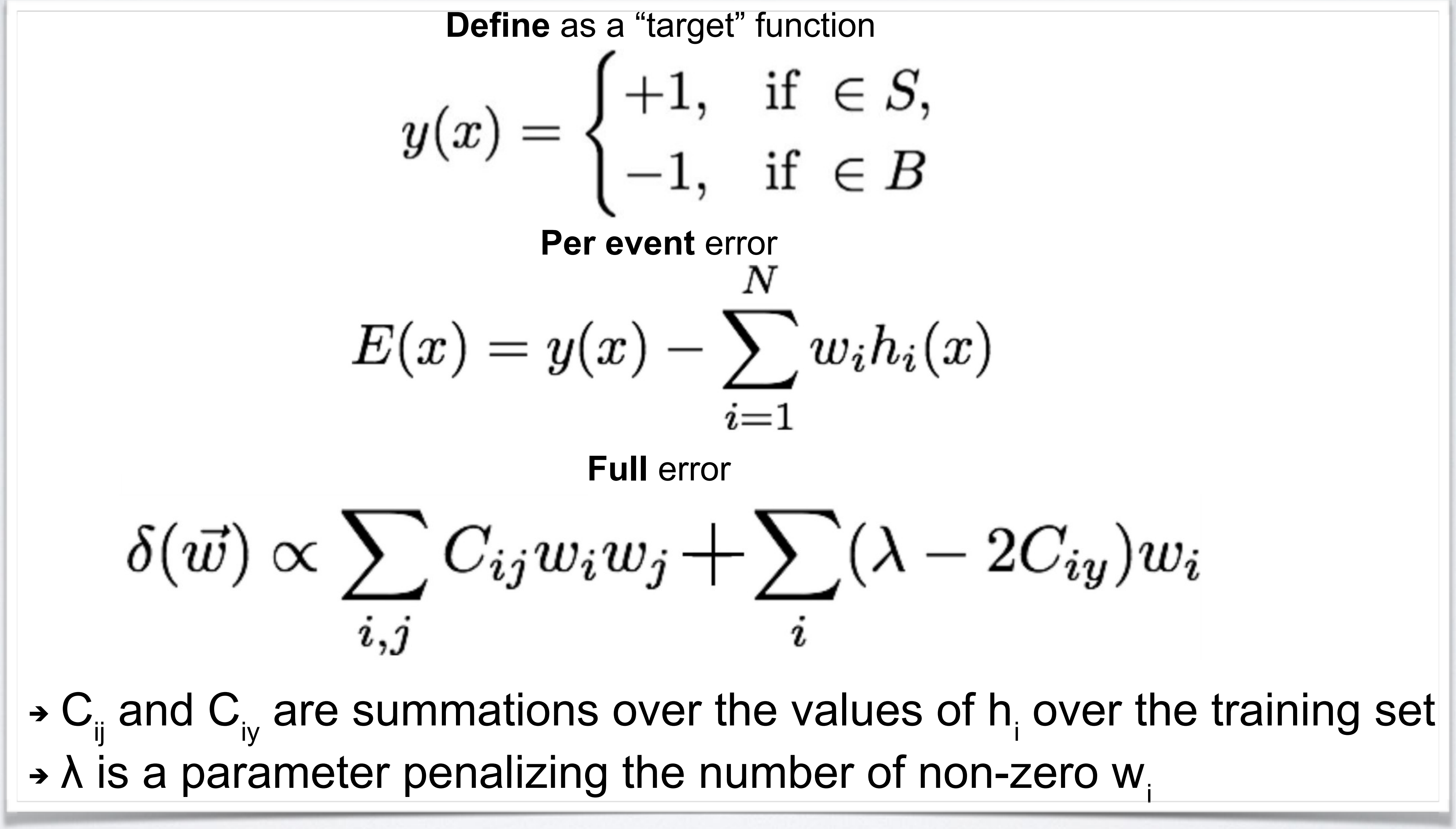
Define w$_i$ as binary linear combination of h$_i$



$$O(x) = \sum_i w_i h_i(x)$$

Slide credit: J. R. Vlimant

QAML OBJECTIVE FUNCTION



Define as a “target” function

$$y(x) = \begin{cases} +1, & \text{if } \in S, \\ -1, & \text{if } \in B \end{cases}$$

Per event error

$$E(x) = y(x) - \sum_{i=1}^{N} w_i h_i(x)$$

Full error

$$\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_{i} (\lambda - 2C_{iy}) w_i$$

- → C_{ij} and C_{iy} are summations over the values of h_i over the training set
- → λ is a parameter penalizing the number of non-zero w_i

Pudenz and Lidar, arXiv:1109.0325

Slide credit: J. R. Vlimant

IMPLEMENTATION WITH QUBO

$$\delta(\vec{w}) \propto \sum_{i,j} C_{ij} w_i w_j + \sum_i (\lambda - 2C_{iy}) w_i$$

Simple conversion of binary weights to ±1

$$H_{\text{Ising}} = \sum_i h_i \sigma_i^z + \sum_{ij} J_{ij} \sigma_i^z \sigma_j^z$$

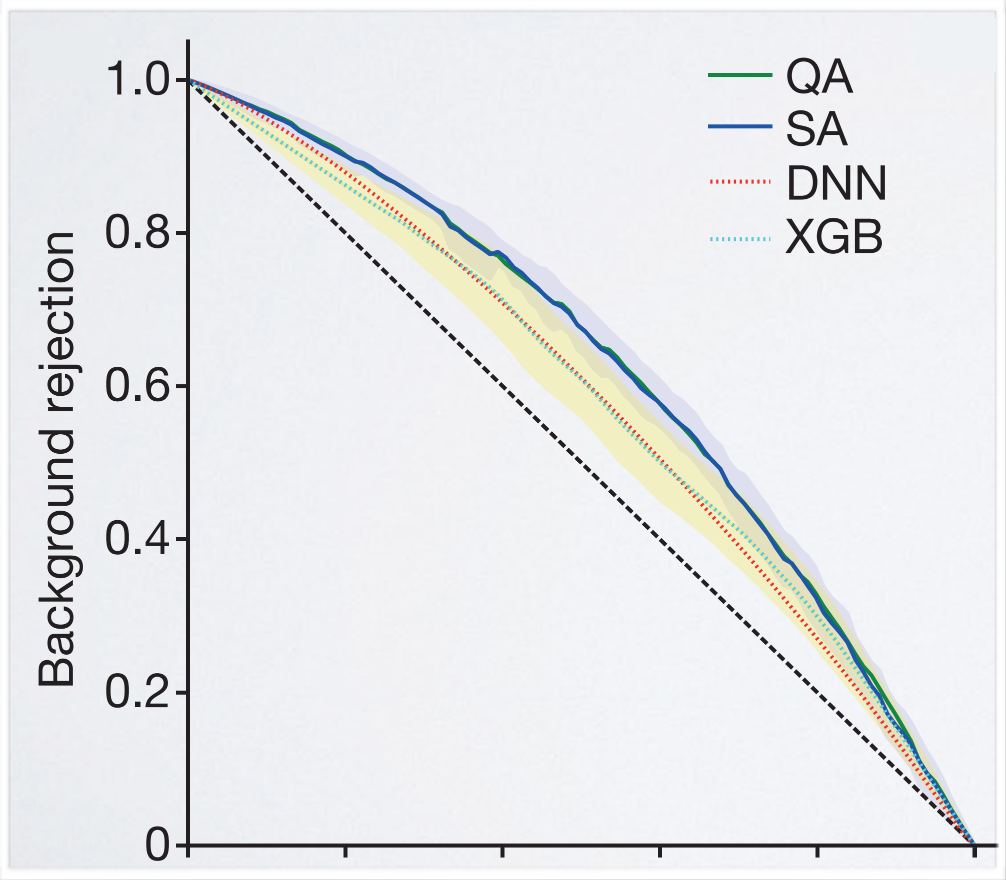
Slide credit: J. R. Vlimant

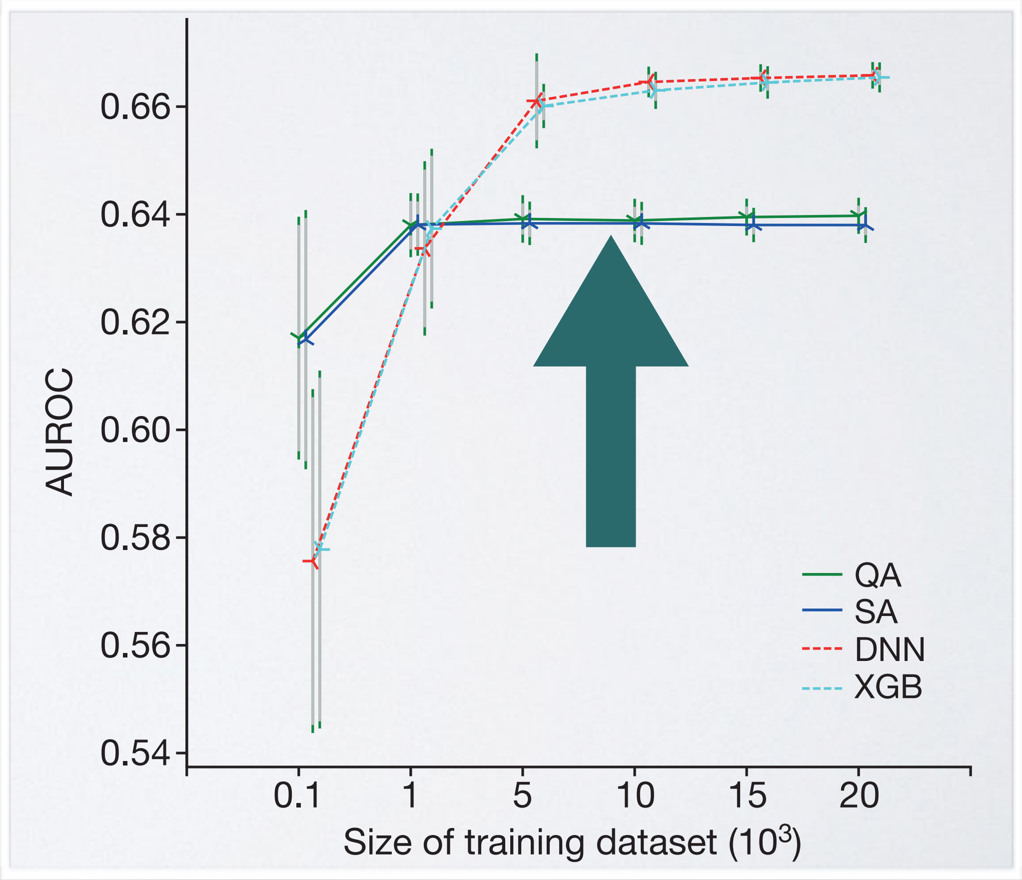
$H \to \gamma\gamma$ ON D-QWAVE

doi:10.1038/nature24047

Solving a Higgs optimization problem with quantum annealing for machine learning

Alex Mott[1]†*, Joshua Job[2,3]*, Jean-Roch Vlimant[1], Daniel Lidar[3,4] & Maria Spiropulu[1]

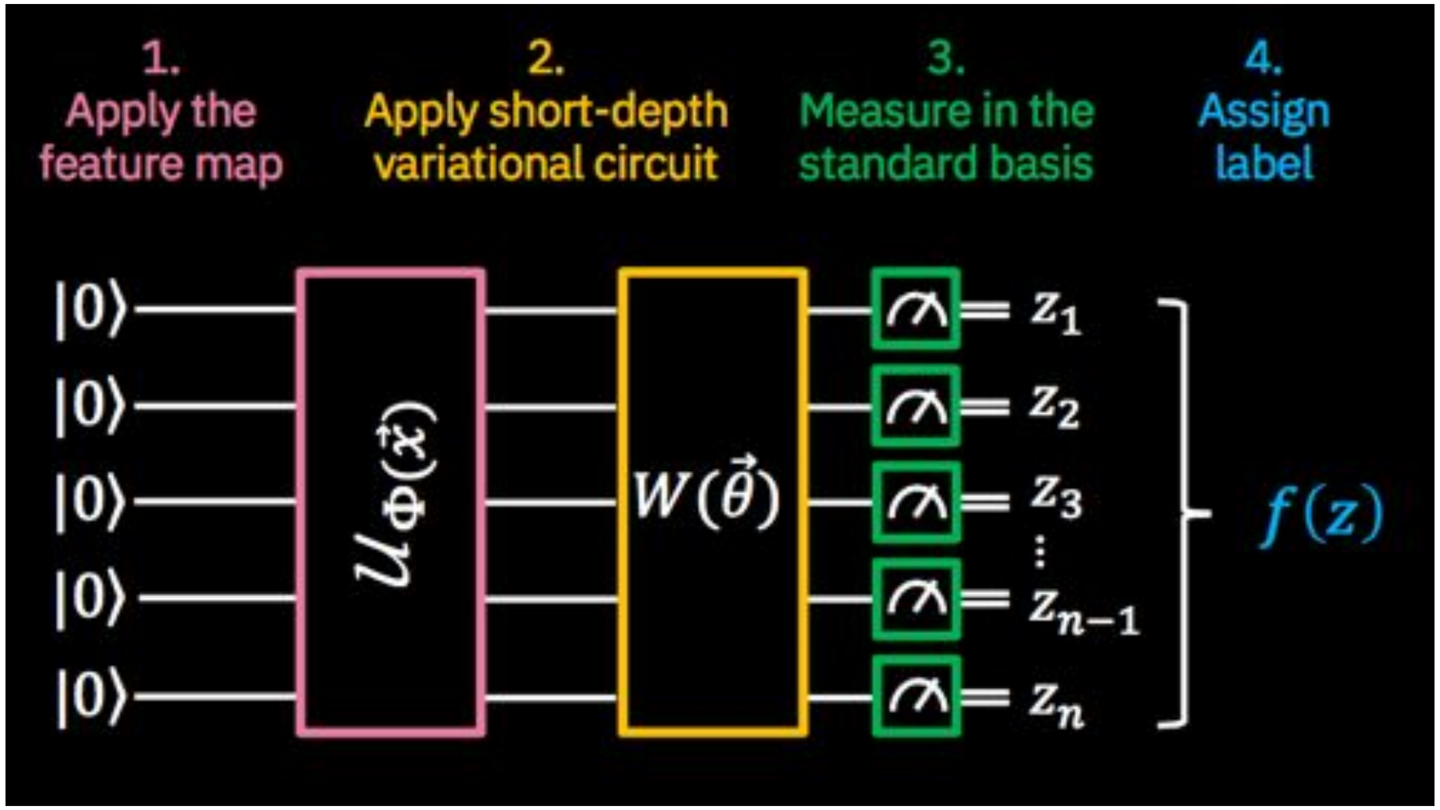




nature24047

VARIATIONAL SVM

- **In 2018, a variational Quantum SVM method was introduced by IBM, published in Nature 567 (2019) 209. The variational Quantum SVM method can be summarized in four steps:**

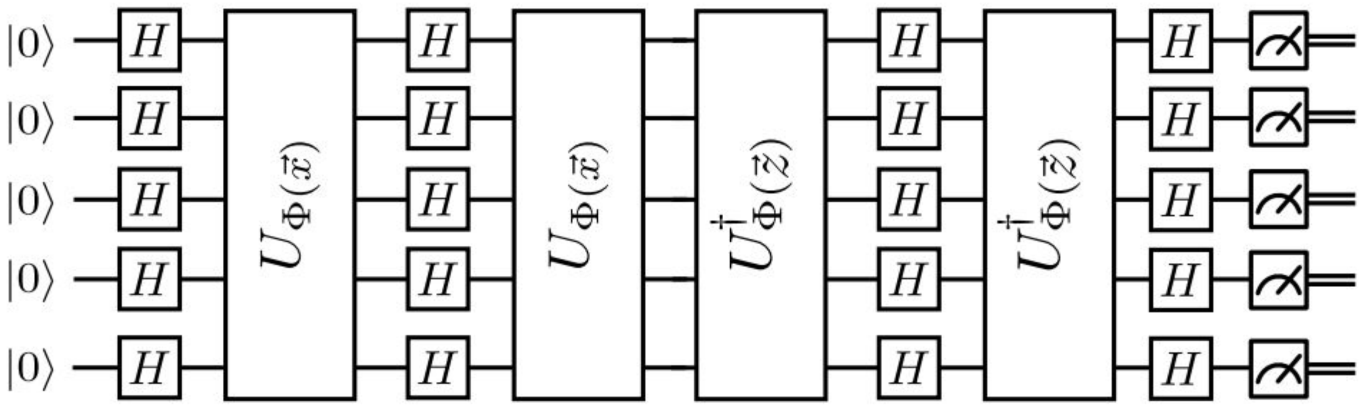


- **During the training phase, a set of events are used to train the circuit W(θ) to reproduce correct classification**

Slide Credit: S.L. Wu and C. Zhou

SVM KERNEL METHOD

- ***Kernel Trick for Classical SVM**: mapping the non-linear separable data into a higher dimensional feature space using a kernel function that measures the similarity between two data points; then using the kernel to find a separating hyperplane.*
- ***Quantum Kernel Estimation** (introduced by IBM, published in Nature 567 (2019) 209): mapping classical data $\vec{x}$ non-linearly to a quantum state using Quantum Feature Map function; calculating the kernel matrix $K(\vec{x}, \vec{z})=|\langle\Phi(\vec{x})|\Phi(\vec{z})\rangle|^2$ using a quantum computer; then training the quantum SVM in the same way as a classical SVM.*

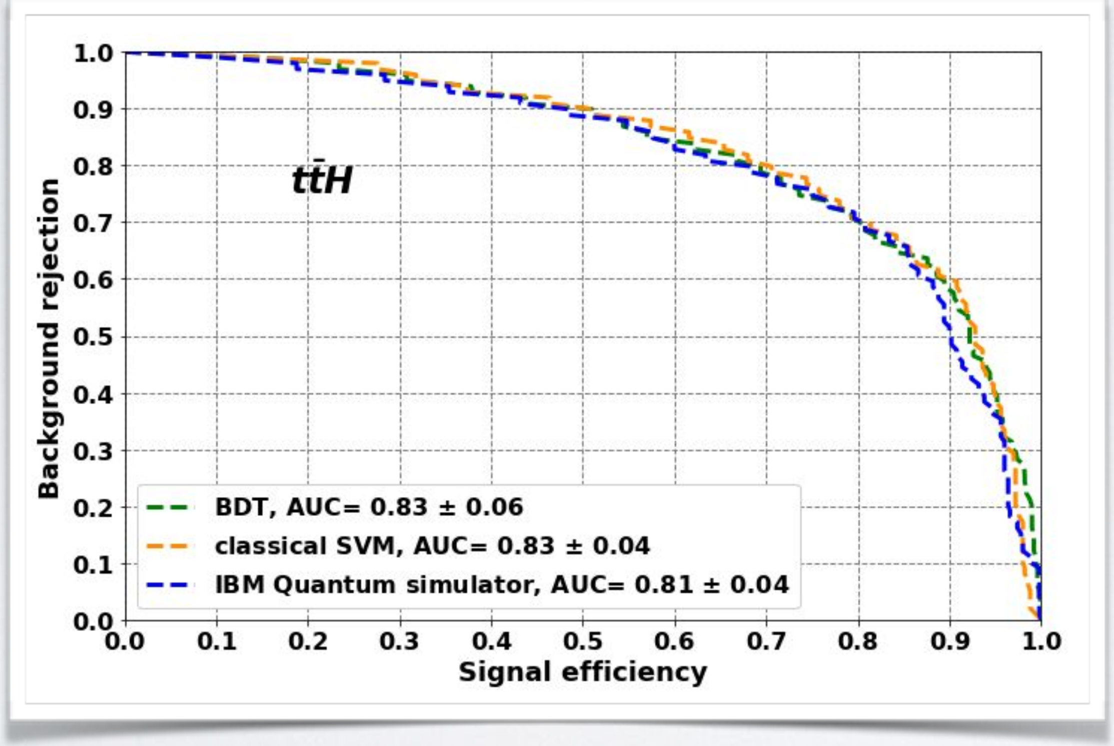


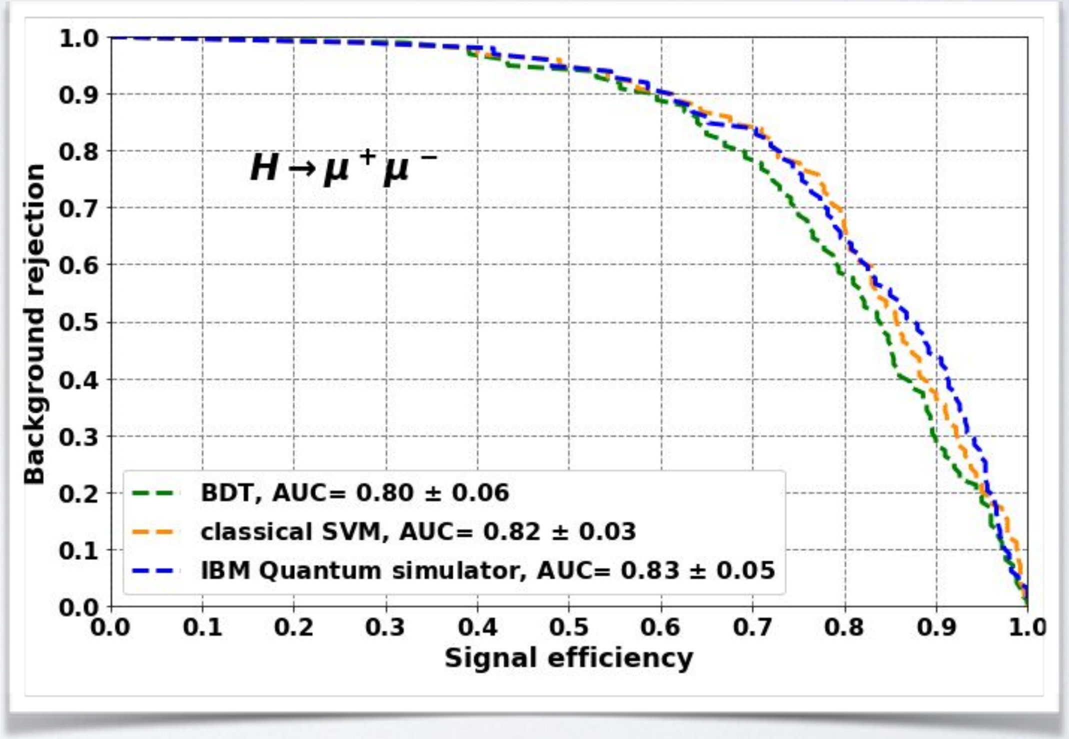
Nature 567 (2019) 209

Slide Credit: S.L. Wu and C. Zhou

SIMULATION RESULTS

Using the Qiskit Qasm simulator

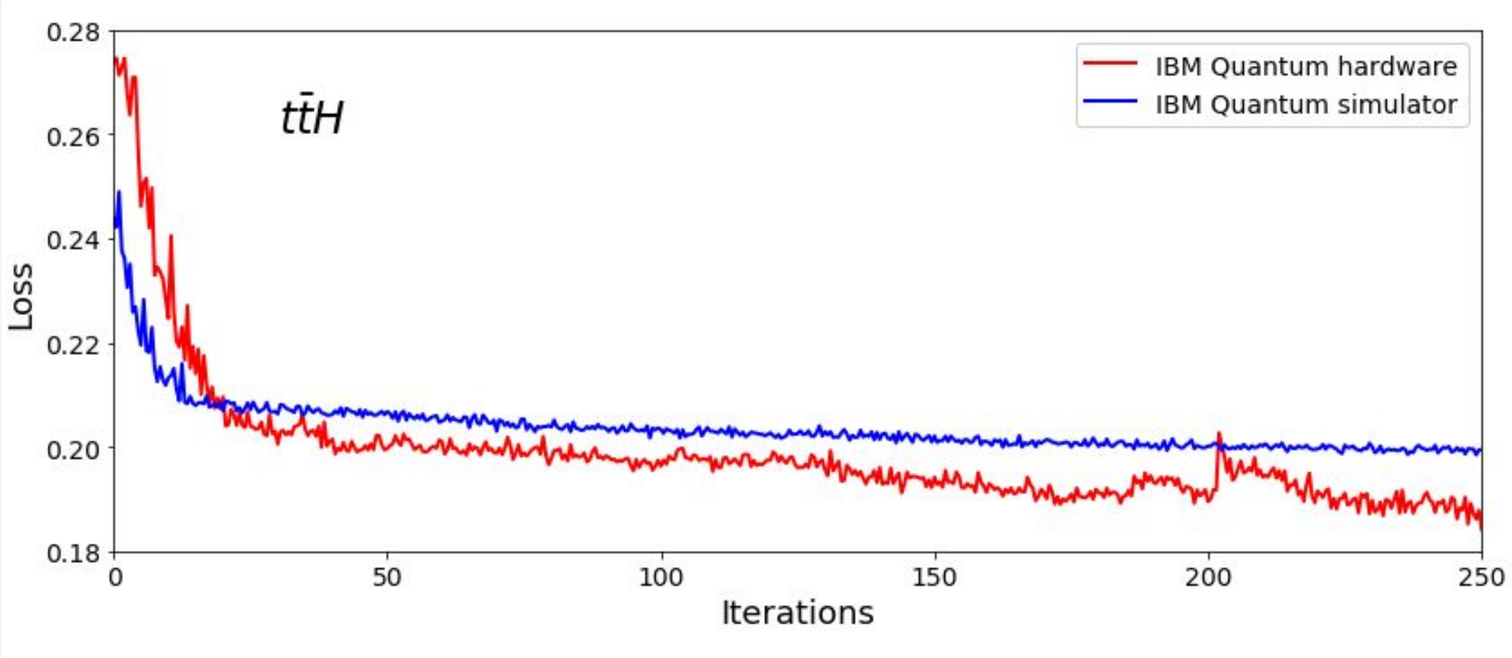




10 qubits, 100 training/testing events

Slide Credit: S.L. Wu and C. Zhou

RESULTS ON IBM HARDWARE

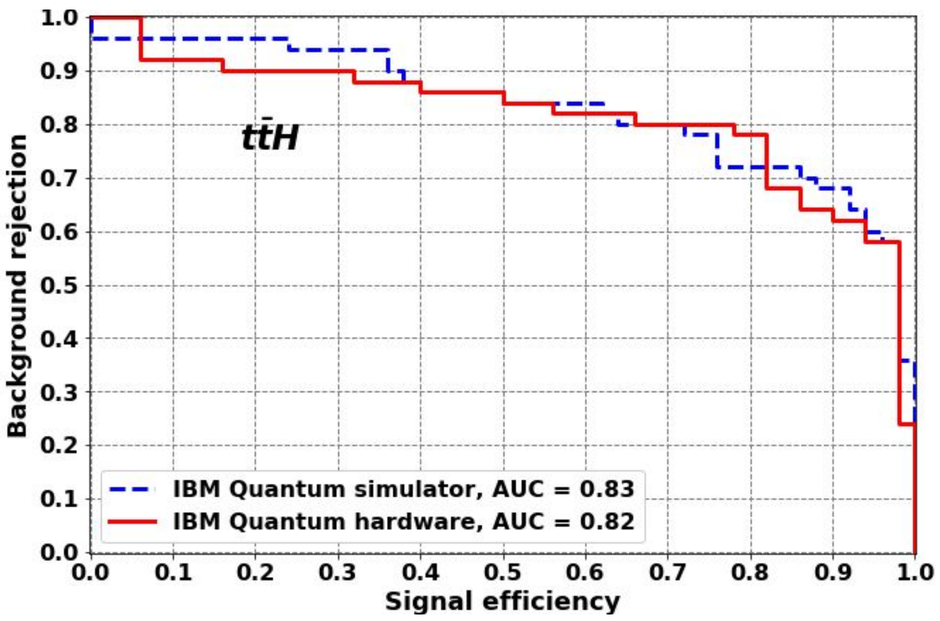


Blue: Quantum Simulation

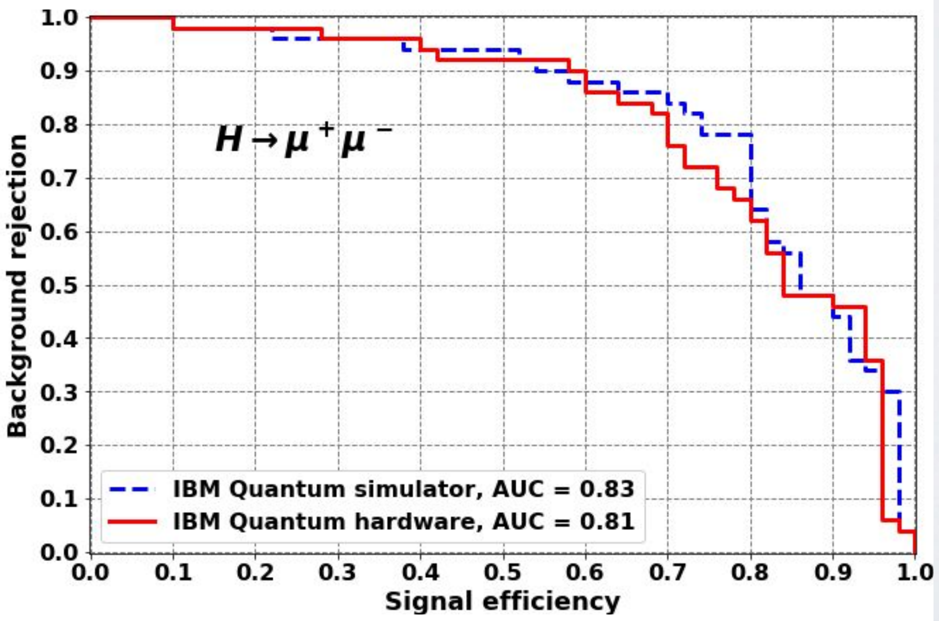
Red: Quantum Hardware

Loss: the mean of the squared differences between the output scores from the quantum algorithm and the ideal scores

ibmq_boelingen (20 qubits); ibmq_paris (27 qubits)



hardware AUC = 0.82, simulator AUC = 0.83



hardware AUC = 0.81, simulator AUC = 0.83

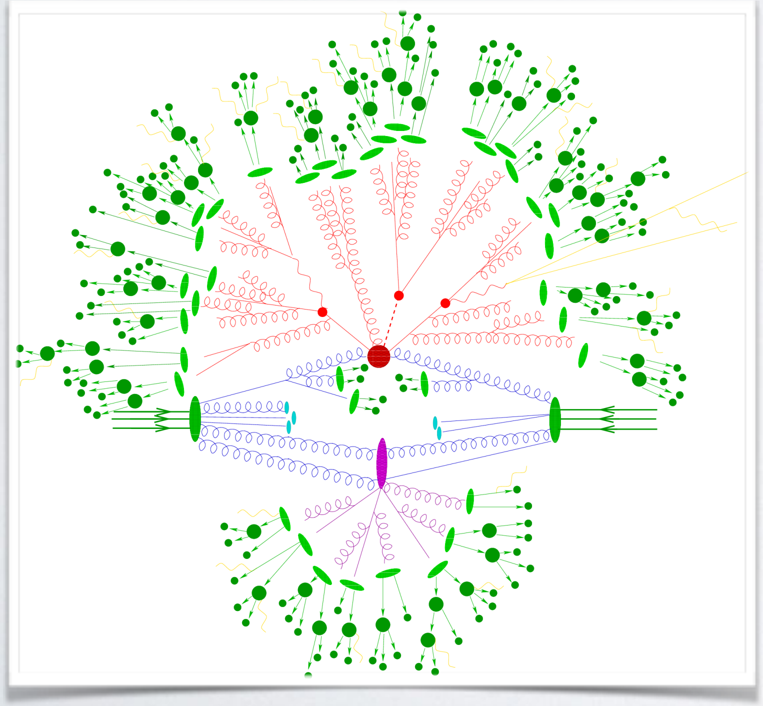
Slide Credit: S.L. Wu and C. Zhou

SUMMARY

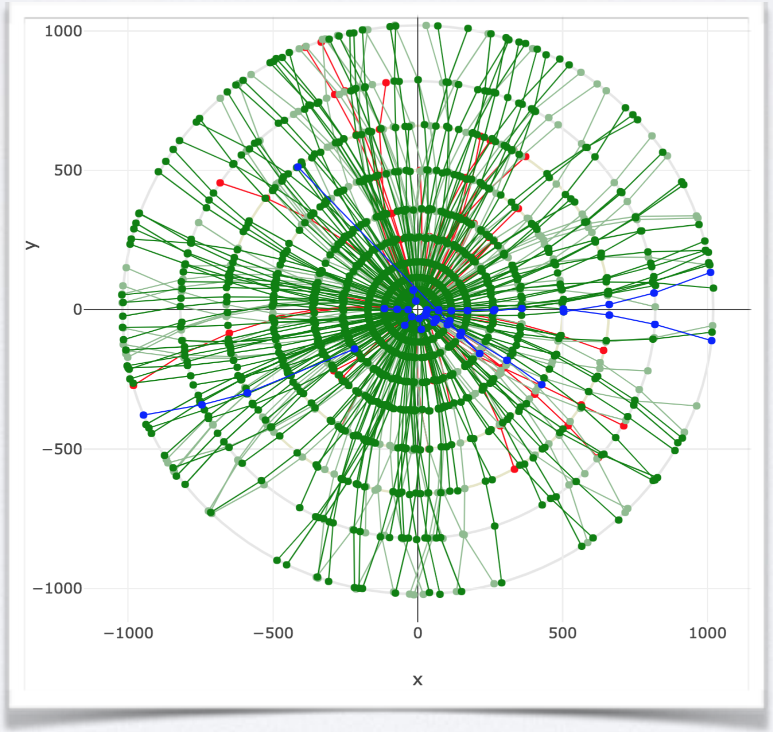
Exciting recent developments in quantum computing

How might they be useful for particle physics?

Simulation



Track Reconstruction



Analysis