Calorimeter Reconstruction Innovations for the LHCb Experiment

Núria Valls Canudas

UNIVERSITAT RAMON LLULL

DOCTORAL THESIS

Title	Calorimeter Reconstruction Innovations for the LHCb Experiment
Presented by	Núria Valls Canudas
Centre	La Salle Digital Engineering School
Department	Departament d'Enginyeria
Directed by	Dr. Xavier Vilasís Cardona and Dra. Míriam Calvo Gómez

Per tu i la teva ciència, Avi.

Abstract

Calorimeter Reconstruction Innovations for the LHCb Experiment

by

Núria Valls Canudas

Universitat Ramon Llull, La Salle Campus Barcelona

This thesis focuses on software contributions to the LHCb experiment at CERN, specifically for the Calorimeter system in LHCb Upgrade I context. The main contributions concern the study of alternative algorithms for calorimeter data reconstruction. The first approach employs a segmented deep learning technique, breaking down the reconstruction problem into steps learned by small convolutional neural networks. Although promising, it lacked an efficient inference engine inside the LHCb framework. The second approach presents a graph-based clustering algorithm, showing equivalent cluster resolution to the LHCb's existing method, but with higher efficiency and significantly improved execution time, which is now the default solution for calorimeter reconstruction in the Run 3 period. This work also comprises a first approach to improve the current calorimeter reconstruction algorithm in the GPU Allen framework for HLT1, while the other approaches aim to the CPU reconstruction sequence in HLT2.

Additionally, the thesis addresses part of the calorimeter commissioning for Run 3, detailing the time alignment task for the Electromagnetic and Hadronic calorimeters. This involves the adaptation of the previous method to new electronics, gathering and analyzing data, and providing fine time alignment for the around $10,000$ calorimeter channels within 1 ns precision.

Resum

Calorimeter Reconstruction Innovations for the LHCb Experiment

by

Núria Valls Canudas

Universitat Ramon Llull, La Salle Campus Barcelona

Aquesta tesi se centra en contribucions de software per l'experiment LHCb al CERN, específicament pel sistema de calorímetres en el context de l'anomenat Upgrade I. Les contribucions principals se centren en l'estudi d'algoritmes alternatius per la reconstrucció de dades del calorímetre d'LHCb. En el primer enfocament, s'utilitza una tècnica d'aprenentatge profund segmentat, basada en descompondre el problema de reconstrucció en passos que són apresos per xarxes neuronals convolucionals petites. Tot i que els resultats són prometedors, el mètode manca d'un motor d'inferència eficient dins del marc de software d'LHCb. En el segon enfocament, es presenta un algoritme de reconstrucció basat en grafs, que presenta una resolució dels clústers reconstruits equivalent al mètode usat a l'expeiment però amb una eficiència més alta i un temps d'execució significativament millorat. Aquesta proposta ha passat a ser la solució predeterminada per la reconstrucció del calorímetre durant el període de presa de dades actual anomenat Run 3. A més a més, aquesta tesis també inclou una primera proposta per millorar l'algoritme actual de reconstrucció del calorímetre en el marc del sistema de *trigger* en GPU, anomenat Allen, mentres que les dues propostes anteriors estan dissenyats per la seqüència de reconstrucció del *trigger* en CPU, anomenat HLT2.

D'altra banda, la tesi aborda part de la posada en marxa del calorímetre pel Run 3, detallant la tasca de *time alignment* per al calorímetre Electromagnètic i l'Hadrònic. El que implica l'adaptació del mètode utilitzat anteriorment a la nova electrònica, la recopilació i l'anàlisi de dades, i donar un alineament temporal als aproximadament $10,000$ canals dels calorímetres amb una precisió d'1 ns.

Resumen

Calorimeter Reconstruction Innovations for the LHCb Experiment

by

Núria Valls Canudas

Universitat Ramon Llull, La Salle Campus Barcelona

Esta tesis se centra en contribuciones de software para el experimento LHCb en el CERN, específicamente para el sistema de calorímetros en el contexto del llamado Upgrade I. Las contribuciones principales se centran en el estudio de algoritmos alternativos para la reconstrucción de datos de los calorímetros. El primer enfoque utiliza una técnica de aprendizaje profundo segmentado, descomponiendo el problema de reconstrucción en pasos que son aprendidos por pequeñas redes neuronales convolucionales. Aunque los resultados son prometedores, el método carecía de un motor de inferencia eficiente dentro del marco de software de LHCb. El segundo enfoque presenta un algoritmo de reconstrucción basado en grafos, con una resolución de los clústeres reconstruidos equivalente al método existente en LHCb, pero con una mayor eficiencia y un tiempo de ejecución significativamente mejorado. Esta propuesta ha pasado a ser la solución predeterminada para la reconstrucción del calorímetro durante el período de toma de datos actual llamado Run 3. Además, esta tesis también incluye una primera propuesta para mejorar el algoritmo actual de reconstrucción del calorímetro en el marco del sistema de *trigger* en GPU, llamado Allen, mientras que los dos enfoques anteriores están diseñados para la secuencia de reconstrucción del *trigger* en CPU, llamado HLT2.

Por otro lado, la tesis aborda parte de la puesta en marcha de los calorímetros para el Run 3, detallando la tarea de *time alignment* para los calorímetros Electromagnético y Hadrónico. Esto implica la adaptación del método anterior a la nueva electrónica, la recopilación y el análisis de datos, y hacer la alineación temporal de los aproximadamente $10,000$ canales de los calorímetros con una precisión de 1 ns.

Contents

List of Figures

List of Tables

Acknowledgments

La primera vegada que vaig sentir a parlar del CERN va ser durant una de les disperses classes de Física de segon de carrera d'en Xavier Vilasís. Vuit anys després, puc dir que he tingut la sort de descobrir i formar part d'un món apassionant que m'ha fet créixer tant professionalment com personalment en molts aspectes. Aquesta tesi és el resultat d'anys de feina, experiències i el suport de molta gent durant tot el camí.

Vull agrair, en primer lloc, a l'Àlex, per estar-hi a les bones i a les dolentes, per confiar en mi i no deixar-me mai la mà, per tota la paciència i per deixar-me compartir la vida al teu costat. També a tota la família, pel seu suport incondicional. Als meus pares, per esforçar-se a intentar entendre tot el que faig i dir-me sempre com d'orgullosos estan de mi. A l'Avi, per inspirar-me i motivar-me des de petita a fer-me preguntes sobre la ciència. I a l'Àvia, perquè durant tota la seva vida va procurar el millor per mi i per tota la família, sense importar el cost.

No puc deixar d'agrair a tots els amics que han fet més amena la feina aquests anys. Al Manel i a l'Alejandra, per totes les tardes de diumenge i per seguir creixent al nostre costat. Als "júniors" del grup de recerca, Sergi, Juanma, Jessie i Guille, per les bromes, les cerves i per reservar l'Àgora alguna tarda per fer *team-building* jugant a ping-pong. Also all the colleagues and friends that I have met at CERN, specially Yuya, for the laughs and long days and nights at the pit, to whom I hope to keep a long friendship. A l'Aniol, per totes les anècdotes i moments divertits, i a la Paula, per fer la convivència al CERN molt més amena i per ser una amiga amb qui comptar per qualsevol cosa i amb qui espero continuar compartint moltes més experiències.

On the other hand, I would also like to mention the many great experts that have trusted me and have made everything easier since the beginning, specially Edu, Carla and Frédéric.

Finalment, tinc molt a agrair a tot el meu grup de recerca, per acollir-me amb els braços oberts i fer-me sentir com a casa. A l'Elisabet, a qui he trobat molt a faltar al costat de la meva taula l'últim any, gràcies per compartir ensenyances professionals, literàries i de la vida, per mi ets tota una referent. A la Míriam, per ser, dels dos, la codirectora amb els peus a terra, per la paciència responent les meves preguntes infinites i per confiar en mi des del primer moment. I al Xavier, per ser el meu codirector i no posar-me límits, per incentivar la meva creativitat a trencar barreres quan tot sembla impossible. Gràcies per confiar en mi i donar-me ales. Tant de bo no deixis mai de donar classes i puguis continuar inspirant a tants altres alumnes la passió per la recerca que em vas transmetre a mi.

Moltes gràcies a tots.

Chapter 1

Introduction

The Large Hadron Collider (LHC) [1], together with all its experiments at CERN, are continuously evolving at the edge of technology to improve the quality of their research. The physics conducted at the LHC not only advances the understanding of the universe's fundamental properties but also has far-reaching implications for science, technology, and our perspective on the cosmos. By confirming theories such as the Higgs boson and delving beyond the Standard Model, it rigorously tests the boundaries of scientific knowledge. The technological progress in computing, data analysis, and engineering driven by the LHC have led to innovations in medical imaging, materials science, and other fields. Moreover, the international collaborative efforts promote the sharing of knowledge, techniques, and resources, contributing to the advancement of science and technology on a global scale.

Among the four main physics experiments at the LHC, the Large Hadron Collider beauty experiment (LHCb) is specifically designed to study and measure the properties of hadrons containing charm and beauty quarks [2]. Through the investigation of these heavy quark-containing particles, the LHCb experiment aims to comprehend the underlying mechanisms responsible for one of the fundamental puzzles in the fields of particle physics and cosmology, the matter-antimatter asymmetry [3].

These studies are conducted through the analysis of proton-proton collisions produced by the LHC inside the particle experiments during data taking periods known as Runs. After the two first periods of data taking in the LHCb experiment, Run 1 (2010-2012) and Run 2 (2015-2018), data from a total of 900 trillion proton-proton collisions was collected and studied [4]. However, the precision of many of the key physics measures remained limited by a large statistical uncertainty. In 2012, a proposal for a major upgrade to operate the LHCb experiment at larger luminosities was first formalised [5].

The implications of the named LHCb Upgrade I reach many different levels. Some sub-detectors in LHCb have been entirely replaced with newer and more precise technologies, like the Vertex Locator or the Upstream Tracker. Other sub-detectors like the Calorimeter system have updated its readout electronics to handle the increased collision rate and the radiation damage. On the other hand, the Upgrade I involves a major upgrade of the experiment's data acquisition system [6].

In order to cope with the collision data rate of 40 MHz produced by the LHC, data is filtered by a set of triggers to select interesting collisions, or events, prior to being stored into long-term storage. During Runs 1 and 2, the trigger system consisted of a first Level zero trigger (L0) implemented in hardware and a High Level Trigger (HLT) in software [7, 8]. The L0 trigger selected high momentum particles using information from the Calorimeter and Muon Chamber detectors. It was also used to remove events that were hard to reconstruct due to their high multiplicity, leaving an output rate lower than 1 MHz of events to process in the HLT. The software trigger was in charge of reconstructing the events using the detector information and selecting the relevant ones using dedicated algorithms. After the trigger, data was sent to storage and re-processed offline with finer reconstruction algorithms to obtain the best possible reconstruction quality of the selected events.

While the L0 hardware trigger effectively managed the high data rate generated by the detector, it was saturating the acquisition rate of specific crucial decay channels. This resulted in the omission of significant events that should be selected and stored. Moreover, the software trigger provides enhanced flexibility and adaptability. As a result, the Upgrade I phase involved the removal of the L0 trigger, resulting in a trigger system transformed into a fully software-based trigger. The upgraded HLT performs offline quality reconstruction, efficiently processing up to 30 MHz of non-empty proton-proton collisions. With an average event size of 100 kB and an estimated total bandwidth of 40 Tb/s, the LHCb trigger system currently has a unique approach to data taking and analysis process called Real-Time Analysis (RTA) [9]. The increased throughput requirements have consequently enhanced the need to optimize and accelerate the reconstruction and selection algorithms of the trigger.

The installation and commissioning of the Upgrade I detectors and infrastructure was started in 2021 and has culminated during the entire 2022. This thesis has its context in the LHCb commissioning for the Upgrade I, focusing on software contributions for one of the eight sub-detectors of LHCb, the Calorimeter system, which provides a high precision measurement of position and energy deposits of neutral particles.

The second chapter of the thesis gives an introduction to the LHCb experiment and its sub-detectors, as well as the trigger system. Chapter 3 details the insights of the Electromagnetic Calorimeter reviewing its electronics and the data reconstruction procedure.

The initial objectives of this thesis derive in two branches. The first one comprises the development and implementation of the time alignment procedure for the electromagnetic and hadronic calorimeters during the Upgrade I commissioning period for Run 3. The motivation behind this line of work relies on the opportunity to contribute to the LHCb commissioning with one of the key synchronization tasks that allow an accurate data taking from the detectors. In the course of this thesis, two specific objectives are defined within the time alignment branch. The first one concerns the adaptation, testing and validation of the time alignment method used in previous runs to the Upgrade I electronics and conditions. The second one involves taking and analysing data with the detector to provide a fine time alignment configuration for the 7486 individual measurement channels of the calorimeter system. Chapter 4 describes in detail the time alignment process as well as the steps performed during the commissioning stage and the results reached until the end of this thesis.

The second main branch of this work focus in the study of alternative algorithms to perform the reconstruction of the calorimeter data using innovative and optimized approaches. The initial objective concerns providing a calorimeter reconstruction algorithm that improves the complexity of the current algorithm used in LHCb while maintaining its physics performance. The motivation behind this objective relies in the overall acceleration of the reconstruction sequence in the Real Time Analysis framework of the experiment. The task known as the calorimeter clustering, consists in grouping the energy deposits from the particles into clusters and computing its total energy value. The high occupancy of particles in the detector makes it a complex task with many clusters overlapping in the same event.

The first approach to calorimeter reconstruction, presented in Chapter 5, takes advantage of the increasing popularity of Deep Learning techniques. However, instead of building a deep neural network architecture to perform the clustering, the procedure is segmented into small steps that can be formulated as a cellular automaton and learned by simple convolutional neural network architectures. The results, presented in the 25th International Conference on Computing in High-Energy and Nuclear Physics (2021) [10] and published in Applied Sciences (2021) [11], show that a good learning is achieved but the energy resolution of the obtained clusters is not comparable to the LHCb resolution. However, the inference time obtained is almost constant with respect to the event's complexity. Although the very interesting results, further developments of this approach were stopped due to the lack of an efficient inference engine for neural networks inside the LHCb framework. Further discussion on the inference problem is presented in the 25th International Conference of the Catalan Association for Artificial Intelligence (2023) [12].

The second approach to calorimeter reconstruction presented in this work concerns a graph-based clustering algorithm that is the current solution for the calorimeter reconstruction in the second part of the trigger system (HLT2) for Run 3. Chapter 6 details the methodology and performance of the called Graph Clustering algorithm. It takes advantage of graph structures to store the energy deposits of an event and optimizes the clustering process. Results, published in The European Physical Journal C (2023) [13], show that Graph Clustering has a cluster resolution equivalent to the previous method used in LHCb while bringing a slightly higher efficiency and a 65.4% improvement in execution time on average.

As a final reconstruction contribution, Chapter 7 concerns a first effort to improve the calorimeter reconstruction for the first part of the trigger system (HLT1), run on GPUs. Given that the current algorithm is a very simplified version of the full reconstruction procedure done in HLT2, the cluster overlap resolution logic from the Graph Clustering is added to the current algorithm using three different approaches in CUDA. Results concerning the efficiency and resolution, as well as the throughput impact have been presented in 26th International Conference on Computing in High-Energy and Nuclear Physics (2023) [14].

Finally, Chapter 8 discusses about the metrics used to evaluate the performance of the reconstruction algorithms and presents a general conclusions of the thesis as well as the future lines of work.

Chapter 2

The LHCb detector at CERN

CERN, the European Council for Nuclear Research, established in 1954 the largest and most complex particle physics laboratory in the world to study the basic constituents of matter. There, a wide variety of physics experiments have taken place with the purpose of learning more about fundamental mysteries, starting with understanding the structure of elemental particles and evolving to the study of new physics.

Located in the Franco-Swiss border near Geneva, CERN has a network of purpose-build particle accelerators that lead to the Large Hadron Collider (LHC) [1], a 27 km ring of superconducting magnets placed 100 meters underground where two high-energy particle beams travel at close to the speed of light. Around the accelerator ring, there are four locations in which the beams inside LHC are made to collide, corresponding to the positions of the four main particle detectors ATLAS, CMS, ALICE and LHCb. Although each detector has its own physics programme, they all share the principles of measuring physics parameters like velocity, mass and charge of the particles produced in the collisions using dedicated sub-detectors. The data produced in every collision is processed in real time and gathered by an online system, which decides which information to store for later physics analysis.

The accelerator complex at CERN, shown in Figure 2.1, is a succession of machines that accelerate proton particles to increasingly higher energies up to the current record of 6.8 TeV per beam for the LHC. The first element of the chain is the linear accelerator *Linac4*, then particles travel through synchrotrons *Booster*, *PS* and *SPS* after being injected to the two beam pipes of the LHC in opposite directions. At the collision points, the combined energy is equal to 13.6 TeV. The accelerated protons circulate through the complex in groups called *bunches*. Therefore, when two opposite bunches cross, there is a known probability that two individual protons collide, which defines the average number of collisions per bunch crossing. A bunch crossing is referred to as an *event* and occurs every 25 ns, which represents a designed event rate of 40 MHz.

Through the operational history of LHC, there have been data-taking periods, called Runs, interleaved by Long Shutdown periods, used to upgrade and maintain the experiments. Between some of the Runs, the *instantaneous luminosity* of LHC is increased, which determines the average number of particles that collide in an event, leading to more complex

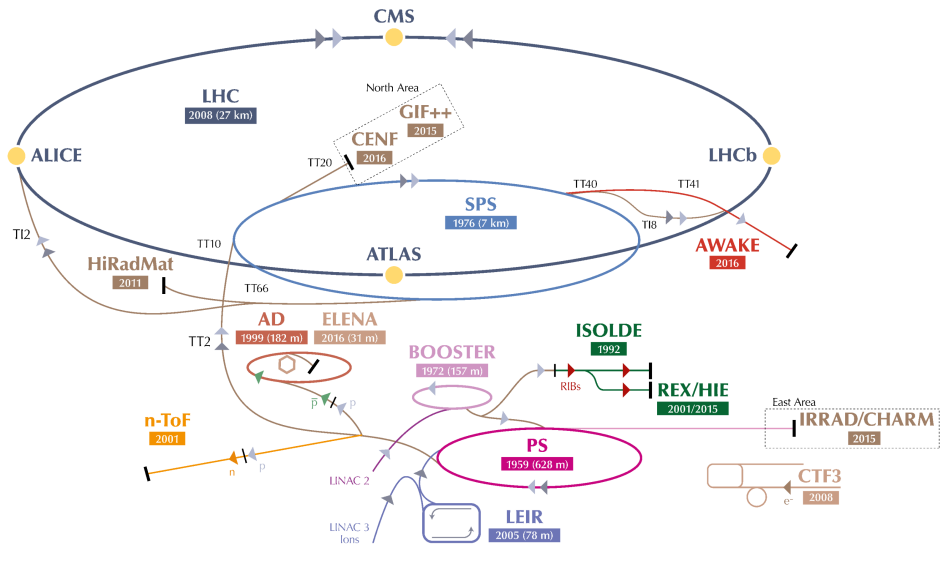


Figure 2.1: The CERN accelerator complex [15].

experimental systems and data. During the recent Long Shutdown 2 that started in 2018 and lasted until 2022, the LHCb experiment underwent a major upgrade (Upgrade I) to cope with the increased luminosity, one order of magnitude above the original LHCb design value. In the current period of data taking started in 2022 (Run 3), the LHC has an *instantaneous luminosity* of $2 \times 10^{33} cm^{-2} s^{-1}$, which is the highest luminosity ever achieved by the experiment. Until the end of the LHC era, expected to be in 2041, plans involve further increasing the luminosity up to the High Luminosity LHC (HL-LHC) operational period, which will require a second major upgrade (Upgrade II) for the LHCb experiment foreseen to start in 2030 [16].

2.1 The LHCb Detector

The Large Hadron Collider beauty (*LHCb*) experiment is one of the four main experiments at the LHC [2, 17, 18]. It is a single-arm forward spectometer designed to study the decays of the *beauty* quark, which could explain the differences between matter and antimatter, also known as *CP violation*.

The layout of the current LHCb detector, in the context of the *Upgrade I* that took place between 2019 and 2022, is shown in Figure 2.2. The particle collision is produced in the left-most sub-detector, the Vertex Locator (*VELO*). From there, the produced particles decay into new particles that propagate in all directions from the interaction point. However, the LHCb detector only covers a small angle range in the forward direction (Z axis in Figure 2.2), 300 mrad in the horizontal axis and 250 mrad in the vertical axis, which is the angular

acceptance expected for the beauty quark decays.

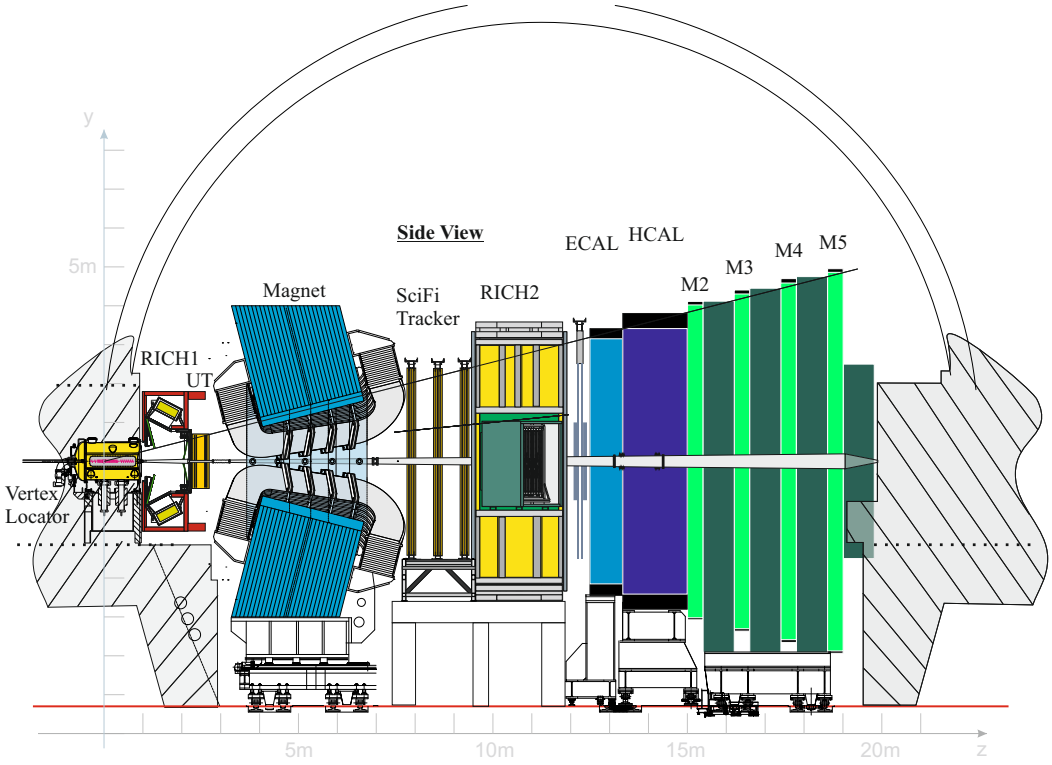


Figure 2.2: Side view of the Upgrade I LHCb detector.

The total of eight sub-detectors from LHCb can be classified in two systems according to its purpose. The *tracking* system [19] is meant to reconstruct the path of charged particles that travel through the detector to measure momenta and collision vertices. The particle identification system (*PID*) is meant to measure other physics parameters such as velocity, energy and mass that allow the identification of individual particles.

2.1.1 Tracking sub-detectors

Three different tracking sub-detectors measure the path of the particles at different positions in the Z axis: the *VELO*, the Upstream Tracker (*UT*) and the Scintillating Fibre Tracker (*SciFi*). A *dipole magnet* [20], designed for a total integrated field of 4 Tm, is placed between the UT and the SciFi. It provides a vertical field that cause charged particles to bend along the horizontal plane, allowing to measure its momentum.

The tracking system of LHCb [19] consists in combining the traces of paths along the three sub-detectors to reconstruct the individual particle trajectories of an event. Depending

on their trajectories, particles may leave a trace in different sub-detectors, giving them a classification as illustrated in Figure 2.3. VELO tracks are only detected in the VELO since they fall out of the angular acceptance of LHCb and are used to reconstruct primary vertices. Upstream tracks originate on the VELO and travel through the UT but are bent outside the acceptance in the magnet. Long tracks also originate in the VELO and traverse all the tracking sub-detectors. Although the primary collisions of an event take place inside the VELO, the produced particles can decay after travelling some distance, originating non-primary vertices of other particles and tracks. Downstream tracks are only detected in the UT and the SciFi, and T tracks only in the SciFi. Both are produced from particles that decay outside the VELO acceptance.

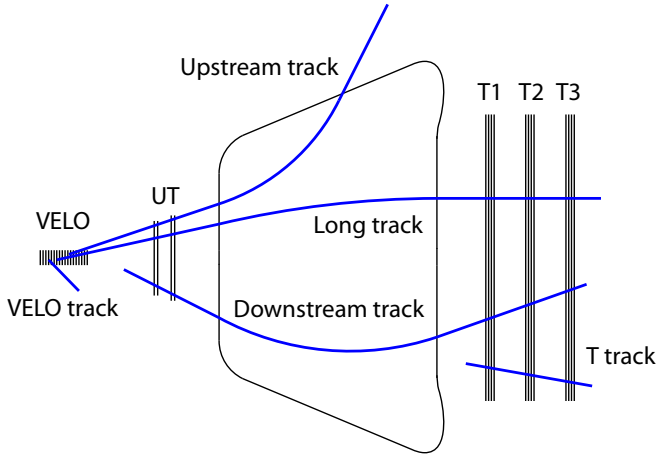


Figure 2.3: Side view of the LHCb tracking sub-detectors and track types.

Different tracking algorithms are used to reconstruct different track types. However, the track extrapolations used to match the track parts from different sub-detectors are based on parametric models of trajectories in the LHCb magnetic field for computational speed reasons. After the tracking process, a separate step based on a *Kalman filter* [21] is used to maximize the accuracy and precision of the particle trajectories.

VELO

The Vertex Locator [22] is the only sub-detector at LHCb that surrounds the interaction region and is closest to the beam pipe. Thereby, it measures the location of the primary vertices, displaced decay vertices and the distance between them. As shown in Figure 2.4, it consists of 52 modules of pixelated silicon detectors placed along the beam pipe. The modules are arranged into two movable halves, allowing the VELO to be retracted when there is no stable beam in the LHC in order to avoid damage to the sub-detector. When closed, the radius of the closest active pixel edge is 5.1 mm from the beam line.

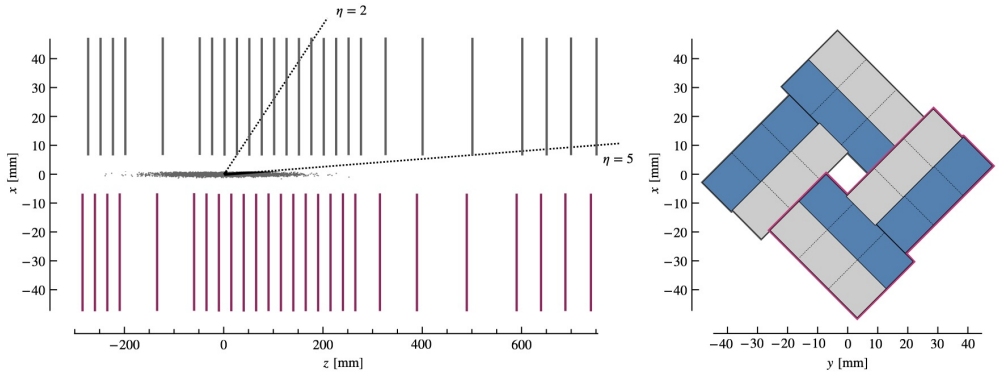


Figure 2.4: Left: schematic top view of the XZ plane inside the VELO. Right: sketch showing the nominal layout of the modules around the Z axis in the closed VELO configuration.

The reconstruction process in the VELO consists in finding the seeds of tracks using pattern recognition techniques [23], as well as a precise measurement of the primary vertices of a collision. As there is effectively zero magnetic field in the VELO, the tracks can be reconstructed as straight lines and therefore extrapolated to the UT.

UT

The Upstream Tracker sub-detector [24] consists of four planes of silicon detectors organized in two stations. The first station is composed of a layer with vertical strips named UTaX, and a stereo layer with strips rotated by 5^o named UTaU. The second station is similar, with a first stereo layer rotated -5^o named UTbV and a straight layer named UTbX. The two pairs of stations are symmetrically arranged along the beam pipe. Precise measurements of the x position are derived from the vertical strips while the y position can be determined by combining the measurements from the rotated U and V planes. Moreover, the region closest to the beam pipe has special sensors to maximize the active area.

The UT is used for the tracking of charged particles and its data is combined with the VELO tracks. Given the presence of residual magnetic field, a first determination of the track momentum and charge is obtained. Although the precision is moderate, it allows to speed-up further tracking algorithms and reduce the rate of reconstructed tracks that do not match any real particle, known as *fake tracks*.

SciFi

The Scintillating Fibre Tracker [19] is located behind the dipole magnet and consists of three tracking stations. Each station has four layers in an X-U-V-X configuration, where the X layers have their fibres oriented vertically and are used to measure the deflection of tracks

caused by the magnet. The two inner layers, U and V, have their fibres rotated by $\pm 5^0$ in the layer plane to reconstruct the vertical position of the track.

The SciFi is used for tracking charged particles and the measurement of their momentum. It allows the reconstruction of Long, Downstream and T tracks.

2.1.2 PID sub-detectors

The particle identification system combines the information of the two Cherenkov detectors, the Calorimeters and the Muon chambers to identify the five basic long-lived charged particle species: electron, muon, pion, kaon and proton, as well as neutral particles decaying in the detector. Compared to the tracking system, particle identification sub-detectors focus in specific physics measurement at a different momentum range. Therefore, the combination of its information into global multivariate classifiers is what allows the optimal particle identification performance.

Cherenkov detectors

The *Cherenkov radiation* [25] is a phenomenon that occurs when a particle travels through a certain medium at a speed faster than the speed of light in that medium. Similarly to the breaking of the sound barrier, at that moment, the particle emits a cone of photons at an angle θ. This angle is related to the refractive index of the material n and the velocity of the particle β as:

$$\cos\theta = \frac{1}{n\beta}. \tag{2.1}$$

In LHCb there are two Ring Imaging Cherenkov (RICH) detectors [26] that make use of the Cherenkov effect to identify the particles nature. They are in charge of performing the hadron discrimination in the momentum range between 2.6 and 100 GeV, which consists primarily on the separation between pions, kaons and protons but also for electrons and muons. They are also used to heavily reduce the combinatorial background. Figure 2.5 shows a simulation of how the Cherenkov photons interact inside the RICH1.

The RICH1 is located between the VELO and the UT and identifies low momentum particles, whereas the RICH2 is placed after the SciFi and provides identification for particles with high momentum. Each of them is filled with radiator gas and uses a combination of spherical and flat mirrors to reflect the cone of Cherenkov light produced when particles traverse the radiator. The photons produced are then detected in a grid of Multi-anode Photo-Multiplier detectors. The readout of the detector results in an image with pixels arranged as almost circles that match an incoming track on its center.

It is studied that for a given particle mass, the angle θ depends on the momentum of the particle. Thus, combining the information of the tracking system with the measurement of the radius of a circle in the RICH system, a particle can be identified.

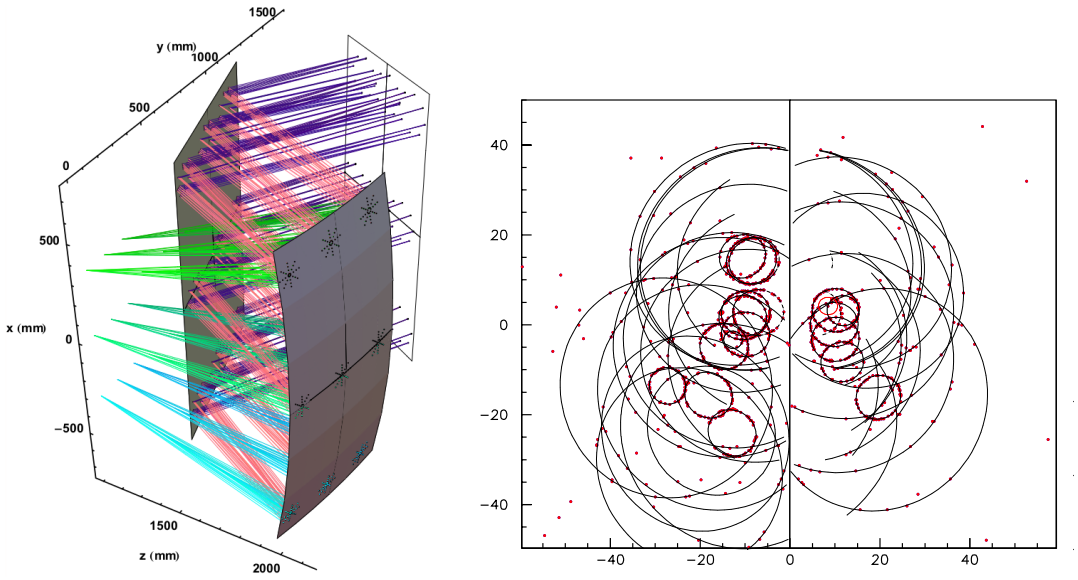


Figure 2.5: Left: simulation of the Cherenkov photons and their reflections in the mirrors of half RICH1. Right: simulation of detected Cherenkov photons in both sides of the RICH1 [27].

Calorimeters

The LHCb calorimeter system [28, 26] consists of two sub-detectors, an electromagnetic calorimeter (ECAL) followed by a hadronic calorimeter (HCAL). Its main purpose is the identification of neutral hadrons with a high precision measurement of position and energy deposited.

The ECAL consists of three different regions of detector modules with different granularities to cope with the particle occupancy in the detection area. The inner region is the closest to the beam pipe and has the detector modules with highest granularity. From there outwards, the middle and outer regions have increasing module dimensions. The ECAL focuses on the measurement of electromagnetic particles that are stopped inside the detector and therefore deposit all of its energy.

The HCAL is segmented in two regions with larger granularity with respect to the ECAL. It is meant for the identification of hadronic particles although the energy measurements do not contain the full hadronic shower but provide a good and fast estimation.

The two calorimeters share the same detection principle consisting on modules with scintillating tiles transmitting light to photo-multipliers (PMTs) with wavelength-shifting fibres combined with metal stoppers. The resulting data from the sub-detectors are energy deposits that are grouped to account for the energy deposited by individual particles. For ECAL, cells are in general clustered as groups of 3×3 modules.

Muon chambers

The LHCb muon system [29, 26] is composed of four stations M2 to M5 located behind the calorimeter system. Since muons have an interaction probability with matter lower than electrons and a longer lifetime, they can traverse all other sub-detectors of LHCb. Hence, the muon chambers provide a precise identification and reconstruction of muons.

Each station is equipped with Multi-Wire Proportional Chambers (MWPCs) interleaved with iron absorber plates to filter low energy particles. The detection area is divided into four regions, R1 to R4, of decreasing granularity moving from the beam pipe outwards, in order to uniformly distribute the particle flux and the channel occupancy across each station.

2.2 The Trigger System

The collision rate provided by the LHC is of 40 MHz, which produces 5 TB/s of data with all the LHCb sub-detectors. This amount is not storable with the current available technology. Therefore, not all the events derived by collisions are stored. The *trigger system* is in charge of reducing the data volume by a factor 500 to around 10 GB/s that are sent to permanent storage. To ensure that the specific signals of interest for physics are stored, the trigger system performs a full reconstruction of the event followed by a set of selection algorithms tuned for a particular signal topology or physics analysis. This approach is called *real-time analysis* (RTA) [9]. Not only it makes an efficient selection of the events but also allows to save only the relevant subsets of information in specific cases to optimize the throughput and storage [30]. In figure 2.6, a detailed diagram of the data flow in the real-time analysis approach is shown.

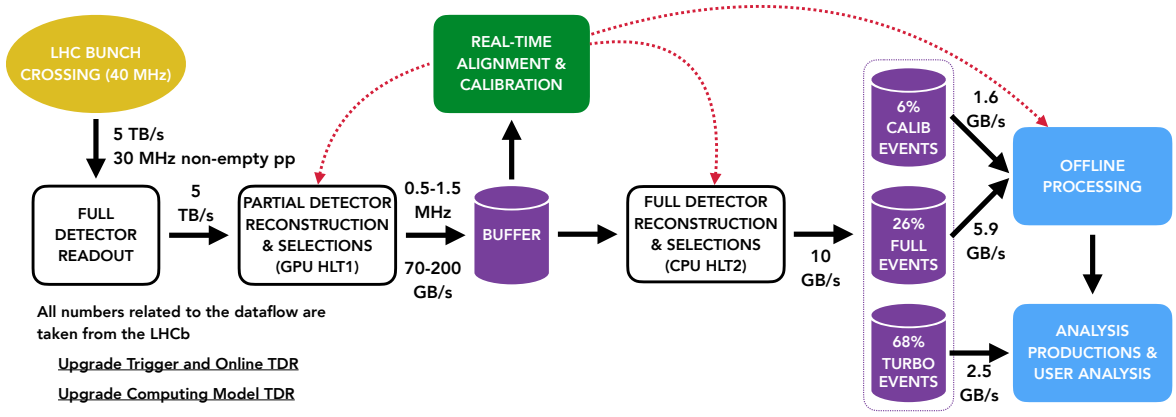


Figure 2.6: LHCb online data flow [31].

In order to meet all the requirements, the approach results in a two stage trigger system. There is a first high level trigger implemented on GPUs, the first high level trigger *(HLT1)*

[32], that reduces the data volume by roughly a factor 20 and is based primarily on the reconstruction of charged particles. Next, a second step implemented on CPUs, the second high level trigger *(HLT2)*, performs a complete reconstruction of the full event, including the tracking, calorimeter reconstruction, particle identification and the Kalman fit. Then, a selection of the physics signatures is done with the order of 1000 selection algorithms tuned for a specific signal topology. To ensure the reconstruction quality in HLT2 is maximum, the *alignment and calibration* performed in quasi-real-time is key. It consists of a set of algorithms that measure with high precision the physical position and calibration parameters of each sub-detector to provide the most accurate alignment and calibration parameters for reconstruction and selections. Between the two trigger stages, a disk buffer of 30 PB is placed to hold the data while the alignment and calibration is performed.

After the selection process in HLT2, the information is sent to permanent storage using three *streams* according to its purpose. The *full* stream stores all the reconstructed particles for events that can benefit from offline re-calibration. The *Turbo* stream allows to store only relevant information of the reconstructed events. The *TurCal* stream stores entire events for offline re-calibration.

2.2.1 High Level Trigger 1

The HLT1 is conceived as a first filter to reduce the event rate to a level at which the data can be buffered to disk for real-time alignment and calibration and further processing in the HLT2 stage. It is designed attending trade-offs between speed, efficiency and output rate. The baseline reconstruction is focused on finding Long, Upstream and Downstream tracks, measuring their momenta with a percent-level precision, associate them to a primary vertex to measure its displacement and identify the particle as a muon or non-muon. On its reconstruction sequence, there is a Global Event Cut (GEC) that removes a fraction of the events with higher occupancy, which imply a significant increase in the reconstruction computing time and have a worse detector performance. Although the GEC has been revised through the detector commissioning, the design criterion is to reject 7% of the events based on UT and SciFi occupancies. The selections made in HLT1 can be divided into four categories: inclusive selections for the majority of LHCb physics interests; calibration samples to evaluate the reconstruction performance; selections for specific physics signatures not covered by the inclusive selections; and technical selections for monitoring and luminosity measurements.

The data acquisition (DAQ) system [6] receives the event fragments from the front end electronics of all the sub-detectors from LHCb and combines them into coherent blocks of data to construct events. This process of *event building* is performed by around 170 event builder PCs that can host up to three graphic processing units (GPU) each. These GPUs are the hardware used to run the HLT1 sequence directly in the event building step [33]. At the start of this thesis in 2020, the HLT1 trigger reconstruction had a total throughput of 38,198 events/s/node, as shown in Figure 2.7 together with the breakdown of the reconstruction throughput rates for all the algorithms in the sequence.

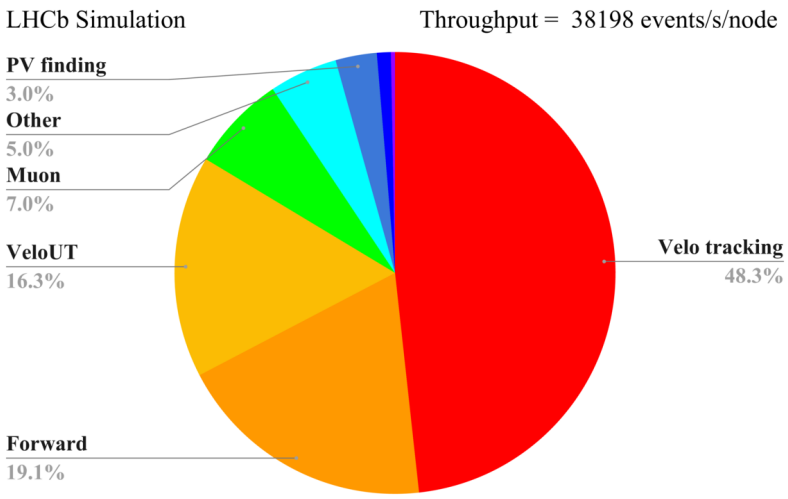


Figure 2.7: Breakdown of the HLT1 reconstruction throughput rate for the LHCb upgrade in 2020. [34].

2.2.2 High Level Trigger 2

Using the information provided by the alignment and calibration of the detector, the HLT2 performs the full reconstruction of events, enabling the selection of events to store with an optimal precision. This approach divides the reconstruction process into four main components.

The **charged particle pattern recognition** comprises the reconstruction of the different track types, illustrated in Figure 2.3. Different algorithms perform the tracking steps starting with the primary vertex finding with the VELO tracks. Then, several algorithms perform a standalone reconstruction of the tracks matching the segments from the tracking sub-detectors according to the track types. Duplicate tracks, named *clones*, can be formed when different algorithms reconstruct the same track segment in one sub-detector. Individual pattern recognition algorithms are used to remove those duplicates and keep the clone tracks to a minimum. All the track extrapolations used in the pattern recognition algorithms use parametric models of trajectories inside the LHCb magnetic field for reasons of speed.

The **Kalman fit** is a separate step based on a Kalman filter that optimizes the properties of the charged particle tracks to maximize its accuracy and precision. It is implemented using parametrizations of the particle propagation through the LHCb material and magnetic field [35].

Although the calorimeter system is composed of both ECAL and HCAL, the **calorimeter reconstruction** accounts for the ECAL reconstruction only, as at present, there are no analyses which foresee using HCAL reconstructed clusters. Therefore, the reconstruction consists in grouping the energy deposits from ECAL into clusters of generally 3×3 cells.

Further detail on the ECAL clustering is given in Section 3.2. Then, multivariate algorithms which use the shower shape and individual cell energies are used to distinguish between single-photon clusters and multiple photon clusters. Electron clusters are identified by matching an ECAL cluster to the extrapolation of tracks. Other tools are used to distinguish photons from hadrons and pile-up clusters.

The **particle identification** step uses a combination of the two RICH detectors, the ECAL and the muon system information to identify the five basic long-lived charged particle species: electron, muon, pion, kaon and proton. Depending on the momentum regime, the particle identification performance is dominated by a different sub-detector, except for the muons where the muon system plays a significant role in all cases. An accurate knowledge of the track trajectory is key to achieve a good performance in the particle identification, therefore, the Kalman fitted tracks are required to maximize the results. An optimal performance is achieved when the information provided by each sub-detector is combined into global multivariate classifiers trained on simulation data and tuned to have a better and more stable performance in different kinematic regions.

At the start of this thesis in 2020, the HLT2 reconstruction sequence had a total throughput of 133 Hz. The breakdown of the rates for all the algorithms in the reconstruction sequence can be seen in Figure 2.8.

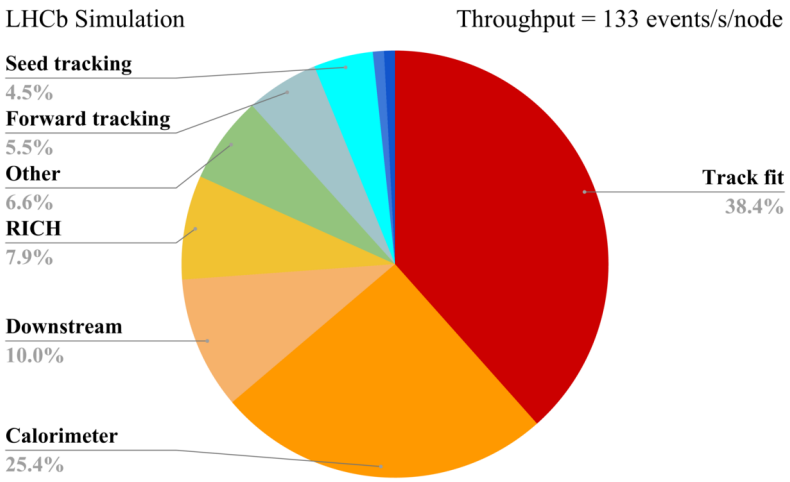


Figure 2.8: Breakdown of the HLT2 reconstruction throughput rate for the LHCb upgrade in 2020. [34].

Once the reconstruction process has been performed, the selection in HLT2 relies on the order of a thousand different selection algorithms, each one tuned for a specific signal topology and/or physics analysis. The algorithms typically use multivariate or artificial intelligence-based selections.

2.2.3 Alignment and Calibration

The HLT2 is able to make a full quality reconstruction and efficient selection of events in the real-time analysis paradigm thanks to the alignment and calibration of the detector in quasi-real-time. Its purpose is to provide the most accurate alignment and calibration parameters to ensure the physics parameters of interest, such as particle mass or decay-times are computed with the best possible resolution. There are several steps in the real-time alignment and calibration procedure that use different input samples and are performed at a different frequency. Figure 2.9 illustrates the timing of the various procedures within an LHCb fill.

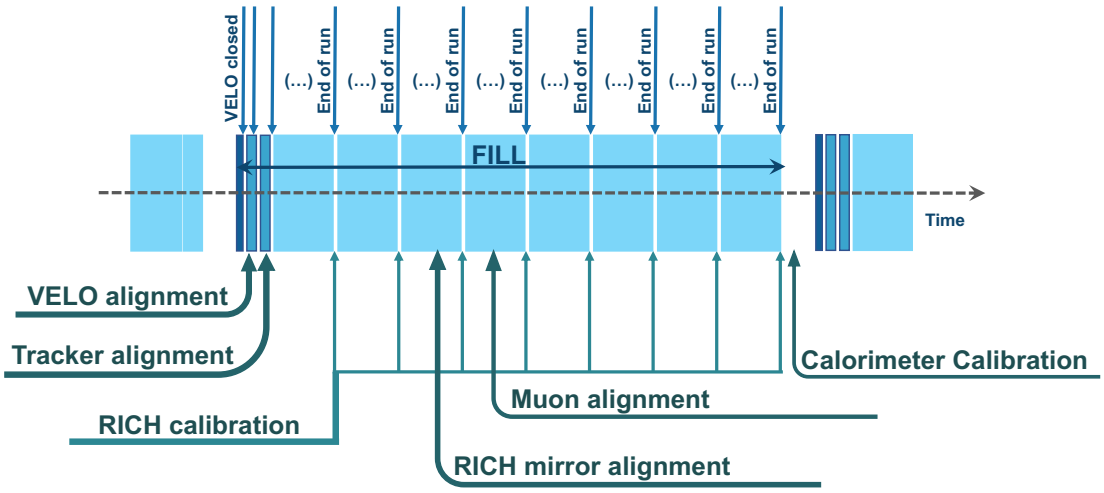


Figure 2.9: Schematic view of the real-time alignment and calibration procedure starting at the beginning of each fill.

The alignment steps comprise in general the measurement of the physical position of entire sub-detectors and its modules, to check for its positional alignment. The tracker alignment step measures the alignment between the tracking sub-detectors. It is done at the beginning of each fill within a few minutes. The VELO alignment is dedicated to the alignment of each of its modules and is also performed at the start of the fill to account for the opening and closing of the detector. The Muon alignment is run as a monitoring task through the fill. The RICH mirror alignment measures the position of all the spherical mirrors. It takes some tens of minutes since it requires a sample of Cherenkov photons equally distributed among the RICH1 and RICH2 mirrors.

The two RICH sub-detectors also require a calibration of the gas radiator refractive index, performed once per run using dedicated samples selected in HLT1. For the calorimeter system, only ECAL requires an accurate calibration of the PMT high voltages after each fill

to ensure the energy measurements are accurate and efficient. A more fine-grained calibration based upon the observation of the π^0 mass on each ECAL cell is performed once per month.

Chapter 3

The Electromagnetic Calorimeter

The LHCb calorimeter system consists of two sub-detectors, the electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL), as introduced in Chapter 2.1.2. This chapter is devoted to detail the general structure, electronics and data reconstruction of the electromagnetic calorimeter.

The main purpose of ECAL is the identification of electrons and photons, and the measurement of their energies and positions with high precision. This can be done as electrons and photons start an electromagnetic shower when entering in the sub-detector. This process happens when an incoming electron produces a bremsstrahlung photon or when an incoming photon creates an electron-positron pair. The resulting photons and electrons produce additional photons and electrons with lower energy, resulting in a cascade. When the electron energies fall below a critical threshold, they stop producing more bremsstrahlung and eventually, the energy of the incoming photon or electron is fully absorbed by the sub-detector material.

The ECAL has a rectangular shape of $7.8 \times 6.3\ m^2$ and is placed perpendicular to the accelerator beam pipe at a distance of 12.5 m from the interaction point. The energy measurement area is segmented into individual square-shaped modules. Each module has a *shashlik* structure with alternated scintillator tiles (4 mm) and lead absorber layers (2 mm). The scintillation light readout is performed by dedicated photo-multipliers (PMTs). The general structure is segmented in three different rectangular shaped regions, as can be seen in Figure 3.1.

Although all modules have the same size of $12 \times 12\ cm^2$, the number of readout cells on a module depends on the region. The inner region is the closest to the beam pipe and has the highest occupancy of incident particles. Thus, it has the highest granularity among the three regions, with nine readout cells of $4 \times 4\ cm^2$ per module. This size has been chosen according to the *Molière* radius of the cells. The middle region surrounds the inner one and has four readout cells of $6 \times 6\ cm^2$ per module. The outer region has a single readout cell of $12 \times 12\ cm^2$ per module. The total number of cells is 6016.

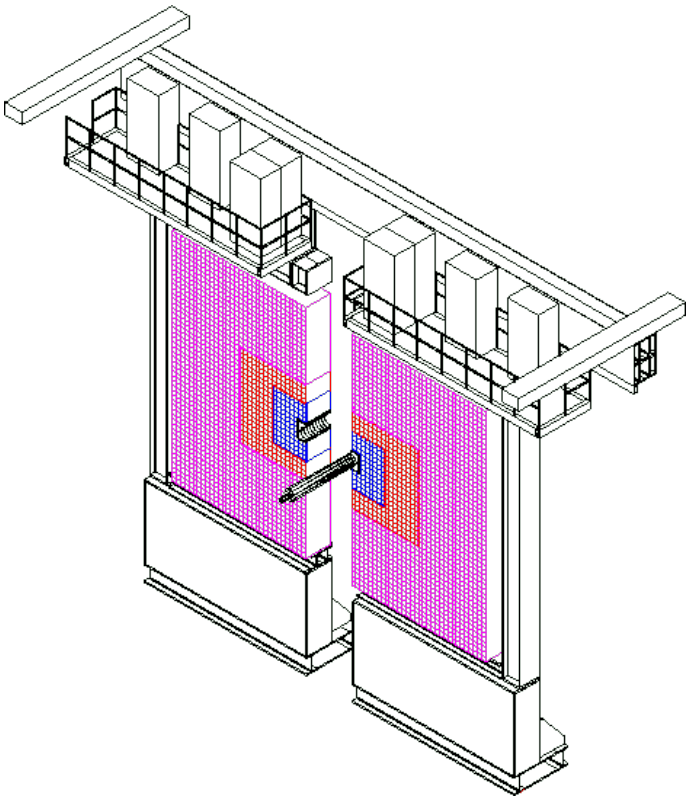


Figure 3.1: The electromagnetic calorimeter 3d view from behind the detector towards the interaction point [28].

3.1 Electronics

With the upgrade of the trigger system, the readout electronics from all the sub-detectors in LHCb have changed significantly with respect to the previous system. The new architecture transmits the data collected from every bunch-crossing directly to the event-builder computing farm, which assembles all the pieces of data that belong to the same bunch-crossing for every collision. Therefore, the complete event information from all the sub-detectors can be used in the trigger system.

3.1.1 General architecture

The general electronics structure is divided into front-end (FE) and back-end (BE) electronics. The FE electronics amplify, shape and digitise the signals generated in each detector cell and send them through optical links to the BE. All components of the FE electronics are located on or close to the detector. The BE electronics are situated in a data center on the surface and connected to the FE by 250 m long optical fibres. There, the BE electronics

pre-process and format the data to transmit it to the event builder. Apart from the detector data, clocks and beam-synchronous commands are distributed by the timing and fast control unit (TFC). Other control signals come from the experiment's control system (ECS) which configures and monitors the BE and FE and implements slow controls like the high voltage (HV), low voltage (LV) and temperature monitoring. To allow the proper reconstruction of events, data packages from the FE are tagged with a unique time-stamp based on the bunch-crossing identifier (BXID). A schematic of the general electronic's architecture is shown in Figure 3.2.

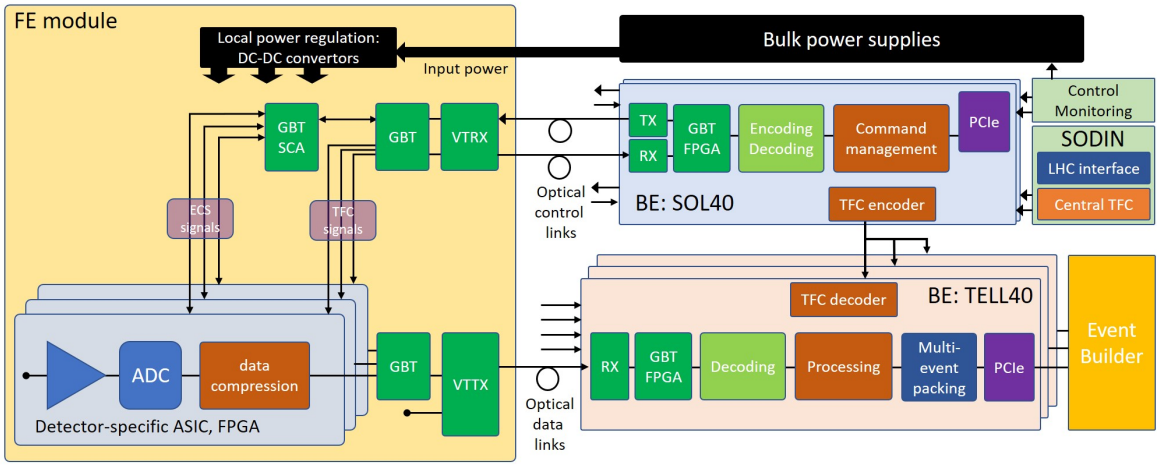


Figure 3.2: Electronics architecture of the upgraded LHCb experiment [36].

The specific implementation of the FE architecture for the calorimeter is described in the following section. The BE electronics generally consist of custom PCI-express modules mounted in the PC servers from the data center. This module, known as PCIe generic back-end board (PCIe40), contains arrays of optical transmitters and receivers connected to an FPGA. There are specific PCIe40 boards dedicated to data acquisition (TELL40), which decode and process the data from the FE and then build multi-event packets that are transmitted to the event builder. Another set of PCIe40 boards are for the control system (SOL40), which are used as the ECS interface for configuring the FE electronics and transmitting the TFC commands to the FE and TELL40s. There are three SOL40 boards for the calorimeter, one for each ECAL side and one for the HCAL. PCIe40 modules also play the role of interface to the LHC machine timing when configured as a SODIN board.

3.1.2 The Front-End Board

The two LHCb calorimeters share the same FE electronics structure, which are organised in crates inside racks of boards located on the detectors. There are 14 crates for the ECAL and

4 for the HCAL. Each crate contains up to 16 front-end boards (FEB) that receive, amplify and shape the PMT signals and a control card unit (3CU), which distributes the clock and the control commands to the FEBs.

Before reaching the FEB, the analog signal from the PMT is clipped inside a Cockcroft-Walton (CW) base to keep it within the 25 ns bunch crossing window set by the LHC. If the signal shape is wider than 25 ns, part of the signal will be integrated also in adjacent bunch crossings, potentially merging the signal of two consecutive events. The *spillover* is a measure of the amount of signal spilled to the next or previous bunch crossing. Clipping the signal also helps to prevent the spillover which is adjusted to be less than 1%. After that, the analog signal is sent to the FEB with a 12 m long 50 Ω coaxial cable. Each FEB is connected to 32 PMT outputs, named *channels* at this stage. One channel corresponds to the readout of one ECAL cell of different size depending on the region. The channels grouped in a single FEB cover a region of 4×8 cells in the calorimeter grid.

The analog circuit of the FEB starts with an ICECAL chip [37] processing four channels. Its input stage consists of a current amplifier with an active line termination in order to avoid resistor noise. After that, the signal is sent to two interleaved lines running at 20 MHz each, synchronous with the 40 MHz global clock, named *subchannels*. Both analyse the signal from the same channel but one subchannel processes the signal from the even bunches and the other from the odd bunches. Each subchannel shapes the signal with a pole zero compensation to minimize the spillover. Then, the signal is integrated with a fully differential amplifier through the entire 25 ns bunch window to accumulate the particle showers from a collision. The two lines are needed to allow the integrators to alternately discharge during a full bunch crossing. After that, the integrated signal is stored in a track-and-hold (TH) module for another 25 ns and then it is sent to an analog-digital converter (ADC) driver through a multiplexer that alternates within the signals of the two subchannels. Figure 3.3 shows a detailed schematic of the FEB electronics.

The four channel analog output of the ICECAL is then sent to two 12-bit dual ADCs with two channels each. A specific clock is produced in the ICECAL and directly injected to the ADC to properly sample the signal. This ADC clock and the TH clock can be adjusted independently for each channel according to the *time alignment* procedure.

The digitized signals are sent to two 16 channel FPGAs which first re-synchronize the channels and process them to remove the low frequency noise and subtract the *pedestal*, defined as the background noise from the electronics. A tunable *latency* is introduced after the data synchronization in order to correct the coarse bunch crossing misalignment between channels. In a second stage of the FPGA, a preliminary clustering groups and sums signals from 2×2 cells inside the FEB region and the neighbouring FEBs. This allows a number of measurements such as the maximum transverse energy from the 2×2 clusters and its address, the total measured transverse energy and the number of cells with a measured energy higher than a programmed threshold. This quantities are added to the raw channel data as a Low Level Trigger to be further processed in the event building farm or the software trigger.

On every FEB there is one gigabit transceiver (GBTx) component which serializes the output data from the FPGAs and sends them through four optical links to the back-end

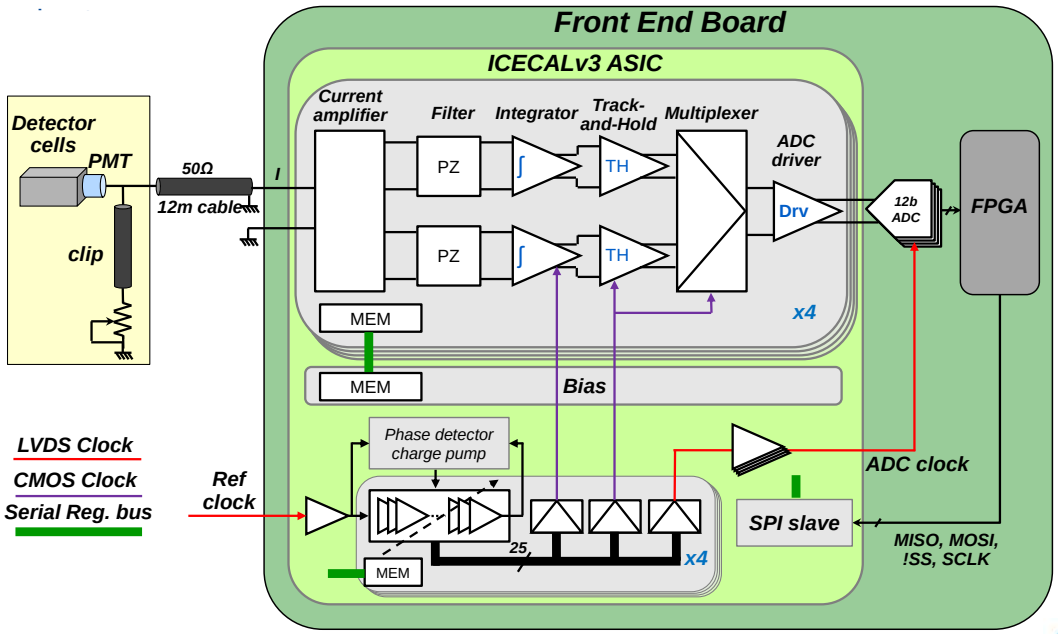


Figure 3.3: Calorimeter FEB scheme for the Upgrade I.

electronics. Thus, the requested output bandwidth per FEB is $12 \times 32 \times 30$ Mbit/s. There is also a slow control adapter (GBT-SCA) that distributes the control and monitoring signals as well as the global clock to the FEB.

3.1.3 The 3CU board

Within a calorimeter crate, the central slot is reserved to the control card unit (3CU) board. Its main role is to distribute signals from the LHCb control system to the FEBs in the same crate. Each crate has two backplanes: the lower one provides the power supplies, the trigger configuration commands and the clock distribution; the upper backplane is used to the exchange of signals between the boards and with other crates.

The 3CU board receives three types of information through an optical link: the 40 MHz clock; the TFC commands which synchronously distributes timing, trigger and control configuration; the ECS commands which allows a remote control of the detector electronics and data acquisition.

3.1.4 High voltage and LED monitoring

The output voltage of each PMT is proportional to the amount of scintillation light collected from the fibres, which can vary between cells even with incident particles of the same energy. This can be caused by factors such as variations in the properties of the fibers and the aging

of the fibers. To mitigate this effect, the gain set for each PMT is controlled by an analog voltage in a range of 0-5 V, known as high voltage, applied to the control input of each CW base. This allows to calibrate the energy readout of all the ECAL cells by performing individual and precise gain adjustments on each PMT.

This energy calibration is performed during data taking by a variety of methods, such as monitoring the cell response in ADCs to electrons, π^0s and minimum ionizing particles. Prior to collisions data, the gain of each PMT is monitored by measuring the response to an LED flash of constant magnitude injected in the PMT entrance window by an optical fibre at the center of each cell. The LED flash magnitude is adjustable by applying a control voltage to the inputs of each LED driver. Dedicated LED trigger signal boards (LEDTSB) perform the overall control and timing for the LED system. During data taking periods, the LEDs are also regularly monitored using specific calibration bunches.

3.2 Data Reconstruction

The output data obtained from the ECAL modules are the values from each readout cell concerning the accumulated energy deposited by particle showers. It is digitized and converted to MeV with a precision of 12 bits as an energy measurement. Another measure used in the data reconstruction is the transverse energy, which is computed using the energy of a cell and its angular position in the ECAL. From each collision event, a list of *digits* is retrieved. Each digit contains the energy deposit, a unique identifier of a cell, and the coordinates of the cell in the detector plane.

The cell size of the inner ECAL region is designed to contain the full shower of a particle if it starts in the center of the cell. However, since this is not the expected behavior, particle showers usually deposit energy in more than one cell. The group of adjacent cells containing energy deposits from the same particle is called a *cluster*. Then, the process of grouping all the clusters of a collision event is called cluster reconstruction.

Clusters are typically defined as groups of 3×3 cells around a local maximum energy peak. Studies have been done regarding the cluster shapes [26] where a combination of 2×2 and swiss-cross cluster shapes show promising performance for high luminosity scenarios. Recent studies also consider the possibility to have different cluster shapes for different regions to optimizing the position and energy resolution. However, the 3×3 cluster is currently used as a base for masking other shapes on clusters. Therefore, the definition of 3×3 cell clusters is maintained through all the regions of the detector.

A classic approach for the calorimeter reconstruction problem is to use the principles of a cellular automaton. The so called Cellular Automaton algorithm [38] for LHCb was first presented in 2001 and has been the benchmark calorimeter clustering solution used in LHCb for Runs 1 and 2 [39]. For this reason, the reconstruction approaches presented in this work are compared in terms of performance to the cellular automaton algorithm implemented inside the LHCb framework.

3.2.1 The Cellular Automaton algorithm

A cellular automaton (CA) [40] is a computational method used to describe the evolution of a discrete system under a set of rules through discrete steps in time. The system is defined as a grid of cells of any dimension where each cell can have a finite number of states. At each time step, every cell updates simultaneously its own state depending on the state of its neighbor cells following a defined set of rules.

Based on this principle, the cellular automaton algorithm [38] has been the baseline solution for the ECAL reconstruction through Runs 1 and 2. It comprises three different steps. The first one consists of a local maxima finder to identify the potential centers of clusters, called *seeds*. A digit is considered a local maxima if it has the highest energy value among its eight adjacent neighbors. In order to reduce noise, only digits with at least 50 MeV of transverse energy can be considered a cluster seed. Once a seed is identified, it is tagged with a unique identifier. The second step involves a proper cellular automaton to expand the cluster tags of the seeds to the neighbouring digits. The propagation rules are as follows:

- A tagged cell does not evolve any more.
- If a cell is not tagged, it checks the status of its neighbors:
  - If none of the neighbors are tagged, no action is done.
  - If only one neighbor is tagged, the cell adopts the same tag.
  - If several neighbors are tagged with a unique tag, adopt that same tag.
  - If several neighbors are tagged with different tags, the cell is identified as shared by several clusters and all the tags are stored.

Figure 3.4 shows an example of the described clustering process where the numbers represent the energy digits and the tags are marked as colors.

The final step consists of an iterative algorithm that resolves the shared cell cases, from now on known as overlapping cells. The implementation of the overlap solver in the Cellular Automaton reconstruction algorithm involves computing the energy separation for every pair of clusters that are overlapping. The separation is computed as a fraction of the overlapping energy assigned to each cluster, following Equation 3.1.

$$\text{fraction}_{\text{cluster1}} = \frac{E_{\text{cluster1}}}{E_{\text{cluster1}} + E_{\text{cluster2}}} \tag{3.1}$$

Once the overlap fractions are computed, the total energy of the involved clusters changes according to the fraction of the overlap energy assigned to them. Then, one can compute again the overlap fractions with the updated energies of the clusters. This iterative process is computed up to five times or until a defined convergence criteria is fulfilled. The stopping condition evaluates the energy resolution difference between iterations: if the energy has

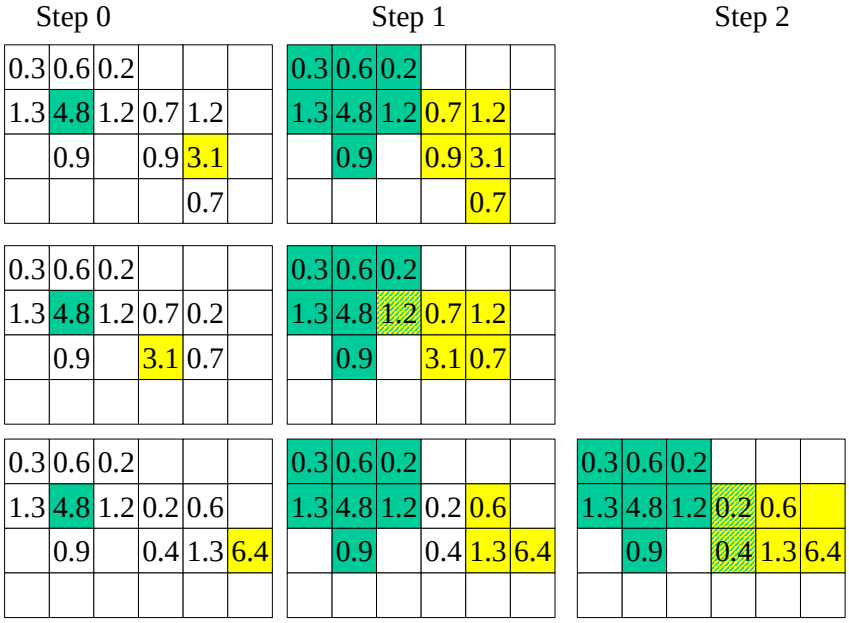


Figure 3.4: Example of digit clustering around local maxima cells. Etched cells are identified to be shared by two clusters [38].

changed less than 1%, the overlap loop stops. The position of the clusters in X and Y coordinates is also evaluated weighted according to the energy of each digit. Therefore, it will be affected by the overlap fractions and is also evaluated in the stopping criteria in the same way as the energy.

The definition of the cluster shape depends on the definition of the neighbors of a cell. In the boundary of two ECAL regions, there will be neighboring cells of different size, therefore a specific neighbourhood criteria is defined. The Cellular Automaton defines the neighbourhood of a cell as all the cells sharing either a side or a corner with it. As an example, Figure 3.5 represents a seed in the Outer ECAL region with 3.2 GeV, in the boundary with the Middle ECAL region. Its neighbours are the cells with energy deposits of 0.4, 0.2, 2.1, 0.6 GeV. As a result, the cells with 2.1 and 0.6 GeV deposit are identified as overlapping between the green and the yellow cluster and the cells with 0.2 and 0.4 GeV are identified as belonging only to the yellow cluster. The mentioned neighbourhood definition is included in the geometry description of the ECAL cells.

After this step, the reconstructed clusters are retrieved as all the digit entries that belong to a cluster with its own fraction. If a digit does not overlap, the fraction is set to 1. The final energy and position of a reconstructed cluster are defined as:

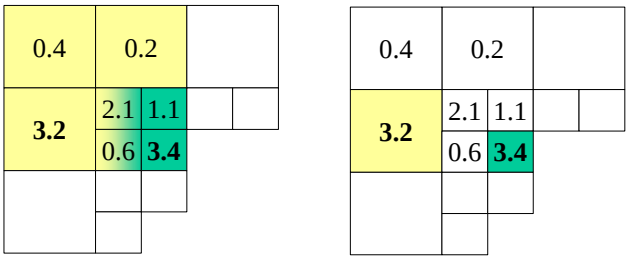


Figure 3.5: Example of Cellular Automaton clustering in a boundary region [38].

$$E_{cl} = \sum_{i=0}^{entries-1} f_i E_i, \quad x_{bar} = \frac{1}{E_{cl}} \sum_{i=0}^{entries-1} x_i E_i, \quad y_{bar} = \frac{1}{E_{cl}} \sum_{i=0}^{entries-1} y_i E_i, \tag{3.2}$$

where E_i is the energy of each cell entry from the cluster, f_i is its contribution fraction to cluster cl and x_i and y_i are its x and y coordinates. Therefore, E_{cl} corresponds to the total energy of the reconstructed cluster and x_{bar} and y_{bar} are the barycenter positions weighted by the cells energy.

In order to correct the biases induced by the detector inefficiencies and non-linearities, three different corrections are applied to the energy and position measurements for each cluster [41]. Summarizing, the S-shape correction takes into account the non-linearities of the transversal profile of the shower to correct a bias in the energy barycenter, the L-correction improves the true photon position by correcting for the penetration depth of the shower, and the incidence angle correction corrects the reconstructed position from the incidence angle dependency. Additionally, an energy correction is applied to minimize the energy bias between the true energy of a photon and the reconstructed energy from a photon cluster.

3.2.2 Merged π^0 clusters

One of the reconstruction requirements for the LHCb calorimeter is the correct identification of neutral pions, π^0, which decay into two photons before reaching the calorimeter. Depending on the energy and momentum of the π^0, the two photons arrive at the calorimeter with a certain separation.

If the seeds of the two photons are distanced more than one cell, they will be reconstructed as separate clusters. This case is called a resolved π^0. Otherwise, the two photons may travel very close to each other and reach the calorimeter at one cell distance or less. In that case, the reconstruction is done as a single cluster, since the definition of maxima does not allow two adjacent cluster seeds. When photons are not separable, it is then called merged π^0 case. Hence, the super-cluster from a merged π^0 can be bigger than the 3×3 window around the seed as can be seen in Figure 3.6.

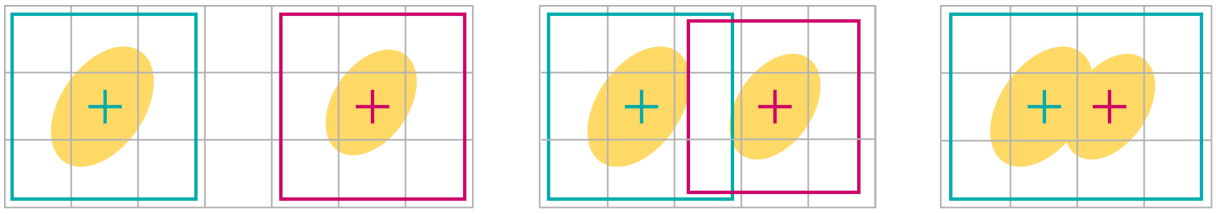


Figure 3.6: Diagram representation of π^0 cluster cases on the calorimeter. From left to right: the two photons are separable and without overlap, it is a resolved π^0. The two photons are separable but have three overlapping digits, it is however a resolved π^0. The two photons are not separable, it is a merged π^0 and is reconstructed as a single cluster bigger than 3×3.

There are other dedicated algorithms in the LHCb sequence that use the output of the calorimeter data reconstruction, together with other detector data, to properly identify and classify π^0 particles. The cluster shape used in these cases is a mask of 5×5 cells around the seed [42]. Therefore, the residual energy outside the 3×3 window of a merged π^0 is crucial, as it contains part of the energy from the second photon.

3.2.3 Reconstruction approaches for other calorimeters

Among other calorimeter detectors in high energy physics experiments, the most similar to ECAL in LHCb is the electromagnetic calorimeter of the HERA-B experiment [43], built in the HERA proton accelerator at DESY, Hamburg [44]. It had a three region geometry with different granularity and employed the same shashlik technology of a sampling scintillator/absorber structure as in the LHCb's ECAL.

Its reconstruction code is focused in the isolation and identification of clusters from particles releasing at least 0.05 GeV transverse energy, to provide a reliable particle identification for electron-hadron discrimination. The conditions during the HERA-B running period involved about 100 particles per interaction, releasing a signal in more than 10% of the ECAL cells [43]. This implied a non negligible probability of particle overlapping, therefore, a good cluster separation is required in the reconstruction algorithm.

The HERA B hierarchical clustering algorithm [45] provides a fast reconstruction of an event where each digit cell is looked at only once. However, since it is based on a hierarchical clustering method [46], the clusters provided are tree structures made of digits and other clusters. This recursive definition of clusters allows to keep information on the distance between clusters at the expense of having a complex structure.

The defined algorithm starts with an empty top cluster list and an array of digit cells sorted in decreasing order of energy, then it starts a loop on the digits. For each digit, it retrieves the list of neighbouring clusters. If the list is empty, it starts another top cluster with that digit. If the list contains only one cluster, that digit is added to the cluster. If the list contains more than one cluster, a new complex cluster is started with that cell and the

clusters in the list, recursively accounting for the overlapping clusters. An example of the cluster representation in the hierarchical tree is shown in Figure 3.7.

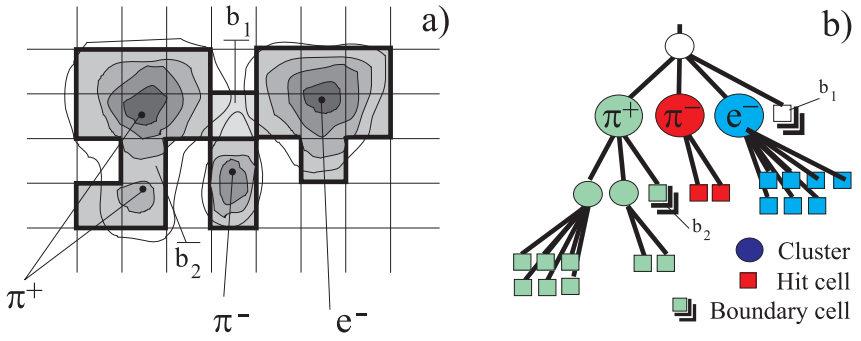


Figure 3.7: On the left, typical pattern that can be found in HERA-B ECAL with three partially overlapping cells. On the right, Hierarchical Tree representation of the left pattern [45].

Other experiments in the LHC also have calorimeter detectors but highly differ on technology and geometry with the ECAL in LHCb. The ATLAS experiment has a Liquid Argon (LAr) calorimeter that surrounds the inner detector with three layers of lead absorbers and a LAr ionization medium [47]. Therefore, the calorimeter clustering is transformed into the reconstruction of three-dimensional energy deposits from particle showers. The algorithm used in this case, consists on building topological clusters following spatial signal-significance patterns from the particle showers [48]. It is highly dependent on the cell significance, defined as the ratio of the cell signal to the average noise in that cell, estimated for each run period.

However, the individual topological clusters derived from the algorithm are not always expected to contain the entire response to a single particle. Rather, depending on the incoming particle types, energies, spatial separations and cell signal formation, these individual topo-clusters represent the full or fractional response to a single particle, the merged response of several particles, or a combination of merged full and partial showers.

In the CMS experiment [49], there is a homogeneous and hermetic electromagnetic calorimeter containing 61200 lead tungstate scintillating crystals mounted in three different sections [50]. Since there are layers of crystals, the energy clustering is also performed as a three dimension reconstruction.

The clustering algorithm proceeds first with the formation of "basic clusters", corresponding to local maxima of energy deposits in the calorimeter crystals. Then, depending on the region, all the basic clusters in a fixed-size window of 3×3 or 5×5 crystals are merged into *superclusters*. In the biggest region, the *barrel*, the basic clusters are expanded using the Hybrid algorithm for high energy electrons [51], which expands the supercluster taking a fixed bar of 3 to 5 crystals in consecutive layers while dynamically searching for

neighbouring deposits in the azimutal angle ϕ. For less energetic deposits, the Island algorithm is used [51], which first searches for neighbouring crystal deposits in ϕ and then in the adjacent layers. With this algorithm, clusters are expanded until a noise-level deposit is found or until another basis cluster is found.

Also in the LHC collaboration, the ALICE experiment [52] has an Electro Magnetic calorimeter (EMCal) which is a layered lead (Pb)-scintillator sampling calorimeter with shashlik technology [53]. The cluster reconstruction strategy used depends on the analysis goals, however, all the algorithms start building clusters from the highest energetic cell in a region, referred to as seed cell. Then, neighbouring cells are associated to the cluster according to different threshold parameters. Variations of this method also allow for overlapping cells between clusters, with a different fraction of its energy associated to each cluster [54].

As seen in the literature, the baseline reconstruction strategy for most calorimeter detectors is based in finding cluster seeds of local maxima energy deposits and further expand the cluster around the seed according to specific geometric and performance conditions. However, since the LHCb's ECAL is a two-dimensional calorimeter, the techniques used for layered calorimeters require adding a third dimension to the problem, transforming the reconstruction into a tracking-like problem.

Chapter 4

Calorimeter Time Alignment

One of the key tasks in the calorimeter commissioning for Upgrade I is the time alignment of all the channels from the electromagnetic calorimeter (ECAL) and the hadronic calorimeter (HCAL). It consists on tuning the phases of the ICECAL integrator and ADC of every channel to minimize the spillover and ensure the maximum energy of the particle shower is being digitized. This is achieved when the ADC captures the signal from the integrator at its maximum before the discharging phase.

Time alignment is studied with the first collisions data using dedicated runs called Time Alignment Event (TAE) runs with isolated signals that record the signal from several consecutive bunch-crossings in the same event. The central sample, labeled as *Current* or BX0, concerns the 25 ns window where signal from a collision event is expected and is the only bunch-crossing which is kept in standard runs. The other samples of the sequence, labeled *Previous1* or BX-1 and *Next1* or BX+1, correspond to the signal recorded 25 ns before and 25 ns after the central one. There are two main sources of misalignment that can affect the calorimeter. The first one is due to the variation of the high voltage from the PMTs, where the difference in the signal collection time inside a PMT follows a square root dependence with the applied high voltage. The second source of misalignment is due to the variation of the FEB configuration values such as the pole zero filter, which can affect the shape of the signal and thus its timing.

Although a time alignment procedure was developed and validated through Runs 1 and 2 [55, 56], the new software trigger and new calorimeter electronics for Upgrade I require an adaptation of the method and re-validation through simulation and the commissioning phase of the calorimeter system.

4.1 Run 1 and 2 method

The signal of a PMT collecting light from a collision in an ECAL or HCAL module is entirely contained in a 25 ns window. Figure 4.1 shows the distribution of the integrated signal after the clipping using Run 2 electronics. It can be seen that there is a plateau of about 2 ns

around the maximum of the integrated signal. Within this range, phase changes of the order of 1 ns or 2 ns generate very small fluctuations of the order of 1% of the signal between bunch-crossings that are difficult to measure. Instead, exploring the mid-height of the integrator signal augments considerably the sensitivity to any misalignment. For example, we can define the relation between signal in consecutive bunch-crossings as:

$$A = \frac{E(Current) - E(Next)}{E(Current)}. \tag{4.1}$$

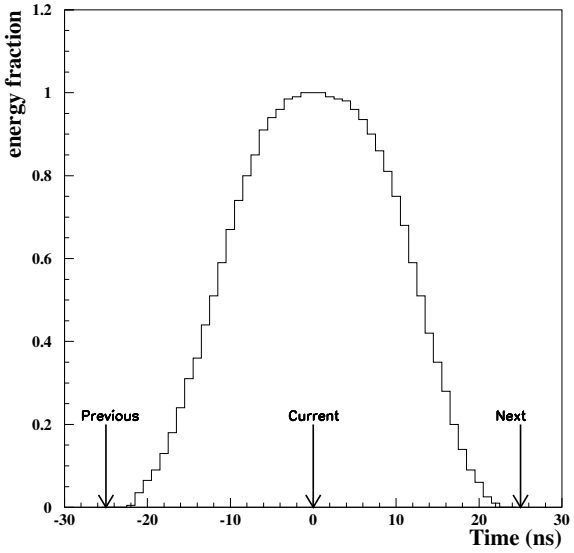


Figure 4.1: Distribution of the fraction of the total energy deposit as a function of time, obtained on test beam data [55].

In a simulation sample where the integrator phase is 12 ns we have $A = 30\%$, whereas if we move the integrator phase to 13 ns we have $A = 4\%$. Therefore, this method exploits the mid-height sensitivity of the integrator signal by measuring an asymmetry R_j between bunch-crossings for each calorimeter cell when the signal phase is shifted 13 ns.

The asymmetry R_j is defined as

$$R_j = \frac{1}{N_{evts}} \sum_{i=0}^{Nevts} \frac{E_{ij}(Current) - E_{ij}(Next)}{E_{ij}(Current) + E_{ij}(Next)}, \tag{4.2}$$

for each calorimeter cell j. Using simulation data, a set of samples were studied by computing R_j in different known phases Δt from -25 ns to 25 ns with a 1 ns step. When plotting the asymmetry measured by a cell as a function of the delay, the asymmetry curve is obtained. The region around $R_j = 0$ is approximated with a linear regression to obtain the relation between asymmetry and misalignment shown in Figure 4.2. Given that the aim is to achieve

a cell alignment with a precision better than 0.5 ns, a cell is considered to have a good alignment when:

$$0.05 < R_j < 0.20 \tag{4.3}$$

where 0.05 corresponds to $\Delta t = 13.5$ ns and 0.20 to $\Delta t = 12.5$ ns.

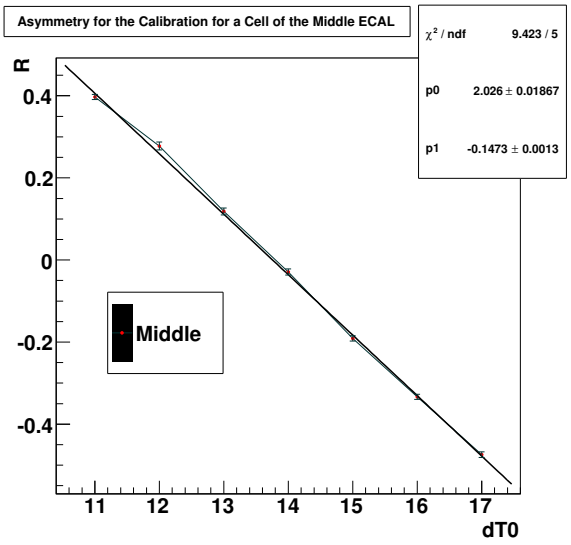


Figure 4.2: Asymmetry distribution for a particular cell of the middle ECAL [55].

Through the commissioning phase of the detector, the misalignment of each calorimeter cell needs to be calculated and corrected if necessary. To do so, p-p collision data is captured in the specific TAE configuration. When in this mode, a window of ±3 bunch-crossings are captured around the triggered bunch-crossing at the same time. Hence, the signal in several *Previous* and *Next* bunches can be checked for every triggered event in *Current*. The length of this window is a configurable parameter usually set to 3 to optimize the time alignment analysis and the throughput of the triggered events.

To determine the alignment of each ECAL and HCAL channel, the value of R_j is computed for every triggered signal within a channel and subsequently normalized by the number of events recorded in TAE mode. By averaging this asymmetry over a substantial number of events, any disparities in the arrival time of particles in the 25 ns event window are effectively mitigated. Subsequently, the asymmetry value is used to estimate the delay for each channel, allowing for the necessary adjustment of the track-and-hold clock by shifting it accordingly.

4.2 Adaptation to Run 3 conditions

In the Upgrade I two main changes affect the time alignment procedure. The first one is the update of the FEB electronics that include two sub-channels for each calorimeter cell. As mentioned in section 3.3, the two sub-channels need to alternate the integration and digitization of the signal. Therefore, a single phase is configured as the TH clock for one sub-channel and the other one uses the inverted phase clock. The TH clock serves also as the integrator clock as they have the same phase. The ADC clock is set to be shifted by precisely 1 ns with respect to the TH clock. This specific shift is determined to maximise the signal readout according to a phase scan performed in a laboratory with one FEB.

The second change concerns the readout chain of events through the online system. The raw data generated from the detector in a run is stored in what is called a *RawEvent*. The RawEvent can be considered as a mini event store, which contains heavily packed and optimised data as close as possible to what is actually shipped off of the detector. These packed data are known as *RawBanks* for which there may be several per sub-detector, and many different types. When taking data for Run 2, RawEvents were produced by the event builder, then sent through the trigger and finally converted to a Raw file. By analysing the Raw file of a TAE run, one could read a single event and access the *Previous* and *Next* digit values from the calorimeter cells around the *Central* BXID. By doing so, an asymmetry value per cell can be calculated for every event.

However, as mentioned in Section 3.1, Run 3 conditions require the complete event information at trigger level. Consequently, the assembly of all pieces of data belonging to the same bunch-crossing is done for every collision of non-empty bunches at 40 MHz rate in the event building process. In the online chain of algorithms that process the Raw files, events from a TAE run are processed sequentially starting from the left most bunch of the TAE window and are analysed independently. This forbids the direct access to consecutive events and complicates the persistence of information from one BXID to the next one even if they are events in the same TAE window.

Therefore, instead of computing the asymmetry on each event, we accumulate the energy deposits per BXID from the TAE window in a separate histogram for each calorimeter cell. Then, the asymmetry for each channel is computed using the accumulated signal in the *BX0*, *BX-1* and *BX+1* histogram bins through all the events of a run. Therefore, the asymmetry R_j is re-defined as

$$R_j = \frac{\sum_{i=0}^{Nevts} E_{ij}(Current) - \sum_{i=0}^{Nevts} E_{ij}(Next)}{\sum_{i=0}^{Nevts} E_{ij}(Current) + \sum_{i=0}^{Nevts} E_{ij}(Next)}. \tag{4.4}$$

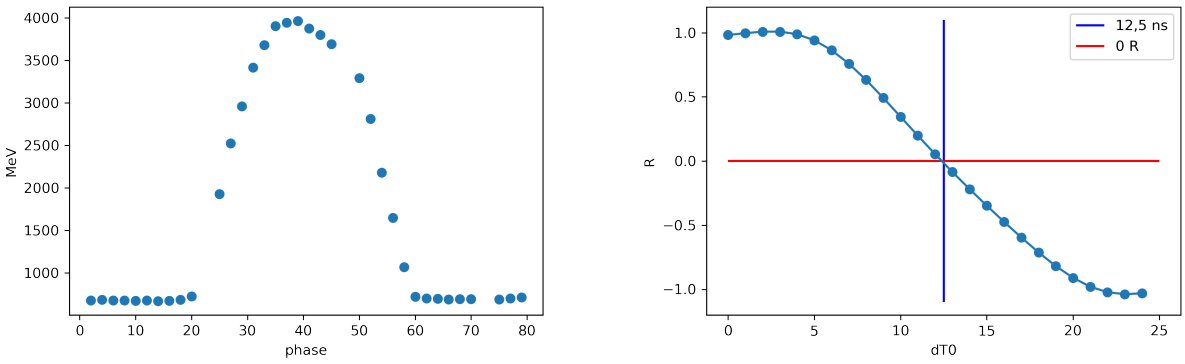
Given that the shape of the signal can be assumed to be the same for a single cell through different events, the relative proportion between BXIDs in the TAE window is independent of the number of events captured. As said in the previous section, the arrival time of a particle might slightly change the signal shape. This effect is mitigated by accumulating a significant amount of events before computing the asymmetry.

Given that the integrator signal shape is almost symmetrical, the asymmetry could also be defined as the relation between the signal at the *Previous* BXID and the *Current* BXID. In the presented study, the asymmetry is defined as in Equation 4.4 for consistency with the previous method.

4.2.1 The Asymmetry Curve

Due to the modifications in the calorimeter electronics and updates to the time alignment method, it is necessary to assess a new asymmetry curve. To compute it, we need to reconstruct the shape of the integrator signal using real detector data. Therefore, a set of scan runs were taken in stable beam conditions at 6.8 TeV in TAE ±3 mode. These scan runs involve systematically shifting the TH clock in increments of 1 ns covering the whole event window from 0 to 24 ns. Since the calorimeter is not time aligned at this point, the integrator signal reconstruction must be done for individual channels. Two selected channels from each calorimeter region are selected for this purpose.

For each scan run, we retrieve the energy deposits on each of the seven BXIDs from the TAE window and a relative phase according to the TH shift. The energy values are averaged within all the events in a run and plotted as a function of the relative phase to show the shape of the integrated signal. Before computing the asymmetry, the average pedestal is subtracted and the signal shape is normalized between 0 and 1. Then, the asymmetry R_j is computed following Equation 4.4 by sampling the normalized signal with a set of delays from 0 to 25 ns in a 1 ns step. The signal at *Current* is sampled at a given delay and the signal at *Next* is sampled at the same phase plus 25 ns. Figure 4.3 shows the reconstructed shape of the integrator signal at BX0 and the corresponding asymmetry curve for a single cell, where $dT0 = 0$ corresponds to the phase of maximum signal value from the integrator.



(a) Integrator signal reconstruction.

(b) Asymmetry curve.

Figure 4.3: Scan analysis for a particular inner ECAL channel.

Given that the phase of the channels has a 1 ns granularity, the null asymmetry value $R = 0$ is defined to be at 12 ns instead of 12.5 ns. The central region of the asymmetry curve, corresponding to $\Delta t \in [9ns, 17ns]$, can be approximated as linear. Therefore, the delay associated to a given asymmetry value is defined with the following linear function:

$$R = \alpha \times \Delta t + \beta \tag{4.5}$$

The averaged fitted values give $\alpha = -7.6 \pm 0.6$ and $\beta = 12.4 \pm 0.1$ for all the selected cells. Figure 4.4 shows the variation of the asymmetry R_j with Δt for the resulting fit compared to a given inner ECAL cell.

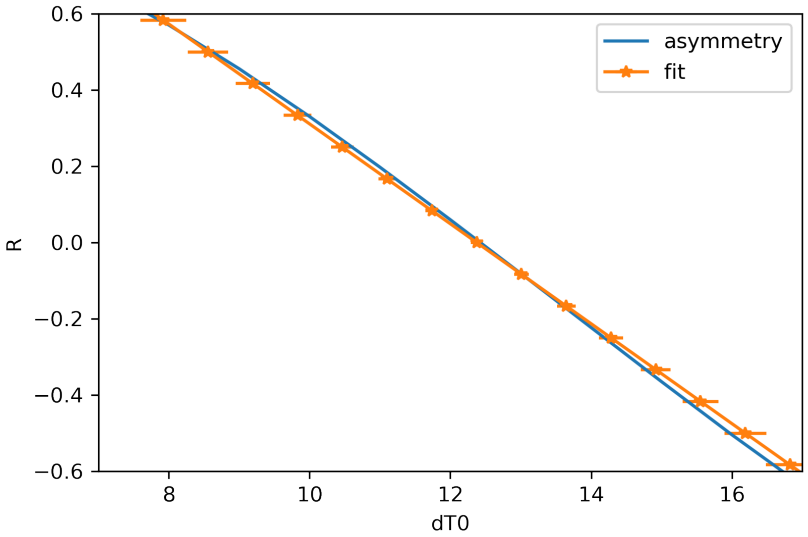


Figure 4.4: Asymmetry curve for a particular inner ECAL channel in blue, compared to the fitted average of the studied cells in orange.

In this case, the goal is still to achieve a cell alignment with a precision better than 0.5 ns. Therefore a cell will be considered to have a good alignment when:

$$-0.02 < R_j < 0.13 \tag{4.6}$$

where -0.02 corresponds to $\Delta t = 12.5$ ns and 0.13 to $\Delta t = 11.5$ ns.

4.3 Commissioning Process

During the calorimeter commissioning for Run 3 period, both ECAL and HCAL sub-detectors need to be time aligned from scratch.

4.3.1 Coarse Alignment

Given that the asymmetry calculation requires to have signal mostly in the *Current* bunch-crossing, a coarse alignment of the calorimeter with respect to the LHC collision bunches needs to be done. The bunch-crossing where the calorimeter sees data can be checked by looking at a 2-dimension histogram plotting the signal as a function of the bunch-crossing. Figure 4.5 shows an example of the mentioned histogram for the p-p collision run number 256274 triggering in calorimeter activity, where there is an isolated bunch seen in BXID 964.

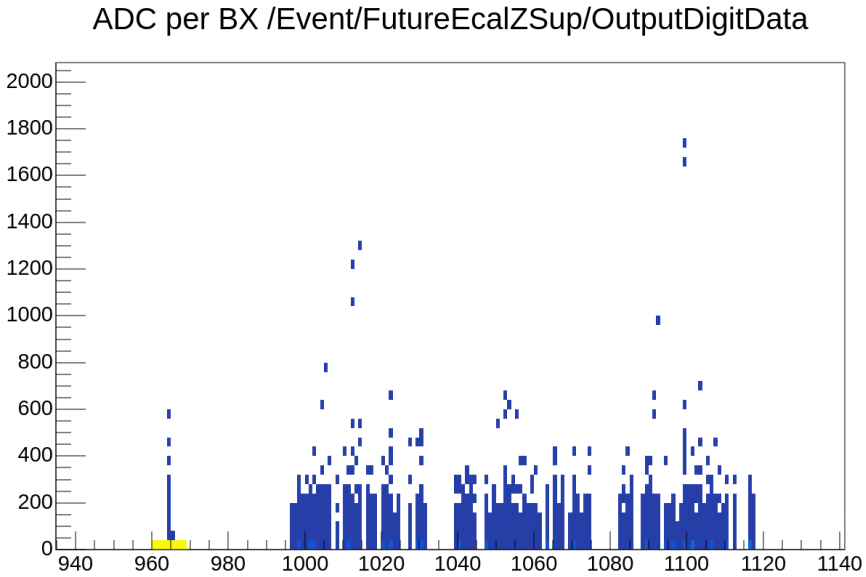


Figure 4.5: 2-dimensional histogram of the signal in ADCs as a function of the BXID for run number 256274, regular p-p collision run, triggering in calorimeter activity.

The difference with the BXID given from LHC is then corrected through the global configuration of the calorimeter at SOL40 level. The mentioned histogram and other visualizations of data are automatically generated using the Monet [57] online monitoring framework. With sufficient statistics triggering the events based on calorimeter activity in p-p collisions, the online monitoring allows to spot the BXID of the majority of the ECAL and HCAL signals just by looking at the plot. It also allows to quickly spot potential loss of alignment in real time.

4.3.2 Fine Alignment

After completing the coarse alignment, it is expected that the majority of ECAL and HCAL channels will be aligned with either the same BXID or within a range of ± 3 BXIDs from the expected value. For the fine alignment process, it is needed to ensure that all channels are precisely synchronized with the expected BXID and that data is captured at the peak of the integrator signal. This is achieved by using the method described in Section 4.2, where

data is collected in TAE mode during stable p-p collisions, specifically triggering on isolated bunches. A bunch is considered to be isolated if there are no other signals in the ±3 adjacent bunches.

Two different runs are indeed required, using different system configurations. The first one has the phase setting for regular collision data, where all the ECAL and HCAL channels should be fine aligned. In the second run, the phase of all channels is shifted 12 ns. Therefore, the signal is expected to be divided into two consecutive BXIDs when aligned. In this configuration, the asymmetry R_j can be computed to fine align each channel accordingly. The specific configurations for all of the channels of a particular run are stored in *recipes* that can be loaded into the hardware through the ECS.

Figure 4.6 show four examples of the plots extracted from the per cell histograms of TAE runs with the two recipes. In these plots the signal is accumulated as a function of the TAE window index, which can be extracted if the isolated BXIDs are known by simply subtracting the event BXID to the list of BX0s and choosing the one inside the TAE window range.

According to the asymmetry values obtained from the TAE window plots, the phase of the TH clock of each channel can be tuned at nanosecond level to properly align each cell. If the signal is not centered at BX0 in the default recipe, the latency parameter of each channel can be used to shift the signal an entire BXID without affecting the phase.

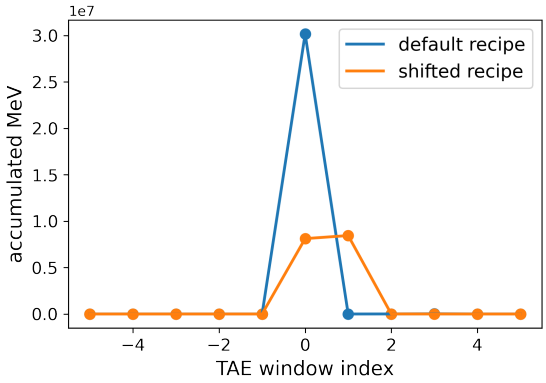
Several things have to be taken into account when estimating the delay of a channel according to the asymmetry values:

- If $-0.02 < R_j < 0.13$, the channel is considered to be well aligned as defined in Section 4.2.1 according to the maximum tolerance of 0.5 ns. In this case, no further correction is applied to the phase of that channel. Figures 4.6a and 4.6b show an example of two well aligned cells.

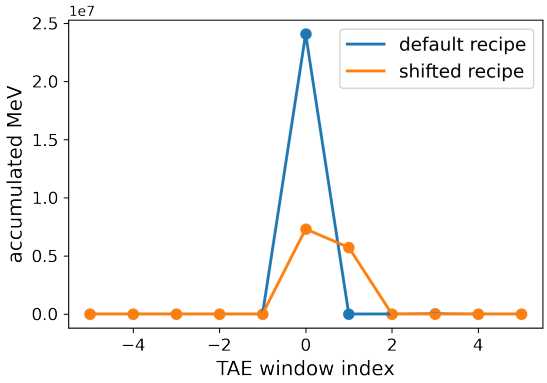
- If $-0.5 < R_j < 0.5$, excluding the previous range, the value is still in the linear range of the asymmetry curve. Therefore, an accurate correction for that channels phase is deducted from the fitted asymmetry curve. Figure 4.6c shows an example of a cell within this range where a phase shift of 3.2 ns should be applied. The delay is extracted as $f(R_j) - 12$ and must be added to the channels phase in the recipe, where f is the linear fit of the asymmetry curve.

- If $R_j < -0.5$ or $R_j > 0.5$, the value falls in the non-linear range of the asymmetry curve, hence, the delay extracted from the curve will not be accurate. Instead, a maximum delay of 8 ns is applied as an estimate for shifting the signal enough to have a better R_j when taking data again with the corrected recipe. Figure 4.6d shows an example of a channel in which the delay extracted from the curve would be 13 ns but shifting the phase more than half a period would be excessive. Instead, the phase is shifted 8 ns.

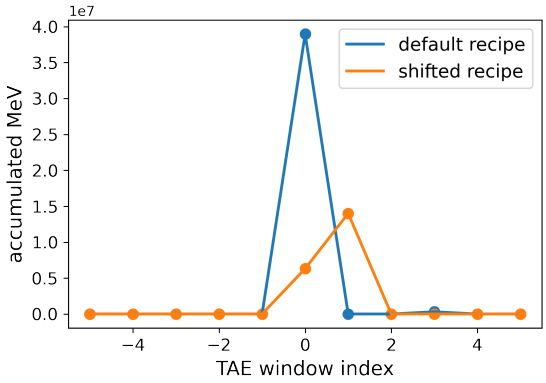
As already mentioned, the corrections to the phase of the channels may not be accurate in some cases. Therefore, the process of fine alignment of all the calorimeter channels is



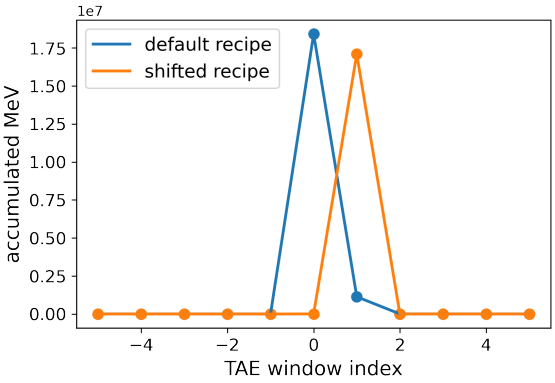
(a) Cell ID 10282. Asymmetry $R = -0.019$.



(b) Cell ID 9328. Asymmetry $R = 0.121$.



(c) Cell ID 9772. Asymmetry $R = -0.378$.



(d) Cell ID 11031. Asymmetry $R = -0.999$.

Figure 4.6: TAE window plots from two TAE runs taken with the default and shifted recipes in stable beams. Asymmetry R is obtained comparing the signal at BX0 and BX+1.

achieved after several iterations. Apart from that, from the 6016 + 1470 channels in ECAL and HCAL it has happened that some of them (less than 1%) are not well configured for a specific run or have a higher noise than expected. In these cases, other TAE runs are needed to ensure we capture a significant amount of signal in those channels.

As seen through the commissioning phase, the fine alignment procedure requires several iterations of data acquisition and correction computation to achieve convergence. The initial reason for this lies in the fact that an accurate correction can only be computed when the asymmetry value of a cell falls within a specific range, as previously mentioned. Therefore, at least two additional iterations will be needed in such case: an iteration in which the asymmetry value is in a valid range, and another one to validate the correction applied. The second reason concerns other detector-related activities that can potentially cause a partial loss of alignment in the ECAL and HCAL channels, which will be further explained in the following section.

4.3.3 Time Alignment maintenance

Through the course of the calorimeter commissioning for Run 3, the time alignment has been affected mainly by two circumstances. The first one is the update of the HV, which is done to calibrate the gain of the PMTs and can affect the timing of the channel. The effect on the channel alignment can be of a few nanoseconds depending on the voltage change. However, to re-align the channels after a HV update, a full TAE analysis is needed since the channels can have individual modulations of the HV and therefore different time shifts.

The second thing that has mainly affected the time alignment was due to a detected issue with the clock transmission inside the SOL40 boards. At the time a reset of a SOL40 board was done, the internal board clock was not properly locked with the input clock, therefore, it might stop at a different phase for each crate. Consequently, a small phase shift was detected in the ECAL and HCAL crates without affecting the relative alignment of individual channels within a crate. Whenever a SOL40 board was reconfigured, a TAE analysis was needed to evaluate the delay added to each crate and make the corrections accordingly. However, a SOL40 reset was only needed in very occasional situations, when there is a firmware update or when there is an issue with the communication with the LHC and the clock is lost. Moreover, this issue was solved by changing the way in which the clock is propagated inside the SOL40 boards and is no longer observed in the calorimeter commissioning.

Once a good fine alignment is achieved for all the calorimeter channels, it needs to be regularly checked during the data taking period in order to detect any misalignment. This can be done using the monitoring tools from Monet by checking the time alignment plots on isolated bunches when taking TAE data as shown in Figure 4.5. However, having an isolated bunch in the LHC filling scheme requires several empty bunches that induce a significant inefficiency in the number of collisions generated. Therefore, we cannot guarantee to have an isolated bunch in every fill. In order to still be able to account for the spillover of the channels to check its alignment, a Fake TAE plot is generated. This new plot takes advantage of the first and the last colliding bunches of the filling scheme, called *leading bunch* and *trailing bunch*. Since there are no colliding bunches after the leading bunch, the spillover on that empty bunch can be seen as *Previous* from a TAE window and the empty bunch after the trailing bunch can be evaluated as *Next* from a TAE window. With enough statistics as in the regular TAE data for the leading and trailing bunches, the time alignment of the channels can be evaluated properly without isolated bunch-crossings in the filling scheme.

4.3.4 Commissioning evolution

To provide an overview of the time alignment process's progression during the commissioning, Table 4.1 presents a summary of some of the time alignment steps including the average delay and latency applied to the channels. The table shows a gradual decrease in the mean and deviation of the delays across all sub-detector channels. This trend indicates that the channels are progressively achieving a state of improved alignment.

Date	Sub-detector	Avg. delay	Avg. latency
20-08-2022	ECAL	7.16 ± 7.01	0 ± 0.04
	HCAL	-11.84 ± 0.73	-1 ± 0.04
25-09-2022	ECAL	0.73 ± 7.71	0 ± 0.12
	HCAL	0.18 ± 7.13	0 ± 0.15
13-10-2022	ECAL	4.05 ± 6.34	0 ± 0.09
	HCAL	3.54 ± 5.99	0 ± 0.12
21-10-2022	ECAL	0.96 ± 5.04	-1 ± 0.06
	HCAL	1.35 ± 4.66	-1 ± 0.08
14-04-2023	ECAL	5 ± 0	0 ± 0.01
	HCAL	5 ± 0	0 ± 0
07-06-2023	ECAL	-1.7 ± 1.19	0 ± 0
	HCAL	-1.29 ± 1.13	0 ± 0.16

Table 4.1: Time alignment average delay and latency applied in some selected iterations performed during the commissioning process.

In addition to the changes in the HV and clock losses, which implied losses of alignment at cell and crate level respectively, there is a relevant set of corrections in Table 4.1 that is worth to highlight. On 14/04/2023, TAE data was taken during one of the first stable beam fills following the year-end technical stop (YETS) in December 2022. Upon analysing the data, it was clearly seen that all channels had a common phase shift, causing the majority of them to fall outside the linear asymmetry range. Since the phase shift could be attributed to either the SOL40 configuration or the LHC clock, the relative alignment of individual channels remained unaffected. Therefore, it was prioritized to make a global correction of 5 ns for all ECAL and HCAL channels and re-taking data for further alignment iterations rather than correcting individual channels with a clearly biased data-set. However, some channels had been replaced during the YETS, needing alignment from scratch. Consequently, latency corrections were defined for individual channels that were sitting in the wrong BXID. This serves as an example of the complexity and hand-craft behind the commissioning of the calorimeter time alignment.

An example of one of the better aligned configurations achieved is shown in Figure 4.7. This figure illustrates the alignment of all ECAL and HCAL channels, based on data acquired on 07/06/2023, using the default recipe for regular data acquisition. It is one of the latest TAE data taken to check the fine alignment status before proceeding to take physics data for Run 3. It can be seen that almost all of the channels are fully centered at the TAE window index 0, which corresponds to the *Central* BXID. There are however, some noisy channels which have signal on all of the window index. This can be related to a faulty channel that needs to be manually checked or a misconfiguration that happened for that specific run. On the other hand, Figure 4.8 shows the alignment status in the same LHCb fill but taken with

the misaligned recipe, where signal is expected to be split between TAE index 0 and 1.

4.4 Conclusions

The Time Alignment task for the ECAL and HCAL detectors in LHCb has been successfully addressed and validated through experimentation and analysis with detector data. The method used for Runs 1 and 2 was successfully adapted and validated in the LHCb Upgrade I conditions, demonstrating that the approach remains valid and effective.

Through the commissioning year in 2022 and the first half of 2023, the consistent and regular TAE data taking and analysis performed has yield to an accurate time alignment of the total of 7486 channels in the calorimeter system. As of the data taken at 30/06/2023, 90% of the ECAL channels and 97% of the HCAL channels are considered to have a good fine alignment within the criteria established in this chapter. Which is not far from reaching the expected 100% of the calorimeter channels fine aligned.

The time alignment commissioning process often uncovers hidden complexities within the detector operation, and encountering these challenges served as a valuable learning experience. By facing and overcoming these difficulties, we gained a deeper understanding of the detector's behavior and performance, helping in the identification and debugging of various issues.

Indeed, the time alignment task remains an ongoing and essential aspect of the detector's operation, since regular checks are required through the Run 3 operations to ensure an optimal alignment of ECAL and HCAL channels. To facilitate this continuous process, a robust analysis framework has been established. Moreover, global and per-crate time alignment plots for both detectors are currently monitored in Monet for regular data taking, including a general time alignment plot for filling schemes without isolated bunches. Although this is an easy tool to spot any time alignment issue in real-time, dedicated analysis would be needed in case the alignment needs to be corrected.

In conclusion, while the commissioning process brought forth its share of difficulties, the time alignment method has demonstrated its effectiveness achieving an overall good fine alignment state for all of the calorimeter channels, contributing to the continued success of the LHCb experiment in acquiring precise and reliable data.

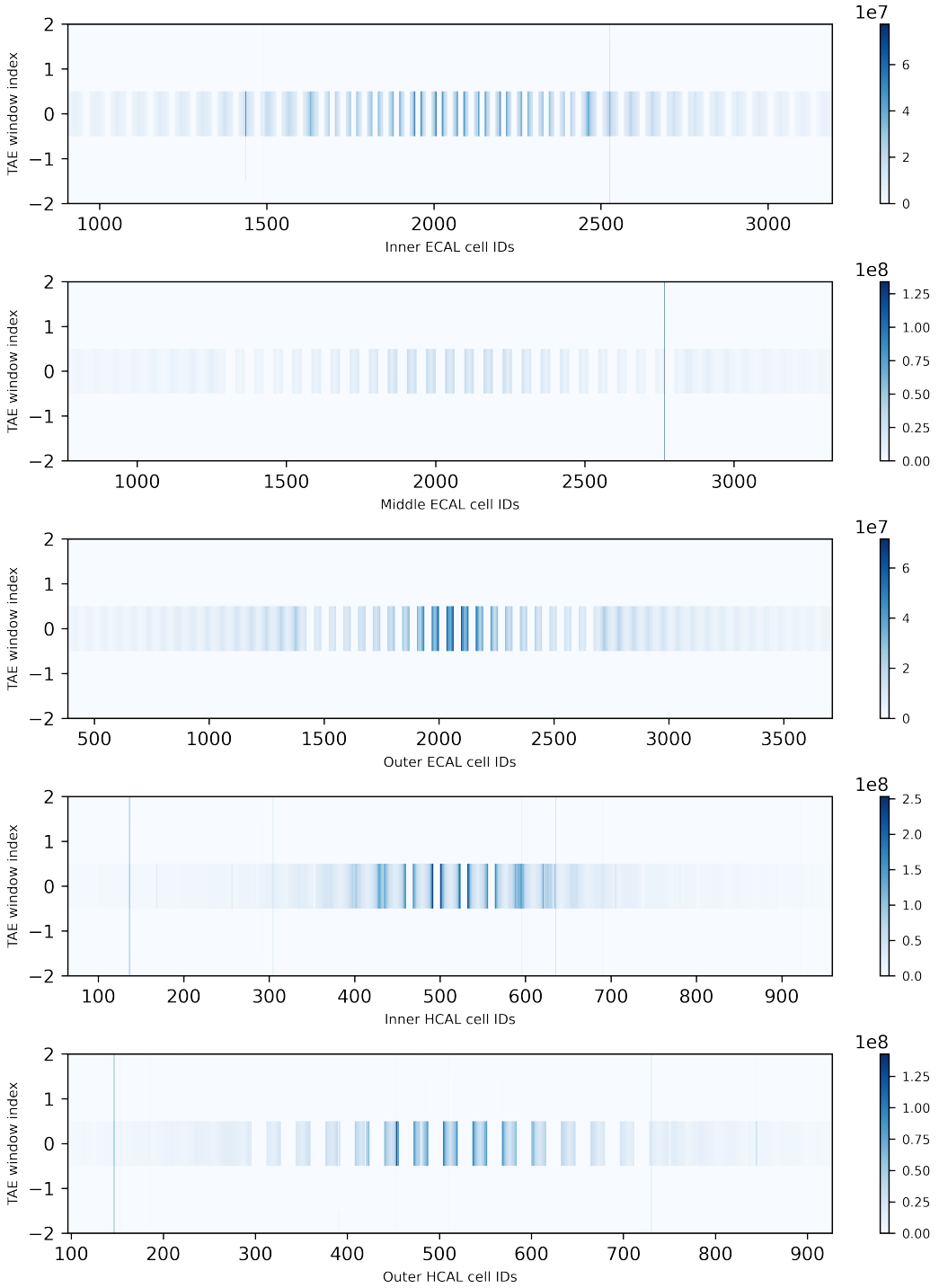


Figure 4.7: Time alignment status for all ECAL and HCAL channels using data taken at 07/06/2023 with the default, aligned for regular data taking.

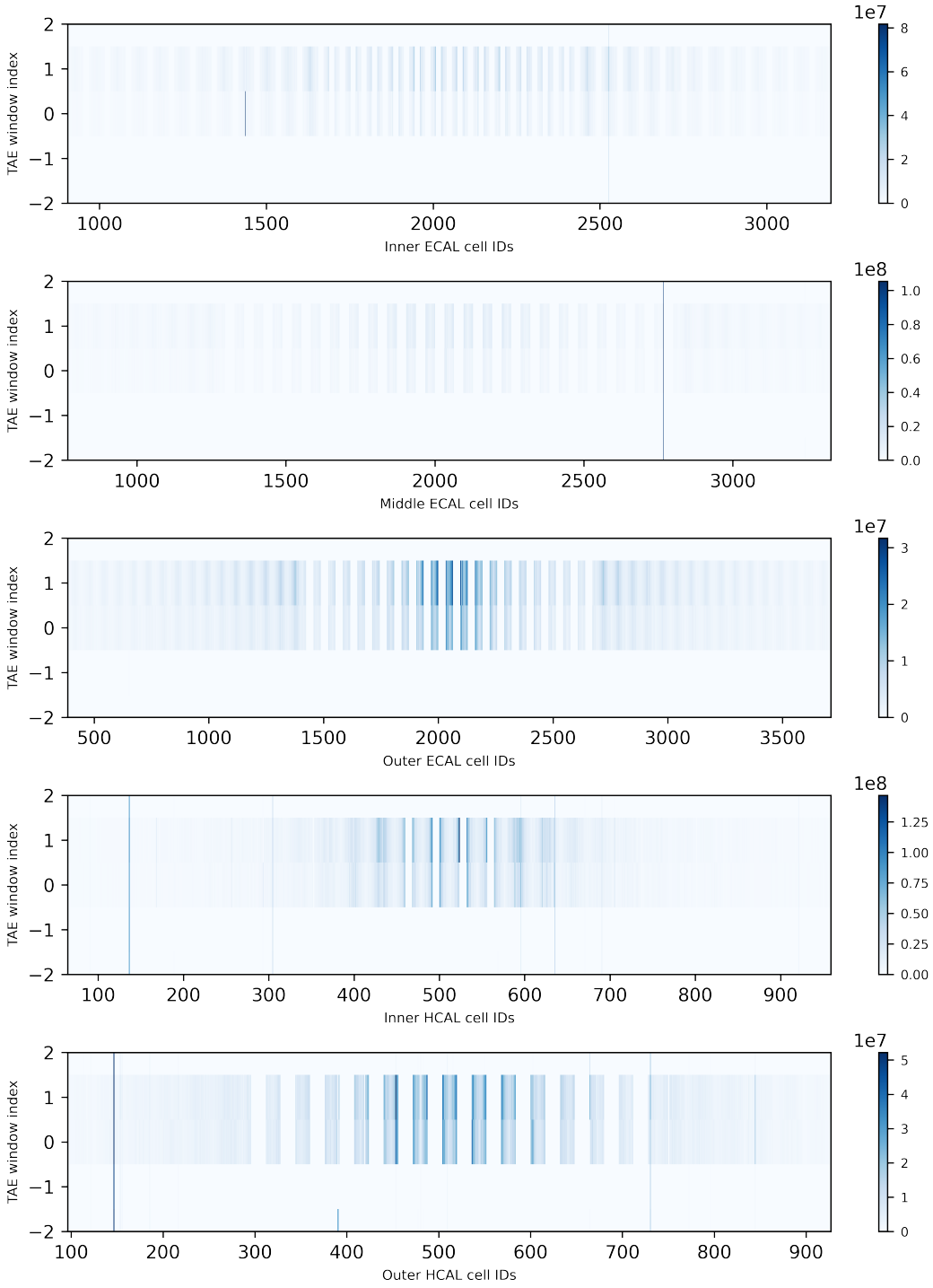


Figure 4.8: Time alignment status for all ECAL and HCAL channels using data taken at 07/06/2023 with the misaligned recipe, with all channels phases shifted 12 ns.

Chapter 5

Deep Learning reconstruction

As mentioned in Chapter 1, the LHCb Upgrade I implies a significant increase of the input event rate in the High Level Trigger. At this point, the optimization of reconstruction techniques is needed. Moreover, the vision of future upgrades enhances the importance of developing scalable and flexible software methodologies. In the case of LHCb, the time performance analysis of HLT2 algorithms before Run 3 points the calorimeter reconstruction as the fourth most computational expensive process, representing around 15% of the LHCb reconstruction time. Consequently, this chapter, along with the subsequent ones, focus on the study of alternative algorithms to optimize the calorimeter reconstruction.

In this chapter, a deep learning formulation of the LHCb calorimeter reconstruction problem is proposed. It makes use of deep learning techniques together with the understanding of the current reconstruction algorithm to propose a method that decomposes the reconstruction process into small parts that can be formulated as a cellular automaton. This approach is shown to benefit the generalized learning of small convolutional neural network architectures and also simplify the training data-set. Final results applied to a complete LHCb simulation reconstruction are compatible in terms of efficiency, and execute in nearly constant speed regardless of the event complexity [58, 11].

5.1 Background

The calorimeter reconstruction challenge, under very tight execution time constraints, invites one to think of neural network techniques that can provide the capability of learning complex problems and a fast inference. Considering its increasing popularity in recent years, there have been many improvements in the optimization of reconstruction algorithms. Deep learning models in particular, are able to solve many complex issues at very high speeds at the cost of increasing the time and complexity of its training in most cases. However, such proposals usually approach the whole scenario at once, forcing the networks to process hundreds of thousands of data samples and understand their insights, to be able to provide a complete solution at the output. As problems become more challenging, deeper networks

need to be trained with more data. In the case of HEP scenarios, the data used to train and test reconstruction algorithms is obtained through very time-consuming Monte Carlo simulations. As a result, obtaining large quantities of data for neural network training is not a resource-free process.

The benchmark algorithm used for the calorimeter reconstruction in Runs 1 and 2, detailed in Section 3.2, is based on a cellular automaton (CA). Due to the typical square-shaped modular structure of the ECAL, its geometry can be easily mapped into a two-dimensional grid. Therefore, the cellular automaton strategy has long been used in calorimeters for high energy physics [59, 60]. Such method provides an efficient reconstruction of the clusters, although the classical formulation of the cellular automaton requires several iterative processes along the energy digits that are programmed as loops in the algorithm's code. As this causes a strong dependency of the algorithm's complexity on the number of clusters and digits in the data, other approaches have attempted to avoid it by exploiting the architecture similarities between cellular automata and neural networks [61]. The approach presented in [62] defines a cellular neural network implementation for the identification of energy peaks in a general structure of a 2-dimensional detector. However, the mentioned approaches tend to focus on a proof of concept rather than providing a specific reconstruction solution for ECAL.

Within recent years, the evolution of deep learning models has encouraged the use of image processing techniques for this challenge. It has been shown that convolutional neural networks [63, 64] can achieve a good performance in calorimeter clustering solutions for the ATLAS detector [65], which has a three-dimensional layered calorimeter. An early stage project has shown promising results when approaching the LHCb calorimeter reconstruction with a deep neural network structure [66] based on the YOLOv3 network [67]. This particular case uses the order of $100,000$ simulation samples to train a network of the order of 65 million parameters.

The following sections are devoted to explain the insights of a new deep learning algorithm for ECAL reconstruction. The proposed approach comprises a specific formulation of the calorimeter reconstruction problem that benefits from small neural network architectures trained, in part, with artificially generated data.

5.2 The method

In this section, the deep learning proposal is explained in detail, starting with an explanation of the fundamental principles applied.

5.2.1 Fundamentals

The current implementation for data reconstruction in the LHCb calorimeter consists of a cellular automaton based clustering [38]. Summarizing the definition in Section 3.2.1, this cluster reconstruction can be segmented into three different sequential steps. The first one

consists of a local maxima finder, to identify the potential cluster seeds. The second step is the proper cellular automaton, which iteratively tags each digit to the closest maximum and enhances the overlapping cases. The final step consists of an iterative algorithm that performs the separation of those overlapping clusters.

Each of the three steps are indeed iterative algorithms that analyze the grid of calorimeter readout cells using a set of specific rules to give a certain condition in the end. Going further, all steps can be defined as a CA. Given the universality theorem of the CAs, there exists a set of rules and a set of states that can model the behavior of the mentioned algorithms as a dynamic system.

This approach is based in modeling each of the three steps of the reconstruction process as an independent CA with an ad-hoc formulation. With this, we can benefit from the fact that convolutional neural networks have been proven to learn the generalized rules of cellular automata [68]. Hence, we will train a convolutional neural network with the rules of each of the reconstruction steps to build a pipeline of networks that perform the full reconstruction of the ECAL.

5.2.2 Local maxima formulation

Starting with the local maxima finder, we want to identify cells that are local maxima among its neighbors. Hence, the cellular automaton characteristics can be defined as follows:

- States: two. One (1) if the cell is a maximum and zero (0) otherwise.
- Neighborhood: eight cells. In order to check if it is a local maxima, the surrounding cells at distance one need to be checked. In a two-dimensional grid of cells, the number of distance one neighbors is eight.
- Rule-set function: to define the condition of a cell to be a maximum (1) its value needs to be higher or equal to its neighbors. Although the equal condition is not obvious, it is needed for the case of merged π^0s. Since two photons from a π^0 could leave adjacent readout cells with the same energy, the formulation is adapted to include both as a seed. Therefore, Equation 5.1 defines the rule-set where $c_{i,j}^t$ stands for the value of each cell at time t and the case $M = 0, N = 0$ is excluded. The initial states of the grid cells concerning $t = 0$ are the values of the calorimeter readout cells.

$$c_{i,j}^{t+1} = \begin{cases} 1, & \text{if } c_{i,j}^t >= c_{i+M,j+N}^t \text{ for } M \in \{-1,0,1\},\ N \in \{-1,0,1\} \\ 0, & \text{otherwise.} \end{cases} \tag{5.1}$$

By looking at the function, it can be seen that the only operation is a comparison between the value of the central cell and one of its neighbors repeated eight times. Yet this comparison will perform the same way with independence of the values we have to compare. Hence, there

is no dependence on the numerical scale value of the cells in the application of this rule-set function.

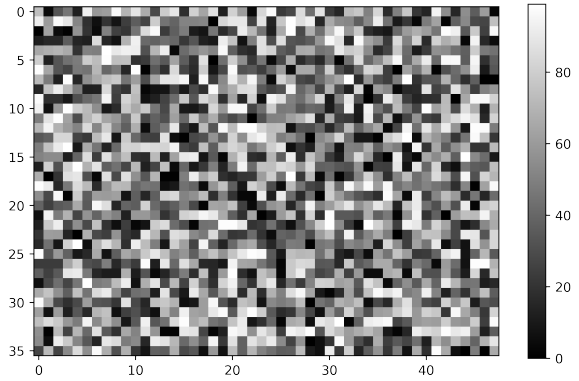
Consequently, the data-set used for the training of the local maxima finder network needs to reflect significantly the two possible cases of the rule-set function in any numerical value scale. To achieve that, the input test samples are generated as a two-dimensional grid of the same size as one of the ECAL regions, with uniformly distributed random values from 0 to 99. The range of values was chosen, taking into account the statistical number of ones that appear on a sample. The maximum number of local maxima that could fit in a sample is one fourth of the total readout cells. Then, we consider one sixth of the total cells as a good estimation on the number of local maxima that can be in the artificially generated samples to make sure that some local maxima happen to be adjacent in some cases, simulating the merged π^0 scenario. The random generation with values from 0 to 99 happen to match these conditions. Once the random samples are generated, the expected output test samples are computed by applying the rule-set function defined in Equation 5.1 to all the cells of the input samples.

As a visual example, Figure 5.1 shows one input sample and the respective expected output from which the network is trained. Given that we want the network to learn to reproduce the CA rule-set on the ECAL data, the testing data-set is made of LHCb Monte Carlo simulation samples of the ECAL in Upgrade I conditions, with the corresponding digit values. The expected output of the testing samples is obtained again with the application of the defined CA to the training input samples. Figure 5.1e shows the output generated by the trained neural network when it is given Figure 5.1c at the input. Compared to the expected output (Figure 5.1d), the differences are minimal, and it is shown that the trained network has effectively learned to extrapolate the comparative knowledge to other numerical ranges.

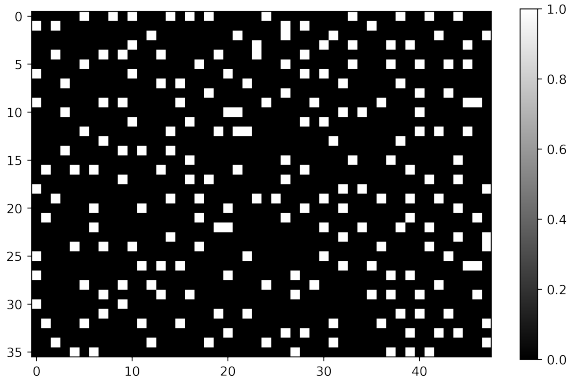
Given that the three regions of the calorimeter have different cell sizes, samples from each region have different shapes. Hence, we will train one network for each region. All of them have the same structure that consists of a two-dimensional convolutional layer followed by two/three dense layers, depending on the region, and finally, an output dense layer of two neurons, since the network is trained as a classifier understanding the output class as the state of a cell in the CA formulation. All layers of the networks have a ReLU activation function [69]. Table 5.1 shows a summary of the network parameters and characteristics obtained in the training. To evaluate the network performance, we will use the accuracy metric, since we want to maximize the number of correct classifications. The accuracy is measured as the number of correct classifications over the total number of cells.

5.2.3 Clustering formulation

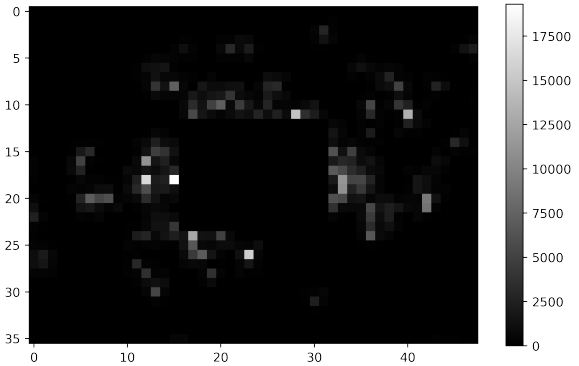
The following step of the reconstruction process is the proper clustering, which is already formulated as a cellular automaton in the classical implementation. In this case, the algorithm needs to identify the cells that belong to a cluster including the overlap cases, in which a cell can belong to more than one cluster. If a cell is not identified as a seed and is not close



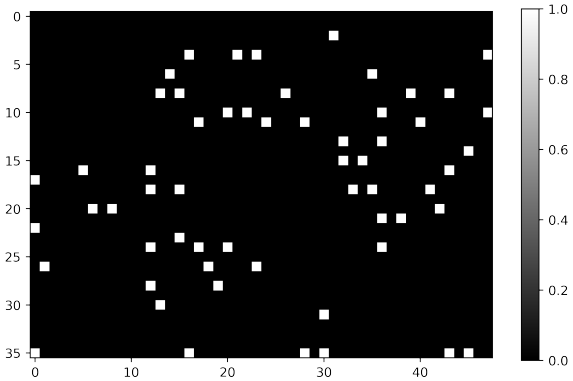
(a) Input sample for network training.



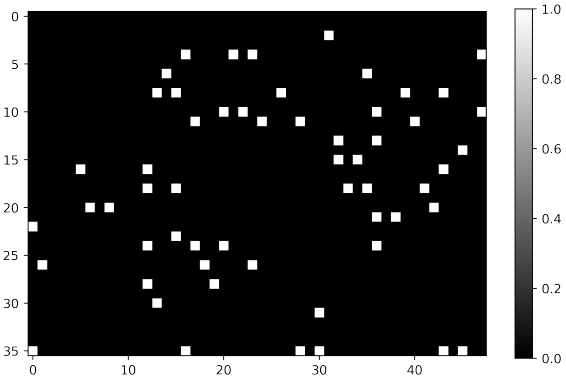
(b) Output sample for network training



(c) Input sample for network testing



(d) Output sample of the testing.



(e) Output sample of the network.

Figure 5.1: Samples of the data used for training and testing the local maxima finder network for the ECAL inner region. The training input sample (**a**) is artificially generated and the output sample for training (**b**) is obtained applying the CA rules on the (**a**) sample. The testing input sample (**c**) is a simulation and the testing output sample (**d**) is again generated applying the CA rules on sample (**c**). Then image (**e**) is the output obtained from the network when sample (**c**) is on the input, and is compared to sample (**d**) to obtain the accuracy value.

Region	Image Shape	Training Samples	Neurons per Layer	Parameters	Training Time	Accuracy
Outer	64×52	10,000	[20, 20, 20, 10, 2]	1272	1354.7 s	99.96%
Middle	64×40	10,000	[20, 20, 20, 10, 2]	1272	1052.3 s	99.92%
Inner	48×36	10,000	[10, 10, 10, 2]	342	461.8 s	99.93%

Table 5.1: Parameter summary of the local maxima finder neural networks.

in the neighbourhood of a seed, it is assumed to be a residual digit and will not be accounted as part of any cluster. In order to normalize the cluster shape, this approach assumes all the clusters to be of shape 3×3. The mentioned CA characteristics can be formulated as follows:

- States: three. Since we need to identify three types of cells: cells that belong to one cluster (1), overlapping cells (-1), and the rest of them (0).
- Neighborhood: eight cells. In order to differentiate cells as overlapping or belonging to a cluster from the others, the neighborhood at one cell distance needs to be checked.
- Rule-set function: in order to identify the cell states, the algorithm needs to check how many local maxima are in the neighborhood of a cell. Cells that belong to a cluster are the ones that have a single local maximum in its neighborhood. In the same way, overlap cases are identified when there is more than one local maximum in its neighborhood. If there are no local maxima in a cell neighborhood then it is either a local maximum itself or not relevant for the clustering. This is formulated in Equation 5.2, where K is defined as the number of local maxima in the neighborhood of a cell.

$$c_{i,j}^{t+1} = \begin{cases} 1, & \text{if } K == 1 \\ -1, & \text{if } K > 1 \\ 0, & \text{otherwise.} \end{cases} \tag{5.2}$$

Once defined the clustering formulation, we define the overlap solving algorithm. At that point, we realise that the design characteristics from this third step included all the requirements from the second step. Therefore, we propose to formulate the clustering and the overlap solver steps as a single CA.

5.2.4 Clustering and overlap formulation

For the last step of the reconstruction, apart from the cluster construction itself, the overlap algorithm needs to resolve the cases where a cell belongs to more than one cluster. To do so, the energy of the overlap cell needs to be distributed among the involved clusters, depending

on the total energy of each cluster. Since the energy measured in an overlap cell may come from the addition of two different particles, one fraction of the overlap cell energy is linked to one particle and the rest of the fraction to the other particle. Therefore, the desired output for this step is, given an input cell with overlap, the part of the energy designated to each of its contributing clusters. If a cell does not have overlap, the output needs to be the same energy value as the input.

To give a visual representation of a general overlap case, Figure 5.2 shows a representation of the calorimeter grid where the blue X marks one cluster seed. The eight neighbors of the seed, marked in blue, represent the cells belonging to the blue cluster. Those are the cells that can have overlap. We can distinguish two cases depending on the position of the cell with respect to the cluster. If the overlap cell is in a corner of the cluster, there are five positions where a seed could be in order to overlap with the given cell. As an example, the purple X's mark the seed positions that can cause overlap in the purple circled cell. In order to cover the distance-one neighbors from all the purple seed positions together with all the cells from the blue cluster, a window of 5×5 around the circled cell needs to be defined. The second case would be if the overlap cell is not in a cluster corner. Then, there are only three positions where a seed could be to cause overlap. In the same example of Figure 5.2, the second case is marked in green. It can be seen that in order to cover the blue cluster and all the clusters from the green seeds, the same window of 5×5 cells is needed.

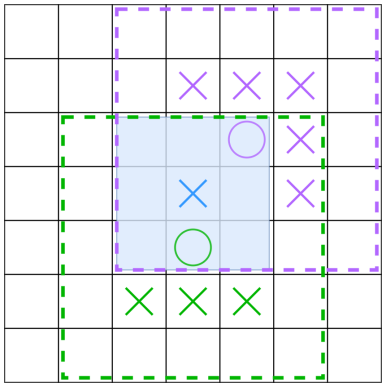


Figure 5.2: Diagram representing the possible cluster centre positions overlapping with the central cluster in a 7×7 window. The maxima positions are marked with crosses and the two overlap cells that need to be predicted are marked with a circle.

Once determined the shape of the window, in case of overlap, there would be a minimum of two seeds in the same frame. However, the energy distribution of the overlap cell is calculated for one of the involved clusters at a time. Therefore, an extra layer of information needs to be added to indicate to which seed the fraction is going to be assigned. The mentioned characteristics can be summarized as follows:

- The neighborhood window for this step needs to be big enough to identify all the cells

belonging to the possible clusters causing overlap. This number corresponds with 24 neighbors inside a 5×5 square window around the given cell, as can be seen in Figure 5.2 as the dashed squares.

- Extra information needs to be added to the 5×5 window, indicating the cluster to which the energy fraction is going to be part of. Since each fraction of energy needs to be assigned to a certain cluster, in the overlap cases, the same 5×5 window must be evaluated as many times as the number of clusters involved, but each time selecting a cluster that the fraction is going to contribute. This selected cluster, within a 5×5 window, will be called central cluster.

In order to provide the central cluster information, before transforming the input image into blocks of 5×5 windows, there is a first transformation into windows of 7×7. Within this 7×7 window, we can identify if the cell located at the centre is a local maximum and generate another stream of data to indicate that, for this particular window, the central cluster is the one in the middle of the window. This is represented as the entire diagram in Figure 5.2. Regarding the identification of the central cluster, this data is generated with a masking of the local maxima stream of each 7×7 window. Where the mask is a 7×7 matrix of zeros and a single one on the central position where we expect to find the central cluster.

Once we include this third stream of information in the window of 7×7, we can subsample all nine possible 5×5 windows that can fit inside the 7×7. At this moment, we have, on a single 5×5 sample, the data regarding the readout cells of the calorimeter, the local maxima information and the central cluster information. It is prepared to generate a prediction of the designated energy partition of the cell located at the centre of the window concerning the central cluster.

Gathering the previous concepts, the cellular automaton formulation of this step has the following characteristics:

- States: $10,240$. Since the algorithm needs to give a value concerning the energy of a cell as output, the CA must have enough states to model the full calorimeter sensitivity, which is of 12 bits on the ADC lecture with a gain of 2.5 MeVs per ADC value ($2^{12} \times 2.5$). Although there can be negative energy values in the calorimeter readout within the mentioned sensitivity, the presented approach is simplified to use only positive values.
- Neighborhood: 24 cells. As explained above, the window around a cell to predict its value needs to be of 5×5 cells.
- Rule-set function: at this point of the reconstruction, we have three streams of information:
  - Original data sample. Obtained from the calorimeter simulation with values from 0 to $10,240$.
  - Local maxima information. Obtained from the output of the local maxima finder network. With values from 0 to 1.

– Central cluster information. Obtained from the masking of the central cell on each 7×7 window from the image. With values from 0 to 1.

Then, the rule-set function for this step defines the fractioning of a cell's energy in case of overlap in Equation 5.3. In the same equation, *num_clusters* refers to the number of local maxima that could be causing overlap. As an example, if we look at the purple circle in Figure 5.2 to account for *num_clusters*, the positions marked with a purple X should be checked.

$$c_{i,j}^{t+1} = \begin{cases} \frac{c_{i,j}^{t} C_0}{\sum_{k=0}^{K} C_k}, & \text{for } K = num\_clusters \text{ if } K > 1 \\ c_{i,j}^{t}, & \text{otherwise,} \end{cases} \tag{5.3}$$

where

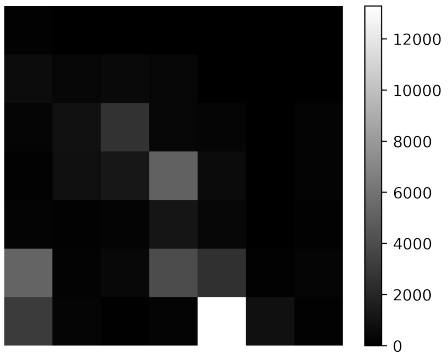
$$C_k = \sum_{m=-1}^{1} \sum_{n=-1}^{1} c_{o+m,p+n}^{t} \tag{5.4}$$

and variables o and p stand for the local maxima coordinates of cluster k in the 5×5 image. For $k = 0$ the cluster is specifically the marked as central cluster, the one in the center of the 7×7 window.

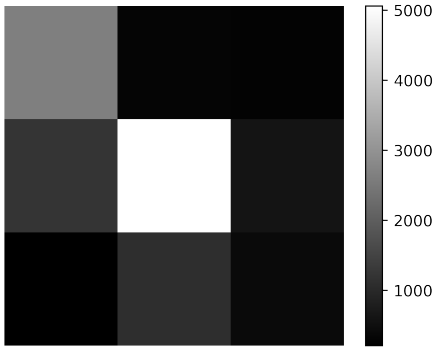
Before starting with the network training, there is a key aspect on the rule-set function that affects the learning capacity of a convolutional network, like the ones used on the first step. It can be seen that, for the second condition, there is a division that transforms the function into a non linear behavior. Following the universal approximation theorem [70], there needs to be at least one hidden layer to approximate non-linear functions. However, the previous used convolutional architectures had in fact two and three hidden layers, yet the convolutional layer itself does not have a hidden layer structure on the convolution operation. The convolution is performed through the multiplication between the data and a linear kernel of parametric values; hence, there is no chance for this convolution to be able to learn the desired non linear behavior before losing the neighborhood information on the dense layers. Therefore, the strategy for the network architecture is to use a multi-layer perceptron (MLP) structure and train it to be the kernel of the convolution.

Given that sampling into 5×5 windows normalizes the input beyond the different region granularity, a single MLP network can be trained for the three regions. In this case, the network architecture consists of four dense layers. The output layer is a single neuron representing the predicted value of the central input cell. Hence, the network is trained as a regressor. The output values need to be aggregated in groups of nine, concerning the predicted values from a cluster at all cells of the calorimeter. Figure 5.3 shows an example of the data used for the training of this network.

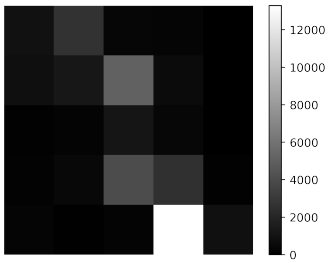
Regarding the training data-set generation, the same numerical value independence from the first formulation is observed in this rule-set function. However, in this case, the conditions of the function are not so simple to achieve in a homogeneous scenario using randomly generated samples. At this point, we take a set of only 2000 samples of LHCb simulation



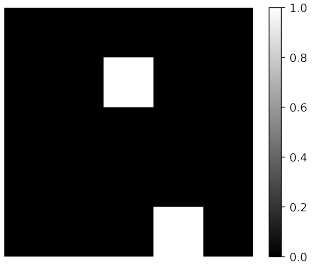
(a) Calorimeter 7×7 window.



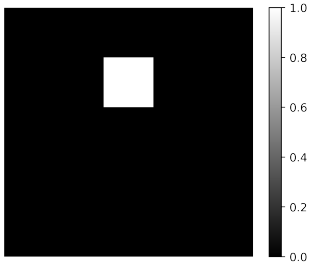
(b) Clustering network output.



(c) Digits sub-sample.



(d) Local maxima sub-sample.



(e) Central cluster sub-sample.

Figure 5.3: Samples of the data used for the training of the clustering and overlap network. Image (**a**) shows a 7×7 window of digits from an ECAL simulated event. The three streams of data (**c**–**e**) are 5×5 sub-samples from image (**a**). Image (**c**) contains the digit data, image (**d**) contains the local maxima data and image (**e**) contain the central cluster data from image (**a**). Image (**b**) shows the output of the network using the three (**c**–**e**) streams. It represents the reconstructed cluster in the centre of image (**a**) with the reconstructed value of its digits.

data from $B \to K^*\gamma$ decays in Run 3 conditions and take a selected subset of approximately 30,000 windows of 7×7 cells centered on different cluster seeds. The selection takes into account the balancing of the data-set between six different cases specified in Table 5.2. Case 6 is included to enhance the training of the fraction operation when clusters have a big energy difference between them. In these specific cases, big clusters tend to mask completely smaller clusters on its surroundings. Since case 5 has the lowest number of samples, we choose to increase its number by rotating each window 90°, 180° and 270°, reaching more than 5000 different samples for that case. Finally, the balanced data-set is constructed, collecting 5000 samples from each of the six cases and sub-sampling nine windows of 5×5 for each of them, reaching a combined number of 270,000 samples for the training. The expected output from

the network is generated following the defined application of the rule-set for this step.

Case	Number of Clusters (K)	Overlap with Central Cell	Samples on 2k Events	RMSE (MeV)
1	0	No	153,519	96.281
2	1	No	121,316	135.616
3	2	Yes (1 cluster)	45,066	147.501
4	3	Yes (2 clusters)	9937	199.644
5	4+	Yes (3+ clusters)	1367	244.312
6	>1	Yes (energy difference of 1 order of magnitude)	6816	181.693

Table 5.2: Case characteristics in the balanced data-set for the MLP training. Case 6 is a selection of samples with overlap with at least one cluster, but where the energy difference between clusters is bigger than an order of magnitude. The RMSE values were extracted comparing the network predicted values and the samples generated with the application of the CA rule.

As can be seen in Table 5.2, there is an RMSE value for each datum case. This gives an overview of the precision of the network as a function of the data complexity. Since case 1 comprises samples in which there are no overlapping clusters around the central one, it is expected to have the lowest RMSE value. Once the samples start increasing in number of clusters involved, the RMSE value goes up as the complexity of the reconstruction increases, which is also expected. A good "symptom" is to see the case 6 samples, which show a good performance even with increased complexity regarding the energy difference between clusters. Even though the RMSE values are considerably low inside the full calorimeter range, it must be stated that the average energy value seen in the data-set used is of 1202.64 MeVs. With respect to that value, the maximum RMSE obtained from group 5 represents 20.3% of the average signal. However, the RMSE metric is sensitive to out-layers.

A summary of the network parameters for this reconstruction step is provided in Table 5.3. Except for the first dense layer, which has a linear activation function, the subsequent ones have a ReLU activation. Given the regressive nature of this network, results in terms of training performance are measured with the RMSE metric, comparing all the values predicted by the network to the results of applying the CA rules to the training input samples. Considering the reference of the mean energy value seen in the training samples, the RMSE of the network represents 14% of the average energy.

Aiming to provide a detailed overview of the structured proposal, Figure 5.4 shows the entire data-flow of the reconstruction process for the inner calorimeter region. The diagram gathers the relation between the neural networks and the data in the whole pipeline. The algorithm starts with the list of energy cells transformed into an image and ends with the list of reconstructed clusters. The list of image clusters will have a fixed length equal to the

Region	Image Shape	Training Samples	Neurons per Layer	Parameters	Training Time (s)	RMSE (MeV)
All	5×5	270,000	[64, 64, 64, 32]	108,993	2130.4	168.884

Table 5.3: Parameter summary of the cluster and overlap neural network.

number of cells of the specific calorimeter region, therefore the reconstructed clusters relate to the calorimeter coordinated by their position in the list. If one cell is identified as a seed of a cluster, the corresponding image of the stack will have the energy values of the digits in the reconstructed cluster. If one cell is not found to be a cluster seed, the corresponding image of the stack will contain only zeros.

For the other two regions, the structure is maintained, but the shapes of the constructed images are adapted to each region size.

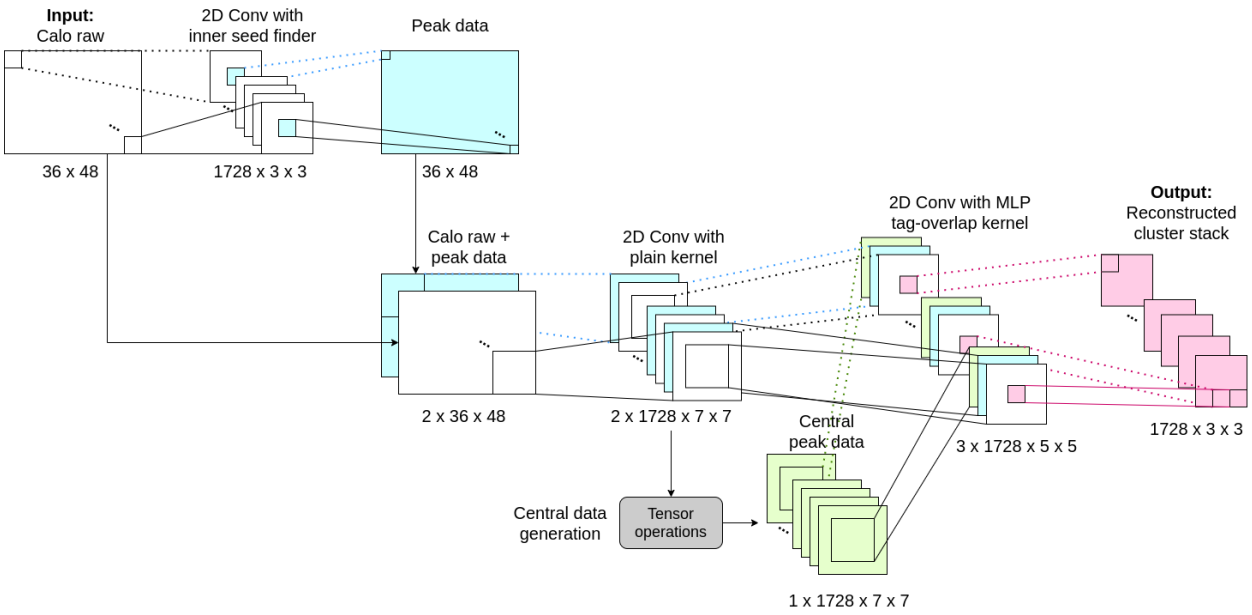


Figure 5.4: Detailed scheme of the proposed reconstruction data-flow for the inner calorimeter region.

5.3 Results

The results provided in this chapter are obtained operating on a computer with the following properties: memory of 15.5 GB, processor Intel Core i7-6500U CPU @ 2.50 GHz $\times 4$, disk

capacity of 512.1 GB with Ubuntu 20.04.1 LTS 64-bit OS. The algorithms are coded in Python 3.8 [71] and the explained neural networks have been built and trained using the TensorFlow 2.3.0 library [72].

Aiming to make a fair comparison in terms of computational performance between the proposed reconstruction algorithm and the original implementation of LHCb in a local environment, we have implemented an iterative Python algorithm following the rule-set functions defined for the Deep Learning method that has a similar computational complexity as the LHCb implemented Cellular Automaton.

To make sure the comparison between both algorithms is fair, we define a metric of efficiency on the reconstruction as relative error. This relative error computes the difference of energy between the reconstructed clusters and the energy value from the Monte Carlo particles, specifically for photons, without any corrections. Using the relative error metric in the reconstructed clusters from the proposed deep learning algorithm, we obtain the result from the first entry in Table 5.4. In the same table, we find the relative error for the reconstructed clusters obtained with the Python version of the Cellular Automaton. The values shown, concerning the two compared algorithms in relative error, are obtained as the average from over 200 simulation samples from ECAL not used in the training of the networks. It can be observed that the proposed deep learning approach shows, in general, a lower relative error value than the python cellular automaton version.

Algorithm	**Mean of Relative Error**	**STD of Relative Error**
Deep Learning	0.056	0.105
Python version of Cellular Automaton	0.079	0.159

Table 5.4: Results concerning the mean value and standard deviation of the relative error measured as the difference of energy reconstructed per cluster from a total of 200 simulated events.

However, when plotting the distribution of the energy resolution in Figure 5.5, the large positive tail indicates that, for both methods, the energy of the reconstructed clusters is underestimated compared to the Monte Carlo particle energies.

This could be explained given that the rule-set functions of the python CA method define a reconstruction specifically for 3×3 shaped clusters, whereas in the LHCb Cellular Automaton implementation, the clusters can be bigger if there are deposits around the cluster without overlap. Moreover, the studied approaches do not compute any corrections on the clusters energy. Although the resolution obtained cannot be compared to performance of the actual LHCb implemented algorithm, we have demonstrated that the deep learning proposal has efficiently learned how to implement the defined rule-set.

Results, in terms of computational performance, are measured as the time in seconds elapsed between the reading of the digits, and the generation of the list of clusters for the

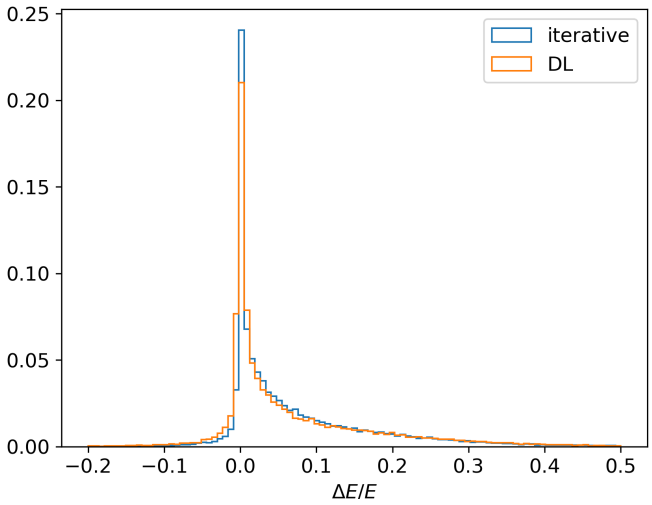


Figure 5.5: Normalized histogram of the energy resolution computed as the difference between the Monte Carlo particle energy and the reconstructed energy from the iterative Python version of the CA and the proposed DL method evaluated in 200 simulation events from B decays not used in the network training.

three regions of the calorimeter. The execution of both methods is done using a single thread in each case. Figure 5.6 shows a plot of the computational time as a function of the number of energy cells (hits) on a single sample of a calorimeter simulation (event). The time is measured as a mean of one hundred iterations on the same event for 200 different events. Looking at the curve from the Python version of the LHCb algorithm (iterative), it performs really fast in events with a low number of energy cells. However, it shows a clear quadratic growth with the number of digits. On the other hand, the deep learning approach (DL total) shows nearly constant behavior towards the number of digits per event. Although it has a small positive slope of 4.97×10^{-6}, the tendency shows to be linear. Around 72% of the events processed in the testing have less than 2575 digits and, therefore, stay under the time performance curve of the deep learning approach. Even so, we achieved a constant computational time with independence of the events complexity.

In addition, as stated in Section 5.2.2, for the first step of the proposed reconstruction process, the information from the three regions of the calorimeter needs to be treated separately. Given that the reconstruction process is the same for each region, except in the region dependent local maxima neural network, another way of accelerating the execution time is by running the reconstruction of the three regions in parallel. To approximate the behavior of such execution, Figure 5.6 shows the execution time measured by each of the three region reconstructions independently (DL inner, DL middle, and DL outer). It is observed that, in this parallel condition, the maximum time is achieved by the outer reconstruction, since it has the highest number of readout cells. Although more studies should be made in this direction, the proposed deep learning algorithm shows that it could benefit from a parallel

execution.

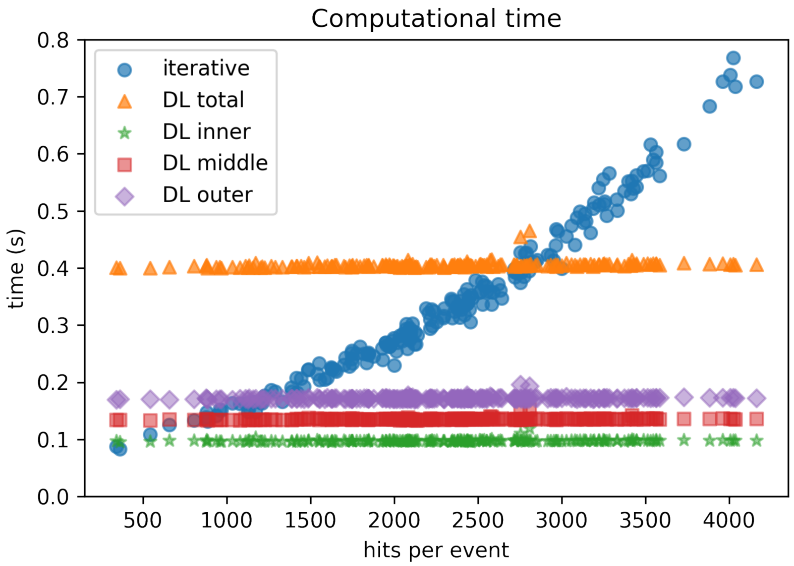


Figure 5.6: Scatter plot of the mean computational time over the number of readout cells per event from LHCb simulations. Comparing the Python version of the LHCb algorithm (iterative) and the proposed deep learning implementation (DL total) with executions segmented by regions (DL inner, DL middle, DL outer). Hits refer to the readout cells with energy on a sample.

5.4 Limitations and constraints

Once proven the efficiency and performance of the proposed DL method in a local environment, there is a need to use an inference engine to move the implementation into the LHCb framework for further testing. However, this has become a huge stopper.

In the latest years, there has been an increasing interest in machine learning and deep learning inference engines as a tool for fast deployment of AI models into production. One of the standards used in HEP is the TMVA tool for performing multivariate analysis in the ROOT framework [73]. It has long been used in LHCb for offline analysis of large data samples using mainly boosted decision trees (BDT) and MLPs. But precisely because this tool is thought to be used in offline analysis, it prioritizes easy usability rather than inference performance. As an example, a fast neural network based algorithm was proposed in 2017 for the identification of fake tracks in LHCb [74]. However the use of TMVA tools was not sufficient to cope with the HLT2 throughput requirements and it required by-hand modifications of the code to allow auto-vectorization and further optimizations of the implementation.

Since this is not scalable and hard to maintain, other tools have started to come into play. Focusing on the model deployment, ONNX is an open format built to represent machine learning models [75]. It defines the building blocks of ML and DL models as common operators and creates a common file format that allows to use the AI models in different frameworks. On the same line, TensorRT is a software development tool from NVIDIA that provides high-performance deep learning inference for CUDA environments [76]. It also allows to read and use ONNX files.

Taking advantage of the NVIDIA tools for CUDA, recent studies have used TensorRT to test the inference of two benchmark dense neural networks in the HLT1 GPU platform in LHCb [77]. The two networks tested are both MLP architectures using 17 input features from the LHCb tracking detectors. Looking at the overall results, the kernel overhead is the main bottleneck for throughput but large batch sizes minimize the throughput decrease. To give more detail, the first model tested is a dense neural network with two hidden layers. With the order of 1000 parameters, the HLT1 reconstruction sequence shows almost no throughput reduction using one instance of the network. However, compared to the second, larger approach, with six hidden layers of up to 128 neurons each, the throughput shows a decrease of almost 5%.

We can extrapolate those numbers to the proposed deep learning approach for the ECAL reconstruction. Overall, the overhead of the data processing and the MLP network inference can be approximated to have the same cost as the six hidden layer model tested in HLT1. To have a broad estimation of the whole impact of the ECAL reconstruction approach, we need to add the inference cost of the first CNN. Although it is negligible for one instance, the approach has one instance per region, which has an impact of almost 6% to the total throughput. Therefore, with a broad estimation, we can say that the DL approach for ECAL reconstruction would imply a throughput reduction of 11% of the whole HLT1 sequence when executed inside the GPU framework of LHCb. Without taking into account the overhead cost of the data preparation before the network inference.

5.5 Discussion and conclusions

Within the development of this proposal, there are several things that have been learned. We have seen that, for the specific problem of calorimeter reconstruction in LHCb, segmenting the reconstruction steps can help in simplifying the development of a deep learning solution. Moreover, as seen in Section 5.2.2, data can be artificially generated as long as they equally represent all the cases to be learned. For more complex functions, such as the one seen in Section 5.2.4, the understanding of the rules also leads to a simplification of the data-set, since we are able to extract thousands of samples from only 2000 full LHCb simulated events. Understanding that there is no need to work with full simulation data to train specific networks can simplify the training data-set generation on further deep learning developments for the calorimeter reconstruction. In other words, we have trained neural networks on the rules that solve a general formulation of the problem. It has been proved

that the network learns the application of the formulated rules in a generalized context. The complexity reduction on the training data has been also reflected into a fast training process and the simplification of the networks, in terms of architecture and the number of parameters, compared with previous deep learning approaches.

Comparing the results with the state-of-the-art, we improved the relation between the network's complexity and the amount of training data. Furthermore, the proposed model is validated by construction, since the same reconstruction steps as the ones used in the current method are being reproduced.

As a proof of concept, the performance comparison is done with a version of the current reconstruction self-implemented in Python. In terms of computational time, there is a clear gain in the reconstruction complexity with the proposed approach. However, the execution time could possibly be reduced with a vectorized implementation of the proposal. Apart from that, the proposed implementation clearly benefits from parallel execution, reducing the computational time by nearly a factor three. Moreover, its convolutional formulation could benefit even more from a GPU architecture without conditioning the efficiency, as the insight neural networks and convolutional operators are highly parallelizable.

In terms of energy resolution, although the results obtained are not comparable to the performance of the current LHCb implementation, we have demonstrated that the proposed networks achieved to learn the insights of the training data. However, due to the region-independent strategy used in this approach, clusters that fall in the boundary regions of the calorimeter are now reconstructed partially as two separate clusters in each region. There is the idea of using a graph neural network (GNN) with similar training as the MLP, in order to perform the reconstruction in the boundary regions, as GNNs can model irregular neighborhoods. Another aspect that needs to be worked on is the identification of π^0 particles. By nature, the two photons of the decay of energetic π^0 particles arrive at the calorimeter as two very close similar energy particles, but need to be reconstructed as a single cluster. Hence, the window that surrounds a pair of photons is bigger than the defined 3×3. With the current training of the MLP in our proposal, the network wrongly reconstructs these specific photons as two very close overlapping clusters. There is the idea of improving this shortcoming by fine tuning the network of the current implementation to identify π^0 candidates and make an ad-hoc reconstruction for those cases. As a general conclusion on the models performance, a further line of work that could improve the current proposal may include using Monte Carlo truth data to build the reconstructed cluster images instead of the application of the rule.set on the raw images. Using them to train the network could improve the energy resolution of the reconstructed clusters.

To summarize, we implemented and tested an alternative approach to the LHCb calorimeter reconstruction. It adapts the current reconstruction steps to a formulation that can be learned by simple deep learning structures. With this, we make sure that the reconstruction process is correct as it mimics the implementation of the current algorithm, gaining in computational complexity. Results from the testing show interesting behavior in terms of computational time, which could be promising for a full calorimeter reconstruction implementation on GPUs.

However, although the many application of AI models in HEP have demonstrated to be very effective in data analysis and reconstruction, the inference of such models to the real experiments frameworks is a clear bottleneck.

Through the study of the LHCb calorimeter reconstruction, we have seen that the standard HEP tools for the inference of models are not scalable and require an expert knowledge in advanced code optimization which sets a huge barrier for deploying or testing the models.

On the other hand, newer tools are starting to be mature enough to allow a generalized format for inferencing AI models that provide fast inferences in an optimized environment. However, the ECAL reconstruction process requires a set of complex operations that are non trivial for an AI application. Even with an optimized and simplified network architecture, the resulting algorithm is expected to have a non-negligible decreasing effect on the throughput even with parallel architectures.

Therefore it is important to highlight that deep learning models will only be suited for HEP trigger-like applications if they are small, simplified and well optimized. This can only be achieved when the insights of the problem are well understood and a network is modeled according to them, instead of expecting a model to learn the general rules of the problem by itself.

As a final message, it is key for the future development of deep learning applications in HEP to push the development of fast and optimized tools for inferencing AI models inside HEP frameworks either with CPUs or GPUs.

Chapter 6

Graph based reconstruction

The previous chapter demonstrates the feasibility of employing small convolutional neural networks, modeled as a cellular automaton, for ECAL reconstruction using deep learning. However, integrating the model into the LHCb framework has proven to be more challenging than anticipated. As a result, rather than further pursuing this approach, the decision was made to explore alternative reconstruction methods that do not rely on deep learning models.

Through the course of this thesis, the throughput of the HLT2 sequence has evolved with many improvements in different algorithms and processes. Figure 6.1 shows the evolution of the trigger rates in one year. Compared to the rated from 2020, shown in Figure 2.8 from Section 2.2.2, the calorimeter reconstruction has improved by almost a factor 2. This was achieved by the optimization of the matching algorithm between ECAL clusters and tracks, which is included under the "Calorimeter" tag in the throughput breakdown. However, this significant optimization is not related to the cluster reconstruction process and the ECAL reconstruction is still the fourth most time consuming process in HLT2.

Derived from the fact that the ECAL regions have a non uniform cell size, there is the idea to use graphs to model the neighbourhood of the cells and use a generalized cluster definition with independence of the cell or cluster shape. Developing this idea, a new algorithm for the calorimeter data reconstruction is proposed. The method, called Graph Clustering, makes use of graph data structures to optimise the clustering process. It outperforms the previously used method by 65.4% in terms of computational time on average, with an equivalent efficiency and resolution. The implementation of the Graph Clustering method is detailed in this chapter, together with its performance results tested inside the LHCb framework using simulation data. The Graph Clustering C++ code can be found in GitLab [79].

6.1 Background

Calorimeter data reconstruction can be understood as a clustering problem, as it aims to group the energy deposits from particles following a set of rules. Some of the classical approaches to clustering problems involve partitioning algorithms, such as k-means [80],

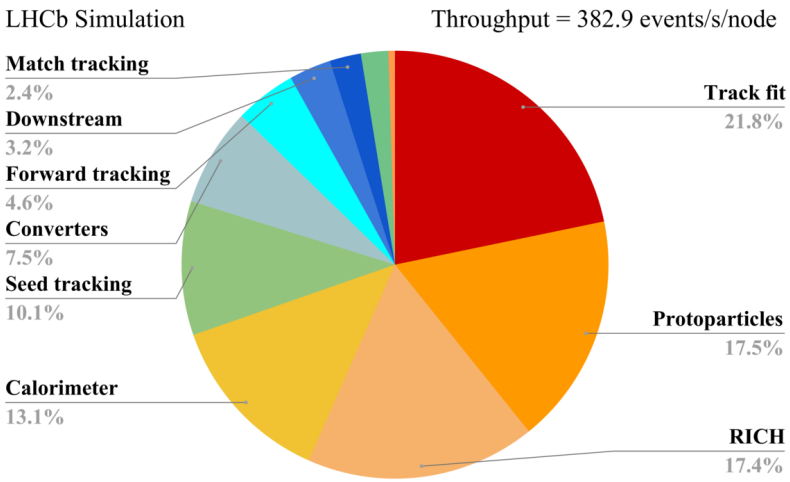


Figure 6.1: Breakdown of the HLT2 reconstruction throughput rate for the LHCb upgrade in 2021, using the "fastest" configuration. [78].

which use distance based metrics to organize data into clusters. The main drawback of these approaches is that for calorimeter clustering in high energy physics (HEP), the number of clusters k is not known in advance. Hierarchical methods build dendrograms structures of clusters by splitting or merging them, such as the HERA-B algorithm [45]. However, these methods do not scale well due to the high cost of merge and split functions [81]. Other approaches use density-based methods, such as DBSCAN [82] and OPTICS [83], build clusters according to high-density regions of data.

Focusing on the field of calorimetry in HEP, the Cellular Automaton has been used in LHCb for Runs 1 and 2 [38]. In 2020, the density-based clustering algorithm CLUE was presented [84]. Although its good performance, it is made for the CMS high granularity calorimeter with layers, which provides better separation conditions for the overlap cases in comparison to the LHCb ECAL's geometry. In 2004 an approach using spanning trees was proposed, using this flexible data structures to exploit the neighbourhood definitions in general calorimeter data [85], but it does not consider the cluster separation needed in LHCb. Graph data structures started to appear in the field with the increasing popularity of deep learning in the form of a neural model based on graphs [86]. Several approaches have used these graph neural networks on layered calorimeters [87, 88] showing promising results on clustering energy deposits in consecutive calorimeter layers. However, the LHCb calorimeter geometry is bi-dimensional. Within this context, other approaches have also used graph neural networks [89] and convolutional neural networks [66, 11] with similar conditions as ECAL in LHCb. That said, the inference of some deep learning models is still not mature enough to be incorporated in the LHCb software framework.

Since graph structures have demonstrated to be well suited for calorimeter data, the

Graph Clustering algorithm is based on storing the calorimeter digits into graphs and make use of its flexible neighbourhood properties to define the clusters. Moreover, it follows the same reconstruction principles from the Cellular Automaton strategy, which has proved to give a good performance in terms of reconstruction efficiency.

6.2 The method

The baseline idea behind the Graph Clustering algorithm is to use graphs as a data structure to store the event digits. It transforms the calorimeter digits into independent graph structures, where only relevant digits for a cluster are contained into isolated graphs. Following graph theory nomenclature, each energy digit from an event is represented as a vertex v in the graph, also called node. The relations between digits, representing links to the same cluster, are defined as directional edges (u, v) between the source digit node u and the target node v. By design, the target nodes of all edges in the graph are the seeds of the reconstructed clusters, where a seed is defined as a local maximum energy digit in the calorimeter grid over a threshold of 50 MeV in transverse energy. With this, the cluster seeds can be easily identified as nodes with only incoming edges. Furthermore, a node can be linked to more than one seed if it has energy deposits from more than one particle. These particular cases are then called overlap cells. Overall, the graph derived from an event may contain structures like the example shown in Figure 6.2.

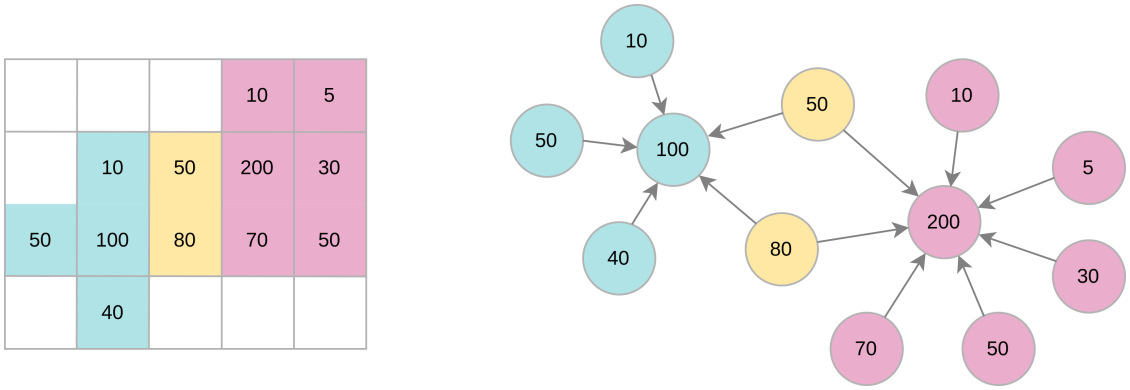


Figure 6.2: An example of two clusters with overlapping cells on the calorimeter on the left and its graph representation on the right. Empty cells in the grid have zero energy.

The following subsections describe in detail the four steps needed in the Graph Clustering reconstruction process.

6.2.1 Sorting

To achieve the mentioned representation of the digits, the algorithm needs to make an efficient insertion of the edges into the graph structure. Since all the edges are based on the cluster seeds, the initial key point is to identify seed candidates. As defined previously, a cell in the ECAL grid is considered a seed if it is a local maximum and has a minimum transverse energy value of 50 MeV. A local maximum in this context defines a cell that has the highest energy value among a 3×3 cell area around it in the calorimeter grid. This definition is the same as the one used in the Cellular Automaton algorithm.

In order to process the seed candidates of an event in the first place, all the digits above 50 MeV need to be sorted by decreasing transverse energy value. In the proposed algorithm, the sorting is computed using Introspective Sorting [90], which is a hybrid sorting algorithm that combines three different methods to provide fast average performance and optimal worst-case performance.

6.2.2 Insertion

The role of the insertion step is to build the graph edges between the event digits such that the graph structures of Figure 6.2 are obtained. A pseudo-code notation of this process is stated in Algorithm 1.

Algorithm 1 Graph insertion

```
1: G ⇐ directional weighted graph
2: for each energy, id ∈ sortedDigits do
3:     if id not inserted in G then
4:         if id is local maxima then
5:             add node id in G
6:             for each n_energy, n_id ∈ neighbours of id do
7:                 add node ne_id and edge (ne_id, id, w = 1) in G
8:                 if id is a merged π^0 candidate then
9:                     add id and n_id to mergedPi0
10:                end if
11:            end for
12:        end if
13:    else if id ∈ mergedPi0 then
14:        seed = first seed from id in G
15:        for each n_energy, n_id ∈ neighbours of id do
16:            if energy > n_energy & n_id not in G then
17:                add node ne_id and edge (ne_id, id, w = 1) in G
18:            end if
19:        end for
20:    end if
21: end for
```

It essentially iterates over each sorted digit. That digit may have already been inserted in the graph. If so, this means it is a neighbour of a more energetic digit. In that case, it cannot be a seed since there cannot be two adjacent maxima by construction, except for the case of merged π^0s, which is explained in section 3.2.2. Therefore, that digit is not inserted. On the other hand, if the digit has not yet been inserted on the graph, it can be either a seed or a residual digit, meaning it is not a local maximum and does not have any seed on its neighbourhood. To distinguish between the two, the algorithm checks if that digit is a local maximum by comparing its energy to its distance one neighbours. If that cell has the maximum value among the neighbors, it is considered a seed and it is inserted in the graph together with all its neighbour digits linking them with edges to the seed. The default weight value for all edges is one.

Additionally, if a merged π^0 candidate is identified following the metrics described in the following subsection, there is a second seed added to the same cluster. The neighbours of the second seed are also linked with an edge to the first seed, as it they were distance one neighbors if the first seed. This is done only if the new neighbors are not already inserted in the graph and their energy deposits is lower than the energy of the first seed.

At the end of the insertion step, the clusters are already grouped in the graph. However, the overlap cases still need to be processed to adjust the weight of the overlap edges.

Merged π^0 case

As explained in Section 3.2.2, the correct identification of merged π^0s is crucial in LHCb. Since the two photons of a merged π^0 may arrive at the ECAL as two adjacent seeds but only one of them is considered a local maxima, there might be residual energy outside the 3×3 window non negligible for the π^0 reconstruction. That is why the Graph Clustering algorithm adapts the shape of potential merged π^0 candidates, expanding the cluster up to the neighbours of the second most energetic digit in the cluster.

To avoid adding complexity to the data reconstruction algorithm, the energy deposits of the 3×3 cluster are the only source of information used to find a metric that can provide a soft selection filter for merged π^0 candidates at run time. Therefore, we have studied the relation between the two most energetic digits as a ratio labeled $R1$. Figure 6.3 shows a normalized histogram of the $R1$ ratio for over $46,000$ samples of single π^0 deposits from $B^0 \to \pi^+\pi^-\pi^0$ decays simulated using Run 3 conditions compared to photon samples from from $B^0 \to K^*\gamma$ decays also simulated using Run 3 conditions.

The main difference between the two distributions is that the majority of photons have an energy ratio between 0 and 25% whereas π^0 tend to have higher energy ratios in most cases. Therefore, the algorithm sets a threshold of value 25% in $R1$ to determine if a cluster needs to be expanded more than 3×3. This value has been optimized and ensures that the residual energy left outside the cluster is less than 9% for the studied π^0 samples and that the cluster expansion affects an average of 8.2% of the clusters in an event. Further studies have determined that small variations around 10% of the selected threshold value do not significantly change the time complexity of the algorithm nor the π^0 resolution.

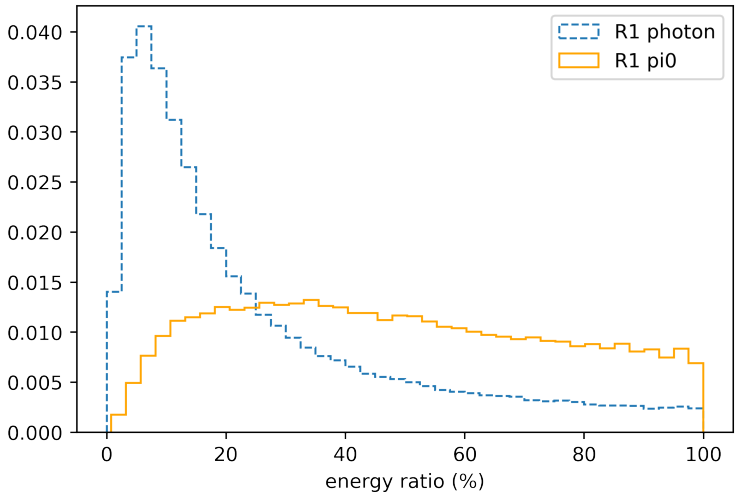


Figure 6.3: Normalized histograms of the energy ratio between the second most energetic digit and the cluster seed for photon samples and π^0 samples.

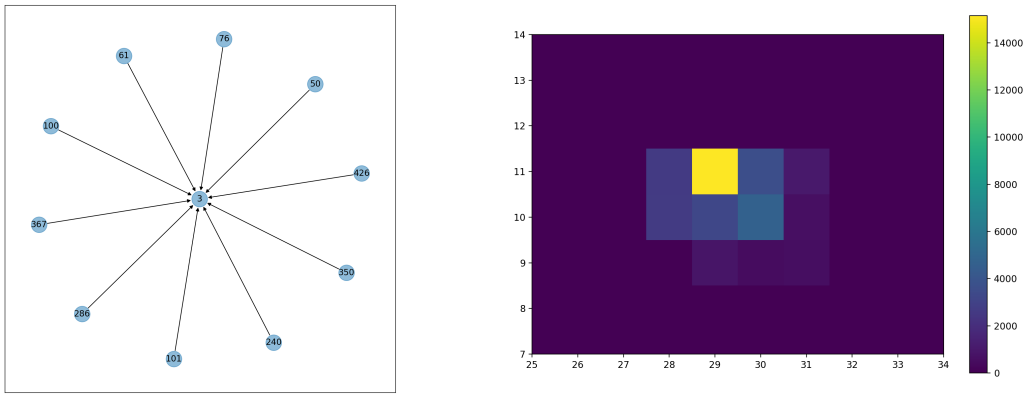
Moreover, given that only high energetic π^0s will be merged, a second threshold is added cutting the seed candidates under 1 GeV in the merged π^0 candidate selection. This value has been chosen according to the π^0 samples studied since it is the minimum seed value for a π^0 to be merged and not resolved.

The merged π^0 clusters that fall under the energy ratio threshold will simply be reconstructed as a 3×3 cluster, without adding expanded cells to the seed.

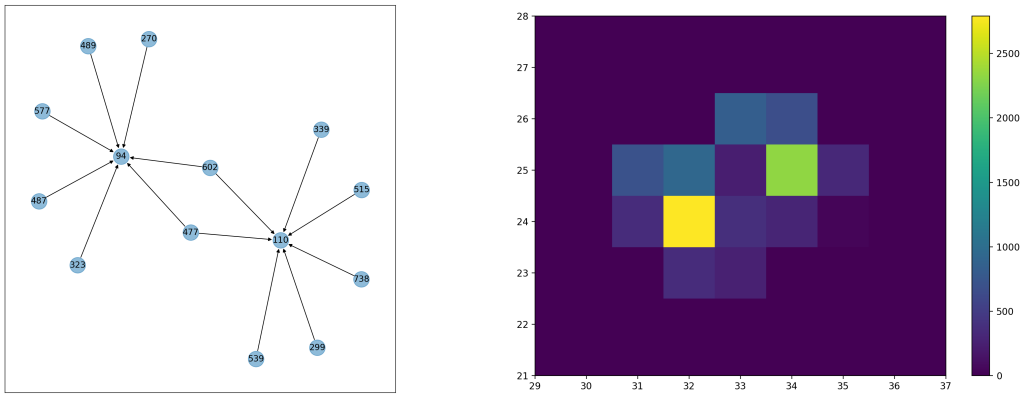
6.2.3 Connected Components

Once the insertion is finished, the graph structure contains all the relevant energy digits as nodes linked with the elements of each cluster and other overlapping clusters, if any. From this point on, the algorithm needs to process each cluster or group of overlapping clusters separately. Using graph theory terminology, a subset of nodes from a graph connected by some path is called a weakly connected component. Therefore, to retrieve the list of nodes that belong to the same cluster or group of overlapping clusters, the algorithm needs to find all the weekly connected components of the graph. In the proposed algorithm, this process is implemented as a depth-first search [91], which explores an entire graph exploring all its branches as far as possible before backtracking. Its time complexity is $O(\mid V \mid + \mid E \mid)$ [92] where V is the number of vertices or nodes in the graph and E is the number of edges. Once all the vertices of the graph are visited, the nodes and edges on each weakly connected component are obtained. Figure 6.4 shows three examples of connected components from a simulated event and its digit representation in the calorimeter grid.

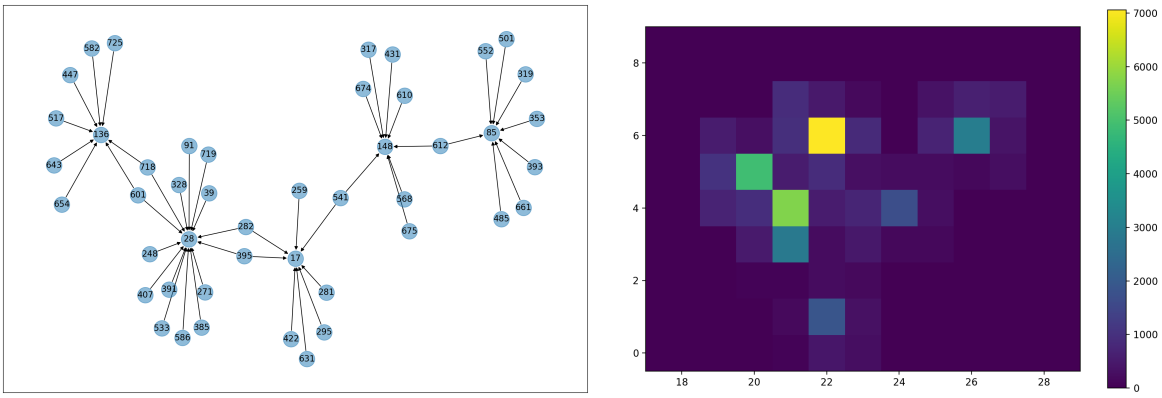
The first example, Figure 6.4a, shows an isolated cluster that has passed the thresholds to be considered a merged π^0 candidate. Therefore, the second most energetic digit of the



(a) Isolated cluster considered a merged π^0 candidate.



(b) Two overlapping clusters.



(c) Big connected component with 5 clusters.

Figure 6.4: Graph representations of the connected components in Graph Clustering and the corresponding digits in the ECAL grid. The graph nodes are numbered according to their position in the ordered digits list.

cluster is considered a second seed and its five neighbour digits are also linked to the original seed and its original five digits. The resulting graph consists of a single seed with 10 digits linked directly to it. In the second example, in Figure 6.4b, there are two overlapping clusters as the most energetic cells are both local maxima. There are only two overlapping cells, which are linked to both seeds in the connected component graph. The third example in Figure 6.4c represents one big connected component with five clusters overlapping. One of them also fulfills the merged π^0 candidate requirements in two of its neighbouring digits, therefore four and three extra digits are also linked to the seed, which ends up with 15 digits linked to it.

In order to study the complexity of the connected components, the number of nodes on each one is studied. From a total of 1000 simulated events from $B \to K^*\gamma$ decays in Run 3 conditions, 42% of the total connected components have nine nodes, which correspond to a 3×3 cluster shape. From the rest of them, 52% have more than nine nodes and an average of 22 nodes per connected component. Therefore, it can be said that almost half of the connected components of an event are isolated clusters. Moreover, this assumption is not taking into account that clusters can be bigger than 3×3 if the merged π^0 requirements are fulfilled. Consequently, the average complexity of the connected components is not particularly high, suggesting that the cost of individual analysis is expected to remain reasonable.

6.2.4 Analysis of Clusters

The final step of the reconstruction is to analyze each weakly connected component to resolve the overlap cases if any, and transform the graph clusters into the regular output cluster format. The processing of a weakly connected component can be done independently of the others, since each one contains only the relevant nodes and edges for a cluster. The analysis of clusters consists on iterating through the list of weekly connected components following the pseudo-code in Algorithm 2.

Algorithm 2 Analysis of connected components

```
for each wcc ∈ weaklyConnectedComponents do
    if wcc.size() > 1 then
        calculate overlap weights (Algorithm 3)
        for each id ∈ wcc do
            if id in-edges > 1 & id out-edges == 0 then
                add id as a cluster seed to clusters
                for each vertex connected to id do
                    add vertex as entry of id in clusters
                end for
            end if
        end for
    end if
end for
```

Only connected components with more than one node are considered as reconstructed clusters. Any isolated node is likely to be a residual energy deposit from a cluster and should not be considered a reconstructed cluster itself. If there is more than one seed in a connected component, there is at least one cell overlapping between two clusters. In that case the overlap resolution, defined in Algorithm 3, consists in assigning a fraction of the energy of the overlapping cell to each of the seeds linked to it. The fraction is calculated as a function of the energy of the clusters and is stored as the weight of that edge.

Algorithm 3 Calculate overlap weights

```
clusterEnergy ⇐ empty map
for each vertex ∈ wcc do
    if vertex out-edges >= 2 then
        for each end_vertex ∈ vertex out-edges do
            energy = accumulate energy from the nodes linked to end_vertex.
            energy+ = end_vertex energy / num out-edges.
            store energy to clusterEnergy
        end for
        totalEnergy = accumulate clusterEnergy energies with entries ∈ vertex out-edges
        for each end_vertex ∈ vertex out edges do
            weight = clusterEnergy at end_vertex / totalEnergy
            set edge (vertex, end_vertex, w = weight)
        end for
    end if
end for
```

Entering in more detail, this algorithm iterates through all the vertices in a connected component. It searches for overlap vertices, identified by having two or more output edges, and accumulates the energy of all the connected nodes on all the clusters involved in the overlap. The energy of the overlap node is equally fractioned among the number of involved clusters to avoid accounting it more than once. Then, the weight of every overlapping edge is computed as the fraction between the energy of the target cluster and the sum of all the clusters involved in the overlap, following Equation 6.1 in the case of two overlapping clusters.

$$\text{weight}_{\text{cluster1}} = \frac{E_{\text{cluster1}}}{E_{\text{cluster1}} + E_{\text{cluster2}}} \tag{6.1}$$

Given that the Cellular Automaton algorithm implemented several iterations of the overlap computation until the fractions converge, we have studied the evolution of the fractions over several iterations in the Graph Clustering. Comparing the fractions of 1000 clusters from $B \to K^*\gamma$ simulations in a single overlap iteration and up to five overlap iterations, updating the total energy of the clusters. The fractions studied change 2.1% from the first to the second iteration and 2.4% from the first to the fifth iteration. These differences translate

to a variation on the total energy of the clusters of less than $0.8 \pm 0.5\%$. Moreover, the analysis of the connected components represents a 27.3% of the computational cost of the algorithm. As a result, the presented algorithm only performs one iteration of the overlap fractions computation.

6.3 Results

All the algorithm tests have been done within the GAUDI framework [93, 94]. For comparison purposes, in this section the performance of the Graph Clustering algorithm and the Cellular Automaton algorithm are compared, as the latter has been a benchmark reconstruction solution for Runs 1 and 2. Both are tested with the same Monte Carlo data from $B^0 \to K^*\gamma$ simulations using Run 3 conditions.

The quality of the reconstruction in calorimeter algorithms in LHCb is evaluated using metrics of efficiency, energy resolution and position resolution. The efficiency is defined as the fraction between reconstructed particles over reconstructible particles in a set of events. Reconstructible particles are photons that have deposited at least 90% of their energy in the calorimeter cells. The majority of photons outside this range fall on the boundary of the calorimeters acceptance. On the other hand, reconstructed particles are reconstructible particles matching a cluster from which at least 90% of their energy belong to that particle. This ratio is later referred to as match fraction. Table 6.1 shows that Graph Clustering has a higher efficiency than the Cellular Automaton, with $1.0 \pm 0.2\%$ more reconstructed clusters.

Algorithm	**Reconstructible**	**Reconstructed**	**Efficiency** (%)
Graph Clustering	43234	35313	81.68 ± 0.19
Cellular Automaton	43234	34872	80.66 ± 0.19

Table 6.1: Efficiency results in number of reconstructed versus reconstructible clusters from 80,000 $B^0 \to K^*\gamma$ events.

As mentioned before, the efficiency metric assumes all photons that have deposited at least 90% of its energy as reconstructible particles. This definition doesn't take into account the pile-up effect, understood as the superposition of particles in the ECAL cells. This effect includes the overlap cases, which are separable in the shower overlap algorithms, and cases where particles are fully superposed. As an example, a photon deposit may be completely overlapped by another particle, adding energy to the same digits. In such a case, the photon would still be accounted as reconstructible but the reconstructed cluster's energy would be larger and therefore it would not be accounted as reconstructed. As a result, with the current definition of the efficiency metric, a reconstruction algorithm will never reach 100% efficiency with the current ECAL detector properties. The obtained values of efficiency are therefore assumed to be good.

On the other hand, the energy resolution and position resolution metrics aim to measure the difference in energy and position between the reconstructed clusters and the associated Monte Carlo particles. Resolutions are evaluated for γ and π^0 particles. For both cases, we evaluate the difference in position on the X and Y axis and the difference in energy as a percentage. For γ resolution, a total of $80,000$ simulation samples of $B^0 \to K^*\gamma$ decays have been used, and another $80,000$ samples of $B^0 \to \pi^+\pi^-\pi^0$ decays have been used for π^0 resolution. The study accounts for all the clusters with a match fraction higher than 0.9 since it is the standard match threshold for a cluster to be considered reconstructed in terms of efficiency. For all the resolution studies, clusters do not have any corrections applied in order to compare the raw performance of the methods.

Figure 6.5 shows the energy distribution for both methods, where ΔE stands for the difference in the reconstructed energy of a cluster and truth energy of the Monte Carlo photon. It can be seen that for both γ and π^0 samples the distributions from the two methods look very alike. For energy resolution, Graph Clustering is slightly more shifted to negative values, but overall it can be said that the resolution in energy is equivalent to the Cellular Automaton one.

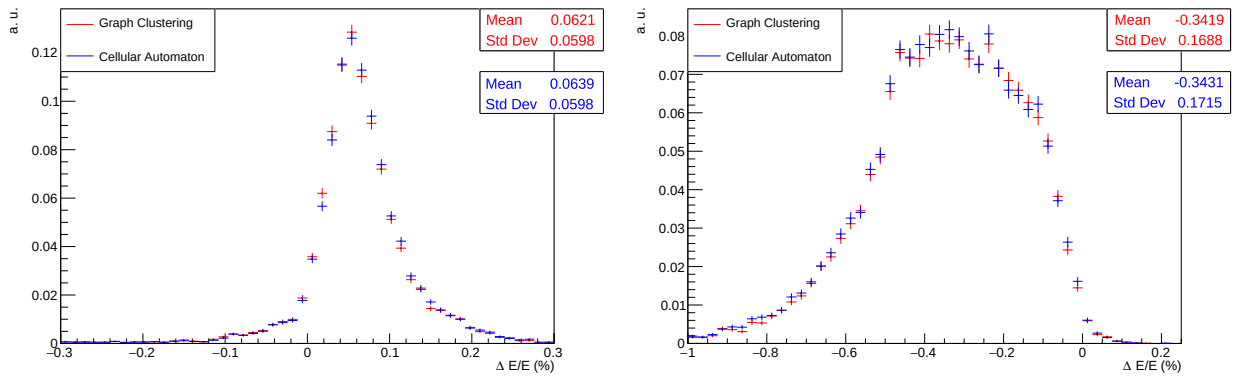


Figure 6.5: Normalized histograms of the energy resolution for clusters with a match fraction over 0.9 using γ samples in the left plot and merged π^0 samples in the right plot, both without corrections

Regarding the position resolution, Figure 6.6 shows that the X and Y distributions have again an equivalent behavior for both methods when using γ samples. In Figure 6.7, the same position resolution is evaluated for π^0 samples. Results show again that both methods are equivalent in terms of resolution before any corrections are applied. There is a small difference on the mean values for the position resolutions between X and Y axis of less than 3% of the standard deviation of the distributions. Given that this effect is seen in the same amount for the Graph Clustering and for the Cellular Automaton algorithms, it can be said that, although it requires further investigations, it does not have a direct impact on the comparison of the algorithms.

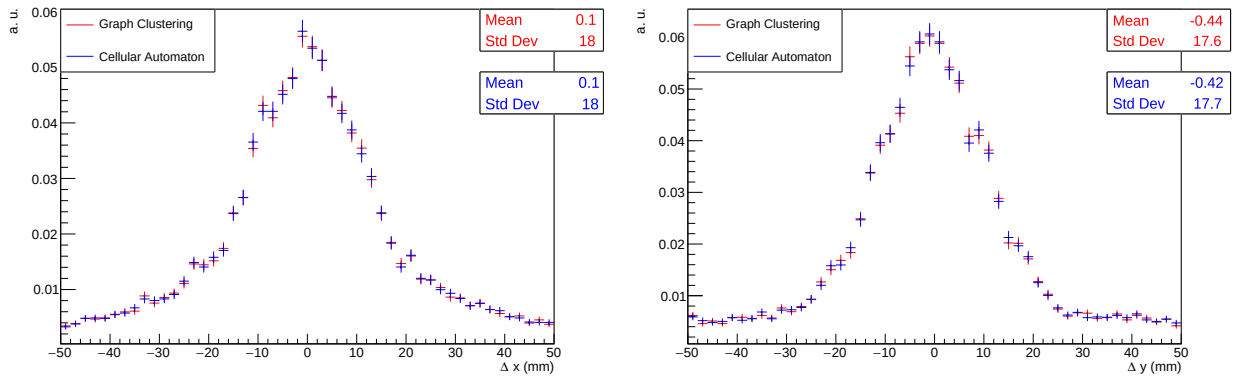


Figure 6.6: Normalized histograms of the X axis resolution at the left and the Y axis resolution at the right. Both using γ samples and clusters with a match fraction over 0.9 without corrections.

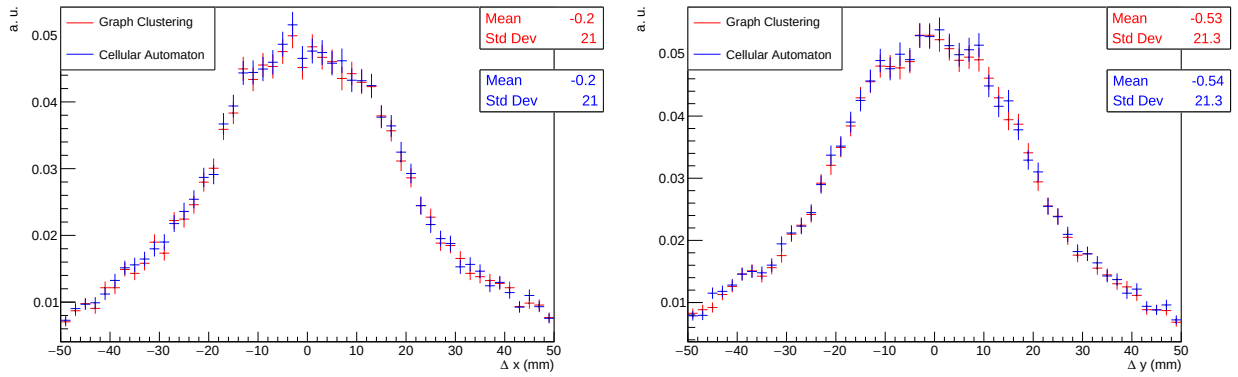


Figure 6.7: Normalized histograms of the X axis resolution at the left and the Y axis resolution at the right. Both using merged π^0 hypothesis and clusters with a match fraction over 0.9 without corrections.

Regarding the execution time, it is defined as the time elapsed between the first and the last lines executed in an algorithm. Figure 6.8 shows a plot of the execution time in arbitrary units as a function of the number of digits per event with energy greater than zero. The plotted time measurements are obtained as the average measured time from all the events with the same number of digits, from a total of $100,000$ events from $B^0 \to K^*\gamma$ simulation. The figure also includes error bars from the standard deviation of the samples with the same number of digits. This error reflects the small variation of complexity that the algorithm may have according to the distribution of the digits in the event as well as, more significantly, the variations in the available resources from the distributed computing

environment where the tests have been executed. Taking as a reference the fitted curves from the plot, for events with less than 150 digits, the Cellular Automaton is faster. However, from that point on, Graph Clustering outstands the benchmark algorithm showing a flatter complexity curve. Furthermore, the average number of digits per event from the analysed samples is 1520 digits. At that complexity level, Graph Clustering is 65.4% faster than Cellular Automaton on average.

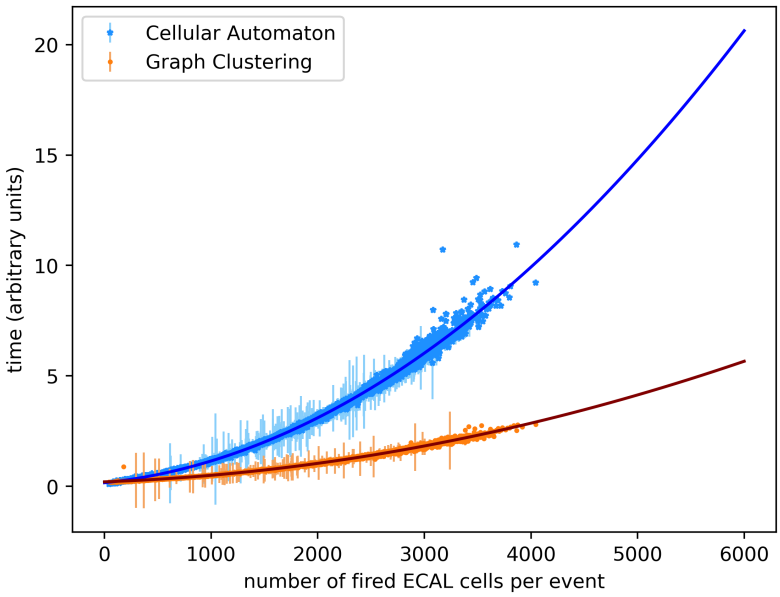


Figure 6.8: Execution time measured in arbitrary units as a function of the number of digits per event for the Cellular Automaton algorithm and the Graph Clustering algorithm. On top of them, a fitted curve for every algorithm is shown.

Further studies of efficiency and resolution performance of the Graph Clustering method can be found in Appendix A. Regarding the effect on the throughput rate in the HLT2 sequence, the Graph Clustering implies a 4.09% of throughput increase in the fastest reconstruction sequence. Consequently, the calorimeter reconstruction represents now a 9.01% of the HLT2 reconstruction in comparison to the 2021 13.1% seen in Figure 6.1.

6.4 Discussion and Conclusions

Graph Clustering has shown to improve the computational complexity of the calorimeter data reconstruction in LHCb. Furthermore, it is the default reconstruction solution for the ongoing Run 3 data taking period. The baseline of the algorithm is to reproduce the same

reconstruction steps as in the previously used algorithm, the Cellular Automaton, but with an optimized codification using graph data structures. Hence, it is expected and observed to have similar results compared to the benchmark in terms of efficiency and resolution. The observed efficiency is consistent with the efficiencies in Run 1 and Run 2 [78]. It is considered good in terms of performance since the definition of a reconstructible particle does not take into account noise or other fully overlapping particles, known as the pileup effect. Hence, the data reconstruction efficiency is not expected to reach 100% but gives an overall idea of the algorithms performance.

Graphs have demonstrated to be suited for calorimeter data reconstruction. Within the proposed implementation, such data structures also provide a flexible interpretation of the neighbour cells in the calorimeter grid. This could also be used to adapt the shape of the clusters to an optimized pattern depending on the region at reconstruction time and significantly accelerate its execution. Currently, the definition of an optimal cluster shape for ECAL clusters is being studied considering pileup and overlap effects as well as precision.

Within the steps of the presented Graph Clustering, as mentioned in section 6.2.4, the analysis of each connected component is completely independent of the rest of the graph. Although it is not the most time consuming part of the algorithm, it represents a 27.3% of the total algorithm's execution time, which could benefit from parallel execution.

As a final conclusion, the complexity curve that Graph Clustering exhibits makes it a useful alternative for other calorimeters with higher occupancy. Furthermore, the vision of future upgrades in the LHCb calorimeter is a challenging opportunity to test the limits of this algorithm.

Chapter 7

HLT1 reconstruction

The first level of the trigger system in LHCb processes events at 30 MHz level. In order to make an accurate selection of the events to further process, there is a preliminary reconstruction of the events that include tracking, primary and secondary vertex finding, and also an ECAL clustering. Given the tight time constraints of this trigger, the algorithms need to be as optimized as possible to make use of the GPU parallel execution. However, the more similar the results are to the HLT2 reconstruction, the more consistent the selections will be between the two trigger stages.

The current calorimeter reconstruction algorithm in the HLT1 is a very simplified clustering algorithm in CUDA. Given the good performance of the Graph Clustering algorithm presented in the previous chapter in HLT2, there is the idea of translating the same logic processes of the Graph Clustering into a CUDA algorithm optimized for parallel computing in HLT1. Three different approaches are proposed in order to implement the cluster overlap separation, leading to three different versions of the algorithm. The insights from these proposals are addressed in this chapter, along with certain results concerning efficiency, performance, and throughput impact. It can be observed that the overall efficiency of the reconstruction process improves at the cost of losing throughput, which is expected due to the increased complexity of reconstruction. However, results concerning energy and position resolution do not exhibit a clear gain.

7.1 Background

The first part of the LHCb trigger system is build in a CUDA framework called Allen that make use of the GPU parallel execution to accelerate the code and cope with the throughput requirements [33]. The CUDA programming model runs one algorithm, called *kernel*, at a time. Every kernel is launched with many threads on the GPU executing the same instruction on different parts of the data in parallel, independently from each other. These threads are grouped into blocks within a grid. Threads within one block share a common memory and can be synchronized, while threads from different blocks cannot communicate. The

threads are mapped onto the thousands of cores available on modern GPUs for processing. In the Allen framework, a single event is assigned to one block, while intra-event parallelism is mapped to the threads within one block. This ensures that communication is possible among threads processing the same event.

The baseline reconstruction of the HLT1 is focused on the reconstruction of tracks, associating them to primary and secondary vertices and performing a muon particle identification. As observed in Figure 7.1, a Global Event Cut (GEC) is applied before the reconstruction sequence. This cut removes a fraction of events with higher occupancy, eliminating complex events characterized by low detector performance and high reconstruction computing time.

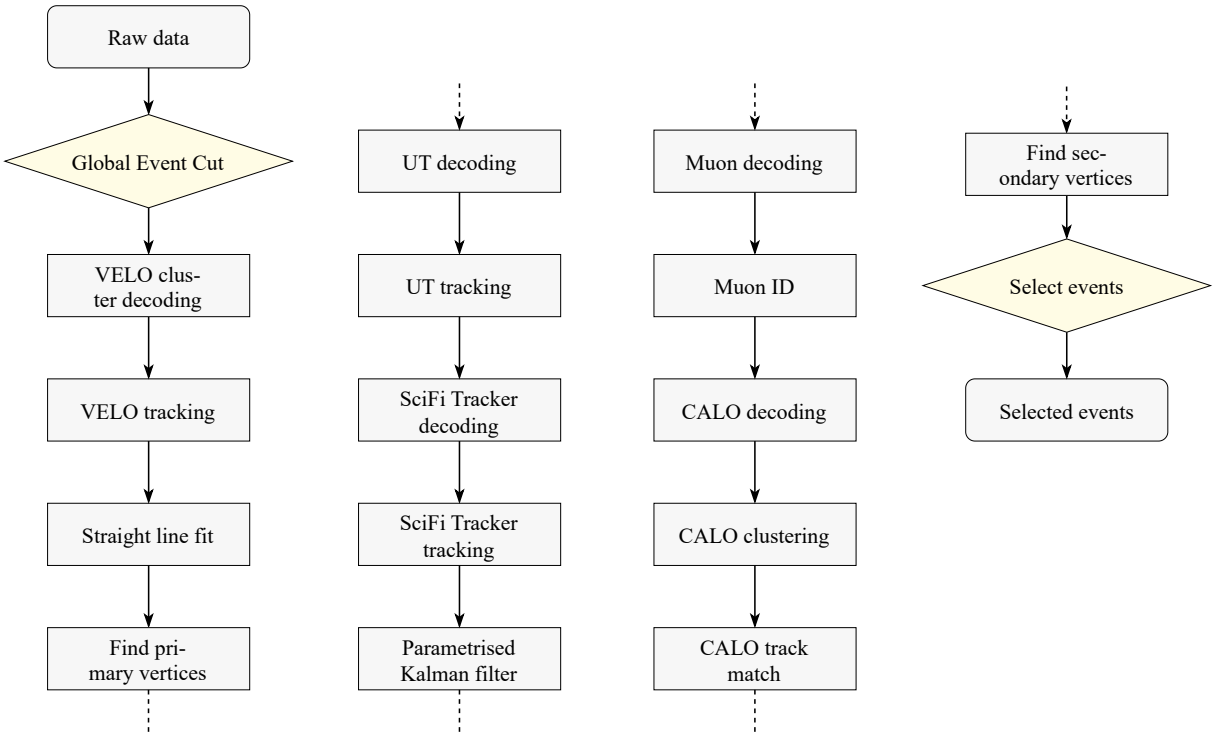


Figure 7.1: Baseline HLT1 sequence, updated from [36]. Rhombi represent algorithms reducing the event rate, while rectangles represent algorithms processing data.

The existing algorithms in the sequence are specifically programmed to operate efficiently across parallel architectures [95, 96, 33]. While the initial stage of HLT1 did not incorporate calorimeter clustering, other experiments in the collaboration have demonstrated the feasibility of efficient calorimeter clustering algorithms for parallel architectures [48, 97, 98]. Nevertheless, the current implementation of calorimeter clustering in the HLT1 reconstruction sequence is at a very preliminary stage and has lower efficiency when compared to HLT2 reconstruction.

7.2 ECAL reconstruction in HLT1

The calorimeter clustering implemented in HLT1 consists in searching for the seeds of clusters and building clusters with the 3×3 neighbors that are sitting around the seed. Comparing this approach to the current HLT2 reconstruction, the most relevant difference is that the overlap cases are not taken into account. This means that if there is an overlap cell in between of two clusters, the energy of that cell is accounted twice, once for every cluster. Therefore, the total energy of the resulting clusters may be higher than the true energy of the particles. Apart from that, the current approach only considers 3×3 shaped clusters.

To identify the cluster seeds, the current algorithm retrieves the distance one neighbors of each cell and checks if it has the maximum value among them. If it is the case, and the cell value is greater than a threshold of 10 ADC counts[1], that cell is accounted as a cluster seed. The accounting must be atomic in order to give a unique identifier to each cluster seed. This operation is parallelized for each energy digit in an event. Once it has finished, the number of unique seed identifiers in an event is the indicator of the number of clusters. Therefore, to build the reconstructed clusters, a parallel loop iterates for each cluster seed and retrieves again its distance one neighbors. If a neighbor's energy is higher than a certain threshold, that cell is accounted as part of the cluster. This implies that the unique identifier of the neighbor is added to the list of entries of the cluster, and the neighbor's energy is added to the total cluster's energy.

Depending on the energy value set as the threshold for the neighbors of a seed, the performance of the reconstruction will be considerably affected. To make an estimation of this effect, Figure 7.2 shows the evolution of the algorithms efficiency as a function of the neighbors threshold value. It can be seen how the lowering the threshold value considerably increase the efficiency. However, the maximum efficiency is achieved when the threshold has 10 ADC value, not zero. This suggests that filtering some of the low energy digits actually increases the number of reconstructed clusters rather than accounting for all of the neighbors. In other words, this could mean that accounting for all of the digits implies reconstructing a higher energy than the truth. This gives a hint that resolving the overlap cases may improve the inefficiency seen when the threshold is lowered to zero.

Figure 7.2 also plots the benchmark HLT2 ECAL reconstruction efficiency provided by the Graph Clustering algorithm, which does not have any threshold value for the neighbor digits of a seed. Compared to the maximum efficiency currently achieved with the HLT1 algorithm, there are still opportunities for improvement. Therefore, the approaches presented in the following sections implement the shower overlap mechanic as an additional process in the current HLT1 ECAL clustering algorithm.

[1] One ADC count equals 2.5 MeV according to the gain of the ECAL PMTs.

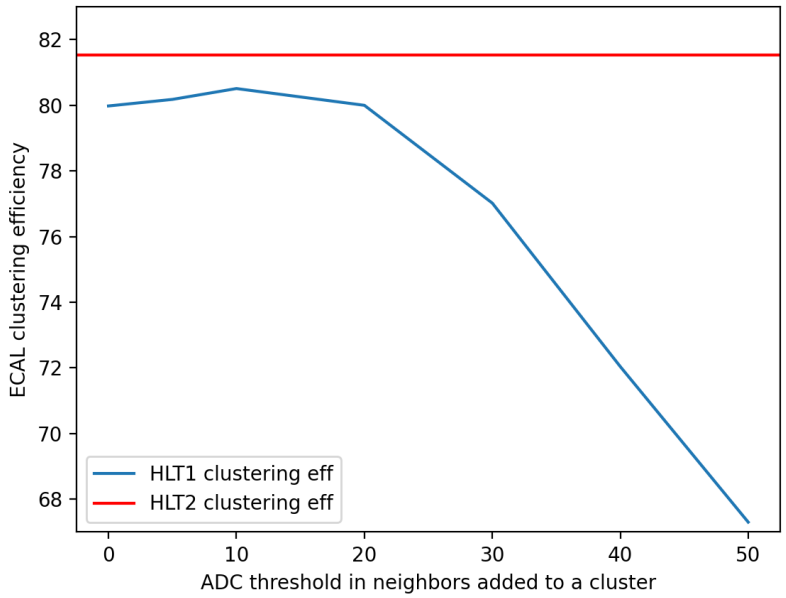


Figure 7.2: Efficiency of the ECAL reconstruction as a function of the threshold value for the neighbor digits of a cluster seed. The efficiency is evaluated using $10,000$ $B \to K^*\gamma$ Monte Carlo events in Upgrade conditions.

7.3 The method

In order to improve the reconstruction efficiency of the CUDA algorithm, the shower overlap process is implemented following the same logic used in the Graph Clustering and previous HLT2 algorithms: cells that are overlapping between two or more clusters need to be identified. Then, its energy is split between all the overlapping clusters in fractions according to the energy of each involved cluster. In order to optimize the process as much as possible without the need to filter digits according to its energy, three different implementations of the shower overlap are proposed. Each one is then evaluated in terms of efficiency and throughput.

7.3.1 First approach

In the first approach, the seed finder kernel is maintained. As a second step, in order to identify the overlap cases, another kernel iterates again through all the digits of an event. Then, the eight neighbors of the digit cell are retrieved and the number of seeds among them is accounted. If there is more than one seed, that digit cell is identified as an overlap case, as the example shown in Figure 7.3. An additional vector containing atomic instances

of the overlap cells is then filled. Each overlap instance contains the digit identifier, its energy value and the unique seed identifiers of the two overlapping clusters. This particular implementation only takes into account two overlapping clusters at a time. Algorithm 4 summarizes the behavior of the overlap kernel for this approach in pseudo code.

Algorithm 4 Overlap kernel first approach

```
1: overlapClusters = []
2: for each digit ∈ eventDigits do
3:     seedCounter = 0
4:     overlapSeedIDs = []
5:     for each neighbor ∈ digit neighbors do
6:         if neighbor is a seed then
7:             increment seedCounter
8:             add neighbor to overlapSeedIDs
9:         end if
10:    end for
11:    if seedCounter > 1 then
12:        atomic add of (digit, overlapSeedIDs) to overlapClusters
13:    end if
14: end for
```

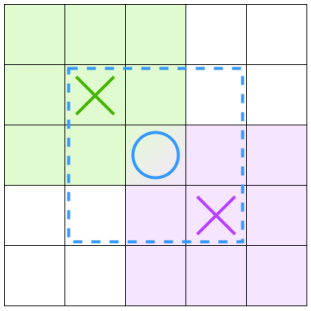


Figure 7.3: Diagram representing two overlapping clusters marked with an X and an overlapping cell marked with a circle.

Before the next step, a prefix sum kernel is needed to account for the number of overlap instances per event created in the new variable. As a third step of the algorithm, the original kernel that builds the clusters with the neighbors of the seeds is maintained, but with the neighbor threshold value set to zero. Finally, a fourth kernel is added in order to make a correction of the clusters energies according to the overlap cases. In this function, the kernel iterates through all the overlap cells and computes the energy correction for each involved cluster as defined in Equation 7.1. Since the energy of the clusters E_{cluster1} and E_{cluster2} is accounted including the energy of the overlap cell on both cases, half of E_{overlap} is subtracted from each cluster to avoid accumulating twice the energy of the overlapping cell.

$$\text{correction}_{\text{cluster1}} = E_{\text{overlap}} \frac{E_{\text{cluster2}} - \frac{E_{\text{overlap}}}{2}}{E_{\text{cluster1}} + E_{\text{cluster2}} - E_{\text{overlap}}} \tag{7.1}$$

Finally, the correction is subtracted from the total accumulated energy of the corresponding cluster. At the end of this step, the reconstructed clusters for an event are obtained with the total energy accumulated taking into account the overlap corrections.

Two limiting factors found in this implementation. The first one concerns that accounting for the overlap cells is an atomic function that stores data to a new variable and is only needed in the last step of the reconstruction. The second one is that in order to modify the energy of the clusters with the overlap corrections, an entire copy of the cluster vector needs to be done, which is an overhead cost added to the algorithm.

7.3.2 Second approach

The second approach tries to overcome the previous limitations by making a pre-calculation of the overlap energy correction before building the clusters. Initially, the first local maxima kernel is maintained. As a second step, instead of iterating through the digits of an event, we take advantage of the fact that the number of clusters of the event is already known. Moreover, it is certain that any overlap cell will be in the neighborhood of a seed. Therefore, the kernel iterates through all the seeds of the event and retrieves its neighbors, which are potential overlap candidates. For each neighbor, its own distance one neighbors are retrieved, and we account for the number of other seeds among them and accumulate its energy value in a separate variable. If the counter is higher than one, an overlap cell is identified. However, instead of an atomic accounting of the overlap cell, this approach directly computes the overlap correction that needs to be applied to the cluster from the initial seed. In this function, the total energy of the clusters around the seeds has not been computed yet. Therefore, the overlap correction is estimated only with the energy of the overlapping seeds, following Equation 7.2.

$$\text{correction}_{\text{cluster1}} = E_{\text{overlap}} \frac{E_{\text{seed2}}}{E_{\text{seed1}} + E_{\text{seed2}}} \tag{7.2}$$

Using the diagram of Figure 7.3 as a reference, if the kernel starts processing the green X seed, its neighbors are marked in light green. Eventually, the kernel will retrieve the neighbors of the blue circled cell and find there is the second seed marked in purple on its neighbourhood. Therefore, the correction for the green cluster will be computed using the overlap energy of the blue circled cell and the energies of the green and purple seeds. Additionally, since the energy of all the overlapping seeds in the neighbourhood of a cell is accumulated, this approach is not limited to two overlapping clusters. Algorithm 5 summarizes in pseudo code the mentioned behavior.

Algorithm 5 Overlap kernel second approach

```
1: clusterCorrections = []
2: for each seed ∈ eventSeeds do
3:     for each n1 ∈ seed neighbors do
4:         seedCounter = 0
5:         seedsEnergy = 0
6:         for each n2 ∈ n1neighbors do
7:             if n2 is a seed and n2 =! seed then
8:                 increment seedCounter
9:                 add n2.energy to seedsEnergy
10:            end if
11:        end for
12:    end for
13:    if seedCounter > 0 then
14:        correction = n1.energy * seedsEnergy / (seed.energy+seedsEnergy)
15:        clusterCorrections[seed] += correction
16:    end if
17: end for
```

Although the computed correction is not as accurate as taking into account the total energy of the clusters, this allows to simplify the overlap identification process and the number of intermediate variables needed.

As a third step, the original cluster building kernel is modified to directly subtract the energy correction at the time of accumulating the energy of all of the neighbors of a seed.

7.3.3 Third approach

The third approach consists of adding a small modification to the second approach such that the energy corrections is computed using the total energy of the clusters instead of only using the energy of the seeds. To do so, the first kernel is updated to accumulate the energy of the neighbour cells or each event digit at the time of checking the local maxima. If the digit is a local maxima, the accumulated energy is also stored in the seed object, together with its unique identifier.

Then, in the second kernel, once an overlap cell is identified, the energy correction associated to a cluster can be computed as in the first approach, given that the total energy of the clusters is known. Therefore, the overlap kernel for this approach follows Algorithm 5 except for the correction computation in line 14 that follows Equation 7.1. Finally, the third kernel is maintained as in the second approach.

7.4 Results

The three presented approaches have been implemented inside the LHCb Allen framework and evaluated in terms of efficiency, energy and position resolution and throughput impact. The efficiency measurement is evaluated in the same way as for the Graph Clustering method: the fraction of reconstructed clusters over the number of reconstructible clusters in a set of events. For this purpose, a total of $50,000$ events from $B \to K^*\gamma$ Monte Carlo simulations in Run 3 conditions has been used. The measurement of the throughput impact has been done in a GPU shared environment using an NVIDIA GeForce RTX 2080 Ti. The throughput number is obtained as the maximum value among 20 independent executions of the HLT1 default sequence evaluating 1000 events, including the respective calorimeter reconstruction algorithms. Table 7.1 shows the efficiency and throughput values of the three proposed approaches as well as the metrics for the current HLT1 ECAL clustering. Additionally, the efficiency of the Graph Clustering algorithm in HLT2 is evaluated using the same simulation samples for comparison purposes.

Algorithm	**Efficiency (%)**	**Throughput (events/second)**
HLT1 original	80.51 ± 0.29	$99,911.03$
HLT2 Graph Clustering	81.54 ± 0.28	-
First approach	81.17 ± 0.29	$96,515.43$
Second approach	81.04 ± 0.29	$97,383.57$
Third approach	81.32 ± 0.29	$97,141.25$

Table 7.1: Performance in terms of efficiency and throughput of the HLT1 clustering algorithm, the three proposed approaches and the Graph Clustering algorithm in HLT2.

The results in terms of efficiency show that there is an improvement in the number of reconstructed clusters in all of the presented approaches that implement the shower overlap resolution. From the three of them, the second approach has the least improvement in efficiency. This is expected since the overlap corrections are computed using only the energy of the seeds. Although the correction is computed with the same formula in the first and third approaches, the last one has a higher efficiency. This can be explained given that the first approach only takes into account a maximum of two overlapping clusters on each overlapping cell, whereas in the second and third approaches, any number of overlapping clusters in the neighbourhood of an overlapping cell will be accounted. Therefore it is expected that the third approach gives the highest accuracy among the three. The results achieved show an improvement in the efficiency which is compatible with the Graph Clustering within errors for the three proposed approaches.

In terms of throughput, all of the proposed approaches entail a reduction of the throughput. Which is expected since the implementation of the shower overlap is adding complexity to the reconstruction process and adding more kernel functions in the trigger chain. Giving a

little more of detail, the first approach is the one with a highest impact lowering the original throughput in 3.4%. The second approach has the smallest impact representing only a 2.5% of throughput reduction. The third proposal stays in between of the two with a throughput reduction of 2.8%. Overall, the decrease in throughput is not large and could be accepted in the current HLT1 sequence.

Regarding the resolution of the reconstructed clusters, 80,000 simulation samples from $B \to K^*\gamma$ in upgrade conditions have been used to evaluate the difference in position on the X and Y axis and the difference in energy as a percentage. The study accounts for all the clusters with a match fraction higher than 0.9 since it is the standard match threshold for a cluster to be considered reconstructed in terms of efficiency. Figure 7.4 shows the energy resolution for the three presented approaches as well as the current ECAL HLT1 reconstruction and the Graph Clustering algorithm from HLT2 without any cluster corrections. The mean of the Graph Clustering distribution has been taken as reference zero value.

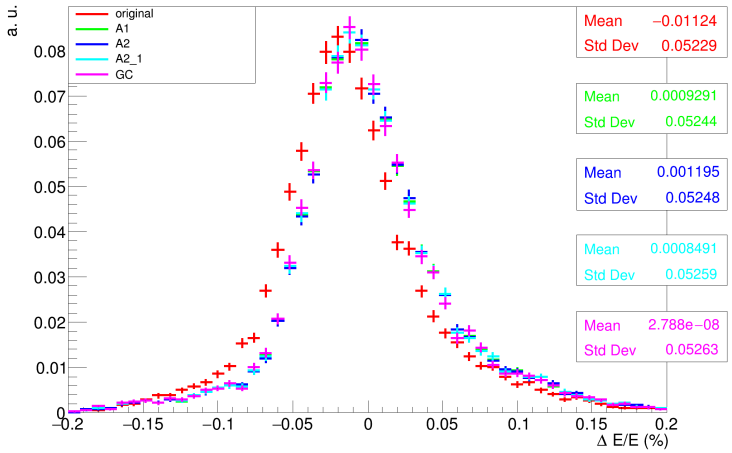


Figure 7.4: Normalized histograms of the energy resolution with no corrections for clusters with a match fraction over 0.9 using γ samples.

Upon analyzing the mean energy resolutions, a negative bias is observed in the original HLT1 clustering when compared to the Graph Clustering approach. This discrepancy can be attributed to the original implementation's lack of overlap separation and the exclusion of digits with less than 10 ADCs from the clusters. As a result, the total energy of the clusters is reduced, impacting the energy resolution more than the overlap itself. In contrast, the presented approaches, which incorporate overlap separation and do not impose any energy cut on the digits, show energy resolutions that are much closer to the performance of the Graph Clustering method. The differences between them are minimal, but the third approach ($A2\_1$) has the closest resolution to the Graph Clustering performance.

While the standard deviation of the distributions appears nearly identical, it alone cannot

serve as an indicator of improvement. The efficiency, on the other hand, plays a crucial role, particularly as the evaluated samples already have a match fraction of at least 0.9. In this context, a higher efficiency indicates that a greater number of samples are above the 0.9 match fraction threshold. However, this does not necessarily imply that the new samples have better resolution; rather, they are now successfully reconstructed, which was not the case before.

Regarding position resolution, the plots from Figure 7.5 do not show any significant difference between the studied methods. However, they are all close to the Graph Clustering performance, taken as a reference.

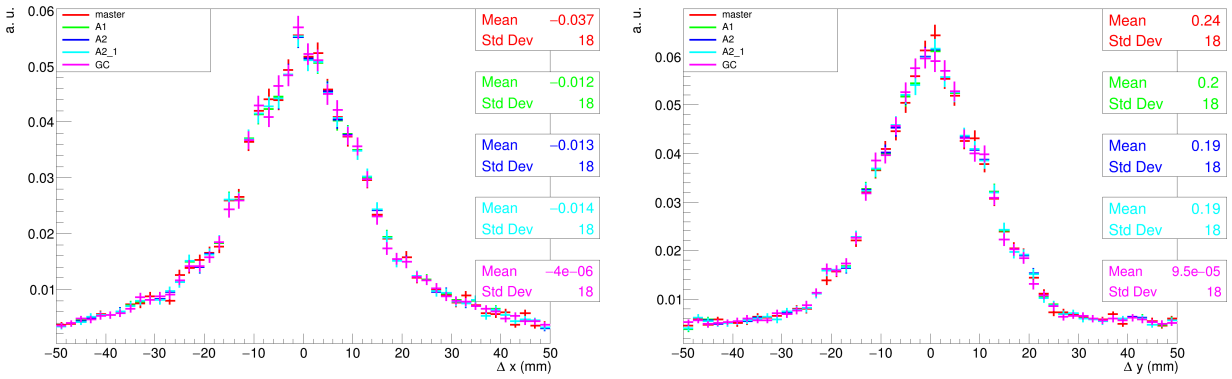


Figure 7.5: Normalized histograms of the X axis resolution at the left and the Y axis resolution at the right. Both using γ samples and clusters with a match fraction over 0.9 with no corrections.

7.5 Conclusions

It this chapter, three alternative clustering algorithms that implement the calorimeter shower overlap have been designed, developed and tested inside the LHCb Allen framework. The results demonstrate that the addition of the shower overlap in the reconstruction improves the efficiency of the clusters with a minor throughput decrease. Among the three approaches presented, the third one has the best trade-off between efficiency and throughput, making it a compelling improvement for the current HLT1 reconstruction.

However, the presented approaches still have a notable difference with the HLT2 reconstruction, which is the capacity to expand the shape of the clusters to more than 3×3 cells. This characteristic benefits the reconstruction of the merged π^0 particles that arrive at the calorimeter as two closely-spaced photons. Therefore, further studies should explicitly evaluate the reconstruction performance of π^0s using the proposed algorithms and investigate the impact of expanding the cluster shapes on the reconstruction of such cases.

As the prospect of future upgrades draws near, the need to enhance and optimize data reconstruction algorithms becomes imperative. In this context, parallel architectures offer a substantial advantage in accelerating these algorithms. However, the translation from the C++ CPU-based Graph Clustering to CUDA is not a straightforward task.

The purpose of this work is to establish a starting point for exploring alternative reconstruction algorithms for calorimeter clustering that can achieve the performance levels of the current HLT2 algorithms while efficiently running on parallel architectures. By addressing this challenge, we aim to pave the way for improved data reconstruction methods in the LHCb experiment and the HEP community.

Chapter 8

Conclusions

This final chapter of the thesis gives a summary and analysis of the presented work. First, a dedicated discussion on the metrics used to evaluate the performance of the calorimeter reconstruction algorithms is presented from an engineering point of view. Then, an analysis and conclusions of the thesis is presented including minor discussions and the future lines of work derived from the thesis.

8.1 Discussion on the efficiency measure

Throughout this thesis, the performance evaluation of calorimeter reconstruction algorithms has been based on standardized metrics of efficiency and resolution. These metrics involve comparing the energy deposits of Monte Carlo photons from simulations with the corresponding reconstructed clusters. The unique design of the ECAL causes photons to deposit their entire energy in the calorimeter cells, whereas other particles may only deposit a portion of their energy and travel beyond the ECAL range. This limiting factor allows a fair comparison between the clusters energy and a Monte Carlo particle's energy only for the case of photons and electrons.

In the case of the efficiency metric, it is computed as the fraction of reconstructed particles over the number of reconstructible particles. The first condition for a Monte Carlo particle to be considered as *reconstructible* is that it must be a photon. The second criterion requires that the photon has deposited at least 90% of its energy in the calorimeter cells. This allows to exclude photons that have fallen out of the ECAL's acceptance range, particularly those located in the beam pipe hole. Then, a reconstructible Monte Carlo particle is considered reconstructed if there is a cluster that contains a minimum of 90% of the particle's deposit, as defined in Equation 8.1. This deposit is retrieved as the largest contribution of that Monte Carlo particle to a reconstructed cluster.

$$\frac{weight_{\text{MCP to cluster}}}{E_{\text{cluster}}} \geq 0.9 \tag{8.1}$$

While the current efficiency metric makes a comparison between a cluster and a specific deposit of a Monte Carlo photon, it does not account for the pileup effect. High pileup situations can occur when a photon's deposit is overlapped by deposits from other particles in neighboring or even the same cells. In such cases, if there is only a single local maximum energy deposit, the reconstructed photon will include the pileup energy from other particles, as there are no indicators within the data to discern the presence of multiple particles in that cluster.

Consequently, when comparing the energy of such clusters with the energy deposit of the corresponding Monte Carlo photon, the cluster's energy will appear higher and the fraction between the weight of the photon and the cluster's energy will be smaller, potentially falling below the acceptance threshold depending on the level of pileup. This inevitably leads to inaccurate reconstructions and affects the efficiency of the reconstruction algorithms.

High pileup cases are not uncommon in Run 3 simulations. In such cases, even under the best conditions, the current reconstruction algorithms are unable to accurately isolate the Monte Carlo photon from the pileup using the available data from the detector readout. Consequently, a calorimeter reconstruction algorithm will never be able to achieve a 100% efficiency with the current conditions if such cases are accounted as reconstructible.

Conventionally, algorithms are evaluated by comparing their performance to the best-case scenario, even if it is not practically achievable. In the context of photon reconstruction, the best-case scenario would involve correctly reconstructing even the high pileup cases. On the other hand, future upgrades to the ECAL detector are expected to reduce the size of calorimeter cells, leading to an improved energy resolution. Additionally, timing resolution will be incorporated in the upcoming Upgrade II. With this in mind, if a best-case scenario metric is employed to evaluate the algorithms performance, comparisons with future algorithms using new information from the collisions will be valid.

By maintaining the same efficiency metric, it becomes possible to assess the impact of new information on the algorithms' performance, such as the time resolution in direct comparison to the Graph Clustering or the Cellular Automaton algorithms. It will certainly provide valuable insights into the algorithms' capabilities and how they are affected by potential upgrades.

However, the efficiency metric currently used is quite generic, as it treats all cases, including pileup, overlap, clusters in boundary regions, and merged π^0s, in the same manner when contributing to the efficiency value. This limits our ability to fully comprehend how well an algorithm performs and its strengths and weaknesses in handling different complex scenarios. It would be highly valuable to evaluate the algorithms' performance separately for each specific case, since it would allow us to address the low performance scenarios by adapting the algorithms for those specific cases in order to maximize the overall performance.

Furthermore, in the context of the Upgrade II, this approach becomes an especially powerful tool for comparing algorithms that make use of different detector information. It would allow to precisely measure the impact of a higher resolution or timing information on the performance of new reconstruction algorithms when handling the specific cases.

An approach like this would require first the definition of the specific case scenarios and

the automatic selection of those among simulation events. Below, a list of relevant case scenarios is presented, as a first proposal of cases to be evaluated independently with the same efficiency:

- Overlap photons: A reconstructible Monte Carlo photon that has contributions from other Monte Carlo particles in the same cluster and each of them being less than 30% of the particles energy. This would take into account that the overlapping particles may have a major deposit in another cluster and therefore would be potentially separable by the shower overlap.
- Pileup photons: A reconstructible Monte Carlo photon that has contributions from other Monte Carlo particles in the same cluster and at least one of them being more than 30% of the particles energy. This number should be optimized by studying the effect on the photon cluster through simulation cases. This would account for any particle with a major contribution of its energy to the cluster as pileup. However, the pileup effect may vary depending on the energy ratio between the photon and the pileup particle. If the energy of the pileup particle is small compared to the photon, the cluster could potentially be correctly reconstructed. Such cases would still be accounted in this case scenario.
- Merged π^0s: Two reconstructible Monte Carlo photons that come from the same π^0 mother and that its track extrapolations to the ECAL z position are separated less than 1.5 of the Inner cell size. This case would only be filtered by the particle type without taking into account overlap or pileup conditions defined previously.
- Boundary photons: A reconstructible Monte Carlo photon in which the associated cluster has deposits in two different regions (Inner - Middle or Middle - Outer). This case would not take into account overlap, pileup or merged π^0 conditions defined previously.

In order to implement this, a new efficiency test should be assessed, based on the current calorimeter reconstruction efficiency checker for HLT2. It would require to select a set of simulation events including photons in a wide energy range, similar to the $B^0 \to K^*\gamma$ simulations and also π^0 instances such as the ones from $B^0 \to \pi^+\pi^-\pi^0$. Moreover, the occurrences of the mentioned cases in the data-set should be checked to ensure there is a significant amount of each one.

Within this test, all the Monte Carlo particles from the simulation data-set would be evaluated using the previous definitions of a reconstructible particle. Then, there should be an evaluation of each Monte Carlo particle in order to check if it belongs to one of the four selected cases. Each case would be evaluated with the global criteria of reconstructed particles and accounted as a reconstructed particle in the global efficiency metric and in the specific case efficiency metric.

This would allow us to have a global efficiency evaluated for all of the reconstructible particles and also a segmented efficiency for all of the specific cases using the same data-set. Comparing those would give us a better understanding of the performance of any

reconstruction algorithm and highlight the differences between the critical specific cases so that they can be further addressed.

In summary, a tool to have a more detailed evaluation of the efficiency of reconstruction algorithms would add valuable information when comparing algorithms with different detector's information, granularity or cluster shapes. Which would in general help to provide an overall better performance of the current and future calorimeter clustering algorithms.

8.2 Conclusions and future work

The LHCb experiment at CERN underwent a major hardware and software upgrade that was culminated through the commissioning year in 2022. In this upgrade, the electromagnetic and the hadronic calorimeters had to entirely change its readout electronics to adapt to the increased 40 MHz event rate for the Run 3 conditions.

Prior to data taking, there are many complex engineering challenges involving the design, control and operation of such detectors. One of the key aspects to accomplish accurate measurements from the calorimeter detectors is the time alignment of each of its almost $10,000$ channels. This alignment consists on adjusting the phase of the integrator of each channel to ensure the entire energy shower from the particles is captured in the collision time frame of 25 ns. This is achieved through a process of taking data in a special timing and trigger conditions in stable collisions and its analysis, as well as a prior analysis of the shape of the integrator signal with detector data.

Starting with the adaptation of the previously used time alignment procedure for Run 3, the work presented in Chapter 4 comprises the validation of the method and the process of time aligning all the ECAL and HCAL channels. Through this process there have been some expected inconveniences, such as high voltage updates slightly altering the channel's time alignment, and some unexpected ones, such as an issue with the clock propagation in the control boards or the stopping of data taking caused by issues with other sub-detectors or with the LHC. Even so, experiencing part of the commissioning process firsthand has been an extremely enriching experience that tests one's abilities to make quick decisions and adapt to new situations.

Moving to the software implications of the LHCb Upgrade I, the full software trigger system in the RTA framework enhances the need to optimize and accelerate the reconstruction algorithms to cope with the HLT throughput requirements while providing an offline-quality reconstruction at the same time. At the start of this thesis, the ECAL reconstruction was the fourth most time consuming algorithm of the HLT2 reconstruction sequence. Therefore, the study of alternatives to optimize the calorimeter clustering process has been the main motif of this thesis.

Deep learning techniques have demonstrated to be efficient in many complex problems related to high energy physics. However, rather than using deep convolutional neural network architectures to perform the calorimeter clustering at once, the first approach presented in this thesis decomposes the clustering problem into smaller steps that can be modeled as

the behavior of a cellular automaton. It has been demonstrated that each step can be learned by a simple convolutional neural network structure. Given that the nature of the formulation is independent from the numerical values, the networks can be trained using randomly generated data and a reduced data-set produced from LHCb simulations.

The resulting architecture is a sequence of two convolutional neural networks that take the ECAL digits as an image in the input and outputs a stack of 3×3 pixel images of the reconstructed clusters found in the input image. The results obtained in terms of energy resolution are good but are not comparable to the resolution of the benchmark Cellular Automaton algorithm. However, it has been proven that the learning of the general formulation is effectively done and the inference time of the trained networks have a nearly constant response with independence of the number of active digits in the calorimeter image.

Further improvements on this approach were stopped due to the bottleneck of the neural network inference inside the LHCb framework. More recent studies have started to work on testing inference engines inside the HLT1 framework using custom GPU tools, which starts to bring light into an efficient deep learning inference in high throughput systems like the HLT. As the inference engines are also rapidly evolving, deep learning models are expected to be key tools for the increasing complexity of data reconstruction in high energy physics experiments.

Future work regarding this approach is segmented into two different lines. The first one involves testing the network models inside the LHCb framework. This would allow to deeply compare the algorithms performance and inference time while also comparing different inference engines in both HLT1 and HLT2 frameworks. The second line would be dedicated to improve the reconstruction quality of the clustering and overlap model. One of the key aspects would be to use images of the Monte Carlo energy deposits as the output training data of the network, instead of the result of applying the defined formulation to the input data. Other than that, merged π^0 samples should be added to the training data-set, in order to correctly reconstruct or even detect those cases. Graph neural network architectures have also a great potential to model the different granularity of the ECAL in the boundary regions and could help in future improvements of this approach.

Exploiting the idea to use graphs to model the neighbouring geometry of the ECAL, the second approach to calorimeter reconstruction presented in this work is named the Graph Clustering algorithm. This algorithm implements the same logic processes as the benchmark Cellular Automaton but using graph data structures to store the event digits, which accelerates the clustering and overlap processes. Results in terms of performance show a slightly greater efficiency than the previously used algorithm, with consistent energy and position resolutions. The impact of the Graph Clustering in the HLT2 throughput implies an overall increase of 4.09%, as well as in comparison to the Cellular Automaton, where it outperforms the previously used method by 65.4% in execution time on average.

Graph Clustering is the ECAL reconstruction algorithm used for Run 3. Moreover, it has the potential to be used in further upgrades of the calorimeter. Due to the flexible definition of neighbourhood of the graphs, the algorithm could be easily adapted to use different cluster shapes in different regions. This also includes being able to define bigger clusters in case of an

increased cell granularity. In case of an increased occupancy of the detector, the most critical part of the algorithm would be the analysis of the connected components, since its average size per event could increase the computational complexity of the algorithm. However, since the analysis of each one is independent, the process could be parallelized.

As explained in Section 6.2.3, the groupings of connected components obtained from the graph structures for each event show interesting patterns in terms of how the nodes are distributed and connected. This sparks curiosity about whether there's a connection between the shapes of these cluster graphs and the types of particles they represent. While this kind of study would be even more meaningful with a finer granularity calorimeter, it's still intriguing to explore whether there might be certain graph patterns that could be linked to merged π^0 particles or jets, for instance.

In the context of the Upgrade II calorimeter, the increased luminosity will certainly increment the number of digits per event, implying an increase on the overlap and pile-up effects. However, the increased granularity of the inner part of the detector may improve the cluster separation. Although the complexity curve of the current graph clustering algorithm will not be the same in prospects of the upgraded calorimeter, it is certainly a very challenging opportunity to test the limits of the proposed algorithm.

The last contribution from this thesis comprises a first effort on improving the CUDA ECAL reconstruction algorithm in the HLT1 framework. In order to increase its reconstruction efficiency, the same shower overlap logic used in the Graph Clustering is added into the ECAL clustering reconstruction sequence using three different strategies to implement it. Each of them results in an overall increased efficiency with a trade-off in the throughput. The best candidate achieves an efficiency range compatible with the Graph Clustering within errors with a throughput impact of 2.8% decrease of the HLT2 reconstruction sequence.

This work is meant to be a starting point on the improvement of the HLT1 ECAL reconstruction. Future efforts should be put in improving the reconstruction of the merged π^0s using cluster expansion strategies as in the Graph Clustering. Other strategies would rely on testing the adaptability of graph data structures in parallel architectures to replicate the Graph Clustering algorithm in CUDA.

In conclusion, this thesis has contributed to the LHCb commissioning tasks and Real-Time Analysis software for the Run 3, expanding the reconstruction alternatives to the calorimeter clustering problem. Despite the challenges faced by artificial intelligence methods on its inference, the presented Graph Clustering algorithm demonstrated a remarkable improvement in computational complexity for the calorimeter reconstruction problem. Its inherent flexibility also promises good prospects for future upgrades. Overall, this work has contributed to the growth of the LHCb experiment and has also left a valuable impact on software for the High Energy Physics community.

Appendix A

Graph Clustering performance

The Graph Clustering algorithm for ECAL reconstruction in HLT2 is evaluated in terms of performance in Section 6.3, compared to the Cellular Automaton method. Since it is now the default solution for ECAL reconstruction in HLT2, this appendix provides an additional set of benchmark plots regarding efficiency and resolution performance of the Graph Clustering that can be compared to the Cellular Automaton performance in Upgrade conditions [78].

Plots using γ samples, $40,000$ Monte Carlo events of $B^0 \to K^*\gamma$ decays are used. For the π^0 samples, $100,000$ Monte Carlo events of $B^0 \to \pi^+\pi^-\pi^0$ decays are used, both in Run 3 conditions.

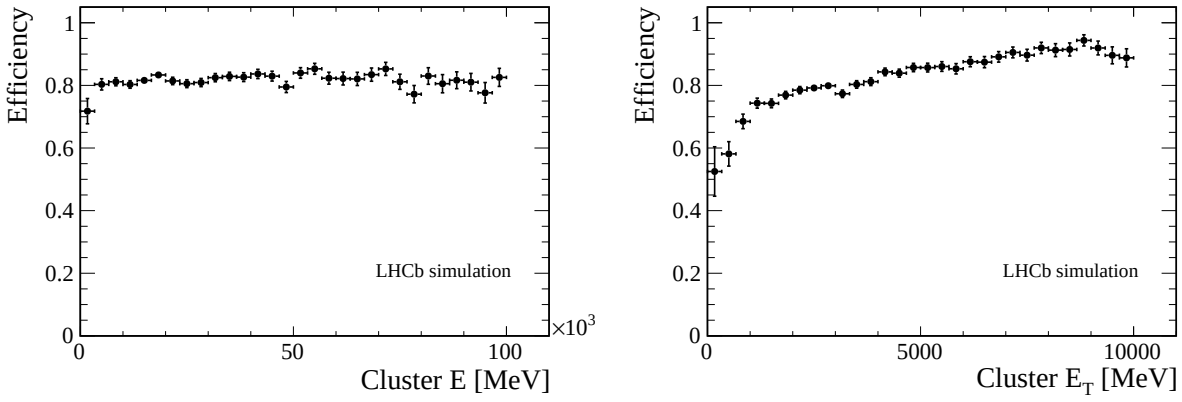


Figure A.1: ECAL cluster reconstruction efficiency versus energy E and transverse energy E_T using photon hypothesis with Run 2 corrections.

Further analysis study the comparison of both methods in terms of resolution evaluating the three ECAL regions separately in Figures A.5, A.6 and A.7.

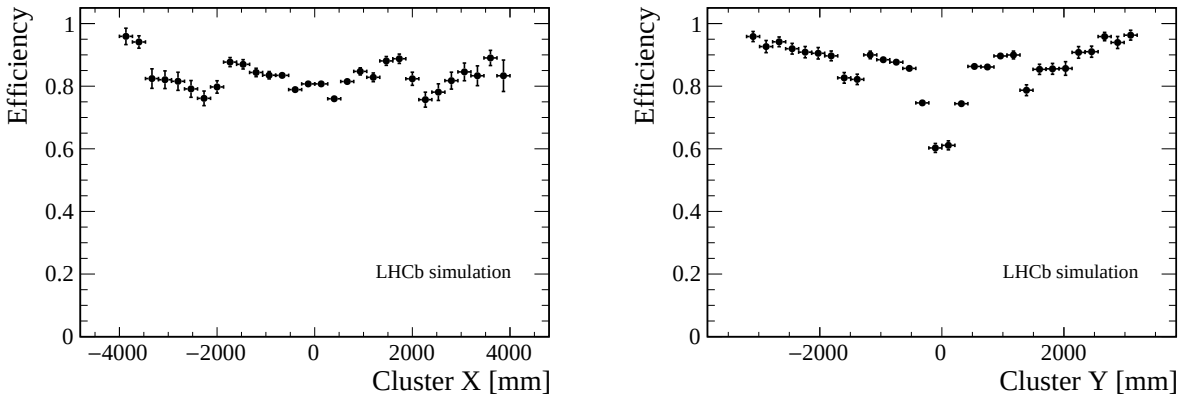


Figure A.2: ECAL cluster reconstruction efficiency versus position in the ECAL X and Y using photon hypothesis with Run 2 corrections.

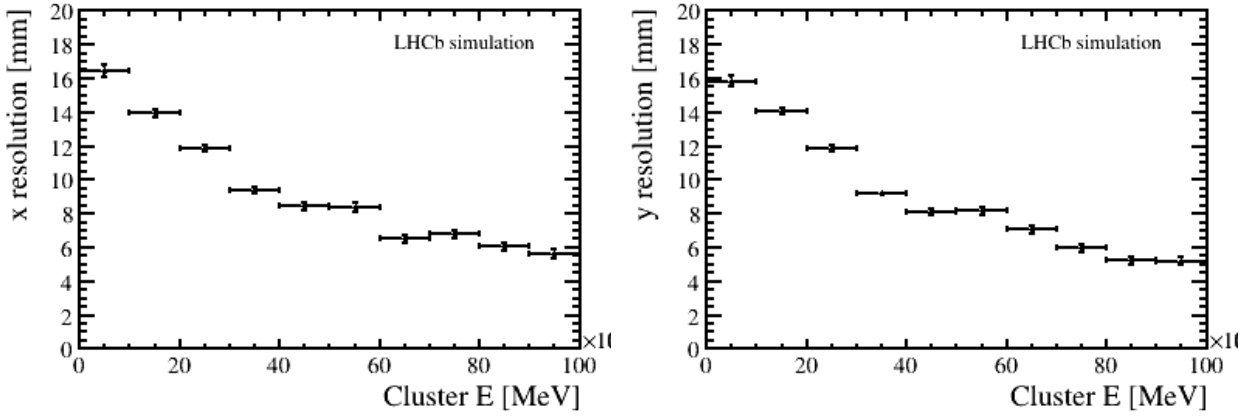


Figure A.3: ECAL cluster (left) X position and (right) Y position resolution versus energy for reconstructible photons from B decays using Run 2 corrections.

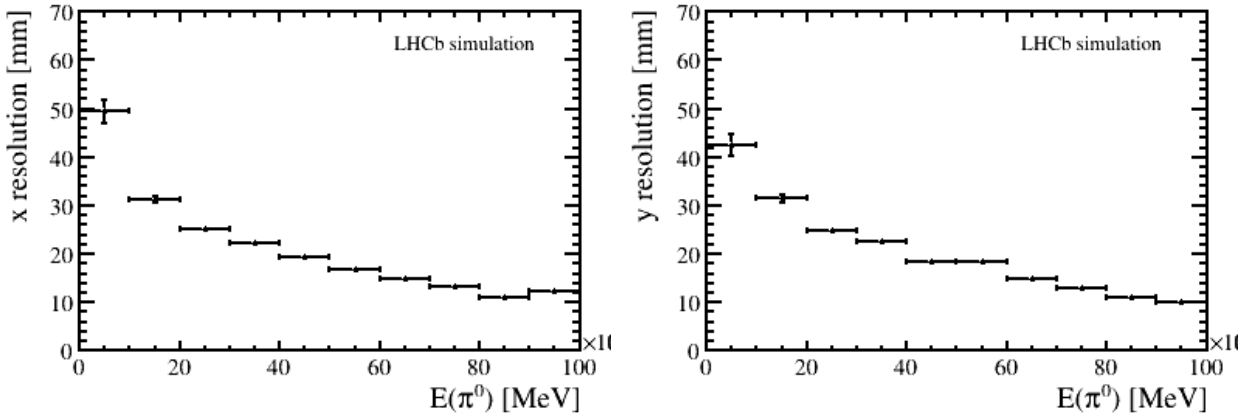
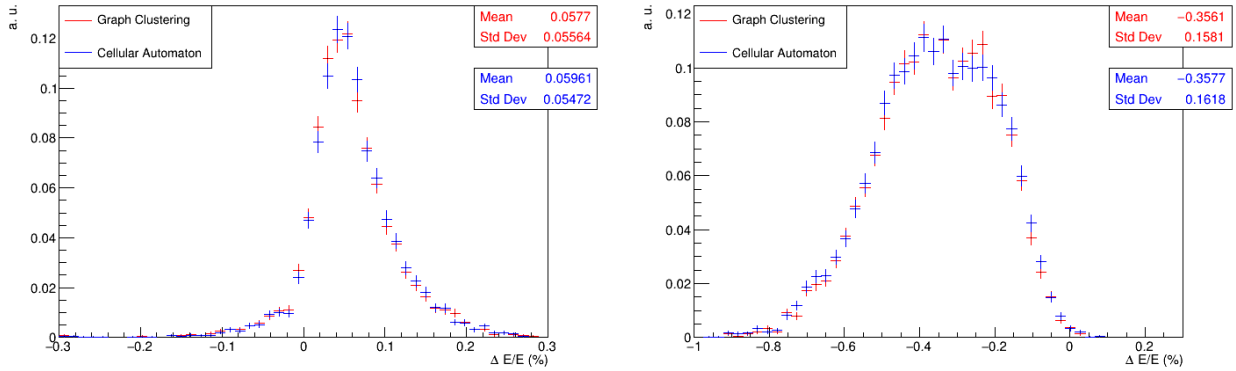
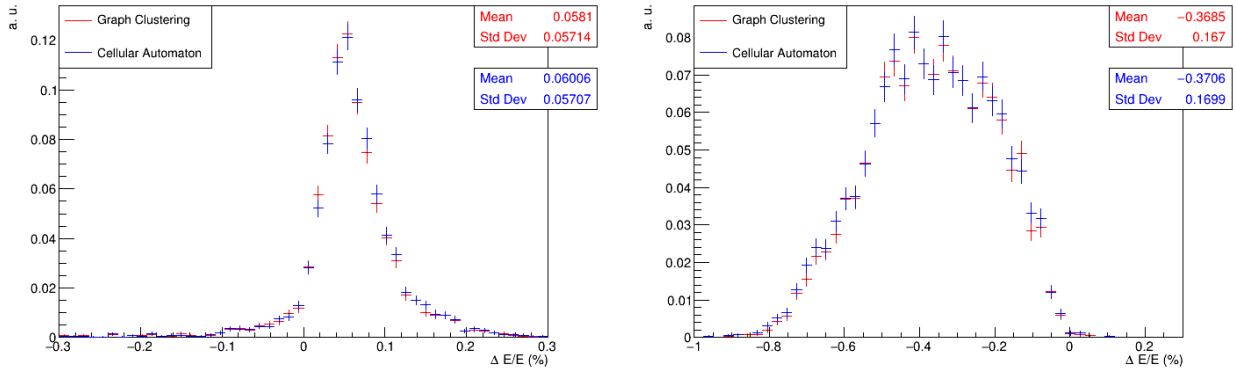


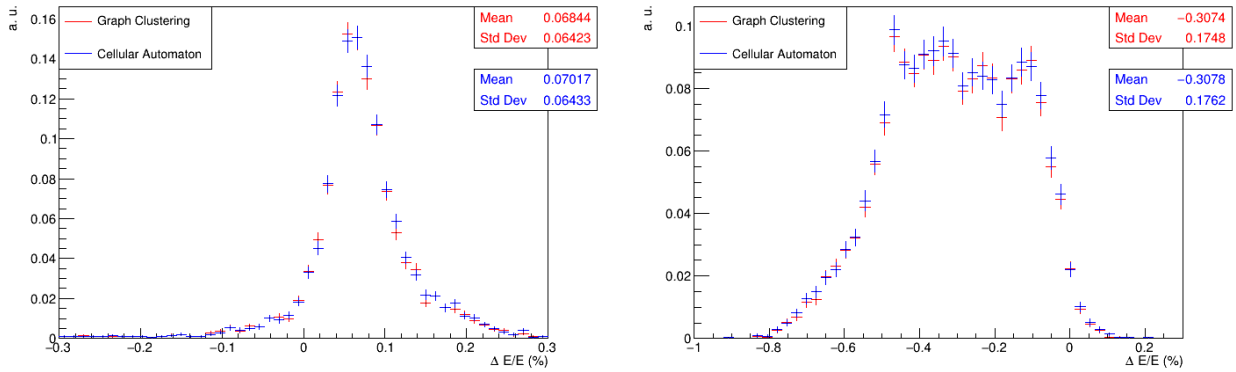
Figure A.4: Merged π^0 hypothesis (left) X position and (right) Y position resolution versus energy for $\pi^0 \to \gamma\gamma$ from B decays using Run 2 corrections.



(a) Inner region.

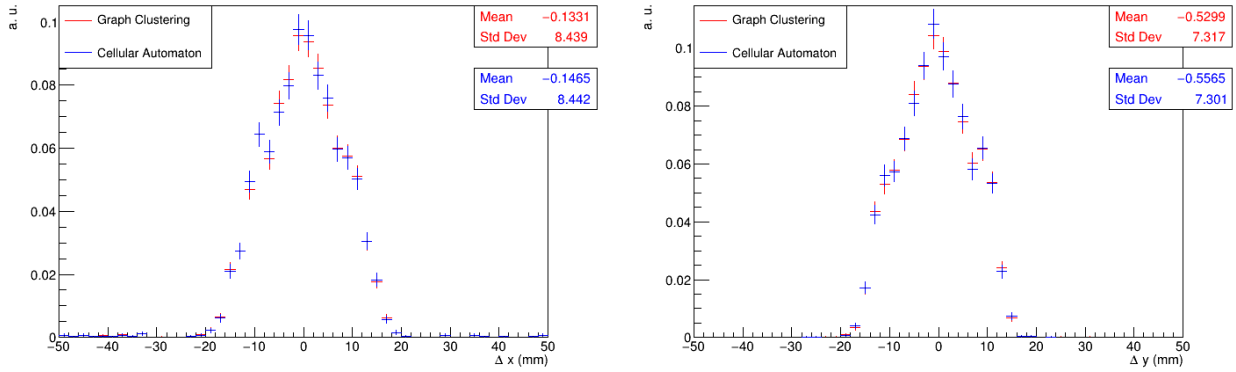


(b) Middle region.

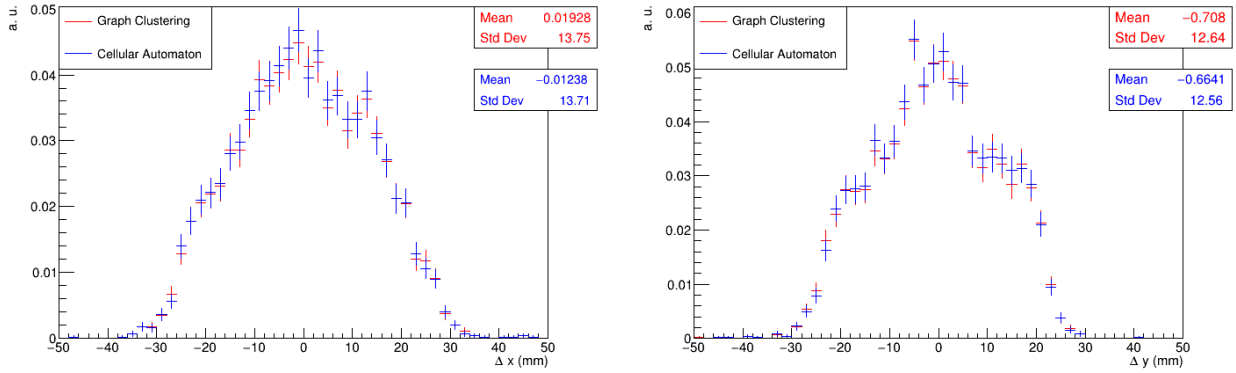


(c) Outer region.

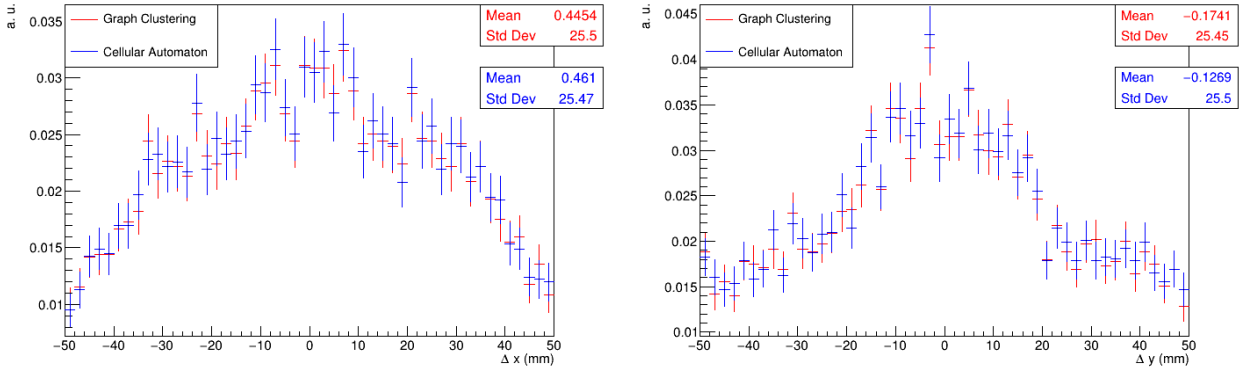
Figure A.5: Energy resolution for the three regions using γ samples (left) and π^0 samples (right), both without cluster corrections.



(a) Inner region.

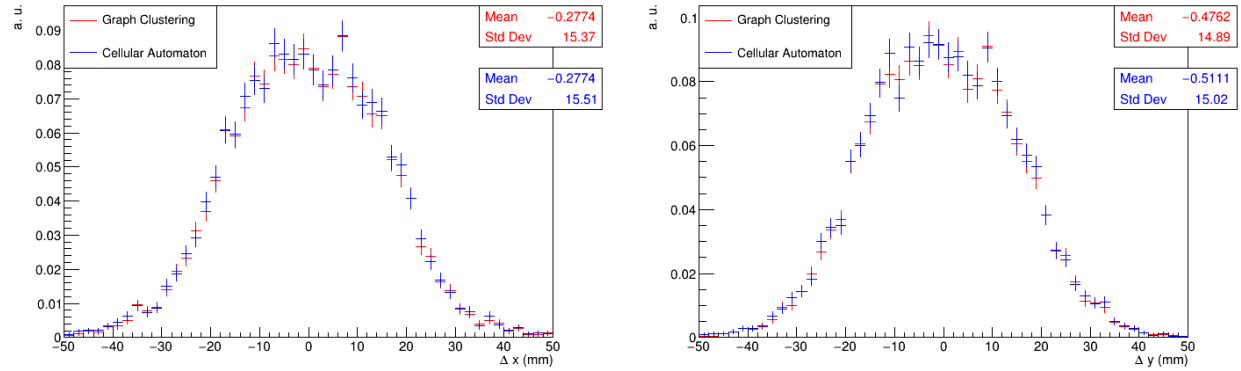


(b) Middle region.

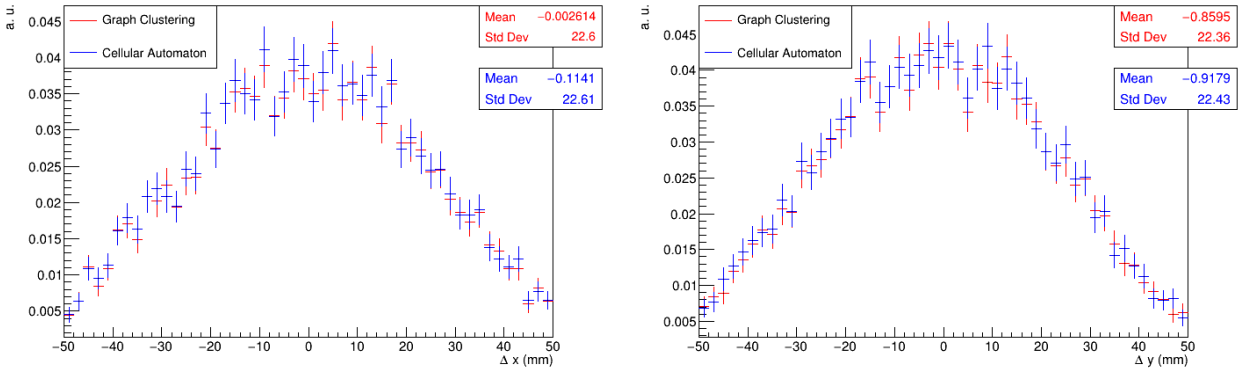


(c) Outer region.

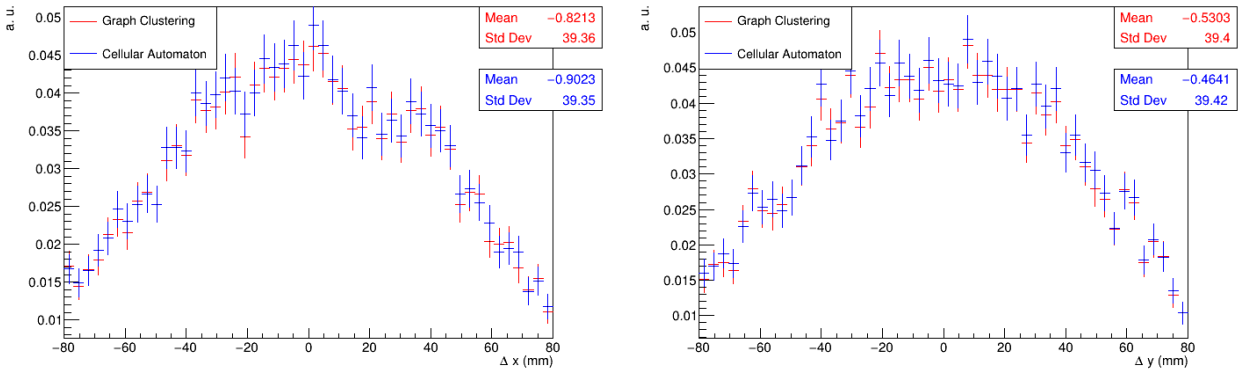
Figure A.6: X and Y ECAL position resolution for the three regions using γ samples without cluster corrections.



(a) Inner region.



(b) Middle region.



(c) Outer region.

Figure A.7: X and Y ECAL position resolution for the three regions using π^0 samples without cluster corrections.

Bibliography

[1] L. Evans and P. Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[2] A. A. Alves Jr. et al. "The LHCb detector at the LHC". In: *Journal of Instrumentation* 3.LHCb-DP-2008-001 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[3] A. D. Sakharov. "Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe". In: *Soviet Physics Uspekhi* 34.5 (1991), p. 392. DOI: 10.1070/PU1991v034n05ABEH002497.

[4] LHCb Experiment LHCb Collaboration. *Future physics potential of LHCb*. Tech. rep. LHCb-PUB-2022-012, CERN-LHCb-PUB-2022-012. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2806113.

[5] LHCb collaboration. "Framework TDR for the LHCb Upgrade: Technical Design Report". In: CERN-LHCC-2012-007 (2012).

[6] LHCb collaboration. "LHCb Trigger and Online Upgrade Technical Design Report". In: CERN-LHCC-2014-016 (2014).

[7] R. Aaij et al. "The LHCb trigger and its performance in 2011". In: *Journal of Instrumentation* 8.04 (2013), P04022–P04022. DOI: 10.1088/1748-0221/8/04/p04022.

[8] R. Aaij et al. "Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC". In: *Journal of Instrumentation* 14.04 (2019), P04013–P04013. DOI: 10.1088/1748-0221/14/04/p04013.

[9] R. Aaij et al. "A comprehensive real-time analysis model at the LHCb experiment". In: *Journal of Instrumentation* 14.04 (2019), P04006. DOI: 10.1088/1748-0221/14/04/P04006.

[10] N. Valls Canudas et al. "A Deep Learning approach to LHCb Calorimeter reconstruction using a Cellular Automaton". In: *EPJ Web of Conferences* 251 (2021), p. 04008. DOI: 10.1051/epjconf/202125104008. URL: https://cds.cern.ch/record/2814342.

[11] N. Valls Canudas et al. "Use of Deep Learning to Improve the Computational Complexity of Reconstruction Algorithms in High Energy Physics". In: *Applied Sciences* 11.23 (2021), p. 11467. DOI: 10.3390/app112311467.

[12] N. Valls Canudas et al. *Reconstruction of the LHCb Calorimeter using Machine Learning: lessons learned*. Paper presented at the 25th International Conference of the Catalan Association for Artificial Intelligence (CCIA 2023), Món Sant Benet, Barcelona, Spain. 2023.

[13] N. Valls Canudas et al. "Graph Clustering: a graph-based clustering algorithm for the electromagnetic calorimeter in LHCb". In: *The European Physical Journal C* 83.2 (2023), p. 179. DOI: 10.1140/epjc/s10052-023-11332-1.

[14] N. Valls Canudas et al. *Preliminary Performance Study of an Alternative Calorimeter Clustering Solution for Allen in LHCb*. Presented at the 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP23), Norfolk, Virginia, USA. 2023.

[15] E. Mobs. "The CERN accelerator complex. Complexe des accélérateurs du CERN". In: OPEN-PHO-ACCEL-2016-009-2 (2016). General Photo. URL: https://cds.cern.ch/record/2197559.

[16] LHCb Collaboration. "Framework TDR for the LHCb Upgrade II Opportunities in flavour physics, and beyond, in the HL-LHC era". In: CERN-LHCC-2021-012, LHCB-TDR-023 (2021). URL: https://cds.cern.ch/record/2776420.

[17] LHCb collaboration. "LHCb reoptimized detector design and performance: Technical Design Report". In: CERN-LHCC-2003-030 (2003).

[18] R. Aaij et al. "LHCb detector performance". In: *International Journal of Modern Physics* A30 (2015), p. 1530022. DOI: 10.1142/S0217751X15300227. arXiv: 1412.6352 [hep-ex].

[19] LHCb collaboration. "LHCb Tracker Upgrade Technical Design Report". In: CERN-LHCC-2014-001 (2014).

[20] LHCb collaboration. "LHCb magnet: Technical Design Report". In: CERN-LHCC-2000-007 (2000).

[21] P. Billoir et al. "A parametrized Kalman filter for fast track fitting at LHCb". In: *Computer Physics Communications* 265 (2021), p. 108026. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2021.108026.

[22] LHCb collaboration. "LHCb VELO Upgrade Technical Design Report". In: CERN-LHCC-2013-021 (2013).

[23] A. Hennequin et al. "A fast and efficient SIMD track reconstruction algorithm for the LHCb upgrade 1 VELO-PIX detector". In: *Journal of Instrumentation* 15.06 (2020), P06018–P06018. DOI: 10.1088/1748-0221/15/06/p06018.

[24] R. Matthew Scott. "The LHCb Upstream Tracker Upgrade". In: *Proceedings of Science* Vertex2019 (2020), p. 013. DOI: 10.22323/1.373.0013.

[25] P. A. Cherenkov. "Visible luminescence of pure liquids under the influence of γ-radiation". In: *Doklady Akademii Nauk SSSR* 2.8 (1934), pp. 451–454. DOI: 10.3367/UFNr.0093.196710n.0385.

[26] LHCb collaboration. "LHCb PID Upgrade Technical Design Report". In: CERN-LHCC-2013-022 (2013).

[27] LHCb collaboration. "LHCb RICH: Technical Design Report". In: CERN-LHCC-2000-037 (2000).

[28] LHCb collaboration. "LHCb calorimeters: Technical Design Report". In: CERN-LHCC-2000-036 (2000).

[29] LHCb collaboration. "LHCb muon system: Technical Design Report". In: CERN-LHCC-2001-010 (2001).

[30] R. Aaij et al. "Tesla: An application for real-time data analysis in High Energy Physics". In: *Computer Physics Communications* 208 (2016), pp. 35–42. DOI: 10.1016/j.cpc.2016.07.022.

[31] "RTA and DPA dataflow diagrams for Run 1, Run 2, and the upgraded LHCb detector ". In: LHCB-FIGURE-2020-016 (2020). URL: https://cds.cern.ch/record/2730181.

[32] LHCb collaboration. "LHCb Upgrade GPU High Level Trigger Technical Design Report". In: CERN-LHCC-2020-006 (2020).

[33] R. Aaij et al. "Allen: A High-Level Trigger on GPUs for LHCb". In: *Computing and Software for Big Science* 4.1 (2020). DOI: 10.1007/s41781-020-00039-7.

[34] LHCb Collaboration. "Throughput and resource usage of the LHCb upgrade HLT". In: LHCB-FIGURE-2020-007 (2020). URL: https://cds.cern.ch/record/2715210.

[35] P. Billoir et al. "A parametrized Kalman filter for fast track fitting at LHCb". In: *Computer Physics Communications* 265 (2021), p. 108026. DOI: 10.1016/j.cpc.2021.108026.

[36] LHCb collaboration. "The LHCb upgrade I". In: LHCb-DP-2022-002 (2023). DOI: 10.48550/arXiv.2305.10515. arXiv: 2305.10515 [hep-ex].

[37] E. Picatoste et al. "Low noise front end ICECAL ASIC for the upgrade of the LHCb calorimeter". In: *Journal of Instrumentation* 7.01 (2012), p. C01080. DOI: 10.1088/1748-0221/7/01/C01080.

[38] V. Breton, N. Brun, and P. Perret. *A clustering algorithm for the LHCb electromagnetic calorimeter using a cellular automaton*. Tech. rep. CERN-LHCb-2001-123, 2001.

[39] P. Perret. "Electromagnetic cluster reconstruction in LHCb". In: *2010 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA 2010)*. 2010, pp. 1–6. DOI: 10.1109/CNNA.2010.5430344.

[40] J. Neumann, A. W. Burks, et al. *Theory of self-reproducing automata*. Vol. 1102024. University of Illinois press Urbana, 1966.

[41] A. Vallier. "Measurement of the CKM angle gamma in the $B^0 \to DK^{*0}$ decays using the Dalitz method in the LHCb experiment at CERN and photon reconstruction optimisation for the LHCb detector upgrade." 2015. URL: https://cds.cern.ch/record/2144873.

[42] O. Deschamps et al. *Photon and neutral pion reconstruction*. Tech. rep. Geneva: CERN, 2003. URL: https://cds.cern.ch/record/691634.

[43] E. Tarkovsky. "The HERA-B electromagnetic calorimeter". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 379.3 (1996). Proceedings of the Sixth International Conference on Instrumentation for Experiments at e+ e- Colliders, pp. 515–517. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(96)00558-X.

[44] Edward P. Hartouni et al. *HERA-B: An experiment to study CP violation in the B system using an internal target at the HERA proton ring. Design report*. Tech. rep. DESY-PRC-95-01. 1995. URL: https://cds.cern.ch/record/1478622.

[45] D. Galli et al. "The reconstruction for the Electromagnetic Calorimeter of the Hera-B experiment". In: (1997). URL: http://www-hera-b.desy.de/subgroup/detector/ecal/publications/calor97_villa.ps.

[46] P.A. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice/Hall International, 1982. ISBN: 9780136542360. URL: https://books.google.es/books?id=Em9QAAAAMAAJ.

[47] M. Aleksa et al. *ATLAS Liquid Argon Calorimeter Phase-I Upgrade: Technical Design Report*. Tech. rep. CERN-LHCC-2013-017, ATLAS-TDR-022. Final version presented to December 2013 LHCC. 2013. URL: https://cds.cern.ch/record/1602230.

[48] G. Aad et al. "Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1". In: *The European Physical Journal C* 77.7 (2017). DOI: 10.1140/epjc/s10052-017-5004-5.

[49] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[50] "CMS: The electromagnetic calorimeter. Technical design report". In: CERN-LHCC-97-33, CMS-TDR-4 (Dec. 1997).

[51] G. L. Bayatian et al. "CMS Physics: Technical Design Report Volume 1: Detector Performance and Software". In: CERN-LHCC-2006-001, CMS-TDR-8-1, CERN-LHCC-2006-001, CMS-TDR-8-1 (2006).

[52] F. Carminati et al. "ALICE: Physics Performance Report, Volume I". In: *Journal of Physics G: Nuclear and Particle Physics* 30.11 (2004), p. 1517. DOI: 10.1088/0954-3899/30/11/001.

[53] A. Fantoni and (On behalf ofthe ALICE collaboration). "The ALICE Electromagnetic Calorimeter: EMCAL". In: *Journal of Physics: Conference Series* 293.1 (2011), p. 012043. DOI: 10.1088/1742-6596/293/1/012043.

[54] ALICE Collaboration. *Performance of the ALICE Electromagnetic Calorimeter*. 2022. arXiv: 2209.04216 [physics.ins-det].

[55] Y. S. Amhis. "Time alignment of the electromagnetic and hadronic calorimeters, reconstruction of the $B \to D^- \rho(770)^+, B_s \to D_s^- \rho(770)^+$ and $B_s \to D_s^- K^{*+}(892)$ decay channels with the LHCb detector". 2009. URL: https://cds.cern.ch/record/1210675.

[56] C. Abellan Beteta et al. "Time alignment of the front end electronics of the LHCb calorimeters." In: *Journal of Instrumentation* 7 (2012), P08020. DOI: 10.1088/1748-0221/7/08/P08020.

[57] M Adinolfi et al. "LHCb data quality monitoring". In: *Journal of Physics: Conference Series* 898.9 (2017), p. 092027. DOI: 10.1088/1742-6596/898/9/092027.

[58] N. Valls Canudas et al. "Deep Learning approach to LHCb Calorimeter reconstruction using a Cellular Automaton". In: *EPJ Web of Conferences*. Vol. 251. EDP Sciences. 2021, p. 04008. DOI: 10.1051/epjconf/202125104008.

[59] V. Breton et al. "Application of neural networks and cellular automata to interpretation of calorimeter data". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 362.2-3 (1995), pp. 478–486. DOI: 10.1016/0168-9002(95)00217-0.

[60] M. Casolino and P. Picozza. "A cellular automaton to filter events in a high energy physics discrete calorimeter". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 364.3 (1995), pp. 516–523. DOI: 10.1016/0168-9002(95)00520-X.

[61] B. Denby. "Neural networks and cellular automata in experimental high energy physics". In: *Computer Physics Communications* 49.3 (1988), pp. 429–448. DOI: 10.1016/0010-4655(88)90004-5.

[62] C. Baldanza et al. "A cellular neural network for peak finding in high-energy physics". In: *Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA 2000)(Cat. No. 00TH8509)*. IEEE. 2000, pp. 443–448. DOI: 10.1109/CNNA.2000.877369.

[63] Y. LeCun et al. "Handwritten digit recognition with a back-propagation network". In: *Advances in neural information processing systems* 2 (1989), pp. 396–404.

[64] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[65] Graeme Niedermayer. "Investigations of calorimeter clustering in ATLAS using machine learning". PhD thesis. 2017. URL: https://dspace.library.uvic.ca/handle/1828/8970.

[66] M. Mazurek. *Deep learning solutions for 2D calorimetric cluster reconstruction at LHCb*. Tech. rep. LHCb-TALK-2020-178. 2020.

[67] J. Redmon and A. Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018). DOI: 10.48550/arXiv.1804.02767.

[68] W. Gilpin. "Cellular automata as convolutional neural networks". In: *Physical Review E* 100.3 (2019), p. 032402. DOI: 10.1103/PhysRevE.100.032402.

[69] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pp. 807–814.

[70] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274.

[71] G. Van Rossum and F.L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[72] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[73] A. Hoecker and P. Speckmayer et al. *TMVA - Toolkit for Multivariate Data Analysis*. 2009. arXiv: physics/0703039 [physics.data-an].

[74] M. De Cian et al. *Fast neural-net based fake track rejection in the LHCb reconstruction*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2255039.

[75] J. Bai, F. Lu, K. Zhang, et al. *ONNX: Open Neural Network Exchange*. https://github.com/onnx/onnx. 2019.

[76] NVIDIA TensorRT. *NVIDIA TensorRT Programmable Inference Accelerator*. 2020.

[77] A. Sclocco et al. *High-Throughput Machine Learning Inference with NVIDIA TensorRT*. Presented at the 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP23), Norfolk, Virginia, USA. 2023.

[78] LHCb Collaboration. "Selected HLT2 reconstruction performance for the LHCb upgrade". In: LHCB-FIGURE-2021-003 (2021). URL: https://cds.cern.ch/record/2773174.

[79] *C++ Graph Clustering algorithm code*. https://gitlab.cern.ch/lhcb/Rec/-/blob/master/CaloFuture/CaloFutureReco/src/GraphClustering.cpp. Accessed: 25-09-2023.

[80] S. Lloyd. "Least squares quantization in PCM". In: *IEEE Transactions on Information Theory* 28.2 (1982), pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

[81] J. Han, J. Pei, and M Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[82] M. Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *kdd*. Vol. 96. 34. 1996, pp. 226–231. DOI: 10.5555/3001460.3001507.

[83] Mihael Ankerst et al. "OPTICS: Ordering points to identify the clustering structure". In: *ACM Sigmod record* 28.2 (1999), pp. 49–60. DOI: 10.1145/304181.304187.

[84] M. Rovere et al. "CLUE: a fast parallel clustering algorithm for high granularity calorimeters in high-energy physics". In: *Frontiers in big Data* 3 (2020), p. 591315. DOI: 10.3389/fdata.2020.591315.

[85] G. Mavromanolakis. "Calorimeter clustering with minimal spanning trees". In: *arXiv preprint physics/0409039* (2004). DOI: 10.48550/arXiv.physics/0409039.

[86] F. Scarselli et al. "The graph neural network model". In: *IEEE transactions on neural networks* 20.1 (2008), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[87] X. Ju et al. "Graph neural networks for particle reconstruction in high energy physics detectors". In: *arXiv preprint arXiv:2003.11603* (2020). DOI: 10.48550/arXiv.2003.11603.

[88] S. R. Qasim et al. "Multi-particle reconstruction in the High Granularity Calorimeter using object condensation and graph neural networks". In: *EPJ Web of Conferences*. Vol. 251. EDP Sciences. 2021, p. 03072. DOI: 10.1051/epjconf/202125103072.

[89] S. R. Qasim et al. "Learning representations of irregular particle-detector geometry with distance-weighted graph networks". In: *The European Physical Journal C* 79.7 (2019), pp. 1–11. DOI: 10.1140/epjc/s10052-019-7113-9.

[90] D. R. Musser. "Introspective sorting and selection algorithms". In: *Software: Practice and Experience* 27.8 (1997), pp. 983–993. DOI: 10.1002/(SICI)1097-024X(199708)27:8%3C983::AID-SPE117%3E3.0.CO;2-%23.

[91] M. Ginsberg. *Essentials of artificial intelligence*. Newnes, 2012, pp. 52–56.

[92] T. H. Cormen et al. *Introduction to algorithms*. MIT press, 2022.

[93] LHCb Collaboration. *Upgrade Software and Computing*. Tech. rep. CERN-LHCC-2018-007, LHCB-TDR-017. Geneva: CERN, 2018. URL: https://cds.cern.ch/record/2310827.

[94] G. Barrand et al. "GAUDI—A software architecture and framework for building HEP data processing applications". In: *Computer Physics Communications* 140.1-2 (2001), pp. 45–55. DOI: 10.1016/S0010-4655(01)00254-5.

[95] D. H. Cámpora Pérez, N. Neufeld, and A. Riscos Núñez. "Search by triplet: An efficient local track reconstruction algorithm for parallel architectures". In: *Journal of Computational Science* 54 (2021), p. 101422. ISSN: 1877-7503. DOI: https://doi.org/10.1016/j.jocs.2021.101422.

[96] P. Fernandez Declara et al. "A Parallel-Computing Algorithm for High-Energy Physics Particle Tracking and Decoding Using GPU Architectures". In: *IEEE Access* 7 (2019), pp. 91612–91626. DOI: 10.1109/ACCESS.2019.2927261.

[97] T. Reis for the CMS Collaboration. "Developing GPU-compliant algorithms for CMS ECAL local reconstruction during LHC Run 3 and Phase 2". In: *Journal of Physics: Conference Series* 2438.1 (2023), p. 012027. DOI: 10.1088/1742-6596/2438/1/012027.

[98] Z. Chen et al. "GPU-based Clustering Algorithm for the CMS High Granularity Calorimeter". In: *EPJ Web of Conferences* 245 (2020). Ed. by C. Doglioni et al., p. 05005. DOI: 10.1051/epjconf/202024505005.