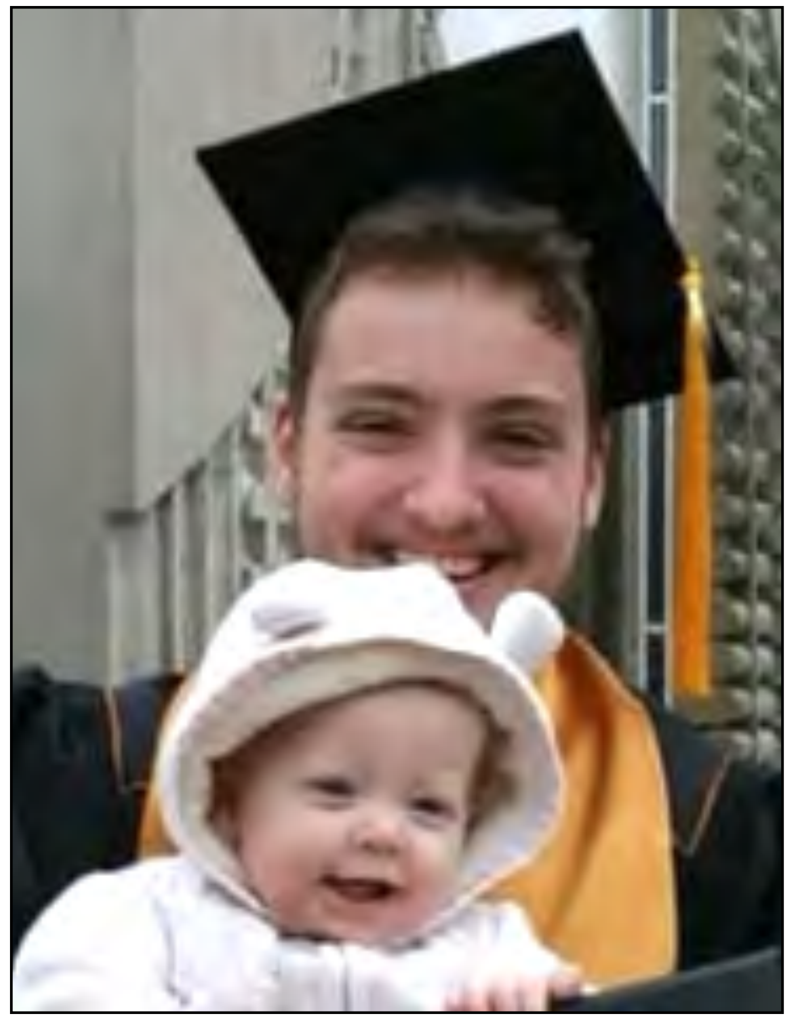
Reconstructing Proton-Proton Collision Positions at the Large Hadron Collider with D-Wave




Andrew J. Wildridge **Sachin B. Vaidya** **Andreas Jung** **Souvik Das**

PURDUE
UNIVERSITY
DEPARTMENT OF PHYSICS AND ASTRONOMY

Newport, RI
September 26, 2018

Outline

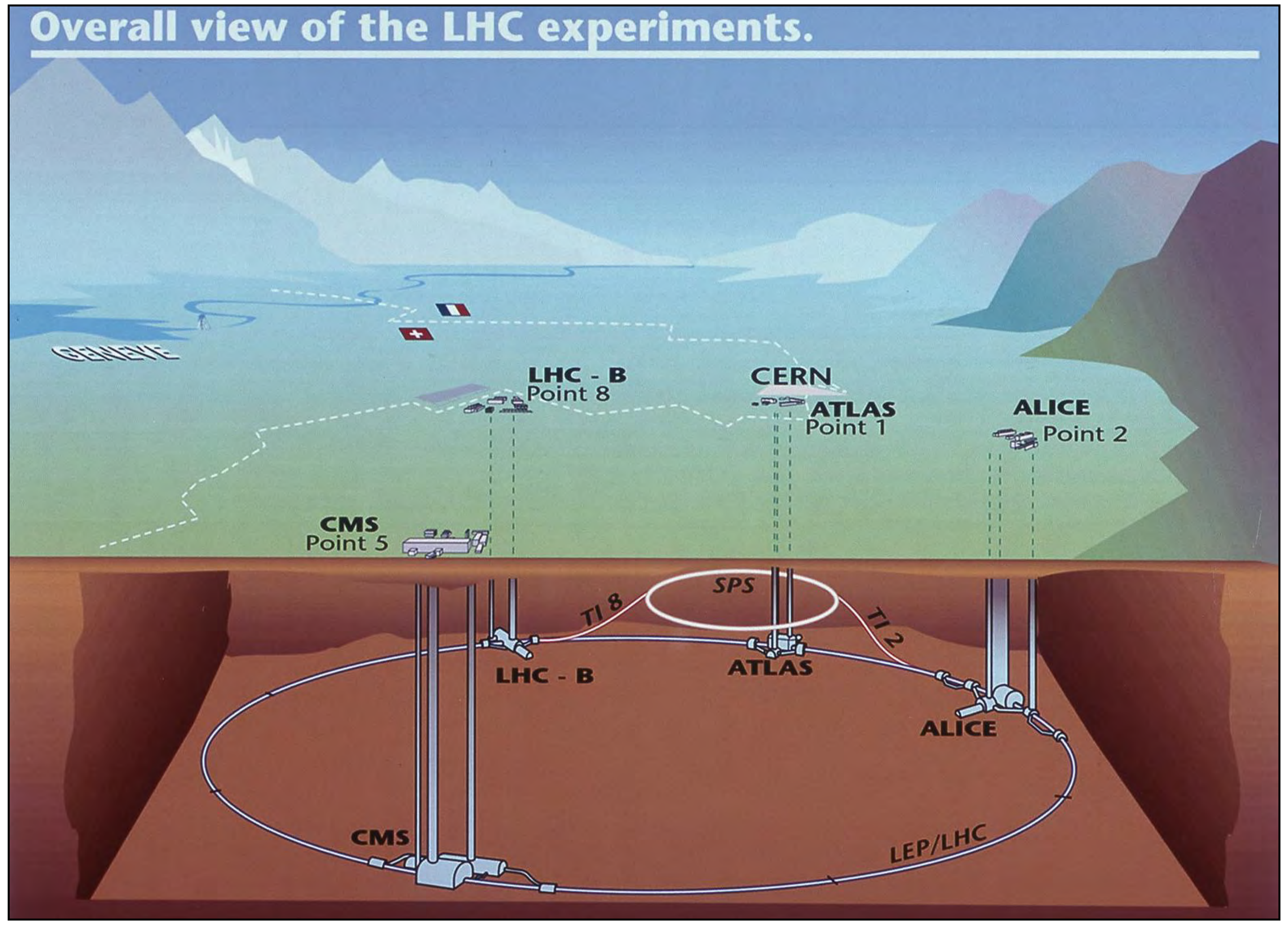
- Introduction to the Large Hadron Collider and the Compact Muon Solenoid detector
- QUBO formulation of the problem for D-Wave
- D-Wave performance, benchmarked against CPU
- Conclusions and outlook

The Large Hadron Collider

The Large Hadron Collider

- The Higgs boson
- Large extra-dimensions
- Supersymmetry
- Dark matter
- Baryogenesis

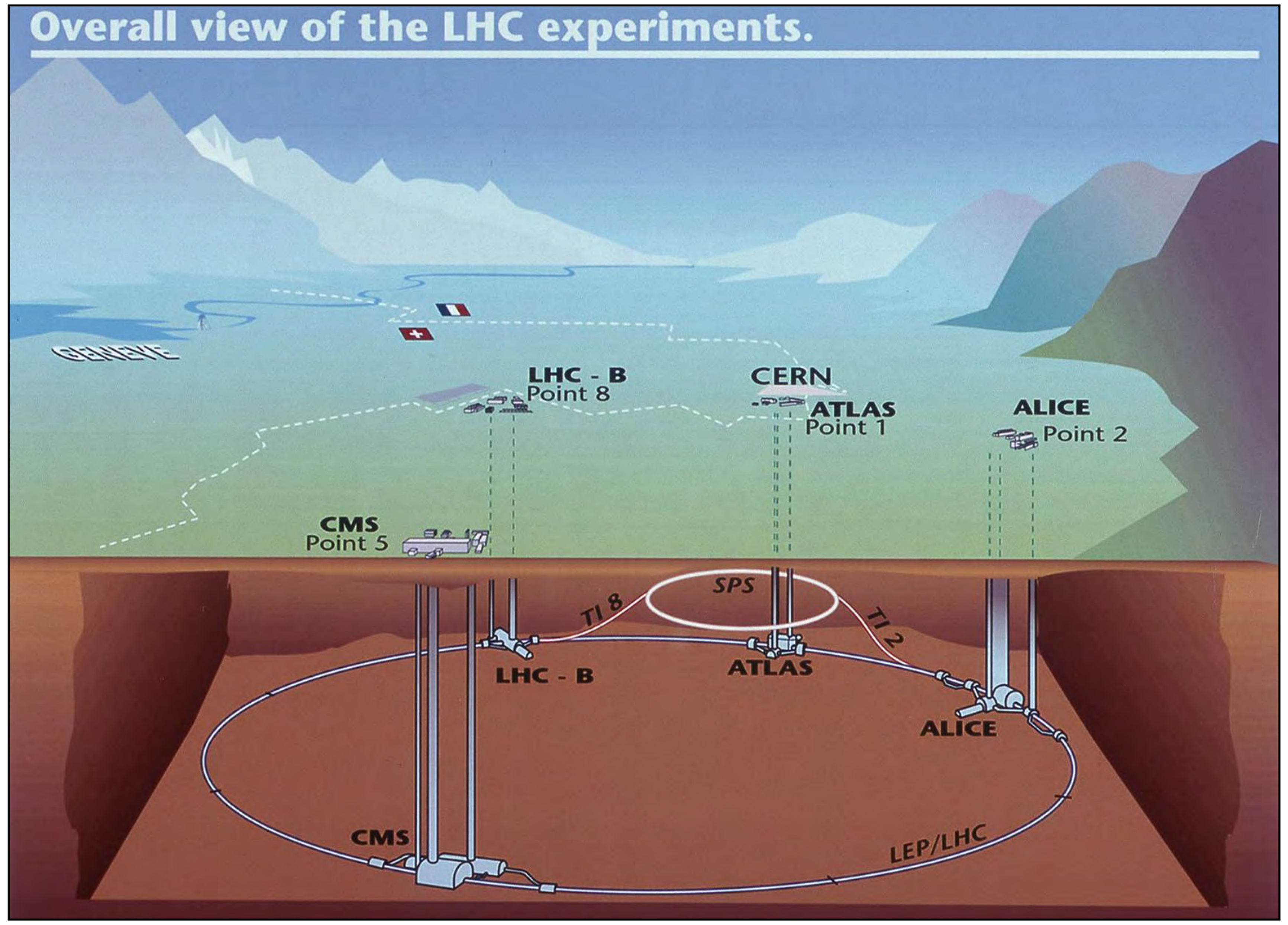
Proton-proton collisions at the Large Hadron Collider



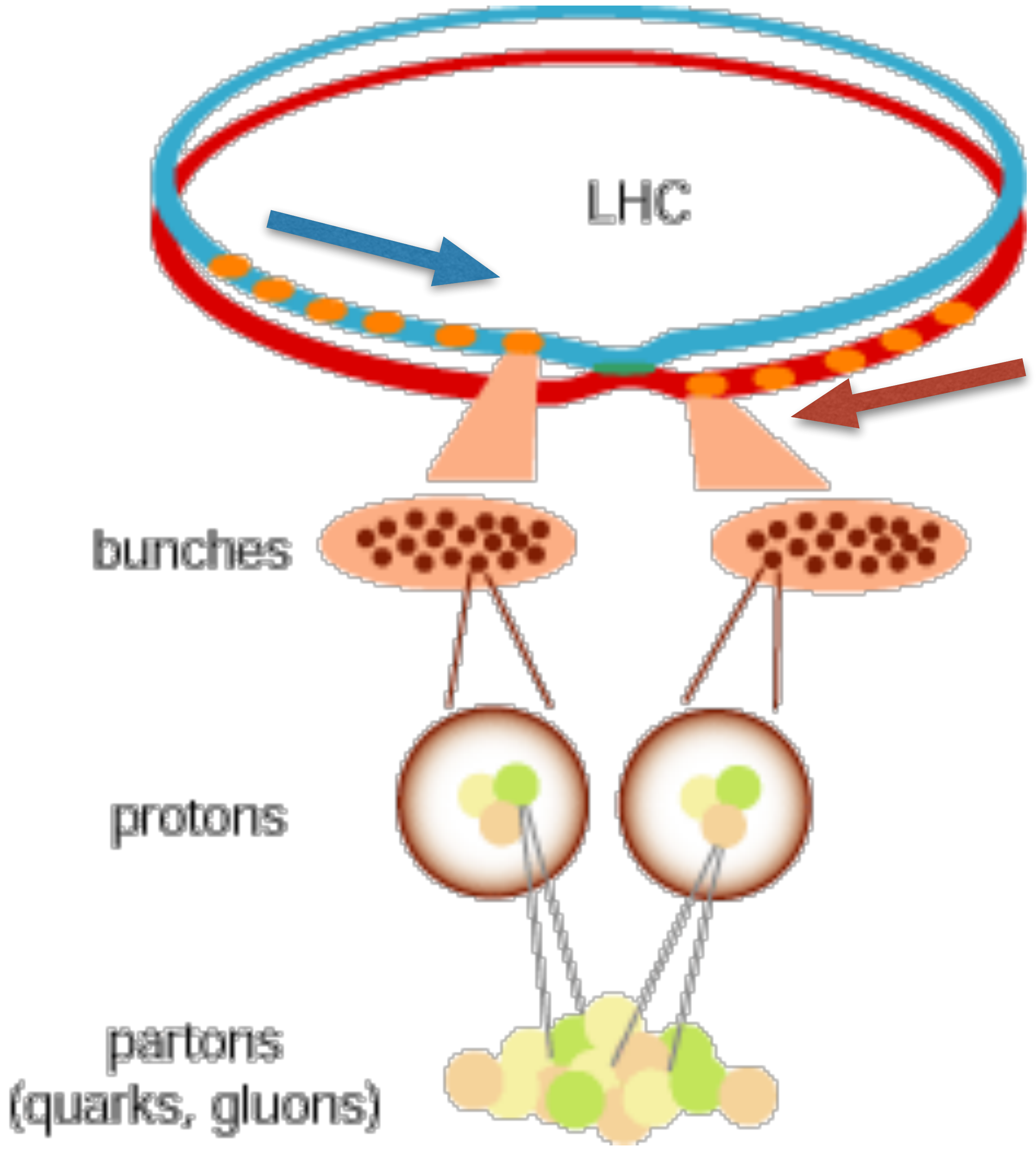
The LHC is 100 m below the surface, 27 km in circumference

Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple

Proton-proton collisions at the Large Hadron Collider



The LHC is 100 m below the surface, 27 km in circumference



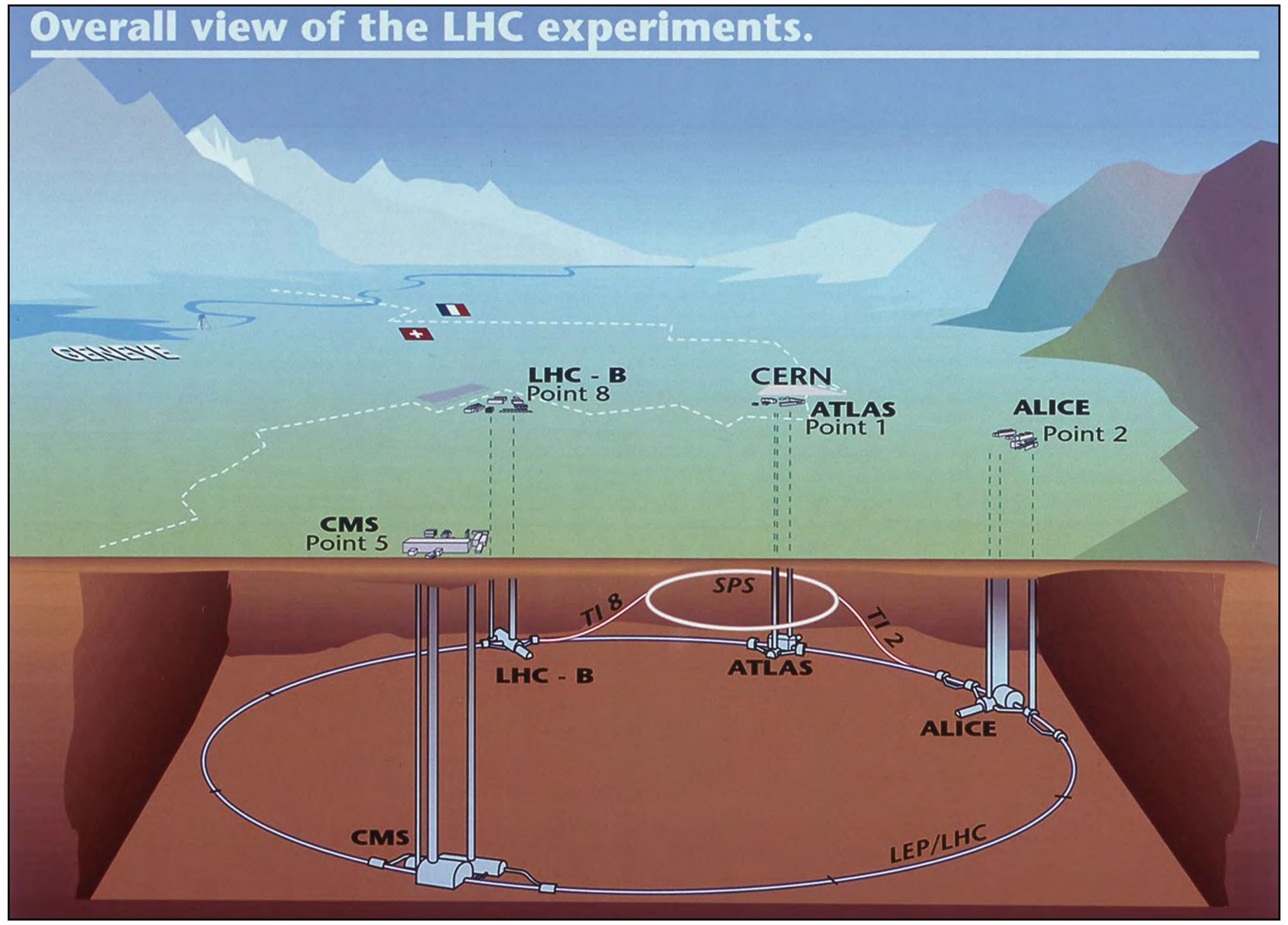
Structure of colliding proton bunches

Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple

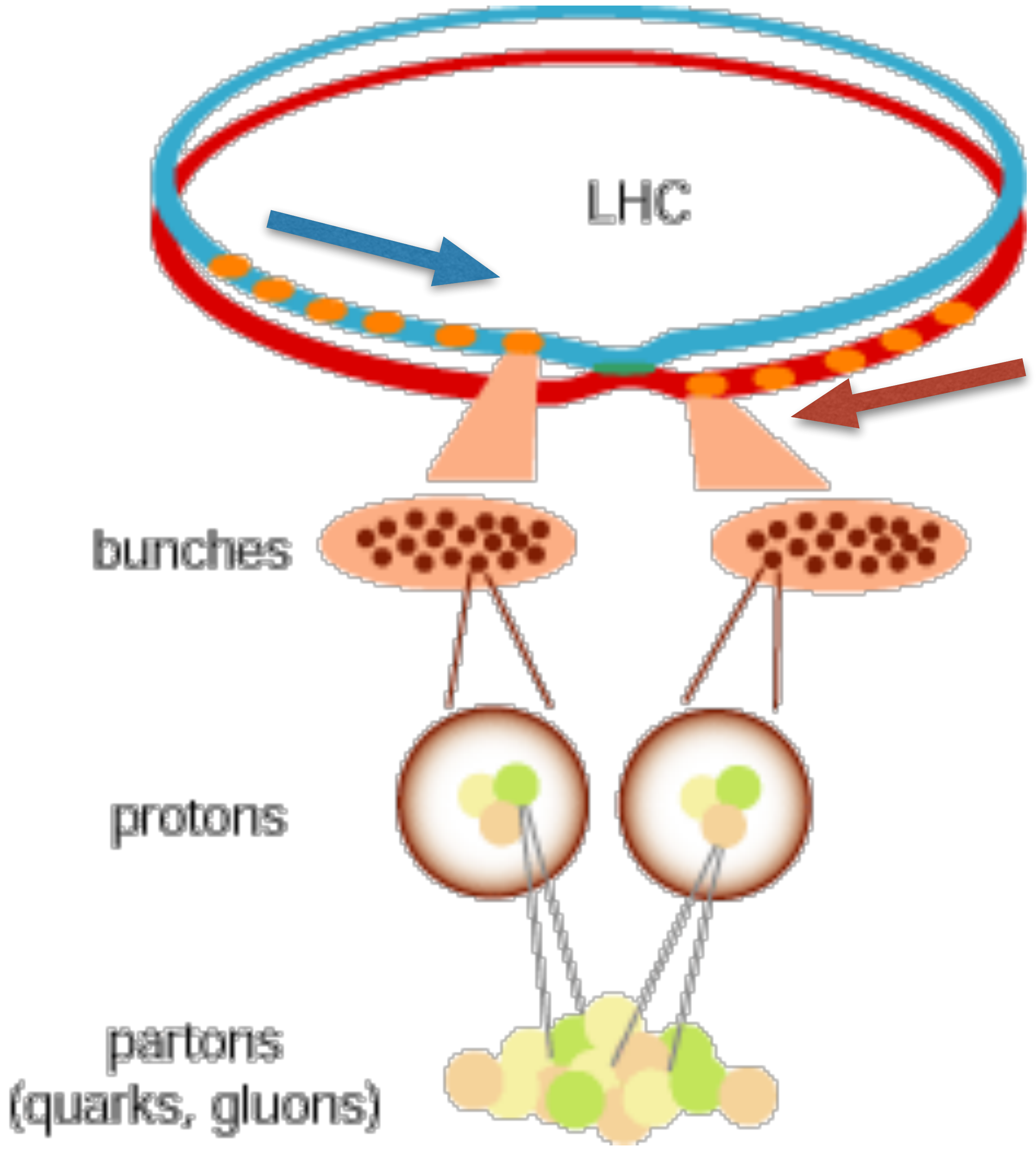
Chosen problem: Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)

- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed **bunches**.
- Each bunch contains ~ 100 billion protons

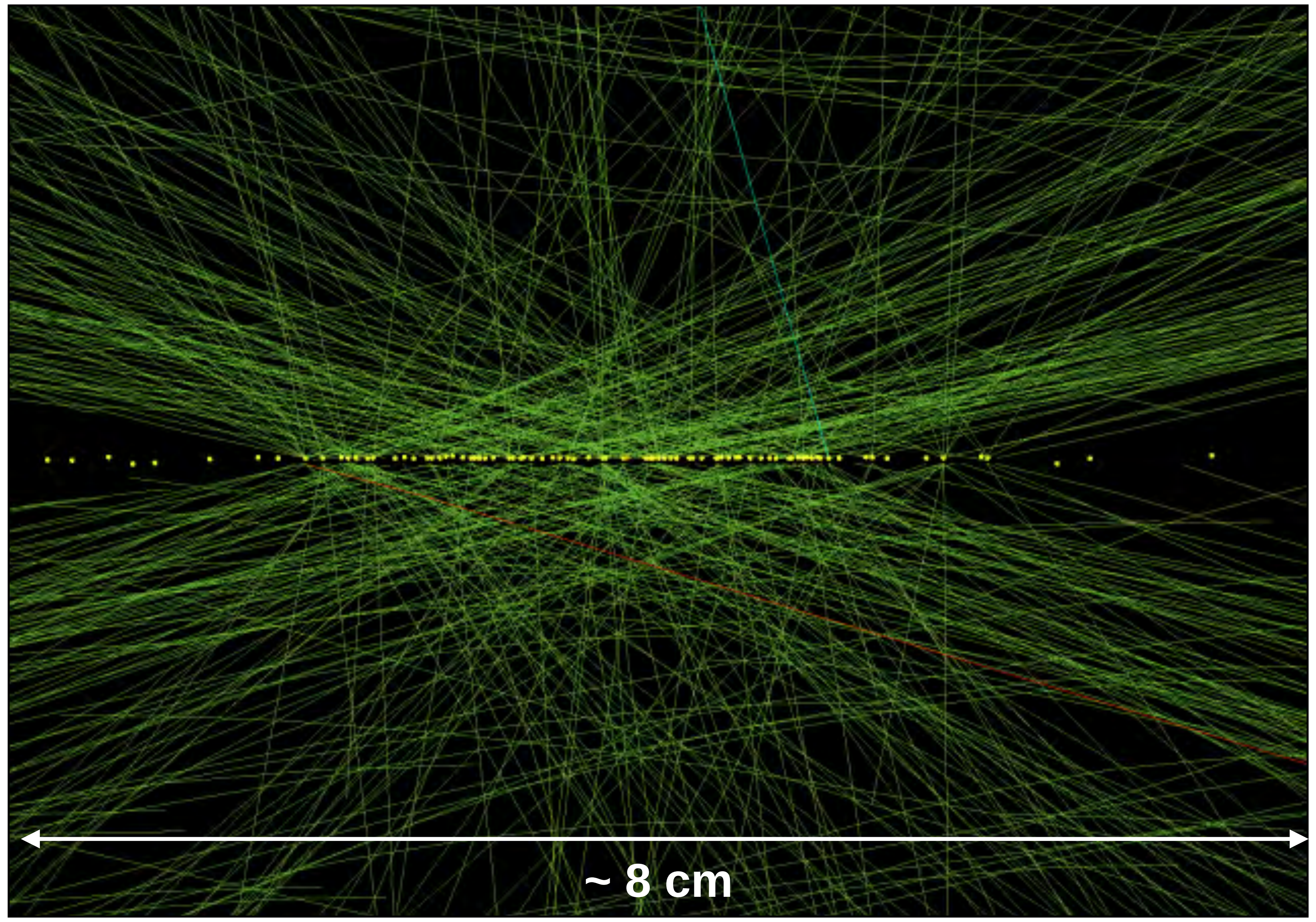
Proton-proton collisions at the Large Hadron Collider



The LHC is 100 m below the surface, 27 km in circumference



Structure of colliding proton bunches



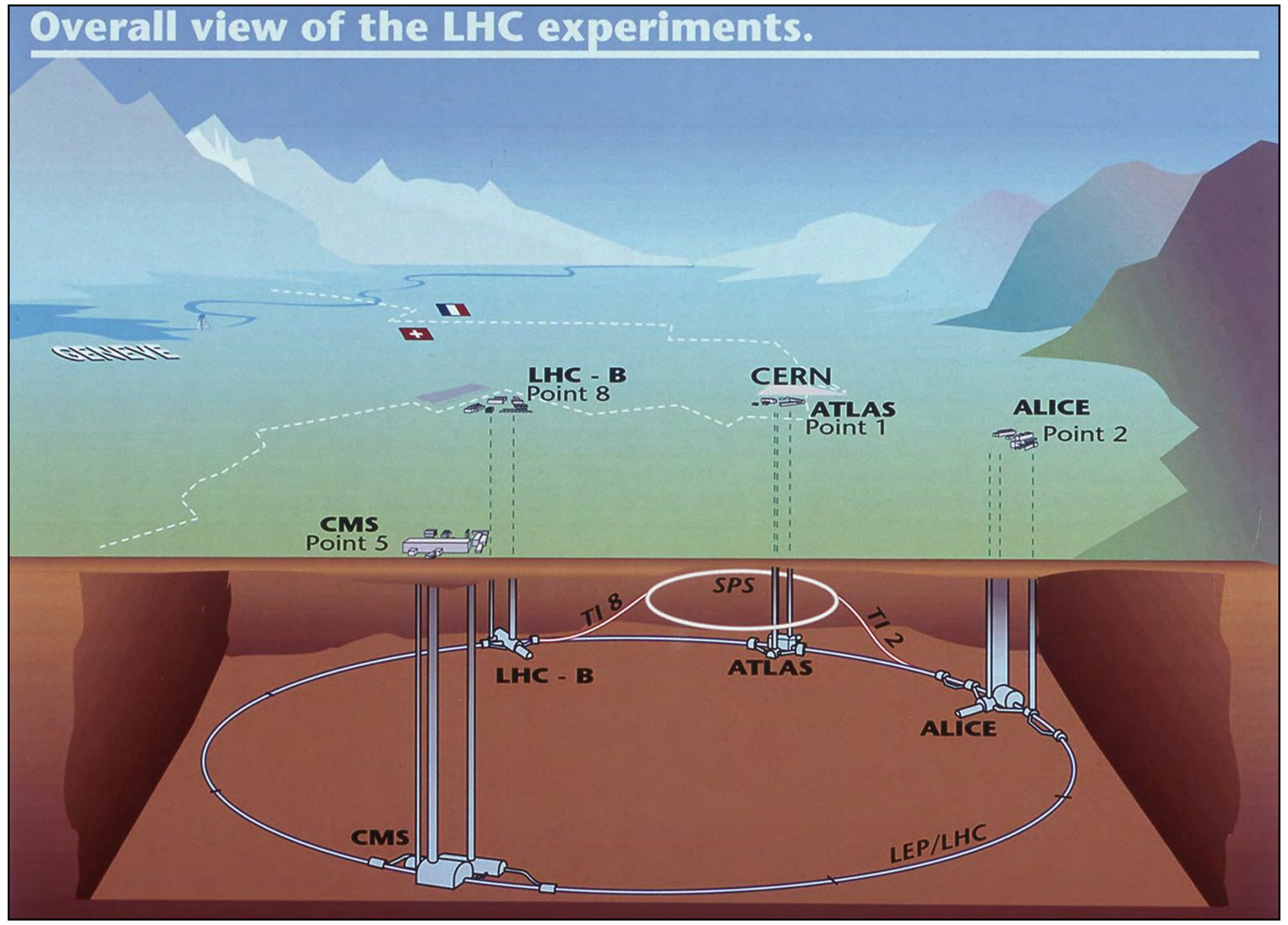
Which tracks originate together from a p-p collision?

Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple

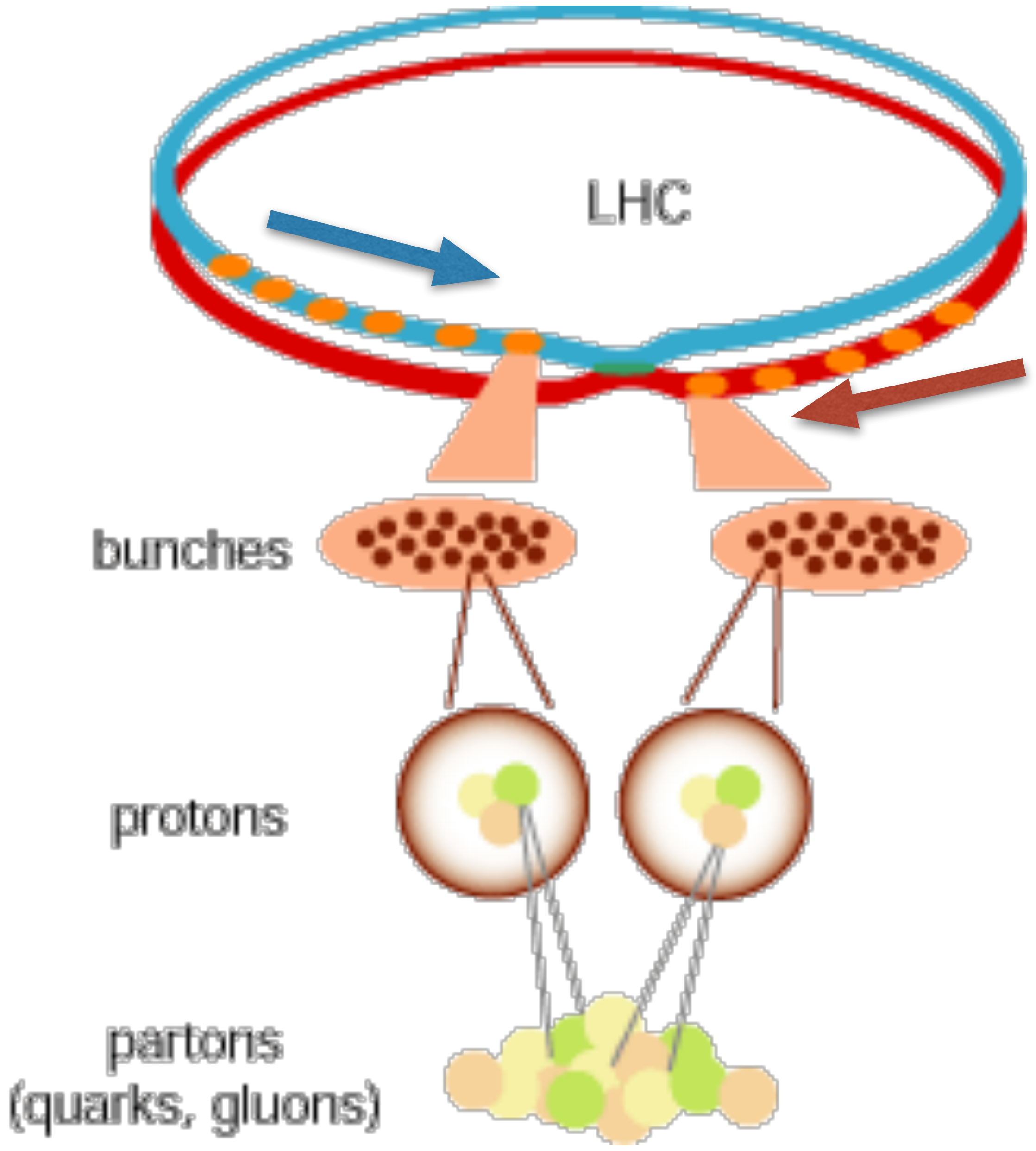
Chosen problem: Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)

- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed **bunches**.
- Each bunch contains ~ 100 billion protons
- When counter-rotating bunches cross, **only ~ 20 protons collide** in one straight line
- Each p-p collision results in **~ 50 “interesting” tracks** from charged particles produced

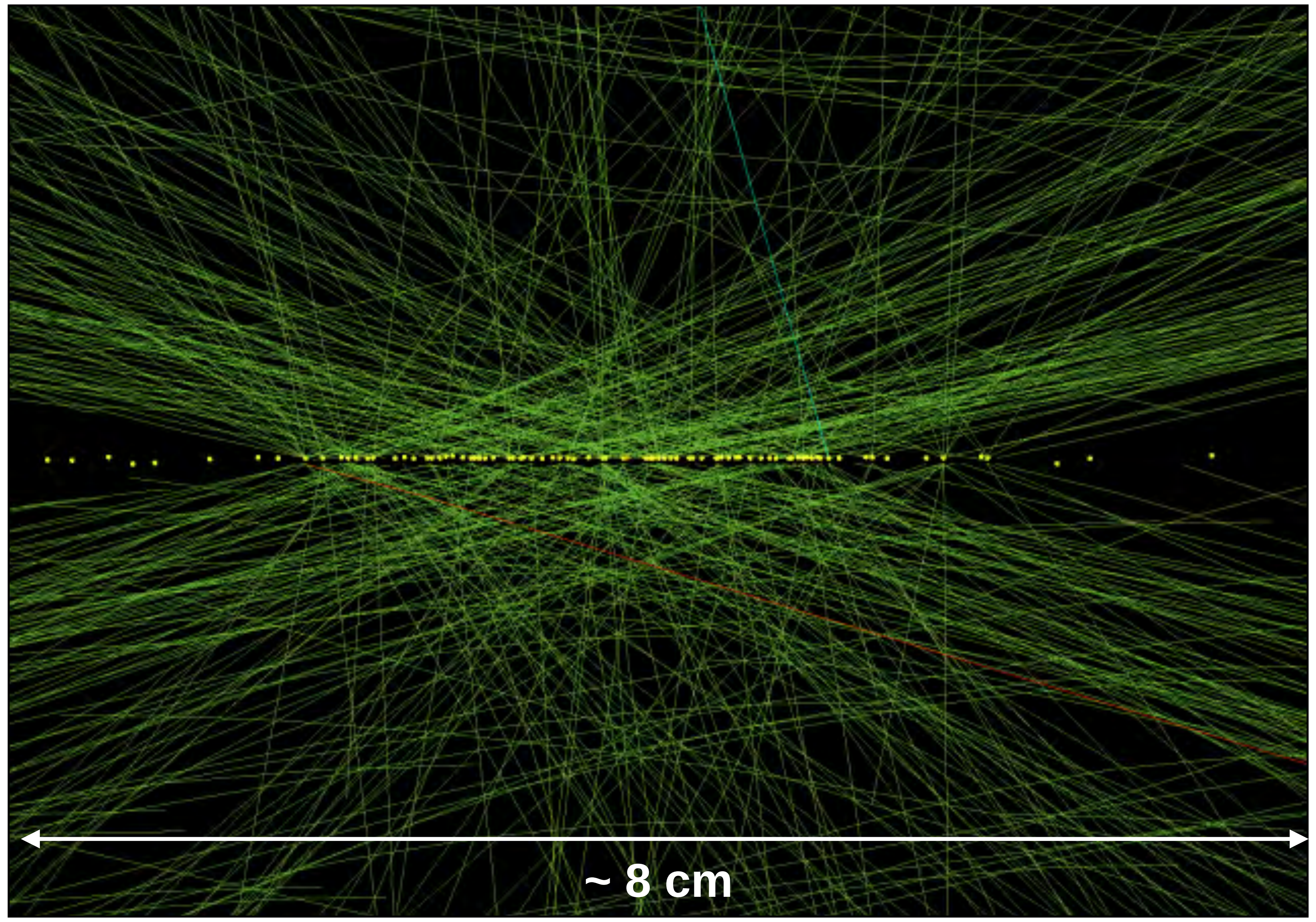
Proton-proton collisions at the Large Hadron Collider



The LHC is 100 m below the surface, 27 km in circumference



Structure of colliding proton bunches



Which tracks originate together from a p-p collision?

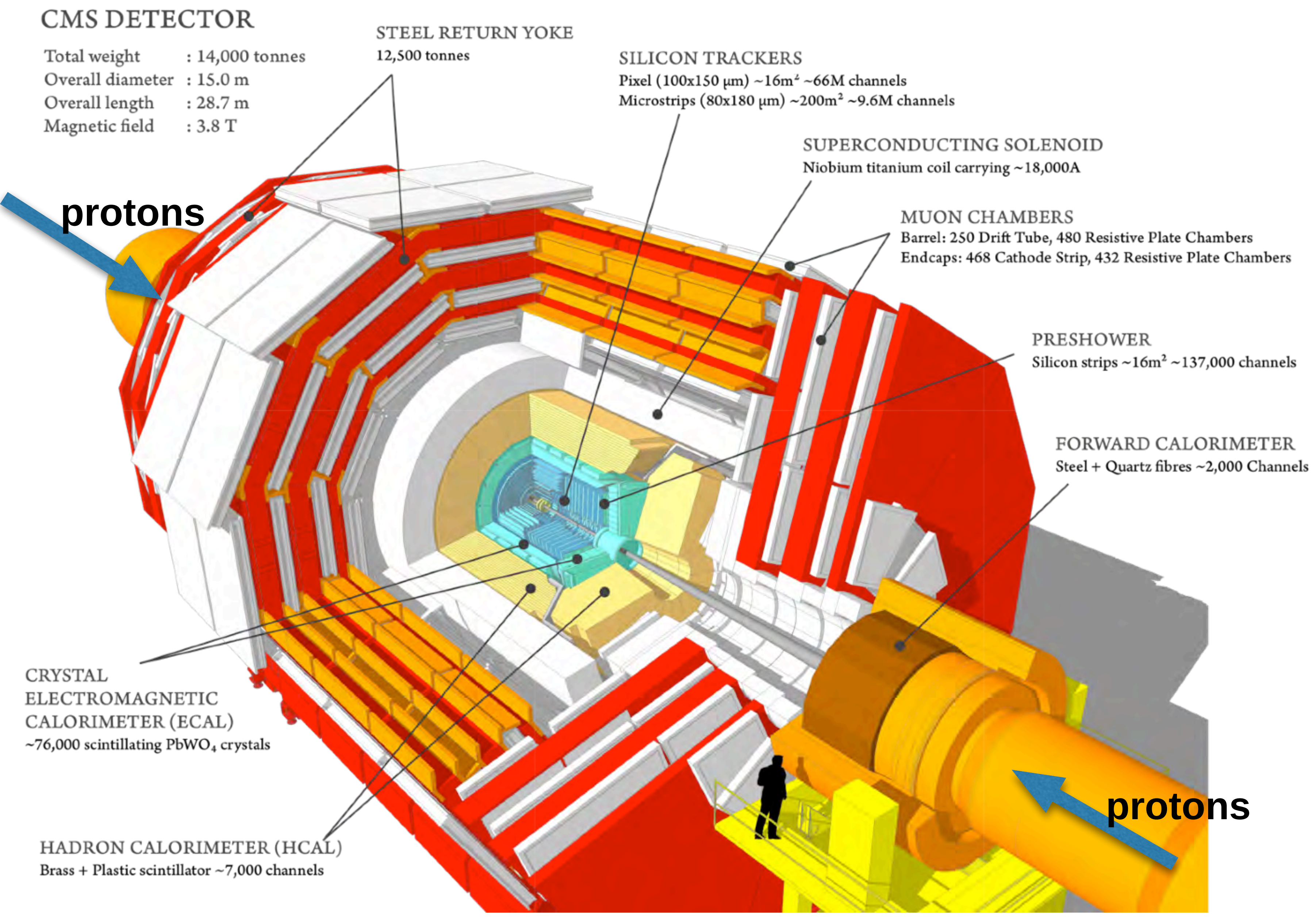
Looking for: A high-energy physics problem that has a natural formulation for quantum annealing, and is simple

Chosen problem: Reconstructing proton-proton collision positions at the Large Hadron Collider (LHC)

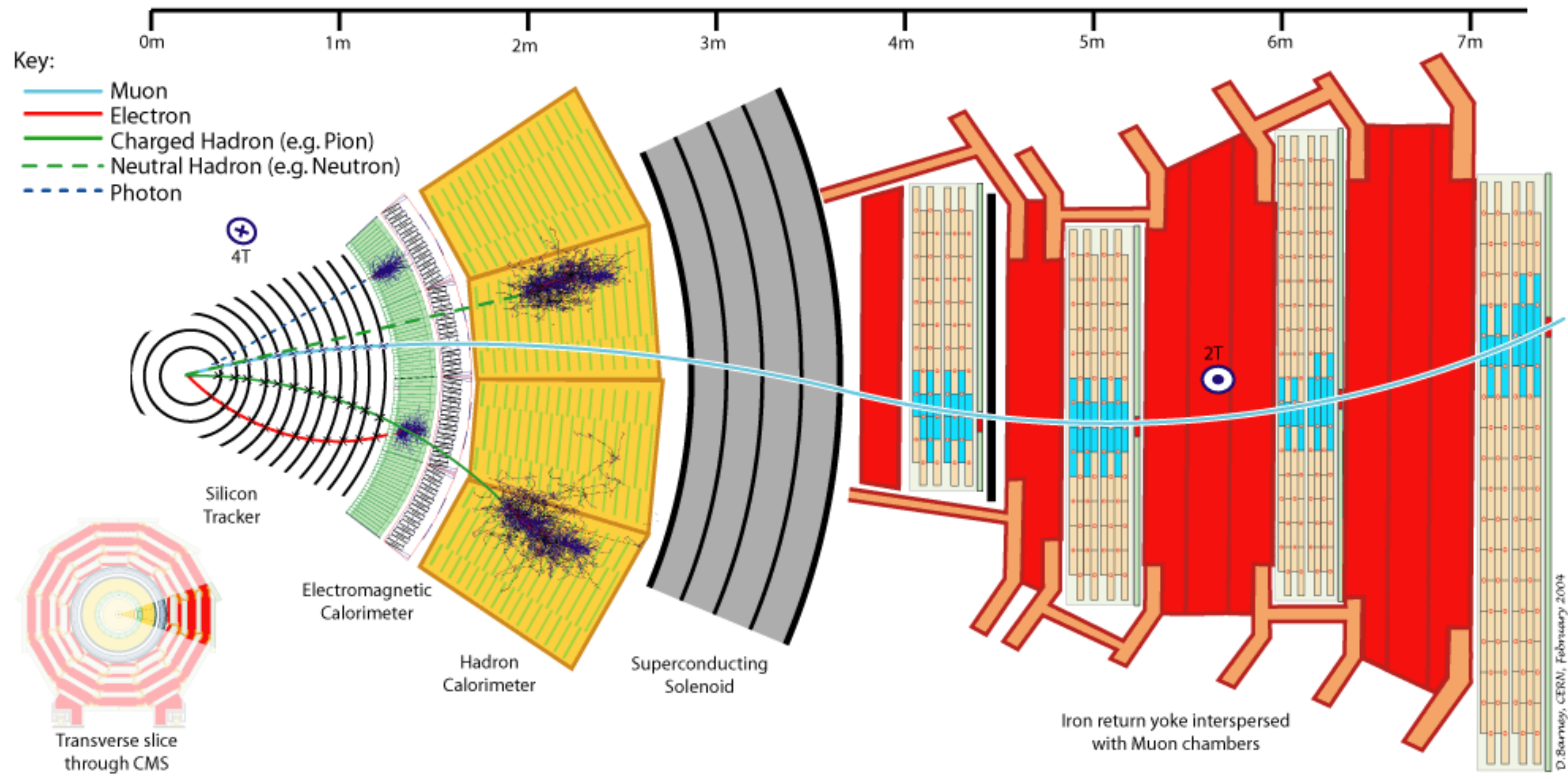
- The LHC circulates protons inside its beam-pipes not in a continuous stream but in several closely packed **bunches**.
- Each bunch contains ~ 100 billion protons
- When counter-rotating bunches cross, **only ~ 20 protons collide** in one straight line
- Each p-p collision results in **~ 50 "interesting" tracks** from charged particles produced

Which tracks come from which p-p collision?
Where are the p-p collision points in a bunch?

p-p collision position reconstruction at the Compact Muon Solenoid



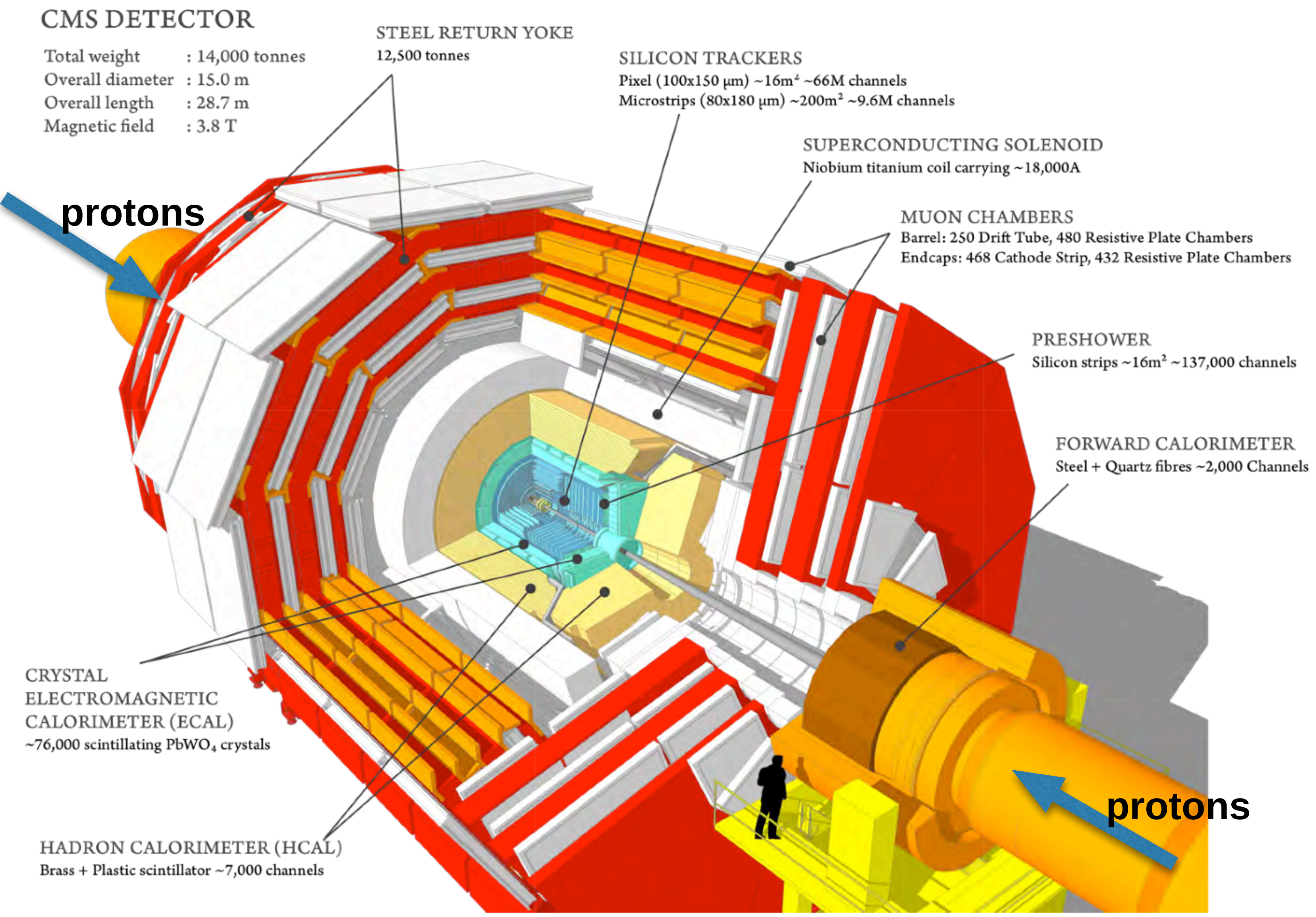
The CMS detector observes particles created at LHC collisions



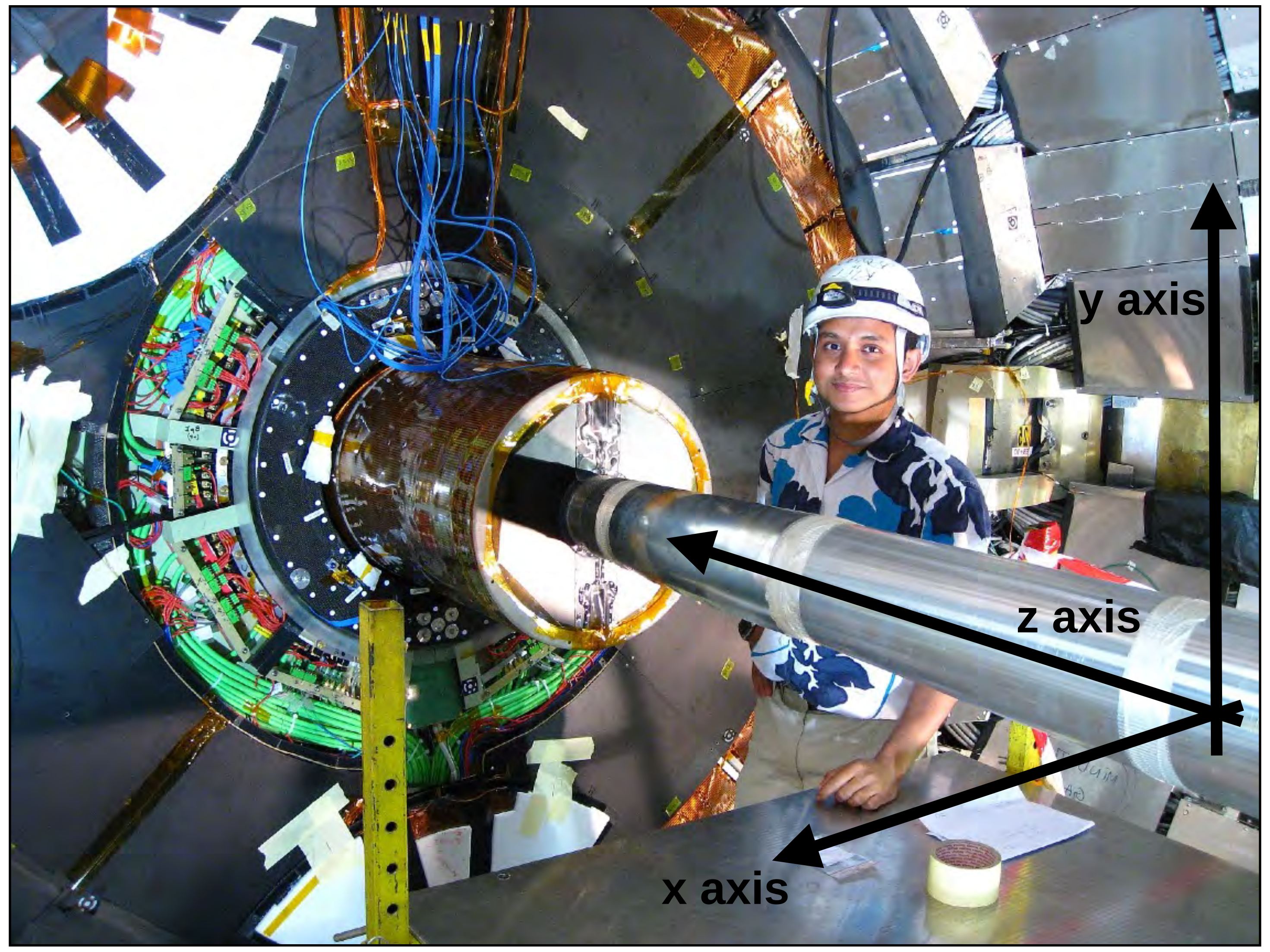
In the x-y plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties

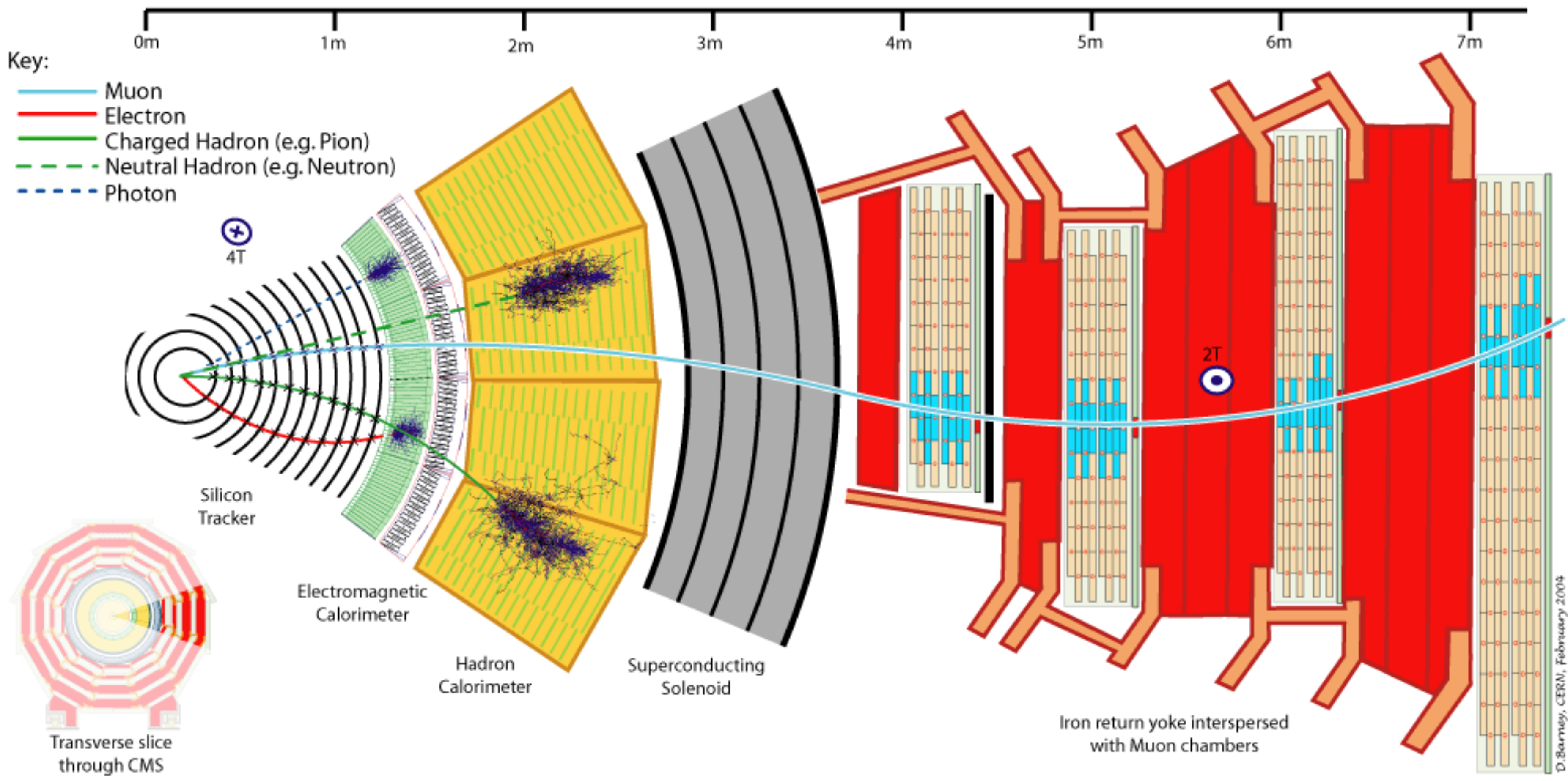
p-p collision position reconstruction at the Compact Muon Solenoid



The CMS detector observes particles created at LHC collisions



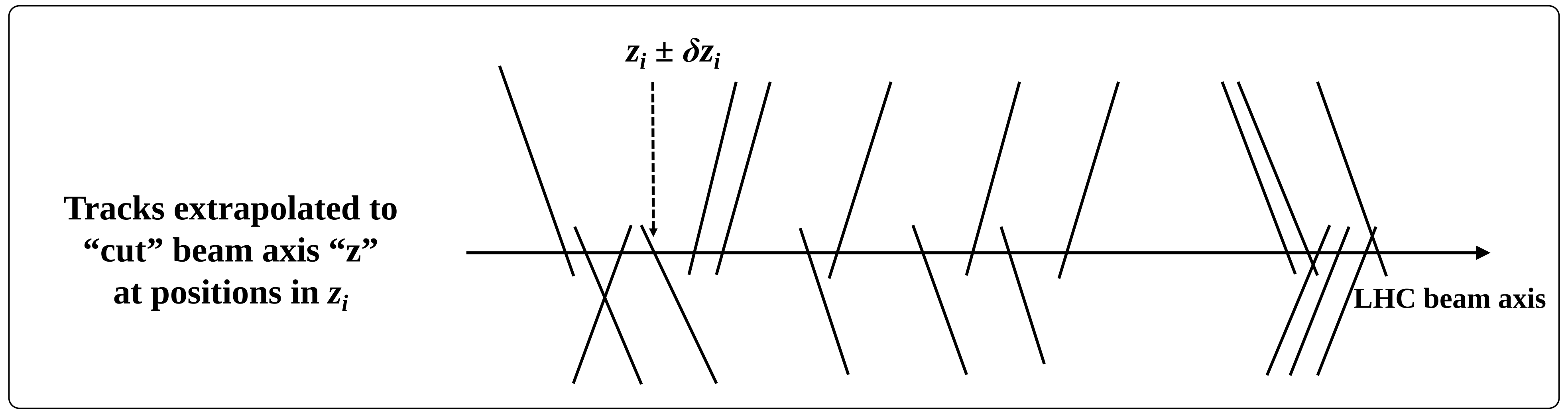
Graduate school, 2007



In the x-y plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

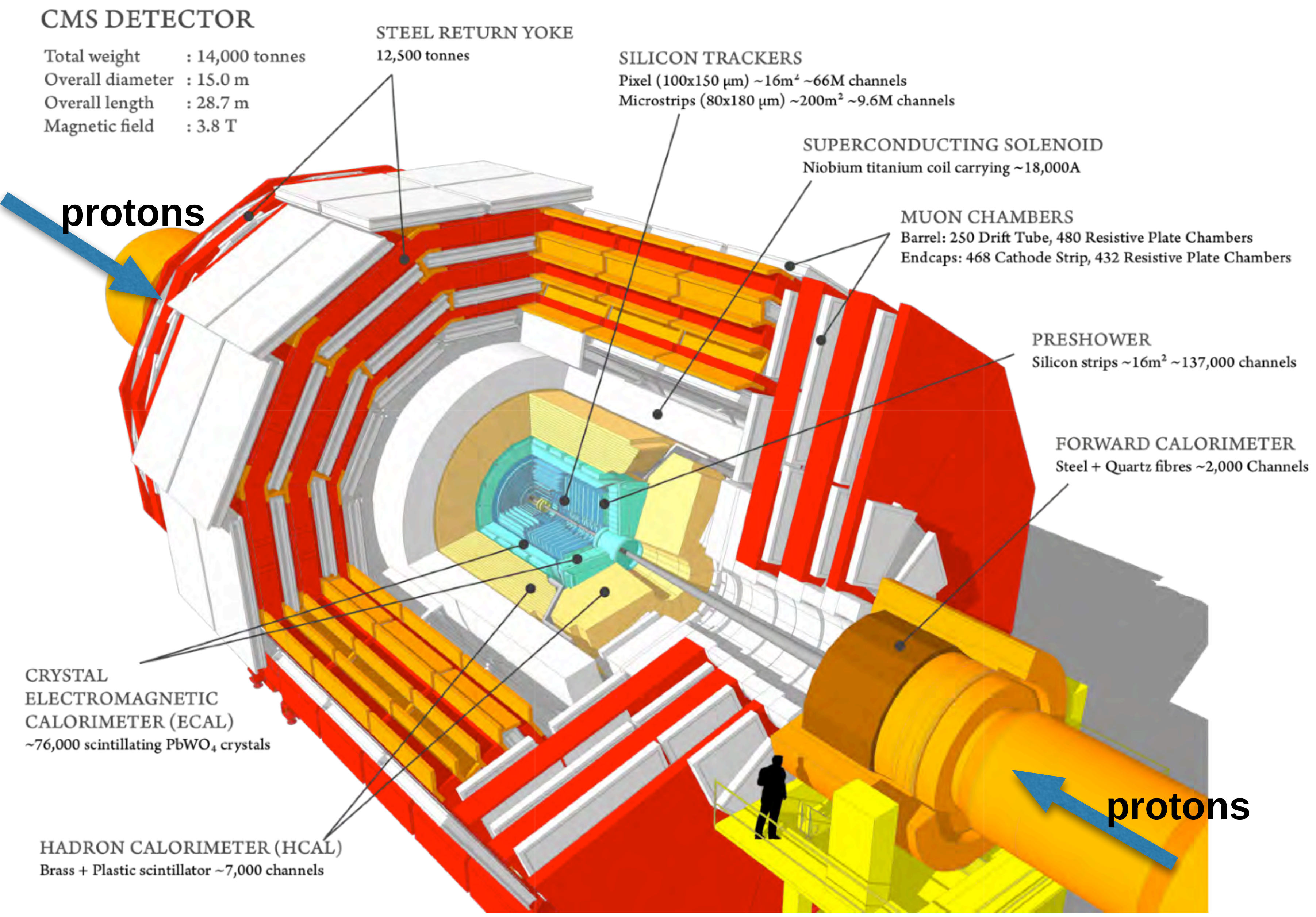
- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties

- Where a track approaches the beam (z-)axis closest has uncertainties $\boldsymbol{z_i \pm \delta z_i}$
- Uncertainties obscure which tracks originated together at p-p collision

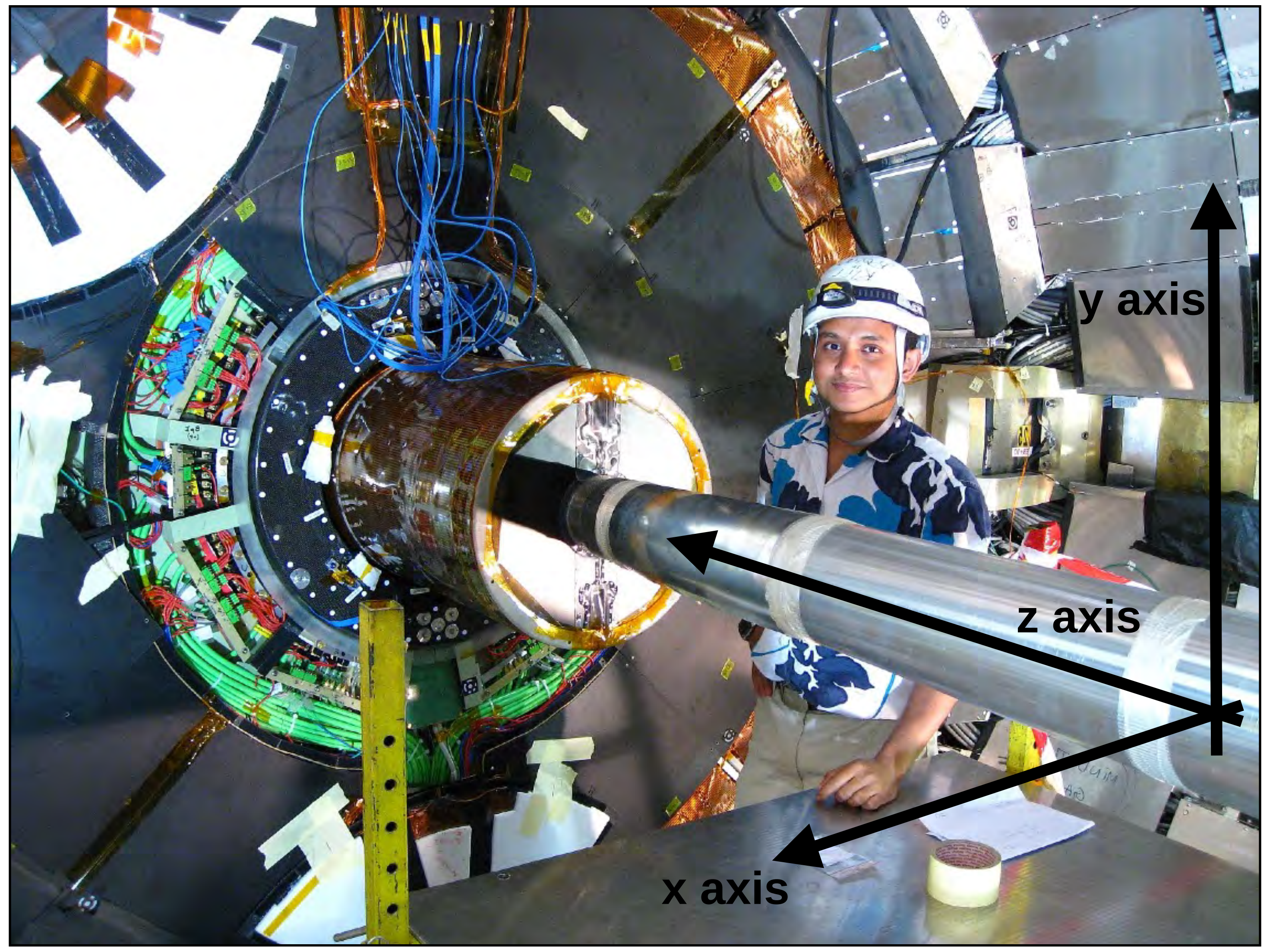


In the y-z plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

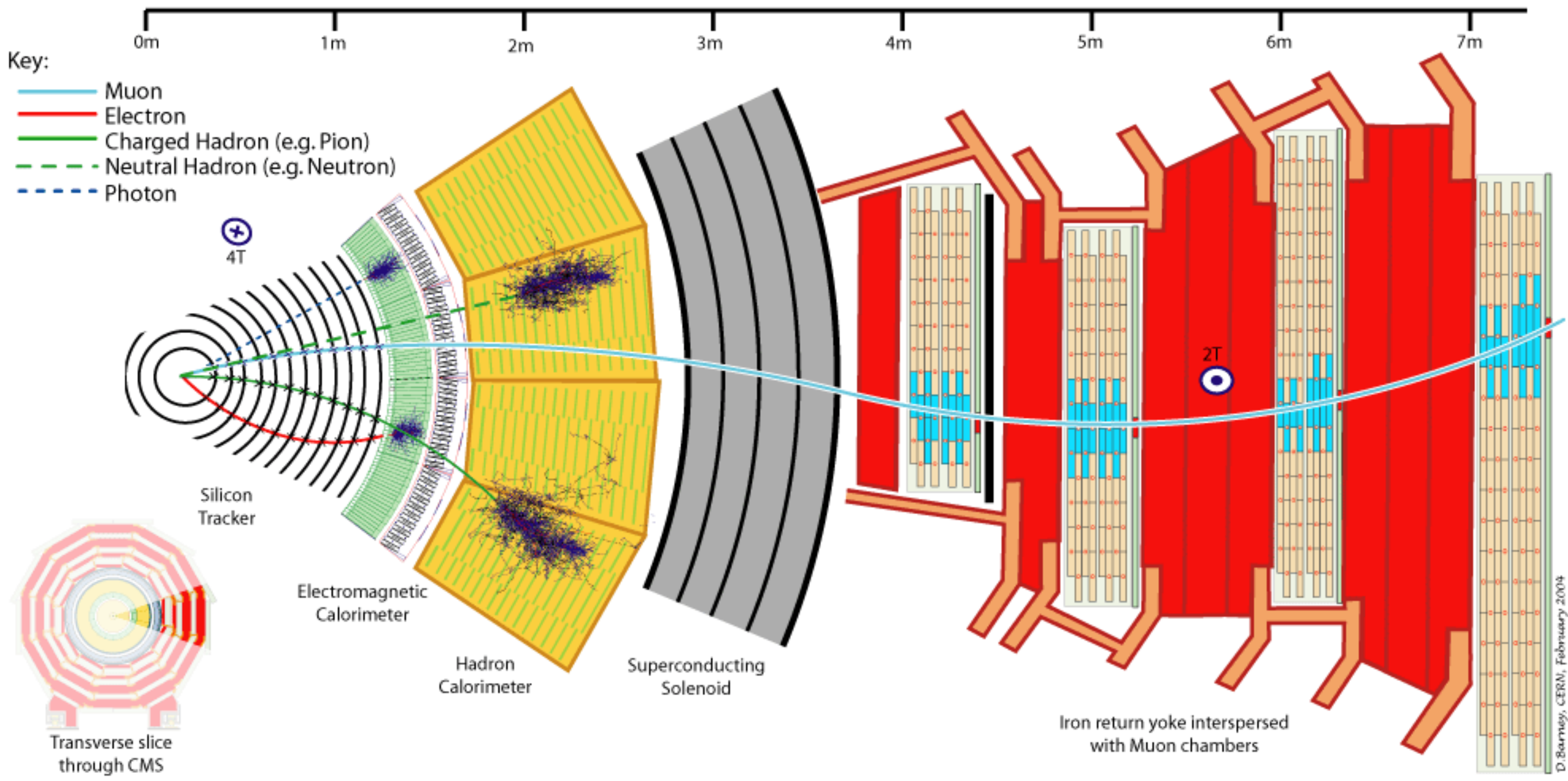
p-p collision position reconstruction at the Compact Muon Solenoid



The CMS detector observes particles created at LHC collisions



Graduate school, 2007

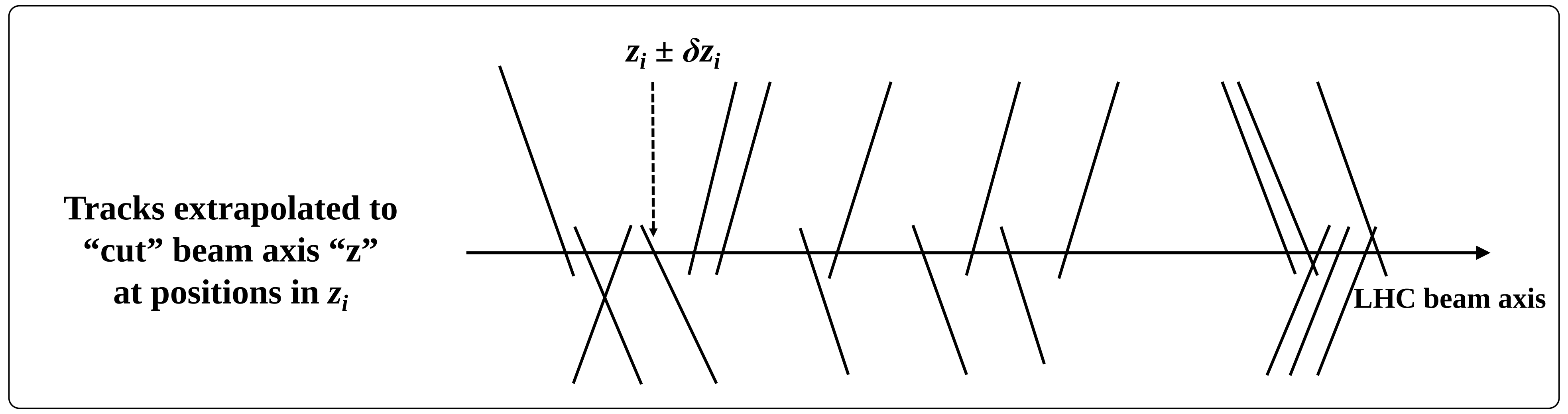


In the x-y plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties

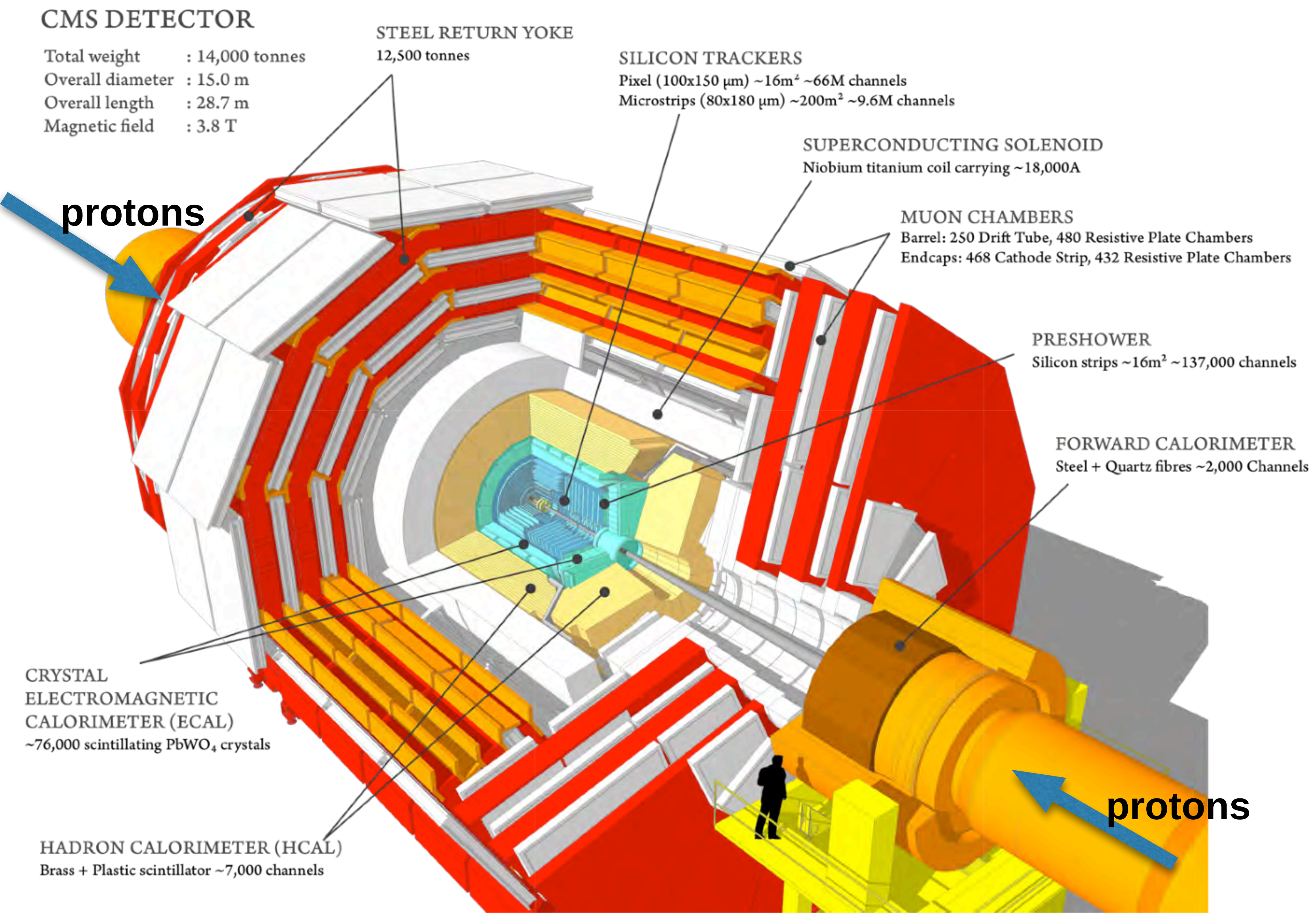
- Where a track approaches the beam (z-)axis closest has uncertainties $z_i \pm \delta z_i$
- Uncertainties obscure which tracks originated together at p-p collision

- Position of p-p collisions reduced to a clustering problem in 1-D
- Solved in CMS using Deterministic Annealing. Called “Primary Vertexing”

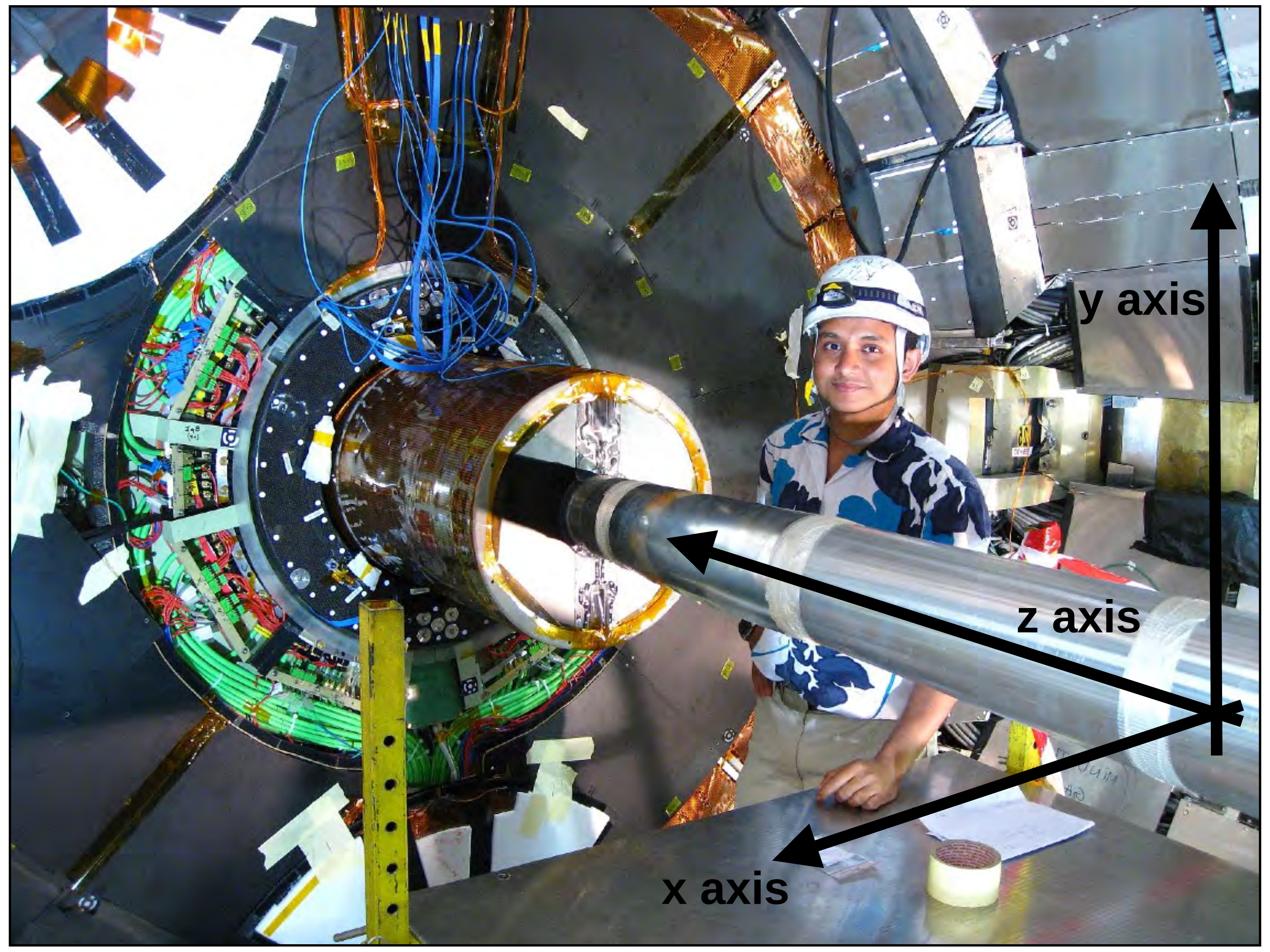


In the y-z plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

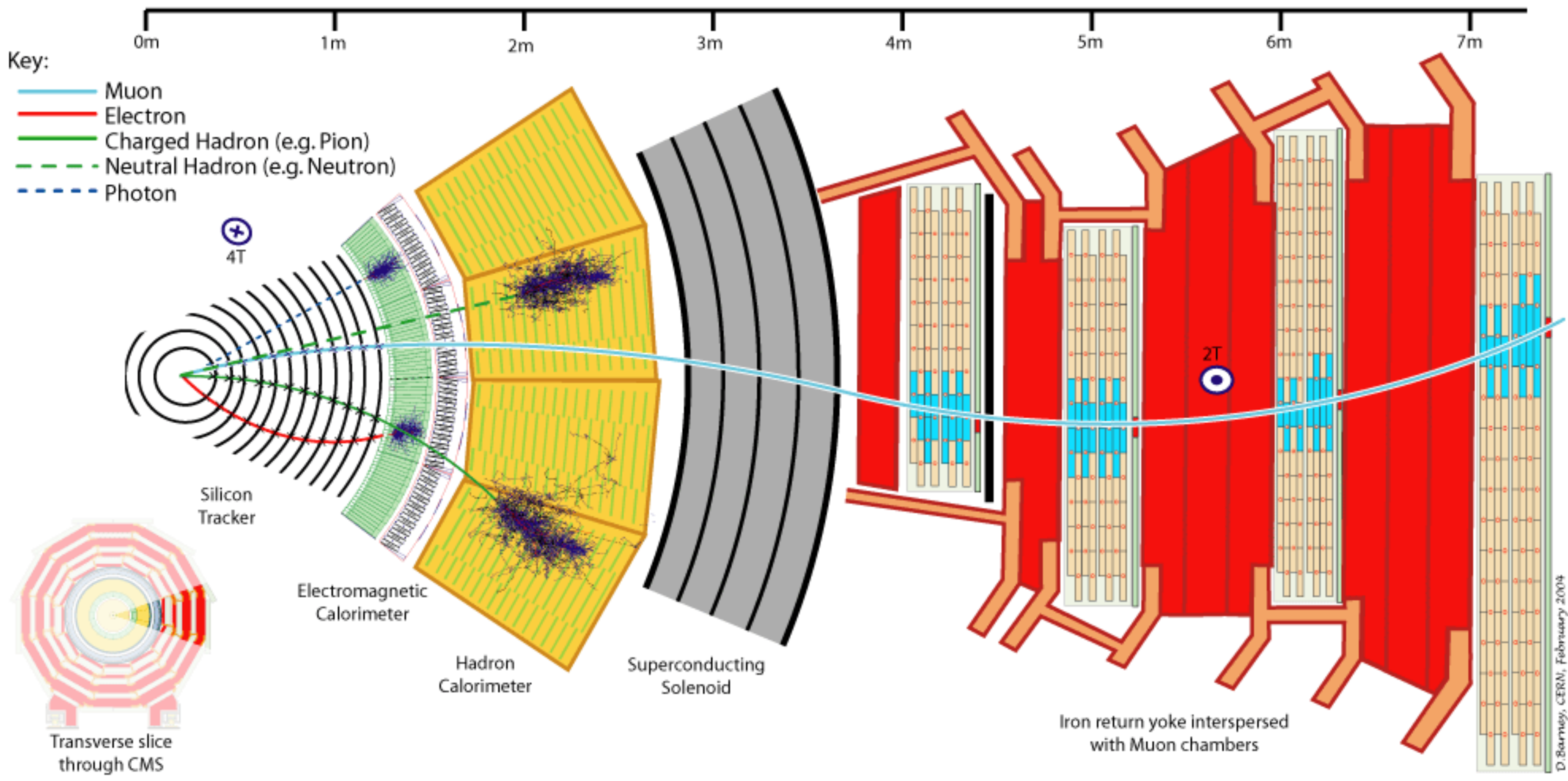
p-p collision position reconstruction at the Compact Muon Solenoid



The CMS detector observes particles created at LHC collisions



Graduate school, 2007

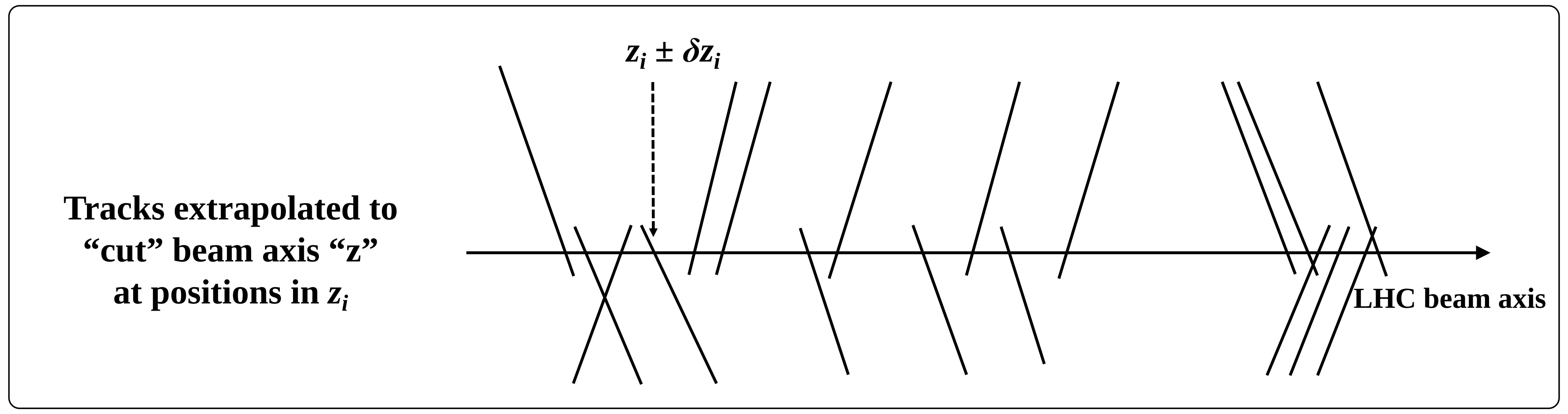


In the x-y plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

- The Compact Muon Solenoid is a particle detector at one (of four) p-p crossing point at the LHC
- Charged particles are reconstructed as tracks. All reconstructions come with uncertainties

- Where a track approaches the beam (z-)axis closest has uncertainties $z_i \pm \delta z_i$
- Uncertainties obscure which tracks originated together at p-p collision

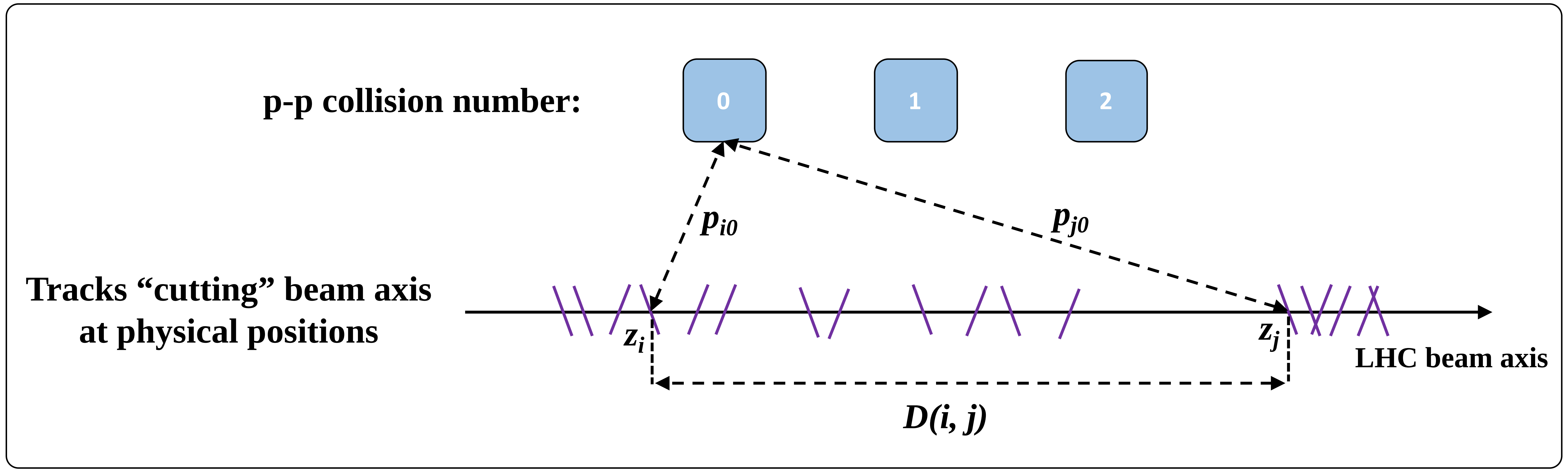
- Position of p-p collisions reduced to a clustering problem in 1-D
- Solved in CMS using Deterministic Annealing. Called “Primary Vertexing”



In the y-z plane. Particle trajectories reconstructed as tracks. Reconstructions come with uncertainties

Can D-Wave solve it using quantum annealing?

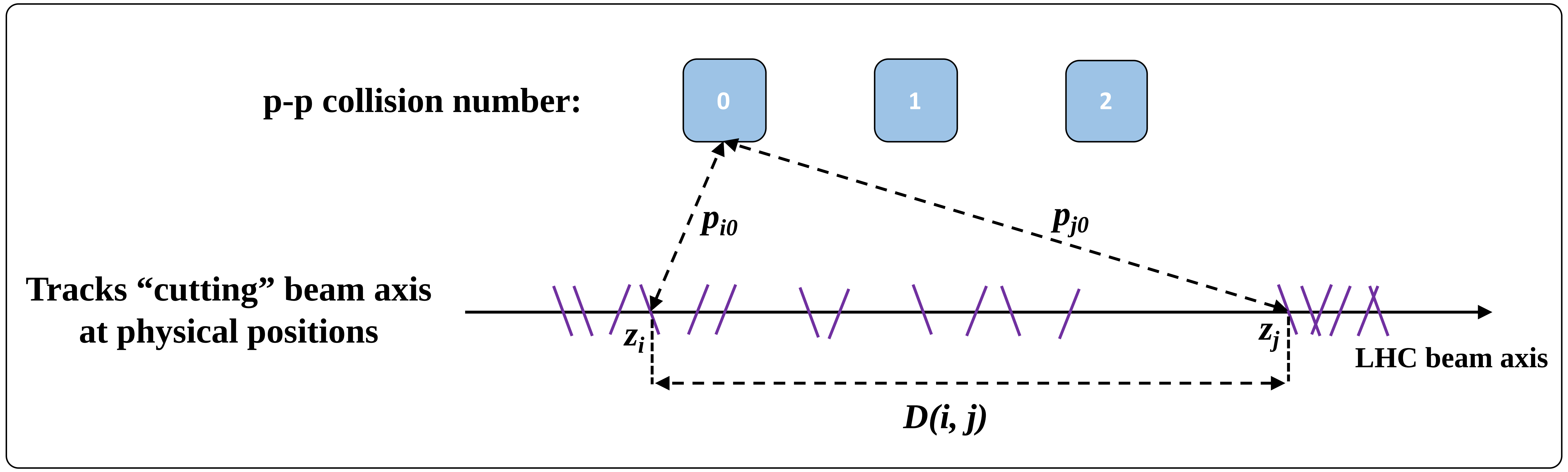
Track clustering QUBO formulation for D-Wave



$$H_p = \sum_{k}^{n_V} \sum_{i}^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j); m) + \lambda \sum_{i}^{n_T} \left(1 - \sum_{k}^{n_V} p_{ik}\right)^2 ,$$

- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *“Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39*)

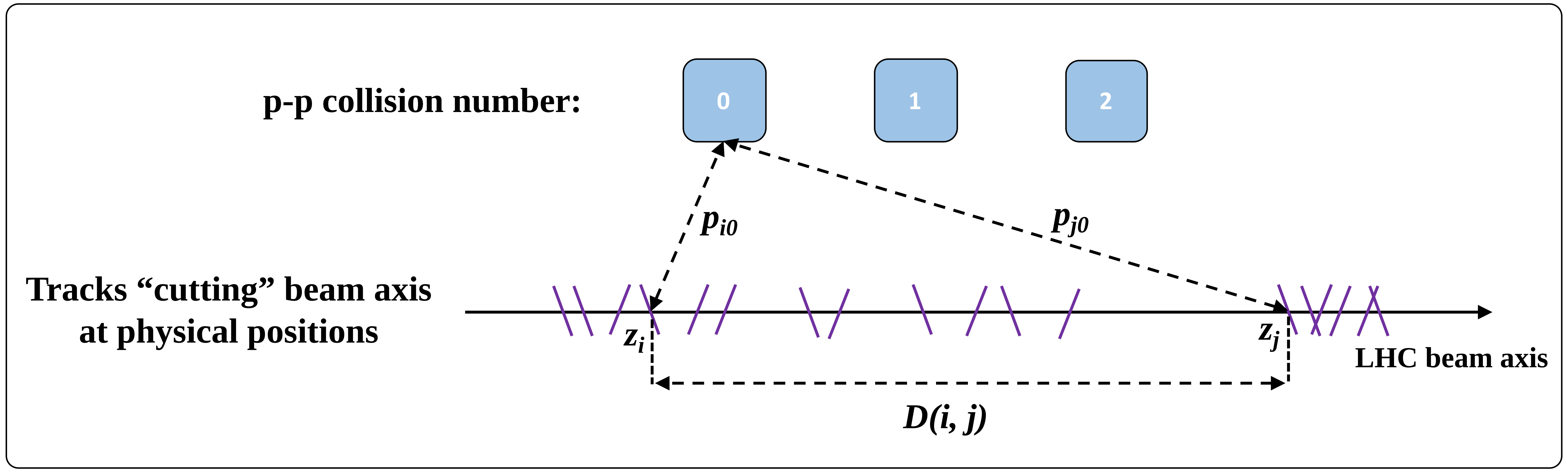
Track clustering QUBO formulation for D-Wave



$$H_p = \sum_{k}^{n_V} \sum_{i}^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j); m) + \lambda \sum_{i}^{n_T} \left(1 - \sum_{k}^{n_V} p_{ik} \right)^2 ,$$

- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *"Quantum annealing for combinatorial clustering" Quantum Inf. Processing 17 (2018) 39*)
  - Probability (0 or 1) of i^{th} track to have come from k^{th} p-p collision is p_{ik}. Element $\boldsymbol{p_{ik}}$ **is represented by a qubit**

Track clustering QUBO formulation for D-Wave

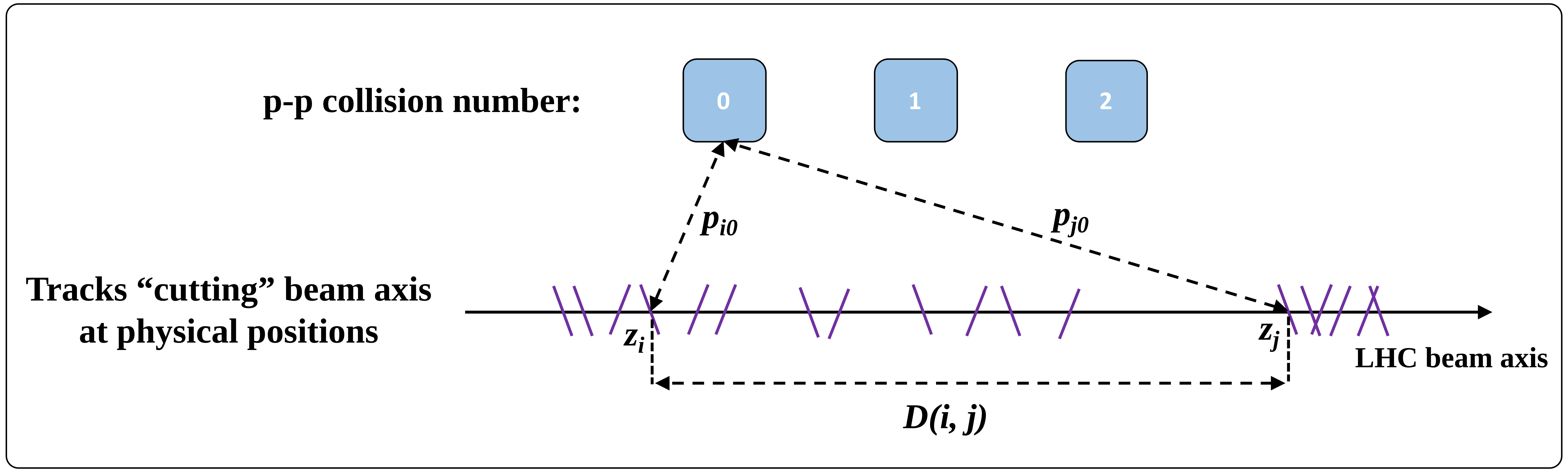


Coupling terms

$$H_p = \sum_k^{n_V} \sum_i^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j);m) + \lambda \sum_i^{n_T} \left(1 - \sum_k^{n_V} p_{ik}\right)^2 ,$$

- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *“Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39*)
  - ➡ Probability (0 or 1) of i^{th} track to have come from k^{th} p-p collision is p_{ik}. Element $\boldsymbol{p_{ik}}$ **is represented by a qubit**
  - ➡ Coupling between two qubits p_{ik} and p_{jk} that represent association of two tracks to same p-p collision k is a distance measure between the tracks *D(i, j)*. **Punish associations corresponding to widely separated tracks**
    - ➡ *D(i, j)* is Manhattan distance attenuated by uncertainty $D(i,j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}.$

Track clustering QUBO formulation for D-Wave

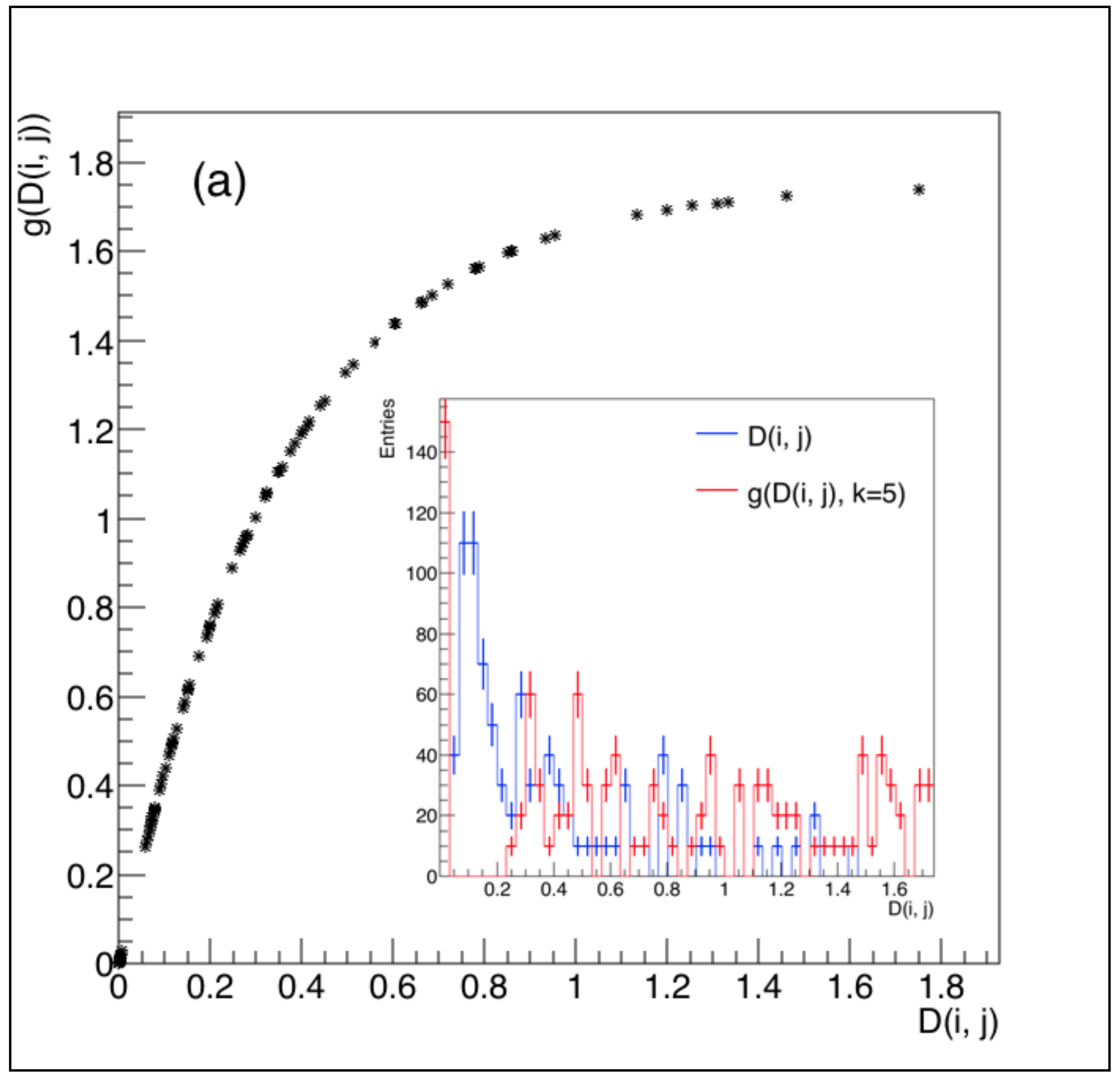


Coupling terms

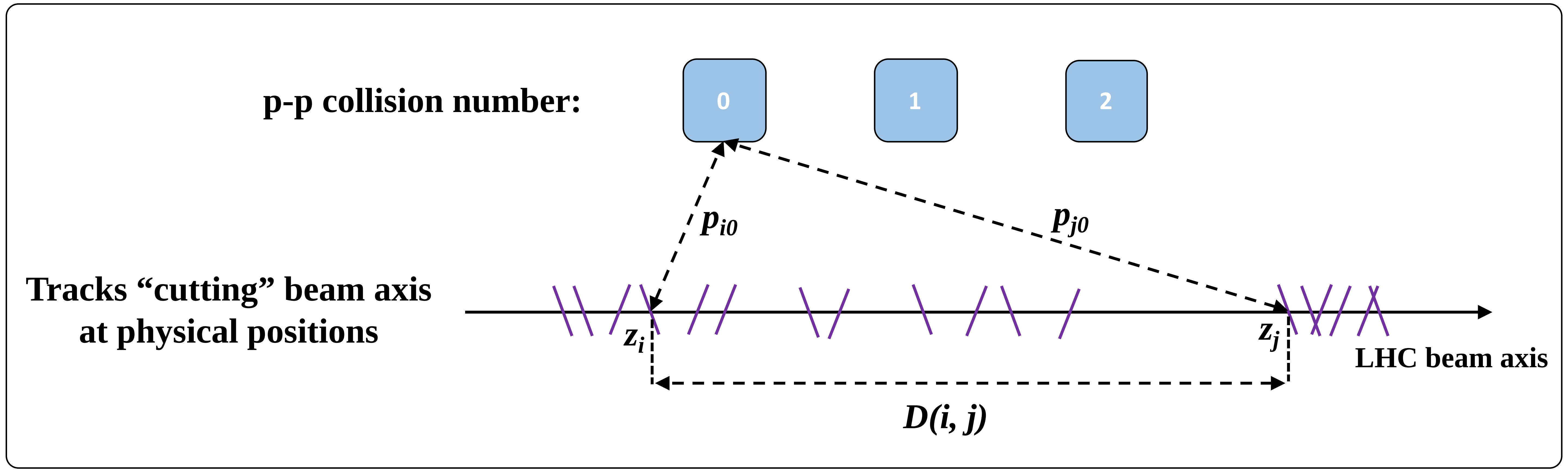
$$H_p = \sum_k^{n_V} \sum_i^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j); m) + \lambda \sum_i^{n_T} \left(1 - \sum_k^{n_V} p_{ik}\right)^2,$$

- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *"Quantum annealing for combinatorial clustering" Quantum Inf. Processing 17 (2018) 39*)
  - Probability (0 or 1) of i^{th} track to have come from k^{th} p-p collision is p_{ik}. Element $\boldsymbol{p_{ik}}$ **is represented by a qubit**
  - Coupling between two qubits p_{ik} and p_{jk} that represent association of two tracks to same p-p collision k is a distance measure between the tracks $D(i, j)$. **Punish associations corresponding to widely separated tracks**
    - $D(i, j)$ is Manhattan distance attenuated by uncertainty $$D(i,j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}.$$
    - $g(D(i, j); m)$ seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with $m = 5$

$$g(x; m) = 1 - e^{-mx},$$



Track clustering QUBO formulation for D-Wave



Coupling terms

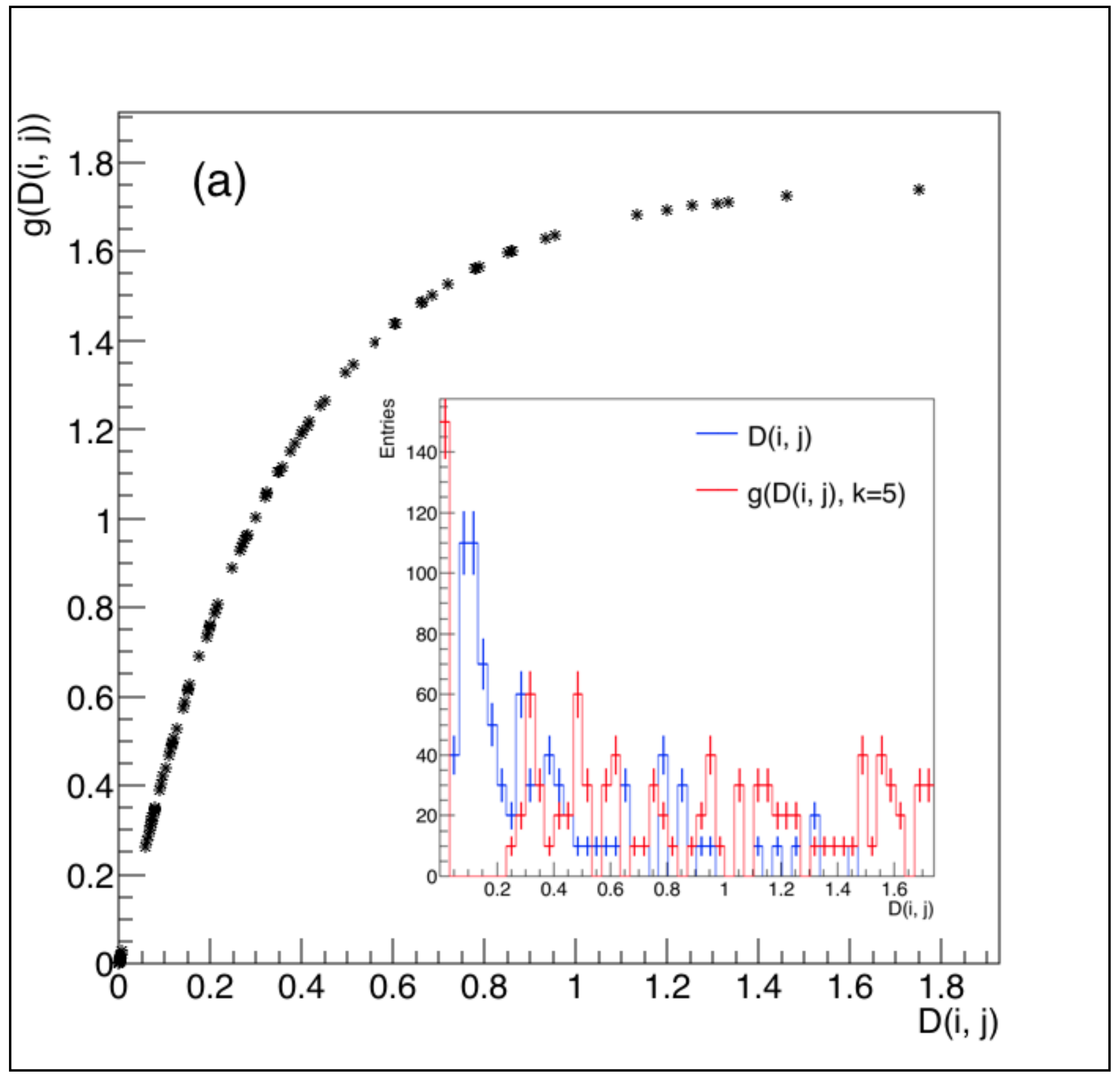
$$H_p = \sum_k^{n_V} \sum_i^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j); m) + \lambda \sum_i^{n_T} \left(1 - \sum_k^{n_V} p_{ik}\right)^2 ,$$

Bias terms come from here

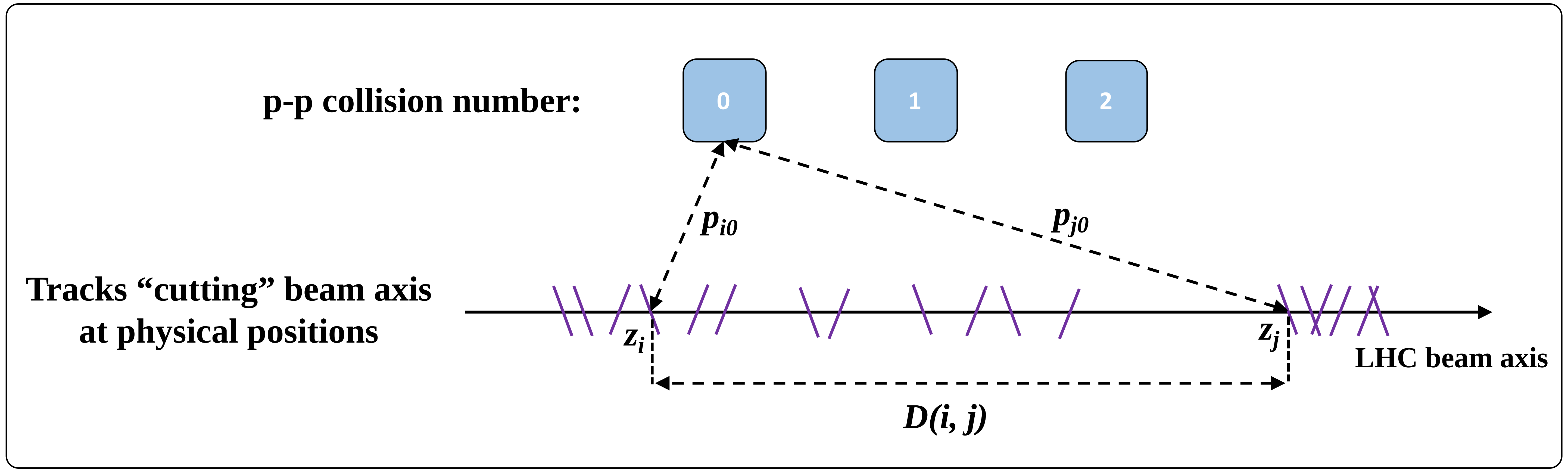
- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *“Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39*)
  - Probability (0 or 1) of i^{th} track to have come from k^{th} p-p collision is p_{ik}. Element p_{ik} **is represented by a qubit**
  - Coupling between two qubits p_{ik} and p_{jk} that represent association of two tracks to same p-p collision k is a distance measure between the tracks *D(i, j).* **Punish associations corresponding to widely separated tracks**
    - *D(i, j)* is Manhattan distance attenuated by uncertainty $D(i,j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}.$
    - *g(D(i, j); m)* seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with $m = 5$

$$g(x; m) = 1 - e^{-mx},$$

  - Bias per qubit comes from λ term enforcing one track associated with one p-p collision. $\lambda = 1.2 \max(D(i, j))$ is optimal



Track clustering QUBO formulation for D-Wave



Coupling terms

$$H_p = \sum_k^{n_V} \sum_i^{n_T} \sum_{j>i}^{n_T} p_{ik} p_{jk} g(D(i,j); m)$$

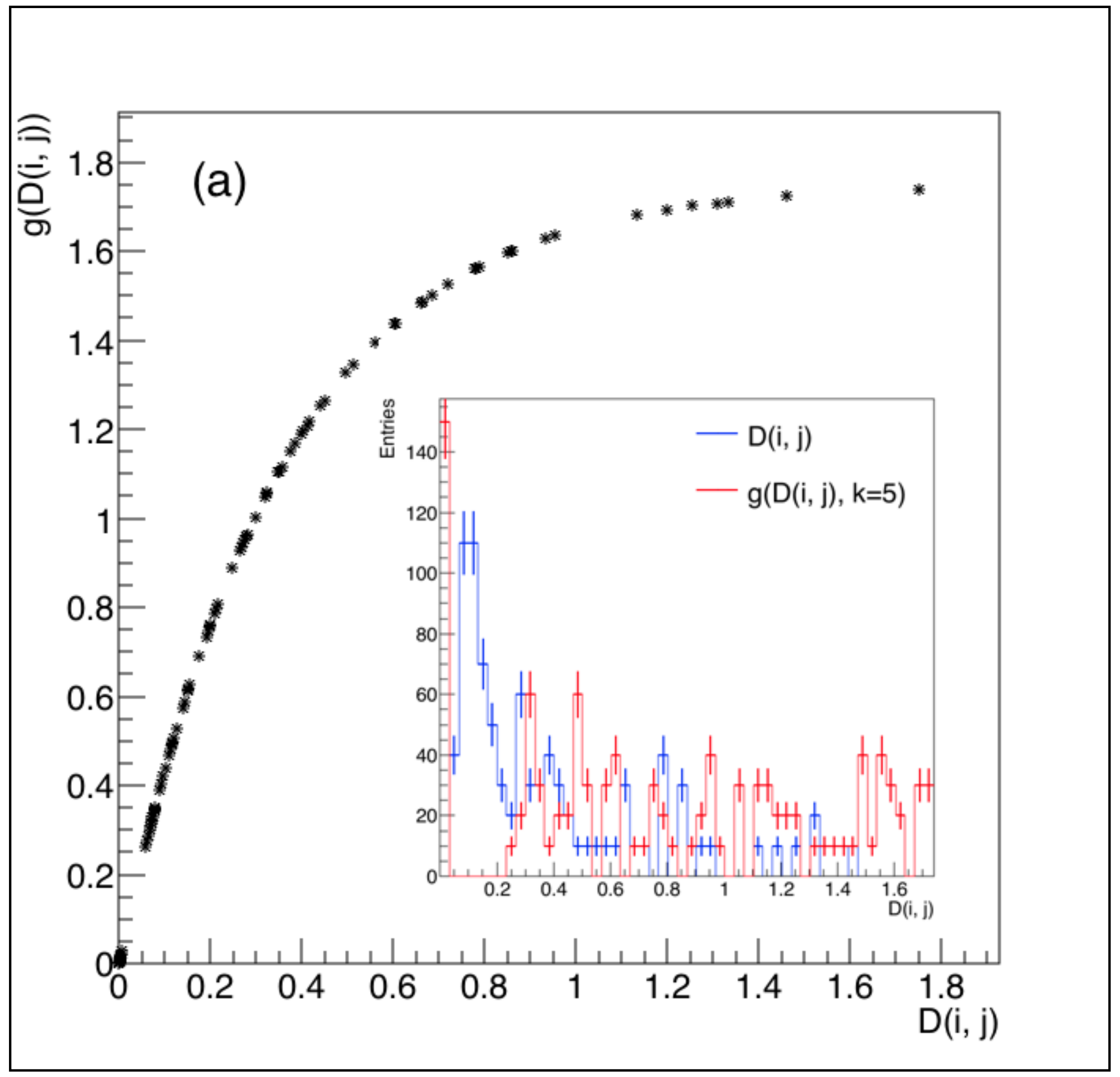
$$+ \lambda \sum_i^{n_T} \left(1 - \sum_k^{n_V} p_{ik}\right)^2 ,$$

Bias terms come from here

- Clustering problem naturally expressed in QUBO form (***V. Kumar, et. al.*** *“Quantum annealing for combinatorial clustering” Quantum Inf. Processing 17 (2018) 39*)
  - ➡ Probability (0 or 1) of i^{th} track to have come from k^{th} p-p collision is p_{ik}. Element p_{ik} **is represented by a qubit**
  - ➡ Coupling between two qubits p_{ik} and p_{jk} that represent association of two tracks to same p-p collision k is a distance measure between the tracks $D(i, j)$. **Punish associations corresponding to widely separated tracks**
    - ➡ $D(i, j)$ is Manhattan distance attenuated by uncertainty $D(i,j) = \frac{|z_i - z_j|}{\sqrt{\delta z_i^2 + \delta z_j^2}}$.
    - ➡ $g(D(i, j); m)$ seeks to distribute the couplings evenly without changing order. Else, lots of small couplings and some large couplings for p-p finding problem. Not for all clustering problems. Empirically seen to improve results with $m = 5$

$$g(x; m) = 1 - e^{-mx},$$

  - ➡ Bias per qubit comes from λ term enforcing one track associated with one p-p collision. $\lambda = 1.2 \max(D(i, j))$ is optimal



- We use D-Wave’s default embedding algorithms

Performance on one event with 3 p-p collision, 15 tracks

Performance on one event with 3 p-p collision, 15 tracks

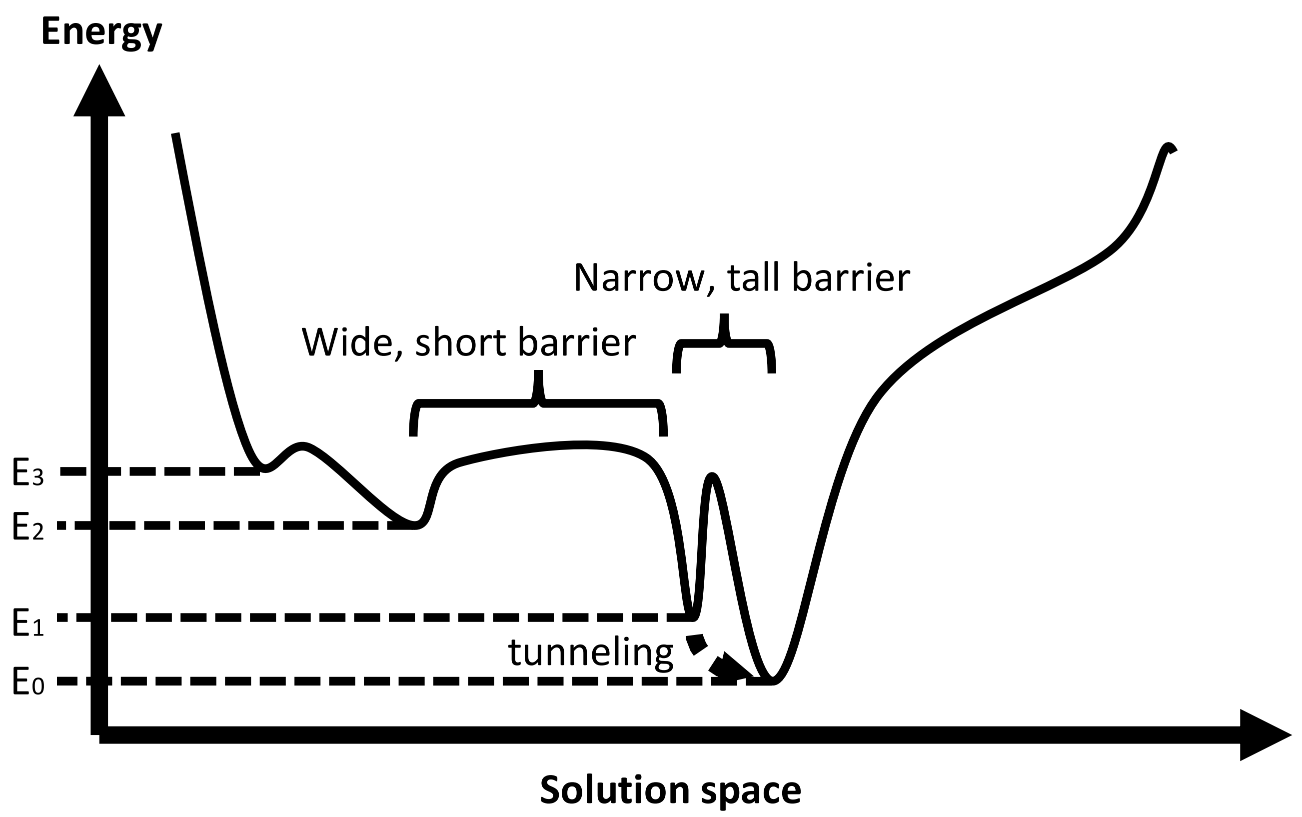
Generating artificial events

- Algorithm tested on artificial events drawn from **measured LHC distributions of collision points** and **measured CMS distributions of tracks**
- **Realistic track reconstruction uncertainties** used. *CMS Collaboration, JINST 9 (2014) P10009*

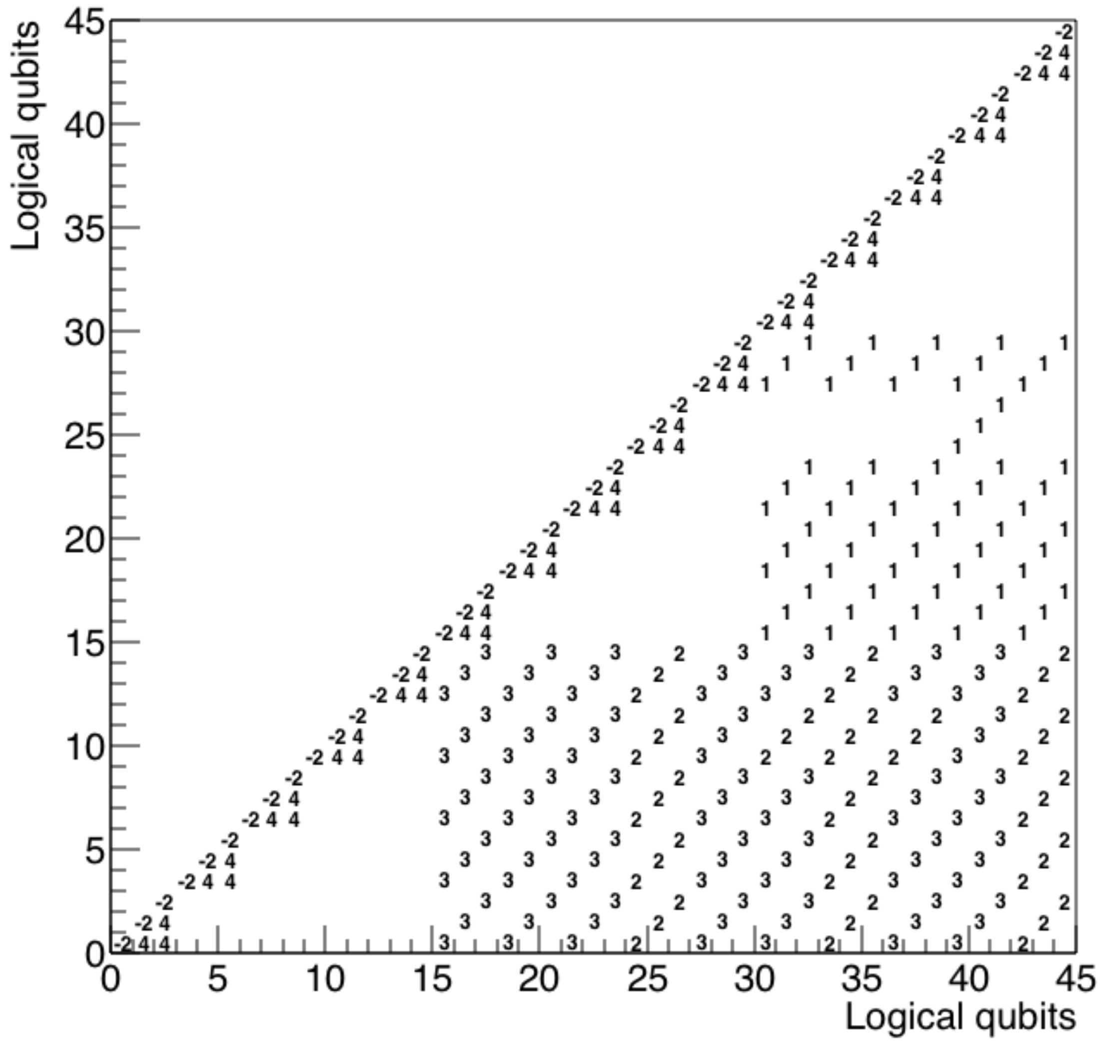
Performance on one event with 3 p-p collision, 15 tracks

Generating artificial events

- Algorithm tested on artificial events drawn from **measured LHC distributions of collision points** and **measured CMS distributions of tracks**
- **Realistic track reconstruction uncertainties** used. *CMS Collaboration, JINST 9 (2014) P10009*



Cartoon of energy profile in solution space. Tunneling is easy through tall narrow barriers. Classically difficult

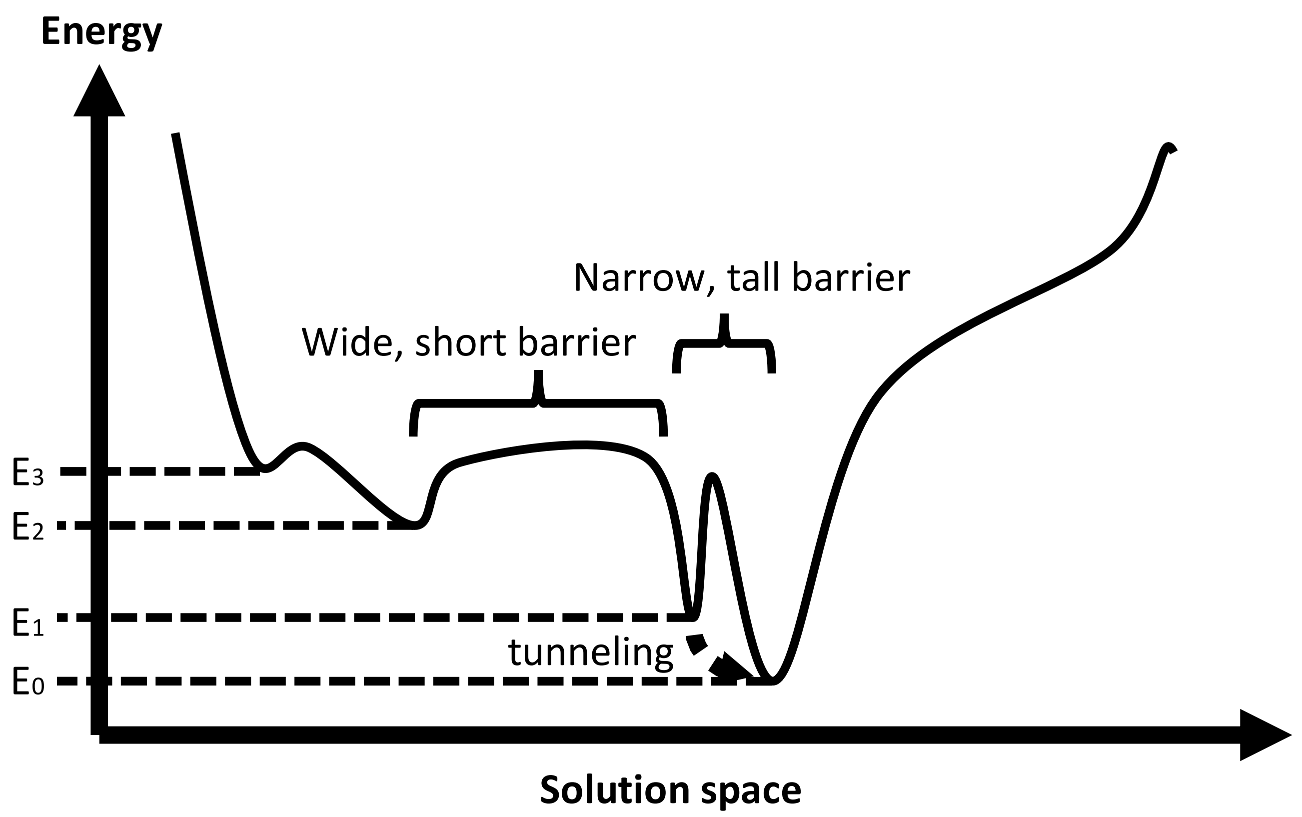


Coefficients of QUBO form used to solve particular event with 3 collisions, 15 tracks

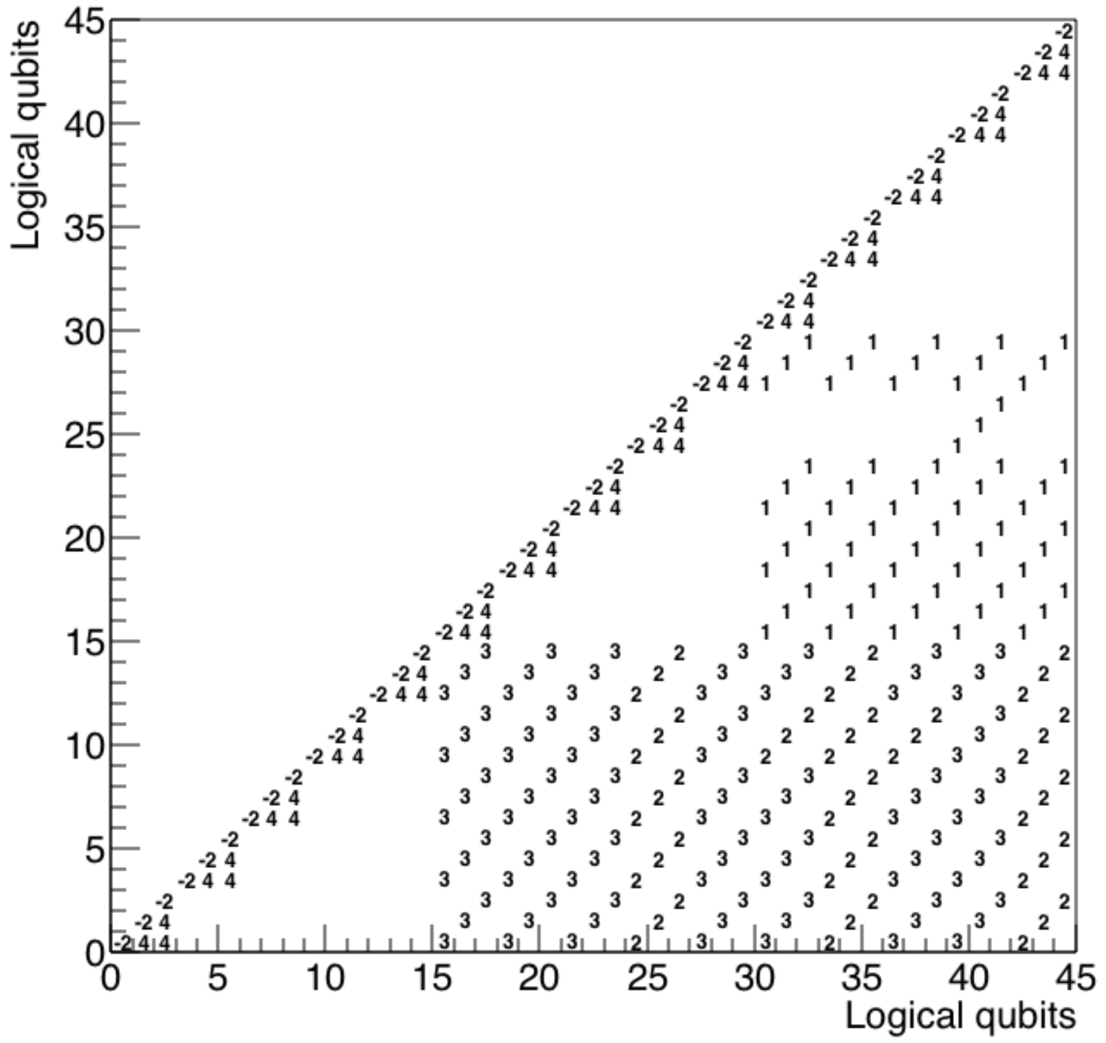
Performance on one event with 3 p-p collision, 15 tracks

Generating artificial events

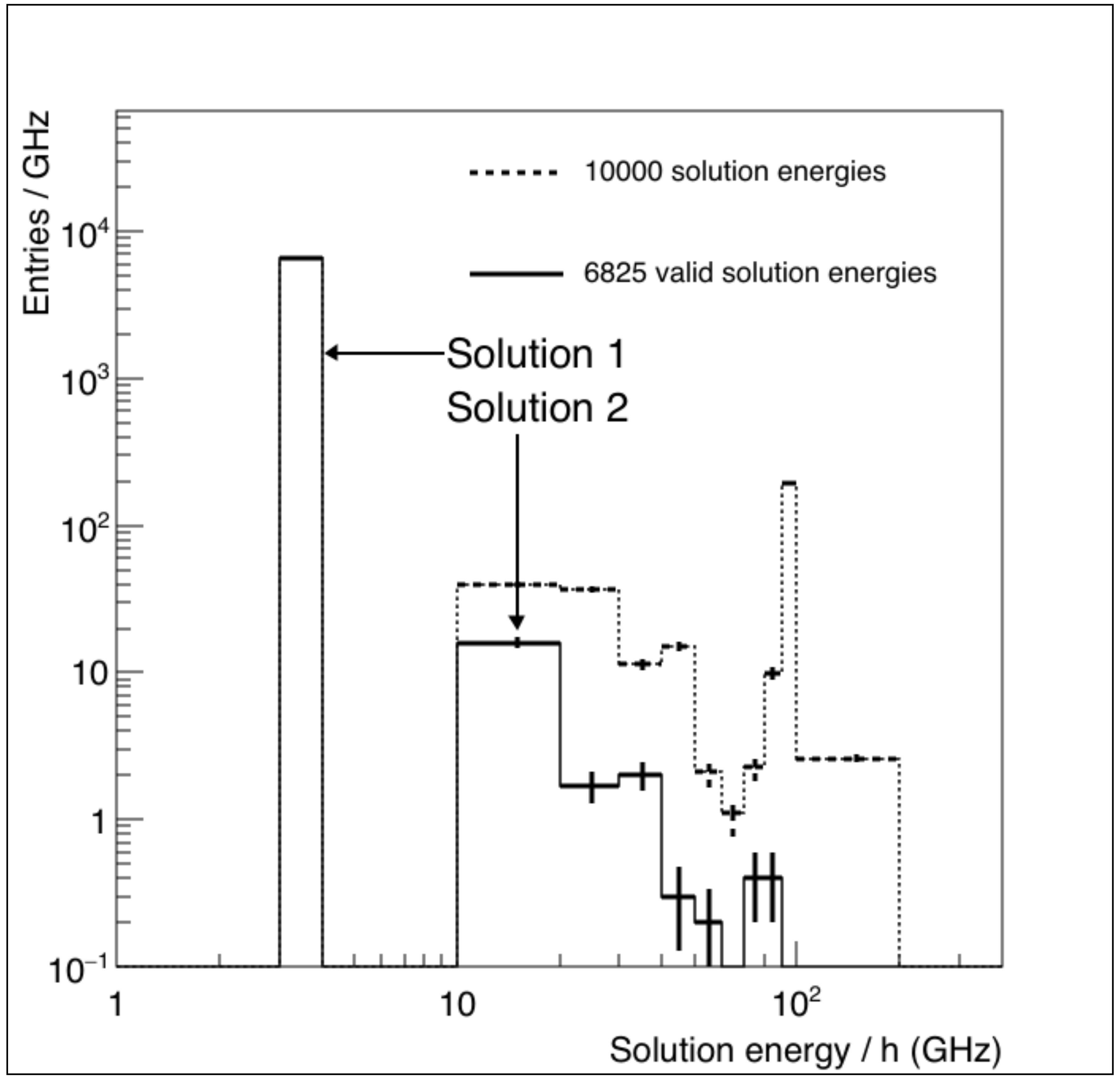
- Algorithm tested on artificial events drawn from **measured LHC distributions of collision points** and **measured CMS distributions of tracks**
- **Realistic track reconstruction uncertainties** used. *CMS Collaboration, JINST 9 (2014) P10009*



Cartoon of energy profile in solution space. Tunneling is easy through tall narrow barriers. Classically difficult



Coefficients of QUBO form used to solve particular event with 3 collisions, 15 tracks



Energy spectrum of solutions for one event with 3 p-p collisions and 15 tracks explored by the D-Wave 2000Q_2_1 with 10,000 samples. Energies corresponding to valid solutions, where p_{ik} add up to 1 for every track, are plotted with solid lines while invalid solutions are plotted with dashed lines. Error bars correspond to statistical uncertainties. The best and next-to-best valid solutions are indicated as Solutions 1 and 2, respectively. For clarity, the histogram is binned by 1 GHz below 10 GHz, by 10 GHz for 10 — 100 GHz, and by 100 GHz above 100 GHz

Performance on one event with 3 p-p collisions and 15 tracks

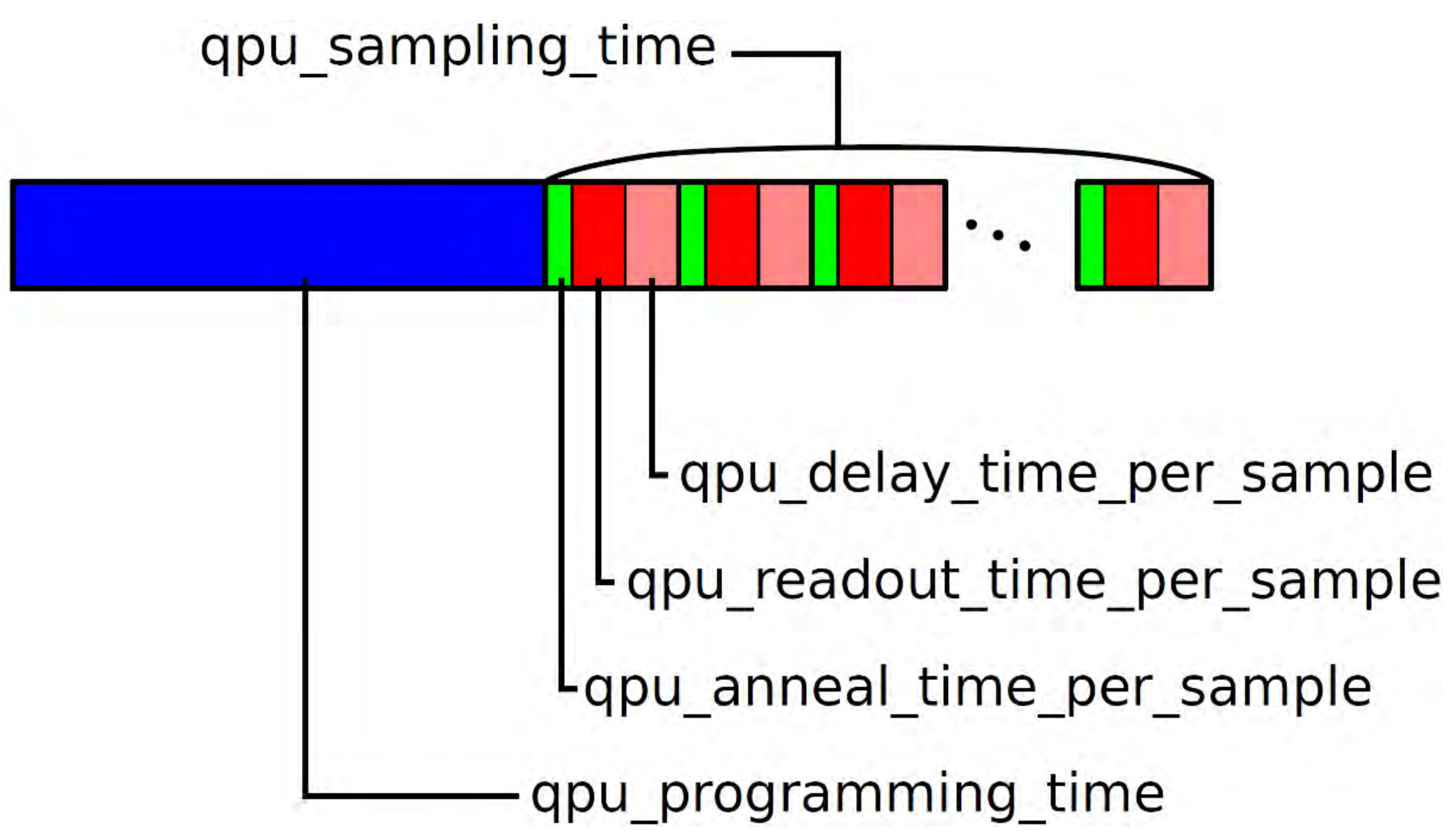
- QUBO bias terms are equal and come from the λ constraint
- Quantum state prepared and annealed 10,000 times. **DW_2000Q_2_1** used
  - 6,825 solutions are valid, i.e. λ constraint is strictly met ($\Sigma_k\, p_{ik} = 1$ for all tracks)
  - 6,615 solutions are correct (Solution 1). **Convergence efficiency = 66%**
  - Small number of valid secondary solutions where one track has been misassociated

Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency

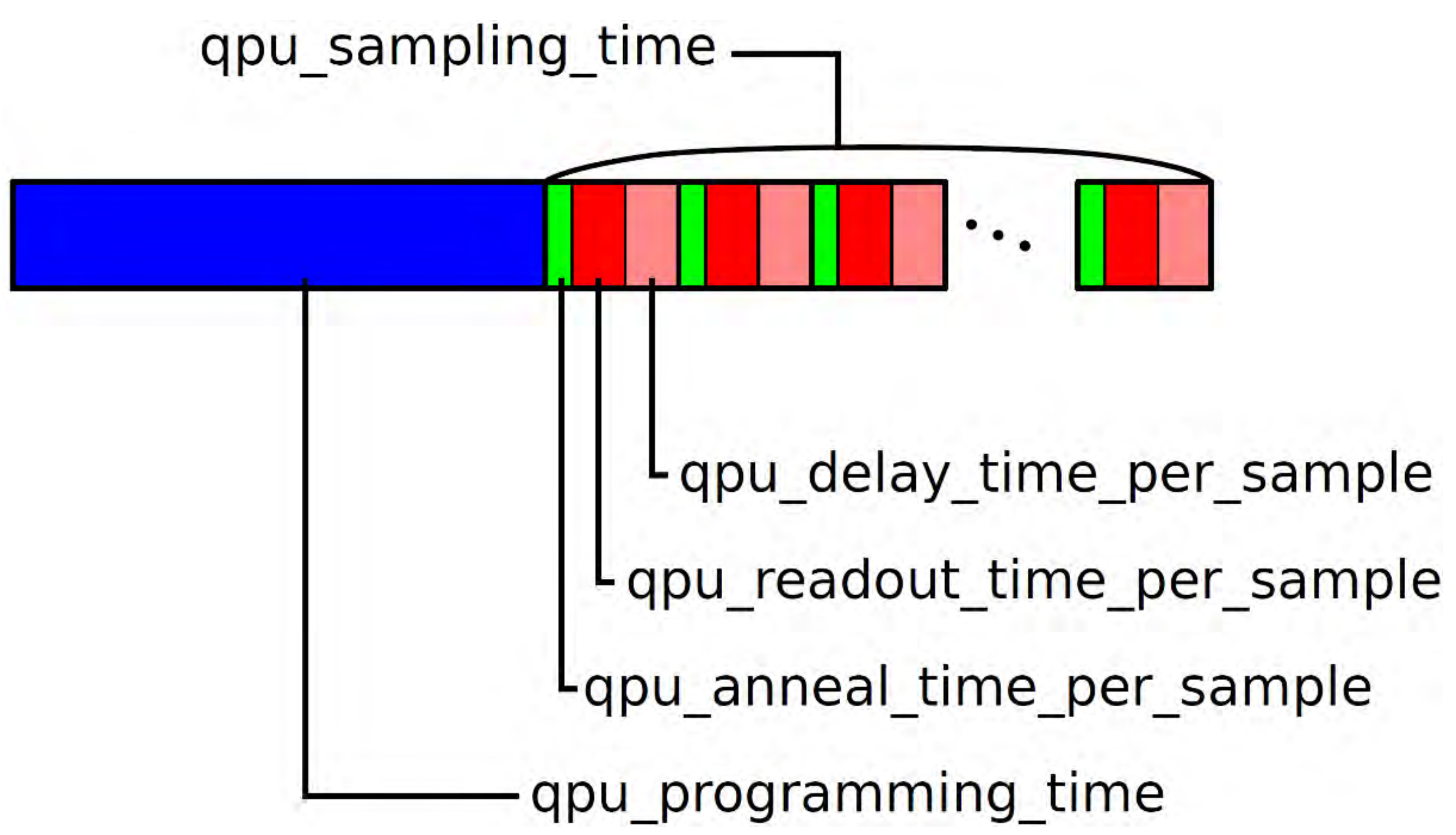


Sampling time of D-Wave 2000Q_2_1

- Total sample time = **164 µs**
  - Anneal time = 20 µs
  - Readout time = 123 µs
  - Delay time = 21 µs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency



Sampling time of D-Wave 2000Q_2_1

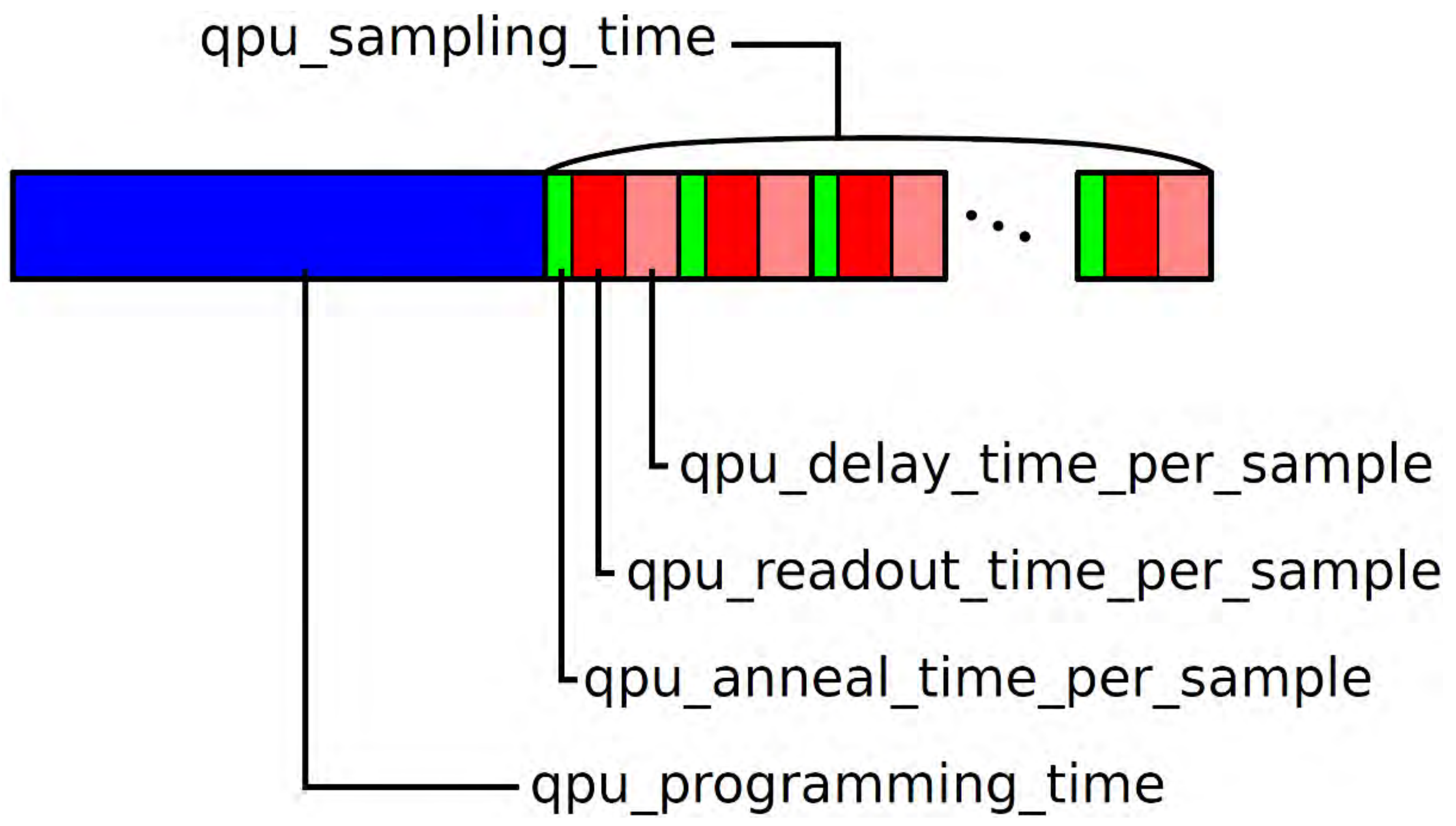
- Total sample time = **164 µs**
  - Anneal time = 20 µs
  - Readout time = 123 µs
  - Delay time = 21 µs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

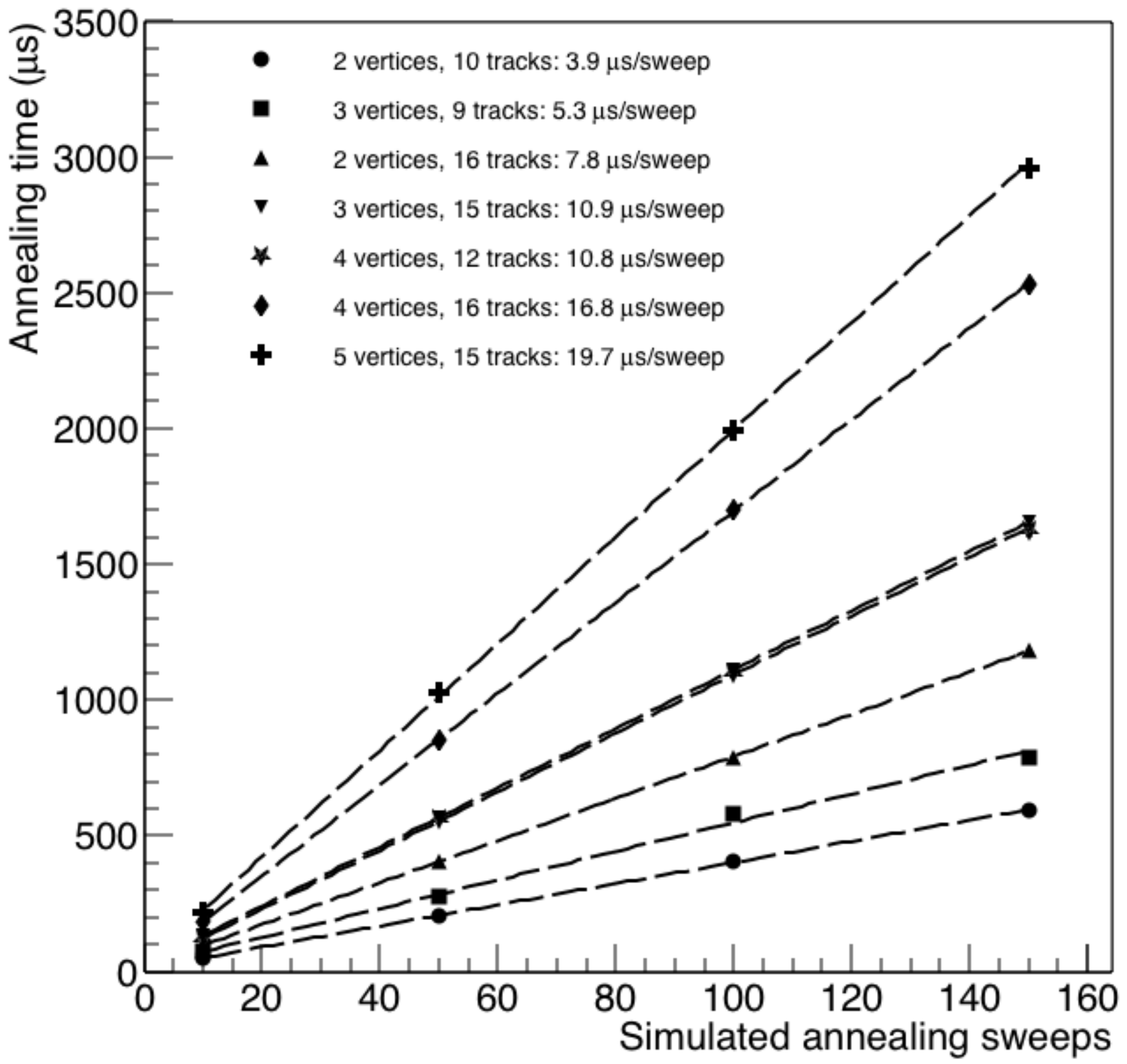
CPU: 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)
Algorithm: Simulated annealing. Time optimized
- Use a sorted `std::map` with keys = bit index, value = list of other bits it couples to and the coupling
- Bit flip only requires to compute energy difference

Compiler: C++, -O2 optimization

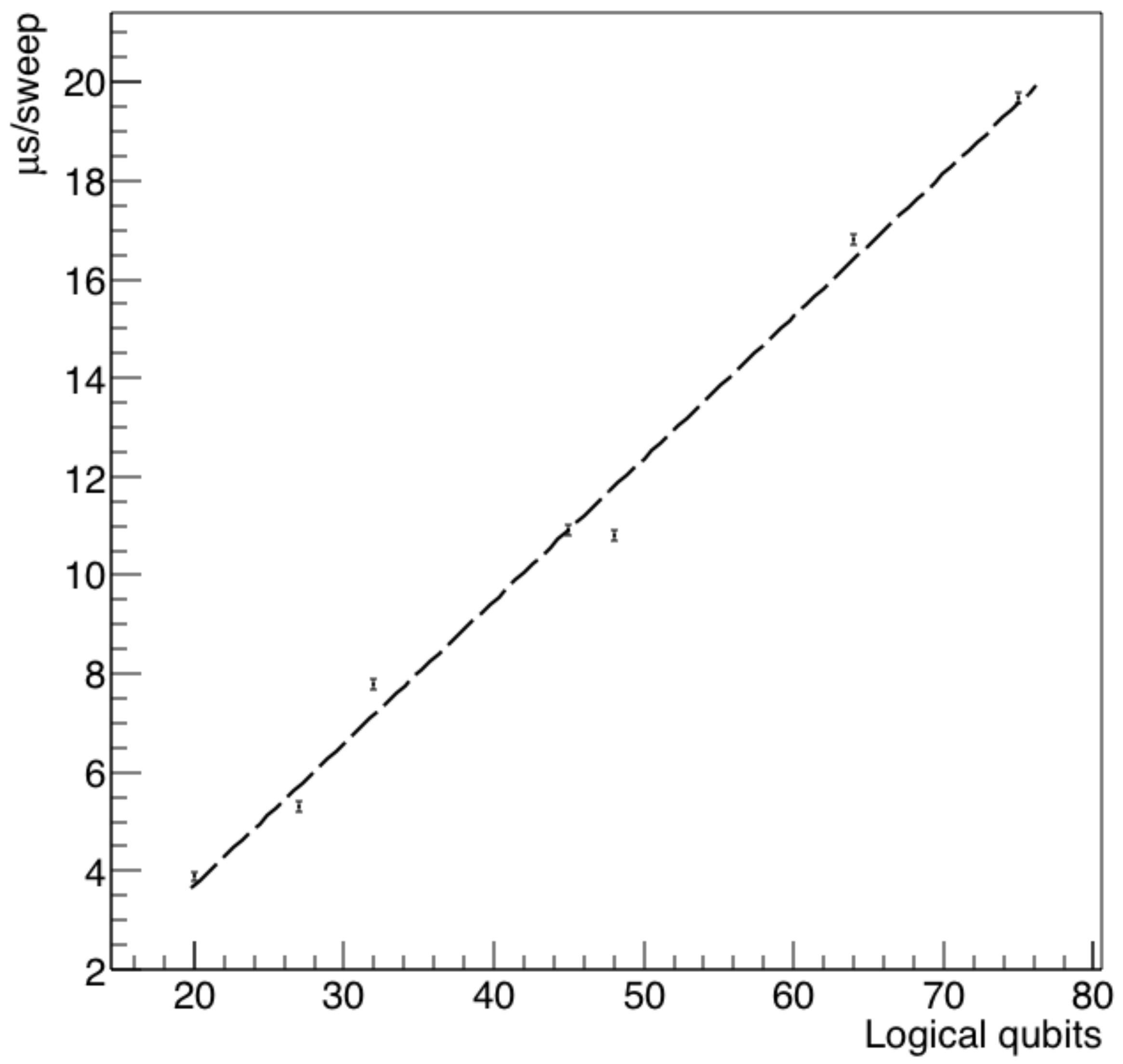
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency





CPU process time against number of sweeps for various event topologies under consideration



Time per sweep scales linearly with number of bits involved

Sampling time of D-Wave 2000Q_2_1

- Total sample time = **164 µs**
  - Anneal time = 20 µs
  - Readout time = 123 µs
  - Delay time = 21 µs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

CPU: 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)

Algorithm: Simulated annealing. Time optimized

- Use a sorted `std::map` with keys = bit index, value = list of other bits it couples to and the coupling
- Bit flip only requires to compute energy difference

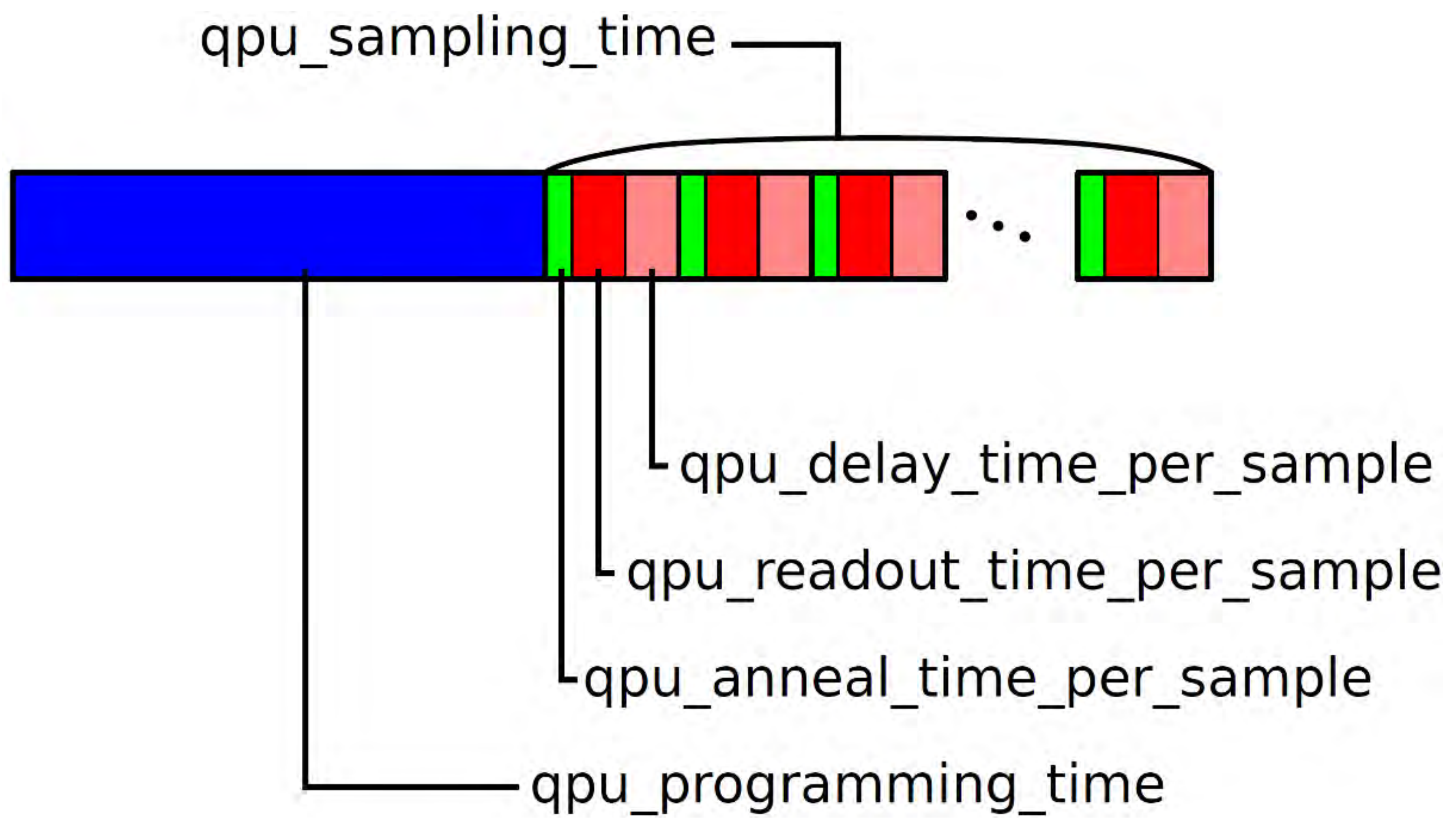
Compiler: C++, -O2 optimization

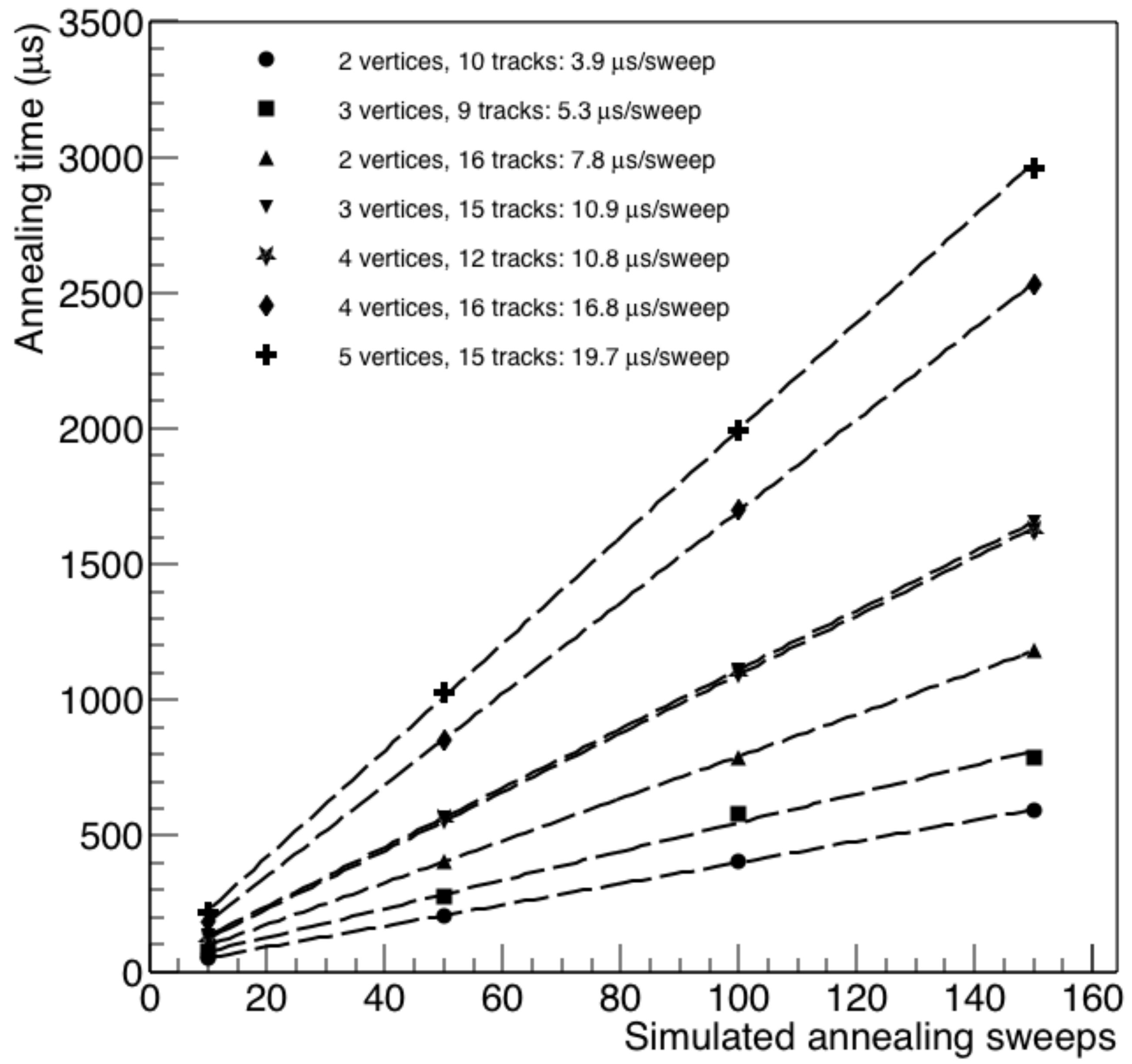
Estimating CPU time per sweep

- Measure **process time**, not wall time
- Plot **process time against nSweeps** for different event topologies
- Discard overhead. **Slope** is time per sweep.
  - For 3 collision 15 tracks, 10.9 µs/ sweep. Thus, 15 sweeps would fit in D-Wave's sampling time

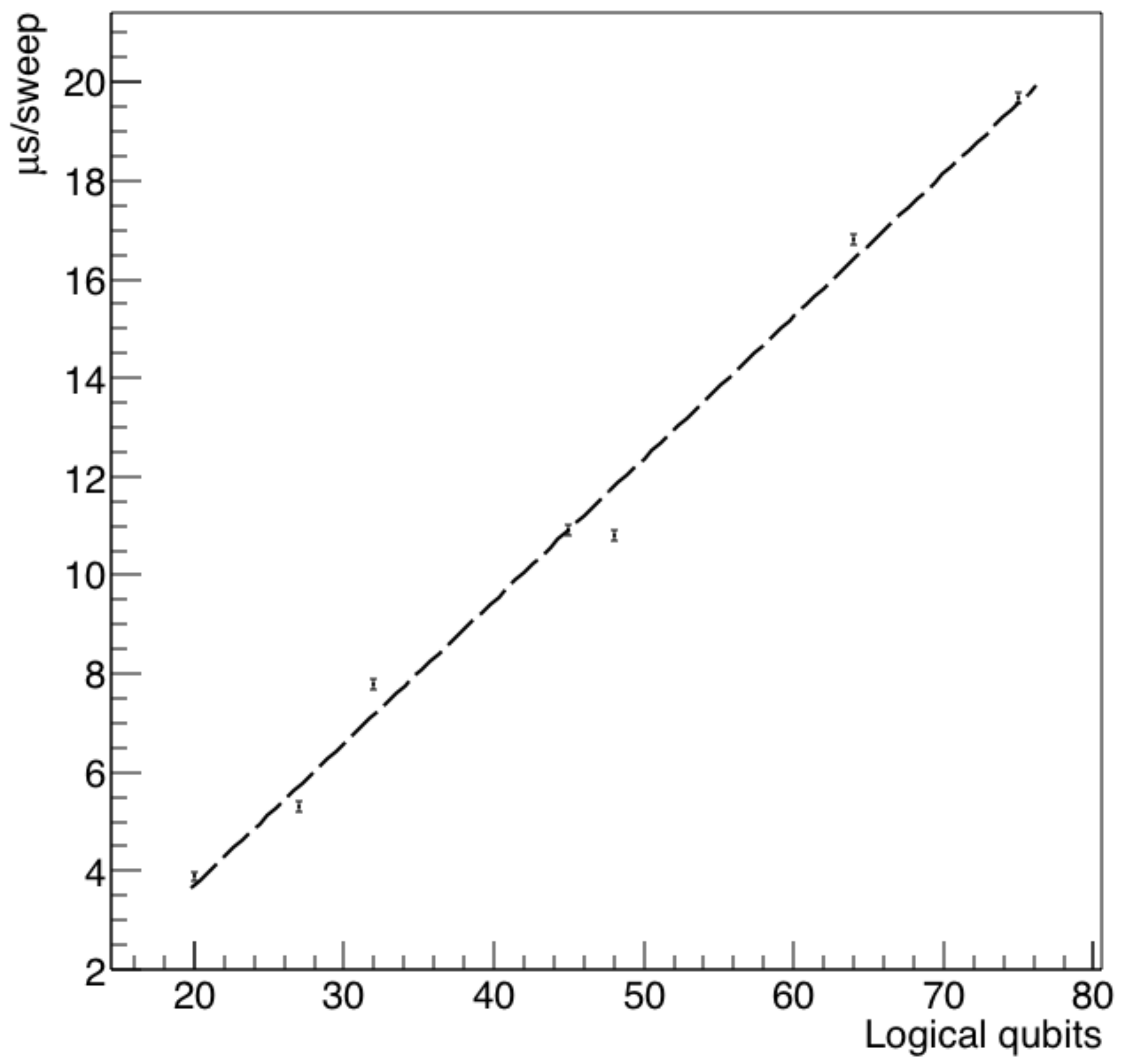
Benchmarking against simulated annealing in equal time

Principle: Equalize working time between CPU and QPU, and compare convergence efficiency





CPU process time against number of sweeps for various event topologies under consideration



Time per sweep scales linearly with number of bits involved

Sampling time of D-Wave 2000Q_2_1

- Total sample time = **164 μs**
  - Anneal time = 20 μs
  - Readout time = 123 μs
  - Delay time = 21 μs
- How many Simulated Annealing sweeps can we fit in this?
  - Depends on problem complexity

CPU: 3.1 GHz Intel Core i7-5557U (MacBook pro 2017)
Algorithm: Simulated annealing. Time optimized
- Use a sorted `std::map` with keys = bit index, value = list of other bits it couples to and the coupling
- Bit flip only requires to compute energy difference

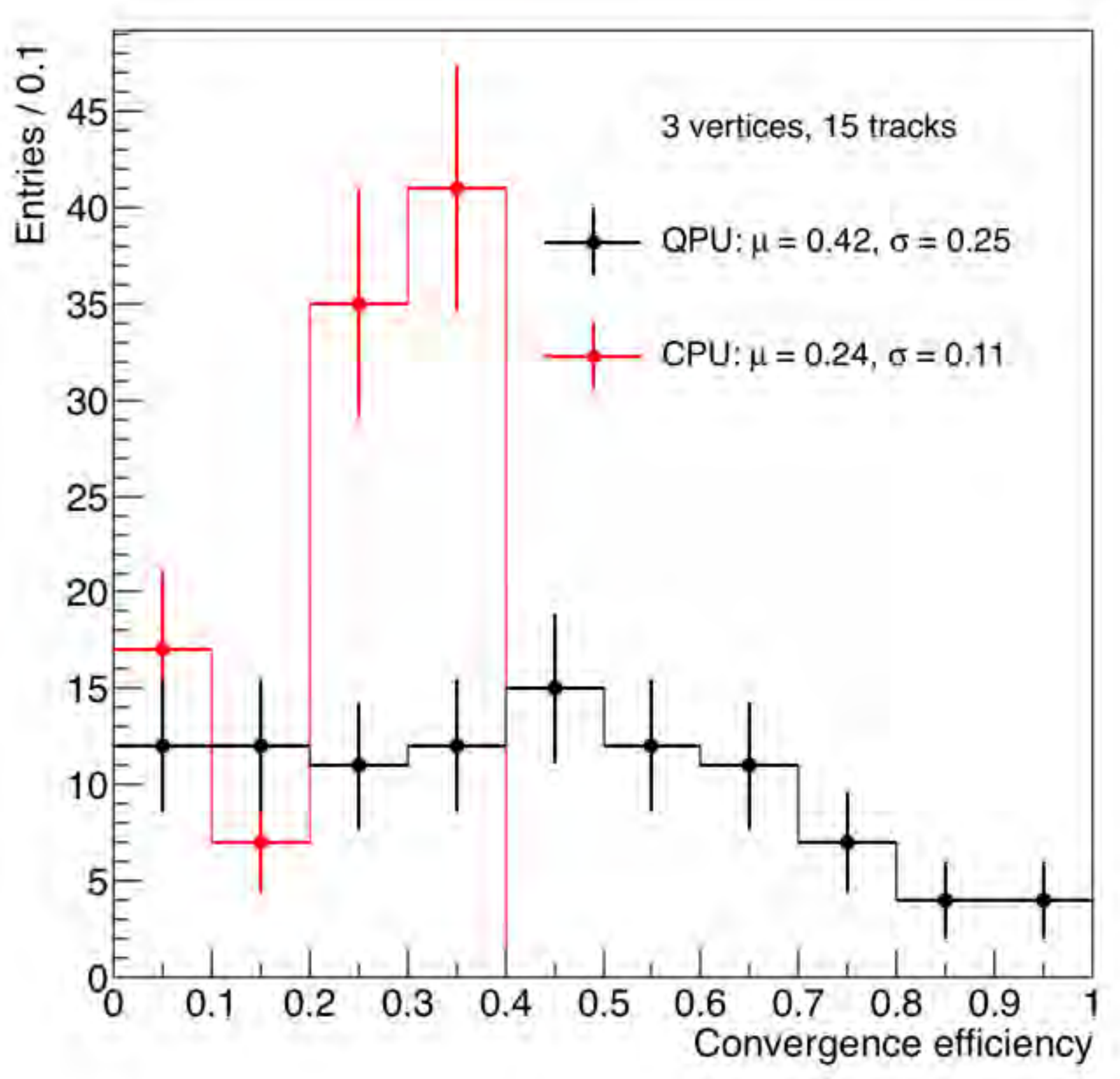
Compiler: C++, -O2 optimization

Estimating CPU time per sweep

- Measure **process time**, not wall time
- Plot **process time against nSweeps** for different event topologies
- Discard overhead. **Slope** is time per sweep.
  - For 3 collision 15 tracks, 10.9 μs/sweep. Thus, 15 sweeps would fit in D-Wave's sampling time

Simulated annealing on CPU is only allowed as many iterations between β_{init} = 0.1 and β_{final} = 10 as would fit 164 μs

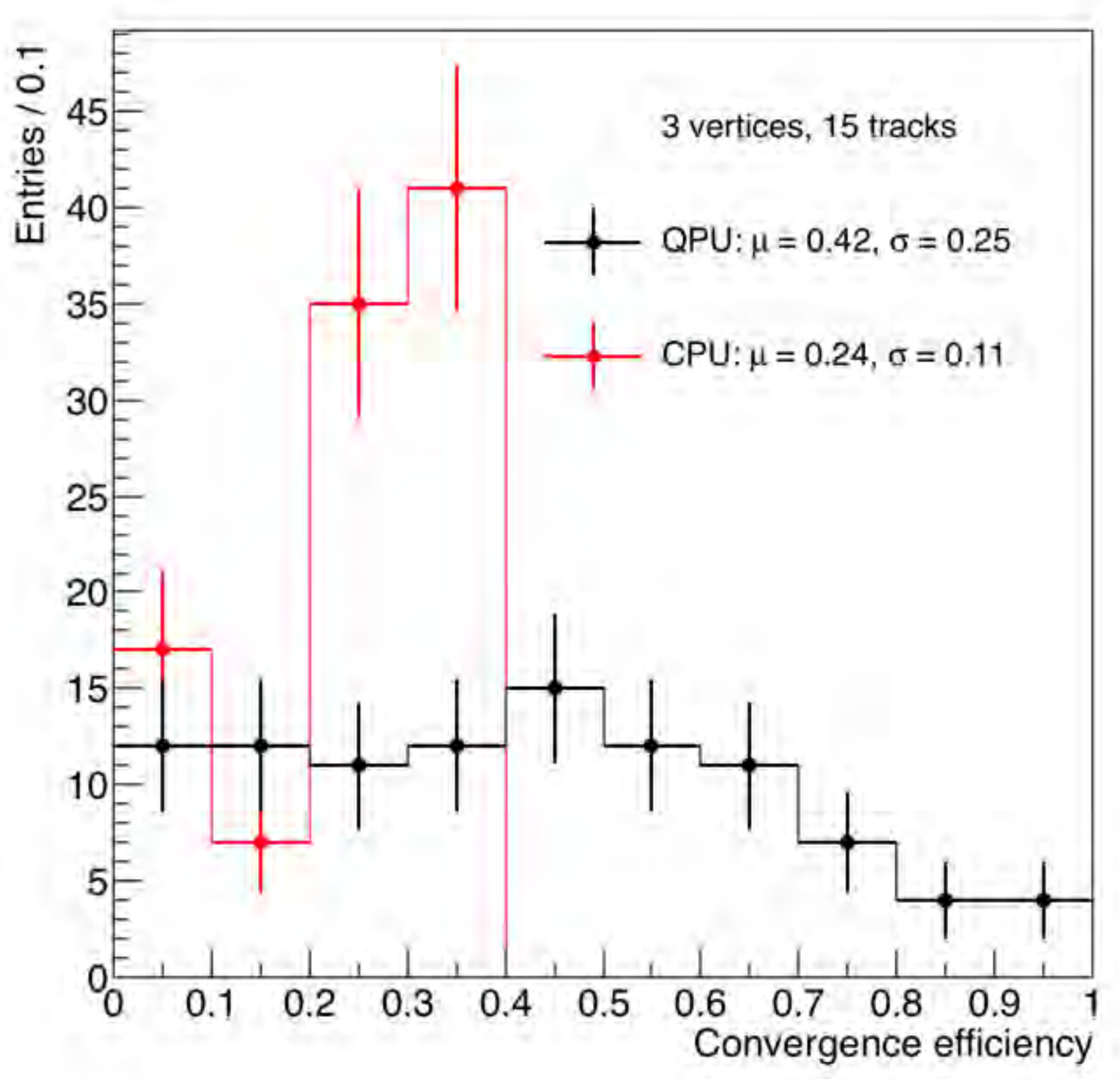
Performance on 100 events with 3 p-p collisions and 15 tracks



Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with *collisions spread closely compared to the spread of their tracks are hard* for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Performance on 100 events with 3 p-p collisions and 15 tracks

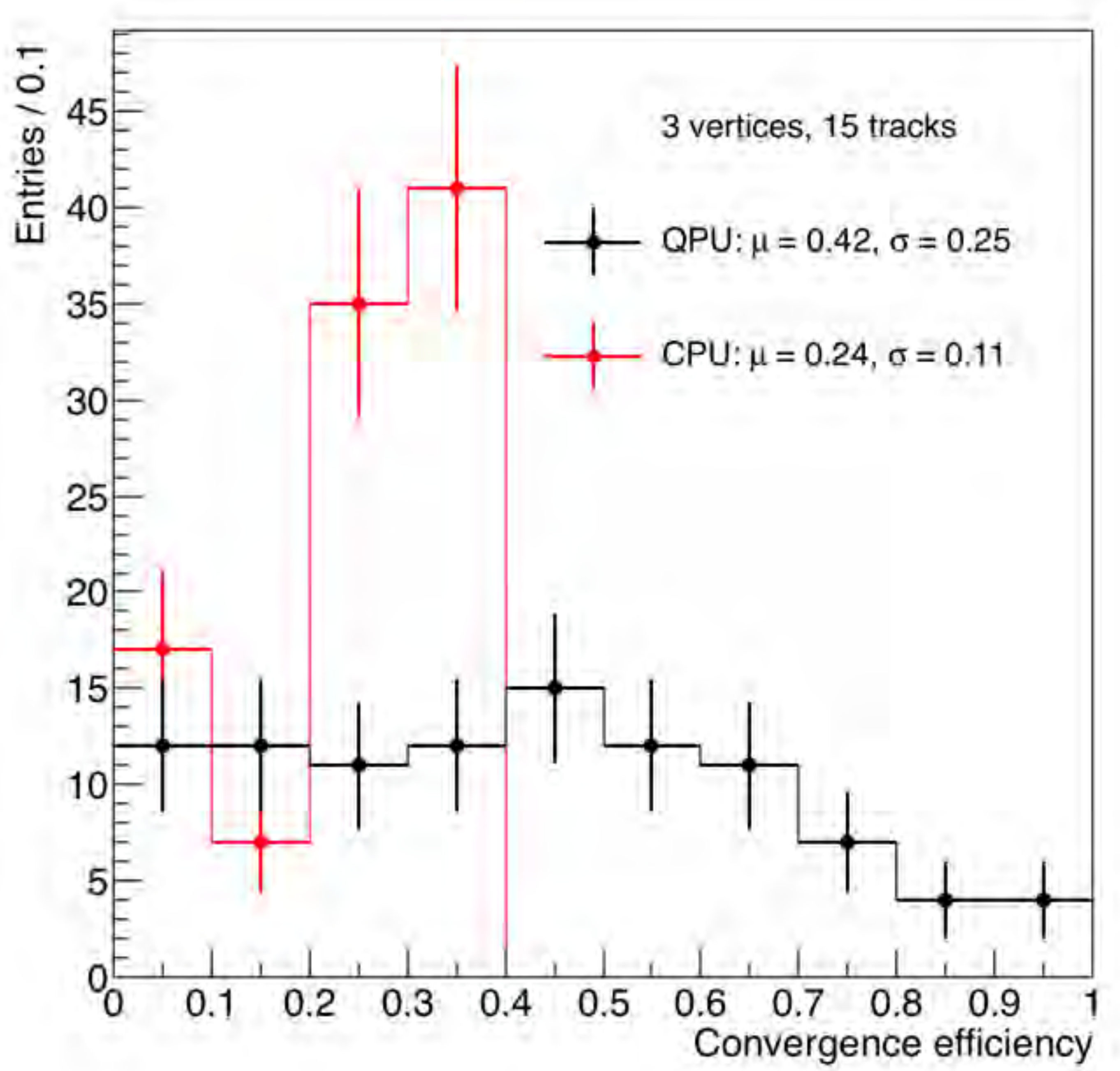


Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with *collisions spread closely compared to the spread of their tracks are hard* for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?

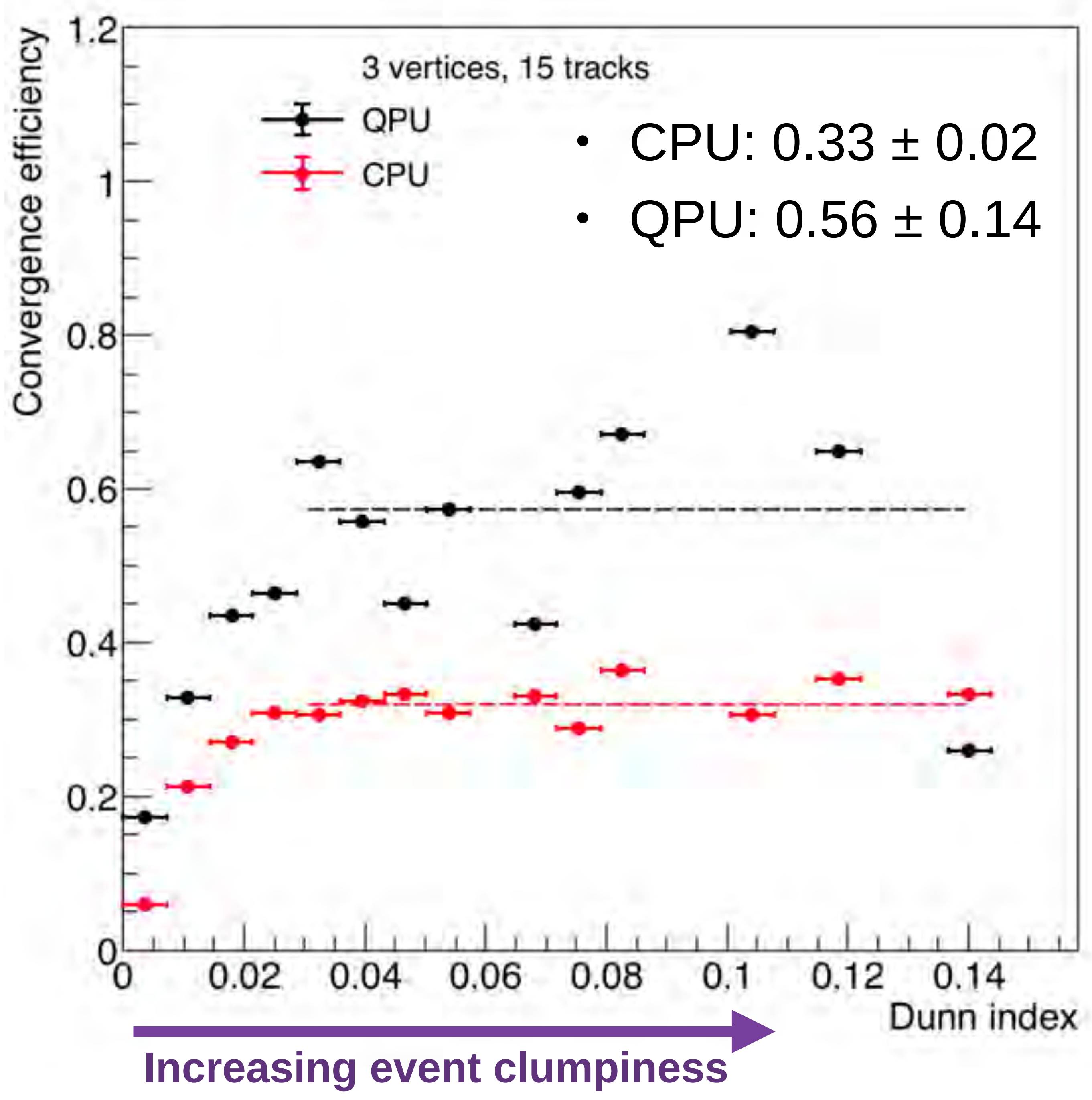
Performance on 100 events with 3 p-p collisions and 15 tracks



Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with *collisions spread closely compared to the spread of their tracks are hard* for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?



- CPU: 0.33 ± 0.02
- QPU: 0.56 ± 0.14

Performance against event "clumpiness"

- A measure of event clumpiness is the Dunn index. Low Dunn index = diffuse event, high = clumpy event

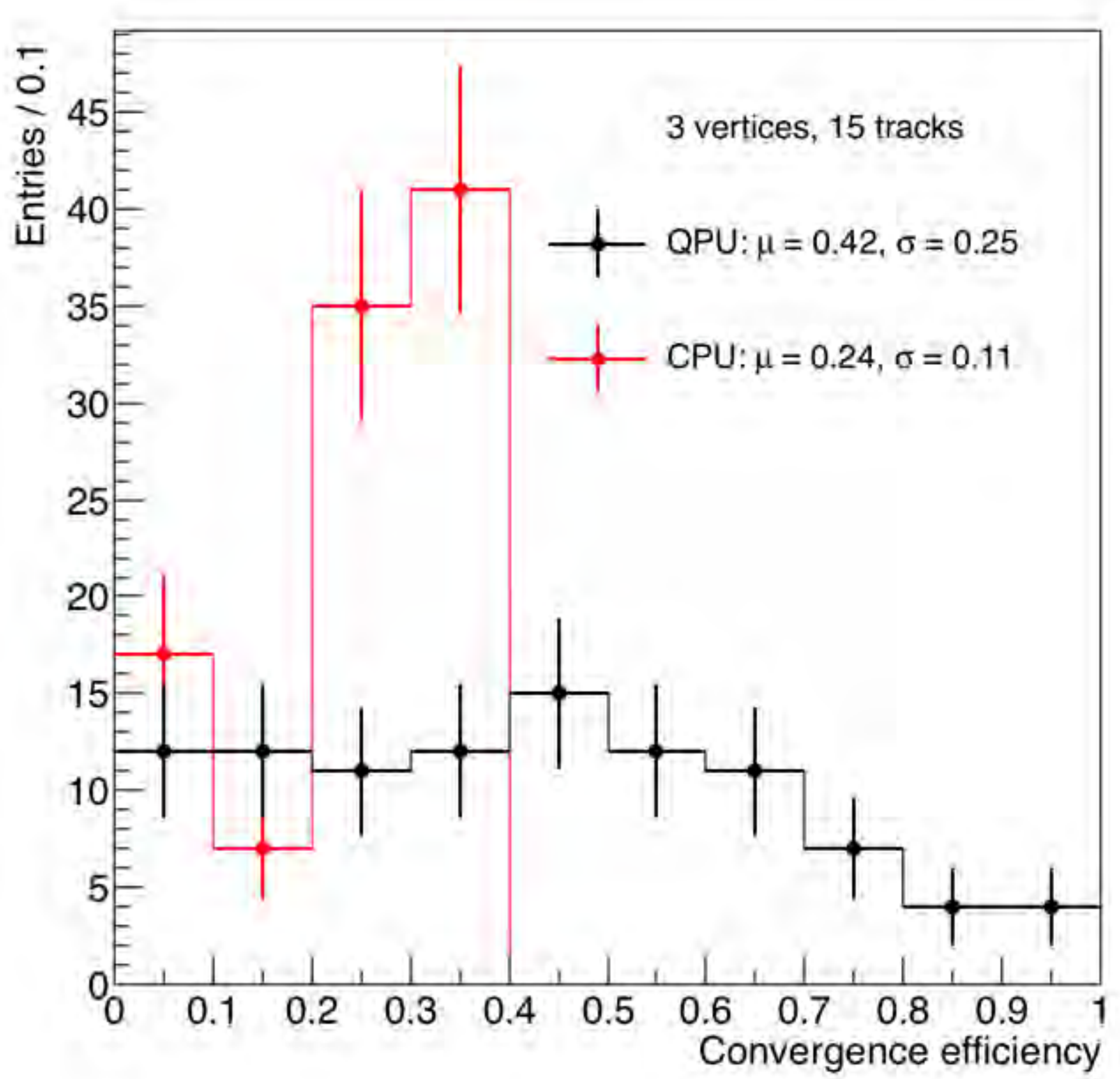
$$D = \frac{\min_{1 \leq i \leq j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)}$$

numerator = minimum inter-cluster distance

denominator = maximum intra-cluster distance between tracks

- Convergence efficiency is plotted as a function of Dunn index. Shows expected structure

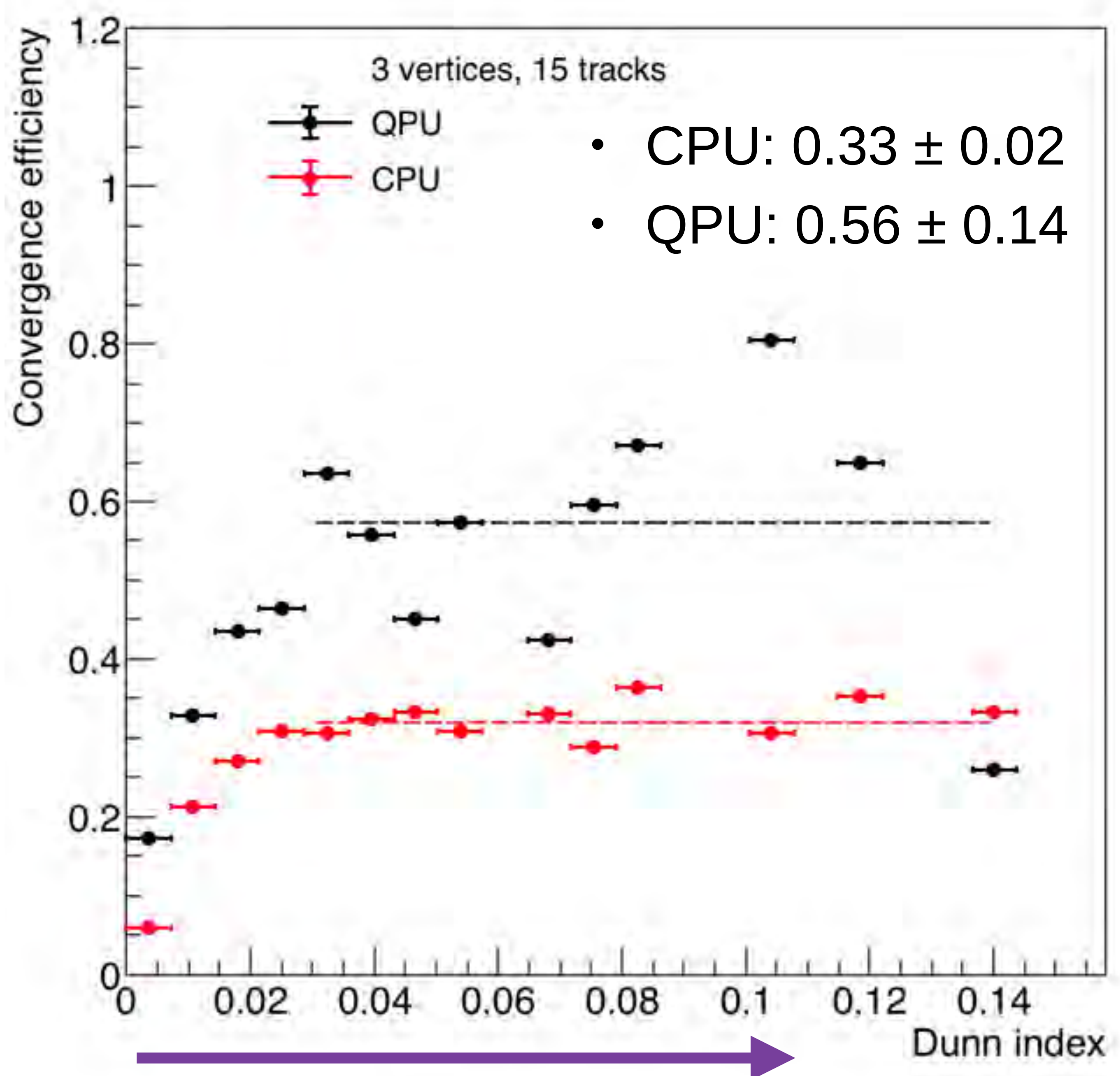
Performance on 100 events with 3 p-p collisions and 15 tracks



Performance on an ensemble of events

- 100 events with 3 p-p collisions and 15 tracks are thrown from measured CMS distributions
  - Each event is sampled 10,000 times by both the QPU and the CPU (in equivalent time)
- Events with *collisions spread closely compared to the spread of their tracks are hard* for both QPU and CPU to solve
- A distribution of convergence efficiencies is observed
  - QPU: mean = 42%, std. dev. = 25%
  - CPU: mean = 24%, std. dev. = 11%

Is there underlying structure to these distributions?



Increasing event clumpiness

Performance against event "clumpiness"

- A measure of event clumpiness is the Dunn index. Low Dunn index = diffuse event, high = clumpy event

$$D = \frac{\min_{1 \le i < j \le n} d(i,j)}{\max_{1 \le k \le n} d'(k)}$$

numerator = minimum inter-cluster distance

denominator = maximum intra-cluster distance between tracks

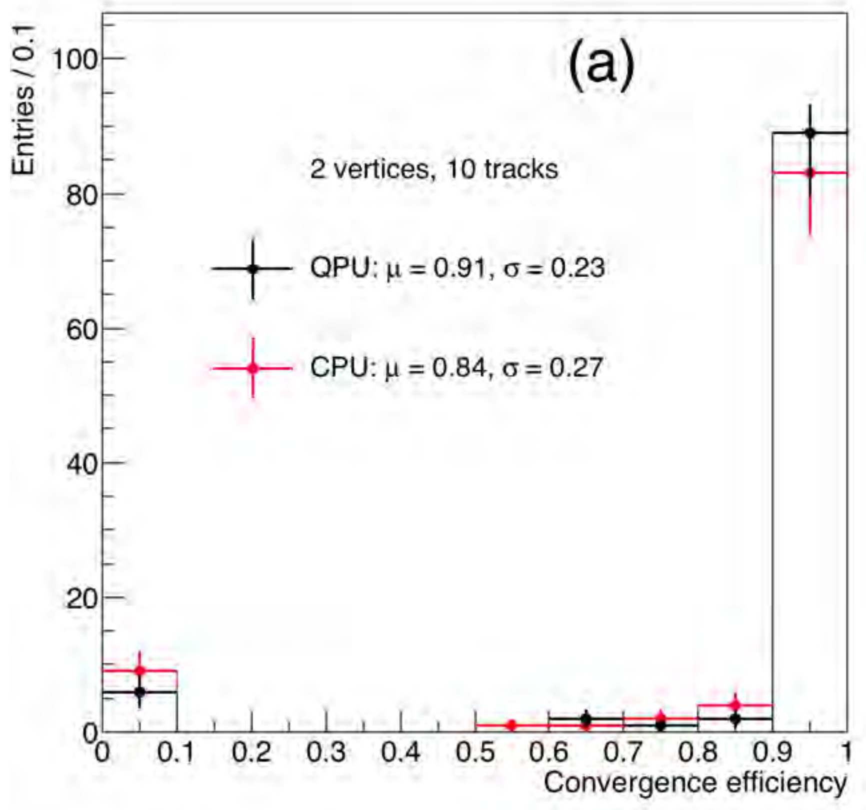
- Convergence efficiency is plotted as a function of Dunn index. Shows expected structure

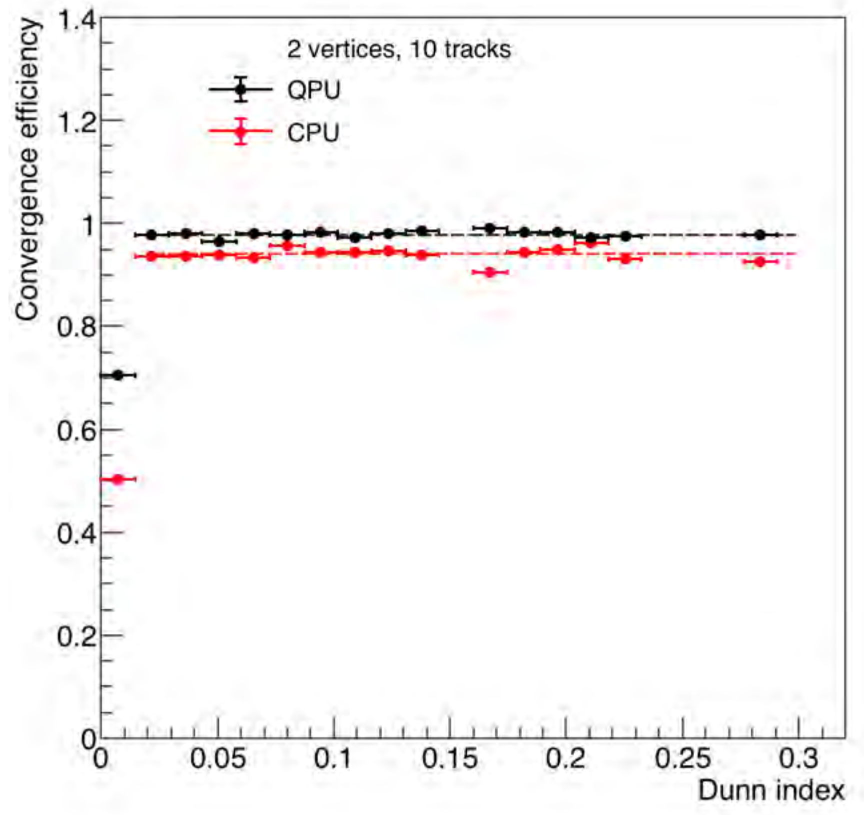
Convergence efficiency increases with event clumpiness. QPU beats CPU in efficiency for same running time

QPU vs CPU scaling with event complexity

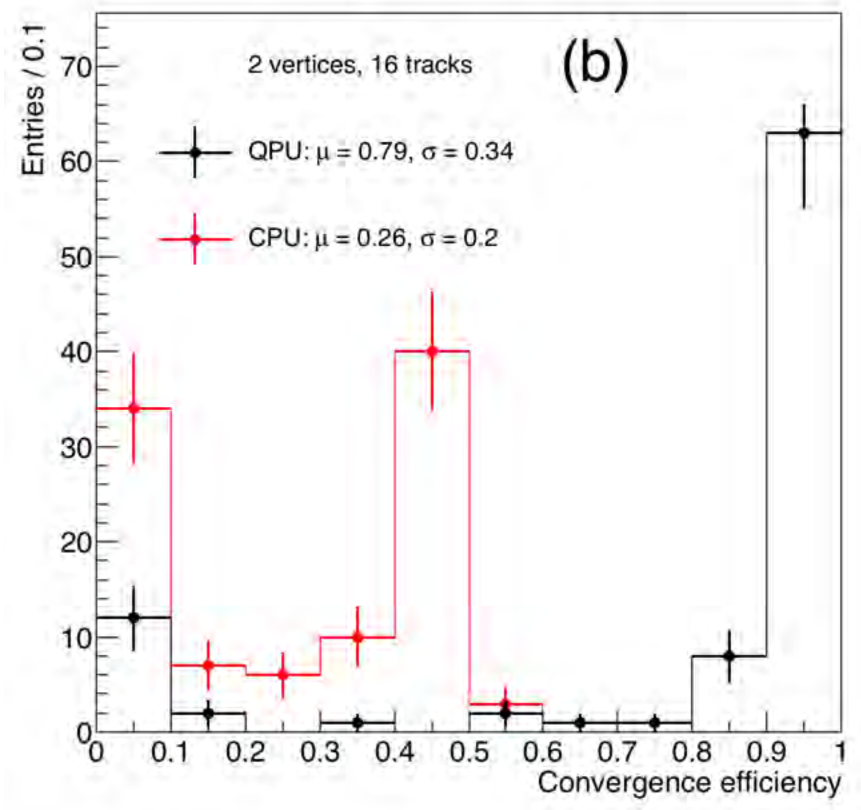
We scan over event topologies with increasing complexity

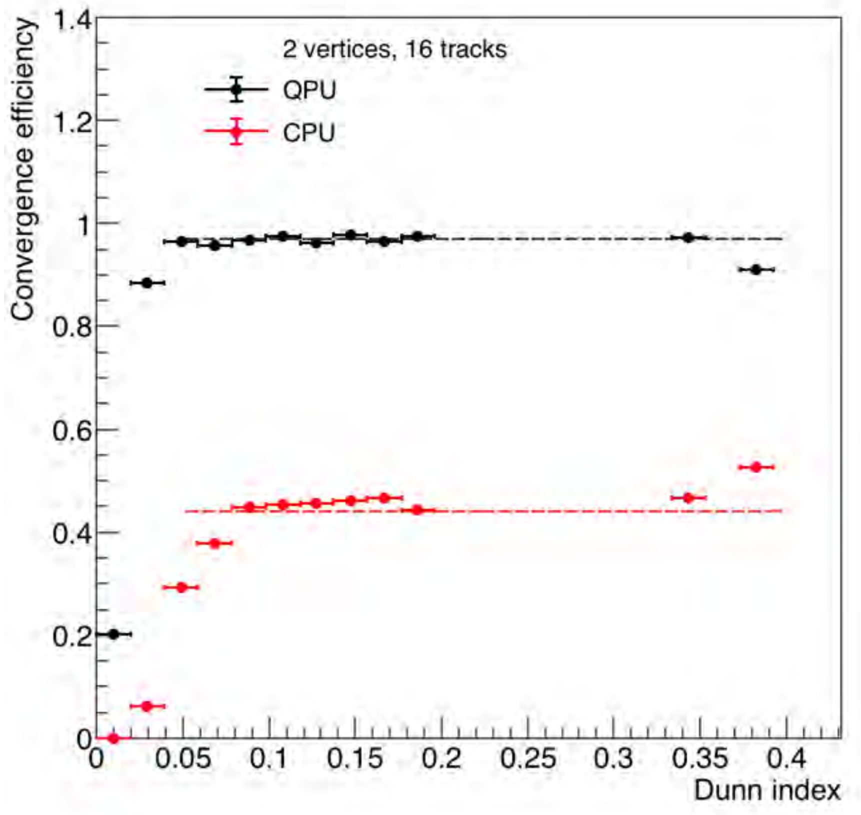
2 collisions, 10 tracks



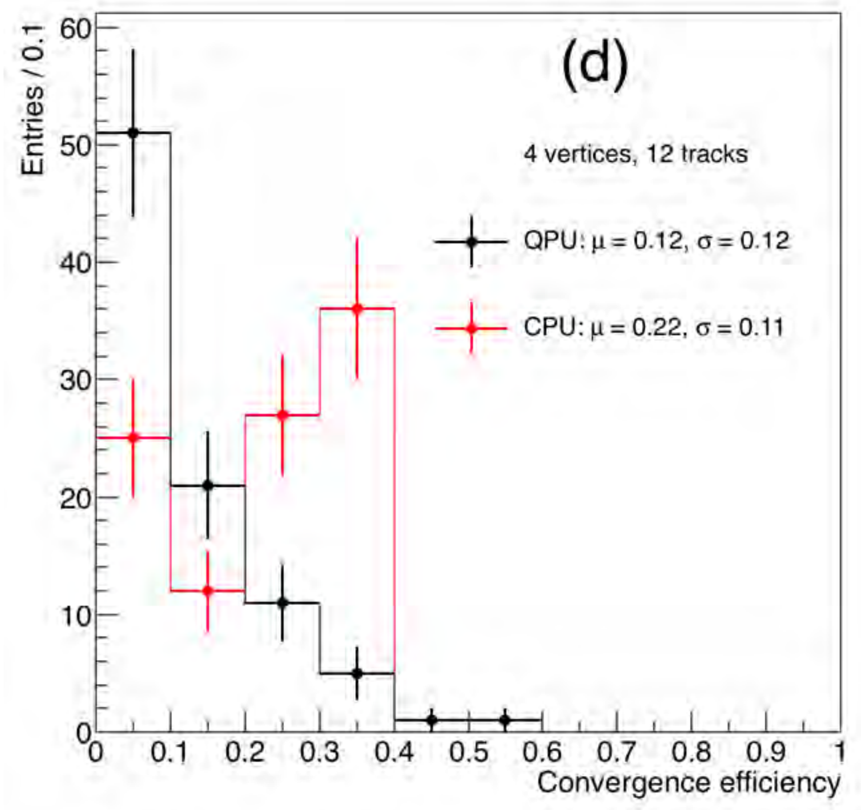


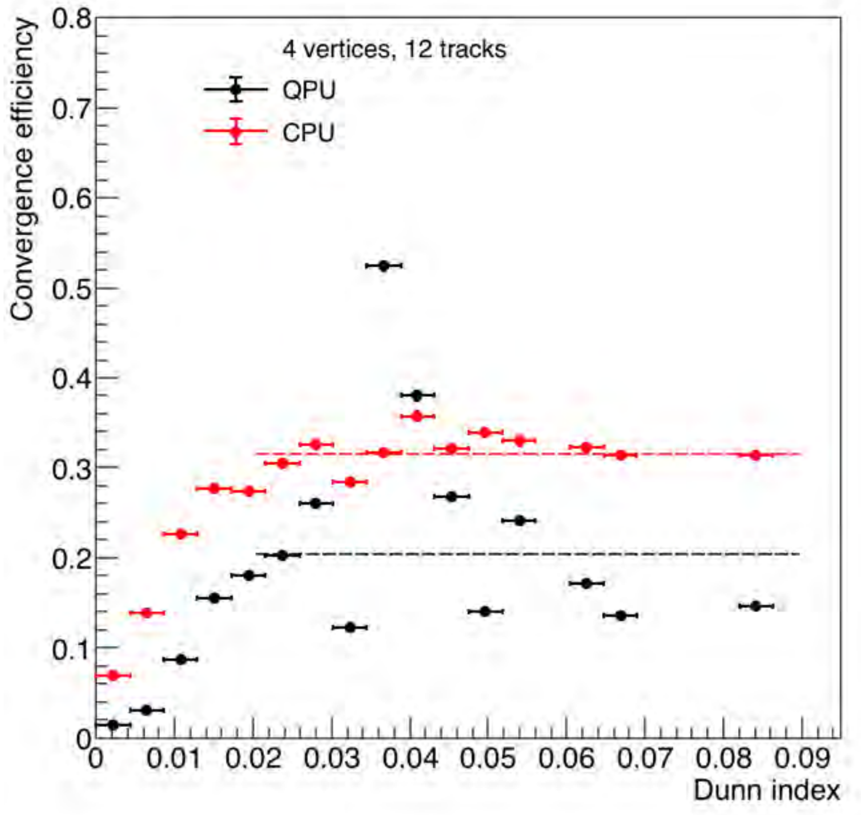
2 collisions, 16 tracks



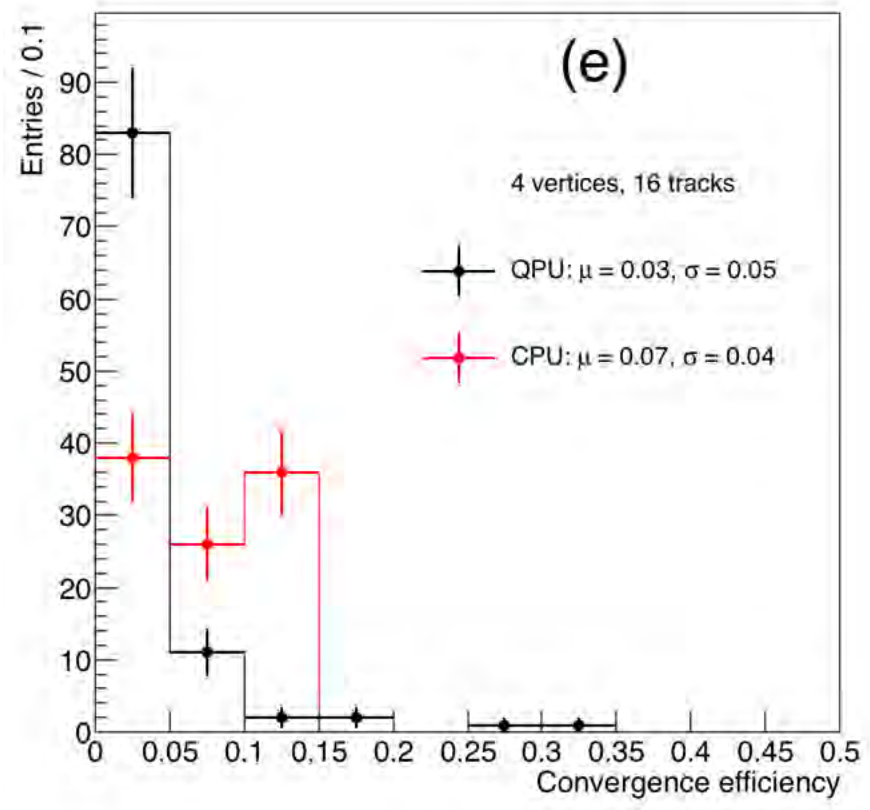


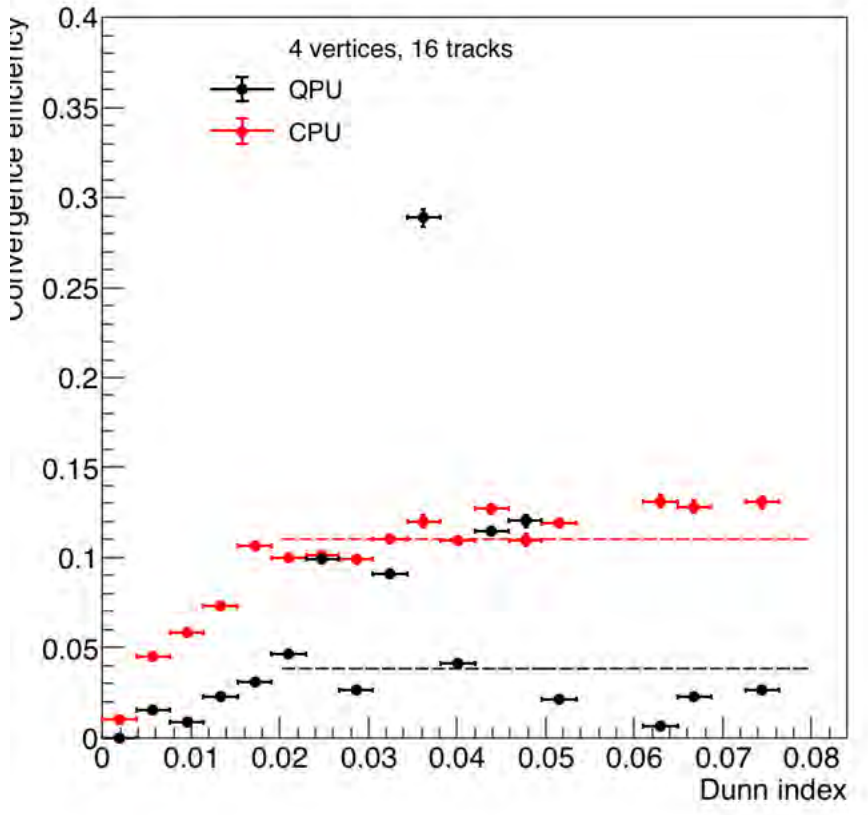
4 collisions, 12 tracks





4 collisions, 16 tracks





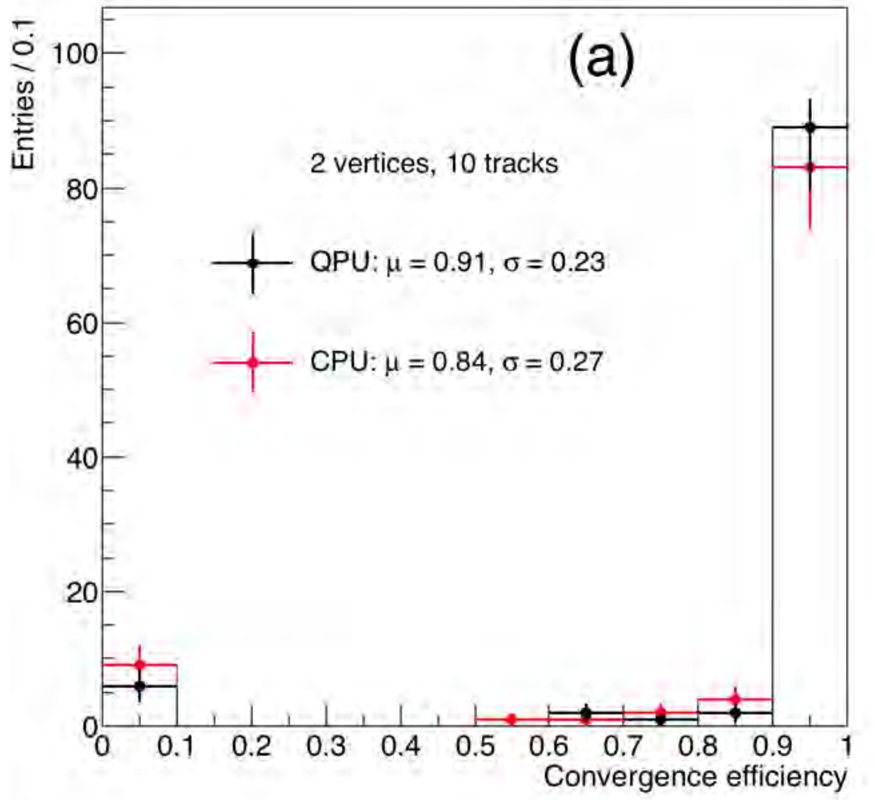
Maximum convergence efficiencies

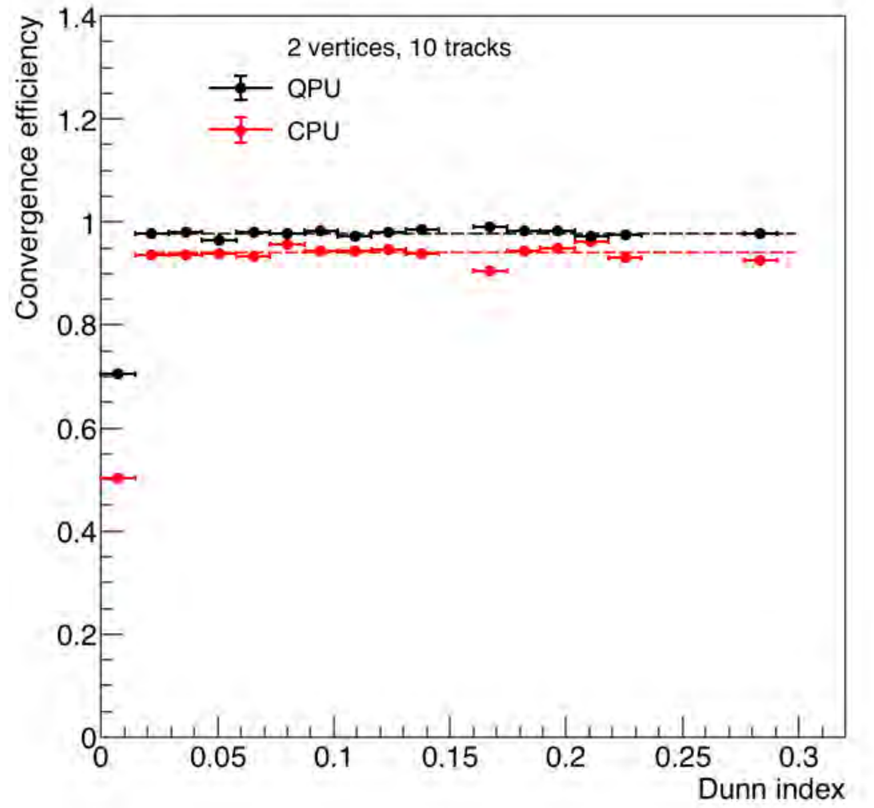
2 collisions, 10 tracks	2 collisions, 16 tracks	4 collisions, 12 tracks	4 collisions, 16 tracks
CPU: 0.94 ± 0.01	CPU: 0.46 ± 0.01	CPU: 0.32 ± 0.02	CPU: 0.17 ± 0.01
QPU: 0.98 ± 0.01	QPU: 0.96 ± 0.01	QPU: 0.24 ± 0.12	QPU: 0.08 ± 0.08

QPU vs CPU scaling with event complexity

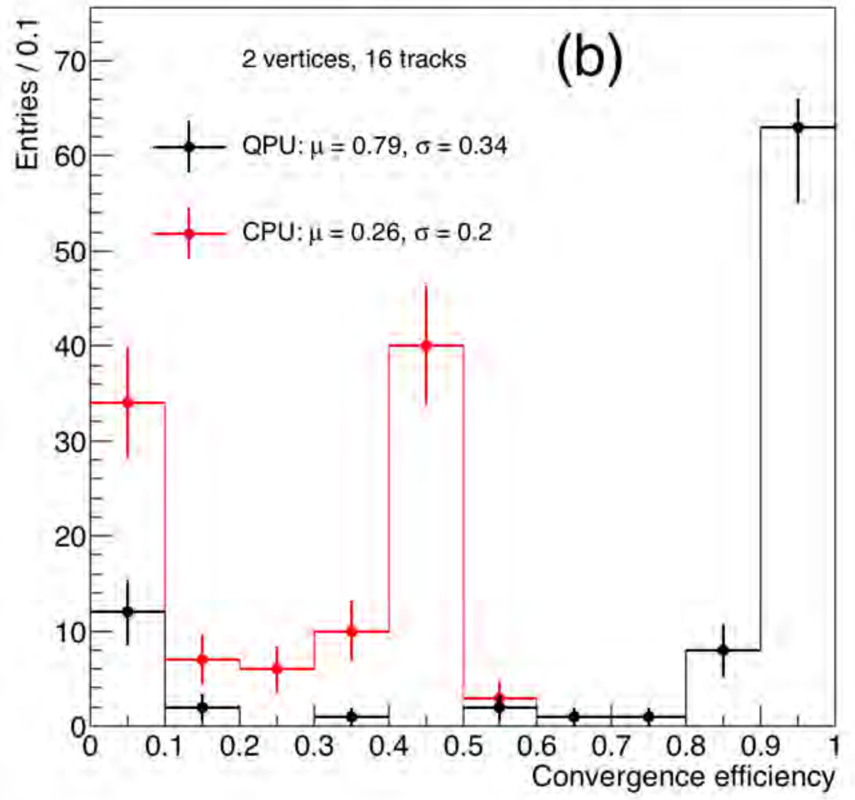
We scan over event topologies with increasing complexity

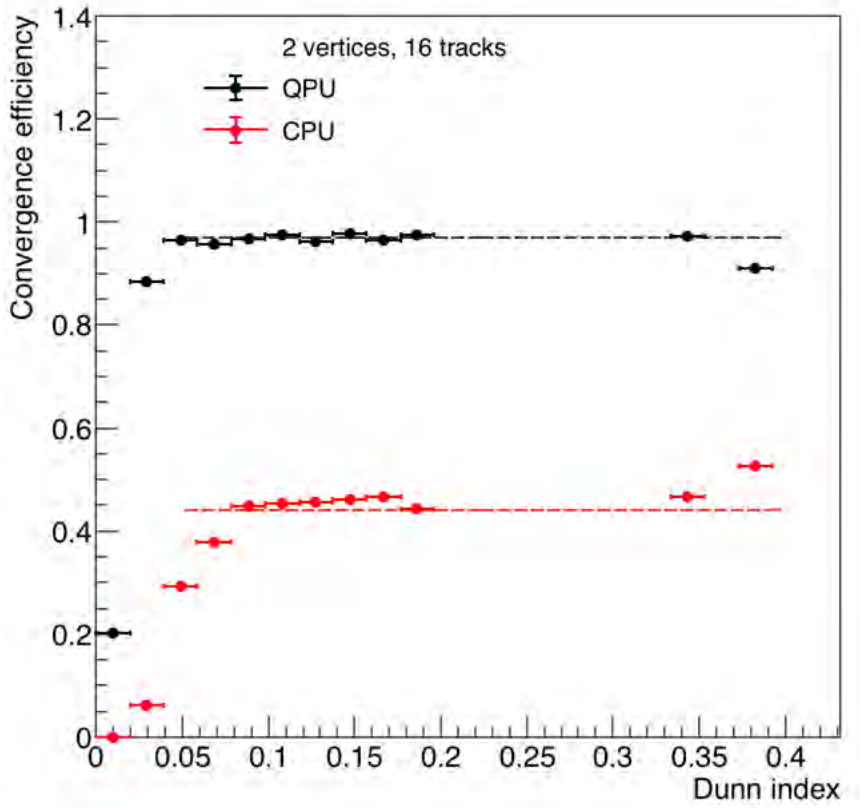
2 collisions, 10 tracks



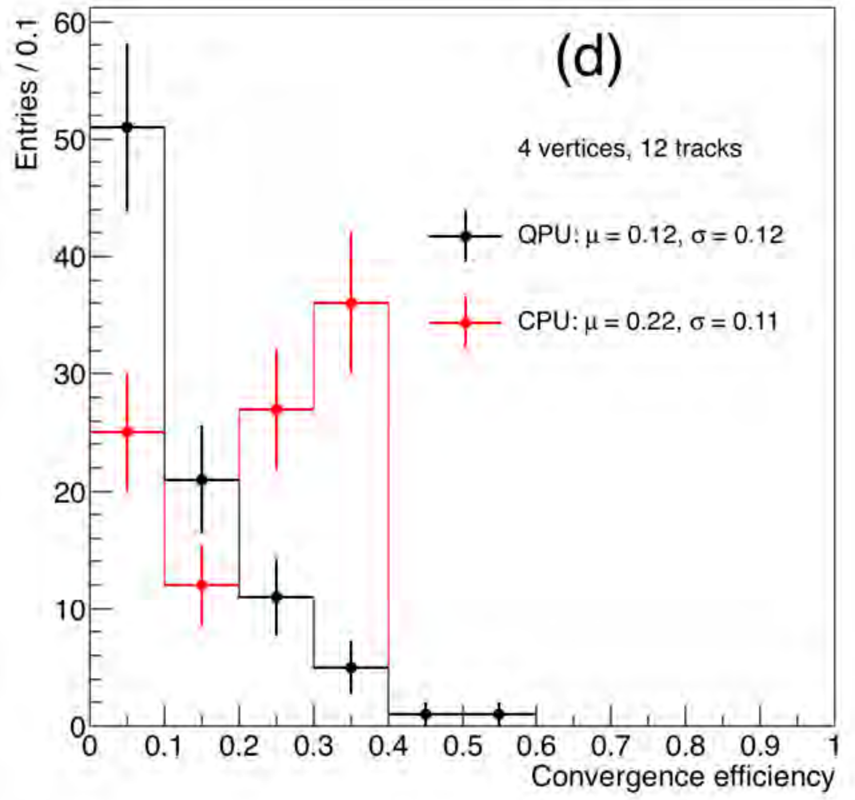


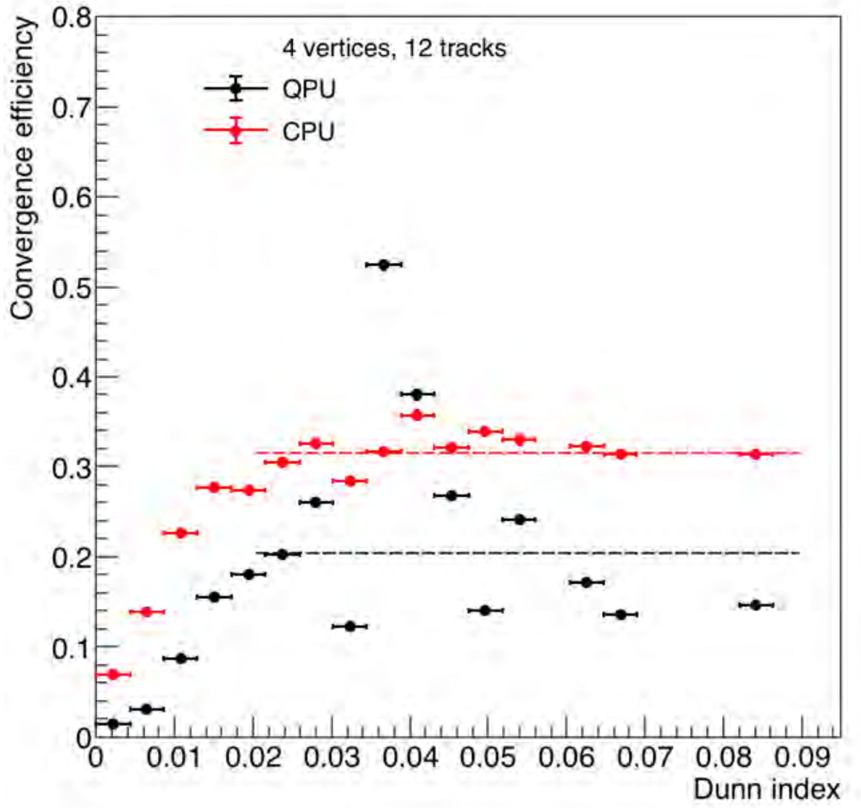
2 collisions, 16 tracks



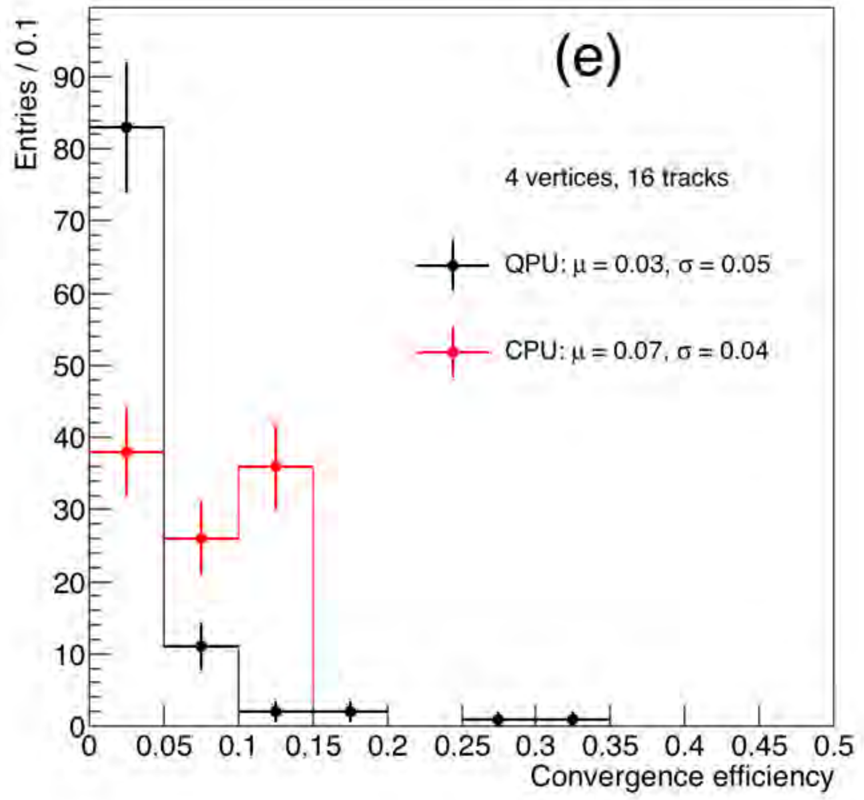


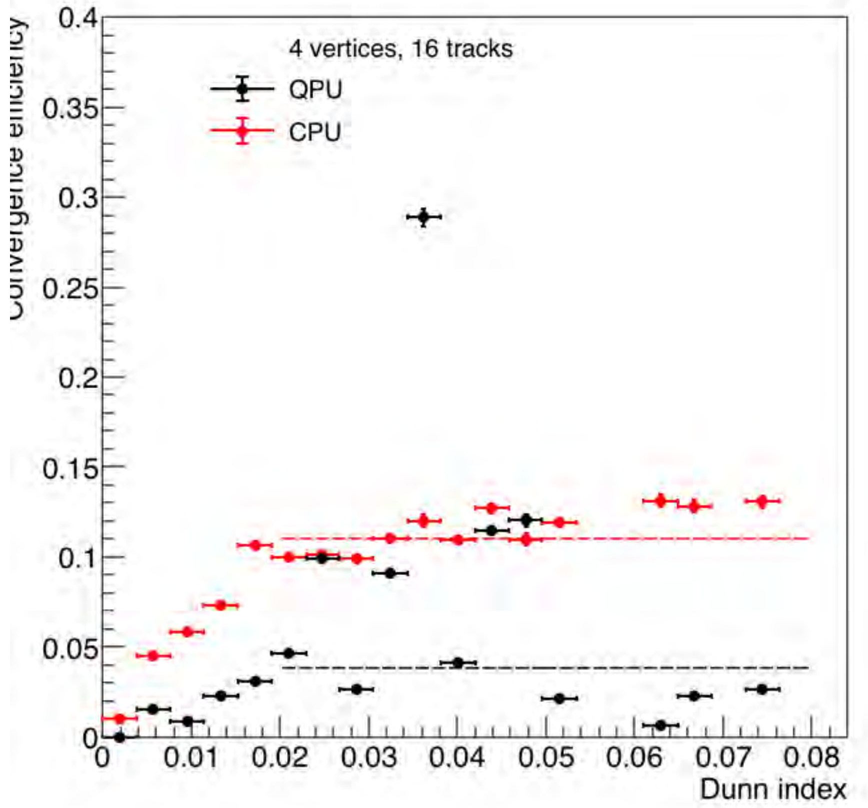
4 collisions, 12 tracks





4 collisions, 16 tracks



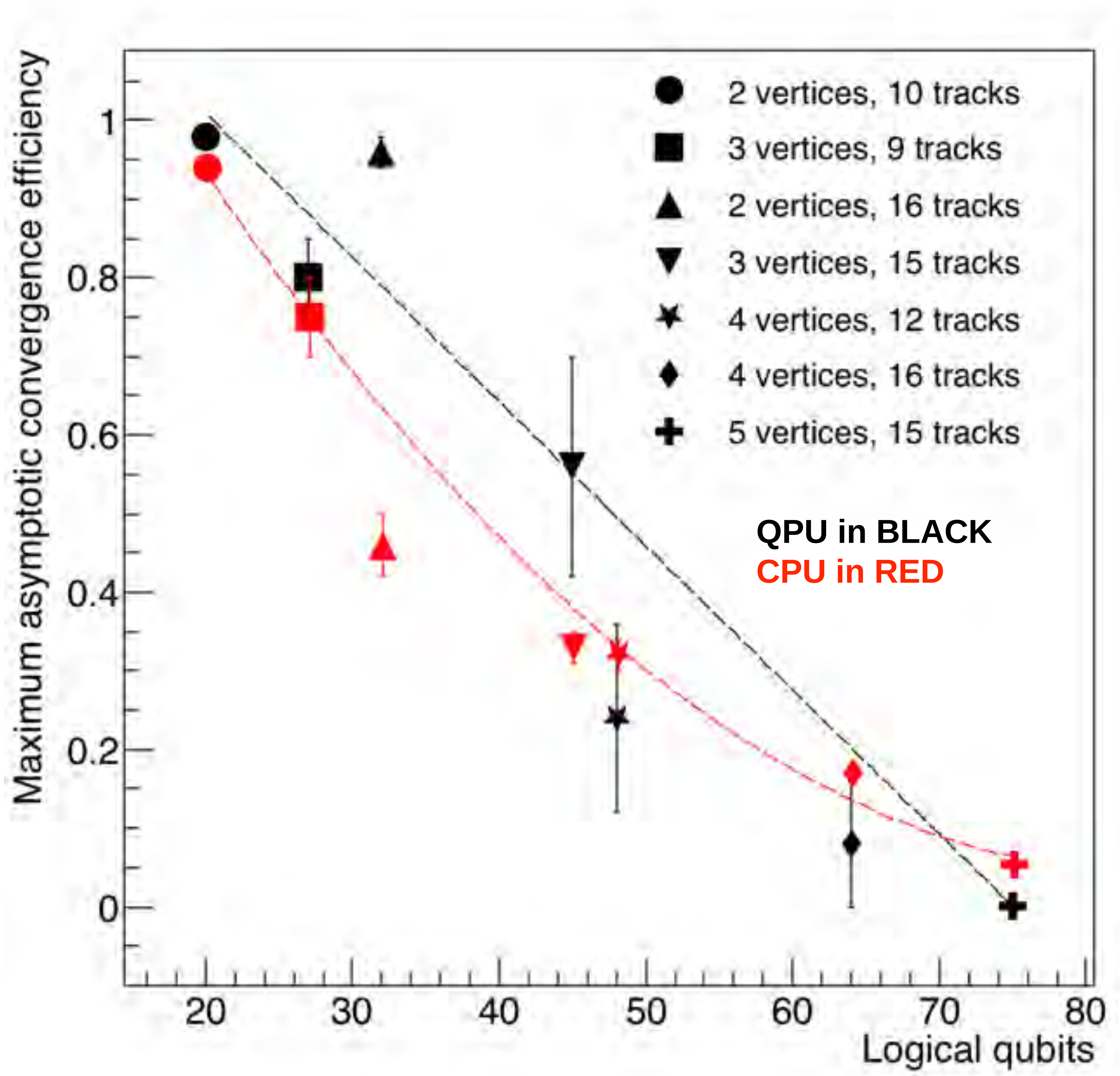


Maximum convergence efficiencies

2 collisions, 10 tracks	2 collisions, 16 tracks	4 collisions, 12 tracks	4 collisions, 16 tracks
CPU: 0.94 ± 0.01	CPU: 0.46 ± 0.01	CPU: 0.32 ± 0.02	CPU: 0.17 ± 0.01
QPU: 0.98 ± 0.01	QPU: 0.96 ± 0.01	QPU: 0.24 ± 0.12	QPU: 0.08 ± 0.08

- QPU has advantage at low complexity. Why?
- **Can any measure highlight the tunneling advantage?**

QPU vs CPU scaling with event complexity



- One measure of complexity: Number of logical qubits used = number of collisions × number of tracks
- Trend: **Asymptotic maximum of convergence efficiency** plotted against logical qubits
  - **Spread of maximum convergence efficiency** represented by uncertainty bars

QPU performance comparable to a modern CPU
QPU running may be further optimized

Conclusions and outlook

Conclusions

- The D-Wave 2000Q_2_1 QPU can reconstruct p-p collision positions at hadron colliders in a limited capacity
  - ➡ The Tevatron had ~ 3 p-p collisions per event. Would have been possible with D-Wave
- QPU implementation comparable to Simulated Annealing on MacBook CPU for equal time
- QPU implementation to be optimized for LHC complexities: 50 to 200 p-p collision per event

Outlook

Two research directions to improve QPU implementation:

- Improve convergence efficiency:
  - ➡ Understand how distortion functions like g(x; m) work
  - ➡ Use annealing offsets
  - ➡ Tune annealing time, re-thermalization delay
  - ➡ Try reverse annealing
  - ➡ Optimize chain lengths and weights
- Fit larger problems on QPU:
  - Customized embedding
  - Solve larger problems with hierarchical clustering

Acknowledgements

Support and useful discussions with D-Wave: **Joel Gottlieb**, Mark Johnson, Alexander Condello

Support from the Department of Energy grant DE-SC0007884 and the Purdue Research Foundation

Track clustering with a quantum annealer for primary vertex reconstruction at hadron colliders

S. Das, A. J. Wildridge, S. B. Vaidya, A. Jung

Department of Physics and Astronomy, Purdue University

Abstract

Clustering of charged particle tracks along the beam axis is the first step in reconstructing the positions of hadronic interactions, also known as primary vertices, at hadron collider experiments. We use a 2048 qubit D-Wave quantum annealer to perform track clustering in a limited capacity on artificial events where the positions of primary vertices and tracks are drawn from distributions measured by the Compact Muon Solenoid experiment at the Large Hadron Collider. The algorithm, which is not a classical-quantum hybrid but relies entirely on quantum annealing, is tested on a variety of event topologies from 2 primary vertices and 10 tracks to 5 primary vertices and 15 tracks. It is benchmarked against simulated annealing run on a modern CPU constrained to the same processor time per anneal as time in the physical annealer, and performance is found to be comparable. We chart three research directions to improve the performance of quantum annealers for this class of problems.

1. Introduction

Hadron colliders circulate counter-rotating beams of hadrons in closely packed bunches that cross at designated interaction points. These interaction points are instrumented with experiments that detect particles produced at hadron-hadron collisions when the bunches cross. Reconstructing the positions of these collisions within a bunch crossing, also known as primary vertices, from the trajectories of charged particles detected by the apparatuses is of paramount importance for physics analyses. The Large Hadron Collider (LHC) is a high luminosity collider that produces an average of 20 proton-proton (p-p) collisions at each bunch crossing, distributed in one dimension along the beam axis. At one of the LHC interaction points, the Compact Muon Solenoid experiment (CMS) reconstructs the paths of charged particles from p-p collisions as tracks detected by its silicon tracker [1]. Track reconstruction uncertainties obscure which tracks originated together at a primary vertex. Thus, primary vertex reconstruction begins with a one-dimensional clustering of tracks by their positions along the beam axis where they approach it most closely, also known as the tracks' z_0. In this paper, we demonstrate a method of performing this clustering on a D-Wave quantum annealer and report preliminary results benchmarked against simulated annealing on a classical computer.

The D-Wave 2000Q quantum computer, available from D-Wave Inc., performs computations through quantum annealing [2, 3, 4]. The quantum processing unit (QPU) has 2048 RF-SQUID flux qubits implemented as superconducting niobium loops [5]. Each qubit has a programmable external magnetic field to bias it. The network of qubits is not fully connected and programmable couplings have been implemented between 6016 pairs of qubits. A computational problem is defined by setting the biases (h_i) and couplings (J_{ij}) such that the ground state of the qubits' Hamiltonian corresponds to the solution. We call this the "problem Hamiltonian" (H_p)

$$H_p = \sum_i h_i \sigma_z^i + \sum_i \sum_{j>i} J_{ij} \sigma_z^i \sigma_z^j, \quad (1)$$

where σ_z^i is a spin projection observable of the i^{th} qubit with eigenvalues +1 and -1. (This z direction is not related to the beam axis at CMS.) It may be trivially mapped to a bit observable q_i with eigenvalues 0 and 1 through the shift $2q_i = \sigma_z^i + I$, where I is the identity matrix. The problem Hamiltonian may then be expressed for quadratic unconstrained binary optimization (QUBO) as

$$H_p = \sum_i a_i q_i + \sum_i \sum_{j>i} b_{ij} q_i q_j, \quad (2)$$

notwithstanding energy offsets that are irrelevant for optimization. The D-Wave 2000Q programming model allows us to specify a problem in QUBO form by specifying a_i and b_{ij}.

At the beginning of a typical annealing cycle in the QPU, a driver Hamiltonian puts all qubits in a superposition of the computational basis states by introducing a global energy bias in the transverse $x-$direction. Annealing proceeds by lowering this driver Hamiltonian while si-

Email addresses: das69@purdue.edu (S. Das), anjung@purdue.edu (A. Jung)

Preprint submitted to Nuclear Instruments and Methods *September 13, 2019*

S. Das, A. J. Wildridge, S. B. Vaidya, A. W. Jung,
"Track clustering with a quantum annealer for primary vertex reconstruction at hadron colliders"
https://arxiv.org/abs/1903.08879