μ/π separation using Convolutional Neural Networks for the MicroBooNE Charged Current Inclusive Cross Section Measurement

Jessica Nicole Esquivel

Bachelor of Science in Electrical Engineering and Applied Physics
St. Mary's University
San Antonio, TX, USA 2011

DISSERTATION

Submitted in partial fulfillment
of the requirements for the degree
Doctor of Philosophy in Physics

- * - DRAFT January 30, 2018 - * -

December, 2017
Syracuse University

Syracuse, New York

Abstract

The purpose of this thesis was to use Convolutional Neural Networks (CNN) to separate $\mu' s$ and $\pi' s$ for use in increasing the acceptance rate of $\mu' s$ below the implemented 75cm track length cut in the Charged Current Inclusive (CC-Inclusive) event selection for the CC-Inclusive Cross-Section Measurement. In doing this, we increase acceptance rate for CC-Inclusive events below a specific momentum range.

Dedication

I dedicate this dissertation to the two important women in my life; My wife and my mom. Both have been there cheering me on giving me strength and love as I worked towards the hardest accomplishment I've ever done.

Jessica Nicole Esquivel

Acknowledgements

Of the many people who deserve thanks, some are particularly prominent, such as my supervisor…

Contents

List of figures

List of tables

407

"If they don't give you a seat at the table,
bring a folding chair."
— Shirley Chisholm

Chapter 1

Introduction

This thesis will be a description of work done to further increase effiiciency and purity of the charged current inclusive cross section measurement using the MicroBooNE detector. It will also describe the MicroBooNE detector, what neutrinos are, the charged current inclusive cross section measuremnt and it's importance as well as convolutional neural networks and how they can be used in μ/π separation. Chapter 2 will talk about the background of neutrinos and the people and detectors that discovered neutrinos as well as an in depth history of neutrino oscillation and the discovery that neutrinos have mass.

Chapter 3 will discuss the MicroBooNE experiement, specifically, how Liquid Argon Time Projection Chambers work, the Light Collection System and the Electronic and Readout Trigger systems. This chapeter will also describe the Booster Neutrino Beam sationed at Fermilab.

Chapter 4 will discuss the work that was done to detect the first neutrinos seen in the MicroBooNE detector and the software reconstruction efforts required to create an automated neutrino ID filter that was used to find the first neutrinos and then was later expanded on to create the charged current inclusive filter that will be discussed in chapter 5

Chapter 6 will give a brief description of what Convolutional Neural Networks are and how it will be used for μ/π separation in this selection. Chapter 7 will discuss the hardware frameworks and training methods used to train multiple Convolutional Neural Networks for use in the charged current inclusive cross section measurement. Chapters 8 and **??** will discuss the results of using Convolutional Neural Networks on monte-carlo and data to sift out charged current inclusive neutrino events.

Chapter 2

Neutrinos

2.1 What are Neutrinos

Neutrinos are fundamental particles which help make up the universe. They are also one of the least understood. Neutrinos are not affected by the electromagnetic force because they do not have electric charge. Neutrinos are affected by a weak sub-atomic force of much shorter range than electromagnetism, and are therefore able to pass through great distances in matter without much possibility of being affected by it. Until the late 1990's, neutrinos were thought to have no mass. Neutrinos are created by radioactive decay such as the ones that happen in the sun, in nuclear reactors or when cosmic rays hit atoms. There are three types of neutrinos, ν_e, ν_μ and ν_τ which correspond to their charged lepton pairs.

As previously stated, neutrinos are very weakly interacting; in fact, neutrinos can pass unscathed through a wall of lead several hundred light-years thick. Because neutrinos interact so rarely, studying neutrinos requires a massive detector and a powerful neutrino source. With that being said, we can only "see" a neutrino when they interact in a detector. In a collision, distinct charged particles are produced with each type of neutrino because of the weak force. An electron neutrino will create an electron, a muon neutrino will create a muon, and a tau neutrino will create a tau. The charged lepton track the particle leaves in the detector is how one figures out what type of neutrino interaction was "seen". Liquid Argon Time Projection Chambers are being used to study neutrinos due to their excellent imaging and particle identification capabilities.

2.2 History of Neutrinos

The neutrino was first postulated by Wolfgang Pauli in 1930 to explain how to resolve the conservation of energy, momentum and angular momentum problem in beta decay. Pauli suggested that this missing energy might be carried off, unseen, by a neutral particle (he called neutron) which was escaping detection. James Chadwick discovered a much heavier nuclear particle in 1932 that he also named neutron, leaving two particles with the same name. Enrico Fermi was the first person to coin the term neutrino (which means little neutral one in italian) in 1933 to fix this confusion.

Fermi's paper, which was published in 1934, unified Pauli's neutrino with Paul Dirac's positron and Werner Heisenberg's neutron-proton model and his theory accurately explained many experimentally observed results. Wang Ganchang first proposed the use of beta capture to experimentally detect neutrinos and in 1956 Clyde Cowan and Frederick Reines published their work stating that they had detected the neutrino. The experiment called for antineutrinos created in a nuclear reactor by beta decay that reacted with protons producing neutrons and positrons: $\nu_e + p^+ \rightarrow n^0 + e^+$. Once this happens, the positron finds an electron and they annihilate each other and the resulting gamma rays are detectable. The neutron is detected by neutron capture and the releasing of another gamma ray.

In 1962 Leon M. Lenderman, Melvin Schwartz and Jack Steinberger were the first to detect interactions of the muon neutrino [**?**]. The trio received the 1988 Nobel Prize in Physics for their discovery of the muon neutrino. The experiment used a beam of energetic protons from Brookhaven's Alternating Gradient Synchrotron (AGS) to produce a shower of pions. These pions would then travel 70 ft. toward a 5,000 ton steel wall. The pions then decayed into muons and neutrinos, the neutrinos being the only particle making it through. These neutrinos would enter a neon-filled detector producing muon spark trails that were detected and photographed, proving the existence of muon neutrinos. The experiment's use of the first ever neutrino beam was pioneering work that scientists around the world still use today.

The first detection of the tau neutrino was announced in the summer of 2000 by the DONUT collaboration at Fermilab. The scientists used the Tevatron to produce an intense neutrino beam. The neutrinos then passed through layers of nuclear emulsion. When a neutrino interaction would occur, charged particles would leave visible tracks in the emulsion.

2.2.1 Solar Oscillations and the Solar Neutrino Problem

In the late 1960s, it was found that the number of electron neutrinos arriving from the sun was around 1/3 to 1/2 the number predicted by the Standard Solar Model. This became known as the solar neutrino problem and remained unresolved for around thirty years. This problem was resolved by the discovery of neutrino oscillation and mass. [1]

The standard solar model (SSM) is a mathematical model developed by John Bahcall of the sun as a spherical ball of gas with varying states of ionization.The solar neutrino flux derived from the SSM is shown in figure 2.1 [?]. Nuclear fusion and decay processes produce an abundant amount of neutrinos. The standard solar model predicts that these reactions produce several groups of neutrinos, each with differing fluxes and energy spectra. The figure also shows the ranges of detection of existing solar neutrino experiments in different shades of blue to illustrate that they sample different portions of the solar neutrino energy spectrum. Three of these experiments are discussed below.

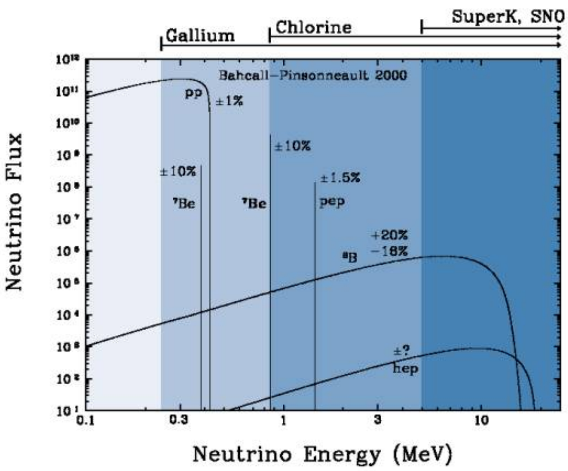


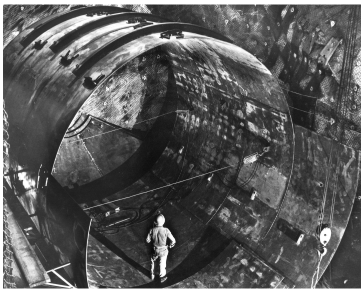
Figure 2.1: The Standard Solar Model [?]

The first experiment to detect the effects of neutrino oscillation was the Ray Davis's Homestake Experiment. The detector was stationed in the Homestake Gold Mine in Lead, South Dakota. It was 1,478 meters underground and was 380 m^3. The detector was filled with perchloroethylene. Perchloroethylene was chosen because of its high concentrations of chlorine. When an ν_e interacted with a chlorine-37 atom, the atom would transform to argon-37 which was then extracted and counted. The neutrino capture reaction is shown in equation 2.1. Davis observed a deficit of about 1/3 the flux of solar neutrinos that was predicted by Bahcall's Standard Solar Model. The unexplained difference between the measured solar neutrino flux and model predictions led to the Solar Neutrino Problem. [4]

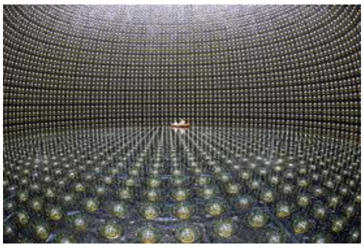
$$\nu_e + {}^{37}Cl \rightarrow {}^{37}Ar + e^- \tag{2.1}$$

While it is now known that the Homestake Experiment results were indicating neutrino flavor oscillation, some physicists were weary of the results. Conclusive evidence of the Solar Neutrino Problem was provided by the Kamiokande-II experiment, a water cherenkov detector with a low enough energy threshold to detect neutrinos through neutrino-electron elastic scattering. In the elastic scattering interaction the electrons coming out of the point of reaction strongly point in the direction that the neutrino was traveling, away from the sun. While the neutrinos observed in Kamiokande-II were clearly from the sun, there was still a discrepancy between Kamiokande-II and Homestake; The Kamiokande-II experiment measured about 1/2 the predicted flux, rather than the 1/3 that the Homestake Experiment saw.

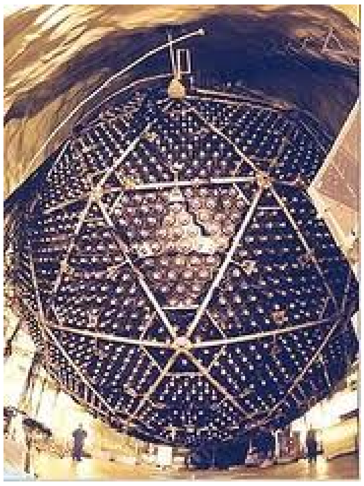
The solution to the solar neutrino problem was finally experimentally determined by the Sudbury Neutrino Observatory(SNO). The Ray Davis's Homestake Experiment was only sensitive to electron neutrinos, and the Kamiokande-II Experiment was dominated by the electron neutrino signal. The SNO experiment had the capability to see all three neutrino flavors. Because of this, it was possible to measure the electron neutrinos and total neutrino flux. The experiment demonstrated that the deficit was due to the MSW effect [5], the conversion of electron neutrinos from their pure flavor state into the second neutrino mass eigenstate as they passed through a resonance due to the changing density of the sun. The resonance is energy dependent, and is visible near 2 MeV. The water cherenkov detectors only detect neutrinos above about 5 MeV, while the radiochemical experiments were sensitive to lower energy (0.8 MeV for chlorine, 0.2 MeV for gallium), and this turned out to be the source of the difference in the observed neutrino rates at the two types of experiments. Figure 2.2 shows Homestake, Kamiokande-II and SNO experiments.



(a) Ray Davis's Homestake Experiment [?]



(b) Kamiokande-II Experiment [?]



(c) SNO Experiment [?]

Figure 2.2: Solar Neutrino Experiments

2.2.2 Atmospheric Oscillations and the Atmospheric Neutrino Anomaly

Atmospheric neutrinos are neutrinos that stem from the decay hadrons coming from primary cosmic rays. The dominant part of the decay chain is shown in equations 2.2 and 2.3

$$\pi^+ \rightarrow \mu^+ \nu_\mu \mu^+ \rightarrow e^+ \nu_e \overline{\nu_\mu} \tag{2.2}$$

$$\pi^- \rightarrow \mu^- \overline{\nu_\mu} \mu^- \rightarrow e^- \overline{\nu_e} \nu_\mu \tag{2.3}$$

In general, these neutrinos have energies from 1 GeV to 100s of GeV and the ratio of ν_μ's to ν_e's equals to 2 (see equation 2.4)

$$R = \frac{(\nu_\mu + \overline{\nu_\mu})}{(\nu_e + \overline{\nu_e})} \tag{2.4}$$

There have been two types of detectors used to study atmospheric neutrinos: Water Cherenkov detectors and tracking calorimeters. Super-Kamiokande is the detector we will focus on. These atmospheric detector experiments measure the ratio of ν_μ to ν_e.

They also measure the zenith angle distribution of the neutrinos. These experiments report a double ratio (shown in equation 2.5). This double ratio is the ratio measured in the detector to the ratio thats expected which is 2. If the double ratio equals to 1, the data agrees with the prediction. Various measurements from multiple experiments are shown in figure 2.1. Except for Frejus, all R measurements are less than 1. This discrepancy between the predicted R and the measured R became known as the Atmospheric Neutrino Anomaly.

$$R = \frac{(N_\mu/N_e)_{DATA}}{(N_\mu/N_e)_{SIM}} \tag{2.5}$$

Experiment	Type of experiment	R
Super-Kamiokande	Water Cherenkov	0.675 $\pm$ 0.085
Soudan2	Iron Tracking Calorimeter	0.69 $\pm$ 0.13
IMB	Water Cherenkiv	0.54 $\pm$ 0.12
Kamiokande	Water Cherenkov	0.60 $\pm$ 0.07
Frejus	Iron Tracking Calorimeter	1.0 $\pm$ 0.15

Table 2.1: Measurments of the double ratio for various atmospheric neutrino experiments

Kamiokande-II has the capability of measuring the direction of the incoming neutrinos. The expectation of atmospheric neutrino detection is that the flux will be isotropic due to the fact that atmospheric neutrinos can reach the detector from all directions. Kamiokande-II noticed that muon-like data did not agree well with this expectation. At low energies approximately half of the ν_μ are missing over the full range of zenith angles. At high energies the number of ν_μ coming down from above the detector seems to agree with expectation, but half of the same ν_μ coming up from below the detector are missing. This anomaly can be easily explained by neutrino flavor oscillations. Due to the fact that the neutrino travels less distance coming straight down into the detector (about 15 km) than coming up from the bottom of the detector(13000 km) changes the probability of oscillation. The probability of oscillation for the muon neutrinos coming down into the detector is roughly zero, whereas for neutrinos coming up, the oscillation probability is $\sin^2(2\theta)$. Both the solar and atmospheric neutrino problems can be explained by neutrino oscillation so it's fitting to derive this phenomenon mathematically. In the next two sections, two flavor and three flavor neutrino oscillation derivations will be explained.

2.3 Neutrino Oscillations

Neutrino oscillation was first predicted by Bruno Pontecorvo. It describes the phenomenon of a neutrino created with a specific lepton flavor (electron, muon or tau) that is later measured to have a different flavor. Neutrino oscillation is important theoretically and experimentally due to the fact that this observation implies that the neutrino has a non-zero mass, which is not part of the original Standard Model of particle physics. [2]

2.3.1 Two Flavor Neutrino Oscillation Formulation

The flavor eigenstates can oscillate between each other because they are composed of an add mixture of mass eigenstates(ν_1,ν_2). Figure 2.3 shows the mass and flavor eigenstates rotated by an angle θ which is the mixing angle.

In matrix form the wavefunctions are:

$$\begin{pmatrix} \nu_\mu \\ \nu_e \end{pmatrix} = \begin{pmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{pmatrix} * \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} \tag{2.6}$$

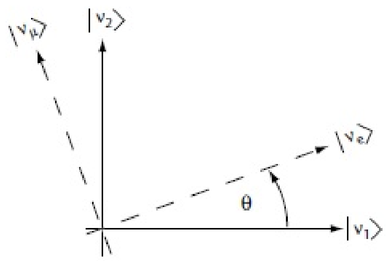


Figure 2.3: The flavor eigenstates are rotated by an angle θ with respect to the mass eigenstates

Applying the time evolution operator to ν_μ:

$$|\nu_\mu(t)>= -sin\theta|\nu_1> e^{-i\frac{E_1 t}{\hbar}} + cos\theta|\nu_2> e^{-i\frac{E_2 t}{\hbar}} \tag{2.7}$$

where $E_1 = \sqrt{p^2c^2 + m_1^2c^4}$ and $E_2 = \sqrt{p^2c^2 + m_2^2c^4}$ and $p_1 = p_2$. For the time being, let us assume $\hbar = c = 1$. With this assumption: $E_1 = \sqrt{p^2 + m_1^2}$ and $E_2 = \sqrt{p^2 + m_2^2}$. The next modifications is to assume neutrinos are relativistic:

$$\gamma = \frac{E}{m_o^2c^2} = \frac{\sqrt{p^2c^2 + m_o^2c^4}}{m_oc^2} \gg 1 \tag{2.8}$$

because of this,

$$p \gg m_o \tag{2.9}$$

$$E = \sqrt{p^2 + m_o^2} = p\sqrt{1 + m_o^2/p^2} \simeq p + \frac{1}{2}\frac{m_o^2}{p} \tag{2.10}$$

where the binomial expansion is used. Now E_1 and E_2 can be written as:

$$E_1 \simeq p + \frac{1}{2}\frac{m_1^2}{p} \text{ and } E_2 \simeq p + \frac{1}{2}\frac{m_2^2}{p} \tag{2.11}$$

Now applying all these assumptions back into equation 2.7 gives us:

$$|\nu_\mu(t)>= -sin\theta|\nu_1> e^{-i\left(p+\frac{1}{2}\frac{m_1^2}{p}\right)t} + cos\theta|\nu_2> e^{-i\left(p+\frac{1}{2}\frac{m_2^2}{p}\right)t} \tag{2.12}$$

$$|\nu_\mu(t)>= e^{-i\left(p+\frac{1}{2}\frac{m_1^2-m_2^2}{p}\right)t}(-sin\theta|\nu_1> +cos\theta|\nu_2>) \tag{2.13}$$

Substituting $\Delta m^2 = m_1^2 - m_2^2$ and $t = \frac{x}{c} = x$ and $e^{-iz} = e^{-i\left(p+\frac{1}{2}\frac{m_1^2}{p}\right)t}$ gives us:

$$|\nu_\mu(t)>= e^{-iz}\left(-sin\theta|\nu_1> +cos\theta|\nu_2> e^{+ix\left(\frac{1}{2}\frac{\Delta m^2}{p}\right)}\right) \tag{2.14}$$

Finding the Probability for a $\nu_\mu \rightarrow \nu_e$:

$$P(\nu_\mu \rightarrow \nu_e) = |<\nu_e|\nu_\mu(t)>|^2 \tag{2.15}$$

Remembering that $< \nu_i|\nu_j >= \delta_{ij}$

$$< \nu_e|\nu_\mu(t) >= e^{-iz}\left(-sin\theta cos\theta + sin\theta cos\theta e^{\frac{i\Delta m^2 x}{p}}\right) \tag{2.16}$$

Taking the absolute value squared gives us:

$$P(\nu_\mu \to \nu_e) = | < \nu_e|\nu_\mu(t) > |^2 = e^{+iz}e^{-iz}sin^2\theta cos^2\theta\left(-1 + e^{\frac{i\Delta m^2 x}{p}}\right)\left(-1 + e^{\frac{i\Delta m^2 x}{p}}\right) \tag{2.17}$$

Since the neutrino is relativistic we can set $p = E_\nu$ and change $x = L$. Also recognizing the trigonometric relation $(1 - cos2\theta)/2 = sin^2\theta$ the above equation becomes:

$$P(\nu_\mu \to \nu_e) = sin^2 2\theta sin^2\left(\frac{\Delta m^2 L}{4E_\nu}\right) \tag{2.18}$$

All that's left to do now is re-introduce $\hbar$ and c doing this we get:

$$P_{\nu_\mu \to \nu_e}(L, E) = sin^2 2\theta sin^2\left(1.27\Delta m^2\frac{L}{E_\nu}\right) \tag{2.19}$$

This equations has three important variables.

- The angle θ: This angle, as mentioned before, is called the mixing angle. It defines the difference between the flavor and the mass eigenstates. When $\theta = 0$ the mass and flavor eigenstates are identical and no oscillations occur.
- The mass squared difference, Δm^2: Again $\Delta m^2 = m_1^2 - m_2^2$. The reason this is an important variable is because it implies that for neutrinos to oscillate, neutrinos must have mass. Furthermore, the mass squared difference also tells us that the neutrino mass eigenstates must be different.
- L/E: This is the variable that is of most interest to experimental physicists due to the fact that it is the variable that we set. L is the distance between the source and detector and E is the energy of the neutrino. For a given Δm^2, the probability of oscillation changes with respect to L/E.

2.3.2 Three Flavor Neutrino Oscillation Formulation

Seeing the quantum mechanics involved in deriving the probability of a two flavor neutrino oscillation, it is now possible to formulate the three flavor neutrino oscillation. The three flavor neutrino oscillation formulation begins similarly to the two flavor, but there is the Pontecorvo-Maki-Nakagawa-Sakata matrix (PMNS) instead of the 2X2 matrix in the previous section. The PMNS matrix is show below:

$$\begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix} * \begin{pmatrix} e^{i\alpha_1/2} & 0 & 0 \\ 0 & e^{i\alpha_2/2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.20}$$

where $c_{ij} = cos\theta_{ij}$ and $s_{ij} = sin\theta_{ij}$

Following the same steps as before we get:

$$P_{\alpha\to\beta} = \delta_{\alpha\beta} - 4\Sigma Re(U^*_{\alpha i}U_{\beta i}U\alpha jU^*_{\beta j})sin^2\left(\frac{\Delta m^2_{ij}L}{4E}\right) 2\Sigma Im(U^*_{\alpha i}U_{\beta i}U\alpha jU^*_{\beta j})sin\left(\frac{\Delta m^2_{ij}L}{2E}\right) \tag{2.21}$$

The main things to notice here are δ_{ij} which is the CP violating term and has not been measured yet, and θ_{13} which has just been measured. CP violation is a violation of the postulated CP-symmetry. CP-symmetry states that the laws of physics should be the same if a particle were to be exchanged with its antiparticle and then if the left hand side of a decay were switched with the right hand side. Table 2.2 shows the current world knowledge of the values of all the fundamental parameters for neutrino oscillations [**?**].

Parameter	Value
θ_{12}	$33.9 \pm 1.0^{\circ}$
θ_{23}	$39^{\circ} < \theta_{23} < 51^{\circ}$
θ_{13}	$9.1 \pm 0.6^{\circ}$
Δm^2_{21}	$(7.50 \pm 0.20) * 10^{-5}\text{eV}^2$
$\lvert\Delta m^2_{32}\rvert$	$(2.32^{+0.12}_{-0.08}) * 10^{-3}\text{eV}^2$
δ_{cp}	unknown

Table 2.2: Current world knowledge of neutrino oscillation parameters [**?**]

Chapter 3

The MicroBooNE Experiment

The purpose of this chapter is to discuss and understand the details of the MicroBooNE detector. A thorough understanding of MicroBooNE and the technology behind liquid argon time projection chambers is important for understanding results as well as understanding how images were made for use in deep learning efforts that will be outlined in later chapters.

3.1 Liquid argon time projection chambers

Liquid Argon Time Projection Chambers (LArTPCs) are an exciting detector technology that provide excellent imaging and particle identification, and are now being used to study neutrinos. The Time Projection Chamber (TPC) was first invented by Nygren in 1974 [**?**] and the proposal for a LArTPC for neutrino physics was made by Rubbia [**?**] in 1977 with the ICARUS collaboration implementing this concept [**?**]. A LArTPC is a three-dimensional imaging detector that uses planes of wires at the edge of an active volume to read out an interaction. When a neutrino interacts with an argon atom, the charged particles that are produced ionize the LAr as they travel away from the interaction. By placing a uniform electric field throughout the LAr volume, the ionization is made to drift towards a set of anode planes, which consist of wires spaced very closely together collecting the ionized charge, which is subsequently read out by electronics connected to the anode wires. The collected ionization creates a spatial image of what happened in the detector on each anode plane. The position resolution of the interaction along the beam direction (perpendicular to drift direction) relies on the wire pitch, while the resolution in drift direction is dependent on the timing resolution of the electronics used and the longitudinal diffusion in the volume. The drift time of the ionization relative to the time of the original signal allows the signal to be projected back along the drift coordinate, hence the name LArTPC. Having very small distances between each wire within an anode plane allows for very fine granularity and detail to be captured, and having multiple wire planes at different angles provides independent two-dimensional views that can be combined into a three dimensional picture of the interaction. Once the charge signal is created on the anode planes, software analysis packages identify particles in the detector by using deposited energy on the wires along their track length. The 30 year development of the ICARUS detector has led to LArTPCs being used as cosmic ray [?], solar neutrino [?] and accelerator neutrino [?] detectors. The ArgoNeuT experiment at Fermilab was the first United States based liquid argon neutrino program that has since produced short-baseline $\nu - Ar$ cross-section measurements in the NUMI beamline [?]. The MicroBooNE experiment is the second experiment in the US based LArTPC neutrino program and will be discussed thoroughly in the next sections. The next phases of the liquid argon neutrino program are under way and are the Fermilab Short Baseline Neutrino (SBN) program [?] and the Deep Underground Neutrino Experiment (DUNE) [6]. The SBN program will include three LArTPC detectors, including the MicroBooNE detector, on the Booster Neutrino Beam (BNB) to do multiple-baseline oscillation measurements. The detector closest to the beam will be the 40 ton Short Baseline Neutrino Detector (SBND) [?] at 150 m and the detector furthest is the 600 ton ICARUS T600 [?] detector positioned at 600 m. The DUNE collaboration will deliver a 30 GeV neutrino beam 1300 km from Fermilab to a 34 kiloton LArTPC detector at Homestake, SD. DUNE will study the leptonic CP phase, δ_{cp}, as well as measure neutrino and antineutrino oscillations.

3.2 The MicroBooNE Time Projection Chamber

MicroBooNE (Micro Booster Neutrino Experiment) is a 89 ton active volume (180 ton total mass) LArTPC which is then inserted into a cylindrical crysotat on axis of the Booster Neutrino Beam (BNB) stationed at Fermilab in Batavia, Illinois. Understanding LArTPC technology and detector physics is necessary to build a LArTPC the size of DUNE, and MicroBooNE has made many advances in developing this technology [7] [8].

MicroBooNE's Time Projection Chamber (TPC) is 10.3 m long (beamline direction), 2.3 m high and 2.5 m wide (which corresponds to the drift distance). The TPC is shown in figure **??**. MicroBooNE is the largest LArTPC currently running in the world [9]. This LArTPC has 3 wire planes: 1 plane that collects the ionization in the wires and is 0° to the virtical with 3456 wires spaced 3 mm apart, and 2 planes where the ionization drifts passed and induces a signal at $\pm 60^{\circ}$ to the vertical each with 2400 wires also spaced 3 mm apart. Each plane has a spacing also of 3 mm from eachother. The first two planes are the induction planes and the last is the collection. The 270 V/cm electric field of the TPC is created using 64 stainless steel tubes shaped into rectangules around the TPC and held in place by G10 to form a field cage. The cathode is charged at a high voltage of -70 kV and this voltage is stepped down across the field cage tubes using a voltage divider chain with an equivalent resitance of 240 $M\Omega$ between the tubes. The field cage tubes are separated by 4 cm from center to center. The electron drift distance is 2.5 m in the x direction with a drift time of 2.3 ms. Maintaining high charge yield is done by continuously recirculating and purifying the argon. The purity is monitored using MicroBooNE's light collection system. Another use of the light collection system is initial timing and drift coordinate of the interaction.

MicroBooNE's light collection system is a crucial part for 3D reconstruction of all particle interactions in the LArTPC. The initial interaction time, t_0, and initial drift coordinate, x_0, are not known from the TPC alone. For beam events, the accelerator clock is used to determine t_0 of the interaction and the x_0 can be inferred using drift time. Non-beam events, however, do not have this capability, which is why scintillation light from an interaction is used. The $\nu - Ar$ interaction produces scintillation light which is collected by photomultiplier tubes (PMTs) which allows the exact time, t_0 of the neutrino interaction to be determined. The scintillation light created propogates within nanoseconds to the light collection system compared to the milliseconds it takes the ionized electrons from the interaction to reach the anode wire planes. Therefore we can precisely know where along the drift direction the particle interaction first took place. The scintillation light is also localized, so combining the PMT information with the wire plane information allows for cosmic background rejection happening outside the beam timing window.

The light collection system is made up of 32 Hammamatsu R5912-02mod cryogenic PMTs with a diameter of 8-inches. The PMTs are located behind the 3 wire anode planes and provides 0.85% photocathode coverage. Each PMT has an acrylic plate mounted in front of it that is coated with a wave-length shifting material called TPB.

The acrylic plates take in the scintillation light, at 128 nm, and re-emits it visible wavelengths visible to the PMTs, with a peak at 425 nm.

Both the light collection system and the TPC create analog signal that is read out and digitized by the electronics system. The process requires amplification and shaping of the signal which then is goes to the data aquisition (DAQ) software for writing of the digitized data to disk. The anode plane wires are connected to detector specific circuit boards (ASICS) that are submerged and operate inside the liquid argon volume. These ASICS send amplified signal to 11 feed-throughs where further amplification of the signal happens outside the cryostat. The signal is received by custom LArTPC readout modules distributed over nine readout crates which do the digitization. The TPC wires are digitized at 16 MHz then downsampled to 2 MHz. The TPC system reads out 4 frames of wire signal data per event, 1 frame before a trigger and 2 frames after the triggered frame. The four frames allows for identification of a neutrino interaction as well as cosmic background rejection. The process of digitization is similar for the light collection system. Each PMT signal undergoes a shaping with a 60 ns peaking time for digitization of multiple samples. The digitization occurs at 64 MHz but are not read out continuously during the TPC readout time. Only shaped PMT signal samples above a small threshold are read out and saved. Both the TPC and PMT readouts are initiated via triggers on a separate trigger board located in a warm electronics crate. The timing trigger is created by a timing signal from the BNB accelerator which is shaped and sent to the trigger board. The PMT trigger is generated when the PMT signal multiplicity is greater than 1 and the summed PMT pulse-height is more than 2 photo-electrons summed up over all PMT channels. When the trigger board gets both a timing trigger and a PMT trigger in coincidence, at BNB trigger is then generated by the board. This signal is then passed to all readout crates intiating the readout of data. The data is then sent to the DAQ software which then saves the data to disk into one event memory.

3.3 MicroBooNE's Physics Goals

3.3.1 The low-energy excess

The primary goal of the MicroBooNE experiment is to study and investigate the low-energy excess seen in MiniBooNE 3.1. MicroBooNE has the capability of confirming or denying this excess as electrons or photons due to the detector being in the same beam, having a similar baseline, and lastly the detector being able to clearly distinguish between electrons and photons. LArTPCs use the topology of events as well as energy loss near the vertex to differentiate petween single e^- tracks and photon-induced induced pair production $\gamma \rightarrow e^+e^-$, which wasn't possible in MiniBooNE, a Cherenkiv detector. This technique has been shown in the ArogoNeuT detector [**?**] and a side by side comparison of both event types in a LArTPC can be seen in figure **??**. An excess in electrons would point towards new oscillation physics beyond the standard model, while photons would be within the standard model. MicroBooNE will ovserve a 4-5σ signal.

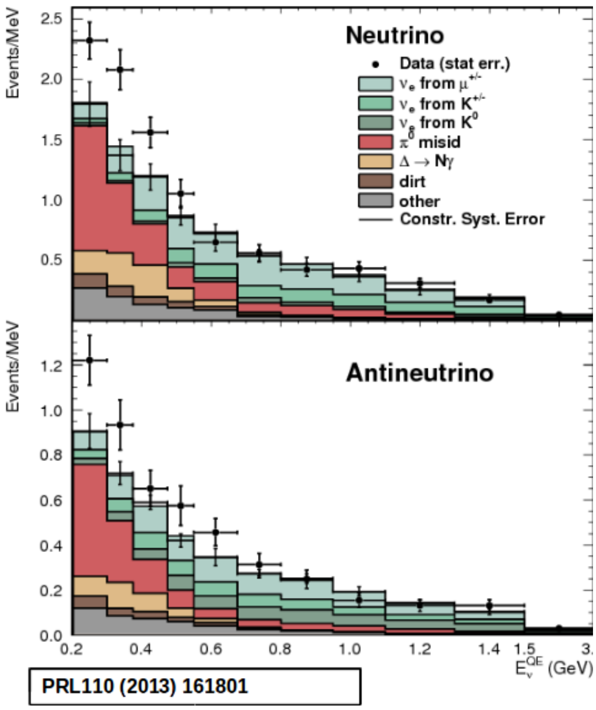


Figure 3.1: Low Energy excess seen in MiniBooNE

3.3.2 Cross sections

MicroBooNE's neutrino cross-section program will be the first $\nu - Ar$ cross-section in the 1 GeV energy range and one of only a few cross-section measurements of $\nu - Ar$ in the world. MicroBooNE is also the first liquid argon detector to collect the highest statistics sample of neutrino interactions. Investigating final-state-interactions in the 1GeV energy range provides information about short range nuclear correlations that affect the interpretations of neutrino oscillation experiment data.

One of the cross-section measurements MicroBooNE can make is an inclusive charged-current cross-section measurement (referred to as CC-inclusive). CC-inclusive events consist of a neutrino exchanging a $W^{\pm}$ boson with an argon atom, producing a charged lepton and any number of other final state particles. In MicroBooNE's case, a CC-inclusive event will mostly have a defining muon track coming out of the vertex due to our neutrinos being predominately ν_{μ}s. A cross-section measurement is the energy dependent probability of $\nu - Ar$ interaction in the detector. Cross-sections however are independent of the intensity or focus of the particle beam so they can be compared among different experiments. A background for a CC-inclusive cross-section measurement are the neutral-current events that contain a pion. It is possible to have a neutral current interaction with a $\pi + p$ event signature that looks like a charged current $\mu + p$ event. Reconstruction tools implemented to date don't efficiently separate muons from pions. A common way to separate these two particles species is to implement a track length cut. On average, muons tend to have longer track lengths in LArTPCs so by requiring that the hypothesized lepton be above a threshold track length, it is possible to increase signal to background.

3.3.3 Liquid argon detector development

The last physics goal for the MicroBooNE collaboration is to provide important information regarding LArTPC technology. Being the first in large scare LArTPCs in the US, MicroBooNE will be albe to provide improvements to High Voltage (HV) distribution, Noise Characterization [**?**], and Michel Electron Reconstruction [8].

3.4 The Booster Neutrino Beam

The MicroBooNE detector is stationed at Fermi National Accelerator Laboratory (FNAL) where it receives neutrinos from both the Booster Neutrino Beam (BNB) and Neutrinos from the Main Injector (NuMI) beams. MicroBooNE is on-axis for the BNB and off-axis by 135 mrad for NuMI. For the purpose of this analysis, only data from the BNB was used. This section will discuss how neutrinos are created using the BNB. How these neutrinos are produced as well as their flux through the MicroBooNE detector is necessary for any analysis because of the systematic uncertainties the beam introduces to a measurement. An arial view of fermilab as well as the BNB is shown in figure 3.2

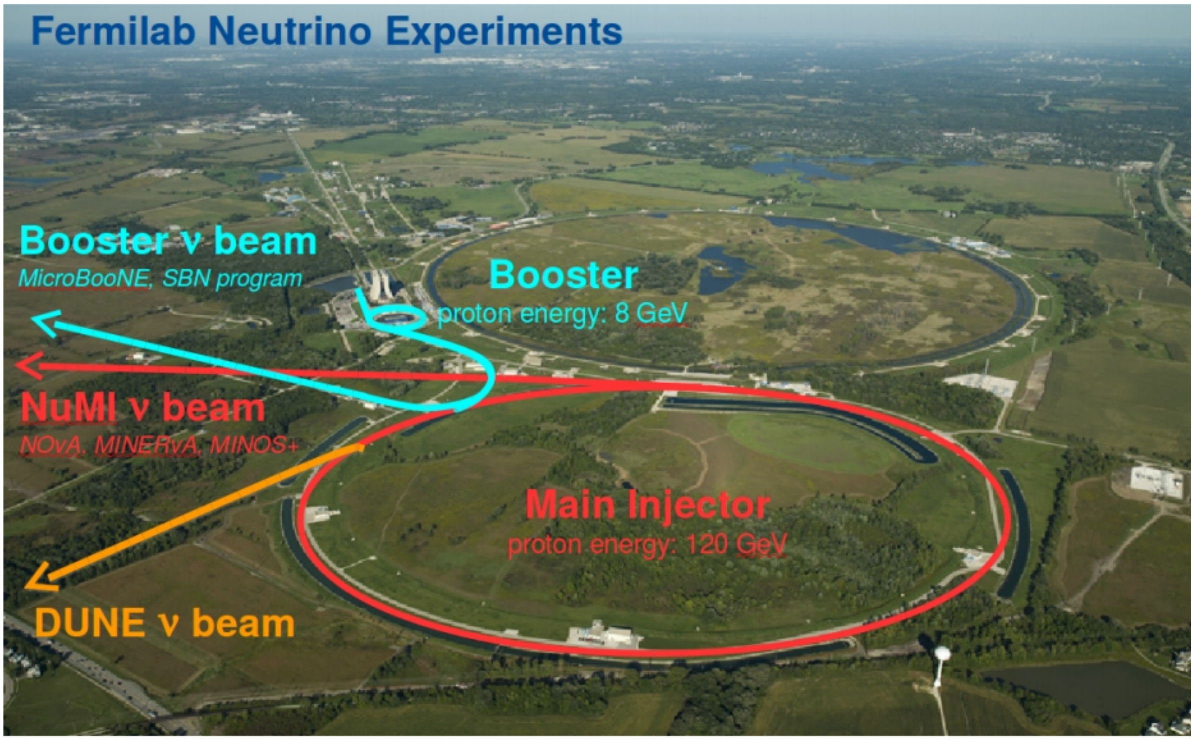


Figure 3.2: Arial view of the Main Injector and the Booster Neutrino Beam at Fermilab

3.4.1 Creating the Booster Neutrino Beam

The BNB is a very pure ν_μ beam, with only 0.6% contamination from $\nu_e s$. The energy also peaks around 700 MeV which is desired based on the probability of oscillation equation which depends on the the value of L/E, where L is the distance of the detector from the neutrino beam and E is the energy of the neutrino beam. L/E was chosen to increase the probability of seeing neutrino oscillations in the MiniBooNE Low Energy Excess (LEE) range based on the probability of oscillation equation, which is $P_{\nu_\mu \to \nu_e}(L, E) = \sin^2 2\theta \sin^2 \left(1.27\Delta m^2 \frac{L}{E_\nu}\right)$. The BNB collides 8.9 GeV/c momentum protons from the FNAL booster synchrotron into a beryllium target which produces a high flux of neutrinos. The protons originate from H^2 gas molecules that are turned into H^- ions by a Cockroft-Walton generator shown in figure **??**. The H^- initially are accelerated to 1MeV kinetic energy and are then passed to a linear accelerator using alternating electromagnetic fields to increase their energy to 400MeV. The ions are stripped of electrons by passing them through a carbon foil. The protons are bunched into beam spills which contain $4 * 10^{1}2$ protons in a 1.6 μs time window per spill. It's at this point that the protons are directed towards the beryllium target. The amount of protons directed towards the target (POT) is measured by two toroids upstream of the target with an error of 2%. Beam intensity, timing, width, position, and direction are monitored by beam position monitors, multi-wire chamber and resistive monitors. The beryllium target is 71.1 cm long, 1.7 proton interaction lengths, and is 0.51 cm in radius. The target is located inside a larger focusing electromagnet called the horn. THe horn is an aluminum alloy pulsed toroidal electromagnet. The pulsed current peaks at 170 kA with a time-width of 143 μs which coincides with the protons arriving on the target. The current flows from the inner conductor to the outer conductor with a maximum magnetic field of 1.5 Tesla. The magnetic field focuses the charged secondary particles produced by the p-Be interactions. The direction of current can be switched to changed to polarity of the secondary particles being focused creating a beam of either primarily neutrinos, with positively charged secondary particles, or antineutrinos.

Further down the beamline is a concrete collimator which absorbs particles not necessary to the neutrino flux. The collimator is 214 cm long and 30 cm in radius. After the collimator comes a 45 meter long, 1 meter raduis, air-filled cylindrical decay region which then ends in a beam-stop made of steam and concrete. The beam-stop contains an array of gas proportional counters to detect muons. The BNB is shown in figure 3.3.

3.5 Event Reconstruction

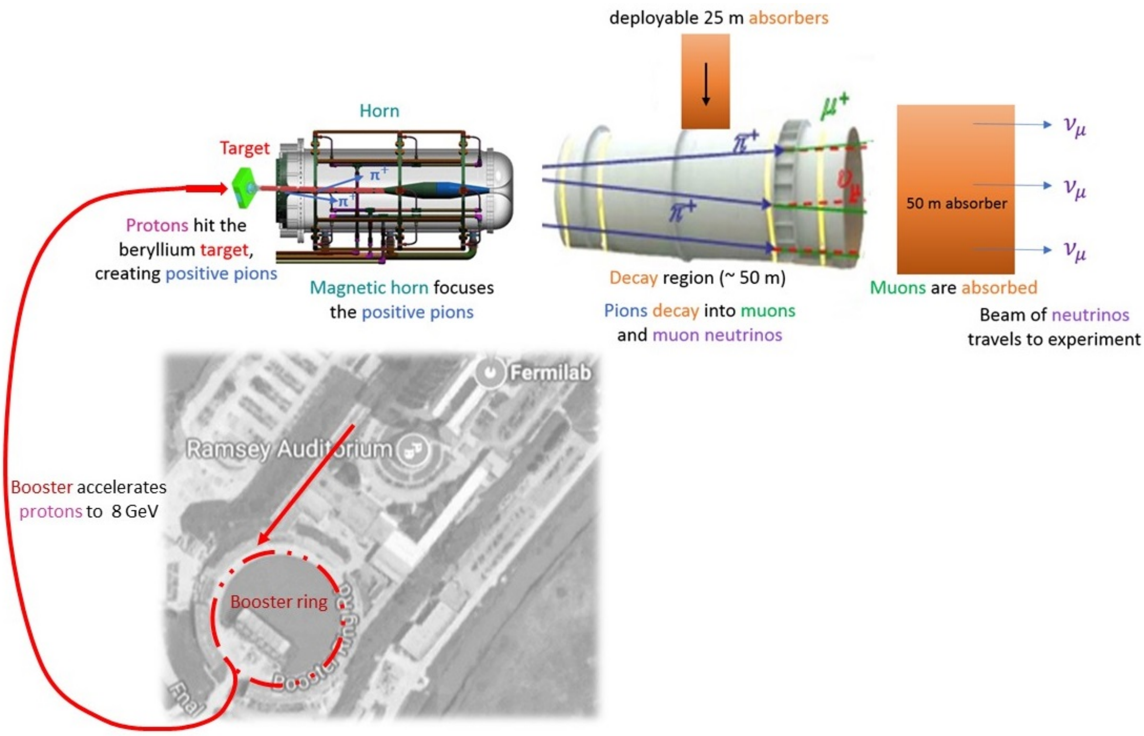


Figure 3.3: Arial view of the Main Injector and the Booster Neutrino Beam at Fermilab

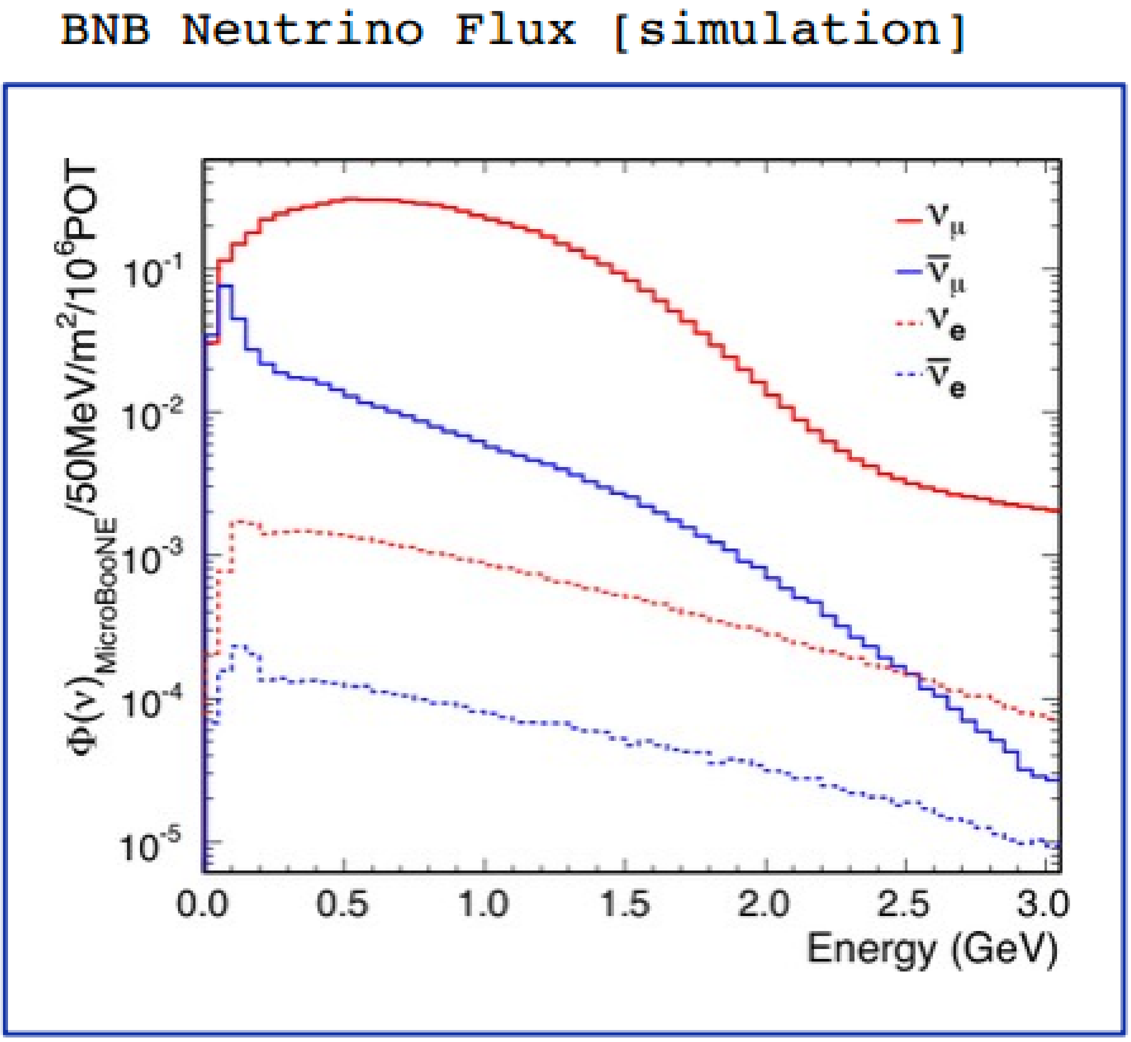


Figure 3.4: Energy spectrum of the Booster Neutrino Beam at Fermi National Laboratories

Chapter 4

Neutrino Identification: Finding MicroBooNE's first Neutrinos

The goal of the Neutrino Identification analysis was to positively identify BNB neutrino interactions in the MicroBooNE detector collected during the first days of running. Neutrino event candidates were identified in part by using a cut on detected flash of scintillation light during the 1.6 μs beam-spill length of the BNB as well as identifying reconstructed object from the TPC that are neutrino like. After this selection, 2D and 3D event displays were used for verification of the selection performance. This selection was targeted to reduce the ratio of neutrino events to cosmic-only events from the initial 1 neutrino to 675 cosmics to a ratio of 1 to 0.5 or better which is equivalent to a background reduction by a factor of 1000 or more. These selected events were used for MicroBooNE's public displays of neutrino interactions. A clearly visible neutrino interaction with an identifiable vertex and at least 2 tracks originating from the vertex was what the analysis focused on. This analysis wasn't optimized for high purity or efficiency, but rather for very distinguishable neutrino interactions that could be identified by the public.

4.1 Flash Finding

Flash finding is the first step used in finding neutrino interactions. This section will detail how optical information is reconstructed as well as analysis scripts and event filters were used.

4.1.1 Flash Reconstruction

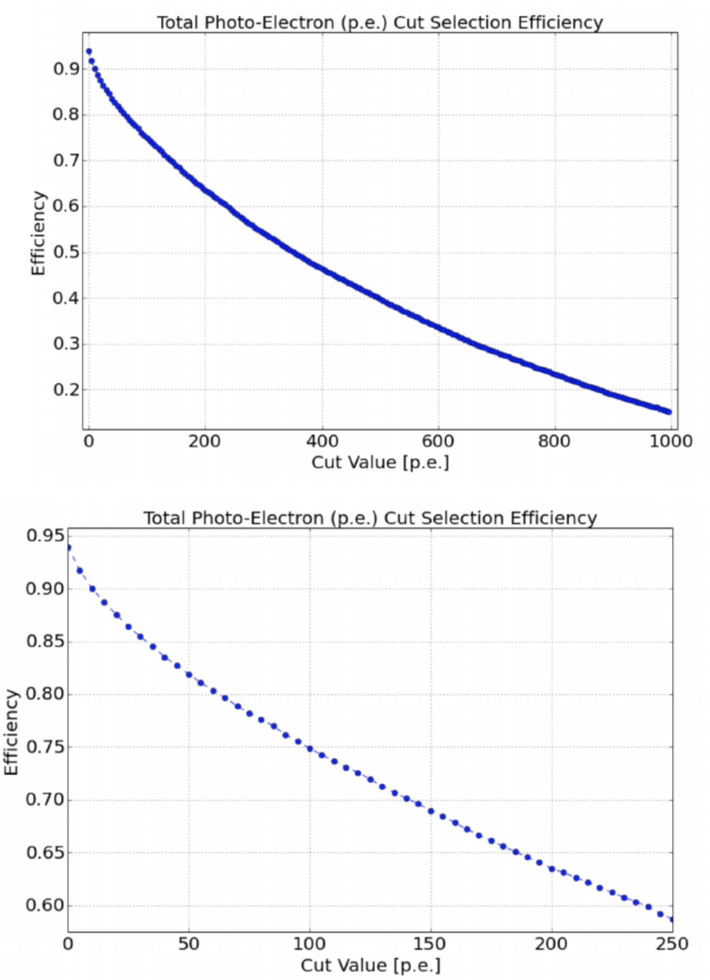
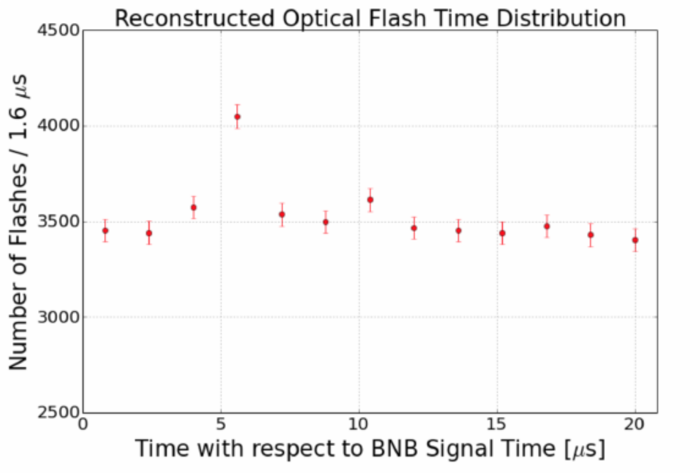


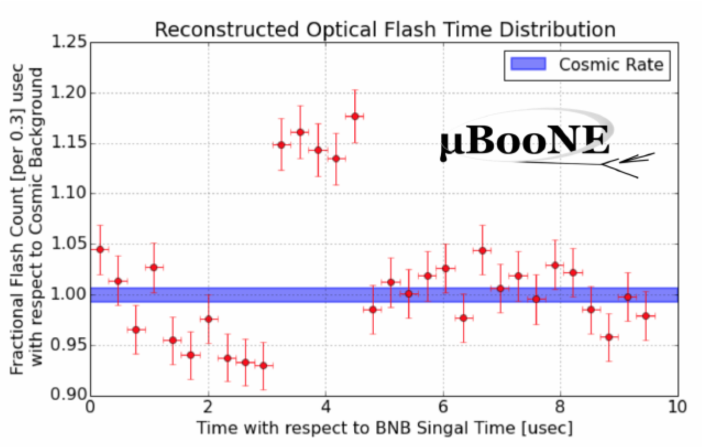
Figure 4.1: Efficiency for selecting beam events as a function of minimum total PE cut for all PE cuts as well as zoomed into interesting region.

A flash is described as a collection of light seen at the same time within the detector. They are then reconstructed by identifying signal from the PMTs above a specific photoelectron (PE) threshold. These signals are called optical hits. Optical hits from all the PMTs are then accumulated into 1 μs bins of time. If a specific bin is above a set PE threshold, then the optical hits that overlap in time are the labeled as the hits from the flash. All flash reconstructed properties like average time and x/y positions are then found via the flash labeled optical hits. The total size of the flash is found by summing up the total number of photoelectrons from all PMTs. Neutrino interactions and cosmic muons will have a larger flash size compared to noise and other low-energy backgrounds, therefore a total PE cut is used to reject these backgrounds. A total PE cut of 50 PE was deemed sufficient for this analysis. Figure 4.1 show the total PE versus the selection efficency of selecting neutrino beam events.

4.1.2 Beam Timing



(a) Predicted distribution of flash times with respect to trigger time for 1 day of data taking at nominal rate and intensity



(b) Measured distribution of flash times with a 50 PE threshold cut, with respect to trigger time. Shown as a ratio to the expected cosmic rate from off-beam data. A clear excess from neutrinos is visible between 3- 5 μs after the trigger time.

It is necessary to get the specific time from flashes if one uses flashes to filter out neutrino interactions coincident with the neutrino beam spill period and background. Before a filter can be applied, an understanding of the timing of the trigger and PMT readout with respect to the arrival of neutrinos from the BNB. To do this, a 1.6 μs window near the expected beamtime was created and verified by finding that the number of flashes was significantly above the cosmic-ray background flashes. Beam data during the first week of running, October 16th 2016 through October 22nd 2016 and were used for a timing measurement. The total POT uses corresponds to roughly 24 hours of data taking at nominal intensity ($4X10^{12}ppp$) and a 5 Hz repetition rate. Figure 4.2a shows size of the expected neutrino signal in time using Monte Carlo predictions and figure 4.2b shows the neutrino signal in data. The intensity in data is lower, however there can still be seen a significan excess above data.

4.1.3 Event Rates

Applying a 50 PE threshold cut inside a 1.6 μs window reduces the cosmic-ray passing rate to 0.8%. With a 5 Hz beam rate, this corresponds to 135 cosmics passing per hour. The neutrino passing rate for this filter is about 22 events per hour. To further increase the neutrino to cosmic ratio, TPC topology cuts were implemented and will be discussed in the following section.

4.2 TPC Topology Selection

In order to further reduce the background of cosmic events, two independent selection streams using TPC wire data reconstruction was implemented. The first using 2D reconstructed clusters, and the second using 3D resonctructed tracks. Both streams look for neutrino interactions in the active TPC volume which are identifiable by two or more tracks originating from the same vertex.

Both 2D and 3D channels were optimized using monte carlo simulation which used a 128 kV cathode voltage. Passing rates were calculated using a 0.008 efficiency factor for cosmic events passing to simulate the flash finding described in section 4.1. This efficiency factor was an overestimation and was just used to get a general feel of what signal and background rates we would actually see in data.

4.2.1 Cosmic Tagging

The first step in TPC selection was based on the geometry of cosmic tracks in an event. The cosmic ray muon geometry tagger runs on 3D tracks and assigns a score to each reconstructed track on the likeliness of the track originating from a cosmic. The cosmic scores are detailed below:

- 1: The track is tagged as entering or entering the TPC
- 0.95: The track is a delta ray associated with a tagged track
- 0.5: The track is either entering or exiting, but not both
- 0.4: The track is entering or exiting through the Z boundary
- 0: The track isn't tagged

Clusters are assigned either a 0 or 1, 1 being a cosmic. In simulation, 90% of cosmics are tagged as cosmics. These tracks are no longer considered when looking for a neutrino topology. Requiring that the tracks be contained in turn affects the neutrino efficiency by 20%. The algorithm checks that each track is contained within a boundary region of 10 cm from all sides of the TPC. This boundary region was optimized via handscanning of experimental data.

As can be expected, cosmic tagging is more efficient in the 3D channel (tracks) than the 2D channel (clusters) because the reconstructed tracks can use the full 3D position information of the entering and exiting points while the 2D channel mainly use the reconstructed x position of the cluster which is associated to timing.

Cosmic tagging uses timing information to reject tracks and clusters that are outside of drift window. The drift window for 128 kV is 1.6 μs while for 70 kv, the actual voltage MicroBooNE is running at, is 2.3 μs. Due to this variation between simulation and data, we expect to see $2.3/1.6 = 1.44$ times more cosmic induced tracks or clusters in the drift window.

4.2.2 2D Cluster Selection

This selection was spearheaded by myself and Katherine Woodruff. After looking at experimental cosmics data, 2D clustering performs well, while 3D track reconstruction is affected by more variations in simulation, for example noise filters. This was the motivation for having a selction only on 2D clusters in the collection (Y) plane. As stated previously, the goal of this analysis was to find identifiable neutrino interactions for use in public event displays, in future analyses, the 3D track reconstruction has been modified to further increase the tracking efficiency and has more information that just the clusters. For this analysis, however, 2D cluster information was sufficient enough for neutrino selection.

Primary Cuts

The first cuts were used to select which clusters to consider. First the clusters must have at least ten hits on the collection plane and have a cosmic tagging score < 0.4. Only events that have at least two clusters that satisfy these primary cuts continue on.

After the initial cosmic tagging is applied, the following cuts are used to furture separate identifiable neutrinos for background cosmics.

The next cut was to remove long, vertical clusters. This was applied after seeing that most cosmic induced clusters passing were long with high angles, while neutrino induced clusters were mainly forward going. We required a good cluster to either have a projected start angle less than 30 degrees from the z axis or be less than 200 wires long. The length cut was added to make sure we don't cut any short high angle clusters that can correspond with a proton, or other highly ionizing particle associated with a long muon cluster. The 200 wire cut roughly equates to 0.6 m in the z direction, with a 3 mm wire pitch. Also, the projected angle is defined by tan $\alpha = \Delta T/\Delta W$ where T is the time ticks and W is the wires.

The last cut requires the clusters to be either 30 time ticks or 30 wires. This cut was applied to reduce small delta rays associated with a cosmic without removing proton clusters associated with a long muon cluster, which saves ideal neutrino events that have both a long minimum ionizing muon like cluster and a short highly ionizing proton like cluster.

Secondary Cuts

The secondary cuts look to match long, low-angle clusters with short, high-charge clusters. Only clusters that have passed previous cuts are used. First clusters with length greater than 100 wires are chosen, which is approximately 0.3 m in the z

Cluster set	No Cuts	Primary Cuts	Secondary Cuts
Neutrinos only	570	303	32
Cosmics only (no flash)	308,016	291,879	602
Cosmics only (w/ flash)	2464	2335	5
Neutrinos/Cosmics	0.23	0.13	6.4

Table 4.1: Passing rates for 2D cluster cuts for neutrino on MC set and a cosmic only MC set. First column shows event rates with no cuts applied to both sets. Columns two and three show event rates after primary and secondary cuts are applied. Line three shows the seccond line scaled with the flash finding factor of 0.008. All events are normalized to per day assuming we are running at 5 Hz.

direction. Then we search for any cluster that is within approximately 3 cm (10 wires and 30 time ticks) away from the low-z end of the long cluster. This cluster must also be shorter than the first. In our reconstruction, the start and end point of a cluster can be swapped so both ends of the short cluster are compared to the long cluster.

Now that there is a vertex match, cuts based on charge and projected opening angle are implemented. We require the short cluster to have a higher start charge than the long cluster or the long cluster be longer than 500 wires. Start charge is defined as the charge on the first wire in ADC counts. The projected opening angle must also be between 11 and 90 degrees. This last cut is intended to remove clusters that are entirely overlapping or are part of the same long track. The resulting neutrino/cosmic event rate per day is shown in table 4.1. Figures 4.3 and 4.4 shows the percentages of clusters that pass each primary and secondary cuts.

4.2.3 3D Tracks and vertices Selection

The neutrino selection for the 3D channel was based on a reconstructed vertex and two tracks. All vertices and tracks were looped over that had a cosmic tag score < 0.4 and the distances below were calculated:

- d: distance between the start points of the two tracks.
- d_1: distance between vertex and start of track 1.
- d_2: distance between vertex and start of track 2.

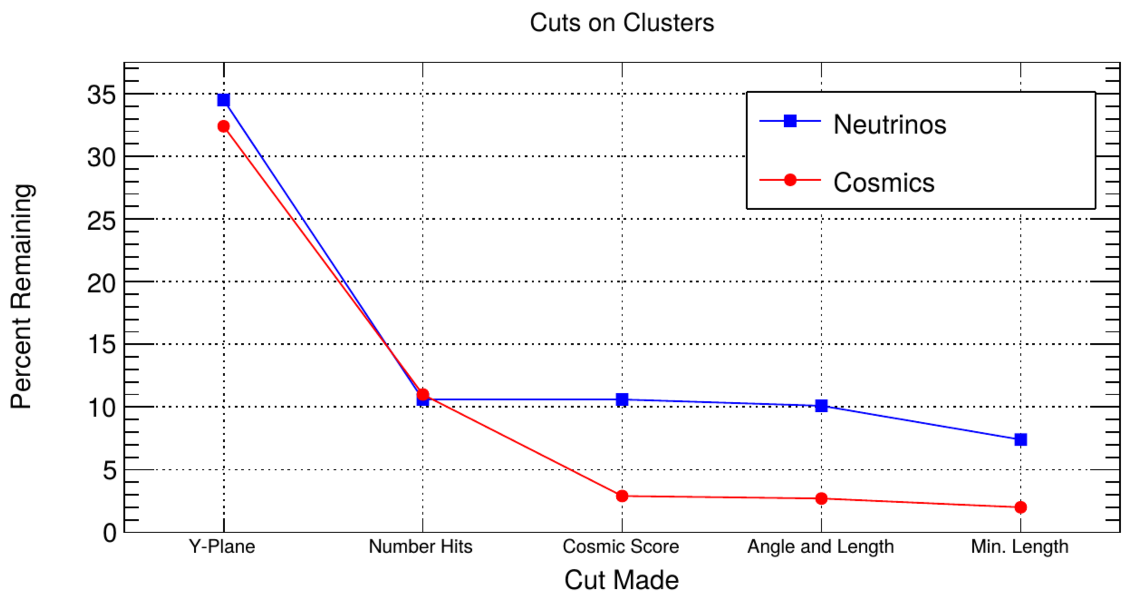


Figure 4.3: Percent of good clusters remaining for neutrinos and cosmics after the primary cuts were applied. This is relative to total number of initial clusters.

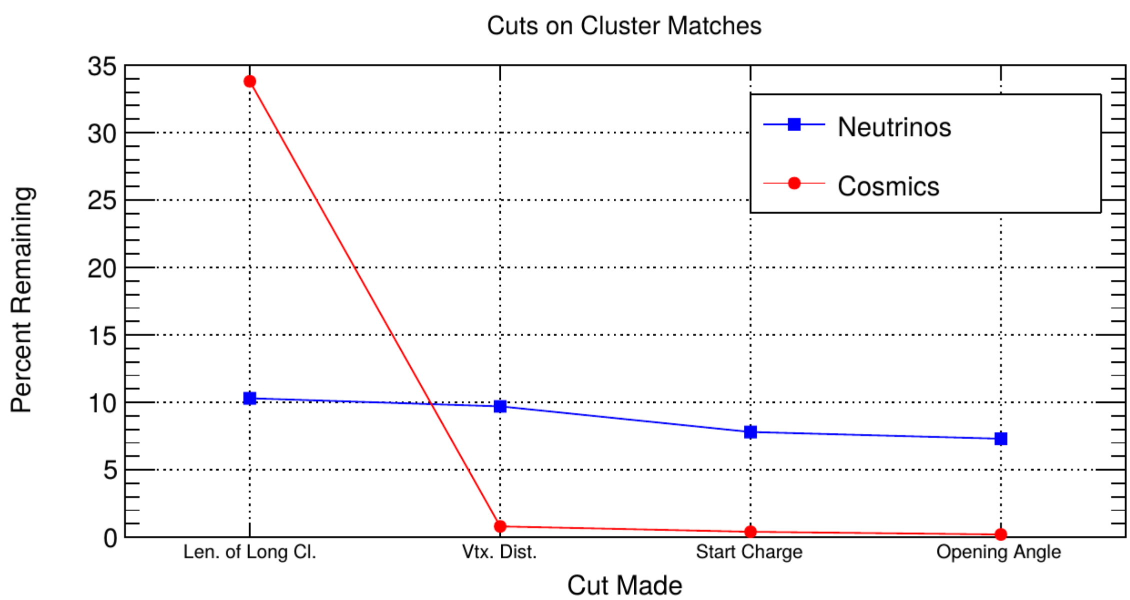


Figure 4.4: Percent matched cluster pairs remaining for neutrinos and cosmics after secondary cuts applied. This is relative to the number of events that contain clusters which pass the primary cuts.

The maximum distance of all three is then selected as the important characteristic per trio. The best trio is the one that has the smallest maximum distance. The $min(max_d)$ for all trios in an event were plotted for BNB neutrino events and for cosmics to find the best cut value for each tracking algorithm. The distribution of $min(max_{d,i})$ is smaller for neutrinos than for cosmics. The cut values for different tracking and clustering algorithms are shown below. These cut values were chosen to minimize the cosmic background to 20%.

- trackkalmanhit with cccluster $min(max_{d,i})$ < 3 cm.
- trackkalmanhit with pandoraNu $min(max_{d,i})$ < 4.5 cm.
- pandoraNu with cccluster $min(max_{d,i})$ < 5 cm.

4.2.4 TPC Updates

After doing a visual hand-scanning of the first beam data processed with the filters detailed above, the events passing had a larger contamination of background than expected. This was mainly in part due to the reconstruction performing better on simulation than on data. Due to this, additional cuts on both streams needed to be implemented in order to increase signal/background ratio. These cuts were added on top of the filters described above and further reduce the event count.

2D Filter Updates

The main background observed in the 2D filter were Michel events, where the muon and electron formed two connected clusters. These events were rejected by comparing the start and end charge deposition of the long cluster (i.e muon particle). The start charge deposition must be less than the end charge deposition. This cut is implemented because muons have a higher ionizaiton loss at the end.

3D Filter Updates

It was seen that cosmic tracks can often originate or end at the same point, therefore faking a signal. Cosmic tracks, however, are mostly vertical. By requiring the angle of the longer track have a cosine greater than 0.85 with respect to the z-axis as well as requiring the longer track to have a length greater than 10 cm, we can reduce this background.

4.3 Conclusion

After proccesing these filters in parallel, it was shown that the 3D filter had a higher purity than the 2D filter because of the higher cosmic rejection being used due to 3D reconstruction. The 2D filter is blind to track entering/exiting from the top or bottom of the TPC. Although the 3D filter had a higher purity, the 2D filter was still able to find identifiable events in data that were used as public event displays. A sample of event displays are shown in figures 4.5 and 4.6.

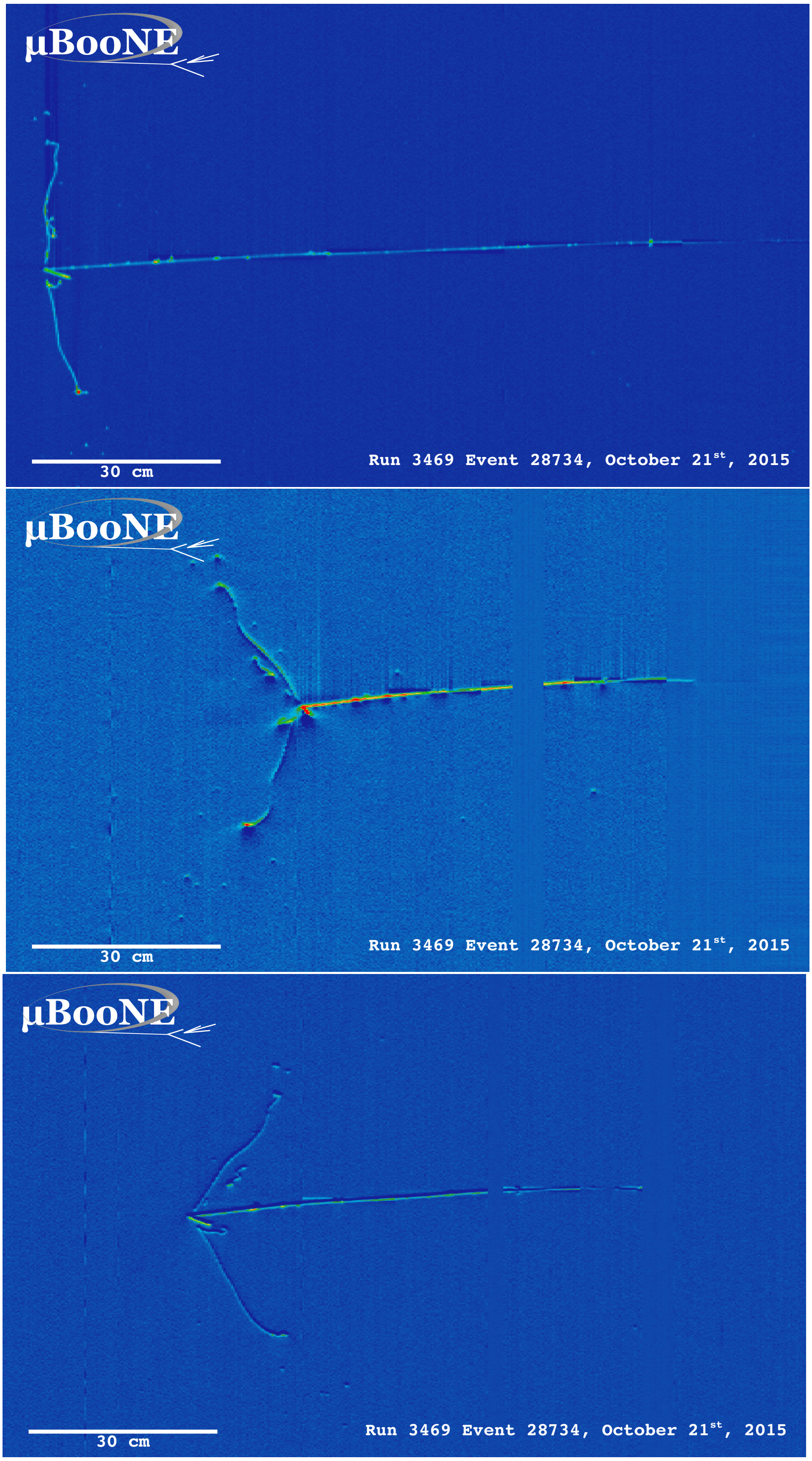


Figure 4.5: First Neutrino Interaction Candidate Events from MicroBooNE

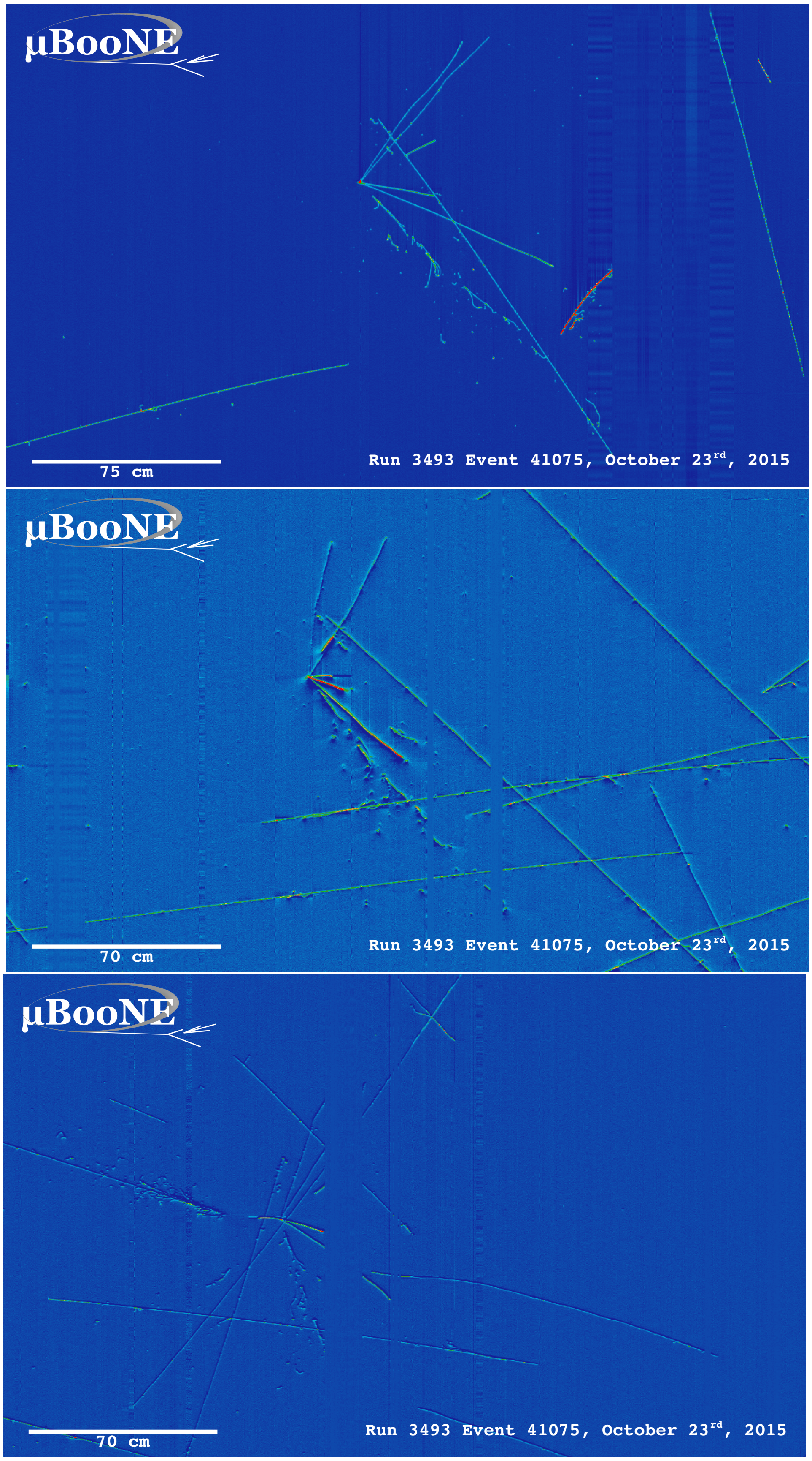


Figure 4.6: First Neutrino Interaction Candidate Events from MicroBooNE

Chapter 5

CC-Inclusive Cross Section Selection Filter

The CC-Inclusive cross-section selection I and selection I modified filters used in this analysis will be described in the following sections below. These filters are an expansion of the Neutrino ID filter. The work done in this thesis was to further improve these selections by increasing both efficiency and purity as well as increasing acceptance without further affecting the kinematic distributions of the selected neutrino events.

MicroBooNE requires fully automated event reconstruction and selection algorithms for use in the many physics measurements being worked on to date due to the large data rate MicroBooNE receives. Being able to automatically pluck out the neutrino interaction among a sea of cosmics proved to be challenging but was accomplished. MicroBooNE has developed two complementary and preliminary selection algorithms to select charged-current $\nu_\mu - Ar$ interactions. Both are fully automated and cut based. The results of this thesis will focus on selection I and selection I modified and will focus on further improving these algorithms using Convolutional Neural Network (CNN) implementations. These selections identify the muon from a neutrino interaction without biasing towards track multiplicity. To combat cosmic and neutral current background, the analysis is strongly biased towards forward-going long tracks which are contained. This limits phase space and reduces acceptance.

5.1 Data and MC Processing Chain

The data used for this analysis were based on hardware and software triggers. Events used came from the *BNB_INCLUSIVE* and *EXT_BNB_INCLUSIVE* streams and were used for signal and background. The *BNB_INCLUSIVE* stream is chosen by requiring that the hardware trigger bit is fired and that the event passed an optical software trigger within a BNB spill timing window. The *EXT_BNB_INCLUSIVE* stream requires the EXT hardware trigger to fire as well as pass the same optical software trigger within a BNB spill size timing window similar to the *BNB_INCLUSIVE*.

The two MC samples used in this analysis and for determining selection efficiencies and purities were GENIE BNB neutrino interactions with CORSIKA cosmic ray overlay within the readout window and inTime CORSIKA cosmic rays. The MC samples generated used ***uboonecode v04_36_00*** and are based on the following packages:

- larsoft v04_36_00
- GEANT v04_09_06_p04d
- GENIE v02_08_06d
- GENIE xsec v02_08_06a
- pandora v02_03_0a
- CORSIKA v07_4003

Both data and MC samples were processed using the same reconstruction release, ***uboonecode v05_08_00*** and the fcl files used for reconstruction are listed below:

- MC fcl files
  - reco_uboone_mcc7_driver_stage1.fcl
  - reco_uboone_mcc7_driver_stage2.fcl
- Data fcl files
- reco_uboone_data_Feb2016_driver_stage1.fcl
- reco_uboone_data_Feb2016_driver_stage2.fcl

On top of the hardware and software triggers, the data also had to pass more criteria to be identified as part of the good run list. The criteria is detailed below.

- **Detector conditions:** the detector has to be in a good operating condition. The detector conditions are read from the slow monitoring database and are required to be within the alarm thresholds. The variables of interest for events passing the good run list criteria include DAQ, PMT, HV, Drift HV, wire bias, electron lifetime and detector power. These conditions need to be met on a run-by-run basis in order to pass the selection.
- **Data quality:** normal and stable behavior for basic reconstruction quantities. These reconstruction variables include average number of tracks, hits, and flashes in each event, the average length of tracks, the average amplitude and area of hits, the average PE and the average spread of each one of these quantities.
- **Beam Conditions:** the BNB must be on and stable and the POT per spill needs to above the intensity threshold. Beam quality conditions include checking the fraction of proton beam interacting within the target, the horn current, and the intensity of protons per spill. The final sample is $5 * 10^{19}$ and a per-spill intensity of $4 * 10^{12}$
- **Run processed:** the full run must be processed completely without missing subruns or crashes in the data processing.

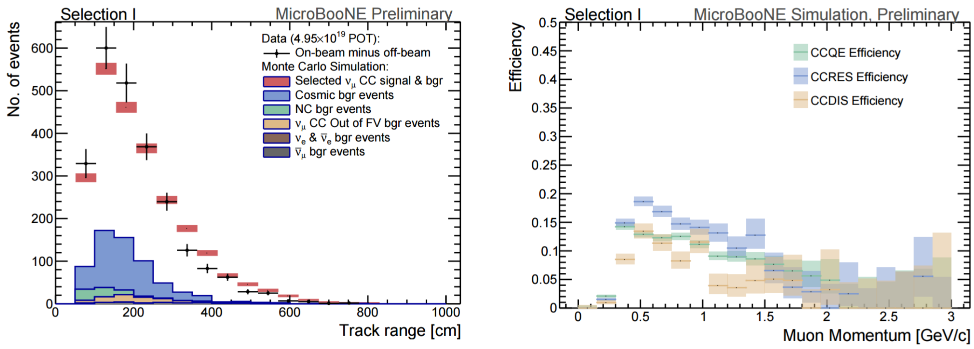
5.2 Normalization of data and MC

The off-beam sample is used to measure beam unrelated backgrounds. For normalization, one needs the total number of BNB spills (N_{BNB} and the total number of external triggers. The BNB spills used need to pass the beam quality cuts. The normalization factor is then N_{BNB}/N_{EXT} which is 1.23.

To normalize generated BNB MC events to POT, we used the following:

- $5 * 10^{19} POT = 41524.3$ generated events

where this scaling factor only applies to mcc7 generated events. The inTime cosmic sample is normalized with respect to the open cosmic sample so an understanding of both is necessary. The POT per beam spill for mcc7 BNB samples is $5 * 10^{12}$. To calculate how many spills are necessary to produce a specific POT one would multiply the total POT by the average 1/POT per spill. For a total POT of $5 * 10^{19}$ the amount of spills necessary is $\frac{5*10^{19}}{5*10^{12}} = 1 * 10^{7}$. This is only one in ~ 241 events therefore each cosmic event needs to be scaled up by a factor of 240.8 when comparing to BNB MC. For inTime cosmics however, two filters are applied to reduce computing and processing time and only leave cosmics that will interact within the detector. The passing rate after these two filters is 0.02125, therefore the total inTime cosmic scaling factor to compare inTime cosmics to BNB is $0.02125 * 240.8 = 5.12$.



(a) Track range distribution of selection I

(b) Selection efficiency as a function of the true muon momentum

Figure 5.1: 5.1a Track range distribution for selection I. The track range is defined as the 3D distance between the start and end of the muon candidate track. No data is shown below 75 cm due to the track length cut described previously. 5.1b Efficiency of the selected events by process quasi-elastic (QE), resonant (RES), and deep-inelastic (DIS). Statistical uncertainty is shown in the bands and the distributions are a function of true muon momentum. The rise of the efficiency between 0 GeV and 0.5 GeV is due to the minimum track length cut and the decreasing efficiency for higher momentum tracks is caused by the containment requirement.

5.3 Optical Software Trigger and Reconstruction

5.3.1 Software Trigger

Most of the BNB spills from the accelerator do not have a neutrino interaction in MicroBooNE. To save computation resources and reduce data-rates, we require a burst of light in the light collection system in coincidence with the 1.6 μs beam spill. Requiring light activity in coincidence with the beam spill eliminates the vast majority of triggers with no neutrino interaction in the detector, however, it doesn't guarantee the activity in the detector is a neutrino interaction since a cosmic ray can interact in coincidence with the beam spill as well.

To implement this, a software trigger was used on the PMT waveforms to decide whether or not to keep that event. The software trigger is implemented after the event builder combines data from the PMTs and triggers into a single event. The software trigger uses the digitized output of the 32 PMT channels in the light collection system. Only the waveform region in coincidence with the beam spill is used to search for possible triggers. For each PMT, a waveform is found by taking the difference of ADC values is calculated between t and $t + s$. This waveform is then scanned for ADC values above a threshold X_0. Once an ADC is above this threshold, a discriminator window is opened for a fixed number of time ticks (W_0). If the ADC count within this window W_0 is greater than a second larger threshold X_3, a final window of width W_3 is opened. The max ADC value within this final window is set as the peak amplitude for the PMT and then summed across all 32 PMTs and set to the variable PHMAX. The software trigger places a final cut on the PHMAX variable to decide whether or not to keep the event. The thresholds were found by the Trigger task force using Monte Carlo Studies and are as follows:

- X_0 = 5 ADC
- X_3 = 10 ADC
- W_0 = 6 Ticks
- W_3 = 6 Ticks
- PHMAX cut = 130 ADC

5.3.2 Flash Reconstruction

MicroBooNE collects light from each of the 32 PMTs either in a continuous readout window of 23.4 μS activated by a beam gate signal on the trigger board, or in discriminated pulses of $\sim$1 μs duration activated if the ADC count for any PMT goes above 80 ADC count. These two formats are saved as output waveforms and put onto an event. Additionally, each PMT can provide two output streams, high-gain ($\sim$20 ADC/PE) and low-gain ($\sim$2 ADC/PE) channels. The first step in the reconstruction is to merge both these channels into a "saturation corrected waveform" which uses information from the low-gain waveform to correct for saturating high-gain pulses.

The saturation corrected waveform in the continuous readout window is used to reconstruct optical hits. Each PMT's waveform is scanned for hits then a threshold based hit reconstruction algorithm is applied which requires pulses of a minimum area in order to be reconstructed. Each reconstructed hit is associated to a PMT, a time in µs, and a PE count.

Once hits are reconstructed for all 32 PMTs, all PMT information is then combined into optical flashes which represent optical information seen by the PMTs from interactions in the detector. Each flash has information on total light seen per interaction, the distribution of the light across all 32 PMTs, the flash time with respect to the trigger time of the flash, and lastly, the spacial information of the flash in Y-Z plane of the detector. These flashes are reconstructed by requiring that there is a ~ 1 µs coincidence between the reconstructed hits in all 32 PMTs. The total PE is summed up among all coincident hits across the PMTs and if the total PE is greater than 2 PE, a flash is reconstructed. There are also safe guards in place to take care of late scintillation light.

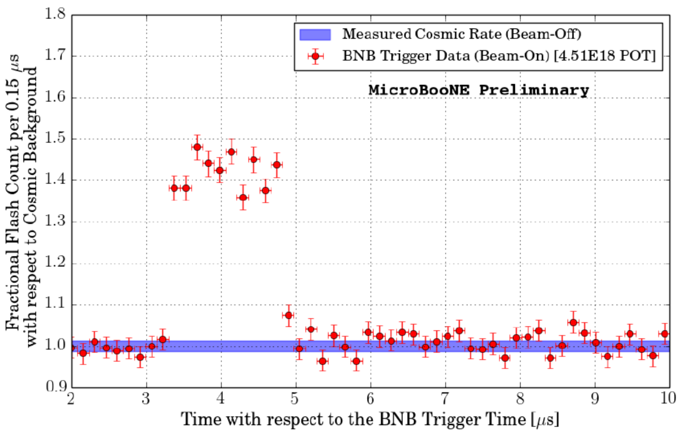


Figure 5.2: Time distribution of reconstructed optical flashes with a PE value of 50 or more for a sample of BNB unbiased triggered events.

Figure 5.2 shows the time distribution of reconstructed optical flashes using the BNB continuous stream. You can see a clear excess in coincidence with the expected arrival time of neutrinos. The same flash reconstruction that was used in the cc-inclusive filter detailed here was used to create this plot in data.

5.3.3 Beam Window

Figure 5.3 shows the distribution of flashes for on-beam, off-beam and various MC samples. The software trigger has been applied to these samples. The pile-up seen just after 0 µs is a feature of the flash finding algorithm and consists of low PE flashes and

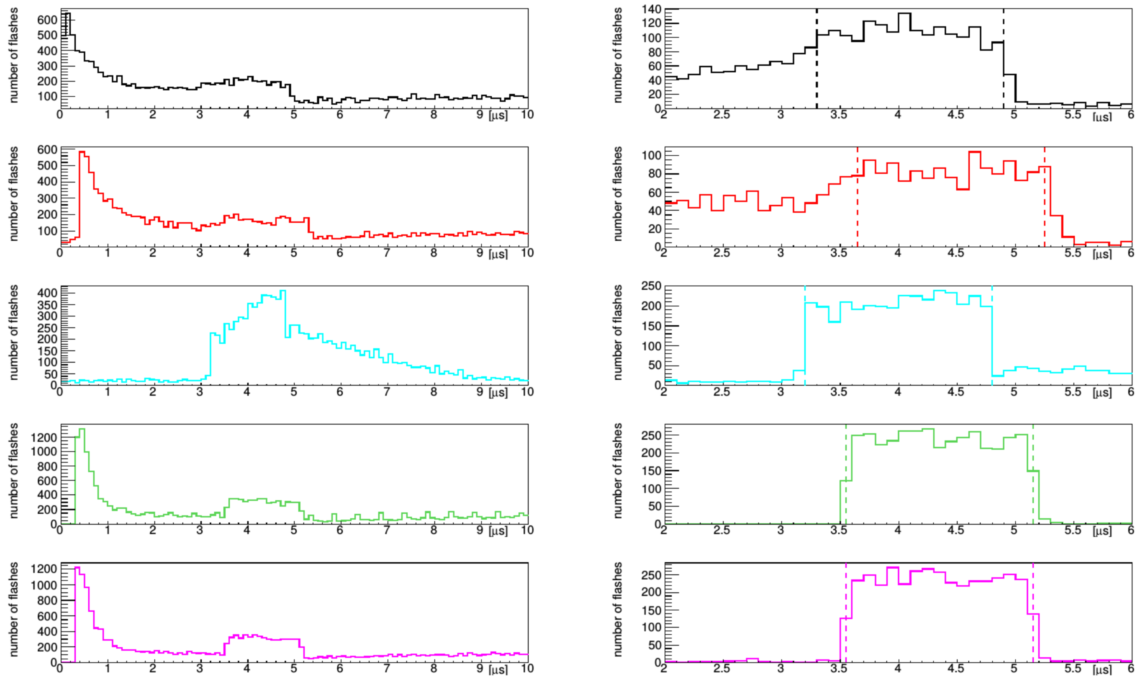


Figure 5.3: Flash time distribution for all flashes (left plot) and flashes > 20PE (right plot). The different curves are as follows: on-beam data (black), off-beam data (red), CORSIKA inTime MC (light blue), BNB only MC (green), and BNB+Cosmic MC (purple). The dashed vertical lines mark the time window that was chosen for each sample

is removed in the second column of distributions with a low 20 PE threshold cut. The plots show that the time window for the distributions are shifted a small amount from each-other. This is caused by different hardware configurations per sample. Using these distributions, the windows chosen per sample are as follows:

- On-Beam: 3.3 to 4.9 μs
- Off-Beam: 3.65 to 5.25 μs
- CORSIKA inTime: 3.2 to 4.8 μs
- BNB only: 3.55 to 5.15 μs
- BNB+Cosmic: 3.55 to 5.15 μs

Each window has a width of 1.6 μs.

5.4 TPC Reconstruction

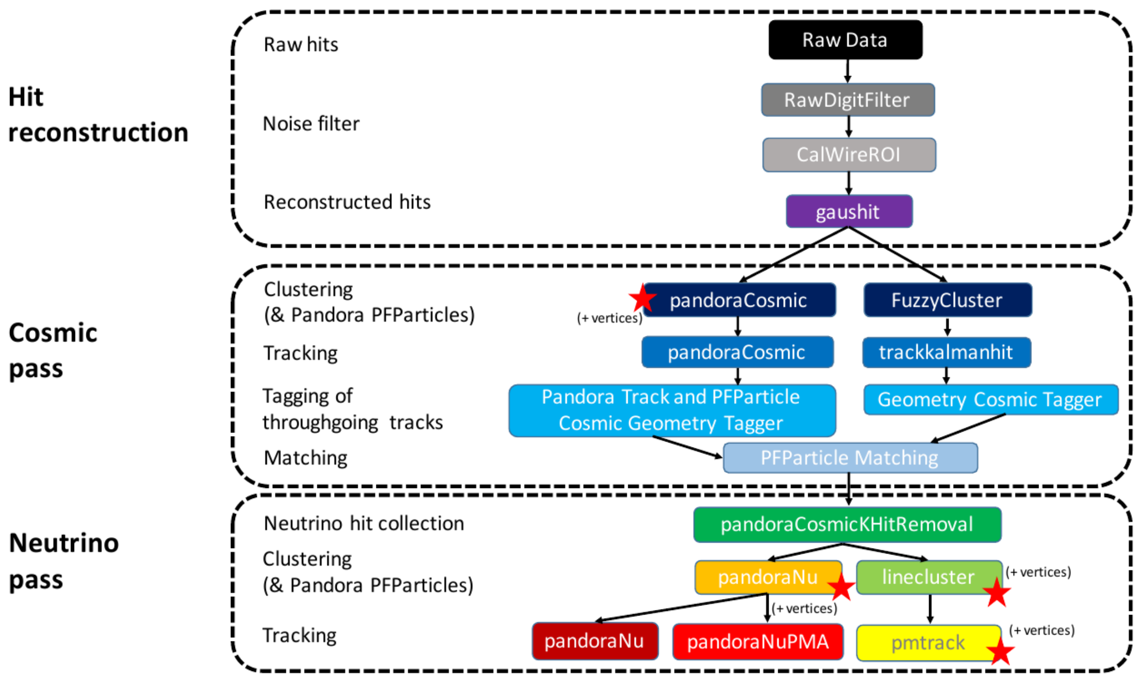


Figure 5.4: Reconstruction chain run on both data and MC. The red stars mean that the algorithm returns reconstructed 3D vertices.

Figure 5.4 summarizes the reconstruction chain applied to both MC and data for this analysis. After the hit reconstruction, a cosmic pass is applied which removes all hits associated to through-going tracks. A description of these TPC reconstruction algorithms will be detailed below.

5.4.1 Hit Reconstruction

The waveforms used for hit reconstruction consist of charge deposited on the sense wire in drift time. The first step in hit reconstruction is to pass the waveforms through a filtering algorithm to filter out the noise introduced from the electronics. The input waveforms are also truncated from 9600 time ticks to 6400 time ticks in this first step to reduce the data footprint of these waveforms.

Once noise filtering is complete, a deconvolution algorithm is applied to the waveforms to remove the drift field and electronics response, therefore leaving only the ionized electrons kicked off the argon atoms by an incident track. During this process, Region of Interests (ROI) are identified and cut out of the waveforms to further reduce the data volume.

The hit finding algorithm then finds candidate peaks in these ROI's and fits the peaks to Gaussian curves. These Gaussian shaped peaks are now called hits and represent the charge deposition on a wire by the incoming track. These hit objects have a peak time and width and are the basic object input to further algorithms down the reconstruction chain.

5.4.2 Clustering

There are multiple clustering algorithms used in this analysis. The main purpose of all the clustering algorithms is to associate hits together in 2D space to create objects like tracks, vertices and showers. For the fuzzy cluster algorithm, three steps are used to achieve this. The first step is to associate hits to each-other using a fuzzy clustering algorithm which gives each hit a degree of belonging to the cluster. Second, a Hough transform is used to find hits associated to candidate tracks and showers within each of the clusters found in the first step. The last step merges smaller candidate tracks and showers into large clusters. The last step also associates unclustered hits into nearby objects which helps shower reconstruction. The result is a set of clusters made up of associate hits that represent tracks or showers per plane.

The pandora algorithm utilizes it's own clustering algorithm and will be detailed in the next section. The last clustering algorithm is called linecluster. The linecluster algorithm reconstructs 2D linear clusters per plane by fitting a line onto nearby hits which is then extrapolated to neighboring wires. 2D vertices are found per plane by using the intersection points of the ends of nearby clusters. These 2D vertices are then matched in time across all three planes to get a 3D vertex in space.

5.4.3 Pandora

5.4.4 Trackkalmanhit

The trackkalmanhit algorithm takes 2D clusters returned from the fuzzy cluster algorithm and outputs track objects. There are no hierarchy structure as it is in pandora, each track is independent. There also is no vertex reconstruction with this algorithm as well.

5.4.5 Cosmic Hit Removal

The Pandora algorithm is applied to the events twice, the first to remove downward going tracks primarily from cosmic ray muon like particles. The second pass only runs on a subset of hits that aren't associated with cosmic ray muon tracks.

After the first pass, the output of PFParticle hierarchy is then passed to a cosmic ray tagger to look through all hits to determine start and end points. If the start or end point trajectories are consistent with entering or exiting the TPC, then these hits are removed from the second pass. Hits are considered entering or exiting the TPC if the drift time are outside of the neutrino drift window or outside of the fiducial volume of the TPC. The fiducial volume was based on a montecarlo study and is 20 cm from the top or bottom of the TPC and 10 cm from the TPC ends. Hits associated with candidate cosmic ray tracks are removed from the input hit collection and the remaining hits are passed to the neutrino optimized pass of Pandora.

5.4.6 Projection Matching Algorithm

The projection matching algorithm (PMA) was inherited from ICARUS and has been implemented in LArSoft. PMA differs from traditional LArSoft 3D reconstruction algorithms. Most 3D reconstruction attempts to match 2D objects from all three planes by drift time, while the PMA algorithm projects a track hypothesis on each plane then the distance between this projection and the hits on each plane is minimized simultaneously. More information can be found in [**?**].

5.5 Event Selection

The first requirement for selecting ν_μ CC events is that the event has at least one scintillation light flash in the beam trigger window with more than 50 PE on all PMTs combined. From the flashes that pass, the most intense is chosen and considered to be originating from a neutrino interaction and will be the only flash used in further cuts.

Vertices are then required to have at least one reconstructed track start or endpoint within a 5 cm radius. Showers associated with a vertex do not pass this cut. All tracks associated with a vertex are then used to calculate a track length weighted average of the θ-angle. Of all the vertices that do pass, only the vertex with the most forward going θ-angle average of all associated tracks is considered the neutrino vertex candidate. The most forward going θ-angle average is chosen by picking the largest track range weighted average of $|cos(\theta)|$, seeing as $cos(\theta) = 1$ is the beam direction. Next, it is required that the reconstructed neutrino vertex candidate be within the fiducial volume as well as within the drift time starting at t_0. The fiducial volume boundaries chosen are 10 cm from the edges of the TPC in x and z which is the drift direction and beam direction respectively, and 20 cm from the edges of the TPC in y which is the vertical direction. For all further cuts, only the longest track associated with the neutrino vertex candidate and this track is assumed to be the muon candidate of the neutrino event.

The next cut requires the position of the flash in the z-direction and the track z-projection to be compared. This basic flash matching algorithm is rudimentary and a placeholder for a more sophisticated algorithm. The z-position of the flash needs to be within 80 cm to the z-positions of track start or endpoints. If the flash is between the track start and endpoint, the distance of the flash to the track is considered to be 0 cm.

Lastly, the track needs to be fully contained within the fiducial volume and have a track range greater than 75 cm. The range is the 3D distance between the track's start and endpoint. The length cut was optimized to remove NC background that contain a pion due to the pion interaction rate to be $\sim$70 cm. A track that makes all the cuts is considered to be the muon of a ν_μ CC event. The list of cuts for this selection is described below:

1. At least one falsh > 50 PE within the beam gate.
2. At least one track witin 5 cm around a vertex.
3. Vertex with flattest tracks is chosen to be vertex candidate.
4. Vertex candidate in fiducial volume.
5. Longest track associated with vertex candidate is chosen to be track candidate.
6. Longest track is within 80 cm (z-axis only) of the flash.
7. Longest track is fully contained.
8. Longest track is greater than 75 cm.

The event selection scheme can also be seen in figure 5.5. Table 5.1 lists the passing rates for MC events for the selection scheme describes above. Table 5.2 lists the passing rates for on-beam and off-beam data for the selection scheme. The normalization factors applied between on-beam and off-beam data are described in section 5.2.

5.5.1 Expected Bakgrounds

Most of the selected background events will be of cosmic origin. Ther are two types of cosmic backgorund, one triggered by a cosmic-ray event ocuuring in the beam gate time window, the other triggered by a beam induced interaction in the cyrostat followed by a misidentification of a cosmic event as a neutrino event. The first cosmic background can be subtracted from the selected events using the off-beam BNBEXT sample normalized to the on-beam. The second cosmic background events are modeled by MC by using BNB+Cosmic MC sample.

Other backgrounds originate from neutrino beam contaminates. A major contribution in this sector is by neutral current neutrino events for example a charged pion track misidentified as a muon. Another contribution are ν_e-like and anti-muon-neutrino

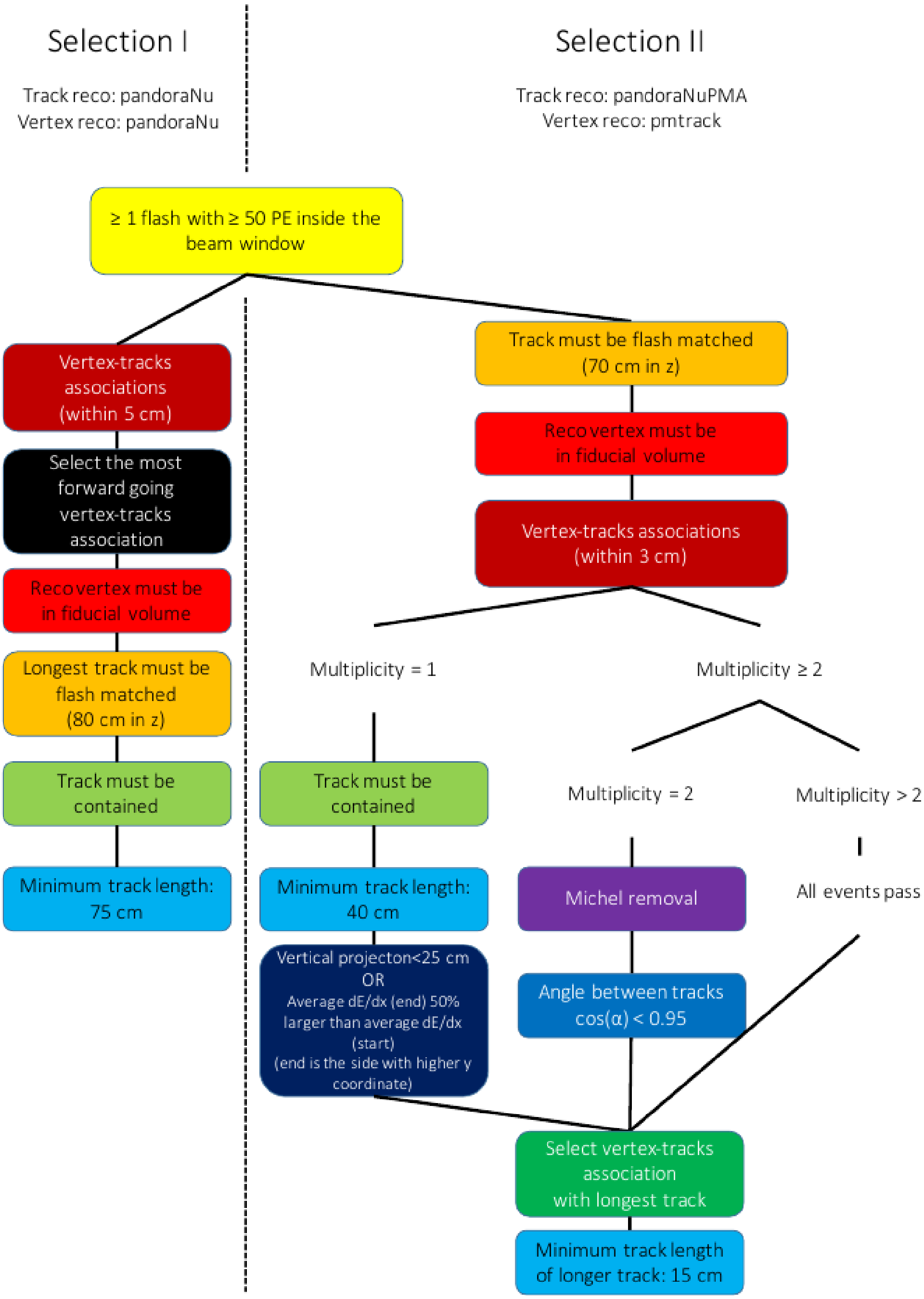


Figure 5.5: Event selection diagram for selection I and selection II. This analysis focused on optimizing selection I. Boxes with the same color across the two selections symbolize similar cuts.

	BNB+Cosmic Selection	BNB+ Cosmic MC-Truth	Cosmic Only	Signal:Cosmic Only
Generated Events	191362	45273	4804	1:22
$\geq$ 1 flash with $\geq$ 50 PE	136219 (71%/71%)	44002 (97%/97%)	2970 (62%/62%)	1:14
$\geq$ 1 track within 5 cm of vertex	135830 (99%/71%)	43974 (99%/97%)	2975 (99%/62%)	1:14
vertex candidate in FV	79112 (58%/41%)	34891 (79%/77%)	1482 (50%/31%)	1:8.9
flash matching of longest track	40267 (51%/21%)	25891 (74%/57%)	340 (23%/7.1%)	1:2.8
track containment	19391 (48%/10%)	11693 (45%/26%)	129 (38%/2.7%)	1:2.3
track $\geq$ 75 cm	6920 (36%/3.6%)	5780 (49%/13%)	17 (13%/0.4%)	1:0.6

Table 5.1: Passing rates of Selection I. Numbers are absolute event counts and cosmic background is not scaled. The BNB+Cosmic sample contains all events, not just ν_μ CC inclusive. The numbers in brackets give the passing rate wrt the step before (first percentage) and wrt total generated events (second percentage). The BNB+Cosmic MC-Truth column shows how many true ν_μ CC inclusive events are left in the sample. This number includes mis-identifications where a cosmic track is picked by the selection instead of the neutrino interaction in the same event. The cosmic only sample is used just to illustrate the cut efficiency. The last column Signal:Cosmic only gives an estimate of the ν_μ CC events wrt the cosmic only background at each step. For this number, the cosmic background has been scaled as described in section 5.2.

	on-beam	off-beam
Generated Events	546910	477819
$\geq$ 1 flash with $\geq$ 50 PE	135923 (25%/25%)	96748 (20%/20%)
$\geq$ 1 track within 5 cm of vertex	134744 (99%/25%)	95778 (99%/20%)
vertex candidate in FV	74827 (55%/14%)	51468 (54%/11%)
flash matching of longest track	22059 (29%/4.0%)	12234 (24%/2.6%)
track containment	10722 (49%/1.9%)	5283 (43%/1.1%)
track $\geq$ 75 cm	3213 (30%/0.6%)	1328 (25%/0.3%)

Table 5.2: Passing rates for Selection I selection applied to on-beam and off-beam data. THe numbers in brackets give the passing rate wrt the step before (first percentage) and wrt the generated events (second percentage). Off-beam data has been scaled with a factor 1.23 to normalize to the on-beam data stream.

events. These beam related backgrounds are an order of magnitude smaller than the cosmic misidentification backgrounds. These backgrounds can not be subtracted and are estimated using MC truth.

The efficiency and purity of Selection I are calculated below:

- Efficiency: Number of selected true ν_μ CC events divided by the number of expected true ν_μ CC events with interaction in the FV.
  - (12.3 $\pm$ 3.4) %
- Purity: Number of selected true ν_μ CC events divided by the sum of itself and the number of all backgrounds.
  - (53.8 $\pm$ 4.4) %

5.5.2 Truth Distributions

The truth distributions of MC truth variables before and after the selection are detailed in this section. The overall efficiencies are calculated for all ν_μ CC signal events with a true interaction within the fiducial volume and a fully contained muon track originating from said vertex. Figures 5.6 through 5.8 detail the truth distributions for muon momentum, $cos(\theta)$ and ϕ and figures 5.9 through 5.11 detail the total efficiency of the selection for charged current quasi elestic (CCQE) events, charged current resonant (CCRES) events, and charged current deep inelastic (CCDIS) events.

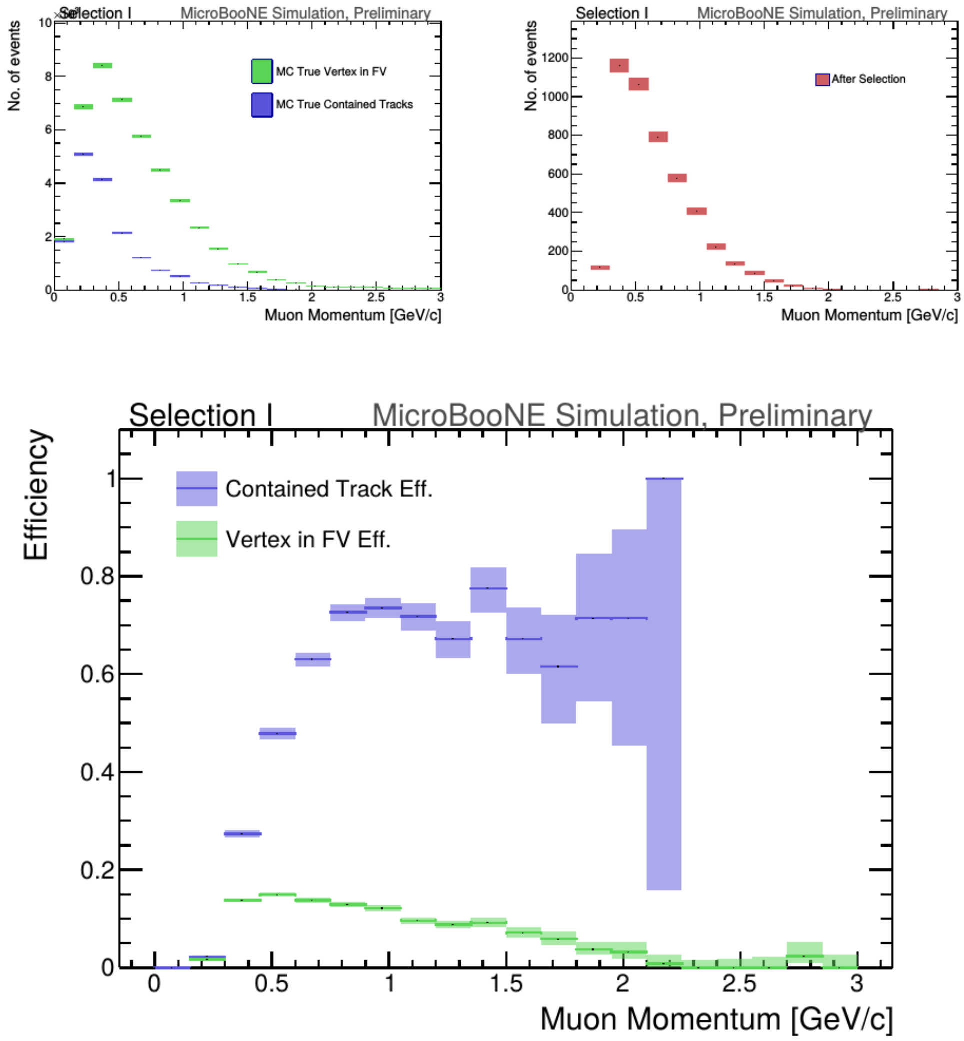


Figure 5.6: MC momentum distributions of the muon originating from a ν_μ CC interaction. Upper left is the momentum distributions of events with a vertex within the FV (green) and the events with a fully contained track (blue) before the selection. The upper right side is the momentum distribution after the selection (red). The lower plot is the selection efficiencies.

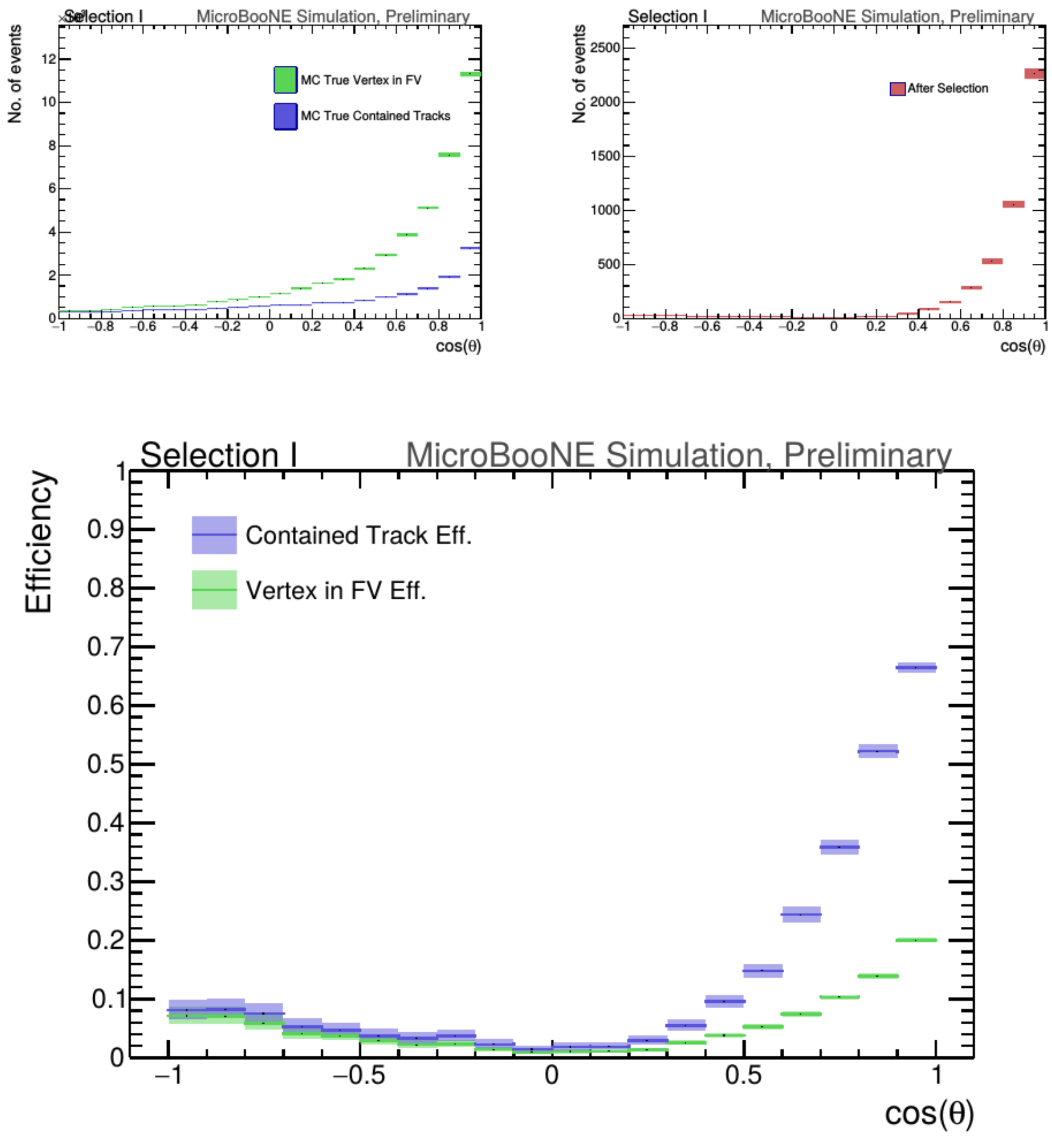


Figure 5.7: MC $cos(\theta)$ distributions of the muon originating from a ν_μ CC interaction. Upper left is the $cos(\theta)$ distributions of events with a vertex within the FV (green) and the events with a fully contained track (blue) before the selection. The upper right side is the $cos(\theta)$ distribution after the selection (red). The lower plot is the selection efficiencies.

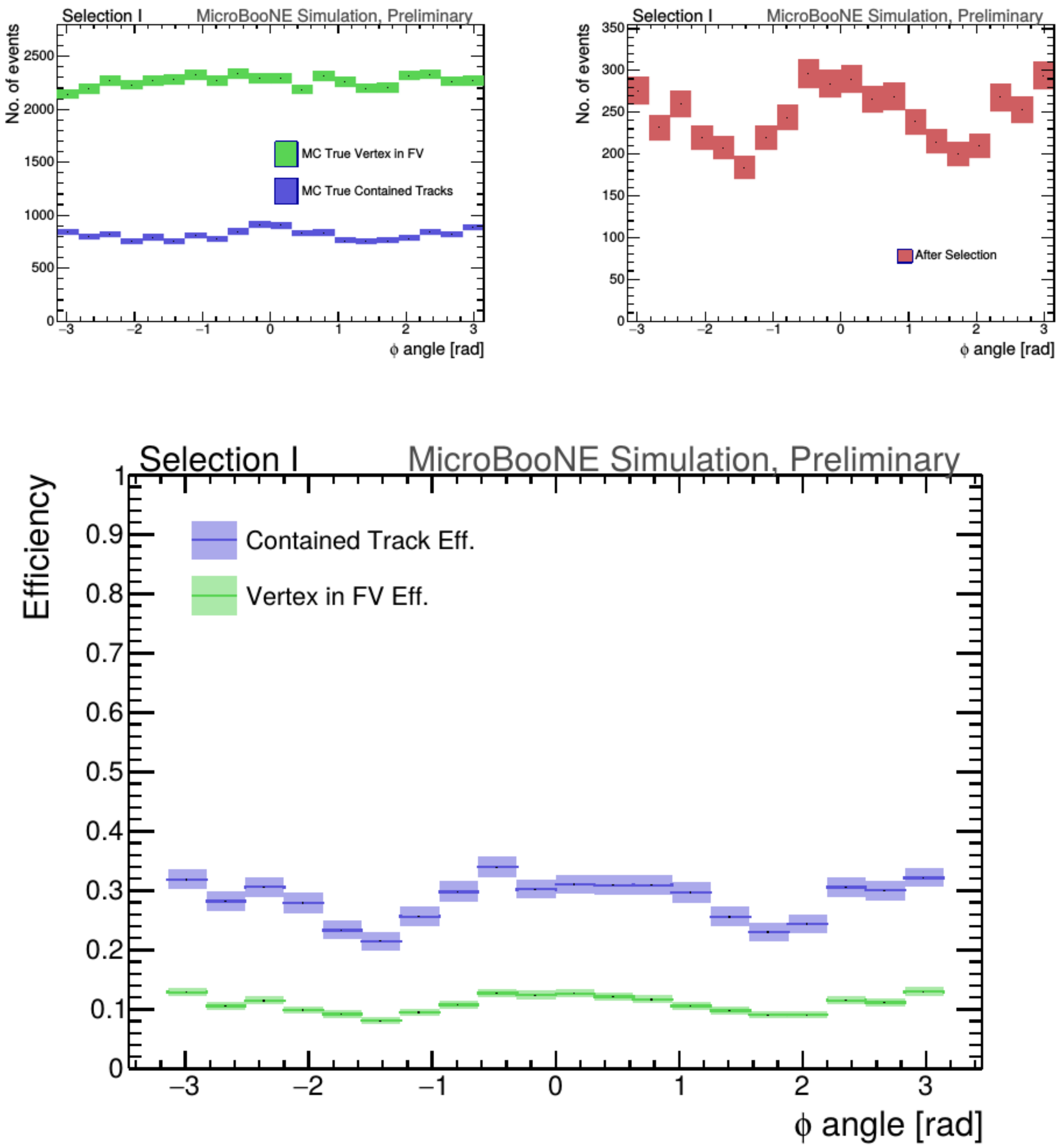


Figure 5.8: MC ϕ distributions of the muon originating from a ν_μ CC interaction. Upper left is the ϕ distributions of events with a vertex within the FV (green) and the events with a fully contained track (blue) before the selection. The upper right side is the ϕ distribution after the selection (red). The lower plot is the selection efficiencies.

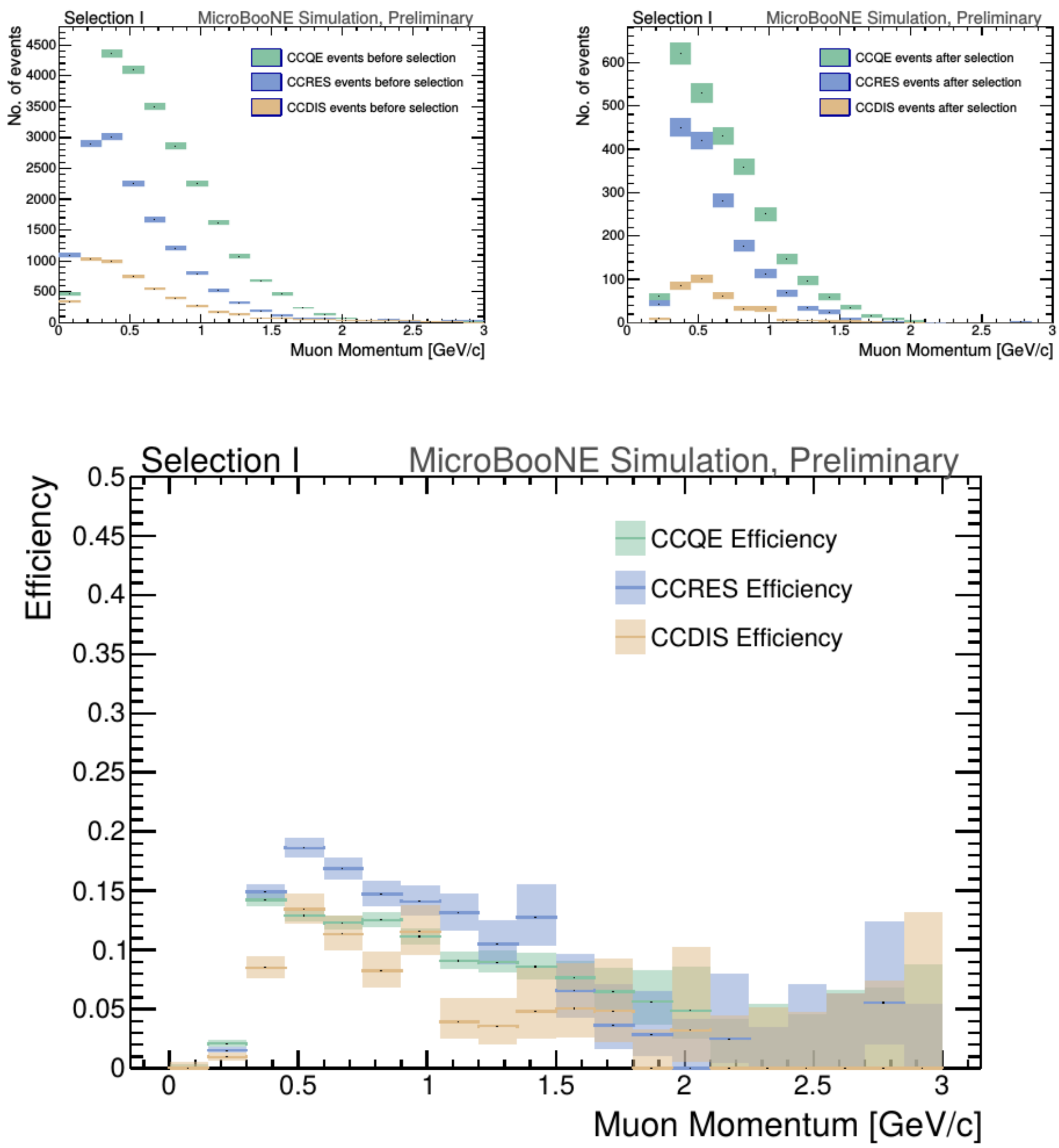


Figure 5.9: MC momentum distributions of the muon originating from a ν_μ CC interaction. Upper left is the momentum distributions of events with a vertex within the FV split up into CCQE (red), CCRES (yellow), and CCDIS (green) before selection. The upper right side is the momentum distribution after the selection with the same color schemes. The lower plot is the selection efficiencies for all three interaction types. The definition of QE, RES, and DIS is based on GENIE.

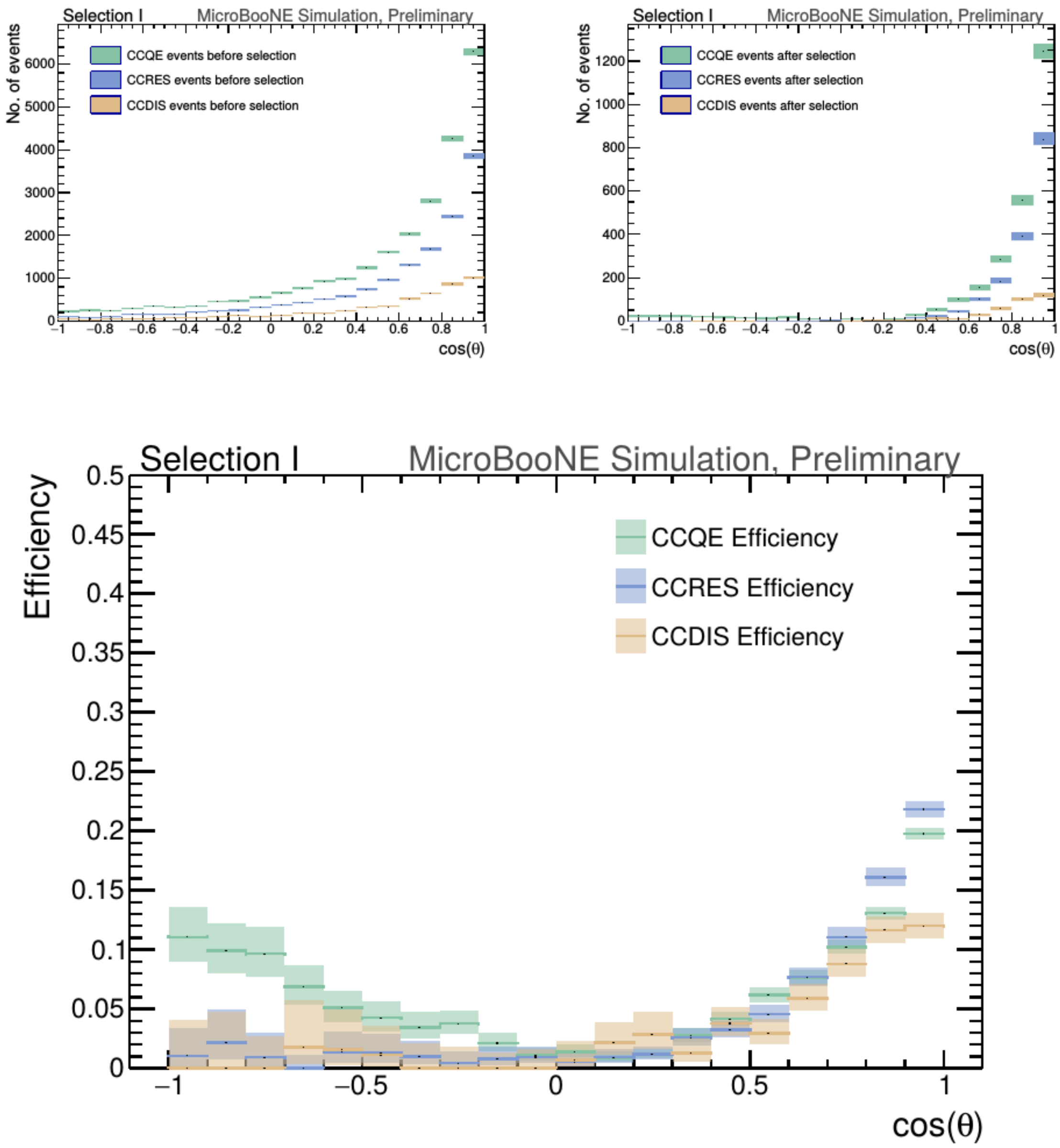


Figure 5.10: MC $cos(\theta)$ distributions of the muon originating from a ν_μ CC interaction. Upper left is the $cos(\theta)$ distributions of events with a vertex within the FV split up into CCQE (red), CCRES (yellow), and CCDIS (green) before selection. The upper right side is the $cos(\theta)$ distribution after the selection with the same color schemes. The lower plot is the selection efficiencies for all three interaction types. The definition of QE, RES, and DIS is based on GENIE.

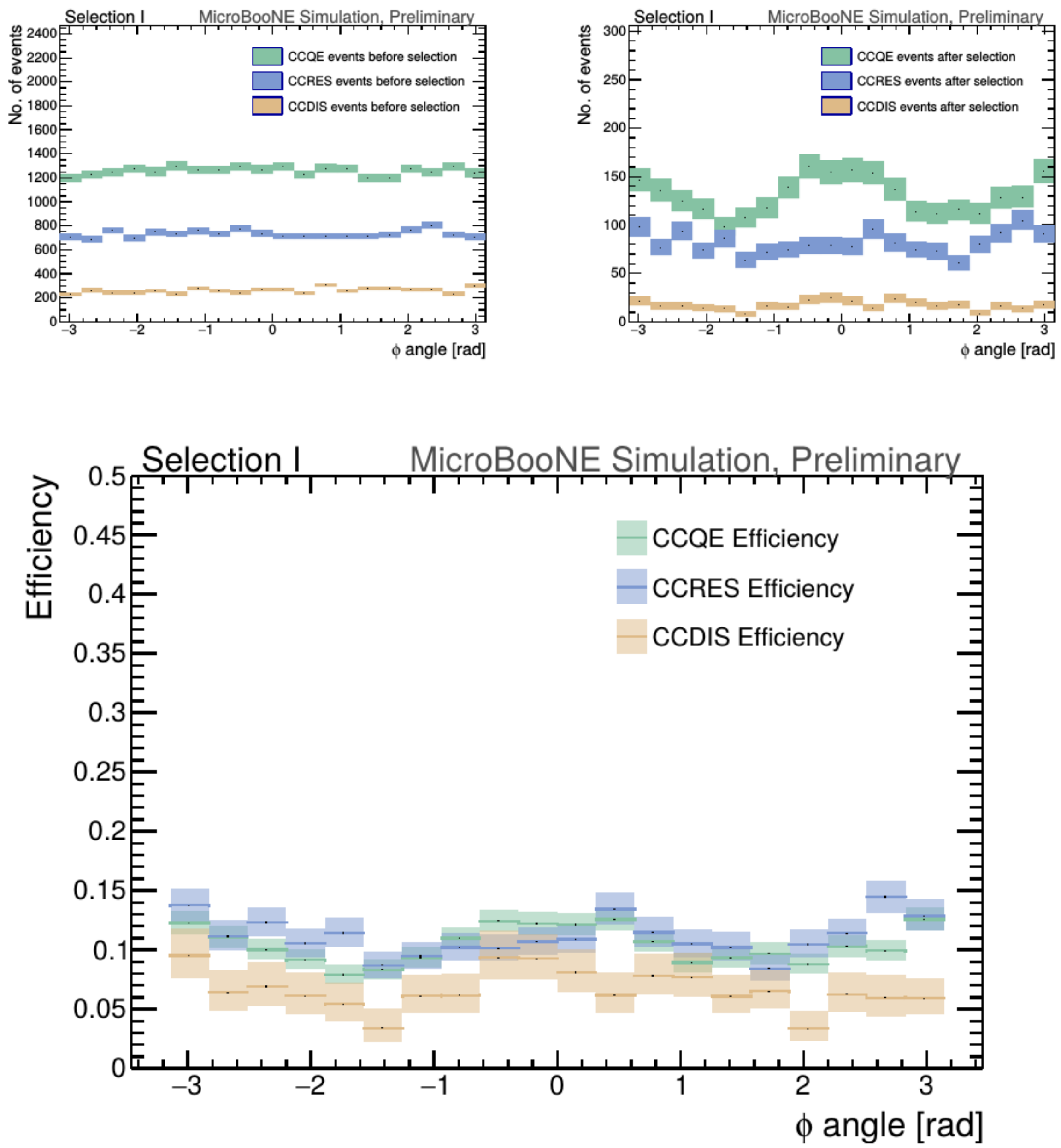


Figure 5.11: MC ϕ distributions of the muon originating from a ν_μ CC interaction. Upper left is the ϕ distributions of events with a vertex within the FV split up into CCQE (red), CCRES (yellow), and CCDIS (green) before selection. The upper right side is the ϕ distribution after the selection with the same color schemes. The lower plot is the selection efficiencies for all three interaction types. The definition of QE, RES, and DIS is based on GENIE.

Chapter 6

Background on Convolutional Neural Networks

Convolutional neural networks (CNNs) have been one of the most influential innovations in the field of computer vision. Neural networks became popular in 2012 when Alex Krizhevsky used them to win that year's ImageNet competition [**?**] by dropping the error from 26% to 15%. Since then, many companies are using deep learning including Facebook's tagging algorithms, Google for their photo search and Amazon for product recommendations. For the purpose of this thesis CNNs were used for image classification, specifically, images of varying particles created using LArTPC data.

6.1 Image Classification

Image classification is the process of inputting an image into the CNN an receiving a probability of classes that best describes what is happening in the image. As humans, image classification is something that is learned at a very young age and is easy to do without much effort. This is also apparent when hand-scanning LArTPC images. After learning what a neutrino event looks like in MicroBooNE, it is relatively easy to recognize simple neutrino events from cosmic ray background as well as highly ionizing particles like protons from minimum ionizing particles like muons. The very detailed images LArTPC detectors output are prime candidates for input images into a CNN. CNNs mimic a human's ability to classify objects by creating an architecture that can learn differences between all the images it's given as well as figure out the unique features that make up each object. CNNs are modeled after the visual cortex. Hubel and Wiesel [?] found that there are small regions of neuronal cells in the brain that respond to specific regions of the visual field. They saw that some neurons fired when exposed to vertical edges while others fired when shown horizontal or diagonal edges. They also saw that these neurons were organized in columns. The idea of specific neurons inside of the brain firing to specific characteristics is the basis behind CNNs.

6.2 CNN Structure

When used for image recognition, convolutional neural networks consist of multiple layers that extract different information on small portions of the input image. How many layers is tunable to increase the accuracy. The output of these collections are then tiled so that they overlap to gain a better representation of the original image and allow for translation. The first of these layers is always a convolution layer. To the CNN, an image is an array of pixel values. For a RGB color image with width and height equal to 32x32 the corresponding array is 32x32x3. Filters, also known as neurons, of any size set by the user is then convolved with the receptive field of the image. If the filter is 5x5, the receptive field will by a patch of 5x5 on the input image. The filter is also an array of numbers called weights. The convolution of the filter and image are matrix multiplications of the weights and the pixel values. By stepping the receptive field by 1 unit, for an input image of 32x32x3 and a filter of 5x5x3 you'd get an output array of 28x28x1. This output array is called an activation map or feature map. The use of more filters preserves the spatial dimensions better. The filters can be described as feature identifiers. Examples of features in an image consist of edges, curves, and changes in colors. The first filters in a CNN will primarily be straight line and curve feature identifiers. An example of a curve filter is shown in figure 6.2. When a curve in the same concavity is found in the input image, the corresponding pixel in the output feature map will be activated. Going back to our example of a 32x32 input image and a 5x5 filter, if there were to be a curve in the top left corner of the input image, our output feature map would have a high pixel value in the top left. Therefore, feature maps tell us where a specific feature is located in the original image. Figure **??** shows a visualization of filters found in the first layers of many CNN architectures. These filters in the first layer convolve around the image and activate whe nthe specific feature it is looking for is in the receptive field.

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

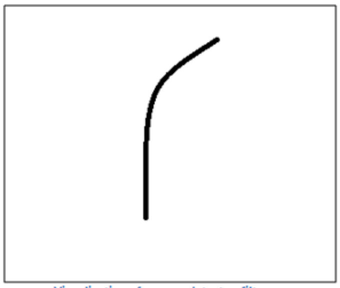


Figure 6.1: Pixel representation and visualization of a curve detector filter. As you can see, in the pixel representation, the weights of this filter are greater along a curve we are trying to find in the input image

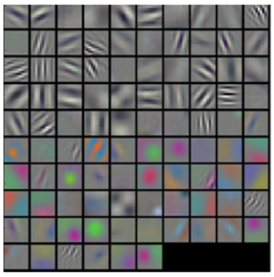


Figure 6.2: Visualization of filters found in first layer of a CNN.

In figure 6.3 you can see how an edge detection filter is used to save only necessary information for recognizing different types of clothes. You can also see by having multiple filters you can get more detail or less detail from an image which can then simplify or complicate the object recognition task. Being able to distinguish between a shirt or a leg garment is as much information you want, having a filter that extracts outline edge or shape information would be all that you need. But if instead you wanted to distinguish between a formal cocktail dress or a summer dress, more information would need to be saved equating to many more filters for one image. Rather than trying to come up with how many filters and what features are important for detection, CNNs do this automatically. CNNs take input parameters, called hyperparameters, for example number of layers, number of filters per layers, number of weights per filter, and uses these to create the output feature maps. The layers build upon each-other, for example if we were creating a CNN for facial recognition the convolutional layers will start learning feature combinations off of the previous layers. The low level features like edges, gradients, and corners of the first layers become high level features like eyes, noses, and hairs. This process is visualized in figure 6.4

Figure 6.3: Applying a feature mask over a set of fashion items to extract necessary information for auto-encoding. Unnecessary information for example color or brand emblems are not saved. This feature map is an edge detection mask that leaves only shape information which helps to distinguish between different types of clothes.

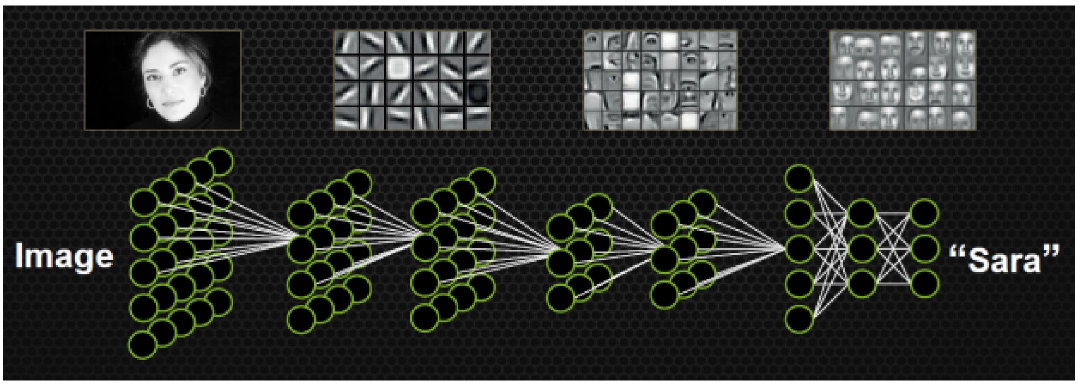


Figure 6.4: Pictorial Representation of Convolutional Neural Networks as well as a visual representation on CNN’s complexity of layer feature extraction

There are other layers in a CNN architecture that will not be covered in the scope of this thesis but in a general sense, these layers are interspersed between convolution layers to preserve dimensionality and control overfitting of the network. The last layer is called a fully connected layer and it’s job is to output an N dimensional vector where N is the number of classes the network has been trained on. Each number in this vector represents the probability that the input image is a certain class. Fully connected layers use the feature maps of the high level features to compute the products between the weights of the previous layer to get the probabilites of each class. These weights are then adjusted throught the training process using backpropagation.

6.2.1 Backpropagation

A CNN at it's onset has weights that are randomized. The filters themselves don't know how to pull out identifying information per class. For a neural network to learn, it must be trained on a training set that is labeled. Backpropagation has four seperate steps: foward pass, loss function, backward pass and updating weights. In the forward pass, a training image is passed through the whole network. All of our weights at this time are randomly initialized so the output for the first image will have no preference to a specific class. A common loss function is mean squared error (MSE):

$$E_{total} = \sum \frac{1}{2}(actualclass - predictedclass)^2 \tag{6.1}$$

If we assume that the MSE is the loss of our CNN, the goal would be that our predicted label (output of CNN) is the same as our training label. To do this, we need to minimize the loss function. To do this, it is necessary to find out which weights most directly affect the loss of the network i.e $\frac{dL}{dW}$ where L is our loss function and W are the weights of a specific layer. The next step is the backward pass which determines which weights contribute the most to the loss and finds ways to adjust these weights so that the loss decreases. After the derivative is computed, the last step updates the weights in the opposite direction of the gradient.

$$w = w_i - \eta \frac{dL}{dW} \tag{6.2}$$

$$w = \text{Weight} \tag{6.3}$$

$$w_i = \text{Initial Weight} \tag{6.4}$$

$$\eta = \text{Learning Rate} \tag{6.5}$$

The learning rate is a parameter given to the CNN and it describes the steps the network takes to update the weights. Higher learning rate equals large steps and a lower training time, but a learning rate that is too large can mean the CNN never converges.

Going through backpropagation consists of one training iteration. Once the network completes a specific number of iterations, another parameter given, and runs over all training images that are split up into batches, the process is considered complete. User input parameters, called hyperparameters, help the network converge to optimal weights for each layer. Batch size, learning rate, and training iteration are just some of the user input hyperparameters that help. Lastly, to check if the network has learned, a different set of labeled images are fed to the CNN iteravely through the training process to see how well it's learning. This process is especially important to make sure the network architecture isn't being affected by overfitting (memorizing training input rather than learning).

6.3 Choosing Hyperparameters

Convolutional neural networks are a relatively new tools in computer vision. Choosing hyperparameters for your specific dataset is a non-trivial task. Hyperparameters can range from the amount of layers and filters per layer in an CNN architecture to the stride the receptive field of a filter takes, not to mention training hyperparameters such as learning rate and batch size described above. They're ways to optimize these hyperparameters via hyperparameter optimization using Bayesian Optimization [**?**] but as you can imagine, optimizing an CNN architecture from scratch can be very computationally intensive. For the purpose of this thesis, two well known CNN architectures were used, AlexNet [**?**], which won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012 and therefore bringing awareness of CNNs, and GoogleNet [**?**], which won the ILSVRC in 2014, giving rise to deep networks. Both AlexNet and GoogleNet architectures were used to train on LArTPC images and their low level filter weights. Higher level filter weights were randomly initialized before training so the network can learn high level features of LArTPC image classes. The AlexNet architecture is shown in figure 6.5 and the GoogleNet architecture is shown in figure 6.6

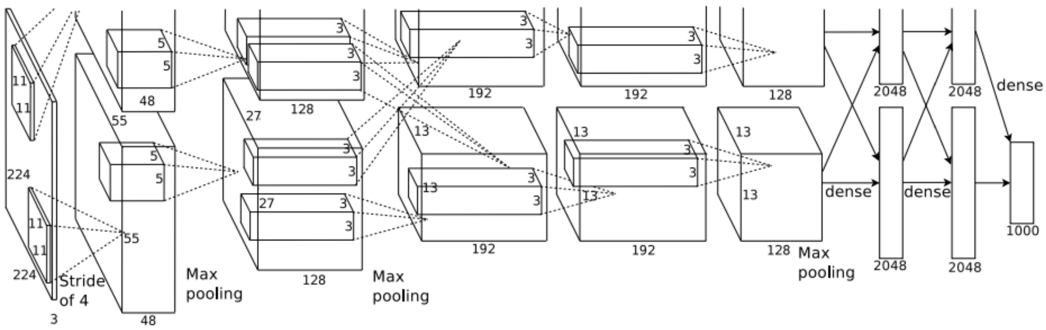


Figure 6.5: Pictoral representation of the AlexNet model. The AlexNet model consists of 5 convolution layers and 3 fully connected layers.

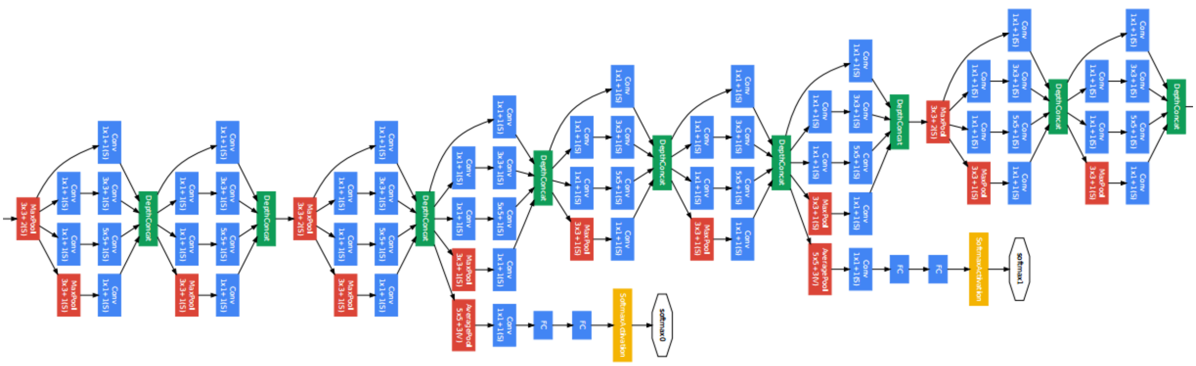


Figure 6.6: Pictoral representation of the GoogleNet model. The GoogleNet model consists of 22 layers. The model implements 9 Inception modules which performs covolution and pooling in parallel and strays away from the basis that CNN layers need to be stacked up sequentially. The GoogleNet model also doesn't use fully connected layers, insteat it uses average pooling which greatly reduces the amount of parameters. GoogleNet has 12x fewer parameters than AlexNet.

Chapter 7

Training process of Convolutional Neural Networks

Three Convolutional Neural Networks (CNNs) were trained throughout this analysis. There are differences to each CNN and will be described fully in the next sections but the main difference are the amount of particle images used for training and validation. CNN1075 used 1,075 muons and 1,075 pions for training and the same amount of each particle for validation. CNN10000 used 10,000 muons and 10,000 pions split in half for testing and training. Lastly CNN100000 had muons, pions, protons, electrons, and gammas in it's training and validation set. Each particle had 20,000 images and training and validation was split 90% training, 10% validation. This chapter will also describe the different hardware frameworks used for training beginning on a CPU and ending on a GPU cluster.

7.1 Hardware Configurations for Convolutional Neural Network Training

The first training iteration, CNN1075, was a proof of concept. This CNN was trained on my local machine for $\sim$4-5 weeks. The batch size had to be very small as well as the image size due to the lack of computation resources. The second iteration of training, CNN10000, was trained on a Fermilab stationed Syracuse University machine. This machine had 6 TB of disk space, 6 cores at 2.1 GHz and 32 GB of RAM. The use of this machine allowed me to increase the training sample as well as the batch size and hence further increase the accuracy of the neural network. Lastly, the CNN100000 was trained using two GTX 1080 Ti GPUs with 11GB of memory on a node on the Syracuse University GPU cluster, SUrge, that has 8 cores and 16GB of memory. This increase in memory as well as the capability to use 2 GPUs drastically cut down on training time from $\sim$4-5 weeks to $\sim$8 hours. SUrge also allowed for hyperparameter optimization by being able to run multiple training iterations over the two GPUs. Lastly, SUrge allowed for training over higher resolution images and a larger particle class of 5 particles vs 2 particles.

7.2 Creating images using LArTPC data for training/validation of CNNS

The μ/π image dataset used to train and validate CNN1075 was created using single generated isotropic muons and pions from 0-2 GeV energy range. 2,150 muons and 2,150 pions were used for training and testing split 50%. The images were created using LArSoft, a liquid argon software, and were based on wire number and time tick in the collection plane. Uboonecode reconstruction version v05_08_00 was used. The raw ADC value after noise filtering was the wire signal. Each collection plane greyscale image was 3456x1600x1 where 6 time ticks were pooled into 1 bin.

After the image was created, the region of interest (ROI) in the image was found by using Open CV, a image processing open source software package, to scan the image starting from the edges and stopping once a bright pixel is encountered. At this step, the ROI can be larger or smaller than the necessary size of a training image and the XY ratio of the image is not kept. This ROI is then resized to an image of 224x224x1.

The greyscale color standard is 8bit therefore the ADC value of wire and time tick was also downsampled due to the 12bit ADC value MicroBooNE has. To do this, the highest ADC pixel in the image was found and then this was divided by the rest placing all pixel values between 0-1. From there, all pixel values are then multiplied by 255.

The μ/π image dataset used to train and validate the CNN10000 was also created using single generated isotropic muons and pions from 0-2 GeV energy range. 10,000 muons and 10,000 pions were used for training and testing split 50%. Uboonecode v06_23_00 was used instead of v05_08_00. Each collection plane greyscale image was 3456x1280x1 where 5 time ticks were pooled into 1 bin which is different than the previous dataset and was implemented due to the fact that the time ticks of an event went from 9400 to 6400 with the change of uboonecode version. Issues that arose in CNN1075 that were fixed in CNN10000 include zero-padding images in X and Y that are smaller than 224X224 to eliminate over-zooming effect and fixing a bug that shifted pixels separated by a dead-wire region.

The $\mu/\pi/p/e/\gamma$ image dataset used to train and validate the CNN100000 were created using single generated isotropic particles with energy range from 0-2 GeV. 20,000 of each particle were used for training and were split 90/10 between training and testing sets. Uboonecode v06_23_00 was used for these images. The collection plane greyscale iamge had the same dimensions as CNN10000, 3456x1280x1 and the ROI algorithm was the same except for resizing these images to 576x576.

A major change other than the higher resolution images was the treatment of the ADC values. In the first two image making schemes, the highest pixel value was found per image and the image was then normalized by that. The issue arising from this ADC normalization wasn't inherent in μ/π training due to the fact that both particles are minimum ionizing particles in liquid argon, however, when dealing with a larger particle class, it was necessary to try and make sure energy deposition by each particle was preserved. The energy deposition in a particle image corresponds to the ADC value or pixel brightness. To preserve energy deposition, the ADC float value was passed straight to the image rather than doing any image normalization. This then makes sure that minimum ionizing particles like muons and pions appear dimmer than highly ionizing particles like protons.

Images were also made from BNB+Cosmic events that passed the cc-inclusive selection 1 filter right before the 75 cm track length cut and were classified using the CNN10000. The dataset used to create these images is the same one used in [**?**], *prodgenie_bnb_nu_cosmic_uboone_mcc7_reco2*. These images were created using information from the track candidate that passed the filter. Only wire number and time ticks associated to the track candidate were drawn on the image to mimic a single particle generated image.

These images were then classified using CNN10000. Two approaches were taken in making these images. The first was using the image normalization above where the maximum pixel in each image is used as a normalization constant to get all pixels between 0-1 then multiply all pixels by 255. As described above, this is the incorrect way to normalize. The second way the images were created was by passing the ADC float to the image. The results of CNN10000 performance are shown in section 7.1.

Lastly, multiple BNB+Cosmic images per event were made for CNN100000 by reducing many of selection I cuts to try and let the CNN do particle as well as event identification. This image making scheme used for CNN100000 will be described in more detail in later sections.

7.3 Convolutional Neural Network Training

7.3.1 Training CNN1075

The results of CNN1075 are described in this section. The accuracy is how well CNN1075 is doing by epoch and was 74.5%. The loss is gradient descent or minimization of the error of the weights and biases used in each neuron of each layer of CNN1075 and was 58% with a trend sloping downwards on the loss curve as well as a trend sloping upward in the accuracy curve. The accuracy and loss of CNN1075 are shown in figure 7.1. Due to the depth of the neural network framework, it was necessary to train with a larger dataset and for more epochs, however, the downward slope of the loss curve is an indication that once trained for longer with a higher training sample, neural networks can be used for μ/π separation. The hyperparameters used to train CNN1075 are detailed below:

- *train_batch_size: 50*
- *test_batch_size: 50*
- *test_iter: 50*
- *test_interval: 50*
- *base_lr: 0.01*
- *lr_policy: "step"*
- *gamma: 0.1*
- *stepsize: 200*
- *display: 50*
- *max_iter: 5000*
- *momentum: 0.9*
- *weight_decay: 0.0005*
- *snapshot: 100*

The confusion matrices shown in figure 7.2 show the accuracy for both the training and testing datasets. The fact that these two have similar accuracies is important because if the training dataset had a much higher accuracy, that indicates an overtraining of the training sample which means the neural network didn't learn features to separate muons from pions, it just memorized what was in the training dataset.

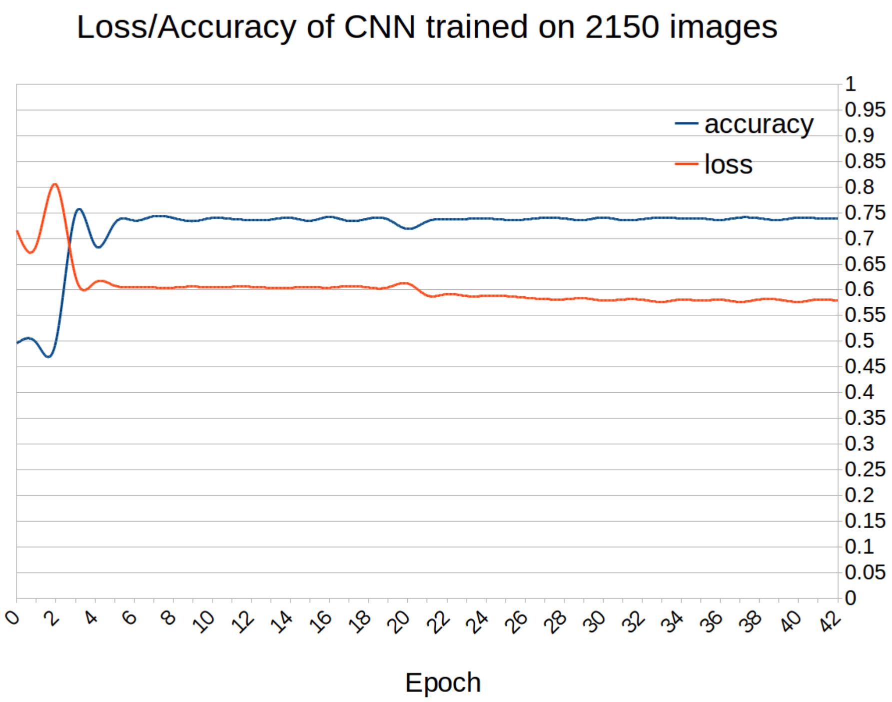
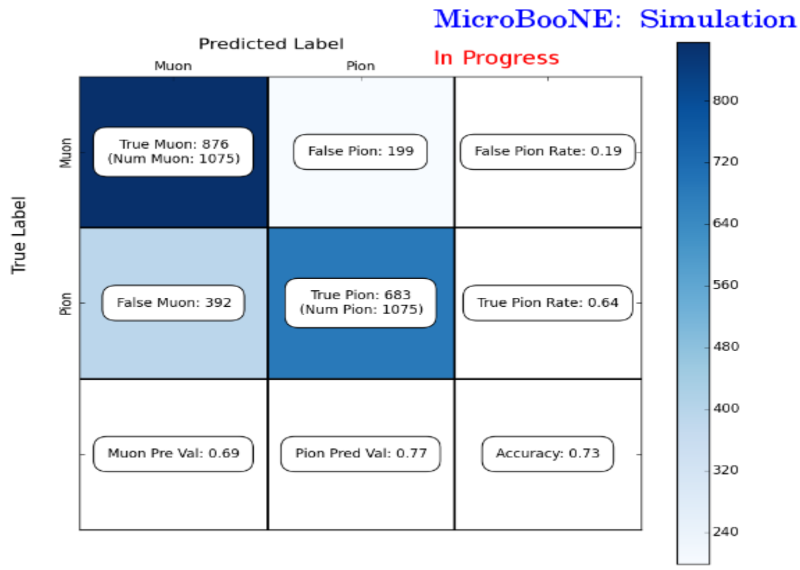
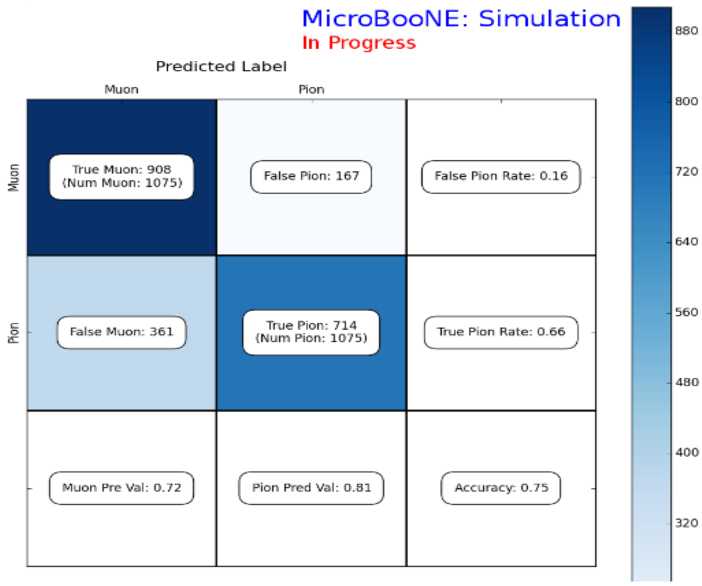


Figure 7.1: Accuracy vs. Loss of AlexNet 2-output μ/π sample consisting of 2,150 images each.



(a) Confusion Matrix showing Accuracy of CNN1075 using training MC data



(b) Confusion Matrix showing Accuracy of CNN1075 using testing MC data

Figure 7.2: Description of confusion matrix variables: False pion rate = $false\pi/total\pi$ True pion rate = $true\pi/total\pi$ Accuracy = $(true\pi rate + true\mu rate)/2$ Pion prediction value = $true\pi/(true\pi + false\pi)$ Muon prediction value = $true\mu/(true\mu + false\mu)$

Also note that the neural network does a better job of identifying muons than pions. This can be attributed to the more complex event scenes pions tend to leave in the detector due to pion interacting more in LAr than muons do. The CNN may do better at identifying pions with a larger training sample.

7.3.2 Training CNN10000

The hyperparameters used for CNN10000 are shown below. The batch size for the training and testing as well as the test_iter were chosen to encompass the whole training/testing image set when doing accuracy/loss calculations. To do this, multiplying the test_iter by the test batch size gives you the amount of images used when calculating accuracy/loss curves.

- *train_batch_size: 100*
- *test_batch_size: 100*
- *test_iter: 100*
- *test_interval: 100*
- *base_lr: 0.001*
- *lr_policy: "step"*
- *gamma: 0.1*
- *stepsize: 1000*
- *display: 100*
- *max_iter: 10000*
- *momentum: 0.99*
- *weight_decay: 0.0005*
- *snapshot: 100*

The same architecture that was used to train CNN1075 was employed on CNN10000, AlexNet. Caffe [?] was the software package used for both CNNs. The differences include batch size and test_iter and momentum to account for the larger dataset. Figure 7.3 shows the loss and accuracy of CNN10000. There is around a 10% increase in accuracy from CNN1075 to CNN10000, 85%, and around a 20% decrease in loss, 36%.

Figure 7.4 show a breakdown of μ/π separation for CNN10000. It also shows the network is not being over-trained due to the Accuracy of both the training and testing datasets being within .01% of each-other. Figure 7.5 shows how well the neural network is doing at μ/π separation with respect to muon probability. The red bins corresponds to true pions and the blue bins correspond to true muons. There is still pion contamination in the high muon probability bins but by choosing a muon probability of $\geq 80\%$ we can reduce this. The CNNs increase in total accuracy can be attributed to an increase in accurately classifying pions as pions as seen in both the confusion matrix in figure 7.4 and the large number of events in the zero bin of the muon probability plot seen in figure 7.5 that corresponds to high probability pions.

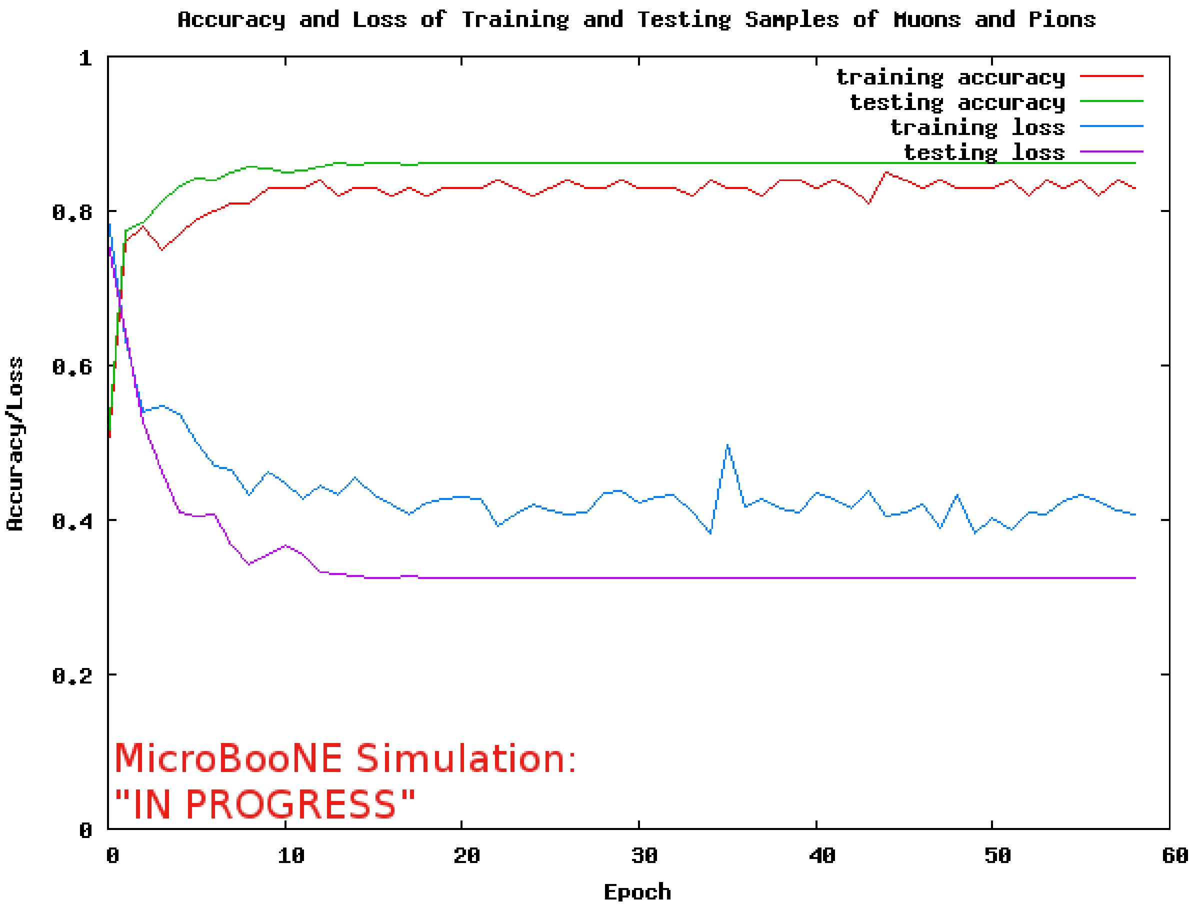
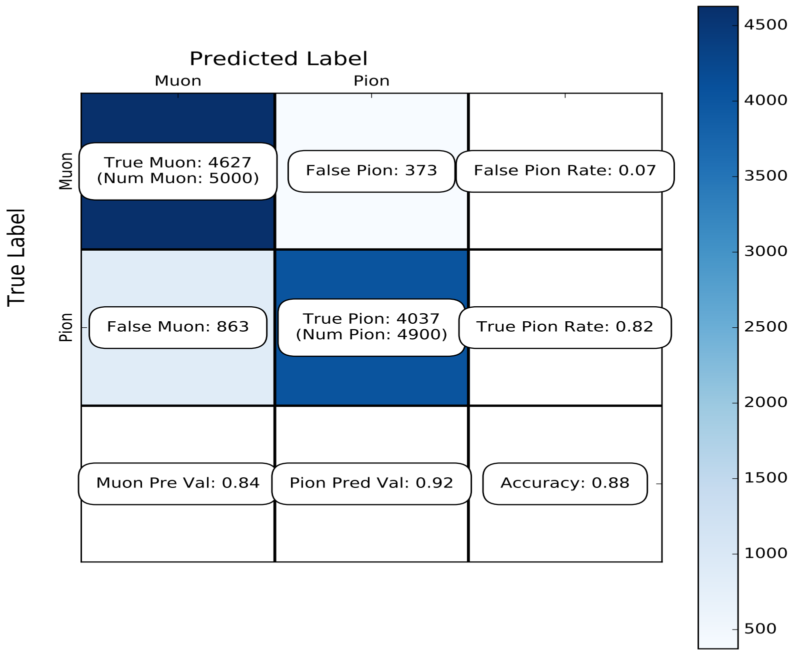
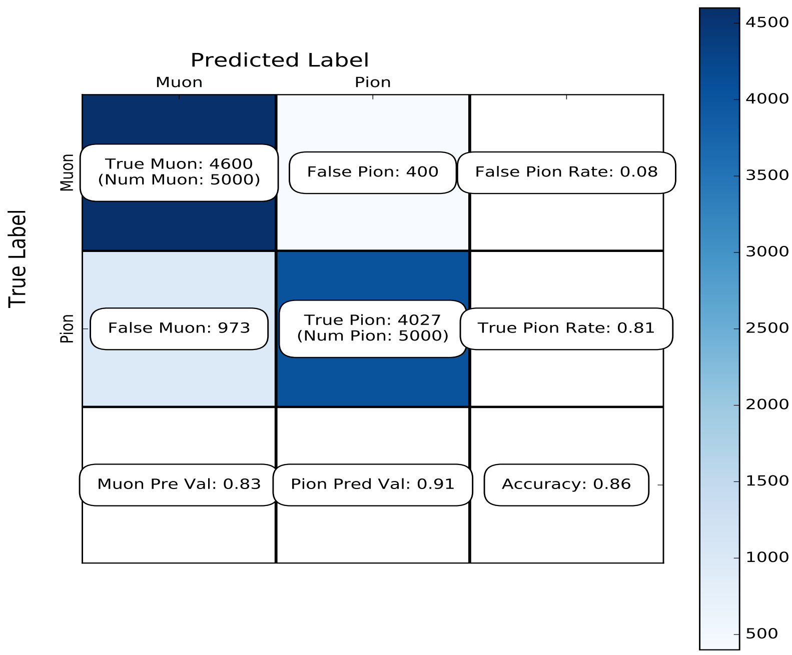


Figure 7.3: Accuracy vs. Loss of AlexNet 2-output μ/π sample consisting of 10,000 images each.



(a) Confusion Matrix showing Accuracy of CNN10000 using training MC data



(b) Confusion Matrix showing Accuracy of CNN10000 using testing MC data

Figure 7.4: Description of confusion matrix variables: False pion rate = $false\pi / total\pi$; True pion rate = $true\pi / total\pi$; Accuracy = $(true\pi rate + true\mu rate)/2$; Pion prediction value = $true\pi / (true\pi + false\pi)$; Muon prediction value = $true\mu / (true\mu + false\mu)$

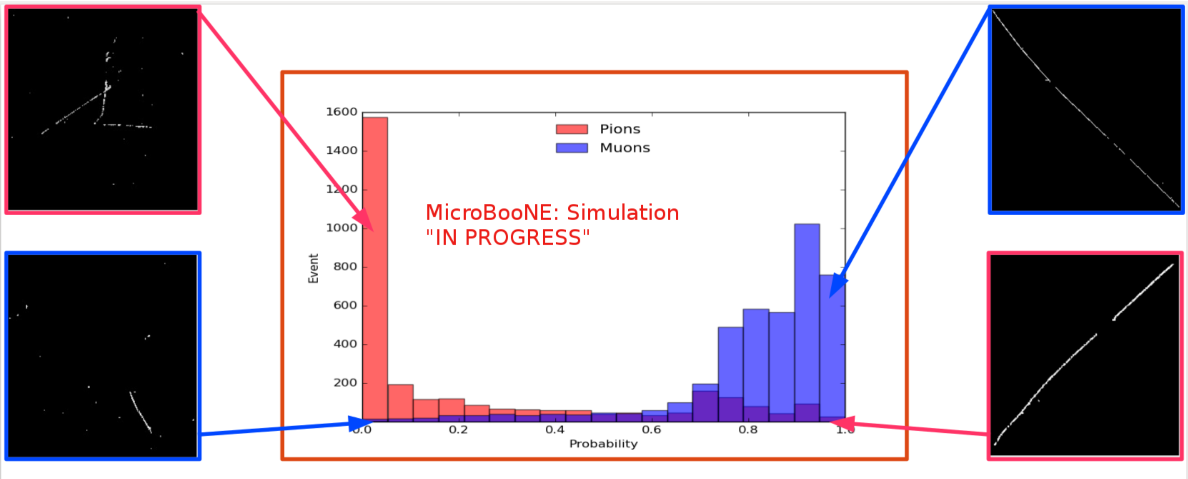


Figure 7.5: Probability plot of muons and pions from testing set. Images surrounding histogram are a random event from lowest bin and highest bin for each particle.

7.3.3 Training CNN100000

CNN100000 used the GoogleNet architecture rather than the AlexNet architecture used in the two previous trained CNNs. This is the first time the neural network was trained on a larger particle class, $\mu/\pi/p/\gamma/e$, and on higher resolution images. This CNN also employed GPUs during the training process. The hyperparameters are shown below:

- *train_batch_size: 18*
- *test_batch_size: 2*
- *test_iter: 2000*
- *test_interval: 2000*
- *base_lr: 0.001*
- *lr_policy: "step"*
- *gamma: 0.96*
- *stepsize: 10000*
- *average_loss: 40*
- *display: 40*
- *max_iter: 10000*
- *momentum: 0.99*
- *weight_decay: 0.0002*
- *snapshot: 50000*

The accuracy and loss for CNN100000 are shown in figures 7.6 and 7.7. The jumps shown in both figures are when the training was stopped to fine-tune the weight decay and the learning rate. The accuracy leveled off at $\sim$80% and the loss was at $\sim$0.48.

Figure 7.8 shows the confusion matrix of CNN100000. The proton identification of the neural network is at 85% and the highest out of all five particles. One thing to note is clear separation between particles that leave track like objects in the MicroBooNE detector, $\mu/\pi/p$, versus particles that leave shower like objects in MicroBooNE, e/γ.

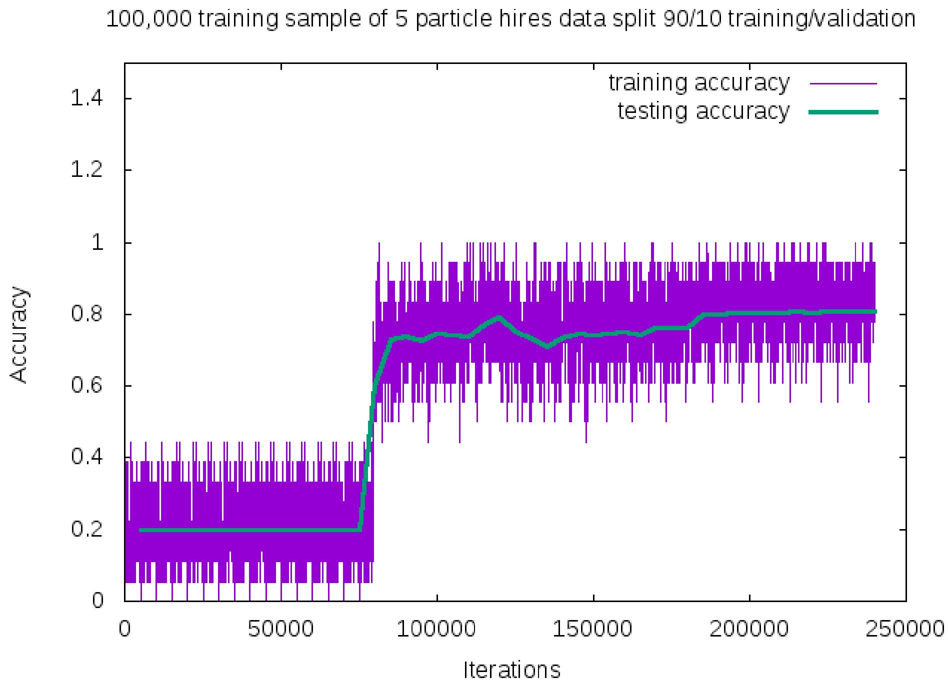


Figure 7.6: Training and testing accuracy of CNN trained on 100,000 images of $\mu/\pi/p/\gamma/e$ with 20,000 images of each particle. Each image was a size of 576x576 and the images per particle were split 90% use for training and 10% used for testing the network

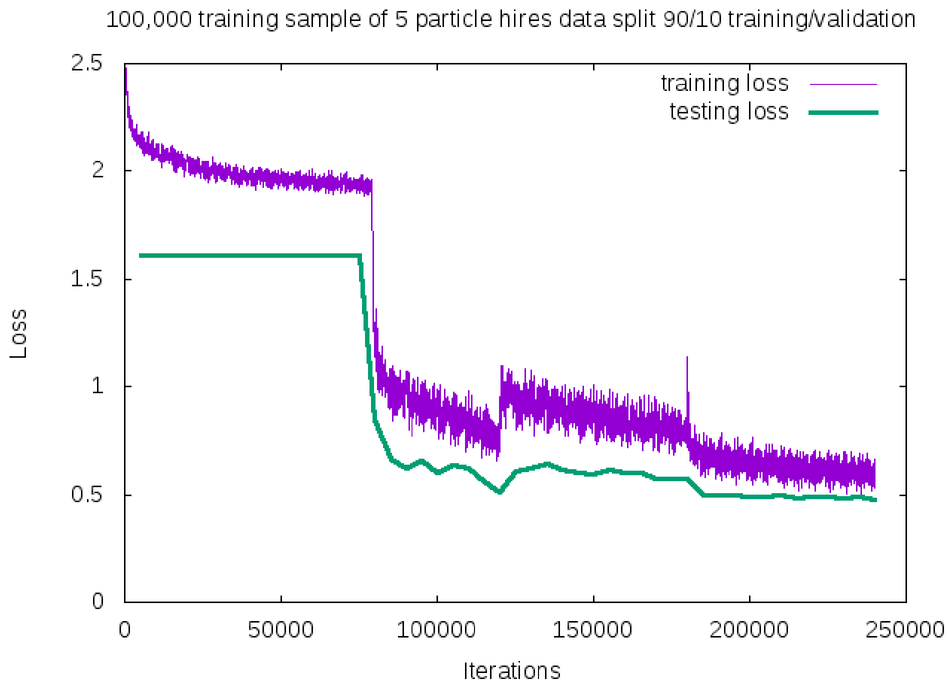


Figure 7.7: Training and testing loss of CNN trained on 100,000 images of $\mu/\pi/p/\gamma/e$

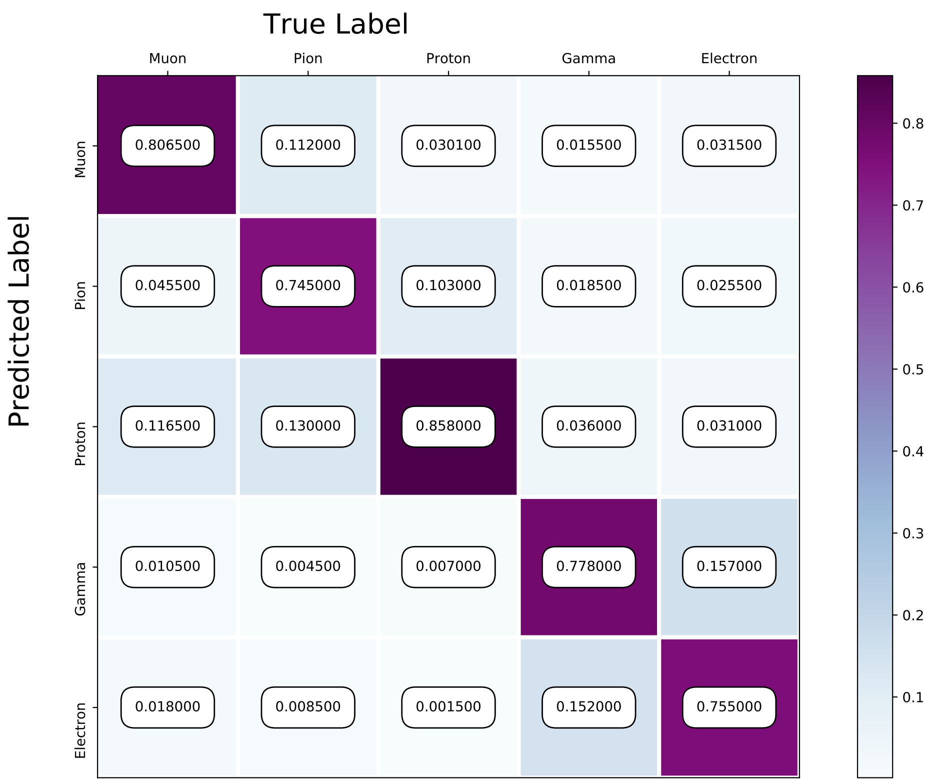


Figure 7.8: Confusion Matrix of all five particles

Another visualization of how the neural network is learning is shown in 7.9. t-SNEs [?] is a technique used for dimensionality reduction developed for use in visualizing high-dimensional datasets. Each datapoint is given a location in a two or three-dimensional map by using stochastic neighbor embedding to convert high-dimensional euclidean distances between datapoints into conditional probabilities that represent the similarities between these datapoints. For datapoints close together on the map, their conditional probabilities are high, for datapoints with a wide separation between them, their conditional probabilities are very small. Figure 7.9 is a t-SNE of the final training iteration of a subset of the training sample used in CNN100000. You can see a clear separation between track like objects and shower like objects. You can also see that electrons and gammas are not as separated as muons, pions, and protons. For the purpose of this thesis, this isn't an issue but later iterations of training could include more images for the gamma and electron classes to help the CNN further separate these classes.

Figure 7.10 shows the probability of each particle class and the highest probability misidentification for each class. For muons, the largest misidentification is from

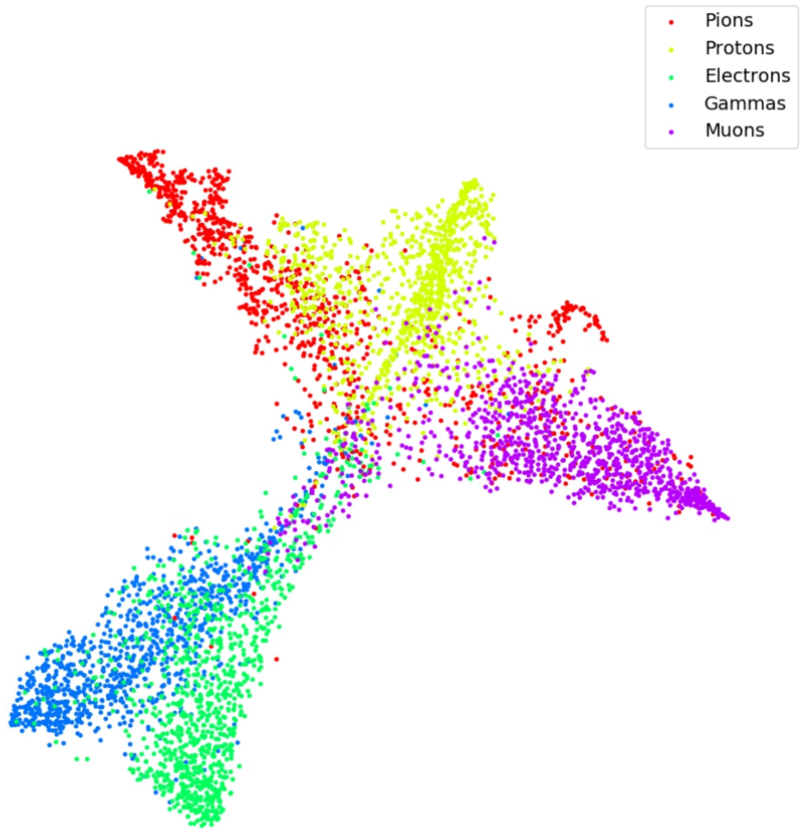
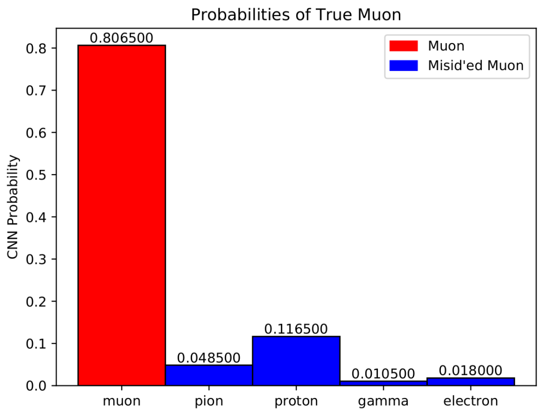
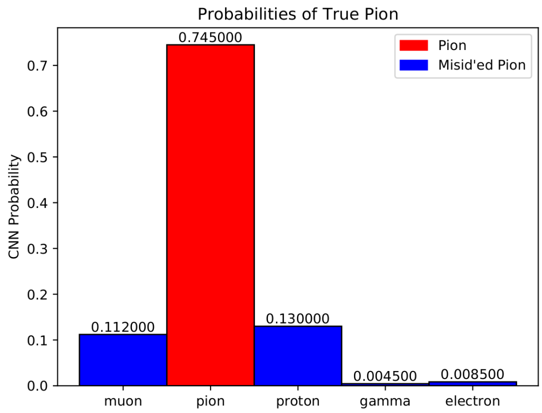


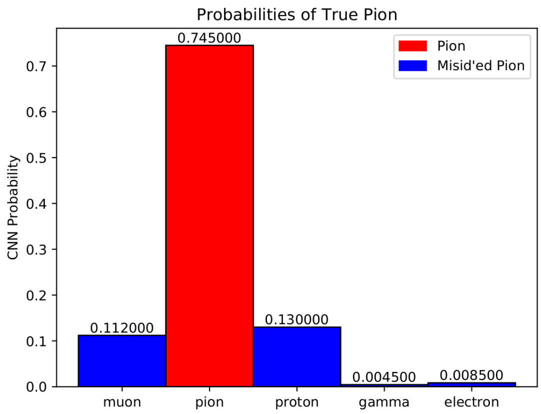
Figure 7.9: t-SNE of CNN



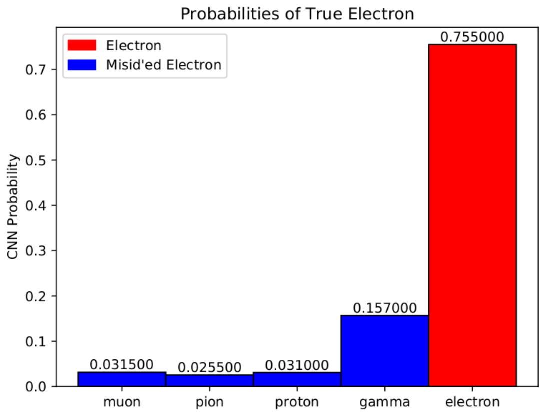
(a) Muon Prob



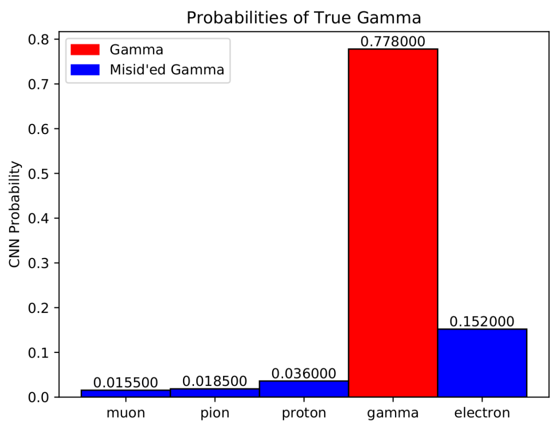
(b) Pion Prob



(c) Proton Prob



(d) Electron Prob



(e) Gamma Prob

Figure 7.10: Probabilities of different particle classes as well as their contamination from other classes

protons. For pions, both protons and muons get misidentified as pions at around the same probability. Similar behavior is also seen for proton identification. Electrons and gammas are misidentified as each-other with similar probabilities.

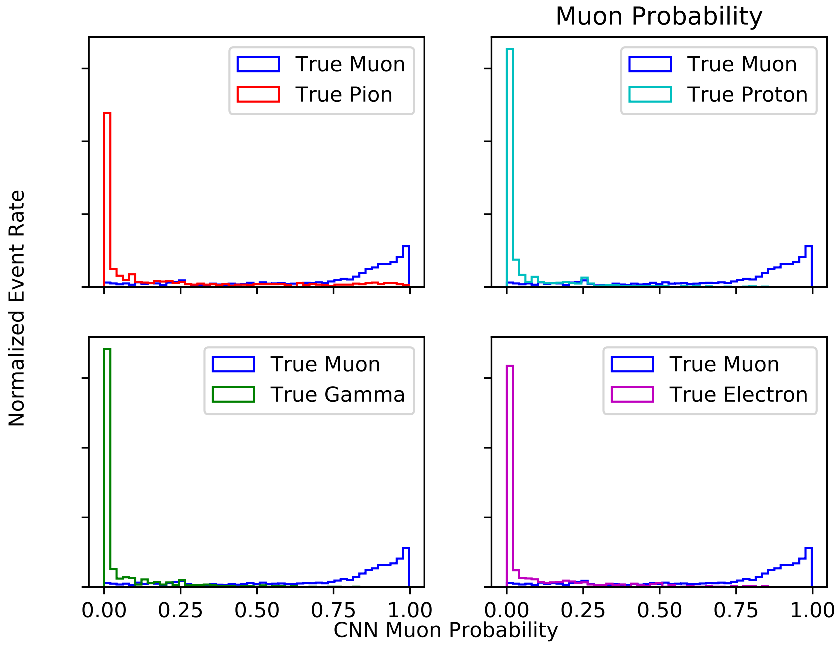
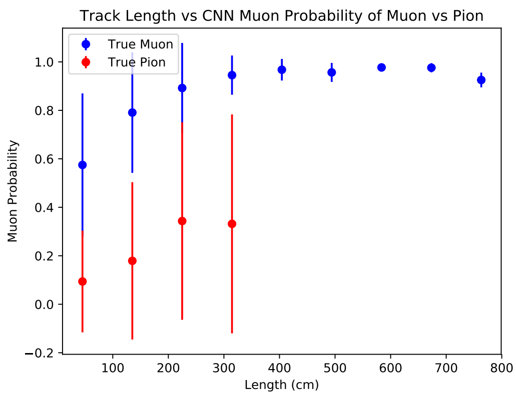


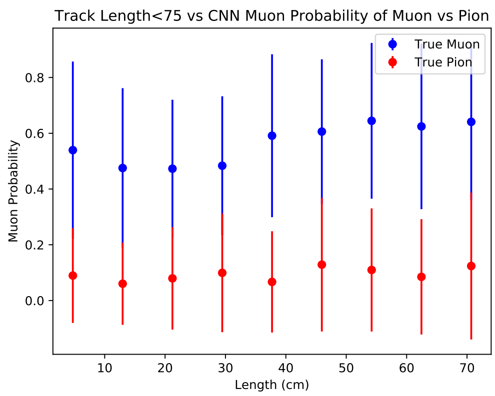
Figure 7.11: Muon probability of true muons (blue) versus pions (red), protons (cyan), gammas (green) and electrons (magenta).

To see what type of background contamination one would be dealing with when doing muon identification, muon probabilities for each particle class was plotted against the probability of true muons to see how well muon signal vs other particle background separation can be done with CNN100000. Figure 7.11 is showing the true muon probability for true muons, versus the rest of the particle classes. This plot describes which muon probability value should be chosen for the least amount of other particle contamination. For electrons and gammas, a muon probability of $\sim 75\%$ would eliminate e/γ contamination. For pions and protons, there is contamination at all values of muon probability, but the contamination is drastically reduced at a muon probability $\geq 75\%$.

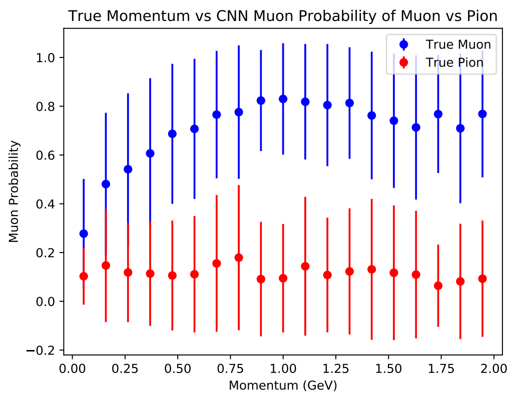
One of the main concerns with training a neural network was that the features the network would learn to separate muons from pions would be track range, which is what was used to begin with in selection I. To make sure that wasn't the case, the next thing that was looked at was the muon probability versus track range and momentum of the track. Figures 7.12 through 7.15 show the muon probability in blue for all plots against all other particles. A zoomed in version of track range for all particles was also plotted to make sure there is separation between the particles at low track range. The μ/π separation in track range and momentum is less than for $p/e/\gamma$ but that was to be expected. Although the separation isn't as good as the other particles, there still is separation at low momentum and low track range which cannot be done by using a track range cut like selection I does.



(a) Track range versus muon probability for true muons (blue) and true pions (red).

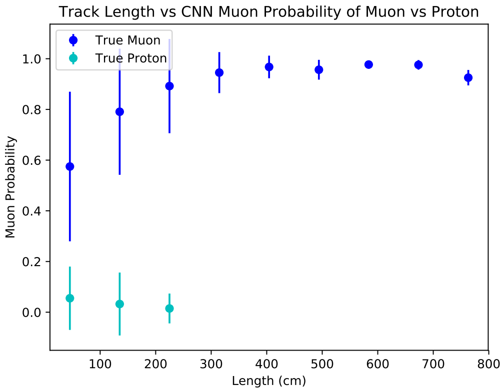


(b) Track range ≤ 75 cm versus muon probability for true muons (blue) and true pions (red).

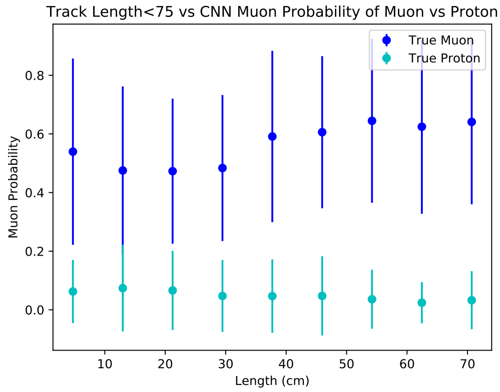


(c) Momentum versus muon probability for true muons (blue) and true pions (red).

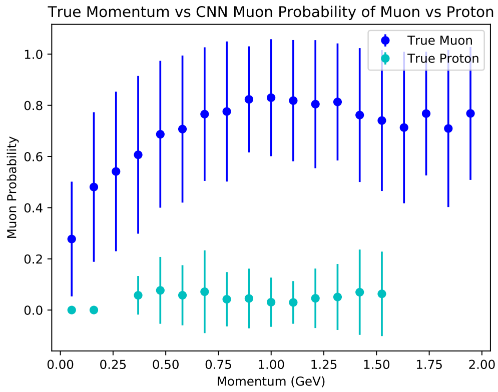
Figure 7.12: Kinematic distributions versus muon probability for true muons and true pions.



(a) Track range versus muon probability for true muons (blue) and true protons (cyan).

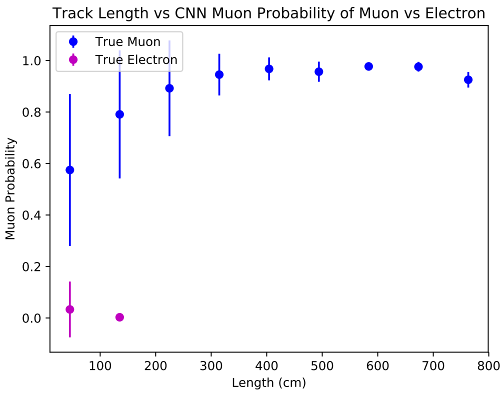


(b) Track range $\leq$ 75 cm versus muon probability for true muons (blue) and true protons (cyan).

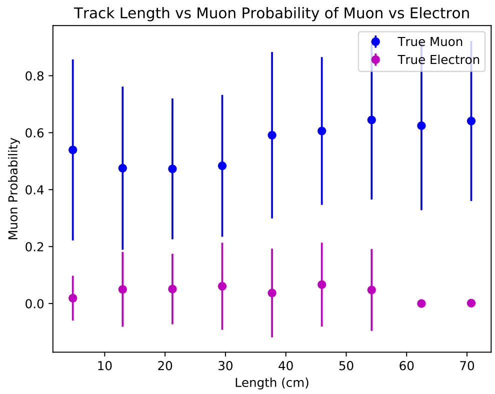


(c) Momentum versus muon probability for true muons (blue) and true protons (cyan).

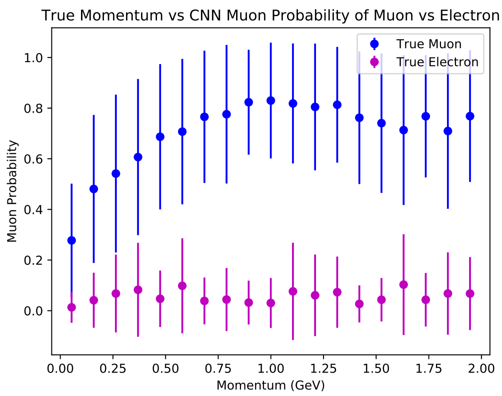
Figure 7.13: Kinematic distributions versus muon probability for true muons and true protons.



(a) Track range versus muon probability for true muons (blue) and true electrons (magenta).

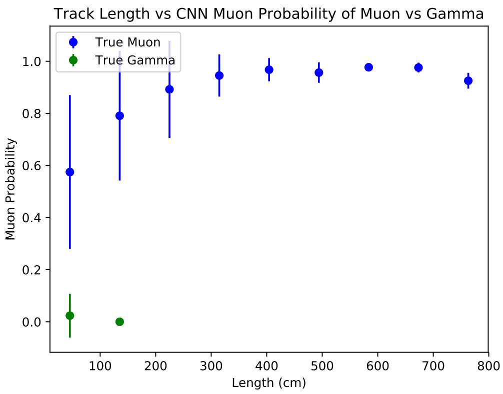


(b) Track range ≤ 75 cm versus muon probability for true muons (blue) and true electrons (magenta).

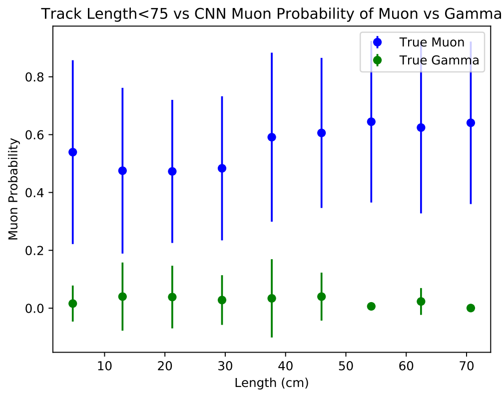


(c) Momentum versus muon probability for true muons (blue) and true electrons (magenta).

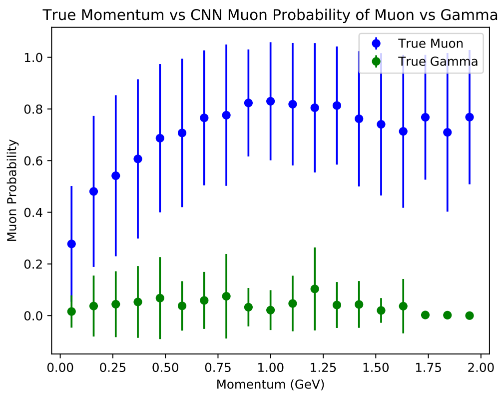
Figure 7.14: Kinematic distributions versus muon probability for true muons and true electrons.



(a) Track range versus muon probability for true muons (blue) and true gammas (green).



(b) Track range ≤ 75 cm versus muon probability for true muons (blue) and true gammas (green).



(c) Momentum versus muon probability for true muons (blue) and true gammas (green).

Figure 7.15: Kinematic distributions versus muon probability for true muons and true gammas.

Chapter 8

Using Convolutional Neural Networks for ν_μ CC event classification

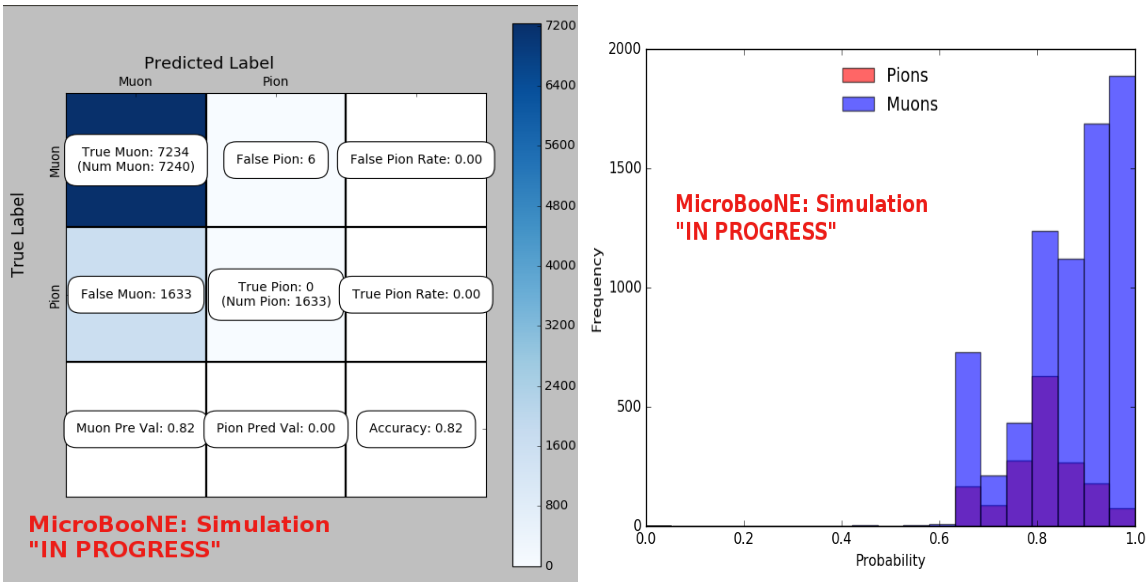
8.1 Classification using CNN10000

8.1.1 Classification of MC data using Selection I CC-Inclusive Filter

CNN10000 was also used to classify track candidate images that were identified by the selection I cc-inclusive filter described in [**?**]. Passing rates for each cut in this filter are shown in figure **??**. As seen in section **??**, wrong image normalization had a higher muon classification probability so all work done using selection I cc-inclusive filter was done using this normalization. Out of 188,880 events, 19,112 passed the cut right before the 75 cm track length cut which is a 10.1% passing rate and comparable to the 10% passing rate shown in figure **??**. In time cosmics were also run over, out of 14,606 in time cosmics events, 302 passed the cut right before the 75 cm track length cut which is a 2.1% passing rate comparable to the 2.7% passing rate in the cc-inclusive tech-note. Figures 8.1a and 8.1b show the accuracy and μ/π separation. Both plots are only composed of muons and pions and like selection I original data, all other particles were id'ed as muons. Also like selection I original data, muons are being identified at a very high rate. Figure 8.2a shows the track range distributions of all events from selection I being classified by the CNN as a muon with a probability of 70% regardless of true particle type. We get entries for the CNN curve in the lowest bin and none for the 75 cm curve. To see how many true CC events were identified by CNN10000 breaking down figure 8.2a by event type was necessary. Figures 8.2b and 8.2c show track range distributions separated by signal and various backgrounds.

Particle type was not taken into consideration in these plots so true CC event images can be any track candidate particle passing selection I cut right before track length cut including pions and protons.

To gain an even deeper understanding on how CNN10000 is performing, plotting these distributions with only muons and pions was done due to the fact that CNN10000 was trained with only those particles for μ/π separation. Figures 8.2d-8.3d show the stacked histograms of signal and background of the track range distributions with varying CNN probabilities starting from 70% and ending at 90% probability. With higher probabilities we get a purer sample in the lower bin but we end up losing events as well. Momentum distributions for all signal/background events are shown in figure 8.4.

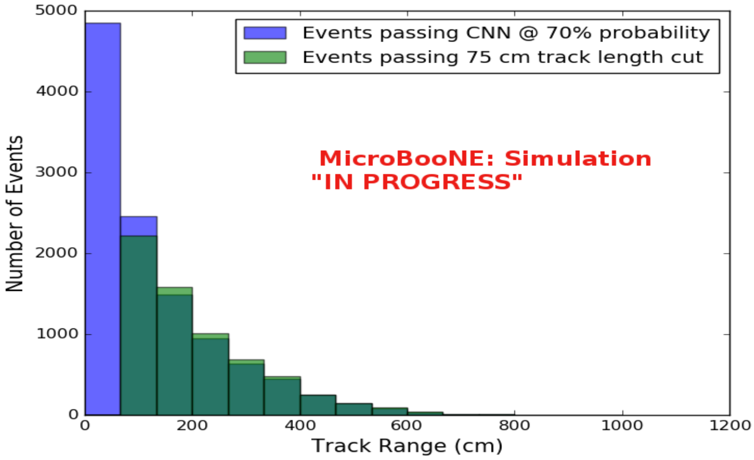


(a) Confusion Matrix for CNN10000 classified events from selection I

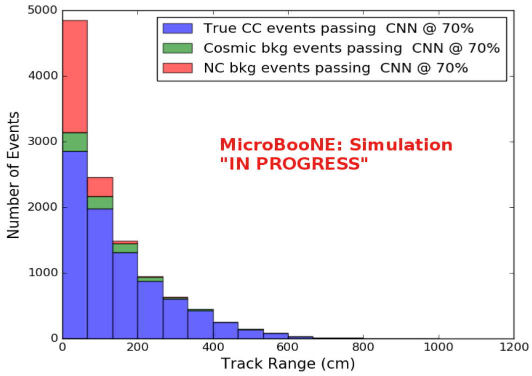
(b) Probability plot for CNN10000 classified events from selection I

Figure 8.1: Confusion matrix and probability plot of events passing selection I cc-inclusive cuts right before 75cm track length cut

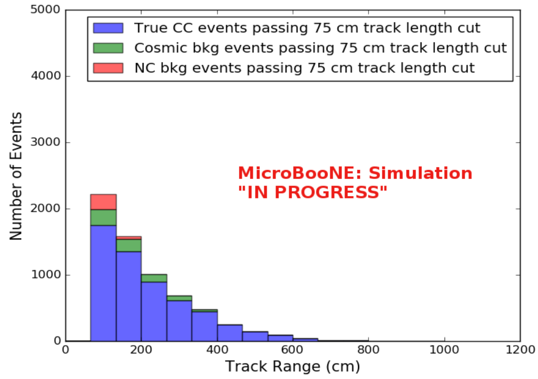
Another check was to see if any true CC pions were passing through the cut right before the 75 cm track length cut. Figure 8.5 shows the comparison of the stacked track range distribution with only true CC muon signal versus the stacked distribution with true CC muons and pions signal. As you can see, we gain more events when plotting CC events with a particle type of either muons or pions due to the CNN classifying all pions in this dataset as muons. This is an interesting scenario and a sample of topologies of these images are represented in figure 8.6, at least 3 tracks are coming out



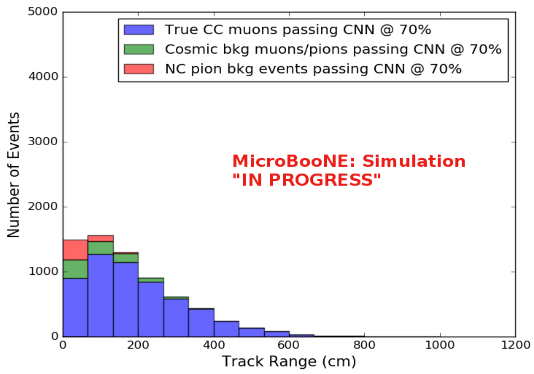
(a) Track range distribution of events from Selection I Modified passing CNN with 70% accuracy



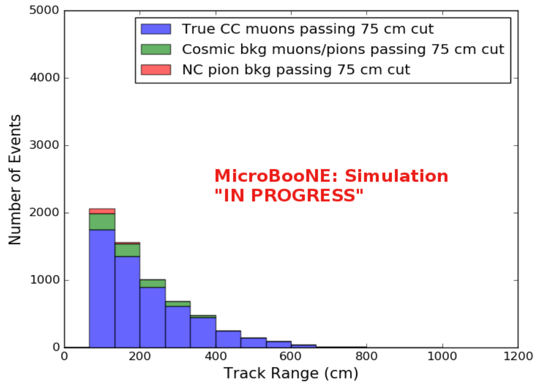
(b) Stacked signal and background track range distributions from Selection I Modified passing CNN with 70% accuracy



(c) Stacked signal and background track range distributions from Selection I Modified passing 75 cm track length cut

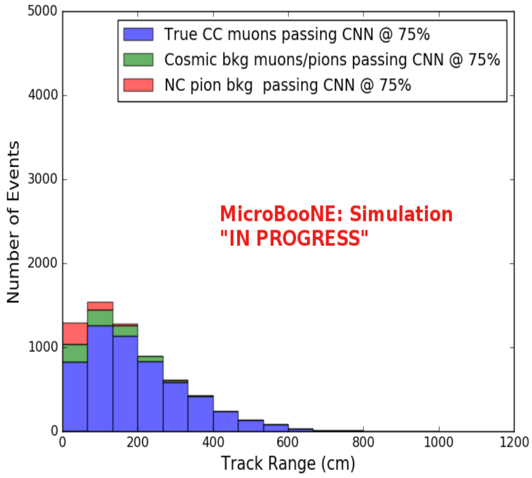


(d) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing CNN with 70% accuracy

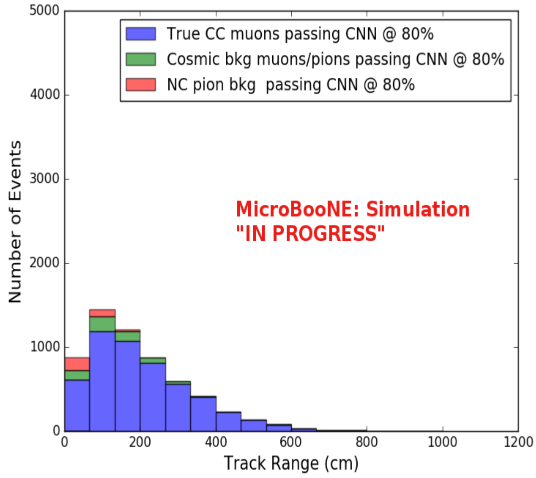


(e) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing 75 cm track length cut

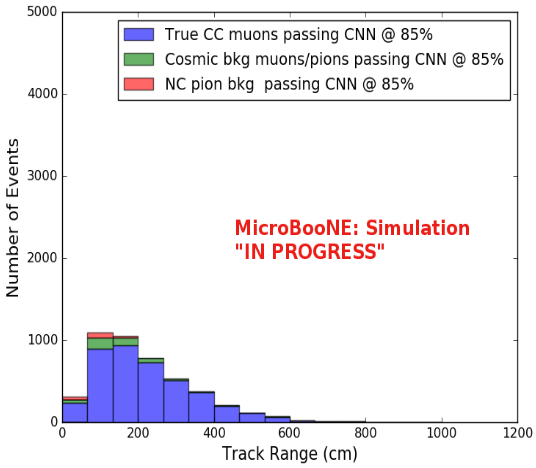
Figure 8.2: CNN10000 distributions of track candidate images output from Selection I Modified cc-inclusive filter



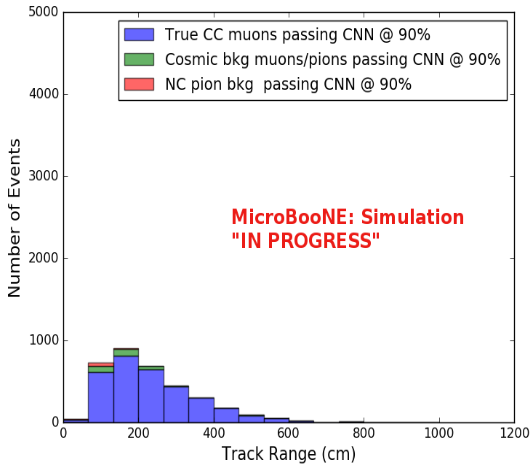
(a) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing CNN with 75% accuracy



(b) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing CNN with 80% accuracy

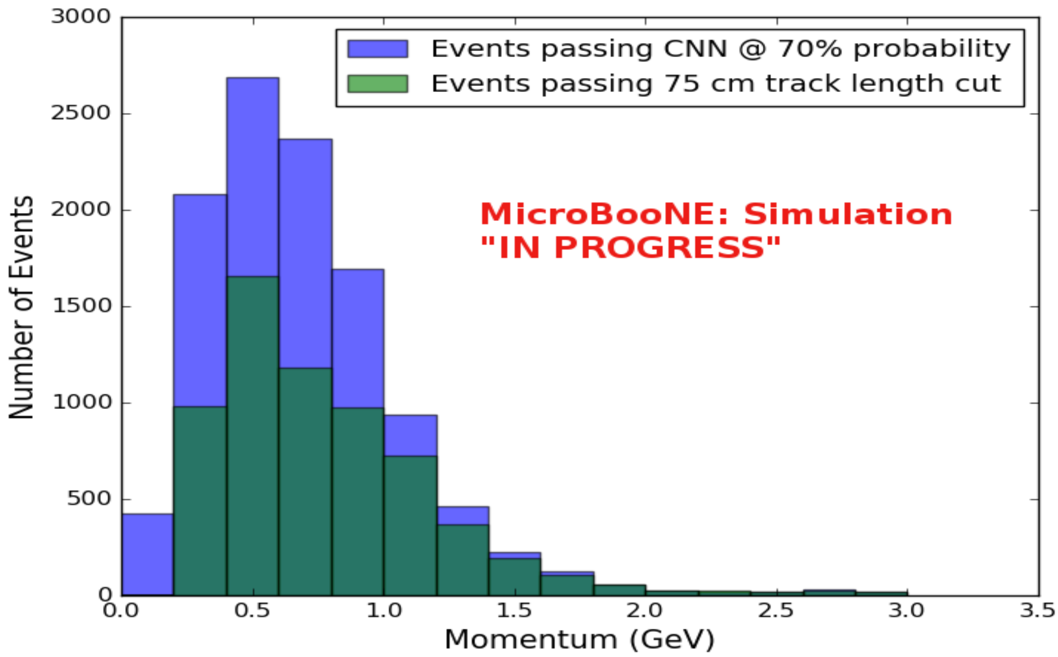


(c) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing CNN with 85% accuracy

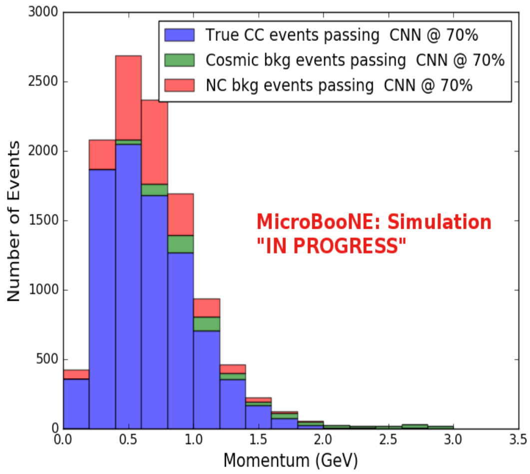


(d) Stacked signal muons and background muons/pions of track range distributions from Selection I Modified passing CNN with 90% accuracy

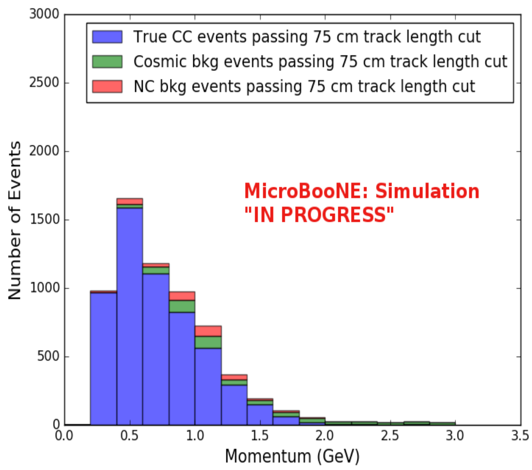
Figure 8.3: CNN10000 stacked signal/background track range distributions of track candidate images output from Selection I Modified cc-inclusive filter



(a) Momentum distribution of events from Selection I Modified passing CNN with 70% accuracy



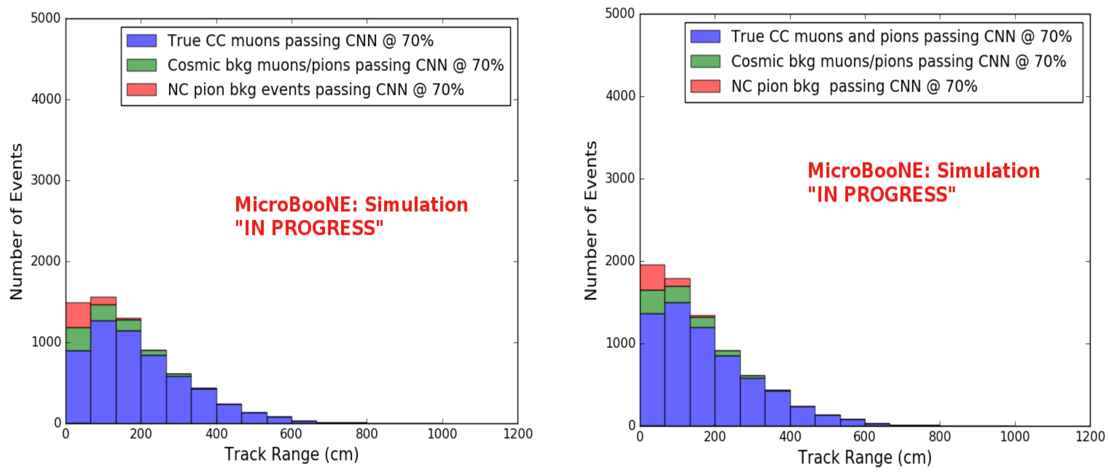
(b) Stacked signal and background momentum distributions from Selection I Modified passing CNN with 70% accuracy



(c) Stacked signal and background momentum distributions from Selection I Modified passing 75 cm track length cut

Figure 8.4: CNN10000 momentum distributions of track candidate images output from Selection I Modified cc-inclusive filter

of the vertex for these types of events. With the 75 cm track length cut, the selection is cutting event topologies like this where the pion is the tagged track candidate. Figure 8.6a has a defined longer muon track, but because of dead wires through the track, the reconstructed range is 1. less than 75 cm and 2. shorter than the reconstructed pion whose length is also less than 75 cm. This is a very interesting event, but because of issues with the tracking algorithm, the 75 cm cut would get rid of this event. The CNN was able to recover this event only because it has classified all pions as muons. Figure 8.6b shows the second case to think about, the pion, while still less than 75 cm has a reconstructed track length longer than the muon. Again, the CNN recovered this event due to pions being classified as muons. Lastly, figure 8.6c shows a pion with a reconstructed track length greater than 75 cm and the muon. These three cases show that a broader question must be asked when training the network other than is it a muon or pion. There are different routes to recover interesting events like these. One route is to ask the network "Is it a CC event or is it an NC event?" and obtain an image dataset consisting of whole CC/NC events that will train the network to answer this question. The other route is to ask the network "Is this a $\mu/\pi/p/$ from a CC event or NC event and obtain an image dataset consisting of primary particles from a CC/NC event. Both these paths will be explored in future work.

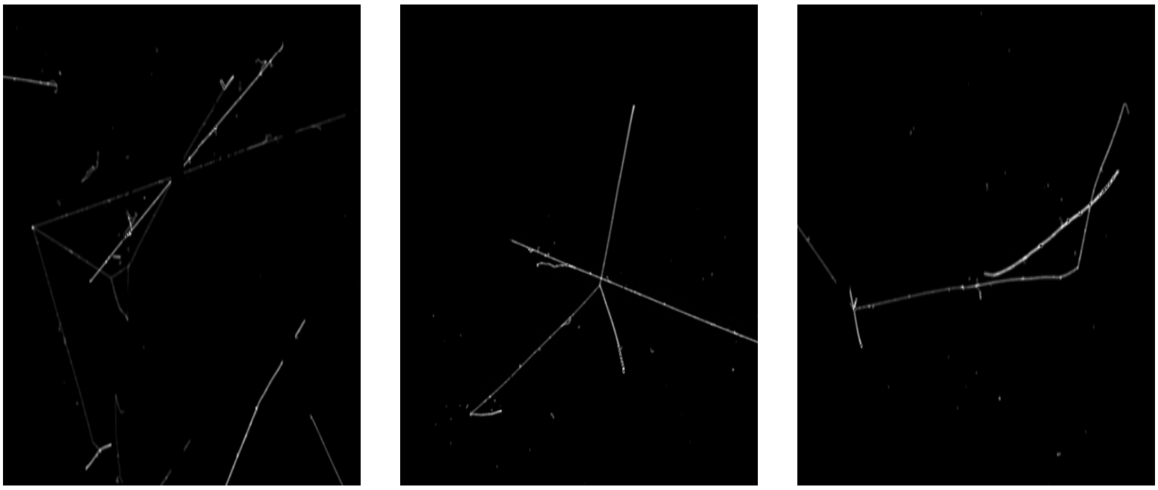


(a) Stacked signal μ/backround μ and π track range distribution of CNN @ 70%

(b) Stacked signal $\mu\&\pi$/backround $\mu\&\pi$ track range distribution of CNN @ 70%

Figure 8.5: Track distribution comparisons of true CC muons plotted vs true CC muons and pions plotted

Table 8.1 shows the passing rates for the 75 cm track length cut and the CNN cut at 70% and 83%. The passing rates at the track containment level for the 75 cm track length cut compared to the CNN are comparable with only a 0.6% difference in the



(a) Pion recontructed track range is less than 75 cm and longer than muon track due to dead wires

(b) Pion recontructed track range is less than 75 cm and larger than muon reconstructed track

(c) Pion reconstructed track range is greater than 75 cm and larger than muon reconstructed track

Figure 8.6: Images of true CC events where the pion was the tagged track candidate

		BNB + Cosmics		Cosmic Only	Signal:
		Selection	MC Truth		Cosmic Only
75 cm Cut passing rates	Generated Events	191362	45723	4804	1:22
	Track Containment	19391 (48%/10%)	11693 (45%/26%)	129 (38%/2.7%)	1:2.3
	track $\geq$ 75 cm	6920 (36%/3.6%)	5780 (49%/13%)	17 (13%/0.4%)	1:0.6
CNN passing rates	Generated Events	188880	44689	14606	1:21
	Track Containment	19112 (/10%)	11554 (/26%)	302 (/2.1%)	1:1.73
	CNN cut @ 70% Probability	16502 (86%/8.7%)	10605 (92%/23%)	205 (68%/14%)	1:1.28
	CNN cut @ 83% Probability	7511 (46%/4.0%)	6142 (58%/14%)	32 (16%/0.2%)	1:0.4

Table 8.1: Comparing passing rates of CNN at different probabilities versus 75 cm track length cut: Numbers are absolute event counts and Cosmic background is not scaled appropriately. The BNB+Cosmic sample contains all events. The numbers in brackets give the passing rate wrt the step before (first percentage) and wrt the generated events (second percentage). In the BNB+Cosmic MC Truth column shows how many true ν_μ CC-inclusive events (in FV) are left in the sample. This number includes possible mis-identifications where a cosmic track is picked by the selection instead of the neutrino interaction in the same event.The CNN MC True generated events were scaled wrt the MC True generated events for the 75 cm cut passing rates due to only running over 188,880 generated events versus the 191362 generated events. The last column Signal:Cosmic only gives an estimate of the ν_μ CC events wrt the cosmic only background at each step. For this number, the cosmic background has been scaled as described in [**?**]. Note that these numbers are not a purity, since other backgrounds canâĂŹt be determined at this step.

		#Events(Fraction) passing Sel I	#Events(Fraction) passing CNN @ 83% Probability
Signal	ν_μ CC events with true vertex in FV	1168(53.8%)	6142(61%)
Backgrounds	Cosmics Only Events	725(33.4%)	2582(26%)
	Cosmics in BNB Events	144(6.6%)	492(4.9%)
	NC Events	75(3.5%)	778(7.7%)
	ν_e and $\bar{\nu}_e$ Events	4(0.2%)	32(0.3%)
	$\bar{\nu}_\mu$ Events	40(1.8%)	67(0.7%)

Table 8.2: Signal and background event numbers of selection I and selection I with CNN cut estimated from a BNB+Cosmic sample and Cosmic only sample normalized to $5 * 10^{19}$ PoT. The last column gives the fraction of this signal or background type to the total selected events per CNN probability.

in time cosmic bin which may be due in part to the larger in time cosmic statistics used for the CNN dataset. These passing rates need to be comparable to then be able to compare the passing rates after the CNN cut to the 75 cm cut. Again, the same BNB+Cosmic sample was used for both selection I with 75 cm cut and selection I with CNN cut. As it stands, a CNN cut at 83% probability has a MC true CC event passing rate of 14% compared to the 13% passing rate of the 75 cm track length cut. The Signal:Cosmic Only background is also reduced from 1:0.6 to 1:0.4 The total passing rate is also higher than the 75 cm cut, 3.6% vs 4.0%. Table 8.2 shows the breakdown of signal and backgrounds for the CNN at the different probabilities. We have a 61% signal passing rate with the CNN cut @ 83% versus the 53.8% signal passing rate of the 75 cm cut.

Based on these numbers, the following performance values of the selection with 75 cm cut versus selection with CNN @ 83% probability cut were calculated:

- Efficiency: Number of selected true ν_μ CC events divided by the number of expected true ν_μ CC events with interaction in the FV.
  - Selection I: 12.3%
  - Selection I with CNN10000 cut @ 83% probability: 14%
- Purity: Number of selected true ν_μ CC events divided by sum of itself and the number of all backgrounds.
  - Selection I: 53.8%
  - Selection I with CNN10000 cut @ 83% probability: 61%

Lastly, figure 8.7 shows a more representative performance of the CNN. Due to the fact that the CNN was trained on muons and pions, showing the performance of CC muon events versus NC pion events with respect to CNN probability gives a better picture of how the network is performing. Figure 8.7 shows that at 83% we are below the 75 cm cut NC pion threshold and still above the CC muon threshold. Using 83% probability not only reduced the NC pion background, it also dramatically reduced the in time cosmics and cosmics in the BNB.

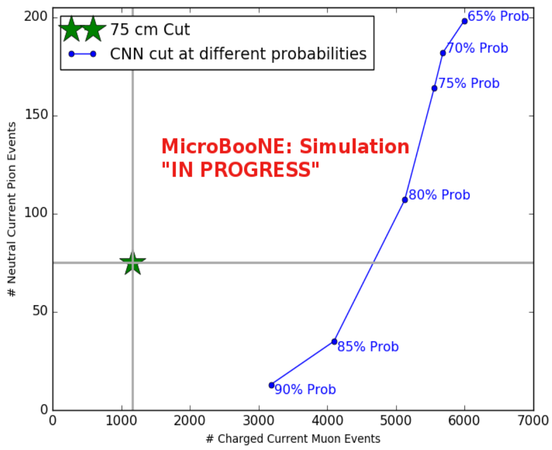


Figure 8.7: CNN performance of classified muons and pions compared to the already implemented 75 cm track length cut

8.1.2 Conclusions of CNN10000 classification of MC data

It was shown that even though CNN10000 was trained with single particle generated muons and pions, it performs fairly well at classifying track candidate images from BNB+Cosmic events. Events have been regained below the 75 cm track length cut and the momentum and track range distributions have similar shapes to the distributions of Selection I. Efficiencies and purities were calculated for selection I events before 75 cm track length cut with the CNN at 83% probability and are 14% and 62% respectively. Although the CNN doesn't have separation between muons and pions and although all particles passing CNN are classified as muon, increasing CNN probability allows us to increase the purity as well as maintain an efficiency comparable to the 75 cm track length cut all while recovering events below that 75 cm cut. Out of the 6142 events that passed the CNN @ 83% 1470 events were below the 75 cm cut, a recovery of 3.3% of data with an purity of 15%. Although these numbers are low, it is an improvement from the selecion I in both total efficiency and purity and an increase in phase space by recovering these events.

8.2 Classification using CNN100000

CNN100000 was trained using 20,000 images of each $\mu/\pi/p/\gamma/e$. For classification of BNB+Cosmics and data, images were made from track candidates that passed the Selection I filter, however, unlike for classifying BNB+Cosmics using CNN10000, the classification of CNN100000 went further up Selection I's cut chain. For CNN100000, steps 1,2,3 and 4 seen in 5.5. The results of using CNN100000 to classify BNB+Cosmics will be outlined below.

8.2.1 Classification of MC data using Selection I CC-Inclusive Filter

Kinematic truth distributions of BNB+Cosmic events passing Selection I+CNN10000

8.2.2 Classification of MicroBooNE data using Selection I CC-Inclusive Filter

8.2.3 Comparing two CC-Inclusive Cross Section Selection Filters

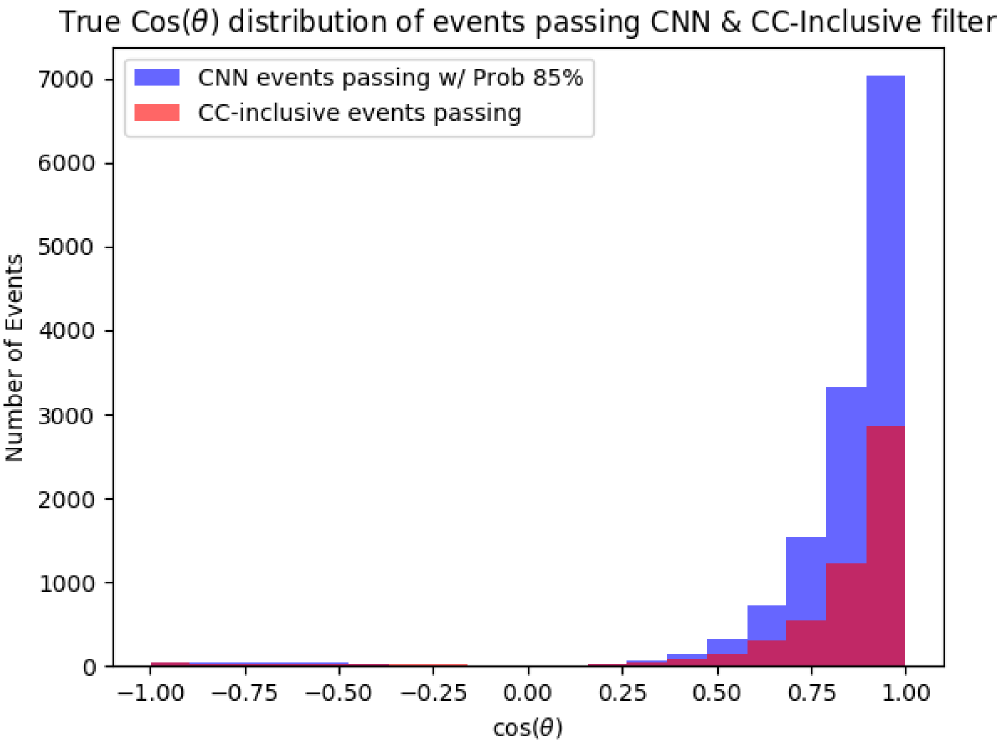


Figure 8.8: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

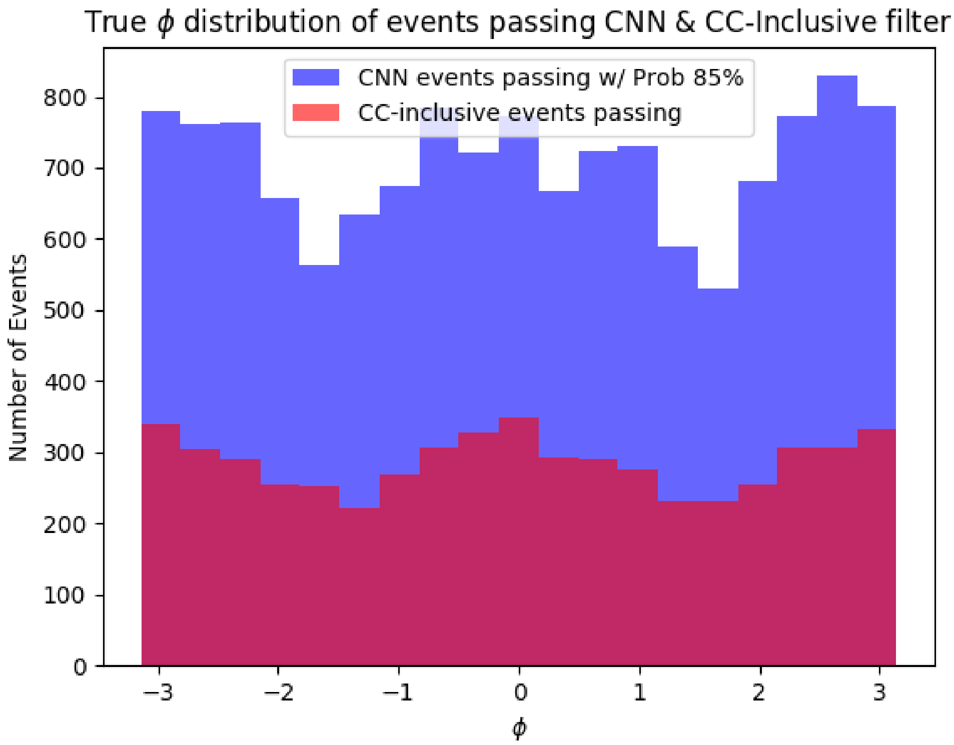


Figure 8.9: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

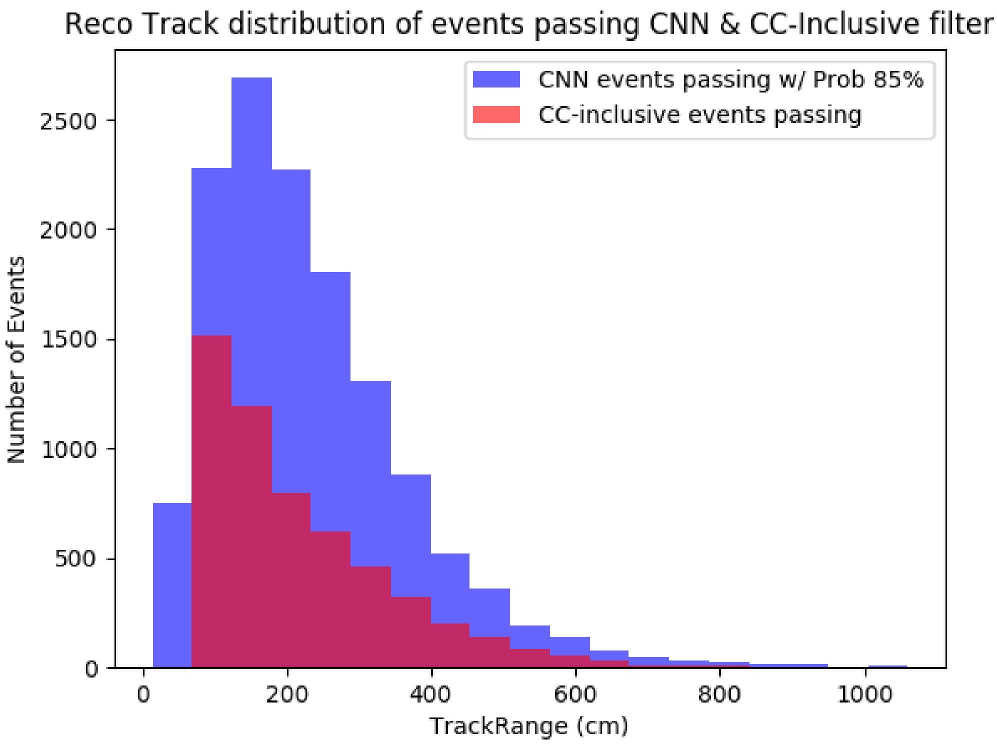


Figure 8.10: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

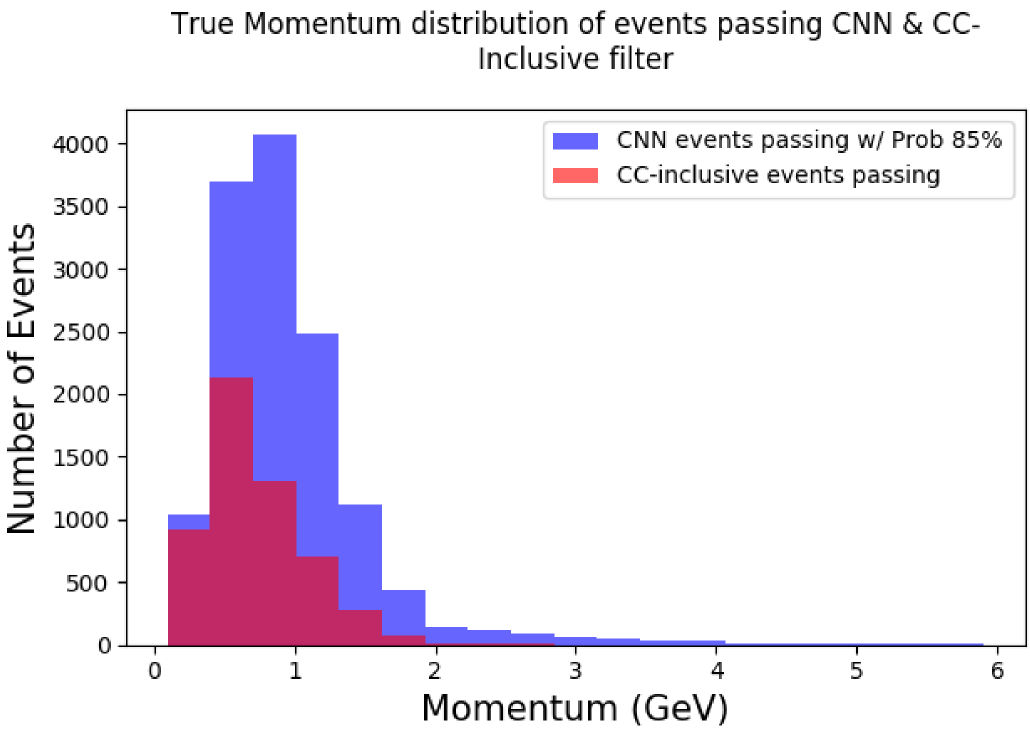


Figure 8.11: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

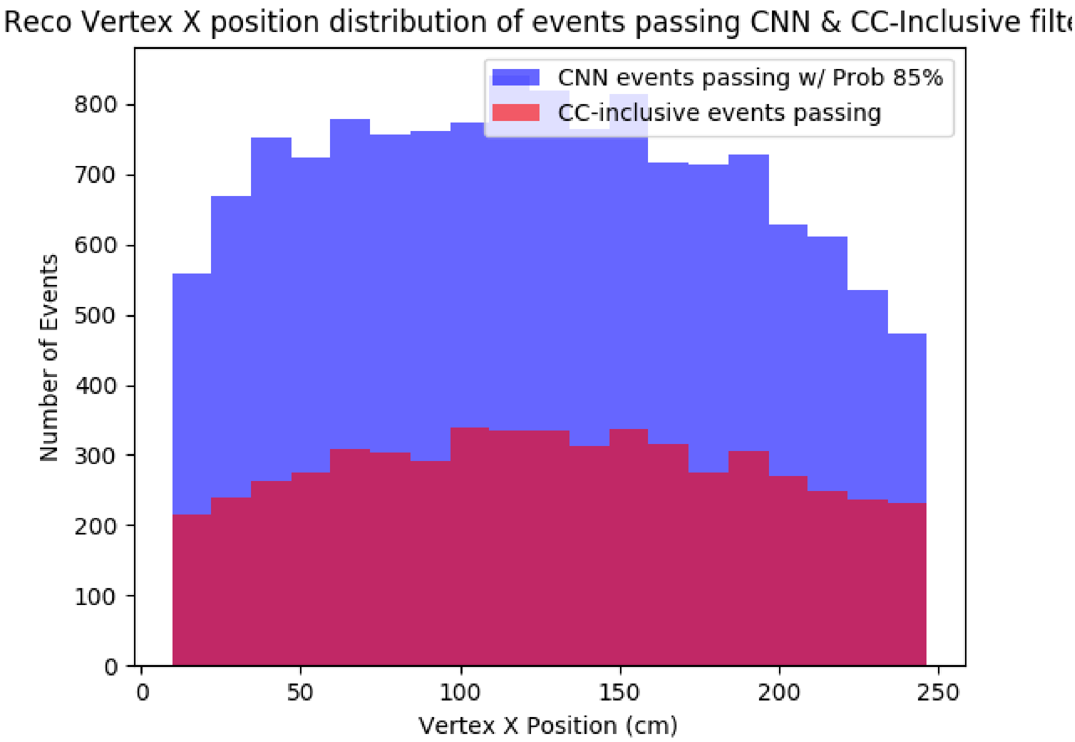


Figure 8.12: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

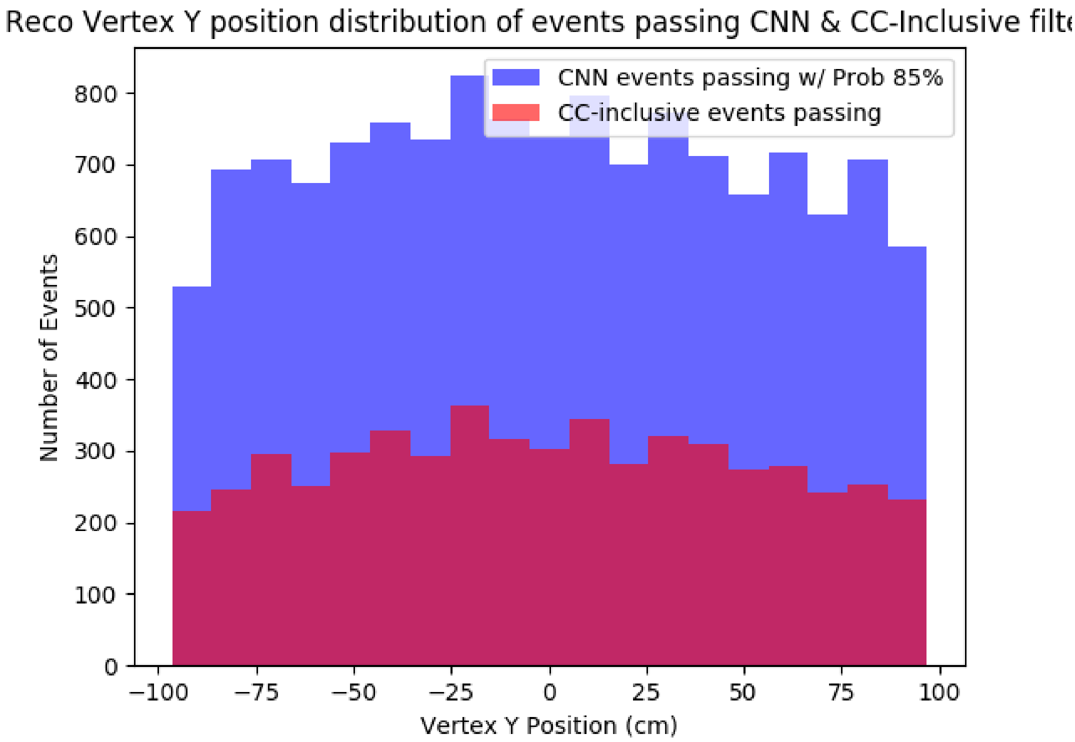


Figure 8.13: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

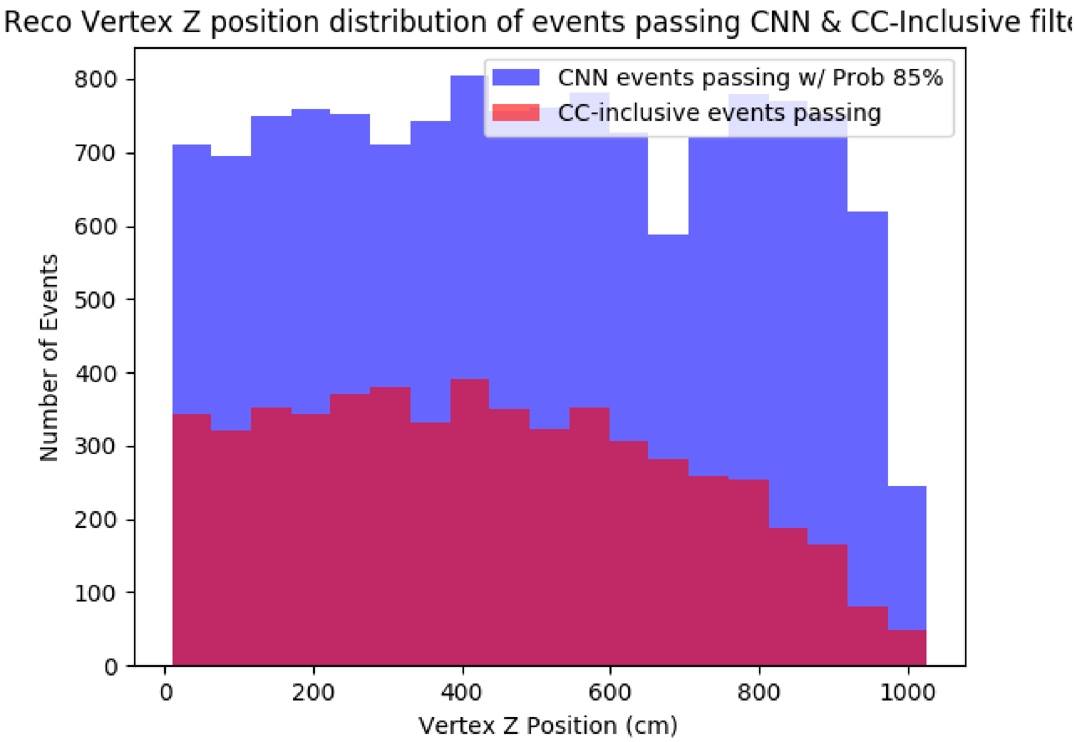


Figure 8.14: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

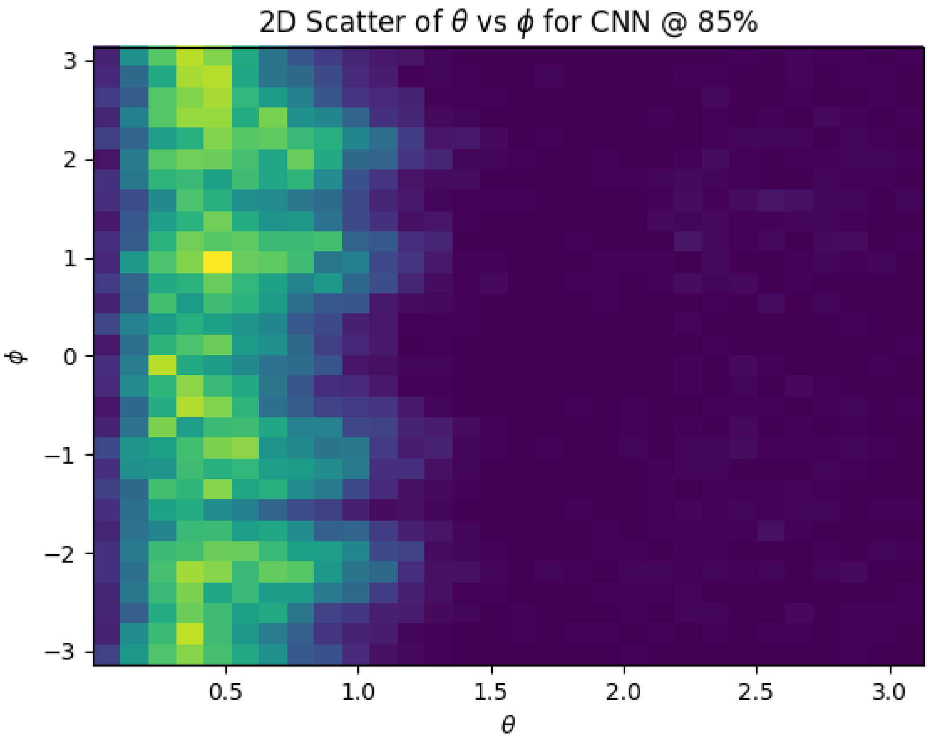


Figure 8.15: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

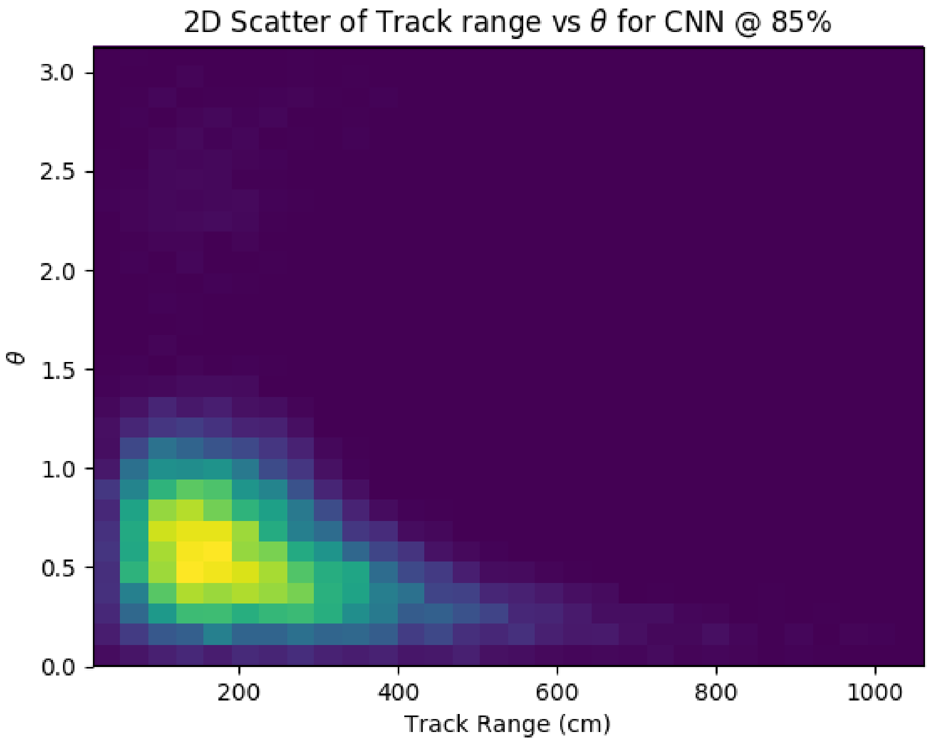


Figure 8.16: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

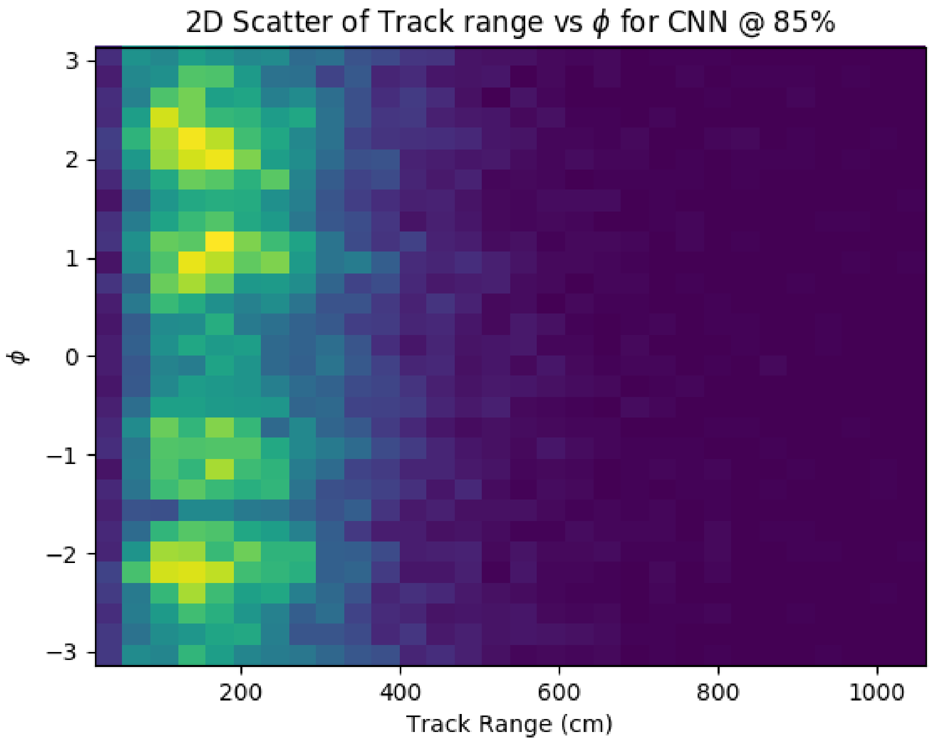


Figure 8.17: $Cos(\theta)$ distribution at CNN10000 $\geq$ 85%

Chapter 9

Conclusion

Your Conclusions here.

Bibliography

[1] Wikipedia, Neutrino, http://wikipedia.org/wiki/Neutrino, 2013.

[2] Wikipedia, Neutrino oscillation, http://en.wikipedia.org/wiki/Neutrino_oscillation, 2013.

[3] B. N. Laboratory, Neutrinos and nuclear chemistry, http://www.chemistry.bnl.gov/sciandtech/sn/default.htm, 2010.

[4] K. Heeger, Big world of small neutrinos, http://conferences.fnal.gov/lp2003/forthepublic/neutrinos/.

[5] A. Y. Smirnov, p. 23 (2003), hep-ph/0305106.

[6] The MicroBooNE Collaboration, R. e. a. Acciari, (2015), arXiv:1512.06148.

[7] The MicroBooNE Collaboration, R. e. a. Acciari, (2016), arXiv:1705.07341.

[8] The MicroBooNE Collaboration, R. e. a. Acciari, (2017), arXiv:1704.02927.

[9] The MicroBooNE Collaboration, R. e. a. Acciari, JINST **12** (2017).