

Improving ATLAS Muon Segment Regressor with Deep Learning

Tomoya Tanaka

Supervisor: Davide Di Croce, Johannes Junggeburth

Summary

As part of the CERN Summer Student Programme 2025, I worked on improving the ATLAS muon segment regressor using deep learning. The High Luminosity Large Hadron Collider will increase the pile-up from 60 to 200, requiring upgrades to the ATLAS detector and the Trigger and Data Acquisition system. The muon segment reconstruction is one of the steps in the Event Filter, and it faces several challenges, such as background noise and ambiguities. To deal with these challenges, we used graph neural networks for reconstructing the muon segments. During the programme, I developed and evaluated a one-segment regressor and a two-segment regressor, and worked on the optimization of the hyperparameters of the model. As a result, the regressor models predicted the true segment parameters with high accuracy, with all the errors mostly less than 2%.

1 Introduction

The Large Hadron Collider is a particle accelerator for proton-proton collisions at $s = 13.6$ TeV. One of the experiments at the LHC is the ATLAS (A Toroidal LHC ApparatuS) experiment, which uses the ATLAS detector to record particles produced in the collision. The LHC is scheduled to be upgraded to the High-Luminosity LHC (HL-LHC) in 2030. The HL-LHC will increase its peak luminosity by a factor of 3.7 and increase the number of proton collisions per bunch crossing (pile-up) from 60 to 200. To record the data in such high pile-up conditions, upgrades are required for the ATLAS detector and its Trigger and Data Acquisition (TDAQ) system.

In the ATLAS experiment, TDAQ is mainly divided into two components: the hardware trigger and the software trigger. For the Phase-2 upgrades, TDAQ will be upgraded completely (see figure 3). The hardware trigger called the Level-1 trigger will be replaced by the Level-0 trigger (L0), and the decision rate will be improved from 100 kHz to 1 MHz. The software trigger will also be upgraded from the High Level Trigger to the

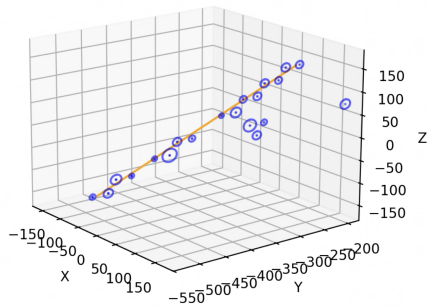


Figure 1: An example of one segment fitting. The orange line is the segment, and the blue dots are the hits of the MS. The blue rings around the hits show their drift radii. In this figure, hits that are not tangent to the segment are background noise.

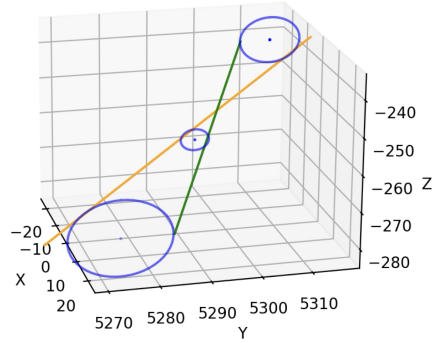


Figure 2: An example of the left-right ambiguity. The orange line and the green line are segments, and both of them are tangent to the hits.

Event Filter (EF), which will collect the data at 10 kHz, five times higher than the Run 3 output [4]. Because the EF will make use of offline-like algorithms for reconstruction, a lot of research is underway to reduce the processing time and to enable algorithms to run on various platforms, such as CPU, GPU, and FPGA. One of the powerful approaches is machine learning (ML).

To accomplish these goals, I worked on improving the muon-segment parameter regressor using a Graph Neural Network (GNN). A muon segment is a trajectory of a muon, approximated as a straight line, that is reconstructed from hits of the Muon Spectrometer (MS).

In the muon segment reconstruction process, there are mainly two challenges. The first one is a background hit. Among the MS hits are genuine signal hits from muons, as well as background hits originating from hadronic punch-through and gamma radiation. In figure 1, there are several background hits that are not tangent to the segment. The second one is the segment indeterminacy. Figure 2 is an example of it. In the figure, the blue rings represent the drift radii of MDT hits. Segments are reconstructed so that they are tangent to the rings. There are two segments in the figure, and both meet this condition. However, only one of them corresponds to the true segment, and it is difficult to distinguish which one is correct. This ambiguity is referred to as the left-right ambiguity.

I worked on these difficulties using GNNs. GNN is one of the models of deep learning, and is specialized for dealing with graph structure data. Because the MS hits and the muon segments can be regarded as a graph structure, GNN is well-suited to the muon segment reconstruction.

This model enables segment reconstruction while effectively rejecting background noise. But also reduce the number of seed candidates by providing fewer segments to the final track fitter, in contrast to the classical segment fitter which supplies multiple candidates. This means that the segments generated by this model do not contain the fake segments that do not consist of the final tracks, while the classical segment fitter contains

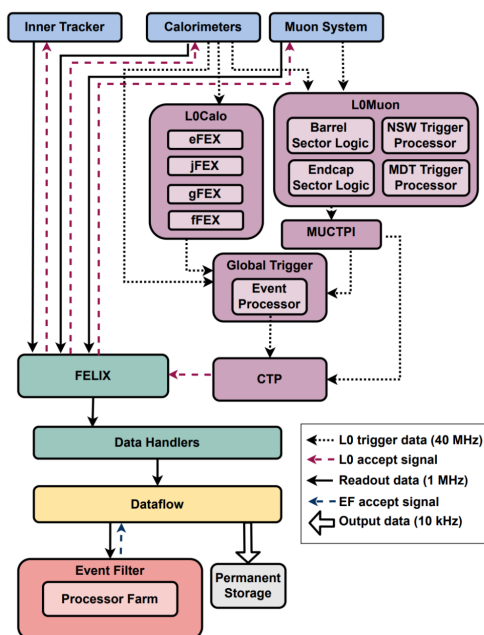


Figure 3: The ATLAS TDAQ architecture for Phase-II. The purple boxes are the Level-0 hardware trigger, and the orange box is the Event Filter [3].

them as candidate. By providing fewer seed candidates, the time required to find final tracks is expected to be reduced. Furthermore, GNNs offer a solution to the left-right ambiguity. Although the segments consist of this ambiguity cannot be distinguished by the classical segment fitter, GNN models can identify only the true segment.

During the program, I developed and evaluated a one-segment regressor and a two-segment regressor. Both predict segments with low errors with values almost less than 2%.

2 Muon Segment Reconstruction

2.1 Muon Spectrometer

The Muon Spectrometer [1, 2] is the outermost sub-detector of ATLAS. It consists of three air-core superconducting toroidal magnets, which provide a field of approximately 0.5 T. The bending of a muon trajectory in the magnetic field is measured by three layers of monitored drift tubes (MDTs), which provide precision tracking by measuring the muon's position projected on the magnetic bending plane. In the region of high pseudorapidity near the interaction point, Micromegas (Micro-Mesh Gaseous Structure, MM) chambers are used. MDTs consist of drift tubes, which have wires in their center and are filled with a gas mixture. A high voltage is applied between the wires and the tube walls. When a muon approaches the wire, an electron avalanche occurs. By measuring the drift time, the position of the muon can be determined precisely. For the trigger tracking, three layers of resistive plate chambers (RPCs) in the barrel region and the three or four layers of

thin gap chambers (TGCs) in the end-cap region are used. These detectors provide the measurements of the muon’s position in a direction orthogonal to the coordinate recorded by the MDT or the MM.

The momentum resolution of the MS is roughly $\Delta p_T/p_T = 4\%$ for high p_T , and increases to 11% for $p_T = 1$ TeV. Here p_T denotes the transverse momentum, which is perpendicular to the beam axis. The alignment accuracy is 30–40 μm for high p_T tracks, and the single hit resolution is 80–90 μm .

2.2 Muon Track Reconstruction Algorithm

There are four steps to reconstruct a trajectory of a muon.

1. Bucket finding

As the first step of the reconstruction, collections of hits called buckets are created. Each bucket has its local coordinate frame; The x-axis aligns with the longitudinal direction of the strip detectors, the y-axis spans the readout channel dimension, and the z-axis is defined to pass through the detector layers. The bucket formation is based on the spatial density of the hits. When a density exceeds a predefined threshold, a bucket is created under the constraint that its absolute length along the y-axis is less than two meters.

2. Bucket filtering

After making muon buckets, the Bucket Filter is applied to them to classify the number of segments. It is also developed using machine learning.

3. Segment Finding

A muon segment is an approximate straight line of a muon trajectory (see Figure 1). With the non-ML algorithms, segments are found using straight-line fitting, resulting in multiple candidates. They contain some segments that are not true segments, which cannot construct final tracks. In addition to this, they also include the left-right ambiguity. In our new method, segments are identified using the Muon Segment Regressor, which is the deep learning model to find muon segments within a bucket. There are two types of segment regression models: a one-segment regressor and a two-segment regressor. Each regressor identifies a given number of segments within a single bucket. The choice of model is determined by the results of bucket filtering. One segment is determined by four variables: x and y coordinates for showing its local position in the $z = 0$ plane, and θ and ϕ values for specifying its local direction.

4. Track Finding

Identified segments are used for finding the trajectories of muons. After this process, the trigger decision is made by these muon tracks.

3 Performance Evaluation of the One Muon Segment Regressor

The one-segment regressor is the GNN model that identifies a single segment within a single bucket. I evaluated its standalone performance.

3.1 Data Set

For the dataset, we used the Monte Carlo (MC) data set of $t\bar{t}$, J/Ψ (1S), and $Z \rightarrow \mu\bar{\mu}$, and we required that the buckets have only one true segment. The true segment refer to the muon segments generated directly from the MC data, and it was used for training to prevent the model from learning left-right ambiguities. For each hit, six variables in Table 1 were used as input variables, and four variables in Table 2 were regressed as the segment parameters. As pre-processing, both input values and labels were normalized, and each bucket was converted to a fully connected graph structure, in which nodes correspond to hits.

Table 1: Input variables.

Name	Description
<code>localX</code>	Local x coordinate of a hit
<code>localY</code>	Local y coordinate of a hit
<code>localZ</code>	Local z coordinate of a hit
<code>stationIndex</code>	Index of the station
<code>driftR</code>	Drift radius of MDT
<code>relative_layer</code>	The number of the layer a hit is in, normalized by the number of all layers in the bucket

Table 2: Output variables.

Name	Description
<code>segment_localX</code>	Local x coordinate of a segment
<code>segment_localY</code>	Local y coordinate of a segment
<code>segment_localTheta</code>	Local θ value of a segment
<code>segment_localPhi</code>	Local ϕ value of a segment

3.2 Model and Training

For the one-segment regressor, we used the Edge Convolution Model. The architecture of it is shown in figure 4. In this model, the features of adjacency nodes are aggregated by the specified aggregation function. We chose to take the average of them as the method of aggregation. This model has two Edge Convolution layers in the residual blocks, which add input values to output values. The loss function of this model is the Mean Squared Error (MSE), and the optimizer is the AdamW. If the validation loss does not improve

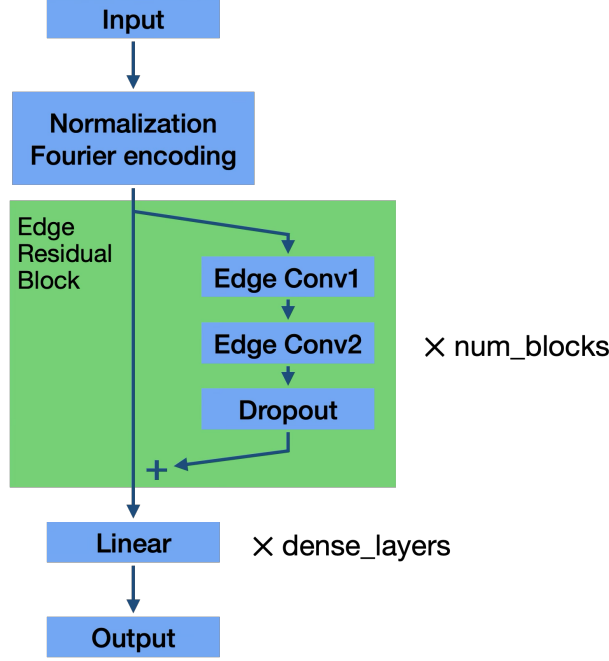


Figure 4: The architecture of the Edge Convolution model.

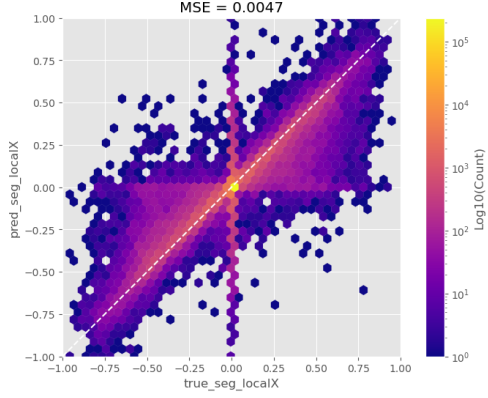
after 5 consecutive epochs, the learning rate is reduced by half, and early stopping is applied if no improvement in the validation is observed after 20 consecutive epochs.

3.3 Standalone Performance Evaluation

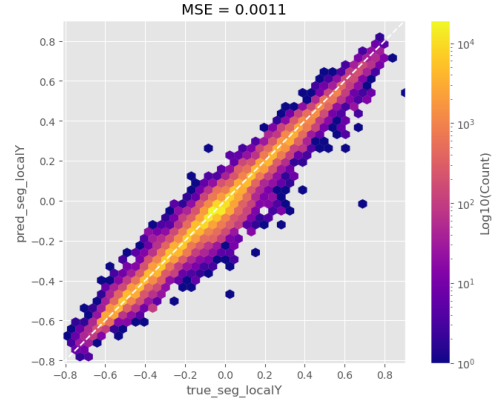
The ATLAS software framework (ATHENA) is used for several purposes, e.g., reconstruction, event generation and simulation. To implement our GNN model, we leverage the C++ ONNX (Open Neural Network Exchange) API, an open library for machine learning, which is available within ATHENA. Therefore, we performed accuracy and time benchmark of the GNN ONNX model.

The results are shown in Figure 5. They represent the errors between the true segment parameters (the x -axis) and the predicted segment parameters (the y -axis). The errors are listed in Table 3. The errors normalized by range show the error rates. The ranges of x , y , and ϕ are normalized from -1 to 1, so error rates are given by MSEs divided by 2. For θ , its range is normalized from 0 to 1, so its MSE directly represents its error rate. Regarding the y coordinates, the predicted values closely matched the true values, with 0.055% error. Although a distribution was relatively broad, the θ values also showed the same pattern as the true values. While the graph of the x -coordinate showed very small MSE, the predicted values are clustered at `true_segmentX = 0`. This can be explained by the fact that if all the hits in a bucket are from MDT, they do not measure x/ϕ , yet the model attempted to estimate the x coordinate. A similar thing happened in the ϕ graph. If all the hits are at $x = 0$, the ϕ values are $\pm\pi/2$, so they become $\pm 1/2$ by normalization. Overall, the one-segment regressor predicted true segments accurately with errors less than 0.3% for x , y and ϕ , and approximately 2% for θ .

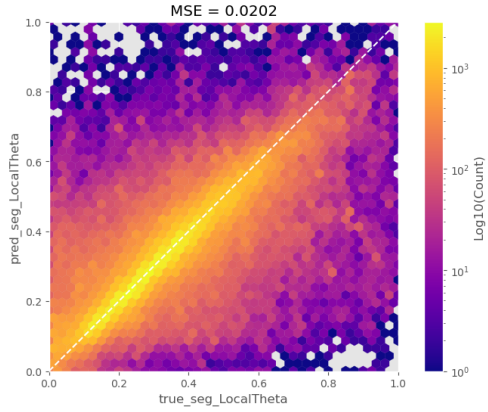
Regarding inference speed, we measured the standalone inference time on an NVIDIA



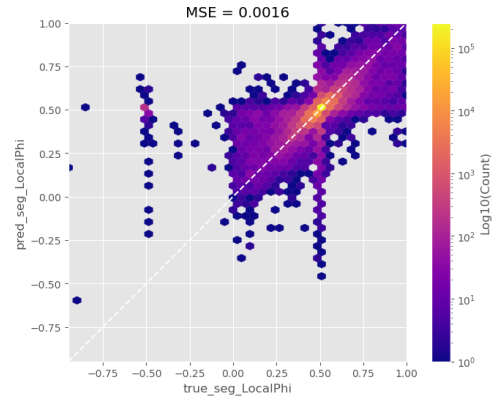
(a) The errors of `segment_localX`



(b) The errors of `segment_localY`



(c) The errors of `segment_localTheta`



(d) The errors of `segment_localPhi`

Figure 5: The errors of the one-segment regressor between the true segment parameters and the predicted segment parameters. The scores on the figures are mean squared errors (MSE).

Table 3: Errors of the one-segment regressor.

Variable	MSE	Error normalized by range
x	0.0047	0.24%
y	0.0011	0.06%
θ	0.0202	2.02%
ϕ	0.0016	0.08%

H100 NVL GPU using the model converted to ONNX. The batch size was 128, and inference was run in parallel. The time was 319 ms/batch, which corresponds to 2.5 ms/bucket.

4 Two Muon Segment Regressor

The two-muon segment regressor is similar to the one-segment regressor, except it identifies two segments within a single bucket. It will be used for the buckets that are classified as having two segments by the Bucket Filter.

4.1 Dataset, Model and Training

The model of this regressor is the same as the one-segment regressor, but it predicts 8 parameters: 4 parameters of the first segment and 4 parameters of the second segment. These segments are ordered by comparing the absolute values of their y coordinates, with the segment having the smaller absolute y coordinate placed first. Regarding the dataset, we used true segments for the training because they do not contain left-right ambiguities. When using true segments, we required that a bucket contained two true segments.

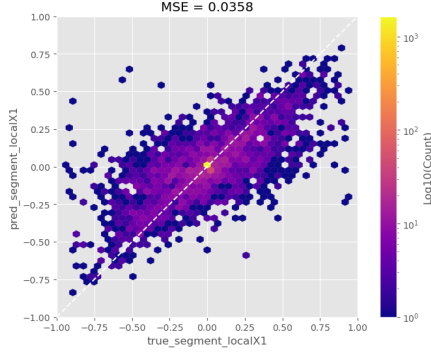
Training reached convergence at 93 epochs, and it took about 314 minutes.

4.2 Results

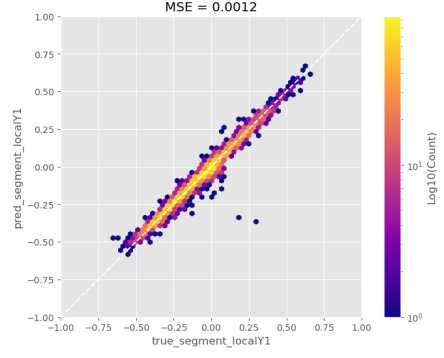
The errors of the regressor are illustrated in figure 6 and are listed in Table 4. These graphs were generated from the model that was not converted to ONNX. The top four graphs are the parameters of the first segment, and the bottom four graphs are the parameters of the second segment. Both of them had similar distributions. The y graphs and θ graphs showed good prediction performance of this model, with errors less than 0.3%, and the x graphs and the ϕ graphs had broad distributions similar to the one-segment model. Regarding the MSEs, they were very small, so this model could identify two segments in a single bucket accurately, with all of the errors less than 2%.

5 Conclusion

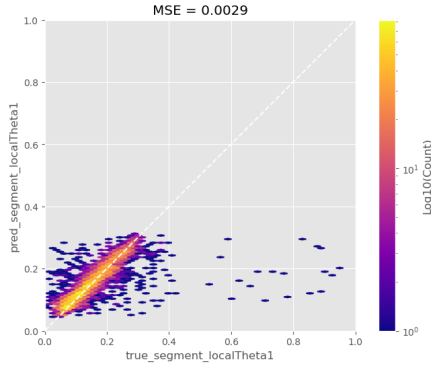
In this program, I developed and evaluated the performance of the one muon segment regressor and the two-segment regressor, and performed hyperparameter optimization. Both of the models showed that they could predict segments with high precision.



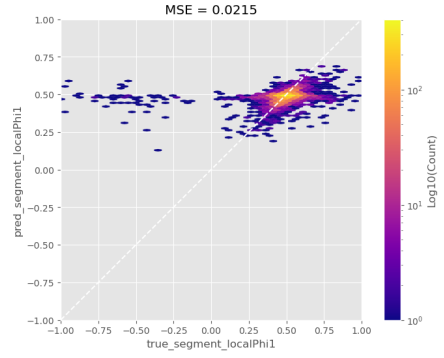
(a) The errors of `segment_localX1`



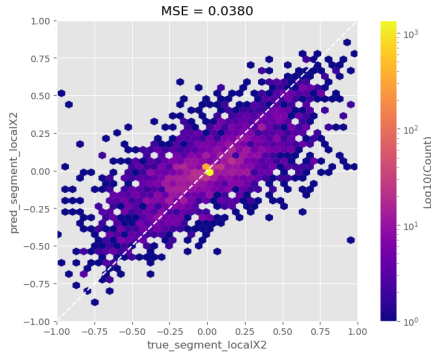
(b) The errors of `segment_localY1`



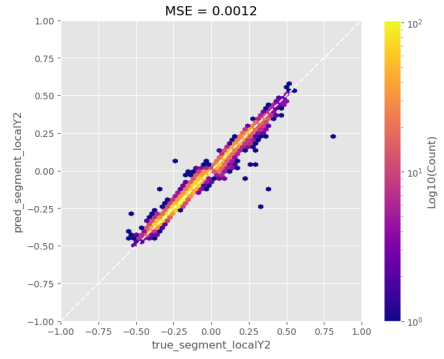
(c) The errors of `segment_localTheta1`



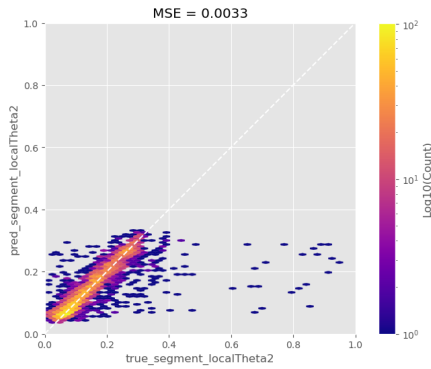
(d) The errors of `segment_localPhi1`



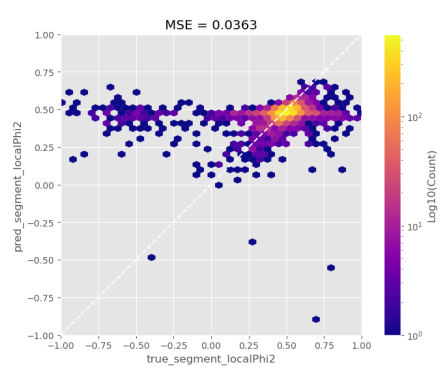
(e) The errors of `segment_localX2`



(f) The errors of `segment_localY2`



(g) The errors of `segment_localTheta2`



(h) The errors of `segment_localPhi2`

Figure 6: The errors of the two-segment regressor between the true segment parameters and the predicted segment parameters.

Table 4: Errors of the two-segment regressor.

Variable	MSE	Error normalized by range
x_1	0.0358	1.79%
y_1	0.0012	0.06%
θ_1	0.0029	0.29%
ϕ_1	0.0215	1.08%
x_2	0.0380	1.90%
y_2	0.0012	0.06%
θ_2	0.0033	0.17%
ϕ_2	0.0363	1.82%

As next steps, we have several tasks to do: evaluation of the inference performance within ATHENA, conversion of the two-segment regressor to ONNX for performance evaluation and implementation of the inference to ATHENA, and retraining of the one-segment regressor with the optimal hyperparameters.

6 Acknowledgment

I would like to express my appreciation to my supervisors, Davide Di Croce and Johannes Junggeburch, for their continuous support and guidance during the program. I could learn a lot of things about deep learning and physics throughout this project. I would not accomplish it without their proper advice.

References

- [1] *ATLAS muon spectrometer*. Technical design report. ATLAS. CERN, Geneva, 1997. URL: <https://cds.cern.ch/record/331068>.
- [2] Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer. Technical report, CERN, Geneva, 2017. URL: <https://cds.cern.ch/record/2285580>.
- [3] Collaboration ATLAS. Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - Event Filter Tracking Amendment. Technical report, CERN, Geneva, 2022. URL: <https://cds.cern.ch/record/2802799>, doi:10.17181/CERN.ZK85.5TDL.
- [4] Francesca Pastore. ATLAS TDAQ upgrades for Phase-2. Technical report, CERN, Geneva, 2024. URL: <https://cds.cern.ch/record/2907275>.