EPJ Web of Conferences **214**, 06038 (2019) https://doi.org/10.1051/epjconf/201921406038
CHEP 2018

Deep Learning Approach to Track Reconstruction in the upgraded VELO

Kurt Rinnert[1,*] and *Marco* Cristoforetti[2,**]

[1]University of Liverpool, Liverpool, United Kingdom
[2]FBK, Trento, Italy

Abstract. The LHCb experiment will undergo a major upgrade for LHC Run III, scheduled to start taking data in 2021. In order to exploit the full physics potential of the upgraded detector, LHCb will employ a high level event filter entirely implemented in software. The event filter has to operate in real time at the 40 MHz LHC bunch crossing rate. The LHCb collaboration is currently exploring a variety of new new computing paradigms in order to cope with the challenges posed by the high data taking rates.

One contribution to this effort is the application of Machine Learning (ML) techniques to particle track reconstruction.

We explore an ML approach to the track reconstruction in the upgraded LHCb Vertex Locator (VELO). The reconstruction algorithm is evaluated with fully simulated Minimum-Bias data that reflects the input to the real time event filter. We have achieved a reasonable track reconstruction efficiency and low ghost rate with our first approach and are currently exploring several methods to further quality of the track reconstruction.

1 Introduction

The LHCb experiment will undergo a major upgrade for LHC Run III, scheduled to start taking data in 2021 [1]. The upgrade of the LHCb detector introduces a radically new data-taking strategy: the current multi-level event filter will be replaced by a trigger-less readout system, feeding data into a software event filter at a rate of 40 MHz.

The entire LHCb tracking system will be replaced by upgraded detector components. In particular, the current VELO, a silicon strip detector surrounding the interaction region, will be replaced by a hybrid pixel system [2].

The main purpose of the VELO is to provide high-precision primary and decay vertex measurements. The track and vertex reconstruction in the VELO is the first step in the software reconstruction chain. The particle tracks and decay vertices provided by the VELO provide a large fraction of the background suppression power of the software event filter. Full event reconstruction at a rate of 40 MHz is a big challenge, given ever limited computing resources. It is therefore important to explore novel approaches in order not to limit the experiment's physics potential due to algorithmic shortcomings.

---

[*] e-mail: kurt.rinnert@cern.ch
[**] e-mail: mcristofo@fbk.eu

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

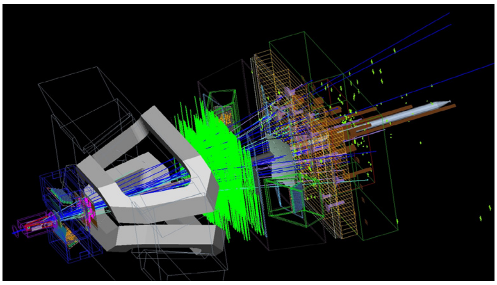

Figure 1: LHCb Run II event display

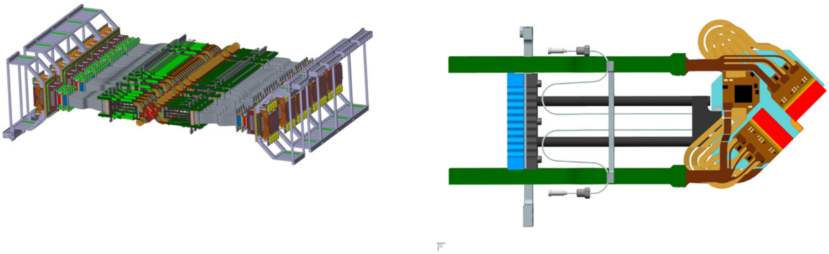

Figure 2: VELO upgrade detector (left) and single module (right)

Recent developments in Machine Learning (ML) and parallel processing enable radically new approaches to event reconstruction. We have obtained promising results from the application of ML techniques to track reconstruction in the VELO.

2 The LHCb VELO

Figure 1 shows a computer rendering of the LHCb experiment at CERN. The lines going from the lower left to the upper right represent the paths of particles passing through the various sub-components of the LHCb detector. The particles are produced by proton-proton collisions of the LHC collider. The VELO is located at the collision point in the lower left.

Figure 2 (left) shows a rendering of the upgraded Vertex Locator (VELO) component of the LHCb experiment. The VELO surrounds the proton-proton interaction region and is designed to provide extremely high-precision vertex measurements. The VELO is comprised of two movable halves that close around the interaction region at the beginning of each LHC fill. Each half consists of 26 modules with four pixel sensors arranged in an L-shape. Figure 2 (right) shows a single VELO upgrade module. The pixel sensors matrix features 768 x 256 square pixels with a pixel size of 55 μm. The distance of the innermost pixels to the LHC beams is only 5 mm. This extreme proximity and the small pixel size result in an excellent impact parameter resolution of 20 μm for high momentum tracks.

3 Track Reconstruction

The VELO records charged particles from proton-proton collision events at a rate of 40 MHz. At the raw data level each event is the collection of the addresses of all pixels that recorded a signal. Track reconstruction is then the task of associating pixels that belong to unique particle trajectories.

The conventional base line algorithm starts by grouping neighbouring pixels on all sensors into clusters. The local cluster positions are then transformed to the global coordinate system, yielding a collection of space points.

The algorithm then proceeds by creating doublets of space points. Each doublet is a seed for a particle track hypothesis. These track seeds are then propagated through detector planes to add space points to the track hypothesis.

Building and pursuing all possible doublets in a sequential algorithm is too time consuming. Several techniques are employed to mitigate this problem: doublets in the regions farthest from the interaction point are prioritised and track hypothesis are required to point towards the interaction region.

The base line algorithm performs well, both in speed and quality of the results. However it has many tunable parameters, like search windows, which makes it hard to find the optimal working point. It also does not lend itself well to parallel architectures like GPU accelerators. We therefore explore novel approaches to the track reconstruction in the VELO.

4 Machine Learning in Track Reconstruction

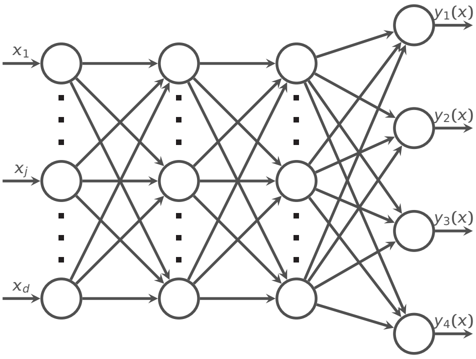

Figure 3: Neural network architecture

The ML approach we explore employs a neural network to reduce the track seed combinatorics. The network (TrackNN) takes (r, ϕ) coordinates from two detector planes as input and produces the probabilities that a pair of coordinates define a true track segment. The z coordinate is not explicitly used as it is fixed for each detector plane (barring small misalignments that do not affect the results).

The TrackNN network is a fully connected neural network (FCNN) with five layers. Figure 3 shows a schematic of an FCNN. The FCNN acts as a a classifier that computes the probabilities for N target (r, ϕ) coordinates to to form a true track segment with a given seed (r, ϕ) coordinate on an adjacent detector layer. The input of the first layer consists of $N + 1$ (r, ϕ) coordinates. The first coordinate pair, (r_s, ϕ_s), is the seed space point. The following N (r_i, ϕ_i) coordinates are candidate coordinates on a detector plane adjacent to the seed plane. Hence the input layer has a width of $2 \cdot (N + 1)$. The input layer is followed by three hidden layers with a width of $32 \cdot N + 1$, $32 \cdot N + 2$ and $(32 \cdot N + 2)/2$, respectively. The non-linear activation function for these layers is Relu. The width of the output layer is N. The output layer activation function is a sigmoid. This allows to interpret the values in the output layer as the probabilities p_i that (r_i, ϕ_i) and (r_s, ϕ_s) form a true track segment.

The network is trained on fully simulated minimum-bias events using the Monte Carlo truth information. The simulated data is a realistic representation of events the detector will record. We use the Adam optimizer for the training. We train a network for each pair of adjacent detector planes. The quality of the training is monitored automatically and the training stops when the test loss does no longer decrease for ten consecutive epochs. The network is implemented and trained using the Pytorch framework [3].

Track reconstruction with the TrackNN network proceeds as follows. A track is defined as a collection of space points from distinct detector planes. At the beginning of the event reconstruction, all space points in the detector plane furthest from the interaction region are considered seeds. Then the TrackNN is used to determine the connection probabilities for all candidate points on the adjacent plane towards the interaction region. Only the space point with the highest probability as predicted by TrackNN is considered. Furthermore, the probability has to be higher than a configurable cut. For the results presented in Section 5, the cut was set to 0.4. The above procedure is then repeated to propagate a track back to the interaction region. The propagation stops when the highest connection probability does not pass the cut. No further assumptions are made. We do, however, have plans to exploit *a-priory* knowledge about the geometry and the physics to improve the algorithm's performance.

5 Results

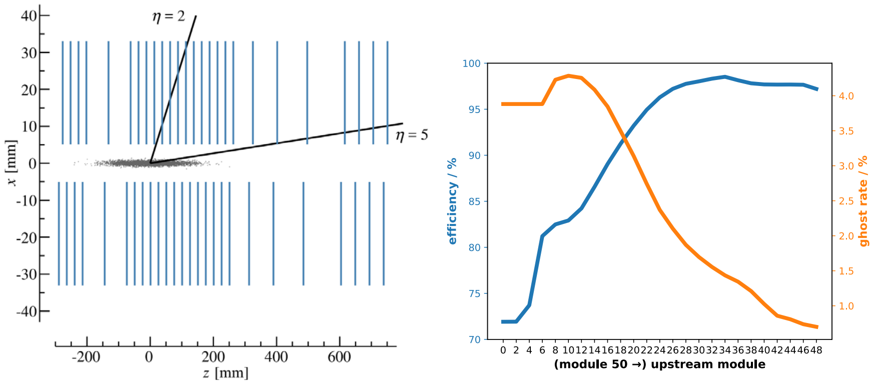

Figure 4: VELO planes and interaction region (left) and reconstruction quality (right)

We compare the performance of the TrackNN algorithm to the conventional base line evaluated in [2]. The results presented here a preliminary. The algorithm was only applied to one detector half as it was understood it would not cope well with overlapping modules from both halves. The base line evaluation uses the following definition for a track that is considered reconstructible:

- at least three clusters recorded by the VELO
- particle momentum > 5 GeV
- pseudo rapidity in the range $2 < \eta < 5$

Table 1: Performance Comparison

algorithm	**eff%**	**ghost%**	**clone%**
conventional ($p > 5$ GeV etc.)	98.9	2.5	1.0
TrackNN, module 50 → 30	98.0	1.6	0.8
TrackNN, module 50 → 20	93.2	3.1	0.9
TrackNN, module 50 → 0	71.1	3.9	0.7

- separation from the beam line in the xy plane less than 1 mm
- particle originates from the interaction region
- a track is considered reconstructed if at least 70% of space points match a single true particle

The labeled data used for training and evaluation of the TrackNN lacked some of the Monte Carlo truth necessary to fully replicate the efficiency denominator from [2]. The bias introduced by this is in favour of the TrackNN algorithm, because the efficiency denominator is less restricted. A "ghost" is defined as a track that cannot be associated to a particle because less than 50% of the space points come from a single true particle track. A "clone" is a track that has been reconstructed more than once.

The data set used for evaluation is fully simulated minimum bias data with $\nu = 7.6$, reflecting the input to the software event filter for LHC Run III. Table 1 shows the algorithm's performance for different scenarios. The scenarios are defined by how far towards and beyond the interaction regions track candidates are propagated. For instance, 50 → 30 designates propagation from the most downstream detector plane (50) towards the interaction point, stopping 20 planes in at plane 30. The closer the propagation gets to the interaction region, the harder it gets to reliably create correct track segments, which is reflected in Table 1. The table and Figure 4 show that the algorithm's performance degrades in the interaction region. As mentioned in Section 4, we anticipate ways to mitigate this.

6 Conlusions

We have obtained promising results applying ML techniques to charged particle track reconstruction in the upgraded LHCb VELO. An FCNN determining plane-to-plane space point connection probabilities has been shown to perform well. The nature of the algorithm lends itself well to implementations on massively parallel architectures, like GPU accelerators. It is likely that exploiting more *a-priori* available information will further improve the performance of the algorithm.

References

[1] LHCb Collaboration, Framework TDR for the LHCb Upgrade: Technical Design Report, CERN-LHCC-2012-007. LHCb-TDR-12 (2012)

[2] LHCb Collaboration, LHCb VELO Upgrade Technical Design Report, CERN-LHCC-2013-021. LHCB-TDR-013 (2013)

[3] https://pytorch.org