Reconstruction and Selection of Neutrino Interactions in MicroBooNE using Deep Convolutional Neural Networks

MICROBOONE-NOTE-1123-PUB

The MicroBooNE Collaboration*

Fermi National Accelerator Laboratory, Batavia, Illinois

Abstract

In this document, we describe a new reconstruction workflow developed for the MicroBooNE experiment. It features the use of Deep Convolutional Neural Networks trained to recognize key structures within the data sufficient for the 3D reconstruction of neutrino interactions within the detector. As a test of the reconstruction utility, the products of the reconstruction workflow are used to select inclusive charged-current (CC) ν_e and ν_μ interactions in both simulated and real MicroBooNE data. In simulation, our ν_e and ν_μ selections achieve an efficiency of 57% and 68%, respectively, with a purity of 91% and 96%, respectively. We find that these selections are competitive with the inclusive selections used for the most recent MicroBooNE LEE searches. In particular, the CC-ν_e inclusive selection efficiency improves by over 20% while also improving sample purity. As a first step in quantifying potential bias, the data and Monte Carlo expectations are compared for both selections using the MicroBooNE open data. Within statistical and systematic uncertainties, both the electron and muon CC-inclusive event samples agree. A comparison of the real data events chosen by our work and another reconstruction framework shows that the two analyses each identify a sizeable fraction of events the other does not. This suggests that future analyses integrating the strengths of each could lead to combined gains. This work demonstrates, for the first time on real LArTPC data, state-of-the-art neutrino interaction reconstruction centered around deep learning algorithms.

* MICROBOONE_INFO@fnal.gov

CONTENTS

I. INTRODUCTION

The liquid argon time projection chamber (LArTPC) is the detector technology of choice for several future and current neutrino experiments. Current experiments include MicroBooNE [1], the Short Baseline Neutrino Detector [2], and ICARUS [3]. Future experiments notably include the Deep Underground Neutrino Experiment (DUNE) [4] an effort towards which several prototype LArTPCs [5] have been constructed. LArTPCs have now found their way into many experiments due to their combination of resolution and scalability. LArTPCs can track charged particle trajectories with millimeter-scale position resolution for detectors with target volumes into the tens of kilotons.

The output of LArTPCs can be characterized as very image-like. The waveforms recorded from planes of sense wires can be naturally arranged to produce images of the ionization patterns left behind by charged particles traversing the detector. Image formats are also relevant for alternative readout designs for LArTPCs, such as those that directly measure the 2D location of ionization in order to naturally capture voxelized 3D trajectories [6]. The format of this spatial data has facilitated the application of newly developed machine learning techniques, in particular from the domain of computer vision, to the task of reconstructing the trajectories and particle interactions captured by LArTPCs. Early applications focused on the classification of either entire images cropped from the data or for individual pixels [7]. Structures traditionally important in the reconstruction of interactions, such as the location of neutrino vertices have been searched for. High-level, more abstract quantities such as neutrino interaction flavor [8], the energy of electromagnetic (EM) showers [9], and the neutrino energy [10] have been targets of ML algorithms. While these applications were on 2D image data, there has been much progress in developing a full reconstruction chain for 3D voxelized data.

A fully end-to-end machine learning workflow outputs pixel-wise particle classification, the location of key points on particle trajectories, particle clusters, and assembled neutrino interactions into a fully-differentiable workflow [11]. The application of ML to reconstruction has indeed seen rapid progress in the past several years. However, demonstrations of these ML-based tools in the context of analyses of real LArTPC data are only starting to be realized. One such analysis centered around a CNN performing pixel-wise particle-type classification [12, 13] which was used as a central input for the targeted exclusive selection of one-lepton and one-proton final state interactions [14]. This analysis was part of MicroBooNE's search for an excess of low-energy electron neutrino interactions [15], conducted to investigate the reported event excess observed by the MiniBooNE experiment [16].

This document provides a description of a new "DLGen2" reconstruction and applies it to the selection of inclusive ν_eCC and ν_μCC interactions in MicroBooNE. Unlike the previous MicroBooNE DL-based analysis, this iteration has aimed for the general reconstruction of all charged particle trajectories coming from neutrino interactions. An overall evaluation and demonstration of the reconstruction is conducted through the execution of the selection on the MicroBooNE open neutrino data set. We find that the efficiencies for this analysis are competitive with the highest-efficiency search previously published by MicroBooNE [17], which utilizes the Wire-Cell reconstruction [17–19]. Furthermore, the inspection of events selected by our reconstruction finds unique events not found by the analysis of Ref. [17].

II. A CNN-BASED NEUTRINO RECONSTRUCTION FOR LARTPCS

A. Overview of the Reconstruction

The reconstruction utilizes convolutional neural networks to enable both 3D energy deposit reconstruction and perform particle ID on 2D images. The approach taken makes use of the different advantages inherent in the 2D image and 3D point cloud representations. We start the description of the reconstruction chain with a brief overview of the major components of the reconstruction, which are illustrated in figure 1. Later sections then will describe the algorithms used in each component in more detail.

The input to the reconstruction is a set of three 2D images, one for each of the wire-planes installed inside the MicroBooNE liquid argon TPC (LArTPC) detector [1]. The waveforms arranged in these images are the output of the first pre-processing stage applied to the raw waveforms. This stage includes the removal of coherent noise seen in sets of neighboring channels [20]. It also reconstructs the original space charge distribution from the measurements on wires by reverting the detector response (e.g., electronic response and field response) and sparsifying the input images. We call this process "Signal Processing." For more details on the pre-processing stage see Refs. [21, 22].

The image set is then provided to two convolutional neural networks (CNN). The first

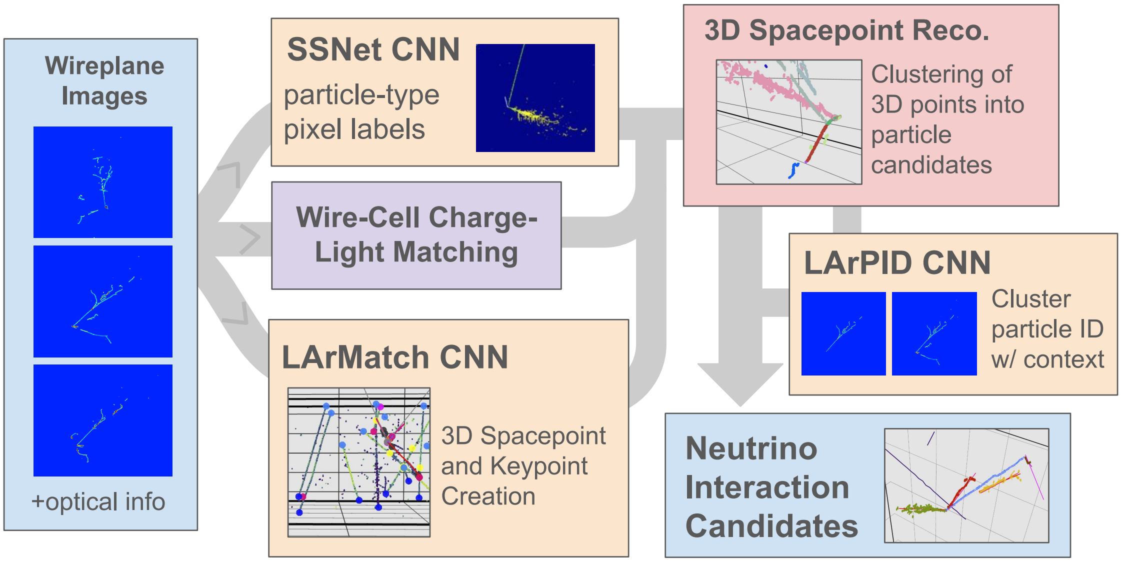


Figure 1: Overview of the reconstruction workflow. The wireplane images and optical information are passed into several components to produce labeled spacepoints reconstructing the location of ionization left behind by charged particles traversing the LArTPC. The labels associated to each spacepoint include particle type and a tag estimating if a given spacepoint is from beam-related or cosmic-ray particles. These points are then clustered into candidate particle trajectories by a set of reconstruction algorithms. A CNN is used to provide a particle-type label for each trajectory. The final output of the workflow are candidate neutrino interactions formed by associating one or more particle clusters to neutrino vertex candidates.

CNN acts on each wire plane image separately and is responsible for labeling each pixel in the image according to two broad particle categories, split by the spatial pattern of ionization produced. The first type is "track"-like trajectories coming from particles such as muons, charged pions, and protons. The second type is "shower"-trajectories produced by electromagnetic cascades initiated by electrons or photons interacting in the detector. This pixel labeling CNN is referred to as "SSNet" for the semantic-segmentation network and was used in the first MicroBooNE DL analysis. The details of SSNet can be found in Ref. [13].

A second, new CNN is applied to the set of three 2D images collectively and is referred to as the "LArMatch" network. The network produces two outputs. The first is a set of candidate 3D spacepoints which represent the location of energy depositions consistent with input images. The second product is a set of scores for six different classes of "keypoints." Keypoints are useful locations at the start or end of tracks and showers, which, if known, can greatly simplify the algorithms required to help with clustering and the formation of neutrino interaction candidates. The keypoint classes consist of 1) potential neutrino interaction vertices, 2) the start of track-like particles, 3) the end of track-like particles, 4) start points of EM showers excluding those from delta rays and muon decay, (5) the start points of delta ray showers, and 6) the start point of showers from muon decay.

The input images also go through a reconstruction workflow separate from the one described here. This workflow, referred to as the "Wire-Cell" reconstruction [18], builds spacepoints, does clustering, and matches clusters of charge to pulses of light seen in the optical detectors [19]. The workflow uses many non-ML approaches and features the application of compressed sensing. What we utilize in our reconstruction workflow is the association of charge clusters to pulses of light either inside or outside the neutrino beam window. This information is used to provide a tag for the spacepoints made by the LArMatch network, as either in-time or out-of-time with the neutrino beam.

At this point in the reconstruction, we have a set of spacepoints with various tags deriving from pixel-based labels along with a collection of keypoints. The next step in the workflow is to reconstruct 3D spacepoints, which are then clustered into subclusters covering individual particle trajectories. The purpose of starting with subclusters is to emphasize the purity of the clustering over completeness. Here the purity refers to the largest fraction of points whose ground truth label is associated to the end of the event. A pure cluster would contain spacepoints associated to only one particle. The completeness measures the fraction of possible pixels or spacepoints in the cluster.

The 3D clustering algorithm implements the commonly used Density-Based Scan (DB-Scan) algorithm which uses the distances between k-nearest neighbors. What allows for this simple clustering routine is the many tags coming from LArMatch, SSNet, and the Wire-Cell in-time/out-of-time algorithm. These labels are used to partition the spacepoints before clustering, helping to reduce overclustering where spacepoints from two different particle trajectories are included into one cluster. For example, the LArMatch track and shower start keypoints are used to temporarily remove nearby spacepoints in order to prevent spacepoints from particles coming out of a common interaction vertex from being grouped together.

After the subclustering step, non-ML algorithms are then used to combine the subclusters to form sets of spacepoints intended to represent the ionization produced by a single particle. These algorithms make use of the LArMatch ouputs for track endpoints and shower starts to seed the particle-building algorithms.

After this clustering stage is complete, the reconstruction has formed candidate particle trajectories. The following stage forms neutrino interaction candidates by associating primary particle trajectories to interaction vertices. Secondary trajectories are also associated to interactions by looking for trajectories that seem to emerge from previously included trajectories. Cosmic muon trajectories are also formed by using track start and end keypoints to seed the track-building algorithm applied to only out-of-time subclusters. Neutrino and cosmic muon candidates are the core outputs of the 3D reconstruction workflow.

For the individual neutrino interactions, further analyses are performed. Another CNN, referred to as "LArPID", assigns particle identification scores to individual particle trajectories. This LArPID network acts on two sets of images for a given individual particle cluster. The first set of images are sub-images formed by cropping around the cluster's projected position on each wire plane image. These images include values for only those pixels at the projected locations of spacepoints. The second set of images provided to LArPID is a set of "context" images which use the same cropped location but include more pixels, only masking out pixels with an out-of-time tag (those likely not produced by interactions associated with the beam). The purpose is to provide LArPID with both a given cluster's pixels and information pertaining to the entire interaction. We believe (see section III5) that the context images are critical in maximizing the particle ID accuracy of LArPID. The context images provide information the network can use to better ID the cluster. The context images also provide the means to overcome clustering errors from the 3D spacepoint algorithms by providing information that might have been lost during clustering but is still present in the images around the location of the clusters. The primary output of LArPID is particle class scores for five particles: muon, charged pion, proton, electron, and photon. Particles and their anti-particles are combined into the same class. LArPID also provides auxiliary outputs in order to provide the option to make data selection cuts based on estimates of the cluster reconstruction quality and as to whether the particle in question is a primary particle emerging from a neutrino interaction vertex or a secondary particle descended from the interactions of the primary particles.

The final outputs provided by the reconstruction are collections of candidate neutrino interactions and cosmic muons. For each neutrino candidate, each prong (reconstructed track or shower cluster) is provided a particle ID score from LArPID. Using this network's ID, the energy and 3-momentum are estimated for each particle. The kinematics estimator for muons, protons, and charged pions is based on the visible tracklength and uses the relationship between particle energy and the estimated length of fully ranged-out particles.[1] The energy estimator for the electromagnetic showers uses calorimetry based on charge. Additional network outputs related to keypoint scores and LArPID estimates are also passed along as outputs. The information for candidate neutrino interactions and their constituents can then be used to develop physics analyses.

In the rest of this section, we provide more details for a subset of the components discussed above. We do not include discussions of the image pre-processing algorithms, the in-time/out-of-time Wire-Cell tagger, and the SSNet CNN since their details can be found in the indicated references. For each component described, we focus on outlining the core approach of the algorithms, reference previous related work, and document key heuristics in tuning their behaviors.

1. Detector coordinate system and Data set terminology

We will often visualize the outputs of the reconstruction or define performance metrics assuming a specific 3D coordinate system. For the basis vectors, the positive $\hat{x}$-direction runs in the direction of the anode to the cathode and points in the direction opposite to the drift of ionization electrons towards the anode. The positive $\hat{z}$-direction runs in the same direction of the neutrino beam. The positive $\hat{y}$-direction points upward to the sky. The origin of the coordinate system is defined at the boundary of the TPC where $z = 0$ is the side closest to the source of the beam, i.e. upstream, $x = 0$ is at the induction plane closest to the drift volume, and $y = 0$ is located at the midpoint of the vertical TPC dimension. The MicroBooNE TPC is a rectangle whose lengths are (256 cm, 233 cm, and 1036 cm) along the (x, y, z) axes, respectively.

Another important definition is what constitutes an "event". The values specified here are particular to the MicroBooNE experiment. However, the overall data schema will be similar for other LArTPCs utilizing sense-wires. Each event includes a set of waveforms from each of the three wire planes that are arranged in a 2D array to make three wire plane images, which we will refer to as "TPC images" or simple "images." The three wire planes of the MicroBooNE detector – from closest to the TPC drift region to the furthest – are

[1] This estimate is applied to tracks regardless of whether they range out inside or exit the detector. A more accurate estimate for exiting tracks will be the subject of future work.

the first induction plane, the second induction plane, and the collection plane. They are so-named by the process with which ionization produces a current signal within the sense wires. The three wire planes are given a label, 'U', 'V', and 'Y', respectively. Every event will have exactly one TPC image from each of the three planes.

Each of the waveforms that makes up the images in an event consists of a time series of 9600 voltage measurements, or samples, recorded every 0.5 microseconds. The primary DAQ system for the MicroBooNE detector must be externally triggered (in other words instructed) to capture a synchronous set of waveforms for all channels. The two trigger types relevant for this work include (1) a signal synchronized with a firing of the neutrino beam, typically referred to as a "spill" (which references the release of a bunch of protons from the accelerator into a carbon target), and (2) a trigger signal produced by a signal generator programmed to fire at regular intervals in a time window between beam spills. The data recorded using the latter, non-beam spill, triggers are referred to as the "externally triggered" or EXT data set. The MicroBooNE detector records waveforms in sync with the Booster Neutrino Beam (BNB) produced by Fermi National Laboratory, and the data set recorded in coincident with the firing of this beam is referred to as the "BNB" data. Later in the sections demonstrating the performance of the reconstruction workflow through its use in a neutrino event selection, only data from the BNB and EXT data sets are used.

B. LArMatch: 3D spacepoints and keypoints generation

The purpose of the LArMatch network, illustrated in figure 2, is to use the TPC wire plane images to infer information related to the 3D location of ionization made by charged particle trajectories. Inferring the true location of such ionization is not trivial as this essentially requires inverting a tomographic projection, which by its nature will be an under-specified problem due to the information lost during the projection operation. To make this difficulty more concrete, we can consider trying to infer the location of energy depositions coming from a uniform line of ionization where the line is parallel to the wire readout planes. Figure 3 provides an illustration showing the signal that would be seen in the wire plane images, which is simply a line of uniform intensity across some set of wires all occurring at the same time (i.e. region of TPC samples). A naive approach would be to ask "what is the set of spacepoints that is consistent with producing a wire signal in all three planes?". This

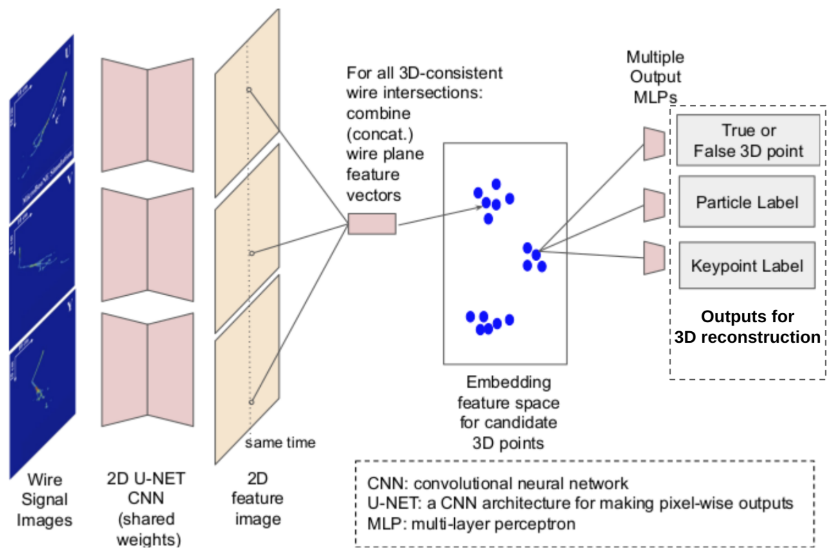


Figure 2: LArMatch network schematic. First, a U-Net CNN with residual convolutions takes as input three TPC images. For each input pixel, the CNN outputs a (16-dimensional) vector whose purpose is to represent the relevant patterns around a given pixel. Next, a non-ML algorithm proposes candidate spacepoints by forming all possible locations consistent with the charge deposition pattern in the images. The location of each proposed spacepoint is projected into the wire plane images in order to associate it a pixel from each wire plane. A (48-dimensional) feature vector for each spacepoint is made by concatenating the feature vectors belonging to the associated pixels. Three sets of multi-layer perceptrons (MLPs) then map the spacepoint vector to three types of outputs. One output is the score determining if a proposed spacepoint is located where a true energy deposition occurred. The second is a score for five particle types. The other output is a score indicating the location of several types of keypoints.

defines a 2D region of possible spacepoints, indicated by the purple region in the bottom illustration of Figure 3. The true locations of ionization would occur along a line within this region, indicated by the dashed line in the figure. One can select a subset of spacepoints in this region by utilizing some physical priors. If one assumes prior knowledge that (1) the true trajectory comes from a line segment and (2) the ends of the line must be consistent across the planes, then the set of possible spacepoints reduces to the correct region around the true path of ionization, as indicated by the yellow regions in the illustration of Figure 3. Another important refinement is to enforce some consistency in the signal intensity between the planes. In our example, one can use what in principle should be differences between the planes for the intensity per wire due to the different projected lengths of the ionization path onto the region around each wire. One can also impose a regularizing constraint such as biasing towards solutions that minimizes the number of spacepoints, which in this example

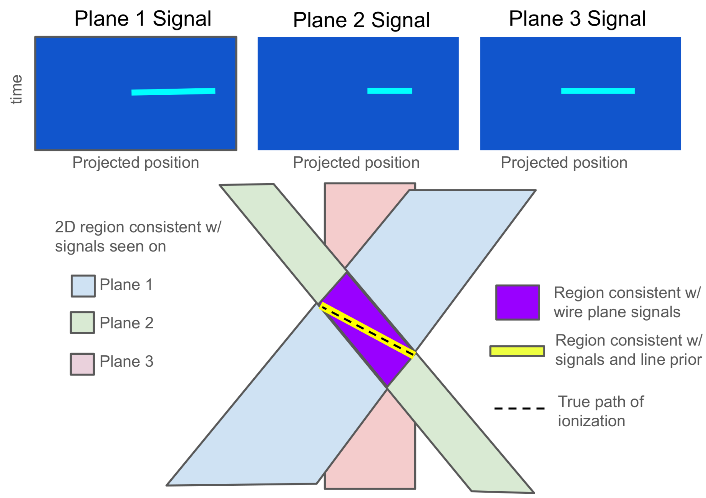


Figure 3: Example illustration of how inferring the 3D location of ionization can be an under-specified problem. We assume a line of uniformly distributed ionization occurs within the TPC and that the line is parallel to the wire planes. In this case, the wire plane images (top row) will contain a line of signal (cyan) occurring at the same time. Using only the knowledge of which wires on the planes see a signal, there is a 2D region in the TPC of possible spacepoints that are consistent with the wire plane signals (shown in purple). Only by also assuming a line shape and testing for consistency of the line length across planes can one determine the true region of ionization (shown in yellow) that corresponds to the true path.

can be seen to have a similar effect to having a line-like prior. These two latter approaches are a core part of the approach employed by the Wire-Cell reconstruction framework [18]. What this example is meant to illustrate is the type of prior information or strategies needed to pick out the true points of ionization. While this example is the worst-case scenario for simple line trajectories, for LArTPC wire planes on the surface or in regions with many particles emerging from a neutrino interaction other degeneracies will arise.

The motivation for the LArMatch network is to complement charge consistency and regularization by using machine learning to find additional features to match across the planes which improves the identification of true ionization points. One can imagine separating true 3D energy depositions from false ones by learning to “match” local energy deposition patterns in one plane to another. These patterns must follow coherently from the underlying 3D patterns of ionization. The algorithm thus proceeds in two steps. The first is to use a simple, deterministic algorithm to propose a large set of possible 3-wire intersections that might correspond to the location of real energy depositions in the detector. Next, a convolutional neural network (CNN) is trained to identify which 3-wire intersections are real or false. We use what is known as a 'U-Net' for the CNN architecture. A U-Net maps the input images to a set of latent vectors at each pixel whose purpose is to summarize the relevant information in the neighborhood of the pixel. For a given 3D spacepoint, we project its location onto each wire plane and associate a pixel from each. We then concatenate the feature vectors from each pixel and pass them to a multi-layer perceptron which outputs a score indicating if the spacepoint is real or false. The feature vectors are also used by other MLP heads to produce additional information. In total, each feature vector is mapped to three outputs: (1) a score indicating if the spacepoint is true or false, (2) a set of scores classifying the spacepoint as one of five particle types, and (3) a set of scores related to how far away the point is from five types of keypoints. In the following sections, we first describe the algorithm that produces the spacepoint proposals from the wire plane images. We then provide details on the image-to-feature vector U-Net. Finally, we discuss the three different output heads.

1. Proposal of 3-wire intersections, or "triplets"

The first step to the LArMatch approach is to generate spacepoint candidates simply based on minimal geometric plausibility. Initial spacepoints represent the location of 3-wire intersections for wires with an above threshold signal coincident in time. We represent these wire combinations as a "triplet" of integers whose components contain the index associated with wires from each of the three wire planes. When the wire plane data is represented as an image, the triplet refers to the tuple of column indices for the three wire plane images. In order to not miss spacepoints that project onto non-responsive wires, the wire combinations can include one wire which has been tagged as non-responsive. About 10% of the sense wires in the MicroBooNE detector are classified as non-responsive. We do not try to make up for missing spacepoints due to below threshold wire signals caused by ionization patterns that cause destructive interference on the induction wires. These are associated with ionization patterns where local segments are perpendicular to the wire planes. False positive and false negative errors can also be induced by the presence of noise features on the wires.

We form a set of candidate triplets for each time tick (represented by a row in the 2D wire-plane images) in the wire plane signals. The three column indices and row index specify the projected pixel locations in the three wire planes. This information also specifies a 3D location determined by (1) the location of the 3-wire intersection and (2) knowledge of which image row represents the time coincident with the beam trigger combined with a drift velocity assuming a perfectly uniform drift field.

The proposed spacepoints for an example simulated MicroBooNE event are shown in Figure 4. Metadata, which captures the "truth" about the particle trajectories present in each event, is saved during the simulation and used to create the ground truth labels for the LArMatch network. This includes both a list of charged particle trajectories passing through the TPC and the location of energy deposited by the particles. To save disk space, much of this information is projected into an 2D array with the same dimensions as the wire plane images, thereby facilitating the ability to determine the particle type or individual trajectory ID that deposited the most ionization observed at a given 2D pixel. In the simulation, the locations where energy was deposited by a particle is stored. For each pixel in the wire plane image, we assign to it the largest energy deposition that contributed to the value in the pixel. We then project this position into the other wire planes. The pixels on the other wire planes then are used to calculate the shift in the number of columns between the pixels on the two wire planes. In order to recover the YZ location of the largest energy deposit cluster that contributed to the pixel in the starting plane, one calculates the 2D intersection of the two wires from the different planes. The distance of the energy deposit from the wire plane can be calculated from the time relative to the event trigger and the drift velocity. Given that the wire planes are a tomographic projection of the 3D space points, this method of saving the 3D locations does not allow for perfect inference. However, we find the accuracy is sufficient to construct the ground truth for the LArMatch network, while reducing the amount of data to be saved. For future work, it would be worth exploring a better method of compressing the 3D energy deposition information that is not inherently lossy.

In figure 4, the proposed spacepoints for one event are shown along with the ground truth 'true' or 'ghost' labels built from the simulation metadata. Spacepoints near a true location of ionization are given the 'true" ground truth label shown in red. The rest of the spacepoint proposals are given the 'false" ground truth label shown in blue. We highlight two regions of this figure. The first is the volume between z=[600 cm,800 cm]. Here, many short line-like regions of false spacepoints are seen. This is due to a fairly large region of unresponsive wires on one of the wire planes (the "Y" collection plane). In these regions, the requirement to propose a spacepoint is relaxed from requiring ionization to be observed on

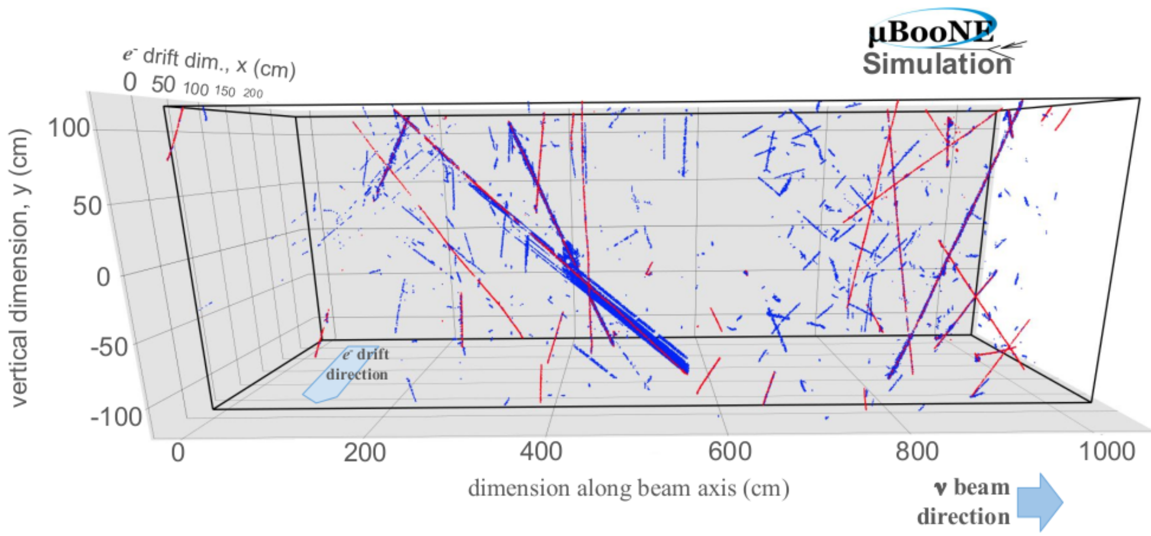


Figure 4: Initial spacepoint candidates for an example simulated MicroBooNE event. The "true" spacepoints located near regions of ionization are colored in red. The "false" or "ghost" spacepoints that are not near ionization are colored in blue.

only two (as opposed to all three) wire plane images. We make this accommodation with the aim of minimizing the amount of missing ionization at the cost of potentially accepting more false positives. The idea will be to use a downstream algorithm, specifically a particle-level CNN, to correct for false spacepoints or clustering mistake. The other region to point out in Figure 4 is the relatively large region of false (i.e. blue points) surrounding one muon track between z=[400 cm, 600 cm]. This is where a portion of a cosmic muon is parallel to the wire planes, similar to the illustration discussed earlier.

2. *Feature Generating U-Net*

The core part of the LArMatch network is the U-Net [23] CNN mapping images to pixel-wise feature vectors. We use residual convolutions [24] over standard convolutions. There are a total of six convolutional layers including five downsampling layers, each time with stride two. When normalization layers are used, we use instance normalization [25]. We upsample on the decoder part of the network with convolution-transpose operations. Because the wire plane images are sparse, i.e. most pixels have a value very nearly zero, the network uses sparse-submanifold convolutions [26] as implemented by the MinkowskiEngine library [27]. The U-Net takes in a sparse tensor representation of a single wire plane image at a time and is applied to each wire plane, separately. In other words, the input of the U-Net is a sparse tensor, $\mathbf{s}$, which consists of a list of N pixels, which is a subset of all the pixels in the image. To be included into the sparse tensor, a pixel had to pass two criteria. One was that the pixel had a value above some threshold value. We used a threshold of ≥ 10.0, or what is about a quarter of the average pixel value for minimum ionization sections from cosmic muon tracks. The second criterion is that a pixel, with a below threshold value, was from a non-responsive wire while also being the projected location of a proposed spacepoint. The sparse image tensor, $\mathbf{s}$, is represented through a pair of tensors. The first is a coordinate tensor, $\mathbf{c} \in \mathbb{W}^{N\times 2}$, which contains the indices of the above threshold pixels. The second is a feature tensor, $\mathbf{f} \in \mathbb{R}^{N\times 1}$, containing the associated pixel values. The U-Net, therefore, maps $\mathbf{s} = (\mathbf{c}, \mathbf{f})$ to N 16-dimensional feature vectors, $\mathbf{v} \in \mathbb{R}^{N\times 16}$.

3. Spacepoint real/ghost classifier

A 2-layer MLP is used to classify each proposed spacepoint as either real or ghost. This classifier takes in the concatenated 48-dim feature vector, $\vec{v}$, formed from the individual 16-d feature vectors from the project pixels from each plane. The MLP has two hidden layers, each with 32-features, and outputs both a real and ghost class score. A softmax function normalizes the sum of these scores to 1.0. We use the normalized score for being a true spacepoint, $p(\vec{v})$, for classifying proposed spacepoints.

We train the network to optimize this prediction using a focal loss [28] objective. We also weight each spacepoint based on the relative total number of ground truth-labeled real and ghost points. This weighting is used to mitigate bias that might favor true negative predictions coming from the higher frequency of ghost spacepoints compared to real spacepoints. Our training objective is

$$\begin{aligned} \min_{\theta} \mathcal{L}_{ghost} = \min_{\theta} \Bigg[\sum_{b}^{N_b} \Bigg[& \sum_{i_t}^{N_{i_t,b}} w_{b,t} \log(p_\theta(v_{i,t}))(1 - p_\theta(v_{i,t}))^{\gamma} + \\ & \sum_{i_f}^{N_{i_f,b}} w_{b,f} \log(1 - p_\theta(v_{i,f}))(p_\theta(v_{i,f}))^{\gamma} \Bigg] \Bigg]. \end{aligned} \tag{1}$$

We optimize the objective using AdamW, an implementation of stochastic gradient descent with features such as adaptive gradient normalization and momentum. We train our models with randomly sampled batches of data. The parameters of both the UNet and the MLP heads producing the LArMatch outputs are learned simultaneously in one combined training procedure. Within a batch of N_b training samples, the b-th sample consists of N_b feature vectors, v_i produced by the UNet for all N_b candidate spacepoints. We use the simulation meta-data to produce ground truth labels for the set of vectors, v_i. The subset of $N_{b,t}$ 'real' or 'true' ground truth-labeled vectors is $v_{i,t}$; the subset of $N_{b,f}$ 'ghost' or 'false' labeled vectors is $v_{i,f}$ (with $N_b = N_{b,t} + N_{b,f}$). The likelihood estimate for being a real point is $p_\theta(v_i)$ and is approximated by the output of the true/ghost MLP, parameterized by θ. The $(1 - p_\theta(v_{i,t}))^\gamma$ and $(p_\theta(v_{i,f}))^\gamma$ are the focal-loss factors. As $p_\theta(v)$ approaches the ground truth value (1.0 for real points, 0.0 for ghost points), the focal loss factors increasingly down-weight these examples with the γ meta-parameter controlling how quickly the downweighting occurs with increased confidence. Conversely, the spacepoints whose classification is incorrect will contribute more to the update of the model parameters. In effect, the focal loss is intended to nudge the optimization towards improving "harder" examples over increasing the confidence for easy examples.

4. Spacepoint Keypoint score

The LArMatch network also is tasked with providing the outputs to identify locations of ionization that can be useful for later 3D trajectory reconstruction. We defined six classes of "keypoints": (1) a neutrino interaction vertex, (2) the start of a track-like trajectory (defined as belonging to a muon, proton, charged pion, and other heavy mesons), (3) the end of a track-like trajectory, (4) the start of EM shower not produced by processes in the following types, (5) the start of EM showers produced by the decay of a muon, and (6) the start of EM showers form delta rays (typically radiating from energetic muon tracks).

The way the location of possible keypoints is represented in the output of the network is through a score made for each spacepoint. The score ranges from 0 to 1.0, with scores inversely proportional to the distance to a keypoint. The network is trained to reproduce a score distribution that follows a Gaussian with a uniform, uncorrelated variance. In other words, the network is asked to produce a heat map near keypoints with the hotspots having a set shape. A post-processing step can then be used to identify hotspots and use the spatial score distribution to fit to the precise keypoint location. The ground truth-score is calculated using meta-data from the simulation which retains the creation point of charged track-like particles and the earliest location of ionization within the TPC of an EM shower. All particles whose meta-data information was used to define EM shower keypoints were required to have at least one wireplane image with 20 or more pixels whose signal was attributed to its ionization.

A 2-layer MLP, s_θ is used to map each spacepoint's feature vector, $\vec{v}$, to a vector, $\vec{k} \in \mathcal{R}^6$, whose components are the scores of each keypoint class. The value of each component is independently kept within the range of $[0, 1]$ by applying a sigmoid-function element-wise. This bounded output is then compared to the ground truth scores for each keypoint class.

Both the dedicated keypoint MLP and the UNet parameters are optimized to minimize the keypoint training objective, $\mathcal{L}_{keypoint}$, given by

$$\begin{aligned} \mathcal{L}_{keypoint} = \frac{1}{N_b} \sum_b^{N_b} \Bigg[\frac{1}{N_c} \sum_c^{N_c} \Bigg[& \sum_i^{N_{b,t}} w_{b,c,t} (\hat{s}_{i,c,b,t} - s_\theta(\vec{v}_{i,b,t}))^2 + \\ & \sum_j^{N_{b,f}} w_{b,c,f} (\hat{s}_{j,c,b,f} - s_\theta(\vec{v}_{j,b,f}))^2 \Bigg] \Bigg] . \end{aligned} \tag{2}$$

For the above equation, the sum over N_b is over the number of examples in each training batch. Each example consists of proposals from one set of wire plane images from one TPC readout event. The sum over N_c is over the six different keypoint classes. Because most spacepoint proposals are unlikely to be near a true keypoint, to aid training, we use weights to balance the contribution of examples near true keypoints, for which the MLP needs to output a score, and those far away from true keypoints, for which the MLP only needs to output zero. Therefore, for each class we split the total number of spacepoint proposals in the b-th example, N_b, into "true example" points within 10 cm of a true keypoint and "false example" points which are not. Thus, in the above equation, the sum over $N_{b,c,t}$ is for the true example spacepoints for class c, while the sum over $N_{b,c,f}$ for the false examples for class c within the b-th example of the batch. Regardless of the class, the number of true and false examples total to the same number of spacepoint proposals, i.e. $N_{b,c,t} + N_{b,c,f}$. The true example weight for class c, $w_{b,c,t}$, is set to the ratio of the total number of spacepoints, $N_{b,c,t} + N_{b,c,f}$, over the number of true examples in the b-th training example. Similarly, the weight, $w_{b,c,t}$, is the ratio of the total number of spacepoints over the number of false examples in the b-th training example. For each true (false) spacepoint, the contribution to the loss is the weighted squared-difference between the keypoint MLP output, $s_\theta(\vec{v}_{i,b,t})$ ($s_\theta(\vec{v}_{j,b,f})$, which is a function of the feature vector, $\vec{v}_{i,b,t}$ ($\vec{v}_{j,b,f}$), of the i-th (j-th) spacepoint in the b-th training example. The ground truth score for a given spacepoint is labeled by $\hat{s}_{i,c,b,t}$ and $\hat{s}_{i,c,b,f}$.

5. *Weighting the Multi-objective Loss*

We use a dynamic weighting of the different task objectives when forming the final, overall loss function. This technique changes the relative weights of the tasks based on an estimate of the uncertainty. This method in effect aims to encourage parity in the contribution of the terms to the total loss throughout the training period. In our application, the total loss is

$$\mathcal{L}_{larmatch} = e^{-s_{ghost}}\mathcal{L}_{ghost} + e^{-s_{keypoint}}\mathcal{L}_{keypoint} + s_{ghost} + s_{keypoint}. \tag{3}$$

C. 3D Particle Trajectory Reconstruction

The 3D reconstruction of individual particle trajectories is designed around the outputs produced by the LArMatch CNN, the SSNet CNN, and the Wire-Cell out-of-time tagger. The fundamental input to the reconstruction is the set of spacepoints produced by the LArMatch stage. The algorithms described below first create clusters belonging to individual particles. This is followed by building a representation of the trajectory. A line segment, fit to a cluster's spacepoints, is used to represent track-like particles. A cone, whose axis is fit along an initial path of spacepoints, represents shower-inducing particles. These outputs are more easily achieved by the pattern recognition performed by the previous CNNs, alleviating the need to find the necessary patterns within the set of spacepoints, directly.

The upstream outputs are first used to refine and partition the candidate spacepoints. First, the LArMatch real/ghost score is used to filter ghost points. A 'real' spacepoint score threshold of 0.8 is applied to remove ghost points. This score value removes approximately 90% of ghost points and keeps approximately 75% of true points. The cut value chosen favors background rejection in order to keep the typical run time of downstream algorithms between

10-20 seconds per event. Next, the Wire-Cell out-of-time/in-time label splits the spacepoints into two sets: 'in-time' and 'cosmic'. Both of these sets are subdivided into track and shower hits using the scores from the 2D SSNet CNN. At this stage of the reconstruction, we have four buckets of spacepoints: in-time-track, in-time-shower, cosmic-track, and cosmic-shower.

Because of spacepoint proposal's very forgiving criteria, the density of points around the true trajectory can be high with many spacepoints providing redundant information. We apply a heuristic to remove points away from the core of the trajectory. For each plane, we only tag one spacepoint to keep per pixel, choosing the spacepoint with the largest LArMatch real/ghost score. The final set of spacepoints we keep are the union of all the spacepoints associated with pixels from each plane. This heuristic is applied to the in-time-track, in-time-shower, and cosmic-track spacepoint partitions. Figure 5 shows the fraction of space points that are within some distance of a true muon, charged pion, or proton trajectory within the TPC. The simulated data used to make the plot in this figure contained simulated cosmic particles (mostly muons) and neutrino interactions. About 90% of spacepoints are within 1 cm of track-like trajectories.

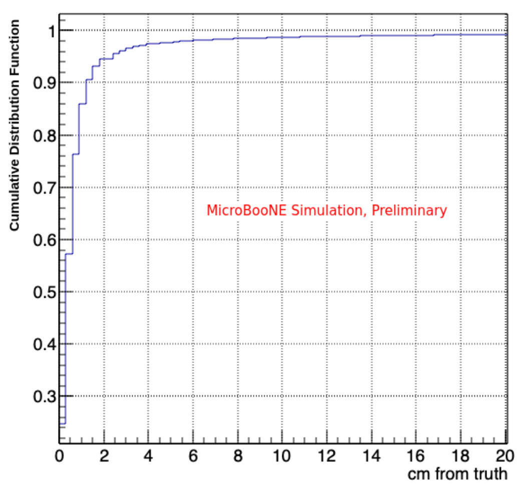


Figure 5: Fraction of spacepoints vs. distance from the ground truth trajectory of a true muon, charged pion, or proton trajectory within the TPC.

D. Keypoint Generation

The output of the LArMatch keypoint proposal CNN are pixel-wise scores. This information needs to be distilled into individual keypoint candidates. Ideally, the scores from the network should be arranged spatially as spherical hotspots. We separate the hotspots by first retaining keypoints with a minimum score and then using density-based clustering algorithm, DBSCAN, to identify individual candidates. The location of a keypoint is taken to be the spacepoint within a cluster with the highest keypoint score. Spacepoints part of a reconstructed keypoint are tagged and removed from the original pool. The scores of points within the remaining pool are modified by using the set of newly created keypoint locations to subtract the expected score contributions (defined by a Gaussian function). The modified score is clamped to be zero or greater. This keypoint-finding procedure is applied twice, first with a high keypoint score threshold and again with a lower score threshold. This procedure is applied separately for each type of keypoint. As a result, a spacepoint can be a part of keypoint clusters for multiple classes. However, a spacepoint can only be part of one keypoint cluster within the same class.

Keypoints from all six classes are searched for within the 'in-time' spacepoint partition. Only track-start and track-end types are constructed from the 'out-of-time' spacepoint partition. All of the above keypoints will be used by the next set of algorithms to seed the creation of particle clusters.

In the future, keypoints with the remaining classes can be built using the out-of-time spacepoints. This could be used to reconstruct cosmogenic particle clusters, beyond those for cosmic muons, with the intended use of creating side-band datasets. For example, it might be useful to reconstruct out-of-time neutron-induced interactions. These could be identified by track-start keypoints or regions with high neutrino-like scores as they often mimic NC-like final states.

E. Particle and Interaction Reconstruction with 3D Spacepoints

Two different sets of algorithms are used to form track-like and shower-like clusters. However, both take the approach of trying to form pure sub-clusters and then using heuristics to stitch together the subclusters belonging to individual particles.

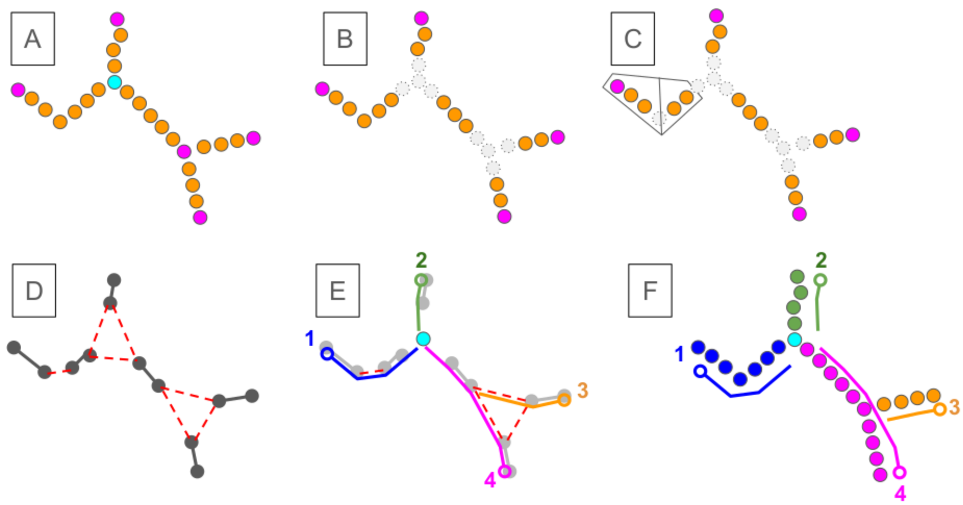


Figure 6: Schematics of the steps of the track reconstruction. (A) Spacepoints with track-like and in-time labels are collected along with start and end keypoints (here a start keypoint is shown in cyan; end keypoints are in magenta). (B) Spacepoints around certain keypoints are removed from clustering. (C) Spacepoints are clustered and then broken into straight pieces based on the convex hull around the points. (D) Clusters are represented as a line segment with two ends. Line segments between end points within a certain distance are formed. A graph is defined with nodes defined by the cluster end points and two types of edges defined by the line segments within charge clusters (black solid lines) and those between clusters (red dashed lines). (E) Track-start keypoints are used to seed a depth-first graph traversal algorithm that proposes possible tracks. (F) Spacepoints are assigned to nearby line-segments and together define a candidate particle track. In this schematic, four candidates are proposed.

1. Forming track candidates

For track-like particles, we form subclusters by looking for neighboring spacepoints arranged in a straight line. We start by using DBScan to form clusters of neighboring spacepoints. To avoid clustering spacepoints from multiple particles, we apply several heuristics. The first is that we remove spacepoints within a 3 cm radius of reconstructed keypoints. This length is three times the DBScan maximum distance parameter of 1 cm in order to ensure points from different particles emerging from a vertex or secondary interaction point will not be clustered together. (A cartoon of this step is shown in Figure 6B). This is done to separate spacepoints coming from the locations of neutrino interactions, secondary interactions, or decay. The other heuristic aims to find locations of intersecting trajectories. The approach is to recognize 'vee' patterns using the convex hull around the set of pixels corresponding to the projected spacepoint location on the wire planes. This reuses algorithms built for the one 1-lepton-1-proton exclusive analysis in the previous MicroBooNE low-energy excess search [29]. We use convex hull defect points to identify intersecting trajectories and the location to split the clusters assuming either an 'X'- of 'V'-shaped 3D spatial pattern. (A cartoon of this step is shown in Figure 6C).

We build individual particle tracks by chaining, end-to-end, the straight-line subclusters. We use a recursive depth-first graph traversal algorithm to build chains of subcluster segments. This algorithm begins by defining a graph whose nodes are the collection of line segment endpoints. Two types of edges are defined between the nodes. The first edge type (A) connects endpoint nodes belonging to the same subcluster. The second edge type (B) connects endpoint nodes below some max distance. (A cartoon of this step is shown in Figure 6D with the A-type edges represented as solid black line segments and B-type edges as dashed red line segments.) Multiple B-type edges between the endpoints of two line segments can be formed.

Once the graph is formed using all the subcluster line segments, a recursive graph traversal algorithm is used to build chains of line segments representing candidate track trajectories. The sub-set of endpoint nodes sufficiently close to designated keypoints serve as starting points. (A cartoon of this step is shown in Figure 6E.) Using depth-first recursion, a tree-structured subgraph for each seed node is built by traversing edges of alternating type, starting with A-type edges. Heuristics based on angles and distances between edges and segments were used to choose and prioritize B-type edges to include in the tree. For potential B-type edges less than 3 cm, the cosine between the track segments that this B-edge would connect is required to be greater than zero. For edge lengths between 3 to 10 cm, the cosine between track segments must be greater than 0.7. For lengths longer than 10 cm, the cosine must be greater than 0.9. We allow for such fairly large distances between track segments in order to cross regions of unresponsive wires that occur within the detector. These values were optimized to maximize the completeness of simulated cosmic muon tracks. To prevent loops, nodes can only be visited once.

Paths within a tree subgraph that connect the root node to the leaf nodes represent candidate trajectories. Possible paths are selected using the heuristic that true trajectories run along regions of ionization. This is quantified by the fraction of the trajectory length that projects onto locations within the wire plane images that are near pixels with sufficient ionization. Paths satisfying this criterion are scored based on weighted sum of the total path length, the trajectory fraction near ionization, whether the end of the path coincides with a track-end keypoint, and a measure of the overall trajectory straightness. Valid paths are then sorted by descending score.

Candidate track particles are created from the set of paths, starting with the highest scoring. The ordered collection of line segments define the trajectory of the track. Track-like spacepoints close to the line segments form a cluster associated with the trajectory. Any remaining paths that fork from the current highest-scoring path are used to define secondary tracks that begin at the last common node. Once a track trajectory is created, any segments included in the path are forbidden to be reused. Any remaining path is removed if it includes any A-type edges corresponding to segments included in a track. Track creation continues with the remaining highest-scoring path and completes once the set of valid paths is empty.

Candidate tracks are created in this way for all starting nodes. An individual subcluster line segment can be a part of multiple track candidates, as long as the tracks were created using different starting nodes. Once all candidate tracks have been formed for a given pool of track subclusters, refinements to the line segment trajectory are made for each track candidate. The refinements consist of creating new points along the segmented line such that a chosen maximum distance occurs between points. The locations of the expanded set of points are iteratively perturbed using gradient descent in order to minimize squared-distance between projected wireplane positions and pixels with ionization. Once refinements of the line-segment trajectories are completed, 3D spacepoints are associated with each trajectory. First, spacepoints in the sub-clusters used to make the trajectory are added. Second, spacepoints close to keypoints, which were vetoed initially, are added to the trajectory. The collection of spacepoints along with the line segment trajectory represent the candidate particle tracks. (The final track candidates in the diagram of Figure 6, defined with both a line segment trajectory and associated spacepoints, are shown in sub-figure F.)

2. Forming shower candidates

The approach taken for forming shower candidates is to assume that shower keypoints are located at the start of a shower and then to collect 3D spacepoints, tagged as shower-like by SSNet, belonging to the shower. Points are added to a candidate shower if they fall within a cylindrical region around the shower's reconstructed direction. To determine the shower

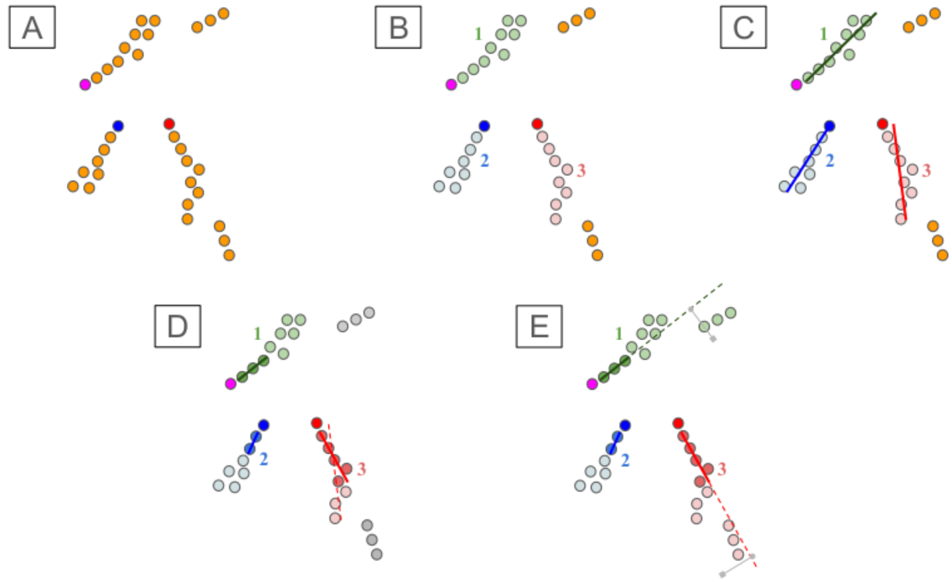


Figure 7: Schematic illustrating steps of the shower reconstruction. (A) Spacepoints with shower-like and in-time labels are collected along with shower keypoints. The shower points are colored orange. Shower keypoints are given random colors. (B) DBscan is used to cluster the points based on neighbor distances. Found clusters are assigned a random color and index. There are some points shown which did not have enough points to form a cluster. (C) The first principal component direction and keypoint define the initial direction of the shower. (D) A subset of points within each cluster near the keypoint is used to define a shower trunk. (E) The trunk direction and keypoint defines a line. Clusters close to this line are added to the shower candidate, defined by a keypoint, shower trunk, and cluster of shower spacepoints.

direction, a shower "trunk" is defined as a line that running along the first 3-10 centimeters of ionization at the start of the shower.

The shower reconstruction begins by gathering, as input, spacepoints that have been filtered to have (1) an SSNet shower score greater than 0.5, (2) a LArMatch 'true' score of 0.8 (the same threshold for accepting spacepoint candidates), and (3) to be associated with pixels that have not been tagged as being out-of-time with the beam, (in other words does not have a cosmic tag). The algorithm also requires a set of shower keypoints. In figure 7a, spacepoints satisfying the conditions are shown as orange circles. Different candidate shower keypoints are also shown in figure 7a in random colors, here blue, red, and magenta.

Next, DBscan is used to build shower subclusters and is run with the following parameters: a maximum distance of 5.0 cm, minimum cluster size of 20, and max nearest neighbors of 20. The somewhat higher minimum points to form a cluster is meant to reduce the number of small disconnected shower fragments. In the illustration in figure 7b, three clusters are found are given a random color and index. Note the smaller fragments which do not form a cluster. The first principal component (PC) is then calculated for each cluster. To remove very short clusters, any cluster whose first PC length – measured by the maximum distance between pairs of points projected onto the first PC line – is shorter than 1 cm is rejected. In the illustration in figure 7c, examples of PC axes for each cluster is shown.

Next, the shower keypoints are used to find a good shower "trunk." For all keypoints within 10 cm of the axis-aligned bounding box of each shower cluster, a subset of cluster points within a certain radius of the keypoint are collected. For each subset of points, a first PC axis is found and defines the axis of the shower trunk. A subset of points is made three times for each keypoint and cluster pair, using a radius of 3 cm, 5 cm, and 10 cm. In all cases, the keypoint is required to be at most 1 cm from the nearest cluster spacepoint in order to be able to define a valid shower trunk. In studies of the dE/dx along the true initial direction of simulated electron showers, the energy loss per unit length was most separable between electrons and photons between 1 cm to 3 cm. The upper bound was interpreted to be approximately the length scale where some aspect of the shower's cascade has begun, causing a widely varying dE/dx that differs from just ionization. Of the three candidate trunks, we choose the one to represent the cluster's trunk based on which is the most line-like. This is quantified using the ratio between the second-to-first principal component of the cluster's spacepoints.

At this point in the shower reconstruction, each shower candidate includes the cluster points, a shower trunk, and the seeding shower keypoint. Figure 7d provides an example where a shower trunk has been found for each of the three clusters in the illustration. Note that a subset of points, marked with a darker color, are tagged as being part of the shower trunk, can fall within different radii of their respective keypoints.

Finally, we attempt to add additional shower points to each candidate shower cluster. We do not add individual points, but instead test to see if we should add an entire cluster's points to a shower candidate. For each shower candidate, we loop through all shower clusters and ask which fraction of the cluster's points are within a volume around the line defined by the shower trunk's direction and the position of the keypoint. The acceptance volume is split into two regions. The first volume is for points in the "forward direction" of the shower, which must be (1) within 5 cm of the trunk axis while (2) it's projected distance along the axis is less than 50 cm from the keypoint. The second volume is for points in the "backward direction" of the shower. These points must have as projected distance within 3 cm of the keypoint and be within 3 cm of the trunk axis. This second volume is meant to help with cases where the shower keypoint is reconstructed a few centimeters into the trunk. If half of the points within the cluster falls within the volume defined by the shower candidate trunk, all of the spacepoints are added to the shower candidate. In our illustrative examples, figure 7e depicts successful tests to add a cluster of points to shower candidates 1 and 3 as each have a cluster whose points are within some distance of the trunk's line.

Note that the shower cluster merging condition at this point of the reconstruction is relatively restrictive. We later use the neutrino vertex to help merge shower cluster candidates into bigger, more complete shower clusters. This cluster merging occurs when neutrino interaction candidates are being constructed. We start the process of building showers by tagging a subset of shower candidates found from the procedure above as "shower prong" candidates. These are intended to represent the beginning of a possibly larger shower emerging from a neutrino interaction. Prongs are first identified based on how well they point back to the neutrino vertex. To qualify as a prong, the shortest distance between the line defined by a shower cluster's trunk direction and the neutrino vertex (often described as the impact distance in other scattering contexts) is below 20 cm. The shower prongs that qualify are then sorted by those with the smallest distance to those with the largest. Then, beginning with the prong with the smallest distance, we then loop over all shower clusters and decide on merging each into the prong if a cluster falls within a cone defined by the prong. The cone is defined by an axis whose starting point and direction are defined by the prong's trunk. Because the trunk is a line segment, the endpoint of the trunk closest to the neutrino vertex is designated as the tip of the cone. The trunk line segment is used to define the ray of the cone such that the direction of the ray is away from the neutrino vertex. The opening angle of the prong cone is 30 degrees. This is the angle of a right triangle whose height is approximately 2 Moliere radii (9.04 cm in liquid argon) and whose base is 50 cm (or about 3.5 radiation lengths) [30]. We add other 'test' shower clusters to the prong by determining if the test cluster falls within the prong's cone. To determine this, we ask if the test cluster's trunk endpoint closest to the neutrino vertex is within the 30-degree opening angle. We iteratively define a new prong and associate available shower clusters to it, all the while tagging clusters as unavailable and skipping them if they were used to define a prong or were merged into a prong. In this way, we use the neutrino vertex to help sort the order of shower candidates and thus bias which ones should serve as the starting prongs on which to build.

F. Interaction Candidate Formation

The formation of neutrino interaction candidates begins with associating neutrino keypoint candidates with track subcluster segments and shower-trunk candidates. The track subclusters derive from the in-time-track spacepoints. The shower subclusters and trunks are made from the in-time-shower spacepoints. Both types of subclusters are added as 'primary prongs' based on the distance between (1) the vertex and the closest segment endpoint and (2) the neutrino keypoint and the prongs first principal component axis. The graph-based algorithm described above is used to build candidate tracks using only the neutrino keypoint as a seed. The cone-based procedure described above is used to construct shower candidates using the associated shower prongs. For any shower subclusters assigned to multiple shower candidates, we prevent over-counting of visible energy by forbidding subclusters from being added to multiple shower prongs in this context. The track and shower candidates created at this stage of interaction reconstruction are tagged as primary prongs. At this stage, we also correct for potential reconstruction errors due to small clusters of spacepoints being mislabeled as track-like by the SSNet CNN. One algorithm checks for such track-like clusters that might occur at the beginning of a shower. For each shower prong, a line between its start point and the vertex is defined. We then check for any track clusters with over 90% of its spacepoints within 3.5 cm of this line or within 2 cm of the shower's trunk, for which the latter is defined by a line segment up to 10 cm long. We also check for short track-like subclusters that lie deeper within a shower, beyond the trunk. When such track clusters are detected, the track cluster is removed and its spacepoints are added to the shower's spacepoint container.

With the set of primary particles defined, the next step of the reconstruction is to add secondary particles to the candidate neutrino interaction. The search for secondaries starts by finding track and shower subclusters whose starting points are within 2 cm of any of the associated primary tracks or whose first principal component forms a line that approaches within 2 cm. For each secondary track prong, the graph-based track builder is used to create track particles. Likewise, for each secondary shower prong, the cone-based shower building algorithm is used to construct a secondary shower particle. This secondary particle reconstruction continues to iterate until no additional track-like or shower-like subclusters are associated with the interaction.

The construction of neutrino candidates concludes with routines to estimate the energy of tracks using range under the assumption of a few particle species. The initial direction of the track is also estimated. Calorimetric estimates of the total ionization are made by summing the pixel values of individual wire plane images. This leads to three plane-specific estimates for each shower. An initial integral-to-energy conversion is applied. Details for these energy estimates are in Section II H.

In addition to neutrino interactions, the reconstruction also builds cosmic muon tracks using the graph algorithm seeded by track-start and track-end keypoints. The tracks are made using clusters coming from the cosmic-track spacepoints.

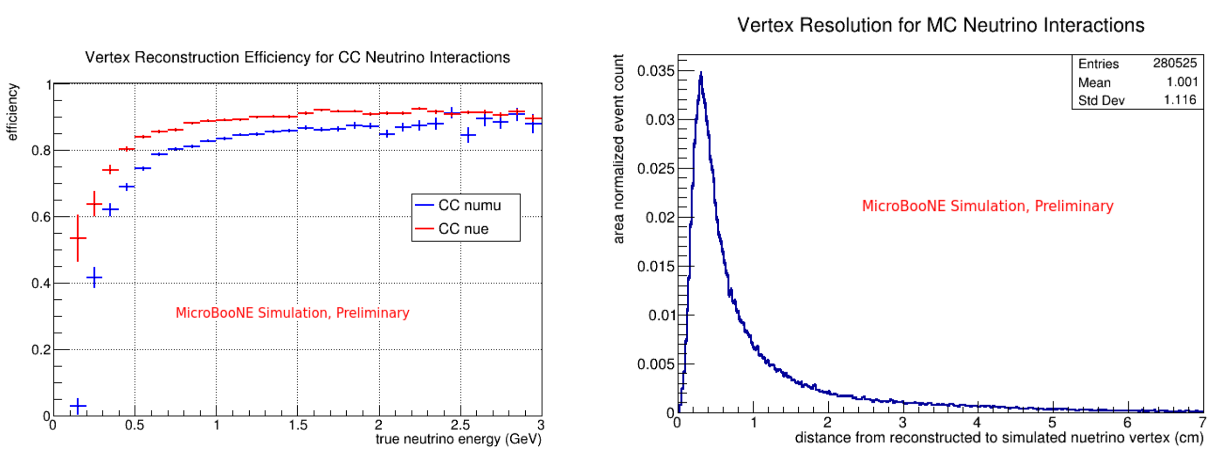
This interaction-building stage is the final step of the 3D spacepoint-based reconstruction. The neutrino interactions and their candidate primary and secondary particles are saved for analysis. The spacepoints and their associated scores are saved only for those assigned to a particle that is part of a neutrino candidate. These products will be used by the CNN-based particle ID to be described in later sections. The scores of the keypoints are also stored for selection and analysis purposes.

G. Reconstruction Validation

The algorithms described in the previous sections are able to efficiently reconstruct neutrino interactions and final state particles. This section presents validation plots exploring vertex and prong reconstruction quality using a sample of MC neutrino interactions (overlaid over cosmic-ray background data) occurring inside the MicroBooNE fiducial volume (defined as 3cm from the edge of the space-charge-corrected TPC boundary as in [17]). The ability of our reconstruction outputs to select CC ν_μ and CC ν_e events with high efficiency and purity is explored in sections III A and III B. These metrics are defined by the fraction of selected events relative to all charged-current events whose vertex occurs within the fiducial volume.

1. *Vertex Validation*

Figure 8a shows the efficiency of our neutrino vertex reconstruction as a function of simulated neutrino energy for CC ν_μ and CC ν_e interactions. Vertex efficiency is higher for CC ν_e events, but is high in both cases, rising above 80% by 0.5 - 0.6 GeV in neutrino energy and leveling off at around 85% - 90% above 1 GeV. Below 0.5 GeV, vertex reconstruction efficiency drops steeply (as expected), falling below 60% around 0.2 - 0.3 GeV. Figure 8b shows how accurate the vertex reconstruction is in cases where a candidate neutrino vertex is found. From this area normalized distribution of the distance between the reconstructed and true neutrino interaction vertex (for all MC neutrino interactions), we can see that the vast majority of reconstructed vertices are within 1cm of the true position. More specifically, 68% of reconstructed vertices are within 9.2mm of the simulated interaction position. The spacing between wires is 3mm, so we can reconstruct vertices within within about three wires, which is quite close to the one-wire-limit on the accuracy of a perfect reconstruction.



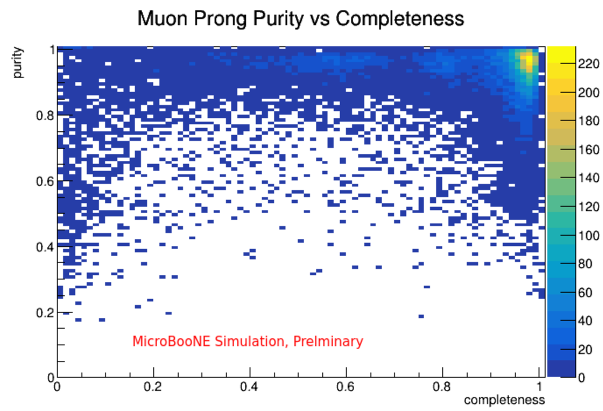
(a) (b)

Figure 8: (a): The fraction of MC ν_μCC and ν_eCC interactions occurring inside the MicroBooNE fiducial volume in which a neutrino candidate vertex was reconstructed (as a function of simulated neutrino energy). (b): The distance between the true and reconstructed neutrino vertex for all MC neutrino interactions inside the MicroBooNE fiducial volume.

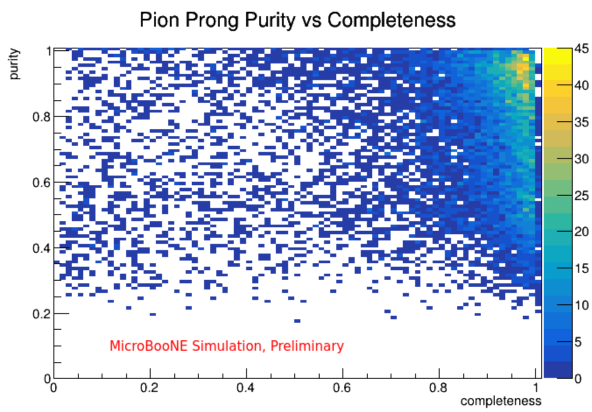
2. *Prong Validation*

To determine the quality of reconstructed prongs, we select a sample of tracks and showers attached to reconstructed neutrino vertices from MC neutrino interactions. Prongs are truth-matched to simulated particles from the interaction by projecting all of their spacepoints back into the 2D wire plane images and finding the simulated particle that deposits the most charge in associated 2D pixels. To allow for accurate truth-matching, we require that no more than 20% of the prong's 2D pixel charge come from the overlaid cosmic-ray background data. For each prong, we calculate the reconstruction quality metrics of purity and completeness, where purity is defined as the fraction of the prong's total 2D pixel charge that was produced by the truth-matched simulated particle, and completeness is defined as the fraction of the total 2D pixel charge deposited by the simulated particle that is included in the reconstructed prong.

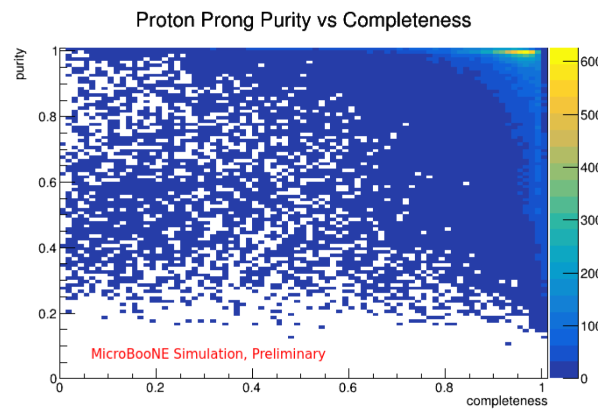
Figure 9 shows plots of purity vs. completeness for prongs that are truth-matched to simulated muons, charged pions, protons, electrons, and photons. The vast majority of reconstructed prongs occupy the high-purity, high-completeness upper-right corner of these plots, indicating a quality reconstruction. However, for protons and, to a greater extent, charged pions, there is a non-trivial population of prongs with high completeness but relatively low (roughly 40 - 70%) purity. This is largely due to the difficulty in separating out short tracks from the often dense clusters of charge surrounding the immediate vicinity of interaction vertices. Additionally, charged pions often decay, producing (through an intermediate muon) a small electron shower. In these cases, the electron shower and charged pion track are sometimes reconstructed as part of the same prong, contributing to the population of lower-purity pion prongs. There is also a non-trivial population of electron prongs with high purity, but low completeness. This is caused by either an early, spatially isolated branch of the electron shower getting reconstructed as a separate shower or by the pixels associated with a small stub of the trunk of the electron shower getting tagged as track pixels, causing that stub to get reconstructed as a separate track. However, in these cases, most of the electron shower is almost always reconstructed in a separate prong, and these occasional reconstruction issues can be overcome by identifying such prong fragments by estimating their purity and completeness with the LArPID network (discussed in section III).



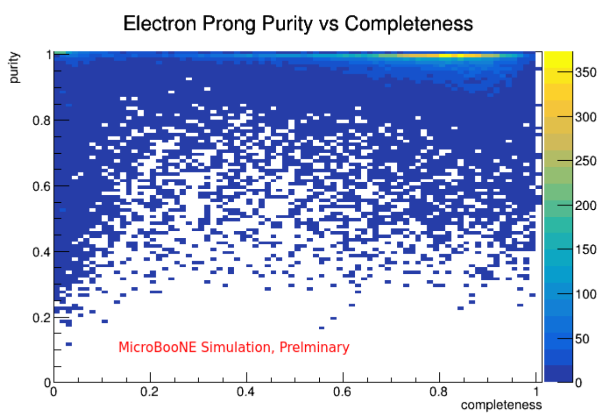
(a)



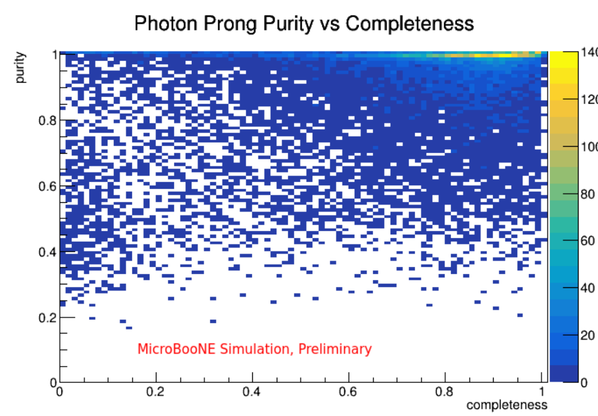
(b)



(c)



(d)



(e)

Figure 9: Purity vs. completeness for reconstructed prongs (from MC neutrino interactions occurring inside the MicroBooNE fiducial volume) that are attached to neutrino candidate vertices and truth-matched to simulated muons (a), charged pions (b), protons (c), electrons (d), or photons (e).

H. Energy Reconstruction

To reconstruct the energy of tracks, we first use the LArPID network discussed in section III to determine the particle type: tracks are classified as either muons, charged pions, or protons based on which of those three LArPID particle scores is highest. Knowing the particle type, we can then use the track length and fits to the kinetic energy vs. range curves of these three particles to determine a kinetic energy for the track. These fits are shown in figure 10 overlaid over a scatter plot of kinetic energy vs. length for simulated muon, charged pion, and proton trajectories (from MicroBooNE MC ν_μCC interactions) that begin and end at least 10cm from the edge of the MicroBooNE active volume. The fit for muons provides accurate results; however, charged pions and protons often interact before ranging out, causing a spread above the fit line for particles in which our range calculation underestimates energy. This range calculation will also of course underestimate the energy of track-like particles that exit the detector. These shortcomings will be addressed in future studies.

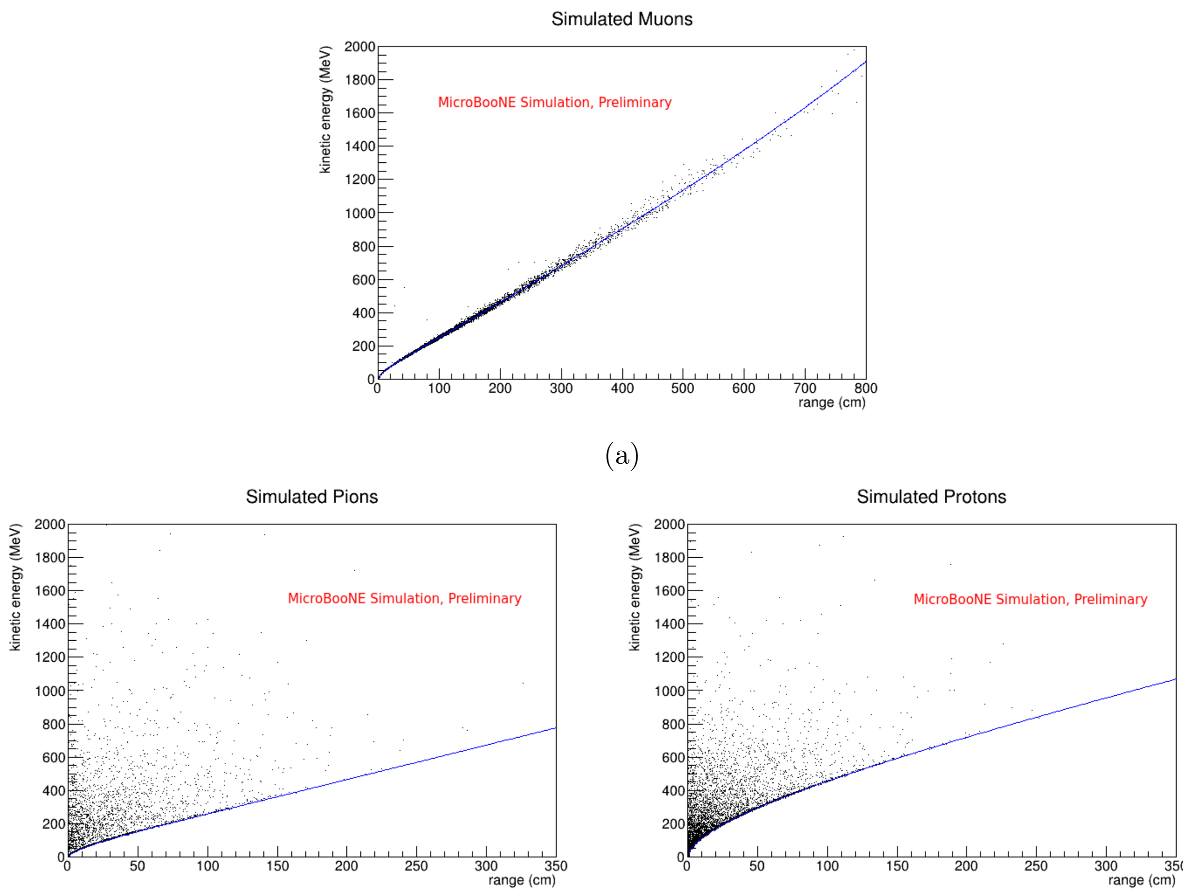
Shower energy is reconstructed from the visible charge observed in the collection plane, Q_{sh}, which is linearly related to shower energy. As discussed in more detail in [31], a fit to shower energy vs. Q_{sh} yields the conversion:

$$E[MeV] = (0.0126 \pm 0.0001)Q_{sh}, \tag{4}$$

where the error is from statistical uncertainty in simulated events used in the fit.

Our neutrino energy estimate is then simply calculated as the sum of the reconstructed track and shower energies for all prongs attached to the reconstructed neutrino interaction vertex. This is a measure of visible energy, not a sophisticated neutrino energy reconstruction. A more accurate energy reconstruction that addresses the limitations of the track energy estimate and includes more sophisticated techniques will be introduced in future works.

The accuracy of this visible neutrino energy estimate is illustrated in figure 11, which shows reconstructed vs. true neutrino energy for a sample of MC ν_eCC and ν_μCC events. Only simulated interactions in which all neutrino final state particles are contained inside the detector and that were reasonably well reconstructed (a neutrino vertex must have been found with an attached prong that contains at least 60% of the primary lepton's deposited



(a)

(b) (c)

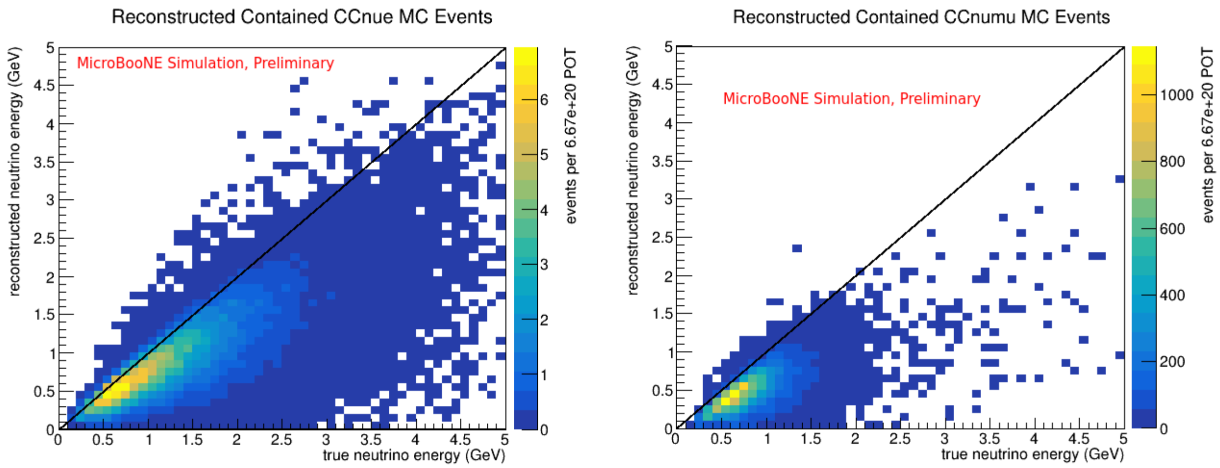
Figure 10: Kinetic energy vs. range for simulated muons (a), charged pions (b), and protons (c) that begin and end in the detector fiducial volume and were produced in MC ν_μCC interactions. The fits used in track energy reconstruction are shown in blue.

charge) are included in these plots[2]. While the majority of events' reconstructed energy falls below the overlaid one-to-one line and underestimates the true neutrino energy (as expected for this visible energy calculation), there is a reasonably strong linear relationship.

I. LArPID: A Prong Classification CNN

To aid in event selections and physics analyses, another CNN, the **L**iquid **Ar**gon **P**article **ID**entification (LArPID) network, was developed to provide additional information about reconstructed prongs. CNNs have effectively been used for particle identification in the past (e.g. in NOvA [32]) and hold particular promise for LArTPCs given their ability to

[2] Of simulated contained ν_eCC events with a reconstructed vertex, 66% have an attached shower containing at least 60% of the simulated electron's deposited charge. For ν_μCC events (same conditions), 74% have an attached track with at least 60% of the simulated muon's deposited charge.



(a) (b)

Figure 11: Reconstructed vs. true neutrino energies for MC ν_eCC (a) and ν_μCC (b) interactions that were successfully reconstructed (a prong from a neutrino candidate vertex was reconstructed with at least 60% of the simulated primary lepton's deposited charge) and in which all simulated neutrino final state particles are contained (begin and end inside the detector).

image neutrino interactions with mm-scale precision. While the primary aim of LArPID is to perform particle identification, it also predicts the input prong's production process and useful reconstruction quality metrics. Specifically, for each input prong, LArPID predicts:

- Particle type: The network outputs five scores indicating how likely it is that the prong was produced by a muon, electron, photon, charged pion, or proton. As the vast majority of charged particles produced in MicroBooNE neutrino interactions are of one of these types, other very rare particles (e.g. kaon, lambda, or sigma particles) are ignored. While the prong is assigned the particle type with the highest score, as we will see in section III A, taking into account all five particle scores is useful in quantifying how confident we can be with this classification.

- Production process: The network outputs three scores indicating how likely it is that the prong represents a primary final state particle produced in the neutrino interaction, a secondary particle with a charged parent, or a secondary particle with a neutral parent. Rather than attempt to classify all common secondary particle production processes (Michel decays, delta ray production, pion decay, etc.) these broad classes were chosen to simplify the prediction while accomplishing its primary aim: distinguishing primary final state particles from secondaries. The two most general types of observable secondary topology classes - those where the secondary prong does (charged parent) and does not (neutral parent) originate at the end point of another cluster - were chosen to provide additional information about secondaries and aid the network in organizing prongs into general topology classes. This production process prediction provides another valuable tool to aid in interpreting events.

- Purity: The fraction of visible energy in the prong that was deposited by the true particle. Here, visible energy is calculated as the sum of all pixel values for all 2D wire-plane-image pixels used in the 3D space points that make up the reconstructed prong. As discussed in section II I 3, when training the network on Monte Carlo simulation, the labelled truth particle is the simulated particle that deposits the most visible energy in the input reconstructed prong.

- Completeness: The fraction of all visible energy deposited by the true particle that was reconstructed in the input prong (where visible energy and "true particle" are defined as above for the purity prediction).

Analyzing these network outputs for all reconstructed prongs attached to a candidate neutrino interaction vertex provides valuable information about both the prongs and the candidate neutrino interaction. The particle classification outputs not only aid in identifying particles and selecting neutrino interactions with desired final states, but could also allow for a more accurate neutrino energy estimate by providing a robust particle hypothesis in e.g. range-based momentum calculations or neural-network based energy estimation techniques utilizing high-level reconstruction outputs. In addition to organizing particles into parent/daughter hierarchies for true neutrino interactions, the particle production process can be used to veto mis-reconstructed neutrino interaction vertices placed at the position of e.g. a particle decay. As shown in section III A, it is particularly helpful in CC ν_e event selections as it can veto background events where the candidate primary electron prong is in fact a secondary (for example a charged pion decay product or cosmic Michel electron). And the completeness and purity metrics allow for the identification of poorly reconstructed prongs: prongs reconstructed from energy depositions of a variety of different particles or those representing a fragment of a true particle. These reconstruction quality metrics can be used to better understand reconstructed neutrino interactions and, in the future, to improve the reconstruction as inputs to downstream algorithms or networks that could reorganize clusters into prongs that better represent particle trajectories.

The following sections detail this LArPID network's inputs (section III1), architecture (section III2), training details (section III3), and performance (section III4). Preliminary studies on interpreting the model are discussed in section III5.

1. Network Inputs and Image Preprocessing

The LArPID network operates on all three 2D wire-plane images (with tagged cosmic pixels removed) of both the input reconstructed prong and the full event. Going back to the 2D images provides the network with information that may have been lost during reconstruction as a result of dead channels or other errors. For example, in the probable CC ν_e data event shown in figure 12, shower pixels near the interaction vertex are present in the collection plane, but are missing in both the U and V planes. As a result, no 3D space points could be formed near the vertex, and the reconstructed 3D shower prong begins at a distance from the reconstructed neutrino interaction vertex. If considering only the 3D reconstruction outputs, this might indicate that the shower is a photon (which travel a distance from the vertex before pair-converting and depositing energy). However, by operating on the original 2D wire-plane inputs, the LArPID network can see the shower extending back to the vertex in the collection plane and classifies this shower as an electron.

Including the full-event context images, with all non-cosmic-tagged pixels, provides the network with a wealth of additional information that aids in all the network's tasks. Seeing the full event improves particle identification accuracy by allowing the network to learn physical features associated with different particle types. For example, photons start at a distance from the interaction vertex, whereas electrons begin depositing energy directly at the vertex. Neutral pions decay into a pair of photons, so the network can learn that when it sees two showers pointing back to the vertex these are likely photons. Preliminary studies indicating that the network indeed picks up on this kind of context information in assigning particles scores are presented in section III5. The context information is of course also needed to observe parent or fellow neutrino final state particles in distinguishing between primaries and secondaries in the production process classifier. Finally, allowing the network to observe when prong pixels are embedded in a larger cluster or a mix of different clusters

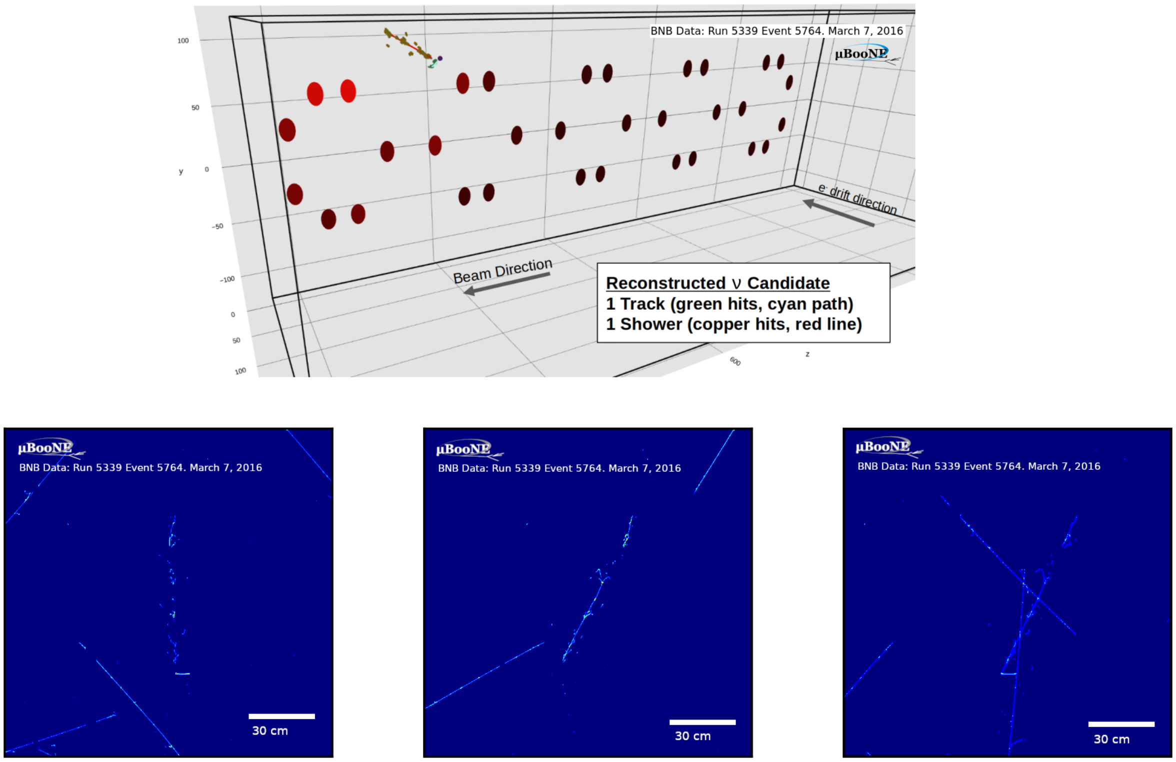


Figure 12: Bottom: zoomed in view of three wire plane images for a probable CC ν_e interaction in a MicroBooNE open dataset. The views are from the U induction plane (left), the V induction plane (center), and the Y collection plane (right). Top: Reconstructed neutrino candidate for the same event with PMT positions recording flashes highlighted (purple: reconstructed neutrino vertex position, copper: 3D space points in reconstructed shower, green: 3D space points in reconstructed track). This event passed our selection even though there is a visible gap in 3D spacepoints between the shower and vertex caused by unresponsive wires in the U and V planes. However, the prong-CNN still designated the shower electron-like likely due to the shower visibly starting from the vertex in the Y plane. This is a candidate example where using the low-level 2D information has helped overcome downstream reconstruction errors.

is needed for accurate completeness and purity estimates.

The prong and context images passed to the network are processed as follows. To obtain the prong images, 3D space points in the reconstructed prong are projected back into the 2D wire-plane images to obtain all associated pixels. To reduce computational requirements, these images are then cropped to a 512x512 pixel (153.6 x 153.6 cm) window. This window size is large enough to encompass most charged particles from neutrino interactions in MicroBooNE. This crop is performed separately in each wire plane. In a given wire-plane image, when the full set of prong pixels does fit within this window, the crop is centered around the middle of the prong (the point half-way between the min and max row and column). If the prong pixels do not fit within this window, the crop is centered around the end of the prong for tracks and around the beginning of the prong for showers. This decision was taken as most of the information in distinguishing muon, charged pion, and proton tracks lies at their end (as the particle slows down and potentially decays), whereas the information needed to distinguish between photon and electron showers lies at the beginning, in the shower trunk. In each wire-plane, the context image is cropped around the same region as the prong image. Before passing these six images (one prong and one context image for all three wire-planes) to the network, pixel values are normalized into the range of roughly -1 to 1 by subtracting the mean and dividing by the standard deviation of all final state particle pixels in a large set of simulated neutrino interactions. An example of the six input images passed to the network for a reconstructed charged pion track from a simulated CC ν_μ interaction is provided in figure 13.

2. Network Architecture

The LArPID network architecture is illustrated in figure 14. A 34-layer residual network (ResNet34) [24] was chosen for the CNN[3]. LArPID's ResNet34 CNN has two input channels for the prong and context image of one wire plane, and it operates separately on each wire plane using shared weights for all three. A 2D adaptive average pooling operation is performed on the output of the final layer, providing 512 learned features for each of the three wire planes. These features are concatenated into a single 1536-length feature vector summarizing information learned about the input prong. This concatenated feature vector is then used as input to four multi-layer perceptrons (MLPs) used to perform the four network tasks. Each MLP has a single hidden layer[4]. The particle classification MLP has a five-neuron output layer, whose logits are passed through a softmax to provide the muon, charged pion, proton, electron and photon scores. The particle production process classification MLP uses a length three output layer and softmax to provide the primary, secondary with charged parent, and secondary with neutral parent scores. The purity and

[3] While slightly improved performance might be achieved by using an deeper network, this would have increased the computational complexity and made the network difficult to deploy on cpus as required for large-scale MicroBooNE data processing. An 18 layer ResNet CNN was tested as well, but was found to provide lower performance.

[4] Increasing the number of hidden layers was found to increase the time it took the model to converge during training without improving performance

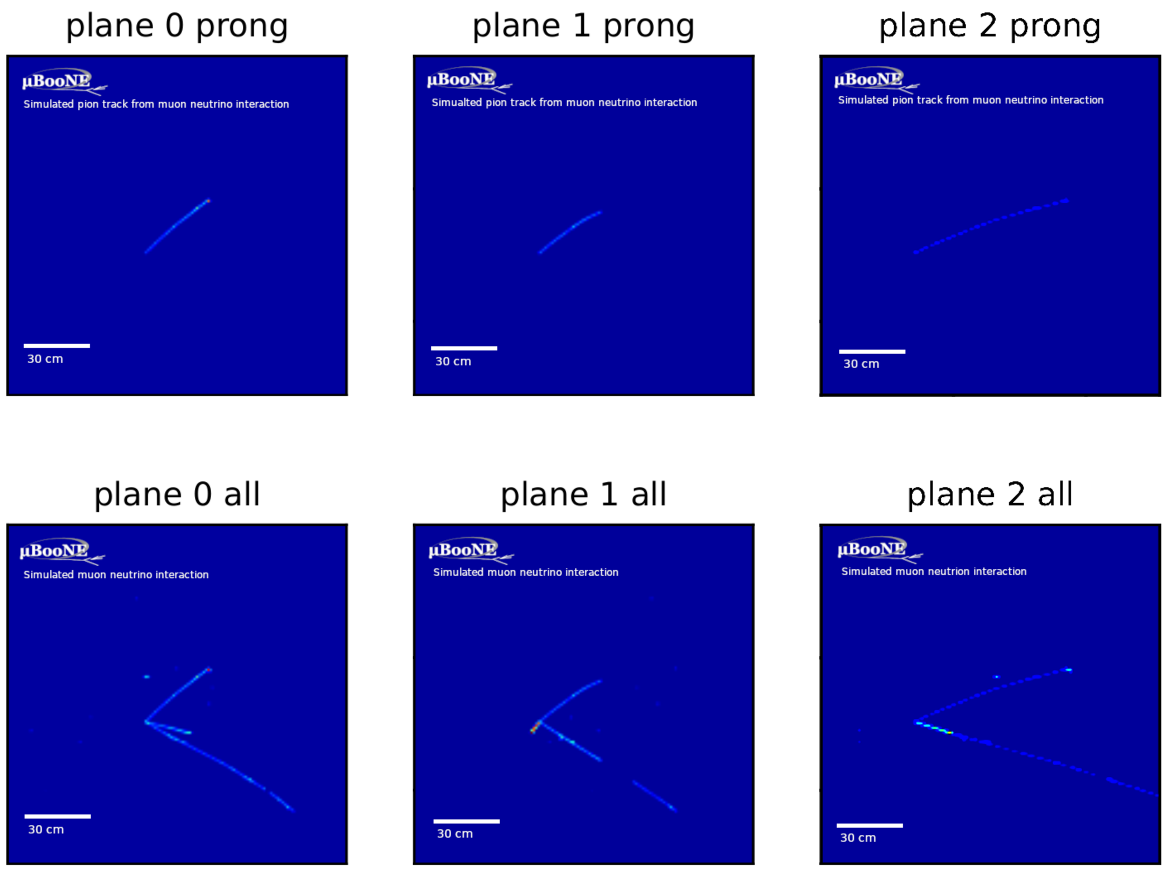


Figure 13: Example LArPID input images for a reconstructed charged pion track from a simulated CC ν_μ interaction. Top row: prong images for all three wire planes. Bottom row: full event context images for all three wire planes

completeness regression MLPs use a single-neuron output layer and sigmoid to provide purity and completeness predictions in the required physical range of 0 to 1.

3. Training

The LArPID network was trained on a sample of over 652,000 reconstructed prongs from MicroBooNE neutrino Monte Carlo simulations overlaid over off-beam cosmic-ray background data [MC citation]. Only prongs attached to reconstructed candidate neutrino interaction vertices were considered. If an event had multiple reconstructed neutrino vertices, only prongs attached to the vertex with the highest LArMatch neutrino keypoint score were selected for training. Two additional requirements were imposed on prongs selected for the training sample:

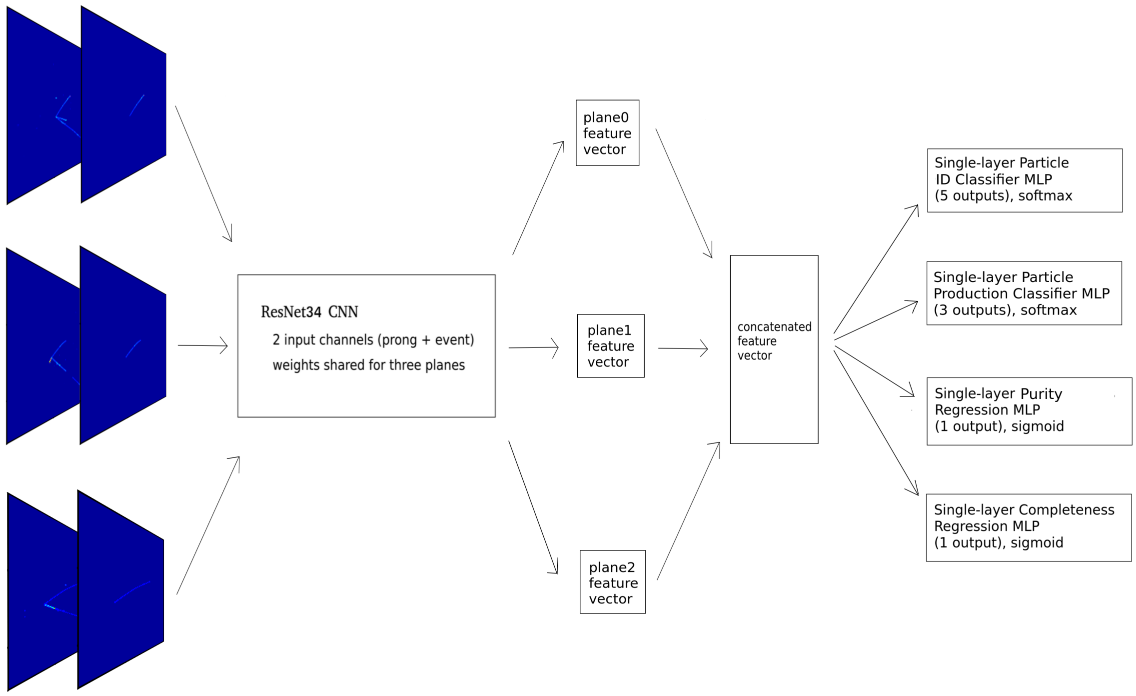


Figure 14: LArPID network architecture with the example inputs from figure 13.

- Training prongs must have at least 10 above-noise-threshold pixels (which span 3cm in the detector) in all three wire-plane images. Studies were performed to test if increasing this minimum-pixel threshold might improve performance on larger prongs that are primarily considered in neutrino event selections. However, it was found that the network, when trained only on larger (minimum pixel threshold = 90) prongs, did not perform better on a validation set including only such large prongs than when trained on a prong sample including the smallest reconstructed prongs.

- To allow assignment of truth labels, no more than 20% of charge included in training prongs could come from cosmic-ray contamination.

To assign truth labels for training and validation, 3D space points included in reconstructed prongs are projected back into the three wire-plane images to obtain all associated 2D pixels. The total amount of charge in all prong pixels that was deposited by each simulated charged particle produced in the neutrino interaction is summed, and the true-particle-type label is assigned as the simulated particle that deposited the most charge. The total charge deposited in the entire detector by this truth-matched simulated particle is also calculated, and the true completeness value is assigned as the fraction of this charge included in the reconstructed prong. The purity value is assigned as the fraction of all charge in the reconstructed prong that is contributed by this truth-matched simulated particle. Finally, the particle production process label is assigned according to how the truth-matched simulated particle was generated. The number of prongs in the training sample truth-matched to each of the five particle types is shown in table I below.

muons	charged pions	protons	electrons	photons
163,987	53,871	266,653	90,940	76,915

Table I: Number of prongs truth-matched to each of the five particle classes in the LArPID training sample

The network was trained over this sample for 20 epochs using a multi-task loss function with learned weights following the procedure outlined in [33]. A loss must be defined for each of LArPID's four tasks (predictions) and combined to form the network's full loss function. The relative weight attached to each task-specific loss function can significantly impact the network's performance, so the value chosen for these weights is important. The method employed in [33] treats task weights as learned network parameters that can be optimized during training. The authors found that this technique can provide superior performance than even optimal hard-coded weights (e.g. hard-coded values of w_1 and w_2 in the two-task loss function $\mathcal{L} = w_1\mathcal{L}_1 + w_2\mathcal{L}_2$). We confirmed this in the case of LArPID by varying hard-coded task weights used in combining task specific loss functions. We found that the network trained with learned weights outperformed the network trained with any set of hard-coded weight values. We further found that the network's particle classification performance, perhaps the most important network task, was no better when trained only on that task than when trained on all four tasks using learned task weights. Therefore, using the technique outlined in [33], the loss function used to train LArPID was defined as:

$$\begin{aligned}\mathcal{L} &= e^{-s_{cr}}\mathcal{L}_{MSE}(\hat{y}_{cr}, y_{cr}) + e^{-s_{pr}}\mathcal{L}_{MSE}(\hat{y}_{pr}, y_{pr}) \\ &\quad + 2e^{-s_{ic}}\mathcal{L}_{CE}(\hat{y}_{ic}, y_{ic}, w_{ic}) + 2e^{-s_{pc}}\mathcal{L}_{CE}(\hat{y}_{pc}, y_{pc}, w_{pc}) \\ &\quad + s_{cr} + s_{pr} + s_{ic} + s_{pc}\end{aligned} \tag{5}$$

where $\mathcal{L}_{MSE}$ and $\mathcal{L}_{CE}$ are the mean squared error (for regression tasks) and cross entropy (for classification tasks) loss functions, respectively; y and $\hat{y}$ are the true and predicted network outputs, respectively, for a given input and network task; the s parameters are the learned task-specific loss weights; the subscripts cr, pr, ic, and pc denote quantities for the **c**ompleteness **r**egression, **p**urity **r**egression, particle-**i**dentification **c**lassification, and particle-**p**roduction-process **c**lassification tasks, respectively; and w_{ic} and w_{pc} are vectors of class weights based on the number of prongs belonging to each particle identification and production process classes, respectively, in the full training sample. These class weights are calculated once upfront and used to weight contributions to the cross entropy loss functions to account for class imbalances. This ensures, for example, that the network doesn't learn to increase the probability of classifying all inputs as protons simply because there are more proton-labelled prongs in the training sample than any other type of particle.

Additional training details are as follows:

- One data augmentation technique was employed to reduce over-fitting: input images were randomly flipped horizontally or vertically, each with a probability of 50%. As demonstrated in figure 15, which shows overall particle classification accuracy for the training sample and an independent validation sample with 50,000 prongs (10,000 per particle type), over-fitting was not a significant issue. Similar few percent differences were observed in the training vs. validation performance of the other network tasks. More details and discussion on network performance in the validation sample are presented in section III4.

- The AdamW gradient descent algorithm [34] was used to update network weights.

- The single-cycle cosine annealing learning rate scheduler shown in figure 16 was used. This type of variable learning rate has been found to reduce the number of training epochs required for convergence [35]. The minimum and maximum learning rate values used in the scheduler were chosen by varying the learning rate in a test training run and determining the range of rate values over which the model continues to converge. A variety of other single-cycle, oscillatory, and stepped learning rate schedulers were tested, but none achieved better performance or faster convergence than the chosen schedule depicted in figure 16.

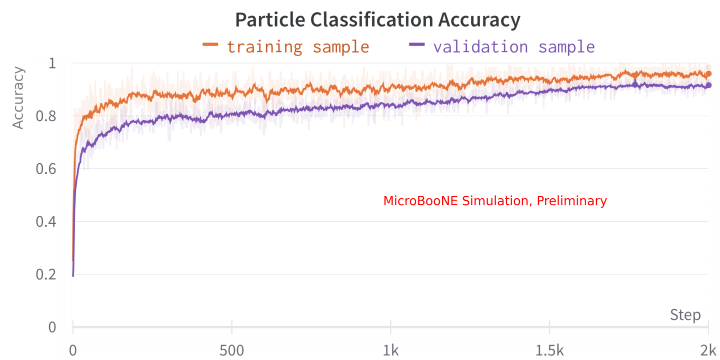


Figure 15: Overall particle classification accuracy for the training sample and an independent validation sample with 50,000 prongs (10,000 per particle type) as a function of training iteration.

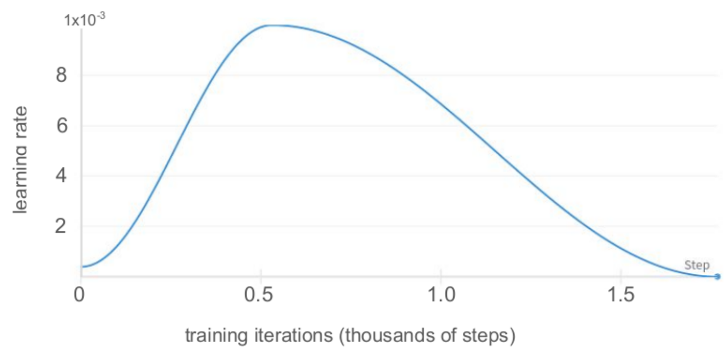


Figure 16: The learning rate scheduler used to train the LArPID network, shown as a function of training iterations. The full schedule lasts for 20 epochs.

4. Network Performance

The performance of the LArPID network was tested on an independent validation sample of 50,000 reconstructed prongs (10,000 of each particle type: muons, charged pions, protons, electrons and photons). The same selection criteria detailed in section III1 used to construct the training sample were used for this validation sample. For the classification task results, an additional requirement that at least 60% of the validation prongs' total pixel charge be contributed by a single simulated particle (true prong purity $> 60\%$) was imposed. This requirement was placed to ensure sensible truth labels could be assigned.[5]

The network achieved an impressive overall validation accuracy of 91.8% on the particle

[5] A version of the network was also trained with the same true prong purity $> 60\%$ requirement, but it was found that this did not improve the network's classification performance on a purity $> 60\%$ validation sample. This requirement was therefore not ultimately imposed on the training sample so that the network could be trained to make more accurate purity and completeness predictions in cases where purity $< 60\%$.

identification task. A confusion matrix showing the accuracy for each particle type is shown in table II. The per-particle validation accuracy is very high for all five particle types, and when the network does mis-classify a prong, it almost always applies the label of a particle that leaves similar signatures in the detector. For example, mis-classified electrons are almost always assigned a photon label (and vice versa) and mis-classified charged pions are almost always assinged a muon label (and vice versa).

	True $e^{\pm}$	True γ	True $\mu^{\pm}$	True $\pi^{\pm}$	True p
Fraction classified as $e^{\pm}$	**84.5%**	5.2%	0.1%	0.5%	0%
Fraction classified as γ	12.7%	**94.3%**	0.2%	0.2%	0.1%
Fraction classified as $\mu^{\pm}$	0.4%	0.1%	**93.9%**	11.5%	0.3%
Fraction classified as $\pi^{\pm}$	2.3%	0.3%	5.6%	**86.5%**	1.6%
Fraction classified as p	0.1%	0.1%	0.2%	1.4%	**97.9%**

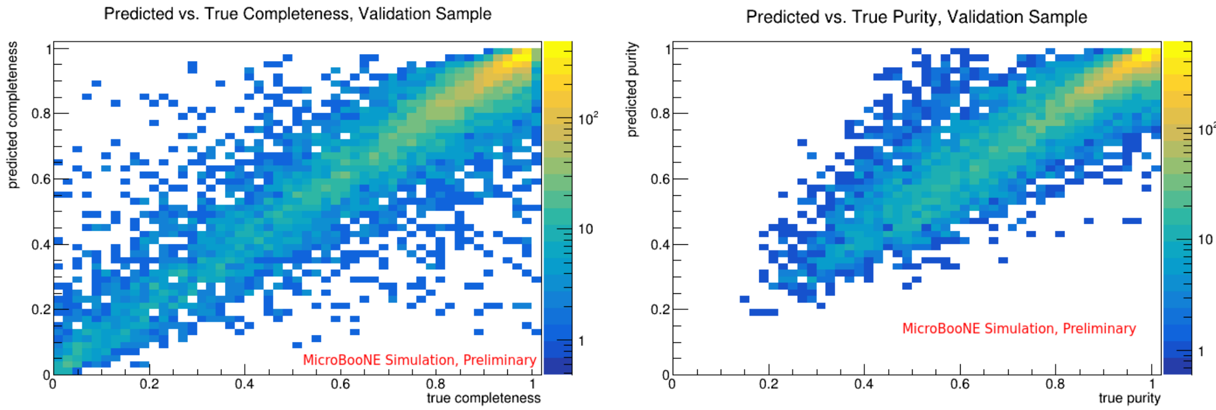
Table II: A confusion matrix showing LArPID's particle classification accuracy in the validation sample: the fraction of prongs truth matched to each particle type that LArPID classified as an electron ($e^{\pm}$), photon (γ), muon ($\mu^{\pm}$), charged pion ($\pi^{\pm}$), or proton (p) (columns sum to 100%).

The network's overall validation accuracy on the particle-production-process classification task was similarly high, at 89.0%. As shown in table III, accuracy is high for all three production-process classes. Accuracy is highest for identifying secondaries with a neutral parent, likely as this signature (a prong created with no other tracks or showers originating at it's start position) is more unique than the other two classes.

	True primary	True neutral parent	True charged parent
Fraction classified as primary	**87.8%**	3.4%	6.5%
Fraction classified as neutral parent	2.9%	**93.6%**	6.9%
Fraction classified as charged parent	9.3%	3.0%	**86.7%**

Table III: A confusion matrix showing LArPID's particle-production-process classification accuracy in the validation sample: the fraction of prongs with each true production-process class that LArPID classified as a primary neutrino-final-state particle, a secondary with a charged parent, or a secondary with a neutral parent (columns sum to 100%).

The validation accuracy of the network's completeness and purity predictions are illustrated in figure 17, which shows 2D histograms of the predicted vs. true reconstruction quality metrics. For a given range of true completeness or purity values, there is a non-trivial spread in the predicted values. However, the bulk of the prongs in these heat maps



(a) (b)

Figure 17: A 2D histogram showing predicted vs. true completeness (a) and purity (b) for prongs in the validation sample.

do roughly follow a one-to-one line. So while the exact value of LArPID's completeness or purity prediction for a given prong should not be interpreted with extremely high confidence, these predictions are robust in revealing whether the input prong is likely to be a small fragment of a true particle, a mostly complete reconstruction, or something in between (completeness) and, similarly, whether it is likely to be reconstructed from a mix of different particles or mostly from a single particle (purity).

5. *Interpreting the Model*

Preliminary image manipulation studies were carried out to shed light on what information LArPID is using from input images to make its predictions. While clear decision-making algorithms cannot be extracted from the complex network of neurons in deep learning models, these interpretability studies can provide valuable insights into how neural networks are making decisions. The technique employed here involves testing hypotheses on what information is being used in the network by providing the model with a set of counter-factual images. This is done by replacing an input reconstructed prong or a particle from the context image with another simulated particle and seeing how the network output changes.

An example of one such image manipulation study is shown in figure 18. Here, we checked to see if the network is learning, from examples of π^0 decay photons, that when two electromagnetic showers' initial trajectories can be traced back to an intersection (the π^0 decay position) near the interaction vertex, that these showers are likely photons. Figure

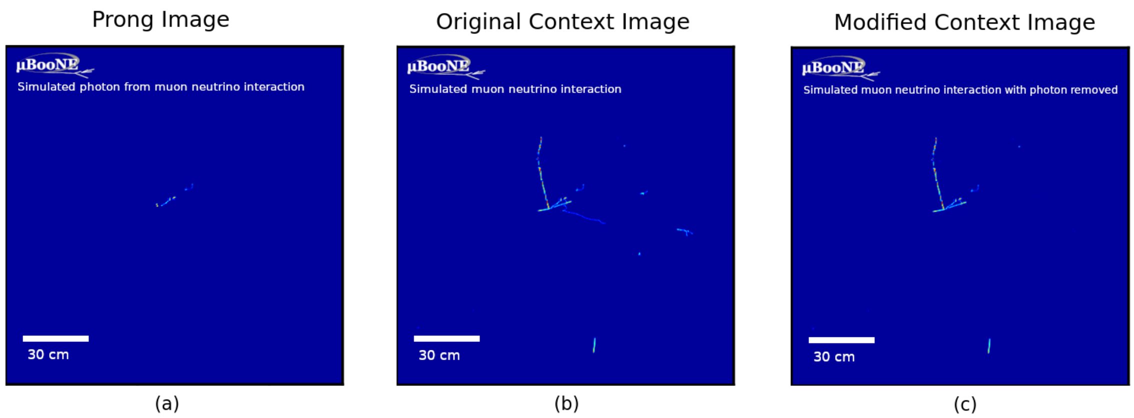


Figure 18: One wire-plane prong image (a) and context image (b) for a reconstructed photon prong produced during a π^0 decay in a simulated CC ν_μ interaction. A modified context image (c) in which the input photon prong's companion π^0 decay photon is removed. With the original inputs (a and b), LArPID outputs an electron score of 0.027 and photon score of 0.97. With the modified inputs (a and c), LArPID outputs an electron score of 0.21 and photon score of 0.77.

18 (a) and (b) show (for one wire-plane) the prong and context images for a reconstructed photon shower produced during a π^0 decay. The second π^0 decay photon is clearly visible in the context image. With these inputs, the network confidently and correctly predicted that the input prong is a photon, with photon and electron scores of 0.97 and 0.027, respectively. The network was then presented with the same input prong images but modified context images (figure 18c) that had the second π^0 decay photon removed. With these inputs, the network's confidence in its photon prediction decreased significantly, with photon and electron scores of 0.77 and 0.21, respectively. The same context manipulation had a similar effect in other examples of input photon prongs from π^0 decays, indicating that the network has indeed, as expected, learned to increase its photon scores for input showers that are consistent with π^0 decays.

Another set of image manipulations for an example LArPID input is shown in figure 19. Panels (a) and (b) show (for one wire-plane) input images for a prong (from a simulated CC ν_μ interaction) that is reconstructed from both a short charged pion track and the electron shower produced following the pion decay. As the majority of this reconstructed prong comes from the simulated electron, the network predicts that this prong is an electron, with electron and charged pion scores of 1 and $2.4{\cdot}10^{-3}$, respectively.

We wanted to test how the network's individual particle scores are impacted when an

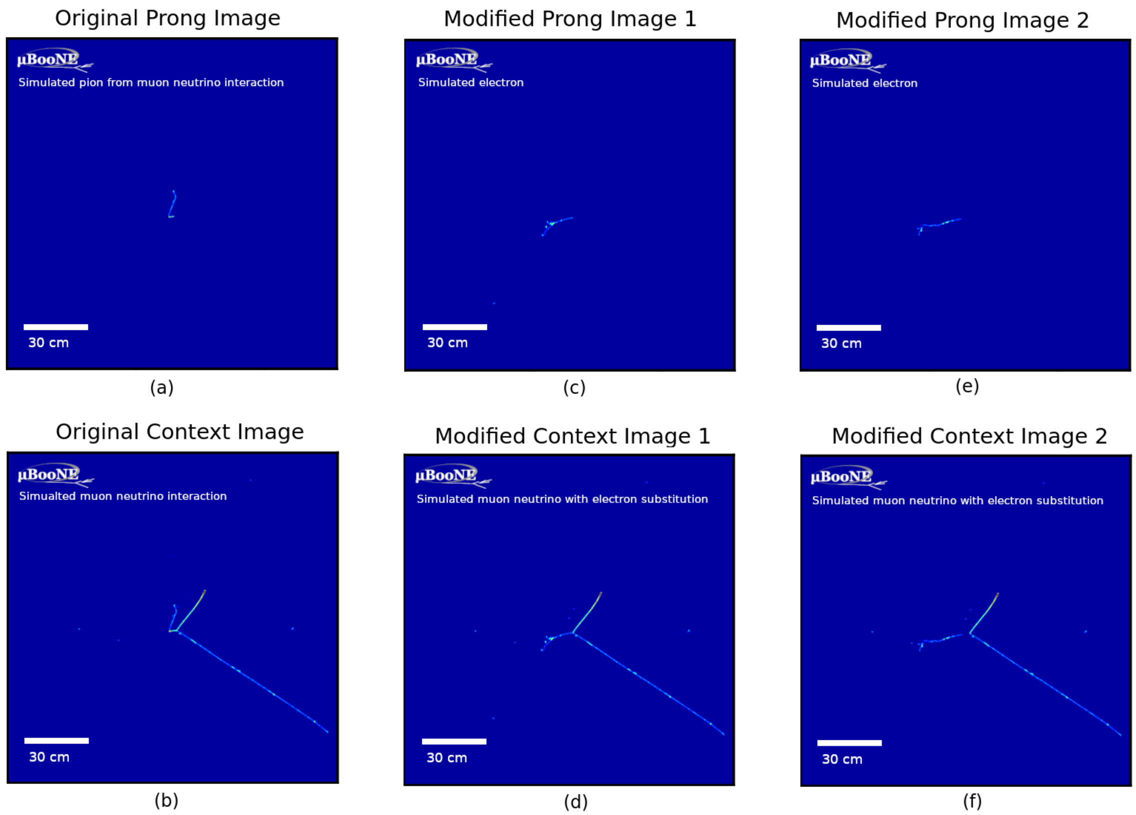


Figure 19: One wire-plane prong image (a) and context image (b) for a prong reconstructed from a charged pion and an electron produced during the pion decay in a simulated CC ν_μ interaction. A modified prong (c) and context image (d) in which the charged pion and its decay electron were replaced by a simulated electron with the same start position and momentum and the original charged pion. A second set of modified prong (e) and context (f) images in which the same substitution was performed with another simulated electron, but the replacement electron was simulated to begin at a short distance from the neutrino interaction position. LArPID's output electron, photon, and charged pion scores were 1, $8.9{\cdot}10^{-4}$, and $2.4{\cdot}10^{-3}$, respectively, for the original images (a and b); 0.99, $6.5{\cdot}10^{-3}$, and $1.8{\cdot}10^{-4}$, respectively, for the first set of modified images (c and d); and $3.8{\cdot}10^{-4}$, 1, and $2.7{\cdot}10^{-6}$, respectively, for the second set of modified images (e and f).

input prong is reconstructed with significant contributions from different particles. In cases such as these, can information about what combination of particles contribute to a low-purity prong be gleaned from the individual particle scores? In figure 19 (c) and (d), the input prong was replaced with a pure simulated electron with the same start position and momentum as the original charged pion. With these inputs, the network's pion score decreased to $1.8{\cdot}10^{-4}$. A similar set of manipulations on combined charged pion-electron prongs yielded similar results, indicating that the charged pion score for classified electron prongs can be used to determine if such prongs likely came from a charged pion decay. This feature is exploited in the CC ν_e selection of section III A to reduce CC ν_μ backgrounds.

An additional set of image manipulations shown in figure 19 verify that the network is using the context images to learn that showers that start at a distance from an interaction vertex are more likely to be photons, and those that start at the vertex are more likely to be electrons. In panels (e) and (f), a similar manipulation is performed in which the input prong was replaced with a simulated electron, but this time with a start position at a small distance from the interaction vertex. With these inputs, even though the input is a true electron, the network confidently classifies it as a photon prong, with photon and electron scores of 1 and $3.8 \cdot 10^{-4}$, respectively. A similar set of manipulations in which simulated electrons were placed at a distance from an interaction vertex yielded the same result (a confident photon prediction), indicating that the network has indeed learned to use this context information for electron/photon discrimination.

These tests demonstrate the utility of such image manipulation studies in learning how the network is making decisions. In a future work, these manipulations will be performed at scale and the results quantified for a more complete set of counter factuals. This will improve understanding of the model, increase confidence in its predictions, and inform how its outputs might more effectively be used in event selections and physics analyses.

III. DEMONSTRATION: SELECTION OF INCLUSIVE ν_eCC AND ν_μCC INTERACTIONS IN MICROBOONE

A. CC nue inclusive selection cuts

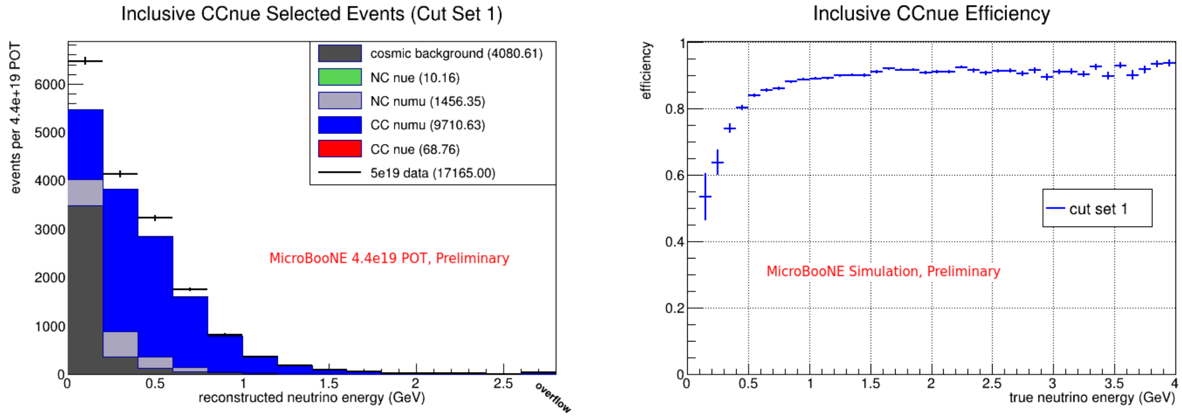
As a demonstration of the effectiveness of these reconstruction tools, we have developed a set of inclusive CC ν_e selection criteria utilizing the output of the LArMatch and LArPID networks. As we will show, an effective, high purity and efficiency CC ν_e selection can be achieved by selecting LArMatch-identified neutrino candidate vertices and filtering out cosmic and neutrino backgrounds with the Wire-Cell based cosmic tagger discussed in section II A and cuts on LArPID outputs of attached prongs. For this study, we use off-beam data to analyze the cosmic-ray background and simulated neutrino interactions overlaid over cosmic-ray background data for CC ν_e and neutrino background events.

The full set of CC ν_e selection criteria is provided in table IV. We will examine the impact of adding each selection cut one at a time. With each new cut, we will then discuss the motivation for its inclusion, show distributions of the cut variable for signal and background, show data/MC comparisons of reconstructed neutrino energy for events passing the new selection criteria using a small $4.4{\cdot}10^{19}$ POT MicroBooNE open data sample, and show the MC-predicted impact of the new cut on efficiency as a function of true (simulated) neutrino energy. A new "cut set" is defined in table IV as a set of selection criteria including a newly specified cut along with all previous cuts.

Table IV: Inclusive CC ν_e Selection Cuts

Cut	Notes
LArMatch-identified neutrino candidate vertex found inside the fiducial volume	Added in cut set 1 (included in cut sets 1-6 and final set)
3D space points of prongs attached to neutrino candidate do not all overlap with Wire-Cell-tagged cosmics	Added in cut set 2 (included in cut sets 2-6 and final set)
No LArPID-identified muon tracks are attached to neutrino candidate	Added in cut set 3 (included in cut sets 3-6 and final set)
At least one LArPID-identified electron shower attached to neutrino candidate	Added in cut set 4 (included in cut sets 4-6 and final set)
The largest identified electron was classified by LArPID as a neutrino final state particle	Added in cut set 5 (included in cut sets 5-6 and final set)
No tracks attached to neutrino candidate have a high LArPID muon score: max log(muon score) < -3.7	Added in cut set 6 (included in cut set 6 and final set)
The largest identified electron was classfied by LArPID as an electron with high confidence: log(electron score) $-$ (log(pion score) + log(photon score))/2 > 7.1	Added in final cut set

The first CC ν_e selection criteria is that a LArMatch-identified neutrino interaction vertex was found inside the detector fiducial volume. For our neutrino selections, we define the fiducial volume as 3cm from the space-charge corrected TPC boundary as in [17]. Data and MC reconstructed neutrino energy distributions with only this requirement are shown in figure 20a. The data excess seen here and as additional cuts are applied is not unique to our CNN-based reconstruction and selection and has been seen in other frameworks. As we will show in section III D, data and predictions with the full set of selection cuts are consistent once accounting for statistical and systematic uncertainties. The MC-predicted CC ν_e selection efficiency with just this vertex reconstruction requirement is shown in figure 20b. Vertex finding has a non-trivial impact on efficiency at low (<500 MeV) neutrino energies where electron showers and other prongs are small and more difficult for LArMatch



(a) (b)

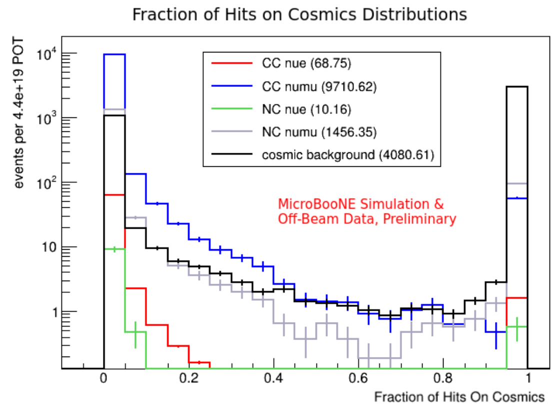
Figure 20: (a): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 1 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (b): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut set 1. The CC ν_e cut set 1 only contains the requirement that a LArMatch-identified neutrino vertex was reconstructed inside the fiducial volume. Cut sets are defined in table IV.

to identify and separate from cosmic background.

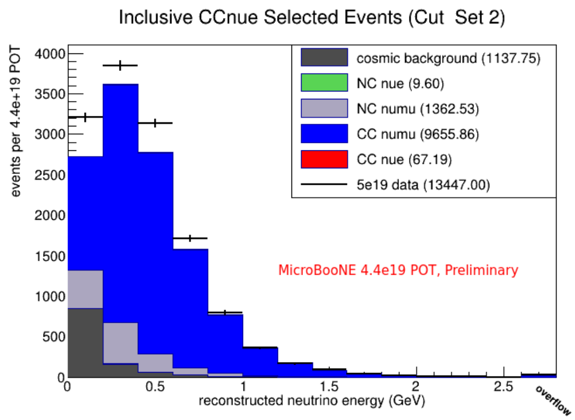
If LArMatch identifies more than one neutrino keypoint cluster in an event, we select the one with the highest keypoint score as the candidate neutrino interaction vertex. Further selection criteria apply to prongs attached to this candidate interatction vertex.

There is still a significant cosmic-ray background after selecting events with LArMatch-identified neutrino candidates. The majority of this background can be removed with the Wire-Cell cosmic tagger discussed in section II A. Figure 21a shows neutrino and cosmic background distributions for the fraction of 3D points in any cluster attached to the neutrino candidate vertex that was tagged as cosmic. Events in the final cosmic-dominated bin with 100% cosmic overlap (events where all hits in all prongs attached to the vertex have at least one constituent pixel that was tagged as cosmic by the Wire-Cell cosmic tagger) are rejected. Figure 21b shows the new reconstructed neutrino energy distributions with this requirement included (with "cut set 2"). Figure 21c compares the efficiency curve for this "cut set 2" to "cut set 1," which only includes the neutrino vertex reconstruction requirement. This cosmic-ray rejection cut does not have a large impact on CC ν_e efficiency.

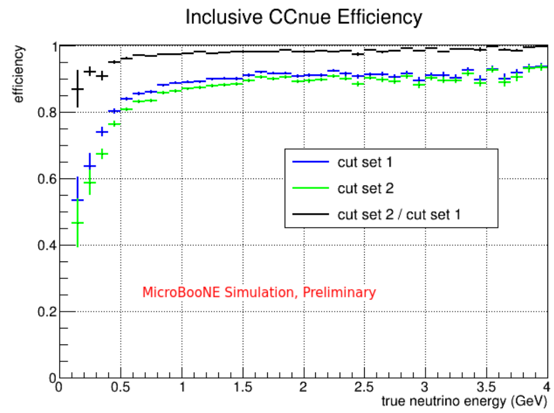
Further cuts on the LArPID outputs of prongs attached to the candidate neutrino vertex can remove almost all of the remaining cosmic and neutrino backgorund. As a first step, to remove most of the CC ν_μ background and some of the remaining cosmic background,



(a)



(b)

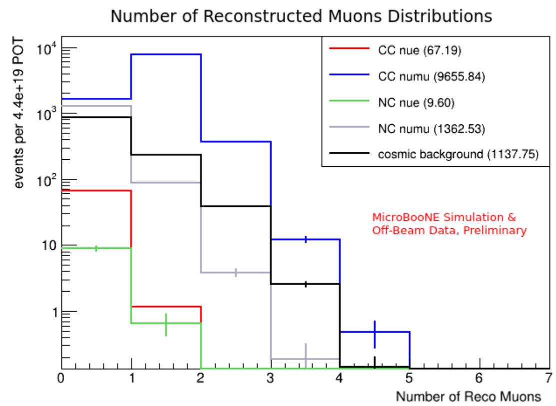


(c)

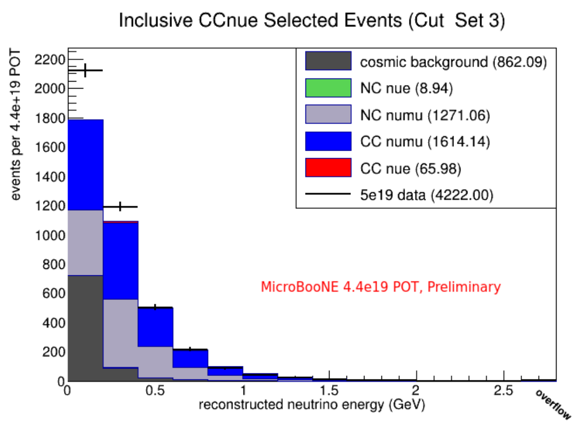
Figure 21: (a): MC and off-beam cosmic background data distributions of the fraction of hits associated with the candidate neutrino vertex that were constructed from pixels tagged as cosmics. Events in the final bin were rejected as the new requirement in CC ν_e cut set 2. (b): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 2 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut sets 1 and 2, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

events with identified muons are rejected. Figure 22a compares signal and background distributions for the number of reconstructed muons - the number of LArPID identified muon tracks attached to the candidate neutrino vertex - in events remaining after applying "cut set 2." The requirement that no reconstructed muons are present in the event was added in "cut set 3." Figure 22 (a) and (b) show the reconstructed neutrino energy distributions and efficiency curve comparisons for cut sets 2 and 3. This muon-track rejection cut does not have a large impact on efficiency.

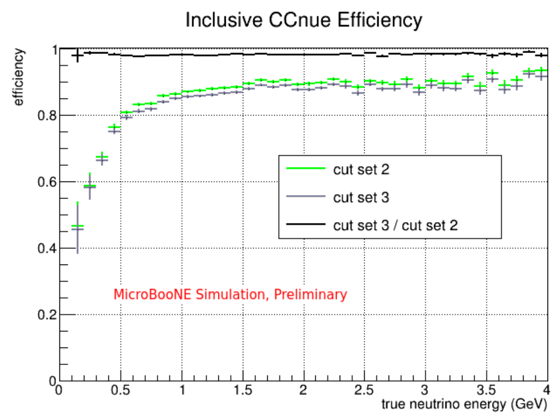
Now that events with identified muons have been removed, we select events with identified



(a)



(b)



(c)

Figure 22: (a): MC and off-beam cosmic background data distributions of the number of LArPID-identified muon tracks (reco muons) attached to the candidate neutrino vertex. The requirement that no reco muons are present was added in CC ν_e cut set 3. (b): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 3 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut sets 2 and 3, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

electrons. Figure 23a shows the distribution of the number of LArPID-identified electron showers for remaining signal and background events. The requirement that at least one electron shower was identified was added in "cut set 4." This removes the majority of the remaining neutrino and cosmic backgrounds, but has a moderate impact on our CC ν_e selection efficiency across neutrino energies (see figure 23c). The data/MC reconstructed neutrino energy comparison with "cut set 4" is shown in figure 23b.

As can be seen in figure 23a, it is not rare in true CC ν_e events for multiple electron showers to be identified. This is generally not due to LArPID incorrectly classifying showers, but

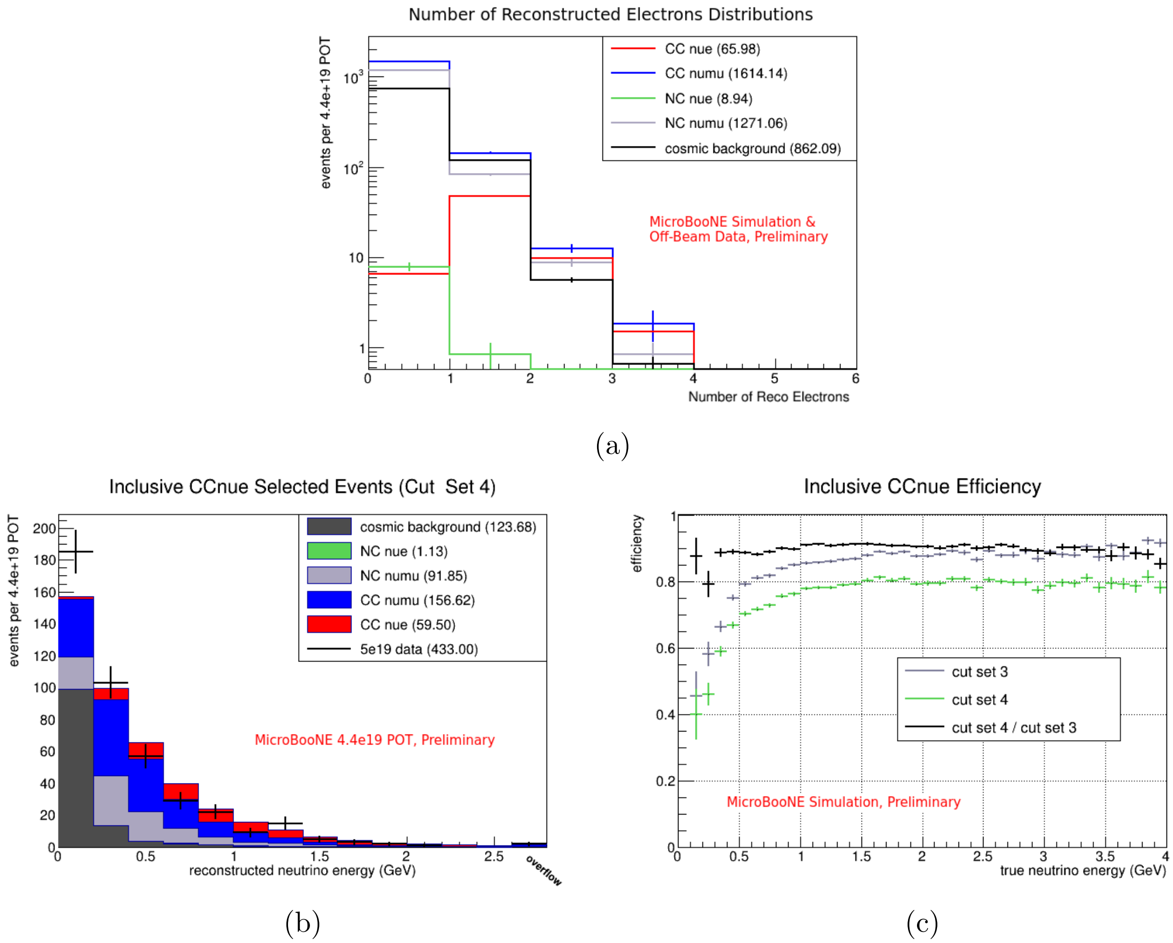


Figure 23: (a): MC and off-beam cosmic background data distributions of the number of LArPID-identified electron showers (reco electrons) attached to the candidate neutrino vertex. The requirement that at least one reco electron is present was added in CC ν_e cut set 4. (b): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 4 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut sets 3 and 4, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

clustering errors in which small fragments of the true primary electron are reconstructed as a different shower. In these cases, one reconstructed shower tends to carry the majority of the true electron's deposited charge. The candidate primary electron is therefore identified as the LArMatch-identified electron shower with the most charge.

The majority of the remaining cosmic and neutrino background can be removed by placing additional cuts on the LArPID outputs for this candidate primary electron shower. In much of the remaining background, a true electron is in fact present, but as a secondary, e.g. a Michel electron, delta ray, or an electron produced after a charged pion decay. Figure 24a

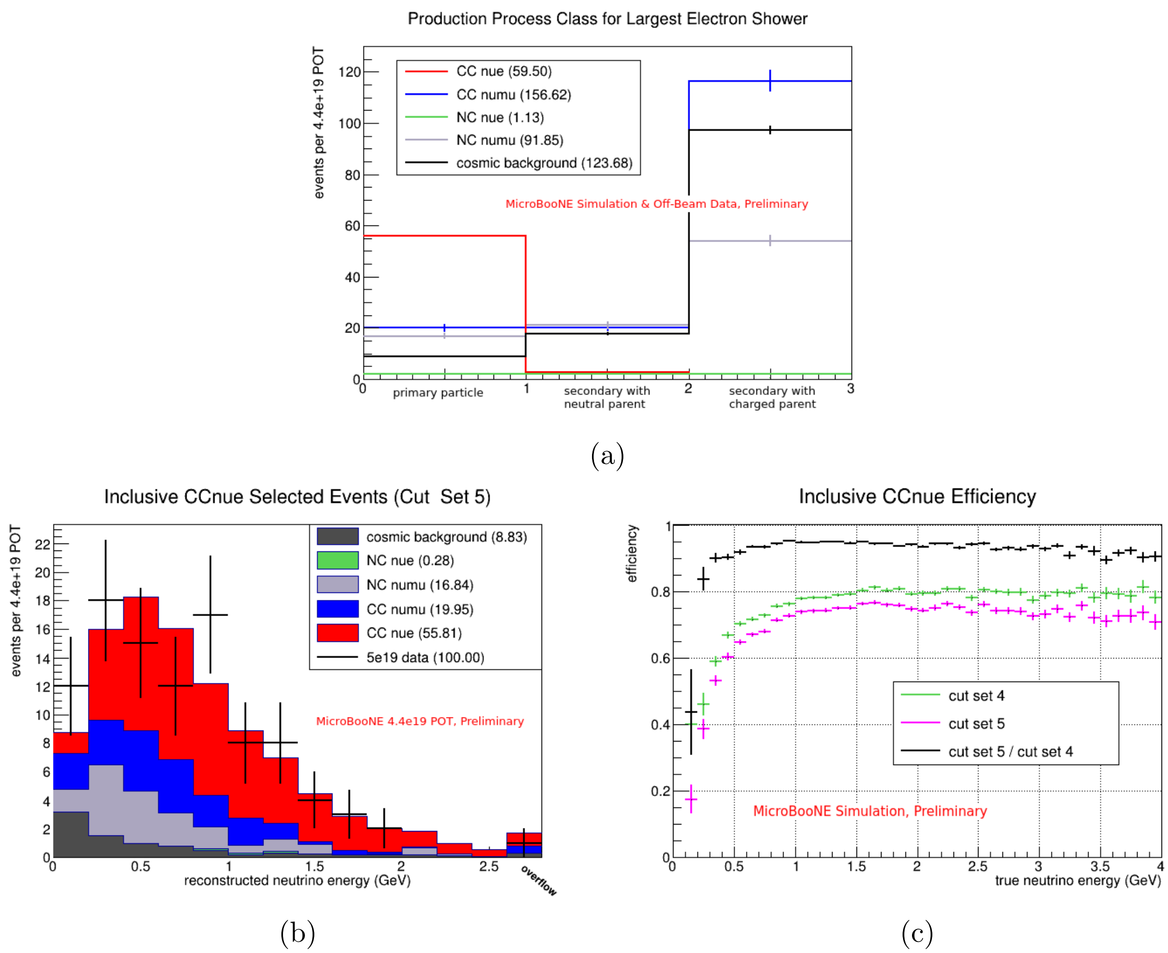


Figure 24: (a): MC and off-beam cosmic background data distributions of the candidate primary electron shower's LArPID production process class output. The requirement that the candidate primary electron was classified by LArPID as a primary particle was added in CC ν_e cut set 5. (b): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 5 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut sets 4 and 5, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

shows the output of the LArPID particle production classifier for the candidate primary electron in signal and background events remaining after "cut set 4." This classifier is able to accurately separate out the true primary electron showers from the candidates produced in background events, almost all of which are classified as secondaries. The requirement that the primary electron candidate was classified by LArPID as a neutrino final state particle was therefore added in "cut set 5." This new requirement has a fairly small impact on CC ν_e selection efficiency except at very low (<200 MeV) neutrino energies (see figure 24c). The new reconstructed neutrino energy distributions are shown in figure 24b.

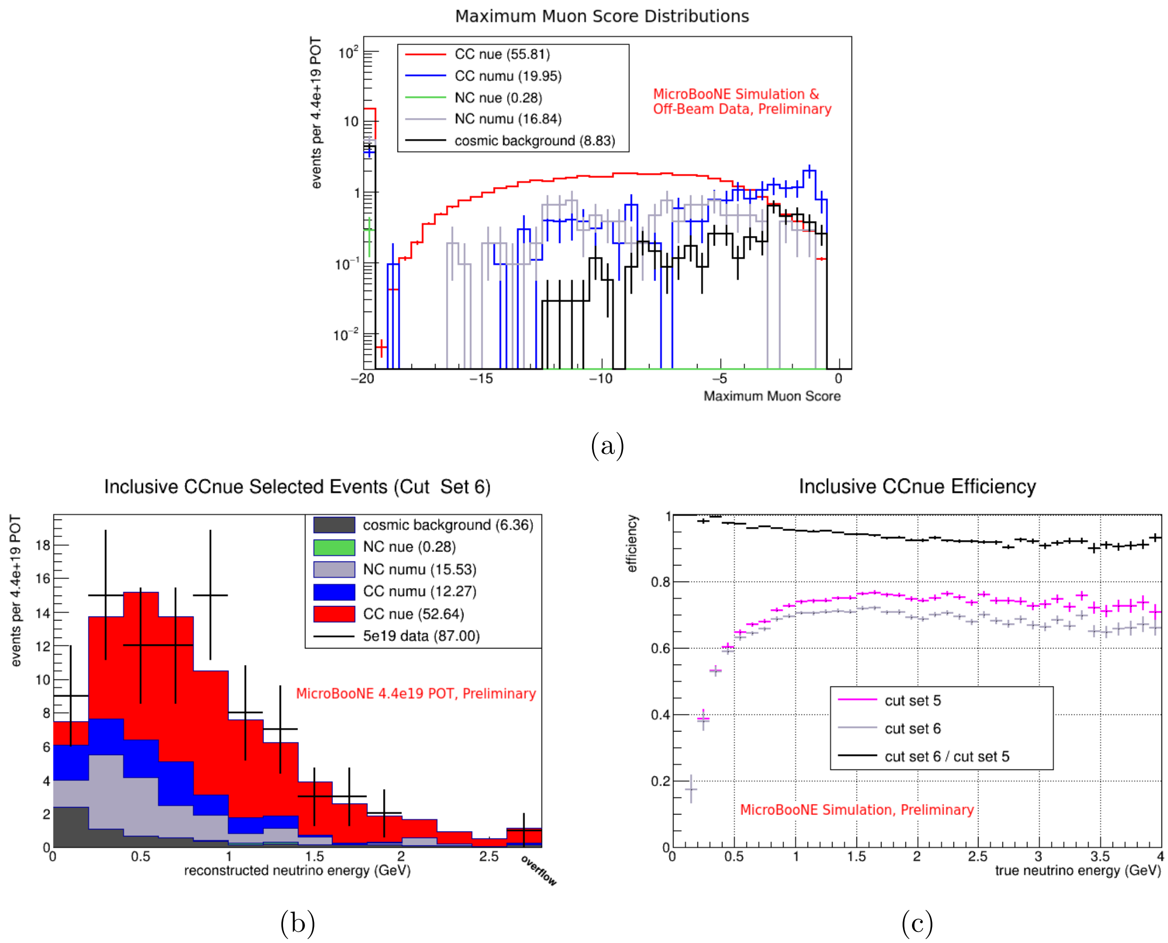


Figure 25: (a): MC and off-beam cosmic background data distributions of the log of the highest LArPID muon score for any track attached to the candidate neutrino vertex. The bin at -20 contains events with no tracks. The requirement that no track have a log(muon score) above -3.7 was added in CC ν_e cut set 6. (b): Data/MC reconstructed neutrino energy comparison of events selected by CC ν_e cut set 6 with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with CC ν_e cut sets 5 and 6, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

At this stage, much of the remaining background is present due to photons or secondary showers being mis-classified as primary electrons or muons being mis-classified as charged pions (and therefore not getting rejected by the no reconstructed muons cut). The latter issue is addressed first.

Generally, when muon tracks are mis-classified as charged pions, they still get a fairly high LArPID muon score. Events remaining after "cut set 5" with a true muon present can therefore be identified by searching for those containing a track with a high muon score (even though events with muon-classified tracks have already been removed). Figure 25a shows the distribution for the log of the maximum LArPID muon score for any track attached to the candidate neutrino vertex in remaining events. The cut on this distribution that provides the largest CC ν_e selection purity*efficiency product is -3.7. Events with a track that has a muon score above this value were therefore removed as the new requirement in "cut set 6." The cut set 6 neutrino energy distributions and efficiency curves are shown in figure 25 (b) and (c). This cut does not have a large impact on selection efficiency.

The final cut seeks to remove events where a photon or a mis-reconstructed charged pion prong (composed of a charged pion and the electron produced in its decay) was mis-classifed by LArPID as a primary electron. As with the mis-classified muon tracks, in these cases, the network tends to hedge its bets and also give these showers a high photon or charged pion score. An electron class confidence metric for the candidate primary electron was therefore defined as the difference in its LArPID electron score and the average of its charged pion and photon scores: $\log s_e - (\log s_\pi + \log s_\gamma)/2$, where s_e, s_π, and s_γ are the electron, charged pion, and photon scores, respectively.[6] The distribution of the primary electron candidate's class confidence metric for signal and background events remaining after "cut set 6" is shown in figure 26a. The signal and background distributions separate out well. The electron class confidence cut that maximized the CC ν_e purity*efficiency product, class confidence > 7.1, was added as the final CC ν_e selection cut. The data and MC reconstructed neutrino energy distributions with all selection cuts is shown in figure 26b, which includes a 1σ uncertainty band on the predictions including both statistical and systematic (discussed in section III C) errors. As we will show in section III D, the data and predictions are consistent within the quoted uncertainties. The final selection efficiency is shown in figure 26c along with a comparison to "cut set 6." Adding in the electron class confidence cut does significantly impact efficiency, but is necessary to remove remaining backgrounds.

Additional cuts (including utilizing the LArPID completeness, purity, and production process score values) were tested, but none outperformed these results. With the cuts enumerated in table IV, we are able to achieve a CC ν_e selection with an overall efficiency of 56.8% and purity of 91.1%, which is very competitive with PRD 105:112005 [17], the highest-efficiency CC ν_e search previously published by MicroBooNE. Additional data/MC comparisons (including hand scans of selected data events), performance plots, and a detailed

[6] We also attempted cutting separately on the charged pion and photon scores, but this did not achieve better results.

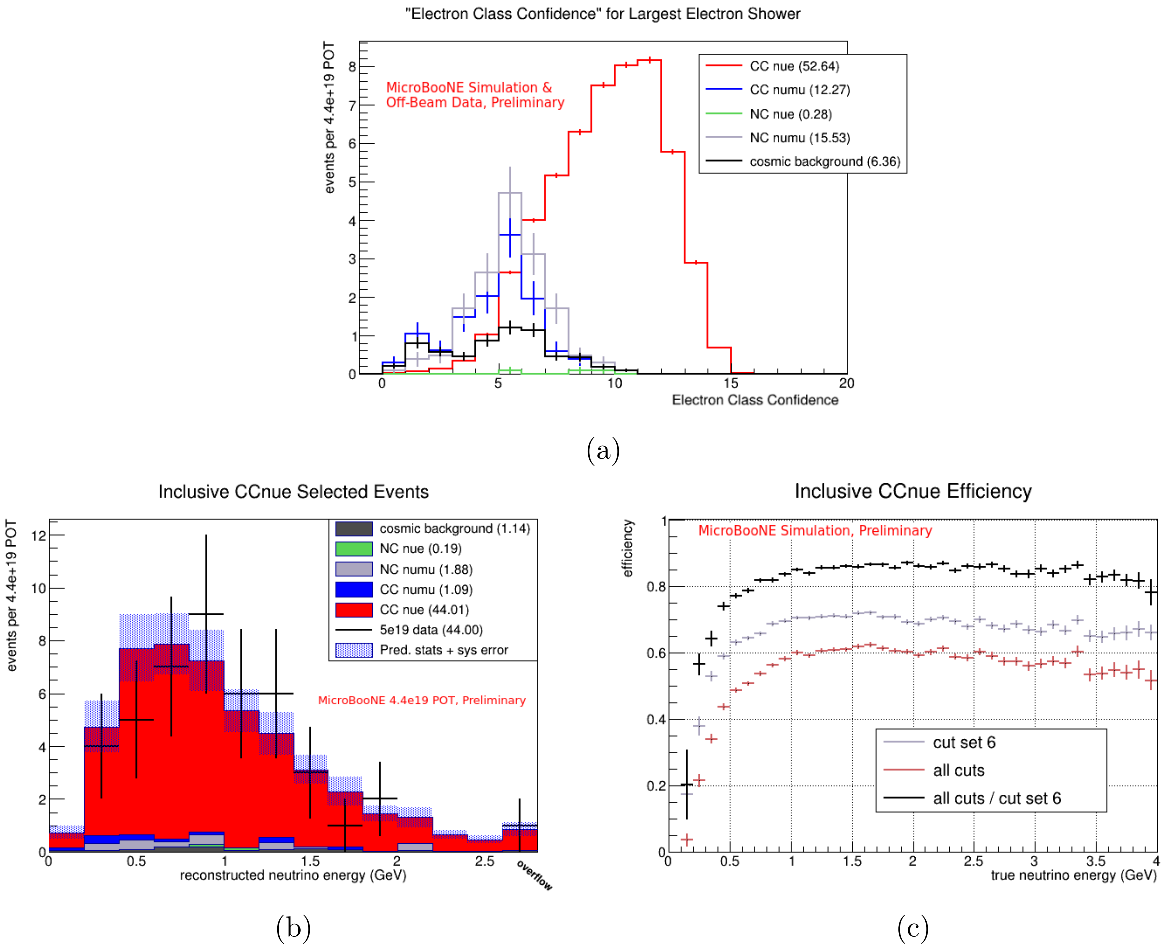


Figure 26: (a): MC and off-beam cosmic background data distributions of the candidate primary electron shower's electron class confidence (LArPID electron score minus average of charged pion and photon scores). The requirement that the primary electron candidate have an electron class confidence value above 7.1 was added to the final CC ν_e cut set. (b): Data/MC reconstructed neutrino energy comparison of events selected by the full CC ν_e cut set with $4.4 \cdot 10^{19}$ POT of MicroBooNE run1 data. (c): MC-predicted CC ν_e efficiency vs. true neutrino energy with all CC ν_e cuts and cut set 6, and the ratio of these two efficiency curves. Cut sets are defined in table IV.

comparison of these results with the inclusive CC ν_e selection of PRD 105:112005 are shown in section III D.

B. CC numu inclusive selection cuts

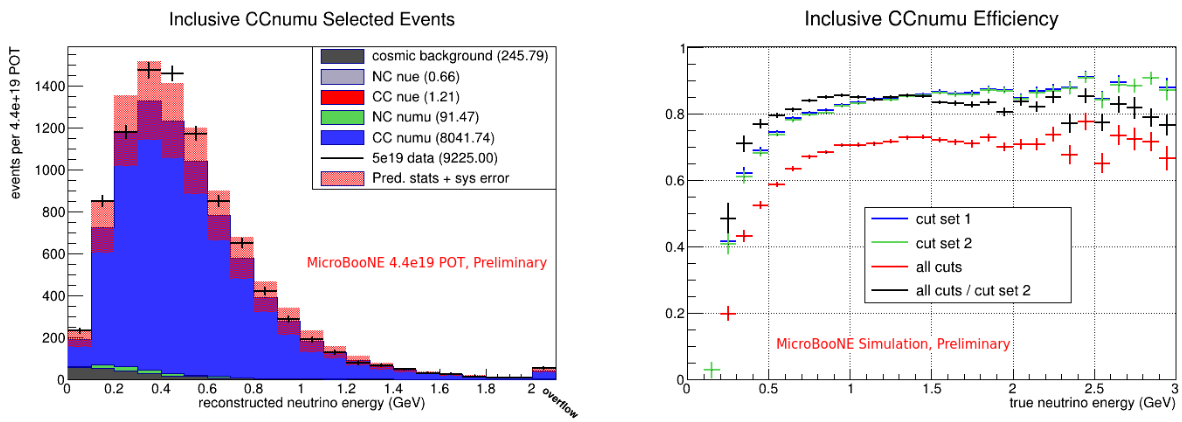
Here, we use a similar approach - taking events with LArMatch-identified neutrino vertex candidates (selecting the one with the highest score if there are multiple), applying the same Wire-Cell cosmic rejection cuts discussed in section III A, and cutting on the LArPID outputs of attached prongs - to achieve a highly effective inclusive CC ν_μ selection. Figure 22a from section III A shows the number of LArPID-identified muon tracks attached to the neutrino candidate for MC CC ν_μ events, cosmic, and (in this context) other simulated neutrino backgrounds (after applying Wire-Cell cosmic rejection cuts). A CC ν_μ dominated sample can be selected by simply requiring that there be at least one identified muon track attached to the LArMatch neutrino candidate vertex. (Note that figures 22b and 22c show results with the CC ν_e requirement of 0 reconstructed muons, not the ≥ 1 condition discussed here for the CC ν_μ selection.) When there are multiple reconstructed muons in true CC ν_μ events, this is generally because of clustering errors in which a small section of the track is reconstructed as a separate cluster. In these cases, according to MC, the track with the highest LArPID muon score is truth-matched to the simulated muon 96.4% of the time, and the identified muon track with the most charge is matched to the true muon 95.7% of the time.

This simple selection yields a predicted overall CC ν_μ purity of 96.0% and efficiency of 67.9%. While significant improvements were not achieved by applying further cuts on the LArPID outputs of the identified primary muon track or other attached prongs, we found that purity can be increased (at the expense of efficiency) by cutting on the neutrino keypoint score of the candidate neutrino vertex, the angle of the muon track (to remove downwards going cosmic-background muons), and the LArPID production process scores of the muon track.

The full set of CC ν_μ selection criteria are enumerated in table V. The predicted selection efficiency as a function of true neutrino energy after applying each cut is shown in figure 27b. As with the CC ν_e selection, the largest impact on efficiency comes from the neutrino vertex reconstruction and the primary lepton identification. Distributions of reconstructed neutrino energy for data and MC (including the 1σ uncertainty band calculated in section III C) with the full selection are shown in figure 27a. The overall data excess seen here is not unique to our CNN-based reconstruction and is consistent with the excess seen in the inclusive CC ν_μ selection of MicroBooNE's Wire-Cell reconstruction [17]. Furthermore, as we will show in section III D, the data and predictions are consistent within quoted uncertainties. Additional data/MC comparisons and a more detailed comparison to the Wire-Cell CC ν_μ selection are presented in section III D as well.

Table V: Inclusive CC ν_μ Selection Cuts

Cut	Notes
LArMatch-identified neutrino candidate vertex found inside the fiducial volume	Added in cut set 1 (included in all cut sets)
3D space points of prongs attached to neutrino candidate do not all overlap with Wire-Cell-tagged cosmics	Added in cut set 2 (included in final set as well)
At least one track attached to the candidate neutrino vertex was identified by LArPID as a muon	Added in final cut set

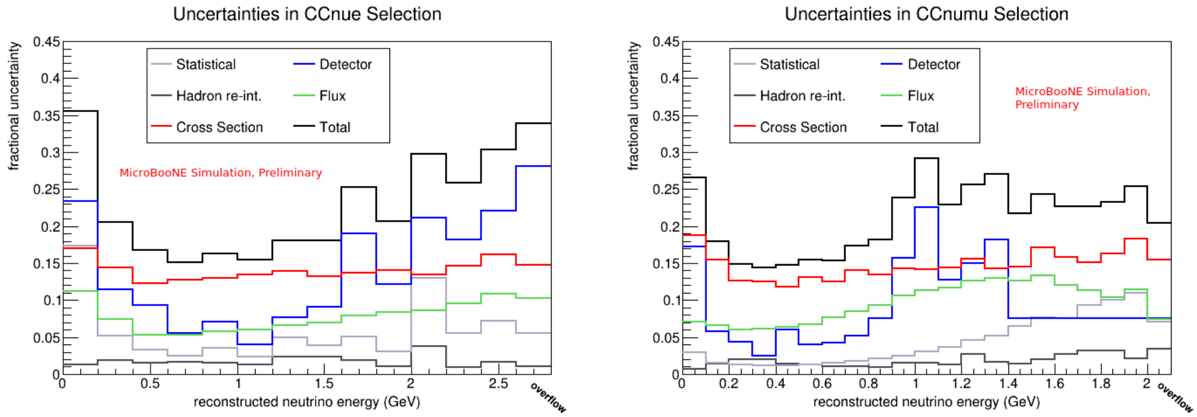


(a) (b)

Figure 27: (a): Data/MC reconstructed neutrino energy comparison of events selected by the full CC ν_μ cut set with $4.4{\cdot}10^{19}$ POT of MicroBooNE run1 data. (b): MC-predicted CC ν_μ efficiency vs. true neutrino energy after adding each CC ν_μ cut, along with a ratio of the full cet set and cut set 2 efficiency curves. Cut sets are defined in table V.

C. Systematic uncertainty estimates

Modeling uncertainties that contribute to our systematic errors come from four main sources: modeling of the neutrino flux, modeling of the MicroBooNE detector, modeling of neutrino-argon cross sections, and modeling of hadron-argon cross sections. To account for detector systematics, we modify a variety of detector parameters, re-simulate a neutrino sample for each variation, and analyze their impact on our predictions. This is discussed in more detail in section III C 1. While these detector variations can change observables in any event, variations in the parameters associated with the other sources of systematic uncertainty simply alter the rate at which different events occur. These flux and cross section uncertainties can therefore be studied by re-weighting individual events without the need to re-simulate new neutrino samples. This method and our flux and cross section systematics



(a) (b)

Figure 28: Fractional uncertainties in each reconstructed neutrino energy bin for the inclusive (a) CC ν_e selection and (b) CC ν_μ selection. Statistical uncertainties on the cosmic background + neutrino predictions; systematic uncertainties from our modeling of the detector, flux, neutrino-argon cross sections, and hadron-argon interactions; and the total combined uncertainty is shown. The use of a flat detector systematic uncertainty in the CC ν_μ selection above 1.4 GeV is discussed in section III C 1.

are discussed in more detail in section III C 2.

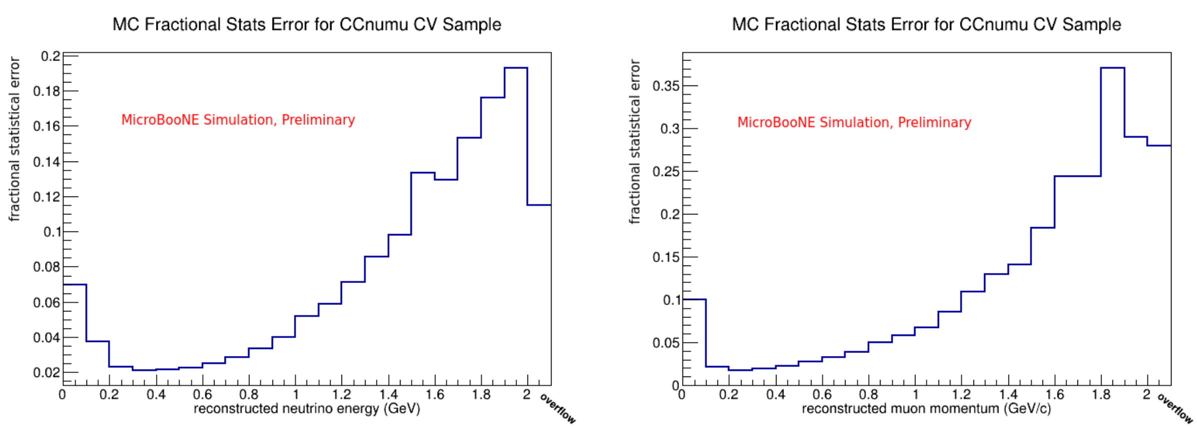
The results of those studies is summarized in figure 28, which shows uncertainties in our predicted event counts in each reconstructed neutrino energy bin used in the inclusive CC ν_e and CC ν_μ selections. Our total uncertainty in each bin is shown, along with contributions from the four sources of systematic uncertainty discussed above and statistical errors from our finite cosmic background and simulated neutrino samples.

1. *Detector Systematic Uncertainties*

To account for uncertainties in detector modeling, we vary parameters associated with the light yield (LY), light attenuation, and Rayleigh scattering length; "wiremod" modifications to the amplitudes and widths of wire waveforms as a function of x position, (y,z) position, and angles θ_{xx} and θ_{yz} of particle trajectories; a variation in electron-ion recombination parameters ("recomb2"); and an alternative electric field inside the TPC from the space charge effect (SCE). See e.g. [14] for additional information on these variations.

For each variation, we re-simulate the same sample of Monte Carlo neutrino events and calculate a covariance matrix for each kinematic observable to quantify the bin-by-bin shift in event counts: $F^k_{ij} = (N^k_i - N^{CV}_i)(N^k_j - N^{CV}_j)/(N^{CV}_i N^{CV}_j)$, where F^k is the fractional covariance matrix for the kth variation, N_i^k is the number of events in bin i of the simulation with variation k, and N_i^{CV} is the number of events in bin i for the central value simulation with default detector parameters. Overall 1σ detector-related fractional uncertainties in each bin are then given by $\sigma_i/N_i = \sqrt{\sum_k F_{ii}^k}$.

The computational expense associated with re-simulating neutrino events for each variation presents a significant challenge in quantifying these detector systematics. The $O(10^5)$ event samples used in this analysis provided inadequate statistics to provide a robust estimate of detector uncertainties in certain regions. For each variation, two simulations were combined to predict event counts: one ν_μ dominated sample in which neutrinos are simulated in the same proportion estimated to occur in the beam, and one involving only the intrinsic CC ν_e component. While the intrinsic ν_e simulation provided adequate statistics for CC ν_e predictions, roughly just 10 raw neutral current and ν_μ background events (there are small variations between the different simulated samples) from the former beam simulation passed our CC ν_e selection. Statistical variations from this background prediction therefore caused large artificial fluctuations in our estimated detector systematic uncertainties. In the CC ν_μ selection, statistics are also very low in the high energy tails of the reconstructed neutrino energy and muon momentum distributions (see figure 29), again causing large fluctuations in estimated systematics in those regions.



(a) (b)

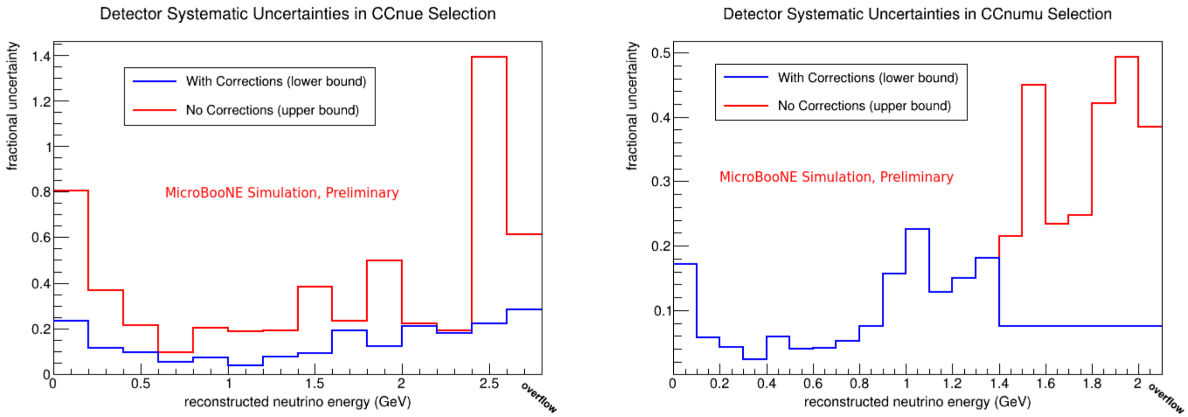
Figure 29: Monte Carlo statistical errors on events passing the CC ν_μ selection for the central value simulation used to estimate detector systematics, binned in reconstructed neutrino energy (a) and muon momentum (b).

We will address these statistical shortcomings prior to publishing these results by re-processing one-order-of-magnitude larger samples for these detector variations. As a temporary solution, we place a lower bound on our detector systematics with the following corrections. For the CC ν_μ selection, we combine events above 1.4 GeV in reconstructed neutrino energy and above 1.2 GeV/c in reconstructed muon momentum into a single bin for those respective distributions. When calculating our total uncertainty in bins above those thresholds (as in figure 28b), we use the flat high energy/momentum detector systematics obtained with this approach. For the CC ν_e selection, we assume that the kinematic dependence of $N_i^k - N_i^{CV}$, the central value vs. detector variation excess in each bin of all kinematic distributions, is the same for the NC and ν_μ backgrounds that pass the selection as it is for the CC ν_e signal events that pass the selection. With this assumption, we estimate the predicted signal + background event counts by scaling the CC ν_e signal distributions by $(N_s + N_b)/N_s$, where N_s and N_b are the total number of signal and background events that pass the selection, respectively.

Our total detector systematic uncertainties in each reconstructed neutrino energy bin of the inclusive CC ν_e and CC ν_μ selections with and without the low-statistics corrections discussed above is shown in figure 30. The uncertainties without the corrections provide a lower bound for our estimated detector-related modeling errors, while the larger, statistical-fluctuation-driven uncertainties without the corrections provide an upper bound. To avoid inflating our detector systematics as a result of low-statistics fluctuations and provide a more strict test on data / Monte Carlo consistency in our selection results (discussed in section III D), we use the lower-bound detector uncertainties with the statistical corrections discussed above. The full fractional covariance matrix (binned in reconstructed neutrino energy) for all detector variations with these low-statistics corrections is shown in figure 31.

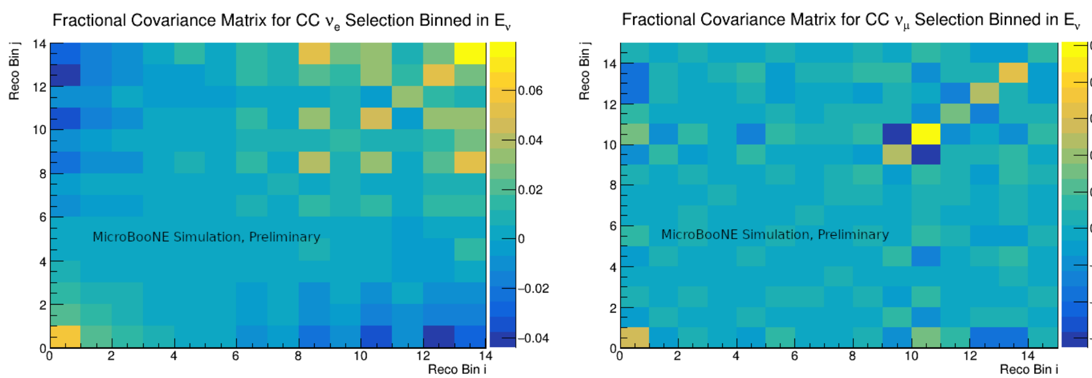
2. Flux, Cross Section, and Hadron Re-Interaction Uncertainties

To calculate systematic uncertainties arising from flux, neutrino cross section, and hadron re-interaction predictions we employ the same method outlined in [14]. Flux uncertainties arise from three main sources: the properties of the magnetic focusing horn, hadron production in the target, and secondary hadron interactions. Neutrino cross-section uncertainties arise from a large number of parameters associated with each neutrino interaction mode and final state interactions that affect all modes. The hadron re-interaction uncertainty calculations consider variations in parameters associated with hadron-argon cross sections



(a) (b)

Figure 30: Total detector variation uncertainties in each reconstructed neutrino energy bin of the inclusive (a) CC ν_e and (b) CC ν_μ selections. Results are shown with and without the low-statistics corrections discussed in the text.

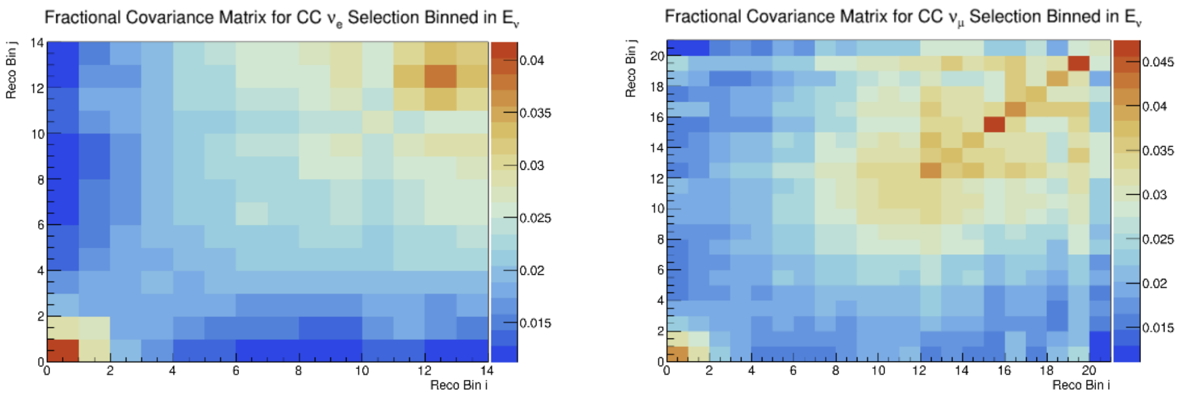


(a) (b)

Figure 31: Fractional covariance matrices binned in reconstructed neutrino energy from all detector variations for the inclusive (a) CC ν_e and (b) CC ν_μ selections.

for protons and charged pions. These uncertainties are accounted for by re-weighting events with each systematic parameter variation and comparing the modified reconstructed spectra with the nominal simulation. Additional details on the variations considered are provided in [14].

Given the reconstructed spectra with bin counts N_i for each set of varied parameters, a covariance matrix M can be constructed where the variance in bin counts (resulting from the parameter variations) is provided in the diagonal entries and the covariance between the counts in each pair of bins in the off-diagonal entries. The fractional covariance matrices $F_{ij} = M_{ij}/(N_i N_j)$ including all flux, cross section, and hadron re-interaction variations for



(a) (b)

Figure 32: Fractional covariance matrices binned in reconstructed neutrino energy from all flux, cross section, and hadron re-interaction variations for the inclusive (a) CC ν_e and (b) CC ν_μ selections.

the reconstructed neutrino energy spectra in the inclusive CC ν_e and CC ν_μ selections are shown in figure 32.

As shown in figure 28, cross section uncertainties provide the largest contribution to our model-based systematics. Flux uncertainties are substantial as well; they are roughly half the size of cross section uncertainties at lower energies and larger at higher energies. Hadron re-interaction systematics contribute very little to our overall uncertainty.

D. Results

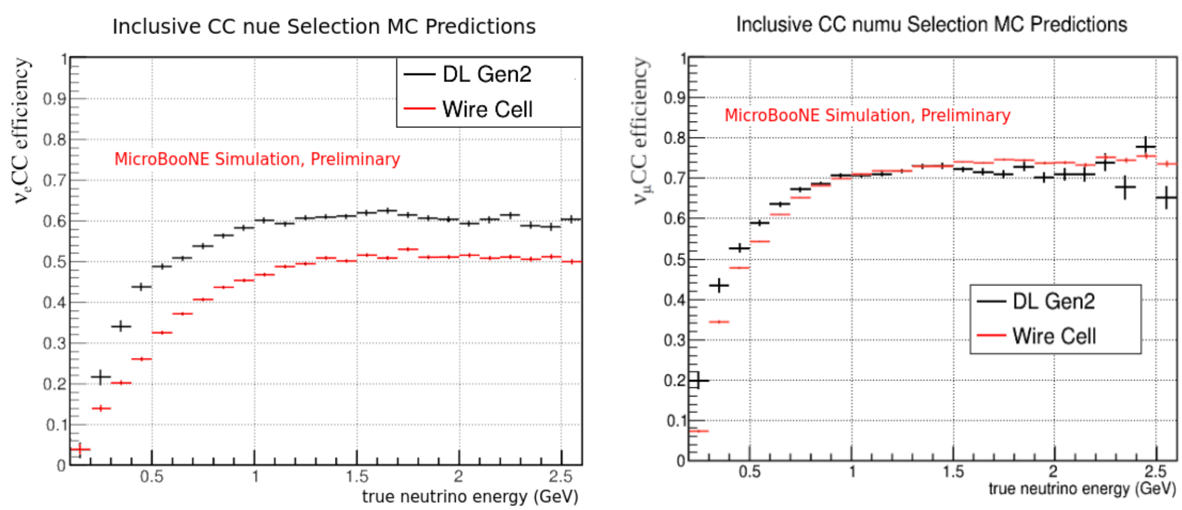
The performance of the inclusive CC ν_e and CC ν_μ selections demonstrate the effectiveness of this new CNN-based reconstruction. Our predicted selection efficiencies and purities outperform PRD 105:112005 [17], the highest-efficiency result previously published by MicroBooNE. This comparison is shown in table VI.

For the inclusive CC ν_e selection, our deep-learning based reconstruction provides significantly higher predicted purities (91% compared to 82%) and efficiencies (57% compared to 46%). This amounts to a predicted 24% increase in the number of CC ν_e events selected with significantly lower background. As shown in figure 33a, an improvement in efficiency is achieved across all true neutrino energies (except in the lowest 100-200 MeV bin, in which there is no statistically significant difference). For the inclusive CC ν_μ selection, we achieve the same overall efficiency as PRD 105:112005 but with a reduced background (purity of

96% compared to 92% from PRD 105:112005). However, as shown in figure 33b, the two analyses provide different efficiencies at different neutrino energies, with our reconstruction yielding a higher efficiency below 1 GeV and the analysis of PRD 105:112005 providing a higher efficiency above 1.5 GeV.

	DL Reco	PRD 105:112005
CC ν_e Selection Efficiency	57%	46%
CC ν_e Selection Purity	91%	82%
CC ν_μ Selection Efficiency	68%	68%
CC ν_μ Selection Purity	96%	92%

Table VI: Inclusive CC ν_e and CC ν_μ selection results for our deep-learning-based reconstruction and PRD 105:112005 [17].



(a) (b)

Figure 33: The predicted CC ν_e (a) and CC ν_μ (b) selection efficiency of our reconstruction and selection (DL Gen2) and that of PRD 105:112005 (Wire-Cell) [17] as a function of true neutrino energy.

In sections III A and III B, we showed predicted (using MC neutrino simulations plus data cosmic-ray background) and MicroBooNE open data distributions of reconstructed neutrino energy for selected inclusive CC ν_e and CC ν_μ events. Figures 34 and 35 show the same data/MC comparisons for reconstructed primary lepton momentum and $\cos(\theta_l)$ (where θ_l is the angle between the primary lepton and the beam). Also included are the reconstructed neutrino energy distributions, along with predicted purity and efficiency as a function of reconstructed neutrino energies. For both selections, purities are high for all

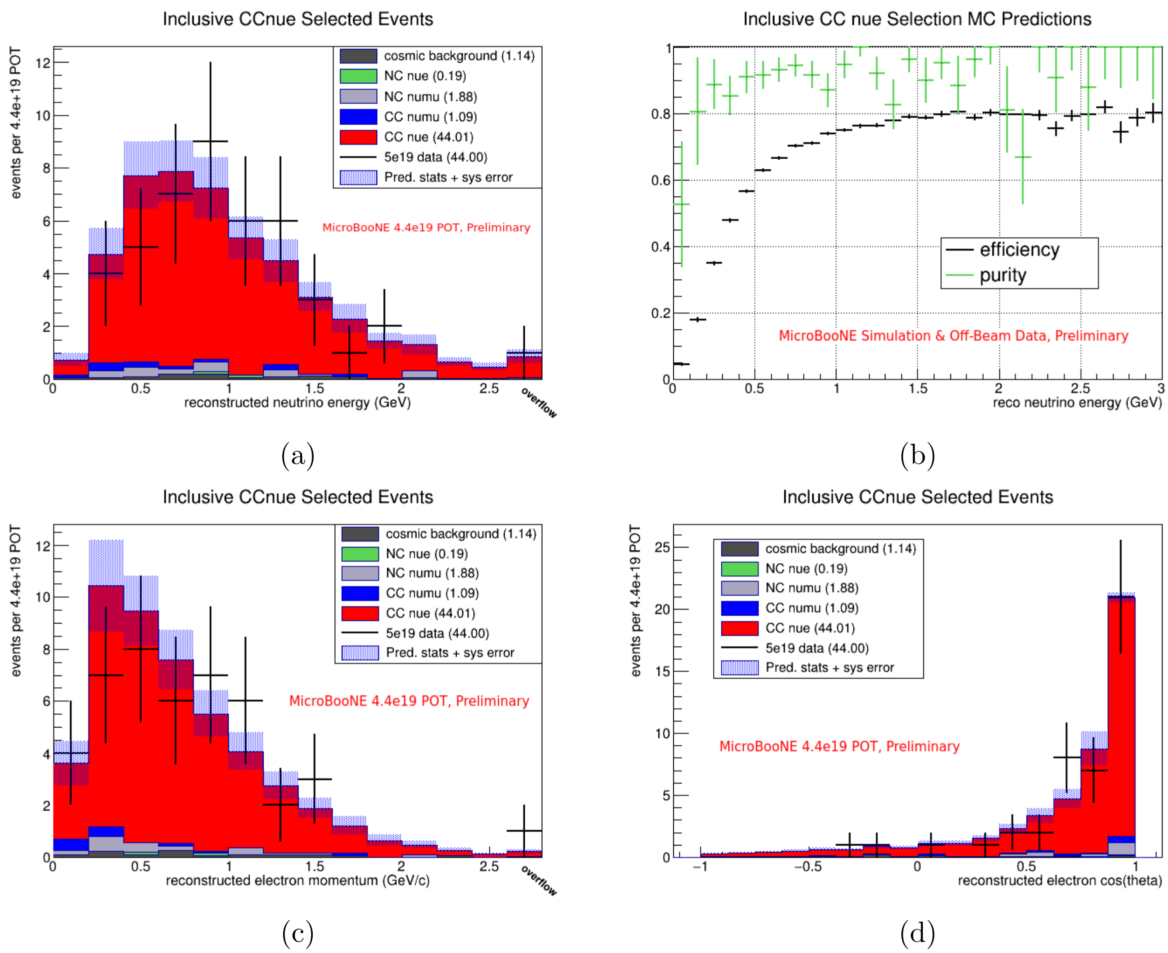


Figure 34: Predicted and MicroBooNE open data distributions of events passing the CC ν_e selection in (a) reconstructed neutrino energy, (c) reconstructed electron momentum, and (d) reconstructed $\cos(\theta_e)$, where θ_e is the angle between the reconstructed electron shower and the beam. (b): The predicted efficiency (from MC) and purity (from MC and off-beam cosmic background data) of the CC ν_e selection as a function of reconstructed neutrino energy.

energies, whereas efficiency drops more dramatically, as expected, at low energies where it is more difficult to separate signal from background.

To assess the consistency between the predictions made for these distributions with our deep-learning based reconstruction framework and observations from the MicroBooNE open data set, we employ a χ^2 goodness-of-fit test using the combined Neyman-Pearson (CNP) χ^2 test statistic [36] with the covariance matrix formalism:

$$\chi^2 = (M - \mu)^T \cdot V_{\text{full}}^{-1} \cdot (M - \mu) \tag{6}$$

where M and μ are vectors of the observed and predicted event counts in each bin and V_{full}

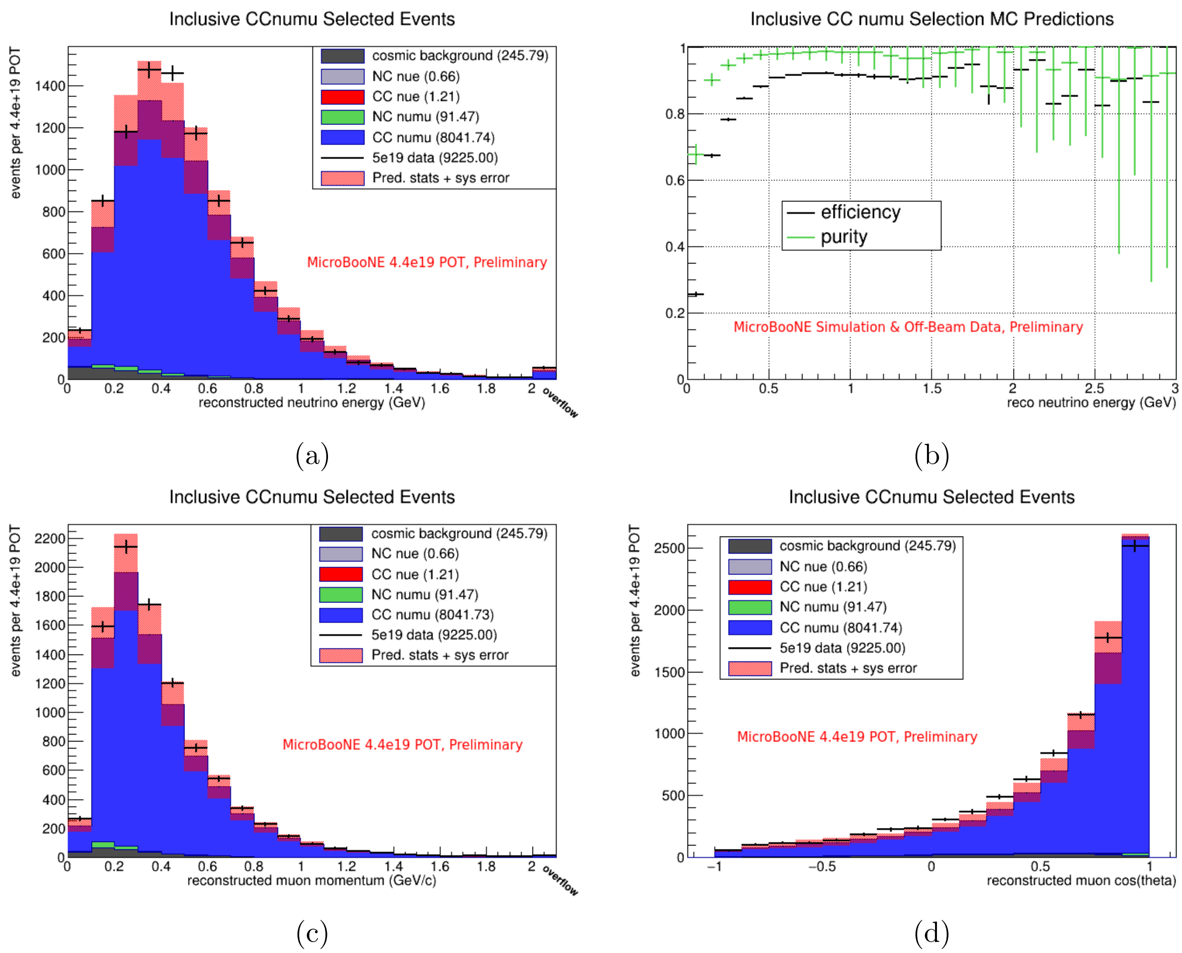


Figure 35: Predicted and MicroBooNE open data distributions of events passing the CC ν_μ selection in (a) reconstructed neutrino energy, (c) reconstructed muon momentum, and (d) reconstructed $\cos(\theta_\mu)$, where θ_μ is the angle between the reconstructed muon track and the beam. (b): The predicted efficiency (from MC) and purity (from MC and off-beam cosmic background data) of the CC ν_μ selection as a function of reconstructed neutrino energy.

is the full covariance matrix in the CNP method:

$$V_{\text{full}} = V^{\text{stat}}_{\text{CNP}} + V^{\text{stat}}_{\text{pred}} + V^{\text{sys}}_{\text{flux}} + V^{\text{sys}}_{\text{xsec}} + V^{\text{sys}}_{\text{det}} \tag{7}$$

This covariance matrix is constructed from the flux and neutrino-argon and hadron-argon cross section covariance matrices ($V^{\text{sys}}_{\text{flux}}$ and $V^{\text{sys}}_{\text{xsec}}$) discussed in section III C 2, the detector systematics covariance matrix ($V^{\text{sys}}_{\text{det}}$) from section III C 1, a diagonal matrix containing the variance in each bin from uncertainties arising from the finite statistics used to make predictions ($V^{\text{stat}}_{\text{pred}}$), and a diagonal matrix containing the CNP terms: $(V^{\text{stat}}_{\text{CNP}})_{ii} = 3/(1/M_i + 2/\mu_i)$. As discussed in section III C 1, we altered the binning used to calculate $V^{\text{sys}}_{\text{det}}$ (and therefore changed its dimensions) for the reconstructed neutrino energy and muon momentum distributions of the CC ν_μ selection by combining high energy/momentum bins. Here, when constructing $V_{\text{det}}^{\text{sys}}$ in Eq. 7 for those distributions, we estimate the covariance between individual bins in the overflow region and bin i as the covariance between the overflow bin and bin i. This maintains the approach of section III C 1 to (temporarily, pending processing of higher statistics detector variation samples) address low statistics with lower-bound estimates that provide a stricter test for data/MC consistency tests.

By comparing the χ^2 from Eq. 6 with the distribution of a χ^2 with N degrees of freedom (where N is the number of bins), we can calculate a p-value for our observations (the probability of seeing the observed or a more extreme fluctuation) and assess data / Monte Carlo consistency. However, the Gaussian assumptions used in the covariance matrix formalism followed here break down at low statistics, where much of the Gaussian probability distributions fall into the un-physical region of negative bin counts. While this is not an issue for the CC ν_μ selection where predicted bin counts are sufficiently high across all bins of each distribution, in the CC ν_e selection, predicted bin counts are almost all below 10 and fall below 1 in the tails. To make the CC ν_e statistics closer to the Gaussian assumption, we combine all bins with a predicted event count below 2 for these goodness-of-fit tests. The rebinned CC ν_e distributions and the full covariance matrices (equation 7) for all distributions in both selections can be found in appendix A.

The χ^2 and associated p-values calculated with this method for the reconstructed neutrino energy, lepton momentum, and lepton $\cos(\theta)$ distributions for both selections is shown in table VII. All p-values are high: close to or above 90% for most tests and no lower than 24.4%. This indicates that the open data observations are consistent with our predictions and is evidence, for the kinematic variables considered, of a lack of any concerning data / Monte Carlo domain shift introduced by our deep-learning based reconstruction algorithms.

	CC ν_e Selection, E_ν Binning	CC ν_e Selection, p_{e^-} Binning	CC ν_e Selection, $\cos(\theta)$ Binning	CC ν_μ Selection, E_ν Binning	CC ν_μ Selection, p_μ Binning	CC ν_μ Selection, $\cos(\theta)$ Binning
χ^2/DOF	3.80/9	3.06/8	5.18/6	25.08/21	11.91/21	9.73/16
p value	0.924	0.931	0.521	0.244	0.942	0.880

Table VII: Goodness of fit test results: χ^2 / degrees of freedom and associated p values for the reconstructed neutrino energy, lepton momentum, and lepton $\cos(\theta)$ distributions in the inclusive CC ν_e and CC ν_μ selections.

E. Results of Data and MC comparison using Open Data Sample

To further test our predicted inclusive CC ν_e selection results and comparison to PRD 105:112005 [17], we manually hand scanned all MicroBooNE run1 open data events selected by our analysis and that of PRD 105:112005, classifying each as either CC ν_e or background. This open data set contains $4.4{\cdot}10^{19}$ POT of run 1 data.

In the analysis of PRD 105:112005, 40 events were selected, of which (according to our hand scans) 37 or 38 were true CC ν_e interactions. In our analysis, 44 events were selected, of which 42 or 43 were true CC ν_e interactions (there is some uncertainty in the hand scan classifications). These results, along with comparisons to predictions made in the previous section, are summarized in table VIII.

	DL Reco Data Hand Scan Estimate	DL Reco MC Prediction	PRD 105:112005 Data Hand Scan Estimate	PRD 105:112005 MC Prediction
Total Events	44	48.3	40	41.2
Signal Count	42 - 43	44.0	37 - 38	33.8
Background Count	1 - 2	4.3	2 - 3	7.3
Purity	95% - 98%	91%	93% - 95%	82%

Table VIII: Hand scan results of inclusive CC ν_e events selected from the open data sample by our deep-learning-based reconstruction and PRD 105:112005 [17].

These results are generally consistent with our expectations, given the large statistical uncertainties with these small numbers of events. With 4 - 6 more probable signal events selected by our framework, these results support our prediction of an increase (compared to PRD 105:112005) in inclusive CC ν_e selection efficiency. Additionally, while the *total* number of selected probable signal events was higher by 4 - 6, there were 11 probable signal events selected by our framework that were not present in the analysis of PRD 105:112005 and 6 probable signal events found in PRD 105:112005 that did not appear in our selection. This indicates that combining the events selected by our framework and PRD 105:112005 (which utilizes the Wire-Cell reconstruction [17–19]) could yield an even more substantial improvement in efficiency, a promising avenue for future work.

The three hand-scan-classified CC ν_e events lowest in reconstructed neutrino energy are shown in figures 36 - 38.

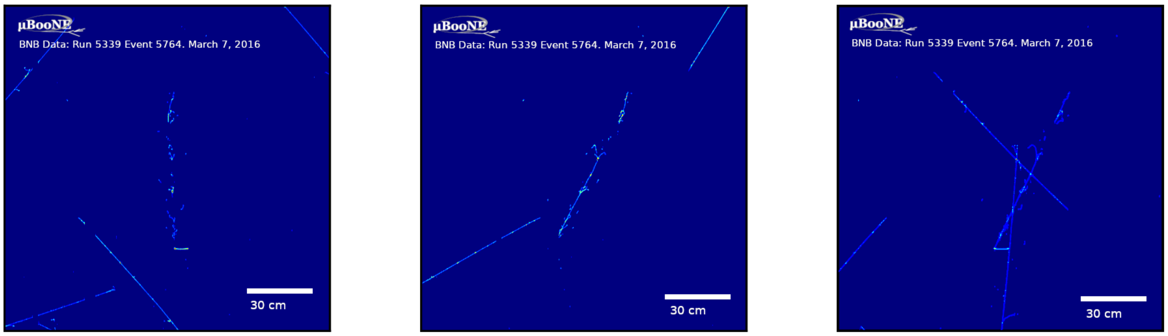


Figure 36: Wire plane images of a probable CC ν_e event selected by our analysis from the MicroBooNE open data set.

Figure 37: Wire plane images of a probable CC ν_e event selected by our analysis from the MicroBooNE open data set.

IV. DISCUSSION

A new reconstruction workflow has been developed that utilizes three convolutional neural networks to perform pattern recognition relatively early in the reconstruction workflow. This leverages the powerful ability of CNNs to recognize features in low-level data, specifically, the image-like data produced by the LArTPC wire planes. The many outputs and the ability to partition the spacepoints into topological classes greatly simplified the reconstruction of 3D spacepoints. Though this is also in part due to the energy range of interactions for

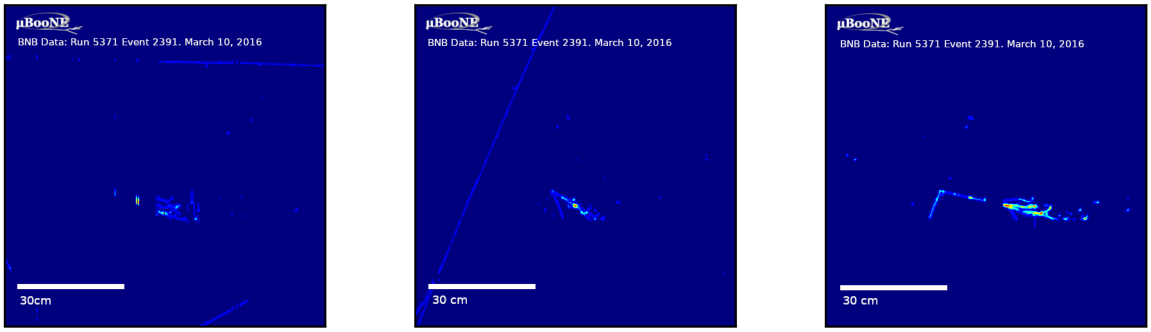


Figure 38: Wire plane images of a probable CC ν_e event selected by our analysis from the MicroBooNE open data set.

MicroBooNE. With median energies below 1 GeV, the final state particles emerging from neutrino interactions often do not overlap. Final state particles more often than not emerge from the vertex at well-separated angles – which adds to the advantage working in 3D space has for clustering. However, we believe the utility of feature recognition by 2D CNNs are on display given the modest complexity of the non-DL algorithms implemented in this work. Furthermore, the particular use of the LArPID CNN mitigates the impact of mistakes made in the 3D reconstruction. This comes from the use of information about both the particle under consideration and the entire image of the interaction. We believe one lesson that should carry over to analyses being built for future LArTPCs like SBND and DUNE, is that access to the 2D image information will have much utility.

The quality of the reconstruction and the utility of the CNN outputs were tested through the exercise of selecting inclusive charged-current ν_μ and ν_e interactions. We looked for potentially large, show-stopping domain shift effects by testing the selection on the MicroBooNE open dataset. The cuts employed are deceptively simple in that they are flat cuts on particle ID scores or on the numbers of certain particles. They are deceptively simple, because in other MicroBooNE selections BDTs, using a collection of kinematic observables and spatial patterns, was often required to reach the best efficiency and/or purity. However, the CNNS used in this work, we conjecture, likely utilizes the same kinematic correlations extracted directly from the 2D images and contributes to the improvement in ν_e-CC efficiency across a large range of true neutrino energies.

Interestingly, this improvement in the ν_e-CC selection constrasts with the roughly equal performance between this work and the inclusive ν_μ-CC selection of PRD 105:112005 [17] utilizing the Wire-Cell reconstruction. In the CC ν_e selection, we leverage the LArPID scores to further reduce backgrounds by vetoing events with an identified muon or cutting events with sufficient evidence that the primary shower either derived from charged pions or is a photon. For the inclusive CCν_μ event selection, there was not a set of single particles associated strongly with potential background interactions. Instead, improvement in the signal acceptance or rejection of background will likely need to come through the use of correlations between particle kinematics or tuning of cuts in different regions of kinematic phase space. We leave explorations to improve the selections using the reconstructed particle kinematics to future work. For now, our conjecture is that the inclusive CCν_μ selection is primarily defined by the upstream Wire-Cell cosmic tagging algorithms. The fact that our efficiency and purity are similar possibly reflects a similar ability to identify the muon within the in-time charge cluster.

Further conjecture is based in part on the last two cuts applied in the ν_e-CC selection. These cuts can be interpreted as examples of how CNNs utilize fine image details to great effect. The last two cuts both target the separation of primary electrons from secondary electrons coming from the decay of a low-energy muon or charged pion. In these cases, the information to complete the tasks is located in small regions of the image: the beginning of showers and around the vertex. The utilization of this information is what we hypothesize to be the source of the efficiency and purity gains. For example, there can be difficult edges cases when estimating dE/dx for particle ID. One such case are the potential presence of additional particles near the vertex. These particles can be of a low enough energy such that they are missed by the reconstruction, but high enough in energy to impact estimates like the dE/dx for identified trajectories. Specific scenarios include a short co-linear proton or a localized region of high energy deposition from a Brem photon emitted early in the trunk of the shower. Both can push a dE/dx estimate to mis-identify an electron shower as photon. We conjecture that the LArPID network is able to learn a set of image features that can detect these edge cases and influence the electron PID. Demonstrating this is one area for future study. But one piece of circumstantial evidence is the lack of dependence on the energy scale of the neutrino interaction. In this hypothesis, the occurrence of such scenarios are broadly distributed across the range of neutrino energies. The possible lack of comparable scenarios for identifying muons would then explain why improvements are not observed for the ν_μ selection.

Inspecting the distributions of the LArPID score for the ν_e selection in figure 26, the most similar distribution to the CCnumu events that remain are the NCnumu events. Cosmic interactions also peak in a similar region. As discussed above, the electron confidence score was developed to ID interactions with evidence that the candidate electron shower is actually a secondary electron from a decay muon or charged pion. For NC events with true neutrino energies near the peak of 800 MeV, the final state momentum for the charged pions is likely small. Alternatively, a charged pion with an early decay in flight leads to what is now easily misinterpretable as an event with a primary muon in the final state. But in addition to such visual evidence, the LArPID CNN likely has learned to use particle kinematics better separate CC ν_e events from the various backgrounds. Such kinematic information is also the kind of information that would be effectively exploited by a BDT-based selection – as was used in the 2022 inclusive search of PRD 105:112005 [17]. At the high-level observable distributions studied, we do not find evidence that the CNN-based reconstruction and selection are more sensitive to argon-interaction modeling uncertainties when compared to past analyses. But future work will focus on how to dig deeper into this potential bias.

However, as with all machine learning methods, we must vet the CNNs presumed ability to correlate latent physical quantities or scenarios to distributions over possible images. The goal of this work is to report on the completed workflow, and provide evidence of its competitiveness to past analyses. But more work is ongoing to evaluate the robustness of this analysis centered on CNNs. The model learning to recognize the features discussed above derives from training on our simulation data. This data is produced using models of the MicroBooNE detector and physics like the ionization produced by charged particles. Though not perfect, one might hypothesize that the level of detector mis-modeling here is at a manageable level. This is supported by the level of change in the number of selected events. Even with the higher bound estimates, which we believe are likely due to low MC statistics, the uncertainties from detector-related effects are similar to the analysis of PRD 105:112005 [17]. Future work can also be done to directly address certain aspects of this type of domain shift such as adversarial training.

While the gains in the ν_e might come from what has been discussed, the context images also allow the use of correlations between particle frequency and particle kinematics. The underlying correlations in the training data, in this case, come from neutrino-argon interaction modeling, which has uncertainties larger relative to the physics discussed before. Future work will aim at understanding the degradation of performance of the LArPID network when confronted with images that contain particles with kinematic correlations some distance outside of the support of the training data. This means that wildly out-of-domain examples could simply be ignored by the models. Developments in anomaly detection are one of several directions to research how to improve model robustness or detect issues related to exotic final states or unusual particle kinematics. There is also the area of domain adaption which aims to find ways to improve robustness.

CNN models, such as the LArMatch keypoint model, will also be impacted by the use of training images simulating a LArTPC at the surface. In such a detector, there will be a high rate of cosmic interactions in each image. For MicroBooNE, this was approximately 10-15 interactions. Generally, our cosmic simulations left more interactions per image than was seen in the data. Mismodeling cosmic interaction rates are important for several backgrounds, in particular to interactions with low-energy showers. Such backgrounds include stopping muons entering near the cathode and leaving a short track and Michel electron, entering photons, and those produced from hadronic interactions, e.g. from the decay of neutral pions. As a potential impact, the neutrino vertex finder scores in this context could be sensitive to the relative rate of neutrino-induced and cosmogenic single-shower events.

Another topic of discussion is what the potential impact this work might have in the future for both DL and non-DL reconstruction in LArTPCs. For one, the LArPID strategy of using contextual information around a defined cluster is readily adaptable to existing MicroBooNE Pandora and Wire-Cell analyses. One might speculate that the differences in what amounts to the LArPID image pre-processing step for the particle cluster image will have a limited impact on the LArPID behavior. Future work includes plans to investigate LArPID integration into these existing workflows.

Many of the hand-engineered algorithms in this work are very simple in their core approaches, but might require heuristics to tune their behaviors and/or handle edge cases. This leads to several parameters per algorithm and an overall large number of parameters affecting the behavior of the reconstruction. Such parameters for each algorithm were tuned during development on a relatively small MC sample size, O(100) events, in order to iterate and tune these parameters on a day-length time-scale. In contrast, for the DL-based reconstruction framework in Ref. [11] (referred to as "mlreco3d"), these algorithms have ML-based counterparts which one might expect to perform better just due to the simple fact that the algorithms are learned by optimizing them over the entire available training data. The mlreco3d framework centers around a 3D voxel representation of the LArTPC data, similar to the use of spacepoints in this work. In particular, the task of forming subclusters and then collecting them into particle candidates is addressed through the use of graph neural networks (GNNs). GNNs are likely much more accurate than our shower reconstruction, which is merely a simple cone-based aggregator. Indeed, the purity versus completion plots for the electron clusters are the least accurate for our workflow. Similarly, in mlreco3d associations of particle candidates to potential neutrino interactions are also done with graphs. Furthermore, mlreco3d's determination of keypoints and voxel-wise particle labels make use of the 3D structure in a more direct way than in LArMatch, which relies only on 2D image features correlated across the wire planes.

In contrast to the 3D spacepoint algorithms, the CNN components in our work would be the component that would best integrate with a fully ML framework such as mlreco3d. Indeed, the LArMatch real/ghost classifier was developed to provide a CNN-based preprocessing step leading into the mlreco3d pipeline. One direction is to investigate if improvements could be made by injecting the spacepoint feature vectors into key parts of the mlreco3d framework. Because the pipeline fully leaves behind the 2D low-level data, one might believe that LArMatch's image feature vector can be used to preserve useful details otherwise lost when moving the representation of the data from 2D images into 3D voxels. Furthermore, one would also expect that individual particle clustering will not be perfect. And, therefore, a LArPID-like stage will be useful in similar ways to the reconstruction and selection described in this work.

V. CONCLUSIONS

This work represents a milestone in the development of ML tools for LArTPC analysis. We demonstrate – for the first time on real LArTPC data – a deep-learning based generic neutrino interaction reconstruction framework that is competitive with the current state-of-the-art: The inclusive CC ν_e and CC ν_μ selections obtained with the outputs of our reconstruction compare favorably to the highest-efficiency results previously published by MicroBooNE [17], with reduced backgrounds and a predicted 24% increase in the number of selected CC ν_e events. Hand scans of selected CC ν_e events from a small MicroBooNE open data set are consistent with these predictions. These results demonstrate the power of CNNs to leverage the full set of information provided in LArTPC wire-plane images at multiple stages of the framework: in low-level reconstruction of vertices and 3D space points (LArMatch), tagging pixels as track or shower like to aid in downstream clustering algorithms (SSNet), and analyzing reconstructed 3D prongs with the aid of full wire-plane images to fold in-context information that may have been lost by inaccuracies in upstream algorithms (LArPID).

A possible downside of our approach stems from the black-box nature of these deep networks and their potential to introduce biases from the use of supervised learning on simulated data. While a more thorough investigation of network-based systematic uncertainties and model interpretation studies will be the subject of future work, we have demonstrated that simulated MC distributions of high-level reconstructed kinematic variables for selected charged-current neutrino interactions are consistent with data. This provides evidence of the robustness of our framework and a lack of highly significant data/MC domain shifts introduced by the use of CNNs trained on simulated data.

These results show promise for the deep-learning based reconstruction tools developed here to improve the sensitivity of LArTPC physics analyses. Future studies will employ this reconstruction framework in cross-section measurements and new physics searches. In the near term, individual tools within the framework could be quickly integrated into alternative reconstruction packages. The LArPID network, for example, could easily be run over 3D tracks and showers reconstructed by Wire-Cell or other frameworks. As the use of such tools in high energy physics analyses proliferates, this work contributes towards understanding the power and robustness of computer vision techniques when applied to LArTPC neutrino data. It also points towards the improvements these methods can make on the physics that will come out of future LArTPC experiments, specifically from the short-baseline experiments over the next few years.

[1] R Acciarri et al. (MicroBooNE Collaboration). Design and construction of the microboone detector. *Journal of Instrumentation*, 12(02):P02017, 2017. URL `https://dx.doi.org/10.1088/1748-0221/12/02/P02017`.

[2] R Acciarri et al. (ICARUS-WA104, LAr1-ND, and MicroBooNE Collaborations). A proposal for a three detector short-baseline neutrino oscillation program in the fermilab booster neutrino beam. *arXiv:1503.01520*, 2015. URL `https://doi.org/10.48550/arXiv.1503.01520`.

[3] S Amerio et al. (ICARUS Collaboration). Design, construction and tests of the icarus t600 detector. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 527(3):329–410, 2004. URL `https://doi.org/10.1016/j.nima.2004.02.044`.

[4] B Abi et al. (DUNE Collaboration). Volume i. introduction to dune. *Journal of Instrumentation*, 15(08):T08008, 2020. URL `https://dx.doi.org/10.1088/1748-0221/15/08/T08008`.

[5] B Abi et al. (DUNE Collaboration). The single-phase protodune technical design report. *arXiv:1706.07081*, 2017. URL `https://doi.org/10.48550/arXiv.1706.07081`.

[6] D A Dwyer, M Garcia-Sciveres, D Gnani, C Grace, S Kohn, M Kramer, A Krieger, C J Lin, K B Luk, P Madigan, C Marshall, H Steiner, and T Stezelberger. Larpix: Demonstration of low-power 3d pixelated charge readout for liquid argon time projection chambers. *Journal of Instrumentation*, 13(10):P10007, 2018. URL `https://dx.doi.org/10.1088/1748-0221/13/10/P10007`.

[7] R Acciarri et al. (MicroBooNE Collaboration). Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber. *Journal of instrumentation*, 12 (03):P03011, 2017. URL `https://dx.doi.org/10.1088/1748-0221/12/03/P03011`.

[8] B Abi et al. (DUNE Collaboration). Neutrino interaction classification with a convolutional neural network in the dune far detector. *Physical Review D*, 102(9):092003, 2020. URL `https://doi.org/10.1103/PhysRevD.102.092003`.

[9] Kiara Carloni, Nicholas W Kamp, Austin Schneider, and Janet M Conrad. Convolutional neural networks for shower energy prediction in liquid argon time projection chambers. *Journal of Instrumentation*, 17(02):P02022, 2022. URL `https://dx.doi.org/10.1088/1748-0221/17/02/P02022`.

[10] Pierre Baldi, Jianming Bian, Lars Hertel, and Lingge Li. Improved energy reconstruction in nova with regression convolutional neural networks. *Physical Review D*, 99(1):012011, 2019. URL `https://doi.org/10.1103/PhysRevD.99.012011`.

[11] François Drielsma, Kazuhiro Terao, Laura Dominé, and Dae Heun Koh. Scalable, end-to-end, deep-learning-based data reconstruction chain for particle imaging detectors. *arXiv:2102.01033*, 2021. URL `https://doi.org/10.48550/arXiv.2102.01033`.

[12] C Adams et al. (MicroBooNE Collaboration). Deep neural network for pixel-level electromagnetic particle identification in the microboone liquid argon time projection chamber. *Physical Review D*, 99(9):092001, 2019. URL `https://doi.org/10.1103/PhysRevD.99.092001`.

[13] P Abratenko et al. (MicroBooNE Collaboration). Semantic segmentation with a sparse convolutional neural network for event reconstruction in microboone. *Physical Review D*, 103(5): 052012, 2021. URL `https://doi.org/10.1103/PhysRevD.103.052012`.

[14] P Abratenko et al. (MicroBooNE Collaboration). Search for an anomalous excess of charged-current quasielastic ν_e interactions with the microboone experiment using deep-learning-based reconstruction. *Physical Review D*, 105(11):112003, 2022. URL `https://doi.org/10.1103/PhysRevD.105.112003`.

[15] P Abratenko et al. (MicroBooNE Collaboration). Search for an excess of electron neutrino interactions in microboone using multiple final-state topologies. *Physical review letters*, 128 (24):241801, 2022. URL `https://doi.org/10.1103/PhysRevLett.128.241801`.

[16] A A Aguilar-Arevalo et al. (MiniBooNE Collaboration). Updated miniboone neutrino oscillation results with increased data and new background studies. *Physical Review D*, 103(5): 052002, 2021. URL `https://doi.org/10.1103/PhysRevD.103.052002`.

[17] P Abratenko et al. (MicroBooNE Collaboration). Search for an anomalous excess of inclusive charged-current ν_e interactions in the microboone experiment using wire-cell reconstruction. *Phys. Rev. D*, 105:112005, Jun 2022. URL `https://doi.org/10.1103/PhysRevD.105.112005`.

[18] P Abratenko et al. (MicroBooNE Collaboration). Wire-cell 3d pattern recognition techniques for neutrino event reconstruction in large lartpcs: algorithm description and quantitative evaluation with microboone simulation. *Journal of instrumentation*, 17(01):P01037, 2022. URL `https://dx.doi.org/10.1088/1748-0221/17/01/P01037`.

[19] P Abratenko et al. (MicroBooNE Collaboration). Cosmic ray background rejection with wire-

cell lartpc event reconstruction in the microboone detector. *Physical Review Applied*, 15: 064071, 2021. URL https://doi.org/10.1103/PhysRevApplied.15.064071.

[20] R Acciarri et al. (MicroBooNE Collaboration). Noise characterization and filtering in the microboone liquid argon tpc. *Journal of Instrumentation*, 12(08):P08003, 2017. URL https://dx.doi.org/10.1088/1748-0221/12/08/P08003.

[21] C Adams et al. (MicroBooNE Collaboration). Ionization electron signal processing in single phase lartpcs. part i. algorithm description and quantitative evaluation with microboone simulation. *Journal of Instrumentation*, 13(07):P07006, 2018. URL https://dx.doi.org/10.1088/1748-0221/13/07/P07006.

[22] C Adams et al. (MicroBooNE Collaboration). Ionization electron signal processing in single phase lartpcs. part ii. data/simulation comparison and performance in microboone. *Journal of Instrumentation*, 13(07):P07007, 2018. URL https://dx.doi.org/10.1088/1748-0221/13/07/P07007.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. URL https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

[24] K He, X Zhang, S Ren, and J Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015. URL https://doi.org/10.48550/arXiv.1512.03385.

[25] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016. URL https://doi.org/10.48550/arXiv.1607.08022.

[26] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv:1706.01307*, 2017. URL https://doi.org/10.48550/arXiv.1706.01307.

[27] JunYoung Gwak, Christopher B Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. *arXiv:2006.12356*, 2020. URL https://doi.org/10.48550/arXiv.2006.12356.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. URL https://doi.ieeecomputersociety.org/10.1109/ICCV.

2017.324.

[29] P Abratenko et al. (MicroBooNE Collaboration). Vertex-finding and reconstruction of contained two-track neutrino events in the microboone detector. *Journal of instrumentation*, 16 (02):P02017, 2021. URL https://dx.doi.org/10.1088/1748-0221/16/02/P02017.

[30] R L Workman et al. Review of Particle Physics. *PTEP*, 2022:083C01, 2022. doi: 10.1093/ptep/ptac097.

[31] P Abratenko et al. (MicroBooNE Collaboration). Electromagnetic shower reconstruction and energy validation with michel electrons and π^0 samples for the deep-learning-based analyses in microboone. *Journal of Instrumentation*, 16(12):T12017, 2021. URL https://dx.doi.org/10.1088/1748-0221/16/12/T12017.

[32] F Psihas, E Niner, M Groh, R Murphy, A Aurisano, A Himmel, K Lang, M D Messier, A Radovic, and A Sousa. Context-enriched identification of particles with a convolutional network for neutrino events. *Physical Review D*, 100:073005, 2019. URL https://doi.org/10.1103/PhysRevD.100.073005.

[33] A Kendall, Y Gal, and R Cipolla. Multitask learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv:1705.07115*, 2018. URL https://doi.org/10.48550/arXiv.1705.07115.

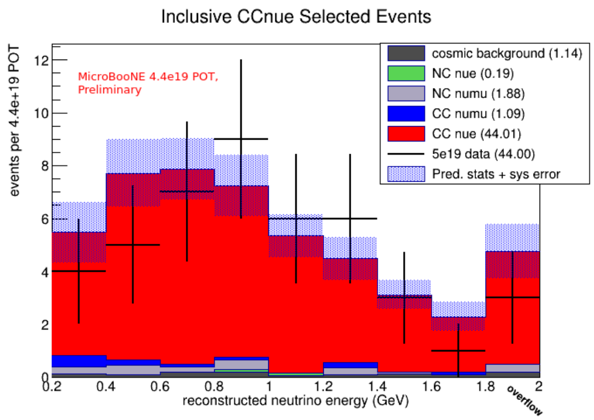
[34] I Loshchilov and F Hutter. Decoupled weight decay regularization. *arXiv:1711.05101*, 2019. URL https://doi.org/10.48550/arXiv.1711.05101.

[35] L N Smith and N Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv:1708.07120*, 2018. URL https://doi.org/10.48550/arXiv.1708.07120.

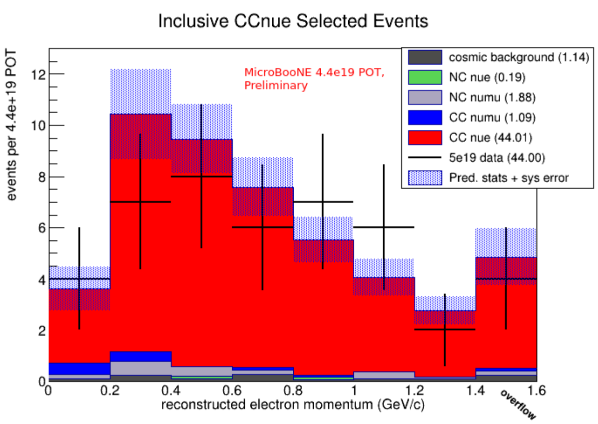
[36] Xiangpan Ji, Wenqiang Gu, Xin Qian, Hanyu Wei, and Chao Zhang. Combined neyman–pearson chi-square: An improved approximation to the poisson-likelihood chi-square. *Nuclear Instruments and Methods in Physics Research Section A*, 961:163677, 2020. URL https://doi.org/10.1016/j.nima.2020.163677.

Appendix A: Additional distributions for data vs expectation comparisons

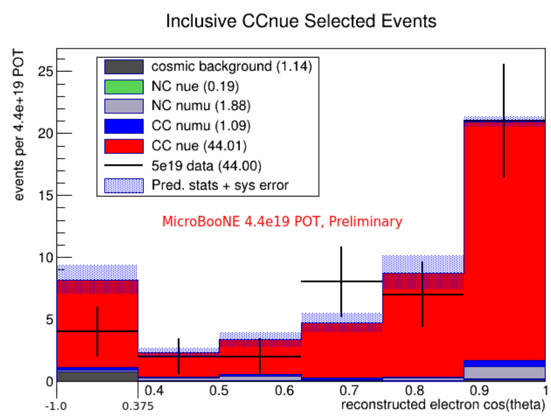
In this appendix, we show the re-binned kinematic variable distributions for the CC ν_e selection (figure 39) and the full covariance matrices (figure 40) used for the χ^2 goodness of fit tests discussed in section III D.



(a)



(b)



(c)

Figure 39: Predicted and MicroBooNE open data distributions, with the binning used in the χ^2 goodness of fit tests of section III D, of events passing the CC ν_e selection binned in (a) reconstructed neutrino energy, (b) reconstructed electron momentum, and (c) reconstructed $\cos(\theta_e)$, where θ_e is the angle between the reconstructed electron shower and the beam.

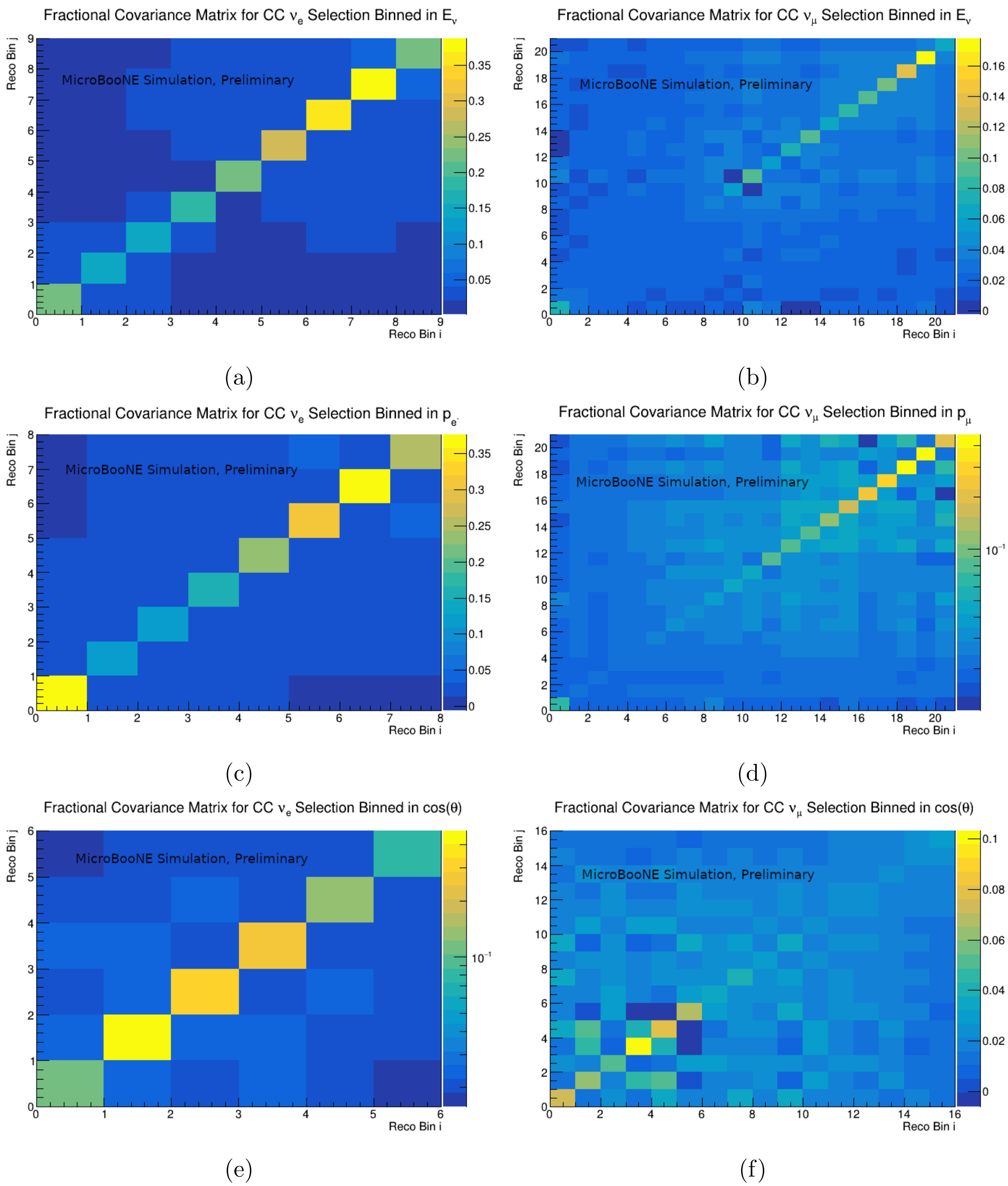


Figure 40: The full fractional covariance matrices used in the χ^2 goodness of fit tests from section III D.