University of Padova

Department of 'Dipartimento di Fisica e Astronomia Galileo Galilei'

Master Thesis in Physics

Implementation of Graph Neural Networks for reconstruction of neutrino energy at the IceCube Real-Time Alert system

Supervisor
Prof. Elisa Bernardini
University of Padova

Co-supervisor
Dr. Sarah Mancina
University of Padova

Master Candidate
Diego Alberto Coloma Borja

Student ID
2071392

Academic Year

2023-2024

Abstract

The IceCube Neutrino Observatory is currently the largest active neutrino detector in the world, aiming to detect high-energy emissions from astrophysical sources. As such it is a cornerstone of experimental astroparticle physics. Particularly, it is an important part of the efforts to pursue a multimessenger approach to astronomical detections, being able to send alerts to collaborating observatories when a significant event is observed.

With the intention of staying at the forefront of efficiency in its detection, IceCube is constantly updating their techniques for reconstruction of physical parameters from detected events. Taking this into account, the work presented in this thesis seeks to improve on the reconstruction of energy by implementing Graph Neural Networks, machine learning algorithms designed to recognize patterns from data structured in an irregular manner without loss of information.

Graph Neural Networks are a good candidate for the reconstruction of parameters. They have historically shown their ability to perform efficient observations in record times. This fact makes them a good candidate for implementation in systems that reconstruct event parameters during online data taking. The greatest challenge for the implementation of these kinds of algorithms then is the training process, where network architectures need to be trained on proper datasets so they can be deployed without constant supervision.

The work on this thesis focuses on applying Graph Neural Networks algorithms for energy reconstruction on the full energy spectrum arriving to the IceCube neutrino Observatory. Configuring the network was a challenge because of the need of implementing a graph representation that is efficient in the 100 GeV range as well as the 100 PeV range.

From the results in this work, we observe that even when trained conservatively, the GNN method is comparatively effective to the currently implemented methods in the RealTime Alert System at IceCube. The precision is noticeably improved in the low energy range, while also showing significant accuracy in the rest of the energy spectrum. These results are obtained while maintaining efficient runtimes across the board, with the possibility of increasing the performance by further management of CPU and GPU resources.

we conclude that Graph Neural Networks are effective in the reconstruction of energy in the full energy range of the samples produced from the RealTime Alert System in IceCube, highlighting their possible implementation in online systems as well as possible improvements in the training process.

Contents

Listing of figures

Listing of tables

1
Introduction

Neutrinos are some of the most elusive particles currently present in the Standard Model of particle physics. Once they are produced, their neutral charge and low cross section makes them very hard to observe. Nevertheless, this also makes them excellent probes of the processes that take place at the location of their production. Neutrinos are also key players in determining the answers to many currently relevant physics questions [1]: Are they truly stable or do they interact with matter not yet predicted by the Standard Model? Can they be considered part of the hot dark matter permeating the universe? What is the origin of cosmic Rays?

To access information that reveals the mysteries of physical phenomena beyond the Standard Model, high energies are needed. At the energy scales of accelerators available on Earth, particles show no clear evidence of new physics. This leads scientists to look to natural sources of high energy particles. From this, as well as the original observation of Cosmic Rays by Victor Hess in 1912, the discipline of astroparticle physics was born. Astroparticle physics focuses on using the natural occurring highly-energetic particles to study and observe mechanisms currently not observable on the surface of the planet.

In recent years, astronomy and astroparticle physics have given increasing importance to a multimessenger approach of observations. The processes involved in creating electromagnetic radiation through which we observe astronomical sources are intrinsically connected to the production of other messengers like cosmic rays and neutrinos. Taking this into account, a multimessenger approach integrates observatories of many different kinds of particles to create a complete picture of high-energy particle physics phenomena.

An important player in the multi-messenger approach is The IceCube Neutrino Observatory. IceCube is a cubic-kilometer detector located deep in the Antarctic ice and designed to perceive the flux of neutrinos arriving to the Earth. The detector has been fully functional since 2011 and since then it has been able to provide insight on many mysteries surrounding particle and astroparticle physics, confirming the existence of an astrophysical neutrino flux on Earth and identifying a number of detailed neutrino emitters in the night sky.

One of the systems in place in IceCube that make it very relevant in the multimessenger landscape is the RealTime Alert System, which processes events as they arrive to the detector to produce a sample of neutrino candidate events which can be used for alerting the astrophysics community worldwide of exceptional emissions. From this sample, the Gamma-Ray Follow Up (GFU) program identifies the clustering of these significant events from known Gamma-Ray emitters and sends notices not collaborating Gamma Ray Telescopes.

An important highlight from the history of IceCube and the GFU program is the detection of neutrinos from the flaring blazar TXS 0506+0506 [2]. When a high energy neutrino was detected in IceCube coming from the direction of said blazar, an established alert system automatically sent notices to gamma rays observatories throughout the world, prompting a coordinated worldwide observation of this object. This observation has been a cornerstone of the multimessenger approach to astronomical observations, where observations of neutrinos alongside Very High Energy Gamma Rays provide not only information on the location of sources, but also the mechanisms that govern the production of cosmic messengers. Particularly, this detection highlights the usefulness of neutrino detectors like IceCube, which are able to detect events from the whole sky with virtually no downtime, to prompt multimessenger studies.

An important part of the systems in place for alerts is the reconstruction of physical parameters from detected events like energy and angular direction. Although no method is completely effective at reconstruction, the observatory is constantly updating and testing new technologies for this effort. From this idea, machine learning algorithms have been tested in the past years, showing their efficiency in obtaining results with very fast runtimes. Considering this, the present work aims to test the energy reconstruction capabilities from Graph Neural Networks, machine learning algorithms designed to process complex and irregular data structures without loss of information.

The following thesis is structured in the following manner: Chapter 2 presents an overview of Multimessenger Astrophysics, describing the production mechanisms that link different cosmic messengers and how they can be detected on Earth; Chapter 3 then explains how the IceCube detector functions, with particular emphasis on the characterization of events as well as the reconstruction of physical parameters from said events; Afterwards, Chapter 4 explains the functioning of the RealTime Alert system, particularly, how it creates a sample of neutrino-candidate events for analysis; In Chapter 5 instead we introduce the concept of machine learning techniques, with focus on the context of reconstruction of physical parameters from neutrino experiments through the use of Graph Neural Networks; Finally, Chapter 6 explains how we implement a Graph Neural Network to energy reconstruction, emphasizing the challenges to apply them to samples relevant to the IceCube RealTime Alert system and how the implementation compares to current energy reconstruction methods. A summary and evaluation, including future possibilities are given in Chapter 7.

2
Multimessenger Astrophysics

Throughout history, astronomers have pointed instruments to the sky in the hope of observing objects outside of our planet, identifying their properties, and use them to gain insight on physical phenomena, beyond what can be tested on ground. In contrast, grounded particle physicists have historically devised ways to study the properties of matter on Earth. In typical Earth conditions, particles behave in conformity to macroscopic properties of systems. Nevertheless, as energies increase, fundamental particles are allowed to interact freely according to the intrinsic properties of the physical processes that govern them, For this purpose, physicists have built particle accelerators that allow for the study of particle physics at high energies, not naturally occurring on Earth.

The perspective that particle physics was limited to what scientists can recreate at labs on Earth changed in the year 1912, when Victor Hess took the skies in a number of hot air balloon flights to measure the radiation in the atmosphere. Upon the observation that radiation increased with height and that there is no significant change at night or an eclipse, Hess concluded: "*The results of the present observations seem most likely to be explained by the assumption that radiation of very high penetrating power enters from above into our atmosphere*" [3]. Victor Hess had for the first time observed Cosmic Rays, charged particles at high energies arriving to the Earth from sources in space. With this discovery, the field of *astroparticle physics* was started. There are natural sources of high energy particles in the Universe, and the methods used for detection in Earth accelerator lab environments can be applied to these types of radiation. Furthermore, this natural sources can reach energies still unavailable on Earth.

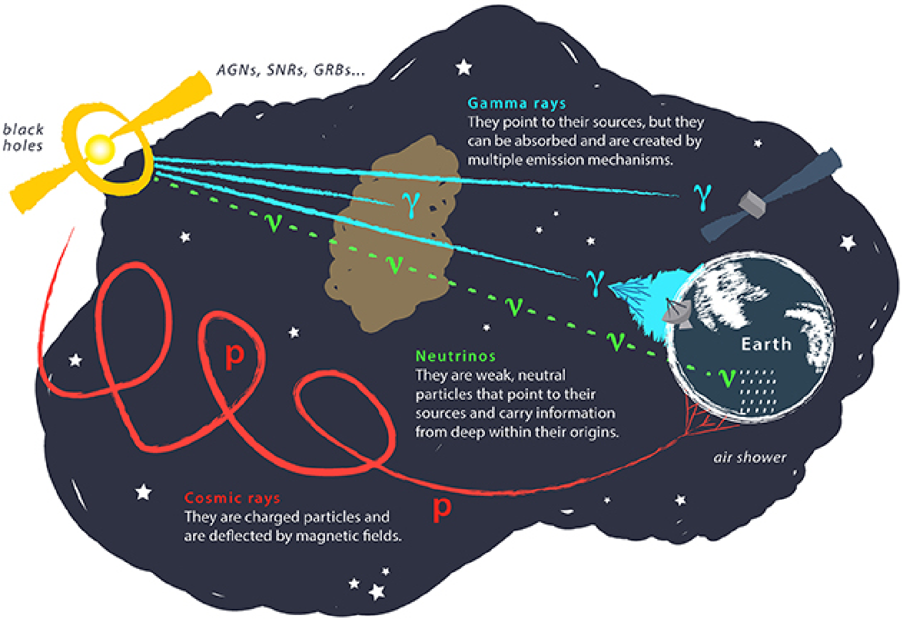
However, Astroparticle physics has been historically somewhat disconnected from traditional astronomy. Astronomers study physical objects using the electromagnetic radiation they emit, applying methods from spectroscopy to study their composition. On the other hand, astroparticle physicists aim to detect the particles penetrating through the atmosphere. In recent years, this separation has become increasingly more diffuse, the same process that produce flares of electromagnetic radiation in astrophysical objects also lead to the acceleration of cosmic rays, making the fields intrinsically connected. The recognition of this has lead to the discipline of **Multimessenger Astrophysics**. Which consists of combining the study of different *messengers* from astronomical sources in order to obtain a better picture of the physical processes that govern particles at this high energies. There are 4 types of distinct *messengers* typically considered in these kind of studies:

1. Cosmic Rays
2. Gamma Rays
3. Neutrinos
4. Gravitational Waves

Although Electromagnetic radiation and Cosmic Rays have been used as messengers for years, the quick development in recent years for neutrino and gravitational wave detection has solidified the possibility of performing studies on astrophysical phenomena by taking a *multi-messenger approach*. An example of the fruits of the Multimessenger approach was evidenced in 2017. In September of that year, a high energy **Neutrino** event, IceCube-170922A, was detected at an energy of 290 TeV in the IceCube Neutrino Observatory. Because its direction was consistent with the astrophysical object TXS 0506+0506, several follow-up observations were performed in the same direction, identifying a flare-up in activity with the detection of several highly energetic **Gamma Rays** [2]. Identifying for the first time a source in coincidence of high-energy neutrinos and very high energy gamma rays.

In this section, three main messengers are presented: Cosmic Rays, Gamma Rays and Neutrinos. A simplified sketch of these messengers being detected on Earth from one source is presented in Figure 2.1. Among these, neutrinos are the most important for this thesis and the experimental methods used for their detection are explored throughout the following chapters in the context of the IceCube Observatory.

Gravitational waves are an important messenger in the this field, they are perturbations in space-time caused by the accelerated movement of massive objects, like neutron stars in-spiraling before merging. While observations have related the production of gravitational waves and gamma rays [4], processes that correlate gravitational waves and neutrinos have not been definitively determined, and therefore they are not a focus on this thesis.



Gamma rays
They point to their sources, but they can be absorbed and are created by multiple emission mechanisms.

Neutrinos
They are weak, neutral particles that point to their sources and carry information from deep within their origins.

Cosmic rays
They are charged particles and are deflected by magnetic fields.

Figure 2.1: Simplified sketch of an astrophysical source producing different messengers, and the path they take to arrive on Earth. Taken from [5]

2.1 Cosmic Rays

Cosmic Rays are an example of high energy messengers, charged particles traveling at enormous speeds through space. Typically coming from outside the Solar System, they traverse long distances from sources before they can be detected on Earth. Understanding the mechanisms that govern their production and dispersion through the Universe promises to offer rich insights into the processes that govern particles in and outside of the Standard Model of Particle Physics.

2.1.1 Energy Spectrum of Cosmic Rays

Cosmic rays can be observed in a wide range of energies, reaching up to around 300 EeV when they are detected on Earth. The full spectrum of cosmic rays include a variety of compositions.

99% are composed of nuclei, mostly very light like protons, but heavier ions have been observed. The Pierre Auger Observatory [6] has performed observations that identify a composition of light nuclei like protons with mixing of heavier nuclei up to iron at higher energies [7]. On the other hand, data from the Telescope Array indicates a composition of mostly light nuclei at all energies [8]. Although these observations can be considered contradictory, the uncertainty associated with them makes them compatible with each other. Then, in combination they poiny to the main component of the cosmic ray spectrum being protons, with at least some mixing of higher nuclei as detected energies increase.

The energy spectrum of Cosmic Rays span from a few 10^9 eV up to 10^{20} eV, presenting steep falls with increasing energies. The shape of this spectrum can be observed of Figure 2.2. The energy flux can be parameterized by the following power law equation:

$$\frac{d\varphi}{dE} = \varphi_0 \left(\frac{E}{E_0}\right)^{\gamma} \tag{2.1}$$

where φ_0 and E_0 define normalization constants while γ, called the ***spectral index***, determines the steepness of the power law's spectral shape. Two particular structures can be highlighted from Figure 2.2: the *knee* and the *ankle*, both of which indicate notable changes in the slope of the spectrum and therefore spectral index. Before the knee, the spectrum can be characterized by a spectral index of $\gamma = -2.7$, while after the knee it drops to $\gamma = -3.1$. An interpretation for the drop is that, at this energy, the main contribution shifts from galactic source to extragalactic sources [10]. For lower energies, cosmic rays from outside the Milky Way are dissipated by interstellar matter, and do not arrive to the Earth. Furthermore, at higher energies, Cosmic Rays produced inside the Milky Way are no longer bound by the galactic magnetic field and can therefore escape from possible detection. Additionally, after the knee, the composition of cosmic rays start to include heavier nuclei. On the other hand, the ankle instead shows a softening from $\gamma = -3.1$ to $\gamma = -2.5$, the full nature of which is still subject to debate.

When cosmic rays carry energies greater than 1 EeV, they are referred to as Ultra High Energy Cosmic Rays (UHECRs). At these energies, Cosmic Rays are less likely to be deflected from their original production or acceleration site, being useful tools for multimessenger studies of sources. Additionally, these energies are beyond the current scope of the LHC, offering perspectives on physics mechanisms at this otherwise unavailable range. These facts make UHECRs particularly interesting to study. However, it should be noted from the spectrum, that the maximum energy of detection is around 300 EeV, at which point there is a sudden

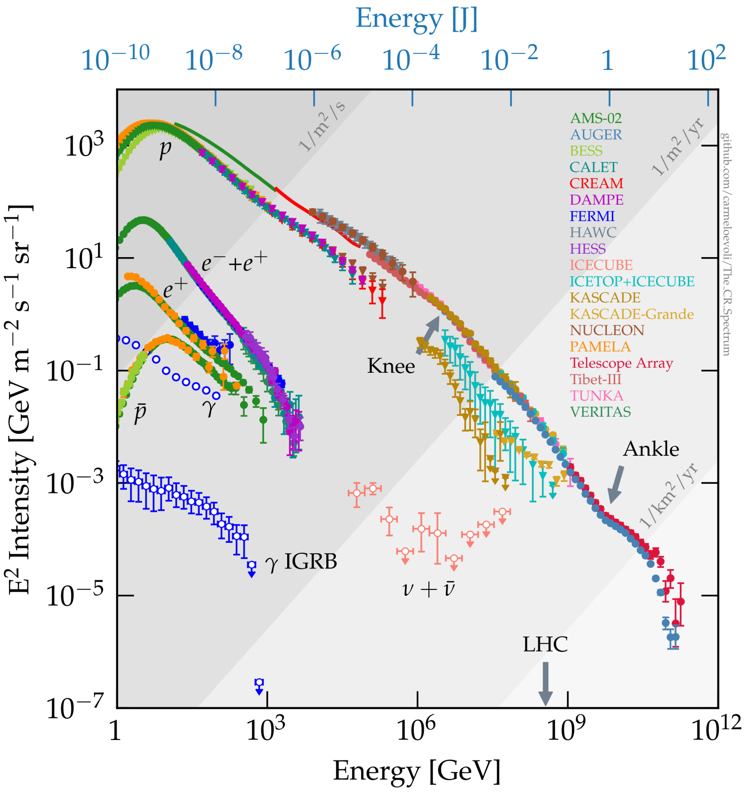


Figure 2.2: Image taken from [9] detailing the observed Cosmic Ray spectrum gathering data from many experiments on Earth. The different trends correspond to different particle compositions, including hadronic, leptonic and photonic, as well as neutrinos. The different shaded areas represent different orders of magnitude of the expected detection fluxes. The maximum energy achieved by the Large Hadron Collider is also marked to signal the potential of Cosmic Rays to elucidate mechanisms relevant to particle physics.

drop in detection. The suppression at these energies is an interesting question in understanding the production mechanisms of Cosmic Rays. On one hand, hypothesis exist that this is the maximum range that cosmic accelerators can produce. On the other hand, this phenomenon is explained by the proposed Greisen-Zatespin-Kuzmin (GZK) cut-off [11, 12]. This explains that the interaction of cosmic ray protons with the Cosmic Ray Background can limit the energies of CRs [13]:

$$p + \gamma_{CMB} \rightarrow \Delta^+ \rightarrow \begin{cases} p + \pi^0 \\ n + \pi^+ \end{cases} \tag{2.2}$$

This is a process that can occur at energies higher than 50 EeV, with the cross-section increasing with energy. The nucleons produced in these interactions then carry lower energies than the originals, continuing to interact, while pions decay into photons or neutrinos, making this process an effective limit for Cosmic Rays to be able to cross large distances in space.

2.1.2 Cosmic Ray Sources and Acceleration Models

In order to produce UHECRs, there need to exist mechanisms violent enough to accelerate charged particles to this energy. On Earth, this is obtained by accelerators creating powerful magnetic fields over large effective sizes. Similar mechanisms to this must be at play in the Universe to generate the observed energies of Cosmic Rays. Based on this idea, a tool was developed to classify potential candidates for acceleration of Ultra High Energy Cosmic Rays. This tool is called the **Hillas Plot**, shown in Figure 2.3. The horizontal axes shows the size of of candidate sources, with the magnetic field strength presented on the vertical axis. Diagonal lines show the limits necessary to attain observable energies of cosmic rays, with types of sources displayed on shaded areas.

According to observations, Cosmic Rays should be observed from Galactic and Extragalactic sources. One type of sources proposed for galactic cosmic rays are Supernovae Remnants (SNRs) [15]. When a star of sufficient mass (bigger than 1.4 solar masses) dies, it collapses in a violent process that results in an explosion of the star's material [16]. This process leaves behind an expanding front of accelerated matter which can produce a shock center for acceleration of matter.

The natural question is how a system like this works for acceleration of Cosmic Rays. When a particle at an initial energy E_0 hits a plane wave shock front, it gets imbued with an Energy gain $\Delta E = \xi E$ where ξ is directly proportional to the speed of the shock front β. The charged particle can bounce and collide with the shock front many times until it achieves an energy sufficient to escape this cycle. The fact that the energy gain is linear with respect to the speed of the shock front leads to the name *first-order Fermi acceleration* [16].

This name comes from the mechanism previously derived by Fermi: The *second-order Fermi acceleration* [16]. Fermi designed his theory for this acceleration mechanism assuming the existence of localized irregular portions of concentrated interstellar magnetic fields. By constant scattering in these clouds, charged particles gain energy proportional to the square speed of the traveling cloud β^2. This process is comparatively slow so it would be inefficient in providing the necessary energy to CRs.

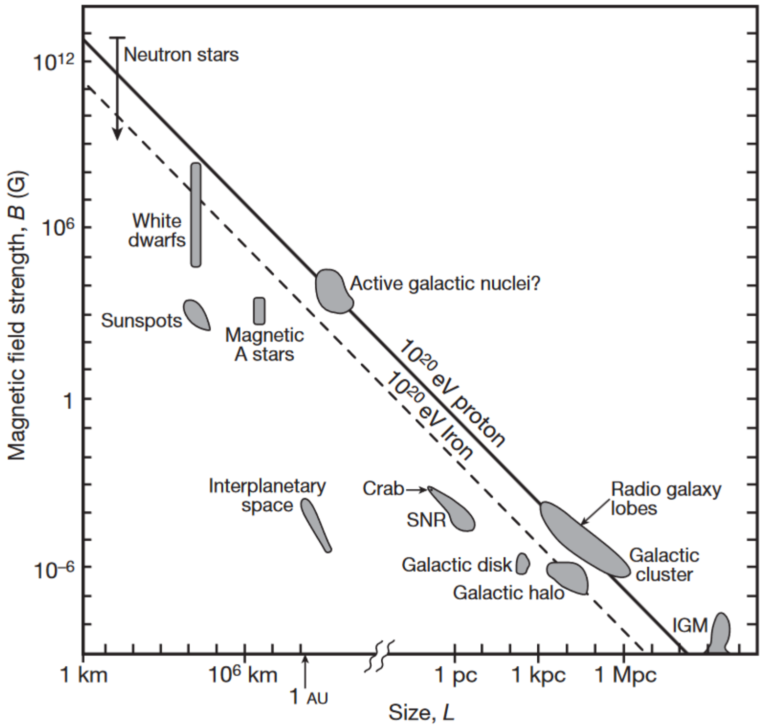


Figure 2.3: Hillas diagram of candidates for UHECR sources. Different candidates are presented according to their size and the strength of the magnetic fields. Graph Taken from [14].

As the Hillas Diagram suggests (Figure 2.3), one of the possible type of sources for cosmic rays up to 100 EeV are Active Galactic Nuclei (AGN), compact regions in the centers of galaxies with high radiation activity. A diagrammatic picture of an AGN is presented in Figure 2.4. At the core of these galaxies, there is a super massive black hole. Because of its gravitational pull, matter falling in into it forms a rotating accretion disk from which radiation is emitted. Matter also shoots outwards from the plane of the accretion disk in form of Jets of relativistic radiation. Finally, a torus of dusty matter surrounds the nucleus, obscuring radiation from the accretion disk [16]. It should be noted that AGNs can be classified into different types according to their observation on Earth. Depending on the angle of observation to the accretion disk plane they can be sorted as Blazars, Radio Quasars, Radio Galaxies or Seyfert Galaxies, but in reality they refer to the same kind of objects.

Due to their complex structure, Active Galactic Nuclei present many cites for acceleration of charged particles. The Pierre Auger Collaboration has identified a degree of correlation between the direction of Ultra High Energy Cosmic Rays and the position of AGNs in the sky

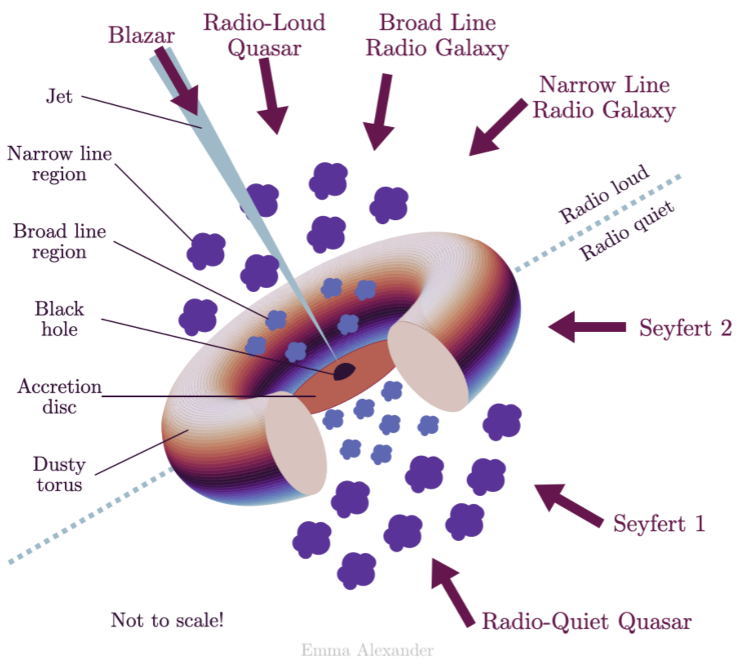


Figure 2.4: Graphic Schematic of Active Galactic Nuclei (AGN). The different parts are shown and the thick arrows indicate the viewing angle from Earth that classifies the observed objects. Taken from [17].

[18]. However, these results were not unambiguous and haven't been confirmed. Finally, the different mechanisms at play provide sites for the production not only of Cosmic Ray acceleration, but also high energy Gamma Rays (Section 2.2) and neutrinos (Section 2.3), making them very interesting objects for Multi-Messenger observations.

2.1.3 Detection of Cosmic Rays on Earth.

At this point, we have a good understanding of Cosmic Rays as they run through the Universe, before they reach the Earth. To complete the picture, now we turn to their arrival on Earth. When the charged particles arrive to Earth, they have to pass through the atmospheric layers before they touch the Earth surface. At the moment of arrival, they are referred to as *primaries*, they interact with the atmospheric components which leads to the development of wide-spanning showers of *secondary particles.* A sketch of a typical development of a shower is depicted in Figure 2.5. When the primary interacts with the nuclei in the atmosphere, the hadronic interaction ultimately leads to the production of light mesons, like pions $\pi^{0,\pm}$. The pions then decay or interact with further atmospheric components, leading to the development of subsequent hadronic cascades, composed mainly of charged mesons, as well as electromagnetic showers, which are composed primarily of photons and electrons. After enough interactions, the particles in the shower reach a threshold energy where the decay process take over the interactions, so the particles decay. The products of the electromagnetic shower tend to dissipate into the atmosphere. Conversely, some mesons decay into muons and neutrinos, which are highly penetrating and reach the ground as a front of the shower. Neutrinos from showers compose the atmospheric neutrino flux, while muons, being charged particles, provide the main avenue for Cosmic Ray detection on Earth.

The more secondary particles detected from the shower, more quality information is gained to reconstruct the physical parameters of the primary incoming Cosmic Ray. Then, because secondaries expand outwards from the axis of the primary's trajectory, it is important to build detectors that span a wide area on the surface of the Earth. One prominent example of such detector is the Pierre Auger Observatory [6]. This Observatory consists of an array of Water Cherenkov Detector (WCD) array spanning 3000 km^2 in Western Argentina, with the support of 24 Fluorescent Radiation Detectors to observe the development of the shower secondaries in the atmosphere. This hybrid method of detection leads to clearer pictures in the reconstruction of Cosmic Ray Energies and trajectories. A similar technique is employed in the northern hemisphere by The Telescope Array[20]. The system at place here consists of 507 surface scintillation detectors (SDs) distributed in a 762 km^2 grid Millard County in Utah with 3 Fluorescence Radiation Detector overlooking the array.

It should furthermore be noted that, looking back at the spectrum of Cosmic Ray energies (Figure 2.2), the small rate of flux at high energies require detector spanning large areas to improve probabilities for detection. This is a fact that turns out to be common for all kinds of high energy messengers.

2.2 Gamma Rays

The processes and sources described in the previous sections lead not only to the production of Cosmic Rays, but other kinds of prominent, high energy messengers. In this section, we discuss Gamma Rays, high energy photons that can be produced from energetic processes in the Universe. Photons are neutrally charged bosons, which means they do not get curved along their trajectory like cosmic rays do, therefore pointing directly towards sources.

Gamma Rays can be produced in many manners, but we highlight two main general avenues.

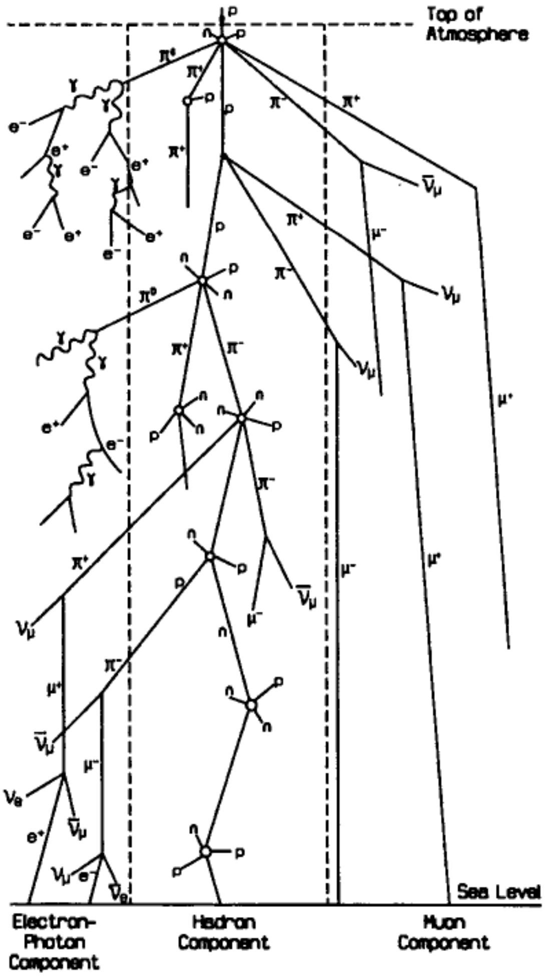


Figure 2.5: Sketch of a primary cosmic ray proton entering the atmosphere and producing a Shower of secondaries. The different components of the Shower are shown separately for illustrative purposes. Image taken from [19].

The first one is through what we refer to as *leptonic* mechanisms: electrons produced at sources typically do not arrive to Earth. Instead they interact with galactic and extragalactic magnetic fields as well as with lower energy radiation permeating the universe. Through this interactions, Gamma Rays are produced. On the other hand, *hadronic* mechanisms typically refer to photons produced by the interactions of Cosmic rays with permeating radiation backgrounds as well with other charged particles in the Universe. These high energy interactions can ultimately result in mesons, from which the neutral pion π^0 decays into a photonic pair.

Photons are highly penetrating, but as their energy increase, they are likely to interact with

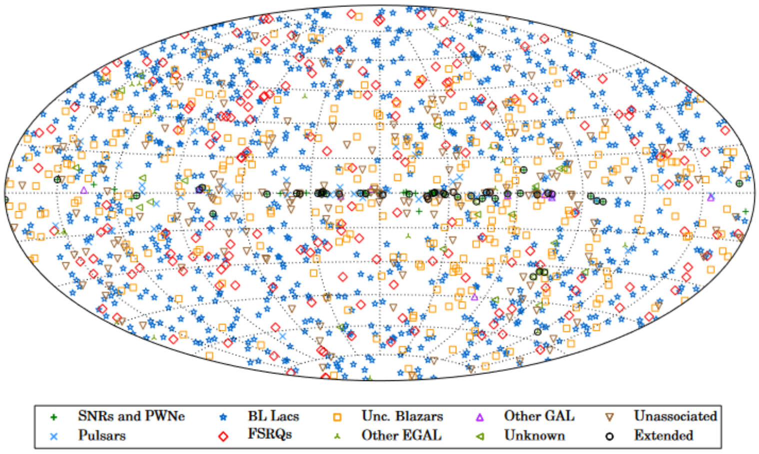


Figure 2.6: Image of the sky in galactic sources with the position of sources identified in the Third Catalog of Hard Fermi-LAT Sources (3FHL) and classified according to the speculated type of source. Image taken from [22].

background radiation in space, producing electron pairs and limiting the maximum energy at which Gamma Rays may travel through the Universe. At low energies (below 100 GeV), gamma ray photons are mostly dissipated into the atmosphere. Because of this, detections at this range need to be carried out by satellites outside from the Earth. An example of such experiment is carried out by the Fermi-LAT mission from NASA [21], which has obtained rich information on the full night sky, leading to a very comprehensive catalog of gamma-ray sources, as can be observed in Figure 2.6 [22].

At energies higher than 100 GeV, and below 100 TeV, Gamma Rays are classified as *Very-High Energy (VHE) Gamma Rays*. In this range, photon detections are suppressed due to their interaction with Extragalactic Background Light (EBL), limiting the effective distance at which we can observe Gamma Ray sources. Because of the lowered flux, satellites are inappropriate for detection and larger effective areas for detection at necessary. Luckily, at this energy Gamma Rays can penetrate into the atmosphere, where the photons interact with the atmosphere producing an electromagnetic shower. The leptons traveling downwards from this showers through air lead to the production of Cherenkov radiation in a cone expanding downwards. Using this as a medium for detection, instruments known as Imaging Air Cherenkov Telescopes (IACTs) have been devised, this type of detectors reflect the Cherenkov radiation into Photomultiplier tubes, allowing to reconstruct a large area of Cherenkov radiation and reconstruct the incoming Gamma from said information. Relevant and operating IACT experiments include MAGIC in La Palma, Canary Islands [23, 24], H.E.S.S in the Khomas region of Namibia [25], and VERITAS in Southern Arizona [26]. These experiments have led to a good exploration in the Very High-Energy range, offering detailed catalogs on sources, with promising developments in the future from more sensitive planned detectors, like the Cherenkov Telescope Array (CTA). The disadvantage from this kind of detectors is that, due to their reduced field of view, they are directional, so detector need to be pointed towards sources of incoming Gamma Rays. This disadvantage is fixed by a multi-messenger approach, using other messengers such as lower energy photons from satellite experiments or neutrinos to prompt detections from Very-High Energy Gamma ray Experiments. However, it should be noted that this experiments are still limited in the distance at which they can observe sources due to the interaction of Gamma Rays with EBL and the catalogs are restricted to this area.

Beyond the VHE range, above 0.1 PeV, events are called Ultra-High-Energy Gamma Rays. These events are heavily suppressed, therefore their detection requires instruments spanning large areas. An example of this is LHAASO in China [27], which has been successful in identifying a number of UHE Gamma Rays [28]. The mechanisms governing photon messengers at this range are still somewhat of a mystery, but they are very interesting in the search of sources for Cosmic Rays, since they are expected to be produced from interactions of charged particles in the UHECR range and point directly to their production site. Nevertheless, the mechanisms that limit observing distances for Gamma Rays at higher energies make it so only nearby galactic objects are typically observable in the UHE range.

2.3 Neutrinos

The final messengers to discuss in this Chapter, and the central object for the context of this thesis, are neutrinos, ν. Neutrinos are neutral particles that interact seldom and only through weak interactions. They belong to the leptonic families, typically identified by the associated lepton that intervenes in their interaction (electron, muon or tau). The Standard Model predicts the existence of massless neutrinos, however they have been found to possess a small mass which allows the phenomenon of neutrino oscillations, where a neutrino identified as one flavor (i.e. electron neutrino ν_e) might oscillate into a different flavor (i.e muon neutrino ν_μ) upon traveling some distance [13].

Because neutrinos are neutral and interact rarely, they arrive from straight paths from the sources of production. Many possible types of sources have been identified. Among of those most relevant to this work are:

1. **Atmospheric Neutrinos.** As explained in Section 2.1.3, cosmic rays arriving on Earth produce showers of particles. When the particles in this shower interact and/or decay, neutrinos maybe produced. These neutrinos span energy ranges starting from below the GeV range and can be found at increasingly higher energies up to above 500 TeV [29, 30]. Because of their characteristics, and the volume at which they are produced naturally on Earth, atmospheric neutrinos provide an effective source on information to understand phenomena like neutrino oscillation [31]. They nevertheless provide a significant background for the observation of neutrinos coming from astrophysical sources.

2. **Astrophysical Neutrinos.** When Cosmic Rays interact with matter or radiation in their trajectory, this results in violent processes with many different products. The interactions with matter result in the ultimate production of light mesons:

$$p + N \rightarrow \pi^{\pm} + K^{\pm} + \text{others...} \tag{2.3}$$

   Similarly, the interaction with radiation result in the production of light mesons like pions as described in equation 2.2. The decay of the charged mesons from these reactions is a direct source of neutrino production.

$$\pi^{\pm} \rightarrow \mu^{\pm} + \bar{\nu}_{\mu}(\nu_{\mu}) \tag{2.4}$$

   On the other hand, neutral mesons lead to the production of electromagnetic radiation in the form of Gamma Rays. Because the production mechanisms are connected, the joint detection of high energy neutrinos and gamma rays provide a direct avenue to understand the mechanisms of production and interaction of Cosmic Rays, an serve as indications of interesting activity in production sites Additionally, astrophysical neutrinos can occur at energies above the TeV limits currently in place for accelerators on Earth, making them probes for physics phenomena Beyond the Standard Model [1].

Neutrinos can also be found naturally in the sun, radioactive deposits on Earth, as well as the cosmological background [32]. Artificially, they are produced by nuclear reactors and accelerators on Earth. Nevertheless, they are produced outside of the energy range relevant to astrophysical sources, so we do not mention them in detail. All sources are however in the energy spectrum depicted in Figure 2.7.

Although rarely, neutrinos do interact through the weak force, mediated by the $W^{\pm}$ and Z^0 bosons. Although neutrinos are neutral, the products of this interaction lead to charged particles. This provides the main avenue for particle detection, therefore there is a need to build instruments that are favorable to the detection of such products. The first thing to take into account is the dimension requirements of such a detector. Waxman and Bahcall studied cosmic ray observations to set an upper bound for the flux of high energy neutrinos on Earth

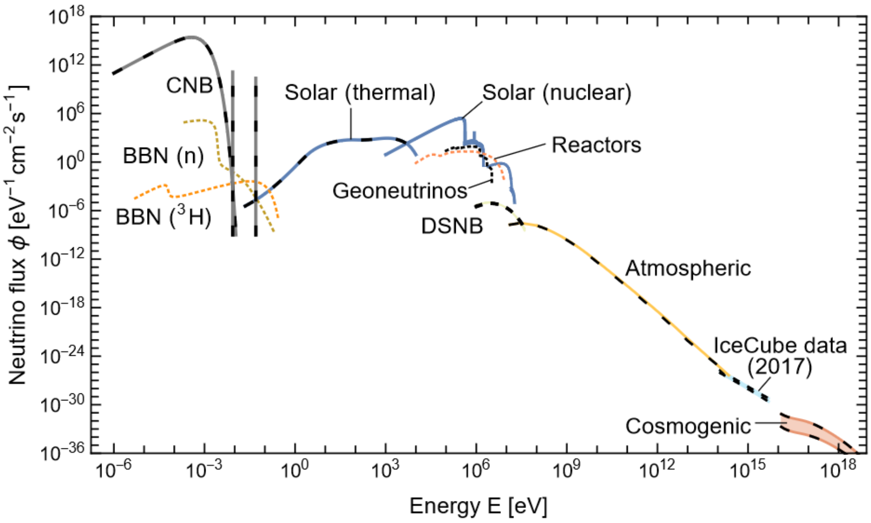


Figure 2.7: Neutrino energy spectrum depicting the flux from the different known sources as a function of energy. Taken from [32].

[33]. Based on their observations, and the low cross section for neutrino interactions, they suggest that neutrino detectors need to reach an effective volume of approximately 1 km^3 to be able to detect neutrinos from astrophysical sources.

The need for high volume detectors has led to the development of large-scale experiments, embedded into large bodies of water or ice, to take advantage of the optical qualities. Further details that motivate specifications on detectors are explored in the next chapter.

Throughout this chapter, we have explored the different messengers that carry information about particle physics processes at astrophysical sources. We have also highlighted the difficulties for their detection, particularly at high energies where they offer the clearest insight into unexplored phenomena. However, it should be noted that this difficulty is precisely what leads to the motivation to explore a multimessenger approach. While the sections in this chapter have been presented separately, the processes that govern them are all inextricably linked, which means that a proper model of astrophysical events will take into account the information they can all provide together. Neutrinos are elusive, therefore it is hard to get the full picture from astrophysical sources just from the detection of neutrinos, but they arrive directly from sources which makes the search for them a prompt to follow-up observations from gamma-ray observatories. Especially because at Very High Energies, IACTs need to be pointed in the right direction, consequent Cosmic Ray observations are also very important, they offer the models and expectations that integrate the different messengers into place.

3

Overview of the IceCube Neutrino Observatory

As explored in the previous chapter, neutrinos provide an excellent but elusive messenger for information on astrophysical sources. Because neutrinos don't interact through the electromagnetic force, the method to detect them consists of detecting the product of their interactions. When a neutrino interacts, it produces charged particles that then can be detected through several methods. Neutrinos however, interact very rarely. Therefore detectors that span a huge volume are required to increase the probability of detection. Additionally, since astrophysical events happen at high energies, large detectors provide an opportunity for events to deposit more energy and therefore richer information for analysis.

A method to detect charged particles is through the use of Cherenkov radiation. When a charged particle passes through a medium, if its speed is larger than the phase speed of light in said medium, the particle will produce light whose wavefront will be defined by a cone in the direction of the traveling particle. This light then can be detected by the use of photo-multiplier tubes (PMTs), providing signals proportional to the event's energy. In order to take advantage of this phenomenon a detector should be built into a medium with a high refractive index while also being transparent to radiation. Fortunately, water provides an effective medium for this to occur, leading to the idea of developing a large volume detector built into water.

3.1 The IceCube Detector

It should be noted that building artificial structures that can house a sufficient volume of water for astrophysical neutrino detection is unreasonable according to the Waxman-Bahcall bound, at least by current technology standards. Instead the idea is to take advantage of the naturally occurring large volumes of water likes seas or lakes, which lead to the developments of detectors like ANTARES [34] and the currently in construction KM3Net [35]. On the other hand an alternative is to build the a detector on the naturally occurring deep Antarctic ice. This idea lead to the construction of the **Antarctic Muon And Neutrino Detector Array (AMANDA)**, which was eventually succeeded by the **IceCube Neutrino Observatory** [36].

3.1.1 Properties of Ice at The South Pole

There are many advantages to the choice of the ice to house a neutrino detector. The Antarctic ice in particular offers a very clear medium for the detection of Cherenkov light while also being very still in comparison to other possible mediums like water. However, ice does trap dust and air bubbles which offer scattering and absorption centers. Nevertheless, the fact that the detector is positioned below over 1 km of ice makes it so the pressure of the glacier minimizes the presence of these air bubbles.

The interaction Length for the absorption of light is significantly smaller in the Antarctic ice when compared to water, however it still takes place. As well as the scattering on dust particles. Both of which are not uniform throughout the volume of the detector. Because of this reason, the IceCube Collaboration has studied these properties by using the built-in flasher in the Digital Optical Module units to test and parameterize the properties [37] with the results presented in Figure 3.1. This Figure characterizes the effective scattering and absorption for light at 400 nm throughout the depth of the IceCube detector positioned between 1450 and 2450 km of depth. Although the distribution is fairly stable, the existence of a peak is to be noted at around 2000 meters of depth. This peak in absorption and scattering can be attributed to a layer of trapped dust observed in the ice [38] accumulated in the history of the formation of the ice at approximately 65000 years ago.

The ice is the medium in which detections for IceCube take place, and the proper understanding of its properties helps contextualize events, as well as create simulations that are realistic to observations. Because of this, IceCube continuously performs studies refining the estimation of the properties studied in this section

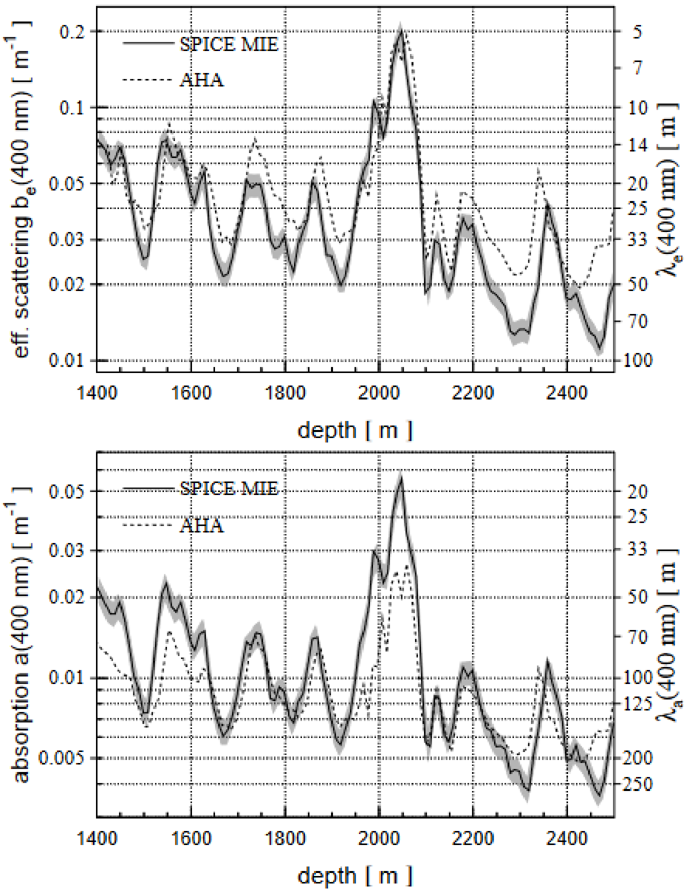


Figure 3.1: Effective absorption an scattering coefficients for light at a wavelength of 400 nanometers. The solid line represents the *SPICE MIE* model for the ice obtained by using the IceCube DOM flashers, with the shaded area covering the uncertainties. Additionally, the dotted lines represent the Additional Heterogeneous Absorption (AHA) model from the ice description obtained years prior in the context of the AMANDA detector. Image taken from [37].

3.1.2 Structure of The IceCube Detector

The IceCube Detector is located on the geographic South Pole. The IceCube Laboratory (ICL) is found on the surface of the ice and is dedicated to housing the South Pole system that handles data acquisition and filtering software. The full operational detector then consists of three distinct parts: the in-Ice Array, DeepCore, and IceTop [36].

The main section of the detector is the **In-Ice Array**, 78 vertical strings are deployed from the surface of the ice into a depth of 2500 meters. The effective detector volume is contained between 1450 and 2450 meters below the surface. In this range, each of the strings includes 60 Digital Optical Modules (DOMs), which are the main unit of the detection in IceCube and responsible for recording light pulses from events. Strings are separated by 125m from each other, and the DOMs in each string have a vertical spacing of 17m.

Then, 8 additional strings are deployed as part of **DeepCore**, a sub-array within IceCube designed for the detection of lower-energy neutrinos (5 GeV to 100 Gev). The DeepCore array of DOMs are placed in a more condensed range, the bottom 50 DOMs in each string are placed with a vertical spacing of 7 meters from each other between 2100 and 2450 meters. then the remaining 10 DOMs are place above a depth of 2000 with a spacing of 10 meters and act as a veto. DeepCore's design is optimized for atmospheric neutrino oscillation experiments among other physical problems [39].

The final component of the detector system is **IceTop**, a cosmic ray air shower array located at the surface of the ice. This array is composed of 81 stations, each consisting of two tanks filled with a height of 0.90 meters of ice. Each of the tanks also include 2 DOMs for detecting Cherenkov radiation produced by incoming charged particles. The main purpose of the IceTop array is to detect primary cosmic rays in the PeV to EeV range, although this is limited by the reduced size of the detector. Additionally, it also serves as a veto to reject vertical atmospheric events that reach the in-Ice array.

The sketch presented in Figure 3.2 shows the different parts of the IceCube detector, providing a full overview on the system described in the previous paragraphs.

3.1.3 Digital Optical Modules (DOMs)

The units for detection deployed throughout the length of the strings in the IceCube array are called **Digital Optical Modules (DOMs)**. These electronic structures where developed by IceCube to house the full system for receiving photons and communicating signals. The Modules are housed inside a spherical shell of 13" in diameter. The shell is made of pressurized glass, that can withstand the pressures at over 1500 meters of depth in ice[36]. An image depicting the structure of the DOM is presented on Figure 3.3.

Each DOM houses a downward-facing photo-multiplier tube (PMT), 10" in diameter, developed by Hamamatsu. Each of these PMTs are capable of detecting light in the range of 300 to 650 nanometers of wavelength, with an efficiency of 25% at 390 nanometers. When a photon is received by the PMT it propagates through the neck of the PMT where the amplification

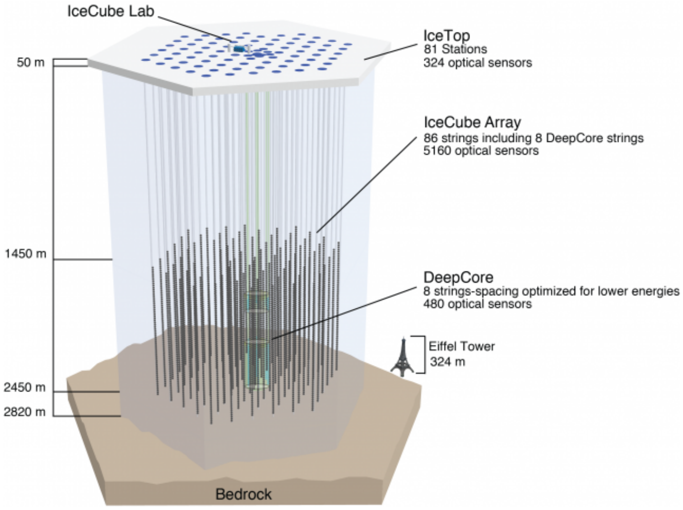


Figure 3.2: Schematic of the IceCube detector, depicting the In-Ice Array, DeepCore, IceTop as well as the IceCube Laboratory. Taken from [36].

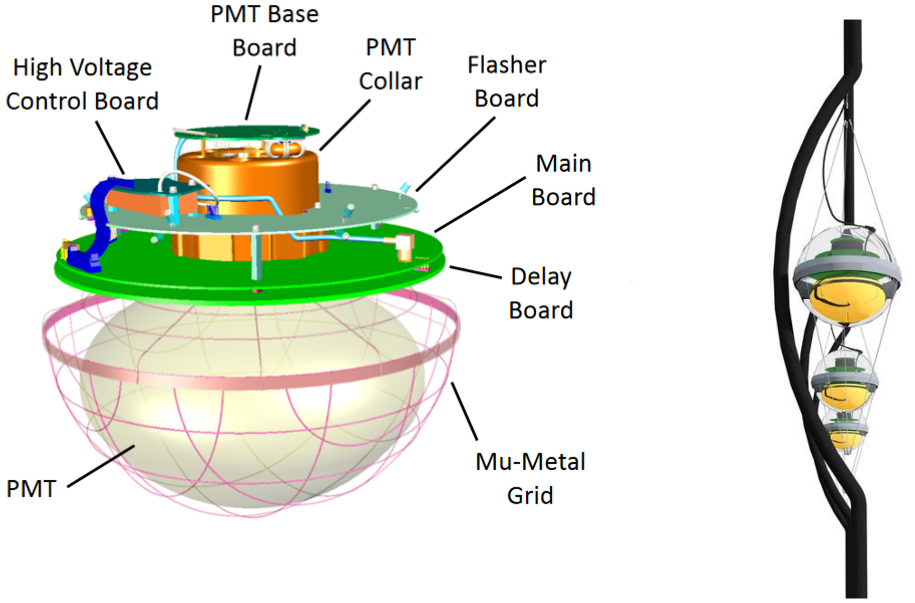


Figure 3.3: Schematic of the Digital Optical Module (DOM), Left: The internal structure, including the photo-multiplier tube (PMT) and the electronics systems. Right: Schematic of how DOMs are deployed along a string.

process produces a distinct signal.

The DOM also houses a system of electronics, including a High Voltage system, a Main

Board, a Delay Board, among others [36]. The High Voltage system is responsible for setting and reading the voltage to and from the PMT. The Main Board on the other hand is in charge of the main operations in the DOM: control of the components, calibration, communication with other DOMs and the data acquisition system on the surface; as well as sampling and digitizing the PMT waveforms.

When the PMT signal crosses a threshold corresponding to 0.25 photo-electrons, the system records a waveform for photons arriving up to 6.4 microseconds later [40]. At this point, the DOM defines this as a ”hit”. When a DOM receives a hit, it communicates through cables to the adjacent modules on the string to check for local coincidences. If a hit is detected on neighboring or next-to-neighboring DOMs within 1 microsecond, this is defined as a hard local coincidence and the full information of the pulse waveform is sent to the surface.

The system additionally contains a Flasher Board capable of generating light for calibration purposes, measuring PMT positions, as well as checking the properties of the Antarctic Ice. The electronics boards are set in place, being bolted down to the PMT structure which is secured in its position inside a silicone gel that surround the entire photocatode area, mechanically supporting the PMT. This gel's refractive index matches the glass sphere, so that the reflection when light passes from the glass into the gel and the PMT is minimal into the gel is minimal. Additionally a Mu-Metal cage surrounds the PMT to ensure magnetic fields do not interfere with detections. The whole internal system as a whole is put in place to ensure a fixed system that works properly for detections.

Approximately, 5500 DOMs were delivered to the South Pole, of which 5484 have been deployed in the detector. From this, 55 modules have failed due to malfunctions early in the deployment process, possibly due to water leaks or freeze-in damage. After deployment, 32 more DOMs have failed during regular operation, with no discernible pattern except for the failure occurring during non-standard events such as a power outage, a filesystem upgrade or during calibration runs. At a mean failure rate of 4 DOMs per year (effectively less than 2 per year in recent time), around 97% of the DOMs are expected to be functional after 25 years of original deployment in 2006 [36].

As explained in this section, the DOMs represent the essential detection unit of the IceCube Observatory. Events will be recorded in the system as a collection of signals in many DOMs in coincidence. This signals in coincidence then take many shapes and signatures depending on which modules are activated. Therefore, identifying the characteristics that form these different event signatures is key to understanding how IceCube detects a variety of physical events.

3.2 Characterization of Events in IceCube: Types and Signatures

IceCube records the detection of neutrino as events. These events are defined by a grouping of signal pulses received by the DOMs in coincidence. In order to properly characterize these events, we have to understand how IceCube looks at events in the sky. The direction of events passing through the detector are characterized by the The **zenith** angle, θ, measured from the detectors vertical axis, pointing out of the surface of the Earth at $\theta = 0$, and the **azimuth** angle, φ which defines the projected direction on the horizontal plane of the detector.

These definitions then allow us to define two categories important for IceCube detection: up-going and down-going events. An up-going event is, as the name describes, an event passing through the Earth and towards the surface of the ice through the detector. Typically described by a zenith angle $\theta > 82°$. Because up-going events detected in IceCube have to pass through the Earth, the muon background is mitigated as the muons get absorbed into the Earth and events in this direction can offer clearer reconstruction of high-energy astrophysical events.

Conversely, down-going events come from the Southern Sky and enter through the surface of the ice, with zenith angles $\theta < 82°$. Because particles coming from this side have to traverse less distance inside of the planet, the muon background can enter the detector. Therefore, events coming from this part of the sky are subject to more complex analysis to overcome this background and offer clearer reconstructions.

In IceCube, there are about 100,000 neutrinos detected each year. Out of these, typically 10 come from astrophysical sources, with the rest forming the neutrino background coming from every direction. For every neutrino detected, there are approximately 1 million muons detected, the majority of which come from the Southern sky as explained previously.

A simplified graph depicting up-going and down-going events is offered in Figure 3.4

3.2.1 Types of neutrino events and signatures in the detector

Neutrinos, being charge-less particles, cannot be detected directly . Instead, they must be observed through the charged products of their interactions. When a neutrino arrives in the Antarctic ice they can interact through different processes that offer distinct detections at IceCube. Charged Current (CC) interactions occur when the neutrino interacts with the particles in the ice, via an intermediary charged boson $W^{\pm}$ producing a charged lepton which is then detectable by the PMT array through Cherenkov emission:

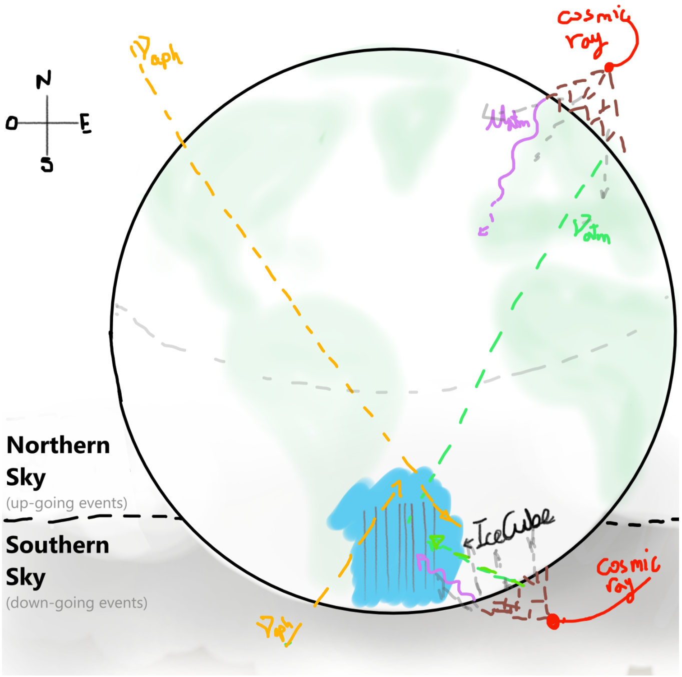


Figure 3.4: Simplified sketch of particles arriving to the detector volume evidencing the difference between up-going events (coming from the Northern Sky) and down-going events (coming from the Southern Sky) In the IceCube detector. Atmospheric muons (purple) ans neutrinos (green) as well as astrophysical neutrinos (orange) are presented to show how this two types of events have different characteristic backgrounds.

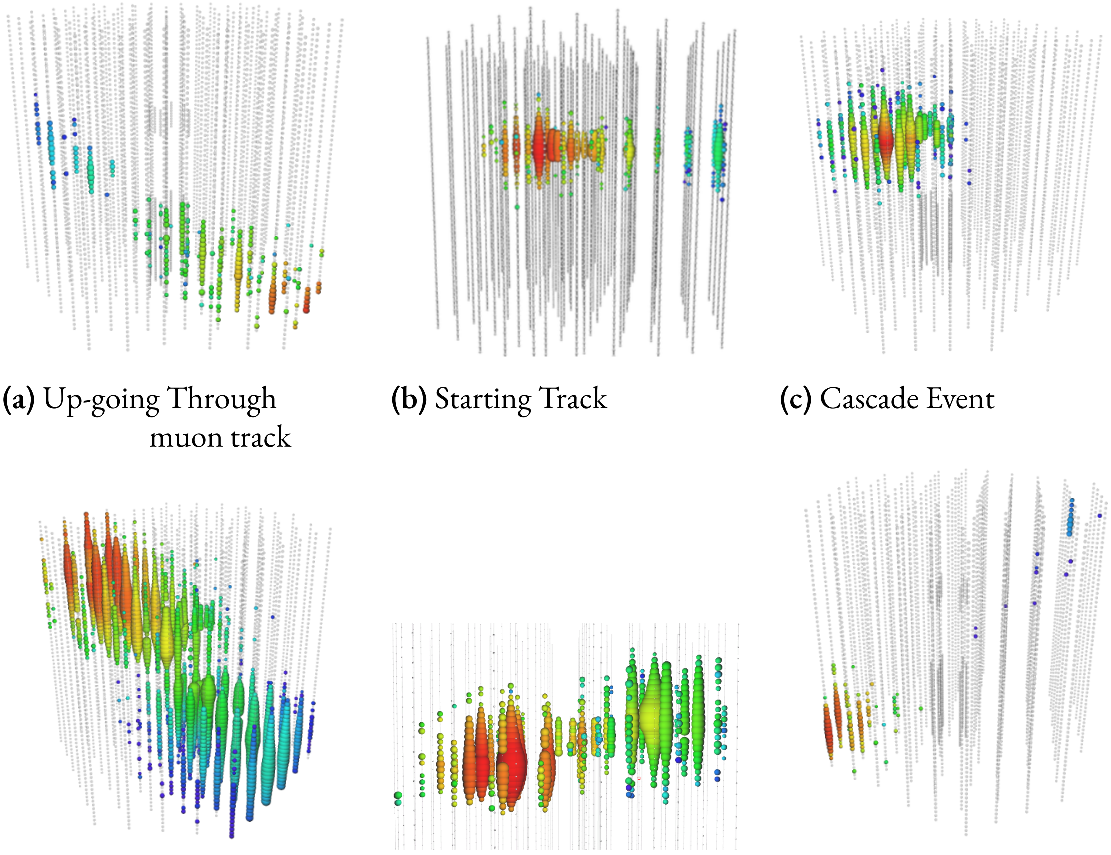
$$\nu_l + X \xrightarrow{W^{\pm}} l + X'$$

Additionally, neutrinos can interact via the intermediate Z^0 boson in what is called a neutral current interaction.

$$\nu_l + X \xrightarrow{Z^0} \nu_l + X'$$

Both of these type of interaction can occur for the three lepton families l =e, μ, τ. At the energies IceCube operates, both types of interactions occur through Deep Inelastic Scattering (DIS), which leads to a portion of the energy being deposited into the components of the interacting nucleons, leading to jets of hadrons that can produce signals in the detector. DIS can occur for both NC and CC interaction, but they can be identified depending on the morphology of the event as detected in IceCube. Several of the typical morphologies are shown in

Figure 3.5 and detailed in the following sections.



(a) Up-going Through muon track

(b) Starting Track

(c) Cascade Event

(d) Muon Bundles from an Air Shower

(e) Tau Double Bang

(f) Two coincident low energy muons

Figure 3.5: Different kinds of event signatures identified in IceCube. Each bubble represents a DOM activated by the event. Sizes increase with the amount of charge registered at each module, while the color represent the time of detection from red (earliest) to blue (latest)

Cascades: Electromagnetic and Hadronic

Cascades can be produced by the decay of the hadronic products from either NC or CC interactions as well as showers produced by the electron or tau leptons produced in CC interactions, where the tau can as well decay into electrons or hadronic jets [1]. The full development of a

[1] For tau at very high energies, IceCube will detect a ”Double Bang”3.5e. The first part of this morphology corresponds to the production of the tau lepton, while the second one comes from the following decay.

cascade is composed from the original interaction as well as any subsequent decays along the development of the event.

Electromagnetic cascades occurs as the charged particles are scattered and deflected in the ice. Because of this characteristic, the light is spread out in a spherical manner. Similarly, hadronic cascades occur as particles decay inside the detector, producing different types of particles, including muons from meson decays. These are not energetically enough to be confused with high-energy events and instead their signature form an important part of the reconstruction of the cascade signature.

Although these types of events develop over some distance inside the detector, by comparison of the full development, they tend to appear spherical with a centered interaction point. This makes it more likely that a cascade event will be contained inside the detector, therefore the full energy of events is usually well registered within the DOMs. This characteristic makes it so that the energy of events is well reconstructed (with an average uncertainty of approximately 10% [41]) Conversely, also because of the shape, the general direction of the event is more difficult to reconstruct, with high uncertainties. Searches for sources using these types of events have to take all of the mentioned characteristics into consideration when performing analysis and show increasingly effective results [42].

Muon Tracks

As opposed to cascade events which are spherical in shape, another distinct kind of events are called track events which can usually be seen developing from an identifiable straight line through the detector. Muons are the only kind of particle able to produce this track morphology due to their small cross section with matter. The In contrast to cascade events, track events offer better angular resolution when compared to cascade events due to the identifiable long lever arm of the event. In fact, the average resolution for this type of events is around 0.6 °[36]. The main purpose of IceCube is to detect muons coming from CC interactions of muon neutrinos, but they can also be produced from the decay of tau leptons coming from the tau neutrino CC interaction, as well as atmospheric muons from air showers, the latter of which creates a significant background for track events.

Because of the quality of the angular resolution in track events, they are excellent to point towards astrophysical sources. Therefore they are the focus of this thesis. There are three different types of tracks that can be identified in the detector depending on their location:

- **Through-going tracks.** Tracks generated by muons produced outside of the detector volume and passing through before decaying outside.

- **Starting Tracks.** For this type of track the neutrino interacts within the detector volume so the detector captures the start of the track. These type of tracks allow IceCube to work as a calorimeter.

- **Contained Tracks**. These type of tracks develop from interaction vertex to final decay inside of the detector geometry. Because most of the muon energy is deposited in the detector, the energy reconstruction of this events is the most accurate.

- **Stopping tracks.** Muons are produced outside of the detector and lose their energy inside of its volume, eventually decaying.

- **Skimming tracks.** These refer to muons that pass near the detector but don't directly enter its volume, the Cherenkov light they produce is however perceived by the modules.

It should be noted that the last two events presented in Figure 3.5 are also produced by muons entering in the detector. Nevertheless, they are unlikely to correspond to astrophysical sources. The first one (3.5d) is a combination of muon events coming from an air shower produced by a cosmic ray proton, producing a full event that corresponds to the background of more interesting events. On the other hand, the other signature (3.5f) corresponds to two atmospheric muons crossing the volume in a short window of time. Although these events are usually too low in energy to survive the data acquisition system, their coincidence can be misconstrued for a full track. This types of possible background events need to be considered in the data acquisition to properly identify interesting events. The process through which track events are filtered is detailed in Chapter 4.

In general, the energy reconstruction for track events is worse than for cascades. An estimate of the energy is made by understanding the loss of muon energy along the tack and obtaining an initial estimate for that. However, because muons coming from outside the detector are produced at an unknown distance, like those of through-going tracks, the energy reconstruction can only offer a lower bound for the neutrino energy. Methods currently in place for energy and angular direction reconstruction are explained in the next section while the rest of this work focuses on the implementation of an algorithm to improve this reconstruction in the context of the RealTime Alert System.

The tracks described in this section are a collection of electrical signals forming a picture that correspond to physical events. But in order to characterize them properly, the system needs to use the information provided by the signals to reconstruct physical parameters. The most important of which are the energy and the angular direction of events.

3.3 Reconstruction Methods for IceCube events

As explained in the previous section, different types of events produce different track morphologies. This occurs as a result of the nature of the particles and interactions involved in the detected events. This also means that depending on the type of event, different processes are used to infer the original physical properties from the raw data in the DOMs. Starting at this section, the work presented in this thesis focuses on events in IceCube that produce a linear track morphology in the detector. Thereby, the items explored will focus on events that are likely produced by a highly energetic muon crossing the detector. With this hypothesis in mind, this section explains different methods currently implemented in IceCube for the reconstruction of angular direction and energy for muon events coming from the detector volume.

3.3.1 Reconstruction of Angular direction

There are many different methods for reconstructing the direction of events based on the signal received. The many different methods currently in place have advantages over each other based on the speed of their implementation for the online reconstruction of events, as well as the precision.

The most basic among this is the so called **LineFit Method**. This methods models the muons central track in a very simple way. Considering a muon that passes a position $\vec{r_0}$ at a time t_0 with a velocity $\vec{v}$, the position at time t can be expressed as:

$$\vec{r}(t) = \vec{r_0}(t_0) + (t - t_0)\vec{v} \tag{3.1}$$

Then, a fully defined track is obtained from the position and velocity parameters of the track. Then, assuming the signal pulses in the DOMs are produced by plane waves, the LineFit algorithm minimizes the square distance x_i between a DOM activated at time t_i and the position of the muon along the track as given by the following expression.

$$\sum_{i=1}^{N} ||\vec{r_i}(t_i) - \vec{x_i}|| \tag{3.2}$$

This method is quick for reconstruction because the above expression has analytical solutions for all the reconstruction parameters [43]. Therefore the determination of the track can be made very quickly. As a final result, this algorithm offers a rough estimate of the direction as well as an estimation of the velocity of the particle. The latter can be then used as a parameter

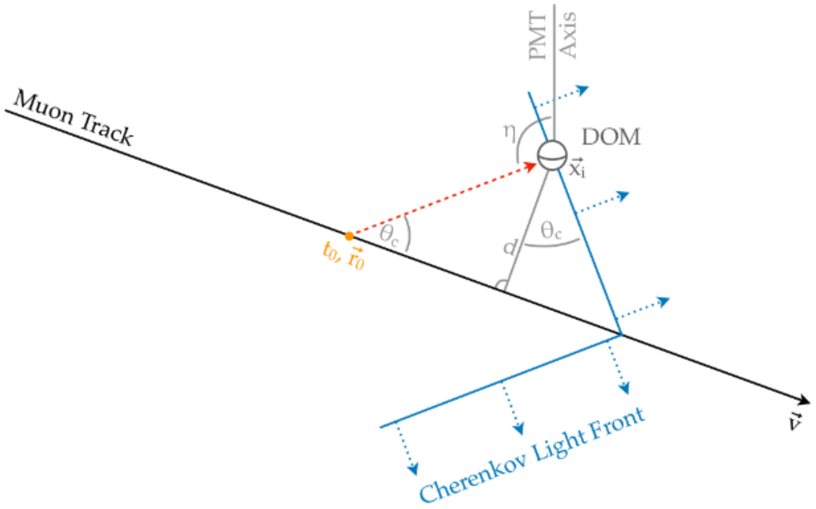


Figure 3.6: Sketch of a cone of Cherenkov light emitted along a muon track at an arbitrary point in space $\vec{r_0}$ and time t_0. Also in the sketch a Digital Optical Module that detects the Cherenkov light at an arbitrary position. Image taken from [44]

for event selection. On the other hand, the angular estimation of direction then acts as as a starting point for more refined calculations of event direction.

In order to obtain a better estimation of the direction, the **Single Photo-electron Fit (SPE)** fit implements a more realistic model of the light emission. According to the Cherenkov model, a high-energy muon traveling in ice at a speed larger than the phase space in the medium produces light in a cone defined by the Cherenkov angle:

$$cos\theta_c = \frac{1}{n\beta} \tag{3.3}$$

where β is the velocity of the muon ($\beta \approx 1$ for relativistic muons, and n is the refractive index in the ice (n ≈ 1.32). Then, along the muon track, t_0 and $\vec{r_0}$ define the time and position at which the muon emits the cone of Cherenkov light. A full sketch can be seen on Figure 3.6.

Taking in mind this simplified model of Cherenkov emission, the algorithm defines the expected time t_{geo} and the t_{res} residual time between this and the actual detected time of observation.

$$t_{geo} = t_0 + \frac{\vec{v}(\vec{r_0} - \vec{x_i}) + dtan\theta_c}{c_0} \tag{3.4}$$

$$t_{res} = t_{obs} - t_{geo} \tag{3.5}$$

With this definition in mind, the SPE method implements a maximum likelihood test for the time residuals to correspond to a specific set of track parameters (Equation 3.6). From the way it is currently implemented, a first hypothesis for the likelihood comes from the LineFit Model. Full details on the minimization function are explored in [44].

$$\mathcal{L} = \prod_{n=1}^{N} p_1(t_{res,i}|\vec{r_0}, \theta, \varphi) \tag{3.6}$$

The SPE method will only take into account the first photon detected at each DOM. This works on the reasoning that the first photon is the least scattered and provides the most quality information. However, because of this, the method has shortcomings at higher energies, where further photons at each DOM provide quality information from the event's structure. In this case the **Multi Photo-electron Fit (MPE)** method is applied by modifying the expression for the Likelihood.

$$\mathcal{L} = \prod_{n=1}^{N} \left[N_i \cdot p_1(t_{res,i}) \cdot \left(\int_{t_{res,i}}^{\text{inf}} p_1(t_{res,i}) \right)^{N_i - 1} \right] \tag{3.7}$$

A further refinement in the reconstruction of angular is performed with the **Spline Multi Photo-electron Fit (SplineMPE)**. For this method, the analytical function for the expression of the parameter p_1 is replaced by interpolating splines[2]. These Splines have been prepared based on simulations of photon propagation on the ice sheet. This method allows the reconstruction to take into account the structure-dependent properties of the ice in an effective quick way, that allows for the methods to be implemented on online reconstructions.

Further improvements to the angular reconstructions algorithms that take into account more refined models of the ice as well as the stochastic energy losses that might affect the shape of the reconstructed track [46, 47]. Some of these further improvements take into account segmented models for the ice interactions for more accurate adjustments or combine different versions of the previously mentioned algorithm to increase speed of reconstruction and optimize the estimation for further parameters.

[2] Splines [45] are piece-wise functions defined by polynomials so that they are continuous up to a certain order. They are very useful to fit and save values for simulations like in this case. Particularly, the reduce memory consumption and execution times.

3.3.2 Reconstruction of Angular Error

Besides the reconstruction of the angular direction, another important parameter to obtain for the physical events is the angular error of said reconstruction *per event.* Besides offering a region around the direction of events to search for point sources, the angular direction offers an important parameter in statistical analysis for determining how well events are reconstructed.

Although a general error estimation for every track could be applied in order to determine this parameter, a proper error reconstruction for tracks needs to take into account the circumstances of each event. For this reason, methods that reconstruct the direction for each event is necessary. There are three main methods currently in place for the reconstruction of angular error.

The first method involves the application od **Cramer-Rao's inequality.** [48, 49] Taking the parameters from the reconstructed track of the muon event, this method states that the covariant matrix is bounded by the inverse of the Fisher Information matrix I$(\vec{x_i}, \vec{x_j})$, where $\vec{x}$ refer to the reconstructed track parameters $\vec{x} = (r_0, \theta, \varphi)$:

$$I(\vec{x_i}, \vec{x_j}) = -\langle \frac{\partial}{\partial \vec{x_i} \partial \vec{x_j}} \log \mathcal{L}(\vec{x}|t_{res}) \rangle \tag{3.8}$$

$$\text{Cov}(\vec{x_i}, \vec{x_j}) \geq I(\vec{x_i}, \vec{x_j})^{-1} \tag{3.9}$$

the diagonal elements of the covariance matrix offer estimates for the variances for the zenith and azimuth $(\sigma_\theta^2, \sigma_\varphi^2)$. In a way that a circularized error can be effectively defined [44].

$$\sigma = \sqrt{\frac{\sigma_\theta^2 + \sigma_\varphi^2 sin^2\theta}{2}} \tag{3.10}$$

Fortunately, the covariance matrix elements can be determined from analytical expressions. Since there is no need of numerical minimization, it is the fastest method of approximating the angular error. However, it is also the least precise.

On the other hand, a more detailed approach is offered by the **Paraboloid Method.**[50] As the reconstruction of the direction of the muon event is performed by a likelihood function, the shape around the likelihood extrema of said function is expected to be related to the angular resolution. By modeling the function as a Gaussian for which the standard deviation can be estimated by [44]:

$$-log\mathcal{L}(x \pm \sigma_x) = \mathcal{L}(x) + 0.5 \tag{3.11}$$

By using the zenith and azimuth of the parameters as definitive in determining the shape of the likelihood, a paraboloid fit of the extrema can be used to determine the angular error. In order to obtain this fit, a sample of 24 points are taken. Then after minimization the error taken is reported as

$$\sigma = \sqrt{\frac{\sigma_x^2 + \sigma_y^2}{2}} \tag{3.12}$$

This method yields the best estimation of the error, making it the choice by excellence in offline analysis. However, this requires a number of numerical minimization that make it improper to be implemented perfectly in online analysis.

Finally, the last method takes a **Bootstrapping** approach [51]. This kind of method takes the parameters that characterize the track of the event and randomly re-sample some of the parameters requiring the total charge recorded by the DOMs to be the same as the original event, then reconstructing a nw track from this re-sampling. This process is repeated many times, so that the system obtains a sample of tracks defined by the sub-sampling of a single event. Finally, the median separation between the re-sampled tracks and the original can be taken as an estimation of the angular error.

3.3.3 Reconstruction of Energy

In the previous section, we evidenced the algorithms for the reconstruction of angular direction of events. Now, we turn our attention to the energy of the events. A well-reconstructed energy parameter can be useful as a discriminator between background and signal events. Additionally, an event with a well-defined energy is useful to study the spectrum from neutrino sources, making it a fundamental part of the analysis.

Nevertheless, it should be acknowledged that signatures detected in the volume of IceCube correspond to a part of the event. Neutrino interactions are likely occurring outside of the detector volume at an unknown distance, generating through-going tracks from which only the muon energy in the detector can be measured, at the very least offering a lower bound on the neutrino event.

Events passing through IceCube produce near-constant light emissions which scale linearly with the energy of the event. Thanks to this, a reconstruction method can be devised starting from the light yield of reference simulated events (usually of around 1 GeV), scaling from those through template functions that take into account the detector response and the ice properties [52]. If an event produces Λ photons per unit energy, the number of detected photons is expected to follow a Poisson distribution with a mean of $\lambda = \Lambda E$. Therefore, the likelihood that an event of True Energy E produces k photons can be modeled as:

$$\mathcal{L} = \frac{\lambda^k}{k!} e^{-k} \tag{3.13}$$

by adding noise depositions and other possible contributions, $\lambda \rightarrow \Lambda E + \rho$, where ρ is the expected number of noise photons. The function of the likelihood to maximize becomes:

$$\log \mathcal{L} = k \log(E\Lambda + \rho) - (E\Lambda + \rho) - \log(k!) \tag{3.14}$$

The maximization of this expression does not have an analytical solution for all cases, in case different algorithm have to be implemented while considering the most appropriate models for the template function Λ.

A first approximation comes from the **MuEx (Muon Energy Estimation)** method. This considers the muon to emit Cherenkov light in an uniform manner along its track. Then, the only dependence between the number of detected photons k at any DOMs is to the distance from the track. Near to the track, the dependence will go like 1/r, due to a dominance of absorption processes over scattering ones, while at larger distances, the behavior is approximated and made compatible with the lower distance regime by use of simulations and data benchmarks [52]. The model then consists on simply maximizing the expression for the likelihood taking into account these models of production. The accuracy of the method then also depends on taking into account other possible stochastic emissions.

An alternative reconstruction is the **Truncated Energy** method [53]. In order to estimate the energy, this technique starts by segmenting the energy track. Based on the principle that the mean energy loss of the muon $\langle dE/dx \rangle$ is directly related to its energy, the method will calculate the energy at each of its segments to estimate the full energy of the muon track. The segments are defined at approximately 120 meters to be similar to the distance between the DOMs. The cuts are designed to include the modules with the most robust estimates of the energy, meaning not the one closest to the track and definitely excluding those further than 80 meters . The track hypothesis obtained from SplineMPE reconstructions are used to define this tracks and then the likelihood shown previously is maximized for every segment. Finally, an extrapolation is made for the muon energy. By coming from different segments, this method is robust with respect to the stochastic variations happening from segment to segment.

This **Truncated** method is very effective at providing accurate energy readouts, with the disadvantage of requiring large enough tracks to be consistent. On the other hand, the **MuEx** method is consistent at every track length. Because of this, it is usually presented as a benchmark in studies that cannot make the quick distinction between tracks, particularly RealTime studies.

Additionally, in recent times there have been methods that take advantage of machine learning algorithms to reconstruct the energy based on the training with simulated events. A widely implemented example of this is presented in [54]. In that work, they explored the use of Convolutional Neural Networks (CNNs) on energy reconstruction for Cascade Events (For a brief explanation on CNNs, see Section 5.1.1). CNNs were first applied on image data and are able to offer predictions based on the properties of pixels of data and those of their surrounding pixels. IceCube Events can be extrapolated into an image's framework by considering DOMs as pixels with properties representing the charge of pulses as well as time of arrival.

This Neural Network method takes in data from an event and divides it into arrays representing the In-Ice Array and the columns of DeepCore. To normalize the hexagonal structure of the IceCube detector, the In-Ice array is approximated to an orthogonal structure with padding corners set to values of zero. Then, the input consists of arrays containing information for all the DOMs, including the padding ones. the features that express this information are for each DOM:

1. Total DOM charge
2. Charge within 100 ns of the first pulse.
3. Charge within 500 ns of the first pulse.
4. Relative time of the first pulse.
5. Time at which 20% of charge is collected.
6. Time at which 50% of the charge is collected.
7. Time of last pulse.
8. Mean Time of pulse arrival weighted by charge.
9. Standard Deviation of the mention Mean Time of pulse arrival.

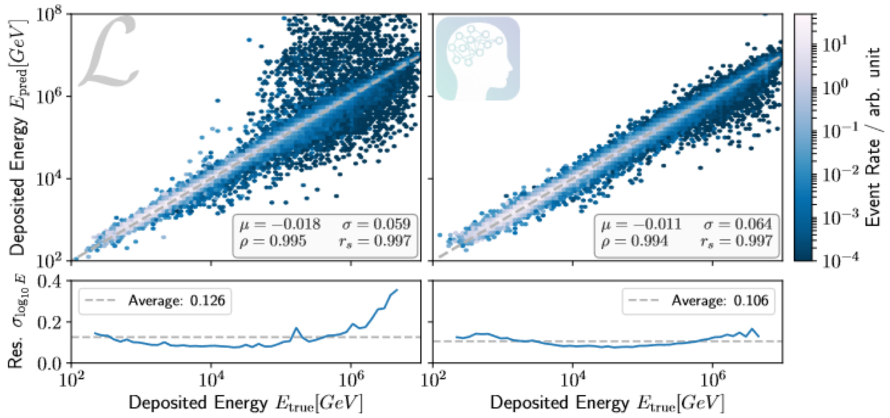


Figure 3.7: Correlation between true deposited energy of simulated events and the energy reconstruction offered by standard methods (left) and the CNN method (right). Residuals are presented showing how well the reconstruction adjusts to the data. Plots taken from [54]

These 9 parameters are used to account for the full picture of events in a wide energy range, while keeping the input structured as a CNN requires. Then, after the CNN is trained it is ready to perform the task of Energy reconstruction. Results from implementing trained networks for energy reconstruction in cascade events are presented in [54] (As seen on Figure 3.7). These results show the performance of this kind of algorithm when compared to traditional likelihood methods on cascade events. The Neural Network shows a more robust performance on reconstruction with less outliers. It should be noted that some bias is observed in the edges of the region of estimation, which can be attributed to biases on the boundaries of the dataset.

Additionally,it should be noted that this method improves substantially on the runtime of energy reconstruction methods. The CNN has a median runtime of 0.024 seconds per event when compared to standard reconstruction methods at 10s making them very interesting for applications to online data-taking [54].

With this idea in consideration, the next chapter explores the structure of the RealTime Alert System, which is a framework in place for filtering and reconstruction of events during online data acquisition. Particular emphasis will be placed on the algorithms currently in place in the system, focusing again on muon track events; as well as the timing requirements presented for such a system.

4
The IceCube RealTime Alert System

In Chapter 2, We have explored the discipline of Multimessenger astrophysics, explaining how an approach of this style is useful at providing rich information about the astrophysical sources and processes. However, in order for this approach to be effective, the scientific community capable of detecting the different messengers needs to be observing sources in a similar time window. In this context, IceCube serves as an excellent starting point from which to prompt observations. The IceCube detector is sensitive to events from the full sky while being consistently operational and taking data at a rate of 99.8% of the time [55]. This allows IceCube to be constantly on the lookout for events of note.

Taking advantage of this, IceCube can serve as a system that triggers follow up searches for astrophysical phenomena from observatories all over the world. For this purpose, the collaboration has set up a RealTime Alert System that quickly identifies these significant events during data acquisition and reports them as alerts for the scientific community. IceCube sends different types of alerts through public and private channels which are mainly sent through the Astrophysical Multimessenger Observatory Network (AMON) as notices in Gamma-Ray Coordinate Network format (GCN)[56, 57]. These notices include the energy, direction, angular uncertainty and probability of the event potentially coming from an astrophysical source. All of which need to be reconstructed effectively and quickly after data acquisition.

This work develops specifically in the context of the Gamma-Ray Follow Up (GFU) Program [58]. This program is based on the fact that Astrophysical sources of gamma rays are believed to be sources of high energy neutrinos as well. Therefore, GFU searches for neutrino flares whose direction corresponds to the location of sources of gamma rays from known catalogs. The GFU selection processes large amounts of data during data acquisition. However the number of full alerts per year is considerably small, at approximately 10 alerts per year. These alerts are sent privately to Gamma Ray Telescopes that form part of the program which include IACTs like MAGIC [23, 24], VERITAS [26], and H.E.S.S. [25].

The datasets we use for energy reconstruction in this work comes from the context of these systems, therefore this chapter describes the data processing that occurs in the context of the RealTime Alert system. In the following sections I present the architecture of the system and the steps for filtering and reconstruction of events that produce the samples a energy reconstruction system would have to run on.

4.1 Structure of the RealTime Alert System

This section provides an outline of the processes involved in the RealTime alert system, as visualized on 4.1. at the start of the process is the signal produced in the PMTs at each of the DOMs, each of this digitizes the waveforms received in order to be processed. The data in this form is then passed through trigger processes that group electronic signals by physical events. Then, passes through filters that select data by quickly reconstructing parameters in a coarse manner, reducing the dataset progressively until they arrive to the final GFU filter. The end result of which is a sample with well reconstructed parameters. If an event or a cluster of events then crosses a significant threshold, an alert is sent.

The process then aims to reduce quickly the enormous volumes of data flowing through the IceCube detector by identifying well-reconstructed muon track events. The triggers and filters in place are detailed in the following sections.

4.2 Level 1 Trigger

The first step in the real time system is the **Level 1 Trigger** that aims to select detected events by identifying groups of coincident pulses on nearing PMTs. When a pulse is perceived in a specific DOM, the internal software of the module checks for the "Hard Local Coincidence" (HLC) condition[36]: If a signal is detected on neighboring or next-to-neighboring DOMs within $\pm 1\,\mu$s of the pulse, the hit satisfies the condition and the full digitized waveform is passed to surface computer. Otherwise, it is considered as noise and only the amplitude of the is passed.

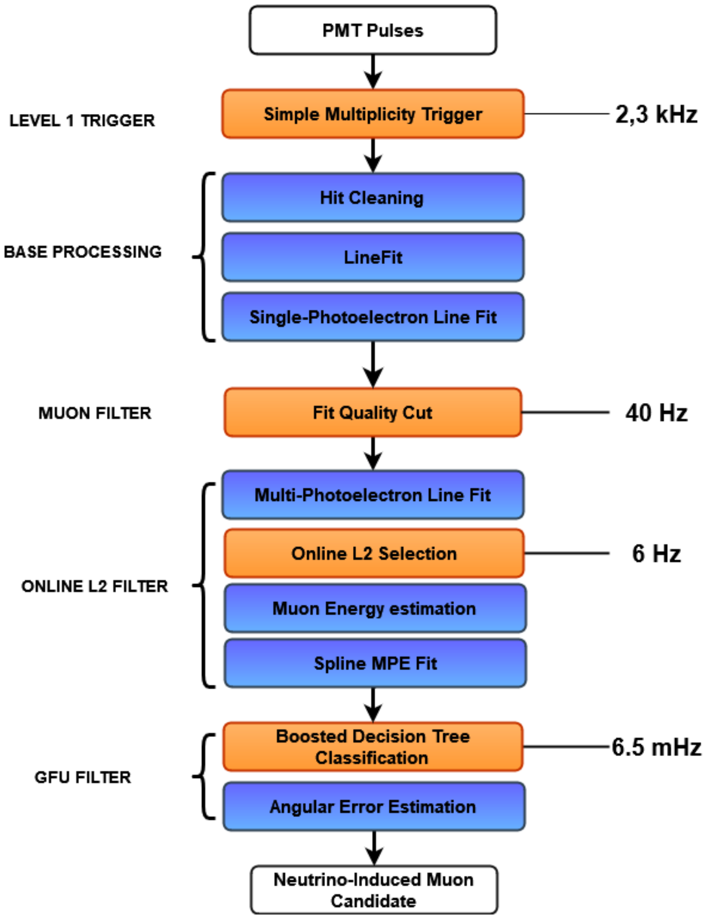


Figure 4.1: Scheme of the procedure in place for real time event selection and reconstruction. The algorithms in place occur at different stages that flow in the direction following the arrow in the graph. Blue boxes correspond to reconstruction steps while orange boxes correspond to filtering steps. The resulting approximate data rate after each filter is presented on the right. This chart has been adapted from the one presented in [44].

This information is then passed to the **Simple Multiplicity Trigger (SMT-8)** which checks for at least 8 HLC hits during a sliding time window of 5 μs. If this condition is met, a time window for data acquisition is started, whose end comes once the sliding time window does not contain HLC hits. This gets combined with other trigger time windows into a global trigger time window with the hope of identifying the full shape of a physical event. The hits recorded during the time window are then passed to base processing as Data Acquisition Events (DAQ Events). The median trigger rate for these events is quite large (2.7 kHz) at which point they need to be processed to properly separate significant events from the background

The previously described trigger condition is quite conservative and aims mainly at eliminating noise from the data. Because DAQ Events are accumulated in time, they might correspond to multiple particles detected in the same window. These events need to be separated using a **Trigger Splitting** method. By taking information from the SMT-8, the system identifies single events by identifying time separations between clustered events. Because this trigger splits events in time but not space, events in different parts of the detector occurring at the same time are still processed as a single event. This is inappropriate for event reconstruction and should be split further in the RealTime reconstruction.

4.3 Base Processing and Muon Filter

After the first trigger is done processing, the system still has a large input of events shall be further processed by analysis in IceCube. For the purpose of the Alert system, the amount of events needs to be trimmed down and event parameters need to be reconstructed. To satisfy these requirements the next step in the system's architecture is **Base Processing**. The main objective at this point is to work with the hypothesis that the detected events correspond to a track morphology to reconstruct the direction from which the event originated.

At this point the algorithm executes the simplest methods outlined in section 3.3.1, the **LineFit** which then feeds the initial seed of the **SPE** method. As a result, the system identifies the direction parameters from the reconstructed track as well as the maximized likelihood that it corresponds to the physical event. Using this information, the system needs to reduce the volume of data by selecting the events which most likely correctly correspond to a muon track. This is what is called the **Muon Filter**.

If the reconstructed direction indicates that the event is a candidate for an up-going muon [1], the likelihood value is used as a measure of the quality of reconstruction to discriminate shower-like events. For down-going candidates, the event rate is larger due to the volume of the background atmospheric events. Considering that the astrophysical neutrino spectrum is harder than the one for atmospheric muons ($\gamma_{\nu}^{(aph)} \approx$ -2 vs $\gamma_{\mu}^{(atm)} \approx$ -3.7), the filter selects events based on the total integrated charge of the track (Q_{tot}), depending on declination. The full criteria is

[1] Reminder that an up-going muon arrives from the Earth core and up the detector while a down-going muon enters the detector from the top of the detector. (See Section 3.2)

as such:

$$\begin{cases} \frac{\log \mathcal{L}}{N_{ch}-3} \leq 8.7 & \text{for } -1.0 < \cos\theta \leq 0.2, \\ \log Q_{tot} > 3.9\cos\theta + 0.65 & \text{for } \quad 0.2 < \cos\theta \leq 0.5, \\ \log Q_{tot} > 0.6\cos\theta + 2.3 & \text{for } \quad 0.5 < \cos\theta \leq 1.0, \end{cases} \tag{4.1}$$

At this point, the rate of events is reduced to 40 Hz. These events can be sent daily through satellite links to data warehouses in Madison, WI where they can be used for offline analysis. Additionally the rate at this point is low enough that more complex operations are able to be performed as part of the real system.

4.4 Online Level 2 Filter. Reconstruction and Filtering

After the Muon Filter, the system has been able to identify muon tracks with a reasonable rate for data transfer and offline analysis. However, the RealTime studies taking place at the South Pole require further filtering and reconstruction. This is the work that occurs as part of the **Online Level 2** Filter. It receives this name being the next step after the Level 1 Trigger and Processing while also occurring during online data acquisition.

As mentioned in Section 3.3.1, the **SPE** fit performed before the Muon filter is effective at a coarse track reconstruction. With the results obtained from this fit as a new first seed, the **MPE** algorithm runs over the reduced dataset to better reconstruct the physical tracks of the events.

This new determination of event direction is enough to provide a more stringent selection (**Online L2 Selection**). A similar type of cuts is applied as during the Muon Filter, but with a more strict criteria as detailed in [59]. With this cuts applied the event rate is reduced from 40Hz to 6 Hz. At this point the direction is finally reconstructed using the **SplineMPE** algorithm which is very efficient at the current volume of data.

During this section, the Energy is reconstructed by using the **SplineMPE** hypothesis track using the **MuEX** and **Truncated** methods explained in section 3.3.3. Finally, at this stage the full reconstructed event is passed into the final steps of the RealTime alert system, called the **GFU** filter because of it being developed traditionally for the Gamma Follow-Up program, but now applied more generally as a selection in IceCube for different analyses.

4.5 Gamma-Ray Follow-Up (GFU) Filter

At this stage of the Online system, the filtered sample consists of events reasonably expected to correspond to muons passing through the detector. However, the selection at this point will still include of a majority of misreconstructed down-going particles produced in air showers. This events need to be filtered out by more sophisticated algorithms. The so called **Gamma-Ray Follow-Up (GFU) Filter** aims at performing this task. After this filter is applied the final reduced rate of events goes down to 6.5 mHz or around 200000 events per year and will consist in its majority of well reconstructed significant track events.

The main filtering mechanism applied during this selection consists of the implementation of a Boosted Decision Tree (BDT) algorithm. This type of algorithm aims at classifying data by implementing a set of gradually more specific binary decisions based on an elements characteristics, in the end assigning a final value, based on the decisions starting from the stem of a tree and going to the most outward leaves. The *Boosted* part of the name refers to the process of gradient boosting, that allows the tree to make errors in their predictions. Then, the algorithm trains a new tree that aims to solve the discrepancy from the previous tree. This occurs in a sequential manner until a series of trees is combined to offer final classification data.

In the case of the GFU filter, the BDT takes the events from the previous filters and uses a set of different variables from the reconstruction to assign to each a value of probability that the event is effectively a signal. This is then used to automatically make the decisions that filter out the events. It should be noted that the implementation of the BDT is different for up-going and down-going events, taking into account the volume and characteristics of each grouping of events.

4.5.1 The Northern Sky (up-going events)

The northern sky is defined in IceCube as the part of the sky from where up-going events are coming from. Remembering that up-going events are those that pass through a significant section of the planet before interacting with the detector, this filter treats them with a similar cut as the Online L2 Filter (with a zenith angle of $\theta > 82°$). In this section of the sky, the background of events which need to be excluded from the sample consists of down-going events not reconstructed properly, and misidentified cascades. And therefore the following variables are used as classifiers on the BDT [44]:

1. **Goodness of Track Likelihood**. The Likelihood maximized from the SplineMPE angu-

lar reconstruction fit is normalized to the number of modules and then used as a measure of compatibility for the event to fit a well-reconstructed track morphology.

2. **Shape of the Likelihood**. By using the Cramer-Rao method of angular resolution, the system obtains a measure of the second derivative of the angular direction likelihood. This provides a measure of how well reconstructed the maximum is.

3. **Speed of the Particle**. The LineFit algorithm used during Base Processing offers a coarse reconstruction of the speed of the particle. A muon's speed should not be significantly different than the speed of light.

4. **Angle difference between the LineFit and SplineMPE**. Well reconstructed events should be robust to increasingly effective methods of angular direction reconstruction. Although the SplineMPE provides a more accurate reconstruction it should not offer results drastically different from the first LineFit.

5. **The Split Fits**. At this point the system could have reconstructed as track events the incidence of two muons in coincidence in different parts of the detector. To eliminate this, the pulse series is split and then reconstruction algorithms are applied to the splits. The directions from this events shouldn't be significantly different from each other or the full track reconstruction for an event corresponding to a single muon.

6. A **Bayesian Likelihood** method is used to consider down-going events in the formula for track reconstruction. This modified likelihood being to different from the original indicates the possibility of a misreconstructed down-going event.

7. **The Center of Gravity.** Tracks that are only detected in a small part of the detector may be improperly reconstructed. Then a useful discriminator is the center of gravity, calculated based on the position of hit DOMs and deposited charge. The radial and depth components of this center of charge are then used as part of the BDT to help discriminate events too close to detector borders.

8. **Number of direct hits**. A direct hit is defined as one where the residual time t_{res} between the expected photon arrival and observed arrival time to a DOM is within a short time window. Then the number of direct hits for a track should be above a minimum for a track to be considered well-reconstructed

9. **The Direct Length**. Simply calculated as the distance between the first and last direct hit. Is a measure of the track's lever arm

10. **Smoothness of Direct Hits**. Additionally, direct hits should be distributed smoothly along the track's length. So a measure of the deviation of the hits of the track is used as an additional measure.

11. **Distribution of hits**. The general distribution of hits along a track can also be used as a measure of the quality of reconstruction. One of this measures is the **Empty Track Length**, defined as the maximum distance along the track for which no hits are observed nearby and should be small for a track corresponding to a single muon. Furthermore, a **Separation Length** is defined for the track as the distance between the center of gravity for the first and last quarter of hit DOM, which should be long for full tracks. Finally, a measure of the average distance of the hit DOMs to the track is calculated and weighed by the charge deposited. This is useful as the most charge should be deposited near to the track.

Before passing to the BDT discrimination, early cuts are made based on this variables, as well as the values of previous reconstructions. Then the BDT is applied on them. It should be noted that this BDT has been previously trained on a sample of well-defined simulated events.

4.5.2 The Southern Sky (down-going events)

When dealing with down-going events, a different background of events should be taken into account. As explained in Section 3.2.1, muons coming from air showers are less mitigated by traversing the inside of the Earth and therefore produce muon bundles in the detector which can be reconstructed as tracks during the previously implemented process. A possibility to implement a veto could be considered to eliminate this kind of events. This possibility nevertheless risks eliminating a number of significant through-going tracks and rely heavily on a small number of DOMs working effectively at all times. Instead, to separate these events a number of different variables are applied as discriminators for the BDT [44].

1. **Northern Sky Variables.** The system of variables implemented for the Northern sky is reliable to remove the same kind of misreconstructed events in the Southern sky so they are re-applied in this analysis.

2. **Time Residuals.** Muon bundles coming from air showers are often reconstructed as tracks used. However, because of their difference in how light reaches the DOMs, the distribution of time residuals t_{res} should differ slightly. The produced light should arrive at earlier or later times than that of muon tracks. Then a likelihood expression can be implemented to establish the probability that an event correspond to a muon track or a bundle.

3. **Light Deposition Profile.** A high-energy signal muon producing a track is likely to suffer stochastic energy losses along its development. However, lower energy muons coming from air showers are less likely to suffer these losses, offering a smoother profile

of Cherenkov light along the reconstructed track. To quantify this difference a **DDDR (Data-Derived Deterministic Differential Deposition Reconstruction)** method is defined that reconstructs the energy losses of segments of the tracks in an approach similar to the Truncated Energy method. Then stochastic losses are observed as segments of high loss followed by dim segments of low energy loss, indicating the behavior of a single energetic muon. A likelihood is then built on this statistic that helps discriminate bundles.

The process of application and training is then applied in a similar manner to that of the Northern Sky. The only difference is that some variables are only used for pre-cuts, to avoid overtraining by increasing the number of training variables.

4.5.3 Angular Error Reconstruction.

As mentioned in Section 3.3.2, the angular error is an important element to report neutrino events. It is a parameter to be used in the search for sources as well as a measure of the wellness of reconstruction. As mentioned previously, the most precise method is the **Paraboloid method.** However, because of the need of minimizing 24 points from the likelihood function, this is a time-consuming process, often too large for the requirements of the online system. On the other hand, the **Bootstrapping** method's efficiency depends on the number of re-samples performed. At 8 steps of resampling, the performance is similar to Paraboloid. However, it is equally slow. Because of this, a compromise is made in implementing the algorithm consisting of 6 six iterations of sampling. However the time of execution is still proportional to the number of hits. The least accurate method explored was **Cramer-Rao**, and it offers a solution when the other algorithms cannot converge in time.

Because of these characteristics, the system applies cuts to the events to decide what error reconstruction method to use [44]. For Low Energy events (MuEx Energy < 4TeV), the Paraboloid method is able to converge in sufficient time and is the preferred one. For higher energies, the Bootstrapping method is applied as long at the number of hit DOMs is below 300 channels. For cases of high energy with a large number of hits, then the reconstruction defaults to the Cramer Rao method. Unfortunately this means that the most energetic events, which tend to be the most interesting, cannot be reconstructed to the best accuracy.

All of these methods offer some bias in the reconstruction of angular errors when compared to the true angular error for simulated events. Improvements to the method have been tested, but require computational times excessive for online implementation. Other methods for the reconstruction of angular error have been tested in different selections such as the use of machine learning algorithms like Boosted Decision Trees [60]. These possibilities are still being explored within the context of the GFU selection and should offer alternatives in the future.

Finally, Table 4.1 shows the execution time of the algorithms in place at the RealTime Alert System. As shown the total reconstruction time per event has a median of approximately 1 second. Once events pass of all the filters they are reported as part of the Gamma-Ray Follow up Analysis sample, consisting of well-reconstructed muon events and used for further analysis.

		Execution Time (seconds)	
Reconstruction Algorithm	**Type**	**Median**	**Max**
Online L2 FIlter			
SPE Fit	Direction	0.04	1.25
MPE Fit	Direction	0.03	2.71
SplineMPE	Direction	0.04	2.68
MuEX	Energy	0.06	0.53
TruncatedEnergy	Energy	0.015	0.04
Cramer-Rao	Angular Error	0.01	0.03
Total		0.40	7.76
GFU Filter			
Paraboloid	Angular Error	0.24	10.6
Bootstrapping	Angular Error	0.17	4.7
Total		0.40	13.18
Total Online Execution		0.99	14.83

Table 4.1: Execution times per event for the algorithms described in this section. The total Execution time for each section takes into account the full time for the filter section to execute beyond the individual algorithms. Table adapted from [44].

4.6 The GFU data sample and analysis

Finally, the system has processed the input of data and offers a set of events corresponding to neutrino-induced muon candidates. This data is sent to IceCube servers and saved within the collaboration as part of the **GFU Sample.** While the data has been cleaned in the process, there is still a significant background of events. The distribution of the different types of events can be seen in Figure 4.2 based on simulated samples. It is evident that the composition of events is different in the Northern and the Southern Sky. Because the algorithm for selection of up-going events focuses on eliminating misreconstructed events, the sample on this region is composed exclusively of neutrino events, mostly muon-neutrinos. On the other hand, since the system in place for the Southern Sky selects events at higher energies, atmospheric neutrinos are suppressed, and the sample consists of mostly atmospheric muons in this region.

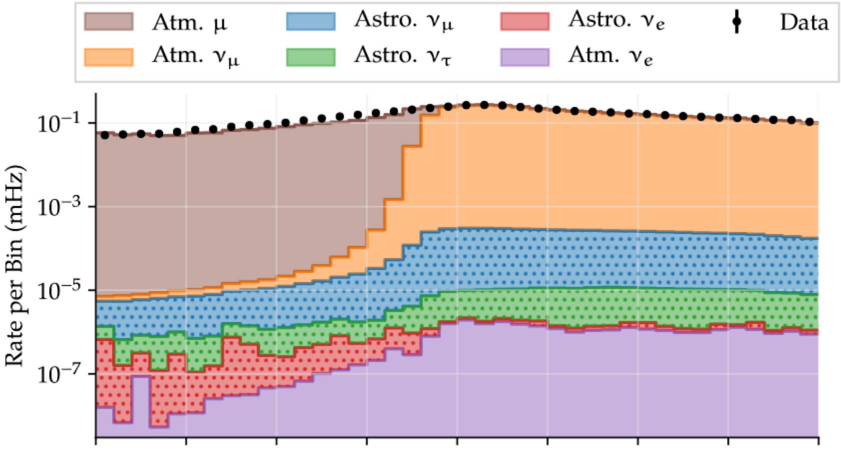


Figure 4.2: Distribution of events present on the GFU sample by declination. The different colors represent the different contributions of particles. Dotted events are astrophysical in origin while solid colored events are atmospheric. Image taken from [44].

Nowadays, the GFU sample of events is used in many high-level analyses that focus on the identification of astrophysical emissions. However, as the name implies, it was first developed as part of the analysis for the **Gamma-Ray Follow-Up Program.** As mentioned previously, this program generates alerts for significant events to be followed-up by collaborating institutions like MAGIC [23], VERITAS[26] and H.E.S.S. [25].

The GFU analysis [58] works by implementing a time-dependent approach. Because neutrino sources are expected to show temporal variability in their emission. The system needs to be designed to look for an accumulation of events in a significant time window. In pursue of these goal, a Time Clustering Algorithm is implemented. Because many sources are expected to have time-dependent emissions, this algorithm looks for grouping of events in a similar time window from the same region in the sky. Essentially, it performs a likelihood analysis including the timing of these events on the GFU Sample.

Now that we have described how the RealTime Alert system constructs samples of events designed for searches of neutrino sources, this thesis focuses on the Energy reconstruction section of the Analysis. As evidenced in this chapter, the energy is obtained during the Online L2 Filter.

This work seeks to implement an **energy reconstruction method based on machine learning algorithms** in the context of the RealTime Alert System architecture, testing its performance against the methods currently in place. Chapter 5 explains the algorithms used in this work and then Chapter6 explores the implementation of this algorithm as well as results obtained.

5
Machine Learning Algorithms

A Machine Learning algorithm is a tool designed to offer predictions on data based on input features. This type of algorithms are based on specific tasks recognizing patterns on datasets. After being *trained* on a prepared data sample, the system then is able to provide results from datasets different from the one it was trained on. A way to understand them is to consider them like black-box functions that take input features (x) to offer a predicted output. The function is composed of trainable parameters (w) which are optimized to fit a specific task.

$$\text{PREDICTION} = f(x, w) \tag{5.1}$$

There are two main types of Machine Learning Algorithms: Classification and Regression. Classification Algorithms focus on making decisions based on input data in order to assign them to a specific category. A classical example of this is identifying features of animal images to classify them as separate species, cats and dogs for example. On the other hand, regression algorithms aim to offer numerical results based on the input data. One of the best-known regression algorithms is the *Linear Regression*, which adjusts a set of data to a linear function and is then able to predict information in the continuous range even if it is not covered by the data.

Algorithms can be supervised, if the training of the algorithm is performed on *labeled* data. Basically, the training process occurs by constantly minimizing the difference between the results offered by the algorithm and the true value of the prediction. Conversely, unsupervised learning algorithms work on unsupervised data, identifying patterns in the data without the need for guidance and are particularly useful in identifying relationships in datasets where they are not apparent initially, as well as reducing human bias that might misidentify patterns based on different experiences.

Once an algorithm has been trained, it can be used to swiftly obtain predictions on new data obtained from the same source. For physics experiments, this is very useful to obtain parameters where typical numerical minimization might have a hard time converging. Also, they are able to be adjusted to the requirements of each experiment, providing a tool that goes from experiment-specific information to widely understandable physical parameters while avoiding the problem of biased and simplified assumptions.

In IceCube in particular, Machine Learning Algorithms have been widely implemented to the reconstruction and classification of data. As explained in Section 4.5, the GFU Filter implements a Boosted Decision Tree (BDT) Learning algorithm as a method for identifying and classifying events inside and outside of the data sample by assigning a probability of the event corresponding to a signal rather than the background. It performs this task by training the algorithm on the reconstructed event characteristics and identifying what makes an event more ”signal-like”. Furthermore, BDTs have been implemented in different analyses to reconstruct parameters such as the angular error [60]. Additionally, Section 3.3.3, presented a machine learning algorithm implementation for Energy reconstruction based on Convolutional Neural Networks [54]. This method takes the structure of the detector and deconstructs it into hexagonal pieces to identify patterns in event signatures to offer the energy of the events, effectively providing a method with increased accuracy and reduced runtime.

In this thesis, we implement a Graph Neural Network Algorithm for the task of energy reconstruction, seeking to improve the accuracy on previously presented methods. This section explains the functioning and construction of Neural Networks as well as the current framework for implementation within IceCube using the package known as **GraphNet**. Additionally we will present some examples of how this framework has been used within IceCube to tackle reconstruction challenges.

5.1 The Idea for Neural Networks

The idea is born from an analogy to the functioning of neurons in the human brain. A single neuron is taken to be the computational unit of the human brain. Neurons are able to communicate with each other based on electrical signals. Every unit receives many signals from the neighboring units. When a certain unit reaches a specific threshold, it fires a new signal. In another way, if we observe the system of neurons as a network, the state of each unit in the network is defined by the previous states of itself as well as nearby neurons. An algorithm based on this reasoning is called a Perceptron [61].

A Perceptron algorithm is a neural network of a single Neuron. The unit receives the input signals, it weighs them based on a specific function, then aggregates the weighed inputs. The aggregated result of this operation then is passed by an *activation function*. This function defines the final output. For instance, a classification Perceptron then will define the ranges at which the input is assigned a value. Essentially, this can be defined in the following manner:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{5.2}$$

$$\theta(y) = \begin{cases} 1 & \text{for y > a} \\ 0 & \text{for y < a} \end{cases} \tag{5.3}$$

Where x_i represent the inputs, w_i the applied weighs and $\theta(y)$ represents a simple binary step activation function with a threshold a for activation. A bias b, which is also an optimizable parameter, offers additional flexibility to the Perceptron

When this gets extended to a combination of Perceptron units, the algorithm becomes what is conventionally called an artificial neural network (ANN), or just neural network (NN). From this point forward, we use the terms Perceptron and Neuron interchangeably.

This type of algorithm receives a vector of input features, these are matched by an equal number of Neurons which process the information. This is called the input layer. Then, one or more layers can be implemented where the inputs to the next layer of units come from the previous layers, this is applied consecutively until you reach an output layer, returning a result of the desired type. This can be better visualized in Figure 5.1.

A general neural network is not immediately prepared to solve a physical problem. Being a Machine Learning Algorithm, it needs to be trained on labeled data. At its first state, a Neural Network will be initialized to a random state. Then it will evaluate the performance of the algorithm by measuring its output to the true label via a loss function. The purpose of the algorithm will then be to minimize the loss function on subsequent applications. The algorithm for optimization is called Gradient Descent [63]. What this algorithm does is calculate the gradient for the loss function, basically obtaining the direction in which the loss function has the steepest rate of increase and then updating the next iteration of the networks' weights

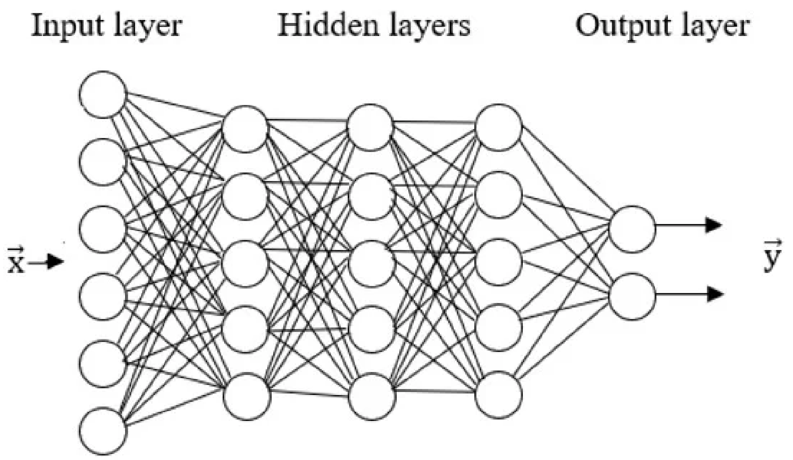


Figure 5.1: Sketch showing the functioning of an artificial neural network with 3 hidden processing layers, 6 input features and 2 output features. Image adapted from [62].

and biases in the opposite direction. Essentially, if the vector of weights is defined as w_i, the next iteration will be defined as:

$$w_{i+1} = w_i - \eta \nabla \mathcal{L}(w_i) \tag{5.4}$$

, where η is called the learning rate and is an input parameter of the training process.

An adaptation of this method, and the one used in this work, is the *Mini-Batch Gradient Descent*. Usually, the loss function and the gradient is calculated on the full dataset, averaging the loss from the all of the data. For the mini-batch version instead, the network is trained on a piece of the dataset at a time, applying the gradient descent on a smaller piece. This type of algorithm is useful on large datasets since it manages computational resources effectively.

5.1.1 Convolutional Neural Networks (CNNs)

As we have explored, a neural network receives an input of data in the form of a vector, a one-dimensional representation. However, problems frequently rely on inputs that favor a two-dimensional grid of features. A good example of this is an image, for an picture to be reduced to a one dimensional representation, an enormous amount of information needs to necessarily be lost. This problem has led to the development of Convolutional Neural Networks.

These networks take information in higher-dimensional representations, essentially a tensor.

An image, for example can be represented as list of 3 matrices, each of which contains a grid representing the values of intensity of a specific color (red, green, blue) for each pixel in the image). A *Convolutional Layer* is applied on this representation, aggregating the information from pixels into a ”feature map.” This is the algorithm then fed into the network which can then be managed by reducing information to more meaningful representation in subsequent layers. Eventually reaching the activation layer that offers outputs desired by the algorithm.

Because of the way they are built, CNN Neurons are connected to a small set of neighbors rather than the whole batch, making them computationally efficient, while still being powerful recognizing patterns in the data.

As presented in Section 3.3.3, IceCube has explored Convolutional Neural Network methods for Energy reconstruction in cascade events. For this method, the IceCube structure is extrapolated to an orthogonal shape and the pulses are summarized into 9 features for each DOM. This is done because of the requirement of Neural Networks for structured data inputs. However, this is a simplification of the irregular geometry and structure present at IceCube, which cannot be handled by a CNN. An attempt to solve this problems is explored in this thesis through the use of a type of neural network algorithm that generalizes to less conventionally structured data representations: **Graph Neural Networks**.

5.2 Graph Neural Networks

In a simple manner, graphs are a type of data structure very useful for many kinds of datasets. A graph is composed of *nodes*, which are entities possessing certain features; and *edges*, which define the relationship between the nodes. A graph is a natural representation for many real life problems like atoms in a molecule connected through chemical bonds; people on a social network connected by follows and likes; and very interesting to our case, IceCube events that can be defined as hit DOMs connected by proximity.

Graph neural networks then follow a similar workflow to general neural networks by applying the layers in a space defined by the graph features [64]. In this thesis, we are interested in a type of Graph Neural Network called Message-Passing Neural Network (MPNN), which are defined in a similar manner as typical Convolutional Networks but applied to graph data. In a sense, an image can be seen as a type of graph for which the nodes are the pixels and the edges connect it to the nearest neighbouring pixels. By extending this to general graphs, a GNN takes a node and updates its information in each layer based on their connected neighbors. The name comes from considering the updating process as a message passed through the edges to each node. Each message (m_{ij}) is constructed by the connected node features and their relationship:

$$m_{ij} = f_m(h_i, h_j, e_ij) \tag{5.5}$$

Then, the full message m_i passed to the node h_i is the aggregation of the individual messages and the node in the next layer will be defined based on this information.

$$\begin{aligned} m_i &= \textstyle\sum_j m_{ij} \\ h_i^{(k+1)} &= f_u(h_i^{(k)}, m_i) \end{aligned} \tag{5.6}$$

At each layer where a convolution is performed, a new graph is defined, whose space can be increasingly more abstract.

From a computational perspective, GNNs present similar advantages to CNNs where a problem can be approximated by reducing the identification of solution to local neighborhoods of nodes. They differ heavily in which kind of data they are best adapted for. Inputs to a CNN should be well structured in a grid like manner and can only offer results in similar structures. On the other hand, GNNs work well for less structured datasets by taking advantage of the abstraction for a graph representation, allowing them to handle problems with irregular structures and scale them to different graph representations for the same task. Because the IceCube detector is a system composed of many units (i.e the DOMs) embedded in an irregular geometry, a GNN offers a natural approach to reconstruct event parameters taking full advantage of the information available.

The computing costs of training and implementing a Graph Neural Network then depend on transforming data to graph representation. The training process can be lengthy in proportion to the number of steps and volume of data at which the Gradient Descent algorithm needs to be employed. However, experiments have shown that 5-15 iterations over a simple graph are sufficient to reach good results [64].

The following section explores a framework established for the implementation of GNNs in the context of neutrino-detection experiments, the **GraphNet** framework.

5.3 The GraphNet Framework

GraphNet [65] is a python-based open-source framework whose purpose is to create a pipeline to train and implement Graph Neural Networks for the reconstruction of event parameters for neutrino events. The objectives of the team developing GraphNet is to provide tools that are adaptable to different neutrino telescopes, taking the data provided from optical modules to train GNNs in order to reconstruct event parameters like energy and direction. The models developed using GraphNet should additionally be able to run in an efficient manner, providing results for an event's physical parameters in real time with better precision than traditional reconstruction methods.

The way GraphNet constructs a Neural Network is by taking each event as a graph. The pulses detected in the DOMs are the nodes and then each node is connected to its nearest 8 neighbors using the Euclidean distance as a measure of space. The features in each node are then: the characteristics of the DOM that recorded the pulse, as well as the time of detection and recorded charge. The process of convolution is applied on these features.

On the programming side, GraphNet integrates state-of-the-art packages and tools in Python. Among these, the implementation of Torch handles the constructions of tensor structures for construction of neural networks [66], while lightning [67] provides structure to the code and awkward provides the handling the variable-size data arrays common for neutrino experiments [68]. The integration of these different packages into specific modules allows users to focus their effort on the particular challenges required by the task at hand. Specifically, the GraphNet modules allow users to separate the workflow into: preparation of the data (using the *graphnet.data*); configuring and building the models (*graphnet.models*); the Training of the GNN (*graphnet.training*) and the implementation of already trained models (*graphnet.deployment*). Figure 5.2 shows a schematic of a typical workflow using these modules. Additionally, the following paragraphs offer a brief overview of these modules and their operation

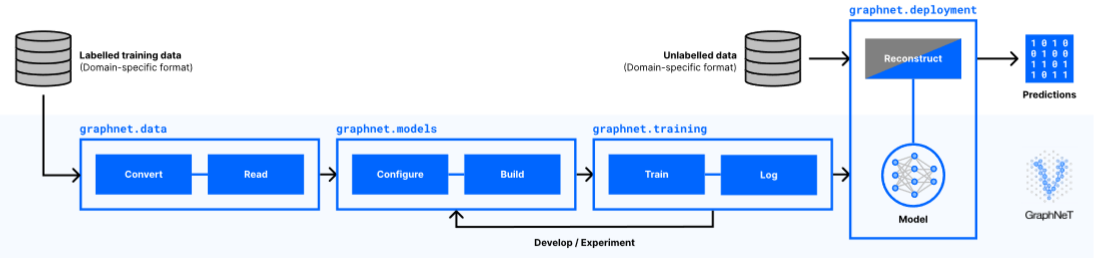


Figure 5.2: High Level Overview of a typical GraphNet workflow using the different modules in a sequential way, from converting data until the application of trained models. Taken from [65]

5.3.1 The GraphNet Data Module

The first of the modules included in GraphNet is the *Data* Module. The functions included in this module are used to read data from raw, detector specific formats as well as convert them to intermediate formats applicable to Torch GNNs.

Particularly for IceCube, the *DataConverter* functions use configurable extractors that implement the IceCube internal data handling functions (through the icetray framework) to obtain the relevant features from the IceCube *i3* files [1].

The end goal of this part of the workflow is to prepare *Dataset* objects that can be fed to subsequent steps to build a GNN. A properly built Graphnet dataset includes features obtained for the detector responses from an event that serve as input parameters to the GNN, this typically include the position of the activated DOMs, the accumulated charge and the time of photon arrival. Additionally the dataset must include *Truth* data, meaning the data that will be reconstructed or classified by the network, like energy, direction or Particle ID. Finally, the Dataset also needs to include a *GraphDefinition* that handles the transforming of the features to proper implementation in a Graph. For this work we use the built in definition of graph for IceCube.

5.3.2 The GraphNet Models Module

Models in GraphNet are the main communication between the data and the implementation of the algorithm. Additionally, they define the physics operation that will govern the process of training and reconstruction as they pass the layers of the network. These models are built in such a way that they can be instantiated from a specific type of configuration depending on the objective of the work (called *Task* inside the Model). A model should be built on an appropriate *Graph Definition*. This part of the model, as the name implies, defines how the system will convert events into the graphs. As mentioned previously, each event corresponds to a single graph. Typically, each DOM will correspond to a single node but there exist other possible node definitions implemented in IceCube. In this work we implement the *"Nearest Neighbors"* Graph Definition connecting each node to its 8 nearest neighbors.

Additionally, the way the convolution will be performed across the different layers of the network is defined through the **Backbone.** In this work, the implementation is based on the *Dynedge* architecture [69]. The implementation performs the convolution by using an *Edge-*

[1] i3 files encode the information from experimental and simulated events in IceCube. They contain the raw information from the DOMs as well as higher-level reconstructions for each event stored in "frames".

Conv algorithm, essentially a Perceptron algorithm with inputs of the DOM features as well as the difference between it and the 8 nearest neighbors in Euclidean space as the connected nodes. This redefines the graph in a more abstract space. After completing 4 layers of convolution, the algorithm passes the combined input of the 5 graphs from each layer to another Multi-layer Perceptron that maps it into a unified set and then aggregates the nodes into a single feature list, combined with 5 global parameters that characterize the number of pulses as well as the ratio of DOMs hit multiple times. The final step of Dynedge passes this list to a final Perceptron that offers the prediction. A simplified sketch of the algorithm is presented in Figure 5.3 with a more in depth description available in [69]

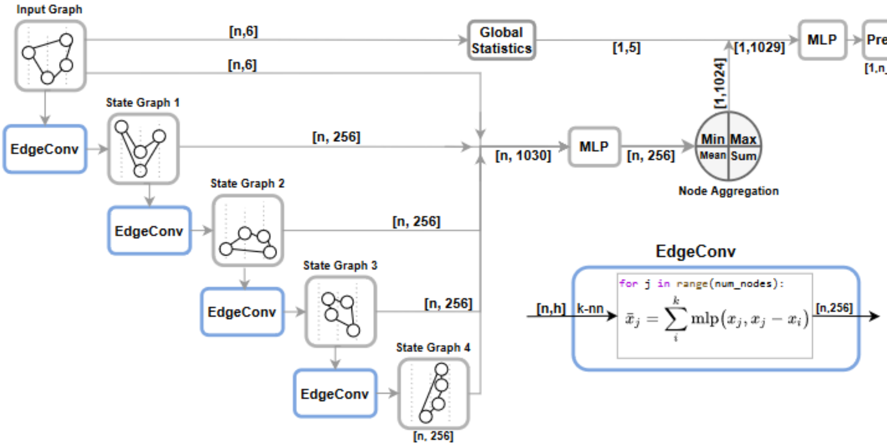


Figure 5.3: Sketch of the architecture in place at Dynedge. The input presents a simplified graph whose embedding space corresponds to Euclidean space. For the following convoluted *State Graphs*, the embedding space becomes more abstract and connectivity changes at each redefinition. The inner workings of the EdgeConv blocks are presented to the right. Taken from [69].

A GraphNet Model should be easily redeployable to different tasks and datasets by changing the Graph Definitions and Tasks. In this work, we implement that built-in *StandardModel* class that allows for quick customization of the Dynedge on the Nearest Neighbors Graph Definition for the Task of Energy reconstruction.

5.3.3 The GraphNet Training and Deployment Modules

When Data and Models have been defined, the training module is able to effectively create GNNs from datasets by calling the appropriate functions. The training requires determining a set of features to take from the dataset which will serve as the inputs to correlate with the target of the task of the model. Additionally, the training requires setting the *Loss Function* to be minimized at after every step of training, as well as the implementation of the Gradient Descent algorithm.

During training, the dataset can be split into training and validation sets. This way, at every epoch, the performance of the model can be tested so that the algorithm for training can be adjusted as the training occurs. On top of that, the validation allows for the performance of hyperparameters can be tested. Because GraphNet is built on Torch, the main hyperparameters to tune are batch size, learning rate and number of epochs. Further explanation of these hyperparameters are explored in Section 6.2.1.

After the model is built, it is saved as the configuration of the tasks defined in the previous section, as well as the state of the weighs on which the GNN architecture is built. The *Deployment* then consists mainly on two steps: first, the definition of the GNN according to the saved model is set by an *"Inference"* function; then a *"Deployer"* function takes this implementation and applies it on the data containing the input information for reconstruction. In the context of IceCube, they can be directly deployed on *i3 files* by use of a well-defined *Deployer*.

5.4 Previous uses of GraphNet for reconstruction challenges

As a framework for implementation of GNNs into neutrino physics challenges, GraphNet is still relatively new and in continuous development. It has nevertheless been implemented into a number of studies for neutrino classification and reconstruction, demonstrating its effectiveness. The development of Dynedge is explained in [69], alongside its deployment for neutrino events of low energies. Typically, these types of events suffer from a large background of atmospheric neutrinos, which motivated the implementation of the GNN. In said work, the performance of the GNN was tested for six different tasks: the classification of events into muon or neutrino (ν/μ), as well as classification intro track or cascade events (T/C) and the reconstruction of deposited energy (E), direction of the events (θ, φ) and interaction vertex (V_{xyz}). For each of the tasks, the training dataset included simulations of neutrinos produced through the GENIE framework, typically used in oscillations, with an interest in neutrinos below 1 TeV, the majority of which are below 100 GeV of energy. Those for the classification tasks were selected so that the sample contained equal numbers of both types of event classifications in each while the reconstruction training sets are selected to contain equal numbers of all neutrino flavors.

The performance of the reconstruction algorithm is tested against the RETRO algorithm [70], which is the standard for low energy reconstructions. This algorithm is complex in that it is a likelihood method that models events in a full 8-parameter space that account for a full event reconstruction, allowing for minimization of all reconstruction parameters. Although the algorithm is very efficient, it is relatively slow, with a mean application time of 40 seconds per event on a single CPU core, because of this, it is usually applied at late stages of analysis, when samples have been trimmed down significantly. Conversely, the performance of classification is tested against a Boosted Decision Tree (BDT) Algorithm.

For classification, this study compares the results of the GNN and BDT Receiver Operating Characteristic curves (ROC), which plot the True Positive rates (TPR) against the False Positive rates (FPR) from the binary classification, estimating the score of the task using the area under the curve (AUC). For both classification tasks, the score is clearly improved (by 4% for the ν/μ task, and 6% for the T/C task). Particularly, for the ν/μ classification, at a fixed FPR of 0.025%, the TPR is improved by the GNN from 0.6 to 0.78. Otherwise at a fixed TPR of 0.6, the FPR improves from 0.025 to 0.003. Overall, the classification offers a clearer and more accurate separation between event types.

On the other hand, for the reconstruction tasks, the GNN is shown to over-estimate deposited energy for low-energetic events. For all reconstructed parameters, the distribution of results differ slightly from both the true value and the RETRO reconstruction, but show overall better resolutions of the estimation. A visualization of this results taken from [69] is presented in Figure 5.4. This graph shows the width of the residuals distribution as a function of energy for both GNNs and RETRO to quantify the resolution of the reconstruction. The improvement appeared to be the most evident at very low energies (1 GeV to 30 GeV), where Dynedge exhibits an improvement of up to 20% for all parameters. However, at higher energies in the sample, the improvement decreases heavily. This has been ascribed to lower statistics in the training sample and is something that should be taken into account for reconstruction tasks that aim to be accurate in the full spectrum of energies.

A further implementation of GraphNet on low energy events is presented in [71] in the context of neutrino oscillation studies taking advantage of the expected IceCube upgrade. In this work, the effectiveness of the classification is improved when compared to the previous study [69] through the implementation of a larger sample. On the other hand, the reconstruction of parameters has been improved in the whole low energy range.

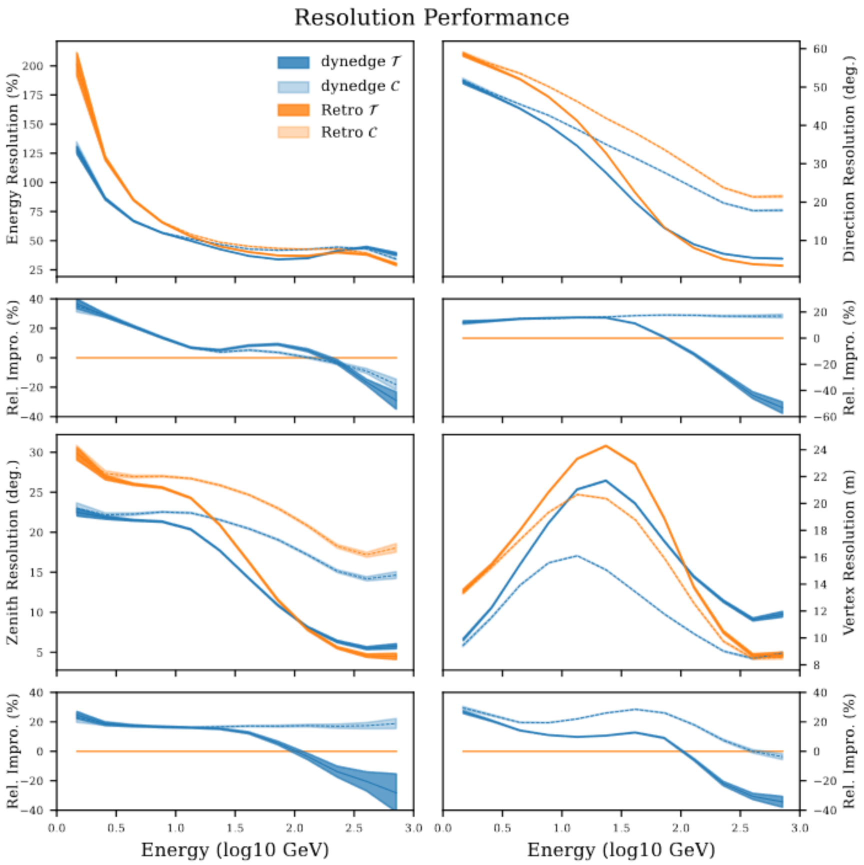


Figure 5.4: Performance of GNN reconstruction of event parameters as presented in [69]. The Resolution is defined as the width of the residuals distribution for each reconstructed parameter. Each graph shows the resolution distributed in the energy range of the testing samples, with the thickness representing the 1σ uncertainty, also showing the improvement in resolution when compared to the benchmark RETRO algorithm.

Another implementation of GraphNet is presented in [72] in the context of searches for neutrino sources on the night sky, focusing on NCG1068. They studied the possibility of improving the reconstruction of energy, by training a Dynedge model on a small sample. In this work, the performance of the GNN is compared to a DNN algorithm, the training of the GNN was performed on one sixth of the data used for the DNN, while showing similar performances, suggesting that a GNN is at the very least able to be configured with more ease in comparison to other neural network methods. Another proposition is that, when trained on similarly-sized datasets, the GNN could improve over other machine learning methods, but this possibility requires further study.

The implementations explained above show the potential of GNNs to offer improved precision in energy reconstruction, in contrast to current methods used in IceCube while evidencing the need of training on rich samples in the energy range needed for the specified tasks. From this starting point, this determines the work needed for this study. We work on the reconstruction of energy events in the context of GFU, which spans a wide range of energies, because this selection focus on the determination of astrophysical sources, the energy range is expanded when compared to the previous studies. The following chapter explains the selection of the datasets from which we train a Dynedge GNN as well as the implementation of the modules explained previously in this chapter for energy reconstruction. This chapter will explore the preparation of the Datasets as well as the results of training for the tuning of hyperparameters and the final deployment of trained models on relevant sets.

6

Implementation of Graph Neural Networks for Energy Reconstruction for RealTime applications in IceCube

In this work, the objective is to implement the GraphNet framework in the context of the RealTime System, focusing on on the reconstruction of energy. As shown previously, GNNs have been able to precisely reconstruct event energy for well reconstructed samples on offline data. Particularly, [69, 71] have shown that the use of GNNs are effective at reconstructing and classifying neutrino events at low energy. On the other hand, [72] has shown that even when trained conservatively. These results approximate current methods effectively, while demonstrating their efficiency at improving runtime. Although this implementations show effective results, they also demonstrate the biases that can arise from the choice of training sample. When the training datasets do not contain even statistics in the energy range, the GNN tends to bias to the energies with higher event statistics in the sample. The suggestion made in these studies is that this bias would decrease when training on samples that contain more information on events at all energies. For an application in the RealTime Alert System, a model should be able to reconstruct energies in a wide range, especially at high energies, if they are expected to be useful for alerts.

The challenge posed for this section is using sets of simulated data for training that include tagged energy data in the full IceCube energy range. Therefore the choice of training sample is important, as well as studying who the GNN behaves in different energy ranges. While previous studies have focused on very low energies, in this work we focus on a wide energy spectrum starting at 100 GeV and up to 100 PeV. The training of a GNN is simpler for low energies, where the number of pulses is small, as the computational difficulties in training will increase as the graph representation size increases. In the following sections, the configuration of a GraphNet architecture appropriate to handle the full energy range is explained, looking out for the best implementation and finally comparing it against the results offered by the methods currently implemented for online energy reconstruction 3.3.3.

For this work we implement the *Dynedge* algorithm [69] with the task of energy reconstruction. The input parameters chosen to train the GNN comes from the pulses detected by the DOMs. Specifically, the features used are: the positions of the DOMs triggered for each event in cartesian coordinates (DOM$_X$, DOM$_Y$, DOM$_Z$); the built up charge at each DOM (Q); the time relative to the trigger of the event start (t); and the Relative DOM Quantum Efficiency of each unit (RDE). This results in an input array of dimension [n$_{pulses}$, 6], corresponding to a single value of energy. This number then scales with the number of events in the training dataset to account for the amount of input data into the training algorithm. The end result of training is therefore a model that takes an input of n$_{events}$ arrays of [n$_{pulses}$, 6] and the outputs are a prediction of Energy for each event.

Similarly to [69], A LogCosh [73] function is used as the loss function for the training:

$$Loss = \log(\cosh R_E) \tag{6.1}$$

where R_E denotes the energy residual for training:

$$R_E = \log_{10}(E_{reco}/GeV) - \log_{10}(E_{true}/GeV) \tag{6.2}$$

the choice to embed the Energy in the logarithm space is made to account for the large range of deposited energies. The LogCosh loss is chosen because it is symmetric around 0, avoiding biases to under or over-estimation. Additionally, the function is smooth and differentiable everywhere, making it optimal for the gradient descent calculation.

Furthermore, the Adam [74] optimizer algorithm is used for the application of the gradient descent. This algorithm saves the information from previous gradients to improve on memory usage while converging faster than classical algorithms. At the same time, it implements an adaptive learning rate, making the training process flexible without the need of precise user supervision.

The GraphNet algorithm pipeline has been modified so that upon data extraction and conversion it takes the true energy of the muon that creates the track, calculated differently for through-going tracks, starting tracks and skimming tracks, which is the energy we aspire to reconstruct. Additionally, we implement our algorithm in a way that we can test hyperparameters to be used on a reduced set before proceeding to a full training of the GNN.

First of all, We explain the simulation sets that were used for this work, and how they were processed for use within GraphNet. Then we will show the process of optimization of hyperparameters, the subsequent training of the GNN, and finally the results of efficiency of the trained models on relevant sets of data.

6.1 Characteristics of the Simulation Datasets

The simulated events that are used in the context of this work need to consist of rich sample of neutrino events in the full energy range, so that our energy reconstruction can tackle the determination of which events are energetic enough to be considered for alerts. The selection of events to train on are generated from the Neutrino Generator (NuGen) framework [75] that injects neutrinos inside the Earth and propagates them until they arrive to the detector volume, where they are forced to interact based on the provided Ice Model and a model of the detector response.

NuGen simulation sets are generated for neutrinos of different generations and energy cuts. Since we focus on the reconstruction of muon tracks to identify high-energy sources, in this work we use sets of generated muon neutrinos. The implementation proceeds on selected sets of different energy cuts.

For training and validation, we use high statistics sets denoted with IDs: 21633 for low energy (100 GeV to 10 TeV), 21634 for medium energies (10 TeV to 1 PeV) and 21635 for high energies. Meanwhile we use sets 21813, 21814 and 21938 (low, medium and high energy respectively) for the testing of results. These sets have been further processed to include the filters and reconstruction appropriate for GFU. Further characteristics of these sets are summarized on Table 6.1

Dataset ID	Energy range	Spectral Index	Events per File	Number of Files
21633	100 Gev to 10 Tev	-2	100,000	10,000
21634	10 TeV to 1 PeV	-1.5	10,000	20,000
21635	1 PeV to 100 PeV	-1	500	10,000
21813	100 Gev to 10 Tev	-2	100,000	20,000
21814	10 TeV to 1 PeV	-1.5	10,000	20,000
21893	1 PeV to 100 PeV	-1	250	10,000

Table 6.1: NuGen Simulation datasets used for the training and testing of the GNN implementation

6.2 Training of the Graph Neural Network

On the studies presented on the previous chapter, the GNNs for reconstruction were trained on datasets containing an order of magnitude of 10^6 events. In this work, we have to consider the datasets available at all energies as well as their characteristics. Low Energy datasets are processed easily and there is very little issue with using 10^6 and higher orders of magnitude of events. However, datasets for high energy neutrino events are computationally expensive to process in the training of a GNN. To maintain reasonable execution times and memory consumption for the training of the Graph Neural Network, we limit the number of high energy events for training to 10^4 events. Based on this, in order to maintain a degree of uniformity in the training set statistics, we limit the selection from the medium and low energy sets to orders of magnitude of 10^5 events.

6.2.1 Hyperparameter Optimization

Before training the GNN that will provide final results, we need to investigate the hyperparameters that guide the training of the model. In order to do this we implement a version of the training/validation algorithm on different sets of hyperparameters. These are:

1. **Learning Rate**: Determines how much the algorithm gets corrected at each step of training. A learning rate too big will lead to divergence while a learning rate to small might prevent the algorithm from obtaining proper results. The learning rate set in the code is the aim learning rate of the optimizer. The Adam optimizer starts at a learning rate of 10^{-5} and sets up a linear scheduler to scale it up to the set value at fixed intervals to optimize results during the training process. Typically becoming smaller to fine-tune the reconstruction in the later steps.

2. **Batch Size**: We implement GraphNet using the *mini-batch* method described in Chapter 5, defining the number of events included in every batch. larger batch sizes might produce coarser corrections, but small batch sizes might lead to over-fitting while increasing training times due to a large amount of steps.

   In previous works [69], the batch size was chosen to be 1024. However, this poses a computational challenge. When processing the training, the batches have to process this number of events, which have to be converted to a graph representation. For low energy events, this does not pose too much a problems, but as the number of pulses gets larger, as can be expected for events at increasingly higher energies, the graph representation size will increase exponentially, which places strains on the computing systems.

   To solve this problem, we reduce batch sizes by an order of magnitude of 10, maintaining the convention of using powers of 2. In order to tackle the new arising problem of possible overfitting, we implement an *accumulation of gradients* in the algorithm. What this process does is to delay the calculation of the gradient and updating the weighs until a certain number of batches have passed. This creates a larger effective batch size while keeping the computational cost low. Because we reduce the batch size by an approximate order of 10 with respect to [69], we choose to accumulate over the same number of batches.

3. **Number of epochs**: The number of times the training will go over the entire dataset. Having a bigger number of epochs can lead to better functioning of the training algorithm with an adaptive learning rate as the task gets more precise in each iteration. Essentially, it is analogous to a person revisiting a situation with more experience and abilities. This of course increases training times, while not always significantly increasing precision.

The values we test for the hyperparameters are presented in Table 6.2. Additionally, to avoid overfitting, we allow for the training to stop after 3 epochs if no significant improvement on the loss function is observed.

Learning Rate	0.01	0.03	0.05
Batch Size	128	192	256
Number of Epochs	5	7	10

Table 6.2: Values to test for the different training hyperparameters. The values were chosen by taking into account computational restrictions while keeping them consistent to previous studies [69, 72].

We set a proportion of 0.9 to 0.1 for training/validation. Therefore, at the start of the training, the algorithm selects at random 90% of the events from the input dataset to perform the training. Then, after every epoch, a validation step is performed on the remaining 10% of events. We obtain the metrics to judge the parameters configuration from this validation step.

We perform this optimization separately on each of the energy sets. This allows us to save computer time and resources by distributing the tasks while also observing the performance of the models on different energy scales. By the end of this process, we are able to evaluate which subset, or subsets of hyperparameters will provide the best results to train the final architecture of the network.

6.2.2 Optimization on Low Energy Events

We perform this first section of optimization by training on 80,000 events from the low energy section (100 GeV to 10 TeV). The validation is performed automatically after the training and we study the performance of the model on the validation sample. In order to evaluate the performance of the model, we take the mean absolute error from the validation prediction.

$$\text{MAE} = \frac{1}{N_{events}} \sum_{i}^{N_events} |E_{true} - E_{pred}| \tag{6.3}$$

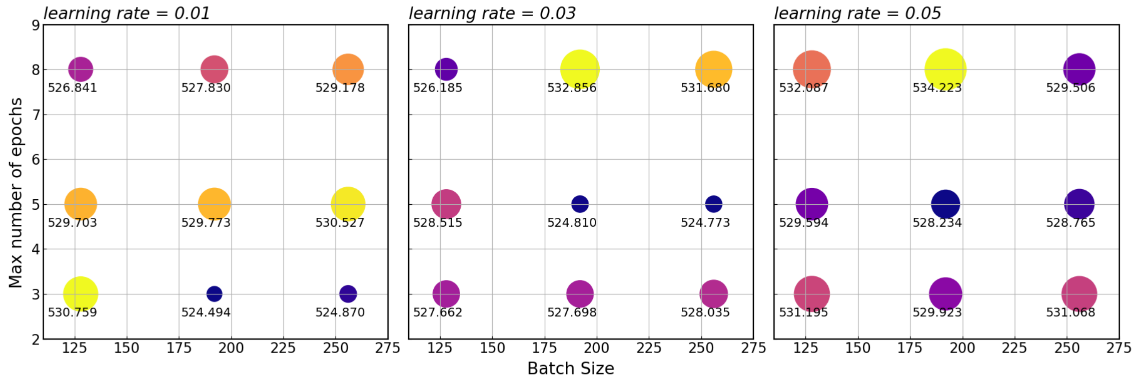


Figure 6.1: Performance of different hyperparameter settings for a graph Neural Network model for Energy Reconstruction in Low Energy Events (100 GeV to 10 TeV). Each graph represents a different learning rate setting (left: 0.01, center:0.03, right: 0.05). The dots' positions on the graphs represent the hyperparameter configuration (batch size on the x axis, max epochs on the y axis. The size of the dots is related directly to the value of the mean average error, also written next to each dot. The color of the dots represent the value of the error within the single graph (blue = smallest to yellow = largest).

Results from optimization are presented in figure 6.1, the different configurations of hyperparameters are presented as dots with definite coordinates in the graphs, with the size of each dot representing the value of the mean absolute error. Immediately, a few features jump out. The first thing to be noticed is that the highest learning errors occurring for the highest learning rate (lr=0.05), with the lowest errors happening for the smallest value of learning rate (lr=0.01). For the Max number of epochs, it seems that the choice for 8 epochs rather than 5 does not improve the MAE significantly, worsening it in half of the cases. Nevertheless, it should be noted that in most cases, the differences are too close together to make definite statements about this. For the batch size on the other hand, there does not seem to be an overall trend. However, we can identify a set of 4 best configuration of hyperparameters, and it is notable that none of them include the smallest batch size of 128:

1. lr = 0.01, batch size = 192, max epochs = 3 (MAE = 524.494)
2. lr = 0.01, batch size = 256, max epochs = 3 (MAE = 524.870)
3. lr = 0.03, batch size = 192, max epochs = 5(MAE = 524.810)
4. lr = 0.03, batch size = 256, max epochs = 5 (MAE = 524.773)

6.2.3 Optimization on Medium Energy sets

At this point, when moving into the optimization of hyperparameters for medium energy sets (10 TeV to 1 PeV), a problem arises. The first step in the algorithm converts a dataset in appropriate format into a number of graph representations, where each event is represented by a single graph with the each node representing every pulse that composes the detection in IceCube. The problem in the context of the GNN is that, as energies increase, the number of pulses per event increases exponentially, and with it the number of nodes and consequently size of the graph representation of each event. This means that for a set of events of similar size to the ones in low energies, the medium energy events take many times longer to train and occupy unreasonable memory resources.

To solve this issue, we take an approach similar to the one used in [54] for a Convolutional Neural Network. Instead of taking every pulse as a node, we represent every node as a DOM and take the 10th, 50th and 90th percentiles in buildup of time and charge. Effectively, this changes the dimensionality from [n_{pulses}, 6 features] dimensions to [$n_{activedoms}$, 6 features × 3 percentiles]. This reduces the memory consumption of training while maintaining a reasonable characterization of events for training. Using this implementation, we look for the most optimal set of parameters in the same manner we did for low energy events, training networks with different configurations on sets of 5̃0,000 events.

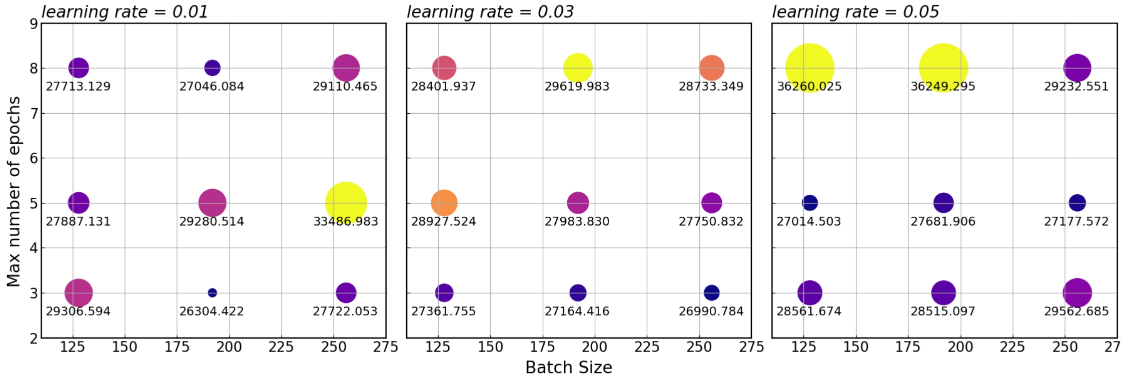


Figure 6.2: Performance of different hyperparameter settings for a graph Neural Network model for Energy Reconstruction in Medium Energy Events (10 TeV to 1 PeV). Each graph represents a different learning rate setting (left: 0.01, center:0.03, right: 0.05). The dots' positions on the graphs represent the hyperparameter configuration (batch size on the x axis, max epochs on the y axis. The size of the dots is related directly to the value of the mean average error, also written next to each dot. The color of the dots represent the value of the error within the single graph (blue = smallest to yellow = largest).

This results are shown in a similar manner to the previous section in Figure 6.2. Immediately, the differences in the size of the mean absolute error are more evident. Again, the biggest errors are found for the highest learning rate, while the lowest learning rate also offers the smaller errors. Nevertheless, the range these errors span is larger than the previous case. On the other hand, for lr = 0.03, the errors are more consistent for all configurations. Overall, the best configuration of hyperparameters seems to be the same at the one for the low energy case, by a much wider margin in this case (lr = 0.01, batch size=192, max epochs = 3).

6.2.4 Optimization on High Energy Events

While it could be expected that the same issue encountered for medium energies would happen when scaling to higher energies, the method of clustering on percentiles for DOMs described in the previous section is robust enough that it avoids unreasonable increases in memory consumption. Therefore at this point, the architecture of the network is found to be stable computationally. However, the files containing the information still consume a lot of memory when being loaded into the training, in fact for this reason each file contains less elements. For this reason, the training is performed on less events than the previous two sections at 1̃0000events. Then we look at hyperparameter configurations in the same way as we did for the previous two cases.

This results can be similarly observed in Figure 6.3. From this figure it is more complicated to observe overall trends in the performance of the algorithm. However, we can identify the two best performing configurations, and compare them to the observations from the previous sections:

1. lr = 0.03, batch size = 192, max epochs = 5 (MAE = 5.189 $\times 10^6$)
2. lr = 0.01, batch size = 192, max epochs = 3 (MAE = 5.260 $\times 10^6$)

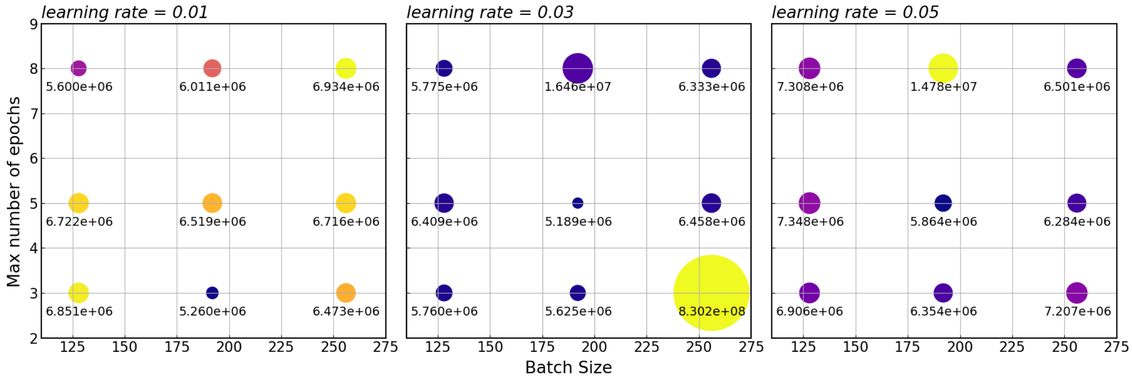


Figure 6.3: Performance of different hyperparameter settings for a graph Neural Network model for Energy Reconstruction on High Energy Events (1 PeV to 100 PeV). Each graph represents a different learning rate setting (left: 0.01, center:0.03, right: 0.05). The dots' positions on the graphs represent the hyperparameter configuration (batch size on the x axis, max epochs on the y axis. The size of the dots is related directly to the value of the mean average error, also written next to each dot. The color of the dots represent the value of the error within the single graph (blue = smallest to yellow = largest).

From the observations made in this section we can conclude that the performance of the the neural network depends on the combination of the 3 hyperparameters. However, by noting the common patterns of the best-performing configurations in all three cases, we choose the best configuration from the most frequently occurring one:

- **Learning Rate:** 0.01
- **Batch Size:** 192
- **Number of Epochs:** 3

6.3 Results from Implementation of the Trained Graph Neural Network

Having defined the configuration to train our GNN architecture, we choose a sample of 4.5 $\times 10^5$*events* to train the neural network, consisting of low energy events, medium energy events, high energy events. The sample was chosen to take into account the computational costs of implementing the training with GraphNet compatible files while maintaining significant statistics across the full energy range. The training of the GNN require 48 hours to complete on this set of events while maintaining a reasonable handling of memory (below 10 GB).

Having trained the architecture of our Network model, we apply it on the test datasets described earlier in the chapter, a combined sample of 2.3×10^6 events are used in this section (1.4×10^5 low energy events, 9×10^4 medium energy events, and 6×10^4 high energy events).

The runtime of the application of the GNN depends heavily on transforming IceCube data into a graph representation, and as explained previously, this is heavily dependent on the energy of the event. After training on our chosen datasets, we found an average time of implementation of 0.004 seconds per event on low energy datasets, 0.01 seconds per event on medium energy datasets and 1.4 seconds per event on high energy datasets. In comparison to the median values for the runtime of MuEx (0.04 s) and Truncated Energy (0.01) presented in Table 4.1, the performance is improved by the GNN in the low and medium energy ranges (from 100 GeV), while the performance is worsened for the high energy case. However It should be noted that an advantage for runtime is that a GNN can be deployed on multiple CPU cores. For the implementation in this work, we use 10 cores, but this could be expanded to increase the speed of reconstruction if the resources are available. On the other hand, a deployment on a GPU would be expected to increase the rate of processing by a few orders of magnitude.

Figure 6.4 shows the spectrum of the energy reconstructions applied to the datasets mentioned above. Because of the different statistics managed in each section, each of the dataset reconstructions is presented on a different histogram. Each graph also includes the true muon

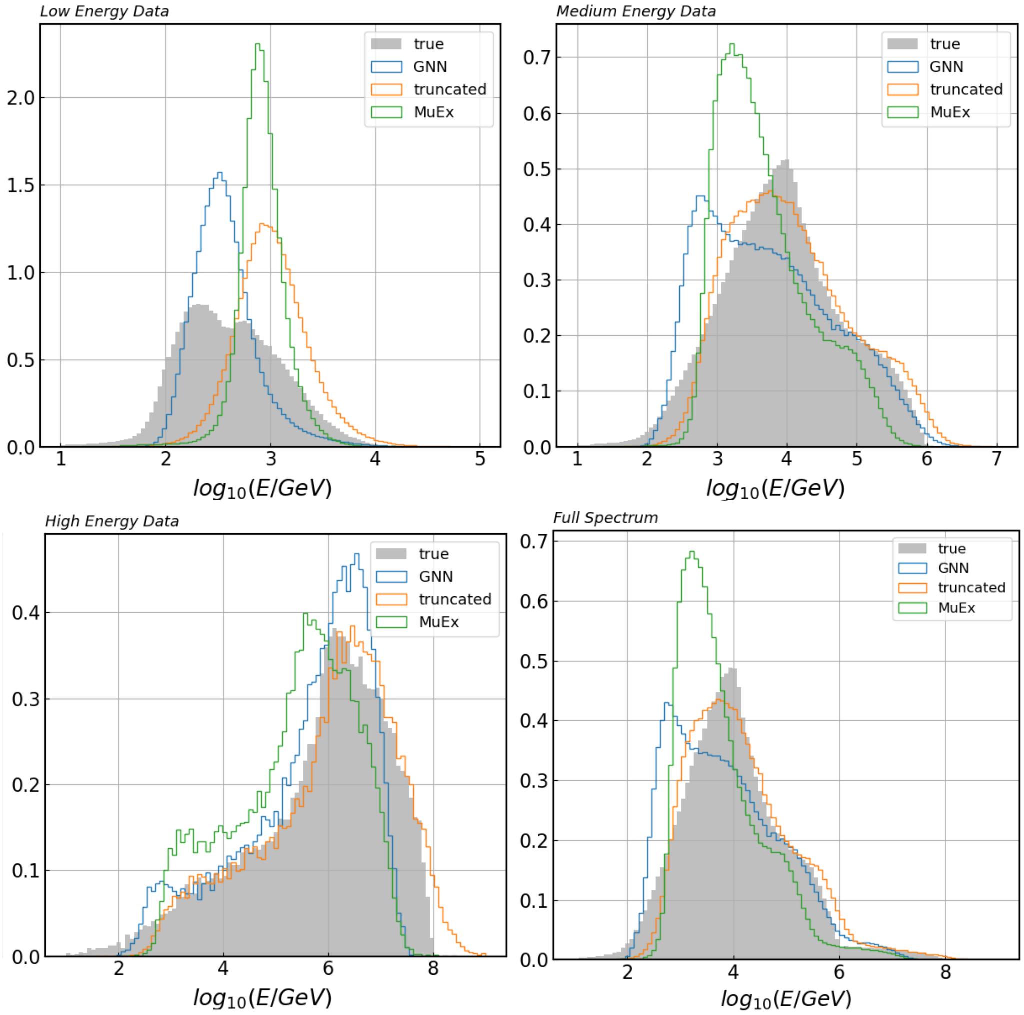


Figure 6.4: Distribution of true values of energy (shaded grey) from simulated sets in comparison to the energy reconstructed for the same events using our GNN Dynedge Method (blue) and compared against the Online L2 methods: Truncated Energy (Orange), and MuEx (green). Each of the graphs corresponds to events from datasets with different features: Low Energy sets (top-left), Medium Energy Sets (top-right), High Energy Sets (bottom-left) and the full studied dataset (bottom-right).

energy distribution from the simulated events as well as the reconstructions performed during the Online L2 Filter's process. From this figure we can see that, as expected, the MuEx method offers consistently the worst adjustment to the true energy distribution.

For the low and medium energy datasets, the GNN method seems to have a bias towards lower energies. For the low-Energy Datasets and medium-energy, this shows an under-performing in the 1 TeV to 10 TeV range while over-performing in the 100Gev to 1 TeV range. This feature implies a bias from the algorithm towards lower energies in this range. This fact was also observed in previous works using GraphNet for low energy datasets, and has been attributed to a feature of low statistics in certain regions of the datasets. Given that this region lies in the border of the low and medium energy datasets, this can explain this feature in the distributions. A less severe version of this phenomenon can also be observed in the distribution for the high energy dataset, as energies on the higher end of the distributions tend to be reconstructed as lower energies. Suggesting that this border effect might be indeed very relevant to consider for the training of a Graph Neural Network.

Out of the 3 distributions, the one coming from the low energy set shows the clearest improvement when compared to the Truncated energy and MuEx methods, being contained within the true energy distribution range and having a peak in the same region as that one. For the Medium Energy Datasets, The GNN method improves substantially from the MuEx method but is comparatively worse than that of Truncated Energy. However it is reasonable that this is a feature of the previously noted effect and could be smoothed out by an appropriate redefinition of the training sample. The bottom-right figure is the aggregation of the 3 distributions on the full range of energies and gathers the features explored in this section.

Furthermore, Figure 6.5 shows the correlation between the true energy and the reconstructed energy for both the Graph Neural Network Dynedge method implemented in this work, as well as the Truncated Energy Method. As well as the distribution of residuals calculated as:

$$R_E = \frac{E_{reco} - E_{true}}{E_{true}} \tag{6.4}$$

The graph shows an additional plotted line representing the median correlation of the method at each value of true energy, as well as a black line that follows $E_{reco} = E_{true}$. From this data we can compare and contrast the accuracy of both methods. We can confirm that the GNN method clearly performs better than the Truncated energy method at very low energies. While the median reconstruction is typically good for both, the reconstruction for the truncated energy is consistent although typically higher than the true value of energy. Conversely, our Dynedge method shows a median reconstruction that matches the central line very well around $10^5 GeV$, in the center of the the full range

It should be noted that the precision of both methods, as observed by the width of the residual distribution, is inversely related to the true energy of the muon, with the Dynedge method

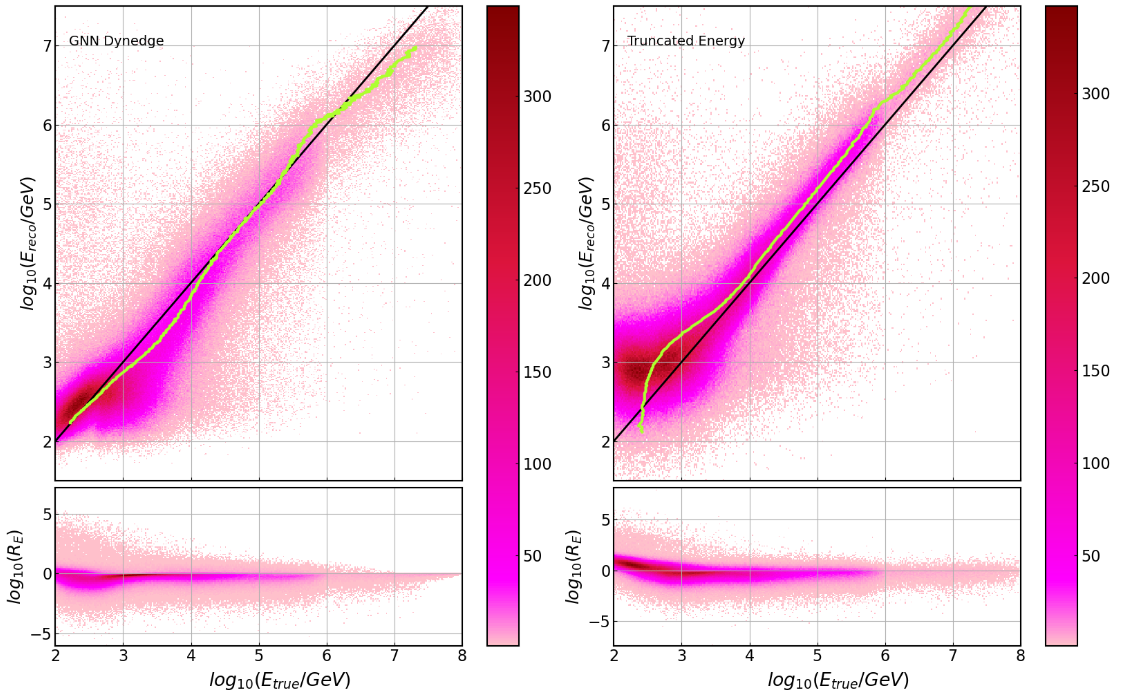


Figure 6.5: Correlation plots of reconstructed and true energy for the currently implemented Truncated Energy and our GNN Dynedge methods. The intensity of the colors represent the concentration of points, the black line represents the line $E_{reco} = E_{true}$ and the green line represents the median reconstruction at each value of True Energy.

improving significantly as it rises. On the other hand, the Truncated Energy precision decreases more slowly with energy. Finally, it should be highlighted that the distribution of residuals is typically narrower for the Truncated method rather than Dynedge.

7 Conclusion

7.1 Summary of Results

recap of theory and intro and icecube

At the time of this work, the development of Graph Neural Networks (GNNs) for neutrino event reconstruction is an active field of research, with many promising results being offered in the past few years in the context of the IceCube collaboration, especially considering low neutrino energies (below 100 GeV) relevant to oscillations in the DeepCore section of IceCube [69, 71]. This thesis in particular explored the implementation of the state-of-the-art GraphNet algorithm to the reconstruction of event energy in the range relevant to the RealTime Alert System in IceCube.

After studying the configuration of best hyperparameters, we trained a Neural Network architecture on simulated neutrino events from 100 GeV to 100 PeV and tested the performance of this algorithm on a set of neutrino events with similar characteristics. The results from this implementation show that a GNN method of reconstruction for energy ss able to produce effective results across the energy range, with a similar accuracy to currently implemented methods. Additionally, when implemented on 10 CPU cores, the reconstruction runtime was improved in comparison to current methods in the medium and low energy ranges. While the performance is worsened in the high energy case. However the runtime performance can be scaled linearly by increasing the number of cores, or exponentially if applied on a GPU. Showing how the implementation can be further improved to the relevant application in the IceCube RealTime Alert System.

We additionally identified a problem in low energies, where the algorithm tends to underestimate the energy of neutrinos between 1 Tev and 10 TeV as lower energy, this bias is likely due to overfitting of the networks architecture on this region. This is consistent to previous observations which have suggested that uneven statistics across the energy range can lead to worse performance of the algorithm. It should be noted that this effect is not observed clearly at higher energies. At these energies however, the neutrino spectrum is expected to be smoother. Which might make the training samples naturally more even, disfavoring overtraining on a single region.

This work can serve as a starting point to fully develop the implementation of Graph Neural algorithms in neutrino detectors, not only in IceCube, but in many present and future astrophysical neutrino detectors. As such, care has to be taken on how the results presented in this thesis are improved and built on in the coming years. The next section describes some proposals to start looking at improvements on this system.

7.2 Evaluation and Future Outlook

While the results obtained for the GNN architecture show the power that a method of this type can offer to the reconstruction of energy, there are many possibilities to improve the accuracy of the algorithm that can be considered to increase the effectiveness of the Neural Network. The first thing to consider is the fact that the distribution of residuals becomes narrower with increasing energy, with the reconstruction suffering from wider error margins below 10 TeV.

A fundamental part of the process involved in GraphNet requires defining IceCube events as graphs. On previous studies of neutrino events at very low energies (below 100 GeV) [69, 71, 72], the graphs are defined taking each pulse waveform as a node in the graph. However, the number of pulses increases drastically with energy, increasing the computational cost of training a GNN. Because we need to handle events above 10 TeV, we instead defined graphs by taking each DOM as a node, and summarize the detected pulses in percentiles for the activation time and charge build-up. Although this change of definition is useful and provides good results, we can anticipate that there is a loss of information that could significantly impact events where the amount of information is smaller, such as those containing a neutrino energy below 10 TeV.

With this in mind, a proposal could be made to change the Graph Definition to one that loses less information at lower energies. One potential solution is that we could reasonably expand the number of features by taking more than three percentiles in charge and time for the features of each node, which could be expected to increase precision across the whole energy range while maintaining computational costs reasonable. However, the impact for the precision of low energy events would be minimal.

A different approach could be considered to split the implementation of the algorithm into two architectures that would acct at two different energy regimes, one for low energy events and one for high energy events. A discriminator can be applied that takes a first approximation from the MuEx reconstruction to decide which Neural Network implementation to apply. This kind of solution would not only tackle the possible problem of loss of information in the input, but also smooth out the defects observed at the borders of the training datasets in Figure 6.4 for all energies. By training separate architectures, biases to energy ranges with higher statistics in the datasets could also be eliminated. Since every architecture would be trained solely on their designed energy range.

However, it should be noted that other methods also present problmes

A different approach one could consider is looking at the training sample. Increasing the richness of the samples at lower energies can expand upon the training of the neural network to account for biases in this range. Nevertheless, it should be noted that events below 10 TeV already compose the majority of the training sample in this work and increasing the number of events in this range might lead to overfitting, increasing the errors across the entire range rather than improving it. If an approach of this style is implemented, it should take care to increase statistics only in the underrepresented sections of the sample.

However, when considering the issues arising from the training dataset, we should remember that we use a smaller number of events when compared to previous implementations of GraphNet. This was done to save on computational times while maintaining significant statistics between events across the range. Improvements can be made by uniformly increasing training set statistics at all energies. This should however be implemented with care to avoid overfitting of results. Further studies should be performed to understand how the precision of the method changes as the training sample is increased. Nevertheless, in order to properly increase the train sample, GraphNet also needs to be adapted to work with lighter file formats. Although we improved on computational memory consumption, high energy files are large even before being loaded into training and this limits the training sample at this range.

Furthermore, we should remember that we applied our GNN architecture to simulated events. While, at the moment, there does not exist a rich enough sample to perform proper training of the neural network, the performance of the network on real data should be tested to confirm the results presented in this thesis.

Additional studies should also take into account and maximize the use computational resources, studying the feasibility of deployment conditions in the South Pole. Because GNNs can be implemented on multiple CPU cores, as well as GPU cores, the full computational implementation across different configurations of computational resources should be studied within the context of IceCube machines.

Finally, while in this work we have explored the implementation of the GNN architecture for energy reconstruction, previous studies have shown that this tool is powerful in the determination of other parameters such as angular direction and interaction vertex on low energies [69]. A similar work to the one performed on this thesis, extending the reconstruction of this parameters into higher energies can be expected to offer important insights into how GNN algorithms can be further implemented in IceCube.

In conclusion, this work has shown that the application of Graph Neural Networks across the relevant energy range for IceCube offers a promising approach to the reconstruction of a physical parameter such as energy. As GraphNet develops, the potential for GNNs for neutrino detectors will continue to show their power to reconstruct events is an efficient and fast wave. This becomes particularly relevant as IceCube continues to grow, increasing its capabilities for detection with further enhancements such as the coming IceCube Upgrade and the eventual construction of IceCube-Gen2. While these plans for the future of IceCube are encouraging in the landscape of neutrino detection, they also promise to increase both the volume of data and the complexity of detected events, therefore making the application of novel tools such as GNNs more and more relevant.

References

[1] Markus Ackermann et al. Fundamental Physics with High-Energy Cosmic Neutrinos. *Bull. Am. Astron. Soc.*, 51:215, 2019.

[2] M. G. Aartsen et al. Multimessenger observations of a flaring blazar coincident with high-energy neutrino IceCube-170922A. *Science*, 361(6398):eaat1378, 2018.

[3] Victor Hess. [translation to English] On the Observations of the Penetrating Radiation during Seven Balloon Flights. *arXiv:1808.02927*, 2018. Translated, commented by Alessandro De Angelis, Cornelia Arcaro b. Schultz.

[4] B. P. Abbott et al. Multi-messenger Observations of a Binary Neutron Star Merger. *Astrophys. J. Lett.*, 848(2):L12, 2017.

[5] Miroslav D Filipović, Jeffrey L Payne, and Nicholas F H Tothill. Multimessenger astronomy in practice: Celestial sources in action. In *Multimessenger Astronomy in Practice*, 2514-3433, pages 1–1 to 1–27. IOP Publishing, 2021.

[6] Alexander Aab et al. The Pierre Auger Cosmic Ray Observatory. *Nucl. Instrum. Meth. A*, 798:172–213, 2015.

[7] Alexander Aab et al. Depth of Maximum of Air-Shower Profiles at the Pierre Auger Observatory: Measurements at Energies above $10^{17.8}$ eV. *Phys. Rev. D*, 90(12):122005, 2014.

[8] Daisuke Ikeda and William Hanlon. Hybrid Measurement of the Energy Spectrum and Composition of Ultra-High Energy Cosmic Rays by the Telescope Array. *PoS*, ICRC2017:515, 2018.

[9] Carmelo Evoli. The cosmic-ray energy spectrum, December 2020.

[10] Thoudam, S., Rachen, J. P., van Vliet, A., Achterberg, A., Buitink, S., Falcke, H., and Hörandel, J. R. Cosmic-ray energy spectrum and composition up to the ankle: the case for a second galactic component. *A&A*, 595:A33, 2016.

[11] Kenneth Greisen. End to the cosmic-ray spectrum? *Phys. Rev. Lett.*, 16:748–750, Apr 1966.

[12] G. T. Zatsepin and V. A. Kuz'min. Upper Limit of the Spectrum of Cosmic Rays. *Soviet Journal of Experimental and Theoretical Physics Letters*, 4:78, August 1966.

[13] Particle Data Group, P A Zyla, et al. Review of Particle Physics. *Progress of Theoretical and Experimental Physics*, 2020(8):083C01, 08 2020.

[14] Pablo M. Bauleo and Julio Rodriguez Martino. The dawn of the particle astronomy era in ultra-high-energy cosmic rays. *Nature*, 458N7240:847–851, 2009.

[15] A. M. Bykov, D. C. Ellison, A. Marcowith, and S. M. Osipov. Cosmic ray production in supernovae. *Space Sci. Rev.*, 214(1):41, 2018.

[16] Maurizio Spurio. *Probes of Multimessenger Astrophysics. Charged cosmic rays, neutrinos, γ-rays and gravitational waves*. Astronomy and Astrophysics Library. Springer, 2018.

[17] Emma L. Alexander. Unified model of agn. https://emmaalexander.github.io/resources.html, 2022. This work is licensed under the Creative Commons Attribution 4.0 International License.

[18] J. Abraham et al. Correlation of the highest-energy cosmic rays with the positions of nearby active galactic nuclei. *Astropart. Phys.*, 29:188–204, 2008. [Erratum: Astropart.Phys. 30, 45 (2008)].

[19] Otto Claus Allkofer and Peter Karl Grieder. *Cosmic rays on earth*. Fachinformationszentrum Energie, Physik, Mathematik Karlsruhe, 1984.

[20] Hiroyuki Sagawa and ByungGu Cheon. The status and prospect of Telescope Array Experiment. *AAPPS Bull.*, 23(2):15–22, 2013.

[21] W. B. Atwood et al. The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission. *The Astrophysical Journal*, 697(2):1071–1102, June 2009.

[22] M. Ajello et al. 3FHL: The Third Catalog of Hard Fermi-LAT Sources. *Astrophys. J. Suppl.*, 232(2):18, 2017.

[23] The Magic Collaboration. The major upgrade of the magic telescopes, part i: The hardware improvements and the commissioning of the system. *Astroparticle Physics*, 72:61–75, January 2016.

[24] The Magic Collaboration. The major upgrade of the magic telescopes, part ii: A performance study using observations of the crab nebula. *Astroparticle Physics*, 72:76–94, January 2016.

[25] Mathieu de Naurois. H.E.S.S.-II - Gamma ray astronomy from 20 GeV to hundreds of TeV's. In *European Physical Journal Web of Conferences*, volume 136 of *European Physical Journal Web of Conferences*, page 03001, March 2017.

[26] J. Holder et al. Status of the VERITAS Observatory. In Felix A. Aharonian, Werner Hofmann, and Frank Rieger, editors, *American Institute of Physics Conference Series*, volume 1085 of *American Institute of Physics Conference Series*, pages 657–660. AIP, December 2008.

[27] Andrea Addazi et al. The Large High Altitude Air Shower Observatory (LHAASO) Science Book (2021 Edition). *Chin. Phys. C*, 46:035001–035007, 2022.

[28] Zhen Cao et al. Ultrahigh-energy photons up to 1.4 petaelectronvolts from 12 γ-ray Galactic sources. *Nature*, 594(7861):33–36, June 2021.

[29] Morihiro Honda, T. Kajita, K. Kasahara, S. Midorikawa, and T. Sanuki. Calculation of atmospheric neutrino flux using the interaction model calibrated with atmospheric muon data. *Phys. Rev. D*, 75:043006, 2007.

[30] M. G. Aartsen et al. Development of a General Analysis and Unfolding Scheme and its Application to Measure the Energy Spectrum of Atmospheric Neutrinos with IceCube. *Eur. Phys. J. C*, 75(3):116, 2015.

[31] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Physical Review Letters*, 81(8):1562–1567, August 1998.

[32] Edoardo Vitagliano, Irene Tamborra, and Georg Raffelt. Grand Unified Neutrino Spectrum at Earth: Sources and Spectral Components. *Rev. Mod. Phys.*, 92:45006, 2020.

[33] Eli Waxman and John N. Bahcall. High-energy neutrinos from astrophysical sources: An Upper bound. *Phys. Rev. D*, 59:023002, 1999.

[34] M. Ageron et al. The antares telescope neutrino alert system. *Astroparticle Physics*, 35(8):530–536, March 2012.

[35] Annarita Margiotta. The km3net deep-sea neutrino telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 766:83–87, December 2014.

[36] M. G. Aartsen et al. The IceCube Neutrino Observatory: Instrumentation and Online Systems. *JINST*, 12(03):P03012, 2017.

[37] M.G. Aartsen et al. Measurement of south pole ice transparency with the icecube led calibration system. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 711:73–89, May 2013.

[38] Ryan Bay, Robert Rohde, and Nathan Bramall. South pole paleowind from automated synthesis of ice core records. *J. Geophys. Res*, 115, 07 2010.

[39] R. Abbasi et al. The design and performance of icecube deepcore. *Astroparticle Physics*, 35(10):615–624, May 2012.

[40] R. Abbasi et al. The IceCube Data Acquisition System: Signal Capture, Digitization, and Timestamping. *Nucl. Instrum. Meth. A*, 601:294–316, 2009.

[41] M. G. Aartsen et al. Search for Sources of Astrophysical Neutrinos Using Seven Years of IceCube Cascade Events. *Astrophys. J.*, 886:12, 2019.

[42] Steve Sclafani and Mirco Hünnefeld. A search for neutrino sources with cascade events in icecube, 2021.

[43] V.J. Stenger. Track Fitting for DUMAND-II Octagon Array. *DUMAND External Report HDC-1-90*, 1990.

[44] Thomas Kintscher. *Rapid Response to Extraordinary Events: Transient Neutrino Sources with the IceCube Experiment*. PhD thesis, Humboldt-UniversitÃ¤t zu Berlin, Mathematisch-Naturwissenschaftliche FakultÃ¤t, 2020.

[45] Nathan Whitehorn, Jakob van Santen, and Sven Lafebre. Penalized splines for smooth representation of high-dimensional monte carlo datasets. *Computer Physics Communications*, 184(9):2214–2220, September 2013.

[46] Kai Schatto. *Stacked searches for high-energy neutrinos from blazars with IceCube*. PhD thesis, Mainz U., 6 2014.

[47] R. Abbasi et al. A muon-track reconstruction exploiting stochastic losses for large-scale cherenkov detectors. *Journal of Instrumentation*, 16(08):P08034, August 2021.

[48] Harald Cramér. *Mathematical Methods of Statistics (PMS-9)*. Princeton University Press, 1999.

[49] C. Radhakrishna Rao. *Information and the Accuracy Attainable in the Estimation of Statistical Parameters*, pages 235–247. Springer New York, New York, NY, 1992.

[50] Till Neunhoffer. Estimating the angular resolution of tracks in neutrino telescopes based on a likelihood analysis. *Astropart. Phys.*, 25:220–225, 2006.

[51] B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1):1 – 26, 1979.

[52] M G Aartsen et al. Energy reconstruction methods in the icecube neutrino telescope. *Journal of Instrumentation*, 9(03):P03009–P03009, March 2014.

[53] R. Abbasi et al. An improved method for measuring muon energy using the truncated mean of de/dx. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 703:190–198, March 2013.

[54] R. Abbasi et al. A convolutional neural network based cascade reconstruction for the icecube neutrino observatory. *Journal of Instrumentation*, 16(07):P07041, July 2021.

[55] M. G. Aartsen et al. The IceCube Realtime Alert System. *Astropart. Phys.*, 92:30–41, 2017.

[56] Hugo A. Ayala Solares, Stephane Coutu, D.F. Cowen, James J. DeLaunay, Derek B. Fox, Azadeh Keivani, Miguel Mostafá, Kohta Murase, Foteini Oikonomou, Monica Seglar-Arroyo, Gordana Tešić, and Colin F. Turley. The astrophysical multimessenger observatory network (amon): Performance and science program. *Astroparticle Physics*, 114:68–76, 2020.

[57] Gcn/amon notices. https://gcn.gsfc.nasa.gov/amon.html. accessed: 08.08.2024.

[58] M. G. Aartsen et al. Very High-Energy Gamma-Ray Follow-Up Program Using Neutrino Triggers from IceCube. *JINST*, 11(11):P11009, 2016.

[59] Markus Voge. *Searches for Neutrinos from Supernovae Using Cherenkov In-Ice Detectors*. PhD thesis, Bonn U., 9 2016.

[60] Sarah Mancina. *Astrophysical neutrino source searches using IceCube starting tracks*. PhD thesis, U. Wisconsin, Madison (main), 2022.

[61] Frederic B. Fitch. Warren s. mcculloch and walter pitts. a logical calculus of the ideas immanent in nervous activity. bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. *Journal of Symbolic Logic*, 9(2):49–50, 1944.

[62] Kavya Saravanan and Abbas Z. Kouzani. Advancements in on-device deep neural networks. *Information*, 14(8), 2023.

[63] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Neural Networks*, pages 389–416. Springer New York, New York, NY, 2009.

[64] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[65] Andreas Søgaard, Rasmus F. Ørsøe, Morten Holm, Leon Bozianu, Aske Rosted, Troels C. Petersen, Kaare Endrup Iversen, Andreas Hermansen, Tim Guggenmos, Peter Andresen, Martin Ha Minh, Ludwig Neste, Moust Holmes, Axel Pontén, Kayla Leonard DeHolton, and Philipp Eller. Graphnet: Graph neural networks for neutrino telescope event reconstruction. *Journal of Open Source Software*, 8(85):4971, 2023.

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv e-prints*, page arXiv:1912.01703, December 2019.

[67] WA Falcon. Pytorch lightning. *GitHub. Note: https://github. com/williamFalcon/pytorch-lightning Cited by*, 3, 2019.

[68] J Pivarski et al. scikit-hep/awkward-array: 0.13.0. https://doi.org/10.5281/zenodo.3952674, July 2020.

[69] R. Abbasi et al. Graph neural networks for low-energy event classification & reconstruction in icecube. *Journal of Instrumentation*, 17(11):P11003, November 2022.

[70] R. Abbasi et al. Low energy event reconstruction in IceCube DeepCore. *Eur. Phys. J. C*, 82(9):807, 2022.

[71] Philipp Eller, Kayla Leonard DeHolton, Jan Weldert, and Rasmus Ørsøe. Sensitivity of the icecube upgrade to atmospheric neutrino oscillations, 2023.

[72] Chiara Bellenghi, Martin Ha Minh, Tomas Kontrimas, Elena Manao, Rasmus Orsoe, and for The IceCube Collaboration Wolf, Martin. Extending the IceCube search for neutrino point sources in the Northern sky with additional years of data. *PoS*, ICRC2023:1060, 2023.

[73] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 1–7, 2020.

[74] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[75] A. Gazizov and M. Kowalski. Anis: High energy neutrino generator for neutrino telescopes. *Computer Physics Communications*, 172(3):203–213, November 2005.

Acknowledgments

First of all, I would like to prepare the reader, this is going to be long. The work presented here is the culmination of two years full of intense physical, emotional and academic labour.

In the first place I would like to thank Professor Elisa Bernardini and Dr. Sarah Mancina who have been nothing short of wonderful mentors in the world of multi-messenger astrophysics. I came to UniPD looking to get close to the IceCube Experiment and high-energy astroparticle physics and they have taught me so much of what I take forward in my career.

Next, I need to thank my parents, who have gone through the most difficult task in the world: accepting to have to spend christmas's and birthdays faraway from not one but two children as they support them following their dreams on the other side of the world. In that same vein, I thank my sister, Dr. Valentina Coloma Borja for bringing me the news and updates of the world back in Ecuador and as we support each other while we face the challenges to come.

I thank Ester, Emilio and Juanpi, who have shown me that as we grow up and adult life brings us all over the world, the ties that bind us are forged stronger everyday. I Similarly thank Emilio, Dome and Azul who constantly show me that true friendship sometimes means calling to catch up at 3am despite the time difference. You can't leave home without your roots.

Thank You to Pauli, who gave me the strength I needed before moving to Italy and paved a lot of the way to go in the adventure that was to come.

I give thanks to Martin, who pushed me to stop my nonsense and choose to commit to coming to Italy when I needed it the most; to Jonathan who, even two countries away, for the past two years has given me an avenue to remember that the best parts of who I was in Ecuador come with me wherever I go, including those dear to me. Both of them and the sensei Alejo, my dear CVV, have given me a solid base to make sure I am at my best anywhere in the world.

I need to offer my gratitude to Dora, Alba, Marta, Deniz, Begüm and especially Aleyna. For making me feel warmth when i first arrived in Padova, and giving me spaces in parts of the world I could never imagine where I know I will be welcome. Similarly thanks of Tabea for showing me a smile and a perspective of life, adventure, friendship and trust that we share no matter where in the world we are. All of you guys carry important pieces of my hearts with you wherever life takes you next.

I give thanks to Catita, who received me with open arms in Padova when I first arrived, creating an Ecuadorian home to turn to faraway from the homeland. To Sol, from the Paraguayan conspiracy, for giving me and everyone around her a beautiful smile and a space to belong and for everything she has taught me through example. To Claudia Valentina and to Anapaula for showing me how to find family faraway from home, we always do our best support to each other, even in confusing times. Equal thanks to Camilo, Milena, Leo, Kevin, Marcelo, Ana and the rest of the latino community here in Padova for giving me a homely space away for home to laugh, hugh and connect with our culture and community sentiment.

I need to mention Josue, Diego R, and Edgar, Jimmy and Lui and Raul, Francisco, Alegria, Damian and Karina. All of whom have given me reasons to know that there is always someone waiting for me back at the place where the adventure started.

I would be remiss if I did not mention my dear friends from Pollaio and NCS, who have been with me through exams, presentations and research here at UniPD. Endless nights with a glass of beer and conversation while we go through this life together. Thanks to Denis, Francesca, Gabrielle, and Alessandro, as well as Tima, Leon and Alireza. You guys have been a support system so annoyingly effective you are even here to pick me up even when I least expect it.

An important part of this journey has been developed in the dance floor. For tango dancing has been there for me as an avenue to express myself, to exit the world for a second and re-contextualize the worst of times in 2x4 beats. Because of that I thank my dance partner Valentina, I will always remember our embraces when you become a world-class dancer and I am telling my children that I was lucky enough to dance with you for a spell, Similarly I thank Anna for the peaceful relaxing space I can enter when we dance. Thank you to Nathan and Luis and Galo who initiated me in the world of tango, to Carlos and Sofiani, for helping me improve my bases so much that I can show Italy what Ecuadorian dancers are made of; and Silvio, Marta and Alberto who have given me a tango family so far away from home.

Deep thanks to Güendy, the biggest ray of sunshine in Padova packed into the smallest Quiteña package, who I seem to have always been destined to meet. To Fernanda, that has been a support, a lifeline, a shoulder to cry on and a friend to vent in my best and worst times. And of course Dieguito, who shows me sometimes the truest comfortable places to be can be found in peruvian multiversal variants of yourself that share your name and your birthday. You guys are some of the most splendid solid rocks I could have never wished for.

And Finally, thanks to myself. You have gone through every challenge, not always gracefully, and not always successfully, but you have built something wonderful that I promise to care for in the years to come.