The Pandora multi-algorithm approach to automated pattern recognition in LAr TPC detectors

The MicroBooNE Collaboration

Public Note:
MICROBOONE-NOTE-1015-PUB
July 1, 2016

Abstract Pattern recognition is the identification of structures and regularities in data. In high energy physics, it is a vital stage in the reconstruction of events recorded by fine-granularity detectors. The development and operation of Liquid Argon Time Projection Chambers (LAr TPCs) for neutrino physics has created a need for new approaches to pattern recognition, in order to fully exploit the superb imaging capabilities offered by this technology. Whereas the human brain excels at identifying features in the recorded events, it is a significant challenge to develop an automated solution.

The Pandora Software Development Kit (SDK) provides functionality to aid the process of designing, implementing and running pattern recognition algorithms. In particular, it promotes the use of a multi-algorithm approach to pattern recognition: individual algorithms each aim to address a specific task in a particular topology; a series of many tens of algorithms then carefully build-up a picture of the event and, together, provide a robust automated pattern recognition solution.

Building on successful use of the Pandora SDK for pattern recognition at collider experiments, a sophisticated chain of algorithms has been created to perform pattern recognition for neutrino experiments utilising LAr TPCs like MicroBooNE. The input to the Pandora pattern recognition is a list of 2D Hits. The output from the chain of over 70 algorithms is a hierarchy of reconstructed 3D Particles, each with an identified particle type, vertex and direction.

In this document, we present details of the Pandora pattern recognition algorithms used to reconstruct cosmic-ray and neutrino events in LAr TPCs. We also present metrics that assess the current reconstruction performance using simulated data from MicroBooNE.

1 Introduction

The development and operation of Liquid Argon Time Projection Chambers (LAr TPCs) for neutrino physics has created a need for new approaches to neutrino event reconstruction, in order to fully exploit the superb imaging capabilities offered by this technology. Whereas the human brain excels at identifying features in the recorded images, it is a significant challenge to develop computer-based solutions to this pattern recognition problem. The development of automated pattern recognition algorithms for LAr TPC neutrino detectors is currently an active area of research, and there has been considerable progress in recent years [1] [2] [3] [4].

The MicroBooNE experiment is playing a leading role in the development of automated algorithms for LAr TPC event reconstruction. MicroBooNE has been operating in the Booster neutrino beam at Fermilab since October 2015, and has already demonstrated automated identification of neutrino interactions using its early data [5]. Some particular reconstruction challenges for MicroBooNE include the high level of cosmic-ray background inherent in a surface-based neutrino experiment, the various sources of partially correlated noise observed in the data, and a number of unresponsive readout channels in the TPC. During its first year of operation, the first strategies for selection and reconstruction of neutrino interactions have been developed. The MicroBooNE results in summer 2016 include first automated selections and kinematic distributions of inclusive ν_μ charged-current (CC) interactions [6] and first selected events in the CC π^0 channel [7]. One of the main reconstruction tools used by MicroBooNE is the Pandora Software Development Kit (SDK) [8].

The purpose of this note is to present details of the Pandora reconstruction, and to demonstrate the current performance using simulated cosmic-ray and neutrino data from MicroBooNE.

The Pandora SDK was created to address the problem of identifying energy deposits from individual particles in fine-granularity detectors. It promotes the idea of a multi-algorithm approach to solving pattern recognition problems. In this approach, the input building-blocks (Hits) describing the pattern recognition problem are considered by large numbers of decoupled algorithms. Each algorithm targets a specific event topology and controls operations such as collecting Hits together in Clusters, merging or splitting Clusters, or collecting Clusters in order to build Particles. It is important for each algorithm to only perform pattern recognition operations when it is deemed "safe", deferring complex or unexpected topologies to later algorithms, designed to target such difficulties. In this way, the algorithms can remain decoupled and there is little inter-algorithm tension. Some algorithms may prove to be rather complex or sophisticated, whilst others can remain simple. The key point is that the algorithms gradually build-up a picture of the underlying events and collectively provide a robust reconstruction.

The Pandora SDK provides a reliable, unit-tested environment for developing and running pattern recognition algorithms. It provides carefully designed Application Programming Interfaces (APIs) to create a development environment, in which:

- It is easy for a Pandora application, which can run in any host software framework, to provide the input Hits (and MCParticles, for validation) to define the pattern recognition problem.
- The pattern recognition logic is cleanly implemented in algorithms. The algorithms make changes to objects in the Event Data Model (EDM) by requesting services from Pandora.
- Pandora performs all operations to modify objects in the EDM and provides all the book-keeping and memory-management. These services are encapsulated, with only high-level services exposed to the algorithms.
- Advanced services are available to enable complex algorithms that use recursion or reclustering approaches.
- The PandoraMonitoring package enables a visual approach to algorithm development and debugging.

The Pandora project was initiated in 2007 to provide the first particle flow calorimetry implementation for the proposed International Linear Collider (ILC). The subsequent development of the Pandora SDK was a substantial software-engineering effort and took place in 2009-2010, when it was decided to develop a fully-featured software framework for pattern recognition algorithms and to reimplement the ILC particle flow approach in this framework. With the reusable Pandora SDK framework in place, it became possible to implement the particle flow approach for other particle physics applications such as neutrino event reconstruction, and a parallel Liquid Argon reconstruction effort began in 2012. A complete set of automated and reusable pattern recognition algorithms has now been developed for use in Liquid Argon neutrino experiments. These algorithms have been fully integrated with the LARSOFT [9] and UBOONECODE [10] software packages used to simulate, reconstruct and analyse data in the MicroBooNE experiment.

This technical note concentrates on the details of the Pandora neutrino event reconstruction developed for MicroBooNE. Section 2 will discuss the input to and output from Pandora, whilst Section 3 will provide an overview of algorithms to reconstruct cosmic-ray and neutrino events at MicroBooNE. Section 4 will discuss metrics for assessing the pattern recognition performance and Section 5 will present performance figures obtained using the MicroBooNE simulation.

2 Inputs and Outputs

The Pandora reconstruction for MicroBooNE is integrated into the LArSoft framework via the LARPANDORA "translation" module or application, whose primary role is to translate the input pattern recognition building-blocks from the LArSoft EDM to the Pandora EDM, initiate and apply the Pandora reconstruction chain, then to translate the output Pandora pattern recognition results back to LArSoft. Pandora translation modules are responsible for controlling the Pandora reconstruction and are described in detail in [8]. Some details specific to the MicroBooNE translation module are discussed below.

In the initialisation step, the translation module must use the Pandora APIs to:

1. Create a Pandora instance. The MicroBooNE reconstruction requires only a single Pandora instance, whilst more complex detectors (such as those considered by DUNE) require one Pandora instance per logical "drift volume".
2. Register named Factories for all the Algorithms, Algorithm Tools and Plugins to be used in the reconstruction.
3. Provide simple detector geometry information, such as readout wire pitches and wire angles to the vertical. This is used to configure the Plugin that provides coordinate transformations between the three readout planes.
4. Pass a PANDORASETTINGS XML configuration file to the Pandora instance. This XML file contains a full description of the algorithms and their configurations to be applied each event by the Pandora instance in its multi-algorithm reconstruction approach.

On a per-event basis, the translation module must use the Pandora APIs to:

a. Translate input Hits in the LArSoft EDM to 2D Pandora Hits, which provide a self-describing input to the Pandora pattern recognition algorithms.
b. For simulated data, MCParticle instances can optionally be provided to aid algorithm tuning and enable evaluation of in-algorithm performance metrics. Each MCParticle can have a parent/daughter relationship hierarchy and links to the Pandora Hits.
c. Ask the Pandora instance to process the event. The thread will then be passed to the Pandora instance and the algorithms applied to the input Hits, as specified in the PandoraSettings XML configuration file.
d. Ask the Pandora instance to provide its output list of reconstructed Particles. Each Particle has a vertex position, associated Clusters, constituent 2D Hits, and reconstructed 3D positions. A hierarchy of parent/daughter relationships between reconstructed particles is also outputted.
e. Ask the Pandora instance to reset, so that it is ready to begin receiving new input objects for processing the next event.

The input Hits each represent a signal detected on a single wire at a definite drift time. The Pandora Hits are conventionally placed in the $x-z$ plane, with x representing the drift time coordinate, converted to a position, and z representing the wire number, converted to a position. The MicroBooNE detector has three planes of wires oriented at $\pm 60^{\circ}$ and 0° to the vertical, labelled U, V and W. Note that the vertically-oriented W plane in the MicroBooNE detector is also commonly referred to as the Y plane. The readout plane is specified for each Pandora Hit, so three 2D images are provided of events within the detector active volume. The x coordinate, obtained from the drift time, is common to all three images. This common coordinate can be exploited by the pattern recognition algorithms for 3D reconstruction.

The output Pandora objects are translated to the LArSoft EDM and provide information as illustrated in Figure 1. The main output in the LArSoft EDM is a list of reconstructed 3D Particles (termed "PFParticles") for the event. Each PFParticle corresponds to a distinct track or shower in the event, and has an associated collection of 2D hits. Each particle also has a set of reconstructed 3D positions (termed "SpacePoints") and pairs of 3D positions and directions (termed "Seeds") that define its trajectory, as well as a reconstructed vertex position that defines its interaction point or first energy deposit. The type of particle is currently not reconstructed, but PParticles are instead characterised as track-like or shower-like according to the pattern of their hits. The topological associations between Particles are also used to identify parent/daughter relationships. These relationships are then collated into a "particle hierarchy". As part of this hierarchy, a "neutrino" Particle can be created, forming the primary parent particle in the event.

3 Algorithm Overview

Two Pandora multi-algorithm reconstruction options have been provided for use at MicroBooNE. One option, PANDORACOSMIC, is optimised for the reconstruction of cosmic rays and their daughter delta rays. The second option, PANDORANU, is optimised for the reconstruction of neutrino interactions.

The PandoraCosmic and PandoraNu reconstruction chains are applied to the MicroBooNE data in a two-pass approach. The PandoraCosmic reconstruction is first used to process all Hits identified during a provided time-window and provide a list of candidate cosmic-ray Particles. This list of Particles is then examined by a

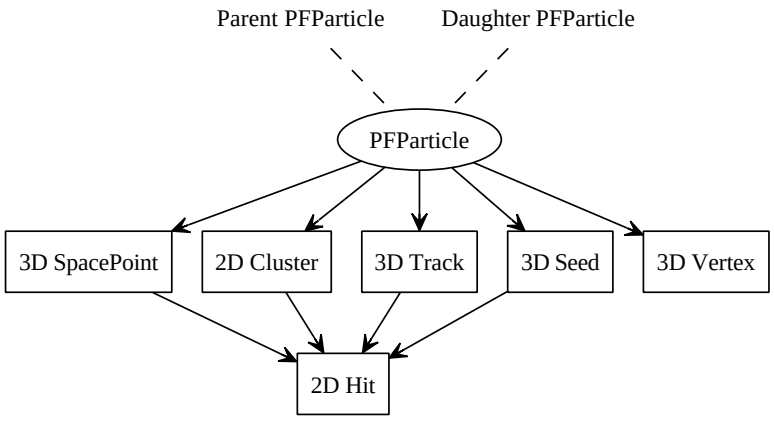


Fig. 1: Illustration of Pandora output in the LArSoft EDM [9]. Navigation along PFParticle hierarchies is achieved using the PFParticle interface, represented by dashed lines. Navigation from PFParticles to their associated objects is represented by solid arrows.

cosmic-ray tagging module, implemented within LArSoft, which identifies unambiguous cosmic-ray muons based on their start and end positions and associated Hits. Particles are flagged as unambiguous cosmic-rays if: (1) some of the associated Hits are placed outside the detector when the event time is taken to be the neutrino beam trigger time; (2) the reconstructed trajectories are through-going, with the exception of Particles that pass through both the upstream and downstream faces of the detector. Hits associated with Particles flagged as cosmic rays are removed from the input Hit collection and a new cosmic-removed Hit collection created. This second Hit collection provides the input for the PandoraNu neutrino event reconstruction, which outputs a list of candidate neutrino interactions.

The LArSoft translation module used for PandoraCosmic and PandoraNu is completely unchanged, but the input Hit collections differ, as described above. Crucially, the choice of Pandora algorithms also differs, by using different PandoraSettings XML files to specify alternative algorithm configurations. Many algorithms are shared between the PandoraCosmic and PandoraNu reconstructions, but the overall algorithm selection results in the following key features:

– **PandoraCosmic:** This reconstruction is more strongly track-oriented, providing top-level particles representing cosmic-ray muons. Showers are assumed to be delta rays and are added as daughter particles to the most appropriate cosmic-ray muon. The reconstructed vertex/start-point for the cosmic-ray muon is the track high-y coordinate in the detector.
– **PandoraNu:** This reconstruction is more careful to identify a neutrino interaction vertex and to protect all particles emerging from the vertex (i.e. ensure that each is reconstructed as an individual, primary Particle). There is careful treatment to reconstruct tracks and showers. Parent neutrino Particles are created and reconstructed visible Particles are added as daughters of the neutrino.

The first step of the PandoraNu reconstruction is to use Pandora's ability to provide recursive algorithm functionality: it runs a "fast" version of the core track reconstruction up to the point of 3D Hit creation. The 3D Hits are then divided into separate 3D volumes, or "slices", that should each contain single, distinct neutrino interactions or cosmic-ray remnants. Each slice (including those containing cosmic-ray remnants) is processed by the main PandoraNu reconstruction and will yield exactly one output candidate neutrino Particle, with a hierarchy of daughter Particles.

The remainder of this Section will provide a high-level description of some of the key algorithm features in the PandoraCosmic and PandoraNu reconstruction.

3.1 Cosmic-Ray Reconstruction

3.1.1 Two-Dimensional Reconstruction

The very first step in the Pandora LAr TPC reconstruction is to separate the input Hits into three separate Hit lists; corresponding to each of the wire readout planes (U, V and W). This operation is performed by

the EVENTPREPARATION algorithm. In the early stages of the reconstruction, only 2D information will be considered (i.e. Hits in each readout plane will be considered separately).

The Pandora multi-algorithm reconstruction then begins by running a TRACKCLUSTERCREATION algorithm, which aims to provide a list of "ProtoClusters". These Clusters represent continuous, unambiguous lines of Hits. Separate Clusters should be created for each structure in the input Hit image, with Clusters starting/stopping at each "branch" feature or any time there is any bifurcation or ambiguity. Figure 2 shows a typical example of the output of the 2D clustering.

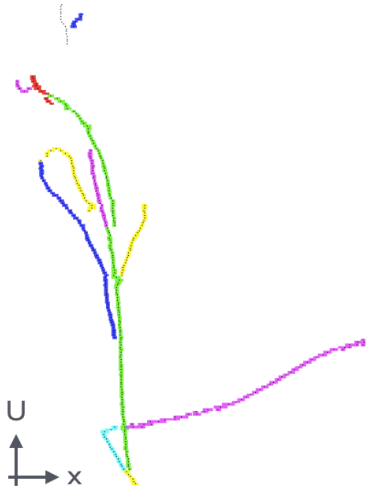


Fig. 2: Typical output of the initial track-oriented clustering algorithm, with a random colour scheme used to provide an indication of separate Clusters.

The initial clustering is extremely careful to provide Clusters of high purity (i.e. represent energy deposits from exactly one true particle), even if this means that the Clusters are initially of low completeness (i.e. only contain a small fraction of the total Hits associated with a single true particle). This plays to the strengths of the multi-algorithm approach. The Clusters are then examined by a series of algorithms collectively named "Topological Association" algorithms. These algorithms examine the associations between the 2D Clusters and look to grow the Clusters to improve completeness, without compromising purity.

Each of the algorithms compare Clusters in the relevant Cluster lists and request Cluster merging operations or, if necessary, Cluster splitting operations to refine the Hit selection for each Cluster. This series of algorithms is very important and early development quickly revealed that sophisticated approaches were required. A common limitation in pattern recognition algorithms is the tendency to consider only pairs of Cluster objects at a given time, and so make decisions based on local, rather than full environmental information. In the Pandora topological association algorithms, a decision was made to author "design pattern" algorithms, whereby different definitions of Cluster association could be provided, but a common algorithm implementation would ensure that Cluster association was evaluated for all possible Cluster combinations. Possible chains of Cluster associations could then be discovered and the final Cluster merging decisions could based upon an understanding of the overall event topology.

The CLUSTERASSOCIATION and CLUSTEREXTENSION algorithms represent two such design pattern algorithms. Algorithms inheriting from these base classes need only to provide an initial selection of potentially interesting Clusters (based on length, number of Hits, etc.) and a metric for determining whether a given pair of Clusters should be declared associated. Algorithms providing concrete implementations of the selection and association are used to carefully extend 2D Clusters in both the longitudinal (along the beam direction) and transverse directions. They are also used to merge Cluster candidates across registered detector gaps, representing those wires in the MicroBooNE detector that have been assigned a bad channel status.

Following attempts at Cluster extension, the final topological association algorithms revisit the Clusters to identify whether any Clusters should be split-up at specific points of interest, such as where there are significant kinks in the Cluster, or where multiple Clusters intersect. The final output of the 2D reconstruction

then provides the input to the process used to "match" features reconstructed in multiple readout planes and construct Particles.

Much of the 2D Cluster reconstruction is shared between PandoraCosmic and PandoraNu, but PandoraCosmic provides a number of additional topological association algorithms (re-using some of the existing base class functionality), tuned to split or extend cosmic-ray or delta-ray Clusters appropriately.

3.1.2 Three-Dimensional Track Reconstruction

The main aim of the 3D track reconstruction is to collect-together three consistent, track-like Clusters (one from each readout plane). The Clusters can be formally bound-together by adding them as daughter objects of a new Pandora Particle. The 3D track reconstruction algorithms must therefore identify consistent groupings of Clusters from the different readout planes. If there are inconsistencies, the algorithms must identify common features in the three readout planes in order to iteratively correct the 2D clustering and allow unambiguous Particles to emerge.

The majority of the 3D track matching is performed by the THREEDTRANSVERSETRACKS algorithm. This algorithm considers the suitability of all possible combinations of Clusters from the three readout planes and stores the results in a rank-three tensor. The three tensor indices are the Clusters in the U, V and W views and, for each combination of Clusters, a detailed TRANSVERSEOVERLAPRESULT is stored. The tensor is essentially just a means of storing the information associated with each combination of three Clusters, and its interface provides simple access to the data it contains. The tensor is interrogated by a series of algorithm tools, which identify Cluster matching ambiguities and make careful changes to the 2D Clusters until the tensor is diagonal and the Cluster combinations are unambiguous.

Figure 3 illustrates how a TransverseOverlapResult can be constructed for a set of three Clusters (one from each of the U, V and W readout planes). In the x-region (drift time) common to all three Clusters, a number of x sampling points is defined. The algorithm then looks to see if the Cluster is matched at each sampling point. To determine whether a sampling point is matched, and to assess the goodness of matching, the following procedure is used:

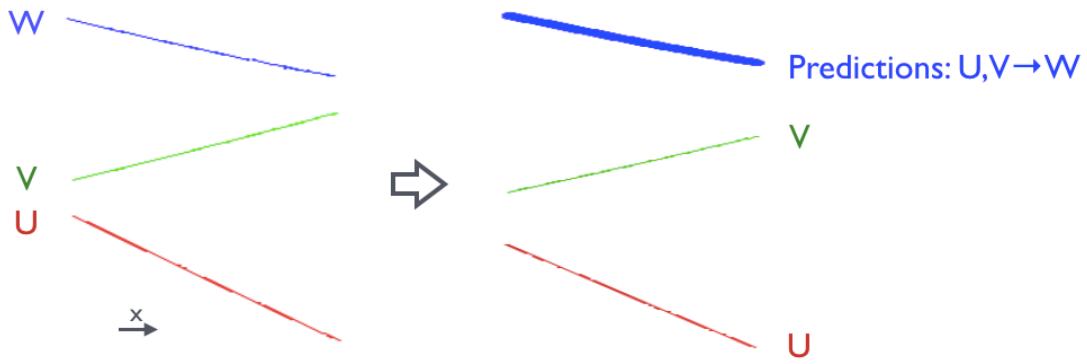


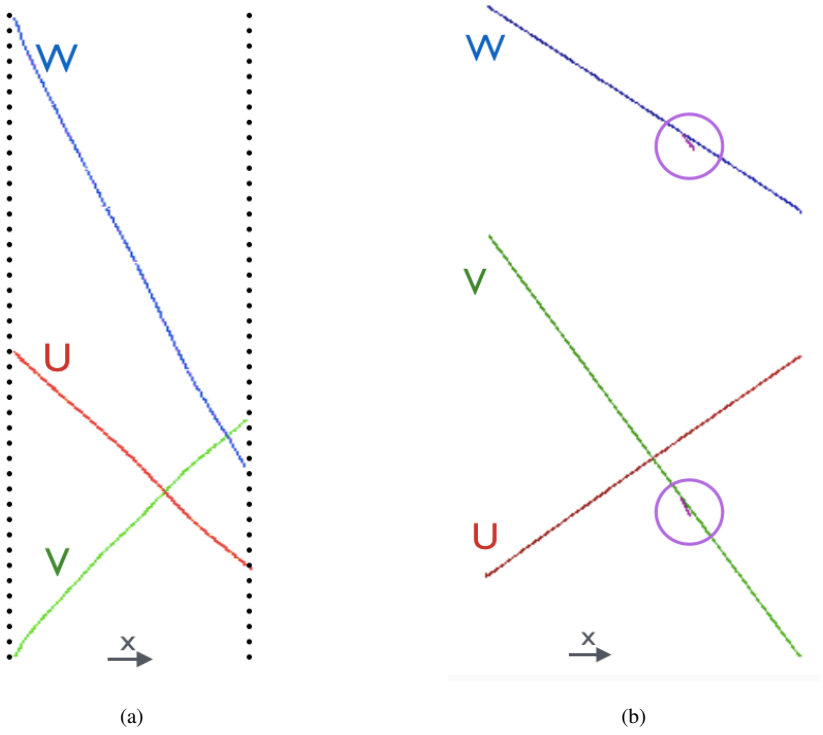
Fig. 3: Illustration of the approach used to assess the compatibility of Clusters from the U, V and W views in the ThreeDTransverseTracks algorithm.

1. 2D "sliding linear fits" to each Cluster are used. These record the results of a linear fit, sliding over a local region of the Cluster. They also provide extensive extrapolation/interpolation and position look-up functionality.
2. For a given x coordinate, look-up the sliding fit position for a pair of Clusters (e.g. in the U and V views).
3. Use these positions, and the coordinate transformation Plugin, to predict the position of the third Cluster (e.g. in the W view) at the given x coordinate.
4. Compare the true sliding fit value for the third Cluster with the prediction, identifying whether or not the sampling point is successfully matched to the Cluster.
5. Use all three possible predictions (U,V→W; V,W→U; U,W→V) to calculate a χ^2 value.

The TransverseOverlapResult records the x-overlap span, the number of sampling points, the number of matched sampling points and the χ^2 value. Crucially, the tensor also provides a means by which the algorithm can understand the connections between multiple Clusters. It is these results and connections that are queried by a series of algorithm tools in order to create Particles or to modify the 2D pattern recognition in a bid to enable Particle creation.

The algorithm tools have a specific ordering and, if any tool makes a change (either by creating a new Particle or modifying the 2D reconstruction), the full list of tools runs again, repeating from the first tool. The tensor is processed until no tool can perform any further operations. Care is taken to ensure that it is not possible to become stuck between conflicting operations performed by different tools.

The first tool looks to directly build Particles from unambiguous groupings of three Clusters. It examines the tensor to find regions where only three Clusters are connected, one from each of the U, V and W views. Quality cuts are applied to the TransverseOverlapResult and, if passed (the common x-overlap must be >90% of the x-span for all Clusters at this stage), a new Particle is created. This ClearTracks tool is illustrated in Figure 4a, which displays an unambiguous grouping of three Clusters from the U, V and W views with excellent overlap in the common x coordinate.



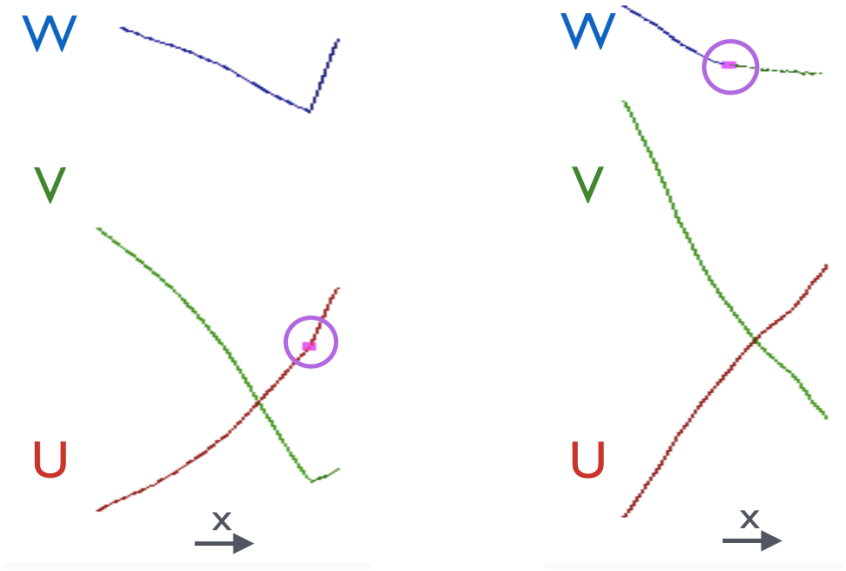
(a) (b)

Fig. 4: (a) Example topology addressed by the ClearTracks algorithm tool, which will combine the separate Clusters from the U, V and W views into a single Particle. (b) Example topology addressed by the LongTracks algorithm tool, which looks to address minor Cluster matching ambiguities.

The second tool is the LongTracks tool, which aims to resolve any obvious ambiguities. Figure 4b shows an example where the presence of two small delta-ray Clusters on long cosmic-ray tracks mean that Clusters are matched in multiple configurations and the tensor is not diagonal. One of the TransverseOverlapResults is, however, significantly better than the others (with better x-overlap and many more matched sampling points) and the correct groupings of Clusters can be selected. The delta rays can then be associated with the cosmic-ray Particle at a later stage in the reconstruction.

The subsequent tools, the OvershootTracks and UndershootTracks tools, emphasise the strength of the tensor-based approach. The OvershootTracks tool examines the tensor to find Cluster matching ambiguities of the form 1:2:2 i.e. a single Cluster in one view, e.g. U, is matched to two Clusters in each of the other two views, e.g. V and W, leading to two conflicting TransverseOverlapResults in the tensor. An example of this kind of topology is shown in Figure 5a. In this example, the two Clusters in the V view and the two

Clusters in the W view connect at a common x coordinate. There is a single, common Cluster in the U view, which spans the full x-extent of the Clusters. The OvershootTracks tool can use the collection of Clusters to assess whether they represent a kink topology in 3D. If a 3D kink is identified, the U cluster can be split at the relevant position and two new U Clusters fed-back into the tensor. By construction, the initial ClearTracks tool will then be able to identify two unambiguous groupings of three Clusters and create two Particles.



(a) (b)

Fig. 5: (a) Example topology addressed by the OvershootTracks algorithm tool, which will here look to split the single U Cluster at the highlighted position and remove the Cluster matching ambiguity. (b) Example topology addressed by the UndershootTracks algorithm tool, which will here look to merge the two W Clusters to remove the ambiguity.

The UndershootTracks tool examines the tensor to find Cluster matching ambiguities of the form 1:1:2, as illustrated in Figure 5b. In this example, two Clusters in the W view are matched to common Clusters in the U and V views, leading to two conflicting TransverseOverlapResults in the tensor. The UndershootTracks tool examines the Cluster pairings to assess whether the Clusters represent a kink topology in 3D (implementation shared with the OvershootTracks tool). If a 3D kink is not found, the two W Clusters can be merged and a single W Cluster fed-back into the tensor. The ambiguity is thereby removed and a single new Particle can be created by the ClearTracks tool.

Subsequent algorithm tools continue to query the tensor and attempt to adjust the 2D pattern recognition in order to allow unambiguous Particle creation. One tool, the MissingTracks tool, understands that particle features may be obscured in one view, with a single Cluster representing multiple overlapping particles. If this tool identifies appropriate Cluster overlap, using the Cluster relationship information available from the tensor, the tool can create two-Cluster Particles. An additional tool, the TrackSplitting tool, looks to remove sections of Clusters where the matching between views is unambiguous, but there is a significant discrepancy between the Cluster x-spans and evidence of gaps in a Cluster. A similar tool, the MissingTrackSegment tool, tries to perform the reverse operation by adding missing Hits to the end of a Cluster in order to address the x-span discrepancy.

In addition to the ThreeDTransverseTracks algorithm, there are also ThreeDLongitudinalTracks and ThreeDTrackFragments algorithms. These algorithms also form a Cluster association tensor and query the tensor using algorithm tools. The difference is the OverlapResult that is stored in the tensor. The ThreeDLongitudinalTracks algorithm examines the case where the x-extent of a Cluster grouping is small. In this case, there are too many ambiguities when trying to sample the Clusters at fixed x-positions. Instead, the algorithm postulates that the Cluster start and end positions match in the U, V and W views, allowing creation of 3D end-points and a 3D trajectory to be defined in order to assess the compatibility of the three Clusters.

The ThreeDTrackFragments algorithm is optimised to look for situations where there are single clean Clusters in two views, associated with multiple fragment Clusters in a third view. Its algorithm tools can request that the fragment Clusters be merged-together in order to resolve the ambiguity and enable Particle creation.

The 3D track reconstruction described above is common to both the PandoraCosmic and PandoraNu reconstructions. PandoraCosmic also provides an additional two algorithms, COSMICRAYTRACKMATCHING and COSMICRAYTRACKRECOVERY, which look to aggressively match any unassociated track-like Clusters remaining at this stage of the reconstruction.

3.1.3 Delta-Ray Reconstruction

Following the 3D track reconstruction, the PandoraCosmic reconstruction dissolves any 2D Clusters that have not been included in a reconstructed Particle. The Hits are then reclustered using a simple, proximity-based clustering algorithm to try to begin the process of identifying delta-ray showers. Most delta-ray Clusters will be created in this way, although it should be remembered that any long, track-like delta-ray Clusters (with common features in multiple views) may already exist in track Particles and must also be addressed by the algorithms.

A number of topological association algorithms, which re-use implementation from the earlier 2D reconstruction, begin the process of carefully merging Clusters to provide a more complete delta-ray reconstruction. Subsequent algorithms then perform the key operations of matching delta-ray Clusters between views and identifying the most appropriate parent cosmic-ray Particles. All possible combinations of delta-ray Clusters are assessed, initially searching for matches between all three views, then falling back to two-view matches and use of a single view. The Cluster matching is rather simple and assesses the x-overlap if there are Clusters in multiple views. If there are Clusters in all three views, the consistency of the Clusters is checked by sampling at multiple points across the common x coordinate, as described in Section 3.1.2. Parent cosmic-ray Particles are identified by simple comparison of inter-Cluster distances and Particle parent-daughter links are formed.

3.1.4 Three-Dimensional Hit Reconstruction

At this point, the PandoraCosmic pattern recognition is complete and the reconstructed Particles contain 2D Clusters from one, two or (typically) all three readout planes. For each input (2D) Hit in a Particle, an attempt is made to create a new 3D Hit (SpacePoint). The mechanics of the SpacePoint creation differ depending upon the Cluster topology. A series of algorithm tools is used to create the SpacePoints, with tools for each topology e.g. Hits on transverse tracks with Clusters in all views, Hits on longitudinal tracks with Clusters in all views, Hits on tracks that are multi-valued at specific x coordinates, Hits on tracks with Clusters in only two views, Hits in showers, etc.

For simple transverse tracks with Clusters in all three views, the approach is to take a 2D Hit in one view e.g. U and sliding linear fit positions for the e.g. V and W Clusters at the same x coordinate. A function, compiled as part of the translation module (so the implementation can be highly detector-specific), is then called that provides an analytic χ^2 minimisation to provide the optimal y and z coordinates at the given x position. It is also possible to run in a mode whereby the chosen y and z coordinates are such that they represent a projection of the two fit positions onto the specific wire associated with the 2D Hit.

For a 2D Hit in a shower Particle (a delta ray, e.g. in the U view), all combinations of Hits (e.g. in the V and W views) that are located in a narrow region around the Hit x coordinate are considered. For a given combination of Hit U, V and W values, the most appropriate y and z coordinates can be calculated. The position yielding the best χ^2 value is identified and a rather tight χ^2 cut is applied to help ensure that only satisfactory positions emerge.

The SpacePoints are used to create a new 3D Cluster, which is added to the relevant Particle. After the SpacePoint creation, the PandoraCosmic reconstruction finishes by placing cosmic-ray vertices/start-positions at the muon track high-y coordinate. Vertices are also reconstructed for the daughter delta-ray Particles and are placed at the point of closest approach between the SpacePoints associated with the parent cosmic ray and those associated with the delta ray. A 3D sliding fit is performed to all track Particles, providing position and direction information along the track trajectory. This information from this “fast fit” is later persisted in LArSoft as a reconstructed track.

3.2 Neutrino Reconstruction

3.2.1 Initial Reconstruction Steps

Many of the features of the PandoraNu reconstruction have already been described throughout Section 3.1. The reconstruction begins by running a fast/minimal version of the 2D reconstruction, 3D track reconstruction and 3D Hit reconstruction. The 3D Hits are then divided into slices, using proximity and direction-based metrics. The intent is to isolate neutrino interactions and/or cosmic-ray remnants in individual slices. The original 2D Hits associated with each slice are used as an input to the PandoraNu reconstruction described in this Section. Each slice (including those containing cosmic-ray remnants) results in one candidate reconstructed neutrino Particle, which consists of a 3D interaction vertex and a list of visible daughter Particles, each of which may have additional daughters.

The PandoraNu reconstruction begins with a track-oriented clustering algorithm and series of topological association algorithms, precisely as described in Section 3.1.1. The lists of 2D Clusters for the different read-out planes are then used in order to identify the neutrino interaction vertex. This vertex plays an important role throughout the subsequent PandoraNu algorithms, helping the algorithms to identify individual primary particles and ensure that they each result in separate reconstructed Particles.

3.2.2 Three-Dimensional Vertex Reconstruction

Reconstruction of the neutrino interaction vertex proceeds in two steps:

1. Use pairs of 2D Clusters from different views to create plausible candidate 3D vertex positions.
2. Select the best candidate by examining the distribution of 2D Hits around projections of the candidates into each view.

The CANDIDATEVERTEXCREATION algorithm compares all possible pairings of Clusters (above a certain length) in different views. For each pairing, up to four candidate positions can be created to provide an exhaustive list of feature points for later consideration. The low-x point of the first Cluster is considered and the position of the second Cluster at the same x coordinate is identified. To obtain this second position, a sliding linear fit is used and a small discrepancy between the Cluster x-extents is allowed (i.e. can extrapolate outwards a small distance away from the second Cluster, or consider a position a small distance along the length of the second Cluster). The pair of positions from two different views is sufficient to provide a 3D vertex candidate. Candidates can be formed from all permutations of the start and end positions for the two Clusters. Example candidate vertex positions, projected into the W view, are shown in Figure 6.

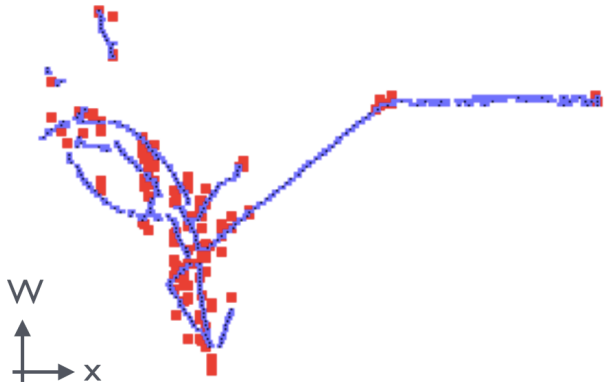


Fig. 6: Typical candidate vertex positions, projected into the W view (red points), and superimposed on the 2D Clusters in the same view.

Each candidate vertex is assessed by projecting it into each of the three views and considering the distribution of 2D Hits around the projected positions. The mechanism by which a score is obtained for each candidate

is illustrated in Figure 7. For each 2D view, the projected position of the candidate vertex defines a plane-polar coordinate system. For each 2D Hit in the specific view, the Hit r and ϕ coordinates are evaluated. It is the Hit ϕ distribution that is of particular interest, because centering the coordinate system on the true vertex position will tend to mean that the Hits are grouped into tight ϕ ranges, each grouping representing a distinct particle leaving the interaction. Centering the coordinate system on a fake vertex candidate will tend to mean that the Hit ϕ distribution is smeared significantly.

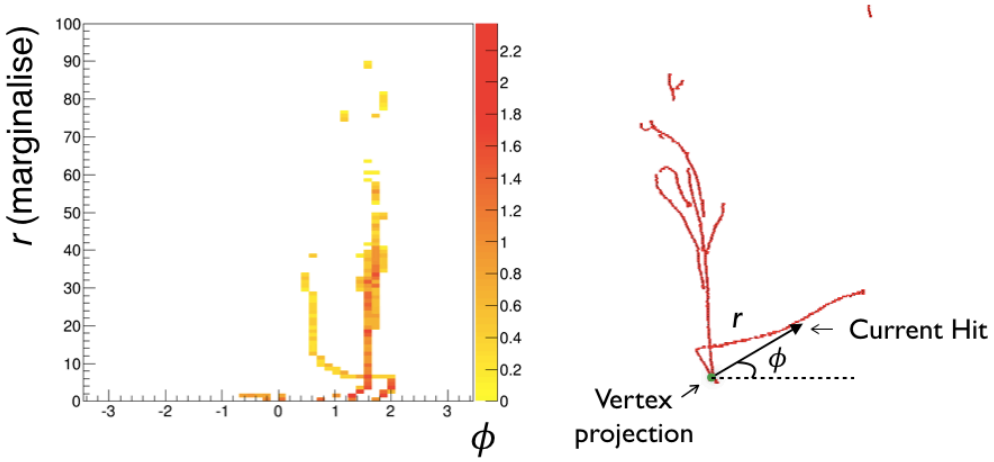


Fig. 7: 2D histogram showing distance-deweighted Hit $r-\phi$ distribution for Hits in a plane polar coordinate system centered on the projected position of a vertex candidate.

To reduce the contribution from Hits far from the vertex candidate, the Hit ϕ distribution is deweighted, with each Hit making a contribution to the distribution proportional to $r^{-1/2}$. This prevents downstream features from having undue influence. The initial algorithm approach was to construct a histogram of the deweighted Hit ϕ distributions for each of the three views. A score that promotes the idea of grouping the Hits into narrow ϕ ranges could then be obtained by simply squaring the histogram bin contents and summing the results. Subsequent improvements led to the removal of binning effects by using Kernel Density Estimation to provide a non-parametric estimation of the deweighted Hit ϕ distributions. The distributions can be sampled at the ϕ position of each Hit, and the sampling results summed, to provide a score (denoted S_{topology}) for each vertex candidate.

Additional improvements to the vertex selection included "folding" the ϕ distribution into the range $0 \rightarrow \pi$, with cancellation between contributions from π-separated Hits. This adjustment helps to disfavour vertex candidates placed astride long, straight tracks, which would otherwise appear as favourable locations. The large number of input vertex candidates are also filtered and required to sit on or near a Hit, or in a registered detector gap, in all three views in order to be put forward for assessment. Finally, with the knowledge that the major use of the algorithm would be to identify interaction positions for beam neutrino candidates, a beam weighting was introduced as a configurable, multiplicative correction to the vertex score. The range of z-positions for filtered vertex candidates is used to define a length representing two decay lengths (2λ), and a modified score (denoted S_{beam}) is calculated as shown in Equation 1:

$$S_{\text{beam}} = S_{\text{topology}} \times e^{-(z_{\text{candidate}} - z_{\text{min}})/\lambda} \tag{1}$$

where: $z_{\text{candidate}}$ is the z-position of a candidate vertex in a given view, and z_{min} is the most upstream z-position of the 2D Hits in that view.

Following identification of the neutrino interaction vertex, any 2D Clusters crossing the vertex are split in two pieces, one either side of the projected vertex position.

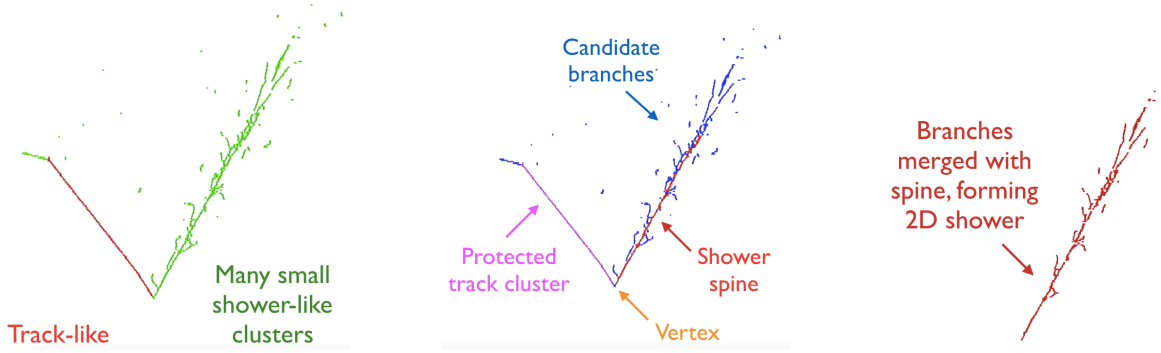
3.2.3 Track and Shower Reconstruction

The 3D track reconstruction proceeds exactly as described in Section 3.1.2. It is immediately after this stage that the PandoraNu and PandoraCosmic reconstructions diverge, as the PandoraNu reconstruction attempts to reconstruct both primary tracks (typically from muons, protons and charged pions) and primary showers (typically from electrons or photons).

The PandoraNu reconstruction attempts to perform 2D shower reconstruction, adding branches to any long Clusters that represent "shower spines". These spines may already exist in track Particles and any tracks that do start to acquire multiple branches are deleted, i.e. the Particle is deleted and the constituent 2D Clusters are released. These 2D Clusters will instead acquire the identified branches and will provide an input to the later 3D shower reconstruction algorithms.

The 2D shower reconstruction proceeds by:

1. Characterisation of 2D Clusters as track-like or shower-like, using topological measures (some use of calorimetric information would be desirable in the future). Track selection cuts are placed on the length of the Clusters, measurements of the transverse width of the Clusters and measurements of how sparse the Hit distribution is along the Clusters. An example of the typical topologies under investigation is shown in Figure 8a.
2. Identification of long, often vertex-associated, shower-like 2D Clusters that could represent shower spines. These 2D Clusters could already be constituents of reconstructed track Particles, but the successful growth of shower spines will release the track Particle and enable creation of a shower Particle, as discussed above. An example shower spine, with nearby shower branches is shown in Figure 8b.
3. Addition of 2D branch Clusters to the most appropriate 2D shower spine Clusters. This algorithm operates recursively, finding branches on a shower spine candidate, then branches on branches, etc. For every branch, a strength of association to each candidate shower spine is recorded. Informed decisions about which branches to add to which shower spines can then be made in the context of the overall event topology. An example of successful branch addition is shown in Figure 8c.



(a) (b) (c)

Fig. 8: Illustration of the key steps in the 2D shower reconstruction, which aims to add 2D branch Clusters to 2D shower spine Clusters.

Following the 2D shower reconstruction, the (now hopefully rather complete) 2D shower Clusters are matched between readout planes in order to form 3D shower Particles. The ideas and base classes developed for the 3D track reconstruction are re-used for this process. A THREEDSHOWERS algorithm builds a rank-three tensor to store Cluster overlap and relationship information, then a series of algorithm tools query the tensor, making changes to the 2D reconstruction in order to ensure that the tensor is diagonal and 3D shower Particles can be formed without ambiguity.

The procedure for calculating a SHOWEROVERLAPRESULT to store in the tensor is illustrated in Figure 9. A 2D sliding shower fit is performed to the 2D shower Clusters. This shower fit consists of three sliding linear fit results: one to all of the Hits in the 2D Cluster and one for fits to each of two "shower edges". The shower

edge fits consider only Hits with the extremal transverse coordinates (with respect to the shower axis). The two sliding shower edge fits provide a mechanism for parameterising the envelope of the shower Cluster. In order to calculate a ShowerOverlapResult for a group of three Clusters, the shower edges from two Clusters (e.g. in the V and W views) are combined in order to predict the shower envelope for the third Cluster (e.g. in the U view). The fraction of Hits in the third Cluster contained within the envelope can then be recorded. All three combinations of Clusters are used to evaluate the final ShowerOverlapResult, which also stores the Cluster x-overlap.

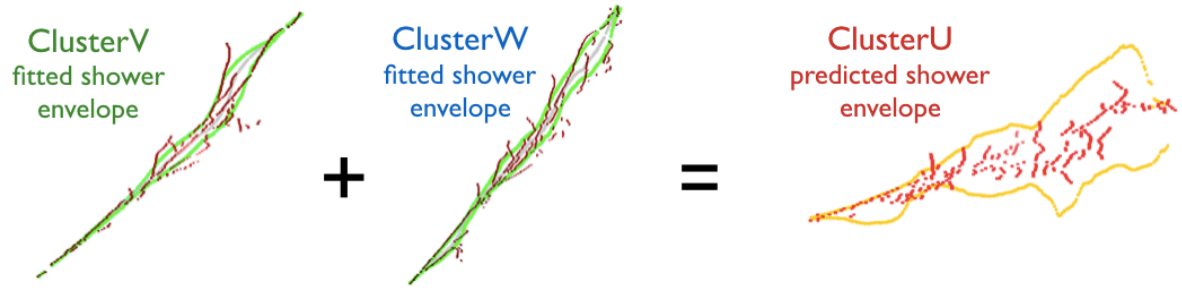


Fig. 9: Illustration of the approach used to assess the compatibility of Clusters from the U, V and W views in the ThreeDShowers algorithm.

Following the approach developed for 3D track reconstruction, the shower tensor is queried by a series of ordered algorithm tools, with the full list of tools running again if any tool makes a change to the 2D reconstruction. The first tool is the ClearShowers tool, which looks to form shower Particles from any unambiguous associations between three Clusters. The association between the Clusters must satisfy quality cuts on the common x-overlap and fraction of Hits enclosed in the predicted shower envelopes. The next tool is the SplitShowers tool, which looks to address the kind of topology illustrated in Figure 10. In this example, three tensor elements share common Clusters in the V and W views. This flags the three Clusters in the U view as being worthy of further investigation. In two passes of the SplitShowers tool (each pass merging one pair of Clusters), the associations between the U Clusters will be examined and the Clusters merged together, thereby resolving the cluster matching ambiguity.

The SplitShowers tool can operate in modes where it examines ShowerOverlapResults with either one or two Clusters in common. It always looks to see if plausible 2D Cluster merges can be made in order to help diagonalise the tensor. Further tensor tools are foreseen to aid the 3D shower reconstruction, for instance by also introducing Cluster splitting to improve the separation of nearby showers. Presently, the 3D shower reconstruction is completed by the SimpleShowers tool, which does not attempt to resolve ambiguities, but simply looks to see whether there are any ShowerOverlapResults remaining that pass required quality cuts, recovering missing shower Particles whenever the cuts are satisfied.

Following 3D shower reconstruction, a ParticleRecovery algorithm is used in a number of different configurations to try to identify any remaining particles, missed by the earlier algorithms. This algorithm aims to address groups of Clusters that (due to earlier 2D reconstruction or Hit finding failures) fail the quality cuts for Particle creation, or for which key features are missing in one or more view. Many of the ideas from the earlier 3D track and shower reconstruction are re-used, but the thresholds for matching Clusters between views are reduced. The requirement to identify features in all three views is also relaxed and the ParticleRecovery algorithm builds three Cluster association matrices, rather than a single rank-three tensor to allow for the fact that Clusters may be missing in one view. Both track and shower Particles can be created, depending on configuration of the ParticleRecovery algorithm and the results of the earlier track-like or shower-like 2D Cluster classification.

The particle identification labels (“PDG codes") attached to the final Particles identify whether the Particles were created by the 3D track reconstruction algorithms (these will have PDG code 13 for a muon) or the 3D shower reconstruction algorithms (PDG code 11 for an electron).

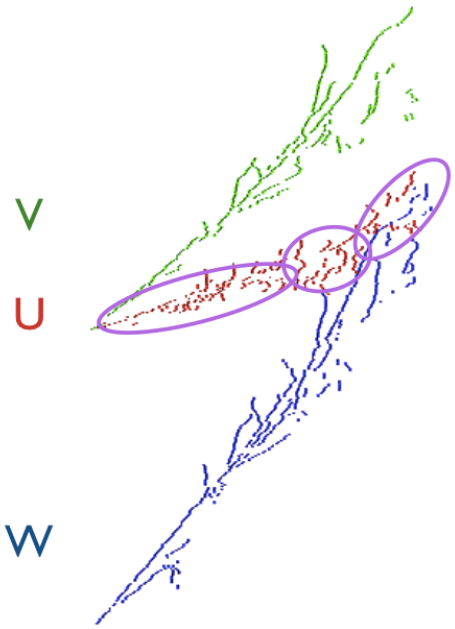


Fig. 10: Example topology addressed by the SplitShowers algorithm tool, which will here look to merge the multiple U Clusters in order to remove the Cluster matching ambiguity.

3.2.4 Particle Refinement

The Pandora reconstruction began by providing initial 2D Clusters, then using a series of topological association algorithms to merge or split the 2D Clusters to improve their purity and completeness. In the same way, the initial list of 3D track and shower Particles can be examined and refined, to provide the final output of the multi-algorithm reconstruction. One step in the refinement process uses shower envelopes and cone fits to the 2D Clusters that have been assigned to Particles. These constructs can then be used to pick-up any remaining small, unassociated 2D Clusters as appropriate. The next key algorithm operation is to merge together reconstructed Particles that look like they represent elements of the same true particle. This is particularly important for very sparse showers, where there can be sizeable distances between grouping of Hits associated with a single true particle.

Figure 11 displays the kind of topology that is addressed by the VERTEXBASEDPARTICLEMERGING algorithm. The Figure shows two views in which a single true shower remains split into multiple reconstructed Particles. Two of these Particles were reconstructed as showers and one was reconstructed as a track. The algorithm works outwards from the interaction vertex and looks to iteratively pick-up downstream Particles for which associations can be identified in multiple views. In the example shown, the shower Particle displayed in red is identified as being associated with the interaction vertex. A simple cone is fitted to all of the daughter 2D Clusters for this Particle and the fraction of Hits enclosed in downstream 2D Clusters is identified. If two Particles prove compatible in multiple views, the Particles (and constituent 2D Clusters) are merged. The algorithm continues until all appropriate downstream Particles (displayed in blue) have been picked-up. If any of the Particles merged was a shower, the final output Particle will be flagged as a shower.

3.2.5 Particle Hierarchy Reconstruction

Upon completion of the Particle reconstruction algorithms, 3D Hits are created for all Particles, as described in Section 3.1.4. The reconstructed Particles are then organised into a hierarchy, which assumes that the 2D Hits in the input slice were from a neutrino interaction.

- A neutrino Particle is created and the 3D interaction vertex is added to this Particle.
- The 3D Hits associated with the reconstructed track and shower Particles are considered and any Particles associated with the interaction vertex are added as primary daughters of the neutrino Particle.

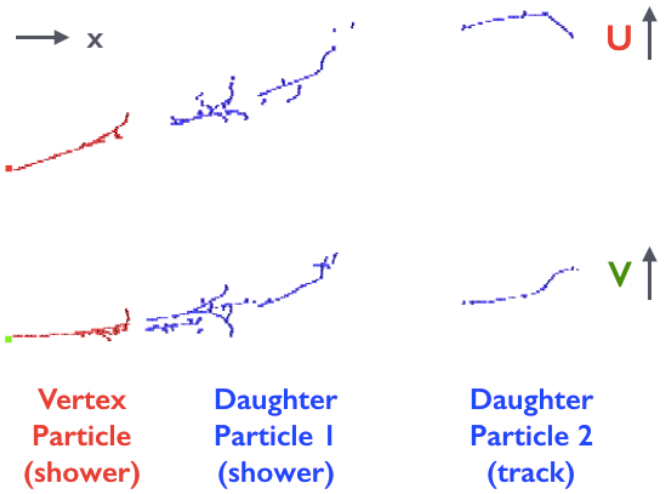


Fig. 11: Illustration of the approach used to merge together multiple Particles representing elements of the same true shower.

- A number of algorithm tools look to add subsequent daughter Particles to the existing primary daughters of the neutrino (e.g. a decay electron may be added to a primary muon Particle, which is itself a daughter of the neutrino Particle).
- The particle identification label for the neutrino Particle specifies whether the daughter Particle with the largest number of Hits is a track or a shower. The neutrino will then have PDG code 14 for a ν_μ or 12 for a ν_e respectively.
- 3D vertex positions are calculated for each of the reconstructed Particles added to the neutrino hierarchy. The vertex positions are the points of closest approach to their parent Particles, or to the neutrino interaction vertex.

Each input slice will be processed and will result in a reconstructed neutrino Particle. The reconstructed neutrinos are obtained by the translation module in LArSoft and translated to the LArSoft EDM as described in Section 2, ready for downstream processing or analysis.

4 Performance Metrics

Pattern recognition metrics are important to assess the output of the multi-algorithm reconstruction and to drive further algorithm development. There are myriad ways in which to define and interpret performance metrics for event reconstruction, so it is important that any metrics used are described in full detail. The development of metrics for LAr TPC event reconstruction is an active area of study within MicroBooNE and also in the LArSoft and Pandora projects. The preliminary performance metrics presented in this note have been developed specifically to study the performance of the Pandora pattern recognition chains in MicroBooNE, highlight current issues, and drive the future direction of algorithm development.

This Section will describe an approach to performance validation developed for multi-particle final states. The performance metrics are well-defined, but are not very "forgiving". The aim is to match at least one reconstructed particle to each and every true primary particle in an interaction. Events are deemed to have a "completely correct" reconstruction if they have exactly one reconstructed particle for every true particle. Events with minor errors, readily dismissed by eye, are often classed as failures in this validation scheme.

The performance studies presented in this note are based on simulated events in the MicroBooNE detector. The following procedure is used to match reconstructed particles to the true simulated particles:

- **Selection of MCParticles** The full hierarchy of MCParticles is first extracted from the simulated data. For neutrino interactions, the hierarchy comprises the incident neutrino, the final-state particles, and cascades of daughter particles produced by decays or interactions. A list of "target" MCParticles is selected from the data by navigating down through the MCParticle hierarchy, considering each particle in turn,

and selecting "visible" MCParticles (defined as one of $e^{\pm}, \mu^{\pm}, \gamma, K^{\pm}, \pi^{\pm}, p$). In the case of multi-particle parent/daughter cascades, then the most upstream charged particle is used as the target particle, and the downstream particles are then folded into this target particle.

– **Matching of Reconstructed 2D Hits to MCParticles** Each reconstructed 2D Hit is matched to the target MCParticle responsible for the largest deposit of energy in the region of space and time covered by the hit. The list of 2D Hits matched to each MCParticle is known as its collection of "true hits".
– **Matching of MCParticles to reconstructed Particles** For each target MCParticle, a record is stored if the MCParticle and a reconstructed Particle share a 2D Hit. As with the MCParticle hierarchy, the reconstructed Particle hierarchy can be used to fold Hit associations with daughter Particles into the parent Particles (i.e. the primary daughter of the neutrino).

The output of the above matching procedure is a matrix of associations between target MCParticles and reconstructed Particles. Each matrix element stores the number of 2D Hits shared between a given pair of MCParticles and Particles. The matching information obtained is comprehensive, but requires further interpretation to assess the performance of the reconstruction. An example print-out (prior to interpretation) is shown below, for a simple CC ν_μ quasi-elastic event.

```
-------------------------------------------------------------------------------------------------------------------
MCNeutrino: PDG 14, InteractionType 1001
RecoNeutrino: PDG 14

PrimaryMCParticle: ID #0, PDG 13, nMCParticleHits 925 (425, 101, 399)   [ Hit display format: Total (U, V, W) ]
 -MatchedRecoParticle: ID #0, RecoPDG 13, nMatchedHits 876 (401, 88, 387), nRecoParticleHits 876 (401, 88, 387)
 -MatchedRecoParticle: ID #1, RecoPDG 13, nMatchedHits 3 (2, 1, 0), nRecoParticleHits 21 (7, 10, 4)

PrimaryMCParticle: ID #1, PDG 2212, nMCParticleHits 20 (5, 9, 6)
 -MatchedRecoParticle ID #1, RecoPDG 13, nMatchedHits 18 (5, 9, 4), nRecoParticleHits 21 (7, 10, 4)
-------------------------------------------------------------------------------------------------------------------
```

In this example, there are two primary MCParticles, a muon and proton, and two primary reconstructed Particles (tracks, with PDG code 13). The total number of Hits and matched Hits is shown, as is the breakdown of Hits into the U, V and W views. The matching between the MCParticles and reconstructed Particles is not completely clean because the Particle representing the proton also contains 3 hits that are truly from the muon. A matching procedure is used to provide some kind of human interpretation of this reconstruction (the example event above is actually reconstructed quite accurately). This procedure finds all the "strong" matches between MCParticles and reconstructed Particles, then picks-up any remaining "weak" matches.

1. Find the strongest (most shared Hits) match between any MCParticle and reconstructed Particle.
2. Repeat step 1, using each MCParticle and reconstructed Particle at most once until no further strong matches are possible.
3. Assign any remaining reconstructed Particles to the MCParticle with which they share most Hits.

In step 3, the number of reconstructed Particles matched to a MCParticle can increase from e.g. one to e.g. two or three, but it can never increase if it is zero after the strong matching (this MCParticle must have been lost). It is possible to place selection cuts on the number of true Hits associated with target MCParticles and on the number of Hits that must be shared in order to define a match between a MCParticle and reconstructed Particle. Throughout this document, MCParticle targets are only considered if they have at least 15 associated Hits, whilst there must be at least 5 shared Hits in order to declare a match. These thresholds represent numbers of Hits deemed to identify reasonable/realistic targets for the reconstruction and to identify reasonable matching between target and reconstructed Particles.

Using the output of the interpretive matching scheme, it is possible to calculate some performance metrics to assess the pattern recognition, as shown below. It is also possible to construct tables showing the number of reconstructed Particles matched to each target MCParticle for specific interaction types. If the tables typically show one or more reconstructed Particles matched to each expected MCParticle, it indicates that the reconstruction is performing well. The following overall performance metrics are defined for each type of target MCParticle in an event sample:

Efficiency = Fraction of MCParticles with at least one matched reconstructed Particle
Completeness = Fraction of 2D Hits in a MCParticle shared with the reconstructed Particle
Purity = Fraction of 2D Hits in a reconstructed Particle shared with the MCParticle

In addition, an event is deemed to have a "completely correct" overall reconstruction if there is exactly one reconstructed Particle for each target MCParticle. This additional metric provides a useful but unforgiving picture of the overall performance of the pattern recognition.

Note that pattern recognition (i.e. the clustering of hits into reconstructed particles) is just one component of the MicroBooNE reconstruction and analysis, and therefore the performance metrics presented in this note must be folded in with the efficiencies of high-level reconstruction and neutrino event selection to obtain the overall efficiency of the MicroBooNE analysis.

5 Performance

The performance of the Pandora algorithms, described throughout Section 3, is now assessed using the metrics described in Section 4. The performance of the PandoraNu reconstruction is examined in detail by looking at specific neutrino interaction types in simulated neutrino data from the Booster Neutrino Beam (BNB). The performance of the PandoraCosmic reconstruction is then tested using simulated "off-beam" cosmic-ray data. Finally, a combined reconstruction chain containing both PandoraCosmic and PandoraNu is studied using simulated BNB neutrino interactions overlaid with simulated cosmic-ray data. This involves first running the PandoraCosmic reconstruction chain on the overlaid data, then tagging and removing clear cosmic-ray candidates, then running PandoraNu with a cosmic-removed Hit collection. The resulting neutrino reconstruction performance is evaluated.

The simulated event samples used in this performance assessment have been produced by the MicroBooNE experiment for its summer 2016 results. The data samples are based on LArSoft v05-08-00, which includes v2.10.4 of the GENIE [11] neutrino Monte Carlo event generator. The simulation of the MicroBooNE detector geometry incorporates the unresponsive parts of the readout; however, a full simulation of detector noise has not yet been implemented. The BNB sample comprised approximately 250k events, whilst the BNB sample with overlaid cosmic rays consisted of approximately 40k readout time-windows. For assessing neutrino events, a selection cut was placed to demand that there was a generated neutrino interaction position located in a fiducial volume of the detector. The fiducial volume was defined to be the volume enclosed up to 10 cm from the detector edges in the x and z dimensions and up to 20 cm from the edges in y. Following this cut, the BNB sample comprised approximately 100k events and the BNB sample with overlaid cosmic rays comprised approximately 13k neutrino events.

5.1 BNB CC quasi-elastic events: $\nu_\mu + N \rightarrow X + p + \mu^-$

This Section assesses the pattern recognition performance for BNB CC ν_μ quasi-elastic interactions. A specific subset of these interactions is examined: those producing exactly one muon and one proton, each with at least 15 true Hits. By using this specific subset of interactions, the challenge posed to the pattern recognition is clarified and the desired output from the reconstruction is made clear. An example event topology is displayed in Figure 12. The muon and proton momentum distributions for selected events both peak at approximately 400 MeV. Table 1 provides a thorough assessment of the pattern recognition performance for this kind of interaction. The columns label the target MCParticles for the events, the muon and proton, whilst the rows indicate how many reconstructed Particles are matched to these target MCParticles, using the approach outlined in Section 4. Events with a completely correct reconstruction would match exactly one Particle to the muon and exactly one Particle to the proton.

The Table shows that 90.7% of muons and 76.4% of protons yield exactly one reconstructed Particle. In addition, 71.1% of neutrino interactions in this event sample yielded exactly one reconstructed Particle for each target MCParticle and are deemed to be completely correct. A small fraction of muons are not reconstructed and a significant fraction (nearly 20%) of protons are also found not to be reconstructed. The most common mechanism by which target MCParticles are lost in this way proves to be accidental merging of the muon and proton into a single reconstructed Particle. Whichever of the two MCParticles has most Hits shared with

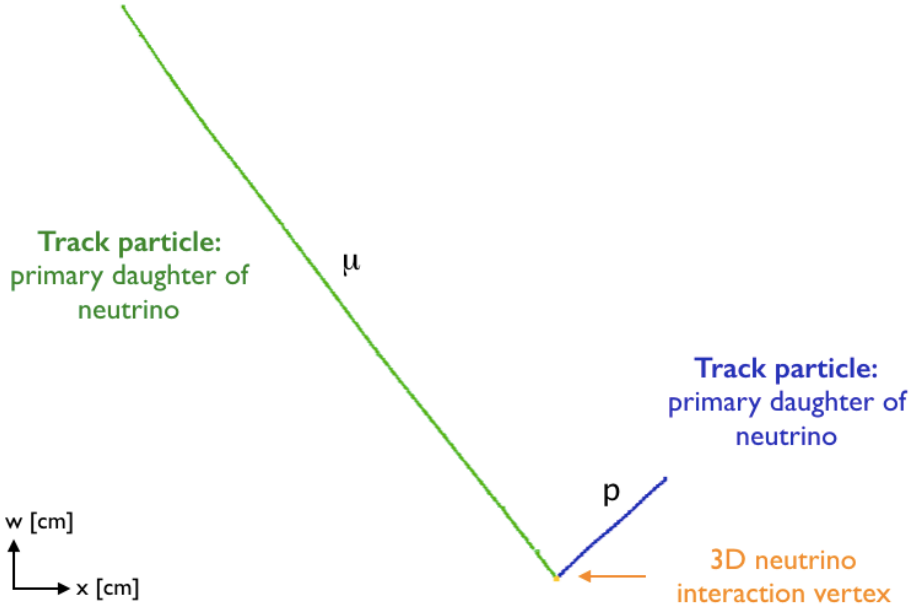


Fig. 12: An example CC ν_μ quasi-elastic interaction. Target particles for the reconstruction are the muon and proton.

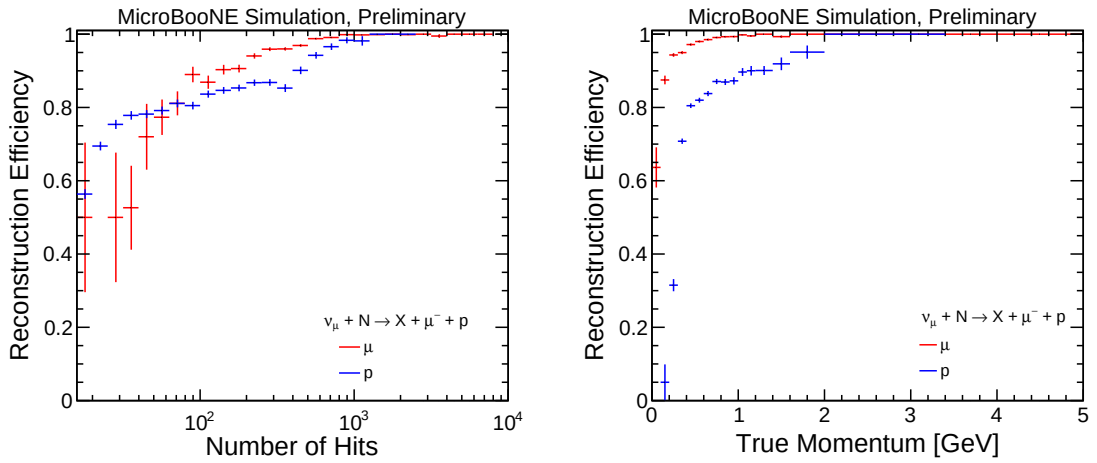
#Matched Particles	0	1	2	3+
μ	$(2.7 \pm 0.1)\%$	$(90.7 \pm 0.2)\%$	$(6.1 \pm 0.2)\%$	$(0.5 \pm 0.0)\%$
p	$(19.8 \pm 0.3)\%$	$(76.4 \pm 0.3)\%$	$(3.5 \pm 0.1)\%$	$(0.3 \pm 0.0)\%$

Table 1: pattern recognition performance for the target muon and proton in CC ν_μ quasi-elastic interactions. The total number of events was 22,102 and 15,718 of these had exactly one reconstructed particle matched to the muon and one matched to the proton, a success rate of 71.1%.

the reconstructed Particle will be declared as the matched MCParticle, whilst the other MCParticle will be declared lost. Some muons and protons are split-up into two reconstructed Particles; rather few are split-up into three or more. This kind of detailed information will help to drive the next phase of Pandora algorithm development. Events in which the muon is split into two Particles can, for instance, be scanned visually to look for recurring "problem topologies" and new algorithms added to address the most common issues.

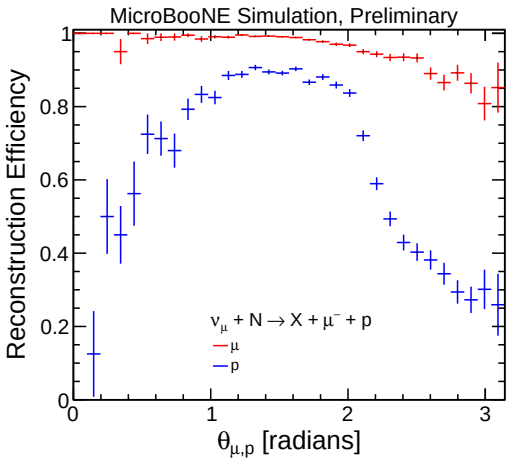
Figure 13 displays the reconstruction efficiencies for the target muon and protons as a function of their number of true Hits, as a function of their true momentum and as a function of the true opening angle between the muon and proton. The proton reconstruction efficiency is lower than the muon reconstruction efficiency across the full range of momenta, with the most common failure mechanism being merging of the muon and proton into a single Particle, as discussed above. The efficiency is actually a little better for protons with small numbers of Hits than for muons of the same size, presumably because of their respective $\mathrm{d}E/\mathrm{d}x$ distributions. As anticipated, the muon and proton are most likely to be merged into a single Particle when the two tracks are close to collinear. The single reconstructed Particle will then be matched to the target with which it shares most Hits, which will preferentially be the muon. In the specific configuration where the muon and proton tracks are almost back-to-back and the muon moves back towards the beam source, the muon is likely to be rather low energy and there is an increased chance of matching the single reconstructed Particle to the proton.

Figure 14 shows the completeness and purity of the reconstructed Particles with the strongest matches to the target muon and proton. Hand-scanning of events reveals that it is difficult to achieve high reconstructed completenesses for protons, as they are typically associated to a series of "isolated" Hits, in addition to those lying along the main proton track. It is highly unlikely that these isolated daughter Hits would be picked-up by track-oriented pattern recognition algorithms. These Hits are almost certainly from low energy neutrons generated as the proton propagates through the detector, but are associated to the proton via the MCParticle hierarchy, making the current definition of completeness extremely conservative. The purity distribution shows



(a)

(b)



(c)

Fig. 13: Reconstruction efficiencies for the target muon and proton in CC ν_μ quasi-elastic interactions (a) as a function of their number of true Hits, (b) as a function of their true momentum, and (c) as a function of the true opening angle between the muon and proton.

that proton Particles do not tend to track significantly into nearby muon Particles, although the reverse can often occur (and will occur in the dominant failure mechanism where a single Particle includes both the muon and proton targets).

Finally, Figure 15 displays the distribution of reconstructed vertex positions, relative to the generated neutrino interaction position. The rise at a ΔR value of 0.8 cm is associated with cases in which a single Particle has been reconstructed that encompasses both the muon and proton, placing the vertex at the end of one of these two target particles. It is found that 16% of events have ΔR values above 5 cm. Of these events, it would appear that a significant failure mechanism results from placing the vertex at the incorrect end of the muon track. This happens most commonly when the event kinematics are such that the muon direction is back towards the beam source. The presence of decay electrons can also make the incorrect end of the muon track appear favourable to the vertex reconstruction. Additional developments to the VertexSelection algorithm are foreseen in the near future.

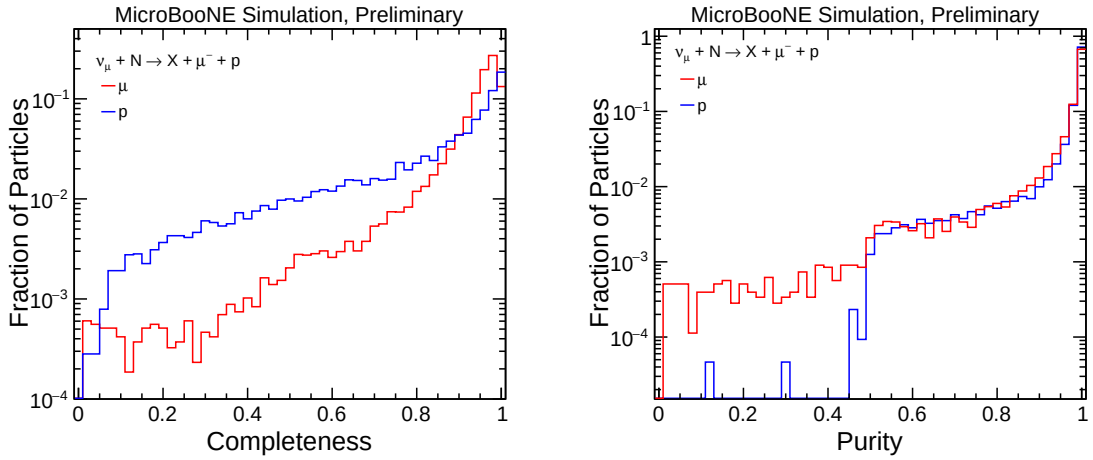


Fig. 14: Completeness and purity of the reconstructed Particles with the strongest matches to the target muon and proton in CC ν_μ quasi-elastic interactions.

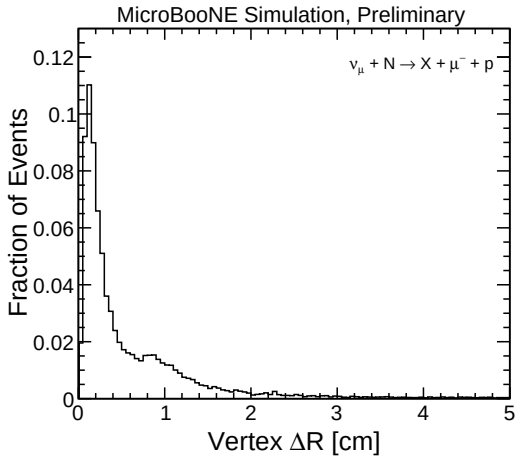


Fig. 15: Distribution of vertex resolution (ΔR, distance between generated and reconstructed 3D vertex positions) for CC ν_μ quasi-elastic interactions. 16% of events have ΔR values above 5 cm.

5.2 BNB CC resonance events: $\nu_\mu + N \to X + p + \pi^+ + \mu^-$

This Section assesses the pattern recognition performance for BNB CC ν_μ interactions with associated charged pion production, through the excitation of baryon resonances. A specific subset of events of this interaction type is selected: those producing exactly one muon, one proton and one charged pion, each associated to at least 15 Hits. An example event topology is shown in Figure 16. This example event also happens to show clear decay of the muon to an electron. The momentum distributions for particles in selected events peak at approximately 300 MeV for muons, 400 MeV for protons and 200 MeV for charged pions. In comparison to the quasi-elastic events considered in Section 5.1, the challenge for the pattern recognition is to handle the presence of an additional track, from the charged pion.

Table 2 provides a thorough overview of the performance for this interaction type. The fractions of target MCParticles that result in a single reconstructed Particle are: 87.7% for muons, 75.0% for protons and 71.3% for pions. A total of 50.8% of neutrino interactions in this event sample are deemed to be completely correct, yielding exactly one reconstructed Particle per target MCParticle. The performance for muons and protons is similar to that observed in quasi-elastic events. One notable difference, however, is the small increase in the

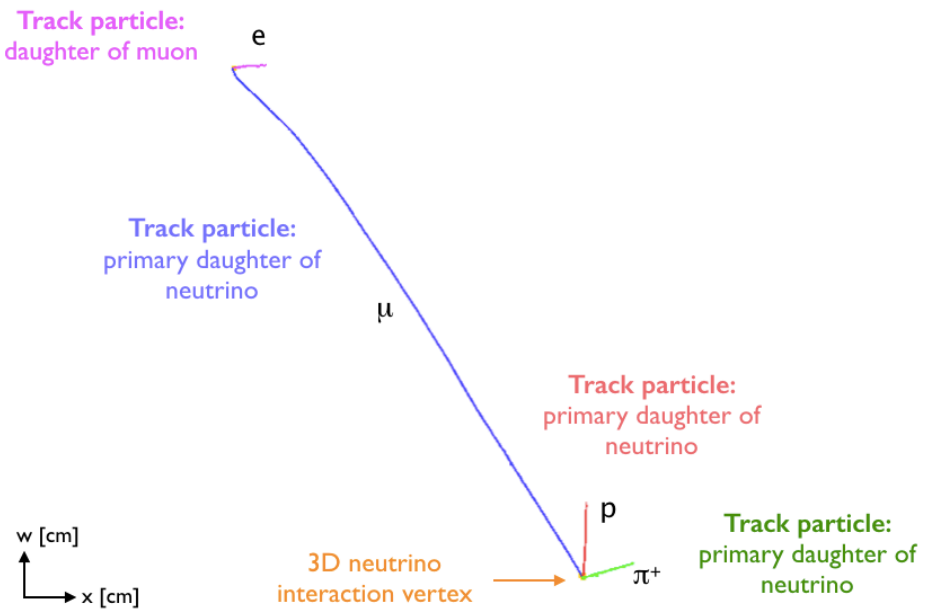


Fig. 16: An example CC ν_μ interaction with resonant pion production. Target particles for the reconstruction are the muon, proton and charged pion.

#Matched Particles	0	1	2	3+
μ	$(6.8\pm0.3)\%$	$(87.7\pm0.4)\%$	$(5.1\pm0.3)\%$	$(0.4\pm0.1)\%$
p	$(20.5\pm0.5)\%$	$(75.0\pm0.6)\%$	$(4.0\pm0.3)\%$	$(0.5\pm0.1)\%$
π^+	$(11.6\pm0.4)\%$	$(71.3\pm0.6)\%$	$(13.0\pm0.4)\%$	$(4.1\pm0.3)\%$

Table 2: Pattern recognition performance for the target muon, proton and pion in CC ν_μ interactions with resonant pion production. The total number of events was 6,070 and 3,084 of these had exactly one reconstructed particle matched to each MCParticle target, a success rate of 50.8%.

number of muons that have no matched reconstructed Particles. This is associated with events for which the muon and pion tracks are merged into a single Particle, and the matching procedure associates the Particle to the pion. In these events, a loss of efficiency does not tend to mean that a true particle has been missed entirely, but rather that its Hits have been accidentally Clustered together with those from a different true particle. The pions typically have a complicated series of true daughter particles and it is a challenge to provide individual reconstructed Particles for each of these and to ensure that their hierarchy is reconstructed completely. This explains the frequency at which the target pion is reported to be matched to two, three or more reconstructed Particles.

Figure 17 displays the reconstruction efficiencies for the target muon, proton and pion as a function of their number of true Hits, their true momentum and their generated angle to the beam (z) direction. Figure 18 shows the completeness and purity of the reconstructed particles with the strongest matches to the target muon, proton and pion. The reported completeness is low for the target pions because of their often complex and extended series of true daughter particles, plus their reported true associations to multiple isolated Hits, which are unrealistic targets for inclusion by the pattern recognition. Finally, Figure 19 displays the distribution of reconstructed vertex positions, relative to the generated neutrino interaction position.

5.3 BNB CC resonance events: $\nu_\mu + N \rightarrow X + p + \pi^0 + \mu^-$

This Section assesses the pattern recognition performance for BNB CC ν_μ interactions with associated neutral pion production. A specific subset of events of this interaction type is selected: those producing exactly one muon, one proton and two photons, from pi-zero decay. Each of these target particles must be associated to at least 15 Hits. An example event topology is shown in Figure 20. The momentum distributions for particles in

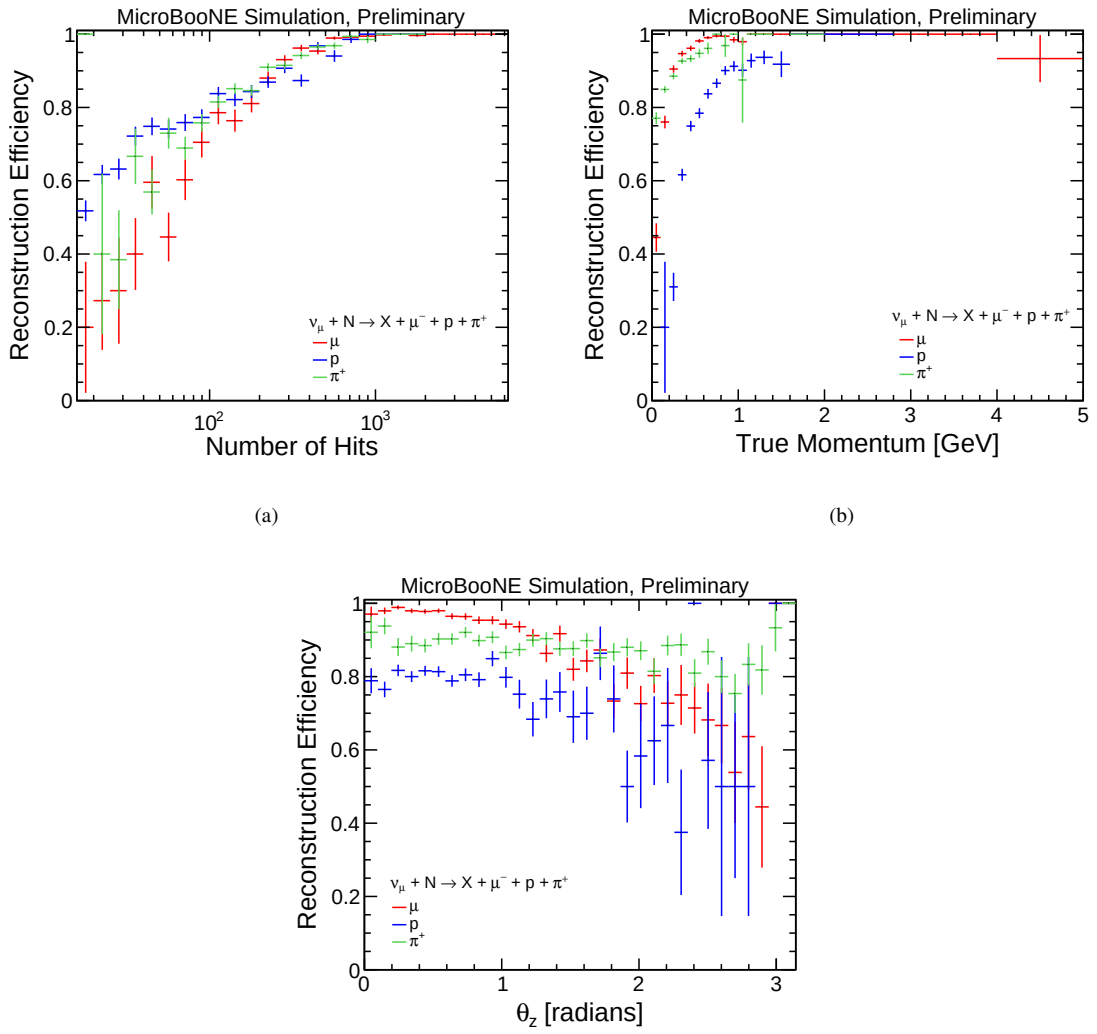


Fig. 17: Reconstruction efficiencies for the target muon, proton and pion in CC ν_μ interactions with resonant pion production (a) as a function of their number of true Hits, (b) as a function of their true momentum, and (c) as a function of their angle to the beam direction.

selected events peak at approximately 300 MeV for muons and 400 MeV for protons. The energy distributions for photons in selected events peak at approximately 150 MeV for the larger photons (most associated Hits) and 60 MeV for the second, smaller photons. It is clear that the presence of two photon-induced showers presents a different challenge to the track-only topologies considered in Sections 5.1 and 5.2. Small opening angles between the two showers will likely cause them to be merged into a single reconstructed Particle. Sparse shower topologies can alternatively lead single showers to be split into multiple reconstructed Particles.

Table 3 provides a thorough overview of the performance for this interaction type. The reconstruction performance for muons and protons remains similar to that seen in previous Sections. The fractions of MC-Particles that yield exactly one reconstructed Particle are 87.4% for muons, 74.0% for protons. A slightly larger fraction of lost muons associated with a new failure mechanism, whereby muons could be merged into showers if the topology is complicated and the vertex reconstruction is poor. As anticipated, the diverse and complex shower topologies lead to problems with both incorrect merging and splitting of Particles. In the Table, γ_1 is defined to be the target shower with the largest number of true Hits and it is matched to exactly one reconstructed Particle in 60.0% of events. In nearly 10.8% of events, no Particle is matched to the largest shower and this failure mechanism is typically associated with small showers being accidentally merged with

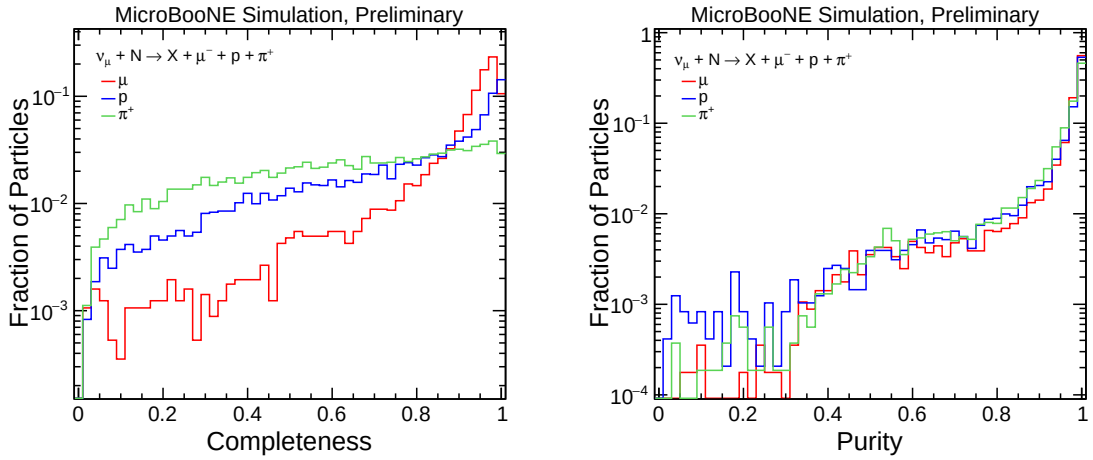


Fig. 18: Completeness and purity of the reconstructed Particles with the strongest matches to the target muon, proton and pion in CC ν_μ interactions with resonant pion production.

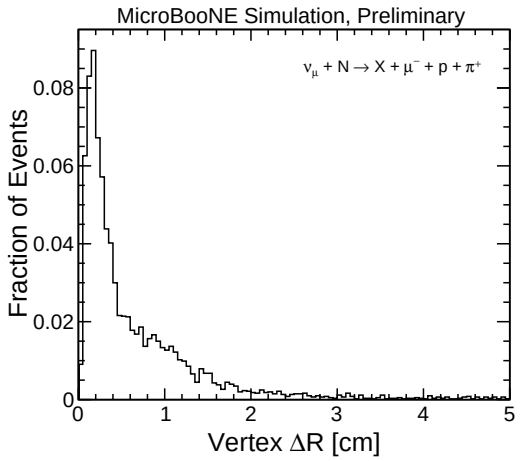


Fig. 19: Distribution of vertex resolution (ΔR, distance between generated and reconstructed 3D vertex positions) for CC ν_μ interactions with resonant charged pion production. 15% of events have ΔR values above 5 cm.

one of the track particles. Sparse shower topologies can often mean that the largest shower is reconstructed as multiple, distinct shower Particles.

The smaller of the two showers, γ_2, is matched to exactly one reconstructed Particle in 51.0% of events. This smaller shower can also prove to be extremely sparse and can be split between multiple reconstructed Particles. The dominant failure mechanism for the smaller shower is, however, accidental merging into a nearby Particle, typically that associated with the larger shower. There is a fundamental tension to address in the pattern recognition, whereby algorithms need to be inclusive to avoid splitting true showers into multiple reconstructed Particles, but algorithms also need to avoid merging-together Hits from nearby separate Particles. Improved labelling of 2D Clusters to indicate whether they are track-like or shower-like can help to address this tension, as can careful identification of separate shower spines and use of the neutrino vertex to protect individual particles emerging from the interaction.

This type of interaction yields particularly challenging topologies, and thus only 22.3% of events have exactly one reconstructed Particle for each target MCParticle with the current chain of algorithms. Further algorithm improvements across all areas are foreseen in the near future.

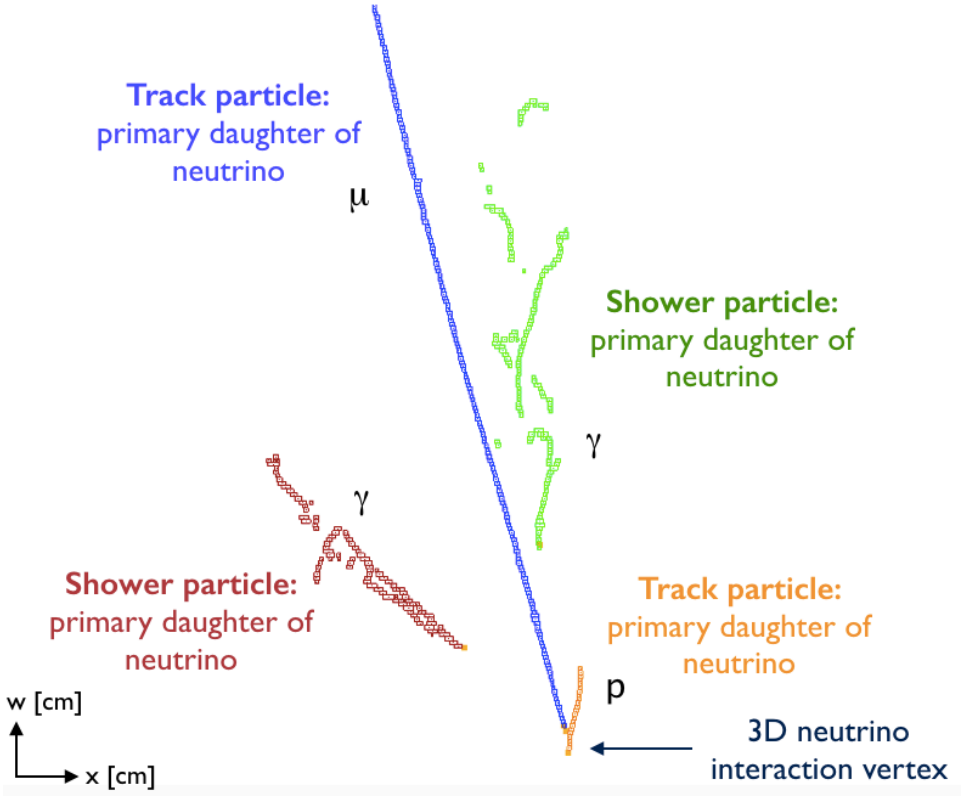


Fig. 20: An example CC ν_μ interaction with resonant pion production. Target particles for the reconstruction are the muon, proton and two photons (from pi-zero decay).

#Matched Particles	0	1	2	3+
μ	$(7.9 \pm 0.6)\%$	$(87.4 \pm 0.8)\%$	$(4.4 \pm 0.5)\%$	$(0.3 \pm 0.1)\%$
p	$(19.8 \pm 0.9)\%$	$(74.0 \pm 1.0)\%$	$(5.5 \pm 0.5)\%$	$(0.7 \pm 0.2)\%$
γ_1	$(10.8 \pm 0.7)\%$	$(60.0 \pm 1.1)\%$	$(18.2 \pm 0.9)\%$	$(11.0 \pm 0.7)\%$
γ_2	$(35.9 \pm 1.1)\%$	$(51.0 \pm 1.2)\%$	$(10.1 \pm 0.7)\%$	$(3.0 \pm 0.4)\%$

Table 3: Pattern recognition performance for the target muon, proton and two photons (photons are ordered by number of true Hits) in CC ν_μ interactions with resonant pion production. The total number of events was 1,874 and 417 of these had exactly one reconstructed particle matched to each MCParticle target, a success rate of 22.3%.

Figure 21 shows the reconstruction efficiency for the target muon, proton and two showers as a function of their number of true Hits. Figures 22 and 23 show the completeness and purity of the reconstructed Particles with the strongest matches to the target particles. The completeness is markedly lower for the reconstructed shower Particles than for the reconstructed track Particles. This is associated with the problems of splitting-up sparse showers into multiple Particles, but also indicates that there are additional, probably small, shower remnants that could be collected by a new Particle refinement algorithm. Finally, Figure 24 displays the distribution of reconstructed vertex positions, relative to the generated neutrino interaction position.

5.4 All BNB Interactions

No longer considering specific interaction types, this Section provides a short summary of particle reconstruction efficiency and vertex reconstruction performance for all BNB interactions, including NC interactions. Figure 25 displays the reconstruction efficiencies for the largest (most true Hits) muon, proton, charged pion and photon in all interactions. These efficiencies are shown as a function of the number of true Hits associated to the MCParticle. Figure 26 focuses on reconstruction efficiency as a function of true particle momentum,

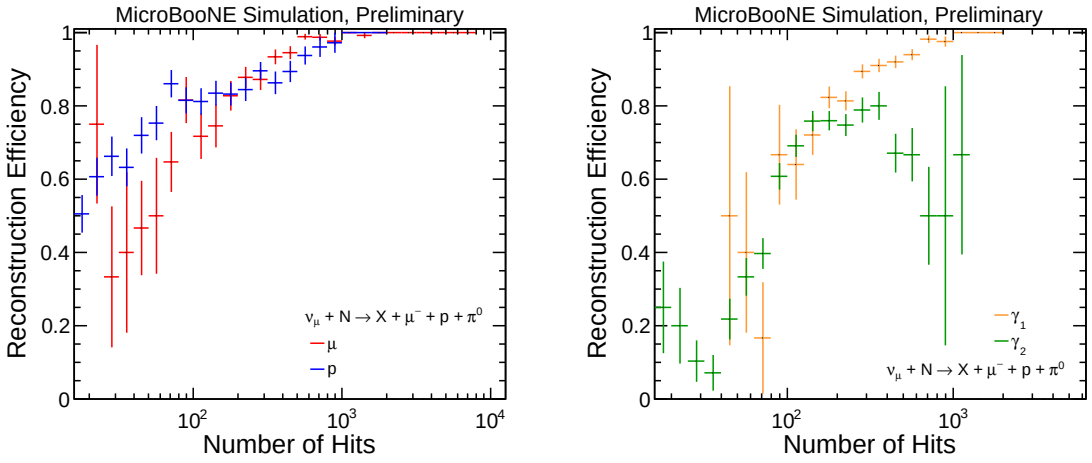


Fig. 21: Reconstruction efficiencies for the target muon, proton and photons in CC ν_μ interactions with resonant pion production as a function of their number of true Hits.

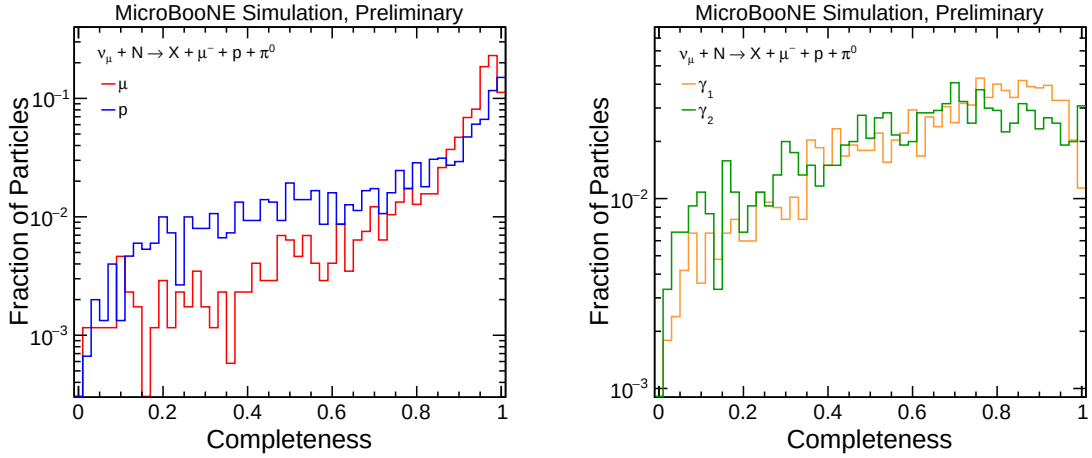


Fig. 22: Completeness of the reconstructed Particles with the strongest matches to the target muon, proton and two photons in CC ν_μ interactions with resonant pion production.

for the largest muon and proton in all interactions. All efficiencies shown are broadly consistent with the performance observed in the previous Sections, looking at specific interaction types for the BNB.

Figure 27 shows the distribution of reconstructed vertex positions, relative to the generated neutrino interaction position, for all interaction types. The vertex resolution is also broken down into its component Δx, Δy and Δz distributions. As expected, the vertex resolution is best for the x coordinate, common to all views. There is a small asymmetry in the Δz distribution, which is under investigation and believed to be associated with Hits shared by multiple, overlapping particles (e.g. muons and protons) near the interaction vertex. Looking at all BNB interactions means that there are a sizeable number of interactions for which there are only a few Hits in the events and/or only a few primary particles, often neutrons. In these events it is not always possible to reconstruct a neutrino. Of all interactions, both CC and NC, for which a neutrino was produced, 15% of these interactions had a vertex ΔR value above 20 cm.

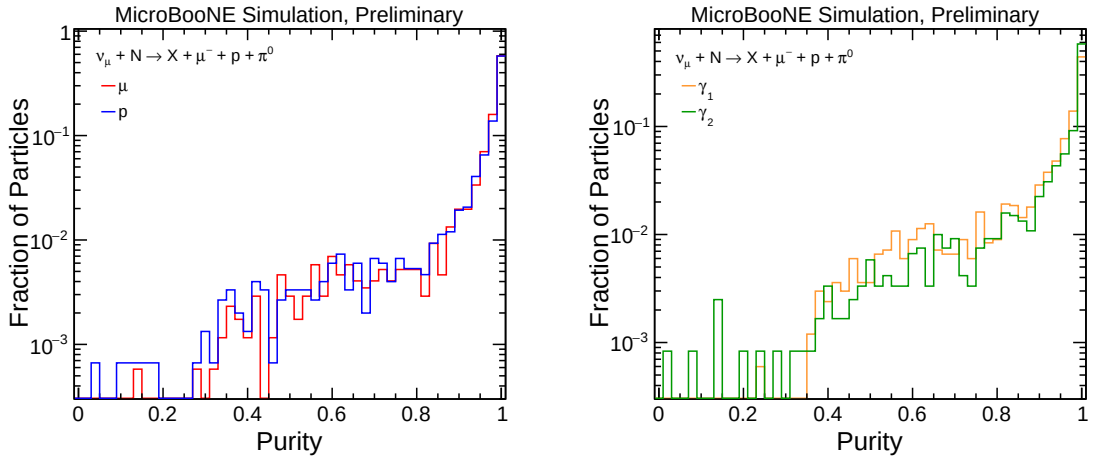


Fig. 23: Purity of the reconstructed Particles with the strongest matches to the target muon, proton and two photons in CC ν_μ interactions with resonant pion production.

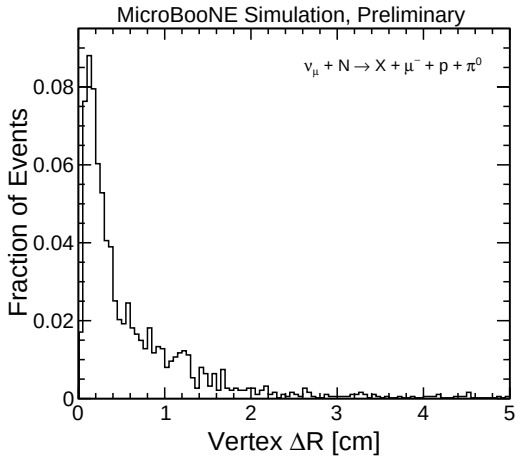


Fig. 24: Distribution of vertex resolution (ΔR, distance between generated and reconstructed 3D vertex positions) for CC ν_μ interactions with resonant pi-zero production. 18% of events have ΔR values above 5 cm.

5.5 Cosmic Rays

The pattern recognition performance of the PandoraCosmic reconstruction is assessed using 10k readout time-windows from a BNB plus cosmic ray event sample. The target MCParticles are the cosmic-ray muons and Hits from the daughter delta rays are folded-back and associated with the parent muon. Figure 28 displays the cosmic-ray muon reconstruction efficiency as a function of the number of true associated Hits and as a function of angle with respect to the z-axis. The same Figure also presents the completeness of the reconstructed Particles most strongly associated to each target muon. The reconstruction efficiency is above 80% for cosmic rays associated to even small numbers of Hits. It rises to about 95% for cosmic rays associated to at least 100 Hits and slowly approaches perfect efficiency. The most probable mechanism by which individual cosmic rays may be lost is accidental merging of multiple true particles, rather than a failure to cluster or match features between views for any one true particle. The reconstruction efficiency is rather flat as a function of angle. The completeness of the reconstructed Particles is encouraging, considering that the completeness metric assesses the ability of the reconstruction to identify all delta rays and associate them with the correct parent muons.

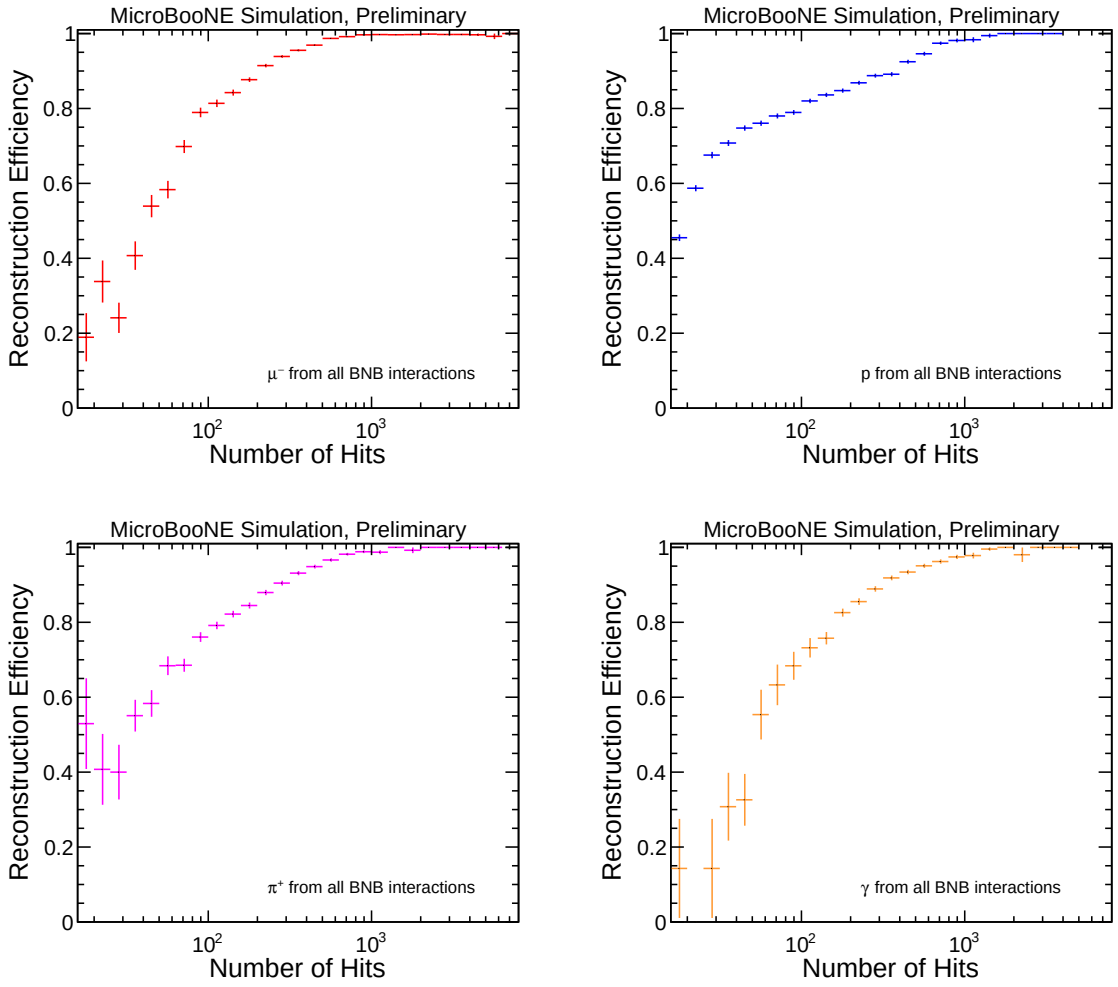


Fig. 25: Reconstruction efficiencies for the largest (most Hits) muon, proton, charged pion and photon in all BNB interactions, shown as a function of their number of true Hits.

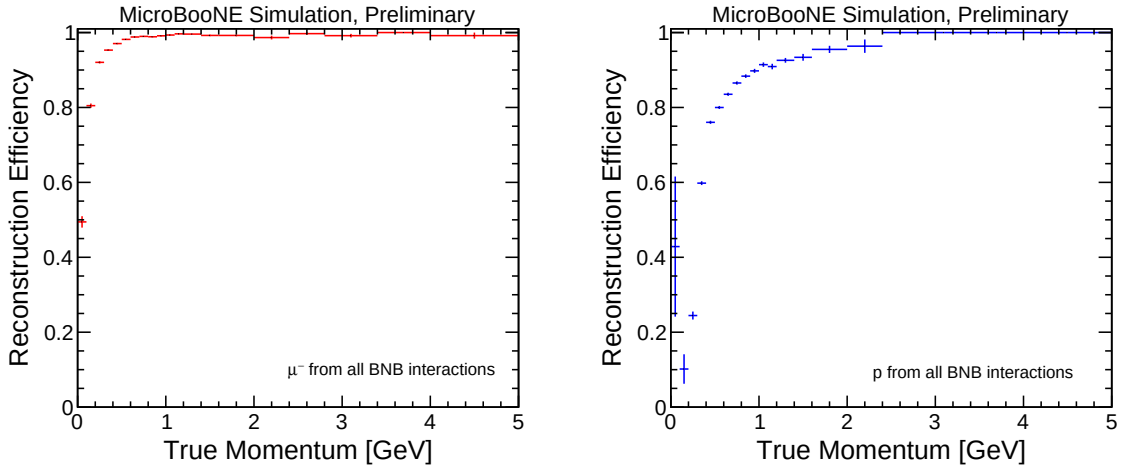


Fig. 26: Reconstruction efficiencies for the largest (most Hits) muon and proton in all BNB interactions, shown as a function of their true momentum.

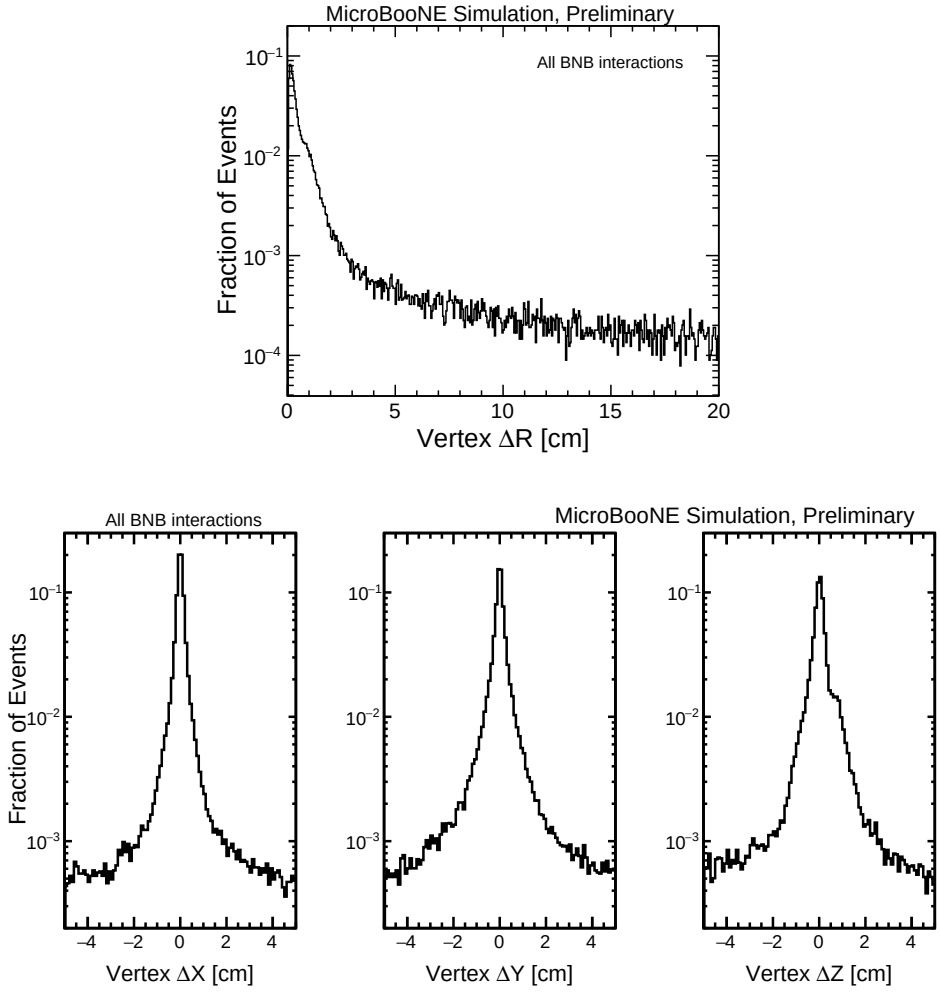


Fig. 27: Distribution of vertex resolution (ΔR, distance between generated and reconstructed 3D vertex positions) for all BNB interactions. 15% of all events yielding a reconstructed neutrino have ΔR values above 20 cm. The component Δx, Δy and Δz distributions are also shown.

5.6 Impact of Cosmic Ray Overlay

As a final assessment of Pandora pattern recognition performance at MicroBooNE, the neutrino reconstruction performance is investigated after the full procedure of running PandoraCosmic, tagging and removing clear cosmic-ray candidates, then running PandoraNu on a cosmic-removed Hit collection. The interaction type considered is BNB CC ν_μ quasi-elastic interactions producing exactly one muon and one proton, each with at least 15 true Hits after cosmic-ray removal. The results are to be compared with those obtained in Section 5.1, where there was no cosmic-ray background. Likely failure modes due to the presence of cosmic rays are:

- Removal of key features of the neutrino interaction before the PandoraNu reconstruction. This could be associated with an inability of the PandoraCosmic reconstruction to separate the neutrino interaction from nearby cosmic rays. It could also be associated with incorrect tagging of the neutrino interaction as a cosmic ray, perhaps because one or more of its daughter Particles (as reconstructed by PandoraCosmic) approach the edge of the detector.
- Confusion of the PandoraNu pattern recognition by the presence of cosmic-ray remnants in the cosmic-removed Hit collection. It is then the responsibility of the Pandora event slicing tools to try to ensure that the Hits from the neutrino interaction and the Hits from the cosmic-ray remnants end up in different slices, and so appear as separate reconstructed candidate neutrinos. Nearby neutrinos and cosmic remnants can make the slicing rather challenging.

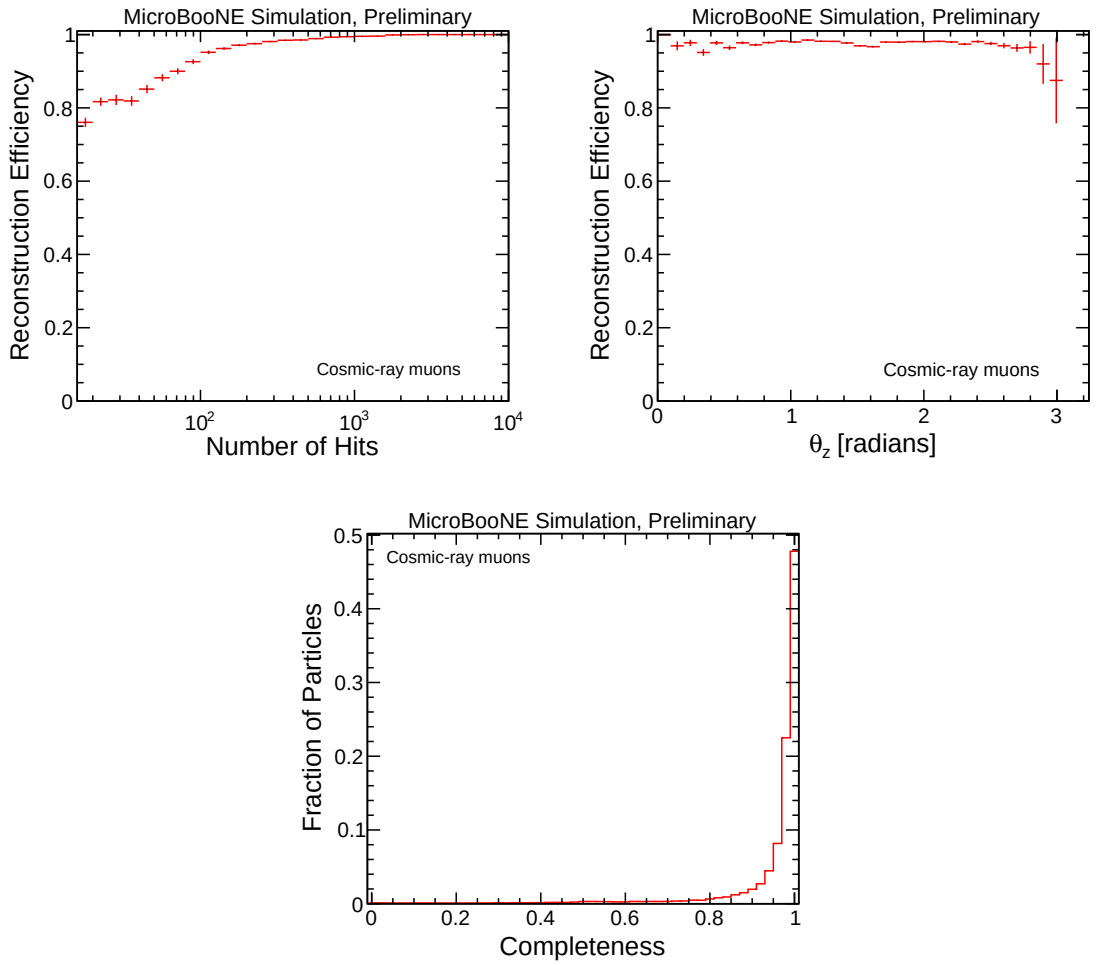


Fig. 28: Cosmic-ray muon reconstruction efficiency as a function of the number of true associated Hits and as a function of angle with respect to the z-axis. The completeness of the reconstructed Particles with the strongest matches to the target cosmic rays is also shown.

Table 4 provides a thorough assessment of the pattern recognition performance for this kind of interaction. In comparison with Table 1, without cosmic-ray background, the performance is encouragingly similar. The only significant difference is the small rise in the fraction of target MCParticles for which no Particles are reconstructed. This can be associated with the two-pass reconstruction approach and the cosmic-ray removal procedure. The same feature, a small loss of efficiency, can be observed in Figure 29, which shows reconstruction efficiencies for the target muon and proton with and without cosmic-ray background. These efficiencies, shown as a function of the number of associated true Hits in the input Hit collection, are encouragingly similar with and without the background.

#Matched Particles	0	1	2	3+
μ	$(5.7 \pm 0.6)\%$	$(87.0 \pm 0.8)\%$	$(6.9 \pm 0.6)\%$	$(0.4 \pm 0.2)\%$
p	$(24.1 \pm 1.0)\%$	$(73.8 \pm 1.1)\%$	$(2.1 \pm 0.3)\%$	$(0.0 \pm 0.0)\%$

Table 4: pattern recognition performance for the target muon and proton in CC ν_μ quasi-elastic interactions with cosmic-ray background, to be compared to the data in Table 1, which are calculated without cosmic-ray background. With cosmic-ray background, the total number of events was 2,598 and 1,742 of these had exactly one reconstructed particle matched to the muon and one matched to the proton, a success rate of 67.1% This is to be compared with a success rate of 71.1% without background.

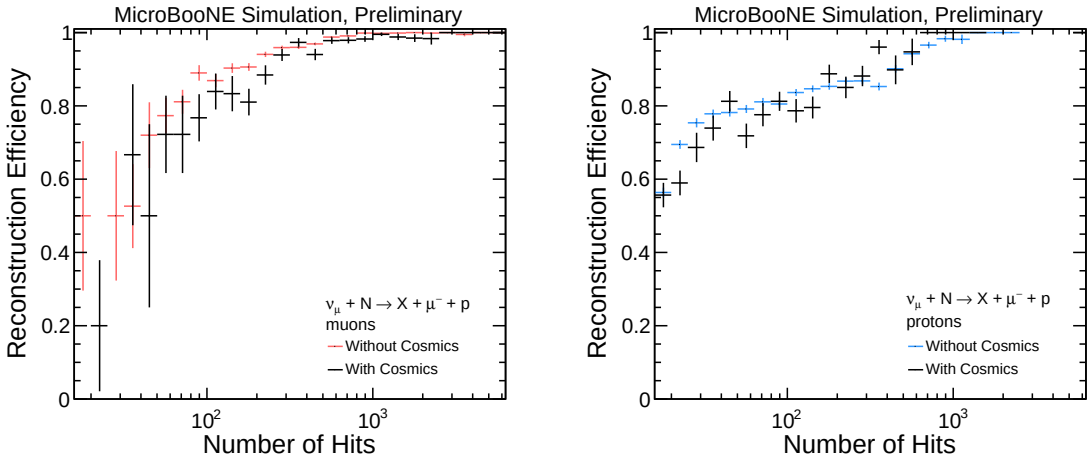


Fig. 29: Reconstruction efficiencies for the target muon and proton in CC ν_μ quasi-elastic interactions with and without cosmic-ray background. The efficiencies are shown as a function of their number of true Hits in the input Hit collections.

6 Concluding Comments

This note has introduced the Pandora multi-algorithm approach to pattern recognition in LAr TPC detectors and discussed the specific chains of algorithms used to reconstruct neutrino interactions and cosmic-ray muons in MicroBooNE. The reconstruction performance has been assessed for a number of different particle and interaction types using simulated data from MicroBooNE both with and without cosmic-ray background. A good level of performance is observed for many event types; in particular, a high reconstruction efficiency is obtained for inclusive muons from BNB neutrino interactions. The reconstruction of short tracks and complex topologies remains a challenge and a number of issues have been highlighted for future development.

This note also supports the latest results on inclusive ν_μ CC interactions by MicroBooNE [6], in which Pandora is utilised for pattern recognition. It should be noted that the current inclusive ν_μ CC analysis applies a number of tight selection cuts, such as requirements on the length and containment of reconstructed tracks, in order to separate ν_μ CC interactions from their NC and cosmic-ray backgrounds. These requirements lower the overall selection efficiency of the analysis relative to the reconstruction efficiencies presented in this note. A number of improvements to the reconstruction and analysis of the MicroBooNE data are either planned or in progress, as described in [6].

The multi-algorithm approach to pattern recognition implemented by the Pandora SDK framework is a powerful technique, and the reconstruction developed for the MicroBooNE experiment contains a number of sophisticated pattern recognition algorithms. The current level of reconstruction performance is encouraging. The Pandora reconstruction is well-placed for the addition of new algorithms to address highlighted issues and increasingly complex topologies.

References

1. C. Anderson et al., JINST 7 (2012), 10020
2. M. Antonello et al., Adv. High Energy Phys. 2013 (2013), 260820
3. J.J. Back et al., Eur.Phys.J. C73 (2013) 2591
4. MicroBooNE Public Note 1019 (2016): http://www-microboone.fnal.gov/publications/publicnotes/
5. MicroBooNE Public Note 1002 (2016): http://www-microboone.fnal.gov/publications/publicnotes/
6. MicroBooNE Public Note 1010 (2016): http://www-microboone.fnal.gov/publications/publicnotes/
7. MicroBooNE Public Note 1012 (2016): http://www-microboone.fnal.gov/publications/publicnotes/
8. J. S. Marshall and M. A. Thomson, The European Physical Journal C 75, 439
9. LArSoft Software: https://cdcvs.fnal.gov/redmine/projects/larsoft
10. MicroBooNE Software: https://cdcvs.fnal.gov/redmine/projects/uboonecode
11. C. Andreopoulos et al., Nucl. Instrum. Meth. A 614 (2010), 87