My Amazing Thesis

A. Author

Thesis submitted for the degree of
Doctor of Philosophy

UNIVERSITY OF
BIRMINGHAM

Particle Physics Group,
School of Physics and Astronomy,
University of Birmingham.

January 19, 2023

ABSTRACT

X was measured, we showed that $Y \neq Z$ and that $M_{\mathrm{H}} = 126$ GeV$/c^2$.

DECLARATION OF AUTHORS CONTRIBUTION

I did this, and that, and some of the other.

ACKNOWLEDGEMENTS

I would like to thank bla, and bla ...

Motto or dedication

Contents

List of Tables

List of Figures

DEFINITIONS OF ACRONYMS

LHC Large Hadron Collider
Superconducting collider occupying the 27 km ring at CERN.

QCD Quantum Chromodynamics

CHAPTER 1

Introduction

CHAPTER 2

Theory

CHAPTER 3

The ATLAS detector at the Large Hadron Collider

CHAPTER 4

Object reconstruction

CHAPTER 5

Contributions to the ATLAS Level-1 Calorimeter Trigger

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

5.2 Introduction

This project aims to provide an outline of possible discriminating variables for triggering on e/γ events with the GEP in the Phase-II upgrade of the LHC. This is achieved by using MC simulations to evaluate the performance of candidate algorithms, given the expected hardware available for the trigger at that time.

The studies outlined in this chapter use two main MC samples: the objects of interest are taken from electrons in simulated events of a Z boson decaying to two electrons, this will be described as the signal sample; the background objects that the trigger aims to filter out are taken from minimum bias QCD events producing (mostly) low-energy jets, this will be described as the background sample. Each of these samples are initially simulations of the raw physics process. The signal sample is generated using Powheg [?] and Pythia 8 [?, ?], the background sample is generated with Pythia 8. The physics simulations are then passed through a simulation of the ATLAS detector, as described in Section ??. Additional simulations of the upgraded trigger are then performed by the Phase-I upgrade software, henceforth referred to as `R3L1Sim`.

In order to achieve the project's goals there was an initial period of learning how to use the `R3L1Sim` software, this included recreating some typical results made by the developers of the software; this work is detailed below under preliminary studies. In order to study e/γ performance with the GEP at Phase-II some modifications had to be made to the simulation software; the software development section describes the necessary changes made. Once these foundations had been laid the first candidate variable was investigated; the implementation, performance, and optimisation of the $\mathrm{E}_{\mathrm{ratio}}$ algorithm is described in Section 5.5

5.3 Preliminary Studies

Preliminary studies were undertaken to gain familiarity with the codebase for Phase-I studies, which the Phase-II software detailed here is built upon, and to learn how typical trigger results are created. The key results from this study are two plots: one comparing efficiency of two different trigger thresholds at selecting signal events, as a function of the transverse energy of the candidate electron; and one comparing the output e/γ rate from the L1 trigger for different versions of the trigger algorithm, as a function of the electron transverse energy threshold required for an event to contribute to the output rate. These results are recreations of existing results from prior studies, but using original code to analyse the output of L1 trigger simulations.

The exercise was intended to be more straightforward than it turned out to be. There were issues with code, documentation, and MC samples that had to be resolved to be able to obtain the results. The following sections document the method used to generate each of the results.

It is useful to define the following quantities: 'truth electrons' refers to the properties (transverse energy, η and ϕ coordinates, etc.) of electrons as simulated at truth-level, before detector reconstruction; 'L1 electrons' refers to the same set of properties as measured by the simulated L1 system for objects that are identified as electrons at L1; 'offline electrons' refers to objects that have been identified as electrons by the full, more sophisticated, ATLAS software identification algorithms that would be used as electrons in typical analyses.

5.3.1 L1 trigger efficiency

The trigger efficiency for a given threshold is determined by calculating what fraction of all possible electrons pass that threshold. This requires determination of two quantities: the initial number of electrons present in the simulated data, and the number of electrons that pass the simulated L1 trigger threshold. The calculation

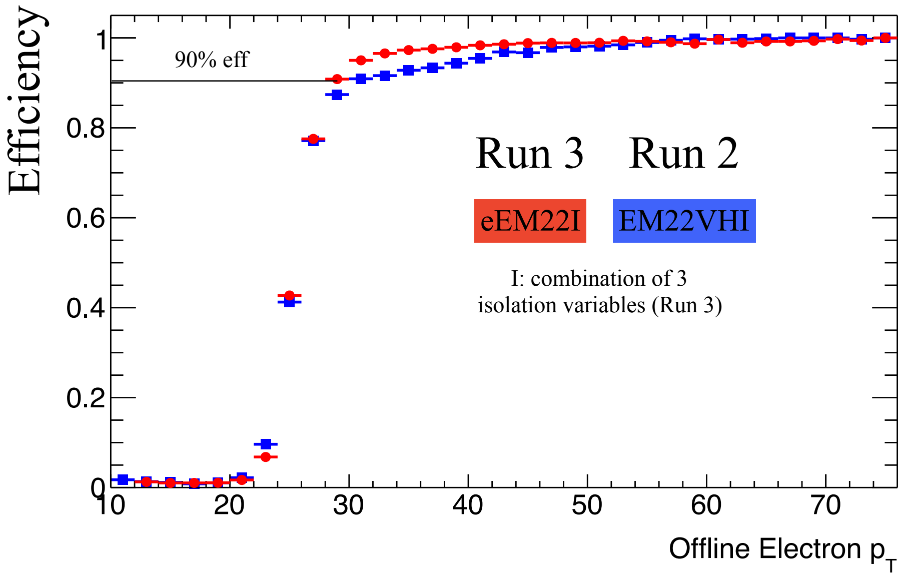


Figure 5.1: Efficiency of triggers selecting electrons from $Z \to ee$ sample as a function of the electron p_T (in GeV) as measured by the offline reconstruction system. Shown for two trigger thresholds: 'eEM22I', a Phase-I trigger threshold, and 'EM22VHI', a legacy trigger threshold. Result from Reference [?].

is dependent only on the signal sample as the efficiency in question is the efficiency of selecting signal events.

The particular result that this study was based on is shown in Figure 5.1. The graph compares the efficiency of a legacy trigger threshold to that of a comparable Phase-I threshold, demonstrating the increased performance with the Phase-I upgrade. The below details the steps taken to create an equivalent plot using the $Z \to ee$ signal sample processed with `R3L1Sim`.

The initial number of electrons is most simply determined from truth-level data in the MC sample. Given that this calculation is for the efficiency of the trigger, not the overall detection of electrons, some requirements must be set on truth electrons for them to be included among the total 'initial' electrons. Truth electrons that fall outside of the calorimeter's acceptance could not possibly be triggered on, thus the number of initial electrons can be taken as the number of truth electrons within calorimeter acceptance.

The number of electrons passing L1 trigger requirements can be taken directly from the output of `R3L1Sim`. This gives a set of L1 electrons along with their energy and isolation (or other variables in the Phase-I case) as it would be measured by the L1 trigger. Efficiency is typically calculated for a particular trigger threshold as a function of the 'real' electron E_T, e.g. truth electron E_T. The calculation is done in bins of E_T^{truth}, with the efficiency in each bin being the ratio of the number of electrons passing the chosen threshold to the number of initial electrons. The complication here is that to work out which bin the L1 electron falls in the E_T of the corresponding truth electron has to be used, this may differ from the L1 measured E_T. L1 electrons and truth electrons are not connected in the simulations a priori, so it is not trivial to find the truth E_T for an L1 electron. However, if the truth electron and L1 electron are both representing the same underlying electron, it should be possible to match between L1 and truth electrons if they have similar η and ϕ coordinates. The values may not be exactly identical due to the limited resolution of the trigger. Electron matching may be done by searching for truth and L1 electron pairings with a small ΔR:

$$\Delta R = \sqrt{\Delta\phi^2 + \Delta\eta^2}, \tag{5.1}$$

where $\Delta\eta$ and $\Delta\phi$ are the differences in the η and ϕ coordinates, respectively, of the two types of electron. Once L1 electrons in each event have been appropriately matched to a truth electron, the E_T of the truth electron can be used to split the L1 electrons into bins for the efficiency calculation.

To get the efficiency for a specific trigger threshold the L1 electrons have to be limited to those passing the threshold requirement. In this case there are two thresholds for which efficiencies are calculated: a legacy trigger threshold requiring electrons to have a transverse energy of at least 22 GeV and to satisfy isolation criteria, 'EM22VHI'; and a Phase-I trigger threshold also requiring electrons to be 22 GeV or higher and to pass all discriminating variable requirements, 'eEM22I'. The latter threshold is straightforward to implement in the code as the sample has variables for the energy and discriminating variables as measured in the Phase-I system. Any particular legacy threshold could be implemented by requiring a certain bit of the threshold word was equal to one. The issue was that there was no documentation about which bit corresponded to which threshold. The bits are set by a trigger menu, but there are multiple different trigger menus that could have been used. By inspecting the E_T and isolation distributions after each threshold bit was required to be one, the trigger menu used for the simulations was identified.

The expectation for the efficiency curves is for electrons well below the energy threshold (22 GeV in both the legacy and Phase-I case) to have zero or very small efficiency and for electrons well above the energy threshold to have close to 100% efficiency. In the region around the threshold energy there will be some smooth turn-on curve, not an immediate switch from 0 to 1 as the energies measured at L1 will have limited resolution. This pattern can be clearly seen in the example result (Figure 5.1).

In initial iterations of the efficiency calculations the shape had some clear discrepancies compared to the expectation described above. The first of these was that the low energy bins (10-20 GeV) had larger efficiency values than expected. This is speculated to be an issue with the definition of electrons at truth level compared to how they are detected. When an electron is travelling through the detector towards the calorimeter it may radiate a photon via bremsstrahlung. This will reduce the energy of the electron, but it is possible for the photon to be radiated at a small enough angle that it contributes to the same energy cluster in the calorimeter. The truth-level information takes the energy of the electron as its energy after radiating the photon, but the detector measures the entire energy of the electron before bremsstrahlung if the photon is included in the shower. This causes the truth electron energy to be lower than expected, thus some electrons in the low energy bins should actually be in higher energy bins, where the higher efficiencies are expected.

The solution to this is to change the estimate of the total number of electrons to use offline electrons instead of truth electrons. Technically this is not as complete a definition as using truth electrons as offline reconstruction itself will not be perfect. It serves as a suitable option for this purpose however as anything that fails offline

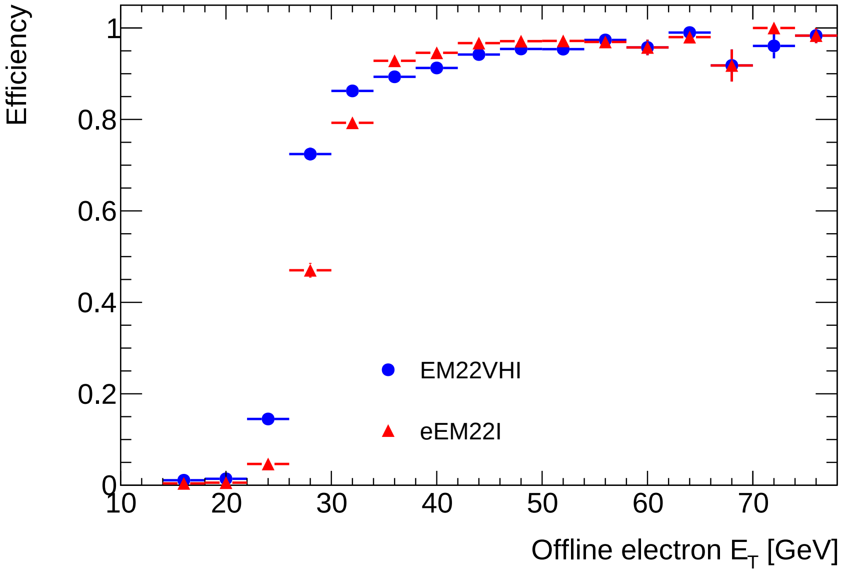


Figure 5.2: Efficiency of triggers selecting electrons from $Z \to ee$ sample as a function of the electron E_T as measured by the offline reconstruction system. Shown for two trigger thresholds: 'eEM22I', a Phase-I trigger threshold, and 'EM22VHI', a legacy trigger threshold.

electron ID should be out of reach of the trigger as it is essentially a simplified version of the offline system. Using offline electrons in place of truth electrons fixes the discrepancies for low-energy electrons, as hypothesised.

The high-energy electrons also had deviations from the expected form: the plateau for efficiencies was significantly below 100%. Upon investigating the distribution of efficiencies across η it appeared that these efficiency losses were coming from the transition region of the calorimeter. For the purposes of this study, electrons falling in the transition region, both L1 and offline, were excluded from calculations. This brought the plateau closer to the ideal value of 1.

The final efficiency plot is shown in Figure 5.2. This follows the expected shape, and matches the result from Figure 5.1.

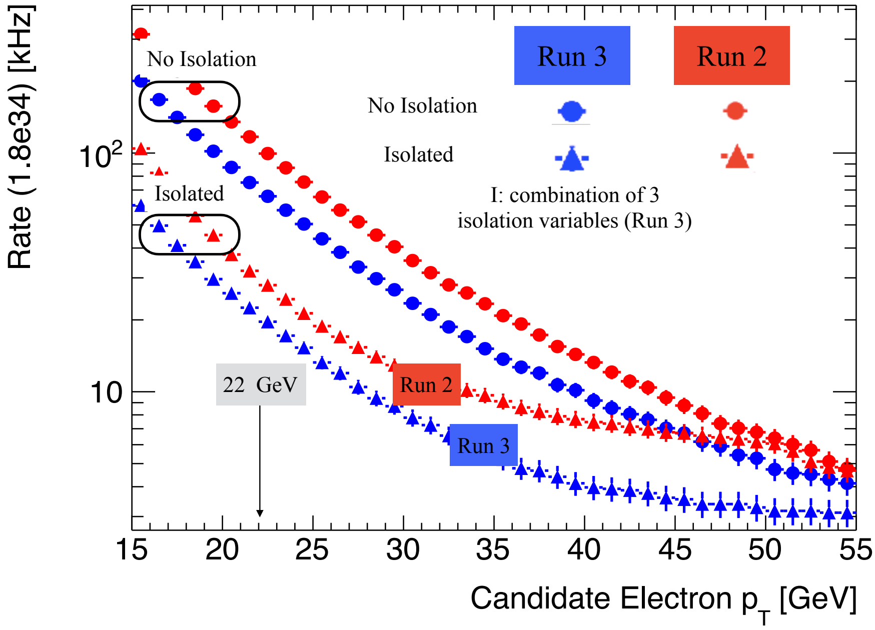


Figure 5.3: L1 output trigger rate for a selection of thresholds as a function of the required minimum L1 electron p_T. Calculated from simulated minimum bias jet samples. The rate given is assuming an instantaneous luminosity of 1.8×10^{34} cm^{-2}s^{-1}. Result from Reference [?].

5.3.2 L1 trigger rate

The rate can be calculated for a given trigger threshold by taking the fraction of all events that pass that threshold, and applying the appropriate normalisation. The majority of events passing the L1 trigger are still background events (as the decisions are further refined in later stages), so the rate can be estimated accurately using only the background sample from simulations.

This study aims to reproduce the result in Figure 5.3. This shows the rate for trigger thresholds with a range of E_T requirements for both the legacy and Phase-I systems. Separate curves are present for each system to demonstrate the rate with and without applying isolation/discriminating variables.

Rate calculation is performed in bins of E_T. For each bin, require that electron candidates have an E_T greater than that of the bin and measure the resulting rate. The E_T used for the candidate is as measured by the trigger system in question.

This calculation is repeated for both the Phase-I and legacy systems. Isolation requirements can also be applied in addition to the E_T threshold.

Determining if an electron, with a known legacy E_T in simulations, passes a given E_T threshold is more complicated than in the Phase-I case. The legacy system employs variable thresholds that are dependent on which η region of the detector candidates are found in, to account for detector response differences. This means that with a 26 GeV variable threshold an electron with an E_T of 25 GeV in the outermost region of the calorimeter could pass the trigger. The variable threshold can be simulated by modifying the L1 electron E_T based on its η coordinate. This is an approximate solution, the hardware implementation of the variable threshold in the trigger may give different results; this is why the threshold bits were used in the efficiency calculation, to avoid manual simulation of the variable thresholds.

Once the number of events passing the trigger for each threshold is known, the result can be scaled to obtain the predicted rate of the L1 trigger. In theory, this can be done by scaling the number of events by the ratio of the instantaneous luminosity of collisions in ATLAS to the integrated luminosity of the MC sample. This is complicated by the pileup collisions present in the background sample. The sample contains minimum-bias jet events, constrained to low transverse momentum, as the primary collision in each event. There are also around 80 additional pileup collisions in the event. The pileup is also simulated by minimum-bias jet events, but this time with no momentum constraint. This means that the pileup collisions are essentially the same as the primary collision, so each event actually contains around 80 instances of the collision of interest. This could be accounted for by dividing the rate by 80, but because of the differences in momentum constraints this will not be precise. The solution to this is instead to use rates measured from data to normalise the rate curve for the legacy system, and scale the Phase-I rates by the same factor. The relative difference in rate between the legacy and Phase-I systems should be accurate in the simulations, so this normalisation will give an accurate estimate for the Phase-I rate.

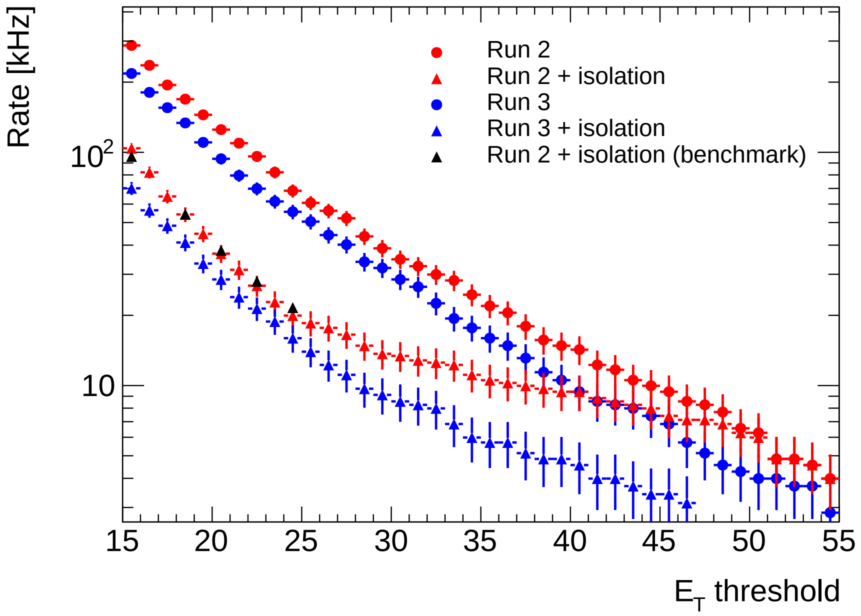


Figure 5.4: L1 output trigger rate for a selection of thresholds as a function of the required minimum L1 electron E_T. Calculated from simulated minimum bias jet samples. The rate given is assuming an instantaneous luminosity of 1.8×10^{34} $\text{cm}^{-2}\text{s}^{-1}$. The black points, marked as Run 2 benchmark, are calculated using the threshold words built into the simulation, as opposed to using approximate variable E_T thresholds, to validate the method.

Once all rate curves had been calculated, there were some clear discrepancies when compared to the example result (Figure 5.3), the shapes for the legacy system curves were clearly different. In the end this was determined to be an issue in the underlying MC sample used for the study. The nature of the issue is not precisely known, but updating to a more recent sample resolved the discrepancies.

The final result from this study is shown in Figure 5.4.

5.4 Software Development

The software used in preliminary studies, `R3L1Sim`, simulates the planned form of the trigger in the Phase-I upgrade. To be able to use this for performance studies required for the Phase-II upgrade some modifications must be made. The most significant of these is collecting information from individual calorimeter cells in the region around any e/γ candidates identified by the simulated Phase-I L1 trigger, referred to here as seeds. A framework for handling these clusters of cells has been developed, designed for ease of use in Phase-II performance studies.

The goal is to form a cluster of cells around each seed, containing all cells from Layer 1 of the EM calorimeter within a given distance in η and ϕ. Clusters were chosen to have a total width of 0.3 in both η and ϕ. This size should be suitable to contain all activity surrounding detected particles.

Clusters are complicated by the changing geometry of the calorimeter, described in Section **??**. For algorithms to be able to access cells sequentially without explicit knowledge of each cell's coordinates, the cluster must know the geometry of the cells in the calorimeter region it occupies. This is easily achievable if a cluster is fully contained within a single calorimeter region, but becomes difficult if a cluster crosses a region boundary.

The solution employed here is to split each cluster into subclusters, where each subcluster covers the extent of the cluster contained by a single calorimeter region. For instance, a cluster that falls towards the edge of the barrel might have two subclusters, one in the barrel region and one in the barrel-endcap transition region. Individual cells can then be accessed by indexing the subclusters with a coordinate system based on distance from the centre of the cluster.

The cluster structure described above is implemented through a collection of classes. `Cluster`, `SubCluster`, and `Cell` classes contain the core mechanics including constructing the cluster from a seed location, constructing appropriate subclusters, storing cell energies and coordinates if they are within cluster bounds, and indexing and locating cells. Additional classes are used to help with some of the more complicated tasks; Figure 5.5 shows the classes and functions used, and the relationships between them.

To access cell cluster information for trigger analysis, the cluster classes need to be output to the ROOT [**?**] file generated by `R3L1Sim`. Initially this was done by

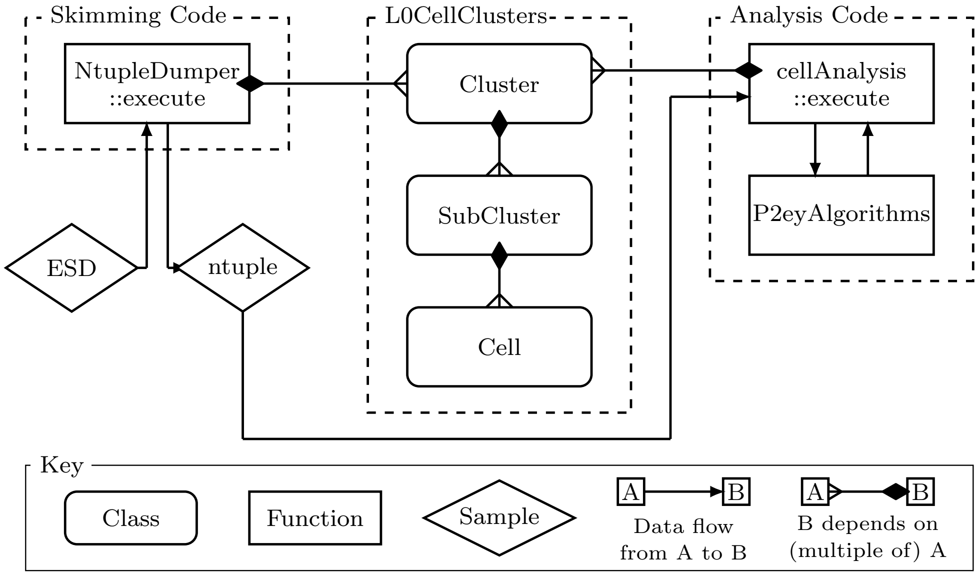


Figure 5.5: Classes, functions, and datasets used in the code, and the relationships between them. NtupleDumper represents the (modified) `R3L1Sim` software.

directly saving the `Cluster` class to the ROOT file, by providing dictionaries for each of the defined classes to enable the interpreter to interact with them. This worked at first when executed locally on test data but caused issues when run on the Grid, which is necessary to be able to process the large amounts of data required for the study.

After a large amount of troubleshooting, no cause could be found for these issues. This was solved by taking an alternative approach to saving the data. The class structure can be decomposed to a set of arrays (STL vectors in C++) containing all of the information associated with a cluster, its subclusters, and all contained cells. The classes have the capability to output this decomposed information, and also to be reconstructed from it.

Integrating these classes into `R3L1Sim`, and adding a few lines of code to the event loop to create and save clusters, provides the information needed for Phase-II studies with Layer-1 strip information. In future this code will likely be expanded to include more calorimeter layers.

5.5 $\mathrm{E_{ratio}}$ Algorithm Design

Performance studies so far have focused on the $\mathrm{E_{ratio}}$ variable described in Section ??. This work expands on previous studies which highlighted the potential of the $\mathrm{E_{ratio}}$ algorithm for the Phase-II trigger [?]. The focus here is on designing and tuning a version of the $\mathrm{E_{ratio}}$ that could be applied in firmware, as will be required when the upgrade is implemented. The algorithm, combined with the software described above, is then used to obtain estimates for the performance of the Phase-II e/γ trigger.

5.5.1 Initial algorithm

Calculating $\mathrm{E_{ratio}}$ is done for every cluster generated by the modified `R3L1Sim` software. An initial algorithm was defined, chosen to make each step as simple as possible for firmware implementation, before refinements are made to improve performance.

The cluster contains the location of the seed SuperCell that it receives from the Phase-I simulation. To find the seed for the $\mathrm{E_{ratio}}$ algorithm, the cells in the SuperCell are compared against each other to find the cell with the highest energy.

Once the seed cell is located, the algorithm searches for possible secondary maxima. This is done by stepping out one cell at a time away from the seed, and along each step calculating the energy gradient:

$$\Delta E = E^{\mathrm{cell}}_{\mathrm{next}} - E^{\mathrm{cell}}_{\mathrm{prev}},$$

where $E^{\mathrm{cell}}_{\mathrm{next}}$ is the energy of the cell being stepped to and $E^{\mathrm{cell}}_{\mathrm{prev}}$ is the energy of the cell being stepped from. Initially ΔE should be negative, as the seed is the highest energy cell in the cluster. At some point ΔE may become positive, indicating that a secondary maximum is being approached. If ΔE becomes negative again then

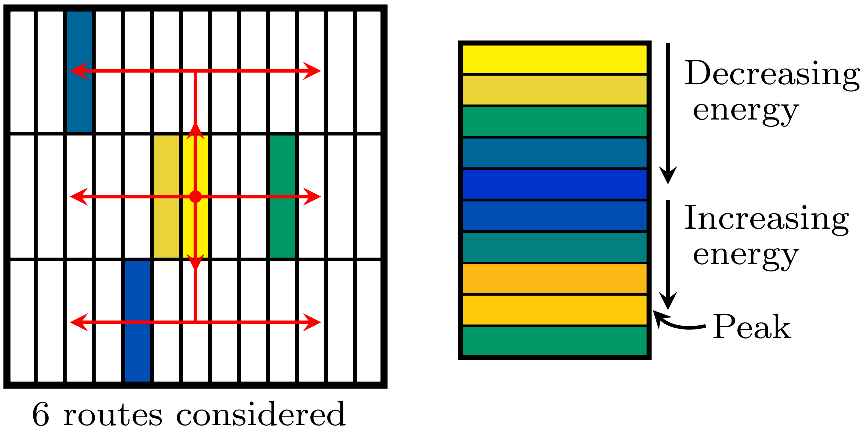


Figure 5.6: Depiction of the 6 different routes in which the $\mathrm{E_{ratio}}$ algorithm searches for secondary maxima (left) and how the algorithm identifies secondary maxima by tracking energy gradients along each step (right).

the cell before the change is marked as a candidate secondary maximum. If the algorithm reaches the edge of the cluster and ΔE is positive then the last cell is taken as a candidate instead.

This iterative process is done in six different routes through the cluster: with the same ϕ coordinate as the seed and one either side, each either in positive or negative η. If the route is one with a different ϕ coordinate to the seed then the first step is out in ϕ from the seed and all subsequent steps are in η. Figure 5.6 illustrates the different routes.

Each of the six routes can return a candidate secondary maximum. The largest of these six is taken to be the secondary maximum and used to calculate $\mathrm{E_{ratio}}$, as in Equation **??**, with the seed as the primary maximum.

Figure 5.7 shows the $\mathrm{E_{ratio}}$ distributions for the signal and background samples. The discriminating power of the variable is evident from the difference in shape between the two distributions, requiring candidates to have $\mathrm{E_{ratio}}$ in the lower end of the range clearly could reject a large amount of background and maintain much of the signal. This is quantified in the figure by showing the fraction of clusters that would be selected by the trigger for a given $\mathrm{E_{ratio}}$ threshold, for both signal and

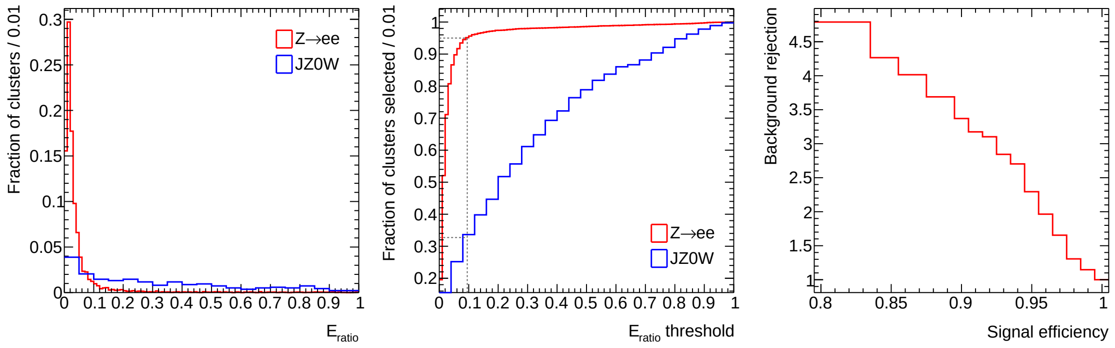


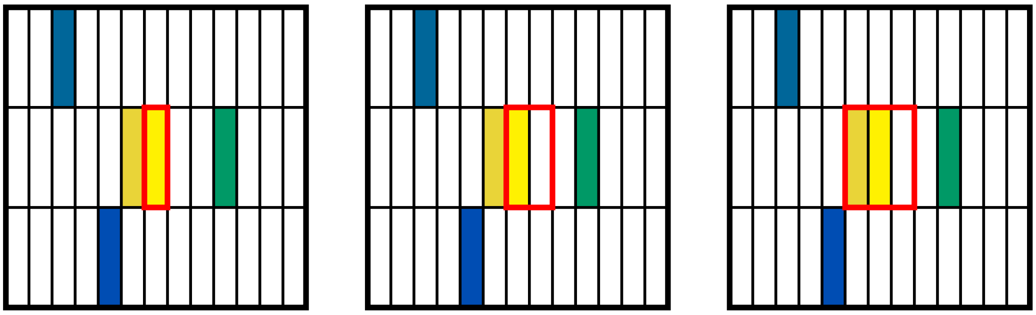
Figure 5.7: Normalised E_{ratio} distributions for clusters from signal and background samples (left); the integral of these distributions, showing what fraction of clusters pass a given E_{ratio} cut (middle); and background rejection obtained with the E_{ratio} cut corresponding to the given signal efficiency (right).

background. Any suitable E_{ratio} threshold is possible to implement in the trigger; the figure of interest is how much of the background can be rejected whilst maintaining a set fraction of signal events. The third plot in the figure shows the background rejection for a range of signal efficiencies. Background rejection is defined as the inverse of the fraction of background events passing the selection.

The primary figure of merit used to evaluate the performance of different E_{ratio} algorithms in this study is the background rejection at 95% signal efficiency. This initial algorithm achieves a background rejection of 2.3 at 95% signal efficiency.

5.5.2 Peak size

Several parameters were identified in the initial algorithm that could be adjusted to potentially improve the background rejection. The first of these parameters is the size of the 'peak' used to measure energies. In the first form of the algorithm energies were measured in single cells, which is equivalent to a peak size of one. With a larger peak each energy measurement is taken as the sum of the energies of the cell in question and its nearest neighbours in η, such that the number of cells summed over is equal to the peak size. Figure 5.8 shows how cells are selected for different peak sizes.



1 cell peak 2 cell peak 3 cell peak

Figure 5.8: Illustration of how adjacent cells are included in the energy measurement for a given cell. The horizontal axis represents η and the vertical axis ϕ. In all three cases shown the combined energy of the cells contained within the red box is considered to be the energy of the central, bright yellow, cell.

The $\mathrm{E_{ratio}}$ algorithm was applied to the signal and background samples with a range of different peak sizes. Figure 5.9 shows how changing the peak size impacts the background rejection of the algorithm. It is clear that in the region of interest, particularly at 95% signal efficiency, the original peak size of one cell gives the best background rejection, so no improvement has been identified.

5.5.3 Exclusion region

Another parameter investigated was the exclusion region. The exclusion region is defined as the distance in η from the seed in which secondary maxima are not considered distinct from the primary maximum. The initial algorithm has no exclusion region. An n-cell exclusion region is implemented by making the first comparison with the cell n cells in η away from the seed, regardless of ϕ coordinate of the route in question. Figure 5.10 shows the differences between different exclusion regions.

A range of exclusion regions were applied to the $\mathrm{E_{ratio}}$ algorithms to check their impact on background rejection. Figure 5.11 shows the background rejection with each variation of the algorithm. In this case improvements were seen over the initial algorithm, with a one cell exclusion region giving the best background rejection of

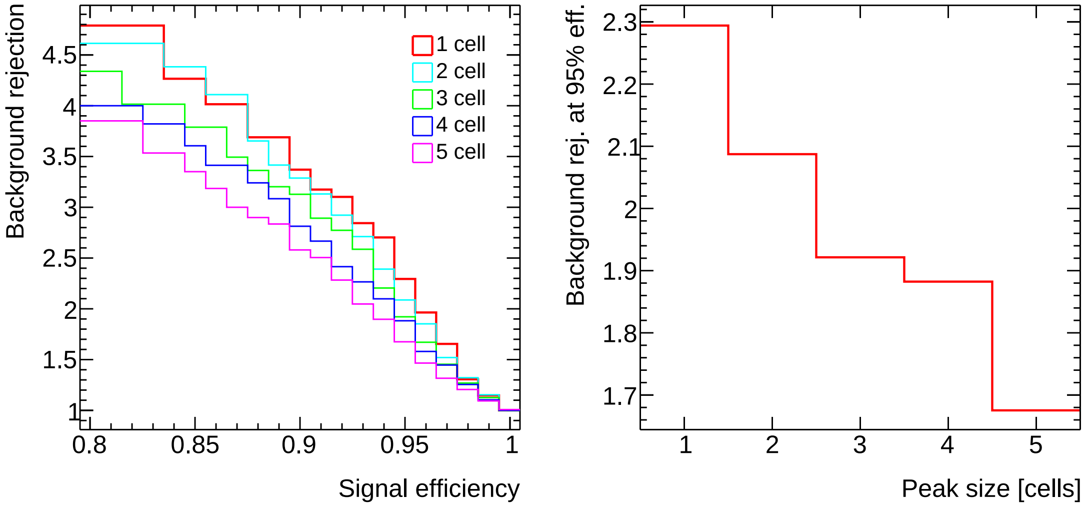


Figure 5.9: Background rejection for the E_{ratio} algorithm with different peak sizes. Left plot shows how background rejection varies with signal efficiency for each of the five peak sizes tested. Right plot gives how background rejection for a fixed 95% signal efficiency varies with the choice of peak size.

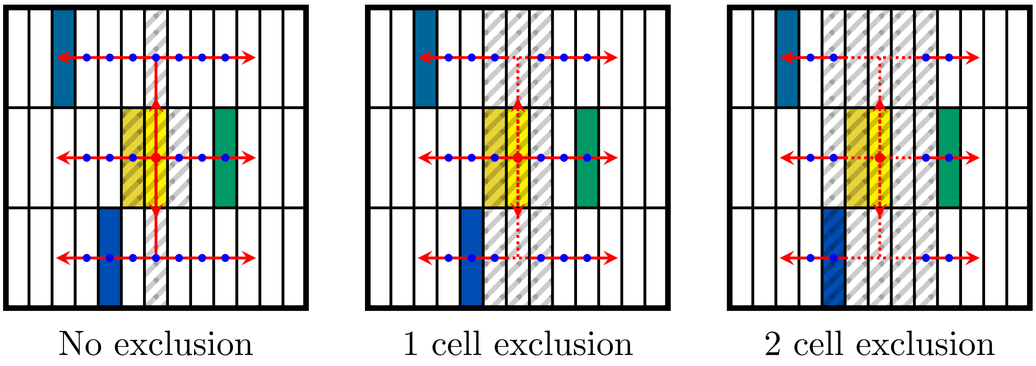


Figure 5.10: Illustration of how an exclusion region impacts the comparisons made in the E_{ratio} algorithm and where a secondary maximum can be found. Blue dots mark where energies are compared to calculate gradients. The first comparison on each route is always made against the seed cell. The shaded area shows where a secondary maximum cannot be found assuming that the seed is the highest energy cell in the region.

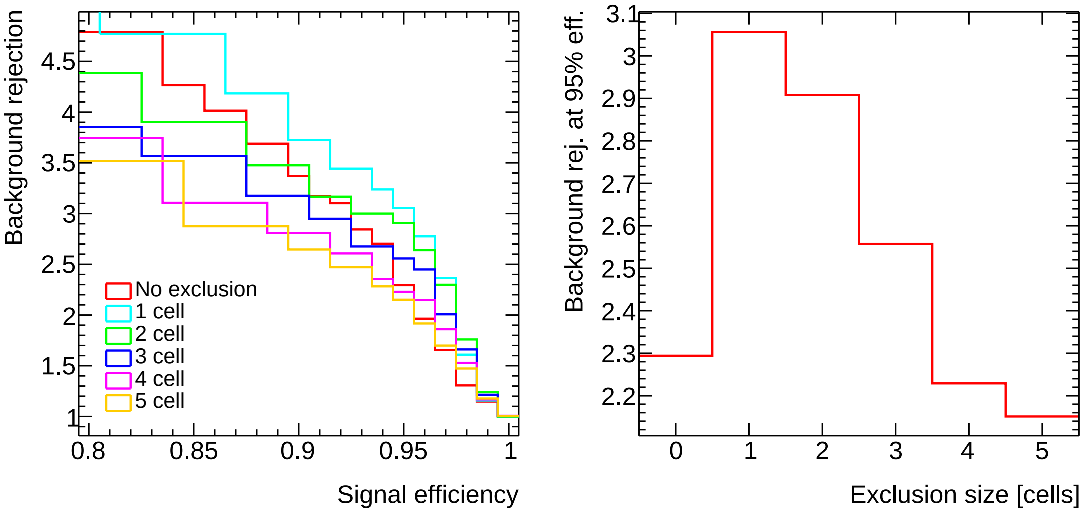


Figure 5.11: Background rejection for the E_{ratio} algorithm with different exclusion regions. Left plot shows how background rejection varies with signal efficiency for each of the five exclusion regions tested. Right plot gives how background rejection for a fixed 95% signal efficiency varies with the choice of exclusion region.

3.05 at 95% signal efficiency.

Given that cell widths vary significantly in different regions of the calorimeter, the exclusion region was investigated over a range of η values. Figure 5.12 compares the performance of the E_{ratio} algorithm with different exclusion regions against η. The one cell exclusion region performs best in each bin except the second. The second bin contains all clusters that touch the transition region of the calorimeter, where the algorithm is not expected to perform sensibly due to the lack of strips, so the drop in performance here is not an issue. This shows that the choice of exclusion region has no significant dependence on the strip width.

5.5.4 Search limit

The final parameter to have been investigated so far is the search limit. The search limit places an upper limit on the distance, in η, travelled from the seed in search of a secondary maximum. In the initial algorithm the search is only limited by the width of the clusters, this is equivalent to a search limit of 0.15.

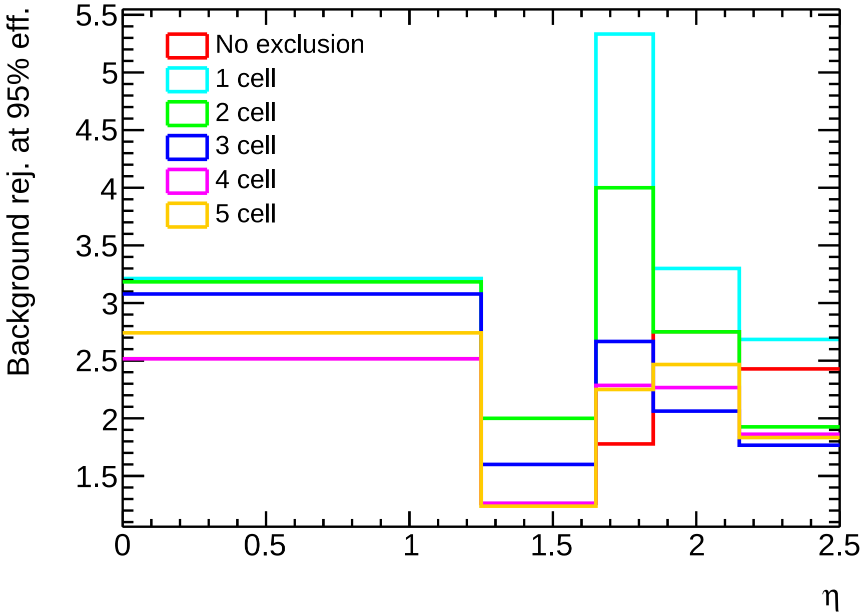


Figure 5.12: Background rejection at 95% signal efficiency versus η for E_{ratio} algorithms using each of the exclusion regions shown.

A range of search limits, from 0.15 down to 0.025, were implemented in the E_{ratio} algorithm to see if it could improve background rejection. Figure 5.13 hows how the search limit impacts background rejection. No improvement is found over the initial 0.15 search limit. There appears to be a plateau in performance, the maximum background rejection could still be obtained with a search limit of 0.1. This could prove useful if the current cluster size is too large to be implemented in firmware.

5.5.5 Algorithm summary

Of the three parameters investigated only the exclusion region gave an increase in performance over the initial algorithm. The recommended form of the algorithm is therefore one with a peak size of one cell, an exclusion region of one cell, and a search limit of 0.15 (or anywhere between 0.1 and 0.15).

There are still changes to be made to the current algorithm and simulation. Currently the Phase-II simulation relies on the Phase-I eFEX simulation to seed the algorithm. This is not representative of the hardware plans, discussed in Section

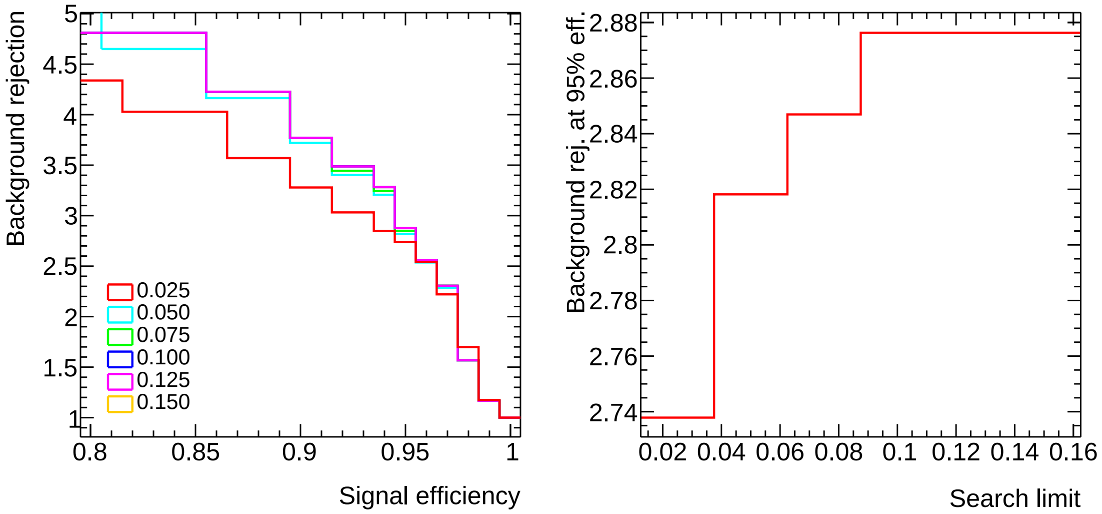


Figure 5.13: Background rejection for the E_{ratio} algorithm with different search limits Left plot shows how background rejection varies with signal efficiency for each of the six search limits tested. Right plot gives how background rejection for a fixed 95% signal efficiency varies with the choice of search limit.

??. The Phase-II simulation needs to be adapted to form seeds without input from eFEX. The studies presented here all use samples with 80 pileup collisions per bunch crossing; this should be increased to around 200 collisions per bunch crossing to properly represent expected Phase-II conditions.

Further studies could include investigating the impact of noise in the calorimeter energies on E_{ratio} performance. This could include adding a threshold that energy differences have to exceed to be considered a change in gradient.

CHAPTER 6

Search for vector-boson scattering production of a Z boson and a photon

CHAPTER 7

Search for triboson production of VZy through its semi-leptonic decay mode

CHAPTER 8

Conclusions

REFERENCES

APPENDIX A

FIRST APPENDIX

Tables of datapoints...