EPJ Web of Conferences **295**, 12001 (2024) https://doi.org/10.1051/epjconf/202429512001
CHEP 2023

Application of quantum computing techniques in particle tracking at LHC

Wai Yuen Chan[1,*], *Daiya* Akiyama[2], *Koki* Arakawa[2], *Sanmay* Ganguly[1], *Toshiaki* Kaji[1], *Juri* Minami[2], *Ryu* Sawada[1], *Junichi* Tanaka[1], *Koji* Terashi[1], and *Kohei* Yorita[2]

[1]International Center for Elementary Particle Physics (ICEPP), The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan
[2]Department of Physics, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan

Abstract. After the next planned upgrades to the LHC, the luminosity it delivers will more than double, substantially increasing the already large demand on computing resources. Therefore an efficient way to reconstruct physical objects is required. Recent studies show that one of the quantum computing techniques, quantum annealing (QA), can be used to perform particle tracking with efficiency higher than 90% in the high pileup region in the high luminosity environment. The algorithm starts by determining the connection between the hits, and classifies the topological objects with their pattern. The current study aims to improve the pre-processing efficiency in the QA-based tracking algorithm by implementing a graph neural network (GNN), which is expected to efficiently generate the topological object needed for the annealing process. Tracking performance with a different setup of the original algorithm is also studied with data collected by the ATLAS experiment.

1 Introduction

The Large Hadron Collider (LHC) is planned to be upgraded into the High Luminosity LHC (HL-LHC) between 2026 and 2028 and start taking data in 2029. The luminosity is expected to reach $L = 5 \times 10^{34}\text{cm}^{-2}\text{s}^{-2}$ at the HL-LHC, resulting in a large number of collisions (pile-up) per bunch crossing ($\langle\mu\rangle \sim 200$) and a high readout rate in the sub-detector systems [1]. In order to process such a huge amount of data, new techniques are required in the object reconstruction and the physics analysis. Track pattern recognition (tracking) is one of the challenging tasks in the large pile-up situation. This is because the current track reconstruction algorithm is based on a combinatorial algorithm and therefore the object reconstruction time will increase logarithmically with the number of pileup [2].

A tracking algorithm based on one of the quantum computing techniques, quantum annealing (QA), has been developed and it has been demonstrated that it can be used to perform the tracking with efficiency higher than 90% in the dense environment of an HL-LHC tracker [3]. In this algorithm, the track reconstruction starts by forming topological objects based on the spatial positions of activated sensors, called *hits*. There are 2 types of topological objects used in the algorithm: *Doublets* (D), a pair of hits; and *Triplets* (T), formed by three hits. Another type of object, *Quadruplet* (four hits), has been used to check the quality of triplets, but

*e-mail: wachan@icepp.s.u-tokyo.ac.jp

not directly used in the reconstruction. In order to perform the annealing process, these topological objects are used to construct a Quadratic Unconstrained Binary Optimization (QUBO) object, which is directly assigned to the qubit of the annealing machines.

In this paper, recent developments based on the tracking algorithm used in Ref.[3] are presented. The first study attempted to apply graph neural network (GNN) techniques in the QUBO formation. The second study applied the algorithm using a GPU-based annealing machine in a realistic detector environment.

2 Methodology

2.1 Track pattern recognition

The purpose of track pattern recognition is to recognise the correct combination of hits left by the charged particles. The consistency of hit positions and a possible trajectory is the key to tracking. The correct combination of hits must be selected while rejecting all the combinatorial random coincidences.

In the first study presented in this paper, the dataset prepared for the TrackML Particle Tracking Challenge [4], which was a dataset created based on the HL-LHC pileup environment, is used. Corresponding to the HL-LHC environment, the number of tracks per bunch crossing is about 10,000, and the total number of hits is about 100,000 per bunch crossing. In this dataset, only the hits recorded in the barrel layers of an LHC-like inner tracking detector are included. Only tracks with transverse-momentum (p_T) more than 1 GeV are retained.

In the second study, an ATLAS [6] dataset has been used to demonstrate annealing tracking in a realistic environment. The data was taken in 2017 by random triggers. Simulated samples were generated with the ATLAS detector simulation [7]. Only doublets with $|\eta| < 1.0$ are allowed in this study.

The tracking reconstruction relies on 5 parameters. The trajectory left by the charged particles is curved in the detector, which can be measured by $1/R$, where R is a Menger curvature. Other parameters are: polar angle (θ), azimuthal angle (ϕ), and the transverse and longitudinal distance from the collision point, represented by d_0 and z_0, respectively.

2.2 Hamiltonian

Quantum annealing is an optimization process to find the global energy minimum of a given Hamiltonian using quantum fluctuations [5]. In order to apply the annealing process in tracking, the Hamiltonian is implemented as a QUBO using topological objects like doublets or triplets. The Hamiltonian used in Ref.[3] can be defined as follows:

$$E = \sum_{i}^{N} \alpha_i T_i - \sum_{i,j} S_i j T_i T_j + \sum_{i,j} \zeta_i j T_i T_j, \tag{1}$$

where T_i is the triplet assigned to the qubit. The first term gives a penalty if the triplets act like a combinatorial fake, which is defined by the impact parameters of a triplet,

$$\alpha_i = \frac{1}{2}(1 - e^{-d_0/C_{d_0}}) + \frac{1}{5}(1 - e^{-z_0/C_{z_0}}), \tag{2}$$

where C_{d_0} and C_{z_0} are normalization constants controlling the effect of the penalty. In Ref.[3] the values are set to be $C_{d_0} = 1.0$ mm and $C_{z_0} = 0.5$ mm. The second term gives a reward if two triplets tend to belong to the same trajectory. This is determined from their curvature,

direction, and the presence of a missing hit (hole) in the triplet. The coefficient (S_{ij}) is defined as

$$S_{ij} = \frac{1 - \frac{1}{2}(P^R_{i,j} + P^\theta_{i,j})}{(1 + H_i + H_j)^2}, \quad \text{where } P^R_{i,j} = \frac{\left|(1/R)_i - (1/R)_j\right|}{C_R} \text{ and } P^\theta_{i,j} = \frac{max(\delta\theta_i, \delta\theta_j)}{C_\theta}, \tag{3}$$

where H_i is the number of holes in the i-th triplet, $(1/R)_i$ is the curvature of the i-th triplet, and $\delta\theta_i$ is the difference of polar angles between two doublets included in the i-th triplet. The C_R and C_θ are the normalization constants controlling how much inconsistency is allowed. In Ref.[3], $C_R = 0.1$ mm^{-1} and $C_\theta = 0.1$ radians are considered. The third term gives a penalty to a triplet pair if there are shared hits. A track having a shared hits tends to be a combinatorial fake, since it is unusual for charged particles to share the same hit. The coefficient $\zeta = 1$ if shared hits exists, otherwise $\zeta = 0$. Eq.1 can be written simply as:

$$O(a, b, T) = \sum_i^N a_i T_i + \sum_i^N \sum_{j<1}^N b_{ij} T_i T_j, \qquad b_{ij} = \begin{cases} -S_{ij} & \text{Pair of triplets align in sequence} \\ \zeta_{ij} & \text{Pair of triplets share a hit} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The coefficient a_i is called the *bias weight*, and the coefficient b_{ij} is called the *coupling strength*. These terms are collectively called the *QUBO strength*, which gives the bias to the initial state of the quantum system in the annealing process. The bias weight is the bias applied on the individual qubit; while the coupling strength gives a bias to the connection between 2 qubits. Fig.1 shows the distribution of the QUBO constructed using the algorithm in Ref.[3] with 2708 hits from a single event.

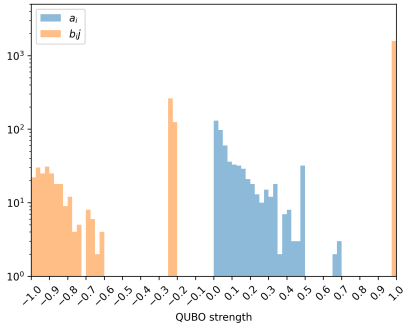


Figure 1: The QUBO strength distribution of the QUBO constructed using the algorithm in Ref.[3] with 2708 hits from a single event. The blue histogram represents the distribution of coefficient a_i in Eq.4 and the orange histogram represents the distribution of coefficient b_{ij} in Eq.4.

3 Application of graph neural network

The QUBO formation described in Eq.4 requires the pre-selection of triplets and triplet pairs. This section introduces a study focus on this pre-selection process using graph neural network (GNN). The study can be separated into 3 steps: 1) Graph generation, 2) GNN architecture construction, and 3) model training and performance test.

3.1 Graph generation

In the graph generation, the TrackML dataset is being used along with a target QUBO generated by the Ref.[3] algorithm. Hits being used in the target QUBO are first being extracted, and defined as *signal* hits. The *background* hits are then extracted by searching for the nearest hit to the signal hits along the same detector layer.

A graph is required to contain exact 4 hits and therefore 3 doublets. The Cartesian coordinates of the hit are being used as the node features. The doublet selection requirements described in Ref.[3] are being used as the edge features. A bi-directed graph has been considered in this study. Thus, 2 edges are being constructed between 2 hits, and the message is passing into opposite directions along 2 edges. Therefore, 1 graph contains 4 hits, which is embedding into 4 nodes, and 3 doublets, which is embedding into 6 edges. Two types of graph are being generated based on the number of signal hits: *Signal* graph and *background* graph. The signal graph only contains signal hits, and background graphs are constructed by anything number of signal and background hits. Therefore, a background graph can either contain both signal and background hits, or purely contain background hits. Eventually, 269550 graphs are generated, with 6% signal graphs and 94% of background graphs.

3.2 GNN architecture construction

The GNN architecture contains 2 fully connected graph modules, as shown in Fig.2. First, the module transforms the node and edge features into their latent representations, then performs message passing to update latent features, and finally computes edge classification scores. The final output is the edge score calculated in the second graph module (e_s^N).

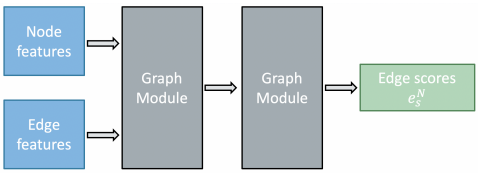
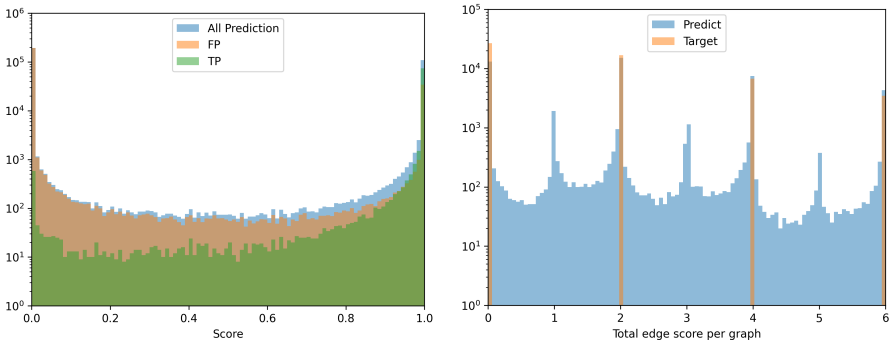


Figure 2: The Graph Neural Network architectures.

3.3 Results

The GNN is trained on an **NVIDIA V100** GPU for 50 epochs, resulting in the performance shown in Fig.3. The performance is examined by using the trained model with a test sample which contains 53910 graphs. In Fig.3 (a), the predicted score for each of the edges is shown. the "Predict:TP" ("Predict:FP") edges represent the edges that are true positive (false positive), which required a target edge score = 1 (0). In Fig.3(b), the sum of edge scores per graph is shown. The target score, showing that one can expect the graph to have a total edge score = $0, 2, 4, 6$. This is the result of doublet embedding as described in Sec.3.1. However, the distribution of predicted score suggested that the total edge score can be any integer from 0 and 6 for each graph. The difference between these distributions is originated from the leak of information about the doublet embedding in the network. Therefore, during the training, the score for each of the edges is calculated separately and being considered to be independent of the physical doublet. In the future studies, this information has to be passing into the network in order to have a physical result.



(a) Edge score per edges (b) Total edge score per graph

Figure 3: Distribution of edge classification score per (a) edges and (b) graph. The performance is examined by using the trained model with a test sample which contains 53910 graphs.

The coupling strength can be calculated using Eq.4. The distribution of $-S_{ij}$ for both target and GNN-generated QUBO has been shown in Fig.4. In the GNN-generated QUBO, triplet pairs belong to graph which have total edge score ≥ 4 are being selected. The target QUBO has 2 peaks around -0.95 and -0.25. Similar distribution can also be observed in the GNN-generated QUBO. However, most of the triplet pairs in the GNN-generated QUBO have coupling strength within the range $-0.25 < -S_{ij} \leq -0.85$. This is because 94% of the samples are background graphs. Thus, most of the triplet pairs in the GNN-generated QUBO are background-like. Although only graphs with total edge score ≥ 4 are being considiered, part of the GNN-generated QUBO is still constructed by the backgound-like triplet pairs. It is expected to have a more signal-like distribution by increasing the signal-to-background ratio during the graph generation stage.

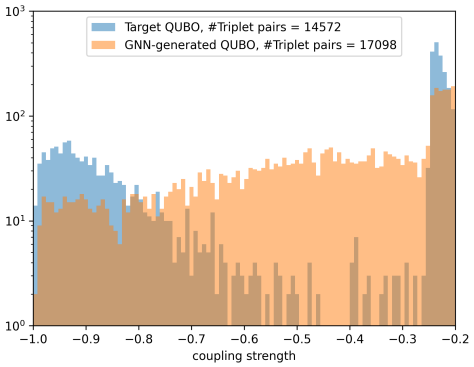


Figure 4: The coupling strength of the target (blue) and GNN-generated (orange) QUBO. The performance is examined by using the trained model with a test sample which contains 53910 graphs. In the GNN-generated QUBO, triplet pairs belong to graph which have total edge score ≥ 4 are being selected.

4 Application to the ATLAS dataset

The TrackML Particle Tracking Challenge [4] dataset simulates an ideal environment in HL-LHC, but the tracking algorithm used in Ref.[3] needed to be verified in the realistic environment.

In this section, the result of verification in ATLAS dataset is shown without applying any of the GNN techniques mentioned in the previous section. This study also modified the original algorithm used in Ref.[3], replacing the quantum annealer by the GPU-based annealing machine, called Fixstars Amplify AE (Annealing Engine)[8], which uses the **NVIDIA A100** GPUs and has 131k fully connected bits. Another modification is that the QUBO is constructed from doublets, instead of triplets. The Hamiltonian is therefore defined as:

$$H = \sum_{i}^{N} a_i D_i + \sum_{i}^{N} \sum_{j<i}^{N} W_{ij} D_i D_j + \sum_{i}^{N} \sum_{j<i}^{N} S_{ij} D_i D_j + \sum_{i}^{N} \sum_{j<i}^{N} \zeta_{ij} D_i D_j, \tag{5}$$

where D is the doublet. The first term is a penalty term which gives positive energy according to the number of holes in the doublets using the coefficient a_i, defined as:

$$a_i = C_1 (H_i + 1)^{C_2}, \tag{6}$$

where H_i is the number of holes in the i-th doublet. Constants C_1 and C_2 are set to be 1 and 2, respectively. The second term rewards cases where two doublets share a hit and form a triplet along the same direction. The energy decreases along with the differences in the curvature in the X-Y plane or the angle in the R-Z plane between the doublet pair, using the coefficient W_{ij}, defined as:

$$W_{ij} = -C_3 \{ C_4 e^{-[\frac{\Delta(1/R)}{C_R}]^{C_5}} + (1 - C_4) e^{-(\Delta\theta/C_\theta)^{C_5}} \}, \tag{7}$$

where constants C_R, C_θ, C_3, C_4 and C_5 are set to be 0.002, 0.1, 1.5, 0.5 and 1, respectively. The third term gives a penalty to doublet pairs which do not share a hit. If they have similar curvature in X-Y plane or similar angles in the R-Z plane, the coefficient S_{ij} is defined as:

$$S_{ij} = -C_6 \{ C_7 (1 - P_{ij}^{R\,C_8}) + (1 - C_7)(1 - P_{ij}^{\theta\,C_8}) \}, \tag{8}$$

where P_{ij}^R and P_{ij}^θ are the same as defined in Sec.1. Constants C_6, C_7 and C_8 are set to be 0.6, 0.5 and 2, respectively. The fourth term is a penalty term if a doublet pair that shares a hit and forms a V-shaped triplet. The coefficient ζ_{ij} is set to be 5.

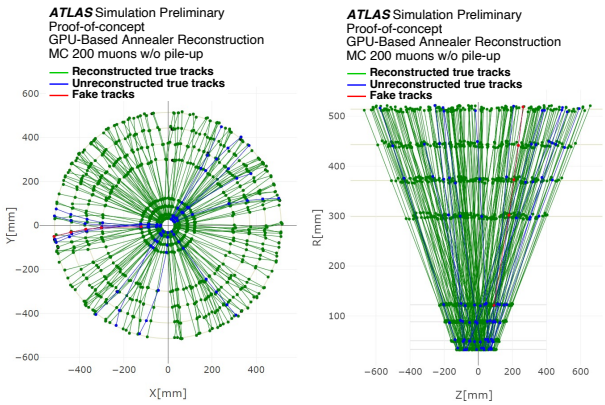
The size of QUBO is optimised by geometrically divided into 16 sub-QUBOs in the ϕ direction. Each pair of neighbouring sub-QUBOs has an overlapping region of 0.2 rad. Fig.5 shows a typical event display for 200 muons/event without a pileup MC sample generated with the ATLAS software [7]. These muons are generated with 0.5 GeV $< p_T <$ 10 GeV in $1/p_T$ flat distribution with $|\eta| < 1.0$. Since our current algorithm is applicable to only barrel regions, the η cut restricts generation to barrel tracks. The result indicates that track finding by annealing machines works successfully in a realistic environment.

The algorithm is then applied to real ATLAS data taken by non-physics random triggers with the ATLAS detector in runtime within 2017 with LHC fill 6371 and $\langle\mu\rangle = 21$. Fig.6a shows the reconstruction efficiency with respect to ATLAS offline tracks as a function of the track p_T. The matching to the ATLAS tracks is performed if tracks reconstructed with annealing machines share more than 50% of hits with the ATLAS tracks. The efficiency depends on track p_T and it is worst in the low p_T region. The potential efficiency loss might come from the forbidden patterns in the current algorithm, such as shared hits between tracks.

However, more investigations into the causes of inefficiency are needed. Considering this study as a first demonstration of technical feasibility, the result already looks promising.

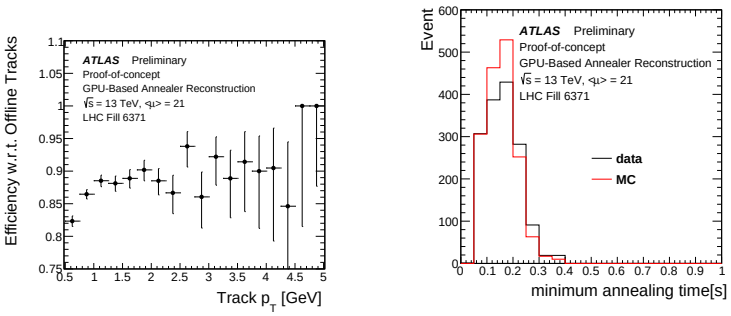
The annealing time, which is the time required to run one annealing, is also measured and shown in Fig.6b. This measurement is comparing ATLAS data and a MC sample with 10 pions/event, and $\langle\mu\rangle = 20$. For the data, the average preprocessing time with a single core of **11th Gen Intel(R) Core(TM) i9-11900K @ 3.50GHz** is about 0.6 seconds. An average QUBO size without slices for the dataset with $\langle\mu\rangle = 20$ is 109k bits.

The doublet-based QUBO typically needs 10 times longer annealing time than the triplet one, while the preprocessing time is 10 times faster.



(a) X-Y Plane (b) R-Z Plane

Figure 5: Event display of reconstructed tracks for 200 muons/event in (a) X-Y plane and (b) R-Z plane. Each point stands for a hit, and each line between points is a doublet. Reconstructed true tracks are green, blue shows unreconstructed true tracks, and fake tracks are represented by red. In (b) the 4 inner layers in the lower R region represents the pixel detectors, and the larger R region represent the SCT detectors.



(a) Efficiency (b) AnnealingTime

Figure 6: (a) Efficiency as a function of track p_T for the ATLAS data. (b) The minimum annealing time for data (black) and MC samples (red). The MC sample with 10 pions/event, and $\langle\mu\rangle = 20$ simulated with the ATLAS detector has been used.

5 Conclusion

In this paper, recent developments based on the tracking algorithm used in Ref.[3] are reported. The first study indicates that the implementation of GNN is a potential way to improve the pre-processing efficiency. The second study shows that the algorithm with doublet-based QUBO works well in a realistic environment, with both simulated ATLAS MC samples and real ATLAS data. Together, these studies demonstrate the feasibility of this approach, and improvements can be made in future developments.

Acknowledgements

T.K. is supported by the Center of Innovations for Sustainable Quantum AI (JST Grant Number JPMJPF2221).

References

[1] G. Apollinari, I. Béjar Alonso, O. Brüning, P. Fessia, M. Lamont, L. Rossi, L. Tavian, *High-Luminosity Large Hadron Collider (HL-LHC): Technical Design Report V. 0.1*, CERN Yellow Reports: Monographs (CERN, Geneva, 2017), https://cds.cern.ch/record/2284929

[2] *ATLAS Computing Public Result*, https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults

[3] M. Saito et al. *Quantum annealing algorithms for track pattern recognition*. EPJ Web Conf. **245**, 10006 (2020), doi:10.1051/epjconf/202024510006

[4] *TrackML Particle Tracking Challenge*, https://www.kaggle.com/c/trackml-particle-identification

[5] S. Morita, H. Nishimori, *Mathematical foundation of quantum annealing*. J. Math. Phys. **49** (12): 125210 (2008), https://doi.org/10.1063/1.2995837

[6] The ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, JINST **3**, S08003 (2008), https://dx.doi.org/10.1088/1748-0221/3/08/S08003

[7] The ATLAS Collaboration, *The ATLAS Collaboration Software and Firmware*, ATL-SOFT-PUB-2021-001, https://cds.cern.ch/record/2767187

[8] *FIXSTARS Amplify AE*, https://amplify.fixstars.com/en/docs/amplify-ae/about.html