

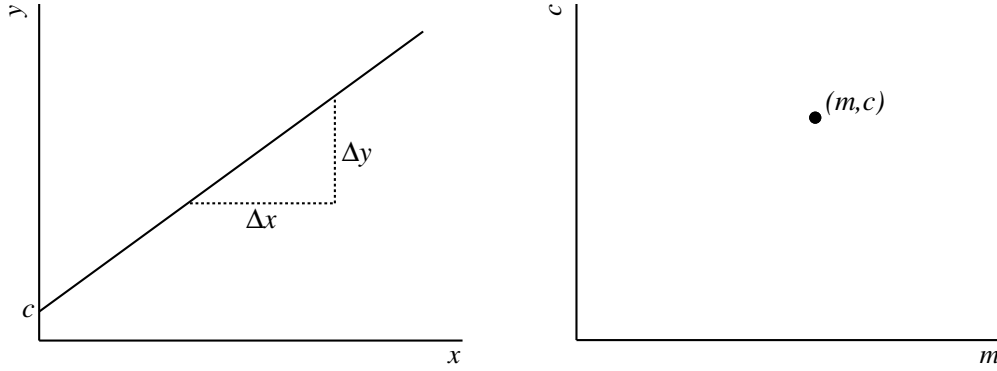
Chapter 1

Enhanced ECal reconstruction

The current implementation of the ECal reconstruction software was designed to reconstruct particles which originate from the ND280 Tracker and enter the ECal. As section ?? shows, this was realised by only considering reconstructed ECal clusters under the single track-like or shower-like hypothesis. It should be evident that a neutrino interaction occurring within the ECal does not well fit this topology. While it is true that there is some power in the current reconstruction to distinguish a neutrino interaction from an entering track or shower, there is little feature information available. How many final state particles propagated from the interaction? How much visible energy was deposited by each of the particles? Where in the ECal did the interaction occur? These basic questions can not be trivially answered when using the current reconstruction. To maximise the ability of distinguishing ECal neutrino interactions from entering backgrounds, the reconstruction must be revisited.

1.1 The Hough transform

The Hough transform is a popular method of machine pattern recognition used by, but is not limited to, high energy physics experiments. Originally designed for machine track recognition in bubble chamber pictures [1], the version most widely used throughout the world was developed in 1972 [2]. The Hough transform is used to isolate specific features or shapes from a digital image. The simplest implementation, which is of most interest in event reconstruction, allows the extraction of straight 2D lines from a complex pattern. This is achieved by exploitation of a simple, but remarkable, feature of 2D geometry.



(a) Line in 2D Cartesian space. The line is defined by its intercept with the y -axis, c , and its gradient, $\Delta y / \Delta x = m$ (not shown). (b) Point representation of the line defined in Fig. 1.1a in 2D parameter space. Because the line is solely defined by its intercept, c , and its gradient, m , the line is defined as a point in this parameter space.

Figure 1.1: Representations of a 2D line in Cartesian space.

1.1.1 Line-point duality

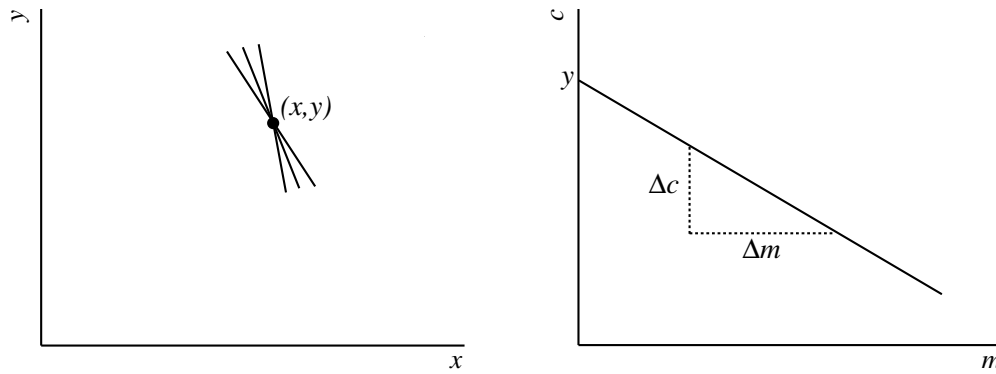
Consider a straight line formed in a 2D Cartesian space as shown in Fig. 1.1a. The line is usually described by

$$y = mx + c \quad (1.1)$$

where y and x are used as coordinates, m is the gradient of the line and c is the intercept location of the line with the y axis. While it is not necessary to analyse this simple shape in great detail, it is important to note that m and c are the only parameters necessary to completely describe the line.

Now consider a new 2D space where the axes are defined by m and c , rather than x and y (hereafter referred to as the parameter space). As this parameter space is described by the parameters of a general 2D Cartesian line, there is an underlying symmetry between the two spaces. The parameters of the 2D line shown in Fig. 1.1a can be used to form a pair of coordinates (m,c) in the parameter space as shown in Fig. 1.1b. It is important here to state clearly the general result; a straight line in Cartesian space is represented by a single point in parameter space.

Now consider the 2D Cartesian space again. Unlike before, we will define a sin-



(a) Point in 2D Cartesian space. The point can be defined as an infinite set of lines, all of which cross at said point (three example lines are shown).
 (b) Line representation of the point defined in Fig. 1.2a in 2D parameter space. The infinite set of lines which represent the point in Fig. 1.2a follow a relation, $c = -xm + y$, which itself is a line in the parameter space.

Figure 1.2: Representations of a 2D point in Cartesian space.

gle point rather than a straight line. Such a point is traditionally described by a pair of coordinates (x, y) . However, an alternative description of the point is an infinite number of lines all of which pass through (x, y) . This is highlighted by Fig. 1.2a where three lines of the infinite set are shown along with the point they represent. As the infinite line set are used to describe a single point, all lines in the set must follow a pattern. This relationship is revealed by simple algebraic manipulation of equation 1.1 to give

$$c = -xm + y. \quad (1.2)$$

Despite the manipulation, equation 1.2 still resembles the equation of a 2D line; however,¹ the parameters are x and y and the coordinates are m and c . Specifically, equation 1.2 is represented by a line in the parameter space defined above. The gradient of this line is

$$x = \frac{\Delta c}{\Delta m} \quad (1.3)$$

and the intercept of the line with the c axis is y as shown in Fig. 1.2b. As before, it is important to clearly state what has been shown; a point in Cartesian space is represented by a line in parameter space.

¹ADDRESSED - was “, however”; see <http://www.nationalpunctuationday.com/semicolon.html>

1.1.2 The parameter space

As section 1.1.1 shows, there is a clear relationship between the Cartesian space and the parameter space. Specifically, there is a symmetry between lines and points contained in the two spaces. This relationship between the Cartesian and parameter spaces is not only interesting but also very powerful. Consider again the parameter line defined by equation 1.2 and shown in Fig. 1.2b. As shown in section 1.1.1, equation 1.2 was derived by considering the infinite set of lines which pass through a Cartesian point. As this infinite set represents the parameter line, it must also be true that the parameter line represents the infinite line set. Using one of the results from section 1.1.1, any point along the parameter line represents one of the lines from our infinite set. This key feature of the parameter space is the central component of the Hough transform.

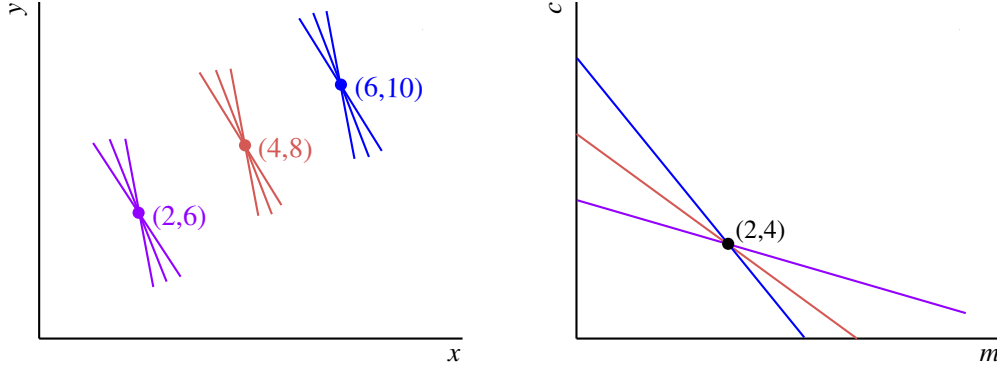
We will now return to the Cartesian space for an example of how the Hough transform works. Let's define three points in this space,

$$\begin{aligned} p_1 &: (2, 6) \\ p_2 &: (4, 8) \\ p_3 &: (6, 10). \end{aligned} \tag{1.4}$$

The three points defined in equation 1.4 are shown in Fig. 1.3a. Using one of the results from section 1.1.1 and equation 1.2, the three points can be Hough transformed into the following parameter lines:

$$\begin{aligned} c &= -2m + 6 \\ c &= -4m + 8 \\ c &= -6m + 10. \end{aligned} \tag{1.5}$$

The three parameter lines are shown in Fig. 1.3b. From Fig. 1.3b, it is clear that the three parameter lines all cross at a common point with parameter space coordinates (2,4). Using the results from section 1.1.1, this common point in parameter space is represented by a line in Cartesian space. In addition, as (2,4) is common to all three parameter lines, the Cartesian line represented by (2,4) must also pass through all



(a) The three Cartesian points defined in equation 1.4 along with example lines from each point's infinite line set. The coordinates of each point are also shown. (b) The three parameter lines defined in equation 1.5. The coordinates (2,4) define the point of intersection of the three lines.

Figure 1.3: The three points defined in equation 1.4 and their representation in the parameter space. The colour coding matches the Cartesian points to their respective parameter lines.

three Cartesian points defined by equation 1.4. This new Cartesian line is defined by

$$y = 2x + 4 \quad (1.6)$$

and is shown in Fig. 1.4 with the original points used to generate the parameter lines. While this example is relatively simple, it demonstrates the capability of the Hough transform to recognise linear patterns in sets of points.

1.1.3 Redefinition of parameters

Unfortunately, a complication in computation arises when $m \rightarrow \infty$. However, this complication can be removed by redefining the line parameters such that one is bounded. An alternative 2D line parameterisation is to specify a line in terms of the angle it makes with the x axis, θ , and the perpendicular distance of the line from the origin, ρ , as illustrated in Fig. 1.5. The functional form of the Cartesian line becomes

$$y = x \tan \theta + \frac{\rho}{\cos \theta}. \quad (1.7)$$

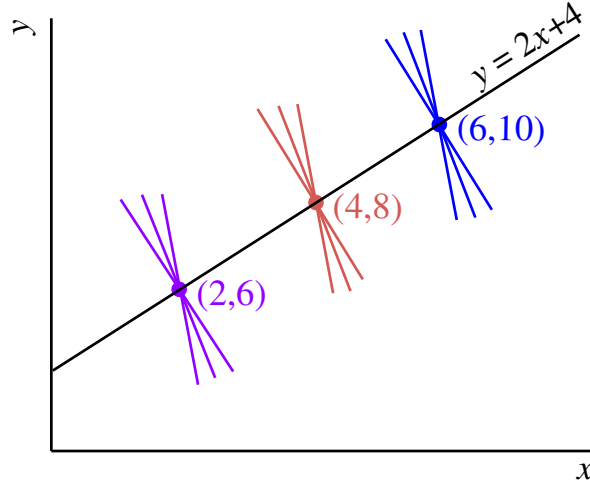


Figure 1.4: The line represented by the parameter space intersection in Fig. 1.3b with the Cartesian points it intercepts.

Using this new parameterisation, we must also define a new parameter space with axes θ and ρ . Using equation 1.7, a line in this new parameter space is defined by

$$\rho = y\cos\theta - x\sin\theta. \quad (1.8)$$

By definition, the θ axis of the parameter space must be bounded to the interval $[0, 2\pi]$. If the directionality of the line is meaningless to the analyser, then the interval can be restricted to $[0, \pi]$.

The disadvantage of this parameterisation is that parameter line generation now involves trigonometric calculations which can be computationally expensive. However, this problem is small when compared to the unbounded complication of the traditional parameterisation.

1.1.4 Discretisation of the parameter space

It must now be considered how the parameter space is analysed. When a large number of parameter lines are generated, it becomes computationally expensive to analyse the resultant parameter space. While approaches exist to analyse the parameter space with very high precision [3], it is often only necessary to extract parameters with finite resolution. In such a case, it is convenient to discretise the parameter space. Under this

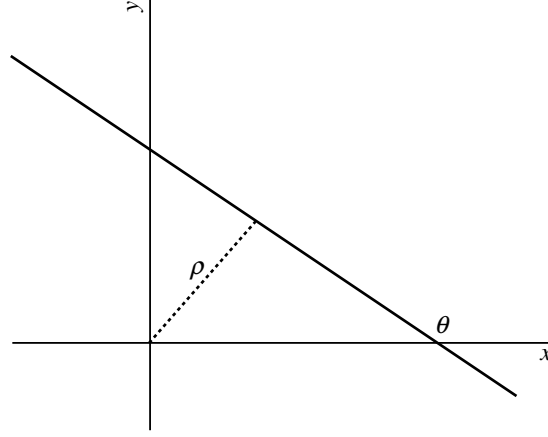


Figure 1.5: $\theta - \rho$ parameterisation of a 2D line.

regime, the parameter space is split into $\theta - \rho$ bins. Then, a parameter line is generated by incrementing the value of each $\theta - \rho$ bin it passes through. After each line has been added to the parameter space, the crossing locations can be readily found by searching for the $\theta - \rho$ bins with content larger than unity. An example of this discretisation is illustrated in Fig. 1.6, where the parameter lines defined in equation 1.5 have been re-parameterised using equation 1.8. The content of each bin in Fig. 1.6 records how many of the three parameter lines pass through each bin. The bin with value 3 is the crossing point of the three parameter lines.

If a discretised approach is acceptable, which is the case in event reconstruction, construction and analysis of the parameter space is reduced to filling a 1D array N times, where N is the number of parameter lines, followed by a 1D grid search of the array to find the bin with the highest content.

1.2 ECal application of the Hough transform

We must now address how the Hough transform can be used as a reconstruction tool in the ECal. To do this, let's consider a neutrino interaction which occurs in the ECal as illustrated in 1.7. While the propagating neutrino is invisible to the ECal, the charged final state are definitely not. To first order, the final state particles propagate in straight lines depositing energy in the scintillator bars as they go. From this, we can infer that

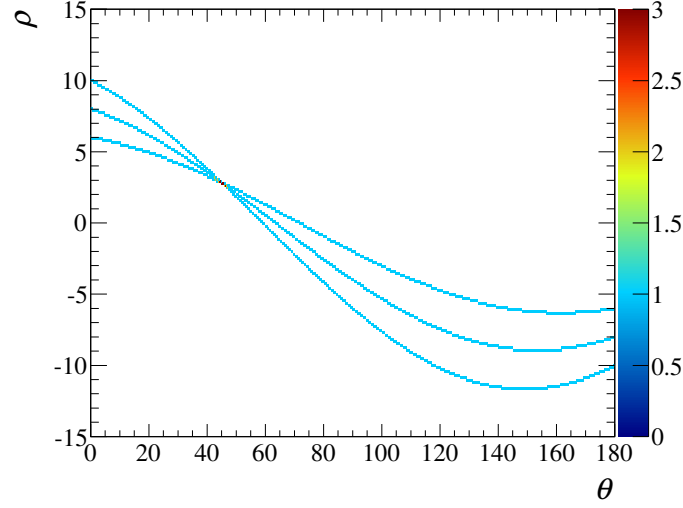


Figure 1.6: The discrete $\theta - \rho$ space. The plotted lines are those defined in equation 1.5 and reparameterised using equation 1.8.

the hit bars arranged in straight lines should reveal the trajectory of the final states. As shown above, the Hough transform is capable of identifying straight lines from a set of coordinates. However, there are two complications in the ECal which the above sections have not addressed. We have only specifically discussed how to extract a single straight line from a pattern. As Fig. 1.7 shows, the number of final states can be, and is often, greater than one. This is merely a problem of computation which will be addressed in section 1.2.3. A much more severe problem is that the above demonstrations only deal with patterns constructed from infinitesimal points. While the centre of a scintillator bar can be used as a point for parameter line generation, it is unlikely that a final state particle will pass through the central point of the scintillator bars that it propagates through. If this is not addressed, the Hough transform will be of little use in trajectory reconstruction.

1.2.1 Modelling the ECal bar

To make the Hough transform viable as a reconstruction tool, the finite dimensions of the ECal bar need to be incorporated into the parameter space generation. This feature of the ECal bars would be very problematic if the parameter space was continuous. However, it is only necessary to know the line parameters with finite resolution and a discrete parameter space can be used. This means that the ECal bar can be modelled as a set of Cartesian points and each of said points can be Hough transformed in turn

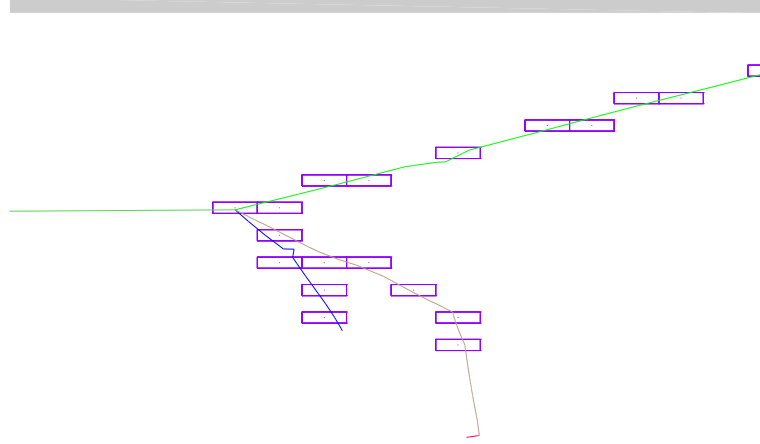


Figure 1.7: Simulated neutrino interaction with 3 final states in the side-right ECal. The green line entering from the left is the ν_μ . The green line exiting to the right is a μ^- , the brown line is a π^+ and the blue line is a proton. The purple rectangles represent the hit ECal bars.

to build up the parameter line representation of the ECal bar.

There are now two steps to consider. Firstly, how should the points be arranged? Remember that the Hough transform of a point represents all of the lines that pass through that point. So, the points should be arranged in such a fashion that any line which passes through the 2D cross-section of the ECal bar also has to pass through one of the points in the configuration. Secondly, the spacings between the points should be small enough that no gaps appear in the generated parameter line.

Assuming that every bin of the parameter space will have a $1^\circ \times 1$ mm area, an obvious choice would be to use a rectangular grid of points with 1 mm spacing superimposed over the 2D cross-section of an ECal bar. The total number of points used to model the ECal bar is 451. To Hough transform the ECal bar, every point in the grid array can be Hough transformed individually with care taken to ensure that each $\theta - \rho$ bin is filled exactly once. The result is illustrated in Fig. 1.8b. The finite size of the ECal bar is evident by the finite size of the resultant parameter line.

While the generated parameter line accurately represents every line which passes through the ECal bar, there are two problems with this approach. Firstly, the large number of points to be Hough transformed is very large which results in a long CPU time. Secondly, there is a very high number of redundant calculations involved in

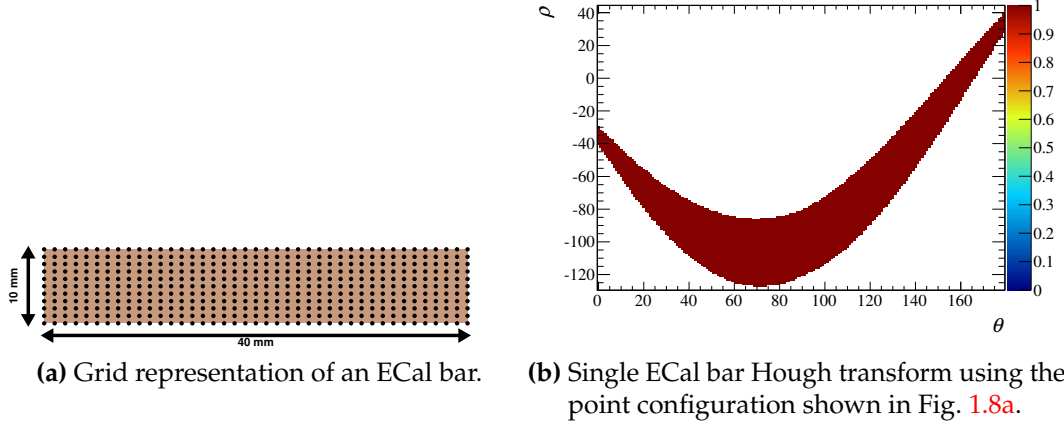


Figure 1.8: The grid representation of an ECal bar and its representation in parameter space.

the parameter line generation. Consider an exactly vertical line which passes through one of the points in the grid array. This line also passes through 10 other points in the same column of the grid. This means that when the parameter line is being generated, this vertical line is calculated 11 times for each column. Bearing this in mind, there are many points along the parameter line which are repeatedly calculated and provide no extra information. This would mean that any algorithm which uses this approach would be very CPU inefficient.

An alternative is to model the ECal bar as a set of points arranged in a cross as shown in Fig. 1.9a. Assuming that the spacing between the points on each line of the cross is infinitesimal, any line which passes through the ECal bar would also have to pass through one of the points in the configuration. As the parameter space is discrete, the spacing between the points need not be infinitesimal but only small enough to ensure that no gaps appear in the parameter line. Using 45 points on each line of the cross, the ECal bar can be Hough transformed by Hough transforming each point in the cross configuration. An example of this result is shown in Fig. 1.9b using the same ECal bar used to generate Fig. 1.8b. Clearly, Fig. 1.8b and Fig. 1.9b are identical showing that the cross model achieves the same result as the grid model. Comparing the two, the cross model uses a 90 point representation whereas the grid model uses a 451 point representation. This should mean that an algorithm utilising the cross model would be a factor of five faster than one using a grid model.

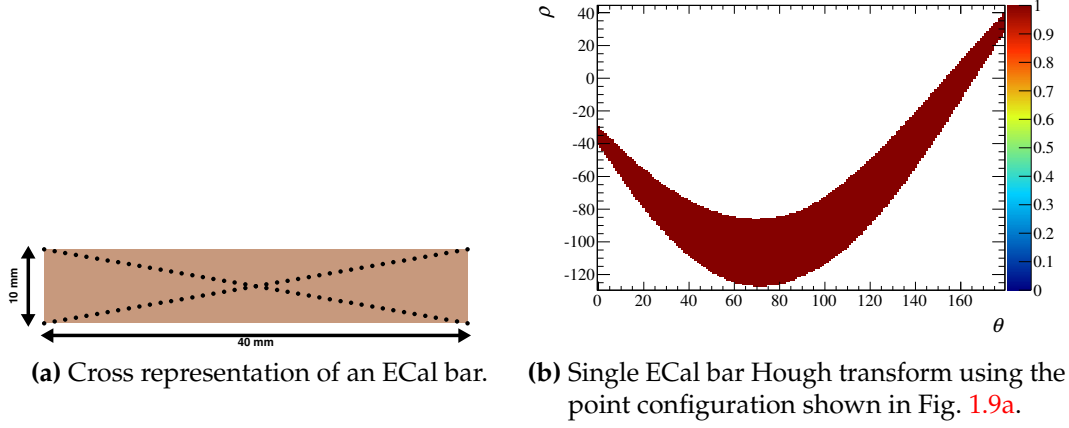


Figure 1.9: The cross representation of an ECal bar and its representation in parameter space.

1.2.2 Parameter space generation

As we have addressed how to Hough transform an ECal bar, we are now in suitable position to generate the full parameter space for an ECal cluster. As described in section 1.1.3, the parameterisation of the 2D lines requires some point in space to act as the origin. It is possible to use the origin defined by the global ND280 geometry; however,² this is located in TPC 1 which would mean ρ will usually be the order of metres. It is more convenient to define an origin in the vicinity of the ECal cluster being reconstructed. A simple option is to use the charge-weighted centre of the ECal cluster as the origin of the Hough transform. This location is simple to calculate and generally keeps ρ small.

The provided description of the Hough transform in all previous sections is strictly defined in 2D and so the ECal cluster should be split in such a way that this definition can be used. Fortunately, a 3D ECal cluster is built up using the two 2D views that the scintillator layers provide. So, it is relatively easy to split the 3D cluster into a pair of 2D clusters by collecting the cluster's hits into their respective 2D views.

We can now partly answer one of the problems raised in section 1.2 which is how to handle the track multiplicity aspect of the reconstruction? This is partly addressed by generating N parameter spaces with the same $\theta - \rho$ bin configuration where N is the number of hits in the 2D cluster. Each of the N parameter spaces will hold one parameter line generated by one of the 2D hits (in a similar fashion to Fig. 1.9b).

²ADDRESSED - was “, however”

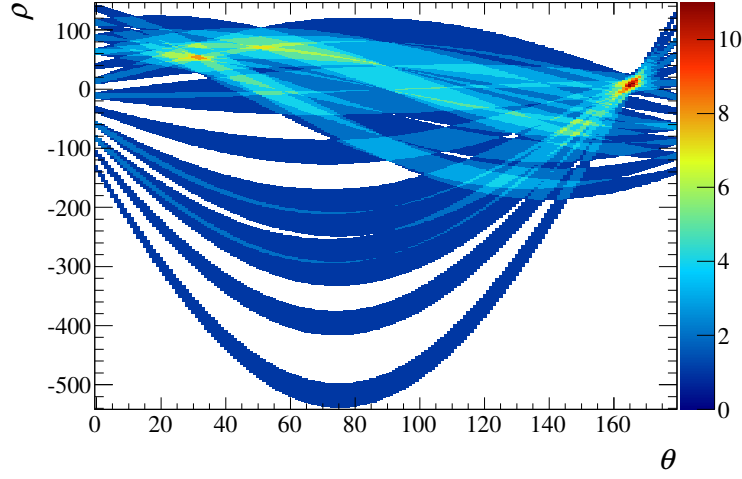


Figure 1.10: The full parameter space of the 2D cluster shown in Fig. 1.7. The height of each point in the parameter space corresponds to the number of hit scintillator bars intersected by the Cartesian line which the parameter space point represents.

The final parameter space can then be generated by adding together each of the N parameter spaces. The parameter space of the 2D cluster in Fig 1.7 is shown in Fig. 1.10.

1.2.3 Parameter space analysis

The full parameter space can look arbitrarily complicated. However, it contains a vast amount of trajectory related information about the cluster. Every $\theta - \rho$ bin of the parameter space describes a 2D track and the content value of said bin describes how many 2D ECal hits the track passes through. As described in section 1.2, a particle's trajectory should be straight in the ECal which means that the particles path should be revealed by finding the most hits arranged in a line. This hit arrangement can be found by finding the bin in the full parameter space with the highest value. The track candidate parameters can be found by fetching the (θ, ρ) coordinate of the found bin.

While the preferred bin can reveal how many hits the track candidate passed through, it contains no information about which hits were contributors. However, this full parameter space was generated by summing the N parameter spaces discussed in section 1.2.2. So, the contributing hits can be readily found by looking at the same (θ, ρ) bin in each of the N parameter spaces and recording which have a non-zero value. We now have the track candidate's parameters and its contributing hits which is enough

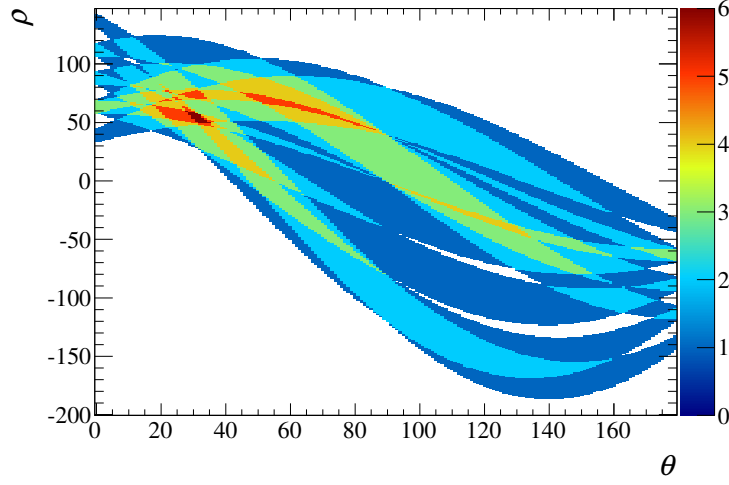


Figure 1.11: The reduced parameter space of the 2D cluster shown in Fig. 1.7. The reduced parameter space was formed by removing all of the hit scintillator bar representations which contributed to the full parameter space's (see Fig. 1.10) maximum.

to describe the 2D trajectory.

A new search now needs to begin to find any other track candidates. However, repeating the same search of the full parameter space will return the track candidate that has already been found. To find the next track candidate, the presence of the previous track candidate must be removed. So, a reduced parameter space must be generated. The previous step found which of the N parameter lines contributed to the previous track candidate. So, this new parameter space can be formed by subtracting the contributing parameter lines from the full space. An example of this is shown in Fig. 1.11 where the reduced parameter space was formed by subtracting the contributing parameter lines to the highest bin in Fig. 1.10. The next track candidate can then be found by searching for the highest content bin of this reduced parameter space and said bin's contributing parameter lines.

This process can be repeated until some threshold is reached. This threshold is nominally set by demanding that at least three hits are required to form a track candidate.

1.2.4 2D track quality checks

While the approach outlined above is very powerful for recognising track-like shapes in the 2D ECal clusters, it is not capable of checking whether the selected track candidates are of sound quality. So, external checks need to be done which validate each track as it is returned from the parameter space. Fortunately, the objects returned from parameter space are simple in structure and so the quality checks can be designed to reflect this. Two necessary checks were implemented in the 2D reconstruction:

- The track can not skip a scintillator layer in a given view.
- The track can skip a maximum of 1 bar in a scintillator layer.

If a track fails either of the above conditions, the track is flagged as bad and rejected. To ensure that the same bad track is not selected in the next interaction of the parameter space analysis, the track also needs removing from the parameter space. To do this, every bin in the parameter space is checked to see if the bin is filled purely from the hits contained in the bad track. If this is the case, the bin content is set to 0.

1.2.5 3D track reconstruction

Section 1.2.3 describes the track reconstruction of a 2D ECal cluster. However, a 3D cluster consists of two sets of 2D clusters. So, the process described in section 1.2.3 must be performed on each of the 2D clusters. The result of this process is two sets of 2D tracks. To form full 3D tracks, the tracks from each view must now be matched together. This is achieved by making every pairwise comparison of the tracks from each view to find the pair which are most similar to each other. After such a pair is found, the tracks are removed from the pool and the process is repeated to find the next pair until either no tracks are left or one 2D track is left. In the latter case the single 2D track is discarded. Every pairwise combination of tracks is used to form a likelihood \mathcal{L} . The pair which produces the highest \mathcal{L} is declared the best match and removed from the pool. Three pieces of information about the matching pair are used to calculate \mathcal{L} , all of which make use of probability density distributions generated using beam Monte Carlo.

As should be expected, a vertex with one visible track will have different characteristics to a vertex with three visible tracks. So, to maximise the ability of the matcher,

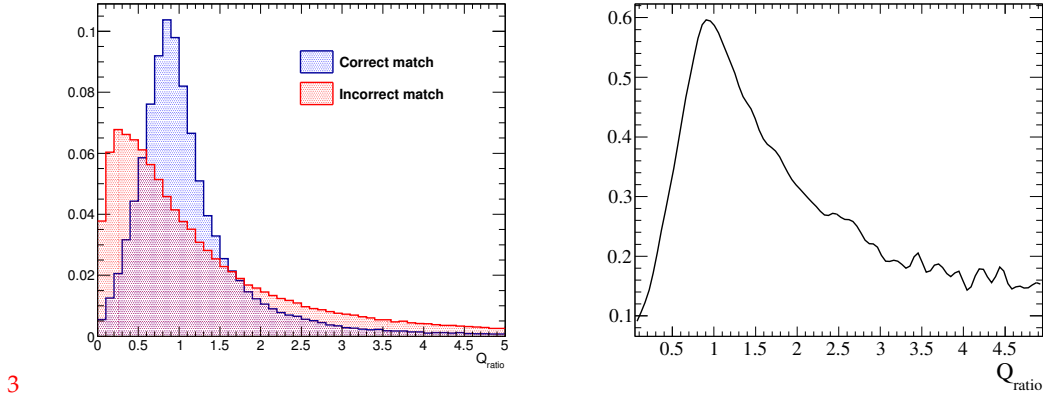
a different set of probability density distributions are used for the 1, 2 and 3 track cases. If the number of tracks in each view is not identical, the higher number of tracks is used to find the correct probability density distributions. Separately, due to their geometrical differences, the reconstructed shape of vertices in the DS ECal will differ to those in the barrel ECals. This leads to a separate set of probability density distributions for the barrel and DS ECal modules.

The first input to the likelihood is the ratio of the total deposited charge on each track, Q_{ratio} . The denominator is taken as the track which comes from the view with the most hits. Generally speaking, a particle propagating through an ECal module should deposit a similar amount of charge in each of the two views. So, Q_{ratio} should have a value close to 1 if the two 2D tracks are created by the same particle. An example of the Q_{ratio} distribution is shown in Fig. 1.12a, taken from beam Monte Carlo in the barrel ECals for cases where the maximum number of tracks found in a given view is 2. In Fig. 1.12a, correctly matched (in blue) shows the Q_{ratio} distribution for matching pairs which come from the same particle and incorrectly matched (in red) shows the Q_{ratio} distribution for matching pairs which were created by different particles. As Fig. 1.12a shows, Q_{ratio} well separates the two cases. To generate a probability density distribution for Q_{ratio} , the two distributions shown in Fig. 1.12a are used, but without applying any normalisation. By comparing the bins of each distribution, the probability for correctly matching two tracks in a given bin p_i can be formed by

$$p_i = \frac{s_i}{s_i + b_i}, \quad (1.9)$$

where s_i is the number of correctly matched tracks in bin i and b_i is the number of incorrectly matched tracks in bin i . A discrete probability density distribution for Q_{ratio} can then be formed by calculating p_i for every bin. The discrete probability density distribution is then interpolated with splines to create the final probability distribution. An example of this for the two track, barrel case is shown in Fig. 1.12b. When a matching candidate pair is being considered, the value of Q_{ratio} is calculated and used in the spline to retrieve $\mathcal{L}_{Q_{\text{ratio}}}$.

The second input to the likelihood, called $\Delta_{\text{layer, first}}$, is the difference in the starting layer of each 2D track which forms the matching candidate pair, where starting layer refers to the layer closest to the ND280 Tracker. For 2D tracks which should be matched together, $\Delta_{\text{layer, first}}$ should be 1. The separation ability of this variable for the



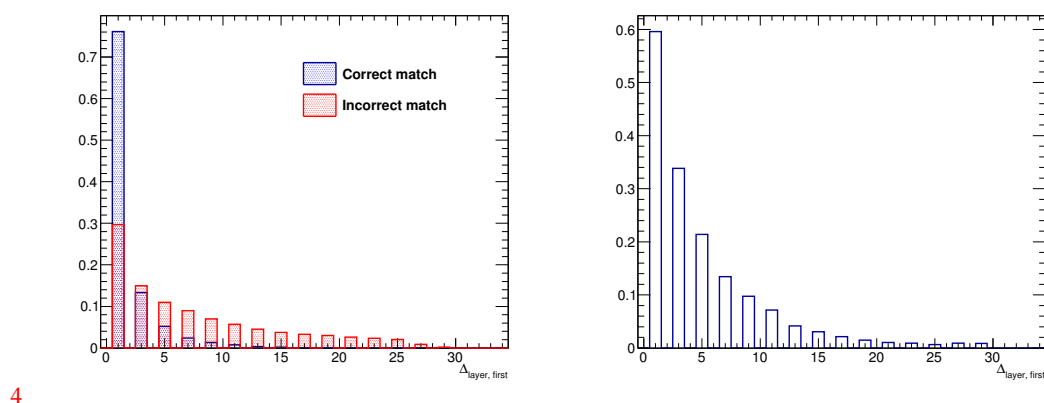
(a) Q_{ratio} distribution (area normalised). The blue and red distributions refers to matching pairs which were matched to the same true particle and different true particles respectively. The two distributions peak at different values of Q_{ratio} , which implies Q_{ratio} provides good separation of the correctly and incorrectly matched events.

(b) Q_{ratio} probability density distribution. The peak of the distribution aligns well with the correct matches peak in Fig. 1.12a which aides in maximising the number of correct matches made.

Figure 1.12: Q_{ratio} and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.

two track, barrel is shown in Fig. 1.13a. The discrete probability density function was created using eqn. 1.9. It was not necessary to interpolate using splines as $\Delta_{\text{layer, first}}$ is itself discrete. The probability density function for $\Delta_{\text{layer, first}}$ is shown in Fig. 1.13b for the two track, barrel case. For each matching candidate pair, the value of $\Delta_{\text{layer, first}}$ is calculated and the corresponding $\mathcal{L}_{\Delta_{\text{layer, first}}}$ is retrieved from the probability density function.

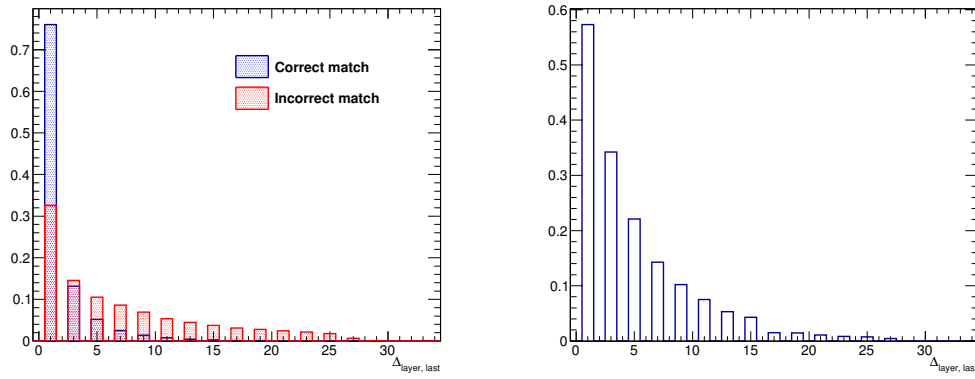
The third and final input to the likelihood, called $\Delta_{\text{layer, last}}$, is the difference in the ending layer of each 2D track which forms the matching candidate pair, where the ending layer refers to the layer furthest from the ND280 Tracker. Functionally, how this function is used is essentially identical to $\Delta_{\text{layer, last}}$ so it will not be described in detail. The separation ability of this variable and its corresponding probability density function for the two track, barrel case are shown in Fig. 1.14a and Fig. 1.14b respectively.



4

- (a) $\Delta_{\text{layer, first}}$ distribution (area normalised) (b) $\Delta_{\text{layer, first}}$ probability density distribution. The blue and red distributions refer to matching pairs which were matched to the same true particle and different true particles respectively. While both distributions peak in the same place, the peak for the correct matches distribution is much sharper, indicating a good level of separation between the correctly and incorrectly matched events.
- The peak of the distribution aligns well with the correct matches peak in Fig. 1.13a which aids in maximising the number of correct matches made.

Figure 1.13: $\Delta_{\text{layer, first}}$ and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.



(a) $\Delta_{\text{layer, last}}$ distribution (area normalised) (b) $\Delta_{\text{layer, last}}$ probability density distribution. The blue and red distributions refers to matching pairs which were matched to the same true particle and different true particles respectively. While both distributions peak in the same place, the peak for the correct matches distribution is much sharper, indicating a good level of separation between the correctly and incorrectly matches events.

The peak of the distribution aligns well with the correct matches peak in Fig. 1.14a which aides in maximising the number of correct matches made.

Figure 1.14: $\Delta_{\text{layer, last}}$ and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.

The matching likelihood, \mathcal{L} , for a matching candidate pair is then

$$\mathcal{L} = \mathcal{L}_{Q_{\text{ratio}}} \times \mathcal{L}_{\Delta_{\text{layer, first}}} \times \mathcal{L}_{\Delta_{\text{layer, last}}}. \quad (1.10)$$

As described above, \mathcal{L} is calculated for every matching candidate pair and the pair which maximise \mathcal{L} is selected as a match and removed from the pool. The process is then repeated until no more matches can be made.

3D tracks have now been formed, but the associated directions and positions of those tracks still need to be calculated. The track fitting process for the newly formed 3D tracks is very similar to that described in section ???. The tracks are briefly separated into their constituent 2D views and a charge-weighted average position of each layer is calculated using the track's constituent hits. Then, the hits in the opposing view are used to estimate the third coordinate of a given layer using a least-squares fit. After all of the coordinates have been estimated, a full 3D least-squares fit of the positions in each layer is performed to estimate the 3D track's direction and position in that ECal layer.

1.2.6 Track splitting

During development of the reconstruction, it became clear that a certain topology had been overlooked. An example of this is shown in Fig. 1.15 which shows a MC neutrino interaction in the ECal with 3 charged final states. As can be seen from Fig. 1.15, the muon (solid green line) is emitted back-to-back with one of the final-state protons (solid blue line) in this ECal view. Because the Hough transform implementation only concerns itself with straight lines, the result is that such back-to-back trajectories are typically reconstructed as a single track. However, there are two views available for every ECal module and the back-to-back emittance typically only appears in the ECal view which is perpendicular to the beam direction (the XY view). So, the correctly reconstructed tracks in the other ECal view can be used to split the tracks in the problematic view.

Consider a neutrino interaction with two charged final-states which has been reconstructed as one track in one view (called the merged view) but reconstructed as two tracks in the other view (called the other view). It should be expected that the single track in the merged view is not a particularly good match for either of the tracks in

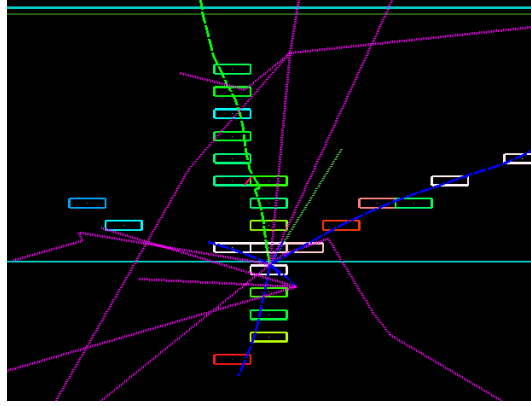


Figure 1.15: Event display (XY view) of a problematic (for the reconstruction) neutrino interaction in the ECal. The solid green track is the muon, the solid blue tracks are protons and the pink tracks are neutrons. The neutrino is the short dashed green line. The proton travelling vertically downwards and the muon are created almost back to back leaving a line of hit scintillator bars. The 2D hough transform would register this as a single straight line.

the other view. However, if the two tracks in the other view were temporarily merged together and this new track was compared to the single track in the merged view, one should expect this match to return a much higher value of \mathcal{L} . This feature can be used to identify a potentially merged track in a given view.

This motivated an extension to the 3D matching aspect of the reconstruction. As a reminder, the 3D matching algorithm makes every pairwise comparison of tracks from each view to find the pair which maximise \mathcal{L} . This matching routine was modified to also include temporary mergings of every pairwise combination of tracks in a single view. These temporary mergings can then be compared to every single track in the other view. As an example, consider a situation where 3 tracks (labelled A , B and C) have been reconstructed in one view and two tracks (labelled Y and Z) have been reconstructed in the other view. With the old method, the following comparisons

would be made:

$$\begin{aligned}
 A &\longleftrightarrow Y \\
 A &\longleftrightarrow Z \\
 B &\longleftrightarrow Y \\
 B &\longleftrightarrow Z \\
 C &\longleftrightarrow Y \\
 C &\longleftrightarrow Z
 \end{aligned} \tag{1.11}$$

With the new method, which includes temporarily merged pairs of tracks in a given view, the following comparisons would be made:

$$\begin{aligned}
 A &\longleftrightarrow Y \\
 A &\longleftrightarrow Z \\
 B &\longleftrightarrow Y \\
 B &\longleftrightarrow Z \\
 C &\longleftrightarrow Y \\
 C &\longleftrightarrow Z \\
 A + B &\longleftrightarrow Y \\
 A + B &\longleftrightarrow Z \\
 A + C &\longleftrightarrow Y \\
 A + C &\longleftrightarrow Z \\
 B + C &\longleftrightarrow Y \\
 B + C &\longleftrightarrow Z \\
 A &\longleftrightarrow Y + Z \\
 B &\longleftrightarrow Y + Z \\
 C &\longleftrightarrow Y + Z
 \end{aligned} \tag{1.12}$$

In terms of the matching comparisons, the temporarily merged comparisons are treated on the same footing as the single track comparisons; \mathcal{L} is calculated for each comparison and the one which maximises \mathcal{L} is selected. However, the post-matching treatment of the best match depends on whether a temporary merge is involved. If the match which maximises \mathcal{L} involves two single tracks then the treatment is as before; they are removed from the pool and the process is repeated. If a temporarily

merged pair are involved, the 2D crossing location of the temporarily merged tracks is calculated and is then used to split the single track in the other view. The original track is removed from the pool and replaced by the two single tracks formed from the split. The matching process is then begun again and repeated until all matches have been made.

1.2.7 Track pairwise crossing reconstruction

The final step of the reconstruction is to estimate where each of the 3D track's paths cross. As each track is reconstructed as a straight line in 3D, the final step is fairly simple. Using the track direction and position information calculated at the end of section 1.2.5, the position at closest approach for every pairwise combination of 3D tracks is calculated analytically. The distance of closest approach is also calculated. Six hits, the closest three from each 3D track, are then associated to the pairwise crossing.

1.3 Output of the reconstruction

The reconstruction is run over every 3D cluster found in the ECal. By applying the steps outlined above, for each 3D cluster the following output is given:

- A set of 3D tracks
- The pairwise crossings of all 3D tracks found in the cluster

Note that no vertex formation beyond the pairwise crossings is calculated at this stage, nor is any analysis of the 3D tracks performed. While this may seem like an oversight of the reconstruction, this approach was decided as no assumptions are made about what the tracks/crossings represent at this point, making the output more generic. Any analysis which wants to make use of the reconstruction is given enough information to apply more targeted reconstruction downstream.

1.4 Validation of the reconstruction

Because of the large scope of the reconstruction and the limited time available for the presented analysis, the validation of the reconstruction was done in parallel to the rest

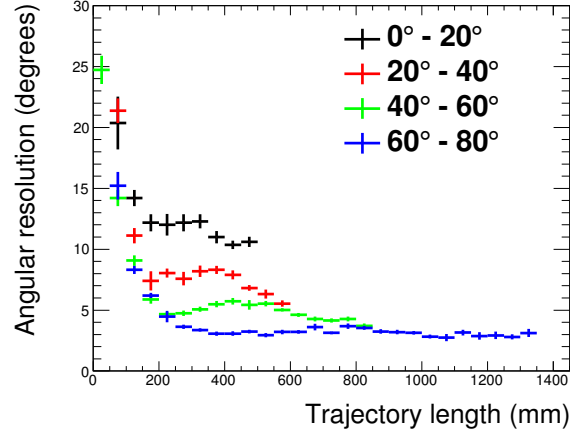


Figure 1.16: The angular resolution as a function of trajectory length in the DS ECal when applying the enhanced reconstruction to muon MC. The colour coding refers to the true entry angle range of the muons. The figure was generated using single muon particle gun fired into the front face of the DS ECal with a controlled range of entry angles. The angular resolution distributions were created by L. Pickering.

of the analysis and is still an ongoing effort. The validation that has been done can be split into two areas: validation of the performance of the algorithms purely using MC and comparisons of MC to data using control samples.

1.4.1 Validation of algorithm performance

The first performance validation investigated the angular resolution using the enhanced reconstruction (by L. Pickering). This study calculated the angular resolution for MC muons fired into the side-left ECal for a range of entry angles. For each MC event, the cosine of the angular separation between the true particle angle and the reconstructed angle was calculated, $\cos \theta^{\text{Sep}}$. The values of $\cos \theta^{\text{Sep}}$ were then binned in a distribution. An outward scan from the peak of the distribution was then performed to find where the height decreased to 68% of the peak. The value of θ^{Sep} and this point was taken as the angular resolution. These results are shown in Fig. 1.16. Generally speaking, the found angular resolutions are very good. For long trajectories (200 mm), the angular resolution is within 15° . It is only for short tracks (40 mm) that the angular resolution becomes large.

The second performance validation studied the success of the 3D matching. To do this, a sample of beam MC events in the ECals were processed through the enhanced

	1 track likelihood	2 track likelihood	3 track likelihood
1 st track matched	96%	92%	80%
2 nd track matched		89%	79%
3 rd track matched			70%

Table 1.1: The percentage number of correct matches in the 3D matching separated by which track matching likelihood was used. NEUT-based Monte Carlo simulation of the T2K beam was used to calculate the matching percentages.

reconstruction. After completion, the 2D components of the 3D tracks were analysed and matched to the true particles which created them. A match was counted as a success if both 2D components were matched to the same true particle. These results are shown in table 1.1, separated out by which track matching likelihood was used. For the 1 and 2 track likelihood cases, the matching performs very well. It is only in cases where the 3 track likelihood is used that the matching start to suffer slightly. However, all of the correct matching rates are well above 50% which suggests that the 3D matching is performing adequately.

1.4.2 Data-motivated validation of the algorithms

As described above, the validation is still an ongoing effort and, so far, all of the data-motivated validation has been done by L. Pickering. At time of writing, validation events measured in ND280 can be split into two groups:

1. **FGD collinear cosmics (named fgdcoll):** A sample of clean FGD-triggered cosmic events which are scaled to have an equal distribution of angles.
2. **Through-going Tracker muons (named bskmu):** A sample of clean beam-triggered events which solely contain a muon depositing energy in TPC2, TPC3 and the DS ECal.

The fgdcoll sample will naturally contain a good sample of barrel ECal events whilst having only a limited number of DS ECal events whereas the reverse is true for the bskmu sample. So, it follows that the fgdcoll and bskmu samples should be used to separately assess the reconstruction in the barrel ECal and DS ECal respectively.

Fig. 1.17 shows the summed charge contained in all tracks reconstructed in each

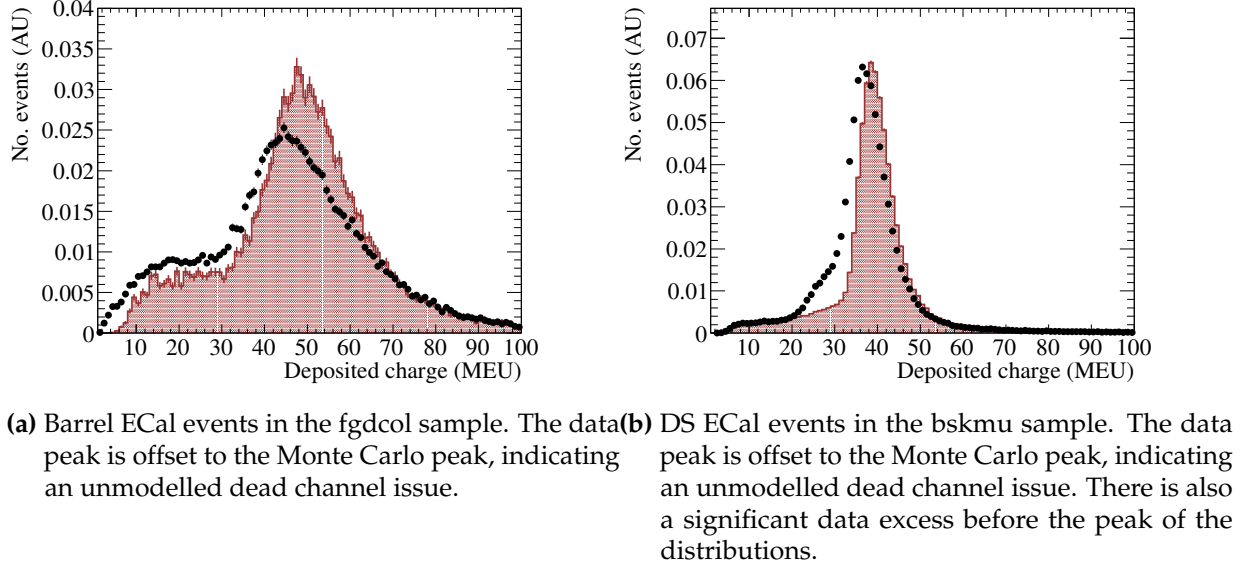
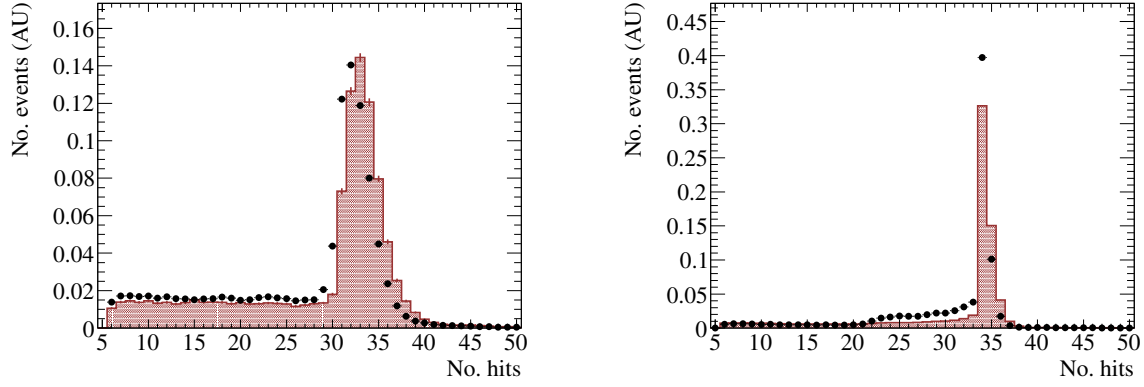


Figure 1.17: The summed charge contained on all tracks reconstructed in each ECal cluster. The red histograms and black points are Monte Carlo and data respectively.

ECal cluster. To clarify what this means, if a reconstructed ECal cluster contained three reconstructed tracks where tracks 1, 2 and 3 contain 20 MEU, 50 MEU and 30 MEU of charge respectively, that event would be registered as having 100 MEU of charge in the relevant distribution in Fig. 1.17. Both the barrel ECal and DS ECal samples show a large discrepancy around the charge peak. This offset can be explained as a hit-level charge discrepancy between data and Monte Carlo and is not caused by the reconstruction. This discrepancy should, however, be considered during analysis of the systematic uncertainties. Unfortunately, the DS ECal events in the bskmu sample (Fig. 1.17b) shows an extra data excess in the 20 MEU to 30 MEU region which is just before the charge peak. This data excess can not be explained by a relative hit inefficiency.

Fig. 1.18 shows the summed number of hits contained in all tracks reconstructed in each ECal cluster. Generally speaking, the distributions in Fig. 1.18 share similar features to those in Fig. 1.17. Importantly, an excess of data events appears just before the peak in Fig. 1.18. The fact that this excess also appears in Fig. 1.18 strongly suggests that the issue is caused by the number of hits associated to the reconstructed tracks, rather than the deposited charge. Further investigation of this discrepancy revealed that a mismodelled hit inefficiency, most likely due to dead DS ECal channels, was the issue. One of the 2D track quality checks (see section 1.2.4) requires that a 2D track

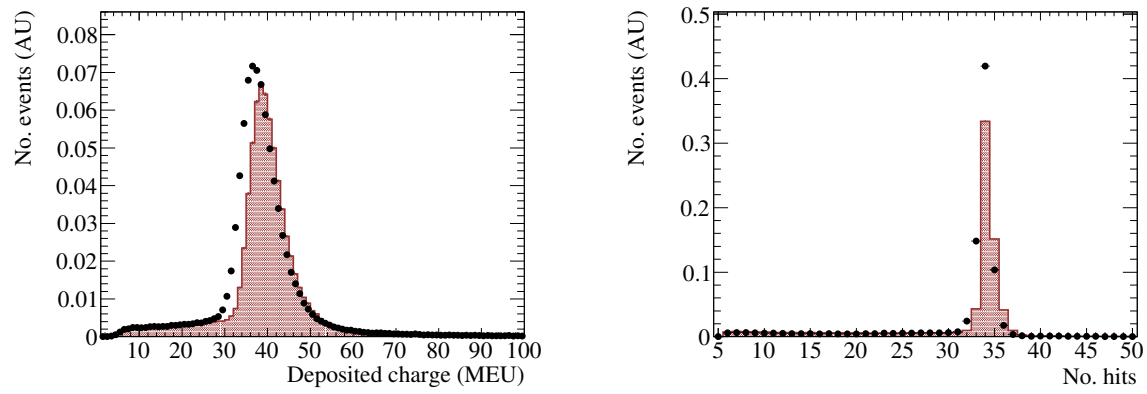


(a) Barrel ECal events in the fgdcoll sample. The only significant discrepancy is the offset of the data peak relative to the Monte Carlo peak, suggesting an unmodelled dead channel issue. (b) DS ECal events in the bskmu sample. There is a significant data excess before the peak in the distributions.

Figure 1.18: The summed number of hits contained on all tracks reconstructed in each ECal cluster. The red histograms and black points are Monte Carlo and data respectively.

candidate can not skip a layer in a given view which is problematic when considering dead channels.

To test this hypothesis, this requirement was relaxed to allow a 2D track candidate to skip a single ECal layer and the Monte Carlo bskmu sample was re-processed. The re-processed DS ECal events are shown in Fig. 1.19. As can clearly be seen, the data excess before the peak in the distributions has now gone. Unfortunately, large scale processing and analysis of beam Monte Carlo events had already started by the time this issue was discovered and so it was infeasible to include this bug fix in the analysis. So, it was decided that this issue would be treated as an additional systematic uncertainty. The collected data events were processed with this fix in place.



(a) Summed charge contained on all tracks in each reconstructed ECal cluster. (b) Summed number of hits contained on all tracks in each reconstructed ECal cluster.

Figure 1.19: The number of DS ECal events in the bskm sample after relaxing the 2D track quality check. The red histograms and black points are the Monte Carlo and data respectively. The track quality fix has corrected the data excesses in both distributions, leaving only the offset between the data and Monte Carlo peaks.

Chapter 2

Selection of neutrino interactions in the ECal

This analysis presents a measurement of the CC inclusive interaction cross-section of ν_μ with lead nuclei using the ND280 Tracker¹ ECals. To make such a measurement, a sample of neutrino interaction vertices within the ECal must be found. The selection of events is based on the enhanced reconstruction outlined in chapter 1, which was specifically designed to be sensitive to² track multiplicity. As a result of this method, vertices in the ECal are naturally separated into topologies defined by the number of reconstructed tracks. Any selection development should take advantage of this situation and tailor cuts to be specific to each topology, which should result in a higher overall sample purity. The 3D track matching aspect of the reconstruction was only tuned to handle up to three tracks simultaneously. However, this analysis aims to measure a CC-inclusive cross-section so it should not be biased against any neutrino energy range. There is a deep connection between the number of reconstructed tracks and the energy of the neutrino that created them. The number of reconstructed tracks should correlate with the number of final state particles involved with the neutrino interaction and the number of final state particles correlates with the energy of the interacting neutrinos. Therefore it is important to not reject events based solely on the number of reconstructed tracks³. Bearing this information in mind, the selection should separate out the events into the following topologies:

- 1 prong topology

¹ADDRESSED - I would say that "Tracker" is a proper noun, the name of the detector, so it should be capitalised throughout. The detector name is then "Tracker ECal"

²ADDRESSED - introduce

³ADDRESSED - state the implied connection between number of prongs and neutrino energy

- 2 prong topology
- 3 prong topology
- 4+ prong topology

The definition of a prong is a reconstructed track associated with a reconstructed vertex.

However, as described in section 1.3, the output of the enhanced reconstruction has been kept generic and is not specifically tailored to this task. The reconstruction outputs a set of clusters which contain a set of 3D tracks and every pairwise crossing that said tracks make. While it is true that in some situations the reconstruction will accurately represent a vertex “out of the box”—e.g., when only two tracks are reconstructed in the cluster—there will be many situations where extra reconstruction steps are needed before any further selection can take place. Hence⁴ the structure of this chapter is as follows: the definition of signal is described first along with the sample used to develop the selection, followed by a discussion of the vertex reconstruction. After the final reconstruction steps have been discussed, a full discussion of the neutrino selection follows.

2.1 Signal definition

The measurement that is the intended outcome⁵ of this analysis is the CC inclusive interaction cross-section of ν_μ with lead nuclei using the ND280 Tracker ECals. The active volume of the ECal consists of layers of plastic scintillator and lead absorbers. Neutrino interactions that occur in the lead absorbers are indistinguishable from those that occur in the plastic scintillator. Because of this fact, the selection does not attempt to separate the two cases out. So, the strict signal definition used by this selection is as follows:

- Interacting neutrino is flavour ν_μ
- Interaction type is charged current

⁴ADDRESSED - I wouldn't start a sentence with “So,”—it seems like spoken English to me. I'd say “Hence” with no comma

⁵ADDRESSED - “measurement of this analysis”

- Interaction occurs within the active volumes of either the barrel ECal or DS ECal

The active volume is defined by the ROOT geometry used in the simulation. Specifically, it is a rectangular box which encompasses all of the lead and scintillator layers for each ECal module. Any target element contained within this volume is a signal target.

2.2 Monte Carlo sample

NEUT was used to generate a neutrino beam sample of Monte Carlo events which corresponds to 3.949×10^{20} POT. All of this sample was generated using a simulated beam power of 178 kW which is the average beam power used for Run III. At this intensity, there are an expected 9.46 neutrino interactions per eight bunch spill in ND280. So, there will be approximately one interaction per bunch across the entire ⁶ ND280, meaning the chance of pileup is small.

The sample ⁷described above only simulated neutrino interactions within ND280 (including the ⁸outermost magnet). However, we expect many interactions in the surrounding pit (referred to as “sand interactions” from now on) to have final states which enter ND280. So, NEUT was used to generate an additional sample of sand interactions (hereafter referred to as sand Monte Carlo) which corresponds to 3.708×10^{20} ⁹POT. There is roughly a 6% difference in the POT of the beam and sand Monte Carlo. As the sand Monte Carlo has a sufficiently high level of statistics, scaling the smaller sample to match the larger is acceptable.

The number of signal interactions from the Monte Carlo sample is shown in table ¹2.1, separated into the ECal module they occurred in. The number of signal interactions is also shown as a function of neutrino energy for the bottom-left, top-right and DS ECals in Fig. ²2.1. Clearly, there are an extremely high number of signal events in the sample, which means this analysis is in no way statistically limited. As an example, consider a selection with a 1% signal efficiency in the DS ECal. Assuming 100% selected sample

⁶ADDRESSED - was ‘of’

⁷ADDRESSED

⁸ADDRESSED - was ‘surrounding’

⁹ADDRESSED

ECal module	No. signal events
Bottom-right	223042
Side-right	231955
Top-right	135476
Bottom-left	262500
Side-left	293065
Top-left	140539
Downstream	134100
Total	1420677

Table 2.1: The number of signal events in the Monte Carlo sample separated by ECal module.

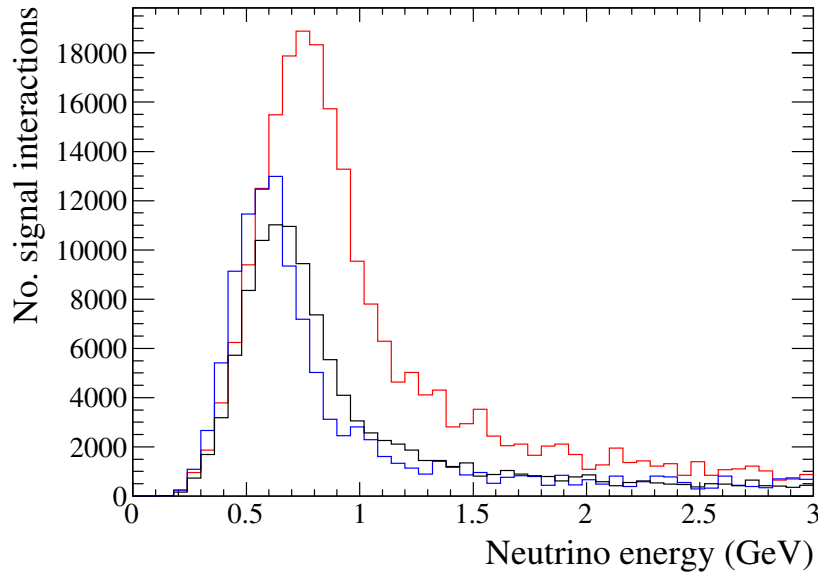


Figure 2.1: The number of signal events as a function of neutrino energy in the bottom-left (red), top-right (blue) and DS (black) ECals.

purity, the number of selected events would be 1341 which correspond to a statistical error of 2.73%. Even in this extreme situation, the statistical error is borderline negligible. It should be clear that any statistical error in this analysis is approximately 0%.

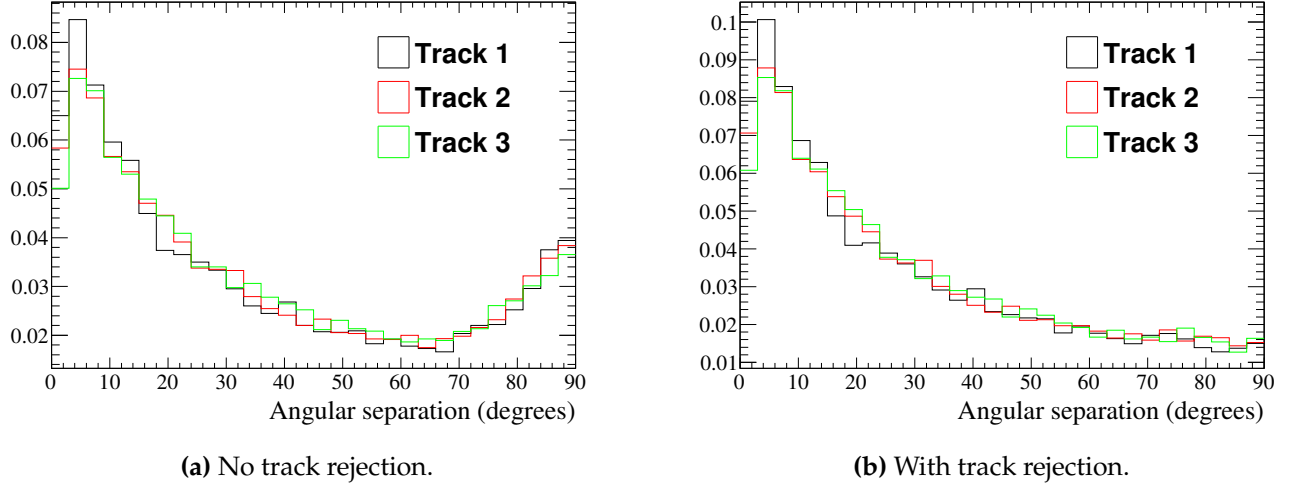


Figure 2.2: Angular separation of the reconstructed tracks in an ECal cluster with the beam-simulated particle that created them. The black, red and green histograms are the tracks that were reconstructed first, second and third respectively. All histograms are area normalised.

2.3 Vertex reconstruction

The output of the enhanced reconstruction already supplies most of the necessary information to reconstruct vertices in the ECal. As a reminder, the reconstruction outputs a set of ECal clusters which contain the following:

- A set of 3D reconstructed tracks
- The position at which every pairwise combination of 3D tracks most closely cross (pairwise crossings)

The reconstruction makes no quality checks on the 3D tracks in each cluster. So, the first step is to remove any poorly reconstructed tracks from the cluster. The need for this step is shown in Fig. 2.2a which shows the angular separation of reconstructed tracks with the simulated particle which created them, taken from beam Monte Carlo. While the majority of tracks generally have a small angular separation, Fig. 2.2a clearly shows a build up of tracks which are offset by 90° to the simulated particles.

There are two categories of bad track reconstruction. The first is where the 3D matching matches two 2D tracks which do not have sufficient information to fully model a 3D track. Specifically, one of the 2D tracks involved in the matching only uses one

ECal layer. As the 3D track reconstruction requires information from both ECal views, tracks which fall into this category do not supply enough information to reconstruct a good quality track. The second category is where the 3D matching produces a grossly incorrect match. The matcher is designed to continually match 2D tracks together until no more 2D tracks are left in the matching pool. So, there are situations where the last possible match made is in no way suitable. This situation is easily identified by matched 2D tracks which do not overlap. For example, one of the 2D tracks uses layers 16, 18 and 20 and the other 2D track uses layers 1 and 3. By removing these two types of tracks from the cluster, the 90° build up in the angular separation distribution is suppressed, as shown in Fig. 2.2b. By removing these categories of tracks, approximately 22% of tracks are rejected.

The final states of a neutrino interaction originate from the same point in space. So, assuming that the reconstructed tracks represent the final states, the reconstructed tracks should most closely cross at the point of interaction. As stated above, one of the outputs of the reconstruction are the pairwise crossings of the constituent tracks in the ECal cluster. Using the above assumptions, the pairwise crossings should be in close proximity to one another when the reconstructed tracks represent the final states of an interaction. This idea promotes a relatively simple method of vertex reconstruction: attempt to cluster the pairwise crossings together if the pairwise crossings are in close proximity. To quantitatively define this proximity, the quality of the pairwise crossing also has to be considered.

As with the 3D tracks, the reconstruction does not perform any quality checks on the pairwise crossings of the tracks. As the reconstruction will always find a crossing location for a pair of 3D tracks, some of the pairwise crossings will not represent anything physical. The quality definition chosen is simple: pairwise crossings are defined as bad if the crossing location is far away from either of the tracks it is associated with. However, the distance definition is not trivial to define and should be closely correlated with the vertex reconstruction method. For example, a two track vertex would provide very little constraint on the distance which defines a bad crossing, whereas a three track vertex may provide a bad crossing distance constraint which is too strict and destroys the majority two track vertices.

The above discussion promotes two parameters which will govern the vertex reconstruction: the required proximity of two crossings to be clustered together, d_c , and

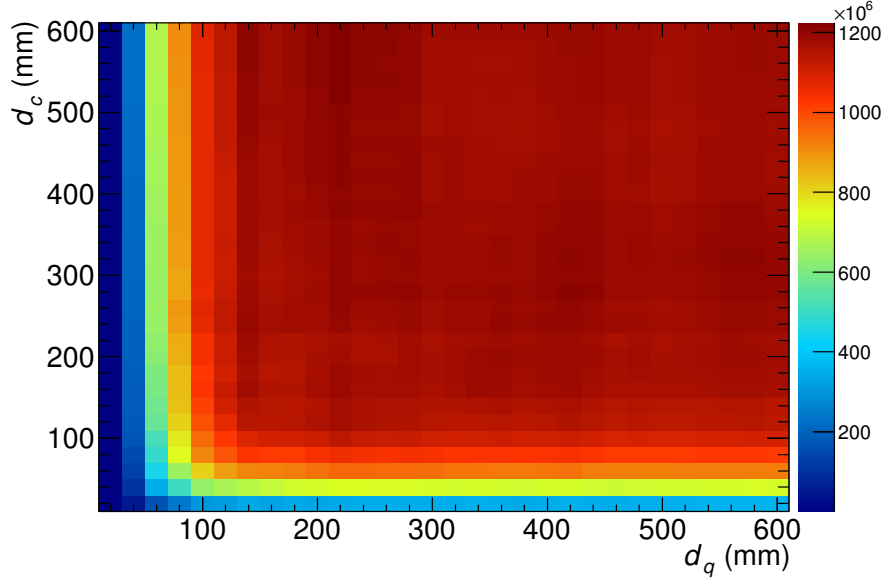


Figure 2.3: Values of ϕ^{vertices} in (d_q, d_c) space. The colour corresponds to the magnitude of ϕ^{vertices} .

the distance of a crossing from its constituents tracks to be classified as bad quality, d_q . To quantify these values, a sample of reconstructed events matched to interactions in the ECals are used which are taken from beam Monte Carlo. The reconstruction (the pairwise crossing rejection and clustering) is repeatedly run over the sample for different values of the d_c and d_q to find the optimum values. To define the optimum value, a figure of merit is necessary. After running the reconstruction for a given d_c and d_q , the number vertices are separated into one, two and three track vertices and the true neutrino interactions which created the reconstructed vertices are associated. A reconstructed vertex is tagged as correctly reconstructed if it contains the same number of reconstructed tracks as the number of charged final states in the associated neutrino interaction. By defining the number of correctly reconstructed one, two and three track vertices as N_1 , N_2 and N_3 respectively, the figure of merit, ϕ^{vertices} , is

$$\phi^{\text{vertices}} = N_1 N_2 N_3. \quad (2.1)$$

By mapping out ϕ^{vertices} in (d_q, d_c) space, information about the preferred values of d_q and d_c can be found. This space is shown in Fig. 2.3. It is clear from Fig. 2.3 that there is no clear maximum, but rather a plateau of ϕ^{vertices} for d_q and d_c greater than 140 mm. So, marginalised distributions of d_q and d_c can be produced to find where ϕ^{vertices} approaches zero which are shown in Fig. 2.4a and Fig. 2.4b respectively. The

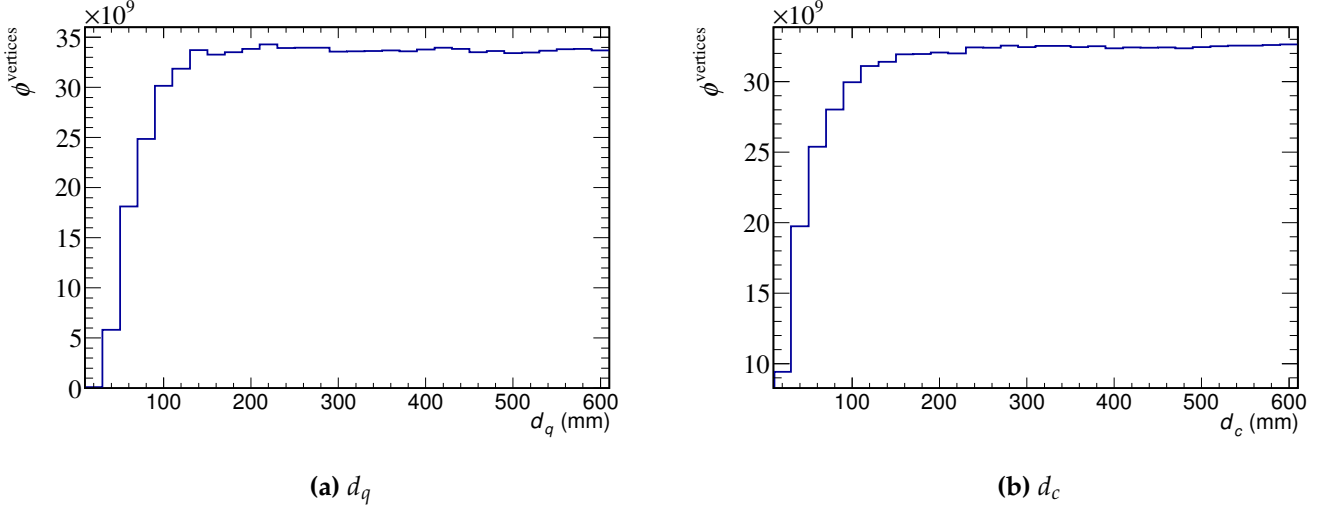


Figure 2.4: ϕ^{vertices} vs the marginalised vertex reconstruction parameters.

ϕ_d	ϕ_c
$< 140 \text{ mm}$	$< 200 \text{ mm}$

Table 2.2: Parameters for the vertex reconstruction in the ECal.

values chosen are shown in table 2.2. The reconstruction now assesses the quality of the crossings and then attempts to cluster the good quality crossings together to form vertex candidates. The final step is to use the constituent tracks of each vertex candidate in a fit to estimate the position of the vertex. The following method was suggested by X. Lu. The position of the vertex, \vec{P} , is defined such that the sum of the squares of the distance of each track to \vec{P} is minimised. An example setup of this is shown in Fig. 2.5 for three constituent tracks. By defining the square of the distance of a line, l_i , to \vec{P} as $|\vec{r}_i|^2$, the function to minimise is

$$D = \sum_i |\vec{r}_i|^2. \quad (2.2)$$

\vec{P} is then defined as the point in space which satisfies

$$\frac{\partial D}{\partial x} = \frac{\partial D}{\partial y} = \frac{\partial D}{\partial z} = 0. \quad (2.3)$$

The value of $|\vec{r}_i|$ is trivially defined by simple vector properties as

$$|\vec{r}_i| = \frac{|(\vec{P} - \vec{a}_i) \times \vec{v}_i|}{|\vec{v}_i|} = |(\vec{P} - \vec{a}_i) \times \hat{v}_i|, \quad (2.4)$$

where \vec{v}_i is the direction vector of \vec{l}_i and \vec{a}_i is a point along \vec{l}_i . By defining \vec{P} as

$$\vec{P} = x\hat{i} + y\hat{j} + z\hat{k}, \quad (2.5)$$

equation 2.3 can be solved. The following derivation is only for $\partial D / \partial x$ as the method is identical for $\partial D / \partial x$, $\partial D / \partial y$ and $\partial D / \partial z$.

$$\begin{aligned} \frac{\partial D}{\partial x} &= \sum_i 2\vec{r}_i \cdot \frac{\partial \vec{r}_i}{\partial x} \\ &= 2 \sum_i [(\vec{P} - \vec{a}_i) \times \hat{v}_i] \cdot [\hat{i} \times \hat{v}_i] \\ &= 2x \sum_i (\hat{i} \times \hat{v}_i) \cdot (\hat{i} \times \hat{v}_i) + 2y \sum_i (\hat{j} \times \hat{v}_i) \cdot (\hat{i} \times \hat{v}_i) + 2z \sum_i (\hat{k} \times \hat{v}_i) \cdot (\hat{i} \times \hat{v}_i) \\ &\quad - 2(\vec{a}_i \times \hat{v}_i) \cdot (\hat{i} \times \hat{v}_i). \end{aligned} \quad (2.6)$$

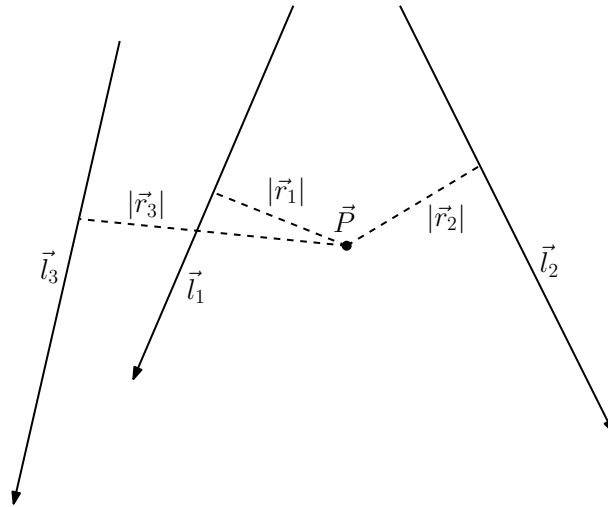


Figure 2.5: Example of the vertex position, \vec{P} for three lines: \vec{l}_1 , \vec{l}_2 and \vec{l}_3 . $|\vec{r}_1|$, $|\vec{r}_2|$ and $|\vec{r}_3|$ are the perpendicular distances of \vec{l}_1 , \vec{l}_2 and \vec{l}_3 to \vec{P} respectively.

By applying the same steps for $\partial D/\partial y$ and $\partial D/\partial z$, the matrix equation

$$\begin{pmatrix} \sum_i [\hat{i} \times \hat{v}_i] \cdot (\hat{i} \times \hat{v}_i) & \sum_i [\hat{j} \times \hat{v}_i] \cdot (\hat{i} \times \hat{v}_i) & \sum_i [\hat{k} \times \hat{v}_i] \cdot (\hat{i} \times \hat{v}_i) \\ \sum_i [\hat{i} \times \hat{v}_i] \cdot (\hat{j} \times \hat{v}_i) & \sum_i [\hat{j} \times \hat{v}_i] \cdot (\hat{j} \times \hat{v}_i) & \sum_i [\hat{k} \times \hat{v}_i] \cdot (\hat{j} \times \hat{v}_i) \\ \sum_i [\hat{i} \times \hat{v}_i] \cdot (\hat{k} \times \hat{v}_i) & \sum_i [\hat{j} \times \hat{v}_i] \cdot (\hat{k} \times \hat{v}_i) & \sum_i [\hat{k} \times \hat{v}_i] \cdot (\hat{k} \times \hat{v}_i) \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \sum_i [\vec{a}_i \times \hat{v}] \cdot (\hat{i} \times \hat{v}) \\ \sum_i [\vec{a}_i \times \hat{v}] \cdot (\hat{j} \times \hat{v}) \\ \sum_i [\vec{a}_i \times \hat{v}] \cdot (\hat{k} \times \hat{v}) \end{pmatrix} \quad (2.7)$$

can be built up. By defining equation 2.7 as

$$\underline{\underline{A}} \vec{P} = \vec{B}, \quad (2.8)$$

where $\underline{\underline{A}}$ is the matrix on the left hand side of equation 2.7 and \vec{B} is the vector on the right hand side of equation 2.7. \vec{P} is finally

$$\vec{P} = \underline{\underline{A}}^{-1} \vec{B}. \quad (2.9)$$

While the inversion of $\underline{\underline{A}}$ is in principle analytically solvable, it was decided that the inversion would be handled numerically. So, to find \vec{P} , the vertex reconstruction builds $\underline{\underline{A}}^{-1}$ (by building and inverting $\underline{\underline{A}}$) and \vec{B} and then applies equation 2.9.

By applying the steps outlined above to every reconstructed ECal cluster, a set of candidate vertices are formed. As the enhanced reconstruction is only capable of reconstructing straight tracks, bending trajectories tend to be reconstructed as two or more tracks. The crossings associated with such tracks can, and do, pass the vertex reconstruction criteria defined in table 2.2. An extreme example of this topology is shown in Fig. 2.6 in which a curving μ^- is reconstructed as four tracks. So, the final step of the reconstruction is to attempt to merge tracks to model the curving trajectory topology. Firstly, the reconstructed crossings associated with a clean curving trajectory should pass the pairwise crossing quality cut but should be sufficiently far away from any other crossings that it is not clustered during the vertex clustering stage. Therefore, track merging candidates can be initially identified by searching for reconstructed vertices with exactly two track constituents. As an example, all three of the crossings shown in Fig. 2.6 are correctly identified by this check. The two constituent tracks of the identified vertex will form the merged track, so the reconstruction checks the shape that the merged track would make against a set of conditions to decide if the merging

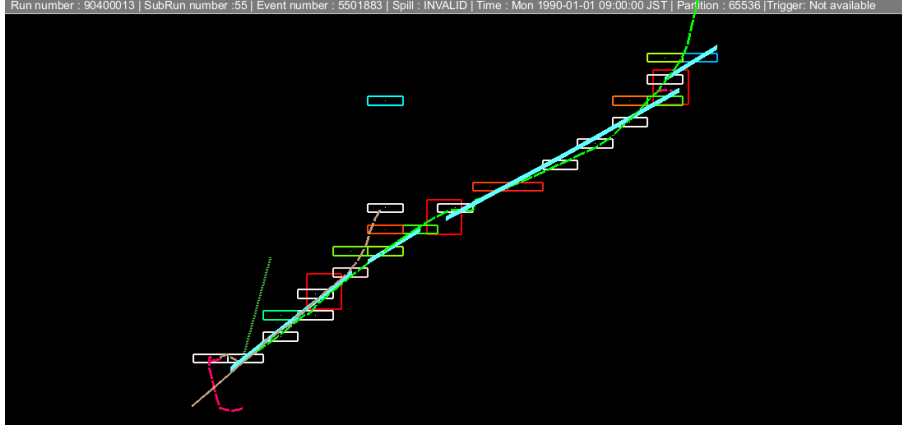


Figure 2.6: Example event display showing a set of track merging candidates. The final state μ^- (in green) curves in a complicated way and is reconstructed as four tracks (in light blue). The pairwise crossings of the reconstructed tracks are the red squares. The muon is produced by a charged current interaction in the top left barrel ECal producing two π^+ (in brown) and the curving μ^- .

should be performed. To identify and tune the conditions for merging tracks, two metrics are used. The first metric is used for identification of the merging conditions and uses the truth information provided by the Geant4 simulation. For every two track vertex that was formed by the vertex clustering, the simulated particle that produced each of the two constituent tracks was checked. If the same simulated particle is matched to both tracks, then the merging candidate is tagged as a correct match, otherwise it is tagged as an incorrect match. The second metric is used for tuning the identified merging conditions. As part of this analysis is a search for neutrino interactions using vertex-based reconstruction, there will be an expected loss of signal by track merging which must be minimised. So, the track merging condition tuning should be based on ECal signal interactions. As described above, the only tracks that are proposed as merging candidates are those which are constituents of a two track vertex. So, the track merging tuning attempts to separate signal and background events which are reconstructed as a vertex with two track constituents. It is preferable that the merging chooses quality over quantity so the tuning figure of merit is defined as

$$\phi^{\text{merge}} = \epsilon \eta^2 \quad (2.10)$$

where ϵ is the efficiency of the track merging to keep signal events reconstructed as two track vertices and η is the purity of the events that remain as two track vertices after merging has taken place.

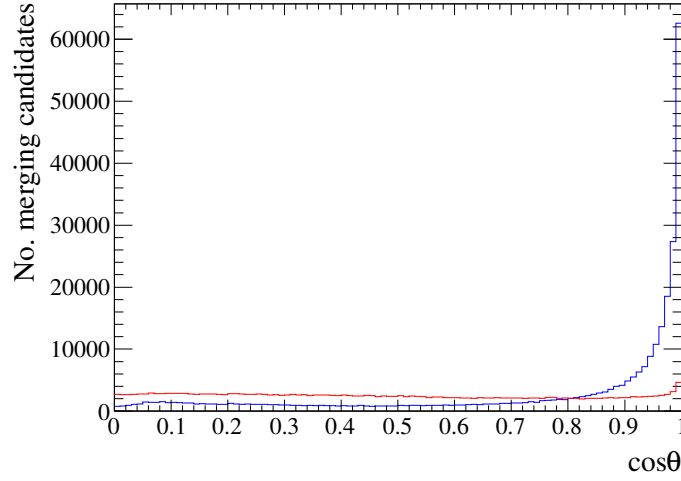


Figure 2.7: The cosine of the angle subtended by the merging candidates, bounded between 0 and 1. The correct matches and incorrect matches are the blue and red histograms respectively.

The first merging condition identified is the cosine of the opening angle, $\cos \theta$, subtended by the two constituent tracks bounded between 0 and 1 which is shown in Fig. 2.7. The opening angle is clearly a powerful discriminator. The distribution for incorrect matches is very flat across the full angular range whereas there is a clear build up of correct matches as $\cos \theta \rightarrow 1$.

The second merging condition identified, called ‘distance ratio’, measures the ratio of the distance between the two constituent tracks’ closest points, d^{small} to the distance between their furthest points, d^{large} . An example of how d^{small} and d^{large} are calculated is shown in Fig. 2.8. The distance ratio distribution is shown Fig. 2.9a. While it may seem that there is very little discrimination power present in the distance ratio distribution, there is an underlying dependency between the distance ratio and the opening angle subtended by the constituent tracks, which has already been identified as a merging condition. To illustrate this dependency, two distributions are shown in Fig. 2.10 which both show the distance ratio vs the cosine of the opening angle. Fig. 2.10a only shows the correct matches and Fig. 2.10b only shows the incorrect matches. While it is true that there is a pileup of both correct matches and incorrect matches for low values of the distance ratio, the opening angle separates the two categories out. Specifically, the correct matches pileup occurs as $\cos \theta \rightarrow 1$ and the incorrect matches pileup occurs as $\cos \theta \rightarrow 0$. To further illustrate this point, a

new distance ratio distribution is shown in Fig. 2.9b, but with a $\cos \theta > 0.8$ guess cut applied. Comparing the distributions shown in Fig. 2.9a and Fig. 2.9b, the effect of the $\cos \theta$ cut can clearly be seen. The original pileup of incorrect matches seen in Fig. 2.9a is now gone, leaving an essentially flat incorrect matches distribution in Fig. 2.9b while leaving the correct matches structure intact.

By utilising both the distance ratio and $\cos \theta$ simultaneously, a better degree of separation can be found. However care must be taken when tuning the distance ratio and $\cos \theta$ cuts to ensure optimal separation of signal and background is achieved. It is clear from Fig. 2.7 and Fig. 2.9b that the correct matches pileup for low values of the distance ratio and high values of $\cos \theta$. So events should only be tagged for merging when they have a distance ratio value lower than some threshold and a $\cos \theta$ value higher than some other threshold. To find these cut values, the track merging reconstruction was run multiple times, using different values of the thresholds for each run. To take the dependency shown in Fig. 2.10 into account, a square grid search in distance ratio and $\cos \theta$ space was used to find optimum cut values. The tuning metric, as described in equation 2.10, was used to find the optimum cut values. The tuning metric values in distance ratio cut vs $\cos \theta$ cut space are shown in Fig. 2.11. As was found in the tuning of the vertex clustering parameters, there is no clear maximum value of the tuning metric, but rather a plateau. So, as was done in the vertex clustering tuning,

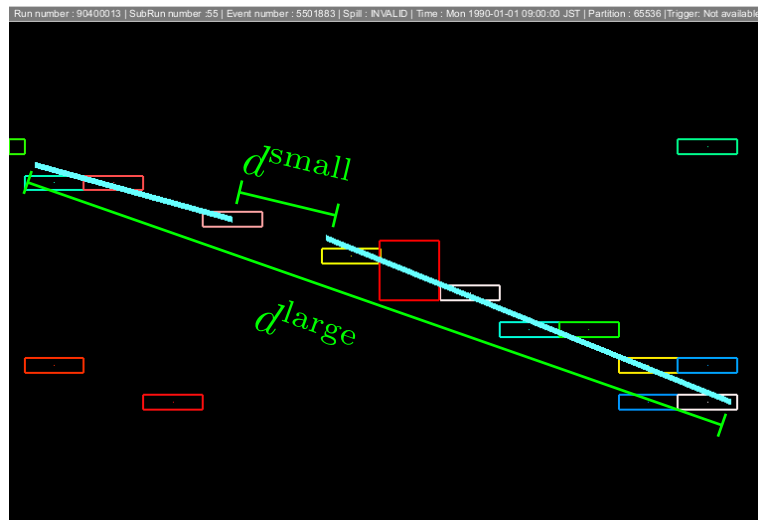


Figure 2.8: Example event display showing how the parameters of the distance ratio are calculated. The distance ratio is defined as $d^{\text{small}} / d^{\text{large}}$. The blue lines are the two reconstructed tracks which form the merging candidate and the red square is the crossing location of those tracks.

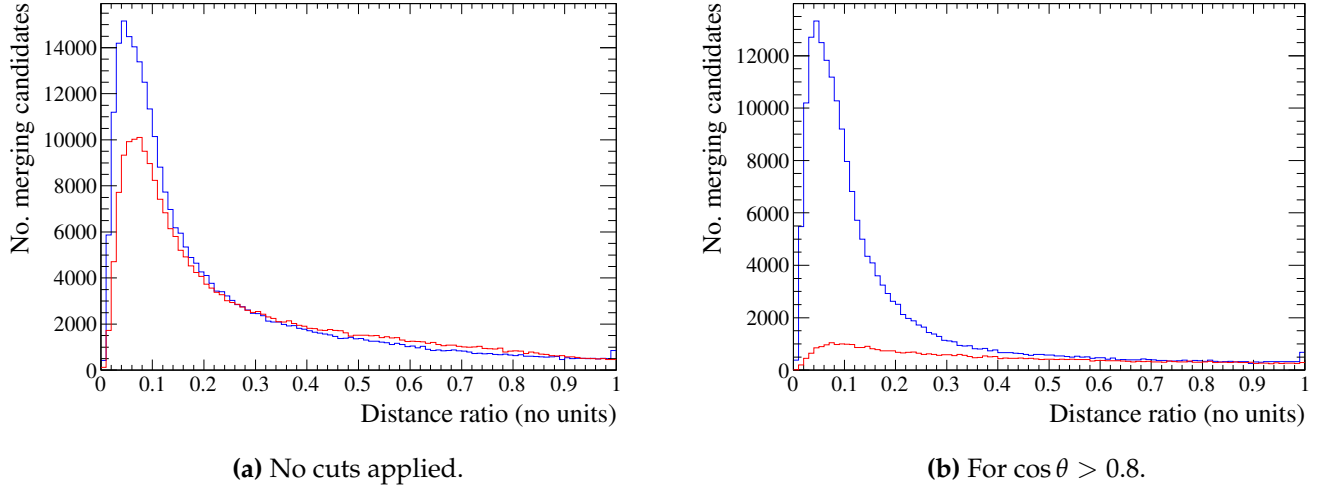


Figure 2.9: The distance ratio of the merging candidates. The correct and incorrect matches are the blue and red histograms respectively.

marginalised distributions of tuning metric in distance ratio cut space and $\cos \theta$ cut space can be produced to find the optimum cut values. The marginalised distributions for $\cos \theta$ and the distance ratio cuts are shown in Fig. 2.12a and Fig. 2.12b respectively. In the case of the $\cos \theta$ cut there is a clearly preferred value, merging candidates should only be merged if $\cos \theta > 0.82$. In the case of the distance ratio cut distribution, there is less of a clear maximum. As the reconstruction is striving for quality over quantity, the cut value should be fairly close to the drop in ϕ^{merge} . It was decided that merging

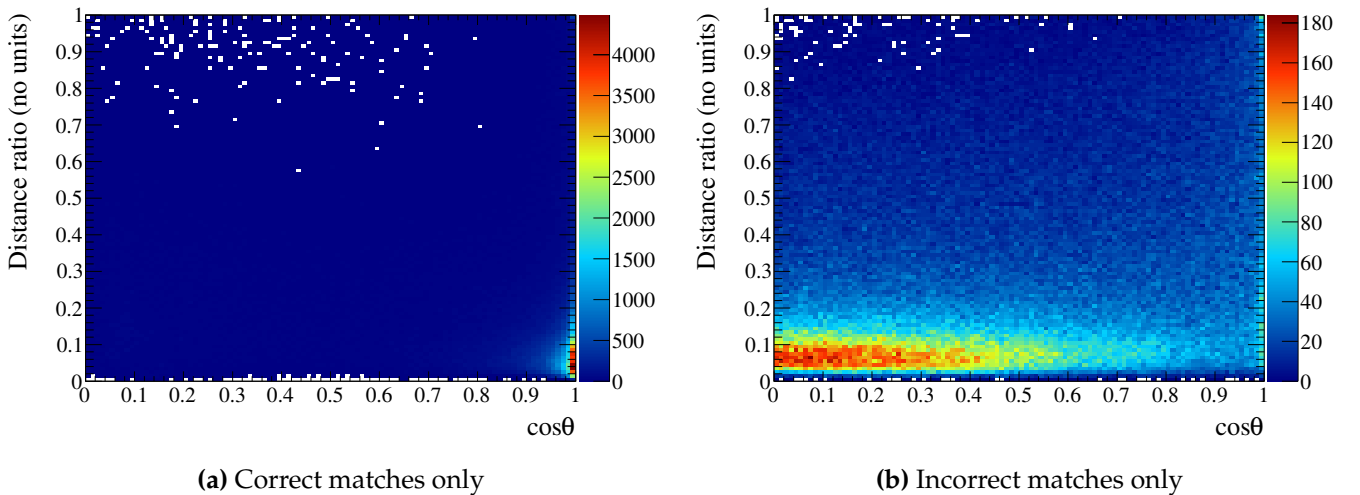


Figure 2.10: The distance ratio vs the cosine of the opening angle for track merging candidates.

candidates should only be merged if the distance ratio is less than 0.32.

While the $\cos\theta$ condition is clearly a powerful discriminator, there is a topology degeneracy which $\cos\theta$ is not capable of separating, which is shown in Fig. 2.13. The diagram on the left of Fig. 2.13 is a representation of a signal event which is reconstructed as two tracks, whereas the diagram on the right of Fig. 2.13 represents a curving trajectory reconstructed as two tracks. Importantly, the same opening angle is measured for both situations which is small enough that both topologies pass the $\cos\theta$ track merging condition. To rectify this, an extra sanity check is needed when considering merging candidates. So, the final merging condition identified, called ‘swing’, measures the rotation of one track relative to the other. Specifically, the swing is the ratio of the longest track length, l^{long} , to d^{large} (the same d^{large} that was used in the distance ratio calculation). An example of how these values are calculated is shown in Fig. 2.14. Provided that the opening angle of a merging candidate is not near 90° , the swing should be less than one for the topology on the left of Fig. 2.13 and should be greater than one for the topology on the right of Fig. 2.13. The separation power of the swing parameter is shown in Fig. 2.15. After applying the merging conditions discussed above, the swing parameter becomes bi-modal as shown in Fig. 2.15b. The swing parameter is introduced purely as an extra sanity cut which is physically motivated and requires no tuning. So, merging candidates are only merged if the measured swing is less than one.

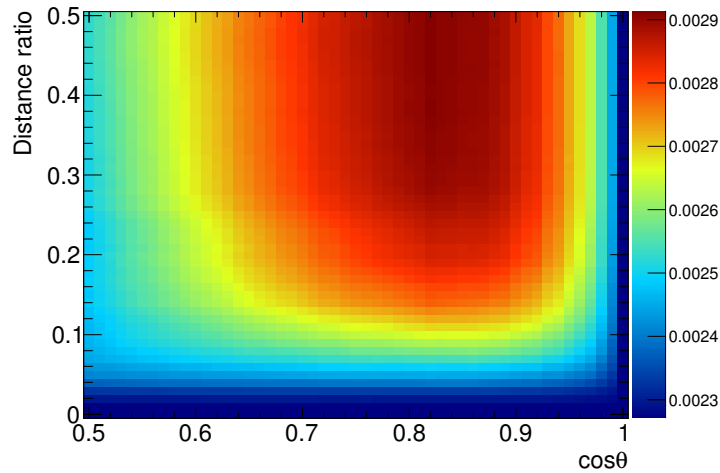


Figure 2.11: Values of the tuning metric, as described in equation 2.10, in distance ratio cut vs $\cos\theta$ cut space.

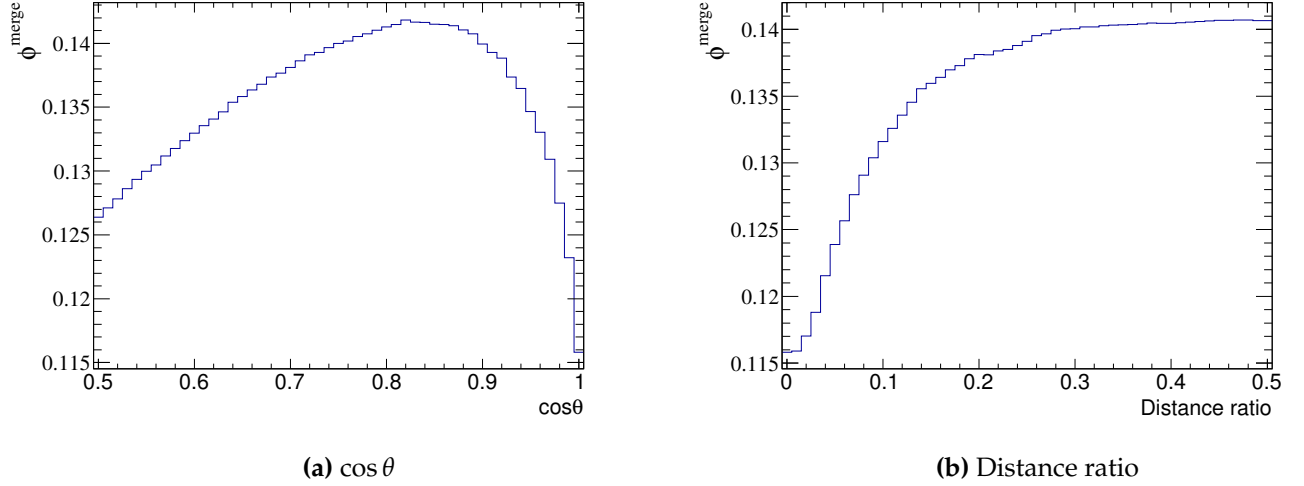


Figure 2.12: ϕ^{merge} vs the track merging conditions.

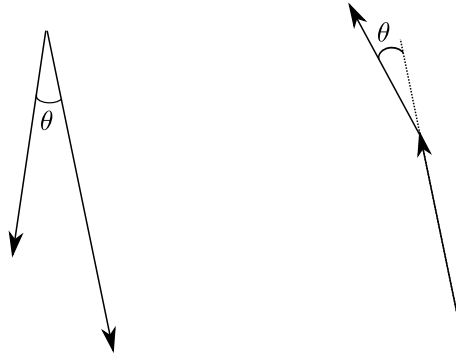


Figure 2.13: Schematic showing the degeneracy of two merging candidate topologies which would pass the $\cos \theta$ cut. The arrows represent reconstructed tracks and θ is the opening angle measured between the tracks.

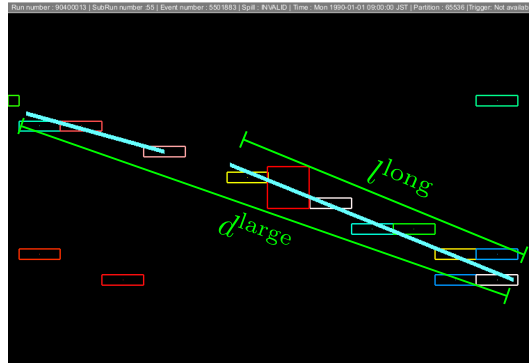


Figure 2.14: Example event display showing how the inputs to the swing parameter are calculated. Swing is defined as $l^{\text{long}} / d^{\text{large}}$. The blue lines are the two reconstructed tracks which form the merging candidate and the red square is the crossing location of those tracks.

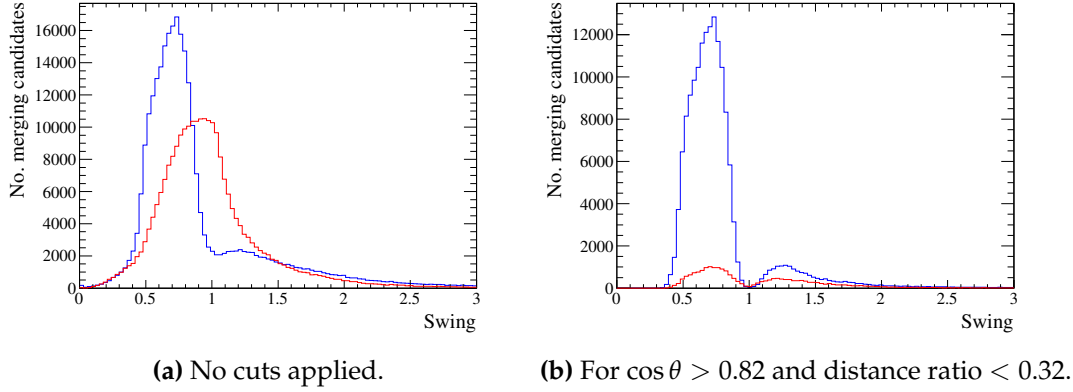


Figure 2.15: The number of merging candidates as a function of the swing parameter. The correct matches and incorrect matches are the blue and red histograms respectively.

All of the merging conditions have now been identified and the associated cut values are summarised in table 2.3.

2.4 Monte Carlo selection

As briefly described at the start of this chapter, the selection separates the reconstructed ECal events into a set of topologies, where each topology is defined by a number of associated prongs. In addition to this separation, the geometrical differences between the barrel and DS ECal suggest a further event separation. The DS ECal lies perpendicular to the beam face. So, particles from neutrino interactions will typically travel at right angles to the DS ECal face. Conversely, all of the barrel ECals lie parallel to the beam axis which means neutrino interaction products will generally pass along the barrel scintillator planes. This motivates a separate selection for the barrel and DS ECal. This essentially means that there are eight individual selections to be made and tuned (1,2,3 and 4+ prong topologies in the barrel and 1,2,3 and 4+ prong topologies in the DS ECal). Clearly, this approach can very quickly become complicated. To mitigate

$\cos \theta$	Distance ratio	swing
> 0.82	< 0.32	< 1

Table 2.3: Cut values for the track merging in the ECal.

this, similar discriminators are used for each topology. As the reconstruction was only tuned for three reconstructed tracks, the 4+ prong topology will use exactly the same cuts as the 3 prong topology.

Care must be taken when defining what a reconstructed vertex is. For the 2,3 and 4+ prong topologies, it is fairly simple, the reconstructed vertex is as described in section 2.3, but it is not possible to fit for a vertex position when dealing with a single prong. In fact, there is actually little that can be done about this. So, beam kinematics are assumed and the 'vertex' for a single prong is the most upstream end of said prong.

The definition of signal has already been described in section 2.1. However, how the reconstructed events map to the signal interactions also needs discussion. The selection takes place at the reconstructed vertex level but there is not a clear 1:1 map of reconstructed vertices to signal interactions. Consider a signal interaction in an ECal module in which the final state particles are involved in one or more secondary interactions. The most likely outcome of this is the reconstruction of two or more vertices: one for the neutrino interaction and one or more for the secondary interactions. By interrogating the associated truth information, each vertex will be matched to the same signal neutrino. It is wholly incorrect to classify each vertex as a reconstruction of a signal event as this will lead to a gross overestimate of the signal rate. To alleviate this, a reconstructed vertex is only tagged as coming from a signal interaction if it obeys the following conditions:

- The reconstructed vertex is matched to a signal interaction
- The reconstructed vertex is the one closest in space to the matched signal interaction

Any reconstructed vertices which pass the first condition but fail the second are tagged as a special case of background events (hereafter referred to as the 'split signal' background).

To illustrate the benefit of separating events into specific prong topologies, Fig. 2.16 shows the truth makeup of events seen in the ECal after applying the reconstruction separation. It is important to note that only the vertex reconstruction and track merging has been applied to the sample at this point. There is a clear difference in the level of backgrounds seen in each topology e.g. the ECal OOFV background is almost

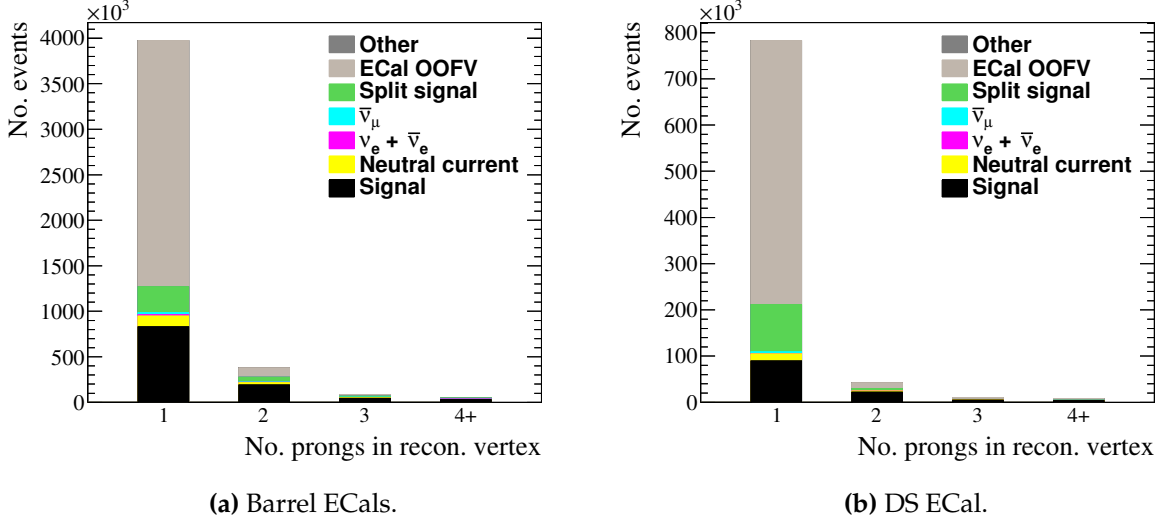


Figure 2.16: The number of reconstructed events in the Monte Carlo sample, separated into the prong topologies. Each event is categorised by the associated truth information from the simulation.

exclusively contained in the one prong topology. By applying such a separation and then focusing cuts on each topology, the overall event purity should be higher.

2.4.1 Selection cuts

The enhanced reconstruction was developed with an ECal cross-section analysis in mind. So, most of complex work should already have been handled by the reconstruction aspect of the analysis. In addition, the selection method discussed involves separating events into prong topologies and focusing individual selections on each. These points motivate a simple, cut-based, selection. Some cuts will inevitably require some tuning and a metric is often useful for this purpose. As demonstrated in section 2.2, a high level of stats is seen in each ECal module which means the selection can safely strive for quality over quantity without inflating the final uncertainty. Bearing this in mind, the metric used for tuning for prong topology i is

$$\phi_i^{\text{selection}} = \epsilon_i \eta_i^2, \quad (2.11)$$

where ϵ_i and η_i are the selection efficiency and purity of prong topology i . Six selection cuts have been identified which will now be discussed.

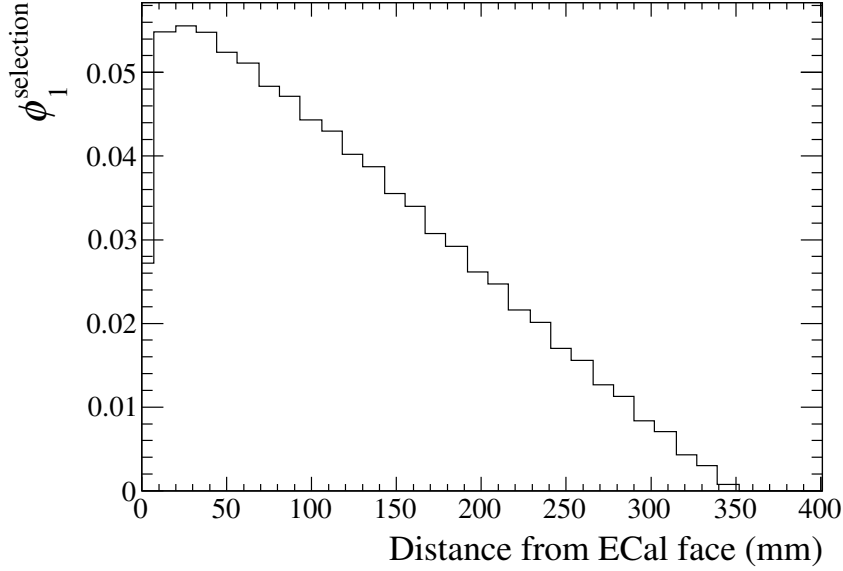


Figure 2.17: $\phi_1^{\text{selection}}$ as a function of the distance from the $-y$ face of the bottom-left barrel ECal for the 1 prong topology.

2.4.1.1 Fiducial volume cut

The first cut, the fiducial volume cut, removes most of the ECal OOFV backgrounds in the Monte Carlo sample. This is achieved by defining some outer veto region in which occurring vertices are rejected. There are two items to bear in mind when defining what the fiducial volumes are. Firstly, the off-axis configuration causes each ECal to be exposed to a different energy and particle rate. A good example of this was shown in Fig. 2.1 which shows a higher event rate, but also a higher peak energy of interactions, in the bottom-left ECal than in the top-right ECal. So, the optimised fiducial volume of one ECal need not be the same as another ECal module. Separately to this, Fig. 2.16 shows that the ECal OOFV is almost completely contained in the 1 prong topology bin. This means that a separate fiducial volume definition should be used for the 1 prong topology (it is sufficient for the 2,3 and 4+ prong topologies to use the same fiducial volume definition).

The fiducial volume for a prong topology/ECal module is defined in terms of the distance from each face of the module. So, there are six numbers which define the fiducial volume: the distance from the $\pm x, \pm y$ and $\pm z$ faces of a module. If the distance of a vertex from each ECal face is further than what the fiducial volume defines, the event passes the cut, otherwise it is rejected. To tune these values, each

	$-x$ (mm)	$+x$ (mm)	$-y$ (mm)	$+y$ (mm)	$-z$ (mm)	$+z$ (mm)
Bottom-right	36	36	31	29	206	28
Side-right	31	29	46	44	206	28
Top-right	39	27	29	31	172	28
Bottom-left	25	50	31	29	205	28
Side-left	29	31	50	25	207	28
Top-left	21	34	29	31	207	28
Downstream	29	29	29	30	41	43

Table 2.4: The fiducial volume definitions for each ECal module for the 1 prong topology.

fiducial volume value was systematically increased from 0 mm to 600 mm in 1 mm increments. At each increment the number of passing/rejected events was recorded and $\phi_i^{\text{selection}}$ calculated. The fiducial volume value which maximises $\phi_i^{\text{selection}}$ is accepted as the optimised value. An example of the variation in $\phi_i^{\text{selection}}$ is shown in Fig. 2.17 which shows $\phi_i^{\text{selection}}$ as a function of the distance from the $-y$ face of the bottom-left barrel ECal for the 1 prong topology. There is a clear maximum found which shows the process is effective in finding an optimum distance from the ECal face.

The found fiducial volumes for each ECal module for the 1 prong topology are shown in table 2.4. While all of the ECal face cuts have some discriminatory power, those which cut out according to layer number reject the most background events. Those cuts are the $\pm y$ cuts for the bottom/top barrel ECals, the $\pm x$ cuts for the side ECals and the $\pm z$ cuts for the DS ECal. In the barrel ECals, the inner and outer two layers are removed by the fiducial volume cut and in the DS ECal, the inner and outer three layers are removed. As described above, the same fiducial volumes are used for the

2,3 and 4+ prong topologies. So, to tune the fiducial volume cuts, the topologies are temporarily combined and the combined sample is used to tune the cuts. Those fiducial volume cuts are shown in table 2.5. There are clear differences between the fiducial volumes shown in table 2.4 and table 2.5. The cuts suggested by the 2,3 and 4+ prong topology tunings are not as strict as those suggested by the 1 prong topology. This should be expected, as ECal OOFV backgrounds reconstructed as 2,3 or 4+ prong vertices will most likely not have their vertex reconstructed near the faces of an ECal module. The one exception to this is the downstream face of each ECal module. It is clear from table 2.4, that the 2,3 and 4+ prong topologies suggest a stricter downstream

face cut ($+z$ cut) than the 1 prong topology. The reason for this strict cut is due to the nature of the neutrino beam. J-PARC's neutrino beam is (almost) parallel with the $+z$ axis defined by the ND280 coordinate system. Because of the very forward nature of the beam, any final state particles from ECal neutrino interactions are most likely to travel downstream. So, any subsequent secondary interactions, which are tagged as background, will also occur downstream. The $+z$ face cut attempts to remove these.

2.4.1.2 Visible energy cut

The second cut rejects events based on the amount of deposited charge which is associated to the constituent prongs. The focus of this cut is to remove particle showers (typically e^\pm and γ) which have been reconstructed as a vertex. For the one prong topology, this information is trivial to evaluate: find which scintillator hits are associated to the single prong and sum their respective charge deposit. The situation is not as clear for a vertex with multiple prongs associated. The variable chosen in this case is the total charge associated to all of the constituent prongs. To further increase the power of this variable as a discriminator, the total prong charge is compared to the total number of scintillator hits associated to the prongs. The aim is to find a 2 dimensional cut which rejects events based on their associated charge and hit information.

The signal and background distributions for the barrel ECal, 1 prong topology are shown in Fig. 2.18a and Fig. 2.18b respectively. To illuminate the separation power, the background/signal ratio is shown in Fig. 2.18c, with the ratio truncated at 2. By

	$-x$ (mm)	$+x$ (mm)	$-y$ (mm)	$+y$ (mm)	$-z$ (mm)	$+z$ (mm)
Bottom-right	19	9	16	0	3	39
Side-right	20	0	6	9	7	65
Top-right	15	9	4	3	5	38
Bottom-left	11	1	34	0	9	69
Side-left	1	4	13	1	3	67
Top-left	11	0	0	7	4	45
Downstream	13	9	13	4	0	64

Table 2.5: The fiducial volume definitions for each ECal module for the 2,3 and 4+ prong topologies.

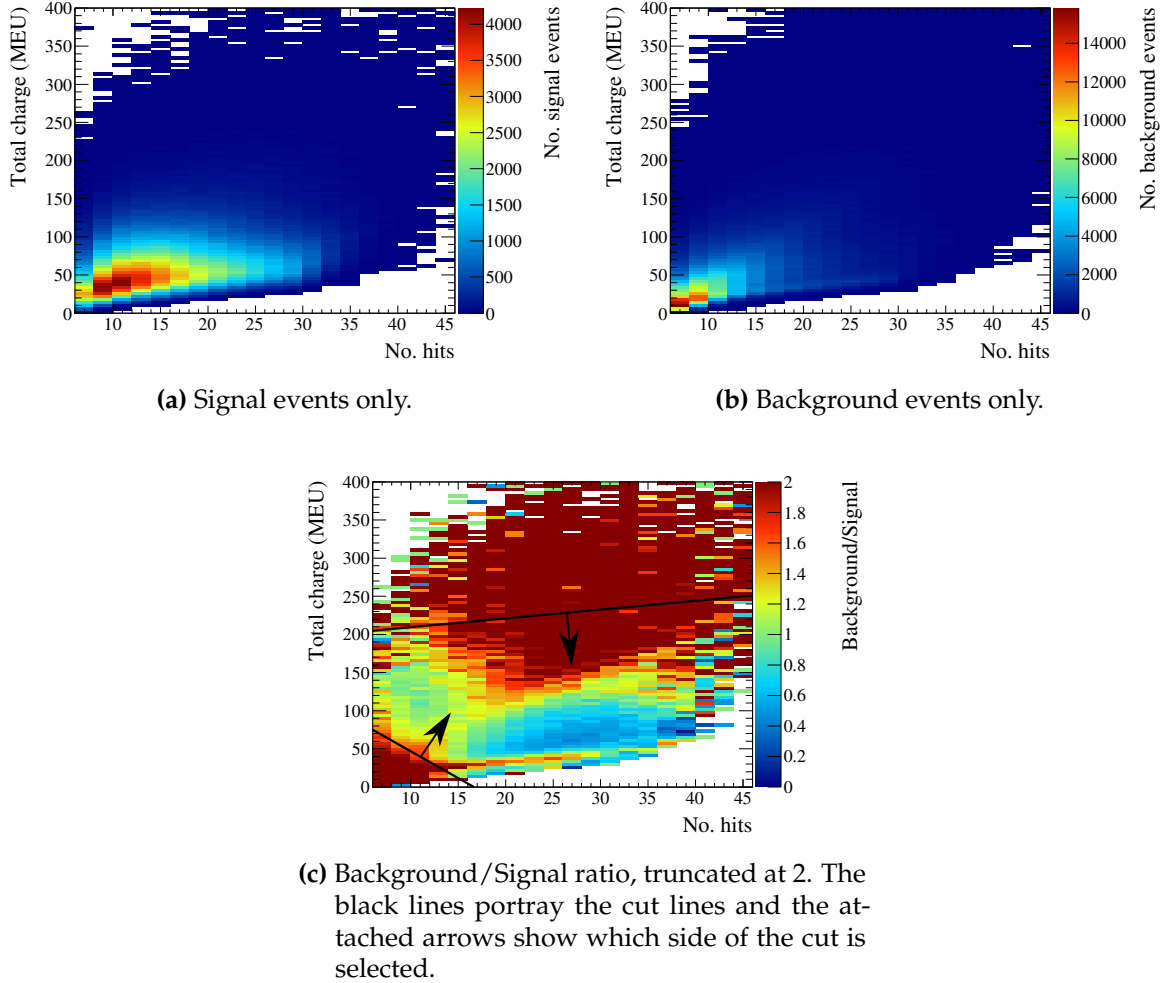


Figure 2.18: Total prong charge vs the number of prong hits for barrel ECal events in the 1 prong topology.

applying the truncation, areas where the background contamination is at least two times as large as the signal population is easily found. To develop the 2 dimensional cuts, a test line is drawn which appears to roughly cut out the highly contaminated (red) areas in Fig. 2.18c. Then, to tune the cut line, the parameters of said line are varied in a grid search. At each point in the grid, $\phi_i^{\text{selection}}$ is recorded. The maximum value in this grid corresponds to the optimised parameters of the cut line. As is evident from Fig. 2.18c, there are two areas of high background contamination which motivated the development two cut lines which are both overlaid on Fig. 2.18c.

Fig. 2.19 and Fig. 2.20 show similar distributions for the 2 and 3 prong topology in the barrel ECal. The ratio distributions also have the cut lines overlaid. An identical process was used to generate and tune the cut lines. By using a definition for the

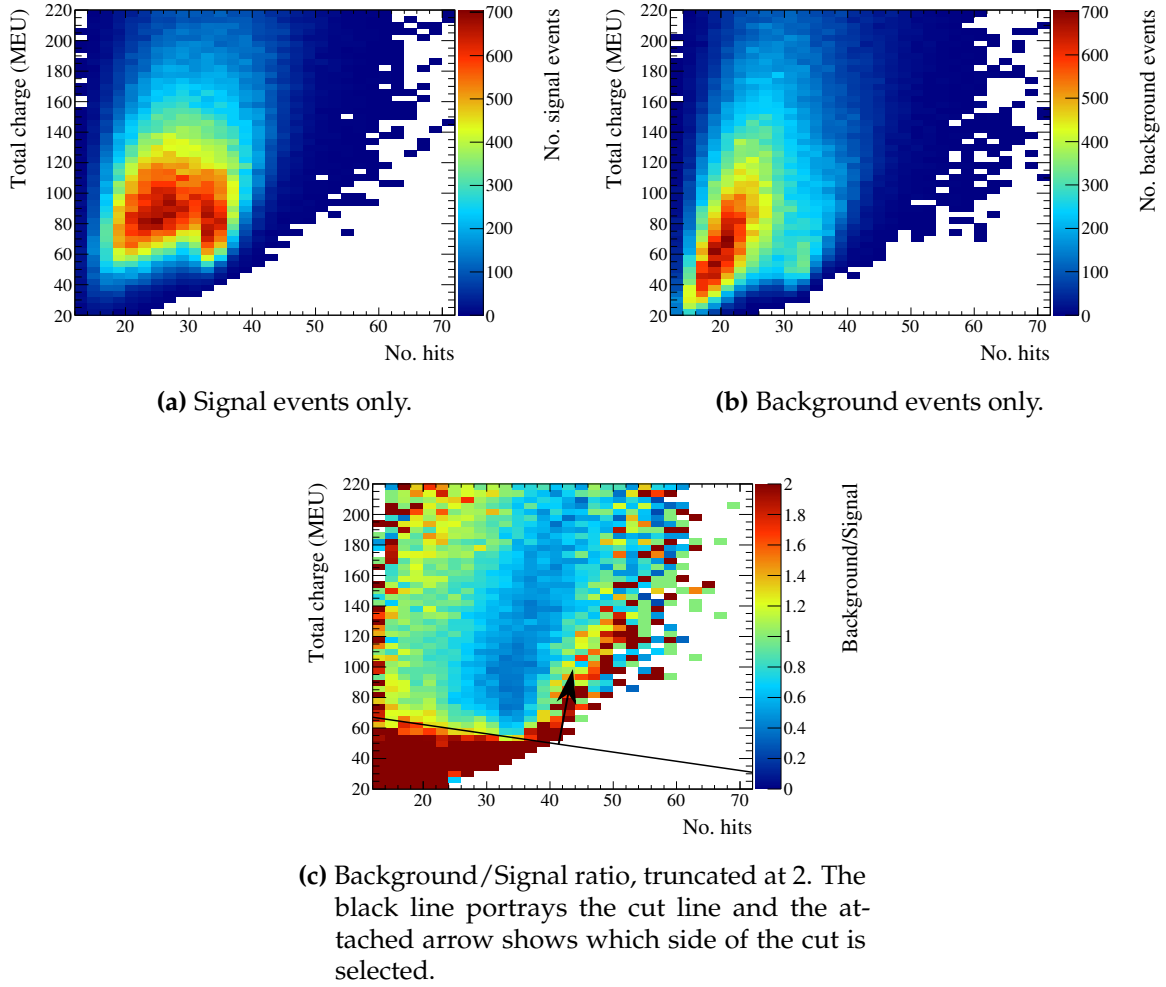
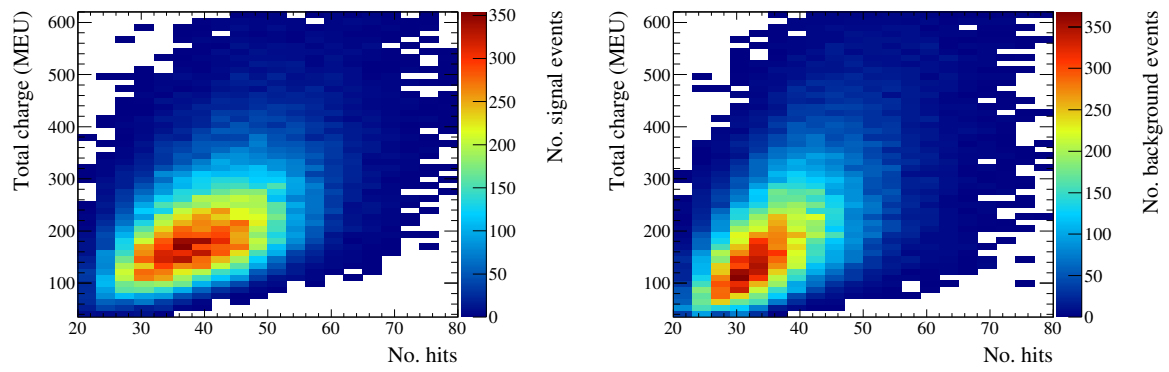


Figure 2.19: Total prong charge vs the number of prong hits for barrel ECal events in the 2 prong topology.

Prong topology	Barrel cuts (MEU)	DS cuts (MEU)
1	$> -4.5N + 75$ $< N + 205$	$> -3.4N + 62$
2	$> -0.5N + 67$	$> 0.2N + 51$
3	$> 0.7N + 103$	$> -1.1N + 117$

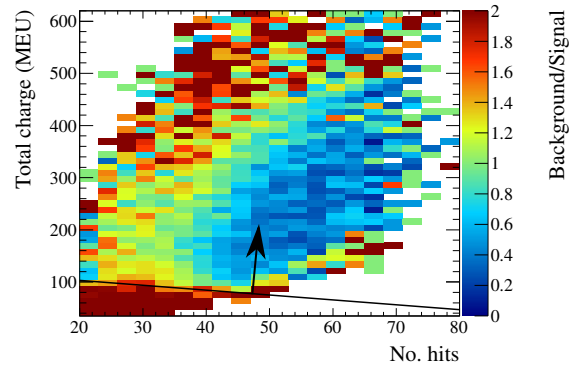
Table 2.6: The definitions of the 2 dimensional cut lines involving the total prong charge and the number of associated scintillator hits. N refers to the total number of scintillator hits associated to the constituent prongs in the vertex.

total number of scintillator hits associated to all constituent prongs in a vertex, N , the definition of the cut lines are shown in table 2.6.



(a) Signal events only.

(b) Background events only.



(c) Background/Signal ratio, truncated at 2. The black line portrays the cut line and the attached arrow shows which side of the cut is selected.

Figure 2.20: Total prong charge vs the number of prong hits for barrel ECal events in the 3 prong topology.

2.4.1.3 Unused hits cut

The output of the enhanced reconstruction is a set of reconstructed clusters. Within those clusters are a set of tracks and their pairwise crossings which are used to reconstruct the ECal vertices used by this analysis. The clusters which contain the tracks and crossings can also be utilised as a discriminator. Particle showers should generally produce a lot of hits and those hits are not necessarily arranged in a linear fashion. So, the third cut implemented assesses how many hits are associated with a cluster but not associated with the vertex. The vertex is defined by its prong constituents so this is really a measure of how many scintillator hits were not associated with the reconstructed prongs. By defining the number of hits in the cluster as N^{Cluster} and the number of hits associated with the i^{th} constituent prong as N_i^{prong} , the discriminator is

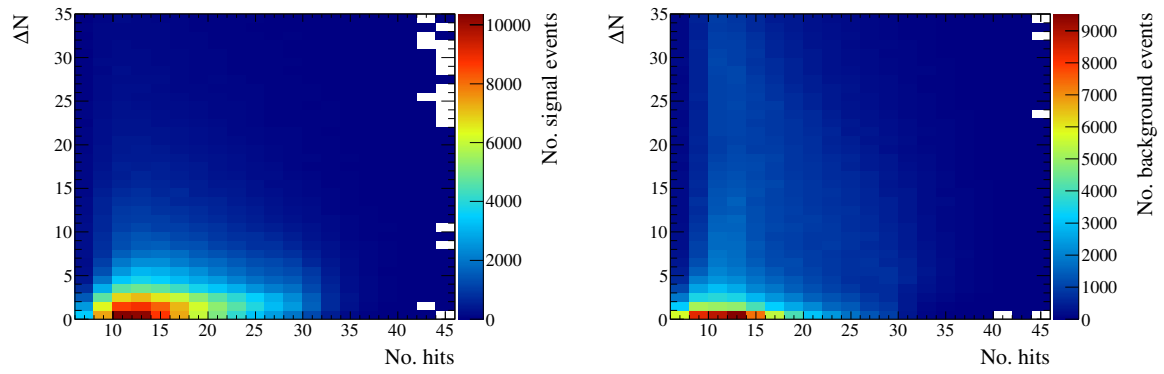
$$\Delta N = N^{\text{cluster}} - \sum_i N_i^{\text{prong}}. \quad (2.12)$$

The cut value really should depend on how many hits are associated to the constituent prongs. So, as was done in section 2.4.1.2, the ΔN cut should be 2 dimensional. Specifically, the ΔN cut should depend on the number of scintillator hits associated to the constituent prongs.

So, an identical approach is used to define and tune the cut lines as was used in section 2.4.1.2. The signal, background and background/signal ratio distributions for the 1, 2 and 3 prong topologies in the barrel ECal are shown in Fig. 2.21, Fig. 2.22 and Fig. 2.23 respectively. The layout of the figures is the same as was shown for the visible energy cut. In a similar manner to table 2.6, the ΔN cut definitions are shown in table 2.7.

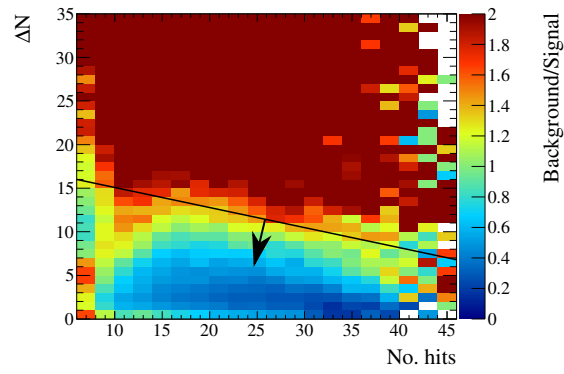
Prong topology	Barrel cuts (no units)	DS cuts (no units)
1	$< -0.2N + 16$	$< 1.2N + 3$
2	$< 1.5N - 16$	$< 0.7N - 4$
3	$< 1.7N + 7$	$< 0.8N + 1$

Table 2.7: The definitions of the 2 dimensional cut lines involving ΔN and the number of scintillator hits associated with the constituent prongs. N refers to the total number of scintillator hits associated to the constituent prongs in the vertex.



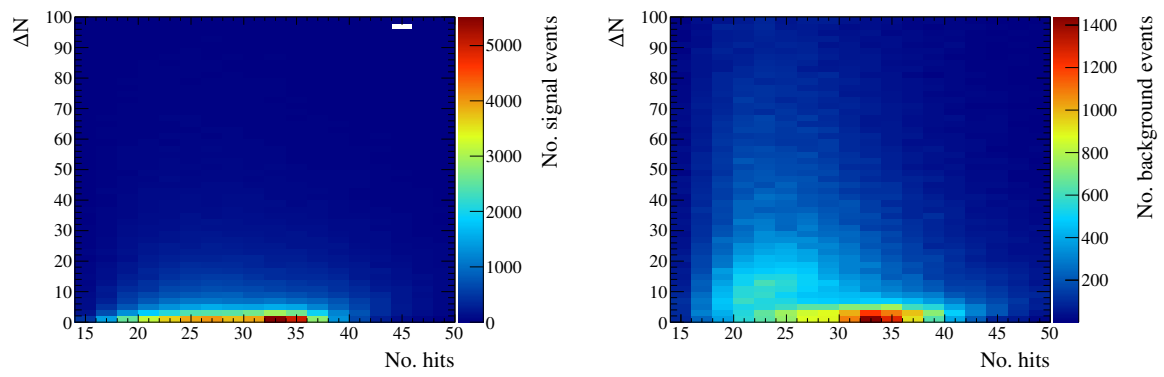
(a) Signal events only.

(b) Background events only.



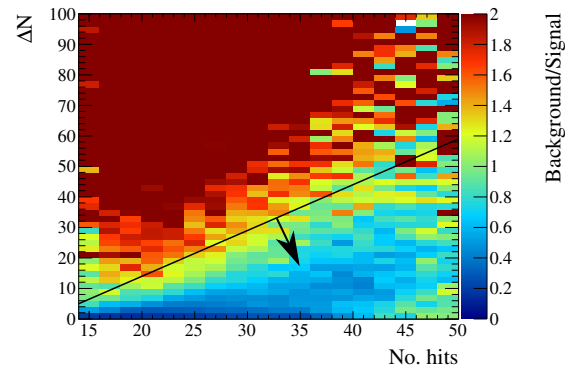
(c) Background/Signal ratio, truncated at 2. The black line portrays the cut line and the attached arrow shows which side of the cut is selected.

Figure 2.21: ΔN vs the number of prong hits for barrel ECal events in the 1 prong topology.



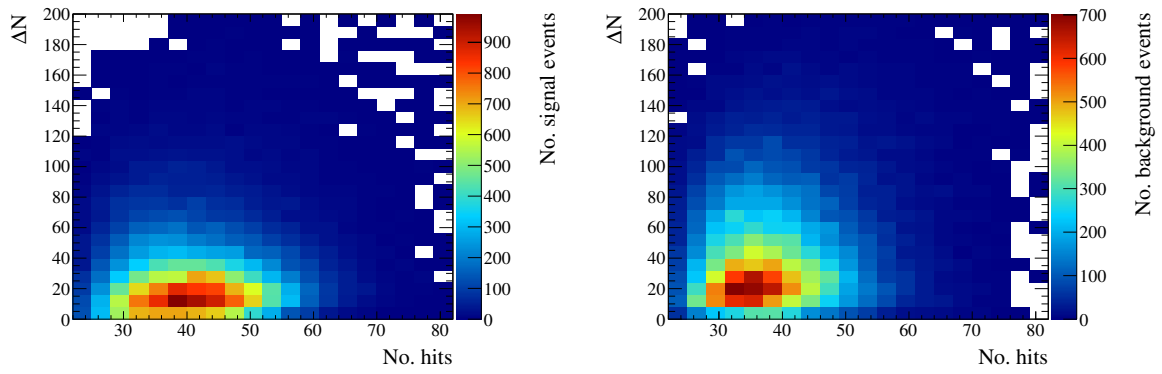
(a) Signal events only.

(b) Background events only.



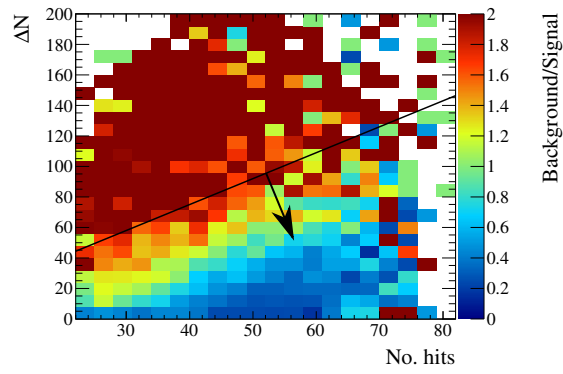
(c) Background/Signal ratio, truncated at 2. The black line portrays the cut line and the attached arrow shows which side of the cut is selected.

Figure 2.22: ΔN vs the number of prong hits for barrel ECal events in the 2 prong topology.



(a) Signal events only.

(b) Background events only.



(c) Background/Signal ratio, truncated at 2. The black line portrays the cut line and the attached arrow shows which side of the cut is selected.

Figure 2.23: ΔN vs the number of prong hits for barrel ECal events in the 3 prong topology.

2.4.1.4 Opening angle

Despite the best efforts of the track merging, not all curving trajectories have reconstructed tracks which are successfully merged. So, there will be an inherent ECal OOFV contamination of the 2,3 and 4+ prong topologies where the reconstruction has failed. While future iterations of the analysis should revisit the reconstruction to combat this, it is sufficient to cut these background events out in the selection. As was described in section 2.3, a key signature of a curving trajectory is two prongs with a small opening angle. This means that the opening angle can be used as a cut to remove such background events. To define an opening angle, at least two tracks are required so, unfortunately, such a method can not be implemented for the 1 prong topology.

The signal and background separation provided by the subtended opening angle for the 2 prong topology is shown in Fig. 2.24. While the barrel ECal sees more separation power in the opening angle, it should be clear from Fig. 2.24a and Fig. 2.24b that events should only be selected if their opening angle is bigger than some threshold. To find this threshold, a test cut was systematically varied between 0° and 90° in 1° increments. At each increment, the number of events with an opening angle larger than the test cut was recorded and $\phi_2^{\text{selection}}$ subsequently calculated. An example of the variation in $\phi_2^{\text{selection}}$ is shown in Fig. 2.25 for the barrel ECal. A maximum in $\phi_2^{\text{selection}}$ is clearly found which shows that the opening angle is a good discriminator and that the method works. By applying this method to 2 prong events, events are

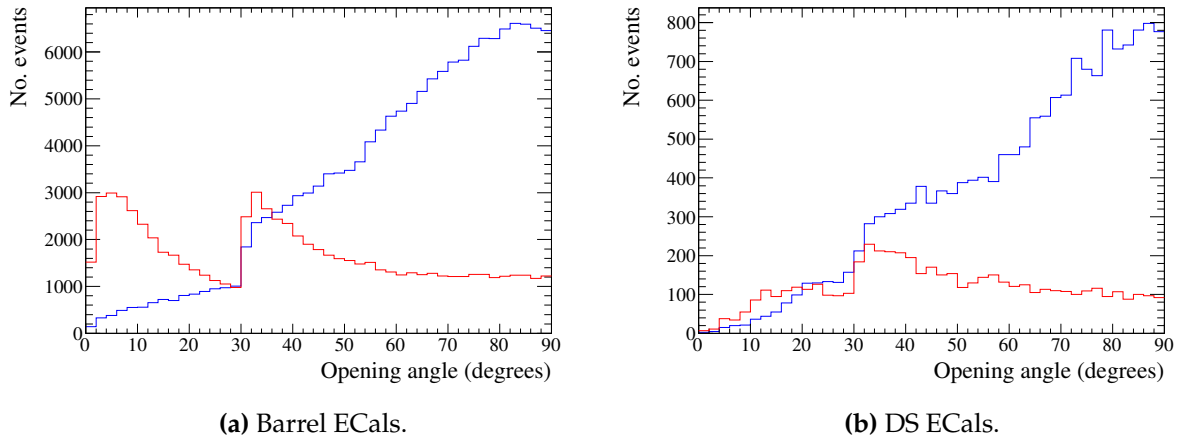


Figure 2.24: The opening angle subtended by the constituents prongs for the 2 prong topology. Signal and background events are the blue and red histograms respectively.

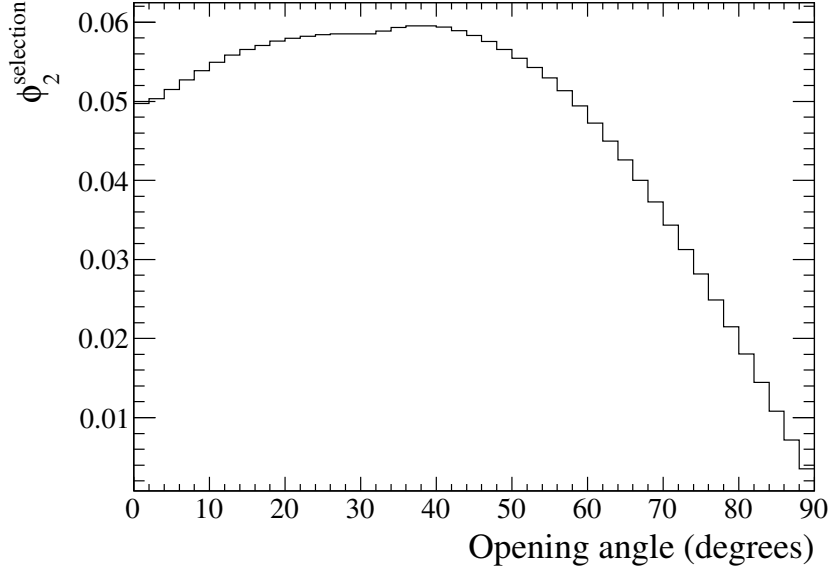


Figure 2.25: $\phi_2^{\text{selection}}$ as a function of the opening angle cut for the 2 prong topology in the barrel ECal.

only selected if the subtended opening angle is greater than 34° for the barrel ECal or greater than 21° for the DS ECal. Similar information should be available for the higher prong topologies. However it is not as trivial to define an opening angle in such situations. The chosen variable for the 3 prong topology is the summed opening angle for every pairwise combination of the constituent prongs. The signal and background separation for this summed opening angle is shown in Fig. 2.26. The cut development process used here was identical to the one used for the 2 prong topology. The study suggested that events should only be selected if their summed opening angle is greater than 81° for the barrel ECal and 49° for the DS ECal. The cuts found by this study are summarised in table 2.8.

Prong topology	Barrel cuts (degrees)	DS cuts (degrees)
2	> 34	> 21
3	> 81	> 49

Table 2.8: The definitions of the opening angle cuts.

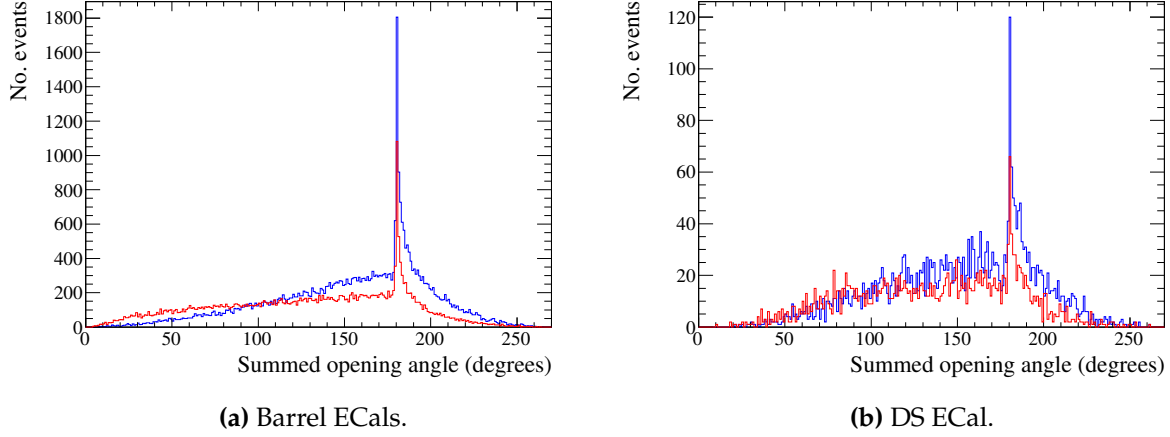


Figure 2.26: The summed opening angle subtended by every pairwise combination of the constituents prongs for the 3 prong topology. Signal and background events are the blue and red histograms respectively.

2.4.1.5 Entering background cut

The fifth cut, simply called the ‘entering background’ cut, uses the prong multiplicity aspect of a vertex to reject events that enter the ECal and then mimic a signal vertex. The cut requires at least two constituent prongs to work, so the 1 prong topology is unaffected by this. The idea is fairly simple: for each vertex, check if any of the prongs have an end which is upstream of the reconstructed vertex position. If so, it may be an entering background and could need rejecting. The position of the upstream prong end is key, as the cut should not reject every event which has an apparent backwards going track. So, the cut checks whether the upstream prong end is outside of the host ECal fiducial volume. As described in section 2.4.1.1, each ECal module uses two fiducial volumes. The 2+ prong fiducial volume would be of little use in this situation as the only strict face definition is for the downstream end of each ECal module. This motivates the use of the 1 prong fiducial volume for this cut. This cut is physically motivated and relies on parameters which have already been tuned. So, this cut requires no tuning to operate.

2.4.1.6 Most upstream vertex

The sixth and final cut implemented is motivated by the beam kinematics. As discussed in section 2.2, the expected pileup per beam bunch is small. As the reconstruction

outputs events in buckets, the same philosophy can be used here; there should only be one signal event per bucket. The assumption used for this cut is any secondary interactions are likely to happen downstream of the interaction. So, the cut selects the most upstream remaining vertex in the beam bucket. As with the 'entering background' cut, the selection of the most upstream vertex requires no tuning to operate.

2.4.2 Performance of the selection

Now that the cuts have been identified, their performance can be assessed. To do this, the Monte Carlo sample was processed through the selection and the event composition assessed at each stage. As each cut was focused on a specific topology, it is important to check the effect of the cuts on each topology. The first check looks at the event survival as a function of the cuts for each topology. These checks are shown in Fig. 2.27 and Fig. 2.28 for the barrel and DS ECals respectively. Each bin in both figures shows how many events survive after the cut has been applied. For example, the FV bin in all of the figures shows how many events remain after applying the fiducial volume cut. As the 'most upstream' cut is the final cut in the selection, the final bin shows how many events remain after the full selection has been applied. The main piece of information shown is that the backgrounds are being rejected at a much higher rate than the signal, resulting in a selected sample that is mostly signal.

While Fig. 2.27 and Fig. 2.28 shows that the cuts are successfully rejecting background while retaining signal, it is not clear what particle species are rejected. It is important to check this so that the developed selection is understood. The background type which is rejected at the highest rate is the ECal OOFV background so it is sufficient to study the particle composition of this to understand the selection's ability to reject particles. The particle type is defined by looking at which simulated particles contributed to the ECal cluster as whole. The particle which deposited the highest amount of energy is accepted as the one which created the ECal cluster. The ECal OOFV event survival as a function of the cuts is shown in Fig. 2.29 and Fig. 2.30 for the barrel and DS ECal respectively. For each prong topology, the ECal OOFV events are separated into the main particle types which constitute the ECal OOFV background. It is clear that the cuts are rejecting the particle species that one would expect them to reject. For example, the fiducial volume cut in the 1 prong topology almost exclusively rejects entering MIPs. The visible energy and ΔN cuts work together as an effective particle shower

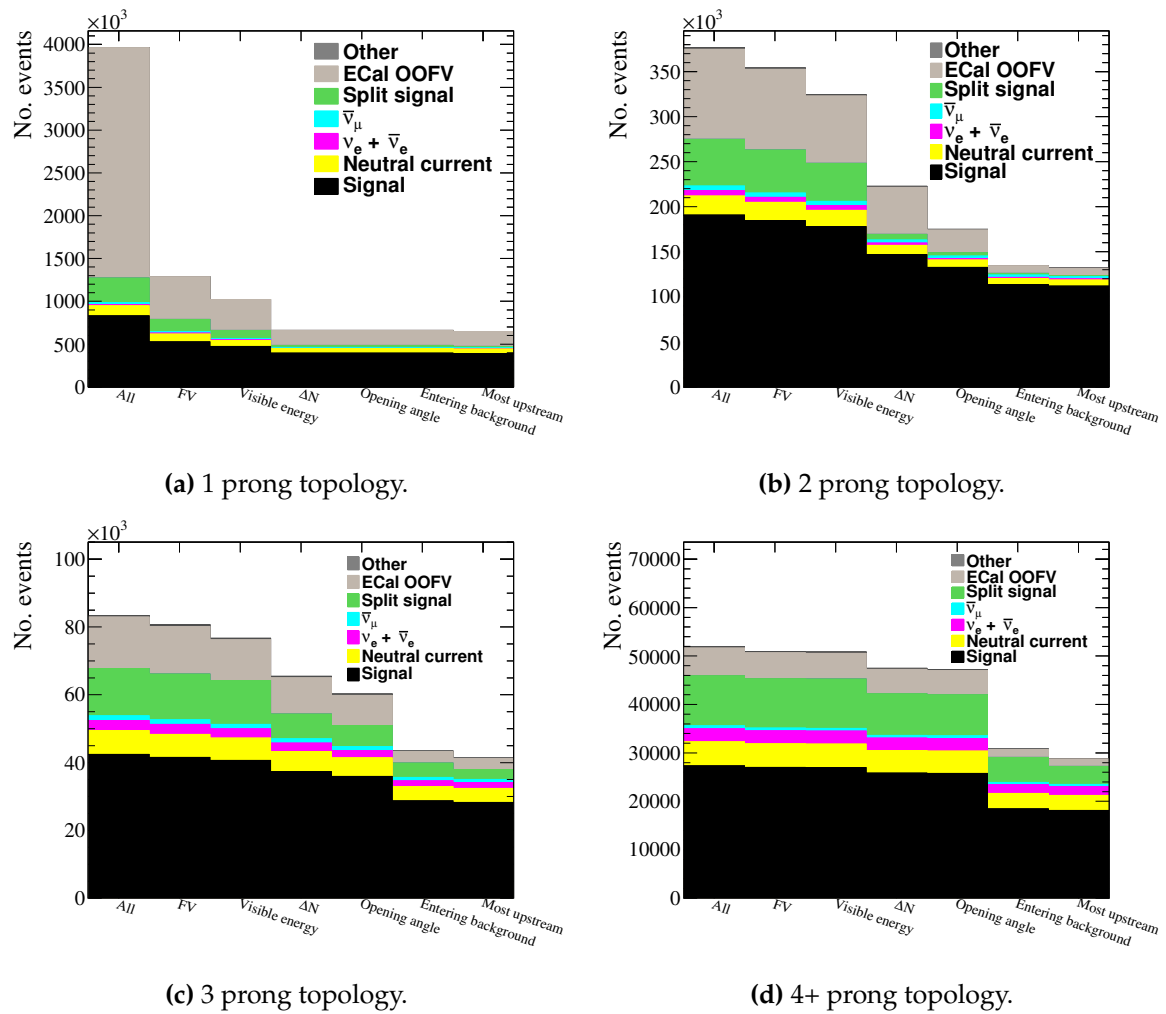
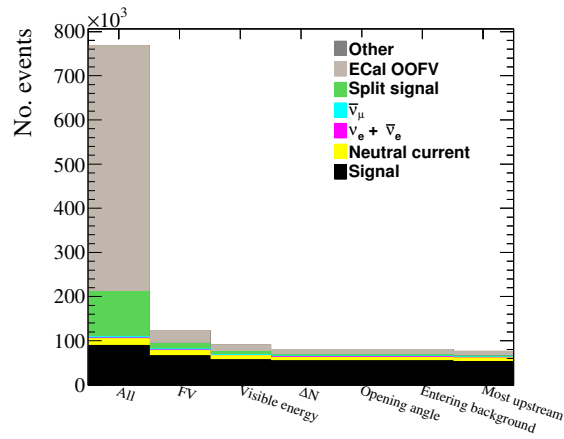
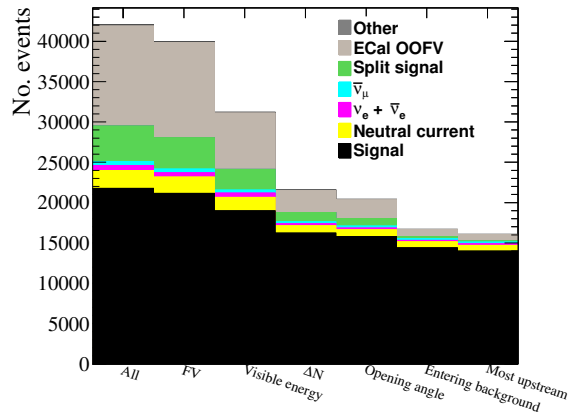


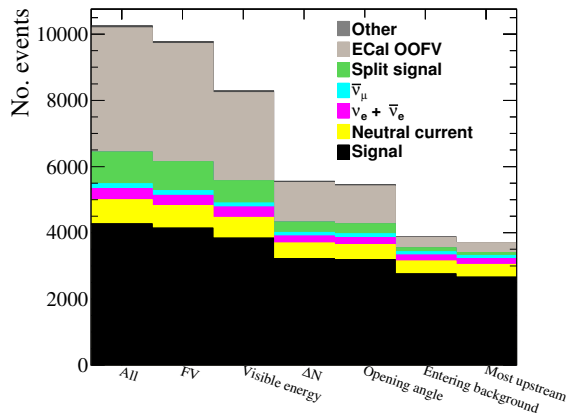
Figure 2.27: Event survival as a function of the cuts for all topologies in the barrel ECal. Each bin refers to the number of events after a cut has been applied.



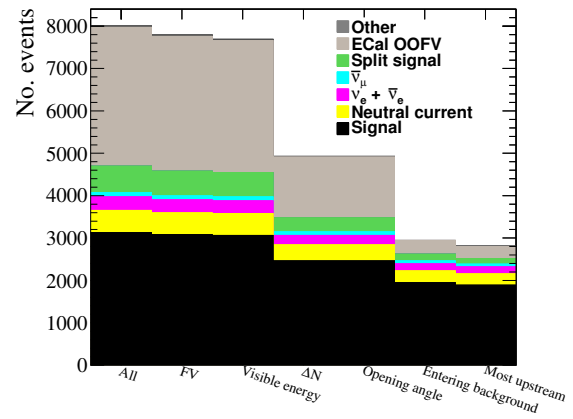
(a) 1 prong topology.



(b) 2 prong topology.



(c) 3 prong topology.



(d) 4+ prong topology.

Figure 2.28: Event survival as a function of the cuts for all topologies in the DS ECal. Each bin refers to the number of events after a cut has been applied.

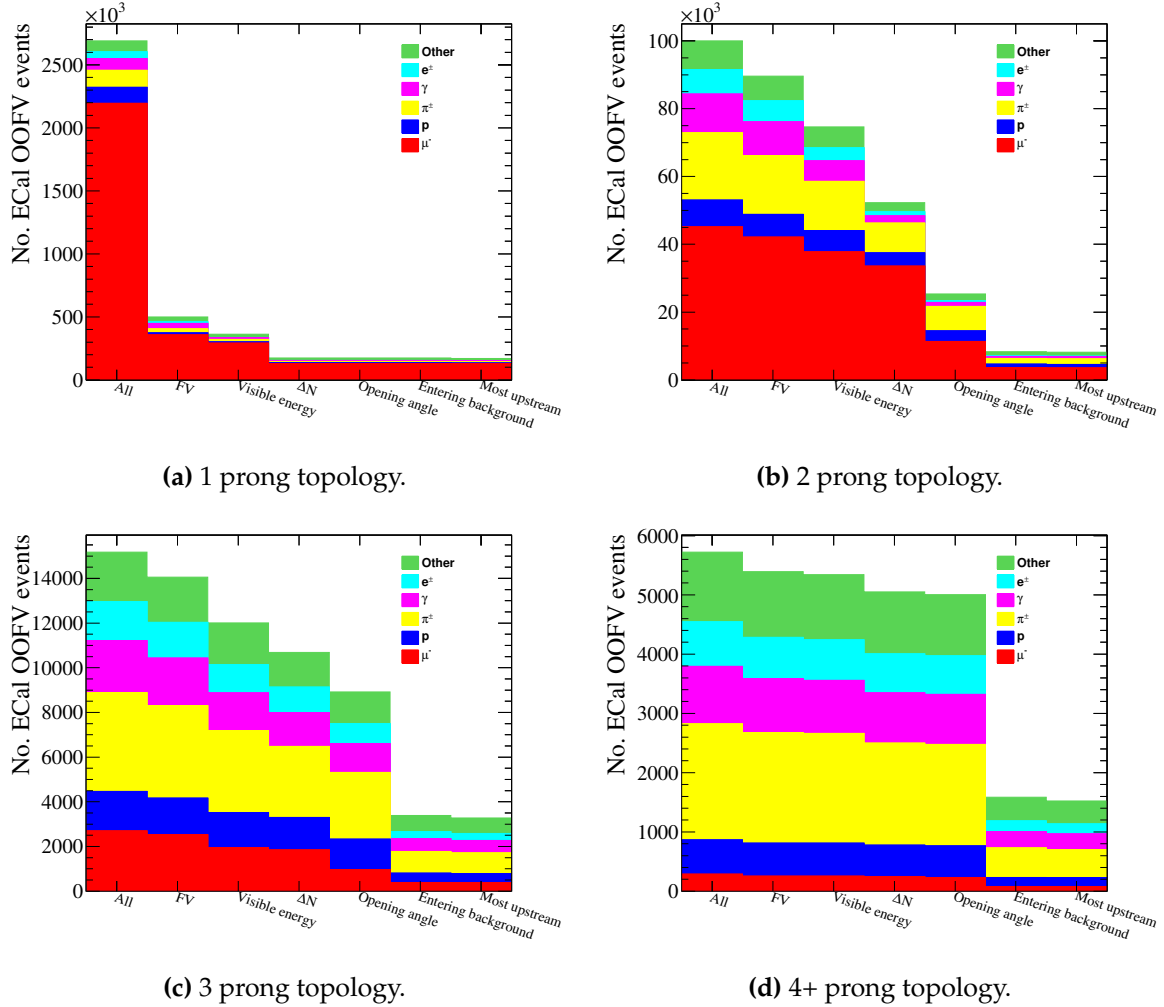
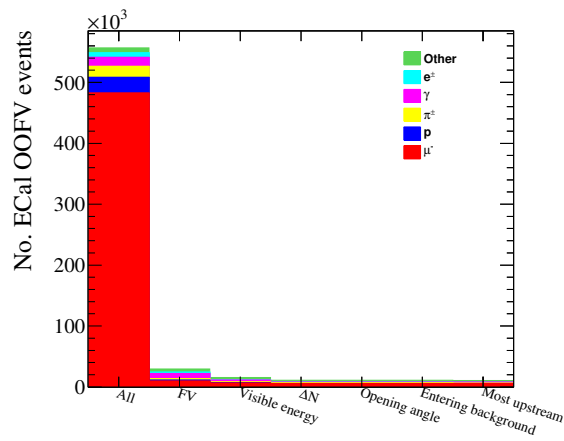


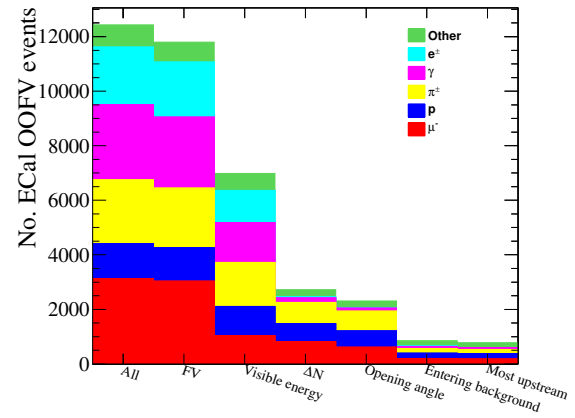
Figure 2.29: ECal OOFV event survival as a function of the cuts for all topologies in the barrel ECal. Each bin refers to the number of events after a cut has been applied.

tagger to reject e^\pm and γ particles. The opening angle and entering background cuts work as an additional MIP rejector in cases where the track merging was unsuccessful. The composition of the ECal OOFV background has interesting features, regardless of the cuts applied. Specifically, the particle species varies to a great degree between each prong topology. For example, the 1 prong and 2 prong topologies in the barrel ECals (Fig. 2.29a and Fig. 2.29b) see a muon dominated ECal OOFV background. Whereas, the 3 prong and 4+ prong topologies see a very different background which is π^\pm dominated. The natural separation provided by the topologies should allow investigation of a specific particle species if required by any future analyser.

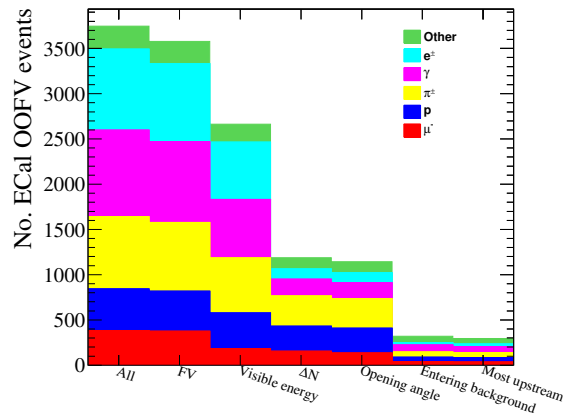
The selection efficiency and purity as a function of the cuts are shown for the barrel



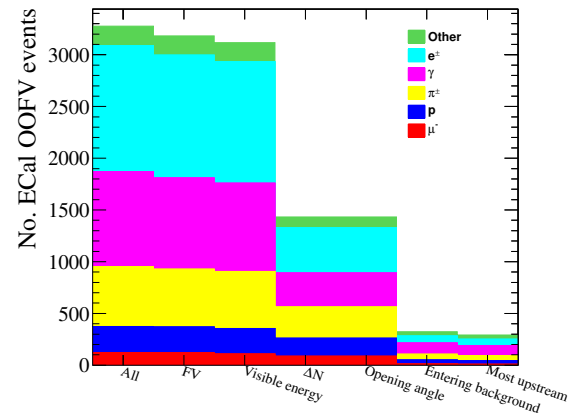
(a) 1 prong topology.



(b) 2 prong topology.



(c) 3 prong topology.



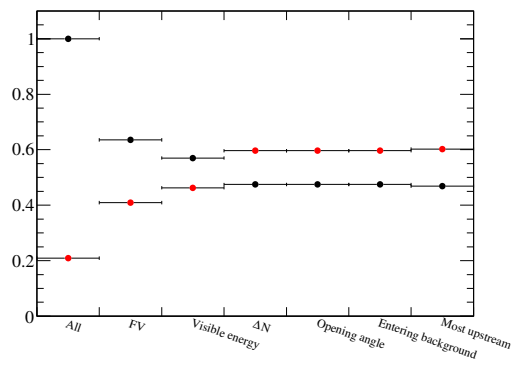
(d) 4+ prong topology.

Figure 2.30: ECal OOFV event survival as a function of the cuts for all topologies in the DS ECal. Each bin refers to the number of events after a cut has been applied.

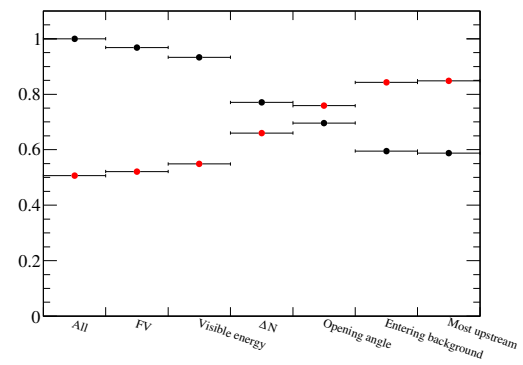
ECal and DS ECal in Fig. 2.31 and Fig. 2.32 respectively. Every cut which is relevant to a prong topology has an expected effect, the purity of the sample increases while the efficiency decreases. What is not so clear from the distributions is whether the tuning metric, $\phi^{\text{selection}}$, increases as a function of the cuts. So, $\phi^{\text{selection}}$ is shown in a similar manner for the barrel and DS ECal in Fig. 2.33 and Fig. 2.34 respectively. The tuning metric optimisation in both the barrel and DS ECal shows some interesting features. Firstly, in the barrel, it is clear that for the 1 prong and 2 prong topologies, every applied cut shows a clearly increasing $\phi^{\text{selection}}$. Unfortunately, this is not the case for the 3 prong and 4+ prong topologies. Specifically, the entering background cut degrades the value of $\phi^{\text{selection}}$. This is most obvious in the 4+ prong topology; however,¹⁰ it has already been stated that this topology is effectively an overflow bin and is being included more for completeness. What is perhaps more concerning is the degradation of $\phi^{\text{selection}}$ in the 3 prong topology. Fortunately, the decrease in $\phi^{\text{selection}}$ is small when the entering background cut is applied and the change is well within error. It is important to bear in mind that $\phi^{\text{selection}}$ is used as a tuning guide and small deviations from the optimisation process are allowed. So, it was decided that the entering background cut would remain in place for both the 3 prong and 4+ prong topologies. The $\phi^{\text{selection}}$ distribution in the DS ECal shows some similar and some different features to that of the barrel. As with the barrel ECal case, $\phi^{\text{selection}}$ clearly increases as a function of the cuts for the 1 prong and 2 prong topologies. Unlike the barrel ECal, there is also an increase in $\phi^{\text{selection}}$ when applying the entering background cut to the 3 and 4+ prong topologies. However, there is a minor decrease in $\phi^{\text{selection}}$ when selecting the most upstream vertex. It is not wholly surprising that this is the case as the DS ECal is the most downstream detector in ND280. Such a cut can easily cause a bias where events in the barrel are generally preferred because of their extent in the global Z direction. Fortunately, this effect is very minor and also well within error.

The cross-section measurement presented in this analysis is an inclusive measurement. This means that the neutrino interaction cross-section needs to be measured without bias across the J-PARC beam energy range. All of the performance checks presented so far have only assessed each prong topology individually. So, the final check is to assess how the selection performs after combining the topologies together. The selection efficiency and purity after combining the topologies is shown in Fig. 2.35. For both the barrel ECal and DS ECal, there is a clear dependence of the purity on the neutrino

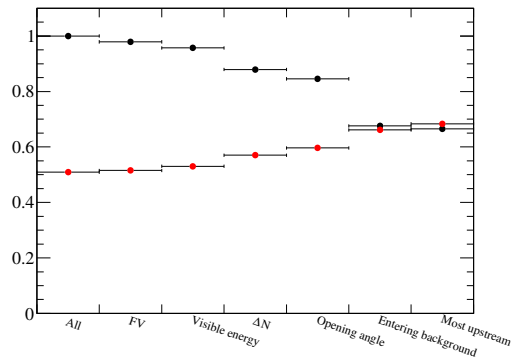
¹⁰ADDRESSED - was “, however”



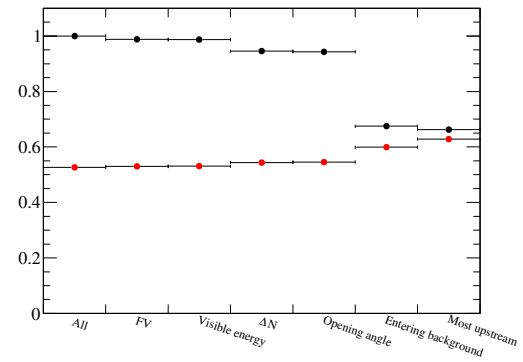
(a) 1 prong topology.



(b) 2 prong topology.

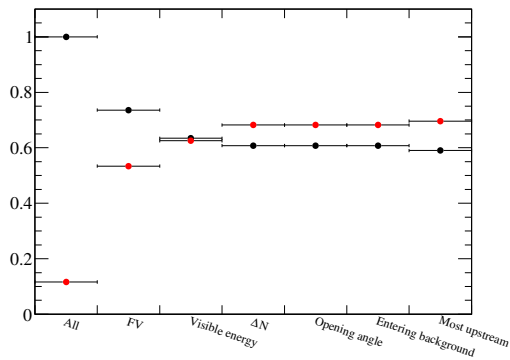


(c) 3 prong topology.

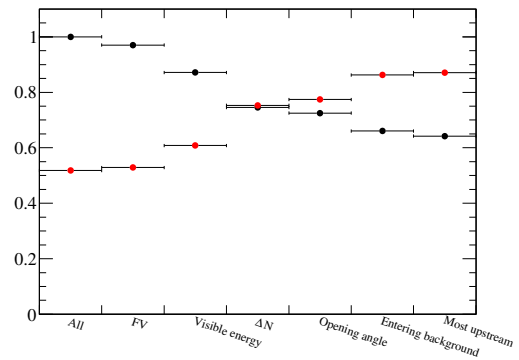


(d) 4+ prong topology.

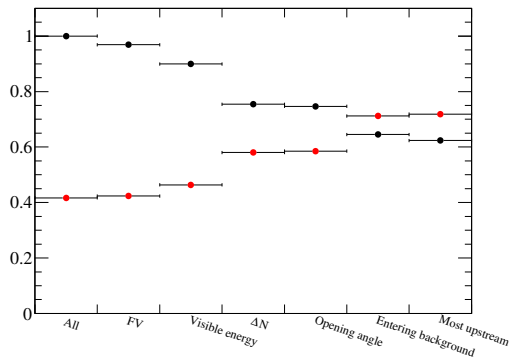
Figure 2.31: The selection efficiency (black) and purity (red) as a function of the cuts in the barrel ECal. Each bin refers to the efficiency and purity after a cut has been applied.



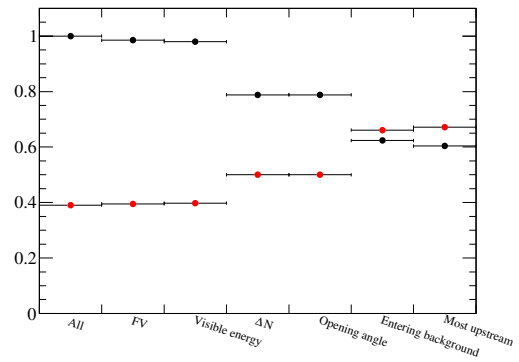
(a) 1 prong topology.



(b) 2 prong topology.

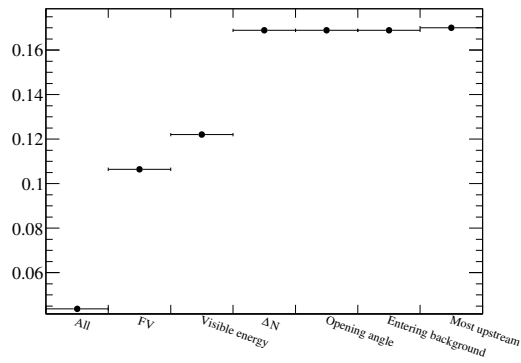


(c) 3 prong topology.

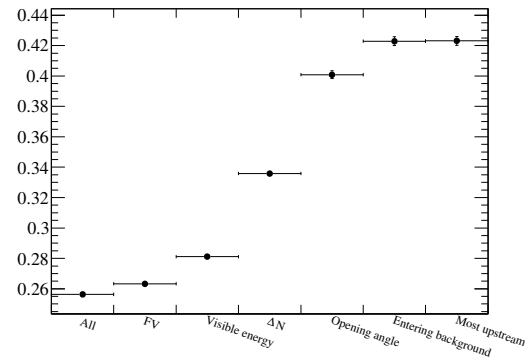


(d) 4+ prong topology.

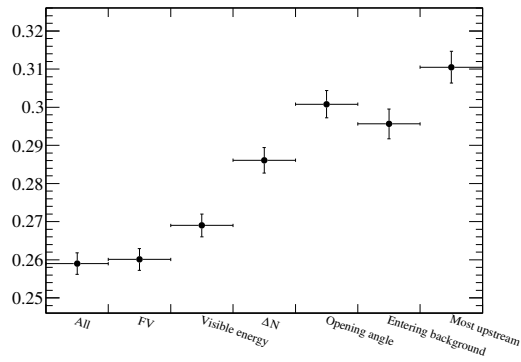
Figure 2.32: The selection efficiency (black) and purity (red) as a function of the cuts in the DS ECal. Each bin refers to the efficiency and purity after a cut has been applied.



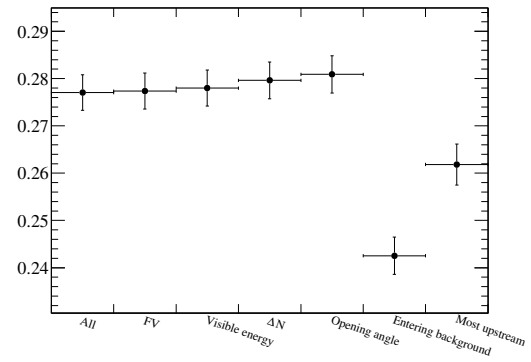
(a) 1 prong topology.



(b) 2 prong topology.

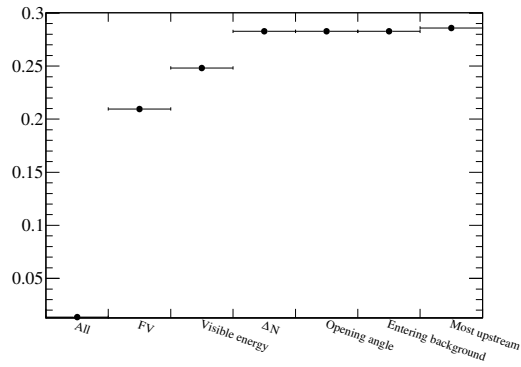


(c) 3 prong topology.

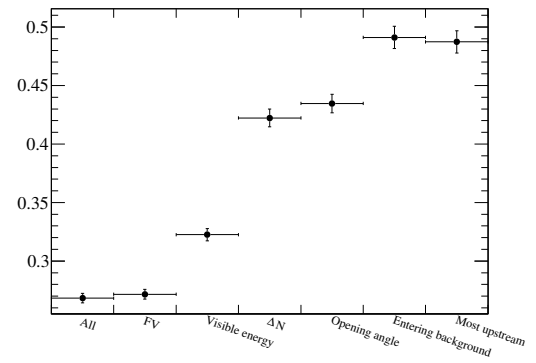


(d) 4+ prong topology.

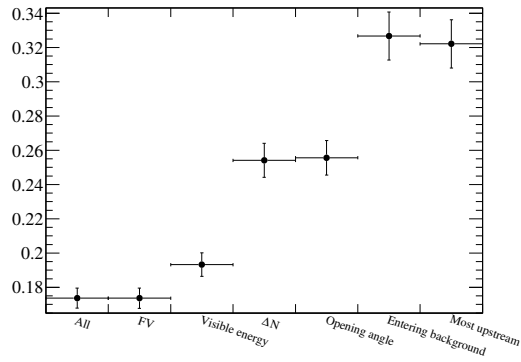
Figure 2.33: $\phi^{\text{selection}}$ as a function of the cuts in the barrel ECal. Each bin refers to the value of $\phi^{\text{selection}}$ after a cut has been applied.



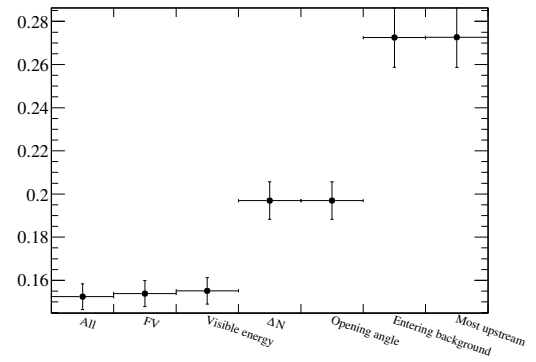
(a) 1 prong topology.



(b) 2 prong topology.



(c) 3 prong topology.



(d) 4+ prong topology.

Figure 2.34: $\phi^{\text{selection}}$ as a function of the cuts in the DS ECal. Each bin refers to the value of $\phi^{\text{selection}}$ after a cut has been applied.

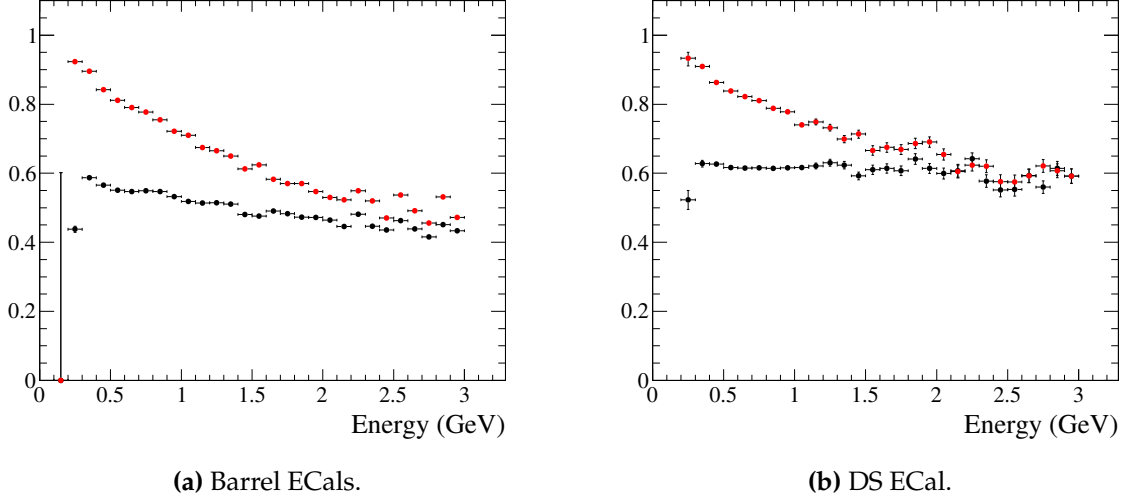


Figure 2.35: The selection efficiency (black) and purity (red) for the combined prong topologies as a function of neutrino energy.

energy. However, this should be expected. It is clear that the purity decreases as the neutrino energy increases which can be explained by the self-shielding effect of the ECal. In the low energy regime, the ECal OOFV background is suppressed because there is insufficient energy to either penetrate the ECal or deposit enough energy to allow cluster reconstruction. As the background energy increases, the suppression is lifted, resulting in a decrease in the selection purity. The key piece of information shown in Fig. 2.35 is the selection efficiency. As described above, it is of vital importance that the selection does not introduce a bias which measures one energy regime more than another. The flatness of the efficiency distributions for both the barrel ECal and DS ECal shows this is not the case which means the selection presented here would be suitable for a CC-inclusive measurement.

The number of selected events for each prong topology is shown in Fig. 2.36, with each topology broken down by truth categories. For both the barrel and DS ECal, signal events dominate each prong topology. The most impure topology for both detectors is the 1 prong topology. This is primarily due to having an insufficient number of prongs to really benefit from what the reconstruction can provide. Despite this, it is clear that the selection results in a pure sample of events.

The selection efficiencies and purities for each prong topology are shown in table 2.9 and table 2.10 respectively. The final purities and efficiencies are generally good.

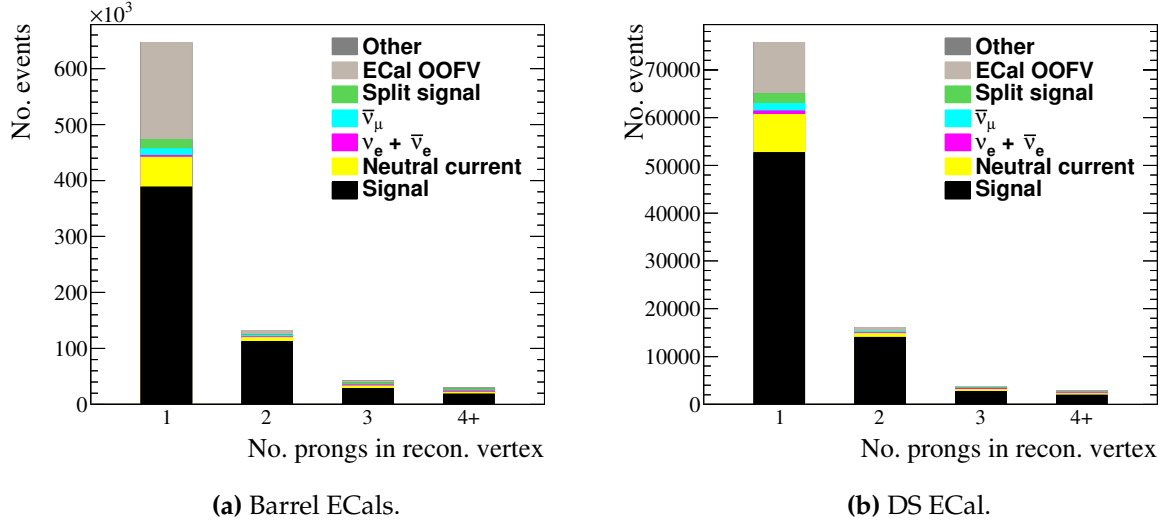


Figure 2.36: The number of selected events in the Monte Carlo sample, separated out into the prong topologies. Each event is categorised by the associated truth information from the simulation.

ECal module	1 prong topology efficiency (%)	2 prong topology efficiency (%)	3 prong topology efficiency (%)	4+ prong topology efficiency (%)
Barrel	46.8	58.7	66.5	66.3
DS	59.0	64.2	62.3	60.4

Table 2.9: The selection efficiencies for each prong topology and ECal module.

The selection efficiency is defined to be 100% when no cuts have been made. There are inevitably signal events which are not reconstructed, primarily because the energy is below reconstruction threshold. It is non-trivial to include these events in an efficiency calculations for a specific prong topology, but it is also unnecessary to do this as the prong topologies are to be summed for the CC-inclusive cross-section measurement.

ECal module	1 prong topology purity (%)	2 prong topology purity (%)	3 prong topology purity (%)	4+ prong topology purity (%)
Barrel	60.3	85.0	68.5	63.0
DS	69.6	87.2	72.0	67.3

Table 2.10: The selection purities for each prong topology and ECal module.

ECal module	Efficiency (%)	Purity (%)
Barrel	42.4	64.4
DS	53.0	72.4

Table 2.11: The topology combined efficiency and purity.

So, the topology combined efficiency and purity are presented in table 2.11. It is these numbers, along with the sample itself, which are the final output of the Monte Carlo selection.

Bibliography

- [1] P. a. Hough, Conf.Proc. **C590914**, 554 (1959).
- [2] R. O. Duda and P. E. Hart, Commun. ACM **15**, 11 (1972).
- [3] J. Vuillemin, Fast linear hough transform, in *Application Specific Array Processors, 1994. Proceedings. International Conference on*, pp. 1–9, 1994.

List of Figures

1.1	Representations of a 2D line in Cartesian space.	4
1.2	Representations of a 2D point in Cartesian space.	5
1.3	The three points defined in equation 1.4 and their representation in the parameter space. The colour coding matches the Cartesian points to their respective parameter lines.	7
1.4	The line represented by the parameter space intersection in Fig. 1.3b with the Cartesian points it intercepts.	8
1.5	$\theta - \rho$ parameterisation of a 2D line.	9
1.6	The discrete $\theta - \rho$ space. The plotted lines are those defined in equation 1.5 and reparameterised using equation 1.8.	10
1.7	Simulated neutrino interaction with 3 final states in the side-right ECal. The green line entering from the left is the ν_μ . The green line exiting to the right is a μ^- , the brown line is a π^+ and the blue line is a proton. The purple rectangles represent the hit ECal bars.	11
1.8	The grid representation of an ECal bar and its representation in parameter space.	12
1.9	The cross representation of an ECal bar and its representation in parameter space.	13
1.10	The full parameter space of the 2D cluster shown in Fig. 1.7. The height of each point in the parameter space corresponds to the number of hit scintillator bars intersected by the Cartesian line which the parameter space point represents.	14

1.11	The reduced parameter space of the 2D cluster shown in Fig. 1.7. The reduced parameter space was formed by removing all of the hit scintillator bar representations which contributed to the full parameter space's (see Fig. 1.10) maximum.	15
1.12	Q_{ratio} and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.	18
1.13	$\Delta_{\text{layer, first}}$ and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.	19
1.14	$\Delta_{\text{layer, last}}$ and its probability density distribution in the barrel ECal for the two track case. NEUT-based simulation of ND280 beam events were used to produce the distributions.	20
1.15	Event display (XY view) of a problematic (for the reconstruction) neutrino interaction in the ECal. The solid green track is the muon, the solid blue tracks are protons and the pink tracks are neutrons. The neutrino is the short dashed green line. The proton travelling vertically downwards and the muon are created almost back to back leaving a line of hit scintillator bars. The 2D hough transform would register this as a single straight line.	22
1.16	The angular resolution as a function of trajectory length in the DS ECal when applying the enhanced reconstruction to muon MC. The colour coding refers to the true entry angle range of the muons. The figure was generated using single muon particle gun fired into the front face of the DS ECal with a controlled range of entry angles. The angular resolution distributions were created by L. Pickering.	25
1.17	The summed charge contained on all tracks reconstructed in each ECal cluster. The red histograms and black points are Monte Carlo and data respectively.	27
1.18	The summed number of hits contained on all tracks reconstructed in each ECal cluster. The red histograms and black points are Monte Carlo and data respectively.	28

1.19	The number of DS ECal events in the bskmu sample after relaxing the 2D track quality check. The red histograms and black points are the Monte Carlo and data respectively. The track quality fix has corrected the data excesses in both distributions, leaving only the offset between the data and Monte Carlo peaks.	29
2.1	The number of signal events as a function of neutrino energy in the bottom-left (red), top-right (blue) and DS (black) ECals.	34
2.2	Angular separation of the reconstructed tracks in an ECal cluster with the beam-simulated particle that created them. The black, red and green histograms are the tracks that were reconstructed first, second and third respectively. All histograms are area normalised.	35
2.3	Values of ϕ^{vertices} in (d_q, d_c) space. The colour corresponds to the magnitude of ϕ^{vertices}	37
2.4	ϕ^{vertices} vs the marginalised vertex reconstruction parameters.	38
2.5	Example of the vertex position, \vec{P} for three lines: \vec{l}_1 , \vec{l}_2 and \vec{l}_3 . $ \vec{r}_1 $, $ \vec{r}_2 $ and $ \vec{r}_3 $ are the perpendicular distances of \vec{l}_1 , \vec{l}_2 and \vec{l}_3 to \vec{P} respectively.	39
2.6	Example event display showing a set of track merging candidates. The final state μ^- (in green) curves in a complicated way and is reconstructed as four tracks (in light blue). The pairwise crossings of the reconstructed tracks are the red squares. The muon is produced by a charged current interaction in the top left barrel ECal producing two π^+ (in brown) and the curving μ^-	41
2.7	The cosine of the angle subtended by the merging candidates, bounded between 0 and 1. The correct matches and incorrect matches are the blue and red histograms respectively.	42
2.8	Example event display showing how the parameters of the distance ratio are calculated. The distance ratio is defined as $d^{\text{small}}/d^{\text{large}}$. The blue lines are the two reconstructed tracks which form the merging candidate and the red square is the crossing location of those tracks.	43
2.9	The distance ratio of the merging candidates. The correct and incorrect matches are the blue and red histograms respectively.	44

2.10	The distance ratio vs the cosine of the opening angle for track merging candidates.	44
2.11	Values of the tuning metric, as described in equation 2.10, in distance ratio cut vs $\cos \theta$ cut space.	45
2.12	ϕ^{merge} vs the track merging conditions.	46
2.13	Schematic showing the degeneracy of two merging candidate topologies which would pass the $\cos \theta$ cut. The arrows represent reconstructed tracks and θ is the opening angle measured between the tracks.	46
2.14	Example event display showing how the inputs to the swing parameter are calculated. Swing is defined as $l^{\text{long}}/d^{\text{large}}$. The blue lines are the two reconstructed tracks which form the merging candidate and the red square is the crossing location of those tracks.	46
2.15	The number of merging candidates as a function of the swing parameter. The correct matches and incorrect matches are the blue and red histograms respectively.	47
2.16	The number of reconstructed events in the Monte Carlo sample, separated into the prong topologies. Each event is categorised by the associated truth information from the simulation.	49
2.17	$\phi_i^{\text{selection}}$ as a function of the distance from the $-y$ face of the bottom-left barrel ECal for the 1 prong topology.	50
2.18	Total prong charge vs the number of prong hits for barrel ECal events in the 1 prong topology.	53
2.19	Total prong charge vs the number of prong hits for barrel ECal events in the 2 prong topology.	54
2.20	Total prong charge vs the number of prong hits for barrel ECal events in the 3 prong topology.	55
2.21	ΔN vs the number of prong hits for barrel ECal events in the 1 prong topology.	57
2.22	ΔN vs the number of prong hits for barrel ECal events in the 2 prong topology.	58

2.23 ΔN vs the number of prong hits for barrel ECal events in the 3 prong topology.	59
2.24 The opening angle subtended by the constituents prongs for the 2 prong topology. Signal and background events are the blue and red histograms respectively.	60
2.25 $\phi_2^{\text{selection}}$ as a function of the opening angle cut for the 2 prong topology in the barrel ECal.	61
2.26 The summed opening angle subtended by every pairwise combination of the constituents prongs for the 3 prong topology. Signal and background events are the blue and red histograms respectively.	62
2.27 Event survival as a function of the cuts for all topologies in the barrel ECal. Each bin refers to the number of events after a cut has been applied.	64
2.28 Event survival as a function of the cuts for all topologies in the DS ECal. Each bin refers to the number of events after a cut has been applied.	65
2.29 ECal OOFV event survival as a function of the cuts for all topologies in the barrel ECal. Each bin refers to the number of events after a cut has been applied.	66
2.30 ECal OOFV event survival as a function of the cuts for all topologies in the DS ECal. Each bin refers to the number of events after a cut has been applied.	67
2.31 The selection efficiency (black) and purity (red) as a function of the cuts in the barrel ECal. Each bin refers to the efficiency and purity after a cut has been applied.	69
2.32 The selection efficiency (black) and purity (red) as a function of the cuts in the DS ECal. Each bin refers to the efficiency and purity after a cut has been applied.	70
2.33 $\phi^{\text{selection}}$ as a function of the cuts in the barrel ECal. Each bin refers to the value of $\phi^{\text{selection}}$ after a cut has been applied.	71
2.34 $\phi^{\text{selection}}$ as a function of the cuts in the DS ECal. Each bin refers to the value of $\phi^{\text{selection}}$ after a cut has been applied.	72

2.35	The selection efficiency (black) and purity (red) for the combined prong topologies as a function of neutrino energy.	73
2.36	The number of selected events in the Monte Carlo sample, separated out into the prong topologies. Each event is categorised by the associated truth information from the simulation.	74

List of Tables

1.1	The percentage number of correct matches in the 3D matching separated by which track matching likelihood was used. NEUT-based Monte Carlo simulation of the T2K beam was used to calculate the matching percentages.	26
2.1	The number of signal events in the Monte Carlo sample separated by ECal module.	34
2.2	Parameters for the vertex reconstruction in the ECal.	38
2.3	Cut values for the track merging in the ECal.	47
2.4	The fiducial volume definitions for each ECal module for the 1 prong topology.	51
2.5	The fiducial volume definitions for each ECal module for the 2,3 and 4+ prong topologies.	52
2.6	The definitions of the 2 dimensional cut lines involving the total prong charge and the number of associated scintillator hits. N refers to the total number of scintillator hits associated to the constituent prongs in the vertex.	54
2.7	The definitions of the 2 dimensional cut lines involving ΔN and the number of scintillator hits associated with the constituent prongs. N refers to the total number of scintillator hits associated to the constituent prongs in the vertex.	56
2.8	The definitions of the opening angle cuts.	61
2.9	The selection efficiencies for each prong topology and ECal module. .	74

2.10 The selection purities for each prong topology and ECal module. . . .	74
2.11 The topology combined efficiency and purity.	75