DEFINITIONS OF ACRONYMS

ATLAS A Toroidal LHC Apparatus . . . 5
BCID Bunch Crossing ID . . . 19
CP Cluster Processor . . . 6
CPM Cluster-processor Module . . . 17
eFEX Electromagnetic Feature Extractor . . . 6
EM Electromagnetic . . . 24
FEX Feature Extractor . . . 6
FPGA Field-programmable Gate Array . . . 11
GEP Global Event Processor . . . 9
gFEX Global Feature Extractor . . . 6
HL-LHC High-Luminosity LHC . . . 5
HLT High-level Trigger . . . 24
JEP Jet/Energy Processor . . . 6
jFEX Jet Feature Extractor . . . 6
L1Calo Level-1 Calorimeter . . . 5
LAr Liquid Argon . . . 6
LATOME Liquid Argon Trigger Optical Mezzanine . . . 19
LHC Large Hadron Collider . . . 5
RoI Region of Interest . . . 6
TOB Trigger Object . . . 6
UI User Interface . . . 10

CHAPTER 5

Upgrading the ATLAS Level-1 Calorimeter Trigger

5.1 Evolution of the Level-1 Calorimeter Trigger

The Level-1 Calorimeter (L1Calo) system, as used for Run 2 of the Large Hadron Collider (LHC) in the A Toroidal LHC Apparatus (ATLAS) detector, is described in Section X. This chapter focuses on work done towards upgrading this system for later LHC runs. In the long shutdown between Run 2 and Run 3, significant improvements were made to the system as part of the Phase-I upgrade. The long shutdown following Run 3 will facilitate further changes with the Phase-II upgrade, preparing L1Calo for the High-Luminosity LHC (HL-LHC) in Run 4.

Sections 5.2 and 5.3 discuss work on the Phase-I upgrade and Section 5.4 discusses work on the Phase-II upgrade, all with a focus on triggers for e/γ signatures. Details of the L1Calo system as implemented/planned for Phase I and Phase II, with the major changes in comparison to the Run-2 system and to each other, are given in

Sections 5.1.1 and 5.1.2 respectively.

5.1.1 Phase-I upgrade

The Phase-I upgrade has progressed alongside the work discussed in this chapter, to the point where the system is fully implemented and in use in Run 3 at the time of writing. The goal of the Phase-I upgrade to L1Calo is a redesign of the core components of the trigger in order to handle more data and make more refined decisions while rejecting events.

The main components introduced in the Phase-I upgrade are the Feature Extractor (FEX) systems, which replace the Cluster Processor (CP) and Jet/Energy Processor (JEP) from the Run-2 system. The Electromagnetic Feature Extractor (eFEX) provides discrimination for e/γ and τ objects, the Jet Feature Extractor (jFEX) focuses on jets whilst providing additional τ identification, and the Global Feature Extractor (gFEX) triggers on large-radius jets and global quantities such as missing energy.

An overview of the Phase-I system architecture is shown in Figure 5.1. The key change to the e/γ signature trigger is that it now receives digital information from the Liquid Argon (LAr) Calorimeter in the form of SuperCells, rather than the analogue tower energies that were available to the Run-2 system. This information is processed by the eFEX to generate Trigger Objects (TOBs), equivalent to the Region of Interests (RoIs) generated by the CP in Run 2.

The move to digital input comes with an increase in granularity, a trigger tower now being split in both η and calorimeter layer to give up to 10 SuperCells: typically one SuperCell each from Layers 0 (presampler) and 3 and four SuperCells each from Layers 1 and 2, segmented in η. Each SuperCell is formed by summing energies from between four and eight calorimeter cells. This division of a tower into SuperCells is shown in Figure 5.2. The granularity received from the Tile Calorimeter is the same as in Run 2, the energy in a tower (this is later referred to as Layer 4).

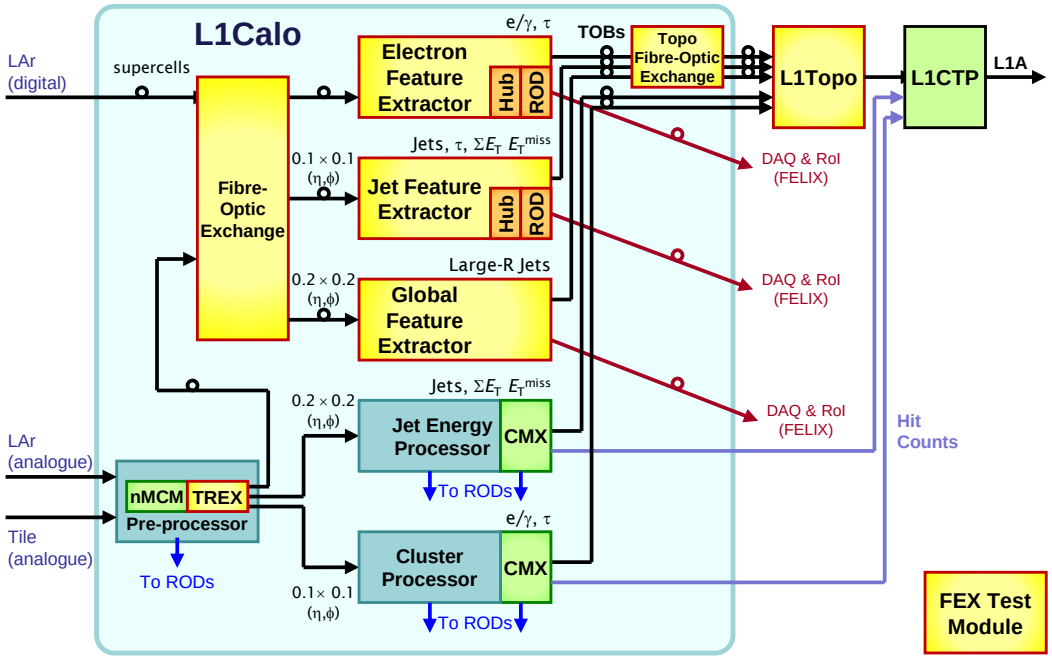


Figure 5.1: Diagram showing the L1Calo modules in use for Run 3 of the LHC. Gold rectangles represent modules introduced in the Phase-I upgrade. Blue and green rectangles represent existing components from the Run 2 system, included still as part of the trigger whilst transitioning to the new system. [1]

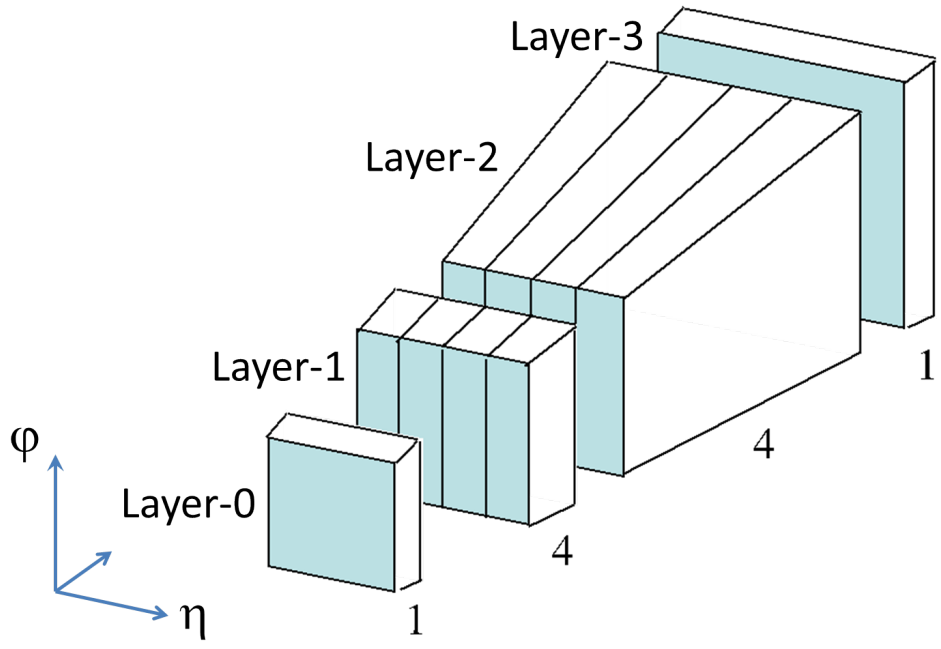


Figure 5.2: Diagram showing division of a single trigger tower into SuperCells. [2]

The eFEX introduces new algorithms to use the SuperCell information in order to trigger on e/γ objects. In comparison to the CP in Run 2 which calculates a cluster energy and a single isolation value, the eFEX calculates three separate variables used to identify and isolate a candidate object and can also more accurately calculate the energy from SuperCell information.

To calculate any of these variables a seed SuperCell is first identified, as the highest energy SuperCell in Layer 2 of the calorimeter compared to the surrounding region. The energy of the cluster is calculated by summing the energy of the seed with its highest energy neighbour in ϕ and both neighbours in η, adding also the six corresponding SuperCells in Layer 1, and the two SuperCells from Layers 0 and 3 that are in the same tower as the seed. The three other discriminating variables are calculated as follows:

$$R_{\eta} = \frac{\text{energy in } 3 \times 2 \text{ area of cells}}{\text{energy in } 7 \times 3 \text{ area of cells}},$$

with each area (in $\eta \times \phi$) centred on the seed and calculated in Layer 2;

$$R_{\text{had}} = \frac{\text{core energy}}{\text{environment energy}},$$

where the core energy is calculated in the same manner as for the cluster energy but including both neighbours in ϕ (so a 3×3 area of SuperCells in Layers 1 and 2 and a 1×3 area in Layers 0 and 3) and the environment energy is the energy in a 3×3 tower area in Layer 4 (i.e. the hadronic calorimeter); and

$$w^2_{s,tot} = \frac{\sum_i i^2 \cdot E_i}{\sum_i E_i},$$

where $i \in [-2, 2]$ is the η coordinate of the SuperCell relative to the seed, calculated for SuperCells in Layer 1 with both neighbours in ϕ also summed for each E_i. Visual representations of the areas included for these algorithms are presented alongside the work done in Section 5.2.

Candidate τ particles considered by the eFEX have a similar set of variables (cluster

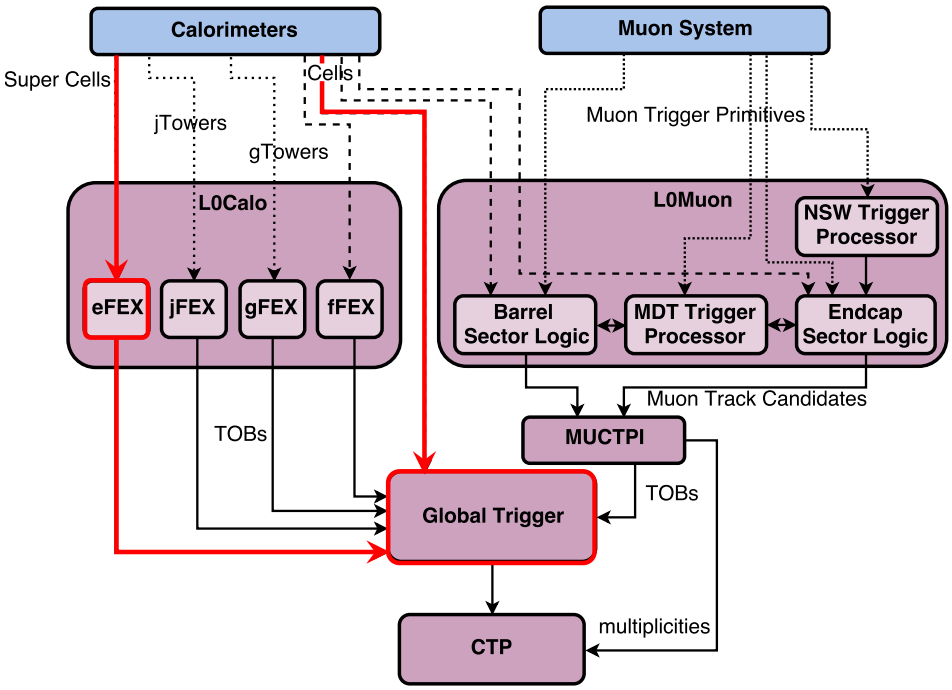


Figure 5.3: Schematic of the ATLAS hardware trigger as planned for the Phase-II upgrade in Run 4 of the LHC. The red lines highlight the main parts relevant to the e/γ trigger, with the addition of the Global Trigger being the main change with respect to the Phase-I system. [3]

energy, R_η, and R_{had}) with small differences in the areas used in their calculation.

5.1.2 Phase-II upgrade

As luminosity and pileup is increased even further with the high-luminosity era of the LHC in Run 4, the trigger needs to be further improved to operate in increasingly difficult conditions. The Phase-II upgrade to the hardware trigger aims to do this primarily by adding a new component, the Global Event Processor (GEP) (or Global Trigger). The GEP will be downstream of the Phase-I FEX modules, which will continue to contribute to the trigger, and will refine decisions made by employing additional information: information from a larger area than typically available to a single FEX but also finer in granularity.

An outline of how the GEP fits in with the existing systems is shown in Figure 5.3. Information from the calorimeters will be sent directly to the GEP in finer granularity than is available to the eFEX, with energies in each individual cell without them being grouped into SuperCells. This gives a 4-8 times increase in granularity, depending on the region of the calorimeter.

The additional information available to the GEP means it can work together with the eFEX. The eFEX will create TOBs with associated variables (discussed in Section 5.1.1) which are sent to the GEP, which can then further probe the same region of the calorimeter to determine if the candidate object should be accepted. The algorithms used by the GEP to do this are the topic of the study in Section 5.4.

5.2 Visualisation of eFEX inputs and algorithms

In order to aid in debugging minor differences between different implementations of eFEX algorithms, an algorithm visualisation program was created. The program reads input data and performs aspects of eFEX algorithms whilst also providing a visual representation of what the algorithm is doing and where the result comes from.

The visualiser is written in JavaScript, using Node.js to interface with some server-side C++ scripts, needed to access energy decoders from online software, and Express.js to handle the web-based User Interface (UI).

5.2.1 Motivation

During development of algorithms for the hardware trigger, each algorithm is implemented multiple times. First, algorithms will be implemented in offline software to be tested and tuned against simulations or existing data. Then, in order to run on hardware, the algorithm needs to be ported to firmware. Often, to provide closer

cross-checks of the firmware algorithms, they are also simulated in online software. Inevitably, due to software and firmware bugs, subtle differences will exist between these algorithms; these differences need to be understood and corrected to have a complete bug-free implementation.

The need to find these subtle differences between algorithms motivates the visualisation software discussed here. Although at first it seems illogical to add an additional, independent, implementation of the algorithm (since the issue is in part due to having multiple different implementations), the added visualisation aspect makes it easier to understand where a particular algorithm implementation might have gone wrong in cases where there are discrepancies. This has been demonstrated through the use of the visualisation software in tests, discussed in Section 5.2.5.

5.2.2 Input Data

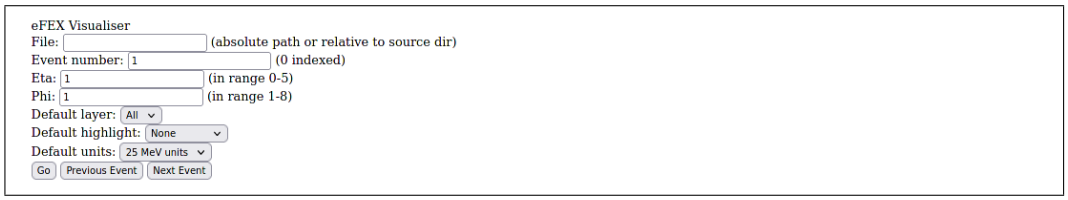
The visualisation software takes as input the calorimeter energies visible to a single eFEX Field-programmable Gate Array (FPGA). This covers a 6×10 area in $\eta \times \phi$ of trigger towers, with each tower being split into SuperCells across 5 calorimeter layers, as described in Section 5.1.1. For each event, the input data provides one energy value per SuperCell, encoded with either LAr or Tile energy encoding.

From this input data, a 3×3 area of trigger towers, centred on an (η, ϕ) coordinate provided by the user, is extracted and displayed on-screen. This area covers all energy values used for eFEX algorithms if the seed of the TOB is located in the central trigger tower.

5.2.3 User Interface

The eFEX Visualiser program provides a minimal UI to explore input data and results of the eFEX algorithms. The basic interface is shown in Figure 5.4. It prompts the user to specify an input file, (η, ϕ) centre-tower coordinates, and an

event number, then on receipt of these inputs it reads the information and displays the requested energies in a grid.



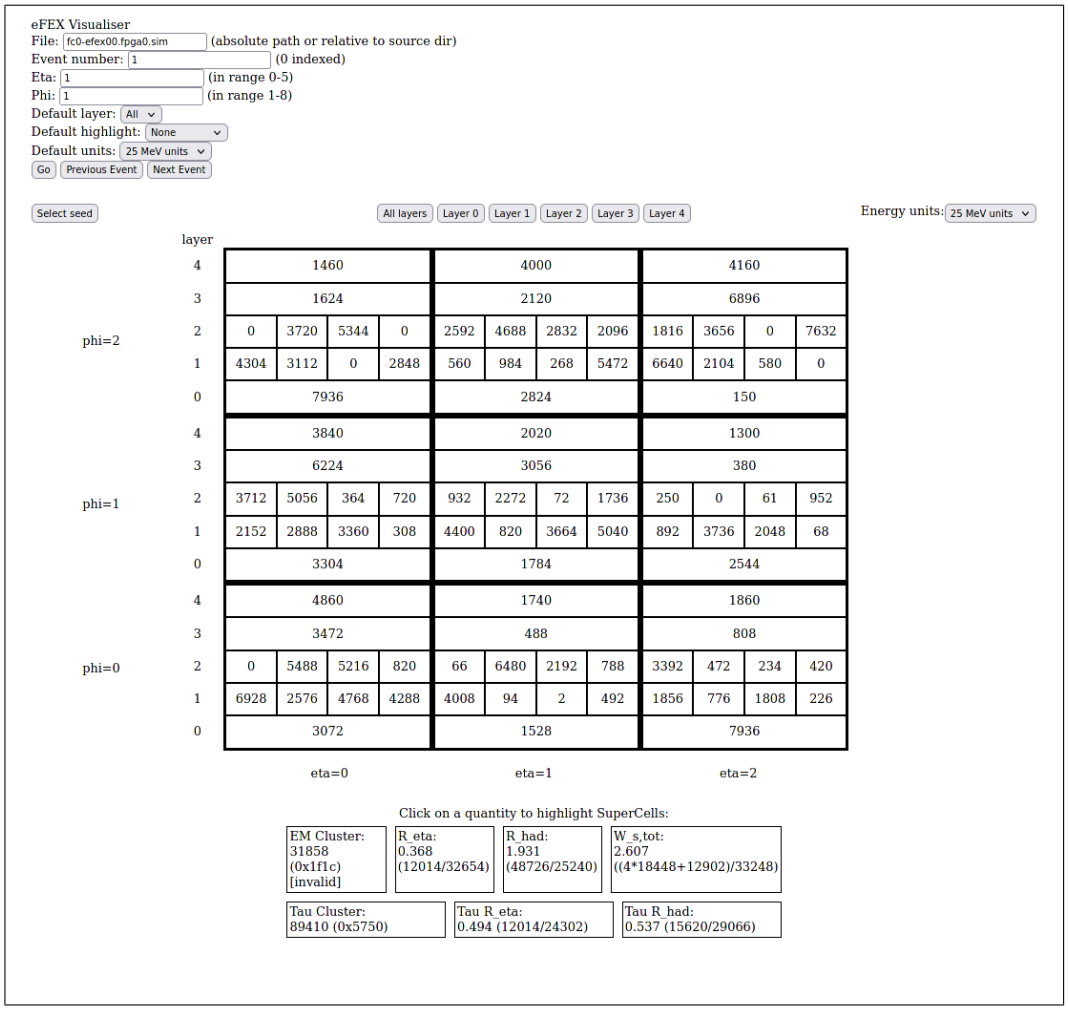


Figure 5.4: Initial interface on launching the eFEX Visualiser program (top) and the default view once data is read from a file (bottom).

The full interface becomes visible after the grid is displayed. The grid itself is a 3×3 area divided by bold lines, with each square representing a trigger tower, and each trigger tower square divided further into SuperCells. The horizontal axis represents the η coordinate of the tower or SuperCell, and the vertical axis represents the ϕ coordinate. These coordinates are labelled with the same indices the user gave as

initial input. In order to show all layers simultaneously, in the default view layers are stacked on top of each other, in the phi axis, within each tower. Controls are provided to instead view each layer individually if preferred.

Below the grid, a list of all the quantities calculated for the current TOB is displayed. Clicking on one of these quantities will highlight all of the SuperCells involved in the calculation. The details of how these values are calculated and how the algorithms are visualised are discussed in more detail in Section 5.2.4.

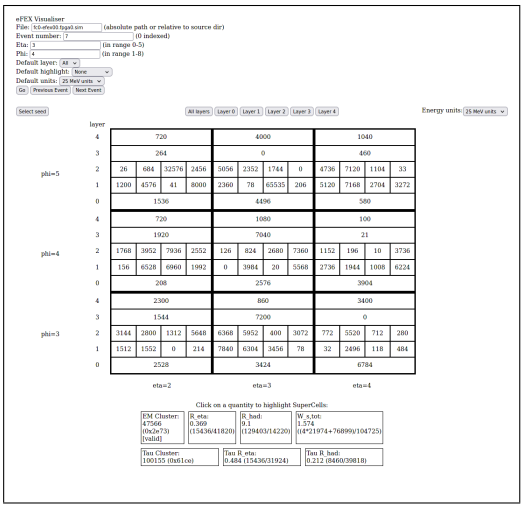
Above the grid, alongside the layer selection buttons, are options to manually set the seed SuperCell and to select the units used to display energies. The unit selection input is a drop-down box that allows the user to choose between 25 MeV (default units in firmware) or GeV units. Changing this option instantly updates all displayed energies. Pressing the "Select seed" button will toggle the layer view to display Layer 2, prompt the user to click on the SuperCell with the highest energy, and then on its ϕ-neighbour with the highest energy. This aids the user in selecting the correct seed for TOB generation, but is not normally necessary as the program will apply these criteria to automatically set the seed as soon as the grid is loaded. The manual override is included in case the automatic selection is wrong, or if looking at algorithms with a different seed may help debugging.

5.2.4 Algorithms

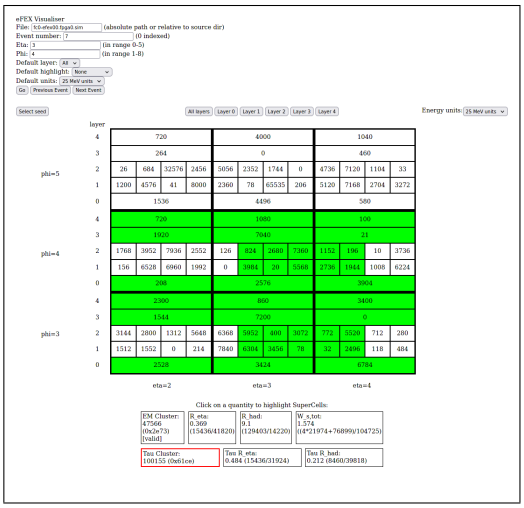
For each TOB processed (i.e. each particular event, coordinate location, and seed), several algorithms are run to calculate the quantities displayed on-screen. These are the same algorithms used by the eFEX to calculate TOB energies and isolations. The following variables are calculated: EM cluster energy, EM R_η, EM R_{had}, EM $w_{s,\text{tot}}$, tau cluster energy, tau R_η, and tau R_{had}. All of these are either sums of SuperCell energies (EM and tau cluster energies), ratios of sums of SuperCell energies ($R_\eta$and R_{had}), or a ratio with weighted sums ($w_{s,\text{tot}}$).

The values of these variables are calculated immediately after the data for a given

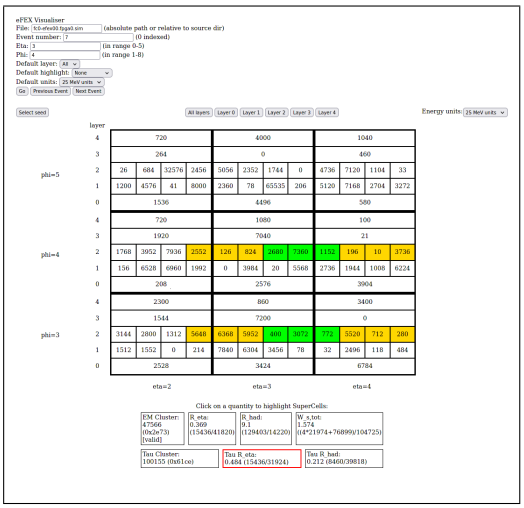
TOB is collected, or once the seed is re-specified, and displayed on-screen below the grid. If the user clicks on a displayed quantity, the SuperCells involved in the sums for the corresponding algorithm are highlighted with colours corresponding to whether those cells are used in the numerator (lime green); the denominator (gold); or, in the case of $w_{s,\mathrm{tot}}$, in the numerator with a larger weight (dark green). Figures 5.5 and 5.6 demonstrate the highlighting for all of the algorithms.



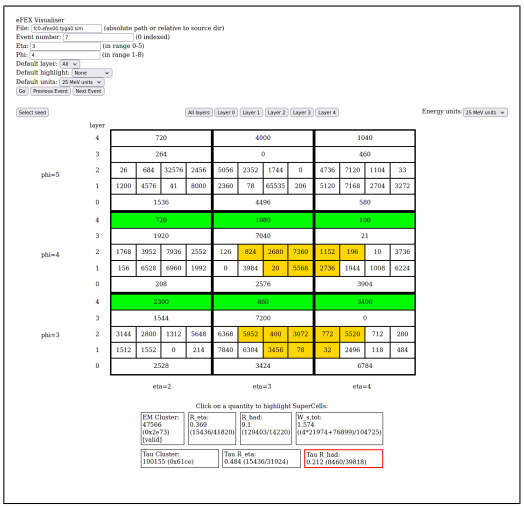
(a)



(b)

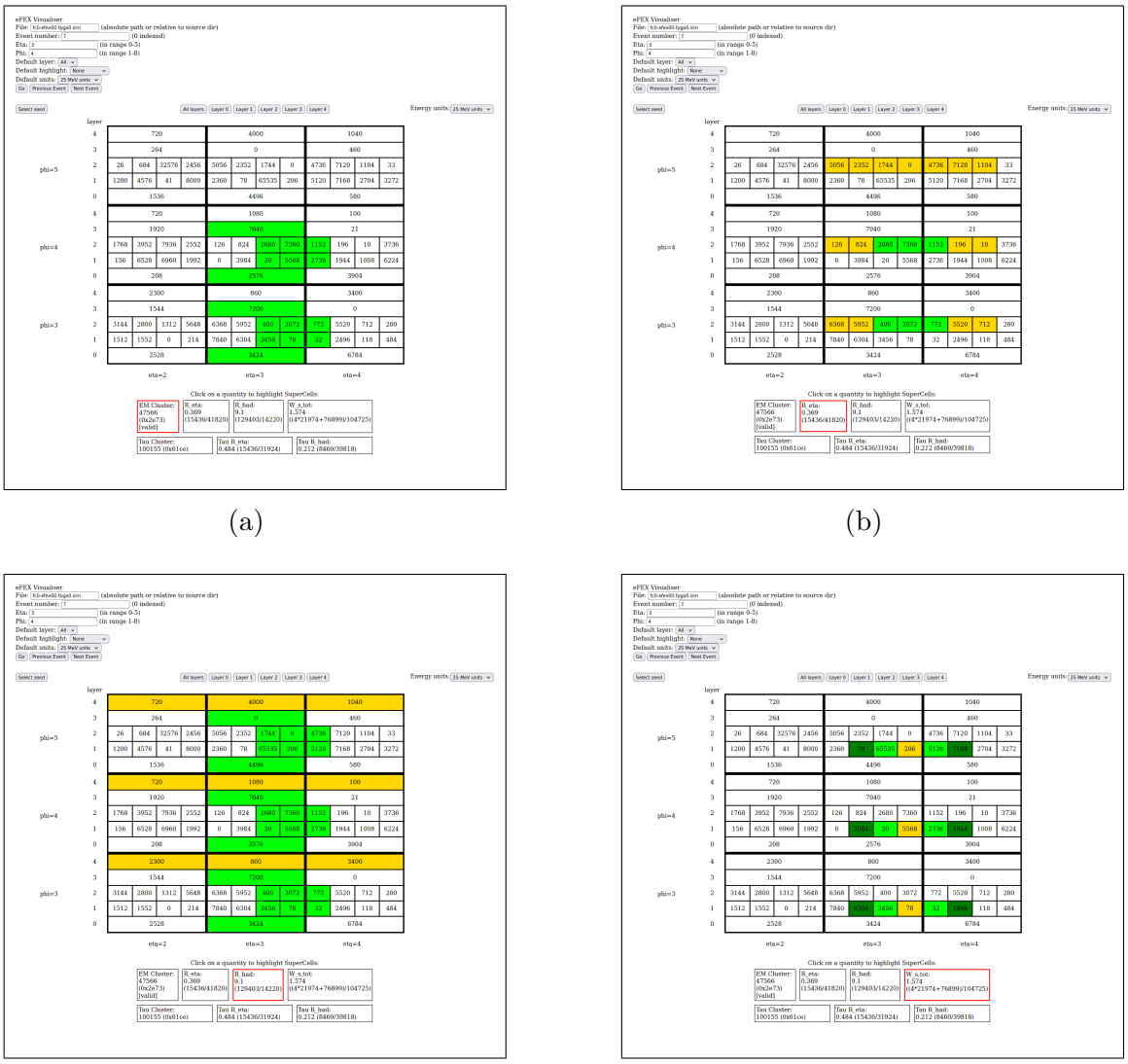


(c)



(d)

Figure 5.5: Demonstration of highlighting used to visualise algorithms, all shown for the same TOB. Showing (a) initial view without highlighting, (b) highlighting for tau cluster energy, (c) highlighting for tau R_η, and (d) highlighting for tau R_{had}.



(a)

(b)

(c)

(d)

Figure 5.6: Demonstration of highlighting used to visualise algorithms, all shown for the same TOB. Showing (a) highlighting for EM cluster energy, (b) highlighting for EM R_η, (c) highlighting for EM R_{had}, and (d) highlighting for EM $w_{s,\text{tot}}$.

5.2.5 Usage

The visualisation tool was used at several stages during commissioning of the eFEX. Primarily it was used to compare firmware algorithms to their implementation in online software. Each time there was a difference found between the two, the event could be checked with the visualiser to help determine which of them was correct and to work out how the other might have gone wrong.

Once there was sufficient confidence in the similarity between online simulation and firmware implementations, the visualiser was again used to help in tests comparing online and offline simulations. The same technique was applied here to help ensure offline simulations were running with the same results as their online equivalent, and to help find errors in cases where they were not.

5.3 Analysis of early Run-3 data for commissioning

At the start of Run 3, the Phase-1 L1Calo trigger was being used for the first time, having just been installed in the ATLAS detector. In these early stages, the new Phase-1 system was running in parallel to the Run-2 system, but the Run-2 system was still being used in the trigger menu over the new system.

One of the key goals of this time period was validating the Phase-1 trigger system, comparing it to the Run-2 system to identify any differences which may have arisen from bugs or hardware issues. This section describes analysis of some early Run-3 data contributing to this goal.

5.3.1 Data

Two runs were used to provide the data for this analysis: Run 423433, taken on 31 May 2022, and Run 427885, from 10 July 2022. These runs were taken in quite

different conditions, the first with lower intensity beams and no stable beam conditions, and the second with high intensity stable beams. Notably, the second of these runs had bunch trains with 25 ns separation between bunches, whereas the first had only isolated bunches.

Events are taken from the `physics_Main` stream. This stream contains 1,636,636 events for Run 423433 and 107,016 events for Run 427885.

5.3.2 TOB and RoI selection

Phase-1 TOBs and Run-2 RoIs in events are compared to find instances in the same event that have the same, or very similar, η-ϕ coordinates. A pair is formed by selecting, for each TOB, the nearest RoI that has not already been matched to a TOB. A match is considered to be a pair of objects within ± 1 trigger tower in both η and ϕ, i.e. a TOB matches an RoI if it falls within the 3×3 area of trigger towers centred on the tower containing the RoI. Matched objects are considered to be the same physics object, identified independently by both systems. Instances where there is a TOB or RoI with no analogue in the opposing system are also tracked.

Only the barrel region was considered for this as a preliminary investigation, since it has a simpler geometry and as such it is easier to isolate bugs. Also, at the time of analysing, only half of the eFEX modules were installed so the Phase-1 system had coverage for half of the ϕ range; only RoIs inside of this coverage are accepted.

5.3.3 Results

From the 1,636,636 events in Run 423433, 292,498 RoI/TOB pairs are selected. Of these, 271,854 matched in $\eta - \phi$ coordinates, giving a total match rate of 93%. For Run 427885, 22,337 of 27,973 pairs were matched for a match rate of 80%. Figures 5.7 and 5.8 show the match rate for objects in Run 427885 as a function of energy, using Cluster-processor Module (CPM)-measured and eFEX-measured

energies respectively. This shows that the bulk of the mismatches come from low-energy objects, with a notably higher match rate at higher energies. At all energies the match rate is worse here than in the earlier run.

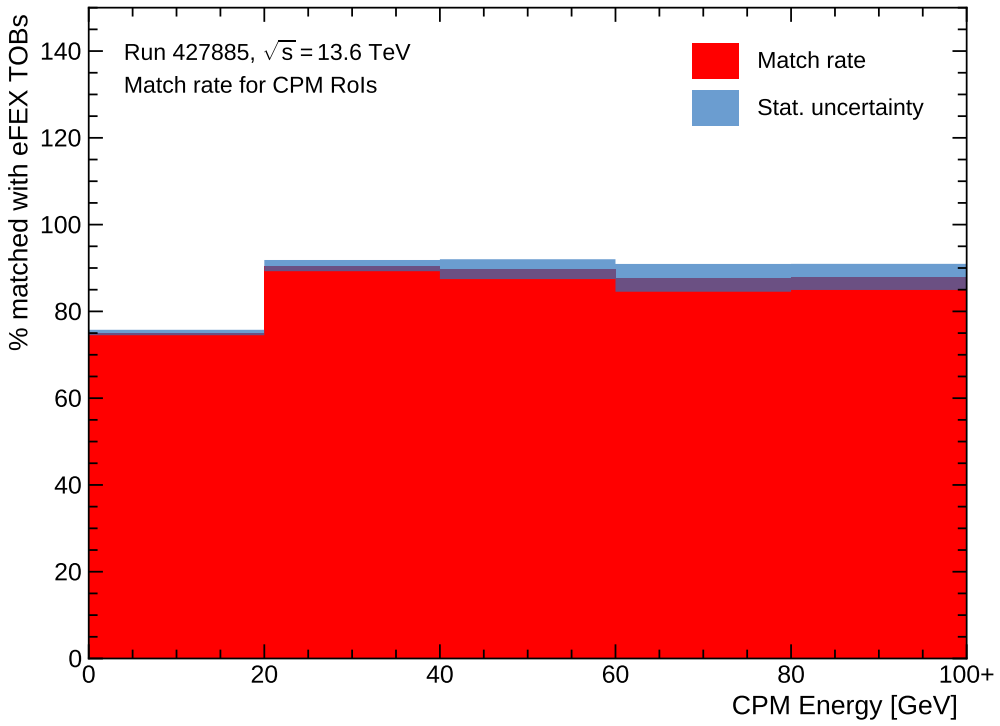


Figure 5.7: Match rate for TOBs/RoIs as a function of energy as measured by the CPM. Objects grouped in 20 GeV bins, with the last bin including all overflow.

Figures 5.9 and 5.10 compare the energies recorded by the Run-2 and the Phase-1 systems for matched objects in the two runs. In Run 423433 it is clear that the majority of matched objects have approximately the same energy, with an additional cluster where in a few cases the eFEX-measured energy is much lower than the CPM-measured.

In the later run, Run 427885, however, there is no longer such a strong correlation in energies. It seems that in general the eFEX energies are lower than the CPM energies – seen by the gradient of the area containing the majority of objects being less than the equal-energies line. Once again there is another cluster of objects with very low eFEX energies at high CPM energies.

The general trend is a high but imperfect match rate and decreased performance in

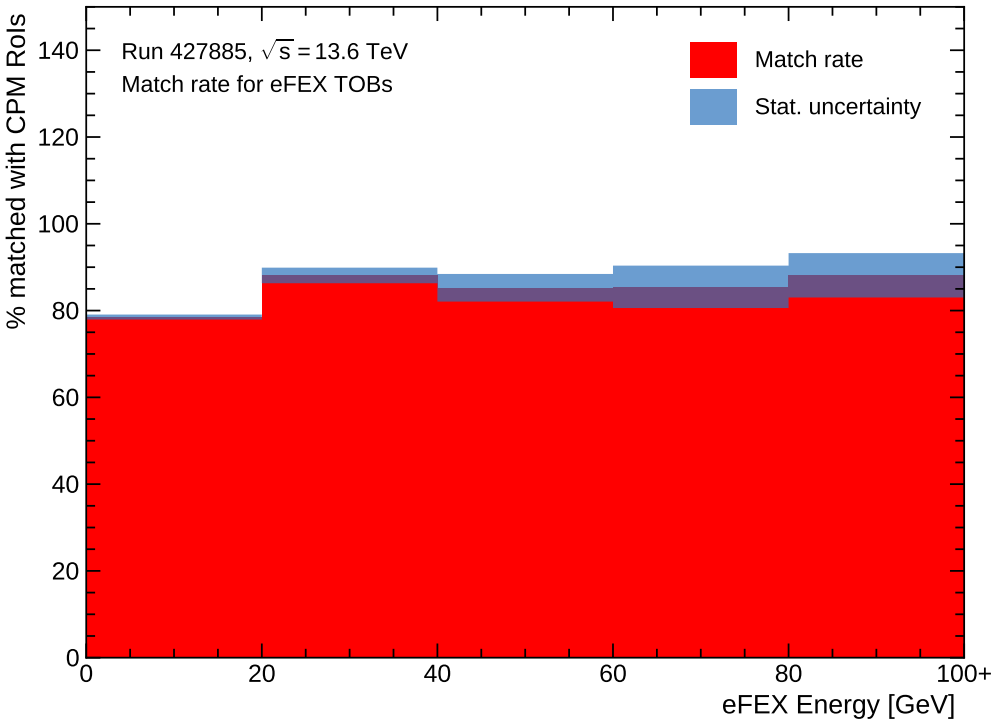


Figure 5.8: Match rate for TOBs/RoIs as a function of energy as measured by the eFEX. Objects grouped in 20 GeV bins, with the last bin including all overflow.

the later run compared to the earlier run, both in terms of match rate of objects and energy correlation between the two systems. From this information, issues in the system could be identified and solved. In the case of the degraded performance for Run 427885, the different beam conditions in this run (bunch trains, that were not present for Run 423433) were deemed to have caused issues with the Bunch Crossing ID (BCID) on the Liquid Argon Trigger Optical Mezzanine (LATOME) modules providing the eFEX with digitised energies from the calorimeter.

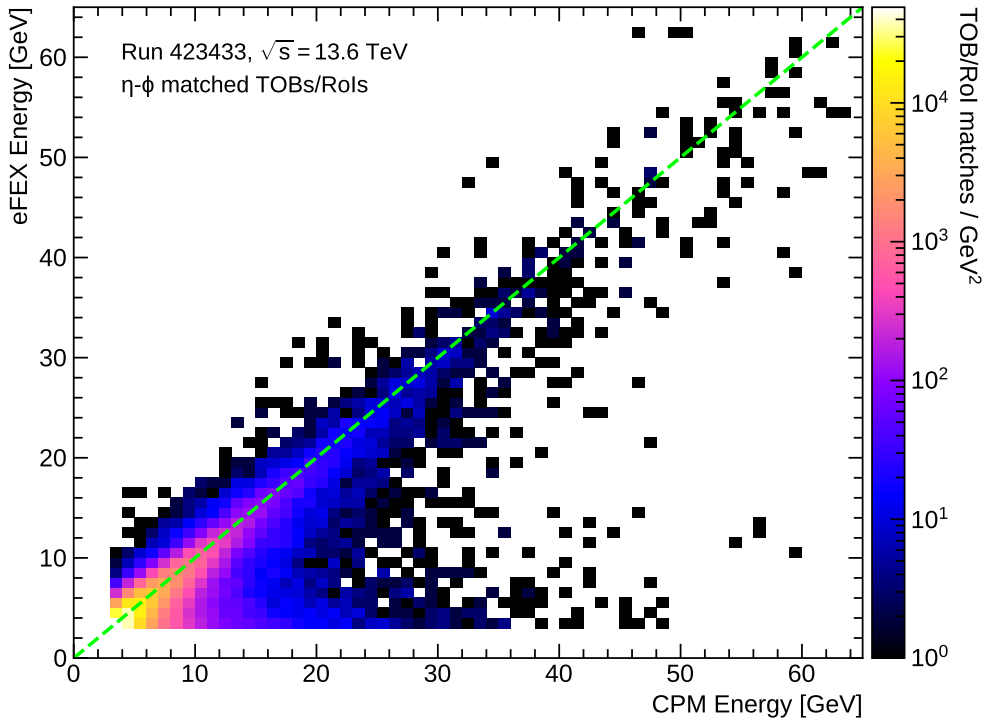


Figure 5.9: Comparison of energies for matched TOBs/RoIs with the energy as measured by the CPM given on the x-axis and as measured by the eFEX on the y-axis. Contains data for all matched objects in Run 423433. The dashed line marks the set of points where the CPM and eFEX energies are equal.

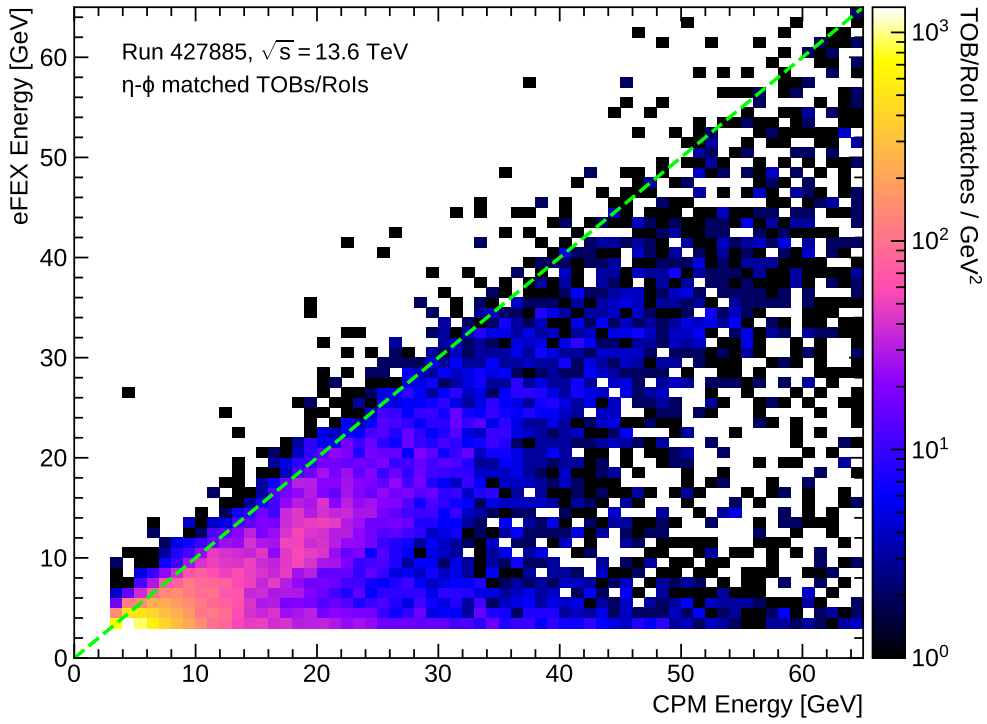


Figure 5.10: Comparison of energies for matched TOBs/RoIs with the energy as measured by the CPM given on the x-axis and as measured by the eFEX on the y-axis. Contains data for all matched objects in Run 427885. The dashed line marks the set of points where the CPM and eFEX energies are equal.

5.4 Performance studies of electron and photon algorithms for the Global Event Processor

5.4.1 Introduction

The GEP, when introduced in the Phase-II upgrade, will aim to improve discrimination in the hardware trigger for many signatures, but notably for e/γ objects. The GEP will be working alongside the eFEX system, introduced in Phase I, but will have access to more information, giving it potential to improve upon decisions made by the eFEX. To realise this improvement, new algorithms will need to be implemented in the GEP to take advantage of the finer granularity information available to it.

Designing algorithms to be used in future hardware systems is achieved through prospective performance studies. Performance studies use simulations of the expected response of a system to evaluate the performance of individual algorithms. These studies benefit from the ease of implementation of algorithms in high-level software but it is still important to consider the complexity of implementation in firmware when designing algorithms. Performance studies are typically the first step in designing a system as evaluating performance in simulations before a system is built can inform the design of the hardware.

This section explores the specific implementation and possible performance of the E_{ratio} algorithm in the GEP, expected to significantly improve discrimination for e/γ by making use of fine granularity input information [3, p. 126]. Section 5.4.2 details the samples used for evaluating algorithm performance, Section 5.4.3 discusses how the GEP itself is simulated, Section 5.4.4 gives metrics used to evaluate performance, then Section 5.4.5 goes through the process of designing an algorithm, the outcome of which is evaluated in Section 5.4.6.

5.4.2 Monte Carlo samples

Simulations used for the performance studies presented here are from two Monte Carlo samples: a Z→ *ee* sample providing signal EM objects that the trigger should be accepting, and a minimum bias QCD sample providing background objects, typically low-energy jets, that the trigger should be rejecting. The signal sample is generated by POWHEG [4] and PYTHIA 8 [5, 6], and the background sample is generated by PYTHIA 8.

After being processed by the typical ATLAS detector simulation (see Section X), additional simulations of the upgraded trigger are performed by the Phase-I offline software in order to simulate the eFEX response to each event.

5.4.3 Phase-II simulation

Producing prospective results for the Phase-II trigger requires simulation of the requisite algorithms. On top of the existing simulations of the Phase-I simulations, two things are needed to produce the results possible with Phase II: collection of the higher granularity calorimeter data that will be available to the GEP, and any algorithms that the GEP will run on its input data.

The first of these tasks is done by taking the location of e/γ candidate TOBs identified by the simulated eFEX, collecting calorimeter cell energies in a region around this location, and storing it in a cluster. This method was chosen over storing calorimeter cell output in the entire detector to reduce computing requirements. The size of stored clusters is 0.3×0.3 in $\eta \times \phi$, centred on the seed TOB location, chosen conservatively to be sufficiently large that it will contain all information required by any algorithm.

Samples with these clusters of high-granularity calorimeter data included are then used for developing prospective algorithms for the GEP, explored in detail in Section 5.4.5.

5.4.4 Performance benchmarks

These studies focus on performance of the e/γ trigger at hardware level. As such, the goal is to maintain a signal efficiency, fraction of signal events selected by the trigger, as high as possible. At the same time the amount of background being rejected should be as high as possible; this responds to maximising the background rejection, where

$$\text{background rejection} \quad = \quad \frac{1}{\text{fraction of background events selected}}.$$

Both signal efficiency and background rejection will be dependent on the selections made by different algorithms. To compare algorithms, or different variants of an algorithm, both of these quantities must be considered. The typical benchmark used in these studies will be the background rejection at 95% signal efficiency.

5.4.5 E_{ratio} algorithm design

The focus for this study is on the impact of a single variable in e/γ discrimination, E_{ratio}. E_{ratio} is a shower-shape variable, already used in the High-level Trigger (HLT). The definition used here is

$$E_{\text{ratio}} = \frac{E_2}{E_1} \tag{5.1}$$

where E_1 and E_2 are the first and second most energetic cells in Layer 1 of the Electromagnetic (EM) calorimeter in an area around the centre of the shower.[1]

The E_{ratio} variable is designed to discriminate against substructure in a shower. A shower with multiple distinct branches (e.g. $\pi^0 \to \gamma\gamma$) might produce two peaks of similar energy and give an E_{ratio} value close to one, whereas a shower with a single peak (as expected from e/γ clusters) would give an E_{ratio} value close to zero.

[1] This is different to the definition used in the HLT, which instead is $E_{\text{ratio}} = (E_1 - E_2)/(E_1 + E_2)$. The simpler definition is preferred here in the spirit of reducing calculation in firmware, though the two forms are a transformation of one another.

Calculating this variable in high-level software is straightforward and requires no optimisation. However, identifying the two required maxima involves a large number of comparisons between cell energies. An algorithm developed to run on hardware should be as simple as possible, therefore designing an alternate implementation is beneficial to minimise the impact of this algorithm on the latency of the GEP system.

A simple approach to finding the two highest energy cells in a cluster is to form a sorted list of all energies from Layer 1 cells, or at least sufficiently sorted to be confident in the highest two energies. Sorting algorithms are a very well-understood problem and heavily optimised but this approach is very rigid, not allowing for any tuning of the algorithm. For example, a cluster may have no substructure but fall on the boundary between two cells, depositing a similar amount in each. This would result in a high, background-like, E_{ratio} value. To avoid this the algorithm could include a minimum distance between cells considered to be the two maxima, or try to identify minima between the two, this would greatly complicate a list-sorting approach.

The most complete, but resource-heavy, method might consist of fitting some functional form to the energies as a function of η and ϕ to extract the peak energies. This might work in software but is very computationally expensive, even if possible to implement in firmware it is likely not worth the latency it would require.

The desired solution is an algorithm for calculating an E_{ratio}-type variable that comes somewhere between these two options, more adaptable than the list-sorting approach and less resource-heavy than the peak fitting. The following sections explore such an algorithm: a baseline algorithm for finding secondary maxima in clusters in the GEP is established in Section 5.4.5.1; parameters of the algorithm are tuned using simulations in Sections 5.4.5.2, 5.4.5.3, and 5.4.5.4; and a summary of the results and recommended parameters, as well as additional adjustments that could be made with further studies, is given in Section 5.4.6.

5.4.5.1 Initial algorithm

Identifying the two highest energy cells is done in three stages: locating the seed, identifying candidate secondary maxima, and comparing results.

The GEP will receive a seed location from the eFEX identifying which SuperCell has the highest energy. The cells within this SuperCell are compared with one another to find which has the highest energy, this becomes the seed cell for the E_{ratio} algorithm.

The algorithm will then perform a stepwise search from the seed outwards to identify peaks in energy. On each step the energy gradient is calculated as $\Delta E = E^{\text{cell}}_{\text{next}} - E^{\text{cell}}_{\text{prev}}$, where $E^{\text{cell}}_{\text{next}}$ is the energy of the cell being stepped to, and $E^{\text{cell}}_{\text{prev}}$ is the energy of the cell being stepped from. From the first step ΔE should be negative, as the seed will have a higher energy than the surrounding cells, but on subsequent steps ΔE may become positive, marking that a minimum-energy point has been passed. If, after this, ΔE becomes negative again it indicates that the previous cell was a local maximum; in this case that cell is added to a list of candidate secondary maxima, and the search stops along this route. If the edge of the available range of cells is reached before ΔE turns positive then no candidate is saved. If the edge is reached after ΔE turns positive, but before it turns negative again, then the last cell in the range is taken to be the candidate.

This stepwise search is done in 6 different routes from the seed: one route where each step from the seed is in positive η, one in negative η, two where the first step is in positive ϕ before proceeding in positive or negative η, and two following the same pattern with the first step in negative ϕ. Figure 5.11 cells, alongside a schematic depicting the peak location strategy.

Once the stepwise search is complete, up to 6 candidate secondary maxima will have been identified. The candidate with the largest energy is taken as the secondary maximum and, with the seed as the maximum, E_{ratio} can be calculated using Equation 5.1.

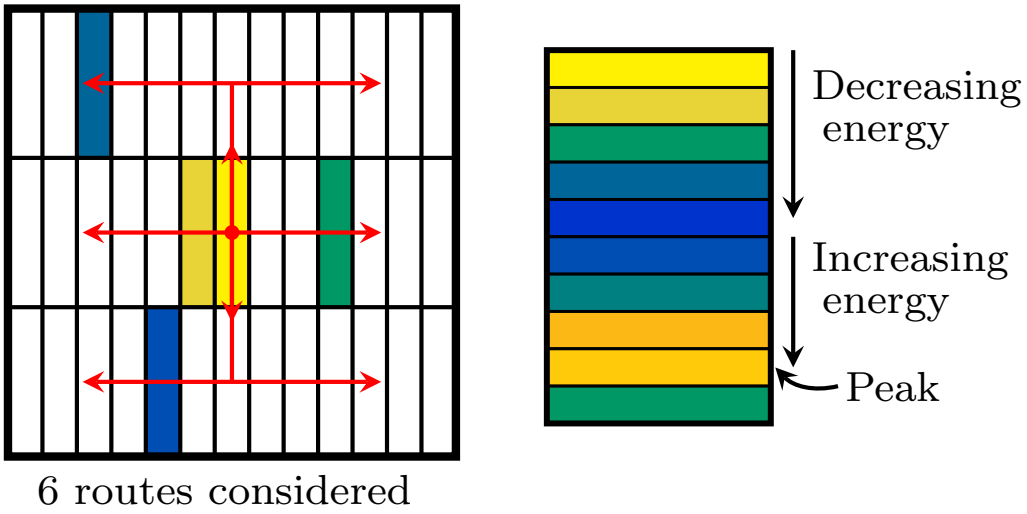


Figure 5.11: Diagram showing the 6 different routes in which the E_{ratio} algorithm searches for secondary maxima (left) and how the algorithm identifies secondary maxima by tracking energy gradients along each step (right).

The performance of this baseline algorithm was investigated using simulations. Figure 5.12 shows the results, comparing the response in signal and background as a function of the calculated E_{ratio} value and the fraction of each that would pass a given E_{ratio} threshold. The background rejection as a function of signal efficiency is also shown, the baseline algorithm achieves a background rejection of 2.3 at 95% signal efficiency.

5.4.5.2 Peak size

The first parameter to investigate is the size of the area used to calculate energies. In the algorithm as described in Section 5.4.5.1, the energies used in comparisons and in the final E_{ratio} calculation are always the energies of a single cell. This could be modified by instead summing the energy of a cell with that of its neighbours in η to reduce sensitivity to small fluctuations. The number of cells summed is labelled the 'peak size', where the default algorithm would have a peak size of one. With a peak size greater than one the algorithm uses a 'sliding window' approach, so the step size is still a single cell despite the energy value coming from a larger area. For

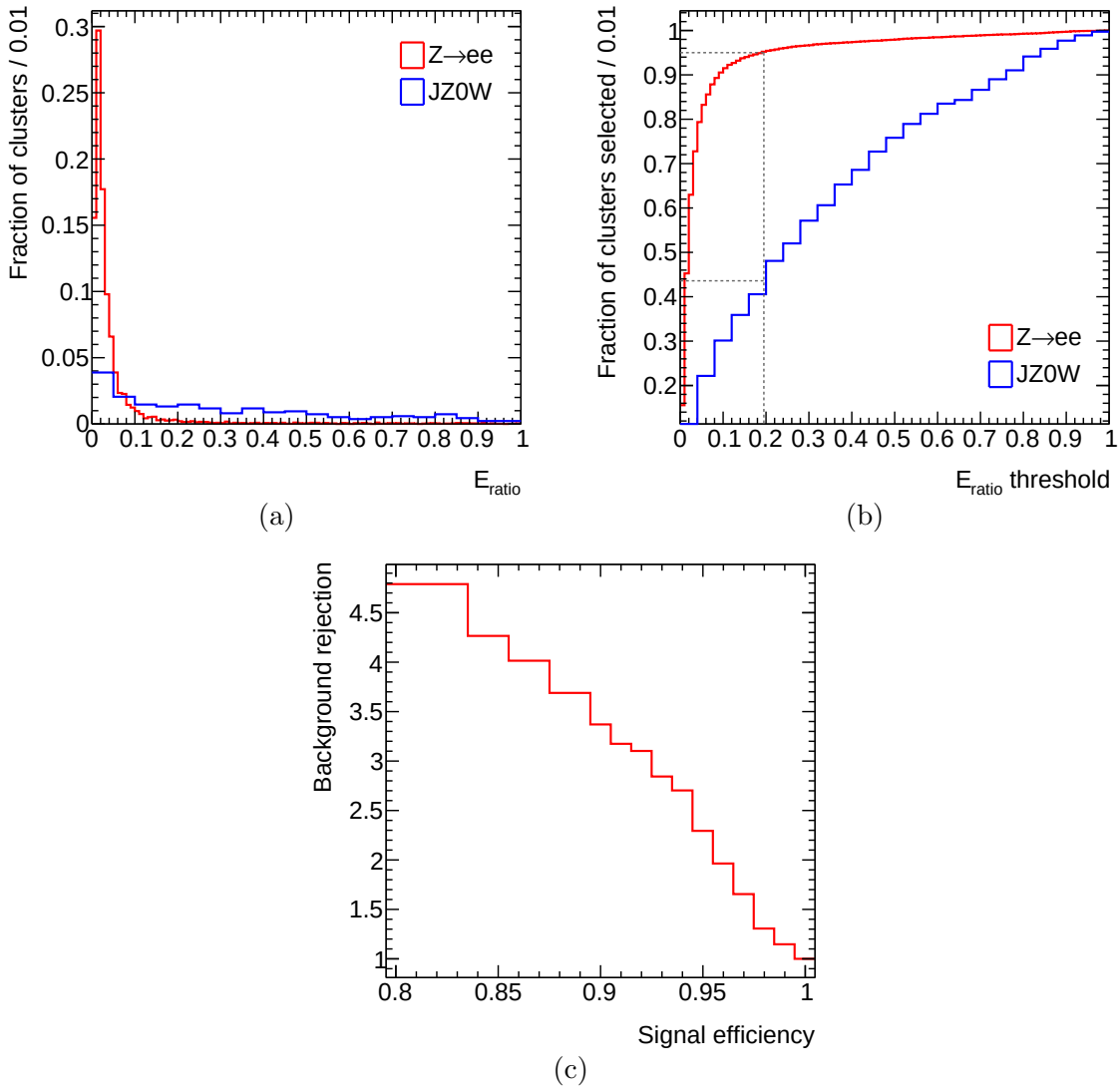


Figure 5.12: Performance of baseline E_{ratio} algorithm on signal ($Z \to ee$) and background ($JZ0W$) clusters. Plots show (a) a histogram of calculated E_{ratio} values for each cluster, (b) the integral of (a) with a grey dashed line indicating the values at 95% signal efficiency, and (c) the background rejection of an E_{ratio} threshold corresponding to a given signal efficiency.

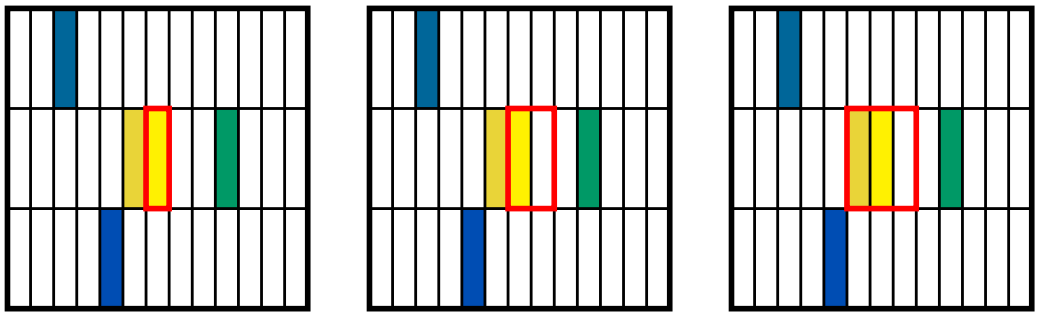


Figure 5.13: Diagram showing which cells contribute to the energy sum for the seed cell (bright yellow) for different peak sizes. The horizontal axis represents η and the vertical axis ϕ. In each case, the calculated energy would be the sum of the energies of the cells contained within the red box.

an odd numbered peak size the energy of a cell is added to that of its neighbours on each side. For an even numbered peak size, neighbours in positive η are preferred. Figure 5.13 shows how cells are included in the calculated energy.

Performance for the E_{ratio} algorithm was tested with peak size values from one to five. The results are shown in Figure 5.14. Comparing the benchmark value of background rejection at 95% signal efficiency, it is clear that a peak size of one (i.e. the same as the baseline algorithm) gives the best results, with performance degrading as more cells are added to the energy sum. This appears to be generally true for background rejection at all signal efficiencies. This suggests that the benefit of the fine granularity of each energy measurement outweighs the negative impact of any potential fluctuations that the increased peak size would smear out.

5.4.5.3 Exclusion region

Another alteration tested on the E_{ratio} algorithm is an 'exclusion region' around the seed, i.e. a number of cells close to the seed in which secondary maxima will not be searched for. An n-cell exclusion region means making the first step along any route n cells away from the seed in η. Since secondary maxima can be found as

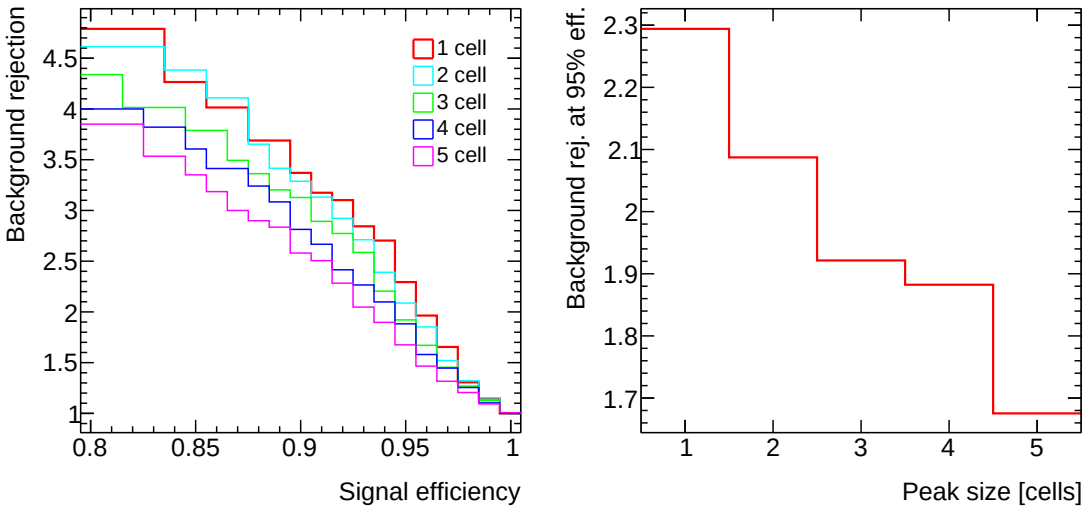


Figure 5.14: Results for calculating E_{ratio} with different peak size options. Plots show background rejection as a function of signal efficiency for each peak size tested (left) and background rejection at 95% signal efficiency as a function of peak size (right).

soon as two steps have been taken from the seed cell (they cannot be found on the first step as it will always be a step down from the seed), this excludes all cells in an η range from $-n$ to n (in relative coordinates) from being considered secondary maxima. Comparatively, the baseline algorithm with no exclusion region can find secondary maxima anywhere but the four cells directly adjacent to the seed. Figure 5.15 highlights the effect of the exclusion region.

Performance for the E_{ratio} algorithm was tested with exclusion regions from between one and five cells, shown in Figure 5.16 alongside the baseline algorithm with no exclusion region. This time a clear increase in performance is visible compared to the initial form of the algorithm, with a one-cell exclusion region attaining a background rejection of 3.1 at 95% signal efficiency. For most signal efficiencies the one-cell exclusion still seems to perform best, though perhaps competing with a two-cell exclusion region for very high signal efficiencies. Since the only difference between no exclusion region and the one-cell case is that cells diagonally adjacent to the seed are excluded, these results suggest signal clusters frequently create secondary peaks on these diagonal; this could stem from incident particles falling close to the corner of a cell.

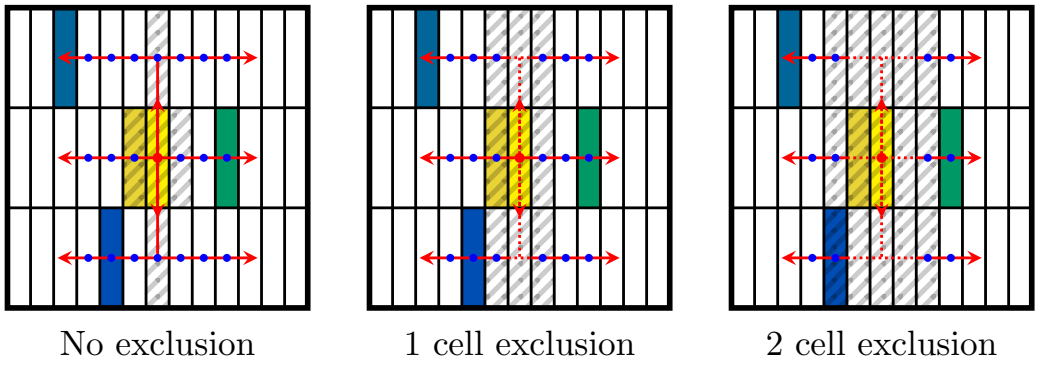


Figure 5.15: Diagram showing how the introduction of an exclusion region to the E_{ratio} algorithm prevents secondary maxima close to the seed from being selected. Red arrows mark each of the six paths traversed by the stepwise algorithm. Blue dots mark each step where the energy gradient is calculated. The shaded grey area shows cells that cannot be selected as a candidate secondary maximum, due to either being skipped over or being the first step from the seed.

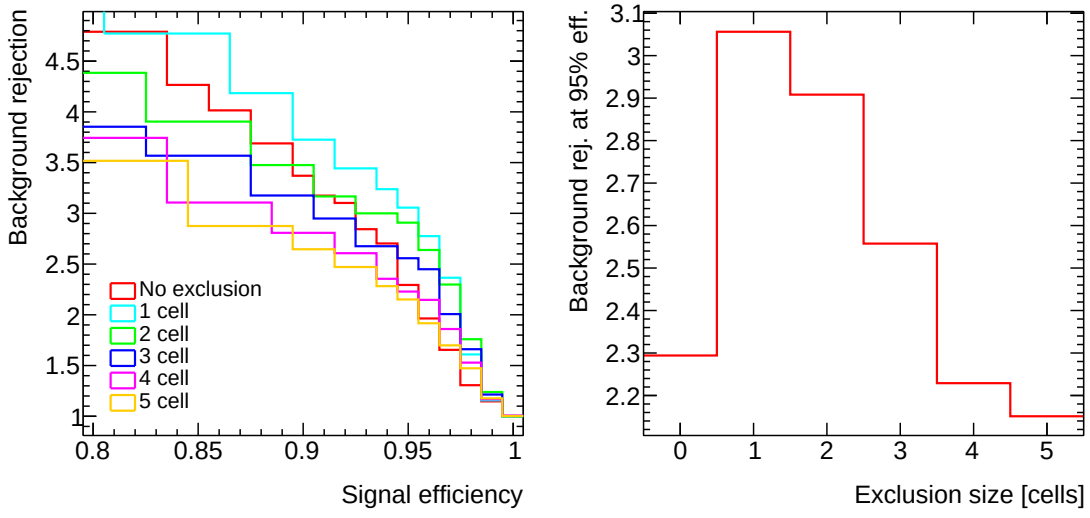


Figure 5.16: Results for calculating E_{ratio} with different or no exclusion region definitions. Plots show background rejection as a function of signal efficiency for each tested exclusion region (left) and background rejection at 95% signal efficiency as a function of exclusion region size (right).

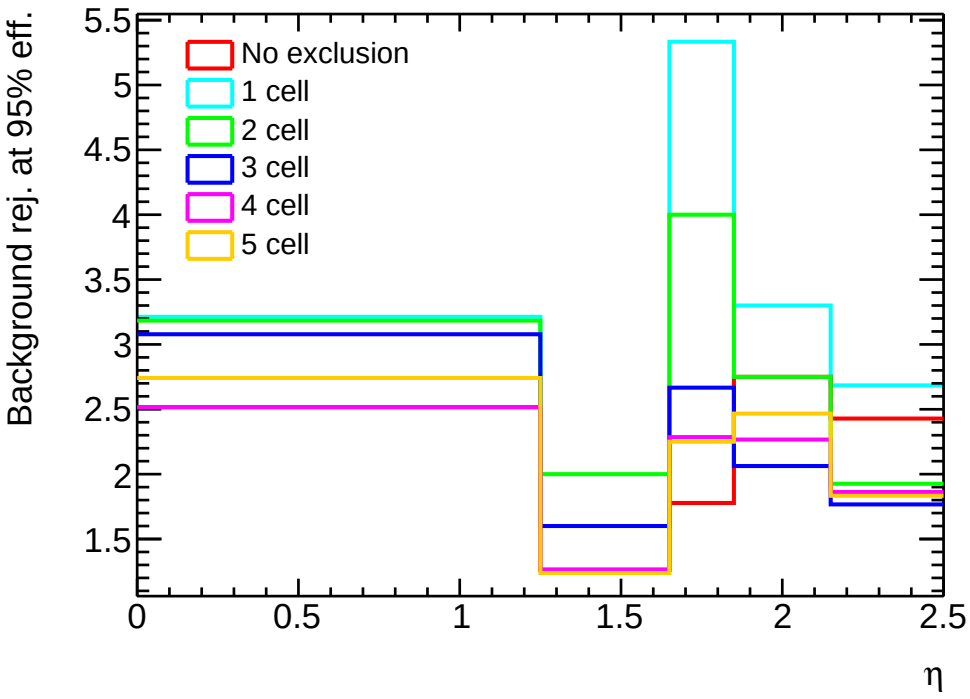


Figure 5.17: Plot of background rejection at 95% signal efficiency as a function of pseudorapidity, η, for E_{ratio} algorithms with different exclusion regions.

Given that cell widths vary significantly in different regions of the calorimeter, the performance for the E_{ratio} algorithm with different exclusion widths was also tested as a function of η. Figure 5.17 compares background rejection at 95% signal efficiency in several η regions. It is evident that the one-cell exclusion region performs best regardless or calorimeter geometry. The difference between one-cell and two-cell is much more drastic in the high-η endcap regions, here the strips are less granular so likely the larger exclusion regions are starting to miss real secondary peaks in background clusters.

5.4.5.4 Search limit

In the baseline E_{ratio} algorithm, the stepwise search for secondary maxima extends as far as the available data allows, in this case to the edge of the, conservatively large, stored cluster size. To minimise the amount of processing required by the algorithm, and potentially improve performance by reducing overlap with other clusters, a limit

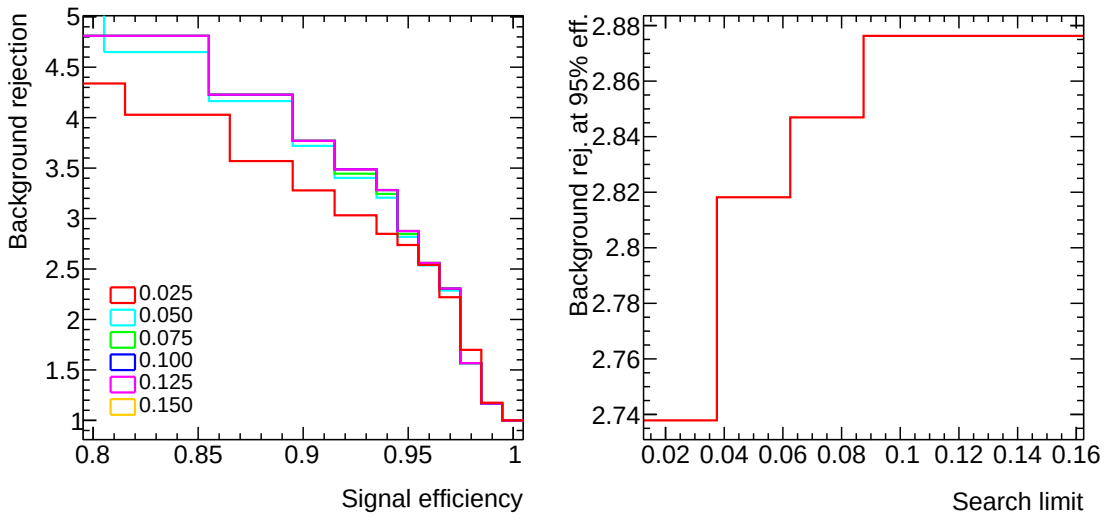


Figure 5.18: Results for calculating E_{ratio} after varying the search limit parameter, given as a distance in η from the seed cell. Plots show background rejection as a function of signal efficiency for each tested search limit (left) and background rejection at 95% signal efficiency as a function of the search limit (right).

can be placed on the distance this search will traverse. Since the ϕ range of the search is already limited to one cell either side of the peak, this search limit is implemented as a maximum distance traversed in η. This distance is calculated in pseudorapidity units rather than number of cells to give a consistent response across calorimeter regions.

Performance for the E_{ratio} algorithm with different search limit values was tested on simulations, with the results presented in Figure 5.18. Distances in $\Delta\eta$ from 0.025 up to 0.15 were tested, with 0.15 being the width of the clusters and thus the limit in place in the baseline algorithm. While no performance gains are seen by reducing the search limit, there is a plateau in performance from $\Delta\eta > 0.1$. This means the required cluster size, and thus the amount of computation required, can be reduced without degrading performance of the algorithm.

5.4.6 Algorithm summary

Given the results presented in Sections 5.4.5.2-5.4.5.4, the most performant and resource-efficient algorithm for calculating E_{ratio} in the GEP is the baseline algorithm established in Section 5.4.5.1 with an exclusion region of one cell and a search limit of $\Delta\eta = 0.1$. No further improvement was found by varying the peak size. This algorithm achieves a background rejection of 3.1 for 95% signal efficiency.

The E_{ratio} algorithm presented here is functionally complete and serves as an option for e/γ discrimination in the GEP, but further improvements could be made with additional study. More parameters could be varied and tested for performance against simulations. One example is a threshold in the energy gradient between steps to allow a change in gradient to be identified, which might improve the response of the algorithm to noise or statistical fluctuations.

REFERENCES

[1] I. Brawn. *L1Calo Overview, Status, Installation & Commissioning* (Oct 2019). https://indico.cern.ch/event/829769/contributions/3572289. [ATLAS Internal].

[2] The ATLAS Collaboration. *Technical Design Report for the Phase-I Upgrade of the ATLAS TDAQ System*. Technical Report ATLAS-TDR-023-2013 (2013). http://cds.cern.ch/record/1602235/.

[3] The ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System*. Technical Report ATLAS-TDR-029, CERN, Geneva (2017). https://cds.cern.ch/record/2285584.

[4] S. Frixione, P. Nason, and C. Oleari. *Matching NLO QCD computations with parton shower simulations: The POWHEG method*. J. High Energy Phys. 2007(11) (Nov 2007) p. 070. ISSN 11266708. http://dx.doi.org/10.1088/1126-6708/2007/11/070. 0709.2092.

[5] T. Sjöstrand, S. Mrenna, and P. Skands. *PYTHIA 6.4 physics and manual* (May 2006). http://dx.doi.org/10.1088/1126-6708/2006/05/026. 0603175.

[6] T. Sjöstrand, S. Mrenna, and P. Skands. *A brief introduction to PYTHIA 8.1*. Comput. Phys. Commun. 178(11) (Jun 2008) pp. 852–867. ISSN 00104655. http://dx.doi.org/10.1016/j.cpc.2008.01.036. 0710.3820.