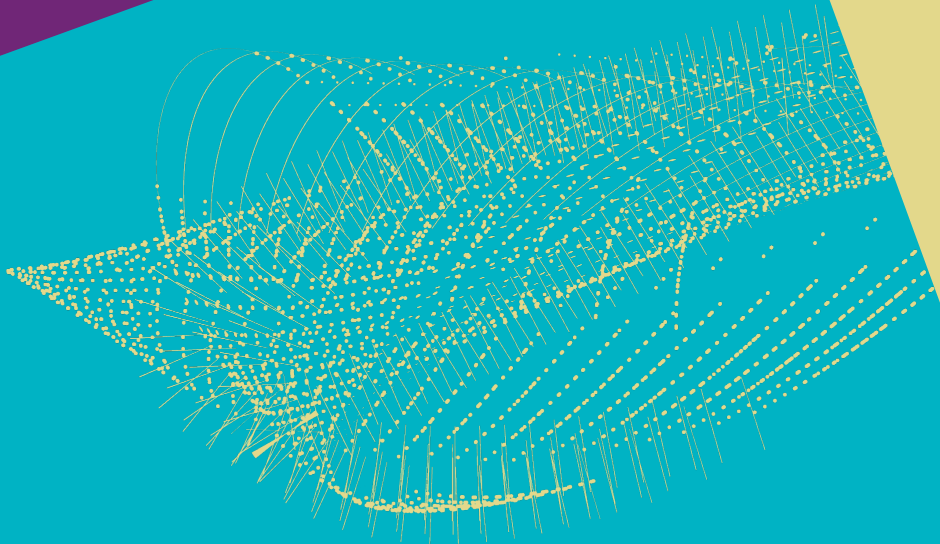
Nikhef

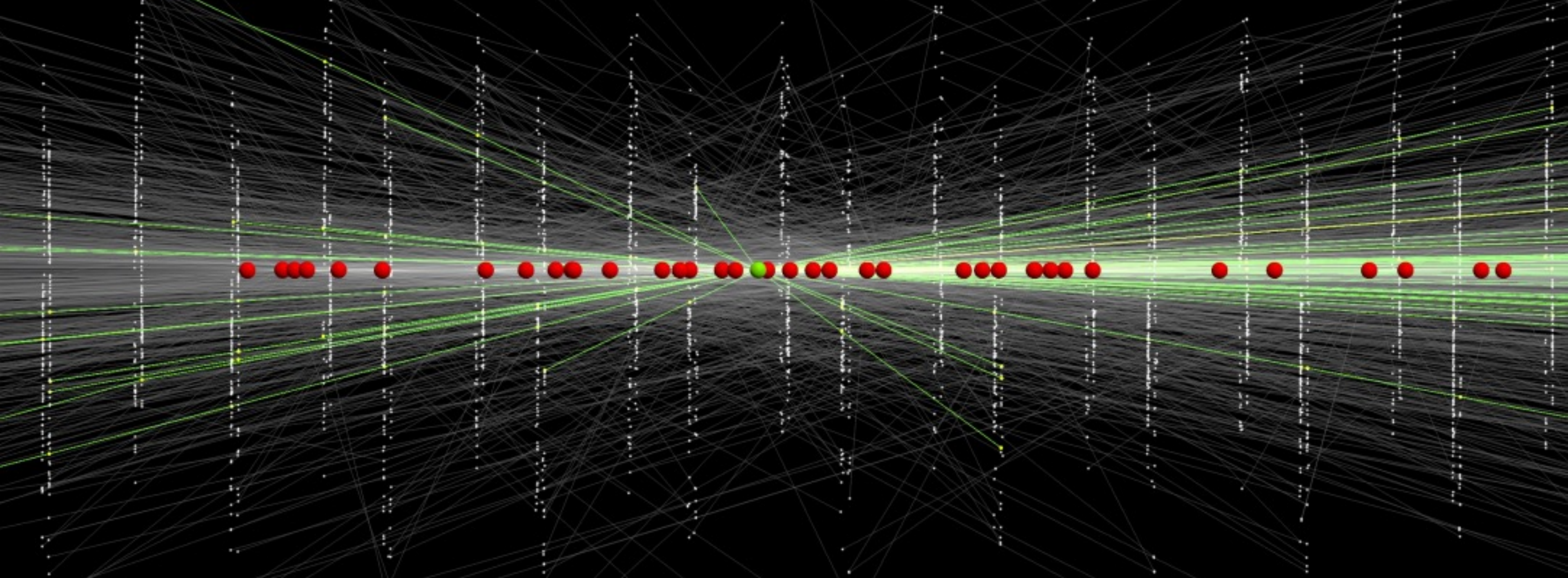
Maastricht University



Quantum Particle Tracking with TrackHHL

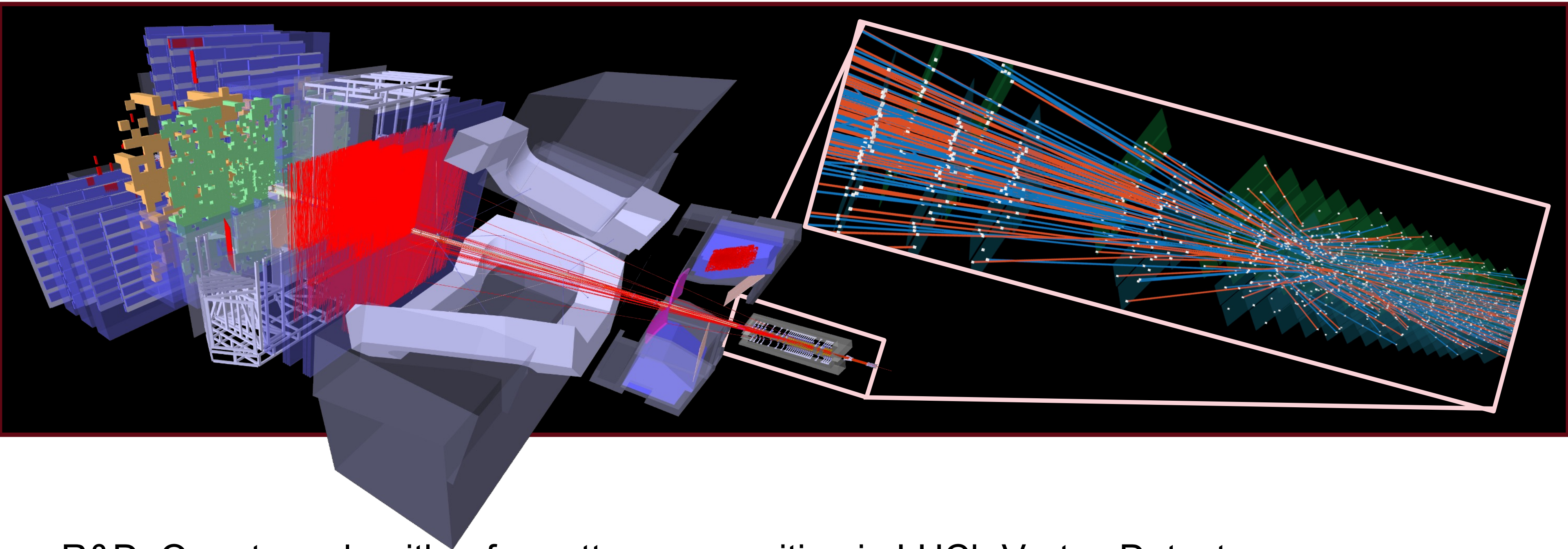
Xenofon Chiotopoulos
Nikhef and Maastricht University
Faster Day
October 2025

Current approach: massive parallel GPU-based combinatorics



R&D: Can a quantum algorithm help?

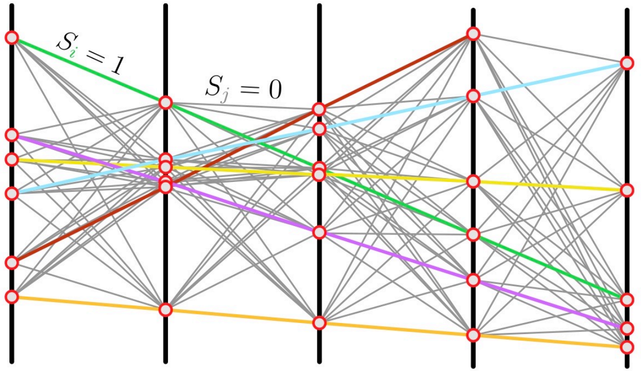
R&D: LHCb Velo detector



R&D: Quantum algorithm for pattern recognition in LHCb Vertex Detector:

- Algorithmic use case for a small scale quantum computer
  - Applications long term depending also on hardware developments
- R&D towards longer term with potential spin-offs
  - Investigated various approaches: HHL, QAOA, VQE, VQLS, annealing

Method: “Global” algorithm for trackfinding

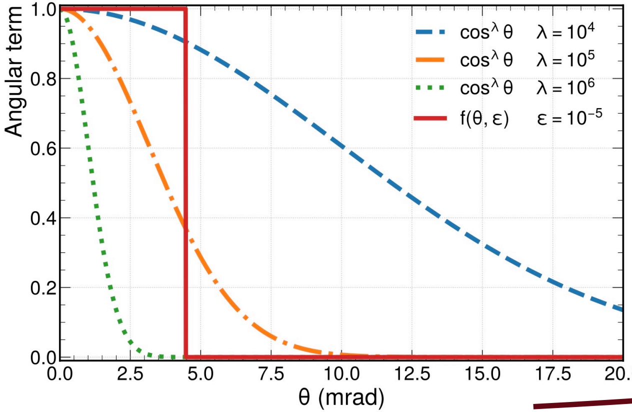


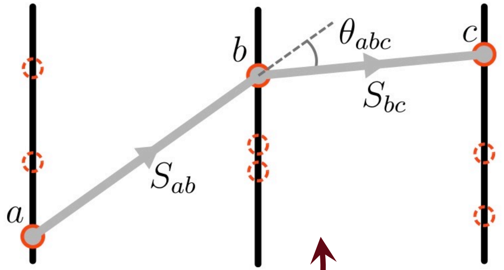
1) Build an Ising-like (quadratic) Hamiltonian

Define a Hamiltonian based on doublets of hits S_i being ”on track” $S_i = 1$ or “off track” $S_j = 0$

$$\mathcal{H}(S) = \sum_{abc} f(\theta) S_{ab} S_{bc} + \gamma \sum_{ab} S_{ab}^2 + \delta \sum_{ab} (1 - 2S_{ab})^2$$

- Angular term: assigns value for scattering
- Spectral term: ($\gamma = 2.0$) makes the spectrum of A_{ij} positive
- Gap term: ($\delta = 1.0$) ensures gap in the solution spectrum





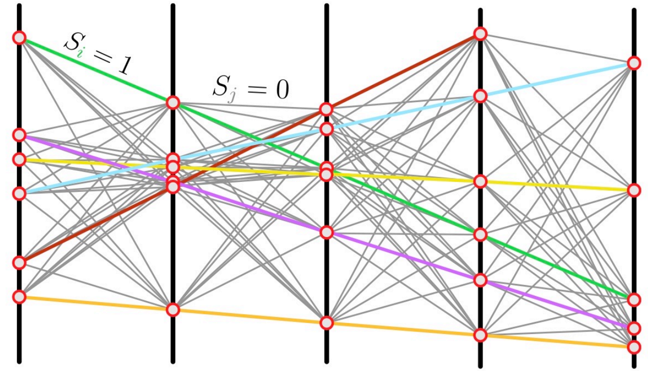
Nikhef

Method: “Global” algorithm for trackfinding

1) Build an Ising-like (quadratic) Hamiltonian

Define a Hamiltonian based on doublets of hits S_i being ”on track” $S_i = 1$ or “off track” $S_j = 0$

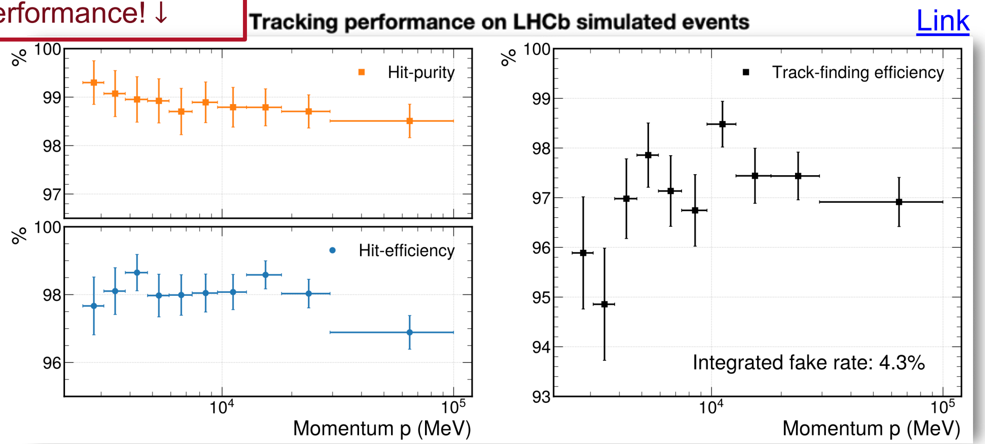
$$\mathcal{H}(S) = \sum_{abc} f(\theta) S_{ab} S_{bc} + \gamma \sum_{ab} S_{ab}^2 + \delta \sum_{ab} (1 - 2S_{ab})^2$$



2) Find ground state of:

$$\mathcal{H}(S) = \sum_{ij} A_{ij} S_i S_j + \sum_i b_i S_i \quad S_i \in \{0,1\}$$

Try by solving classically good performance! ↓

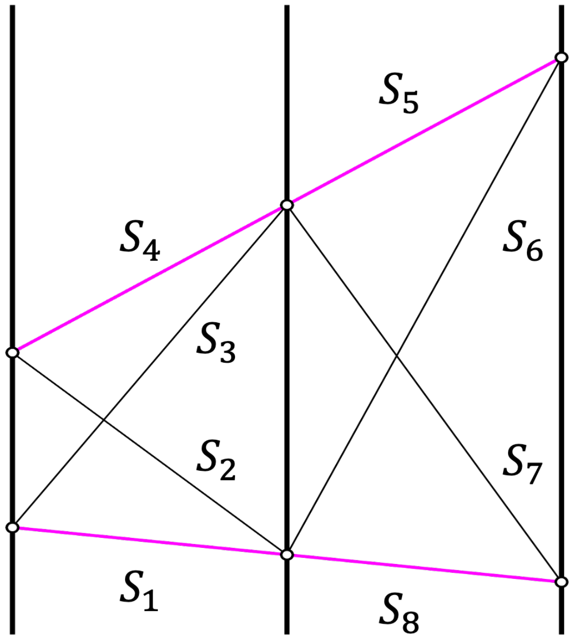


3) Apply a trick:

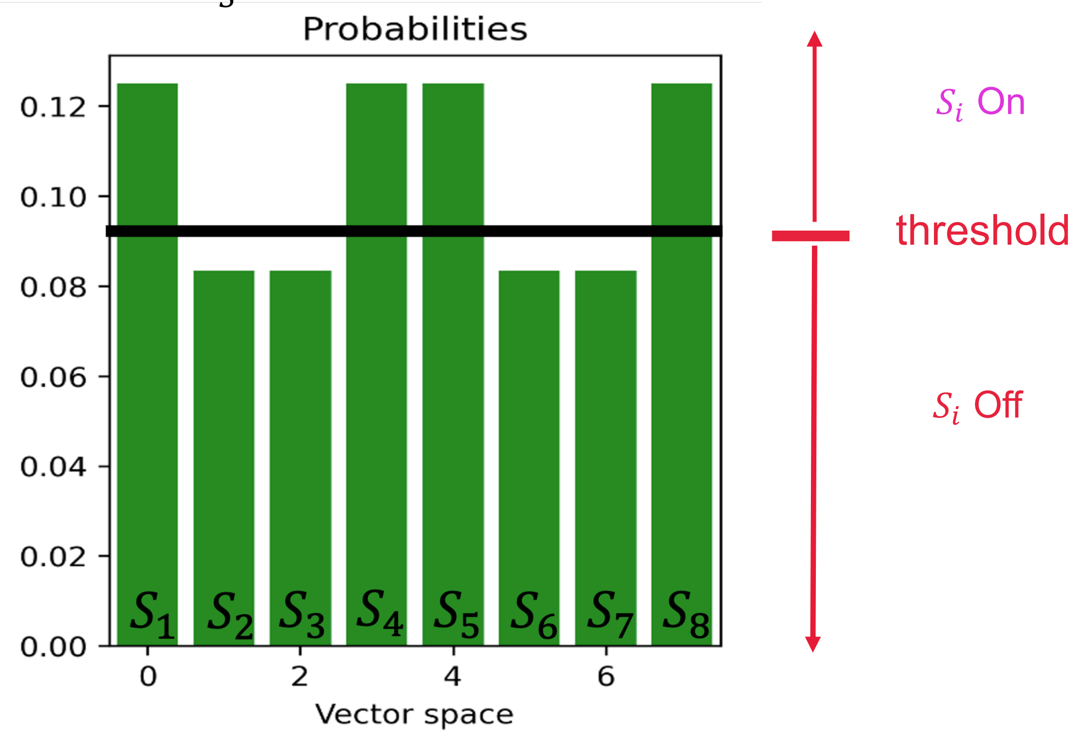
- *Relaxation*: $S_i \in \{0,1\} \;\Rightarrow\; S_i \in \mathbb{R}$
- *Minimization*: $\nabla_S \mathcal{H} = 0 \;\Rightarrow\; A\boldsymbol{S} = b$
  - Solve the large matrix inversion to find $\boldsymbol{S}$
- *Discretization*: $S_i \in \mathbb{R} \;\Rightarrow\; S_i \in \{0,1\}$

Simplest (trivial) case how it works classically

Two tracks in three layers



$\nabla_S \mathcal{H} = 0 \Rightarrow \boldsymbol{S} = A^{-1} b$

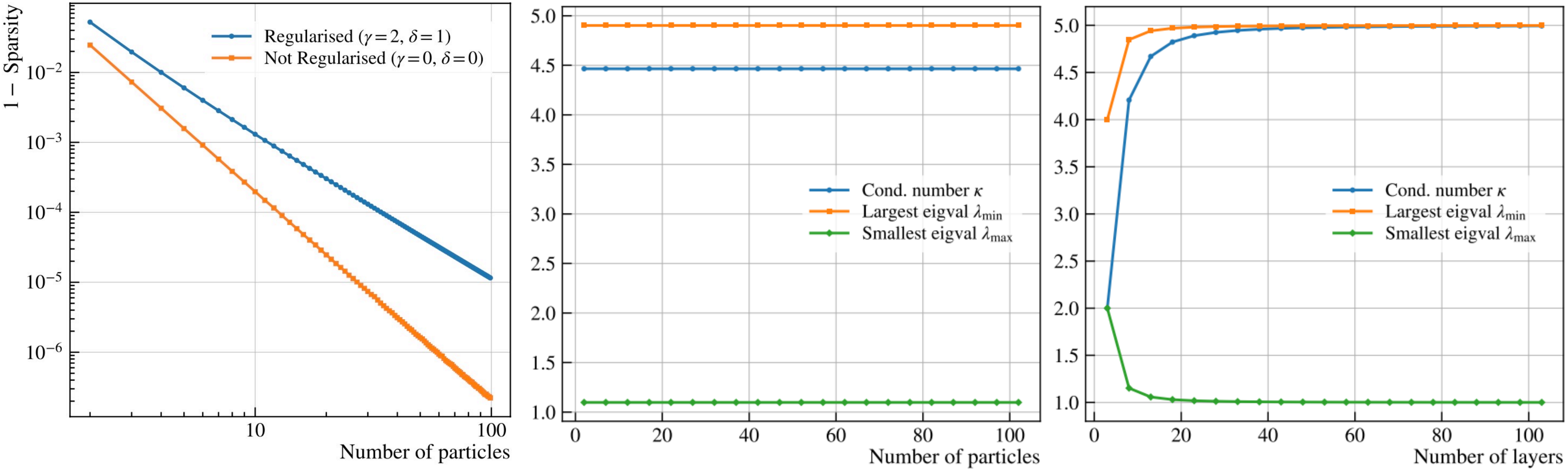


- Algorithm finds that:
- S_1, S_4, S_5, S_8 are good segments and S_2, S_3, S_6, S_7 are wrong cobinations

Matrix Properties

A is very well behaved: very Sparse

Low condition-number: $\kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|} \approx 5$



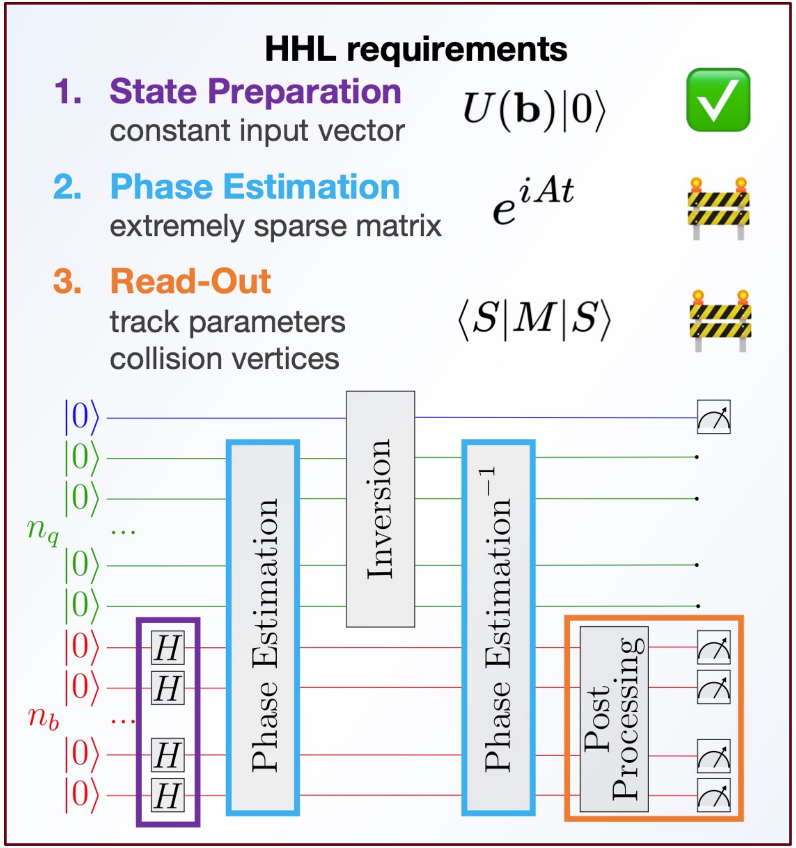
HHL Promises $O(\log(N)\kappa^2)$ runtime vs $O(N\sqrt{\kappa})$

The Quantum Algorithm: Harrow-Hassadim-Lloyd

Solving the matrix equation to find solution S

$\boldsymbol{AS} = \boldsymbol{b} \Rightarrow \boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{b}$ Solve using HHL:

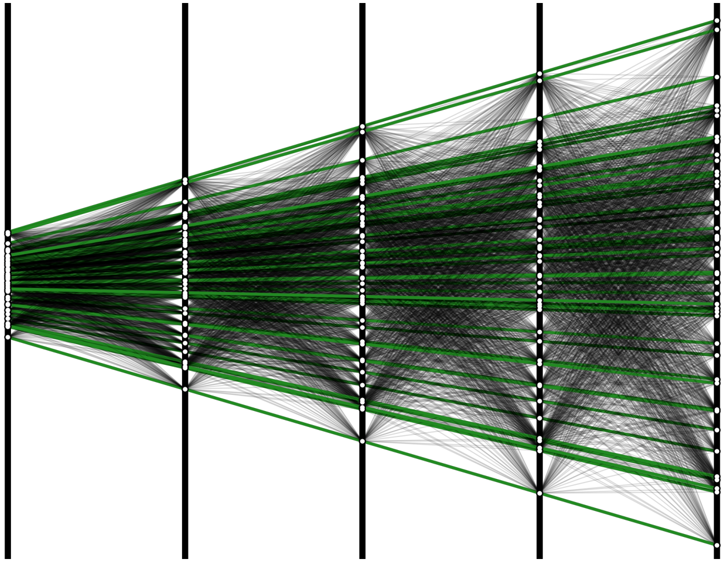
1. State preparation
2. - Quantum Phase estimation
   - Inversion
   - Quantum Phase Estimation†
3. Measurement



Quantum state: $|\Phi\rangle_i|\Phi\rangle_b|\Phi\rangle_q$

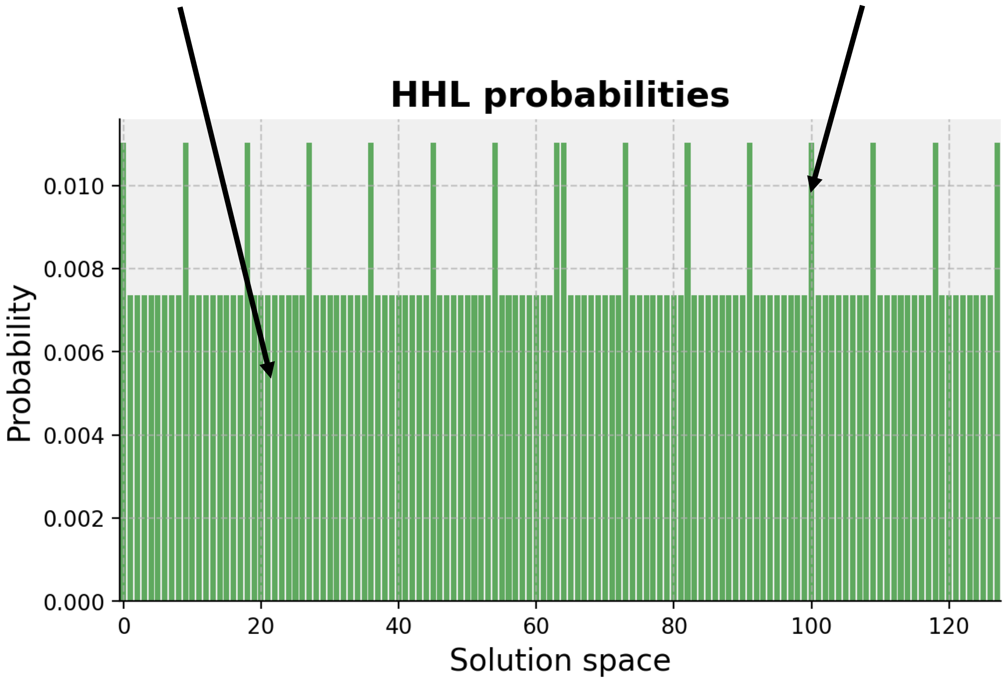
- $|\Phi\rangle_i$ ancilla qubit for controlled phase rotation
- $n_b = \log_2 N$ qubits to store vector $\boldsymbol{b}$
- Prepare $\boldsymbol{b} = (1,1,1,1,1,1)$ with Hadamards
- QPE: apply $U = e^{iA}$ with n bit precision
- n_q ancilla qubits for phase estimation
- Readout S

Consider the case:

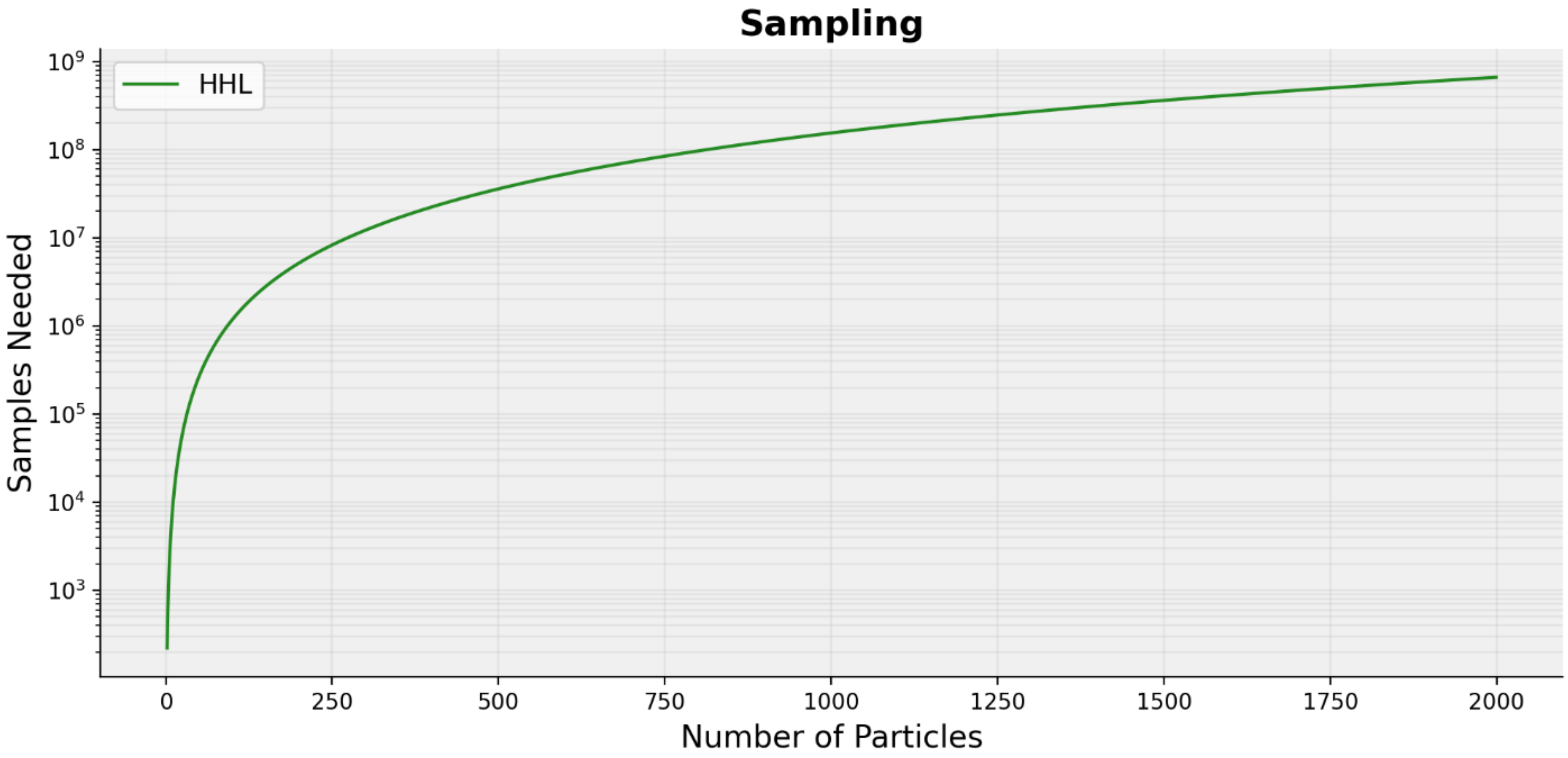


Solutions “off”
$S_i = 0$

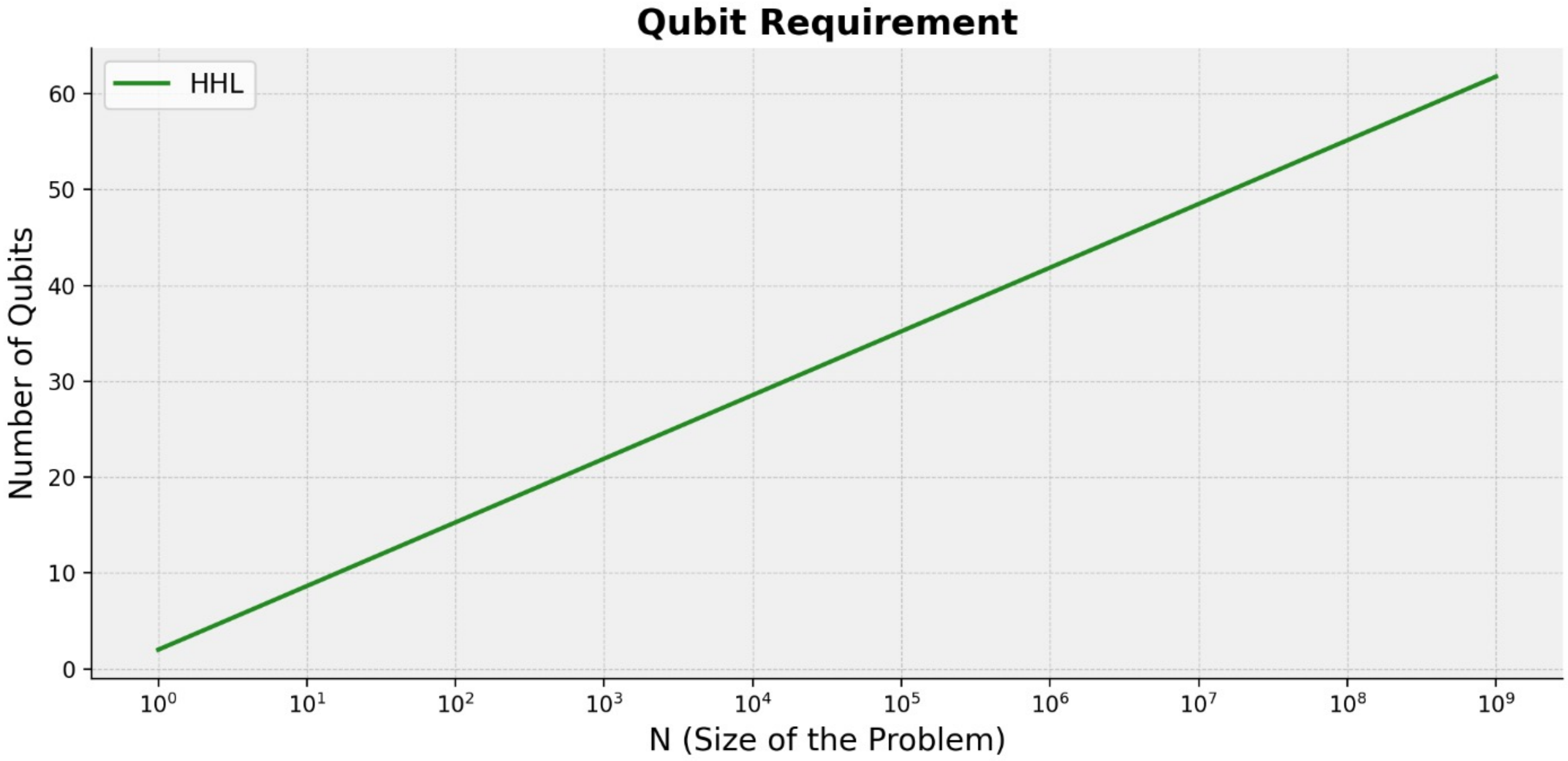
Solutions “on”
$S_i = 1$



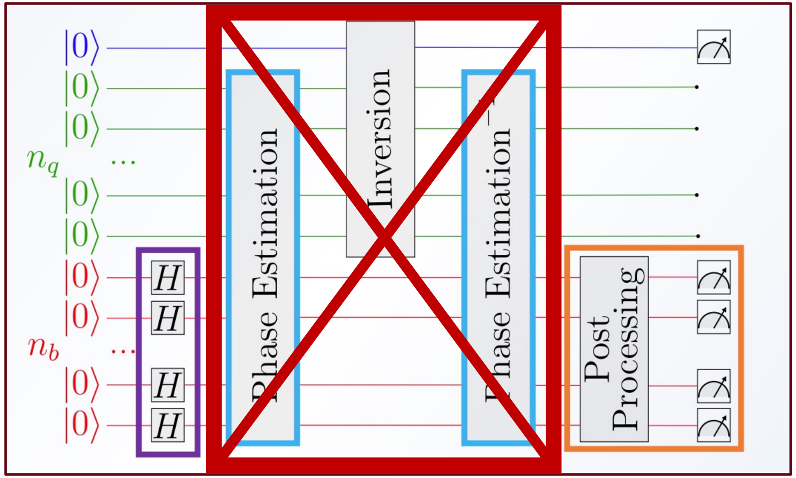
Sampling Problem



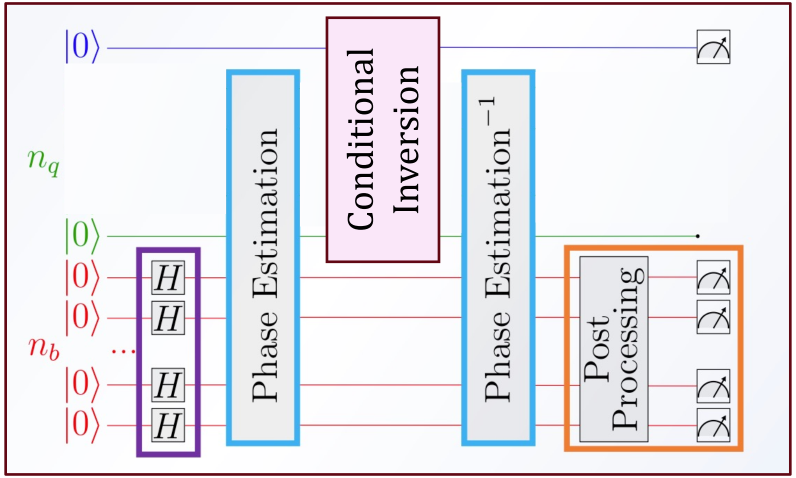
Qubit Requirement



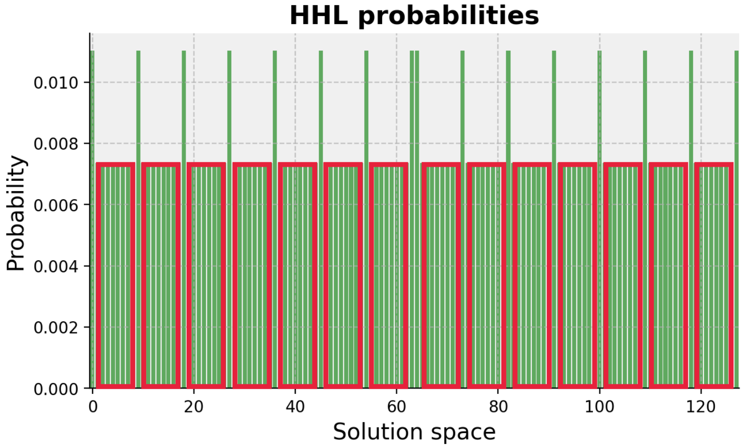
1-Bit HHL



|0⟩
n_q
|0⟩
n_b
H
Phase Estimation
Inversion
Phase Estimation
Post Processing

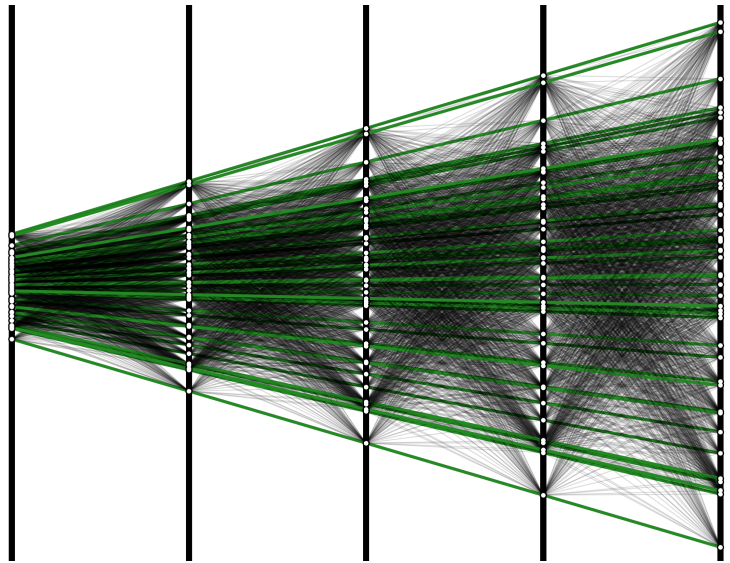


|0⟩
n_q
n_b
H
Phase Estimation
Conditional Inversion
Phase Estimation^{-1}
Post Processing



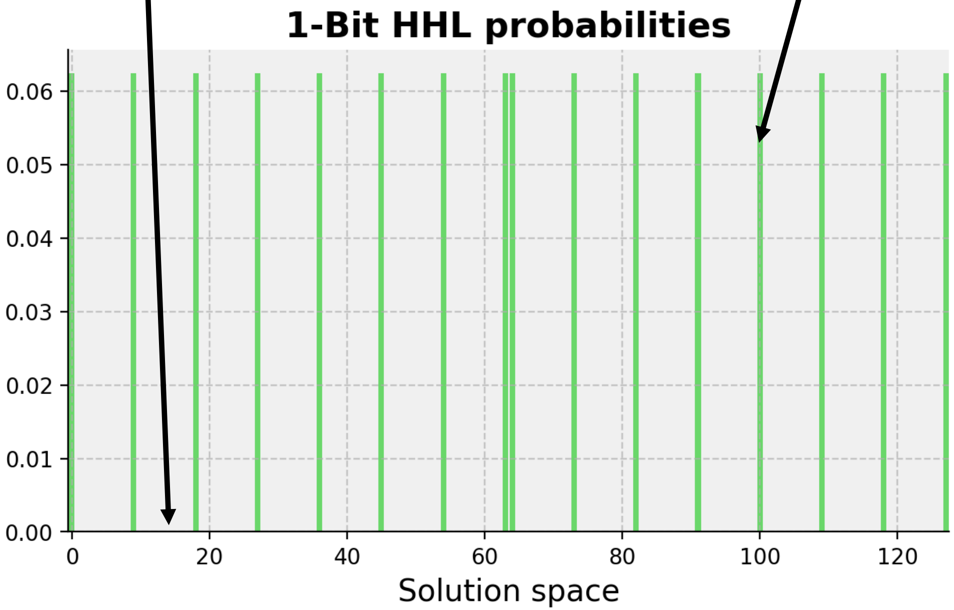
HHL probabilities
Probability
Solution space
0.000
0.002
0.004
0.006
0.008
0.010
0
20
40
60
80
100
120

Consider the case:

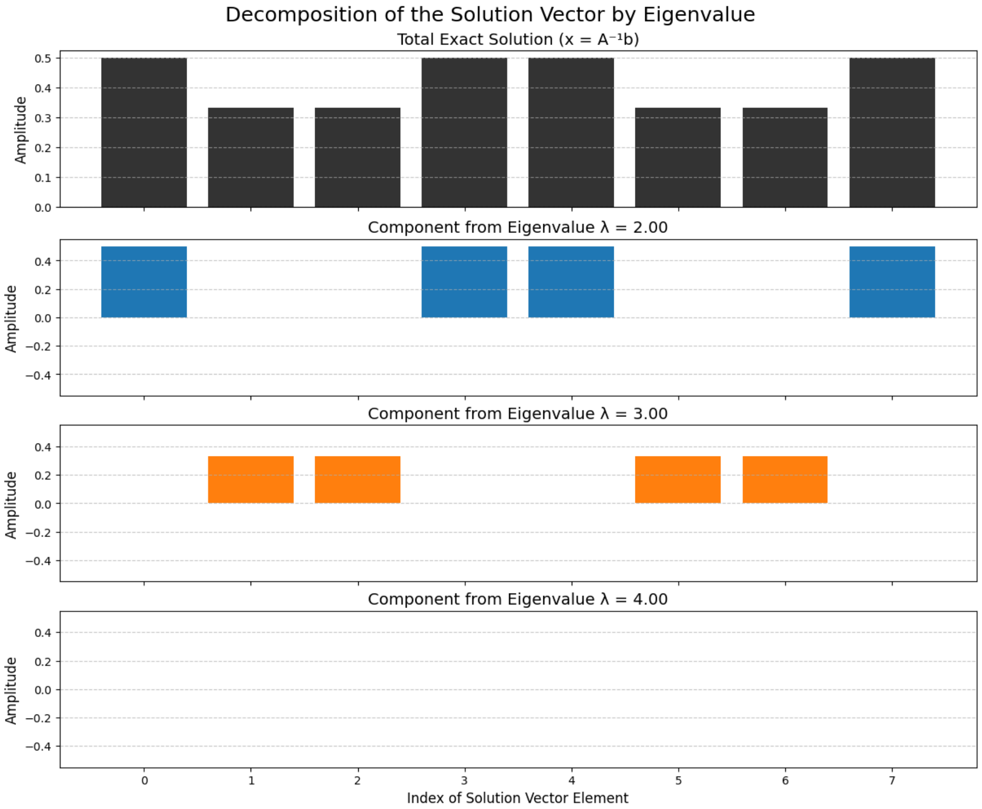


Solutions “off”
$S_i = 0$

Solutions “on”
$S_i = 1$

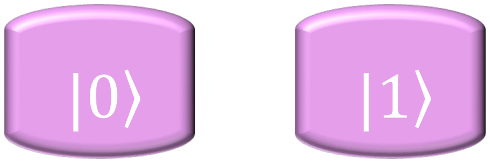


How does this work?



Results from this week

QPE BIN



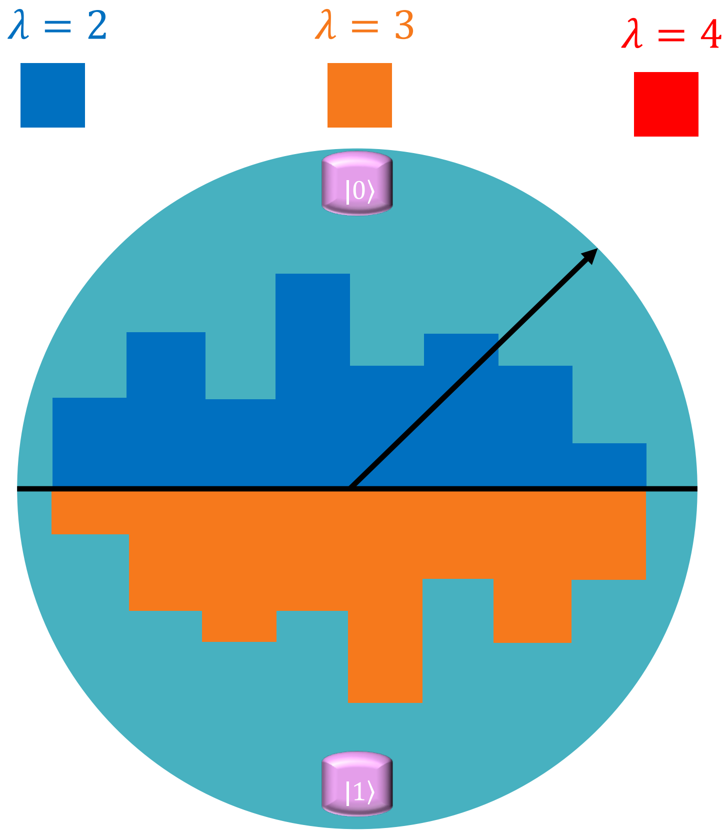
$$P(j=0|\lambda_k) = \cos^2\left(\frac{\lambda_k t}{2}\right)$$

$$\lambda_c = \frac{\lambda_{\min} + \lambda_{\max}}{2} \implies t = \frac{\pi}{\lambda_c}$$

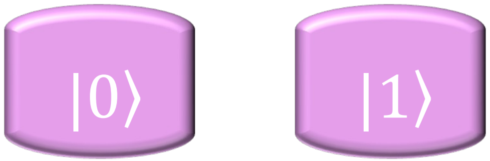
$$P(j=0|\lambda_c) = \cos^2\left(\frac{\lambda_c \pi}{2\lambda_c}\right) = \cos^2\left(\frac{\pi}{2}\right) = 0$$

$$P(j=1|\lambda_c) = \sin^2\left(\frac{\lambda_c \pi}{2\lambda_c}\right) = \sin^2\left(\frac{\pi}{2}\right) = 1$$

A Quantum Clock



QPE BIN



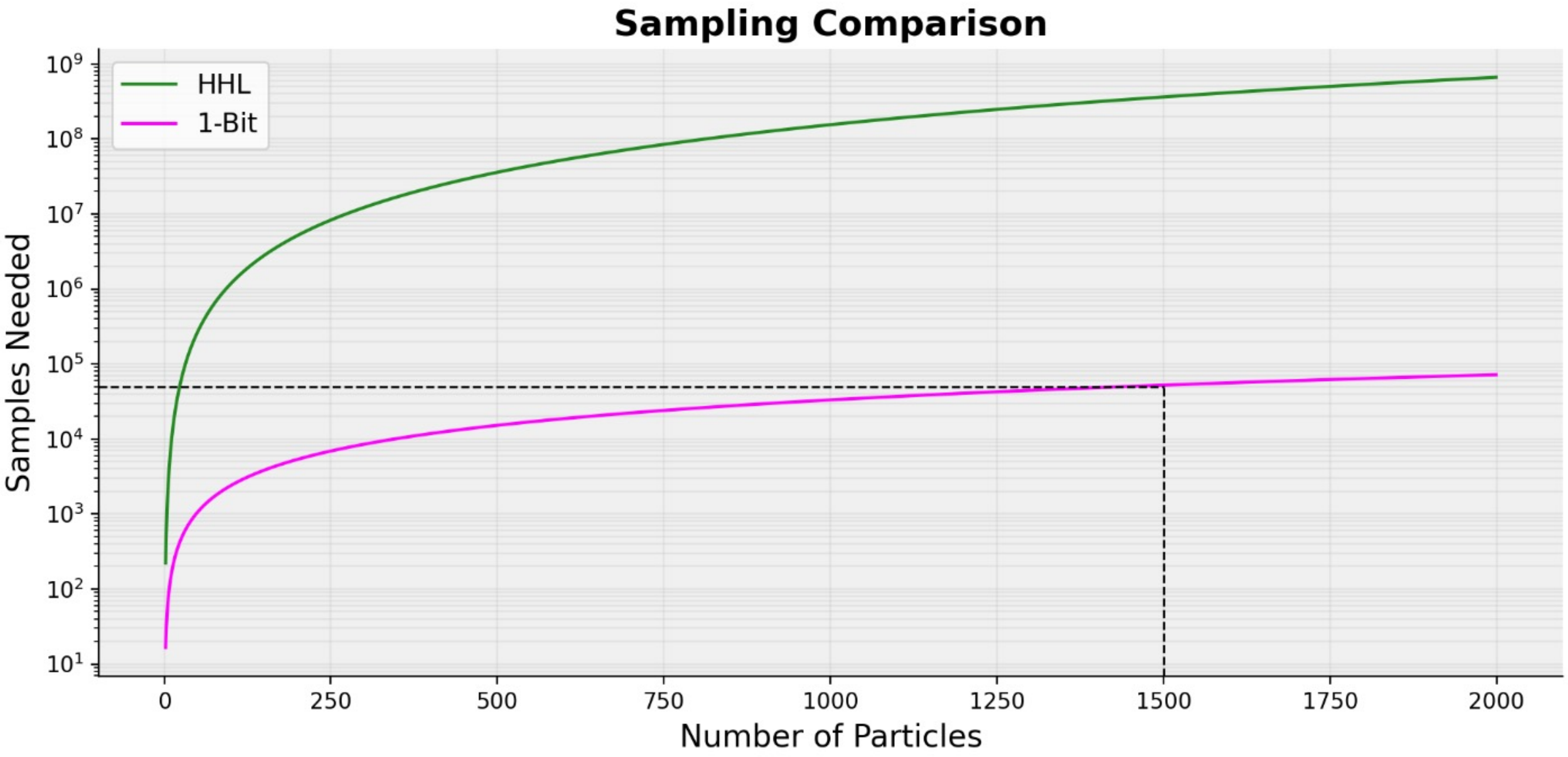
$$P(j=0|\lambda_k) = \cos^2\left(\frac{\lambda_k t}{2}\right)$$

$$\lambda_c = \frac{\lambda_{\text{min}} + \lambda_{\text{max}}}{2} \implies t = \frac{\pi}{\lambda_c}$$

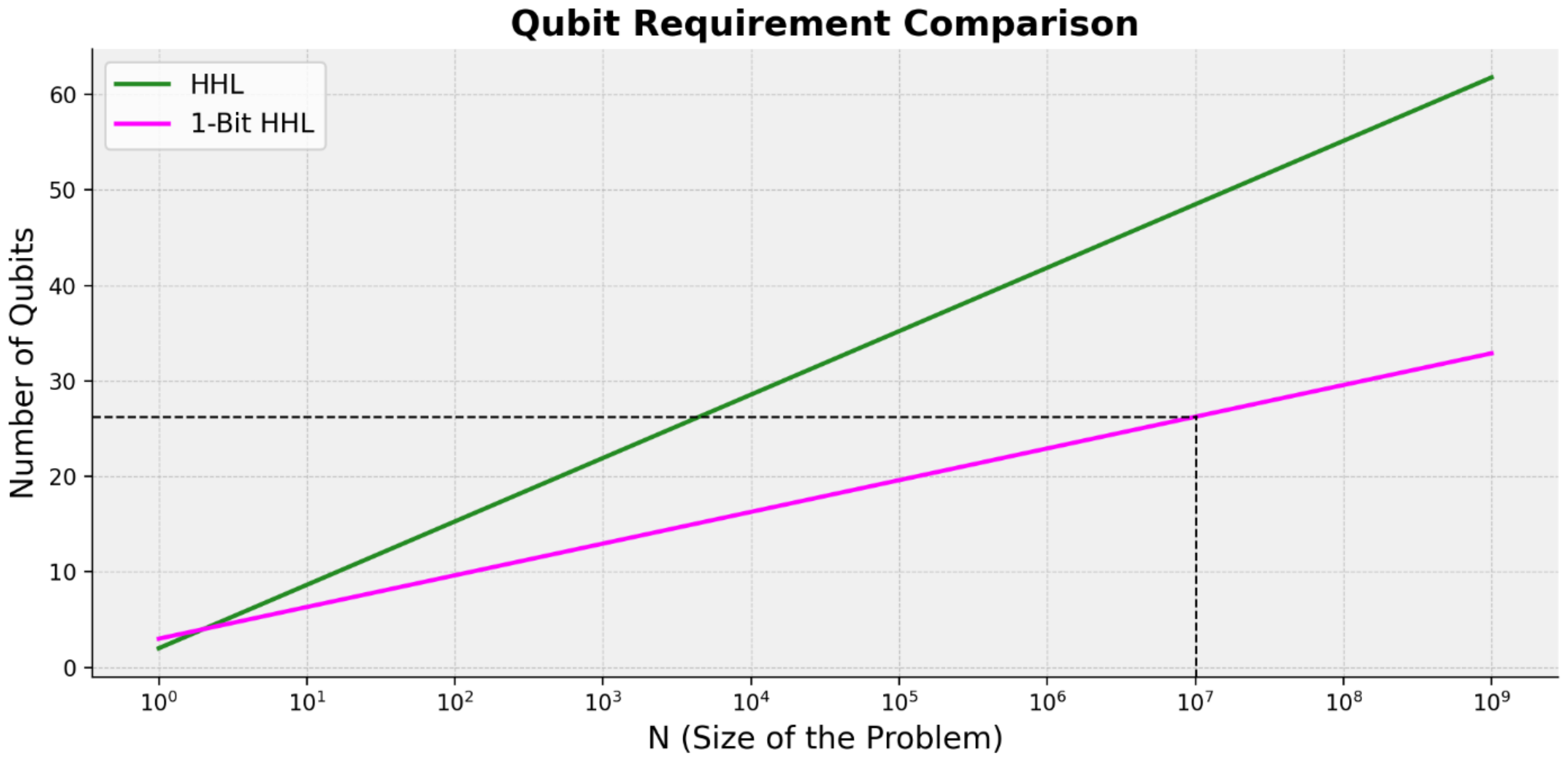
$$P(j=0|\lambda_c) = \cos^2\left(\frac{\lambda_c \pi}{2\lambda_c}\right) = \cos^2\left(\frac{\pi}{2}\right) = 0$$

$$P(j=1|\lambda_c) = \sin^2\left(\frac{\lambda_c \pi}{2\lambda_c}\right) = \sin^2\left(\frac{\pi}{2}\right) = 1$$

Sampling Problem

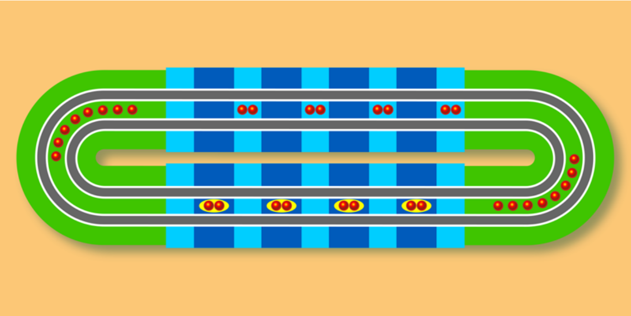


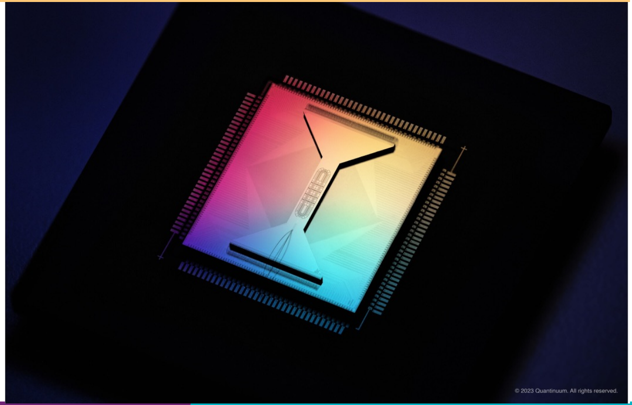
Qubit Requirement



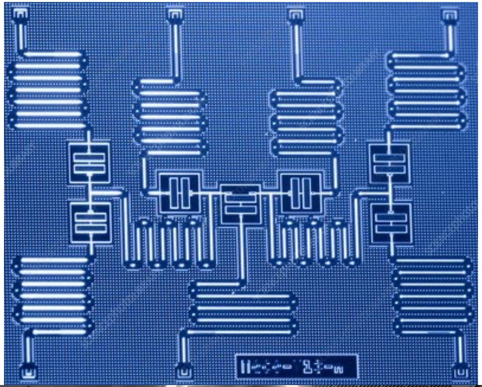
Hardware Choices

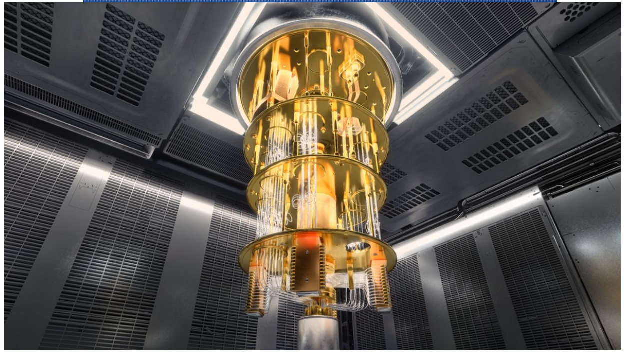
Trapped Ions (Quantinuum)



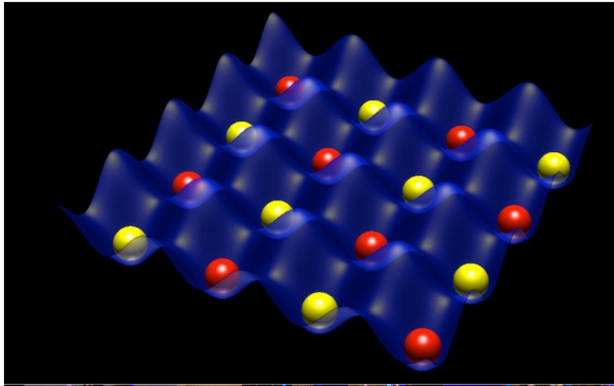


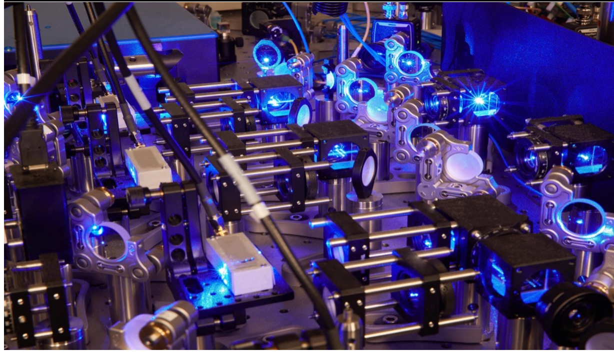
Superconducting (IBM)



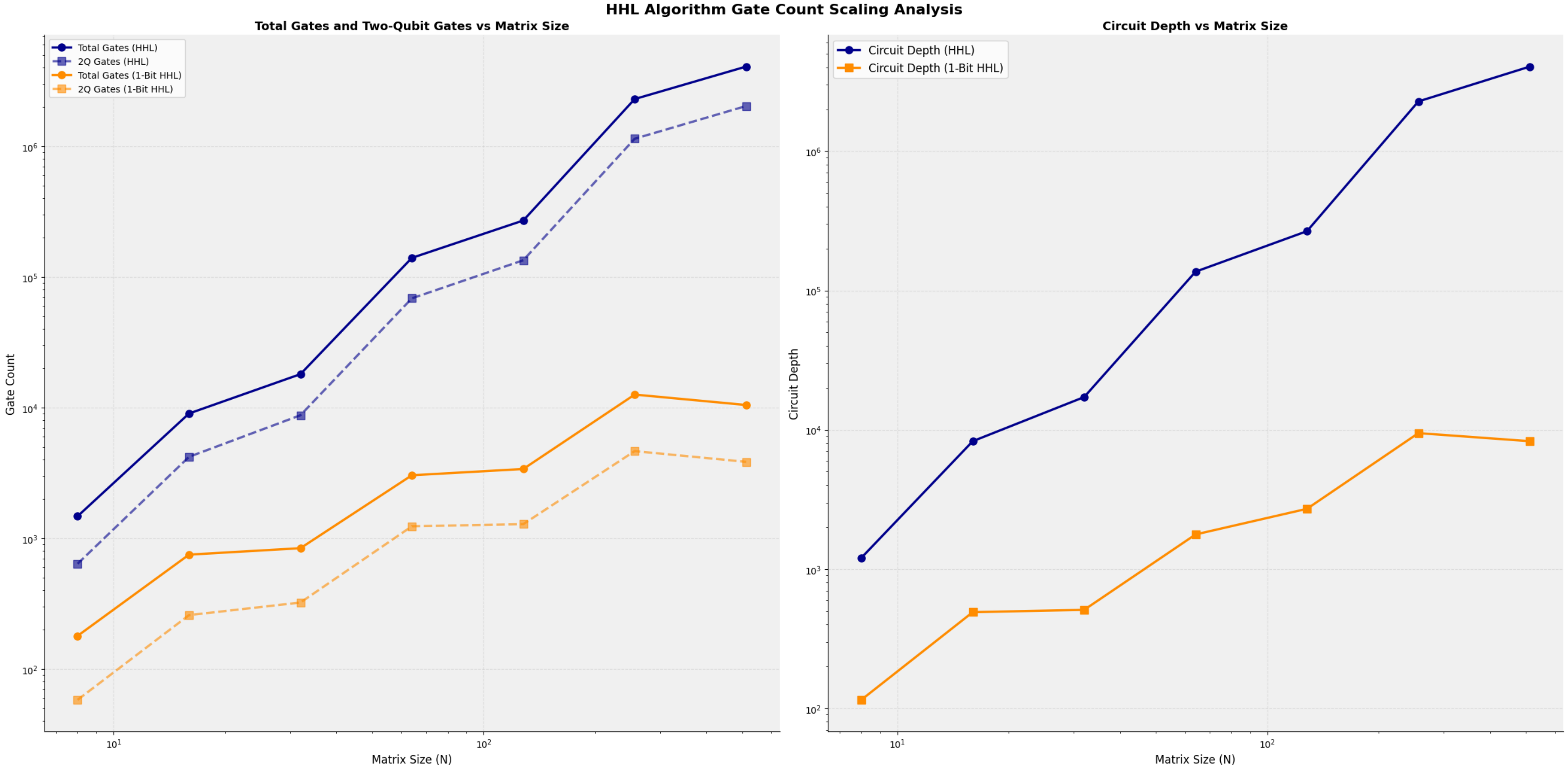


Neutral Atoms (TU Eindhoven)

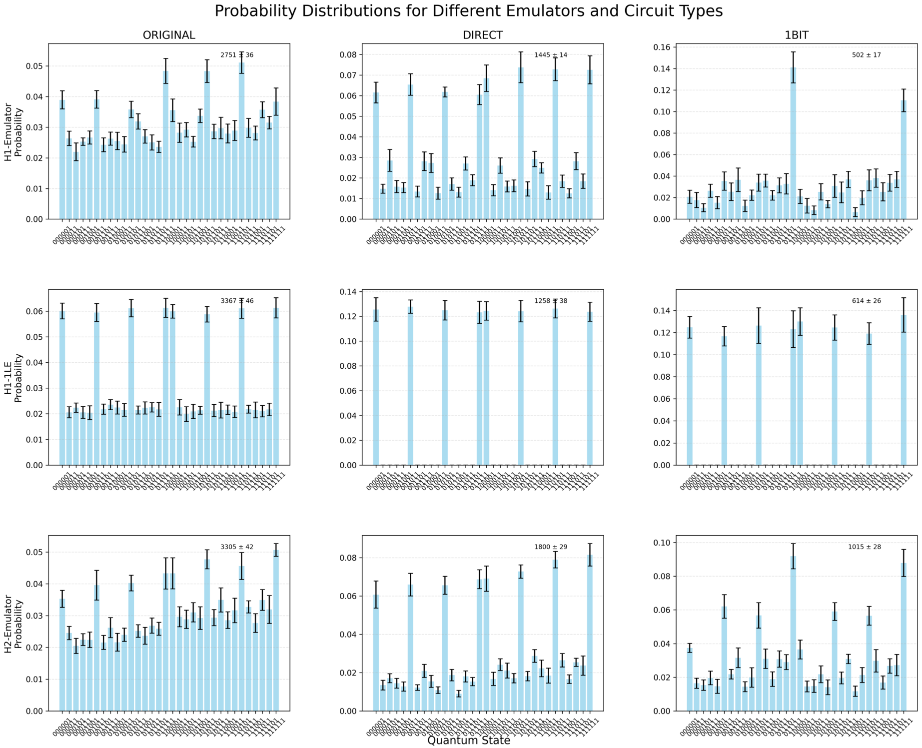


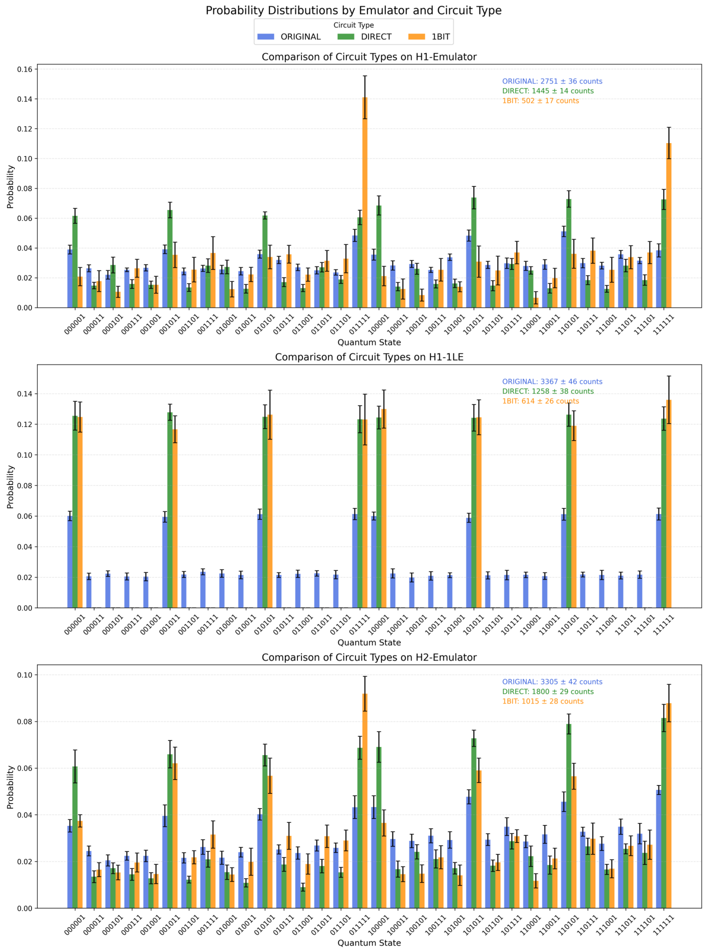


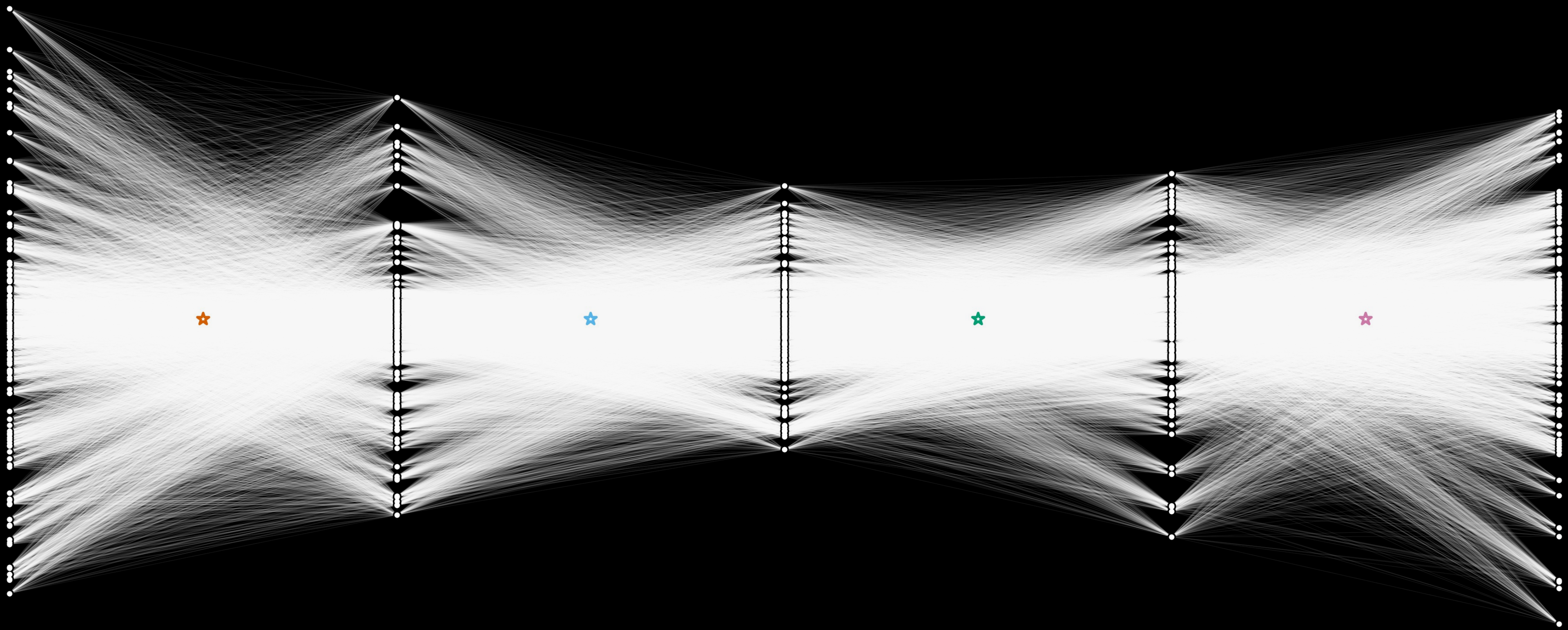
Circuit Depth (Hardware Agnostic)

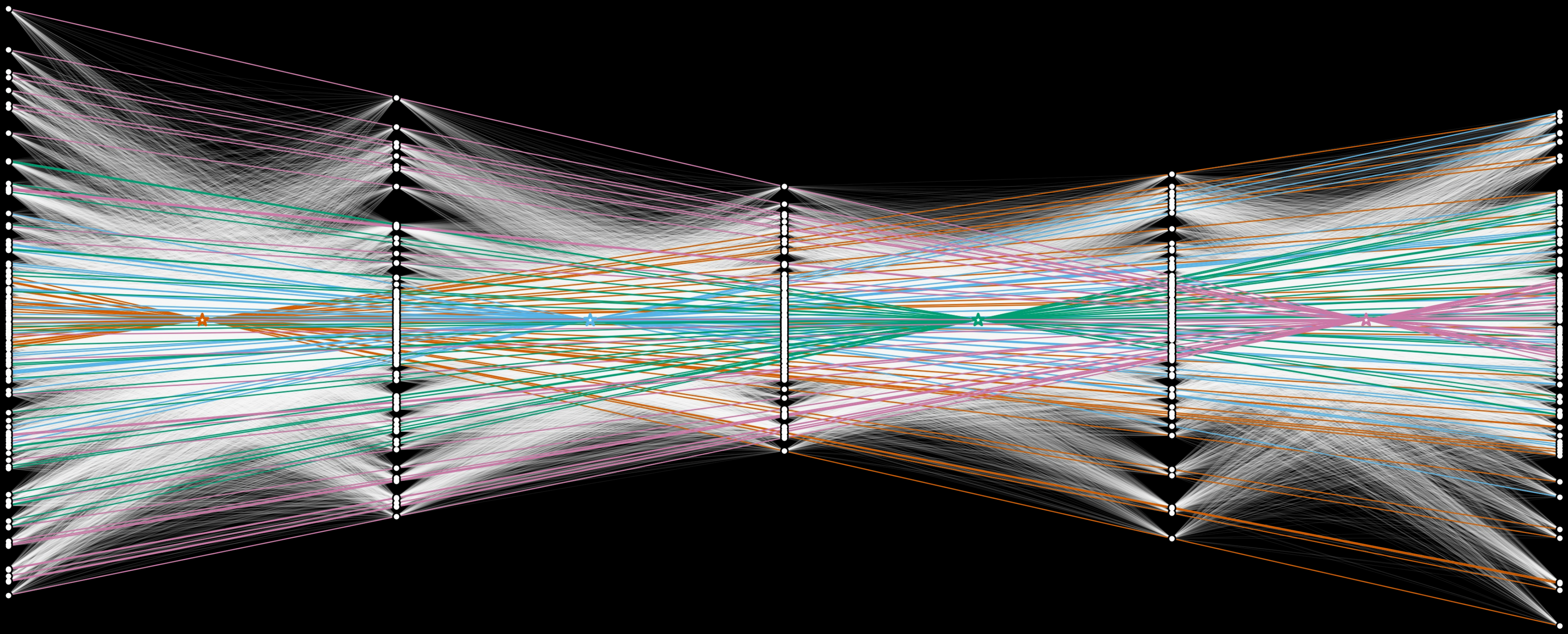


Testing Quantinuum Hardware Emulators

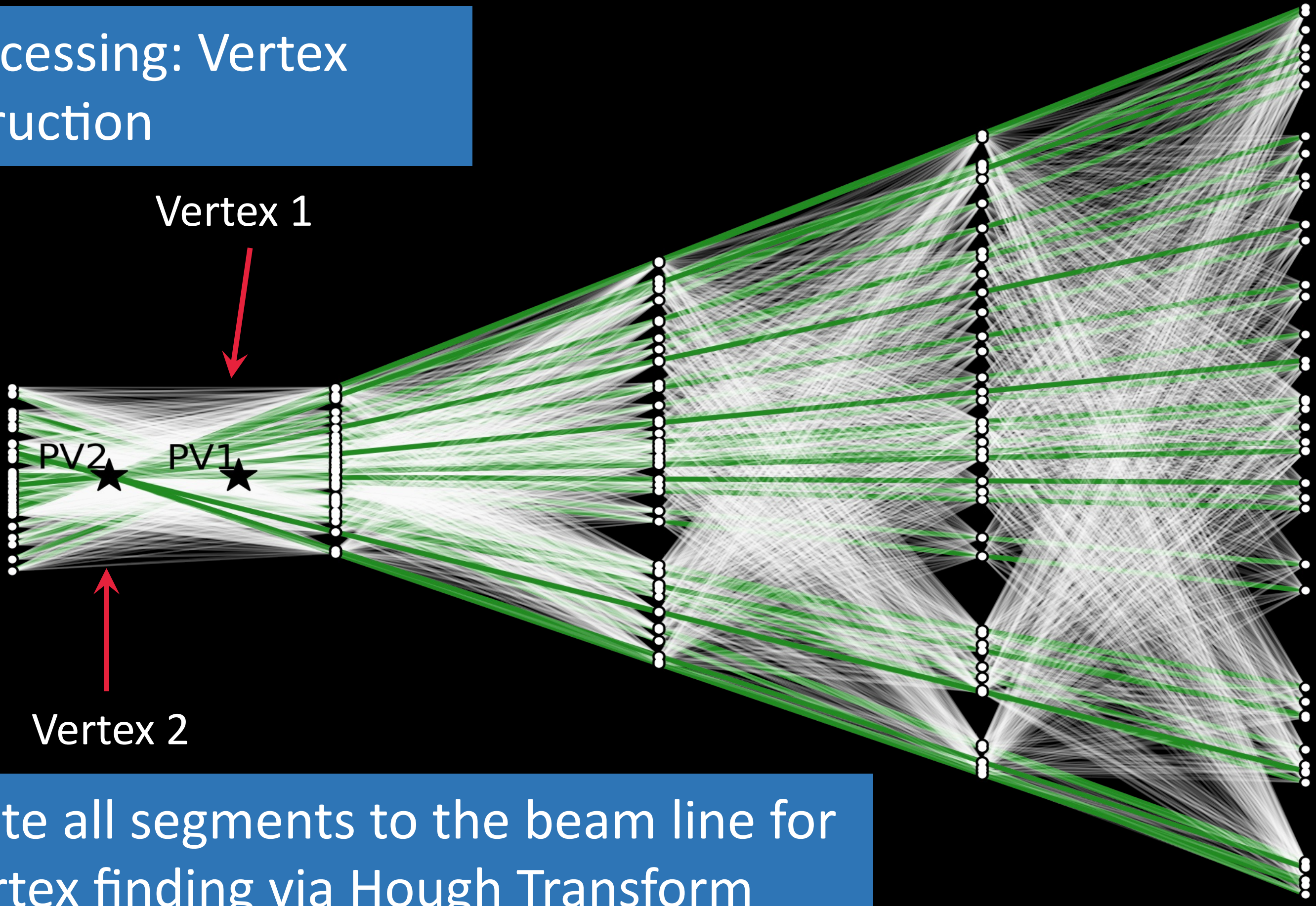






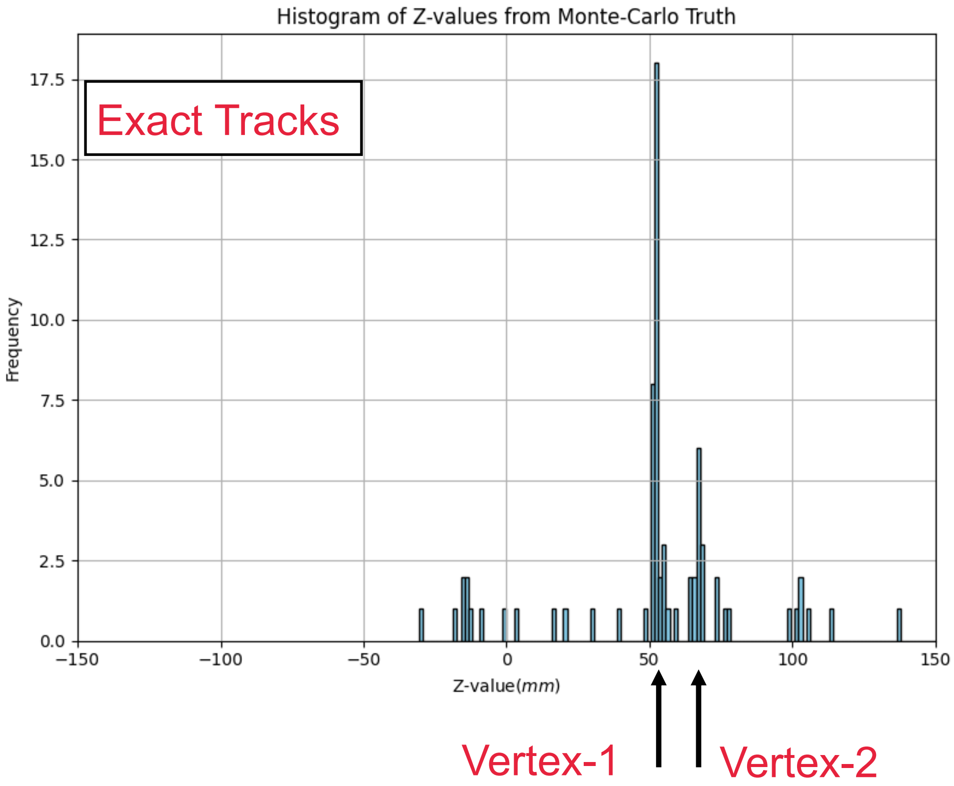


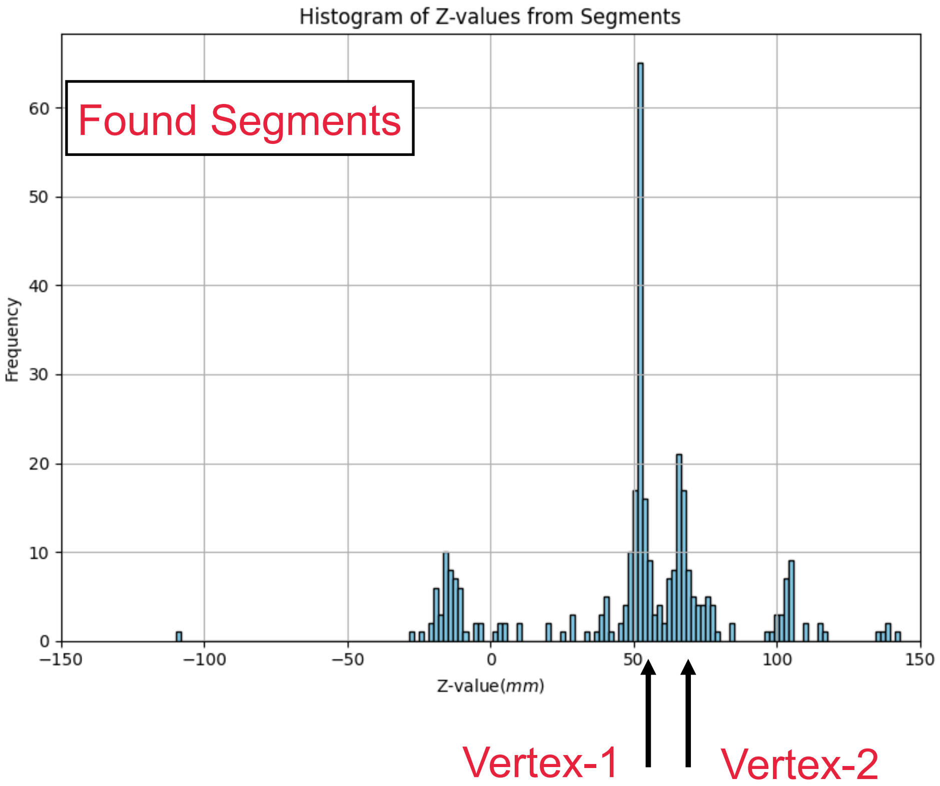
Post-processing: Vertex Reconstruction



Extrapolate all segments to the beam line for direct Vertex finding via Hough Transform

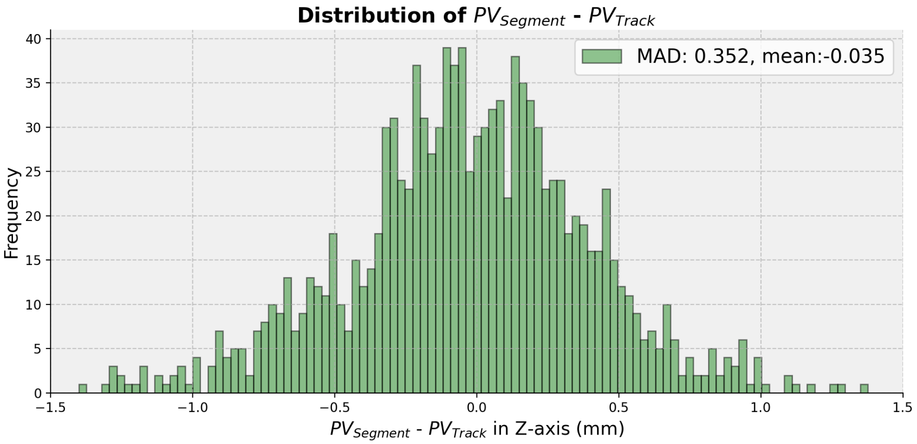
Vertex finding precision



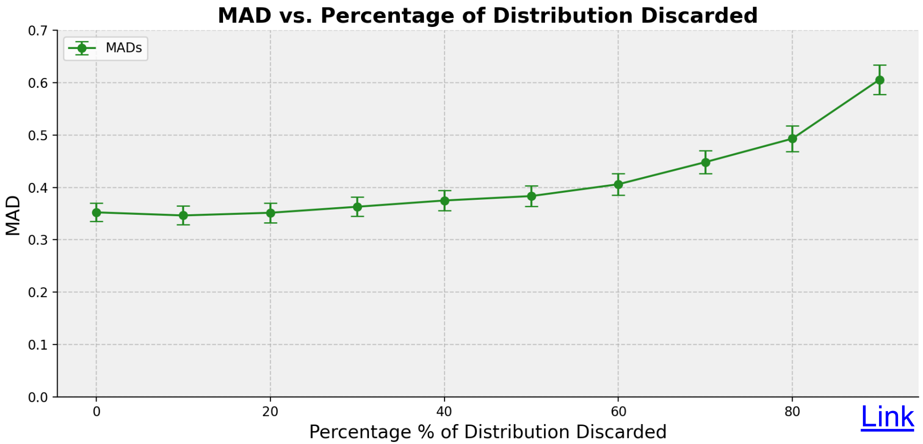


Vertex precision with fraction of segments

Distribution of precision of the extrapolated vertex point



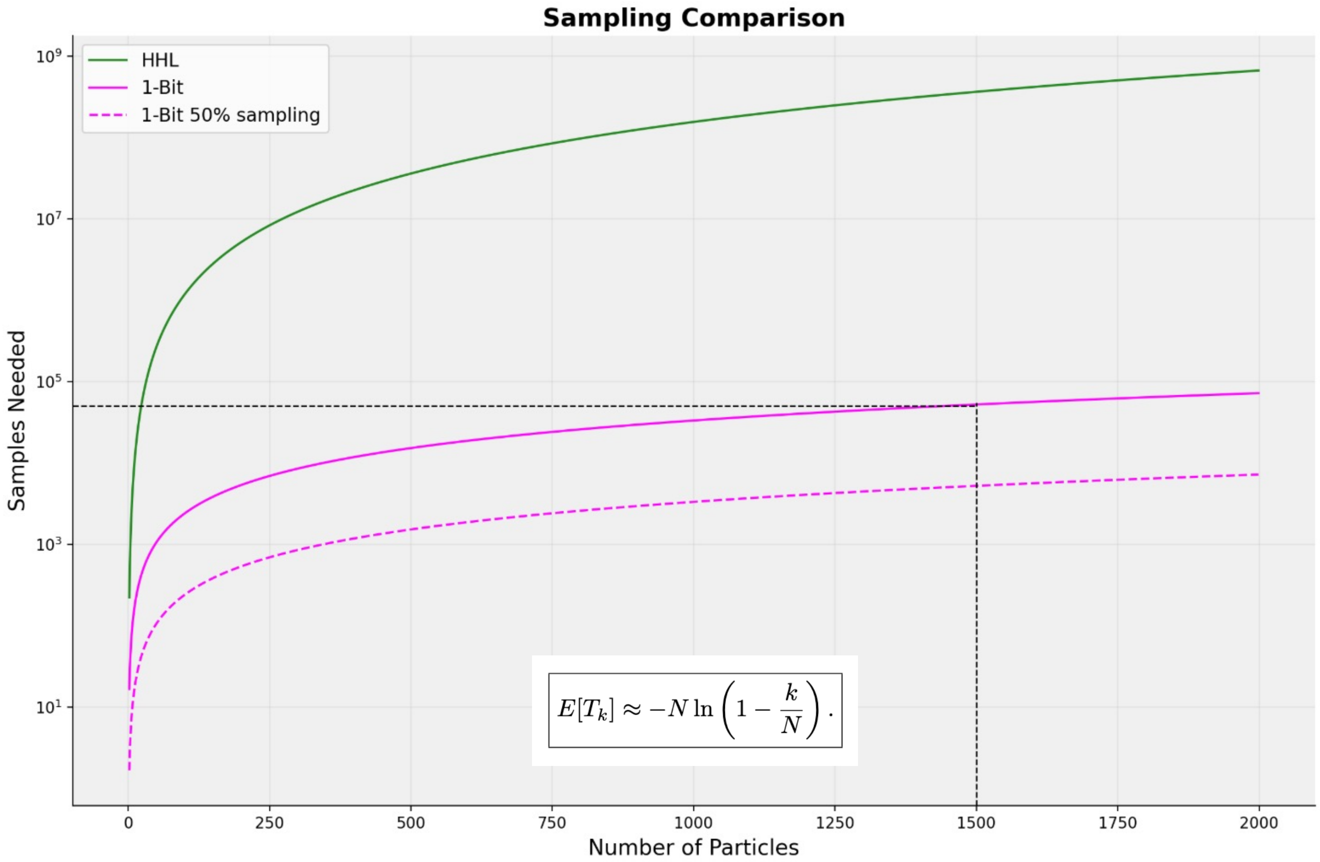
Mean Average Distance (MAD) of recontructed vertex versus the fraction of segments obtained in readout



Link

Requiring **only fraction (~50%) of all states** → strong reduction of number of readouts

Readout issue: 1-bit QPE



Remaining Tasks

$$\frac{1}{n(l-1)}$$

$n = Tracks, l = detecor\ layers$

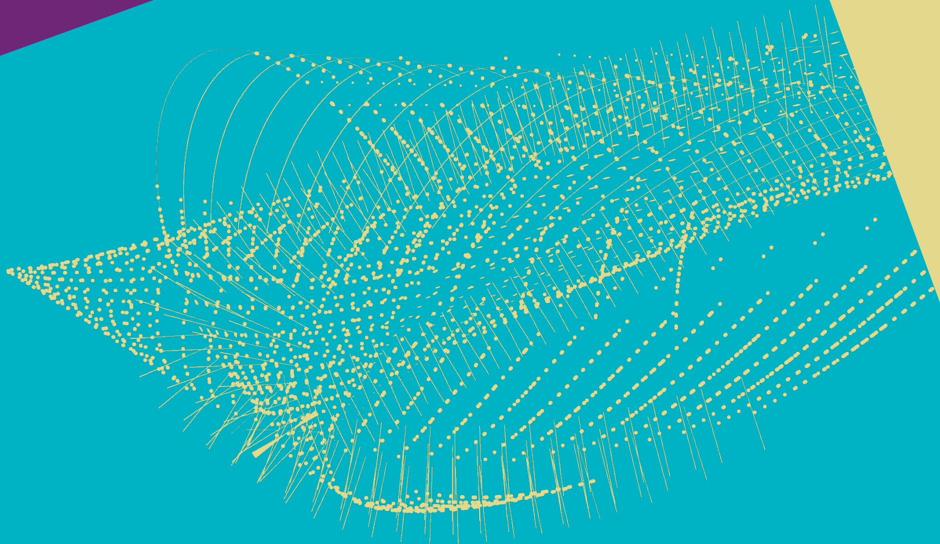
Test Toy with Realistic detector inefficiency

Writing

More efficient inversion algorithms QSVT

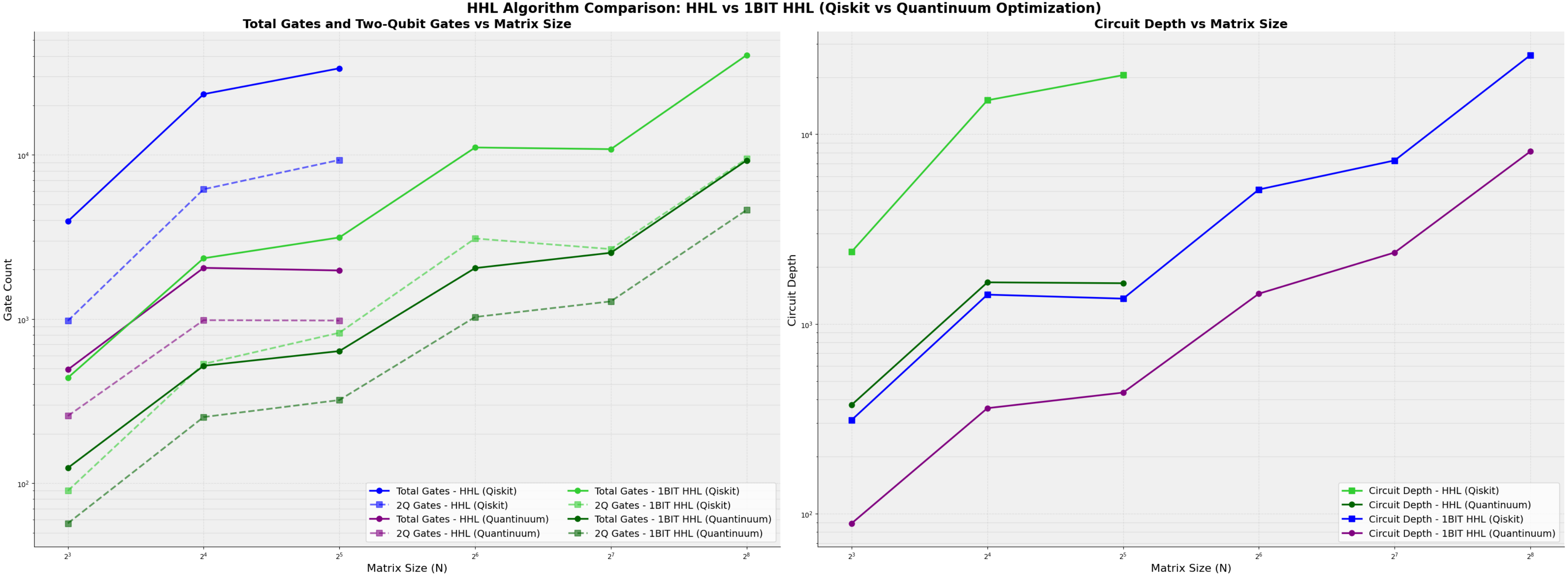
Quantum Method for PVs

Backup Slides



Xenofon Chiotopoulos
Nikhef and Maastricht University
Faster Day
October 2025

Circuit Transpiled for different hardware



Nikhef

Circuit Depth 1-Bit