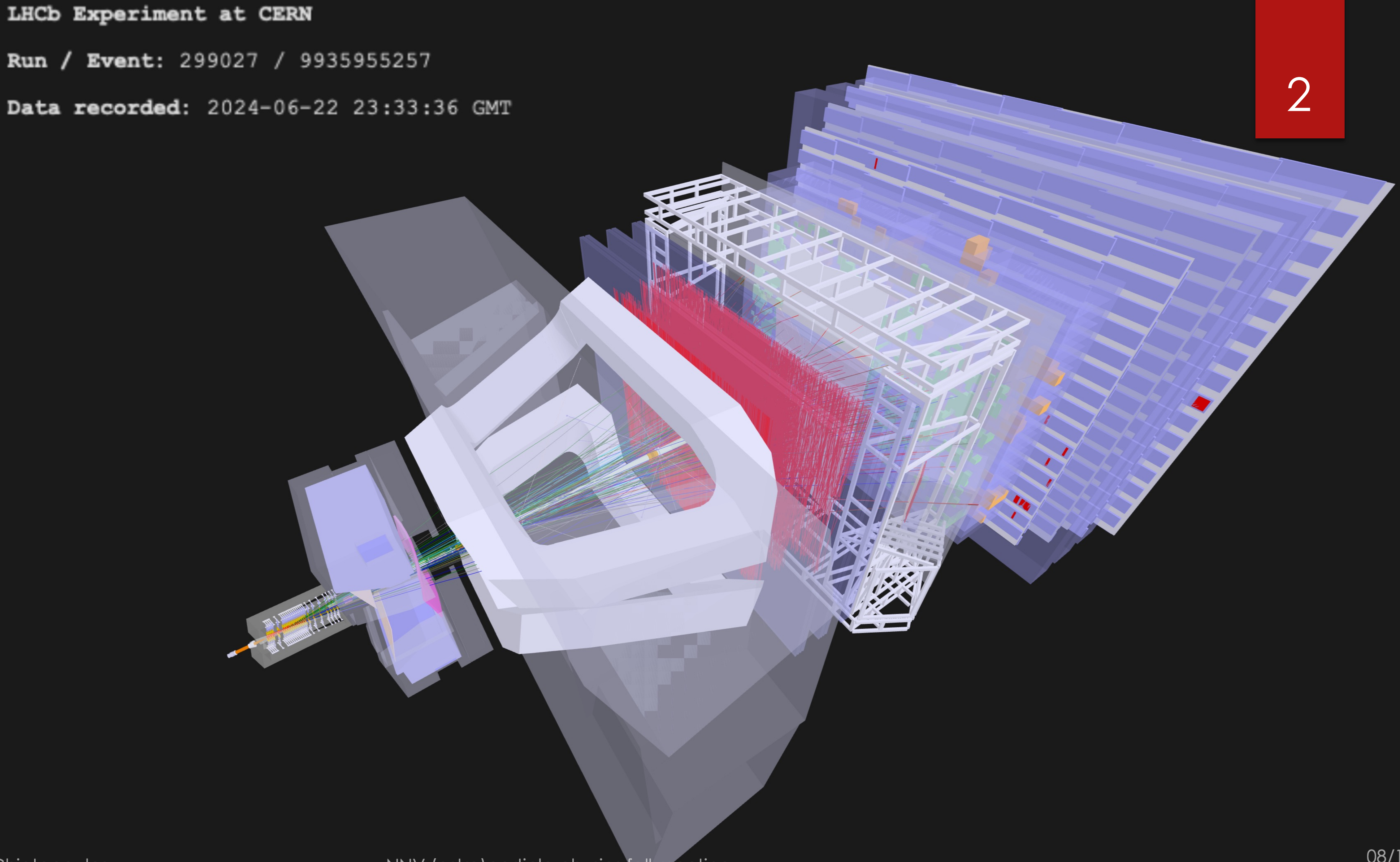
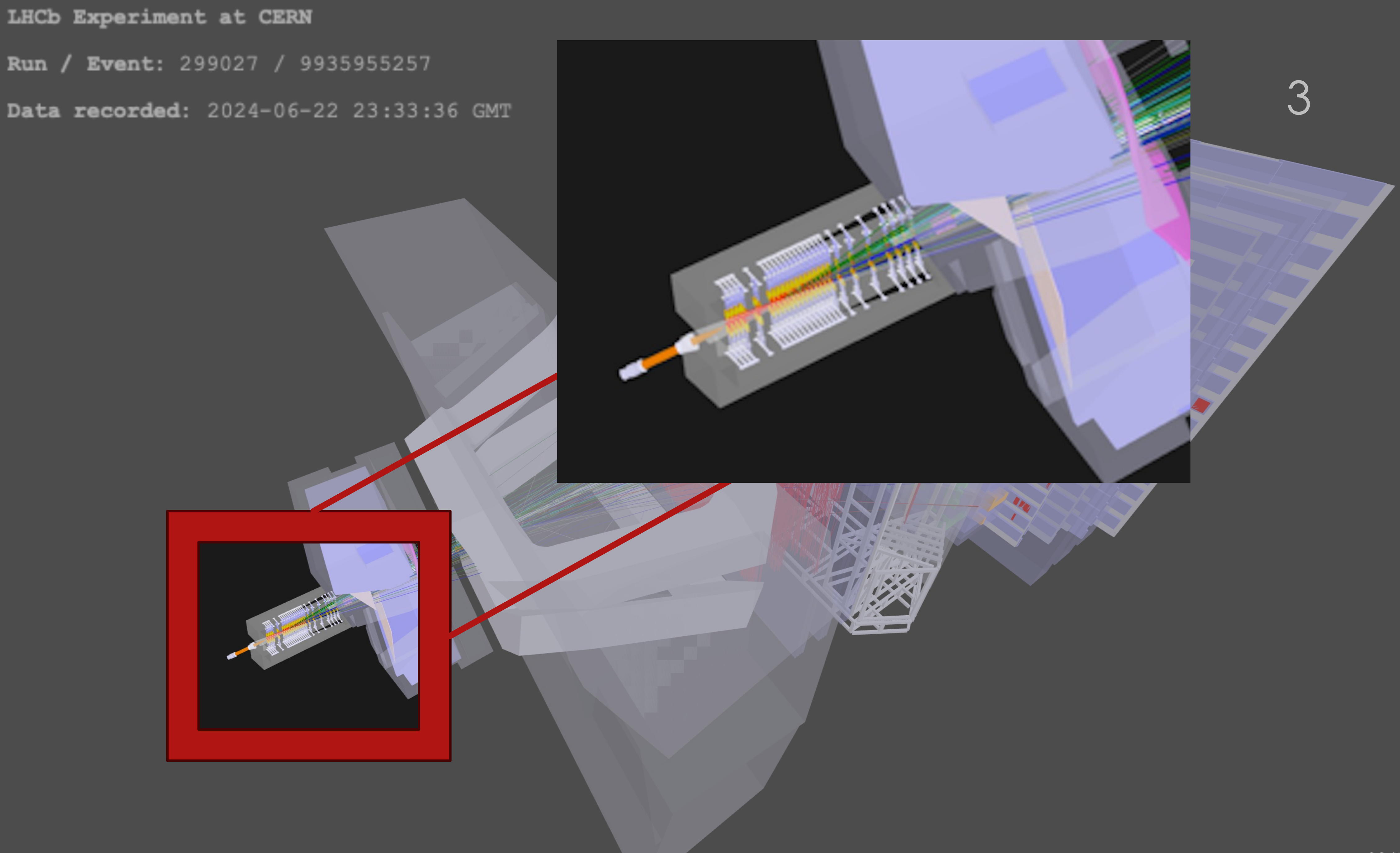
TrackHHL: LHCb Track Reconstruction using novel quantum algorithms

XENOFON CHIOTOPOULOS

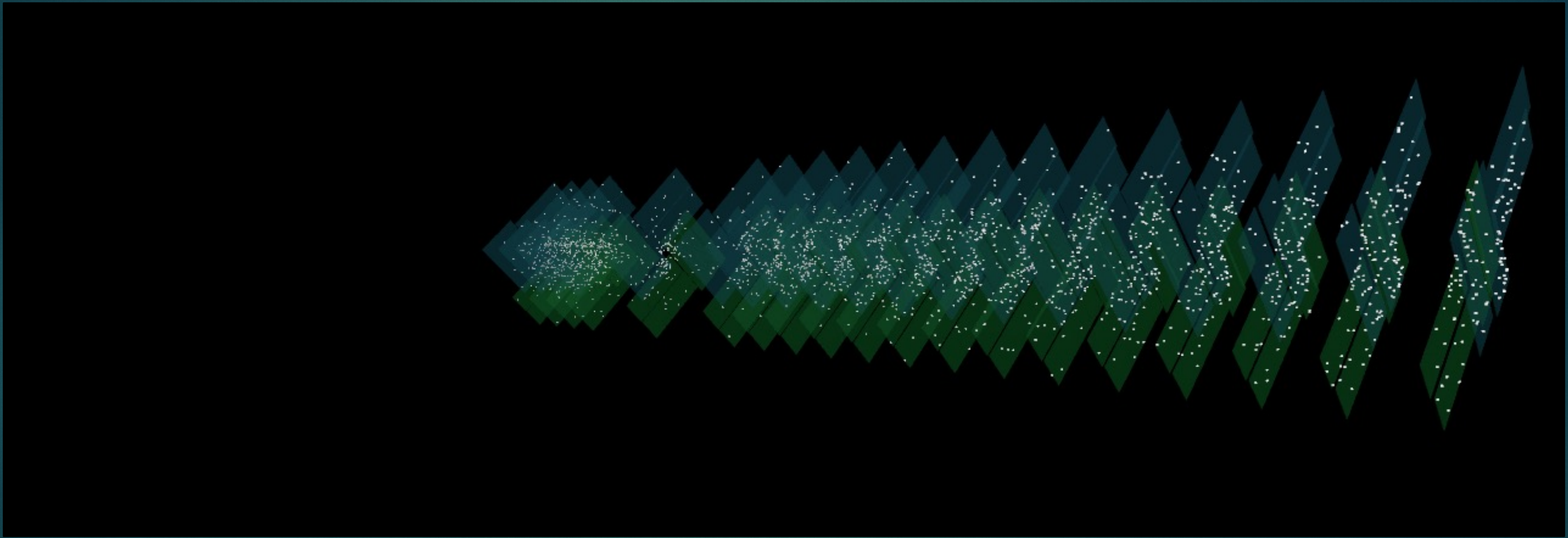
xenofon.chiotopoulos@maastrichtuniversity.nl

NNV Section for (astro)particle physics fall meeting


LHCb Experiment at CERN
Run / Event: 299027 / 9935955257
Data recorded: 2024-06-22 23:33:36 GMT

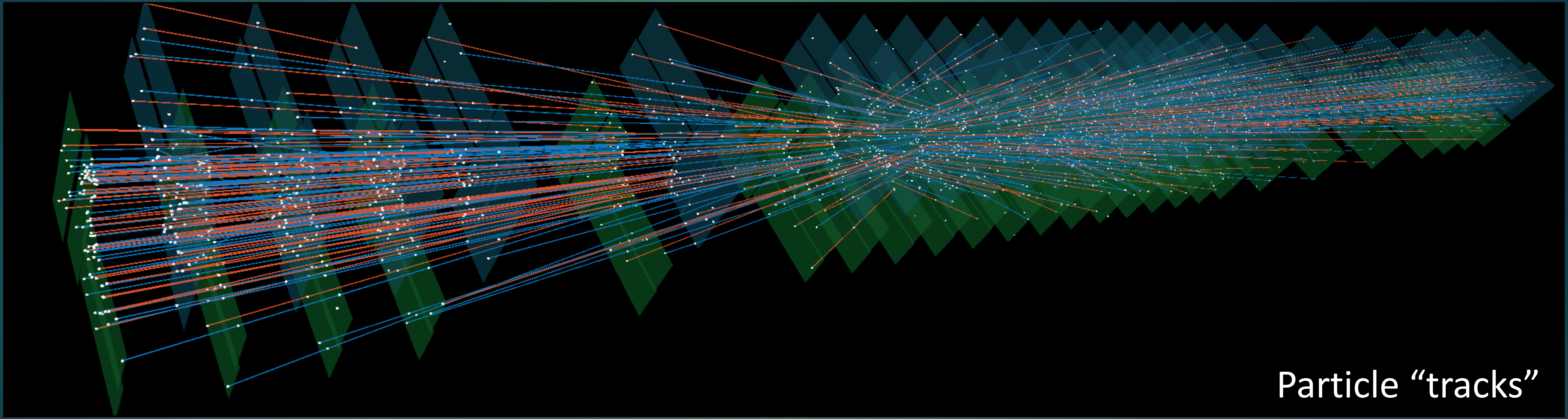

LHCb Experiment at CERN
Run / Event: 299027 / 9935955257
Data recorded: 2024-06-22 23:33:36 GMT

Hits in the VELO



Creator: Davide Nicotra

Tracks in the VELO

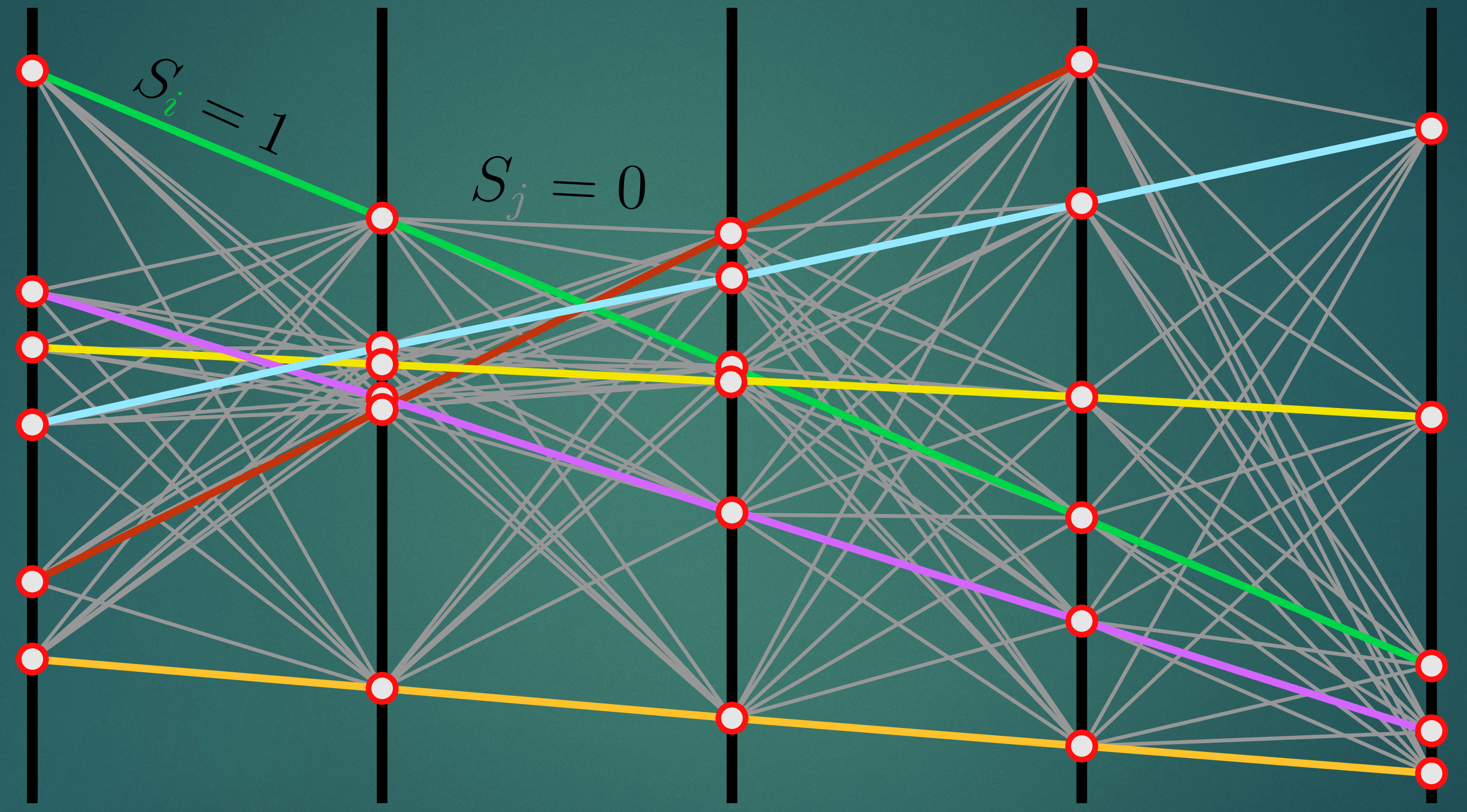


Creator: Davide Nicotra

New Quantum Computing algorithm for Track Reconstruction

Translation For Quantum Advantage

[arXiv:2308.00619] [JINST]



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

Translation For Quantum Advantage

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

(a) (b) (c)

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

- **(a) Angular term:** assigns values for straight doublets
- **(b) Regularization term**: makes the spectrum of A positive
- **(c) Gap term**: ensures gap in the solution spectrum

Translation For Quantum Advantage

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

Relaxation of binary S values allows

$$\nabla_S H = 0$$

$$-AS + b = 0$$

$$AS = b$$

Translation For Quantum Advantage

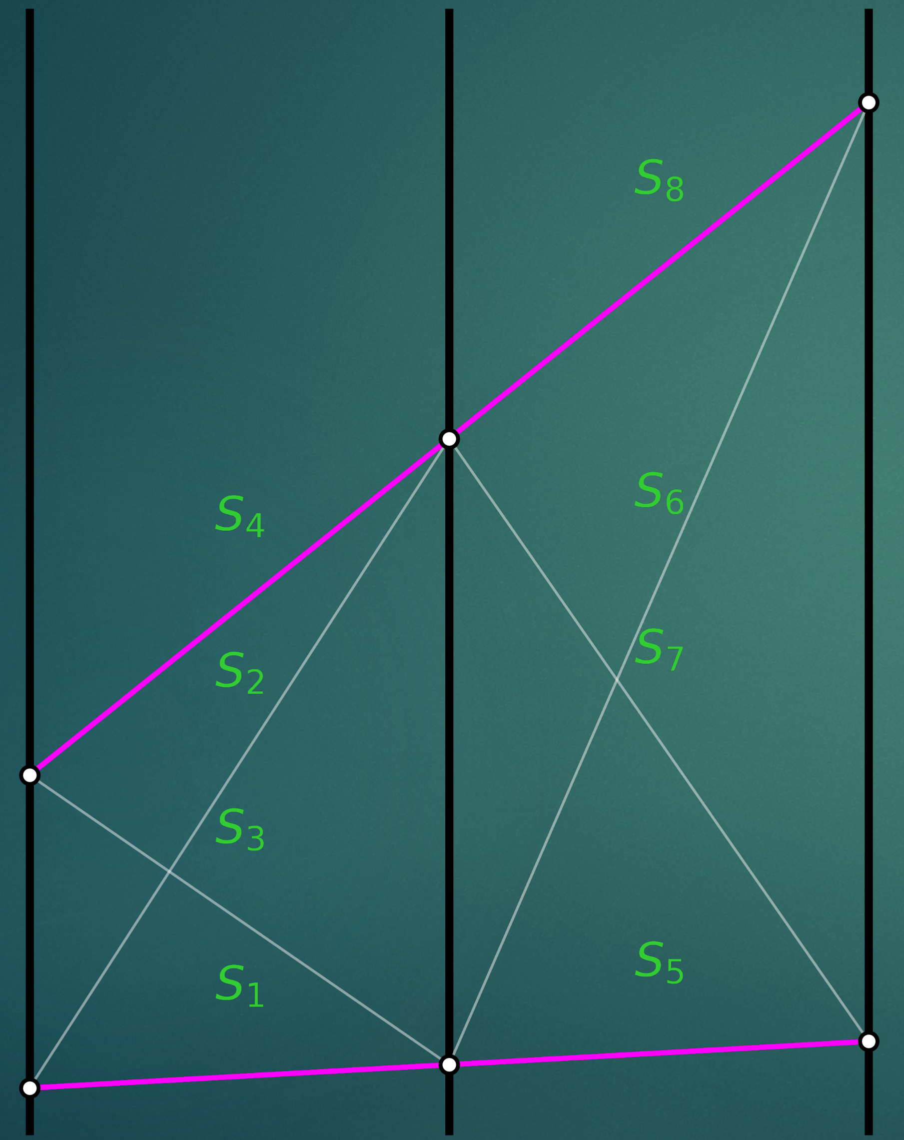
$$\mathcal{H}(\mathbf{S}) = -\frac{1}{2}\left[\sum_{abc} f(\theta_{abc}, \varepsilon) S_{ab} S_{bc} + \gamma \sum_{ab} S_{ab}^2 + \delta \sum_{ab} (1 - 2S_{ab})^2\right]$$

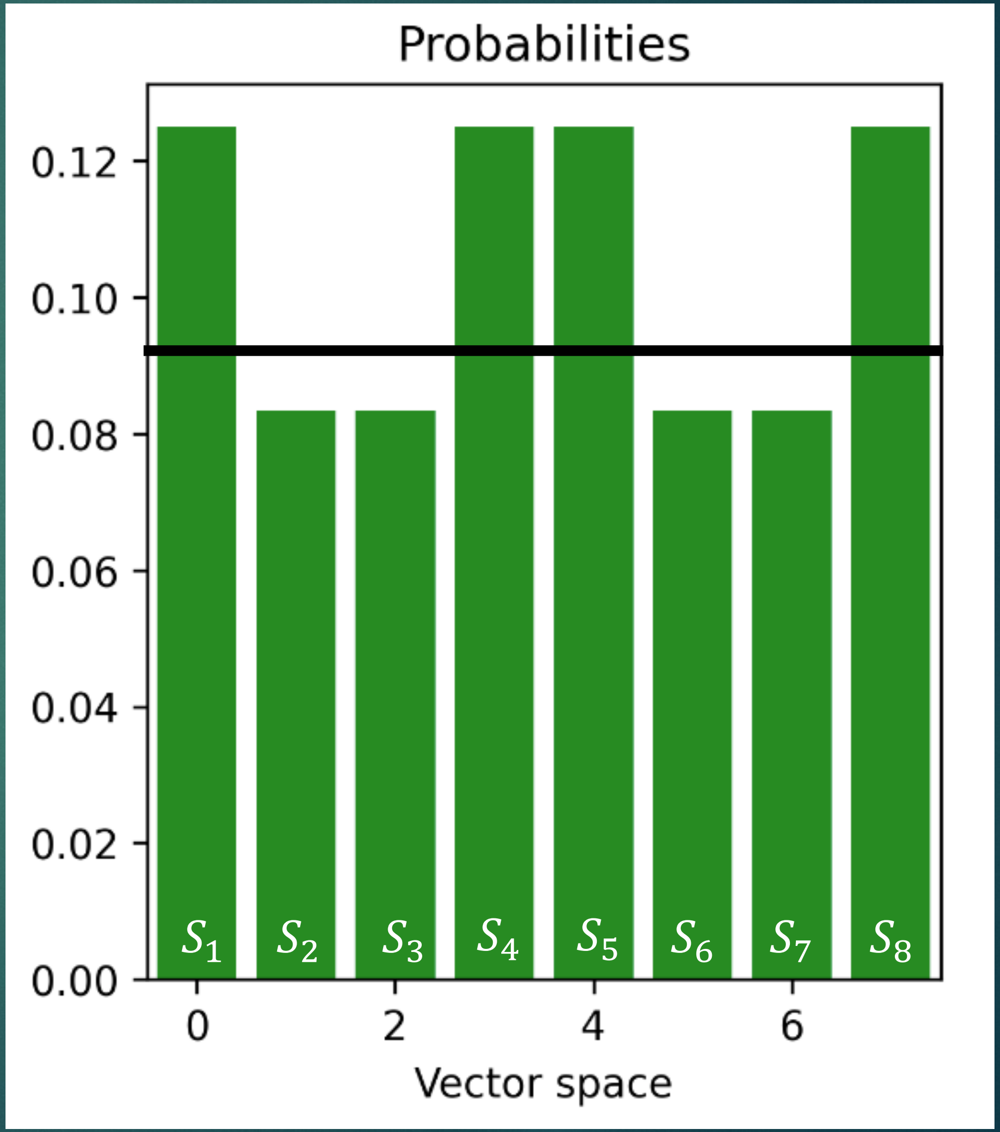
$$AS = b$$

$$A =$$

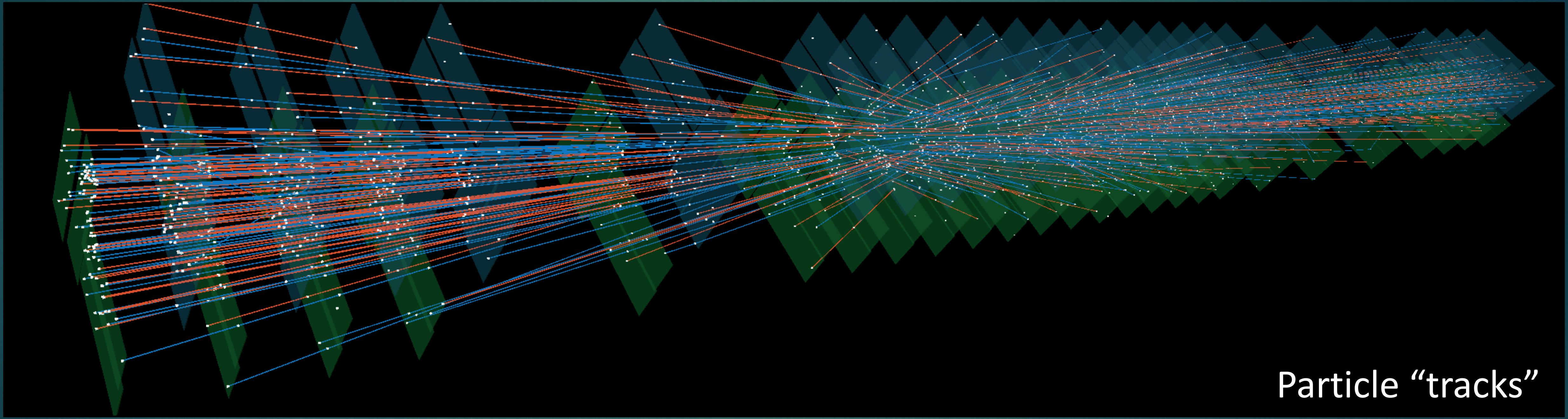
$$b =$$

Most Trivial Tracking Scenario



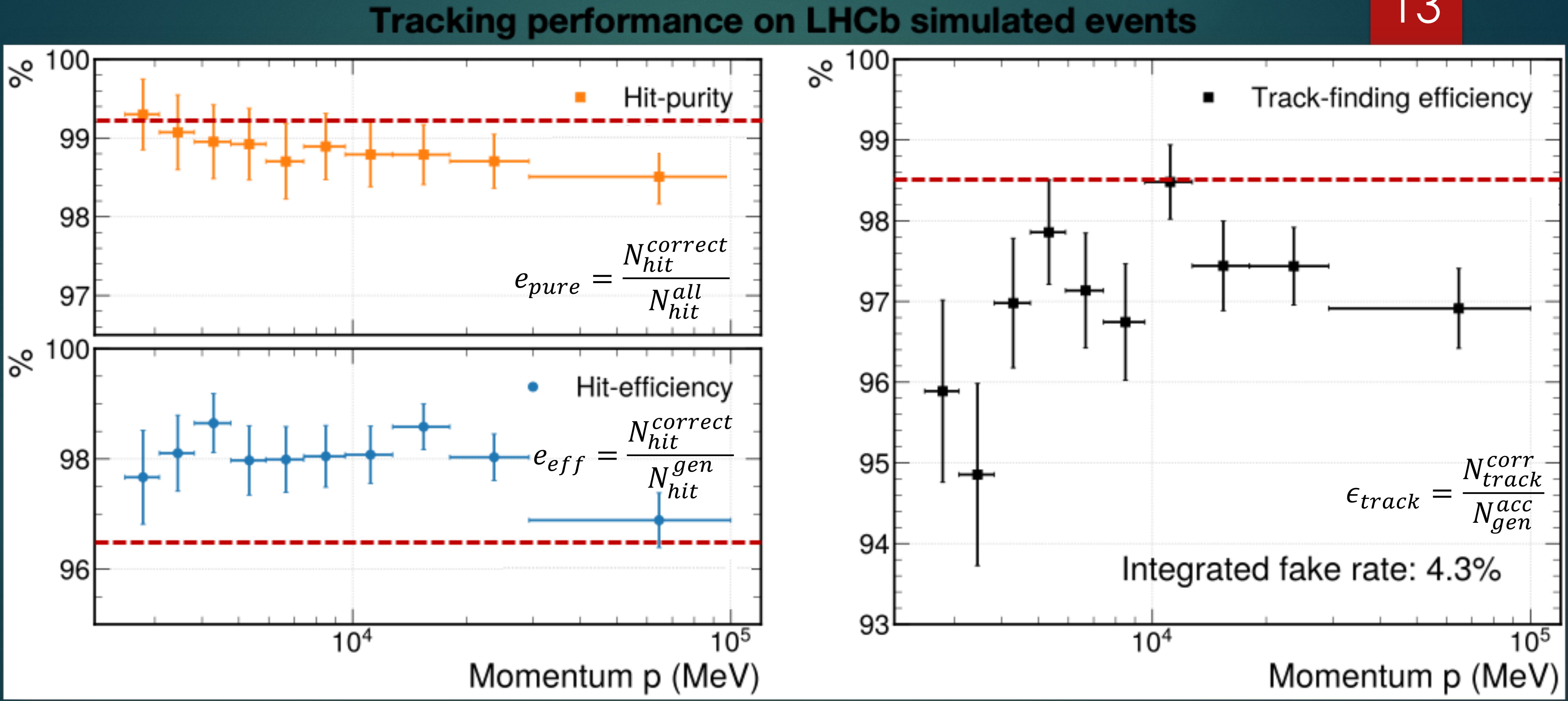


Solving it Classically



Creator: Davide Nicotra

Solving it Classically



Benchmarked on a Classical Equivalent of the Quantum algorithm [arXiv:2308.00619], [JINST]

HHL(Harrow–Hassidim–Lloyd) algorithm

$$AS = b$$

Classical Complexity: $\mathcal{O}(n^2)$*

HHL Complexity: $\mathcal{O}(\kappa^2 \log n)$

- $n = input\ matrix\ size$
- $k = condition\ number$

System size scales with:

- $\mathrm{n} = p^2 \times Average\ Hits\ Per\ Track$
- $p = particles\ in\ detector$

n	Qubits	Depth	2-qubit gates
8	8	12 071	5 538
12	10	185 817	93 213
18	12	1 665 771	834 417
27	12	1 714 534	840 780
32	12	901 255	442 694
48	14	14 197 046	7 110 044
50	14	14 515 229	7 107 317

HHL(Harrow–Hassidim–Lloyd) algorithm

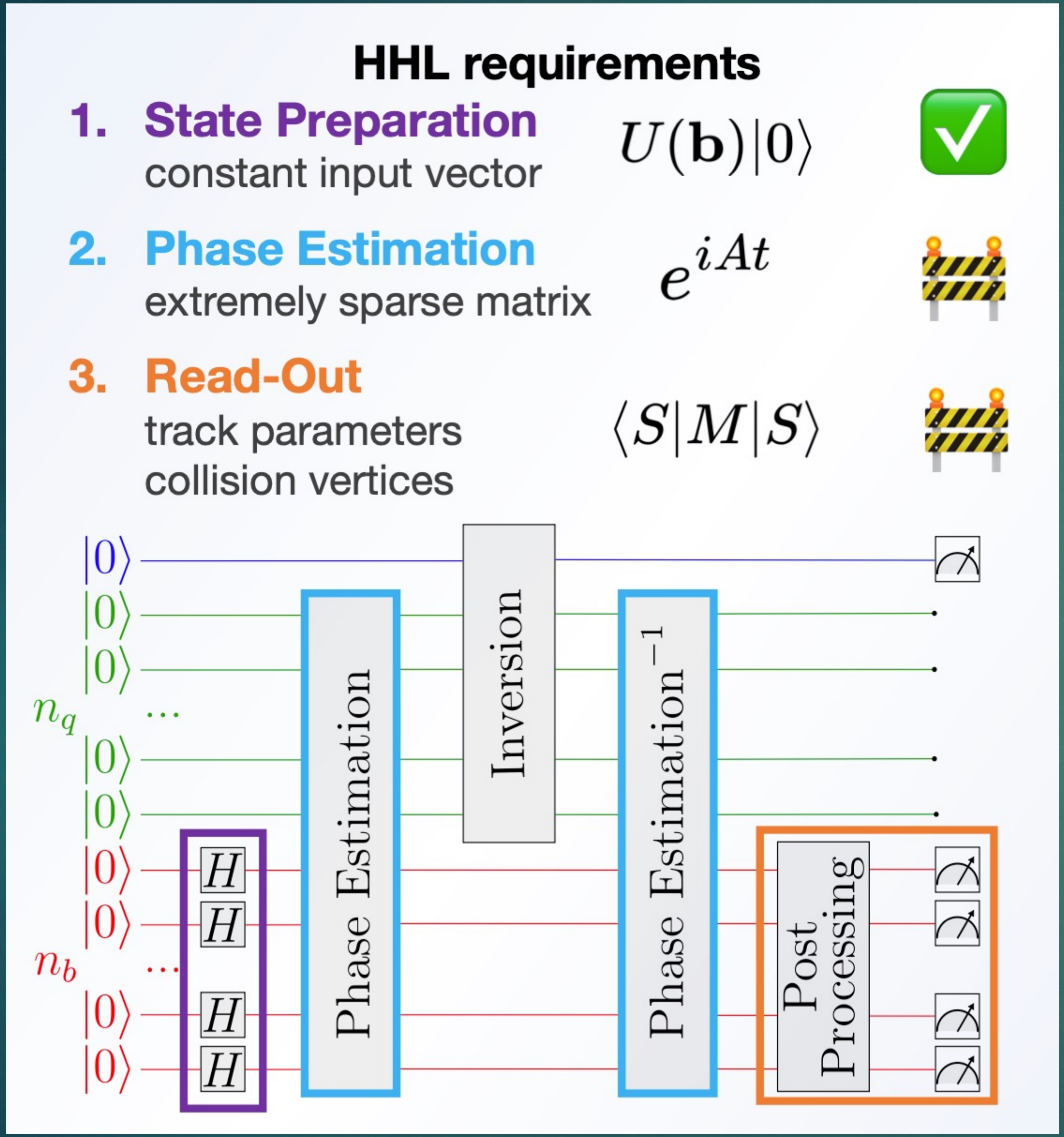
$$AS = b$$

Current Best IBM:
~3000 gates

n	Qubits	Depth	2-qubit gates
8	8	12 071	5 538
12	10	185 817	93 213
18	12	1 665 771	834 417
27	12	1 714 534	840 780
32	12	901 255	442 694
48	14	14 197 046	7 110 044
50	14	14 515 229	7 107 317

System size scales with:

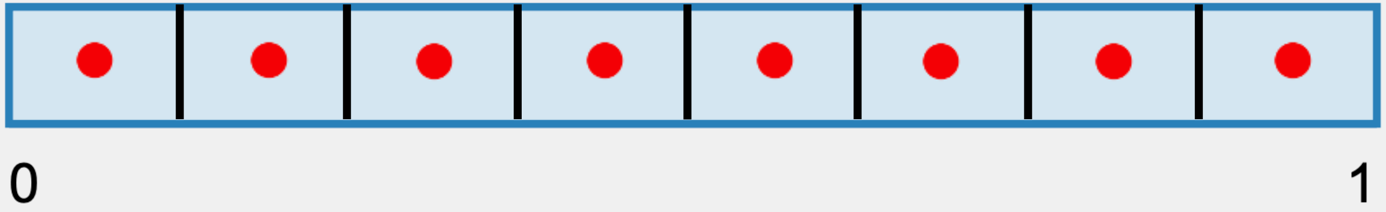
- $\text{n} = p^2 \times Average\ Hits\ Per\ Track$
- $p = particles\ in\ detector$



Challenges with HHL

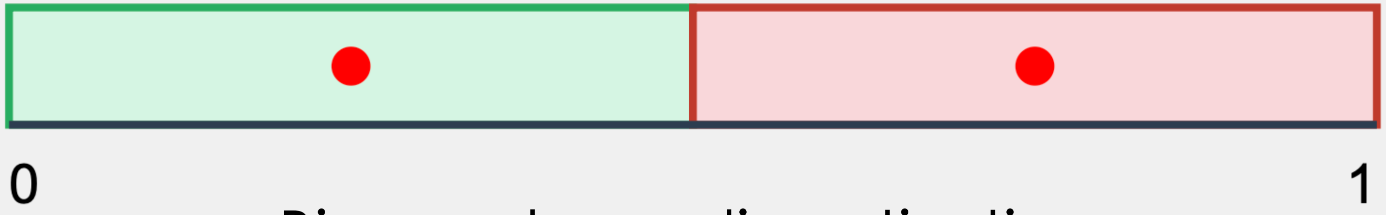
1-Bit Phase Estimation

3-Bit Phase Estimation



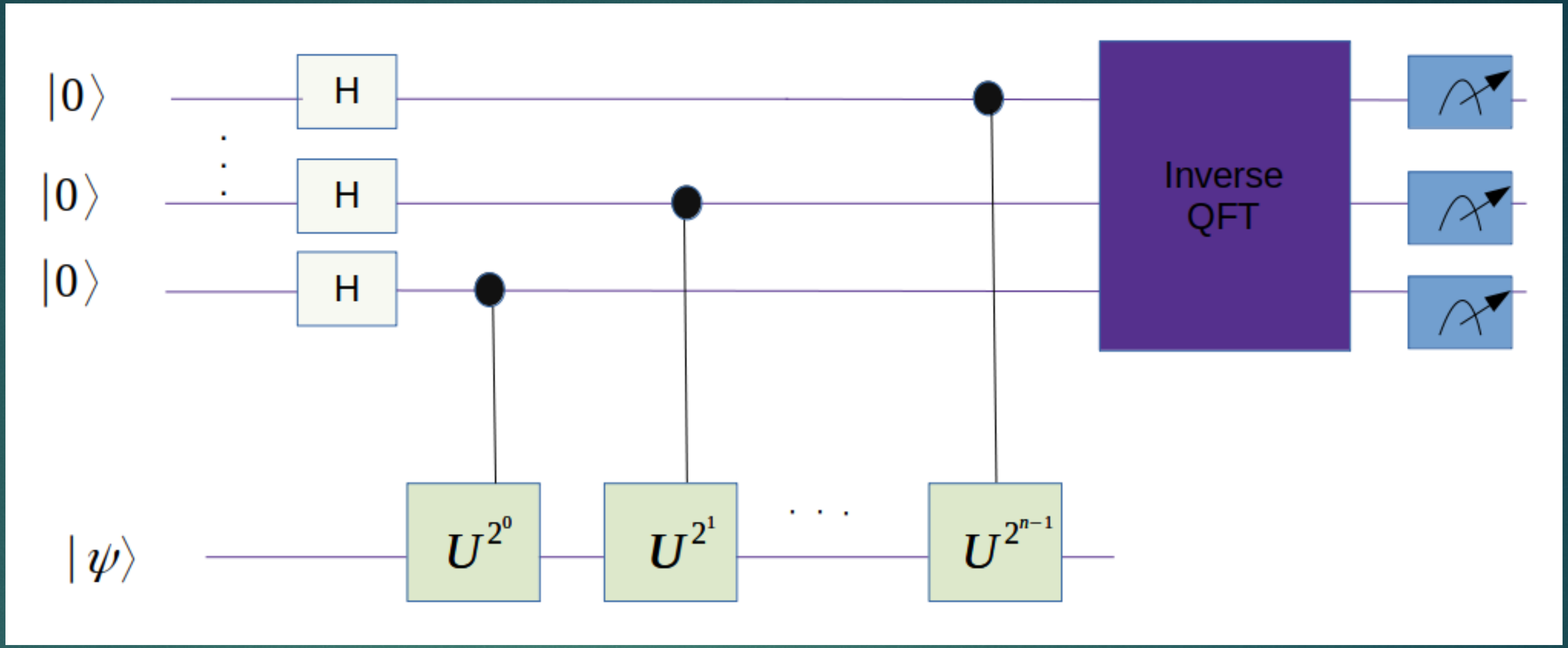
8 possible phase discretization

1-Bit Phase Estimation

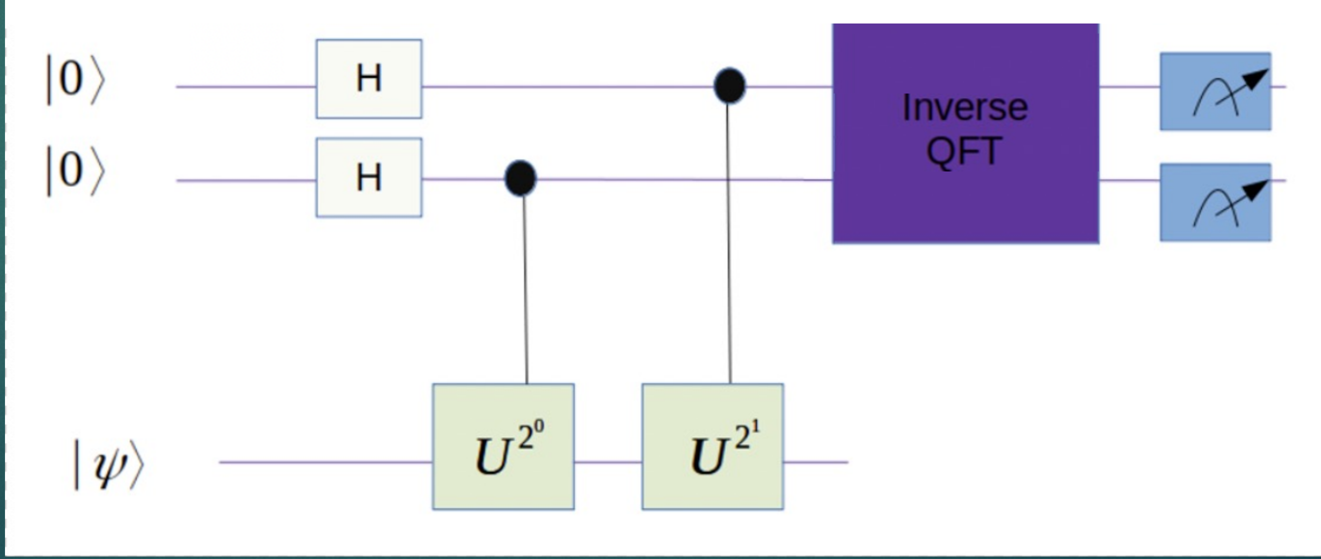


Binary phase discretization

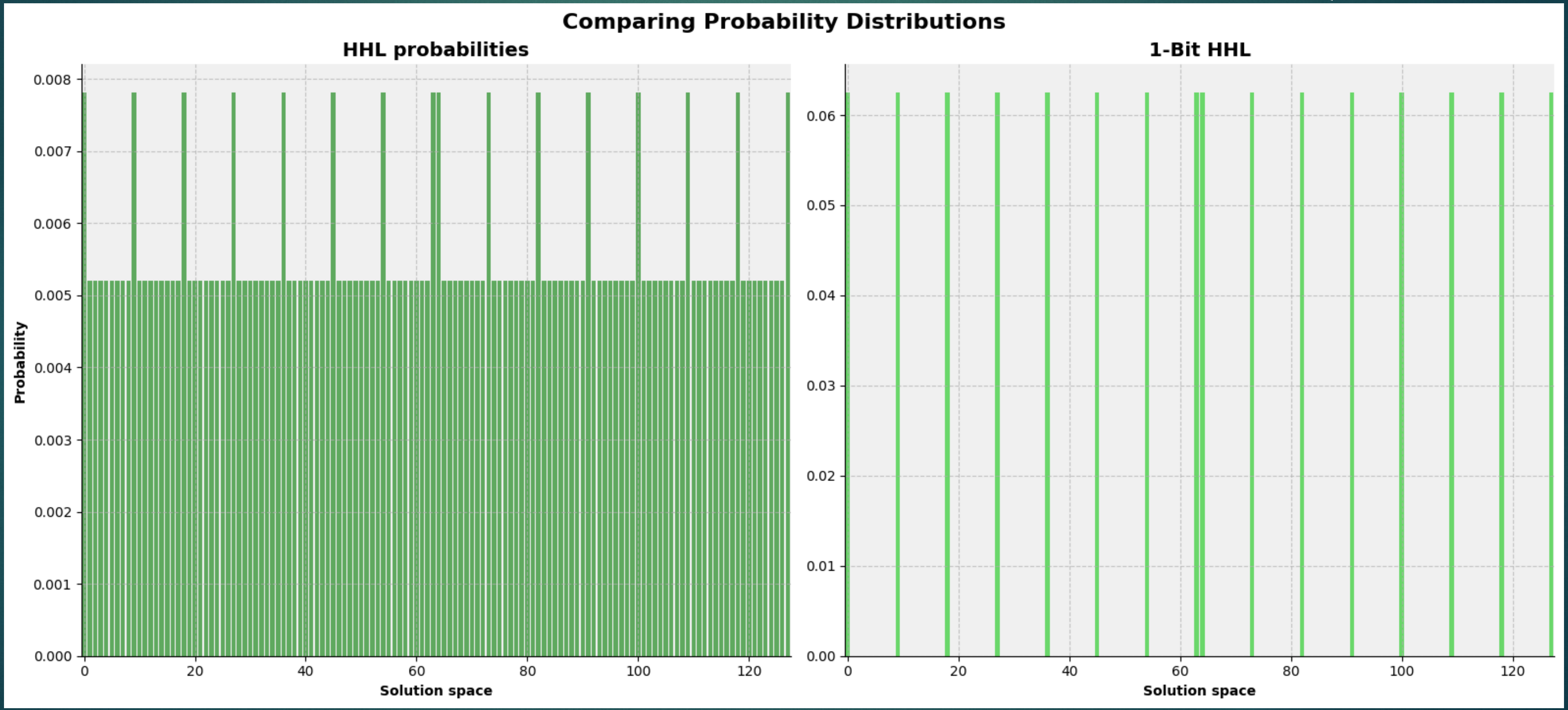
N-Bit QPE



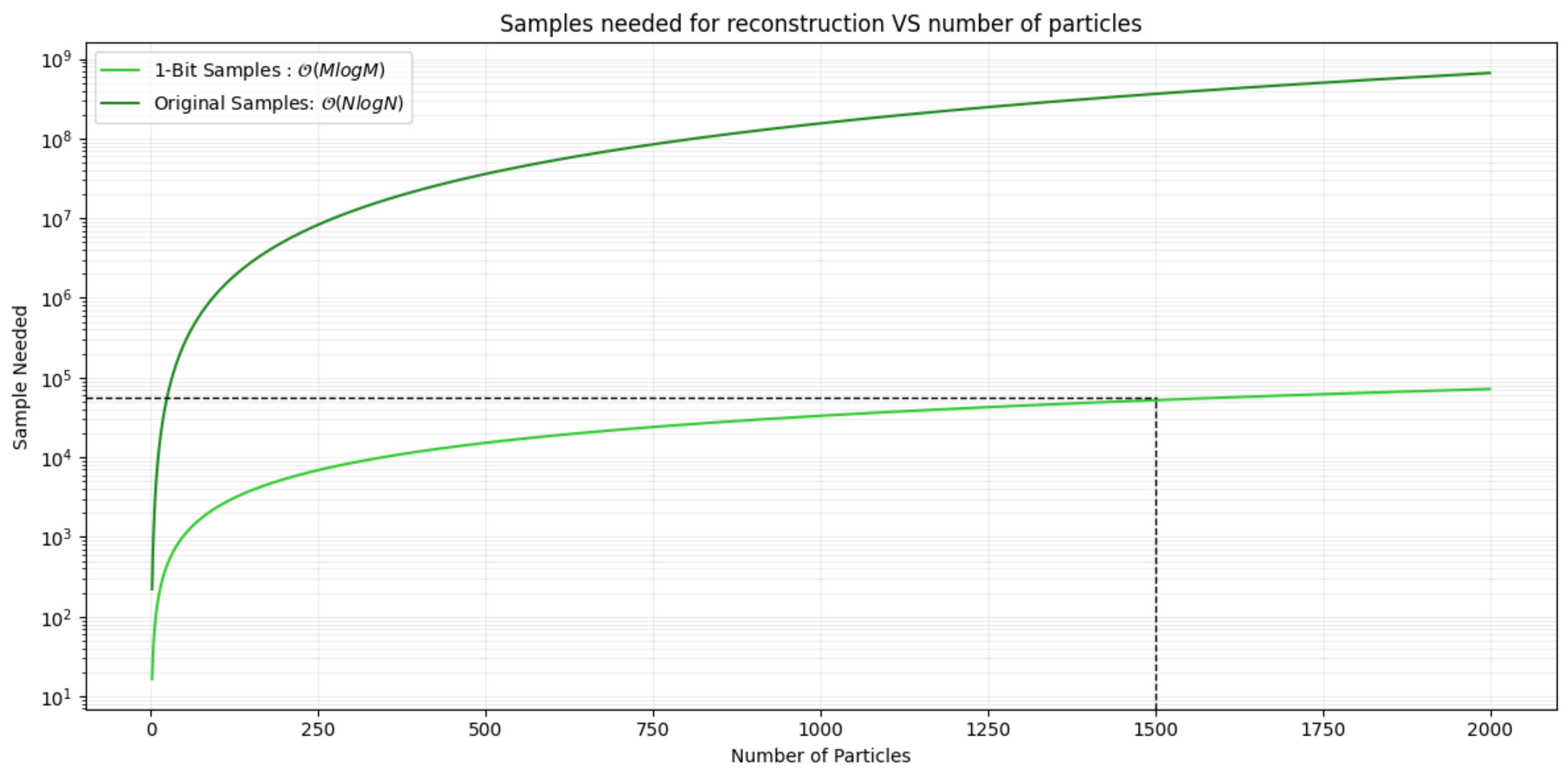
1-Bit QPE



Trying to solve our Phase Estimation Problem



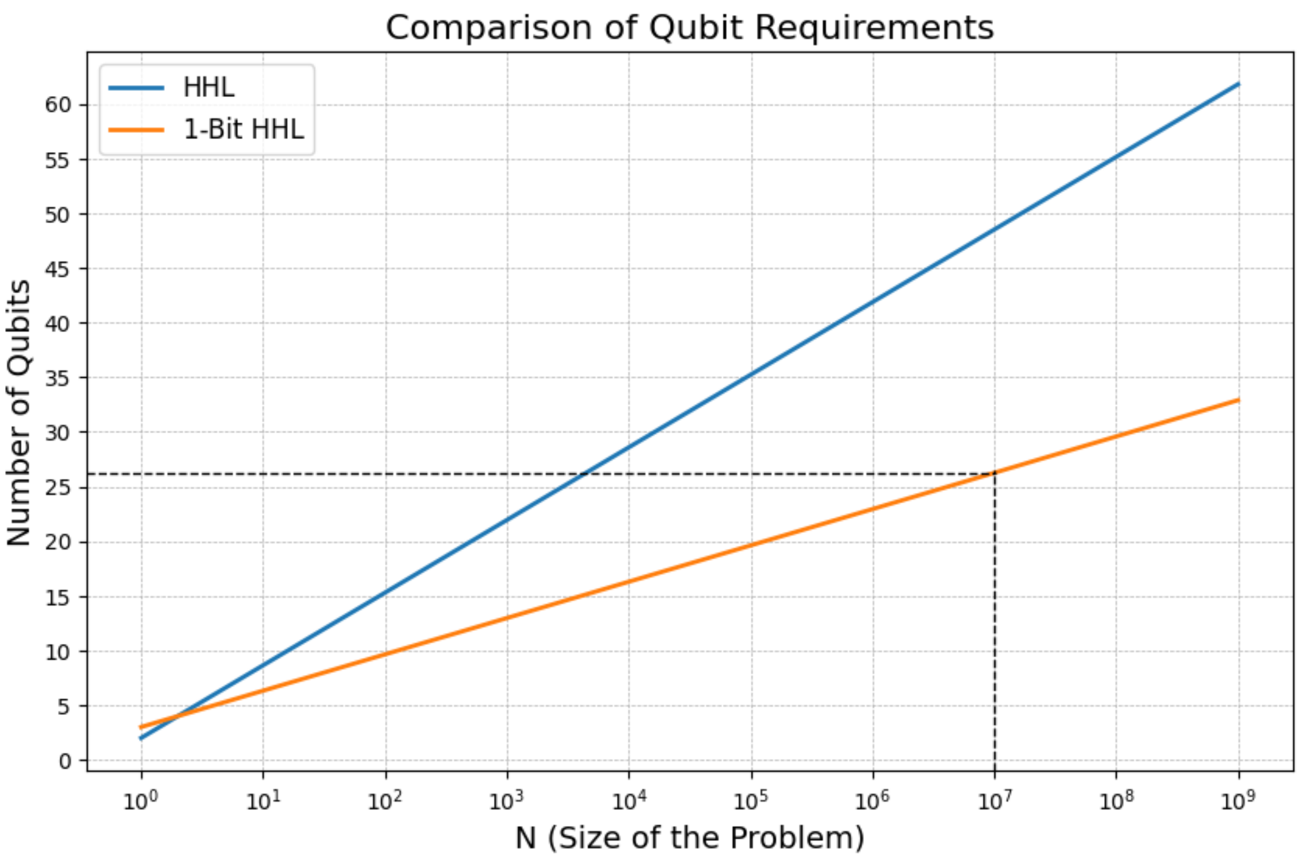
Output Problem



- Event with 1500 particles needs $10^{4.8}$ samples for 1-Bit
- Event with 1500 particles needs $10^{8.6}$ samples for HHL

- $M = p \times Average\ Hits\ Per\ Track$
- $N = p^2 \times Average\ Hits\ Per\ Track$

Qubit Reduction



- $2\log_2 N + 2$ qubits for HHL
- $\log_2 N + 3$ qubits for 1-Bit HHL

n	10^1	10^2	10^3	10^4	10^5	10^6	10^7	10^8	10^9
Particles	1	5	15	50	158	500	1581	5000	15811

*5 Hits Per Track Assumption

Solving our Phase Estimation Problem

1-Bit Phase Estimation

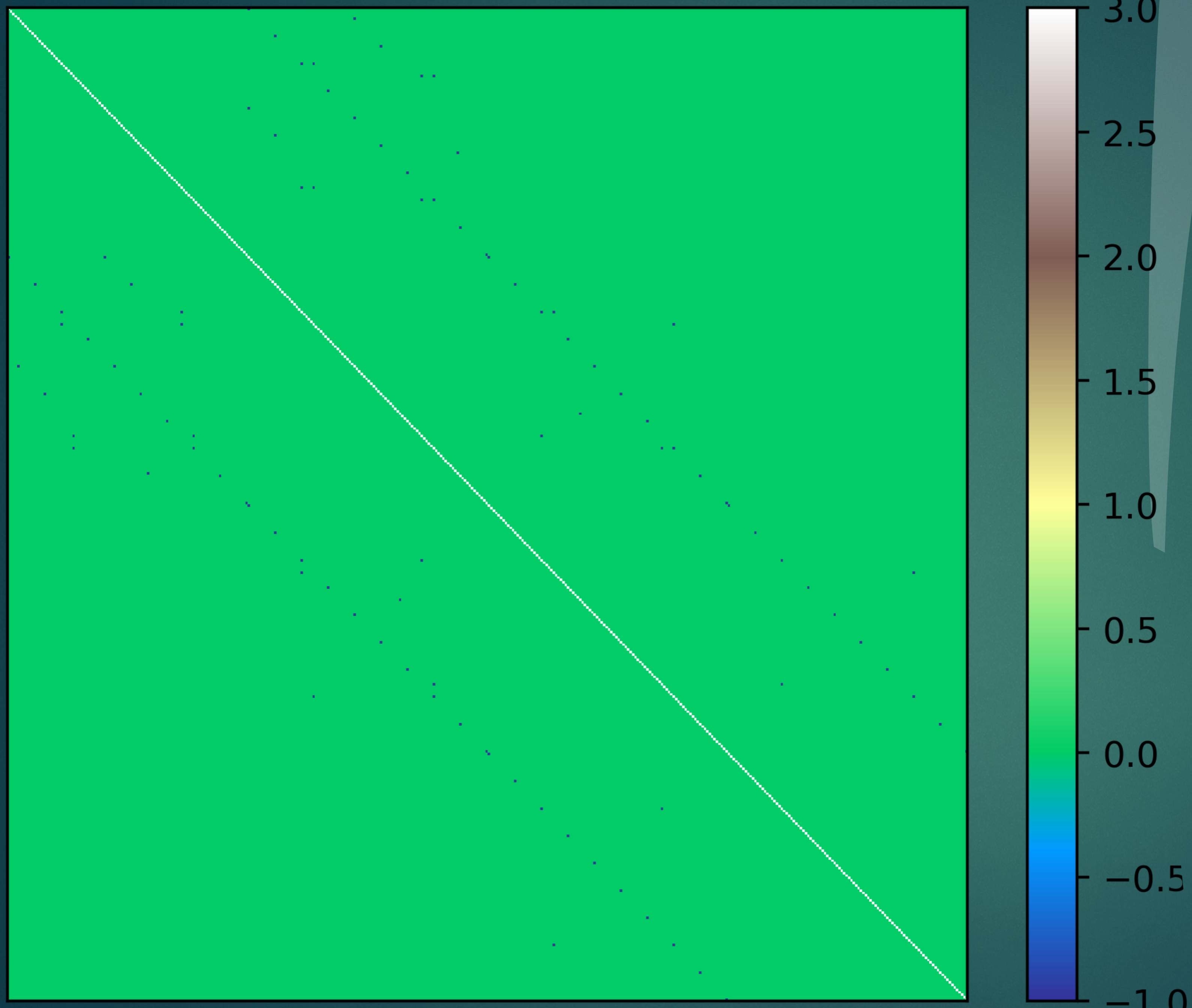
n	Qubits	Qubits*	Depth	Depth*	2-qubit gates	2-qubit gates*
8	8	6	12 071	371	5 538	219
12	10	7	185 817	2 005	93 213	1 264
18	12	8	1 665 771	7 732	834 417	4 609
27	12	8	1 714 534	14 512	840 780	8 780
32	12	8	901 255	1 229	442 694	749
48	14	9	14 197 046	5 439	7 110 044	3 429
50	14	9	14 515 229	24 172	7 107 317	14 804

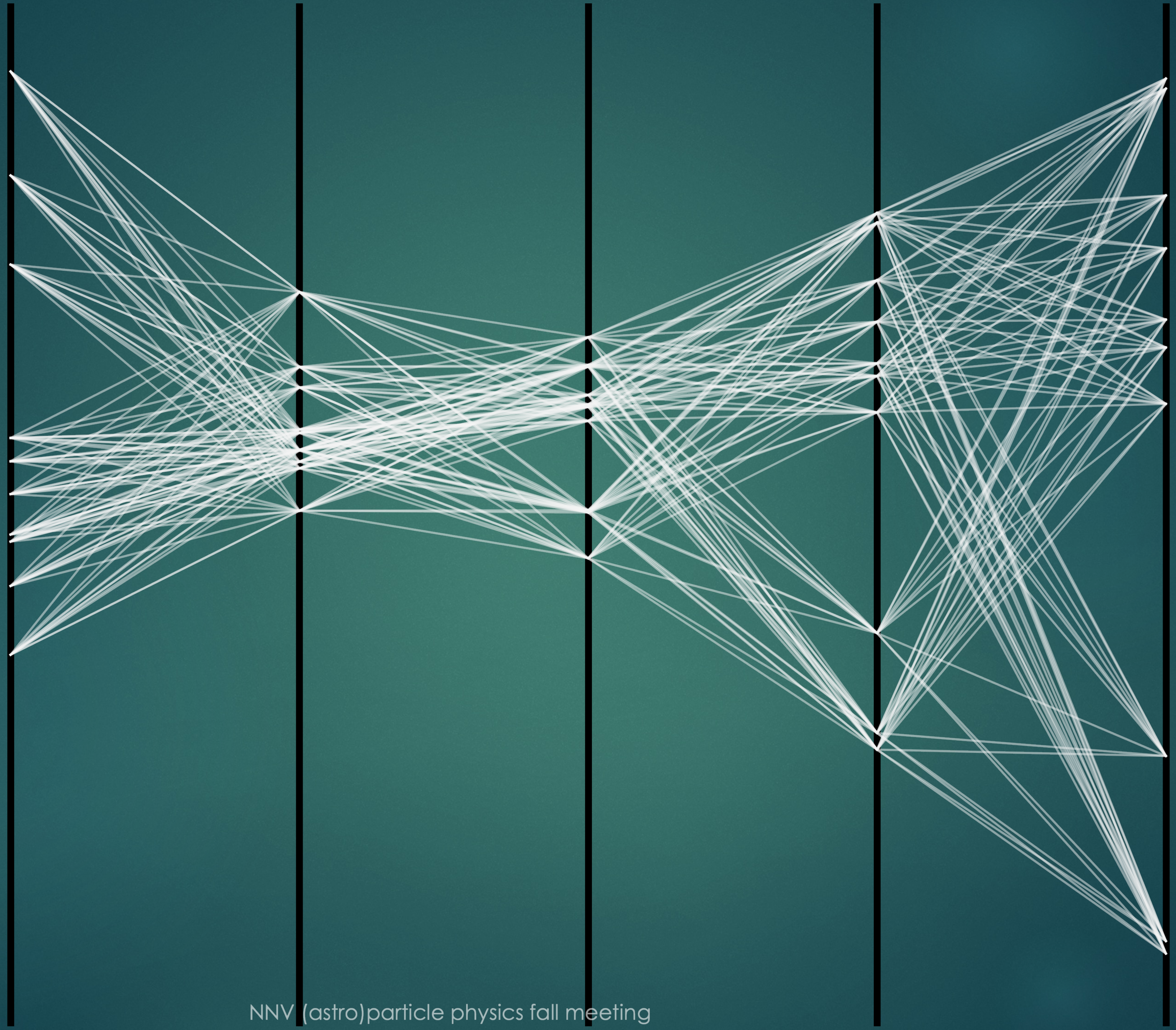
Achieved through circuit optimization and 1-bit phase estimation

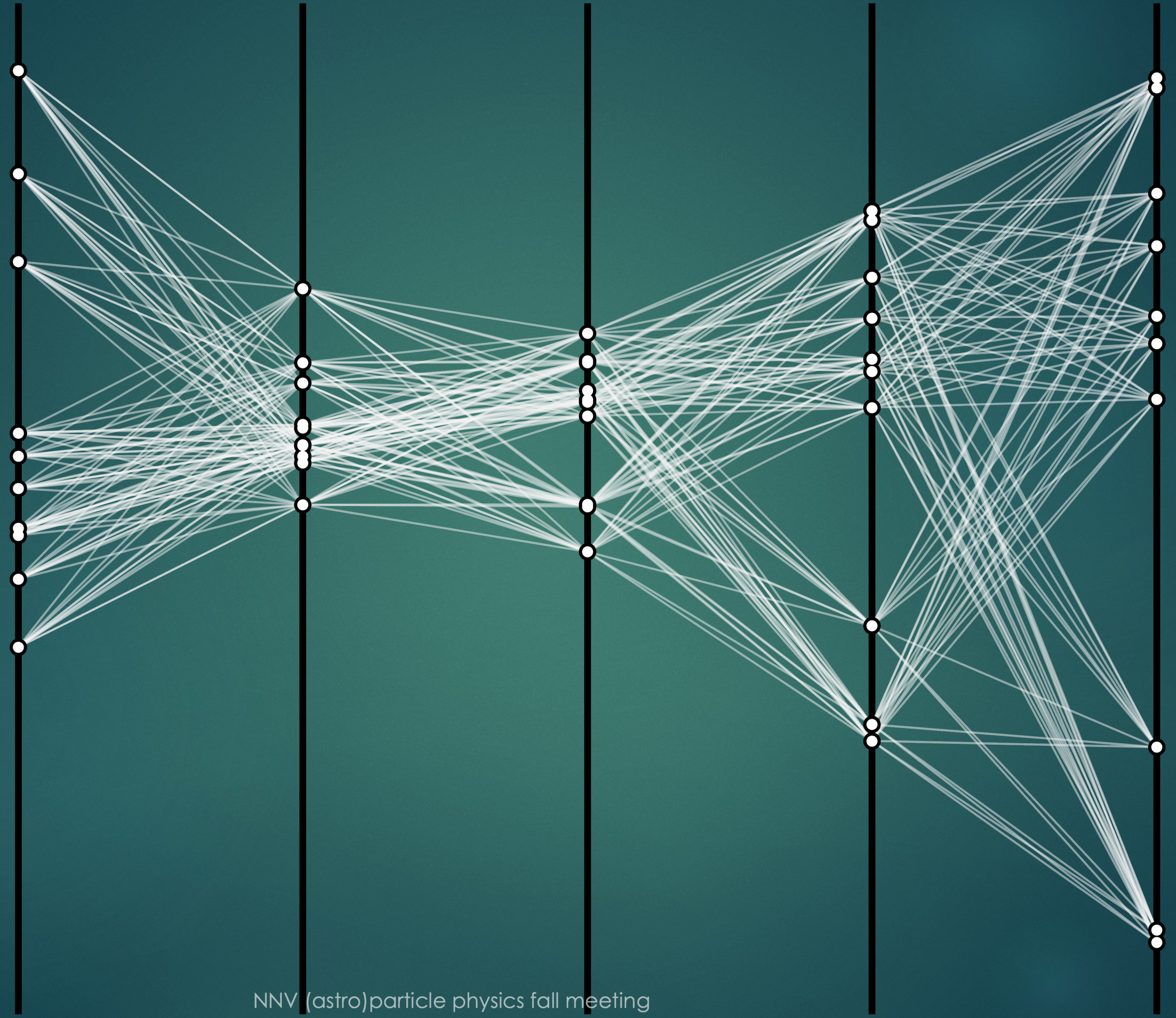
* Represents the 1-Bit Phase estimation results

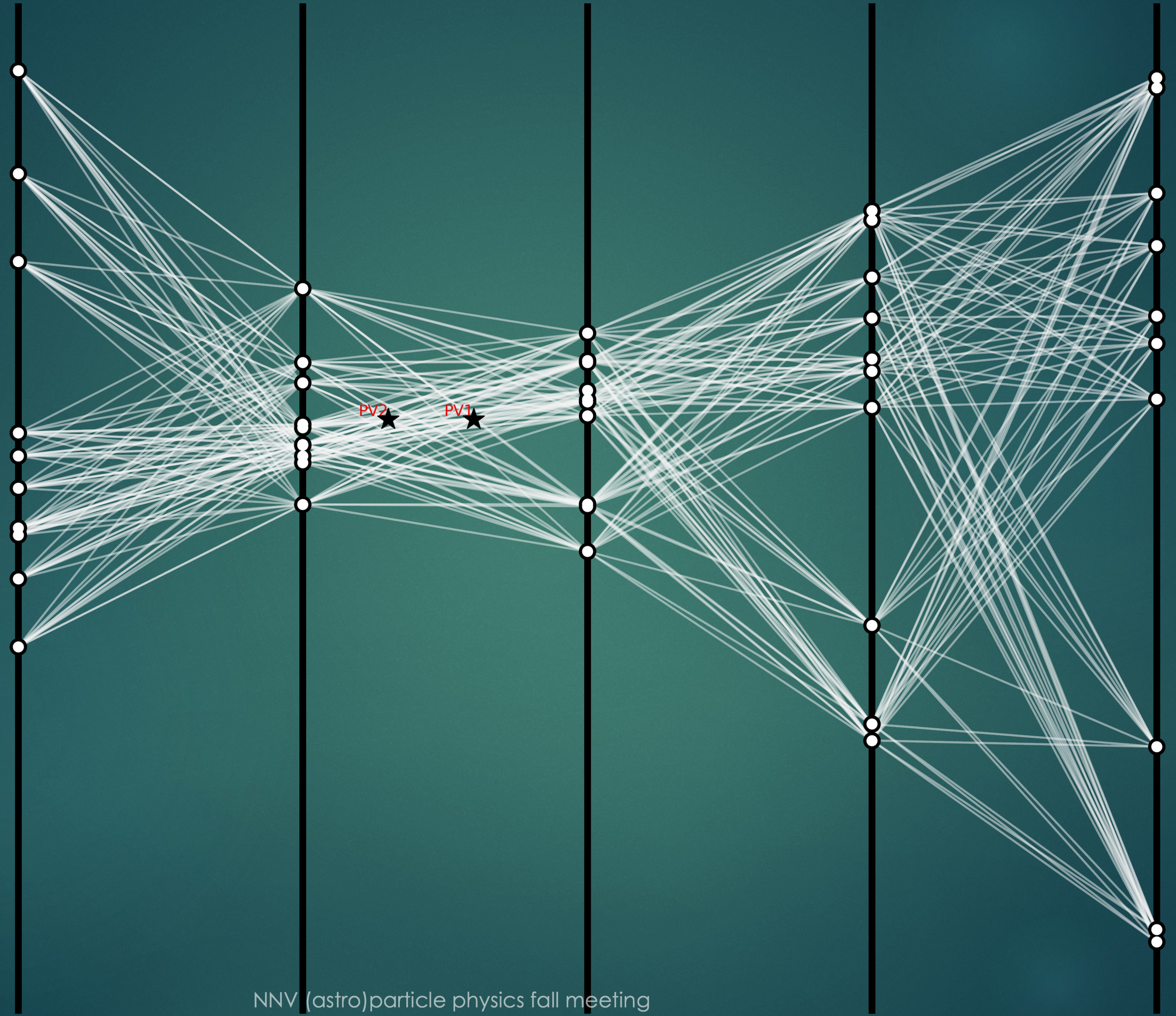
Largest Simulated Event

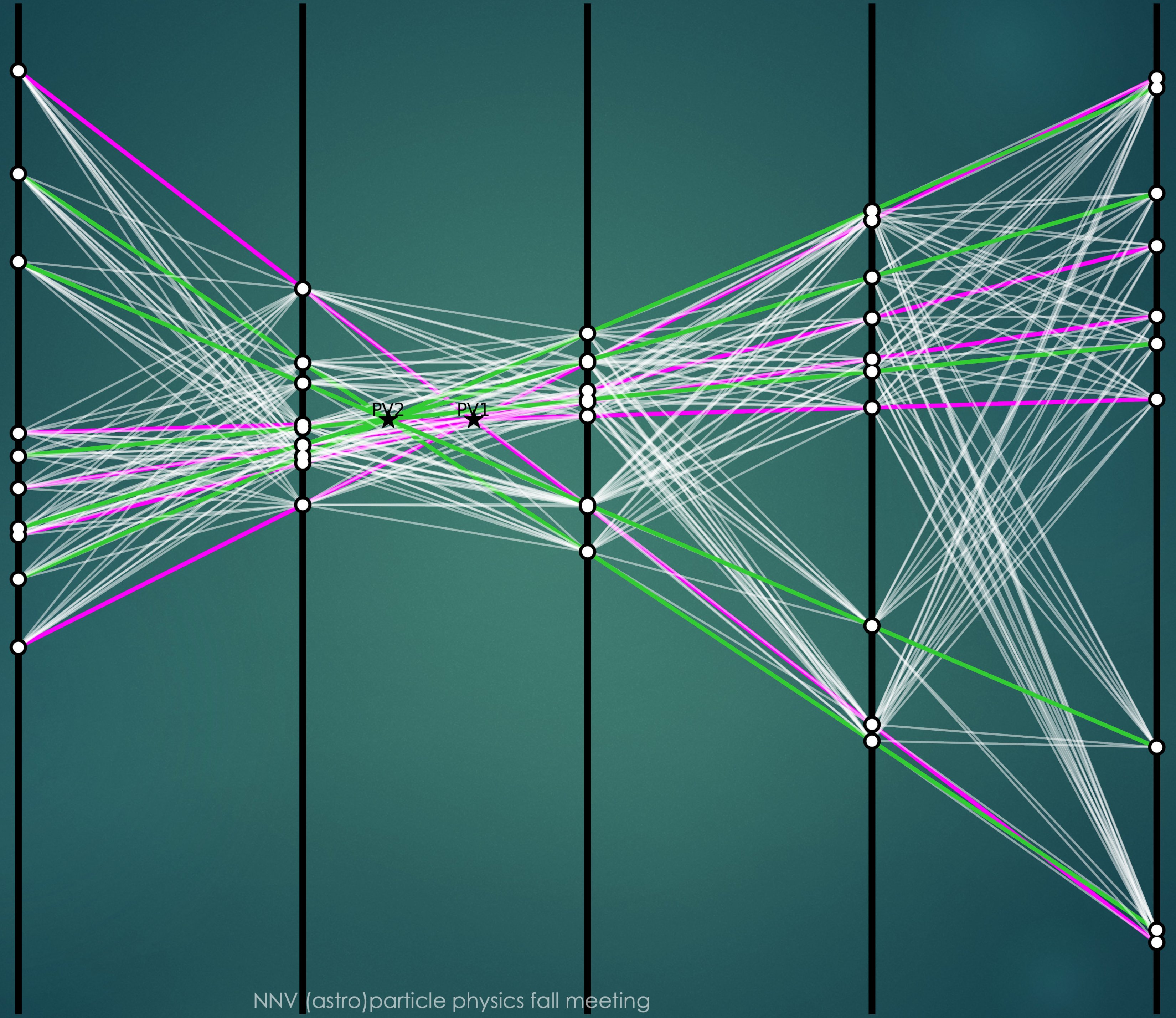
Largest Simulated Event Matrix





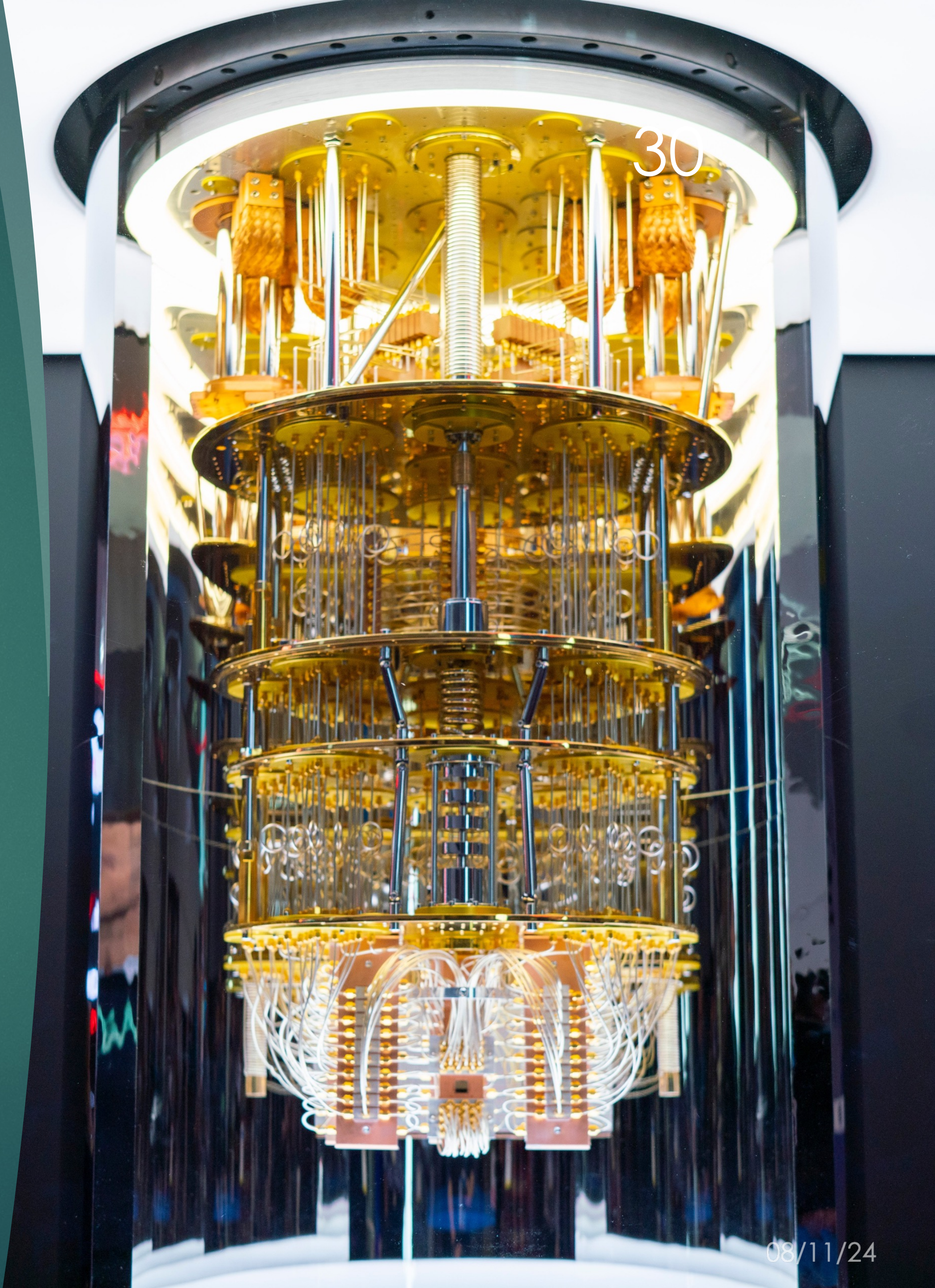



PV2
PV1


PV2
PV1

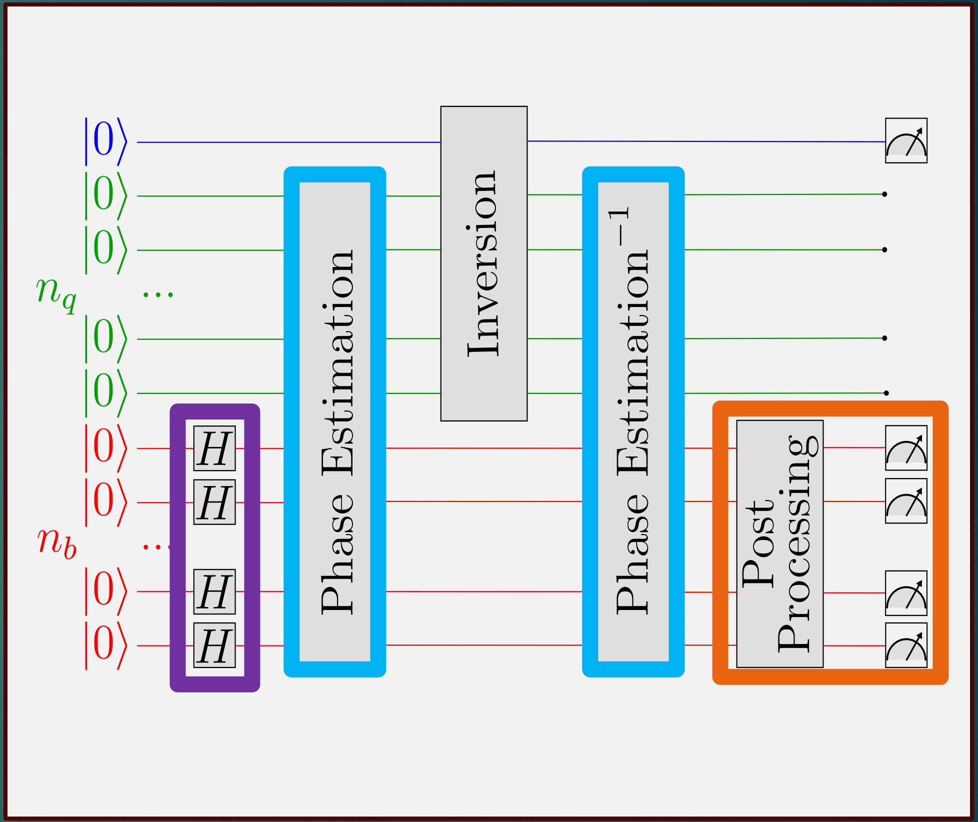
Benefits of 1-Bit HHL

- Upto a $\times 10{,}000$ reduction in circuit depth
- Pre-processing inside quantum circuit, logarithmic reduction in samples needed for reconstruction
- Reduction in qubits needed (where N is matrix dimension):
  - $2\log_2 N + 2$ qubits originally
  - $\log_2 N + 3$ qubits for 1-Bit HHL



Conclusions

- Matrix inversion track solvers have a good performance classically, HHL quantum version also shows good results
- Adopting 1-bit phase estimation HHL significantly improves feasibility in qubits, circuit depth and read-out



Future Work

- Take advantage of sparsity structures
- Encoding geometry information into the Hamiltonian
- Benchmarking the Primary Vertex finding on data with PV information

Backup Slides

Further Solving our Read-Out Problem

Time Evolution

$$H|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}|\psi(t)\rangle$$

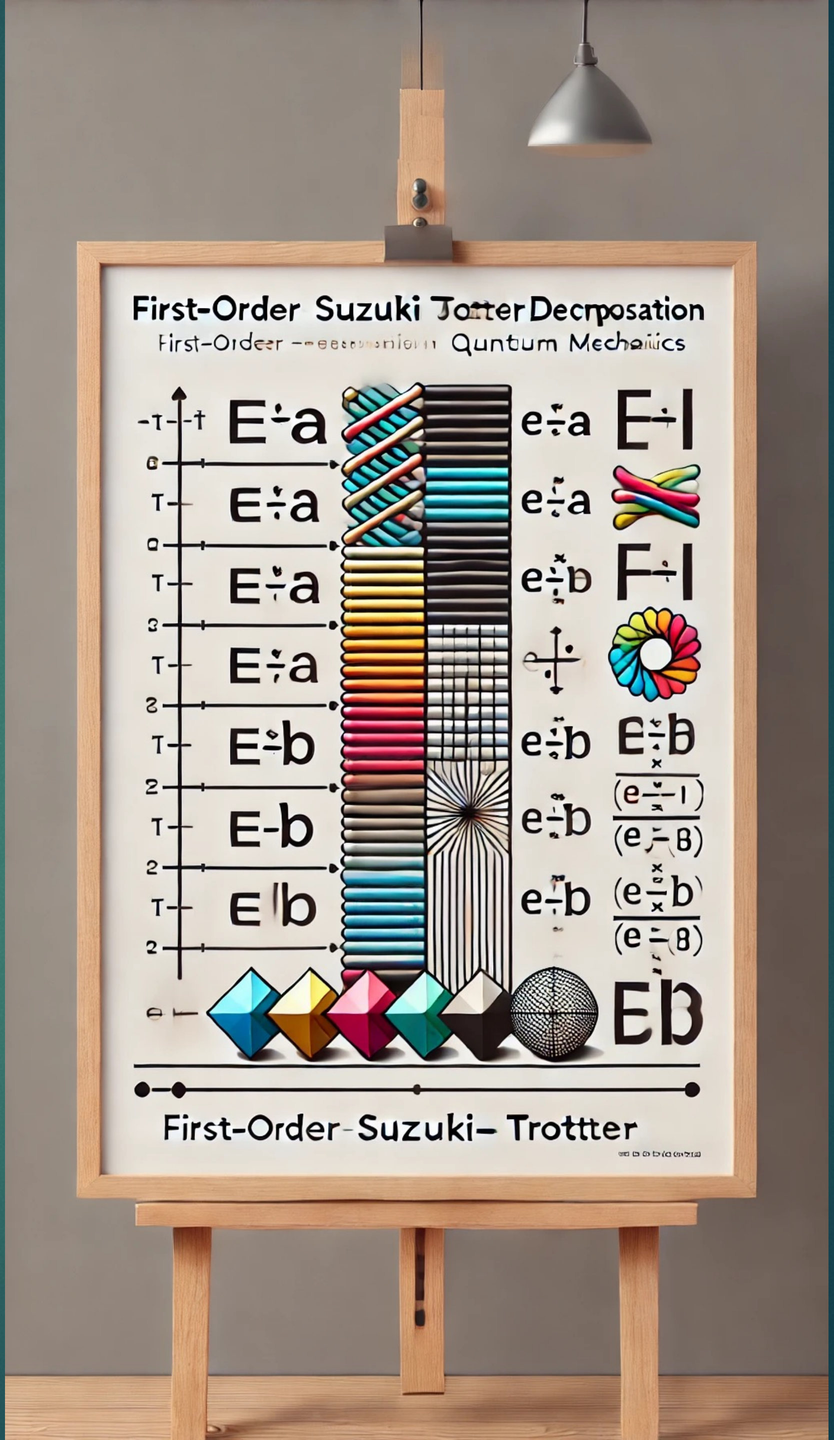
Time Evolution Operator

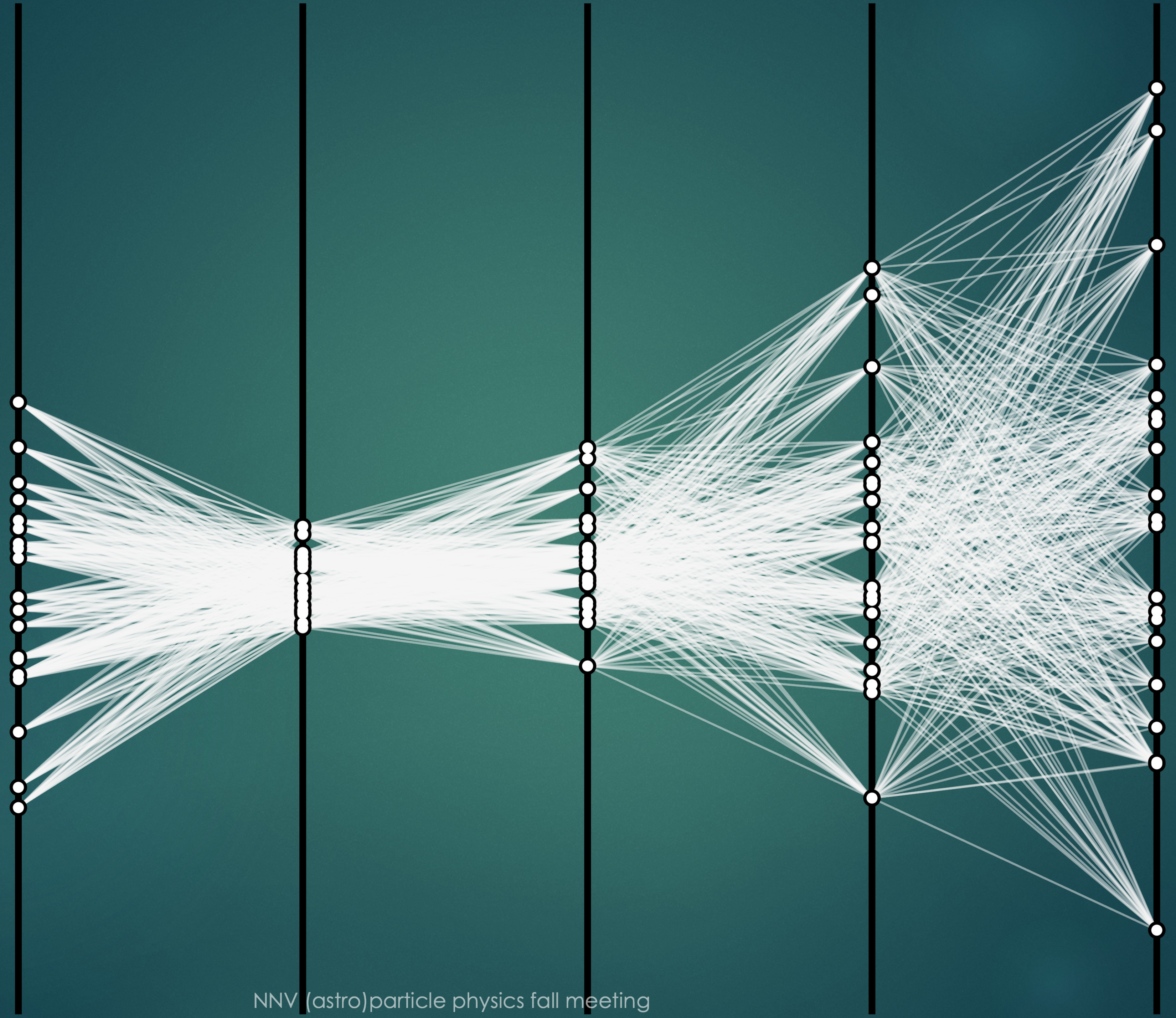
$$|\psi(t)\rangle = U(t)|\psi(0)\rangle = e^{-iHt/\hbar}|\psi(0)\rangle$$

Via Trotterization

Suzuki Trotter Decomposition

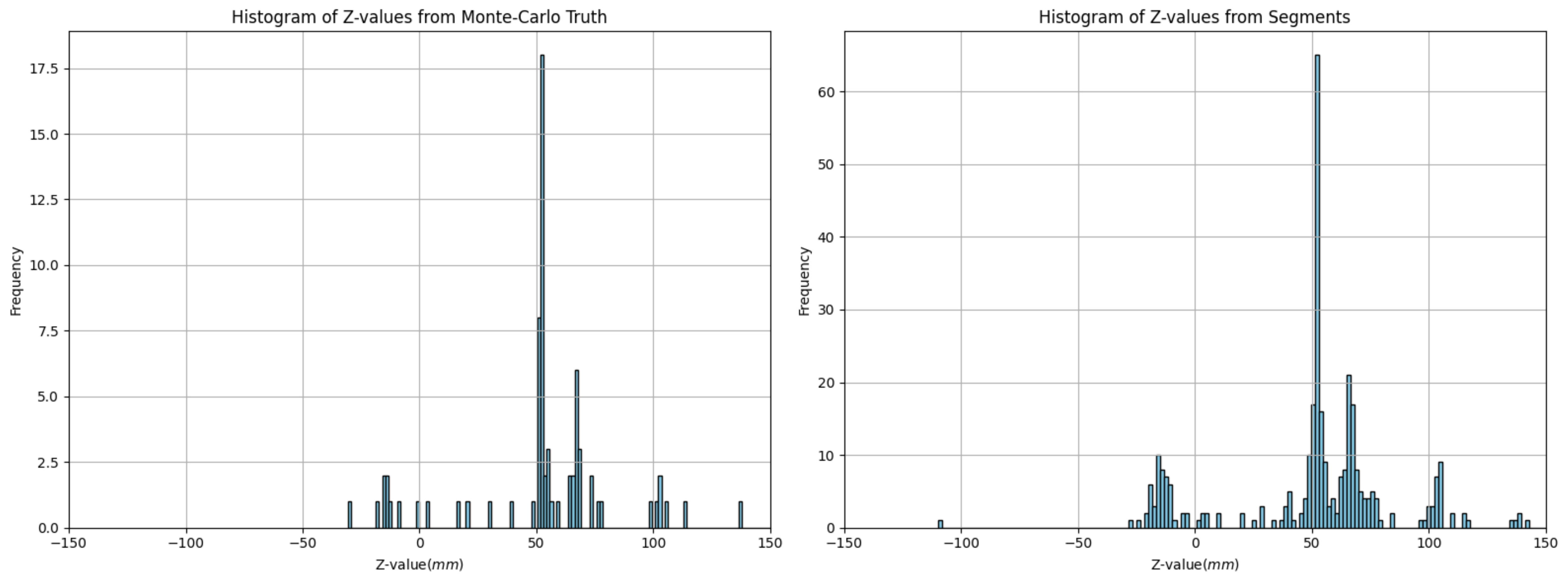
- $H = H_A + H_B$ *and* $U(t) = e^{-iHt/\hbar}$
- If $[H_A, H_B] \neq 0$ computing $e^{-i(H_A+H_B)t/\hbar}$ is very expensive
- So, we use the Suzuki-Trotter decomposition to do time evolution
- $e^{-i(H_A+H_B)t/\hbar} \approx (e^{-iH_A\Delta t}e^{-iH_B\Delta t})^N$
- where $\Delta t = {}^{t}/_{N}$ and as $N \to \infty$ the approximation becomes exact.

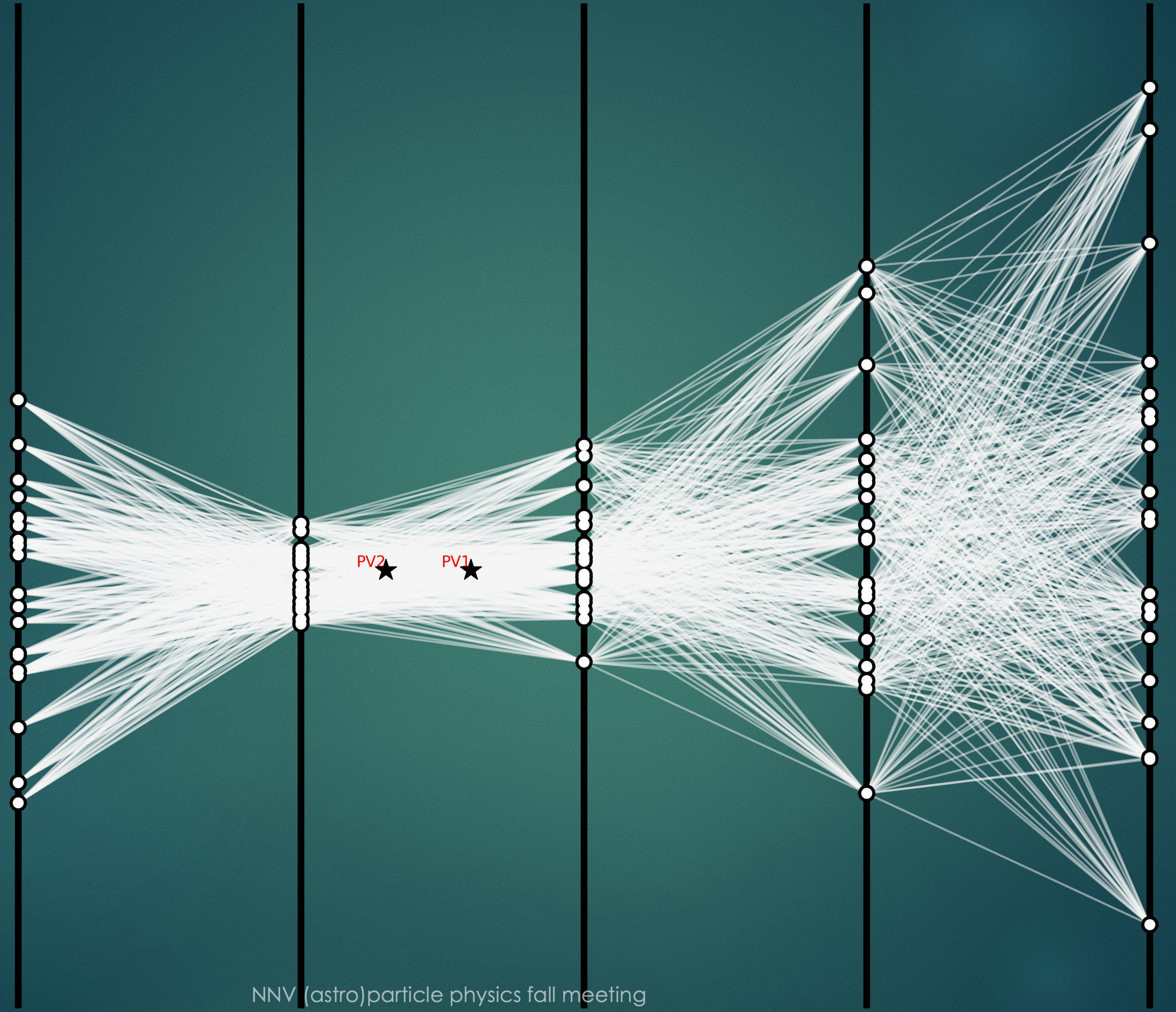




Solving the Readout Problem:

Reconstruct the Primary Vertices and re-find all tracks




PV2
PV1

The Hough Transform