Tracking studies for quantum computer at Tokyo

1) Track finding with quantum annealing
2) Attempt for track finding with gate-based system

Not really generic talk; more like working status…

Track finding with quantum annealing

Track finding with quantum annealing

Started by following Lucy's talk at CPAD conference **_Thanks!_**

- tried to mimic Lucy's approach…
- skipped a few steps (just for simplification or laziness..)

What we have done so far:

- simulate hits using FCC-hh inner detector
- reconstruct triplets from inner detector hits
- define bias and interaction weights for each triplet and triplet pairs
  - *weights not optimized at all so far…*

→ details in backup

- running simulated annealing to solve QUBO
- get track candidates from selected triplets

Input to QUBO = pre-selected triplets (details in backup)

Output from QUBO = triplets as those associated with tracks

M. Saito has started looking into Lucy's code

- Exploring track finding with doublet-based QUBO approach

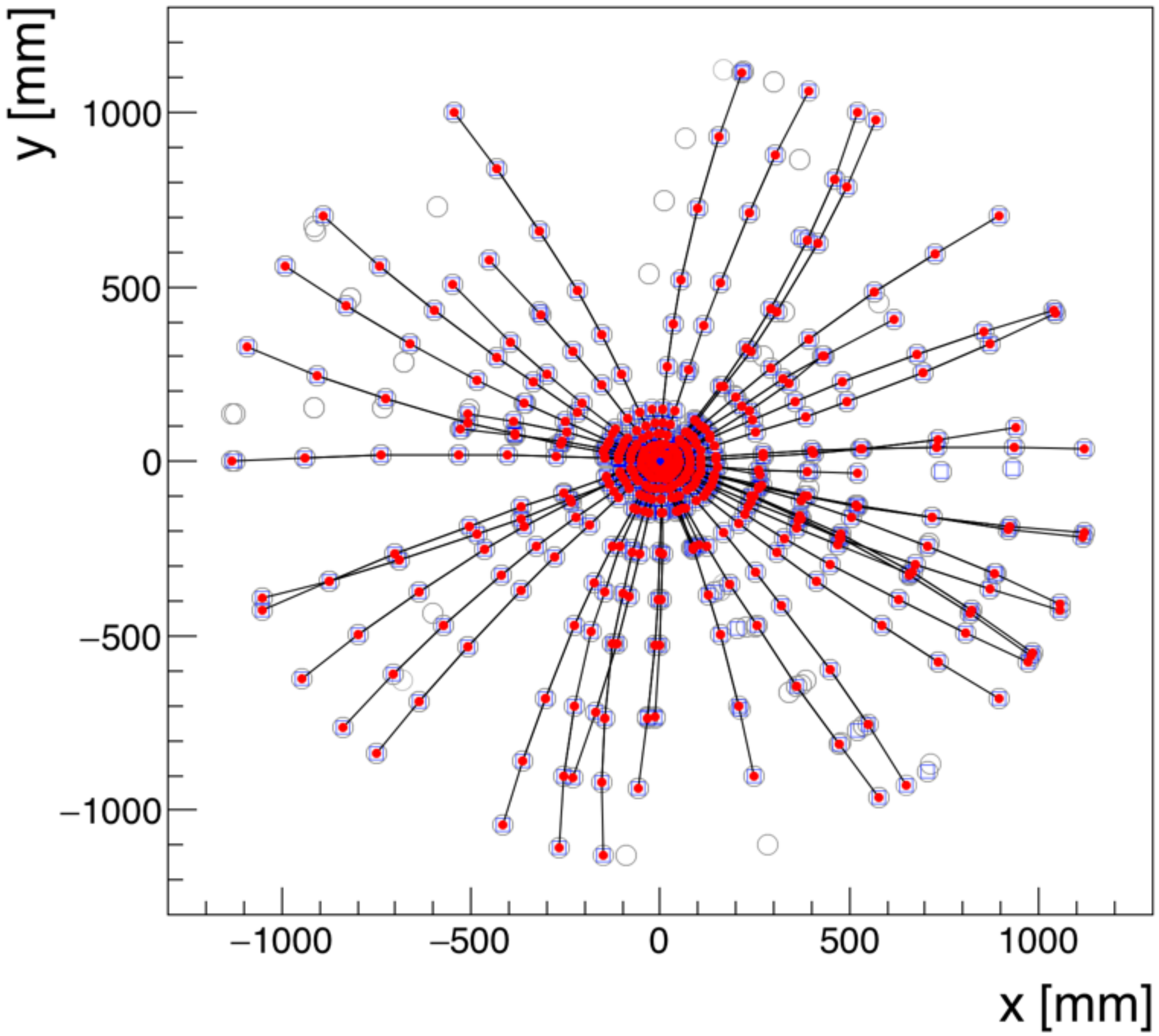
Results

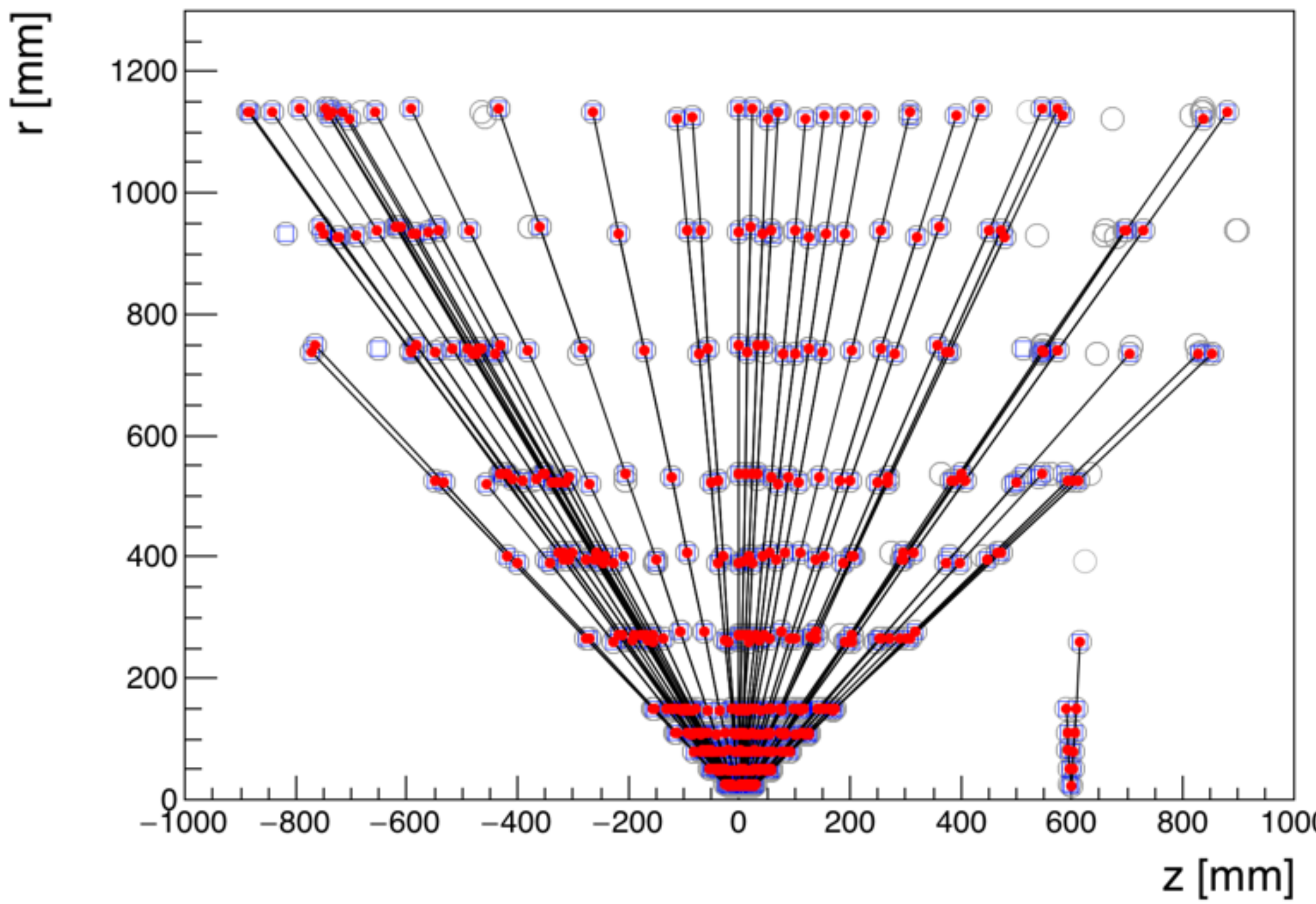
50 muons

hits in all 2416 triplets

hits in 478 triplets selected by θ'<0.05 cut (= input to QUBO)

hits in 378 triplets found by annealing (= output from QUBO)





Note: lines are drawn by checking positions of shared hits in selected triplets

# of found *tracks* (total):

11-hits = 18	7-hits = 3 (42)
10-hits = 5 (23)	6-hits = 5 (47)
9-hits = 7 (30)	5-hits = 6 (53)
8-hits = 9 (39)	4-hits = 10 (63)

Quick test on D-Wave (just one event, though)

➔ Same set of triplets found in D-Wave and simulated annealing

Modification to QUBO Hamiltonian

Q) Is it possible to select only triplets associated with *specific* tracks?

Original hamiltonian:

$$\mathcal{O}(a;b;T) = a\sum_i^N T_i + \sum_i^N \sum_{j>i}^N b_{ij}T_iT_j \quad T \in \{0,1\}$$

Add additional constraint to select a specific number of triplets

$$a\sum_i^N T_i \rightarrow a\sum_i^N T_i + a'\left(\sum_i T_i - N_f\right)^2$$

N = # of selected triplets (= qubits)

N_f = Required # of triplets to be found

Modified hamiltonian:

$$\mathcal{O}(a;b;T) \rightarrow a'N_F^2 + (a + a'(1-2N_f))\sum_i^N T_i + \sum_i^N \sum_{j>i}^N (2a' + b_{ij})T_iT_j$$

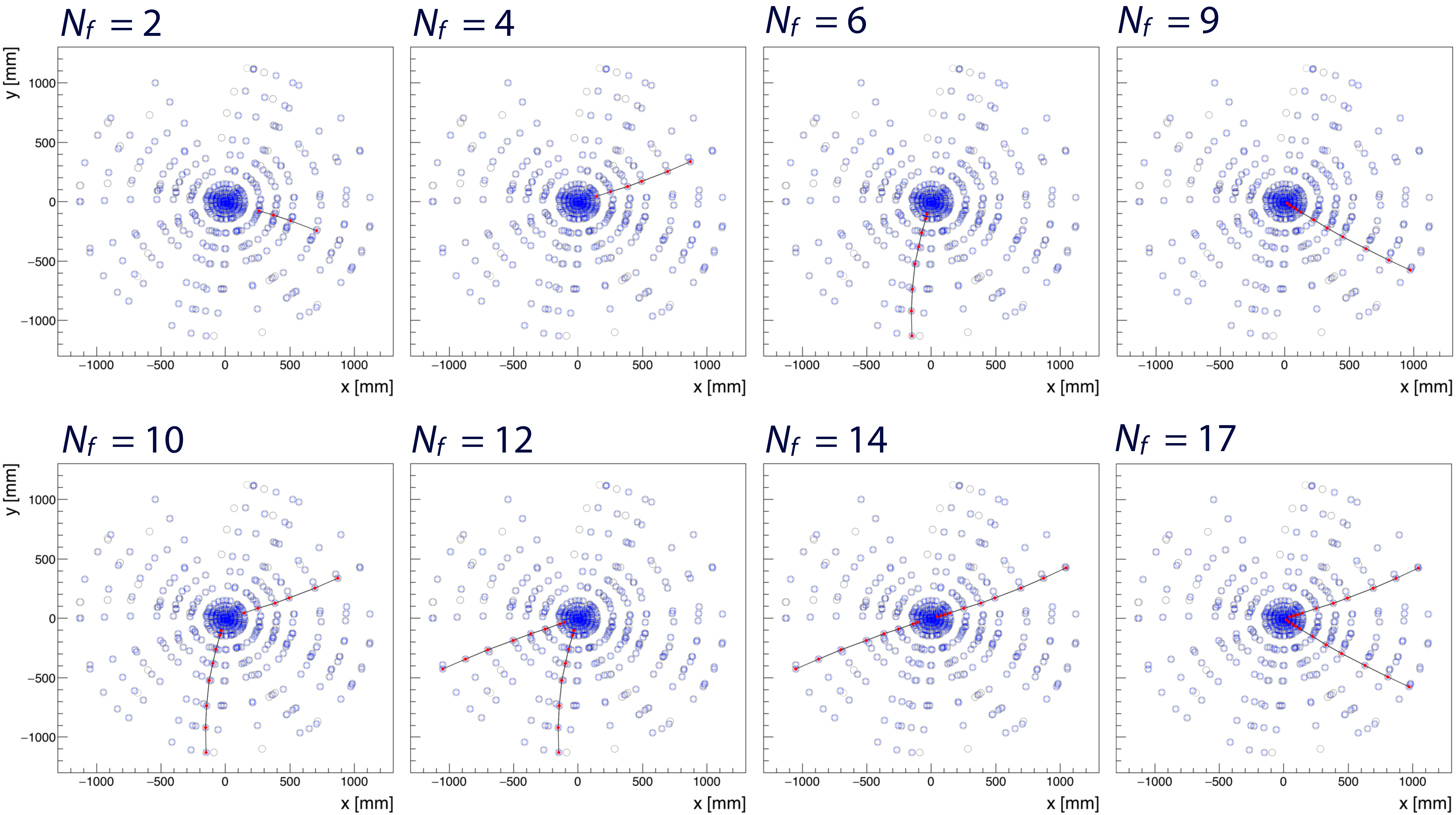
$$\sim (a + a'(1-2N_f))\sum_i^N T_i + \sum_i^N \sum_{j>i}^N (2a' + b_{ij})T_iT_j$$

Set to a = 1 and a'=1

Results don't seem to depend on the default values…

Results

By changing N_f…

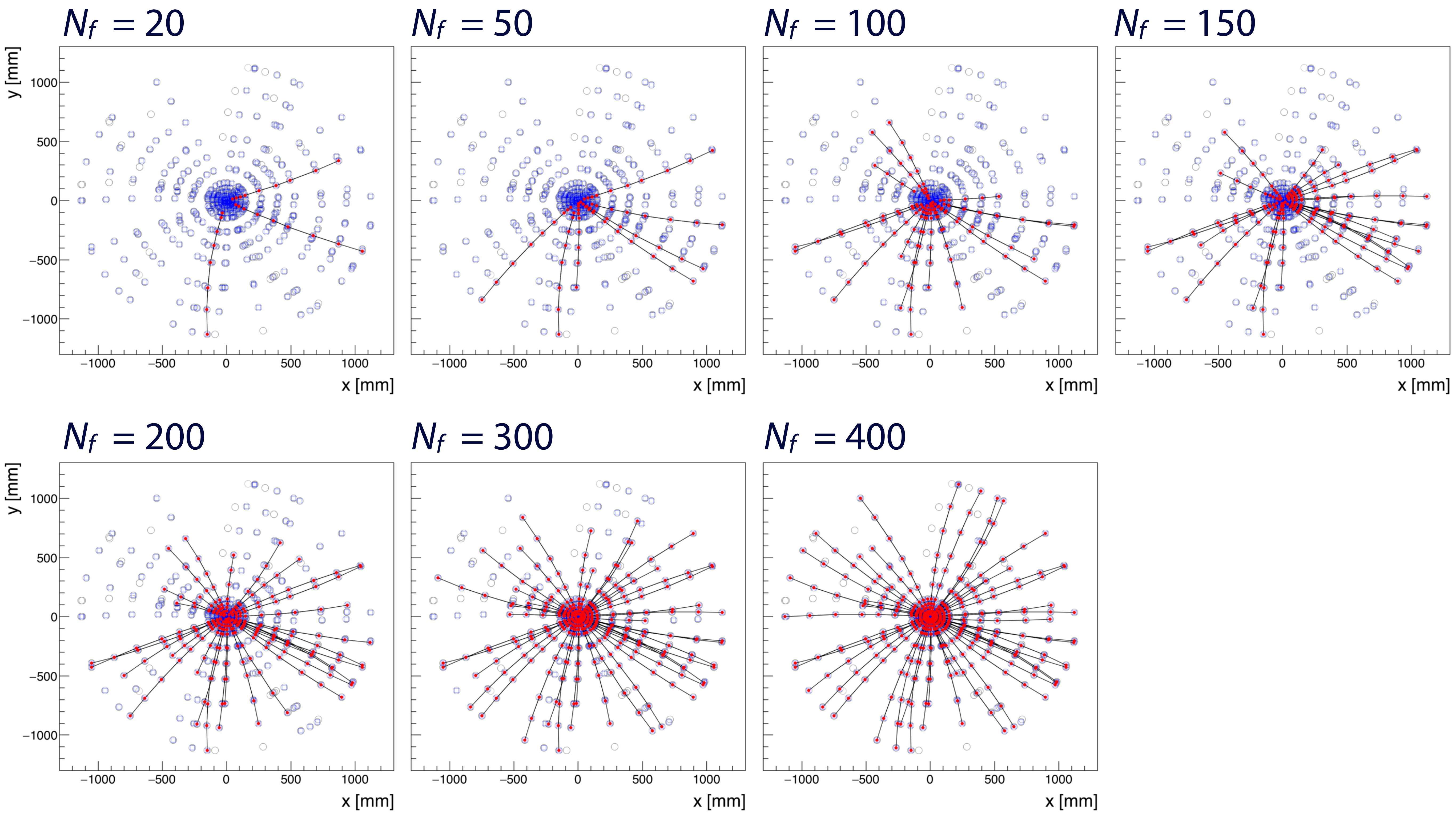


Seems like it's working as hoped…

Maybe some application for selecting specific tracks by annealing?

Results

By changing N_f…



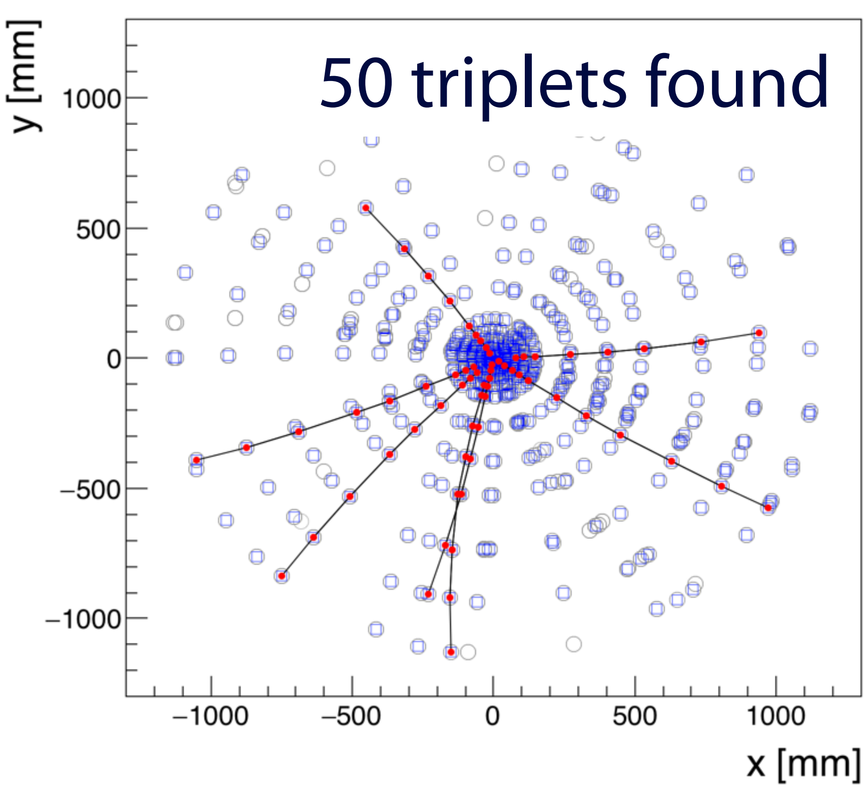
Setting N_f to be large appears to restore the original QUBO solution it would be interesting to find out why there is an initial preference for shorter tracks, and why with an increase in N_f, it's not the shorter tracks that are first extended. Naively, I'd think that a pattern recognition that's restricted with Nf would be more useful if it first completes tracks.

➔ The N_f constraint is currently applied equally to all triplets, i.e, no preference for those triplets that are lined up to make a single track candidate. Therefore, triplet combinations with the smallest $-s_{ij}$ are chosen no matter whether they belong to same tracks or not (I think).

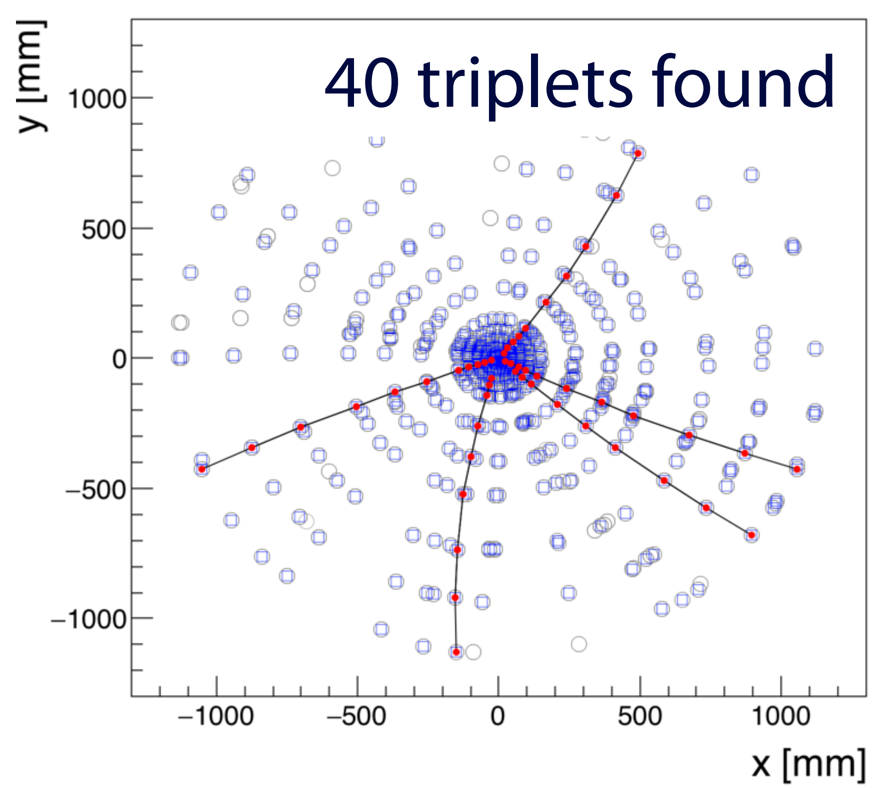
Also, related to the size of "unrelated weight" (set to 0 by default)

$N_f = 50$ for all

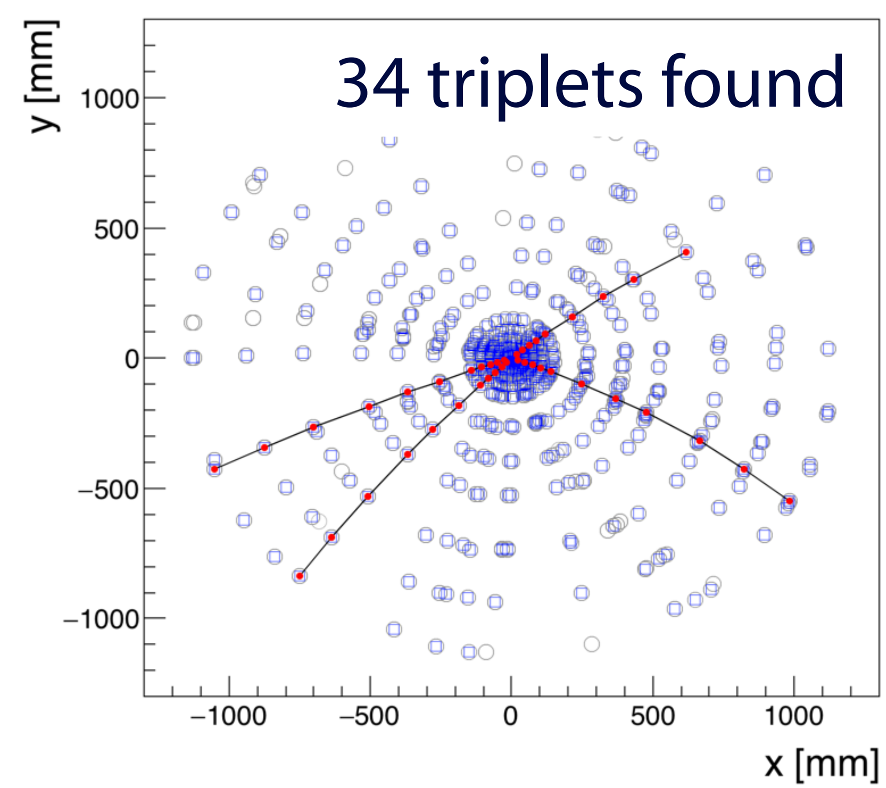
$w_{unrelated}$=0



$w_{unrelated}$=0.5



$w_{unrelated}$=1.0



Attempt for track finding with gate-based system

Attempt for track finding with gate-based system

Modify QUBO hamiltonian used above to be Ising hamiltonian
Solve eigenvalue problem with Ising hamiltonian

Original QUBO hamiltonian:

$$\mathcal{O}(a;b;T) = a\sum_{i}^{N} T_i + \sum_{i}^{N}\sum_{j>i}^{N} b_{ij}T_iT_j \quad T \in \{0,1\}$$

➔ Translated to Ising hamiltonian by converting T_i to s_i with $T_i = \frac{1-s_i}{2}$

$$H = \sum_{i}^{N} h_i s_i + \sum_{i}^{N}\sum_{j>i}^{N} J_{ij}s_is_j \quad s_i \in (-1,1)$$

$$J_{ij} = Q_{ij} + Q_{ji} \quad h_i = -\sum_{k}^{N}(Q_{ik} + Q_{ki}) \quad Q_{ij} = \begin{cases} a & (i=j) \\ b_{ij} & (j>i) \\ 0 & (j<i) \end{cases}$$

➔ Obtain the ground state by solving eigenvalue equation with Ising hamiltonian

Both a (=0.01) and b_{ij} are same as used for the annealing

QC code

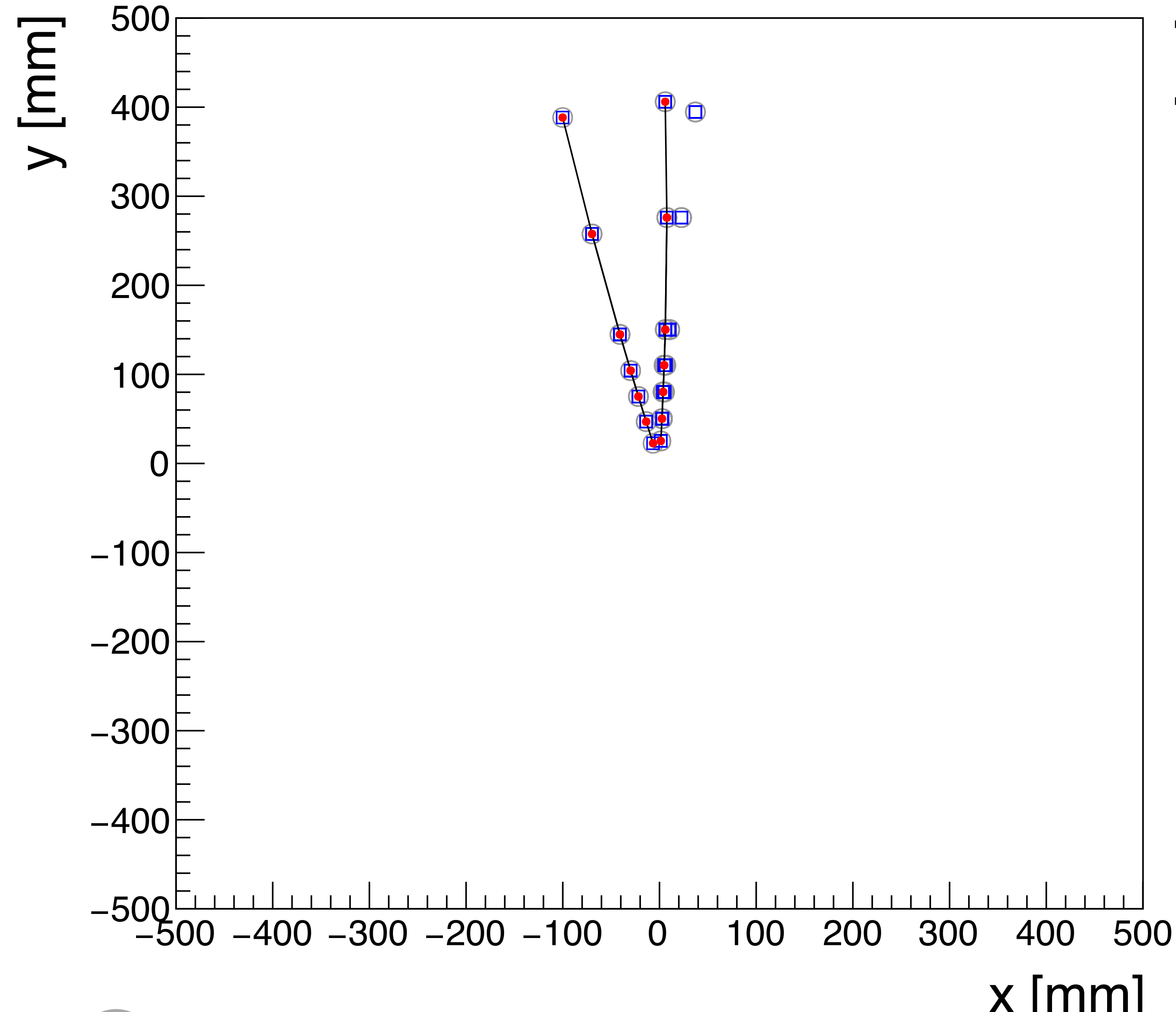
- IBM Qiskit framework

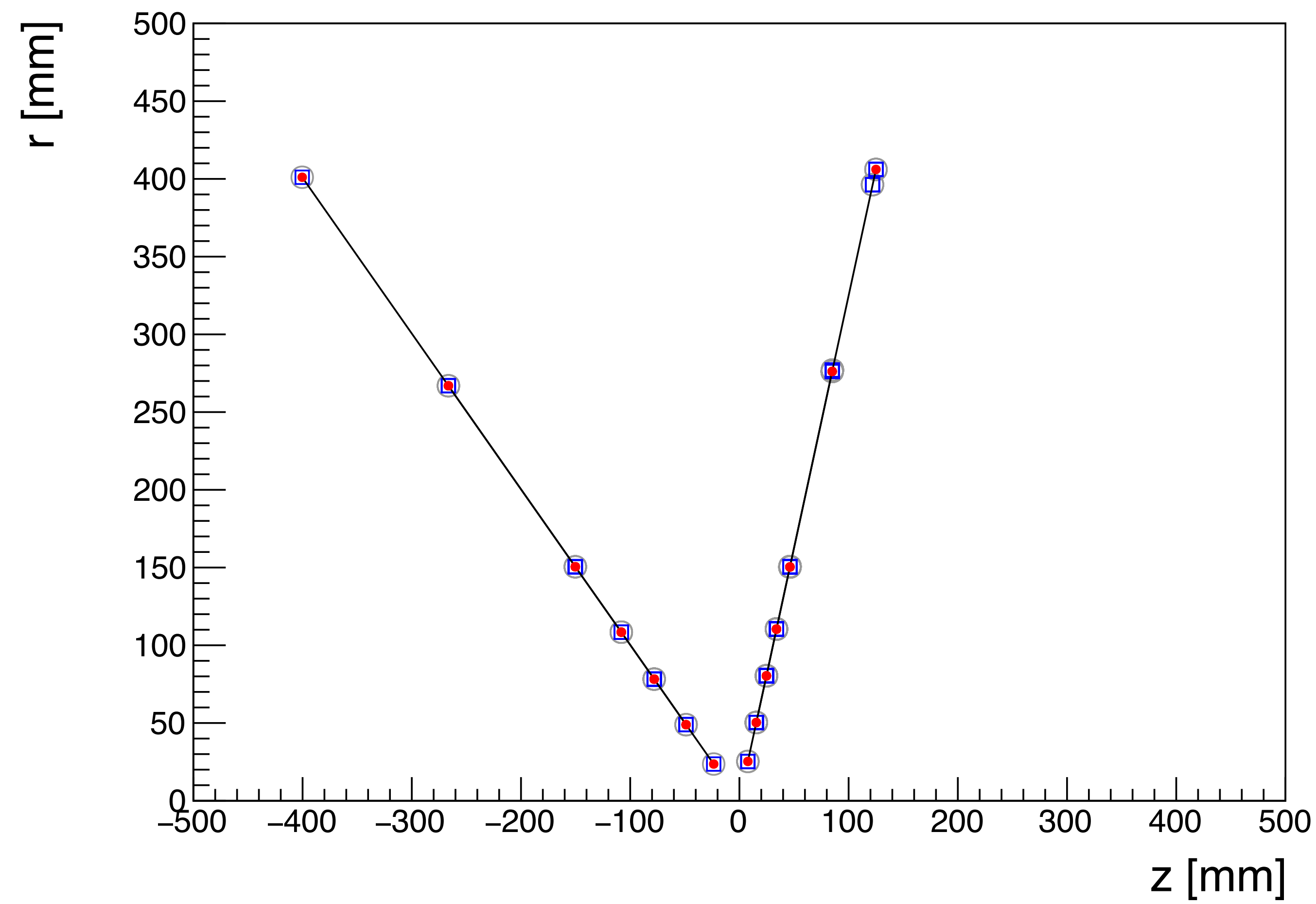
Results

5 muons

7 layers only

Use only triplets with $|\eta|<1$, $|\phi-0.5\pi|<0.1\pi$





Processing time exponentially grows with the number of qubits (=triplets)

- hits in all 27 triplets
- hits in 27 triplets selected by $\theta'<0.05$ cut
- hits in 10 triplets found by **ExactEigensolver**

#triplets	time
11	<0.1 sec
27	~30-45 min

I presume the exponential growth in time is in the simulation only? (Unless you're doing full QPE.)

➔ Hope so… No success on hardware yet

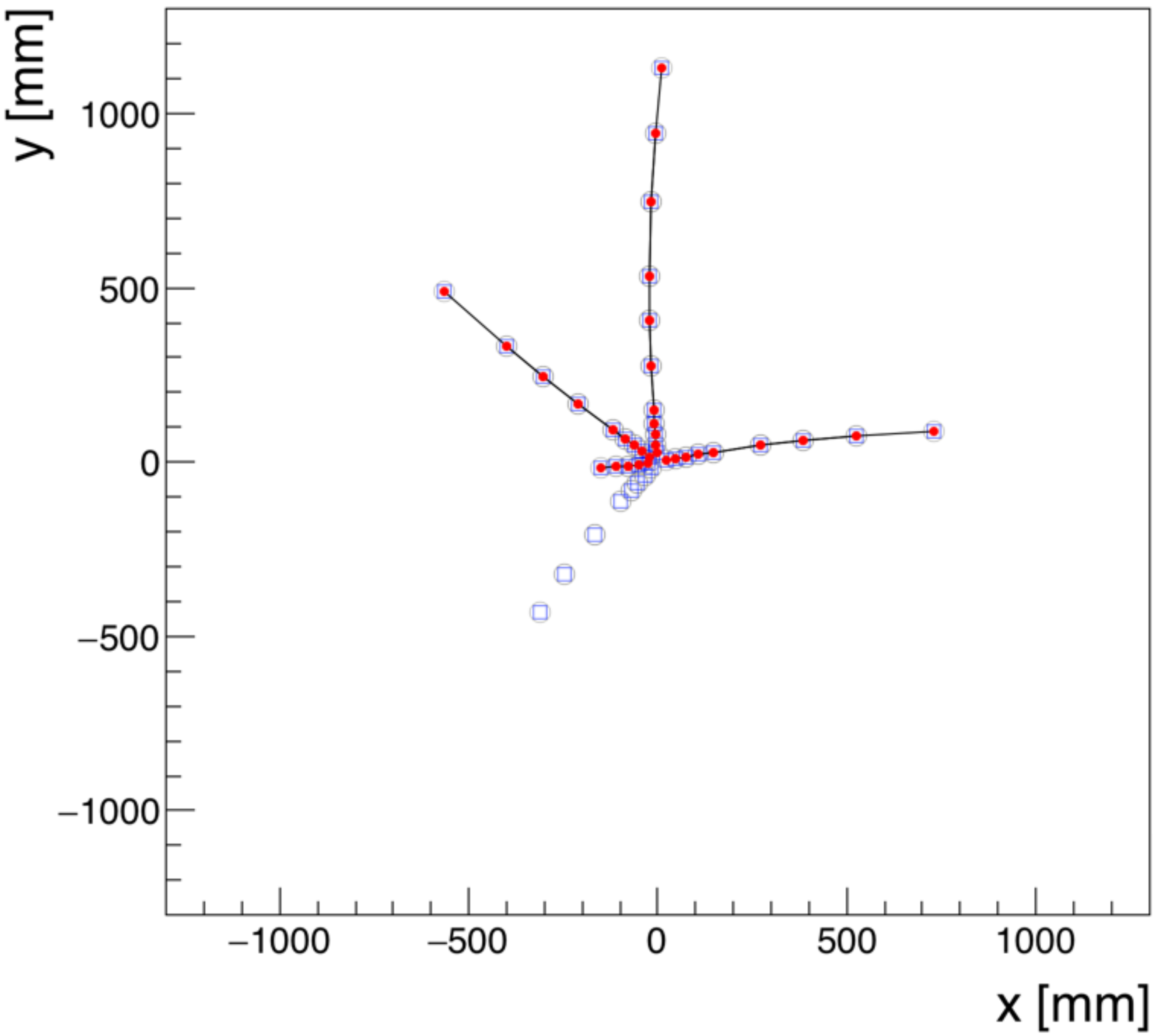
Results

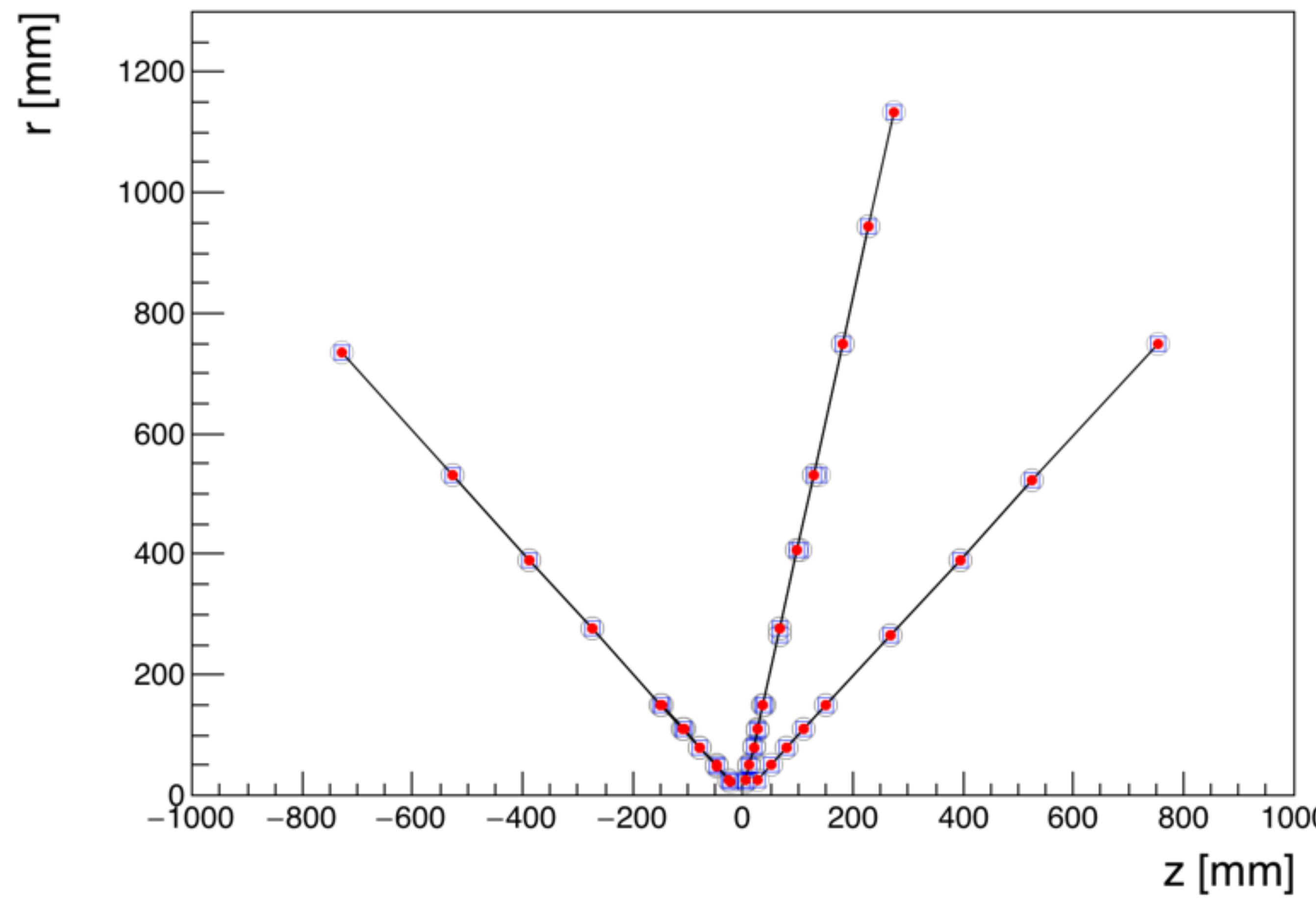
5 muons

11 layers

Use only triplets within a slice ($|\eta|<1$, $\Delta\phi<0.2\pi$) at a time

Repeat measurement by sliding over ϕ





- hits in all 33 triplets
- hits in 33 triplets selected by $\theta'<0.05$ cut
- hits in 26 triplets found by **VQE statevector_simulator**

$\Delta\phi$	#triplets (found)	time [sec]
$[-\pi, -0.8\pi]$	3(3)	230
$[-0.8\pi, -0.6\pi]$	7(0)	1020
$[0, 0.2\pi]$	7(7)	890
$[0.4\pi, 0.6\pi]$	9(9)	1530
$[0.6\pi, 0.8\pi]$	7(7)	920

yes, you can run VQE on real hardware, but you'll be limited by the classical optimizer step (the QisKit minimizers are mostly the same as those in SciPy and only COBYLA is more or less usable when noise is present, but only if you have a good initial).

➔ Not sure what to use… Used SPSA so far; Quick test with COBYLA looks similar

As for Hamiltonian design: the trick with VQE is to generate Hamiltonians that are mostly local. That width and depth of circuits are reduced and some can even be run independently. Ie. in your slide 13, a whole lot of Jij terms should be zero. (Since VQE is iterative, it may make sense to do so artificially for the first bunch of iterations, just to establish a good initial.)

There are other issues with scaling of VQE. In particular, progress in Hilbert space need not translate to progress in classical space. This problem becomes exponentially worse with increasing # qubits and the only real mitigation is by better ansatz design (and noise reduction).

➔ Slicing in (η, ϕ) can make the hamiltonian size small and run independently (need treatment for particles traversing the boundaries, though)

How important is the initial state for VQE?

VQE setup in Qiskit aqua:

- depth = 5 → #parameters = #qubits × (depth + 1) ~ O(100) in present case
- variational form = Ry (testing RyRz, …) → Optimizing ansatz for this problem?
- initial state = zero
- optimizer = SPSA
- linear entanglement
- #shots = 100

To DO

Track finding with quantum annealing

- Application of specific track reconstruction with annealing?
- Quantitative evaluation of processing time relative to standard reconstruction using D-Wave system
  - maybe better to improve algorithm first?
- Using annealing for shorter objects (before triplet selection)?
  - at doublet selection? or at even hit level?? → M. Saito
- Can we exploit more quantum effects in QUBO tracking?
  - superposition? entanglement?
  - any conclusive example of quantum annealing superior to classical annealing?
- Optimization of QUBO tracking algorithm?
  - any improvement in decomposing QUBO problem into sub-QUBOs?
  - check performance with different annealing times, # of annealing runs, etc.
  - different sampling scheme??

To DO

Track finding with gate-based system

- Exercise VQE + Ising hamiltonian model on real hardware

  IBMQ QASM HPC simulator:

  - Succeeded in one slice (3 qubits), then failed in next slice with message:
    qiskit.providers.exceptions.JobError: 'Invalid job state. The job should be DONE but it is JobStatus.ERROR'
  - Slightly faster response than hardware, but still slow…

  IBMQ 16-bit system (ibmq_16_melbourne):

  - No success… essentially no response; one occasion with error message:
    Got a 502 code response to /api/Jobs/5c55a8dd5a747200565b113a/status: 502 Bad Gateway: Registered endpoint failed to handle the request.

- More thorough studies on algorithm, ansatz design, etc. needed
  - Currently requires lots of CPU time if #qubits > ~20…
  - Consider alternative hamiltonian model?

Interesting exercise for VQE etc., but less likely for general-purpose tracking

Need more preferred hardware access (e.g, IBM Q Network) for realistic study?

Other Directions?

- Simulation, e.g,
  - Parton-shower simulation with interfering quantum trees (arXiv:1901.08148)
  - Quantum simulation of Yang-Mills theory and hadronization (arXiv:1810.09213)
  - Simulation of physical events by sampling quantum states?
- Machine learning, e.g,
  - Quantum SVM (→ R. Sawada)

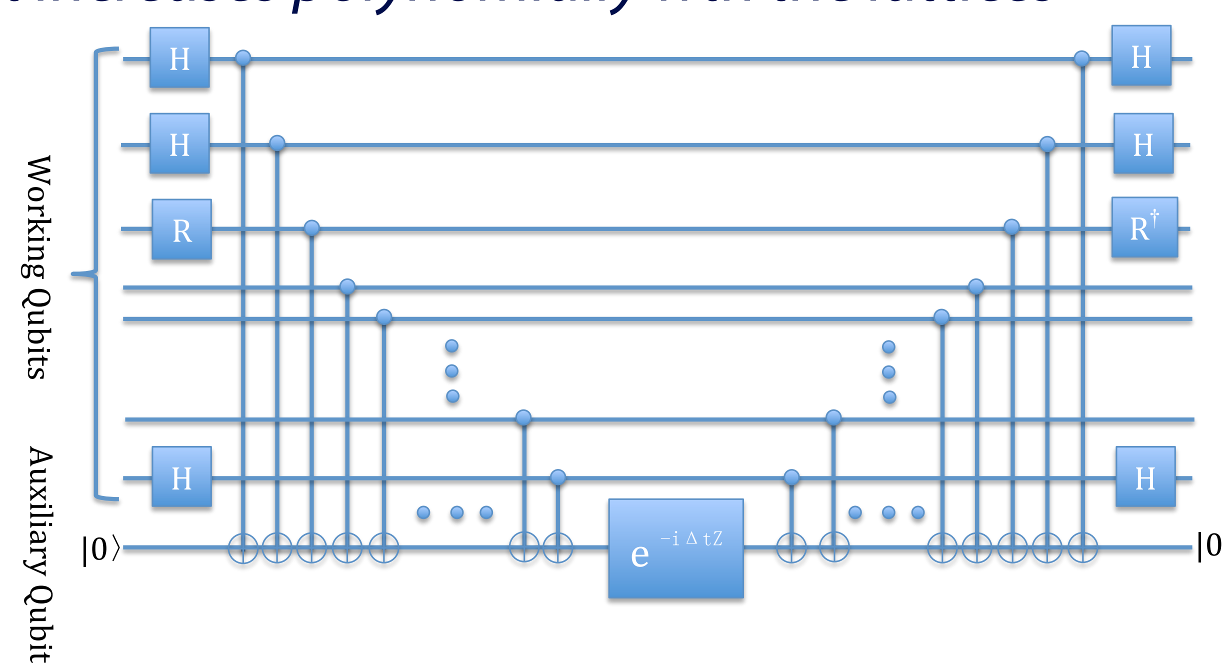
Quantum simulation of *SU*(*N*) Yang-Mills gauge theory

- *Quantizing the theory with annihilation and creation operators*
- *Mapping particles to qubits (based on Lattice approach) with Jordan-Wigner transformation of annihilation/creation operators to Pauli matrices on qubits*
- *"Exponential speed-up with the # of qubits that increases polynomially wth the lattices"*

Claimed to work in both perturbative and non-perturbative regimes

➔ Proposed study for QCD Hadronization

But no test on simulator yet…?



Backup

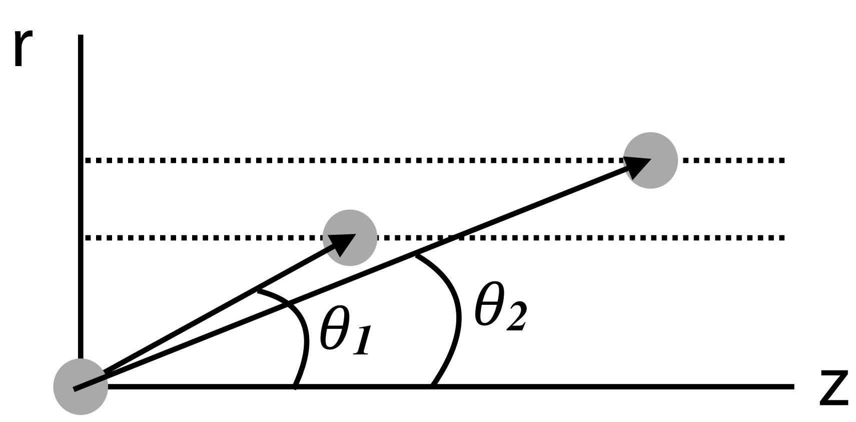
Setup

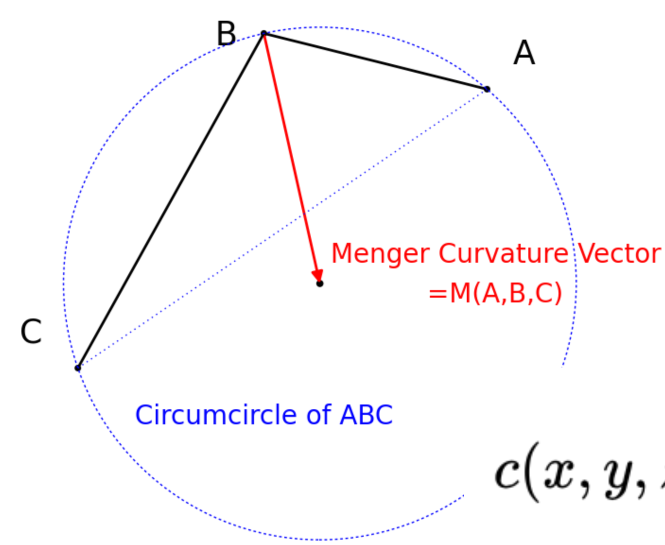
Simulation

- single muon ($1<p_T<10$ GeV, $|\eta|<1$) events → overlaid to produce 50 muons per event
- FCC barrel inner detector: 5 pixel layers, 8 macro-pixel layers
- put hits into grid with size : $(x, y, z) = (100, 100, 100)$ μm
- merge adjacent hits

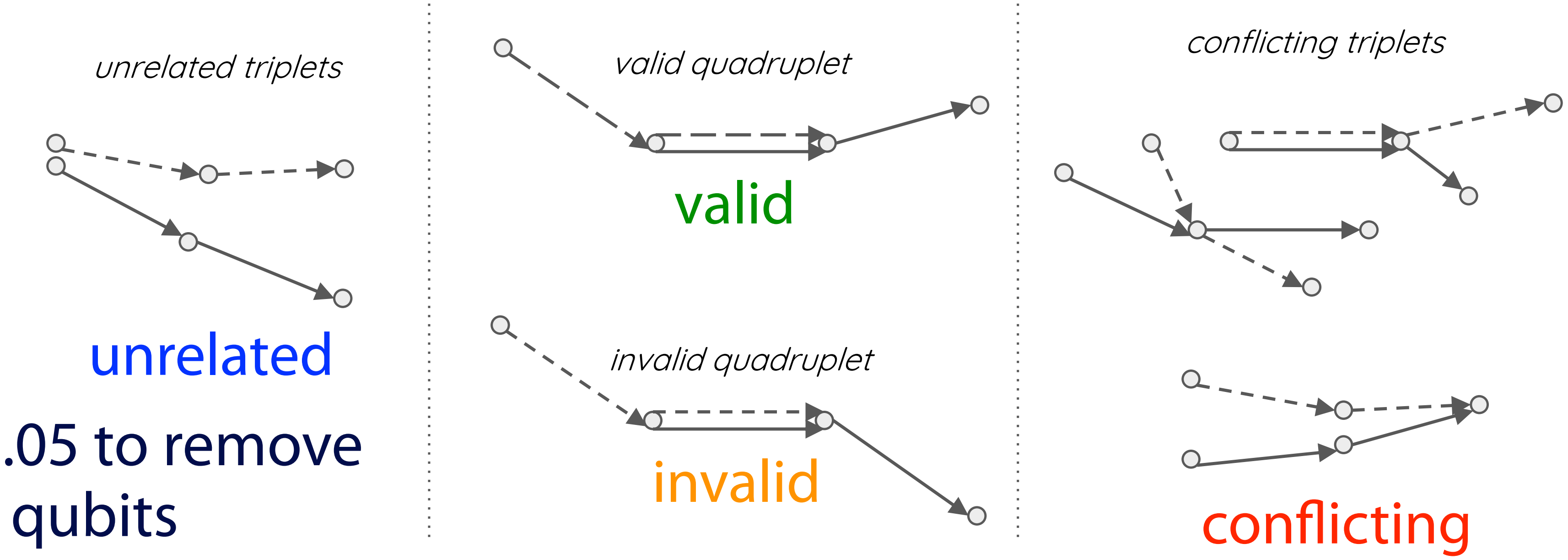
Reconstruction

- triplet reconstruction:
  - require $\Delta R<0.02$ (0.04) for the 3 innermost (other) layer hits
  - scan over (1,2,3), (2,3,4), (3,4,5), … layers up to (9,10,11)
  - only triplets with 3 consecutive layer hits (i.e, no hole) considered
- calculate sign of menger curvature for each triplet (positive or negative charge)
  - no requirement on the size of menger curvature
- calculate $\theta'=|\theta_1-\theta_2|$ (for triplet selection) and $\theta^*=(\theta_1+\theta_2)/2$ (for triplet connection) in *r-z*





$$c(x,y,z) = \frac{1}{R} = \frac{4A}{|x-y||y-z||z-x|}$$



Analysis

- select only triplets with $\theta' < 0.05$ to remove non-aligned triplets in *r-z* ⇒ qubits
- bias weight = 0.01 (fixed)
- interaction strength b_{ij} :
  - triplets at same layers → non-shared hit triplets (unrelated) = $w_{unrelated}$
    → shared hit triplets (conflicting) = $w_{conflict}$
  - triplets shifted by 1 → non-shared hit triplets (unrelated) = $w_{unrelated}$
    → shared hit triplets
      → only 1 shared hit (conflicting) = $w_{conflict}$
      → 2 shared hits (quadruplet candidates)
        → opposite-sign curvature (invalid) = $w_{conflict}$
        → same-sign curvature = $-s_{ij}$ (valid)
  - triplets shifted by ≥ 2 : unrelated triplets = $w_{unrelated}$

$w_{unrelated} = 0$

$w_{conflict} = 1000$

$$s_{ij} = 1 - \frac{1}{2}(dcurv_{ij} + drz_{ij})$$

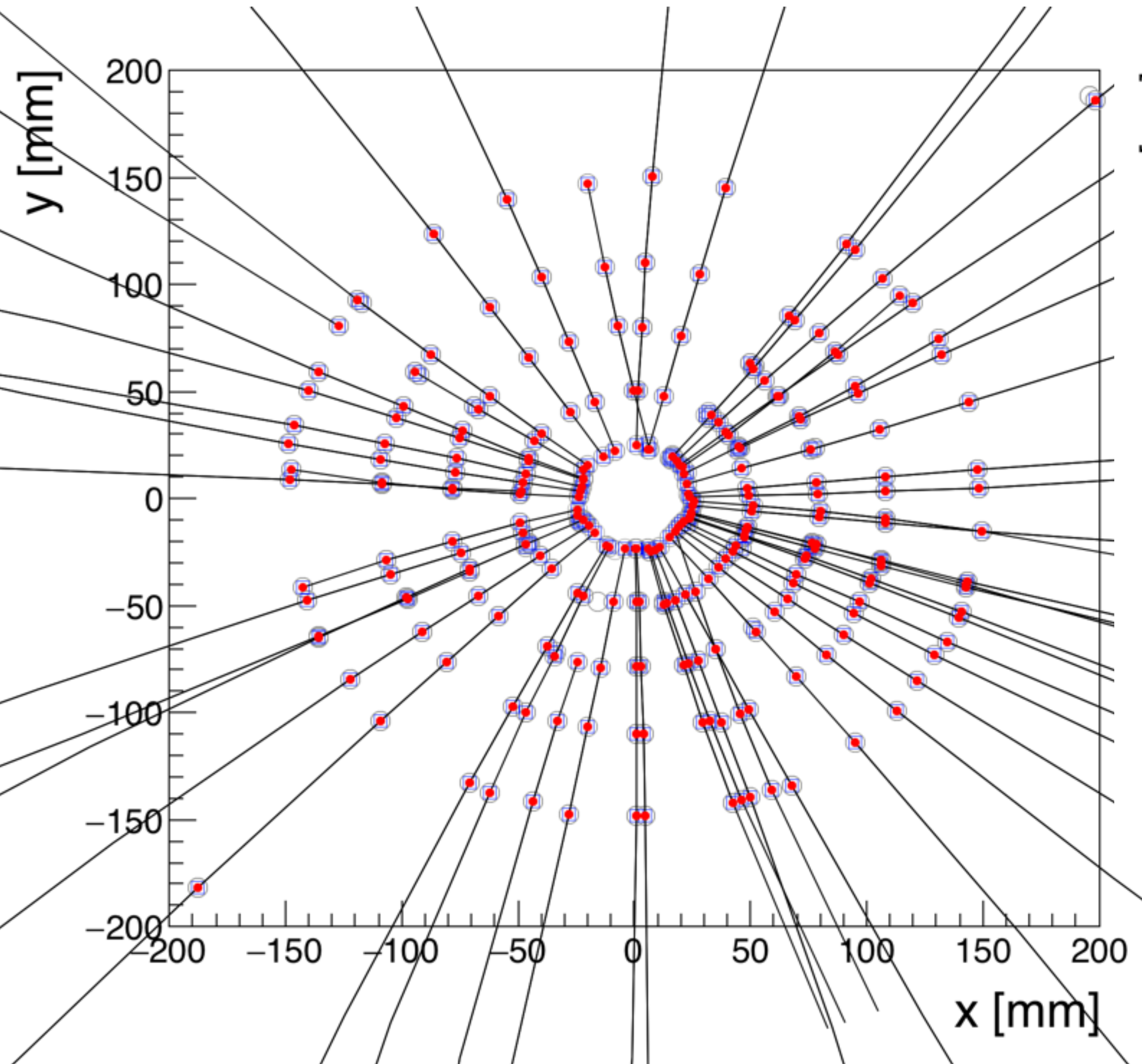
QC code

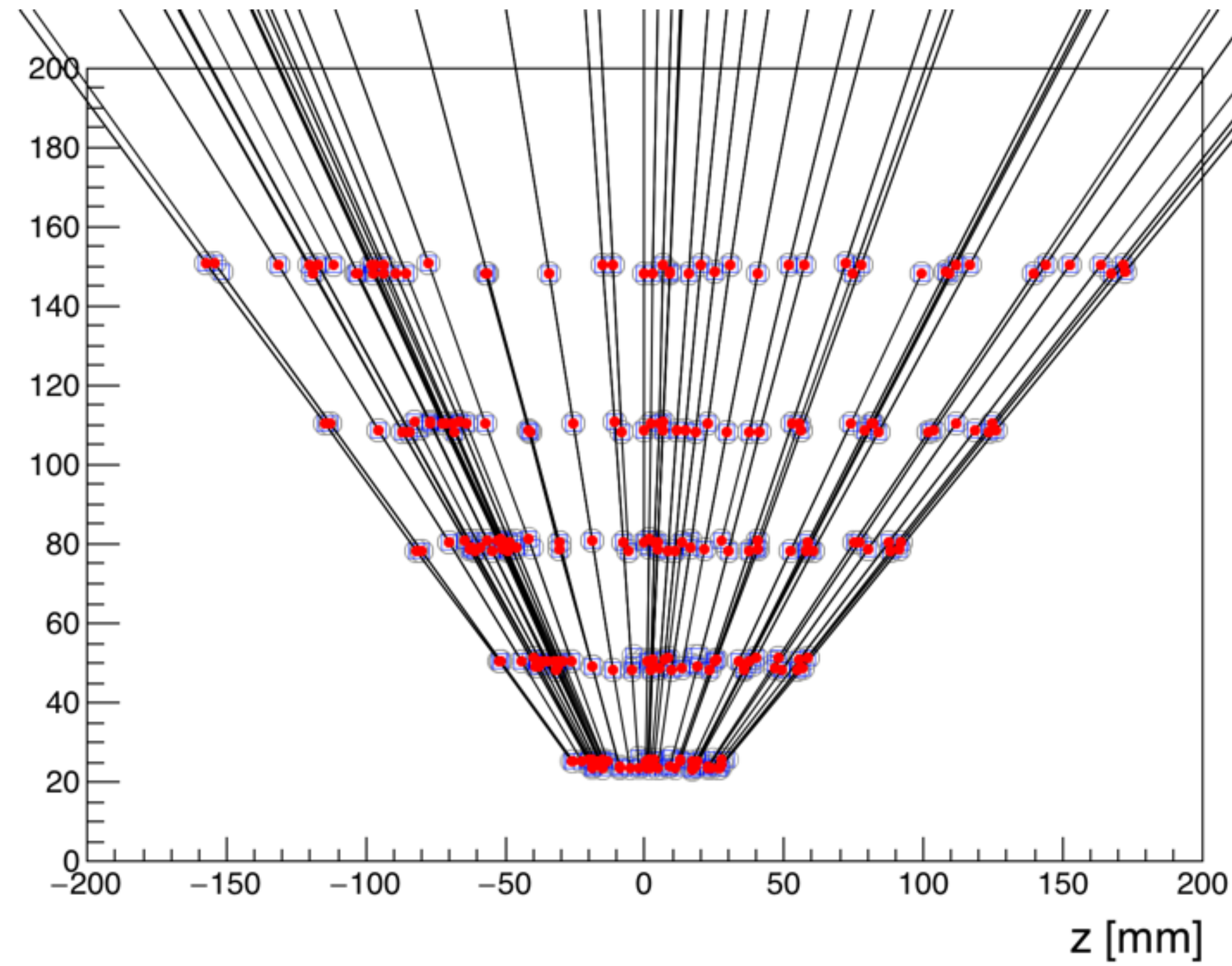
- D-Wave Ocean framework
- Simulated annealing (dwave-neal) and qbsolv for decomposing QUBO problem

Results

50 muons

- hits in all 2416 triplets
- hits in 478 triplets selected by $\theta'<0.05$ cut (= input to QUBO)
- hits in 378 triplets found by annealing (= output from QUBO)





Quick test on D-Wave seems to work: same set of triplets found in D-Wave and simulated annealing

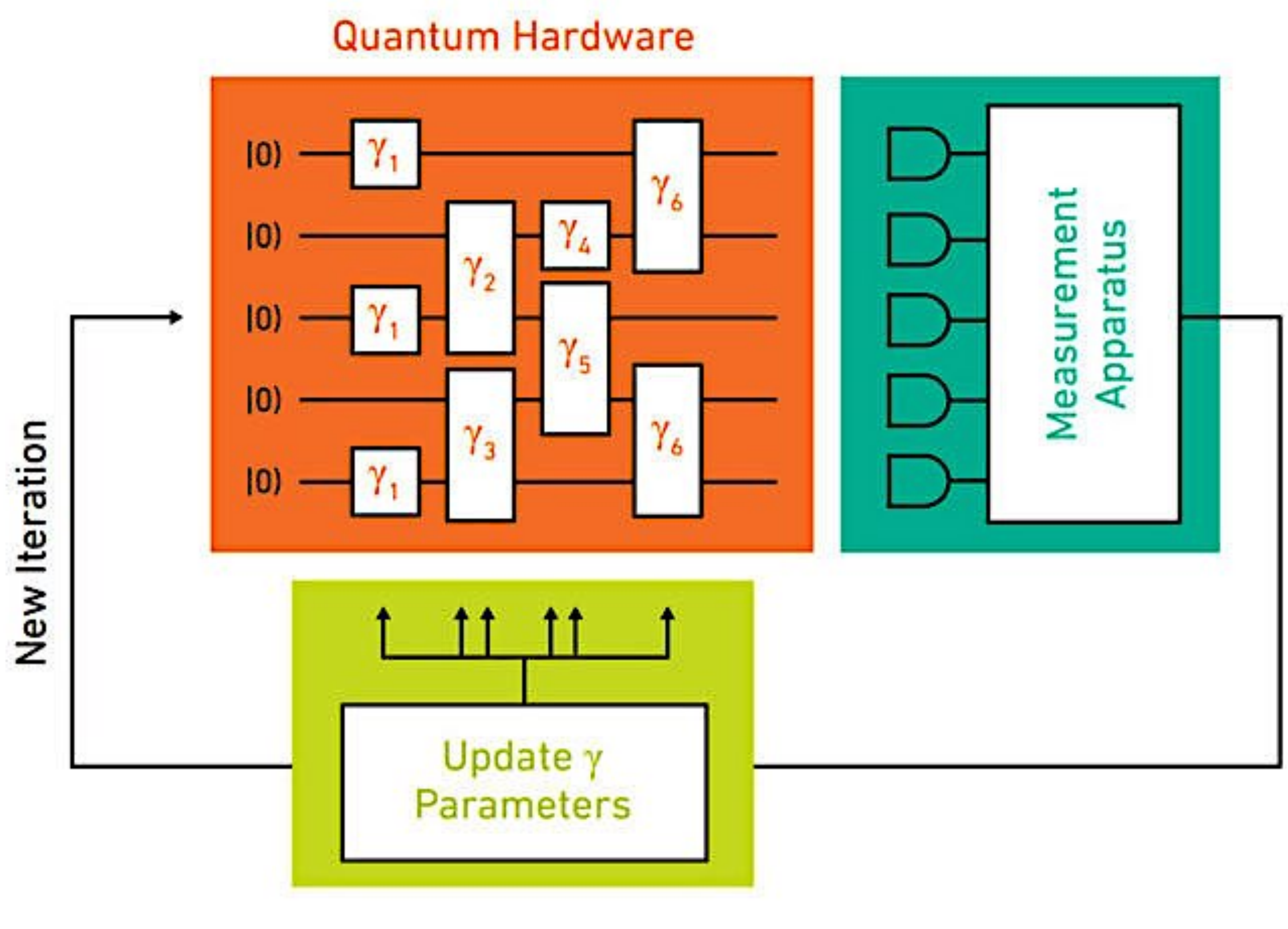
# of found *tracks* (total):

11-hits = 18	7-hits = 3 (42)
10-hits = 5 (23)	6-hits = 5 (47)
9-hits = 7 (30)	5-hits = 6 (53)
8-hits = 9 (39)	4-hits = 10 (63)

Variational Quantum Eigensolver (VQE)

- Represent quantum states (= combination of triplets) by applying parametrized operators (= gates) to initial states
- Obtain energy expectation value by running on quantum hardware
- Update quantum states by updating the parameters (using CPU) such that the expectation value becomes minimum

→ **Can approximately obtain ground state with small # of qubits**



Processing Time

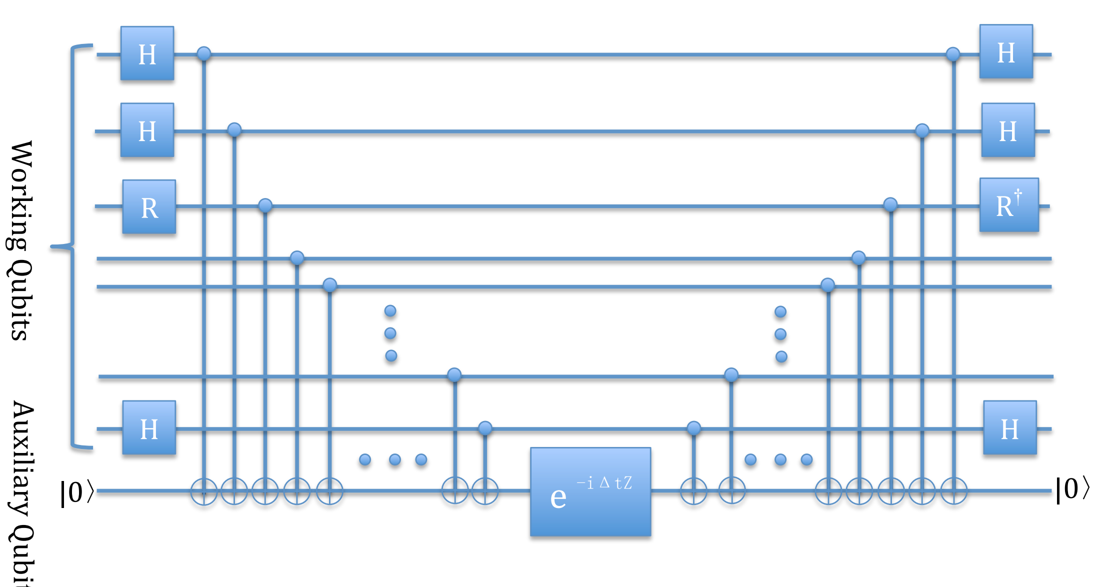
ExactEigensolver		VQE statevector_simulator		VQE qasm_simulator	
#triplets (found)	time [sec]	#triplets (found)	time [sec]	#triplets (found)	time [sec]
3(3)	~0.02	3(3)	~240	3(3)	~240
7(7)	~0.03	7(7)	~900	7(7)	~950
27(10)	~1800	?	?	?	?

Possible Other Directions

Application utilizing more quantum effects?

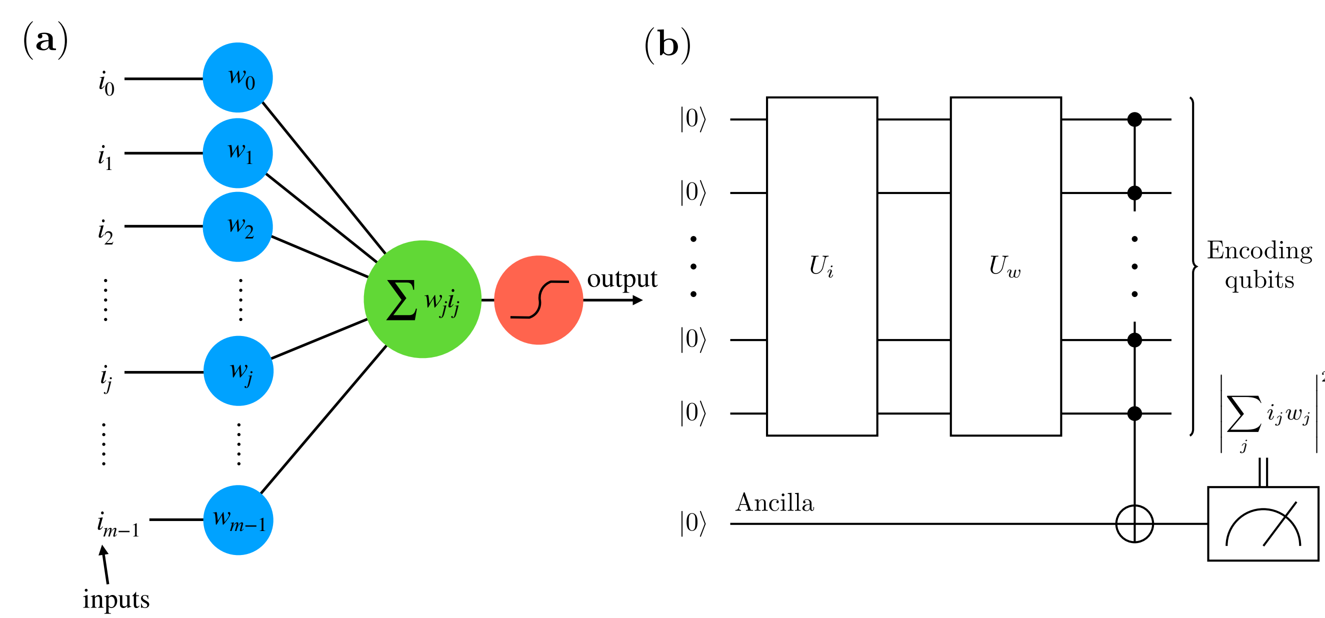
- Simulation
  - Generation of physics events including quantum effects from first principle
  - Simulation of physical events by sampling quantum states?
- Data analysis
  - Machine learning application, e.g, Quantum SVM (→ R. Sawada)
  - Application of deep network?

Quantum simulation of Yang-Mills theory and hadronization
Gate-based simulation of quark/gluon → hadron processes



arXiv:1810.09213

Application of neural network (parceptron) in Quantum computer
Tested on IBM 5-bit system



arXiv:1811.02266