

COPENHAGEN UNIVERSITY

MASTER THESIS

Observing The Universe Using The Antarctic Ice Sheet

Classifying And Reconstructing Neutrinos In Real Data From
The IceCube Neutrino Observatory Using Graph Neural
Networks

Author:
Peter Andresen

Supervisor:
Troels Christian Petersen

*A thesis submitted in fulfillment of the requirements
for the degree of Master in Computational Physics*

at the

High Energy Particle Physics Department
Niels Bohr Institute

May 20, 2023

Declaration of Authorship

I, Peter Andresen, declare that this thesis titled, "Observing The Universe Using The Antarctic Ice Sheet" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made so clear.

Signed: *Peter Andresen*

Date: 18/05 2022

"I have done a terrible thing, I have postulated a particle that cannot be detected. "

Supposedly Wolfgang Pauli

"Well.. Lets see about that. "

Peter Andresen, 2023

COPENHAGEN UNIVERSITY

Abstract

High Energy Particle Physics

Niels Bohr Institute

Master in Computational Physics

Observing The Universe Using The Antarctic Ice Sheet

by Peter Andresen

The IceCube Neutrino Observatory (IceCube) uses a cubic kilometre of the Antarctic ice sheet to detect neutrinos. The IceCube neutrino oscillation measurements and potential contributions to multimessenger astronomy heavily depends on accurate and fast classification and reconstruction of neutrinos in real data. GraphNeT, a Graph Neural Network (GNN) open-source Python framework founded at the Niels Bohr Institute, has been shown to improve both classification and reconstruction of low-energy simulated neutrinos, while speeding the process up by orders of magnitude. This work shows that the method works in actual data as well. The first large neutrino sample is classified using a GNN from "raw" lvl 2 data and compared to a Monte Carlo neutrino selection across a range of reconstructed and calculated variables. The comparison indicates that the sample has a high neutrino purity. The amount of neutrinos in the clean GNN selection is compared to that of existing methods in the IceCube Oscillation work group using simulated data, suggesting that the GNN method can generate a 50-70% larger sample at a similar purity. Implementing the GNN in IceCube and increasing the neutrino rate would improve existing analyses and potentially open up the possibility of creating early warnings for electromagnetic telescopes using low-energy neutrinos. This work also contributes to a benchmark study trying to improve the GNN performance for high energy track neutrinos. No significant improvements were obtained.

Acknowledgements

First and foremost thanks to my supervisor, Troels Christian Petersen, without whom this thesis would never have been written. Troels comes up with a ton of ideas and suggestions, the vast majority of which are quite brilliant. Simultaneously he is unwavering in his positivity. Troels always has time to help and look at new plots and he rarely fails to derive new insights or point out interesting details. Troels also tells the most incredible tales, some of which would be really difficult to believe, were they not coming from a person as extraordinary as him. From tours in his sailboat to bicycling on Bornholm, Troels truly made writing my thesis an enjoyable experience.

The IceCube experiment and the machine learning methods in this work represented a tremendous learning curve. Morten Holm, whom I worked closely with in the first half year, was invaluable in getting a head start. Morten is a very friendly and analytical person, whom I had the great pleasure becoming friends with. Without Morten, this thesis would be substantially less interesting. I could not imagine a better person to collaborate with.

When the visions of Troels needed translating into code, no greater help could be found than that of the GraphNeT team. Especially Andreas Søgaaard and Rasmus Ørsøe were always ready to help. The IceCube/GraphNeT experience of Rasmus and the coding/debugging skills of Andreas are unparalleled. They are also inspiring and enjoyable to engage with.

Thanks to Tom Stuttard and Tania Kozynets for being tremendously helpful in reviewing our results and pointing out potential fixes. They are experts on all things OsxNext/IceCube, very kind and freely shared their knowledge.

Finally, the process of writing the thesis was much more enjoyable than anticipated. This was first and foremost due to our office atmosphere. I was lucky enough to write my thesis on the same subject as some of my close friends, sitting in the same room. Thanks for all the boardgames, talks and good times, Andreas Mosgaard, Simon Debes, Morten Holm and Moust Holmes. It was a great pleasure sharing the year with you.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
1 Introduction	1
2 Readers Guide	2
2.1 How To Use This Readers Guide?	2
2.2 What Is IceCube?	2
2.3 What Is The Purpose Of IceCube And What Are The Challenges? . . .	3
2.4 What Do Events In IceCube Look Like?	4
2.5 What Does This Work Try To Improve On?	4
2.6 Why Use GraphNeT and GNNs To Do So?	5
2.7 Which Variables Are Important?	5
2.8 How Can The Analyses In This Work Be Replicated?	6
2.8.1 Classification And Reconstruction Of Neutrinos In Real Data . .	6
2.8.2 Northern Track Benchmark	6
3 The Standard Model of Particle Physics	7
3.1 The Standard Model Of Particle Physics	7
3.1.1 Fermions	8
3.1.2 Bosons	8
3.2 Neutrinos	8
3.2.1 Neutrino Prediction And Discovery	9
3.2.2 Neutrino Interaction Types	9
3.2.3 Neutrino Flavor Oscillation	10
3.2.4 How To Spot A Neutrino In the Wild? - Cherenkov Radiation .	12
3.3 Cosmic Rays And Air Showers	13
4 The IceCube Neutrino Observatory	15
4.1 The Purpose Of IceCube	15
4.1.1 Neutrino Oscillation	15
4.1.2 Neutrino Astronomy And Cosmic Alerts	16
4.1.3 Exotic Physics Beyond The Standard Model	18
4.2 The IceCube Detector	19
4.2.1 Geometry Of The Detector	19
IceTop	21
4.2.2 Digital Optical Modules	22
4.2.3 Planned Upgrade of IceCube	23
4.2.4 Detector Triggering, Noise Cleaning and Pulsemaps	24

4.3	IceCube Data	24
4.4	IceCube Simulation Methods	25
4.4.1	GENIE (Neutrinos)	25
4.4.2	MuonGun (Muons)	25
4.4.3	Vuvuzela (Noise)	26
4.5	Types of Events - Tracks Vs Cascades	26
4.6	Oscnext Selection Levels	28
4.7	IceCube Reconstruction	30
4.7.1	RetroReco (Low Energy)	30
4.7.2	SplineMPE (High Energy)	30
5	Machine Learning	31
5.1	Supervised Vs Unsupervised	31
5.2	Supervised Machine Learning Techniques	31
5.2.1	Boosted Decision Trees	32
5.2.2	Neural Networks	33
5.2.3	Graph Neural Networks	34
5.3	How To Train Your Machine Learning Model	35
5.3.1	Loss Function	35
5.3.2	Stochastic Gradient Descent, Learning Rates & Backpropagation	35
5.3.3	Train, Validation, Test Split	37
5.3.4	Hyperparameter Optimisation	38
5.4	GraphNet & DynEdge	38
5.4.1	Benchmark Model Architecture	39
5.4.2	Benchmark Preprocessing	40
5.4.3	Benchmark Loss Functions	41
	Angular Reconstruction - Von Mises-Fisher Loss	41
	Energy Regression - Log-Cosh Loss	41
	Classification And Multiclassification - Cross Entropy Loss	42
	Interaction Vertex Position - Euclidean Loss	42
5.4.4	Benchmark Learning Rate Scheduler	42
6	Improving the Reconstruction of High Energy Northern Tracks	43
6.1	General Method	43
6.2	Learning Rate Adjustments	46
6.3	Changing The Pulsemap	47
6.4	Focus On Highest Energy Range	48
6.5	Gaussian Prescaling Of Input	49
6.6	Going Forward	51
7	Classification And Evaluation Of Neutrino Sample In Real Data	53
7.1	Motivation And Outline	53
7.2	Foundation Of The Results Presented In This Chapter	54
7.3	Data Selection And Train, Validation, Test Split	54
7.3.1	Data Selection And New Muon Sample	54
7.3.2	Training, Validation, Test Split For Each Model	56
7.4	Monte Carlo Classification And Reconstruction Results - Including Benchmark Against Retro Reconstructions	58
7.4.1	Performance Measurement Tools	58
	ROC Curve	58
	Residuals And Resolutions	59

Chi Square Test	59
7.4.2 Multiclassification - Noise, Muons Or Neutrinos?	60
7.4.3 Track/Cascade Classification	62
7.4.4 Energy Reconstruction	66
7.4.5 Interaction Vertex Position Reconstruction	67
7.4.6 Zenith Reconstruction	68
7.4.7 Azimuth Reconstruction	71
7.4.8 Multiclass Neutrino Selection Efficiency	72
7.5 Necessary Data Cleaning Required To Ensure Good Enough Monte Carlo / Data Agreement To Apply GraphNet	73
7.6 Clean Track/Cascade Neutrino Selection and Comparison Between Monte Carlo and Real Data	74
7.7 Comparison Of GraphNet Neutrino Selection With The OscNext Selection In Monte Carlo Data	86
7.7.1 How Many Additional Neutrinos Do We Select In Comparison To OscNext In Monte Carlo data?	86
7.7.2 How Are the GraphNeT And OscNext Monte Carlo Neutrinos Distributed?	90
7.7.3 What Are The Quality Of The Neutrinos In The GraphNeT Selection Compared To The OscNext Selection?	92
7.8 Real Neutrino Events In Real Data	94
7.9 Sneaky Muons That Make It Though To The Neutrino Selection In Monte Carlo	96
8 Conclusion and Potential For Future Work	100
8.1 Improving The Reconstruction Of High Energy Northern Tracks	100
8.2 Classification And Evaluation Of Neutrino Sample In Real Data Using GraphNeT	100
8.2.1 Conclusion	100
8.2.2 Simple Additional Analysis Possibilities And Improvements	102
8.2.3 Long Term Potential	103
A Azimuth Results In Northern Track Benchmark Project	105
A.1 DynEdge Baseline Comparison To SplineMPE	105
A.2 Learning Rate Adjustment	107
A.3 Changing The Pulsemap	108
A.4 Focus On Highest Energy Range	109
A.5 Better Prescaling of Input	110
B Monte Carlo Classification And Reconstruction Results - Including Benchmark Against Retro Reconstructions - Additional Results	111
B.1 Multiclassification - Noise, Muons Or Neutrinos?	111
B.2 Azimuth Reconstruction	112
B.3 Interaction Vertex Position Reconstruction	113
C Necessary Data Cleaning to Ensure Good Monte Carlo / Data Agreement	115
C.1 Level 3 Variables - Cuts And Replicated Plots	115
C.2 Neutrino Comparison Without Lvl 3 Cuts Applied	120
D Comparison of Neutrino Selections in Monte Carlo and Data - Additional Plots	124

D.1	Additional Distribution Comparisons Between Data and Monte Carlo Neutrino Selections	124
D.2	Un-scaled And Variations Of Multiclass Probability (Logit) And Track Probability Plots	133
D.3	Real And Simulated Noise And Muons In Reconstruction and Track/Cascade Classifier Results - Where Would Contamination Show Up?	141
E	Comparison Of GraphNeT Neutrino Selection With OscNext Neutrino Selection - Additional Figures	143
E.1	Additional Figures Showing Neutrino Rates For GraphNeT And OscNext Selections	144
E.2	Additional Figures Showing Difference In Selected Neutrino Properties	148
E.3	Additional Detector Signature Plots for Data Track and Cascade Neutrinos	153
F	Data sources and details	157
F.1	Where is everything located on the HEP server at NBI?	157
F.1.1	Databases And How To Match Events Between Them	157
F.1.2	Trained Model State Dictionaries	158
F.1.3	How To Reweight Events To Get The Estimated Rate?	159
F.1.4	Final Details	159
	Bibliography	161

List of most important Abbreviations

IceCube	IceCube Neutrino Observatory
GraphNeT	Graph Neural Networks for Neutrino Telescope Event Reconstruction
GNN	Graph Neural Network
MC	Monte Carlo
DOM	Digital Optical Module (The individual sensor in IceCube)
CC	Charged Current
NC	Neutral Current
DC	DeepCore (central part of the detector)
EM	Electromagnetic
PMT	Photomultiplier Tube
AGN	Active Galactic Nuclei
ROC Curve	Receiver Operating Characteristics Curve
AUC	Area Under the Curve

Chapter 1

Introduction

Suppose sometime in the future, intelligent aliens were to visit and explore the Earth. In that case, the IceCube Neutrino Observatory (IceCube) in the Antarctic Ice Sheet stands as one of the pinnacles of human achievement. Instrumenting a cubic kilometer of ice in one of the most inaccessible and remote areas of Earth, simply to explore the properties of an invisible particle is almost beyond understanding. But the neutrino and multimessenger astronomy are interesting research topics as will hopefully be clear in this thesis. Neutrinos are incredibly hard to measure since they rarely interact with anything. Even with a detector the size of IceCube, very few neutrinos are measured and they are buried in a vastly larger background of noise and atmospheric muons. Therefore it is of utmost importance for the IceCube collaboration to classify and reconstruct the neutrinos as well as possible. The properties of the neutrinos, for instance the phenomenon of neutrino oscillations, can then be investigated. And the neutrinos can be used in multimessenger astronomy to point to fascinating events in the universe.

This work aims to show that the Graph Neural Network (GNN), DynEdge, from the GraphNeT framework performs classification and reconstruction tasks successfully in a raw selection of actual data. The GNN has been shown to beat existing methods for low energy in Monte Carlo[1], and as will be evident, it seems to successfully classify a substantially larger neutrino selection from raw data than existing methods.

The first part of the thesis, chapter 2, contains a readers guide to be used while reading.

The second part outlines the theory required to understand the remainder of the chapters. It begins with chapter 3 on particle physics and neutrinos in particular. Then follows chapter 4, dedicated to explain the purpose and workings of IceCube, after which machine learning is introduced in chapter 5 with a focus on the methods applied in the analysis presented in this work.

The third part of the thesis consists of two self contained result chapters. In chapter 6 contributions to a high energy northern track benchmark is presented. In chapter 7 the classification and reconstruction of neutrinos in raw IceCube data is available. Furthermore, a comparison of data and Monte Carlo neutrinos are presented, and the rate (amount) of pure neutrinos are compared to existing classification methods.

The final part of the thesis, chapter 8, contains a conclusion of the results and proposed extensions of the analysis.

Chapter 2

Readers Guide

2.1 How To Use This Readers Guide?

This section should not necessarily be read in full to begin with. Instead it represents an outline of some of the more important things to understand when reading through this thesis. The reader is advised to return to it whenever he/she does not remember what is going on. Most of the content in the readers guide will also be introduced in the thesis itself.

2.2 What Is IceCube?

IceCube is a neutrino telescope at the South Pole, which mainly consist of a cubic-kilometer of instrumented ice, as illustrated in figure 2.1

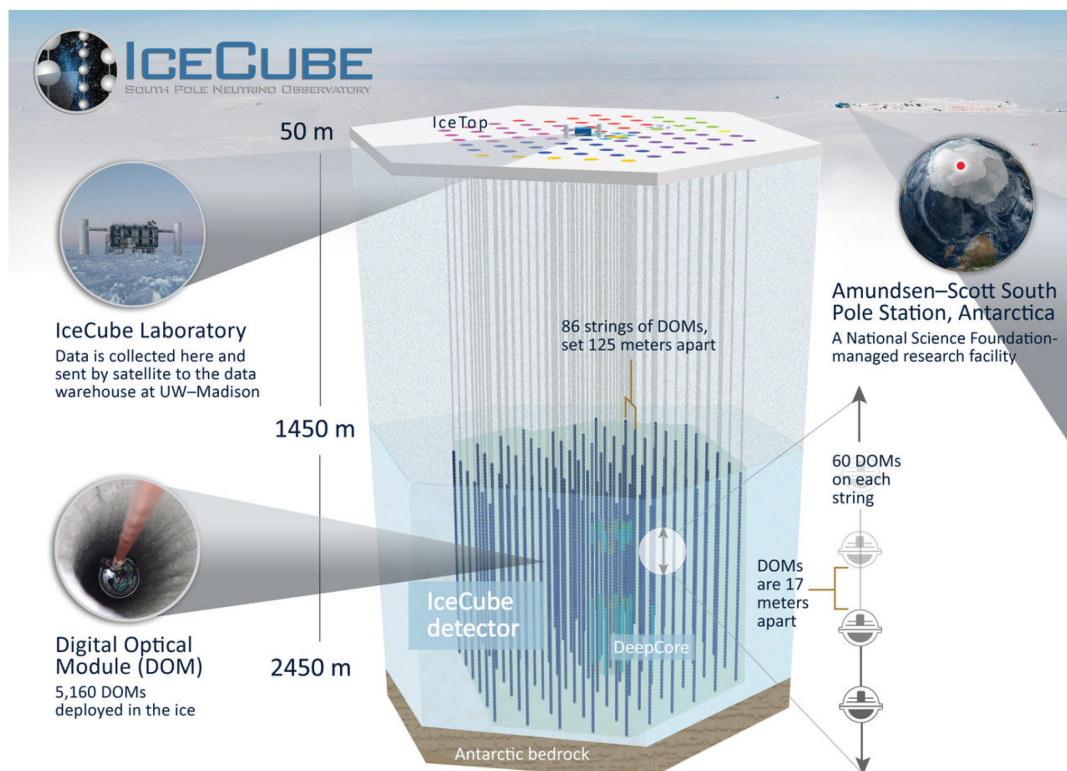


FIGURE 2.1: Overview of the IceCube Neutrino Telescope detector. kindly borrowed from [2].

2.3 What Is The Purpose Of IceCube And What Are The Challenges?

The purpose of IceCube is to observe neutrinos. These observations are for instance used to analyse neutrino oscillation and carry out neutrino astronomy¹. One of the main challenges is that the majority of recorded events in IceCube are not caused by neutrinos. Instead there is a substantial background caused by noise and atmospheric muon.

As seen in figure 2.2, Muons cannot penetrate the Earth, which means they only arrive from the Southern sky (down-going).

Noise has no direction, but appears as a diffuse triggering of individual detectors, which in some cases resemble low energy neutrinos.

Neutrinos are able to travel through the Earth, thus arriving from all directions. Finally, there are both atmospheric neutrinos and cosmic neutrinos. Atmospheric neutrinos are generated when cosmic rays hit our atmosphere, whereas cosmic neutrinos are from elsewhere in the universe. At low energies atmospheric neutrinos dominate over cosmic neutrinos, but at very high energies, the reverse is true.

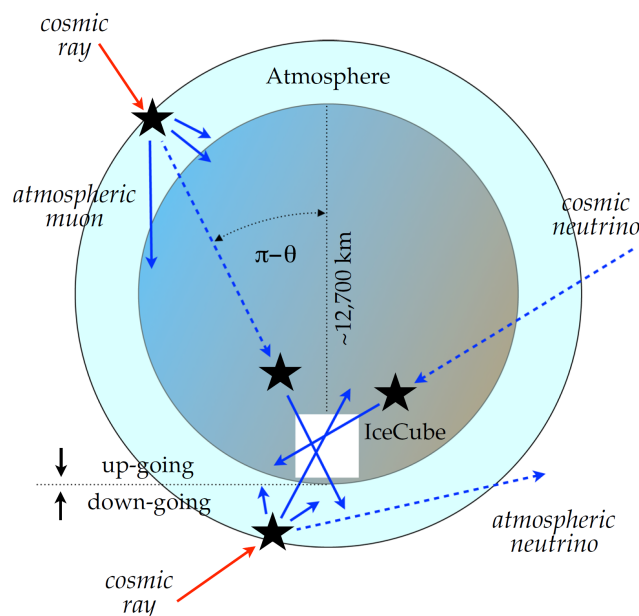


FIGURE 2.2: Overview of which particles can be detected in IceCube. Atmospheric neutrinos are created when cosmic rays hit the atmosphere. Cosmic neutrinos originate elsewhere. Atmospheric muons are the main background in IceCube, but are only down-going since they cannot penetrate the Earth. Neutrinos arrive from all directions and can be detected when they interact (black star) close to or in the IceCube detector. kindly borrowed from [3].

¹IceCube has a variety of other purposes, but these two are the important ones in relation to this thesis.

2.4 What Do Events In IceCube Look Like?

The IceCube collaboration has both simulated (Monte Carlo) and real data. Each event in data consists of a number of DOM (individual light detector) hits, as illustrated in figure 2.3. It shows a high energy event in actual data, which is classified as a neutrino in section 7.6.

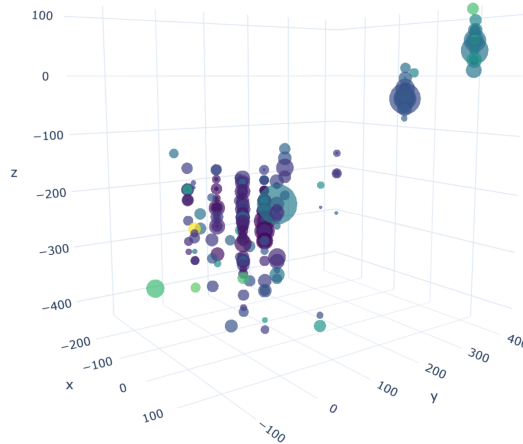


FIGURE 2.3: Event signature for the event with the highest predicted energy in the data track neutrino selection, event number 90278349. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap `SRTInIcePulses` is used. If this caption is not clear, just read ahead and everything should be explained in due time.

Although the simulated data does not match real data perfectly, it is used in this work to train the GNN models. Two data sources are used. Low energy neutrinos, muons and noise events from the oscillation workgroup, `OscNext`. And high energy Northern track neutrinos from what is called the `Snowstorm` dataset.

2.5 What Does This Work Try To Improve On?

The analyses in this work is twofold, but both are related to using GNNs from the open source framework `GraphNeT`, which has been shown to improve the classification and reconstruction of low-energy Monte Carlo neutrinos[1].

The main part of this thesis is an attempt to investigate if the performance of `GraphNeT` carries over to actual data from the oscillation workgroup. An attempt is made to classify neutrinos directly from `lv12 + DC2` "raw" data, and investigate whether or not the selection consists of a pure neutrino sample or not. The selection is also compared to current methods to see if `GraphNeT` can generate a larger neutrino sample. Finally, the additional neutrinos that `GraphNeT` selects in Monte Carlo data are analysed to see if they are of a similar quality to those from the existing method.

The other part of this thesis seeks to improve the reconstruction performance of `GraphNeT` on high energy Monte Carlo track neutrinos, where the current reconstruction is still substantially better.

²What this means will clear once you have read chapter 5.

2.6 Why Use GraphNeT and GNNs To Do So?

GNNs are used because they are perfectly suited to the irregular geometry of IceCube and varying number of input variables. Other attempts at utilizing more complex machine learning techniques include using a CNN (Convolutional Neural Network), which requires you to decompose the detector into sections and rearrange DOMs to get the input to fit the quite restricted format of a CNN.

GraphNeT is an open-source Python framework for using GNNs in neutrino telescopes, which was founded at NBI and is very well made. The first GNN model, which was implemented by Rasmus Ørsøe[1] is called DynEdge. In the analysis sections of the thesis, the terms GraphNeT, GNN and DynEdge is used interchangeably, although DynEdge is an specific GNN model, which is now integrated into the GraphNeT framework.

Recently, in an IceCube kaggle competition it was shown that combining GNNs with Transformers³ leads to an even better performance [4]. Thus for future students, this might be interesting to try.

2.7 Which Variables Are Important?

In this work, various reconstructed variables are of interest. The energy is naturally important and quite self explanatory. On the other hand, the two angles that define the incoming direction of the particles, zenith and azimuth, needs an introduction. The zenith (polar) angle is defined as being 0 directly above IceCube, and π at the North Pole, as seen in figure 2.4. The azimuth angle is defined from the x coordinate in a counter-clockwise fashion as seen from the positive z direction. The z coordinate axis points towards the surface and the y coordinate towards Greenwich.

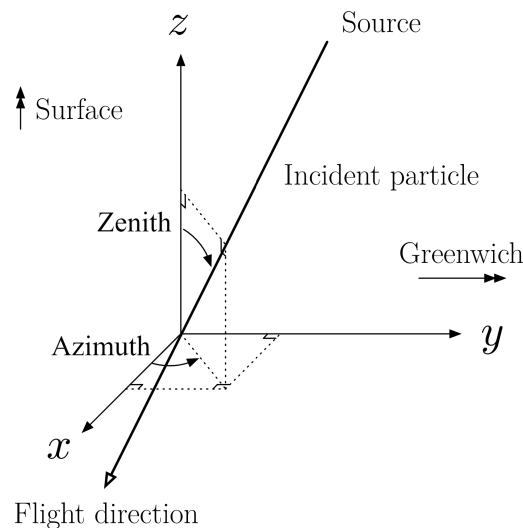


FIGURE 2.4: Overview of the IceCube coordinate system, including the x,y,z position and zenith and azimuth angles. kindly borrowed from [5].

³A new machine learning techniques that add attention to GNNs, effectively allowing them to learn which parts of the input are important for which events.

Finally, the variable ρ is the orthogonal distance to the interaction vertex from string number 36, which is located at $(x,y) = (46.29, 34.88)$, approximately at the center of the IceCube detector.

2.8 How Can The Analyses In This Work Be Replicated?

In general, if anything is unclear, feel free to reach out to the author at peterandresen@hotmail.dk.

2.8.1 Classification And Reconstruction Of Neutrinos In Real Data

The analysis presented in chapter 7 of this work should be easy to replicate for people with access to the HEP (High Energy Physics) server at the Niels Bohr Institute. Python scripts for training and utilizing the GNN models can be found in this location [Analysis On GitHub](#). All plotting scripts are located there as well.

To be able to run the scripts that require using or training GNNs, this branch of GraphNeT should be compatible: [Compatible GraphNeT Branch](#).

The internal IceCube wiki page that explains the analysis presented in this work is available here: [Internal Wiki Page With This Analysis](#).

For the locations and details of the data used and the actual trained models, see appendix F.

2.8.2 Northern Track Benchmark

To get started with the Northern track benchmark (Chapter 6 in this work), this Github page contains the ongoing and finished improvement attempts: [Northern Track Benchmark Project On Github](#).

Otherwise, Rasmus F. Ørsøe is the main person to talk to.

Chapter 3

The Standard Model of Particle Physics

3.1 The Standard Model Of Particle Physics

Imagine every single thing you have ever interacted with. Would it not be amazing if these were all essentially combinations of 17 fundamental particles (not including the antiparticles of quarks and leptons). Well that is exactly how physicists perceive things. The standard model of elementary particles, as can be seen in figure 3.1, contains all constituents of matter along with the particles that allow them to interact via the weak, strong and electromagnetic force. While the gravitational force is not included yet, the achievement is remarkable.

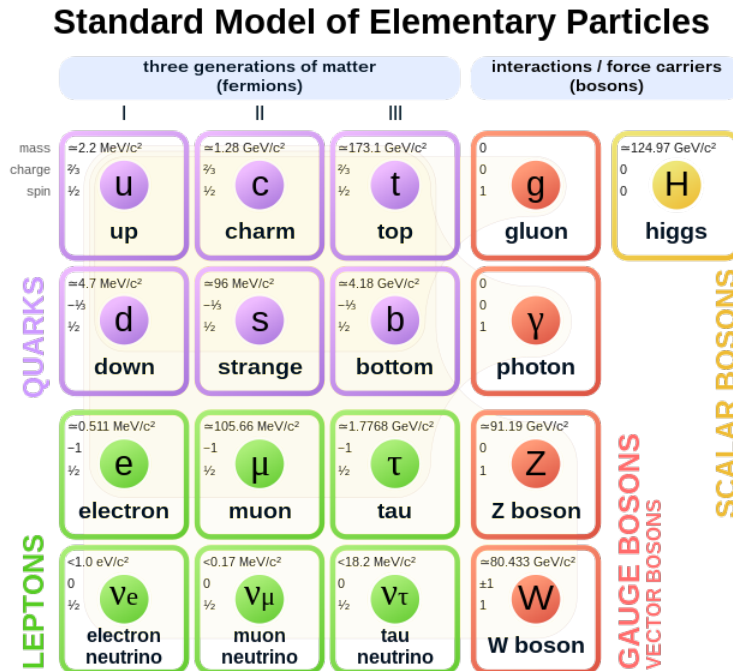


FIGURE 3.1: Overview of the standard model of particle physics taken from [6] & [7]. Includes all fundamental particles, their mass, electromagnetic charge and spin.

3.1.1 Fermions

There are three generations of fermions, where the particles from one family (e.g. up, charm and top) exhibits much the same characteristic across generations, but with increasing mass. The fermions are all spin $1/2$ and each have a corresponding antiparticle. The fermions are further divided into quarks and leptons.

The quarks are particles with an electromagnetic charge of either $-1/3$ or $2/3$ and they carry isospin. As such they interact via the electromagnetic (EM) and the weak force. The quarks also have color charge, meaning that they also interact via the strong force. The strong force grows with distance above a certain threshold, which keeps the quarks contained in color neutral combinations. Thus quarks never appear unbound naturally, a phenomenon called color confinement.

Leptons in contrast do not carry color charge and do not interact via the strong force. They consist of the electron, muon and tau particles as well as their corresponding neutrinos. The neutrinos are neutral particles, in contrast to the electron, muon and tau lepton, which have an EM charge of -1 . This means that neutrinos do not even interact via the electromagnetic force, only with the weak force¹, making them very difficult to detect.

3.1.2 Bosons

If fermions are the building blocks of the universe, then bosons are what allow them to interact. There are four vector bosons (with spin 1) and a single scalar boson (with spin 0). The lone wolf is the higgs boson, which gives rise to the mass of all fermions, with the exception of the neutrinos. This happens through spontaneous symmetry breaking and is quite a complicated process, which is beyond the scope of this thesis.

The scalar bosons act as force mediators. The gluon for the strong force, the photon for the electromagnetic force and the Z and W bosons for the weak force. All bosons are electrically neutral except for the W boson, which can have either -1 or 1 in EM charge. The gluon and photon are massless, whereas the W and Z boson along with the higgs are massive.

Out of all the standard model particles, neutrinos are amongst the most interesting and they lie at the center of this thesis.

3.2 Neutrinos

Neutrinos are some of the most sought after and lesser known particles in the standard model. This naturally arises from the fact that neutrinos only interact via the weak force, which means that neutrino interactions are incredibly rare compared to those of other particles. About 100 trillion pass through a human body every second without interacting in any way whatsoever [8]. Neutrinos also behave in a manner which goes beyond the standard model, making them prime candidates for new exciting physics. They were first predicted by Wolfgang Pauli in 1930.

¹Neutrinos are also expected to interact with gravity, since they have been shown to have masses. Yet the gravitational interactions can be disregarded in comparison to the weak interactions given the small neutrino masses.

3.2.1 Neutrino Prediction And Discovery

The existence of the neutrino was predicted by Wolfgang Pauli in a famous letter. It came as a solution to experimental results which were contrary to the theories of the time. Beta decay was believed to arise from nuclear transitions, which should emit an electron at a very precise energy. However, measurements showed that the emitted electrons had energies across a range that only terminated at the energy they were all expected to have. Allowing another neutral particle (the neutrino) to carry away part of the energy was Pauli's solution to the problem [9].

Once the solution was proposed it became a question of how to experimentally test the idea. Since neutrinos can't be detected directly, another method was required. C. L. Cowan and F. Reines [10] used the process

$$\bar{\nu} + p \rightarrow e^+ + n$$

and the Savannah River nuclear reactor. The experiment was carried out by letting the reaction happen within a scintillation² tank containing $CdCl_2$. The neutrons slow down and are captured by Cd, which produce a photon. The positron also slows down and annihilates with an electron, producing a pair of photons. The light was then captured by photo multiplier tubes (PMTs) [9], in much the same way as we will see happens in the IceCube detector.

In order to detect neutrinos more generally, one must understand the type of interactions they can participate in.

3.2.2 Neutrino Interaction Types

Given that neutrinos only interact via the weak force³, they necessarily do so through either a Z or a W boson. Since the W boson is charged, this is called the charged current (CC) interaction, leaving the Z boson to mediate the neutral current (NC) interaction. Both interactions are illustrated in the Feynman diagram vertices in figure 3.2. Depending on the energy of the neutrino, the reactions with nucleons can be "elastic, quasielastic, inelastic or deep inelastic scattering"[11], which determines if the neutrino interact with the overall nucleus, a particular proton/neutron or a single quark. The interactions produce secondary particles, which are then detectable in the IceCube detector⁴.

²Scintillation is when a luminescent substance is hit by a particle and re-emits the energy as light.

³Their masses are so small that gravity in essence does not play a role for neutrinos.

⁴See section 4.5 for details on detector signatures for each interaction type and neutrino flavor.

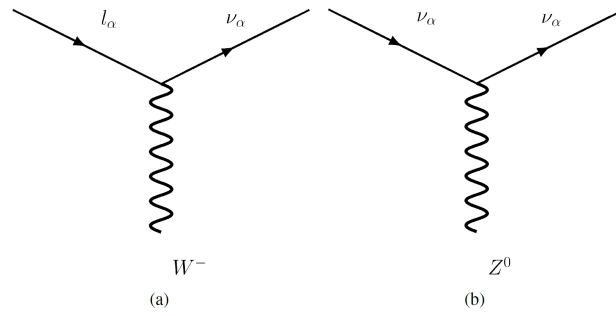


FIGURE 3.2: Schematic representations of the CC (a) and NC (b) weak force interaction vertices for neutrinos. l is a lepton and ν is a neutrino. α represents the flavor, which can be electron, muon or tau (e, μ, τ). The processes can also happen with antiparticles, in which case (a) is mediated by W^+ . Kindly borrowed from [12].

Neutrinos are quite fascinating particles. Even when they do not interact with other matter, they exhibit an interesting behavior called neutrino flavor oscillation.

3.2.3 Neutrino Flavor Oscillation

Neutrinos were at first thought to be mass-less, but this has since been disproved by the observation of neutrino flavor oscillation. Neutrinos are measured in their weak force flavor eigenstates, electron (e), muon (μ) and tau (τ). However, these eigenstates do not have definite mass and are instead combinations of the mass eigenstates (1), (2) and (3).

The unitary matrix that relates the two bases of eigenstates are called the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix, which contains three mixing angles (θ_{12} , θ_{13} & θ_{23}) and a single complex phase δ_{CP} . CP refers to the fact that the phase violates charge-parity symmetry, which is very important⁵, but beyond the scope of this thesis.

The PMNS matrix can be written in the following form⁶ [14]:

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13}e^{i\delta_{CP}} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Which resembles three rotation matrices. Note that $c_{ij} = \cos(\theta_{ij})$, $s_{ij} = \sin(\theta_{ij})$.

The PMNS matrix relates the flavor eigenstates to the mass eigenstates as[14]:

$$\begin{pmatrix} \nu_e(x) \\ \nu_\mu(x) \\ \nu_\tau(x) \end{pmatrix} = U^* \begin{pmatrix} \nu_1(x) \\ \nu_2(x) \\ \nu_3(x) \end{pmatrix}$$

⁵"The observed imbalance or asymmetry in the matter-antimatter ratio may have been produced by the occurrence of CP violation in the first seconds after the big bang(...)"[13].

⁶It should be mentioned that in the case of the neutrino being considered a Majorana particle, two additional complex phases are involved. However, they do not change the oscillation probabilities [14].

Where the $*$ refers to the complex conjugate transpose of the matrix. Once a neutrino is produced in a weak interaction, it is in a flavor eigenstate. However, it propagates through vacuum in terms of its Hamiltonian mass eigenstates, since these have definite mass and as such energy [14]. Given that each mass term is different, the configuration of mass states changes, allowing the probability of observing the neutrino in a particular flavor eigenstate to oscillate. Mathematically it can be written as [14]:

$$|v(t)\rangle = \sum_i U_{\alpha i}^* e^{-iE_i t} |v_i\rangle$$

Deriving the full probability of observing the flavor states for the three state system becomes a bit involved ⁷, but the resulting formula in the case of a two-state neutrino system is quite illuminating. There is only a single mixing angle θ , and it is written as [14]:

$$P(\nu_\alpha \rightarrow \nu_\beta) = \sin^2(2\theta) \sin^2\left(\frac{\Delta m^2 L}{4E}\right)$$

As can be seen, there would be no neutrino oscillations, if their masses were identical (which would definitely be the case if they were mass-less). The oscillation probability depends on the energy of the neutrino, as well as the length it has travelled. These relations are important in the oscillation experiment in the IceCube group, where the appearance of the tau neutrinos from the northern atmosphere is investigated.

The amount of each flavor eigenstate there is in the mass eigenstates, is visualized in figure 3.3. It shows that the electron neutrino is mainly in the first and second mass eigenstate, while the muon and tau are composed of quite even distributions, being more predominant in the third mass state.

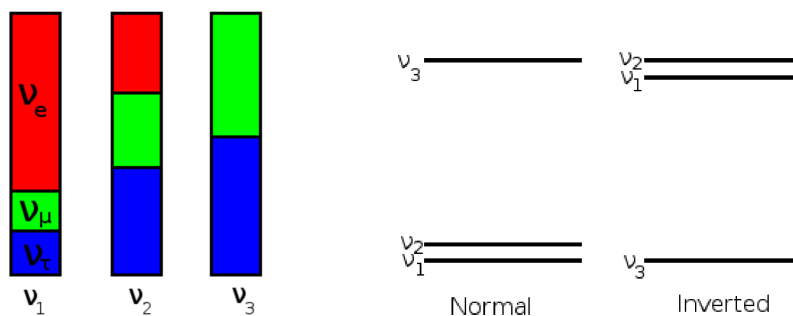


FIGURE 3.3: (left) The distribution of neutrino flavor eigenstates in each of the mass eigenstates. (right) schematic overview of the normal and inverted neutrino mass schemes. Both are kindly borrowed from [14].

Somehow it would be intuitive to assume that the electron neutrino is the lightest (like the electron is lighter than the muon and tau), and as such that the first neutrino mass eigenstate has the lower mass. But given that the mass difference is squared, we cannot know for certain. There are two distinct mass schemes that give rise to the same oscillations. They are called normal and inverted ordering, and are represented

⁷It can be found here [14].

graphically in figure 3.3. The exact values of the oscillation angles as well as the squared mass differences are reported in table 3.1.

	Normal Ordering	Inverted Ordering
$\theta_{12} [^\circ]$	$33.44^{+0.78}_{-0.75}$	$33.45^{+0.78}_{-0.75}$
$\theta_{23} [^\circ]$	$49.0^{+1.1}_{-1.4}$	$49.3^{+1.0}_{-1.2}$
$\theta_{13} [^\circ]$	$8.57^{+0.13}_{-0.12}$	$8.61^{+0.12}_{-0.12}$
$\Delta m_{12}^2 [10^{-3} eV^2]$	$7.42^{+0.21}_{-0.20}$	$7.42^{+0.21}_{-0.20}$
$\Delta m_{3l}^2 [10^{-3} eV^2]$	$2.514^{+0.028}_{-0.027}$	$-2.497^{+0.028}_{-0.028}$

TABLE 3.1: Current best fit oscillation parameters. Note that $\Delta m_{3l}^2 = \Delta m_{31}^2$ for normal ordering and $\Delta m_{3l}^2 = \Delta m_{32}^2$ for inverted ordering. Kindly borrowed from [15].

To carry out the experiments that measure the oscillation parameters, one must be able to detect neutrinos in the first place.

3.2.4 How To Spot A Neutrino In the Wild? - Cherenkov Radiation

Cherenkov radiation was, as the name suggest, discovered by P. A Čerenkov in 1934 [16]. When charged particles move in a polarizable medium, they excite the stationary molecules, which then radiates energy in the form of light in spherical waves (seen from the stationary medium) as they relax back into their ground states. If the particle moves faster than the phase speed of light in the medium, these spherical waves experience constructive interference along a cone shape with the pointy end in the particles path as illustrated in figure 3.4 [17].

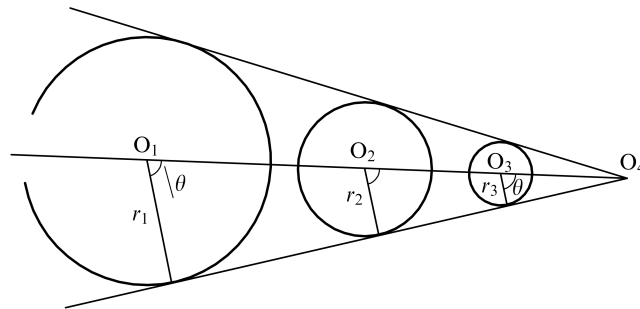


FIGURE 3.4: Illustration of Cherenkov Radiation, where constructive interference of spherical waves give rise to a cone-shape of light in the path of a charged particle moving faster than the speed of light in a medium. Kindly borrowed from [17].

The angle with which this happens, θ on the figure, depends on the speed of the particle (v_p) as well as the speed of light in the medium: $v_c = \frac{c}{n}$, where n is the refractive index of the material. If we consider it in classical terms (disregarding special relativity), after time t , the wave has a radius of $t * v_c$, whereas the particles has moved $t * v_p$. From the illustration and simple trigonometric relations, the angle

θ is given by

$$\theta = \text{ArcCos}\left(\frac{t * v_c}{t * v_p}\right) = \text{ArcCos}\left(\frac{v_c}{v_p}\right) = \text{ArcCos}\left(\frac{c}{v_p * n}\right)$$

This formula becomes slightly more complex with special relativity effects. Furthermore, the formula only holds when the particle propagates faster than the light phase speed, since otherwise $\cos(\theta) > 1$, which is not feasible. Or said without math, if the particle does not move fast enough, the spherical waves do not exhibit constructive interference in a coordinated manner [17].

It is most fortunate that the phenomenon of cherenkov radiation exists, since the IceCube and other large-scale neutrino experiments rely on it to observe neutrinos. They are not directly observable, but the charged daughter particles from neutrino interactions send out cherenkov radiation, which can be detected. Unfortunately there are quite a lot of other charged particles in the detector, besides daughter particles from neutrino interactions. Most of them originate in air showers caused by cosmic rays.

3.3 Cosmic Rays And Air Showers

Now one might be left thinking that the task of finding neutrinos using the cherenkov radiation from the charged leptons, they create or set in motion should be relatively straight forward. However, the vast majority of cherenkov radiation in the detector comes from air showers that are caused by cosmic rays.

Cosmic rays are high energy charged particles (90% protons, 9% alpha particles & 1% heavier nuclei). Their origin is a field of active research, which the IceCube collaboration contributes too, as will be detailed in section 4.1.2.

Once the cosmic rays hit the atmosphere, they interact with the air molecules (mainly nitrogen and oxygen) and cause a cascade of daughter particles called an air shower. These air showers have different components, which are illustrated in figure 3.5.

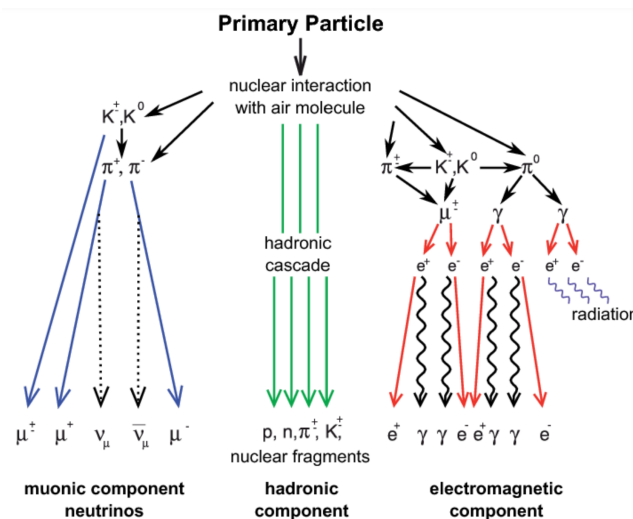


FIGURE 3.5: Illustration of the different components that make up air showers, which are caused by high energy cosmic rays. Kindly borrowed from [18].

The electromagnetic component is caused by photons and electrons, which multiply by pair production (i.e photon creating an electron positron pair) and slow down due to bremsstrahlung (radiation of energy when a particle is accelerated/decelerated) [18]. This component is not very important in IceCube, since the particles do not reach the detector. At best, they can signal that an air shower took place if they are detected at the surface.

The hadronic component mostly consist of mesons (bound states of two quarks), where the pions are predominant; π^+ , π^- , π^0 . They interact with each other and the nucleus's and nucleons in the atmosphere. The neutral pions quickly decay and contribute to the electromagnetic component in the following two ways [18]:

$$\pi^0 \rightarrow \gamma + \gamma \quad (98.8\%)$$

$$\pi^0 \rightarrow e^+ + e^- + \gamma \quad (1.2\%)$$

The charged pions survive for longer, interacting with the atmosphere and generate more pions before finally decaying once the energy is low enough in this manner [18]:

$$\pi^\pm \rightarrow \mu^\pm + \nu_\mu / \bar{\nu}_\mu$$

As such, the charged pions contribute to the muon/neutrino part of the air showers, which are crucial to understand the background in the IceCube detector. The muons travel quite far in ice, allowing those with enough energy to penetrate to the actual detector. Similarly these atmospheric neutrinos can be detected in IceCube, which is interesting for neutrino oscillation measurements. On the other hand they serve as a background over cosmic neutrinos up until very high energies.

Having understood what neutrinos are and how they can be observed, lets turn to the purpose and properties of the IceCube Neutrino Telescope.

Chapter 4

The IceCube Neutrino Observatory

4.1 The Purpose Of IceCube

IceCube was designed with several purposes in mind. In part it explores the properties of neutrinos, contributing to e.g oscillation measurements [19][20][21]. But perhaps as importantly it observes the universe using neutrinos. The weakly interacting nature of neutrinos makes them appropriate for directional pointing, travelling straight from the source, unobstructed and without scattering until they interact in the detector [19][22][23][24]. In addition to this, the IceCube Experiment also searches for exotic physics beyond the standard model, such as sterile neutrinos and dark matter [25][26][27][28][29]. Below follows an introduction to a selection of interesting research areas IceCube contributes to.

4.1.1 Neutrino Oscillation

Understanding the properties of neutrinos lie at the foundation of the IceCube mission. The OscNext group in IceCube carries out neutrino oscillation measurements. The IceCube detector is suited to measure muon neutrino disappearance in atmospheric neutrinos generated when cosmic rays hit the atmosphere. If there were no oscillation, all neutrino types should appear approximate uniformly from any direction in the detector. However, for energies of 5-60GeV, the diameter of the earth is enough to induce a significant probability for muon neutrinos to oscillate to tau neutrinos [20]. The IceCube observatory contributes with measurements of Δm_{32}^2 and θ_{32} ¹, using higher energy neutrinos than more classical accelerator-based experiments, where the neutrinos are generated in particle accelerators [21].

The experiment design is illustrated in figure 4.1. The zenith angle acts as a measure of the distance the neutrino travels from it is generated at the atmosphere to the IceCube detector. This along with the energy of the neutrino determines the probability of muon neutrinos to be measured as tau neutrinos once they hit the detector. It should be noted, that the figure 4.1 (b) is an ideal illustration, which is distorted by matter effects in real measurements.

¹ Δm_{32}^2 and θ_{32} were introduced in section 3.2.3.

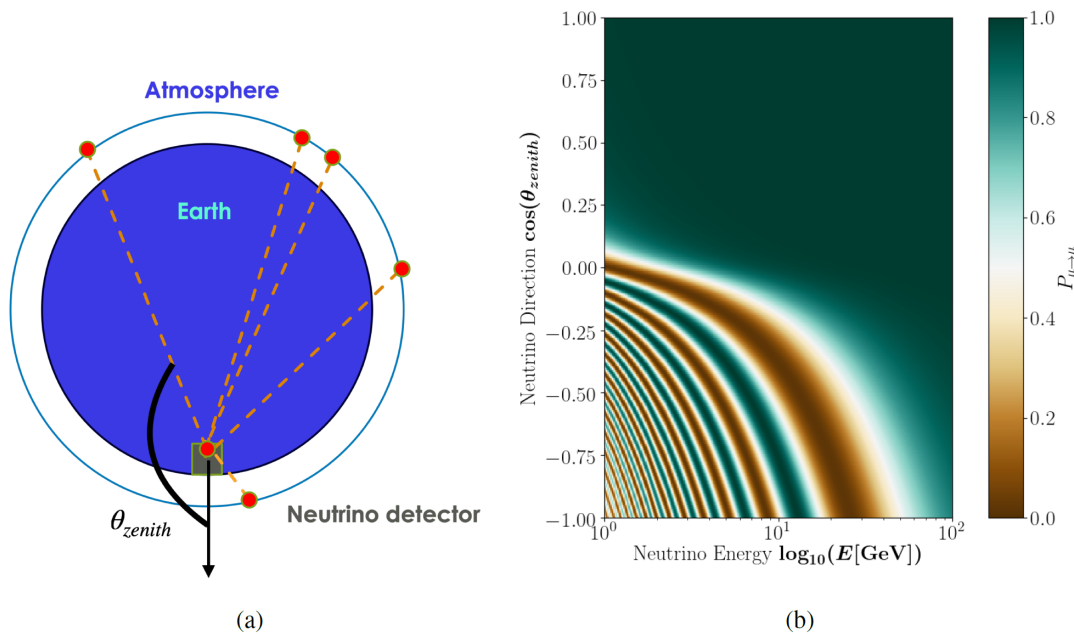


FIGURE 4.1: a) Illustration of the principle behind the neutrino oscillation experiment in IceCube. b) The expected measurement results shows that the probability of muon neutrinos oscillating to tau neutrinos depends on the travelled distance (the zenith angle) as well as the energy of the neutrino. This is an idealised figure, whereas the actual measurements are distorted by matter effects. Kindly borrowed from [12].

4.1.2 Neutrino Astronomy And Cosmic Alerts

Neutrinos interact so weakly that if the direction they come from is reconstructed perfectly, it points directly back to their source. That is, if you can actually detect them. In order to significantly "see" anything using neutrinos, a massive instrument such as the IceCube is required, simply from the need to have enough interactions within the detector [24]. Cosmic neutrinos (not atmospheric) can be a great contribution to multi-messenger astronomy, in which astronomical events are observed using different particle types. If neutrinos are reconstructed real-time, they can serve as early warnings for traditional EM telescopes, which can then point in the right direction. This is possible since neutrinos often arrive prior to the light emitted from high energy events in the universe [24] [30].

The Icecube Experiment has several cosmic alerts in place. The HESE alert triggers on single high-energy track neutrinos starting within the detector and the EHE alert similarly triggers on single extremely high energy track neutrinos originating on the northern hemisphere (going through the earth).

Besides from single neutrino alerts, Icecube has what is called the "Gamma-Ray, Optical and X-Ray Follow-up" [30]. Here, Icecube searches for bursts of neutrinos that are correlated in time and space, in such a manner that there is reason to believe that they originate at an interesting source, which electromagnetic telescopes can then point towards. The type of alert that is sent depend on the time-frame of the signal. Up to 100 seconds gives optical and x-ray alerts (OFU) and up to three weeks

gives gamma-ray alerts (GFU) [30]. Given the background of muons from the southern hemisphere and atmospheric neutrinos, only quite high energy cosmic neutrinos are considered for these alerts. This is due to the fact that only at high energies, do cosmic neutrinos dominate over muons and atmospheric neutrinos[30].

Finally IceCube also searches for the collapse of nearby stars, so-called supernovas. They send out massive bursts of neutrinos with $\mathcal{O}(10\text{MeV})$ energy. Although IceCube cannot detect these neutrinos individually (being 3 orders of magnitude below its sensitivity), it can detect the general increase in DOM activity that results, if the supernova is close enough to Earth. Furthermore, with its size and dark location, IceCube is optimal for detecting these low energy neutrino bursts [31].

Alerts are either issued publically or privately. In the case of the high energy HESE and ESE alerts, the Astrophysical Multimessenger Observatory Network (AMON) is used to access the Gamma-ray Coordinates Network (GCN) to share the alerts publicly [30]. Gamma-Ray, Optical and X-Ray Follow-up alerts are sent privately through the same channels to other telescopes that have an agreement with IceCube. The search for supernovas are the oldest alert and is broadcast trough the Supernova Early Warning System (SNEWS), in which other neutrino detectors participate as well[31].

Understanding the sources of extra-galactic and galactic neutrinos as well as cosmic rays are of importance in itself. In figure 4.2, an overview of the current neutrino flux spectrum (measured or theorized) along with their estimated sources is available.

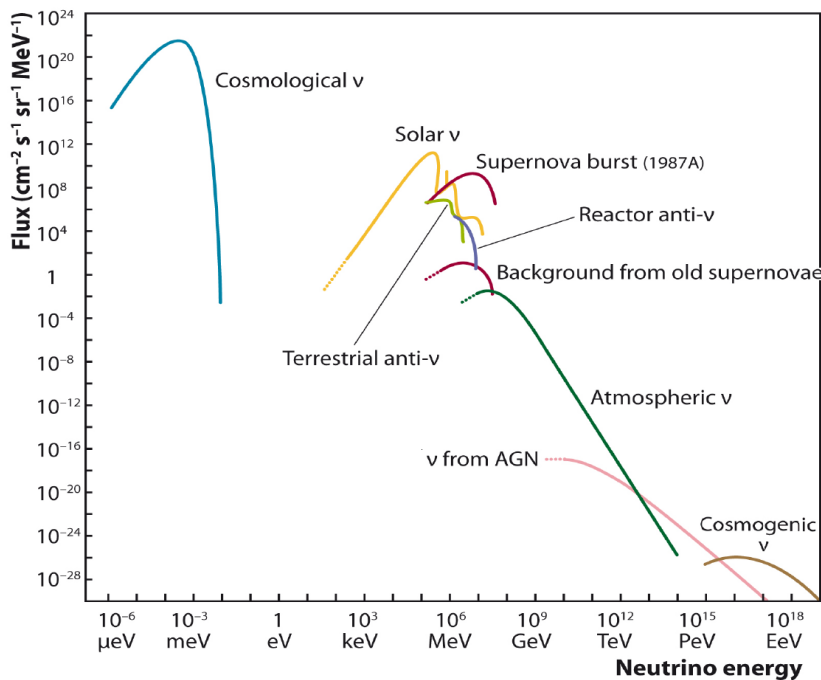


FIGURE 4.2: Overview of the flux (measured or theorized) of neutrinos and their estimated origin as a function of energy. Kindly borrowed from [32].

Currently it is believed by some that galactic rays could be originating in supernova remnants, whereas extragalactic cosmic rays come from supernovas or active galactic nuclei (AGN) [24]. The Icecube is sensitive to Supernova bursts, atmospheric and cosmogenic neutrinos as well as those from AGN.

The IceCube experiment has already published astronomic results. In 2017 IceCube made the first measurement that suggested high-energy neutrinos could originate in flaring blazars. A blazar is an AGN which happens to have its ray of relativistic particles pointed towards earth [33]. A single neutrino with an energy of ~ 270 TeV was observed and its direction pointed towards the known gamma-ray blazar TXS 0506+056. IceCube sent out an alert and more than 20 observatories witnessed that the blazar was in a flaring state [33]. The actual event in the IceCube detector can be seen in figure 4.3 and is a perfect example of a high energy track event. Prompted by the discovery, IceCube analysed all its prior data and found a significant excess of neutrinos from the direction of the blazar. This makes it even more probable that blazars are the first detected source of high energy neutrinos besides the sun and supernovas [34].

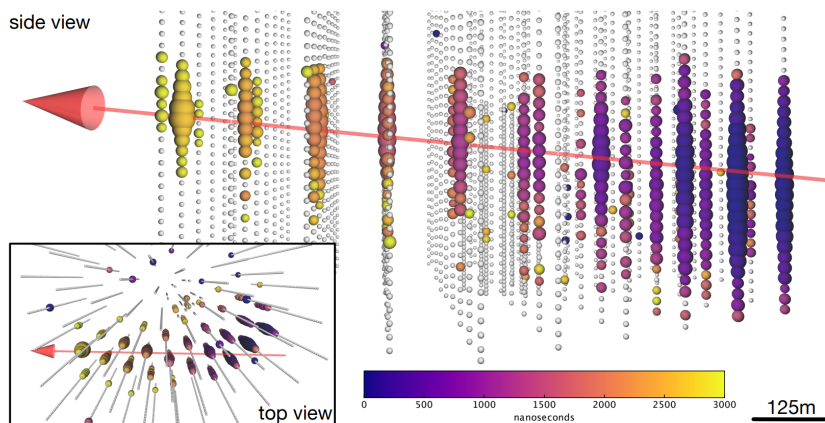


FIGURE 4.3: View of the event that pointed towards the TXS 0506+056 blazar. The colors indicate the time of DOM hits and the size of spheres represents the charge deposited. The event is estimated to have had an energy of ~ 270 TeV and passes completely through the detector. Kindly borrowed from [33].

More recently in 2022, IceCube published results indicating that the active galaxy NGC 1068 is a source of neutrinos, having found an excess of TeV range neutrinos of approximately 79. This direct observation through a large number of neutrinos further establishes AGNs as a source of high energy neutrinos [23].

Improving the astronomic properties of IceCube can be achieved through a better, faster reconstruction. A better angular resolution would improve detection significance and expand the search range. Furthermore, accurate high speed low energy reconstructions could allow for early warnings from low energy neutrinos. This thesis attempts to contribute to a better reconstruction for high energy track-like neutrinos (Chapter 6) as well as fast accurate classification and reconstruction for low energy neutrinos (Chapter 7).

4.1.3 Exotic Physics Beyond The Standard Model

The IceCube detector is able to search for more exotic things than neutrino oscillation and point sources. Two examples are dark matter and sterile neutrinos.

It is generally agreed that a large proportion of the matter in the universe is composed of dark matter, the nature of which we poorly understand. There are various

models that try to explain the phenomenon, one of which the IceCube experiment can contribute to testing. Namely the Weakly Interacting Massive Particles (WIMPs), which is predicted by "supersymmetric extensions of the Standard Model" [27]. How and where does one look for WIMPs then? They do not interact in a way that allows us to measure them directly. Instead, we look for decay or annihilation products (especially neutrinos) from places in which there is a surmount of dark matter [26].

It is theorized that WIMPs can become gravitationally trapped within the core of heavy astronomic objects. Therefore the IceCube detector was used to search for them within the core of the sun [25]. Another study searched for a neutrino excess from the core of the earth [26], while a third investigated the dark matter halo in the milky way [28]. None of the studies resulted in a detection of a significant excess hinting at WIMPS. But they successfully set upper limits on the self-annihilation cross-section and the WIMP-nucleon cross-section. There are still hopes for detecting WIMPs, when the planned IceCube upgrade is installed and the search can be extended downwards in energy [26].

For reasons that are beyond this thesis, so called sterile neutrinos have been theorised. Interacting only via gravity, they are even less participating than their normal companions, which at least also interact via the weak force. IceCube participates in the search for sterile neutrinos of eV scale energy, which could "manifest itself as a resonant, matter-enhanced flavor transition for either muon antineutrinos or muon neutrinos traversing the core of the Earth" [29]. If they exist, IceCube should detect fewer muon antineutrinos at an energy magnitude of TeV. IceCube cannot distinguish neutrinos from antineutrinos, and has to look for an overall deficit in muon neutrinos [29]. Recent published results suggest that there are no such sterile neutrinos within the energy range IceCube investigates [35].

Having understood some areas of IceCube research, lets turn towards the practicalities of searching for neutrinos in the South Pole ice sheet.

4.2 The IceCube Detector

Since neutrinos are inherently difficult to observe, creating a detector with that purpose was never going to be an easy task. Yet that is exactly what the IceCube Collaboration, which includes hundreds of physicists from 14 countries, have achieved. But where does one go about making such an apparatus. It turns out that the clear ice properties on the South Pole are perfectly suited [22].

4.2.1 Geometry Of The Detector

The Icecube detector is made up of a part on the surface (IceTop) and one within the ice that consists of DOMs (Digital Optical Modules), which are distributed on strings throughout approximately one cubic kilometer of ice as illustrated in figure 4.4.

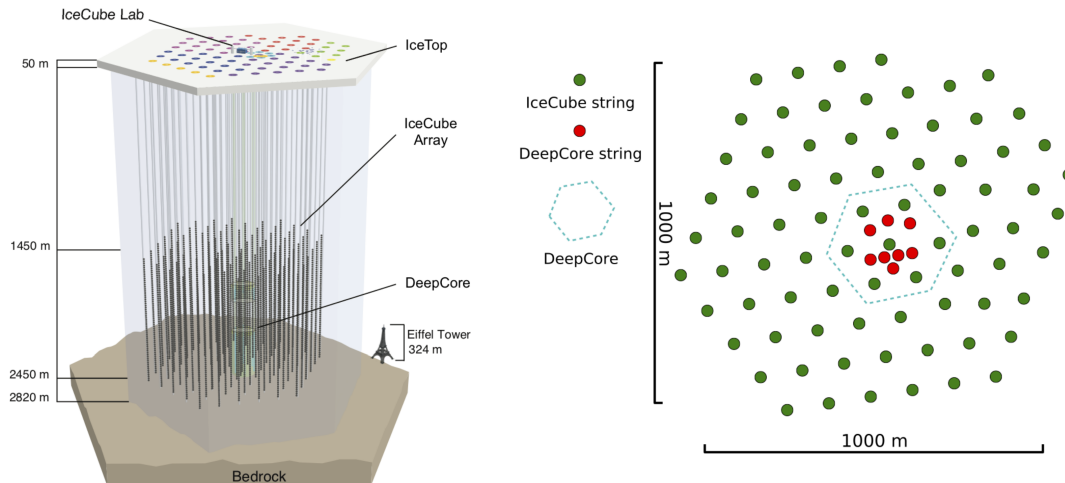


FIGURE 4.4: Overview of the IceCube detector. The green dots represent normal IceCube strings whereas the red strings represent the additional DeepCore strings that are more densely instrumented. kindly borrowed from [19].

The original part of the IceCube detector (green strings in fig. 4.4) are made up of 80 strings on a 125 meter grid. Each string has 60 DOMs with a vertical spacing of about 17 meters located at a depth of 1450-2450m. These strings allow IceCube to detect neutrinos of about 100 GeV and above [24]. To extend the energy sensitivity down to about 10 GeV, eight DeepCore strings were added (red strings in fig. 4.4). They are about 55 meters apart and have 50 DOMs below the dust layer² with a vertical spacing of about 7 meters as seen in figure 4.5. They also have 10 veto DOMs above the dust layer with a vertical spacing of 10 meters. Six of the DeepCore strings have special high quantum efficiency (HQE) DOMs [36].

²A section of Ice where the optic properties are worse than the rest, due to a higher concentration of dust.

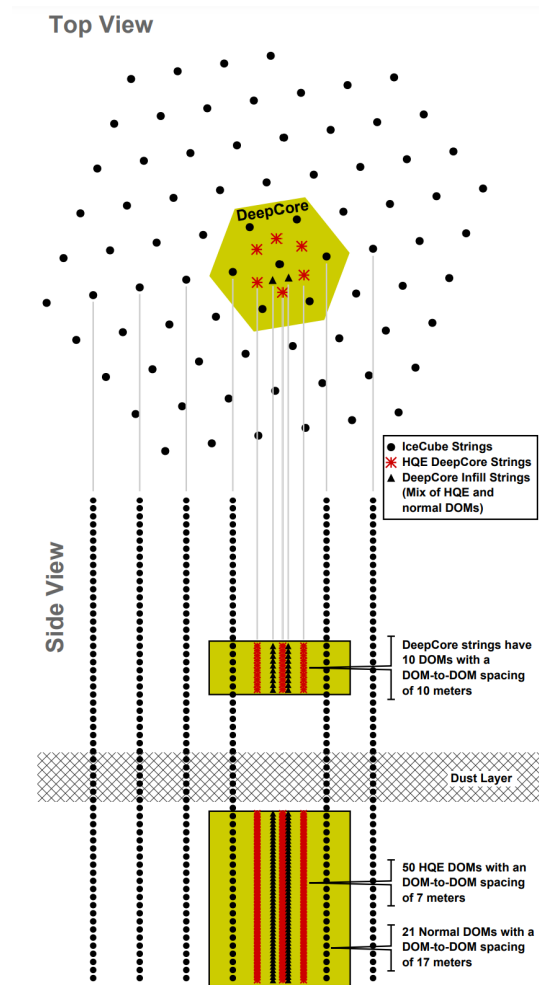


FIGURE 4.5: Overview of the IceCube detector from [36]. Photo shows the vertical spacing of DeepCore and traditional strings including the DOM positions above and below the dust layer.

IceTop

The IceTop array is an extensive air shower detector. It consists of 81 stations, each of which has two tanks spaced 10 meters apart. Each IceTop station is placed "on top" of one of the strings that are connected to the deep array, as can be seen in figure 4.6 [37].

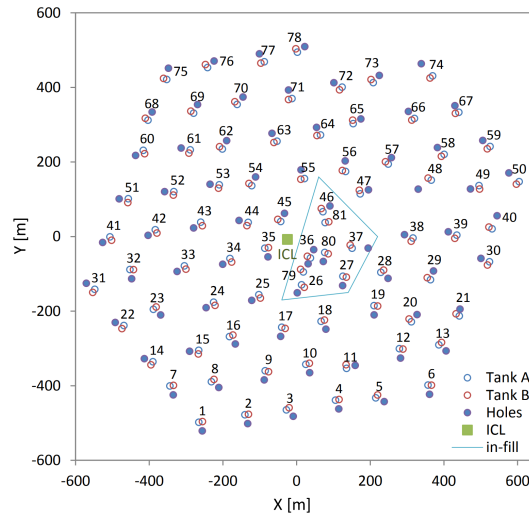


FIGURE 4.6: Overview of the IceTop array. Each IceTop station consists of two tanks, A and B, which contains two classic DOMs for detecting air showers. Kindly borrowed from [37].

The IceTop tanks contain two classical DOMs, suspended in clear ice, which is used to detect the Cherenkov radiation from charged particles in air showers, as illustrated in figure 4.7.

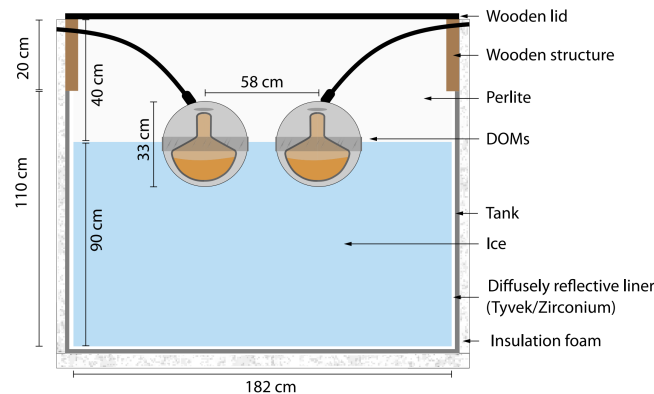


FIGURE 4.7: Illustration of the two DOMs that are inside each IceTop tank. Kindly borrowed from [37].

The IceTop array is mainly used to study the atmospheric air showers. But interestingly it can also act as a veto for the deep detector array. As was seen in section 3.3, muons are created by cosmic ray air showers and can travel all the way into the deep IceCube detector, causing a massive muon background. Some of the muons can be filtered away, since they are also detected in IceTop or their timing is consistent with an air shower [18].

4.2.2 Digital Optical Modules

The DOMs are designed to measure the Cherenkov radiation (light) emitted by the charged particles that pass through the detector at relativistic speeds. DOMs are spherical glass containers whose main measurement device is a photomultiplier tube (PMT) that detect the incoming photons. The DeepCore HQE DOMs have a more

sensitive PMT, which increases the likelihood of triggering on lower neutrino energies [36]. The DOMs also contain an LED which is used for calibration, as well as a power supply and mechanics to store signals. The DOMs are connected on strings, which are essentially caples that allow the DOMs to communicate with the IceCube Laboratory (ICL) on the surface [38].

4.2.3 Planned Upgrade of IceCube

A plan exists to upgrade the current detector with 700 new DOMs on seven new strings withing the current DeepCore array. The upgrade will expand downwards the energy sensitivity to a few GeV, and allow for more precise measurements of the tau neutrino appearance. Furthermore, it will improve the calibration and understanding of the ice properties through new calibration devices [39]. Instead of deploying more of the classical DOMs, the existing model has been improved and two new types have been developed. All three are visualized in figure 4.8.

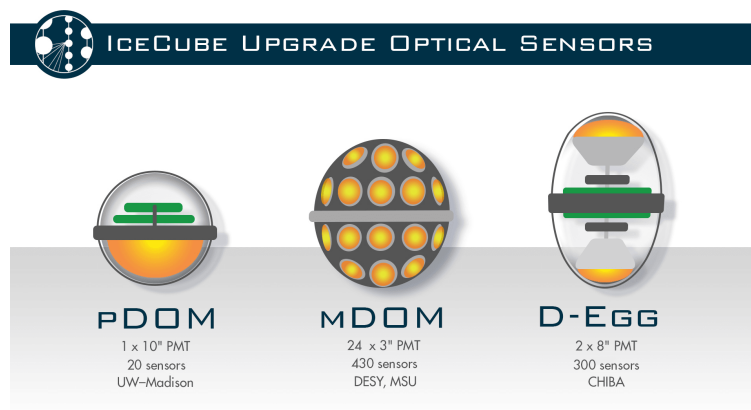


FIGURE 4.8: Illustration of the planned DOMs in IceCube upgrade.
Image kindly borrowed from [40].

The pDOM is very similar to the current IceCube DOM, but features upgraded electronics. In contrast, the D-EGG has two slightly smaller, but high QE PMTs, pointing upwards and downwards, effectively allowing the D-EGG to detect particles from both directions [41]. Perhaps more importantly, it features a calibration LED, which will be used to probe the properties of the "hole ice". When the ice was melted and strings were inserted for the detector, the re-frozen ice properties became different from the surrounding ice, with for instance a higher concentration of air bubbles. This has been shown to have a large systematic effect, which can be reduced using the calibration system of the D-EGGs [42].

The other new DOM type is the multi-PMT DOM (mDOM), which features 24 PMTs uniformly distributed on a sphere. This allows for a single mDOM to obtain information on which direction the particle was propagating, simple from detecting the same signal using multiple PMTs [43].

The three DOM types will also be used in the future IceCube Gen2, which will expand the detector volume to $8km^3$, improving the detection and increasing the sensitivity for high energy neutrinos [44].

4.2.4 Detector Triggering, Noise Cleaning and Pulsemaps

How then, does an event in IceCube actually get recorded. To understand this, one must know how a single DOM is triggered. Each DOM is calibrated every year and a voltage threshold is established, which has to be exceeded for a single multiplicity trigger (or a hit) to be recorded. Furthermore, once a DOM is triggered, it communicates with its nearest neighbours, to check if any of them were also hit within $\pm 1\mu\text{s}$. If that was the case, the hit is recorded as a hard local coincidence (HLC) and otherwise as a soft local coincidence (SLC). Depending on the type, different waveforms (voltage as a function of time) are recorded³ [12]. The waveforms are also converted into a pulse-series, which contains a discrete number of pulses (each carrying a specific time and charge).

Filters are then applied to select events of relevance to the IceCube working groups. For instance, a typical high energy filter is requiring more than 8 hits, either SLC or HLC. For the oscillation group, the DeepCore filter is used. This requires more than three hits. The filter then applies a Seeded Radius-Timing (SRT) and a Time Window (TW) cleaning. The SRT cleaning starts with the HLC hits and includes SLC hits that are within a radius and time around the hit, such that it could be causally related. These hits are then included in the post-cleaned selection and the process repeats up to three times [45]. The TW cleaning simply removes pulses outside a specific interval around the event trigger time. Finally in the DeepCore filter, another veto is applied, which checks to see if any of the hits near DeepCore could be causally related to those nearer to the edges (thus potentially coming from a muon crossing the detector)[45].

The event information is recorded in so-called pulsemaps, which contain information about the DOM hits. For each event, there are several pulsemaps that corresponds to different methods of removing noise. Further details on the IceCube data is presented in section 4.3.

4.3 IceCube Data

The data from IceCube is written in a format called I3 files. Each event in the detector, i.e a single instance of connected DOM triggers is saved individually and contains a variable number of hits (when a DOM detects a photon). For each hit, the seven variables in table 4.1 are recorded and will be used as inputs to the machine learning in this project.

³More information on the waveforms can be found here: [12].

Variables	Description
Dom_x	X position of the DOM relative to the center of the IceCube detector array
Dom_y	Y position of the DOM relative to the center of the IceCube detector array
Dom_z	Z position of the DOM relative to the center of the IceCube detector array
Dom_time	The time the DOM was hit relative to the first hit in the event
Charge	The charge that was recorded by the PMT in the DOM, which is related to the number of photons it was hit by
rde	A measure of the relative quantum efficiency of the DOM, which is the sensitivity to photons. Two possible values, one for normal DOMs and a higher for specific HQE DOMs on some DeepCore strings
PMT_area	The area of the Photomultiplier tube

TABLE 4.1

In order to understand IceCube data, detailed simulations are run to create Monte Carlo data.

4.4 IceCube Simulation Methods

4.4.1 GENIE (Neutrinos)

GENIE is a neutrino event generator which is used in a multitude of neutrino experiments, including IceCube. It relies on Monte Carlo simulations to generate neutrino events across a wide range of energies from MeV to PeV [46]. The GENIE simulation is object-based and separates the mechanics of the simulation from the physics models. It incorporates nuclear physics (relativistic fermi gas model), cross section models (which determines what interactions take place and the kinematics) and neutrino-induced hadronic multiparticle production modeling (which models the hadronic showers that are produced from interactions) [46]. The GENIE simulation is used in the oscillation analysis of the IceCube experiment and it is the basis of all neutrino events used in this work.

4.4.2 MuonGun (Muons)

MuonGun [47] is an efficient muon simulation tool, which relies on the more advanced Corsika [48] air shower simulation. Using a cylinder generator volume, which is illustrated in figure 4.9, each muon from a Corsika simulation that passes the boundaries of the volume are investigated. They are then only simulated if they pass through DeepCore. Afterwards the muon distributions are weighted such that they match those of Corsika as closely as possible [47].

There are two caveats of using muons from MuonGun. First, they are only kept if they point towards DeepCore. This makes sense if the analysis is neutrino oscillation, which requires deepcore precision, but it might not be optimal for other aims.

Furthermore, the muons are simulated individually, which means that it is not possible to have multiple muons hitting the detector in a single event. This occurs in about 10% of real events, which could make the comparison between real data and our simulated events problematic. MuonGun is used in this work, since the events were already available from the OscNext analysis.

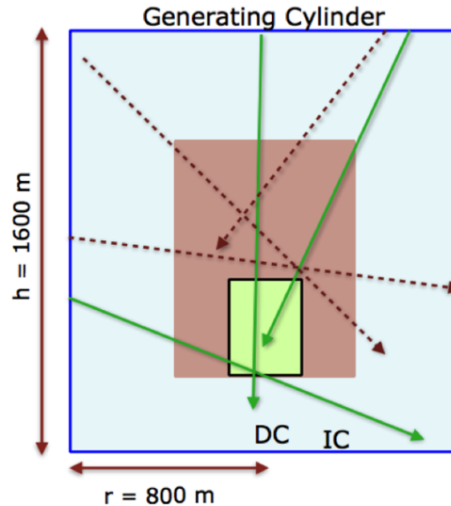


FIGURE 4.9: Illustration of the MuonGun simulation method, which only simulates single muons from Corsika that pass through the generator and aims towards DeepCore. Image taken from [47].

4.4.3 Vuvuzela (Noise)

The noise simulations used in this thesis comes from the tool Vuvuzela. It was developed in response to the need for better noise modelling after the installment of DeepCore, given its higher sensitivity and less energetic events [49]. Noise had previously been estimated as an uncorrelated Poisson process, which turned out to be insufficient. It was evident that sometimes noise hits would happen in "bursts" within the same DOM [49] and that there was a temperature-dependent part of the noise, both of which could not be captured by existing models [49].

The Vuvuzela noise model tries to capture these effects. It consists of three parts. "the uncorrelated thermal noise, the uncorrelated radioactive noise, and the correlated scintillation noise" [49]. The first part is the classical Poisson process but temperature dependent. The second part is another Poisson process, but for radioactive decay in the glass of the DOM and PMT. These decays give rise to particles, which then trigger the scintillation process in the glass and creates light. The number of hits is characterized by another Poisson distribution independently estimated for each DOM and the time between hits come from log-normal distributions which are also evaluated from data. In the end, using the vuvuzela noise model improved the agreement between simulated and real data [49].

4.5 Types of Events - Tracks Vs Cascades

All neutrinos are not created alike. Therefore they also do not leave the same impression on the IceCube detector. In general there are two types of event signatures, called tracks and cascades. Depending on the flavor and interaction type (CC vs

NC), the neutrino interactions produce either of the two signatures as specified in figure 4.10.

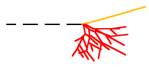
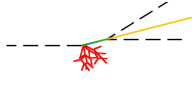
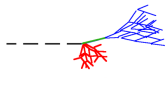
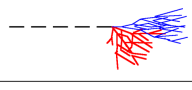

Interaction	Secondary particles	Detector signature	
CC ν_μ		μ track and hadronic cascade	Track with cascade
CC ν_τ		τ decays into μ ($\sim 17\%$ b.r.)	
		τ decays into e / hadrons	
CC ν_e		Hadronic and EM cascades	Cascade
NC ν_α		Hadronic cascade	

FIGURE 4.10: Overview of the types of detector signals each interaction type generates, depending on the neutrino flavor. Kindly borrowed from [50].

In neutral current interactions the neutrino generally transfers a third of its energy to a quark in the ice, "producing a short ($\sim 5\text{m}$) shower of relativistic charged particles" [51], which results in a cascade-like shape of light which is detectable in IceCube.

Charged current interactions are different in nature. They produce an initial hadronic cascade of particles, along with a lepton of the same flavor as the incoming neutrino [51]. CC muon neutrino interactions generate a long lived muon, which can traverse up to kilometers of ice [52]. Given that the muon propagates faster than the speed of light, it can outrun the cherenkov radiation of the initial hadronic cascade which is also produced in the interaction, resulting in the characteristic track-like pattern [53].

The CC electron neutrino interaction creates an electron which results in an electromagnetic cascade, since the electron is easily scattered and loses energy quickly. Finally the tau neutrino CC interaction creates a tau, which would in principle leave a track, but simply does not live long enough to travel far enough in ice to be observable⁴. Instead the tau decays to either an electron (83%) creating an EM cascade, or a muon (17%) which then leaves a track if it has enough energy [51].

In principle if the tau neutrino has enough energy, it could generate a so-called double-bang signature, where the first bang is the initial hadronic cascade, and the second is the EM cascade that is produced when the tau has travelled a distance and decays to an electron. This has not yet been observed in IceCube [54]. The event signatures are visible in figure 4.11.

⁴At least for energies below a few tens of PeV.

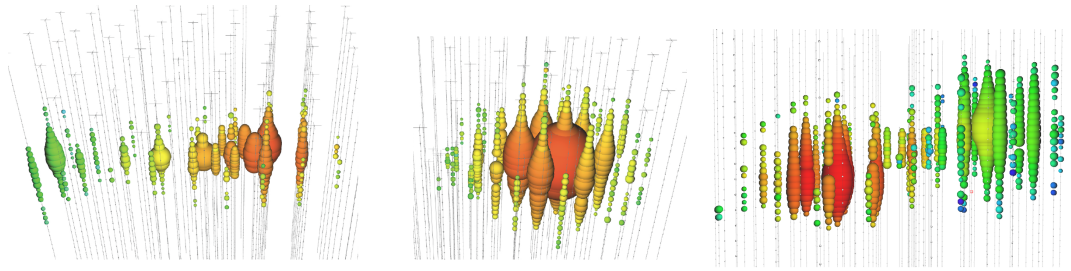


FIGURE 4.11: Example of the types of event signatures that are observable in the IceCube detector. (left): track. (middle): cascade. (right): double-bang. The double-bang is a simulated event, whereas the others are real data events. Kindly borrowed from [54].

To arrive at a pure neutrino sample that can be analysed, the IceCube data goes through a series of cleaning steps in the OscNext group.

4.6 Oscnext Selection Levels

The oscillation group at IceCube, OscNext, relies on a data cleaning process which are ordered in levels to get a pure neutrino selection. The rate of each particle type at each cleaning level can be seen in figure 4.12.

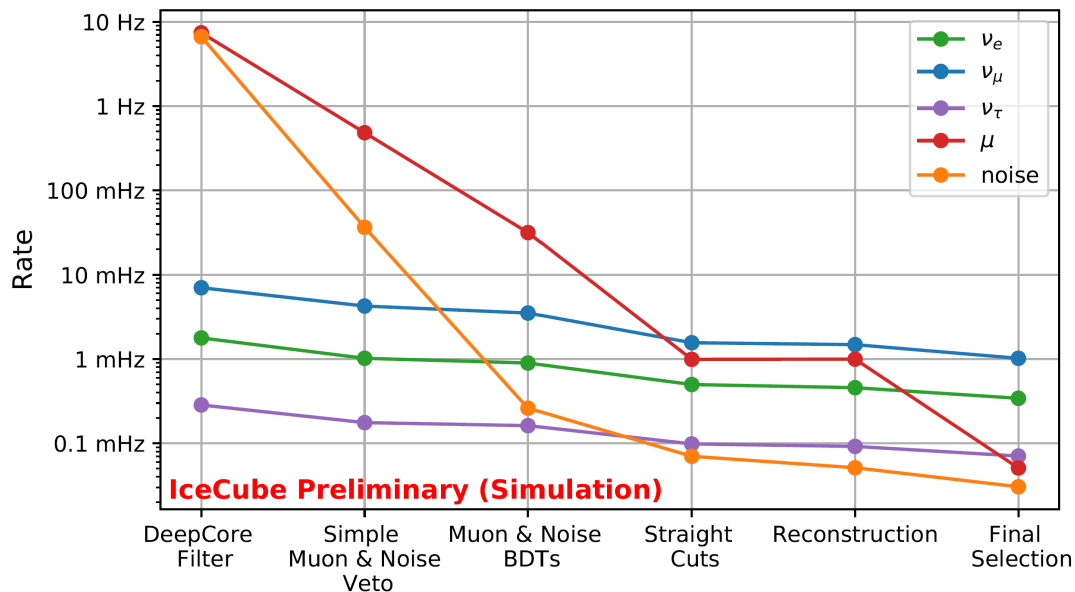


FIGURE 4.12: Survival rate for each particle type in the event selection levels used by OscNext. From the left, the names are also referred to as lvl2 + DC, lvl3, lvl4, lvl5, lvl6 and lvl7. Kindly borrowed from [12]

The selection levels are presented below and further information is available here: [12][45]

2. Level 2 is the IceCube wide initial event selection, which is used as a basis across the collaboration. In the OscNext group, a DeepCore filter is added, which seeks to remove atmospheric muons, by investigating if hits in the veto

region of the detector (everything that is not around DeepCore), is likely to have caused any hits in the fiducial region (around DeepCore).

3. Level 3 uses a selection of simple cuts to improve the similarity between simulated data and actual data. The cuts remove the obvious muon and noise events which are easily identified. It relies on variables such as the number of cleaned DOM hits, z-coordinate of first cleaned DOM hit, number of hits, time duration of events etc. The level 3 selection also seeks to remove coincident⁵ events, which are not present in the simulated data. The level 3 cuts will turn out to be important to the results in this work and we will return to it in section 7.5.
4. Level 4 consist of simple machine learning algorithms called Boosted Decision Trees (BDTs)⁶, more specifically LightGBM's version. At this level, it is thought that data/MC agreement is sufficient to yield good results. Two individual classifiers are trained, one selecting noise and one selecting muons.
5. Level 5 uses another two sets of cuts. One of which tries to remove events that happen at the edge of the fiducial volume. The other is a corridor cut, which removes events where muons could have passed through one of the "channels" with few detectors in IceCube as illustrated in figure 4.13.
6. Level 6 is where the first real reconstruction happens. RetroReco (described in section 4.7.1) is applied as well as two BDT classifiers, one to identify the remaining muons and one to identify tracks (CC μ_{nu}) from other neutrino interactions. At this level only a few events are removed, due to a bad reconstruction.
7. Level 7 uses the level 6 reconstruction to try and remove the sneaky muons, using the reconstructed variables and another BDT. Finally the flavors of the remaining neutrinos are classified and the event selection is finished.

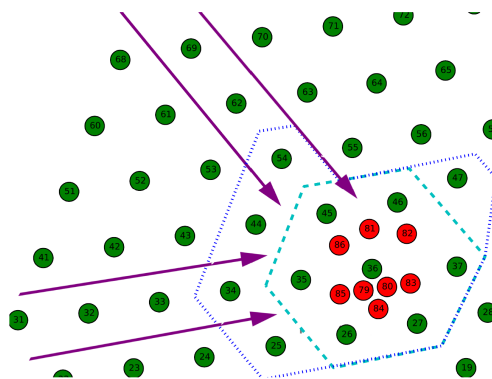


FIGURE 4.13: Example of corridors with thinly instrumented ice where muons can enter DeepCore without being detected before. Kindly borrowed from [12].

IceCube also relies on precise reconstruction methods to specify event properties such as energy and direction.

⁵More than one particle hitting the detector simultaneously.

⁶See section 5.2.1.

4.7 IceCube Reconstruction

IceCube has multiple reconstruction and classification algorithms, two of which are used as benchmark to compare this work against. RetroReco (low energy) and SplineMPE (high energy).

4.7.1 RetroReco (Low Energy)

RetroReco is a low energy likelihood table-based reconstruction algorithm, used by the oscNext group. It is computationally intensive, taking about ~ 40 s per event⁷, which is why it is not used prior to level 6 in the selection process [55].

RetroReco relies on four main parts: "light survival probability tables, event hypothesis, likelihood function and optimizer"[55]. Firstly, for each DOM, there are large tables that contain the survival probability of photons, as a function of e.g the angles of the direction they are emitted in. Having a single table for each DOM is too memory intensive, which means that they are grouped in clusters with similar properties.

The hypothesis of the events are constructed using eight variables, the vertex position and time, the potential track length and cascade energy as well as the incoming zenith and azimuth angles. Finally Retro relies on a likelihood function that accounts for the probability of observing said charge in the DOMs at the specific times and an optimizer to maximize said likelihood[55].

4.7.2 SplineMPE (High Energy)

SplineMPE is another reconstruction method specifically designed for muon tracks at high energies. It uses GPUs to produce "very fine binned Cherenkov light distributions [...] for all muon configurations."[56]. To reduce memory, a multidimensional spline was fit to the distributions, which is basically a tool to smooth out curves or surfaces between points. These splines are then used in likelihood optimisation as the probability density function [56]. SplineMPE is e.g used to reconstruct high energy northern tracks (CC muon neutrinos coming from the northern sky).

In chapter 6, GraphNets performance against SplineMPE on these tracks are investigated and sought optimised. But before that is possible, an understanding of machine learning in general and our model in particular is required.

⁷As people who do machine learning are wont to point out, this can be sped up by several orders of magnitude, while hopefully achieving as good a reconstruction. Stay tuned for section 7.4.

Chapter 5

Machine Learning

Machine learning is a discipline in which computers are "taught" to perform specific tasks, which they can often do with greater success than humans. Machine learning has greatly developed in recent decades to a point where it is now applicable within fields as varied as medicine, physics and finance. Here follows a quick introduction to machine learning in general and a deep dive into the particular methods applied in this work.

5.1 Supervised Vs Unsupervised

Machine learning can generally be classified as either of two categories¹, supervised and unsupervised learning. The main difference between the two are whether the computer is trained on data where the truth is known or not [57].

In supervised learning, the computer is trained on data with a truth value and the goal is to predict as accurately as possible while remaining general enough to work on new data². Supervised Learning consists of two main categories, regression and classification, both of which are applied in this thesis[57].

Regression is using the input data to try to predict a numeric value, such as the energy or incoming angles of a neutrino hitting the IceCube detector. The continuous output is what distinguishes it from the other type of supervised learning; classification. Classification is used to try to predict into which category a sample falls. A classical example in IceCube is determining if an event in the detector is noise, an atmospheric muon or a neutrino interaction[57].

Unsupervised Learning is quite different. It deals with topics such as clustering and dimensionality reductions. It is less useful within IceCube and not used in this work, which is why no further details are provided.

5.2 Supervised Machine Learning Techniques

The methods introduced in this section are used in this work or required to understand existing methods in IceCube.

¹Although some semi-supervised methods lie between the two. Also reinforcement learning is perhaps a third category in which the computers is trained in iterations by e.g playing chess.

²More about this in section 5.3.3, where the importance of dividing the data into a training, validation and test set is explained.

5.2.1 Boosted Decision Trees

Boosted decision trees (BDTs) is a machine learning technique that works on spreadsheet like data. It starts with a simple decision tree, which takes any number of inputs and depending on a "tree" of cuts, it outputs either a classification label or a regression value as illustrated in figure 5.1. The figure contains a decision tree, which determines if a golfer ends above or below par, depending on a few variables. The amount of splits, nodes and leaves is flexible and the optimal cuts are determined by gradient boosting³. In this case it makes yes/no splits, but they could also be cuts based on continuous variables[58].

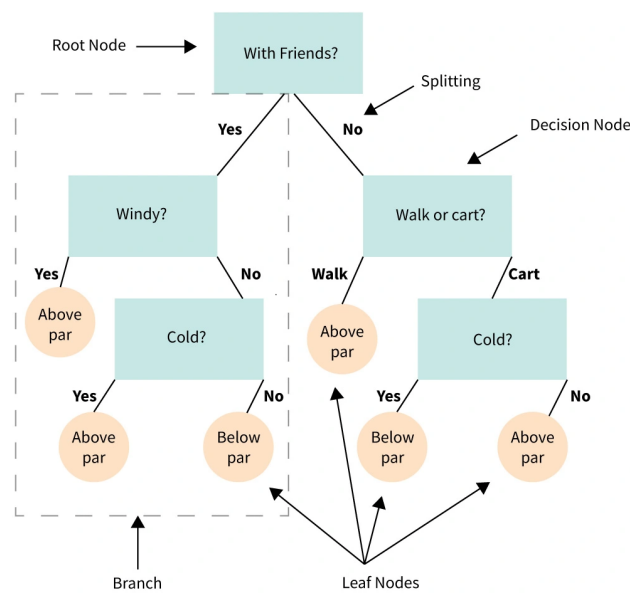


FIGURE 5.1: Overview of how a decision tree arrives at its predictions. Kindly borrowed from [59].

A boosted decision tree is a weighted average of individual decision trees that have been trained on only parts of the training data. In each iteration, the parts of the training data that is difficult to predict is assigned a larger importance in the loss function of the next decision tree in the ensemble. This method resolves some of the limitations of decision trees, namely that they are only able to make linear cuts and have issues with generalization[58].

The BDT algorithm that is used in the level 4 selection in the OscNext analysis is LightGBM, which is a relatively new algorithm. It has become very popular since it handles large data sizes well, keeping almost the same precision as other software while being approximately 20 times faster [60]. BDTs are easy to set up and train, but for complex data they do not perform as well as certain types of neural networks.

³See section 5.3.2

5.2.2 Neural Networks

When trying to teach a machine to "think", it was perhaps natural to turn to the human brain for inspiration. This inspiration led to the Neural Network (NN) algorithms, of which the first working version, the Perceptron was demonstrated in 1957 [61].

Just as a brain is made up by neurons, so is a neural network. Each neuron has a set of inputs and a single output as is illustrated in figure 5.2 in the case of two inputs.

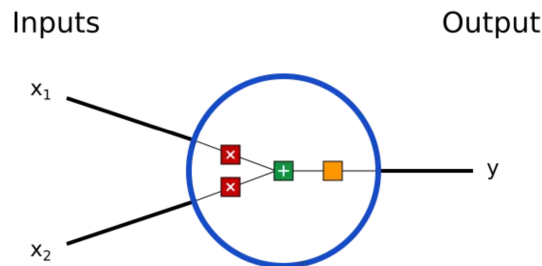


FIGURE 5.2: Schematic illustration of a neuron in a neural network. Kindly borrowed from [62].

The neuron works by applying the following formula[62]:

$$y = f(x_1 * \omega_1 + x_2 * \omega_2 + b)$$

ω_1 and ω_2 are weights that are applied to the inputs, and b is a bias that is added to their sum. This is then passed to an activation function, f , kind of like when a neuron in a brain triggers, making the whole network non-linear [62]. There is a variety of activation functions, but the one used in the models in this thesis is the leaky relu, f_{lr} :

$$f_{lr}(x) = \begin{cases} x, & \text{if } x > 0 \\ ax & \text{if } x < 0 \end{cases}$$

Where a is a small positive number.

To get from a neuron to a neural network is as simple as grouping neurons together. In the network, the output of the first layer of neurons acts as input to the second layer. The neurons can be connected in various ways, but a typical fully connected⁴ example is available in figure 5.3. A deep neural network simply refers to having multiple hidden layers.

⁴In a fully connected neural network, all neurons in a layer are connected to all neurons in the next layer.

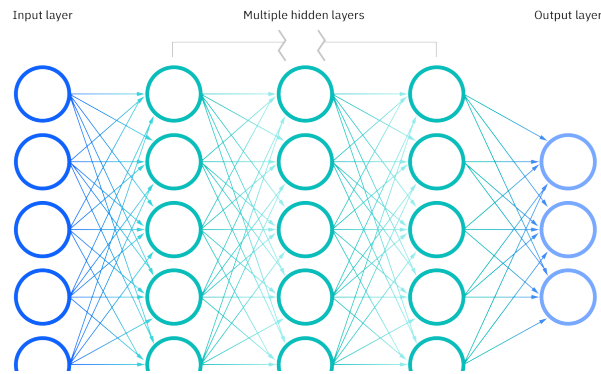


FIGURE 5.3: Schematic illustration of a fully connected deep neural network. Kindly borrowed from [63].

So essentially a neural network takes a number of inputs and uses clever weights and biases to turn them into outputs or predictions of certain quantities. As such, it is basically fitting a complex⁵ non-linear function in the high-dimensional input parameter space.

Training a neural network requires splitting the dataset into sets, specifying a loss function and using a method called stochastic gradient descent to adapt the weights and biases, all of which will be explained in section 5.3. Before that we have to understand the specific type of neural network that lies at the foundation of this work: Graph Neural Networks.

5.2.3 Graph Neural Networks

Graph Neural Networks (GNNs), invented in 2007, are a quite recent contribution to machine learning [64]. As hinted at by the name, GNNs take graphs as input. Graphs are collections of nodes that are connected by edges, as illustrated in figure 5.4. Information can reside in the nodes, in the edges or in the overall graph [64].

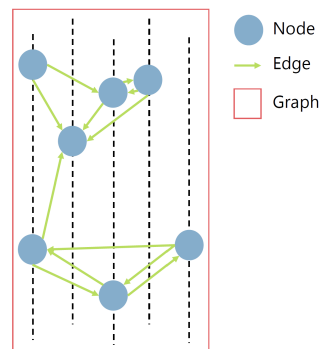


FIGURE 5.4: Illustration of a graph with nodes and edges. Each node in the graph is connected to its two nearest neighbours by one-directional edges. If one imagines the nodes as DOMs and the dashed lines as strings, it could represent a small event in the IceCube detector.

For instance, in social media, the nodes would be users (all their private information resides in the node), the edges would be their connections (with information on

⁵The complexity depends on the number of weights and biases, also called learnable parameters.

how they are connected, directly or through others) and overall network information could e.g be the total number of users in the network.

The purpose of GNNs can be to predict something about the overall graph, in the case of IceCube, which event took place in the detector. Or it could be to predict on the node level, for instance which of the DOM hits were actual information and which were noise triggers. To arrive at the predictions, the input graph are processed through what is called convolutions, in which the graph is updated. Convolutions are similar to those of a CNN, but more general in nature. They work on each node individually and takes its own information as well as that of the nodes it is connected to. This information is then passed through some form of multi-layer perceptron (MLP), which outputs the updated node in the new graph. This node could contain a larger amount of information variables than it had in the input graph.

Once the graph has undergone a specific number of convolutions, a pooling scheme is used to aggregate the information across all nodes, ensuring the dimensionality is similar regardless of the size of the input graph. This aggregated information is used as inputs in a final MLP, which then outputs the predictions. As such a GNN can be trained on / applied to data containing a variable number of nodes. A particular example of a GNN model, the one used in this work, is introduced in section 5.4. But first an understanding of how to train a machine learning model is required.

5.3 How To Train Your Machine Learning Model

This sections explains some of the details in the actual training of machine learning models, including how to avoid overtraining and optimising learnable parameters and hyperparameters.

5.3.1 Loss Function

Having set up an entire machine learning model, a good way to evaluate its performance is required. To do so, one applies a loss function to quantify the difference between the model predictions and the truth values. There are a variety of loss function optimised for particular types of data and models. Specific examples used in GraphNeT will be explained in section 5.4.3.

5.3.2 Stochastic Gradient Descent, Learning Rates & Backpropagation

Equipped with the machine learning model and a way to evaluate how well it performs (the loss function), we are ready to explore the process in which one optimises said performance. This section will not include many equations, since the methods are mostly intuitive and including the math becomes more confusing than illuminating.

To minimize the overall loss, an optimizer is used to adapt the learnable parameters (weights and biases) of the model, until its predictions are as close to the truth as possible. Most optimizers, including the one used in this work, builds upon the simple concept of gradient descent. The model is essentially build of functions upon functions, which means that there exists a gradient,⁶ which if it can be approximated, can be used as a guideline to change the learnable parameters [65].

⁶A direction in the high-dimensional space of the learnable parameters in which the overall loss increases the most. Moving in the opposite direction will then decrease the total loss.

This naturally leads to the question of how large a step to take once a desired direction is determined. The answer lies in the learning rate, which is a hyper parameter⁷ that can be adjusted. The learning rate essentially determines how much along the opposite direction of the approximated gradient we move. As illustrated in figure 5.5, too large a learning rate will prevent the model from converging in a minimum, while too small a learning rate will substantially increase the time it takes to reach said minimum. A common approach is to use a learning rate scheduler, which adapts the learning rate throughout the training. For instance, in the beginning, one might want to explore the parameters space, before settling into a local minimum. This requires having a large learning rate initially, but decreasing it in some fashion, to allow it to actually get stuck and descent into a specific minimum[65]. Learning rate schedulers can be fashioned in a multitude of ways, but the standard GraphNeT one is explained in section 5.4.4.

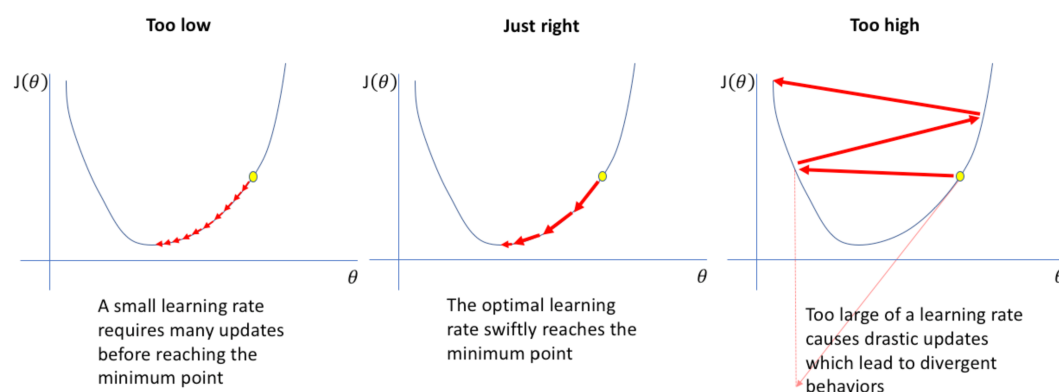


FIGURE 5.5: Illustration of the problems with having too large or small a learning rate. It is of course greatly simplified in two dimensions and with a single minimum. The actual loss landscape is high-dimensional and contains a multitude of local minima. Kindly borrowed from [66].

Understanding the optimizer used in this work, ADAM[67], requires understanding two additional concepts. First, stochasticity is added by approximating the gradient using only parts of the data (a batch) in each step. This is smart for two reasons. It is a lot faster to compute the gradient using batches of say 512 data points than the entire training data. Furthermore, it avoids the problem with simply ending in the first best local minimum, which is what the classical gradient descent method does. The parameter spaces are incredibly complex, which means that one might need to explore wide volumes before converging. During training, a single batch of training data is used at a time, until all data has been through the model. At that point one training epoch is said to be finished. The number of training epochs is another hyperparameter, which is closely related to the concept of early stopping, which is introduced in section 5.3.3.

The second thing ADAM adds is "momentum". This simply means that the gradient is not calculated from scratch each time, but the one from the prior step is adjusted towards the newly calculated one. This means that the gradient is a moving average, which avoids too drastic changes. Finally, ADAM also approximates the second

⁷A variable that influences the training and performance of a machine learning model, but is not itself adjusted by the optimizer in the training process.

order gradient, which allows it to use different step sizes for the different learnable parameters. This is efficient, since the loss space might be flat in one direction, but changing drastically in another[65].

The gradients required in the stochastic gradient descent are calculated using a method called backpropagation. It relies on the chain rule of differentiation to carry out the derivative estimations for each learnable parameter with respect to the loss function. In essence one first applies the neural network to the input parameters, after which one propagates the result backwards in the model, and uses the result to get the gradients[65]. The specifics of back propagation is beyond this thesis.⁸

Knowing how to train a model, lets see when to stop the training.

5.3.3 Train, Validation, Test Split

It is not difficult to imagine that if one uses a large machine learning model, with enough learnable parameters, one can perfectly predict the data that is being used to train the model. In some ways it is similar to the fact that all functions in one dimension can be fit with a polynomial of large enough order. Just as with fitting a function, one wants the model to learn as much as possible from the data, while still being general enough to perform similarly on data the model has never seen before.

The way to avoid this issue, is to split up your data set in three parts. A training set, which is used for training. A validation set, which is used to decide when to stop training. And finally a test set, which the model has never seen before to test the final performance. As illustrated in figure 5.6, during the training epochs⁹ the training and validation loss is monitored. Once the validation loss stops decreasing and starts increasing, the model has reached a point where further adapting the weights, results in a worse performance on data which is not a part of the training set. In essence it is learning to predict the exact data, more than the general trends. Thus one uses what is called early stopping and simply quit training and use the model as it was, when the validation loss was at its minimum.

Since the validation loss is in essence what we minimize in training, a test set is used to gauge how the model performs, once we use it on unknown data.

⁸More information about back propagation, including equations and implementation can be found in [65], section IX C.

⁹An epoch is one entire passing of the training data during the training procedure.

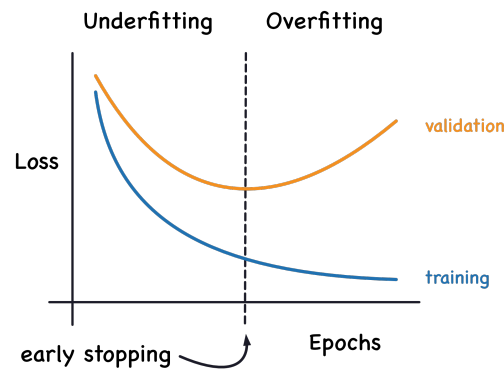


FIGURE 5.6: Illustration of the idea of early stopping. At some point during the training of a neural network the loss of the validation set stops decreasing and even increases. At this point the training should be stopped. Kindly borrowed from [68].

5.3.4 Hyperparameter Optimisation

As mentioned in a previous footnote, hyperparameters are not learnable parameters. They include things such as the learning rate, number of neurons per layer, which loss function to use etc. If the machine learning model is fast to train, one can do a grid search, where all combinations of hyperparameters are compared against each other, in terms of some objective performance evaluation. However, this is often not possible, since models can take days to train and there are easily thousands of reasonable hyperparameter combinations.

More sophisticated approaches exist, but are often difficult to implement on large machine learning models. The content of chapter 6 is essentially a hyperparameter optimisation of DynEdge in regards to reconstructing a specific type of event. In this case we start with a model that already performs well on low energy, and simply adjust a single hyperparameter at a time, to see if we can improve its performance on high energy northern track events. But before we jump to the results, let's have a look at the specific GNN used in this work.

5.4 GraphNet & DynEdge

IceCube data with its irregular geometry and variable number of hits per event is perfectly suited for GNNs in theory. Recently it was shown that it works in practice as well. A GNN was applied to classifying and reconstructing IceCube events in the energy range 1-100 GeV, where it beat the best-in-class algorithm that is currently being used. In particular it improved the signal efficiency by 18% for classification at a fixed false positive rate (FFR) and the resolution of energy and angular reconstruction by 13-20% for energies of 1-30 GeV [1]. These performance measures will be explained further in a later section.

If one wants to apply GNNs to IceCube data, it is difficult to find a better framework than GraphNeT (Graph neural networks for Neutrino telescope event reconstruction) [69] [70]. It is an open-source git-based python repository which aims at becoming the standard out of the box method for using GNNs to reconstruct and classify events in neutrino telescopes. It allows for an easy setup of complex GNN machine learning models and easy implementation of additional functionality. It is

also optimised for speed, reconstructing and classifying events orders of magnitudes faster than classical methods [69]. All machine learning in this thesis is based on the GraphNet framework and builds upon the work of previous master students at NBI.

5.4.1 Benchmark Model Architecture

The benchmark model architecture, which is primarily used in this thesis is called DynEdge and was developed by a former master student at NBI, Rasmus F. Ørsøe, and applied in [1]. The model architecture is illustrated in figure 5.7.

DynEdge takes a graph of hits in the IceCube detector as input. Each hit represents a node and contains all of the variables in table 4.1 as information. Each node is then connected by a one-sided edge to its k^{10} nearest neighbours, calculated using the euclidean distance in the first three variables (This represents actual spacial distance in the input graph, but becomes more abstract for subsequent graphs). The graph then undergoes four iterations of the EdgeConv operator [71].

$$\bar{x}_j = \sum_i^k mlp(x_j, x_j - x_i)$$

Each node is updated in the following way. For each of its k neighbours, the information of the node itself, along with the difference in information between it and its neighbours are fed to a fully trainable MLP. The MLP contains a single hidden layer, and it is identical for all nodes in a single instance of the EdgeConv. It outputs a new information vector with a dimension of 256. The eight vectors, one for each neighbour are then added and this new vector constitutes the node in the updated graph. Once all nodes have been updated, a new graph is created by connecting the updated nodes to their 8 nearest neighbours. Since the MLP are free to combine the variables as it sees fit, the nearest no longer represent only spacial information, but a more abstract combination of variables.

The four instances of the EdgeConv each outputs a new graph. Each of the graphs, the first 6-dimensional and the subsequent four 256-dimensional are concatenated, such that each node now contains 1030 variables. These are now run through another MLP. Finally, node aggregation is used to get rid of the number of nodes, by only using the min, max, mean and sum of each of the variables that each node has. This means that it can now be fed to a final MLP, which can be standardised to accept the resulting 1030 variables and five additional global statistic variables and use them to output the actual predictions.

The global statistics are the homophily rate for $DOM_{(x,y,z,t)}$ and the total number of nodes [1]. The homophily rate is calculated as follows[72]:

$$\beta = \frac{1}{|V|} \sum_{v \in V} \frac{\text{Number of } v\text{'s neighbors who have the same label as } v}{\text{Number of } v\text{'s neighbors}}$$

In our case, for instance the homophily rate of DOM_x runs over all nodes and asks how many of its k nearest neighbours have the same DOM_x value as itself. So lets say we use eight neighbours and all the DOMs that are hit, are hit eight times. Then each term in the sum would be 1, meaning β would be 1 in total. On the other hand, if each DOM is only hit once, and each of the DOMs which are hit is located at

¹⁰ $k=8$ is used in all applications in this work.

different x positions, then each term is simply zero (no neighbours are located at the same Dom_x value), and so would β be. The homophily rate is a measure of how locally connected versus interconnected the nodes are [72].

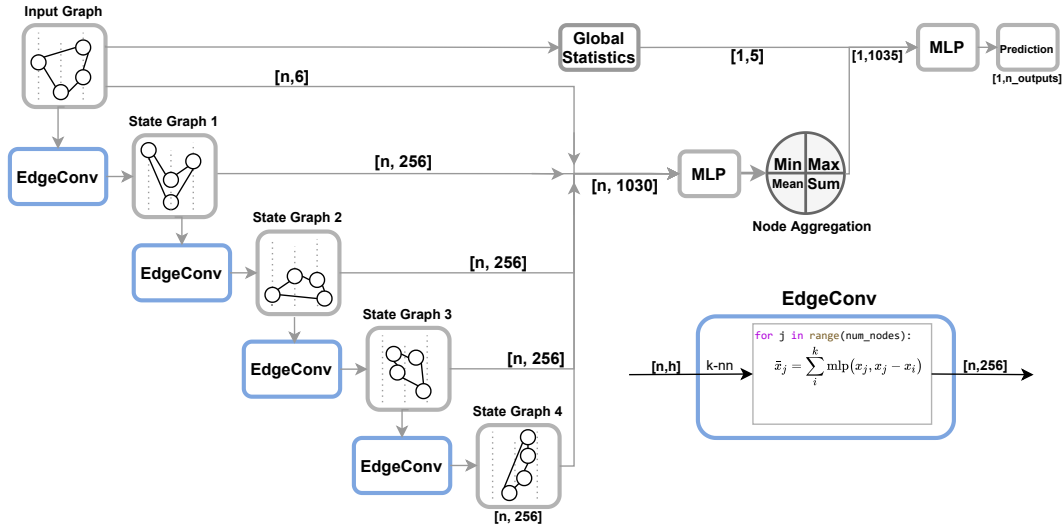


FIGURE 5.7: Overview of the DYNEDGE architecture taken from [1]. The model is based on the EdgeConv [71] convolution, classical Multilayer Perceptrons as well as node aggregation. See text for more information.

Understanding the architecture, lets turn to the preprocessing of the input data.

5.4.2 Benchmark Preprocessing

The standard preprocessing of the model is as follows

$$\begin{aligned} \text{dom}_x^{\text{scaled}} &= \frac{\text{dom}_x}{100} \\ \text{dom}_y^{\text{scaled}} &= \frac{\text{dom}_y}{100} \\ \text{dom}_z^{\text{scaled}} &= \frac{\text{dom}_z + 350m}{100} \\ \text{dom}_{\text{time}}^{\text{scaled}} &= \left(\frac{\text{dom}_{\text{time}}}{1.05 * 10^4} - 1 \right) * 20 \\ \text{charge}^{\text{scaled}} &= \text{charge} \\ \text{rde}^{\text{scaled}} &= \frac{\text{rde} - 1.25}{0.25} \\ \text{pmt}^{\text{scaled}} &= \frac{\text{pmt}}{0.05} \end{aligned}$$

which is supposed to center the variables around 0 and constrain them in size. Ideally the preprocessed variables should each have the distribution of a standard Gaussian with a mean of zero and a standard deviation of 1[73]. This is one of the attempted improvements in section 6.5, where the distributions prior and after

the preprocessing can also be viewed. Another important factor to consider is the benchmark loss functions.

5.4.3 Benchmark Loss Functions

The loss function used in the benchmark model depends on the purpose of the GNN.

Angular Reconstruction - Von Mises-Fisher Loss

For a single angular reconstruction (either azimuth or zenith), the 2D Von Mises-Fisher loss function was found to lead to the best angular resolution by former NBI student Rasmus F. Ørsøe [1].

The Von Mises-Fisher loss is "considered the directional equivalent of Gaussian distribution[s]" [74]. The loss function accepts two vectors of the same dimension and a concentration parameter κ , which is an estimate of the inverse variance.

In the 2D case, take zenith (ϕ) for example, the target vector μ is constructed as $\mu = (\cos(\phi), \sin(\phi))$. DYNEDGE predicts the estimated angle ϕ_{est} and the estimated inverse covariance $\kappa_{est} = \frac{1}{\sigma_{est}^2}$. Using the predicted angle, an estimated vector is constructed as $\mu_{est} = (\cos(\phi_{est}), \sin(\phi_{est}))$.

The Von Mises-Fisher distribution then yields the probability as:

$$p(\mu, \mu_{est}, \kappa_{est}) = C_m(\kappa) e^{\kappa \mu * \mu_{est}}$$

Where $C_m(\kappa)$ is a normalization term that relies on a modified Bessel function of first order. Since μ and μ_{est} are unit vectors, the dot product in the exponential can be rewritten: $p(\mu, \mu_{est}, \kappa_{est}) = C_m(\kappa) e^{\kappa \cos(\Delta\phi)}$, with $\Delta\phi = \phi_{est} - \phi$. The negative logarithm is taken to create a loss function that can be minimized. In the end the loss function looks like this:

$$NLLvMF(\mu, \mu_{est}, \kappa) = -\log(C_m(\kappa)) - \kappa \cos \Delta\phi$$

Now to reconstruct the actual direction of IceCube events, the zenith and azimuth angles are reconstructed individually along with their uncertainty estimates. While it is possible to directly estimate both angles using a 3D Von Mises-Fisher loss, it is generally found to be less accurate.

Energy Regression - Log-Cosh Loss

The loss function that is used to train the model that carries out the energy regression is the log-cosh [1]:

$$loss_{logcosh} = \log(\cosh(R_E))$$

Where R_E is the residual of the energy prediction defined as $R_E = \log(E_{pred}/GeV) - \log(E_{true}/GeV)$ [1]. The LogCosh loss has been used a lot in regression problems to smooth out the loss function around zero [75], compared to other classical loss functions, such as the mean squared error. Furthermore, it does not diverge as fast when the energy residual grows [1].

Classification And Multiclassification - Cross Entropy Loss

For binary classification, the traditional Binary Cross Entropy loss is used[1]. It looks as follows:

$$loss_{BCE} = -(p(x) * y + (1 - p(x)) * (1 - y))$$

Where $p(x)$ is the predicted probability of being the type 1, and y is the label, which is either 1 or 0. As such, if we correctly predict the class, we get a contribution of -1 to the overall loss. Either from the first term if $p(x)$ and y are one, or from the second term if $p(x)$ and y are zero. On the other hand, if $p(x)$ and y are 0 and 1 or vice versa, then the contribution to the loss is 0. Since we are trying to minimize the loss function, the model tries to correctly predict the probabilities.

For multiclassification, the Cross Entropy Loss is used. It is a generalisation of the binary case that looks like this:

$$loss_{CE} = - \sum_{i=1}^n p_i(x) * y_i$$

Where n are the number of classes we are trying to predict, p_i is the probability of class i as predicted by the GNN and y_i is the label, which is one if the event is of the i 'th type and zero otherwise.

Interaction Vertex Position - Euclidean Loss

For the interaction vertex position, the Euclidian distance loss function is used. It is the classical length between the predicted $(\hat{x}, \hat{y}, \hat{z})$ and actual (x, y, z) point of interaction:

$$loss_{Euclid} = \sqrt{(\hat{x} - x)^2 + (\hat{y} - y)^2 + (\hat{z} - z)^2}$$

5.4.4 Benchmark Learning Rate Scheduler

The learning rate scheduler used in the benchmark GraphNeT model, is a piecewise linear scheduler. If a learning rate of $LR = 10^{-4}$ is specified, it ramps linearly up from $0.01 * LR$ to the LR over the first half epoch. It then linearly declines from the LR to $0.01 * LR$ from the first half epoch, until the number of epochs specified in the training. As such, it is a slightly arbitrary learning rate scheduler, since it depends on the number of epochs one wishes to train. It is possible that other schedulers, such as having the learning rate decline exponentially, or decreasing it a step each time the validation loss reaches a plateau, would give better results. One attempted change in chapter 6, is to simply have a constant learning rate, instead of changing it throughout the training.

Equipped with a fundamental understanding of machine learning, it is time to investigate the results obtained by its application in this work.

Chapter 6

Improving the Reconstruction of High Energy Northern Tracks

Observing the universe using neutrinos and contributing to multi messenger astronomy is a main objective for IceCube¹. High energy northern (through the earth) track neutrinos have the lowest pointing uncertainty and therefore play an important role in identifying sources of cosmic neutrinos.

Improving the angular reconstruction for this type of events in the IceCube detector would greatly enhance its ability to function as a neutrino telescope. The pointing accuracy improves with the square of the angular resolution (precision) and small improvements allow us to observe weaker signals and astrophysical events that are beyond the current frontier.

The baseline DynEdge model² was optimised to perform on low energy neutrinos below 10^4 GeV. The developer of the DynEdge model, Rasmus F. Ørsøe orchestrated a project to try to tune DynEdge to a dataset containing Northern tracks events with energies in the range of 10^2 GeV - 10^9 GeV. He came up with suggested adaptations of the model and its hyperparameters. The suggestions were then carried out by the author and Andreas Mosgaard Jørgensen, another Master student at NBI. Each of us worked on different improvement attempts, meaning that the implementations presented in this chapter are entirely my own work. However, the ideas mainly stem from Rasmus F. Ørsøe, as does the original code.

The DynEdge performance is compared to the high energy reconstruction algorithm, SplineMPE, which was introduced in section 4.7.2. So far, none of the optimisation attempts have improved the performance of DynEdge significantly. As such, DynEdge is not yet able to perform as well as SplineMPE. However, documenting the attempted changes is still important for future development of the DynEdge model.

6.1 General Method

To benchmark the improvements, only a single aspect of the baseline model is changed at a time. While this could be what prevents a significantly improved performance, it makes it easy to see if any single feature has an impact. The training, validation and test samples are exactly identical for each attempt, as are the number of epochs

¹See section 4.1.2.

²See section 5.4.

and any hyperparameters that are not the focus of the specific attempt. The only exception is that if a training did not converge during the first training, it was retrained with a larger number of epochs. The cases in which this happened, is indicated in the label with a star in the following figures.

The performance indicator used in the project, is the angular resolution, which is calculated as:

$$w = \frac{p84(R) - p16(R)}{2} \quad , \quad R = \text{angle}_{\text{true}} - \text{angle}_{\text{predicted}}$$

Where p84 and p16 are the 84th and 16th percentile. The resolution corresponds to a standard deviation for a gaussian distribution, but it is a measure which is robust to outliers. The resolution is illustrated in the case of the Benchmark DynEdge and SplineMPE zenith residuals in figure 6.1³.

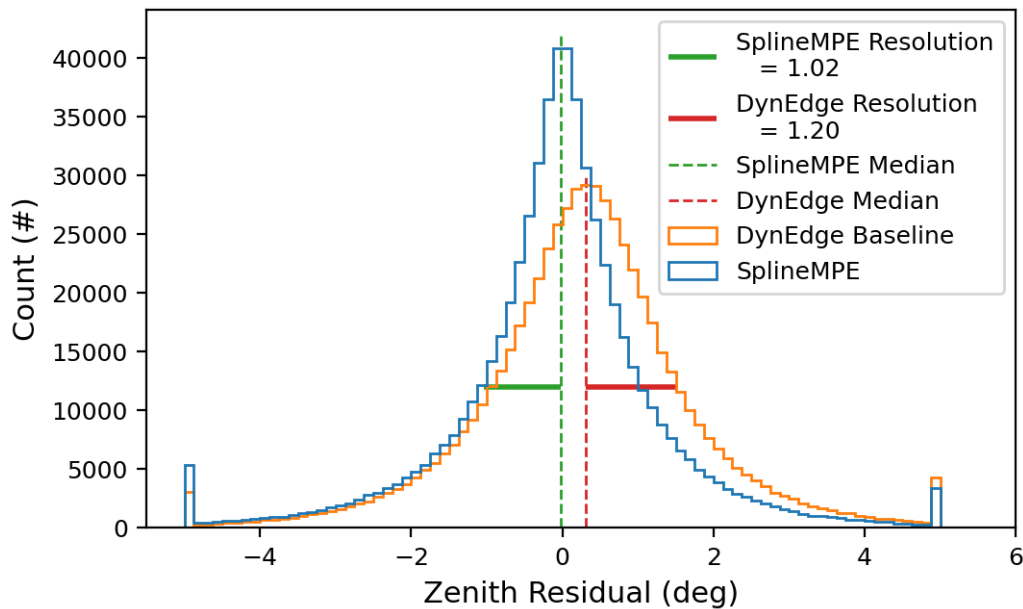


FIGURE 6.1: Distribution of zenith residuals for the baseline DynEdge model and SplineMPE. Overlaid are calculated resolutions illustrated from the medians of the distributions. The resolutions are robust measures of how well the model predicts the zenith angle. If they had a Gaussian shape, the resolution would correspond to a standard distribution. Outlier bins are used in the figure.

This figure shows that SplineMPE performs better than the DynEdge baseline overall, with a resolution of 1.02 compared to 1.2. SplineMPE's lower resolution results from the more narrow distribution and it's median is closer to zero than the median of DynEdge. However, this is not the entire story, since the pointing accuracy depends strongly on energy.

The angular resolution is therefore calculated as a function of energy. For each energy bin, a bootstrap method is used to obtain a measure of the uncertainty. The events in a given bin are sampled with replacement a hundred times. Each sample

³See Appendix A for a similar plot for the azimuth residuals.

has a slightly different angular resolution. The standard deviation of these samples is a measure of the uncertainty of the angular resolution, which if the test set is representative of the underlying population, should correspond to the actual standard deviation.

A relative improvement to the baseline model is also calculated as follows:

$$\text{relative improvement} = \left(1 - \frac{\text{resolution}_{\text{modified}}}{\text{resolution}_{\text{baseline}}} \right) * 100$$

The method is illustrated in figure 6.2. It is clear that SplineMPE not only performs better than the DynEdge baseline overall, but does so at all energies. At the same time, it is clear that DynEdge's performance lacks especially for energies above 10^3 GeV, where the relative improvement in zenith resolution from DynEdge to SplineMPE oscillates around 40%. This is a strong motivation for trying to improve the performance of DynEdge at high energies.

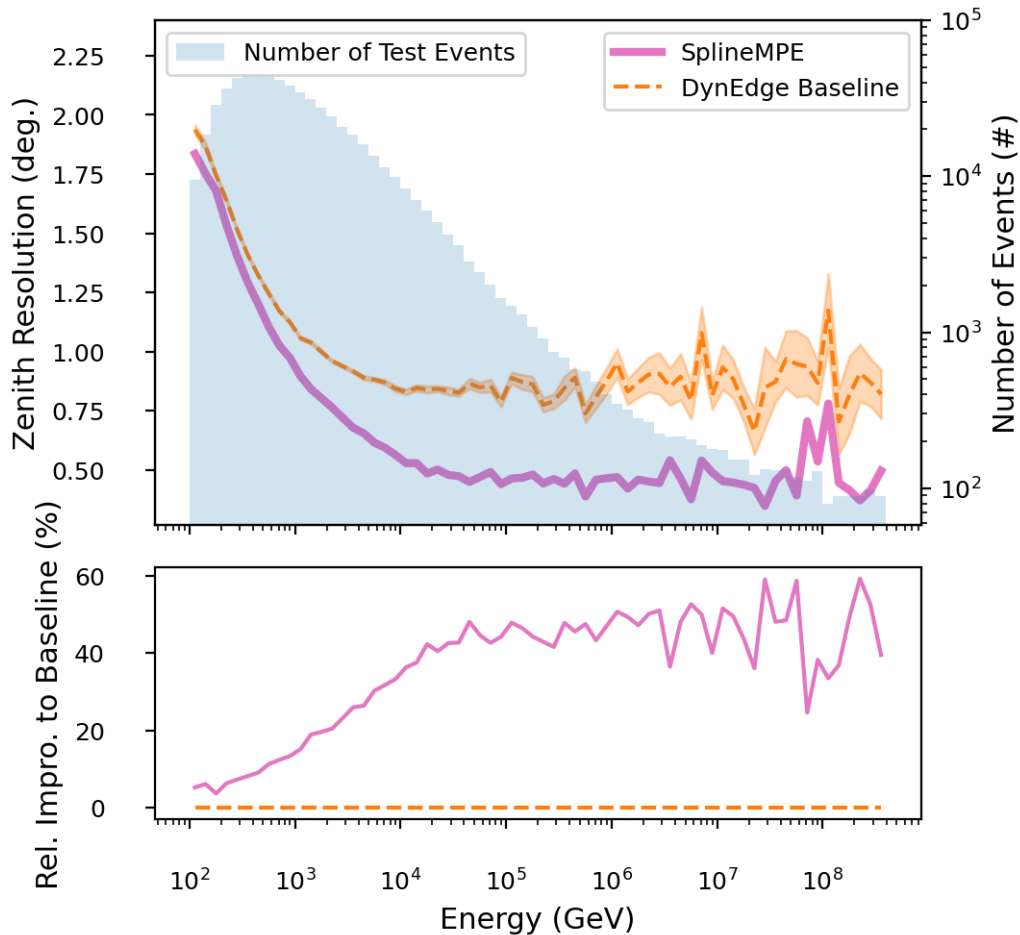


FIGURE 6.2: (top) Zenith resolution with bootstrap std as a function of energy for the baseline DynEdge model and SplineMPE. (bottom) Relative improvement of SplineMPE to Dynedge baseline.

The performance is benchmarked individually for zenith and azimuth. The zenith results are presented below, but given that the azimuth results are quite similar, they

can be found in Appendix A. In the following plots, the relative improvement of SplineMPE compared to the DynEdge baseline is not included, since it makes it difficult to see the slight variation in performance of the attempted improvements of DynEdge.

6.2 Learning Rate Adjustments

The learning rate in the baseline model is based on a step-wise linear learning rate scheduler⁴. To test if this was the hindering factor, three different constant learning rates were tested: 10^{-3} , 10^{-4} , and 10^{-5} . However, as can be seen in figure 6.3, the models with fixed learning rates have a worse (higher) resolution than the baseline. As such, a fixed learning rate does not improve the angular reconstruction for high energy neutrinos. It is possible that improvements could result from changing the learning rate scheduler to a more sophisticated version, such as exponential decay or one which drops the learning rate as the validation loss plateaus. This is something that should be tested in future attempts to optimize DynEdge to high energy neutrinos.

⁴See section 5.4.4.

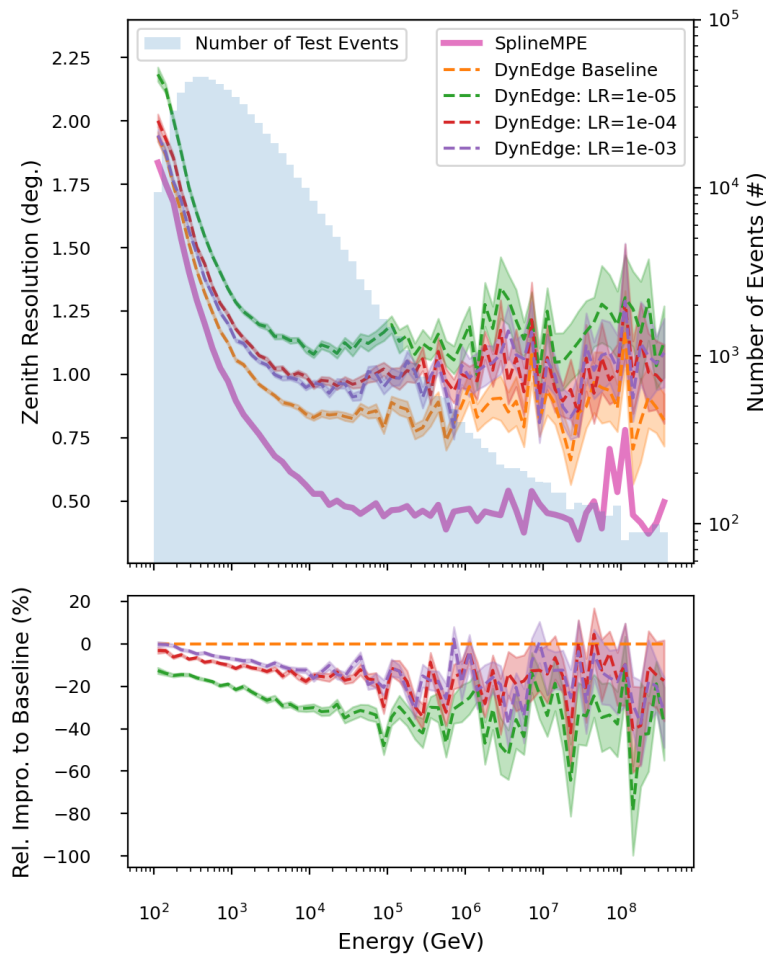


FIGURE 6.3: (top) Zenith resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model with fixed learning rates. (bottom) relative improvement of the adjusted DynEdge models to the baseline.

6.3 Changing The Pulsemap

There are a number of pulsemaps in the Northern Track I3 files, where different degrees and methods of cleaning is applied. The TWSRTHCInIcePulsesIC is used in the baseline and is the cleanest available. However, DynEdge has previously been shown to disregard noise to a large extent⁵. Therefore, a model was trained on each available pulsemap, to see if this would improve the angular resolution. The following pulsemaps were tested:

- InIceDSTPulses
- HWInIcePulses
- SRTHVInIcePulses
- InIcePulses

⁵In the whole of Chapter 7, the models are trained on an uncleaned pulsemaps, since it gave better performance than a Seeded-Radius-time cleaned one for multiclassification.

As can be seen in figure 6.4, some of the pulsemaps actually seem to consistently improve on the performance at lower energies, especially HWInIcePulses is quite good. However, at higher energies, the uncertainty becomes too large to say anything with confidence, and the slight improvement is far from good enough to reach the level of SplineMPE. However, for future testing, it would make sense to test the HWInIcePulses pulsemap along with other adaptations of DynEdge.

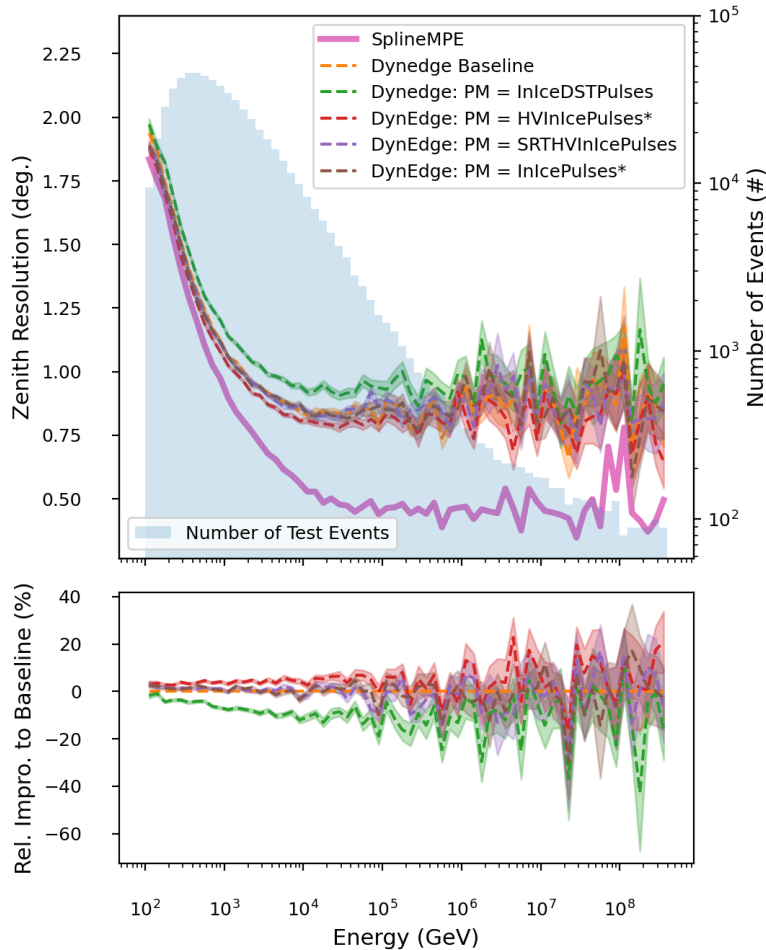


FIGURE 6.4: (top) Zenith resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on different pulsemaps. The * in the labels indicates extended training. (bottom) Relative improvement of the adjusted DynEdge models to the baseline.

6.4 Focus On Highest Energy Range

DynEdge was optimised on events with energies below 10^4 GeV. This is evident in the figures above, where the resolution is much closer to that of SplineMPE for energies below 10^3 GeV, but becomes worse as the energy increases. This could be because there are far more events with energies below 10^4 GeV than above in the training set, as can be seen in figure 6.4. This might result in a model that focus mostly on the low energy range, thus not learning enough about high energy events.

To check if this was the case, a model was trained only on events with energies above 10^4 GeV. The results are visible in figure 6.5. Unfortunately excluding the majority of events from the training does not lead to a better performance for high energies. On the contrary, it results in a much worse resolution, which is probably just a result of lower statistics. This also suggest that for angular reconstruction, DynEdge is able to generalize across energies, learning trends at low energies, which are also applicable for high energies. Unfortunately, this means that restricting the energy range is not the right method to beat SplineMPE. However, in a future attempt, it would be interesting to simply weigh the high energy events more in the loss function, instead of completely disregarding the low energy events.

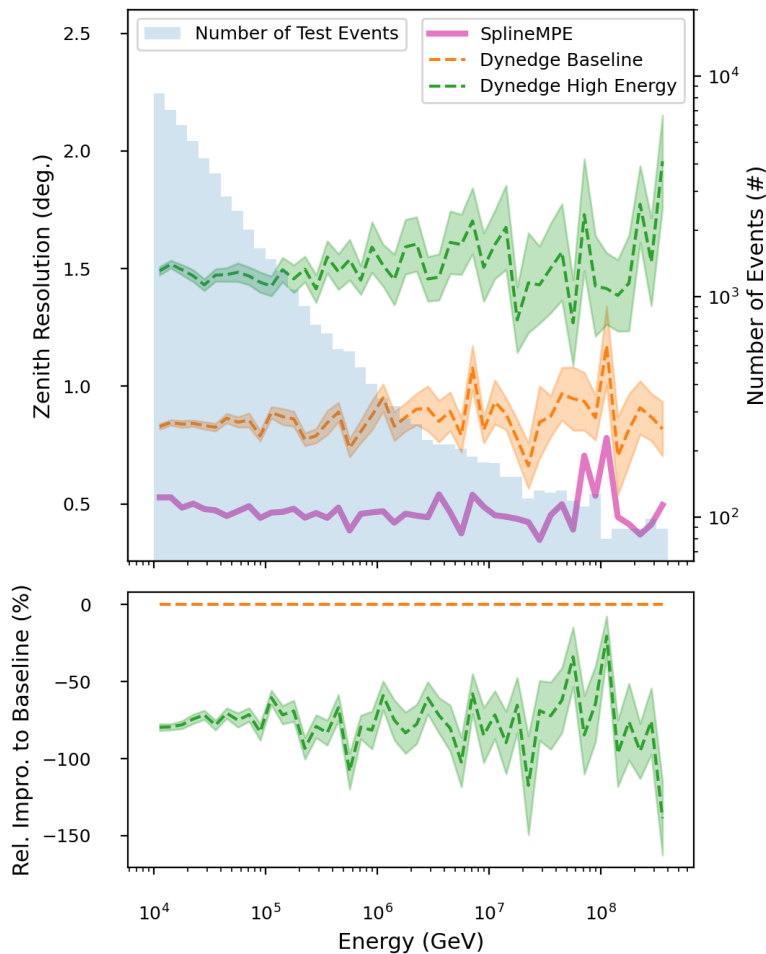


FIGURE 6.5: (top) Zenith resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on neutrinos with energy above 10^4 GeV. (bottom) Relative improvement of the adjusted DynEdge models to the baseline. Note that the energy range is limited to above 10^4 GeV, since the adjusted model has no predictions for lower energies.

6.5 Gaussian Prescaling Of Input

The stochastic gradient descent method of optimising learnable parameters works best if the distributions of input parameters are smoothly distributed around 0, without too wide tails[73]. The benchmark DynEdge prescaling was explained in section

5.4.2, and can be seen in figure 6.6. In the same figure, are distributions of input parameters using a Gaussian prescaling, which is an attempt at improving the angular resolution of DynEdge.

The Gaussian scaling is carried out with the sklearn quantile transformer[76] that was fitted using 50.000 events from the training set. The gaussian scaling succeeds in smoothing out the distributions of the parameters as well as centering them around 0. The standard DynEdge scaling mainly reduces the range of the parameters and only in some cases center them around 0. The outlier bins in the gaussian scaled histograms of dom_x and dom_y has not been understood, but is probably due to some quirk in the scaling algorithm.

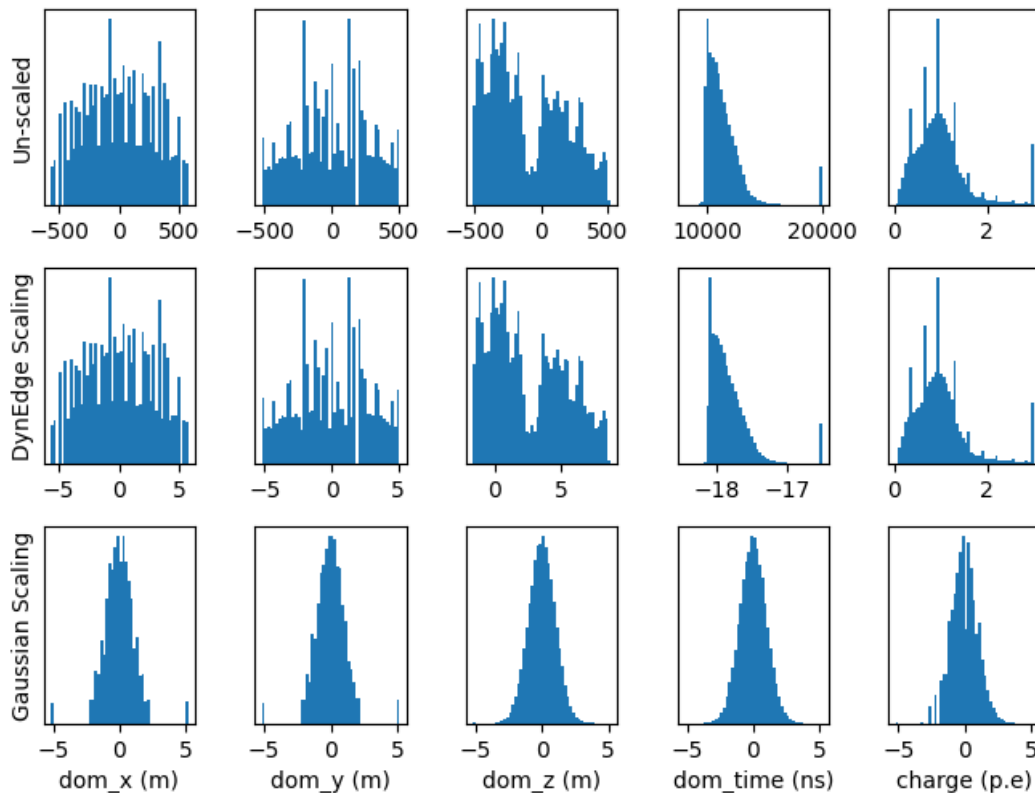


FIGURE 6.6: Histograms of the five input parameters for which the scaling is changed. Un-scaled histogram in the top row. Standard DynEdge scaling in the middle row. Gaussian scaling using a quantile transformer with a standard distribution output in the bottom. 50.000 events with a total of 2.946.429 pulses from the training sample was used to fit the transformer.

Different combinations of Gaussian prescaling was attempted. Including all 5 variables in figure 6.6 or excluding either the three positions (dom_x, dom_y & dom_z) or the time and charge. In the case where some variables are not transformed using Gaussian scaling, they are transformed using the benchmark DynEdge method.

The results are available in figure 6.7 and unfortunately shows that there is no significant improvement to be gained from a Gaussian prescaling of all or some of the

input variables. However, it does seem as if the Gaussian scaling of only the position variables result in a slight improvement, which could be tested along with other adaptations of the DynEdge baseline.

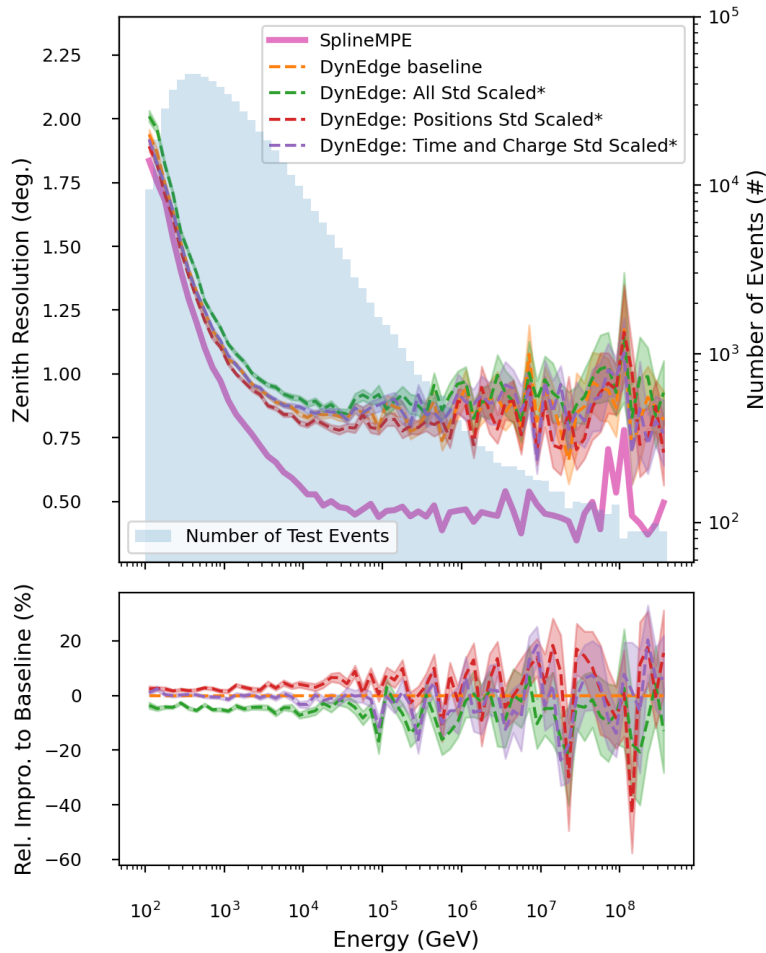


FIGURE 6.7: (top) Zenith resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on gaussian scaled features. "All std Scaled" means that all five input features from figure 6.6 are scaled using new transformer. "Positions Std Scaled" means only dom positions are scaled using new transformer. "Time and Charge Std Scaled" means only dom time and charge are scaled using new transformer. The * in the labels indicates extended training. (bottom) Relative improvement of the adjusted DynEdge models to the baseline.

6.6 Going Forward

SplineMPE turned out to be a hard nut to crack. Despite a number of attempts, DynEdge was not able to produce competitive angular resolutions. However, there are many adaptations that can still be tested. So far in the project, only a single parameter or aspect of DynEdge was changed at a time. While this makes sense to begin with, it is possible that performance can be improved by simultaneously changing several aspects of the model. Furthermore, more complex changes might

also be needed. For instance another message passing scheme in the graph convolutions, a new learning rate scheduler, a different type of optimiser or perhaps simply a much larger training sample. The current effort to implement the winning model from the recent Kaggle IceCube competition is also very interesting. The competition showed that transformers in combination with GNNs are very effective[4].

Thus concludes the first result chapter with less than optimal results. Luckily we now turn to the second result chapter, which should brighten the mood a bit.

Chapter 7

Classification And Evaluation Of Neutrino Sample In Real Data

7.1 Motivation And Outline

Essential to most of the research IceCube is engaged with, is the reliable and efficient classification of neutrino events in data. GraphNet has been shown to outperform the classical methods of classification and reconstruction in low energy (1-100 GeV) Monte Carlo data, while speeding the process up by orders of magnitude [1].

It is easy to test the performance in simulated data, whereas it is a slightly more difficult task to show that the results carry over to actual data from the detector. Yet that is exactly the goal of this chapter. The proposed method is to compare GNN neutrino selections in simulated and actual data. If their distributions in reconstructed and calculated variables are similar, it indicates that the neutrino selection in data consists mainly of neutrinos.

To determine if GraphNet, and particularly the DynEdge model, can classify and reconstruct a clean neutrino sample in data, the following sections will be covered in this chapter:

- The foundation of this work is presented and those who have part in the results are credited.
- The data selection and Monte Carlo train/val/test splits for each model are explained.
- The classification and reconstruction results in Monte Carlo data are presented and discussed, with and without a comparison to the OscNext reconstruction algorithm Retro.
- The necessary data cleaning to ensure Monte Carlo - real data agreement is presented.
- A selection of clean track and cascade neutrino samples in Monte Carlo and real data is selected using a multiclassification and track/cascade classifier. The distributions of the neutrino selections are compared across reconstructed and calculated variables to ensure that the neutrinos in data resemble those in Monte Carlo.
- The rate/amount of events we get in the clean Monte Carlo neutrino sample is compared to the OscNext selection process.

- The quality of the neutrinos in the GraphNeT selection are compared to those in the OscNext selection.
- Examples of detector signatures from the actual neutrinos in real data are presented along with the sneaky muons that make it into the Monte Carlo sample.

7.2 Foundation Of The Results Presented In This Chapter

The results in this chapter is a direct extension of work Morten Holm and I did together, which was presented in his master thesis. Therefore the methods and ideas should be as much credited to him, and naturally Troels Christian Petersen who supervised us both. The GNN models that are used in this section were trained by Morten and I together, except for the multiclassification and interaction vertex reconstruction models. The results presented in this chapter expands and details many of our initial results. The analysis presented is thus my own work, but built upon a foundation Morten Holm and I established together with Troels Christian Petersen and the entire GraphNeT team.

7.3 Data Selection And Train, Validation, Test Split

7.3.1 Data Selection And New Muon Sample

The data used in this chapter comes from the OscNext workgroup in IceCube. The simulated noise, muons and neutrinos are from the simulations vuvuzela, muongun and Genie respectively¹. The real data is approx. 1% of the OscNext burnsample, 62 million events. More precisely, all subruns² ending on 00 from 2011-2021 are used. The exact data samples and locations are available in table 7.1.

Sample Nr	Particle Type	Data Type	Cobalt Location	Amount
120000	Electron neutrinos	Simulation: Genie	/data/ana/LE/oscNext/pass2/genie/level3_v02.00	8,301,908
140000	Muon neutrinos	Simulation: Genie	/data/ana/LE/oscNext/pass2/genie/level3_v02.00	20,106,952
160000	Tau neutrinos	Simulation: Genie	/data/ana/LE/oscNext/pass2/genie/level3_v02.00	8,886,081
130000	Muons	Simulation: MuonGun	/data/ana/LE/oscNext/pass2/muongun/level3_v02.00	15,990,173
139008	Muons	Simulation: MuonGun	/data/ana/LE/oscNext/pass2/muongun/level3_v02.00	719,737
888003	Noise	Simulation: Vuvuzela	/data/ana/LE/oscNext/pass2/noise/level3_v02.00	15,186,239
1% Burnsample	All	Data (2011-2021)	/data/ana/LE/oscNext/pass2/data/level3_v02.00	61,739,862

TABLE 7.1: Overview of all data used in this chapter, both simulated and actual data. Includes the data type, number of events as well as the location on the IceCube Cobalt Server at NBI for the reference of future students.

As written in table 7.1, two separate muon samples have been used. Initially Morten and I used the muon sample nr: 130000, which contains muons aimed mainly at DeepCore. However, plenty of high energy muons contaminated the track neutrino sample we classified in real data. Upon discussion with members of the OscNext group, it was determined that the muon sample nr: 139008 would be more suitable for training the GNN.

In figure 7.1, the distributions of the Monte Carlo neutrinos and the two muon samples in energy, zenith, azimuth and interaction vertex position are presented. The

¹These simulations were explained in section 4.4.

²The data in IceCube are divided into runs and subruns.

139008 sample contains higher energy muons than sample 130000, whereas their angular distributions are relatively similar. The 139008 sample also has muons which interact in the entirety of the detector, instead of mainly around DeepCore. Given that the muon sample nr: 130000 did not contain muons with energies as high as the neutrinos in the Monte Carlo data, it was thought that perhaps this was the reason why muons were classified as neutrinos. Therefore the multiclassification was re-trained, replacing the muon sample nr. 130000 with sample nr. 139008. While this actually alleviated much of the muon contamination, it was not enough to get a truly clean neutrino sample. In section 7.5, the additional step required to get a high purity neutrino sample is explained. It is also worth noting that the old sample 130000 contains a much larger amount of muons than the new sample 139008. Therefore the large muon sample 130000 is used in the analysis, while the model was trained on the muon sample 139008.

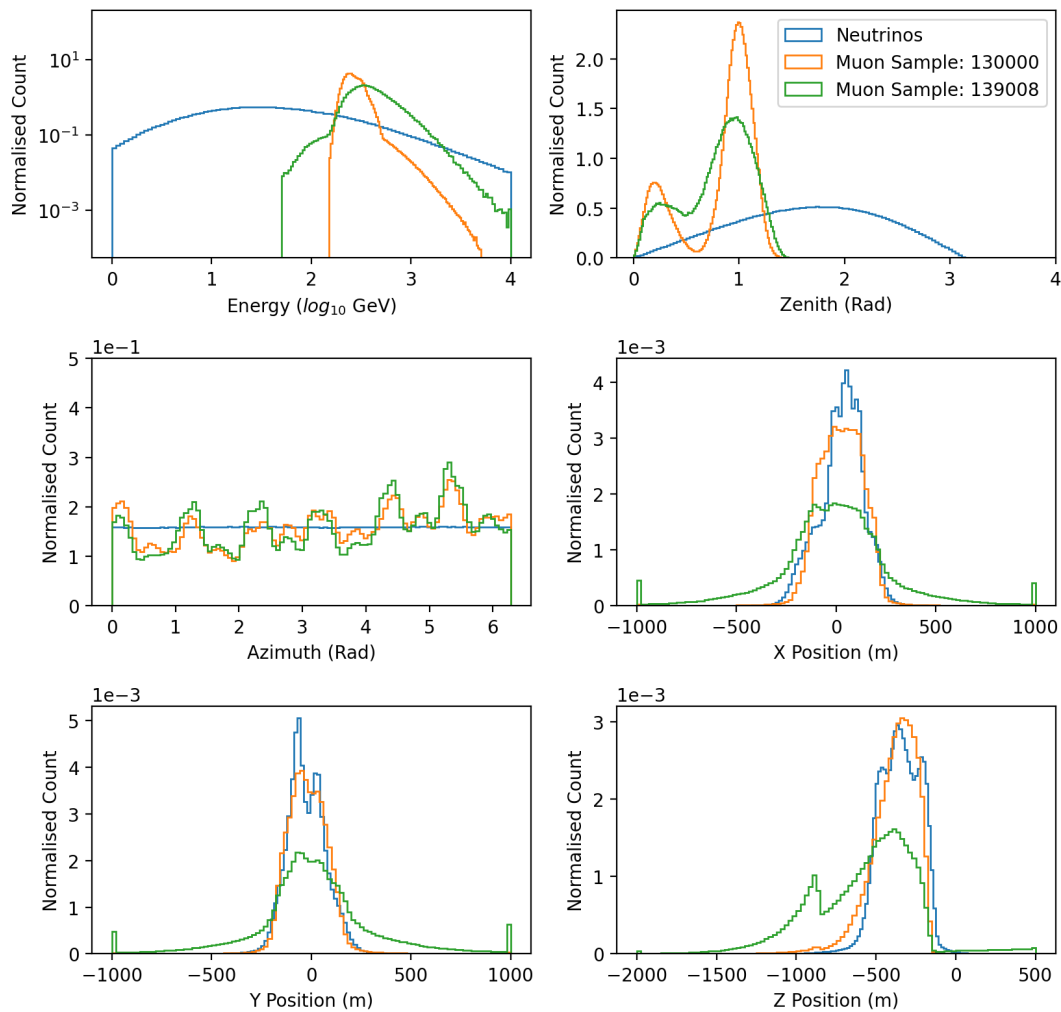


FIGURE 7.1: Distribution of energy, azimuth, zenith, and xyz position for MC neutrinos and MC muon sample nr: 130000 and 139008. Note that outlier bins are used for the position histograms. Normalised counts refer to the fact that the distributions are normalised such that their integral are 1. Finally it is the truth values that are plotted.

7.3.2 Training, Validation, Test Split For Each Model

Various considerations are taken into account in deciding the training/validation/test splits and how many of each particle type to use to train each GNN model.

For the multiclassification model (see table 7.2), it was decided to use equal amounts of noise, muons and neutrinos, to ensure that the model learns to classify each type equally well. Furthermore, each neutrino flavor make up a third of the total neutrino sample. The total amount of each particle type was restricted by the relatively low statistics in the muon sample nr. 139008. Therefore, the multiclassification model was trained on 1.65 million events. Training on a larger sample would probably result in a slightly better model, but as we will see, it seems to work very well as it is.

Multiclassification	Total	Noise	Muons (139008)	Total neutrinos	Electron neutrinos	Muon neutrinos	Tau neutrinos
Training	1.65 mill	33.3%	33.3%	33.3%	11.1%	11.1%	11.1%
Validation	300.000	33.3%	33.3%	33.3%	11.1%	11.1%	11.1%
Test	509.211	33.3%	33.3%	33.3%	11.1%	11.1%	11.1%
Total in database	38.9 mill	871.103	719.737	37.3 mill	8.3 mill	20.1 mill	8.9 mill

TABLE 7.2: Multiclassification training, validation test split. There is an equal amount of noise, muons and neutrinos, and an even distribution in neutrino flavors. The training:val:test split is 62.5% : 13.9% : 23.6%.

The interaction vertex position model (see table 7.3) is only trained on neutrinos, since it is to be applied to a clean selection of neutrinos. The training and validation sets are the exact neutrinos from the multiclassification training and validation sets. The test set contains all remaining neutrinos.

Vertex Position	Total	Total neutrinos	Electron neutrinos	Muon neutrinos	Tau neutrinos
Training	450.000	100%	33.3%	33.3%	33.3%
Validation	100.000	100%	33.3%	33.3%	33.3%
Test	36.7 mill	100%	22.0%	54.3%	23.7%
Total in database	37.3 mill	37.3 mill	8.3 mill	20.1 mill	8.9 mill

TABLE 7.3: Vertex Position training, validation, test split. There is an even distribution in neutrino flavors.

The track/cascade classification model (see table 7.4) does not have an optimal split of neutrino interaction types and flavors in the training and validation sets. They do contain 50% tracks (CC ν_μ) and 50% cascades (all remaining types), which makes sense. Furthermore, no CC ν_τ events are present in the training sample, since they create tracks with a 17% branching ratio. This is a choice that has also been implemented by others: [1]. The minor flaw in the sets, is that the CC ν_τ should be present in the validation set, and that there should be an even split between cascade types. However, this should not make a large difference, and as we shall see, the model performs well.

Track/Cascade Classification	Total	NC ν_e	CC ν_e	NC ν_μ	CC ν_μ	NC ν_τ	CC ν_τ
Training	1 mill	4%	28%	9%	50%	9%	0%
Validation	100,000	4%	28%	9%	50%	9%	0%
Test	4,112,510	10%	10%	10%	50%	10%	10%

TABLE 7.4: Training, validation, test split for track/cascade classification model. There is an even split between tracks (CC ν_μ) and cascade events (others).

The energy, azimuth and zenith reconstruction models were trained and validated on the same sets as those from the track/cascade classification model. This is not exactly ideal, given that they do not have equal neutrino flavors. On the other hand, the main difference is whether an event is track or cascade-like, which the model should prioritise equally, given the equal training split between tracks and cascade neutrinos.

Energy, Azimuth and Zenith Reconstruction	Total	Total neutrinos	Electron neutrinos	Muon neutrinos	Tau neutrinos
Training	1,000,000	100%	32%	59%	9%
Validation	100,000	100%	32%	59%	9%
Test	36.7 mill	100%	22.0%	54.3%	23.7%
Total in database	37.3 mill	37.3 mill	8.3 mill	20.1 mill	8.9 mill

TABLE 7.5: Energy, azimuth and zenith reconstruction model. Training, validation, test split.

The test sets presented above are those used to test the models performance in Monte Carlo. When it comes to comparing the neutrino selections in Monte Carlo and actual data, all Monte Carlo data that the multiclassification has not been trained or validated on are used. This is 15.99 million muons (from sample 130000), 14.80 million noise events and 36.75 million neutrinos.

There is a general caveat of this chapter, which pertains to the use of training data. One million of the neutrino sample, which is used in the general test set of the following analysis (36.7 million neutrino events), was used in the training of the models Morten and I trained together (track/cascade, energy, zenith and azimuth). While this could skew the results slightly, it is less than 3% of the test neutrinos. Furthermore, it would only cause the results for these neutrinos to be minutely better, given that the training loss of the models were comparable to the validation loss when early stopping ended the training. However, given more time, I would retrain the models using the same training/validation/test splits that was used in the interaction vertex position model. The reason why the one million neutrinos are not simply removed, has to do with the way Monte Carlo particles are re-weighted to estimated rates (how many hit the detector per second). Simply removing them would unfortunately skew this re-weighting.

Understanding the data the models were trained on, lets turn to analysing their performance in Monte Carlo data.

7.4 Monte Carlo Classification And Reconstruction Results - Including Benchmark Against Retro Reconstructions

In this section the Monte Carlo results of the DynEdge models will be presented. The reconstruction models (energy, zenith, azimuth & interaction vertex position) can be directly compared against the OscNext reconstruction algorithm Retro, on the subset of events that make it far enough in the OscNext selection process to be reconstructed (lvl 6). As such the reconstruction comparison against retro only includes events that are actually reconstructed by retro. It is important to bear in mind that the GNN reconstruction models have not been subject to hyperparameter optimization beyond what's presented in [1]. Furthermore, the models in this work are only trained on relatively few events (450,000 - 1 mill). Thus they are not necessarily an attempt to beat Retro, since this was already achieved in a published paper on DynEdge (GraphNeT) [1]. Instead the reconstructions are essentially what allow us to compare the distributions of the neutrino selections in Monte Carlo and real data. Despite this, the reconstruction models perform surprisingly well when considered that they are trained on lvl2+DC³ neutrinos and a relatively uncleaned pulsemap⁴.

The multiclassification and track/cascade classification models are not benchmarked against other methods in this section, since there exist no comparable methods this early in the event selection. Instead the rate of clean neutrinos our models achieve are compared to OscNext in section 7.7.

Before we jump to the results, a brief explanation of performance measurers is required.

7.4.1 Performance Measurement Tools

ROC Curve

A Receiver Operating Characteristics (ROC) curve is a classical measurement of the performance of binary classifiers in Machine Learning, but it can also be applied to Multiclassification models. An example is available in figure 7.4. A traditional binary classifier, for instance the track/cascade model, outputs a probability of the event being a track neutrino. The ROC curve is simply a scan across that probability from 0 to 1. For each value, the events that have higher probabilities are predicted to be track neutrinos and the others are predicted to be cascade neutrinos. The percent of actual track neutrinos that are correctly classified as track neutrinos using the threshold is called the True Positive Rate (TPR). The percent of cascade-like events that are wrongly classified as track neutrinos is called the False Positive Rate (FPR). The ROC curve is simply a plot of the TPR as a function of FPR. The optimal result is to have a TPR of 1, while having a FPR of 0, since this corresponds to correctly predicting all the events. This point lies in the top left corner of the plot, whereas randomly guessing would correspond to a diagonal line from 0 to 1. In addition to visually inspecting the ROC curve, one can also calculate the Area Under the Curve (AUC), which is then a measure of how well the model works. It is as the name suggest, simply the area below the ROC curve. The closer the AUC is to 1, the better.

³See sections 4.6. Lvl2 + DC is the rawest form of data that OscNext receives from the detector.

⁴All the models are trained on the uncleaned pulsemap SplitInIcePulses, since the multiclassification model worked better on it than the cleaned SRTInIcePulses. It was not tested if this is also the case for the reconstruction models.

For a multiclassifier (which outputs one probability for each class), the approach "all versus one" is used to give a ROC curve for each class. For instance, the ROC curve for the neutrino probability, is calculated by considering it to be a binary classification of neutrinos vs the rest (in this case muons and noise), using the neutrino probability [77]. ROC curves can be very misleading if they are calculated on an unbalanced dataset. If for instance 99% of the data are muons, simply classifying all events as muons would result in a very good ROC curve. Therefore, all ROC curves in the work are calculated using balanced test sets.

Residuals And Resolutions

For reconstruction models, it is typical to consider the residual distributions. In table 7.6, the definitions of the residuals of the reconstruction targets are listed.

Target	Residual Definition
Energy	$\text{Log}_{10}(E_{reco}) - \text{Log}_{10}(E_{true})$
Zenith/Azimuth	$\text{Angle}_{reco} - \text{Angle}_{true}$
Vertex Position for resolution	$ \vec{P}_{reco} - \vec{P}_{true} $
Vertex Position individual	$P_{reco}^{\alpha} - P_{true}^{\alpha}$ where α can be (x,y,z)

TABLE 7.6: Residual definitions for reconstruction targets.

Furthermore, to gauge the width of the residual distributions, the resolution is calculated and visualized as a function of energy. This is done exactly as in chapter 6 for the angles and energy:

$$w = \frac{p84(R) - p16(R)}{2}$$

However, for the interaction vertex position, the resolutions is calculated as:

$$w = p50(R)$$

This is a more appropriate measure for distributions that are bounded in one direction, such as the vertex position residuals, which cannot be negative.

Chi Square Test

To test statistically if the distributions in data match those in Monte Carlo, χ^2 tests are applied. In general the uncertainties of the amount in each bin are calculated using the assumption that the number of events in a single bin is Poisson distributed. As such, the uncertainty in a data bin is equal to the square root of the number of events in the bin. The uncertainty in the rate⁵ is then the count uncertainty multiplied by the rate of the bin divided by the count of the bin.

With the uncertainties, the χ^2 value can be calculated as:

$$\chi^2 = \sum \left(\frac{r_{\text{data}} - r_{\text{MC}}}{\text{sigma}(r_{\text{data}})} \right)^2$$

⁵The rate is the amount of particles hitting the detector per second. In Monte Carlo, the OscNext groups theoretical weights are used to convert the number of events to rates. In data, an estimate of the number of seconds in which the data was taken, is use to calculate an approximate rate. The rate will be used in most plots comparing data to Monte Carlo. It is also used in section 7.7 to determine the rate of our neutrino selection vs that of the OscNext selection.

Where r is the rate and $\sigma(r)$ is the Poisson uncertainty of the rate.

The number of degrees of freedom is the number of non-empty bins minus potential scale factors that are used to normalise Monte Carlo to data. Finally a p-value is calculated, which represents the probability of obtaining an identical χ^2 or worse if the real data was drawn from the Monte Carlo distribution. This means that a high (low) p-value suggest a good (bad) data/MC correspondence.

7.4.2 Multiclassification - Noise, Muons Or Neutrinos?

Previously the method for classifying neutrinos using GraphNet consisted of applying two separate binary classification models. First separating particles from noise, then selecting neutrinos from muons in the clean particle sample. The problem with this approach is that the noise that inevitable ends up being classified as particles, could resemble the neutrinos more than the muons. As such, it is possible that the noise events which are wrongly classified, ends up in the "clean" neutrino sample⁶.

A potential solution and simultaneously simpler method is to use multiclassification to directly divide events into either noise, muons or neutrinos, as illustrated on the overview in figure 7.2. This method was implemented by Morten Holm and myself.

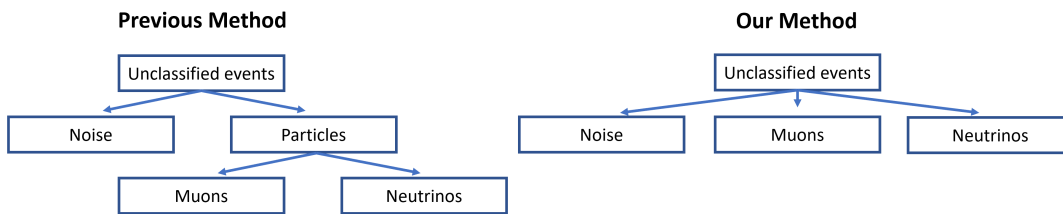


FIGURE 7.2: Graphic overview of event classification methods. (Left) Using two separate binary classifiers. (Right) Using a single multiclassification classifier.

The multiclassification model outputs three probabilities, one for each of the particle types, which sum to one. Histograms of the neutrino probability for each particle type in the equal sized⁷ test set can be seen in figure 7.3a. It is evident that the model works quite well, with the actual neutrinos distributed much further to the right than the muons and noise events. Similar plots for the muon and noise probabilities can be found in appendix B.1.

⁶It has unfortunately not been tested if the multiclassification model outperforms two separate binary classifiers. The multiclassification is still used due to its simplicity and the good results it gives in Monte Carlo.

⁷Equal sized, or even split test set refers to one where there are equally many of each particle type.

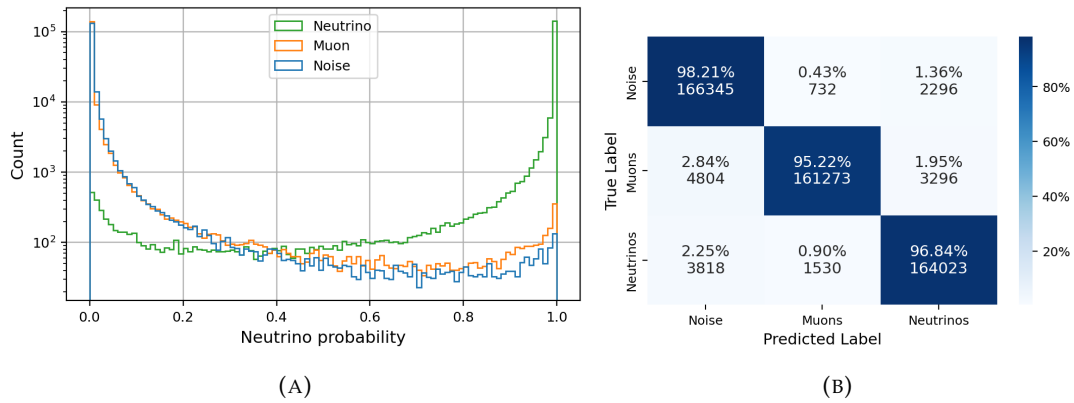


FIGURE 7.3: (A) Histograms of all test events as a function of their predicted probability of being neutrinos. Colored by actual particle type. (B) Confusion matrix for multiclassification test set, with equal fractions of noise, muons and neutrinos. Shows the number and percent of each particle type that is classified as either neutrino, muon or noise. In the confusion matrix, an event is classified based on which probability is the highest.

The confusion matrix of the multiclassification is available in figure 7.3b. It shows the number of each particle type that is classified as noise, muons or neutrinos. For each particle, the prediction is simply taken to be the highest probability of the three particle types.

From the confusion matrix, we see that about 95% - 98% of events are classified correctly in an even split test set, which seems quite good. However, once the data has been weighted to the expected, physical rates, there are orders of magnitudes more muons and noise than neutrinos. Therefore, using the highest probability to determine the prediction, would still result in the predicted neutrino sample being dominated by wrongly classified muons and noise. Therefore it is also interesting to look at the ROC curves, which were explained in section 7.4.1 and can be seen in figure 7.4.

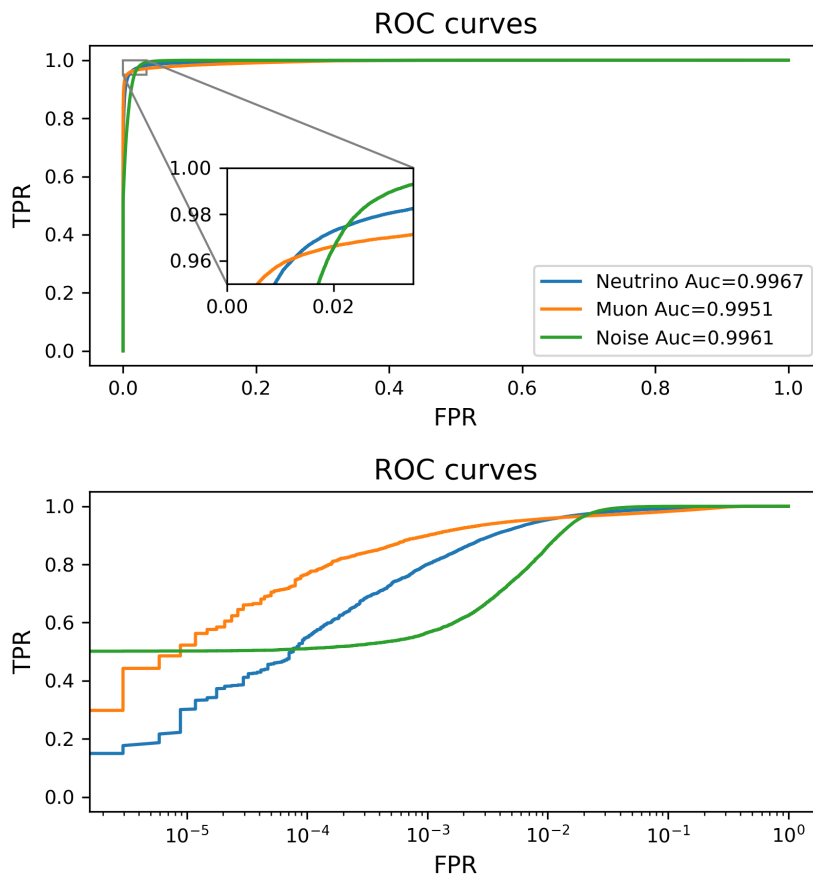


FIGURE 7.4: ROC Curves for the multiclassification test set. Shows the true positive rate (TPR) as a function of the false positive rate (FPR) depending on where in the neutrino, muon and noise probabilities the cut is made.

The ROC curves also show that the multiclassification works well. The AUC scores are almost 1 and depending on the cut in probabilities, one can obtain a particle sample of high purity. For instance, we could get a TPR of about 0.5 for neutrinos, while only having a FPR of $10^{-5} - 10^{-6}$. Thus we could select half the actual neutrinos, while only classifying $\frac{1}{100,000}$ to $\frac{1}{1\text{mill}}$ of the muons and noise as neutrinos. This is exactly the type of consideration that is required when classifying neutrinos in actual data, where they are outnumbered by muons and noise. We will return to the question of selection a clean neutrino sample in section 7.6.

7.4.3 Track/Cascade Classification

The track/cascade classifier is a traditional binary classification model. It outputs a single probability, that of being a track neutrino (CC ν_μ). The resulting histograms of track and cascade neutrinos in the equal split test set is available in figure 7.5a.

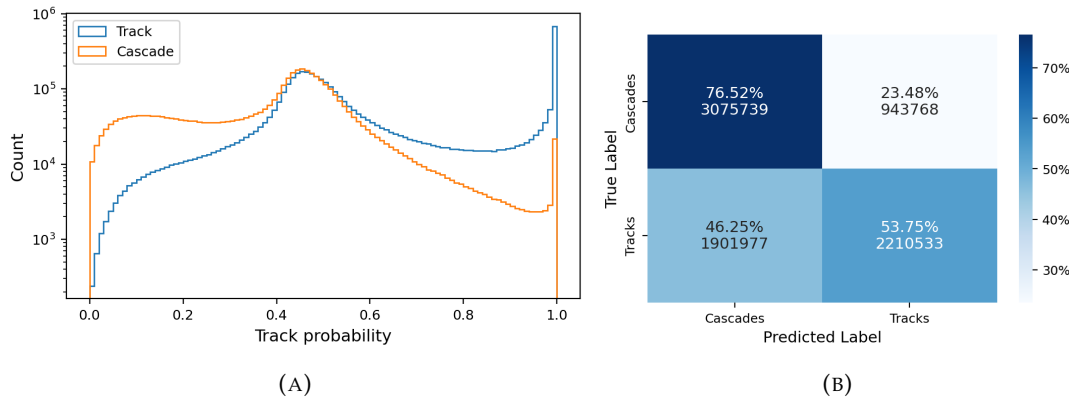


FIGURE 7.5: (A) Histograms of predicted track probability for all events in the test set. Separate histograms for actual track and cascade neutrinos. Colored by actual particle type. (B) Confusion matrix for track cascade test set, with equal fractions of track and cascade neutrinos. Shows the number and percent of each type that is classified as either a track or cascade neutrino. In the confusion matrix, an event is classified as a track neutrino if the track probability is above 0.5 and otherwise as a cascade neutrino.

The track/cascade classifier is not able to distinguish as well between the two types of events as were the multiclassification model. However, the track distribution is still skewed to the right and the cascade distribution to the left, while they share a peak in the uncertain center. A detail to note, is that the cascade events also have a small peak around 1 in the track probability, which we shall see is caused by the 17% of CC ν_τ that also produce track signatures in the detector⁸.

The confusion matrix in figure 7.5b gives a similar picture of the classifier performance. If we just predict based on a cut of 0.5 in the track probability, 76.52% of cascade neutrinos are correctly predicted, while the same is only true for 53.75% of the track neutrinos.

Track neutrinos are not difficult to predict because the model cannot tell the difference between track-like and cascade-like signatures in the detector. Instead it is highly dependent on the energy, since the track neutrinos also produce an initial cascade, which the resulting muon only outruns if it is energetic enough⁹. Put differently, if the CC ν_μ is low-energetic, it looks exactly like a cascade neutrino. This argument is based on figure 7.6, which shows a 2D histogram of energy and track probability for the CC ν_μ particles. It can be seen that at energies below $10^1 - 10^{1.5}$ GeV, the probabilities are mainly around 0.4-0.5, the uncertainty bump in the histograms in figure 7.5a. When the energy is higher, the vast majority of track neutrinos have a track probability of practically 1. Those that do not, could be interacting near the edge of the detector and thus not depositing a track signature.

⁸See section 4.5.

⁹This was also explained in section 4.5.

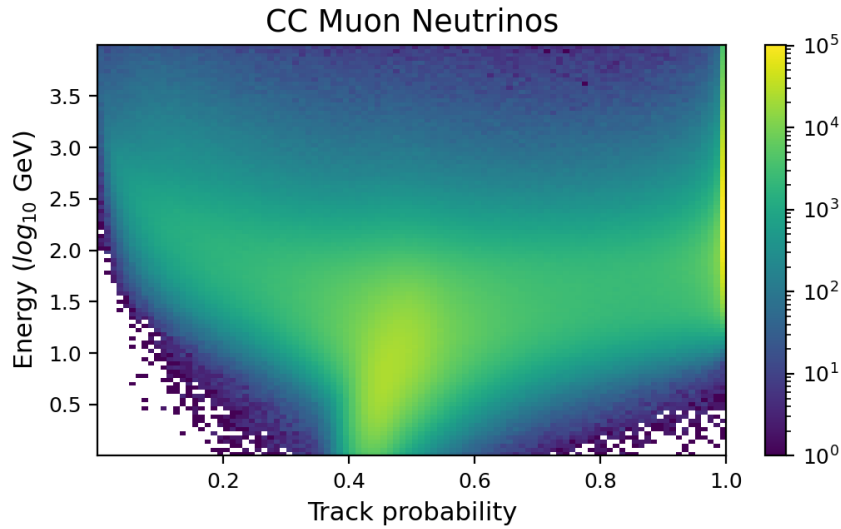


FIGURE 7.6: 2D histogram of predicted track probability vs energy for all track ($CC \mu_\tau$) events in an extended test set. Shows that the probability of a track event being classified correctly is highly dependant on the energy, since the resulting muon has to outrun the initial cascade.

As can be seen in figure 7.7, which shows normalised track probability histograms for each combination of neutrino interaction type and flavor, the cascade neutrino peak near a track probability of 1 comes from the $CC \nu_\tau$. A simple check to see if the distribution of $CC \nu_\tau$ match the theoretic branching ratio can be carried out using the rightmost bin in figure 7.7. The ratio $\frac{CC\nu_\tau/NC\nu_\tau}{CC\nu_\mu/NC\nu_\mu}$ for the rightmost bin is an estimate of how large a percentage of $CC \nu_\tau$ behaves as a $CC \nu_\mu$ in the track probability. The ratio is calculated to be 0.19, or 19% which is quite close to the expected 17% from section 4.5. An even better estimate could be calculated by scaling the $CC \nu_\mu$ and $NC \nu_\mu$ distributions to the $CC \nu_\tau$ distribution. The scaling factors would then give away the percent of $CC \nu_\tau$ events that behaves as $CC \nu_\mu$ events. This was not explored further due to time constraints.

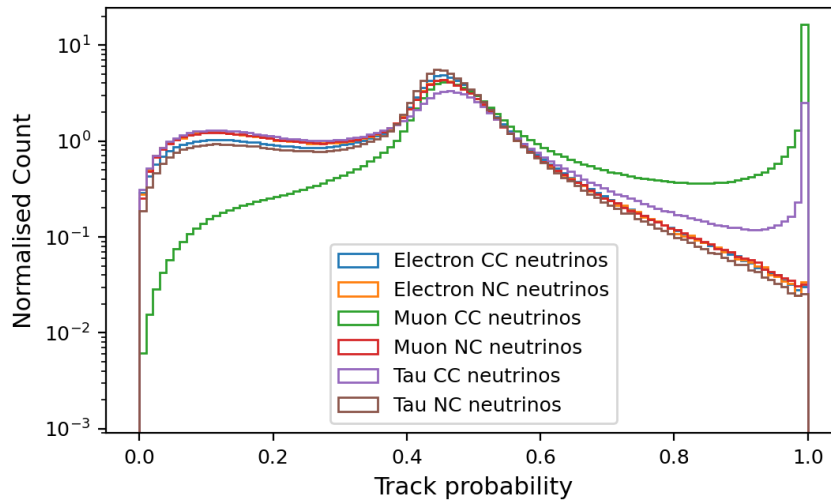


FIGURE 7.7: histograms of predicted track probability for all events in an extended test set. Separate histograms for each combination of neutrino interaction type and flavor. It is clearly visible from the peak at high track probabilities for CC μ_τ that it is capable of producing track event signatures.

The ROC curves in figure 7.8 also shows the difficulty in predicting track neutrinos. The AUC is 0.7346 for all events. This is comparable to the published result in [1], which had a AUC of 0.713. Their model was trained on OscNext lvl 7 Data and a cleaned pulsemap, which just shows that the track-cascade classifier works well on lvl 2 + DC events and a noisy pulsemap. The ROC curves also show that the higher energy neutrinos, the better the track/cascade classifier works. This is evident from the increasing AUC scores, as a function of the energy of the neutrinos used to produce the ROC curves.

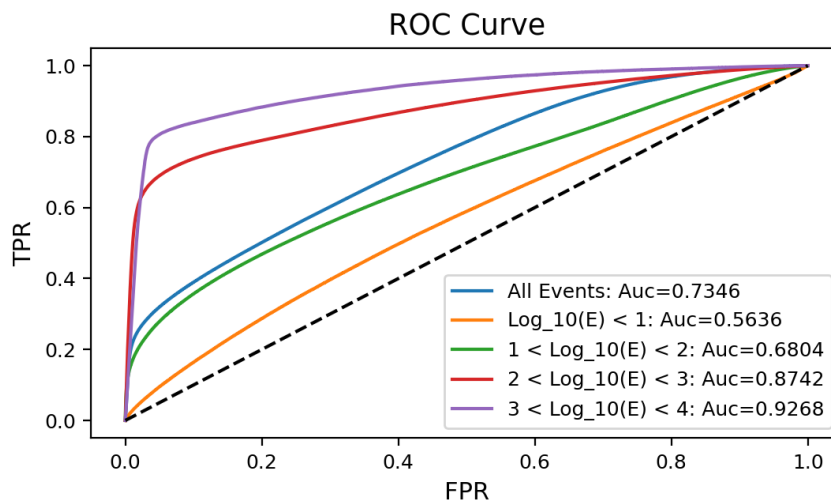


FIGURE 7.8: ROC Curves for the track/cascade test set. Shows the true positive rate (TPR) as a function of the false positive rate (FPR) depending on where in the track probability the cut is made. Separate ROC curves are plotted for all events and for specific energy ranges as well. It is evident that the higher energy neutrinos, the better the track/cascade classifier works.

7.4.4 Energy Reconstruction

The performance of the reconstruction models are shown first for all events, then in comparison to OscNext reconstruction algorithm Retro (on the subset of the neutrinos that make it to OscNext lvl 6).

As can be seen to the left in figure 7.9, the distribution of DynEdge predicted energies follow the true energy distribution decently, except for low energies, which DynEdge has difficulty predicting. In the same figure to the right, a 2D histogram of predicted vs true energy is visible. It shows a clear diagonal trend, which is good, but it has a natural blur since the model does not predict perfectly.

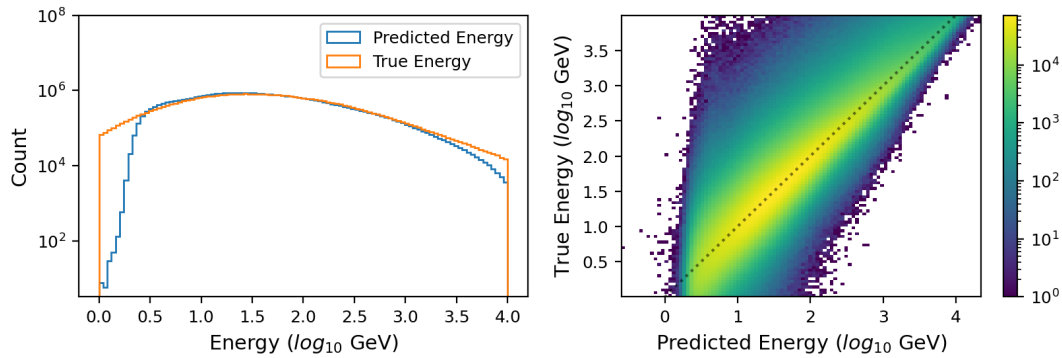


FIGURE 7.9: (Left) Predicted and True energy distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted energy. It can be seen that DynEdge is not likely to predict very low energies, but otherwise match the distribution decently.

For the events in the reconstruction test set that makes it far enough in the OscNext selection, a direct comparison to Retro is possible. The results are visible in figure 7.10. Especially the right plot shows that DynEdge generally has lower residuals than does Retro. This is despite the fact that Retro is only used on lvl 6 events specifically, whereas DynEdge is trained on all lvl 2 + DC events.

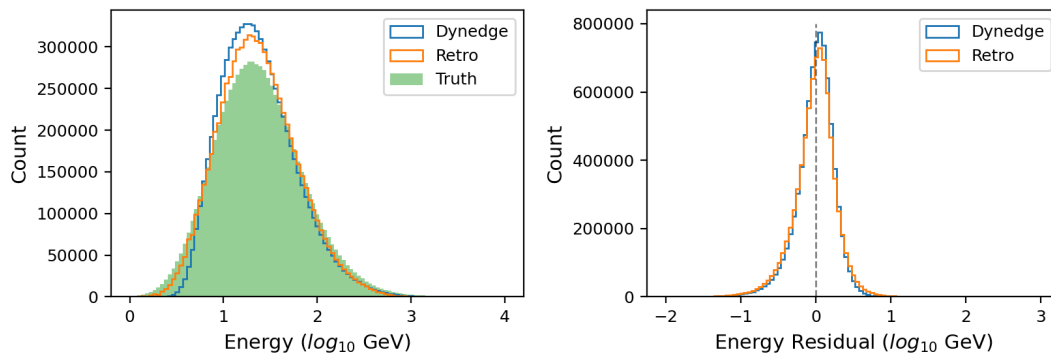


FIGURE 7.10: (Left) DynEdge predicted, Retro predicted and true energy distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

While DynEdge is better at energy reconstruction overall, the resolution depends strongly on energy, which can be seen in figure 7.11a. Here it is clear that DynEdge is best for energies below 10² GeV, whereas Retro is better at higher energies. The

results match those published in [1] decently, but are perhaps slightly worse. This could be because we did not use any hyperparameter optimisation, or trained on too few events. Otherwise it might be because the model in the paper is trained solely on lvl 7 events and a clean pulsemap, whereas ours is trained on lvl2 + DC and a noisy pulsemap.

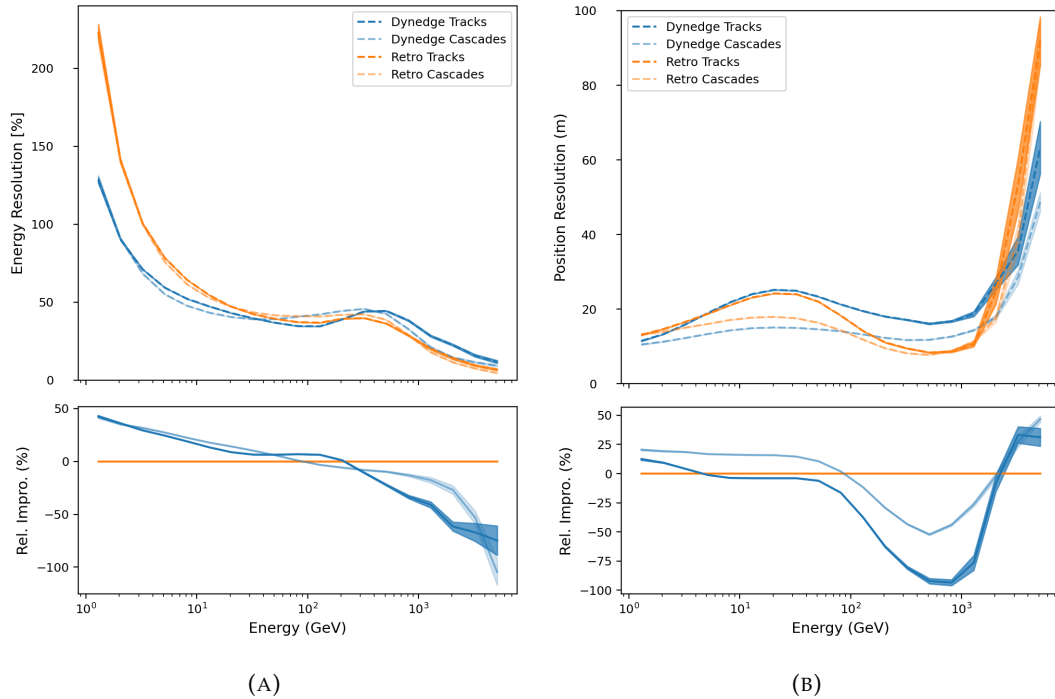


FIGURE 7.11: (A) (Top) DynEdge and Retro energy resolution divided into track vs cascade events. The track/cascade division is based on the actual flavor and interaction type, since this is Monte Carlo data. (Bottom) Relative improvement in energy resolution from Retro to DynEdge. (B) Identical to (A) but for vertex position resolution.

7.4.5 Interaction Vertex Position Reconstruction

The interaction vertex position is the location at which the neutrino interacts with the ice. The results are shown primarily for the z position in this section, except for the resolution, which is a mixture of all three coordinates. The x and y position results are available in appendix B.3.

In figure 7.12 the true and predicted vertex z position distributions are shown to the left and a 2D histogram of true vs predicted values to the right. The distributions match well and the diagonal trend in the 2D histogram is quite strong.

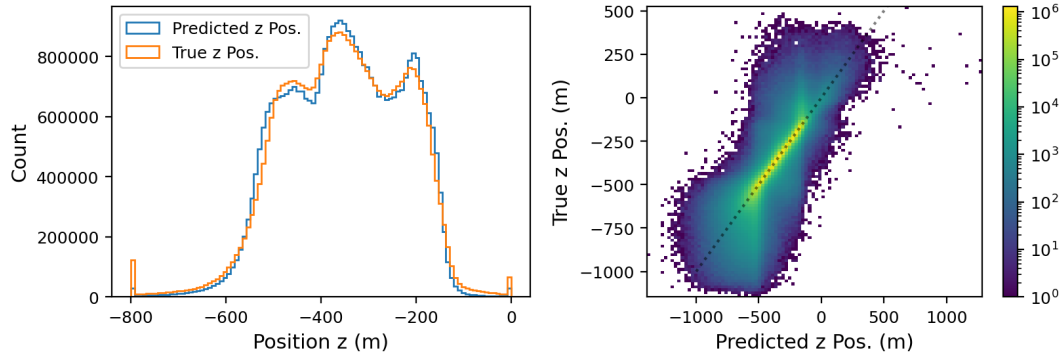


FIGURE 7.12: (Left) Predicted and True vertex position z distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted vertex position z .

Comparing against Retro in figure 7.13 shows a good agreement in distributions between DynEdge and Retro. On the other hand, Retro generally performs better, with a more narrow resolution distribution.

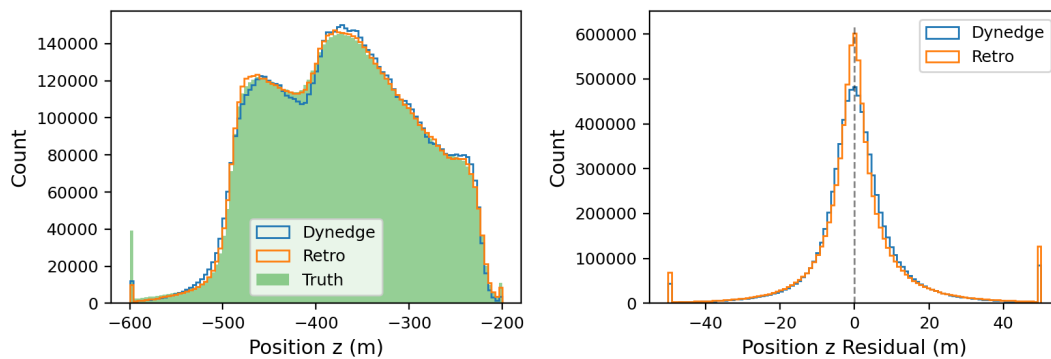


FIGURE 7.13: (Left) DynEdge predicted, Retro predicted and true vertex position z distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

A resolution plot is available for the vertex position in figure 7.11b. The Retro resolution is mostly better across the energy spectrum for track neutrinos, and DynEdge is practically only better for cascade neutrinos below 10^2 GeV. The results are worse than in [1], which could be due to the fact that the vertex position model is only trained on 450,000 events, instead of the 1 million used in the energy and angular reconstruction models.

7.4.6 Zenith Reconstruction

The zenith reconstruction results are along the same lines as the energy reconstruction results. In figure 7.14 the distributions and 2D histogram for all test events are visible. It can be seen to the left that the model has a hard time predicting events near directly south (zenith = 0) or near directly north (zenith = π). This is because the actual area a small deviation of the zenith angle traces out, is very small near the poles. This means that few events should come from those zenith ranges. However, the model exaggerates this. To the right in the figure, one can see a diagonal trend,

with quite substantial blur. Luckily the model also outputs an estimated standard deviation, which can be used to select events with good angular resolution.

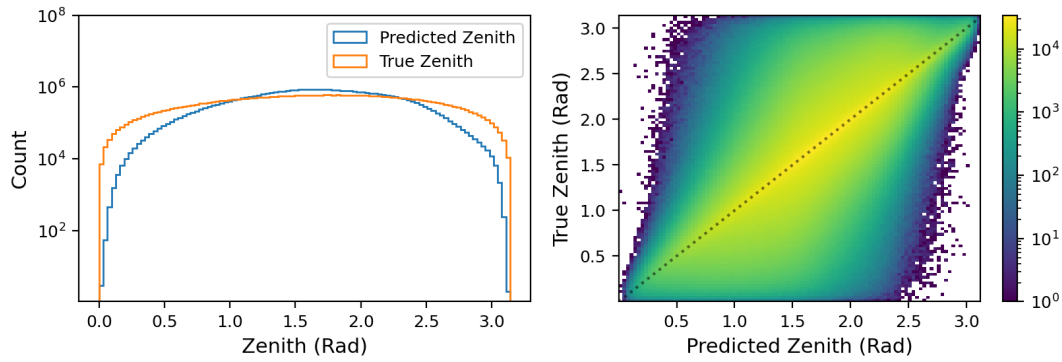


FIGURE 7.14: (Left) Predicted and True zenith distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted zenith angle.

In addition to the zenith angle, the model outputs an inverse variance called kappa:

$$\kappa = \frac{1}{std_{\kappa}^2}$$

It is used to calculate an estimated standard deviation. A pull plot, as seen in figure 7.15 is a way to estimate how correct the std_{κ} is. It shows the z-score of the zenith predictions, which is simple the residual divided by the estimated standard deviation:

$$Z - score = \frac{angle_{reco} - angle_{true}}{std_{\kappa}}$$

If the estimated std's were perfect, the pull plot would resemble a unit Gaussian. In the figure, a Gaussian has been fitted to the z-scores. It can be seen that the mean is practically zero, whereas the standard deviation of the Gaussian is 1.1 instead of 1. This means that the model standard deviation estimates are slightly too low. As such, the model can be said to be a little overconfident. On the other hand, the pull distribution does not follow the fitted Gaussian exactly, as can be seen in the p-value of 0. Instead the z-score distributions has a higher peak in the center, which means that the overconfidence is not as bad as the fit suggests.

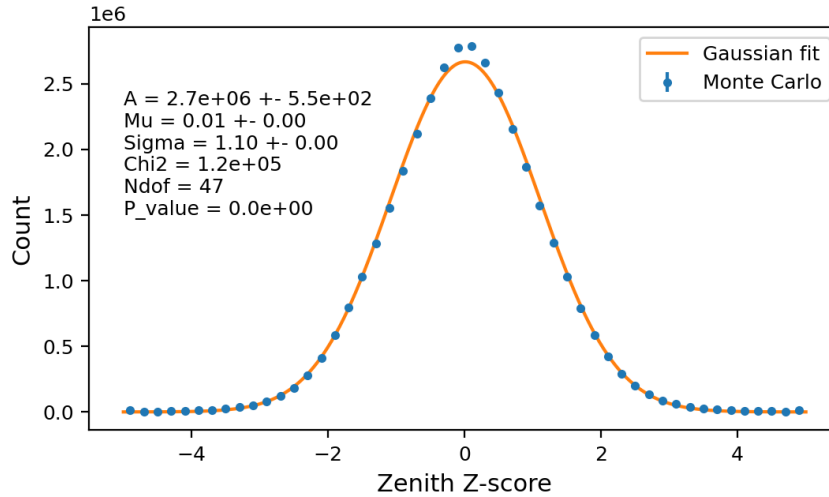


FIGURE 7.15: Pull plot. Histogram of number of predicted standard deviations the predicted zenith angle falls from the truth. Overlaid is a gaussian fit, since this should correspond to a unit gaussian, if the std is estimated correctly. A is the normalisation, mu is the mean, sigma is the standard deviation, Chi2 is the χ^2 value of the fit, Ndof is the degrees of freedom and p-value is the probability of obtaining a worse χ^2 if the data distributions is consistent with the fit. One can see that the mean is practically 0, while the sigma is 1.10 instead of 1. This means the model slightly underestimates the zenith standard deviations.

The zenith angle can also be compared against Retro. In figure 7.16, the zenith distributions are shown to the left and the residual distributions to the right. The distributions show that Retro follows the truth distribution better, but the residual distributions are quite similar, and difficult to distinguish.

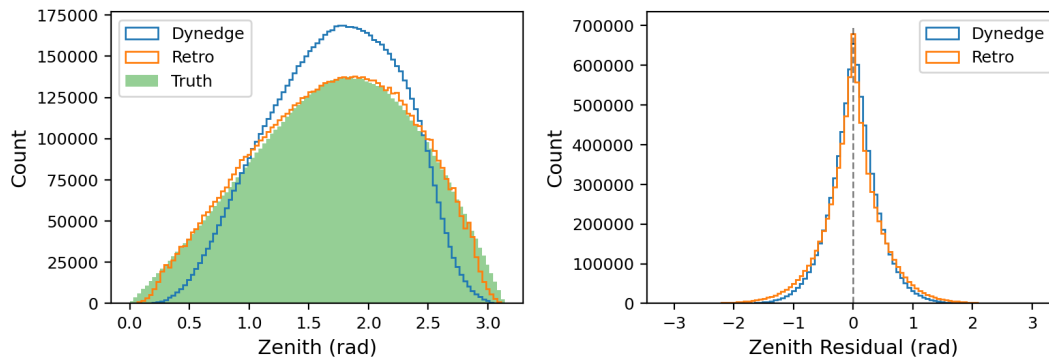


FIGURE 7.16: (Left) DynEdge predicted, Retro predicted and true zenith angular distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

A more clear picture emerges if we investigate the zenith resolution as function of energy. As can be seen in figure 7.17a, DynEdge generally has a better resolution, except for the energy range of $0.7 \cdot 10^2 - 2 \cdot 10^3$ GeV for track events. These results again agree quite closely with those of [1].

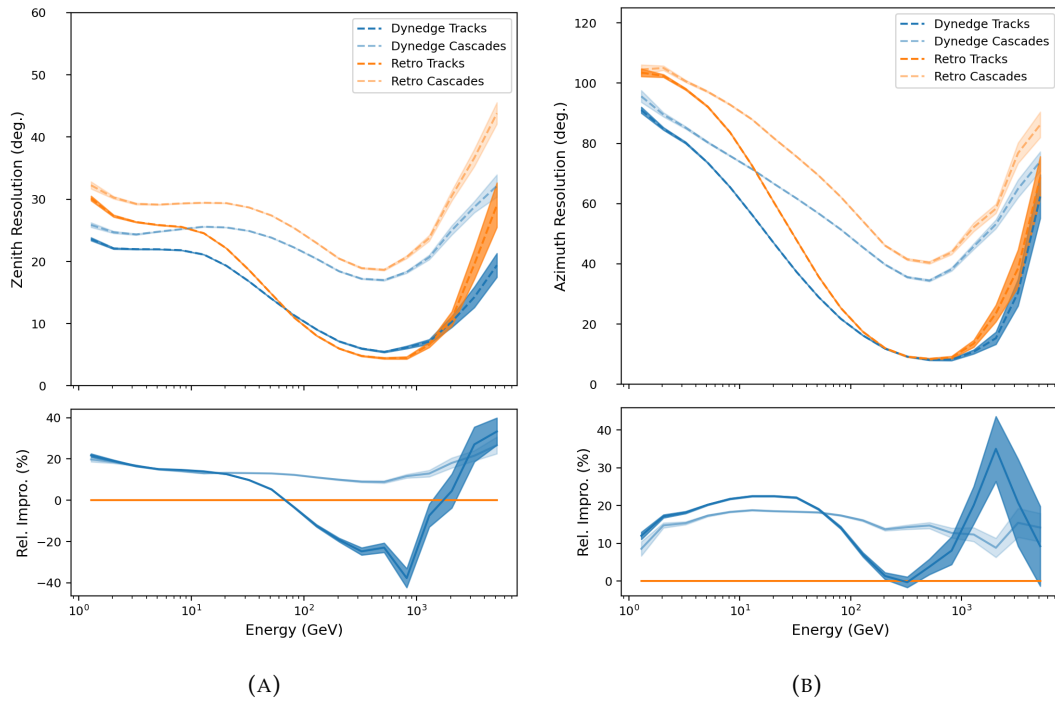


FIGURE 7.17: (A) (Top) DynEdge and Retro zenith resolution divided into track vs cascade events. The track/cascade division is based on the actual flavor and interaction type, since this is Monte Carlo data. (Bottom) Relative improvement in zenith resolution from Retro to DynEdge. (B) Identical to (A) but for azimuth.

7.4.7 Azimuth Reconstruction

The results of the azimuth reconstruction are quite similar to those of the zenith reconstruction. In figure 7.18 the distribution and 2D histogram for all test events are visible. It can be seen to the left that the model does not output a uniform azimuth distribution, but predicts too many events in the center and edges. To the right in the figure, one can see a diagonal trend, with a very substantial blur. As with the zenith reconstruction, the model also outputs an estimated standard deviation, which can be used to select events with good angular resolution.

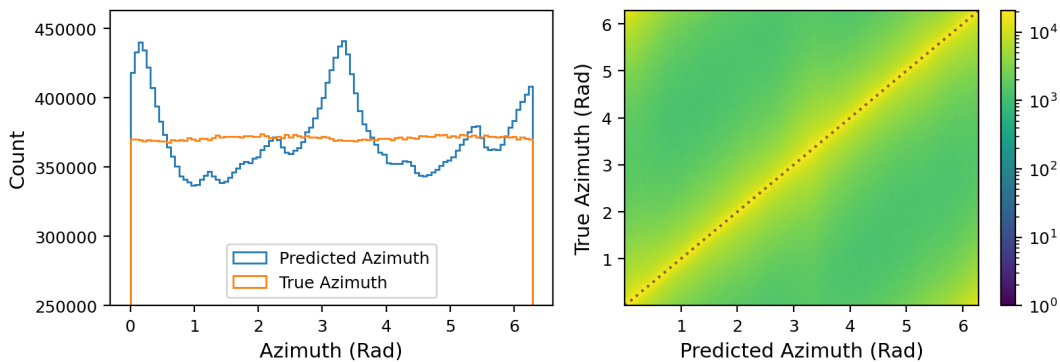


FIGURE 7.18: (Left) Predicted and True azimuth distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted azimuth angle.

A pull plot for the azimuth angle is available in figure B.3 in appendix B.2. In essence the conclusion regarding the azimuth pull plot is identical to that of the zenith pull plot.

The azimuth angle can also be compared against Retro. In figure 7.19, the azimuth distributions are shown to the left and the residual distributions to the right. The distributions show that Retro follows the truth distribution better, but the DynEdge distribution looks better for this subset of events, than overall. The residual distributions are quite similar, and difficult to distinguish.

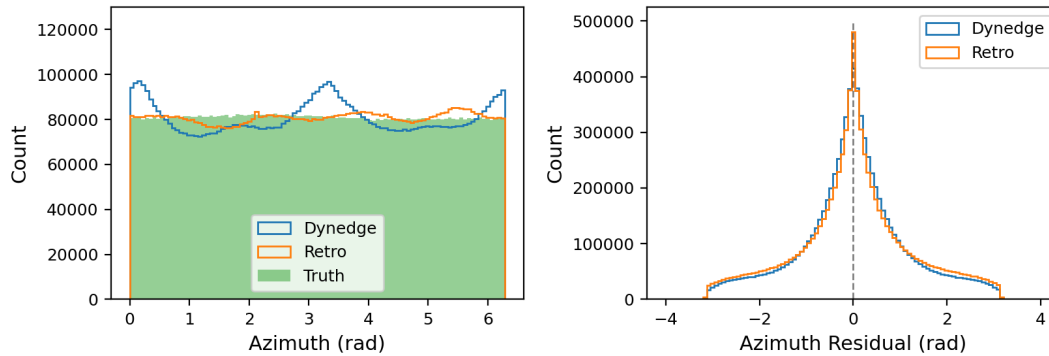


FIGURE 7.19: (Left) DynEdge predicted, Retro predicted and true azimuth angular distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

As for zenith a clearer picture emerges if we investigate the azimuth resolution as function of energy. As can be seen in figure 7.17b, DynEdge generally has a better resolution than Retro across the energy spectrum.

7.4.8 Multiclass Neutrino Selection Efficiency

The performance of the multiclass neutrino classification can be tested in Monte Carlo. In figure 7.20, the neutrino selection efficiency is plotted as a function of energy, zenith, azimuth, vertex position z and ρ . The efficiency is defined as the fraction of neutrinos that end up in our final neutrino selection, which will be introduced in section 7.6.

ρ is the orthogonal distance to the interaction vertex from string number 36, which is located at $(x,y) = (46.29, 34.88)$, approximately at the center of the IceCube detector. Thus it is a measure of the distance from the center of the detector. It can be seen that the efficiency is higher for neutrinos passing through the earth with zenith $> \frac{\pi}{2}$. Also, the efficiency is generally higher for high energies, except for a drop around $10 - 10^{2.2}$ GeV. The events that fall close to the center of the detector is also more likely to be classified as a neutrino, which can be seen in the efficiency as a function of ρ . Finally the efficiency does not depend on the azimuth angle, but generally increases the lower in the detector the events interact as seen in the z position figure.

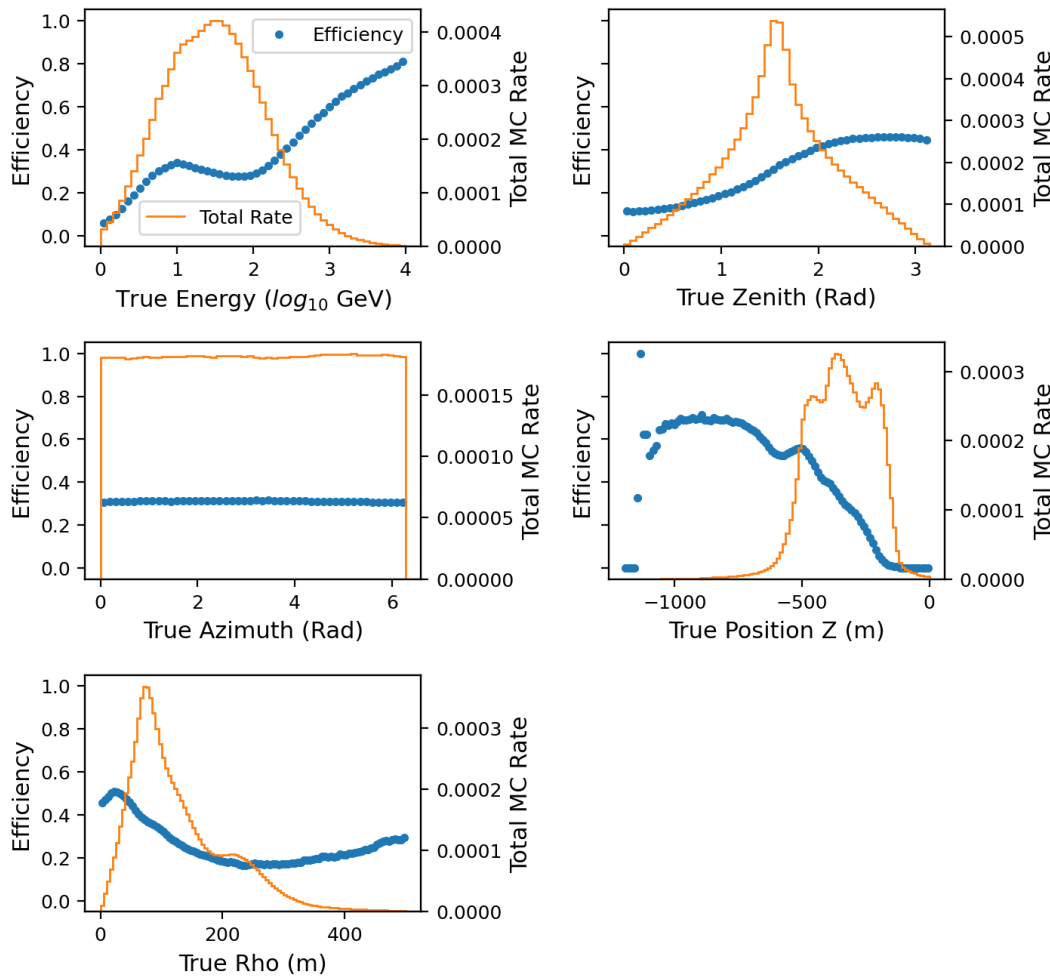


FIGURE 7.20: Multiclass neutrino selection efficiency in Monte Carlo as a function of energy, zenith, azimuth, vertex position z coordinate and rho. Also plotted is the total neutrino rate.

The rate that is presented in figure 7.20 represents the estimate of how many neutrinos hit the detector per second. Many of the figures in this chapter are plotted as rates instead of number of particles. In Monte Carlo data, the events are weighted to match the theoretical expected rate of particles as functions of energy and angles. This is done using weights calculated by the OscNext group. In real data, the rate is estimated from the total detector runtime of events used in this analysis¹⁰. The weight of each event in the actual data is then simply the inverse of the total runtime.

7.5 Necessary Data Cleaning Required To Ensure Good Enough Monte Carlo / Data Agreement To Apply GraphNet

The analysis that follows was first carried out directly on all lvl2 + DC events. It was quickly discovered that there was a large muon contamination in the supposed clean track neutrino sample. Furthermore, there also seemed to be some noise left.

¹⁰Thanks to Thomas Simon Stuttard for providing me with the runtime for each subrun of data used.

The new muon sample helped alleviate the issue with muons, but the Monte Carlo - data agreement in the neutrino selections were still not encouraging.

It turned out that at lvl2 + DC the data - Monte Carlo agreement is simply too poor for machine learning to yield the same results in data as in Monte Carlo. For instance, coincident and very high energy muon events were classified as neutrinos, simply because the model had not been trained on such events. One solution would be to obtain simulations of said events, which turned out to be difficult. Instead the cuts used by OscNext in going from lvl2 + DC to lvl 3 were used. These are exactly implemented to improve data - Monte Carlo agreement. The variables along with the cuts are available in appendix C.1. There plots of the lvl 3 variable distributions of neutrinos, muons and noise, in both Monte Carlo and data are also available.

There are several indications that the lvl 3 cuts are necessary for our method to work in actual data. First of all, the neutrino selection in actual data has a rate which is 60% higher than the Monte Carlo rate, when using lvl2 + DC data. With the lvl 3 cuts, the neutrino selection in data is 0.7% smaller than in Monte Carlo. This suggests that the neutrino sample in data without the lvl 3 cuts contains a lot of muons and/or noise. This can also be seen in figures C.8, C.9, C.10, C.11 and C.12 in appendix C.2. When compared to the plots in section 7.6, they clearly show that the data - Monte Carlo agreement for the neutrino selections are much worse, when the lvl 3 cuts are not applied.

7.6 Clean Track/Cascade Neutrino Selection and Comparison Between Monte Carlo and Real Data

Selecting a clean neutrino sample in data requires understanding approximately what the data contains. Therefore, the data distribution of neutrino probabilities are compared to the neutrino probabilities for Monte Carlo noise, muons and neutrinos. The rate of the data is plotted and the distributions for MC noise, muons and neutrinos are scaled individually such that they together match the data distribution as well as possible. The resulting plot is available in figure 7.21. To get a relatively pure neutrino sample, the cut has to be somewhere above 0.9999, which means that it is easier to visualize in a logit transformed probability space. Thus the following transformation is applied to the probabilities:

$$\begin{aligned} \text{logit}(p) &= \log\left(\frac{p_\epsilon}{1 - p_\epsilon}\right) \\ p_\epsilon &= p * (1 - 2 * \epsilon) + \epsilon \\ \epsilon &= 10^{-7} \end{aligned} \tag{7.1}$$

Furthermore, the noise, muon and neutrino distributions are scaled by 0.118, 1.308 & 1.085 using χ^2 minimization. The scaling is applied, since the data has a total rate that is 20% higher than in Monte Carlo, and the noise, muon and neutrino rates in Monte Carlo might not match those in data exactly¹¹. The error bars are calculated as Poisson errors (square root of number of data points, then multiplied by the total rate those data points represent, divided by the count of data points) for the data

¹¹In the OscNext analysis it is completely normal to scale Monte Carlo to data, since the estimated Monte Carlo rates are not perfect and it is their distributions that are compared, not the overall scale.

only, since they by far outweigh the Monte Carlo errors. In the bottom of the figure, the ratio of data to Monte Carlo rates are available.

The data is generally decently described by the scaled Monte Carlo distributions, but there are some areas in which the ratio of Data/MC is quite high. A χ^2 test, with null-hypothesis that the distributions are identical, results in a p-value of 0, which suggest that the data/Monte Carlo agreement is not perfect, especially when considering the Poisson errors.

This might perhaps be expected since the simulations aren't perfect. Overall the distributions agree well enough to assume that there are no substantial deviations. To select a relatively clean neutrino sample, a cut is made at 12 in the neutrino logit probability. This corresponds to 0.999994 in normal probability space.

This cut corresponds to a neutrino purity of 98.84% neutrinos with 1.16% muons and no noise in Monte Carlo. In data this cut yields a sample of 8753¹² events, which hopefully have approximately the same purity as in the Monte Carlo selection. To test if that is the case, we now split the neutrino samples from data and Monte Carlo in cascade and track neutrinos. We then investigate if the data and Monte Carlo distributions agree for track and cascade selections separately across reconstructed and calculated variables. It is the 130000 muon sample that is used in figure 7.21 and in the following analyses. This is partly due to the fact that only 3 muons in the low statistic muon sample 139008 makes it past the neutrino selection cut, which makes them difficult to include in the analysis. The much larger muon sample 130000 results in 347 muon passing the cut. Furthermore, the OscNext group uses the 130000 sample, which makes comparisons with their results more appropriate.

In appendix D.2, figure 7.21 is available without scaling of the MC distributions as well as only for track-like events, cascade-like events, upgoing events and downgoing events.

¹²Out of the approx. 62 million events we started out with.

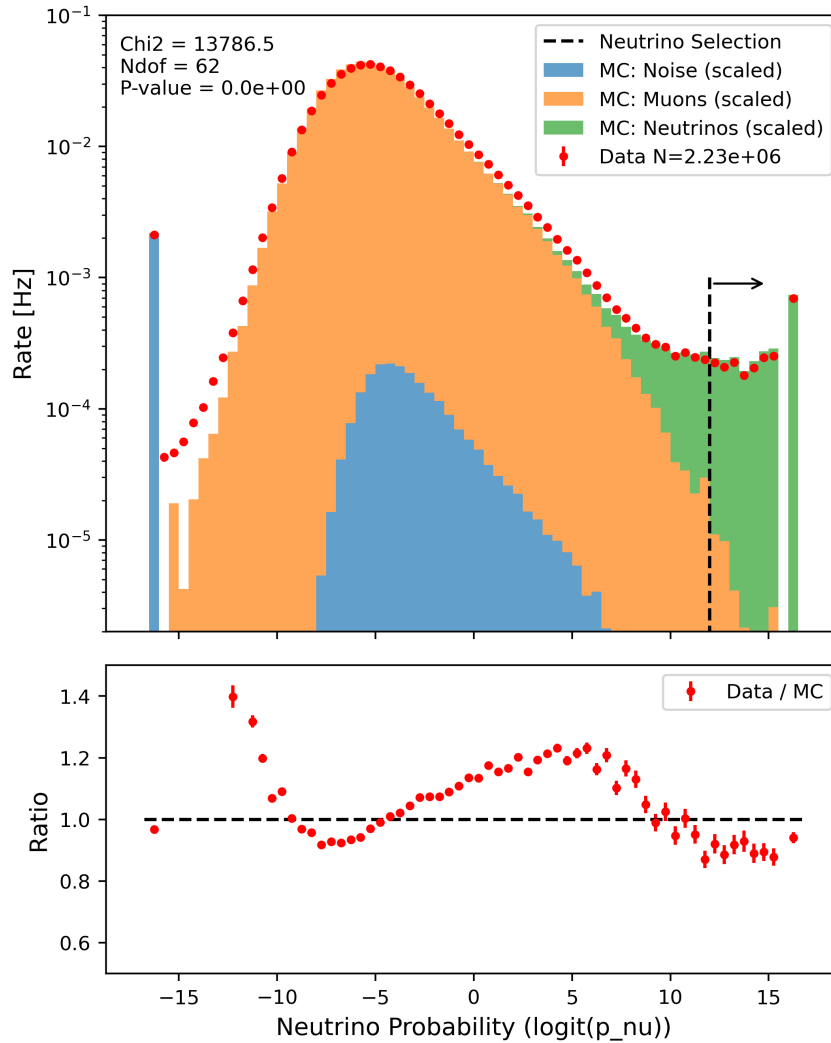


FIGURE 7.21: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. See text for more information. (Bottom) Ratio of data to Monte Carlo total rate. The plot shows that the data is decently described by the Monte Carlo distributions, although there are places where they deviate as seen in the ratio plot. The P-value of 0 also shows that the distributions are not identical.

The track and cascade neutrino selections are illustrated in figure 7.22, which is quite similar to that for neutrino probabilities in logit space. Here the Monte Carlo cascade and track neutrino distributions in track probability are scaled to match the distribution in data. The Monte Carlo track neutrinos are scaled by 1.057 and the Monte Carlo cascade neutrinos are scaled by 0.863. The muons that make it into the neutrino selection are also plotted, but are not scaled or included in the data/MC ratio, due to the relatively low statistics. As can be seen in the ratio plot, the distributions are very similar, although the ratio are generally above one in the first part, then generally a bit below in the second part, whereas the remainder looks very good.

A χ^2 test with null-hypothesis that the distributions are identical, results in a p-value of 0.03, which also suggest that the data/Monte Carlo agreement is quite good, although not perfect.

The track neutrino selection is set to be above 0.9 in track probability, whereas the cascade neutrinos selection is below 0.5 in track probability. This is not a rigorous choice, but the cuts were chosen to ensure that the track selection mainly consists of very track-like events, whereas the cascade selection contains events that are cascade-like. The comparisons for the neutrinos between 0.5 and 0.9 in track probability are also interested, but not included in this work due to time constraints. In appendix D.2, figure 7.22 is available without scaling of the MC distributions.

In figure D.19 in appendix D.3, the predicted track probabilities for noise and muon events in Monte Carlo and data is available. It shows that a general muon contamination would most likely show up in around 1 in track probability and noise would appear around 0-0.1 and 0.4-0.6 in track probability. That there are no major disagreements particularly in these areas, suggest that there are no dominating muon or noise contamination in the data neutrino selection. However, as can be seen in figure 7.22, the sneaky muons (those that make it into the neutrino selection) do not follow the general muon distributions exactly. Instead, a lot of the sneaky muons are more cascade like than muons in general. We will return to the properties of the sneaky muons in section 7.9.

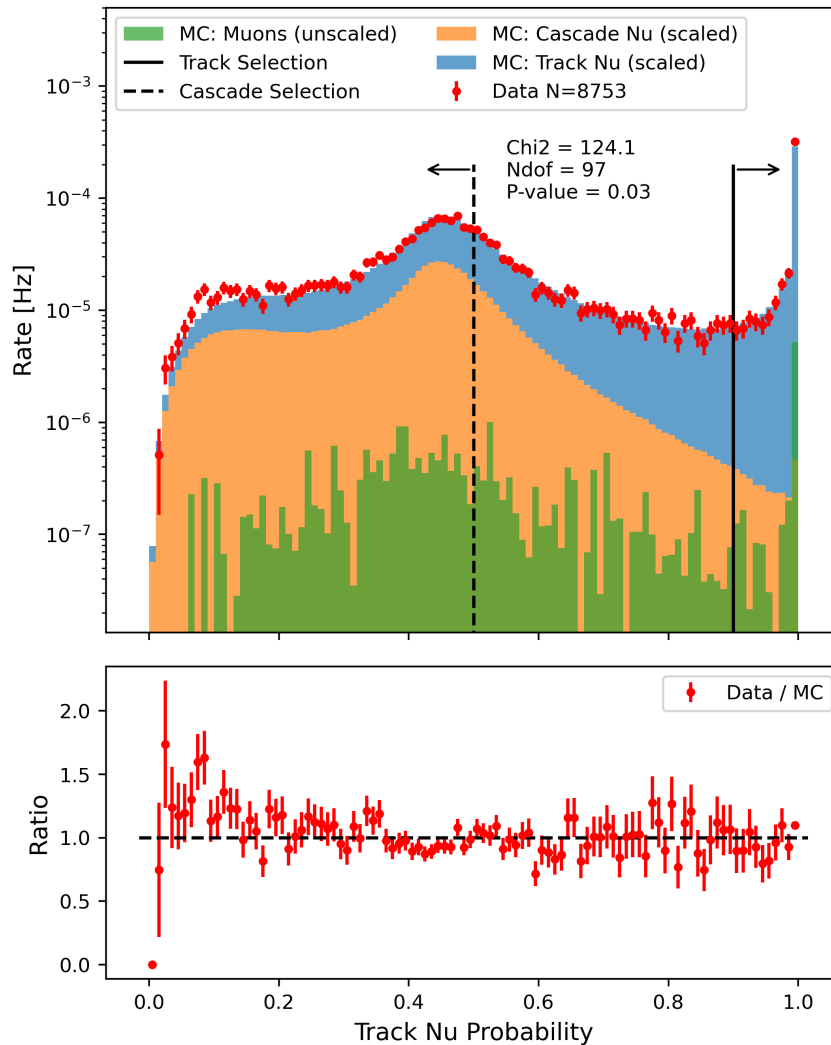


FIGURE 7.22: (Top) Stacked histogram of track neutrino probabilities for the Monte Carlo distributions of track and cascade neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. The surviving sneaky muons are also plotted, but not scaled or included in the ratio plot. (Bottom) Ratio of data to Monte Carlo total rate. The plot shows that the data is very well described by the Monte Carlo distributions, although there are areas where the ratio generally fall above or below, as seen in the ratio plot. The P-value of 0.03 also shows that the distributions are relative similar.

Having selected the events which are probable track and cascade neutrinos in Monte Carlo and data, it is time to determine if their distributions are similar across the following variables in table 7.7:

Variable	Explanation
Energy	Reconstructed energy
Zenith	Reconstructed zenith angle
Zenith std	Reconstructed zenith angle std
Azimuth	Reconstructed azimuth angle
Azimuth std	Reconstructed azimuth angle std
Position x,y,z	Reconstructed interaction vertex position x,y,z coordinate
Rho	Orthogonal distance to the 36th string which is located at (x,y) = (46.29,34.88)
Pulses	Number of total DOM hits in the event (from SplitInIcePulses pulsemap)
Unique DOMs	Number of unique DOMs being hit in the event (from SplitInIcePulses pulsemap)
Unique string	Number of unique strings being hit in the event (from SplitInIcePulses pulsemap)

TABLE 7.7: Variables used in comparison of track and cascade neutrino selection between data and Monte Carlo. Energy, zenith, azimuth, position z and rho are presented below, while the remainder can be found in appendix D.1.

In the following figures, 7.23, 7.24, 7.25, 7.26 and 7.27 there are four plots. The left two have to do with the track neutrino selections and the right two contain events from the cascade neutrino selections. The top plots are data and Monte Carlo distributions of the variable in consideration (including the truth for Monte Carlo when possible). The bottom contains a ratio of data to Monte Carlo. The rate of track neutrinos in Monte Carlo is normalised to the data track neutrino rate, being scaled by 1.099. Similarly the cascade neutrino rate in Monte Carlo are scaled by 0.966. A χ^2 test determines if the distributions for Monte Carlo and data are similar. A large p-value suggest that they are, whereas a low p-value corresponds to a disagreement between the distributions. The errors are again Poisson and only calculated for data. The number of degrees of freedom are the number of non-empty bins minus one from the scaling factor.

It is important to understand that these "neutrino" selections are not exclusively neutrinos. In Monte Carlo, they include those of the 347 muons that are track or cascade like. In data, it is the hope that there are mainly neutrinos in the sample, which is exactly what the following plots indicate.

In figure D.18 in appendix D.3 histograms of Monte Carlo and data noise and muons for most of the variables in table 7.7 are available. This could help determine if there are obvious muon or noise contamination in the data neutrino selection. For instance if there is an excess in the data neutrino selection with a zenith angle below π (coming from the southern sky), this would suggest that muons contaminate the sample. While the plot could help determine where contamination might show up, it should be kept in mind that the muons and noise that would enter the neutrino selections probably have different distributions than the typical muons and noise.

In figure 7.23 the reconstructed energy distributions for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the ratio plots in log space have linear slopes, which corresponds to an additive offset in energy. This means that the data events have a very similar shape to the Monte Carlo events, but shifted slightly in energy. This would be worrisome, were it not because the same trend has been observed in data - Monte Carlo agreement analyses in IceCube. As such, it is encouraging that the shape is similar, despite the fact that the distributions are not identical. However, the muon distribution in real data tends to have higher energies than the neutrinos, which means that the problem could also result from muon

contamination, although it is unlikely to be the main cause, since a corresponding disagreement is not seen in the zenith distributions.

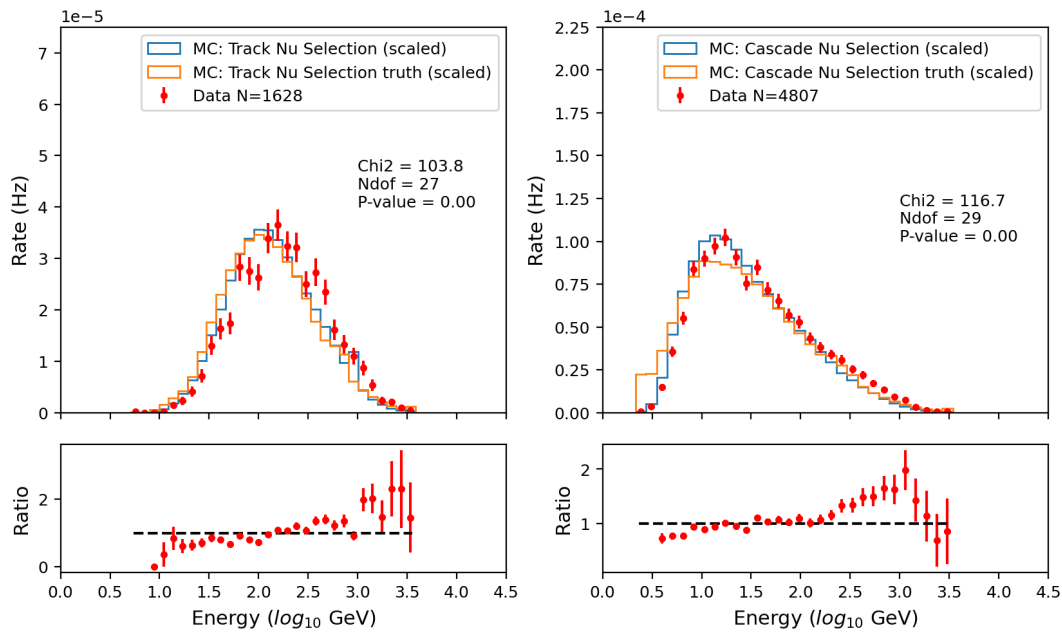


FIGURE 7.23: Energy distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. The distributions do not match perfectly and a linear relationship is visible in the ratio plots. For more information, see text.

In figure 7.24 the reconstructed zenith distributions for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar. Furthermore, for the track neutrino selections the χ^2 test even suggest that the fit is good with a p-value of 0.07. The Cascade ratio plots does seem to slope gently downwards, but overall looks pretty good too. Especially considering muons mainly come from a zenith angle below $\pi/2$, and there is no large excess in this range.

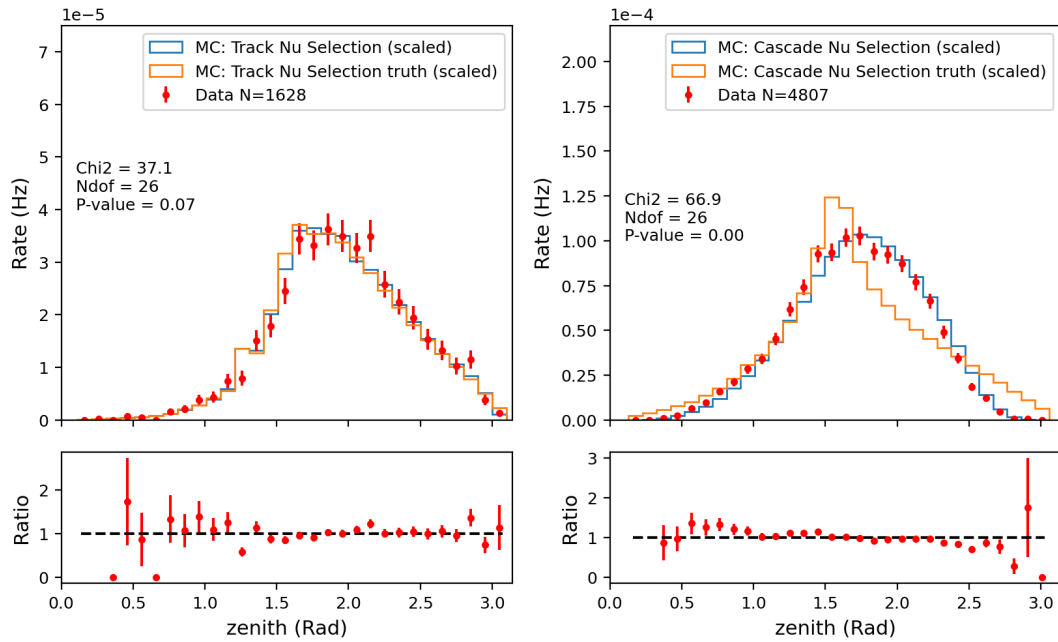


FIGURE 7.24: Zenith distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. The track distributions match well. The cascade distributions match well visually, but not statistically with a p-value of 0. For more information, see text.

In figure 7.25 the reconstructed azimuth distributions for track and cascade neutrino selections in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar and for both selections the χ^2 test suggest that the agreements are excellent with p-values of 0.19 and 0.46. This is perhaps also to be expected, given that potential muon contamination is expected to be more or less uniform in azimuth as well. A large noise contamination would tend to show up as an excess in data for azimuth in the range of 2-3.5, and a shortage around 4-2 π , which is not evident. The exact distributions of noise and muons can be seen in figure D.18 in appendix D.3.

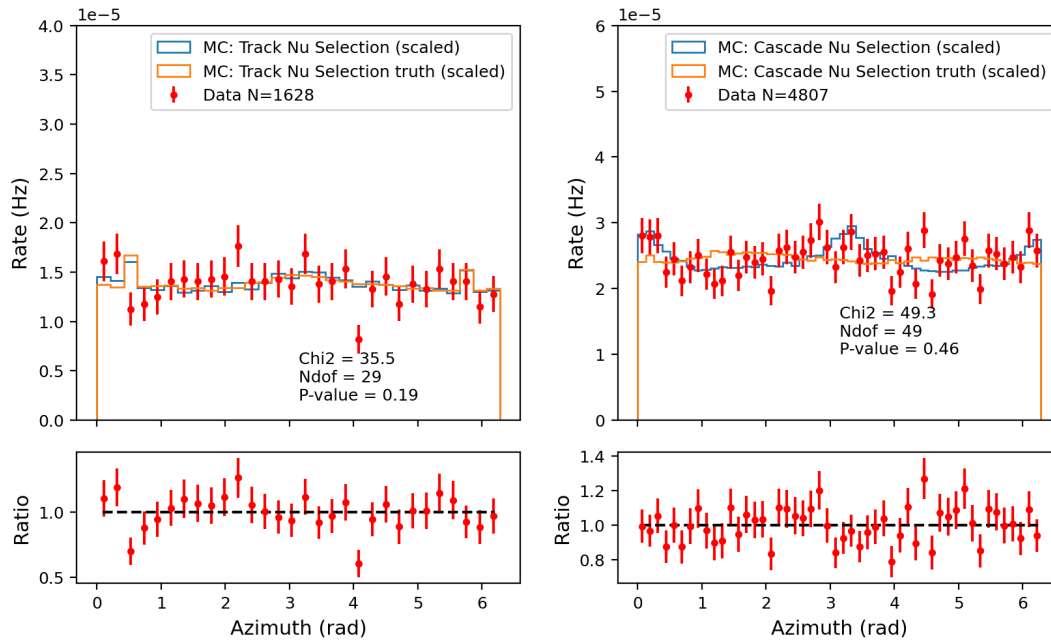


FIGURE 7.25: Azimuth distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions match very well. For more information, see text.

In figure 7.26 the distributions of reconstructed z coordinates for the interaction vertex for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar. For the track neutrino selection, the χ^2 test suggest that the distributions do not match exactly, with a p-value of 0. The χ^2 test for the cascade neutrino selection suggest that the distributions match with a p-value of 0.60.

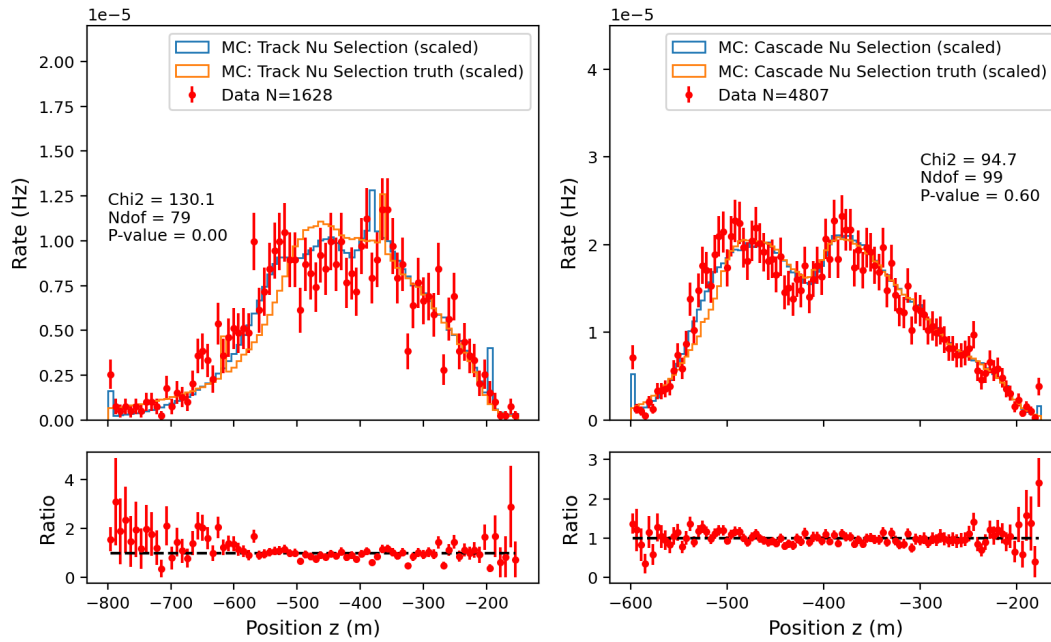


FIGURE 7.26: Z coordinate of interaction vertex position distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Track distributions do not match statistically, but the cascade distributions match very well. For more information, see text.

In figure 7.27 the distributions of rho for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar. For the track neutrino selection, the χ^2 test suggest that the distributions match well, with a p-value of 0.22. The χ^2 test for the cascade neutrino selection suggest that the distributions do not match perfectly, with a p-value of 0.01.

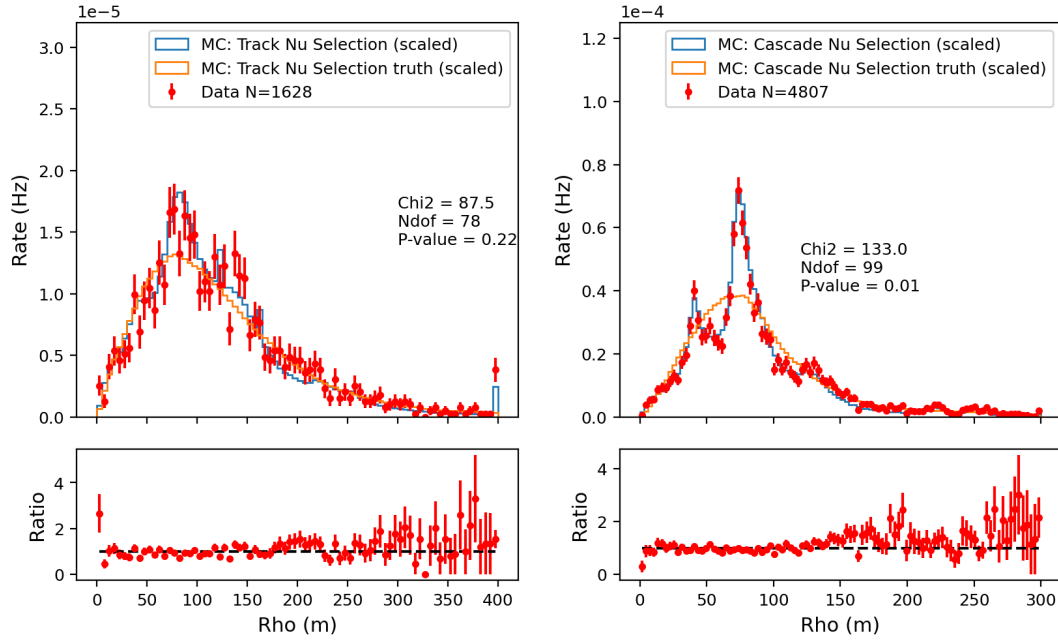


FIGURE 7.27: rho (orthogonal distance to string 36, approximately center of detector) distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Track distributions match decently and cascade distributions match very well. For more information, see text.

Having seen that the neutrino selections look very similar in energy, angles and vertex positions, we investigate the energy further in the following three plots. To make the comparison easier to visualize, the energies of events in the Monte Carlo track neutrino selection are scaled by 1.249 and those in the Monte Carlo cascade neutrino selection are scaled by 1.247. This ensures that for each selection the overall mean energy of Monte Carlo and data is the same, allowing us to look for differences in the energies as a function of zenith, azimuth and z position.

The errorbars are calculated using the error on the mean function: $std_{\text{mean}} = \frac{std_{\text{individual}}}{\sqrt{N}}$, where N is the number of events contributing to the mean. For each bin in zenith, azimuth and z position, the mean of the energies for the events in the bin is taken. The $std_{\text{individual}}$ is estimated by taking the standard deviation of the energies of the same events. This method is only valid for bins with reasonably many events. For instance it gives an std of 0 if only a single event falls in a bin.

The mean energy as function of zenith distributions agree quite well between Monte Carlo and data, except for a few very low statistic bins on the edges.

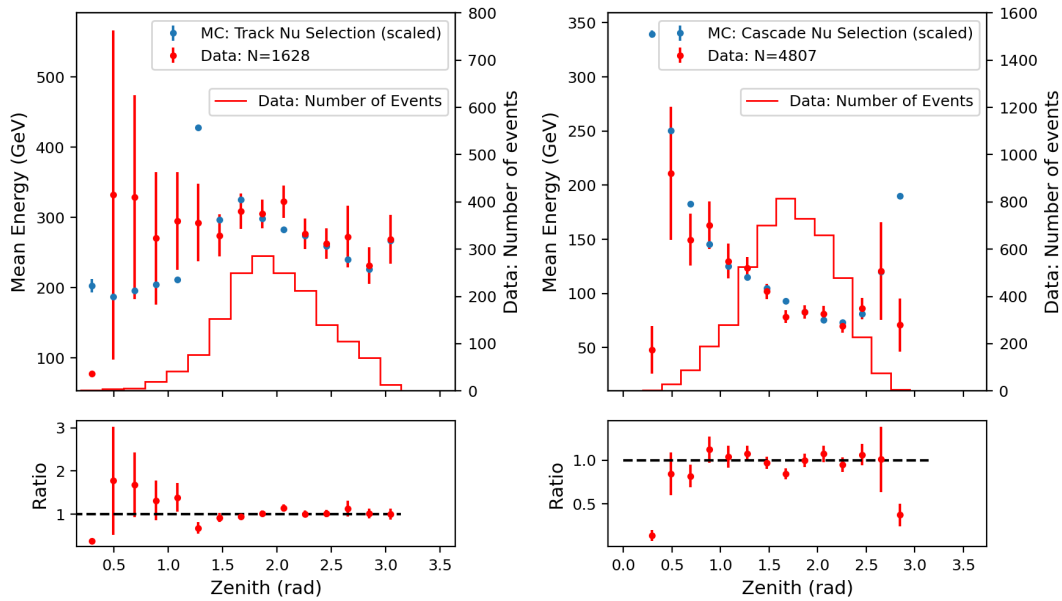


FIGURE 7.28: Distributions of mean energy as a function of zenith for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. For more information, see text.

The mean energy as function of azimuth distributions also agree quite well between Monte Carlo and data.

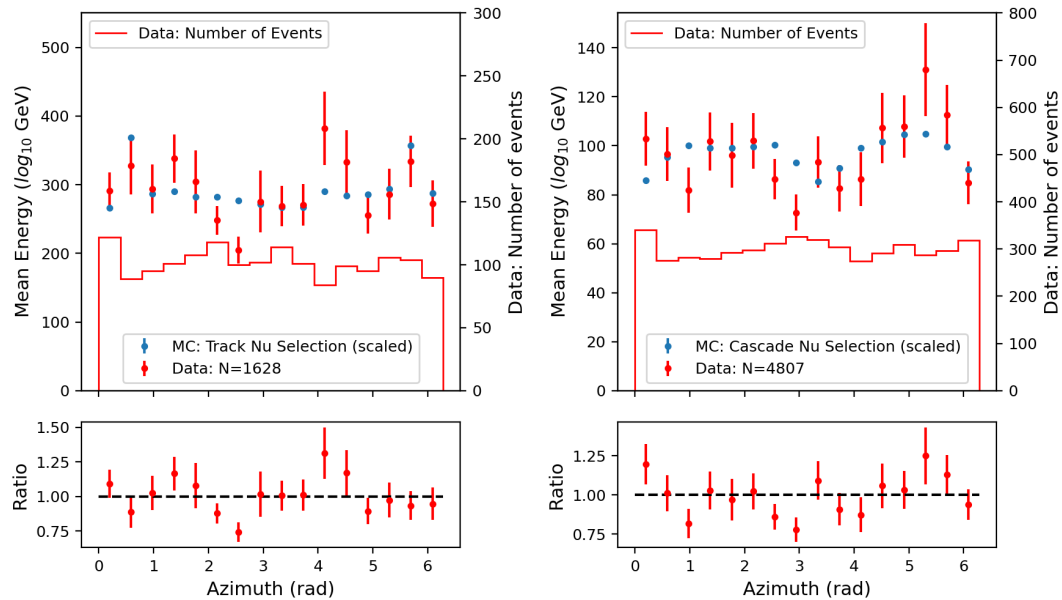


FIGURE 7.29: Distributions of mean energy as a function of azimuth for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. For more information, see text.

The mean energy as function of z position distributions also agree quite well between Monte Carlo and data, except for the low statistics bins.

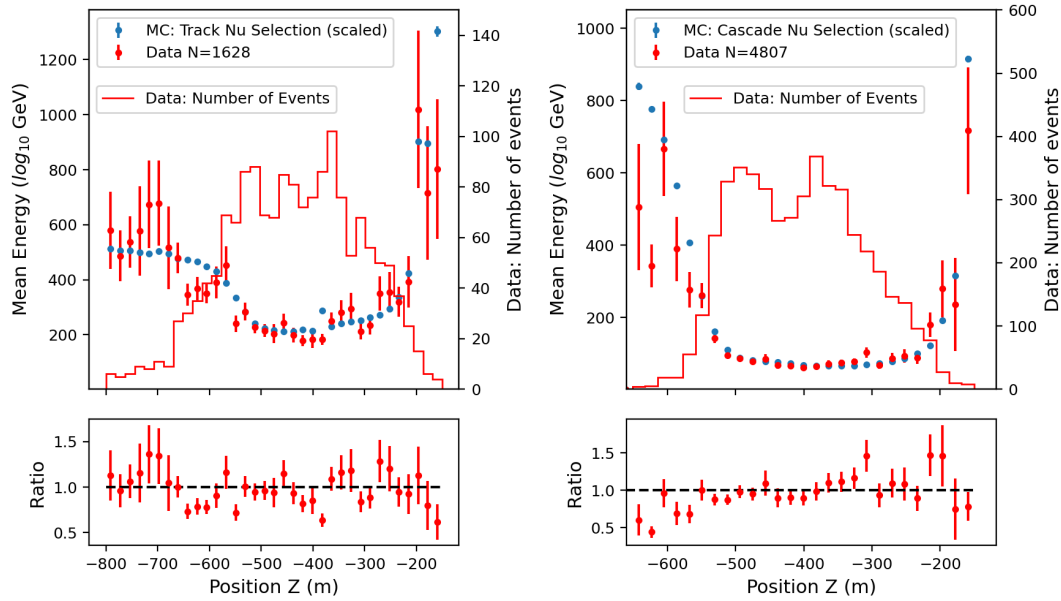


FIGURE 7.30: Distributions of mean energy as a function of z position for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. For more information, see text.

Thus also when investigating the energy dependence on a few of the other key variables, it seems as if the events in data are similarly distributed as those in Monte Carlo. This is further argument that the data events are mainly neutrinos.

Considering the agreement between data and Monte Carlo for the track and cascade neutrino selections across the reconstructed and calculated variables, it does indeed seem as if the data selection is composed mainly of neutrinos. There is naturally not always a complete statistical agreement, but this is not necessarily to be expected, given that Monte Carlo simulations are not perfect. Another way to investigate if the data neutrino selection is reasonable, it is compare the neutrino rate to the expected in Monte Carlo and to the one obtained by the existing methods in the OscNext group. That will be the topic of the next section.

7.7 Comparison Of GraphNet Neutrino Selection With The OscNext Selection In Monte Carlo Data

7.7.1 How Many Additional Neutrinos Do We Select In Comparison To OscNext In Monte Carlo data?

The previous section suggests that DynEdge is able to select a clean neutrino sample, not only in Monte Carlo, but also in data. As such, it becomes interesting to investigate the amount or rate of neutrinos in the DynEdge selection in comparison to the existing OscNext neutrino selection. In figure 7.31, a replication of the OscNext particle rates as function of level in the cleaning process is available. It generally agrees well with figure 4.12. The additional step beyond level 7 comes from a final muon

cut¹³, which is actually a part of the level 7 cut. Finally, there are additional analysis specific cuts applied afterwards, but these are not specifically targeted at removing muons or noise. As such, one appropriate place to compare our neutrino selection is after the OscNext final muon cut.

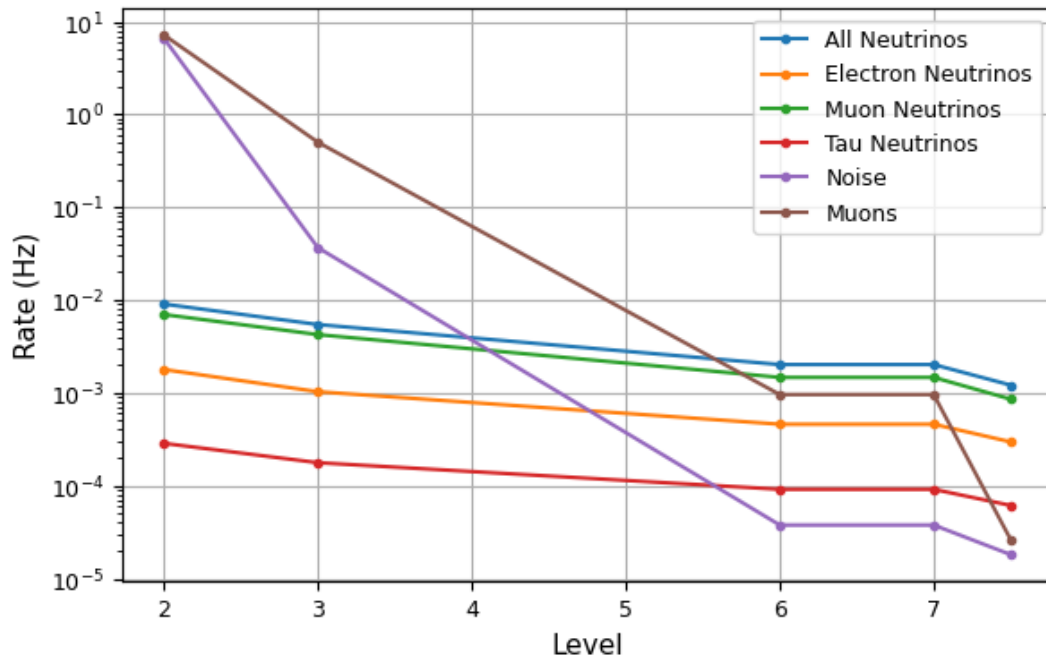


FIGURE 7.31: Monte Carlo rates for each particle type individually at each level in OscNext selection process. Remake of figure 4.12 to be compared against GraphNeT selection.

The DynEdge version of the rate as function of cleaning level plot is available in figure 7.32. It shows the rate of each particle type that survives a particular cut in the neutrino probability logit space. Note that the muons in this figure is from sample nr: 130000, which is the one OscNext uses in their analysis, making the comparison more appropriate than using the muon sample nr: 139008. In appendix E.1 identical plots for track-like events with track probability above 0.9, and cascade-like events with track probability below 0.5 in can be found in figures E.2 and E.1. Furthermore, similar figures using only up or downgoing events is available in figures: E.3 and E.4.

It is evident in the figure that the rate of particles being sorted away as function of the cut, is different in data than Monte Carlo. The cut is stricter in data than in Monte Carlo, which is a good sign in terms of ensuring a clean neutrino sample in data. Furthermore, as mentioned earlier, before any neutrino selection is made, the data rate is 20% above the Monte Carlo rate. This plots illustrates that the rate difference arises mainly from noise and muons. Around the point where neutrinos begin to dominate in Monte Carlo (4-10 in logit space), the ratio of data to Monte Carlo quickly drops to around 1.

¹³It is a cut in the neutrino probability of a BDT model, which is illustrated in figure E.5 in appendix E.1.

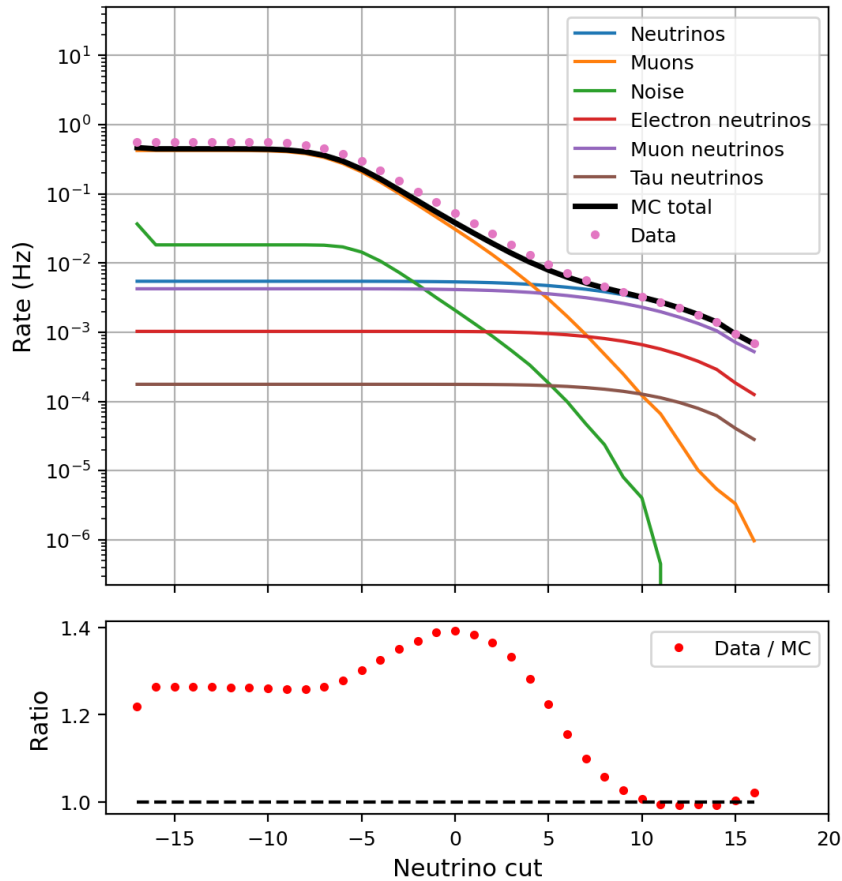


FIGURE 7.32: (Top) Survival rate of each particle type in Monte Carlo depending on where the cut is made in logit neutrino probability space. Also plotted are the survival data rate. Note that the muons are from sample nr: 130000. (Bottom) ratio plot of total data rate to Monte Carlo rate.

It can be difficult to compare the two above figures. Instead table 7.8 shows the rates and percent of each particle types for the DynEdge and OscNext selections. Green is when there should be mainly neutrinos in the selections, yellow is after most of the analysis cuts and red represents the stated final selection from OscNext. The table shows that DynEdge improves the total rate by 79% (from 1.257 mHz to 2.250 mHz) in comparison to the OscNext selection after their lvl 7 classifier. At the same time the purity increases from 96.48% neutrinos in OscNext selection to 98.84% in the DynEdge selection. DynEdge also manages to remove all noise events, whereas OscNext has 1.44% noise.

The uncolored rows illustrate the rates of particles in a DynEdge selection based on a cut in logit space of either 8 or 10. They indicate that 12 is a decent place to make the cut, since there are a substantially higher percent of muons in the looser selections¹⁴.

¹⁴It is possible that the looser selections are interesting for other analyses where muon contamination matters less. For instance in early warnings.

Rate comparison in mHz	Total	Muons	Noise	All Neutrinos	Real Data	Data/MC ratio
DYNEDGE: Logit 8 cut.	4.359	0.801 (11.01%)	0.024 (0.54%)	3.855 (88.44%)	4.607	1.057
DYNEDGE: Logit 10 cut.	3.217	0.122 (3.78%)	0.004 (0.13%)	3.091 (96.09%)	3.240	1.007
DYNEDGE: Logit 12 cut.	2.250	0.026 (1.16%)	0 (0%)	2.224 (98.84%)	2.235	0.993
OSCNEXT: After lvl 7 classifier.	1.257	0.026 (2.07%)	0.018 (1.44%)	1.212 (96.48%)	?	?
DYNEDGE: Logit 12 cut and zenith, energy, containment cuts.	1.382	0.010 (0.70 %)	0 (0%)	1.373 (99.30%)	1.289	0.932
OSCNEXT: After lvl 7 classifier and zenith, energy, containment cuts.	0.818	0.014 (1.70%)	0.011 (1.29%)	0.794 (97.0%)	?	?
OSCNEXT: Stated final selection.	0.715	0.005 (0.7%)	0 (0%)	0.705 (99.3%)	?	?

TABLE 7.8: Table showing the rate of each particle type in different DynEdge and OscNext neutrino selections. White rows indicate looser DynEdge selections, which are more polluted by muons. Green rows indicate a comparison at a point where the selections should contain mostly neutrinos. Yellow rows are the green selections but after most of the analysis cuts, which are introduced further down. Red is the final stated selection of OscNext. At both green and yellow comparisons, DynEdge gives a larger rate (approx 70-80%) at a higher purity. All rates are in mHz and the percentages are of the total rate in the particular selection.

While DynEdge is able to select a much larger sample of clean neutrinos, it is possible that the additional neutrinos are of "poor quality". This means that they for instance interact in the edges of the detector, or mainly come from the Southern sky. To check that this is not the case, a comparison after the majority of analysis cuts are also available in table 7.8. The cuts remove events where:

$$\text{Cos}(\text{zenith}) < 0.3$$

$$\text{Energy} < 5\text{GeV}$$

$$\text{Energy} > 300\text{GeV}$$

$$\text{Positionz} > -200\text{m}$$

$$\text{Positionz} < -500\text{m}$$

$$\text{Rho} > 300\text{m}$$

At this point, the Dynedge selection still improves the overall rate from OscNext's 0.818 mHz to 1.382 mHz (an increase of 68.9%). Simoultaneously the purity of the DynEdge selection is 99.30% neutrinos, in comparison to OscNext's 97.0% neutrinos. The cuts are made in the OscNext selection using their retro reconstructions, whereas it is made in the DynEdge selection using the GNN reconstructions. This makes the comparison slightly less valid, since it might introduce biases from the reconstruction techniques. Besides from this, it is still encouraging that DynEdge can select a much larger rate of neutrinos with a higher purity than the OscNext method. There is a final pulse quality cut applied in OscNext, which was not replicated in this work. When it is included, the OscNext purity increases to 99.3%, the same as the DynEdge purity before the final pulse quality cut.

In table 7.8, it can also be seen that the DynEdge neutrino selections in Monte Carlo and data have approximately matching rates. In the green selections, the data to Monte Carlo ratio is 0.993 and in the yellow selections it is 0.932. These ratios also suggest that the neutrino selection in data is reasonable and could contain a pure neutrino sample.

7.7.2 How Are the GraphNeT And OscNext Monte Carlo Neutrinos Distributed?

To visualize the DynEdge and OscNext Monte Carlo neutrino selections, below is a comparison of their distributions in energy, zenith, azimuth, vertex z position, rho and neutrino logit probability. The comparison is made before the analysis cuts are applied (green selections in table 7.8) and it is the truth values that are plotted, NOT the predictions. The illustration in figure 7.33a visualizes the neutrino selections and their overlap.

Figure 7.33b shows that all shared and DynEdge Monte Carlo neutrinos are above neutrino logit probability of 12, whereas the neutrinos OscNext manages to capture, which DynEdge does not, naturally all have neutrino logit probabilities below 12.

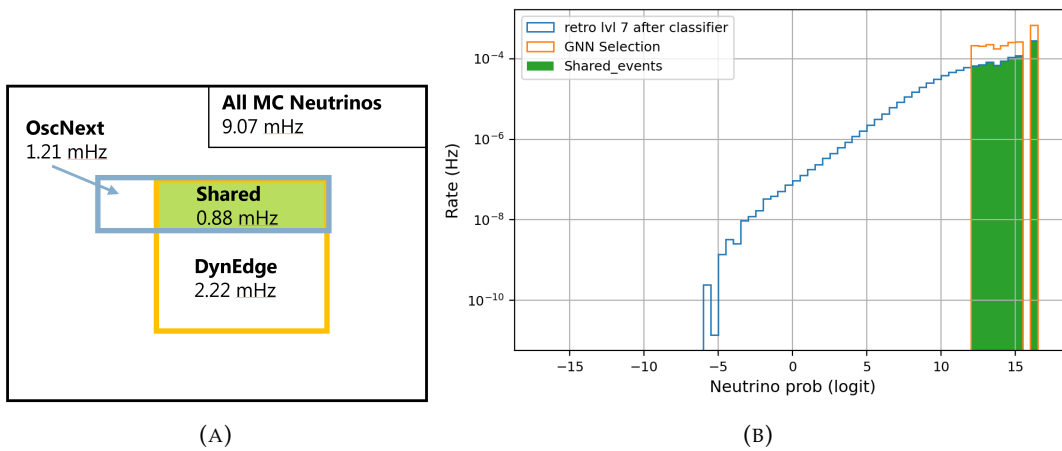


FIGURE 7.33: (A) Illustration of Monte Carlo neutrino selections for DynEdge and OscNext. (B) Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. GraphNet reconstructed Logit neutrino probability histograms of shared events and of the two selections are plotted.

In figure 7.34 the energy distributions are shown. It can be seen that the excess DynEdge neutrinos outnumber the OscNext neutrinos everywhere, but especially at high energies. However, most of the excess DynEdge neutrinos are in the OscNext range, which is good.

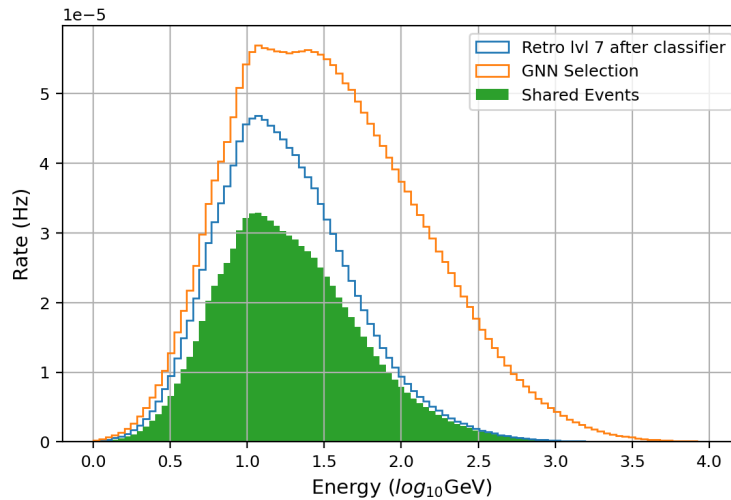


FIGURE 7.34: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Energy histograms of shared events and events in the two selections are plotted.

In figure 7.35a the zenith distributions are shown. The thing to notice is that most of the excess DynEdge neutrinos comes from the Northern sky, which is good. However, some of these neutrinos are removed by containment cuts ensuring that the neutrinos interact around DeepCore and not below the detector for instance. However, it is still encouraging that the DynEdge additional neutrinos are not mainly from the Southern sky, where OscNext implements the cut of $\cos(\text{zenith}) < 0.3$.

In figure 7.35b the azimuth distributions are shown. There is not really any interesting takeaways here.

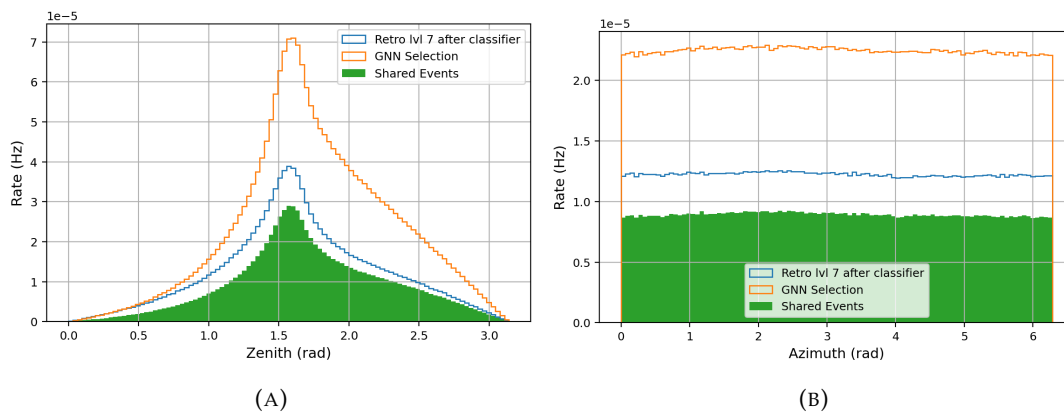


FIGURE 7.35: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Zenith (A) and azimuth (B) histograms of shared events and events in the two selections are plotted.

In figure 7.36a, the vertex position z coordinate distributions are shown. It shows that the GNN selection is higher at all places. While the GNN selection has more events below the detector, it also has more in the critical region around DeepCore.

In figure 7.36b, the ρ distributions are shown. The GNN selection is quite a lot higher in the edges of the detector (high ρ), but also substantially higher in the

center region (low rho). Distributions for the vertex position x and y coordinates give similar results and can be seen in appendix E.2¹⁵.

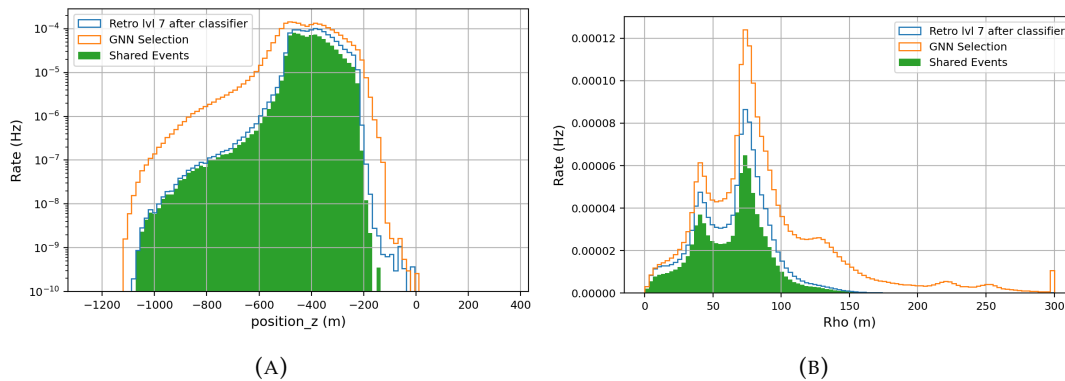


FIGURE 7.36: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Interaction vertex z position (A) and ρ (B) histograms of shared events and events in the two selections are plotted.

Thus it seems that the improvement in neutrino rates in Monte Carlo data using the DynEdge selection survives most of the analysis cuts, but how easy are the additional neutrinos to reconstruct?

7.7.3 What Are The Quality Of The Neutrinos In The GraphNeT Selection Compared To The OscNext Selection?

The additional neutrinos in the GNN selection could potentially be much harder to reconstruct. This can be tested by plotting the distributions of the energy and angular residuals, for the GNN selection, the OscNext selection and the shared selection in Monte Carlo, since this can not be done in real data. The three resulting plots are available in figures 7.37 and 7.38. Importantly the residual distributions are all calculated using the GNN predictions (Not the retro prediction for OscNext). Therefore the quality comparison only investigates how well the neutrino selections are to reconstruct. If the OscNext predictions were used to calculate the residuals in the OscNext selection, the difference in reconstruction performance between retro and the GNN would make the results harder to interpret. The neutrino selections plotted are before the analysis cuts introduced before (green selection in table 7.8), but identical figures with the distributions after the analysis cuts (yellow selection in table 7.8) are available in figures E.10 and E.11 in appendix E.2. The figures of the distributions after the analysis cuts in appendix show more or less the same behavior as the figures of the distributions without the analysis cuts below.

In figure 7.37, the GNN and OscNext energy residual distributions are available along with the ratio between the GNN and OscNext rates. It can be seen that the GNN selection also has a lot more events with a low energy residual than the OscNext selection.

¹⁵In the same place there are also plots that show the neutrino selections differently for energy and vertex position coordinate z . They have a shared distribution and the exclusive OscNext and GNN distributions. There are no new information, but it is easier to see exactly where the additional neutrinos in the GNN selection are distributed.

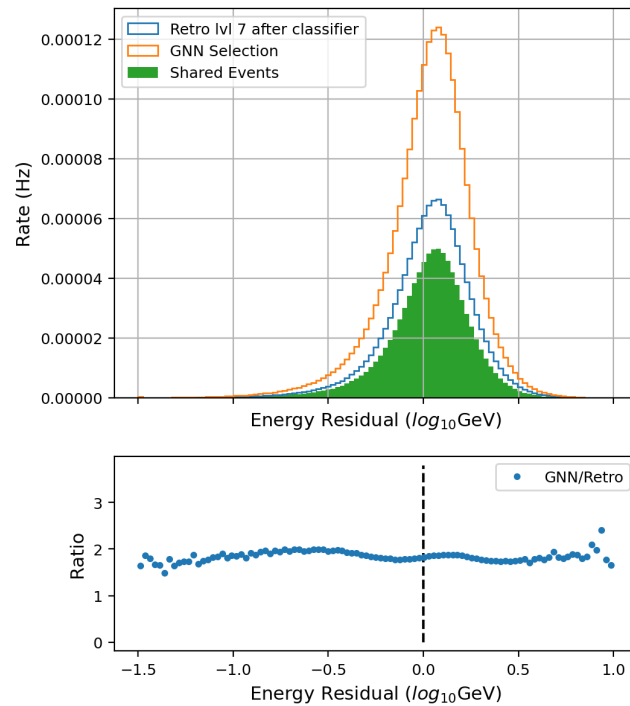


FIGURE 7.37: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. (Top) Energy residual histograms of shared events and events in the two selections are plotted. (Bottom) Ratio of GraphNeT rate to OscNext rate is plotted. For more information, see text.

In figure 7.38, the GNN and OscNext zenith and azimuth residual distributions are available along with the ratio between the two. It can be seen that the GNN selection has an even larger improvement over OscNext for events with low angular resolution than in general. This bodes positively for the impact of the additional neutrinos in the GNN selection on the oscillation analysis.

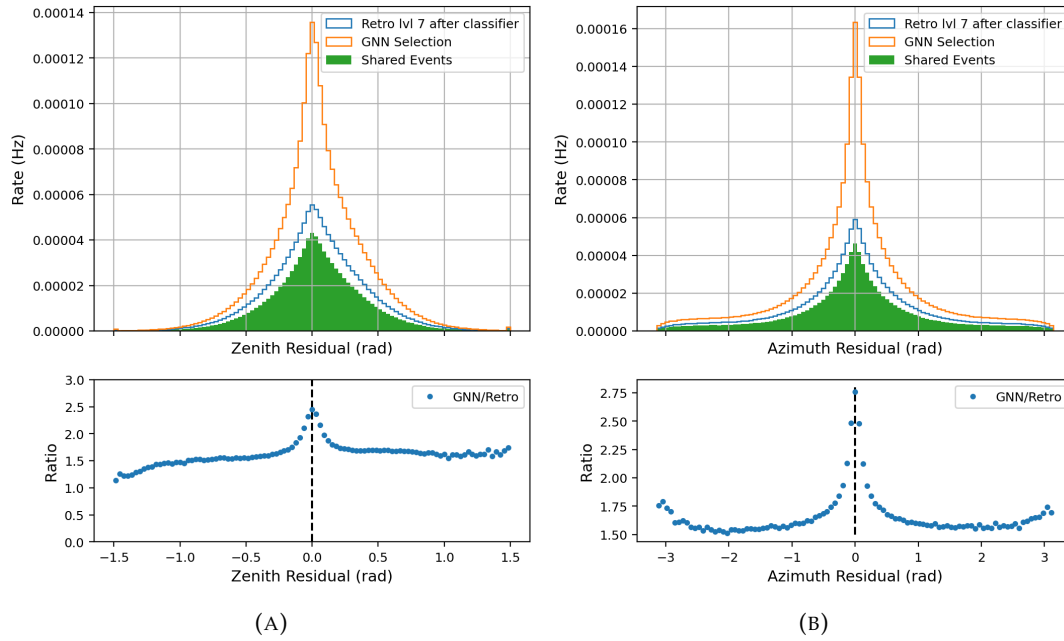


FIGURE 7.38: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Zenith residual (A) and azimuth residual (B) histograms of shared events and events in the two selections are plotted. (Bottom) Ratio of GraphNeT rate to OscNext rate is plotted. For more information, see text.

High energy neutrinos are generally easier to reconstruct accurately. Thus it is possible that our additional neutrinos have lower residuals, simply because they have higher energies. To investigate this, the above figures of the residuals are also plotted with the analysis cuts and an even stricter energy cut. From investigating the oscillation plot in figure 4.1, the strict energy cut removes neutrinos with an energy higher than $10^{1.75}$ GeV = 56.23 GeV. The figures showing the residuals of these distributions are available in figures E.12 and E.13 in appendix E.2. They support the argument that the lower residuals in our neutrino selection compared to the OscNext selection come mainly from the high energy neutrinos. The quality in zenith and energy reconstruction are approximately the same in the additional neutrinos as in the OscNext, but the quality of the azimuth reconstruction is lower for the additional neutrinos. Furthermore the overall rate improvement in Monte Carlo is reduced to 47.8% with the stricter energy cut.

Having investigated the rate and quality of neutrinos in the DynEdge selection compared to OscNext, lets have a look at the actual data neutrino signatures in the detector.

7.8 Real Neutrino Events In Real Data

One thing is to show that the neutrino selections in data and Monte Carlo have the same distribution in reconstructed and calculated variables. However this does not prove that the data events are actually real neutrinos. To make this even more probable, one can investigate the actual detector signals and see if they appear to behave as neutrinos would (a track or cascade shape appearing within the detector, or potentially from outside the detector). Most muons should appear from the

top or edge of the detector and make a track shape downwards. However, some muons pass through corridors of thinly instrumented Ice, meaning their signal appears inside the detector like neutrinos. These sneaky muons are illustrated in the next section, section 7.9. Noise would appear as a diffuse scattering of DOM hits, which can sometimes resemble a low energy neutrino event.

In figure 7.39, the detector readout is presented for one of the events with the highest predicted energy (2883 GeV) in the data track neutrino selection. It clearly appears to be a real track neutrino, since it has an elongated shape, with the earlier DOM hits below the later DOM hits. This indicates that it is travelling upwards, and rules out the possibility that it is a muon.

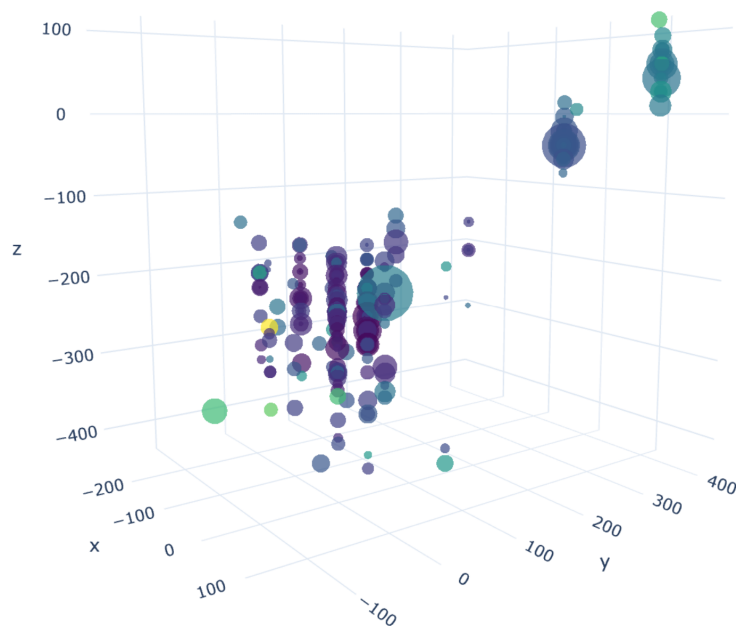


FIGURE 7.39: Event signature for the event with the highest predicted energy in the data track neutrino selection, event number 90278349. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTInIcePulses is used.

In figure 7.40, the detector readout is presented for one of the events with the highest predicted energy (2796 GeV) in the data cascade neutrino selection. It clearly appears to be a real cascade neutrino, since it has a sphere shape. Furthermore at this energy level, it is very unlikely that a muon could make such a signature, since even if it travelled through a corridor, it would emit light in amounts that would still be detected. This is supported by the fact that most "sneaky" muons have predicted energies below 300 GeV, although a few have predicted energies as high as 800 GeV.

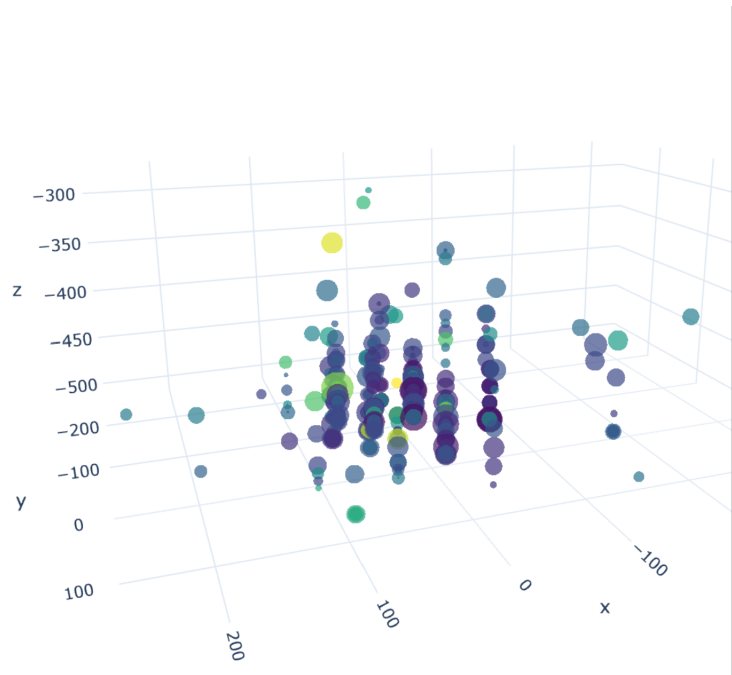


FIGURE 7.40: Event signature for the event with the highest predicted energy in the data cascade neutrino selection, event number 34882051. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTInIcePulses is used.

Additional detector signature plots are available in appendix E.3. There the three most energetic and least energetic events in the data track neutrino selection are presented along with three random ones as well. Similarly for the data cascade neutrino selection. There are no obvious muons or noise in said event signatures either.

7.9 Sneaky Muons That Make It Though To The Neutrino Selection In Monte Carlo

Three muons from the muon sample 139008 make it into the Monte Carlo cascade neutrino selection. These are what I refer to as "sneaky" muons. They are also called corridor muons because they travel through corridors of thinly instrumented ice, which can be seen in figures 7.41c, 7.42c and 7.43c. As such they only deposit light in the center of the detector around DeepCore, similar to neutrinos. This can be seen in figures 7.41a, 7.42a and 7.43a. The properties of the three muons are available in table 7.9. They have energies in the range of 200-300 GeV, but are predicted to have energies around 40-80 GeV, which is possibly because they only hit DOMs in the center bottom part of the detector.

The three sneaky muons are all naturally all coming from the south with zenith angles below $\pi/2$. They have low track probabilities, which sounds odd since they are muons and should produce track signatures in the detector. But this happens because once the muon light start hitting DOMs, they only hit those in a small sphere

around DeepCore. Finally, only a single of the three muons make it past the analysis cuts. The two others are cut away in the $\cos(\text{zenith}) < 0.3$ requirement.

Sneaky Muons	1	2	3
Event Number	9236982	36628082	37560539
Make it pas analysis cuts?	No	No	Yes
Rate (mHz)	0.022	0.022	0.007
Energy (GeV)	219.17	291.45	221.86
Energy Prediction (GeV)	40.19	76.57	78.91
Track Probability	0.35	0.19	0.16
Zenith (rad)	0.90	1.11	1.25
Zenith Prediction (rad)	0.92	0.97	1.42
Azimuth (rad)	0.124	5.86	2.29
Azimuth Prediction (rad)	0.148	5.35	1.65
Neutrino Probability	1.0	0.999999	0.999995
Neutrino Probability Logit	14.9	14.2	12.28

TABLE 7.9: Sneaky Muon properties.

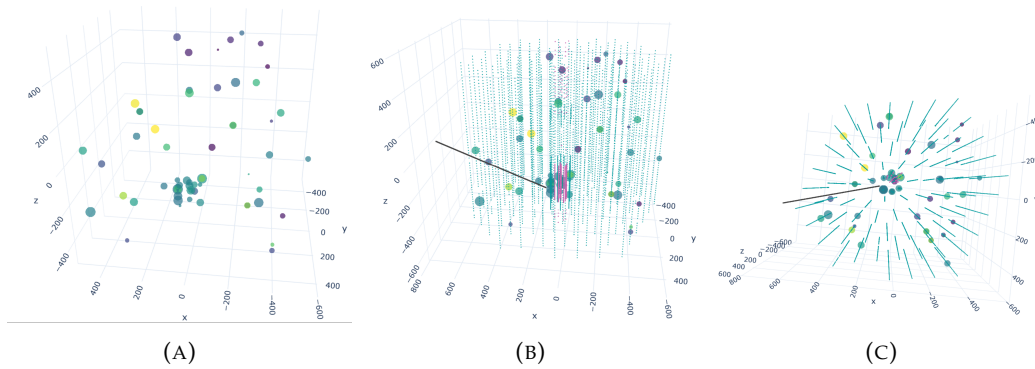


FIGURE 7.41: Detector signature for sneaky muon with event number 9236982. (A) From side without DOMs and track. (B) From side with DOMs and track. (C) from top with DOMs and track. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The un-cleaned pulsemap InIcePulses is used.

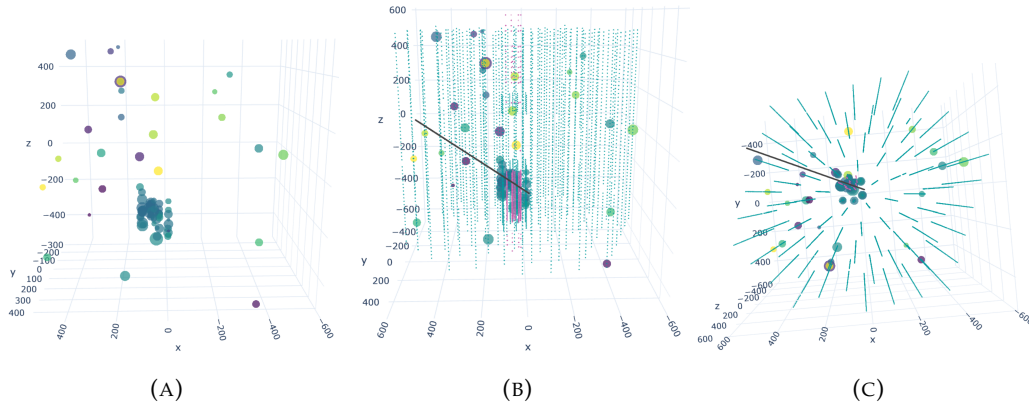


FIGURE 7.42: Detector signature for sneaky muon with event number 36628082. (A) From side without DOMs and track. (B) From side with DOMs and track. (C) from top with DOMs and track. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The un-cleaned pulsemap InIcePulses is used.

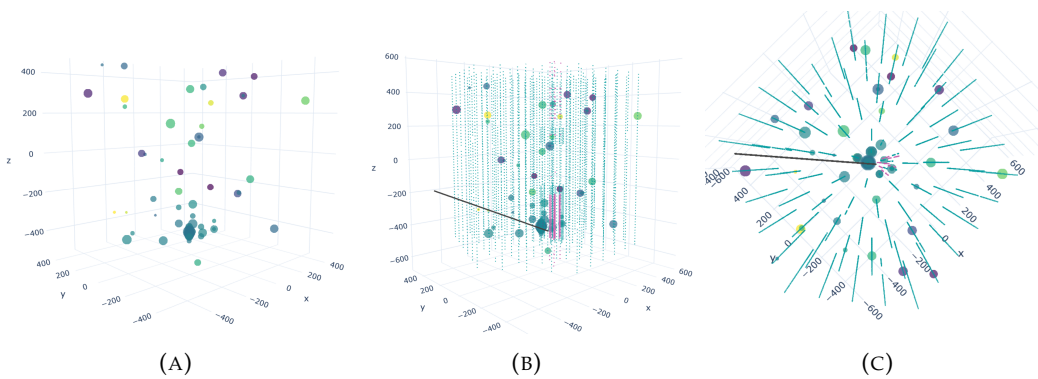


FIGURE 7.43: Detector signature for sneaky muon with event number 37560539. (A) From side without DOMs and track. (B) From side with DOMs and track. (C) from top with DOMs and track. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The un-cleaned pulsemap InIcePulses is used.

An analysis of the 347 muons from muon sample 130000 that are included in the neutrino selection give similar results. Their distribution in various reconstructed variables are plotted in figure 7.44. The truth distributions are shown as well. Obviously the muons all come from the Southern sky, but about half of them are reconstructed to arrive from through the earth. Their energy is also generally reconstructed to be much lower than the truth. It also appears as if there are two specific azimuth angles that are particular good to sneak through, as seen from the two substantial peaks in the azimuth distributions. The sneaky muons are of interest in the OscNext group, given that they are the most difficult muons to get rid of and they have a negative impact on the oscillation analysis.

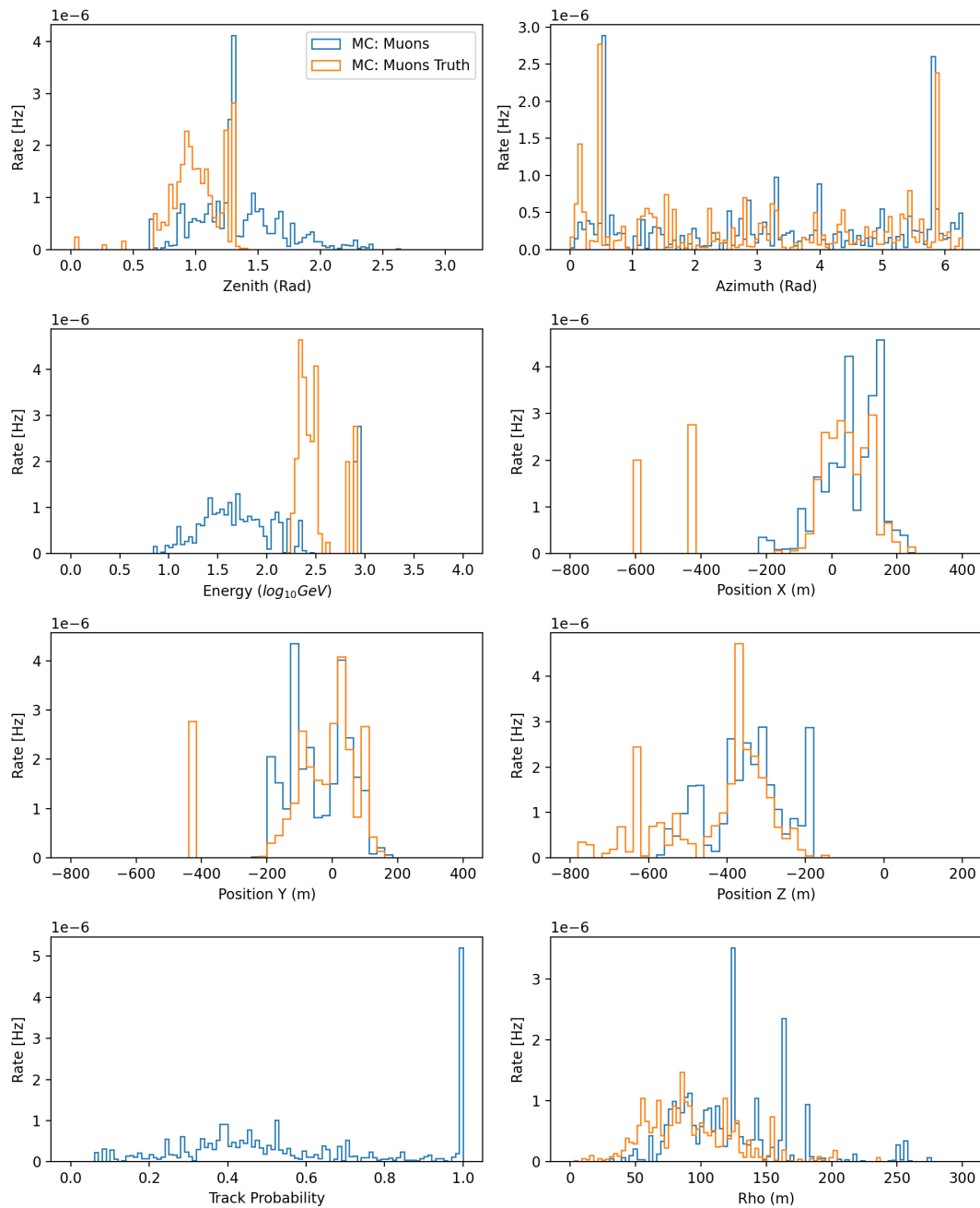


FIGURE 7.44: Histograms of the muons from sample 130000 that make it into the neutrino selection are plotted for various variables. The truth values are also plotted.

The investigation of the sneaky muons concludes this chapter and the analysis of classifying neutrinos using GraphNeT in actual data.

Chapter 8

Conclusion and Potential For Future Work

8.1 Improving The Reconstruction Of High Energy Northern Tracks

DynEdge is optimised for low energy reconstruction, where it outperforms the Retro reconstructions. Unfortunately the simple attempts at improving its performance on high energy northern tracks presented in this work failed. DynEdge simply is unable to compete with SplineMPE as is. While these are slightly disappointing results, they do pave the way for further optimisation, since future master students do not need to start from scratch.

Having tried adjusting simple parameters such as the Pulsemap, learning rate and prescaling, it is my belief that more complex, larger changes are required. These could be combinations of the attempted adjustments or for instance a better learning rate scheduler, another message passing scheme, larger dimensionality in the layers or a much larger training sample. Another promising avenue is the current effort to implement the winning model from the recent Kaggle IceCube competition, which showed that a combination of a transformer and a GNN is very effective[4].

While the first part of the results could have been better, the second part is more uplifting.

8.2 Classification And Evaluation Of Neutrino Sample In Real Data Using GraphNeT

8.2.1 Conclusion

The application of GraphNeT using the DynEdge model to classify and reconstruct the first clean low energy neutrino sample from actual raw lvl 2 data using a GNN seems to be successful.

The DynEdge reconstruction models perform very well in Monte Carlo, even when trained on lvl2 + DC events and the uncleaned pulsemap SplitInICEPulses. Furthermore, no hyperparameter optimisation was applied beyond those presented in [1] and the reconstructions still performed nicely when compared to Retro on the sub-sample of events Retro reconstructs. It was already shown in the published paper that GraphNeT outperforms Retro [1], which means that the comparison on reconstructions are secondary in this work.

The DynEdge multiclassification model works excellently in Monte Carlo data. It also represents a simpler method of selecting neutrinos, relying on a single model and neutrino probability, in contrast to using two separate binary classifiers. The track cascade classifier also does well, when the AUC from the ROC curve is compared to that of the GraphNeT paper[1].

It turns out that in order for the classification results to generalize to actual data, the OscNext lvl3 cuts are required. They ensure that the Monte Carlo - data agreement is good enough for GraphNeT to work on data. They achieve this by cutting away high energy muons, coincident events and obvious noise events using 12 variables.

Relying on the lvl3 cuts, it has been demonstrated that DynEdge seems to be able to classify a clean low energy neutrino selection in data. A total of 8753 events from an approx. 1% burnsample of OscNext data were classified as neutrinos, representing a rate of 2.235 mHz. This corresponds to approximately 875,300 neutrinos in all OscNext data, which if correct could be the largest clean low energy neutrino selection to date. The track and cascade neutrino selections in data and Monte Carlo are generally very similar across reconstructed and calculated variables. Even when considering the energy as function of zenith, azimuth and vertex z position, the agreement is very good. As such, there are strong reasons to believe that the classified neutrino selection in data is approximately as clean as the one in Monte Carlo.

In Monte Carlo, the DynEdge neutrino selection is about 70% larger than that of the OscNext selection, both when compared after their final lvl7 neutrino classifier and when compared after the majority of the analysis cuts (energy, zenith and containment). It is simultaneously of a higher purity. It remains to be seen if the improvement consists after the final direct pulse cleaning, but it is quite likely. Given that IceCube has been collecting data for about 12 years, the 70% additional neutrinos would correspond to approx. 8.4 years of runtime. The rate in the DynEdge data neutrino selections also match the theoretical rates from the DynEdge Monte Carlo neutrino selections quite well, with data/Monte Carlo rate ratios of 0.993 and 0.932 before and after the analysis cuts. This suggests that the improvement in neutrino statistics carry over to actual data as well.

The additional Monte Carlo neutrinos in the DynEdge selection are also of a similar "quality" (if not higher) than those in the OscNext selection when compared before and after the analysis cuts. They have a similar energy residual distribution, but a more narrow angular residual distribution. However, once a stricter cut in energy (removing events with energy above $10^{1.75}$ GeV) is enforced, the additional rate of neutrinos drops to 47.8% and the quality is then similar in zenith and energy, but worse in azimuth than the existing neutrinos.

Overall it seems likely that the additional neutrinos would have an impact on the oscillation analysis, hopefully lowering the uncertainties on the oscillation parameters IceCube measures. Furthermore, a number of additional IceCube analyses use the OscNext neutrino selection and would also benefit from the improved statistics.

As such, it has been successfully demonstrated that DynEdge is a reliable method of selecting and classifying low energy neutrinos, not only in Monte Carlo, but also in actual IceCube data. Perhaps one could argue that slight improvements in the classification and reconstruction is not groundbreaking, but one should remember that DynEdge is also very fast. It is fast enough to run live at the South Pole on a single GPU [1]. The speed allows, for instance, to quickly test calibrations and new simulations. It also opens the door for low energy early warnings.

Thus the results are overall very promising. They can also easily be expanded upon.

8.2.2 Simple Additional Analysis Possibilities And Improvements

The analyses and results presented in this thesis can easily be extended. The following points are what I would have investigated given a few more months.

- In Monte Carlo, it is possible to test the effect of the additional neutrinos on the oscillation analysis. One can generate the 90% significance contour of the two oscillation parameters $\Theta_{2,3}$ and $\Delta m_{2,3}^2$, measured in IceCube. This can be done for the OscNext Monte Carlo neutrinos, and the DynEdge Monte Carlo neutrinos, which would illustrate the extra precision obtained from the additional neutrinos in the DynEdge selection. Unfortunately this was not attempted due to time constraints.
- In the comparison of neutrino selections between Monte Carlo and data, a split into track and cascade neutrinos were made (track prob. > 0.9 for tracks and track prob. < 0.5 for cascades). This means that neutrinos that are in-between these selection in track probability have not been compared. This is quite a simple additional analysis, which would hopefully further demonstrate that the classification works in data.
- Another idea that was only briefly investigated is that of splitting the analysis into up and down-going events, using the zenith reconstruction. This would allow the cut in neutrino probability in logit space to depend on whether the zenith reconstruction predicts that the event comes from the southern sky or through the earth. By loosening the cut for those that are predicted to pass through the earth, a larger neutrino sample could possibly be achieved.
- A further check that our neutrino selection in data makes sense, would be to compare directly with the real data events that OscNext classifies as neutrinos. If the majority of the OscNext neutrinos in data are also in our selection, it would be a point in our favor. If on the other hand, there was no overlap in the data neutrino selections, it would suggest that our method does not work as well as has been argued.
- In the neutrino selection rate comparison in Monte Carlo, it would make sense to replicate their pulse quality cut to ensure that our additional neutrinos are not all removed by it. It should be relatively straight forward.
- The motivation for using a multiclassification method was partly the simplicity of only having to make a single cut. But partly it was also from a belief that with two binary classifiers (particle vs noise & neutrino vs muon), the noise that passed the first would resemble neutrinos more than muons and hence also end up in the final neutrino selection. However, this claim has not been tested directly. As such, it would be interesting to train two binary classifiers on the same training set used in this work and with similar hyperparameters. Then an analysis of which method produces the cleanest and largest neutrino selection could be carried out.
- In figure 7.32, showing the rate of each particle that survives depending on the cut in neutrino probability in logit space, the data / Monte Carlo ratio is not constant. While this has to do with the data - Monte Carlo agreement, it is not well understood. One could try to scale each particle and perhaps smear the

neutrino probabilities of the Monte Carlo muons to get it to match. This would help us to understand in what way the simulations differ from the actual data.

- Recently we were made aware that the energy bias between neutrinos in data and Monte Carlo might be related to the energy deposited in the tracks of CC ν_μ . Therefore it would be interesting to make a new model, which predicts the amount of energy deposited in the track part of the events, as contrasted to the initial cascade and investigate this further.
- In the analysis, the neutrino selection was based on a cut of 12 in the neutrino probability in logit space. This was not based on solid analysis, but was chosen from investigating figure 7.21, which shows the Monte Carlo particle distributions in logit neutrino probability scaled to data. It would be interesting to carry out an analysis of varying the cut and seeing what happens to the comparisons. In general if DynEdge were to be used directly in IceCube, the initial cut would be much looser, and then for each separate analysis, additional cuts in neutrino probability and reconstructed variables should be determined.
- As Andreas Mosgaard Jørgensen is currently investigating, it is interesting to see if we can get away with looser lvl3 cuts to get Monte Carlo to match data, given that we are using DynEdge instead of the OscNext classification methods. Depending on what cuts are eventually chosen, it might make sense to retrain the models only on the Monte Carlo events that make it past the cuts. Retraining the models, one should make sure that no neutrinos in the multi-classification test sets are used to train the reconstruction models.
- It was chosen to use the uncleaned pulsemap SplitInICEPulses, since it resulted in a better multiclassification model than the cleaned SRTInICEPulses. I present no particular evidence, but the validation loss was quite a lot better when we trained on SplitInICEPulses. However, this might not be the case for the reconstruction models. Therefore it would also be interesting to see if for reconstruction, the SRTInICEPulses leads to better results. However, given the limited difference in using various pulsemaps in the high energy northern track benchmark, I do not expect a large improvement. However, work is being carried out to allow DynEdge to construct a clean pulsemap, which has been shown to improve the current reconstruction methods a lot. At the moment it requires IceCube Upgrade data, since there is no truth flags for whether a DOM hit is noise or not in the non-upgrade Monte Carlo data. It should, however, be possible to work around this by training only on the old doms in the IceCube upgrade simulated data. It would be interesting to see how a GraphNeT cleaned pulsemap improves the reconstruction models.

The above things are what I would look into given a few more months. In the next section a longer outlook is presented.

8.2.3 Long Term Potential

The potential of DynEdge is excellent in low energy Monte Carlo data and seems to generalize to actual data.

The quality of the models and the data - Monte Carlo agreement is at a level where it would make sense to test on a larger scale. Given the speed of the models, it should be possible to relatively quickly run through the entirety of IceCube data.

Given the 8753 neutrinos in the approx. 1% sample, one would expect around 800 thousand neutrino candidates from the entire data. This would be excellent statistics to confirm that the method works. Furthermore, by being able to quickly run on all data, small configurations and calibrations can be tested fast and on a large scale, compared to the existing reconstruction methods. Running on all data would require being able to run on the GPUs in the IceCube Hub in Madison, given the size of the data.

The above would yield a low energy - DeepCore centred neutrino selection by DynEdge. However, it would be great if we could simultaneously classify high energy neutrinos and neutrinos in the entirety of the detector. This would require simulations of high energy neutrinos and muons that are compatible with the current Monte Carlo data. Furthermore, including coincident events in the training, could allow a loosening of the lvl3 cuts. Eventually it might even be possible to select atmospheric neutrinos where a muon simultaneously hits the detector. To include coincident events, the Corsika simulation model could be useful, but otherwise merging neutrino and muon event pulsemaps might make sense.

Currently the early warning sent out by IceCube stem from very high energy neutrinos, since these are easy to distinguish from muons. Low energy neutrinos are not used in early alerts yet, since by the time they have been classified and reconstructed, it has been too long since the event took place. Utilizing GraphNeT to reconstruct low energy events live at the South Pole could potentially open the window to early warnings from low energy neutrinos. This could represent a significant contribution to multi-messenger astronomy and is specifically suited to the fast reliable classification and reconstruction of DynEdge.

Given the strong potential, it will be most interesting to follow the development of GraphNeT in the future.

Appendix A

Azimuth Results In Northern Track Benchmark Project

A.1 DynEdge Baseline Comparison To SplineMPE

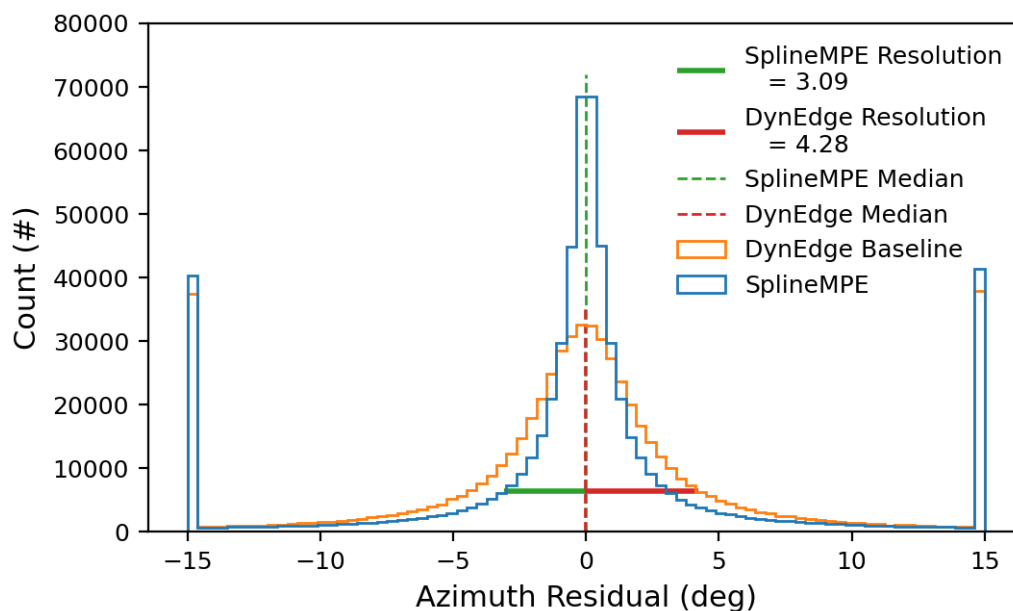


FIGURE A.1: Distribution of azimuth residuals for the baseline DynEdge model and SplineMPE. Overlaid are calculated resolutions illustrated from the medians of the distributions. The resolutions are robust measures of how well the model predicts the zenith angle. If they had a Gaussian shape, it would correspond to a standard distribution. Note that outlier bins are used.

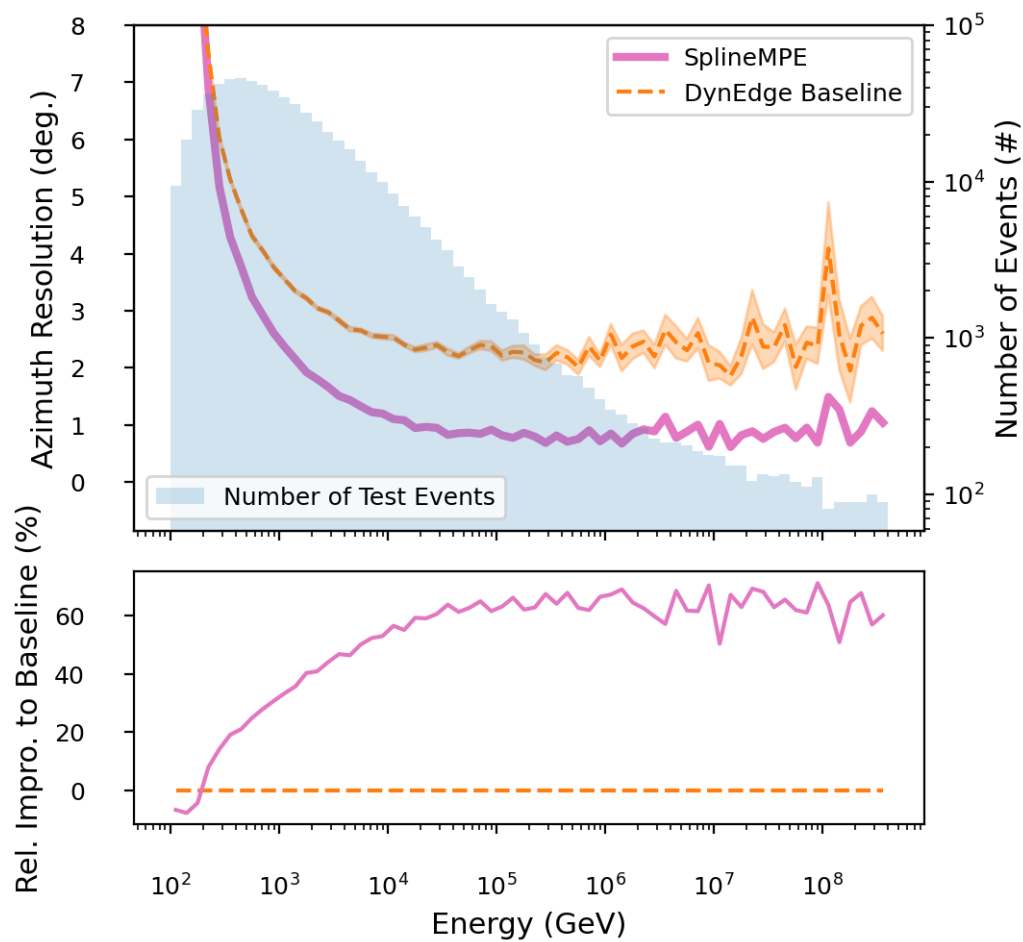


FIGURE A.2: (top) Azimuth resolution with bootstrap std as a function of energy for the baseline DynEdge model and SplineMPE. (bottom) relative improvement of SplineMPE to Dynedge baseline.

A.2 Learning Rate Adjustment

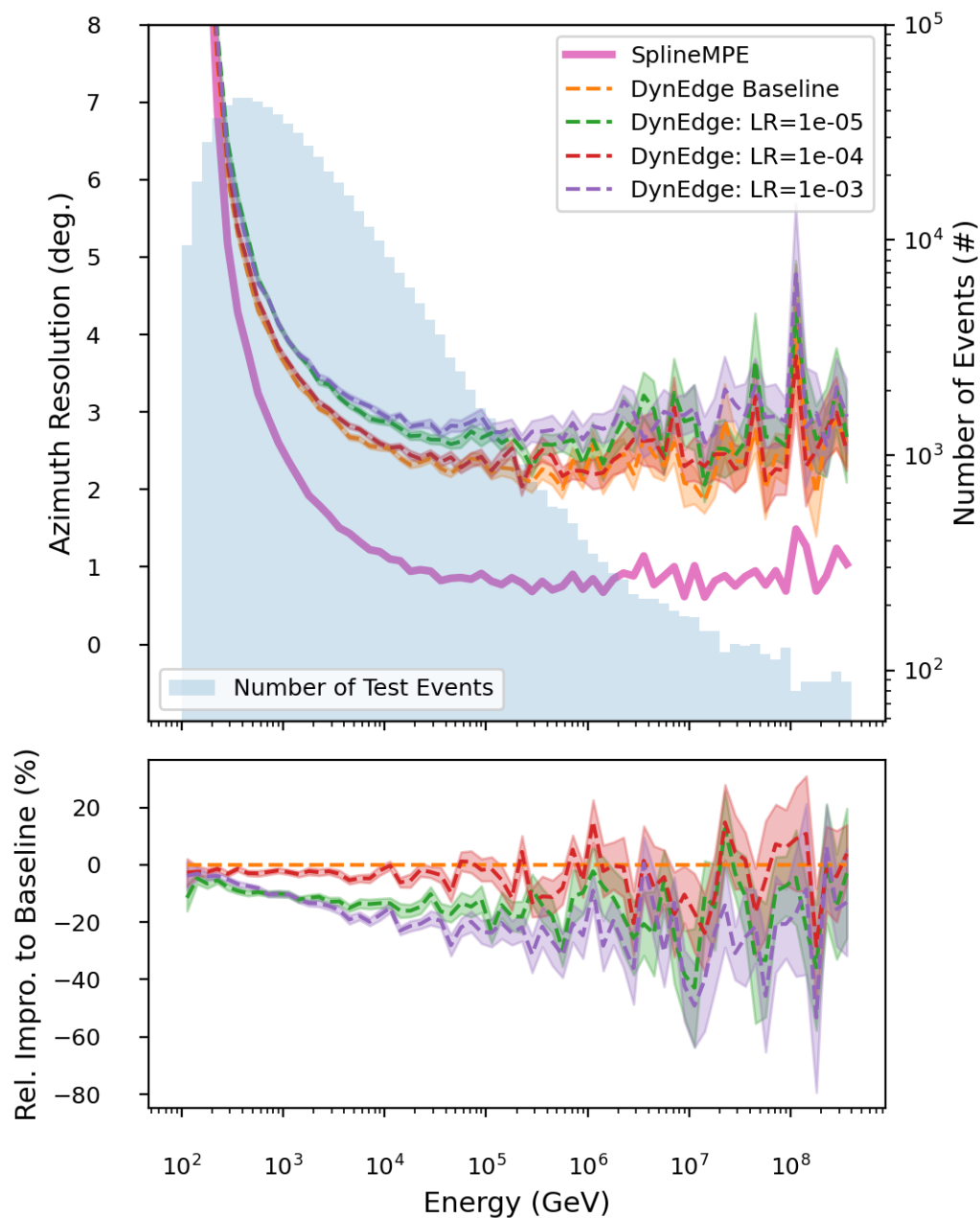


FIGURE A.3: (top) Azimuth resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model with fixed learning rates. (bottom) relative improvement of the adjusted DynEdge models to the baseline.

A.3 Changing The Pulsemap

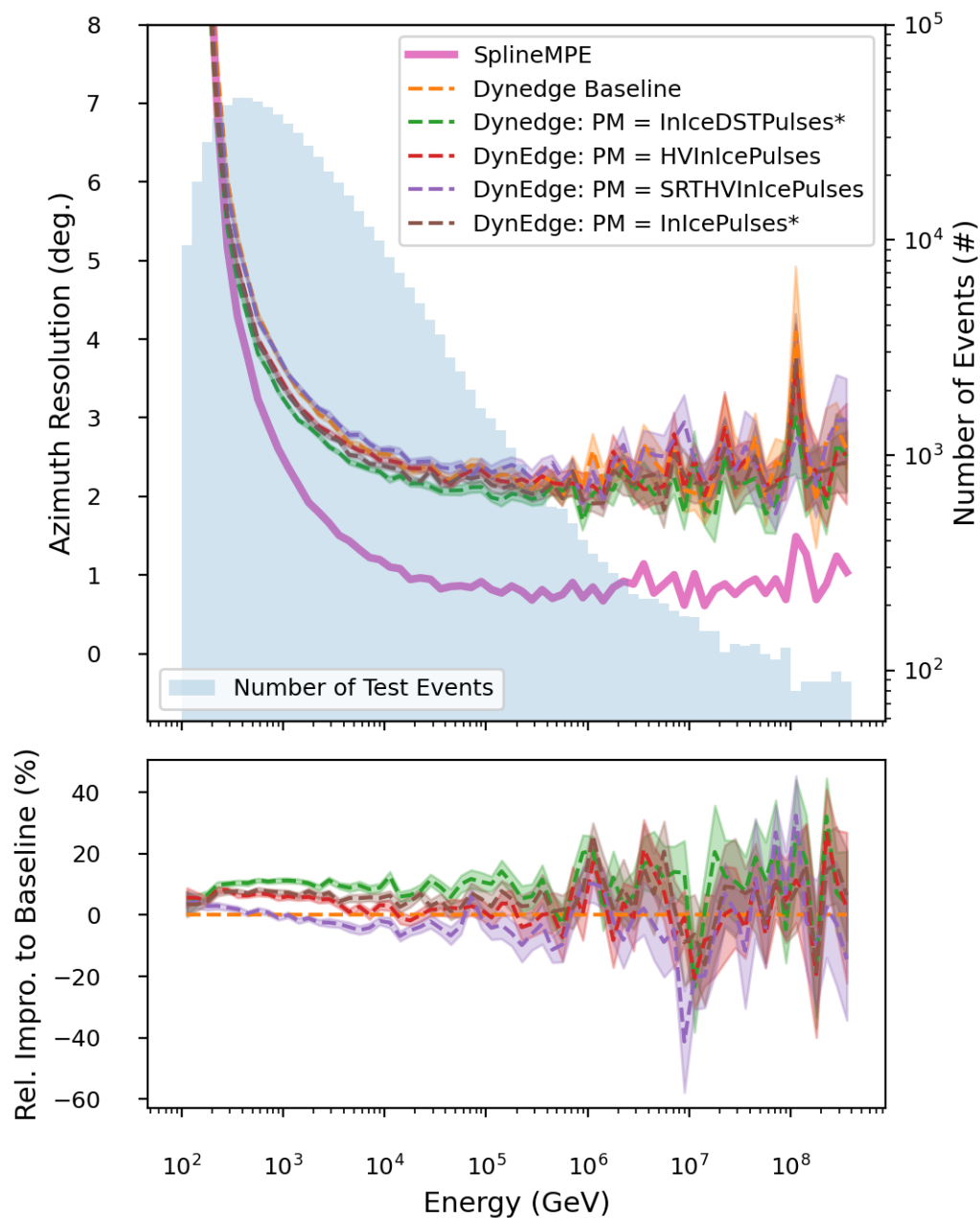


FIGURE A.4: (top) Azimuth resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on different pulsemaps. (bottom) relative improvement of the adjusted DynEdge models to the baseline.

A.4 Focus On Highest Energy Range

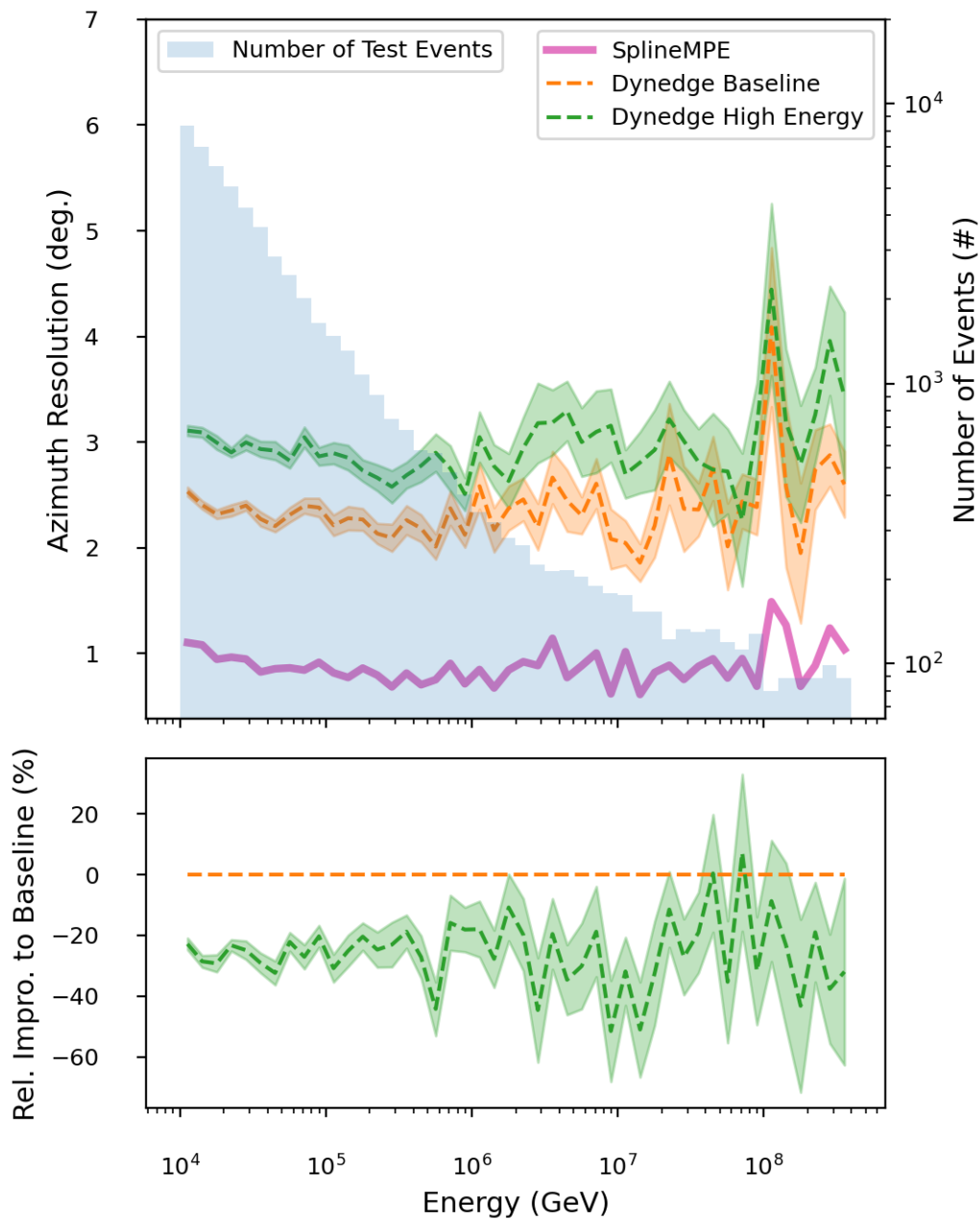


FIGURE A.5: (top) Azimuth resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on neutrinos with energy above 10^4 GeV. (bottom) relative improvement of the adjusted DynEdge models to the baseline. Note that the energy range is limited to above 10^4 GeV, since the adjusted model has no predictions for lower energies.

A.5 Better Prescaling of Input

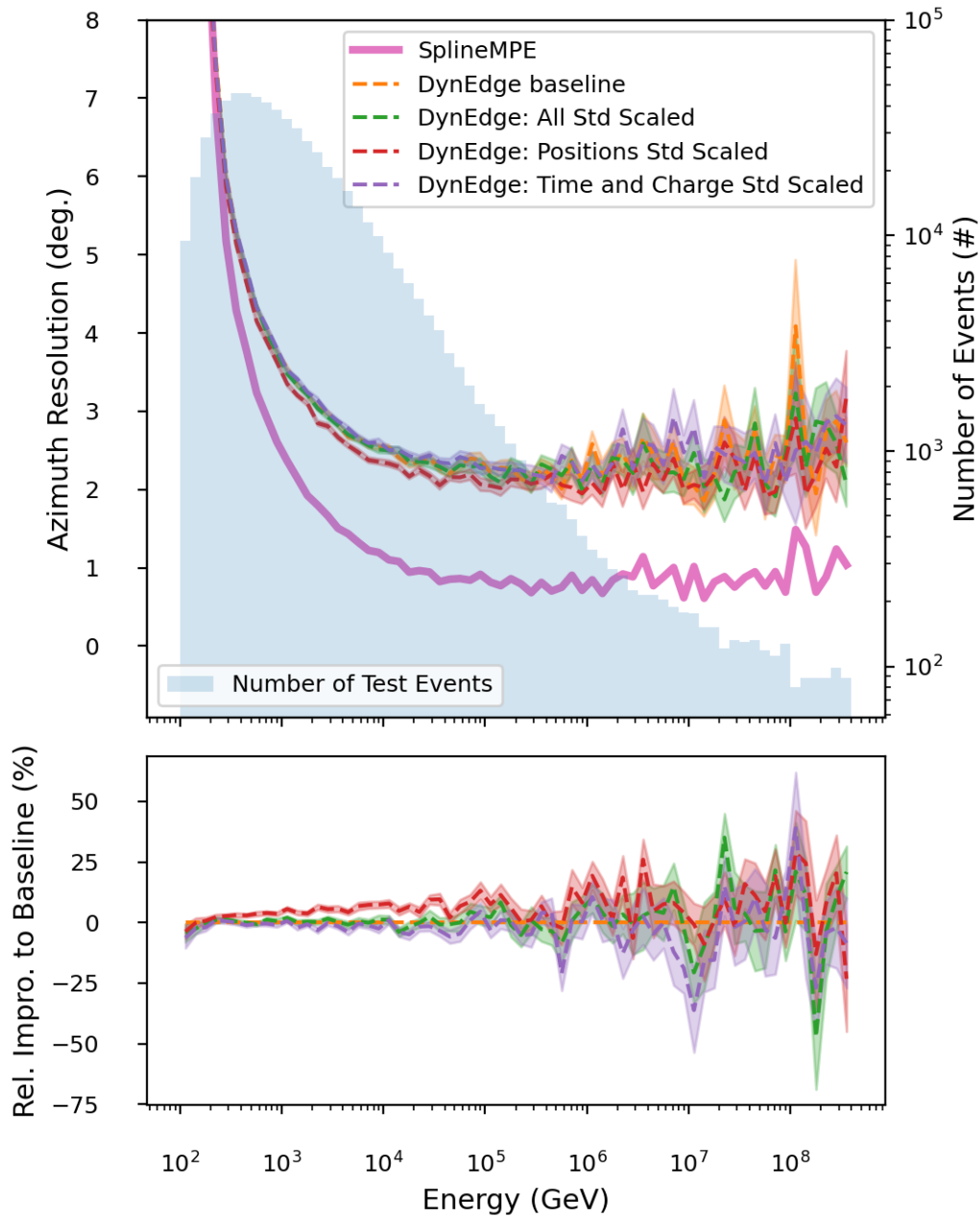


FIGURE A.6: (top) Azimuth resolution with bootstrap std as a function of energy for the baseline DynEdge model, SplineMPE, and the baseline DynEdge model trained on gaussian scaled features. "All std Scaled" means that all five input features from figure 6.6 are scaled using new transformer. "Positions Std Scaled" means only dom positions are scaled using new transformer. "Time and Charge Std Scaled" means only dom time and charge are scaled using new transformer. The * in the labels indicates extended training. (bottom) relative improvement of the adjusted DynEdge models to the baseline.

Appendix B

Monte Carlo Classification And Reconstruction Results - Including Benchmark Against Retro Reconstructions - Additional Results

B.1 Multiclassification - Noise, Muons Or Neutrinos?

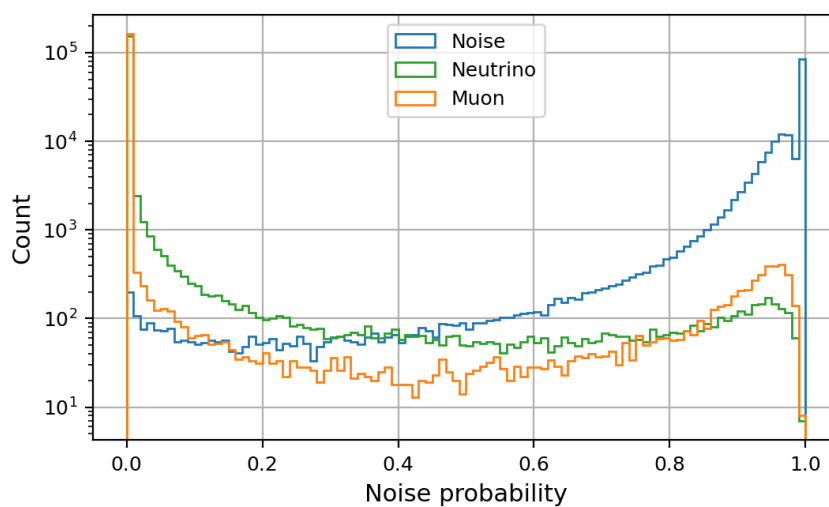


FIGURE B.1: Stacked histogram of all test events as a function of their predicted probability of being noise. Colored by actual particle type.

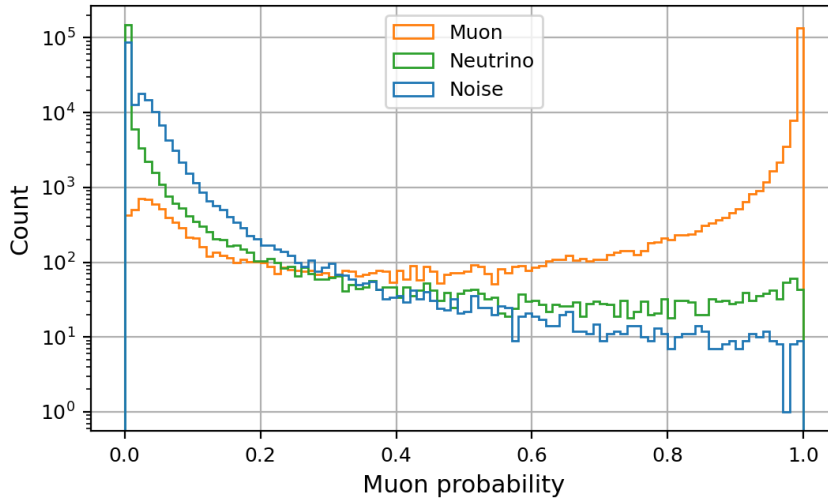


FIGURE B.2: Stacked histogram of all test events as a function of their predicted probability of being muons. Colored by actual particle type.

B.2 Azimuth Reconstruction

A pull plot for the azimuth angle is available in figure B.3. It can be seen that the mean is zero, whereas the standard deviation of the Gaussian is 1.09 instead of 1. This means that the azimuth model standard deviation estimates are also slightly too low.

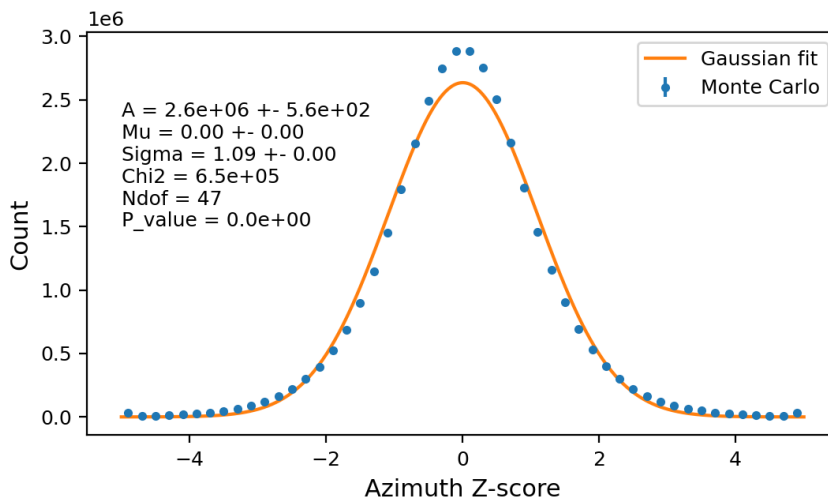


FIGURE B.3: Pull plot. Histogram of number of predicted standard deviations the predicted azimuth angle falls from the truth. Overlaid is a gaussian fit, since this should correspond to a unit gaussian, if the std is estimated correctly. A is the normalisation, mu is the mean, sigma is the standard deviation, Chi2 is the χ^2 value of the fit, Ndof is the degrees of freedom and p-value is the probability of obtaining a worse χ^2 if the data distributions is consistent with the fit. One can see that the mean is 0, while the sigma is 1.09 instead of 1. This means the model slightly underestimates the azimuth standard deviations.

B.3 Interaction Vertex Position Reconstruction

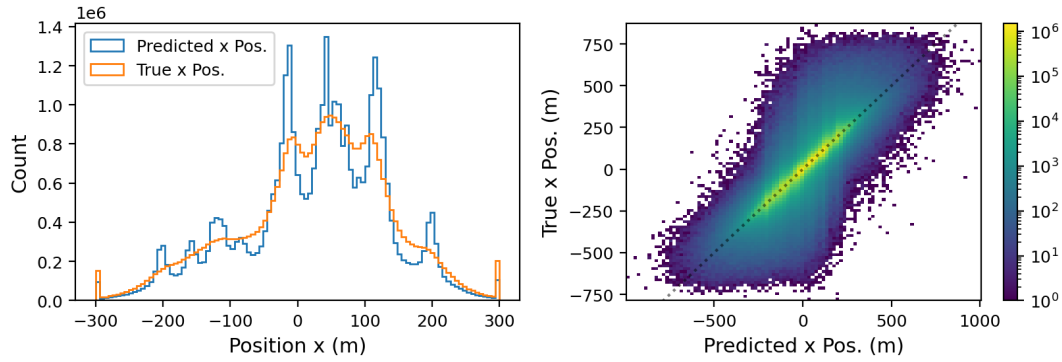


FIGURE B.4: (Left) Predicted and True vertex position x distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted vertex position z .

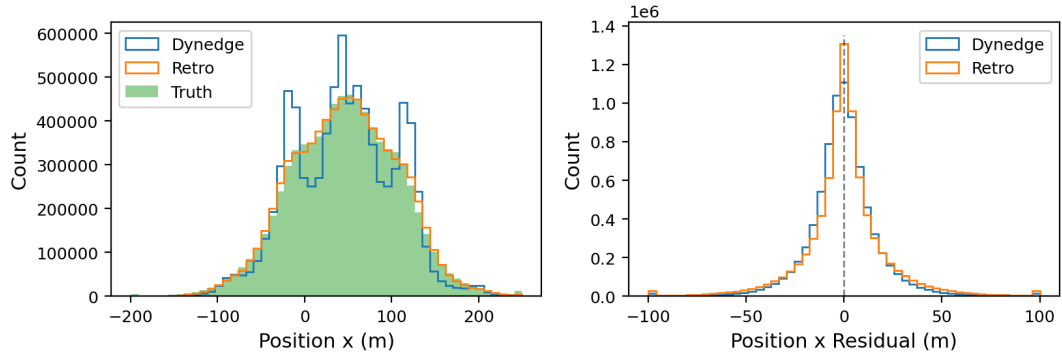


FIGURE B.5: (Left) DynEdge predicted, Retro predicted and True vertex position x distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

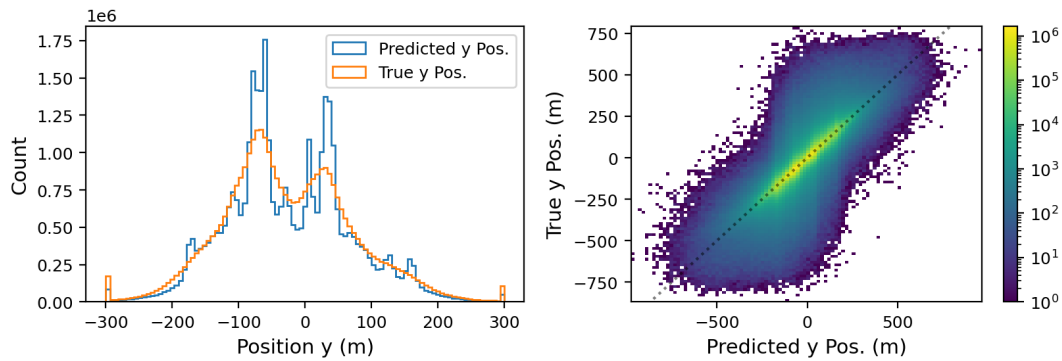


FIGURE B.6: (Left) Predicted and True vertex position y distributions for all neutrinos in test set. (Right) 2D histogram of true vs predicted vertex position y .

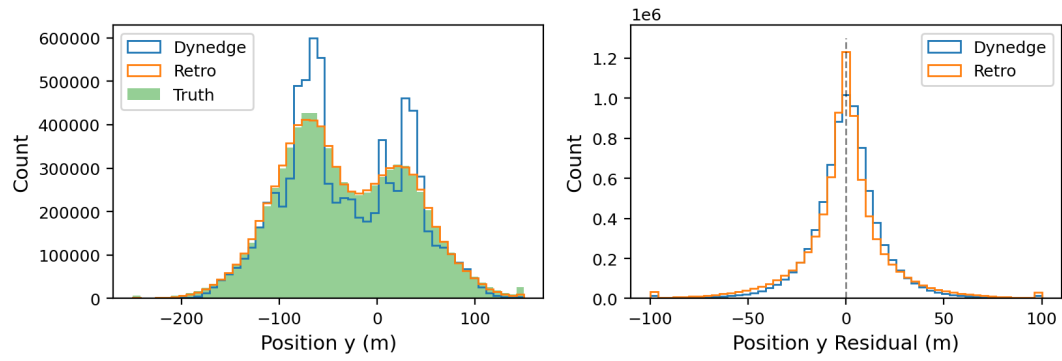


FIGURE B.7: (Left) DynEdge predicted, Retro predicted and True vertex position y distributions for all neutrinos in test set that make it to OscNext lvl 6. (Right) DynEdge and Retro residual distributions.

Appendix C

Necessary Data Cleaning to Ensure Good Monte Carlo / Data Agreement

C.1 Level 3 Variables - Cuts And Replicated Plots

The lvl3 cuts are defined and calculated as part of the OscNext selection process. The following explanation of each variable is taken from [45].

The variables calculated are:

- Number of cleaned DOMs (NchCleaned): A number of hit DOMs in the cleaned pulseseries.
- NoiseEngine: DeepCore Level3 uses the NoiseEngine algorithm without charge information.
- MicroCount hits: The reconstructed pulse series withing $[-4 \text{ s}, +5 \text{ s}]$ from DeepCore are scanned with a 300 ns sliding time window. The algorithm saves a number and charge of the time window with the maximal observed number of hits.
- NAbove200: A number of observed pulses in DOMs with $Z > 200\text{m}$ before the DeepCore trigger time.
- VertexGuess Z: Z-coordinate of the first pulse in the cleaned pulse series.
- Fiducial hits (DCFiducialHits): A number of hit fiducial DOMs in the cleaned pulseseries.
- Veto/fiducial hit ratio: Ratio of hit numbers in veto and fiducial modules in the cleaned pulses.
- Causal Veto Hits: A number of veto hits found by the DeepCore filter (run at level 2). the time considered for each hit is the time of the first pulse in a DOM's pulseseries.
- C2HR6: A fraction of hit DOMs within the first 600 ns with the first 2 hit DOMs ignored.
- Uncleaned time length: A time difference between the first and last pulse in the uncleaned pulse series.

- Cleaned time length: A time difference between the first and last pulse in the cleaned pulse series.
- RT Veto: A number of hits found by the RTVeto algorithm. A different cut value is selected for different ranges of hit fiducial DOM numbers as described in the table below:

DC Fiducial DOMs	RTVeto Cut
Nfid <75	RTVeto250Hits <4.0
75 <= Nfid <100	RTVeto250Hits <5.0
Nfid >= 100	no cut

TABLE C.1

The cut values and whether they target muons, noise or coincident events is available in figure C.1 adjusted from [45].

Variable	Name in dictionary	Targets	Cut value
Cleaned DOM number	NchCleaned	Noise	≥ 6
NoiseEngine	NoiseEngineNoCharge_bool	Noise	==True
MicroCount hits	STW9000_DTW300Hits	Noise	>2
Fiducial hits	DCFiducialHits	Noise	> 2
NAbove200	NAbove200Hits	Muons	< 10
VertexGuess Z	VertexGuessZ	Muons	< -120 m
Causal Veto Hits	CausalVetoHits	Muons	< 7
Veto/fiducial hit ratio	VetoFiducialRatioHits	Muons	< 1.5
C2HR6	C2HR6	Muons	> 0.37
RT Veto	RTVetoCutHit	Muons	== True
Uncleaned time length	UncleanedFullTimeLength	Coincident events	< 13000 ns
Cleaned time length	CleanedFullTimeLength	Coincident events	< 5000 ns

FIGURE C.1: Table with overview of the lvl3 cut values and whether they target muons, noise or coincident events. Adjusted from [45].

In all the following figures, Monte Carlo is scaled by 1.411 to get the rate to match data for lvl2+DC. The variables are extracted from the I3 files and have been calculated by the OscNext group. They are presented here to give a better understanding of the necessary data - Monte Carlo agreement cuts. In the top of each figure is shown the distribution of each Monte Carlo particle type in the lvl3 variable in question. Also plotted are the data distribution for the $\sim 1\%$ burnsample. In the bottom the data to MC agreement ratio is plotted. The cut selection is illustrated with dashed lines and arrows indicating which section is considered signal and kept past lvl3 in the OscNext selection. One sees that they have removed sections where the data/MC ratio is high, and there are simultaneously few Monte Carlo neutrinos in the part that is cut away. The figures are replication of those found in [45].

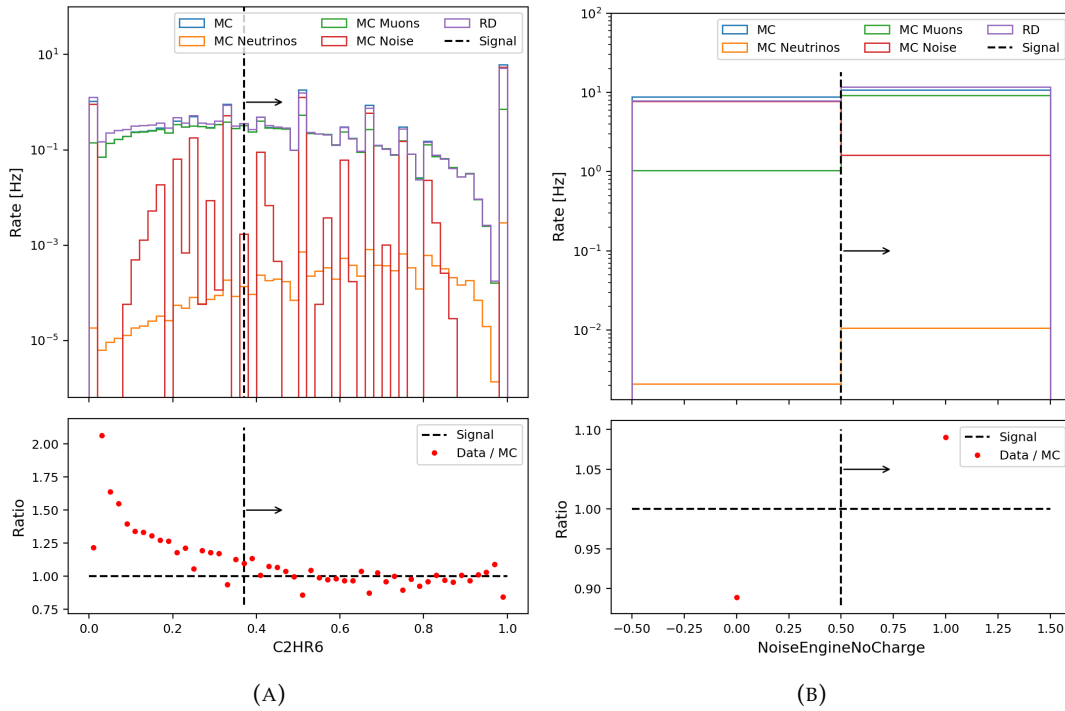


FIGURE C.2: Figures illustrating the lvl3 variables C2HR6 (A) and NoiseEngineNoCharge (B) and their cuts. See text for further explanation.

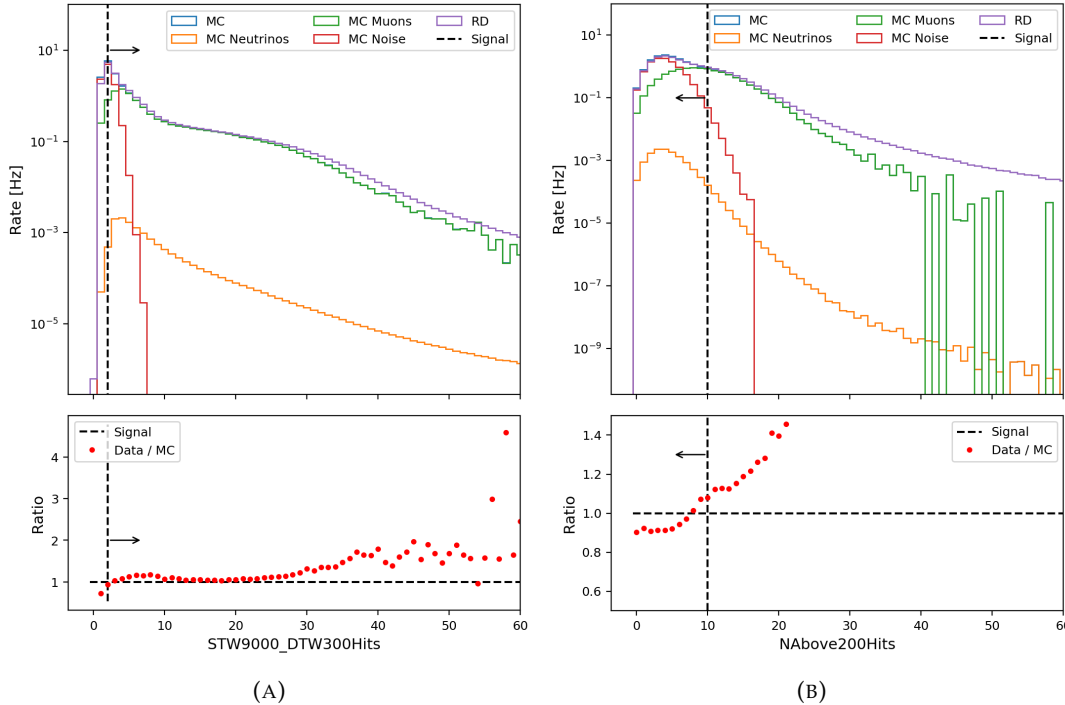


FIGURE C.3: Figures illustrating the lvl3 variables STW9000_DTW300Hits (A) and NAbove200Hits (B) and their cuts. See text for further explanation.

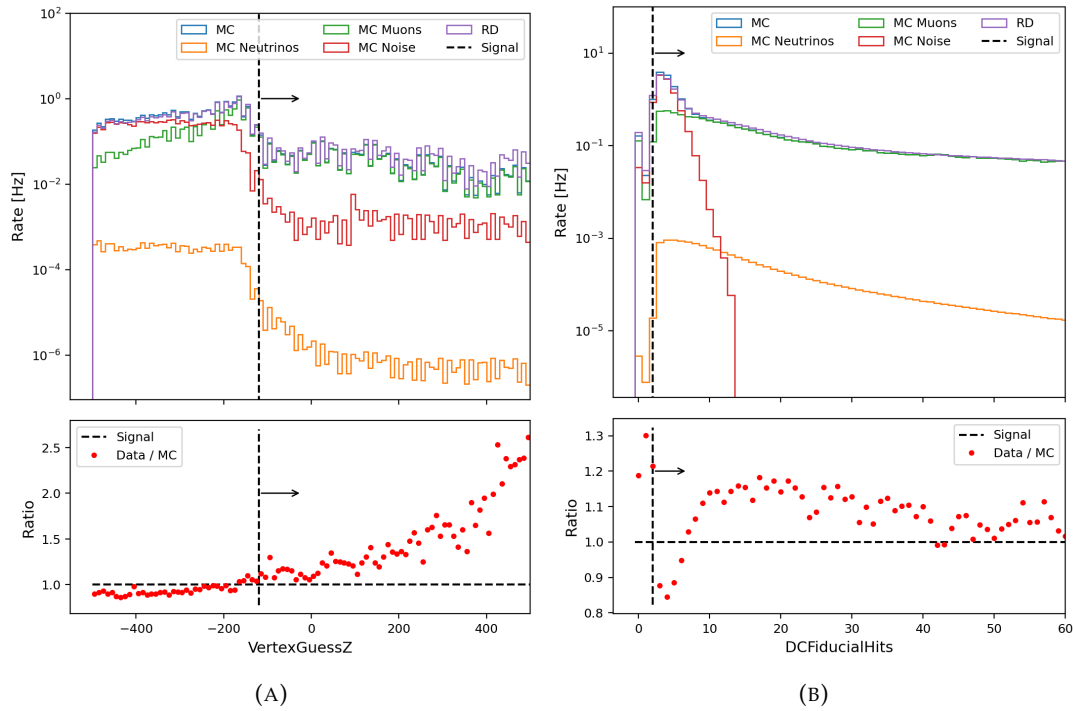


FIGURE C.4: Figures illustrating the lvl3 variables VertexGuessZ (A) and DCFiducialHits (B) and their cuts. See text for further explanation.

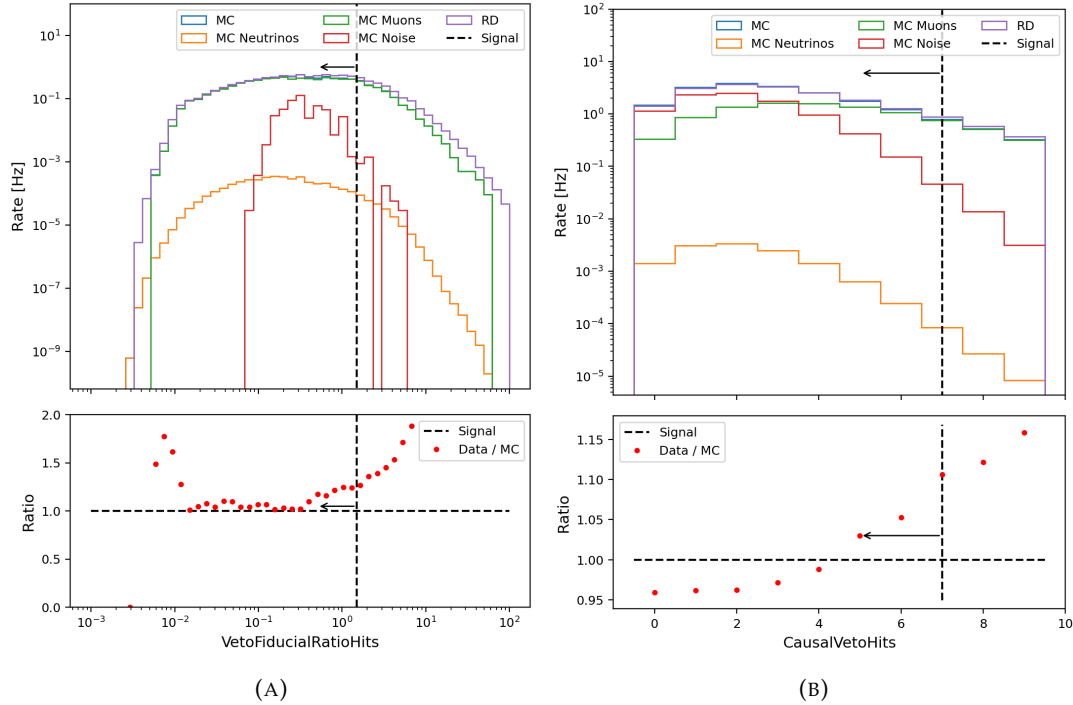


FIGURE C.5: Figures illustrating the lvl3 variables VetoFiducialRatioHits (A) and CausalVetoHits (B) and their cuts. See text for further explanation.

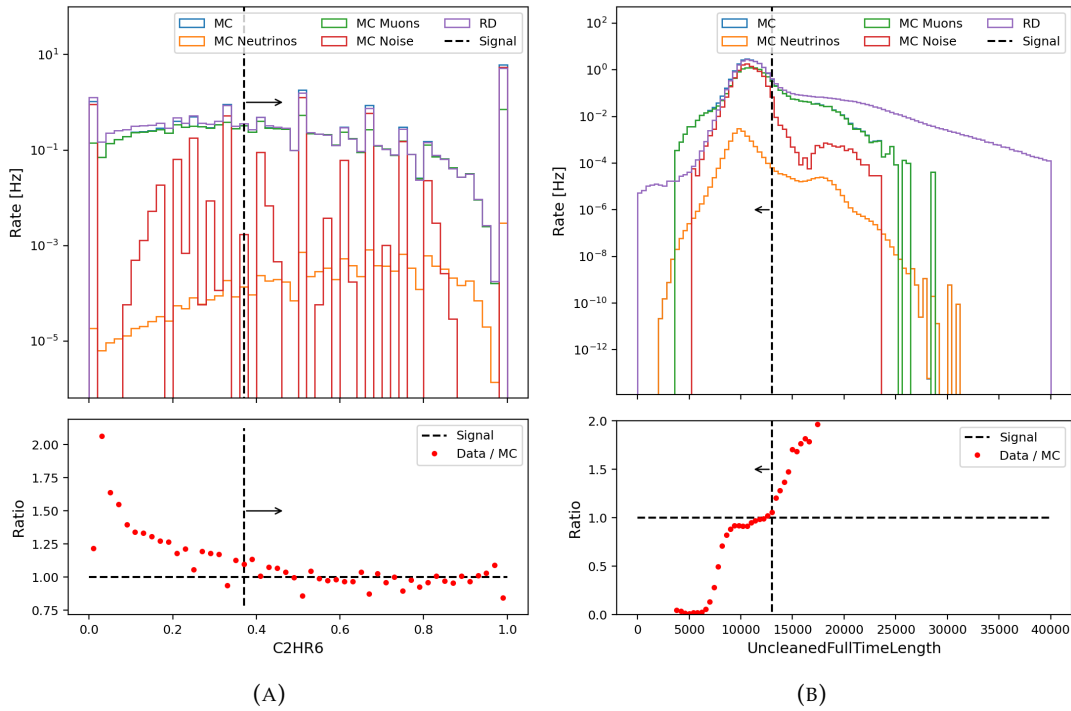


FIGURE C.6: Figures illustrating the lvl3 variables C2HR6 (A) and UncleanedFullTimeLength (B) and their cuts. See text for further explanation.

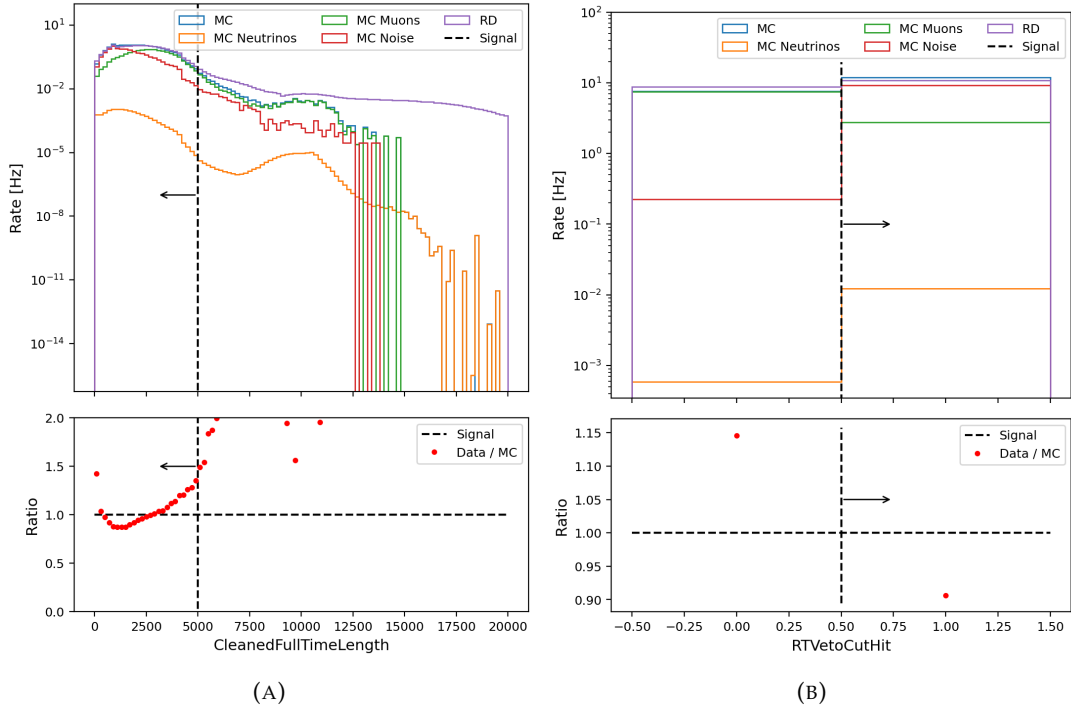


FIGURE C.7: Figures illustrating the lvl3 variables CleanedFullTimeLength (A) and RTVetoCutHit (B) and their cuts. See text for further explanation.

C.2 Neutrino Comparison Without Lvl 3 Cuts Applied

These figures are created using lvl 2 + DC data. When compared to the identical figures in section 7.6, where the lvl 3 cuts are applied, they show the necessity of including the lvl 3 cuts.

The scale factors for Monte Carlo noise, muons and neutrinos in figure C.8 are 0.1179, 2.4492 and 2.478.

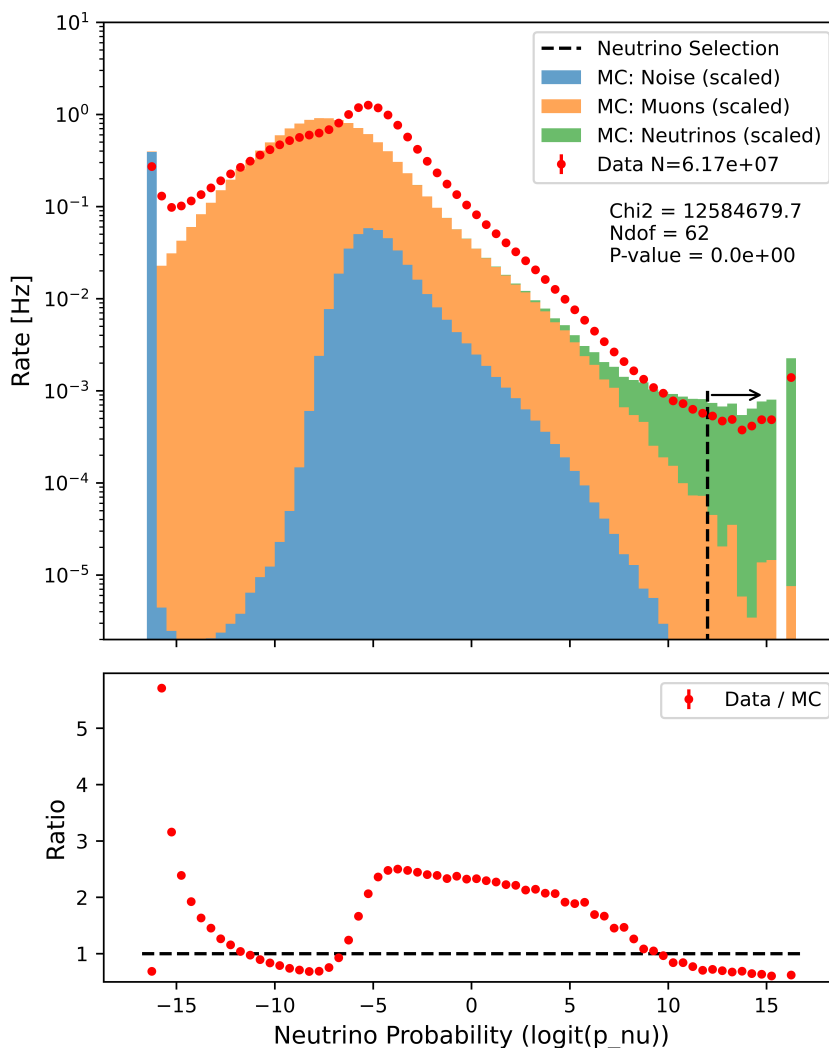


FIGURE C.8: This is without lvl 3 cuts! (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. See text for more information. (Bottom) Ratio of data to Monte Carlo total rate.

The scale factors for Monte Carlo track and cascade neutrinos in figure C.9 are 1.346 and 2.025.

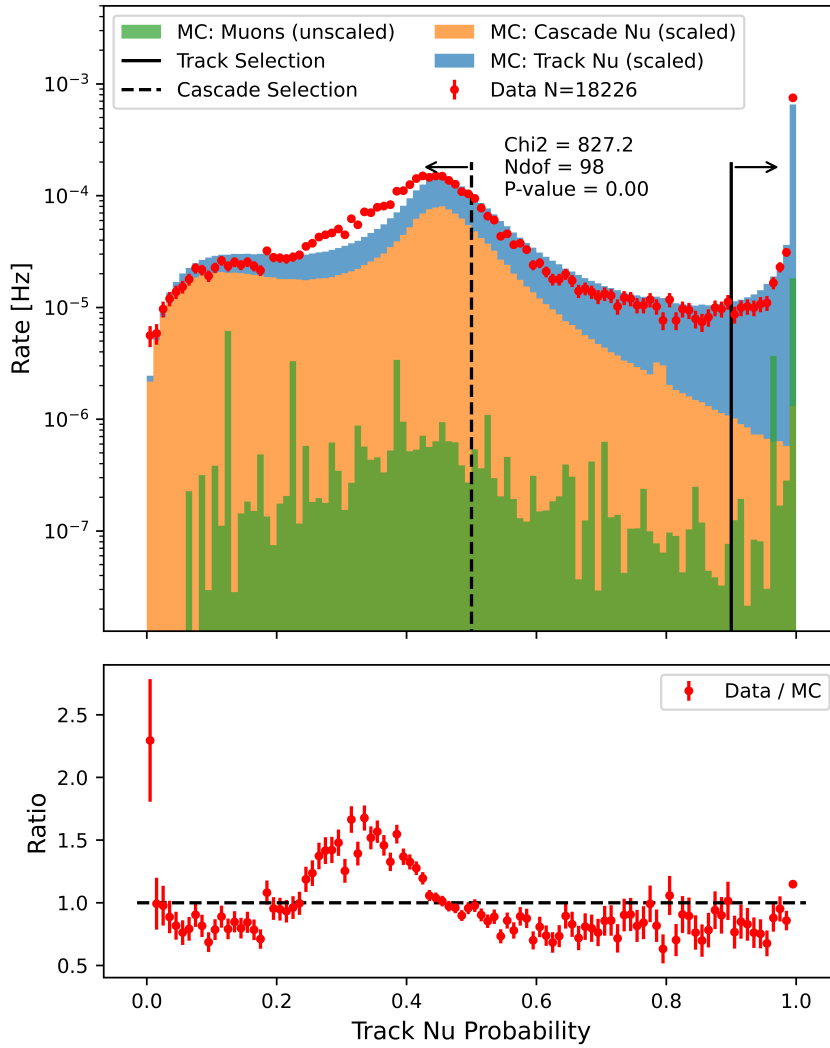


FIGURE C.9: This is without lvl 3 cuts! (Top) Stacked histogram of track neutrino probabilities for the Monte Carlo distributions of track and cascade neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. The surviving sneaky muons are also plotted. (Bottom) Ratio of data to Monte Carlo total rate.

The Monte Carlo track neutrino rate is scaled by 1.412 and the Monte Carlo cascade neutrino rate is scaled by 1.838 in figures C.10, C.11 and C.12.

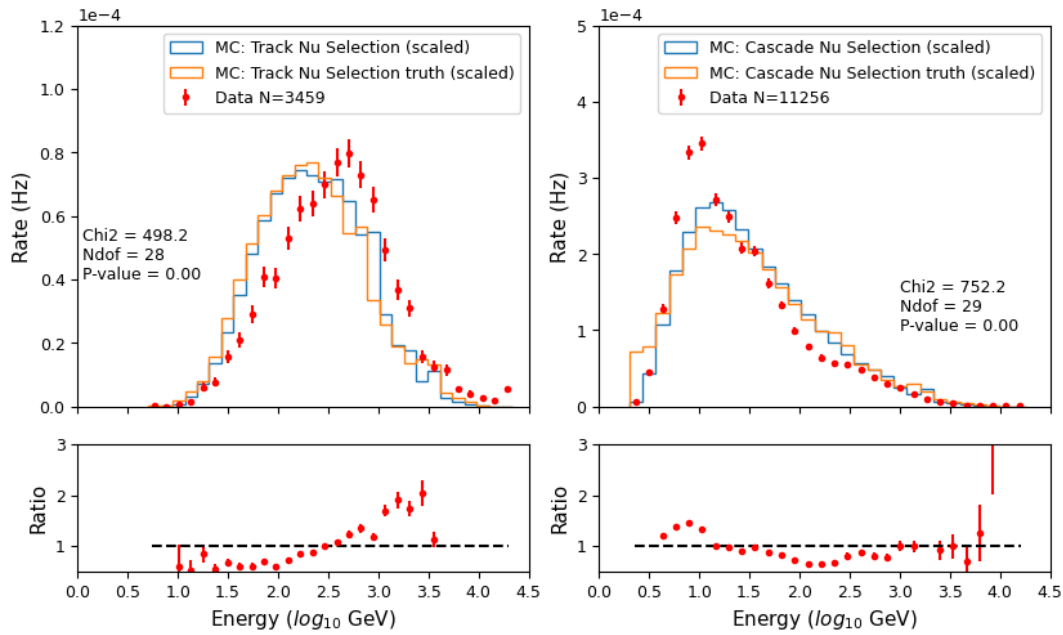


FIGURE C.10: This is without lvl 3 cuts! Energy distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. For more information, see section 7.5.

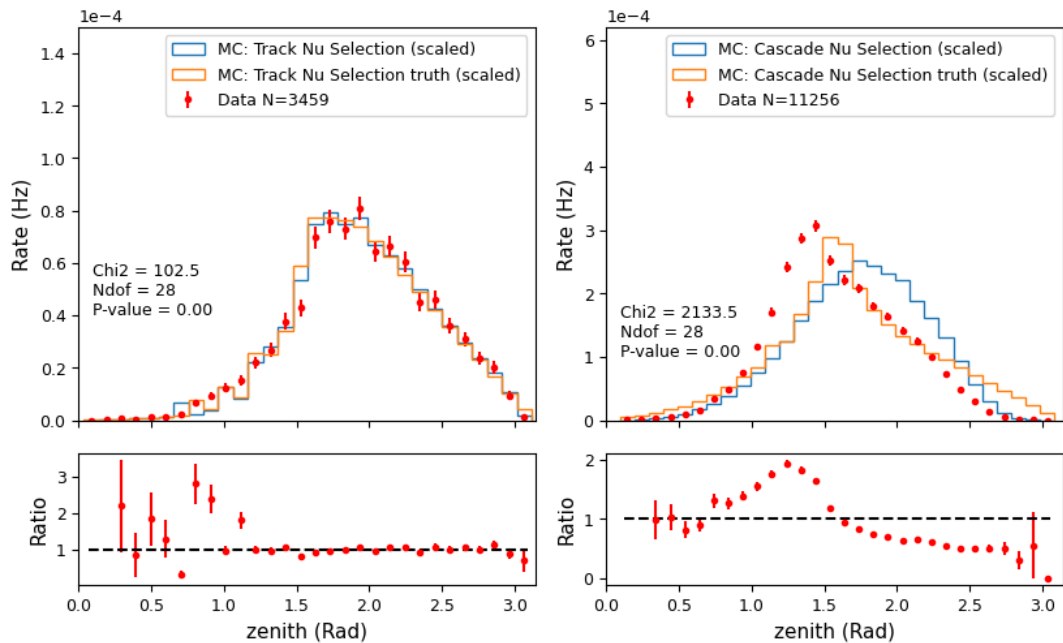


FIGURE C.11: This is without lvl 3 cuts! Zenith distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. For more information, see section 7.5.

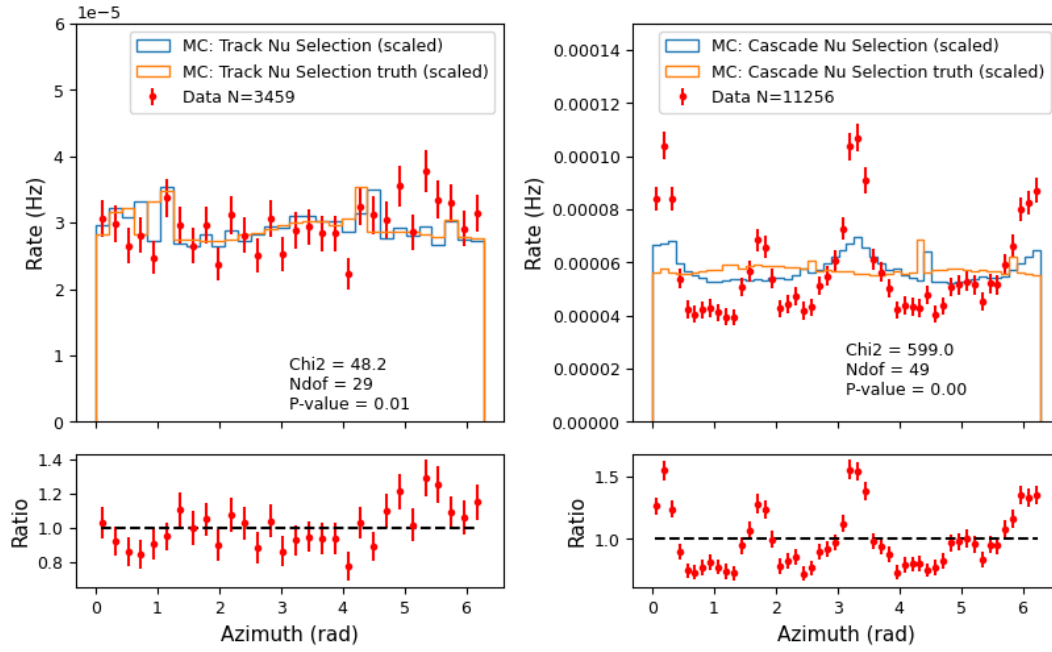


FIGURE C.12: This is without lvl 3 cuts! Azimuth distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. For more information, see section 7.5.

Appendix D

Comparison of Neutrino Selections in Monte Carlo and Data - Additional Plots

D.1 Additional Distribution Comparisons Between Data and Monte Carlo Neutrino Selections

In figure [D.1](#) the reconstructed zenith uncertainty distributions for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar, despite being very different for track vs cascade neutrinos. Furthermore, for both selections the χ^2 test suggest that the fits are excellent with p-values of 0.95 and 0.32. It also makes sense that the track-like neutrinos have much smaller zenith uncertainties than the cascade-like neutrinos, since they should deposit signatures with more directional information.

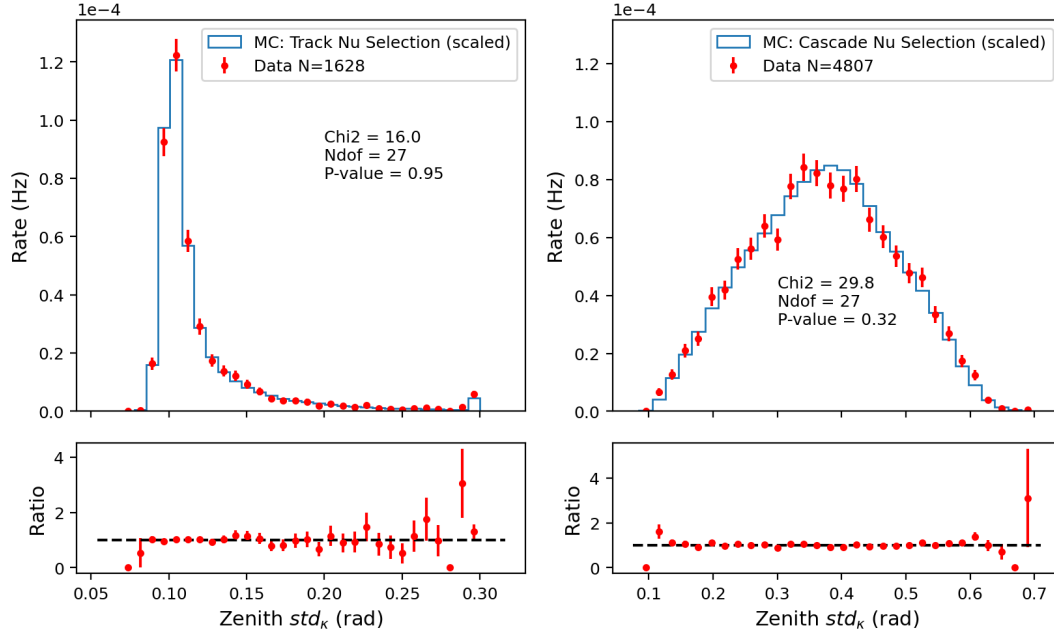


FIGURE D.1: Zenith std distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions match very well.

In figure D.2 the reconstructed azimuth uncertainty distributions for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar, despite being quite different for track vs cascade neutrinos. Furthermore, for the track neutrino selection the χ^2 test suggest that the fit is excellent with p-values of 0.39. Whereas the cascade neutrinos selection has a p-value of 0, but visually looks quite good. As with the zenith uncertainties, it also makes sense that the track-like neutrinos have much smaller azimuth uncertainties than the cascade-like neutrinos, since they should deposit signatures with more directional information.

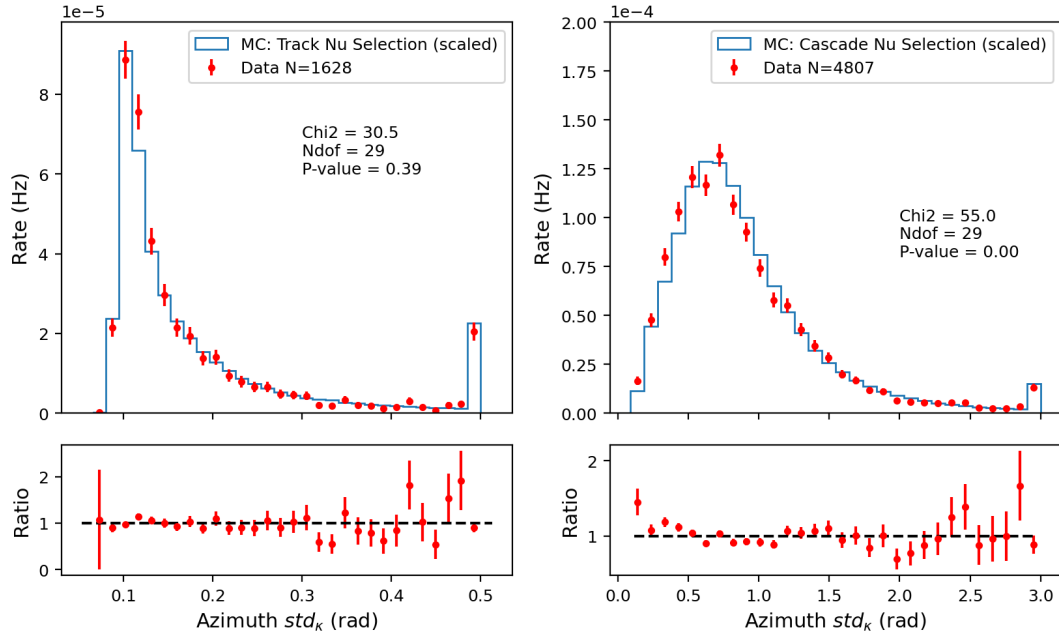


FIGURE D.2: Azimuth std distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. The track distributions match well, whereas the cascade distributions do not in the p-value but visually it is still decent.

In figure D.3 the distributions of reconstructed x coordinates for the interaction vertex for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar, especially when considering the quite weird shape. For both selections the χ^2 test suggest that the distributions are close to being significantly different with p-values of 0.02 and 0.08. The peaked shapes of the reconstructed position distributions in comparison to the true distributions could be the result of the model preferring to predict the vertex close to the DOMs in DeepCore that initially hit. One way to investigate this, would be to look at the x-positions of the DOMs and see if there are specifically many DOMs with x-coordinates near the peaks.

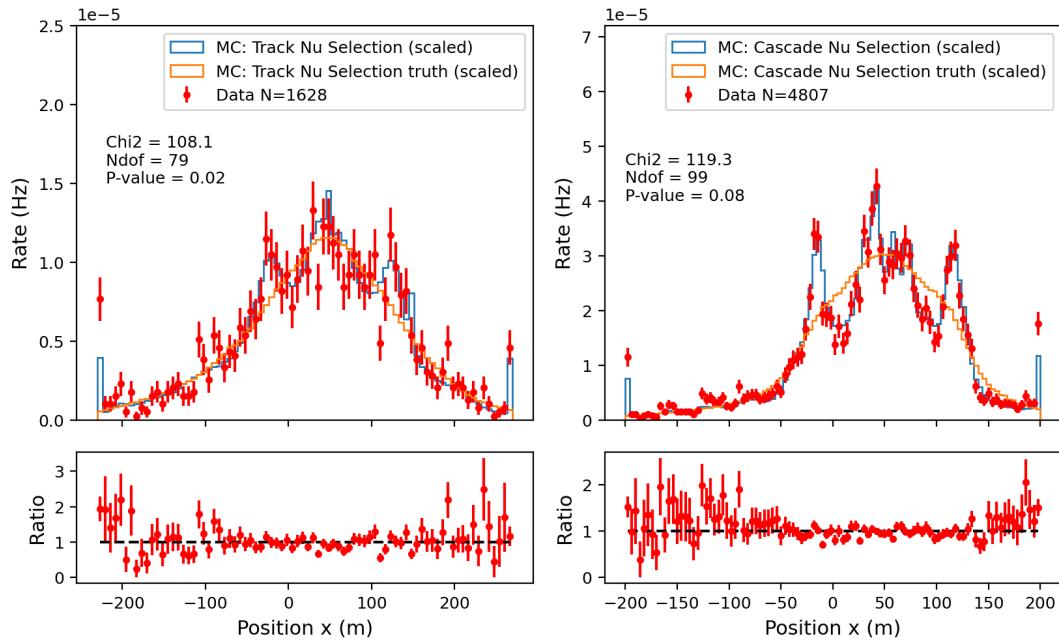


FIGURE D.3: X coordinate of interaction vertex position distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions match decently.

In figure D.4 the distributions of reconstructed y coordinates for the interaction vertex for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually very similar. For both neutrino selection, the χ^2 test suggest that the distributions match decently with a p-value of 0.04 and 0.02.

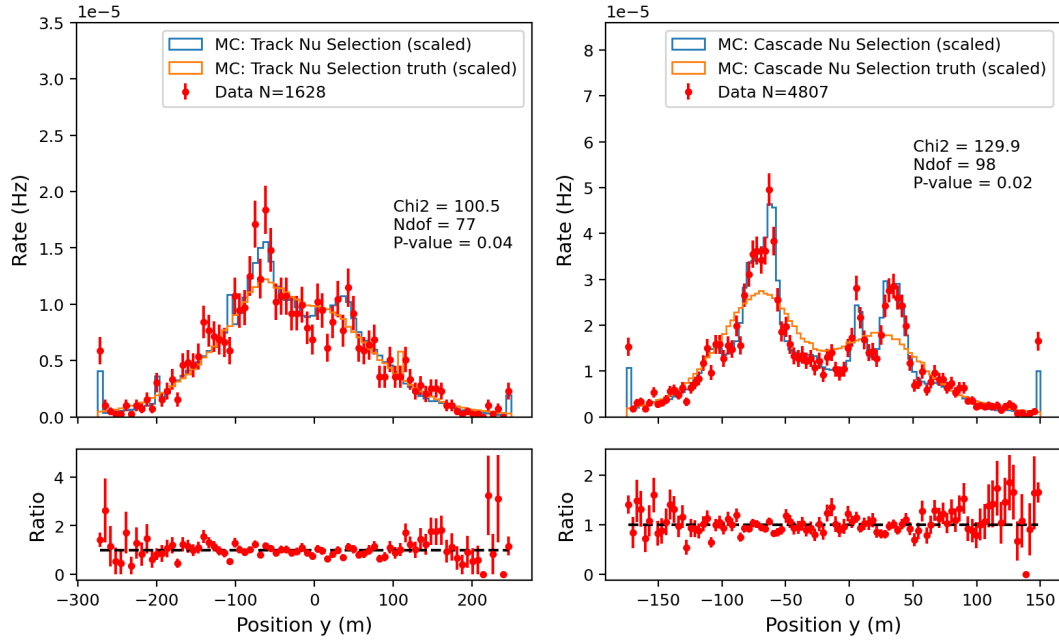


FIGURE D.4: Y coordinate of interaction vertex position distributions for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions match decently.

In figure D.5 the distributions of number of DOM hits (pulses) for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually quite similar. For both neutrino selections, the χ^2 test suggest that the distributions do not match completely well, with p-values of 0 and 0.02.

In general for the interaction vertex positions, there tend to be slight excesses close to and in the outlier bins, which when looking at the muon and noise events in the figure in appendix, could perhaps be a sign of a small contamination. However, it could also arise from the fact that the neutrino simulations are mainly targeted at the DeepCore part of the detector, where most of the neutrinos also end up interaction in the neutrinos selections.

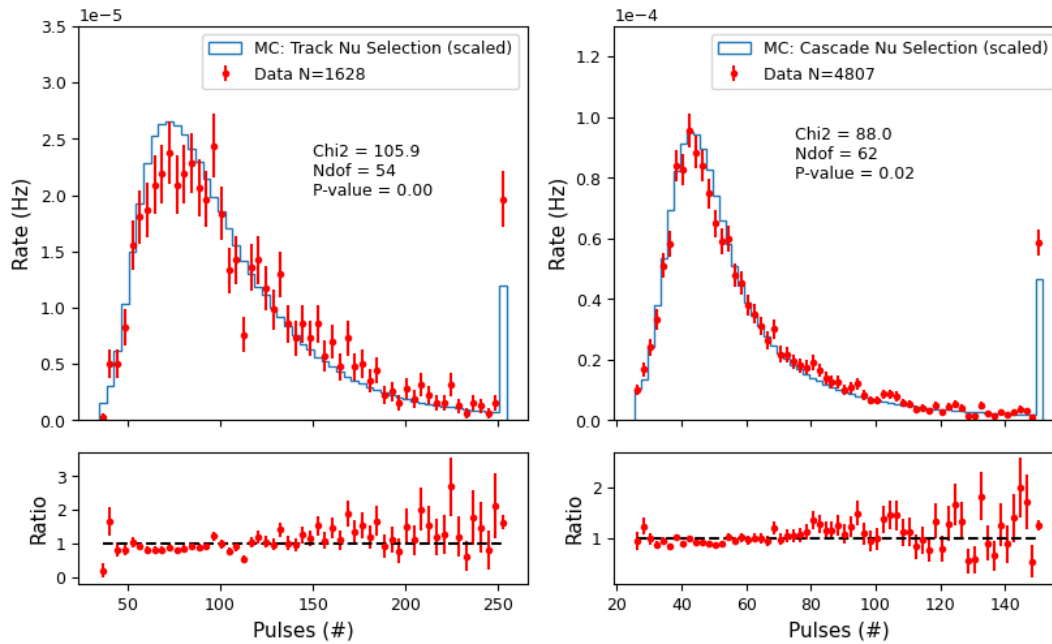


FIGURE D.5: Distributions of number of pulses in the events for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions does not match very well according to p-value, but decently by eye.

In figure D.6 the distributions number of unique DOMs being hit for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually quite similar. For both neutrino selections, the χ^2 test suggest that the distributions match very well, with p-values of 0.36 and 0.17.

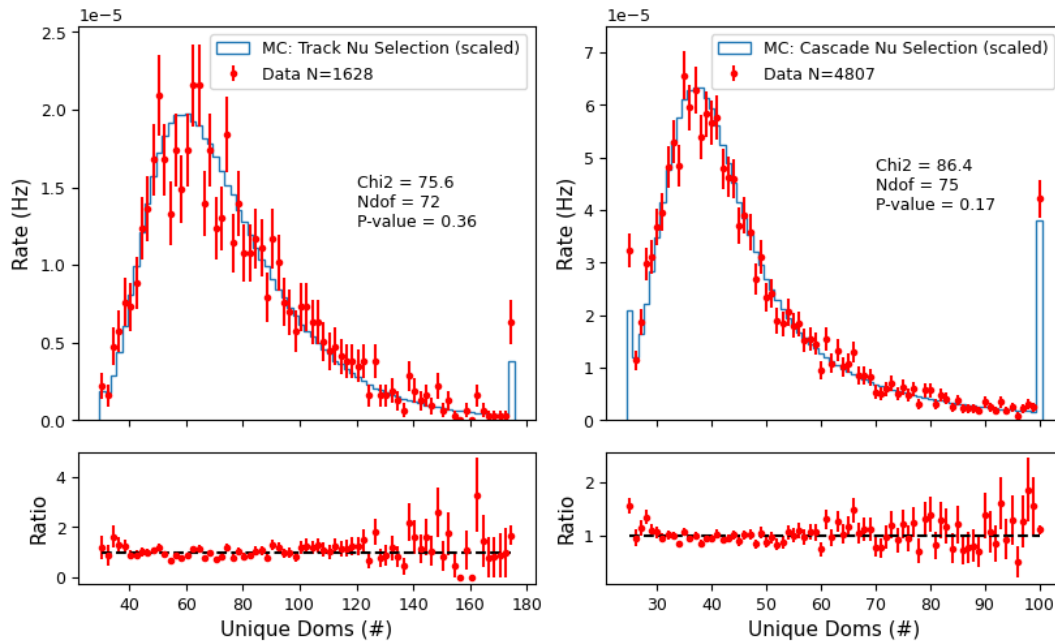


FIGURE D.6: Distributions of unique DOMs being hit in the events for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Both track and cascade distributions match very well.

In figure D.7 the distributions number of unique strings being hit for track and cascade neutrinos in Monte Carlo and data are shown. As can be seen, the distributions are visually quite similar. For both neutrino selections, the χ^2 test suggest that the distributions either very well (tracks) or decently (cascades), with p-values of 0.33 and 0.04.

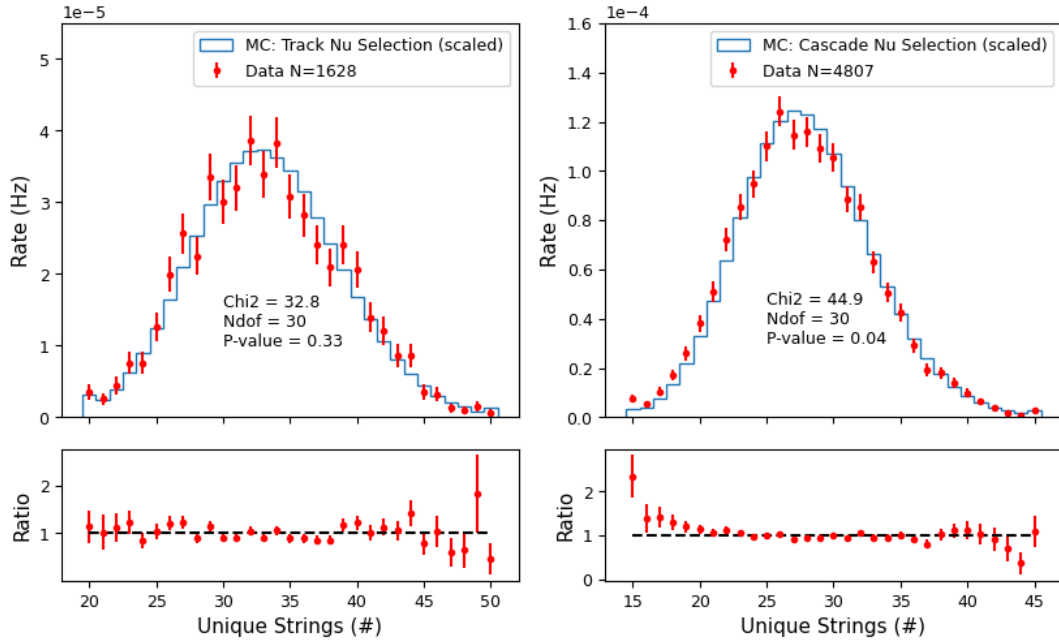


FIGURE D.7: Distributions of unique strings being hit in the events for track neutrino selections in Monte Carlo and data (top left). Similarly for cascade neutrinos selections (top right). In the bottom is shown the ratio of data to Monte Carlo. A χ^2 test with 0 hypothesis that the distributions match are shown as well in each top plot. Track distributions match very well and cascade distributions match decently.

It is also interesting to look at the zenith and azimuth pull plots for the Monte Carlo track and cascade neutrino selections in figures D.8 and D.9. As can be seen, the model generally underestimates the errors for the cascade neutrino selection in both azimuth and zenith, where the Gaussian fits have standard deviations of 1.2 and 1.12. In comparison the uncertainties for the track selections are overestimated, and the Gaussian fits have sigmas of 0.9 and 0.74 in azimuth and zenith.

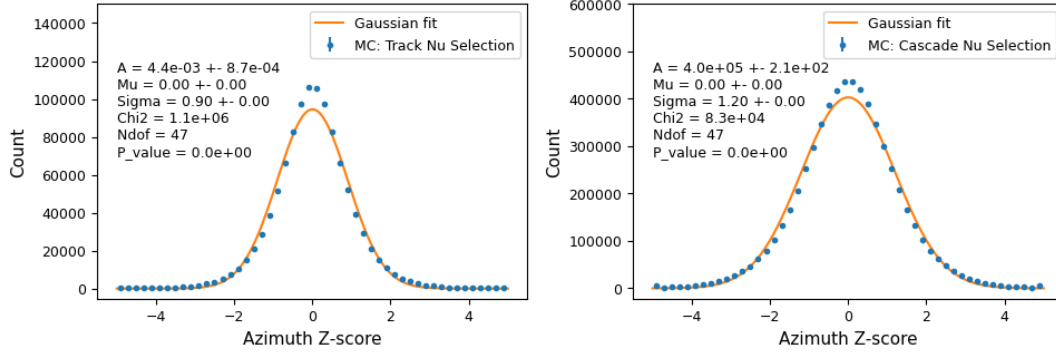


FIGURE D.8: Azimuth pull plot for track and cascade neutrino selection in Monte Carlo. Histogram of number of predicted standard deviations the predicted azimuth angle falls from the truth. Overlaid are Gaussian fits, since they should correspond to unit Gaussians, if the std's is estimated correctly. A is the normalisation, μ is the mean, σ is the standard deviation, Chi2 is the χ^2 value of the fit, N_{dof} is the degrees of freedom and p -value is the probability of obtaining a worse χ^2 if the data distributions is consistent with the fit. The means of the Gaussians are 0, while the σ is 0.90 for the track neutrino selection and 1.20 for the cascade neutrino selection. This means the model slightly underestimates the azimuth standard deviations for cascade neutrinos and overestimate it for track neutrinos.

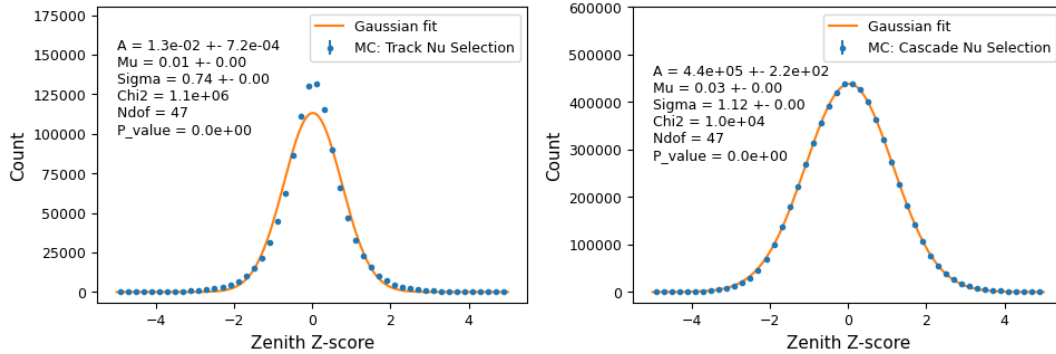


FIGURE D.9: Zenith pull plot for track and cascade neutrino selection in Monte Carlo. Histogram of number of predicted standard deviations the predicted zenith angle falls from the truth. Overlaid are Gaussian fits, since they should correspond to unit Gaussians, if the std's is estimated correctly. A is the normalisation, μ is the mean, σ is the standard deviation, Chi2 is the χ^2 value of the fit, N_{dof} is the degrees of freedom and p -value is the probability of obtaining a worse χ^2 if the data distributions is consistent with the fit. The means of the Gaussians are almost 0, while the σ is 0.74 for the track neutrino selection and 1.12 for the cascade neutrino selection. This means the model slightly underestimates the zenith standard deviations for cascade neutrinos and overestimate it for track neutrinos.

D.2 Un-scaled And Variations Of Multiclass Probability (Logit) And Track Probability Plots

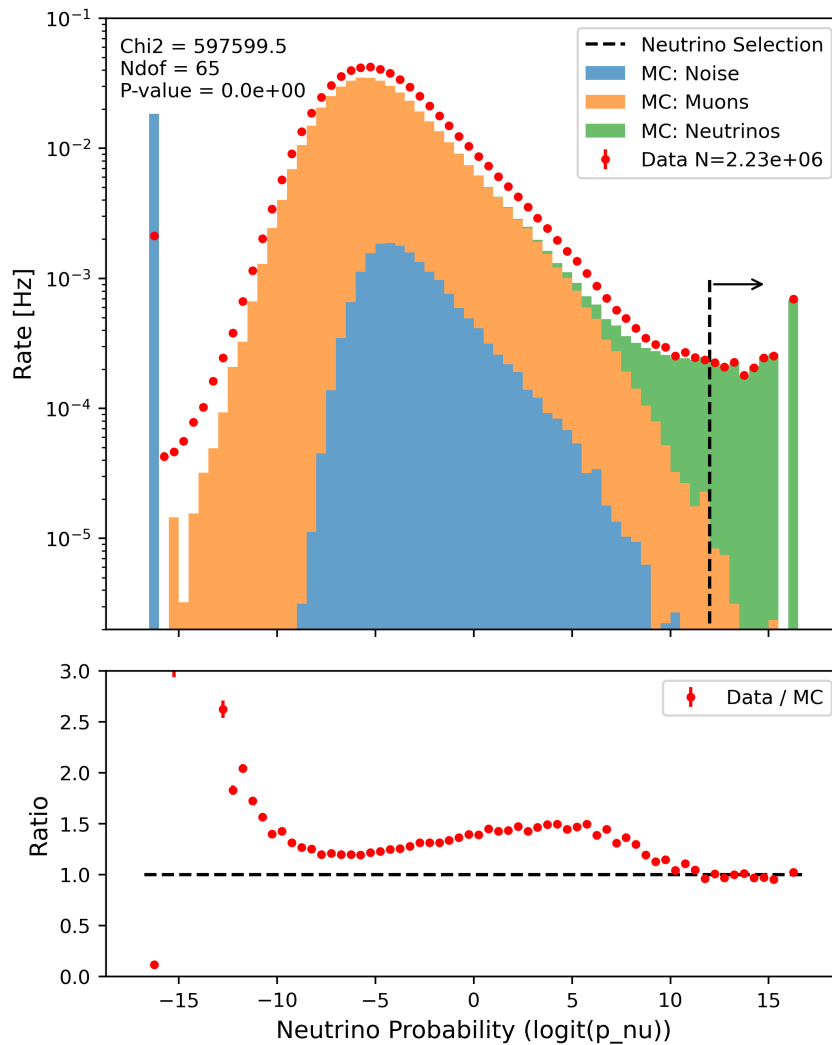


FIGURE D.10: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos. Not scaled! (Bottom) Ratio of data to Monte Carlo total rate.

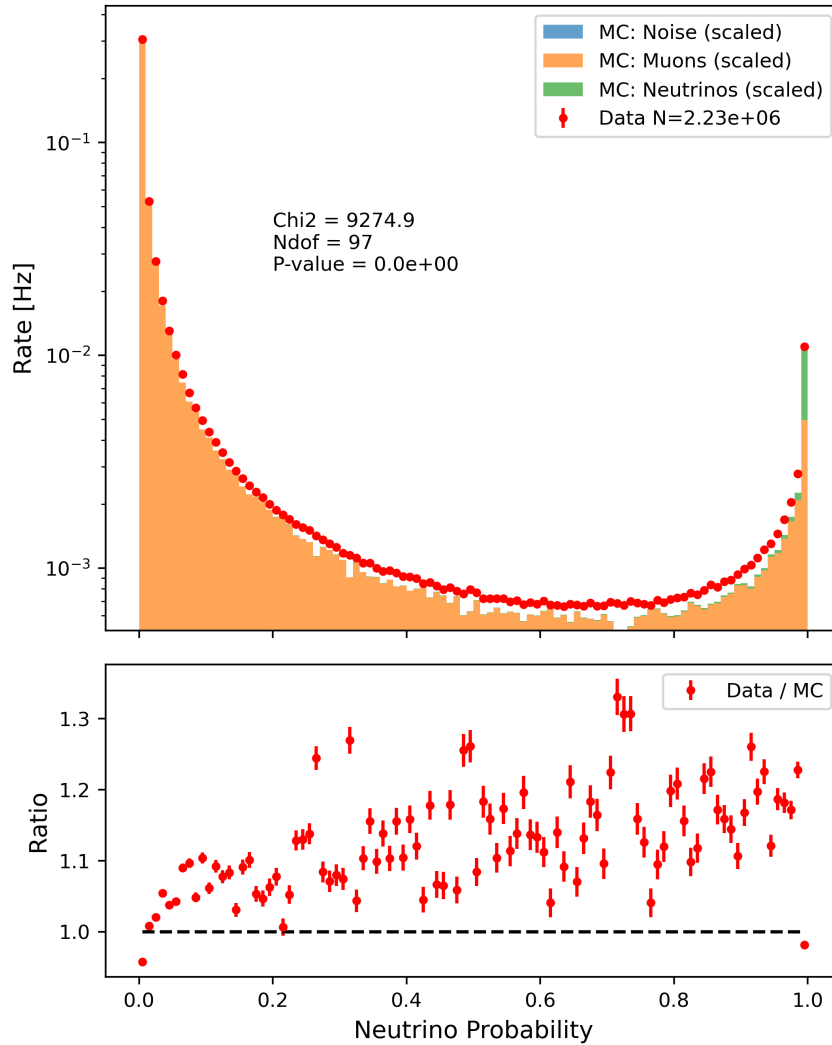


FIGURE D.11: (Top) Stacked histogram of neutrino probabilities for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. Noise, muons and neutrinos scaled by 0, 1.318 and 1.293. (Bottom) Ratio of data to Monte Carlo total rate. Noise is scaled to 0, since it has a similar distribution to the muons, but much smaller. This is one of the reasons why the logit space is better to use.

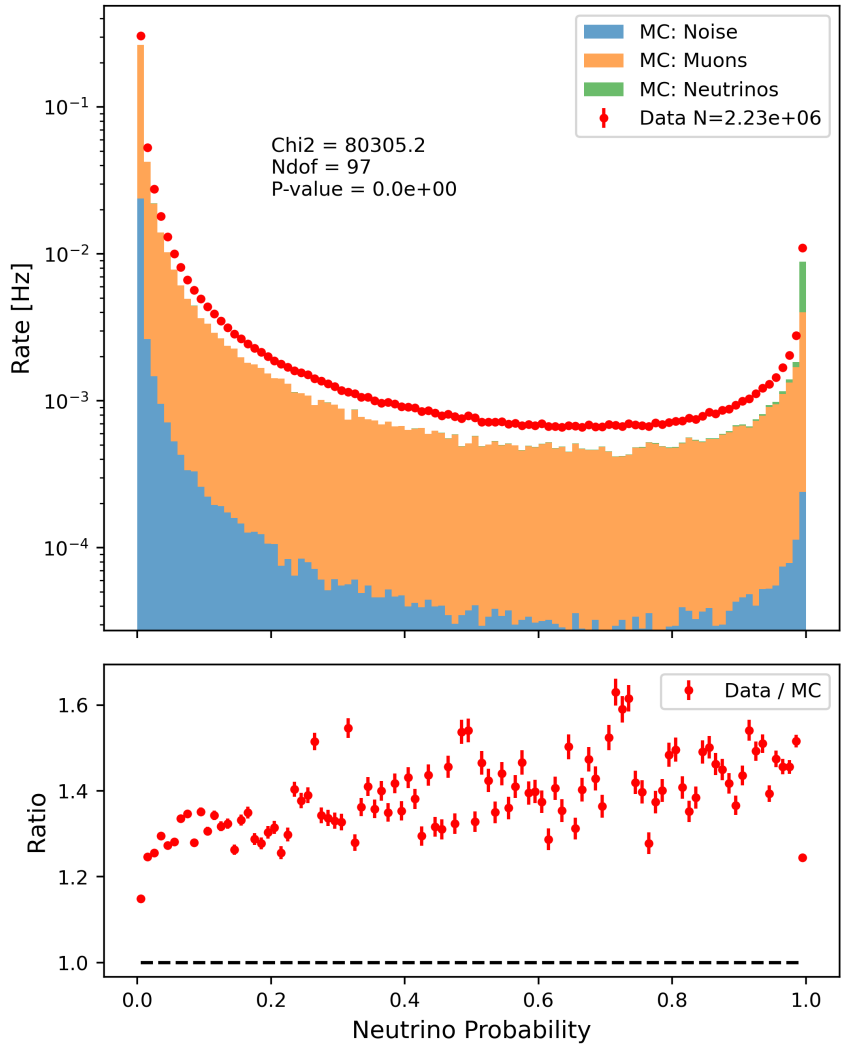


FIGURE D.12: (Top) Stacked histogram of neutrino probabilities for the Monte Carlo distributions of noise, muons and neutrinos. Not scaled! (Bottom) Ratio of data to Monte Carlo total rate.

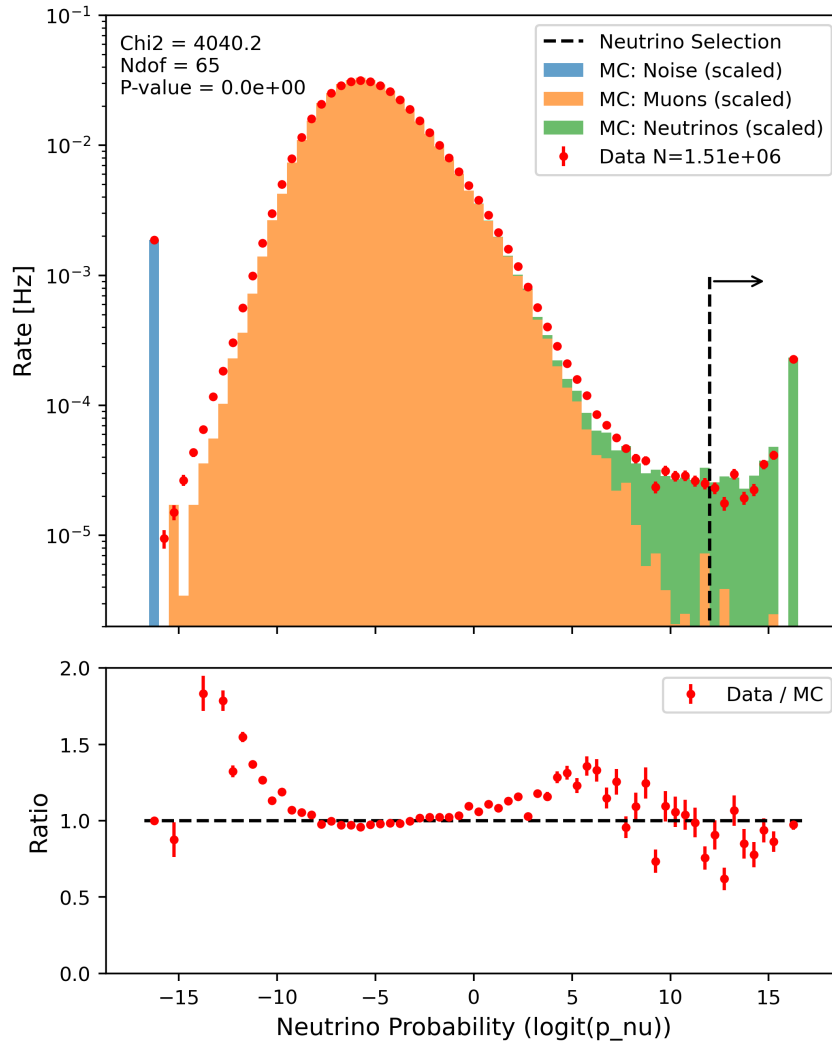


FIGURE D.13: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. Only events with track probability above 0.9 included. (Bottom) Ratio of data to Monte Carlo total rate.

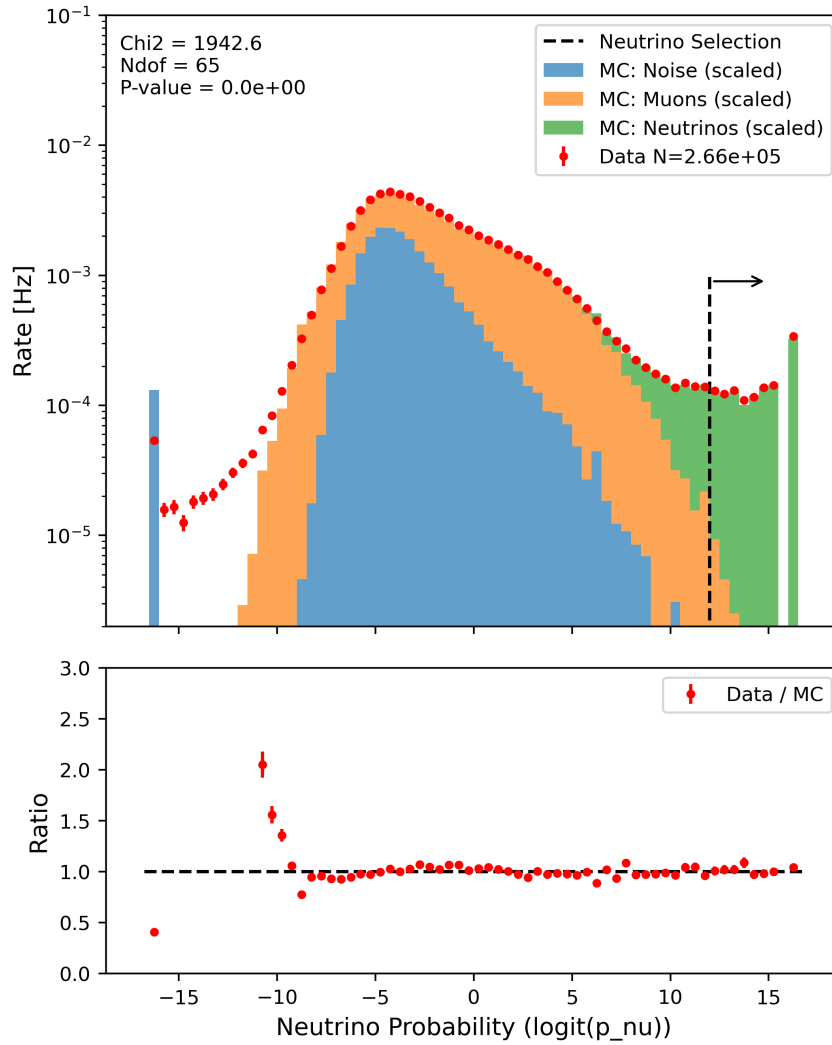


FIGURE D.14: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. Only events with track probability below 0.5 included. (Bottom) Ratio of data to Monte Carlo total rate.

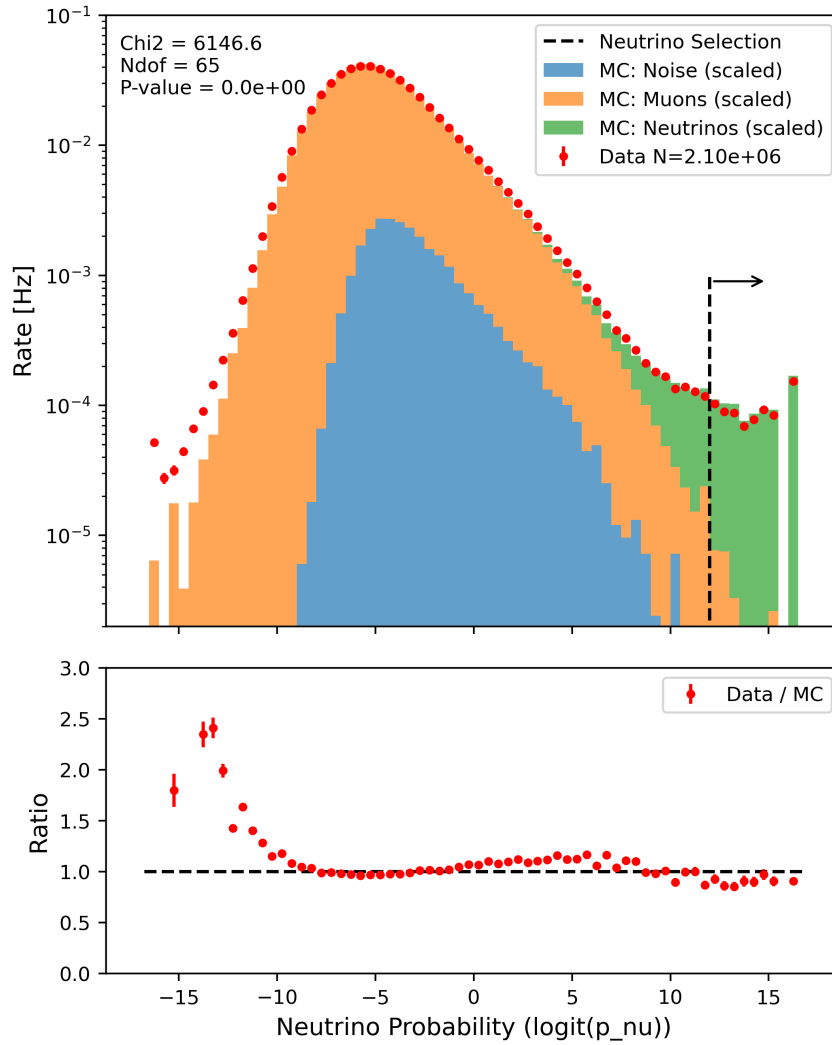


FIGURE D.15: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. Only downgoing events with zenith prediction below $\frac{\pi}{2}$. (Bottom) Ratio of data to Monte Carlo total rate.

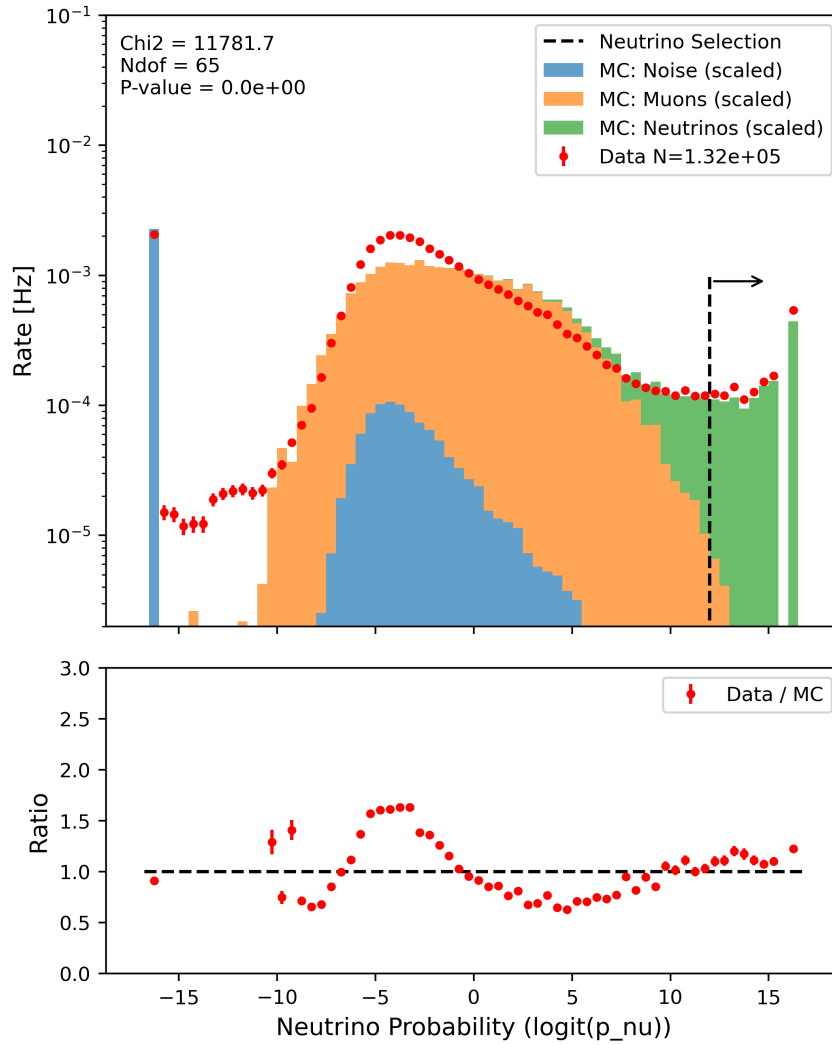


FIGURE D.16: (Top) Stacked histogram of neutrino probabilities in logit space for the Monte Carlo distributions of noise, muons and neutrinos, scaled using χ^2 minimization to match the data points which are overlaid. Only upgoing events with zenith prediction above $\frac{\pi}{2}$. (Bottom) Ratio of data to Monte Carlo total rate.

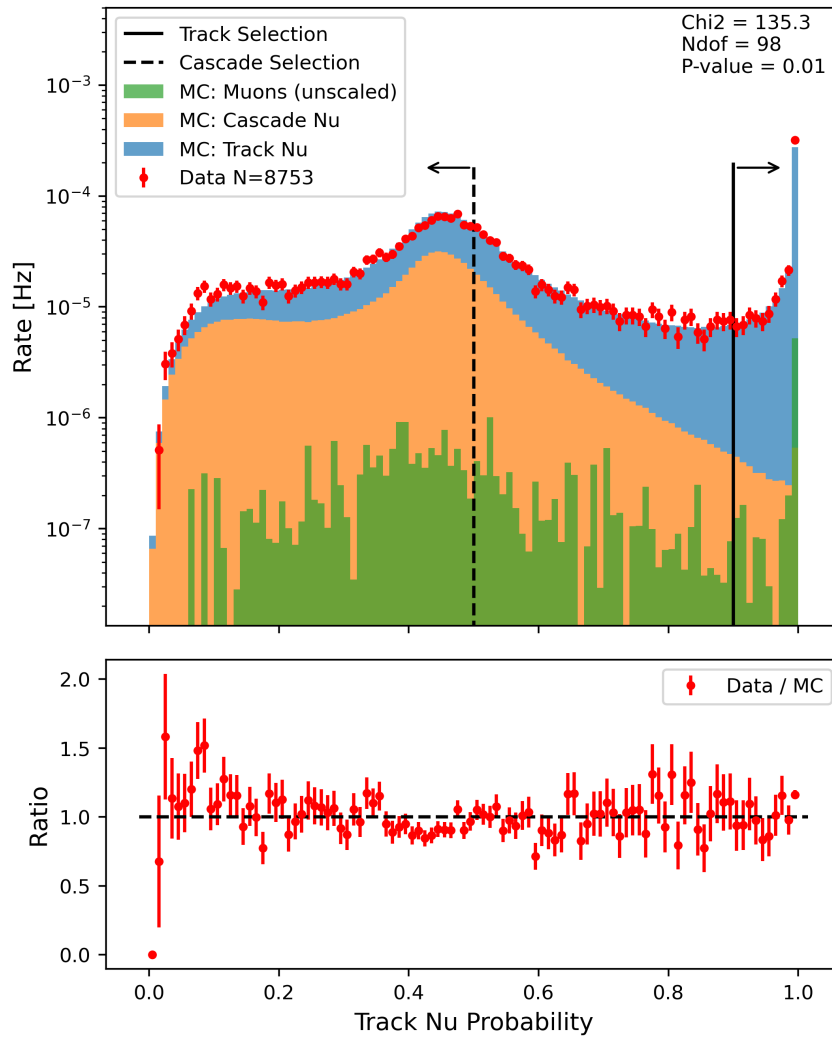


FIGURE D.17: (Top) Stacked histogram of track neutrino probabilities for the Monte Carlo distributions of track and cascade neutrinos. Not scaled! The surviving sneaky muons are also plotted. (Bottom) Ratio of data to Monte Carlo total rate. The plot shows that the data is very well described by the Monte Carlo distributions, even when unscaled, although there are areas where the ratio generally fall above or below, as seen in the ratio plot. The P-value of 0.01 also shows that the distributions are relative similar.

D.3 Real And Simulated Noise And Muons In Reconstruction and Track/Cascade Classifier Results - Where Would Contamination Show Up?

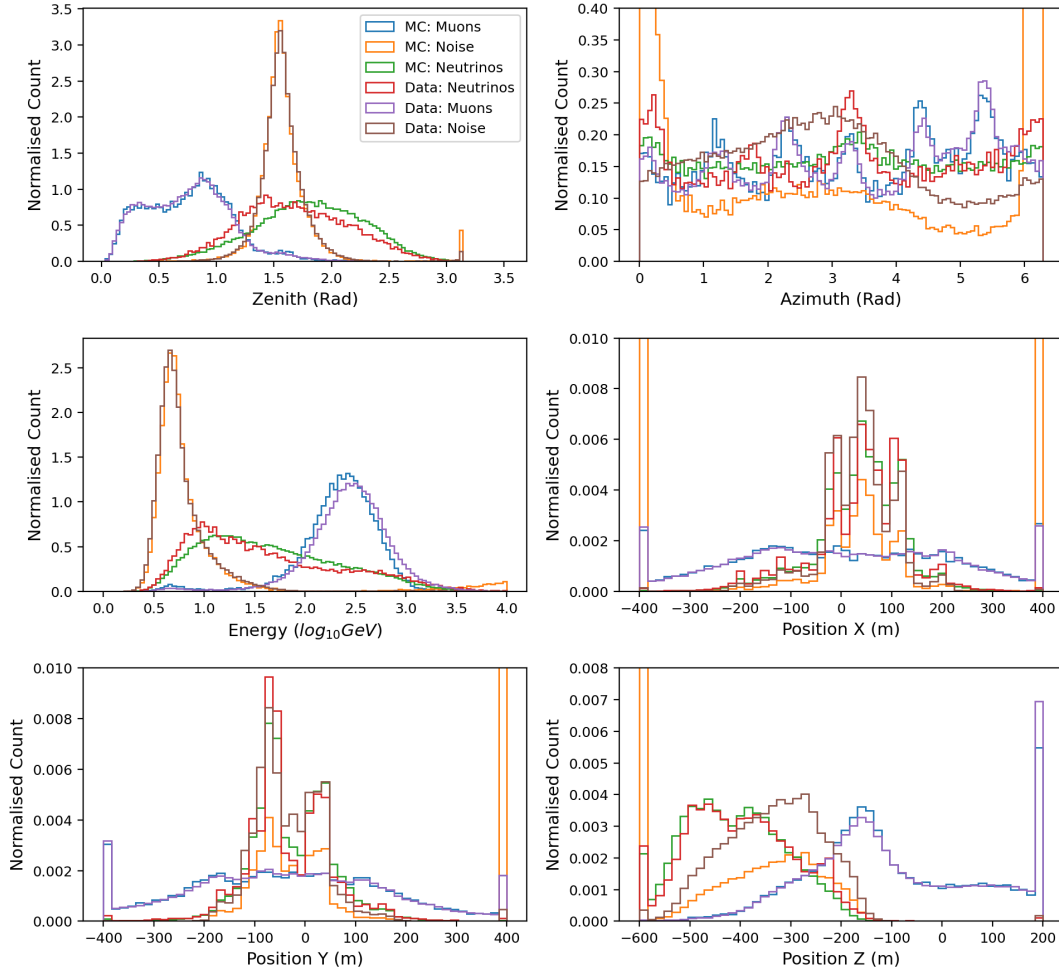


FIGURE D.18: Histograms of energy, zenith, azimuth and vertex position for neutrinos, muons and noise in data and Monte Carlo. Noise are events with noise probability > 0.5 , muons are events with muon probability > 0.5 and neutrinos are events with neutrino probability > 12 in logit space. These plots show where potential muon or noise contamination could potentially show up, although those that would end up in the neutrino selections might not resemble the depicted distributions exactly. Note that outlier bins are used.

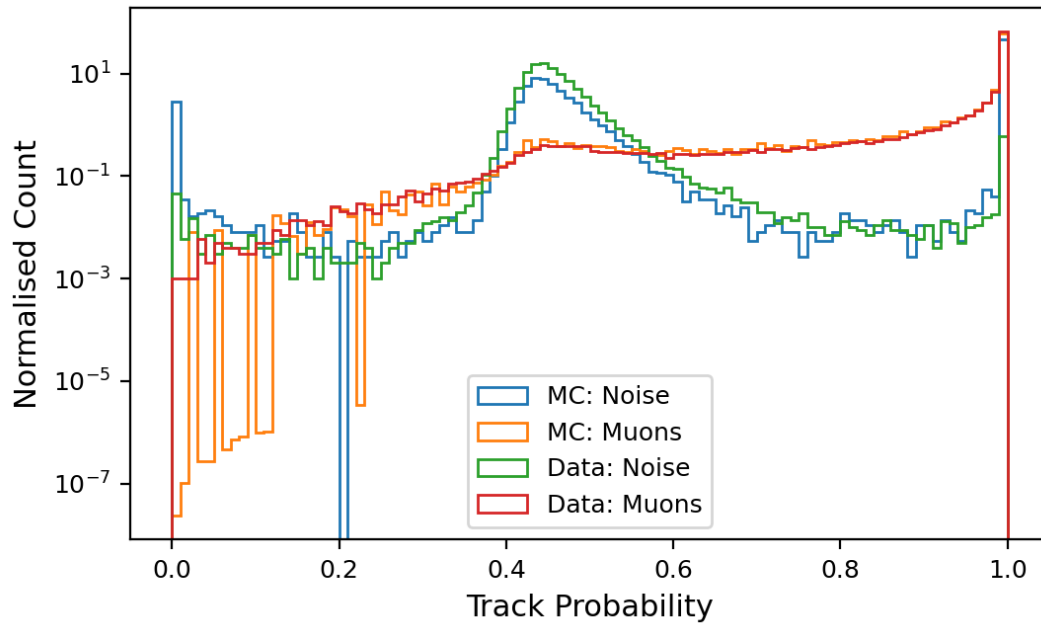


FIGURE D.19: Histograms of track probability for, muons and noise in data and Monte Carlo. Noise are events with noise probability > 0.5 , muons are events with muon probability > 0.5 . The plot show where potential muon or noise contamination could potentially show up, although those that would end up in the neutrino selections might not resemble the depicted distributions exactly.

Appendix E

Comparison Of GraphNeT Neutrino Selection With OscNext Neutrino Selection - Additional Figures

E.1 Additional Figures Showing Neutrino Rates For Graph- NeT And OscNext Selections

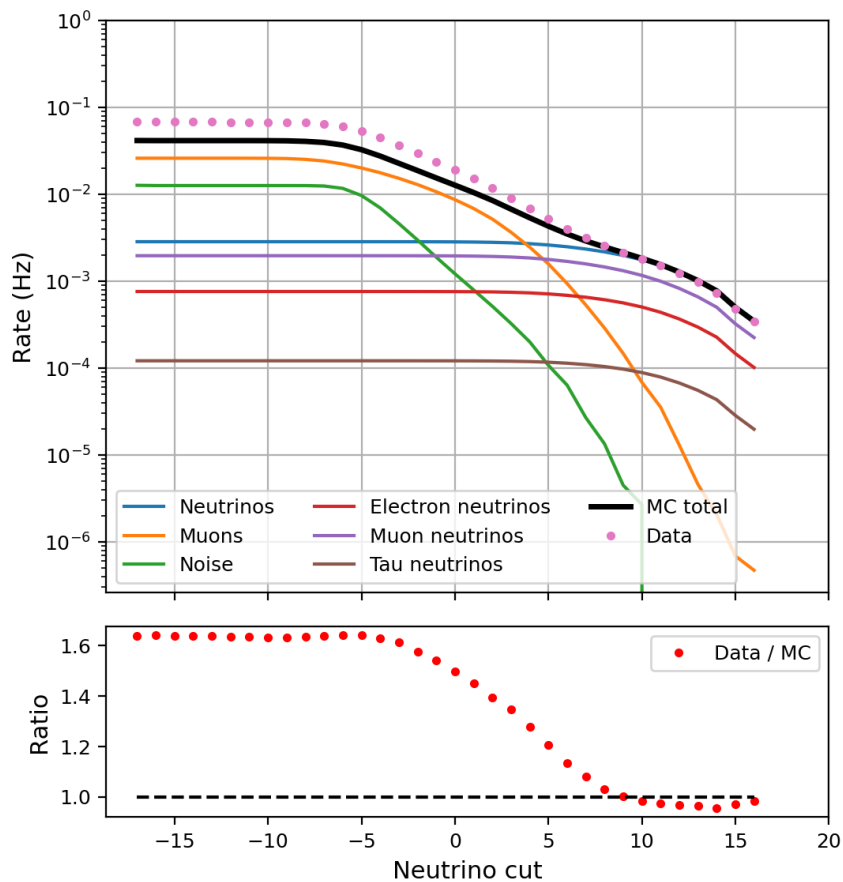


FIGURE E.1: (Top) Survival rate of each particle type in Monte Carlo depending on where the cut is made in logit neutrino probability space. Also plotted are the survival data rate. Note that the muons are from sample nr: 130000 and that only events with track probability below 0.5 are included. (Bottom) ratio plot of total RD rate to MC rate.

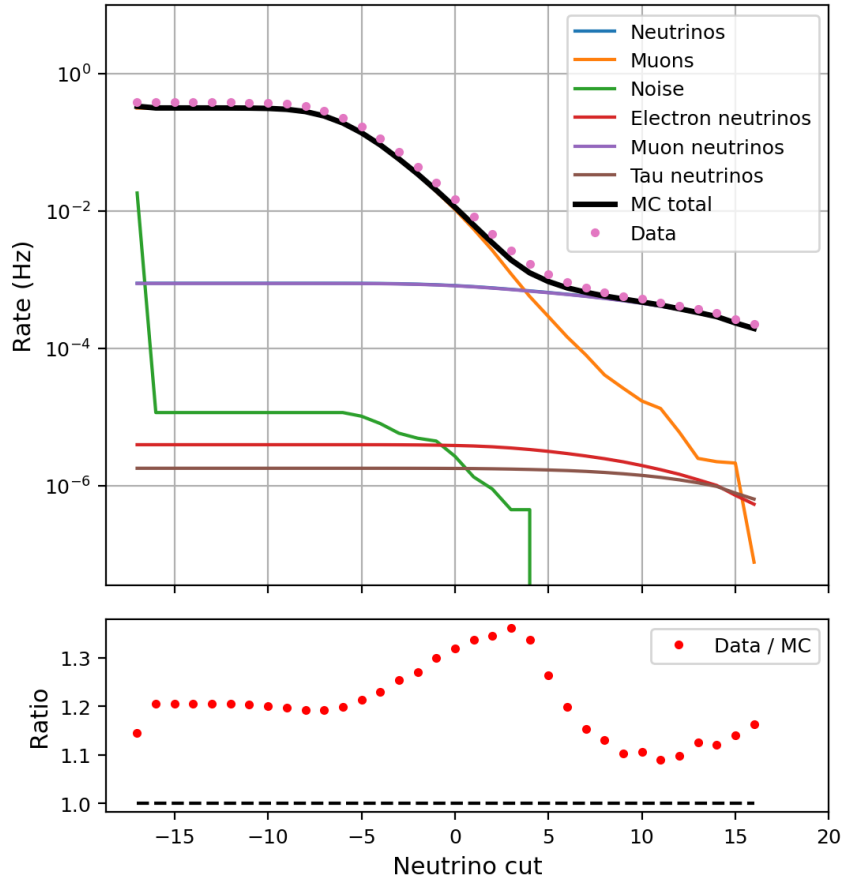


FIGURE E.2: (Top) Survival rate of each particle type in Monte Carlo depending on where the cut is made in logit neutrino probability space. Also plotted are the survival data rate. Note that the muons are from sample nr: 130000 and that only events with track probability above 0.9 are included. (Bottom) ratio plot of total RD rate to MC rate.

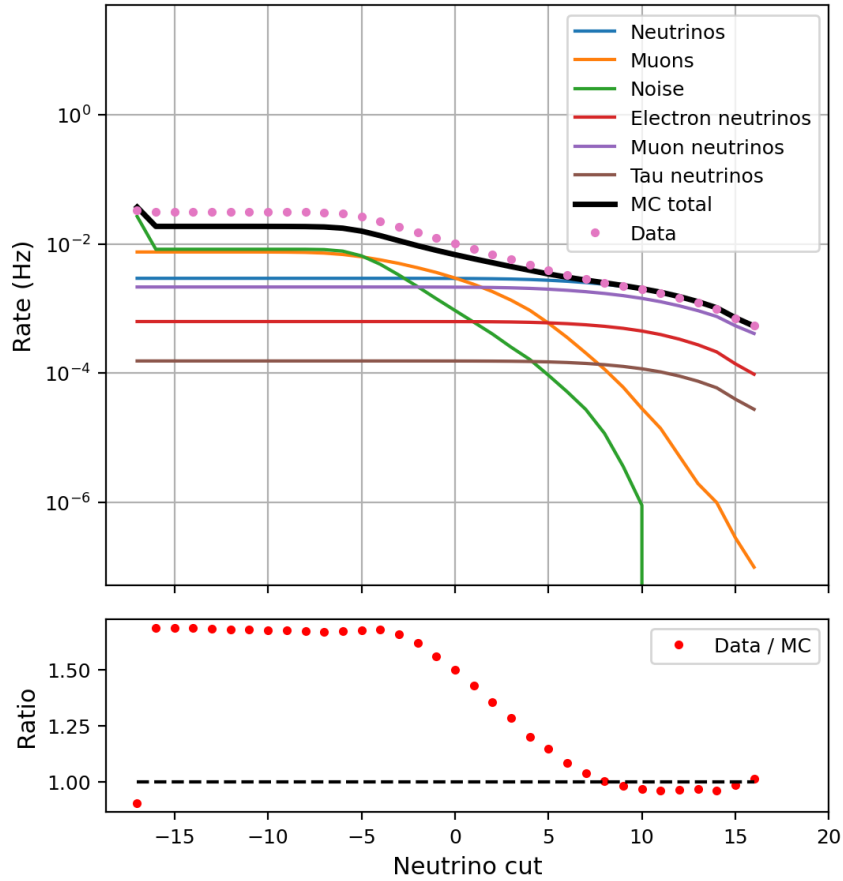


FIGURE E.3: (Top) Survival rate of each particle type in Monte Carlo depending on where the cut is made in logit neutrino probability space. Also plotted are the survival data rate. Note that the muons are from sample nr: 130000 and that only upgoing events with predicted zenith above $\frac{\pi}{2}$. (Bottom) ratio plot of total RD rate to MC rate.

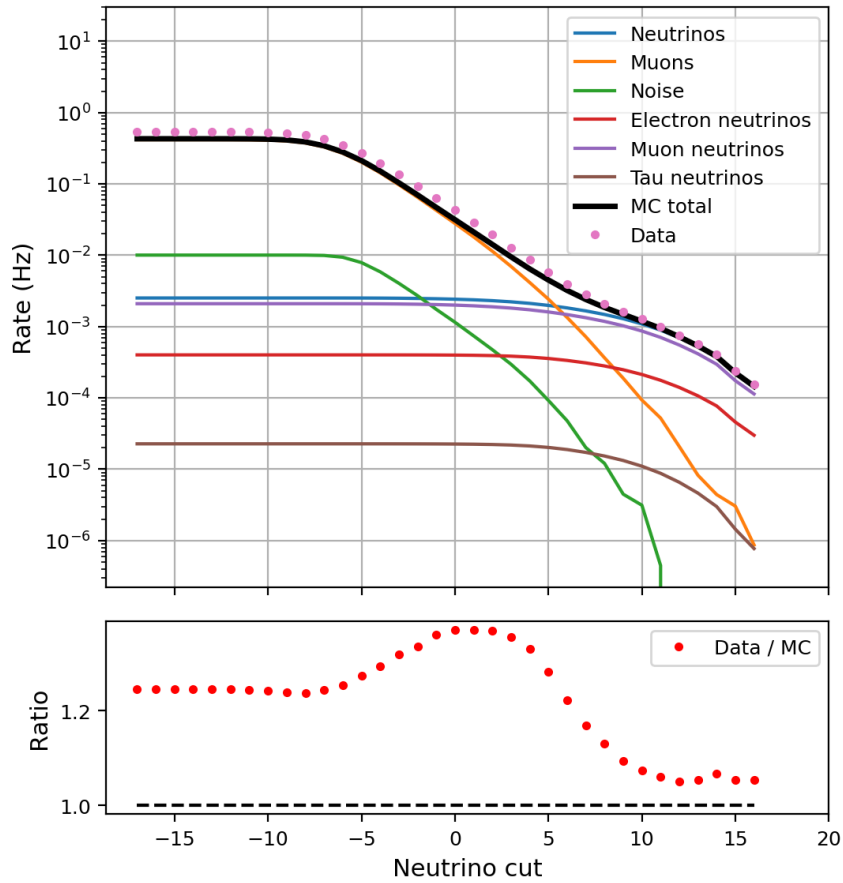


FIGURE E.4: (Top) Survival rate of each particle type in Monte Carlo depending on where the cut is made in logit neutrino probability space. Also plotted are the survival data rate. Note that the muons are from sample nr: 130000 and that only downgoing events with predicted zenith below $\frac{\pi}{2}$. (Bottom) ratio plot of total RD rate to MC rate.

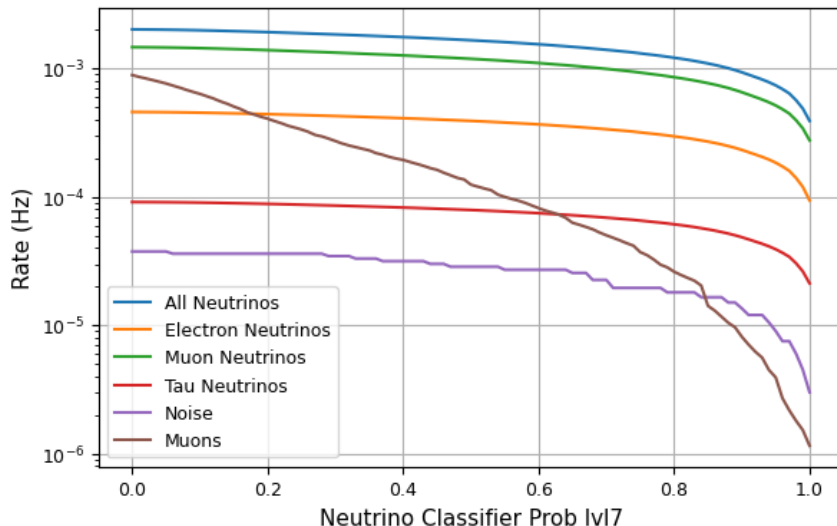


FIGURE E.5: OscNext lvl7 Neutrino Classifier showing the rate of particles surviving a particular cut in neutrino probability. As a part of the OscNext cleaning level 7, only events with neutrino probability above 0.8 is used.

E.2 Additional Figures Showing Difference In Selected Neutrino Properties

In figures E.6, E.7 the interaction vertex coordinate distributions are shown for the x and y coordinates. Some of the DynEdge exclusive neutrinos fall in the edges of the detector, but most of them fall in the same range as the OscNext neutrinos, which is promising.

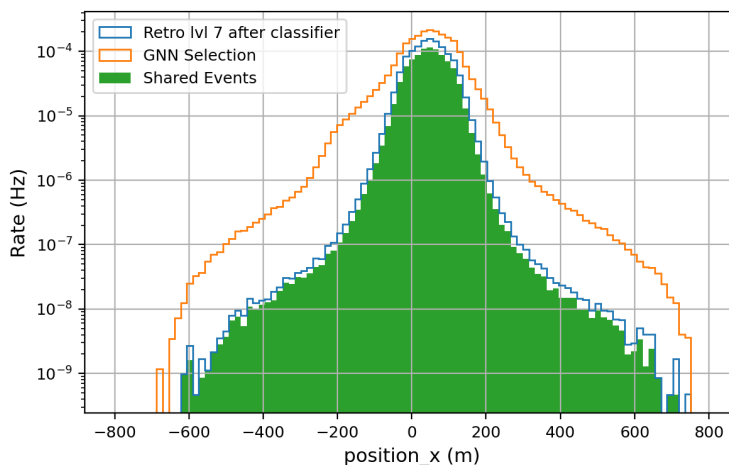


FIGURE E.6: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Vertex position x coordinate histograms of shared events and events in the two selections are plotted.

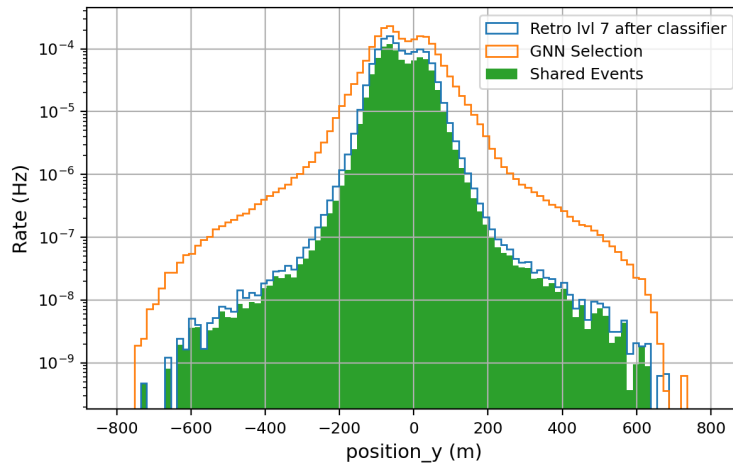


FIGURE E.7: Histograms comparing OscNext neutrinos with Graph-NeT neutrinos in Monte Carlo. Vertex position y coordinate histograms of shared events and events the two selections are plotted.

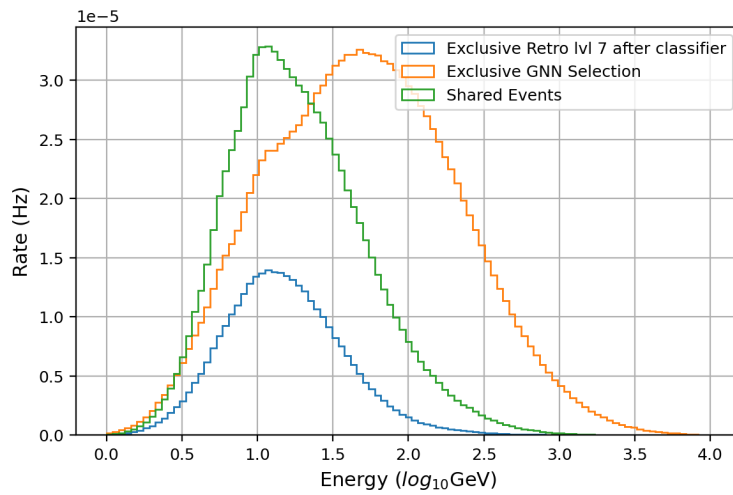


FIGURE E.8: Histograms comparing OscNext neutrinos with Graph-NeT neutrinos in Monte Carlo. Energy histograms of shared events and events exclusive to the two selections are plotted.

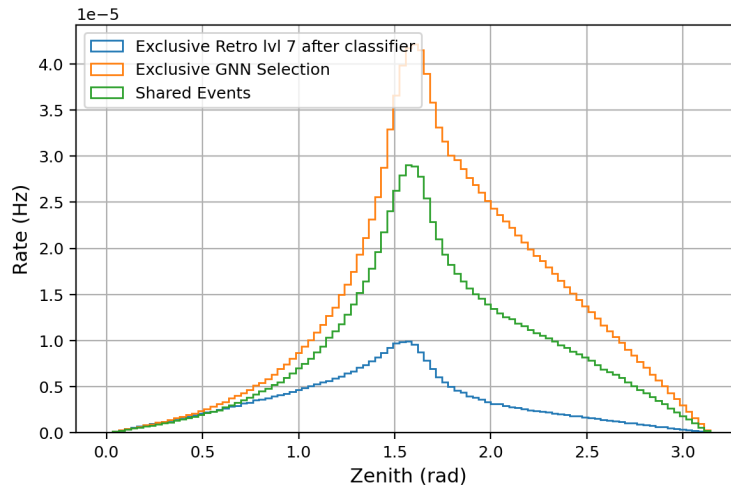


FIGURE E.9: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo. Zenith histograms of shared events and events exclusive to the two selections are plotted.

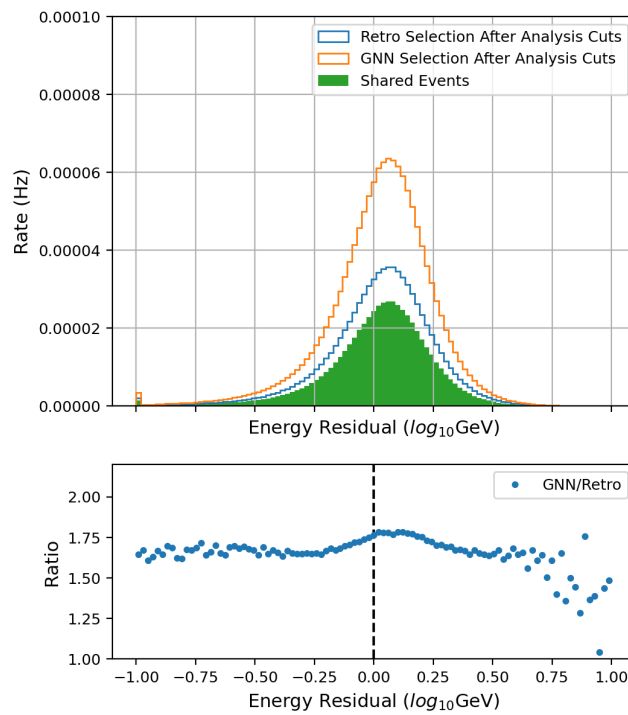


FIGURE E.10: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo after analysis cuts. (Top) Energy residual histograms of shared events and events in the two selections are plotted. (Bottom) Ratio of GraphNeT rate to OscNext rate is plotted.

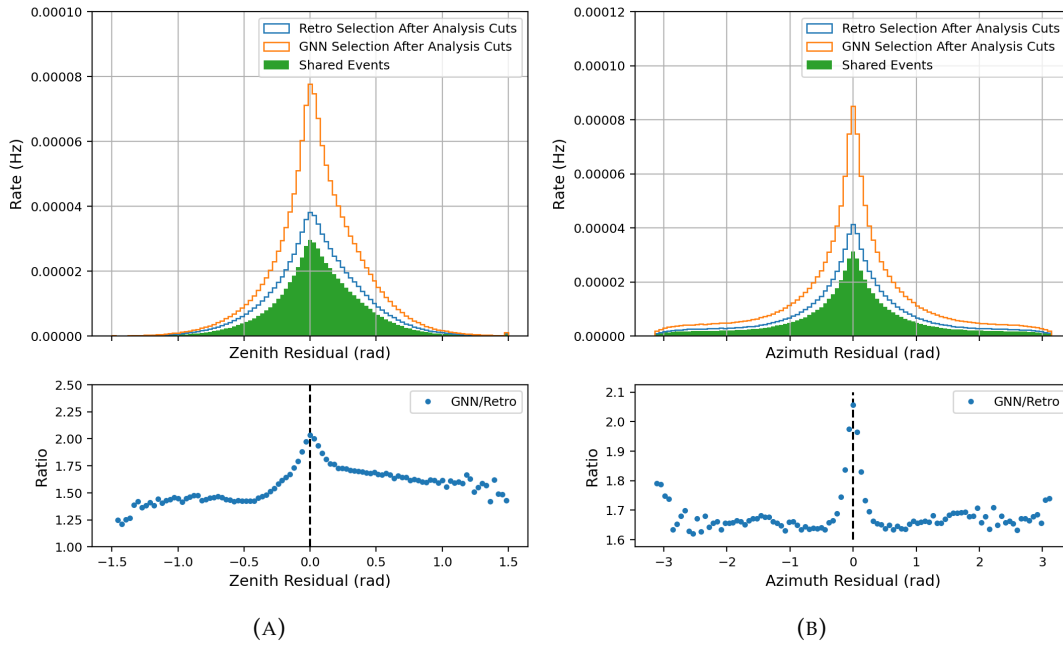


FIGURE E.11: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo after analysis cuts. Zenith residual (A) and azimuth residual (B) histograms of shared events and events in the two selections are plotted.

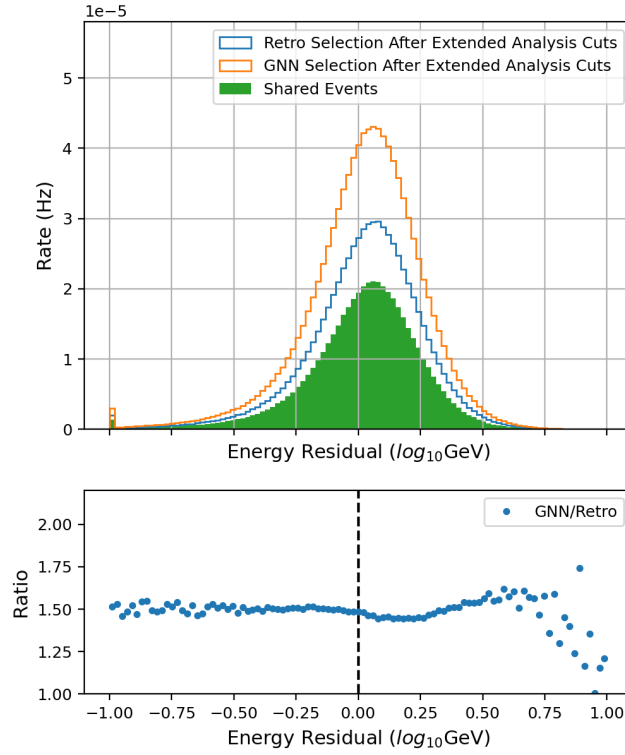


FIGURE E.12: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo after analysis cuts and a stricter energy cut that removes neutrinos with an energy above $10^{1.75}$ GeV = 56.23 GeV. (Top) Energy residual histograms of shared events and events in the two selections are plotted. (Bottom) Ratio of GraphNeT rate to OscNext rate is plotted.

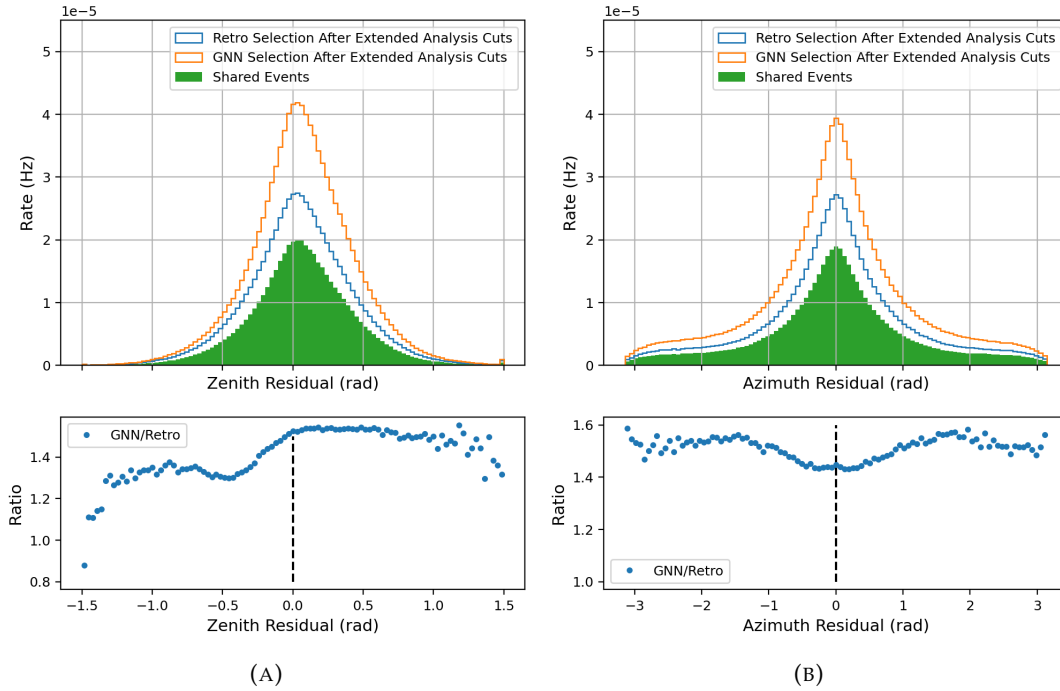


FIGURE E.13: Histograms comparing OscNext neutrinos with GraphNeT neutrinos in Monte Carlo after analysis cuts and a stricter energy cut that removes neutrinos with an energy above $10^{1.75}$ GeV = 56.23 GeV. Zenith residual (A) and azimuth residual (B) histograms of shared events and events in the two selections are plotted.

E.3 Additional Detector Signature Plots for Data Track and Cascade Neutrinos

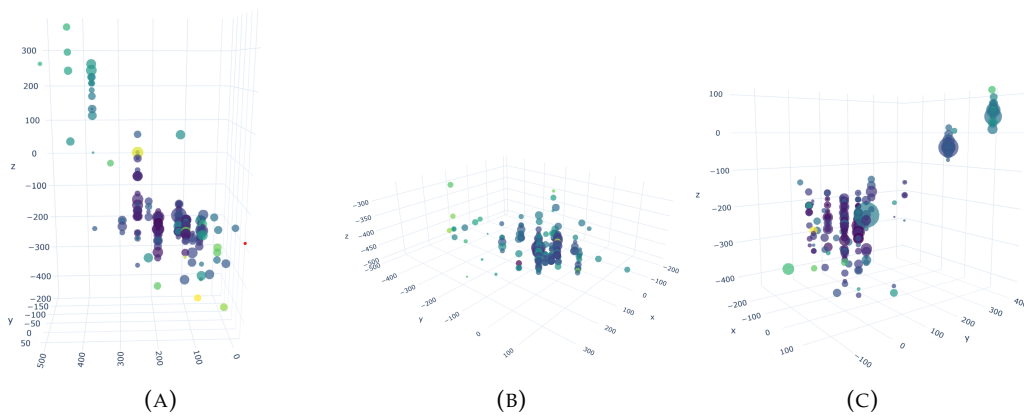


FIGURE E.14: Detector signature for three highest energy data track neutrinos. (A) Event number 36682470, predicted energy 3811 GeV. (B) Event number 11788673, predicted energy 3403 GeV. (C) Event number 90278349, predicted energy 2883 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

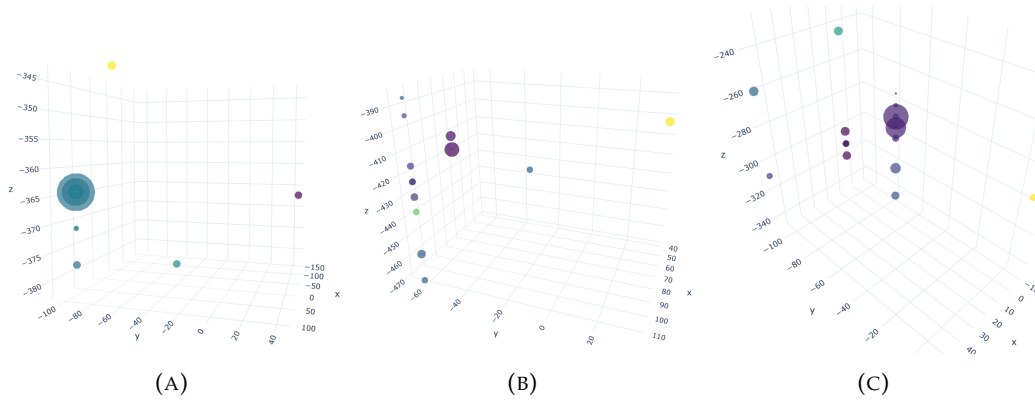


FIGURE E.15: Detector signature for three lowest energy data track neutrinos. (A) Event number 88543974, predicted energy 5.1 GeV. (B) Event number 41783681, predicted energy 11.3 GeV. (C) Event number 40753130, predicted energy 13.2 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

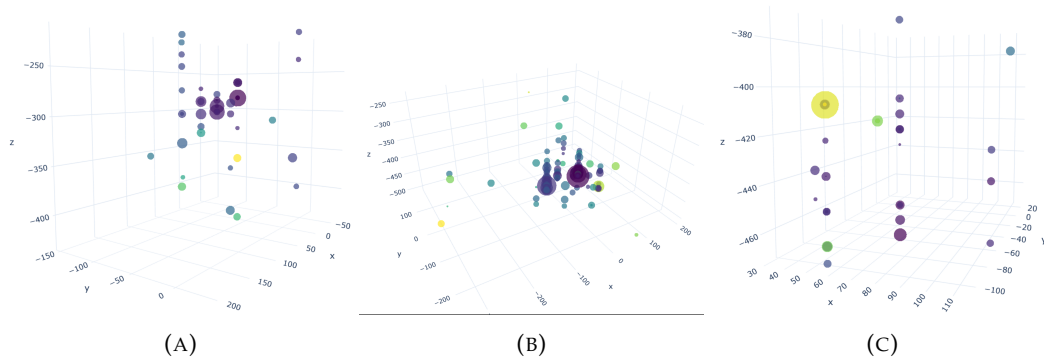


FIGURE E.16: Detector signature for three random data track neutrinos. (A) Event number 94601659, predicted energy 49.1 GeV. (B) Event number 3738302, predicted energy 507.2 GeV. (C) Event number 55066778, predicted energy 21.3 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

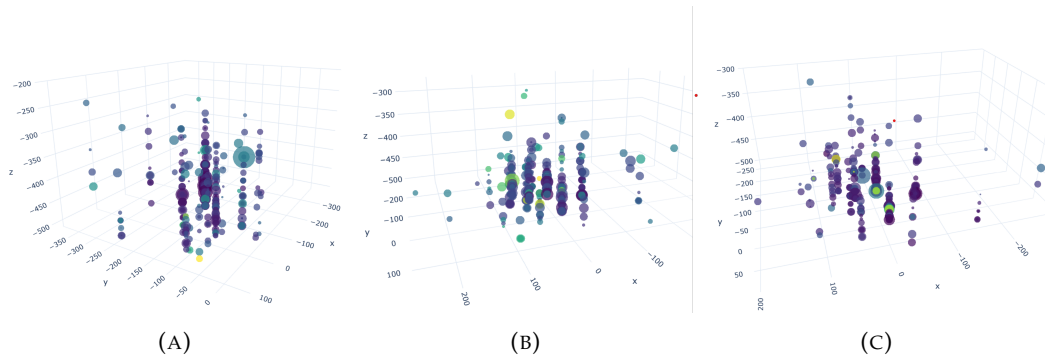


FIGURE E.17: Detector signature for three highest energy data cascade neutrinos. (A) Event number 19926659, predicted energy 3434 GeV. (B) Event number 34882051, predicted energy 2796 GeV. (C) Event number 26001415, predicted energy 2143 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

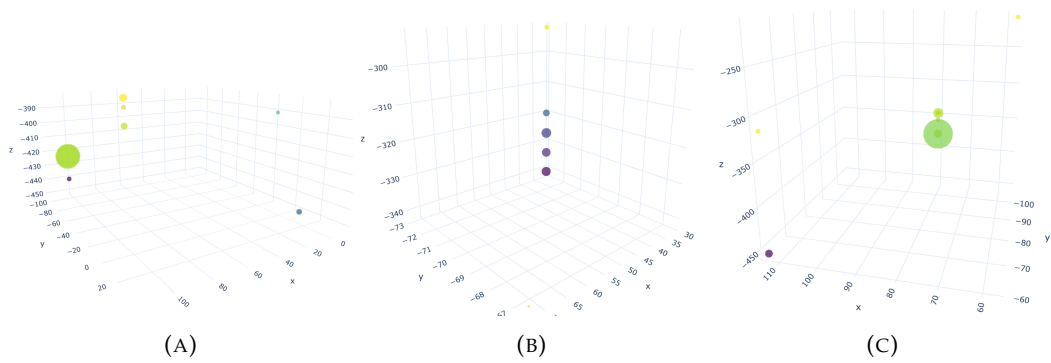


FIGURE E.18: Detector signature for three lowest energy data cascade neutrinos. (A) Event number 57569928, predicted energy 2.1 GeV. (B) Event number 19108109, predicted energy 2.7 GeV. (C) Event number 6870032, predicted energy 2.9 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

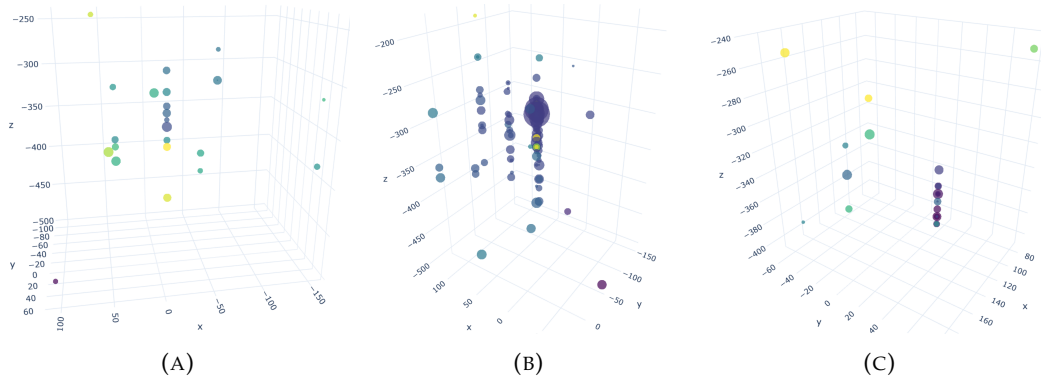


FIGURE E.19: Detector signature for three random data cascade neutrinos. (A) Event number 25849196, predicted energy 89.7 GeV. (B) Event number 3361334, predicted energy 86.0 GeV. (C) Event number 37384179, predicted energy 32.7 GeV. Size of spheres represent deposited charge and color represents relative time, with dark colors as the earliest hits and light colors as the latest hits. The noise-cleaned pulsemap SRTIcePulses is used.

Appendix F

Data sources and details

F.1 Where is everything located on the HEP server at NBI?

In order to access the databases, models, and reconstruction/classification variables used in this work, you need access to the HEP server on NBI. There you should locate Troels Christian Petersens folder at groups/icecube/petersen. Here you should find a folder called GraphNetDatabaseRepository. All paths below are starting from this location.

F.1.1 Databases And How To Match Events Between Them

It is important to understand that four main sqlite databases are used in this work. Two of those contain OscNext low energy Monte Carlo data and are illustrated in figure F.1. The third contains the actual data from the OscNext burnsample. The fourth are Northern Track high energy Monte Carlo data used in the Northern Track benchmark.

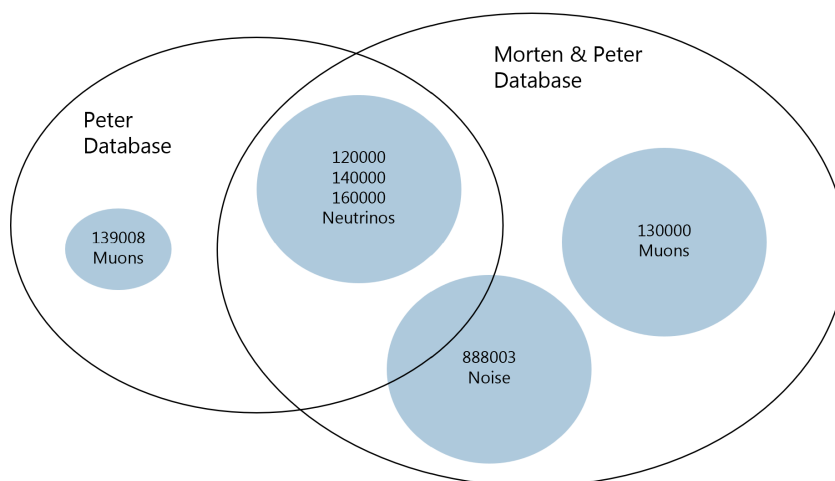


FIGURE F.1: Illustration of the Peter database and the Morten and Peter database. Both contain all the 12/14/160000 neutrinos, but the Morten and Peter database contains the 130000 muon set and a lot more of the 888003 noise than the Peter database, which contains the 139008 muon set.

All four database locations are available in table F.1. Furthermore, the green row contains the location of the most important csv files used in this work. There are

9 files located at this location. 3 for the burnsample database (prefix: Burnsam-ple_1_percent), 3 for the peter database (prefix: Monte_Carlo) and 3 for the peter and morten database (prefix: Old_muon_more_noise_Monte_Carlo). Each of the triplets have one file with all events, one with events that have a neutrino probability in logit space above 1 and one with events with a neutrino probability in logit space above 12. How to read in and apply these csv files can be seen in the plotting scripts in the [analysis on GitHub](#). Note that one should use the neutrinos from the peter database, along with either the noise and muons from the same database OR from the Morten and Peter database.

It is also important to be aware that the 'event_no' column in each database aren't compatible across databases. In table F.1, some rows share a color, indicating that they contain the same events, but different information about them. To merge events from different sqlite databases, do NOT use the 'event_no' column, but instead look at the columns: RunID, SubRunID, EventID, and SubEventID in the truth table. These are indeed unique across sqlite databases for the same events.

What	Path from /groups/icecube/petersen/GraphNetDatabaseRepository
CSV files with all information for both Monte Carlo and real data in the neutrino selection analysis	multi_classification_track_cascade_neutrino/using_new_muons_Peter_database/inference/track_cascade_sets
Database Peter Muons 139008	osc_next_database_new_muons_peter/Merged_db/osc_next_level3_v2.00_genie_muongun_noise_120000_140000_160000_139008_888003.db
Database Peter Muons 139008 Only lvl 3 variables.	osc_next_database_new_muons_peter/Merged_db_3/NOT_full_db_osc_next_level3_v2.00_genie_muongun_noise_120000_140000_160000_139008_888003_truth_and_lvl3_variables.db
Database Peter Muons 139008 Train_validation_test split	osc_next_database_new_muons_peter/train_val_test_split
Database Morten and Peter Muons 130000	osc_next_database_Peter_and_Morten/merged_database/osc_next_level3_v2.00_genie_muongun_noise_120000_140000_160000_130000_888003_retro.db
Burnsample database	dev_lvl3_genie_burnsample/dev_lvl3_genie_burnsample_v5.db
Burnsample database Only lvl3 variables	osc_next_0.01_percent_burnsample_Peter/merged_db/Burnsample_lvl3_v02.00_lvl3_variables.db
Northern Track Benchmark database and train/val/test split	/groups/icecube/petersen/GraphNetDatabaseRepository/northern_tracks/dev_northern_tracks_full_part_1

TABLE F.1: Table containing locations to all databases used in this work on HEP, along with the location of the CSV files that contain all predictions and truths used in chapter 7.

F.1.2 Trained Model State Dictionaries

In table F.2, the locations of the state dictionaries for each trained model in this work is available. Using one of the inference scripts in the [analysis on GitHub](#), one can then use the model on other events.

Model	Path to State dict from /groups/icecube/petersen/GraphNetDatabaseRepository
Multiclassification	multi_classification_track_cascade_neutrino/using_new_muons_Peter_database/trained_models/osc_next_level3_v2/dynedge_pid_Peter_new_muon_db_SplitInIcePulses_equal_frac_run_2_validation_set/dynedge_pid_Peter_new_muon_db_SplitInIcePulses_equal_frac_run_2_validation_set_state_dict.pth
Zenith Reco	multi_classification_track_cascade_neutrino/using_MP_lv13/trained_models/osc_next_level3_v2/MP_data_zenith_1_mill_even_track_cascade_attempt2_test_set/MP_data_zenith_1_mill_even_track_cascade_attempt2_test_set_state_dict.pth
Azimuth Reco	multi_classification_track_cascade_neutrino/using_MP_lv13/trained_models/osc_next_level3_v2/MP_data_azimuth_test_1_mill_attempt2_test_set_equal_track_cascade/MP_data_azimuth_test_1_mill_attempt2_test_set_state_dict.pth
Energy Reco	multi_classification_track_cascade_neutrino/using_MP_lv13/trained_models/osc_next_level3_v2/Peter_Morten_energy_1_mill_even_track_cascade_attempt_2_test_set/Peter_Morten_energy_1_mill_even_track_cascade_attempt_2_test_set_state_dict.pth
Interaction Vertex Position Reco	multi_classification_track_cascade_neutrino/using_new_muons_Peter_database/trained_models/osc_next_level3_v2/dynedge_['position_x', 'position_y', 'position_z']_Peter_new_muon_db_SplitInIcePulses_position_test_set/dynedge_['position_x', 'position_y', 'position_z']_Peter_new_muon_db_SplitInIcePulses_position_test_set_state_dict.pth
Track Cascade Classification	multi_classification_track_cascade_neutrino/using_MP_lv13/trained_models/osc_next_level3_v2/dynedge_track_mu_Track_cascade_MP_data_SplitInIcePulses_on_equal_track_cascade_neutrinos_test/dynedge_track_mu_Track_cascade_MP_data_SplitInIcePulses_on_equal_track_cascade_neutrinos_test_state_dict.pth

TABLE F.2: Locations of all trained model dictionaries on HEP used in the analysis in chapter 7.

F.1.3 How To Reweight Events To Get The Estimated Rate?

In general the comparison between Monte Carlo and real data is carried out using rates. For Monte Carlo, the `osc_weight` variable from Retro is used. It is located in the retro table in sqlite databases and is a weight, which when applied to the events, recreates the estimated rates of actual data across angles and energy. Note that this weight is appropriate when using a single I3 file. Thus if more are used, the `osc_weight` needs to be scaled down by the inverse of the number of I3 files used. This should be done for each particle type separately. For instance, if you have used 2000 I3 files with muon neutrinos, the `osc_weight` is scaled by $\frac{1}{2000}$. To get a good estimate of the actual data rate, you need the run time for each of the data subruns you use. Then you can weight each real data event by the inverse of the total run time for all your data events. The run times for the real data approx. 1% burnsample used in this work are available here:

`osc_next_0.01_percent_burnsample_Peter/Run_times_to_get_rate`. Again this path begins from the GraphNetDatabaseRepository.

In the final CSV files (green in table F.1), the variable `total_osc_weight` is ready to use. It has been scaled by the appropriate factors.

F.1.4 Final Details

As mentioned in the Readers Guide, python scripts for training and deploying the GNN models can be found in this location [Analysis on GitHub](#). All plotting scripts are located there as well.

To be able to run the scripts that require using or training GNNs, this branch of GraphNeT should be compatible: [Compatible GraphNeT Branch](#).

Finally, if anything is unclear, feel free to reach out to the author at peterandresen@hotmail.dk.

Bibliography

- [1] R. Abbasi et al. “Graph Neural Networks for low-energy event classification & reconstruction in IceCube”. In: *Journal of Instrumentation* 17.11 (2022), P11003. DOI: [10.1088/1748-0221/17/11/p11003](https://doi.org/10.1088/1748-0221/17/11/p11003). URL: <https://doi.org/10.1088/1748-0221/17/11/p11003> (cit. on pp. 1, 4, 5, 38–42, 53, 56, 58, 65, 67, 68, 70, 100, 101).
- [2] M.G. Aartsen et al. “Astrophysical Neutrinos and Cosmic Rays Observed by IceCube”. In: *Advances in Space Research* 62 (May 2017). DOI: [10.1016/j.asr.2017.05.030](https://doi.org/10.1016/j.asr.2017.05.030) (cit. on p. 2).
- [3] Markus Ahlers, Klaus Helbing, and Carlos Pérez de los Heros. “Probing particle physics with IceCube”. In: *The European Physical Journal C* 78.11 (2018). DOI: [10.1140/epjc/s10052-018-6369-9](https://doi.org/10.1140/epjc/s10052-018-6369-9). URL: <https://doi.org/10.1140/epjc/s10052-018-6369-9> (cit. on p. 3).
- [4] Kaggle. *IceCube - Neutrinos in Deep Ice*. URL: <https://www.kaggle.com/competitions/icecube-neutrinos-in-deep-ice> (cit. on pp. 5, 52, 100).
- [5] IceCube Collaboration. *Coordinate Systems*. URL: <https://docs.icecube.aq/icetray/main/projects/dataclasses/coordinates.html> (cit. on p. 5).
- [6] R. L. Workman et al. “Review of Particle Physics”. In: *PTEP* 2022 (2022), p. 083C01. DOI: [10.1093/ptep/ptac097](https://doi.org/10.1093/ptep/ptac097) (cit. on p. 7).
- [7] Cush MissMJ. *Standard Model of Elementary Particles*. URL: https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg (cit. on p. 7).
- [8] Ann Finkbeiner. *Looking for Neutrinos, Nature’s Ghost Particles*. November 2010 (accessed 20/01 2023). URL: <https://www.smithsonianmag.com/science-nature/looking-for-neutrinos-natures-ghost-particles-64200742/> (cit. on p. 8).
- [9] S.M. Bilenky. “Neutrino. History of a unique particle”. In: *The European Physical Journal H* 38.3 (2012), pp. 345–404. DOI: [10.1140/epjh/e2012-20068-9](https://doi.org/10.1140/epjh/e2012-20068-9). URL: <https://doi.org/10.1140/epjh/e2012-20068-9> (cit. on p. 9).
- [10] C. L. Cowan et al. “Detection of the Free Neutrino: a Confirmation”. In: *Science* 124.3212 (1956), pp. 103–104. DOI: [10.1126/science.124.3212.103](https://doi.org/10.1126/science.124.3212.103). eprint: <https://www.science.org/doi/pdf/10.1126/science.124.3212.103>. URL: <https://www.science.org/doi/abs/10.1126/science.124.3212.103> (cit. on p. 9).
- [11] M. Sajjad Athar, A. Fatima, and S. K. Singh. *Neutrinos and their interactions with matter*. 2022. DOI: [10.48550/ARXIV.2206.13792](https://doi.org/10.48550/ARXIV.2206.13792). URL: <https://arxiv.org/abs/2206.13792> (cit. on p. 9).
- [12] Etienne Bourbeau. “Measurement of Tau Neutrino Appearance in 8 Years of IceCube Data”. PhD thesis. The Niels Bohr Institute, Faculty of Science, University of Copenhagen, 2021 (cit. on pp. 10, 16, 24, 28, 29).
- [13] Christine Sutton. *CP violation*. URL: <https://www.britannica.com/science/CP-violation> (cit. on p. 10).

- [14] C. Giganti, S. Lavignac, and M. Zito. “Neutrino oscillations: The rise of the PMNS paradigm”. In: *Progress in Particle and Nuclear Physics* 98 (2018), pp. 1–54. DOI: [10.1016/j.pnnp.2017.10.001](https://doi.org/10.1016/j.pnnp.2017.10.001). URL: <https://doi.org/10.1016%2Fj.pnnp.2017.10.001> (cit. on pp. 10, 11).
- [15] Ivan Esteban et al. “The fate of hints: updated global analysis of three-flavor neutrino oscillations”. In: *Journal of High Energy Physics* 2020.9 (2020). DOI: [10.1007/jhep09\(2020\)178](https://doi.org/10.1007/jhep09(2020)178). URL: <https://doi.org/10.1007%2Fjhep09%282020%29178> (cit. on p. 12).
- [16] Čerenkov P. A. “Visible emission of clean liquids by action of radiation”. In: *Dokl. Akad. Nauk SSSR* 2 (1934), 451–454 (cit. on p. 12).
- [17] Boris M Bolotovskii. “Vavilov – Cherenkov radiation: its discovery and application”. In: *Physics-Uspexhi* 52.11 (2009), p. 1099. DOI: [10.3367/UFNe.0179.200911c.1161](https://dx.doi.org/10.3367/UFNe.0179.200911c.1161). URL: <https://dx.doi.org/10.3367/UFNe.0179.200911c.1161> (cit. on pp. 12, 13).
- [18] Kiran Munawar. “Identifying Cosmic Ray induced Cascade events with IceTop”. MA thesis. University of Canterbury, Department of Physics and Astronomy, 2017 (cit. on pp. 13, 14, 22).
- [19] Markus Ahlers, Klaus Helbing, and Carlos Pérez de los Heros. “Probing particle physics with IceCube”. In: *The European Physical Journal C* 78.11 (2018). DOI: [10.1140/epjc/s10052-018-6369-9](https://doi.org/10.1140/epjc/s10052-018-6369-9). URL: <https://doi.org/10.1140%2Fepjc%2Fs10052-018-6369-9> (cit. on pp. 15, 20).
- [20] M. G. Aartsen et al. “Measurement of Atmospheric Neutrino Oscillations at 6–56 GeV with IceCube DeepCore”. In: *Physical Review Letters* 120.7 (2018). DOI: [10.1103/physrevlett.120.071801](https://doi.org/10.1103/physrevlett.120.071801). URL: <https://doi.org/10.1103%2Fphysrevlett.120.071801> (cit. on p. 15).
- [21] Kayla Leonard DeHolton. “Atmospheric Neutrino Oscillations with 8 years of data from IceCube DeepCore”. In: *PoS NuFact2021* (2022), p. 062. DOI: [10.22323/1.402.0062](https://doi.org/10.22323/1.402.0062) (cit. on p. 15).
- [22] J. Kiryluk. *Neutrino Physics with the IceCube Detector*. 2008. DOI: [10.48550/ARXIV.0806.1717](https://arxiv.org/abs/0806.1717). URL: <https://arxiv.org/abs/0806.1717> (cit. on pp. 15, 19).
- [23] R. Abbasi et al. “Evidence for neutrino emission from the nearby active galaxy NGC 1068”. In: *Science* 378.6619 (2022), pp. 538–543. DOI: [10.1126/science.abg3395](https://doi.org/10.1126/science.abg3395). URL: <https://doi.org/10.1126%2Fscience.abg3395> (cit. on pp. 15, 18).
- [24] Francis Halzen and Spencer R. Klein. “Invited Review Article: IceCube: An instrument for neutrino astronomy”. In: *Review of Scientific Instruments* 81.8 (2010), p. 081101. DOI: [10.1063/1.3480478](https://doi.org/10.1063/1.3480478). URL: <https://doi.org/10.1063%2F1.3480478> (cit. on pp. 15–17, 20).
- [25] R. Abbasi et al. “Limits on a muon flux from Kaluza-Klein dark matter annihilations in the Sun from the IceCube 22-string detector”. In: *Physical Review D* 81.5 (2010). DOI: [10.1103/physrevd.81.057101](https://doi.org/10.1103/physrevd.81.057101). URL: <https://doi.org/10.1103%2Fphysrevd.81.057101> (cit. on pp. 15, 19).
- [26] Sebastian Baur. *Dark matter searches with the IceCube Upgrade*. 2019. DOI: [10.48550/ARXIV.1908.08236](https://arxiv.org/abs/1908.08236). URL: <https://arxiv.org/abs/1908.08236> (cit. on pp. 15, 19).
- [27] Giovanni Renzi. *Search for dark matter from the center of the Earth with 8 years of IceCube data*. 2021. DOI: [10.48550/ARXIV.2107.11244](https://arxiv.org/abs/2107.11244). URL: <https://arxiv.org/abs/2107.11244> (cit. on pp. 15, 19).

- [28] Morten Ankersen Medici. “Search for Dark Matter Annihilation in the Galactic Halo using IceCube”. PhD thesis. The Niels Bohr Institute, Faculty of Science, University of Copenhagen, 2016 (cit. on pp. 15, 19).
- [29] R. Abbasi et al. “Search for Unstable Sterile Neutrinos with the IceCube Neutrino Observatory”. In: *Physical Review Letters* 129.15 (2022). DOI: 10.1103/physrevlett.129.151801. URL: <https://doi.org/10.1103%2Fphysrevlett.129.151801> (cit. on pp. 15, 19).
- [30] M.G. Aartsen et al. “The IceCube realtime alert system”. In: *Astroparticle Physics* 92 (2017), pp. 30–41. DOI: 10.1016/j.astropartphys.2017.05.002. URL: <https://doi.org/10.1016%2Fj.astropartphys.2017.05.002> (cit. on pp. 16, 17).
- [31] R. Abbasi et al. “IceCube sensitivity for low-energy neutrinos from nearby supernovae”. In: *Astronomy & Astrophysics* 535 (2011), A109. DOI: 10.1051/0004-6361/201117810. URL: <https://doi.org/10.1051%2F0004-6361%2F201117810> (cit. on p. 17).
- [32] Christian Spiering. “Towards high-energy neutrino astronomy”. In: *The European Physical Journal H* 37.3 (2012), pp. 515–565. DOI: 10.1140/epjh/e2012-30014-2. URL: <https://doi.org/10.1140%2Fepjh%2Fe2012-30014-2> (cit. on p. 17).
- [33] Mark Aartsen et al. “Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A”. In: *Science* 361.6398 (2018). DOI: 10.1126/science.aat1378. URL: <https://doi.org/10.1126%2Fscience.aat1378> (cit. on p. 18).
- [34] Mark Aartsen et al. “Neutrino emission from the direction of the blazar TXS 0506056 prior to the IceCube-170922A alert”. In: *Science* 361.6398 (2018), pp. 147–151. DOI: 10.1126/science.aat2890. URL: <https://doi.org/10.1126%2Fscience.aat2890> (cit. on p. 18).
- [35] M. G. Aartsen et al. “eV-Scale Sterile Neutrino Search Using Eight Years of Atmospheric Muon Neutrino Data from the IceCube Neutrino Observatory”. In: *Physical Review Letters* 125.14 (2020). DOI: 10.1103/physrevlett.125.141801. URL: <https://doi.org/10.1103%2Fphysrevlett.125.141801> (cit. on p. 19).
- [36] R. Abbasi et al. “The design and performance of IceCube DeepCore”. In: *Astroparticle Physics* 35.10 (2012), pp. 615–624. DOI: 10.1016/j.astropartphys.2012.01.004. URL: <https://doi.org/10.1016%2Fj.astropartphys.2012.01.004> (cit. on pp. 20, 21, 23).
- [37] R. Abbasi et al. “IceTop: The surface component of IceCube”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 700 (2013), pp. 188–220. DOI: 10.1016/j.nima.2012.10.067. URL: <https://doi.org/10.1016%2Fj.nima.2012.10.067> (cit. on pp. 21, 22).
- [38] R. Abbasi et al. “The IceCube data acquisition system: Signal capture, digitization, and timestamping”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 601.3 (2009), pp. 294–316. DOI: 10.1016/j.nima.2009.01.001. URL: <https://doi.org/10.1016%2Fj.nima.2009.01.001> (cit. on p. 23).
- [39] Aya Ishihara. *The IceCube Upgrade – Design and Science Goals*. 2019. DOI: 10.48550/ARXIV.1908.09441. URL: <https://arxiv.org/abs/1908.09441> (cit. on p. 23).

- [40] IceCube Collaboration. *NSF approves funding for IceCube Upgrade*. 2019. URL: <https://icecube.wisc.edu/gallery/nsf-approves-funding-for-icecube-upgrade/> (cit. on p. 23).
- [41] Ryo Nagai and Aya Ishihara. *Electronics Development for the New Photo-Detectors (PDOM and D-Egg) for IceCube-Upgrade*. 2019. DOI: 10.48550/ARXIV.1908.11564. URL: <https://arxiv.org/abs/1908.11564> (cit. on p. 23).
- [42] Aya Ishihara and Ayumi Kiriki. *Calibration LEDs in the IceCube Upgrade D-Egg Modules*. 2019. DOI: 10.48550/ARXIV.1908.10780. URL: <https://arxiv.org/abs/1908.10780> (cit. on p. 23).
- [43] Lew Classen, Alexander Kappes, and Timo Karg. *A multi-PMT Optical Module for the IceCube Upgrade*. 2019. DOI: 10.48550/ARXIV.1908.10802. URL: <https://arxiv.org/abs/1908.10802> (cit. on p. 23).
- [44] Vedant Basu et al. "A next-generation optical sensor for IceCube-Gen2". In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)*. Sissa Medialab, 2021. DOI: 10.22323/1.395.1062. URL: <https://doi.org/10.22323%2F1.395.1062> (cit. on p. 23).
- [45] Summer Blot et al. *oscNext - Simulations and sample (v00.07)*. 2022 (cit. on pp. 24, 28, 115, 116).
- [46] C. Andreopoulos et al. "The GENIE neutrino Monte Carlo generator". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 614.1 (2010), pp. 87–104. DOI: 10.1016/j.nima.2009.12.009. URL: <https://doi.org/10.1016%2Fj.nima.2009.12.009> (cit. on p. 25).
- [47] Kayla Leonard and Michael Larson. *OscNext MuonGun Optimization*. Last edited 2018 (accessed 17/01 2023). URL: https://wiki.icecube.wisc.edu/index.php/OscNext_MuonGun_Optimization (cit. on pp. 25, 26).
- [48] D. Heck et al. "CORSIKA: A Monte Carlo code to simulate extensive air showers". In: (Feb. 1998) (cit. on p. 25).
- [49] Michael James Larson. "Simulation and identification of non-Poissonian noise triggers in the IceCube neutrino detector". MA thesis. Alabama U, 2013 (cit. on p. 26).
- [50] Juan Pablo Yáñez Garza. "Measurement of neutrino oscillations in atmospheric neutrinos with the IceCube DeepCore detector". PhD thesis. Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2014. DOI: <http://dx.doi.org/10.18452/17016> (cit. on p. 27).
- [51] M. G. Aartsen et al. "Atmospheric and astrophysical neutrinos above 1 TeV interacting in IceCube". In: *Physical Review D* 91.2 (2015). DOI: 10.1103/physrevd.91.022001. URL: <https://doi.org/10.1103%2Fphysrevd.91.022001> (cit. on p. 27).
- [52] Dmitry Chirkin and Wolfgang Rhode. *Propagating leptons through matter with Muon Monte Carlo (MMC)*. 2004. DOI: 10.48550/ARXIV.HEP-PH/0407075. URL: <https://arxiv.org/abs/hep-ph/0407075> (cit. on p. 27).
- [53] M. G. Aartsen et al. "Search for Sources of Astrophysical Neutrinos Using Seven Years of IceCube Cascade Events". In: *The Astrophysical Journal* 886.1 (2019), p. 12. DOI: 10.3847/1538-4357/ab4ae2. URL: <https://dx.doi.org/10.3847/1538-4357/ab4ae2> (cit. on p. 27).
- [54] Marek Kowalski and the IceCube Collaboration. "Neutrino astronomy with IceCube and beyond". In: *Journal of Physics: Conference Series* 888.1 (2017), p. 012007. DOI: 10.1088/1742-6596/888/1/012007. URL: <https://dx.doi.org/10.1088/1742-6596/888/1/012007> (cit. on pp. 27, 28).
- [55] Etienne Bourbeau et al. *oscNext High Stats Sample (v1.1)*. 2022 (cit. on p. 30).

- [56] Kai Schatto. “Stacked searches for high-energy neutrinos from blazars with IceCube”. PhD thesis. Mainz U., June 2014 (cit. on p. 30).
- [57] Julianna Delua. *Supervised vs. Unsupervised Learning: What’s the Difference?* March 2021 (accessed 28/02 2023). URL: <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning> (cit. on p. 31).
- [58] Yann Coadou. “Boosted Decision Trees”. In: *Artificial Intelligence for High Energy Physics*. WORLD SCIENTIFIC, 2022, pp. 9–58. DOI: 10.1142/9789811234033_0002. URL: https://doi.org/10.1142/9789811234033_0002 (cit. on p. 32).
- [59] *What Is a Decision Tree?* (accessed 01/03 2023). URL: <https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/> (cit. on p. 32).
- [60] Guolin Ke et al. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf> (cit. on p. 32).
- [61] Larry Hardesty. *Explained: Neural networks Ballyhooed artificial-intelligence technique known as “deep learning” revives 70-year-old idea*. April 2017 (accessed 27/02 2023). URL: <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414> (cit. on p. 33).
- [62] Victor Zhou. *Machine Learning for Beginners: An Introduction to Neural Networks*. March 2019 (accessed 01/03 2023). URL: <https://towardsdatascience.com/machine-learning-for-beginners-an-introduction-to-neural-networks-d49f22d238f9> (cit. on p. 33).
- [63] *What are neural networks?* (accessed 02/03 2023). URL: <https://www.ibm.com/topics/neural-networks> (cit. on p. 34).
- [64] Franco Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605 (cit. on p. 34).
- [65] Pankaj Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics Reports* 810 (2019), pp. 1–124. DOI: 10.1016/j.physrep.2019.03.001. URL: <https://doi.org/10.1016/j.physrep.2019.03.001> (cit. on pp. 35–37).
- [66] Jeremy Jordan. *Setting the learning rate of your neural network*. Mar 2028 (accessed 06/03 2023). URL: <https://www.jeremyjordan.me/nn-learning-rate/> (cit. on p. 36).
- [67] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: <https://arxiv.org/abs/1412.6980> (cit. on p. 36).
- [68] Ryan Holbrook and Alexis Cook. *Overfitting and Underfitting*. (accessed 02/03 2023). URL: <https://www.kaggle.com/code/ryanholbrook/overfitting-and-underfitting> (cit. on p. 38).
- [69] Andreas Søgaard et al. *GraphNeT: Graph neural networks for neutrino telescope event reconstruction*. 2022. DOI: 10.48550/ARXIV.2210.12194. URL: <https://arxiv.org/abs/2210.12194> (cit. on pp. 38, 39).
- [70] Andreas Søgaard et al. *GraphNeT*. Version v0.2.3. If you use this software, please cite it as below. Dec. 2022. DOI: 10.5281/zenodo.7414920. URL: <https://doi.org/10.5281/zenodo.7414920> (cit. on p. 38).
- [71] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. 2018. DOI: 10.48550/ARXIV.1801.07829. URL: <https://arxiv.org/abs/1801.07829> (cit. on pp. 39, 40).

- [72] Hongbin Pei et al. *Geom-GCN: Geometric Graph Convolutional Networks*. 2020. DOI: [10.48550/ARXIV.2002.05287](https://doi.org/10.48550/ARXIV.2002.05287). URL: <https://arxiv.org/abs/2002.05287> (cit. on pp. 39, 40).
- [73] Sklearn. *Stochastic Gradient Descent*. URL: <https://scikit-learn.sourceforge.net/0.6/modules/sgd.html> (cit. on pp. 40, 49).
- [74] Sachin Kumar and Yulia Tsvetkov. *Von Mises-Fisher Loss for Training Sequence to Sequence Models with Continuous Outputs*. 2018. DOI: [10.48550/ARXIV.1812.04616](https://doi.org/10.48550/ARXIV.1812.04616). URL: <https://arxiv.org/abs/1812.04616> (cit. on p. 41).
- [75] Shruti Jadon. "A survey of loss functions for semantic segmentation". In: *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, 2020. DOI: [10.1109/cibcb48159.2020.9277638](https://doi.org/10.1109/cibcb48159.2020.9277638). URL: <https://doi.org/10.1109%2Fcibcb48159.2020.9277638> (cit. on p. 41).
- [76] Sklearn. *sklearn.preprocessing.QuantileTransformer*. (accessed 16/03 2023). URL: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html> (cit. on p. 50).
- [77] Sarang Narkhede. *Understanding AUC - ROC Curve*. URL: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5> (cit. on p. 59).