Algorithms for the Level 1 Trigger Primitives Generation in the High-Granularity Calorimeter at the High-Luminosity LHC

Jean-Baptiste Sauvan

Ecole polytechnique
Institut Polytechnique de Paris

MÉMOIRE

Présenté pour obtenir
l'habilitation à diriger des recherches

par

Jean-Baptiste SAUVAN

Algorithms for the Level 1 Trigger Primitives Generation in the High-Granularity Calorimeter at the High-Luminosity LHC

Soutenu le 29 septembre 2022
devant la commission d'examen composée de :

Olivier	**DRAPIER**	Rapporteur
Laurent	**SERIN**	Rapporteur
Isabelle	**WINGERTER-SEEZ**	Rapporteuse
Emmanuel	**PEREZ**	Président du jury
Alexander	**TAPPER**	Examinateur

Contents

Foreword

A large part of my research activity in the last ten years has been devoted to the development of algorithms for Level 1 (L1) trigger systems in CMS. Between 2013 and 2015 I worked on the Phase 1 upgrade of the calorimeter trigger, on the development of e/γ and hadronic tau trigger algorithms. These algorithms have been running during the Run 2 of the LHC and now for the Run 3. Detailed information on these algorithms and on the L1 trigger of CMS can be found in [1, 2, 3, 4].

Since 2014 I have been working on the design of the trigger primitives generation (TPG) system of the High-Granularity Calorimeter (HGCAL), the future endcap calorimeter developed for the Phase 2 upgrade of CMS. This activity started before the HGCAL project was officially chosen by the CMS collaboration. Three different calorimeter technologies were presented to the collaboration and the HGCAL project was chosen in 2015. This choice was followed the same year by the release of the Technical Proposal of the CMS Phase 2 upgrades [5]. The following milestone was the release of the HGCAL Technical Design Report in 2018 [6]. Since then the work has shifted from prototyping towards production. In that aspect the year 2022 is a turning point for many components of the detector, that are being evaluated for their pre-production and production. More specifically on the backend electronics and on the TPG system, algorithms, firmware and hardware are currently being reviewed for the hardware pre-series.

My main contributions, and those I have been coordinating, are related to the developments of TPG algorithms, their simulation and their firmware implementation. Since 2016 I have been coordinating the HGCAL TPG subgroup responsible of the simulation and algorithm developments. This document is consequently a summary of the past and present algorithms developed for the HGCAL TPG system. Some limited firmware and hardware aspects are also presented in order to illustrate how algorithms, firmware and hardware are intertwined. It is voluntarily short and it is by no mean a complete overview of the system. It covers mainly the aspects on which I have been directly or indirectly involved.

Finally I would like to emphasize that the current design of the HGCAL TPG system is the result of the work of many people, in particular PhD students, postdoctoral researchers and engineers. Therefore this document includes not only my work but also, and more importantly, the work from other people, without whom the HGCAL TPG couldn't be built.

Introduction

Algorithms used in L1 trigger systems are based on solid expertise and knowledge accumulated over the years, as well as on new ideas developed to cope with increases of data volume and of data complexity. Not only physics, but also electronics and detector expertises are required in their development, involving physicists and engineers. In that sense the development of L1 trigger algorithms is different from the development of offline algorithms, as it relies on multiple technical domains.

It is also unique due to the strong requirement of reliability. It comes from the fact that algorithms on the trigger path are often difficult or even impossible to modify once they have been implemented. This is indeed especially true for algorithms running on frontend ASICs. Since every processing step towards a trigger decision is critical for the quality of the recorded data, they should all be robust to expected and unexpected situations. In addition to reliability and robustness trigger algorithms also need to be as efficient as possible within tight implementation limits of latency, logic resources and power consumption. It means there is an equilibrium to be found between efficiency, robustness and simplicity.

Reaching such equilibrium for the HGCAL TPG is a particularly challenging task. Both the structure of the data to be processed by the system (highly-granular 3D data) and the environment of the data (unprecedented levels of pileup) are unmatched in complexity. Several trigger primitives reconstruction strategies have been imagined and developed during the past years. This document is mainly covering the current status of the algorithms and of the system, but also comparing them with earlier versions.

- The first chapter introduces the general context, with an overview of the HGCAL detector.
- In the second chapter, the reconstruction of trigger primitives in the HGCAL is discussed with a focus on algorithms, independently of the hardware and firmware constraints.
- The third chapter focuses on the frontend part of the trigger path, mainly addressing the issue of the limited data bandwidth and the reduction of data volume it implies.
- The last chapter covers the hardware and firmware implications of the reconstruction algorithms in the backend electronics.

1 The High-Granularity Calorimeter

1 Overview of the HGCAL upgrade

The LHC will undergo a major upgrade, currently planned to take place between the years 2026 and 2029, and at the end of which it will enter its high-luminosity phase (HL-LHC), or Phase 2. The future HL-LHC is designed to reach a nominal instantaneous luminosity of $5 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$, and up to $7.5 \times 10^{34}\,\mathrm{cm}^{-2}\mathrm{s}^{-1}$ in its ultimate configuration. An integrated luminosity of $3000\,\mathrm{fb}^{-1}$ should nominally be collected during the Phase 2 and up to $4000\,\mathrm{fb}^{-1}$ in the ultimate scenario. This increased luminosity will enable access to rare processes and open up a large physics program. It includes searches for new physics beyond the Standard Model (BSM) as well as precision measurements of the SM, in particular in the electroweak and Higgs sectors. Nevertheless this high luminosity comes with a significantly increased level of pileup (PU): 140 simultaneous interactions per crossing on average in the nominal scenario, and up to 200 in the ultimate scenario. This level of occupancy in the detectors poses significant challenges to the reconstruction of collision events, as well as to the transfer and processing of their data.

The current CMS sub-detectors were not designed to sustain such high luminosities and significant upgrades of some of their components are required to be able to efficiently collect data during the HL-LHC. In particular the CMS ECAL crystals and the HCAL plastic scintillators in the endcap region were designed for an integrated luminosity up to $500\,\mathrm{fb}^{-1}$. Radiation damages beyond this integrated luminosity will be too important and create unrecoverable losses in the response of these detectors. The entire endcap calorimeters of CMS will therefore be replaced. The HGCAL, a highly granular sampling calorimeter with high transverse and longitudinal segmentations, has been chosen by the CMS collaboration in 2015 to replace the current endcap calorimeters [5]. The main drivers of this choice were the radiation hardness of its sensitive materials as well as its intrinsic capability to track particle showers. These tracking capabilities provide enhanced shower identification and rejection of PU. The rejection of PU is also increased by its precise timing measurement capability at the level of a few tens of picoseconds.

The HGCAL covers the pseudorapidity range between 1.52 and 3. It uses silicon as active material in the region receiving the largest lifetime dose of ionizing radiation and uses

plastic scintillator deeper and further away from the beam axis, as indicated in Fig. 1.1. While a total of 50 sensitive layers in each endcap was planned in the previous design, this number has been reduced to 47 at the time of this writing. The electromagnetic section (CE-E) is composed of 26 layers and the remaining 21 layers compose the hadronic section (CE-H).

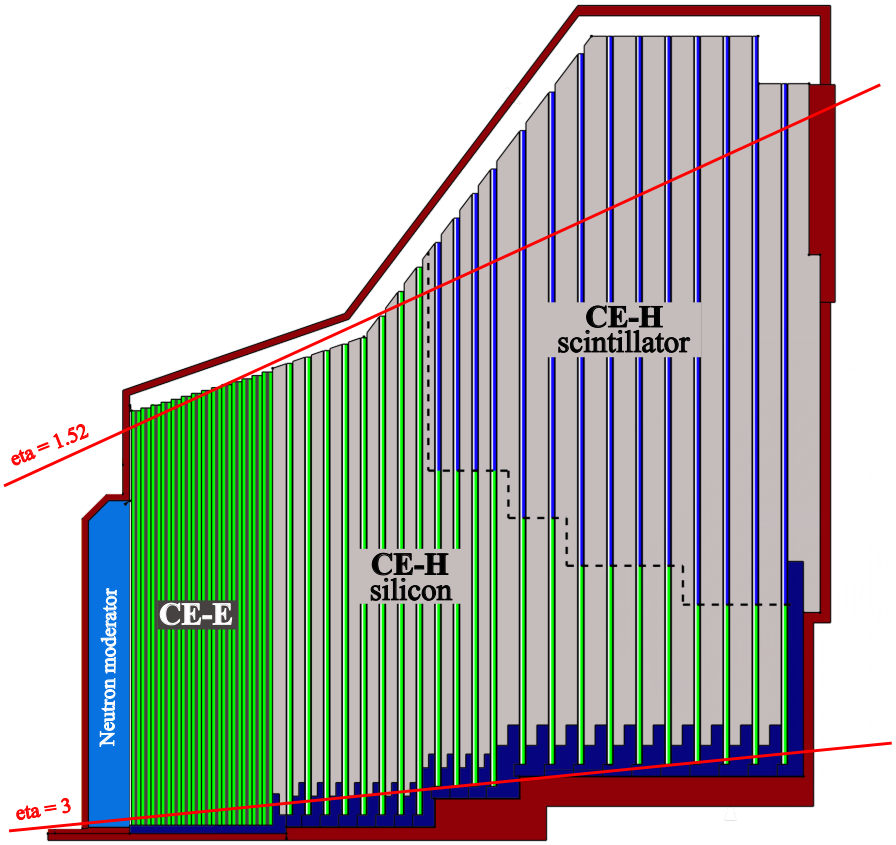

Figure 1.1: *Longitudinal cross section of the upper half of one endcap calorimeter [7]. It consists of an electromagnetic (CE-E) compartment followed by a hadronic (CE-H) compartment. The CE-H is instrumented partly with silicon sensors and partly with scintillator tiles, while the CE-E is instrumented entirely with silicon sensors. Since the publication of this drawing, three layers have been removed.*

The absorber in the CE-E is made of copper, copper-tungsten and lead plates, with a total depth of about $27.7\,X_0$ and $1.5\,\lambda$. Silicon sensors are produced from 8 inch wafers and have an hexagonal shape to maximize the efficiency of the sensor production. They contain hexagonal cells, and depending on the location of the module in the detector two different cell sizes are used. This change of cell size is done to optimize the S/N ratio throughout the detector. Low-density modules are located away from the beam pipe and have cells of $1\,\mathrm{cm}^2$ area. High-density modules are located closer to the beam pipe and have cells of $0.5\,\mathrm{cm}^2$ area. In addition, three different silicon sensor thicknesses are used, depending on the detector region, in order to adjust their radiation hardness to the expected hadron fluence and ensure that the S/N ratio remains sufficiently high during the entire lifetime of the detector.

The absorber in the CE-H is made of stainless steel plates, with a total depth of $8.5\,\lambda$. Silicon sensors or scintillator tiles with on-tile SiPM readout are mounted on copper cooling

plates between stainless steel plates. The first CE-H layers are entirely based on silicon modules, while deeper layers are mixing silicon modules in the higher-radiation region (close to the beam pipe) and scintillator tiles in the low-radiation region. The layout of one CE-E silicon layer and of one mixed layer in the CE-H are shown in Fig. 1.2.

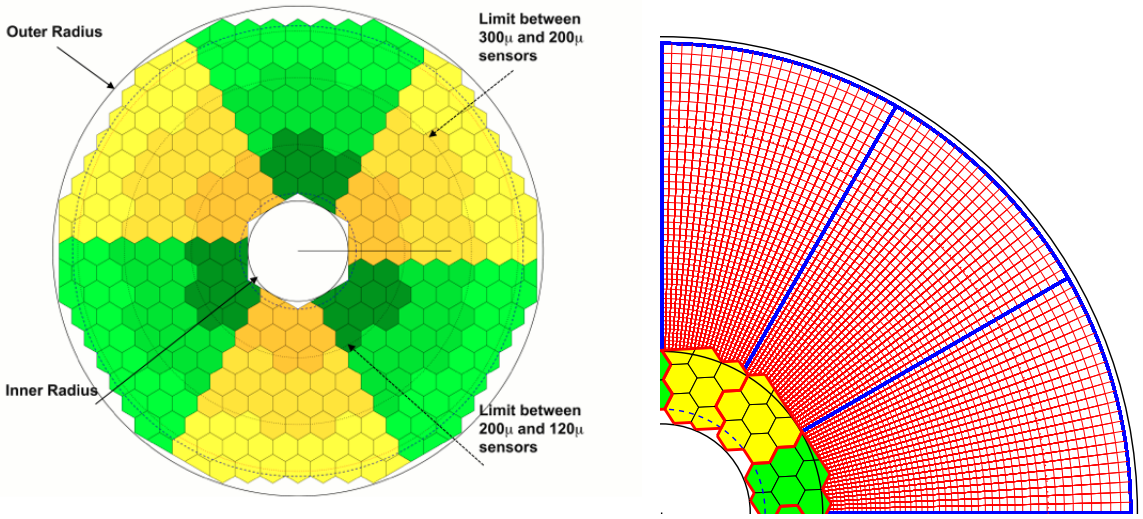

Figure 1.2: *Layout of one CE-E silicon layer (left) and of a* 90° *portion of one layer combining silicon modules and scintillator tiles in the CE-H (right) [6]. Silicon modules are depicted in green and yellow, the two alternating colors showing the division into* 60° *cassettes. The three different shades correspond to the different sensor thicknesses,* 120 μm*,* 200 μm *and* 300 μm*. Scintillator tiles are depicted in red.*

2 Frontend electronics

Two frontend (FE) ASICs are developed to collect and readout the energy deposited by particle showers in the active materials. The first ASIC, called HGCROC [8], is the entry point of the data acquisition chain. It contains both analogue and digital components and is mainly responsible of collecting, amplifying and filtering the charge current collected in the silicon or generated in the SiPMs. It is also measuring the time of arrival (ToA) of the pulses. The trigger data processing starts already there with a digital summation of the charges deposited in groups of adjacent cells, called trigger cells (TC), and the compression of the summed charges on a reduced number of bits. In parallel, the full-granularity data is kept in circular buffers to accommodate for the latency of the L1 trigger and is sent out when a L1 Accept (L1A) command is received.

One specific feature of this readout chip is the usage of Time-over-Threshold (ToT) to measure deposited charges above a configurable threshold of typically 100 fC and up to 10 pC. For charges below this threshold a more standard mode, the ADC mode, is used. In the ADC mode the deposited charge is measured from the amplitude of the pulse, digitized with a 10-bit ADC. While when the preamplifier saturates, in the ToT mode, the time of saturation is used as a proxy to measure the deposited charge, and digitized with a 12-bit

TDC. This saturation can last up to 200 ns, which corresponds to charges of 10 pC, during which the channel is blind to subsequent energy deposits.

The digitized data is processed on two parallel paths, the DAQ path with an output rate of 750 kHz, and the trigger path with an output rate of 40 MHz. Given the full event rate in the trigger path and the limited number of output links associated to it (four 1.28 Gbps electrical links per HGCROC in low-density modules and two in high-density modules), its input data needs to be reduced by up to a factor 35 (in high-density modules). This data reduction is achieved by combining several lossy compression methods. The first step of the trigger path is to rescale the ToT values such that their least significant bit value (LSB) is the same as the LSB used by the ADC. A configurable multiplicative factor is applied to the ToT values, which are expanded to 17 bits. This is required in order to be able to sum cell values within a given trigger cell that might be reconstructed with different modes. Cell values are then summed together and the resulting value is compressed on 7 bits with a non-linear compression scheme. This non-linear compression is able to cover the large dynamic range of the energy deposits and to keep enough precision in the low-end of the spectrum. Fig. 1.3 summarizes the different blocks composing the HGCROC. More details regarding the TC summation and compression are given in Chapter 3. The compressed

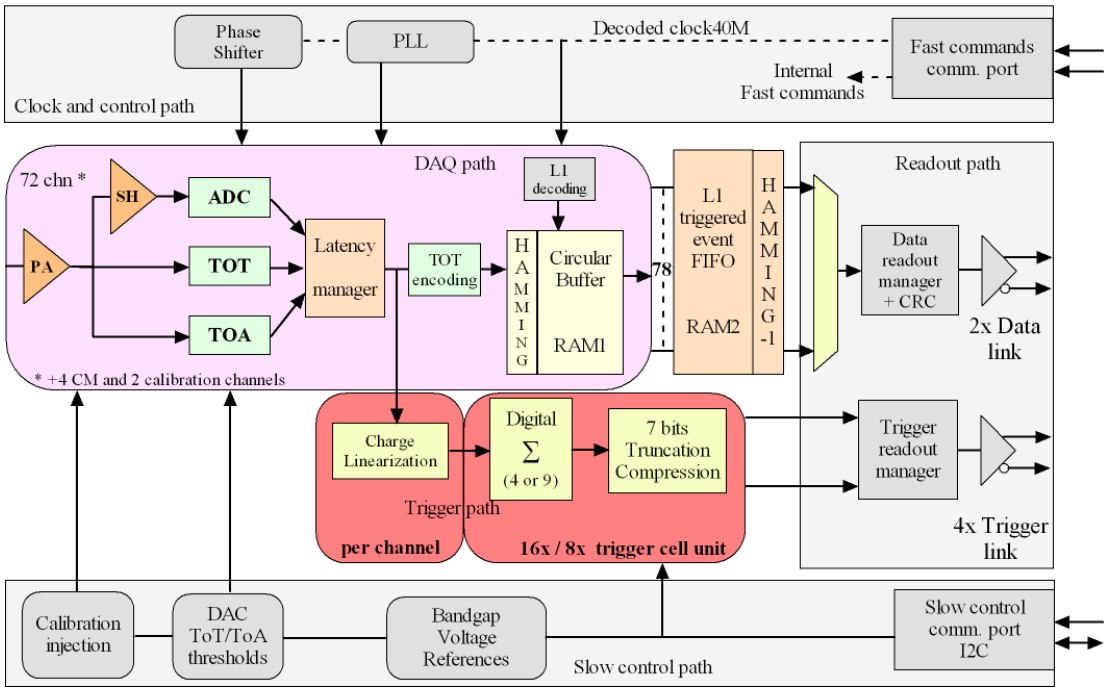

Figure 1.3: *Block diagram of the HGCROC [8]. It is composed of two data paths: the DAQ path (on top), connected to the ECON-D and the trigger path (at the bottom), connected to the ECON-T.*

TC charges are sent on the 1.28 Gbps electrical links to a second ASIC, the ECON-T. In parallel, on the DAQ path, events for which a L1A command has been received are sent to a separate concentrator chip, called ECON-D, which performs channel alignement and zero suppression.

The ECON-T concentrates trigger data from several HGCROC chips. It covers 48 TCs corresponding to full silicon modules. The ECON-T first calibrates the input charges received from the HGCROC and transforms them into transverse energy values. The second step in the ECON-T consists in reducing further the data stream to be sent to

the backend (BE) electronics. This data reduction can be done in several ways and the baseline strategy, foreseen to be used at the time of writing this document, is to apply a simple threshold on the TC transverse energy, selecting only TCs above this threshold. Threshold values around 1.5 $\mathrm{mip_T}$ are typically used, where $\mathrm{mip_T} = \mathrm{mip}/\cos(\theta)$ and one mip is the energy deposited by a minimum ionizing particle. More details on the different data reduction strategies considered for the ECON-T are provided in Chapter 3.

In parallel, the transverse energies of the 48 TCs are summed into a single value without any threshold applied. The summed values are called module sums. Before the calibration is applied and the TC energies are summed, the compressed 7-bit TC inputs are decompressed on a linear scale. The calibrated TC output are then recompressed on 7 bits with a similar compression scheme and the module sums are compressed on 8 bits. Finally, the reduced TC data and the module sums are transmitted to lpGBT ASICs [9] on 1.28 Gbps electrical links, and then serialized to 10.24 Gbps before being transmitted on optical links to the BE electronics through the VTRX+ [10].

The overall schematic of the frontend system in the low-density region is shown in Fig. 1.4.

3 Backend electronics

The backend electronics responsible of the reconstruction of the HGCAL trigger primitives is composed of two processing stages. The first stage (Stage 1) receives the data coming from the ECON-T chips, while the second stage (Stage 2) is connected to the central L1 trigger and delivers the HGCAL trigger primitives. The trigger primitives are of two types:

- clusters of TCs reconstructed in 3 dimensions,
- and projective towers with a transverse granularity similar to the current trigger towers of the CMS calorimeter trigger (0.087 in η and ϕ).

Cluster primitives contain not only the position and energy of the clusters, but also a set of cluster shape variables used downstream in the L1 trigger, for instance to identify the type of shower [7, 12]. These shape variables are typically encoding the size of the clusters in the transverse and longitudinal directions, as well as how the energy is distributed inside the clusters. While clusters are built from TCs on which a threshold has been applied in the ECON-T, towers are built from the module sums for which no suppression of energy has been applied. Therefore, towers contain the total energy deposited in the calorimeter, contrary to clusters for which energy is lost through TC suppression and through clustering itself. Towers are therefore expected to be used for calorimeter-only jet triggers and energy sum triggers, for which granularity is less important, while clusters will be used for particle flow reconstruction and the matching with the primitives from the inner tracker.

The Stage 1, in addition to receiving the ECON-T data and unpacking them, also reduces further the stream of data and concentrates 120° sectors into single Stage 2 FPGA. The system is therefore split into six subsystems covering a 120° region each. In order to correctly reconstruct clusters at the boundaries between two 120° regions, some Stage 1 data are duplicated and sent to two Stage 2 FPGAs. The architecture of three of these interlinked 120° subsystems, covering one endcap in total, is shown in Fig. 1.5. More details on the current backend architecture are given in Chapter 4, as well as on other architectures envisioned in the past.

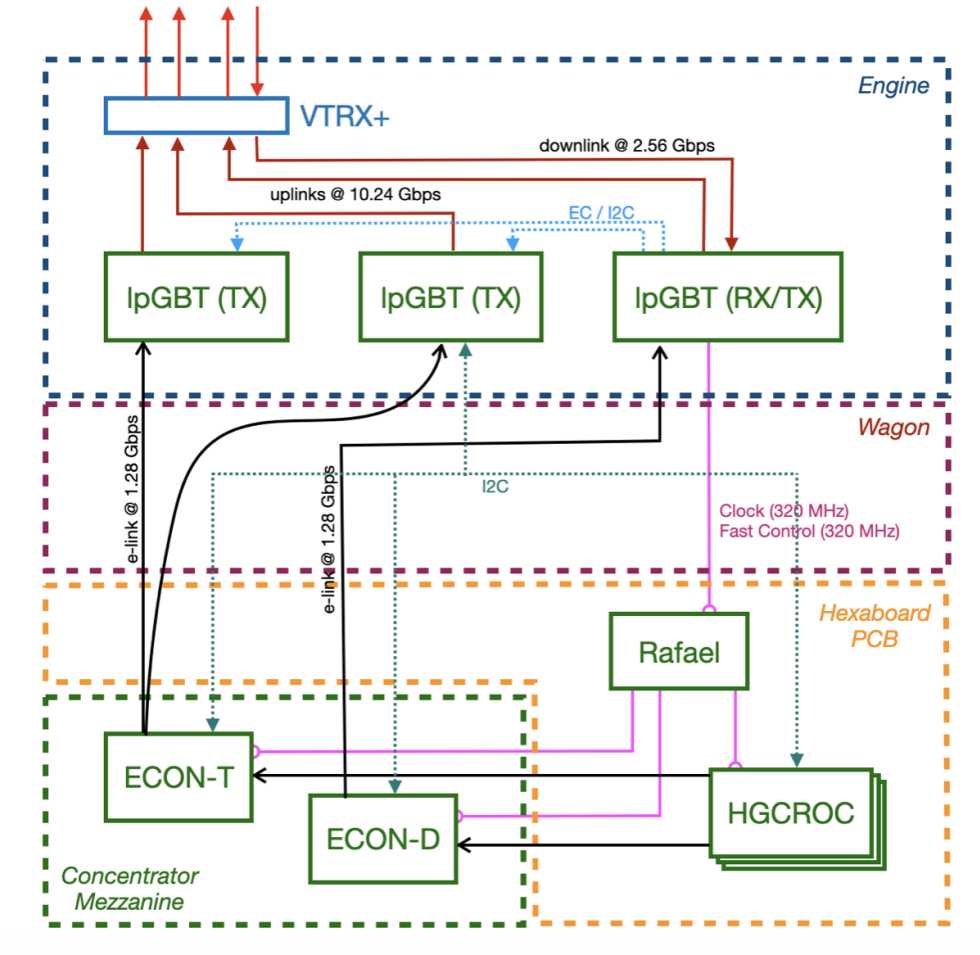

Figure 1.4: *Schematic of the frontend system in the low-density region [11]. Data from the HGCROC are passed through the ECON chips and then to the engine board through the wagon board. In the opposite direction, the clock and fast commands are distributed through the Rafael ASIC. The engine board contains the lpGBT and the VTRX+ and is connected to the BE electronics. The wagon board is a passive board serving to connect the detector modules and the engines.*

Copies of the HGCAL trigger primitives data are sent, with a total latency budget of 5 μs, to two subsystems within the central L1 trigger: the correlator trigger and the global calorimeter trigger (GCT). For each copy, a bandwidth of 11 Tbps is allocated. The correlator trigger receives primitives from all sub-detectors (all the calorimeters, the muon system and the inner tracker), while the GCT processes only information from the calorimeters. The correlator trigger will run particle flow reconstruction algorithms making use of the combined information of the calorimeters and of the inner tracker to reconstruct and identify electrons, photons, hadronic taus and jets. The global calorimeter trigger, on the contrary, will reconstruct calorimeter-only objects: e/γ candidates (without distinction between electrons and photons), hadronic taus, jets and energy sums. A diagram summarizing the different components of the future CMS L1 trigger system and their interconnections is shown in Fig. 1.6.

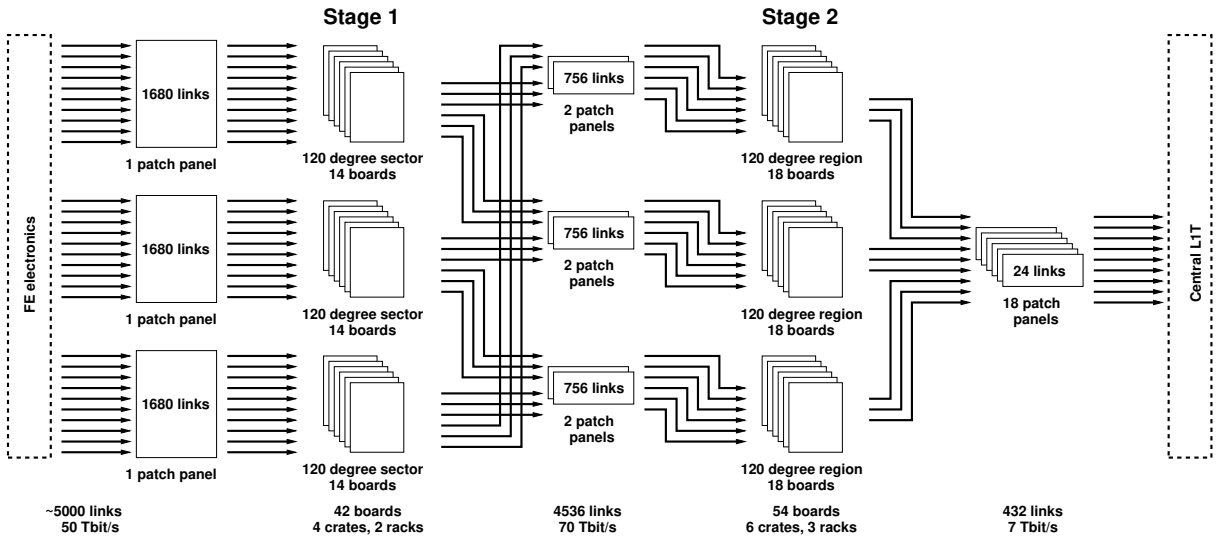

Figure 1.5: *HGCAL TPG subsystem covering one endcap [12]. It is composed of three sets of FPGAs, each set being responsible of the processing of one* 120° *sector. Some Stage 1 data at the boundaries between two sectors are duplicated in order to reconstruct clusters in these boundary regions.*

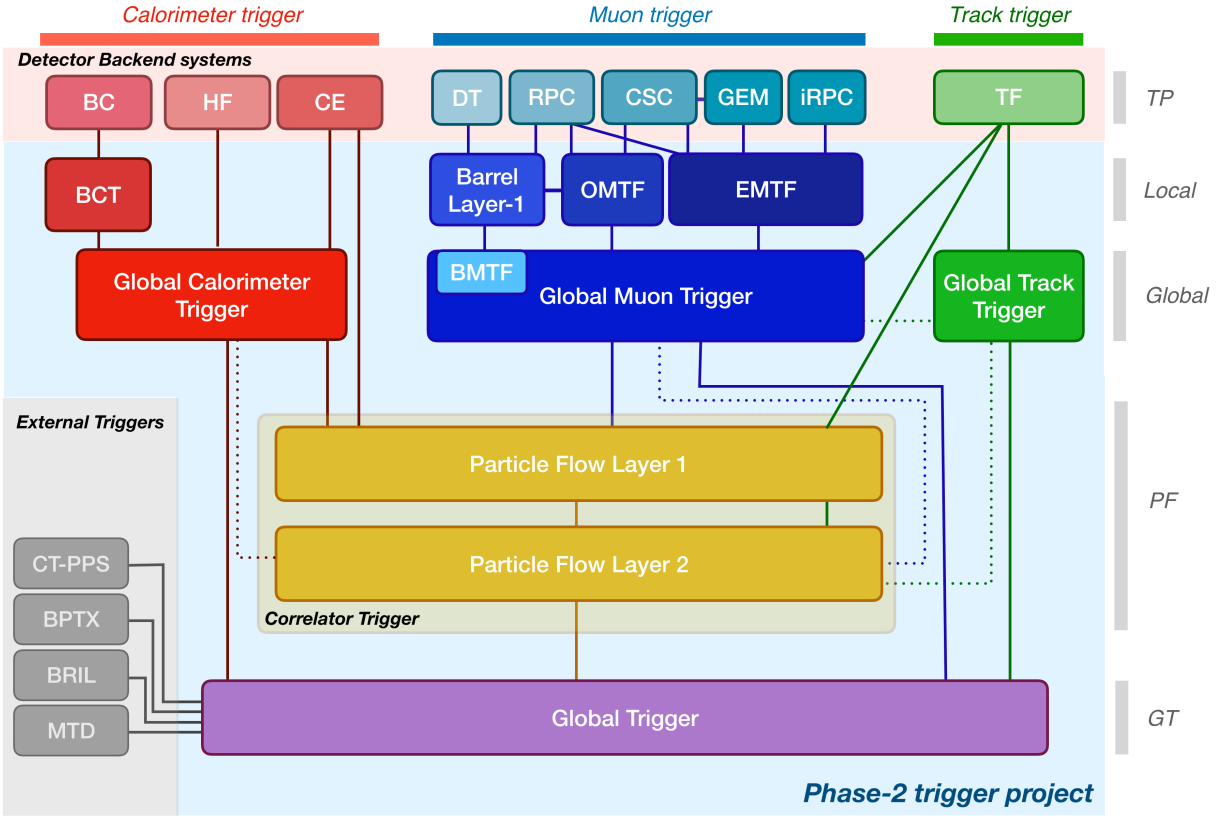

Figure 1.6: *Diagram of the CMS Phase 2 L1 trigger system [7]. It receives data from the different sub-detector backend systems. It is composed of Global Calorimeter, Muon and Track Triggers, and of a Correlator Trigger. These subsystems provide information to the Global Trigger, which issues the final L1 trigger decision.*

4 Simulation

For the development and tuning of the HGCAL TPG algorithms and the study of their performance, a full simulation of the HGCAL and of the rest of the Phase 2 CMS detector has been used, based on the CMS software framework (CMSSW). The geometry of the HGCAL has evolved in the past years and its description in CMSSW has been refined and is still being refined as these lines are written. Most of the results presented in this document are based on recent geometries described in the HGCAL TDR [6] or in the L1T TDR [7]. Some of the studies have been repeated with different versions of the HGCAL geometry and didn't result to changes in the conclusions nor had impact on the choices previously made.

The main samples used to develop and study the HGCAL TPG algorithms are *particle gun* samples. In these samples, particles of a given type are generated with a given 4-momentum distribution. Electromagnetic shower studies are based on electron and photon gun samples, as well as charged pion samples for background definition in identification studies. These particles, in our case, are typically generated with a uniform distribution in η and p_{T}, with 5 GeV $< p_{\mathrm{T}} \leq$ 100 GeV. In addition, particle gun samples of quarks and hadronically decaying tau leptons are also used for jets and tau lepton studies, as well as VBF $H \rightarrow$ invisible samples at $\sqrt{s} = 14$ TeV. Finally, minimum bias samples are used to define background and compute trigger rates.

Events are simulated with 200 PU, corresponding to the ultimate HL-LHC scenario. The detector response is simulated with GEANT4 [13] and an accurate simulation of the electronics is also used. At the time of this writing, a bit-level emulation of the full trigger path is being written. It is not yet available in production and only a bit-level emulation of the trigger steps implemented in the HGCROC were available, the rest of the chain being based on floating point arithmetics.

2

Reconstruction of trigger primitives in the HGCAL

The role of the HGCAL TPG system is to build objects that are used as primitives by the central L1 trigger. These primitives contain information on local energy deposits. They include typically their transverse energy and their position in $\eta - \phi$ coordinates, as well as a certain number of bits used to identify the type of energy deposits. The central L1 trigger of CMS reconstructs higher-level physics objects, such as electrons, photons and jets, from the HGCAL trigger primitives and those from the other subdetectors. It then makes the decision to trigger on an event based on the high-level objects reconstructed in this event.

Two types of primitives are reconstructed in the HGCAL:

- Three-dimensional (3D) clusters, which are dynamically built and ideally aggregate the energy of single particle showers.
- Projective towers, which are static (fixed-size) objects built from a predefined grid in the $\eta - \phi$ plane.

The main primitives on which most of the developments have been focused in the past years are the 3D clusters, and they will be the focal point here. Projective towers are simpler primitives and carry less granular information. They will nevertheless be useful in commissioning phases and to compute quantities that don't require a fine granularity such as event energy sums based on calorimeters information.

The goal of the reconstruction at the L1 trigger is not very different from the offline reconstruction: reconstruct the particle content of collision events. There are nevertheless a large number of implementation limitations coming from the hardware and firmware. The design of reconstruction algorithms at the L1 trigger is tightly tied to the design of the system architecture, to the design of the firmware within processing nodes, and to the data flow between processing nodes. In this chapter, nevertheless, a purely algorithmic view of the HGCAL trigger primitives reconstruction is covered. Hardware and firmware aspects are covered in the next chapters.

1 Noise suppression and resilience to pileup

Given the high-PU environment in which clusters will be built, a key aspect of any clustering algorithm in the HGCAL is its resilience to PU and to noise in general. The robustness to PU can be controlled in two places in the reconstruction: before the trigger cells are actually clustered and within the clustering algorithm itself. First of all, the majority of PU energy deposits are low energy deposits, and applying energy thresholds is an easy and efficient way to reduce the contamination from PU. In addition to that, particle showers have in general a high-density core and their energy density decreases further away from this core. Therefore, constraining the transverse size of the reconstructed clusters, and focusing on the reconstruction of the shower core, reduces the sensitivity to PU contamination. It is also particularly efficient to reduce the contamination from overlapping high-energy showers.

Obviously these two main parameters, the trigger cell energy thresholds and the cluster radius, have two opposite effects on the cluster energy reconstruction. Increasing the thresholds and decreasing the radius reduce the contamination from noise and PU while at the same time also reduce the fraction of signal energy aggregated in the cluster. These effects are competing in terms of energy resolution: the resolution is improved when limiting the integration of noise but it is degraded when limiting the integration of signal. For these two parameters there are therefore optimal values, which depend on the level of PU and noise, and of the signal energy.

It has to be noted that the optimal values of these two parameters depend also on the Moliere radius of the detector (the radius in which 90% of the electromagnetic showers energy is contained) and its transverse granularity. The Moliere radius is determined by the stack of materials used in the detector, and in particular by the choice of absorber material. It is equal to approximately 3 cm in the HGCAL [6], and the cell areas are either $1\,\mathrm{cm}^2$ or $0.5\,\mathrm{cm}^2$ depending on the region of the detector, such that the core of the showers are sufficiently well sampled. Nevertheless, the granularity at the L1 trigger needs to be reduced significantly, as we will see in Chapter 3, down to a trigger cell granularity of $4\,\mathrm{cm}^2$. As a consequence the energy of electromagnetic showers is mostly deposited in a single or a few trigger cells in a given layer. The ability to limit overlaps between showers, using energy thresholds and constrained cluster sizes, is therefore reduced compared to what can be done in the offline reconstruction.

Thresholds and radii can be defined as constant across the detector and optimized inclusively. In that case they are simple to optimize and to implement, but they don't capture the dependency of PU with the pseudorapidity and with the depth in the detector. This is at the moment what is applied to reconstruct electromagnetic showers. Clustering radius and threshold have been optimized looking at the energy resolution and identification performance. The RMS and effective RMS of the $p_{\mathrm{T}}^{\mathrm{L1}}/p_{\mathrm{T}}^{\mathrm{gen}}$ distribution, normalized by its mean value, are used to measure the energy resolution. The effective RMS is the smallest interval containing 68% of the distribution. It measures the core width of the distribution and is less impacted by tails and outliers than the RMS. These values are shown after energy calibration and energy correction are applied, as detailed in Section 2 of this chapter. The electromagnetic cluster identification and background rejection is provided by boosted decision trees (BDT) that will be discussed later in this chapter. Their performance is quantified by choosing certain signal efficiency working points (typically above 95%) and looking at the background efficiency/rejection at those working points. Scans of the resolution and background rejection as a function of the cluster radius and

of the trigger cell threshold are shown in Figures 2.1 and 2.2. There is an increase of the energy resolution and of the background efficiency (i.e. a decrease of the background rejection) when the values of these threshold and radius become too small or two large. These effects come from the two competing effects discussed above and range from about 10% to a factor two or more in the vicinity of the optimal values.

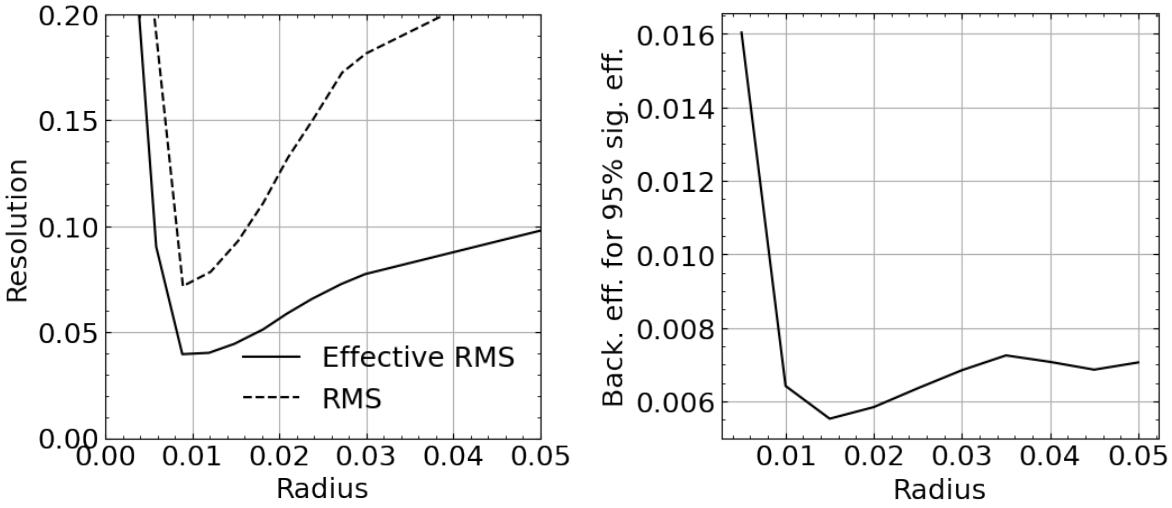

Figure 2.1: *Electron energy resolution (left) and background efficiency (right) as a function of the clustering radius in the $(x/z, y/z)$ plane. The energy resolution is measured with the RMS and the effective RMS of the $p_{\mathrm{T}}^{\mathrm{L1}}/p_{\mathrm{T}}^{\mathrm{gen}}$ distribution, divided by the mean of the distribution. The background efficiency is the probability of misidentifying charged pion showers as electromagnetic showers for a 95% selection efficiency of electromagnetic showers. Electrons and pions are generated with* $10\ \mathrm{GeV} < p_{\mathrm{T}} \leq 100\ \mathrm{GeV}$ *and* $1.6 < |\eta| < 2.9$*, and events are simulated with 200 PU.*

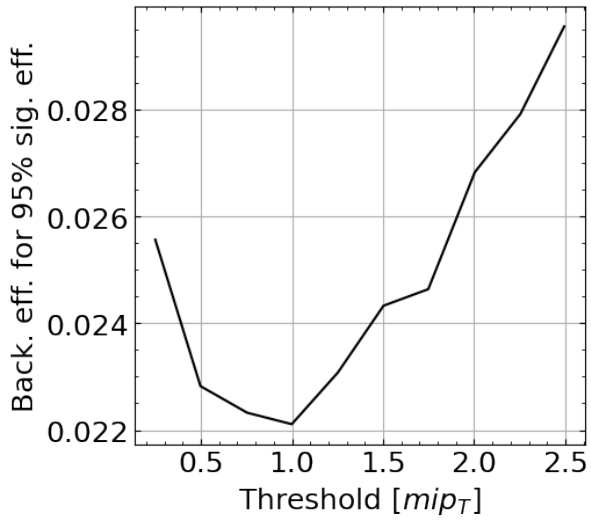

Figure 2.2: *Charged pion (background) cluster selection efficiencies for a 95% electromagnetic (signal) cluster selection efficiency, as a function of the clustering threshold in mip_T units. Particles are generated with* $10\ \mathrm{GeV} < p_{\mathrm{T}} \leq 100\ \mathrm{GeV}$ *and* $1.6 < |\eta| < 2.9$*, and events are simulated with 200 PU.*

Capturing the dependency of the level of PU as a function of the pseudorapidity and of the detector layer depth into clustering thresholds or radii requires more complex optimization procedures because of the large number of parameters to optimize in that case.

Hadron clusters are at the moment built using a radius varying with the depth in the detector. But it has been loosely optimized by hand and a complete optimization has not been tackled yet. The most advanced optimization procedure was investigated at the beginning of the HGCAL TPG developments [5]. The clustering threshold was defined as a function of (1) the pseudorapidity, (2) the detector layer and (3) the global level of PU in the event. This procedure is making use of the energy distributions of PU as a function of these three quantities, and more precisely making use of the percentile values of these PU energy distributions. Thresholds values are defined such that a fixed fraction of the lowest PU energy deposits are removed at any point in the detector. It effectively reduces the multidimensional optimization space into a single value: the fixed fraction of PU hits to be removed. And the dependency of the threshold is fixed by the PU energy distribution in the detector. Thresholds chosen to reject 95% of the lowest PU energy hits are illustrated in Fig. 2.3. A proxy variable was used to quantify the global level of PU in the event: the number of hits in detector regions covered by processing boards. Although a full-fledged optimization procedure was not implemented at that time, and the chosen percentile value was particularly high, it demonstrated the feasibility of implementing complex threshold dependencies for clustering. More developments towards more granular definitions of clustering thresholds and clustering radii are expected in the future.

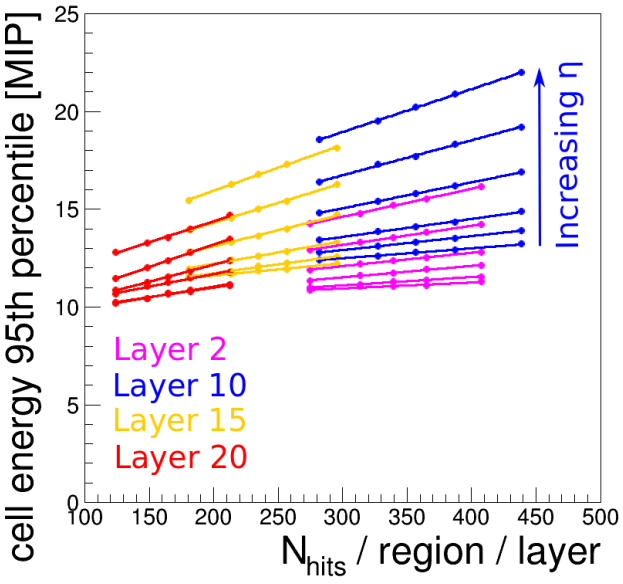

Figure 2.3: *Clustering thresholds defined as the 95th percentile of the trigger cell PU energy, in mip units [14]. The thresholds are shown as a function of the number of hits per layer in predefined detector regions, for different layers and different pseudorapidy rings.*

2 Energy calibration and corrections

An accurate measurement of particle shower energies also implies the application of several calibration and correction factors. First, the energy deposited in the active material needs to be estimated from the collected charges. These calibration coefficients depend on the type of material, its thickness and the charge collection efficiency (which decreases as the

detector becomes more irradiated). This energy is usually expressed in mip units. Ionizing deposits from charged particles in collision data will be used to extract intercalibration factors and ensure the uniformity of the response across the detector [6].

Given the sampling nature of the HGCAL, the total shower energy needs to be estimated from the energy deposited in each active layer. It is done by multiplying the measured energy by layer-dependent weights related to the calorimeter sampling fraction. These weights take into account the amount of material in the absorber layers and will be derived from construction details. The values of these weights used in the current simulation are shown in Fig. 2.4. Finally the absolute scale setting will be performed in situ using standard candles, p_{T}-balance techniques, etc.

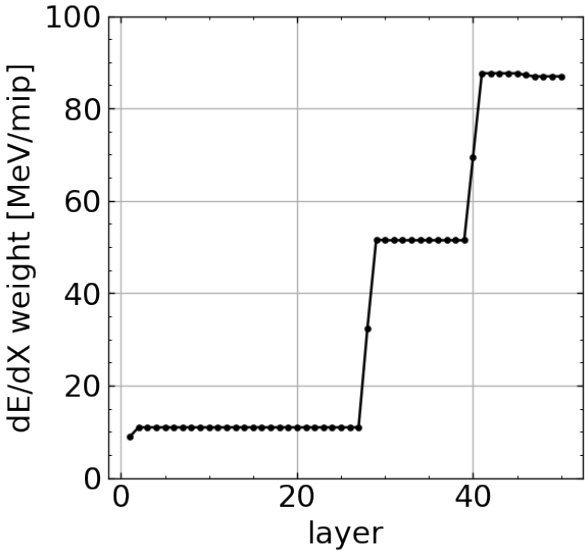

Figure 2.4: *Layer-dependent calibration coefficients based on the material thicknesses between active layers. The jumps in the coefficients reflect the change of absorber type and thicknesses in the CE-E and in the two sections of the CE-H.*

Additional intermediate steps are necessary due to the imperfect reconstruction of the energy deposited by particle showers. These steps are generally called *energy corrections* rather than *energy calibration*. The first energy correction component takes into account the fact that clusters do not integrate all the deposited energy from a given shower due to thresholds applied on the energy hits as well as limited cluster radii. The calibrated energy of the cluster needs therefore to be corrected to take into account these losses of energy. The second component takes into account additional energy from PU integrated into the cluster. Although PU contamination can be limited thanks to thresholds and limited cluster sizes, it cannot be completely eliminated. Energy corrections need therefore to be applied in order to subtract these remaining contaminations. These two energy correction components can potentially be grouped together but it has to be noted that they are of different nature:

- The leakage correction is multiplicative as the energy missing in the cluster is at first order proportional to the initial shower energy.

- The PU correction is additive since the PU contamination is at first order independent from the initial shower energy.

Therefore if these two contributions are grouped together, the overall correction needs to be energy-dependent since the relative importance of the two contributions depends on the

energy (the PU correction is more important at low energy and the leakage correction is more important at high energies).

Leakage energy corrections can be applied on the reconstructed cluster energy as global corrections or layer by layer since different correction factors can be expected for the different layers, in particular with clustering thresholds that depend on the layer. These corrections can also depend on several quantities such as the cluster pseudorapidity, its energy and its shape. The 3D shape of a cluster is an important information for its identification, as will be discussed later in this chapter, but it also provides valuable information for its energy correction. The shape of a cluster is usually encoded in several quantities such as the widths in the three dimensions, the position of the shower maximum, the fraction of energy deposited in the ECAL, the energy density, etc. The corrections finally depend on the type of shower, whether it is an electromagnetic or an hadronic shower, and two sets of corrections need to be derived.

PU corrections should at least depend on the pseudorapidity as there is a strong dependency on the level of PU energy with the distance from the beam axis. They can depend on the global level of PU in the event as well, similarly to what is shown for thresholds in Fig. 2.3. In principle, the level of PU energy also depends on the depth of the detector layer. But since it is an additive correction, only an overall correction summed over all the layers is needed (the layer-by-layer dependency is absorbed in the cluster sum).

At the moment one of the simplest possible correction schemes is used, applying layer-by-layer weights to correct for threshold and leakage effects, and subtracting PU contaminations as a function of the pseudorapidity with a linear dependency:

$$E^{\text{corr}} = \sum_{\text{layer } i} w_i E^i - (a|\eta| + b)$$

The impact of such corrections on the energy reconstruction of electrons is shown in Fig. 2.5, where the $p_{\text{T}}^{\text{L1}}/p_{\text{T}}^{\text{gen}}$ distribution is shown before and after applying the energy corrections. The effective RMS of this distribution divided by its mean is also shown as a function of $p_{\text{T}}^{\text{gen}}$ in Fig. 2.6, together with its mean value as a function of the pseudorapidity.

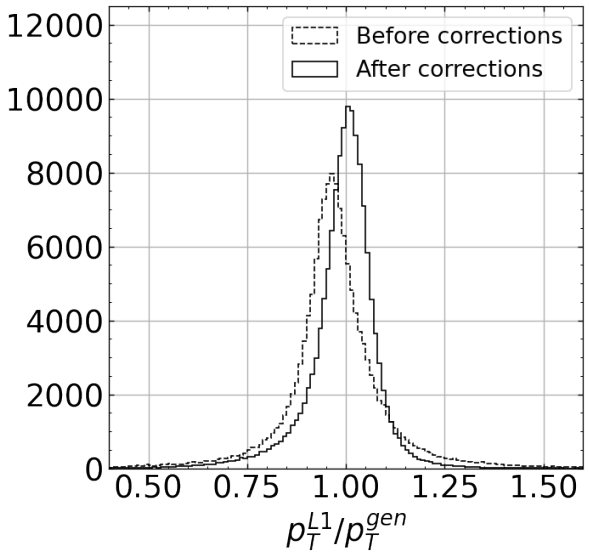

Figure 2.5: *Distribution of the* $p_{\text{T}}^{\text{L1}}/p_{\text{T}}^{\text{gen}}$ *ratio for electrons generated with a uniform* p_{T} *and* η *distribution between* 10 GeV $< p_{\text{T}} \leq$ 100 GeV *and* $1.6 < |\eta| < 2.9$*, before and after applying leakage and PU corrections. Events are simulated with 200 PU.*

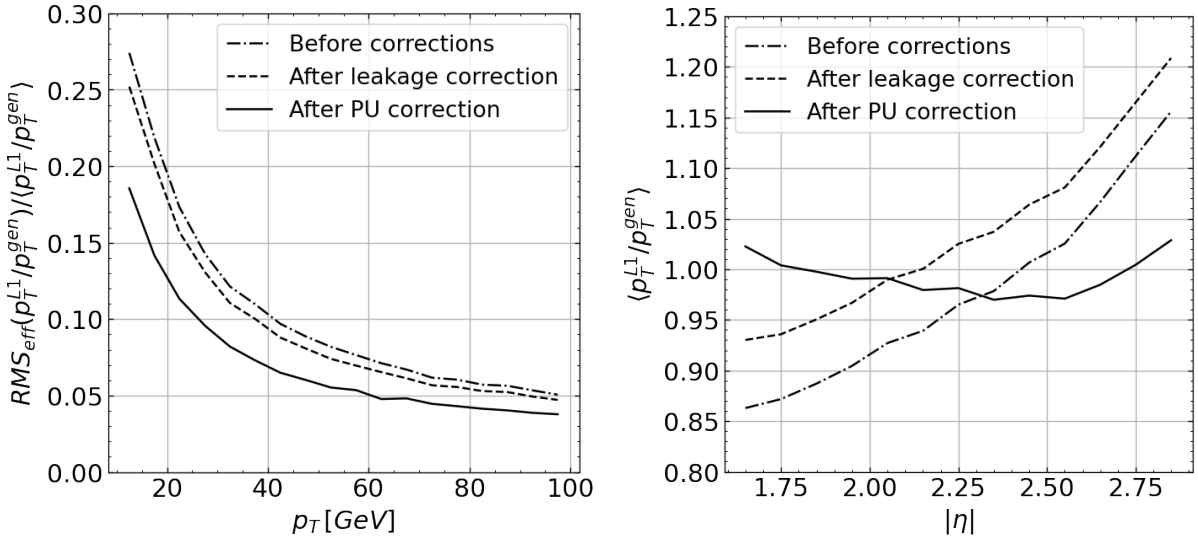

Figure 2.6: *Energy resolution as a function of* p_{T} *(left) and energy response as a function of* $|\eta|$ *(right). The curves obtained before applying energy corrections, after applying the leakage correction, and after applying both the leakage and PU corrections are compared. The resolution is defined as the effective RMS of the* $p_{\mathrm{T}}^{\mathrm{L1}}/p_{\mathrm{T}}^{\mathrm{gen}}$ *distribution, normalized by its mean value. The response is defined as the mean value of the* $p_{\mathrm{T}}^{\mathrm{L1}}/p_{\mathrm{T}}^{\mathrm{gen}}$ *distribution. Electrons are generated with* $10\ \mathrm{GeV} < p_{\mathrm{T}} \leq 100\ \mathrm{GeV}$ *and* $1.6 < |\eta| < 2.9$*, and events are simulated with 200 PU.*

More sophisticated methods, in particular using machine learning techniques, could be used to improve further the energy resolution of the reconstructed clusters. These aspects are covered a bit in Section 6 of this chapter.

3 Cluster identification

Once clusters have been reconstructed, it is critical to be able to identify whether the particle that initiated the shower is an electromagnetic or an hadronic shower. Electromagnetic triggers are indeed the largest rate consumers, with for instance around 30% of the total rate consumed by single and multi e/γ triggers during Run 2 [4]. And identifying accurately electron and photon showers is one of the most effective way to reduce the rate of these triggers. The importance of cluster shape variables has been mentioned for the energy corrections applied to the reconstructed clusters. They are even more important for their identification. Indeed the shape of the reconstructed clusters strongly depends on the differences in the development of electromagnetic and hadronic showers, both in the transverse plane and in depth. Although much more information on the shower can be obtained with the HGCAL compared to calorimeters with lower granularity and lower longitudinal segmentation, it is more complex to leverage as the number of variables needed to encode the 3D shape information of a cluster is larger and these variables are generally strongly correlated. It means that applying simple independent cuts on these variables, as it was typically done at the L1 trigger in the past, is inefficient in exploiting 3D granular information. In order to fully exploit the identification capabilities of the HGCAL, the multiple shape variables should be used together, which implies using multivariate identification

models like BDTs or neural networks.

Among the variables developed so far with the biggest impact in discriminating between electromagnetic and hadronic showers there are:

- The lateral and longitudinal widths of the shower,

$$\sigma^2_{\xi\xi} = \frac{1}{\sum E_i} \sum_{i \in \text{TCs}} E_i \cdot (\xi_i - \langle \xi \rangle)^2$$

  where $\langle \xi \rangle$ is the mean of the variable ξ weighted by energies and ξ is a position variable like $r = \sqrt{(x/z)^2 + (y/z)^2}$, η, ϕ or z.

- The fraction of energy in the ECAL section $E/(E+H)$, where E here denotes the energy deposited in the CE-E and H the energy deposited in the CE-H.

- The fraction of energy deposited in consecutive layers around the electromagnetic shower maximum $E_{i-j}/(E+H)$ and in the first part of the CE-H, $H_{i-j}/(E+H)$, where E_{i-j} is the energy deposited in the CE-E layers i to j and H_{i-j} is the energy deposited in the CE-H layers i to j. The list of such variables can be seen in Fig. 2.8.

- The *ECAL bitmap*, a set of bits encoding the presence or absence of energy deposits above a certain threshold in each of the CE-E layers.

- The *shower length*, which is the number of consecutive layers containing energy in the cluster.

- The barycenter position of the cluster in the longitudinal direction $\langle z \rangle$

These variables have already a good discrimination power by themselves, as is illustrated in Fig. 2.7 for σ_{rr} and $\langle z \rangle$. But significant improvements can be brought by their combination.

Of course the size of the identification models implemented at the L1 trigger needs to be kept as small as possible, which means they should be based on a relatively low number of these shape variables. In order to choose the best combination of variables, their importance in identification models can be gauged with methods borrowed from the field of *explainable artificial intelligence* (xAI), for instance with the SHAP method [15], as shown in Fig. 2.8. The choice of optimal shape variables is actually one of the most critical part of cluster identification. It is driven by multiple objectives, in particular the raw identification performance and the hardware complexity induced by the calculation of these variables. Given the large number of possible variable combinations, it is impossible to explore them manually. And importance methods like SHAP, although useful and very informative, are limited in choosing the best combination in such multi-objective framework. Therefore, a special class of optimization algorithms, called multi-objective evolutionary algorithms are also used to find the best trade-off between performance and complexity [16].

4 2D and 3D clustering

The different aspects of the reconstruction that have been described so far, the resilience to PU, the energy calibration and correction and the cluster identification, are general and can be applied to any reconstruction and clustering algorithm. But different clustering strategies can be applied to actually reconstruct particle showers in 3D calorimeters. Given

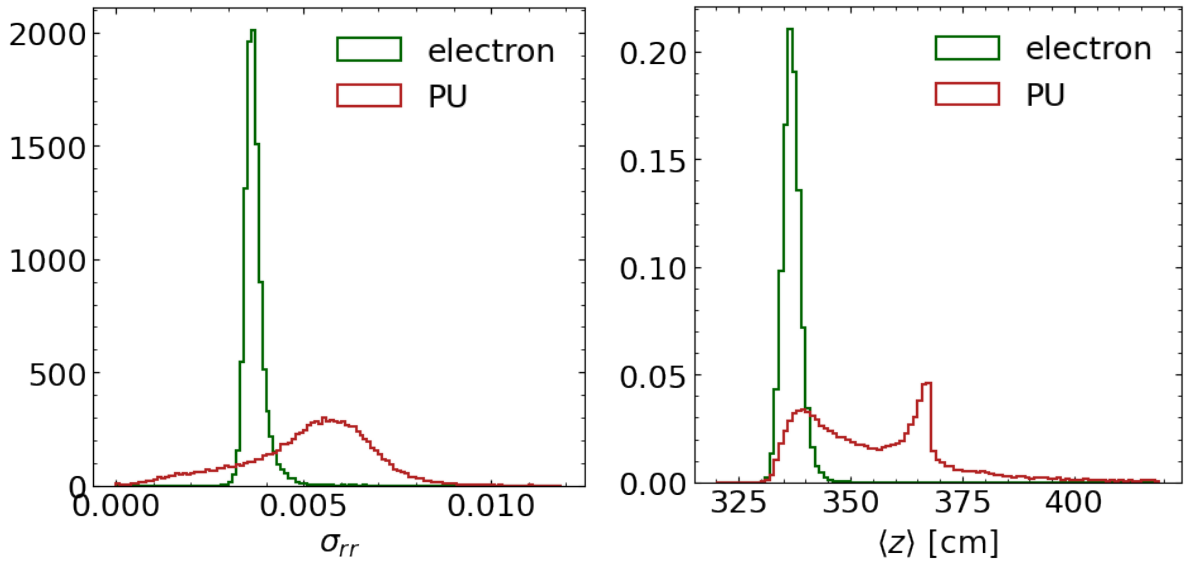

Figure 2.7: *Distributions of the lateral width (left) and of the barycenter in the longitudinal direction (right) of reconstructed clusters. The distributions of electromagnetic clusters and of PU clusters are compared. PU clusters are defined here as any reconstructed cluster with* $p_{\mathrm{T}}^{clu} > 20\,\mathrm{GeV}$ *in minimum bias events simulated with 200 PU. Electromagnetic clusters are defined as clusters matched to a generated electron with* $20\ \mathrm{GeV} < p_{\mathrm{T}}^{\mathrm{gen}} \leq 100\ \mathrm{GeV}$ *in electron gun events simulated with 200 PU.*

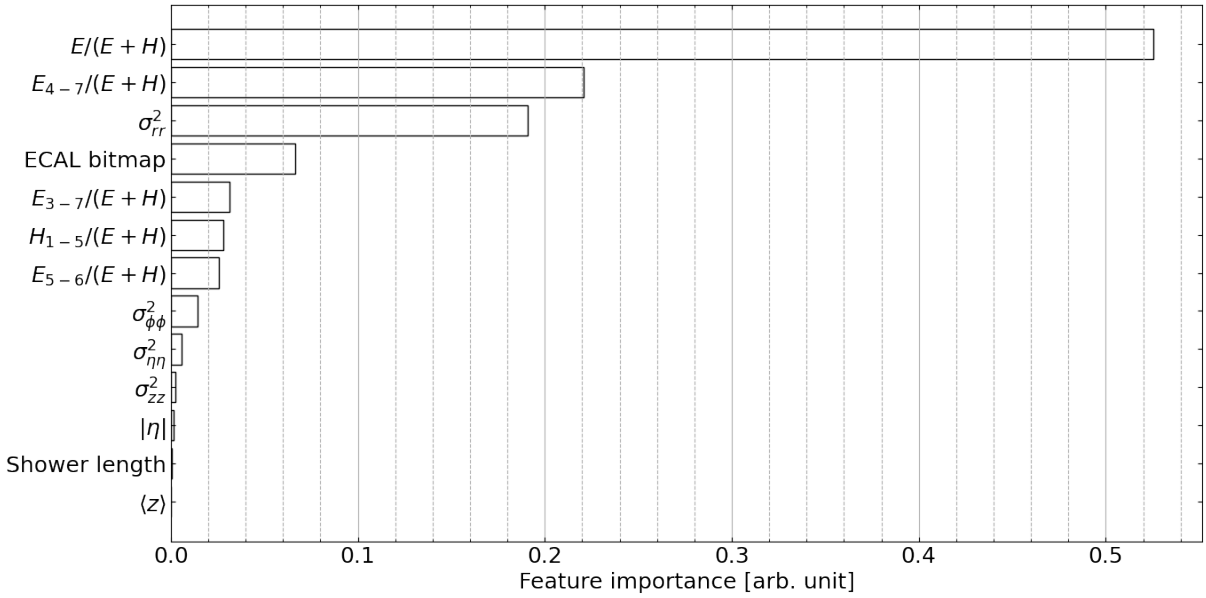

Figure 2.8: *Feature importance calculated with SHAP values [15]. E and H are the energies in the CE-E and CE-H sections, respectively.* E_{i-j} *and* H_{i-j} *are the energies between layers i and j in the CE-E and in the CE-H sections, respectively.* $\sigma^2_{\xi\xi}$ *are the variances, weighted by the energy, along the coordinate* ξ*. The ECAL bitmap encodes the presence or absence of energy deposits above a certain threshold in each of the CE-E layers. The shower length is the largest number of consecutive layers containing energy in the cluster.*

the structure of the calorimeter organized in layers, the first possibility is to proceed in two steps. In the first step energies are aggregated in two dimensions in each individual layers in order to form 2D clusters. These 2D clusters are then linked together from one layer to an other in order to form 3D clusters. This clustering strategy has been explored for the HGCAL TDR released in 2018 [6, 17].

Different 2D clustering algorithms were studied, mainly based on the distance between energy deposits in a given layer and on how these energy deposits are connected together (looking at the adjacency of the cells where the energy is deposited). In distance-based clustering algorithms, energy deposits above a seeding threshold are promoted as cluster seeds and energy deposits passing a (lower) clustering threshold are aggregated to these seeds within a certain distance. In adjacency-based clustering algorithms, energy passing a clustering threshold and deposited in adjacent cells are clustered together. These adjacency-based algorithms are also called topological clustering. These two generic clustering strategies are very common in calorimeter reconstruction, both offline and at the L1 trigger. But clustering energy hits based on their distance without requirement on their connectivity, or clustering connected hits without constraints on their distance, both suffer from important limitations. Indeed a distance-based clustering can cluster together disconnected nearby showers, while an adjacent-based clustering can be very sensitive to PU energy deposits and grow clusters in an uncontrolled manner. Although these issues can be limited with properly tuned parameters, it is very sensitive to the type of shower and to the environment. Therefore an adjacency-based clustering with distance constraints is generally preferred to avoid uncontrolled growth of clusters in high PU environments. An hybrid algorithm of this kind was developed for the reconstruction of the HGCAL trigger primitives. Examples of clusters produced by this algorithm are shown in Fig. 2.9 This strategy shares some common features with the clustering in the existing calorimeter trigger of CMS to reconstruct electrons/photons and hadronic tau leptons [1, 2, 4].

Once 2D clusters are reconstructed in individual layers, they need to be linked together such that the full energy deposited by showers is aggregated in the longitudinal direction. Several strategies have been studied here as well. The simplest one is to aggregate 2D clusters along a trajectory based on their distance from this trajectory. In order to reduce the complexity, trajectories are forced to be straight lines originating from the center of the detector. It effectively reduces the trajectory space to two dimensions. It could be for instance (η, ϕ), but the $(x/z, y/z)$ coordinate system is preferred as it better matches the physical space in which the showers are developing. This reduction of complexity has nevertheless a cost in terms of flexibility of the reconstruction. It doesn't take into account the fact that particles don't originate exactly from the center of the detector (the extreme case being long-lived particles) and that charged particles have bended trajectories. In those cases energy deposits in the $(x/z, y/z)$ plane have distorted shapes, which can potentially lead to unclustered energy. In order to seed such distance-based linking procedure, the 2D clusters are ordered, for instance by p_{T}. The highest-p_{T} 2D clusters form seeds around which lower and lower p_{T} clusters are attached within a given distance. More complex linking strategies using information on the density of 2D clusters can improve the reconstruction performance and in particular the resilience to noise. They aggregate clusters starting from denser regions and progressively move to sparser regions. Regions with too low densities can eventually be discarded to filter low-energy deposits. A well-known density-based clustering algorithm, studied in this context of reconstructing 3D clusters for the HGCAL TPG, is the DBSCAN algorithm [19].

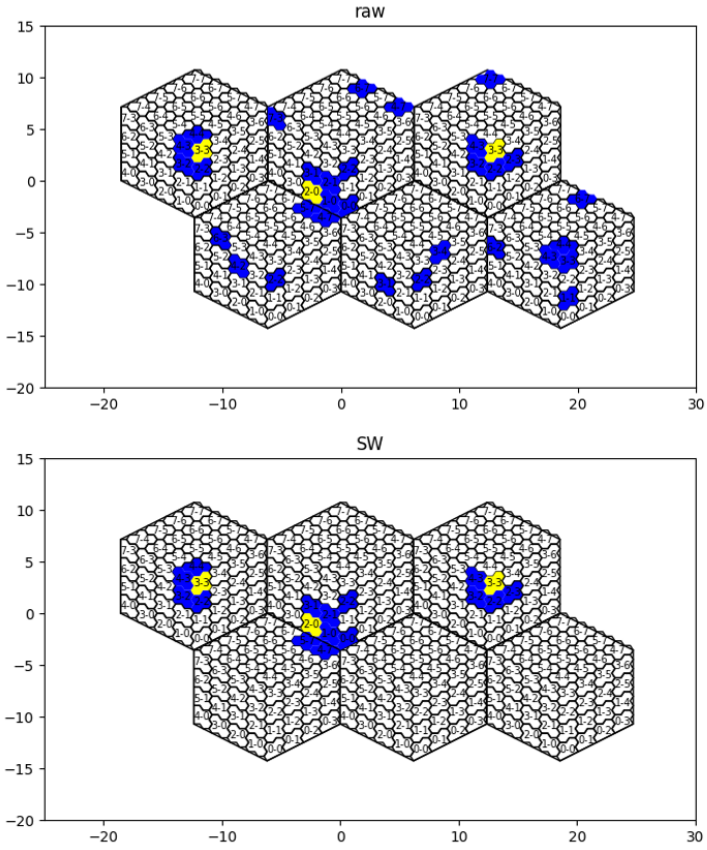

Figure 2.9: *Example of 2D topological clustering with distance constraints [18]. The seed trigger cells are drawn in yellow while trigger cells above the clustering threshold but below the seed threshold are drawn in blue. All trigger cells are shown in the upper plot, while clustered trigger cells are shown in the bottom plot.*

5 Longitudinal seeding and clustering

Shortly after the HGCAL TDR release, a different class of clustering algorithms has been developed and chosen for the prototype firmware being implemented as this text is written. It has been used to produce the results described in the L1T TDR released in 2020 [7]. This change was mostly driven by the firmware complexity of the 3D linking algorithms described in Section 4, which will be discussed in Chapter 4. The idea behind this alternative type of clustering is to separate the seeding procedure from the actual clustering. The data structure in the seeding step is also simplified, using a 2D histogram.

The projection of energy in the 2D seeding histogram is following the same principle as for the 3D linking described above. The projection is made in the longitudinal direction towards the center of the detector such that the energy deposited by a given shower is integrated mainly in one histogram bin. Performing the seeding with a sum of the shower energies over the longitudinal direction makes the clustering less sensitive to noise and PU, compared to a seeding based on single trigger cells or on 2D clusters. As detailed above already, this kind of projection procedure is not perfect for particles that do not originate from the center of the detector. But it is much less computationally demanding compared to reconstructing the detailed trajectory of each individual cluster seed. Reducing the

complex trigger cell point cloud into an histogram has also the advantage to provide a fixed data structure well suited for FPGA implementations. In addition, finding seeds is made easier by the array structure of histograms, and strategies developed for standard 2D calorimeters can be applied. A search of seeds as local maxima with a 3×3 window is performed, associated with the application of a minimum threshold. In order to reduce the sensitivity of the maximum finder to local fluctuations, a smoothing of energy is applied on the seeding histogram.

The coordinate system used to create the seeding histogram is important and should follow the geometry of the trigger cells and the lateral development of showers. In terms of seeding performance, using the $(x/z, y/z)$ coordinates with histogram bin areas matching roughly the area of trigger cells provides the best results. Nevertheless, due to data flow and firmware constraints the $(r/z, \phi)$ coordinate system is currently used, where r is the distance from the beam axis in the transverse plane ($r^2 = x^2 + y^2$). More details about these constraints are discussed in Chapter 4.

In the clustering step, trigger cells are aggregated around the identified seeds. Currently a distance criteria is applied in the $(x/z, y/z)$ plane. This distance can depend on the detector layer although this layer-dependency has not been fully optimized yet (in particular a fixed distance is used to reconstruct electromagnetic showers). The dependency with the detector layer can take into account the evolution of the shower spread in the lateral direction, using smaller distances in the first layers and larger distances deeper in the detector. And as discussed already in Section 1 of this chapter, this dependency with the layer can also help mitigating the contamination of PU, mainly located in the first detector layers.

Although the baseline seeding and clustering algorithm currently implemented in firmware has been made as simple as possible, variations around this idea of seeding-clustering have been studied in the past and will continue to be developed in the future. For instance, a similar strategy had been studied at the time of the CMS Phase 2 Technical Proposal [5, 14]. One of the main differences with the current algorithm was a seeding based solely on the detector layers around the electromagnetic shower maximum. The goal was to filter out PU energies more efficiently at the cost of a slightly reduced signal efficiency. Based on the same idea of filtering out PU energy deposits in the seeding step, it is also possible to apply layer weights when summing energies into seeding bins [20]. The weight profile can follow the expected longitudinal profile of showers, and in particular electromagnetic showers. In that case, larger weights are given to energies around the expected electromagnetic shower maximum, while lower weights are applied to energies deposited in the first layers where more PU is expected. Such weight profile is shown in Fig. 2.10. The selection of only a subset of the detector layers around the shower maximum is actually just a special case of such weighting technique with weights constrained to be equal to either 0 or 1.

6 Machine Learning in the context of trigger reconstruction

It was mentioned already in this chapter that machine learning techniques are used to perform some of the L1 trigger tasks and in particular for the identification of the reconstructed clusters. Using multivariate techniques at the L1 trigger is not completely new. BDTs were for instance developed in 2013 for the current electron and photon triggers as well as for muon triggers [4]. Nevertheless the machine learning models are not directly

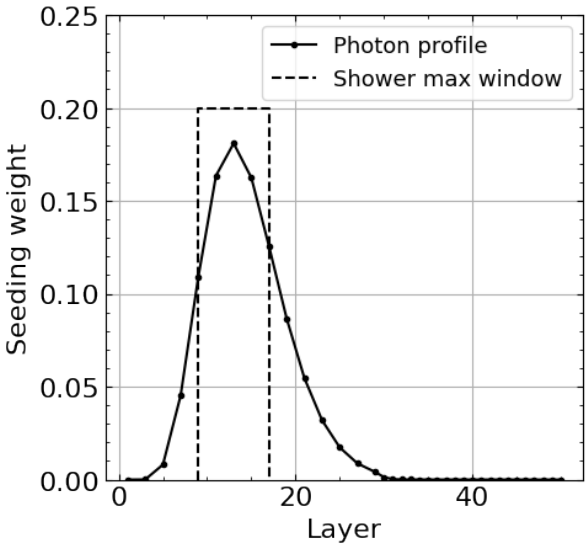

Figure 2.10: *Example of layer weights that can be applied for the seeding of electromagnetic showers. The full curve shows the longitudinal profile of unconverted photons with* 20 GeV $< p_{\mathrm{T}} \leq$ 100 GeV *in the absence of pile-up. The dashed curves shows a seeding window restricted to the electromagnetic shower maximum.*

implemented in firmware. Instead it is only their transfer function that is implemented in the form of large lookup tables. The main drawback of these implementations is the necessity to strongly compress the number of bits used to encode the input variables of the BDTs, due to limitations in terms of block RAM sizes in the FPGAs. For instance, only 12 input bits in total are used for the current electron and photon identification, distributed among three variables. A larger number of input bits (30 bits) are used for the muon trigger, but in that case the transfer function is stored in a dedicated memory module and not within the FPGA matrix.

Since more recently, and in particular in the context of the Phase 2 trigger developments, the actual implementation of machine learning models, in particular BDTs and neural networks, is considered. It has been made possible thanks to larger FPGA matrices available nowadays as well as to the emergence of new tools to convert trained machine learning models into firmware [21, 22]. Given the high dimensionality and complexity of the HGCAL data it is obvious that machine learning models will be used in several places of the TPG reconstruction and of the central CMS L1 trigger. Indeed, simple cut-based selections are not efficient enough in reducing trigger rates in the HL-LHC environment. Shallow models such as BDTs and fully connected neural networks are already foreseen in some places of the trigger reconstruction, in particular for energy corrections and cluster identification as mentioned previously. It is indeed not possible to find just a couple of variables that are powerful enough in discriminating different types of showers. In addition these variables are correlated between each other, and they need to be combined in order to really benefit from the HGCAL 3D information. These tasks can benefit from multivariate machine learning models since such models are built to make use of correlations between input variables. This is illustrated in Fig. 2.11, where the ROC curves obtained with a BDT using 13 variables and with a single cut on the most discriminating variable $E/(E+H)$ are compared. Background clusters are defined here as any reconstructed cluster with $p_{\mathrm{T}}^{\mathrm{clu}} > 20\,\mathrm{GeV}$ in minimum bias events simulated with 200 PU. Although the efficiency of identifying such cluster as an electromagnetic cluster is not equivalent to e/γ trigger rates, it provides a good proxy to assess the level of rate reduction that can be obtained.

So far, only machine learning models using high-level input features, such as cluster

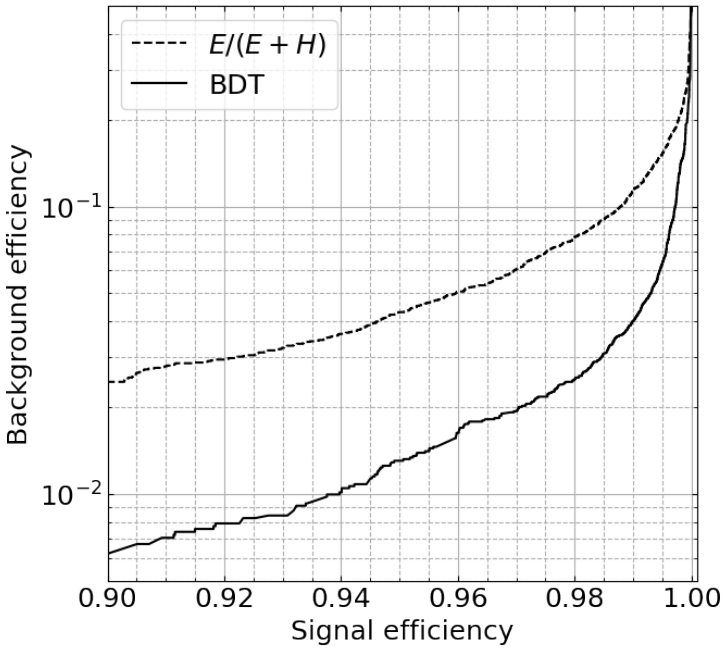

Figure 2.11: *ROC curves for the identification of electromagnetic clusters against PU clusters. The ROC curve obtained with a BDT is compared with the one obtained using only the most important variable $E/(E+H)$ (see Fig. 2.8 for the list of variables). PU clusters are defined here as any reconstructed cluster with $p_{\mathrm{T}}^{clu} > 20\,\mathrm{GeV}$ in minimum bias events simulated with 200 PU. Electromagnetic clusters are defined as clusters matched to a generated electron with $20\ \mathrm{GeV} < p_{\mathrm{T}}^{\mathrm{gen}} < 100\ \mathrm{GeV}$ in electron gun events simulated with 200 PU.*

shape variables, have been discussed. This strategy requires to manually design such variables and to select the most efficient ones for the task under consideration. Deep neural networks (DNN), on the other hand, are able to process raw data directly, such as energy hits. DNNs encode hierarchical models of the data and extract high-level features from these raw data. In practice it means that DNNs can reconstruct the energy of particle showers and identify their type directly from the deposited energies. In fact, DNNs can even process data from the entire detector, reconstruct all the individual showers in the detector, and associate energy hits to these reconstructed showers. This task is called object segmentation. This kind of end-to-end reconstruction is mainly studied and developed in the context of the offline reconstruction [23, 24]. It is at the moment not feasible at the L1 trigger due to the gigantic data throughput, the limited hardware resources and the limited processing latency. Nevertheless, even if object segmentation seems out of reach at the moment at the L1 trigger, DNNs could still be used to process raw trigger cells data from already reconstructed clusters, on a cluster-by-cluster basis. DNNs are much more demanding in terms of FPGA resources compared to more classical shallow models, but the computation of high-level input features is not needed in that case. For such strategy mainly two types of networks can be used, convolutional neural networks (CNN) and graph convolutional neural networks (GCNN).

Standard CNNs have been originally developed to process images and are therefore based on input data organized in the form of grids of square pixels. Image-like data structures have the main advantage of being easily processed in FPGAs, but the HGCAL has a more complex geometry and energy deposits cannot be encoded in a standard image

without some information losses. GCNN on the other hand can work on irregular data structures (graphs) and even for some classes of GCNN, called dynamical GCNN, on point clouds without any predefined structure [25, 26]. In that case there is no information loss due to the conversion to a data structure that doesn't match the detector geometry. Developments on GCNNs have been growing exponentially for the last few years, in particular in the context of the offline reconstruction. Some attempts to use such networks in the context of the L1 trigger have also been made [27]. They usually require much more hardware resources than fully connected networks and BDTs, in particular in the cases where the data structure is dynamically learnt by the network, typically using k-nearest neighbor (kNN) algorithms. But given the continuous improvements in FPGA hardware and given the current developments in neural network compression techniques, it is expected that more and more complex machine learning models will be running in L1 triggers in the future.

7 Conclusion

In this chapter the different steps of the L1 trigger primitive generation in the HGCAL have been described. In the past years different strategies and methods were tested and only the main general principles have been extracted and presented here. These developments did not follow a linear process and multiple paths have been explored, sometimes leading to dead ends, sometimes abandoned and re-explored later. These trials and errors were unavoidable and even necessary in the prototyping phase in which we were. This is also exacerbated by the fact that firmware implementations arrive in general with a delay compared to simulations and performance studies, due to a longer development time. Algorithms need then to be updated or eventually abandoned after feedback from firmware designers.

The two next chapters will touch a bit the hardware and firmware aspects of the HGCAL TPG. The two main things that put constraints on the trigger primitives reconstruction are (1) the very high flux of data that come out of the detector and (2) the fact that the reconstruction algorithms need to run on FPGAs with an extremely low latency of the order of the microsecond. Chapter 3 covers the frontend part of the trigger data processing and in particular the reduction of the data flow. Chapter 4 covers the implications of the data structure coming out of the detector and of the backend electronics on the reconstruction algorithms.

3

Trigger data reduction in the frontend

In the previous chapter, the different steps of the cluster reconstruction and identification developed for the HGCAL TPG have been presented. Different strategies, in particular for the clustering, have been described and can potentially be used. Nevertheless, very little has been said on the actual implementation of these steps and on the constraints and limitations generated by the hardware and firmware.

The first set of constraints come from the FE electronics and, maybe more importantly, from the limited bandwidth between the FE and the BE. The main consequence is that the reconstruction algorithms cannot be directly applied on the full detector data. There are several underlying reasons for that:

- Power consumption: the trigger data, sent at the frequency of 40 MHz, have to be reduced as soon as possible in the electronics chain to avoid consuming too much power by moving, sending, receiving and processing these data. The total power budget for the FE electronics is about 20 mW per channel, which is limited by the cooling power that can be installed.

- Routing: the number of routing lines required to transfer trigger data on the PCBs and between PCBs is limited, in particular given the large number of FE ASICs.

- Cost: the optical links used to transfer trigger data between the FE and the BE, as well as the FPGAs in the BE needed to receive these data are expensive items, and their numbers need to be kept within a reasonable range. About 10000 lpGBT links are currently allocated to transfer trigger data from the two endcaps.

- Space: the space allocated for the HGCAL TPG and DAQ in the CMS underground services cavern (USC) is limited to 16 racks. Since the number of FPGAs required to receive the trigger data coming from the FE is (at first order) proportional to the amount of received data, it also puts limits on the size of these data.

For these multiple reasons, trigger data need to be reduced as much as possible and as soon as possible in the electronics chain. Nevertheless, data reduction strategies are in general lossy, which translates into degradations in the ability to reconstruct and identify

particle showers. In order to limit the impact on the performance of the trigger system and at the same time reduce sufficiently the data rate, several compression strategies based on different dimensions of the data need to be combined. These compression strategies can be divided into three main classes:

- compression methods impacting the spatial granularity of input cells
- compression methods impacting the energy resolution of input cells
- compression methods reducing the population of input cells

1 Data reduction based on spatial granularity

Trigger cells are the core objects of the HGCAL TPG. Reducing the granularity of the detector is the first and simplest data compression strategy applied in the trigger chain. It provides a simple way of reducing the data throughput from the HGCROC by a factor of 4 (in low-density modules) or 9 (in high-density modules). The size of trigger cells is roughly equal to $2 \times 2\,\mathrm{cm}^2$, to be compared with the Moliere radius equal to about 3 cm. It means that electromagnetic showers deposit most of their energy in very few cells in each layer. The consequence is a reduced capability to disentangle overlapping showers and to filter out PU energy deposits. It also impacts the granularity with which cluster shapes are reconstructed and therefore reduces their power to discriminate electromagnetic and hadronic showers.

Trigger cells are formed by summing together the deposited energy (more exactly the digitized charges as no calibration is applied at this stage) in predefined groups of cells geographically close to one another, as illustrated in Fig. 3.1. This summation requires the input quantities to be at the same scale. Given that two digitization modes, the ADC mode and the ToT mode, are used, a rescaling procedure needs to be applied before the trigger cells are summed. Indeed, the digitized values obtained from these two modes are using different scales: their relationship with the input energy is different. The main issue is that the ToT response is highly non linear for low charges close to the ADC regime. Restoring the linearity in the HGCROC before the TC summation would consume too much power as it would need to be done for every sensor cell at a frequency of 40 MHz. Therefore the rescaling of the ToT values onto the ADC scale is only approximate, which results in errors around the transition region between the ADC regime and the ToT regime. In order to minimize the size of the errors, a plateau is used in this transition region. Two possible versions of this plateau are shown in Fig. 3.2. The one chosen and implemented in the HGCROC creates a balance between negative and positive errors, while the alternative one creates only negative errors.

In addition to this simple granularity reduction strategy (building trigger cells), a more sophisticated procedure has been developed, to be applied on top of the TC sums and reduce further the granularity. The general idea is to sum groups of TCs into larger objects, which correspond at first order to larger TCs, but to additionally retain some information on the distribution of energy within these groups. As an example such information could be the barycenter of the energy deposits in each group. A simpler alternative that has been chosen and implemented in the ECON-T [29] is to tag the trigger cell containing the largest energy deposit inside the group. In terms of information it is equivalent to summing the energy of the group and concentrate the energy sum into the TC with the largest energy. The implementation in the ECON-T can build STCs from 4 or 16 TCs,

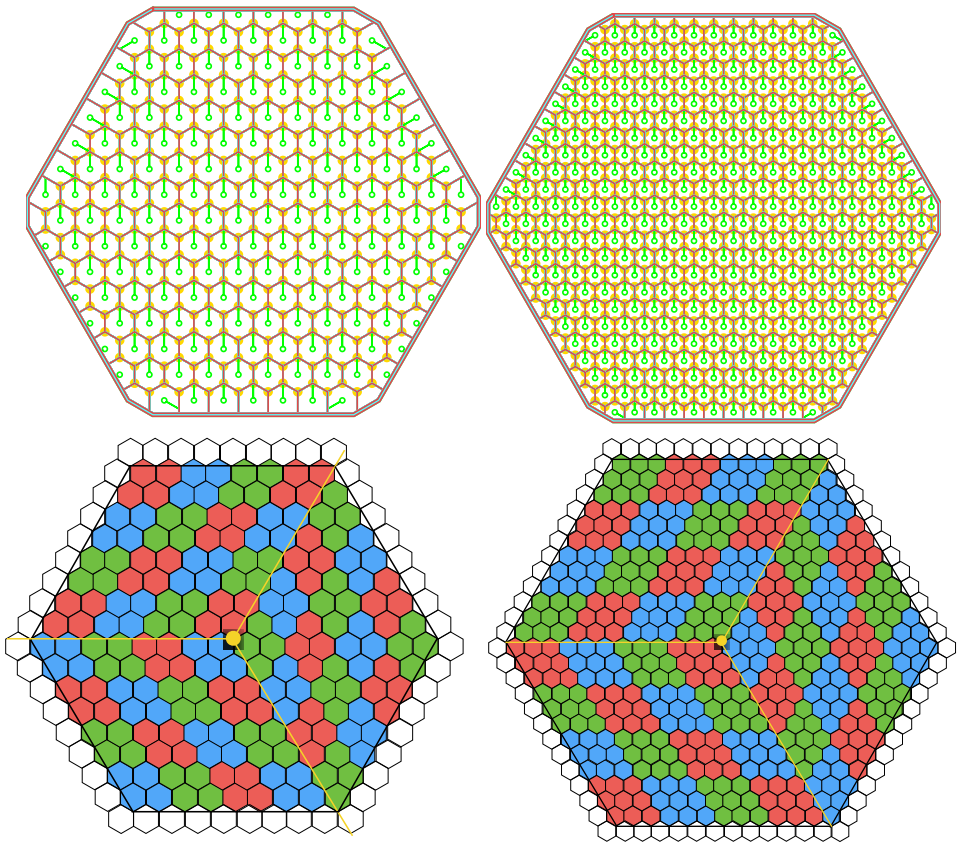

Figure 3.1: *Low granularity (left) and high granularity (right) hexagonal modules [6]. The sensor cells tiling within these modules is shown on top while the grouping of sensor cells into trigger cells is illustrated on the bottom.*

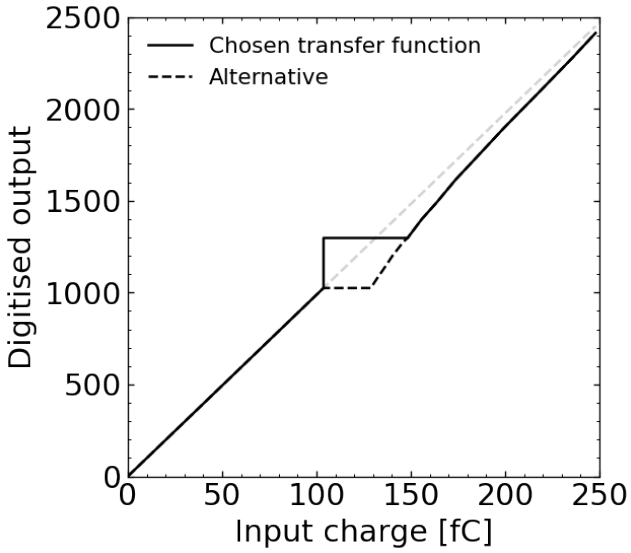

Figure 3.2: *Digitised output as a function of the input charge after rescaling of the ToT values to the ADC scale [28]. The ToT mode is used typically above* $100\,fC$ *when the preamplifier saturates. Since the ToT response is highly non linear for low charges, a plateau is applied in order to minimize errors in the transition region between the ADC and the ToT modes. Two different transfer functions in this transition region are shown.*

depending on the level of data reduction required. There are other, more sophisticated alternatives, for instance making use of the energy fraction deposited in the maximum cell. But the simpler strategy described above has proven to provide the best trade-off between performance and data size. The objects formed in this way are called Super Trigger Cells

(STC), and are illustrated in Fig. 3.3.

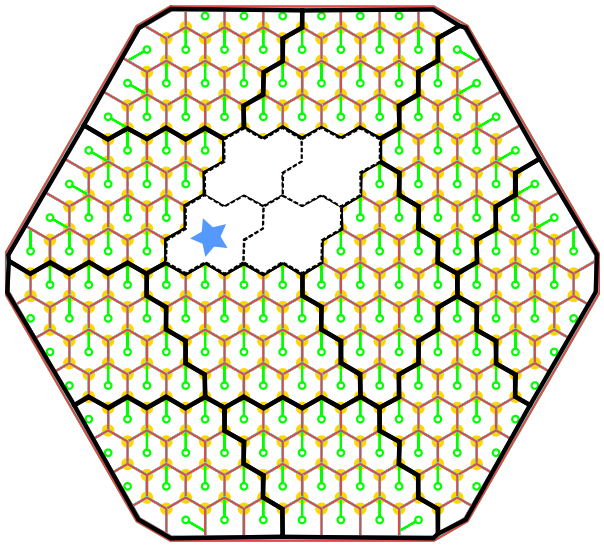

Figure 3.3: *Illustration of Super Trigger Cells (STC) formed from four trigger cells in a silicon module. The TCs contained in one of the STCs are shown in white and the blue star indicates the TC with the maximum energy in this STC. The position of this TC is retained in the STC information.*

2 Energy compression

A total of 19 or 21 bits, depending whether trigger cells are built from 4 or 9 sensor cells, is needed to encode TC charge sums over the full dynamical range. In a module containing 48 trigger cells, it corresponds to 36.48 Gbps and 40.32 Gbps, respectively. It is nevertheless not feasible to send such wide data downstream in the trigger chain. Indeed twelve 1.28 Gbps output links are allocated per module to send trigger data out of the HGCROC, corresponding to a bandwidth of 15.36 Gbps. Among this total bandwidth, only 13.44 Gbps can effectively be used, the rest being needed for synchronization purpose. The TC charge sums are therefore compressed by a factor of ~ 3 immediately after their formation in the HGCROC.

At the L1 trigger a precise resolution is usually not needed for particles with high energy well above trigger thresholds. Therefore, high energies don't need to be encoded with as much precision as lower energies. In addition, the absolute energy resolution in calorimeters typically grows as $\sqrt{E}$ or E for sufficiently large energies. It is therefore not necessary to use a linear coding scheme that provides constant errors as a function of the energy. A logarithmic encoding of the TC charges better follows both the intrinsic calorimeter resolution and the trigger requirements, as the compression errors in that case grow as E. But implementing a real logarithmic encoding is not viable in the FE, and would also be too complex to decode in the BE. A simpler coding scheme, that follows the same principle of increasing the absolute errors with the energy, is the floating point encoding. The floating point encoding is based on two terms, an exponent term to encode the position of the most significant bit (MSB) and a mantissa term to encode the bits right after the MSB. The least significant bits are therefore dropped for high values, which leads to truncation errors increasing in absolute value as the energy increases. Logarithmic encoding and floating point encoding share very similar transfer functions as can be seen in Fig. 3.4, where examples of logarithmic and floating point encodings are compared.

For a fixed total number of bits, the allocation of these bits to the exponent and to the mantissa can vary. The choice of their relative allocation depends on the importance

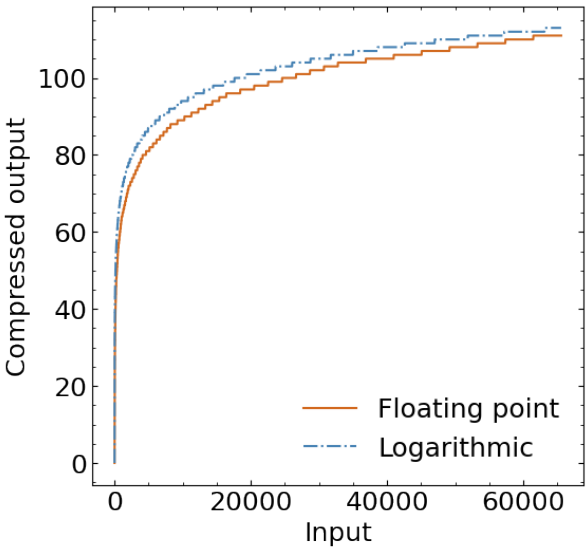

Figure 3.4: *Response functions for logarithmic and floating point compressions on 7 bits. The floating point compression is using 3 mantissa bits and 4 exponent bits. Both compression schemes shown here as example cover inputs on 18 bits. A reduced range is displayed in order to better visualize the differences between the two compression schemes. A floating point compression provides a very similar response as a logarithmic compression while being much less demanding in terms of power and logic resources.*

of limiting truncation or saturation effects. A lower number of mantissa bits gives larger truncation errors, while a lower number of exponent bits provides a lower dynamic range and gives larger saturation effects. Given the large range covered by energy hits in the HGCAL, the number of bits used to encode the exponent is critical as too important saturation effects at the trigger cell level can be disastrous for cluster energies. Reducing the number of exponent bits can be harmless until it reaches a certain point where the bulk of shower energy deposits become saturated. But the cluster energy resolution is then strongly damaged below this point. On the other hand, truncation effects also impact cluster energies, but they can be mitigated by a rounding mechanism such that positive and negative rounding errors compensate in cluster energy sums. Reducing the number of mantissa bits affects the cluster energy resolution in a much more gradual way. This is illustrated in Fig. 3.5 where the energy response and resolution of reconstructed photon clusters in particle gun events simulated without PU are shown for different exponent and mantissa sizes. There is a sharp increase of the resolution and drop of the response for a number of exponent bits below 4, while their dependency with the number of mantissa bits is much milder.

The number of bits to encode TC energies is limited to 7 bits due to bandwidth constraints in the output of the HGCROC. It has been chosen to share these 7 bits into 4 exponent bits and 3 mantissa bits. This 4E3M floating point encoding can cover a linear range of 18 bits. The total TC sum dynamic range can potentially cover 19 bits for low-density modules and 21 bits for high-density modules, so one or three of the least significant bits are truncated before compressing the energy into the 4E3M floating point format. This compression scheme ensures that the full dynamic range is covered, and therefore eliminating saturation effects, at the price of a bit more rounding errors.

The floating point encoding reduces significantly the size of the data to be transmitted, but requires to be decompressed in the later stages every time operations are done on TC energies, in particular when summing together several energies or applying calibration factors. It is for instance what is done in the ECON-T where TC sum values are decompressed, calibrated and recompressed before being sent to the BE electronics [29]. Nevertheless the decompression of such floating point format is relatively lightweight in terms of needed logic operations.

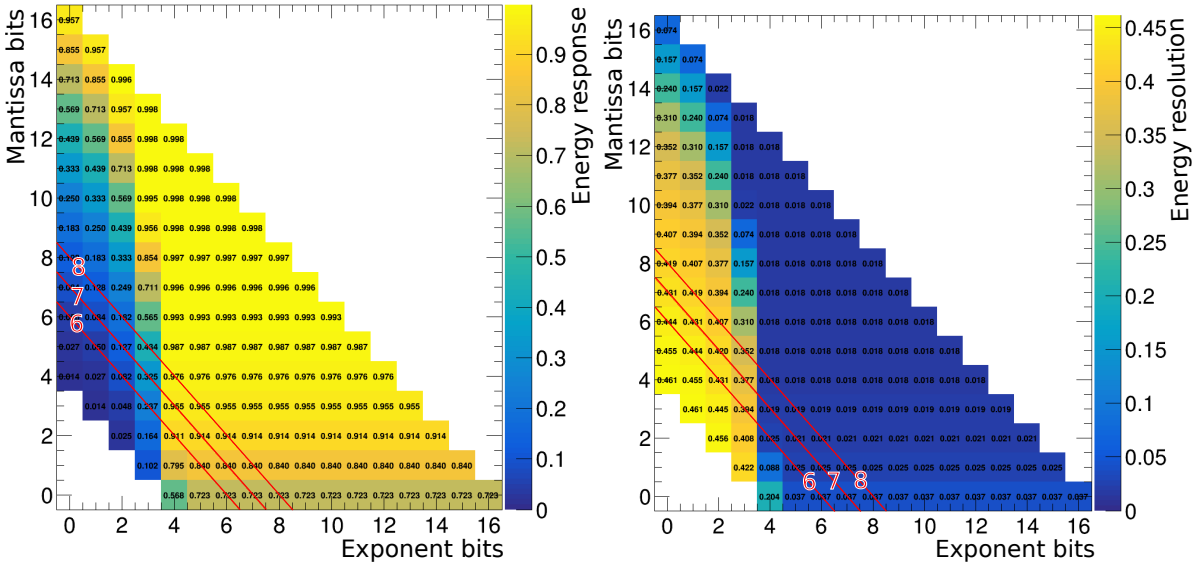

Figure 3.5: *Energy response (left) and resolution (right) for photon showers as a function of the numbers of bits used to encode the exponent and the mantissa in a floating point format. Photons are generated without PU with a uniform distribution in* $10\ \text{GeV} < p_\text{T} \leq 150\ \text{GeV}$ *and* $1.7 < |\eta| < 2.8$*. The response is defined as the mean of the* $p_\text{T}^\text{L1}/p_\text{T}^\text{gen}$ *distribution and the resolution is defined as the RMS of this distribution divided by its mean. The three red lines indicate a total number of bits (exponent + mantissa) of 6, 7 and 8.*

Others, more sophisticated, way of compressing TC energy that had been considered at the beginning of the HGCAL project are variable-length types of encoding, such as Huffman coding. These coding strategies offer lossless compression of the data by making use of their statistical properties and redundancy. The Huffman encoding uses less bits to code the most frequent patterns and uses more bits to code the less frequent patterns, which reduces on average the size of the data. Such method is nevertheless too complex to be implemented in the FE and in the BE, as it typically requires to pass through a binary tree to encode and decode the data. It was therefore quickly abandoned.

3 Population-based data reduction strategies

The last way in which data can be reduced is by acting on the population of trigger cells, selecting only a subset of all the original trigger cells. Usually only the trigger cells with the highest energy deposits are selected. Two ways to perform such selection have been developed and studied:

- Applying a threshold on the trigger cell energy and suppressing trigger cells falling below this threshold.
- Defining a fixed number of the highest-energetic trigger cells to select and suppressing the lowest-energetic trigger cells.

The selection by threshold has the main advantage of being able to send more trigger cell energies in a given bandwidth due to averaging effects across several consecutive bunch crossings. Events with lower-energy deposits can free the bandwidth for events with

higher-energy deposits, provided that a buffering mechanism is implemented. Its main drawback is the variability, event-by-event, of the data size around an average value. This average value and its variability depend on the running conditions of the LHC and of the detector. This variability also implies that the data sent from different locations in the detector are asynchronous, and need to be synchronized at a later stage, typically in the FPGAs receiving the data in the first stage of the BE. This task of synchronization is something highly demanding in terms of logic resources. This selection strategy also requires additional latency, both in the FE in order to benefit from the averaging effect through buffering, and in the BE in order to synchronize and align all the data coming from the FE.

Finally, since the data size depends on the running conditions of the LHC, and in particular on the instantaneous luminosity and the level of PU, the threshold values and the bandwidth allocation require a careful tuning. This tuning is performed for each individual module, looking primarily at the average occupancy (the number of selected trigger cells in the module) for different threshold values and converting it into a number of links. A sufficient margin (typically 30%) in the allocated bandwidth is added to reduce as much as possible the probability of overflow due to positive fluctuations of the occupancy. An example of trigger cell occupancy map and the associated number of links in a CE-E layer is shown in Fig. 3.6. Constraints coming from the architecture of the FE electronics system are then considered to adjust the map of links and the threshold values.

The other selection strategy, selecting a fixed number of the highest-energetic trigger cells does not benefit from averaging effects across bunch crossings as the data size is by definition fixed. Therefore more trigger cell energies are discarded in the case of busy events compared to the application of a threshold, although the fact that the highest-energetic cells are selected alleviates this truncation effect. Averaging effects could be obtained by selecting the highest-energetic trigger cells across several consecutive bunch crossings instead of a single bunch crossing. But this idea has been put aside as it creates artificial correlations between consecutive bunch crossings. Such correlations would prevent to emulate properly the trigger chain on recorded data since only a small fraction of the bunch crossings are kept and stored. Nevertheless, despite its disadvantage, this selection scheme creates a synchronous stream of data since a fixed number of trigger cells is sent for each event. Such synchronous data stream doesn't require as complex firmware in the receiving stage of the BE compared to asynchronous streams. For the same reason the data doesn't need to be buffered and events are sent within a 25 ns window. The latency required to send and receive the data is thus minimal.

At the moment of writing this text, it is foreseen that the selection of TCs in the ECON-T will be done by applying a threshold and selecting TCs with energies above this threshold. But given the complexity expected to unpack an asynchronous stream of data on the receiving stage of the BE, both options have been implemented in the ECON-T. The threshold implementation is the simplest, its main complexity being in the buffering mechanism. Selecting the highest-energetic trigger cells is more complex since trigger cells need to be sorted from the highest energy to the lowest energy. Sorting is in general hungry in terms of logic and/or latency. For our use case with a very high flux of data, a relatively low number of values to sort, and where fixed latencies are required, sorting networks are the best class of sorting algorithms. They are particularly well-suited for ASIC and FPGA implementations as they have an intrinsically parallel and pipelined architecture. In particular bitonic sorters and odd-even mergesort sorters, although not being necessarily the most efficient sorting networks for every cases, are the best network architectures in

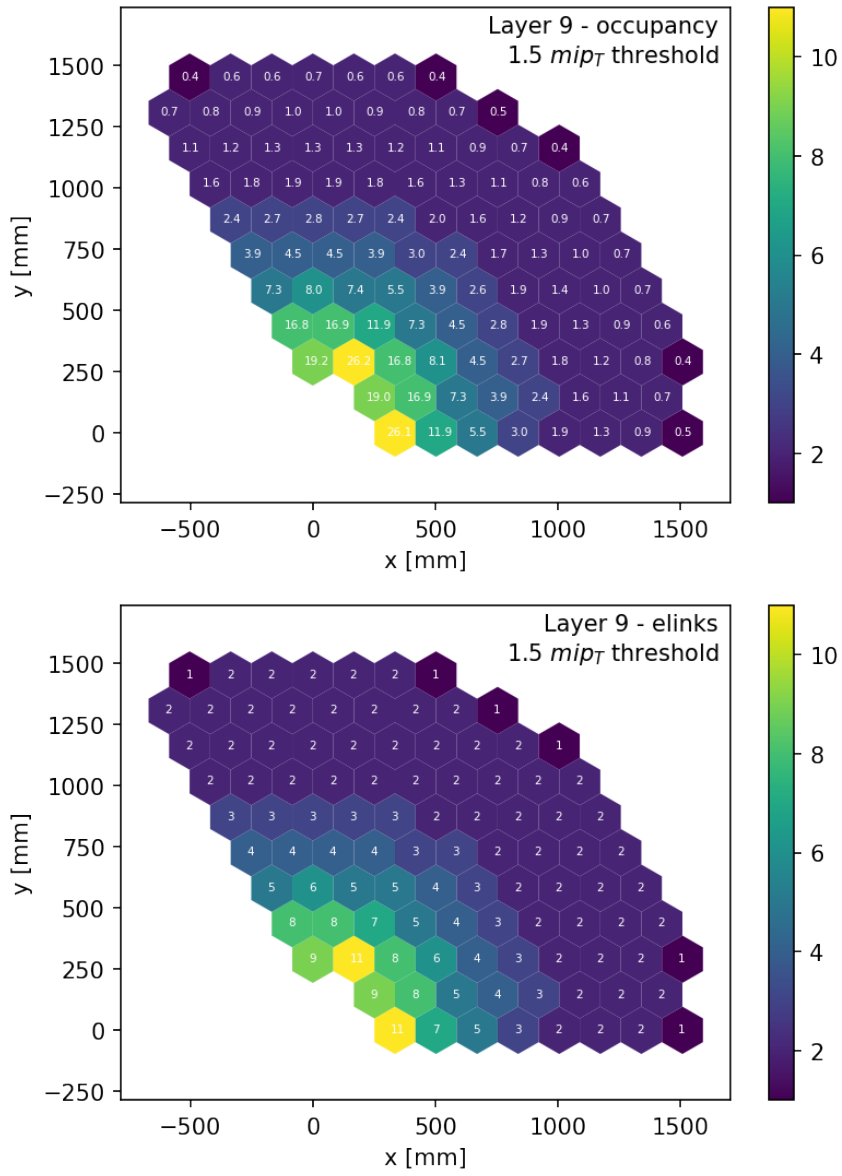

Figure 3.6: *TC occupancy (top) and number of elinks (bottom) per module in a* 120° *sector of layer 9. These values have been obtained with a* 1.5 mip_T *threshold applied on the TC energies. An overhead of* 30% *is applied on the occupancy when deriving the number of required elinks.*

general. A Batcher odd-even mergesort architecture [30] has been chosen to perform the TC sorting followed by the selection of the highest-energetic TCs. This implementation has been called *BestChoice*.

Since the BestChoice selection scheme, implemented as an alternative to the threshold selection, does not benefit from buffering and averaging effects, its performance are quickly degraded when the allocated bandwidth becomes too low compared to the number of hits to be sent. It is especially true for large objects such as hadronic jets. Therefore, it has been chosen to combine the BestChoice selection with the Super Trigger Cells data reduction strategy described in Section 1. BestChoice would be used in the CE-E and STC would be used in the CE-H. This combination provides similar performance as a threshold selection for electrons and photons, hadronic taus and hadronic jets, and it is therefore a more

viable alternative to the threshold selection. The single e/γ (electron/photon), single tau and single jet trigger rates are compared for different data reduction strategies in Fig. 3.7.

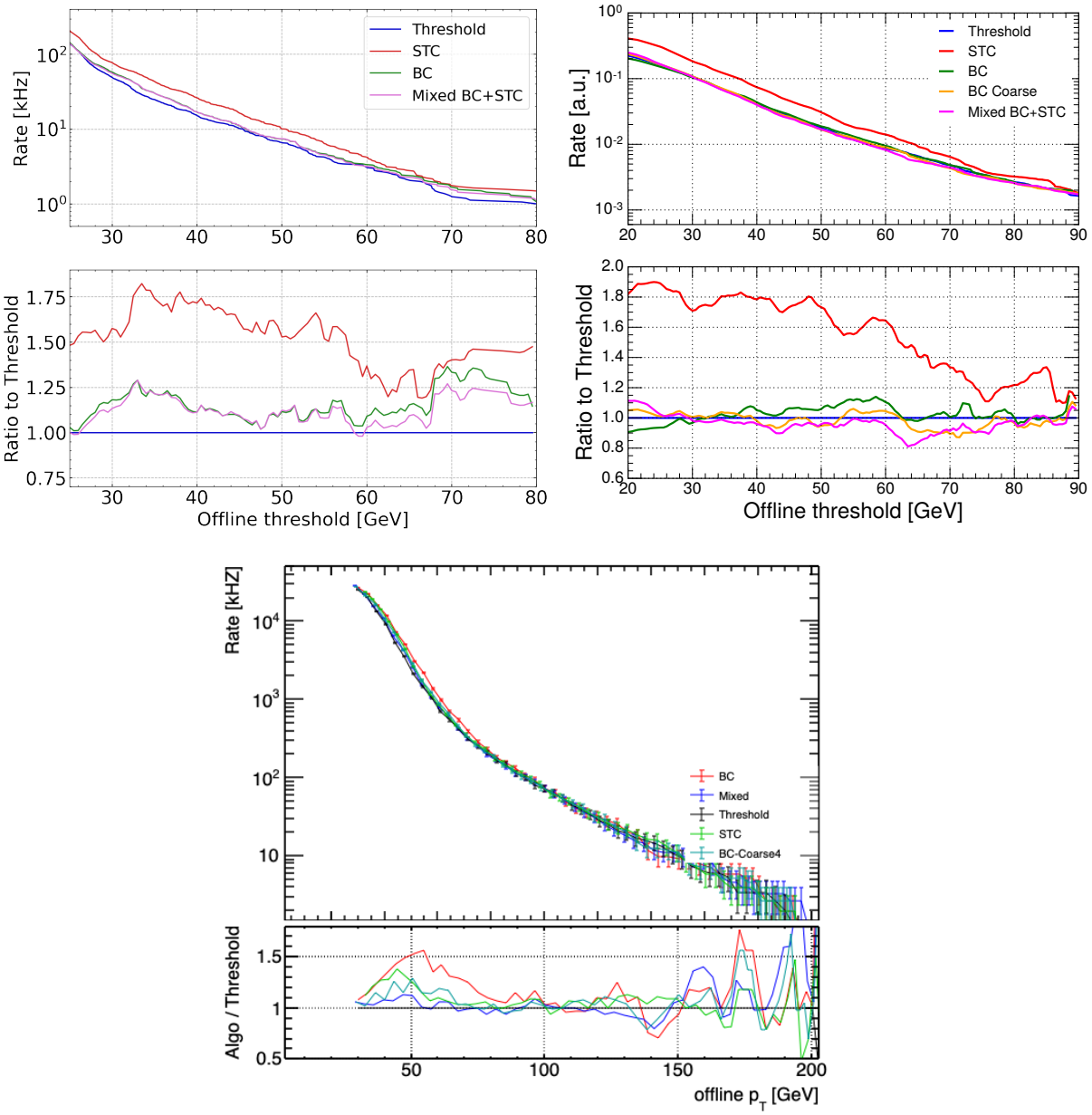

Figure 3.7: *Rates for single e/γ (top left), single tau (top right) and single jet (bottom) triggers as a function of the* offline threshold *for a TC data reduction in the ECON-T based on a threshold (1.35 mip_T), the BestChoice (BC) selection, the SuperTC (STC) granularity reduction or on a combination of the BestChoice and SuperTC (Mixed BC+STC) [31, 32]. The ratios with respect to the rates obtained with a threshold are also shown. The "offline threshold" corresponds to the offline p_T for which a 95% signal efficiency is obtained at a given trigger rate.*

4 Impact of the detector design on compression algorithms

Some of the compression strategies presented above are local (at the level of single cells), while other compression strategies work at the level of a population of cells. But in both cases the design of the detector and of the FE electronics architecture can play an important role on the feasibility and efficacy of these compression algorithms.

The first most important design aspect is the choice of the geometry of the detector. For the HGCAL, hexagonal modules have been chosen in the silicon section, mainly due to a better silicon wafer efficiency compared to square modules. Consequently, hexagonal cells are used in these hexagonal modules. But hexagonal cells cannot be perfectly tiled inside hexagonal modules, and the hexagonal tiling needs to be slightly broken at the boundaries between modules. This tiling issue becomes more complex when considering granularity reduction in trigger data. In that case sensor cells are grouped together into trigger cells, and this grouping needs to be done in the most uniform way as possible. This grouping also needs to be ideally done in a single FE ASIC, which means that trigger cells should be entirely contained in a region covered by single ASICs. The tiling of sensor cells into modules and their grouping into trigger cells have been subject to intense studies during the year 2016 [20]. A module architecture emerged from these studies where the module hexagon and the sensor cell hexagons are rotated between each other by 90°, and the center of the module matches a cell vertex. The module hexagon is virtually divided into three diamond-shaped regions, rotated from one another by 120°. A grid of 4×4 trigger cells (being made of either 4 or 9 sensor cells) can then be defined in each of these diamond regions. This module and trigger cell architecture is illustrated in Fig. 3.1.

It preserves a very good sensor cell and trigger cell uniformity, and the only minor gaps from a perfect tiling are:

- slightly distorted sensor cell shapes and non-uniform areas at the edges of the modules,
- a threefold rotation of trigger cells within modules.

The choice of this specific module architecture has been strongly driven by trigger requirements. These trigger requirements are largely due to the fact that FE ASICs cover small regions. In particular the HGCROC covers 1/3 (for low-density modules) or 1/6 (for high-density modules) of one module. Trigger cells are built in the HGCROC and should therefore be entirely contained in single HGCROCs. Although this is not a strict requirement, as trigger cells could potentially be built in several steps, it simplifies largely the data flow.

More generally, the area covered by FE ASICs and the number of FE processing stages is a critical aspect of the detector design for what concerns the trigger data. Indeed some data reduction algorithms are sensitive to the area (the number of cells) covered by individual FE ASICs. The case of the TC building has already been mentioned, although it is actually more an edge-related issue. But smaller coverage implies necessarily more edge problems. The other data reduction algorithms impacted by the coverage of FE ASICs are algorithms based on a population of cells. In particular, selecting a fixed number of the highest energetic trigger cells among a set of input trigger cells is performing better with a more global selection compared to a more local selection. If a fixed number of output links is assumed in the whole HGCAL, partitioning them in smaller regions means that fewer trigger cells can be sent in each region. At first sight the dependency between the

region size and the number of trigger cells is linear. But non-linear effects arise due to the fact that interesting energy deposits are local and dense. In that case, reducing too much the region size implies that the energy deposited by a particle in one single region could cover more trigger cells than what can be sent from this region. So even though the total number of links in the whole detector stays the same, such selection algorithms are less efficient in filling the available bandwidth with local high-energy deposits if the detector is partitioned into smaller regions.

Similar conclusions holds when trigger cells are selected by applying a threshold on their energy (trigger cells with an energy above a given threshold are selected), even though such threshold is applied at the level of single cells. This is due to the fact that with such selection strategy, buffers need to be used as the number of selected trigger cells varies from one event to an other. The goal of these buffers is to absorb the positive fluctuations above the average number of selected trigger cells. Assuming that the distribution of the number of selected trigger cells above threshold follows a Poisson distribution, a smaller region coverage implies larger fluctuations relatively to the average value. Therefore larger buffers, relatively to the region size, need to be used to absorb fluctuations if smaller regions are used.

In part for these reasons, compression methods that reduce the population of cells are not applied in the HGCROC but rather in the second processing stage, in the ECON-T. The role of the ECON-T is indeed to concentrate data from severals ROCs and therefore covers larger regions of the detector.

5 Conclusion

Due to the limited bandwidth that can be allocated to send trigger data out of the detector, the raw data produced by the detector needs to be significantly reduced. Given the level of data reduction required, a combination of multiple strategies staged on two FE ASICs have been developed. These strategies play with different aspects of the data in order to limit as much as possible their impact on the reconstructed clusters and on the performance of the trigger. The strategies implemented also take into account their implementation complexity and the complexity they imply in the BE electronics. Although most of the compression chain is now finalized, there are still uncertainties at the moment of writing this text regarding the algorithms to be used in the ECON-T. Their final configuration will take into account the expected trigger performance as well as constraints from the FE system and from the BE system. Results and a decision are expected by the end of 2022.

This chapter has been describing how the data taken as input by the reconstruction algorithms need to be altered due to hardware constraints and technical limitations on the detector and in the FE electronics. The next chapter will cover the hardware and firmware constraints in the BE electronics and how they impact the implementation of the reconstruction algorithms described in Chapter 2.

4

Some hardware and firmware aspects of clustering

The different algorithmic steps composing the HGCAL TPG have been detailed in Chapter 2. Among these steps the clustering step is the most critical part of the reconstruction as many other steps, such as the application of energy corrections and the particle identification, depend on the quality of the clustering. The implementation of a clustering algorithm in a multi-stage trigger system composed of many FPGAs is far from being straightforward. The hardware and firmware put strong constraints on the algorithms that can be implemented. It means also that the structure of an algorithm and the hardware architecture on which the algorithm is implemented are interdependent. Some hardware architectures are well suited for certain algorithms but not for other algorithms. The different types of hardware architectures associated with the clustering algorithms described in Chapter 2 will be detailed here. But first the impact of the detector geometry and of the structure of the data coming from the FE is discussed.

1 Data flow and system architecture

The most central architectural requirement is the presence of at least two processing stages to be able to reconstruct 3D clusters. Indeed, in order to efficiently build 3D clusters, data from significant portions of the detector need to be concentrated into single FPGAs. This is particularly true for the HGCAL because of its non-projective geometry. Each optical link from the FE transports data from only one to a few modules from a single layer. Building 3D clusters directly in a single BE stage would require to receive all the FE links in depth from at least a 60° sector, given the symmetries provided by an hexagonal tiling and the way FE motherboards and lpGBT links are organized on the detector. But that would make a total number of around 800 links, while a bit more than 100 transceivers at most are available in state-of-the-art FPGAs (for instance 128 in the Xilinx VU13P FPGA, planned to be used).

In the different HGCAL TPG systems developed so far, two BE processing stages have therefore been foreseen, called *Stage 1* and *Stage 2*. The main goal of the Stage 1 is to

concentrate the FE data and compress or truncate it further such that Stage 2 FPGAs have access to large enough sectors of the detector. The Stage 2 is then designed to perform the main reconstruction work of the HGCAL trigger primitives (the clusters and the towers). Some steps of the clustering and of the reconstruction of the towers can nevertheless be performed in the Stage 1 in some cases in order to reduce the load of the Stage 2.

Single Stage 2 FPGAs should cover in-depth sectors of at least 60° given the symmetries of the detector originating from the hexagonal modules. In the case where the Stage 2 FPGAs have access to only sectors of one endcap, the data close to the boundary of two sectors covered by two different FPGAs need to be duplicated in order to correctly reconstruct clusters in this transition region. It means that additional links and transceivers are needed to transmit and receive the duplicated data. Also, these duplicated data need to be processed twice, in the two different FPGAs, which increases the overall logic resource need of the system. Therefore the data from one full endcap should ideally be processed by a single Stage 2 FPGA in order to eliminate data duplication completely and maximize the resource efficiency of the system. This is nevertheless not always possible, depending on the level of data compression done in the Stage 1.

Each Stage 1 FPGA covers only a small fraction of the detector and receives data organized by FE modules, packed into lpGBT links. In general the number of FE modules covered by a Stage 1 FPGA differs from one to another since the bandwidth allocated to each module varies. This allocation is for instance related to the module occupancy for a TC selection based on a threshold, as shown in Fig. 3.6 in Chapter 3. In addition even if the number of input links is the same for all the FPGAs the total average data throughput is not necessarily the same. FE modules are packed differently into lpGBT links in different regions of the detector, and the packing efficiency in lpGBT links varies across the detector. As a consequence there are unavoidable variations in the quantity of data processed by Stage 1 FPGAs as well as in the size of the physical regions covered by these data. Small variations from one FPGA to an other can be absorbed, but these variations must be as small as possible in order to minimize the hardware resource usage in the different FPGAs and the overall latency of the system.

In addition, since the Stage 2 needs to correlate data from the Stage 1, the data from different Stage 1 FPGAs must be sent in a uniform and consistent fashion. As an example, if the Stage 1 sends TCs to the Stage 2, TCs from a given projective region (for instance an $r/z - \phi$ window) must be sent at the same time from the different Stage 1 FPGAs, such that the Stage 2 can build 3D clusters with minimal latency and minimal resources. This requirement also impacts how the FE links can be mapped to the Stage 1 FPGAs. In the example above, the modules from different projective regions should be uniformly distributed across the different Stage 1 FPGAs. Therefore the distribution of optical links from the FE to the BE boards requires a careful optimization in terms of uniformity of the data size and physical coverage of these data. An illustration of the connections between the FE and the BE Stage 1, as optimized for the current architecture based on longitudinal seeding and clustering, is provided in Fig. 4.1. This optimization is based on a Stochastic Hill Climbing algorithm [33], uniformizing the average number of input TCs in each projective region across all Stage 1 FPGAs.

An additional architectural feature, required to concentrate at least 60° sectors in depth in single Stage 2 FPGA, is the implementation of time multiplexing between the Stage 1 and the Stage 2. The implementation of time multiplexing is common to all the versions of the HGCAL TPG system proposed so far. The concept of time multiplexing trigger (TMT) has been first used in the Phase 1 upgrade of the calorimeter trigger [4]. With a

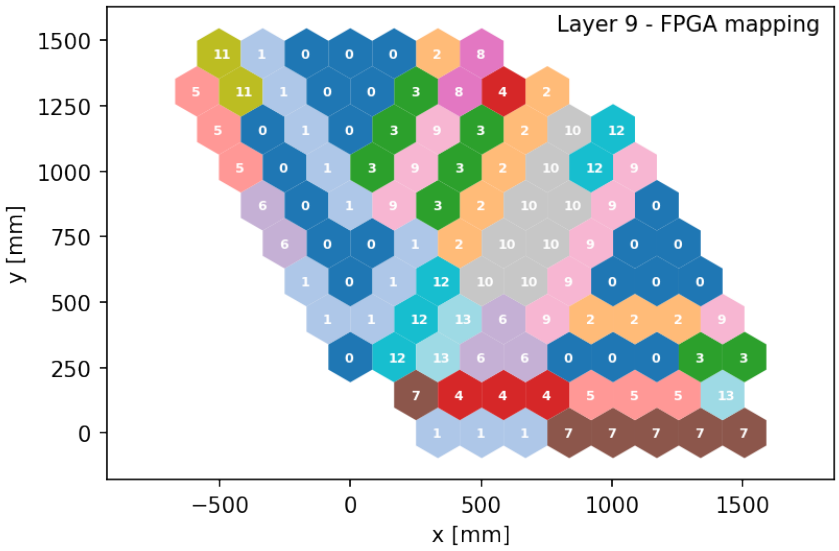

Figure 4.1: *Mapping of the connections between detector modules and Stage 1 FPGAs in one* 120° *sector of layer 9. In each hexagonal module is indicated the index of the Stage 1 FPGA receiving the trigger data from this module. There are 14 FPGAs in total for each* 120° *sector. This mapping is optimized using a Stochastic Hill Climbing algorithm [33], uniformizing the average number of input TCs in each projective region across all Stage 1 FPGAs.*

TMT architecture, data of a given bunch crossing can be sent over a period of time larger than 25 ns. More data from each bunch crossing can therefore be sent, which enables the concentration of larger detector regions into single FPGAs. Obviously since more latency is needed to send the data of one bunch crossing and new data from consecutive bunch crossings arrive every 25 ns, several copies of the receiver stage are needed in order to absorb the flux of data. A key quantity of a TMT is therefore its time multiplexing period, which is expressed in number of bunch crossings. The TM period tells how many LHC clock periods are needed to send the data from one bunch crossing. The number of duplicated FPGAs in this system is therefore equal to the TM period. The concept of TMT, as originally used in the Phase 1 upgrade, is illustrated in Fig 4.2.

2 2D and 3D clustering on two stages

Even though there are general architectural constraints coming from the detector geometry and from the input data flow, as listed above, there is still some flexibility in the choice of clustering algorithm and in the way it is distributed across the system. And the distribution of the algorithm steps on the different processing nodes is a critical question when designing trigger algorithms.

Distributing as much as possible different algorithm steps on the different processing stages can ensure a better utilization of the overall hardware resources. But it also reduces the flexibility in the design of the algorithm since it adds hardware frontiers between algorithmic steps. In particular, having a projective view of the detector is not possible in the first BE processing stage. Therefore, having a first reconstruction step at this stage is only compatible with some specific reconstruction algorithms for which the first steps don't

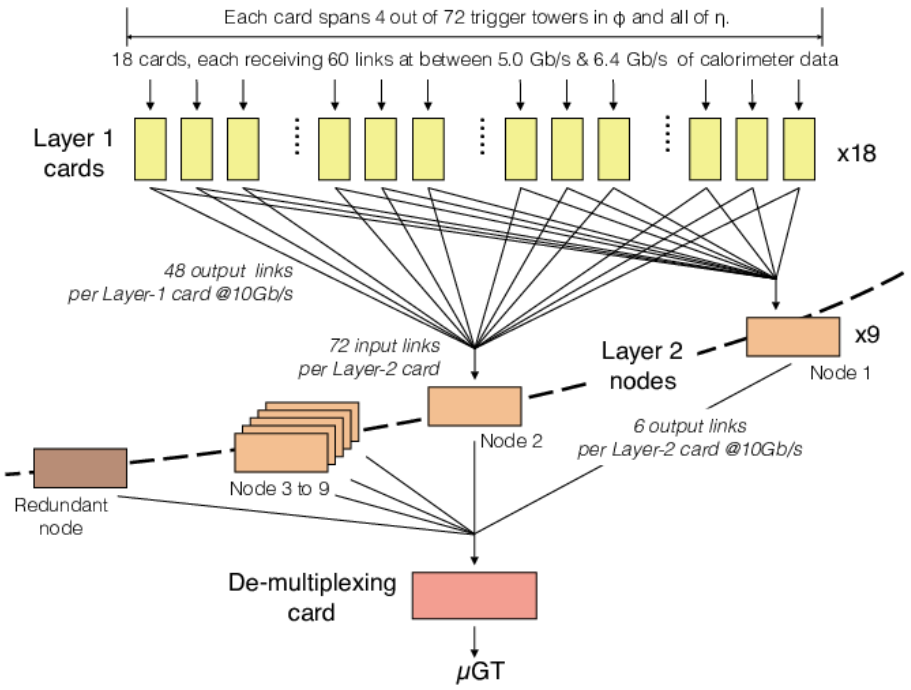

Figure 4.2: *Time multiplexing trigger (TMT) concept as implemented in the Phase 1 calorimeter trigger system [4]. Each Layer 1 board covers a small region of the calorimeter. Each Layer 2 board covers the full detector, receives data from all the Layer 1 boards and processes one every nine bunch crossings (the TM period is 9). A de-multiplexing board regroups the data from different bunch crossings into a single location.*

require a projective view of the detector in depth. The 2D and 3D clustering algorithm described in Section 4 of Chapter 2 is one example of algorithm that can naturally be distributed into two processing stages. 2D clusters are reconstructed in the Stage 1 and the Stage 2 clusters them together to form 3D clusters. The 2D clustering is performed independently in each layer and doesn't require a projective view in depth. The architecture that was designed to implement such algorithm, and described in the HGCAL TDR, is shown in Fig. 4.3. Stage 1 FPGAs can cover entire single layers, or half layers in the CE-E, such that minimal data duplication is needed. In that case, the number of 2D clusters reconstructed per event is sufficiently small such that the 2D clusters from one full endcap can be sent to a single Stage 2 FPGA, with a TM period of 24.

As mentioned already the main advantage of such architecture is the distribution on many FPGAs of the clustering steps. But in that case, due to bandwidth limitations between the two processing stages, the 3D clusters are based on compressed information resulting from the 2D layer clustering. In particular, the granular information of single TCs aggregated into 2D clusters is not retained and only summary quantities are propagated. It is in particular the case for cluster shape quantities, which are the main ingredients of cluster identification. As a consequence, 3D cluster quantities are necessarily computed in two steps, first layer by layer and then combining the layer-based quantities. For quantities using second order moments of the energy distribution within the cluster, like cluster widths, a loss of information necessarily arises from this two-step calculation. This loss of information results in a reduction of the discriminating power of these variables between

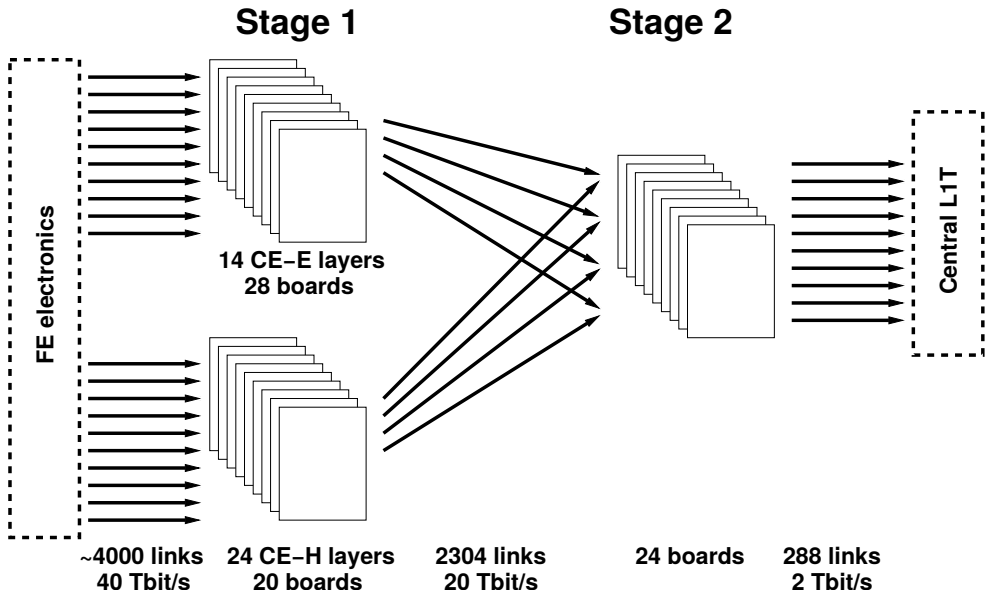

Figure 4.3: *BE components and architecture required for the implementation of a 2D+3D clustering algorithm [6]. The components listed cover one endcap and the other endcap components are identical in structure. FPGAs in the first stage cover single or two layers in the CE-H and half layers in the CE-E. The second stage FPGAs cover one full endcap. A time multiplexing period of 24 is required to concentrate the full 2D cluster data into single Stage 2 FPGAs.*

electromagnetic and hadronic showers.

Additional issues arise when implementing in firmware the 2D and 3D clustering algorithm described in Chapter 2. The first issue comes from neighbor finding in irregular geometries. Although a non-orthogonal coordinate system can be defined to navigate between TCs in the silicon section, as depicted in Fig. 4.4, the same coordinates cannot be used in the scintillator section, which follows an $(\eta - \phi)$-like geometry. And no simple coordinate system can accommodate the transition region between the silicon and the scintillator modules. In order to cover mixed silicon-scintillator CE-H layers, the transition region would need to be handled differently with TC neighbor information encoded into lookup tables. Such special cases would necessarily require additional hardware resources. These limitations could nevertheless be bypassed by forming 2D clusters independently in the silicon and in the scintillator sections. These clusters would then be regrouped together in the 3D step.

An other, more impactful, issue comes from the formation of 3D clusters from 2D clusters. In the case of a distance-based 3D clustering as described in Section 4 of Chapter 2, any 2D cluster can seed a 3D cluster or can be aggregated into a pre-existing 3D cluster. The complexity of such algorithm is typically of $\mathcal{O}(N^2)$. It can be reduced with a proper indexing, based for instance on KD-trees, but building an index is also very resource consuming. Density-based algorithms such as DBSCAN are even more demanding in terms of logic resources. It wouldn't be a problem with a low multiplicity of 2D clusters. But up to three hundred 2D clusters can typically be reconstructed per layer in the CE-E, as shown in Fig. 4.5. Attempts have been made to implement such algorithms but the logic resources required to handle such high multiplicity were found to be out of reach with existing FPGAs.

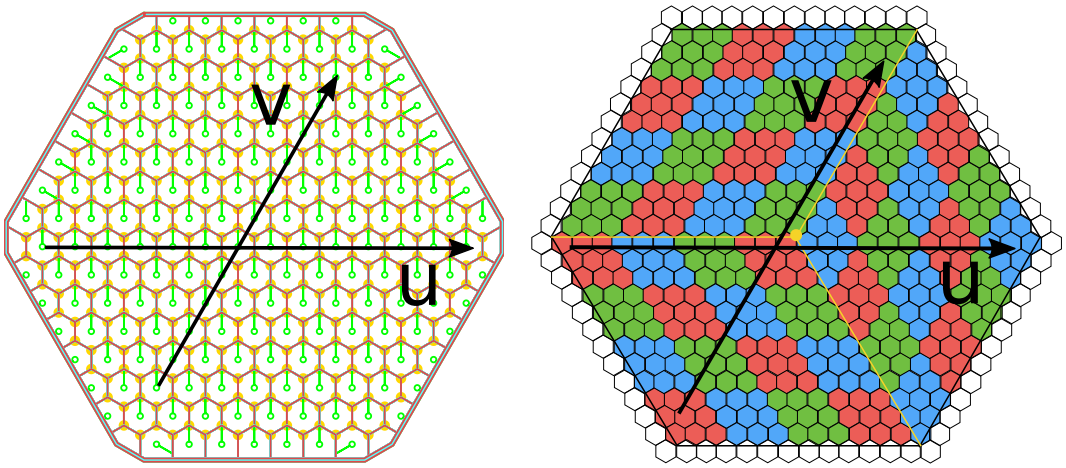

Figure 4.4: *Non-orthogonal coordinate system used to navigate between hexagonal cells and between trigger cells. Due to the* 120° *rotations of trigger cells within modules, their positions don't follow strictly the u and v axes, but these coordinates can still be used to identify unique trigger cells.*

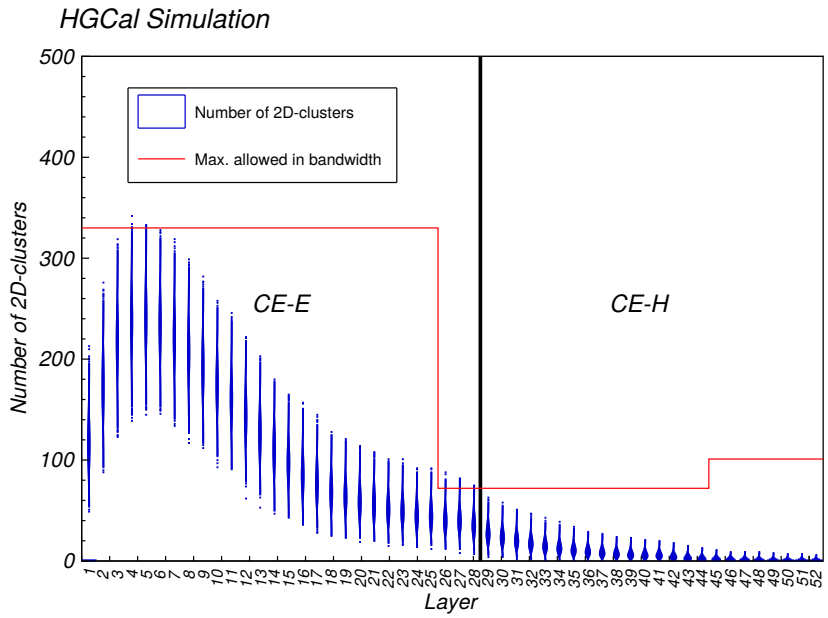

Figure 4.5: *Multiplicity of 2D clusters reconstructed per endcap as a function of the layer, in $t\bar{t}$ events with 200 PU [6]. The red line shows the bandwidth limit coming from the allocated number of links between the Stage 1 and the Stage 2*

3 Direct 3D clustering on a single stage

The last issue mentioned in the previous section has triggered the development of the 3D clustering with longitudinal seeding described in Section 5 of Chapter 2. The key ingredient for hardware implementation is the construction of a projective 2D histogram for seeding and TC indexing. The seed search is facilitated in such 2D histogram and the multiplicity of seeds is reduced thanks to the energy integration over the full detector depth. Additionally,

the packing of TCs into histogram bins provides an easy way to geometrically index TCs for clustering around the identified seeds. The $(r/z, \phi)$ space has been chosen to implement the histogram. Using the ϕ coordinate allows to better match the firmware structure with the division of the system into ϕ sectors. And r/z is used instead of η in order to better correspond to the physical coverage of showers.

Although this 3D clustering could have been used to cluster 2D clusters and simply replace the 3D step, it has been chosen to apply it directly on TCs. As explained in the previous section, direct 3D clustering improves the computation of cluster shape variables used for shower identification. The main drawback of this choice is that all the algorithmic steps are performed in a single processing stage, which means more limited FPGA resources. Nevertheless this is alleviated with the implementation of one part of the seeding step in the Stage 1. The histogram-based seeding first needs TCs to be distributed into histogram bins. Given the irregular geometry of the detector and the fact that the histogram space $(r/z, \phi)$ is different from the TC space (x, y), this distribution of TCs to histogram bins is complex and cannot be done with simple calculations. Large lookup tables are therefore required to encode this mapping. The distribution of TCs along the first dimension of the histogram r/z is done in the Stage 1, such that only the distribution along the second dimension ϕ needs to be done in the Stage 2. After having distributed TCs into r/z rows, the Stage 1 sends TCs row after row to the Stage 2, which can then process them in a pipelined fashion.

An additional drawback of such architecture is that the complete TC data of a full endcap cannot be concentrated in single FPGAs with a reasonable time multiplexing period. Therefore, Stage 2 FPGAs cover only a fraction of one endcap. The minimal sector size required to build 3D clusters and to benefit from the symmetries of the detector is 60°, as mentioned in Section 1. In the current design 120° sectors have been chosen, with a TM period of 18 between the Stage 1 and the Stage 2, as shown in Fig. 4.6. Such larger sector coverage compared to the minimum is beneficial to avoid wasting too much bandwidth with duplicated data.

Nevertheless the bandwidth available between the two stages is still limited and only a fraction of the input TCs can be sent to the Stage 2. Therefore, in addition to the packing of TCs into r/z rows, a truncation is performed for each of the row packets, with a fixed number of TCs selected in each of these packets. This is currently done by sorting TCs by energy and selecting the highest-energy ones in each of the packets. Such truncation can obviously impact the cluster energy reconstruction in a negative way. But a maximum of 420 TCs can be sent from one Stage 1 FPGA to the Stage 2, while on average about 300 TCs above threshold arrive at the input. Given this 40% margin above the average and the fact that only the lowest energy TCs are truncated, it ensures minimal degradation of the performance. This truncation can be optimized by tuning its profile as a function of r/z. 420 TCs can be sent in total but the way these 420 TCs are distributed among all the r/z rows is flexible. Typically three types of truncation profiles can be used, shown in Fig. 4.7.

- A profile that follows the input distribution of TCs, driven by PU, with more input TCs in the low r/z region (high η).
- A profile that follows the detector area covered by the r/z rows, which favors instead the high r/z region.
- A uniform profile where the same number of TCs is sent for all the r/z rows.

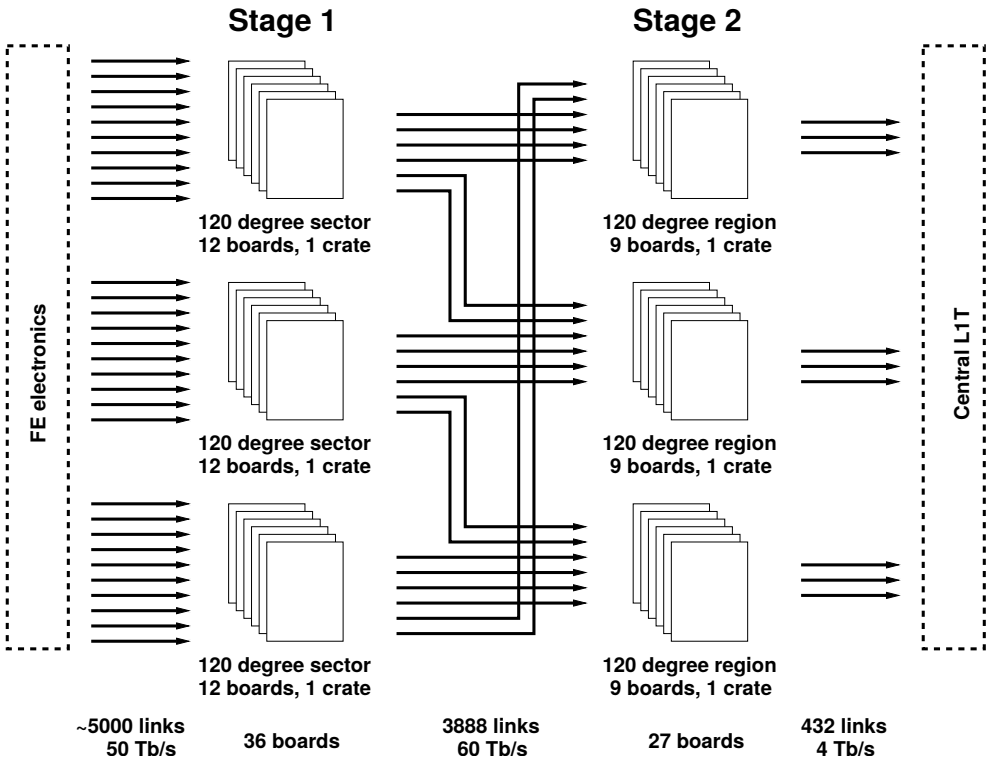

Figure 4.6: *BE components and architecture developed for the implementation of a 3D seeding and clustering algorithm [7]. The components listed here cover one endcap and the other endcap components are identical in structure. FPGAs in the first stage cover multiple detector modules in a* 120° *sector as illustrated in Fig. 4.1. The second stage FPGAs cover full* 180° *sectors in depth (*120°+60° *of duplicated data). A time multiplexing period of 18 is used.*

None of the options degrade the cluster reconstruction, and in particular the reconstruction of electromagnetic clusters, as can be seen in Fig. 4.8 where the energy resolution of electron clusters is compared for different truncation profiles and without any truncation. A tighter truncation in the high-η region can even help reducing PU contaminations, although marginally.

In the end, the implementation of a direct 3D clustering with longitudinal seeding requires to organize the cluster reconstruction by 120° sectors and to truncate further TC data in the Stage 1. But the corresponding firmware has been shown to be implementable within the existing hardware and the truncation is not deteriorating the reconstruction performance.

4 Conclusion

Although all the studied algorithms have been designed with an hardware architecture in mind, it is only when they are actually implemented in firmware that their feasibility can really be assessed. At the time of writing, the key firmware pieces of the longitudinal seeding and direct 3D clustering are implemented [34]. These are mainly located in the Stage 2, with one step in the Stage 1: the allocation of inputs TCs to r/z seeding rows. It is showing that such algorithm will be implementable with the targeted FPGAs (Xilinx VU13P) and hardware architecture.

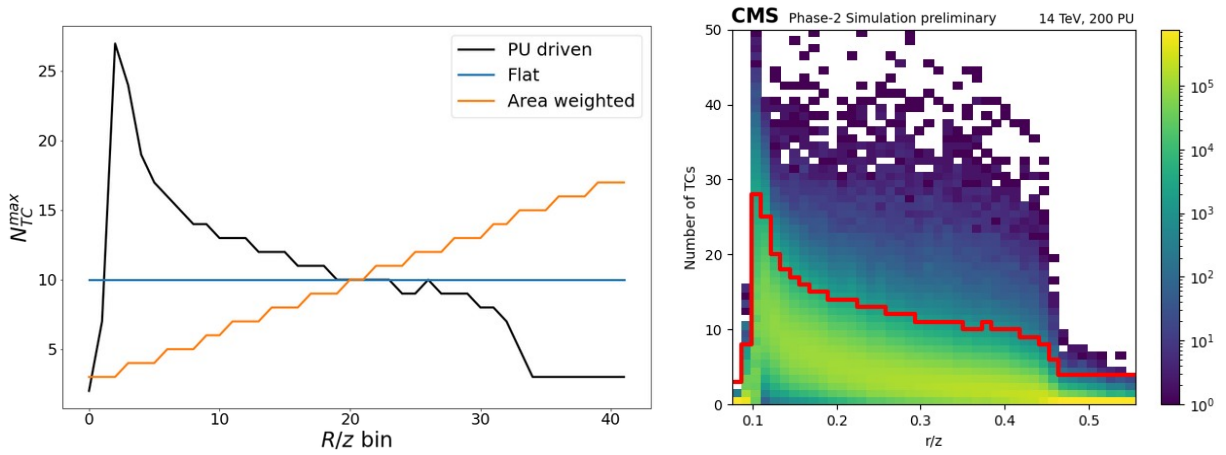

Figure 4.7: *Examples of possible profiles of the maximum number of TCs selected per r/z bin in a Stage 1 FPGA (left) [12]. The bandwidth allocated for each bin can be flat, favor the low-η region (area weighted) or favor the high-η region (PU driven). The PU driven allocation follows the distribution of the number of TCs per bin as produced by 200 PU events (right).*

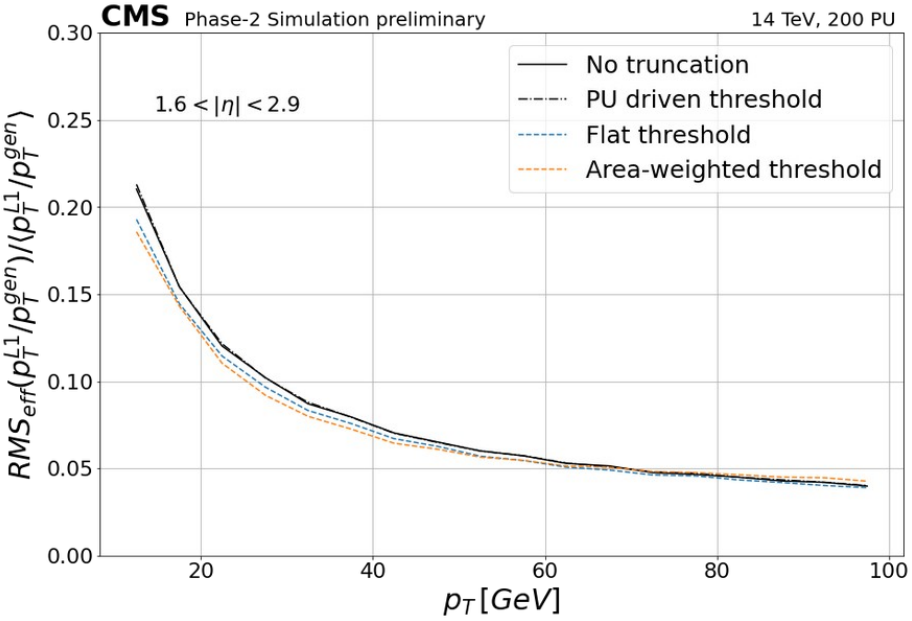

Figure 4.8: *Energy resolution of electrons in particle gun events simulated with 200 PU, as a function of the generated p_T [12]. The energy resolution is measured with the effective RMS of the p_T^{L1}/p_T^{gen} distribution, divided by the mean of the distribution. The resolution curves are obtained using the three different truncation profiles of Fig. 4.7 as well as without any truncation in the Stage 1.*

It is nevertheless obvious that algorithmic and firmware developments won't stop. Although the BE hardware needed to be defined now, together with a workable system, algorithms will likely be improved and eventually totally transformed in the coming years before the HL-LHC starts.

General conclusion

The developments of the HGCAL TPG system started in 2014, before the Run 2 of the LHC. Most of the ideas at that time were inspired by the Phase 1 calorimeter trigger upgrade, trying to push things several steps further. The Phase 1 system upgrade was already at the edge of the technology and introduced for the first time key new concepts like *time multiplexing trigger* and *dynamic clustering* at the L1 trigger. The success of the Run 2 prove that these developments were reliable and effective, and the HGCAL TPG has been further built upon this success. New ideas have then been developed and the HGCAL TPG system has been further consolidated until its current status. It is now entering maturity towards its construction and operation. The entire set of processing and reconstruction steps in the frontend and in the backend have been implemented. The main uncertainty remaining is the ability to efficiently unpack, in the Stage 1 FPGAs, the data coming from the frontend electronics, which is directly linked to the TC data reduction strategy applied in the ECON-T.

In the coming months and years the different pieces of the system will progressively be assembled and tested together in larger and larger test stands. The firmware blocks currently designed will be optimized further in order to benefit as much as possible from the available hardware. And the software emulation of the system, which is being developed in parallel with the firwmare, will be used for validation and for the tuning of parameters.

Of course, in parallel with these critical tasks targeting the construction of a full-scale system, new algorithm ideas will emerge. And although the definition of the hardware is now pretty well settled, the algorithms implemented within this hardware will likely evolve further in the coming years. Among algorithm evolutions it is foreseen that machine learning will play an increasingly important role. Future triggers will most likely be hybrid systems containing both human-designed and machine-learnt algorithms. And the main difficulty, as always, will be the mapping of these algorithms onto hardware.

Bibliography

[1] J.-B. Sauvan, *Performance and upgrade of the CMS electron and photon trigger for Run 2*, J. Phys. Conf. Ser. **587** (2015) no. 1, 012021.

[2] L. Mastrolorenzo, *The CMS Level-1 Tau identification algorithm for the LHC Run II*, Nuclear and Particle Physics Proceedings **273-275** (2016) 2518–2520.

[3] A. Zabi, *Opérations et optimisations du système de déclenchement calorimétrique de l'expérience CMS au LHC*. Habilitation à diriger des recherches, Laboratoire Leprince Ringuet Ecole Polytechnique, 2016. `https://hal.archives-ouvertes.fr/tel-03030251`.

[4] CMS Collaboration, *Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV*, JINST **15** (2020) no. 10, P10017, `arXiv:2006.10165 [hep-ex]`.

[5] CMS Collaboration, *Technical Proposal for the Phase-II Upgrade of the CMS Detector*, CMS-TDR-15-02, CERN, 2015.

[6] CMS Collaboration, *The Phase-2 Upgrade of the CMS Endcap Calorimeter*, CMS-TDR-019, CERN, 2018.

[7] CMS Collaboration, *The Phase-2 Upgrade of the CMS Level-1 Trigger*, CMS-TDR-021, CERN, 2020. `https://cds.cern.ch/record/2714892`.

[8] F. Bouyjou et al., *HGCROC3: the front-end readout ASIC for the CMS High Granularity Calorimeter*, JINST **17** (2022) no. 03, C03015.

[9] lpGBT Team, *The lpGBTv1 manual*. CERN. `https://lpgbt.web.cern.ch/lpgbt/`.

[10] J. Troska et al., *The VTRx+, an optical link module for data transmission at HL-LHC*, PoS **TWEPP-17** (2017) 048.

[11] N. Strobbe, *Readout electronics for the CMS Phase II Endcap Calorimeter: system overview and prototyping experience*, JINST **17** (2022) no. 04, C04023.

[12] L. Portales, *Level-1 Triggering on High-Granularity Calorimeter information at the HL-LHC* , `https://indico.cern.ch/event/847884/contributions/4833233/`, 2022. CALOR 2022 - 19th International Conference on Calorimetry in Particle Physics.

[13] GEANT4 Collaboration, S. Agostinelli et al., *GEANT4—a simulation toolkit*, Nucl. Instrum. Meth. A **506** (2003) 250.

[14] J.-B. Sauvan et al., *EC review - Backend electronics and Level-1 Trigger*, https://indico.cern.ch/event/354811/#4-backend-electronics-and-leve, 2014. CMS internal document (restricted).

[15] S. M. Lundberg and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, p. 4768–4777. 2017. https://dl.acm.org/doi/10.5555/3295222.3295230.

[16] A. Hakimi and J.-B. Sauvan, *Multi-objective optimization for the CMS High Granularity Calorimeter Level 1 trigger*, https://indico.cern.ch/event/1078970/contributions/4833328/, 2022. 5th Inter-experiment Machine Learning Workshop.

[17] T. Strebler, *Design and performance of the CMS High Granularity Calorimeter Level 1 trigger*, J. Phys. Conf. Ser. **1162** (2019) no. 1, 012019.

[18] T. Strebler, *Design and object performance of the CMS High Granularity Calorimeter Level 1 trigger* , https://indico.cern.ch/event/642256/contributions/2961908/, 2018. CALOR 2018 - 18th International Conference on Calorimetry in Particle Physics.

[19] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., *A density-based algorithm for discovering clusters in large spatial databases with noise.*, in *kdd*, vol. 96, pp. 226–231. 1996.

[20] M. Prvan, *Algorithms for the Level-1 trigger with the HGCAL calorimeter for the CMS HL-LHC upgrade*. PhD thesis, Institut Polytechnique de Paris, 2020.

[21] J. Duarte et al., *Fast inference of deep neural networks in FPGAs for particle physics*, Journal of Instrumentation **13** (2018) no. 07, P07027–P07027.

[22] S. Summers et al., *Fast inference of Boosted Decision Trees in FPGAs for particle physics*, Journal of Instrumentation **15** (2020) no. 05, P05026–P05026.

[23] S. Bhattacharya et al., *GNN-based end-to-end reconstruction in the CMS Phase 2 High-Granularity Calorimeter*, 2022. arXiv:2203.01189 [physics.ins-det].

[24] S. R. Qasim et al., *End-to-end multi-particle reconstruction in high occupancy imaging calorimeters with graph neural networks*, arXiv:2204.01681 [physics.ins-det].

[25] S. R. Qasim, J. Kieseler, Y. Iiyama, and M. Pierini, *Learning representations of irregular particle-detector geometry with distance-weighted graph networks*, Eur. Phys. J. C **79** (2019) no. 7, 608, arXiv:1902.07987 [physics.data-an].

[26] X. Ju et al., *Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors*, in *33rd Annual Conference on Neural Information Processing Systems*. 3, 2020. arXiv:2003.11603 [physics.ins-det].

[27] Y. Iiyama et al., *Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics*, Front. Big Data **3** (2020) 598927, arXiv:2008.03601 [physics.ins-det].

[28] P. Aspell et al., *HGCROC3: Working document on specification*, https://edms.cern.ch/file/2324379/1/HGCROC3_Spec_Working_Document_v2.0.pdf, 2021. CMS internal document (restricted).

[29] J. Hirschauer and R. Wickwire, *Trigger Concentrator ASIC (ECON-T) for the CMS HL-LHC endcap calorimeter*, https://edms.cern.ch/file/2206779/1/ECON-T_specification_working_doc_v9_2mar2022.pdf, 2022. CMS internal document (restricted).

[30] K. E. Batcher, *Sorting Networks and Their Applications*, in *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, AFIPS '68 (Spring), p. 307–314. 1968.

[31] C. Martin Perez, *Search for the Higgs Boson Produced in Association with Top Quarks with the CMS Detector at the LHC*. PhD thesis, Institut Polytechnique de Paris, 2020.

[32] C. Martin Perez, J.-B. Sauvan, and S. Webb, *Summary of frontend studies*, https://indico.cern.ch/event/879499/#17-summary-of-fe-studies, 2020. CMS internal document (restricted).

[33] G. Hayes, *mlrose: Machine Learning, Randomized Optimization and Search package for Python*, https://github.com/gkhayes/mlrose, 2019.

[34] P. Dauncey et al., *The HGCAL Backend TDAQ Systems for the Pre-series Design Review*, https://edms.cern.ch/file/2220310/2/DN-19-032-V2.pdf, 2022. CMS internal document (restricted).