A Search for Sterile-Neutrino-Based Muon Neutrino Disappearance using the MicroBooNE Deep Learning Analysis

a dissertation submitted by

Joshua Michael Mills

in partial fulfillment of the requirements for the degree of
Doctor of Philisophy
in
Physics

Tufts University
May 2022
Advisor: Dr. Taritree Wongjirad

Abstract

We describe a search for ν_μ disappearance using the MicroBooNE Deep Learning analysis' $1\mu1p$ selection. Presently, the unexplained MiniBooNE and LSND anomalies could be explained by a sterile neutrino impacting neutrino oscillations. Our analysis searches for the allowed parameter space that could describe such a sterile neutrino. We determine the allowed and excluded region of a 3+1 sterile-based muon neutrino disappearance model in MicroBooNE at 90% confidence. Our allowed region includes both the null model, and current global best fit model. Context for the underlying Deep Learning analysis is provided and several validation studies surrounding both the disappearance search, and originating $1\mu1p$ selection are performed to strengthen confidence in the result. In addition, a next-generation deep learning tool for cosmic-ray-muon event discrimination is proposed and evaluated, demonstrating a removal of 70% of the remaining event background when added to current methods, under the cut criteria used.

Dedication

To family and friends.

Acknowledgements

This work would not be possible without the help of so very many people. In order to accomplish this, get through graduate school, and maintain my sanity through a global pandemic I owe thanks to you all.

First, I want to thank my advisor, Dr. Taritree Wongjirad, for helping guide my research, teaching me all the auxiliary tools I've used along the way, and being patient as I pick up various new skills across my graduate career. You always brought a passion to our meetings, and a willingness to explore new analysis tools. Not to mention you are a pretty cool advisor even if you call yourself a dinosaur. DOOM, and FF7 conversations come to mind. Next I want to thank the rest of my committee, Doctors Hugo Beauchemin, Hugh Gallagher, Georgia Karagiorgi, and Roger Tobin.

I need to also thank Dr. Katie Mason, my partner in this adventure since our undergraduate at Colgate University. You've been a constant source of support, both personal and professional through this entire process. I cannot imagine what it would have been like to accomplish this without you.

Next I want to thank the MicroBooNE Collaboration, and in particular the various Deep Learning Analysis team members, past and present: Dr. Kazuhiro Terao, Dr. Victor Genty, Dr. Ralitsa Sharankova, Dr. Ran Itay, Dr. Adrien Hourlier, Dr. Matthew Rosenberg, Dr. Lauren Yates, Dr. Rui An, Dr. Jarrett Moon, Dr. Davio Cianci, Dr. Katie Mason, Nicholas Kamp, Elizabeth Hall, and Polina Abratenko. As well as Dr. Janet Conrad for directing our bunch of enthusiastic DLers alongside Taritree! Further thanks go to Dr. Georgia Karagiorgi for helping me take over

Davio's analysis, and also to Dr. Mike Shaevitz for always offering statistical advice.

I also appreciate the solidarity and friendship of so many people in the Tufts Physics Department, with special mentions for my year soon-to-be-doctors Alec Drobac and Colette Kaya, Dr. Katie Mason, Dr. Andrew Clark, Dr. Mathew Giso, Dr. Prantik Saha, Dr. Mudit Jain, and Nick Krefting. We nine were the best group of eight ever. That first couple years of classes were great to take and bond alongside you all. I also want to thank Felix J. Yu for his help, inquisitive spirit, and patience, when it came to masking cosmic muons.

Next I want to thank several of my friends from my hometown, Ali, Kelsey, and Vivian, (with Ian and Evan popping in occasionally!) you all helped keep me sane when a sudden global pandemic hit and the world started to look a little darker, you helped keep the lights on with games, and more importantly, friends to talk to. Similarly, my dungeons and dragons/board games/movie night group in Medford, Alec and Katie again, Hannah Johnston, Terence Blue, Dr. Anastasia Yandulskaya-Blue, Dr. Travis Olson, and Laura Rodríguez-Pérez. Fictional adventures, real friends.

I also want to thank my dog, Cinders, who makes an appearance in this thesis, and the fishes Lancelot and Merlin for the constant support.

Last, but never least, I want to thank my family. My parents, Linda and Chris Mills, thank you for always encouraging me to follow my passions, helping keep me on track and making sure that I'm okay. All these smarts and all this success are possible because I get it from you.[1] I also appreciate the endless support from my sister, Stephanie Mills, my aunts and uncles, and in particular my aunt Cathyrn Myers, who has always encouraged me to pursue a higher-higher education.

I also wish to thank the US Department of Energy. Specifically this work was supported by US DOE grant DE-SC0019032.

---

[1] And a special thanks for my dad's advice to make sure my thesis hits its target length by being very, very, very, very, very, verbose.

Contents

List of Figures **9**

List of Tables **22**

1 Introduction **24**

2 Neutrinos **27**

2.1 Neutrinos Background 27

2.2 The Standard Model and Neutrinos 28

2.3 Neutrino Tension in the Standard Model 31

2.3.1 Building to Massive Neutrinos 32

2.3.2 Neutrino Oscillations 33

2.3.3 Neutrino Oscillation Experimental Limits 37

3 The MicroBooNE Experiment **41**

3.1 Origins in MiniBooNE 42

3.2 The Booster Neutrino Beamline 44

3.2.1 Making Neutrinos with a Proton Beam 45

3.2.2 Neutrino Beam Composition 47

3.3 The MicroBooNE LArTPC 49

3.3.1 LArTPC Setup 49

3.3.2 PMT Light Collection System 51

3.3.3 Event Triggering and Electronics Readout 52

3.3.4 The MicroBooNE CRT 55

3.3.5 Unresponsive Wire Regions 56
3.4 Sterile Neutrinos 59
3.4.1 3+1 Sterile Neutrino Model 60
3.5 Blinding Policy 62

4 Deep Learning **63**
4.1 Machine Learning Overview 63
4.2 Deep Learning 64
4.2.1 Convolutional Neural Networks 68
4.2.2 The Importance of GPUs 70
4.3 Boosted Decision Trees 70

5 The MicroBooNE DL Reconstruction **73**
5.1 Overview 73
5.2 Image Reconstruction 75
5.3 WireCell Charge-Light Cosmic Tagging 79
5.4 SSNet Pixel Labeling 79
5.5 Vertexer 82
5.6 Track Reconstruction 84
5.7 Shower Reconstruction 85
5.8 Multiple Particle Identification 87

6 MicroBooNE DL Selections **91**
6.1 Data Sample Definitions 92
6.2 Building an Expectation 96
6.3 One Lepton One Proton Preselection 98
6.3.1 BDT Input Variables 100
6.4 $1\mu1p$ Selection 104
6.4.1 Background Neutrino Interaction Modes to the MicroBooNE $1\mu1p$ Selection 112
6.4.2 The Important Variables in the $1\mu1p$ BDT Ensemble 114

6.4.3 The $1\mu1p$ CCQE Selection Efficiency 115
6.4.4 $1\mu1p$ BDT Ensemble Run 2 / Run 3 Comparison 117
6.4.5 $1\mu1p$ BDT Ensemble Inter-Run Compatibility 118
6.5 $1e1p$ Selection 119

7 A Sterile Neutrino Search in MicroBooNE 123

7.1 Systematic Uncertainties 124
7.1.1 Detector Systematics 125
7.1.2 Reweightable Systematics 127
7.1.3 Normalization Uncertainty 130
7.2 3+1 Sterile Search in MicroBooNE 132
7.2.1 Method of Disappearance Search 134
7.2.2 Creating an Oscillated Prediction with MicroBooNE Simulation 136
7.2.3 Visualizing Disappearance Model Spectra 137
7.2.4 Calculating Our Test Statistics 139
7.2.5 Theoretical Limit on ν_μ Disappearance Sensitivity 141
7.2.6 Drawing Confidence Levels 142
7.2.7 Feldman Cousins Derivation of R_C 144
7.2.8 Quantifying Sensitivity with Feldman Cousins Method 147
7.2.9 Drawing Pseudo-Experiments with SBNfit 150
7.2.10 Pseudo-Experiment Cross-Check 151
7.3 Validation Studies 152
7.3.1 Signal Injection Tests 152
7.3.2 Pseudo-Experiment Allowed Regions 155
7.3.3 Stats-Only Analysis 156
7.3.4 Scaling Test 159
7.3.5 Fake Dataset Tests 160
7.4 3+1 Sterile Neutrino MicroBooNE Data Analysis 162

8 **Next Generation Cosmic Ray Tagging with sMask-RCNN** **167**
8.1 Abstract . . . 167
8.2 Introduction . . . 168
8.3 Background . . . 171
8.3.1 The MicroBooNE LArTPC . . . 171
8.3.2 Existing cosmic identification tools . . . 172
8.3.3 Object detection and Mask-RCNN . . . 173
8.4 Methods . . . 175
8.4.1 Data preparation . . . 175
8.4.2 Sparse submanifold convolutions . . . 176
8.4.3 Sparse Mask-RCNN . . . 176
8.4.4 Network training . . . 179
8.5 Comparing dense and sparse performance . . . 181
8.5.1 Efficiency and purity . . . 182
8.5.2 Interaction coverage . . . 186
8.5.3 Network comparisons discussion . . . 187
8.6 Finding electron neutrinos with sMask-RCNN . . . 190
8.6.1 Electron neutrino identification . . . 191
8.6.2 Cosmic-only event veto . . . 194
8.7 Conclusions . . . 201
8.8 Acknowledgements . . . 201

9 **Conclusion** **203**

10 **Bibliography** **205**

A **Boosting** **212**

B $1\mu1p$ **BDT Selection Variable Distributions** **215**

C **List of Abbreviations** **233**

List of Figures

2.1 The Standard Model of Particle Physics. Own work by uploader, PBS NOVA, Fermilab, Office of Science, United States Department of Energy, Particle Data Group [1] 30

2.2 Examples of the two types of weak interaction. An example of a CC (left) interaction where a W^+ is exchanged as a ν_μ interacts with a neutron. A NC (right) interaction is also shown, where a ν_e interacts with an electron. 31

2.3 Pictorial representation of the possible neutrino mass hierarchies. Note Δm^2_{atm} is equivalent to Δm^2_{32} and Δm^2_{sol} is equivalent to Δm^2_{21}. [2]. For up-to-date makeups of the different flavor and mass states, see [3]. 37

3.1 A diagram of the SBN Program at Fermilab, with various neutrino detectors including MicroBooNE's, placed along the neutrino beam. [4]. 42

3.2 The MiniBooNE detector enclosure (left) and a cut-away drawing (right) of the detector showing the distribution of PMT's in the signal and veto regions [5]. 43

3.3 A photograph of the inside of the MiniBooNE detector [6]. 44

3.4 The observed MiniBooNE low-energy-excess in ν_e CCQE data. [7] . . 45

3.5 The MiniBooNE pulsed horn system. The outer conductor (gray) is transparent to show the inner conductor components running along the center (dark green and blue). The target assembly is inserted into the inner conductor from the left side. In neutrino-focusing mode, the (positive) current flows from left-to-right along the inner conductor, returning along the outer conductor. The plumbing associated with the water cooling system is also shown [8]. 46

3.6 A cartoon of neutrino beam creation for the BNB. A proton beam instigates hadron production at the target, the horn focuses the desired charged pions, which then decay into neutrinos, while oppositely charged pions get sent away. 47

3.7 The absolute neutrino flux prediction through the MicroBooNE detector as calculated by the beam simulation. Shown is the flux for ν_μ, $\bar{\nu}_\mu$, ν_e, and $\bar{\nu}_e$averaged through the TPC volume with dimensions 2.56m×2.33m×10.37m. [9] 48

3.8 Schematic of the cross section of the MicroBooNE LArTPC. In this view, the beam would be directed out of the page (in the z direction). [10]. 50

3.9 A diagram of the LArTPC principle. The signal formation for the second induction plane (V plane) and the collection plane (Y plane) are shown [10]. 51

3.10 A depiction of the coordinate systems defined by MicroBooNE when discussing locations within the LArTPC. 53

3.11 A schematic of the light collection system as arrayed within the MicroBooNE LArTPC, the 32 PMTs are shown as circles, with the 4 light guide paddles as rectangles. 54

3.12 The scintillation spectra of Argon (red-dashed) along with the TPB re-emission spectra (green-dashed) are shown with the absorption spectra in % of TPB (green-solid), the borosilicate glass (black-solid), and platinum (blue-solid) components of the PMT arrays [11]. 55
3.13 MicroBooNE LArTPC and PMT signal processing and readout stages [10]. 56
3.14 The unresponsive regions of the MicroBooNE wire-plane images for all three wire-planes are shown in red against the working regions in blue. Columns > 2400 for the U and V planes are shown in grey-white because those planes only feature 2400 wires. The construction of these images are defined in section 5.2 58

4.1 An example of an artificial neuron and its inputs. 65
4.2 Several examples of activation functions. 66
4.3 A cartoon example depicting gradient descent in a simplified loss-space. The ball represents the network rolling down toward a minimum loss. 67
4.4 An image of the Tufts Neutrino Group in 2018 (left) and one labeled by a traditionally trained Mask-RCNN network (right), where objects are found, classified, and their pixels clustered. 68
4.5 A comparison of FLOPs between GPUs and CPUs. Courtesy of fast.ai, [12] 71
4.6 A cartoon depiction of a Decision Tree. 71

5.1 A flowchart of the workflow for the Deep Learning analysis team's reconstruction. 76
5.2 An example event's wire image displays. 78
5.3 The wire image pixels that are flagged as coming from a cosmic ray muon. 80

5.4 The wire image pixels that are *not* flagged as coming from a cosmic ray muon. 81
5.5 The pixel labeling performed by SSNET. Shower-labeled pixels are shown as blue, while track-labeled pixels are shown as red 83
5.6 The tracks reconstructed by the tracking algorithm are shown in red, stemming from the candidate vertex represented by a black open circle. 86
5.7 Y Collection Plane 88
5.8 The shower triangle reconstructed from the candidate vertex in our example event. 88
5.9 Y Collection Plane 90
5.10 MPID example of a $1e1p$ topology with a tabulated output of particle scores. This image is generated by concatenating a p and an e^- at the same vertex. Scores indicate high probabilities of having a p and e^- in the image. The image applied to MPID has 512×512 pixels. A zoomed-in image of 250×250 pixels is shown here for visualization. [13] 90

6.1 The distribution of tune weights applied to the BNB (left) and ν_e (right) overlay events. 97
6.2 The reconstructed neutrino energy distribution at the preselection stage, broken down by interaction category. Data not shown. 107
6.3 The BDT ensemble average score distribution at the preselection stage, broken down by interaction category. Data not shown. 108
6.4 The neutrino energy distribution for the selection after the BDT cut is applied, much of the background is removed, and data is shown. . . 109
6.5 The proton MPID score distribution after the BDT cut is applied for events with neutrino energy reconstructed less than 400 MeV. 110
6.6 The distribution of $cos(\theta_p)$ after the BDT cut, specifically in the backward-going proton region. 111

6.7 The range-based reconstructed neutrino energy distribution for the $1\mu1p$ final selection. 112
6.8 Example of a charged-current resonant interaction. A muon neutrino interacts with a proton to give a muon and a resonant Δ^{++} which then decays into a π^{+} and a proton. 113
6.9 The relative importance of each input variable distribution to the training of a typical $1\mu1p$ BDT in the Run 1 Ensemble (top left), Run 2 Ensemble (upper right), and Run 3 Ensemble (bottom middle). This highlights how effective each variable is for differentiating signal from background. For discussion of variables, see tables 6.3 and 6.4. . 116
6.10 The efficiency of selecting CCQE $1\mu1p$ events, annotated to explain the major features. The integrated efficiency across this energy range is 0.0327. 117
6.11 The data-to-data comparison across Run 2 and Run 3 in reconstructed neutrino energy. 118
6.12 The data-to-data comparison across Run 2 and Run 3 in reconstructed neutrino energy. 119
6.13 The comparison in BNB overlay for reconstructed neutrino energy, when using the Run 3 BDT ensemble for all run periods. 120
6.14 The comparison in data for reconstructed neutrino energy, when using the Run 3 BDT ensemble for all run periods. 121
6.15 The constrained prediction for the ν_e signal and ν_μ background events. The MiniBooNE LEE signal expectation is included as a blue dashed line. [14]. 122

7.1 The total, fractional systematic covariance matrix for the $1\mu1p$ analysis. 126
7.2 The fractional systematic covariance matrix for all the detector systematics in the $1\mu1p$ analysis. 127
7.3 The fractional systematic covariance matrix for the cross-section uncertainty in the $1\mu1p$ analysis. 129

7.4 The fractional systematic covariance matrix for the flux uncertainty in the $1\mu1p$ analysis. 129

7.5 The fractional systematic covariance matrix for the re-interaction uncertainty in the $1\mu1p$ analysis. 130

7.6 The uncorrelated fractional systematics broken down by type. The statistical uncertainty is also shown for comparison. 131

7.7 The full, systematic covariance matrix for the $1\mu1p$ analysis, scaled to the null disappearance model's expectation. 132

7.8 A cartoon of the method by which one builds a ν_μ disappearance spectrum from an initial prediction under the 3ν hypothesis. 137

7.9 Another cartoon of ν_μ disappearance, however now using 50-MeV-wide bins for a coarser spectrum. 137

7.10 For all 25 values of Δm^2, the resulting spectra after disappearance is shown against the null model spectrum. As $\sin^2 2\theta$ is an amplitude, the maximum grid value is used for each plot to maximize the disappearance effect. 138

7.11 An example of a higher Δm^2, but low $\sin^2 2\theta$ disappeared spectrum that is still very close to the null model spectrum. 139

7.12 MicroBooNE's 90% sensitivity contours for a shape-and-rate (S+R) parameter-space scan. The contours are drawn assuming a R distribution with 2 degrees of freedom. MiniBooNE's ν_μ disappearance sensitivity is overlaid [15]. Additionally a vertical line at $\sin^2 2\theta_{\mu\mu} = 0.367$, the predicted high-$\Delta m^2$ sensitivity calculated from our normalization uncertainty. Finally the sensitivity for a 1-bin, rate-only disappearance analysis using Wilks' theorem is also shown. 144

7.13 Distribution of R across 1000 pseudo-experiments for the null, no-disappearance, hypothesis as Θ_T, shown in blue on the left. Another distribution for 1000 pseudo-experiments for an oscillation hypothesis with a large disappearance model, with parameters Δm_{41} : 3.0, $\sin^2(2\theta_{\mu\mu})$: 0.91 is on the right. Each plot contains expected distributions for two degrees of freedom. Vertical lines are drawn for the 90% CL of each distribution, such that 90% of the distribution is to the left of the vertical line. 146

7.14 The critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario. . 147

7.15 The test R values across every point in our parameter space. The left plot displays the raw R value, while the right shows the same values, with values below 0.1 set to 0.1 and a log scale applied for better visualization. 149

7.16 MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. MiniBooNE's ν_μ disappearance sensitivity is overlaid [15]. 149

7.17 Plot of simulated events with ν_μ flux oscillated away with parameters $\sin^2 2\theta_{\mu\mu} = 0.76$ and $\Delta m^2_{41} = 0.69$ eV^2 (in red). The other colored histograms represent 1000 pseudo-experiments drawn around this central value. Overlaid are the 1, 2, and 3σ uncertainties from a covariance matrix scaled to the disappeared CV (and with Poisson uncertainty) for each bin. 152

7.18 MicroBooNE's sensitivity to ν_μ disappearance at 90% confidence, with four points indicated where signal was injected for the cross-checks described in this chapter. Interpretations of the signal injection can be found in the text. 154

7.19 For various pseudo-experiments thrown from different injected signals we show the 90% confidence limit allowed regions. 156

7.20 The fractional covariance matrix in a statistics-only regime. 157

7.21 For the stats-only regime, the critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario. 158

7.22 For the stats-only regime, MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. Both our normal Feldman-Cousins analysis, and MiniBooNE's ν_μ disappearance sensitivity are overlaid [15], though MiniBooNE's limit still includes systematic errors, and is shown only as a benchmark. 158

7.23 For the 40x scaled regime, the critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario. 159

7.24 For the 40x scaled regime, MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. Both our standard Feldman Cousins sensitivity and MiniBooNE's ν_μ disappearance sensitivity is overlaid [15]. 160

7.25 Fake dataset 1's spectrum (left) and the allowed regions (right) when run through the ν_μ disappearance search. 161

7.26 Fake dataset 2's spectrum (left) and the allowed regions (right) when run through the ν_μ disappearance search. 162

7.27 The R_{Data} values across every point in our parameter space where we use our 6.67×10^{20} POT worth of data as our observation. The left plot displays the raw R value, while the right shows the same values, with values below 0.1 set to 0.1 and a log scale applied for better visualization. 163

7.28 The selected neutrino interaction spectrum as a function of energy. In blue the null model expectation, with the uncertainty band surrounding it, in black our observation, the data, and in red the best fitting 3+1 sterile neutrino model. 164

7.29 Using 6.67×10^{20} POT worth of MicroBooNE data we show the allowed regions of 3+1 model phase space in green, and the excluded region in white in the upper right of the plot, our Feldman Cousins sensitivity is overlaid in blue. MiniBooNE's excluded curve is also overlaid in red [15]. 166

8.1 Several example event images. The vertical and horizontal scales are the same for all images. Each column of pixels along the x-axis refers to a specific wire readout, and each row along the y-axis refers to a different bin of signal readout time. This is described in greater detail in section 8.4.1. (a) is an example of an input image given to sMask-RCNN to process, whereas (b) shows the network's subsequent labeling of the same image. (c) shows a cosmic-only data event. (d) shows a data event containing a neutrino interaction that sMask-RCNN correctly identifies with some confidence score, and clusters. . 170

8.2 A diagram of the LArTPC principle. The signal formation for the second induction plane (V plane) and the collection plane (Y plane) are shown [10]. 171

8.3 Network Architecture for Mask-RCNN in MicroBooNE. 174

8.4 An example of a convolution operation that depends on use of submanifold or normal convolutions. The normal convolution multiples the kernel against the image at the given position, and outputs a convolved value of 17. Meanwhile, the submanifold convolution does not get computed when centered on a zero. The submanifold convolutions used in sMask-RCNN have a kernel size of 3×3 (same as shown in the figure), with a stride of 1. 177

8.5 A visual representation of the definition of efficiency. Only nonzero pixels in the event image are counted. 183

8.6 A visual representation of the definition of purity. Only nonzero pixels in the event image are counted. 183

8.7 The event-averaged efficiencies and purities for the dense and sparse implementations of Mask-RCNN. The dense network has a mean event-averaged efficiency of 0.89 and a mean event-averaged purity of 0.87. For the sparse network these values are 0.86 and 0.85. Each of these evaluations use the same validation dataset. 184

8.8 The one-dimensional distributions for the event-averaged efficiency (a) and purity (b) shown in figure 8.7. Each plot compares the dense and sparse network performances. 185

8.9 The interaction-level efficiency of the dense and sparse versions of Mask-RCNN as measured on the validation set. 186

8.10 The interaction-level charge efficiency of the dense and sparse versions of Mask-RCNN as measured on the validation set. 187

8.11 A zero-efficiency true interaction almost entirely in a region of unresponsive wires. The white box shows a zoom-in of the area of interest, and within it, the colored box should contain true neutrino interaction. However, because this interaction falls in an unresponsive region of the detector, no deposited charge is seen inside the colored box. . . 187

8.12 Another failure mode for the zero-efficiency peak. Here the cosmic interaction is relatively small in size compared to others in the event image. The colored box is the true interaction, and the white box shows a zoom-in of the area of interest. 188

8.13 The fraction of true interactions in events that have greater than 80% efficiency as measured on the validation set. 188

8.14 The number of covered interactions is plotted against the number of true interactions. A dashed line along $y = x$ represents the absolute perfect performance, with all true interactions being covered in each event. Out of an average of 20.8 true interactions per event, the dense network covers an average of 18.2, while the sparse network covers 17.2. 189

8.15 The efficiency of sMask-RCNN broken down by class. The average efficiency is 76.8% for electron neutrinos, and 86.1% for cosmic ray muons. Statistical uncertainty bars are shown. 192

8.16 The charge efficiency of sMask-RCNN broken down by class. The average charge efficiency for electron neutrinos is 77.9%, and 86.8% for cosmic ray muons. Statistical uncertainty bars are shown. 193

8.17 The purity of sMask-RCNN broken down by class. Predicted interactions with zero purity are misclassified, for example a true neutrino labeled as a cosmic ray. The average purity is 64.9% for electron neutrinos and 84.7% for cosmic ray muons. Statistical uncertainty bars are shown. 194

8.18 The size of the largest cluster found by DBScan after cosmic tagging. 196

8.19 ROC curves for the sMask-RCNN with and without WC Q-L matching based on a requirement on largest cluster size. Curves are shown for the two electron neutrino samples against the off-beam background. 198

8.20 The size of the largest cluster found by DBScan after cosmic tagging. Note the logarithmic scale. 199

8.21 ROC curves for the WC cosmic tagger with and without sMask-RCNN based on a requirement on largest cluster size. Curves are shown for the two electron neutrino samples against the off-beam background. Note the suppressed y-axis shown here demonstrates significant improvement in background removal compared to figure 8.19. 200

B.1 The α_T distribution for the $1\mu1p$ selection. 216
B.2 The Boosted Bjorken's x distribution for the $1\mu1p$ selection. 216
B.3 The Boosted Bjorken's y distribution for the $1\mu1p$ selection. 217
B.4 The distribution for the amount of charge gathered around the interaction vertex for the $1\mu1p$ selection. 217
B.5 The range-based reconstructed neutrino energy distribution for the $1\mu1p$ selection. 218
B.6 Via the $E_\nu^{QE-\ell}$ assumption, the reconstructed neutrino energy distribution for the $1\mu1p$ selection. 218
B.7 Via the E_ν^{QE-p} assumption, the reconstructed neutrino energy distribution for the $1\mu1p$ selection. 219
B.8 The $Cos(\theta)$ distribution for the muon in the $1\mu1p$ selection. 219
B.9 The ϕ distribution for the muon in the $1\mu1p$ selection. 220
B.10 The θ distribution for the muon in the $1\mu1p$ selection. 220
B.11 The muon track length distribution for the $1\mu1p$ selection. 221
B.12 The MPID score distribution for an interaction containing an electron in the $1\mu1p$ selection. 221
B.13 The MPID score distribution for an interaction containing a muon in the $1\mu1p$ selection. 222
B.14 The MPID score distribution for an interaction containing a proton in the $1\mu1p$ selection. 222
B.15 The muon's reconstructed kinetic energy distribution for the $1\mu1p$ selection. 223

B.16 The interaction opening angle distribution for the $1\mu 1p$ selection. . . 223

B.17 The distribution for the difference in ϕ variables between the proton and muon for the $1\mu 1p$ selection. 224

B.18 The transverse ϕ distribution for the $1\mu 1p$ selection. 224

B.19 The $Cos(\theta)$ distribution for the proton in the $1\mu 1p$ selection. 225

B.20 The proton's reconstructed kinetic energy distribution in the $1\mu 1p$ selection. 225

B.21 The ϕ distribution for the proton in the $1\mu 1p$ selection. 226

B.22 The θ distribution for the proton in the $1\mu 1p$ selection. 226

B.23 The proton track length distribution for the $1\mu 1p$ selection. 227

B.24 The transverse momentum distribution for events in the $1\mu 1p$ selection.227

B.25 The distribution of the ratio of transverse momentum to total momentum in the $1\mu 1p$ selection. 228

B.26 The Q_0 distribution in the $1\mu 1p$ selection. 228

B.27 The momentum transfer squared, Q^2, distribution in the $1\mu 1p$ selection.229

B.28 The z-component of the momentum transfer squared, Q_3, distribution in the $1\mu 1p$ selection. 229

B.29 The BDT Ensemble score distribution in the $1\mu 1p$ selection. 230

B.30 The QE consistency distribution in the $1\mu 1p$ selection. 230

B.31 The distribution of sum of θ_p and θ_m in the $1\mu 1p$ selection. 231

B.32 The distribution of reconstructed x position of the interaction vertex in the $1\mu 1p$ selection. 231

B.33 The distribution of reconstructed y position of the interaction vertex in the $1\mu 1p$ selection. 232

B.34 The distribution of reconstructed z position of the interaction vertex in the $1\mu 1p$ selection. 232

List of Tables

2.1 Experiments contributing to the present determination of the oscillation parameters [3] . . . 39

3.1 Particle lifetimes, and neutrino-producing decay modes and branching ratios considered in the simulation[8]. . . . 46

3.2 MicroBooNE LArTPC design parameters and nominal operating conditions. [10] . . . 52

3.3 The wire counts, and unresponsive wire counts for the three different wire-planes in the MicroBooNE LArTPC. . . . 56

6.1 The size, in protons-on-target (POT), of the various datasets used in the DL analysis. . . . 93

6.2 A table representing the scaling process of different expected samples to the observation for a *toy model of events*. The POT ratio is calculated by dividing the on-beam sample POT by the expectation sample's POT. . . . 97

6.3 The variables used as inputs to the $1\mu1p$ and $1e1p$ BDTs. If an $*$ appears, the variable is used in the boosted frame of reference. . . . 103

6.4 Kinematic variable definitions as derived from the reconstruction code used in this analysis. . . . 105

7.1 Contributions of each source of systematic uncertainty in MicroBooNE, as used to plot the reconstructed energy distribution in Figure 6.7, to normalization uncertainty. The shape component of each systematic is removed according to Equation 7.3. 131

7.2 The bin centers for our two parameters 135

7.3 The breakdown of components of our test statistic R for the best fit point and the null oscillation model. 163

8.1 The average inference runtimes per 3456 × 1008 pixel image on a CPU. The first row is the runtime for just the ResNet portion of Mask-RCNN on the images. The second row is the time to run the entire network on the images. In the case of sparse ResNet, the time spent making the input image into a sparse tensor and the output features into a dense tensor is included in the sparse ResNet module time. 178

8.2 The different class types and number of occurrences in the training sets for the dense and sparse versions of Mask-RCNN. Note that the sparse network only trained on cosmic ray muon and electron neutrino interactions. 180

8.3 The expected ratio of the two different neutrino sample events to off-beam background events. 191

Chapter 1

Introduction

The frontiers of scientific research have a habit of falling into extremes.

Neutrinos are the lightest-massive particle known to humankind. Not counting massless particles like photons, they're the lightest thing out there. It was only recently we discovered they have mass. At the time this author was born, neutrinos were thought to be massless. Nearly-massless, without any electric charge to control them, they live lonely lives, extremely unlikely to interact with anything, even as they pass right through it.

If neutrinos were rare I would almost hazard a guess to say humans would never find them.[1] But they aren't rare. Neutrinos again fall into an extreme. They are ubiquitous. In a single second, every single second, one-hundred-billion neutrinos passed through your thumbnail.

So it is this small little particle that winds up everywhere, but doesn't particularly like to do anything, is an active area of research. In recent years, there has been experimental evidence of an excess in one flavor of neutrinos, ν_e observed by the MiniBooNE and LSND experiments [7] [16]. This could be explained by a new flavor of neutrino called a sterile neutrino. But this phenomenon would be accompanied by muon neutrino disappearance, as muon neutrinos oscillate away through the newly-introduced sterile mass state into the excess electron neutrinos seen by MiniBooNE and LSND. Our work searches for muon neutrino disappearance in the

---

[1] I don't think I would ever bet against human ingenuity with such finality.

MicroBooNE experiment.

The work within this thesis is broken down into several different parts. Chapter 2 provides background on the Standard Model of particle physics, neutrinos, and neutrino oscillation phenomena, whereby the type of neutrino changes as it moves. Then Chapter 3 provides a brief background on the anomalous experimental motivation for MicroBooNE, before discussing the MicroBooNE neutrino beam and detector apparatus.

As the analysis within this thesis contains several tools designed using combinations of machine learning and deep learning, Chapter 4 provides a useful primer of information regarding these tools. Next the MicroBooNE Deep Learning team's reconstruction framework is described in Chapter 5 which discusses how detector signals are transformed into high-level particle physics interaction attributes.

Chapter 6 describes how these high-level features can be used to create highly-pure selections of $1\mu1p$ and $1e1p$ event topologies. These selections are then used to perform the ν_μ disappearance analysis contained within Chapter 7. This analysis searches for disappearance related to a 3+1 Sterile Neutrino extension to neutrino oscillations. This search is fundamentally important as the yet-to-be-observed sterile neutrino could provide an explanation for the anomalies that motivate the MicroBooNE experiment. Finally Chapter 8 evaluates a technique proposed for future iterations of analysis hoping to further reduce the cosmic-ray-muon background using deep learning tools.

It should be noted that no one in scientific research succeeds alone. Research is built upon the foundations provided by collaborators both past and present. Within the confines of this thesis work performed by other MicroBooNE collaborators is often used to provide context to the analysis performed by the author. As is customary, the author performed service work for the MicroBooNE Collaboration, including but not limited to, late night shifts spent monitoring the detector. The author also was personally responsible for transforming the data products produced by the Wire-Cell Charge-Light matching algorithm into data formats useful within the Deep Learning analysis' reconstruction framework.

The author claims responsibility for taking over the $1\mu1p$ selection as the team worked towards the LEE search performed in [14], which included improving the selection and validating the performance as well as providing support on the LEE project. Additionally, the author is personally responsible for the work performed in the ν_μ disappearance search associated with the same $1\mu1p$ selection. Finally, the author was the lead on the sMask-RCNN cosmic ray muon tagging project.

Chapter 2

Neutrinos

In this chapter we provide a primer for neutrinos, the particle of interest regarding this thesis. It begins with a brief description of the proposal and subsequent discovery of neutrinos. Then we hurtle forward to nearly-present day, with the Standard Model of particle physics. Within this model, neutrinos are massless particles, and their place among the other fundamental particles is staked out.

Next the Standard Model is strained to its limit as we describe the solar neutrino problem, and ultimately discuss the solution: massive, oscillating neutrinos. The oscillation formalism for this extension to the Standard Model is described. Finally, we note the various experimental constraints placed around the fundamental parameters describing neutrino oscillations.

2.1 Neutrinos Background

Neutrinos were first proposed by Wolfgang Pauli in 1930 [17].[1] His proposal was made in order to account for the energy and momentum that was otherwise missing in beta decay. While not known at the time, beta decay is a process where a nucleus emits an electron (or positron) accompanied by an antineutrino (neutrino) and a

---

[1] Pauli actually referred to his proposed particles as 'neutrons', as his theory predated the discovery of neutrons. However, to avoid confusion we'll refer to them by their modern-day name: neutrinos.

neutron (proton) within the nucleus. The electron case is described by

$$n \rightarrow p + e^- + \bar{\nu}_e. \tag{2.1}$$

Issues arose as the exiting electron produced a continuous spectrum of energy, rather than a singular value, but the change in nuclear energy was expected to be a narrow band. This produced a result that seemingly contradicted conservation of energy. A similar issue arose as angular momentum was not conserved with a half-spin missing. Pauli rightly theorized that beta decay also produced an electrically neutral, spin-1/2 particle that was able carry away the missing energy and angular momentum. Famously, he believed his proposed particle was so elusive, it could never be detected.

It was not until 1956 that he was proven wrong. Together Clyde Cowans and Frederick Reines set up a liquid scintillator detector a few meters from a 1000 MW nuclear reactor, which gave them a flux of 1000 billion neutrinos per centimeter-squared per second [17]. Then they found positrons from inverse beta decay, which is described as

$$\bar{\nu}_e + p \rightarrow n + e^+. \tag{2.2}$$

This process is kickstarted by the antineutrino bringing enough energy to the nucleus to change the proton to a neutron and create an electron. This antineutrino-induced inverse beta decay gave evidence for the existence of Pauli's neutrino particles.

2.2 The Standard Model and Neutrinos

Neutrinos are one of the fundamental particles within the Standard Model (SM) of particle physics. In the SM, neutrinos are massless leptons with no electric charge, and come in three flavor states as denoted by the charged lepton family the neutrino is associated with. These flavor states are therefore the electron neutrino, muon neutrino, and tau neutrino.[2] As leptons, these states correspond to the three generations of charged-leptons: the electron, muon, and tau particles.

Neutrinos are shown among all the elementary particles of the standard model in figure 2.1. In this figure we see the different flavors of the neutrino and their corresponding charged leptons. We also see quarks, the other fermions that come together to form hadrons. These hadrons can be either baryons or mesons. Baryons consist of three valence[3] quarks (such as an uud proton or udd neutron). These three quarks must combine to yield net zero color charge via quantum chromodynamics (QCD), wherein one quark of each color charge (red, green, blue) yield net zero color charge. In 2015, evidence for a rare five quark pentaquark state was found by the LHCb experiment at CERN [18]. Whereas mesons are made up of a quark-antiquark pair (such as a $u\bar{d}$ π^+ or a $u\bar{u}$ π^0), and satisfy the QCD requirement by having a color and anti-color pair.

Similar to the leptons, these quarks also come in three generations, each containing a pair of quarks, the up (u) and down (d) quarks, the charm (c) and strange (s) quarks, and the top (t) and bottom (b) quarks. As a general rule, quarks in higher generations than the up and down are not naturally occurring, are unstable, and decay quickly into their lower generation versions.

On the right of the SM figure the bosons are shown. Bosons, as opposed to fermions, are particles with integer spin. In red are the four gauge bosons, with spin of 1. These gauge bosons are the force carriers within the SM. The gluon is the force carrier for the strong force, the γ is the carrier of the electromagnetic force, and the Z and $W^{\pm}$ bosons are the carriers of the weak force. A force carrier for gravity, such as the theoretical graviton, has yet to be observed, and is thus outside the SM. Unlike the vector boson force carriers, the Higgs particle is a scalar boson with spin of 0. It should be noted that for the particles listed, there also can exist

---

[2] As well as their antiparticle forms

[3] In the quark model, there can exist a 'sea' of quark and their antiquark pairs (such as $u\bar{u}$) effectively canceling each other out. Valence quarks can be thought of as the net quarks

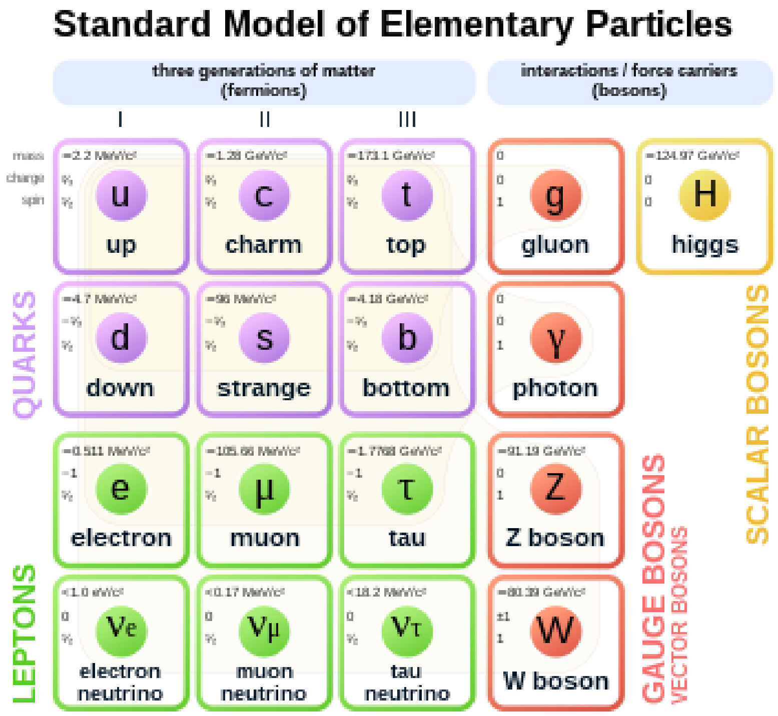
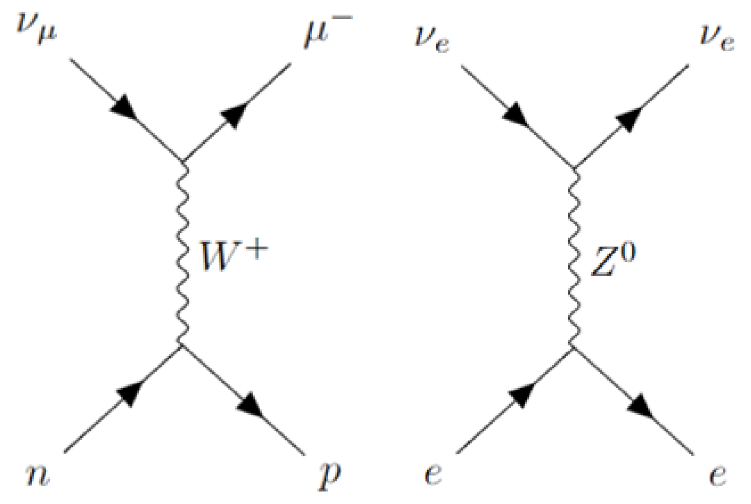


Figure 2.1: The Standard Model of Particle Physics. Own work by uploader, PBS NOVA, Fermilab, Office of Science, United States Department of Energy, Particle Data Group [1]

antiparticle forms, where the particle's innate fundamental charges (such as electric or weak-isospin) are flipped. For example in the case of the up quark u there is the anti-up $\bar{u}$ (u-bar), or for the electron (e^-) there is the positron (e^+). Similarly for the ν_μ there is the $\bar{\nu}_\mu$ which has opposite weak-isospin and lepton number.

Since in the SM neutrinos are chargeless, massless, leptons, they do not interact with electromagnetism's γ, gravity, or the strong force, they only interact with the weak force. These interactions can broadly be categorized in one of two ways, based on the vector boson force carrier involved. Charged-current (CC) interactions are governed by the weak-charged $W^\pm$, and neutral-current (NC) interactions are governed by the Z^0 mediator. Figure 2.2 demonstrates two Feynman diagrams, of a CC (left) and NC (right) interaction. Within these Feynman diagrams, time runs from left to right. This example CC interaction is of particular interest to this thesis, as it demonstrates the principal signal sought by the selection described in section 6.4, and the disappearance analysis present in section 7, as a ν_μ interacts with a



(a) CC Interaction Example (b) NC Interaction Example

Figure 2.2: Examples of the two types of weak interaction. An example of a CC (left) interaction where a W^+ is exchanged as a ν_μ interacts with a neutron. A NC (right) interaction is also shown, where a ν_e interacts with an electron.

nucleon (neutron). Out of this interaction a down quark becomes an up quark as the neutron turns to a proton, and the ν_μ becomes a muon.

Notably, CC interactions, through the exchange of the charged $W^\pm$, conserve lepton family generation number. This corresponds with the fact that experiments have not seen evidence of a lepton-flavor violating process.[4] This is to say that a ν_μ will create a muon, not an electron. Conversely, for NC interactions this is not the case, no weak-charge is exchanged as the Z boson only exchanges energy, momentum, and spin. Thus the lepton number, among other quantum numbers, remains unaffected (and obviously still conserved).

2.3 Neutrino Tension in the Standard Model

The Standard Model of particle physics does a great job of gathering together the particles, and providing an involved description of physics. It is built around the conservation of several different charges (such as weak isospin, electric charge, color charge) associated with the different symmetry groups of quantum field theory

---

[4] Except for neutrino oscillation phenomenology, described below

(SU(3)×SU(2)×U(1)). The resulting model facilitated the prediction of both the top quark and the Higgs boson particle before their eventual discovery. However, there are portions of the SM where it begins to strain and break. One of those major breaking points involves the definition of neutrinos as 'massless'. Next, we build toward an extended Standard Model with massive neutrinos.

2.3.1 Building to Massive Neutrinos

Historically, the flavor states (ν_e, ν_μ, and ν_τ) came about because of observations surrounding beta decay antineutrinos (neutrinos) created with an electron (positron) in beta decay, described in equation 2.1.

These antineutrinos, created in nuclear reactor experiments, always interacted to produce electrons when observed by detectors. Another observation was related to neutrino beam experiments. In these instances the beam was created from charged-pion decay (see section 3.2),

$$\pi^+ \rightarrow \mu^+ + \nu_\mu. \tag{2.3}$$

These neutrinos are created alongside antimuons, then when they interact in the detector they produced muons in CC interactions [19]. These observations led to the concept that the neutrinos produced had a flavor associated with the charged lepton they were associated with.

This observation began to break down as the field of particle physics began to run into what is now referred to as 'the solar neutrino problem'. Nuclear fusion taking place in the sun produces a large flux of ν_e particles, through a variety of nuclear cycles. The highest-energy (15 MeV) electron neutrinos generated by the sun are produced via beta decay around ^{8}B [19]. The first experiment to measure the flux of these neutrinos was the Homestake Mine experiment, which counted the ^{37}Ar argon atoms produced by ν_e interactions in their radiochemical C_2Cl_4 detector. In a puzzling discovery, they observed the rate of neutrino interactions at slightly under a third of their expectation [19, 20]. This neutrino deficit remained as additional radiochemical experiments measured lower-energy electron neutrinos produced from the sun.

The neutrino deficit began to extend into other detector technology. Super-Kamiokande's Cherenkov detector found a deficit in atmospheric muon neutrinos [21] and the Sudbury Neutrino Observatory (SNO) Cherenkov detector again observed deficit of ν_e interactions. However, critically, the SNO experiment had sensitivity to a variety of neutrino channels. Specifically, SNO was sensitive to one channel dependent just on ν_e interactions, and two other channels dependent on the *total* neutrino flux. This allowed the SNO experiment to show that the total neutrino flux received by their detector was consistent with the expected solar flux, consisting of electron neutrinos [22].

This provided strong evidence that neutrinos were changing flavor states as they traveled from the sun to earth. These flavor transformations are known as neutrino oscillations, and are described by attributing to the neutrino that which extends the standard model: mass[5]. Thus the solar neutrino problem was solved by neutrino oscillations.

2.3.2 Neutrino Oscillations

In order to describe the neutrino oscillations observed experimentally, neutrinos are attributed mass. This mass remains very small, sub 1-eV scale, but crucially non-zero. However, each of the three flavors of neutrino are not prescribed unique mass states all their own. Instead, each flavor state is represented as a linear combination of three different neutrino mass states, ν_1, ν_2, and ν_3. Another way of saying this is that the coupling associated with a Feynman diagram's vertex between a W^+, a positron, and a ν_e is actual a linear combination of three vertices between the W^+, the positron, and each of the three ν_i mass states. This is shown for the quantum

---

[5] Massive neutrinos could potentially be described as Majorana particles, which would make them their own antiparticle. However this discussion is beyond the scope of this thesis. A review of Majorana neutrino study is presented in [23]. A yet-to-be-observed, but closely studied phenomenological interaction; neutrino-less double beta decay, would prove that neutrinos are their own antiparticle, and present strong evidence for Majorana neutrinos.

state of an electron neutrino, $|\nu_e\rangle$, as

$$|\nu_e\rangle = U^*_{e1}\,|\nu_1\rangle + U^*_{e2}\,|\nu_2\rangle + U^*_{e3}\,|\nu_3\rangle\,. \tag{2.4}$$

These couplings can be collected in the unitary matrix U, called the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix. In the absence of neutrino oscillations, U is the identity matrix. However, as demonstrated by the experiments outlined previously, this is not the case. As such, the relations between the flavor states and the mass states can be encapsulated in

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix}. \tag{2.5}$$

These terms in the unitary matrix U are fundamental parameters of the extended Standard Model of particle physics. A similar representation can be used to describe the neutrino mass states in terms of the flavor states using the $U^\dagger$ (conjugate transpose) of the PMNS matrix,

$$\begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \begin{pmatrix} U^*_{1e} & U^*_{1\mu} & U^*_{1\tau} \\ U^*_{2e} & U^*_{2\mu} & U^*_{2\tau} \\ U^*_{3e} & U^*_{3\mu} & U^*_{3\tau} \end{pmatrix} \begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix}. \tag{2.6}$$

To observe how a wavefunction describes oscillations from one flavor state to another, we begin by time-evolving the wavefunction for a ν_e in eqn. 2.4. This gives

$$|\nu_e(\vec{x},t)\rangle = U^*_{e1}\,|\nu_1\rangle\, e^{-i(\phi_1)} + U^*_{2e}\,|\nu_2\rangle\, e^{-i(\phi_2)} + U^*_{3e}\,|\nu_3\rangle\, e^{-i(\phi_3)}. \tag{2.7}$$

Here ϕ_i represents the phase of the plane wave for a given mass state. This wavefunction can then be modified by substituting the flavor state linear combinations of $|\nu_1\rangle$, $|\nu_2\rangle$, and $|\nu_3\rangle$. This results in

$$
\begin{aligned}
|\nu_e(\vec{x},t)\rangle = U^*_{1e}(&U_{e1}|\nu_e\rangle + U_{\mu 1}|\nu_\mu\rangle + U_{\tau 1}|\nu_\tau\rangle)e^{-i(\phi_1)} \\
&U^*_{2e}(U_{e2}|\nu_e\rangle + U_{\mu 2}|\nu_\mu\rangle + U_{\tau 2}|\nu_\tau\rangle)e^{-i(\phi_2)} \\
&U^*_{3e}(U_{e3}|\nu_e\rangle + U_{\mu 3}|\nu_\mu\rangle + U_{\tau 3}|\nu_\tau\rangle)e^{-i(\phi_3)}.
\end{aligned}
\tag{2.8}
$$

We are interested in one flavor state of neutrinos oscillating to another flavor state. Therefore, we calculate the probability by computing $|\langle\nu_\mu|\nu_e(\vec{x},t)\rangle|^2$. This represents the probability that a ν_e oscillates to the ν_μ. This is shown to be

$$|\langle\nu_\mu|\nu_e(\vec{x},t)\rangle|^2 = |U^*_{e1}U_{\mu 1}e^{-i(\phi_1)} + U^*_{e2}U_{\mu 2}e^{-i(\phi_2)} + U^*_{e3}U_{\mu 3}e^{-i(\phi_3)}|^2 \tag{2.9}$$

Which represents $P(\nu_e \to \nu_\mu)$. We now adjust this formula for the general case of a generic neutrino flavor state ν_α oscillating to ν_β . We also write the phase term explicitly as $\phi_i = E_i t$ [3], where E is the neutrino's energy and we operate in the standard convention of natural units such that $c = \hbar = 1$. Written in summation form this probability becomes,

$$
\begin{aligned}
P(\nu_\alpha \to \nu_\beta) = |\langle\nu_\beta|\nu_\alpha(t)\rangle|^2 = \\
|\sum_{i=1}^{n}\sum_{j=1}^{n} U^*_{\alpha i}U_{\beta j}e^{-i(E_i t)}\langle\nu_j|\nu_i\rangle|^2.
\end{aligned}
\tag{2.10}
$$

Here α and β represent the starting and resulting flavor states, and i and j represent the mass states. It is now useful to make two related approximations. Namely, that neutrinos have a small mass and are moving at relativistic speed. This allows for the approximations both

$$E_i = \sqrt{p_i^2 + m_i^2} \simeq p + \frac{m_i^2}{2E}, \tag{2.11}$$

and that $t \simeq L$ (in natural units of $c = 1$). L corresponds to the path length the neutrino has traveled, p its momentum, and m_i^2 its mass squared. Note that equation 2.11 has implicitly assumed that since neutrinos are relativistic $p_i \simeq p_j \equiv p \simeq E$. Under the assumption of light, relativistic neutrinos and evaluating the magnitude squared [3], we get

$$
\begin{aligned}
P(\nu_\alpha \to \nu_\beta) = \\
&\delta_{\alpha\beta} - 4\sum_{i<j}^{n} Re(U_{\alpha i}U^*_{\beta i}U^*_{\alpha j}U_{\beta j}) \sin^2\left(\frac{\Delta m^2_{ij}L}{4E}\right) \\
&\qquad + 2\sum_{i<j}^{n} Im(U_{\alpha i}U^*_{\beta i}U^*_{\alpha j}U_{\beta j}) \sin\left(\frac{\Delta m^2_{ij}L}{2E}\right), \quad (2.12)
\end{aligned}
$$

where Δm^2_{ij} corresponds to the difference between the squared-masses of mass state i and mass state j, not the squared difference between the mass states. Here we have leveraged a transformation of the $e^{-i\phi}$ terms via Euler's formula $e^{-i\phi} = \cos(\phi) - i\sin(\phi)$. The $\delta_{\alpha\beta}$ term represents the fact that the flavor states are orthogonal to one another. When $\alpha = \beta$ then $\delta_{\alpha\beta} = 1$, otherwise it is 0.

In this representation, it is now clear that given different mass states, oscillation from flavor state ν_α to ν_β becomes possible. This is witnessed via the Δm^2_{ij} term in the probability equation. It is important to note that experiments probing this fundamental parameter via oscillation studies are sensitive only to the magnitude of the difference in the squared neutrino masses, not the masses themselves, or even which state is larger. Two of the Δm^2_{ij} terms are measured, and it is known that $m_2 > m_1$ due to the observation of matter effects in the sun for neutrinos [24]. However, the sign of Δm_{31} is unknown and Δm_{32} is still an open question. In figure 2.3, the two potential neutrino mass hierarchies are depicted [2]. We note that at present the inverted mass hierarchy is disfavored at slightly more than 2σ [3].

Another important point to note is the probability's dependence on the L/E ratio, where L is the path length of the neutrino since its creation, and E is its energy.

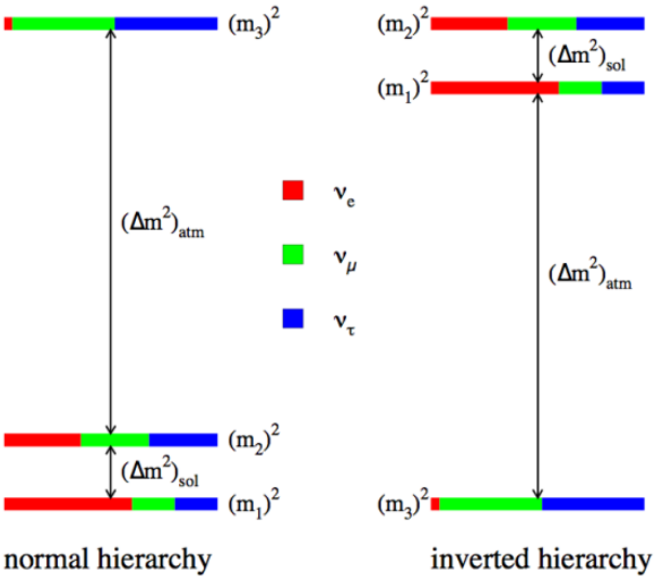


Figure 2.3: Pictorial representation of the possible neutrino mass hierarchies. Note Δm^2_{atm} is equivalent to Δm^2_{32} and Δm^2_{sol} is equivalent to Δm^2_{21}. [2]. For up-to-date makeups of the different flavor and mass states, see [3].

This influences the experimental design of oscillation studies as it determines where in phase space the experiment can probe the fundamental oscillation parameters contained within the PMNS matrix.

2.3.3 Neutrino Oscillation Experimental Limits

Now that we have described the formalism behind neutrino flavor oscillation, we will now turn to the parameters surrounding the mechanism, and the experimental limits placed upon them. Here it becomes beneficial to decompose the PMNS matrix in 2.5 into three rotation matrices,

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{i\delta} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.13}$$

Where c_{ij} and s_{ij} are common shorthands for $\cos(\theta_{ij})$ and $\sin(\theta_{ij})$. θ_{ij} represent the three mixing angles between the different mass states, θ_{12}, θ_{13}, and θ_{23}, and δ

represents a single complex which would induce CP-violation.[6] Multiplying these rotation matrices together we can better see the relation to the typical PMNS matrix,

$$\begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}. \tag{2.14}$$

The key subtlety of this change of framework is the absorption of seemingly 'free' parameters in the originally U framework. The original PMNS matrix is composed of nine parameters. Were it real it would only require three parameters to describe the rotations, but as the unitary matrix is not real, six additional phases are introduced. However, by defining different phases relative to one lepton family generation (typically the electron) then freely defining that phase to be 0, the parametrization shown in 2.14 is achieved. This essentially represents the process of absorbing these phases into the lepton fields and is described in detail in [25]. Now the oscillations can be described via three mixing angles and a complex term [19], alongside the two independent Δm^2_{21} and Δm^2_{32} terms relating the neutrino mass states.

With our parametrization complete, we can examine the different experimental limits set on these parameters via different oscillation experiments. Table 2.1 summarizes the different experiments and their types that have placed the dominant constraints on the different oscillation parameters. Here experiments can be categorized both by their neutrino source, and in some cases, by their baseline. Specifically, the neutrino sources are either solar neutrinos, neutrinos created from decays in nuclear reactors, atmospheric neutrinos, or neutrinos created via an accelerator.

In cases where the neutrino source is on earth, there exists some control over the length of the baseline that experiments set their detectors along. Within this subset of experiments, these dominating measurements are made at either long baselines

---

[6] CP-violation represents a break in charge conjugation parity symmetry, which is a symmetry representing that physics functions the same if you flip a particle to its antiparticle and its parity at the same time.

Table 2.1: Experiments contributing to the present determination of the oscillation parameters [3]

Experiment	Dominant	Important
Solar Experiments	θ_{12}	Δm^2_{21}, θ_{13}
Reactor LBL (KamLAND)	Δm^2_{21}	θ_{12}, θ_{13}
Reactor MBL (Daya-Bay, Reno, D-Chooz)	θ_{13}, $\lvert\Delta m^2_{31,32}\rvert$	
Atmospheric Experiments (SK, IC-DC)		θ_{23}, $\lvert\Delta m^2_{31,32}\rvert$, θ_{13}, δ_{CP}
Accel LBL ν_μ,$\bar{\nu}_\mu$, Disapp (K2K, MINOS, T2K, NOνA)	$\lvert\Delta m^2_{31,32}\rvert$, θ_{23}	
Accel LBL ν_e,$\bar{\nu}_e$, App (MINOS, T2K, NoνA)	δ_{CP}	θ_{13}, θ_{23}

(LBL) or medium baselines (MBL). Recall the the baseline is important due to the presence of L in the oscillation probability in equation 2.12.

We note that in favor of following the presently determined, most-likely path for the neutrino hierarchy, the limits proposed in this section assume a normal (not inverted) hierarchy of neutrino mass states. Further, the constraints mentioned below are taken from the Particle Data Group 2021 version of the 2020 review, [3], which provides values for several global fits.

Presently, SNO [26] and Super-Kamiokande [27] provide the dominant measurement of $\sin^2(\theta_{12}) \approx 0.31^{+0.013}_{-0.012}$ by way of searching for disappearance associated with the solar ν_e flux. Meanwhile, the LBL reactor experiment, KamLAND [28], has the dominant measurement of $\Delta m^2_{21} \approx 7.39^{+0.21}_{-0.20} \times 10^{-5}$ eV^2 [3]. The KamLAND experiment is stationed about 180 km from a series of nuclear reactors and measures the $\bar{\nu}_e$ survival rate from antineutrinos generated in nuclear reactions. Other reactor experiments, such as Daya-Bay [29], Reno [30], and Double-Chooz [31] constrain $\sin^2(\theta_{13}) \approx 2.241^{+0.066}_{-0.065} \times 10^{-2}$.

Long baseline accelerator-based neutrino beam experiments, such as K2K [32], MINOS, T2K, and NOνA [33], measuring disappearing ν_μ flux due to oscillation have made the dominant measurements for $\Delta m^2_{32} \approx 2.449^{+0.032}_{-0.030} \times 10^{-3}$ eV^2 as well as for $\sin^2(\theta_{23}) \approx 0.56^{+0.020}_{-0.033}$ [3].

Our last constraint is placed by long baseline accelerator-based neutrino experiments probing the CP violating term. These search for ν_e appearance, an increase in the ν_e flux relative to expectation based on oscillations from the ν_μ beam. Together, the MINOS, T2K, and NOνA collaborations [33] constrain $\delta_{CP} \approx 222^{+38}_{-28}{}^{\circ}$ [3].

Further measurements on neutrino oscillation parameters have been made using atmospheric neutrinos. Super-Kamiokande has provided limits on several neutrino oscillation parameters using atmospheric neutrino data [34]. These neutrinos are generated due to interactions between cosmic rays and the earth's atmosphere. As an example, a cosmic ray hitting nucleons within the atmosphere can create mesons, such as a π^+ (π^-) which then decays to a $\mu^+ + \nu_\mu$ ($\mu^- + \bar{\nu_\mu}$) providing a first source of neutrinos, followed by the muon's (anti-muon's) own decay processes:

$$\mu^- = e^- + \bar{\nu_e} + \nu_\mu \tag{2.15}$$

$$\mu^+ = e^+ + \nu_e + \bar{\nu_\mu} \tag{2.16}$$

which provide additional atmospheric ν_e and ν_μ flux.

Chapter 3

The MicroBooNE Experiment

The Micro Booster Neutrino Experiment (MicroBooNE) is based at Fermi National Accelerator Laboratory (Fermilab) just outside Batavia, Illinois. At Fermilab, the Short Baseline Neutrino (SBN) Program consists of three different detectors: MicroBooNE, the Short Baseline Near Detector (SBND) and the far detector, ICARUS. ICARUS was moved to Fermilab after operation at Gran Sasso Lab in Italy [4]. Each of these detectors are placed along the same Booster Neutrino Beamline (BNB). These three experiments all aim to study eV-scale neutrino interactions, as well as pioneer the Liquid Argon Time Projection Chamber (LArTPC) technology for use as a neutrino detector.

The MicroBooNE experiment, the first experiment operational on the BNB in the SBN program, is designed with two major goals in mind. The first and foremost is the study of short baseline neutrino oscillations. This goal is motivated primarily due to the observation of the low-energy-excess of electron-like events observed by MiniBooNE. MicroBooNE is also furthering the development of Liquid Argon Time Projection Chamber detector technology, which provides precise calorimetric information as well as excellent identification. This LArTPC technology is important for the future of neutrino physics, both through SBND, and the Deep Underground Neutrino Experiment (DUNE) [35] which will also feature LArTPC detectors. Through this information, MicroBooNE also is able to perform various cross section measurements [10].

Figure 3.1 shows the array of detectors located along the BNB at Fermilab. MicroBooNE can be seen sitting along the middle, right beside MiniBooNE.

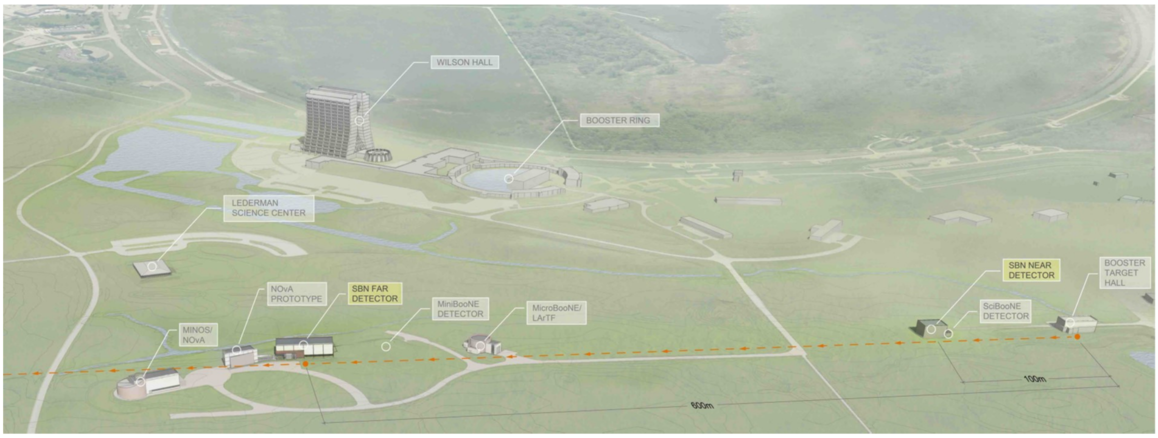


Figure 3.1: A diagram of the SBN Program at Fermilab, with various neutrino detectors including MicroBooNE's, placed along the neutrino beam. [4].

Within this chapter we will describe the MiniBooNE experiment's observed excess and the MicroBooNE experimental setup. Then we will describe a further extension to the SM via the introduction of a sterile neutrino, whose presence could explain the observed excess.

3.1 Origins in MiniBooNE

MiniBooNE or the Mini Booster Neutrino Experiment was created to test an excess of $\bar{\nu}_e$events observed by the Liquid Scintillator Neutrino Detector (LSND) at Los Alamos National Lab [16]. MiniBooNE was designed at the same L/E as LSND. This choice was made so that if neutrino oscillations along a $\bar{\nu}_\mu \to \bar{\nu}_e$ channel explain the LSND excess, then MiniBooNE would be sensitive to the same oscillation.

The MiniBooNE detector is a Liquid Cherenkov detector consisting of a 12.2 m diameter sphere filled with 818 tons of mineral oil. An array of 1520 photomultiplier tubes (PMTs), the bulk of which were reused from the LSND experiment, are used for light detection. Of these, most are placed looking for neutrino interaction detection. However, 240 are placed beyond an opaque barrier in a veto region at the edge of the sphere to detect incoming cosmic-ray muons and identify signals from them as background. This frees the other PMTs to search for Cherenkov and scintillation light coming from charged particles produced through neutrino interactions. This outer layer of PMTs also serves to detect through-going particles from a neutrino interaction. If light is detected outside the inner chamber, but coincident with the neutrino interaction, it indicates a particle passed through the opaque barrier, and the entire interaction cannot be reconstructed. These interactions can then be vetoed from an analysis.

A visual representation of the MiniBooNE Cherenkov detector is shown in figure 3.2. Here we see the detector in its enclosure, and a cross section showing the signal and veto region. Meanwhile figure 3.3 shows a photograph of the inside of the MiniBooNE detector.

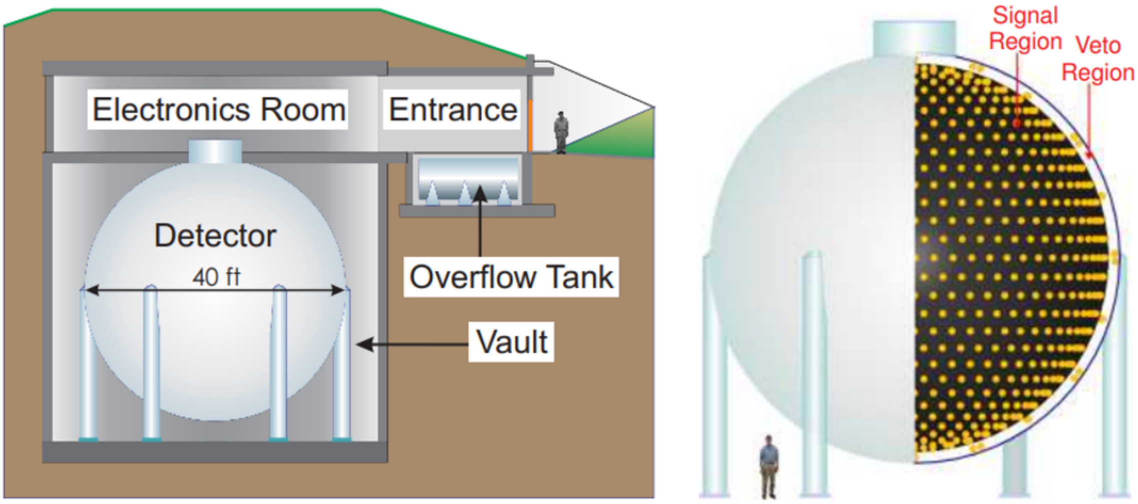


Figure 3.2: The MiniBooNE detector enclosure (left) and a cut-away drawing (right) of the detector showing the distribution of PMT's in the signal and veto regions [5].

The MiniBooNE collaboration performed a counting experiment to measure the number of charged-current quasi-elastic (CCQE) ν_e and $\bar{\nu}_e$ candidate events observed within the Cherenkov detector. In 2007 the collaboration released a result demonstrating an excess of these events observed in their data compared to their expectation specifically at low energies. As MiniBooNE has continued to take more data, this low-energy-excess (LEE) continues to be present. The most up-to-date measurement, released in 2021, is shown in figure 3.4.

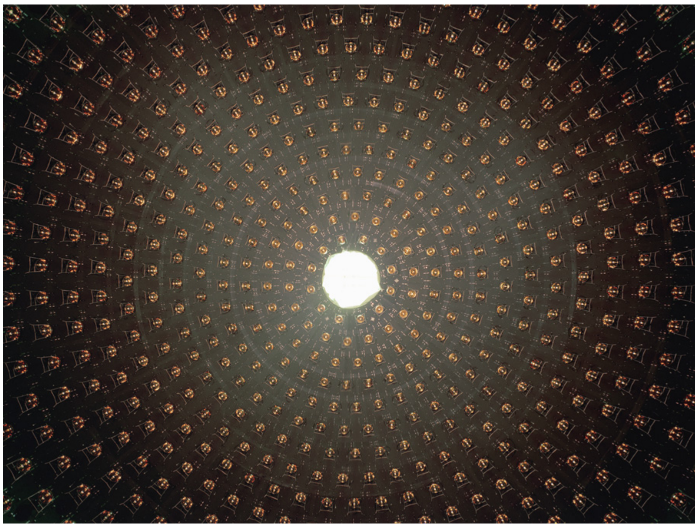


Figure 3.3: A photograph of the inside of the MiniBooNE detector [6].

The excesses observed by LSND and MiniBooNE have driven exploration into searches for a variety of new physics phenomenology, described further in [7]. It also exists as one of the major drivers for MicroBooNE and the SBN Program at Fermilab. This thesis specifically performs a search for a 3+1 Sterile Neutrino model, described in section 7, as investigated through a ν_μ CCQE selection.

3.2 The Booster Neutrino Beamline

The MicroBooNE LArTPC sits 468.5 m down the Booster Neutrino Beamline (BNB) at Fermilab. The Booster Neutrino Beamline is a relatively high-purity ν_μ and $\bar{\nu}_\mu$ beam with a small fraction of ν_e impurity, and is the same beam received by the MiniBooNE experiment 541 m down the beamline. In this section the beam discussion focuses on two aspects of the BNB.

1. The process by which the neutrinos are created and collimated via a proton beam.
2. A breakdown of the different components of the neutrino beam received by MicroBooNE.

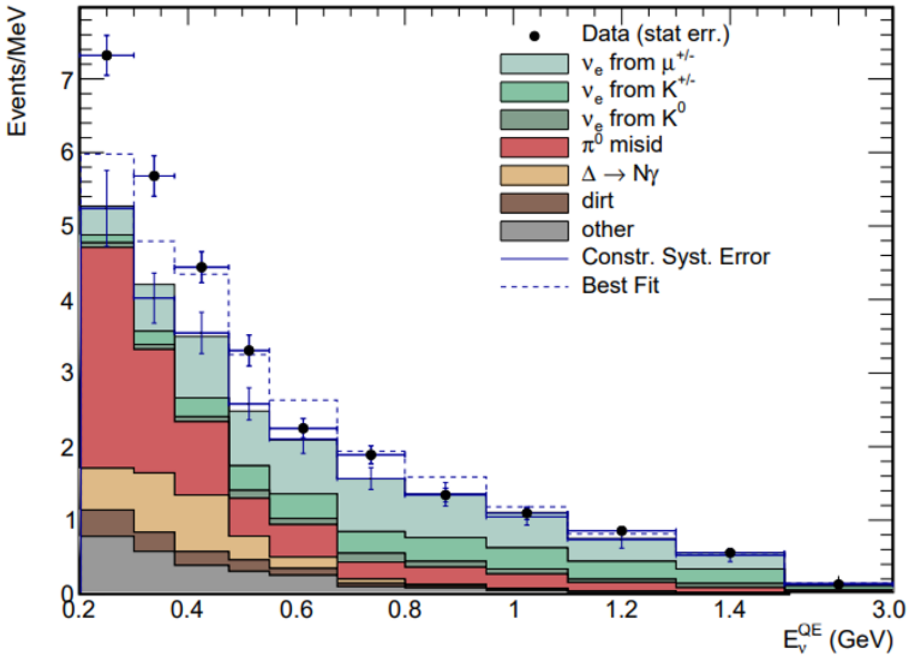


Figure 3.4: The observed MiniBooNE low-energy-excess in ν_e CCQE data. [7]

3.2.1 Making Neutrinos with a Proton Beam

At Fermilab, a linear accelerator (LINAC) accelerates bunches of protons to 400MeV before these protons are injected into a 474-meter-circumference synchrotron which further accelerates the protons to 8 GeV kinetic energy. The synchrotron then dumps out these bunches of protons in 'spills' that are roughly 2 ns wide and 19 ns apart.

These spills are then directed toward a beryllium target. Beryllium was chosen both for its ability to be replaced if necessary, and its ability to stay cool via an air-circulating cooling system while undergoing the high-intensity proton beam [8]. If the high energy protons smash into the target, they rip protons and neutrons apart into their constituent quarks. These quarks immediately recombine leading to hadron production. Different hadrons decay and produce neutrinos. These subsequent decay modes are described in table 3.1

While different kinds of hadrons can be produced, we focus now on the charged pions, $\pi^{\pm}$, produced. These $\pi^{\pm}$ and other hadrons (as well as any non-interacting protons from the beam) enter a pulsed toroidal electromagnet, known as the horn,

Table 3.1: Particle lifetimes, and neutrino-producing decay modes and branching ratios considered in the simulation[8].

Particle	Lifetime (ns)	Decay mode	Branching ratio (%)
π^+	26.03	$\mu^+ + \nu_\mu$	99.9877
		$e^+ + \nu_e$	0.0123
K^+	12.385	$\mu^+ + \nu_\mu$	63.44
		$\pi^0 + e^+ + \nu_e$	4.98
		$\pi^0 + \mu^+ + \nu_\mu$	3.32
K_L^0	51.6	$\pi^- + e^+ + \nu_e$	20.333
		$\pi^+ + e^- + \bar{\nu}_e$	20.197
		$\pi^- + \mu^+ + \nu_\mu$	13.551
		$\pi^+ + \mu^- + \bar{\nu}_\mu$	13.469
μ^+	2197.03	$e^+ + \nu_e + \bar{\nu}_\mu$	100.0

which is depicted in figure 3.5. To maintain function the horn is water-cooled during operation. The horn can operate in two modes, neutrino-mode and antineutrino-mode. The difference between modes is that antineutrino-mode directs current in the reverse direct, which flips the magnetic field.

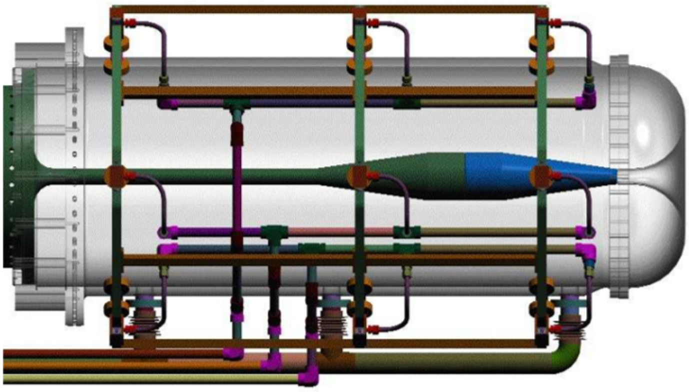


Figure 3.5: The MiniBooNE pulsed horn system. The outer conductor (gray) is transparent to show the inner conductor components running along the center (dark green and blue). The target assembly is inserted into the inner conductor from the left side. In neutrino-focusing mode, the (positive) current flows from left-to-right along the inner conductor, returning along the outer conductor. The plumbing associated with the water cooling system is also shown [8].

As charged hadrons pass through this focusing magnetic field they are turned either back inward toward the beam, or outward based on the sign of their electric charge. This allows the horn to selectively focus pions toward an opening leading into a decay pipe. π^+ or π^- can be selected based on the direction of the horn current which flips the magnetic field direction, meanwhile particles of the incorrect charge are directed away from the beam, and into shielding.

In neutrino-mode π^+ particles are then selected and sent through a collimator that absorbs particles that will not contribute to the neutrino beam. Then the π^+ enter the decay pipe, the air-filled, ~50 m long decay pipe. These π^+ mesons rapidly decay and create neutrinos, through processes outlined in table 3.1.

The muons from these decays are stopped by the earth, while the neutrino products makeup the beamline. In the context of neutrino oscillations in section 2.3, the start of a neutrino's path length begins where its parent meson decayed in the decay pipe.

A visual representation of the neutrino beam creation centered around the horn is depicted in figure 3.6.

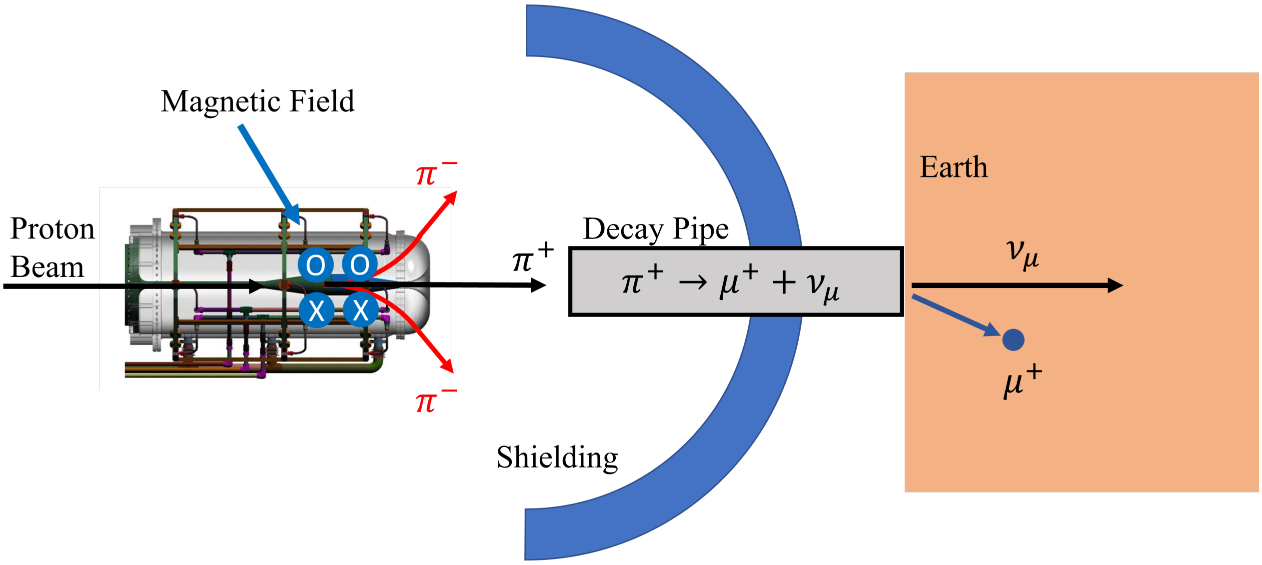


Figure 3.6: A cartoon of neutrino beam creation for the BNB. A proton beam instigates hadron production at the target, the horn focuses the desired charged pions, which then decay into neutrinos, while oppositely charged pions get sent away.

3.2.2 Neutrino Beam Composition

The various decay processes outlined in table 3.1 produce both electron neutrinos and muon neutrinos as well as their antiparticle counterparts even when running in

π^+ neutrino-mode. The BNB flux received by the MicroBooNE LArTPC is determined by the prediction determined by the MiniBooNE collaboration which uses GEANT4 to simulate the horn, decay pipe, and resulting beam [8]. The MicroBooNE collaboration takes this same simulation, and modifies the detector parameters to apply to the setup of the LArTPC [9].

The resulting neutrino flux is broken down by energy as well as flavor in figure 3.7. The bend in the ν_μ distribution at $E > 2.5$ GeV occurs due to contributions from kaon decay. The ν_e flux shown is considered *intrinsic* as it arises from the beam's generation. This is compared to electron neutrinos that may arise somewhere along the beam due to neutrino oscillations. The intrinsic ν_e contamination makes up roughly 0.5% of the total flux, while $\bar{\nu}_\mu$ contributes ~5.9% and muon neutrinos dominate with ~93.6% [8].

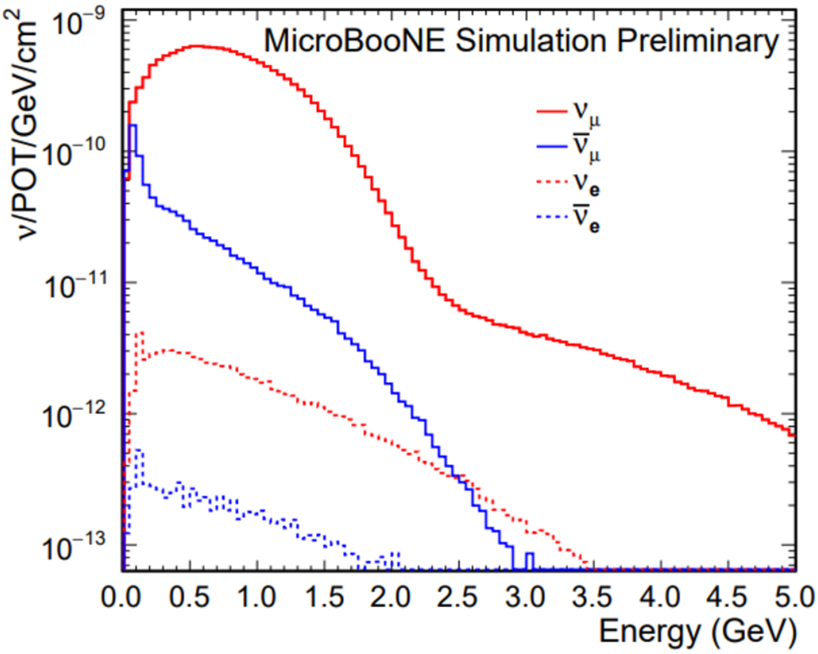


Figure 3.7: The absolute neutrino flux prediction through the MicroBooNE detector as calculated by the beam simulation. Shown is the flux for ν_μ, $\bar{\nu}_\mu$, ν_e, and $\bar{\nu}_e$averaged through the TPC volume with dimensions 2.56m×2.33m×10.37m. [9]

3.3 The MicroBooNE LArTPC

MicroBooNE's detector is a Liquid Argon Time Projection Chamber (LArTPC). LArTPC technology is advantageous because it can provide excellent spatial resolution to tell you where an interaction took place, as well as calorimetry to determine the charge deposition in the detector. Both are used for particle identification within an interaction. The MicroBooNE LArTPC makes use of 170 tonnes of liquid argon, held within a cryostat. Of the 170 tonnes, 90 tonnes are held within the TPC. An array of 32 photomultiplier tubes (PMTs) and lightguide paddles is placed behind the anode wire-planes described below [10].

3.3.1 LArTPC Setup

The Time Projection Chamber within the LArTPC consists of a box, 2.325 m tall, 2.560 m wide, and 10.368 m deep along the BNB direction. Facing along the beam direction, with the y-axis pointing up to the sky, on the right side of the box is the cathode plane held at -70 kV. On the left side of the LArTPC, within a wire frame, exist three anode wire-planes, held near ground, at progressively higher potentials as they get further from the cathode. In order of their progressively higher potentials, these anode wire-planes are called the U induction plane, the V induction plane, and the Y collection plane. Figure 3.8 shows a cross section of the detector.

When charged particles move through liquid argon they ionize electrons as well as make prompt vacuum ultraviolet (VUV) scintillation photons, which are detected via the light collection system. Due to the purity of the argon, the ionized electrons are now free to drift across the detector toward the higher-potential anode wire-planes. There, the electrons induce bipolar signals on the two induction planes before they are collected on the collection plane and create a unipolar pulse.

This process is laid out in the diagram shown in figure 3.9. The figure shows an example of an incoming neutrino interacting to create two charged particles that move through the TPC and create electrons, which then drift to the wire-planes creating signals. The V wire-plane image shown demonstrates the bipolar pulses

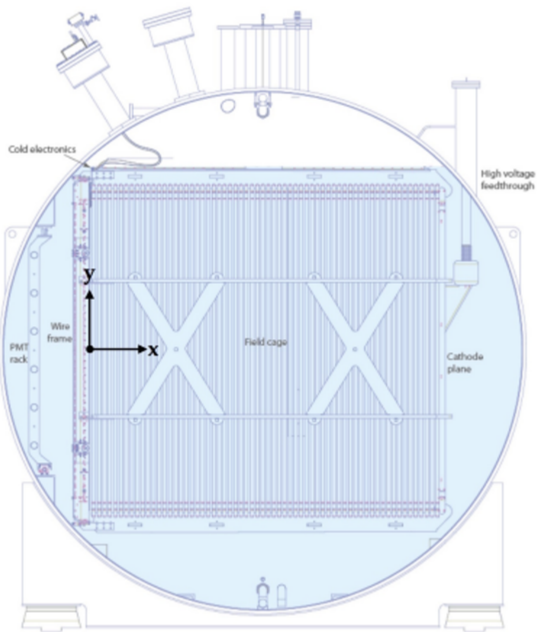


Figure 3.8: Schematic of the cross section of the MicroBooNE LArTPC. In this view, the beam would be directed out of the page (in the z direction). [10].

created, while the Y wire-plane waveforms show only unipolar pulses. These wire-planes are sampled at a rate of 2 MHz.

Table 3.2 shows the nominal values of several key design parameters. A point to note is that the collection plane consists of vertical wires spaced out along the beam direction, while the other wire-planes are held at a fixed angle of 60° relative to the vertical. Since the induction planes are at an angle there are fewer wires needed to span the same area.

Within MicroBooNE, when discussing aspects of our reconstruction and analyses and referring to our coordinate system, the z axis is defined as pointing in the direction of the beam, with 0 at the beginning of the LArTPC. The x axis is defined as pointing across the detector toward the cathode, with 0 at the anode wire-planes. Finally, the y axis points up, as required by the right-hand-rule, with 0 defined as in the middle of the detector. This Cartesian coordinate layout, as well as the analogous cylindrical coordinate version, is depicted in figure 3.10.

Given this coordinate system, the signals received by the wire-planes can then determine the y and z coordinates pertaining to an interaction that caused ionization charge to drift across the LArTPC. The x coordinate can then be determined using

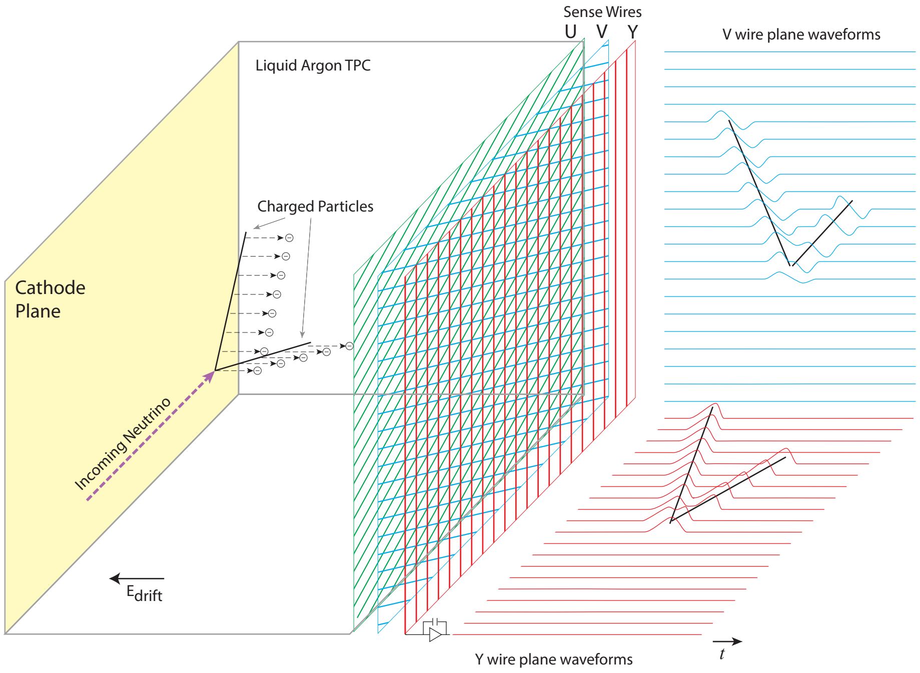


Figure 3.9: A diagram of the LArTPC principle. The signal formation for the second induction plane (V plane) and the collection plane (Y plane) are shown [10].

the drift velocity, which is set by the uniform electric field at $0.11 cm/\mu s$, and the total drift time, which is determined using the scintillation light detected via the light collection system.

3.3.2 PMT Light Collection System

The MicroBooNE light collection system is comprised of 32 Hamamatsu R5912-02mod 14 stage cryogenic 8-inch hemi-spherical PMTs. This high-gain model is chosen to counteract the decrease in gain experienced by operating in a cryogenic environment. In addition to this PMT array, four light guide paddles are installed for *R&D* purposes [36]. Figure 3.11 shows a diagram of the light collection system as it is arrayed within the MicroBooNE LArTPC.

Argon scintillates isotropically at 128 nm, a difficult wavelength to detect using photodetectors. As such both the light paddles, and the PMTs are coated in tetraphenyl-butadiene (TPB), which absorbs the ultraviolet light and emits in the

Table 3.2: MicroBooNE LArTPC design parameters and nominal operating conditions. [10]

Parameter	Value
# Anode planes	3
Anode planes spacing	3 mm
Wire pitch	3 mm
Wire type	SSm diam. 150 μm
Wire coating	2μm Cu, 0.1μm Ag
Design Wire tension	6.9N ±1.0N
# wires (total)	8256
# U Induction plane wires	2400
# V Induction plane wires	2400
# Y Collection plane wires	3456
Wire orientation (w.r.t. vertical)	+60°,−60°,0° (U,V,Y)
Cathode voltage (nominal)	−128 kV
Cathode voltage (operation)	−70 kV
Bias voltages (U,V,Y)	−200 V, 0 V, +440 V
Drift-field (nominal)	500 V/cm
Drift-field (operation)	273.4375 V/cm
Max. Drift Time, Cathode to U (at 500 V/cm)	1.6 ms
Max. Drift Time, Cathode to U (at 273 V/cm)	2.9 ms
# Field-cage steps	64
Ring-to-ring voltage step	2.0 kV

wavelengths of light that pass through borosilicate glass and activate the photocathode material of the PMTs. A thin coating of platinum is applied to increase the photocathode's conductivity at cryogenic temperatures [10]. The emission and absorption spectra are shown in figure 3.12.

As mentioned previously, the detection of light by the light collection system is important in determining the x coordinate for determining where in 3D an interaction took place.

3.3.3 Event Triggering and Electronics Readout

While the MicroBooNE LArTPC is active, the electronics systems are not constantly reading out data for storage. This is a consequence of two major factors. First and foremost, it would be impractical to try to readout and store all the data taken by the LArTPC during its years-long operation at MHz sampling rates, and secondly, during the vast majority of time the BNB is operational, no neutrinos are

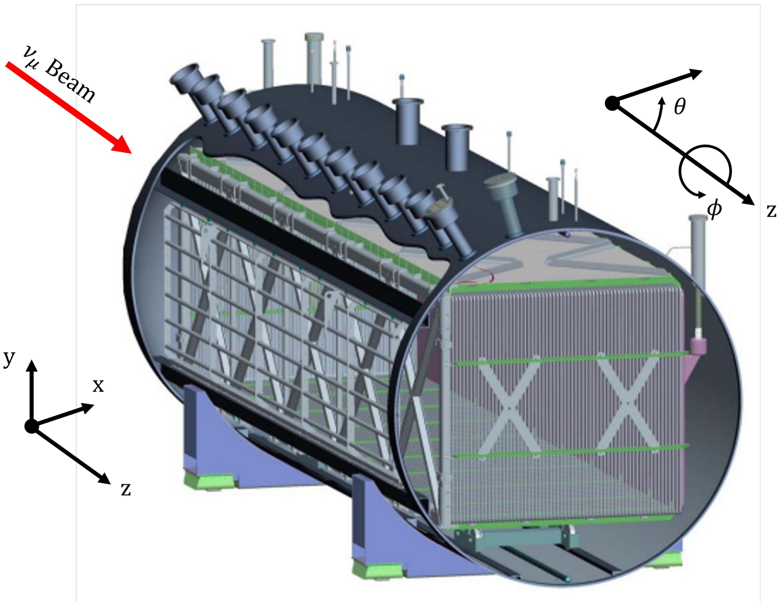


Figure 3.10: A depiction of the coordinate systems defined by MicroBooNE when discussing locations within the LArTPC.

being received by the detector. This creates the need for 'trigger' mechanisms that signal that a period of time or 'event' surrounding an interaction should be saved for analysis.

One such trigger comes directly from the BNB. Spills of neutrinos are sent through the BNB to the MicroBooNE LArTPC at roughly 5 Hz, with a spill width of 1.6 μs [37]. As such, much of the time the BNB is 'on' no neutrino interactions are expected, as the gaps between spills makeup the vast majority of time. The BNB trigger is designed to save the 1.6 ms starting at the spill, as well as 1.6 ms before and after the spill, for a total of 4.8 ms comprising an 'event'.

However, as neutrinos are extremely non-interactive, most of these events will be devoid of a neutrino interaction, and contain only cosmic-ray muon interaction background occurring during this time. In order to avoid saving all these cosmic-ray-only background events, an additional requirement is made that the PMTs receive a total of 6.5 photoelectrons during the 1.6 μs beam spill window. This fractional amount of photoelectrons is possible as the PMTs are calibrated to convert an electrical signal to photoelectrons. This should represent the scintillation light received from particles coming from a neutrino interaction, and removes 97% of empty beam

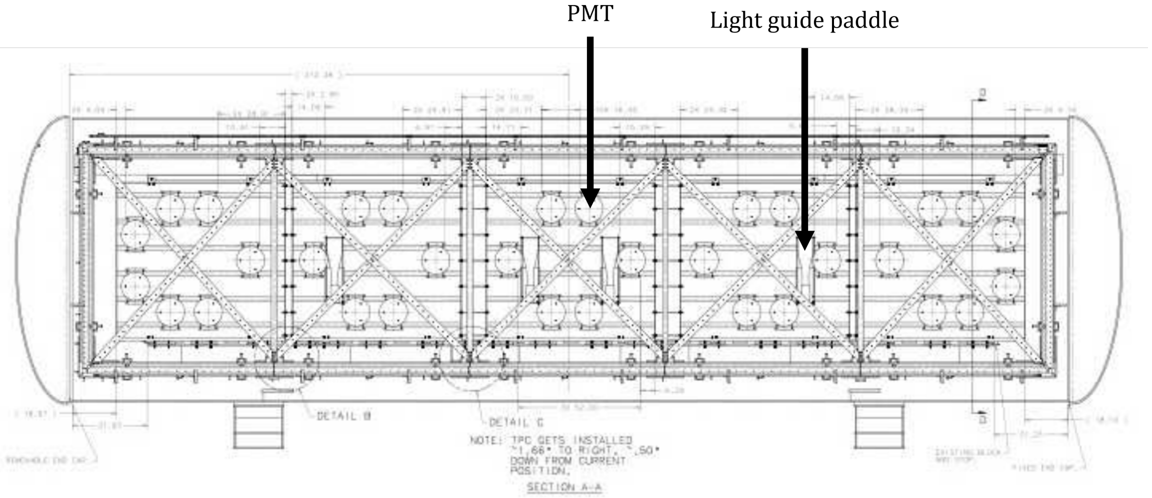


Figure 3.11: A schematic of the light collection system as arrayed within the MicroBooNE LArTPC, the 32 PMTs are shown as circles, with the 4 light guide paddles as rectangles.

spills [38].

In order to study the cosmic-ray muon background, data is taken while the beam is off, but the optical trigger's conditions are still met. These off-beam events, discussed further in section 6.1, are useful as they should be functionally the same as an event taken during beam operation without a neutrino interaction that is saved due to cosmic-ray muons passing the optical trigger.

Initially the signals from the wire-planes and PMTs are analog. MicroBooNE's LArTPC uses a custom low-noise electronic readout system to amplify these signals, then digitize them before they are written to disk. This process is outlined in figure 3.13, and described in detail in [10], and summarized briefly below.

The signals read out by the LArTPC's wire-planes are first sent to a CMOS application-specific integrated circuit (ASIC) electronics located within the cryostat surrounding the LArTPC. These ASICs need to be located within the cold environment so that the wire-signals are not lost to noise by traveling over long distances. This allows the signals to be amplified, before they are sent out of the cryostat to readout crates in the detector hall. Finally the signals are sent through analog-digital-converters (ADCs) which digitize the signals, then into front-end-modules (FEMs) which feed into a buffer. From there, based on the triggers described above,

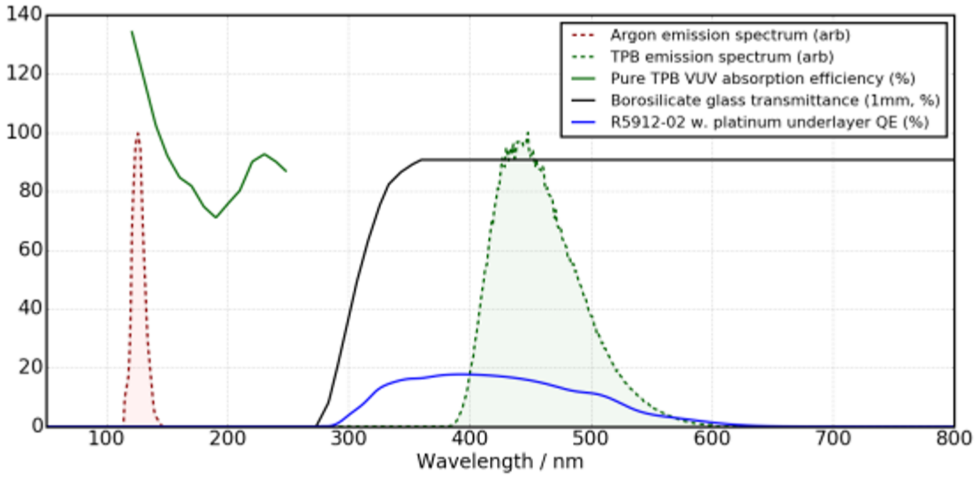


Figure 3.12: The scintillation spectra of Argon (red-dashed) along with the TPB re-emission spectra (green-dashed) are shown with the absorption spectra in % of TPB (green-solid), the borosilicate glass (black-solid), and platinum (blue-solid) components of the PMT arrays [11].

either the data is stored, or overwritten.[1]

3.3.4 The MicroBooNE CRT

MicroBooNE is a surface-level detector and as such cosmic-ray muons are one of the major backgrounds to study neutrino interactions in the LArTPC. These cosmic-ray muons slice through the detector at a rate of 5 kHz [10]. In order to counteract this MicroBooNE designed and constructed an external system to identify and reject cosmic-ray muon interactions within the data. The design and construction of this Cosmic Ray Tagger (CRT) is covered in detail in [39].

The construction and implementation of the CRT was completed for MicroBooNE's Run 3 (Run periods are defined in section 6.1). This means that the CRT was not functional for the bulk of the data available for use in this analysis. In order to avoid dependence on the CRT for cosmic tagging then not having it for a significant portion of the data, it was not used by the DL reconstruction, or in this thesis. Instead, alternative cosmic-ray muon tagging methods were used, as

---

[1] A second stream of data, known as the supernova data stream exists in parallel, for physics unrelated to the BNB and is not used in this analysis.

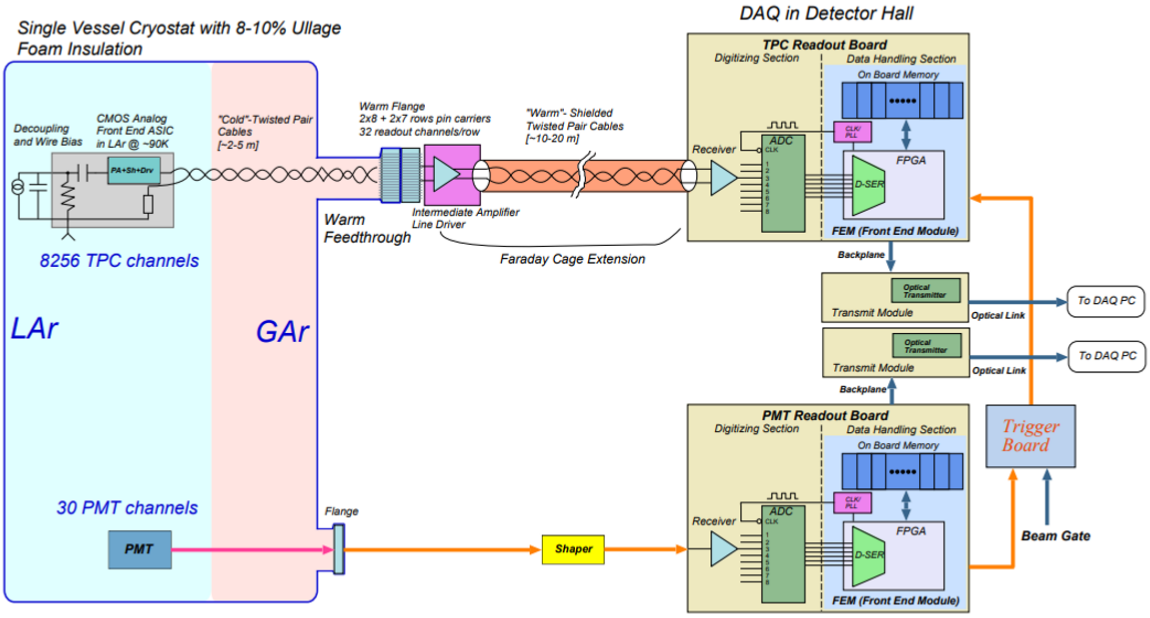


Figure 3.13: MicroBooNE LArTPC and PMT signal processing and readout stages [10].

Table 3.3: The wire counts, and unresponsive wire counts for the three different wire-planes in the MicroBooNE LArTPC.

Plane	Unresponsive Wires	Total Wires	% Unresponsive
U Induction Plane	427	2400	17.79
V Induction Plane	98	2400	4.08
Y Collection Plane	345	3456	9.98
Total	870	8256	10.54

described in section 5.3. In addition, a DL cosmic-ray muon tagging tool is explored in section 8 at length, which was primarily the work of this thesis' author.

3.3.5 Unresponsive Wire Regions

Portions of the TPC wires are unresponsive in MicroBooNE's detector. In the images we build to represent the wire-plane signals, discussed later, these unresponsive regions manifest as vertical lines devoid of charge deposition, noticeable in the event displays shown in section 5.

The rates at which this unresponsive wire problem occurs are described in table 3.3.

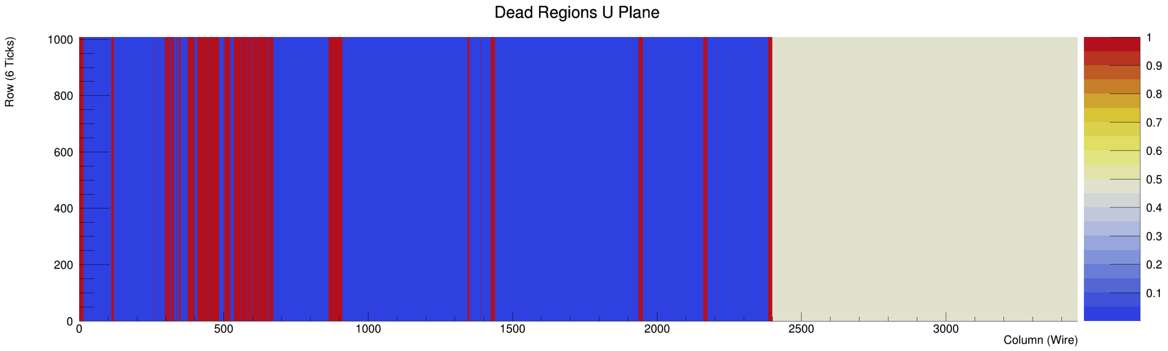
Unresponsive wires can come about from a multitude of causes, and are discussed in more detail in [40]. These causes involve either wires being shorted or issues with the electronics surrounding the application-specific integrated circuits (ASICs). However, because the MicroBooNE LArTPC is welded shut, access to the wires is not possible, and replacement was not a viable option.

A visual representation of the unresponsive regions in our three different wire-plane signal images is shown in figure 3.14. The process with which these images relate to the detector is detailed in the image reconstruction section, 5.2, for the U and V induction planes, which only contain 2400 wires, columns > 2400 are shown in grey-white.

Notably, the U induction plane features the most unresponsive regions, many of which are clustered together in the low wire index portion of the image, which corresponds to the upstream part of the detector with respect to the beam. Meanwhile, the V induction plane is the cleanest, with only about 4% of its wires unresponsive. The Y collection plane strikes a middle-ground, with a chunk of unresponsive wires together just below index 2500, but otherwise spread out among the image. However, it is important to recall that having 3 wire-planes in the LArTPC design affords a layer of redundancy in reconstructing the 3D position of an interaction, as in an ideal case only two wire-planes are needed to give the y and z spatial coordinates. This redundancy means that only $\sim 1\%$ of 3D detector is lost despite $\sim 10\%$ of all the wires being unresponsive.

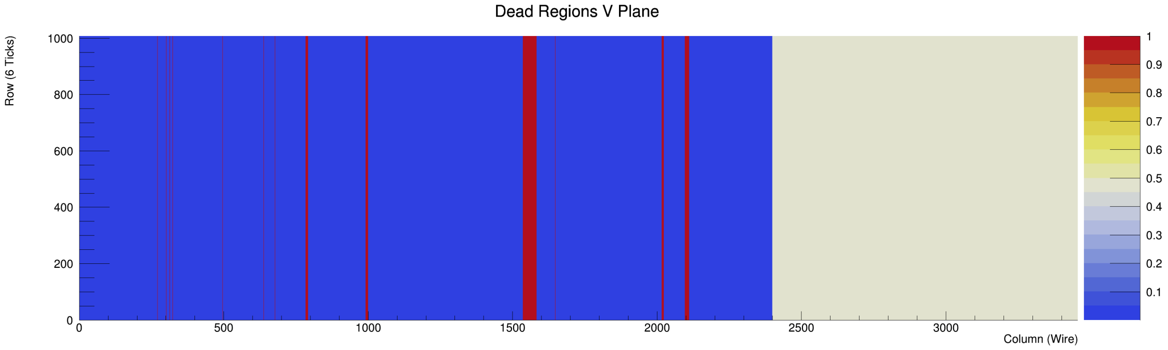
These unresponsive regions within the MicroBooNE detector are an obstacle to track and shower reconstruction. They interrupt the linear and conical topologies with artificial gaps lacking any deposited charge in our images. As such, either our algorithmic solutions need to be robust enough to anticipate unresponsive regions, or we could turn to machine learning tools which can learn to deal with the unresponsive regions from their training data.

For example, section 8 outlines the usage of a neural network, sMask-RCNN, which clusters cosmic-ray muon tracks, often across unresponsive regions, without needing specific programming to do so. Another project in MicroBooNE aimed



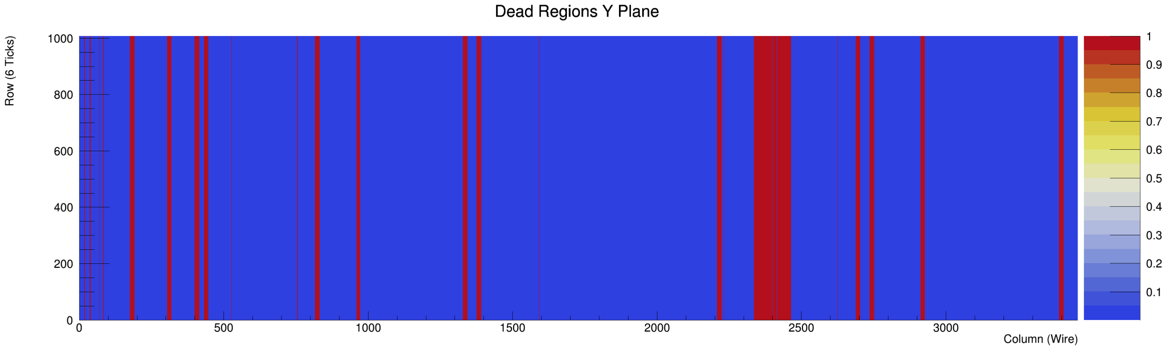
Dead Regions U Plane

(a) U Induction Plane



Dead Regions V Plane

(b) V Induction Plane



Dead Regions Y Plane

(c) Y Collection Plane

Figure 3.14: The unresponsive regions of the MicroBooNE wire-plane images for all three wire-planes are shown in red against the working regions in blue. Columns > 2400 for the U and V planes are shown in grey-white because those planes only feature 2400 wires. The construction of these images are defined in section 5.2

to regenerate the signal that would have been deposited in a unresponsive wires using a Generative Adversarial Network [41], which could run at the beginning of reconstruction, and if effective, would drastically mitigate the issue of unresponsive wires.

3.4 Sterile Neutrinos

Now that we have outlined the MiniBooNE low-energy-excess, and how it motivated the MicroBooNE experiment's design we can explore a solution to the anomalous LEE. In this section we explore a theory-driven explanation to the excess itself.

One way to account for the MB LEE, as well as other experimental phenomena, is to add an additional neutrino mass state to our model of neutrino oscillation, effectively adding an additional mass splitting, Δm^2_{41}. However adding an additional flavor state is slightly more complicated. The LEP experiment has provided evidence that there are only 3 stable, weakly interacting neutrinos [42]. Therefore the introduction of a new flavor state (and new mass state) is of a neutrino that does not interact via the weak force. Namely, it doesn't couple with the $W^{\pm}$ or the Z^0 bosons.

This type of neutrino is termed a sterile neutrino. Sterile here is used in the sense that it interacts only via gravity. But of course gravity does not provide a reasonable handle to test for the sterile neutrino's existence, so a sterile neutrino is functionally non-interactive. This presents a problem. Neutrinos were already famously difficult to observe, with billions and billions passing through everything on earth constantly. Even when we find them, we only are able to observe them from the products of their weak interactions. Sterile neutrinos, by definition, would remove the weak interaction from our list of tools to detect their existence.

The handle that remains lies in neutrino oscillations. We can look for unexplained disappearance effects, where we see fewer neutrinos than the SM expects. We can also search for unexplained appearance effects, where we have an excess of neutrinos compared to the SM prediction. In either case, the mechanism could be explained by a sterile mass state impacting our oscillations formalism.

3.4.1 3+1 Sterile Neutrino Model

There are an infinite number of ways to further push the already extended Standard Model described in 2.3. Many involving sterile neutrinos are discussed in terms of global fits in [43].

Within this thesis we investigate a 3+1 Sterile Neutrino model, where the 3 observed flavors of neutrino in the SM are joined by a single additional sterile neutrino state. We dub the flavor state ν_S and the mass state ν_4. This sterile then mixes with the other neutrino flavor states in the same way as they mix with each other. However due to constraints already put on the neutrino oscillation mixing parameters described in section 2.3.3, the sterile flavor state is only allowed minimal mixing with the first three mass states.

The addition of another mass state to the neutrino oscillation formalism described in section 2.3.2 involves extending the PMNS matrix to a 4×4 matrix, and following the analogous procedure outlined for three neutrinos. However, for MicroBooNE, as shown below, an approximation can be made.

The anomalies observed by MiniBooNE, LSND, and other short-baseline neutrino experiments push the added mass splitting to order of 10 times larger than the presently measured mass splittings, Δm_{21}, and Δm_{32} [43], with the sterile neutrino mass state sitting roughly around 1 eV [44, 45].

This significant difference between the proposed sterile-state mass splitting, and the existing mass splittings allow for a commonly used "short-baseline approximation" to be made when talking about sterile neutrino oscillations. The short-baseline approximation sets

$$\Delta m_{21}^2 \approx \Delta m_{32}^2 \approx 0. \tag{3.1}$$

This simplification then allows sterile neutrino oscillations to be discussed in a framework surrounding Δm_{41}^2, which is effectively the mass splitting between the newly introduced sterile neutrino's mass state, and the SM neutrino mass states. As shown in section 2, the neutrino oscillation probability is dependent on the path length L, and the neutrino energy E.[2] As such, the short-baseline approximation can be made for experiments that feature similar neutrino path lengths. As an example, the MicroBooNE detector sits next door to MiniBooNE's detector.

By setting the other mass splittings to effectively 0 under the short-baseline approximation the neutrino oscillation procedure now appears as the 2-neutrino case dependent on the desired flavor, and the sterile mass splitting Δm^2_{41}. For neutrino survival, the probability becomes

$$P(\nu_\alpha \to \nu_\alpha) = 1 - 4(1 - |U_{\alpha 4}|^2)|U_{\alpha 4}|^2 \sin^2(1.27\Delta m^2_{41} L/E), \tag{3.2}$$

and for neutrino appearance probability in ν_β from ν_α

$$P(\nu_\alpha \to \nu_\beta) = 4|U_{\alpha 4}|^2|U_{\beta 4}|^2 \sin^2(1.27\Delta m^2_{41} L/E). \tag{3.3}$$

In both of these equations, the factor 1.27 comes from a factor of $\frac{1}{4\hbar c}$ in units to allow for the inputs of L to be in km, E to be in GeV, and Δm^2 to be in eV^2. For an example ν_e appearance search, the observation is compared to the expectation, modified by the appearance probability in equation 3.3, checking to see if the fourth mass state allows additional ν_e flux via a connection to the new mass state.

The neutrino survival probability is useful in performing a neutrino disappearance search, wherein one compares the observation to their expectation, modified by the survival probability. This probes whether ν_α neutrinos are disappearing into other flavors via oscillations with the new mass state. In section 7, we perform a ν_μ disappearance search in MicroBooNE across a grid of oscillation parameters.

---

[2] As well, of course, as on the fundamental parameters of the universe, like the mass splittings, and the mixing angles making up the PMNS matrix. But those are not controlled in experiment.

3.5 Blinding Policy

As a brief aside, it is important to mention that MicroBooNE, as part of a collaboration-wide agreement, blinds itself to the signal-region data in its various analyses. Functionally, this means that data is taken by the detector, and stored for eventual use. However, due to this blinding-agreement, members of the collaboration agree to not run their analysis on data pertaining to their signal until such a time as the analysis is fully-formed, and frozen. Frozen here means that the analysis infrastructure will no longer undergo any major changes unless significant errors are discovered. Then fixes are implemented and explicitly documented for transparency.

This blindness policy is implemented to help avoid analyses unintentionally biasing themselves to some expectation of what the data 'should' be, where 'should' is defined by whatever preconceptions the analyzer may have. For example, it is designed to avoid a case where some new physics is present in the data, causing significant tension with expectation, then the analyzer mistakenly tuning the expectation to fit better. This could happen if the analyzer incorrectly assumes the tension is due to a fault in the analysis and not an underlying physical phenomenon.

In practice, as analyses are developed sidebands are examined. These sidebands refer to regions of data where the analysis' signal is not expected, and agreement between expectation and data is anticipated. This allows an analysis to test its machinery without unblinding itself to the signal-region data.

Chapter 4

Deep Learning

Within this chapter context is provided surrounding the moniker ”Deep Learning” used in the ”MicroBooNE Deep Learning Analysis”. We outline several techniques discussed throughout this thesis, most notably Convolutional Neural Networks, and Boosted Decision Trees which are used in our reconstruction and selection of data in sections 5 and 6. Additional exploration into using DL-methods in cosmic tagging is performed in section 8, a significant effort into potential future tools in LArTPC experiments.

The field of machine learning and artificial intelligence is broad, and rapidly expanding. This chapter will focus on topics relevant to those employed in the DL analysis or otherwise used within this thesis.

4.1 Machine Learning Overview

Deep learning exists as a sub-field of machine learning. As a whole, machine learning is a broad field of computer science concerned with developing algorithms that are able to improve or ‘learn’ automatically through experience with data. Considered part of the study of artificial intelligence, machine learning algorithms are distinct from more traditional computer algorithms in that they do not contain an explicitly programmed method to make their predictions. Instead, machine learning algorithms can learn their solutions by ‘training’ against a target solution. This is known as supervised learning. Unsupervised learning is a branch of machine learning that does not require the data to be labeled with a target ahead of time, instead forcing the algorithm to group the data itself. Unsupervised learning is not used within the MicroBooNE DL analysis, and is beyond the scope of this thesis.

Since supervised machine learning requires its algorithms to train with a known target, labeled data is needed in order for a machine learning algorithm to learn its task. In general, assembling a large dataset of training data can require significant time and person-hours to hand-label the data. However, in the field of high energy particle physics simulated datasets are fairly common, and often produced for more traditional reconstruction. This removes a major burden in exploring the use of machine learning tools for particle physicists.

4.2 Deep Learning

Deep learning exists as a methodology underneath the umbrella of machine learning [46]. Deep learning techniques attempt to solve problems by imitating neural networks found in the brain. These artificial neural networks are designed to follow the existing pattern in chains of neurons linked together that allow signal transmission. Each artificial neuron, or node, exists in parallel with many neurons to form a layer. Then neurons within these layers connect to the next layer of neurons and so on. These connections allow information to flow forward through the layers of the network. As layer upon layer of nodes are built up the information flow and connectivity deepens, hence the deep learning.

Figure 4.1 shows an example setup of an artificial neuron or node. Node N receives a vector of inputs X_i each with a corresponding weight w_i. The node sends an output based on the linear combination of inputs. While the image depicts only one branch of outputs for simplicity, it is important to note that this output can be used as an input for many neurons in the next layer of the network. Equation 4.1 provides a mathematical depiction of an artificial neuron. The sum of a linear combination of the inputs X_i multiplied by their weights w_i is fed through some activation function ϕ to determine the node's output O.

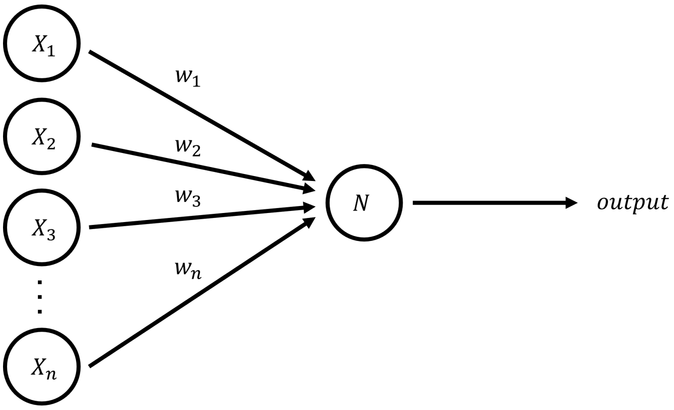


Figure 4.1: An example of an artificial neuron and its inputs.

$$O = \phi(\sum_{i=0}^{n} X_i w_i) \tag{4.1}$$

The activation function ϕ is meant to imitate a neuron's ability to output a signal. While not quite a binary switch, the activation function is typically designed to receive input up to some threshold before it activates like a neuron firing, this provides non-linearity to the neural network which is essential in decision making processes which are frequently non-linear. Figure 4.2 depicts three common activation functions as well as the identity $y = x$. One notable aspect concerning these functions is that they all can be characterized as having a non-firing range of output, then transitioning into some 'ON' state.

The leaky RELU or leaky rectified linear unit function is essentially a piece-wise function where $y = x$ above $x = 0$. Below zero a much smaller slope is used, for example $y = 0.01x$. This allows the simulation of a neuron 'firing' around the $x = 0$ point. This is slightly modified from the original rectified linear unit function which defined to be 0 for all $x < 0$. This additional slope characterizes the 'leaky' aspect, and allows for later neurons to differentiate between different levels of negative input.

These activation functions allow the artificial neurons to 'fire'. Neurons then learn to fire when they spot a feature. This firing announces the feature's presence to later portions of the network. For example, a neuron deep in the network may be trained to fire if it sees a dog. This neuron would likely connect to earlier neurons

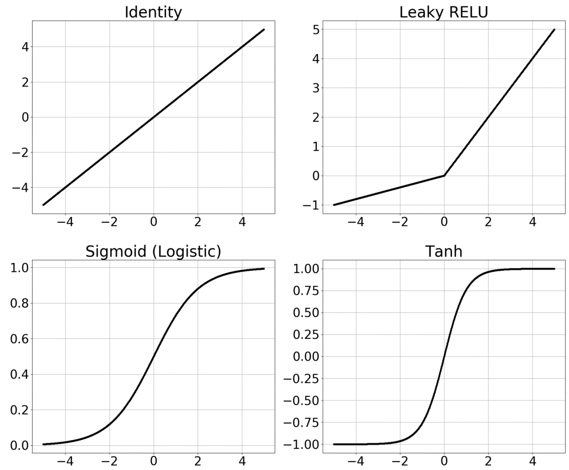


Figure 4.2: Several examples of activation functions.

identifying lower level features such as tail, fur, four legs and more. Then if these input features were present the dog neuron could activate.

At the end of the network the outputs need to be collected and turned into something meaningful. Keeping with the example of a classifier, the network may need to take all the final neurons, containing high-level features of the image, and classify whether the input is of one of N class categories (like dog, cat, horse, person etc). This can be done by taking all the final neurons and connecting them to N output 'scores', each relating to one of the potential classes. These scores can be run through a softmax function

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}, \tag{4.2}$$

where the softmax function takes N real numbers contained in $\vec{z}$ and creates a normalized probability distribution with a probability for each of the N inputs. Now the network is outputting a probability that it thinks the input is each of the different class categories. These output probabilities are then compared to the target label for the given input, and a loss, or cost is computed. There are many different forms of loss function that can be used from a linear or quadratic difference, to more complicated options, but the essential purpose is always the same. To distinguish how far off the network's prediction was from the target.

Now, finally, the learning can begin. The calculated loss is the result of a long chain of mathematical computation flowing back through the network all the way to the original inputs. Therefore, via gradient descent, the network can search for a lower loss value by stepping the different model parameters comprising the neurons in the direction dictated by the gradient. Figure 4.3 represents a simple cartoon of this gradient descent. Here the ball, representing the network and its parameters roll down some loss-space defined by the various network parameters towards some minimum loss value. However, this is an incredibly simplified depiction, as loss functions can depend on hundreds, or thousands of parameters operating in an extremely high dimensional space.

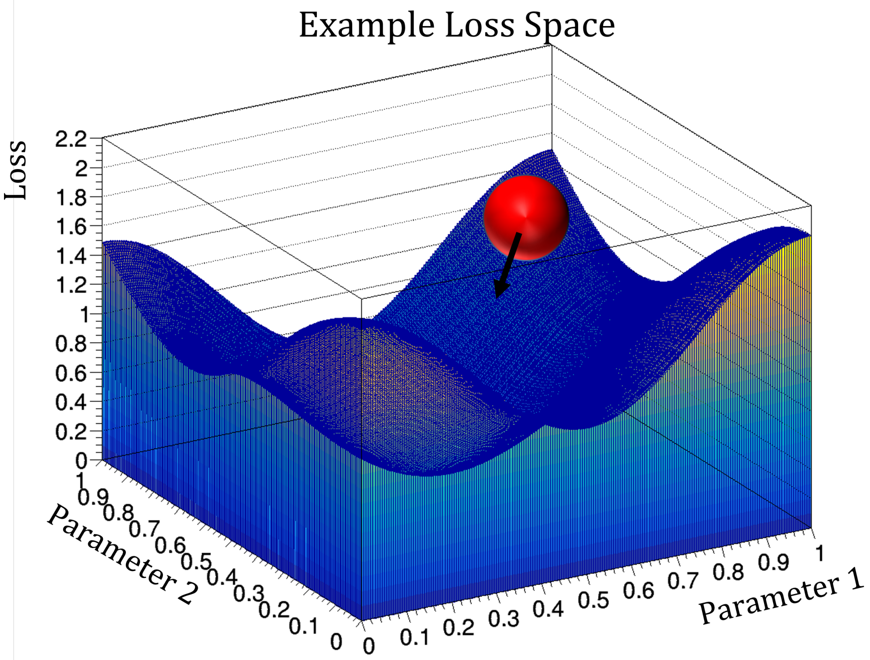


Figure 4.3: A cartoon example depicting gradient descent in a simplified loss-space. The ball represents the network rolling down toward a minimum loss.

One of the key factors leveraged by this gradient descent is that every computation in the network leading toward the loss calculation must be differentiable. However, discontinuity is allowed, such as in the leaky RELU shown above, so long as the gradient is just defined[1] at the points of discontinuity. This procedure for

---

[1] Here, by defined, we mean only in the sense that we can 'cheat' and force the gradient to be defined as something.

(a) Input Image (b) Labeled Image

Figure 4.4: An image of the Tufts Neutrino Group in 2018 (left) and one labeled by a traditionally trained Mask-RCNN network (right), where objects are found, classified, and their pixels clustered.

connecting the gradients leading to loss is known as backpropagation.

Deep learning techniques have proven particularly valuable in the field of computer vision, which strives to use computer algorithms to understand images. For example, a common computer vision task involves image classification, where the goal is to label images based on their contents from among some pre-existing list of categories. Among a dataset of images containing pictures of different animals, a machine learning algorithm will attempt to classify each picture based on the animal it contains, such as a cat or a dog. Figure 4.4 shows the Tufts Neutrino group in 2018, identified via Mask-RCNN, a combination of convolutional neural networks. Mask-RCNN is the subject of our cosmic ray muon analysis described in 8.

4.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are one of the major network architectures in computer vision tasks. CNNs utilize translation-invariant kernels (or filters) that get applied across the image, building up features localized to where the kernel was applied. As the depth of the network grows the 'local' region the kernel is being applied to, relative to the original image, grows larger and larger. A review of CNNs is performed in [47].

The bedrock of CNNs is their implementation of these kernals. These kernels can be thought of as $N \times N$ matrices. In the same as traditional image-processing filters, these kernels get applied to images via a dot-product mechanism whereby the kernel is multiplied with local pixels in the input image. The outputs of these dot-product operations become the feature maps for the next layer of convolutions, analogous to the first layer's input image. Essentially the procedure of applying a kernel is analogous to the artificial neuron described above, where the output of the kernel goes through a similar activation procedure. Many hyperparameters[2] are associated with CNNs and their kernels. These hyperparameters dictate various aspects of CNN operation such as

- how many kernels are applied at each layer and therefore how deep each subsequent feature map is.
- the dimensionality of the kernel, which is typically small, such as 3×3
- the padding applied to the input, which corresponds to 0s padded around the image to allow kernels to be applied to the image's edge effectively
- the kernel's 'stride,' which refers to the spacing in the image between kernel dot-product applications.
- downsampling, such as 'max pooling', where the image or feature map is decreased in size, commonly employed to reduce the size of the information as you go deeper in the CNN. This is to reduce memory and processing costs.

The study performed in section 8.4.2 uses a different form of convolution: sparse submanifold convolutions. These convolutions are more fully described there. To summarize the difference, sparse submanifold convolutions are not applied to the input image or input feature map if centered on an input of 0. This allows the network to save on memory by only storing non-zero parts of the input, as well as save on processing time by skipping many computations that would output 0. There

---

[2] In the field of machine learning, hyperparameters refer to parameters describing the architecture and operation of the network, distinguished from typical model parameters which refer to the numbers that dictate the computation performed to calculate the loss and then that get updated when the loss is minimized.

is a minimal drop in performance as features picked up by the periphery of kernels are missed if the kernel would be centered on a 0 input. The study examines this change more completely.

4.2.2 The Importance of GPUs

Within the field of deep learning, graphics processing units (GPUs) are an important tool. The first implementation of a CNN on a GPU in [48] provided a speed up of about 3 or 4 times compared to a central processing unit (CPU) implementation. The design and typical purpose of GPUs is to perform computations associated with rendering graphics. This allows GPUs to perform many of the same type of task quickly, in parallel. As such they are designed to perform many simple arithmetic operations, quickly. In comparison, a CPU is more adept at general tasks associated with computer processing, like opening, closing, or communicating with, or running executables for, different programs.

A good depiction of the importance of GPUs in deep learning is given in [49], and we will use their analogy for a GPU vs CPU comparison. A CPU is like a sports car, taking a few people quickly from place to place, a GPU is like a truck, moving a lot more stuff, but slower. However, as long as the stuff is all going to the same place, its better than going back and forth time after time. Essentially, GPUs can do lots of simple operations in parallel, whereas CPUs are more linear in their operations.

Figure 4.5 shows a comparison of the number of floating point operations (FLOPs) performed by CPUs vs GPUs with time. The important takeway is that GPUs can provide orders of magnitude gain to processes that are not bottlenecked.

4.3 Boosted Decision Trees

Boosted Decision Trees (BDTs) are used in the DL analysis' selections shown in section 6. *Boosted* decision trees employ machine learning methods in the construc-

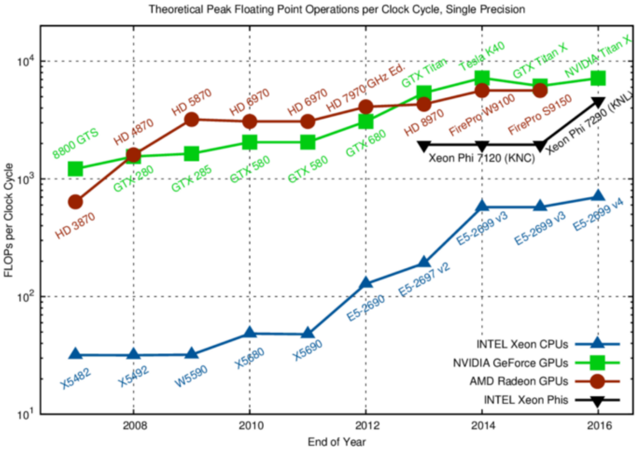


Figure 4.5: A comparison of FLOPs between GPUs and CPUs. Courtesy of fast.ai, [12]

tion of the decision trees. Specifically, BDTs employed in our analyses leverage an implementation called XGBoost [50]. In order to get a reasonably full picture of this, we first start with a simple classic decision tree. Figure 4.6 shows a pictorial representation of a single decision tree. This example tree demonstrates a toy decision trying to classify an image of a dog. The orange nodes represent conditional questions, while the red nodes indicate output scores, these scores can then be used in decision making by adding up the outputs of several constructed trees together, then making a decision from the combined score. The final conditional branches of the tree are called 'leaves'.

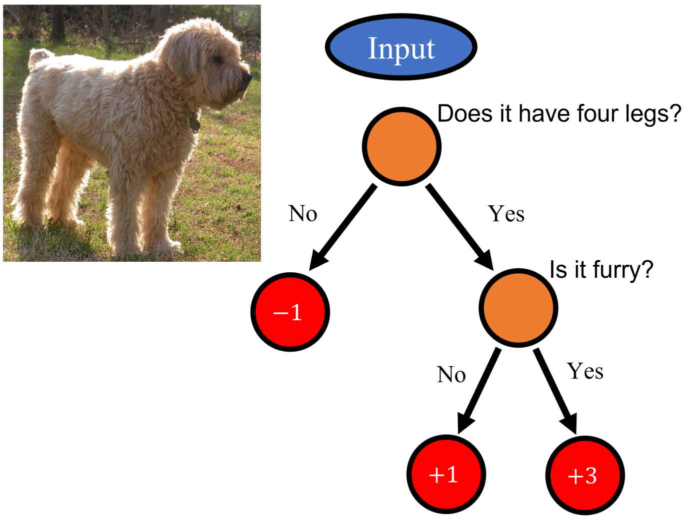


Figure 4.6: A cartoon depiction of a Decision Tree.

The *Boosting* aspect of BDTs comes into play in how the group of trees are built. Individual trees are build by optimizing some ‘objective function’ which consists of a loss function and some regularization term. The tree begins by creating a single conditional statement and its subsequent leaves, optimizing the objective. Then these leaves are split into additional conditional statements (and their leaves) creating the next tree layer in order to further optimize the objective, this is known as gradient boosting.

This gradient boosting is similar to the gradient descent described above, whereby you move in the direction of the gradient to push your objective function in the right direction, except your parameters describe the structure of the tree being added. Sometimes a cost can be required of new leaf additions to ensure they provide enough improvement to justify their added complexity. The depth of the tree in layers is also a hyperparameter of the model. Once a given tree is finished it is locked to the model, and another tree begins construction such that only one tree is optimized at a time. This iterative process can continue until you've reached the desired number of trees (a hyperparameter), or one can stop early if the last several number of added trees fail to increase performance on a dataset held aside from the fit [50].

It is important to note that within this section, our discussion of ”groups of decision trees” should be taken to mean a single BDT with the same training set. In section 6 when discussing ensembles of BDTs we are discussing a group of BDTs each having a unique training set.

Chapter 5

The MicroBooNE DL Reconstruction

5.1 Overview

Information from the MicroBooNE LArTPC comes originally in the form of electronic signals from the wireplane outputs and pulses from PMT array. In order to perform many of the higher-level physics analyses targeted by members of the particle physics community these lower-level data types need to be used to form high-level physics quantities, such as the energy and momentum of particles observed in the detector. This task is not trivial, and requires the 2D information residing in the three different LArTPC wire-planes to be combined into a 3D representation of the particle interactions taking place in the detector.

Within the MicroBooNE analysis, particles interacting with our detector can be classified into two different groups, based on the particle's topology, or the shape of ionization it leaves in our detector. Heavily ionizing particles (HIPs) and minimally ionizing particles (MIPs) leave tracks of charge along the particle's trajectory as the HIP or MIP create ionization electrons in the argon as it moves through the LArTPC. Meanwhile, particles like γ or electrons leave behind a showering cone of ionization, termed 'showers'. Electron showers are created from an electron creating both a typical ionization trail *and* stochastically emitting photons. These photons then produce an electron and positron or Compton scatter. This process then repeats until the energy of the original electron is all used up, or the interaction extends beyond the detector bounds. The process for a gamma shower is effectively the same without the ionization trail from the original electron. This process produces a branching that looks like forked lightning hence the term shower. The energies, momenta, and other physical quantities detailing these showers and tracks are the quantities we can use to perform high-level physics analysis.

The path from the LArTPC raw signals through the deep learning analysis team's reconstruction, and finally onto a neutrino oscillation analysis is shown in figure 5.1. We start by performing signal processing on the LArTPC wire signals, and then format them into three 'wire signal images' corresponding to the three LArTPC wire-planes. This formatting process is described below in detail. Then the DL analysis team uses the MicroBooNE Wire-Cell team's charge-light matching cosmic ray muon tagging algorithms to tag tracks of charge originating from cosmic ray muons, which are a background to any kind of neutrino interaction analysis we want to perform.

After this cosmic tagging, we run a deep learning semantic segmentation network dubbed 'SparseSSNet' to label individual pixels in our image based on the type of particle the pixel's charge originated from. These labels are then used by our vertexing algorithm which reconstructs neutrino interaction vertices by placing 3D candidate vertices within an event. Next we run our track and reconstruction algorithms on each of these candidate vertices. This provides us with a list of vertices and their associated tracks and showers in 3D within an event with which we can begin to perform high-level physics analysis. Using these tracks and showers, we can reconstruct what particles were present in the interaction and various physics quantities associated with the particles, such as their energy and momentum.

In order to perform any analysis, the first step is to define a target sample of events. In the case of the DL analyses from MicroBooNE, we define two target final-states made from neutrino interactions: interactions producing exactly one electron and one proton ($1e1p$) and interactions producing exactly one muon and one proton ($1\mu1p$). Together, these two targets can be generically called one-lepton-one-proton ($1l1p$) events. With our targets now defined, we need to cut our data down to a relative high-purity group of events, while maintaining a reasonable signal efficiency for both interaction types to provide adequate stats. The DL analysis team does this using ensembles of BDTs which use the particle's physics quantities as inputs. The DL team has two ensembles of BDTs, one targeted toward $1e1p$ interactions and the other targeting $1\mu1p$ interactions. These higher purity selections can then be used to perform neutrino oscillation measurements.

Note that both $1e1p$ and $1\mu1p$ feature an event topology that contains two particles coming out of the neutrino interaction, in our detector this consists of a 2-pronged pattern coming from the neutrino interaction point. As such the DL reconstruction was developed with the goal of finding 2-pronged neutrino interactions in order to better isolate these target samples. To aid our selections described in section 6, a particle identification network is also included in the reconstruction chain, which informs the selections what particles are likely present in an interaction.

5.2 Image Reconstruction

The processed signals from the LArTPC wire-planes can be visualized in the format of a trio of images, one image for each wire-plane in the MicroBooNE LArTPC. These images are formatted such that each column of pixels corresponds to a single wire from the wire-plane, and each row refers to a bin of time corresponding to when the signal was read out. So the processed signal read out from the wire is summed across this unit of time, and the resulting signal is stored as the pixel value. The pixel values are correlated to the charge deposited in the detector that then drifts to the wire-planes, though in some arbitrary units, referred to as pixel intensity units (PIU) and shown on the z-axis of our event displays later on in this chapter. Movement horizontally across an image represents moving along different wires, and

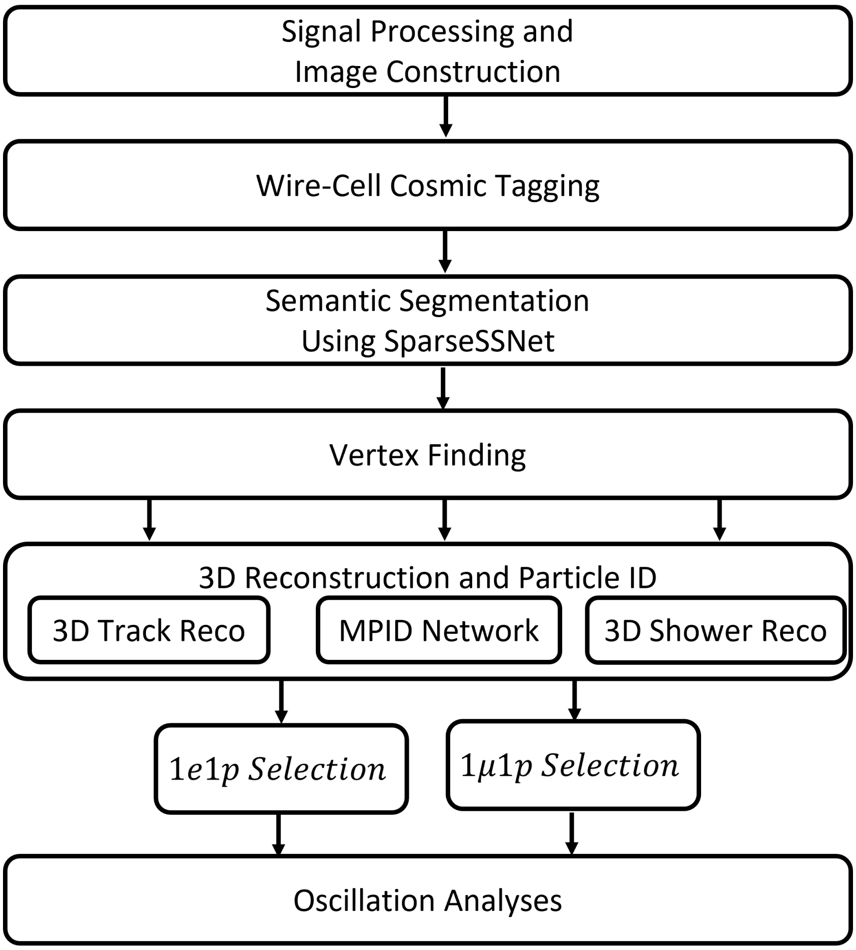


Figure 5.1: A flowchart of the workflow for the Deep Learning analysis team's reconstruction.

movement vertically represents movement in time that the wire signal is being read out.

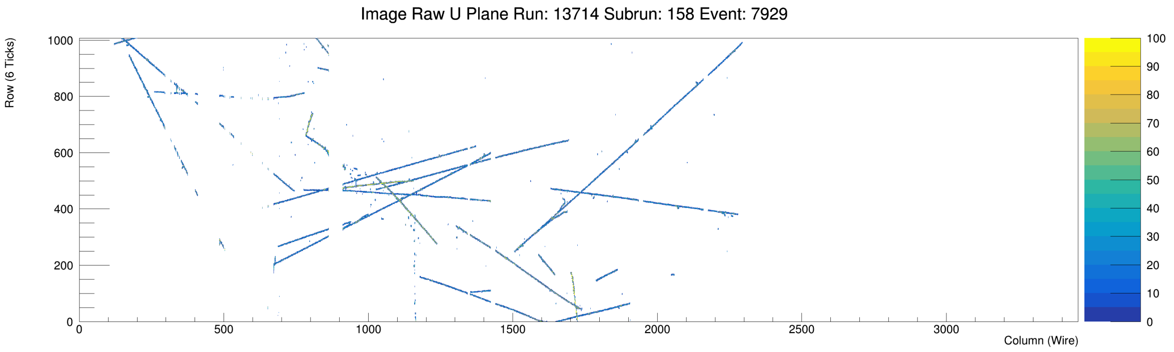
Recall that the MicroBooNE LArTPC, described in section 3.3, has three wire-planes, the first two of which are the induction planes with 2400 wires, while the third plane, the collection plane, contains 3456 wires. In order for our images to be flexible for reconstruction tools, we pad the images associated with the two induction planes with extra columns of zeros so that they contain 3456 columns as well, though columns 2401-3456 are zeros. While the wires are sampled 6048 times over the neutrino beam window of 1.6 ms, for a sampling rate of 2 MHz, the DL analysis team sums these samplings by a factor of 6. This means that the wire-plane images end up with 1008 rows.

This reduction by a factor of 6 was chosen for two reasons, first to reduce the image size to make deep learning applications more tractable, as they can easily run into memory constraints when training. However, the downsampling was also specifically chosen to have an added benefit of making vertical and horizontal movement in the wire-plane images happen at the same scale. When moving across columns horizontally, the distance moved is equivalent to the number of columns crossed multiplied by the wire pitch, which is 0.3 cm in the MicroBooNE LArTPC. Meanwhile, when moving across rows vertically, the distance in the detector being moved is equivalent to the electron drift velocity multiplied by the time associated with the bin. This amounts to $0.5\mu s \times 6 \times 0.1098\frac{cm}{\mu s}$ or 0.33 cm.

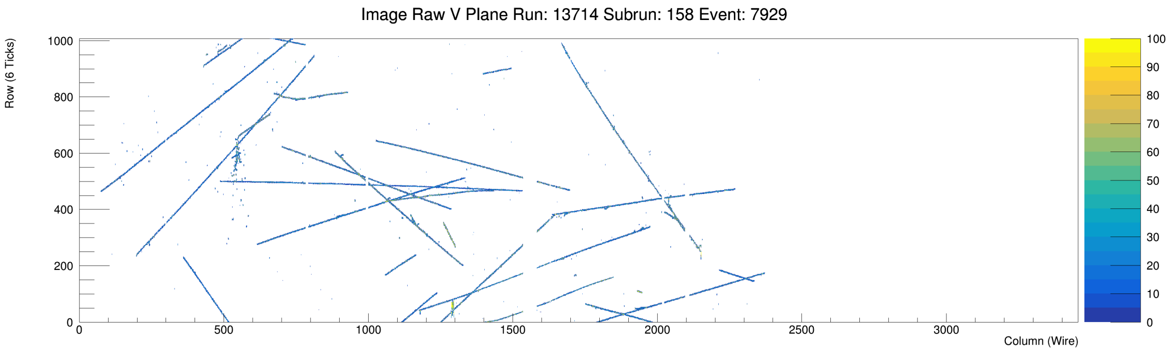
Figure 5.2 shows the wire-plane images for a sample event. The event is a simulated ν_e interaction overlaid atop cosmic ray muon data taken while the neutrino beam was off. This means that the neutrino interaction and resulting daughter particles are simulated, while the cosmic ray muons elsewhere in the event come from data. While the pixel values correspond to the wire signal received during the total rebinned sampling time, for ease of display the image is thresholded at 100.

Note that since the rows correspond to the time that a particular wire was reading its signal, row i in the U induction plane represents the readouts of the wires on the U-plane at the same time as row i for the V and Y planes. So when looking to match an interaction across all 3 planes, one can look at the rows the interaction's features occur on for a single plane, then look for those same features represented in a different projection on the same rows for the other planes.

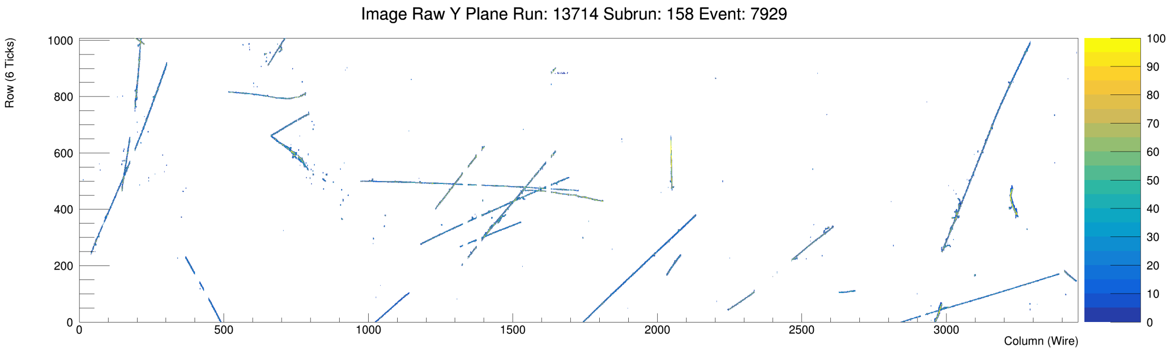
In the different wire-planes images shown, you may notice columns of 'missing' charge, causing breaks in tracks. This is most evident in the U induction plane image 5.2a around columns 850-900 there are breaks in several tracks. These correspond to the unresponsive wire regions in the MicroBooNE LArTPC, described in section 3.3.5. These represent a challenge in our reconstruction.



(a) U Induction Plane



(b) V Induction Plane



(c) Y Collection Plane

Figure 5.2: An example event's wire image displays.

5.3 WireCell Charge-Light Cosmic Tagging

With the event now represented in its image format the next step the DL reconstruction takes is to flag individual pixels as representing charge originating from cosmic ray muons. For analyses targeting any kind of neutrino interaction, these cosmic ray muons represent background, and can interfere with later stages of the reconstruction.

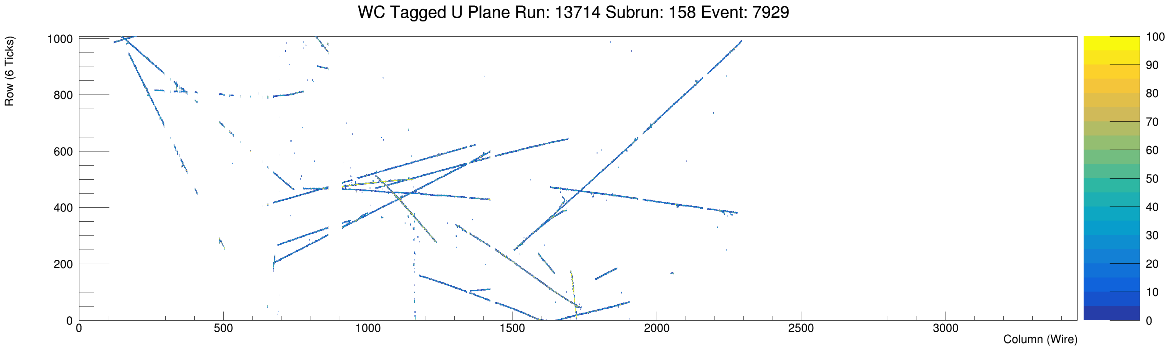
For this process, the DL reconstruction uses the MicroBooNE Wire-Cell (WC) charge-light (Q-L) matching algorithm documented in [51]. In brief summary, the WC Q-L algorithm makes use of the light information detected by the LArTPC array of 32 8-inch cryogenic photomultiplier tubes which sit behind the LArTPC wire-planes. The light information detected during the neutrino beam spill then gets matched to clusters of charge deposited in the LArTPC during the beam spill.

We show the same event from figure 5.2 in figure 5.3, except that only pixels flagged by the WC Q-L algorithm as originating from charge associated with cosmic-ray muons are shown. The inverse is shown in figure 5.4, where only pixels *not* flagged as belonging to cosmic ray muons are shown. In the image containing the flagged pixels we see the many cosmic ray muon interactions cluttering the event.

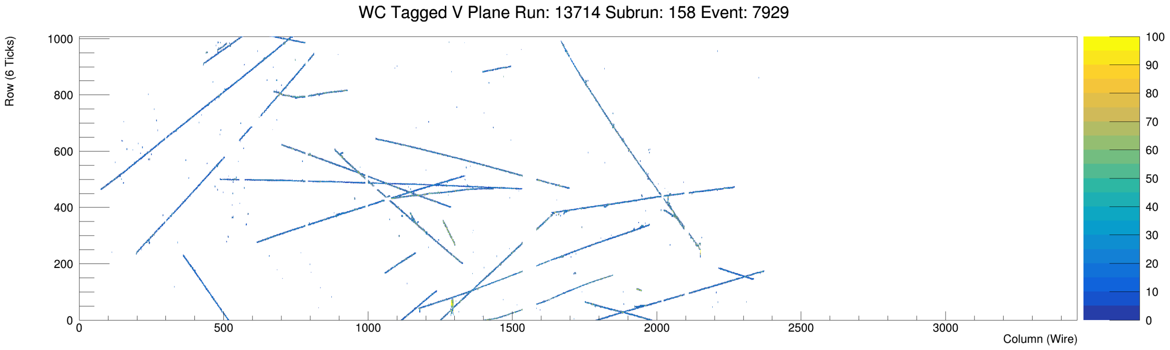
In the figure containing only unflagged pixels we see the success of the WC Q-L algorithm. All that remains is the simulated ν_e interaction with its two prongs. The simulated interaction features one track and one shower. The author implemented this tool into the DL reconstruction chain.

5.4 SSNet Pixel Labeling

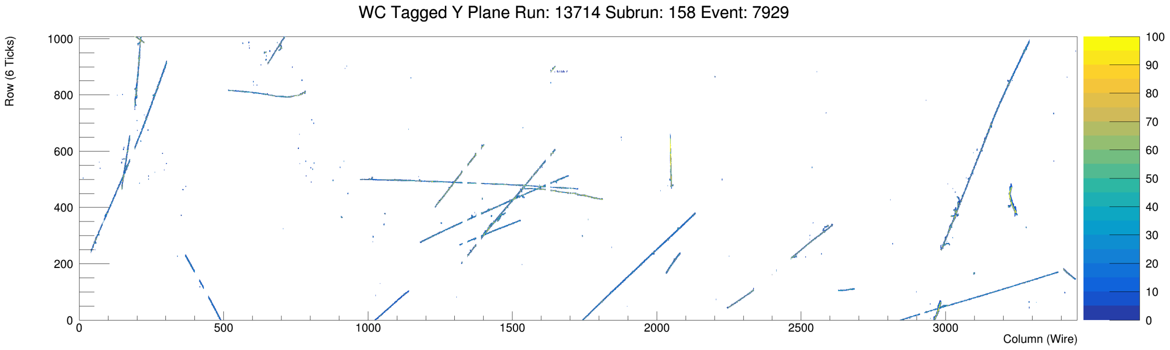
The next step in the DL reconstruction is to run a sparse Semantic Segmentation pixel labeling NETwork dubbed "Sparse SSNET" [52, 53]. Convolutions and sparse convolutions are discussed further in sections 4.2.1 and 8.4.2. The goal of sparse SSNET, hereafter referred to as SSNET, is to classify individual pixels in our detector images as belonging to one of five different categories based on particle



(a) U Induction Plane

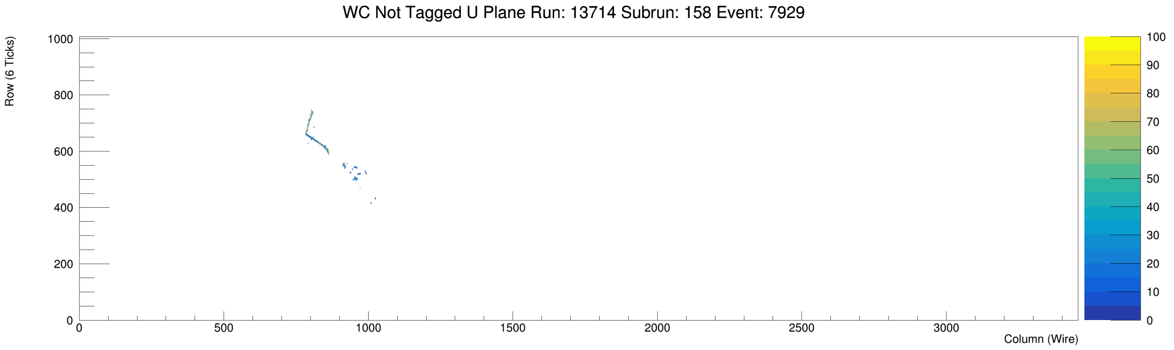


(b) V Induction Plane

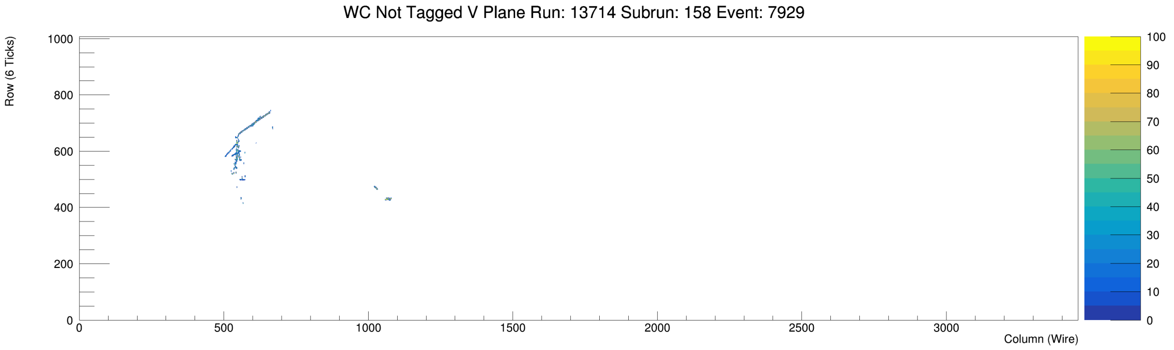


(c) Y Collection Plane

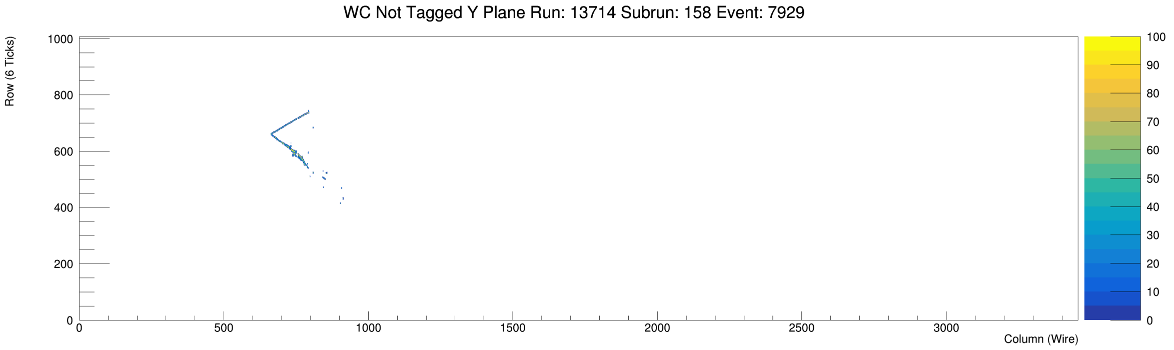
Figure 5.3: The wire image pixels that are flagged as coming from a cosmic ray muon.


WC Not Tagged U Plane Run: 13714 Subrun: 158 Event: 7929

(a) U Induction Plane


WC Not Tagged V Plane Run: 13714 Subrun: 158 Event: 7929

(b) V Induction Plane


WC Not Tagged Y Plane Run: 13714 Subrun: 158 Event: 7929

(c) Y Collection Plane

Figure 5.4: The wire image pixels that are *not* flagged as coming from a cosmic ray muon.

type and origin:

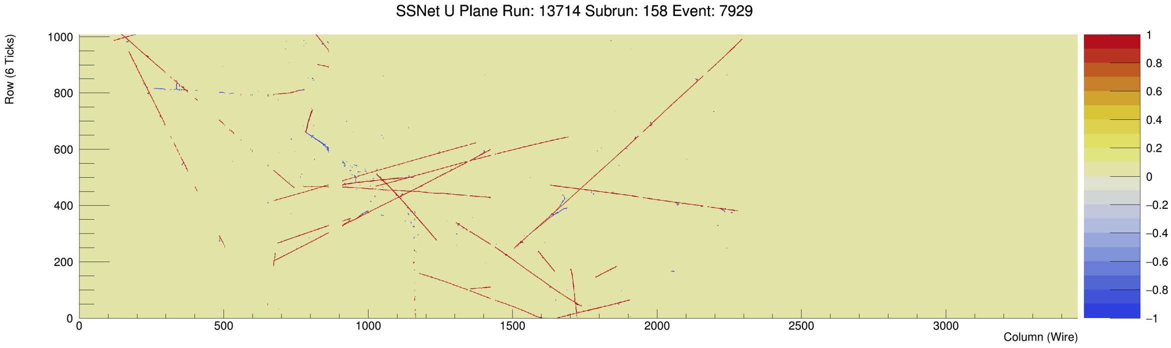
- HIPs: Highly Ionizing Particles, such as a proton
- MIPs: Minimally Ionizing Particles, such as a muon
- Michels: Electrons coming from at-rest muon decay
- Delta Rays: Secondary electrons coming off of HIP and MIP tracks
- Showering Particles: Such as gammas-produced showers, or other non-delta, non-Michel electrons

However, in the DL reconstruction these five different categories get recombined into two categories, track and shower, based on the topology expected to be observed by the LArTPC. The track category is the combination of the HIP and MIP labels, for particles that leave thin lines of ionization in our images. Meanwhile the shower category is the combination of the original Michel, Delta, and Showering Particle labels, and is meant to refer to particles that leave spraying bits of ionization in our images making a cone or triangle shape.

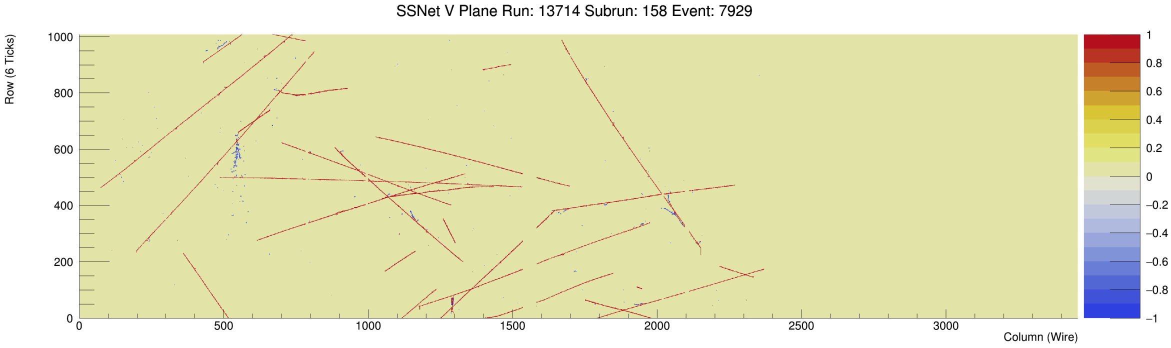
Figure 5.5 shows the pixel labels on our event as determined by SSNET. The track-labeled pixels are shown in red, and the shower-labeled pixels are shown in blue. We see that across all planes the network is able to correctly identify the electron shower coming off the simulated neutrino interaction. SSNET also correctly labels many of the delta rays coming off the various cosmic ray muon tracks passing through the event.

5.5 Vertexer

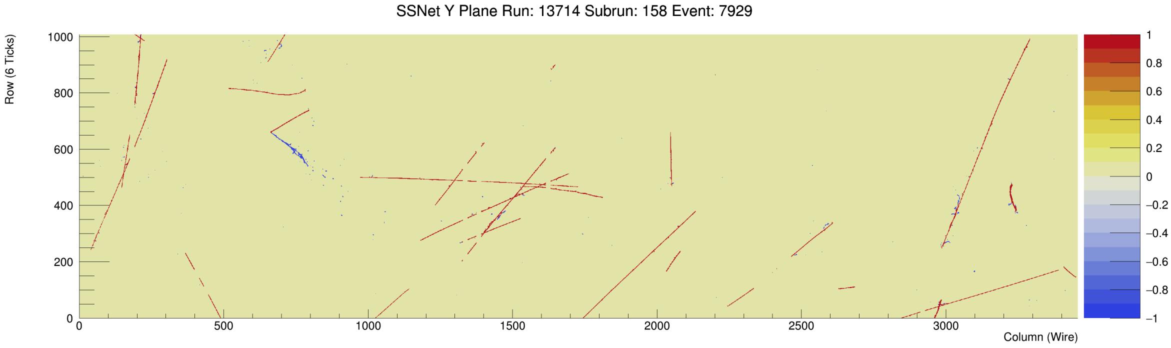
In order to build track and shower objects to constitute a neutrino interaction, the DL reconstruction next turns to placing candidate vertices where a neutrino may have interacted with the argon in the TPC. From these candidate vertices we can start to build up tracks and showers that come from this interaction point. The


SSNet U Plane Run: 13714 Subrun: 158 Event: 7929

(a) U Induction Plane


SSNet V Plane Run: 13714 Subrun: 158 Event: 7929

(b) V Induction Plane


SSNet Y Plane Run: 13714 Subrun: 158 Event: 7929

(c) Y Collection Plane

Figure 5.5: The pixel labeling performed by SSNET. Shower-labeled pixels are shown as blue, while track-labeled pixels are shown as red

DL vertexing algorithm is described extensively in [54]. Here we provide a brief summary of the algorithm.

Recall that the DL reconstruction is primarily interested in reconstructing 2-pronged events, such as $1\mu 1p$ or $1e1p$ interactions. Under this assumption the vertexing algorithm was built to look for ”vee-shaped” kinks or bends in the 2D projections, before searching for 3D consistency in its points by checking the matching rows in the other planes. The vertexing algorithm runs using both the SSNET pixel labels, and the WC Q-L cosmic muon tagged image, allowing it to avoid pixels tagged as cosmics, as well as search for vee-shapes at kinks in clusters of track pixels, and vee-shapes at the intersection of shower pixels and track pixels. The vertex position as found by the vertexing algorithm is shown as a dark, open circle at the neutrino interaction point in figures 5.6 and 5.8.

Within MicroBooNE events in general, there is often no neutrino interaction, and sometimes a single neutrino interaction. While there exists some possibility that two neutrinos interact within the same event time, it is exceedingly unlikely and thus ignored. However, the vertexing algorithm is free to provide multiple candidate vertices for a single event, though some measures are in place to ensure the candidates are not in the same location. It is the job of downstream selection tools described in section 6 to eliminate bad vertices.

5.6 Track Reconstruction

The track reconstruction algorithm runs across all candidate vertices placed in an event. Tracks are built independent of the pixel clustering that is performed by the vertexing algorithm’s prong search. This is because the vertexing algorithm is focused on the region with a vee-shape, and our tracks will often extend beyond its window of interest. The track reconstruction is described in detail in [55], but here a brief summary is provided.

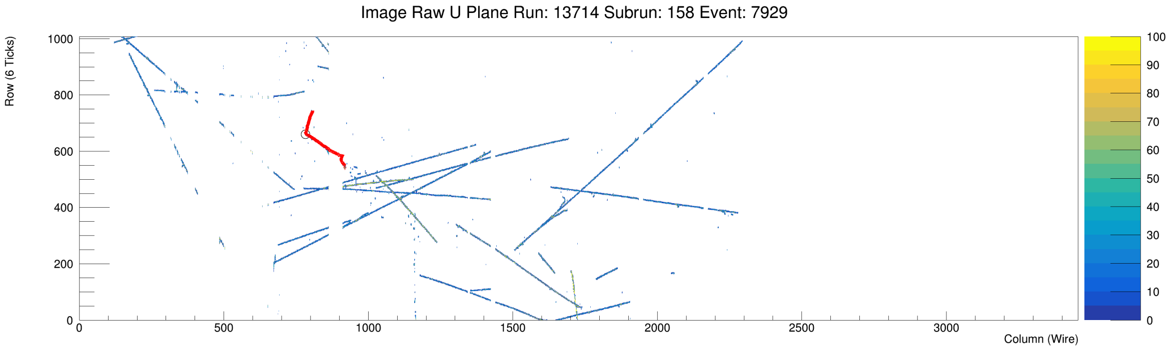
The tracking algorithm takes each 3D candidate vertex and the wire-plane signal images as input, then step by step adds points to the track by throwing a stochastically generated set of points. If operating close to the track's origin, these points are thrown in a sphere around the present point, if greater than 5cm from the origin, then a forward-searching cone of points are also added. From this set of potential points, the step is chosen by favoring points that project onto charge in all 3 planes, and are farther from the current track point. Small allowances are made for the unresponsive regions in our detector, which are described in section 3.3.5, however a track cannot step to a 3D position that projects into a dead region on more than one wire-plane, which limits the track reconstruction somewhat.

Figure 5.6 shows the output of the tracking algorithm on our example event. Two tracks are shown in red, reconstructed from the candidate vertex, which is represented by the black open circle. Since the algorithm works in 3D, the tracks depicted are the 2D projections. The track and shower reconstruction operate independently, so the tracking algorithm is free to build tracks on showers, and it is up to our selections to determine the validity of these tracks. For this particular event, we can see the importance of this aspect, as a track is placed on the electron shower in our ν_e interaction.

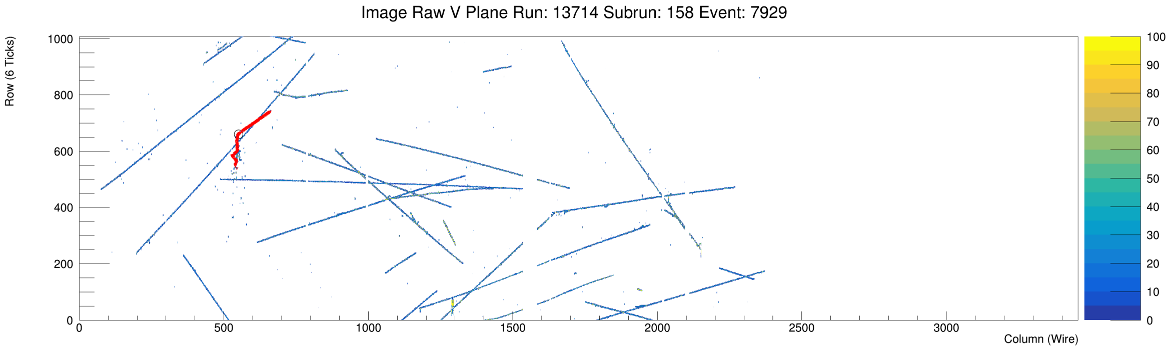
Recall the targeted interaction types the DL analysis aims to reconstruct are $1e1p$ and $1\mu 1p$, both of which contain a proton. In instances where two tracks are built, the proton is defined as the track with greater average ionization energy density, which is to say the track with the higher PIU average when projected into 2D. With a particle ID assigned to the track, we can then calculate the track's energy given the stopping power of the particle, and the 3D track length. The tracking algorithm's energy resolution is $2.5 \pm 0.1\%$ for protons and $3.4 \pm 0.1\%$ for muons [55].

5.7 Shower Reconstruction

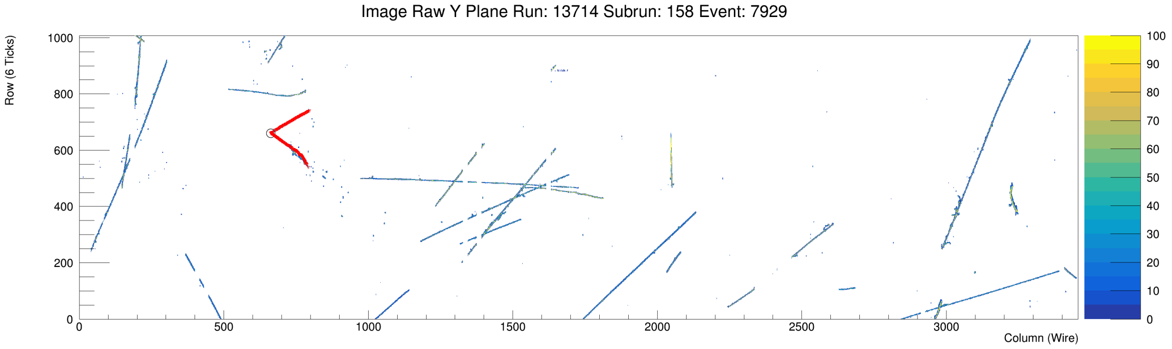
The final step in the DL reconstruction package is dedicated to reconstructing showers. As with previous tasks, the algorithm is summarized briefly here, and explained in detail in [56]. The topology of showers is fundamentally different from tracks, often featuring gaps between their constituent pixels. These gaps come about



(a) U Induction Plane



(b) V Induction Plane



(c) Y Collection Plane

Figure 5.6: The tracks reconstructed by the tracking algorithm are shown in red, stemming from the candidate vertex represented by a black open circle.

from radiated photons spreading from the interaction before ionizing electrons seen by the detector. While figure 5.6 shows the tracking algorithm does a decent attempt at grabbing the shower, a line of points is a poor metric for clustering all these radiated pixels comprising a shower.

The algorithm only uses pixels labeled as shower-like by SSNET, with at least 10 PIU. This PIU cut is typical throughout the DL analysis, as it removes wire signal noise. To reconstruct the shower, the apex of an isosceles triangle is placed on the candidate vertex, then the direction, opening angle, and length of the shower are adjusted to maximize the number of nonzero PIU shower pixels contained within, however constraints are placed on these triangle parameters.

A gap between the vertex and apex of the triangle is also allowed and optimized to allow for showers detached from the vertex. These detached showers can happen when an interaction produces a photon, which then travels some distance before ionizing electrons which are observed by the LArTPC.

Where the energy of a track is determined from its 3D length, the shower energy is determined by the PIU sum of the pixels contained within the 2D shower projection on the Y collection plane image. Only pixels labeled as 'shower-like' by SSNET are used and the shower must pass some requirements indicating the shower is well reconstructed [56]. The Y collection plane is chosen because it requires less signal processing compared to the induction planes. Specifically, the collection plane does not require the transformation of bipolar pulses to unipolar pulses that occur on the induction planes as the charge drifts towards, and then past them, before ultimately collecting on the Y plane.

The output of the shower reconstruction on the Y plane is shown in figure 5.8. The energy resolution of our shower reconstruction is covered in detail in [56].

5.8 Multiple Particle Identification

The final part of the DL reconstruction is the Multiple Particle Identification (MPID) network. The DL analysis makes use of our MPID network in the selections

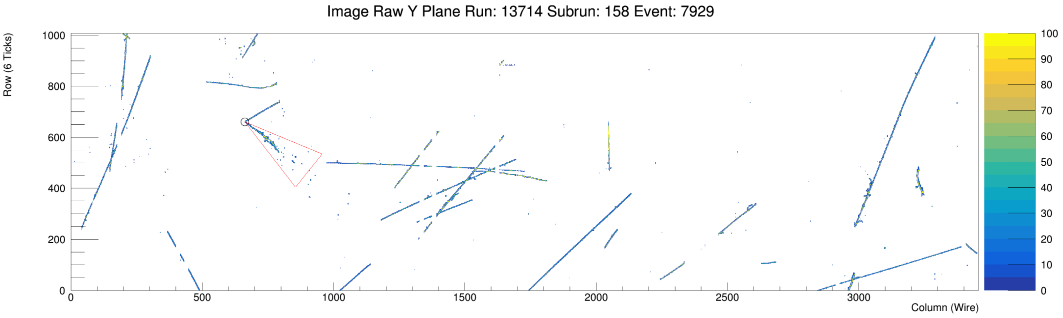


Figure 5.7: Y Collection Plane

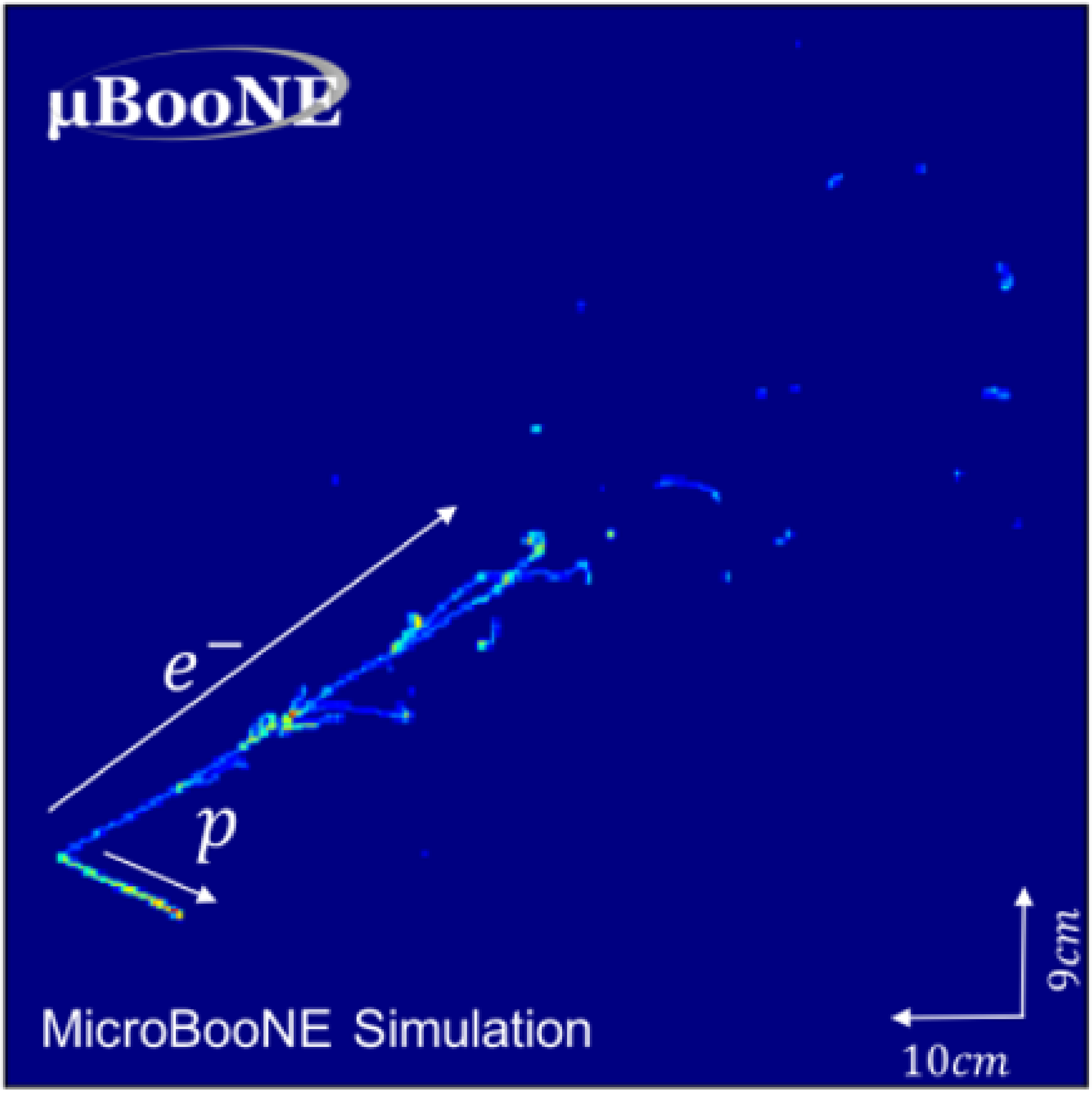
Figure 5.8: The shower triangle reconstructed from the candidate vertex in our example event.

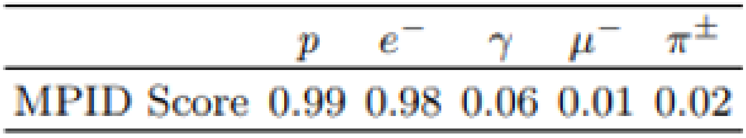
detailed in section 6. The MPID tool is a convolutional neural network designed for particle identification within an interaction. It is covered in more detail in [13], and briefly described here.

The MPID network is trained to look at a 512 × 512 pixel crop around the proposed neutrino vertex in each of the three wire planes. Then it outputs five scores dictating the confidence that the network has that one of five different particles are present somewhere in the interaction. The network outputs these five scores for two different categories, the first category indicates whether the particle is observed *anywhere* in the 512 × 512 pixel crop, the second indicates whether the particle is observed and connected to the vertex at the center of the crop. For the purpose of our analyses only the later scores are used as we are focused on particles present in the neutrino interaction.

The particles the network is trained to identify are the muon, charged pion, proton, electron, and gamma. These scores fall in the range between 0 and 1 with higher values indicating greater confidence the particle is present. Due to the presence of unresponsive wires in the MicroBooNE LArTPC, the maximum MPID score from all 3 planes is used, rather than reliance on a single plane. This helps reduce the case where a particle cannot be connected to the vertex due to an unresponsive set of wires.

The MPID score is used differently for particle identification requirements in both the $1\mu1p$ selection and $1e1p$ selection as outlined in sections 6.4 and 6.5. An example of the visualized output of the MPID network is shown in figure 5.10





	p	e^-	γ	μ^-	$\pi^\pm$
MPID Score	0.99	0.98	0.06	0.01	0.02

Figure 5.9: Y Collection Plane

Figure 5.10: MPID example of a $1e1p$ topology with a tabulated output of particle scores. This image is generated by concatenating a p and an e^- at the same vertex. Scores indicate high probabilities of having a p and e^- in the image. The image applied to MPID has 512×512 pixels. A zoomed-in image of 250×250 pixels is shown here for visualization. [13]

Chapter 6

MicroBooNE DL Selections

Once the MicroBooNE DL team has finished their reconstructions, outlined in section 5, we can use the various high-level reconstructed physics quantities, such as tracks and showers, as well as some low-level quantities, like SSNET pixel labeling, to build up selections for specific interactions types.

At this phase in our analysis, our deep learning tools, SSNET and MPID have been run on our events, and we can now use their outputs to select events pertaining to our desired signals. Machine learning techniques will also come into the selection through our usage of Boosted Decision Trees, shown later on in this chapter.

As mentioned previously, one of the major efforts of the MicroBooNE experiment is to search for the low-energy-excess (LEE) observed by MiniBooNE. This search is conducted in MicroBooNE by the DL analysis using two selections targeting one-lepton-one-proton ($1\ell 1p$) charged-current quasi-elastic events (CCQE).

The first selection targets one-electron-one-proton ($1e1p$) events. If the MiniBooNE-observed LEE were present in MicroBooNE from some unexplained physics in the form of an excess of ν_e flux in the beam, then we would expect an excess observed in our $1e1p$ data selection as compared to our expectation.

The second selection, the $1\mu 1p$, was originally made to constrain the $1e1p$ LEE search. As the BNB contains O(100) times more ν_μ flux, and both $1l1p$ topologies share π^+ decay flux parentage the ν_μ CCQE interactions can be used to constrain the systematic uncertainty of the ν_eCCQE-based LEE measurement. This analogous

$1\mu1p$ selection features two tracks, one muon and one proton, rather than the one-track-one-shower events targeted by the $1e1p$. The original purpose of the $1\mu1p$ selection as a constraint sample is evidenced in many of the design choices described within this section, however the result is still a pure selection of $1\mu1p$ CCQE events. This selection is used in the analysis featured in section 7.

Both these selections are designed in very similar ways, proceeding with many of the same cuts in order for the $1\mu1p$ to maximize the constraint on the $1e1p$. Within this analysis the first group of cuts, termed the 'preselection' cuts, are applied to ensure the event has a two-pronged topology not exiting the detector, and to remove bad quality data, as well as separate events as either potential $1e1p$-like or $1\mu1p$-like. Then the final selections are made using ensembles of Boosted Decision Trees implemented with XGBoost [50], which serve to separate the desired signal from the background remaining after the preselection cuts.

In order to assign meaning to an analysis, it is important to have a studied and understood expectation. In this analysis we use a variety of data samples to determine our expectation through a mixture of non-signal data, and simulated neutrino interactions.

6.1 Data Sample Definitions

When generating neutrinos and their interactions, MicroBooNE uses the GENIE neutrino generator [57]. Then to simulate the detector response to these particle interactions, GEANT4 [58] is used.

Table 6.1 details the size, in protons-on-targets (POT), of different dataset, used in the DL analysis, both data, and the different simulated datasets that comprise our expectation. The datasets are defined later in this section. At a very high level split, these datasets always use simulation to depict neutrino interactions, and data taken when the beam is off to model cosmic ray muons. This is described more specifically for each sample. The metric, POT, is a measure of the number of protons sent to the target. Then through processes detailed in section 3.2 a neutrino

Table 6.1: The size, in protons-on-target (POT), of the various datasets used in the DL analysis.

Sample	Run 1 POT	Run 2 POT	Run 3 POT
BNB Overlay	1.34×10^{21}	1.30×10^{21}	8.88×10^{20}
BNB Low Energy Overlay	1.63×10^{21}	2.03×10^{21}	1.51×10^{21}
ν_e Overlay	1.16×10^{23}	9.21×10^{22}	4.72×10^{22}
ν_e Low Energy Overlay	6.05×10^{23}	N/A	5.97×10^{23}
BNB NC π^0 Overlay	2.91×10^{21}	N/A	2.49×10^{21}
BNB CC π^0 Overlay	6.91×10^{20}	N/A	5.91×10^{20}
Dirt Overlay	N/A	N/A	1.88×10^{20}
Off-Beam Data	1.11×10^{21}	N/A	1.57×10^{22}
On-Beam Data	1.63×10^{20}	2.75×10^{20}	2.29×10^{20}

beam is created. As such, POT is a measurement for the amount of data for a given sample, and a proxy for how many neutrinos the detector will have seen.

In table 6.1 the various data samples have their POT broken down by run. In the MicroBooNE experiment, Run 1 corresponds to data taken from February to September of 2016, Run 2 from October 2016 to October 2017, and Run 3 from October 2017 to September 2018. Simulated data samples are made with different parameters using calibration data taken from the LArTPC from each of these different run periods to account for fluctuations or changes to the detector state with time.

In general, the samples listed in the table can be grouped into two categories, those with simulation (often referred to as Monte Carlo (MC) samples) and those without. The off-beam[1] and on-beam samples make up our samples comprised completely of actual data taken from the LArTPC, with no simulation. The on-beam data is taken when the BNB is on, during a neutrino spill, and the detector triggers to save an event. The off-beam data is taken either when the BNB is off and the detector triggers to save an event, or in between the BNB neutrino spills, when there will not be a neutrino interaction.

On-beam data is therefore the actual data that analyses will treat as their obser-

---

[1] The effective POT of the off-beam samples are calculated by taking the ratio of number of off-beam triggers occurring during the sample with the number of off-beam triggers for a known on-beam data sample, then multiplying by that data sample's POT. This is necessary because the external BNB sample is, by definition, taken while avoiding beam neutrino interactions

vation, searching for neutrino interactions and phenomenology. The off-beam data is useful because it contains events specifically lacking a neutrino interaction, but still saved via a trigger. These events provide a useful handle on the cosmic-ray muon background to MicroBooNE, as they essentially are comprised of only cosmic-ray muons, and can be used to model the rate at which we would expect to select these events from our on-beam data.

The remaining samples listed in table 6.1 contain some form of simulation. The 'overlay' present in the sample names indicates that the event contains a simulated neutrino interaction 'overlaid' on top of an image containing a special set of off-beam data cosmic ray muons. These underlying off-beam data events *do not* require the detector to trigger, as the overlaid neutrino is required to pass the PMT trigger cut to save the event. When true quantities about the cosmic ray muons are not needed it is advantageous to use overlay samples, as they represent the exact detector response to, and observation of, cosmic ray muons. In the cases where true information regarding the cosmic ray muons is required, we turn to CORSIKA [59], to provide the simulation, as in section 8 where a novel machine learning technique, sparse submanifold convolutions, are applied to identify and cluster the cosmic ray muon background.

When simulated events are generated, GENIE [57, 60, 61, 62] v3.00.06 and model set $G18\_10a\_02\_11a$ are used as the primary model. This simulation package uses Valencia CCQE and meson exchange current (MEC) (2p-2h) modeling [63] and the Local Fermi Gas nuclear model. These models provide a good match with the MiniBooNE CCQE-like data [64]. The event generation also uses a data-driven improvement to final state interactions (FSI) as well as a tune to bubble chamber data for pion production. In particular, these FSI account for nuclear effects where the neutrino interaction's outgoing hadrons can interact with the nucleus before they exit. These final state interactions can then change the observed particles coming out of the neutrino interaction. Therefore, these FSI cause systematic uncertainty in our analysis as events can shift into and out of our CCQE $1\mu 1p$ signal. The

MicroBooNE collaboration then performed a tuning procedure [65] using T2K ν_μ CC, zero-pion data [66]. Finally the simulation of these generated particles is done using Geant4 [58] V10.3.03c. In addition to these weights a reweighting is done of simulated events containing π^0 final state particles. This reweighting scheme was designed to alleviate a slight deficit in a π^0-selected sideband sample between the data and expectation. This reweighting predominantly affects the $1e1p$ selection, with minimal impact on the $1\mu1p$ selection. The reweighting process is described more fully in [14].

The BNB overlay samples are a sample of events overlaying simulated neutrino interactions that are meant to represent all neutrino interaction types that could come from the BNB. Recall that the BNB, described in section 3.2 predominantly contains ν_μs and only has a small ν_e impurity, absent any new physics phenomenology. This means that most of the BNB overlay sample contains simulated ν_μ interactions across many different channels, but includes a ν_e contribution as well.

Since these ν_e interactions are so rare, MicroBooNE has generated a ν_e overlay sample, which specifically contains only ν_e interactions in the detector to increase the statistics of these types of interaction. This is necessary because the ν_e interactions create the target signal of the $1e1p$ selection described in section 6.5 searching for evidence of the MiniBooNE LEE. These ν_e interactions come from ν_e flux intrinsic to the BNB, as opposed to ν_e interactions that would come about from neutrino oscillation, and have a different energy distribution.

MicroBooNE also uses two low-energy overlay samples, one for the BNB, and one specifically for $\nu_e$interactions. The low-energy samples are the same as the normal samples, with the exception that only events with ν energy less than 400 MeV are simulated. These samples *are not* used to build the expectation to match against observation in the on-beam data. Instead they are just used to develop and check our selection performance in the LEE signal region.

The next two samples specifically only include ν interactions from the BNB that create a π^0, whether through a charged current (CC) interaction or a neutral current

(NC) interaction depending on sample. These π^0s then decay to two γ particles which then produce showers in the detector. These π^0 samples are therefore useful both in calibrating our shower reconstruction described in section 5.7, as well as measuring the π^0 background in our selections.

The dirt overlay is a special source of simulated events. It is comprised of simulated ν interactions that occur outside the detector volume, but whose daughter particles enter the detector and leave some signal. These events would be incredibly difficult to reconstruct, as the core of the interaction would not be seen. However, the simulated dataset was generated to determine if these events would pose as a significant background to our various signals. As they did not contribute a significant effect to the expectation in our selections, they are omitted from this point onward.

Finally, notice that several samples do not have a Run 2 portion. In these cases we use the Run 3 sample to determine the Run 2 expectation. Run 3 is chosen because there were known issues with wire noise and PMT light collection in Run 1 that were solved from Run 2 onward.

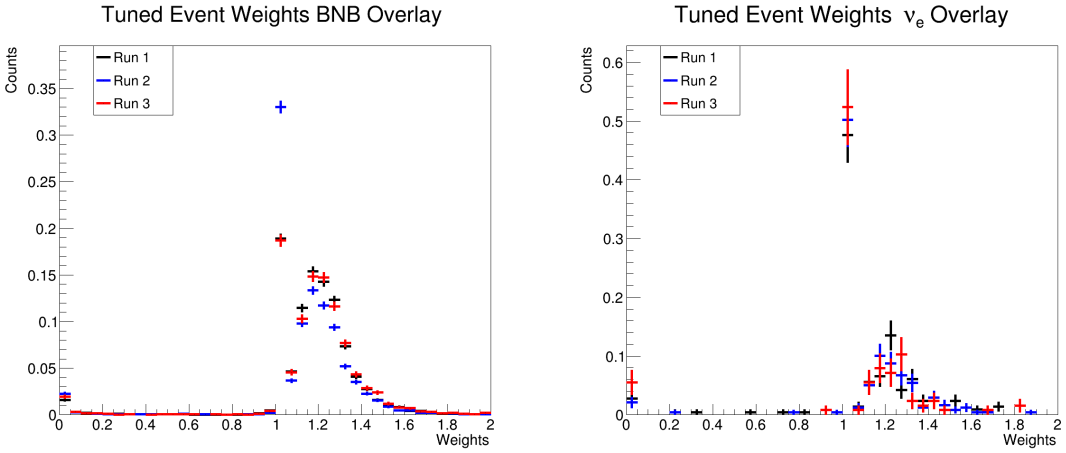
6.2 Building an Expectation

Now that the many different samples have been defined, we can outline how the expectation is built up to compare to the observed on-beam data. We can build a *toy example* of how the expectation is crafted by scaling the various samples to the on-beam observation. In the toy samples represented in table 6.2, it is demonstrated how the raw number of events across our expectation samples can be scaled to the amount of data. This toy example is just for demonstration purposes and is not representative of any data selection, or event count. However, the POTs and POT ratios are calculating using the POTs for the different samples in Run 1.

First the raw events get multiplied by tune weights meant to reshuffle the frequency with which events are expected to happen relative to the original simulated sample. The distributions for these weights range generally from just below 1 up

Table 6.2: A table representing the scaling process of different expected samples to the observation for a *toy model of events*. The POT ratio is calculated by dividing the on-beam sample POT by the expectation sample's POT.

Sample	Raw Events	Weighted Events	POT	POT Ratio	Scaled Events
BNB	187	194.92	1.34×10^{21}	0.1216	23.71
NC π^0	59	62.84	2.91×10^{21}	0.0560	3.52
CC π^0	16	18.19	6.91×10^{20}	0.2359	4.29
ν_e	1623	1643.93	1.16×10^{23}	0.0014	2.31
Off-Beam	73	73.00	1.11×10^{21}	0.1468	10.72



(a) BNB Overlay

(b) ν_e Overlay

Figure 6.1: The distribution of tune weights applied to the BNB (left) and ν_e (right) overlay events.

to around 1.4 with a tail extending beyond that, and a major spike at 1 for events where the simulated sample appropriately generated the interaction mode's rate. These distributions are shown in figure 6.1. Note that tune weights only exist for simulated events. The off-beam data, which determines the expected cosmic ray muon background, does not receive tune weighting. As well as getting their individual event weights, each sample then has to be scaled by some factor to match the POT anticipated in the on-beam data observed.

In order to do this, each sample is multiplied by the ratio of the *on-beam data POT / the sample's POT*. This allows us to use a high stat simulated sample to determine with greater accuracy the expectation at some given on-beam sample. To avoid double counting, ν interaction modes that could appear in multiple samples are only used from the single, higher stat, sample. For example, NC π^0 interactions that are present in the BNB overlay are ignored if they are also simulated in the dedicated NC π^0 overlay sample. The same for ν_e interactions, and CC π^0 interactions.

6.3 One Lepton One Proton Preselection

Before the event selections branch off into the differing $1\mu1p$ and $1e1p$ selections, a suite of requirements or 'cuts' can be made to focus in on well-reconstructed two-prong event topologies. These requirements, used by both the major selections discussed later, are referred to as preselection cuts.

The preselection requirements are listed here and described below:

1. Common Optical Filter Cut
2. Beam Quality Cut
3. Good Runs Cut
4. Vertex Requirements
5. Orthogonality Shower Fraction Cut
6. Boostability Cut
7. Opening Angle Cut
8. Containment Cut
9. Particle Energy Requirements

The first cut, the Common Optical Filter, is used by all LEE analyses within MicroBooNE, beyond just the DL team. It requires that at least 20 photo-electrons are received by the PMTs in the beginning of an event window, right after the beam spill, as well as requires that less than 20 photo-electrons are received right before the beam spill. This optical system cut aims to reduce the background noise in an event by ignoring events that already had something going on before the beam spill.

Next the beam quality cut is in place to remove events taken during periods where the quality of the BNB is questionable based on monitoring systems in place surrounding the BNB generation. The Good Runs cut is in place to remove data associated with periods of time there were known issues as determined through manual monitoring of various detector systems. Note that these cuts only apply to data taken by the detector. Simulation events, by construction, will pass these requirements.

There are also several requirements placed on the interaction topology to ensure a $1\ell 1p$-like event. These "Vertex Cuts" include a requirement that the candidate vertex has exactly two prongs that are greater than 5cm in length, a requirement that the vertex is more than 10cm away from the fiducial volume[2], and that the vertex is not placed in a specific region of the detector associated with a large density of unresponsive wires. This unresponsive region is centered around a plane at $Z = 720$cm, taking up ~3.9% of the detector.

Now the events are split into two paths, one towards the $1\mu 1p$ selection, the other towards the $1e1p$ selection. The DL analysis introduces an orthogonality requirement based on the output of SSNET as described in section 5.4. If either of the two particles reconstructed from a given vertex have more than 20% of their pixels labeled as 'shower' pixels, then that vertex and interaction are considered $1e1p$-like and they continue down the $1e1p$ selection path, otherwise it is considered $1\mu 1p$-like and go down the $1\mu 1p$ selection path. This split serves to ensure that the sample used for the $1e1p$ LEE search is orthogonal to the $1\mu 1p$ sample used to constrain it. This is to say that no single reconstructed neutrino interaction can exist in both final selections.

The $1l1p$ selections also require that the interaction be *boostable*. As noted in table 6.3, several of the variables used later in the selection are calculated in the nucleon rest frame. In order to perform a Lorentz boost, γ must be defined. However,

---

[2] Where the active volume is the volume of LAr that drift electrons can reach the wire-planes, the fiducial volume corresponds to a volume describing the active volume with an extra 10cm buffer on all edges

since our interaction involves a neutrino interacting with a nucleon, Fermi motion adds momentum due to the quantum motion of the nucleon. This nonrecoverable momentum may ultimately yield a non-physical, undefined γ. These events are cut from our selections as they are likely some sort of background and not a well-reconstructed CCQE $1l1p$ event. The process for performing this boost is further explained in appendix A.

Next a cut is applied to the interaction's opening angle, requiring it to be at least 0.5 radians (28.6 degrees), ensuring a wide enough opening to be well-reconstructed.

A containment requirement is applied to the reconstructed tracks, ensuring that the tracks do not get within 15 cm of the active volume. There is an additional containment cut applied to showers, thus only affecting the $1e1p$ selection, that eliminates interactions with showers in heavily unresponsive regions of the detector. Since the energy of track-like particles comes from their length, there is a possibility of jumping the unresponsive wires. But for showering particles, the energy comes directly from the amount of signal detected in the shower pixels, so losing many pixels can disrupt the energy reconstruction.

Both the lepton and the proton have an minimum energy requirement applied to them as well. The reconstructed proton needs to have at least 50 MeV, and the lepton needs to have at least 35 MeV. Note that the lepton's energy calculation is described in section 6.3.1 and is calculated differently for the $1e1p$'s showering electron than for the $1\mu1p$'s track-like muon. Regardless, these energy thresholds are set to ensure the particle has enough energy to reliably produce a multi-pixel signal in the LArTPC's wire-signal images.

6.3.1 BDT Input Variables

Now that the events are split into pools of $1\mu1p$-like, and $1e1p$-like events, they are ready to be fed into the Boosted Decision Tree (BDT) Ensembles. A summary of a BDT is given in section 4.3. In the DL selections, ensembles of BDTs are used. Rather than selecting events from one BDT score, the average score of all BDTs in an ensemble are used. This is different from simply expanding the size of a single BDT because each member of the ensemble receives its own randomly determined training set. This serves to ensure minimal reliance on a good representative draw to train on.

As the behavior of the LArTPC is known to undergo some changes with runtime as the electronics readout is monitored and problems solved, different BDT Ensembles are used for each Run 1, 2, and 3. This means that in total, the selections described here use six different BDT Ensembles. This allows the BDTs to prescribe differing levels of importance to their input variables from run to run.

Following the spirit of keeping the $1\mu1p$ selection similar to the $1e1p$, many of the variables used by the $1e1p$ BDT Ensemble are also used by the $1\mu1p$ BDTs. Table 6.3 details the breakdown of which variables are used in the two different selections. Broadly speaking, the variables used in the BDTs can be ascribed to two different categories.

The first, smaller category of variables are based on the ionization deposited in the wire-plane images surrounding the interaction. There are four ionization-based variables. Only one variable is used in both the $1\mu1p$ and $1e1p$ selections, the amount of charge deposited within 5 cm of the interaction vertex, this serves to indicate the level of vertex activity going on in an interaction. There are then three additional variables used in the $1e1p$ selection based on ionization. The fraction of charge in shower-labeled pixels in the entire image divided by the charge in shower-labeled pixels in the electron shower, the fraction of the proton track that is shower-labeled, and the fraction of the electron shower that is shower-labeled. In all these cases pixels are labeled as shower-like by SSNET.

The second, larger category of variables consists of kinematic variables such as particle energy, angular direction, and various other quantities. These variables are calculated under the assumption that the interaction is the type that is being selected, for example, events going into the $1e1p$ BDTs assume the track is a proton, and the shower is an electron. This allows the BDT to 'learn' to check for consistency between actual $1e1p$ interactions' variable distributions, and background interactions calculated as though they were $1e1p$ interactions. Therefore to calculate these variables, we use $m_\mu = 105.6584$ MeV, $m_p = 938.673$ MeV, $m_n = 939.5654$ MeV and $m_e = 0.511$ MeV.

When examining these variables, recall the coordinate system used for the MicroBooNE LArTPC, defined in section 3.3.1. The z-direction points along the beam, the x-direction points from the anode wire-planes to the cathode, and the y-direction points up vertically. The cylindrical coordinate system has ϕ rotating around the z-axis, and θ describing the angle dropped from the z-axis.

The definitions of these kinematic variables are listed in table 6.4. They begin with the determination of the kinetic energy of the proton and the lepton. For protons and muons, as track particles, the energy is determined via the track length based on the known stopping power of the particle in argon. This process is described in more detail in [55]. For showers, the kinetic energy is determined via a calibrated conversion factor to convert the wire-image's PIU to energy, described further in [56]. The 3- and 4-momenta of our protons and leptons can then be determined given the particle trajectory indicated by the tracks and/or showers, which allows the calculation of our angular variables θ and ϕ. Finally, this analysis assumes a binding energy of 40 MeV from [67].

With these basic variables determined, we move toward the higher-level variables further down the table. Of particular interest is the energy of the neutrino, which can be reconstructed in three different ways. Equation 6.1 outlines the default method, termed the range-based method, because it is determined via the range-based determination of the proton and muon's energy (or in the case of the electron, the ionization conversion). This E_ν^{range} calculation is used when our analyses place events in reconstructed neutrino energy bins for our histograms, and is the default energy used when discussing the interaction. However, given the assumption of a quasi-elastic (QE) interaction containing known particles, the neutrino energy can be determined solely using information about the lepton or proton.

Variable	Used in $1\mu 1p$ BDT	Used in $1e1p$ BDT
Variables Used in BDTs, Based on Ionization		
Charge within 5 cm of vertex	Yes	Yes
Shower charge in event image / shower charge clustered as electron	No	Yes
Proton shower fraction	No	Yes
Electron shower fraction	No	Yes
Variables Used in BDTs, Related to Energy Measurements		
Neutrino Energy	Yes	Yes
Energy of electromagnetic shower	No	Yes
Lepton length	Yes	Yes
Proton length	No	Yes
$p_z - E_\nu$	No	Yes
Variables Used in BDTs, Related to 2-Body Scattering Consistency		
Bjorken's x	Yes *	Yes *
Bjorken's y	Yes *	Yes *
QE Consistency	Yes *	Yes *
Q_0	Yes	Yes
Q_3	Yes	Yes
Variables Used in BDTs, Related to Transverse Momentum		
α_T	Yes	Yes
Event p_T	Yes	Yes
Event p_T/p ("PTrat")	Yes	Yes
ϕ_T	Yes	No
Variables Used in BDTs, Related to Angles		
Proton ϕ	Yes	Yes
Proton θ	Yes	Yes
Lepton ϕ	Yes	Yes
$\phi_p - \phi_\ell$	Yes	Yes
$\theta_p + \theta_e$	No	Yes
Variables Useful for Comparison, Not Used in Either BDT		
Opening Angle	No	No
x Vertex	No	No
y Vertex	No	No
z Vertex	No	No

Table 6.3: The variables used as inputs to the $1\mu 1p$ and $1e1p$ BDTs. If an $*$ appears, the variable is used in the boosted frame of reference.

These QE-assumed energies are defined in eqn. 6.2 and 6.3 for the proton and lepton respectively. These energies are used in the definition of of our 2-body consistency variable, Δ^{QE}. The purpose of this variable is to capture how consistent these different energy calculation methods are. In the case of a well-reconstructed $1\ell 1p$ QE interaction the energies should be similar, and Δ^{QE} should be close to zero.

If the interaction is not well-reconstructed, or not an actual $1\ell1p$ QE interaction, then the equations should calculate differing neutrino energies, with Δ^{QE} growing as the equations indicate inconsistency with the QE $1\ell1p$ assumption.

$$\begin{aligned}
E_\nu^{range} &= \mathrm{K}_p + \mathrm{K}_\ell + M_\ell + M_p - (M_n - B), && (6.1)\\
E_\nu^{QE-p} &= \left(\frac{1}{2}\right)\frac{2\cdot(M_n - B)\cdot E_p - ((M_n - B)^2 + M_p^2 - M_\ell^2)}{(M_n - B) - E_p + \sqrt{(E_p^2 - M_p^2)}\cdot\cos\theta_p}, && (6.2)\\
E_\nu^{QE-\ell} &= \left(\frac{1}{2}\right)\frac{2\cdot(M_n - B)\cdot E_\ell - ((M_n - B)^2 + M_\ell^2 - M_p^2)}{(M_n - B) - E_\ell + \sqrt{(E_\ell^2 - M_\ell^2)}\cdot\cos\theta_\ell}, && (6.3)
\end{aligned}$$

Now that both the preselection cuts and BDT variables have been defined we are ready to split off into the two different $1l1p$ selections. In both selections a BDT is defined and trained, detailed below. In events where multiple candidate vertices are available even after these preselection cuts, the vertex with the highest BDT score is considered, as such no event can have two selected vertices.

6.4 $1\mu1p$ Selection

With preselection cuts out of the way, the remaining expectation events pushed toward the $1\mu1p$ selection path can be used to train the $1\mu1p$ BDT ensemble. Events failing the preselection are not used in training because it is unnecessary for the BDTs to try to learn anything related to them. Instead it is better to allow the BDT features to focus on discriminating the background from signal in the remaining events.

The goal of the BDT ensemble is to provide a strong handle to select the well-reconstructed CCQE $1\mu1p$ remaining after the preselection, while removing poorly reconstructed events. While the signal is well defined as $1\mu1p$ CCQE interactions, the background the BDT is training to remove is varied, covering many different interaction topologies. Figure 6.2 shows the expected distribution of events by category in reconstructed neutrino energy at the preselection stage. At this stage

Variable Name	**Definition**
Base Variables	
K_p	Kinetic energy of proton determined from range [55]
K_ℓ	Kinetic energy determined from range for muons [55] and from calorimetry for electrons [56]
M_ℓ, M_n, M_p	Masses of the lepton, neutron and proton from Ref. [3]
$\cos\theta_p$, $\cos\theta_\ell$	p_p^z/p_p, p_ℓ^z/p_ℓ
ϕ_p, ϕ_ℓ	$atan2(p_p^y, p_p^x),\ \ atan2(p_\ell^y, p_\ell^x)$
$P_p = (E_p, p_p)$, $P_\ell = (E_\ell, p_\ell)$	Reconstructed 4-vector of the proton, lepton
B	Binding Energy for argon; the analysis assumes $B = 40$ MeV
Definitions Related to Neutrino Energy	
E_ν^{range} (Default value of E_ν, eq. 6.1)	$\mathrm{K}_p + \mathrm{K}_\ell + M_\ell + M_p - (M_n - B)$
E_ν^{QE-p}(eq. 6.2)	$(\frac{1}{2})\frac{2\cdot(M_n - B)\cdot E_p - ((Mn - B)^2 + M_p^2 - M_\ell^2)}{(M_n - B) - E_p + \sqrt{(E_p^2 - M_p^2)}\cdot\cos\theta_p}$
$E_\nu^{QE-\ell}$ (eq. 6.3)	$(\frac{1}{2})\frac{2\cdot(M_n - B)\cdot E_\ell - ((Mn - B)^2 + M_\ell^2 - M_p^2)}{(M_n - B) - E_\ell + \sqrt{(E_\ell^2 - M_\ell^2)}\cdot\cos\theta_\ell}$
Δ^{QE} (2-Body Consistency)	$\sqrt{(E_\nu^{range} - E_\nu^{QE-p})^2 + (E_\nu^{range} - E_\nu^{QE-\ell})^2 + (E_\nu^{QE-\ell} - E_\nu^{QE-p})^2}$
Event Kinematics	
Q^2	$2E_\nu^{range}(E_\ell - P_\ell^z) - M_\ell^2$
Hadronic Energy (E_{had})	$E_\nu^{range} - E_\ell$
Björken's Scaling x (x_{Bj})	$Q^2/2M_nE_{had}$
Björken's Scaling y (y_{Bj})	E_{had}/E_ν^{range}
Opening angle	$\cos^{-1}(\hat{p}_\ell\cdot\hat{p}_p)$
p_T	$\sqrt{(p_\ell^x + p_p^x)^2 + (p_\ell^y + p_p^y)^2}$
p_L	$p_p^z + p_\ell^z$
α_T	$\cos^{-1}\left(-\frac{\vec{P}_T^l\cdot\vec{P}_T}{\lvert\vec{P}_T^l\rvert\lvert\vec{P}_T\rvert}\right)$
ϕ_T	$\cos^{-1}\left(-\frac{\vec{P}_T^l\cdot\vec{P}_T^p}{\lvert\vec{P}_T^l\rvert\lvert\vec{P}_T^p\rvert}\right)$
Boosting Parameters	
p_{fermi}^T	$p_p^T + p_\ell^T$
p_{fermi}^z	$p_p^z + p_\ell^z - E_\nu^{range}$

Table 6.4: Kinematic variable definitions as derived from the reconstruction code used in this analysis.

the background is still very much dominant.

As a brief aside, many plots shown in this section demonstrate distributions before the final selection is made. For these plots, systematic uncertainties and expectation-statistical errors are not calculated. The final selection plots do include these uncertainties. For more information concerning how systematic uncertainties are evaluated, see section 7.1. As an additional note, many of the early-in-selection plots omit data, as various early cuts were applied to the data samples available for analysis at the time the plots were made.

The major categories of events can be broken down as:

- **BNB ν_μCCQE Events:** The signal events containing a $1\mu 1p$ CCQE interaction from the BNB that do not fall into one of the background categories.
- **BNB Bad Reco:** Signal events that have their energy reconstruction off by more than 20% the simulated true neutrino energy.
- **BNB $1L1P$ Off Vertex:** Signal events where the reconstructed vertex is more than 5 cm from the simulated neutrino interaction vertex. These events are typically either instances where the vertex is placed partially down a track or shower, away from the true vertex, or placed on a cosmic ray muon within the neutrino event window.
- **BNB Not $1L1P$:** Events that do not contain exactly one muon and one proton, but are in the BNB simulated sample.
- **Cosmic Bkg:** Events in the off-beam sample, where a cosmic-ray muon has passed the preselection, masquerading as a $1\mu 1p$ event.
- **Other:** A classification including all categories that individually constitute less than 5% of the preselection, in this plot this includes ν_μ events with a π^0 or $\pi^\pm$, ν_μ meson-exchange-current (MEC) events, and ν_e events, as well as any other interaction not defined by an above category, but present in the BNB simulation.

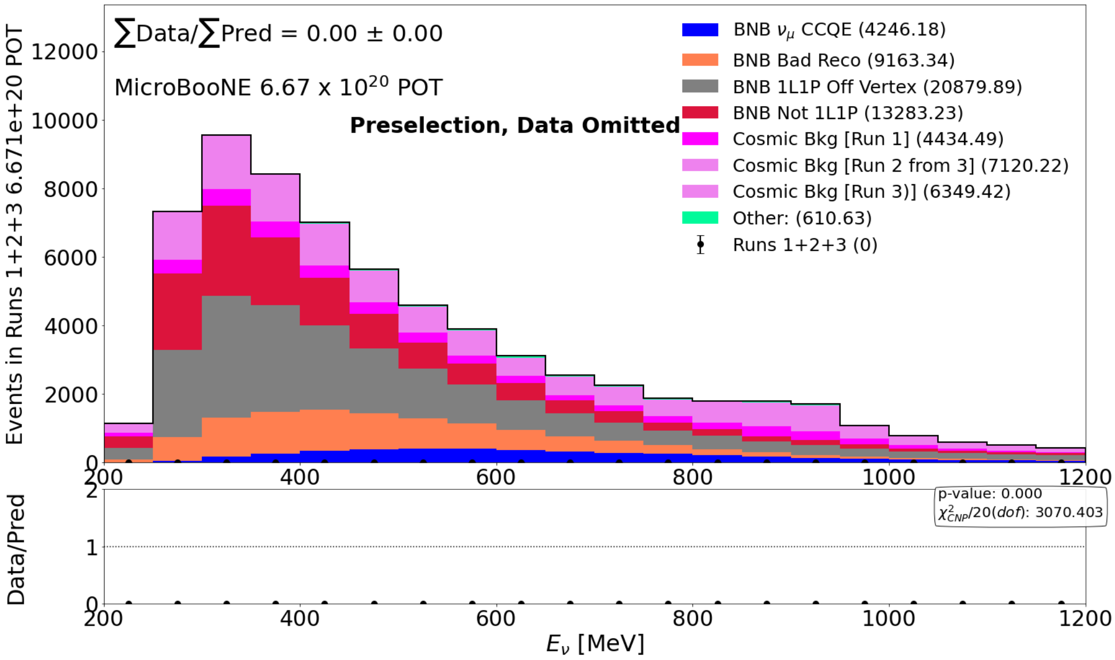


Figure 6.2: The reconstructed neutrino energy distribution at the preselection stage, broken down by interaction category. Data not shown.

We see that the BDT Ensemble has its work cut out for it, with a CCQE purity of about 6.5% and a muon neutrino purity of roughly 72.6%. Several important neutrino background interaction modes are discussed below in section 6.4.1. However, recall that the goal of the preselection is to acquire a set of two-prong events that are contained within the detector. The BDT input variables for all of these events can now be calculated under the assumption that the event is a $1\mu1p$ CCQE interaction. Then the BDT can determine the validity of that assumption.

An important aspect of BDT training, and any sort of tool fitting, is that the BDT can be used on data other than that which it is trained (or fit) on. This extrapolation is necessary in order for the BDTs to reliably be used on actual data. Otherwise the analysis risks overfitting their selection tools to the specific expectation they are using, and being unable to perform on new data.

In order to help avoid overfitting, the BDTs in our BDT ensemble are only trained on half of the expectation, with each BDT in the ensemble splitting the expectation randomly. This serves to decrease the reliance on the random draw, and is the purpose for using a BDT ensemble. For the $1\mu1p$ selection, the BDT ensemble consists of 10 BDTs per MicroBooNE Run period.

The expected spectra was constructed such that no BDT in the ensemble was used to classify an event it was trained on. A BDT that was trained on a given event has already been told the truth of the event, and could bias unfairly towards a correct answer if deployed on that event when building an expectation, this unfair bias could cause observation in data to not match the expectation.

In order to avoid this, when determining the average BDT score for a specific event, only BDTs that were not trained on an event were used. This means that for events making up our prediction, only $10 - N$ BDTs are used in the average score calculation, where N is the number of BDTs that were fit on the given event.

The BDTs within the ensemble are designed such that signal events receive target scores of 1, and all background events have target scores of 0. Figure 6.3 shows the distribution of the BDT ensemble average scores. We can immediately see the BDT ensemble is working properly as the background events pile up in the first bin close to their 0-target. Meanwhile the navy blue, ν_μ CCQE events drift towards the right, pushing up toward their target score of 1.

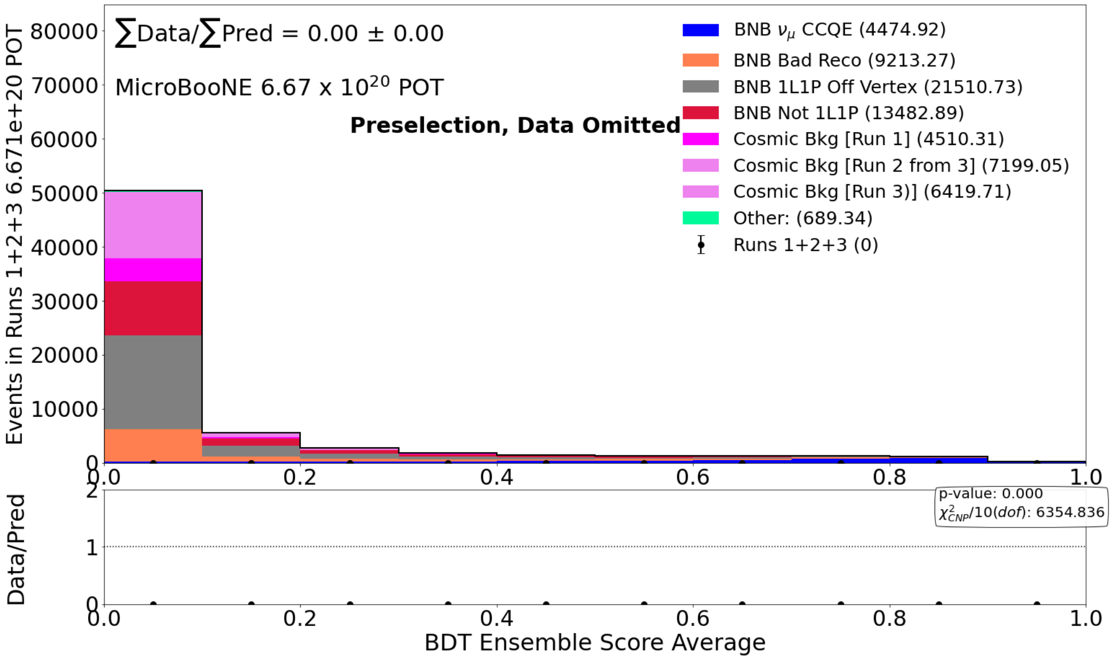


Figure 6.3: The BDT ensemble average score distribution at the preselection stage, broken down by interaction category. Data not shown.

The next step is to determine a cut value to further select the target $1\mu 1p$ interactions. In the case of the $1\mu 1p$ selection, the BDT cut was chosen in order to optimize the $1e1p$ selection's sensitivity to the MiniBooNE LEE signal translated to MicroBooNE's expectation. Ultimately, a BDT average score of 0.5 is required of events in the $1\mu 1p$ selection. It is easy to see in figure 6.3 how impactful this cut will be in removing the background.

This nearly final selection is shown in figure 6.4. While there is agreement between the expectation and the observation, there is a significant pile-up of background at the lower range of energies, specifically in the bad energy reconstruction and off-vertex categories of BNB simulated events.

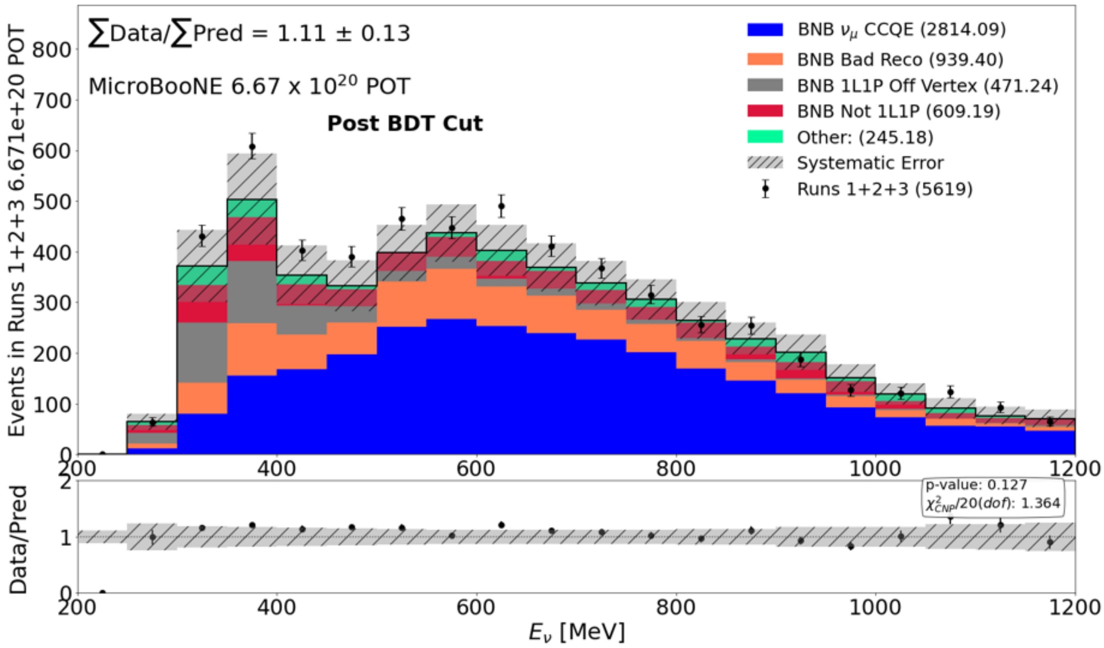


Figure 6.4: The neutrino energy distribution for the selection after the BDT cut is applied, much of the background is removed, and data is shown.

The major issue this pile up of backgrounds represents is a specific state where these background events are able to imitate a proton and go unnoticed by the cuts used thus far. For example, an off-vertex event in the BNB sample may have the vertex placed along a cosmic ray muon, breaking the track in two, with each part representing a prong in the two-track $1\mu 1p$ assumption. Then a portion of the muon track is able to present, incorrectly, as a proton. To clean up the background a bit more, a few final requirements are applied to the selection. These cuts are distinguished from the preselection cuts in that the are applied after the BDT ensemble, and therefore events they remove are still included in the BDT training sample.

In order to remove this background, we invoke another deep learning tool discussed in section 5.8, the Multi-Particle-Identification network (MPID), in a limited, but important way. As mentioned, the MPID network outputs scores indicating the confidence that the event contains a specific particle. For example, a high 'Proton MPID' score indicates a high confidence that the network believes the interaction contains a proton.

In order to tackle this pile up of low energy background events, we employ a requirement that events with reconstructed neutrino energy below 400 MeV have a proton score requirement of 0.9 or greater. This forces the network to be confident that a proton exists in low energy events in the pile-up region. Figure 6.5 shows the proton MPID scores for events in this low-energy region.

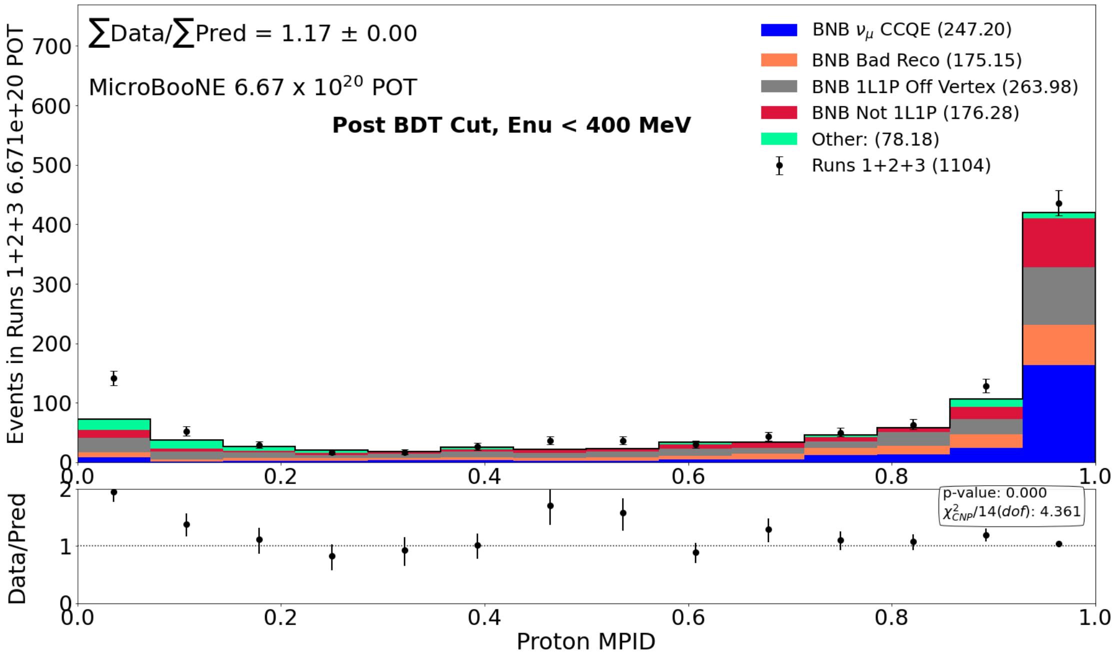


Figure 6.5: The proton MPID score distribution after the BDT cut is applied for events with neutrino energy reconstructed less than 400 MeV.

The penultimate cut involved in the $1\mu 1p$ selection requires that $cos(\theta_p)$, the cosine of the proton's θ angle, be greater than 0, indicating that the proton is forward going relative to the beam direction. While backward-going protons are possible, the forward direction is heavily favored in a QE neutrino interaction. The distribution of events in the cut region, below 0, is shown in figure 6.6. As shown, much of the events in this region are indeed backgrounds to the well-reconstructed $1\mu1p$ CCQE events.

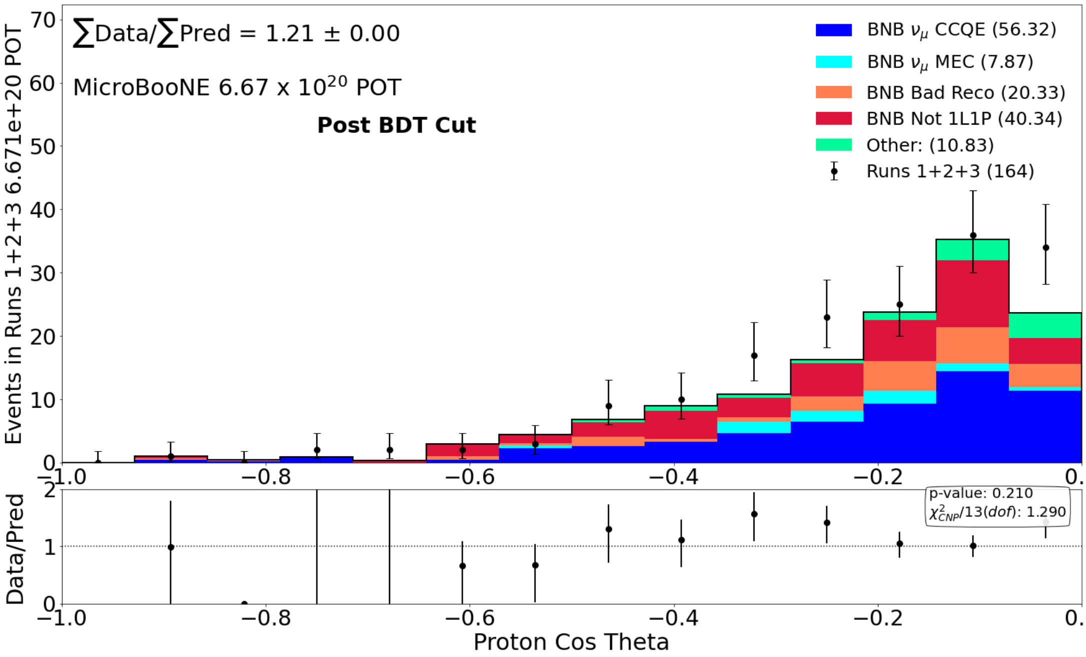


Figure 6.6: The distribution of $cos(\theta_p)$ after the BDT cut, specifically in the backward-going proton region.

Now, the final cut applied to the $1\mu1p$ selection requires that events exist in our analysis' energy range, from $200-1200$ MeV. With this final cut applied, we produce our final selection, as shown in [14], and used for the disappearance analysis work in section 7. For the purpose of our final selection plots, we recombine the categorical breakdown of different events. Specifically, the bad energy reconstruction events are redistributed to their appropriate truth category which explains the increase in the ν_μ CCQE events. All other neutrino events, except for off-vertex events, are rolled into one ”Neutrino Background” classification. Off-vertex events remain separated as they may be, and frequently are, caused by a vertex being placed on a cosmic ray muon rather than a neutrino within the event window.

It should be noted that of this neutrino background expectation, only 1.26 events are expected to be ν_e interactions. This is important to remember for the ν_μ disappearance analysis presented in 7, because it means both the ν_μ CCQE and neutrino background categories are subject to ν_μ disappearance oscillation effects.

Figure 6.7 shows the final selection, binned in the reconstructed energy of the neutrino. Now, the systematic and expectation-based statistical uncertainties are incorporated into the plot. For final selection level plots for different $1\mu1p$ variables, see appendix B. We see within these plots that our selection has 8% more events in data than our expectation, however our uncertainties cover this, with a normalization uncertainty of 13%. Further details surrounding the validation of the $1\mu1p$ BDT ensemble-based selection are explored in the following subsections.

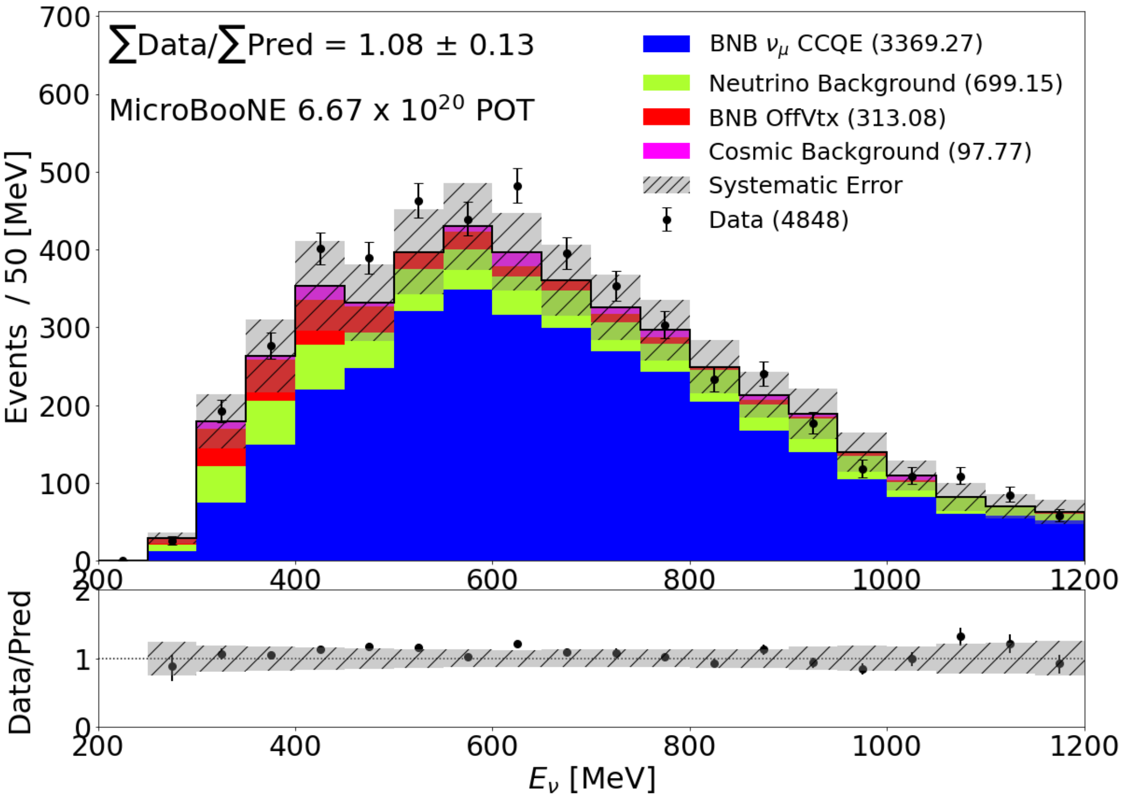


Figure 6.7: The range-based reconstructed neutrino energy distribution for the $1\mu1p$ final selection.

6.4.1 Background Neutrino Interaction Modes to the MicroBooNE $1\mu1p$ Selection

While we have already discussed the CCQE neutrino signal interaction targeted by the $1\mu1p$ selection, there exist several other neutrino background interaction

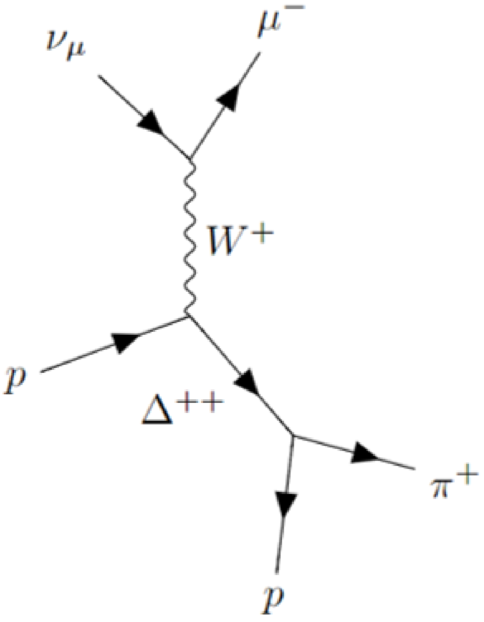


Figure 6.8: Example of a charged-current resonant interaction. A muon neutrino interacts with a proton to give a muon and a resonant Δ^{++} which then decays into a π^+ and a proton.

modes. Here we describe these different interaction channels. In order of their relevance to the $1\mu1p$ selection as neutrino background events, we will discuss resonant pion events, meson exchange current (MEC) interactions, and deep inelastic scattering (DIS).

Our first class of neutrino background interaction are resonant pion events. Specifically, as it relates to MicroBooNE, and our selection, these events are those where the incoming neutrino interacts with an argon nucleon, but rather than eject a nucleon, such as the CCQE interaction's final state proton, instead a resonant Δ state is created, which then decays. These decays can come in different forms, dependent on the type of Δ created, consisting of either a neutron or a proton and a pion. These pions then decay, yielding additional final state particles to the interaction. Figure 6.8 shows a Feynman diagram of an example resonant interaction where an incoming neutrino interacts with a proton to create a Δ^{++} and a muon, then the Δ^{++} decays to a π^+ and a proton. This π^+ can then decay to yield an anti-muon.

Our next class of neutrino background interaction are meson exchange current (MEC) interactions. This interaction mode is commonly classified as a "2 particle-2 hole" (2p-2h) effect [68]. In these interactions, the neutrino exchanges a W boson with a pair of nucleons. This process then results in two nucleons being added to the final state particles.

Our selection's neutrino background also includes deep inelastic scattering (DIS) interactions. These DIS events refer to interactions where the incoming neutrino scatters off a quark 'deep' within a nucleon. This results in high momentum transfer in order to blow apart the nucleon. The quarks then hadronize into complicated final states. The inelasticity of these interactions comes from the loss of kinetic energy in order for this hadronization to occur. As mentioned this process is subdominant at the energies of the MicroBooNE experiment, but scales with neutrino energy as the neutrino becomes more able to probe at the quark level. At 20 GeV, neutrino DIS becomes dominant [69].

The uncertainty associated with the strengths of these background neutrino interaction modes contribute to our analysis' systematic uncertainty associated with the neutrino-argon cross section, discussed in section 7.1.2.

6.4.2 The Important Variables in the $1\mu1p$ BDT Ensemble

The relative power of each of the variables used in the BDT training is shown in figure 6.9, where the feature importance is a measure how frequently a variable is used in the BDT tree. Most variables are self-explanatory, with additional definitions in Tables 6.4 and 6.3, but a few are abbreviations that need to be clarified: *Phis* – the sum of the ϕ; *PTRat* – the ratio of event transverse momentum to total momentum; and *SphB* – the two-body consistency.

What you see is that one variable, the ratio of total transverse momentum to total momentum for the $1\mu1p$ signal, *PTRat_1m1p* is most helpful at distinguishing whether an event is in the signal or background. After that, x_{Bj} and quasi-elastic consistency are slightly more helpful than the other variables. Let us consider why these variables are powerful.

We are aiming to isolate two-body scatters. For the case of no Fermi motion these events would be expected to have zero total transverse momentum for the outgoing scattered particles, since the neutrino enters along the z axis. Another way to say this is that the outgoing event vector will not be rotated with respect to the z axis. The variable p_T/p for the event (both particles added together) is equivalent to measuring the cosine of the angle of rotation away from z. Fermi motion will introduce a rotation with respect to the z axis at a small level. But background events will have much larger rotation. For example, mis-reconstructed cosmic ray background will have very large rotation, since the muons are primarily downward-going. As another example, events with additional particles that were missed, and hence reconstructed as $1\mu1p$ will exhibit large expected rotations. Thus, *PTRat_1m1p*, the measure of this rotation may be expected to be a good discriminator.

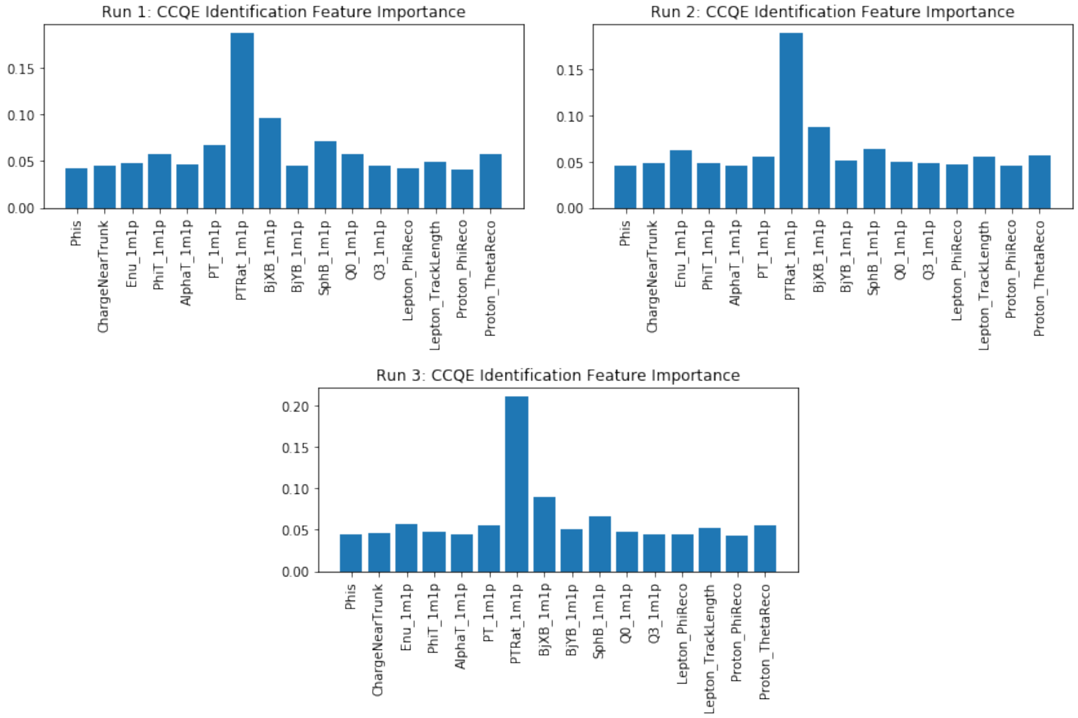
The x_{bj} variable measures the level to which the event is described by elastic scattering. In this case, most muons are correctly identified, so the strength of the variable is not coming from assuming an incorrect mass. Instead, for $1\mu1p$ events, many background events will reconstruct with improbably high Q^2 because the muon energy and angle are both large compared to what is expected for CCQE scattering.

The third most important variable, called *SphB_1m1p*, is the QE consistency, called Δ^{QE} on table 6.4. This cross compares E_ν^{range}, $E_\nu^{QE\ \mu}$ and $E_\nu^{QE\ p}$, which will all have the same value in the case of CCQE scattering. It is unsurprising that this is a useful variable in selecting CCQE events.

Thus the variables indicated as strong in figure 6.9 make sense, and we see that the remaining variables have similar nonzero importances, which shows the BDT relies on the other variables roughly evenly.

6.4.3 The $1\mu1p$ CCQE Selection Efficiency

The total efficiency of our $1\mu1p$ CCQE Selection for our BDT Ensemble is shown in figure 6.10. The efficiency numerator is defined as all ν_μ CCQE events that


Run 1: CCQE Identification Feature Importance

Run 2: CCQE Identification Feature Importance

Run 3: CCQE Identification Feature Importance

Figure 6.9: The relative importance of each input variable distribution to the training of a typical $1\mu 1p$ BDT in the Run 1 Ensemble (top left), Run 2 Ensemble (upper right), and Run 3 Ensemble (bottom middle). This highlights how effective each variable is for differentiating signal from background. For discussion of variables, see tables 6.3 and 6.4.

are in our selection as defined above. The efficiency denominator is defined as all the ν_μ CCQE events that occur with the simulated neutrino interaction vertex within the active volume of our detector. Below 500 MeV, the efficiency is low because the denominator has a significant fraction of events with muon and proton energy too low to pass cuts. Above ∼500 MeV, the efficiency is reduced by longer tracks hitting our containment requirement or crossing dead regions. Note that this is the total selection efficiency, which includes inefficiencies surrounding actually reconstructing the neutrino interaction as well as selecting them. For example, roughly 50% of events are missed by the vertex reconstruction stage. The remaining loss in efficiency is then distributed across track reconstruction, and the various selection cuts, resulting in this total efficiency.

Note that because the $1\mu 1p$ simulated signal requires good energy reconstruction if a simulated CCQE event is selected, but has bad energy reconstruction it will not be counted in the efficiency numerator. Further, the denominator includes some events that are not practically possible for our analysis to reconstruct or select. This can be due to the interaction exiting the detector, making the energy hard to reconstruct, or if the muon or proton energies are small enough to be below the 35 and 50 MeV thresholds set at the analysis. Finally, while the $1\mu1p$ selection has an explicit cut on the reconstructed energy of the neutrino at 1200, the efficiency plot is calculated without that cut applied to the selection in order to show the efficiency at higher true neutrino energies accurately.

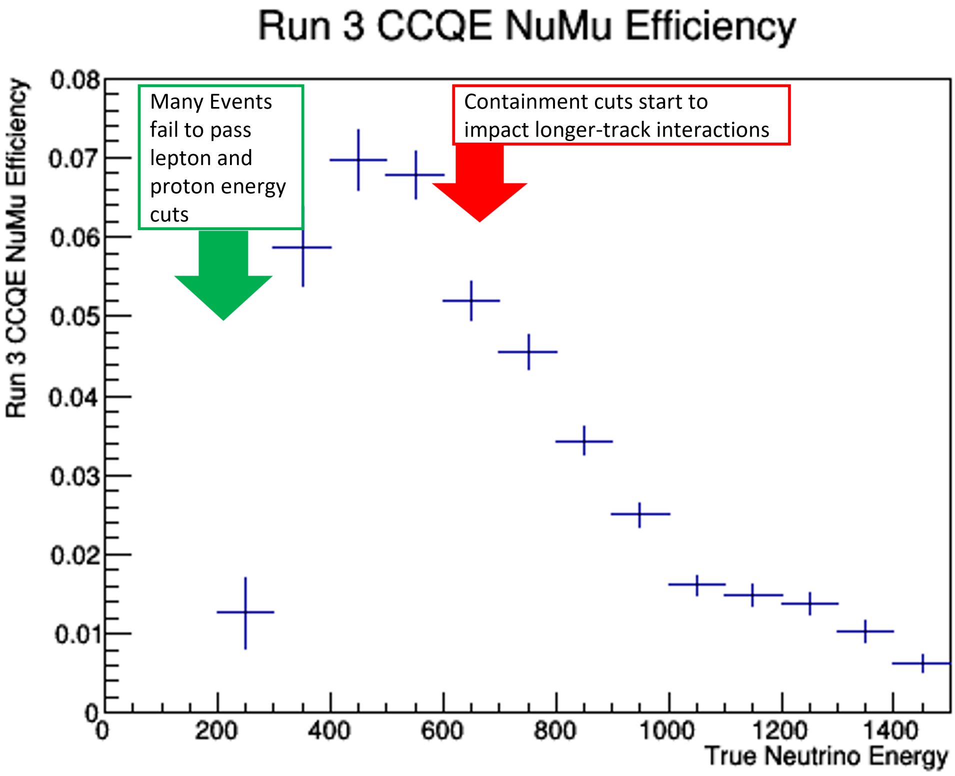


Figure 6.10: The efficiency of selecting CCQE $1\mu1p$ events, annotated to explain the major features. The integrated efficiency across this energy range is 0.0327.

6.4.4 $1\mu1p$ BDT Ensemble Run 2 / Run 3 Comparison

Here, we examine a check on the performance of selection across different run periods. For this section, we compare the $1\mu1p$ selection as applied to the Run 2 period with the Run 3 period. This check serves to confirm that there are no major changes in our selection in data as we move through the data-taking time period.

Figures 6.11 and 6.12 show this data-to-data comparison for the neutrino energy, our analysis-bin variable, and the transverse momentum ratio, our BDT ensemble's most importance input variable.

The events in both samples are scaled to the POT of the Run 2 sample for better matching and the statistical uncertainties have been scaled to compensate. In these figures we can note that the shape and scale of the selections across the two different run periods match well.

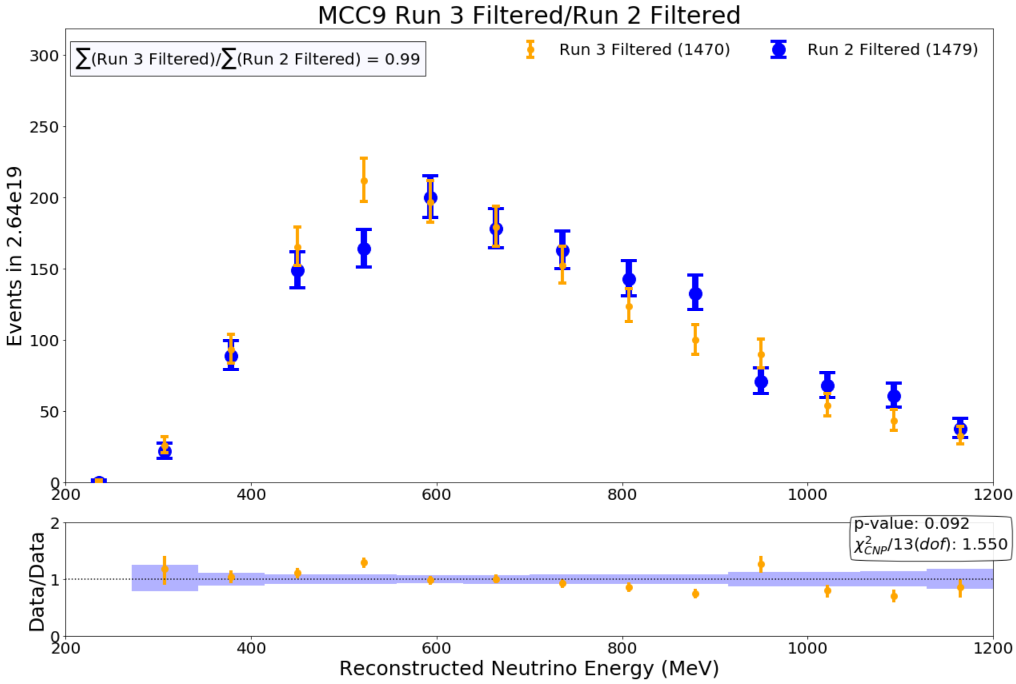


Figure 6.11: The data-to-data comparison across Run 2 and Run 3 in reconstructed neutrino energy.

6.4.5 $1\mu 1p$ BDT Ensemble Inter-Run Compatibility

In this cross-check, we seek to further validate the ability of our BDT to classify events beyond its training set. Specifically, we demonstrate more clearly that our ensembles are not overfit.

As discussed previously, the $1\mu 1p$ selection uses a different BDT ensemble for each Run period 1-3. In this section, we apply the Run 3 BDT Ensemble to the BNB overlay simulation for all three run periods, and compare the resulting POT-scaled selections. We use only the BNB overlay simulation sample for ease, though stress that as shown in figure 6.7, almost 98% of the selection comes from this sample at

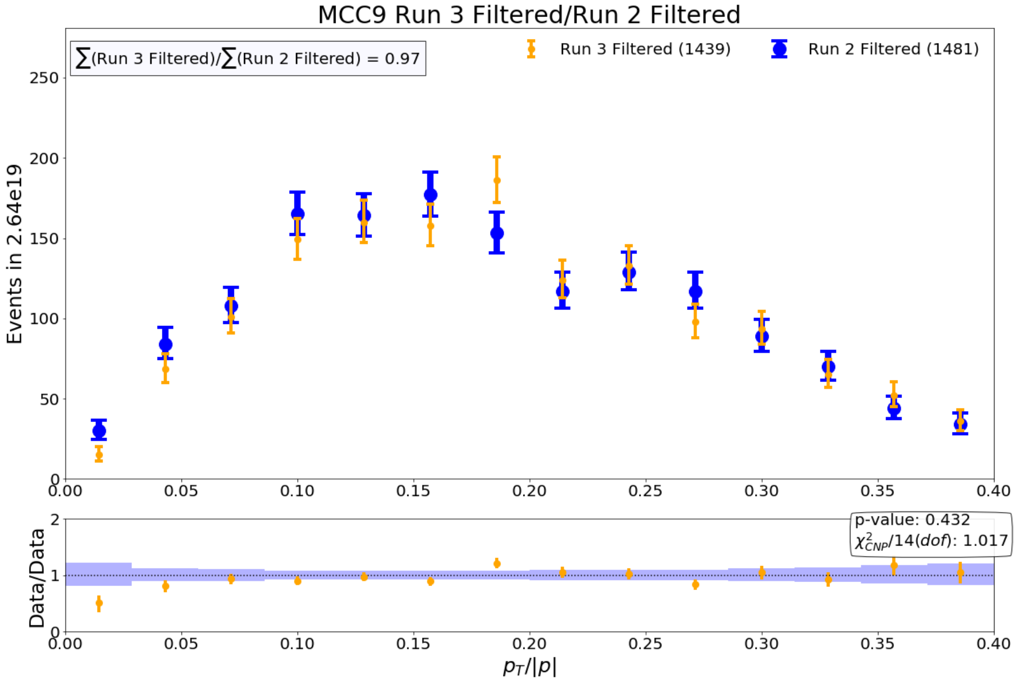


Figure 6.12: The data-to-data comparison across Run 2 and Run 3 in reconstructed neutrino energy.

the final selection stage.

In figure 6.13, the inter-run comparison for the BNB overlay samples in Runs 1-3 when always using the Run 3 BDT ensemble. Each sample has been normalized to the same $2.6e20$ POT. Figure 6.14 shows this same test, but instead ran on data instead of BNB overlay.

Based on these plots, we can conclude that the performance of the BDT ensemble does not swing significantly when applied to the different Run periods, and as such, the BDT ensemble doesn't show signs of being overfitted, though the Run 1 sample does stray briefly just above 400 MeV.

6.5 $1e1p$ Selection

As the work of the author and this thesis more closely surrounds the $1\mu1p$ selection, less detail is afforded to the $1e1p$ selection. However, in this section, some of the key differences between it and the $1\mu1p$ selection are outlined. Where not stated, the $1e1p$ selection mirrors the $1\mu1p$. However, first it is important to note

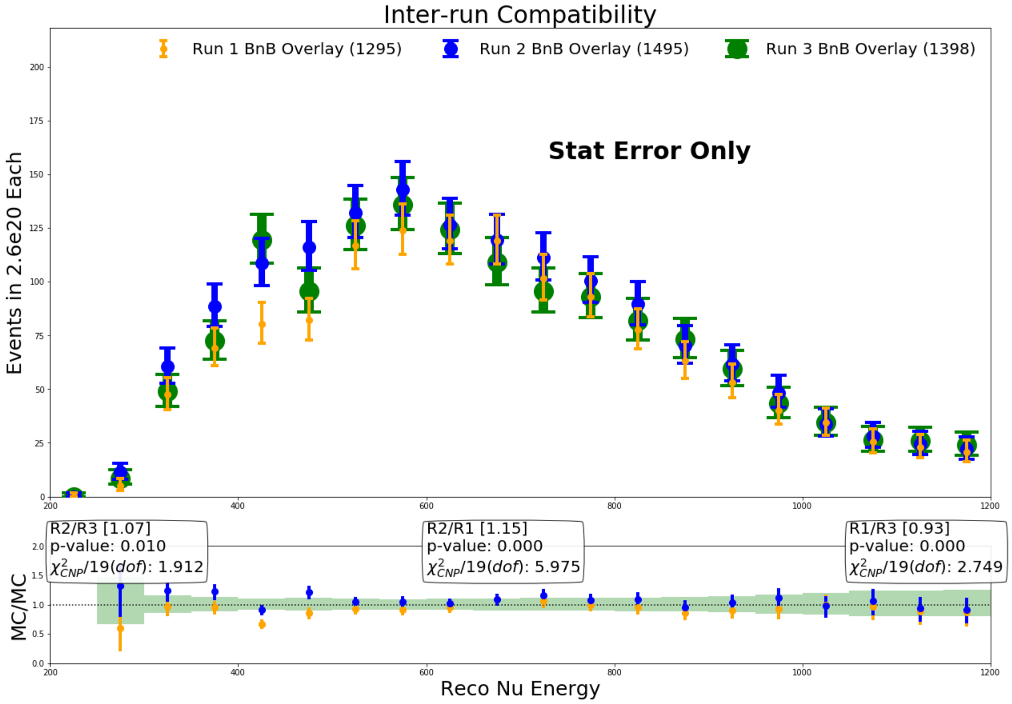


Figure 6.13: The comparison in BNB overlay for reconstructed neutrino energy, when using the Run 3 BDT ensemble for all run periods.

that the $1e1p$ selection is working with a much smaller set of potential events. As shown in section 3.2.2, the ν_e component of the BNB is much smaller than the ν_μ component. So not only is the $1e1p$ selection contending with the same backgrounds as the $1\mu1p$ the $1e1p$ selection also has to remove the $1\mu1p$ events from the selection.

The strong handle on $1e1p$ events is the showering electron. At modest to higher energies the shower topology is quite different. However as the interaction energy is dialed down, the shower's 'spray' of charge begins to look more track-like.

In order to benefit from this showering aspect of the $1e1p$ interaction, several additional selection requirements are introduced, as well as additional BDT variables used.

Of the preselection cuts before the BDT ensemble is trained, one additional requirement is made of the reconstructed shower. This requirement requires energy consistency across the 3 wire-planes. Equation 6.4 defines the shower energy consistency, where E_u, E_v, E_y are the shower energies as reconstructed from each plane. Events are required to have $E^{frac}_{consist} < 200\%$. This is a relatively loose cut that removes mis-reconstructed events, and events where unresponsive regions cloud energy

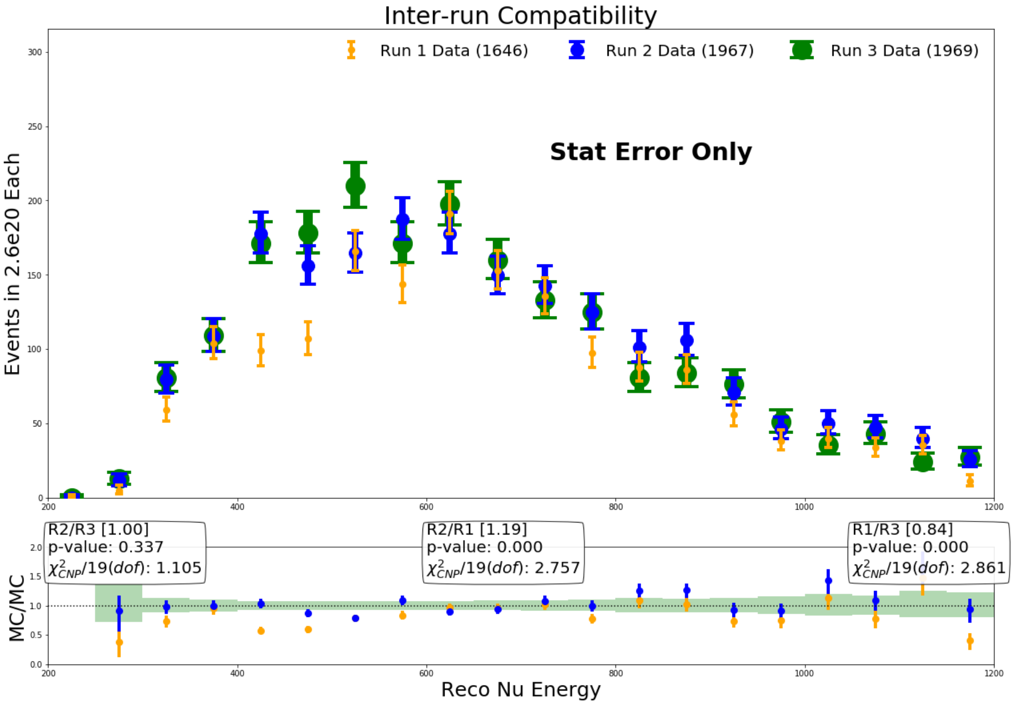


Figure 6.14: The comparison in data for reconstructed neutrino energy, when using the Run 3 BDT ensemble for all run periods.

reconstruction in one of the planes.

$$E^{frac}_{consist} = \sqrt{(E_y - E_v)^2 + (E_v - E_u)^2 + (E_u - E_y)^2)}/E_y \tag{6.4}$$

Then it is on to the BDT ensemble training. The $1e1p$ BDT ensemble consists of 20 BDTs per Run period, whose scores are averaged before a cut is introduced. In the case of the $1e1p$ selection, which is working with a much smaller signal to background ratio initially, a more stringent BDT score cut is used. The cut is still chosen to maximize sensitivity to an LEE search. The result is a requirement that all events have BDT score > 0.95.

After the BDT ensemble score cut, a few final background cuts are made to ensure the particles within the interaction are identified properly, similar to the $1\mu1p$ selection. However here, the purpose is to ensure that we have reconstructed an electron shower, not showers from γs coming out of ν_μ CCπ^0 interactions. The first step here is to attempt to reconstruct a test mass for the π^0. If this test mass is > 50 MeV, the event is rejected from the selection. The calculation and depiction

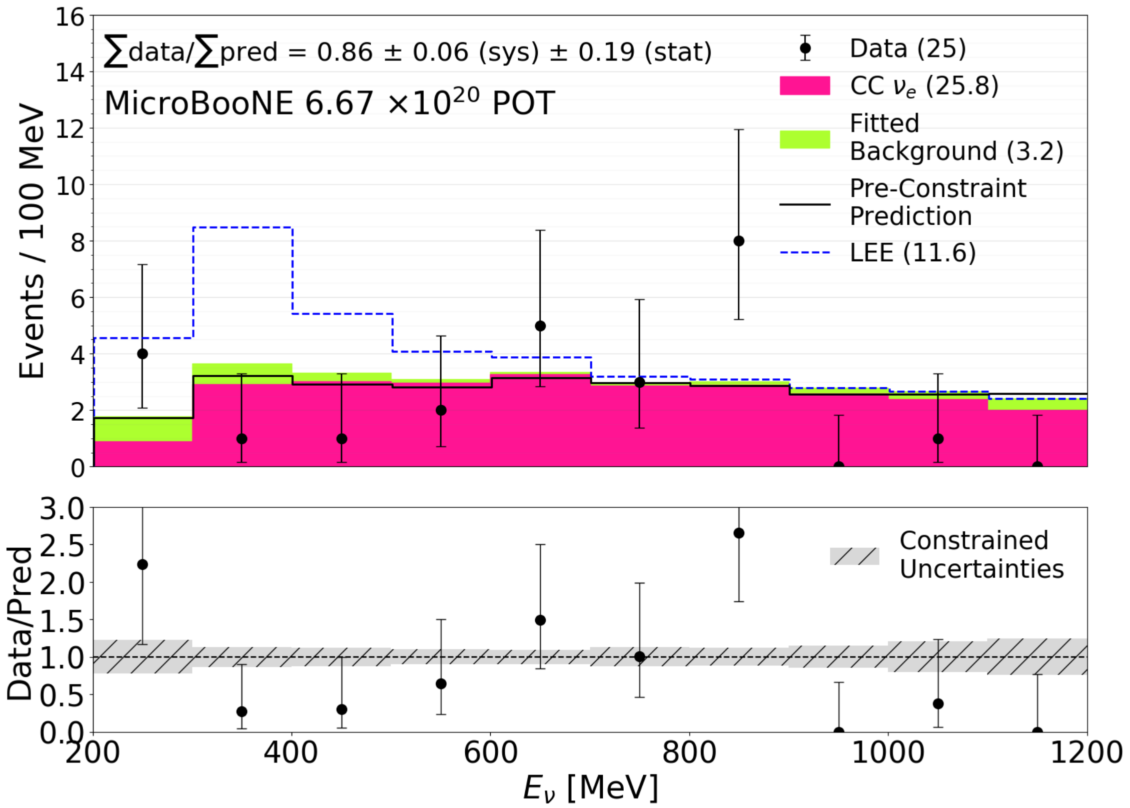


Figure 6.15: The constrained prediction for the ν_e signal and ν_μ background events. The MiniBooNE LEE signal expectation is included as a blue dashed line. [14].

of M_{π^0} is described in further detail in [14].

Next we turn to the MPID network once again., if the energy of the reconstructed electron is high enough, over 100 MeV, then a cut on the MPID muon score is invoked, removing events that the MPID network scores as having a muon at greater than 20%. Next, it is required that the MPID γ/e score ratio be < 2, to help ensure the shower present is from an electron not a γ.

The final $1e1p$ selection after these, and having undergone the $1\mu1p$ constraint analysis detailed in [14], is shown in figure 6.15.

Chapter 7

A Sterile Neutrino Search in MicroBooNE

In this chapter we search for a sterile neutrino existing in the 3+1 extended SM described in section 3.4.1. Our search uses the $1\mu 1p$ selection described in section 6.4, searching for muon neutrino disappearance. Recall that at the baseline of the MicroBooNE LArTPC we can employ the short-baseline approximation which allows us to ignore SM oscillations. This means any observed disappearance is expected to be caused by some new physics.

We begin with a description of our analysis' sources of systematic uncertainty, before moving into a description of our 3+1 search. We describe how an expectation is built under the conditions of different 3+1 models across our parameter space. Then our analysis' test statistic is defined and a 90% confidence level is drawn, determining MicroBooNE's sensitivity to these oscillations. Several validation studies are shown demonstrating our analysis' robustness, before finally a fit is performed on MicroBooNE data corresponding to 6.67×10^{20} POT taken during the first three run periods.

7.1 Systematic Uncertainties

The oscillation analysis contained within this thesis is limited by systematic uncertainties from a variety of sources. In this section we describe the methods used to determine these systematic uncertainties. For the purpose of this discussion we will split the different types of systematic uncertainty by whether they are reweightable or not.

Reweightable systematics come from sources that would alter the weighting assigned to different events used to build our expectation. For example, the systematic uncertainty associated with our flux represents an uncertainty associated with the shape and distribution of neutrinos received by the detector. If we wanted to check the effect of decreasing the muon neutrino flux in some specific energy range by 95%, we can 'reweight' muon neutrino events in that energy range to only count as 0.95 events each. These event weights can be changed, hence 'reweighting' the flux.

Meanwhile non-reweightable systematics are instead referencing uncertainty in parameters that serve to either introduce, or remove entire events from our selection in a binary way. An example here is the systematic associated with the LArTPC optical system, where now the uncertainty is meant to capture whether or not a specific event would have been able to pass the optical filter in our preselection discussed in section 6.3. If the light yield in the detector were increased, an event previously not passing our PMT requirements, could now shift into our selection. In this analysis, our non-reweightable systematics are all associated with the detector, and are thus termed the detector systematics.

The way systematic uncertainties are accounted for within our analysis in MicroBooNE is by building a covariance matrix from the various contributing systematics. This covariance matrix then represents the correlations from bin to bin across the analysis. The covariance matrix is defined by eq. 7.1. M_{ij} is the ij^{th} element of the covariance matrix. $\mathcal{N}$ is the number of different systematics considered. N_i^{CV} is the number of events in the i_{th} bin of the standard expectation, while N_i^k is the number of events in the i_{th} bin of the k-systematically-varied expectation.

$$M_{ij} = \frac{1}{\mathcal{N}}(N_i^{CV} - N_i^k)(N_j^{CV} - N_j^k), \tag{7.1}$$

Then, the total systematic covariance matrix is determined by summing the individual matrices associated with the different systematic components. However, when visualizing this, or scaling it to a different expectation, it beneficial to represent the covariance matrix in its fractional form. The fractional covariance matrix, F, is defined as shown in eq. 7.2.

$$F_{ij} = \frac{M_{ij}}{N_i^{CV} N_j^{CV}}, \tag{7.2}$$

The total *fractional* covariance matrix is shown below in figure 7.1, while the scaled covariance matrix is shown later in figure 7.7. Each element in this scaled matrix represents the covariance between the elements in energy bin i and energy bin j. For elements along the diagonal, where $i = j$ this then represents the variance, which is the squared standard deviation of the events in said bin. Whereas the fractional matrices divide out the number of events in the energy bins, per equation 7.2. This results in the scaled covariance matrix appearing very different from the fractional forms, because when scaling the fractional terms up by our expectation, the bins with large fractional covariance (and variance) contain a small number of events, and ultimately do not produce large values in the scaled covariance matrix.

7.1.1 Detector Systematics

The systematic uncertainties associated with the LArTPC detector are non-reweightable. The adjustments to these uncertainties do not adjust an event's weighting, they adjust the reconstruction in such a way that a given event may shift outside of the selection. Recall the example that a decrease in the light yield within the LArTPC could lead to an event no longer passing the optical system cuts, causing it to fall out of a selection. This either happens or does not happen,

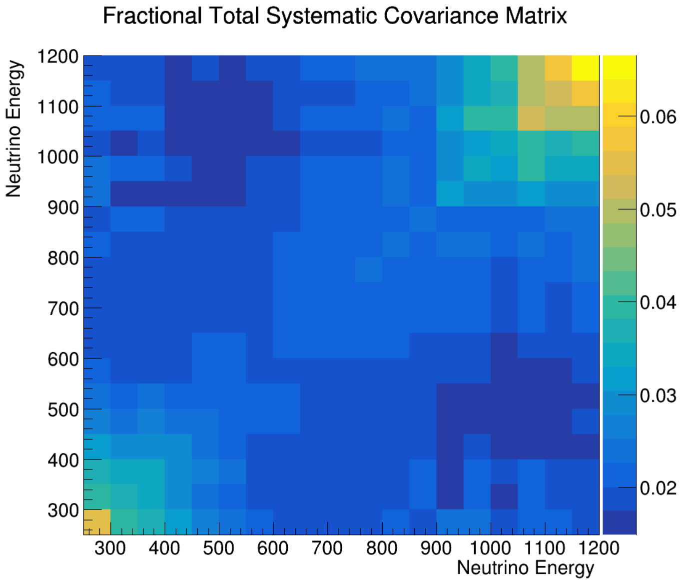


Figure 7.1: The total, fractional systematic covariance matrix for the $1\mu 1p$ analysis.

there can be no partial events.

The systematic uncertainties associated with various detector properties are determined by first creating a set of simulated samples where the properties of the LArTPC are adjusted. These modifications are made to a variety of different aspects of the detector, such as the amplitude and width of the waveforms detected on the wires as a function of the x-coordinate, y and z coordinates, and the detector angles θ_{XZ} and θ_{YZ} of a particle's trajectory [70]. Detector variations also include changing the parameters associated with electron-ion recombination, changing the electric field mapping in the detector, as well as changing the light yield, attentuation, and the Rayleigh scattering length [14].

This breadth of simulated samples can then be compared to the standard simulation for our analyses in order to estimate the various detector systematic uncertainties. However, creating so many different simulated samples requires substantial computational power, and as such the size of the different detector variation samples is limited. This introduces statistical noise into the detector systematic uncertainties. In order to counter this, we leverage a kernel-density-estimation (KDE) algorithm [71] to smooth the expected spectrum for our different detector variation samples.

This assigns a 'kernel' waveform with some width to each event, and then smooths the spectrum by summing these waveforms for each event. The KDE smoothing is valid because we expect the underlying distribution of events in our variables (such as neutrino energy) to be smooth within our analysis windows. For further details surrounding KDE smoothing, and the detector systematics used in this analysis, see [14]. The sum of all the detector systematics is shown in fractional form in figure 7.2.

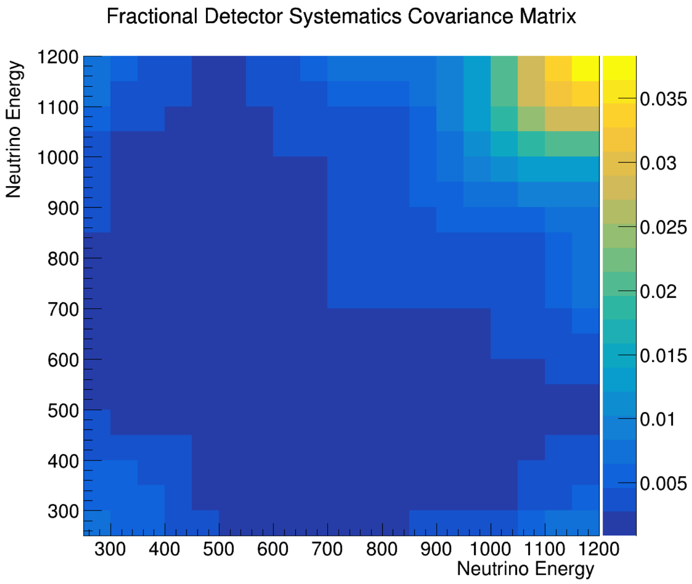


Figure 7.2: The fractional systematic covariance matrix for all the detector systematics in the $1\mu1p$ analysis.

7.1.2 Reweightable Systematics

The MicroBooNE collaboration considers the systematic uncertainty arises from several different reweightable sources. They are the systematic uncertainty associated with the flux, the neutrino cross section on argon, and the finally those associated with hadron re-interactions. In order to determine the uncertainty associated with these different reweightables, events in our simulation are assigned a weight based on their truth-level information, then the parameter associated with a given reweightable is adjusted and the event weights recalculated, the resulting shifts in the simulated spectrum of events then provide the uncertainty.

Our systematic uncertainty on the neutrino's cross-section with argon is determined by tweaking parameters in GENIE [57] dictating the weights of the simulated event spectra. These knobs refer to the strength of different interaction channels available to the neutrino interaction with the argon nucleus. Several of these knobs deal with the cross-section strength of our neutrino interaction backgrounds described in section 6.4.1.

Recall the primary two neutrino interaction backgrounds are charged-current resonant pion interactions and meson exchange current interactions. Both of these interaction modes will be better measured, and their cross-sections better constrained as MicroBooNE as well as future experiments begin to release detailed cross-section measurements. Our third neutrino interaction background, deep inelastic scattering is also present in our cross-section uncertainty. This will be better constrained as neutrino measurements are made with higher-energy neutrino scattering, such as using the neutrino main injection (NuMI) beamline at Fermilab [72] which reaches higher neutrino energies which correspondingly allow for better production of DIS interactions.

The procedure for handling the cross-section reweightable is described further in [65] and [73]. The fractional covariance matrix associated with the cross section systematic uncertainty is shown in fractional form in figure 7.3.

The uncertainty associated with the neutrino flux is related to the hadrons that decay in the decay pipe that create the BNB, which is described in section 3.2. Recall that the hadrons we are concerned with are $\pi^{\pm}$, $K^{\pm}$, and K_L^0. Specifically, these flux uncertainties are tied to how these hadrons are produced by protons striking the beryllium target, how they can re-interact within the target, and how they maneuver about the horn before reaching the decay pipe. The process by which the flux and its uncertainty are determined are discussed more fully by the MiniBooNE collaboration in [8], then modified slightly for the MicroBooNE collaboration in [74]. The fractional covariance matrix associated with the flux systematic uncertainty is shown in fractional form in figure 7.4.

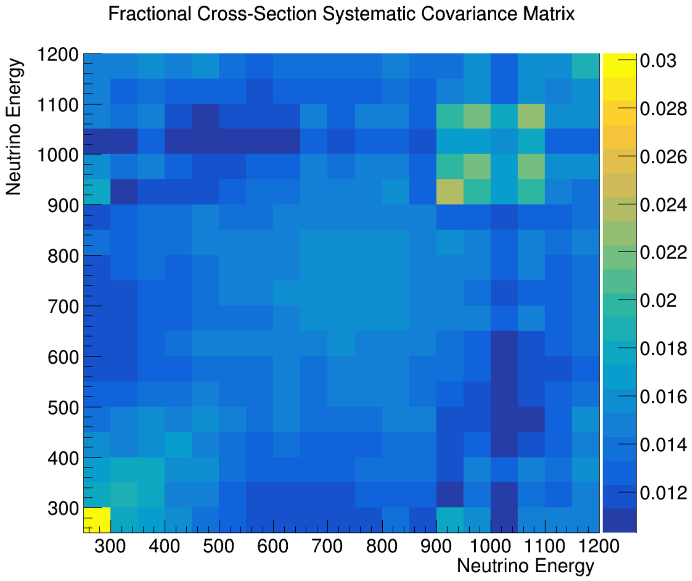


Figure 7.3: The fractional systematic covariance matrix for the cross-section uncertainty in the $1\mu 1p$ analysis.

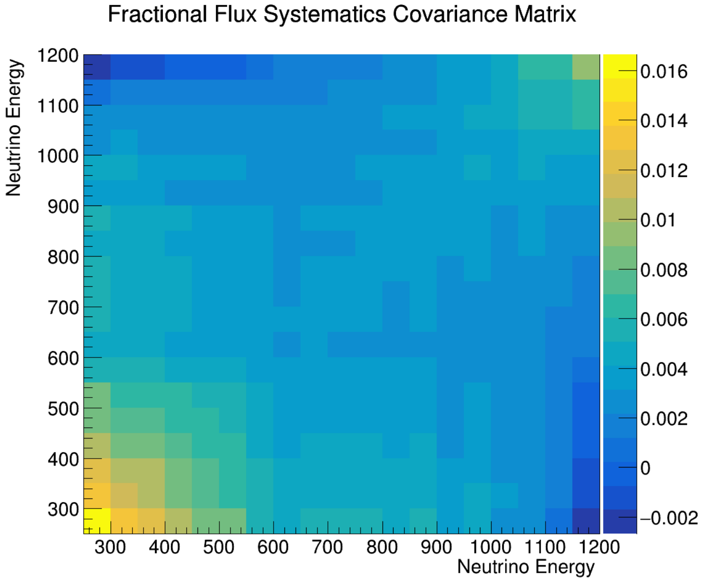


Figure 7.4: The fractional systematic covariance matrix for the flux uncertainty in the $1\mu 1p$ analysis.

The final type of reweightable systematic considered by this analysis is that of hadron re-interaction. This systematic considers the uncertainty associated with proton and $\pi^{\pm}$ particles being created then re-interacting with argon. These uncertainties are determined by reweighting events using Geant4 truth information concerning the trajectories of hadrons after they leave the original argon nucleus [75]. The hadron re-interaction systematic is of very low significance to this analysis when compared to the other sources of uncertainty. The fractional covariance matrix associated with these re-interaction uncertainties are shown in fractional form in figure 7.5, we note that these uncertainties are an order of magnitude less important than the other contributions and are shown for completeness.

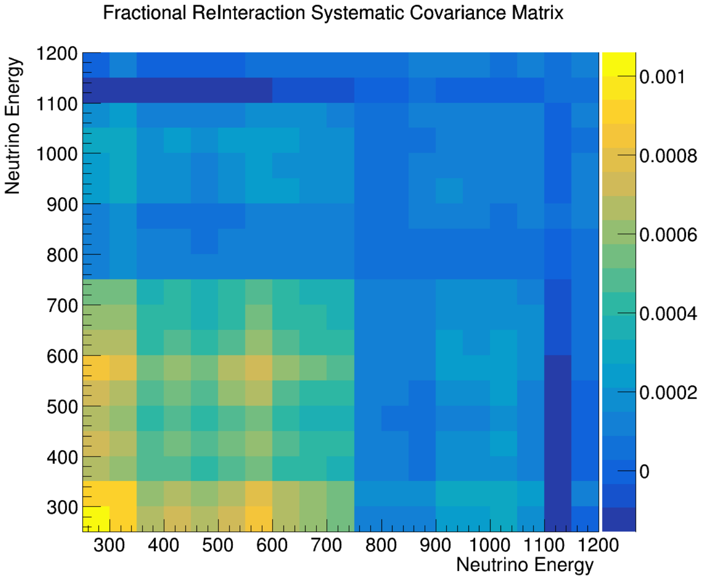


Figure 7.5: The fractional systematic covariance matrix for the re-interaction uncertainty in the $1\mu1p$ analysis.

7.1.3 Normalization Uncertainty

Now that we have our systematic uncertainties defined we can determine our analysis' normalization uncertainty. The fractional normalization uncertainty of the $1\mu1p$ selection is equivalent to the square root of the one-bin fractional systematic covariance matrix. This defined as

$$f^2 = \frac{\sum_{ij}^{Bins} M_{ij}^{sys}}{\left(\sum_{i}^{Bins} N_i\right)^2}. \tag{7.3}$$

The total normalization uncertainty is shown in table 7.1. The individual components of the systematic covariance matrix are also shown by isolating their components of the total fractional covariance matrix. Uncertainties associated with our classification tools, such as our BDTs, and convolutional neural networks, are assumed to be insignificant compared to these major sources of analysis uncertainty, particularly after the individual validation procedures performed for each tool as have been described.

Uncertainty Source	Normalization Uncertainty
Flux	6.3%
Cross-Section	11.8%
Re-Interaction	1.7%
Detector Systematics	4.9%
Total Systematic	14.34%

Table 7.1: Contributions of each source of systematic uncertainty in MicroBooNE, as used to plot the reconstructed energy distribution in Figure 6.7, to normalization uncertainty. The shape component of each systematic is removed according to Equation 7.3.

The contributions of each source of systematic uncertainty are also broken down across our 19-bin $1\mu1p$ selection. This breakdown is shown in figure 7.6. The plot also features the statistical uncertainty, though this is not used in calculating the total systematics curve, and is minor compared to the total systematic uncertainty.

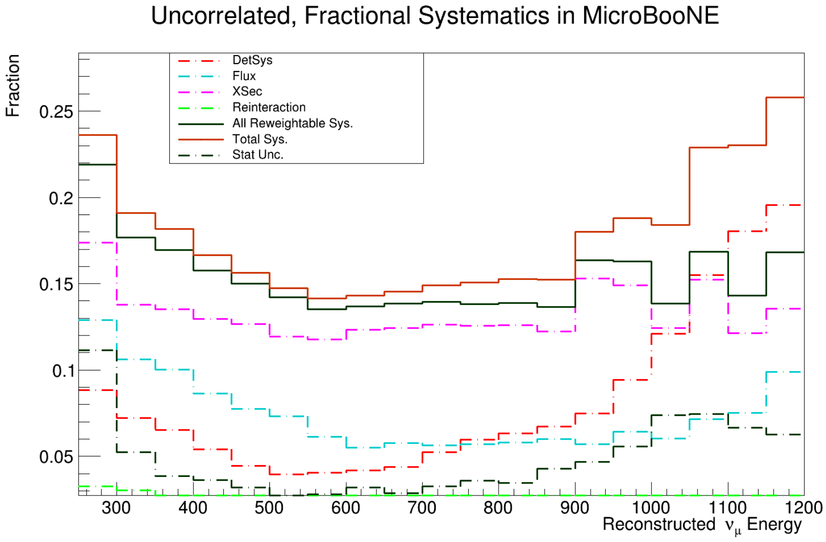


Figure 7.6: The uncorrelated fractional systematics broken down by type. The statistical uncertainty is also shown for comparison.

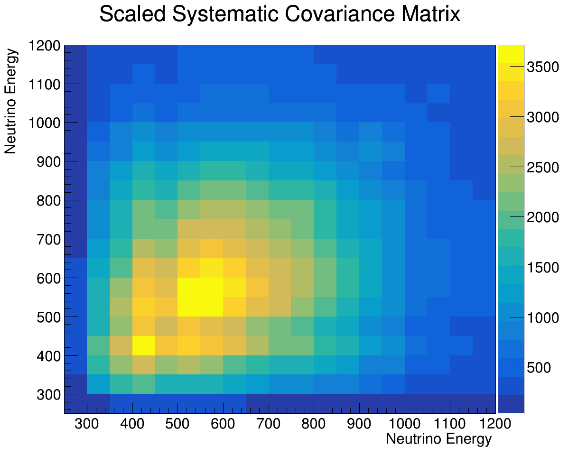


Figure 7.7: The full, systematic covariance matrix for the $1\mu 1p$ analysis, scaled to the null disappearance model's expectation.

7.2 3+1 Sterile Search in MicroBooNE

In this section, we perform a search across a 3+1 Sterile Neutrino Model space using MicroBooNE's $1\mu 1p$ selection described in section 6.4. We compare the results of this selection to the expected selection spectra modified by ν_μ disappearance across a grid search of sterile neutrino oscillation parameters. Using the uncertainties surrounding the selection we then outline a region of model parameters that are allowable at a 90% confidence level.

Recall that a sterile neutrino present in a 3+1 neutrino model would not couple to the weak interaction, and therefore could not be directly detected. However, they could affect neutrino oscillation probabilities to such an extent that the observable ν_μ flux in MicroBooNE becomes altered in a noticeable way, as described in eqn. 3.2 under the short-baseline approximation. Following the analogous manipulations of the PMNS matrix shown in section 2.3.2, we can rewrite this survival probability equation instead as

$$P_{\nu_\alpha \to \nu_\alpha} = 1 - \sin^2 2\theta_{\alpha\alpha} \sin^2(1.27 \Delta m^2_{41} L/E), \tag{7.4}$$

where the $4(1 - |U_{\alpha 4}|^2)|U_{\alpha 4}|^2$ has been replaced with its corresponding mixing angle term $\sin^2 2\theta_{\alpha\alpha}$. Within this equation we see the probability that ν_α doesn't disappear starts at 1, and decreases as the sine terms increase. This means that the sine terms correspond to the neutrinos disappearing to other flavors. In this equation, L is the neutrino propagation distance in km. It represents the distance between the point in the decay pipe that created the neutrino and the neutrino interaction vertex in the detector, as defined in the simulation. Meanwhile E is the neutrino energy in GeV, and $\sin^2 2\theta$ and Δm^2 are the two parameters of neutrino oscillation which define a given 3+1 sterile neutrino model. Using this equation, we can disappear the expected neutrino flux via different 3+1 models by reweighting the simulated ν_μ events by their survival probability, and then test how these new disappeared expectations best fit the data.

While MicroBooNE lacks the statistical power that will exist in future experiments such as will be achieved in the Short Baseline Neutrino program's SBND at Fermilab, the DL LEE analysis is sufficiently mature as to provide a highly pure selection of muon neutrino events ($\sim 98\%$). Note that the events need not be CCQE $1\mu 1p$ in order to disappear, they only need to be muon neutrino interactions. This selection can be used to perform a 3+1 sterile neutrino model ν_μ disappearance search across a spread of the model parameters $\sin^2 2\theta_{\mu\mu}$ and Δm^2_{41}. This modifies equation 7.4 for the survival probability of muon neutrinos to be

$$P_{\nu_\mu \to \nu_\mu} = 1 - \sin^2 2\theta_{\mu\mu} \sin^2(1.27 \Delta m^2_{41} L/E). \tag{7.5}$$

We note that from this point on in this analysis, we will refer to these model parameters as $\sin^2 2\theta$ and Δm^2 where the subscripts are implicitly referring to the sterile neutrino oscillation parameters associated with ν_μ disappearance.

Further, while the excesses observed in MiniBooNE [7] and LSND [16] can be attributed to ν_e appearance associated with a 3+1 sterile neutrino model, such a phenomenon would require accompanying ν_μ disappearance. These disappearing muon neutrinos would be what is turning into the ν_e excess. But presently, ν_μ disappearance has not been observed experimentally.[1] This lack of ν_μ disappearance

---

[1] Recall we are referring to specifically ν_μ disappearance as the result of a sterile, at a short-baseline where 3-neutrino oscillations are not active.

places tension on a 3+1 model explaining the observed excesses.

This search also provides another demonstration of analysis that can be performed using LArTPC technology, and MicroBooNE's analysis tools. However, we stress that the $1\mu 1p$ ν_μ CCQE event selection was designed and implemented with the goal of constraining the ν_e LEE analysis, not for this 3+1 disappearance search.

7.2.1 Method of Disappearance Search

Throughout this ν_μ disappearance search we will test the fitness of many different 3+1 sterile neutrino models to MicroBooNE data. We perform a 25×25 grid search over our two parameters $\sin^2 2\theta$ and Δm^2, with points logarithmically spaced with $\sin^2 2\theta_{\mu\mu} \in [0.01, 1]$ and $\Delta m^2 \in [0.01, 100]$ eV^2. Table 7.2 shows the 25 bin center values of $\sin^2 2\theta$ and Δm^2 used in this grid search. Note that as described in eq. 7.4 the $\sin^2 2\theta_{\mu\mu}$ term describes the oscillation amplitude, and the Δm^2 term corresponds to the oscillation frequency.

We perform this disappearance search using the DL LEE selection of $1\mu 1p$ charged-current quasi-elastic (CCQE) selection across reconstructed neutrino energy in figure 6.7. Note, that of the neutrino background category shown, only 2.3 events are expected to be ν_e, while the remainder are a form of ν_μ background to the CCQE signal desired by the CCQE selection. For this disappearance study, that ν_μ background will disappear just as the ν_μ CCQE events, resulting in a very pure selection for this analysis. We also exclude the first bin, from 200 to 250 MeV which selects no events in the $1\mu 1p$ selection.

We also wish to emphasize that while figure 6.7 includes MicroBooNE data, this disappearance analysis has avoided running the data through its machinery until the analysis scaffolding was vetted by the collaboration and solidified. In practice, because the MicroBooNE collaboration unblinded its data for the $1\mu 1p$ selection in order to complete the DLLEE analysis the authors were aware of the distribution of the data, and the lack of any extreme peculiarities, but said data was not used during the analysis' formation. This is in accordance with the MicroBooNE Blinding

Table 7.2: The bin centers for our two parameters .

Index	$\Delta m^2 eV^2$	$\sin^2 2\theta$
1	0.012	0.011
2	0.017	0.013
3	0.025	0.016
4	0.036	0.019
5	0.052	0.023
6	0.076	0.028
7	0.110	0.033
8	0.158	0.040
9	0.229	0.048
10	0.331	0.058
11	0.479	0.069
12	0.692	0.083
13	1.000	0.100
14	1.445	0.120
15	2.089	0.145
16	3.020	0.174
17	4.365	0.209
18	6.310	0.251
19	9.120	0.302
20	13.183	0.363
21	19.055	0.437
22	27.542	0.525
23	39.811	0.631
24	57.544	0.759
25	83.176	0.912

Policy outlined in section 3.5.

7.2.2 Creating an Oscillated Prediction with MicroBooNE Simulation

In figure 6.7, we show a predicted spectrum of events selected by the $1\mu 1p$ ν_μ CCQE selection. This predicted spectrum is determined through the use of a variety of simulated neutrino event samples, as well as data taken while the neutrino beam is off to estimate cosmic-ray-muon-only event backgrounds. Every event in the simulated portion of our predicted spectrum has a true (as simulated) neutrino energy E_{true} and similarly a true distance from the neutrino's creation at the BNB source L_{true}.

Then, given a pair of oscillation parameters, $\Theta = (\sin^2 2\theta_{\mu\mu}$ and $\Delta m^2)$ each ν_μ event acquires a survival probability

$$P_{\mu\to\mu}(\Theta; E_{true}, L_{true}), \tag{7.6}$$

where $P_{\mu\to\mu}$ is the 3+1 ν_μ disappearance probability from Equation 7.4. In order to determine the expected event spectrum for a given 3+1 model, each simulated muon neutrino event is weighted by this survival probability, using the true, simulated neutrino path length and energy. This allows us to calculate the adjusted expectation, under this new disappearance model.. Figure 7.8 shows a cartoon of this disappearance effect. The black curve shows the original expectation, the red curve shows the portion of events that disappear or $P_{\mu\to\mu}$ multiplied by the original flux, and the blue curve shows the modified spectrum after disappearance beneath the original spectrum. Note that the flux shown is not the BNB flux and designed for illustrative purposes.

Similarly, figure 7.9 shows the same cartoon, with the only difference being that the bin width is widened to 50 MeV, our analysis binning. This demonstrates the change in coarseness of the disappearance effect placed on our spectrum.

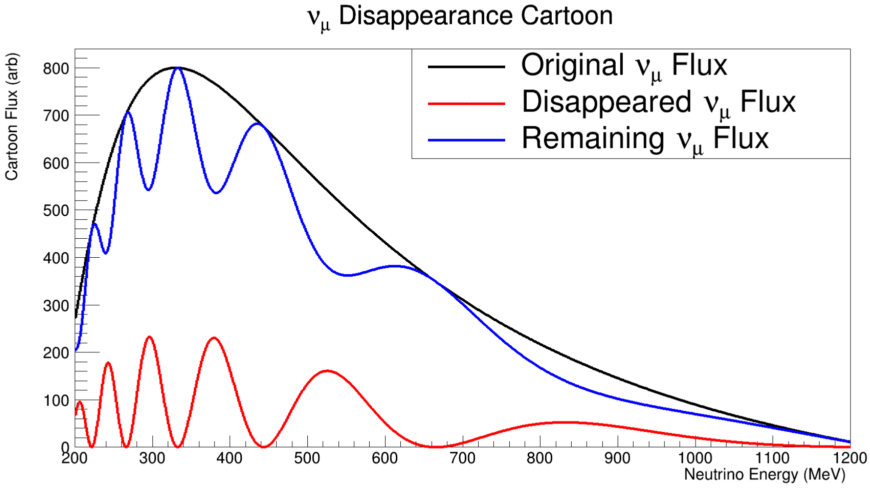


Figure 7.8: A cartoon of the method by which one builds a ν_μ disappearance spectrum from an initial prediction under the 3ν hypothesis.

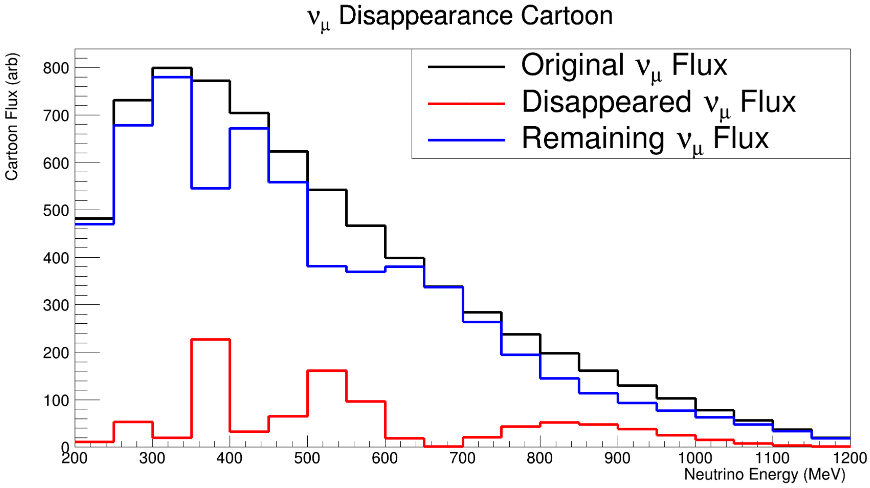


Figure 7.9: Another cartoon of ν_μ disappearance, however now using 50-MeV-wide bins for a coarser spectrum.

We emphasize that these disappearance effects are based on the simulated true neutrino energy, and not the reconstructed neutrino energy observed by analyzers.

7.2.3 Visualizing Disappearance Model Spectra

Here we examine various different spectra resulting from our different disappearance models. Figure 7.10 shows the resulting spectra from our 25 different values of Δm^2, all with our maximum value of $\sin^2 2\theta$. These are the expected spectra from the rightmost column of bins in our various grid plots.

It is useful to look specifically at the spectra for mass grid point indices around

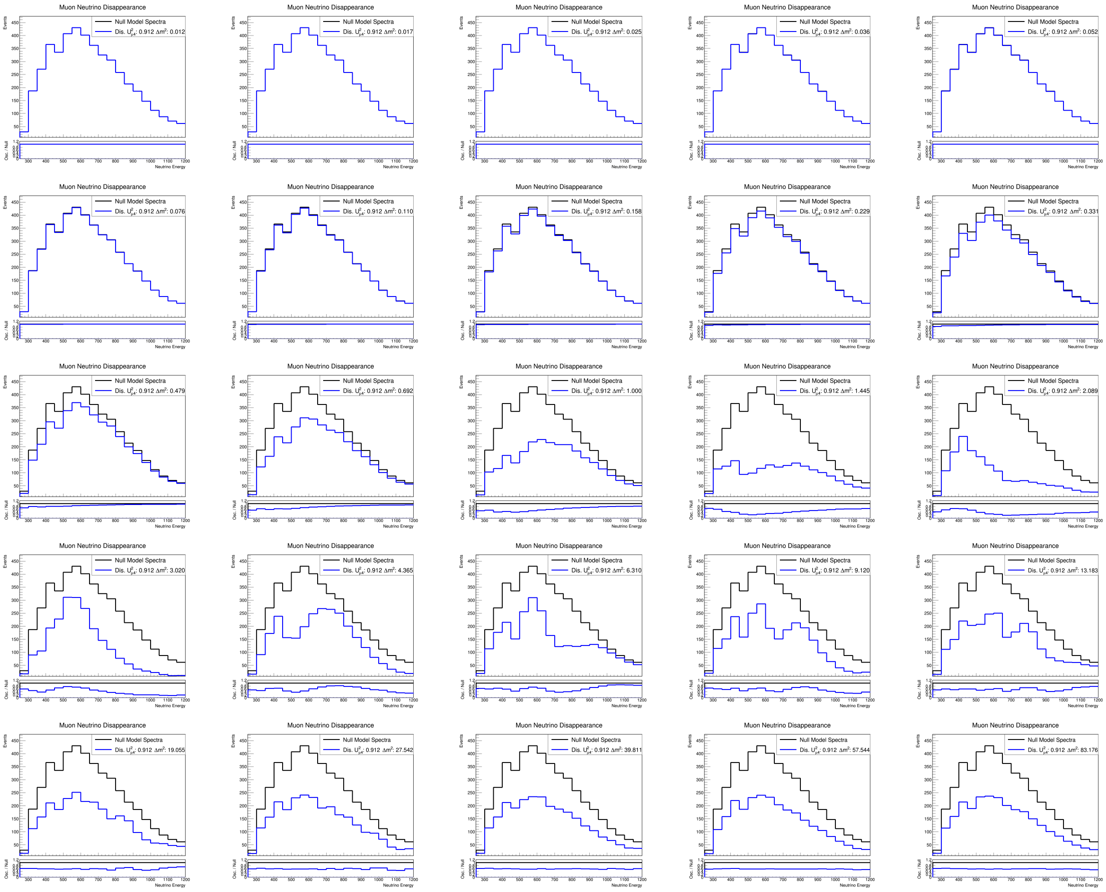


Figure 7.10: For all 25 values of Δm^2, the resulting spectra after disappearance is shown against the null model spectrum. As $\sin^2 2\theta$ is an amplitude, the maximum grid value is used for each plot to maximize the disappearance effect.

13-18. This is a region of very high disappearance, and the spectra become extremely difficult to confuse with the null model when accounting for our uncertainties. When evaluating our test statistic below, we can expect to see tension between this region and the surrounding model phase space.

For low values of Δm^2, the disappearance effect becomes unnoticeable, even at maximum $\sin^2 2\theta$. It isn't until mass grid point index of around 9, or $\Delta m^2 \approx 0.3$ that we can visually start to notice an effect, and even then the fluctuation is almost certainly covered by our various uncertainties.

This grouping of spectra that look almost exactly like the null model also extends into the higher Δm^2 region if the $\sin^2 2\theta$ parameter is low enough. Figure 7.11 shows an example of such a spectrum, with grid index coordinates of (12,19) this is one of our middling $\sin^2 2\theta$ grid points, and higher Δm^2 grid points, but the disappearance effect is certainly covered by our covariance matrix.

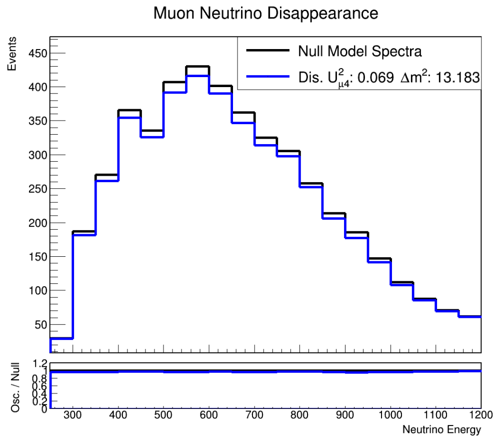


Figure 7.11: An example of a higher Δm^2, but low $\sin^2 2\theta$ disappeared spectrum that is still very close to the null model spectrum. .

7.2.4 Calculating Our Test Statistics

In order to determine both MicroBooNE's sensitivity and the eventual allowed regions for a 3+1 model fit to MicroBooNE data, we require a definition of our test statistic. This statistic will allow us to compare the goodness-of-fit between two different spectra, a given oscillated expectation and an observation. We begin with a typical definition of likelihood

$$L(\Theta) = \frac{1}{(2\pi)^{n/2}\sqrt{|M|}} e^{-\frac{1}{2}\chi^2(\Theta)} \tag{7.7}$$

.

Where Θ represents the 3+1 neutrino model we wish to test, M is the covariance matrix of the *predicted* spectrum including statistical uncertainties, systematic uncertainties, and bin-to-bin systematic covariances. Finally, $\chi^2(\Theta)$ is defined as by

$$\chi^2(\Theta) = \sum_i^{bins} (N_i^{obs} - N_i^{pred}(\Theta)) M_{ij}^{-1} (N_j^{obs} - N_j^{pred}(\Theta)). \tag{7.8}$$

N_i^{obs} is the number of events in the ith bin of the observed neutrino spectrum, and $N^{pred}(\Theta)$ is the predicted spectrum under the sterile neutrino hypothesis described by oscillation parameters Θ. Our χ^2 calculation uses a combined Neyman-Pearson chi-square term added along the diagonal of the covariance matrix. [76] . We then take the negative-log-likelihood

$$-ln(L) = \frac{1}{2}\chi^2(\Theta) + \frac{1}{2}ln(|2\pi M|) \tag{7.9}$$

.

Note, that the 2π has moved into the determinant, and lost its power of n accordingly. Recall that when a constant exists in a matrix, and the determinant is taken, the result features the constant to the power of the rank of the matrix.

Now, we are free to define our test statistic as two times the difference between two negative-log-likehoods. This can also be referred to as the negative-log-likelihood-ratio and is defined as

$$R \equiv -2ln(L_{PT}/L_{bf}) = \chi^2_{PT} + ln(|2\pi M_{PT}|) - \chi^2_{bf} - ln(|2\pi M_{bf}|). \tag{7.10}$$

This test metric, R, is now the difference between two comparisons with the observed spectrum. The two PT terms represent the comparison between the observed spectrum and the expected spectrum of the test grid point's model. Meanwhile the two bf terms represent the minimum negative-log-likelihood comparison between the observed spectrum and all potential grid point models.

Said more explicitly, this means that given an observation, we calculate our test metric for model Θ_{PT} by calculating the negative-log-likelihood between the observation and expectation at Θ_{PT}, and then subtract the negative-log-likelihood minimum when comparing the observation and all Θ models in our search.

We note that the 'observed' spectrum is typically meant to represent data. However, in some instances, other generated spectrum are treated as data, such as when universes are thrown in the Feldman Cousins method discussed later, or when a simulated signal is injected to test our analysis machinery.

Drawing sensitivity curves at specific confidence levels using R as our test statistic is discussed in section 7.2.6

7.2.5 Theoretical Limit on ν_μ Disappearance Sensitivity

Now that we've determined the $1\mu1p$ analysis' normalization uncertainty we can examine a theoretical limit on ν_μ disappearance at high Δm^2. If we examine equation 7.4, relabeled for explicitly ν_μ disappearance, it becomes

$$P_{\nu_\mu \to \nu_\mu} = 1 - \sin^2 2\theta_{\mu\mu} \sin^2(1.27\Delta m^2_{41} L/E). \tag{7.11}$$

The frequency of oscillations is driven by Δm^2, and when high enough the oscillations occur so quickly that they rise and fall within a single energy bin. When this happens the $\sin^2(1.27\Delta m^2 L/E)$ term averages to $\frac{1}{2}$, and the magnitude of the disappearance effect is controlled solely by $\sin^2 2\theta_{\mu\mu}$ which sets the amplitude.[2]

This allows the normalization uncertainty to constrain our analysis' sensitivity to ν_μ disappearance at high Δm^2 through the equation

$$P_{\nu_\mu \to \nu_\mu} = \sin^2 2\theta_{\mu\mu} \sin^2(1.27\Delta m^2 L/E) < \chi^2_C f. \tag{7.12}$$

which comes from equations 4 and 5 in [43]. Here, χ^2_C is defined as the χ^2 for which $\alpha\%$ of χ^2 values fall below given the expected χ^2 distribution. After the elimination of the Δm^2 parameter within this limit, we expect a 1-degree-of-freedom χ^2 distribution. This affords a χ^2_C at 90% confidence of 1.28 via a 1-sided normal distribution [43]. Next we reduce the $\sin^2(1.27\Delta m^2 L/E)$ term to $\frac{1}{2}$ for the high frequency regime, and we get

$$\sin^2 2\theta_{\mu\mu} < 2\chi^2_C f. \tag{7.13}$$

Recall f is the normalization uncertainty of our $1\mu1p$ spectrum. Therefore, this analysis loses the ability to exclude ν_μ disappearance for models with high Δm^2

---

[2] This comes into effect when energy resolution is no longer meaningfully finer than $\frac{\pi}{1.27\Delta m^2 L}$, given by the period of 3+1 sterile neutrino oscillation at the short baseline.

when $\sin^2 2\theta_{\mu\mu} < 0.367$.

7.2.6 Drawing Confidence Levels

With the $1\mu1p$ selection and its corresponding systematic covariance matrix finalized, we can outline how this analysis determines sensitivity curves at a given confidence α using our test statistic R.

The goal of a sensitivity curve is to showcase the region of models in our 2-parameter space $(\sin^2 2\theta_{\mu\mu}, \Delta m^2)$ that could be excluded if the data produced by the experiment were to present itself exactly as null model expectation. This aims to answer the question of 'if the $1\mu1p$ data looks exactly as the three-neutrino SM expects, what ν_μ disappearance model phase space can be excluded at 90% confidence?'.

As stated in section 7.2.1, we search a 25×25 logarithmically spaced points. Each point represents a different pair of model parameters Θ, with a different expected, disappeared, spectrum. While it should be obvious that models exist with parameters set between the points on our grid search, this analysis will only search with the 25×25 granularity proposed.

Once given some observed spectrum, be it data, or a fake signal to determine a sensitivity curve, this observed spectrum can be compared to the disappeared expectation at each grid point model. This results in a grid of negative-log-likelihood values from equation 7.9 for our 25×25 models. Each value in the grid is that grid point model: Θ_{PT}. Sterile neutrino oscillation models with lower values indicate better fits to the observation, and higher values indicate poorer matches.

The model with the best fit negative-log-likelihood can then be used to calculate our test statistic R. We can determine a grid of R values by subtracting the best fit negative-log-likelihood value for model Θ_{bf} from the grid of PT values, this is the same as following equation 7.10. An alternative analysis could be performed using the null model as a baseline rather than the best-fitting model. This would effectively shift all values of R upward, including the calculations of R_C via the

Feldman Cousins method described below, whereas we use the best fitting model from those available to be set our baseline.

Now we can determine a sensitivity at some specific confidence interval by comparing the grid of R values to some critical threshold R_C using Wilks' Theorem. By taking the difference between the negative-log-likelihoods for Θ_{PT} and Θ_{bf} we expect the degrees of freedom associated with the bins to reduce away, leaving us with a 2-degree-of-freedom model, for the two parameters that make up Θ. Taking the χ^2 distribution for 2 degrees of freedom, we establish a R_C critical threshold at 90% confidence of 4.61.

Now, in order to create a sensitivity curve at 90% confidence using Wilks' Theorem, we draw a band around the region of parameter space where our grid of R values is greater than the R_C of 4.61. This region of phase space we expect to be excluded at 90% confidence if the observation were to look exactly as the null model's expectation.

This means that if our experiment were to be repeated many times over, we expect this region to be excluded 90% of the time.

The resulting sensitivity from this method using Wilk's Theorem is shown in figure 7.12, as well as the high Δm^2 limit. In addition, a single-bin sensitivity analysis is shown, where all the energy bins of the analysis are combined. This presents experimental evidence of what MicroBooNE's sensitivity would be in the absence of shape information, as the high Δm^2 limit theorizes. MiniBooNE's sensitivity is also overlaid [15], though the MiniBooNE analysis had use of roughly 40× the statistics of this analysis. We also note that the MiniBooNE disappearance analysis performed a shape-only fit, whereas our analysis uses both shape and rate information. The difference is that a shape-only fit scales the expected spectrum to match the observed spectrum, while subtracting the normalization component from the covariance matrix. Our procedure for determining the shape plus rate sensitivity curve is outlined in further detail in section 7.2.8.

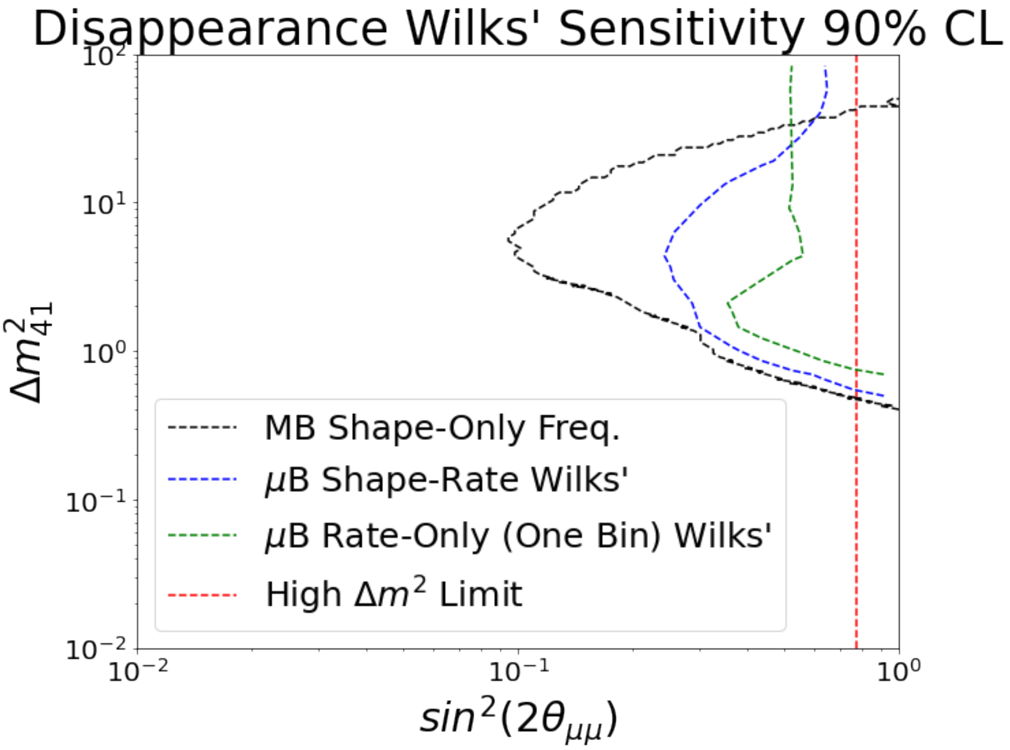


Figure 7.12: MicroBooNE's 90% sensitivity contours for a shape-and-rate (S+R) parameter-space scan. The contours are drawn assuming a R distribution with 2 degrees of freedom. MiniBooNE's ν_μ disappearance sensitivity is overlaid [15]. Additionally a vertical line at $\sin^2 2\theta_{\mu\mu} = 0.367$, the predicted high-$\Delta m^2$ sensitivity calculated from our normalization uncertainty. Finally the sensitivity for a 1-bin, rate-only disappearance analysis using Wilks' theorem is also shown.

7.2.7 Feldman Cousins Derivation of R_C

There are several points why Wilks' theorem may not hold for this analysis. Wilks' theorem assumes a specific number of degrees of freedom. As described in section 7.2.5, we assume a 2-degrees-of-freedom distribution, due to the two parameters that make up our disappearance model Θ. However, across our entire phase space, our signal will not be sensitive to both parameters.

First, our $\sin^2 2\theta_{\mu\mu}$ parameter controls the amplitude of the disappearance effect in equation 3.2. This term has to exist somewhere between 0 and 1, but for lower values it turns off any disappearance effect we hope to see, regardless of what value we use for Δm^2. This means that in the low-amplitude phase space, our model could behave more like it has 1 degree of freedom.

A different effect can be seen in very high values of Δm^2. As discussed in section

7.2.5, at higher values of Δm^2, the frequency term of our disappearance probability oscillates faster than the energy resolution in MicroBooNE, thus averaging to a value of $\frac{1}{2}$. Therefore at high values of Δm^2, we lose dependence on the Δm^2 parameter, and our model again could behave more like it has 1 degree of freedom.

While there may be other inconsistencies in the two degree of freedom assumption, these points alone are enough to indicate that the R value distributions are not Gaussian.[3] This means that it may be beneficial to develop an alternative set of critical R values, or R_C for each grid point in our search. We can do this by examining the R distribution determined by simulating or 'throwing' different observed universes from the grid point model's true oscillation parameters. This allows us to follow the procedure described by Feldman and Cousins [77] for determining our R_C values.

For our Feldman Cousins form of the analysis, we maintain the same parameter space of 25 × 25 logarithmically-spaced grid points across our two parameters $\sin^2 2\theta_{\mu\mu}$ and Δm^2.

For each grid point model, Θ_{PT} we determine the expected disappeared spectrum as described in section 7.2.2. Next, we can take advantage of a software package, SBNfit [78], to simulate fake universes or 'pseudo-experiments' coming from a specific disappearance model Θ_{PT}. The process for drawing pseudo-experiments is further described in section 7.2.9, for now, it is sufficient to understand that the pseudo-experiment spectrum is meant to represent a possible observation seen in a universe where the expectation is the disappeared spectrum at point Θ_{PT}.

For each grid point, we simulate 1000 pseudo-experiments to treat as our observation. Then for each observed spectrum, we can calculate our test statistic, R, by calculating the negative-log-likelihood of its originating model, Θ_{PT}, and subtracting the negateive-log-likelihood of the best-fitting model, Θ_{bf}. Note, that the covariance matrix at each grid point changes according to the newly determined expected spectrum for that grid point's disappearance model. This means that the

---

[3] However, the expected number of events in a given bin are still assumed to follow a multivariate Gaussian.

covariance matrix for Θ_{PT} and Θ_{bf} are only the same if the best-fit point is the very point the pseudo-experiment was thrown from. Now, at each grid point we have a distribution of 1000 R values as determined via our pseudo-experiments.

An example spectrum of R for $\Theta_T = (\sin^2 2\theta, \Delta m^2) = (0.01, 0.01eV^2)$ (the null point in our search) is plotted on the left in figure 7.13. Note, that while technically these parameter values are not a null-disappearance, they correspond to such a minuscule change, that we treat it as our null model. This resulting R distribution, shown as a blue histogram is overlaid with a theoretical χ^2 distribution of 2 degrees of freedom to illustrate the difference between the Wilks Theorem expectation and the Feldman Cousins method. Furthermore, the positions of the 90% confidence limits are marked with vertical lines, each one representing the point in its respective distribution where 90% of the distribution is less than that point.

The right plot in the figure demonstrates a more extreme disappearance model, with $\Theta_T = (\sin^2 2\theta, \Delta m^2) = (0.91, 3.0eV^2)$. Here we see that the R spectrum is spiked at 0, nearly all of the thrown universes' best fit models are the same as the model the universe was thrown from. In fact, as this happens more than 90% of the time, the R_C value for this grid point is 0.

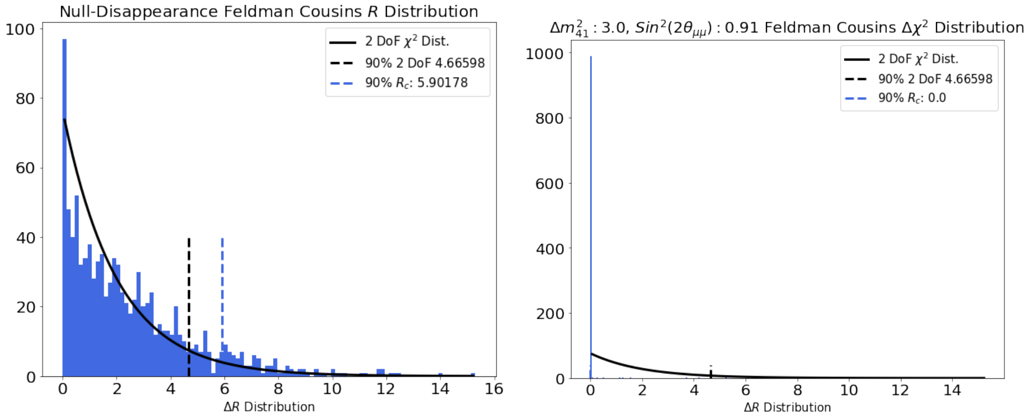


Figure 7.13: Distribution of R across 1000 pseudo-experiments for the null, no-disappearance, hypothesis as Θ_T, shown in blue on the left. Another distribution for 1000 pseudo-experiments for an oscillation hypothesis with a large disappearance model, with parameters Δm_{41} : 3.0, $\sin^2(2\theta_{\mu\mu})$: 0.91 is on the right. Each plot contains expected distributions for two degrees of freedom. Vertical lines are drawn for the 90% CL of each distribution, such that 90% of the distribution is to the left of the vertical line.

We note, that for extreme disappearance effects, as shown in the right distribution of R values the best-fit grid point (BF) is frequently the same as the point the pseudo-experiment was thrown from (PT), which means our test statistic R, by definition, is 0, this corresponds to regions with such extreme disappearance effects that MicroBooNE is particularly sensitive to them.

Now, for each grid point in our search, we can determine the R_C value in the Feldman Cousins method by finding the value where 90% of the R distribution falls below it, in our 1000 pseudo-experiment case this corresponds to the 900 lowest value. The grids of these R_C values, and their fractional difference with the 2-degrees-of-freedom Wilks' threshold of 4.61 are shown in figure 7.14.

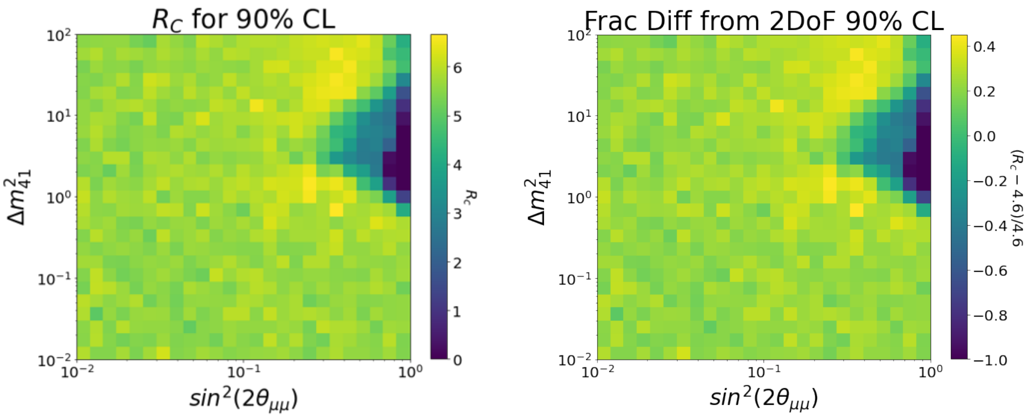


Figure 7.14: The critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario.

7.2.8 Quantifying Sensitivity with Feldman Cousins Method

Before this analysis is applied to MicroBooNE's actual data, it is valuable to determine what region of our 3+1 sterile neutrino model's phase space we will be sensitive to. For our sensitivity study we will treat the null disappearance model's spectrum as though it were our observation exactly, then run through the analysis to determine what part of our phase space we can exclude. This study answers the question: If MicroBooNE were to measure exactly the null disappearance spectrum, what oscillation parameters could we exclude to 90% confidence?

For each grid point we treat the null disappearance spectrum as observation, and then compare it to the expected disappearance spectrum of the corresponding grid point, this gives us a grid of negative-log-likelihood PT values. Then we subtract the minimum value from among this grid from each point in the grid. This gives us a grid of test R values to compare to the grid of R_C values developed via our Feldman Cousins pseudo-experiment throws. This follows the calculation as shown in equation 7.10. Note, that for the sensitivity determined via Wilks' method, shown in figure 7.12 these R test values were compared to a constant threshold of 4.61, for the 2 degrees-of-freedom assumption.

The grid of R test values are displayed on the left in figure 7.15. The same values are shown on the right plot, except that it uses a log scale on the Z-axis and values under 0.1 are set to 0.1 for better visualization. Here, we see that there are several disappearance models that have extreme tension with the null disappearance expectation being treated as observation. As a sanity check, this region of disagreement is in the same space where the R_C values were exactly 0 in figure 7.14. We also note the purple region of interest at high Δm^2 in the upper right portion of the grid, notable in the log-scaled . This corresponds to the region from which the minimum negative-log-likelihood comes from. Note that it is the sum of χ^2 term and determinant term in equation 7.9 that determines the minimum value, not just the χ^2, and in this region decreases in the determinant term offset increases in the χ^2 term. We recognize the fact that this represents the idea that our negative-log-likelihood-ratio test statistic is a biased estimator, biasing towards more disappearance. However, this minimum region fits squarely with the vast majority of degenerate portions of our plot where the R values are low, indicating little disagreement with the null, and a high degeneracy of the 'no disappearance' case.

With both the test values of R and the Feldman Cousins grid of R_C, we can now determine the sensitivity of the analysis within our parameter space. Figure 7.16 shows this Feldman-Cousins-based sensitivity.

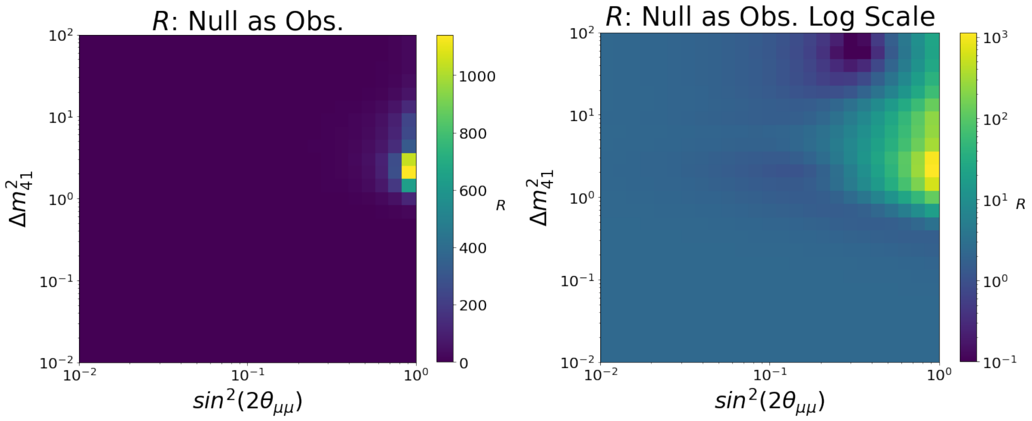


Figure 7.15: The test R values across every point in our parameter space. The left plot displays the raw R value, while the right shows the same values, with values below 0.1 set to 0.1 and a log scale applied for better visualization.

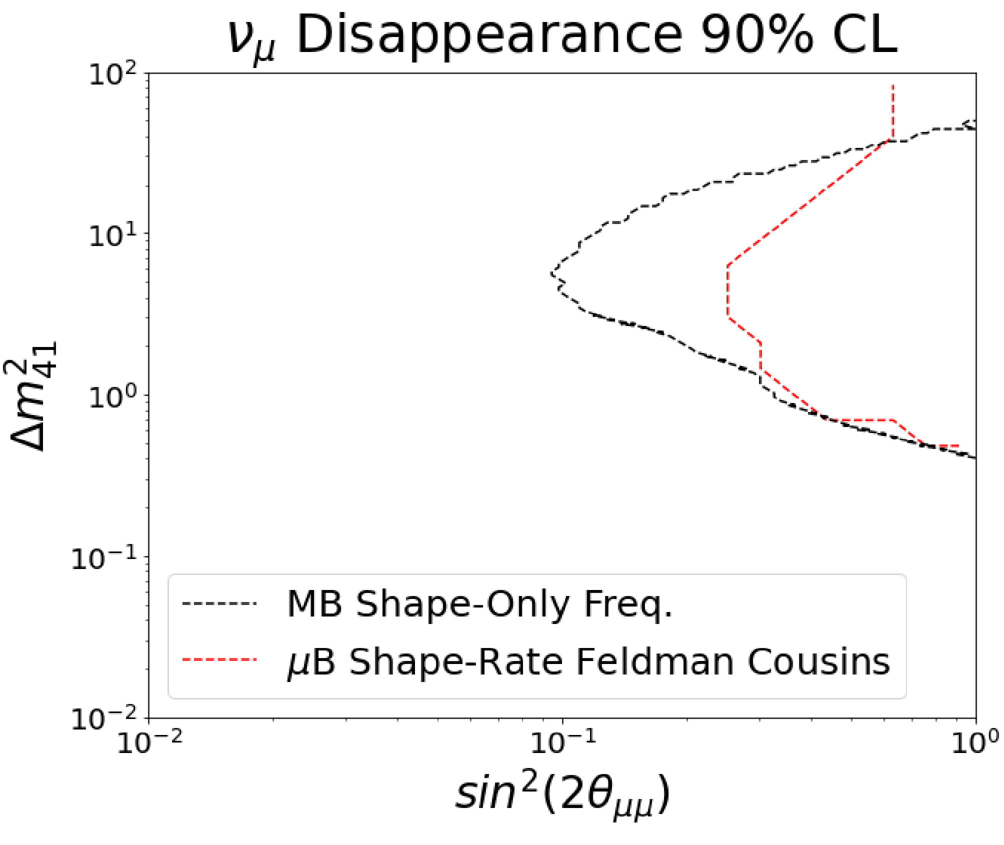


Figure 7.16: MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. MiniBooNE's ν_μ disappearance sensitivity is overlaid [15].

Notably, within the granularity of our 25 × 25 grid of values, we do not see a difference between the sensitivity using Wilks' Theorem or the Feldman Cousins method. We emphasize that, for the reasons provided at the beginning of section 7.2.7, the Feldman Cousins method is preferred. However, it should also be noted that it comes with a significantly increased computational cost due to the task of generating so many pseudo-experiments across the parameter space, and comparing them to all potential expected models. This makes Feldman Cousins procedure hard to scale as the model phase space grows, either by adding a free parameter to extend the model, or by decreasing the coarseness of our grid search points.

In figure 7.16 we note a region in the upper right of our search space which MicroBooNE is sensitive to that MiniBooNE was not. This is a region of strong disappearance with high frequency Δm^2 values, allowing for more rapid fluctuations in the disappeared spectrum. The MiniBooNE analysis was a shape-only analysis, ignoring any rate (normalization) difference. This combined with the high mass splitting limit reduces the information available to discriminate disappearance effects in the region compared to our shape and rate analysis.

7.2.9 Drawing Pseudo-Experiments with SBNfit

In reality, experiments only get one complete set of data with which to perform analysis. Their data is their data. It comes from the disappearance model that describes this universe. We want to build an expectation to that data, whatever it may be. However, our expectation exists as a result of the statistical and systematic uncertainties associated with the experiment and our simulated datasets.

These uncertainties serve to drive the expectation away from the true underlying physics, which create the data. We label our baseline expectation, without incorporating uncertainty the 'central value' (CV). But due to these uncertainties, the data observed from MicroBooNE will not exactly represent this CV expectation. By simulating pseudo-experiments we can gain insight into the variation that we may see between our single observation, and our CV expectation that is covered by our uncertainties.

The process for generating a pseudo-experiment's spectrum is rooted in Gaussian fluctuations. For each bin in our spectrum we draw randomly from a Gaussian centered on the CV expectation, with variance of the bin's statistical and systematic uncertainty. This follows the assumption that our expected distribution is a multivariate Gaussian and results in a spectrum systematically fluctuated around the CV expectation. But we still need to account for statistical fluctuation. In order to do that, we then Poisson smear each of the bins of our expectation to simulate the resulting pseudo-experiment's observation. This pseudo-experiment now represents what MicroBooNE could have observed in another universe due to our systematic and statistical uncertainties.

One of the aspects of the $1\mu 1p$ selection in MicroBooNE is that it features many bin-bin correlations in its covariance matrix. This means that when systematically fluctuating the CV, a random Gaussian draw will not cover these bin-bin correlations. Our software package, SBNfit, uses Cholesky decomposition to correctly draw random spectra from the CV using a covariance matrix with bin-bin correlcations. For further details about the Cholesky procedure, see [79].

7.2.10 Pseudo-Experiment Cross-Check

We can perform a visual check on our pseudo-experiment throwing procedure to confirm that the universes are being drawn appropriately. We choose a specific, stronger disappearance spectrum, corresponding to values of $\Delta m^2_{41} = 0.69$ eV^2 and $\sin^2 2\theta_{\mu\mu} = 0.76$. We then generate 1000 pseudo-experiments around its disappeared CV expectation using our covariance matrix scaled to the disappeared CV expectation.

Figure 7.17 shows the resulting 1000 distributions. The original disappeared CV expectation is plotted as a red line at the center of the distribution. One, two, and three σ bands are plotted around this distribution in black. These bands are calculated by taking the CV and adding (or subtracting) the square root of the diagonal of the scaled covariance matrix including both systematic and Poisson statistical uncertainties. Note that this means the bands ignore bin-bin effects coming from the off-diagonal terms of the covariance matrix, but still serve as a easily-visualized benchmark. The 1000 pseudo-experiment spectra are shown in various different colors.

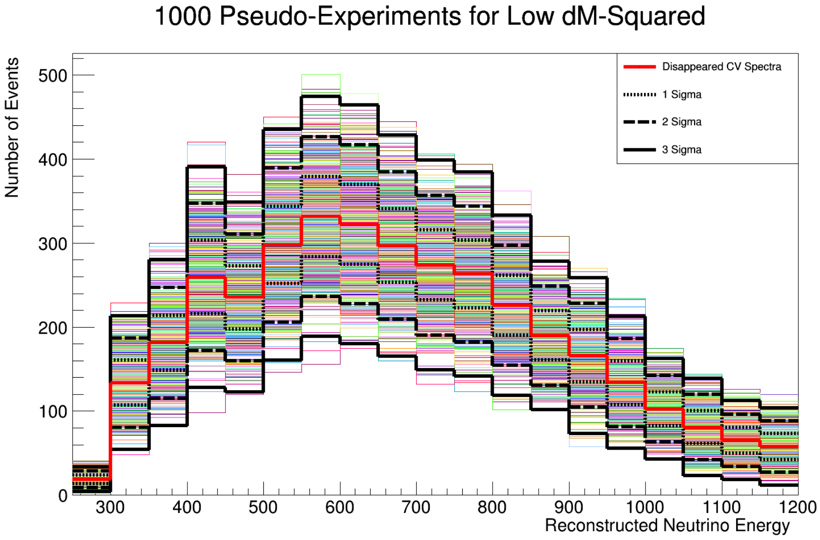


Figure 7.17: Plot of simulated events with ν_μ flux oscillated away with parameters $\sin^2 2\theta_{\mu\mu} = 0.76$ and $\Delta m^2_{41} = 0.69$ eV^2 (in red). The other colored histograms represent 1000 pseudo-experiments drawn around this central value. Overlaid are the 1, 2, and 3σ uncertainties from a covariance matrix scaled to the disappeared CV (and with Poisson uncertainty) for each bin.

We note that the vast majority of thrown bins end up within the 3σ band, with more concentration closer to the disappeared CV expectation. This is precisely what we expect from our pseudo-experiments. Given that 3σ is expected to cover 99.7%, and that over 1000 pseudo-experiments this test has fluctuated 19000 bins, we are unconcerned by some parts of some spectra reaching past the final band.

7.3 Validation Studies

Here, we'll go through a few studies performed to test different parts of the analysis.

7.3.1 Signal Injection Tests

Our sensitivity contour in figure 7.16 tells us what MicroBooNE could say if they observed exactly the 3-neutrino null disappearance expectation. But we can also examine the results of this analysis if its observation were to come from some true disappearance model. Here, we ask the question of 'if the $1\mu 1p$ data comes from **a pseudo-experiment, generated around some disappearance model**, what fraction of the time can we exclude the null disappearance model at 90% confidence?' We emphasize that this is different from the sensitivity test, as now the observation is coming from a thrown universe around an injected disappearance model, rather than the null model exactly.

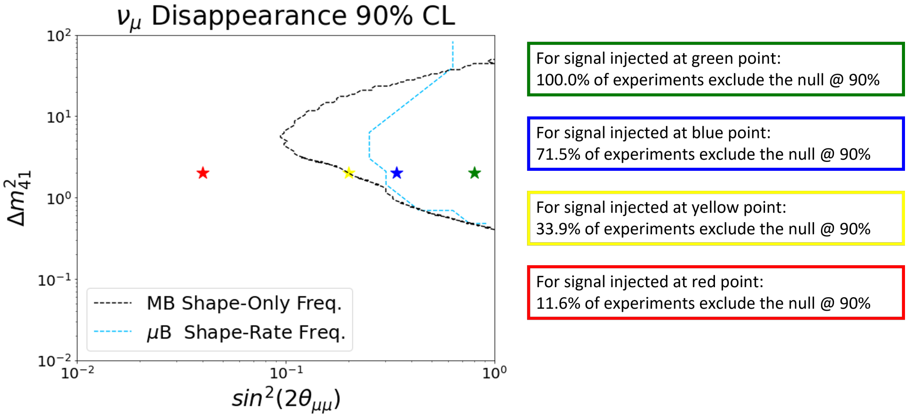
We create this injected signal to treat as observation by first choosing the two parameters that describe our disappearance model Θ. We then apply the disappearance effect to the null expectation. Finally we generate a pseudo-experiment using the covariance matrix for this disappeared expectation. We then perform a fit, exactly as though this pseudo-experiment spectrum were our data, calculating a grid of our test statistics R to compare to the Feldman Cousins R_C values to determine allowed regions, and specifically note whether the null model is allowed.

We generate and fit 1000 pseudo-experiments to get an idea of what fraction of the time MicroBooNE could reject the null.

We perform this signal injection test at four different disappearance model points. Each point has $\Delta m^2 = 2$ eV^2, while we vary our other parameter; $\sin^2 2\theta_{\mu\mu} = 0.04, 0.2, 0.34$, and 0.8.

Figure 7.18 shows the MicroBooNE Feldman Cousins sensitivity band, as well as the four points in our parameter space that we perform our injection test. We note that the points chose were not at the center of our model grid points, and are thus different models. Further, we emphasize that the pseudo-experiment procedure employed means that the final observed spectrum vary from the model's expectation as dictated by our covariance matrix.

The procedure to determine if the null model is allowed by a given pseudo-experiment is as follows. We calculate the negative-log-likelihood between the observed pseudo-experiment, and the expected null spectrum, then subtract the best fit negative-log-likelihood between the observed pseudo-experiment and all grid point expected spectrum. This results in an R value for the null model point, which we



For signal injected at green point:
100.0% of experiments exclude the null @ 90%

For signal injected at blue point:
71.5% of experiments exclude the null @ 90%

For signal injected at yellow point:
33.9% of experiments exclude the null @ 90%

For signal injected at red point:
11.6% of experiments exclude the null @ 90%

Figure 7.18: MicroBooNE's sensitivity to ν_μ disappearance at 90% confidence, with four points indicated where signal was injected for the crosschecks described in this chapter. Interpretations of the signal injection can be found in the text.

then compare to the R_C from our Feldman Cousins method. If $R > R_C$ then we exclude the null. This procedure can be repeated for all model grid points, not just the null thus determining all allowed regions give a specific observation.

We show the allowed regions for several pseudo-experiments in section 7.3.2.

Across our four different signal injection points, we have a variety of null model rejection rates.

- For a signal injected at the green point, $(\sin^2 2\theta, \Delta m^2) = (0.8, 2\text{eV}^2)$, we exclude the null model 100% of the time. This is an extremely disappeared spectrum, deep within the our sensitive parameter space, and thus unsurprising.

- For a signal injected at the blue point, $(\sin^2 2\theta, \Delta m^2) = (0.34, 2\text{eV}^2)$, we are able to exclude the null model 71.5% of the time. This is a notably less than 90%, despite the injection point being close to our 90% confidence sensitivity contour, however recall the injection test asks a different question than our sensitivity test.

- For a signal injected at the gold point, $(\sin^2 2\theta, \Delta m^2) = (0.2, 2\text{eV}^2)$ we begin to lose serious power to exclude the null. Now we only are able to exclude the null 33.9% of the time.

- For a signal injected at the red point, $(\sin^2 2\theta, \Delta m^2) = (0.04, 2\text{eV}^2)$, we see, and expect a significant drop in rejection rate. The disappearance effect here is quite small, with the amplitude term at 0.04 before considering the frequency term, effects will be slight, and more easily covered by our uncertainties. We exclude the null 11.6% of the time.

7.3.2 Pseudo-Experiment Allowed Regions

We can also examine the different allowed regions for pseudo-experiments generated as part of our injection test in section 7.3.1. Allowed regions are determined in the same way we determine if a given pseudo-experiment can exclude the null model, except now we examine each grid point, and draw contours around the region of model parameter phase space that are allowed when testing against the critical R values determined by our Feldman Cousins method.

Figure 7.19 shows the allowed regions for 20 different generated pseudo-experiments, five from each injection point. Each of the four columns of plots in the figure show pseudo-experiments thrown from the four different injection points. We note that for the final injection point, all five allowed region plots are the same. The best fit grid point is the same for each pseudo-experiment, and the allowed region is the same singular grid point. The pseudo-experiments chosen to display were picked at random (the first five pseudo-experiments generated at each point). In viewing these plots it is useful to recall that the plots shown are in logarithmic scale, so that half the **visual** phase space doesn't correspond to half the parameter phase space searched by our analysis.

We also note that the 90% threshold is not perfect, and are not surprised that between all these throws some best fit points lie outside the allowed region.

Examining the various allowed regions gives some insight into a large degeneracy within our two-parameter disappearance model phase space. Based on the allowed regions, there exist potential observed spectra that can reasonably fit (allow) disappeared model expectations with any Δm^2 and middling to lower $\sin^2 2\theta$ values, as

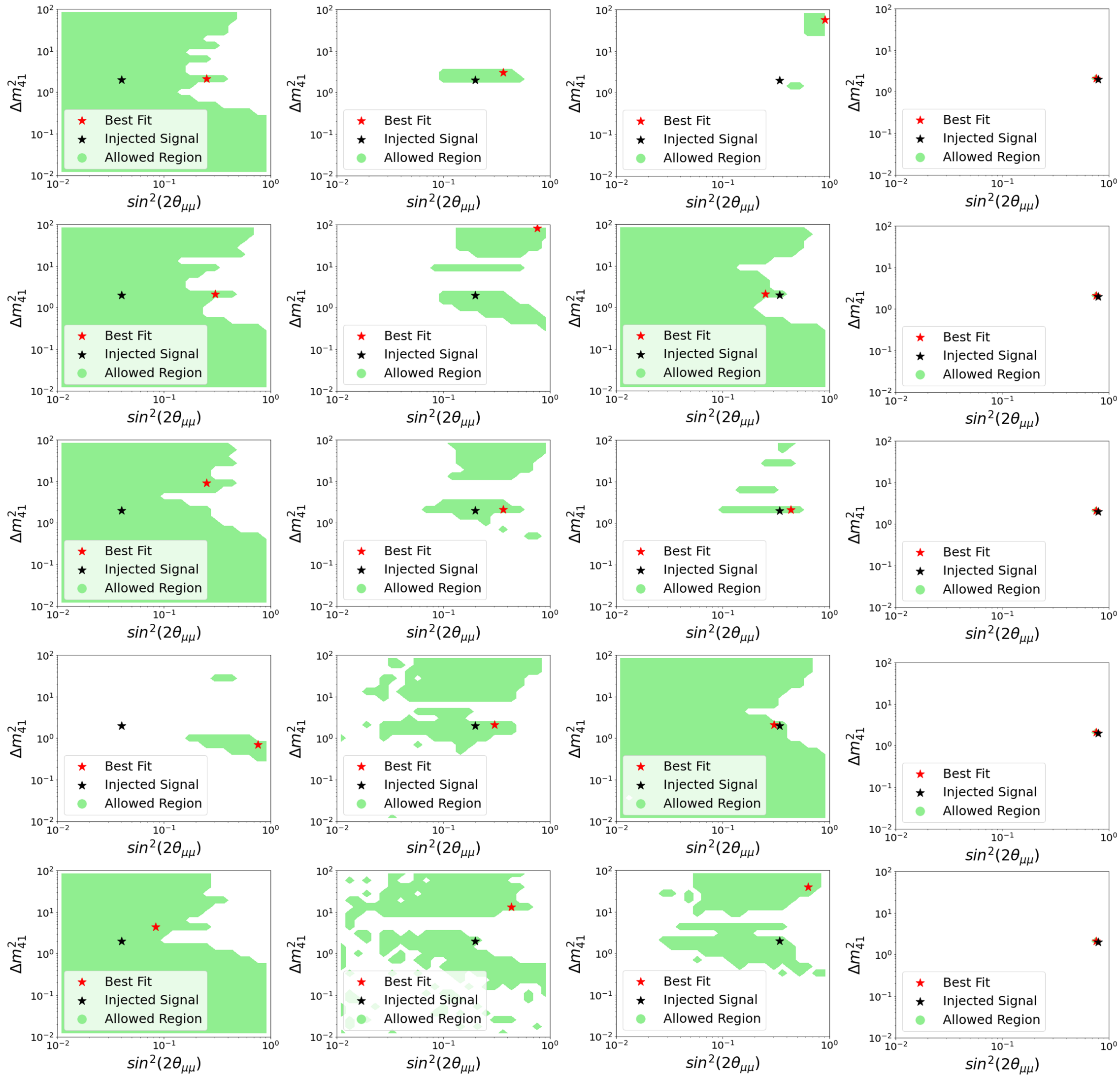


Figure 7.19: For various pseudo-experiments thrown from different injected signals we show the 90% confidence limit allowed regions.

well as fit models with middling to lower Δm^2 and any $\sin^2 2\theta$. We can better understand this aspect of the analysis by visualizing several of the different disappearance model spectra across different parts of our grid, which is done in the section 7.2.3.

7.3.3 Stats-Only Analysis

Due to the nature of the bin to bin correlations in our systematics matrix it can be difficult to anticipate the behavior of different steps in the analysis to determine if our machinery is behaving as expected.

This makes it valuable to repeat our analysis with a covariance matrix that only includes statistic uncertainty on the expectation, and sets all systematic uncertainties to 0. This removes any bin to bin effects, and means that in both throwing universes and our fitting procedure with the negative-log-likelihood, only statistical fluctuation is allowed.

It should be emphasized that the material in this subsection is only representative of what this disappearance analysis would look like if our experiment had no systematic uncertainty. Of course this is not the case, but it remains useful as a validation of our analysis.

Figure 7.20 shows our resulting covariance matrix with only statistical uncertainty. All off-diagonal elements are 0, indicating no bin-bin correlations.

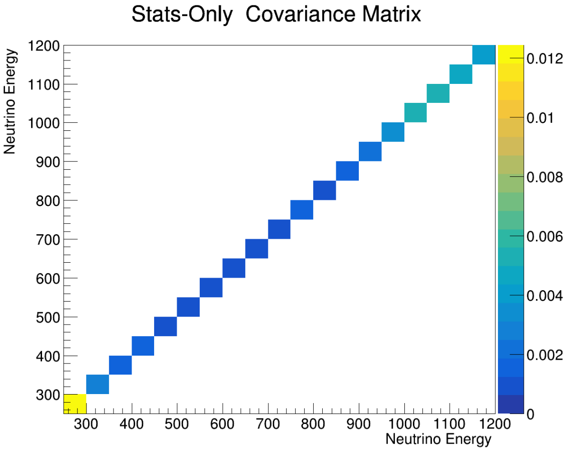


Figure 7.20: The fractional covariance matrix in a statistics-only regime.

Then we can run the Feldman-Cousins analysis to determine the new R_C thresholds for determining sensitivity, shown in figure 7.21.

In the stats-only regime, the phase space featuring $R_C = 0$ exactly grows, indicating that without the uncertainty associated with the experiment's systematics, it becomes easier to fit a thrown pseudo-experiment to its original model Θ_{PT}. Finally, we can determine the resulting sensitivity of a stats-only regime. This result is shown in figure 7.22. MiniBooNE's result is also shown, however it still includes systematic errors, and is only shown as a benchmark.

We see that in a world where MicroBooNE has only statistical uncertainty it becomes far more sensitive to ν_μ disappearance, able to exclude even lighter disap-

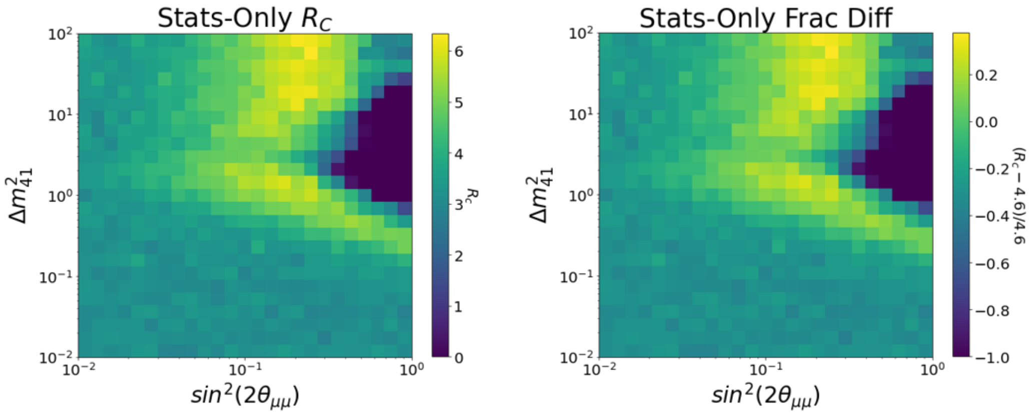


Figure 7.21: For the stats-only regime, the critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario.

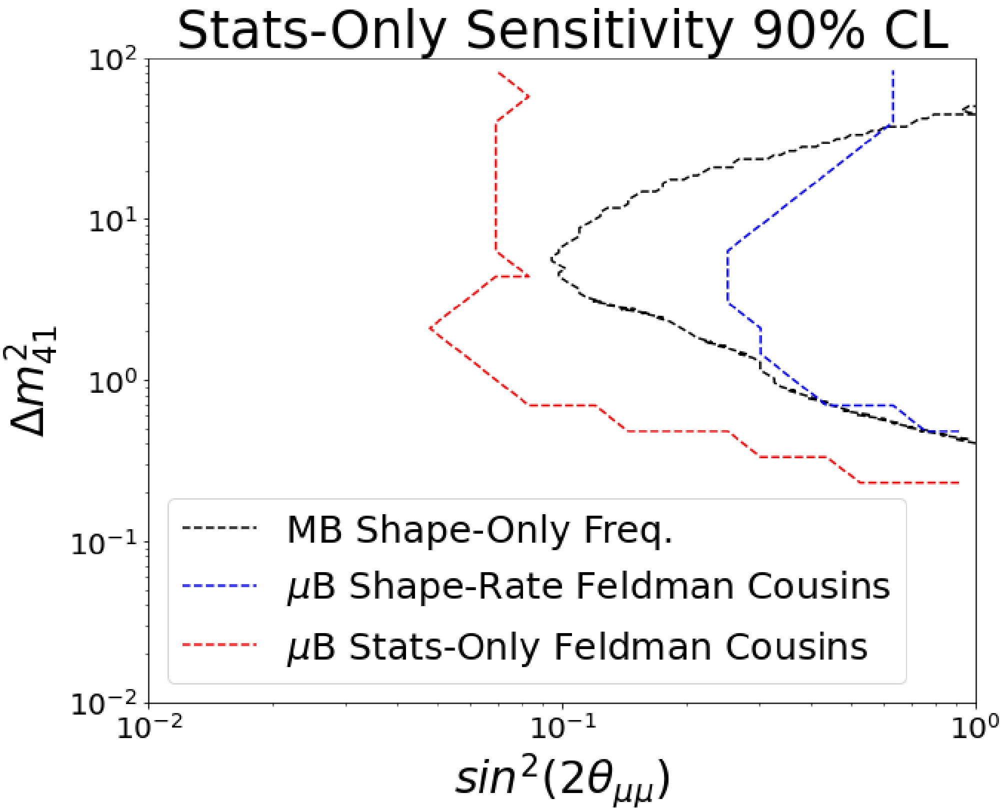


Figure 7.22: For the stats-only regime, MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. Both our normal Feldman-Cousins analysis, and MiniBooNE's ν_μ disappearance sensitivity are overlaid [15], though MiniBooNE's limit still includes systematic errors, and is shown only as a benchmark.

pearance effects if the observation is exactly the null model expectation.

7.3.4 Scaling Test

The analysis can also be studied by examining its performance in a regime where MicroBooNE has far more data that we actually anticipate. In this validation study, we scale the null model's expectation (from which all other spectra are derived) by a factor of roughly 40. This puts the statistical power of MicroBooNE level with MiniBooNE. This has the effect of reducing the statistical uncertainty associated with the observation which comes about from our combined Neyman-Pearson χ^2 term. Combined with the study in section 7.3.3, it also allows us to state whether the analysis is more statistically limited or systematically limited.

Following through the same analysis in this scaled regime, our fractional covariance matrix is the same as underlies the scaled version in figure 7.7. We use this and our scaled expectations to throw new pseudo-experiments and generate a grid of R_C values for the scaled experiment. These values, and their fractional relation to the threshold of 4.61 from a 2 degree-of-freedom assumption are shown in figure 7.23.

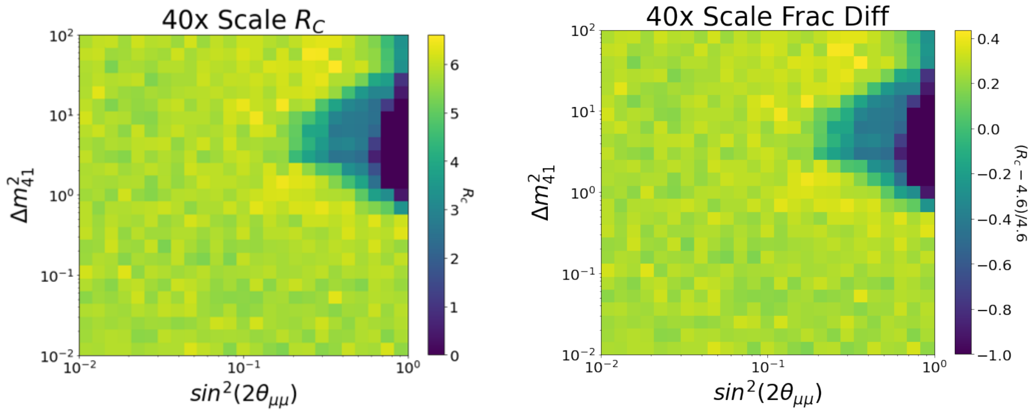


Figure 7.23: For the 40x scaled regime, the critical R values for 90% CL across every point in our parameter space. The left plot displays the raw R_C value, while the right shows the fractional difference in R_C from a 2-degree-of-freedom scenario.

Here the region of $R_C = 0$ again grows, however the growth isn't as strong as the stats-only case, indicating that reducing our statistical error isn't as significant as eliminating our systematic uncertainty. We calculate new test R values and compare them to these R_C values to determine a sensitivity derived from the scaled data regime. The resulting sensitivity is shown in figure 7.24.

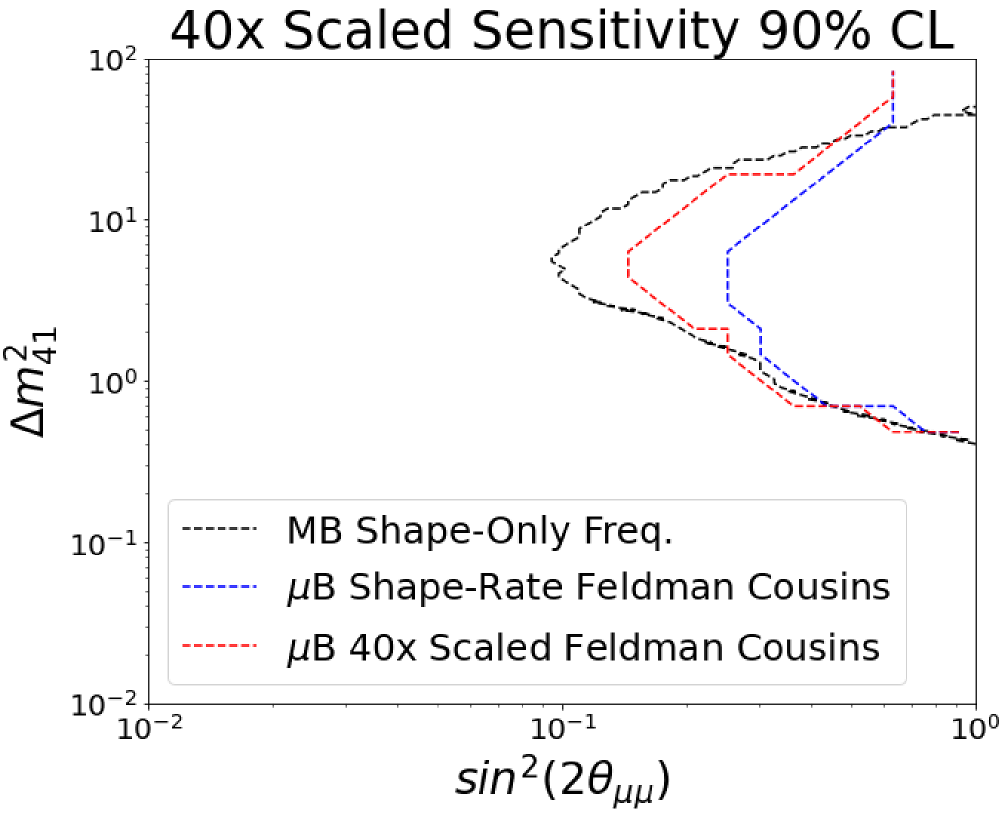


Figure 7.24: For the 40x scaled regime, MicroBooNE's 90% sensitivity contour for our parameter-space scan using the Feldman Cousins method of determining R_C. Both our standard Feldman Cousins sensitivity and MiniBooNE's ν_μ disappearance sensitivity is overlaid [15].

Now, by comparing figure 7.22 and 7.24 we can more definitively claim that the ν_μ disappearance analysis presented here is more systematics limited. The analysis grows more sensitive to lesser disappearance effects when the systematics are removed, compared to when the statistical uncertainty is reduced.

7.3.5 Fake Dataset Tests

We also perform the ν_μ disappearance analysis on two fake datasets generated by the MicroBooNE collaboration, referred to as fake dataset 1 and fake dataset 2. These datasets were explicitly generated with specific signals to test analyses.

Fake dataset 1 was generated with a low energy ν_e signal at 3.5× that of an unfolded MiniBooNE signal. However, the $1\mu 1p$ selection used for this analysis will remain mostly indifferent to more ν_e events. The set also removed MicroBooNE's GENIE central value tune event weights, which decreases the $1\mu 1p$ selection spectrum as the tune is removed.

Figure 7.25 shows both the fake dataset's spectrum and best matching disappearance spectrum (left) and the resulting 3+1 ν_μ disappearance model allowed regions (right) if we run the analysis on the fake dataset. This region is determined by checking where the calculated test R values for the fake dataset are less than the R_C values at each grid point as determined by the Feldman Cousins method described by figure 7.14.

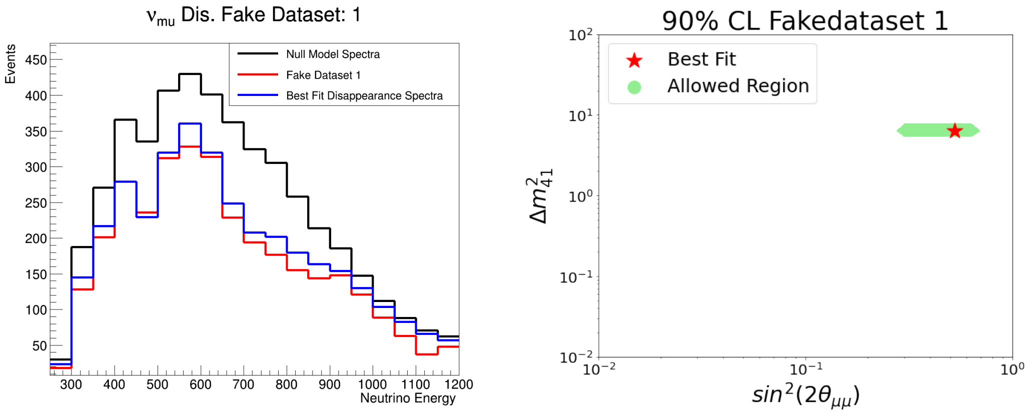


Figure 7.25: Fake dataset 1's spectrum (left) and the allowed regions (right) when run through the ν_μ disappearance search.

This results in a fairly small allowed region, with middling values for both $\sin^2 2\theta$ term and Δm^2 relative to our search phase space.

Fake dataset 2 was generated with increased contributions from quasi-elastic and meson-exchange-current interaction, both of which contribute significantly to the ν_μ $1\mu 1p$ selection. As such, we both expect and observe an increase in our selected spectrum. Figure 7.26 shows both the fake dataset's spectrum and best matching disappearance spectrum (left) and the allowed regions (right) if we run the analysis on the fake dataset.

Here we see the selected event spectrum is indeed higher than the null model, and the allowed region phase space is vast, covering many of our grid points. This is unsurprising as the ν_μ disappearance model is not capable of increasing the spec-

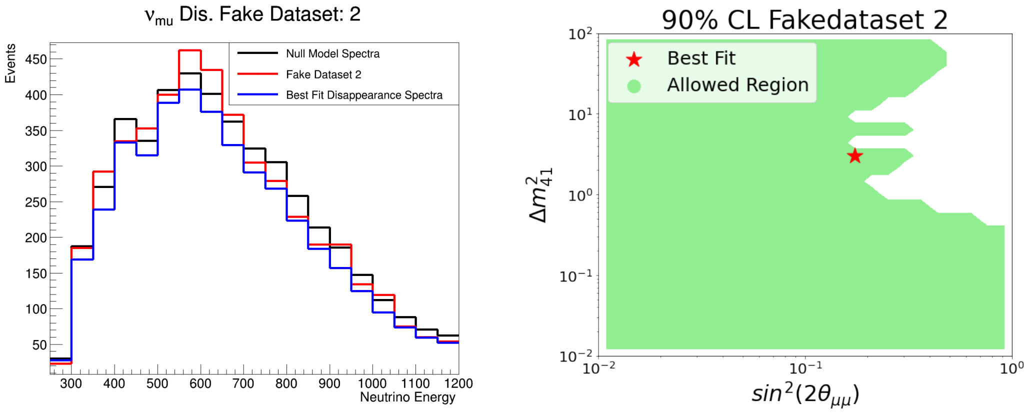


Figure 7.26: Fake dataset 2's spectrum (left) and the allowed regions (right) when run through the ν_μ disappearance search.

trum, as is easily seen in equation 7.5. Therefore the disappearance model cannot be tweaked to explain the increase in the selected spectrum in various bins.

7.4 3+1 Sterile Neutrino MicroBooNE Data Analysis

Now we are ready to perform our 3+1 muon neutrino disappearance search using the first three years of MicroBooNE data. This corresponds to 6.67×10^{20} protons on target (POT) worth of beam data. Recall the data selection is plotted against the expectation in figure 6.7.

First, the test statistic, R, is calculated by inputting this data as our observation. This is different from our sensitivity R values where we assume the observation is exactly the null expectation. The grid of R_{data} across our parameter space is shown in figure 7.27, on the left the values are shown with a linear z-axis scale, on the right, a log-scaled z-axis is used to shower finer features.

Here we see the best fitting model to our observation, and minimum R value, is at model parameters $sin^2(\theta_{\mu\mu}) = 0.12$ and $\Delta m^2_{41} = 3.02$ eV^2. However, we do not attribute much strength to this minimum as it falls squarely in the degenerate region of our search plot. More specifically, the minima falls within the bulk of the plot where the calculated R values are low, indicating that there is not much

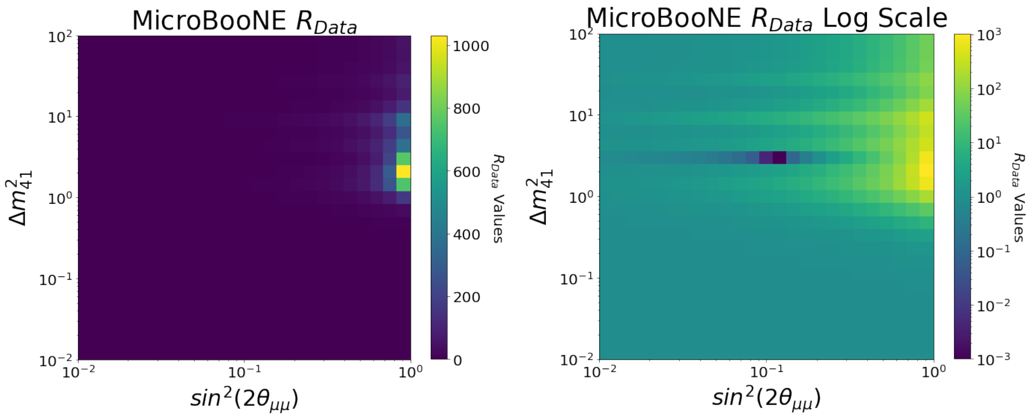


Figure 7.27: The R_{Data} values across every point in our parameter space where we use our 6.67×10^{20} POT worth of data as our observation. The left plot displays the raw R value, while the right shows the same values, with values below 0.1 set to 0.1 and a log scale applied for better visualization.

Table 7.3: The breakdown of components of our test statistic R for the best fit point and the null oscillation model.

	Null Oscillation Model	Best Fit Model	Difference
$ln(\|2\pi M\|)$	153.58	151.59	1.99
χ^2	21.10	22.22	-1.12
Total	174.68	173.81	0.87

discriminating power between the observation and the local expectation, compared to the observation and the best fitting expectation.

In figure 7.28 the best fit expectation (red), data (black) and null oscillation expectation (blue) are plotted together. Our expectation's uncertainties are plotted around the null oscillation in blue hashes. Plotted against one another we can see the best fit expectation is not far from the null model, particularly when considering our analysis' uncertainty bands.

In table 7.3 the different components of our test statistic R, the negative log likelihood ratio, are broken down. Comparing the values, the null model has a slightly better χ^2 value, 1.12 lower than the best fit, but the term containing the determinant of the covariance matrix is 1.99 lower for the best fit, allowing the best fit grid point to slightly beat out the null. However, with a net difference of just 0.87, the two points are quite comparable when considering the scale of the larger R values shown in the figure 7.27.

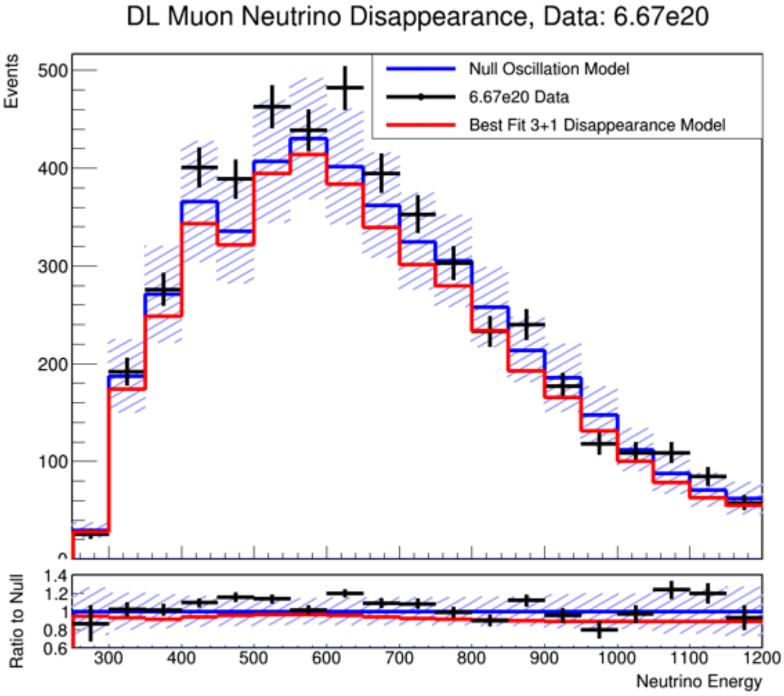


Figure 7.28: The selected neutrino interaction spectrum as a function of energy. In blue the null model expectation, with the uncertainty band surrounding it, in black our observation, the data, and in red the best fitting 3+1 sterile neutrino model.

Lastly we can examine the allowed and excluded regions of parameter space as determined by MicroBooNE's data. We compare the R_{Data} values to the R_C values as when we calculated our sensitivity. Now where R_{Data} is greater than R_C we can exclude at 90% confidence, as R_C was calculated via the 90% mark. The result of this test is shown in figure 7.29. The MicroBooNE allowed region is in green and the excluded region the white space in the upper right of the plot. Overlaid is the MiniBooNE experiment's own excluded region [15], as well as the MicroBooNE sensitivity. We see that generally, MiniBooNE's exclusion region is wider than ours. This is due to a mixture of our analysis being statistically, and systematically limited as demonstrated by figures 7.24 and 7.22. A global best fit for 3+1 sterile-related muon neutrino disappearance is performed in [43], and finds a best fit model of $sin^2(\theta_{\mu\mu}) = 0.0716$ and $\Delta m^2_{41} = 1.32$ eV^2. This global best fit remains in the MicroBooNE allowed region.

Of note is the fact that MicroBooNE's own exclusion region extends beyond our expected sensitivity. This is unsurprising because the sensitivity is estimated by treating the observation as the null expectation, while our actual observation is the data, which is generally in excess of the null expectation. As the only way for our disappearance model to adjust the expectation is by decreasing it, an excess observation places greater distance, and thus more discriminating power between the observed data, and our suite of test model expectations. This allows us to exclude more parameter space than anticipated.

We also see that our analysis is able to exclude phase space in the upper right region of our search that MiniBooNE is unable to, and share a slight border with them on the lower edge of our Δm^2_{41} values searched. In the upper right, at high values of Δm^2_{41}, we expect that MicroBooNE benefits from narrower energy bin widths, which allow more shape information to be observed at the higher frequency oscillations that high values of Δm^2_{41} bring. Our analysis, using the calorimetry of MicroBooNE's LArTPC, features 50 MeV energy bins, whereas MiniBooNE's search generally uses 100 MeV bins. Regardless, the upper right region of phase space has also been ruled out by the CCFR (Chicago-Columbia-Fermilab-Rochester) experiment who also searched for muon neutrino oscillations in 1984 [80].

From this analysis, we conclude that muon neutrino disappearance as part of a 3+1 sterile neutrino model is generally not observed in MicroBooNE's data within the parameter space we search. We add our experimental power to previous experiments in excluding a region of model phase space. Both the null oscillation model as well as the global best fit point for 3+1 muon disappearance, as determined in [43], remain within our allowed region.

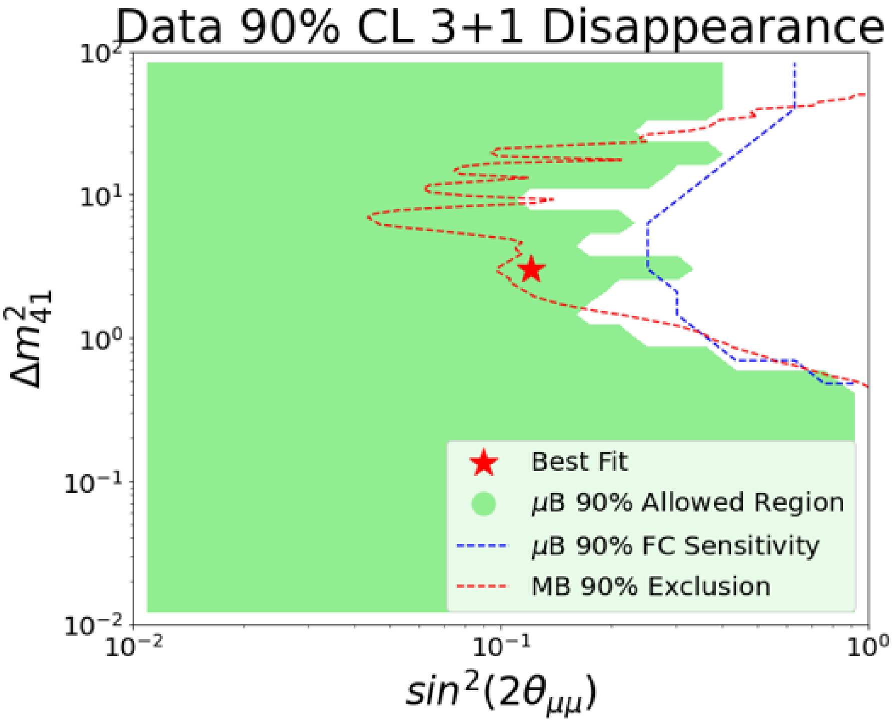


Figure 7.29: Using 6.67×10^{20} POT worth of MicroBooNE data we show the allowed regions of 3+1 model phase space in green, and the excluded region in white in the upper right of the plot, our Feldman Cousins sensitivity is overlaid in blue. MiniBooNE's excluded curve is also overlaid in red [15].

Chapter 8

Next Generation Cosmic Ray Tagging with sMask-RCNN

The material presented in this section is the same work as posted to arXiv [81], which was written by the author of this thesis. This means the document attempts to stand on its own, and some of the explanations present in other parts of this thesis are repeated or simplified. The complete author list is omitted, but is available on arXiv. This document has also been submitted to JINST.

8.1 Abstract

In this article, we describe a modified implementation of Mask Region-based Convolutional Neural Networks (Mask-RCNN) for cosmic ray muon clustering in a liquid argon TPC and applied to MicroBooNE neutrino data. Our implementation of this network, called sMask-RCNN, uses sparse submanifold convolutions to increase processing speed on sparse datasets, and is compared to the original dense version in several metrics. The networks are trained to use wire readout images from the MicroBooNE liquid argon time projection chamber as input and produce individually labeled particle interactions within the image. These outputs are identified as either cosmic ray muon or electron neutrino interactions. We find that sMask-RCNN has an average pixel clustering efficiency of 85.9% compared to the dense network's aver-

age pixel clustering efficiency of 89.1%. We demonstrate the ability of sMask-RCNN used in conjunction with MicroBooNE's state-of-the-art Wire-Cell cosmic tagger to veto events containing only cosmic ray muons. The addition of sMask-RCNN to the Wire-Cell cosmic tagger removes 70% of the remaining cosmic ray muon background events at the same electron neutrino event signal efficiency. This event veto can provide 99.7% rejection of cosmic ray-only background events while maintaining an electron neutrino event-level signal efficiency of 80.1%. In addition to cosmic ray muon identification, sMask-RCNN could be used to extract features and identify different particle interaction types in other 3D-tracking detectors.

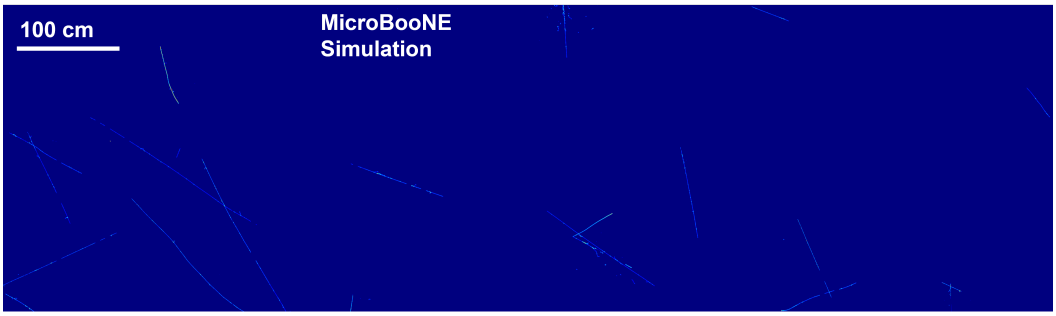
8.2 Introduction

The MicroBooNE [10] experiment uses a liquid argon time projection chamber (LArTPC) with an active volume of 85 tonnes to study neutrinos from the Fermilab Booster Neutrino Beamline, while also receiving neutrinos from the Neutrinos at the Main Injector (NuMI) beam. The MicroBooNE LArTPC is a near-surface detector that does not utilize any overhead shielding for cosmic background mitigation. This, combined with a long TPC readout time, described in section 8.3.1, results in a high ratio of cosmic ray muons to the number of neutrinos that interact within the detector. Techniques must be developed to reduce this cosmic ray muon background so that different neutrino interaction channels are measured with high purity.

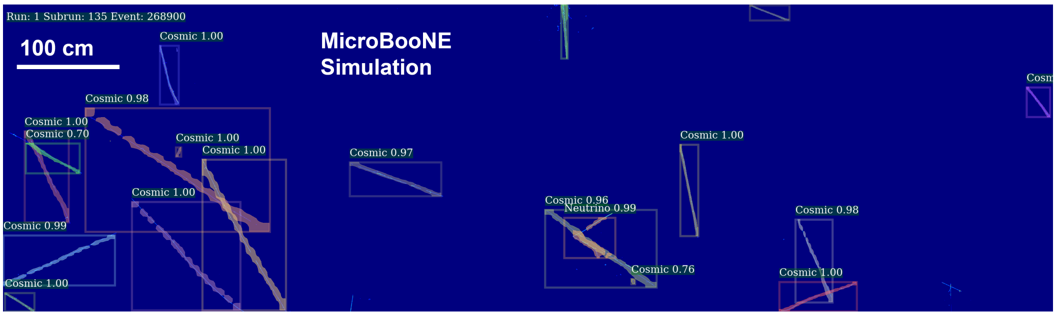
A cosmic ray interaction can be mistaken for a neutrino interaction regardless of whether there is a neutrino interaction in the readout window. We consider any cosmic ray muon depositing charge in the detector an 'interaction' regardless of whether it is captured in, decays in, or traverses the detector. Background events without a neutrino interaction present are called "cosmic-only" events. An example of a cosmic-only event is depicted in figure 8.1c. Due to the prevalence of cosmic rays at this near-surface location, any reconstruction tools designed to tag cosmic ray muons should be deployed early in the reconstruction chain to filter such non-neutrino events. This means that our solution to identifying and removing cosmic ray backgrounds must be deployable across the entire MicroBooNE dataset.

In this article, we present an approach to cosmic ray muon tagging using machine learning. Within the MicroBooNE experiment, machine learning techniques have been applied in other areas, such as particle identification, and pixel identification [82, 52, 53, 13]. In this article, we make use of a neural network called Mask-RCNN or "Mask Region-based Convolutional Neural Network" [83] to locate, identify, and cluster 2D interactions corresponding to the projections of the LArTPC wire planes. The design of Mask-RCNN is described in section 8.3.3.

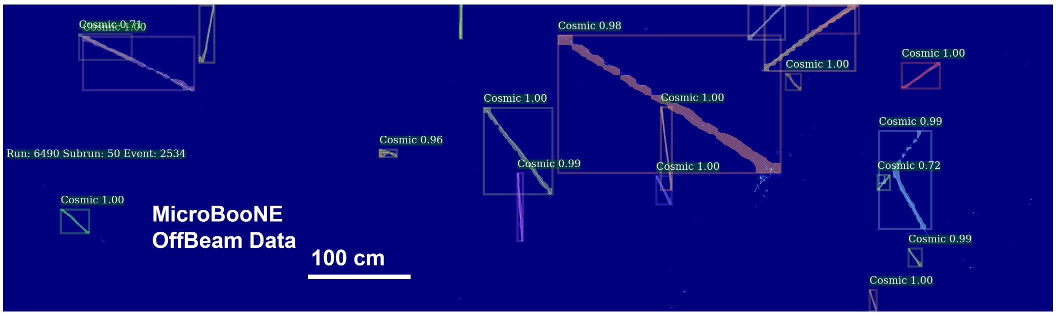
Machine learning algorithms are typically deployed on graphics processing units (GPUs) because their ability to parallelize computations pairs well with the matrix multiplications that are abundant in machine learning code. However, in order to deploy on the full MicroBooNE dataset, our tools need to run on central processing units (CPUs) because the MicroBooNE production chain has access to large amounts of CPUs but not integrated GPUs. While running on GPUs would speed up run time, integrating GPUs would require additional personnel and financial investment that is not presently feasible, therefore operations that use CPUs are required. To solve this problem, we extend Mask-RCNN to use sparse submanifold convolutions [84] which allow for much faster CPU running on sparse datasets by avoiding multiplication when one term is zero. We call this modified version of the network sparse Mask-RCNN or "sMask-RCNN". Section 8.4.2 contains a brief description of submanifold convolutions, and sMask-RCNN's modified state is described in 8.4.3. Several visual examples of sMask-RCNN's performance are provided in the form of event images in figure 8.1. For details on how the event images underlying the sMask-RCNN labels are made, see section 8.4.1. Code detailing our implementation of sMask-RCNN has been made available at: `https://github.com/NuTufts/Detectron.pytorch/tree/larcv1_mcc9`.



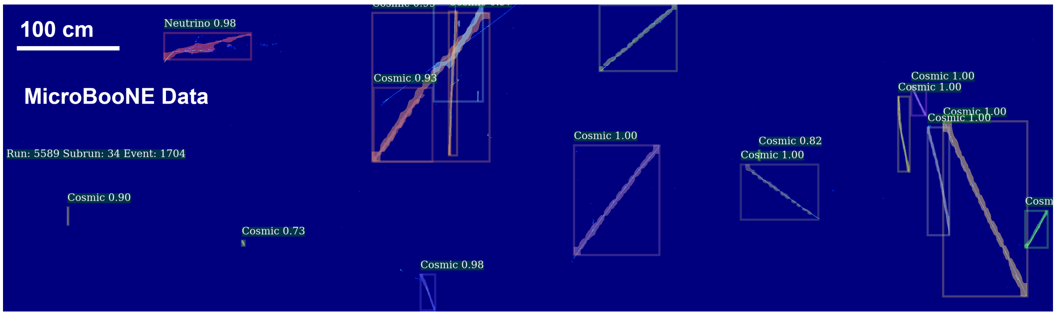
(a) An example of an input image as given to sMask-RCNN to process



(b) A simulated neutrino interaction overlaid on cosmic ray muons from data, labeled by sMask-RCNN



(c) Cosmic-only data event, labeled by sMask-RCNN



(d) Data event containing a neutrino interaction, labeled by sMask-RCNN

Figure 8.1: Several example event images. The vertical and horizontal scales are the same for all images. Each column of pixels along the x-axis refers to a specific wire readout, and each row along the y-axis refers to a different bin of signal readout time. This is described in greater detail in section 8.4.1. (a) is an example of an input image given to sMask-RCNN to process, whereas (b) shows the network's subsequent labeling of the same image. (c) shows a cosmic-only data event. (d) shows a data event containing a neutrino interaction that sMask-RCNN correctly identifies with some confidence score, and clusters.

8.3 Background

8.3.1 The MicroBooNE LArTPC

The LArTPC technology is designed to provide precision calorimetry and particle tracking while remaining scalable to larger sizes for future experiments. In MicroBooNE's LArTPC, a large volume of liquid argon is bounded on six sides within the time-projection chamber (TPC). On one side, the cathode, is a metal plate held at a negative potential of -70 kV. On the other side of the argon, held near ground, is the anode: a collection of three wire planes at progressively higher potentials. Figure 8.2 shows a diagram of the LArTPC principle.

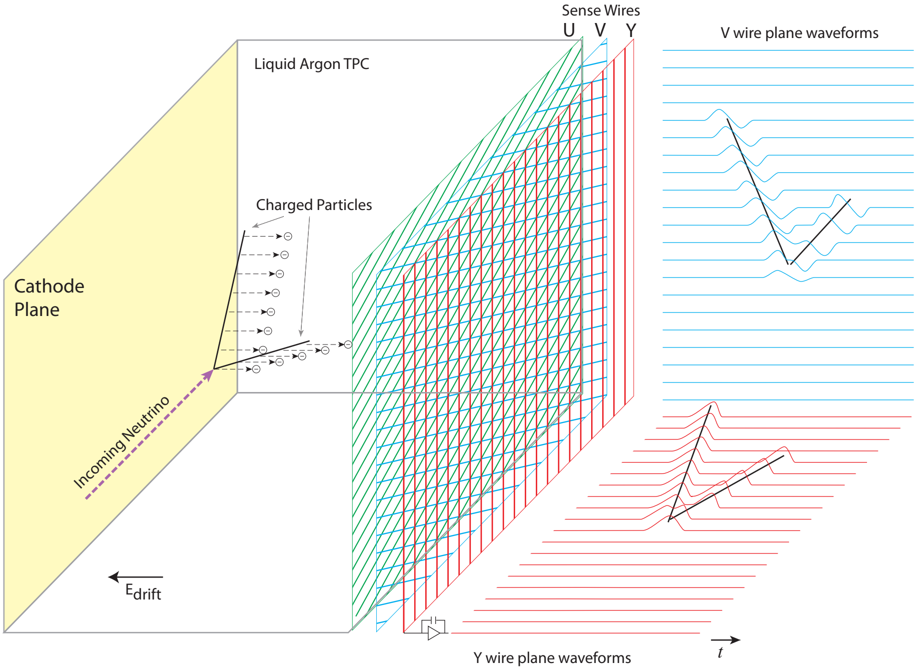


Figure 8.2: A diagram of the LArTPC principle. The signal formation for the second induction plane (V plane) and the collection plane (Y plane) are shown [10].

Each wire plane consists of a series of parallel wires spaced every 0.3 cm. The wires in each plane are oriented at an angle of 60° with respect to the wires in the other two planes. When a charged particle passes through the detector, it creates ionization electrons which drift toward the anode wire planes due to the nearly uniform electric field between the cathode and anode. As the electrons pass by the first two wire planes they induce bipolar pulses on the wires before finally arriving at the third and final wire plane. Here they are collected and create a unipolar pulse. Thus this final wire plane is called the ”collection plane”. The bipolar and unipolar pulses read out from the wires undergo noise filtering and 2D deconvolution described in [85, 86]. This processing removes much of the noise from the wires and transforms the bipolar pulses into unipolar pulses. These wire signals are used to create the input images used by the neural network. The process for creating the input images is described in section 8.4.1.

8.3.2 Existing cosmic identification tools

Cosmic ray muon tagging and background removal have previously been performed in MicroBooNE using a variety of methods. One example uses deep learning with semantic segmentation to differentiate cosmic ray muon pixels from neutrino interaction pixels [87]. Mask-RCNN expands on semantic segmentation, further separating each instance of every individual interaction it finds using bounding boxes. This means that each cosmic muon interaction in the detector receives its own labeled and clustered output.

Cosmic ray muon tagging has also been performed with more traditional algorithmic approaches. In the MicroBooNE experiment, one such method is the PandoraCosmic algorithm [88, 89]. This algorithm clusters hits in 2D and then combines these clusters into 3D tracks. It flags a track as a cosmic ray muon if part of the track is placed outside the detector based on timing information or if the track trajectory begins and ends at a TPC boundary using information related to the track’s timing. It provides an exception for through-going trajectories that are parallel to the beam direction. This is designed to eliminate cosmic ray muons as they will appear as tracks originating from outside of the detector, and will be crossing perpendicular relative to the beam direction.

Another method used in MicroBooNE is the Wire-Cell (WC) cosmic tagger [90, 91] which is made up of several event-level requirements, combined with the WC charge-light (Q-L) matching algorithm [51]. This Q-L matching algorithm uses light information detected during the neutrino spill by 32 8-inch cryogenic photomultiplier tubes (PMTs) mounted behind the TPC wire planes. This light information is then spatially matched to charge deposited in the TPC, selecting TPC pulses created during the beam spill. Therefore, both the Q-L algorithm and the full WC cosmic tagger use additional information beyond the wire planes, which is what sMask-RCNN uses as input. In section 8.6.2 we show results achieved by a combination of sMask-RCNN and WC algorithms to produce a state-of-the-art cosmic ray tagger.

Cosmic ray tagging can also be achieved with hardware solutions. An example in MicroBooNE is the design and construction of the cosmic ray tagger system [39]. This system was introduced partway through MicroBooNE operation, and therefore is only available for part of the MicroBooNE data. It uses plastic scintillation modules to acquire the time and location for particles traversing the TPC. Reconstructed tracks can then be matched to this data and be flagged as pertaining to cosmic ray muons rather than neutrino interactions. Additionally, the cosmic ray tagger system can be used in tandem with software solutions to improve performance.

8.3.3 Object detection and Mask-RCNN

The original Mask-RCNN network is designed to perform three common tasks in the field of computer vision: object detection, classification, and semantic segmentation. In the field of computer vision, classification is a task commonly performed to label an image as one of some predefined list of classes, for instance an image might be labeled a cat or a dog. Semantic segmentation refers to a labeling performed at pixel level, for example in an image with a cat and a dog, the pixels making up the dog are labeled ‘dog’ pixels and those making up the cat are labeled ‘cat’ pixels, while the remaining pixels are given a background label. The network is trained to receive some input image, place bounding boxes around objects of interest, classify these objects within some set of user defined classes, then within each box label each pixel as part of the object or not.

Structurally, Mask-RCNN is comprised of four subnetworks. First is a residual network (ResNet) [92]. This network runs on the input image and creates a feature map for the image. This feature map is then fed into a region proposal network (RPN) [93], which then produces a series of bounding boxes around regions of interest (RoIs) within the image. The bounding boxes are described by a 2D coordinate, a height and a width, and are designed to produce the smallest rectangular box containing the object. The RoIs are aligned in the feature map space via the RoIAlign algorithm, then combined with appropriate features, scaled to a fixed size and fed into the two final subnetworks: a classifier, and a fully convolutional network (FCN) we refer to as the ”maskifier”. The classifier takes each bounding box and its features and predicts which class of object it is with some confidence score. The maskifier produces a semantic segmentation mask of all the pixels within the box, determining which pixels correspond to the object and which are background. This semantic segmentation mask is synonymous with a cluster of pixels within the box, though the cluster need not be connected. Figure 8.3 shows a simplified view of the network architecture.

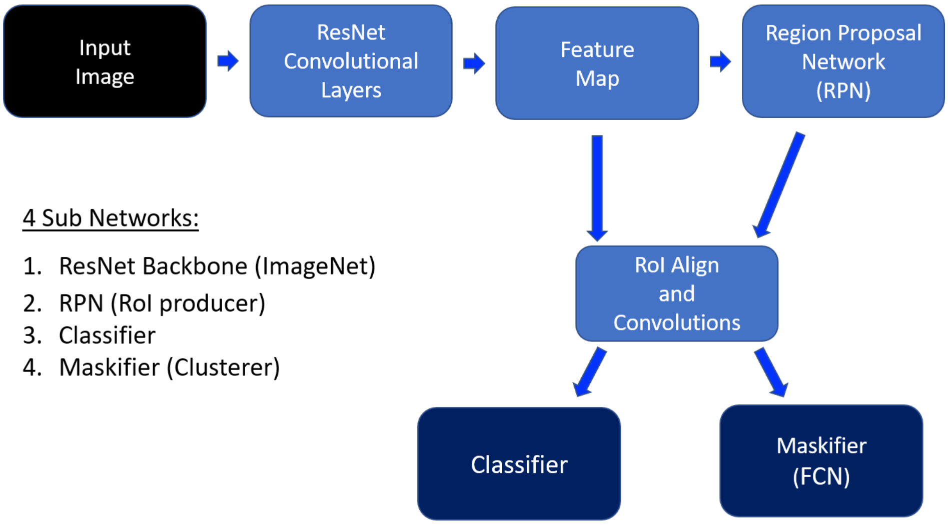


Figure 8.3: Network Architecture for Mask-RCNN in MicroBooNE.

8.4 Methods

8.4.1 Data preparation

To create images for analysis using MicroBooNE LArTPC data, we use the charge readout from the three wire planes at the anode. The neutrino beam window is 1.6 ms, but we record a buffer on either side providing a modified window of about 3 ms. The data taken during this 3 ms comprises an "event". Over the course of this recorded beam window these wires are sampled as a rate of 2 MHz. This equates to 6048 samples per wire.

We create the event images shown in this article by placing the wire number along the x-axis and the sampling time along the y-axis. However, we first downsample the number of time samples per wire by a factor of six, going from 6048 samples per wire to 1008. This downsampling is performed both to reduce image size and to make the drift distance per pixel of an ionization electron roughly 0.3 cm. This distance matches the wire separation within a given wire plane. The value stored in each pixel within an image then is proportional to the charge deposited in the detector. We will refer to this as the "pixel intensity". We then apply a threshold the image by setting any pixel with pixel intensity below 10 (arbitrary units) equal to zero to further reduce noise. In comparison, the pixel intensity distribution from minimum ionizing particles peaks at ~ 40 in these arbitrary units.

While there is one image made for each of the three LArTPC wire planes, for this study we only use the collection plane. We choose the collection plane over the other wire planes because the collection plane does not require signal processing to turn bipolar pulses into unipolar pulses and therefore the signal is cleaner. The event images then have a dimensionality of the number of wires on the collection plane times the number of samplings per wire, or 3456×1008.

Lastly, in the MicroBooNE LArTPC a portion of the wires are unresponsive [40]. In the collection plane this happens for about 10% of the wires. This creates an artifact in the images by creating vertical lines of unresponsive pixels. In some cases large groups of adjacent wires are unresponsive, leading to regions of unresponsive pixels.

8.4.2 Sparse submanifold convolutions

In the event images created from the LArTPC collection plane, about 0.7% of the pixels are nonzero, making the data ”sparse”. This is because the pixel value comes from the ionized electrons drifting away from the charged particles moving in the detector. Most of the time, the wires are reading out low-level noise that is below the threshold. The resultant low pixel occupancy means that when we apply Mask-RCNN to the event images, there are many computations that involve multiplications by zero. These trivial calculations waste computing resources, particularly if performed in sequence via a CPU and not in parallel via a GPU.

A normal convolution in a neural network takes a convolutional kernel or filter and moves it across the image, multiplying at each location to acquire the convolved value. In comparison, a sparse submanifold convolution only multiplies the kernel against positions centered on nonzero pixels, avoiding computations on zeroed regions of the input. Notably, a submanifold convolution is not mathematically equivalent to a regular convolution. In normal convolutions, kernels centered on zeros in the input image can output a nonzero convolved value if the edge of the filter captures some nonzero input. An example of this is shown in figure 8.4. This blurs the features coming out of a convolution, so that convolutions which are performed one after another in a deep neural network, such as ResNet, spread information outward.

8.4.3 Sparse Mask-RCNN

We utilize Mask-RCNN to locate, identify, and cluster interactions within the 2D event images described in section 8.4.1. The network places bounding boxes around, classifies, and then clusters pixels corresponding to deposited charge for each interaction it finds within the image. We define an interaction as all of the charge

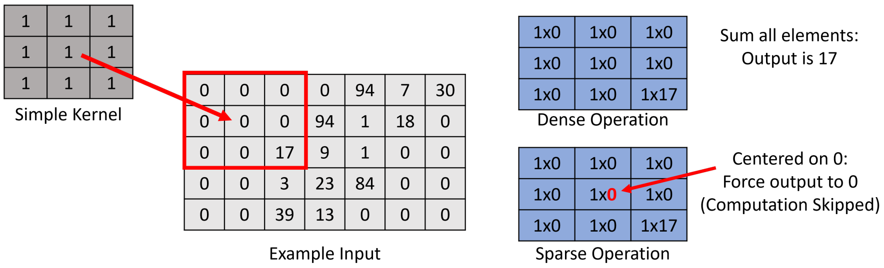


Figure 8.4: An example of a convolution operation that depends on use of submanifold or normal convolutions. The normal convolution multiples the kernel against the image at the given position, and outputs a convolved value of 17. Meanwhile, the submanifold convolution does not get computed when centered on a zero. The submanifold convolutions used in sMask-RCNN have a kernel size of 3×3 (same as shown in the figure), with a stride of 1.

deposited in an image coming from the same ancestor particle. For example, if an electron neutrino interacts with the argon in the LArTPC, and yields a proton and an electron, all of the charge deposited from the proton and electron are combined into one ancestor ”electron neutrino” interaction and should be clustered together. A more detailed description of network training is described in section 8.4.4.

To speed up the network when deployed on CPUs, we swap the ResNet convolutions with sparse submanifold convolutions while maintaining the original network structure. Additional work could be done to make the later subnetworks use sparse convolutions but examinations of the compute times for the individual parts of sMask-RCNN made this unnecessary. For clarity, we will refer to Mask-RCNN without submanifold convolutions as dense Mask-RCNN.

The change to sparse convolutions yields a significant gain in terms of network speed for inference when running on CPUs. The timing information for running the dense and sparse configurations of the network is shown in table 8.1. These timings were performed on an Intel(R) Core(TM) i9-9820X CPU @ 3.30 GHz and measure wall time. This allows the network to be inserted into production code on CPU farms such as FermiGrid [94] and to scale up how quickly the network is run over large data samples. We note that the Intel CPU we tested on is superior to what is generally available on CPU farms. Brief testing done on available CPU farms

Table 8.1: The average inference runtimes per 3456 × 1008 pixel image on a CPU. The first row is the runtime for just the ResNet portion of Mask-RCNN on the images. The second row is the time to run the entire network on the images. In the case of sparse ResNet, the time spent making the input image into a sparse tensor and the output features into a dense tensor is included in the sparse ResNet module time.

	Dense ResNet	Sparse ResNet
ResNet Runtime	3.172 s	0.1758 s
Full Detection Runtime	8.438 s	5.79 s

shows the ResNet runtime difference is exacerbated. This means the difference between sparse and dense implementations is even greater when older CPUs are used. This further prioritizes shortening the ResNet runtime, as when we deploy on CPU farms, we will use a variety of CPUs with lesser performance than an Intel(R) Core(TM) i9-9820X CPU @ 3.30 GHz. Further, an added benefit of using CPUs is that this technique is scalable to future experiments and studies where more data may be analyzed, and the computing resources cannot scale to a reasonable number of GPUs.

The implementation to sparse ResNet also introduces improvements to training when it comes to memory. For the dense version of the network, due to memory constraints, we could only train the network on 832 × 512 crops of the event image. We use the word crop to refer to a random cutout of the original image of this new 832 × 512 size, where the crop must contain a portion of a simulated interaction. However, sMask-RCNN is trained on the full 3456 × 1008 event images because the memory required on the GPU to store a full image for the network is reduced by roughly a factor of the image's occupancy, as all zero pixels are no longer operated on. Regardless of training size, both the dense and sparse forms of Mask-RCNN could be deployed on the full 3456 × 1008 event images, as less memory is used if not actively training the network.

All of the event images shown in this article use the sparse implementation of Mask-RCNN. We examine the difference in performance between the dense and sparse networks in section 8.5.

8.4.4 Network training

When training, the entire network performs a forward pass on an image, and then the backward pass updates the weights of all the subnetworks based on a combined loss function built from the outputs of the maskifier, classifier, and RPN, as described in the original Mask-RCNN article [83]. Rather than train the ResNet from randomized initial weights, we use weights pretrained on the ImageNet dataset [95], which is a publicly available labeled dataset of images of animals and everyday objects. The ImageNet dataset is commonly used in the field of computer vision. We briefly started training ResNet from scratch, with randomly initialized weights, because the ImageNet pretraining is designed to identify animals and everyday objects, but despite this, we found that using the pretrained ResNet gave more useful features for the other components of Mask-RCNN to utilize in this particle physics analysis.

To train both the dense and sparse forms of Mask-RCNN, we use a sample of simulated electron neutrino events featuring simulated cosmic background. This means that every full 3456×1008 image contains a single electron neutrino interaction among many cosmic ray muons. While the dense network was trained on crops containing at least part of a neutrino interaction or cosmic ray muon, they did not always have an example of both within the same crop.

The interactions present in the training data are broken up into six different interaction classes, detailed in table 8.2. Dense Mask-RCNN was trained on each of these interactions, but as we determined our goal was primarily cosmic ray muon tagging, we only trained the sparse implementation on cosmic ray muons and electron neutrino interactions. Simulated interactions where the simulated ancestor particle was one of the four other classes were still present in the data but the network was told to ignore them. The training data uses CORSIKA [59] for cosmic ray muon simulation. For electron neutrino interactions the GENIE neutrino interaction simulator [57] is used. In both instances, GEANT4 [58] is used to model the detector response.

We also note that the version of simulation used to produce the training data for the sparse network is slightly updated as a newer version of simulation became available. Both the dense and sparse networks have their performance evaluated on the same set of 9400 events from the newer version of simulation, meaning there is a slight discrepancy between the training and testing data for the dense network. However, we expect this change to have minimal effect on the network performance.

Table 8.2: The different class types and number of occurrences in the training sets for the dense and sparse versions of Mask-RCNN. Note that the sparse network only trained on cosmic ray muon and electron neutrino interactions.

	Dense Training		Sparse Training	
Interaction Class	Counts	Percentage	Counts	Percentage
Cosmic Ray Muon	2708730	92.99	786050	95.24
Electron Neutrino	97034	3.33	39296	4.76
Neutron	26072	0.90	-	-
Proton	5738	0.19	-	-
Electron	155	0.005	-	-
Other	75026	2.58	-	-

As is common practice in machine learning, we split the data into two orthogonal subsets: a training set with 80% of the events and a validation set with 20%. When training the network, we use events in the training set, and whenever we wish to measure the performance of the network, such as calculating performance metrics shown in section 8.5, we use the validation set. This is a critical part of machine learning because it verifies that the network can generalize and perform its task on events outside of the training set.

The dense version of the network is trained on a sample of 230,000 crops for 1.75 epochs. In the context of machine learning, an epoch is one training pass through the data. While 1.75 is a low number of epochs, each crop features multiple interactions seen and masked by the network. Then the network is trained on a subset of 30,000 of these crops containing examples of high intersection-over-union (IOU) interactions. IOU between two interactions is defined as the number of pixels present in both interaction bounding boxes, divided by the total number of unique pixels present in either bounding box. These crops featured multiple interactions with overlapping bounding boxes. This fine tuned training is performed due to poor performance by the dense network on overlapping interactions. The training on this subset is performed for 8 epochs so the network can focus on learning these difficult events.

The sparse version of the network is trained on a sample of 40,000 full event images for three epochs. No fine tuning needed to be performed on highly overlapping interactions as the sparse network did not appear to suffer from the same issue as the dense network. It should be noted that while at first glance the training sample sizes of the dense and sparse networks differ by a significant factor, in actuality the dense crops were made at a factor of up to 10 crops per full image, and the training datasets are comparable in terms of interaction sample variance. Further, all performance evaluations for both networks are deployed on the same validation set of 9400 full sized images.

8.5 Comparing dense and sparse performance

In this section we define several metrics to test the performance of Mask-RCNN at identifying and clustering interactions within MicroBooNE event images. Then we compare the performance between the dense and sparse versions of the network. For this evaluation, we include interactions found by the network with a class score of 0.4 or higher. The class score is a score between 0 and 1.0 indicating how confident the network is that the class label is correct. This threshold is chosen to provide a balance between the purity and efficiency metrics defined and discussed below. Once this threshold is applied, all remaining predicted interactions are treated equally for the purpose of calculating metrics.

All analysis in this section uses the full scale 3456 × 1008 event images in the validation set of simulated data used to train the sparse network. This refers to the data described in the right two columns of table 8.2.

8.5.1 Efficiency and purity

For the purpose of this section, we will refer to interactions either as "true" or "predicted". A true interaction is one that exists in the simulation, whereas a predicted interaction is one that the network claimed it has found and labeled. We define two metrics to measure the pixel level efficiency and purity of the network's ability to find and cluster interactions. The efficiency is a measure of the percentage of pixels in a true interaction that are masked by the network's prediction. Purity is defined for each predicted interaction as the highest fraction of pixels belonging to the predicted interaction and a single true interaction. For example, if 30% of a prediction maps to true interaction A, and 50% maps to true interaction B then the purity is 50% for that predicted interaction. In both of these definitions, only pixels containing charge deposition above the pixel intensity threshold of 10 are considered, as we do not care about clustering empty pixels. Concretely, the efficiency E is defined as

$$E = \frac{\sum_{ij} T_{ij} \cdot W_{ij} \cdot M_{ij}}{\sum_{ij} T_{ij} \cdot W_{ij}}, \tag{8.1}$$

where T, W, and M are matrices representing the truth interaction, wire event image, and predicted interaction mask, respectively, with dimensions of the event image. Meanwhile, i and j are pixels indices. A visual representation of the efficiency calculation is depicted in figure 8.5. Similarly, the purity P is defined as

$$P = \frac{\sum_{ij} M_{ij} \cdot W_{ij} \cdot T_{ij}}{\sum_{ij} M_{ij} \cdot W_{ij}}. \tag{8.2}$$

The purity calculation is depicted in figure 8.6. For each of these equations, the values in T are 1 if the pixel belongs to the true interaction, and 0 otherwise, while the values in M are 1 if the pixel belongs to the predicted interaction mask, and 0 otherwise. Finally, the values in W are 1 if the pixel has any deposited charge, and 0 otherwise. After the element-wise multiplication of the matrices, the summations then run over the indices of the matrix. This corresponds to a counting of the pixels corresponding to the union of the given matrices.

A true interaction's efficiency is taken as the best value as calculated for all predicted interactions. A predicted interaction's purity is taken as the best value when calculated for all true interactions.

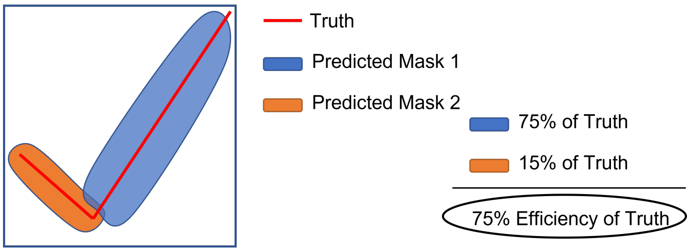


Figure 8.5: A visual representation of the definition of efficiency. Only nonzero pixels in the event image are counted.

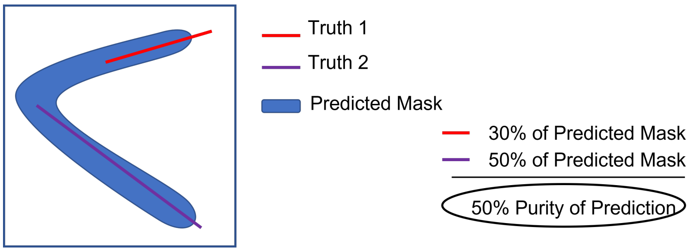
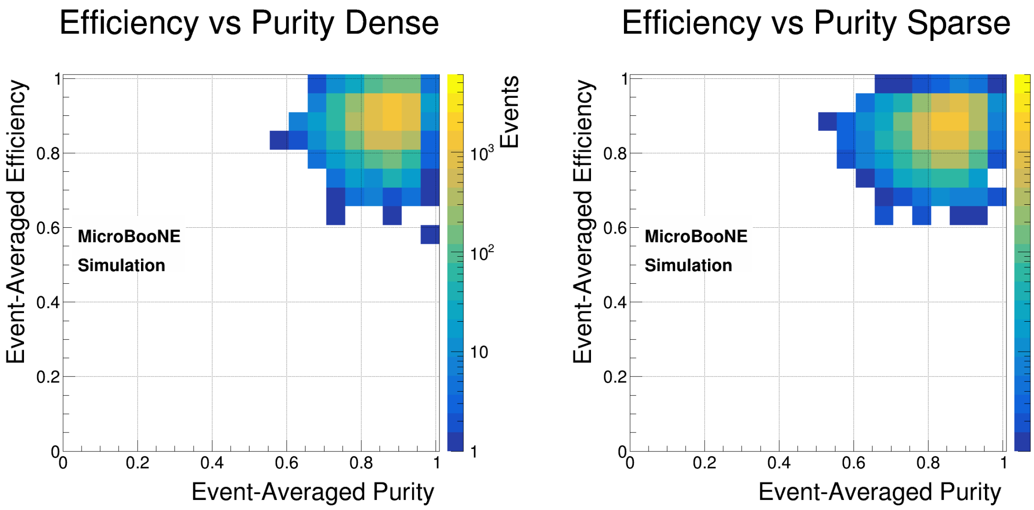


Figure 8.6: A visual representation of the definition of purity. Only nonzero pixels in the event image are counted.

These definitions mean that a given event image will have one efficiency measurement for each true interaction, and one purity measurement for each predicted interaction. While we are aware that object identification customarily uses panoptic quality [96] or intersection-over-union as evaluation metrics, we choose to use efficiency and purity in better keeping with particle physics analysis language.

For each event, we average the purities and efficiencies for the predicted and true interactions. These averages are of O(20) interactions, where there is a single neutrino interaction and many cosmic ray muon interactions. We reiterate that a cosmic ray muon 'interaction' is just any cosmic ray muon and potential daughter particles that deposit charge in the detector. These event-averaged values are plotted in the 2D histograms shown in figure 8.7.

Perfect efficiency and purity would yield values of 1.0 for each, so these histograms have targets in the upper right corners. We can see that the event-averaged



(a) Dense Mask-RCNN

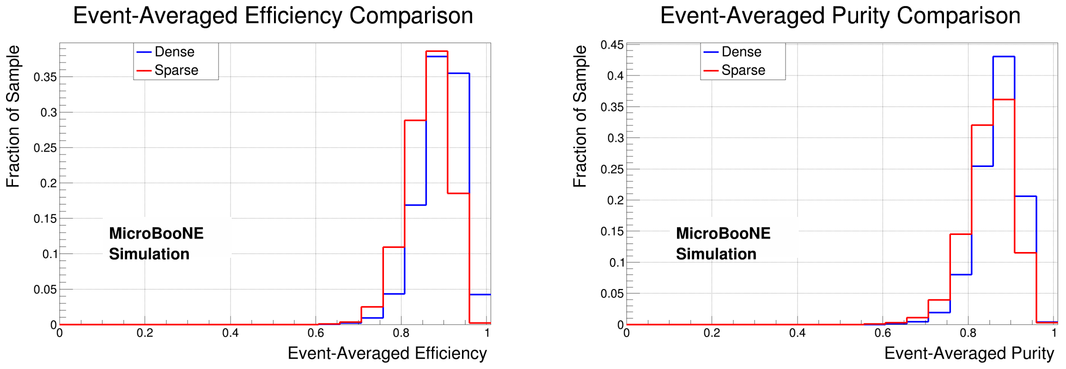
(b) Sparse Mask-RCNN

Figure 8.7: The event-averaged efficiencies and purities for the dense and sparse implementations of Mask-RCNN. The dense network has a mean event-averaged efficiency of 0.89 and a mean event-averaged purity of 0.87. For the sparse network these values are 0.86 and 0.85. Each of these evaluations use the same validation dataset.

purity drops from 87% for the dense to 85% for the sparse, while the event-averaged efficiency drops from 89% to 86%. The one-dimensional projections of figure 8.7 are shown in figure 8.8, where the event-averaged efficiency and purity distributions are compared between the dense and sparse networks.

It is also useful to examine the individual interaction efficiencies, rather than the event-averaged versions. The distributions for both the dense and sparse versions of Mask-RCNN are shown in figure 8.9. Here we can see the sparse (red) distribution is worse than the dense (blue) distribution. Notably the size of the peak at zero is the same for the two versions of the network. A true interaction will have zero efficiency if the network has no prediction that masks part of it. The fact that the two versions have the same sized peak at zero indicates that they each find the same number of interactions, but the dense masks are somewhat more complete.

The efficiency calculation is modified slightly by weighting the pixels within an interaction by their deposited charge. This version of the efficiency we term ”charge efficiency” and is shown in figure 8.10. Here we see a shift to the right for each distribution compared to their pixel-level efficiency in figure 8.9. This indicates



(a) Event-Averaged Efficiency

(b) Event-Averaged Purity

Figure 8.8: The one-dimensional distributions for the event-averaged efficiency (a) and purity (b) shown in figure 8.7. Each plot compares the dense and sparse network performances.

the network's preference for clustering higher value pixels corresponding to larger deposited charge, though this result may be due to the network being more likely to grab the center of tracks in our image, where the higher value pixels lie, compared to the halo of hits along a track's edge. While not surprising, this is a useful feature as the physics quantity we are dealing with is charge, not pixel count.

If we explore the interactions lying within the zero efficiency peaks in these plots then we find two common failure modes. The first is made up of interactions that lie completely or significantly in the unresponsive regions of the event image. Recall that roughly 10% of the MicroBooNE LArTPC wires are unresponsive, corresponding to vertical lines of unresponsive regions in the event images. True interactions within these regions are in the simulation, but have little in the way of signal in the event image for the network to detect. An example of a true interaction simulated in an unresponsive region of the image is shown in figure 8.11. It is unreasonable to expect the network to be able to label such interactions.

The second failure mode that contributes to the zero efficiency peak are true interactions that tend to be smaller in spatial extent compared to a typical simulated interaction, with less charge across fewer pixels in the event image. An example of this is shown in figure 8.12. These interactions are reasonable to expect the network to find as there is nothing to obscure the interaction. However, as they tend to be smaller, they are less likely to overlap with a neutrino interaction in the image or

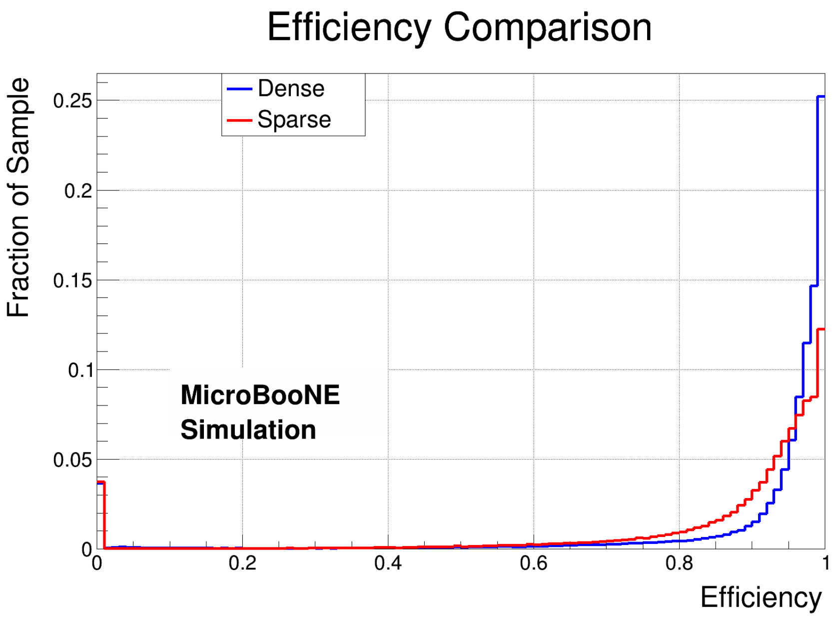


Figure 8.9: The interaction-level efficiency of the dense and sparse versions of Mask-RCNN as measured on the validation set.

confuse the network, and therefore are a less important part of the background.

8.5.2 Interaction coverage

Now that we have explored a pixel-wise efficiency, we next examine interaction coverage within a given event. We define a true interaction as being ”covered” if its pixel-level efficiency as defined in section 8.5.1 is greater than 80%. This means the network has to cluster the majority of the interaction, while still leaving some room for error. Figure 8.13 compares the fraction of true interactions that are covered in a given event for both the dense and sparse implementations of Mask-RCNN. The dense network has a slight edge over the sparse version, but both networks consistently cover the majority of true interactions within a given event.

It is also useful to examine the performance of Mask-RCNN as a function of the number of true interactions in an event. This investigates whether the performance of the network falls off for ‘busier’ events with additional particle interactions cluttering up the image. To examine this, we look at the number of covered true interactions as a function of the number of true interactions in the events. Figure 8.14 shows this measurement for both the dense and sparse networks.

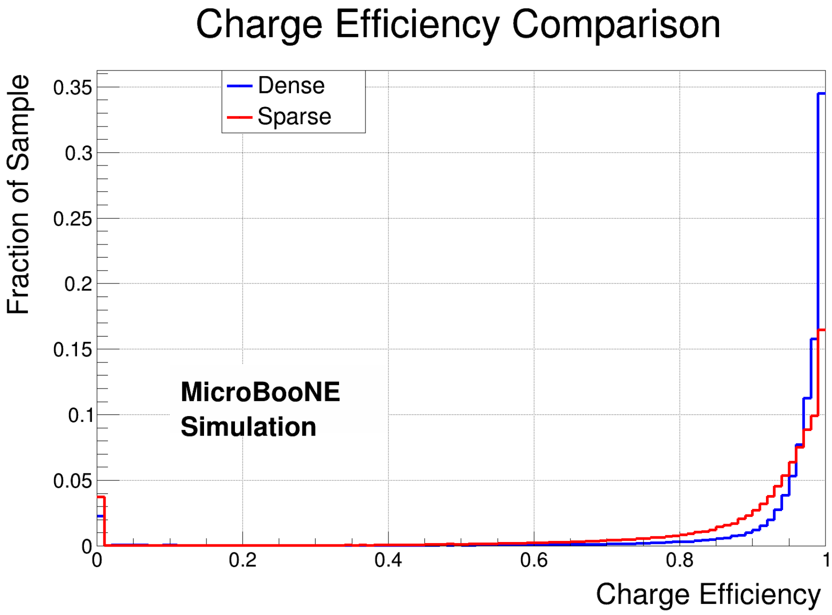


Figure 8.10: The interaction-level charge efficiency of the dense and sparse versions of Mask-RCNN as measured on the validation set.

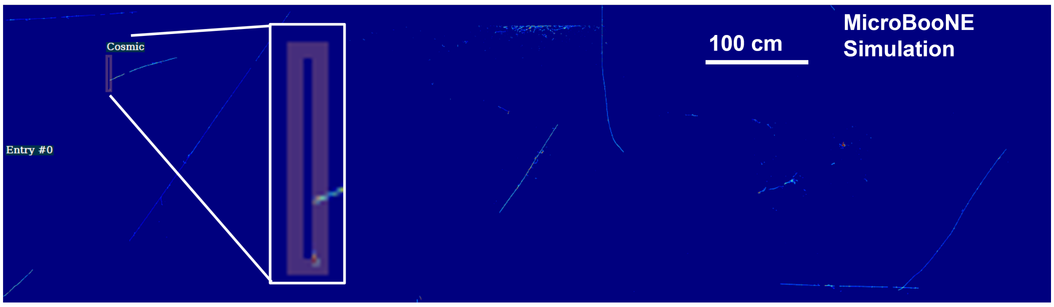


Figure 8.11: A zero-efficiency true interaction almost entirely in a region of unresponsive wires. The white box shows a zoom-in of the area of interest, and within it, the colored box should contain true neutrino interaction. However, because this interaction falls in an unresponsive region of the detector, no deposited charge is seen inside the colored box.

While the ideal network would cover all of the interactions, we see that both versions of the network produce a distribution slightly below the target line $y = x$. The fact that these distributions are linear demonstrates that the network performance does not diminish as the number of true interactions in a given event increases.

8.5.3 Network comparisons discussion

In comparing the dense and sparse versions across these various metrics, we find that the dense implementation performs better than the current version of sMask-

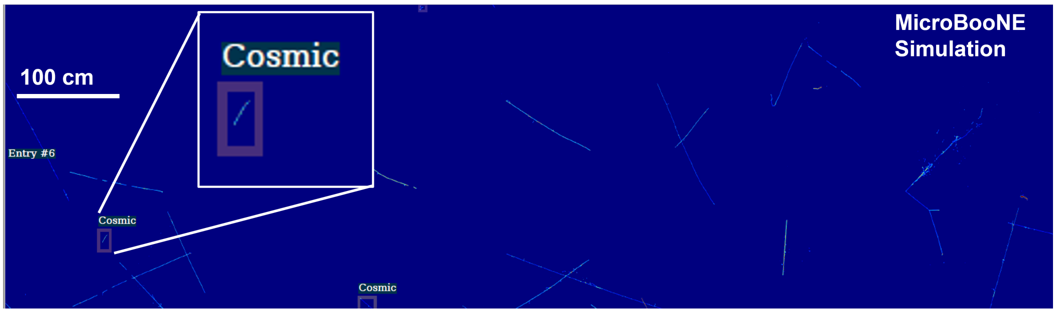


Figure 8.12: Another failure mode for the zero-efficiency peak. Here the cosmic interaction is relatively small in size compared to others in the event image. The colored box is the true interaction, and the white box shows a zoom-in of the area of interest.

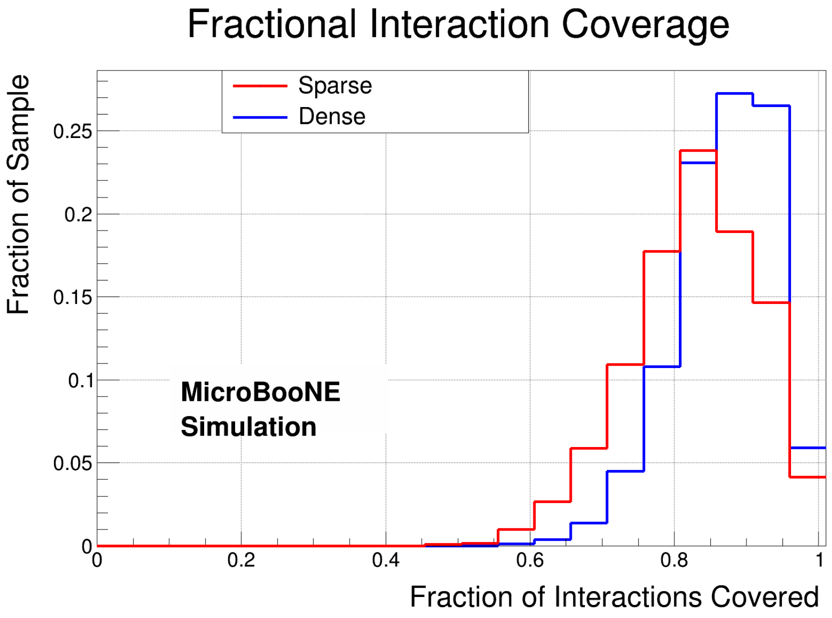
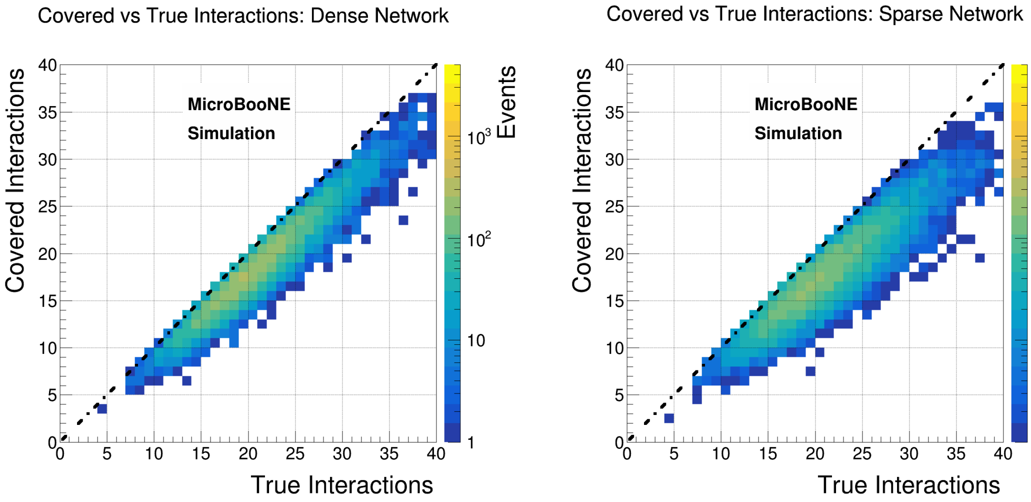


Figure 8.13: The fraction of true interactions in events that have greater than 80% efficiency as measured on the validation set.

RCNN. Viewing the two efficiencies of the networks, the peak at 1.0 for the dense Mask-RCNN is larger, and narrower than the peak in the sMask-RCNN distribution. Similarly the peak at zero-efficiency, representing interactions that are missed, is smaller. The dense Mask-RCNN surpasses sMask-RCNN in average efficiency as well with 89.1% compared to the slightly lower 85.9%. The dense network covers 87.1% of interactions compared to the sparse's 82.7%, where ”covered” is defined in section 8.5.2.

It is important to note that both versions of the network completely miss true interactions with the same frequency. This is shown by observing the peak at 0



(a) Dense Mask-RCNN

(b) Sparse Mask-RCNN

Figure 8.14: The number of covered interactions is plotted against the number of true interactions. A dashed line along $y = x$ represents the absolute perfect performance, with all true interactions being covered in each event. Out of an average of 20.8 true interactions per event, the dense network covers an average of 18.2, while the sparse network covers 17.2.

efficiency in figure 8.9. There we see that each network misses about 4% of interactions. This means that while the metrics point to worse performance for the sparse network, it still finds the interactions themselves, and still covers them to largely the same extent. The difference is that it builds less complete masks of the true interactions compared to the dense network, though it still finds part of them.

We note that it is difficult to track the effects of these differences on the training and learning of the networks. As such, we cannot distinguish whether the difference in performance of the two networks is due to the change from dense ResNet to sparse ResNet, the training on crops versus entire event images, or a combination of both. However, we emphasize that sMask-RCNN's ability to cluster interactions is sufficient for us to compare its cosmic tagging ability to current methods deployed in MicroBooNE, particularly given the speedup acquired by moving to submanifold convolutions. Further, while the dense network's performance does slightly outperform the sparse network, MicroBooNE's data processing prioritizes speed and the use of CPUs in order to scale to the size of its datasets. Therefore, deploying the dense network is not a viable option. This means that regardless of the performance of the dense version of the network, it is prohibitively slow to run at the scale MicroBooNE's dataset requires. Therefore for the analysis performed in section 8.6 we will only use sMask-RCNN.

8.6 Finding electron neutrinos with sMask-RCNN

In this section, we examine using sMask-RCNN in MicroBooNE to reduce the ratio of cosmic ray background events to electron neutrino events. There are two approaches that we explore, one designed to select neutrino interactions explicitly, and another designed to remove cosmic ray muon interactions. The first, an "identification by positive" approach, would use the neutrino-class output from sMask-RCNN and apply some threshold to select neutrino interactions. This approach is discussed in section 8.6.1.

The other approach, "identification by negative", applies an event veto, which targets cosmic-only events to flag them for removal. Then the remaining events are those with a neutrino, as well as cosmic ray muons interacting in the detector during the beam window. Finally, we note that while this article specifically targets electron neutrinos, the tools discussed can be adjusted to target muon neutrinos in the same way.

Recall that the cosmic ray background in MicroBooNE is very large, and therefore must be dealt with early on in any chain of reconstruction tools. To demonstrate the scale of this problem, we define three different samples:

1. **General Electron Neutrino Sample**: Events containing a simulated electron neutrino interaction combined with cosmic ray muon background data.
2. **Low Energy Electron Neutrino Sample**: The same as sample 1 but only for electron neutrinos with energy less than 400 MeV.
3. **Off-Beam Sample**: Data taken by the detector in anti-coincidence with the neutrino beam. This means there is no beam neutrino interaction present. This sample represents the cosmic ray-only background events.

Table 8.3: The expected ratio of the two different neutrino sample events to off-beam background events.

Type of event	Ratio to off-beam sample events
General electron neutrino	1.52×10^{-2}
Low energy electron neutrino	3.85×10^{-4}

The expected ratios of samples 1 and 2 to sample 3 are depicted in table 8.3. These ratios are instructive, as they indicate the initial signal to cosmic ray background event ratio for analyses that seek to remove cosmic ray-only events. These ratios only depict events. Each event contains O(20) cosmic ray muon interactions, and either zero or one neutrino interaction. So the true ratio of cosmic ray muon interactions to electron neutrino interactions is roughly 20 times higher than the ratio of event types. As the purpose of this article is to develop techniques to reduce the significant cosmic ray muon background, we ignore other backgrounds to an electron neutrino signal, such as muon neutrino events.

While the analysis in section 8.6.1 uses the validation data used thus far in this article, section 8.6.2 uses the general and low energy electron neutrino samples and cosmic ray-only sample described above. The validation data contains only simulated interactions, including CORSIKA-simulated cosmic ray muons. However, the three new datasets contain cosmic ray muons from data, rather than simulation. In previous sections, where we need information about the individual cosmic ray muon interactions for the metrics, it is necessary to use simulated cosmic ray muons. However, this is not the case for the event veto described in section 8.6.2. Therefore it is better to use cosmic ray interactions from data, as there is no reliance on the simulation's ability to properly emulate a cosmic ray muon.

8.6.1 Electron neutrino identification

We examine the electron neutrino "identification by positive" approach by looking at the efficiency and purity as defined in section 8.5.1, broken down by the two class categories: cosmic ray muons and electron neutrinos. Figure 8.15 shows the efficiency metric (defined in eq. 8.1) for simulated electron neutrino interactions, separated by class. The average efficiency is 76.8% for electron neutrinos, and 86.1% for cosmic ray muons. If the network proposes no neutrino interactions, then the efficiency of that event's neutrino interaction is 0. Each class has a peak at 0, but the cosmics also have a peak at 100%, whereas the neutrino interactions peak at just over 90%. In the context of neutrino interactions, the interaction has some number of prongs, where a 'prong' refers to a shower, or track coming out of the neutrino interaction vertex. It is possible that, for neutrino interactions with at least two prongs, the network fails to mark a shorter prong, or partially masks the track-like portion of an electromagnetic shower with both track- and shower-like topology. Particularly long tracks are also difficult to capture completely due to rescaling within sMask-RCNN, which may lead to the ends of the track getting truncated.

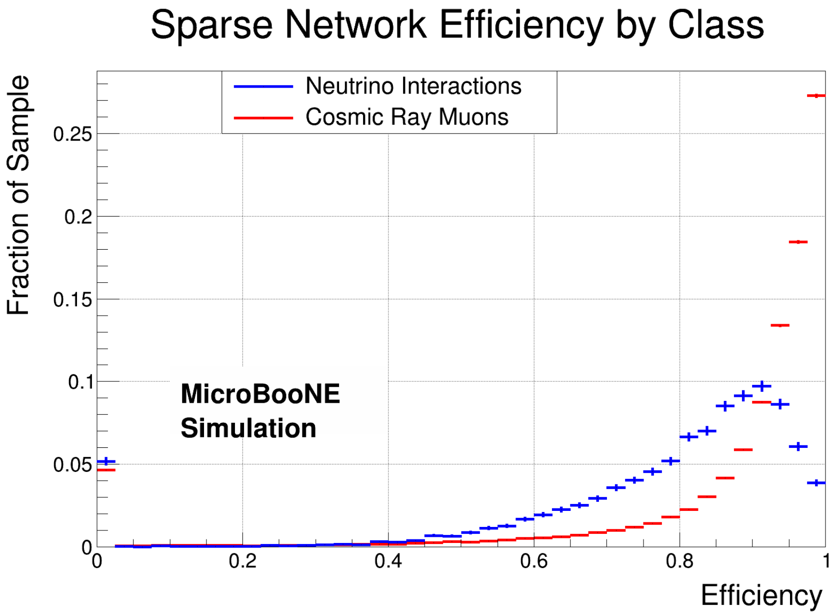


Figure 8.15: The efficiency of sMask-RCNN broken down by class. The average efficiency is 76.8% for electron neutrinos, and 86.1% for cosmic ray muons. Statistical uncertainty bars are shown.

Figure 8.16 is also separated by class, but shows the charge efficiency. The average charge efficiency for electron neutrinos is 77.9%, and 86.8% for cosmic ray muons. Both overall and for each class individually the network has a better charge efficiency than standard efficiency, indicating that the interaction masking prioritizes clustering pixels corresponding to larger charge deposition regardless of class.

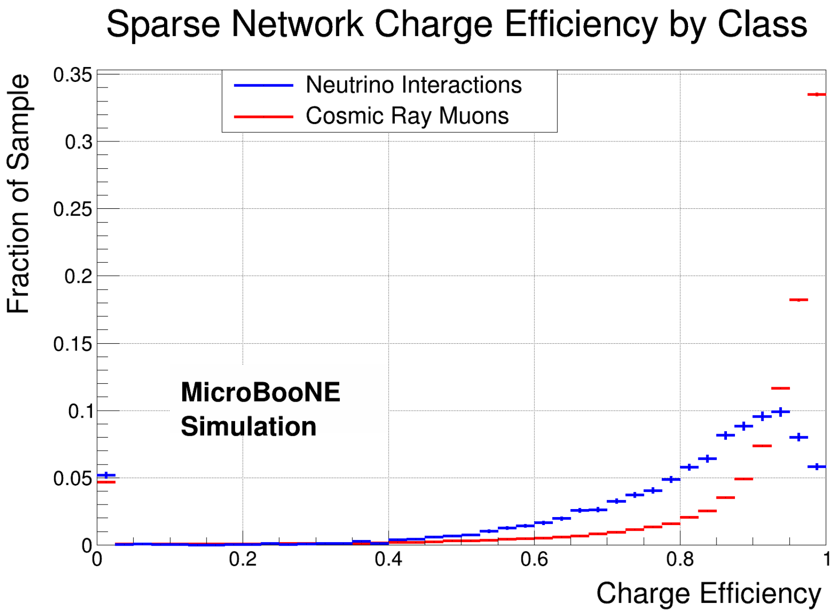


Figure 8.16: The charge efficiency of sMask-RCNN broken down by class. The average charge efficiency for electron neutrinos is 77.9%, and 86.8% for cosmic ray muons. Statistical uncertainty bars are shown.

We see from these two efficiency breakdowns that the network's ability to find an interaction is not strongly tied to the type of interaction, as the peak at zero efficiency is the same for each class. However the masks for cosmic interactions are more complete than those for neutrino interactions.

The purity of sMask-RCNN predicted interactions (defined in eq. 8.2) is broken down by class in figure 8.17. The average purity is 64.9% for electron neutrinos and 84.7% for cosmic ray muons. Here we see an issue with using this version of sMask-RCNN in an identification by positive approach. The peak at zero purity for the neutrino class indicates that, in events that contain simulated neutrino interactions, roughly 22% of predicted interactions labeled neutrinos are actually placed on cosmic ray muons. This implies selecting only predicted neutrino interactions yields a ratio of electron neutrinos to cosmic ray muons of 78:22. However, this only applies if the identification were restricted to events that definitely contain a neutrino. When factoring in the significant number of events in the data that contain no neutrino interaction, as indicated by table 8.3, the number of falsely identified neutrinos grows much worse.

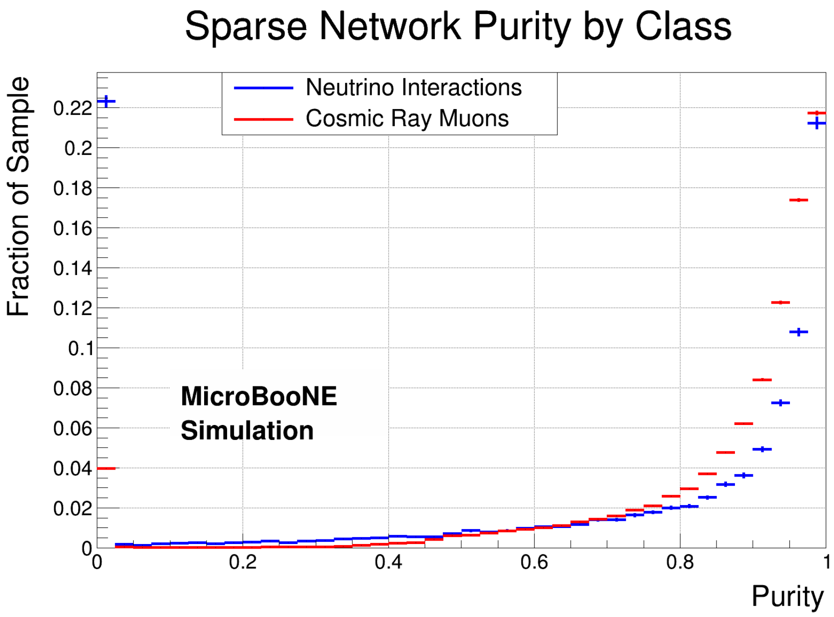


Figure 8.17: The purity of sMask-RCNN broken down by class. Predicted interactions with zero purity are misclassified, for example a true neutrino labeled as a cosmic ray. The average purity is 64.9% for electron neutrinos and 84.7% for cosmic ray muons. Statistical uncertainty bars are shown.

We can imagine ways to improve this identification by positive approach, from increasing the required confidence score the classifier in sMask-RCNN has in a predicted interaction, to retraining the network with increased penalties for falsely predicting the neutrino class. However, these were set aside in favor of exploring the identification by negative approach in the subsequent section.

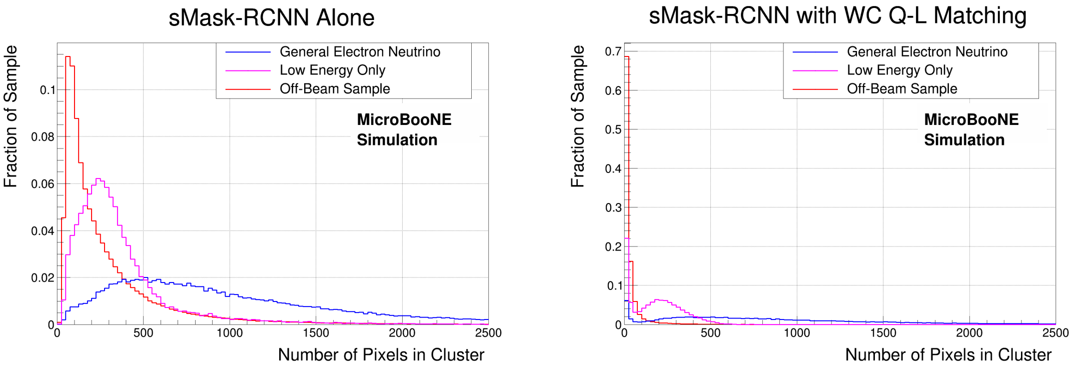
8.6.2 Cosmic-only event veto

The topology of a muon interaction is much more consistent than that of electron neutrino interactions, due to an electron neutrino interaction's variety of final states. A muon creates a track in the detector, and then will either pass through the detector, capture, or create a Michel electron shower at the end, whereas an electron neutrino can interact in argon in many different ways, creating a variety of different daughter particle scenarios. This is consistent with the improved performance of the network labeling cosmic ray muons compared to electron neutrinos, evident in section 8.6.1, where the network must learn to recognize the many different patterns and topologies that make up the electron neutrino class label. Therefore, relying on sMask-RCNN's cosmic ray muon clustering rather than its neutrino clustering may be preferable. As such, the identification by negative approach, which only relies on an understanding of the cosmic ray background, may be more effective.

To study this, we implement an event veto. The goal of this veto is to use sMask-RCNN outputs to separate entire events into those that contain only cosmic ray background, and those that contain an electron neutrino interaction among cosmic rays in the beam window. This task is tested by using this event veto to separate the cosmic ray-only data sample from the general and low energy electron neutrino samples. If this veto were perfect, then the only cosmic ray muons left would be the O(20) interactions per event containing a neutrino interaction. These remaining cosmic ray muons can then be dealt with further down the reconstruction chain.

In order to provide a comparison to current methods used in MicroBooNE, we analyze this event veto using several different versions of cosmic ray tagging. For the first tagger, we include all cosmic ray interaction pixels predicted by sMask-RCNN with a confidence score greater than 0.20. Reducing the confidence score requirement relative to earlier sections of this article allows more cosmic removal at the expense of including multiple overlapping cosmic interaction predictions. Decreasing the confidence score requirement does not significantly impact the electron neutrino efficiency. For individual interaction labeling, shown earlier in the article, this would be problematic. However, for the event veto described below, we are concerned with removing entire events.

For the next tagger, we add the pixels tagged as cosmic ray muons by MicroBooNE's Wire-Cell Q-L described in section 8.3.2. This adds information from the PMT light collection system and the two LArTPC induction planes, none of which is used by sMask-RCNN. Therefore by comparing sMask-RCNN alone to this combined tagger, we can see the additional value provided by the light information, as well as demonstrate the effectiveness of sMask-RCNN operating in a regime with



(a) Cosmic tagging by sMask-RCNN alone (b) With WC Q-L matching

Figure 8.18: The size of the largest cluster found by DBScan after cosmic tagging.

less information. Recall WC Q-L matching is a piece of the full WC cosmic tagger. We repeat the following analysis with the full WC cosmic tagger at the end of this section, exploring two additional tagger configurations.

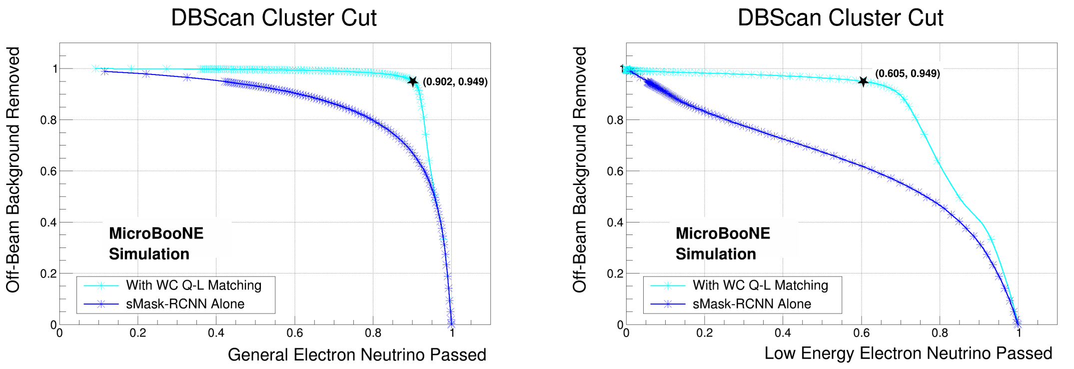
A perfect tagger would tag every pixel containing deposited charge associated with cosmic ray muons. All that would remain in the event image would be pixels holding charge corresponding to a neutrino interaction, if present. However, with the expectation of imperfect performance, we re-cluster the untagged pixels via a ”density-based spatial clustering of applications with noise” (DBScan) algorithm [97]. This means that first we perform our cosmic ray muon tagging, remove those pixels from the image, then run DBScan on the resulting image.

DBScan will output clusters of remaining pixels for each event. These pertain to portions of cosmic ray muon interactions not fully tagged, and the neutrino interaction if present. In the case where both neutrino and cosmic ray muon clusters are present, one large cluster will usually represent the neutrino interaction and several smaller clusters represent the untagged parts of muons. This means the size of the largest cluster is a metric that we can use to isolate events containing neutrino interactions. Figure 8.18 shows the size of the largest of these clusters for sMask-RCNN with and without the WC Q-L matching algorithm. Each figure shows the distribution of the three key samples described above.

Examining the distributions of these three samples, we see a difference between the off-beam sample, which represents the cosmic ray-only event background, and the two electron neutrino samples. Notably, the off-beam sample generally has a smaller number of pixels in the largest cluster found by DBScan. In figure 8.18a, we see that the sMask-RCNN tagger produces a distinctly shaped distribution for each sample. The off-beam sample has a peak closest to zero, the low energy sample is shifted slightly to the right, and the general electron neutrino sample is shifted further. This reinforces the expectation, as the neutrino pixels remain in the image untagged, allowing for DBScan to find larger clusters. Examining sMask-RCNN with the WC Q-L matching algorithm in figure 8.18b, we see a shift to the left in all three distributions, placing a strong peak at zero, indicating that the combined tagger frequently labels more pixels as belonging to cosmic ray muon interactions than the sMask-RCNN-based tagger alone. This is particularly notable in the case of the off-beam sample, where the peak at zero accounts for almost 70% of the sample, compared to about 16% of the low energy sample, and about 6% of the general electron neutrino sample.

Using this metric for the size of the largest remaining cluster, we can create a receiver operating characteristic (ROC) curve to demonstrate the efficiency and rejection power of the different cosmic tagging methods when applied as an event veto. A ROC curve is a measure of signal retention or signal efficiency on one axis, and background rejection on the other. A curve is created by incrementing some requirement, which slowly decreases retention and increases rejection. Ideally the curve has points in the upper right region of the plot such that signal retention and background rejection are both high.

We reiterate that we want to explore this quantity — largest remaining cluster — as an event-level discriminant. As such, it will be the value we increment to create ROC curves. Specifically, these curves are made by applying a requirement to filter events that do not contain a DBScan cluster of size greater than X, where X is incremented from no requirement, to 0, and then incremented by 10 pixels thereafter. The resulting signal retention and background rejection rates give points for the curves. In figure 8.19, we show the ROC curves for the signal retentions of



(a) General electron neutrino sample (b) Low energy electron neutrino sample

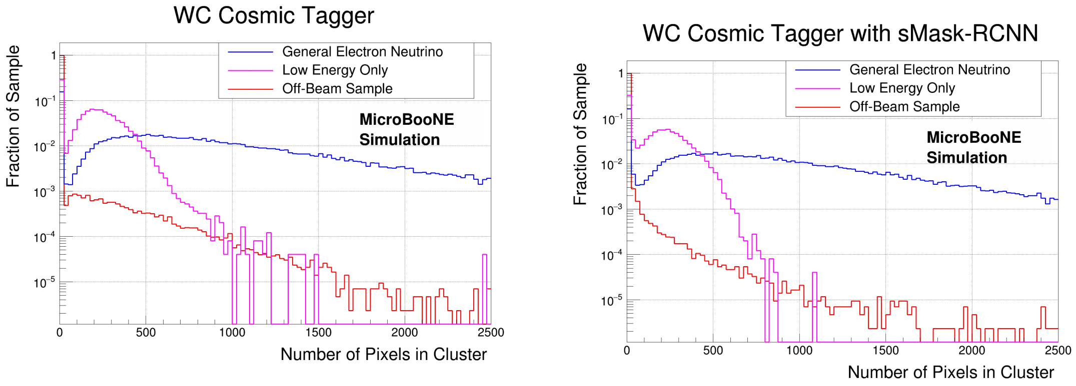
Figure 8.19: ROC curves for the sMask-RCNN with and without WC Q-L matching based on a requirement on largest cluster size. Curves are shown for the two electron neutrino samples against the off-beam background.

the general and low energy electron neutrino samples versus the rejection of the off-beam cosmic ray muon background sample. Curves are made for the two different tagging methods. We observe that the combined version of a cosmic tagger using both sMask-RCNN and WC Q-L matching yields a better combination of signal efficiency and background rejection.

For the low energy sample each tagging method performs worse compared to the general sample. However, this is not surprising as the lower energy electron neutrino interactions correspond to less charge deposited in the event image and fewer neutrino pixels in the event. This means that the remaining DBScan clusters related to the lower energy neutrino interactions will be smaller, and harder to isolate from the untagged off-beam cosmic ray muon sample's distribution.

Examining the ROC curve for sMask-RCNN with WC Q-L matching in figure 8.19a, we are able to achieve a general electron neutrino signal efficiency of 90.2% while rejecting 94.9% of the off-beam cosmic background if we remove events that do not have a cluster of at least 130 pixels after the taggers are run. For the low energy electron neutrino sample in figure 8.19b we can achieve a signal efficiency of 60.5% for the same requirement, though a reduced requirement on remaining cluster size could be applied to increase the efficiency at the cost of rejection power, as indicated by the combined curve.

We also examine the effect of adding sMask-RCNN's cosmic finding to the state-



(a) Cosmic tagging by the WC cosmic tagger alone

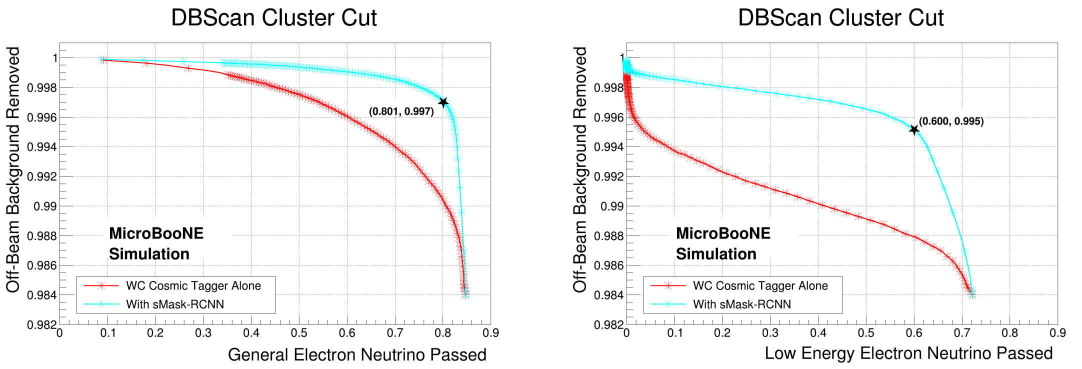
(b) With sMask-RCNN

Figure 8.20: The size of the largest cluster found by DBScan after cosmic tagging. Note the logarithmic scale.

of-the-art complete WC cosmic tagger. This means that we take the event vetoes and cosmic tagging of the WC cosmic tagger, and add the cosmic tagging of sMask-RCNN to get a combined tagger. In figure 8.20 we show the distributions of the largest cluster found by DBScan after running the WC cosmic tagger with and without the cosmic ray muons found by sMask-RCNN. For the events that are rejected by one of the WC cosmic tagger event vetos, the largest DBScan cluster is defined to be zero.

The distributions of cluster size before and after adding sMask-RCNN to the WC cosmic tagger show a shift to the left in the shape of the off-beam sample, indicating the added value of sMask-RCNN in cosmic tagging. Without sMask-RCNN, there appears to be a slight peak beyond zero that gets shifted to zero after sMask-RCNN is added. We observe minimal shift in the two electron neutrino samples and each distribution still has a clear second peak separate from zero.

ROC curves for the WC cosmic tagger with and without sMask-RCNN are shown in figure 8.21. However, as the WC cosmic taggers introduces several of its own event vetos, the point referring to the loosest cut, with the most signal passed, does not allow all events through the veto. Instead it starts with the signal efficiency and background rejection of the WC cosmic tagger, and adjusts as we increase the strength of the DBScan cluster size requirement. We reiterate that the difference between figures 8.21 and 8.19 is the additional event vetos added to WC Q-L matching



(a) General electron neutrino sample (b) Low energy electron neutrino sample

Figure 8.21: ROC curves for the WC cosmic tagger with and without sMask-RCNN based on a requirement on largest cluster size. Curves are shown for the two electron neutrino samples against the off-beam background. Note the suppressed y-axis shown here demonstrates significant improvement in background removal compared to figure 8.19.

to create the WC cosmic tagger.

Examining these ROC curves we see that including sMask-RCNN on top of the WC cosmic tagger does yield improvement to the background rejection at equivalent signal efficiencies. In the general electron neutrino sample, for a signal efficiency of 80.1% the WC cosmic tagger rejects 99.0% of the background, whereas adding sMask-RCNN rejects 99.7% of the background at the same signal efficiency. This represents a reduction of the remaining background by 70%. For the low energy neutrino sample, a similar effect is seen, albeit at lower signal efficiencies.

In order to evaluate the difference between these two rejection strengths, it is important to recall the imbalance between off-beam background events and electron neutrino signal events described in table 8.3. The 99.0% rejection provided by the WC cosmic tagger corresponds to a general electron neutrino signal to off-beam background ratio of 1.26, a vast improvement over the starting ratio in the table. However, the addition of sMask-RCNN increases this signal to background ratio to 4.14 by improving the rejection power to 99.7%. For this same selection, the low energy signal to background ratio is 0.015 with the WC cosmic tagger, and 0.56 with after the addition of sMask-RCNN.

8.7 Conclusions

This article demonstrates a novel approach to cosmic ray muon tagging using sMask-RCNN. We demonstrate the ability of this network to locate, identify, and cluster particle interactions in the MicroBooNE LArTPC. We analyze the ability to cluster both the topologically simple cosmic ray muon interactions, as well as highly varied electron neutrino interactions.

We modify the original Mask-RCNN framework by substituting sparse submanifold convolutions in the ResNet portion of the network to create sMask-RCNN. Due to the low pixel occupancy of MicroBooNE event image data this leads to a 20× speedup in ResNet processing time on a CPU, as well as decreased runtime memory usage. This improvement is critical in allowing sMask-RCNN to be deployed as a reconstruction tool on CPU farms to scale to high volume data samples that particle physics experiments typically employ.

This analysis also includes several versions of an event veto. The strongest of these demonstrates that adding sMask-RCNN to the state-of-the-art WC cosmic tagger which is currently used in MicroBooNE reduces the cosmic ray-only event background by a further 70% and increases the signal to background ratio of electron neutrino events to cosmic ray-only events by more than a factor of three. This means that application of this technique to future measurements in MicroBooNE will result in improvements over current MicroBooNE reconstruction.

8.8 Acknowledgements

This document was prepared by the MicroBooNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. MicroBooNE is supported by the following: the U.S. Department of Energy, Office of Science, Offices of High Energy Physics and Nuclear Physics; the U.S. National

Science Foundation; the Swiss National Science Foundation; the Science and Technology Facilities Council (STFC), part of the United Kingdom Research and Innovation; the Royal Society (United Kingdom); and The European Union's Horizon 2020 Marie Sklodowska-Curie Actions. Additional support for the laser calibration system and cosmic ray tagger was provided by the Albert Einstein Center for Fundamental Physics, Bern, Switzerland. We also acknowledge the contributions of technical and scientific staff to the design, construction, and operation of the MicroBooNE detector as well as the contributions of past collaborators to the development of MicroBooNE analyses, without whom this work would not have been possible.

Chapter 9

Conclusion

Within this thesis, software and analysis I have contributed to the MicroBooNE Collaboration has been outlined. In order to provide adequate context for this work, the efforts and tools of many other collaborators are also described. If it were not for the work of these collaborators, or the researchers who came before me, this thesis would not be possible.

In the early parts of the MicroBooNE DL reconstruction chain, I developed and implemented the code that adapts the Wire-Cell team's charge-light matching algorithm for use by the DL analysis. This involved taking the Wire-Cell team's output formats, and restructuring them to fit as a module within the existing DL analysis framework, and validate the performance was the same.

For tagging cosmic ray muon background interactions, and discriminating events lacking a neutrino interaction, I have developed, analyzed and implemented the sMask-RCNN Deep Learning tool using a novel sparse submanifold convolutional ResNet. With the help of Felix J. Yu, a then-undergraduate of Tufts University, this work is under review by JINST as a paper, with preprint available [81]. Within it we demonstrate that adding the network to the existing Wire-Cell Cosmic tagger, under the selection criteria shown, reduces the remaining cosmic-ray-only background by a further 70%.

For the $1l1p$ analyses leading to [14], I took over the $1\mu1p$ selection from Davio Cianci. The $1\mu1p$ selection was modified as the selection was finalized in order to optimize our DL LEE search. I then validated the performance of the $1\mu1p$ selection which then provided a constraint for our LEE result.

I then took this $1\mu1p$ work, and performed a 3+1 sterile neutrino search. Using this high-purity selection of ν_μ events I developed MicroBooNE's sensitivity to ν_μ disappearance using the $1\mu1p$ selection, while performing many validation studies. This disappearance search was developed using both Wilks' theorem, as well as a Feldman Cousins method of developing a critical test statistic value. While the analysis sensitivities are not world-leading, they, along with the $1\mu1p$ selection demonstrate a high-purity-based disappearance search using LArTPC technology with good calorimetry and spatial reconstruction, despite being optimized for the LEE search.

Finally, this analysis was applied to data, where we establish an excluded region in our model parameter search space at 90% confidence. Our excluded region adds additional power to existing experimentally excluded regions in the past. Both the null oscillation model and the global best fit model for 3+1 sterile-based muon neutrino disappearance remain in MicroBooNE's allowed region. MicroBooNE's own best fit model is found to be in a highly degenerate set of models which describe minimal disappearance, indistinguishable from the null oscillation model.

Chapter 10

Bibliography

[1] A. B. Arbuzov, “Quantum field theory and the electroweak standard model,” 2018.

[2] A. de Gouvea, “On determining the neutrino mass hierarchy,” 2006.

[3] P. Zyla *et al.*, “Review of Particle Physics,” *PTEP*, vol. 2020, no. 8, p. 083C01, 2020.

[4] M. Antonello *et al.*, “A Proposal for a Three Detector Short-Baseline Neutrino Oscillation Program in the Fermilab Booster Neutrino Beam,” 3 2015.

[5] A. Aguilar-Arevalo *et al.*, “The miniboone detector,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 599, p. 28–46, Feb 2009.

[6] “Miniboone phototube support structure installation.”

[7] A. A. Aguilar-Arevalo *et al.*, “Updated MiniBooNE neutrino oscillation results with increased data and new background studies,” *Physical Review D*, vol. 103, mar 2021.

[8] A. A. Aguilar-Arevalo *et al.*, “Neutrino flux prediction at miniboone,” *Physical Review D*, vol. 79, Apr 2009.

[9] M. Collaboration, “Booster neutrino flux prediction at microboone.” `https://microboone.fnal.gov/wp-content/uploads/MICROBOONE-NOTE-1031-PUB.pdf`, 2018. Accessed: 2021-03-22.

[10] R. Acciarri *et al.*, “Design and construction of the MicroBooNE detector,” *Journal of Instrumentation*, vol. 12, pp. P02017–P02017, feb 2017.

[11] B. J. P. Jones, *Sterile Neutrinos in Cold Climates.* PhD thesis, MIT, 2015.

[12] A. Shaw, “Gpu.” `https://course19.fast.ai/gpu_tutorial.html`. Accessed: 2022-04-01.

[13] P. Abratenko *et al.*, “Convolutional neural network for multiple particle identification in the microboone liquid argon time projection chamber,” *Phys. Rev. D*, vol. 103, p. 092003, May 2021.

[14] MicroBooNE Collaboration *et al.*, "Search for an anomalous excess of charged-current quasi-elastic nue interactions with the microboone experiment using deep-learning-based reconstruction," 2021.

[15] A. A. Aguilar-Arevalo *et al.*, "A Search for muon neutrino and antineutrino disappearance in MiniBooNE," *Phys. Rev. Lett.*, vol. 103, p. 061802, 2009.

[16] C. Athanassopoulos *et al.*, "Evidence for $\overline{\nu}_\mu \rightarrow \overline{\nu}_e$ oscillations from the lsnd experiment at the los alamos meson physics facility," *Phys. Rev. Lett.*, vol. 77, pp. 3082–3085, Oct 1996.

[17] D. H. Perkins, *Introduction to High Energy Physics.* Cambridge University Press, 4 ed., 2000.

[18] R. Aaij *et al.*, "Observation of particles composed of five quarks, pentaquark-charmonium states, seen in decays," *Physical Review Letters*, vol. 115, aug 2015.

[19] M. Thomson, *Modern Particle Physics.* India: Cambridge University Press, 4 ed., 2013.

[20] B. T. Cleveland, T. Daily, J. Raymond Davis, J. R. Distel, K. Lande, C. K. Lee, P. S. Wildenhain, and J. Ullman, "Measurement of the solar electron neutrino flux with the homestake chlorine detector," *The Astrophysical Journal*, vol. 496, pp. 505–526, mar 1998.

[21] S. Fukuda *et al.*, "Solar ^8b and hep neutrino measurements from 1258 days of super-kamiokande data," *Phys. Rev. Lett.*, vol. 86, pp. 5651–5655, Jun 2001.

[22] Q. R. Ahmad *et al.*, "Direct evidence for neutrino flavor transformation from neutral-current interactions in the sudbury neutrino observatory," *Phys. Rev. Lett.*, vol. 89, p. 011301, Jun 2002.

[23] A. B. Balantekin and B. Kayser, "On the properties of neutrinos," *Annual Review of Nuclear and Particle Science*, vol. 68, no. 1, pp. 313–338, 2018.

[24] P. F. de Salas, S. Gariazzo, O. Mena, C. A. Ternes, and M. Tórtola, "Neutrino mass ordering from oscillations and beyond: 2018 status and future prospects," *Frontiers in Astronomy and Space Sciences*, vol. 5, 2018.

[25] C. Giganti, S. Lavignac, and M. Zito, "Neutrino oscillations: The rise of the PMNS paradigm," *Progress in Particle and Nuclear Physics*, vol. 98, pp. 1–54, jan 2018.

[26] Q. R. Ahmad *et al.*, "Measurement of the rate ofed→ppe-interactions produced byB8solar neutrinos at the sudbury neutrino observatory," *Physical Review Letters*, vol. 87, jul 2001.

[27] K. Abe *et al.*, "Solar neutrino results in super-kamiokande-III," *Physical Review D*, vol. 83, mar 2011.

[28] S. Abe *et al.*, "Precision measurement of neutrino oscillation parameters with KamLAND," *Physical Review Letters*, vol. 100, jun 2008.

[29] F. An *et al.*, “New measurement of13via neutron capture on hydrogen at daya bay,” *Physical Review D*, vol. 93, apr 2016.

[30] S.-B. Kim, “RENO for neutrino mixing angle Theta(13),” *Prog. Part. Nucl. Phys.*, vol. 64, pp. 346–347, 2010.

[31] T. D. C. Collaboration, “Double Chooz 13 measurement via total neutron capture detection ,” *Nature*, vol. 16, pp. 558–564, 2020.

[32] S. Bose and A. Raychaudhuri, “The neutrino mass implications of the k2k experiment,” *Journal of Physics G: Nuclear and Particle Physics*, vol. 29, pp. 1069–1074, apr 2003.

[33] T. Nakaya and R. K. Plunkett, “Neutrino oscillations with the MINOS, MINOS, t2k, and NOvA experiments,” *New Journal of Physics*, vol. 18, p. 015009, jan 2016.

[34] V. Takhistov, “Review of atmospheric neutrino results from super-kamiokande,” 2020.

[35] B. Abi *et al.*, “Deep underground neutrino experiment (dune), far detector technical design report, volume i: Introduction to dune,” 2020.

[36] T. Katori, “The microboone light collection system,” *Journal of Instrumentation*, vol. 8, p. C10011–C10011, Oct 2013.

[37] D. Cianci, *A Deep-Learning-Based Muon Neutrino CCQE Selection for Searches Beyond the Standard Model with MicroBooNE.* PhD thesis, Columbia U., 2021.

[38] J. S. Moon, *Using Deep Learning Techniques to Search for the MiniBooNE Low Energy Excess in MicroBooNE with* $> 3\sigma$ *Sensitivity.* PhD thesis, MIT, 2020.

[39] C. Adams *et al.*, “Design and construction of the microboone cosmic ray tagger system,” *Journal of Instrumentation*, vol. 14, p. P04004–P04004, Apr 2019.

[40] R. Acciarri *et al.*, “Noise characterization and filtering in the MicroBooNE liquid argon TPC,” *Journal of Instrumentation*, vol. 12, pp. P08003–P08003, aug 2017.

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” 2014.

[42] O. Adriani *et al.*, “Determination of the number of light neutrino species,” *Physics Letters B*, vol. 292, no. 3, pp. 463–471, 1992.

[43] A. Diaz, C. Argüelles, G. Collin, J. Conrad, and M. Shaevitz, “Where are we with light sterile neutrinos?,” *Physics Reports*, vol. 884, pp. 1–59, nov 2020.

[44] A. Aguilar *et al.*, “Evidence for neutrino oscillations from the observation of electron anti-neutrinos in a muon anti-neutrino beam,” *Physical Review D*, vol. 64, nov 2001.

[45] A. A. Aguilar-Arevalo *et al.*, “Significant excess of electronlike events in the MiniBooNE short-baseline neutrino experiment,” *Physical Review Letters*, vol. 121, nov 2018.

[46] Y. Lecun and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 536–444, May 2015.

[47] V.-G. Dhillon, A., “Convolutional neural network: a review of models, methodologies and applications to object detection.,” *Prog Artif Intell*, vol. 9, pp. 85–112, 2020.

[48] K. Chellapilla, S. Puri, and P. Simard, “High Performance Convolutional Neural Networks for Document Processing,” in *Tenth International Workshop on Frontiers in Handwriting Recognition* (G. Lorette, ed.), (La Baule (France)), Université de Rennes 1, Suvisoft, Oct. 2006. http://www.suvisoft.com.

[49] J. Dsouza, “What is a GPU and do you need one in Deep Learning?.” `https://towardsdatascience.com/what-is-a-gpu-and-do-you-need-one-in-deep-learning-718b9597aa0d`, 2018. Accessed: 2022-04-1.

[50] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

[51] P. Abratenko *et al.*, “Neutrino event selection in the MicroBooNE liquid argon time projection chamber using wire-cell 3d imaging, clustering, and charge-light matching,” *Journal of Instrumentation*, vol. 16, p. P06043, jun 2021.

[52] C. Adams *et al.*, “Deep neural network for pixel-level electromagnetic particle identification in the microboone liquid argon time projection chamber,” *Phys. Rev. D*, vol. 99, p. 092001, May 2019.

[53] P. Abratenko *et al.*, “Semantic segmentation with a sparse convolutional neural network for event reconstruction in microboone,” *Phys. Rev. D*, vol. 103, p. 052012, Mar 2021.

[54] V. Genty, *The MicroBooNE Search For Anomalous Electron Neutrino Appearance Using Image Based Data Reconstruction.* PhD thesis, Columbia U., 2019.

[55] P. Abratenko *et al.*, “Vertex-finding and reconstruction of contained two-track neutrino events in the microboone detector,” *Journal of Instrumentation*, vol. 16, p. P02017–P02017, Feb 2021.

[56] P. Abratenko *et al.*, “Electromagnetic shower reconstruction and energy validation with michel electrons and 0 samples for the deep-learning-based analyses in microboone,” *Journal of Instrumentation*, vol. 16, p. T12017, Dec 2021.

[57] C. Andreopoulos, C. Barry, S. Dytman, H. Gallagher, T. Golan, R. Hatcher, G. Perdue, and J. Yarba, “The genie neutrino monte carlo generator: Physics and user manual,” 2015.

[58] S. Agostinelli *et al.*, "GEANT4–a simulation toolkit," *Nucl. Instrum. Meth. A*, vol. 506, pp. 250–303, 2003.

[59] D. Heck, J. Knapp, J. N. Capdevielle, G. Schatz, and T. Thouw, "CORSIKA: A Monte Carlo code to simulate extensive air showers," 2 1998.

[60] C. Andreopoulos, A. Bell, D. Bhattacharya, F. Cavanna, J. Dobson, S. Dytman, H. Gallagher, P. Guzowski, R. Hatcher, P. Kehayias, A. Meregaglia, D. Naples, G. Pearce, A. Rubbia, M. Whalley, and T. Yang, "The GENIE neutrino monte carlo generator," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 614, pp. 87–104, feb 2010.

[61] L. Alvarez-Ruso, C. Andreopoulos, A. Ashkenazi, C. Barry, S. Dennis, S. Dytman, H. Gallagher, A. A. G. Soto, S. Gardiner, W. Giele, R. Hatcher, O. Hen, L. Jiang, I. D. Kakorin, K. S. Kuzmin, A. Meregaglia, V. A. Naumov, A. Papadopoulou, M. Roda, V. Syrotenko, J. Tena-Vidal, J. Wolcott, N. Wright, M. Kabirnezhad, and N. Vololoniaina, "Recent highlights from GENIE v3," *The European Physical Journal Special Topics*, vol. 230, pp. 4449–4467, dec 2021.

[62] J. Tena-Vidal, C. Andreopoulos, A. Ashkenazi, C. Barry, S. Dennis, S. Dytman, H. Gallagher, S. Gardiner, W. Giele, R. Hatcher, O. Hen, L. Jiang, I. D. Kakorin, K. S. Kuzmin, A. Meregaglia, V. A. Naumov, A. Papadopoulou, G. Perdue, M. Roda, V. Syrotenko, and J. W. and, "Neutrino-nucleon cross-section model tuning in GENIE v3," *Physical Review D*, vol. 104, oct 2021.

[63] J. Nieves, I. R. Simo, and M. J. V. Vacas, "Inclusive charged-current neutrino-nucleus reactions," *Physical Review C*, vol. 83, apr 2011.

[64] A. A. Aguilar-Arevalo *et al.*, "First measurement of the muon neutrino charged current quasielastic double differential cross section," *Physical Review D*, vol. 81, may 2010.

[65] The MicroBooNE Collaboration *et al.*, "New theory-driven genie tune for microboone," 2021.

[66] K. Abe *et al.*, "Measurement of double-differential muon neutrino charged-current interactions on c_8h_8 without pions in the final state using the t2k off-axis beam," *Phys. Rev. D*, vol. 93, p. 112012, Jun 2016.

[67] A. M. Ankowski and J. T. Sobczyk, "Argon spectral function and neutrino interactions," *Phys. Rev. C*, vol. 74, p. 054316, Nov 2006.

[68] T. Katori, "Meson exchange current (mec) models in neutrino interaction generators," 2013.

[69] L. E. Yates, *Using the MicroBooNE Liquid Argon Detector to Search for Electron Neutrino Interactions and Understand the MiniBooNE Anomaly.* PhD thesis, MIT, 2022.

[70] MicroBooNE Collaboration *et al.*, "Novel approach for evaluating detector-related uncertainties in a lartpc using microboone data," 2021.

[71] D. W. Scott, *Kernel Density Estimation*, pp. 1–7. American Cancer Society, 2018.

[72] L. Aliaga *et al.*, "Neutrino flux predictions for the NuMI beam," *Physical Review D*, vol. 94, nov 2016.

[73] P. Abratenko *et al.*, "Measurement of the flux-averaged inclusive charged-current electron neutrino and antineutrino cross section on argon using the numi beam and the microboone detector," *Phys. Rev. D*, vol. 104, p. 052002, Sep 2021.

[74] P. Abratenko *et al.*, "First measurement of inclusive muon neutrino charged current differential cross sections on argon at e$\sim$0.8 GeV with the MicroBooNE detector," *Physical Review Letters*, vol. 123, sep 2019.

[75] J. Calcutt, C. Thorpe, K. Mahn, and L. Fields, "Geant4reweight: a framework for evaluating and propagating hadronic interaction uncertainties in geant4," *Journal of Instrumentation*, vol. 16, p. P08042, aug 2021.

[76] X. Ji, W. Gu, H. Wei, and C. Zhang, "Combined neyman-pearson chi-square: An improved approximation to the poisson-likelihood chi-square," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 961, p. 163677, 02 2020.

[77] G. J. Feldman and R. D. Cousins, "A Unified approach to the classical statistical analysis of small signals," *Phys. Rev. D*, vol. 57, pp. 3873–3889, 1998.

[78] D. Cianci and M. Ross-Lonergan, "SBNfit." Available at `https://github.com/NevisUB/whipping_star`(2021).

[79] MicroBooNE Collaboration, "Microboone low-energy excess signal prediction from unfolding miniboone monte-carlo and data." MicroBooNE Public Note 1043, 2018.

[80] I. E. Stockdale, A. Bodek, F. Borcherding, N. Giokaris, K. Lang, D. Garfinkle, F. S. Merritt, M. Oreglia, P. Reutens, P. S. Auchincloss, R. Blair, C. Haber, M. Ruiz, F. Sciulli, M. H. Shaevitz, W. H. Smith, R. Zhu, R. Coleman, H. E. Fisk, B. Jin, D. Levinthal, W. Marsh, P. A. Rapidis, H. B. White, and D. Yovanovitch, "Limits on muon-neutrino oscillations in the mass range $30lt;\Delta m^2 lt; 1000$ ev$^2/c^4$," *Phys. Rev. Lett.*, vol. 52, pp. 1384–1388, Apr 1984.

[81] MicroBooNE Collaboration, "Cosmic ray muon clustering for the microboone liquid argon time projection chamber using smask-rcnn," 2022.

[82] R. Acciarri *et al.*, "Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber," *Journal of Instrumentation*, vol. 12, pp. P03011–P03011, mar 2017.

[83] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

[84] B. Graham, "Spatially-sparse convolutional neural networks," *CoRR*, vol. abs/1409.6070, 2014.

[85] C. Adams *et al.*, "Ionization electron signal processing in single phase LArTPCs. Part I. Algorithm Description and quantitative evaluation with MicroBooNE simulation," *JINST*, vol. 13, no. 07, p. P07006, 2018.

[86] C. Adams *et al.*, "Ionization electron signal processing in single phase LArTPCs. part II. data/simulation comparison and performance in MicroBooNE," vol. 13, pp. P07007–P07007, jul 2018.

[87] R. Acciarri *et al.*, "Cosmic ray background removal with deep neural networks in sbnd," *Frontiers in Artificial Intelligence*, vol. 4, p. 42, 2021.

[88] R. Acciarri *et al.*, "The Pandora multi-algorithm approach to automated pattern recognition of cosmic-ray muon and neutrino events in the MicroBooNE detector," *Eur. Phys. J. C*, vol. 78, no. 1, p. 82, 2018.

[89] J. S. Marshall and M. A. Thomson, "The Pandora Software Development Kit for Pattern Recognition," *Eur. Phys. J. C*, vol. 75, no. 9, p. 439, 2015.

[90] P. Abratenko *et al.*, "Cosmic Ray Background Rejection with Wire-Cell LArTPC Event Reconstruction in the MicroBooNE Detector," *Phys. Rev. Applied*, vol. 15, no. 6, p. 064071, 2021.

[91] P. Abratenko *et al.*, "High-performance Generic Neutrino Detection in a LArTPC near the Earth's Surface with the MicroBooNE Detector," 12 2020.

[92] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[93] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[94] Fermilab, "Grid computing." `https://www.fnal.gov/pub/science/computing/grid.html`, 2015. Accessed: 2021-09-17.

[95] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[96] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollar, "Panoptic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[97] M. Ester, H. P. Kriegel, J. Sander, and X. Xiaowei, *A density-based algorithm for discovering clusters in large spatial databases with noise.* 12 1996.

[98] MicroBooNE Collaboration, "Event selection in the microboone deep learning based low energy excess analysis using two-body scattering criteria." MicroBooNE Public Note 1086, 2020.

Appendix A

Boosting

Here we further describe the procedure for boosting into the nucleon rest frame. This is beneficial because, while the kinematic variables computed in the lab frame can be useful, they ignore the Fermi motion of the nucleon struck by the neutrino in a quasi-elastic (QE) interaction.

Fermi motion describes the quantum motion of the nucleons within the argon nucleus of the LArTPC. When the neutrino strikes the nucleus additional momentum gets added to the interaction, albeit small.

$$\vec{p_\nu} + \vec{p}_{fermi} = \vec{p}_{final} \tag{A.1}$$

In the selections described in section 6 the topology of the targeted interactions is assumed to be $1l1p$. As such, $\vec{p}_{final}$ contains one lepton (electron or muon based on selection) and one proton. This gives us

$$\begin{aligned}(0,0,p_\nu) + (p_f^x, p_f^y, p_f^z) &= (p_p^x, p_p^y, p_p^z) + (p_l^x, p_l^y, p_l^z) \\ \rightarrow \vec{p}_f &= (p_p^x + p_l^x, p_p^y + p_l^y, p_p^z + p_l^z - p_\nu) \\ \rightarrow \vec{p}_f &\approx (p_p^x + p_l^x, p_p^y + p_l^y, p_p^z + p_l^z - E_\nu)\end{aligned} \tag{A.2}$$

Here we make the easy approximation that m_ν is small enough that $p_\nu \approx E_\nu$ and that the neutrino's momentum is completely along the z-axis, along the beam. We then acquire the Fermi momentum.[1]

Now, in order to boost into the nucleon's rest frame all we need is the energy of the nucleon itself, E_n. This is defined as

$$\begin{aligned} E_n = m_n - E_b - T_f & \\ & T_f << (m_n - E_b) \\ & \approx m_n - E_b. \end{aligned} \tag{A.3}$$

Here E_b is the removal energy and T_f is the final state nuclear recoil kinetic energy, which is assumed to be small relative to the other quantities' difference. Now that the nucleon's energy is reconstructed, we can calculate the boost vector β

$$\vec{\beta} = \frac{\vec{p_f}}{E_N} \tag{A.4}$$

With the boost vector calculated we can boost our reconstructed quantities into the nucleon's rest frame. This has the benefit of reducing the risk associated with assuming a stationary target nucleon and ignoring Fermi motion. Equation A.5 shows the procedure for performing this boost.

$$\begin{pmatrix} E' \\ p_x' \\ p_y' \\ p_z' \end{pmatrix} = \begin{pmatrix} \gamma & -\gamma\beta_x & -\gamma\beta_y & -\gamma\beta_z \\ -\gamma\beta_x & 1+\frac{\gamma-1}{\beta^2}\beta_x^2 & \frac{\gamma-1}{\beta^2}\beta_x\beta_y & \frac{\gamma-1}{\beta^2}\beta_x\beta_z \\ -\gamma\beta_y & \frac{\gamma-1}{\beta^2}\beta_x\beta_y & 1+\frac{\gamma-1}{\beta^2}\beta_y^2 & \frac{\gamma-1}{\beta^2}\beta_y\beta_z \\ -\gamma\beta_z & \frac{\gamma-1}{\beta^2}\beta_x\beta_z & \frac{\gamma-1}{\beta^2}\beta_y\beta_z & 1+\frac{\gamma-1}{\beta^2}\beta_z^2 \end{pmatrix} \begin{pmatrix} E \\ p_x \\ p_y \\ p_z \end{pmatrix} \tag{A.5}$$

Reference [98] demonstrates studies showing the improvements in the 'QE-ness' of interactions studied in the nucleon's rest frame rather than the lab frame. It

---

[1] The Fermi momentum coming from the struck nucleon is random, adding momentum into the interaction isotropically

is also worth noting that due to the imperfect nature of reconstructure it can be possible to calculate a non-physical β value larger than 1. In these cases events fail the 'boostability' cut described in section 6.3.

Appendix B

$1\mu1p$ BDT Selection Variable Distributions

In this appendix we display the distributions of the different variables related to the $1\mu1p$ BDT Ensemble. These distributions represent events that are contained in the final selection. The legends show the breakdown of events in different categories for the stacked histograms contained within the plots. However the legends only include events included within the x-axis range of a given plot.

Note that some of the variables, such as the x, y, and z reconstructed positions are not used as inputs to the selection's BDT ensemble, and are included to show observation-expectation agreement and general sample distribution information. For a list of the variables used in the BDT ensemble, refer to table 6.3.

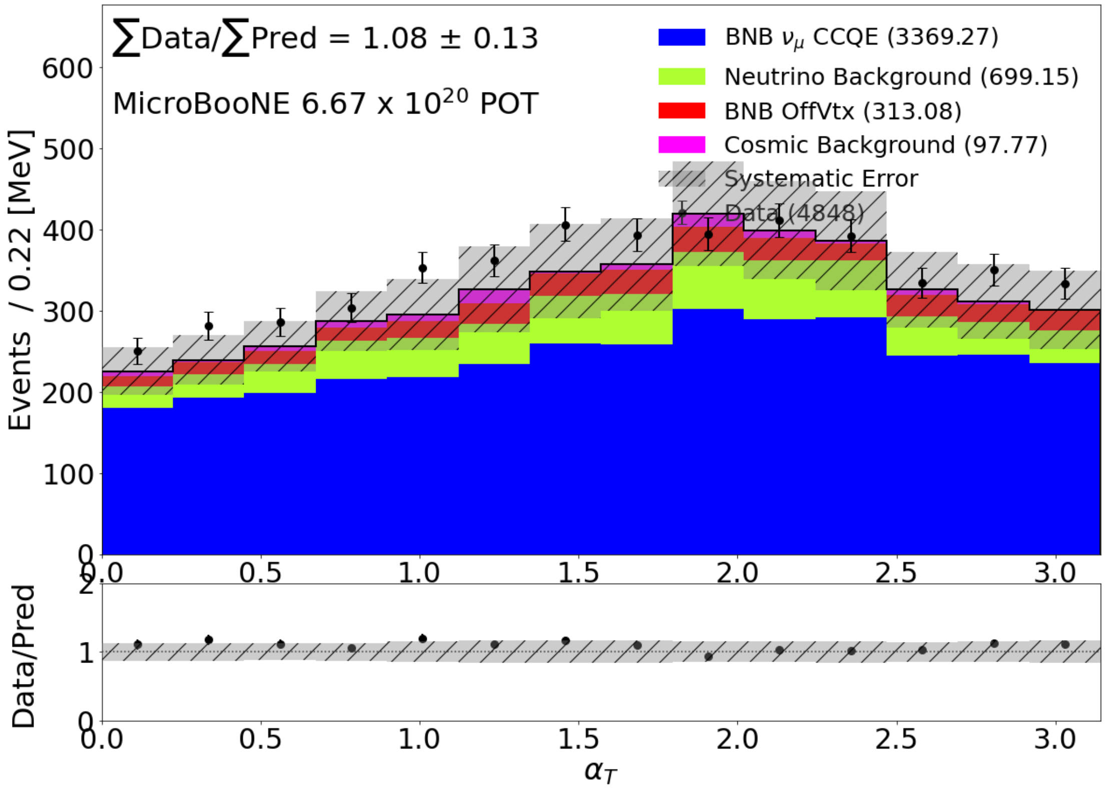


Figure B.1: The α_T distribution for the $1\mu1p$ selection.

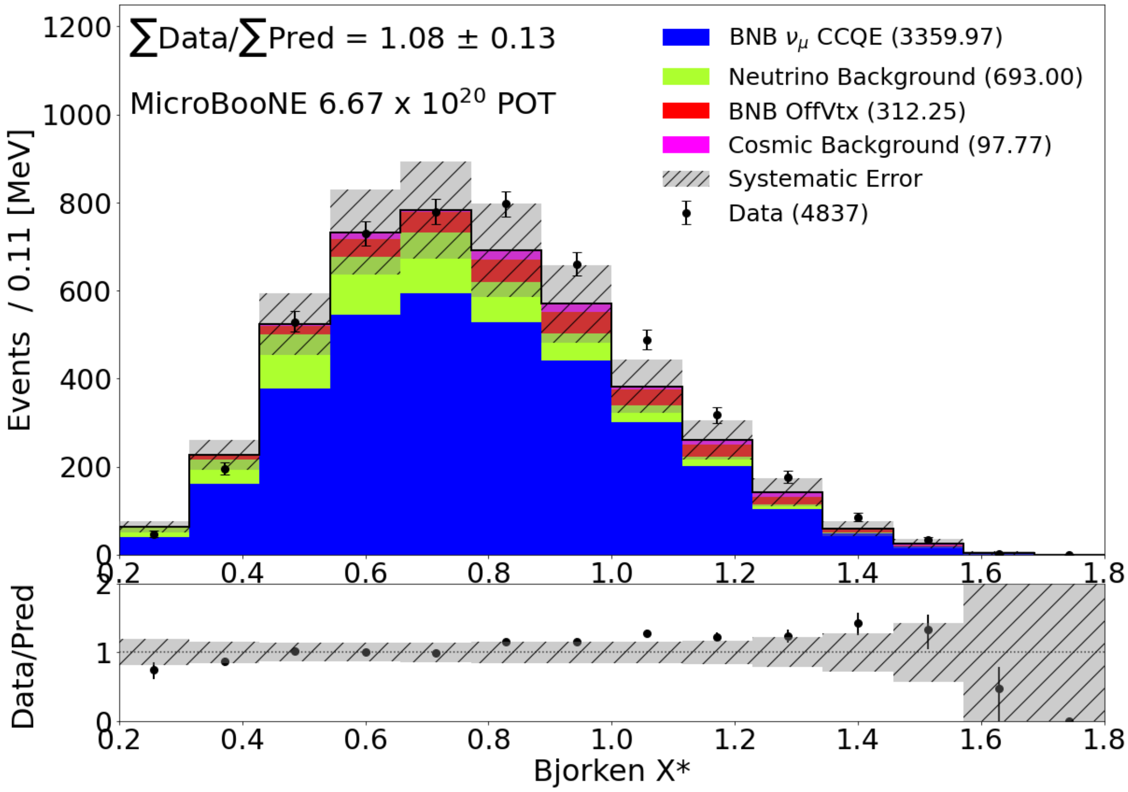


Figure B.2: The Boosted Bjorken's x distribution for the $1\mu1p$ selection.

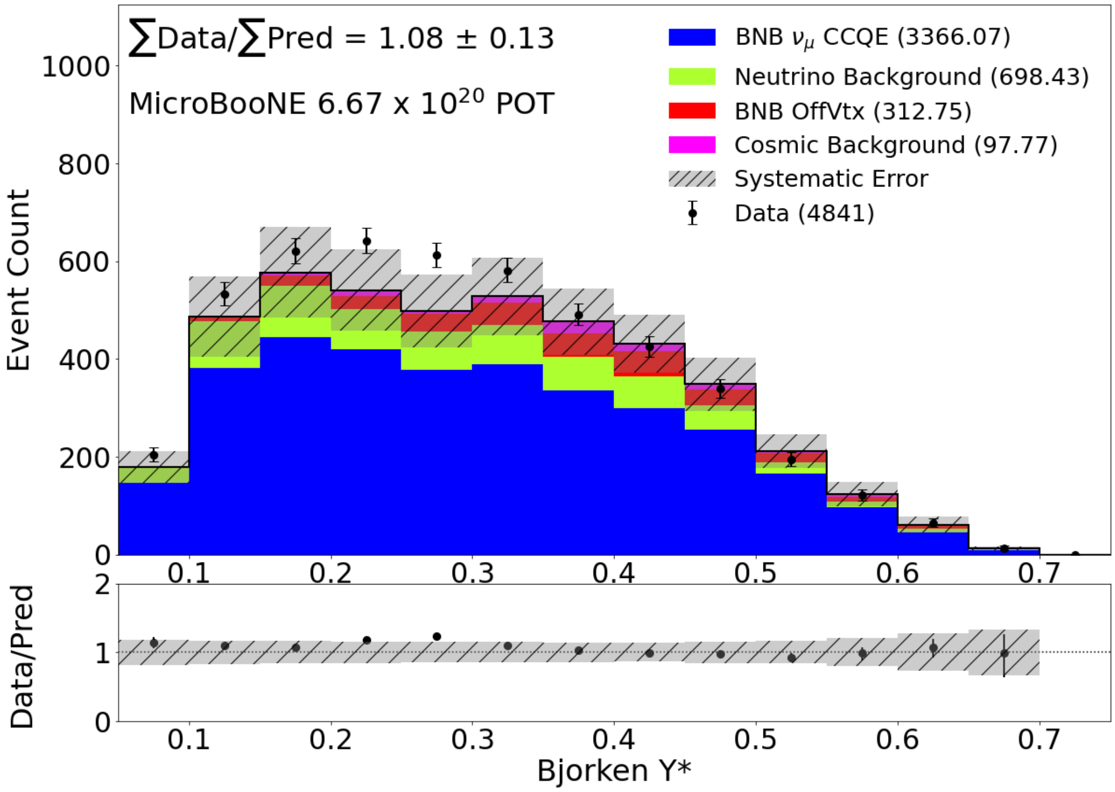


Figure B.3: The Boosted Bjorken's y distribution for the $1\mu 1p$ selection.

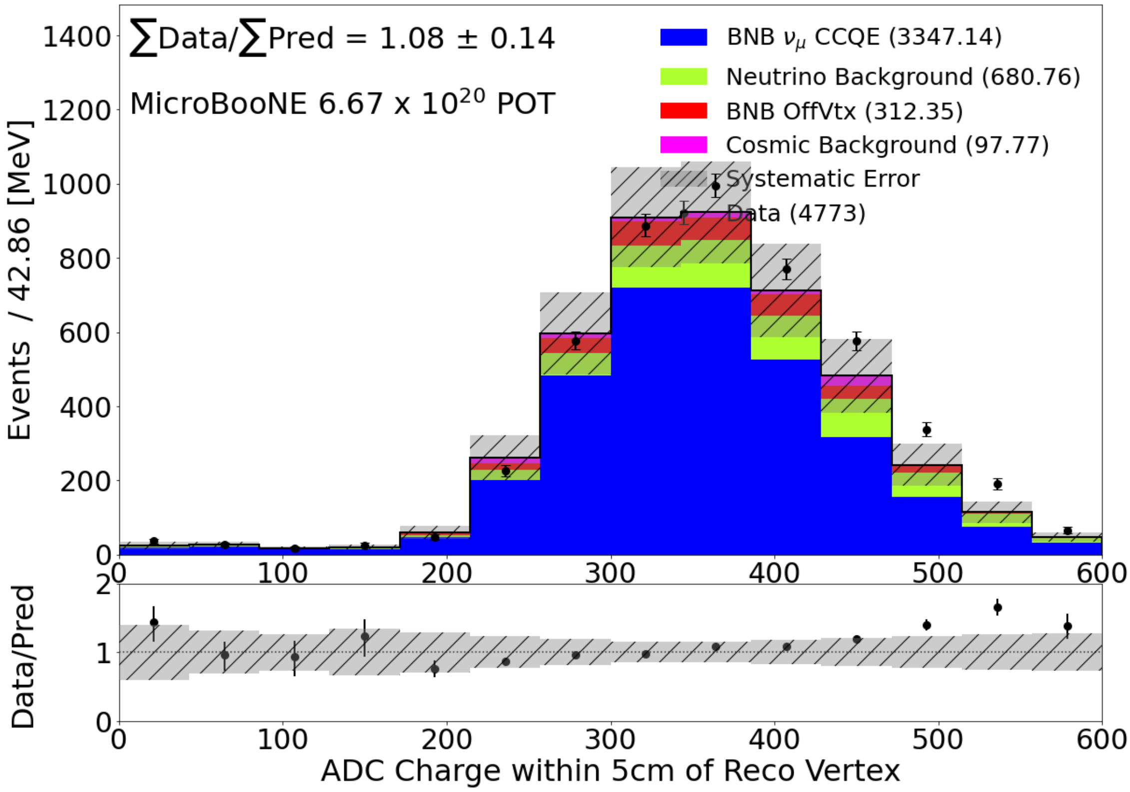


Figure B.4: The distribution for the amount of charge gathered around the interaction vertex for the $1\mu 1p$ selection.

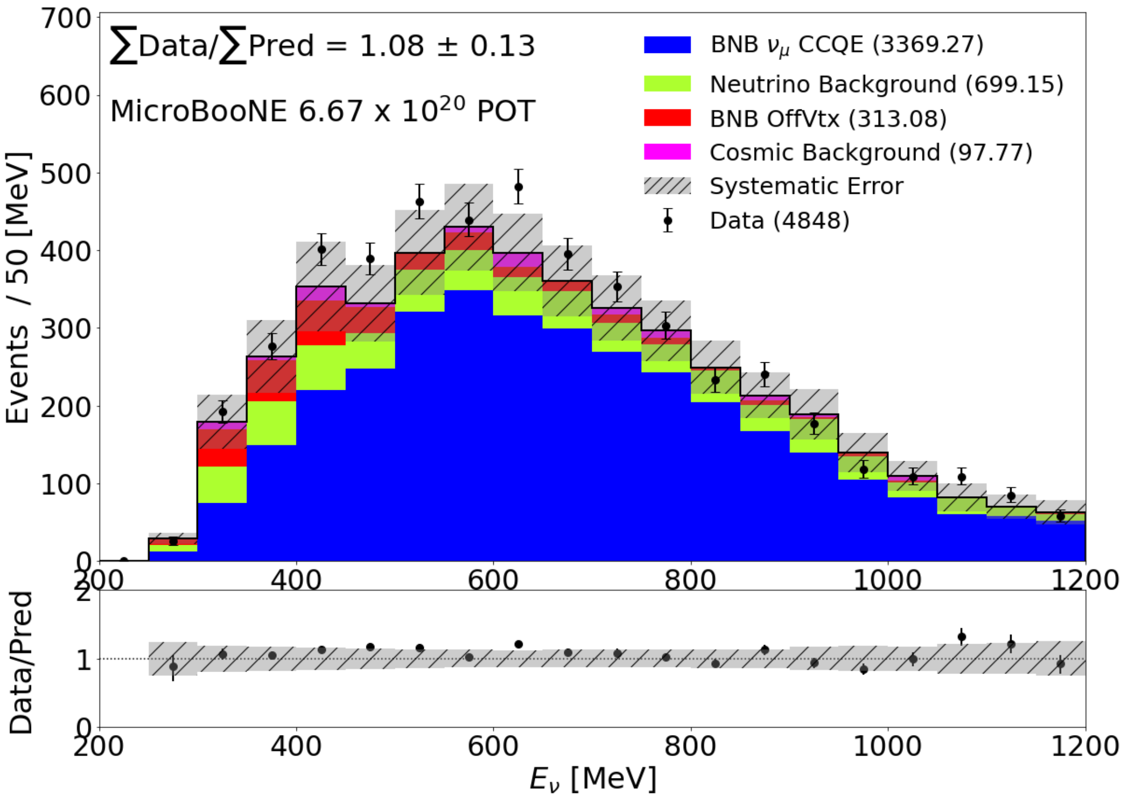


Figure B.5: The range-based reconstructed neutrino energy distribution for the $1\mu 1p$ selection.

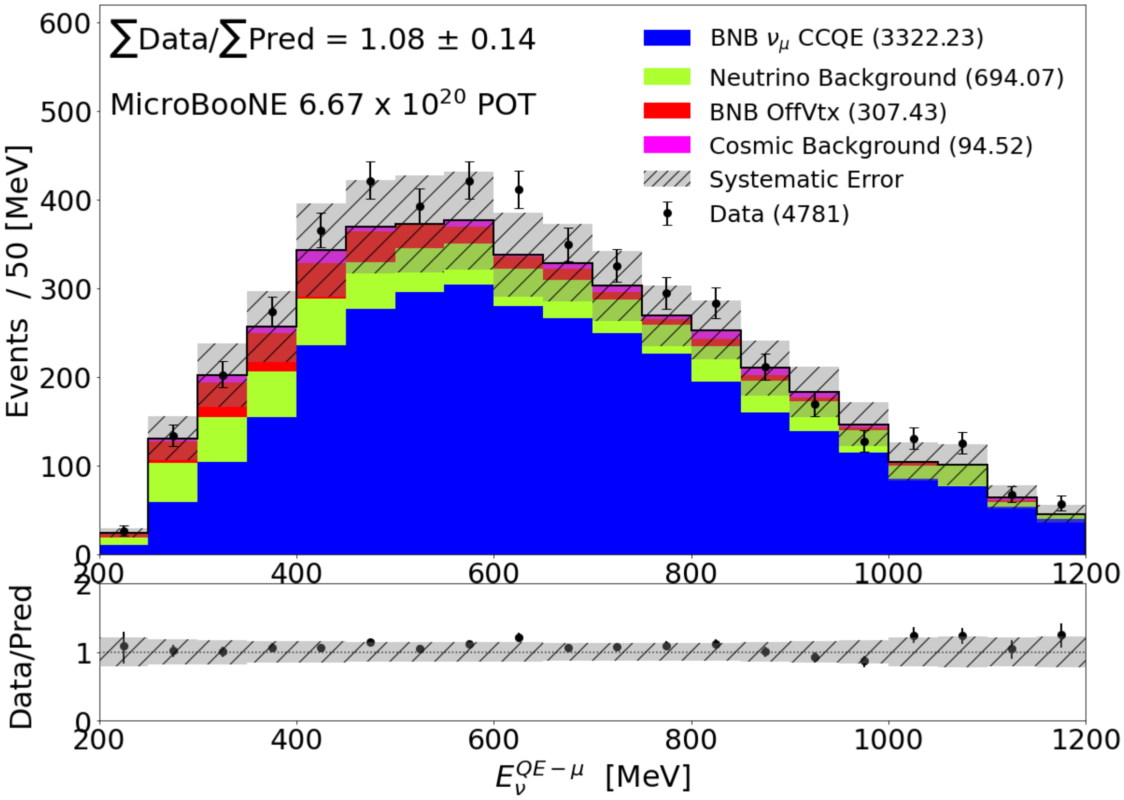


Figure B.6: Via the $E_\nu^{QE-\ell}$ assumption, the reconstructed neutrino energy distribution for the $1\mu 1p$ selection.

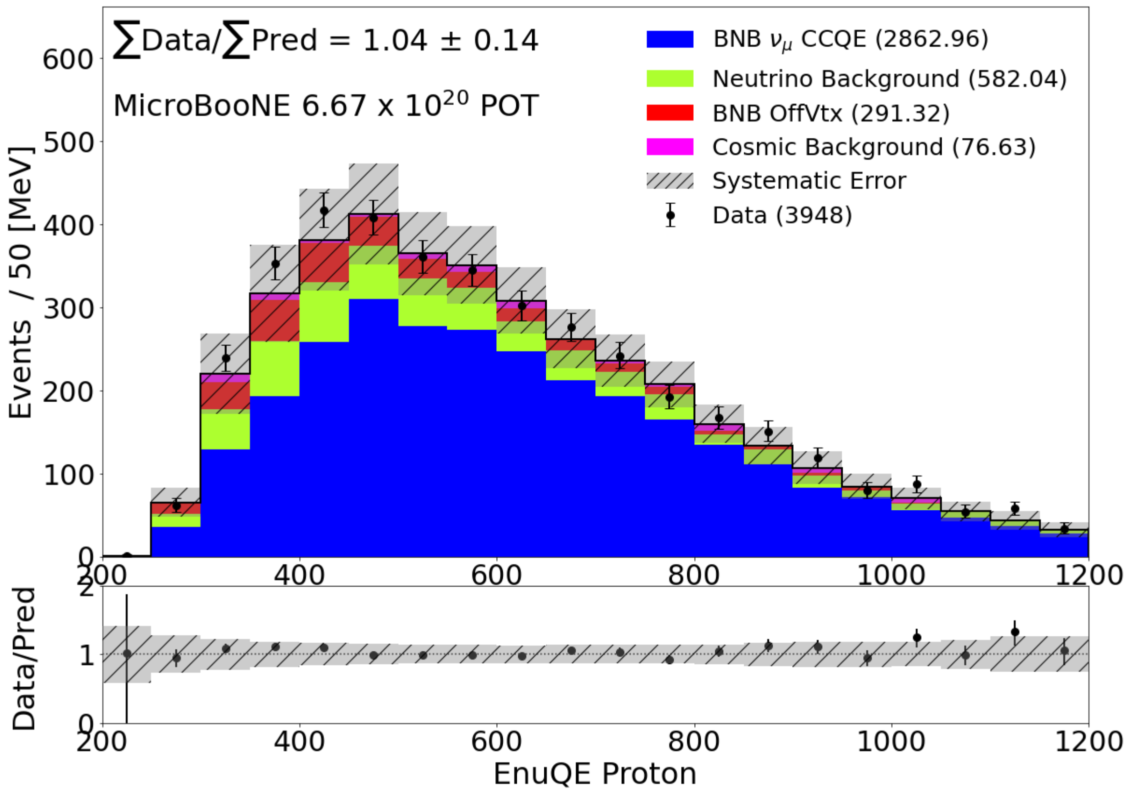


Figure B.7: Via the E_ν^{QE-p} assumption, the reconstructed neutrino energy distribution for the $1\mu1p$ selection.

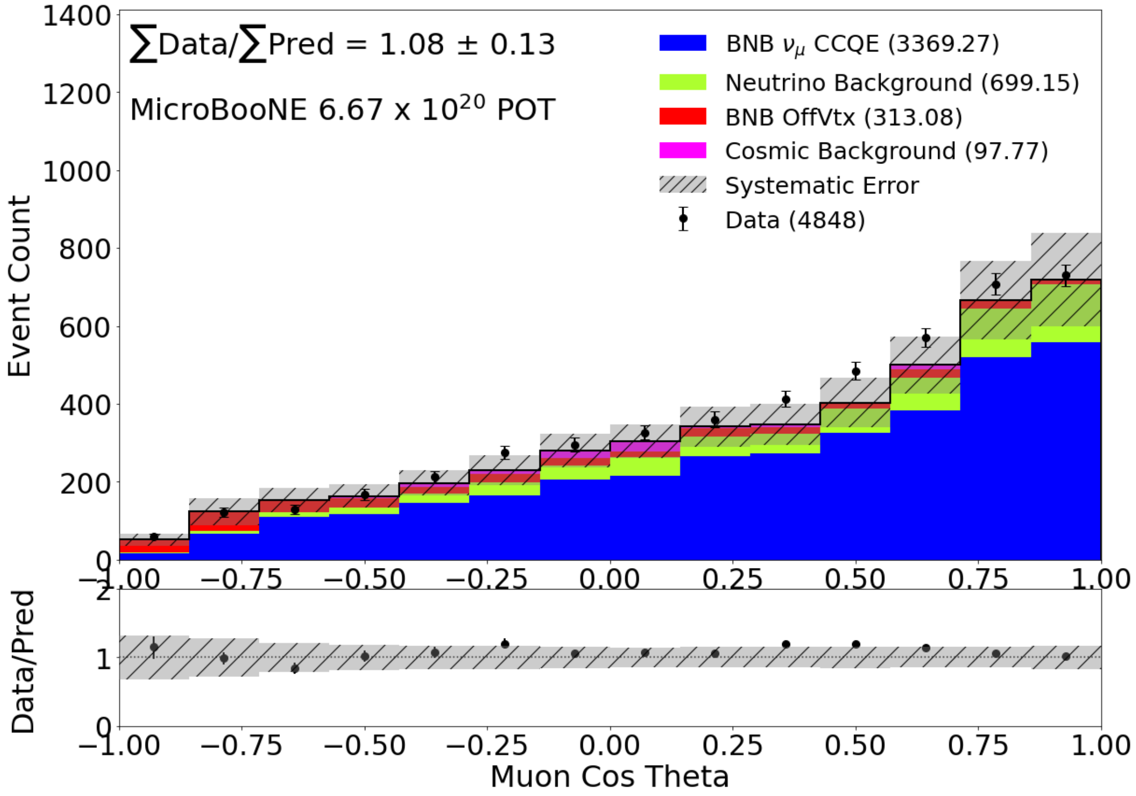


Figure B.8: The $Cos(\theta)$ distribution for the muon in the $1\mu1p$ selection.

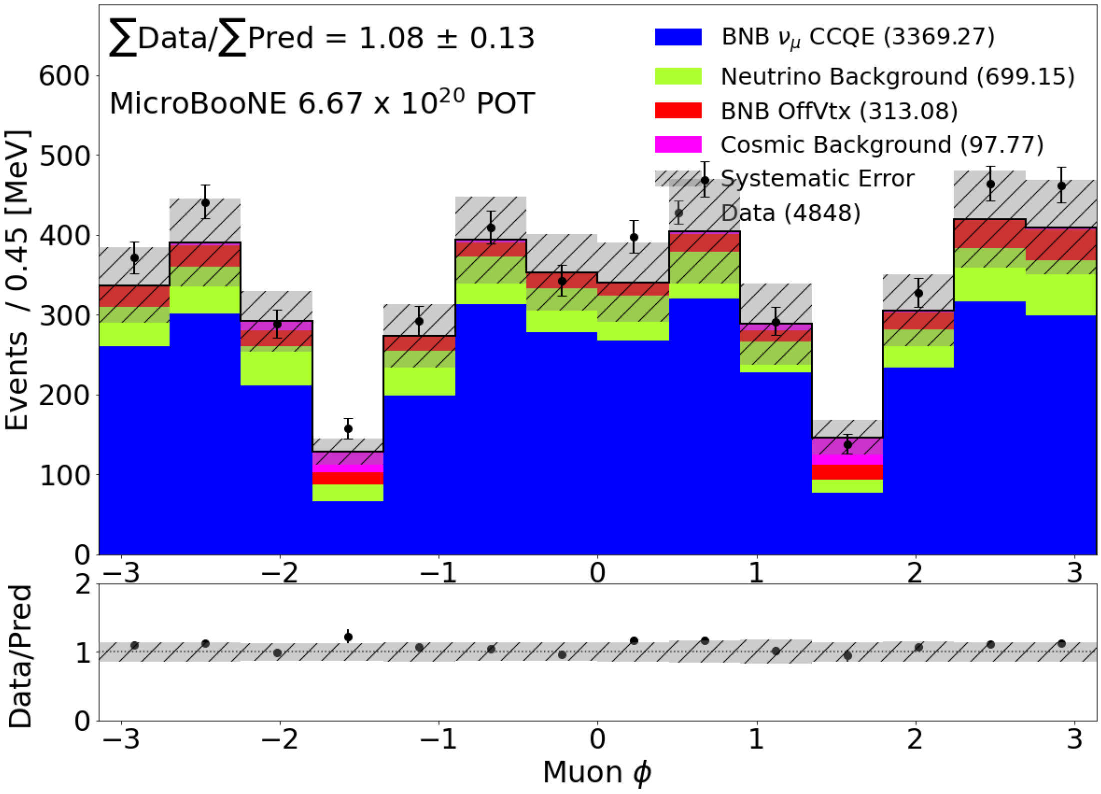


Figure B.9: The ϕ distribution for the muon in the $1\mu1p$ selection.

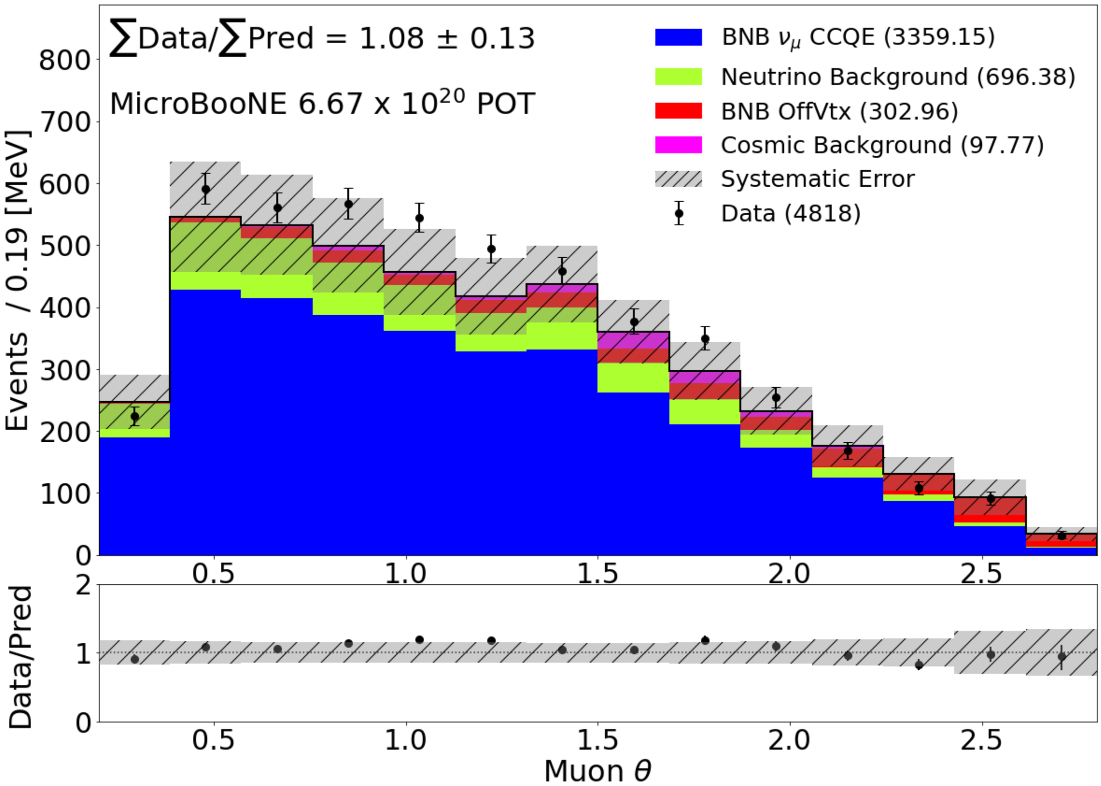


Figure B.10: The θ distribution for the muon in the $1\mu1p$ selection.

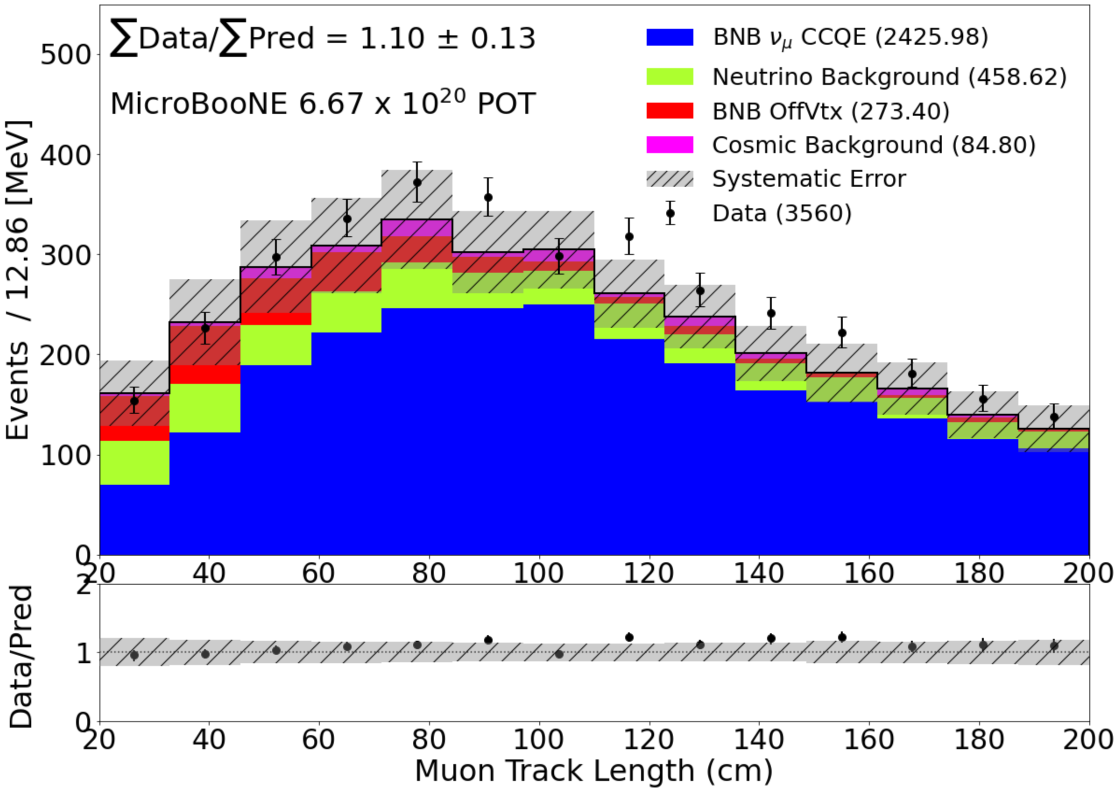


Figure B.11: The muon track length distribution for the $1\mu 1p$ selection.

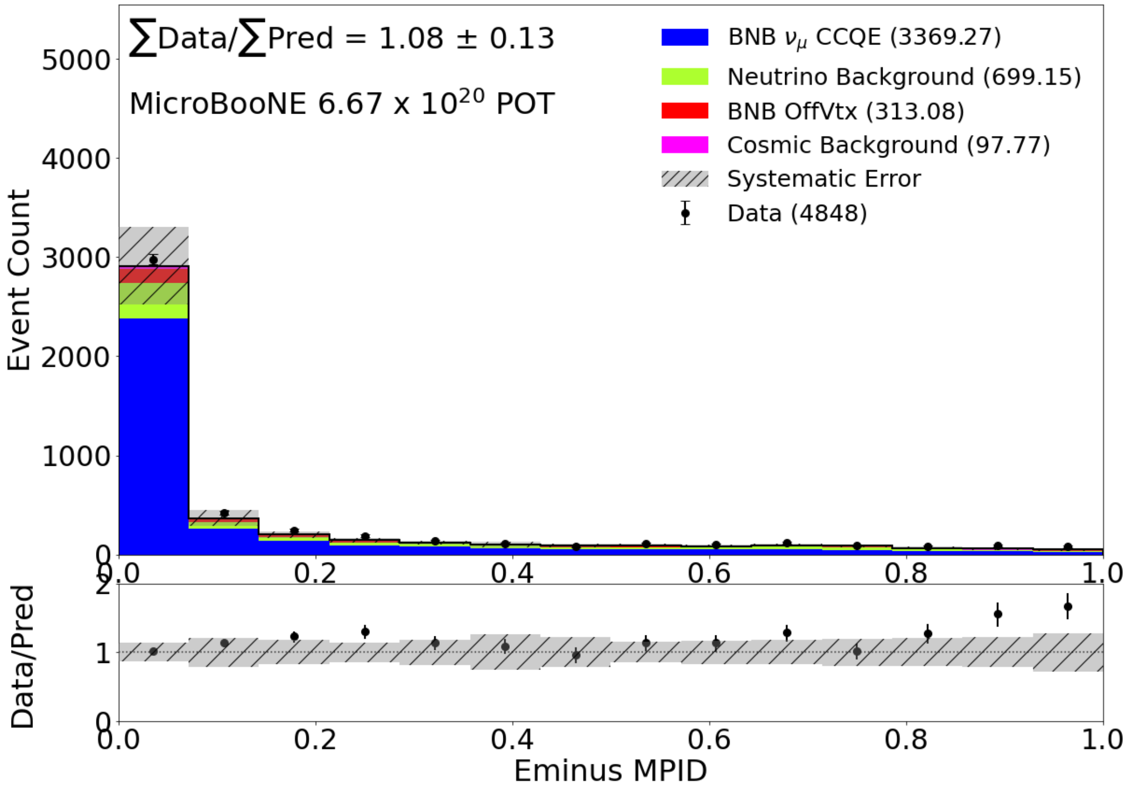


Figure B.12: The MPID score distribution for an interaction containing an electron in the $1\mu 1p$ selection.

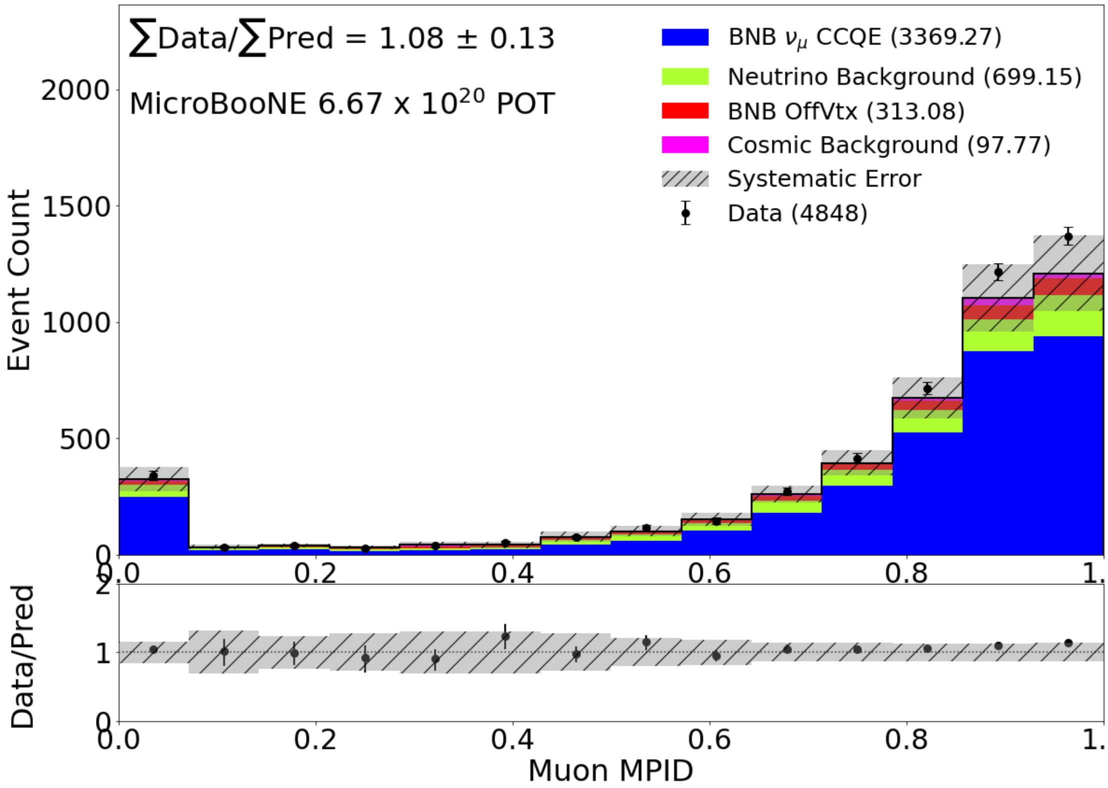


Figure B.13: The MPID score distribution for an interaction containing a muon in the $1\mu1p$ selection.

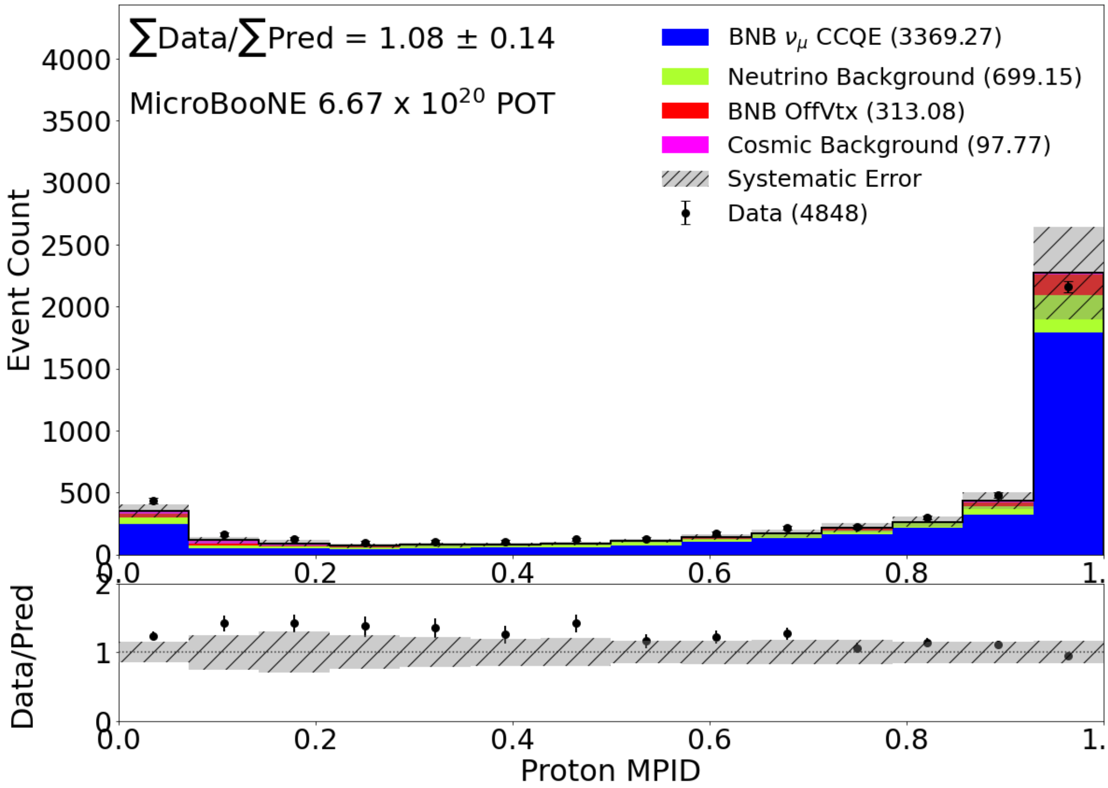


Figure B.14: The MPID score distribution for an interaction containing a proton in the $1\mu1p$ selection.

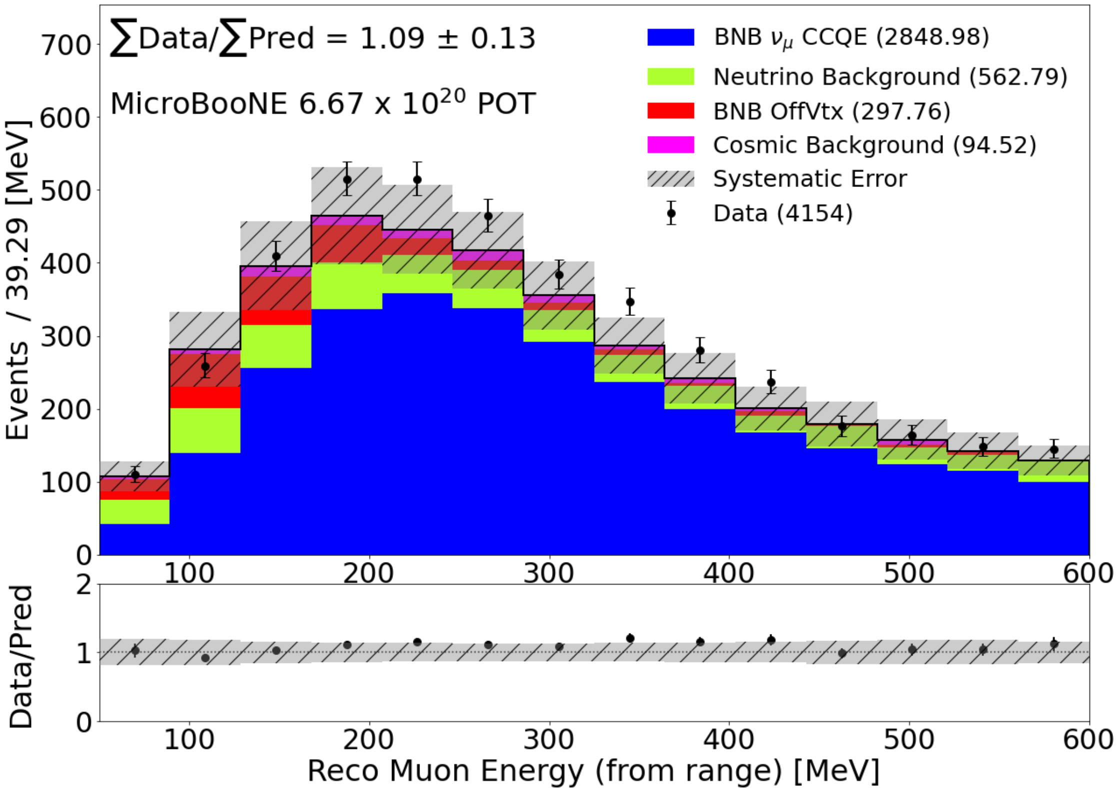


Figure B.15: The muon's reconstructed kinetic energy distribution for the $1\mu1p$ selection.

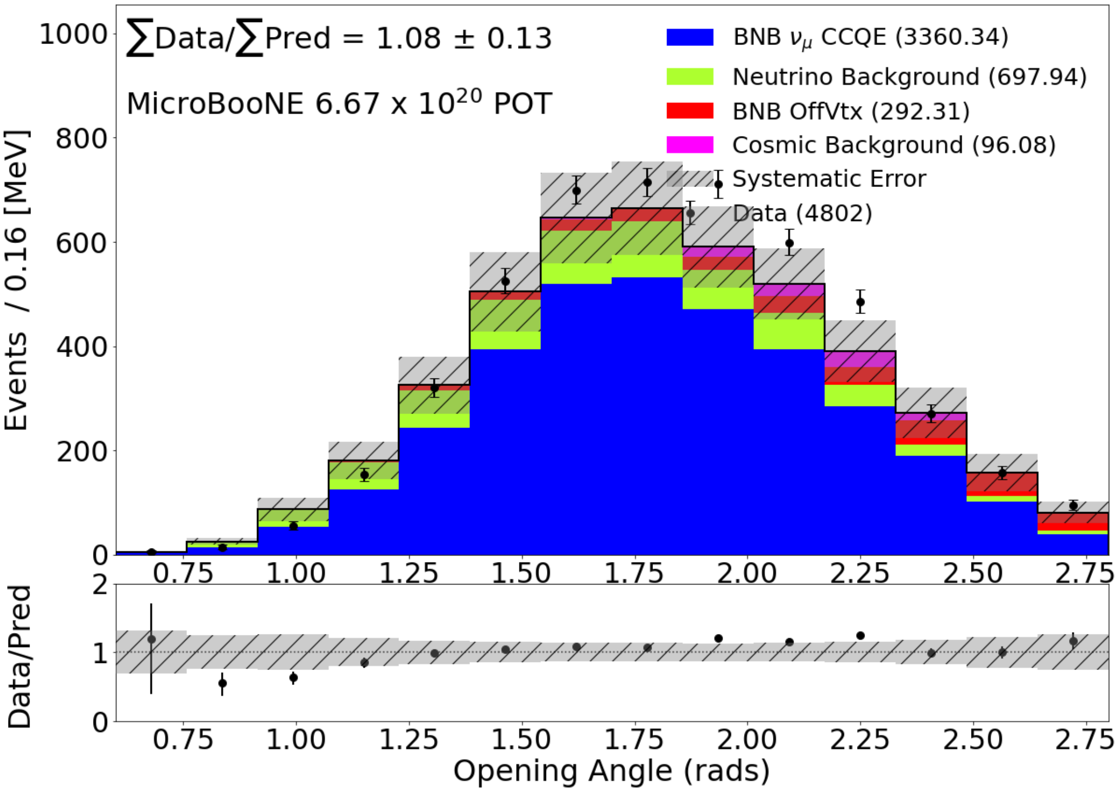


Figure B.16: The interaction opening angle distribution for the $1\mu1p$ selection.

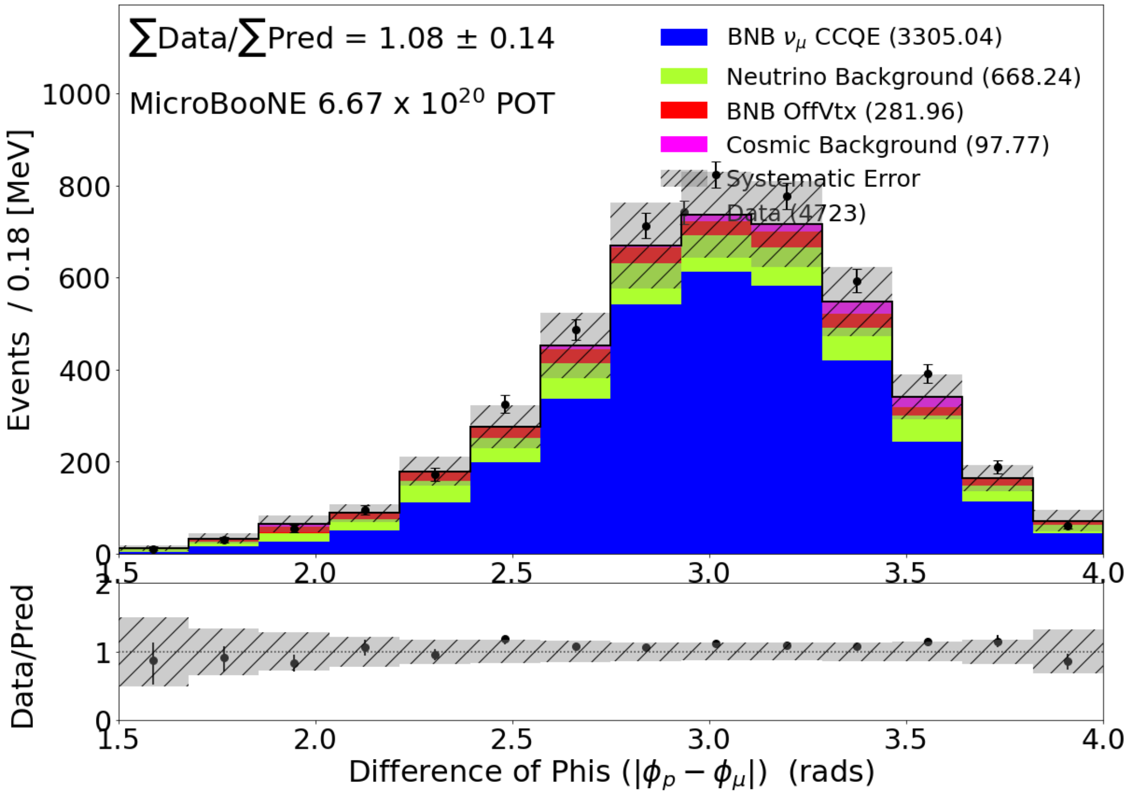


Figure B.17: The distribution for the difference in ϕ variables between the proton and muon for the $1\mu1p$ selection.

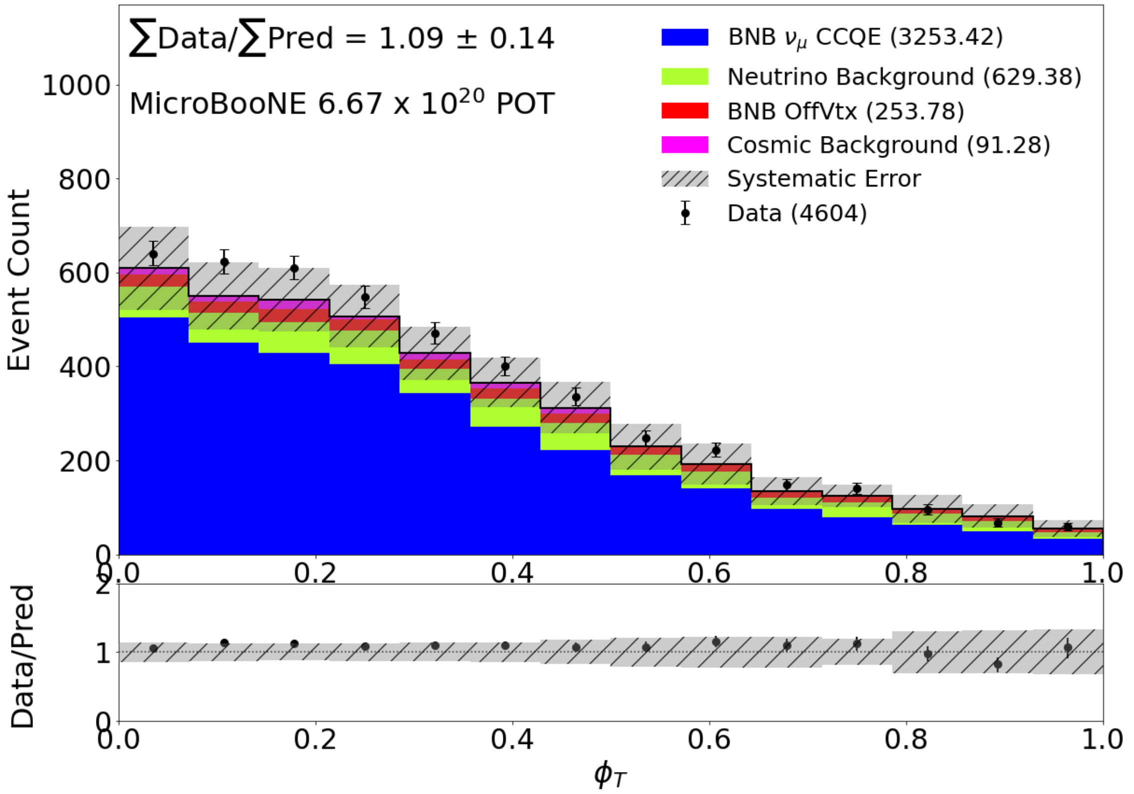


Figure B.18: The transverse ϕ distribution for the $1\mu1p$ selection.

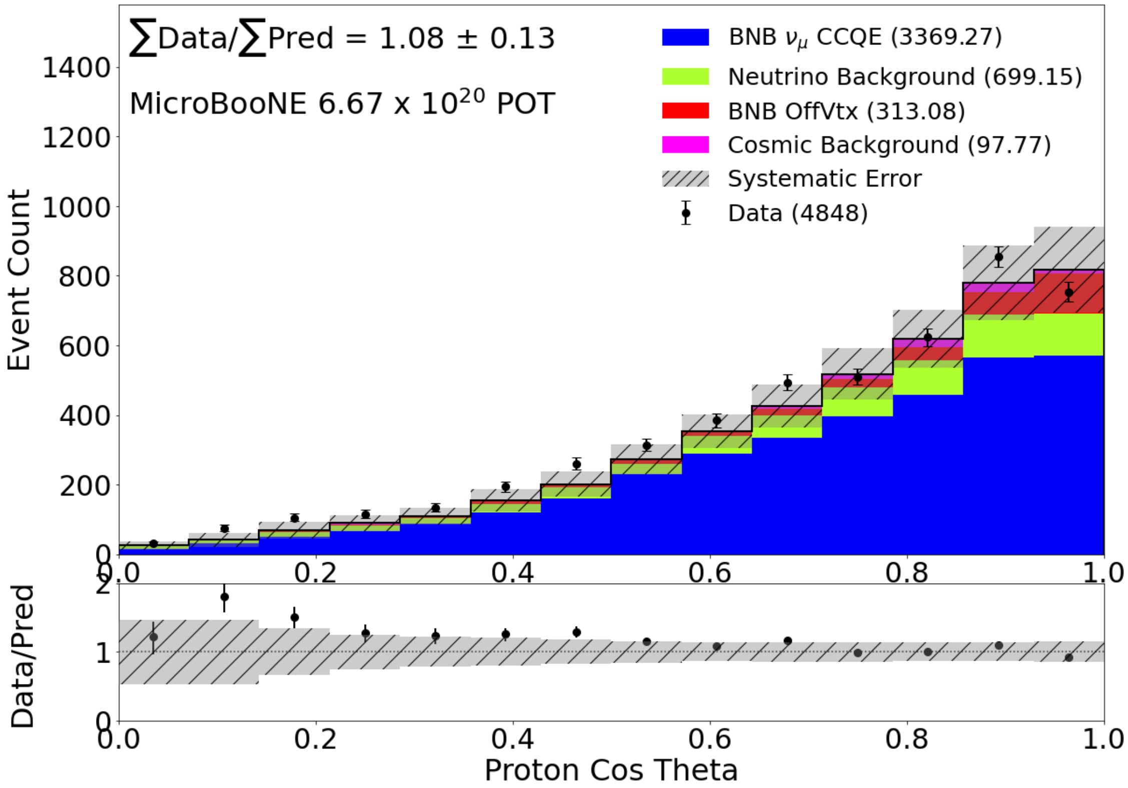


Figure B.19: The $Cos(\theta)$ distribution for the proton in the $1\mu1p$ selection.

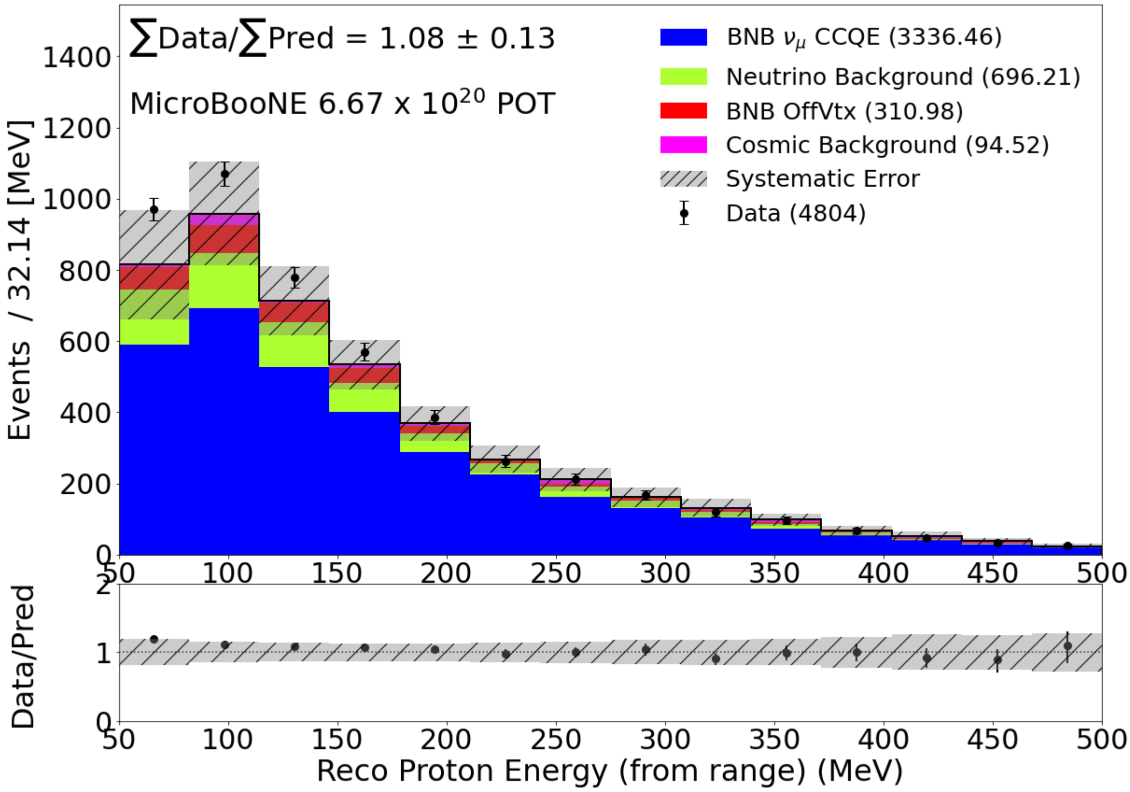


Figure B.20: The proton's reconstructed kinetic energy distribution in the $1\mu1p$ selection.

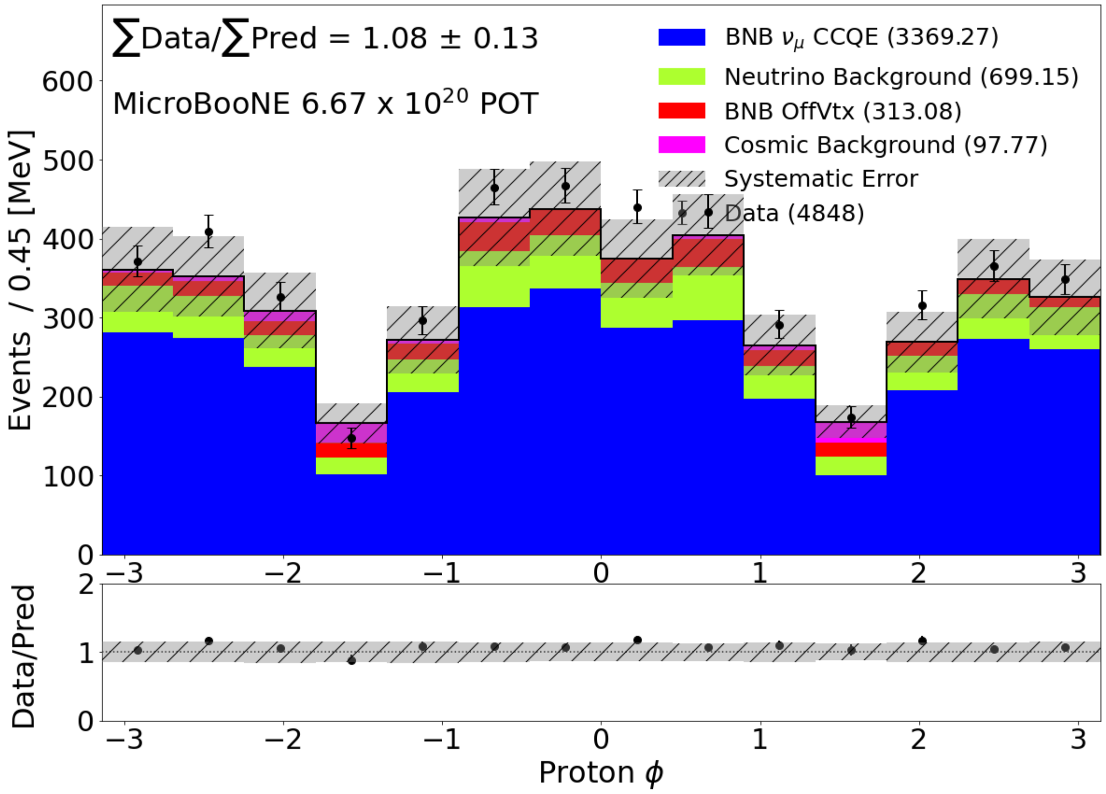


Figure B.21: The ϕ distribution for the proton in the $1\mu1p$ selection.

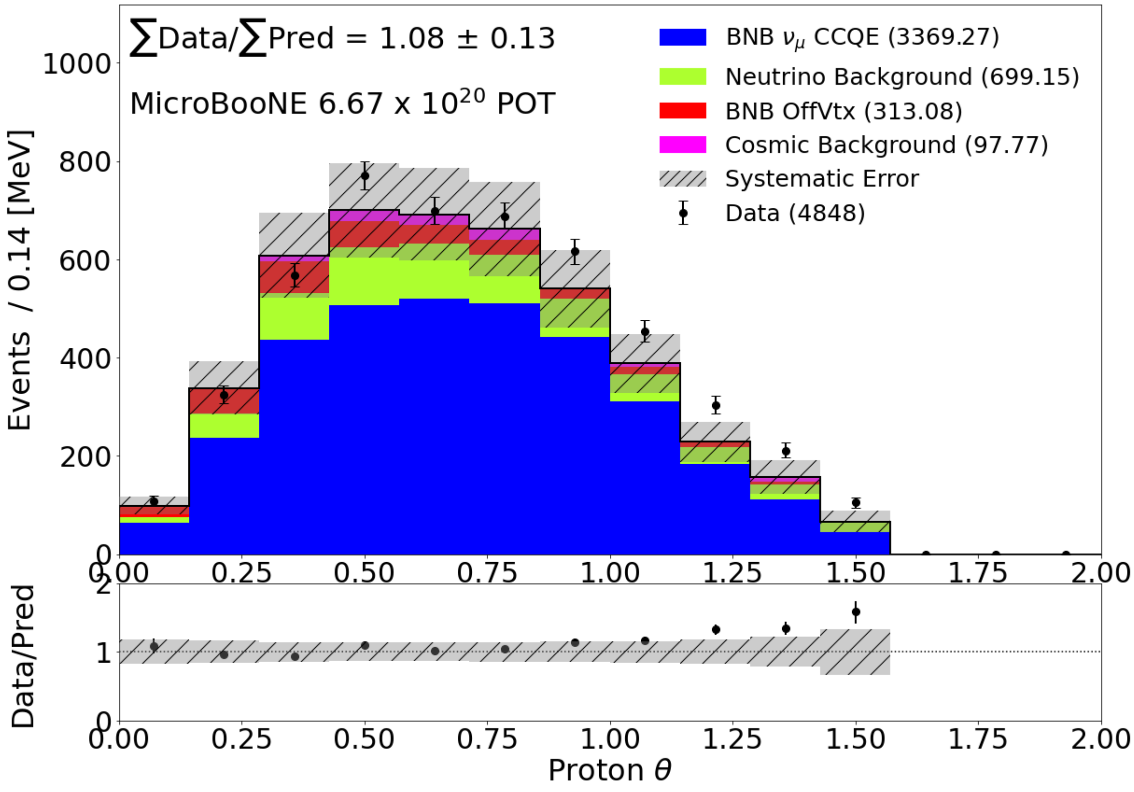


Figure B.22: The θ distribution for the proton in the $1\mu1p$ selection.

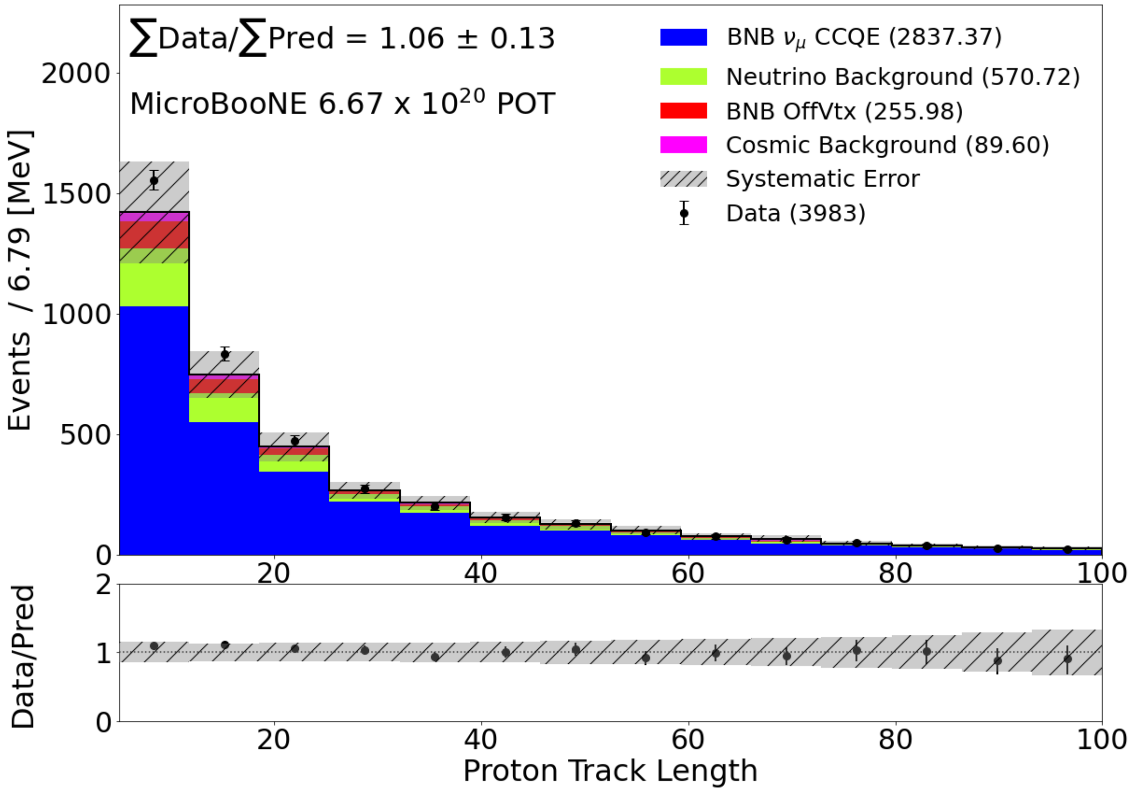


Figure B.23: The proton track length distribution for the $1\mu1p$ selection.

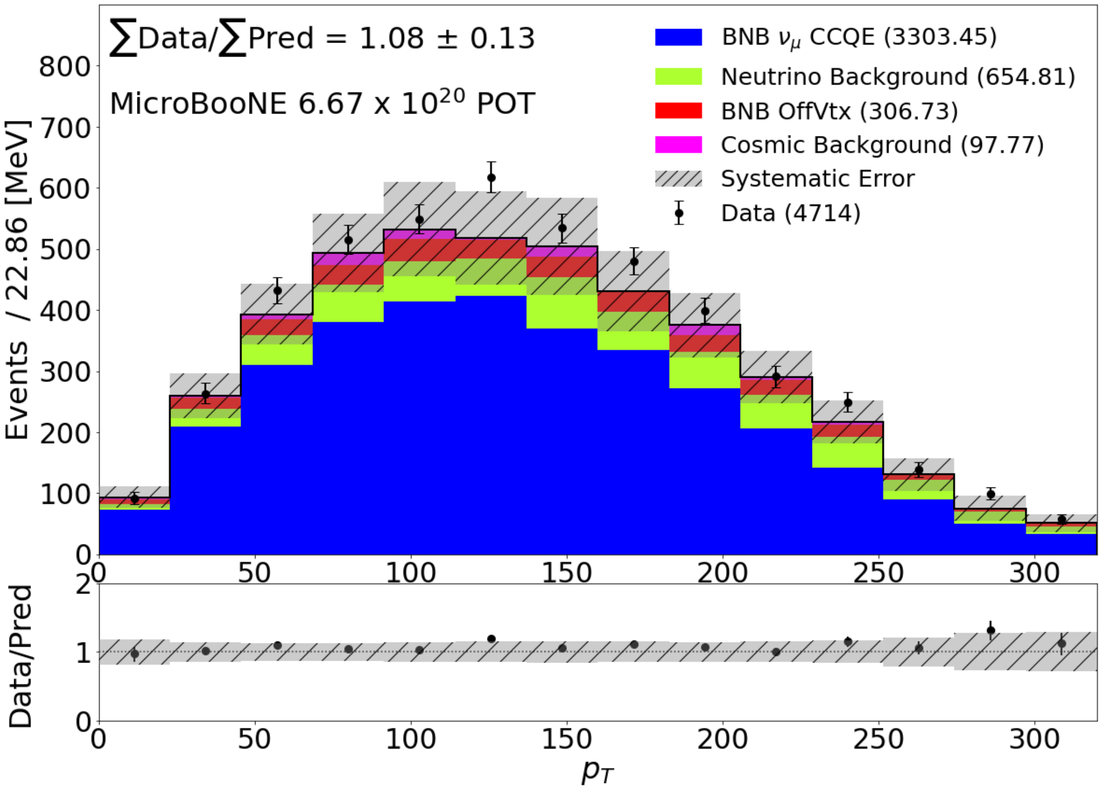


Figure B.24: The transverse momentum distribution for events in the $1\mu1p$ selection.

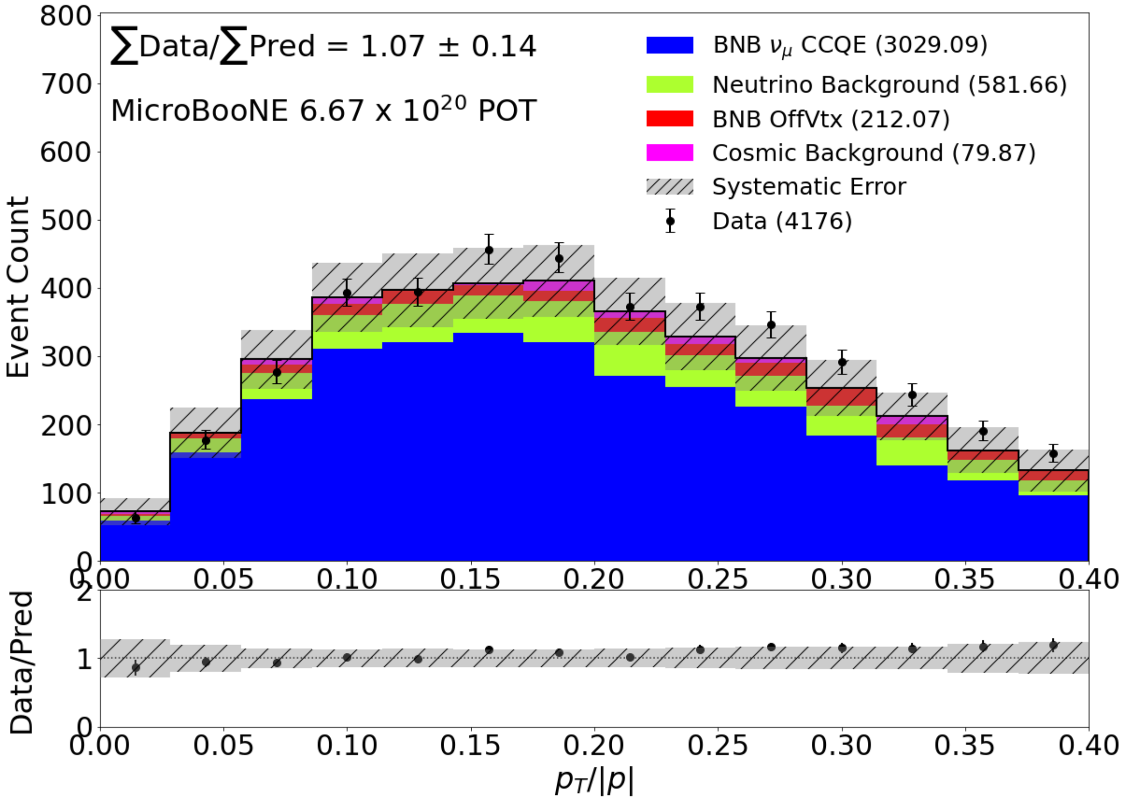


Figure B.25: The distribution of the ratio of transverse momentum to total momentum in the $1\mu1p$ selection.

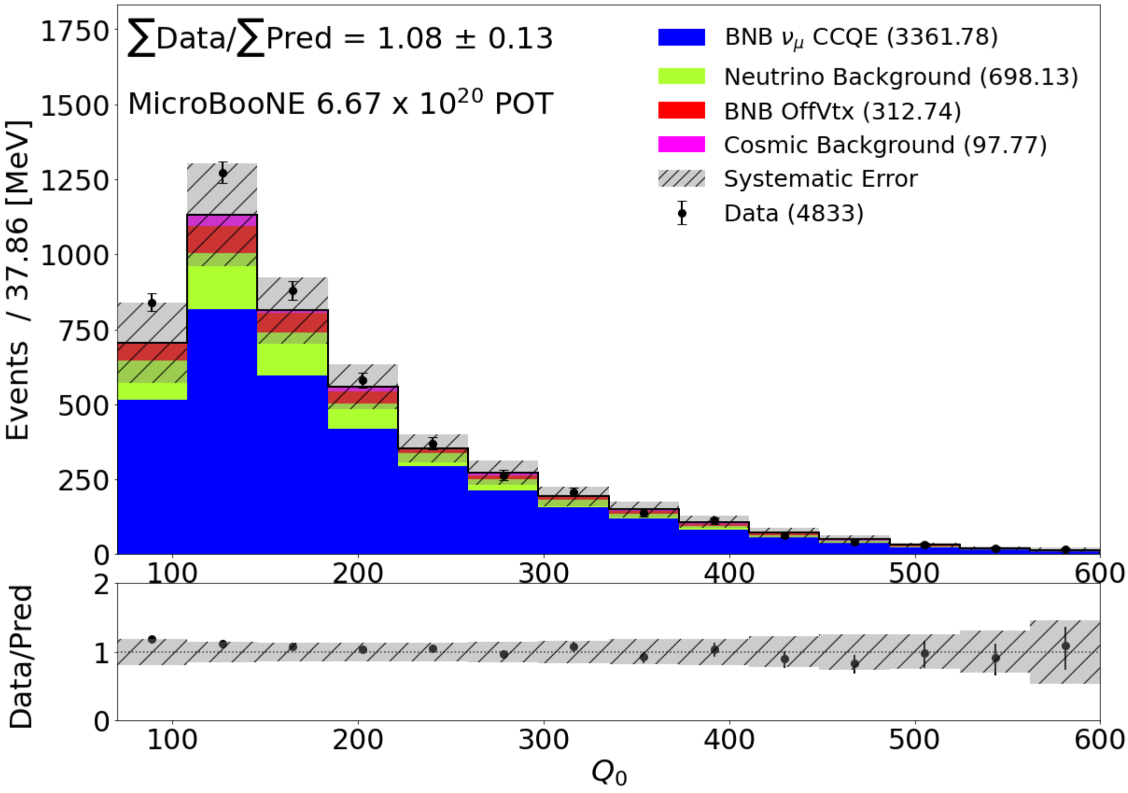


Figure B.26: The Q_0 distribution in the $1\mu1p$ selection.

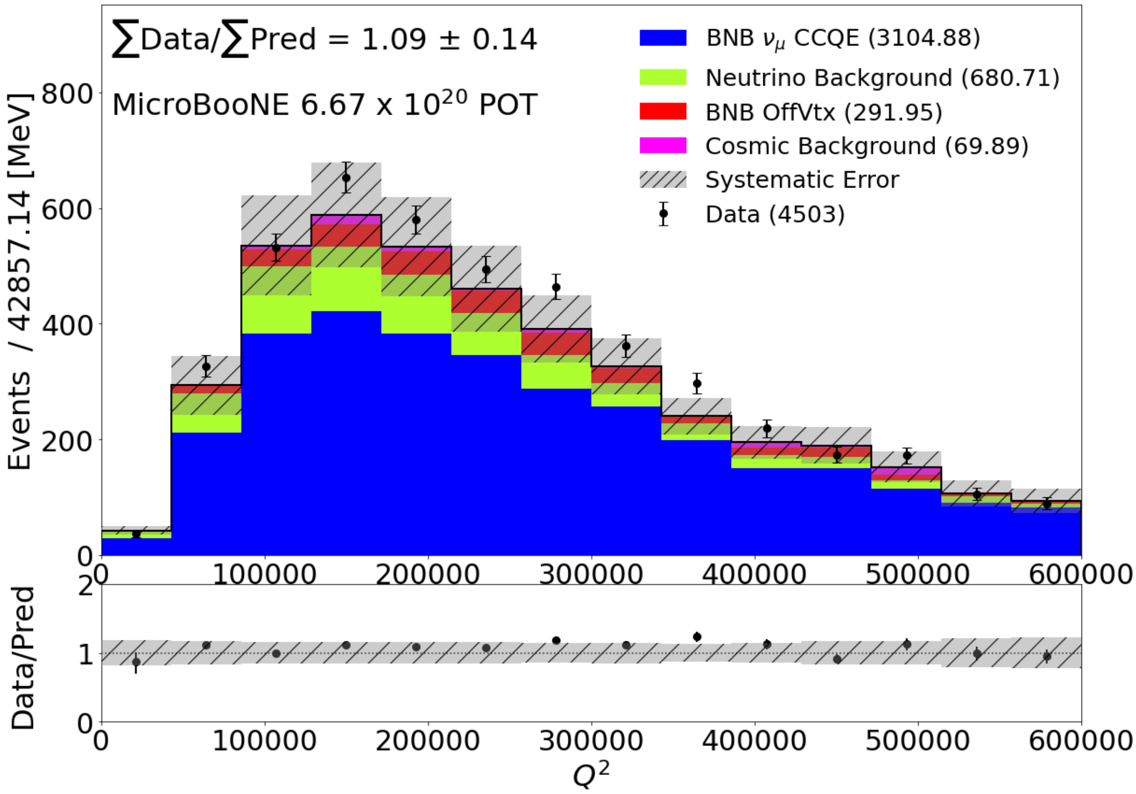


Figure B.27: The momentum transfer squared, Q^2, distribution in the $1\mu1p$ selection.

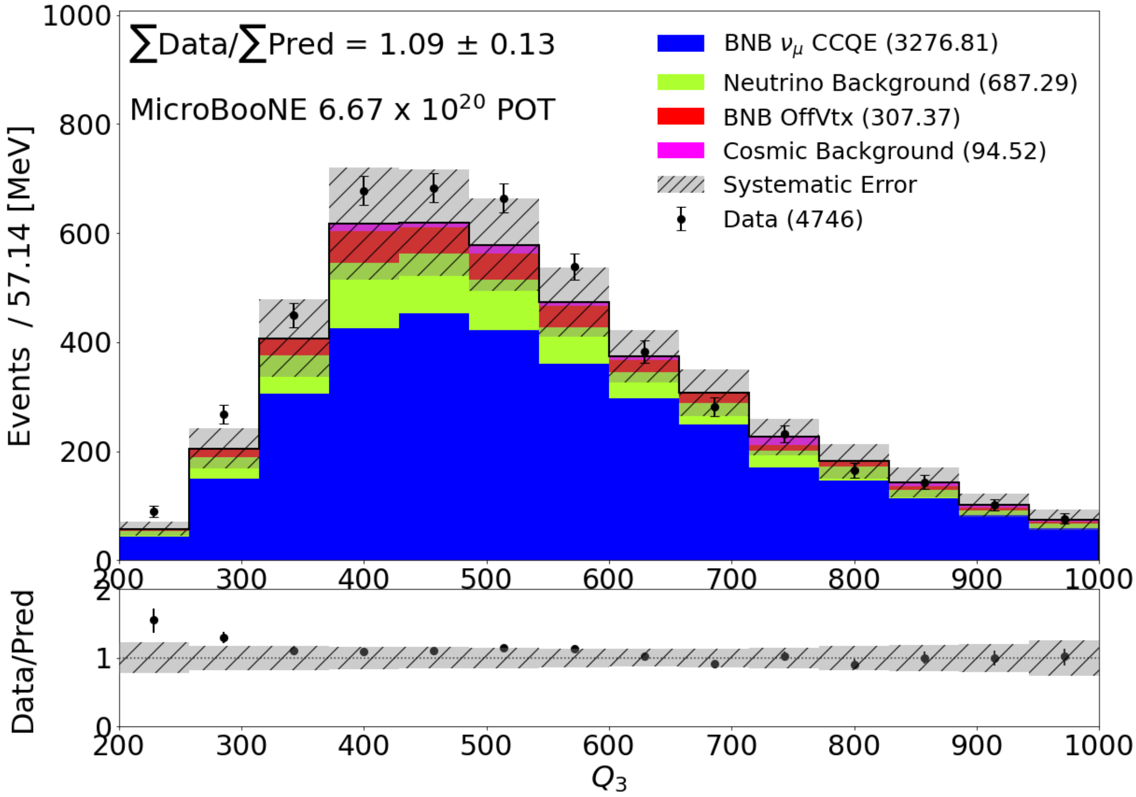


Figure B.28: The z-component of the momentum transfer squared, Q_3, distribution in the $1\mu1p$ selection.

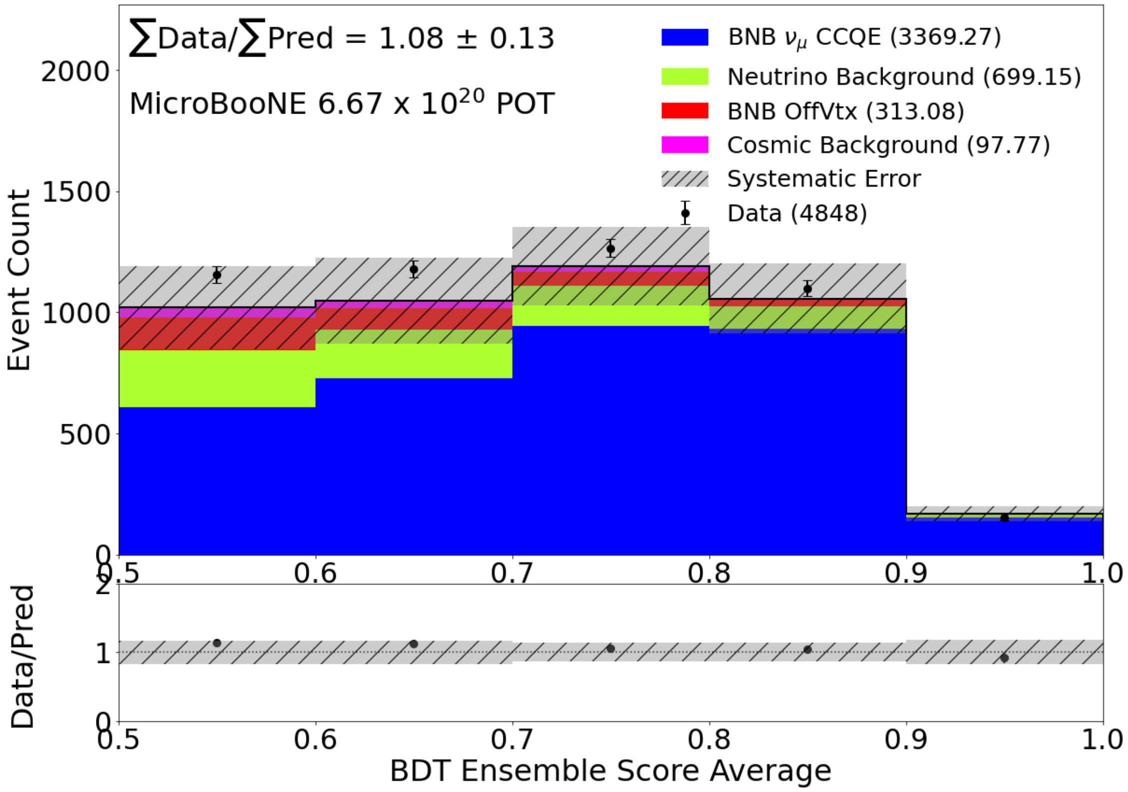


Figure B.29: The BDT Ensemble score distribution in the $1\mu1p$ selection.

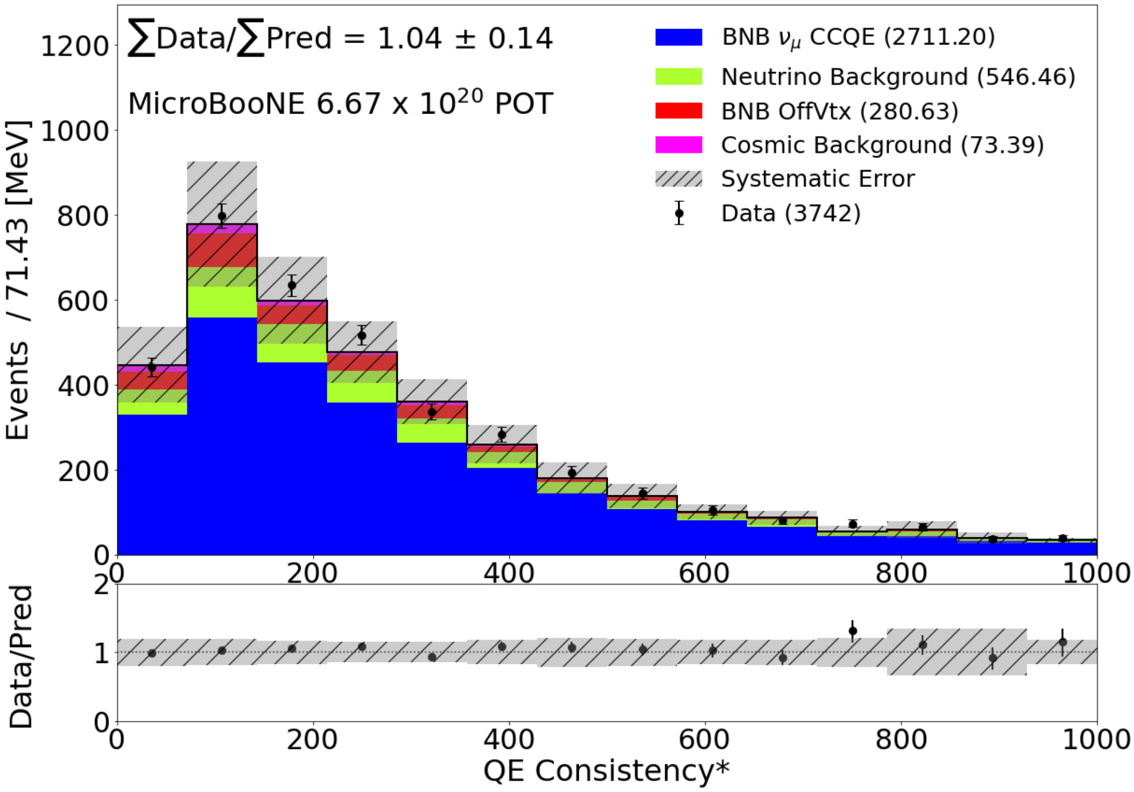


Figure B.30: The QE consistency distribution in the $1\mu1p$ selection.

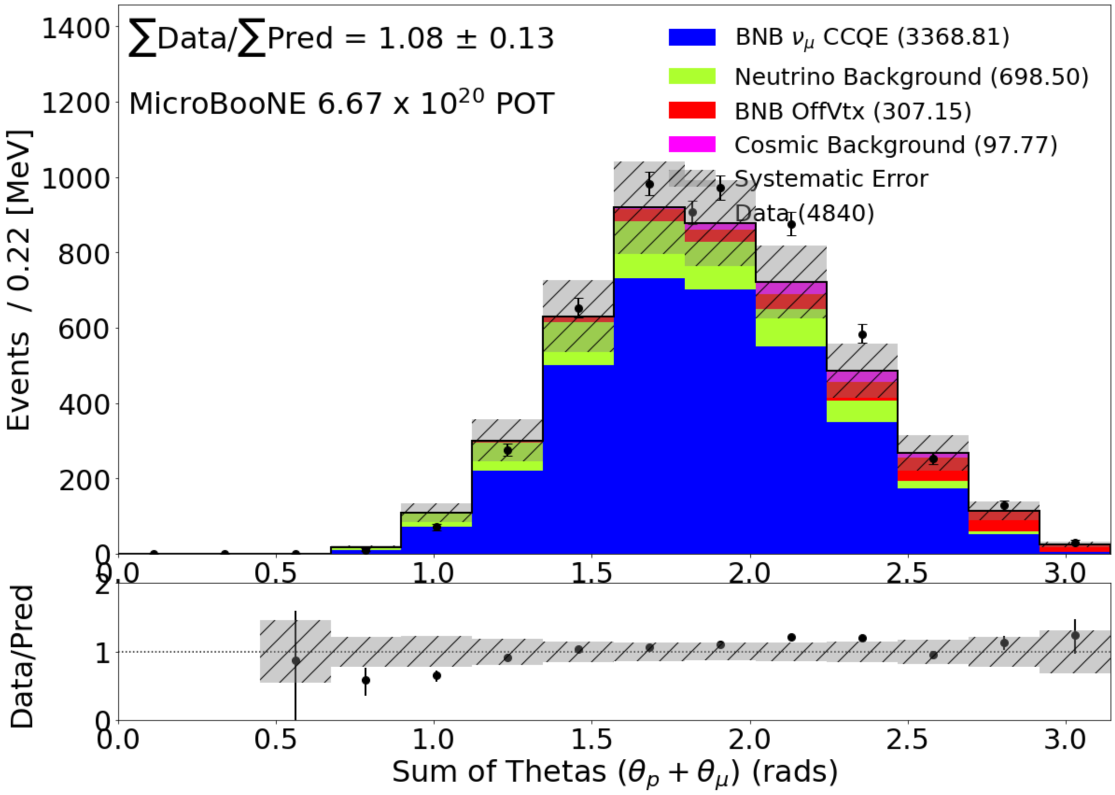


Figure B.31: The distribution of sum of θ_p and θ_m in the $1\mu 1p$ selection.

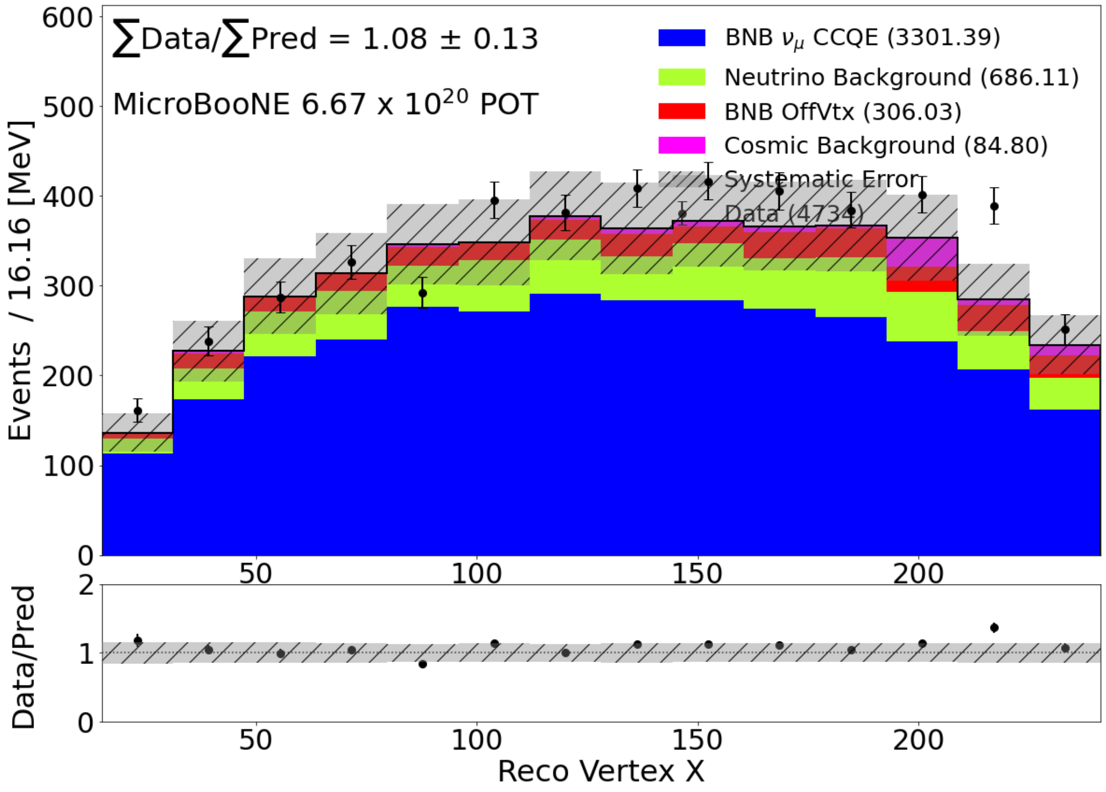


Figure B.32: The distribution of reconstructed x position of the interaction vertex in the $1\mu 1p$ selection.

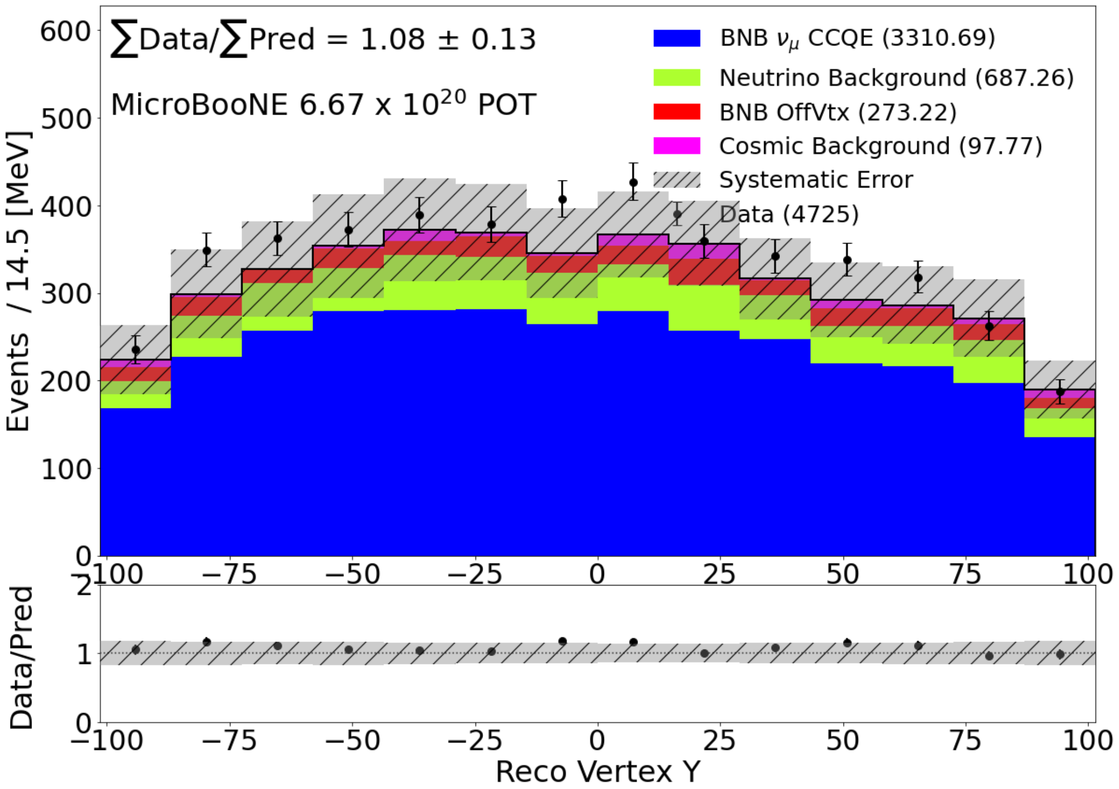


Figure B.33: The distribution of reconstructed y position of the interaction vertex in the $1\mu1p$ selection.

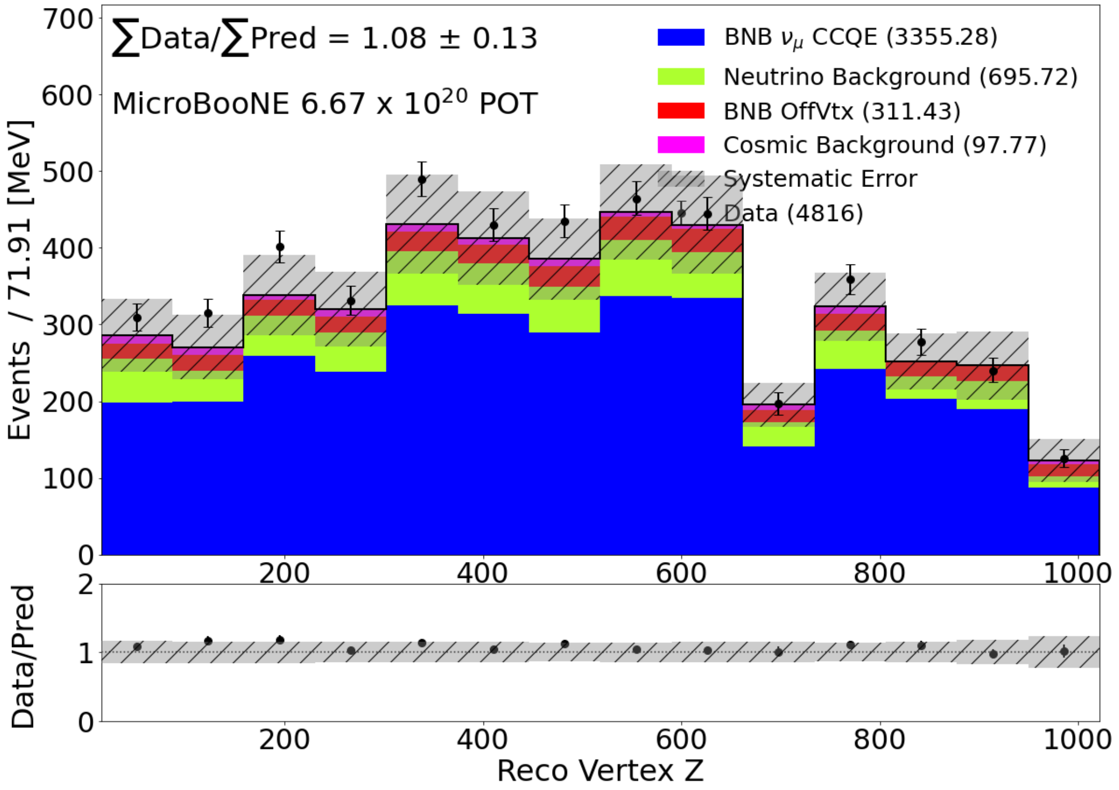


Figure B.34: The distribution of reconstructed z position of the interaction vertex in the $1\mu1p$ selection.

Appendix C

List of Abbreviations

$1e1p$ — One electron one proton, sometimes written as 1e1p. An interaction mode where an electron neutrino creates a single electron and a single proton with nothing else.

$1l1p$ — One lepton one proton, sometimes written as 1l1p. An interaction mode where a neutrino creates a single lepton (muon or electron) and a single proton with nothing else.

$1\mu1p$ — One muon one proton, sometimes written as 1m1p. An interaction mode where a muon neutrino creates a single muon and a single proton with nothing else.

ADC — Analog to Digital Converter which digitizes the raw waveform signals from the wires. Also can refer to the units of the subsequent digital signals.

BDT — Boosted Decision Tree, a machine learning tool used in the DL selections to discriminate signal from background. They consist of several decision trees built by fitting to the data via gradient boosting.

BNB — Booster Neutrino Beamline, the beam delivering mostly muon neutrinos to MiniBooNE and MicroBooNE.

CCQE — Charged-Current Quasi-Elastic, the type of interaction targeted by the DL selections. Charged-current interactions indicate a weak-interaction involving the $W^{\pm}$. Quasi-elastic refers to a nearly-elastic particle scatter where the energy transfer is small relative to that of the incoming neutrino.

CNN — Convolutional Neural Network, a Deep Learning tool where image processing kernels are the artificial neurons in the network. Used in the Deep Learning reconstruction framework, and in the sMask-RCNN cosmic tagging project.

DL — Deep Learning, a subset of machine learning focused around creating artificial neural networks. In the context of MicroBooNE, there exists a DL-analysis framework, which this thesis uses.

EXT — External off-beam data. Data taken using the MicroBooNE detector when the BNB is off, therefore no neutrino interactions are present. Used to analyze the cosmic ray muon background.

LArTPC — Liquid Argon Time Projection Chamber, the detector type for the MicroBooNE collaboration, as well as future neutrino experiments.

LEE — Low Energy Excess, typically referred to the excess in neutrino events observed in the MiniBooNE and LSND experiments compared to their expectations.

ML — Machine Learning, a field of study whereby the tools are fit to solutions via training or experiencing the data compared to traditional handcrafted algorithms.

MPID — Multi-Particle Identification Network, a network used in the DL reconstruction to identify different particle types occurring in an interaction.

ν — A neutrino. Whether electron, muon, tau or sterile is denoted with subscript: ν_e, ν_μ, ν_τ, ν_s

π — A pion particle. Can be charged or neutral: $\pi^\pm$, π^0

PMT — Photomultiplier tube

SM — Standard Model of Particle Physics, where neutrinos are massless. In the commonly accepted extension to the standard model, neutrinos have mass and oscillations become possible.

sMask-RCNN — Sparse Mask-RCNN, or Sparse Mask-Region Proposal Convolutional Neural Network. A DL network using sparse convolutions to locate, classify, and cluster different interaction types.

SSNET — Semantic Segmentation Network. A network used in the DL reconstruction framework to label pixels as different categories of track or shower signals.