Quantum Tracking for Future Colliders

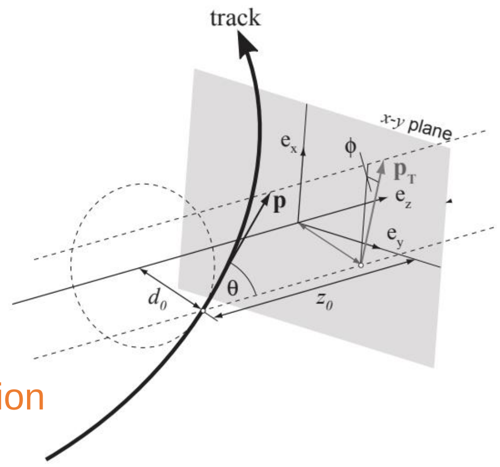
Quantum Computing & Machine Learning Workshop, August 11-14, 2023,
Shandong University

大川 (Okawa) 英希 (Hideki)

中国科学院 高能物理研究所 (Institute of High Energy Physics, CAS)

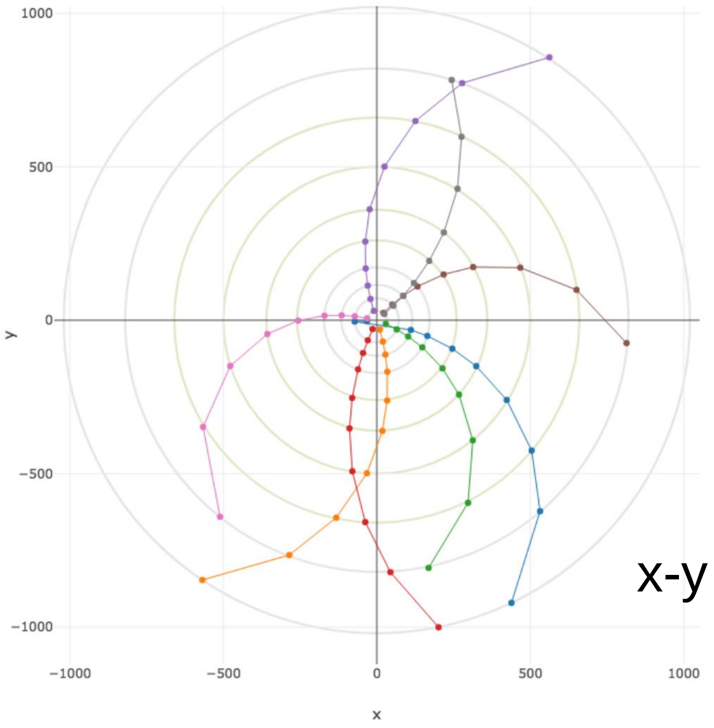
Track Reconstruction

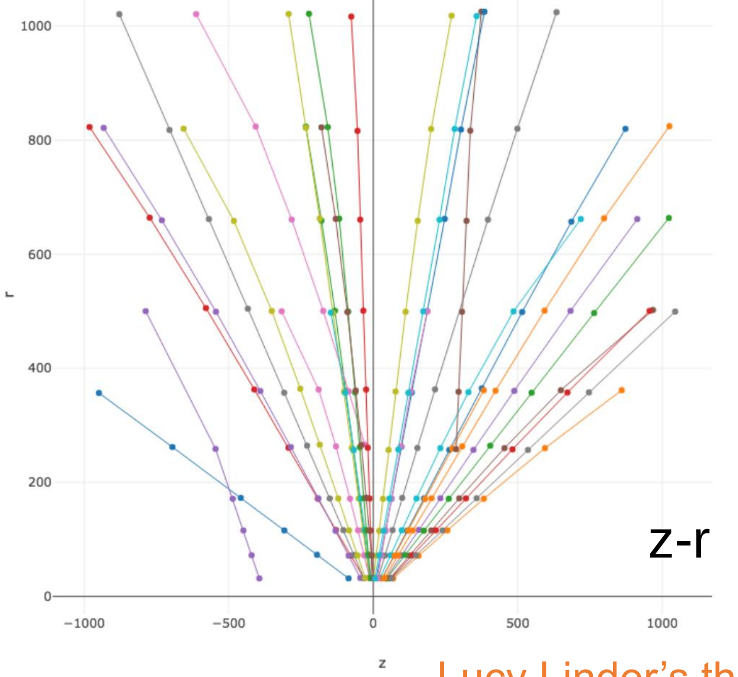
ATLAS Software Documentation



Global track parameters e.g. wrt. perigee

$$\left(d_0, z_0, \phi, \theta, \frac{q}{p}\right)$$

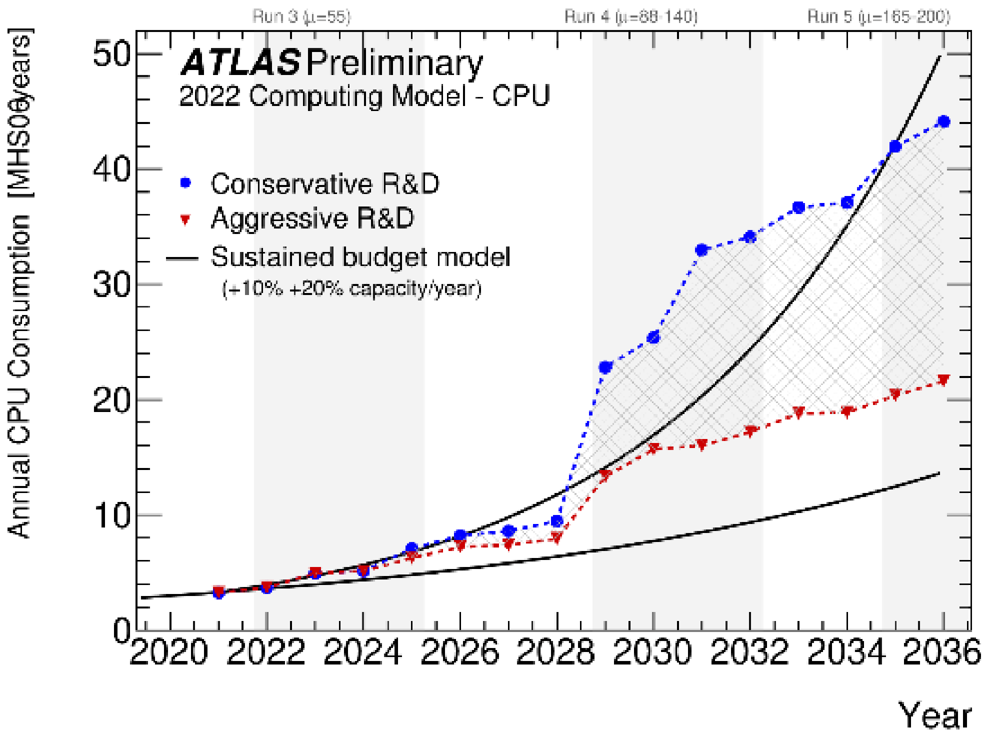




Lucy Linder's thesis

- Measuring curvature of particle trajectory bent in a magnetic field will provide momentum.
- Particle trajectory (track) will be reconstructed from hits in the silicon detectors (have many irrelevant hits from secondary particles)
- One of the most crucial reconstruction in collider experiments.

High Luminosity LHC & Beyond

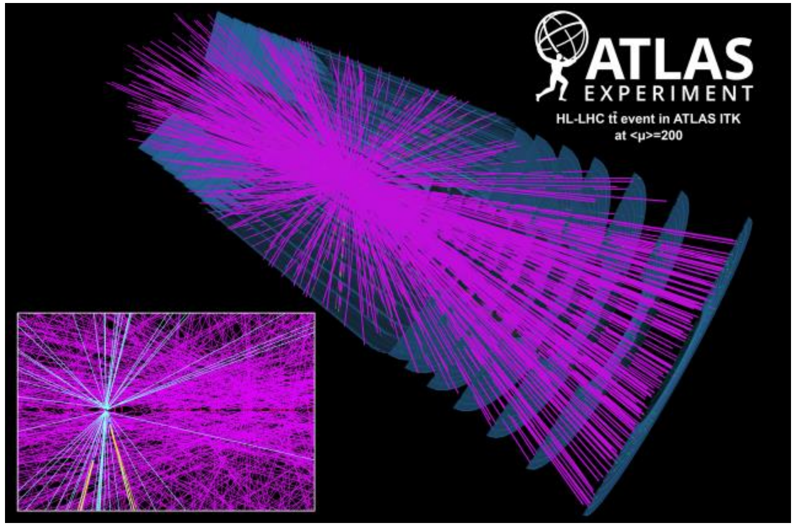


- At the HL-LHC, **we will enter the "Exa-byte" era. Annual computing cost will increase by a factor of 10-20**
- **Without various innovations, the experiment will not be able to operate.** GPUs and other state-of-the-art technologies will be the baseline at the HL-LHC.
- **Quantum computing may bring another "leap".**

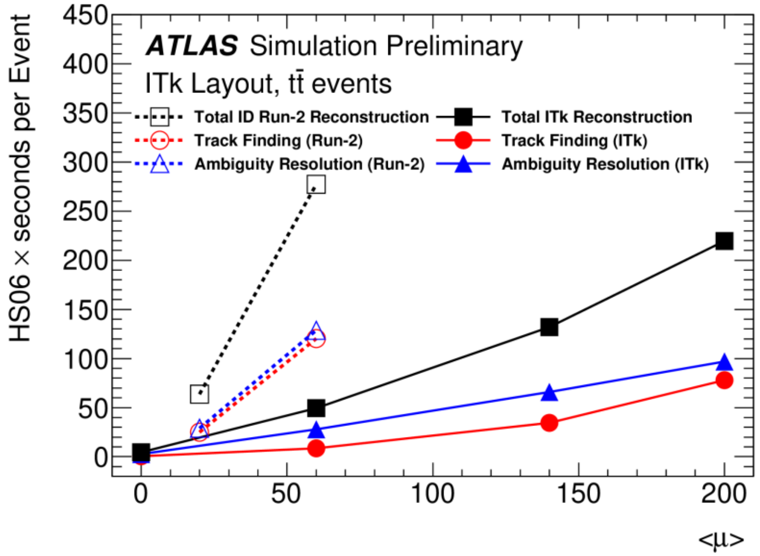
- Two of the highly CPU consuming components: **(1) track reconstruction for both data/simulation & (2) simulation of shower development in the calorimeter.**
  - Collaborative projects w/ DESY. I will only cover tracking in this talk. Xiaozhong Huang will present the latter tomorrow.
- Tackling these challenges will also be useful for other future colliders, such as CEPC & SppC etc.

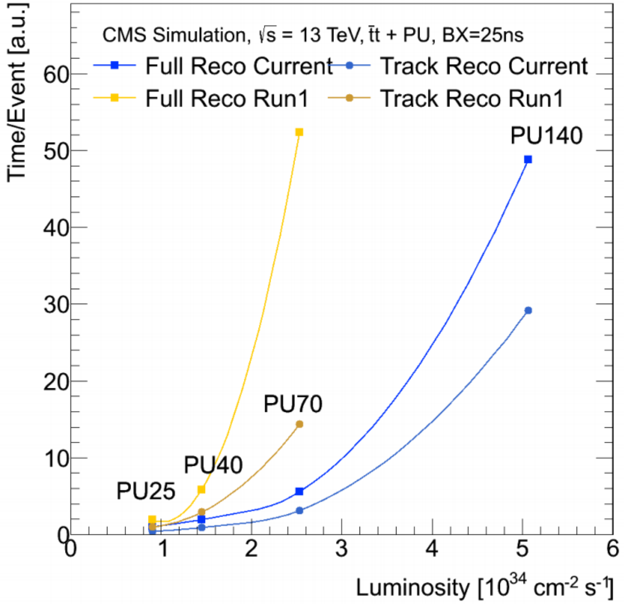
Track Reconstruction at LHC & HL-LHC

https://cds.cern.ch/record/1966040



ATL-PHYS-PUB-2019-041

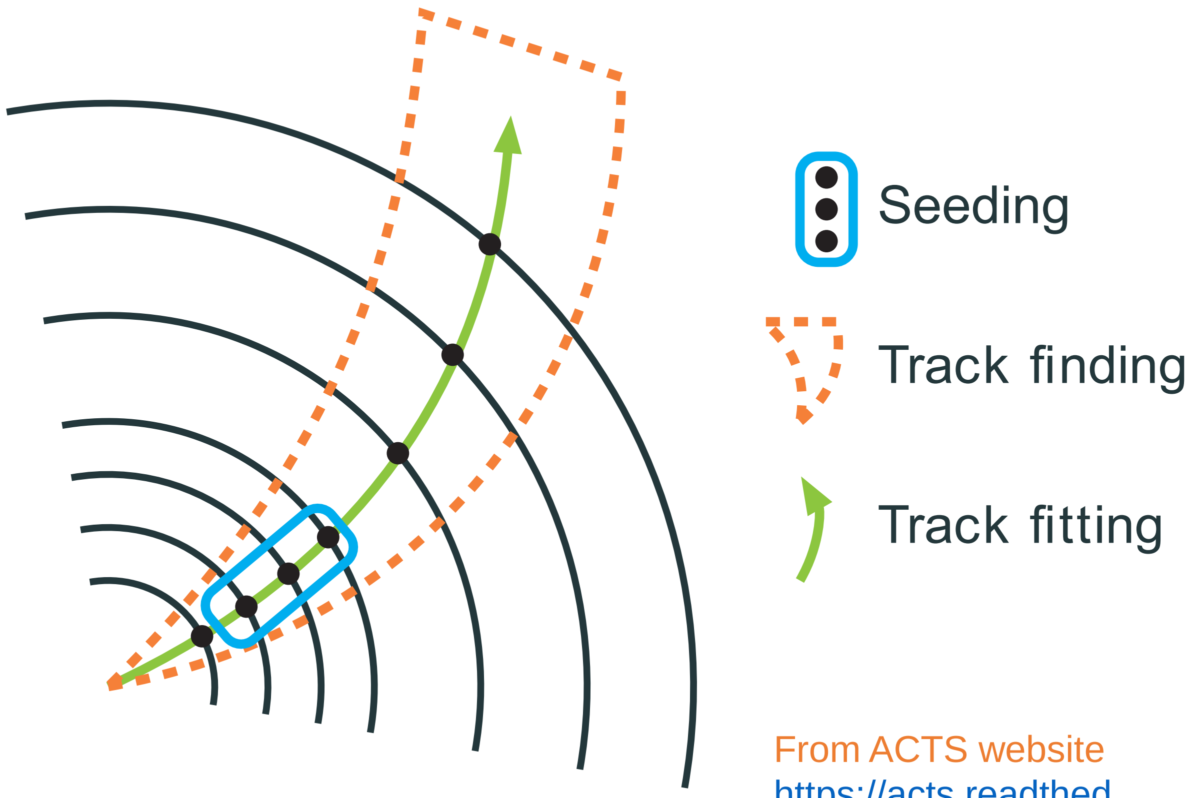




	Run 1	Run 2	HL-LHC
μ	21	40	150-200
Tracks	~280	~600	~7-10k

- At the HL-LHC, additional interactions per bunch crossing becomes exceedingly high & **CPU time blows up with more pileup**.
- GPU & ML-based approaches could be considered as a baseline, but quantum ML may play an important role.

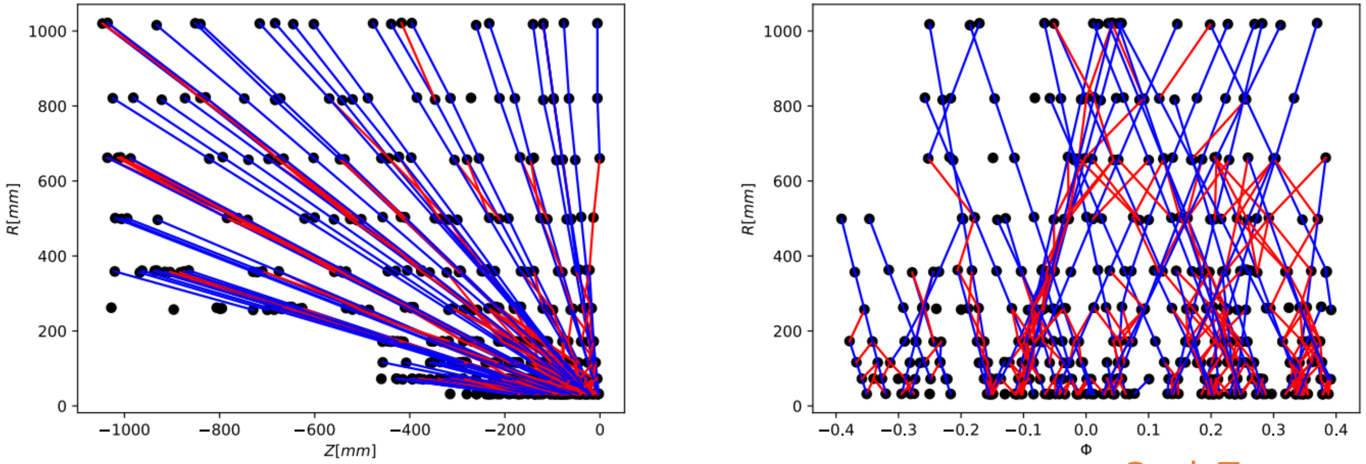
Classical Benchmark: Kalman Filter



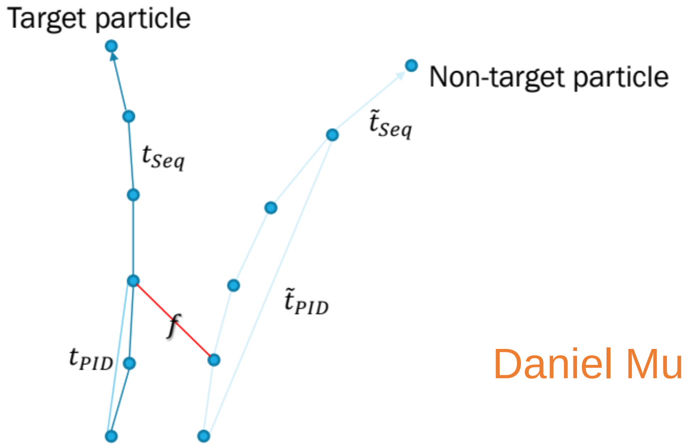
From ACTS website
https://acts.readthedocs.io

- In high energy collider experiments, Kalman Filter technique (e.g. implemented in A Common Tracking Software [ACTS]) has been often used as a standard algorithm.
- Seeding from the inner layers, extrapolated to predict the next hit & iterated to find the best quality combination.

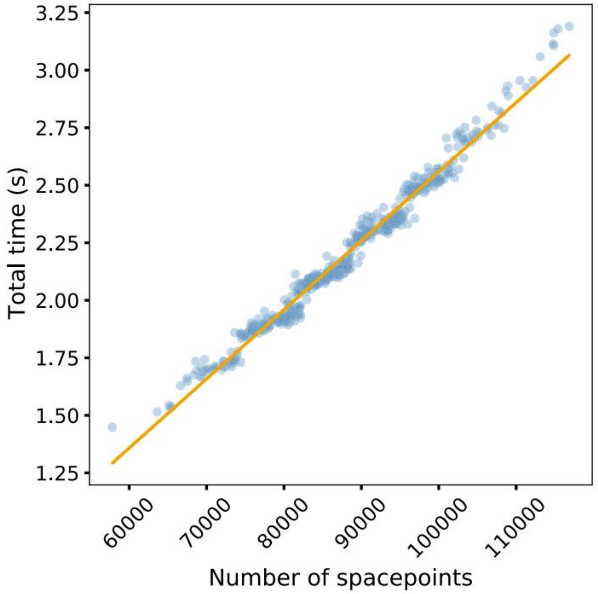
Classical ML Approaches



Cenk Tueysuez

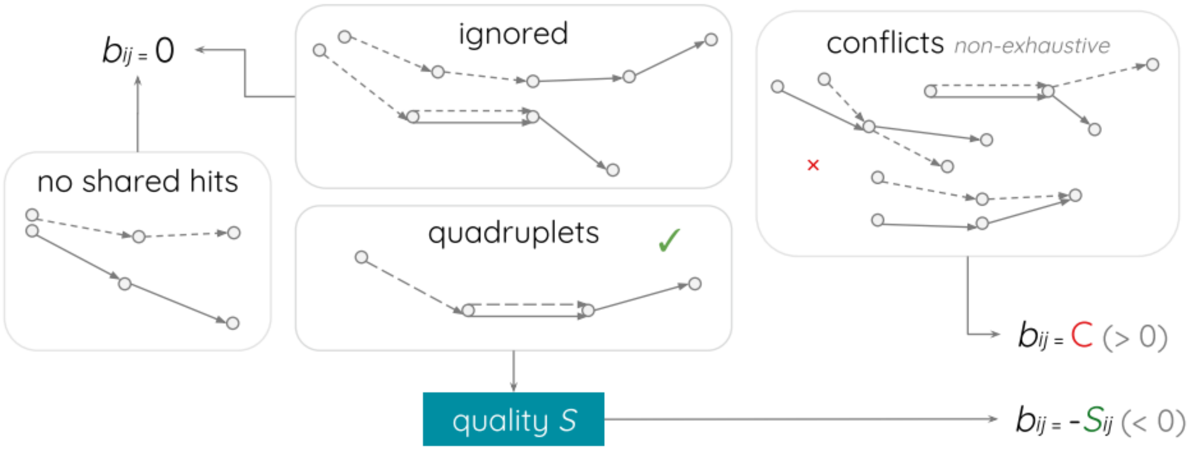


Daniel Murnane



- Graph neural network (GNN) is actively investigated in the LHC & BES-III communities. (Andreas Salzburger's & 贾晓倩's talks)
  - There are also studies using CNN & Point Net at BES-III
- Silicon hits can be regarded as "**nodes**" & connected segments as "**edges**"
- Computing time scales linearly with number of tracks

Quantum Approach: QUBO



F. Bapst et al. Comp. Soft. Big Sci. 4 (2019) 1.

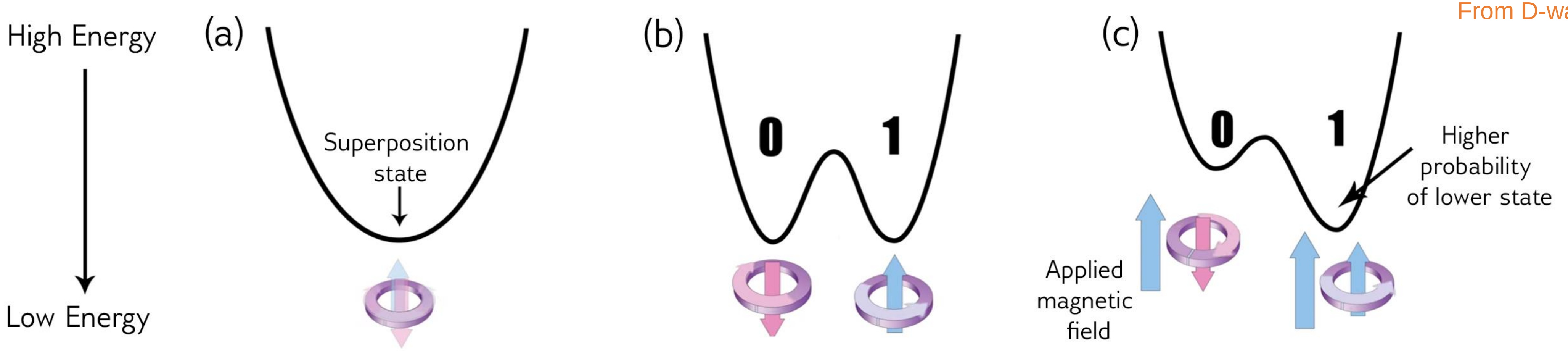
$$O(a,b,T) = \underbrace{\sum_{i=1}^{N} a_i T_i}_{\text{Quality of triplets}} + \underbrace{\sum_{i}^{N} \sum_{j<i}^{N} b_{ij} T_i T_j}_{\text{Compatibility b/w triplet pairs}}$$

b_{ij} = 0 (if no shared hit)
= 1 (if conflict)
= $-S_{ij}$ (if two hits are shared)

- Triplets (segments w/ 3 hits) are formed from doublets (segments w/ 2 hits).
- Triplets are used to reconstruct tracks & can be regarded as a **quadratic unconstrained binary optimization (QUBO)** problem. (QUBO matrices for tracking is generally sparse)
- Minimizing QUBO is equivalent to searching for the ground state of the Hamiltonian.

Quantum Annealing Approach

From D-wave website

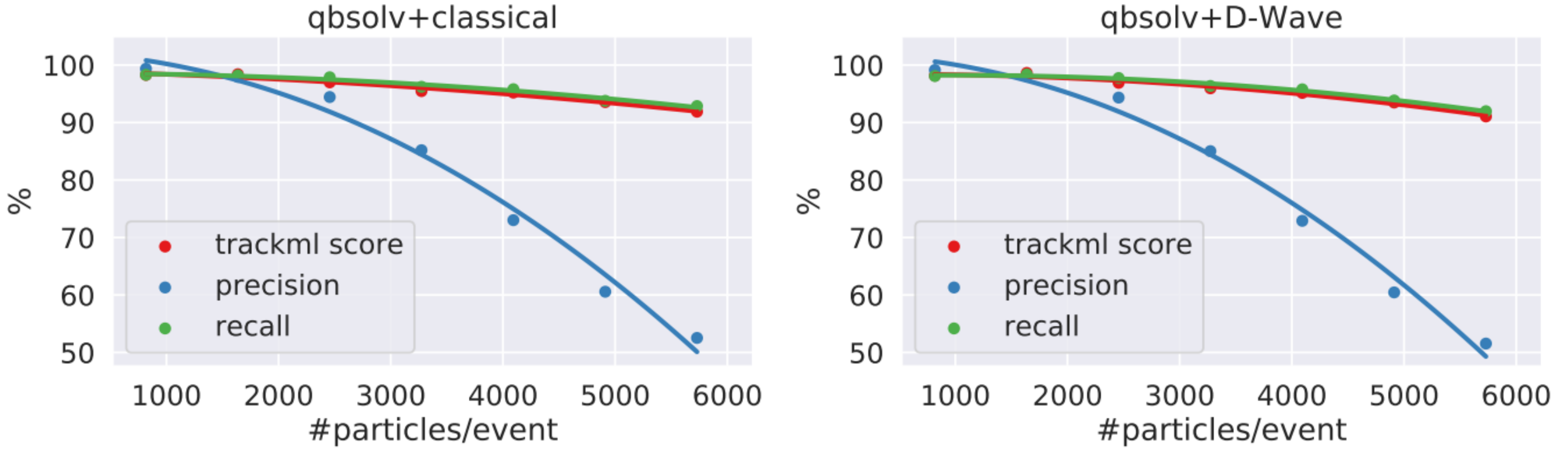


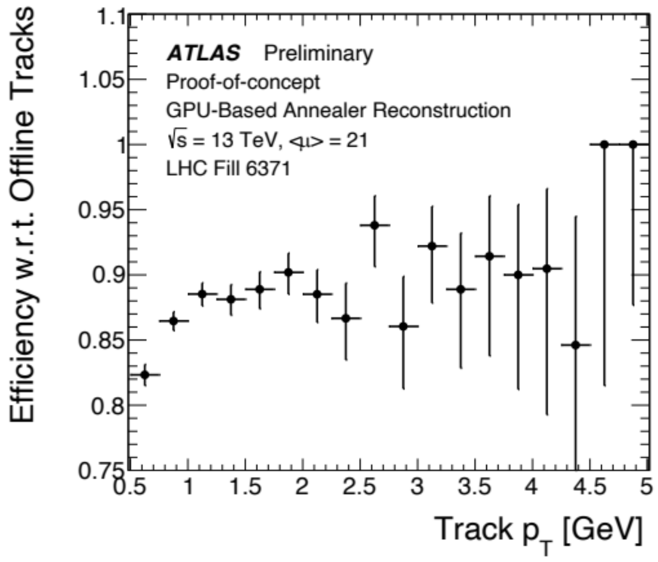
- Quantum annealer looks for the global minimum of a given function with quantum tunneling: a natural machine to search for the ground state of a Hamiltonian.
- D-Wave currently provides 5000+ qubit service (7440 qubits may be available in 2023-2024).
- Pros: High number of qubits available (concept fundamentally different from quantum gates).
- Cons: Can only run QUBO problems. Also, not all qubits are available for fully connected graphs (only a few hundred qubits).
- Simulator studies can be pursued in local machines as well as at the IHEP platform (Yujiang Bi's talk)

Previous Studies w/ Q. Annealing

- Previous studies w/ 1000-qubit machine show that efficiency is almost stable w/ # of particles, but purity (precision) degrades.
- Simulator provides consistent results w/ hardware!
- There are also ongoing studies in LHC-ATLAS experiment implementing GNN w/ annealers.

F. Bapst et al. Comp. Soft. Big Sci. 4 (2019) 1.



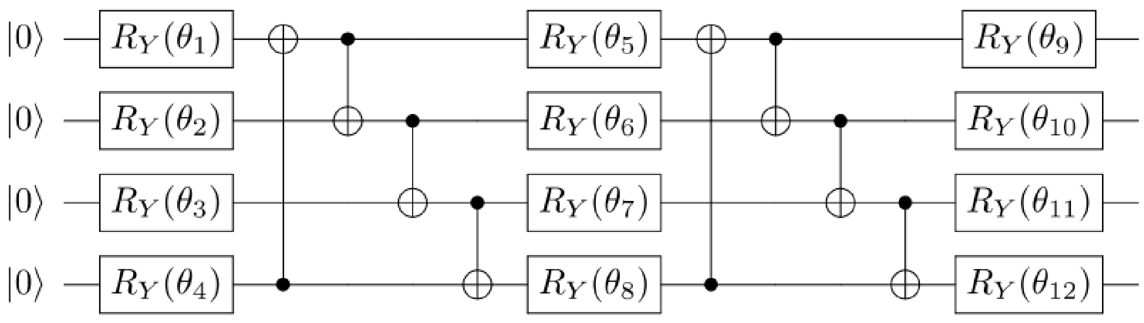


Previous DESY Studies

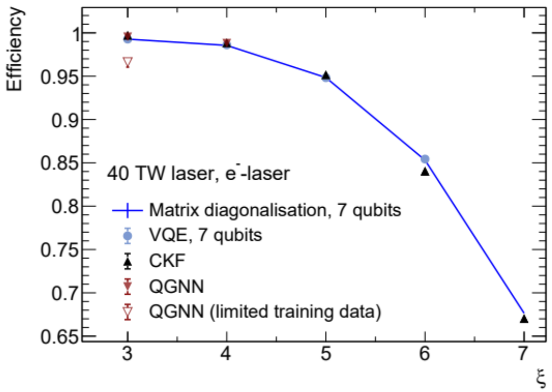
- QUBO can be mapped to Ising Hamiltonian and be solved using Variational Quantum Eigensolver (VQE) or Quantum Approximate Optimization Algorithm (QAOA) w/ quantum gates.

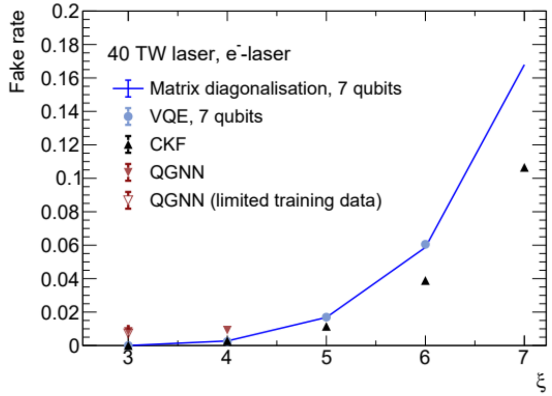
$$\mathcal{H} = -\sum_{n=1}^{N}\sum_{m<n}\bar{b}_{nm}\sigma_n^x\sigma_m^x - \sum_{n=1}^{N}\bar{a}_n\sigma_n^x$$

- Previous LUXE studies considered TwoLocal ansatz w/ R_Y gates & circular CNOT entangling pattern w/ IBM (A. Crippa et al., arXiv:2304.01690, L.Funcke et al., arXiv:2202.06874)
- QAOA did not perform well & optimization was left for future studies. → A scope of this talk



Only 4 qubits drawn for simplicity

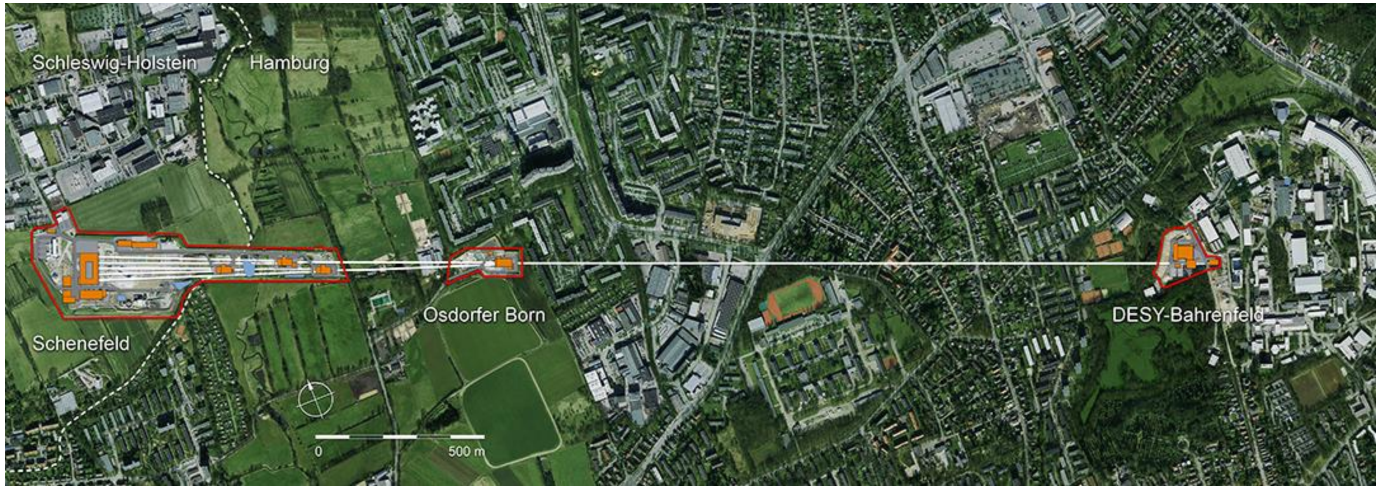


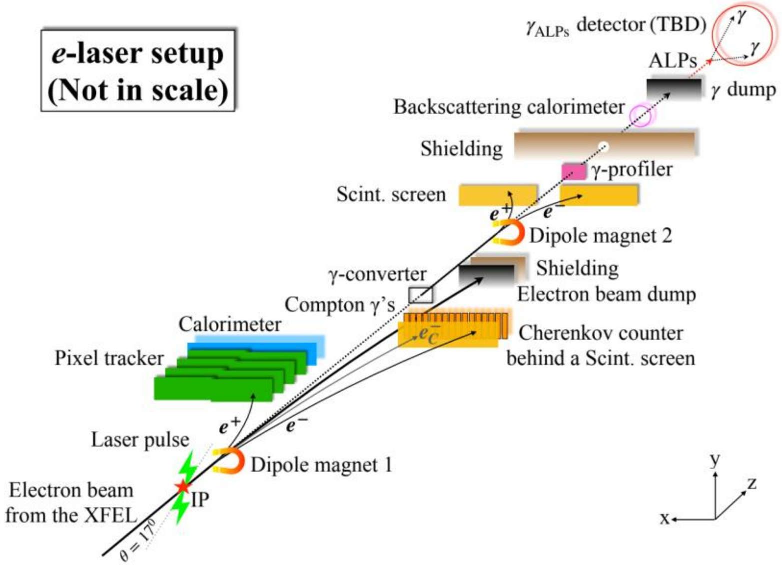


LUXE as a Benchmark

Conceptual Design Report
arViv:2102.02032

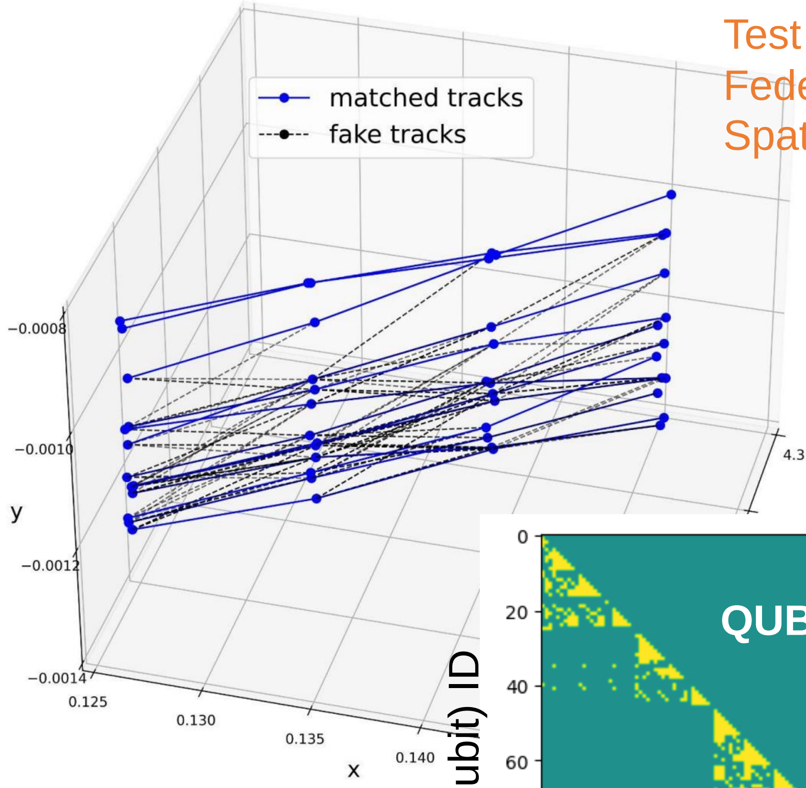
- LUXE (Laser Und XFEL Experiment)
  - QED studies under the strong-field regime (i.e. non-perturbative)
  - Exploits European XFEL electron beam and high-power laser
  - Also searches for new physics (e.g. ALP)



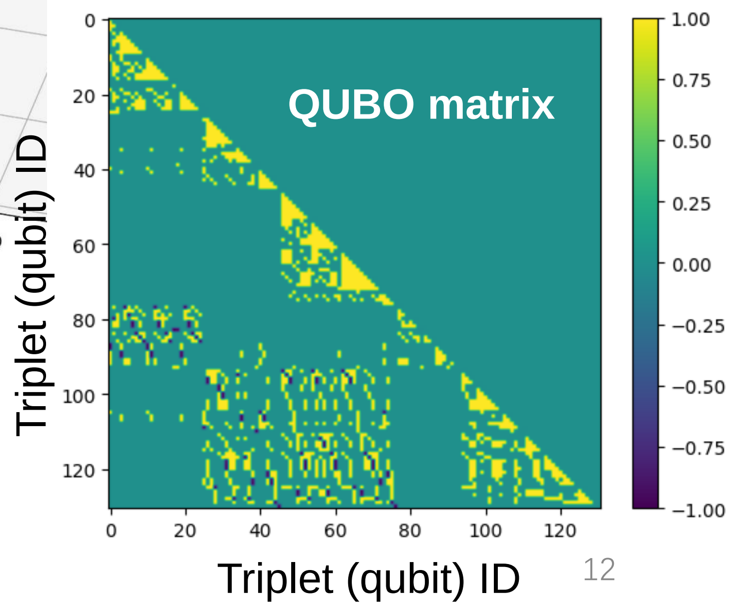


QUBO from LUXE Simulation

- DESY team provided a test QUBO benchmark for 131 qubits that would fit in high qubit machines (e.g. quafu 136). → ~$3\text{x}10^{39}$ possible solutions
- A simulated event from LUXE experiment; originally 10500 particles, a subset of 13 tracks chosen from the densest region (thus a challenging condition).
- **QUBO matrix is pretty sparse**, as is the nature of collider experiments & tracking

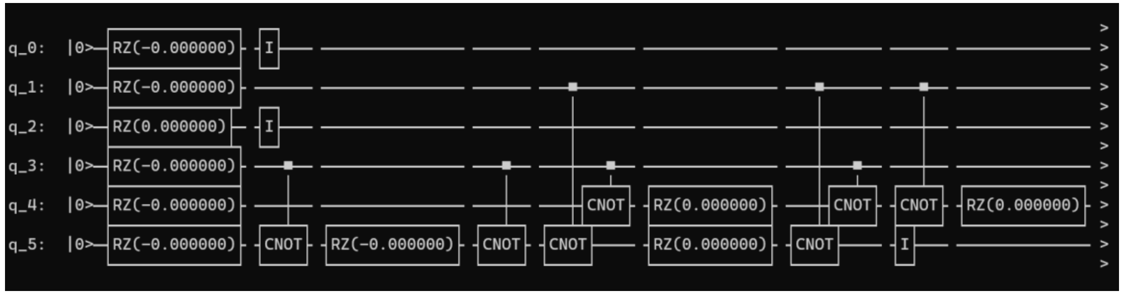


Test sample provided by Federico Meloni, David Spataro et al. (DESY)



QAOA in OriginQ (本源)

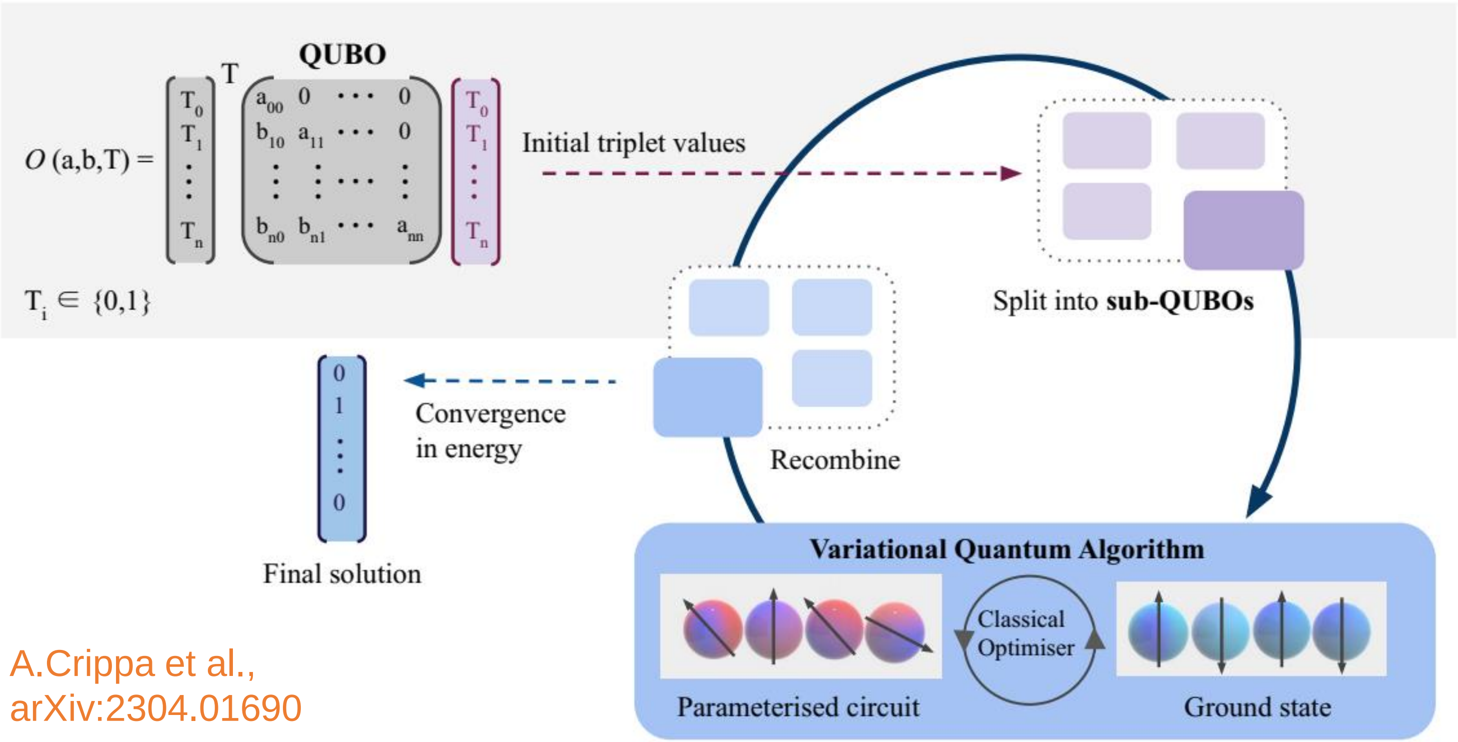
- VQE & QAOA libraries implemented in pyqpanda-algorithm by OriginQ (本源).
- Adopts Quantum Alternative Operator Ansatz for QAOA.
- Utilizes CVaR loss function optimization (P. Barkoutsos et al., Quantum, 2020, 4: 256)
- 6 qubit machine (Wu-Yuan) is used for the real hardware computation in this talk.



An example of circuits from the actual run

Sub-QUBOs

- **Number of qubits required is determined by the number of triplet candidates** → Obviously cannot cover the full QUBO for tracking in the NISQ era
- QUBO is split into sub-QUBOs of size N (N=7 in previous LUXE studies for IBM machine). **Here, I used N=6 to match with OriginQ hardware.**



A.Crippa et al., arXiv:2304.01690

- There are various sub-QUBO algorithms proposed: qbsolv (now moved to dwave-hybrid library), for example.
- I adopted a sub-QUBO method using multiple solution instances from Y. Atobe, M. Tawada, N. Togawa, IEEE Trans. Comp. 71, 10 (2022) 2606.

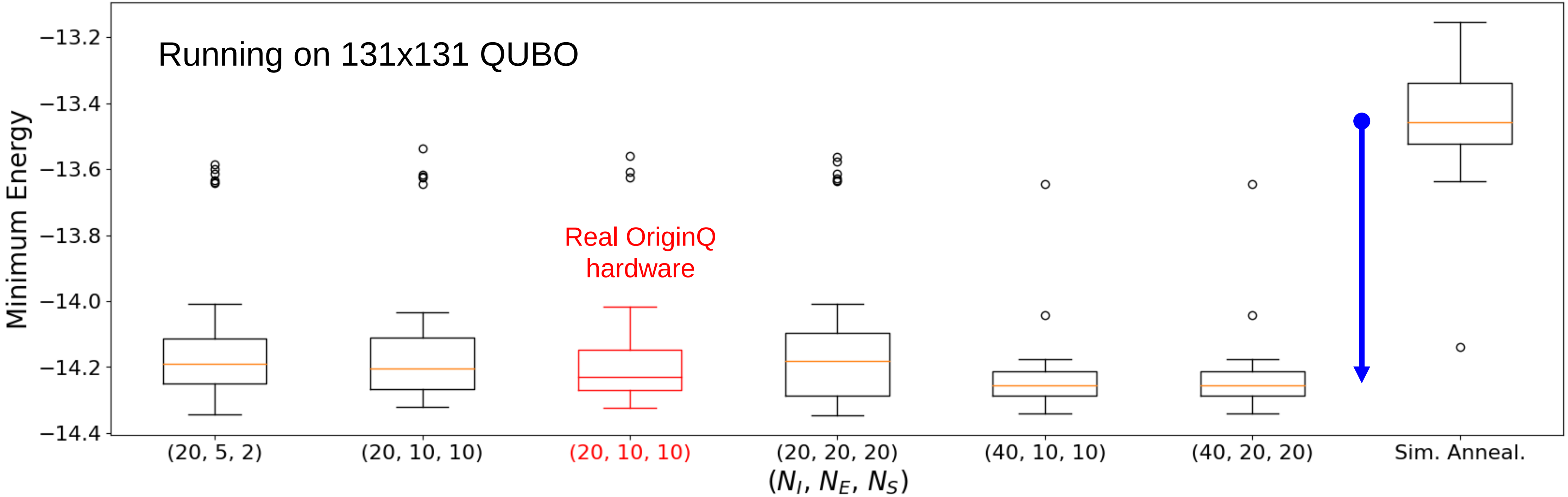
$$QUBO = \sum_i a_i x_i + \sum_{i>j} b_{ij} x_i x_j$$

Multiple Solution Instances

- 3 parameters (N_I, N_E, N_S) in this sub-QUBO method.
- Extract N_I quasi-optimal solutions from full-QUBO classically.
- Randomly select N_S solution instances from N_I.
- Focus on particular binary variable x_i. Rank them in accordance to how much they vary over N_S solution instances. Highly varying x_i will be included in the sub-QUBO model.
- Pick-up process of N_S solution from quantum computing is repeated N_E times & N_E sub-QUBO models are considered.
- Returns a pool of N_I solutions & the best solution will be chosen.

Y. Atobe, M. Tawada, N. Togawa, IEEE Trans. Comp. 71, 10 (2022) 2606

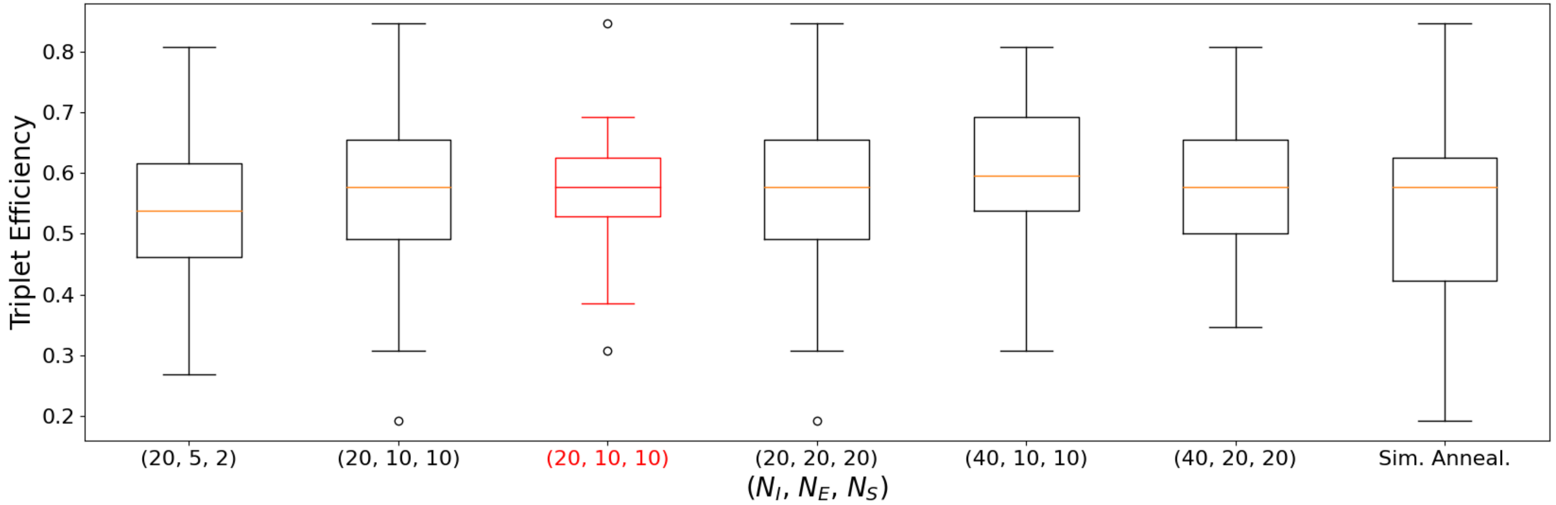
Preliminary sub-QUBO Results

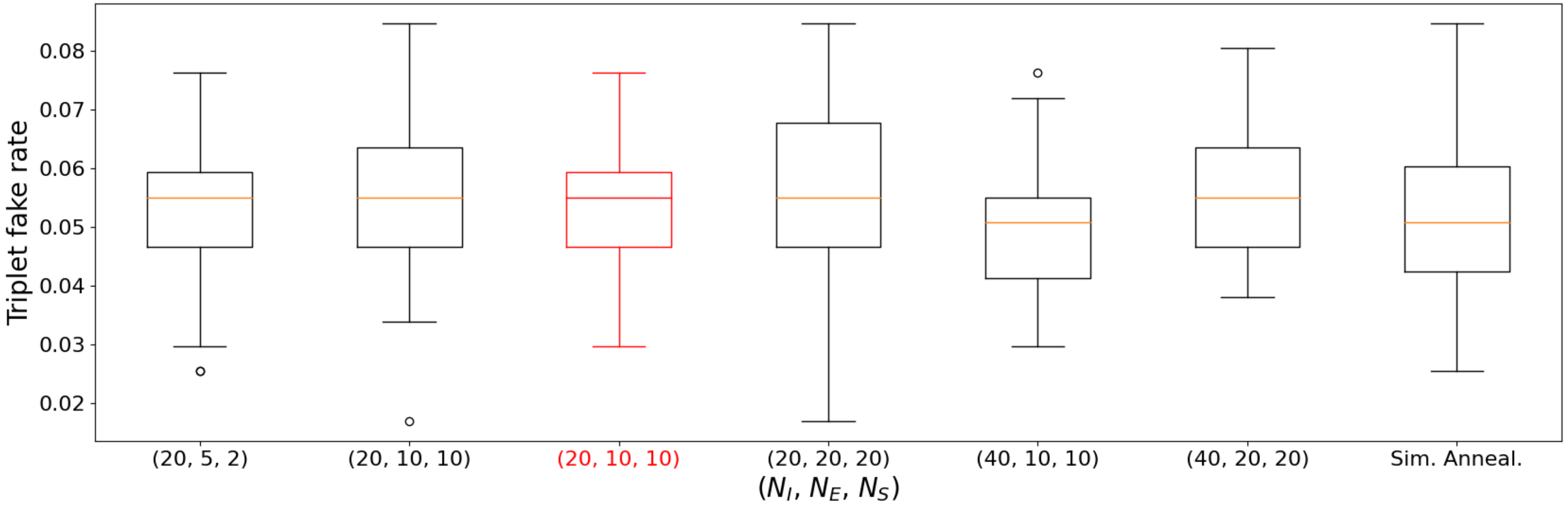


- Ran 40 shots to compare the performance and stability. 3 layers used in QAOA.
- No significant dependence on (N_I, N_E, N_S) but slightly better & smaller fluctuations with larger parameters. Compatible performance between OriginQ simulator & actual hardware!
- **Visible improvement w/ sub-QUBO compared to the simulated annealing only!**

WIP: Triplet Efficiency & Fake Rate

- Evaluated triplet efficiency & fake rate.
- Only 1 event w/ 13 true tracks (i.e. 26 true triplets).
- Relatively low eff. but likely reasonable in the very dense conditions (see p.10).
- Fake rate roughly compatible w/ the various algorithms in the previous LUXE studies.
- Need detailed optimization & more data to conclude.





Near Future Plans

- **Tracking definitely requires high-qubit machines:**
  - Interested in higher qubit machines from OriginQ (本源)
  - Also currently iterating with quafu experts to run QAOA on 136-qubit machine.
  - I’m happy to chat for any other options in China ☺
- Further investigations on parameter optimization in VQE & QAOA as well as in QUBO & sub-QUBO algorithm.
- Look into more datasets & algorithms (e.g. QGNN) to pursue detailed performance studies
  - Publicly available tracking samples w/ HL-LHC conditions
  - CEPC simulation samples

Contact: okawa@ihep.ac.cn

Summary

- Tracking & calorimeter simulation are highly CPU-consuming tasks in the HL-LHC era & beyond. Classical ML methods are bringing in promising improvement.
- Another leap from quantum machine learning would be highly exciting.
- Pursuing international collaboration w/ DESY & exploiting opportunities with Chinese quantum computers & cloud services.
- Presented some preliminary studies on the quantum tracking using OriginQ simulator & real hardware. The sub-QUBO model presented here shows promising performance.
- (Also working on QGAN for calorimeter simulation w/ OriginQ machine; not presented today)
- Further investigations are ongoing. Stay tuned!

谢谢！ Thank you for listening!

非常感谢本源和Quafu专家老师们的反馈和建议！

Backup

$$QUBO = \sum_i a_i x_i + \sum_{i>j} b_{ij} x_i x_j$$

Multiple Solution Instances

Algorithm 2. Proposed Hybrid Annealing Method

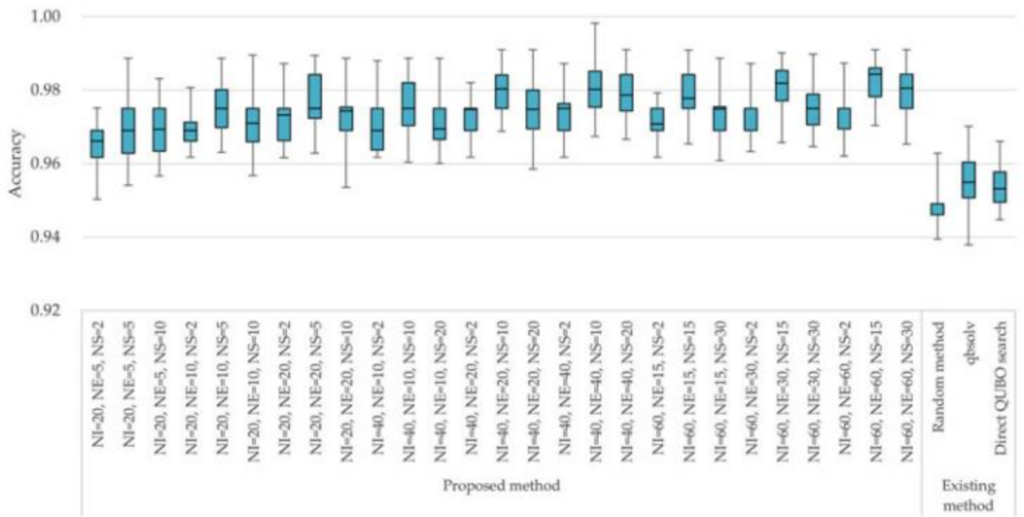
```
procedure PROPOSED METHOD WITH MULTI-INSTANCES
  for (i = 1; i ≤ N_I; i++) do
    X_i ← Initialize(QUBO)
    Pool ← AddInstancePool(Pool, X_i)
  X_best ← FindBest(Pool)
  while not converged do
    for (i = 1; i ≤ N_I; i++) do
      X_i ← Optimize(QUBO, X_i)
                                   ▷ Using a classical computer
    for (i = 1; i ≤ N_E; i++) do
      X_1, X_2, ..., X_{N_S} ← SelectInstance(Pool, N_S)
      for (j = 1; j ≤ n; j++) do
        for (k = 1; k ≤ N_S; k++) do
          c_j ← c_j + x_{k,j}
        d_j ← |c_j − N_S/2|
      subQUBO ← Extract(ArgSort(d_1, d_2, ..., d_n), m, X_t)
      X' ← Optimize(subQUBO, X_t)
                                   ▷ Using an Ising machine
      Pool ← AddInstancePool(Pool, X')
    X_best ← FindBest(Pool)
    Pool ← ArrangeInstancePool(Pool, N_I)
  return f(X_best), X_best
```

Algorithm 3. Random Method

```
procedure RANDOM METHOD
  X ← Initialize(QUBO)
  X_best ← X
  while not converged do
    X ← TabuSeach(QUBO, X)
    subQUBO ← RandomExtract(QUBO, m, X)
    X ← Optimize(subQUBO, X)
    if f(X) < f(X_best) then
      X_best ← X
  return f(X_best), X_best
```

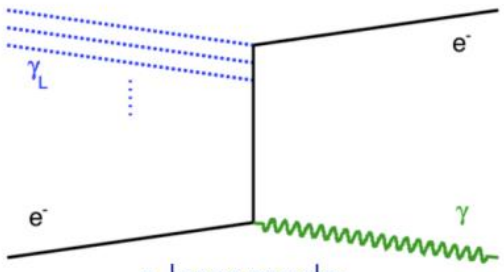
Algorithm 4. Qbsolv[10]

```
procedure QBSOLV
  X ← Initialize(QUBO)
  X_best ← TabuSeach(QUBO, X)
  index ← OrderByImpact(QUBO, X_best)
  while not converged do
    for (i = 0; i < Size(QUBO); i+ = Size(subQUBO)) do
      subQUBO ← Decompose(QUBO,
                      index[i : i+Size(subQUBO)-1], X_best)
      subX ← Optimize(subQUBO, X_best)
      X[index[i : i+Size(subQUBO)-1]] ← subX
    X ← TabuSearch(QUBO, X)
    index ← OrderByImpact(QUBO, X)
    if f(X) < f(X_best) then
      X_best ← X
  return f(X_best), X_best
```

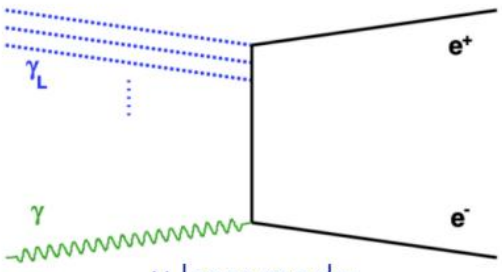


Y. Atobe, M. Tawada, N. Togawa, IEEE Trans. Comp. 71, 10 (2022) 2606

LUXE



e-laser mode:
Non-linear Compton scattering



γ-laser mode:
Non-linear Breit-Wheeler pair creation