UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

Master thesis

Leon Bozianu

Stopped Muon Reconstruction In IceCube DeepCore

Using Background as Signal, Making Simulation Resemble Data

Advisor: Troels Petersen

Handed in: May 20, 2022

Abstract

Low energy oscillation analyses at the IceCube Neutrino Observatory contend with a substantial atmospheric muon flux background as well as large systematics, including optical module efficiency and several ice properties. This work proposes the first step into using the muon background to mitigate some of the large systematics affecting neutrino oscillation analyses. By reconstructing a large population of stopped muons in the DeepCore sub-array we propose evaluating and contrasting different systematic configurations of the detector itself. Muons represent a high quality test beam for the calibration of detector performance because the variety of possible measurements and the high statistics permit detailed consistency checks.

We use novel machine learning methods, in particular graph neural networks (GNNs), in order to accurately and rapidly reconstruct stopped muon candidate events using their collected Cherenkov light in both simulation and experimental data. The remarkable speed of the GNN ($> 10,000$ events s^{-1}) permits the reconstruction of millions of muon events on raw IceCube (L2) data, with only a single filter applied, for the first time. This work consciously prepares for the future application of GNNs to a larger sample of raw IceCube events, with potential for widespread low-level reconstruction.

Finally, the accurate reconstruction of many real muon events allows for the comparison of data to current IceCube simulation. Here we scrutinize muon dominated data taken from a single day of detector livetime and compare to low-energy muon simulation (MuonGun).

Acknowledgements

My year working in IceCube at NBI owes so much to the people who make up the wonderful group there. To my supervisor Troels go my sincerest thanks, gratitude and appreciation for completely epitomizing the role of inspiring mentor. I'm extremely grateful to Rasmus for the excellent environment he created, this thesis would not have been possible without you. I will be forever indebted to Tom for his infinite patience and willingness to help, you went beyond what was necessary, and then even further. Special thanks to the head chef of IceCube, Jason, for your counsel, company and sense of humor. Thanks also go to Andreas for his forgiving reviews of my code and accommodating outlook. To Tania thank you for the assistance but more so for the friendship. I would be remiss in not mentioning those who made the corridor the place it was: Aske, Martin, Kevin, Morten, James and Marc!

The question of how to thank you for the last year Kaare, is one no office mate should have to endure, I can only hope that I have made this project of ours even half as enjoyable for you as you did for me. It really was the best of times.

To my parents, who are and have always been there. Finally, to Fran, for making everyday beautiful, even those spent late in the office.

Contents

1

Introduction Particle Physics

1.1 Motivation

Neutrinos are among the most elusive particles known to exist today, accordingly, the main focus of this work will be muons.

The IceCube South Pole Neutrino Observatory is one part of the considerable efforts here on Earth to isolate and analyse neutrinos originating from astrophysical sources. The challenge of neutrino detection is also their promise, since neutrinos interact so infrequently they offer us a unique opportunity to observe those that have travelled directly from whichever particularly violent astrophysical events produced them. The IceCube observatory was primarily designed to observe astrophysical neutrinos in the Antarctic ice at the South pole, however atmospheric neutrinos constitute an irreducible background for astrophysical neutrino searches. It is fortunate, then, that atmospheric neutrinos provide the main signal for particle physics studies using modern neutrino telescopes. In particular, neutrino oscillation and mass hierarchy (the two most prominent indications of physics beyond the standard model) can both be probed with the same instruments that were originally designed for multimessenger astronomy.

The goals of this project are two-fold, first, to use novel machine learning techniques, specifically graphical neural networks to process and reconstruct IceCube physics events from the raw detector response. Secondly, to compare and contrast real experimentally collected data and current low energy simulation using a sample of stopped muons that has previously been too expensive to accurately reconstruct.

The natural place to start, and where this chapter continues is with an introduction to the relevant particle physics concepts. The next chapter concerns the IceCube Neutrino Observatory itself, which is immediately followed by the tangentially related chapter on machine learning. Here we discuss IceCube data for machine learning and introduce the model used in this analysis.

The penultimate chapter details the recovered results and their potential implications. Finally, a summary is given before a concluding look to the future and potential next steps.

1.2 The Standard Model of Particle Physics

Widely considered the crowning achievement of particle physics in the 20th century the Standard Model (SM) is in some ways the benchmark/measure against which modern particle physics research must compare itself.

The now infamous infrastructure contains 17 particles, shown in Figure 1.1, that are divided into two separate groups - the fermions and the bosons. The vector (or gauge) bosons describe three of the four known forces, electromagnetic interactions through the photon, weak interactions via the $W^{\pm}$ and Z bosons and the strong force by eight types of gluon.

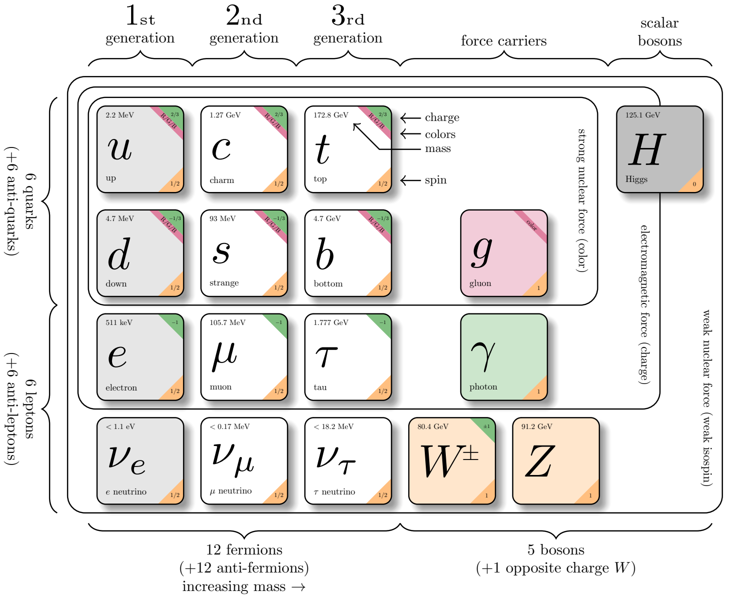

1st generation
2nd generation
3rd generation
force carriers
scalar bosons
6 quarks (+6 anti-quarks)
6 leptons (+6 anti-leptons)
2.2 MeV 2/3 R/G/B u up 1/2
1.27 GeV 2/3 R/G/B c charm 1/2
172.8 GeV 2/3 R/G/B t top 1/2
charge
colors
mass
spin
125.1 GeV H Higgs 0
4.7 MeV −1/3 R/G/B d down 1/2
93 MeV −1/3 R/G/B s strange 1/2
4.7 GeV −1/3 R/G/B b bottom 1/2
color g gluon 1
strong nuclear force (color)
511 keV −1 e electron 1/2
105.7 MeV −1 μ muon 1/2
1.777 GeV −1 τ tau 1/2
γ photon 1
electromagnetic force (charge)
< 1.1 eV ν_e e neutrino 1/2
< 0.17 MeV ν_μ μ neutrino 1/2
< 18.2 MeV ν_τ τ neutrino 1/2
80.4 GeV ±1 W± 1
91.2 GeV Z 1
weak nuclear force (weak isospin)
12 fermions (+12 anti-fermions) increasing mass →
5 bosons (+1 opposite charge W)

Figure 1.1: The common pictorial representation of the particle content of the Standard Model adapted from [1] with fermion masses from [2].

1.2.1 The Vector/Gauge Bosons

The vector bosons arise from the insistence that the underlying interaction, and thus Lagrangian, must be invariant under certain *local* transformations often called gauge transformations. Heuristically this means that the elements of the transformation group transform the states and operators of a theory, in such a way that the Lagrangian does not change [3].

The consequences of this demand are profoundly important. The notion of invariance under such a transformation implies that physical fields can be transformed by arbitrary amounts at each space-time point, while the laws of physics remain the same. In addition, if a theory is invariant under some transformation then one obtains a conservation law for some quantity from Noether's theorem [4].?

For each generator of a symmetry group transformation that permits Lagrangian invariance under such a transformation there is an associated gauge boson that acts as the force mediator.

The Electromagnetic Force A local $U(1)$ symmetry group has just one generator which in turn gives rise to one field whose quantum is the photon, γ. The $U(1)$ gauge symmetry comes from the fact that one cannot measure the absolute phase of the wavefunction of a particle. $U(1)$ refers to the group of rotations around a fixed axis, also called the circle group. Hence, invariance under $U(1)$ transformations is the same as insisting that the calculation of a measurable quantity must not change when the phase is changed or rotated. Conceptually, the other force mediators described in the SM are derived in the same way, with slightly more complex symmetry group transformations. The protocol is simply to enforce local gauge invariance, compensate for this with gauge field with specific gauge transformation include this gauge field in the Lagrangian introducing force carrier interactions. Later, in Section 1.6, we will see the role the electromagnetic force plays in the indirect detection of neutrinos.

The Weak Force Weak interactions may be described by the $SU(2)$ gauge group where there exist three generators. The three generators correspond to the W^+, W^- and Z bosons. Since these bosons have a non-zero mass, they may only act over very short ranges and are relatively short-lived. For weak interactions the quantities weak isospin and weak hypercharge are analogous to electric charge in electromagnetic interactions. Particles with weak isospin (such as the left-handed chiral fermions) may interact through the exchange of $W^\pm$ while those with weak hypercharge (all fermions except for right-chiral neutrinos) may interact through exchange of Z bosons.In such interactions weak isospin or weak hypercharge is conserved.

Weak interactions mediated by $W^\pm$ are unique in so far as they allow the participants to change flavour but also that charge-parity symmetry is violated. Charge-parity symmetry implies that the laws of physics in a system should remain the same if a particle is replaced by its anti-particle (charge symmetry) while the spatial coordinates are simultaneously reversed, or inverted through the origin, (parity symmetry). The weak interaction is referred to as chiral because there is a difference in the treatment of so-called left and right chiral particle states in interactions with $W^\pm$ and Z. The chirality of a particle is a fundamental property of a particle, whereas helicity is defined as the projection of the particle's spin vector onto the vector of its momentum. For massless particles the chirality and helicity are the same, but for particles whose mass is non-zero the helicity becomes frame-dependent and therefore may not be the same as chirality. This has the effect of the $W^{\pm}$ boson only coupling to left-handed fermions and right-handed anti-fermions (only particles with non-zero weak isospin). The Z boson, on the other hand, will couple to both kinds of chiral fermions but with different strengths (due to their respective weak isospin values). Weak interactions are particularly important in the operation of IceCube as the W and Z bosons are the only bosons that can mediate interactions between neutrinos and other fermions (the only method by which the neutrinos can interact). Neutrino interactions will be revisited again in Section 1.3.

The Strong Force The conspicuously named strong force is mediated by 8 massless bosons called gluons, g in Figure 1.1. These arise as the generators of an $SU(3)$ symmetry group and conserve a particle quantity called color that is only carried by all of the quarks and the gluons themselves. For this reason, the gluons may self-interact in theoretically proposed structures such as glueballs. The strong force coupling constant is 137 times stronger than EM and six orders of magnitude stronger than weak interactions at short ranges (approximately the radius of a nucleon) [5]. On an even smaller scale, the strong force binds quarks together to form so-called colorless combinations called hadrons. The structure of the hadrons (of which the nucleons are a subset) has a defining role in weak neutrino-nucleon interactions, in particular the partonic distribution functions of the quarks and gluons within.

A quark-antiquark combination, called a meson, is overall color neutral as the antiquark's color charge must necessarily be the anti-color of the quark itself. Mesons are an integral part of the chain of particle interactions in the development of air showers, which will be discussed further in Section 1.5.

The strong force displays a unique property called *confinement* that prevents free particles with non-zero color charge from existing. If a color-neutral pair of quarks are separated from each other, they instantaneously drag quark/antiquark (and hence color neutral) partners from the vacuum thus forming two new quark antiquark pairs. As their separation increases it becomes more energetically favourable for a new quark antiquark pair to instantaneously appear than for the quarks to separate further.

1.2.2 The Scalar Boson

The last part of the SM to gain experimental verification was the only scalar boson, the **Higgs Boson**, that has zero spin and is a quanta of the Higgs field. The massive particles in the SM all acquire their masses through interactions with the Higgs field (except for the neutrinos) where the strength of coupling is proportional to the particle's mass [6, 7].

1.2.3 The Fermions

The second half of the SM pertains to a group of particles called the fermions, a group which contains the constituents of matter in the universe. Each of these particles has half integer spin and all obey the Pauli exclusion principle [8], that two identical fermions may not occupy the same quantum state in the same system simultaneously. The collection of several fermions may form composite particles that themselves make up everyday matter. The fermions are split into two categories, of which both contain three generations, shown in the first three columns in Figure 1.1. The mass spectra of the fermions is shown in Figure 1.3 and are grouped by their generation (or family in the figure).

Those in the first category, **quarks**, make up the composite particles called hadrons and, as previously stated, are the only massive particles with color charge and thus the only fermionic participants in strong interactions. There are 6 quarks in the SM that are arranged in three generational pairs. Associated with each quark is a corresponding anti-quark with identical mass, but reversed electric charge sign. The antiquarks are arranged in the same generational pairs as the quarks themselves. In the first generation are the "up" and "down" quarks, in the second the "charm" and "strange" quark and in the third the "top" and "bottom" quarks. The up-type quarks (up, charm and top) have a positive electric charge $+\frac{2}{3}$ and masses of 2.16 ${}^{+0.49}_{-0.26}$ MeV, 1.27 ± 0.02 GeV and 172.76 ± 0.30 GeV respectively. The down-type quarks (down, strange, bottom) have negative electric charge $-\frac{1}{3}$ and have masses of 4.67 ${}^{+0.48}_{-0.17}$ MeV, 93 ${}^{+11}_{-5}$ MeV and 4.18 ${}^{+0.03}_{-0.02}$ GeV respectively [2].

As already stated the quarks are color-confined by gluons mediating the strong force allowing hadrons and other composite particles to form. In addition to the strong force the quarks are also subject to the weak force and it is this fundamental force that causes the decay of the more massive quarks. Since $W^{\pm}$ weak interactions allow fermions to change flavour, the quarks may change type after participation though this is restrained in part by the CKM matrix [9] that suppresses quark flavour transitions between the different generations. A famous example of such a flavour changing $W^{\pm}$ weak interaction is beta decay, see Figure 1.2. In this process, a neutron composed of a udd quarks decays into a charged W boson and a proton (uud quarks) before the W boson subsequently decays to a pair of leptons. The properties of the emitted lepton pair depend on the charge of the W mediator, W^{-} leads to an e^{-} electron and $\overline{\nu_e}$ anti-electron neutrino in so called β^{-} decay. For W^{+} decays the products are e^{+} and ν_e called β^{+} decay.

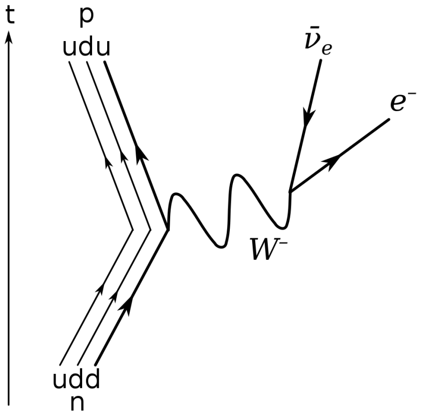


Figure 1.2: Feynman diagram of beta decay (β^-) oriented with time running vertically. Image taken from [10]

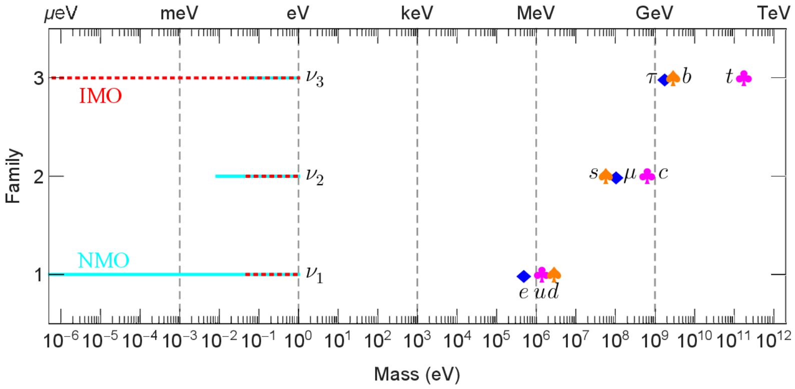


Figure 1.3: Schematic representation of the fermion mass spectra. The neutrino masses are shown as a range where the cyan lines refer to the normal mass ordering (NMO) and the dashed red lines the inverted mass ordering (IMO) cases.

The second category of fermions are the **leptons**, which include three types (or flavours) of indivisible electron-like particles all with an electric charge of -1 and three light neutral particles of corresponding flavours called neutrinos. Like the quarks, the leptons are arranged in pairs of three generations where, as before, the first generation consists of the lightest and most stable particles and subsequent generations contain more massive and less stable particles. The pairs of electronic, muonic and tauonic leptons consist of the electron and electron neutrino, muon and muon neutrino and tau and tau neutrino respectively. The charged leptons are represented in Figure 1.1 as e, μ and τ and the neutrinos by ν_e, ν_μ and ν_τ.

The lightest and most notorious charged lepton, the electron, has a mass of $0.511 \pm 3.1 \times 10^{-9}$ MeV and is the only stable charged lepton. The masses of the muon and tau leptons are orders of magnitude larger than the electron at specifically $105 \pm 2.4 \times 10^{-6}$ MeV for the muon and 1777 ± 0.12 MeV for the tau [2], which can be seen in Figure 1.3. The muon or the tau may decay via the charged W boson into a mixture of an electron and neutrinos that conserves both charge and lepton number. Two representative Feynman diagrams are shown in Figure 1.4.

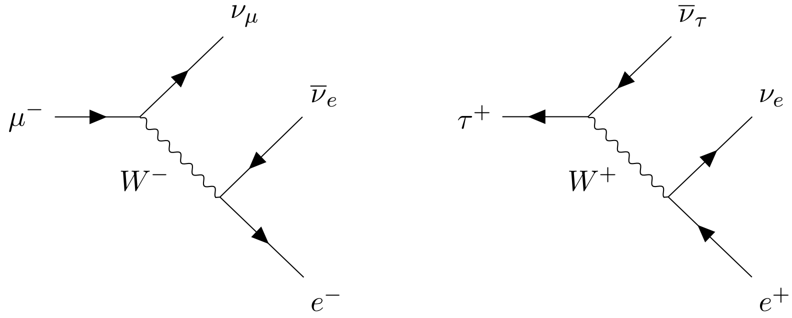


Figure 1.4: Simple example decay modes for muon and anti-tau leptons. One of the decay products must be the same flavour as the initial lepton, hence a muon neutrino is always produced in a muon decay. In order to conserve charge another of the products must have the same charge as the initial.

The mean lifetime of the muon is 2.2×10^{-6} s which is far longer than the tau lepton which decays after just 2.9×10^{-13} s [2]. The muon decay in 1.4 is the dominant decay process and sometimes referred to as a Michel decay $\mu^- - \rightarrow \nu_\mu e^- \overline{\nu}_e$. The tau on the other hand has more potential decay modes as the larger tau mass makes other modes accessible including a muon decay mode $\tau^- \rightarrow \nu_\tau e^- \overline{\nu}_e$ which ignoring the muon mass relative to the tau mass will occur at the same rate as the electron decay mode. Additionally the tau, being more massive than the $\pi^\pm$ (pion) mesons, may also decay to quarks as $\tau^- \rightarrow \nu_\tau d\overline{u}$.

In propagation, the large mass of the tau means that it does not emit as much bremsstrahlung radiation as en electron and is therefore far more penetrating than the electron. Bremsstrahlung is the name of electromagnetic radiation produced when a charged particle is decelerated in the field of another charged particle and ensures the conservation of energy. The emission of bremsstrahlung radiation by charged leptons, particularly those at high energies [11, 12], increases the speed at which they lose energy and prevent further penetration of matter. Unfortunately, the fact that the tau is incredibly short lived prevents it from travelling far before it decays so the reduction in bremsstrahlung losses is insignificant. Tau penetration only becomes noticeable at exceedingly high energies where time dilation extends its lifetime.

Muons, on the other hand, exist in a prosperous middle ground between the other charged leptons. The muons mass means that less bremsstrahlung is emitted than electrons while its lifetime is longer than that of taus. As a result, the muon is highly penetrating and can pass through matter for far longer and with less significant deflection before decaying or interacting than the other charged leptons. These circumstances will be very important later, where we will discuss the propagation of muons further, see Section 1.5.1.

Finally, we turn to the neutral leptons, called **neutrinos**, that play a defining role in this work and the objectives of the IceCube detector. In fact, in IceCube the charged leptons can be thought of as the visible remnants/signatures of the ghostly neutrinos. In the SM, the three neutrinos are impossible to detect directly, as they only interact via weak interactions, an example as in Figure 1.4, and gravity that is not included in the SM.

Additionally, the meaning of the different neutrino flavours is simply defined by the flavour of charged lepton that induces the interaction (or is the interactions' result). For example, $\overline{\nu}_e$ is the neutrino state that is produced along with an electron, as seen in 1.2. For interactions with the Z boson it is inconsequential to assign a flavour to the neutrino involved as there is no way to experimentally verify the flavour state without the neutrino interacting again.

As previously established the neutrinos only directly detectable mechanism of interaction is through the parity violating weak force, thus only left-handed neutrino and right-handed anti-neutrino chiral states have been "observed". In fact, the SM anticipates only the existence of left-handed neutrinos (and right-handed anti-neutrinos) while predicting that all neutrinos are massless. This puts the SM in tension with implications of experimental evidence that neutrinos oscillate and therefore must have a non-zero mass. In the following section we will investigate neutrino oscillation phenomena further, before going into more detail on their weak interactions.

1.3 Neutrino Oscillations and Interactions

A property of neutrinos is that they appear to oscillate between their three (known) flavours. This phenomena dictates that the probability a neutrino is detected in one flavour state after having been emitted in another varies with some periodic frequency. This oscillation probability depends on several factors including the initial flavour state occupied, the distance of propagation and its energy. The discovery of neutrino oscillations took place in a variety of neutrino experiments with differing philosophies across the globe including, among others [13], the atmospheric Super Kamiokande experiment [14], the solar SNO experiment [15] and the reactor experiment KamLAND [16]. The discovery of neutrino oscillations was, at the time, very surprising since it necessitated that neutrinos have a non-zero mass, something that the SM had not accounted for [17].

We are unable to directly observe neutrinos and as such may only detect the products of their weak interactions. Thus, it is by definition that we define ν_l as the neutrino state produced alongside the (anti-)lepton l. Or equivalently, that CC interactions with an incident ν_l will produce a (anti-)lepton of flavour l. This is an important feature of the neutrinos, that the state in which it *interacts* is distinct from the state in which it *propagates*. The three flavour states (ν_e, ν_μ and ν_τ) can be represented as a superposition of three mass eigenstates (ν_1,ν_2 and ν_3) and are related by the unitary lepton mixing matrix, more widely called the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) [18] matrix:

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} U_{e1} & U_{e2} & U_{e3} \\ U_{\mu 1} & U_{\mu 2} & U_{\mu 3} \\ U_{\tau 1} & U_{\tau 2} & U_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} \tag{1.1}$$

Or equivalently, as coherent linear combinations of the mass eigenstates,

$$|\nu_\alpha\rangle = \sum_i U_{\alpha i} |\nu_i\rangle \,, \tag{1.2}$$

where $\alpha = e, \mu, \tau$ and $i = 1, 2, 3$ are the flavour and mass eigenstates respectively.

Intuitively, each element of the PMNS matrix gives the mixing of each flavour state with each mass state. The fraction of the flavour state ν_μ present in ν_1 is defined by the matrix element $|U_{\mu 1}|^2$. Clearly, if there were no mixing of the flavour and mass eigenstates then the PMNS matrix would simply be the identity matrix. The absolute difference between the two mass states ν_1 and ν_2 is very small as compared to the absolute difference between ν_1 or ν_2 and ν_3. As seen in Figure 1.3 the normal mass ordering places the mass eigenstates in the following hierarchy: $\nu_1, \nu_2 \ll \nu_3$ while the inverted mass ordering states $\nu_1, \nu_2 \gg \nu_3$.

We may rewrite the PMNS according to a parameterization that is common in the literature [19, 20] in terms of three rotation angles $\theta_{12}, \theta_{23}, \theta_{13}$ and a complex phase δ (assuming that neutrinos are not their own antiparticle, and hence Dirac fermions).

$$
\begin{aligned}
U &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}e^{-i\delta} \\ 0 & 1 & 0 \\ s_{13}e^{-i\delta} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix}
\end{aligned}
\tag{1.3}
$$

Here s_{ij} and c_{ij} represent $\sin\theta_{ij}$ and $\cos\theta_{ij}$ respectively. The δ term in the exponent allows neutrinos to violate CP symmetry, and although this has been experimentally observed, $\delta = 0$ is often taken as the global best. In addition to the three mixing angles and complex phase the neutrino oscillations are also sensitive to the two independent mass squared differences between the mass eigenstates.

For clarity we will always denote the flavour states using Greek indices α or β and the mass states with Latin indices i, j.

Since neutrinos are produced in weak interactions in a pure flavour state we cannot know which of the mass eigenstates were involved in some interaction. Following [19], we can write the initial pure flavour state produced in the interaction at time $t = 0$ as a coherent linear combination of the mass eigenstates.

$$|\nu(t=0)\rangle = |\nu_\alpha\rangle = \sum_i U^*_{\alpha i} |\nu_i\rangle \tag{1.4}$$

Next, we can evolve the state in time allowing the neutrino to propagate. Each mass eigenstate evolves with its own phase factor, as a plane wave. The neutrino is no longer in a pure flavour eigenstate.

$$|\nu(t)\rangle = \sum_i U^*_{\alpha i} e^{-iE_i t} |\nu_i\rangle = \sum_i U^*_{\alpha i} e^{-iE_i t} \sum_\beta U_{\beta i} |\nu_\beta\rangle \tag{1.5}$$

where after the second equality we have expanded the mass eigenstate as a combination of the flavour states while utilising the unitarity of the PMNS matrix. For neutrinos we may safely assume that their behaviour is relativistic (i.e $E_i \gg m_i$) and we may expand the energy using $E \simeq p$ as,

$$E_i = \sqrt{p^2 + m_i^2} \simeq p + \frac{m_i^2}{2E} \tag{1.6}$$

The transition amplitude of a flavour conversion $|\nu_\alpha\rangle \to |\nu_\beta\rangle$ as a function of t is obtained as follows,

$$A_{\nu_\alpha \to \nu_\beta} \equiv A_{\alpha\beta} = \langle \nu_\beta | \nu_\alpha(t) \rangle = \sum_i U^*_{\beta i} U_{\alpha i} e^{-iE_i t} \tag{1.7}$$

We can use the transition amplitude to obtain the probability of oscillation which is to say the probability that a neutrino produced in flavour eigenstate $|\nu_\alpha\rangle$ is at some time/distance later detected in flavour eigenstate $|\nu_\beta\rangle$.

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) = A^*_{\alpha\beta} A_{\alpha\beta} = \delta_{\alpha\beta} - 4\sum_{i<j} \mathrm{Re}[U_{\alpha i} U^*_{\beta i} U^*_{\alpha j} U_{\beta j}] \sin^2\left(\frac{\Delta m^2_{ji} L}{4E}\right) \\ + 2\sum_{i<j} \mathrm{Im}[U_{\alpha i} U^*_{\beta i} U^*_{\alpha j} U_{\beta j}] \sin\left(\frac{\Delta m^2_{ji} L}{2E}\right) \end{aligned} \tag{1.8}$$

Here we have introduced a new shorthand, $\Delta m^2_{ji} \equiv m^2_j - m^2_i$ which we call the mass squared splittings. The quantity $L \simeq ct$ is the distance travelled by the neutrino in a time t in the relativistic limit. This distance is also often referred to as the *baseline*, and is equal to the distance between the source and point of detection of the neutrino. The arguments of sin and cos in the above are often written as shorthand containing the mass splittings as,

$$\Delta_{ij} = \frac{\Delta m^2_{ij} L}{4E} \tag{1.9}$$

Note that if $\Delta m^2_{ij} = 0$ then equation (1.8) trivially prevents any oscillation from occurring. Neutrino oscillations are therefore only possible when the masses of the different mass eigenstates are distinct (guaranteeing at least two non-zero neutrino masses). Moreover, there is no sensitivity of the oscillation probability to the absolute masses of the three mass states, rather the mass squared splittings. For the three mass eigenstates only two of the mass squared splittings are actually linearly independent as $\Delta m^2_{32} = \Delta m^2_{31} - \Delta m^2_{21}$. With knowledge of just δm^2_{12} one cannot say if m_3 is the largest or smallest of the three mass eigenstates, hence the uncertainty in mass ordering in Figure 1.3.

Of key importance is the spectral dependence of $P(\nu_\alpha \to \nu_\beta) \equiv P_{\alpha\beta} \propto \Delta_{ij}$. This proportionality of oscillation probability to propagation distance and energy allows appearance and disappearance studies to be performed at experiments with various baselines.

The oscillation probability also clearly depends on the three mixing angles defining the PMNS matrix $\theta_{12}, \theta_{23}, \theta_{13}$ and on the two linearly independent mass squared splittings Δm^2_{21} and Δm^2_{31}. Finally, the oscillation also depends on the Dirac CP-violating phase δ.

In the derivation of the oscillation formula (1.8) there have been a number of simplifying assumptions made, most prominently was the assumption that the propagating mass eigenstates were described by plane waves with a well defined momenta which were assumed to be equal for each of the mass eigenstates (i.e negligible wavepacket separation of the mass states).

To get the equivalent oscillation probability expression for anti-neutrino flavour oscillation, all one must do is replace U with U^* in equation (1.8), changing the sign of the last term.

The current global fit from experimental results for the three-flavour oscillation parameter variables [21] with NMO mass squared splittings, are as follows :

$\sin^2\theta_{12} = 0.304\ {}^{+0.013}_{-0.012}$, $\sin^2\theta_{23} = 0.570\ {}^{+0.018}_{-0.024}$, $\delta_{\text{CP}} = 195^\circ\ {}^{+51}_{-25}$,
$\Delta m^2_{21} = 7.42\ {}^{+0.21}_{-0.020}\ 10^{-5}eV^2$, $\Delta m^2_{31} = 2.514\ {}^{+0.028}_{-0.027}\ 10^{-3}eV^2$.

As a useful aid for visualisation purposes, the Figures 1.5 and 1.6 are included showing the neutrino oscillation probability over a range of L/E values using past oscillation parameters (and $\delta = 0$).

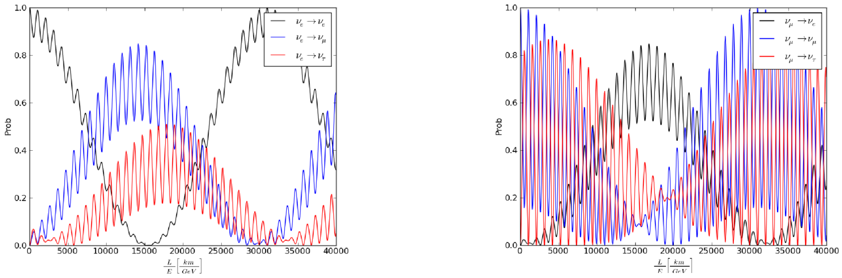


Figure 1.5: Oscillation probability of the electron and muon neutrinos to themselves (survival) or other charged leptons (disappearance). Probability calculated using (2010) oscillation parameters. Figure taken from [22].

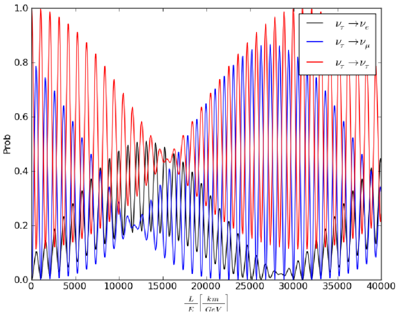


Figure 1.6: Oscillation probability of the tau neutrino to itself (survival) or other charged leptons (disappearance). Probability calculated using (2010) oscillation parameters. Figure taken from [22].

We now turn our attention to the interactions of neutrinos in the SM, and since gravity is currently not represented our exploration is limited to weak neutrino interactions. As neutral particles that only interact via the weak force we cannot directly observe neutrinos. The indirect signals of neutrino interactions are often the charged products of such weak interactions that can be more easily detected. Assuming the incident particle is a neutrino the two Feynman diagrams in Figure 1.7 depict the two of the major interaction modes available to neutrinos.

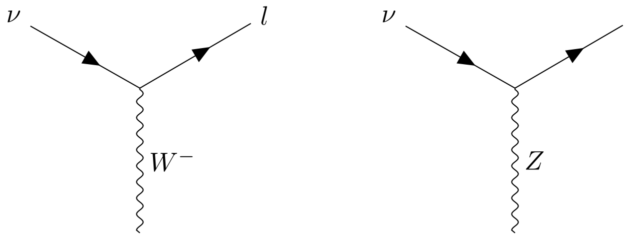


Figure 1.7: Two Feynman diagrams depicting the interaction vertex of neutrinos with the W^- and Z bosons. Created using [23].

These two vertices govern almost all of the interactions relevant for this work. In fact these vertices have already been seen in beta decay Feynman diagrams as well as the muon and tau decays. Reference to *charged current* (CC) interactions specifies that the gauge boson involved was $W^{\pm}$, conversely *neutral current* (NC) refers to those interactions mediated by the Z boson. In CC interactions the charged W boson is exchanged between a neutrino and its target (e.g ice nucleons). The neutrino is then converted into the corresponding flavour, charged lepton. The quark composition of the ice nucleon also changes, as the flavour of the interacting quark is transformed from an up type to a down type (or an up type from a down type).

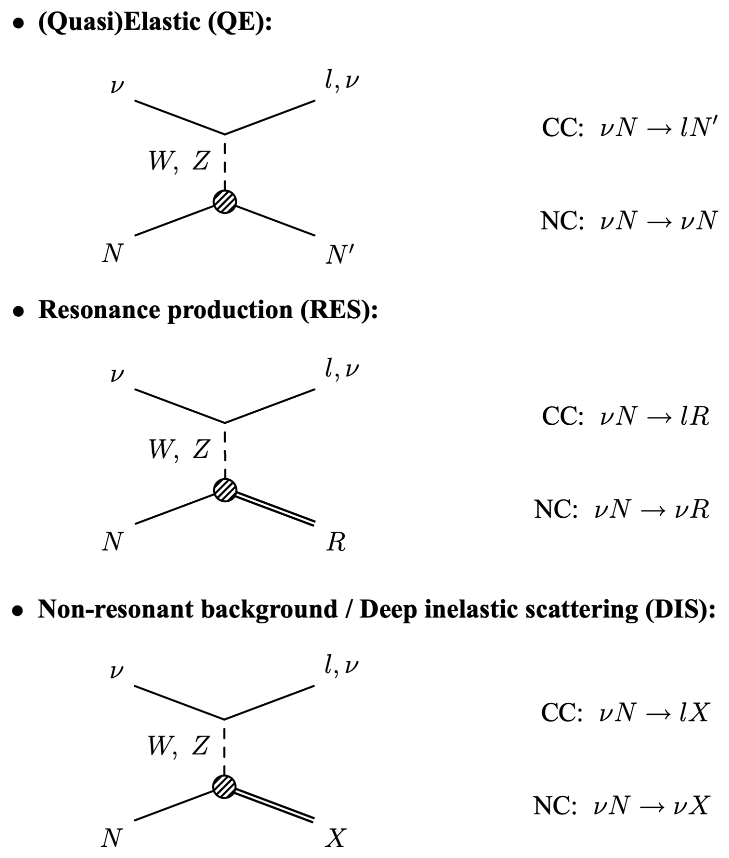
NC interactions induce no flavour change, and can be thought of as a similar process to a deflection or scattering of a charged particle via the exchange of a photon.

We may further divide both CC and NC interactions into three different processes based on the energies of the neutrinos involved. These processes are summarised in Figure 1.8 below.

The first such process, **elastic** (or **quasi-elastic**) scattering is the dominant component of the total cross section for energies below ~ 1 GeV [24, 25] (see Figure 1.9). At these energies neutrinos can be thought of as scattering off the whole nucleon, as such not "seeing" any of the substructure of the nucleon. The term quasi-elastic is used for CC elastic interactions since the neutrino becomes a charged lepton and the target nucleon also changes composition (e.g from neutron to proton), whereas in NC interactions in this energy range the type of participants in the interaction remain unchanged.

Resonant scattering of neutrinos from nucleons begins to overtake (quasi-) elastic scattering as the dominant process in the cross section for energies above 1 GeV [26]. This type of scattering transforms the nucleon into some excited baryon state, (or resonance), such as one of the delta baryons. This excited baryon state may then quickly decay into a combination of hadrons and mesons, (typically pions).

The final process, and for this work the most important, is **deep inelastic scattering** (DIS). At energies above ~ 10 GeV DIS provides the overwhelming contribution to the overall cross section. In events of this nature the target nucleon breaks down entirely as a result of interaction with the neutrino. Here, the neutrino interacts directly with the valence and sea quarks inside the nucleon, that is to say that the exchanged W or Z boson probes the internal structure of the nucleon. The final result is a hadronic shower of debris from the target nucleon. As with resonance production and quasi-elastic scattering, the difference between CC and NC interactions in DIS is the lepton or neutrino final state.



- **(Quasi)Elastic (QE):**

CC: $\nu N \rightarrow l N'$

NC: $\nu N \rightarrow \nu N$

- **Resonance production (RES):**

CC: $\nu N \rightarrow l R$

NC: $\nu N \rightarrow \nu R$

- **Non-resonant background / Deep inelastic scattering (DIS):**

CC: $\nu N \rightarrow l X$

NC: $\nu N \rightarrow \nu X$

Figure 1.8: Where here ν and l represents both neutrinos and anti-neutrinos and leptons and anti-leptons, of all three flavours and N is a nucleon. Taken from [27]

1.4 Cosmic Rays

Cosmic rays suffer from a slightly misleading name, arising from the fact that they were initially believed to be photons and by the time their composition was verified [29], the name had stuck.

Early researchers found that the level of ionizing radiation grew with increasing altitude; this disagreed with the accepted theories of the time, but implied that the source was extraterrestrial in origin. Conspicuously, the intensity of this radiation changes with latitude, indicating that at least part of the signal had a non-negative electric charge and was affected by the Earth's own magnetic field.

We now know this extraplanetary radiation as cosmic rays which are composed of high-energy, electrically charged particles. 90% of these "rays" are

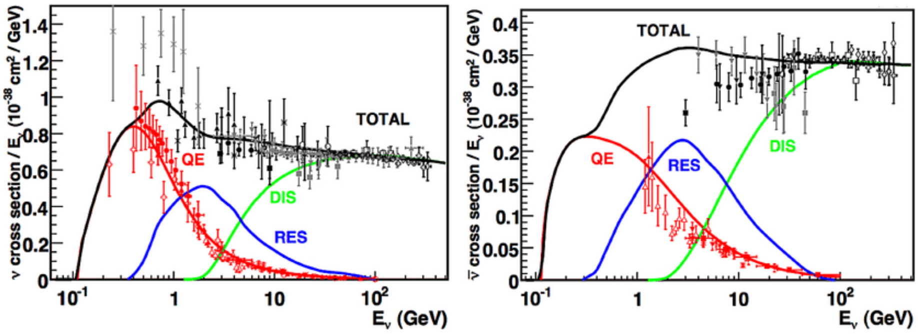


Figure 1.9: Inclusive cross sections for neutrino-nucleon (left) and anti-neutrino-nucleon (right) plotted as a function of energy. Contributions to total cross section from QE, resonant and DIS processes. Taken from [28]

.

composed of protons, 9% helium nuclei (α-particles) while the rest are made up of heavier nuclei[30].

These abundant, highly relativistic particles can be accelerated over large time scales to extremely high energies in energetic astrophysical processes such as supernova explosions [31]. Ultra high energy cosmic rays contain the most energetic particles observed in nature with energies in excess of 10^{20} eV. The energy spectrum of cosmic rays is shown in Figure 1.10 and underscores the vastly different scales of energies that contribute to the overall flux of cosmic rays. The cosmic ray population of energetic nuclei falls steeply with a near power law relation $\mathrm{d}N/\mathrm{d}E \propto E^{-\gamma}$.

The spectrum in Figure 1.10 steepens at around 10^{15} eV from $E^{-2.7}$ to $\sim E^{-3}$, which is often referred to as the "knee" of the spectrum [32], where the expected flux is around 1 particle per m$^2 s$. There is an additional softening of the spectrum at an energy around 10^{17} eV, before a further downturn named the second knee [33]. Around one to two orders of magnitude above the second knee, a final feature of the spectrum was named the ankle, where the index decreases back to approximately $\sim E^{-2.7}$ and the expected flux is 1 particle per km^{2}yr [34]. This "breaking" of the power law relation is theorised to be due to discrete transitions from different sources of astrophysical acceleration of cosmic rays.

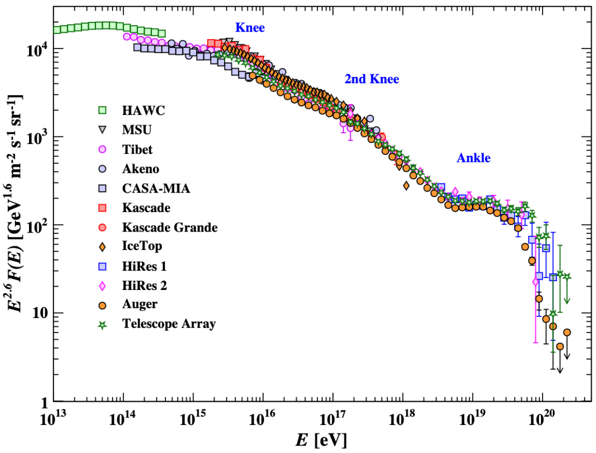


Figure 1.10: The all-particle spectrum of cosmic rays as a function of energy per nucleus, where the differential energy spectrum is multiplied by $E^{2.6}$ to emphasize the spectral features. Taken from [35]

1.5 Air Showers

An air shower is initiated when a single cosmic ray with sufficient energy to produce a cascade strikes molecules in the Earth's atmosphere and the resulting cascade is detectable at the ground. In general, showers have a hadronic core which acts as a collimated source of sub-showers. The development often occurs as a stochastic combination of three components: hadronic, muonic and electromagnetic [36].

Extensive air shower development in the atmosphere depends on a number of parameters including the Earth's magnetic field, seasonal temperature profile variations, cosmic ray flux and the details of the various hadronic and leptonic interactions that take place during propagation to the surface [37]. The modelling and simulation of both the cosmic ray flux and the interactions in air shower development are active areas of research [38, 39, 40].

Once the charged contents of the cosmic rays reach the Earth's atmosphere, they interact with the concoction of nuclei in air molecules inelastically, precipitating the breakdown of those involved nuclei. It is typical that in cosmic ray-atmosphere interactions many π-mesons, (and less abundantly K-mesons), are produced. These mesons, due to their instability, decay into subsequent mesons and/or combinations of leptons, the most common of which is the muon (or for π^+ the anti-muon).

The resulting neutrinos easily penetrate the entire diameter of Earth, as well as those muons with sufficient energy. The dominant production process is shown in Figure 1.11 and can be written via the leptonic or semi-leptonic

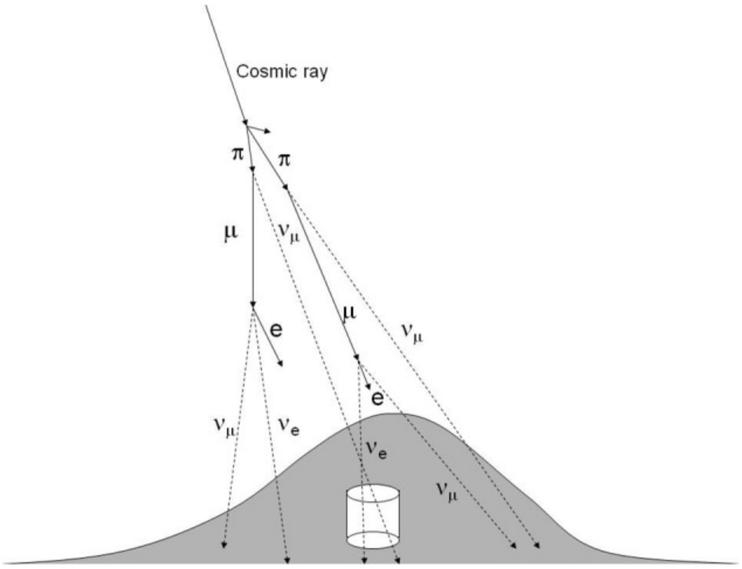


Figure 1.11: Development of air shower and neutrino production after cosmic ray primary incident on nucleons in the atmosphere. Taken from [17]

decays in the following decay chain:

$$\begin{aligned} p^+ + N &\rightarrow \pi^+ + \pi^0 + X \\ \pi^\pm &\rightarrow \mu^\pm + \nu_\mu(\overline{\nu_\mu}) \\ \mu^\pm &\rightarrow e^\pm + \nu_e(\overline{\nu_e}) + \overline{\nu_\mu}(\nu_\mu) \\ \pi^0 &\rightarrow \gamma\gamma \end{aligned} \tag{1.10}$$

Here, N is the target nucleus in the upper atmosphere, and X is the hadronic remnants of the initial interaction. Charged pion decays are overwhelmingly muonic, and since cosmic rays consist of more protons than anti-protons there is an excess of π^+ over π^- and hence μ^+ over μ^-. The neutral pion, and other neutral mesonic, component consists of electrons, positrons and photons, but the flux of electrons at the Earth's surface is dominated by (tertiary) muon decay.

Whilst neutral and charged pions are the most common products of cosmic ray interactions with the atmosphere it is possible that other mesons are produced, particularly for those with a more energetic cosmic primary. With increased energy heavier K-mesons can contribute to the neutrino flux, and at even higher energies charm (D) mesons may contribute to overall neutrino flux via semi-leptonic decay (sometimes called the prompt flux - appendix?).

The ratio of electron and muon neutrinos produced depends on the specific progenitor that interacted, but from equation (1.10) we note that the flavour ratio is 1 : 2 : 0 (as $\nu_e : \nu_\mu : \nu_\tau$). It is important to note that this flavour ratio is energy dependent, as longer lived particles will interact at lower altitudes (travel further towards Earth) rather than decaying, thus altering the flavour ratio of resultant neutrinos. Additional contributions at high energies, including those from the prompt flux, also allow for a small tau neutrino flux.

1.5.1 "Who Ordered That?"

Upon the surprise discovery of the muon Nobel laureate I.I Rabi is quoted as having questioned "who ordered that?". In IceCube and other neutrino observatories pursuing oscillation analyses the same question could be asked more than 2000 times per second.

As seen in the previous section the incidence of cosmic rays on the atmosphere and subsequent development of extensive air showers provides the sought after neutrino flux, but also a far larger and less desirable population of muons. At almost every step of meson decay paths, atmospheric muons are abundantly produced and given their relativistic speeds and relatively long lifetimes these muons are still the most populous energetic product at the Earth's surface. With sufficient energy, these muons may penetrate several kilometres underground.

Atmospheric muons hurtling towards Earth lose energy in a number of ways, which we will discuss here. As muons are charged particles there must be some energy lost via ionization of other charged particles the muon travels past. The muon interacts with the electric fields present in the matter it travels through and may excite/displace some of the bound electrons in the atmosphere/underground. Muons may also lose energy via radiative processes such as bremsstrahlung, e^+e^- production and other photonuclear interactions. All of the methods of muon energy loss described here have both continuous and stochastic components. Though once muons enter the ground the leading energy loss processes are ionisation and pair production [41].

Total muon energy loss may be expressed in the following way [2], as a function of the amount of matter traversed,

$$\frac{\mathrm{d}E_\mu}{\mathrm{d}x} = -a - bE_\mu. \tag{1.11}$$

Here a is the ionization loss and $b = b_{br} + b_{\mathrm{pair}} + b_{\mathrm{ph}}$ is the fractional energy loss via radiation processes. The quantity $\epsilon \equiv a/b$ ($\approx 500GeV$ for muons travelling through rock) defines a critical energy below which continuous ionization loss is more important than radiative losses.

The intensity of muons underground can be estimated from the muon intensity in the atmosphere and their rate of energy loss. Integrating (1.11) gives,

$$E_{\mu,0} = (E_\mu + \epsilon)e^{bX} - \epsilon \tag{1.12}$$

where $E_{\mu,0}$ is the energy of a muon at the point of its production in the atmosphere and E_{μ} is the average energy after propagation a distance X in ice. In the depth regime where $X \ll b^{-1} \approx 2.5$ km we can simplify (1.12) to,

$$E_{\mu,0} \approx E_{\mu}(X) + aX \tag{1.13}$$

While in the opposite limit $X \gg b^{-1}$ then the initial energy of the muon can be written as,

$$E_{\mu,0} \approx (\epsilon + E_{\mu}(X))e^{bX} \tag{1.14}$$

In general the muon spectrum at depth X is given by

$$\frac{\mathrm{d}N_{\mu}(X)}{\mathrm{d}E_{\mu}} = \frac{\mathrm{d}N_{\mu}}{\mathrm{d}E_{\mu,0}}\frac{\mathrm{d}E_{\mu,0}}{\mathrm{d}E_{\mu}} = \frac{\mathrm{d}N_{\mu}}{\mathrm{d}E_{\mu,0}}e^{bX} \tag{1.15}$$

Figure 1.12 plots the vertical muon intensity against the propagation distance, depth, in rock, with water and ice inset. The flattening of the curve occurs because of muons produced in CC ν_{μ} interactions in the matter. Muons lose less energy in water and ice so the gradient of the slope is decreased relative to rock. The penetration of muons deep underground will come up again later as we discuss those muons that stop inside of IceCube.

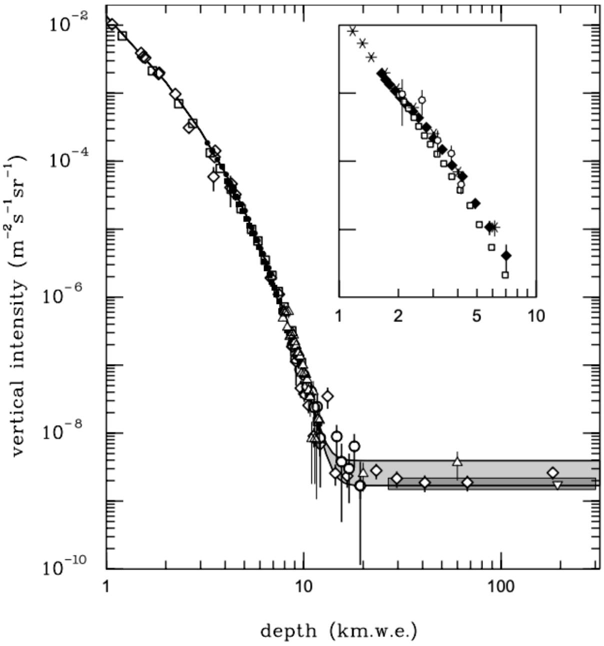


Figure 1.12: Vertical muon intensity plotted against depth. Shaded horizontal area represents neutrino induced muons.Taken from [2]

1.6 Cherenkov Radiation

One aforementioned difficulty that IceCube is forced contend with is the indirect nature of their detection. The neutrinos are neutral, have extremely low mass and often arrive surrounded by hadrons, muons and others in cosmic rays, making them very difficult to isolate and observe.

As a result of their very small interaction cross section it is extremely unlikely that a neutrino interacts with matter as it travels from its astrophysical source and/or the upper atmosphere. The magnitude of neutrino cross sections is one reason why the suppression of background sources of Cherenkov radiation is so important.

As we have already seen (in Section 1.3), we are forced to look for the charged secondaries or remnants that neutrino interactions produce. These charged particles can be found by detecting the Cherenkov radiation they emit.

Cherenkov radiation is emitted as a particle with non-zero electric charge passes through a dielectric at a speed greater than the phase velocity of light in the medium. The Cherenkov radiation is not released by the charged particle itself, rather it is a consequence of the medium being dielectric.

The electric field of the traversing charged particle polarises the medium (displaces the electrons from the nucleons in the dielectric) and radiation is emitted as the electrons return to their de-excited state.

If the charged particle was moving slower than the phase velocity of this emitted radiation then it would destructively interfere with itself, see Figure 1.13. However, when the charged particle moves faster than light, in the medium, the wavelets of the track constructively interfere causing the coherent emission of photons [42]. These photons, named after Cherenkov who first discovered them [43], disperse in the form of a cone with an opening angle relative to the direction of the charged particles travel dependent on its velocity. The Cherenkov angle θ_C depends on the refractive index of the medium, n_{medium}, and the velocity of the particle, v.

$$\cos(\theta_C) = \frac{c}{n_{\text{medium}} v} = \frac{1}{n_{\text{medium}} \beta} \tag{1.16}$$

where c is the speed of light in a vacuum.

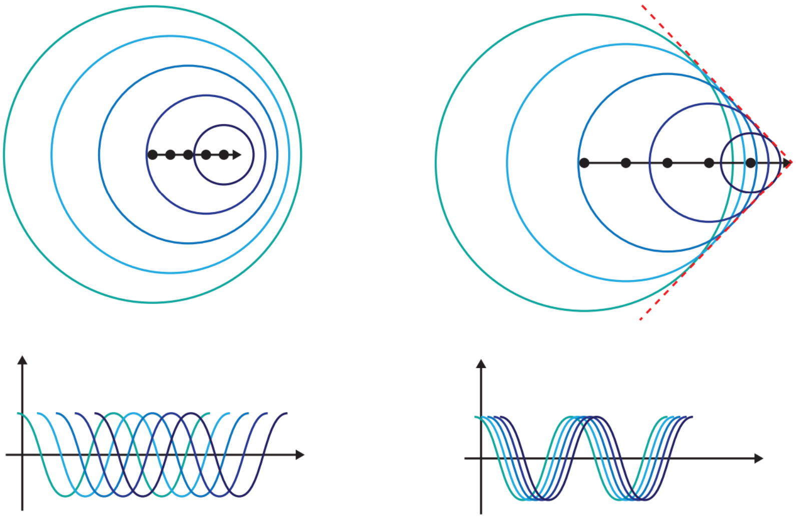


Figure 1.13: Demonstration of Cherenkov radiation emission by a charged particle moving to the right in a dielectric medium caused by polarization and later depolarization of the medium. Below is the electromagnetic wave in the medium for a specific point in time. Left: Situation when $v < c/n_{\text{medium}}$ and the disturbance cancelling. Right: For $v > c/n_{\text{medium}}$ constructive interference of electromagnetic disturbance producing plane wave cone centred on track. Figure taken from [44]

2

The IceCube Neutrino Observatory

The IceCube Neutrino Observatory, henceforth referred to as IceCube, currently consists of three parts; the IceTop surface array, the IceCube main array and the DeepCore subarray contained within. The design objective of IceCube is to detect resultant Cherenkov radiation in the ice from neutrino interactions that occur inside the fiducial volume of the detector. In principle, using the detector's response to the emitted Cherenkov radiation the identity and properties (such as direction, energy etc.) of the neutrino may be reconstructed. This task will be the main focus of this work.

The design of IceCube mimics other contemporary neutrino detectors in that the detection of neutrinos requires the careful observation of a large population of target nucleons, however unlike many other detectors IceCube uses a natural interaction substance (ice) that is also deep underground.

The predecessor of IceCube, AMANDA [45] provided evidence that optical sensors frozen deep under the Antarctic ice could still detect the relevant Cherenkov radiation while maintaining a low enough background rate so as to be feasible. Aside from the substantially increased detector volume (IceCube contains more than 7 times as many optical sensors as AMANDA) the physics programme is also more broad; alongside its function as an ultra-high energy neutrino telescope the IceCube detector (including IceTop and DeepCore) also includes studies of neutrino oscillations, supernova detection, analysis of cosmic rays and beyond standard model particle candidates among others.

IceCube as it functions today was completed 2011, where beforehand a subset of the total strings could be referred to, for example, as *IC78* for the 78 string configuration, that excludes the newest strings drilled for DeepCore.

2.1 The Digital Optical Module (DOM)

Cherenkov photons are detected in Icecube by the lifeblood of all neutrino observatories: digital optical modules. Each of these modules accepts photons

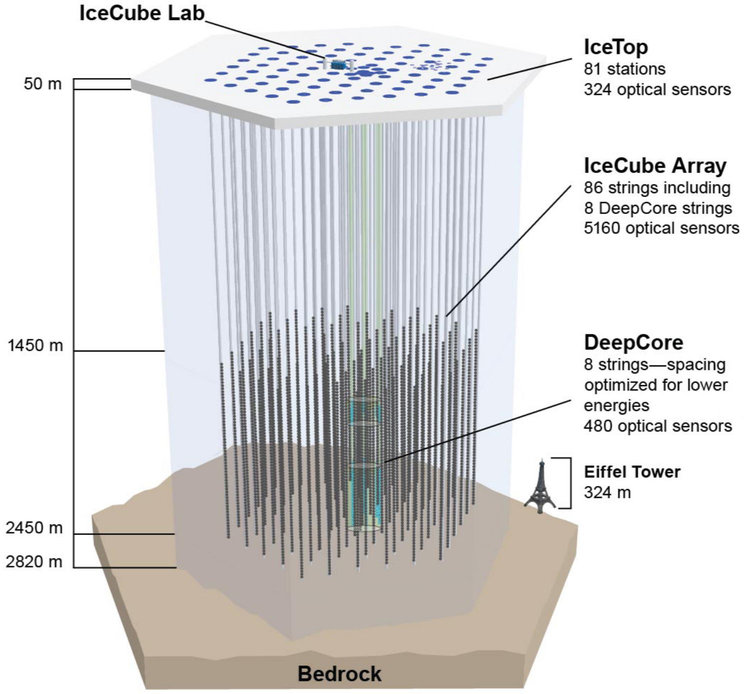


Figure 2.1: IceCube detector displayed in ice, with DeepCore highlighted in blue/green and IceTop by blue spots at surface.

and outputs a variable electronic signal based on the amount of received light. Each DOM in the IceCube main array contains a single downward-facing photomultiplier tube (PMT). The DOM is enclosed in a spherical layer of glass containing its own power supply and electronics responsible for digitizing and sending its response to the laboratory at the surface. In this way, DOMs may operate as a complete and autonomous data acquisition system.

Within a single DOM, data acquisition is initiated when the PMT signal exceeds some pre-determined threshold. When this occurs the DOM records the next 6.4μs and records the subsequent arrival of Cherenkov photons. This trigger determines the information that constitutes a hit in any future event pulsemap.

The PMT of a DOM in the IceCube main array is sensitive to photons with wavelengths between 300nm and 650nm, which coincides with the expected wavelength of Cherenkov radiation in ice. For DOMs in the IceCube main array, as well as those in DeepCore, the choice to point each PMT downwards (towards the centre of the Earth) was to reduce the likelihood of downward travelling atmospheric muons triggering a large number of DOMs (a "look the other way approach").

Additionally it should be mentioned that while the DOMs are incredibly reliable (as of 2016 98.4% of the more than 5000 deployed DOMs were fully operational [46]) and have an estimated life time of ≈ 15 years [47] they are also susceptible to noise of different forms including thermal noise, radioactive decays in the glass casing [48].

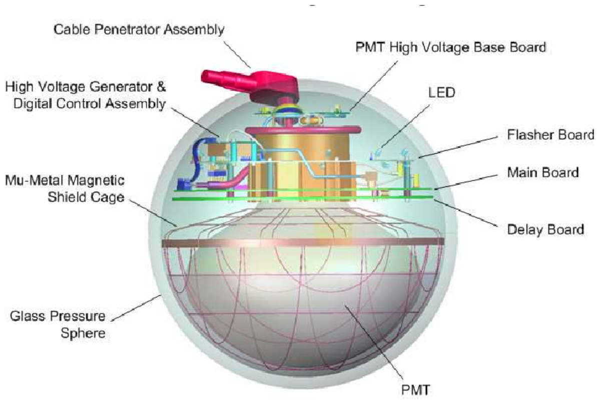

Cable Penetrator Assembly
PMT High Voltage Base Board
High Voltage Generator & Digital Control Assembly
LED
Flasher Board
Main Board
Mu-Metal Magnetic Shield Cage
Delay Board
Glass Pressure Sphere
PMT

Figure 2.2: A schematic cross-section of an IceCube DOM taken from [47].

2.2 Detector Geometry

Located at the south pole in Antarctica, IceCube is the largest neutrino detector in the world. One cubic kilometer of Antarctic ice is the detector medium that is instrumented by 5160 DOMs buried in the ice. The in-ice DOMs are suspended on 86 strings in a quasi-hexagonal grid shown in Figure 2.3. The DOMs that make up the main array have an equal vertical spacing of 17m and are deployed between 1450 and 2450m below the surface. That the 60 DOMs per string are placed at this depth is in part with the aim of reducing the number of background atmospheric muons that enter the detector, but also that the clarity of ice at this depth allows (high energy) Cherenkov photons to propagate for hundreds of metres. This allows a larger volume to be used for detection and a larger number of ice nucleons to be monitored. The horizontal distance between each of the strings is approximately 125m.

In IceCube's initial full operation there were 80 functional strings, however for the study of lower energy events a volume of ice would be required to be observed more closely, with a greater density of DOMs to detect less energetic/abundant Cherenkov radiation.

The DeepCore sub-array was optimised for the purpose of providing sensitivity to these lower-energy neutrinos. This goal is achieved via a combination of increased density of DOMs, higher efficiency PMTs and the use of the rest of the main array as a filter against the abundant atmospheric muon background. The configuration of DeepCore consists of eight new strings, six containing 60

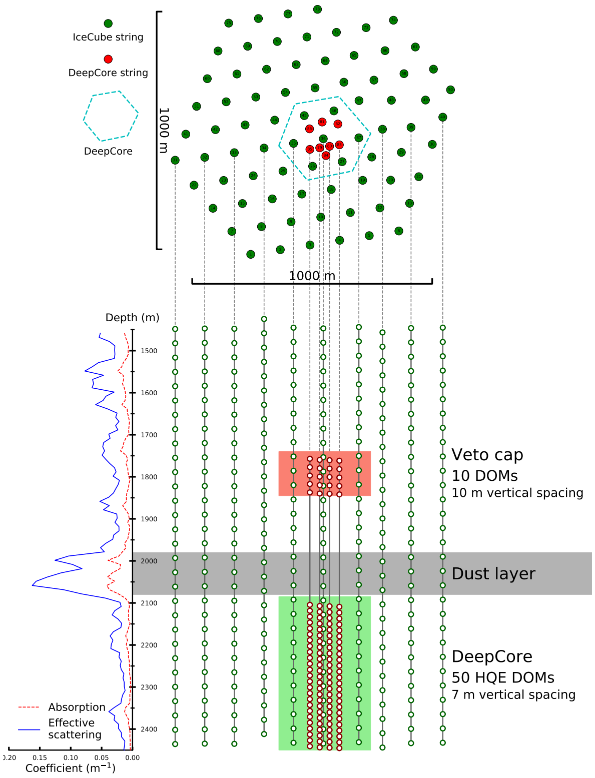


Figure 2.3: IceCube detector geometry from above and in-ice [49]. DeepCore sub-array displayed in green.

high-efficiency DOMs broadly arranged in a hexagon, and two DeepCore Infill strings containing a mix of high efficiency and regular DOMs.

As seen in Figure 2.3, between 2000m and 2100m there is an increase in the dust concentration. This so-called "dust layer" is a relic of the last glacial period 65,000 years ago. The quality of the ice as a function of depth was calculated using the IceCube main array DOMs taking local DOM to DOM measurements of responses to LED flashers. Consequently, DOMs on the DeepCore strings were placed above and below the dust layer to avoid the more opaque ice with greater absorption and scattering properties.

The first six high-efficiency DeepCore strings deployed are much closer together than the previous IceCube configurations, with an average horizontal inter-spacing of 72m [49]. The vertical separation of DOMs on all eight DeepCore strings is 10m above the dust layer and 7m below it. The 10 DOMs above the dust layer are referred to as the veto cap; they add an additional densely instrumented volume that can be used to tag atmospheric muons or other background events. For reference, low energy muons ($E_\mu \approx 1$ TeV) in ice travel approximately 5m per GeV of energy. Additionally, situating the majority of DeepCore DOMs at the centre of the main array allows the use of the outer layers of DOMs as a veto against muon events [50]. The DeepCore fiducial volume is made up of the 8 dedicated DeepCore strings as well as the closest IceCube main array strings. Reference to DeepCore denotes this collection of 15 strings.

The surface array IceTop (sometimes called the cosmic ray air shower array) consists of 162 ice-filled tanks [46] immersed 25m below the surface. Pairs of tanks 10m apart are stationed broadly above each of the strings, with the same nominal horizontal separation of $\sim$ 125m. Inside the ice tanks are a total of 324 DOMs, two per tank. Tank separation aims to ensure that single-station (two tank) hits are caused by air showers containing a muon capable of making it the further distance to the main underground array, as these muons are the main background source.

IceTop is sensitive to primary cosmic rays over the energy range PeV to EeV [51], this partially covers the "knee" region of the cosmic ray spectrum discussed in Section 1.4. For low energy neutrino analysis, IceTop's main utility is as a veto on downward going atmospheric background that triggers IceTop DOMs en route to the in-ice detector.

2.3 Event Recognition and Detector Output

The use of triggers in IceCube is necessitated by the vast amounts of data otherwise generated. The lowest level trigger is that discussed in Section 2.1 whereby DOMs only begin recording waveform information after a certain voltage threshold has been exceeded.

If a DOM registers a "hit", then the next hardware trigger concerns the response of DOMs in the immediate neighbourhood of the initial hit. This local coincidence criterion is passed if a neighbouring DOM's PMT signal also triggered within 1μs of the first DOM. These hits are then said to form a hard local coincidence (HLC) pair. Those DOMs that fail the local coincidence criteria will be saved and produce a less informative soft local coincidence (SLC) [52]. The data acquisition system in IceCube searches for clusters of HLC hits that are indicative of Cherenkov radiation emitted in a particle interaction in the ice, as opposed to random, uncorrelated noise hits.

At this point, different analyses require different filters based on the particular physics goals. Determining which combinations of HLC pairs and SLC DOMs to include varies between the different collaboration working groups. The **SMT8** (single multiplicity trigger) trigger requires at least 8 in-ice DOMs to satisfy HLC within a total time window of 5μs [49] before saving the detector response as an event. Each event is then expanded on the initial detector readout to include DOMs that registered a hit but may not have satisfied HLC. The extended time window is 20μs centred on the initial trigger time. The inclusion of concurrent SLC hits greatly improves our ability to reconstruct low energy events where otherwise the detector response could be incredibly sparse (low multiplicity).

For low energy analysis a different more specific filter is employed. Similarly to the single multiplicity trigger described above, **SMT3** requires just 3 DOMs to pass local coincidence, but in a 2.5μs time period and all 3 DOMs must be in the DeepCore fiducial volume. The veto algorithm takes advantage of its position at the centre of the main array and the DeepCore veto cap above the dust layer. HLC hits in the veto cap that hint at the downward travel of an atmospheric muon. The veto algorithm estimates the "centre of gravity" of an event (w.r.t the position and time) and this is used to calculate the speed of a hypothetical particle inducing each of the HLC hits, if this speed is approximately equal to the speed of light then the event is rejected and fails the DeepCore filter [49].

The DeepCore filter is applied to all triggered events to select events that have both the potential to have come from low energy neutrino interactions, but also the potential to be reconstructed accurately.

2.3.1 Event Selection - oscNext

Part of this work will look at the event selection process undertaken in the neutrino oscillation analyses herein referred to as oscNext [53, 54]. The as yet unpublished work aims to use all available livetime DeepCore data (2011-2019 inclusive), to calculate the neutrino oscillation parameters based on the relative appearance of the different neutrino flavours.

The primary objective of the oscNext event selection in DeepCore is to return a neutrino dominated event sample by removing and rejecting copious background from atmospheric muons and detector noise [55]. Since the rates of muons and noise are so high, the initial filters are required to be quick to remove those that are clearly background and reduce the subsequent number of events to consider. As the number of events to consider decreases, more accurate and computationally intensive methods of rejection may be used.

The focus for this project is placed on the first two steps in the filtration process, where we have yet to discard the entire muon population, however all 7 levels of selection are described extensively in [56].

Level 2 The input to the event selection process is the output as discussed in the previous section, i.e the common detector output supplied to the entire collaboration (satisfying SMT3). With the additional, low energy specific, constraint is the imposition of the DeepCore filter. Those events that pass the DeepCore filter then have a pulse-cleaning algorithm applied called SRT cleaning (see Section 2.3.3).

Level 3 The events that pass L2 are then processed by a number of simple, physically motivated cuts based on the nature of the pulsemap (DOM response). At this level many of the discrepancies between simulation and data are removed (e.g muon bundles and coincident events). The motivation of cuts at this level is speed, a low-hanging fruit approach to remove the easiest-to-remove muons and noise. The tactic broadly used for ridding muons from the sample is by looking for hits in the outer veto areas in the detector, while noise is often found by looking at the number or time separation of hits. The variables calculated are given in Table 2.1 along with the chosen cut-offs.

The definitions of each of the L3 variables calculated are as follows:

1. Cleaned Number of DOMs: The number of hit DOMs in the event after cleaning.

2. NoiseEngine: An algorithm designed to limit the number of noise events that may satisfy the lower SMT3 threshold required in DeepCore [48].

3. MicroCount Hits: The number of hits inside the busiest 300ns time window of an event.

4. Fiducial Hits: Number of DOM hits in the fiducial volume after cleaning.
5. NAbove200 Hits: The number of DOMs hit with depth greater than -200m.
6. Vertex Guess Z: The depth of the first DOM hit after cleaning.
7. Causal Veto Hits: The number of DOMs hit in the DeepCore filter.
8. Veto/fiducial hit ratio: The ratio of DOMs hit in the veto and fiducial detector volumes after cleaning.
9. C2HR6: The fraction of DOMs hit within the first 600ns of the first cleaned hit, excluding the first two.
10. RT Veto: The number of DOM hits found by the RTVeto algorithm, where the condition changes depending on the number of DOMs hit in fiducial volume [57].
11. Uncleaned Time Length: Duration of uncleaned event.
12. Cleaned Time Length: Duration of event after cleaning.

Variable	**Targets**	**Cut Value**
Cleaned Number of DOMs	Noise	≥6
NoiseEngine	Noise	True
MicroCount Hits	Noise	>2
Fiducial Hits	Noise	>2
NAbove200 Hits	Muons	<10
Vertex Guess Z	Muons	<-120m
Causal Veto Hits	Muons	<7
Veto/fiducial hit ratio Hits	Muons	<1.5
C2HR6 Hits	Muons	>0.37
RT Veto	Muons	True
Uncleaned Time Length	Coincident	<13000ns
Cleaned Time Length	Coincident	<5000ns

Table 2.1: Table of variables calculated in the determination of L3 passing events and the values of each of the cuts that defines their use in the oscNext analysis.

Shown in Table 2.1 are the selection criteria for events to pass to L3 of the event selection process. These cuts not only improve data-simulation agreement but also greatly reduce the rates of background while maintaining around 60% of the atmospheric neutrino signal events. The associated rates before and after processing are given in Table 2.2.

Event Type	**Rate L2+DC (mHz)**	**Rate L3 (mHz)**	**Eff.**
ν_e CC	1.61	0.95	58%
ν_μ CC	6.16	3.77	61%
ν_τ CC	0.193	0.129	67%
ν NC	0.86	0.53	62%
μ	7273	505	6.9%
Noise	6621	36.6	0.6%
MC Total CC	13903	547	3.9%
Data (2014 pass2)	16168	582	3.6%

Table 2.2: Table containing rates of the different particle events in IceCube at the first two levels of filtration. Neutrinos simulated with GENIE [58], muons with MuonGun [59] and noise with vuvuzela [48].

2.3.2 Distinguishing Events in IceCube

Now we have determined the checks that must be passed to be declared an event we will discuss the different types of events that can (and do) pass the relevant filters.

The reconstruction of events in IceCube is paramount to interpreting the detector output and distinguishing the different flavours of particle that were involved. In IceCube where DOMs are relatively sparsely distributed we characterise events into two different categories, based on the underlying physics interaction that gives rise to the detector output (see also Figure 2.4):

- **Cascade-like Events**; Secondary particles decay quickly and/or emit a large proportion of the total Cherenkov radiation in a localised area. The main candidates are charged current interaction involving the electron or tau neutrinos, and neutral current interactions of all flavours.

- **Track-like Events**; The long lifetime of the involved particles mean that Cherenkov radiation is distributed more evenly over the path (or track) that the travelling particle probes. The distance scales of such tracks can, with energetic enough particles, be much longer than the total detector itself. The main contributors to such events are atmospheric muons (not shown in Figure 2.4) and charged current muon neutrino interactions.

These events are defined by the topology of their energy deposition and Cherenkov emission. The different leptons flavours leave different footprints in the detector due to their differing properties. The tau lepton decays so quickly that it cannot travel far before decaying via methods described in Section 1.2.3, although in some (17%) cases the tau may decay into a muon that produces a more track-like detector response. The electron, unlike the tau, does not

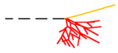
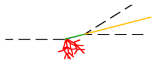
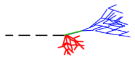
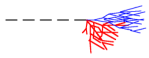
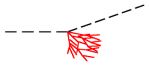
Interaction	Secondary particles		Detector signature
CC ν_μ		μ track and hadronic cascade	Track with cascade
CC ν_τ		τ decays into μ ($\sim 17\,\%$ b.r.)	
		τ decays into e / hadrons	Cascade
CC ν_e		Hadronic and EM cascades	
NC ν_α		Hadronic cascade	

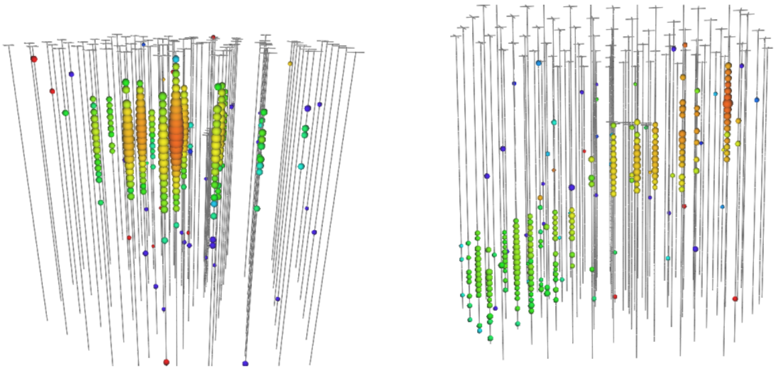
Figure 2.4: Possible interaction signatures of neutrino interactions with ice nucleons and the resultant particles. Dashed lines are neutrinos, orange lines muons and red lines for hadrons. Figure taken from [60]

decay and hence could travel long distances in the ice if it did not deposit its energy via Bremsstrahlung or ionization. Muons, both atmospheric and neutrino induced, leave the longest, track-like footprints in the detector since they have a longer lifespan than the tau and are heavy enough so that they do not lose all of their energy via Bremsstrahlung or ionization.

In all cases in figure 2.4 there is a hadronic shower present after the deep inelastic neutrino-nucleon scattering - these showers are the remains of the nucleon. The detector response to these hadronic showers is indistinguishable from the electromagnetic cascades due to the sparsity of IceCube DOMs. However, hadronic showers/cascades tend to be slightly smaller or more contained than their electromagnetic counterparts as neutrinos produced may carry some of the energy away, and the heavy rest mass of the hadrons themselves [61]. Additionally, the cascade-like events are more frequently entirely contained inside the detector and hence energy reconstruction is easier since less energy is silently carried out of the detector as in neutral current interactions.

By using the information output of the entire detector for each event we may construct a pulsemap of DOMs hit in the event, the pulsemap is the foundation stone of all reconstruction in IceCube. Examples are shown in figure 2.5 of three different pulsemap topologies.

At low energies one devastatingly important category of events are the **noise events**, where by noise we mean the spurious emission of pulses inside individual DOMs or those pulses not pertaining to any external physical Cherenkov emission. In these "events" no actual particle interactions involving



(a) Single Cascade

(b) Track

Figure 2.5: Detector output from three (relatively high energy) events in IceCube, hit DOMs represented by coloured spheres, where colour indicates time of hit (red early, blue late) and size is proportional to the charge recorded. The single cascade arises from a roughly spherically symmetric event with a single point light source (CC ν_e or ν_τ interaction). Track-like event caused by atmospheric μ or CC ν_μ, more evenly distributed in column-like volume.

muons or neutrinos occur, rather the detector noise alone satisfies the trigger conditions.

Noise in IceCube has two main components [62], these are thermal and non-thermal noise. Thermal noise comes from the spontaneous emission of single electrons from the DOMs photocathode due to its thermal energy. This emission is random, and uncorrelated between different PMTs as the number of thermal noise electrons is drawn from a Poisson distribution. Non-thermal noise arises from radioactive decays in the glass sphere around the DOM which can release bursts of several photons. The time of such radioactive decays is uniformly distributed in an event's time window.

2.3.3 Seeded Radius Time (SRT) Cleaning

Given a pulsemap as defined above one may "clean" the hit DOMs in an attempt to include only those that register hits with physics origins. In this way, it is hoped that enough DOMs remain for accurate reconstruction of the event while also removing those spurious noisy hits that decrease reconstruction accuracy.

The SRT algorithm combines cleaning based on separations in space and time of hit DOMs. The radius-time (RT) part of the test enforces that a hit is only kept if there exists an accompanying hit within a designated radius and also occurring within a set time of the first hit. The default settings for R and

δt are 150m and 1000ns respectively. Figure **??** illustrates the SRT process during one iteration.

For SRT cleaning not all hits are checked against the RT criteria - only those that are designated as a "seed", namely the HLC hits. This process happens iteratively over all HLC hits, and may add other SLC hits to the pulsemap, which are then tested against other hits, until no more hits are added. The philosophy of SRT cleaning is as such: begin with a subset of DOMs you are certain contain physics information (the seeds) and only add those that fulfill the requisite RT conditions.

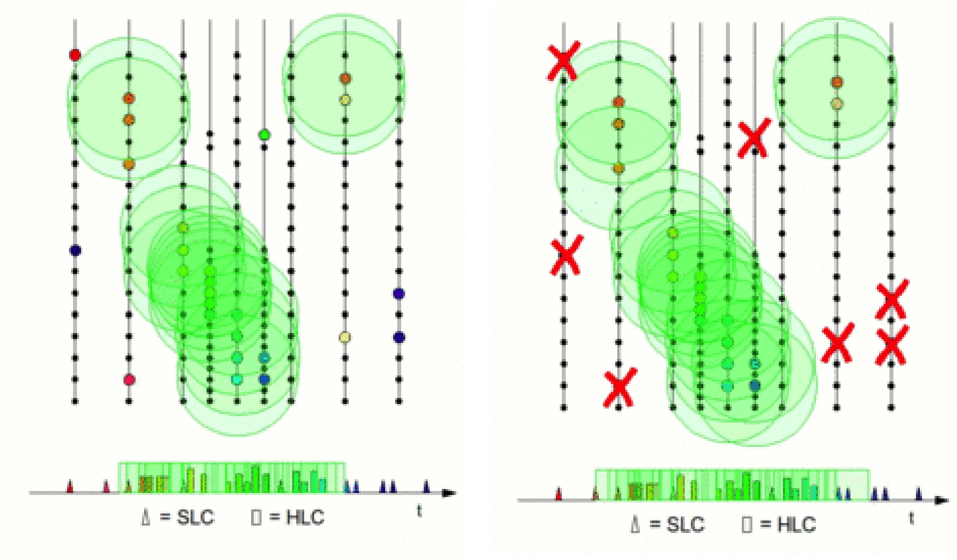


Figure 2.6: An example pulsemap with SRT cleaning being applied. a) Search for hits in RT-range of the initial HLC hits, the "seeds", and add those that satisfy to the list of seeds to check for further RT satisfying hits. b) After checking all new seeds iterate this process until no new hits are added to the seeds. Reject all those that are not in the list of seeds. Figure taken from [52].

2.4 Low Energy Simulation in IceCube

This analysis relies on a comparison of experimental to simulated data, and though never perfect, the statistical differences can give insight into assumptions and systematic errors afflicting simulation, or, more interestingly, unexplained behaviour of real data. For machine learning this problem is acute as the reliance on labelled, and therefore simulated, data requires the data-simulation discrepancies to be small. In this section we will briefly discuss the simulation used in both training and inference of our model as the underpinning physics and simulation pipeline goes far beyond the scope of this section.

There are three types of simulation used in this work: MuonGun [63] for atmospheric muons, GENIE [58] for neutrino events and vuvuzela [48] for pure noise events. The process of simulating all three types of low energy events is split broadly into three stages, which are as follows:

1. Primary particle generation and propagation through the ice. Daughter particles after interactions are also propagated.

2. Photons are propagated in ice, i.e paths of photons are tracked (travel and/or scattering).

3. Detector response - the output of the DOMs given the paths of the simulated photons. This step allows for variation of detector properties (e.g DOM efficiency) and response given the same input (photons).

2.4.1 MuonGun Simulation

At the energy spectrum we are interested in, the role muons play is of vital importance to any analysis/reconstruction. There is an abundant appearance of background muons, that must be simulated before it can best be mitigated. The overall idea of MuonGun is to remove the computationally costly and less controllable air shower simulations. Thus, MuonGun decouples the final state muon from the complex air shower development and constructs a parameterization [64] of the overall muon flux based on the depth under the ice, zenith direction, multiplicity and energy. This gives the user more control over the properties (energy, direction) of the resulting muon event.

One can think of MuonGun targeting just those muons coming from cosmic rays that are destined to interact in the detector's fiducial volume. This focus on detector-bound muons allows for the production of a high statistics sample weighted according to different muon flux models with differing assumptions about cosmic ray interactions in the atmosphere.

For low energy purposes muons are generated on a cylinder slightly larger than the IceCube volume and fired towards the detector (see Figure 2.7). Then, muons are chosen only if their energy and direction means that they will intersect a target cylinder surrounding the DeepCore fiducial volume. Only these muons will be simulated.

2.4.2 Genie Simulation

The neutrino interactions present in IceCube are produced via the GENIE simulation chain. GENIE simulates both the kinematics of the neutrino-nuclei interactions and also the subsequent decay processes that occur after the initial collision. In this framework the possible neutrino interaction cross sections are calculated for all flavours, in an approximate energy range from 1GeV to 1TeV.

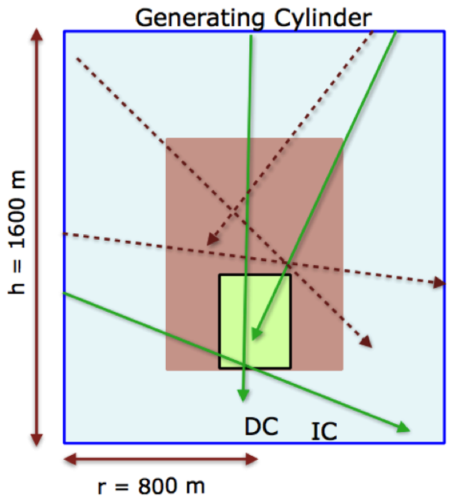


Figure 2.7: Generating cylinder in blue, IceCube in red, and DeepCore in green [59]. Muons are generated on the generating cylinder but are only "thrown" (simulated) if their trajectory intersects DeepCore.

This software is not specific to IceCube but is used widely in the experimental neutrino physics domain. The neutrinos simulated by GENIE are produced according to a power law energy spectrum, before they are made to interact with a target nucleon or electron within a given volume with the same density of targets as ice.

After calculating the cross-section for such an interaction (elastic, quasi-elastic, resonance production or DIS), GENIE also allows the final state hadrons to interact before leaving/exploding the nucleus. The results of the neutrino-nucleon interaction are then propagated individually through the ice whilst emitting Cherenkov radiation.

Note that the oscNext analysis that produced the GENIE simulation contains, for all neutrinos, an effective detector livetime of 70 years.

2.4.3 Noise Simulation

Non-physics DOM hits can cause a significant number of erroneously recorded hits or misfiring triggers, particularly before any cleaning or event selection has taken place. In order to accurately reflect the reality of operating an irreparable detector several kilometres underground we must simulate those events which are the least desirable - noise.

Noise is simulated in IceCube using the Vuvuzela model [48, 46]. In this regime both Poissonian and non-Poissonian contributions are combined to achieve the best data simulation agreement. Where the Poissonian component thought to be the result of radioactive decays or from thermal noise. Thermal noise contributes a rate of around 200Hz, whereas the radioactive decays typically produce a rate of around 100Hz. The non-Poissonian contribution adds an additional rate of 400Hz.

Noise simulation has a dual role in event generation. The first, is to include in muon or neutrino events the sporadic instances of noise triggered DOMs alongside the physics signal hits. The second function of noise simulation is to produce pure noise events, those where there is enough coincident noise that sufficient DeepCore triggers are passed and the entire pulsemap is saved.

3

Machine Learning

Machine learning (ML) is a term that encompasses a broad range of algorithms and modelling tools used in a plethora of different tasks related to the analysis of data. ML can be thought of as the combination of three integral components, data, a model and a loss function. The philosophy of ML may generally be thought of as a combining these three parts in a computationally efficient way while adhering to the scientific principle of "trial and error". ML has become more prominent in the particle physics community in recent years as the field has developed and new applications pursued.

In this section we will briefly recount the basics of ML and neural networks, the key lessons and ideas that underpin intuition in the applications of ML. We will then look at ML application to IceCube data before briefly looking at graph theory and the model used throughout this work - DynEdge.

3.1 Machine Learning Prerequisites

Machine learning models are in general not task-specific, in that they aim to improve the measurable performance (defined by a loss function) on any given task that is defined by the truth labels of the training data. They are hypothesis-agnostic in that if the pattern of the data truth changes so to will the model. That said, there are several types of learning problem that are best addressed using different approaches, including supervised, unsupervised and reinforcement learning. This project exists entirely within a supervised learning regime, which is the focus of this chapter.

It is also worthwhile to mention here that as ML draws from a wide range of fields with different customs, the terminology is rife with synonyms and overloaded definitions. For example, the input vector that is fed into a ML model is often called a variety of names, including: input vector, instance, sample, feature vector, covariates or attribute vector. Though in general the input can be a more complex structured object such as an image, sentence or graph. The value a model tries to predict or reproduce is called both the truth and label interchangeably while the prediction itself may also be called the output, score or response variable.

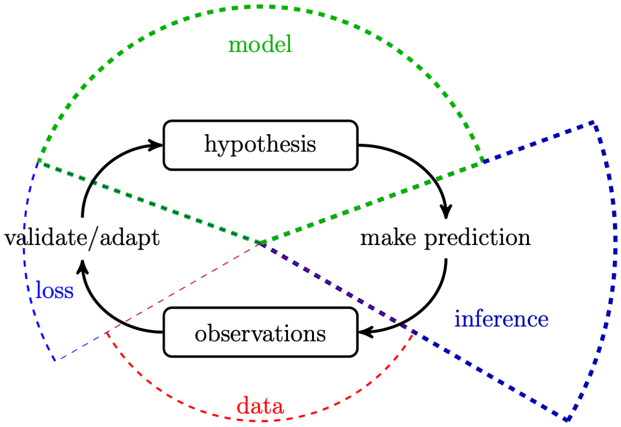


Figure 3.1: Overview of ML cycle. Make predictions using hypothesis, compare to observation, adapt/update hypothesis and repeat cycle taken from [65].

3.1.1 Supervised Learning

In supervised learning we are given a set of n samples of data, denoted $X_i \in \mathbb{R}^p$ (with dimension p) with $i = 1, ..., n$, each with an associated "truth" label $y_i \in \mathbb{R}^d$ that we would like to know.

The objective of supervised of learning is to find a function (or model) f that can take a new data sample X_j and after application of the function $f(X_j)$ can predict the label $\tilde{y}_j$ and that this prediction well approximates the truth y_j [66]. An equivalent perspective is that supervised ML searches for a hypothesis that can best reproduce the true labels of an as yet unseen data, called the **test set**.

In ML parlance, the set $\{X_i, y_i\}_{i=1,...n}$ of data is called the **training set**, where the function f "learns" which data correspond to which truth label. The generation of predictions is frequently referred to as the **forward pass** of training. The function that maps the input data to a predication, f, is commonly expressed as a set of parameters called weights $w \in \mathbb{R}^k$.

Training is completed by defining the loss function, $\mathcal{L}[f(X_i), y_i]$, that takes as argument, the model prediction for a particular sample and the corresponding label. This loss function allows us to judge how well the model currently explains the observations/data. The choice of loss function may vary depending on the nature of the problem and the behaviour that should be penalised/prohibited. The loss is aggregated (the sum is often taken) over all data samples in the training set and called the *training loss*.

It is this training loss that the training procedure aims to minimise in a process known as **back-propagation**. During back-propagation the weights w are updated in such a way that future predictions on the same samples would yield predictions closer to the true label itself.

The process of weight-updating is via gradient descent w.r.t the weights, w. Here the broad idea is to iteratively adjust the weights in the direction where the derivative of the loss function is large and negative. In such a way that local (and preferably global) minima of the loss landscape are found and the agreement between model predictions and labels improves in the context of the selected loss function. This places an important constraint on the loss function, at least for gradient descent minimisation, that it must be differentiable. We can update the weights in the "most improving" direction as,

$$w_{t+1} = w_t - \alpha \frac{\delta \mathcal{L}}{\delta w} \tag{3.1}$$

where the hyperparameter α is a constant termed the **learning rate** that determines the size of the step taken when updating the weights i.e the extent to which each update affects the overall magnitude of the weight.

When training the model, we do not have access to the test set (by construction) however we can partition the training set into two parts, one for training as defined above and one to act as a pseudo-test set called the **validation set** that we can use to measure the predictive power of a model on data it has not been trained on. This process of predicting, validating, loss calculation and updating weights continues for a pre-set number of **epochs** (or training cycles/iterations) or until some other user defined criteria is met.

This is, in essence, the strategy of ML, to cultivate a model f that generalises to unseen data by minimising the loss function for the available training data. And whilst the picture is, of course, more nuanced than the brief explanation above, it does give an instructive overview of the operation.

Separating the nature of a model's goal of a model defines two categories of supervised learning:

Classification When the truth labels take values in a discrete set. This can also be assigning a class label to examples from the problem domain such as in the classic example of classifying emails as "spam" or "not spam".

Regression Where the truth labels take values in a continuous range and the goal is to also return a continuous prediction. Examples include predicting the price of a house based on its area, number of bedrooms etc.

3.1.2 Loss Functions

In supervised learning tasks the choice of loss function by definition determines the quality of evaluation of a given hypothesis (set of weights). This is of vital importance as particular loss functions may be susceptible to outliers or unable to handle particular edge cases. Careful selection of a loss function can transform an ML problem and vastly improve performance.

The choice is especially important as loss functions can (and often do) have a very complex landscape with many local minima or saddle points where optimisation via gradient descent can, quite literally, lead down the wrong path. However, there are techniques designed to address this such as stochastic gradient descent [67] that allows the loss function to exit potential local minima. Manipulation of the learning rate may also mitigate the risk of local minima convergence (see next section).

The choice of loss function is therefore a balancing act between computation, robustness and interpretability. We will now look at some of the loss functions used in this project.

In binary classification tasks where ML models output a number between 0 and 1 that reflects the certainty with which some sample belongs in class 0 or 1. This prediction can in some sense be thought of as a probability of this event lying in class 1.

The **binary cross entropy loss** is a loss function designed to compare the dissimilarity of the truth labels, p_i, and the predictions, q_i, this is also sometimes referred to as **log-loss**. This may be written as,

$$\mathcal{L} = -\sum_{i=0}^{n} p_i \log q_i = -y\log(\hat{y}) - (1-y)\log(1-\hat{y}) \tag{3.2}$$

n is the number of classes, in this case 2, and the truth and predictions are y and $\hat{y}$ respectively. The second equality comes from the fact that this is a binary classification task. As there are only two classes we can state, without

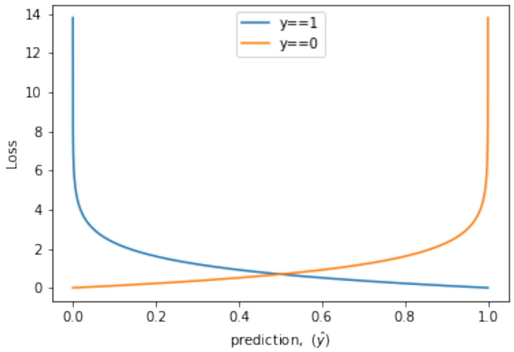


Figure 3.2: BCE loss for the two different truth categories (preliminary)

loss of generality, that $\hat{y} = q_{y=1} = 1 - q_{y=0}$. This step simply makes explicit that the prediction, $\hat{y}$, can be interpreted as $p(y = 1)$. From equation (3.2) it is clear that there are two modes (sets of behaviour) of this loss function. When the truth label is 0 the first term goes to zero, and conversely when the truth is 1 the second term vanishes. This has the intended effect of punishing those whose prediction is far from the truth. See Figure

For regression tasks the output of the model is not constrained, as is the case in classification, and so loss functions must be able to handle the increased domain/range of potential predictions. Among the most popular is the **mean square error** (MSE) which is simply the sum of squared distances between the truth and the predicted values.

$$\text{MSE} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i^2)^2}{n} \tag{3.3}$$

MSE is a quadratic scoring method which means the loss is proportional to the square of the error, and so outliers or particularly poor predictions will more dramatically affect the overall loss during training.

Another common loss function that instead scales linearly with the error, and is equal to MSE in its simplicity is the **mean absolute error** (MAE). The MAE measures the average of the absolute distance between the truth and predicted values.

$$\text{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{3.4}$$

The MAE works well, but around the minima the gradient is steep, which risks overshooting the optimal prediction during back propagation.

The combination of a linear and quadratic scoring method can be perceived as a happy middle ground, thus the loss function selected in this work is log-cosh loss. MSE is an improvement over MAE if our data contain large errors

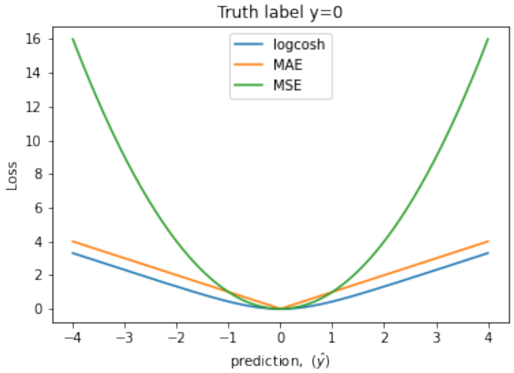


Figure 3.3: Common loss functions used for regression.

as MSE captures/responds to large discrepancies better. Unfortunately, the drawback of this is that MSE is much more sensitive to errors than MAE. Log-cosh is somewhat of a compromise as for small discrepancies/errors log-cosh behaves as MSE and for larger errors and outliers behaves more similarly to MAE.

$$\mathcal{L} = \sum_{i}^{n} \log\left(\cosh(y_i - \hat{y}_i)\right) = \sum_{i}^{n} \log\left(\frac{e^{(y_i-\hat{y}_i)} + e^{-(y_i-\hat{y}_i)}}{2}\right) \tag{3.5}$$

In this project a less commonly employed loss function is used for regression of the specific labels, azimuth and zenith, that is called the **von Mises-Fisher** (vMF) loss. This probabilistic loss function is an alternative to the more frequently seen cosine similarity [68] however both can be said to belong to the spherical family of loss functions and vMF is sometimes referred to as the circular normal distribution.

The intended design of vMF was as a solution in natural language processing, where words are encoded as vectors in some high dimensional embedding space, to reduce computational cost and increase the vocabulary variety possible [69]. At its most basic level the idea of the vMF distribution is to return a high probability when the vector in question lies close to the mean direction; in a directional analogue to a Gaussian distribution. The probability density function for an m-dimensional vector $\vec{x}$, is given as,

$$p(\vec{x};\vec{\mu},\kappa) = C_m(\kappa)e^{\kappa\vec{\mu}\cdot\vec{x}} = C_m(\kappa)e^{\kappa\cos(\Delta\phi)} \tag{3.6}$$

where $\vec{\mu}$ is the mean direction vector (the truth) and both this and $\vec{x}$ are of unit length, and $\Delta\phi$ is the angle between the true and predicted direction. κ is a positive scalar called the concentration parameter, the greater the value of κ the higher the concentration of the distribution around the mean direction $\vec{\mu}$.

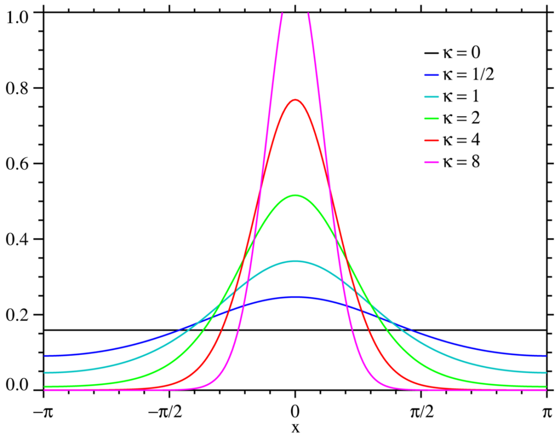


Figure 3.4: vMF distribution in two dimensions with $\vec{\mu} = 0$ and varied κ. Taken from [70]

One can interpret κ as playing the role of σ in a Gaussian pdf. The normalisation term C_m is defined as the following:

$$C_m(\kappa) = \frac{\kappa^{m/2-1}}{(2\pi)^{m/2} I_{m/2-1}(\kappa)} \tag{3.7}$$

and I_n is the modified, order-n Bessel function of the first kind.

For our purposes the two dimensional vFM distribution is sufficient as it describes the probability distribution on a $\mathbb{S}^1$ sphere, see Figure 3.4. We write,

$$p(\vec{x}; \vec{\mu}, \kappa) = \frac{e^{\kappa \cos(\Delta\phi)}}{2\pi I_0(\kappa)} \tag{3.8}$$

To this end the negative log-likelihood of this two dimensional distribution is simply,

$$\mathrm{NLLvMF}(\vec{x}; \vec{\mu}, \kappa) = -\ln(p(\vec{x}; \vec{\mu}, \kappa)) = -\kappa \cos(\Delta\phi) + \ln(2\pi I_0(\kappa)) \tag{3.9}$$

3.1.3 Hyperparameters

A key aspect of ML problems is model creation/development and while this is often completed via user defined design choices there are some overarching and impactful model parameters set before training begins called hyperparameters. Hyperparameters are used to configure various aspects of the learning/training process and can dramatically vary the resulting model performance. They are also notoriously difficult to find, and are often found through intensive, iterative processes (hyperparameter optimization). This is often impractical when the number of hyperparameters is large or with increasing model complexity, and significant research efforts have been focused on faster hyperparameter search algorithms [71].

In this section we will describe some of the most common and important hyperparameters for general models and their potential effectiveness on performance. It is a key balancing act of ML to select an appropriate set of hyperparameters and increase the model complexity to a level where training loss decreases but model generalisation is maintained.

The first hyperparameter, that we have already seen, is *learning rate*. The learning rate controls the magnitude with which the model weights are updated after back-propagation. For gradient descent optimization the learning rate defines the size of the steps taken, this is summarised in Figure 3.5, for a simple loss function landscape, in just one dimension.

This figure is illustrative of a wider point about the learning rate, that cannot be known in advance; if the learning rate is too small then the model will have to go through many rounds of forward passes and back-propagation updates to reach the minimum. On the other hand, if the learning rate is too large then the adjustments made to the weights may be overall in the correct direction, but far too large and over-shoot the desired minima. It is therefore difficult with a large learning rate to discover the subtleties of complex loss landscapes.

A descriptive metaphor for the learning rate could be that of a golf player attempting to put a shot. Too large a learning rate leads to the golfer continually over hitting his shot and the ball running past the hole, whereas too small leads to many tiny taps of the ball towards the pin.

An additional determinant of model performance is the number of times the model sees the entirety of the training data. Depending on the batch size, this is equal to the number of cycles through forward and backward propagation and is also called the *number of epochs*. There is something of a balance when training an ML model where one wants to capture all of the information in the training data (avoid underfitting) whilst also not becoming too specialised to the data seen in training (overfitting).

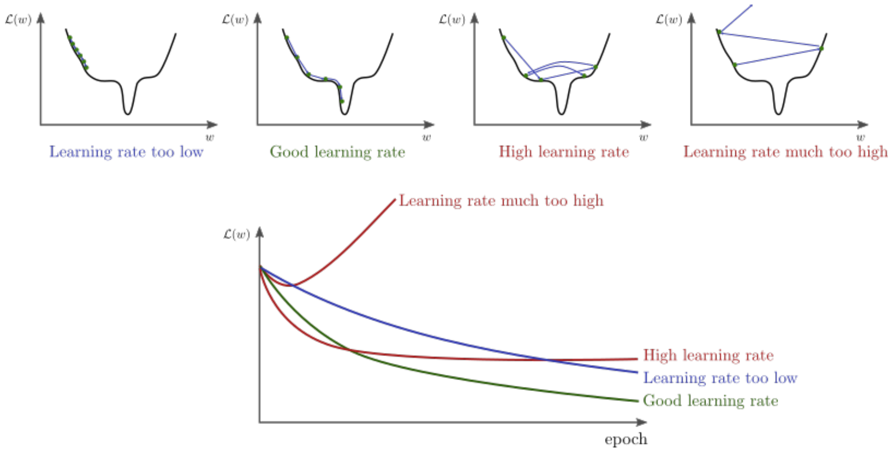


Figure 3.5: Fictitious plot depicting the effects of different learning rates. Lower learning rates will make linear improvements to loss whereas higher learning rates may "bounce" around in search of a new minima without settling in a minima. Taken from [72]

The aim is to learn just those features that are essential in the training data while ignoring the quirks and noisy fluctuations of the specific data in the training set, so that the model in question generalises to unseen data more effectively. While complex models may achieve the best results on the training set they may simply be erroneously recognising patterns that do not exist in real data.

Varying the number of epochs that the model is trained over can mitigate the risk of overfitting the training data however a more robust method exists that aims to address this problem, *early stopping*. Early stopping allows the model to monitor its own generalisation power by looking at the losses generated on predictions of the validation set and reduce the risk of overfitting. As shown in Figure 3.6 the loss for both validation and training sets decreases

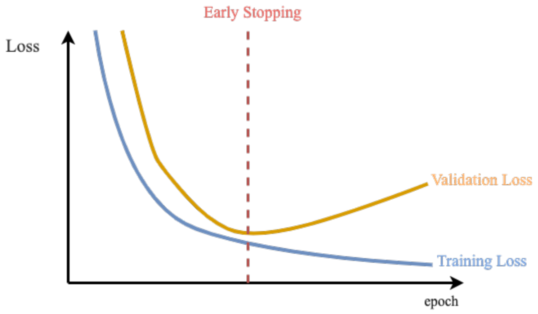


Figure 3.6: General example of overfitting of training data, with associated stagnation/increase in validation loss. Early stopping after arbitrary number of epochs mitigate disparaging loss scores[?].

in initial epochs, however after some number of cycles the validation loss ceases to decrease in tandem with the training loss. Beyond this point the model is overfitting to the training data and losing ability to generalise to the validation set. Early stopping is a form of regularisation that terminates the training process at the point where the validation loss starts to increase.

The aforementioned hyperparameter *batch size* defines the number of samples that are considered before the model weights are updated. This can happen many times during an epoch, each single gradient update is referred to as an iteration and when the batch size is equal to the size of the training set then one iteration is one epoch.

A small batch size will introduce a high degree of variance into the weight updates as a small amount of data dictates the back propagation modifications. Characteristics of each batch of data may not be reflective of the whole training set. Conversely, if the batch size becomes too large it may not fit in the memory of the CPU or GPU used in training and may also lead to poor generalisation (though in the general case this is still debated) [73, 74, 75, 76].

In ML it is difficult, if not impossible, to make sweeping general statements about the effectiveness of different hyperparameters on model performance as the behaviour for different datasets, model architectures and combinations of other hyperparameters. Batch size is no different, and further depends on the computational capacity available. Practitioners often opt for larger batch sizes to take advantage of CPU/GPU speedup, but this may come at the expense of the explorative properties of smaller batch methods that converge to different minima with different generalisation properties [77].

3.2 Neural Networks

Artificial neural networks (ANNs or just NNs) are a relatively old concept [78], especially in the ML community where contemporary papers are frequently rendered obsolete in a few years. While neural networks initially took inspiration from biological brains, there are now a multitude of different types of NN that vary considerably from the feed-forward NNs we will discuss now. This section is intended to act as an introduction to the mechanics of NNs and to provide context for future sections we will not discuss the rich field of exotic architectures and diverse parameter spaces.

NNs are nonlinear models that extend and generalise existing supervised learning methods such as linear/logistic regression[reference]. A standard NN is made up of a varying number of building blocks called nodes, neurons or sometimes units; each of these neurons take as input a vector, $(x_1, x_2, ...x_d)$ of arbitrary dimension, d, and returns a scalar output. Figure 3.7 displays the simplest NN, with just one neuron, that computes the weighted sum of the inputs, $z = \sum_j w_j x_j$ and then applies an activation function to return an output $\sigma(z)$.

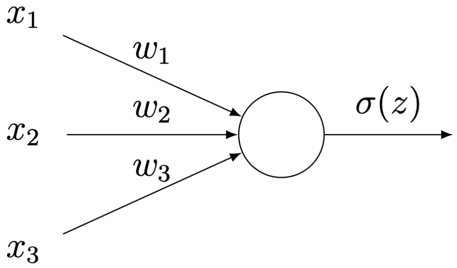


Figure 3.7: Simple NN with a single neuron taking three inputs, applying activation function to their weighted sum. Taken from the [MLBasicsBook]

Clearly, the weight, w_i, associated with each of the inputs x_i relays its importance to the overall neuron output, and this holds for more complex NN structures. Even the simple NN in Figure 3.7 defines a hypothesis space containing all the maps from input to output obtained with different choices for the weights and activation function. Note that this NN reduces to a linear map if the activation function $\sigma(z) = z$.

More generally, NNs are made up of many neurons arranged in layers where, crucially, the output of one layer is the weighted input to the next though in principle the interconnections between neurons can be arbitrary. The first layer that contains the data/features is called the input layer, subsequent layers are called the hidden layers and the final layer the output layer, as shown in Figure 3.8.

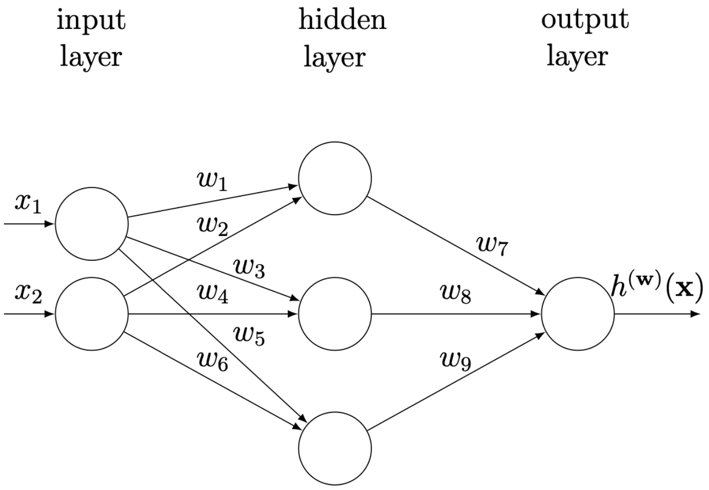


Figure 3.8: More complex NN structure containing just a single hidden layer with weights labelled. This model defines the hypothesis space consisting of all maps $h^{(\mathbf{w})}(x)$ that are obtained via all combinations of weights $(w_1, ..., w_9)$. Also taken from the [MLBasicsBook]

The weighted summation of inputs that happens within a neuron can sometimes also contain a re-centering term, the bias, b_i. Altering the above operation to become $z = \sum_j w_j x_j + b_j$. The choice of activation function will impact, or even negate, a particular neuron's contribution to the input of the next layer. The activation functions also add non-linearity to the NN map between input and output; without them the NN would simply be a linear regression model, as the composition of two or more linear functions is itself a linear function. The non-linear nature of the activation function allows the NN to perform non-linear regression and classification of data that are not linearly separable.

Six of the most common activation functions for neurons are shown below in Figure 3.9

One problem NN contend with is the vanishing gradient problem whereby gradient descent backpropagation methods update the weights in proportion to the partial derivative of the loss function w.r.t that weight, and in some cases the gradient will become vanishingly small preventing any significant update to this weight [80]. In the extreme this can prevent the model from training - an example is the use of tanh as an activation function. The derivative of tanh is sech^2 which is bounded from above and below in $(0, 1]$, successive derivatives in a chain rule multiplication decreases the magnitude of the total gradient exponentially and the gradient "vanishes".

Conversely, activation functions can also be susceptible to the aptly named

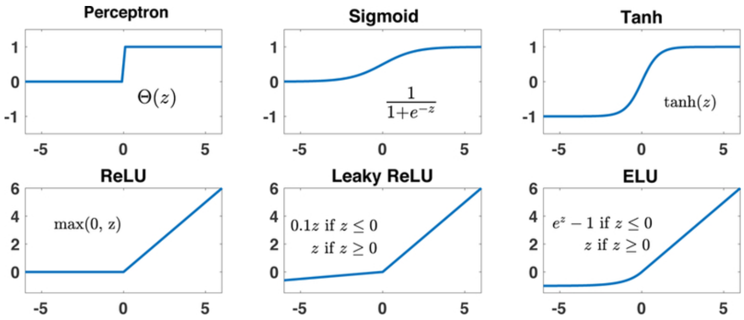


Figure 3.9: Common activation functions used in NNs. The discontinuous nature of the derivative of the perceptron (step) activation function means that it may not be used if optimised via gradient descent. The derivatives of the five other activation functions determine how they update during backpropagation. Taken from [79]

exploding gradients problem. This, as the name suggests, occurs when large partial derivatives of the loss function accumulate during backpropagation and result in large updates to the network weights, and consequently model instability. Practically such large gradients can produce overflow (NaN) errors that prevent further training.

The number of layers in a NN is called the network's *depth*, and if this is larger than some small integer then the model is often referred to as a *deep neural network* (DNN). The number of neurons in a layer is called the *width*. Increasing the width of layers and depth in a NN increases its *capacity* which means that the space of representable functions, the hypothesis space, grows.

Choosing the best NN for a given task is clearly non-trivial and requires extensive trial and error, experimentation and a little intuition. There is no recipe for the most performant architecture although it should be sufficiently complex as to avoid underfitting [81]. Therefore NN solutions require bespoke, task-specific considerations that best align with the training data at hand, which we will see more of in the next section.

3.3 IceCube Data As Input

The IceCube collaboration stores all data from physics runs and official simulation in the collaboration specific *.i3* file format.

These files contain a number of sequentially stored *frames* [82, 83] of which the three most important types are: **geometry** (G) that contain information about the detector geometry, **data acquisition** (DAQ or Q) frames that hold waveforms recorded for a particular event time window, **physics** (P) frames which contain all processing information such as reconstructions, pulsemaps and event identification information. Each P frame represents one IceCube event as defined by the collaboration wide filters.

The data pipeline used in this project relies heavily on the work of previous masters students at NBI [84, 85] who created a method for extracting the relevant information in *i3* files and reformatting this as a SQLite database [86].

Despite their ubiquity in the IceCube collaboration the *i3* file format is unsuitable for our machine learning purposes. Since the data is stored sequentially one cannot load only the frames they are interested in, they must load each of the frames in order before passing to the next. This makes their use in creating training and validation data sets incredibly difficult as these should be easily retrievable for every epoch in training.

The use of SQLite databases solves these issues and allows the user to simply query any specific subset of events inside the database prior to running training/inference. For each event in an *i3* file there is a corresponding event in the SQLite database with three associated tables. The first table contains the features extracted from the pulsemap. This contains the information pertaining to the detector response and a typical example is shown in Table 3.1, dom x,y,z and time refer to the location and time of the pulsed DOM (according to the IceCube conventional coordinate system), charge refers to the charge that the DOM measured, pmt_area the surface area of the PMT, rde refers to the relative DOM efficiency and width gives the width of the pulse in nanoseconds. These features are produced for both simulation and

charge	dom_time	dom_x	dom_y	dom_z	pmt_area	rde	width
0.675	8688.0	-256.14	-521.08	-406.07	0.044	1.00	8.0
1.275	9871.0	114.39	-461.99	107.19	0.044	1.00	8.0
1.425	12466.0	114.39	-461.99	107.19	0.044	1.00	1.0
0.775	9132.0	-334.80	-424.50	-486.61	0.044	1.00	8.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 3.1: A tabular representation of the feature information for one event in an *i3*-converted SQLite database. The number of rows is variable between events as it is the number of pulses in the extracted pulsemap.

real experimentally collected IceCube data and is precisely the input to future ML models.

The data in Table 3.1 is the input to any ML model, and for MC simulation we must have some target that our model aims to predict. These "truths" are given on a per-event basis and a subset of these are listed below.

- RunID; IceCube run identification number specific to *i3* files
- SubRunID; IceCube sub-run identification number
- EventID; IceCube event identification number specific to *i3* frames
- SubEventID; IceCube sub-event identification number
- event_no; Local unique event number serving as index
- pid; particle flavour (following [87])
- interaction_type; boson involved in neutrino interaction, CC or NC
- energy; energy of particle
- azimuth; azimuthal direction of travel of particle
- zenith; polar direction of travel of particle
- stopped_muon; boolean for muons that are contained in the detector volume
- position_x; x-coordinate of interaction position for neutrinos, projected stopping point for muons
- position_y; y-coordinate of interaction position for neutrinos, projected stopping point for muons
- position_z; z-coordinate of interaction position for neutrinos, projected stopping point for muons
- event_time; time of neutrino interaction
- DeepCore filter; boolean for events that pass DeepCore filter
- oscNext L3 pass; boolean for events that pass oscNext analysis level 3 filter

3.3.1 Stopped Muons

One of the stated aims of this work is to identify and reconstruct a sample of stopped muons. This begs the question: what does it mean to be stopped? This section describes the criteria and calculation performed to determine if a muon is stopped or not.

As described in Section 1.5.1 we know that muons lose energy radiatively as they propagate in ice. A subset of all muon events in IceCube will lose enough energy that they "stop" inside the instrumented volume of ice. The track-like signature of these muons will be interrupted and cut short. The stopped muon will then decay to a Michel electron and two neutrinos.

In this work a muon is defined stopped (and its *stopped_muon* boolean set to 1) when it's stopping point lies at least 150m inside the outer boundary of IceCube and at least 150m away from the top and bottom layer of DOMs. The stopping position itself is calculated for simulated muons by tracing the muons track length (the propagation distance) along the direction vector defined by the muon's true azimuth and zenith from the point of generation (in this case the MuonGun generation cylinder). This process is illustrated for for a small number of muons in figure 3.11.

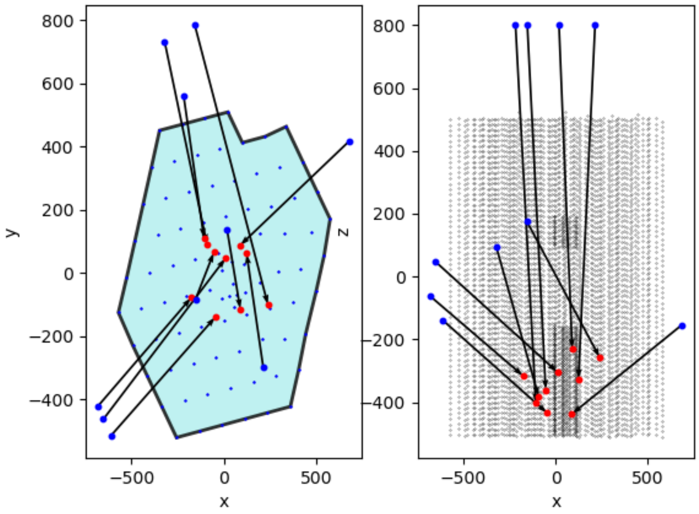


Figure 3.10: The start and finish positions of 10 sample MuonGun muons. Strings colored blue, outer boundary in black. Muons must be at least 150m inside the IceCube border and 150m above the lowest DOM and 150m below the highest in order to be classified as stopped.

Several MuonGun generation artefacts can be observed in the above figures, most clearly is the generation cylinder which presents itself as a circle in the

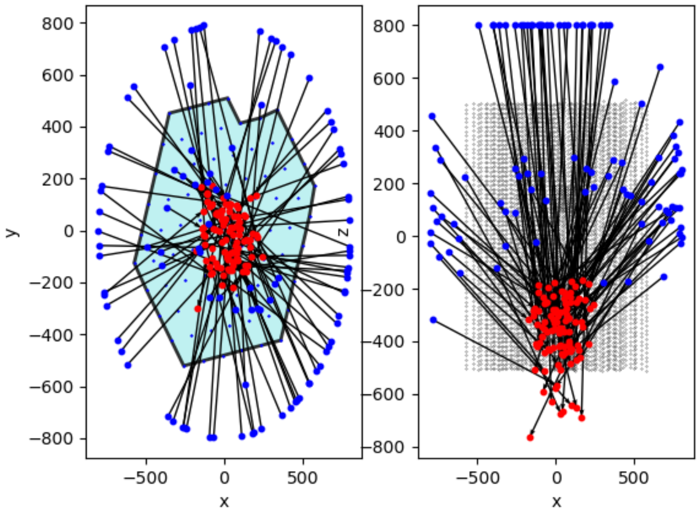


Figure 3.11: The start and finish positions of a sample of MuonGun muons. Strings colored blue, outer boundary in black. Muons must be at least 150m inside the IceCube border and 150m above the lowest DOM and 150m below the highest in order to be classified as stopped.

$x - y$ plane and an apparent majority of muons produced at $z = 800$m in the $x - z$ plane. The insistence that muons intersect DeepCore before they are simulated also skews our sample towards the central region of the detector. Though the difference this makes in real data is mitigated by application of the DeepCore filter.

3.4 Graphs and Graph Neural Networks

This section concerns the physically motivated application of graph neural networks (GNNs) to data output from IceCube. The motivation for this is clear, models that can leverage and incorporate our inherent prior knowledge about the structure of the graph data tend to generalise better [88], and graphs are a natural choice for IceCube events where data can be sparse, irregular and with a large number of dimensions. We begin with a short introduction to graph theory before looking at how GNNs operate in general. As a field, GNNs are very rapidly expanding and this section intends to introduce the mechanics that will be relevant in later sections.

3.4.1 Graph Theory

As most introductions to graph theory start, a graph is a mathematical object containing a set of entities (**nodes**) and the relations between them (**edges**). This definition can be further specialised by enforcing that edges should have an associated directionality. If edges have a defined direction/orientation they are said to be directed, otherwise they are undirected.

There are several useful matrix representations for graphs, the most common is the **adjacency matrix**. For a simple graph G, with a set of nodes (or vertices) V, and edges E, we may write $G = (V, E)$. An edge $e \in E$ has two endpoint nodes $e = u, v$ and we call u and v neighbours (or adjacent). We can then write the adjacency matrix $A \in \mathbb{R}^{n \times n}$ as,

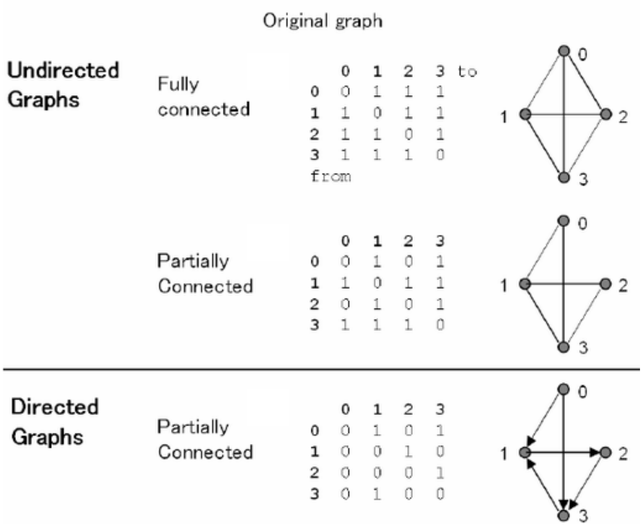
$$A_{ij} = \begin{cases} 1 \text{ if } \{v_i, v_j\} \in E \text{ and } i \neq j \\ 0 \text{ otherwise} \end{cases} \tag{3.10}$$

An adjacency matrix will always be symmetric if the graph itself is undirected. A simple example of directed and undirected graphs and their respective adjacency matrices is shown in Figure 3.12. Unfortunately, the graphs in this work will not be this simple.

The degree of a node is denoted by $d(v)$ and is simply the number of edges that it is connected to. In a directed graph this is the same as the sum of the number of edges flowing into the node and the number of edges emanating from itself. The degree matrix $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix of the following,

$$D_{ii} = d(v_i) \tag{3.11}$$

Many graphs display large heterogeneity in the capacity and intensity values of edges, where practically this means that not all edges in, directed or undirected, graphs have the same importance. One can encode this information into the graph structure by constructing a weighted graph where, associated



	0	1	2	3	to
0	0	1	1	1	
1	1	0	1	1	
2	1	1	0	1	
3	1	1	1	0	
from					

	0	1	2	3
0	0	1	0	1
1	1	0	1	1
2	0	1	0	1
3	1	1	1	0

	0	1	2	3
0	0	1	0	1
1	0	0	1	0
2	0	0	0	1
3	0	1	0	0

Figure 3.12: The adjacency matrices for different directed and undirected graph representations. Note that permutation of the nodes changes the adjacency matrix, but not the graph representation itself. Figure from [89]

with each edge e between two nodes i and j, there is an associated weight w_{ij}. A weighted adjacency matrix is simply defined in the same way as an adjacency matrix where non-zero entries are simply the weight w_{ij} rather than 1.

Graphs are an obvious choice for complex systems where additional information is stored in its topology. Examples include information or social networks and drug discovery.

The data collected in IceCube can be naturally expressed as graphs that preserve the detector structure and physics information in a way previously inaccessible to ML methods. Traditionally, ML paradigms coped with IceCube data by preprocessing the output for an event into a simpler representation such as a series of vectors. In doing so, these methods suffer from the loss of potentially important information or any topological dependency.

Composing IceCube data as graphs, while convenient, requires forethought about the connection of localised spatial data in a high-dimensional feature space.

A toy example of one IceCube event displayed in a graph structure is shown in Figure 3.13. Here we take the detector output data as described in Section 3.3 and create a graph where only those DOMs that triggered are considered nodes, with the edges left to be defined. The node features are simply those defined in Table 3.1, where each node in the graph stores all 8 columns associated with the selected DOMs row. Using graph representations

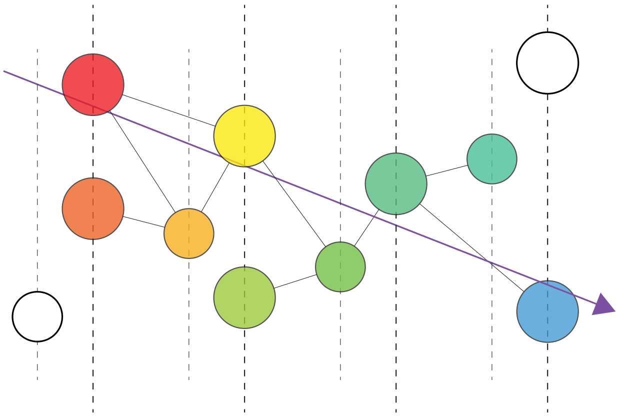


Figure 3.13: Triggered DOMs shown in color, red for early pulses, blue for late centred around muon propagation in purple. Untriggered DOMs in white. Edges are purely for illustration.

of IceCube events as inputs we may employ GNNs on event level graphs and reconstruct the truth labels for the entire event.

3.4.2 Message Passing Networks and Convolutional Layers

GNNs rely on an underlying **message passing scheme** that defines how information is shared within the graph structure. If we let a graph $G \equiv (V, E)$ with nodes $\{v, w\} \in V$ and edges $e_{vw} \in E$ have a vector of node features associated with every node x_v. The forward pass then proceeds in two phases, a message passing phase and a readout phase [88]. The message passing phase is defined in terms of messenger functions M_t and node update functions U_t. In the message passing phase the hidden states h_v^t at each node are updated based on messages m_v^{t+1} which are produced for iteration t as,

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t) \tag{3.12}$$

and the sum over $w \in N(v)$ is over all neighbours of node v defined as those with edges connecting the two. The nodes are then updated using this message according to the function U_t,

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{3.13}$$

The choice of messenger and update function is arbitrary, as long as they are differentiable and contain learnable weight parameters.

There are a number of different message passing schemes but for this work one particular family will be particularly important. This family, graph convolution neural networks [88, 90], can be interpreted as specialised message passing networks that utilise the graph Laplacian in the message passing phase. These layers tend to have simpler message functions but are typically applied on larger citation/knowledge networks that have clearly defined real valued adjacency matrices. [91] proposes the following layer-wise propagation rule:

$$H^{l+1} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^l W^l) \tag{3.14}$$

where $\tilde{A} = A + I_N$ is the graph adjacency matrix with additional self loops. $\tilde{D}_i i = \sum_j \tilde{A}_i j$ is the degree matrix of the graph with enforced self loops. Each $H^l \in \mathbb{R}^{N \times D}$ denotes the D dimensional node states (features?) for the N nodes in the graph, $W^l \in \mathbb{R}^{D \times D}$ is the learnable weight matrix. The function $\sigma(x)$ refers to a non-linear activation function, for example $\sigma(x) = \text{ReLU}(x) = \max(0, x)$ that was introduced earlier.

Reverting to our original notation, for the following we will write $M_{(v)}$ as the row in M indexed by v (which will always correspond to a particular node). For brevity let $L = \tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}$. A node state is then updated with

$$H_{(v)}^{t+1} = \sigma(L_{(v)}H^l W^l) = \sigma\Big(\sum_w L_{vw} H_{(w)}^l W^l\Big) \tag{3.15}$$

The v row of matrix H is equivalent to a column vector for node v we saw earlier h_v. Thus, equation (3.15) above is equivalent to the following.

$$h_v^{t+1} = \sigma\Big((W^l)^T \sum_w L_{vw} h_v^t\Big) \tag{3.16}$$

This allows us to write the message function while expanding L in more familiar notation as,

$$M_t(h_v^t, h_w^t) = L_{vw} h_w^t = \frac{\tilde{A}_{vw}}{(\deg(v)\deg(w))^{1/2}} h_w^t \tag{3.17}$$

And reflecting the form taken in equation (3.12) we write,

$$m_v^{t+1} = \sum_w M_t(h_v^t, h_w^t) = \sum_w \frac{\tilde{A}_{vw}}{(\deg(v)\deg(w))^{1/2}} h_w^t \tag{3.18}$$

Writing the message function in this way automatically determines the update function by comparing equations (3.13) and (3.16) we retrieve,

$$U_t(h_v^t, m_v^{t+1}) = \sigma((W^t)^T m^{t+1}) \tag{3.19}$$

By definition the matrix L_{vw} contains real, scalar values, so in particular for convolutional layers the message passing function is analogous to taking a weighted average of neighbouring nodes at each time step. Crucially, this effect compounds for multiple successive convolutional layers.

As aforementioned, in this project, we are only interested in graph-level classification and regression. Practically, this is achieved using a combination of graph convolution layers, graph pooling/aggregation layers and/or readout layers. This structural philosophy is also true for the model we use described in Section 3.5. Graph convolutional layers task is to extract high-level node representations of the input graph while pooling and aggregation layers coarsen the graph and play the role of down-sampling [92]. A readout layer collapses the node representations from each graph to a vector compatible with a multi-layer perceptron or another NN.

The readout phase simply computes a feature vector for the whole graph using some readout function R following,

$$\hat{y} = R(\{h_v^T | v \in G\}) \tag{3.20}$$

The readout step returns a graph-level output from all updated nodes and must be invariant under permutations of the node ordering. The nature of the readout function depends on the objective and desired output after a chosen number of rounds of message passing and updating phases.

3.5 The DynEdge Model

The model used throughout this project was extensively developed and tuned by a previous student, Rasmus Ørsøe, at NBI and can be found here [85]. The goal of this section is to explain the constituent parts of the model and their function. The results discussed in future chapters were all produced after training by the model architecture described in this section.

After having turned an IceCube pulsemap into a graph there is some feature scaling performed in order to reduce the complexity of the GNN loss landscape and allow the global minima to be found more easily. This is applied separately on the different features (dom_x, dom_time, charge etc.) as their scales vary by several orders of magnitude. Transforming the columns in this way brings each of their ranges to approximately the same order of magnitude and improves the stability of the model.

The model, named **DynEdge**, takes as input an event graph and passes each through four custom convolutional blocks. In each of these blocks a 2-layer perceptron, with 2 ReLU activation functions, takes the feature matrix as input and outputs an augmented "feature" matrix of different dimension. For every node in the graph, the operator convolves the nodes features x_j in the local neighbourhood (all those connected to the node in question) and updates the features as,

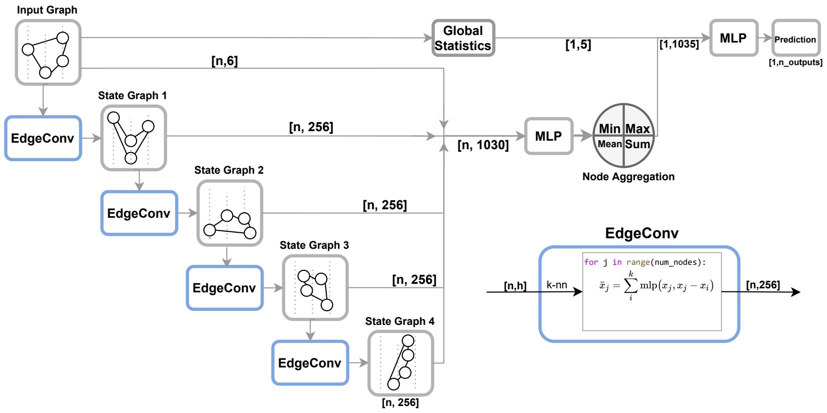
$$\tilde{x}_j = \sum_{i=1}^{N\text{neighbours}} \text{MLP}(x_j, x_j - x_i) \tag{3.21}$$

leaving the updated node features $\tilde{x}_j$ a function of the MLP, the original node features x_j and the pairwise difference between the node and all neighbour nodes. One full convolutional operation on the graph is complete when all nodes have been updated in this way. The output of the MLP need not be the same shape as the input features, and in fact, having more features gives the model more flexibility with which to update more learnable parameters.

Not only does the number (and therefore dimension) of "features" change but also their values, which are now the result of the MLP function on all of its neighbours and with the MLP weights and activation functions.

Before this new "feature" matrix is fed to the second convolutional block a new k nearest neighbours (kNN) algorithm is run in order to connect nodes in their new high-dimensional feature space to their new nearest neighbours. The second and subsequent convolutional blocks repeat this procedure until there are four additional graph representations of the initial event in total.

When all four convolutional blocks have been applied the feature matrices from all five graph representations are concatenated and fed into a shallow NN. After this stage an aggregation of the features occurs, along the dimension of



```
for j in range(num_nodes):
```

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

Figure 3.14: The DynEdge architecture visualised diagrammatically. The convolutional layer EdgeConv [93] occurs four times between graph reconnection, the EdgeConv operator is shown in the lower right.

the number of pulses (this step allows the theoretical size of the input graph to vary) where the max, min, mean and sum of the concatenated feature matrices is extracted.

The these pulse-wise aggregations are combined with global statistics of the initial graph (such as the node homophily ratio and the total number of pulses in the graph). This combination is then sent though a final NN which behaves as the readout function for this model.

4

Results

The aim of this work has been to use the new graphical ML approach to, for the first time, identify and reconstruct a large sample of stopped muons in both simulation and real experimentally recorded data. The real data used for the duration of this work is the entirety of the data taken on the 23$^{\text{rd}}$ January 2019.

The goal of this effort is to determine if we can use the muons that stop inside the detector as a standard candle - a well identified population with a particular detector response - to compare real data to simulation. It is only now possible to process the enormous atmospheric muon rate at low energies with the reconstruction speed and accuracy of GNNs.

The classifiers used also investigate the difference in results between L2+DC and L3 in the oscNext event selection chain as well as comparing classification performance on cleaned and uncleaned pulsemaps to test if the SRT cleaning algorithm aids or hinders our ability to reconstruct the muons.

After classification we engage in a full reconstruction of the classified stopped muons. This allows us to place the real data and simulation in juxtaposition and compare the two and scrutinize any differences.

The results presented throughout this chapter were all obtained using the DynEdge model architecture described in the previous chapter. The training sets used are described in more detail in the appendix. That said, it is meaningful to have some insight into the 24 hours of real data extracted from the IceCube servers at each stage of our classification process.

On 23$^{\text{rd}}$ January 2019 there were $1,420,709$ L2 events that passed the DeepCore filter, it is with these events that we begin our journey. After application of our noise classifier we lose $576,611$ events as noise leaving us with $844,098$ muon or neutrino events. The muon classifier removes just $65,764$ events leaving us with $778,334$ events DynEdge has classified as muons. Finally, the stopped muon classifier picks out $157,607$ events as stopped. This leaves us with an efficiency of slightly over 10%. These chosen events are the subject of the full reconstruction work in the upcoming subsections. Despite starting with data from just 24 hours of detector livetime a stopped muon sample this large has never been found before.

4.1 Classification Of Level 2 Data

Noise-Particle Classifier

The first classification performed for this project is potentially the most important - the noise classification; this could not have been completed without the aid of current NBI master student Kaare Iversen. This classifier, trained on 5 million simulated events (2.5 million noise events, 2.5 muon/neutrino events), attempts to separate pure noise from those particle-induced events by giving each event a score (or probability estimate) with higher scores indicating the model's belief the event was caused by a particle.

The DynEdge model, configured with 8 k-nearest neighbour graph constructors, used the prepackaged PyTorch [94] binary cross entropy loss during training.

Figure 4.1 shows the application of the model to L2+DC data, the initial level of event selection considered in oscNext. In this figure we see that the model is able to separate noise and physics events well (also see figure 4.2) and that this is true for both simulation and real data, though real data contains more events in the central bins where the model is less certain. Since the scale is logarithmic the real data in the central bins is not an issue, almost all the real data is in the outer bins. See Appendix for unscaled histograms for all classifiers

Another way to measure the classification power of a model is to look at the ROC curve, [95], which is a diagnostic aid plotting the proportion of true positives against the proportion of false positives for all possible values of the threshold parameter (noise cut). Figure 4.2 shows the ROC curve for this classifier and also another model trained on cleaned events only. The impressive performance of the model makes the resulting figure slightly hard to read. The main takeaway is that a perfect classifier has a ROC curve that extends maximally into the top left corner of the FPR against TPR plot or, in other words, has an area under the curve of 1.

Pursuing our overall aim of finding a pure stopped muon sample we must now choose to make a cut in the noise classification score, that best reduces the amount of noise in our event sample while keeping the most "signal" events (stopped muons). In many cases the choice of the cut value can have large implications with a trade off between purity and efficiency. Fortunately, in our case we have a little flexibility depending on the overall motivation of a project. Figure 4.3 shows the fraction of events that remain for each particle type for a given cut in the noise classifier.

At our chosen cut of 0.9 the fraction of noise remaining after application is 0.0014%. Throughout the rest of this work we will now only deal with events that have a *prob(particle)* score of greater than 0.9.

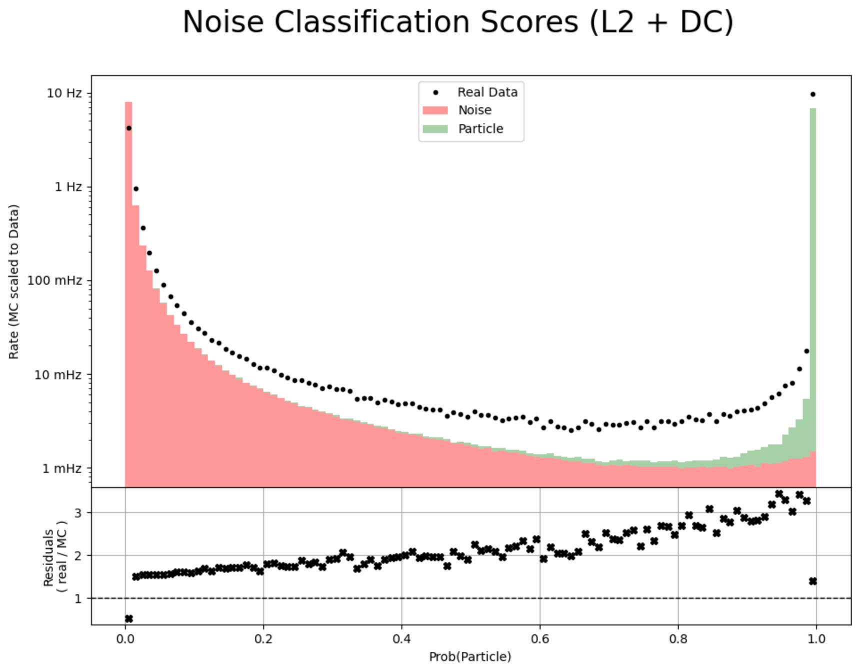


Figure 4.1: Stacked histogram of particle probabilities predicted by DynEdge on both simulation and real data L2+DC. Rates calculated using oscNext flux weights, and MC scaled to reflect rate of real data collected. This histogram was produced using a model trained on a balanced set of 5 million events and inferring on 48 million events using uncleaned (SplitInIcePulses) pulsemap.

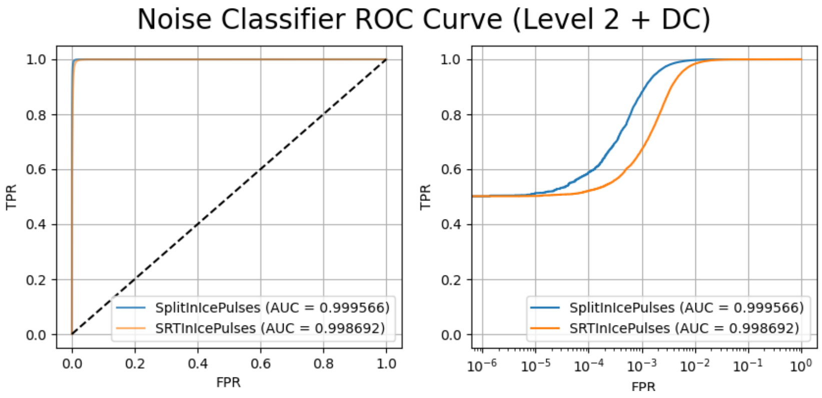


Figure 4.2: ROC curve for two unbiased noise classifier both trained on 5 million L2+DC events using cleaned (SRTInIcePulses) and uncleaned (SplitInIcePulses) pulsemaps respectively. ROC curve produced from the same data in balanced validation set of 1.6 million events with corresponding cleaned/uncleaned pulsemaps. Dashed line is a random (coin flip) classifier.

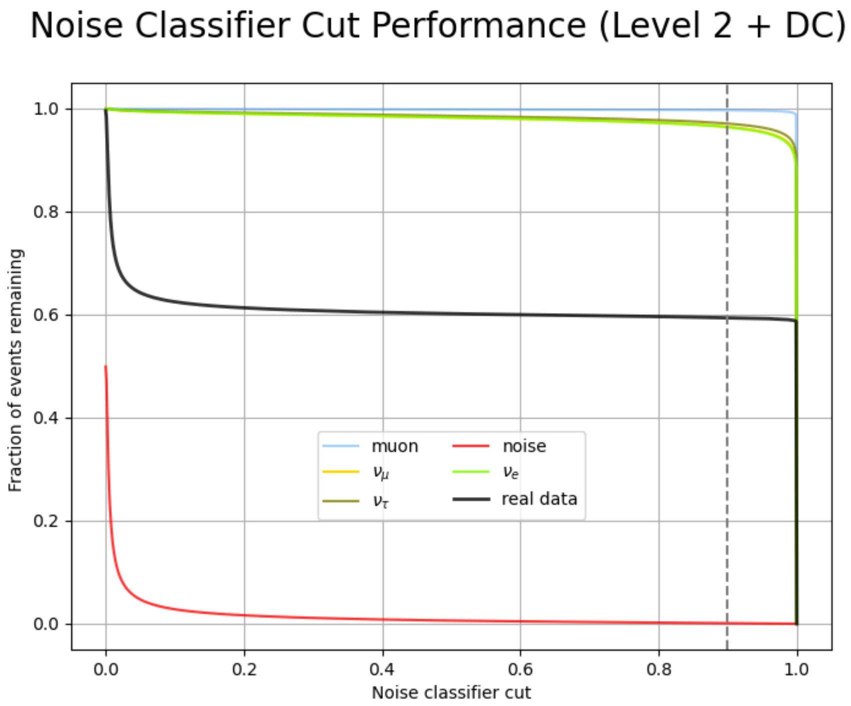


Figure 4.3: Noise cut performance on test sample of events (here events are unweighted) split by pid for SplitInIcePulses trained noise classifier. Performance is tracked by looking at the fraction of events that remain (w.r.t no cut) and the selected 0.9 cut used in this work is displayed in grey.

Muon-Neutrino Classifiers

The data passing the initial noise classifier is then processed by another classifier. This classifier has been trained to separate muons from neutrinos on 5 million events (evenly split between neutrinos and muons). For some analyses the classifiers ability to provide a pure neutrino sample is the foremost desire (see figure 4.4) but for our work we require the far more abundant muon events - figure 4.5 - in any case the ability of the model to generalise and separate muon and neutrino induced detector signatures is of paramount importance.

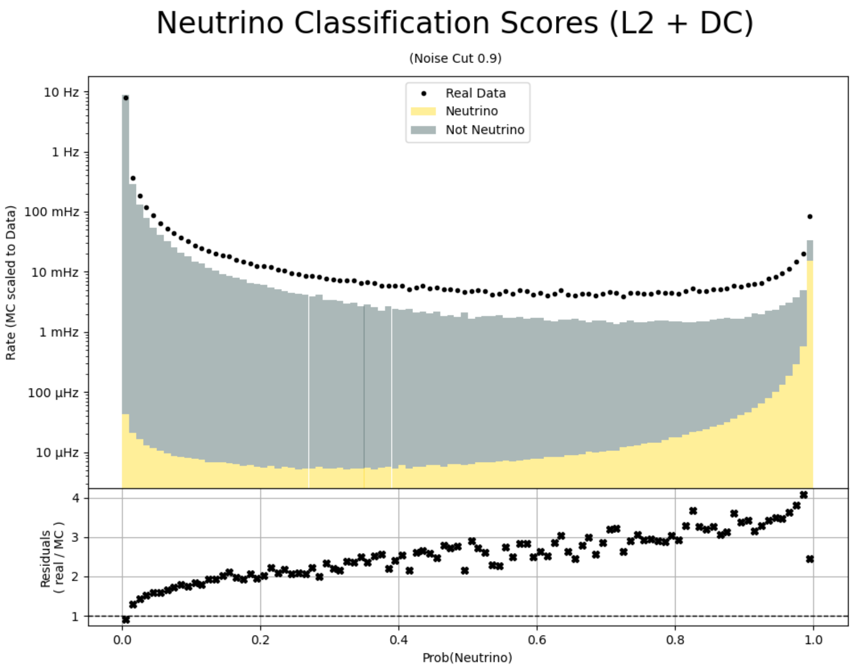


Figure 4.4: Stacked histogram of neutrino probability score predicted by DynEdge on simulation and real data at L2+DC that passes noise cut. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected. Model trained on, uncleaned (SplitInIcePulses) balanced 5 million event sample, and inferred on 48 million test set.

The model still has enough power separation to separate the muon and neutrino events. This is despite the fact that in both simulation and real data there exists some noise leakage. The data-simulation agreement for this classifier is worse than the previous noise classifier - the model predicts more neutrinos than one would expect, but the most populous bins show better agreement. As before, we use the ROC curve to evaluate the classifiers performance for our simulated test set. This is shown in figure 4.6 and once again we compare models trained on cleaned and uncleaned pulsemaps and draw the conclusion that the model trained on uncleaned events proves to have a marginally larger area under the curve.

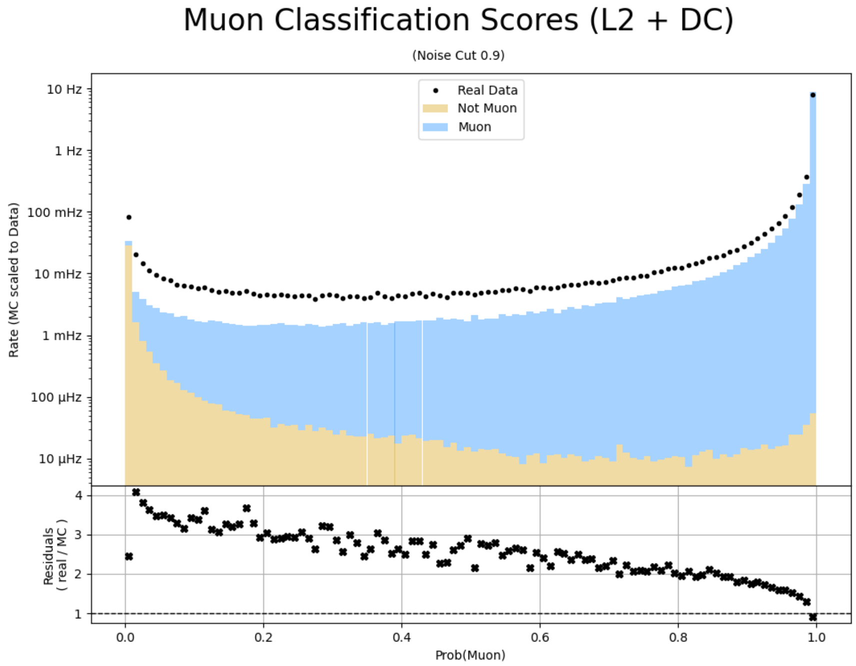


Figure 4.5: Stacked histogram of muon probability score predicted by DynEdge on simulation and real data at L2+DC that passes noise cut. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected. Model trained on, uncleaned (SplitInIcePulses) balanced 5 million event sample, and inferred on 48 million test set.

Once again, we are forced to make a cut based on the models muon score, the effects of which are displayed in figure 4.7. Here, depending on which particle population you are aiming for even a loose cut can provide a sample dominated by either muons or neutrinos. In this case a cut of 0.9 is applied to the *prob(muon)* score.

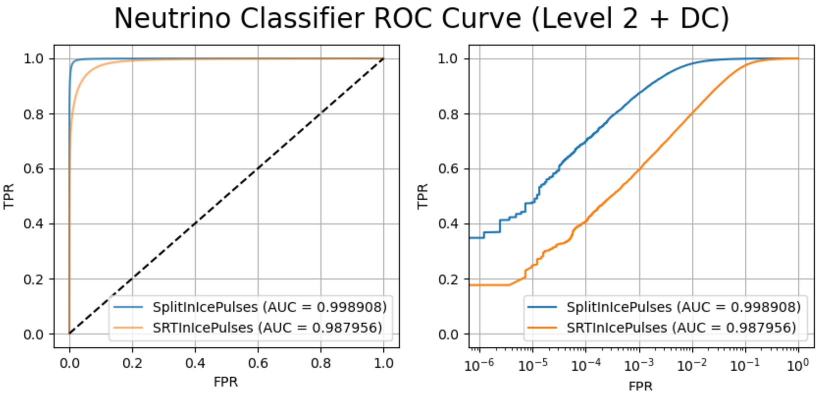


Figure 4.6: ROC curve for two neutrino classifier models each trained on 5 million L2 + DC events using cleaned (SRTInIcePulses) and uncleaned (SplitInIcePulses) pulsemaps. ROC curve produced from the same validation set containing 1.6 million events.

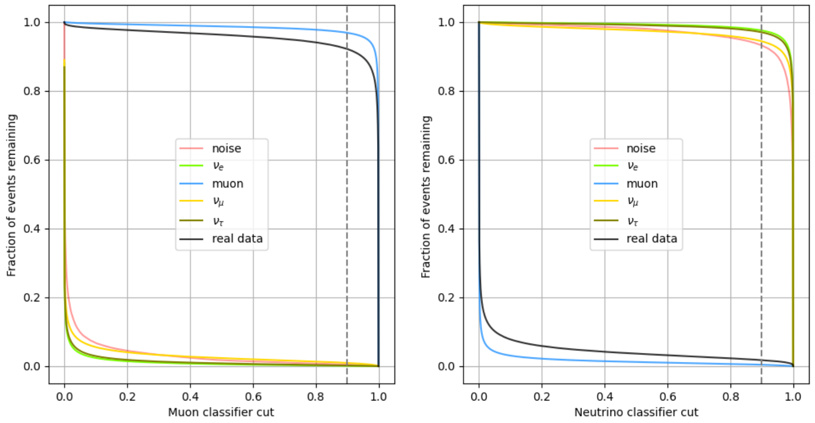


Figure 4.7: Muon and neutrino classifier cut performance on test sample of events (here events are unweighted) split by pid for SplitInIcePulses trained noise classifier. Performance is tracked by looking at the fraction of events that remain (w.r.t no cut) and the selected 0.9 cut used in this work is displayed in grey on the left plot.

Stopped-Through Classifier

The final classification undertaken is to determine if the muons left in our sample are stopped or not. This final stopped muon classifier is also trained on a training set of 5 million consisting solely of muons, and the results are presented in figures similar to the previous two classifiers.

Figure 4.10 shows the false positive rate (FPR) and true positive rate (TPR) of the classifier as a function of the true stopping position for L2+DC simulation with a stopped muon cut of just 0.6. This visualises which regions of the detector the model makes the least mistakes or is most accurate in its predictions. Intuitively, FPR is simply the probability of a false alarm, the probability that a through going muon was misclassified as stopped. TPR states the fraction of actual positives that were identified and predicted correctly. The results as displayed in 4.10 are logical in that they suggest the regions where FPR is greatest and TPR lowest are at the edges of the detector along the boundary of what it means to truly be stopped.

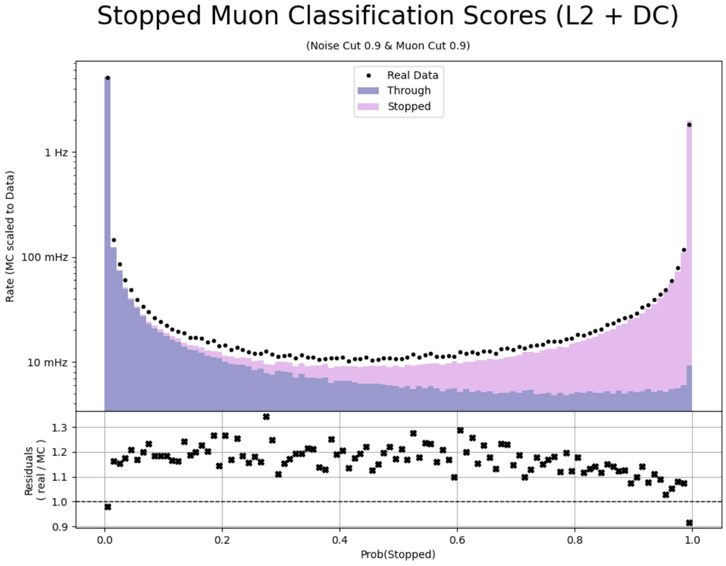


Figure 4.8: Stacked histogram of stopped muon classification score predicted by DynEdge on both simulation and real data. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected. Model trained on 5 million uncleaned (SplitInIcePulses) event pulsemaps and infer on 48 million test set.

The final cut we make on our data sample will be on the stopped muon score of this classifier. The effect of such a cut is shown in the final cut performance figure 4.11. For this classifier we choose a cut of 0.99. The rates

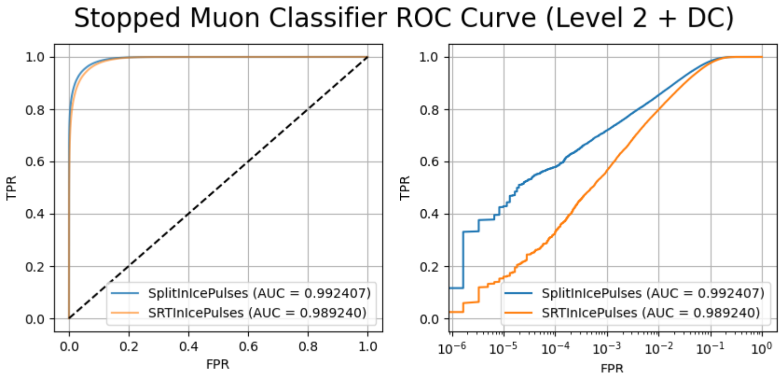


Figure 4.9: ROC curve for two stopped muon classifier models each trained on 5 million L2 + DC muon events using cleaned (SRTInIcePulses) and uncleaned (SplitInIcePulses) pulsemaps. ROC curve produced from the same validation set containing 1.6 million events.

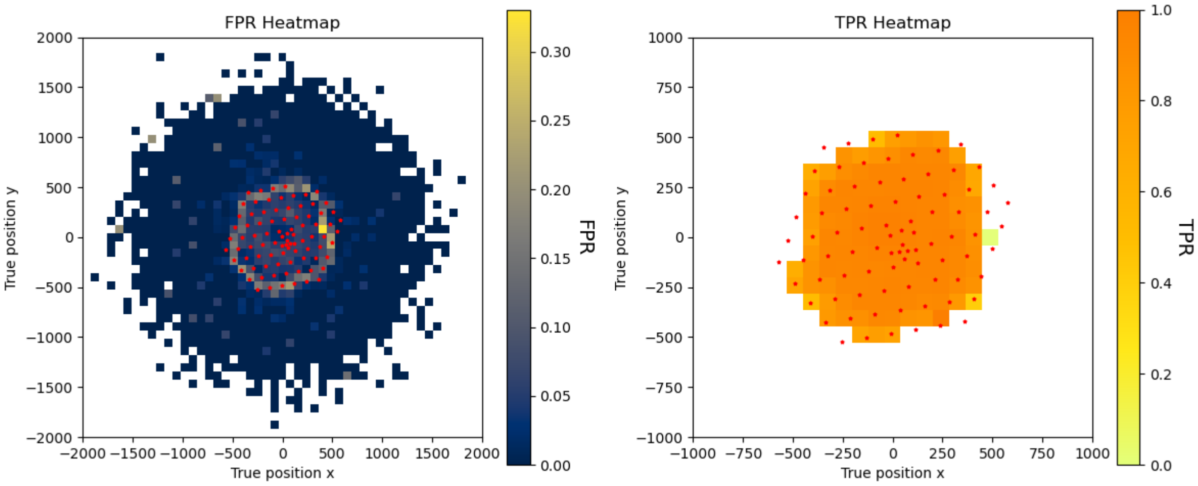


Figure 4.10: FPR and TPR binned by the true stopping point. Any bins with less than 50 events is masked. Since TPR requires a true positive case it only has values inside the fiducial volume of the detector.

for all particle types can be found in table 4.1. After the muon classifier cut the sample becomes overwhelmingly dominated by muons, though this is to be expected since the muon rate pre-selection is so high. The final stopped muon rate for L2 + DC simulation is 1.027 Hz, of which more than 99.5% are truly stopped. These stopped muon candidates, herein referred to as the stopped muon sample, will be the events chosen for further reconstruction and analysis in Section 4.3.

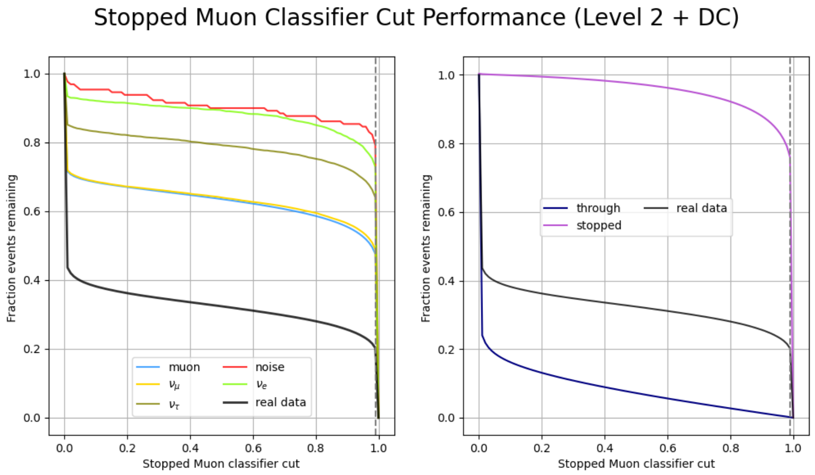


Figure 4.11: Stopped muon cut performance on test sample of events (here events are unweighted) split by pid and stopped/through for a SplitInIcePulses trained stopped muon classifier. Performance is tracked by looking at the fraction of events that remain, relative to the amount before the cut but after applying previous (noise, muon) cuts. The selected stopped muon cut 0.99 used in this work is displayed by the grey dashed line.

	L2 + DC	**Noise Cut**	**Muon Cut**	**Stopped Cut**	**Efficiency**
		(0.9)	(0.9)	(0.99)	(%)
ν_e	1.78 mHz	1.69 mHz	0.00167 mHz	0.00126 mHz	0.07%
ν_μ	7.00 mHz	6.77 mHz	0.0721 mHz	0.0381 mHz	0.5%
ν_τ	0.286 mHz	0.274 mHz	< 0.0002 mHz	-	< 0.05%
μ	4794 mHz	4779 mHz	4642 mHz	1027 mHz	21%
Noise	6621 mHz	8.28 mHz	0.0397 mHz	0.0341 mHz	< 0.0005%
Total	11420 mHz	4796 mHz	4642 mHz	1027 mHz	8.99%
Data	16443 mHz	9770 mHz	9008 mHz	1824 mHz	11.1%

Table 4.1: Event rates at level 2, with DeepCore filter applied and the subsequent rates as passed through the event selection chain. Rates calculated using oscNext weighting. Efficiency is relative to initial L2 + DC rate.

4.2 Classification Of Level 3 Data

Before continuing with the reconstruction of the L2+DC stopped muon sample it may be instructive to look at the performance and output of the classifiers using data that has passed the oscNext level 3 (L3) filter. There are several reasons why it is worthwhile to do this, first and foremost it is known that the data-simulation agreement at L3 is better than L2 and that the computational cost of filtering is low. Since muons are so abundant, they still dominate even at L3 allowing us to take advantage of natural muon abundance as at L2.

However, the drawback of this approach is the size of the real data sample passing L3 available to the author. Given the rate of events passing the L3 filter we begin the event selection process with just $54,170$ events from the 24 hours.

For brevity, in this section, we will show a subset of the same figures as shown in the previous Section 4.1 with discussion saved only for the notable points of difference between the two procedures. The same trained models will be used as before, though only classifying events that pass the L3 filter.

The initial noise classification on events passing the L3 filter is shown in figure 4.12. The stochasticity of the real data predictions is most likely due to the low number of events in the central, uncertain bins. Hence, small fluctuations of events in each bin lead to what appears to be a large difference between neighbouring bins. This problem is in part resolved by using coarser bins (see Appendix), though for consistency here we keep the number of bins to be 100.

The clear signal from the decision score histograms in 4.12,4.13,4.14 is that the model maintains its ability to separate noise, neutrinos and muons. It is also surprising that the data simulation agreement is comparable to that of the L2+DC events however this may simply be due to the lack of a large real data sample.

Differences between the two event selections are still present, however, figures 4.1 and 4.12 show that for L2 data there is a more smeared population of muon/neutrino events which is not present to the same extent in L3. Logically, we know that this is the removal of a number of muon events from L2+DC to L3 which leaves the muons and noise that are more difficult to classify/differentiate from particle events. The L3 filter removes the "easy" noise and muons and therefore leaves us only with events that display more neutrino-like properties.

In figures 4.5 and 4.13 the main discrepancy is in the leftermost bin where the model is most certain the event has been induced by a neutrino, and not a muon. In L2+DC there is a large discrepancy where the model projects that there are more than twice as many neutrino events in real data than in simulation. L3 smooths this disagreement and the left bin contains just 1.2

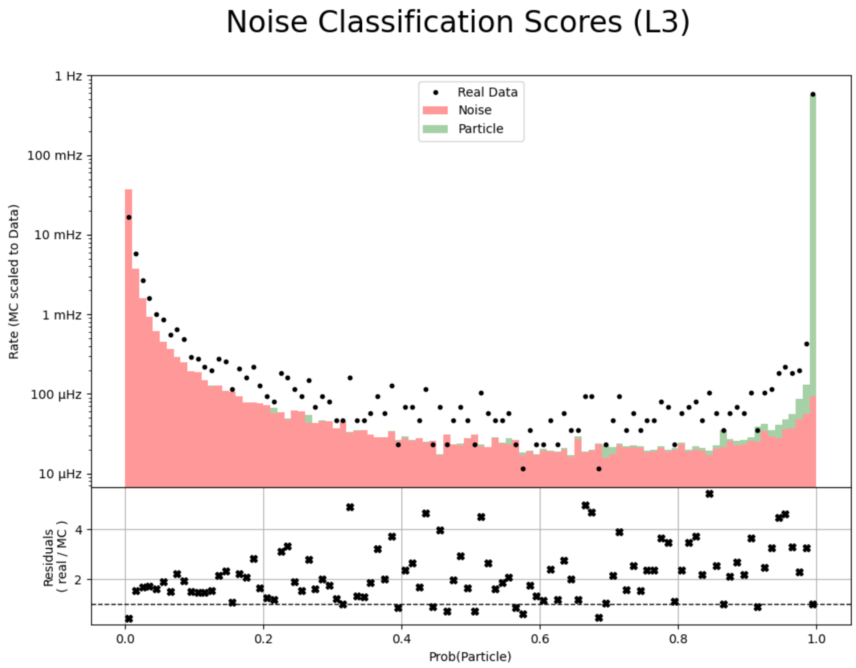


Figure 4.12: Stacked histogram of noise classification score predicted by DynEdge on both simulation and real data at event selection level 3. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected.

times as many neutrino events in the L3 sample. This suggests that events that do not pass the L3 filter are bamboozling the model and causing the overestimate at L2+DC.

The stopped muon classifier is least affected by the change from L2+DC to L3, except for the loss of an order of magnitude of the muons present. The overall characteristics displayed in 4.8 and 4.14 are similar, and the main differences are due to statistical fluctuations arising from the small number of events and the choice of binning in the histogram.

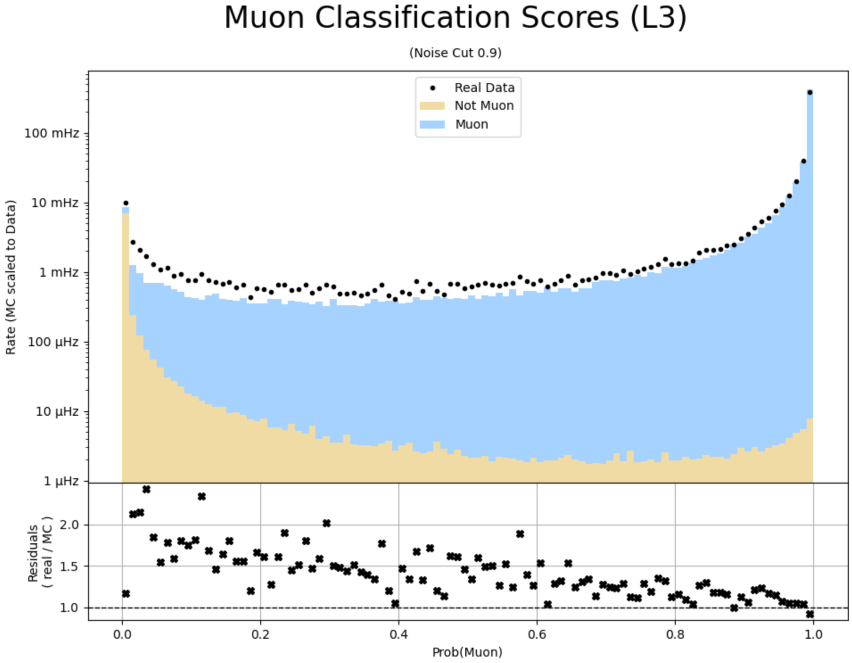


Figure 4.13: Stacked histogram of muon classification scores predicted by DynEdge on both simulation and real data at event selection level 3 and after a cut in noise prediction at 0.9. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected.

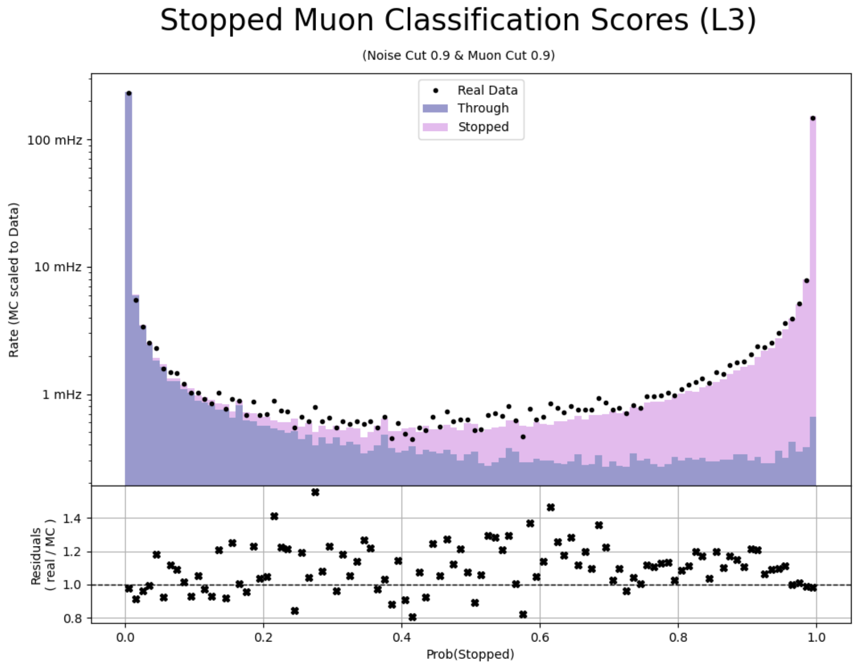


Figure 4.14: Stacked histogram of stopped muon classification scores predicted by DynEdge on both simulation and real data at event selection level 3. After a cut in noise prediction at 0.9 and muon prediction at 0.9. Rates calculated using oscNext flux weights, and MC scaled to reflect the rate of real data collected.

4.3 Reconstructing Stopped Muon Sample

Following the application of the classifiers and their respective cuts and having expressed the rates for the final sample in Table 4.1 we now proceed to the reconstruction of the stopped muon sample at L2+DC. The simulated noise events are nearly completely eliminated after the noise and muon cuts leaving the sample overwhelmingly dominated by muons. This is equally true in real data, where the immense majority of events are muons after removing noise, even before the muon cut is applied. One advantage of the stopped muon sample used for this analysis is the natural abundance of muon events at IceCube.

The structure of this section is as follows; first we display the regression results output from DynEdge trained models for energy, zenith, azimuth and stopping position. Each regression variable will be presented with a figure detailing the accuracy of DynEdge predictions on the simulated stopped muon sample by looking at the prediction residuals. A second figure will show a basic histogram of the results as well as a 2d histogram plotting the simulation predictions against truth. Finally, we will compare the simulated histogram to that predicted by the model for real data. See Appendix for histogram of regression outputs with logarithmic scales and unweighted scatterplots.

Following the regression results we examine some of the key features of each event. These do not require reconstruction as we can engineer the detector output pulsemap to retrieve this information. Looking at these distributions together with our reconstructed values provides us with another avenue through which to compare real data and simulation.

4.3.1 Energy Regression Results

The distribution of energy predictions and truths for the simulated stopped muon sample is shown in figure 4.15. We see good agreement in the densely populated bins but also long tails in truth that are less well reconstructed. In figure 4.16 we quantify the difference between true and predicted energy by looking at the residuals, $true - predicted$, for events binned in their true energy. We see clearly that the model predictions are best in the central bins with the most events.

The discrepancy in the tails is partly explained by the generation of MuonGun simulation. The minimum energy muon produced for our training set was 150 GeV, which means that our model has never seen events below this energy. When we come to reconstruct the stopped muon sample, some low energy neutrino or noise events may confound our model's predictions.

Finally, when comparing to the real data stopped muon sample in 4.17 we see excellent agreement. DynEdge predictions in simulation match those of real data very closely.

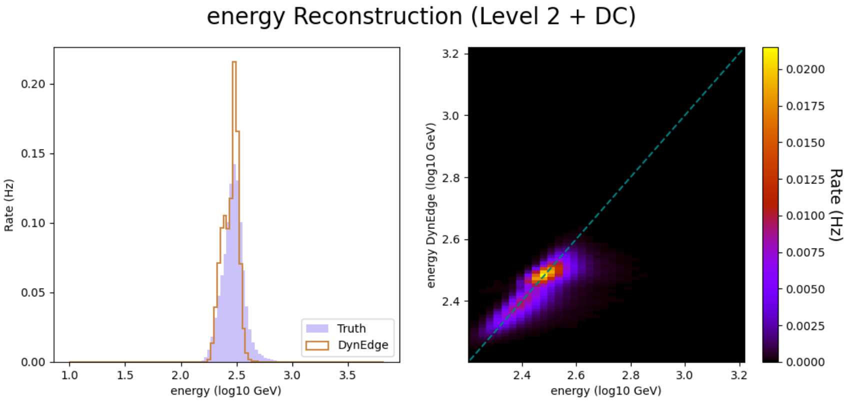


Figure 4.15: A) Histogram of DynEdge energy predictions and base truth for the stopped muon sample. B) 2d histogram of true energy against DynEdge predictions. A perfect classifier is represented with the dashed line.

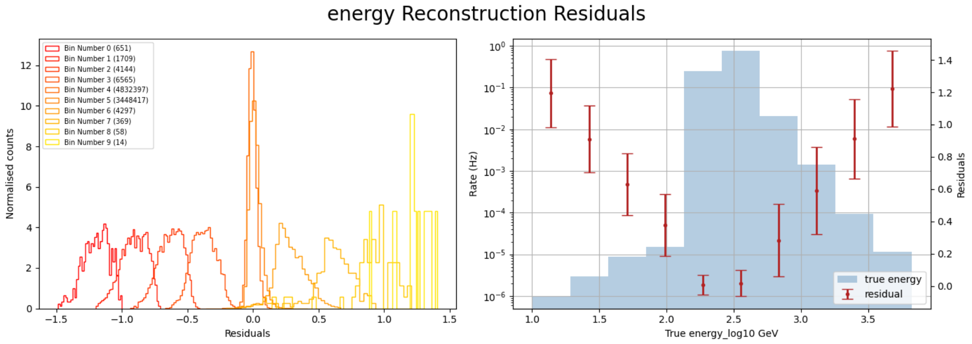


Figure 4.16: A) Illustration of the calculation that produces residual plots in this work. All events in each bin have their residual calculated, from this we take the mean and the confidence interval (difference between the 84th and 16th percentile) for a presumed normal distribution and plot this for each bin of events. Number of events in each is shown in brackets for context. B) The average residual (truth-energy) for events in each true energy bin. Overlaid is the event population binned in their true energy.

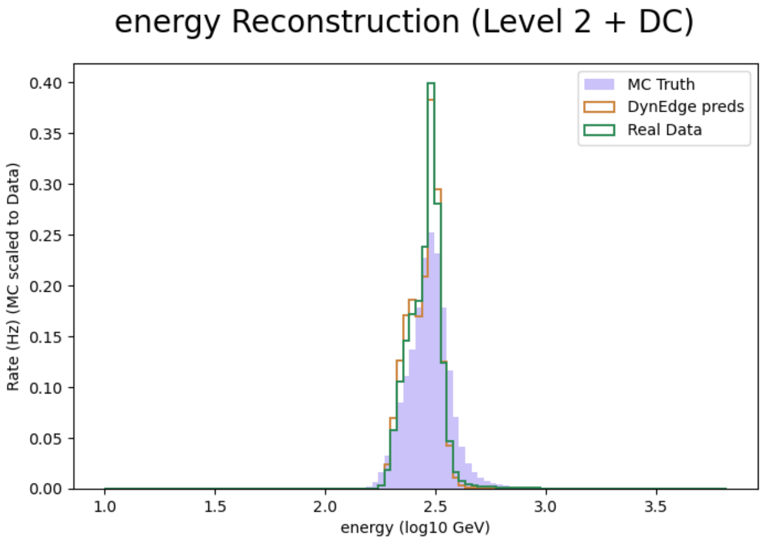


Figure 4.17: Histogram of DynEdge energy predictions for stopped muon sample in both simulation and real data. Simulation scaled to match rate of real data. MC truth provided by IceCube simulation (MuonGun).

4.3.2 Zenith Regression Results

The distribution of zenith predictions and truths for the stopped muon sample is shown in Figure 4.18. As with energy we see good agreement in the densely populated bins and the distribution produced has the right shape. This is also true in Figure 4.19 where we see clearly the model predictions are worse in the bins where there are few events in the stopped sample.

The zenith distribution in real data (4.20) follows very closely our model predictions for the simulated sample.

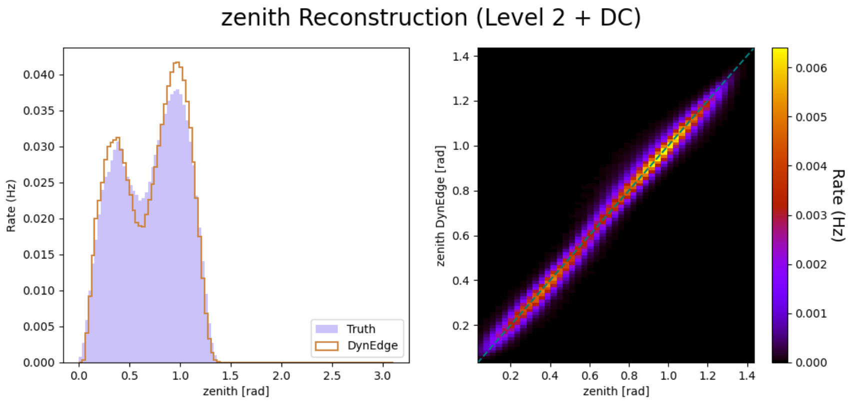


Figure 4.18: A) Histogram of DynEdge zenith predictions and base truth for the stopped muon sample. B) 2d histogram of true zenith against DynEdge predictions. A perfect classifier is represented with the dashed line.

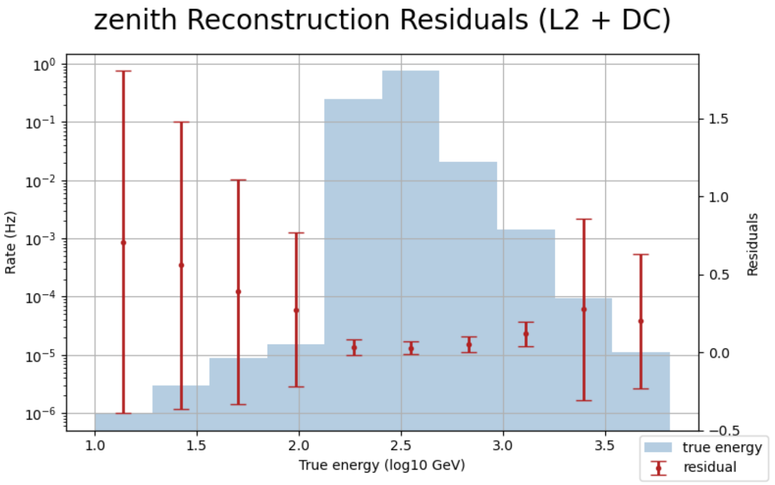


Figure 4.19: Average residual of zenith predictions for events binned based on true energy. Overlaid is the event population in the same binning.

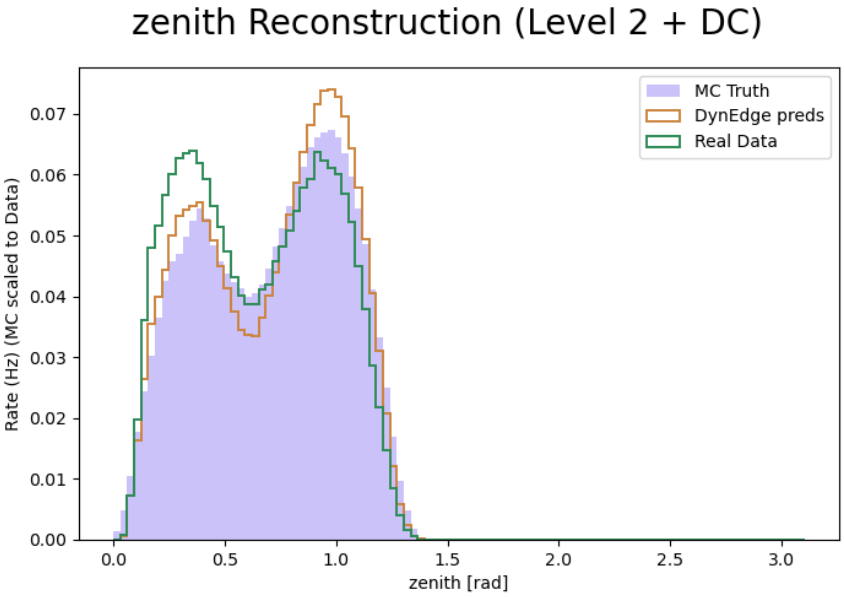


Figure 4.20: Histogram of DynEdge zenith predictions for the stopped muon sample in both simulation and real data. Simulation scaled to match real data rate. Model trained on 5 million muon events (MuonGun).

4.3.3 Azimuth Regression Results

The distribution of azimuth predictions and truths for the stopped muon sample is shown in Figure 4.21. The 2d histogram shows excellent overall agreement that follows the peaks and troughs of simulated data very well. Unsurprisingly, as with both energy and zenith we see good agreement in the densely populated bins and the distribution produced has the right shape. This is also true in Figure 4.22 where we see clearly the model predictions are worse in the bins where there are few events in the stopped sample.

The azimuth distribution in real data (4.23) also matches the simulated distribution well, mirroring the various peak and trough characteristics between 0 and 2π.

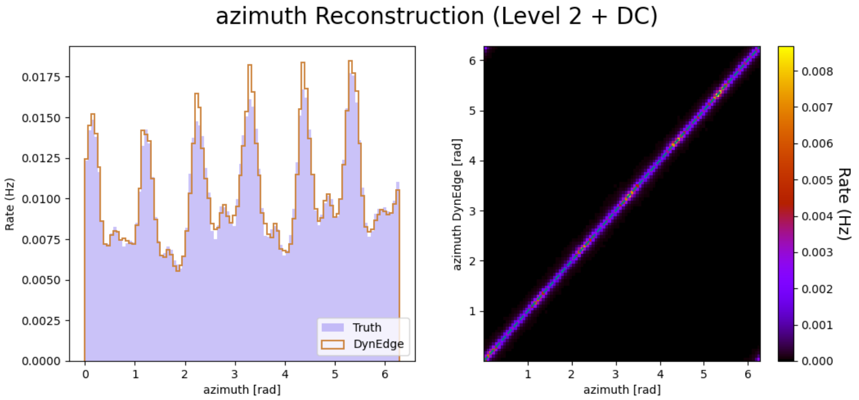


Figure 4.21: A) Histogram of DynEdge azimuth predictions and base truth for the stopped muon sample. B) 2d histogram of true azimuth against DynEdge predictions. A perfect classifier is represented with the dashed line.

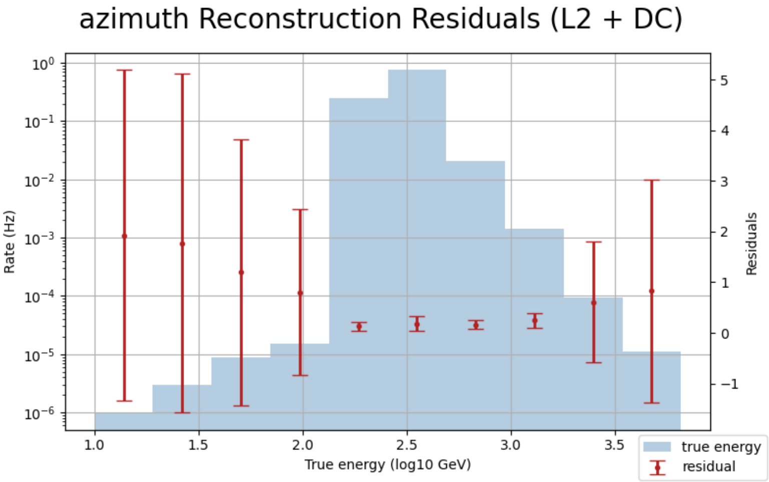


Figure 4.22: Average residual of azimuth predictions for events binned based on true energy. Overlaid is the event population in the same binning.

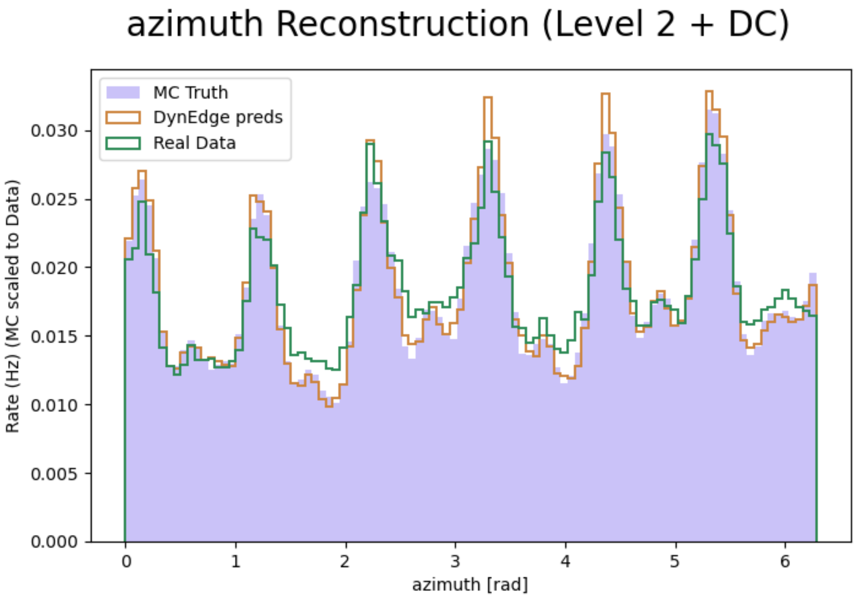


Figure 4.23: Histogram of DynEdge azimuth predictions for the stopped muon sample in both simulation and real data. Simulation scaled to match real data rate. Model trained on 5 million L2+DC muon events (MuonGun).

4.3.4 Stopping Position Regression Results

For stopping position we display all three coordinates in Figure 4.24. The simulation residuals in all three coordinates are worst in the low energy bins where events are unlike those the model has trained on. The real data simulation also strongly agrees in the central bins. That said, all the real data predictions seem to have leakage in long tails, none more so than the z position prediction which sees a relatively large population of the sample have stopping points less deep in the detector.

In Figure 4.25 we plot the real data predictions, simulation predictions and simulation truths in 2d histograms binned in the $x - y$ plane on the left and the $x - z$ plane on the right. This makes clear what the histograms allude to, real data predictions span a greater variety of depths while exhibiting much of the same behaviour in $x - y$, where predictions seems to be concentrated more closely to the string locations than is the case in simulation truth.

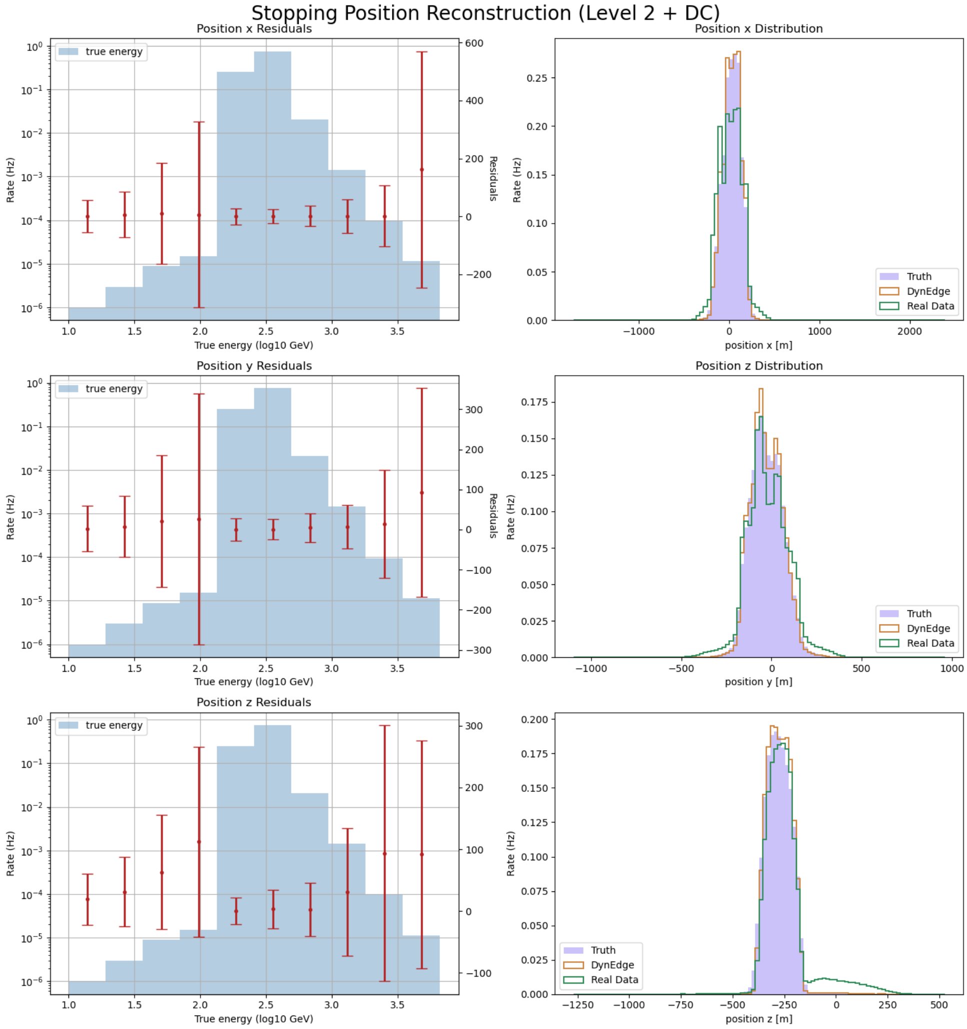


Figure 4.24: Average residual plots for the three coordinates of stopping point x, y, z regression for events binned in true energy. Overlaid is the event population of the same binning. In training the model did not see muons below 150GeV.

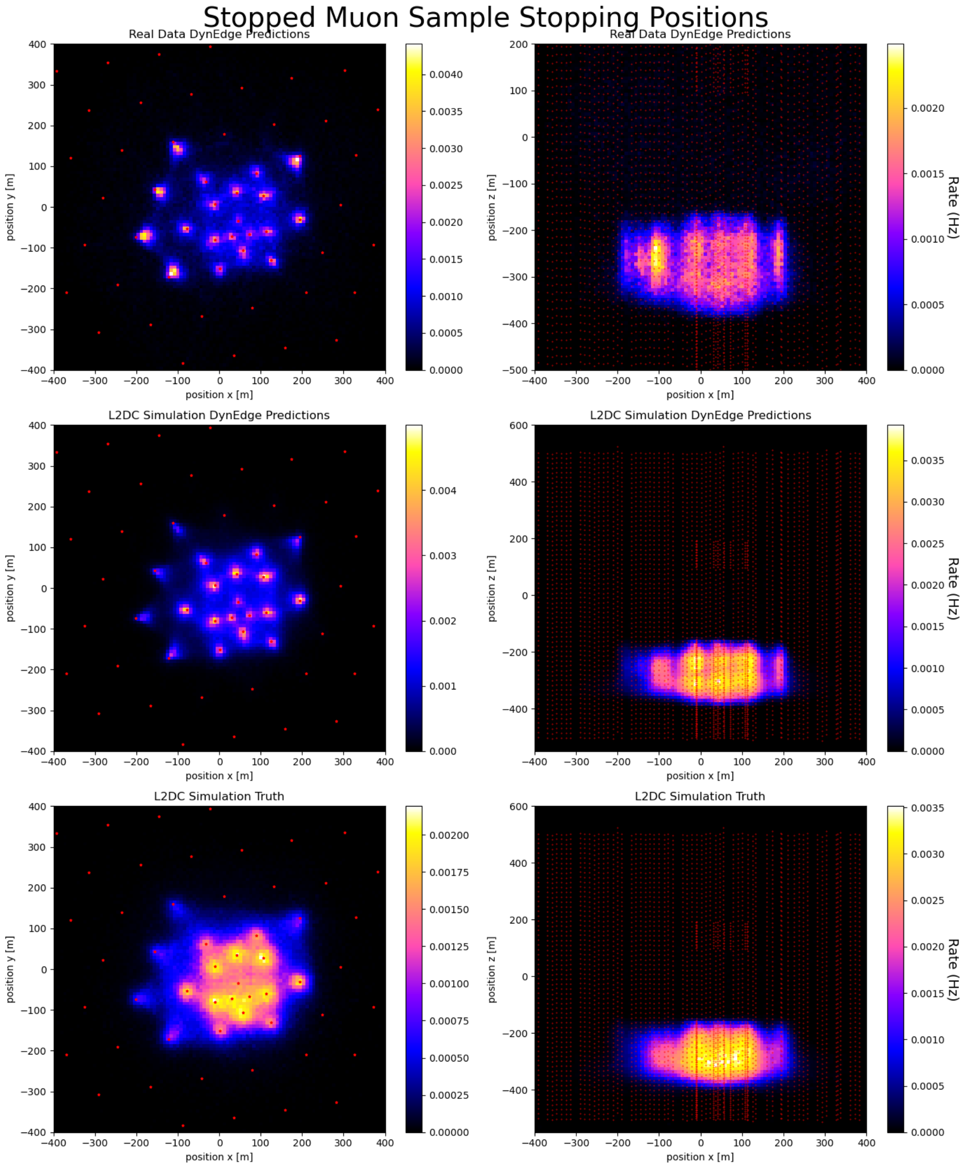


Figure 4.25: DynEdge predictions of stopping point x, y, z expressed in 2 dimensions for both real data and simulation, as well as simulation truth. Events weighted using oscNext flux weights. IceCube strings and DOMs overlaid in red.

4.3.5 Feature Summaries

Another program of comparison between the simulated and real stopped muon sample is through the characteristics and features of the event pulsemaps themselves. Working with uncleaned pulsemaps we may ascertain summary statistics from each of the events in the stopped muon sample including, the number of pulses, number of DOMs triggered, number of unique strings with triggered DOMs, the average number of pulses per DOM, maximum number of pulses recorded on a single DOM, the duration of the event in seconds and the location of the last DOM triggered. The distributions of these survey variables are shown in figure 4.26 for both simulation and real data. Real data looks to have much longer tails in n_pulses, maximum number of pulses on any DOM and event duration. In general, however, there is good agreement between the features of the two samples.

Many of these summary statistics are highly correlated with the event truth variables we have reconstructed thus far. Clear cut examples of these correlations are the numbers of pulses and DOMs in an event and their dependence on the energy of the event; a representative pair plot displaying the relationships between 3 of the feature summary columns and 5 predicted/true variables.

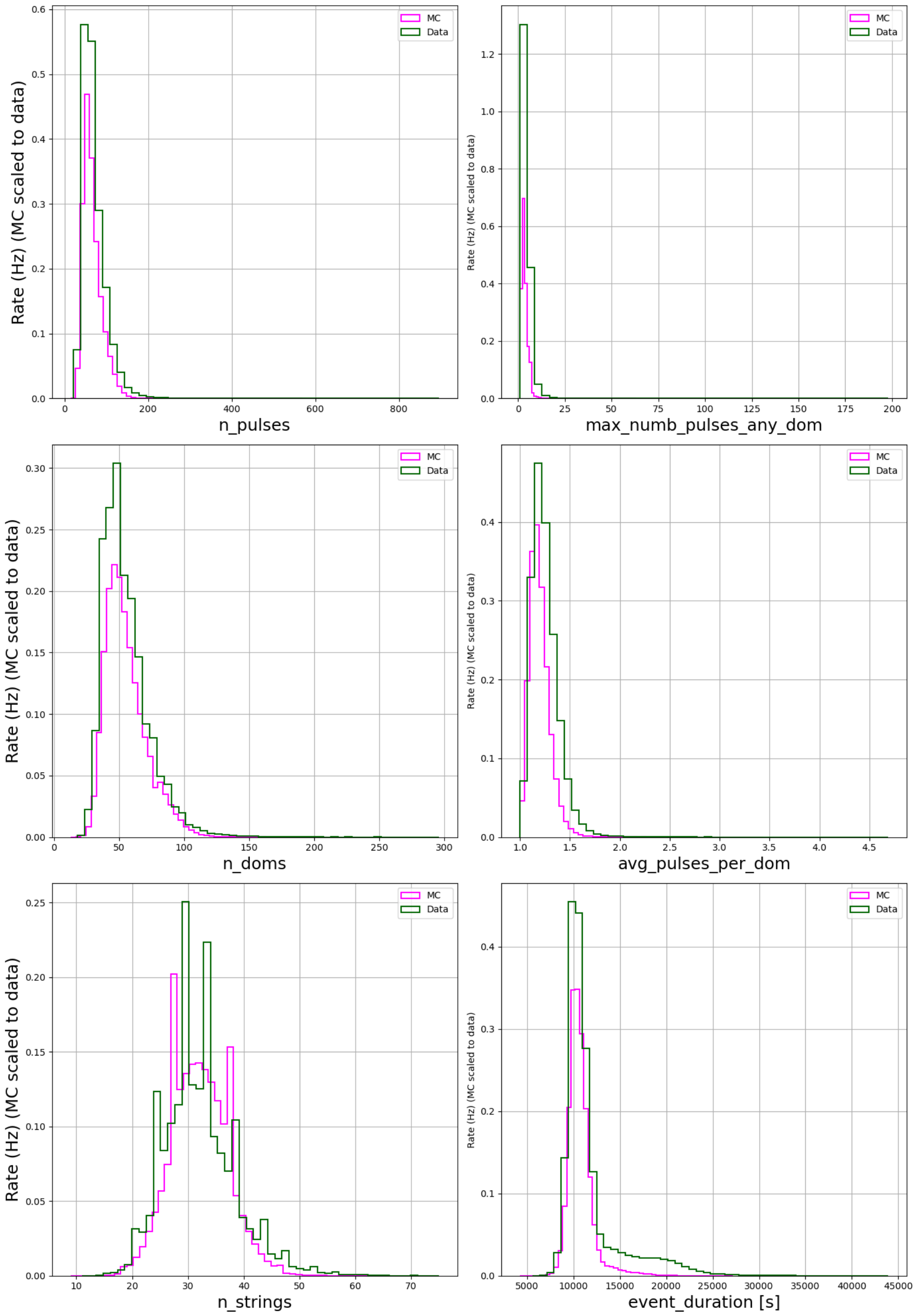


Figure 4.26: Histograms of event features: the number of pulses in the event, the number of triggered DOMs, the number of unique strings triggered, the maximum number of pulses on a DOM in the event, the average number of pulses per DOM and the event duration.

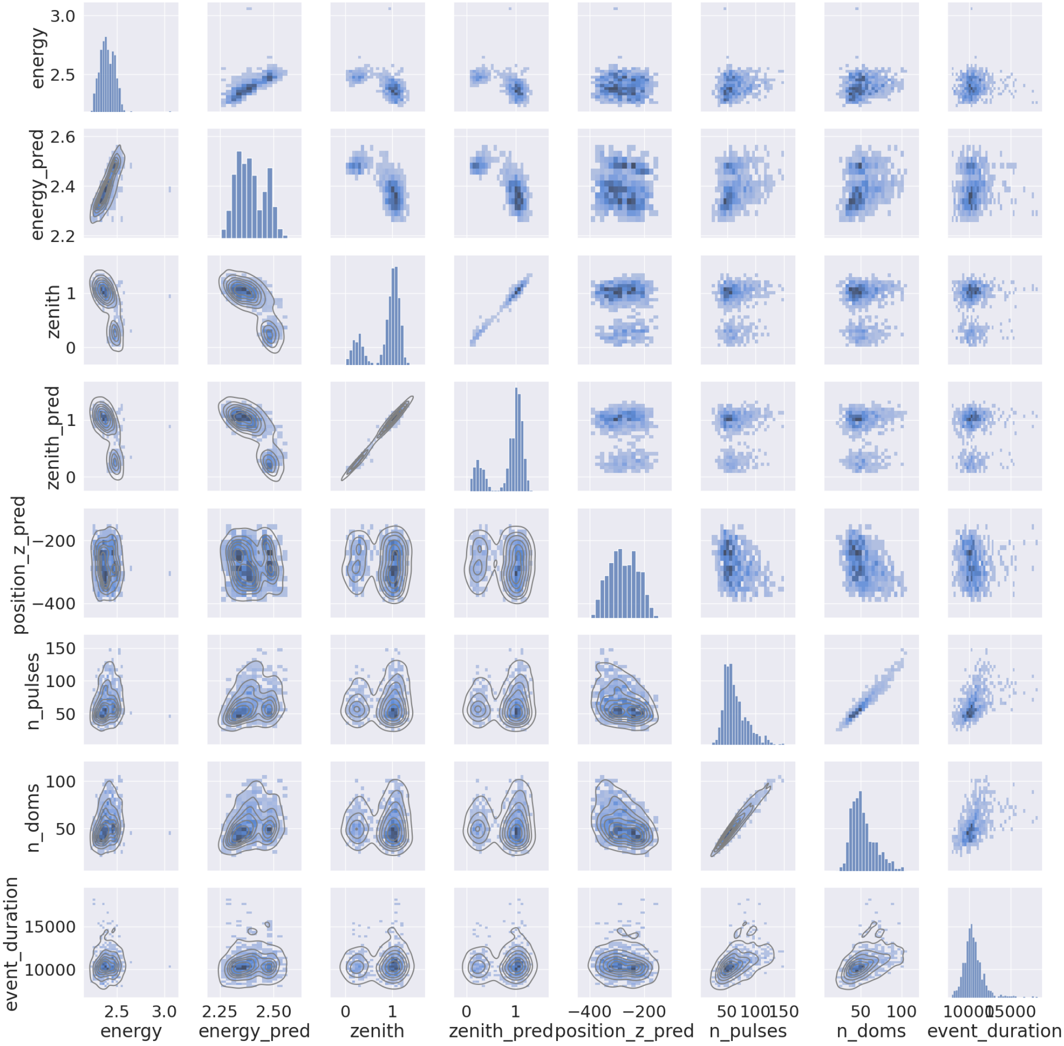


Figure 4.27: A pairplot showing a random 1% sample of the stopped muon sample showing the pairwise 2d histograms of a number of reconstructed variables and feature summaries. The order of the plot is true energy, energy prediction, true zenith, zenith prediction, stopping depth position, number of pulses, number of DOMs and event duration. Approximate KDEs overlaid symmetric 2d histograms in lower triangular half.

5

Conclusion and Outlook

It has been shown in this thesis that the new machine learning paradigm, in particular the use of GNNs, not only works on processed, specially selected IceCube neutrino events but also on low-level noisy data with high variability. The methods presented in this work have potential not just as a tool for fast, accurate regression and reconstruction but also as a tool to separate and categorize the vast amount of background that is a principal challenge for IceCube. This scrutiny cannot come at a better time, with a planned upgraded detector bringing additional noise and complexity with the enhanced sensitivity.

We saw in the classification of L2+DC data that DynEdge still has the capacity to divide noise from particle events, and muons from neutrinos. This has the potential to bring about massive speed up for IceCube's numerous working groups, all of whom share access to the base L2 data. Furthermore, the proficiency of DynEdge is not limited to noise classification, as displayed in the partitioned muon and neutrino populations as well as the stopped muon classifier.

The second aim of this work was to compare and contrast a population of stopped muons in both simulation and real data. Not only was the model able to discriminate the stopped muons well, but it was also able to reconstruct the energy, zenith, azimuth and stopping position with impressive accuracy. The opportunity that this work represents is that muons need not be background. The speed with which we can not only process events but also reconstruct variables of interest allows all manner of analyses to be performed that are otherwise precluded on account of computational cost.

While the figures in Section 4.3 report good agreement between data and simulation and favourable regression residuals there are certainly improvements to be attained. The most glaring is the lack of low energy muon simulation. The current MuonGun i3-files contain muons with a minimum generated energy of 150 GeV. This leaves the model susceptible to real data events that have less than this minimum, this is reflected in the residual error bars for the regression variables when binned in energy. This examination of real data predictions convinces us that the GNN can and will work well when applied more widely.

Additional choices made by MuonGun during generation may limit the data simulation agreement. One example is that there is a lack of coincident muon events (muon bundles) in simulation since MuonGun simulates single muons intersecting the DeepCore fiducial volume. Adding muons with multiplicity greater than 1 may improve the applicability to real data. Another strategy for improvement could be switching MuonGun simulation with the more complete CORSIKA simulation.

Finally, looking forward there are several immediate tasks that one could pursue chief among them being a repeat of the classification, regression performed in this work but applied to the different systematic uncertainty simulation sets. Using a large enough stopped muon sample to probe the systematic uncertainty parameter space is a clear first step. Looking further forward one could envisage predicting the systematic parameters for which stopped muons match best in data and simulation rather than just determining which of the existing systematic combinations are best.

The last and most ambitious assignment is the application of DynEdge to all 11 years of fully-operational IceCube and DeepCore data. This way one would have a sufficient number of neutrino events with which to do a data simulation comparison or more pertinently perform a neutrino oscillation analysis with more neutrino events collected and better reconstruction one could compare the number (and energy and direction) of the neutrinos found by DynEdge with the expected flux before fitting according to free parameters that account for the oscillation and nuisance parameters

A

Appendix

Train and Test Sets

Training for all classifiers and regression models occurred over 5 million LE simulated events. The relevant COBALT files are as follows:

- `/data/ana/LE/oscNext/pass2/muongun/level3/130000`
- `/data/ana/LE/oscNext/pass2/genie/level3/1X000` for $X = 2, 4, 6$ for each neutrino flavour.
- `/data/ana/LE/oscNext/pass2/noise/level3/888003`

Moreover, the test set ~ 48 million events are taken from the same source. The real data from 23^{rd} January 2019 was retrieved from three 8 hour sub-runs:

- `oscNext_data_IC86.19_level3_v02.00_pass2_Run00133652`
- `oscNext_data_IC86.19_level3_v02.00_pass2_Run00133653`
- `oscNext_data_IC86.19_level3_v02.00_pass2_Run00133654`

Supplementary Figures

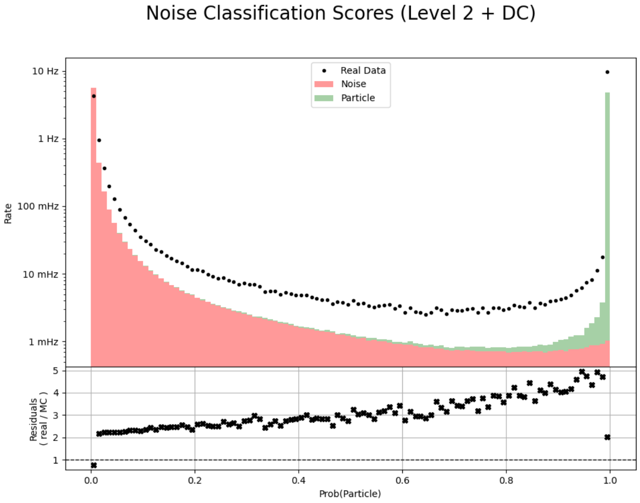


Figure A.1: Stacked histogram of particle probabilities predicted by DynEdge on both simulation and real data L2+DC. Rates calculated using oscNext flux weights. This histogram was produced using a model trained on a balanced set of 5 million events and inferring on 48 million events using uncleaned (SplitInIcePulses) pulsemap.

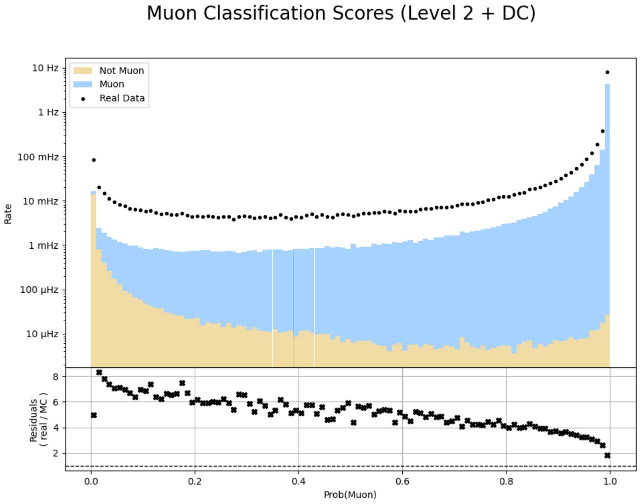


Figure A.2: Stacked histogram of neutrino probability score predicted by DynEdge on simulation and real data at L2+DC that passes noise cut. Rates calculated using oscNext flux weights. Model trained on, uncleaned (SplitInIcePulses) balanced 5 million event sample, and inferred on 48 million test set.

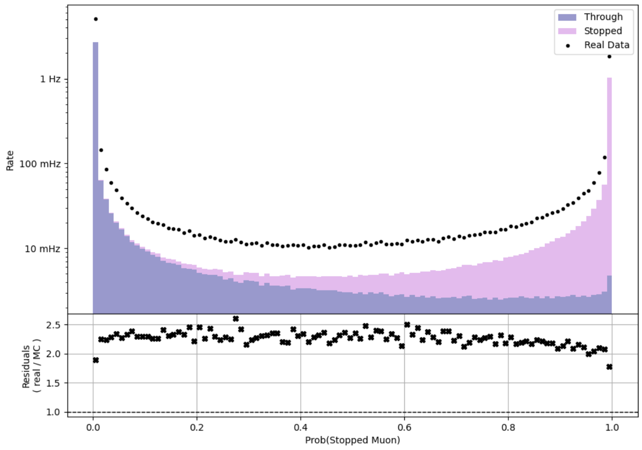


Figure A.3: Stacked histogram of stopped muon classification score predicted by DynEdge on both simulation and real data. Rates calculated using oscNext flux weights. Model trained on 5 million uncleaned (SplitInIcePulses) event pulsemaps and infer on 48 million test set.

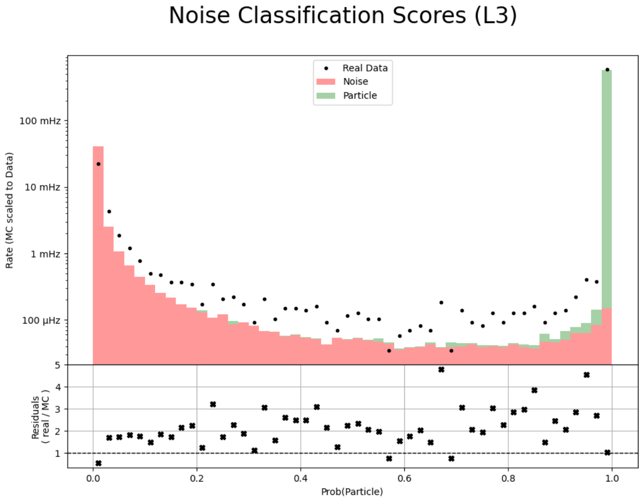

Noise Classification Scores (L3)

Figure A.4: Stacked histogram of noise classification score predicted by DynEdge on both simulation and real data at event selection level 3. Rates calculated using oscNext flux weights.

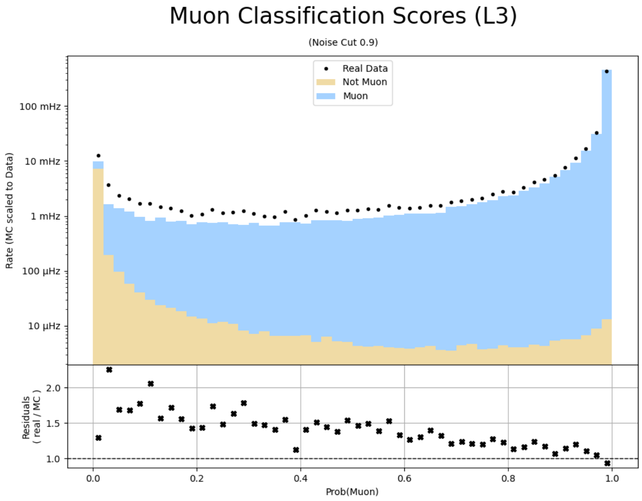


Figure A.5: Stacked histogram of muon classification scores predicted by DynEdge on both simulation and real data at event selection level 3 and after a cut in noise prediction at 0.9. Rates calculated using oscNext flux weights.

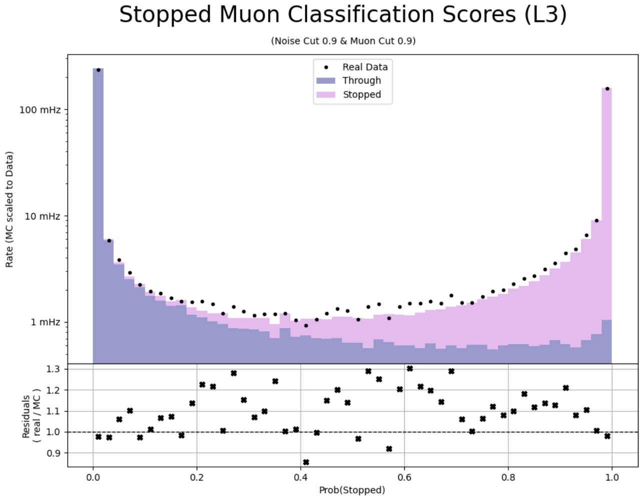


Figure A.6: Stacked histogram of stopped muon classification scores predicted by DynEdge on both simulation and real data at event selection level 3. After a cut in noise prediction at 0.9 and muon prediction at 0.9. Rates calculated using oscNext flux weights

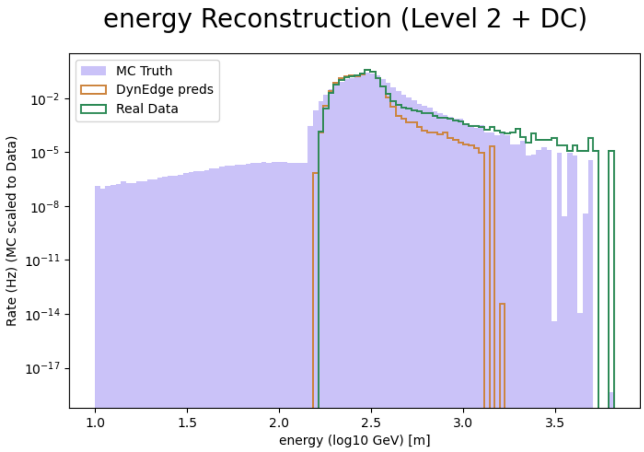


Figure A.7: Histogram of DynEdge energy predictions for stopped muon sample in both simulation and real data on log scale. Simulation scaled to match rate of real data. MC truth provided by IceCube simulation.

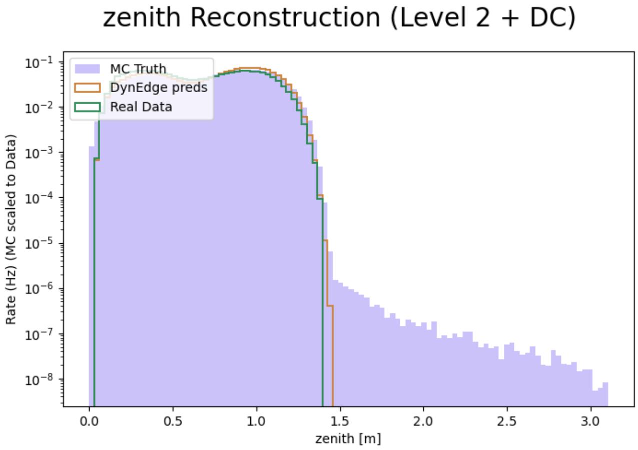


Figure A.8: Histogram of DynEdge zenith predictions for the stopped muon sample in both simulation and real data on logarithmic scale. Simulation scaled to match real data rate. Model trained on 5 million muon events (MuonGun).

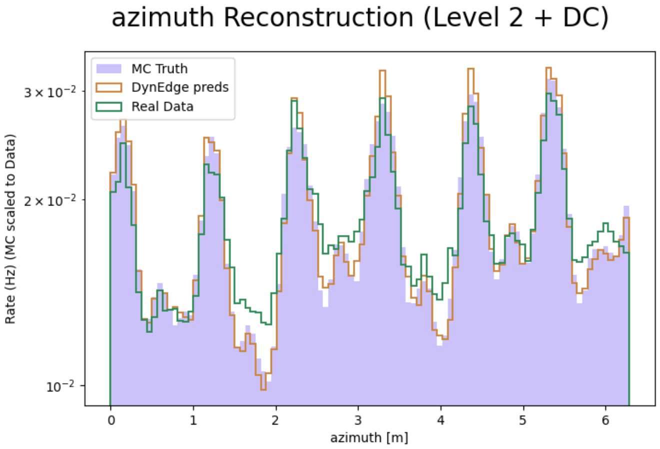


Figure A.9: Histogram of DynEdge azimuth predictions for the stopped muon sample in both simulation and real data log scale. Simulation scaled to match real data rate. Model trained on 5 million L2+DC muon events (MuonGun).

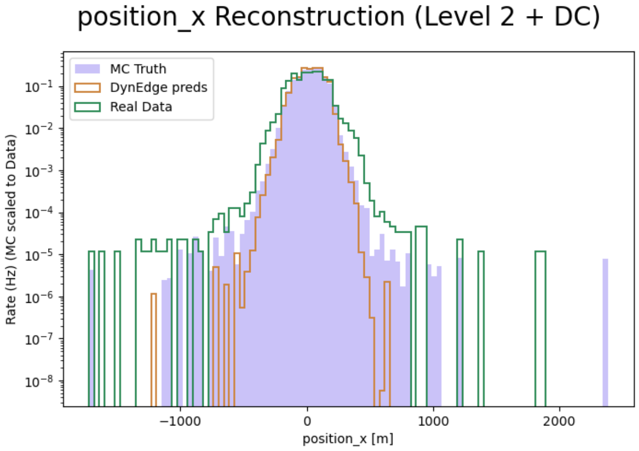


Figure A.10: Histogram of DynEdge position x predictions for the stopped muon sample in both simulation and real data log scale. Simulation scaled to match real data rate. Model trained on 5 million L2+DC muon events.

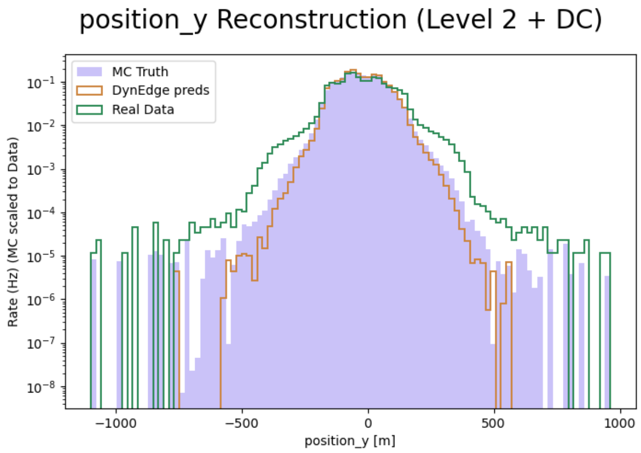


Figure A.11: Histogram of DynEdge position y predictions for the stopped muon sample in both simulation and real data log scale. Simulation scaled to match real data rate. Model trained on 5 million L2+DC muon events.

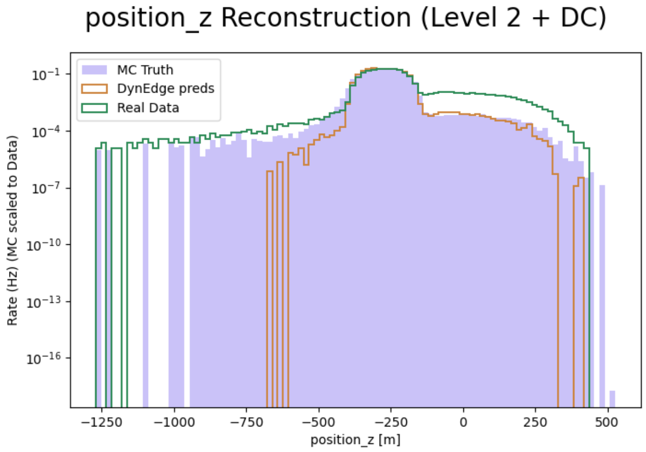


Figure A.12: Histogram of DynEdge position z predictions for the stopped muon sample in both simulation and real data log scale. Simulation scaled to match real data rate. Model trained on 5 million L2+DC muon events.

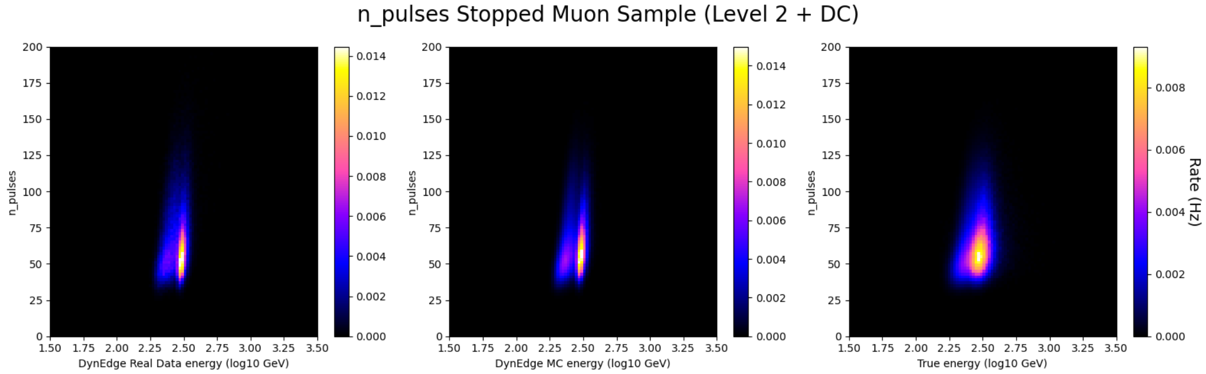


Figure A.13: 2d histograms showing the observed number of pulses against the energy prediction and truth.

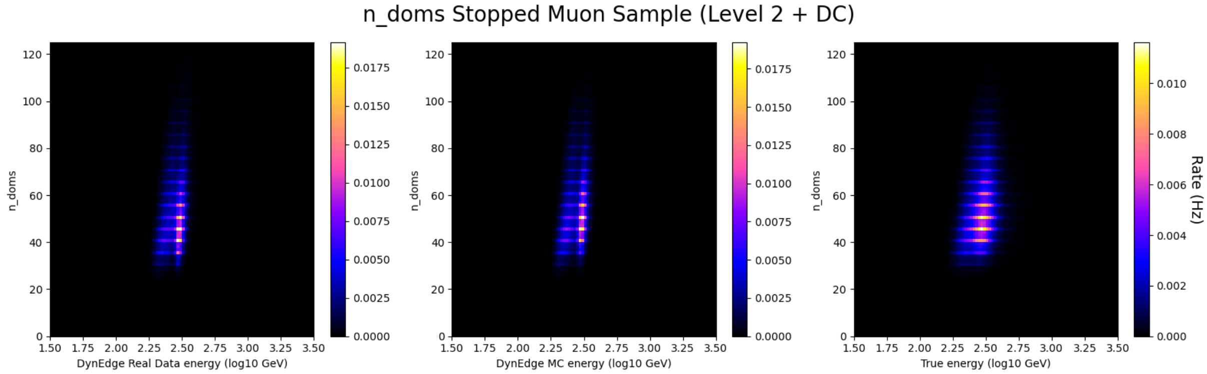


Figure A.14: 2d histograms showing the observed number of unique DOMs triggered in the event against the energy prediction and truth.

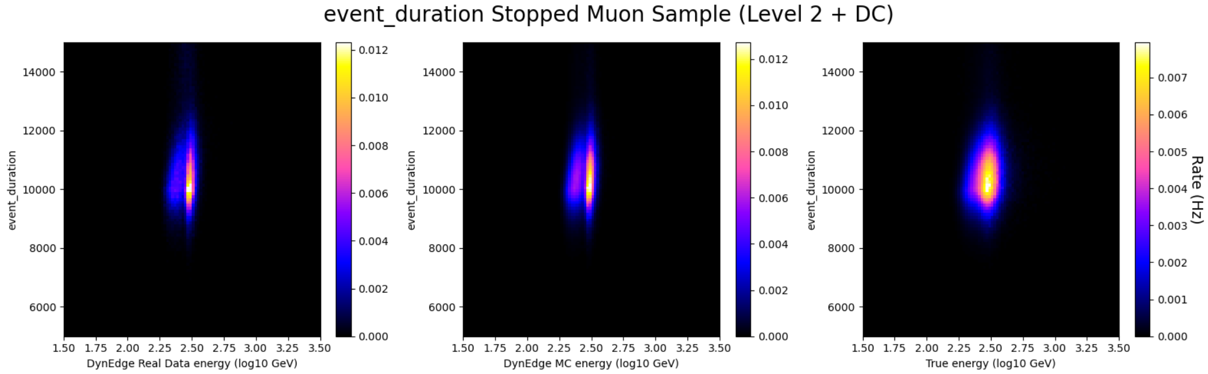


Figure A.15: 2d histograms showing the event duration triggered in the event against the energy prediction and truth.

Bibliography

[1] Carsten Burgard. *Example: Standard model of physics.* Last accessed May 2022. 2016. URL: https://texample.net/tikz/examples/model-physics/ (cit. on p. 3).

[2] P.A. Zyla et al. “Review of Particle Physics”. In: *PTEP* 2020.8 (2020). and 2021 update, p. 083C01. DOI: 10.1093/ptep/ptaa104 (cit. on pp. 3, 6, 8, 20, 21).

[3] Sérgio Ferraz Novaes. “Standard model: An introduction”. In: *arXiv preprint hep-ph/0001283* (2000) (cit. on p. 3).

[4] E. Noether. “Invariante Variationsprobleme”. ger. In: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse* 1918 (1918), pp. 235–257. URL: http://eudml.org/doc/59024 (cit. on p. 4).

[5] Eite Tiesinga et al. “CODATA recommended values of the fundamental physical constants: 2018”. In: *Journal of Physical and Chemical Reference Data* 50.3 (2021), p. 033105 (cit. on p. 5).

[6] ME Peskin and DV Schroeder. “An introduction to quantum field theory Westview Press”. In: *Boulder, Colorado* (1995) (cit. on p. 5).

[7] Suzanne Van Dam. “Spontaneous symmetry breaking in the Higgs mechanism”. In: (2011) (cit. on p. 5).

[8] Mark Thomson. *Modern particle physics.* Cambridge University Press, 2013 (cit. on p. 6).

[9] Marco Battaglia et al. “The CKM matrix and the unitarity triangle”. In: *arXiv preprint hep-ph/0304132* (2003) (cit. on p. 6).

[10] Wikimedia Foundation. *https://en.wikipedia.org/wiki/File:Beta*$_N$*egative*$_D$*ecay.svg.* URL: https://en.wikipedia.org/wiki/Beta_decay (visited on 05/01/2022) (cit. on p. 7).

[11] George B Rybicki and Alan P Lightman. *Radiative processes in astrophysics.* John Wiley & Sons, 1991 (cit. on p. 8).

[12] Walter Heitler. *The quantum theory of radiation.* Courier Corporation, 1984 (cit. on p. 8).

[13] SM Bilenky. “The history of neutrino oscillations”. In: *Phys. Scripta* 121 (2005), p. 17 (cit. on p. 9).

[14] Kamiokande Collaboration et al. “Evidence for oscillation of atmospheric neutrinos”. In: *arXiv preprint hep-ex/9807003* (1998) (cit. on p. 9).

[15] SnO Collaboration et al. “Measurement of the Rate of nu_e+ d–> p+ p+ e^-Interactions Produced by 8B Solar Neutrinos at the Sudbury Neutrino Observatory”. In: *arXiv preprint nucl-ex/0106015* (2001) (cit. on p. 10).

[16] T Araki et al. “Measurement of neutrino oscillation with KamLAND: Evidence of spectral distortion”. In: *Physical Review Letters* 94.8 (2005), p. 081801 (cit. on p. 10).

[17] Takaaki Kajita. “ATMOSPHERIC NEUTRINOS AND DISCOVERY OF NEUTRINO OSCILLATIONS”. In: *Proc. Japan Acad. B* 86 (2010), pp. 303–321. DOI: 10.2183/pjab.86.303 (cit. on pp. 10, 19).

[18] Ziro Maki, Masami Nakagawa, and Shoichi Sakata. “Remarks on the unified model of elementary particles”. In: *Progress of Theoretical Physics* 28 (1962), pp. 870–880 (cit. on p. 10).

[19] Claudio Giganti, Stéphane Lavignac, and Marco Zito. “Neutrino oscillations: the rise of the PMNS paradigm”. In: *Progress in Particle and Nuclear Physics* 98 (2018), pp. 1–54 (cit. on pp. 10, 11).

[20] S. Westerdale. “Neutrino Mass Problem : Masses and Oscillations”. In: 2010 (cit. on p. 10).

[21] 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), pp. 1–22 (cit. on p. 13).

[22] Erica Smith. “Neutrino Oscillations”. In: 2010 (cit. on p. 13).

[23] Joshua P Ellis. “Tikz-feynman: Feynman diagrams with tikz”. In: *Computer Physics Communications* 210 (2017), pp. 103–123 (cit. on p. 14).

[24] H Gallagher, G Garvey, and GP Zeller. “Neutrino-nucleus interactions”. In: *Annual Review of Nuclear and Particle Science* 61 (2011), pp. 355–378 (cit. on p. 14).

[25] T Leitner, L Alvarez-Ruso, and U Mosel. “Neutral current neutrino-nucleus interactions at intermediate energies”. In: *Physical Review C* 74.6 (2006), p. 065502 (cit. on p. 14).

[26] L Alvarez-Ruso, T Leitner, and U Mosel. “Neutrino interactions with nucleons and nuclei at intermediate energies”. In: *AIP Conference Proceedings*. Vol. 842. 1. American Institute of Physics. 2006, pp. 877–879 (cit. on p. 15).

[27] Tina J. Leitner. “Neutrino Interactions with Nucleons and Nuclei”. PhD thesis. Institut f ́ıur Theoretische Physik Justus-Liebig-Universitat GieSSen, 2005 (cit. on p. 16).

[28] P Kyberd et al. “nuSTORM: neutrinos from STORed Muons”. In: *arXiv preprint arXiv:1206.0294* (2012) (cit. on p. 17).

[29] Arthur Holly Compton and Hans Albrecht Bethe. “Composition of Cosmic Rays”. In: *Nature* 134 (1934), pp. 734–735 (cit. on p. 16).

[30] Kiran Munawar. “Identifying cosmic ray induced cascade events with IceTop”. In: 2017 (cit. on p. 17).

[31] Matthew David Healy. “Composition of the Highest Energy Cosmic Rays”. PhD thesis. University of California Los Angeles, 2007 (cit. on p. 17).

[32] Satyendra Thoudam et al. “Cosmic-ray energy spectrum and composition up to the ankle: the case for a second Galactic component”. In: *Astronomy & Astrophysics* 595 (2016), A33 (cit. on p. 17).

[33] Douglas R Bergman and John W Belz. “Cosmic rays: the Second Knee and beyond”. In: *Journal of Physics G: Nuclear and Particle Physics* 34.10 (2007), R359 (cit. on p. 17).

[34] Hans Dembinski et al. “Data-driven model of the cosmic-ray flux and mass composition from 10 GeV to 10ˆ{11} GeV”. In: *arXiv preprint arXiv:1711.11432* (2017) (cit. on p. 17).

[35] A. Albert et al. “The all-particle cosmic ray energy spectrum measured with HAWC”. In: 2021 (cit. on p. 18).

[36] Thomas K Gaisser. “Cosmic rays and particle physics”. In: *Comments on Nuclear and Particle Physics* 11.1 (1982), pp. 25–39 (cit. on p. 18).

[37] Th K Gaisser and M Honda. “Flux of atmospheric neutrinos”. In: *Annual Review of Nuclear and Particle Science* 52.1 (2002), pp. 153–199 (cit. on p. 18).

[38] Stephan Meighen-Berger and Mingyang Li. “Atmospheric muons from electromagnetic cascades”. In: *arXiv preprint arXiv:1910.05984* (2019) (cit. on p. 18).

[39] Thomas K. Gaisser et al. “Precision of analytical approximations in calculations of atmospheric leptons”. In: *Proceedings of 36th International Cosmic Ray Conference PoS(ICRC2019)* (2019) (cit. on p. 18).

[40] Anatoli Fedynitch et al. “A state-of-the-art calculation of atmospheric lepton fluxes”. In: 2017 (cit. on p. 18).

[41] Anatoli Fedynitch, William Woodley, and M-C Piro. “On the Accuracy of Underground Muon Intensity Calculations”. In: *The Astrophysical Journal* 928.1 (2022), p. 27 (cit. on p. 20).

[42] John V. Jelley. “Cerenkov radiation and its applications”. In: *British Journal of Applied Physics* 6 (1955), pp. 227–232 (cit. on p. 22).

[43] Pavel A Cherenkov. “Visible light from clear liquids under the action of gamma radiation”. In: *Comptes Rendus (Doklady) de l’Aeademie des Sciences de l’URSS* 2.8 (1934), pp. 451–454 (cit. on p. 22).

[44] Morten Ankersen Medici. “Search for Dark Matter Annihilation in the Galactic Halo using IceCube”. English. PhD thesis. 2016 (cit. on p. 23).

[45] A Achtenberg. “Contributions to the 29th International Cosmic Ray Conference (ICRC2005), Pune, India, August 2005”. In: (2005) (cit. on p. 24).

[46] Mark G Aartsen et al. “The IceCube Neutrino Observatory: instrumentation and online systems”. In: *Journal of Instrumentation* 12.03 (2017), P03012 (cit. on pp. 26, 28, 37).

[47] Abraham Achterberg et al. “First year performance of the IceCube neutrino telescope”. In: *Astroparticle Physics* 26 (2006), pp. 155–173 (cit. on p. 26).

[48] Michael James Larson. “Simulation and identification of non-Poissonian noise triggers in the IceCube neutrino detector”. In: *Theory of Computing Systems Mathematical Systems Theory* (2013), p. 19 (cit. on pp. 26, 30, 32, 36, 37).

[49] R. Abbasi et al. “The design and performance of IceCube DeepCore”. In: *Astroparticle Physics* 35.10 (May 2012), pp. 615–624. ISSN: 0927-6505. DOI: 10.1016/j.astropartphys.2012.01.004. URL: http://dx.doi.org/10.1016/j.astropartphys.2012.01.004 (cit. on pp. 27, 29).

[50] Christopher Wiebusch. “Physics capabilities of the IceCube DeepCore detector”. In: *arXiv preprint arXiv:0907.2263* (2009) (cit. on p. 28).

[51] MG Aartsen et al. “Measurement of the cosmic ray energy spectrum with IceTop-73”. In: *Physical Review D* 88.4 (2013), p. 042004 (cit. on p. 28).

[52] IceCube Neutrino Observatory. *SLC hit cleaning*. URL: https://wiki.icecube.wisc.edu/index.php/SLC_hit_cleaning (visited on 04/02/2022) (cit. on pp. 29, 35).

[53] Summer Blot. *Neutrino oscillation measurements with IceCube*. June 2020. DOI: 10.5281/zenodo.4156203. URL: https://doi.org/10.5281/zenodo.4156203 (cit. on p. 30).

[54] Kayla Leonard. *Atmospheric Neutrino Oscillations in IceCube DeepCore*. July 2020. DOI: 10.5281/zenodo.4124129. URL: https://doi.org/10.5281/zenodo.4124129 (cit. on p. 30).

[55] 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 (cit. on p. 30).

[56] Étienne 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 p. 30).

[57] Samuel Flis, Martin Wolf, and Matthias Danninger. “Extending IceCube Low Energy Neutrino Searches for Dark Matter with Deep- Core”. In: *International Cosmic Ray Conference*. Vol. 33. International Cosmic Ray Conference. Jan. 2013, p. 920 (cit. on p. 31).

[58] Costas Andreopoulos et al. “The GENIE neutrino Monte Carlo generator: physics and user manual”. In: *arXiv preprint arXiv:1510.05494* (2015) (cit. on pp. 32, 36).

[59] IceCube Neutrino Observatory. *OscNext MuonGun Optimization*. URL: `https://wiki.icecube.wisc.edu/index.php/OscNext_MuonGun_Optimization` (visited on 05/03/2022) (cit. on pp. 32, 37).

[60] Juan Pablo Yáñez Garza. “Measurement of neutrino oscillations in atmospheric neutrinos with the IceCube DeepCore detector”. In: 2014 (cit. on p. 33).

[61] Mark G Aartsen et al. “Energy reconstruction methods in the IceCube neutrino telescope”. In: *Journal of Instrumentation* 9.03 (2014), P03009 (cit. on p. 33).

[62] Martin Antonio Unland Elorrieta. “Studies on dark rates induced by radioactive decays of the multi-PMT digital optical module for future IceCube extensions”. MA thesis. WWU Münster, 2017 (cit. on p. 34).

[63] Jakob van Santen. “Neutrino Interactions in IceCube above 1 TeV: Constraints on Atmospheric Charmed-Meson Production and Investigation of the Astrophysical Neutrino Flux with 2 Years of IceCube Data taken 2010–2012”. In: 2014 (cit. on p. 36).

[64] Y Becherini et al. “A parameterisation of single and multiple muons in the deep water or ice”. In: *Astroparticle Physics* 25 (2006), pp. 1–13 (cit. on p. 36).

[65] Alexander Jung. “Machine learning basics”. In: *Machine Reading Comprehension* (2021) (cit. on p. 40).

[66] Giuseppe Carleo et al. “Machine learning and the physical sciences”. In: *Reviews of Modern Physics* 91.4 (2019), p. 045002 (cit. on p. 40).

[67] Pankaj Mehta et al. “A high-bias, low-variance introduction to Machine Learning for physicists”. In: *Physics reports* 810 (2019), pp. 1–124 (cit. on p. 42).

[68] Andrew Zhai and Hao Wu. “Making Classification Competitive for Deep Metric Learning”. In: *ArXiv* abs/1811.12649 (2018) (cit. on p. 44).

[69] Sachin Kumar and Yulia Tsvetkov. "Von mises-fisher loss for training sequence to sequence models with continuous outputs". In: *arXiv preprint arXiv:1812.04616* (2018) (cit. on p. 44).

[70] HandWiki. *von Mises distribution*. Last accessed May 2022. URL: `https://handwiki.org/wiki/Von_Mises_distribution#cite_note-Risken89-1` (cit. on p. 45).

[71] Marc Claesen and Bart De Moor. "Hyperparameter search in machine learning". In: *arXiv preprint arXiv:1502.02127* (2015) (cit. on p. 46).

[72] Stanford cs231. *Neural Networks Part 3: Learning and Evaluation*. Last accessed May 2022. URL: `http://www.bdhammel.com/learning-rates/` (cit. on p. 47).

[73] Leslie N Smith. "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay". In: *arXiv preprint arXiv:1803.09820* (2018) (cit. on p. 48).

[74] Chen Xing et al. "A walk with sgd". In: *arXiv preprint arXiv:1802.08770* (2018) (cit. on p. 48).

[75] Samuel L Smith et al. "Don't decay the learning rate, increase the batch size". In: *arXiv preprint arXiv:1711.00489* (2017) (cit. on p. 48).

[76] Samuel L Smith and Quoc V Le. "A bayesian perspective on generalization and stochastic gradient descent". In: *arXiv preprint arXiv:1710.06451* (2017) (cit. on p. 48).

[77] Nitish Shirish Keskar et al. "On large-batch training for deep learning: Generalization gap and sharp minima". In: *arXiv preprint arXiv:1609.04836* (2016) (cit. on p. 48).

[78] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (cit. on p. 49).

[79] Pankaj Mehta et al. "A high-bias, low-variance introduction to machine learning for physicists". In: *Physics reports* 810 (2019), pp. 1–124 (cit. on p. 51).

[80] Sunitha Basodi et al. "Gradient amplification: An efficient way to train deep neural networks". In: *Big Data Mining and Analytics* 3.3 (2020), pp. 196–207 (cit. on p. 50).

[81] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks". In: *Neural Networks* 132 (2020), pp. 428–446 (cit. on p. 51).

[82] Zach Griffith. *Visualizing I3 Files*. URL: `https://events.icecube.wisc.edu/event/87/sessions/1470/attachments/4732/5188/viewing_i3_files.pdf` (visited on 05/10/2022) (cit. on p. 52).

[83] Kevin Meagher. *I3File Tools and Visulaization*. URL: https://events.icecube.wisc.edu/event/105/contributions/1081/attachments/857/928/Meagher_2019Bootcamp_I3FileVisualization.pdf (visited on 05/10/2022) (cit. on p. 52).

[84] Mads Ehrhorn. "Convolutional neural network neutrino reconstruction in IceCube". PhD thesis. The Niels Bohr Institute, Faculty of Science, University of Copenhagen, 2020 (cit. on p. 52).

[85] Rasmus Ørsøe. "A Graph Neural Network Approach to Low Energy Event Reconstruction in IceCube Neutrino Observatory". PhD thesis. The Niels Bohr Institute, Faculty of Science, University of Copenhagen, 2021 (cit. on pp. 52, 62).

[86] Rasmus Ørsøe. *I3ToSQLite*. URL: https://github.com/RasmusOrsoe/I3ToSQLite (visited on 05/10/2022) (cit. on p. 52).

[87] L Garren et al. "Monte carlo particle numbering scheme". In: *The European Physical Journal. C, Particles and Fields*. 15.1-4 (2000), pp. 205–207 (cit. on p. 54).

[88] Justin Gilmer et al. "Neural message passing for quantum chemistry". In: *International conference on machine learning*. PMLR. 2017, pp. 1263–1272 (cit. on pp. 57, 60).

[89] Katy Börner, Soma Sanyal, Alessandro Vespignani, et al. "Network science". In: *Annu. Rev. Inf. Sci. Technol.* 41.1 (2007), pp. 537–607 (cit. on p. 58).

[90] Joan Bruna et al. "Spectral networks and locally connected networks on graphs". In: *arXiv preprint arXiv:1312.6203* (2013) (cit. on p. 60).

[91] Thomas N Kipf and Max Welling. "Semi-supervised classification with graph convolutional networks". In: *arXiv preprint arXiv:1609.02907* (2016) (cit. on p. 60).

[92] Zonghan Wu et al. "A comprehensive survey on graph neural networks". In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24 (cit. on p. 61).

[93] Yue Wang et al. "Dynamic Graph CNN for Learning on Point Clouds". In: *ACM Trans. Graph.* 38.5 (Oct. 2019). ISSN: 0730-0301. DOI: 10.1145/3326362. URL: https://doi.org/10.1145/3326362 (cit. on p. 63).

[94] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf (cit. on p. 66).

[95] David J Hand. “Measuring classifier performance: a coherent alternative to the area under the ROC curve”. In: *Machine learning* 77.1 (2009), pp. 103–123 (cit. on p. 66).