Cosmic ray composition measurement using Graph Neural Networks for KM3NeT/ORCA

Messung der Zusammensetzung von kosmischer Strahlung mit Graph-basierten neuronalen Netzwerken in KM3NeT/ORCA

Der Naturwissenschaftlichen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg

zur
Erlangung des Doktorgrades Dr. rer. nat.

vorgelegt von

Stefan Reck

aus Neustadt an der Aisch

Als Dissertation genehmigt
von der Naturwissenschaftlichen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung: ???

Vorsitzender der Promotionskommision: ???

Gutachter: Prof. Dr. Uli Katz
???

Contents

Contents 4

1. Abstract 6

Zusammenfassung 7

2. Cosmic rays in KM3NeT 8
- 2.1. Cosmic rays . . . 8
  - 2.1.1. Energy spectrum . . . 8
  - 2.1.2. Measurements . . . 9
  - 2.1.3. Composition . . . 10
  - 2.1.4. Origin . . . 14
  - 2.1.5. Acceleration . . . 17
- 2.2. Atmospheric showers . . . 19
  - 2.2.1. Hadronic shower . . . 19
  - 2.2.2. Electromagnetic shower . . . 20
- 2.3. Interaction of muons with matter . . . 20
  - 2.3.1. Ionization and excitation . . . 21
  - 2.3.2. Radiation . . . 22
  - 2.3.3. Deflection . . . 22

3. Detection of particles in KM3NeT 23
- 3.1. The KM3NeT detectors: ORCA and ARCA . . . 23
- 3.2. Cherenkov radiation . . . 25
- 3.3. Event signatures . . . 26
- 3.4. Other large volume neutrino detectors . . . 27

4. Processing chain for atmospheric muons 29
- 4.1. Event generation . . . 29
  - 4.1.1. Corsika . . . 29
  - 4.1.2. Mupage . . . 31
- 4.2. Light propagation and detector response . . . 32
- 4.3. Trigger . . . 33
- 4.4. Classical reconstruction . . . 35

5. Introduction to deep learning 36
- 5.1. Neurons and layers . . . 36
- 5.2. Backpropagation and gradient descent . . . 37
- 5.3. Activation functions . . . 40
- 5.4. Cost functions . . . 41
- 5.5. Optimizers . . . 42
- 5.6. Estimating uncertainty in regressions . . . 44

5.7. Deep learning software in KM3NeT 45
5.7.1. OrcaSong 46
5.7.2. OrcaNet 46

6. Graph neural networks 48
6.1. Basics of graph theory 48
6.2. Convolutional graph networks - spectral and spatial 48
6.3. Transforming KM3NeT data to graphs 50
6.4. ParticleNet and the Edge Convolutional block 52
6.5. Advantages and limitations compared to convolutional networks 55

7. Muon bundle reconstruction with Graph Neural Networks 57
7.1. Model design and training setup 57
7.2. Reconstruction of the incident direction 59
7.2.1. Performance evaluation on simulations 59
7.2.2. Uncertainty estimation 62
7.2.3. Data-Monte Carlo comparison 64
7.3. Reconstruction of the bundle diameter 69
7.3.1. Definition 69
7.3.2. Performance evaluation on simulations 70
7.3.3. Biases in reconstructions 73
7.3.4. Data-Monte Carlo comparison 75
7.4. Reconstruction of the muon multiplicity 78
7.4.1. Architecture 79
7.4.2. Performance evaluation on simulations 81
7.4.3. Data-Monte Carlo comparison 83

8. Cosmic ray composition measurement 89
8.1. Corsika dataset 89
8.2. Performance of the GNNs on Corsika 90
8.2.1. Direction 90
8.2.2. Diameter 92
8.2.3. Multiplicity 94
8.3. Connection between bundle observables and the primary mass 96
8.4. Composition measurement 99
8.4.1. Muon multiplicity and zenith angle 99
8.4.2. Bundle diameter 105
8.4.3. Seasonal variation 107
8.4.4. Discussion 112

9. Summary and Outlook 115

Bibliography 116

List of Figures 124

Appendix A. Configuration files 128
A.1. OrcaSong 128
A.2. OrcaNet 129

Acknowledgments 132

Chapter 1. Abstract

The chemical composition of cosmic rays arriving at the Earth's atmosphere at high energies in the PeV region and beyond has been the research target of various experiments, but is still subject to large uncertainties. A precise understanding of the high energy composition allows to constrain the origin of cosmic rays and the processes by which such high energies can be reached in astronomical accelerators. Highly energetic cosmic rays are measured indirectly, for which detailed simulations of air showers and the particle interactions within are required. Recent measurements at LHC have lead to a new generation of high energy interaction models, which allow to simulate the expected cosmic ray composition at higher energies than ever before. However, significant disagreements between these calculations and the measurements from several experiments which study cosmic rays via the atmospheric muons they produce in air showers have been found. This thesis examines the potential of KM3NeT/ORCA to measure the cosmic ray composition, which will allow it to contribute to the aforementioned topics.

KM3NeT/ORCA is a water-Cherenkov neutrino detector, currently under construction in the Mediterranean Sea at a depth of 2450 meters. The project's main goal is the determination of the neutrino mass ordering by measuring the energy- and zenith-angle-resolved oscillation probabilities of atmospheric neutrinos traversing the Earth. However, the majority of the particles observed by the detector are atmospheric muons. While they are a source of background for neutrino measurements, they can also be used to indirectly study the properties of extensive air showers and cosmic ray particles. Due to the high rate at which KM3NeT/ORCA detects atmospheric muons and the high density of its instrumentation, it is especially well suited to study the properties of muon bundle events, in which multiple muons traverse the detector simultaneously. These events are essential for investigating the properties of primary particles.

Among the observables of muon bundles that allow conclusions to be drawn about the primary composition are the number of muons in the event, their lateral spread, as well as their incident direction. However, no dedicated methods for reconstructing these properties exist in KM3NeT so far. In this work, a set of novel reconstructions is developed, which makes use of deep artificial neural networks to analyze low-level experimental data and directly reconstruct the desired properties, independent of existing approaches. In the course of this, the new archetype of graph neural networks is introduced to KM3NeT. By representing the data of the detector as the mathematical structure of graphs, the measured information can be treated more efficiently and with a higher precision than with previous deep learning architectures used in KM3NeT.

Detailed comparisons between the performance on simulations and on measured data for a small version of the detector show decent agreement. In the case of the directional reconstruction, the result from deep learning is compared to the established approach for single muon events. It is found that both approaches produce virtually the same result on measured data. This is an important step towards building trust in the method of deep learning, as these studies are the first time that deep neural networks are applied to measured data in KM3NeT.

Using these new reconstructions, for the first time in KM3NeT an indirect measurement of the mass composition of cosmic rays is successfully performed on six months of measured data. With a dedicated set of CORSIKA simulations, the fraction of proton, helium and iron of the overall, energy-averaged flux is measured. A detailed discussion of systematic uncertainties stemming from different energy flux models and the seasonal variation of the muon flux is included.

Zusammenfassung

Die chemische Zusammensetzung der kosmischen Strahlung, die die Erdatmosphäre mit hohen Energien im PeV-Bereich und darüber hinaus erreicht, ist zwar in vielen Experimente untersucht worden, aber dennoch immer noch mit großen Unsicherheiten behaftet. Ein genaues Verständnis der Zusammensetzung bei hohen Energien ermöglicht es, den Ursprung der kosmischen Strahlung und die Prozesse einzugrenzen, durch die solch hohe Energien in astronomischen Beschleunigern erreicht werden können. Hochenergetische kosmische Strahlung wird indirekt gemessen, wofür detaillierte Simulationen der Luftschauer und der darin stattfindenden Teilchenwechselwirkungen erforderlich sind. Jüngste Messungen am LHC haben zu einer neuen Generation von Hochenergie-Wechselwirkungsmodellen geführt, mit denen sich die erwartete Zusammensetzung der kosmischen Strahlung bei höheren Energien als je zuvor simulieren lässt. Es wurden jedoch erhebliche Diskrepanzen zwischen diesen Simulationen und den Messungen mehrerer Experimente festgestellt, bei denen die kosmische Strahlung über die atmosphärischen Myonen untersucht wird, die sie in Luftschauern erzeugen. In dieser Arbeit wird das Potenzial von KM3NeT/ORCA untersucht, die Zusammensetzung der kosmischen Strahlung zu messen und damit einen Beitrag zu den oben genannten Themen zu leisten.

KM3NeT/ORCA ist ein Wasser-Cherenkov-Neutrino-Detektor, der derzeit im Mittelmeer in einer Tiefe von 2450 Metern gebaut wird. Das Hauptziel des Projekts ist die Bestimmung der Neutrinomassenordnung durch Messung der energie- und zenitwinkelaufgelösten Oszillationswahrscheinlichkeiten von atmosphärischen Neutrinos, die die Erde durchqueren. Die meisten der vom Detektor beobachteten Teilchen sind jedoch atmosphärische Myonen. Sie sind nicht nur Hintergrund für Neutrinomessungen, sondern können auch zur indirekten Untersuchung ausgedehnter Luftschauer und kosmischer Strahlungsteilchen verwendet werden. Aufgrund der hohen Rate, mit der KM3NeT/ORCA atmosphärische Myonen nachweist, und der hohen Dichte seiner Instrumentierung ist er besonders gut geeignet, die Eigenschaften von Myonenbündel-Ereignissen zu untersuchen, bei denen mehrere Myonen den Detektor gleichzeitig durchqueren. Diese Ereignisse sind für die Untersuchung der Eigenschaften von Primärteilchen essentiell.

Zu den Observablen von Myonenbündeln, die Rückschlüsse auf die Zusammensetzung der kosmischen Strahlung zulassen, gehören die Anzahl der Myonen im Ereignis, ihre laterale Ausbreitung sowie ihre Einfallsrichtung. Allerdings gibt es in KM3NeT bisher keine dezidierten Methoden zur Rekonstruktion dieser Observablen. In dieser Arbeit wird eine Reihe neuartiger Rekonstruktionen entwickelt, die tiefe künstliche neuronale Netze nutzen, um experimentelle Daten direkt auf niedriger Ebene zu analysieren und die gewünschten Eigenschaften zu rekonstruieren, unabhängig von bestehenden Ansätzen. Im Zuge dessen wird der neue Archetyp der Graph-basierten neuronalen Netze in KM3NeT eingeführt. Indem die Daten des Detektors als mathematische Struktur von Graphen dargestellt werden, können die gemessenen Informationen effizienter und präziser behandelt werden als mit den bisher in KM3NeT verwendeten Deep-Learning-Architekturen.

Detaillierte Vergleiche zwischen der Leistung auf Simulationen und auf gemessenen Daten für eine kleine Version des Detektors zeigen eine gute Übereinstimmung. Im Fall der Richtungsrekonstruktion wird das Ergebnis von Deep Learning mit dem etablierten Ansatz für einzelne Myonenereignisse verglichen. Es zeigt sich, dass beide Ansätze bei gemessenen Daten praktisch das gleiche Ergebnis liefern. Dies ist ein wichtiger Schritt, um Vertrauen in die Methode des Deep Learning aufzubauen, da diese Studien die erste Anwendung von tiefen neuronalen Netzen auf gemessene Daten in KM3NeT sind.

Unter Verwendung der neuen Rekonstruktionen wird zum ersten Mal in KM3NeT eine Messung der Massenzusammensetzung von kosmischer Strahlung anhand von sechs Monaten an Messdaten durchgeführt. Mit einem speziellen Satz von CORSIKA-Simulationen wird der Anteil von Protonen, Helium und Eisen am gesamten, energiegemittelten Fluss gemessen. Eine ausführliche Diskussion der systematischen Unsicherheiten, die sich aus den verschiedenen Energieflussmodellen ergeben, und der saisonalen Schwankungen des Myonenflusses ist ebenfalls enthalten.

Chapter 2. Cosmic rays in KM3NeT

The majority of cosmic rays are ionized atoms with high energies arriving at vast rates of about 1000 particles per square meter and second at the Earth's atmosphere. Their origin is not fully understood, but most of the particles are assumed to be produced outside of the solar system, or even the Milky Way as a whole [1]. Relevant observables to study are their incident direction, their energy, as well as their elemental composition. Since the cosmic rays are charged, they get deflected by magnetic fields encountered along their path through the extragalactic, galactic and solar space, so they show very high levels of isotropy in their arrival direction. However, slight anisotropic irregularities up to a few percent at the highest energies have been reported [2].

The energy distribution and the composition can give important hints about which type of astrophysical objects produce the cosmic rays, as well as the underlying acceleration mechanism. As cosmic rays enter the atmosphere, they cause atmospheric showers in which large amounts of muons are generated. Despite the loss of energy during their propagation through the atmosphere, some of these muons can reach deep below the Sea level, and can therefore be measured by underwater Cherenkov detectors like KM3NeT [3].

This chapter gives an overview over fundamental concepts of cosmic rays, existing experimental results, and explores how KM3NeT can be used to study them indirectly. It is inspired by the excellent books from Angelis and Pimenta [4], as well as Gaisser, Engel, and Resconi [5], and the extensive review of cosmic ray physics by Becker Tjus and Merten [1].

2.1. Cosmic rays

2.1.1. Energy spectrum

Numerous experiments have measured the cosmic ray energy spectrum over the years, and have shown it to cover an impressive interval of energies, as well as fluxes. The observed energies range from a few GeV per primary particle up to hundreds of EeV (=10^{11} GeV), with the flux being as high as hundreds of particles per square meter and second for low energies, down to merely a few particles per square kilometer and year at the high end of the energy spectrum.

The energy spectrum is remarkably smooth, and can in large parts be described by inverse power laws of the form $F(E) = A \cdot E^{\gamma}$. The exponent γ, also referred to as the spectral index, changes slightly a few times throughout the energy range, but remains somewhat constant in between.

For energies above a few 10 GeV up to 10^6 GeV, γ is about -2.7. Afterwards, at an energy of about $3 \cdot 10^6$ GeV, a transition to a steeper falling curve is observed, commonly referred to as the *knee* of the cosmic ray spectrum. This steeper curve shows a spectral index of $\gamma \approx -3.1$ between energies of 10^7 GeV up to 10^9 GeV. Then, at about $3 \cdot 10^9$ GeV, yet another transition known as the *ankle* can be observed, back to a shallower slope of $\gamma \approx -2.6$. Finally, a cutoff in the rate occurs at very high energies of 10^{11} GeV.

The cause for this cut-off is still an open question. A possible explanation is that protons with energies above the observed maximum are energetic enough to interact with photons of the microwave background. As a result, they would produce a pion and lose a significant amount of energy in the process, staying under the so-called Greisen–Zatsepin–Kuzmin energy limit [6].

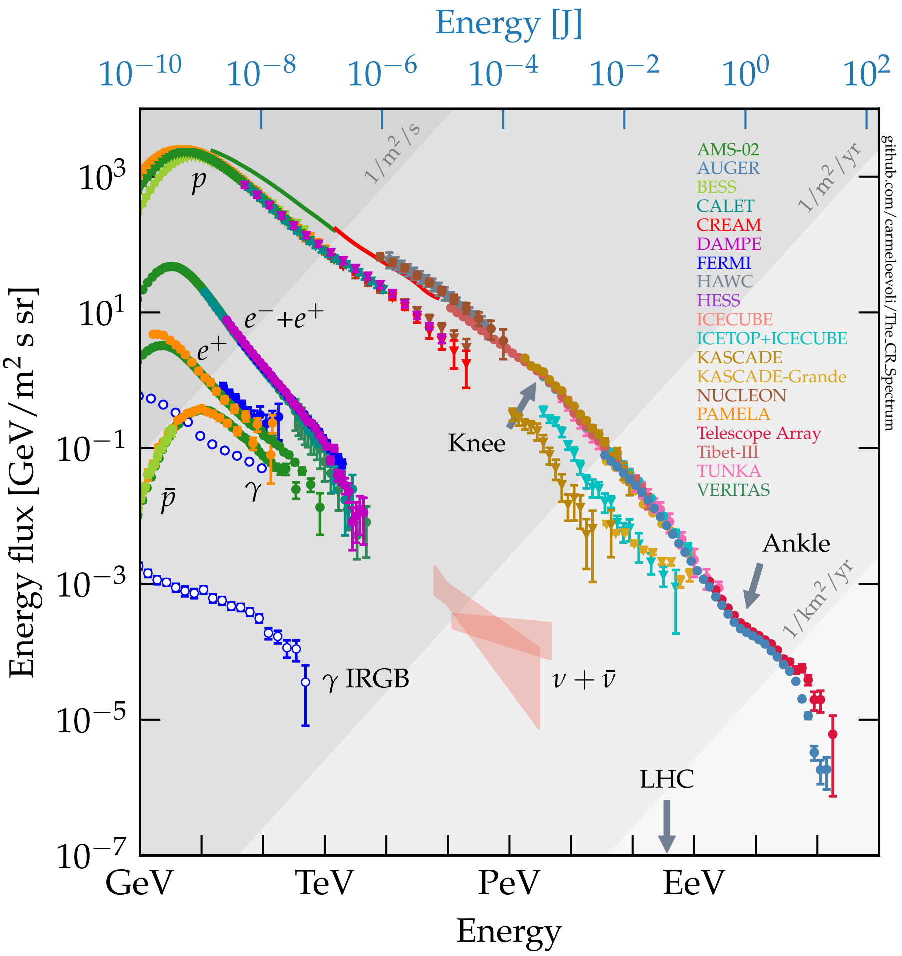


Figure 2.1.: The cosmic ray energy flux multiplied by E^2, obtained from measurements of various experiments since the year 2000. Unless indicated otherwise, the measurements show the all-particle flux [7].

Alternatively, the cut-off might also be intrinsic to the acceleration process from which the cosmic rays originate.

In order to make the change of the exponent in such steeply falling power laws more visible, the flux can be multiplied by a power of the energy for graphical displays. Different measurements of the particle flux multiplied by E^2 (i.e. the energy flux) are shown in Figure 2.1.

2.1.2. Measurements

There is a variety of different strategies in order to measure cosmic rays. If the goal is to identify possible sources via the arrival direction, it is actually reasonable to not even investigate the charged particles themselves, but look for other particles that might get produced in the same object, but do not get deflected as much by the dust and fields along the way to the Earth. Some experiments like H.E.S.S. [8] therefore look for highly energetic photons. Others, like IceCube (see section 3.4) or KM3NeT itself, try to identify astrophysical neutrinos, which are not only unaffected by electromagnetic fields, but also have a very low chance of being stopped by obstructing matter.

However, many experiments can also measure the charged cosmic ray particles. As shown in subsection 2.1.1, the energy distribution of cosmic rays spans over many orders of magnitude, so different techniques of measurements are necessary to cover this vast scale.

Direct measurement

At lower energies up to about the knee, the flux of cosmic rays is so high that even relatively small detection surfaces can obtain sufficiently high statistics. Therefore, direct measurements of the particles before they dive into Earth's atmosphere are possible. For this, the detectors can be mounted on a satellite or space station, which are in orbit around the Earth. Examples include magnetic spectrometers like AMS-02, located at the International Space Station [9], or the PAMELA [10] satellite-based experiment. A possibility for direct measurements at the higher energies in this region are calorimeters without magnetic spectrometers. In the CREAM experiment, such a device was lifted by a large balloon and conducted its measurements during several flights over Antarctica [11].

Indirect Measurement

For higher energies at and above the knee, the low flux requires detection areas that are too large for detectors high off the ground. For that reason, these cosmic rays are detected indirectly by ground based experiments via the extensive air showers the particles cause when entering the atmosphere. The air showers are discussed to greater detail in section 2.2.

The electromagnetic and muonic component of air showers can be measured by an array of small detection stations arranged on the surface. In the TUNKA-grande experiment [12], scintillation detectors are used for this, while IceTop (see section 3.4) consists of containers filled with ice, in which crossing charged particles induce Cherenkov light. Water Cherenkov detectors below the surface, like KM3NeT or the in-ice detector of IceCube, are only sensitive to the muonic component of the air showers, as well as the neutrinos.

The Pierre Auger observatory [13] measures cosmic rays up to the highest energies at the ankle and beyond. It has a large instrumented area of about 3000 square kilometers, since the event rate is extremely low in that region. Two different ways of measuring air showers are used in Pierre Auger: an array of tanks filled with water to detect Cherenkov radiation, and an array of fluorescence detectors, which look for the ultraviolet light emitted by atmospheric nitrogen excited by passing charged particles.

2.1.3. Composition

Most of the charged cosmic ray particles arriving at the Earth are protons (i.e. hydrogen nuclei) and, to a much smaller extent of about 10%, nuclei of heavier atoms. A small fraction of the cosmic rays also comes in the form of electrons, positrons and an even smaller fraction of antiprotons, which have been measured in the GeV energy range among others by the AMS02 experiment [14].

While it is possible to measure the composition directly at lower energies using high altitude experiments, this becomes ineffective at energies above a few hundred GeV (see subsection 2.1.2). Instead, it then has to be studied indirectly via the extensive air showers caused by the cosmic rays as they enter the atmosphere. In the shower, many different particles like hadrons, electrons, muons and neutrinos get produced as parts of electromagnetic and hadronic cascades. Naturally, the composition of particles changes dramatically as the shower develops, with muons and neutrinos becoming increasingly dominant for lower heights.

Obtaining the primary composition from this is therefore not trivial. Two important observables that allow for conclusions to be drawn about the composition are the number of produced muons, and the depth of the maximum of the shower X_{max}. The latter is defined as the atmospheric depth at which the air shower reaches its highest number of particles and is typically measured with detectors for the electromagnetic component of the showers at the surface. Since these two observables are measured using different components of the shower, they are sometimes considered to be independent methods for studying the primary cosmic rays [15].

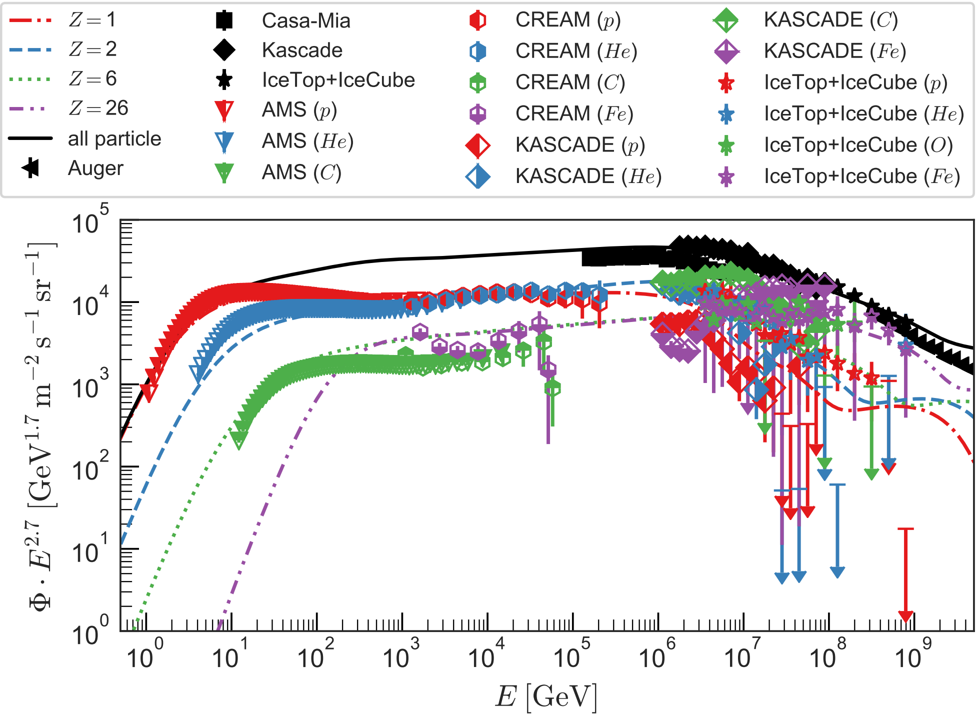


Figure 2.2.: Cosmic ray primary flux up to the ankle multiplied by energy to the power of 2.7 over their energy. The dots show measurements from selected experiments, while the lines are calculated from the combined Gaisser-Honda-Hillas flux model [16]. The all particle flux is plotted in black, and the colors represent different primaries [1].

As mentioned before, most cosmic rays are protons, which are dominant at lower energies in the GeV range. However, at increasing energies where the flux is much lower, the composition undergoes drastic changes, and heavier nuclei play a more significant role. As can be seen in Figure 2.2, helium nuclei supersede protons as the most abundant cosmic ray when getting closer to the knee. As the energy increases, the contributions to the flux start to be dominated by heavier nuclei like carbon or oxygen, while iron is the dominant contribution at the energies closer to the ankle.

Figure 2.2 shows that the uncertainties of the composition for energies at 10^6 GeV and beyond are quite substantial. Since the cosmic ray primaries can only be measured indirectly in this energy range, detailed simulations of the air shower are required in order to infer the mass composition. A large source of uncertainty in these simulations is the modeling of highly energetic hadronic interactions. Collider experiments allow to constrain the models, but since cosmic rays can have energies higher than what can be reached in current accelerators, the models need to be extrapolated. Recent data from the LHC reaching as high as 10^8 GeV was taken into account for the current generation of interaction models [17].

The muon puzzle

While the mass distribution predicted by these models is in decent agreement with measurements based on the depth of the shower maximum X_{max}, a significant difference was found in measurements from numerous experiments that use the muon multiplicity of bundles for inferring the mass. An excess of high multiplicity events at energies above 10^7 GeV was identified, suggesting a high rate of heavy nuclei in this energy range. This is in disagreement to the air shower simulations using post-LHC models and the X_{max} measurements. The reason for this

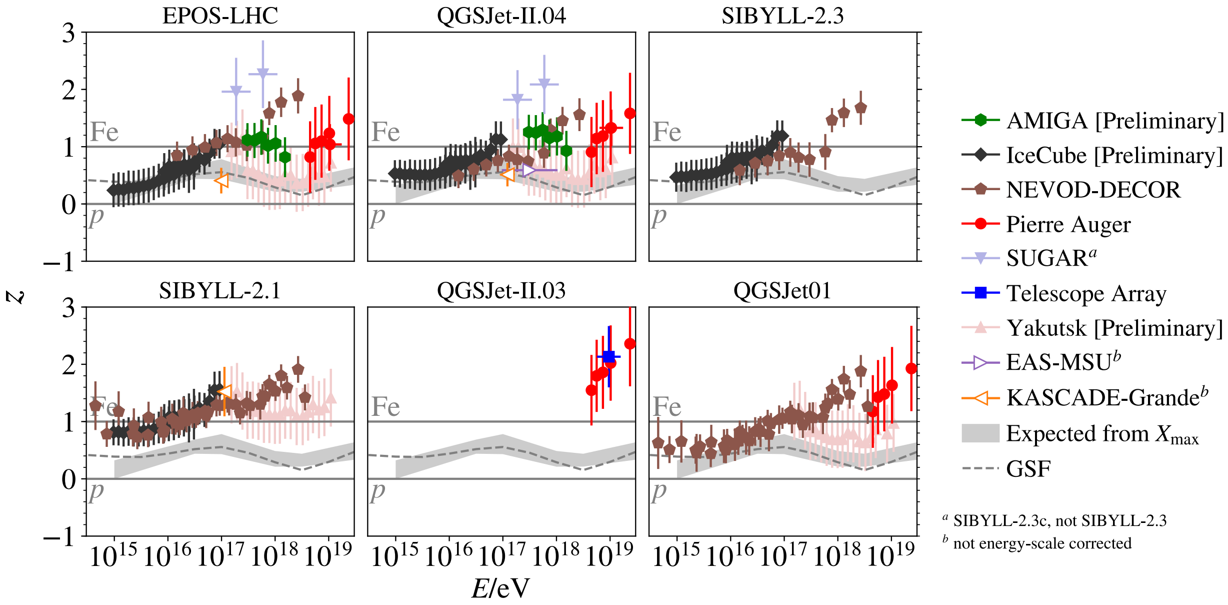


Figure 2.3.: The z scale measuring the logarithmic muon multiplicity plotted over the calibrated primary energy. The colored dots are measurements from various experiments. The grey band represents the expectation of z based on measurements of X_{max}, while the dashed line is the prediction obtained from simulations using the GSF flux model [19]. The z scale is calculated using the hadronic interaction model given in the title of each plot. The first row shows post-LHC models, while the bottom row shows their predecessors. Plot taken from [15].

inconsistency is currently under debate, and is referred to as the muon puzzle.

In a recent meta analysis, the distributions of the muon multiplicity measured by various experiments were compared to the expectations from simulations and the determination using other bundle properties [18]. In order to remove potential biases in the reconstruction of the muon number of the different detectors, the more abstract z scale was introduced. Given the average measured muon multiplicity $\langle N_\mu \rangle$, and the average multiplicities $\langle N_\mu \rangle_p$ and $\langle N_\mu \rangle_{fe}$ of proton and iron primaries predicted by the simulation of air showers, the z scale is given by:

$$z = \frac{\ln\langle N_\mu \rangle - \ln\langle N_\mu \rangle_p}{\ln\langle N_\mu \rangle_{fe} - \ln\langle N_\mu \rangle_p}.$$

z is expected to be between zero and one, as a measured value of $z = 0$ corresponds to the average multiplicity found in a pure proton simulation, while $z = 1$ would be obtained for a pure iron simulation. Since both $\langle N_\mu \rangle_p$ and $\langle N_\mu \rangle_{fe}$ depend on the simulation and therefore on the used hadronic interaction model, the z scale is calculated separately for each model.

The plots in Figure 2.3 show the z scale for different interaction models as measured by various experiments. Additionally, the values for z from measurements of X_{max} and the expectation according to air shower simulations are plotted. A strong deficit in the muon number is visible for energies above 10^7 GeV, following a linear slope that was identified with a significance of 8σ in the combined analysis [15]. Tuning the interaction models could solve this discrepancy, but would in turn be inconsistent with the expectation from simulations and results from experiments using X_{max}. Therefore, it is believed that the models do not accurately describe some physical effect which leads to the observed disagreement. A likely explanation is that less neutral pions get produced in the air shower than currently assumed, as this would have a large impact on N_μ but not on X_{max}.

The Heitler-Mathews model of air showers

This connection between the number of produced neutral pions and the muon multiplicity can be understood by studying a simplified way of describing the development of air showers, the so-called Heitler-Mathews model [20]. Despite being only a rough approximation of reality, it can be used to develop an intuition of the relationship between properties of the primary, observables of the air shower, and the influence of the features of hadronic interactions. In the model, all hadrons in the air shower are assumed to be pions, ignoring heavier particles like kaons or protons. The primary particle interacts and produces multiple charged and neutral pions. At this point, one third of the energy leaves the hadronic component of the shower, as the neutral pions quickly decay into photons. The remaining $\alpha = 2/3$ of the energy is assumed to be distributed evenly among the charged pions, which travel for some time through the atmosphere before they all interact again at the same time. This process is repeated until the energy of the pions falls below a critical value, and they decay into muons and neutrinos. Heavier primaries are treated via superposition as the sum of multiple, independent proton showers, each having the same fraction of the total energy.

In a shower developing according to the Heitler-Mathews model, the number of produced muons N_μ is proportional to both the energy E and the nucleon number A of the primary particle:

$$N_\mu(E, A) \sim A^{(1-\beta)} E^{\beta}, \qquad \beta \sim \ln \alpha, \beta \approx 0.9.$$

As a consequence, primaries of different masses but the same energy can be distinguished based on the resulting muon multiplicity. This is caused by the fact that the pions produced by a highly energetic proton with a single nucleus can undergo more iterations of interactions before reaching the critical energy than the multiple, lower energetic nuclei of a heavier primary. Thus, for lighter primaries, more energy is transferred to the electromagnetic component of the shower, resulting in fewer produced muons.

In its most basic version described above, the Heitler-Mathews model assumed that after each iteration of interactions, $\alpha = 2/3$ of the energy remains in the hadronic component of the shower in the form of positively and negatively charged pions, while the remaining third is lost to neutral pions. In reality, however, showers also produce particles heavier than pions. Since their energy remains in the hadronic cascade, this leads to an increase of α and consequently of the number of muons arriving at the ground. Unlike the depth of the shower maximum X_{max}, the muon number N_μ is very sensitive to α, so if this value is set too small, it could lead to an effect like the one described by the muon puzzle.

While it is not established yet which physical effect could lead to such a reduction of α, a possible cause could be an increased production of particles containing strange quarks in the forward region with pseudo-rapidities above two or so, as they are typically encountered in air showers [15]. The pseudo-rapidity is a scalar quantity describing the angle between the trajectory of a particle and the beam axis, and a value above two corresponds to an angle below about 15 degrees. This would lead to a reduction of the rate at which neutral pions would be produced in the air shower, and consequently increase the value of α. A hint towards this is given by a recent measurement by LHC-ALICE, in which an enhanced strangeness production was found in the pseudo-rapidity range below one (angle above 40 degrees) for high multiplicity pp collisions [21]. Other LHC experiments like LHCb or CMS could perform measurements on the cross sections of corresponding interactions also in the forward region to investigate this issue [15].

Opportunities for KM3NeT/ORCA

KM3NeT/ORCA provides numerous opportunities for performing measurements with atmospheric muons. Since the detector is located deep underwater, only highly energetic muons can reach through the ocean above it and produce a signal. As these muons are often produced early in the shower, they allow for studying the first or early interactions close to the primary. Due to its dense instrumentation, KM3NeT/ORCA is especially well suited for measuring the muon multiplicity of bundles. As described above, this quantity is strongly correlated to the mass of the primary particle.

As such, the presence of the muon deficit could be confirmed by investigating the flux of high multiplicity events. However, a problem arises in the independent determination of the primary energy in addition to the mass, as KM3NeT/ORCA does not feature a surface component like for example IceCube's IceTop (see section 3.4). Such a surface detector allows for measuring the electromagnetic component of the air shower, which does not reach the depths of the KM3NeT/ORCA detector. But even by measuring only the muonic component, it could be possible to detect the muon deficit by making use of the expected energy- and mass-distributions from other measurements and simulations. Examples for this include studies performed by SUGAR [22] and NEVOD-DECOR [23].

Investigations on the basis of air shower simulations show that the muon multiplicity is especially well suited to estimate the primary composition at high energies towards the ankle [24]. KM3NeT/ORCA could therefore help to constraint models for the possible source of extragalactic cosmic rays (see subsection 2.1.4). This is especially true if the muon puzzle can be resolved and the theoretical uncertainties of the hadronic interaction models at high energies can be consequently reduced.

Additionally, the denser instrumentation of KM3NeT/ORCA could allow for a more precise measurement of the lateral separation of muons in bundles as compared to other underground Cherenkov detectors. This quantity is connected to the transverse momentum of the hadrons that have produced the muons in the air shower, which in turn is influenced by the primary mass and effects described in hadronic interaction models [15]. A measurement of the separation could therefore serve as an additional independent observable for determining the mass of the primary, and as an additional way to cross-check the interaction models (see [25] for a corresponding study in IceCube).

2.1.4. Origin

While some cosmic rays can be produced by the sun during solar flares, most of them have an extrasolar origin. At energies in the low GeV range, solar winds have been shown to have a strong influence on the particles arriving in the solar system, decelerating or even preventing them from coming closer at all [26]. For higher energies, these effects become increasingly less influential and reveal the underlying energy distribution of the incident cosmic rays.

The first break in the otherwise smooth energy spectrum is the knee in the PeV range. The reason for this disruption is not conclusively clear, but a common theory is that the knee appears at the maximum momentum that galactic accelerators can emit particles with. Consequently, cosmic rays with higher energies would necessarily be produced in extragalactic sources with a slightly harder spectrum.

An easy but powerful way of estimating whether an astrophysical object comes into question for accelerating particles up to a given momentum is the so-called Hillas criterion [27]. If a particle with charge q and momentum p moves through a homogeneous magnetic field B, it will travel on a circular path due to the Lorentz force. The radius of this circle is called the gyroradius r_g, and it is given by:

$$r_g = \frac{p}{qB} \tag{2.1}$$

Particles can escape the site of a cosmic accelerator if their momentum (and therefore energy) is so high that the gyroradius is larger than the object itself. This means that if the size and

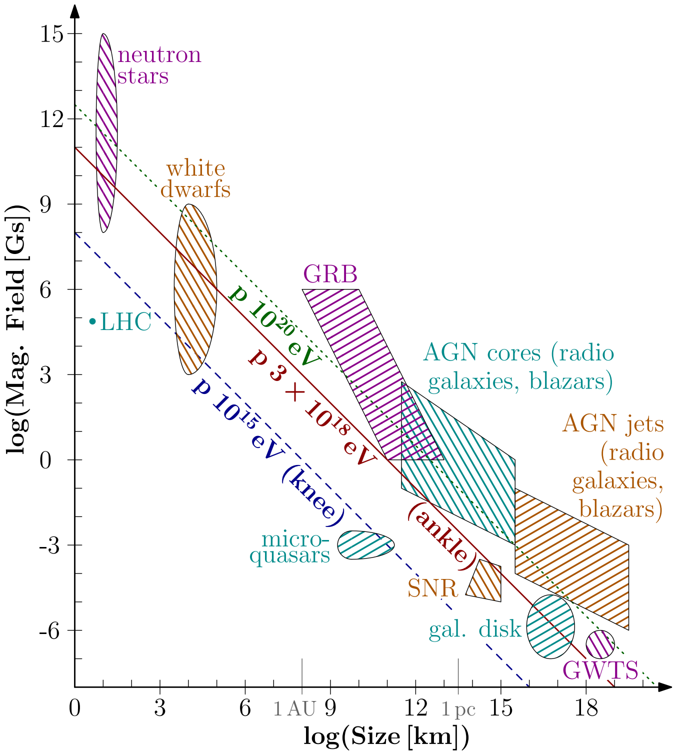


Figure 2.4.: The classical Hillas plot with potential candidates for the acceleration of cosmic rays. It shows typical values for the magnetic field strength and size of astrophysical objects as the hatched areas. In a simplified picture, they can accelerate particles up to the energy at which the gyroradius is larger than the object itself. These energies are shown for protons as diagonals for the knee, ankle and cutoff region of the observed cosmic ray energy spectrum [1].

typical magnetic field strength of an astrophysical object is known, a prediction can be made for the maximum energy it can accelerate particles to, depending on the type of nucleus.

This allows for a rough estimate of which objects are source candidates for the different energy regions in the observed cosmic ray spectrum (see Figure 2.4). However, this does not consider the boosting taking place in relativistic shocks of for example active galactic nuclei or gamma ray bursts, so a modification of this formula which allows for higher energies is necessary for these cases [28]. It is important to note that fulfilling the Hillas criterion is a necessary condition for an accelerator site, but not a sufficient one: The actual energies reached by the emitted particles in the site can be lower than the maximum given by the Hillas criterion, depending on the acceleration process and losses involved in it.

According to the Hillas criterion, several objects come into question for the acceleration of galactic cosmic rays. Some of the most discussed ones are:

Supernova remnants An especially promising candidate are supernova remnants. After a supernova explosion, a part of the stellar mass is propelled outward in the form of a supersonic shock wave that can spread deep into the surrounding interstellar medium. The so-called first order Fermi acceleration taking place in the shock wave provides an explanation not only for the process by which cosmic ray particles are accelerated up to the observed energies, but also for the shape of a power law (see subsection 2.1.5). Additionally, the total luminosity of cosmic rays arriving at Earth can be reasonably explained by supernovae as well, by assuming typical values for the energy released by a supernova explosion, and the rate at which they occur in the Milky Way [1]. If highly energetic protons collide with surrounding matter, they can produce pions which quickly decay into a pair of photons.

The Fermi experiment was able to detect the characteristic feature of this decay in the gamma ray spectrum of several supernova remnants [29], providing experimental evidence that they might be a source of galactic cosmic rays.

Microquasars A microquasar typically refers to a black hole that forms a binary system together with a companion star, which constantly loses mass due to the gravitational pull of the black hole. This material can form an accretion disk and even produce jets of extremely energetic particles that shoot out into space at near light speed.

Pulsars Pulsars are neutron stars or white dwarfs, which may originate as the product of a supernova. They are extremely dense, rotate rapidly, and have a strong magnetic field. This field accelerates particles along the magnetic axis in focused, highly energetic streams.

Since none of these galactic sources both meet the Hillas criterion and have sufficient luminosity, other astrophysical accelerators outside of the Milky Way are discussed as potential sources for the flux at the higher energies beyond the knee:

Active galactic nuclei An AGN is the central region of some galaxies that emits large amounts of radiation distinctively different from the usual stellar spectra. They are typically assumed to contain a supermassive black hole which pulls in surrounding matter, producing a rotating accretion disk and potentially a pair of highly energetic jets, similar to the microquasar.

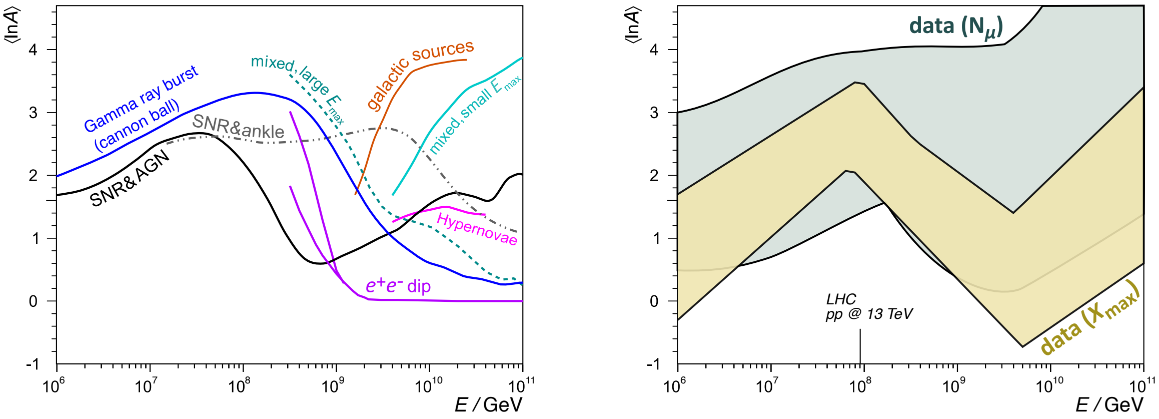
Starburst galaxies Some galaxies exhibit an unusually high rate of forming new stars compared to the observed average. They are referred to as starburst galaxies, and have been suggested as a potential source of ultra high energy cosmic rays by the Pierre Auger observatory [30]. In a recent study, they found an anisotropy in the arrival direction of cosmic rays at energies above $4 \cdot 10^9$ GeV, which could be explained with a significance of 4σ over isotropy by assuming starburst galaxies as the sources [31]. However, other candidates like AGNs or combinations of multiple sources can be used to explain the fluctuations as well, albeit with a lower significance.

Gamma ray bursts Astronomers have observed rare events in other galaxies referred to as gamma ray bursts, which release enormous amounts of energy in very short time periods. Despite their name, they release energy not only in the form of gamma rays, but in other wavelengths as well. The origins of these explosions and the mechanism of acceleration are still under debate, but a potential candidate are super-luminous supernovae.

Apart from studying the anisotropy of cosmic rays, clues on their origin can also be obtained from their elemental composition. However, this measurement has to be done indirectly at higher energies. This in consequence leads to statistical fluctuations of observables like the muon number from shower to shower It can therefore be virtually impossible to distinguish between heavier elements with similar masses on an event-by-event basis. As a consequence, many experiments simulate only a few primaries or primary groups for their composition measurement. To make results comparable, the elemental composition of highly energetic cosmic rays can be summarized by a simple scalar, the mean logarithmic mass $\langle \ln A \rangle$ of the observed primaries.

Depending on the source assumed to produce the cosmic rays, there are different expectations for the distribution of $\langle \ln A \rangle$. In Figure 2.5a, the predictions from multiple astrophysical models are shown, while Figure 2.5b depicts a summary of measurements that have determined $\langle \ln A \rangle$ using either the muon multiplicity N_μ or the depth of the shower maximum X_{max}.

An overview over the models shown in the plot is given in [32]. The two-component SNR&AGN model shows decent agreement with the measurements over the entire plotted energy range. Here, supernova remnants are the dominant source at energies at and above the knee up to 10^8 GeV, while active galactic nuclei take over for higher energies. The dip model is similar to this, except



(a) Predictions from various astrophysical models describing the source of cosmic rays. See the text for a short explanation of the different models.

(b) Envelope around measurements using the muon multiplicity N_μ and the depth of the shower maximum X_{max}. The envelope was calculated using multiple experiments and pre-LHC nuclear interaction models.

Figure 2.5.: Mean logarithmic mass $\langle \ln A \rangle$ of cosmic rays plotted over their energy [15].

that it additionally explains the observed ankle of the energy spectrum by interactions of the cosmic ray particles with the cosmic microwave background producing electron and positron pairs. This process requires a high fraction of protons in the composition to work. The SNR&ankle model instead assumes that the ankle is caused by the transition from galactic to extragalactic sources, and that it happens at higher energies between 10^9 and 10^{10} GeV. The two mixed models assume that the extragalactic sources dominant above the ankle are similar to the galactic ones. Other models assume that the cosmic rays underwent acceleration by gamma ray bursts or hypernovae. Furthermore, there are models which propose that even very high energetic cosmic rays have some rare but exceptionally powerful galactic sources.

In Figure 2.5b, the envelope around measurements from various experiments is plotted. The uncertainty of the measurements based on the muon multiplicity N_μ is substantially larger than the one from the measurements using X_{max}. Up to 80% of this uncertainty is caused by theoretical uncertainties in the high energy nuclear interaction models, which are crucial for the indirect measurement of the primary cosmic rays [15]. While the latest interaction models take into account recent measurements from LHC and can reduce these uncertainties, they also are inconsistent with measurements of N_μ (muon puzzle, see subsection 2.1.3). If the muon puzzle could be solved, the size of the envelopes could be drastically reduced and allow for the exclusion of some of the astrophysical models in the future. With that, it would also improve the potential of KM3NeT as a tool for measuring the cosmic ray composition, as it is sensitive only to the muonic component and not the electromagnetic part of the air shower which is typically used for determining X_{max}.

2.1.5. Acceleration

As was shown in Figure 2.1, the flux of cosmic rays roughly follows piece-wise power laws of the energy. This is an important hint for discovering the source of cosmic rays, since its acceleration mechanism must provide an explanation for why a distribution like this is observed.

Interestingly, charged particles undergoing the so-called first order Fermi acceleration in strong shock waves, like they are present in supernova explosions, can exhibit such an energy spectrum. This is one of the reasons why supernova remnants are often discussed as a promising potential accelerator of cosmic rays in the galaxy. Additionally, such shock waves can also appear in extragalactic sources like active galactic nuclei or gamma ray bursts.

As a shock wave made of magnetized plasma expands into the surrounding matter, particles move through the front into the region containing shocked gas directly behind it. Here, they scatter on the various irregularities of the present magnetic fields. After some scatterings, they can leave the shock wave, either in the direction of its expansion in front of the shock front, or in the opposite direction.

In the latter case, they exit the acceleration process and consequently keep their current energy. When they get in front of the shock wave, however, they not only increase their energy, but can also enter the shocked gas again, and thus have another chance of getting in front of the wave. This leads to a steeply falling energy distribution, as higher energies occur increasingly rarer due to requiring multiple forward scatterings in a row. Based on these assumptions, the expected resulting energy distribution can be estimated.

Let the particle gain a fraction ξ of its current energy whenever it crosses the front of the shock wave, After crossing the shock wave n times, a particle with the initial energy E_0 will then have the following total energy $E(n)$:

$$E(n) = E_0 \cdot (1+\xi)^n. \tag{2.2}$$

Solving this equation for n gives the number of crossings required to reach a given energy E:

$$n(E) = \ln\left(\frac{E}{E_0}\right) / \ln(1+\xi). \tag{2.3}$$

As described above, the particle has a certain probability p of being accelerated again after it has encountered the front. The chance of being accelerated at least n times is therefore given by p^n. As described above, each acceleration increases the energy, so the total number of particles N with at least a given energy E should be proportional to this aforementioned probability:

$$N(>E) \sim p^n. \tag{2.4}$$

By inserting Equation 2.3 into Equation 2.4 and making use of the identity $a^{\ln b} = b^{\ln a}$, a power law for the energy distribution can be obtained:

$$\begin{aligned} N(>E) &\sim p^{\ln\left(\frac{E}{E_0}\right) / \ln(1+\xi)} \\ &= \left(\frac{E}{E_0}\right)^{\ln p / \ln(1+\xi)} \\ &= \left(\frac{E}{E_0}\right)^{-\gamma}, \end{aligned}$$

with the index γ defined as

$$\gamma \equiv \ln(\frac{1}{p}) / \ln(1+\xi). \tag{2.5}$$

This calculation is a simplification, as it does neglect perturbations like the influence of the particles themselves on the magnetic field or the orientation of the fields with respect to the shock front. However, by making use of the kinetic gas theory and some assumptions about the velocities involved in the shock wave, it can be shown that the expected value for the index γ is close to the experimental values for the spectral index of cosmic rays [5].

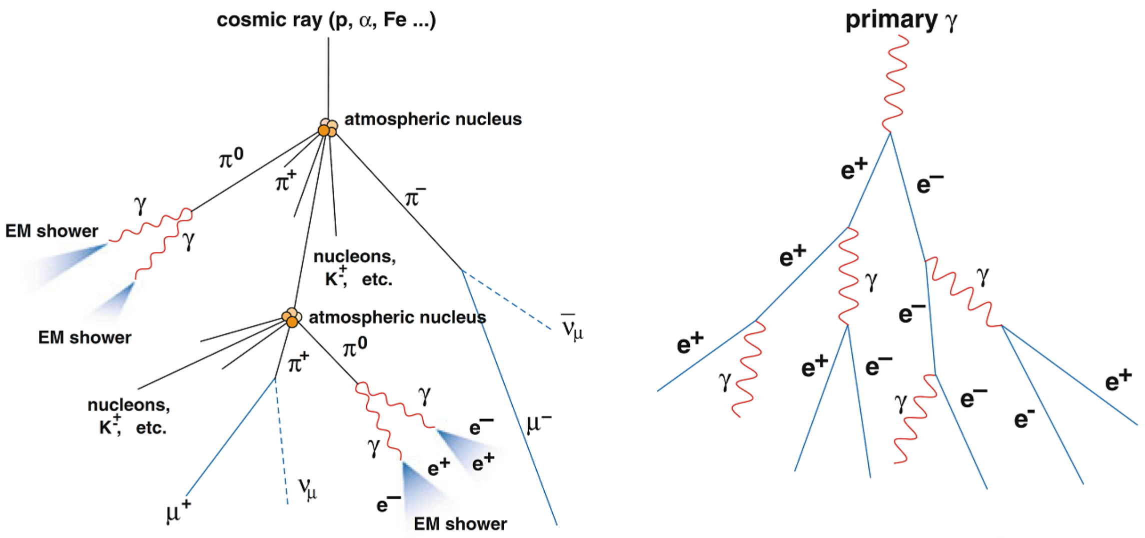


Figure 2.6.: Scheme of an extensive air shower induced by a cosmic ray particle (left). A reoccurring phenomenon in the air shower are electromagnetic showers, which are depicted in detail on the right [33].

2.2. Atmospheric showers

As a cosmic ray particle enters the atmosphere of the Earth, it can interact with the nuclei of an air molecule and produce several secondary particles. They can collide with the surrounding nuclei again, generating new particles, and so on. This cascade of collisions is known as an extensive air shower, and is the production mechanism for atmospheric muons. As depicted in Figure 2.6, the many different interactions that take place in such a shower can be grouped into two components: electromagnetic and hadronic showers.

2.2.1. Hadronic shower

Both the primary particle as well as the hadrons produced deeper into the shower can undergo nuclear interactions and create a hadronic shower. Some of the energy of the incident hadron is consumed in the nuclear process. The rest is mostly contained in neutral and charged pions, which make up the majority of the resulting hadrons, and to a lesser extent in heavier hadrons like kaons. One third of the pions are neutrally charged, and decay extremely quickly into a pair of photons:

$$\pi^0 \to \gamma + \gamma, \qquad \text{branching ratio} = 0.988, \text{ mean lifetime} = 8.5 \cdot 10^{-8}\text{ ns.}$$

These photons can then cause an electromagnetic shower. Two thirds of the pions are charged, and decay much more slowly into muons and neutrinos (see also the Feynman diagrams in Figure 2.7):

$$\begin{aligned} &\pi^+ \to \mu^+ + \nu_\mu \\ &\pi^- \to \mu^- + \bar{\nu}_\mu, \qquad \text{branching ratio} = 0.999, \text{ mean lifetime } 26\text{ ns.} \end{aligned}$$

This decay is the dominant source of atmospheric muons. About three quarters of the energy of the charged pions is contained in the neutrino produced in this interaction. Since the neutrinos have such a low chance of interacting again, this part of the energy is essentially removed from the development of the hadronic shower. The muons themselves are typically minimum ionizing (see section 2.3) and can therefore travel for quite a long time. Many of them reach the surface,

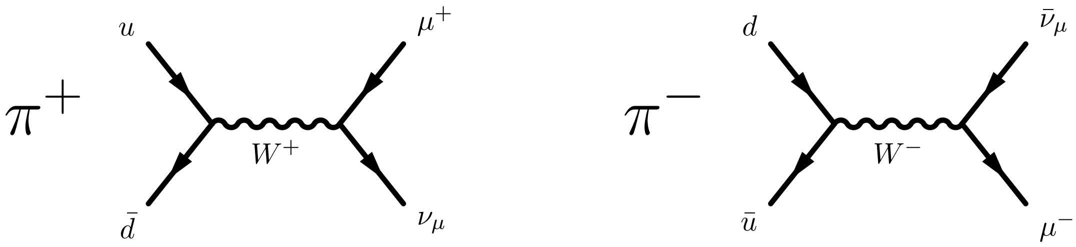


Figure 2.7.: Feynman diagrams of the decay of a positively (left) and negatively (right) charged pion. This is the dominant production mechanism of atmospheric muons.

and some can even go deep underground. Eventually, they decay into an electron or positron, as well as a pair of neutrinos:

$$\begin{aligned} &\mu^+ \to e^+ + \nu_e + \bar{\nu}_\mu \\ &\mu^- \to e^- + \bar{\nu}_e + \nu_\mu, \qquad \text{branching ratio} = 1, \text{ mean lifetime 2197 ns.} \end{aligned}$$

A typical atmospheric muon is produced at an altitude of about 15 kilometers in the atmosphere through the decay of a pion. On its way to the surface, it loses an average of about 2 GeV of its energy due to the ionization losses described in subsection 2.3.1. At the sea level, the average energy of a muon is about 4 GeV, meaning that only extraordinarily energetic muons will be able to reach deep enough into the ocean to arrive at detectors like KM3NeT. Since cosmic rays consist mostly of positively charged nuclei containing protons, there is an excess of positively charged pions in the hadronic showers as well. As a consequence, there are about 30% more positively charged muons at momenta above 1 GeV/c than there are negative ones.

2.2.2. Electromagnetic shower

An electromagnetic shower consists of photons, electrons and positrons, which undergo pair production and lose energy through radiative processes. Many of these showers are initiated by the photons emitted from the decay of neutral pions in the hadronic component of the shower. If an incident photon has an energy higher than twice the rest mass of an electron $m_e = 511$ keV, it can produce an electron-positron pair in the vicinity of an atomic nucleus:

$$\gamma \to \mathrm{e}^- + \mathrm{e}^+.$$

These leptons will lose their energy quickly due to radiation processes like Bremsstrahlung (see section 2.3), which is emitted in the form of additional photons. If the energy of these photons is high enough, they can undergo pair production yet again and thus continue the development of the shower. Eventually, the energy is distributed among too many of the involved particles and the evolution of the shower comes to a halt, since most of the energy is then lost by ionization. Due to the relatively high energy losses of electrons and positrons, most of the particles at the surface level and below are the muons and neutrinos that are produced in the hadronic showers.

2.3. Interaction of muons with matter

As charged particles like muons travel through dense material, they undergo different kinds of interactions with the surrounding matter, resulting in a significant loss of energy. This effect strongly depends on the momentum of the particle.

For relativistic muons with momenta between 0.1 and 100 GeV/c, the energy loss is relatively small, reaching a minimum at about 0.3 GeV/c. Since the energy loss in this region is dominated

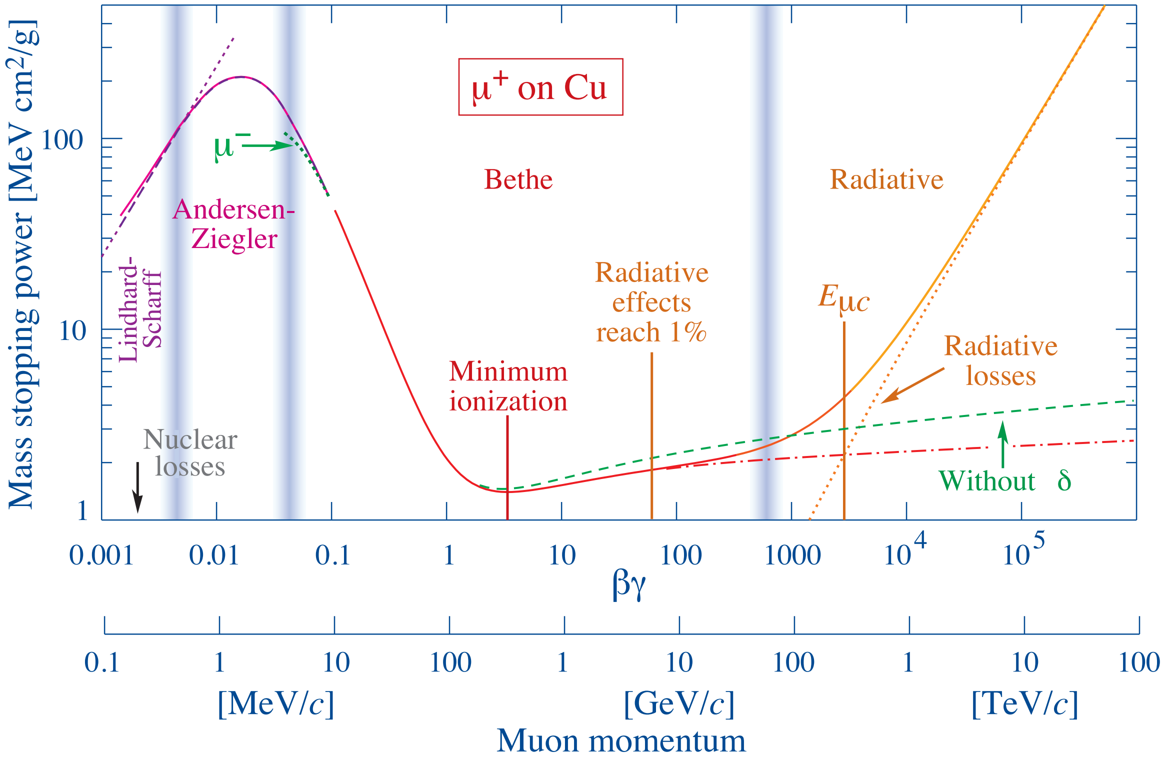


Figure 2.8.: The energy loss of muons in matter for different energy regions. Shown is the mass stopping power as a function of $\beta\gamma = p/Mc$ for the example of a positively charged muon in copper [34]. The mass stopping power is defined as $-\frac{1}{\rho}\langle\frac{dE}{dx}\rangle$, i.e. the average energy loss per length, divided by the density ρ of the material. By inserting the rest mass of muons ($M \approx 106$ MeV/c^2) into the equation for $\beta\gamma$, the x-axis can be given in terms of the muon momentum p, which is shown as the second scale at the bottom.

by ionization processes, particles at this energy are commonly referred to as minimum ionizing particles. In practice, most muons generated by cosmic rays can reasonably be assumed to be minimum ionizing [34], which allows them to travel long distances through air and consequently be detected at the surface or below.

For momenta above 100 GeV/c, the energy loss rises sharply as radiative processes begin to dominate. Below 0.1 GeV/c, the stopping power increases steeply as well, due to ionization. This is shown in Figure 2.8 for a positively charged muon in copper. The energy loss around the minimum has a small correlation to the proton number Z of the traversed material, so the graph would look slightly different for water or air. In general, the stopping power for minimum ionizing particles is typically about 2 MeV cm^2/g, which corresponds to an energy loss of about 200 MeV/m in water.

2.3.1. Ionization and excitation

Charged particles can interact with atoms along their trajectory by expelling some of their electrons (ionization) or by exciting them to a higher energy level. In both cases, the incident particle looses energy, which can be described by a continuous loss per unit length depending on the properties of the material, like its density or atomic number. In the interval of roughly $0.1 < \beta\gamma < 1000$ (muon momenta between 0.01 and 100 GeV/c), the average energy loss for heavy particles like muons can be described with an accuracy of a few percent with the so-called Bethe-Bloch formula.

2.3.2. Radiation

If a charged particle is accelerated, it looses energy by emitting electromagnetic radiation. As a particle travels through matter, it occasionally gets deflected by the electric fields of atoms and therefore radiates photons in the process. This is called Bremsstrahlung, and together with pair and photon production, it makes up most of the radiative losses typically encountered by muons. While being insignificant compared to ionization losses for smaller $\beta\gamma < 1000$ (muon momenta below 100 GeV/c), they quickly ramp up for higher energies and become the dominant source of energy loss. For electrons, this increase is present at much lower momenta due to their low mass.

Another source of radiative energy losses is Cherenkov radiation, which occurs when the charged particle moves faster than the local speed of light in the material. Even though the amount of radiated energy in Cherenkov light is comparatively small - being several orders of magnitude below ionization losses - it is of critical importance for particle measurements in Cherenkov detectors like KM3NeT (see also section 3.2).

2.3.3. Deflection

Charged particles can be scattered elastically by the Coulomb fields of surrounding nuclei. While the energy loss is negligible, many of these small collisions can noticeably change the trajectory of the particle. This effect can be described by the theory of Moliere, according to which the distribution of the deflection of most particles is similar to a Gaussian, while a small percentage forms non-Gaussian tails. As a consequence, the incident direction of an atmospheric muon can get a slight smearing on its way from the atmosphere to the surface.

Chapter 3.

Detection of particles in KM3NeT

3.1. The KM3NeT detectors: ORCA and ARCA

KM3NeT is a network of water Cherenkov detectors currently under construction in the deep Mediterranean Sea. It consists of two different sites with the same collaboration and technology behind it:

- KM3NeT/ORCA – Oscillation Research with Cosmics in the Abyss, located at a sea depth of 2450 meters near Toulon in southern France. It features a very densely instrumented 0.007 km^3 of water for precise measurements of atmospheric neutrino oscillations, with typical neutrino energies between 3 and 100 GeV.
- KM3NeT/ARCA – Astroparticle Research with Cosmics in the Abyss, located at a sea depth of 3500 meters near Capo Passero in Sicily, Italy. Its lower density of instrumentation but large volume of about one km^3 was optimized for the measurement of neutrinos from astrophysical sources, with high energies in the TeV range.

The location of the sites are shown on a map in Figure 3.1. Water Cherenkov detectors can detect charged particles via the Cherenkov radiation they induce when traveling faster than the local speed of light. In KM3NeT, these Cherenkov photons are measured by using three-inch photomultiplier tubes (PMTs, see Figure 3.2).

If a photon hits the cathode of a photomultiplier, it can be absorbed according to the photoelectric effect and cause the emission of an electron. The chance of an absorption happening - also called the quantum efficiency - amounts to about 22 to 27 percent for the Hamamatsu PMTs utilized in KM3NeT, depending on the wavelength [35].

In the photomultiplier, a series of dynodes are arranged in an electrical field, with the potential increasing from one dynode to the next. Because of this, the emitted electron is accelerated towards the first dynode, releasing additional electrons upon impact. These electrons are in turn accelerated towards the second dynode, and each of them can release even more electrons.

This process is repeated through a total of ten dynode stages, amplifying the original signal by a factor of more than 10^6 [35]. Finally, they hit the anode and generate an electrical pulse. If the voltage in this pulse exceeds a defined threshold, the time of the leading edge as well as the timespan in which the voltage was over the threshold (referred to as the time over threshold, or ToT) is digitized with nanosecond precision. Together with the info of which PMT recorded the pulse, this is called a hit.

31 PMTs are mounted in a water-proof and pressure resistant glass sphere with a diameter of 43 cm, which is well suited for a long-term use in the deep sea. Out of the 31 PMTs, 19 are located in the lower half of the DOM, and the remaining 12 are in the upper half. Having so many photomultipliers in each DOM has numerous advantages: Apart form the increased photo-cathode area and the additional reliability due to redundancy, it also allows for measuring the photon arrival direction. As described in later chapters, this additional information can be efficiently used by graph neural networks to aid its reconstructions.

A detection unit (DU, also called string or line) is made up of two long ropes, to which 18 of these DOMs are attached to. It is anchored to the sea floor and has a buoy at the top, which

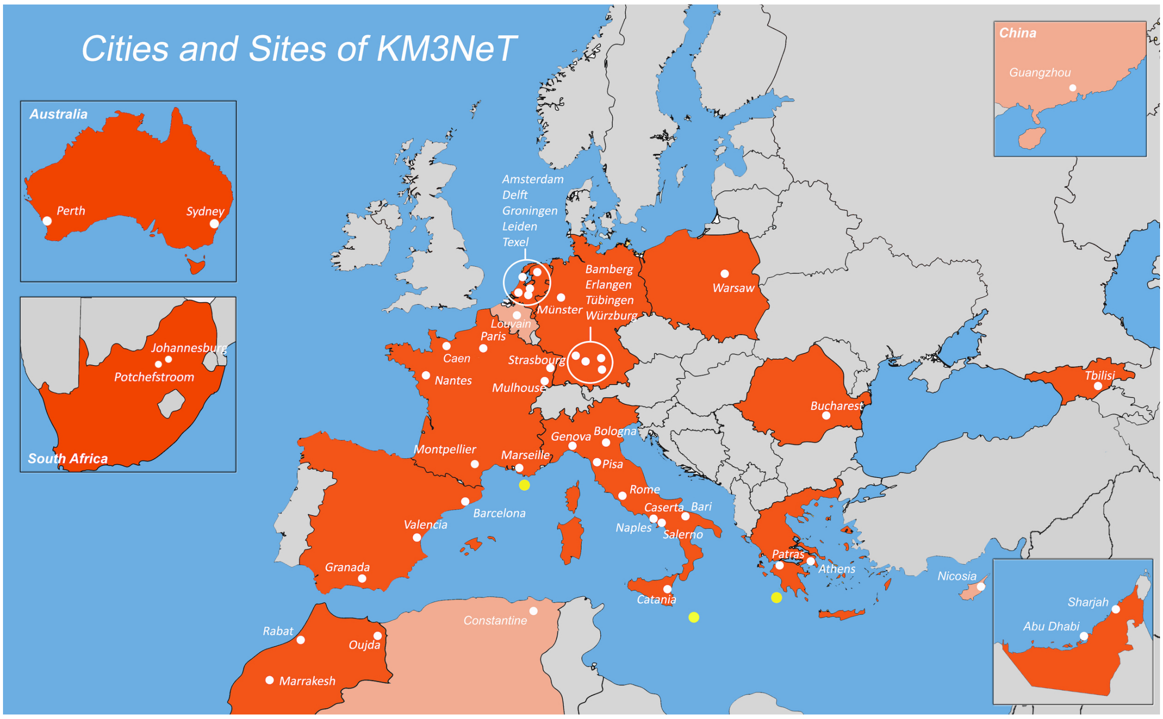


Figure 3.1.: Map of relevant locations of the KM3NeT experiment. The cities of collaborating universities and research institutes are shown in white. The locations of ORCA, ARCA, and the proposed third site of KM3NeT near Greece are shown in yellow (courtesy KM3NeT).

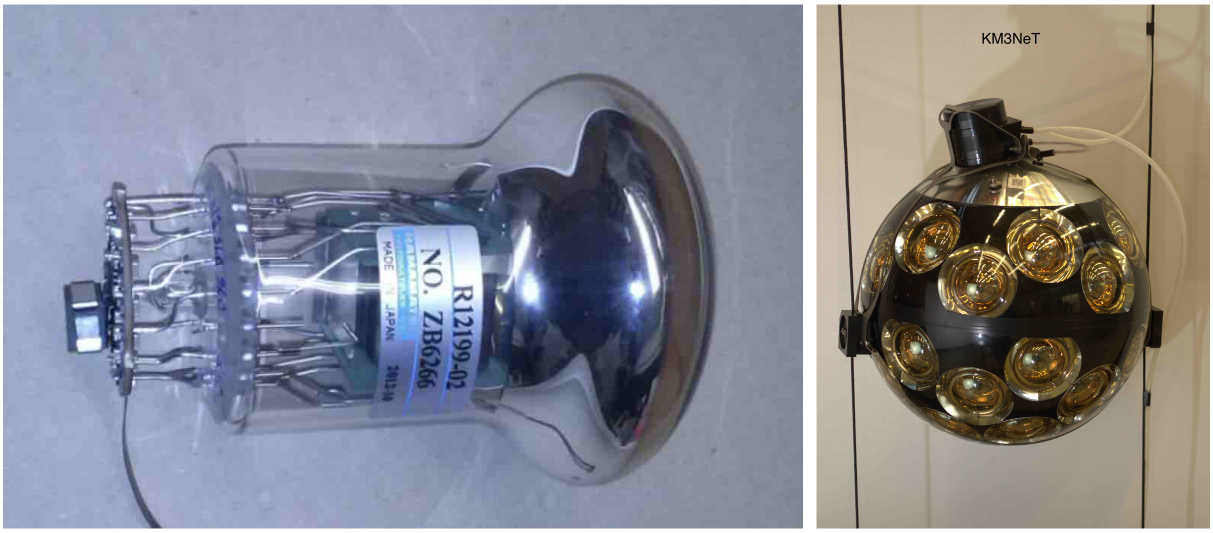


Figure 3.2.: Photos of a a prototype of a Hamamatsu photomultiplier (PMT) on the left, and a KM3NeT digital optical module (DOM) on the right [36]. 31 PMTs are installed in each DOM, visible as the golden disks.

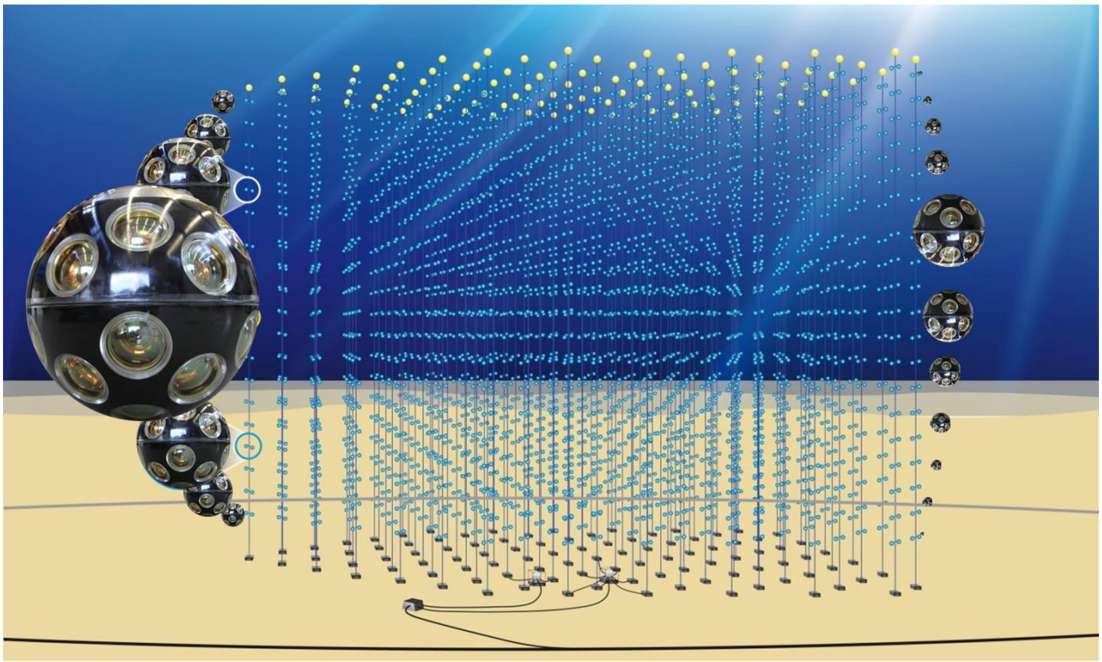


Figure 3.3.: Artists impression of a fully constructed KM3NeT building block with 115 installed detection strings, each of which contains 18 DOMs (courtesy KM3NeT).

increases the buoyancy of the string to keep it in an upright position. The vertical spacing between the DOMs depends on the site. For KM3NeT/ORCA, it is about 9.3 meters, whereas for the sparser KM3NeT/ARCA detector, the distance between the stories is about 36 meters.

115 detection units are distributed on a circular area to instrument a large three dimensional volume of sea water, called a building block. The horizontal spacing between the DUs is planned to be about 20 meters for ORCA, and roughly 90 meters for ARCA. An artists impression of how a completed building block could look like is shown in Figure 3.3. In total, there will be one building block in ORCA, and two blocks in ARCA.

In early 2020, the first phase of the KM3NeT/ORCA detector construction has been successfully completed, with six strings being in place and smoothly taking data. As of July 2022, four additional lines have been added, bringing the total up to ten lines. ARCA currently operates with 19 strings.

3.2. Cherenkov radiation

Water Cherenkov detectors like KM3NeT measure particles via the radiation they induce as they travel through the water in the instrumented volume, the so-called Cherenkov radiation. In order for this effect, to occur, the traversed medium has to be dielectric, the particle has to be electrically charged, and it has to be faster then the phase velocity of electromagnetic fields in the medium [37].

While it is impossible for particles to be faster than light in vacuum, they can be faster than the local speed of light, as electromagnetic waves can be slowed down in matter. For example, the local light speed in water is only about 75 percent of its vacuum counterpart. The Cherenkov effect can be understood easily by comparing it to the bow wave of a ship. The ship produces spherical wave fronts as it moves through the water. If the ship is faster then the wave fronts, the wave maxima overlap and produce the characteristic triangular shape.

in a qualitative fashion, the Cherenkov effect can be described similarly. If a particle is faster than the ratio of the local speed of light c_{loc} divided by the refractive index n of the medium, it induces an electromagnetic shock wave under a specific angle θ to the trajectory. This angle is

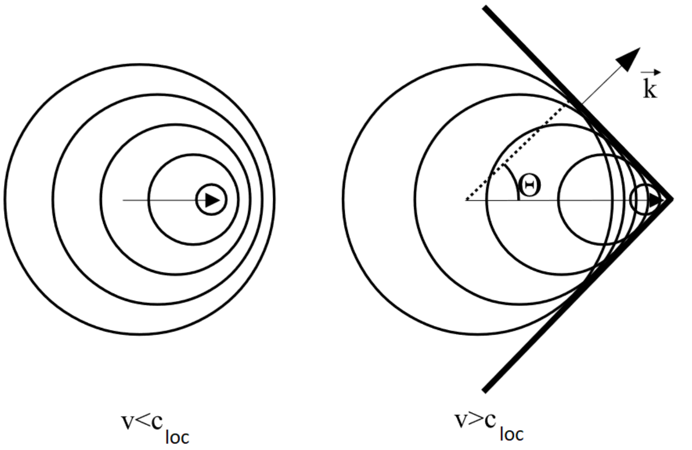


Figure 3.4.: Scheme of the propagation of Cherenkov radiation. Spherical waves are induced by the electric fields of a charged particle moving through a dielectric medium, with the particle's velocity v being $< c_{loc} = c/n$ on the left plot, and $> c/n$ on the right. In the latter case, the maxima overlap and form a cone with the opening angle θ [38].

given by:

$$\cos\theta = \frac{1}{\beta \cdot n}, \quad \text{with } \beta = \frac{v}{c_{vac}}, \tag{3.1}$$

for particles with velocities close to the speed of light ($\beta \approx 1$). Together with the typical refractive index of water $n \approx 1.35$, the Cherenkov angle amounts to $\theta \approx 42\,\text{deg}$.

3.3. Event signatures

The three main sources of events in KM3NeT are triggered by noise, by muons from the atmosphere, and by neutrinos, which can be of atmospheric or cosmic origin.

Noise is present as a background to the light emitted by passing particles. Occasionally, an event does not show a particle at all, but is made up completely of this ambient light. It is emitted mostly by radioactive decays of the $^{4}0$K that is naturally present in the water, as well as bioluminescent microorganisms living in the deep sea [39].

Muons have a low energy loss compared to other leptons, reaching as low as 250 MeV of energy lost per meter of water covered in the extreme (but common) case of minimal ionization. Therefore, they can travel long distances and consequently produce straight tracks of Cherenkov light in the detector, which allow for a precise reconstruction of their direction. Since atmospheric muons are generated in the atmosphere, their tracks never start within the detector and, in most cases, do not stop inside of the detector either, making an energy reconstruction more difficult.

Neutrinos are not electrically charged, so they can only be indirectly measured in water Cherenkov detectors via the charged particles produced in interactions with nuclei of the surrounding water. This interaction produces a hadronic cascade, visible as a shower of light. Additional light can be produced depending on the type of neutrino interaction that took place:

- Neutral current (NC) interaction under exchange of a neutral Z boson, emitting a neutrino of the corresponding flavor. Since the neutrino does not emit Cherenkov radiation, it

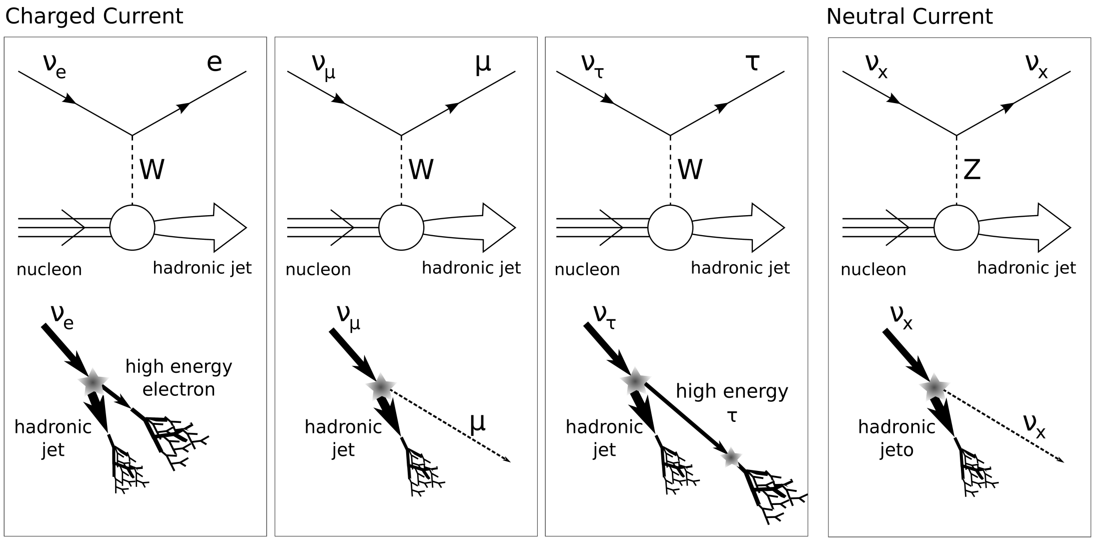


Figure 3.5.: Scheme of the different interactions between a neutrino and a nucleon [40].

escapes undetected, leaving only the electromagnetic shower from the hadronic cascade as a signal.

- Charged current (CC) interaction under exchange of a charged W boson, emitting a charged lepton of the corresponding neutrino flavor. Additional light is produced depending on the flavor:
  - ν_e: the electron loses its energy quickly, producing another shower that overlaps with the hadronic cascade.
  - ν_μ: the muon can travel long distances, leaving behind a track similar to the one from atmospheric muons. These tracks can start in the detector, in which case the hadronic cascade can also be visible. If the hadronic interaction took place outside of the detector, only the muon track is present.
  - ν_τ: the τ-lepton decays shortly after the interaction, resulting in either a hadronic or electromagnetic shower (83%) or a muon track (17%).

In summary, track-like event signatures are produced by ν_μ-CC interactions and 17% of the ν_τ-CC interactions. In all other cases, a shower like signature is generated (see Figure 3.5). Since the showers are often fully contained in the detector, they are well suited for a reconstruction of the energy. Since the flux of neutrinos is many orders of magnitude below that of atmospheric muons for events traveling downwards through the detector, they can often be safely ignored for studies regarding muons.

3.4. Other large volume neutrino detectors

Apart from KM3NeT, there are several other large volume neutrino detectors in operation or under construction, which also measure Cherenkov radiation in water using photomultipliers.

IceCube is an ice-based Cherenkov detector located in Antarctica near the south pole. The construction of its main array was completed in December of 2010, with its digital optical modules being embedded in a cubic kilometer of glacier ice at depths between 1.45 km and 2.45 kilometers [41]. It features a large array of optical modules for neutrinos with energies above 100 GeV, and a smaller array called DeepCore [42] with a five times higher instrumentation density, situated at the bottom center of its larger counterpart. This provides sensitivity to neutrinos with energies as low as 10 GeV. Additionally, the IceTop [43] extensive air shower array is located at the surface right over the detector. Completed in early 2011, its primary goal is to measure the energies of air showers, which can be used to investigate the cosmic ray mass composition.

ANTARES is the predecessor of the KM3NeT telescope, located close to the site of KM3NeT/ORCA. It was operational between May 2008 and February 2022. It used 12 detection units with up to 75 optical modules per line, each containing a single 10 inch photomultiplier, to instrument a total volume of about 0.01 km^3 of deep sea water [44].

Baikal-GVD is currently under construction in lake Baikal in the south east of Russia [45]. It consists of multiple clusters of eight detection units, every DU being composed of 36 optical modules with a single ten-inch photomultiplier each. With the construction having started in 2016, there are eight clusters operational in 2021 [46], instrumenting a total of 0.4 km^3 of water.

Chapter 4.

Processing chain for atmospheric muons

Having a set of simulations that closely resemble the measured data is of crucial importance, especially for supervised deep learning algorithms like the ones described in this thesis. In order to achieve this, a complex chain of several processing steps is necessary, each step utilizing the results of the previous one. An overview of these steps for the simulation of atmospheric muons, and for the processing of measured data, is shown in Figure 4.1.

The first step in the simulation chain is the generation of the particles that can produce light in the detector. This can happen by simulating entire cosmic ray air showers caused by primary particles entering the atmosphere (Corsika), or by making use of parametrized distributions that directly describe the expected atmospheric muon flux near the detector (Mupage).

In the next step, the propagation of the emitted Cherenkov radiation is simulated, which only needs to be done for muons that come close enough to the detector in order to be detectable. This is obviously the case for particles that travel directly through the instrumented volume, but even if they just come somewhat close, the Cherenkov radiation can still reach inside the detector. The chance of this happening depends on the distance of their trajectory to the instrumented volume and on the attenuation length of the radiation in water. For this reason, the propagation of light is simulated in a volume that is larger than the instrumented volume by a few attenuation lengths. This enlarged volume is referred to as the can (see Figure 4.2).

Afterwards, the response of the detector is simulated. At this point, simulations and data can be treated identically, by applying the trigger algorithms and finally reconstructing the desired observables.

4.1. Event generation

4.1.1. Corsika

Corsika (short for COsmic Ray SImulations for KAscade) [48] is a simulation program for extensive air showers caused by primary particles like atomic nuclei, photons or electrons. It was originally developed for the Kascade experiment, but has become the standard solution for simulating air showers in many cosmic ray, gamma ray and neutrino experiments. For this, the particles involved in the shower are tracked through the atmosphere until they either interact or decay. The effects described in section 2.3, like the energy loss along the way depending on the density profile of the atmosphere as well as the potential deflections are taken into account in the process. Additionally, the effect of the Earth's magnetic field on the trajectory is also part of the simulation.

As described in section 2.2, hadronic interactions play an important role in the development of air showers. Models for these interactions are necessary for a detailed simulation over the entire range of energies encountered in cosmic rays. For energies up to several hundreds of GeVs, the collisions can be studied at fixed target detectors, allowing for cross checks with measured data. However, an extrapolation of the observations is necessary for energies in the higher TeV region and beyond.

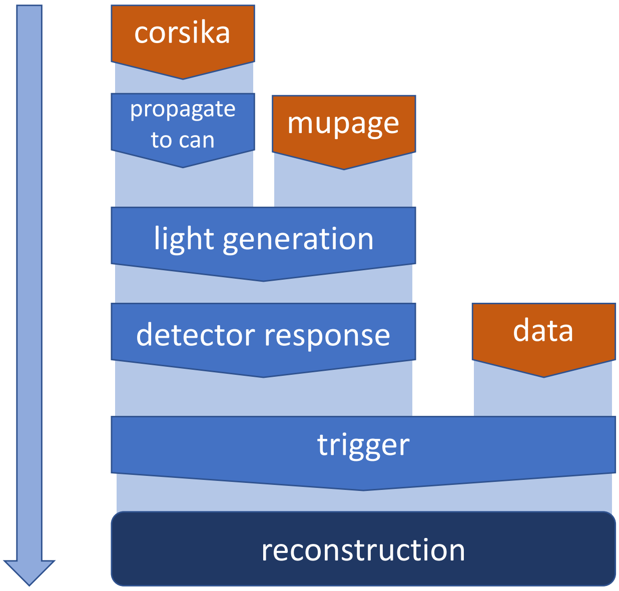


Figure 4.1.: Scheme of the processing chain for atmospheric muons and data in KM3NeT. The left route shows the chain for simulated events, which are generated either by CORSIKA and propagated to the can, or directly obtained from MUPAGE. After the light generation and simulation of the detector response, they undergo trigger algorithms similar to the ones that are used for measured data on the right route.

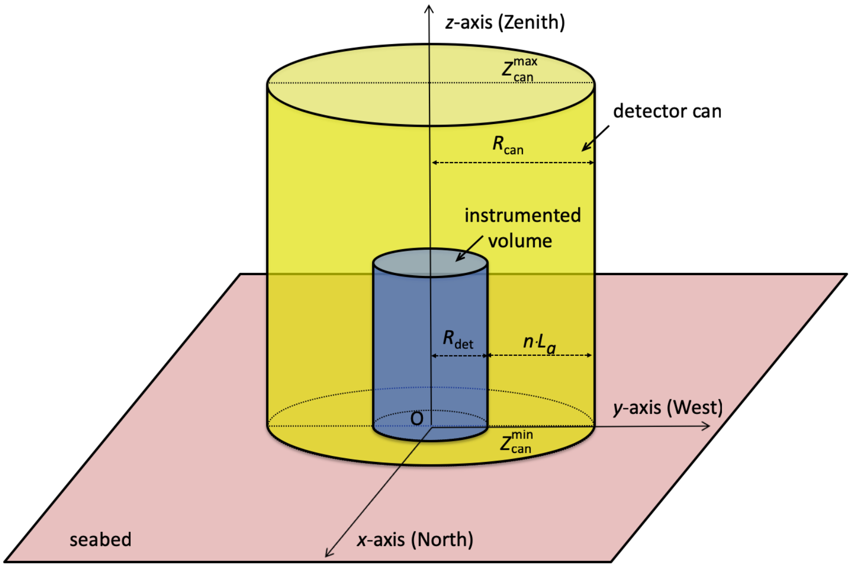


Figure 4.2.: Scheme of the can, in which the light of particles is simulated. The instrumented volume of the KM3NeT detector is shown in blue, surrounded by the can in yellow. The can is enlarged by a multiple of the attenuation length L_a [47].

For this, several so-called high energy interaction models can be used in Corsika. Some commonly used models using different theoretical bases are Sibyll[49], Epos[50] and Dpmjet[51]. For lower energies, there are several different models as well, like Urqmd[52] or Geisha[53]. Since switching between any of these different models can influence the resulting particle distributions from air showers, they can be a source of systematic uncertainty for any study regarding the indirect detection of cosmic rays. As KM3NeT is located in the deep sea and muons need to be quite energetic to reach it, the low energy model is likely to have less of an impact compared to the high energy one, though.

As described in subsection 2.1.3, the elemental composition of cosmic rays changes significantly with increasing energy. Therefore, the flux $\Phi(E, Z)$ as a function of the primary energy E and atom number Z is typically added as a weight to simulated events of the different nuclei. There are multiple models of the cosmic ray flux, so this is another source of systematic uncertainty.

A simple model is the so-called poly-gonato flux [54], which describes the energy spectrum of each primary as a powerlaw up to a Z dependent cut-off resembling the knee, and continues afterwards with a much steeper spectral index. While somewhat accurate at lower energies, this is in disagreement with measurements towards and beyond the ankle region. More complex models, like GST [55] by Gaisser, Stanev and Tilov or H3a [56] by Hillas and Gaisser feature additional free parameters to better represent the observed behavior there.

Once the particles of the air shower are simulated down to the sea level with Corsika, they are propagated all the way through the water down to the can using the corresponding programs, like Music[57] or Proposal[58].

4.1.2. Mupage

A full simulation of a cosmic ray air shower, including the interactions and propagation of the particles therein down to the detector, is a complex task that requires a significant amount of computation time to conduct. The generator program Mupage [59] - short for "muon generator from parametric formulas" - accelerates this process substantially by producing parametrized distributions of muon bundles directly at the can.

For this, atmospheric muons were simulated with the Hemas [60] code back in the 2000s, which handles the cosmic ray interaction and propagation through the atmosphere. Nowadays, the calculations in this step are typically done with the more popular Corsika framework described in the previous section. Five different primary groups (proton, helium, carbon-nitrogen-oxygen, magnesium and iron) with a flux following the poly-gonato model [61] were used for the generation of the air showers, with primary energies per particle ranging from 10^3 GeV up to $2 \cdot 10^9$ GeV [62].

The particle interactions in the atmosphere were simulated with Dpmjet [51] and propagated through the water using Music [57], down to different depths between two and five kilometers water equivalent (w.e.). This way, the parametrization can be constructed also as a function of the depth. Prompt muons were not part of the simulation.

The resulting distributions of various observables of the muon bundle flux were fitted with parametric formulas. For example, the following equation, which was originally derived for the Frejus detector [63], describes the flux Φ of muon bundles depending on the muon multiplicity m, the depth w.e. h and the zenith angle θ:

$$\Phi(m, h, \theta) = K(h, \theta) \cdot m^{-\nu(h,\theta)}, \tag{4.1}$$

$$\begin{aligned} \text{with} \quad & K(h, \theta) = K_{0a} \cdot h^{K_{0b}} \cdot \cos(\theta) \cdot \exp((K_{1a} \cdot h + K_{1b}) \cdot \sec(\theta)) \\ \text{and} \quad & \nu(h, \theta) = (\nu_{0a} \cdot h^2 + \nu_{0b} \cdot h + \nu_{0c}) \cdot \exp(\sec(\theta) \cdot \nu_{1a} \cdot e^{\nu_{1b} \cdot h}). \end{aligned}$$

The nine free parameters ($\nu_{0a}, \nu_{0b}, \nu_{0c}, \nu_{1a}, \nu_{1b}, K_{0a}, K_{1a}, K_{1b}$) were then tuned to the distributions obtained with the Hemas simulation. For this, 63 two dimensional bins were defined for h and θ, and the resulting muon multiplicity distributions were fit with Equation 4.1. Similarly, the lateral spread of muons in a bundle was modeled as a function of depth, zenith and multiplicity using nine free parameters.

Finally, the dependence of the flux on the energy was described by a multiplicative factor to the flux defined in Equation 4.1. It is a function with seven free parameters depending on the zenith and the depth for single muons, and another 15 free parameters depending on the zenith, depth, multiplicity and lateral distance for muon bundles.

The energy spectra of single and double muons from the resulting parametrization were cross checked with measurements from the Macro experiment [64]. Macro was an underground scintillation detector operating from 1989 to 2000, which was situated under the Gran Sasso mountain in central Italy. While its main goal was the search for magnetic monopoles, it also doubled as a neutrino detector and cosmic ray observatory.

For the cross check, the average muon energy at a given depth w.e. and zenith angle was estimated from the depth in rock and arrival direction of the measured muons. This estimation was necessary since the energy loss of muons in water differs from that in the inhomogeneous rock of the mountain, and its irregular shape also leads to a mixture of different zenith angles.

The estimated average energy for single and double muons were measured at four different depths with an uncertainty of 6 to 7 percent. The parametrization agrees well with this result, as the energy obtained from it was shown to be within one sigma at all four depths, both for single and double muons. However, no checks were performed for higher multiplicities, as well as the lateral and zenith distributions.

In summary, the parametric formulas contained in Mupage allow for skipping the expensive propagation of the particles in the air shower, that would otherwise require Corsika to run. All that is required are the dimensions and position of the detector can, at the surface of which the observables of the particles will be produced according to the parametrized distributions [65].

In effect, this allows for simulating much larger quantities of events with the given computational resources. This is especially useful for deep learning applications, since they typically required vast amounts of training data. At the same time, however, this also makes it impossible to study a variety of systematic effects including different interaction models, cosmic ray fluxes or atmospheric models. The approach followed in this thesis is therefore to train neural networks on Mupage simulations, and then apply them on a smaller set of Corsika simulations to study these effects.

4.2. Light propagation and detector response

The positions and properties of atmospheric muons and muon bundles at the can can be generated either from Corsika via simulation of the air shower starting at the top of the atmosphere, or directly from the Mupage parametrization. In both cases, the next step in the chain is the propagation of the muons through the can, and the simulation of the light emission.

In Km3Net, these tasks for muons are usually handled with a custom application called JSirene, which takes into account the specific configuration of the detector. It calculates the probability of a photon arriving at each of the PMTs in the detector from any of the muons traveling through the can based on multi dimensional interpolation tables [66]. Alternatively, the propagation of each photon can also be calculated individually using the program km3sim. However, it is typically only used for the simulation of neutrino events.

The next task is the simulation of the detector response. Even though Km3Net is located in the deep sea and therefore no daylight is present, atmospheric particles are not the only sources of light measured by the detector. For example, β decays of the radioactive element ^{40}K, which is naturally present in the water, pose a substantial background to the signal of particles.

Another source of light are bioluminescent micro-organisms present in the deep Mediterranean Sea [39]. They react to environmental conditions like changes in the strength of the sea current, so the amount of light produced by them can change significantly over time. To account for the background in the simulations, a rate of typically about 10 kHz per PMT for uncorrelated hits is added to the muon signal. Correlated hits on a DOM are simulated with increasingly lower rates for two up to five hits. In the so-called run-by-run simulations, the observed variation of the background rate in measured data over time is included as well.

The simulation of the detector response also has to take into account the PMT electronics in order to resemble the measured data as closely as possible. This includes the pointing direction and areas of the photocathodes present in each PMT, their angular acceptance, the influence of the glass sphere of each DOM, as well as the optical gel that is contained within [36].

4.3. Trigger

In reality, the KM3NeT detectors are constantly running and recording data. This is necessary to ensure no important information is missed, since it is not known beforehand when an event of physical interest will occur. The optical data measured in each PMT is digitized and sent to the shore station, reaching a rate as high as 25 Gigabyte per second for a full building block. In order to reduce the amount of data that is saved to disk, specific algorithms are used to search through the data for correlated hits and potential particle signatures.

These algorithms are called event triggers, and they exist at various levels. A level one trigger (L1) looks for multiple hits on different PMTs of a single DOM within a given time window of typically 10 nanoseconds. However, ^{40}K decays and bioluminescent organisms also often produce L1 hits, so more advanced level two triggers (L2) can be used to get a filtered selection with higher purity.

Since atmospheric muons and some of the neutrinos can produce track-like signatures in the detector, the 3DMuon trigger can be used to look for such patterns in the data. For this, the algorithm looks for multiple L1 hits that could be causally connected to a straight track, as they lie in a 3D cylinder around an assumed muon trajectory [67]. In similar fashion, the 3DShower algorithm looks for multiple L1 hits in a 3D sphere, that could have been produced by a shower-like CC-ν_e, CC-ν_τ or NC neutrino interaction.

A recent addition to the triggers above is the MXShower trigger, which can increase the trigger efficiency for neutrino interactions at lower energies [68]. It looks for any causally connected hits around a single 3DShower L2 hit, taking into account the fact that low energy neutrinos can occasionally not produce enough light for even a single second L2 hit.

When any of the L2 triggers fire, the entire cluster of triggered hits is saved as part of an event. If multiple triggers overlap, they are merged into a single, bigger event. Since not every photon emitted from a particle necessarily produces an L2 hit, a safety time margin is added before and after the first and last L2 hit. All hits in this time window, including the hits that have not triggered the event, are saved as part of the event as well. They are referred to as the snapshot hits. The safety margin is derived from the maximum time it takes a photon to travel through the detector at light speed.

Similar to the process on data, the simulated hits obtained from the detector response described in the previous section 4.2 undergo the same trigger algorithms. This way, the simulated events can be treated exactly the same as the measured events for the reconstructions ahead.

In Figure 4.3, the distributions of several important observables describing atmospheric muon bundles are shown for a set of simulations using MUPAGE and CORSIKA. More information on the dataset, as well as the version and parameters used for CORSIKA are presented in section 7.1 and section 8.1. The zenith angle of the muons has a large influence on the detection rate (top left plot). Towards the horizon - i.e. towards a cosine zenith angle of zero - the rate is dropping fast, as the distance to the air shower in which the muons were produced increases, resulting in a higher

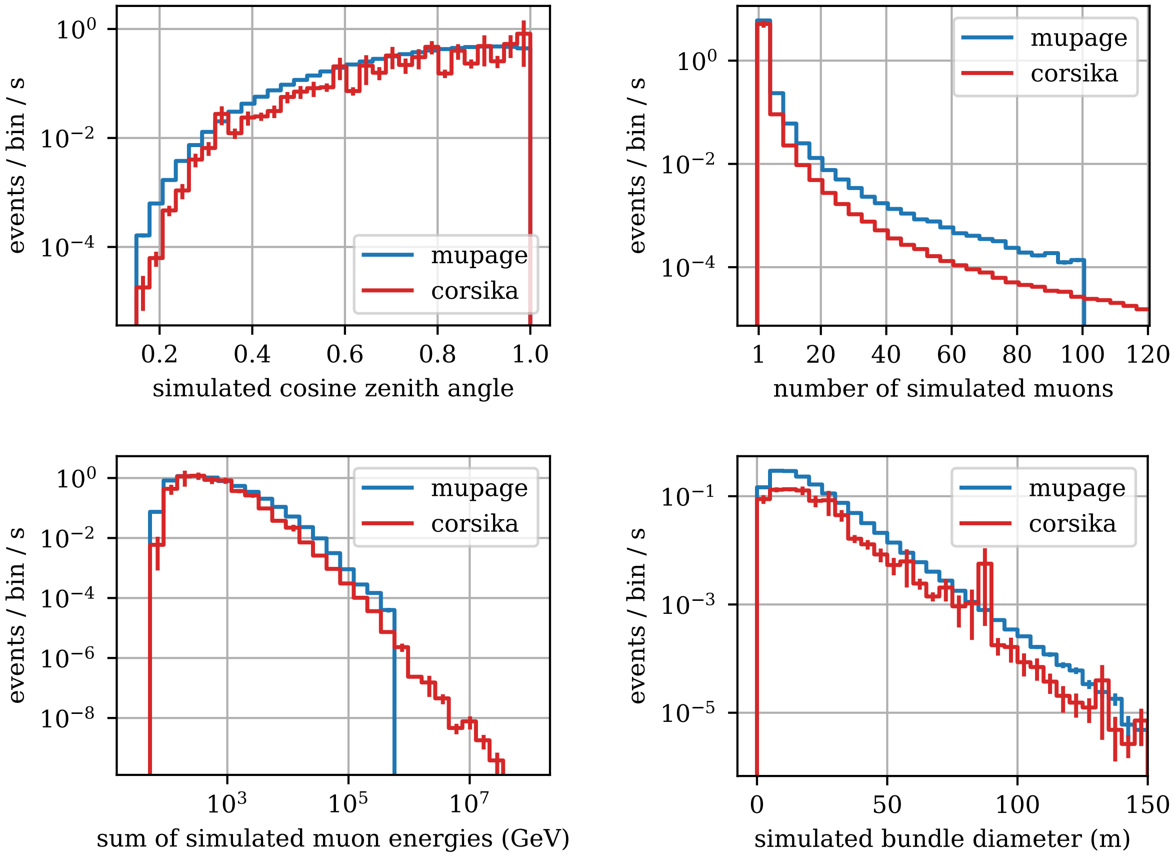


Figure 4.3.: Distribution of various observables of atmospheric muon events after the trigger stage of the simulation chain. The events are simulated using either Mupage (blue) or Corsika (red). In the plots showing the zenith and the diameter, large uncertainties can be observed in some of the bins. This is caused by a single or a few events with an exceptionally high weight being in that particular region of the phase space. The distribution of weights has been improved for upcoming productions of Corsika. More information on the datasets can be found in section 7.1 and section 8.1.

loss of energy along the way. It can also be seen that the overall rate of events is slightly lower in the Corsika dataset as compared to Mupage, since the distribution is shifted downward in the plot. The distribution of the muon multiplicity roughly follows a powerlaw, and thus rapidly decreases towards higher muon numbers (top right plot). It is much more shallow in Mupage as compared to Corsika, which is likely due to an inaccuracy of the Mupage parametrization (see subsection 7.4.3 for a detailed discussion). In the given dataset, Mupage was only used to simulate events with multiplicities up to 100 and energies up to 500 TeV, resulting in the sudden stop of the corresponding distributions in the plots. The energy of a muon bundle is defined here as the sum of the energies of all the individual muons in the event. While most bundles have an energy of about 400 GeV at the height of the Km3Net/Orca detector, some can rarely reach energies as high as 10^7 GeV (bottom left plot). The lateral distribution of muons in a bundle can be quantified by calculating the maximum perpendicular distance between any two muons in a bundle at the detector can. This is defined as the *bundle diameter* in this work, and described to greater detail in section 7.3. The median diameter of a bundle is 15 meters, with the rate decreasing exponentially towards larger diameters (bottom right plot).

4.4. Classical reconstruction

Given an event with its corresponding hits, the goal of a reconstruction is to obtain estimates for the properties of the particle, or for multiple particles in the case of atmospheric muon bundles. This can be done with the deep learning techniques described in this work, or with the existing established methods, which are designed for showers and single muon tracks. Since this classical reconstruction has been widely used and tested in KM3NeT, it offers an excellent opportunity to validate the results of deep learning, especially on measured data.

The strategy for the classical reconstruction of a muon track was originally developed for the ANTARES experiment, and has been further refined over the years [36]. The idea is to perform a maximum likelihood fit on the positions of PMTs and the time of recorded hits based on the probability distribution function (PDF) of a track hypothesis. The PDF itself has been acquired via a fit on noise-less Monte Carlo simulations.

The whole track fit is split in multiple stages: The first step is a prefit, in which multiple fits are made on a subset of causally connected hits of an event. In order to find promising hypotheses for the track direction, the sky is scanned in steps of a few degrees, and a linear fit on the hit times is performed for each step.

The actual maximum likelihood fit of the PDF is done using one or several of the best prefit tracks to estimate the direction of the incident muon. The quality of the fit can be assessed with the resulting likelihood of the PDF divided by the number of hits used in the process. Roughly speaking, the higher this ratio, the better the reconstruction tends to be. Afterwards, the energy of the muon can be estimated by using the result from the directional fit and the track length.

Chapter 5. Introduction to deep learning

A typical analysis task for KM3NeT/ORCA is to reconstruct the properties of a particle based on the data that was measured by the detector. For example, one might be interested in its energy or its direction of arrival. Naturally, the prediction should be a good approximation of the real value of the property.

Mathematically speaking, this prediction is a function $\hat{F}\left(\boldsymbol{x_{in}}\right)$ of the measured data from the detector $\boldsymbol{x_{in}}$. Its output values should be as close as possible to the values of this desired property, where the closeness is defined in terms of some specific metric [69]. In other words, a close approximation of the hypothetical ideal prediction function F is sought, which is assumed to always reconstruct the property correctly. The process of finding such an approximative function can be difficult, since it is a high-dimensional task. An especially promising way of tackling problems of this kind is the use of artificial neural networks and especially deep learning, which has proven to be very successful in the field of pattern recognition in the past years (e.g. achieving super-human performance in the board game Go [70]; image recognition for the image net challenge [71]).

This chapter will give an overview over the theoretical background of artificial networks, and introduce practical aspects for analyzing KM3NeT/ORCA data. Sections 5.1 to 5.5 are in large parts taken verbatim from my Master's thesis [72], which additionally also includes a detailed explanation of some common network components like dense layers, convolutional layers or the batch normalization. Some of these are also used as a part of the network architecture of this work, which is introduced in chapter 6.

5.1. Neurons and layers

The deep learning techniques applied in this thesis are based on artificial neural networks, a term coined by the fact that they were originally inspired by the biological function of brain cells. In this spirit, an artificial neural network is a system of connected nodes named artificial neurons, each of them typically applying a non-linear function - the activation function - to its inputs. The output of this operation can then again be fed into other neurons, or can be taken as the output of the network as a whole.

In practice, a neuron will usually apply the following operation to its N inputs $\{x_j | j = 1, ..., N\}$:

$$f(x_1, x_2, ..., x_N) = \Theta\left(\sum_{j=1}^{N} \omega_j x_j + b\right) \tag{5.1}$$

That is, a sum over every input to the neuron $x_j \in \mathbb{R}$ is taken, each weighted with a real number ω_j. A constant, the bias b, can be added before applying the activation function of this neuron Θ.

Often, the neurons are ordered in layers. If the input to a certain layer consist only of neurons from the previous layer, it is called a feed-forward network. In this case, the output $x_i^{(n)}$ of neuron i in the n-th layer is given as a function from the output $x_j^{(n-1)}$ from the N neurons in the previous $(n-1)$-th layer by the following expression:

$$x_i^{(n)}\left(x_1^{(n-1)}, x_2^{(n-1)}, ..., x_N^{(n-1)}\right) = \Theta\left(\sum_{j=1}^{N} \omega_{i,j}^{(n)} x_j^{(n-1)} + b_i^{(n)}\right), \tag{5.2}$$

with $\omega_{i,j}^{(n)}$ denoting the weight between neuron j in the $(n-1)$-th layer, and neuron i in the n-th layer. Note that the number of neurons N can change from layer to layer. With this terminology, the output $\vec{x}^{\,(n)}$ of all neurons in layer n can be compactly written in vector notation with the weight matrix $\boldsymbol{\omega}$ as follows:

$$\vec{x}^{\,(n)}\left(\vec{x}^{\,(n-1)}\right) = \Theta\left(\boldsymbol{\omega}^{(n)} \cdot \vec{x}^{\,(n-1)} + \vec{b}^{\,(n)}\right), \tag{5.3}$$

where Θ is applied element-wise to a vector. The weights in the matrix $\boldsymbol{\omega}^{(n)}$ and the biases $\vec{b}^{\,(n)}$ are the free parameters of the n-th layer of the network.

Apart from the input and the output, all layers inside the network are called hidden layers. As an example, consider the simple case of a network with just one hidden layer: The input $\vec{x}^{\,(0)}$ is taken as the 0-th layer, and the output $\vec{y}^{\,(2)}$ of the whole network is then given by iteratively inserting Equation 5.3 :

$$\vec{y}^{\,(2)} = \Theta\left(\boldsymbol{\omega}^{(2)} \cdot \vec{h}^{\,(1)} + \vec{b}^{(2)}\right) = \Theta\left(\boldsymbol{\omega}^{(2)} \cdot \Theta\left(\boldsymbol{\omega}^{(1)} \cdot \vec{x}^{\,(0)} + \vec{b}^{(1)}\right) + \vec{b}^{(2)}\right), \tag{5.4}$$

with $\vec{h}^{\,(1)}$ being the output of the hidden layer.

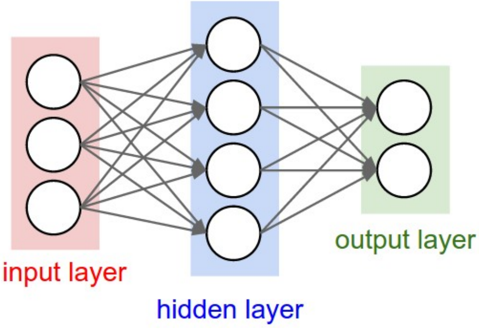


Figure 5.1.: Scheme of a shallow, fully connected feed-forward network. Every circle represents a neuron, and every line connecting two neurons stands for the corresponding weight. Connections only exist between neurons in subsequent layers [73].

A network with only a single hidden layer, like the network defined by the function above, is called shallow. A scheme of such a network is shown in Figure 5.1. If Θ is chosen to be a non-linear function, a network with this simple architecture is theoretically already capable of approximating any continuous function to arbitrary precision, as long as there are enough neurons in the network [74]. For this, the free parameters of the network - that is, the weights $\boldsymbol{\omega}$ and the biases $\vec{b}$ - can be chosen to resemble the target function as closely as desired.

When it comes to practical application of networks, stacking multiple hidden layers, and therefore making the architecture of the network deeper, can improve the ability of the network to reproduce a desired function. This approach is called deep learning.

5.2. Backpropagation and gradient descent

As mentioned in the introduction to this chapter, we seek to approximate the theoretical, ideal prediction function $F\left(\boldsymbol{x_{in}}\right)$, which predicts certain properties of the original particle as a function of the measured data $\boldsymbol{x_{in}}$ from the detector. If this approximation is realized with the help of artificial neural networks, the approximation function $\hat{F}_{\boldsymbol{\omega},\vec{b}}\left(\boldsymbol{x_{in}}\right)$ is defined via its free parameters, the weights $\boldsymbol{\omega}$ and biases $\vec{b}$ of all the layers in the network. These free parameters then have to be chosen to closely approximate the desired ideal prediction function.

Usually, this is achieved in an iterative process by assessing the quality of the prediction of the network on a small sample of so-called labeled data (i.e. data for which the true value for the desired output is known). The difference between the target output and the actual output of the network is measured with a problem-specific metric. Then, the weights and biases of the network are tweaked to reduce the metric distance, and the quality is reassessed. This process of training the network on data samples is repeated over and over until the metric distance reaches a minimum.

Since neural networks can contain millions of free parameters, many training steps are needed before the network is fully converged, and a large pool of labeled data is required. For the KM3NeT/ORCA data analysis in this thesis, this data is obtained by simulating events using the known properties of the detector.

A common technique to optimize the parameters of a network is called back propagation, a concept that is crucial for deep learning [75]. For an input to the network $\vec{x}^{\,(0)}$, the network predicts the desired property of the particle to be $\hat{F}\left(\vec{x}^{\,(0)}\right)$, whereas the target output - the actual property of the particle - is given by $F\left(\vec{x}^{\,(0)}\right)$.

Note, that the input $\vec{x}^{\,(n)}$ is written here as a vector, but this does not imply that the input necessarily has a one-dimensional data structure. It could as well be a two-dimensional image, or a high-dimensional graph, all of which can be linearized into a 1-D vector. Regardless of the dimension, the connections between a layer and the next can still be described by a two-dimensional weight matrix: each of its elements defines the connection between exactly two neurons, independent of each of their positions in the potentially higher dimensional data structure. For this, all neurons can be assigned a unique number, which is used to identify both their position in the vector $\vec{x}^{\,(n)}$, and their corresponding column or row in the weight matrix.

How well the network managed to approximate the desired output is defined by a cost function C, which gives the metric distance between two output vectors depending on the weights and biases of the network. An example for such a cost function is the squared vector norm difference (also known as mean squared error or MSE), in which case the cost function for a specific input $\vec{x}^{\,(0)}$ is given by:

$$C(\hat{F}_{\boldsymbol{\omega},\vec{b}}, \vec{x}^{\,(0)}) = \|\hat{F}_{\boldsymbol{\omega},\ \vec{b}}\left(\vec{x}^{\,(0)}\right) - F\left(\vec{x}^{\,(0)}\right)\|^2, \tag{5.5}$$

with the vector norm $\|.\|$ and the true output $F\left(\vec{x}^{\,(0)}\right)$. In principle, the cost function can be defined arbitrarily, and the best choice is often highly problem-specific (see also section 5.4 for a discussion of cost functions). The lower the value of the cost function is for a certain set of free parameters, the better is the prediction of the network. Therefore, the goal of the training process is to find a set of parameters which minimizes the cost function for the whole training dataset. This would require calculating the above cost function for all samples in the dataset. Since the datasets used for training networks tend to be quite large, the cost function for all samples in the dataset is usually approximated by the cost function for a small batch of training data, so that the calculation can be achieved in a reasonable time frame.

After the cost function of the network for the dataset was computed (or approximated by using a batch), the weights and biases of the network will be adjusted slightly to decrease the value of the cost function via the method of gradient descent. For this, the free parameters are changed in the direction of the negative gradient of the cost function:

$$\boldsymbol{\omega} \mapsto \boldsymbol{\omega} - \eta \frac{\partial C(\hat{F}_{\boldsymbol{\omega},\vec{b}})}{\partial \boldsymbol{\omega}}, \tag{5.6}$$

and similar for the biases $\vec{b}$. The parameter η defines the step size of the descent, and is also called the learning rate.

By applying the chain rule, the derivative of the cost function that appears in the gradient step of Equation 5.6 becomes

$$\frac{\partial C(\hat{F}_{\boldsymbol{\omega},\vec{b}})}{\partial \boldsymbol{\omega}} = \frac{\partial C(\hat{F}_{\boldsymbol{\omega},\vec{b}})}{\partial \hat{F}_{\boldsymbol{\omega},\vec{b}}} \cdot \frac{\partial \hat{F}_{\boldsymbol{\omega},\vec{b}}}{\partial \boldsymbol{\omega}}. \tag{5.7}$$

The first factor depends on the choice of the cost function. For example, when using the squared vector norm distance from Equation 5.5, it would result in

$$\frac{\partial C(\hat{F}_{\boldsymbol{\omega},\vec{b}})}{\partial \hat{F}_{\boldsymbol{\omega},\vec{b}}} = 2 \| \hat{F}_{\boldsymbol{\omega},\,\vec{b}} \left(\vec{x}^{\,(0)} \right) - F \left(\vec{x}^{\,(0)} \right) \|. \tag{5.8}$$

The second factor in Equation 5.7 is the derivative of the output of the network with respect to a certain weight inside of the network. The output is a known function of the input (an example was given in Equation 5.4 for a shallow network), so the derivative can be calculated analytically.

In fact, just like the forward pass through the network, the backpropagation can also be calculated iteratively on a per-layer basis. For this, consider the general case of calculating the derivative of layer n: Let α be a weight or a bias in a layer somewhere in the network before the current layer n. The output of a specific neuron in the current layer n is given as a function of the outputs of the previous layer $n-1$ by

$$x_i^{\,(n)} \left(\vec{x}^{\,(n-1)} \right) = \Theta \left(\boldsymbol{\omega}^{(n)} \cdot \vec{x}^{\,(n-1)} + \vec{b}^{\,(n)} \right)_i = \Theta \left(\sum_{j=1}^{N} \omega_{i,j}^{(n)} x_j^{(n-1)} + b_i^{(n)} \right), \tag{5.9}$$

similar to Equation 5.3. Note that the output of the previous layer $\vec{x}^{\,(n-1)}$ depends on all the weights and biases before that layer in the network, including the target α. The weights $\boldsymbol{\omega}^{(n)}$ and biases $\vec{b}^{\,(n)}$, however, do not, as they are constants. Therefore, the derivative of the above expression is given by

$$\frac{\partial x_i^{\,(n)}}{\partial \alpha} = \Theta' \left(\sum_{j=1}^{N} \omega_{i,j}^{(n)} x_j^{(n-1)} + b_i^{(n)} \right) \cdot \sum_{k=1}^{N} \omega_{i,k}^{(n)} \cdot \frac{\partial x_k^{\,(n-1)}}{\partial \alpha}, \qquad \alpha \text{ not in } \boldsymbol{\omega}^{(n)}. \tag{5.10}$$

This equation expresses the derivative of the output of the current layer using the output from the previous one. It can be applied recursively, until the layer m containing the target parameter α is reached. Depending on whether α is a weight or a bias, the derivative is then given by one of the following:

$$\frac{\partial x_i^{\,(m)}}{\partial \alpha} = \Theta' \left(\sum_{j=1}^{M} \omega_{i,j}^{(m)} x_j^{(m-1)} + b_i^{(m)} \right) \cdot \delta_{i,l}\; x_k^{\,(m-1)}\,, \qquad \alpha = \omega_{l,k}^{(m)} \tag{5.11}$$

$$\frac{\partial x_i^{\,(m)}}{\partial \alpha} = \Theta' \left(\sum_{j=1}^{M} \omega_{i,j}^{(m)} x_j^{(m-1)} + b_i^{(m)} \right) \cdot \delta_{i,l}\,, \qquad \alpha = b_l^{(m)}, \tag{5.12}$$

with the Kronecker delta δ. By introducing the definition

$$\boldsymbol{M}_{i,k}^{(n)} \equiv \Theta' \left(\sum_{j=1}^{N} \omega_{i,j}^{(n)} x_j^{(n-1)} + b_i^{(n)} \right) \cdot \omega_{i,k}^{(n)}\,, \tag{5.13}$$

Equation 5.10 can be compactly written in matrix and vector notation again, similar like during the forward propagation:

$$\frac{\partial \vec{x}^{\,(n)}}{\partial \alpha} = \boldsymbol{M}^{(n)} \cdot \frac{\partial \vec{x}^{\,(n-1)}}{\partial \alpha}, \qquad \alpha \text{ not in } \boldsymbol{\omega}^{(n)}. \tag{5.14}$$

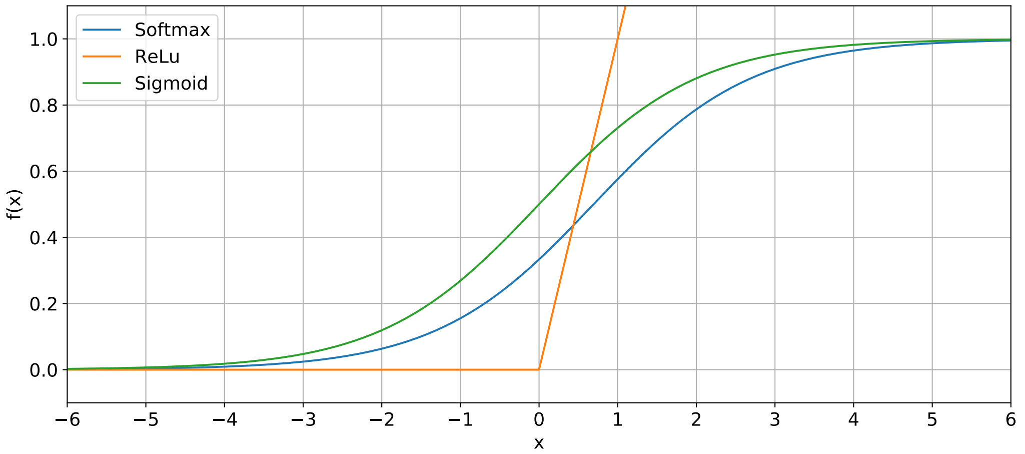


Figure 5.2.: Example of three different, commonly used activation functions: The sigmoid function, the rectified linear unit (ReLu) and the Softmax function. For the first two, x is the output of the neurons prior to applying the activation function. The Softmax function requires a multi-dimensional argument. Here, it was plotted for the case of a layer with three neurons in which two are assumed to output zero, that is, $f(\vec{x}) = f(x, 0, 0)$.

The matrix $\boldsymbol{M}^{(n)}$ defines the step between a layer n and the previous one in the network during the process of backpropagation. It is independent of the target parameter α, provided that this variable is not in the current layer n. This makes the backpropagation algorithm very computationally efficient, since most calculations have to be done only once per input for the whole network, and not specifically for every single weight or bias. Furthermore, since both the forward and the backward pass through the network require the multiplication of large and numerous matrices, they can be run in parallel and therefore sped up substantially by using Graphics Processing Units (GPUs). Popular deep learning frameworks like TensorFlow, which is used in this work, or PyTorch feature easy built-in ways to accelerate network operations with GPUs.

5.3. Activation functions

In Figure 5.1, it has been established that neural networks can represent functions to arbitrary precision if the activation function Θ of layers is chosen to be non-linear. This is of course a very general criterion, leaving plenty of functions to choose from. Historically speaking, artificial neural networks were designed to resemble the neural structure of the brain. In a simplified model, the biological neuron receives signals from its dendrites, sums the signal strengths and, if it is above a certain threshold, the neuron fires along its output, the axon. To mimic this behavior, a possible choice for the activation function in artificial networks is the **sigmoid** function:

$$\Theta(x) = \frac{1}{1 + e^{-x}}, \tag{5.15}$$

which is also plotted in Figure 5.2. Depending on the sum of its inputs, this function can be close to zero (neuron does not fire) or one (neuron does fire). It is also differentiable in the whole range, which is a requirement for the backpropagation algorithm (see Equation 5.10). However, this choice of an activation function has become less popular, since it not only seems to be quite far from how actual neurons in the brain work (see e.g. [76] for a recent discussion of the computation in biological neurons), but is also suboptimal for the training process in artificial networks, since it possesses some undesirable properties.

For example, it can be seen in Figure 5.2 that the sigmoid function converges exponentially to zero or one for very small or very large x. This in turn means that the gradient of the function is very close to zero in these areas. Since the matrix which defines the back propagation step depends on this gradient (Equation 5.13), the weight updates in a layer in which the inputs have a large absolute value can get very small, too. Furthermore, since the weight updates in the previous layers depend on the back propagation matrix of that layer as well, this can actually slow the training process in all preceding layers, a phenomenon known as vanishing gradients.

Nowadays, the sigmoid function is often used exclusively in the last layer of classification networks, together with the categorical cross entropy cost function. This function is explained in greater detail in section 5.4. For this purpose, the sigmoid function can be generalized for multiple neurons, as networks are frequently used to classify their inputs into multiple different categories and therefore have multiple neurons in the output. It is then called a **Softmax** function, whose output is given by:

$$\Theta_j(x_1, ..., x_K) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}, \qquad \text{for } j = 1, \ldots, K. \tag{5.16}$$

Since the sum over the output of all neurons in a layer using the softmax activation function is one, it can be used to model a probability distribution. A plot for the specific case of $K = 3$ neurons, in which two neurons x_k are always zero, is plotted in Figure 5.2. The curve of the Softmax function is identical to the sigmoid function, shifted in the positive x direction by the natural logarithm of the sum of the other neurons' outputs.

A popular choice for the activation in the more recent past is the rectified linear unit (**ReLU**, see Figure 5.2), which is given by:

$$\Theta(x) = \max(0, x). \tag{5.17}$$

Both the ReLU itself, as well as its derivative are very inexpensive to compute, which is a big upside compared to the sigmoid function. Also, this function does not saturate in the positive x regime. However, there is the risk that the weights defining the input of the ReLU are set in such a way that it will always output zero for all samples in the batch or even the entire dataset. In this case, the neuron would not receive updates to its weights anymore, and would thus be permanently stuck in this state. In this case, it is called an inactive neuron.

Nevertheless, ReLUs have shown to yield very good results over the sigmoid function (e.g. in [77], one of the first networks to use the ReLU activation in image classification), and they will be used for most of the neurons in this thesis as well.

To further combat the problem of inactive neurons, various modifications of the ReLU function have been proposed, e.g. the leaky ReLU or the parametric ReLU. They have been shown to be able to improve the performance of networks in specific cases [78].

5.4. Cost functions

In Equation 5.7, the first step of the backpropagation has been described, which depends on the cost or loss function $C(\hat{F}_{\omega,\vec{b}})$. In general, this cost function specifies the goal of the network training, as it defines how close the prediction $\hat{F}_{\omega,\vec{b}}$ of the network is to the desired output F, given a specific input. Typically, networks will conduct one of the two following tasks:

- Classification: The input can be sorted in different categories. For example, the particle in an event can be classified as a neutrino or as an atmospheric muon.

- Regression: The input depends on an underlying observable, and the network should reconstruct this quantity. Such observables are for example the incident direction or energy of a particle traveling through the detector.

Depending on the task, different choices of loss functions are popular.

In the case of a regression, a common choice is the mean squared error loss function, which was already introduced in section 5.2. For a given input, let the network predict the desired quantities to be $\hat{F}_{\omega,\vec{b}} \equiv \vec{y}^{\,pred} = (y_1^{pred}, ..., y_n^{pred})$, while the true values are $F \equiv \vec{y}^{\,true} = (y_1^{true}, ..., y_n^{true})$. The **mean squared error** then amounts to:

$$C(\vec{y}^{\,true}, \vec{y}^{\,pred}) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i^{true} - y_i^{pred} \right)^2 . \tag{5.18}$$

This function is always positive, and reaches its minimum when the prediction from the network is exactly equal to the true value. Another popular choice for regression tasks is the **mean absolute error**, which is defined in a similar fashion as above:

$$C(\vec{y}^{\,true}, \vec{y}^{\,pred}) = \frac{1}{n} \sum_{i=1}^{n} \left| y_i^{true} - y_i^{pred} \right| . \tag{5.19}$$

In the case of a classification task, a frequently used loss function is the categorical cross entropy: Given the prediction of the network $\hat{F}_{\omega,\vec{b}} \equiv \vec{y}^{\,pred} = (y_1^{pred}, y_2^{pred}, ..., y_n^{pred})$, which sorts its input into n different categories, and the desired output $F \equiv \vec{y}^{\,true}$ with the same number of categories, the **categorical cross entropy** loss function is defined as:

$$C(\vec{y}^{\,true}, \vec{y}^{\,pred}) = -\sum_{i=1}^{n} y_i^{true} \ln y_i^{pred} . \tag{5.20}$$

This loss function is often used in conjunction with the softmax activation function defined in Equation 5.16. The outputs from a layer with this activation function will always be between zero and one after passing through this activation function, and the sum over all outputs will be one. This makes it similar to a probability distribution. However, if the softmax function is in the saturated regime, in which the input to one neuron is much higher than those of the others, the gradient of the function becomes exponentially small (see Figure 5.2 for very large or small x). In the mean squared error from Equation 5.18, the activation function itself appears in the definition, since $\vec{y}^{\,pred}$ is the output of neurons which apply this function. If neurons get in the saturated regime, their weights and biases may only receive small updates, thereby slowing the training process. With the cross entropy loss, in contrast, the cost function can be written as:

$$C\left(\vec{y}^{\,true}, \vec{y}^{\,pred}\right) = -\sum_{i=1}^{n} y_i^{true} \left(x_i - \ln \sum_{k=1}^{K} e^{x_k} \right), \qquad \text{with } y_i^{pred} = \theta(x_i), \tag{5.21}$$

by inserting Equation 5.16 in Equation 5.20. Here, the outputs $x_1, ..., x_K$ of the neurons prior to the activation function appear as a linear factor in the cost function, since the logarithm of the the categorical cross entropy cancels out with the exponentiation in the softmax activation. This prevents the neurons from getting stuck in the saturated regime of the softmax activation during backpropagation.

5.5. Optimizers

During backpropagation, the derivative of the cost function $C(\hat{F}_{\omega,\vec{b}})$ with respect to all the weights and biases in the network is calculated. Afterwards, the free parameters are updated in order to minimize the loss.

In Equation 5.6, a simple way of updating was shown, in which a fraction of the gradient of the cost function is subtracted from every parameter, which is known as gradient descent. In fact, since the derivative of the cost function was not calculated for the entire training dataset, but rather approximated via a small batch of randomly chosen samples from the set, it is more accurately called stochastic gradient descent (SGD). However, there are also other strategies of updating the weights and biases, which for example try to accelerate the learning process by modifying Equation 5.6 in various ways. The specific strategy after which the free parameters are updated is referred to as the optimizer. A popular one, which was also chosen for most of the networks trained in this thesis, is the adaptive momentum estimation (adam).

The adam optimizer was introduced by Kingma and Ba in 2014 [79]. It attempts to speed up the convergence of network training by setting the step size η from Equation 5.6 to not be constant for the entire gradient step, but instead makes it adapt to the current situation of convergence, and allows different free parameters to have different effective step sizes. For this, decaying averages over the past gradients and squared gradients are kept track of, which estimate the mean $\vec{m}_t$ and the uncentered variance $\vec{v}_t$ of the gradient updates at the current step t:

$$\vec{m}_t = \beta_1 \cdot \vec{m}_{t-1} + (1-\beta_1) \cdot \vec{\nabla}_\omega C(\hat{F}_{\boldsymbol{\omega},\vec{b}}) \tag{5.22}$$

$$\vec{v}_t = \beta_2 \cdot \vec{v}_{t-1} + (1-\beta_2) \cdot \left(\vec{\nabla}_\omega C(\hat{F}_{\boldsymbol{\omega},\vec{b}})\right)^{\circ 2} , \tag{5.23}$$

where the square operation $^{\circ 2}$ in the last line uses the Hadamard product, in which the vectors are multiplied element-wise. Every element in the vectors $\vec{m}_t$ and $\vec{v}_t$ are the moving moment estimates of a specific free parameter in the network. Similar equations can be defined, if the parameter with respect to which the differentiation is computed is a bias b instead of a weight ω. The two constant hyperparameters β_1, β_2 introduced in the above equations define the rate at which past gradients and variances are exponentially decayed in the saved momentum estimates, and are usually set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as recommended in the original paper.

The above equations will result in estimates that are heavily biased towards zero in the early stages of training, since the original estimates $\vec{m}_0$ and $\vec{v}_0$ are initialized as vectors of zeros. This can be fixed by scaling the contribution of moment estimates in early gradient steps t towards the decaying average:

$$\vec{\widehat{m}}_t = \frac{\vec{m}_t}{1-\beta_1^t} \tag{5.24}$$

$$\vec{\widehat{v}}_t = \frac{\vec{v}_t}{1-\beta_2^t} \tag{5.25}$$

In total, the gradient update with the adam optimizer for a specific weight or bias in the network is given with the corresponding bias-corrected momentum estimates $\widehat{m}_t$ and $\widehat{v}_t$ by:

$$\omega \mapsto \omega - \alpha \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t + \epsilon}} . \tag{5.26}$$

Two new hyperparameters were introduced here: The step size α, which allows to scale the gradient step of all weights simultaneously as with the SGD optimizer, and ϵ, a small constant which was introduced for numerical stability. The authors of the paper recommend a value of $\epsilon = 10^{-8}$.

The fraction $m_t/\sqrt{v_t}$ is compared to the signal-to-noise ratio of the gradient by the authors. If this ratio is small, that is, if there are large changes in the gradient from batch to batch, the adam optimizer automatically adapts to this by decreasing the effective learning rate, since this might indicate a close proximity of the weight to its optimum. In contrast, if the ratio is high, then the direction of the gradient seems to be clear, even within the limited statistics of single batches of samples. Thus, an increase in the effective learning rate could accelerate the learning process.

5.6. Estimating uncertainty in regressions

In section 5.4, some commonly used cost functions were introduced, including the categorical cross entropy combined with a softmax activation function in the last layer of a network. With this combination, the output of the network is a probability distribution over the different output categories. This is a very useful property, as it allows for cuts based on the probability of belonging to one of the categories on an event-by-event basis.

For example, consider the case of a network that classifies neutrinos against the overwhelming background of atmospheric muons. In its simplest form, a classifier would sort each event to be either a neutrino or a muon. However, a network using the categorical cross entropy and softmax combination instead assigns each event a *probability* of being a neutrino or muon. In this case, we can increase the purity of the resulting neutrino sample by demanding a stricter cut on this neutrino probability [80].

For regression tasks, such a quality cut is not possible if we use a popular loss like the mean squared error from Equation 5.18. In this case, the output of the network is a single number - the estimator for the observable that we want to reconstruct - without any indication of the associated uncertainty. This is suboptimal, as the error can play a similarly important role for regressions: It obviously makes a big difference whether the zenith direction of a muon track is reconstructed as 5° ± 0.1° or 5° ± 50°. The latter case indicates a problem with the reconstruction for this event, which we might want to cut away before carrying out any further analyses.

However, it is also possible to acquire the uncertainty for reconstruction tasks by making use of the concept of mixture density networks [81]. In this picture, the minimization of a loss function in neural networks in general is seen as performing a maximum likelihood fit given a probability distribution that is parametrized by the network, independent of whether we are dealing with a categorization or a regression problem.

For understanding this, it is useful to first consider again the case of a categorization over n categories. Let us assume that the output of the network is a categorical distribution $\vec{y}^{\,pred}$, that is, the final layer in the network has n neurons $({y_1}^{pred}, ..., {y_n}^{pred})$ with $\sum {y_i}^{pred} = 1$. Then, the likelihood of getting the correct category given the predicted distribution $\vec{y}^{\,pred}$ is

$$L\left(\vec{y}^{\,true}|\vec{y}^{\,pred}\right) = \prod_{i=1}^{n} (y_i^{pred})^{y_i^{true}}, \tag{5.27}$$

with the the true distribution $\vec{y}^{\,true}$ being 1 for the correct category, and 0 otherwise. For example, if we want to categorize events as showing either a neutrino or an atmospheric muon, the true distribution of a neutrino would be $\vec{y}^{\,true} = (1, 0)$. Given a prediction of $\vec{y}^{\,pred} = (0.8, 0.2)$, L would thus amount to 0.8.

We can then train the network by adjusting its free parameters in order to maximize the given likelihood. This is equivalent to minimizing a cost function C defined as the negative logarithm of the likelihood:

$$\boxed{C(\vec{y}^{\,true}, \vec{y}^{\,pred}) := -\ln L\left(\vec{y}^{\,true}|\vec{y}^{\,pred}\right)} \tag{5.28}$$

For the categorization problem, we can calculate C by inserting the likelihood of the categorical distribution from Equation 5.27:

$$C(\vec{y}^{\,true}, \vec{y}^{\,pred}) = -\ln\left(\prod_{i=1}^{n} (y_i^{pred})^{y_i^{true}}\right) = -\sum_{i=1}^{n} y_i^{true} \ln y_i^{pred}. \tag{5.29}$$

The term on the right is exactly the categorical cross entropy defined in Equation 5.20. The necessary condition $\sum {y_i}^{pred} = 1$ for the output of the network can be assured by choosing an appropriate activation function, like the softmax function.

In summary, we let the output of the network parametrize a distribution by constraining it with an appropriate activation function, and then used Equation 5.28 to obtain the cost function. This is a universal approach, which can be used for regression problems as well.

For example, we can let the network parametrize a normal distribution by letting the output layer have two neurons representing $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{>0}$. It can be assured that the variables are in the defined intervals by choosing e.g. a linear and an exponential activation function, respectively. Then, the likelihood is given by the Gaussian function

$$L\left(y_{true}|\vec{y}_{pred} = (\mu, \sigma)\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right). \tag{5.30}$$

By inserting this into Equation 5.28, we end up with the cost function

$$C(y_{true}, (\mu, \sigma)) = \ln(\sqrt{2\pi}) + \ln(\sigma) + \frac{(y_{true} - \mu)^2}{2\sigma^2}. \tag{5.31}$$

This is the cost function for a regression with an included uncertainty σ. Since we are interested in minimizing C with respect to μ and σ, we can simplify this equation by removing the constant summand $\ln(\sqrt{2\pi})$ and multiplying by 2:

$$C(y_{true}, (\mu, \sigma)) = \ln(\sigma^2) + \frac{(y_{true} - \mu)^2}{\sigma^2}. \tag{5.32}$$

It is worth noting that if we set $\sigma = 1$, we can recover the mean squared error loss function defined in Equation 5.18 for the case of a single output neuron $n = 1$. The case $n > 1$ can be obtained in the same way by letting the output of the network parametrize a multivariate normal distribution with a diagonal covariance matrix instead. This means that using the mean squared error can be seen as letting the network output a set of independent normal distributions with a fixed variance of $\sigma = 1$ each. In other words, Equation 5.32 is a generalization of the mean squared error loss function that includes an estimate of the uncertainty.

Naturally, a normal distribution is not the only choice for the output of the network. For example, we can also generalize the mean absolute error by letting the network parametrize a Laplace distribution $L = (2\sigma)^{-1} \exp(-|y_{true} - \mu|/\sigma)$. In fact, any desired distribution can be chosen by representing each free parameter in the distribution by an output neuron in the network. The best choice for this distribution is highly problem specific, and not necessarily trivial to figure out.

A possible solution for this issue is the concept of normalizing flows [82]. They allow the network itself to parametrize an arbitrary distribution during the training, and automatically adapt it to the reconstruction task at hand. First studies in the field of high-energy physics have shown promising results [83], but additional research for the specific use case in KM3NeT is required.

5.7. Deep learning software in KM3NeT

While modern machine learning frameworks like TensorFlow provide high level implementations for many important aspects of neural networks, processing the data and organizing the training still requires many complex steps. In order to facilitate the future use of deep learning in KM3NeT, it was made sure that all the studies in this thesis can be easily reproduced and followed up by providing well documented and tested software. For this, the first iteration of the software originally developed by Michael Moser [84] has has been rewritten in large parts, and drastically expanded in its functionality. It is split into two Python packages, OrcaSong[1] for preprocessing the data, and OrcaNet[2] [85] for training the networks. Despite the name, the software is also

[1] https://git.km3net.de/ml/OrcaSong

[2] https://git.km3net.de/ml/OrcaNet

used for data from the Arca and the Antares detector [86][87]. All of the code is published under an open source license, and is accessible online.

5.7.1. OrcaSong

OrcaSong is used to preprocess KM3NeT files to a format that is specifically tailored to the needs of machine learning applications. The starting point are KM3NeT offline files in the aanet root data format on the trigger or reconstruction level. The disk space usage and read-out speed of these files can be optimized by removing the information that is not needed for the application of neural networks. In addition, some expensive calculations, like for example counting the number of hits that each muon in a bundle produced in the detector, are ideally only performed once, and not every time an event is encountered during the training. Finally, it should be made sure that events containing different particles or runs from different time periods are properly shuffled. This way, it can be made sure that the statistics in each minibatch of samples during the training is representative of the entire dataset.

OrcaSong addresses all of these points, encouraging reproducibility and sharing of workflows at the same time by letting the user define their setup through a simple, human readable configuration file. With this, the software can be run with commands directly from the command line. An extensive documentation has been published alongside OrcaSong, which describes the necessary steps in more detail. In short, the preprocessing roughly happens in two stages:

1. **Generation of hdf5 DL files** Hdf5 is a open source big data format, which is especially popular in the machine learning community. It features compression, high read out speed, and an easy to understand, self describing format. For this reason, it was chosen as the format of the DL files that are used as an input to the neural networks. The DL files themselves essentially consist of two datasets, which contain the samples (`x`) and the labels (`y`), respectively. The samples in `x` can be either images for convolutional networks, or the full hit information for the generation of graphs. In addition to the labels, the `y` dataset also contains other event-level information that is useful for the later stages of the analysis, like event identifiers or the output of the classical reconstruction. The kind of information that is contained in `x` and `y` is defined by configuration files. OrcaSong comes with examples for such files, which have been used for the analysis of neutrinos and atmospheric muon bundles. The files used for this thesis can also be found in the appendix section A.1.

2. **Shuffling the training files** As mentioned before, the order of events in the training files should be as random as possible. Therefore, OrcaSong contains two high level commands which can be used to merge different DL files into a single, bigger one, and to randomize the order of events in any DL file. Together, these scripts can be used to create large, properly shuffled DL files, with a mixed input from many files of a run-by-run simulation or different neutrino flavors. Since the scripts operate on a file level, the complexity of this step is kept as low as possible. This stage is of course only necessary for the training files, since the order of events is irrelevant during inference, in which the weights are constant.

5.7.2. OrcaNet

OrcaNet handles all the typical tasks that occur when handling deep neural networks, like efficiently reading data, executing and monitoring the training, or saving the predictions of the fully trained model to an hdf5 file. It is designed with the restrictions of working on a computing cluster in mind, as regular checkpoints allow to easily split the training up over several consecutive jobs. Similar to OrcaSong, it can be entirely controlled via simple commands and configuration files. Additionally, it can also be used to construct the model itself from such a configuration file. This is especially useful for newcomers, as the hyperparameters of the model can be varied without any deeper knowledge about Python or TensorFlow. In total, the following three configuration files can be used to set up OrcaNet:

list.toml: A list of the paths to the DL files created by OrcaSong that will be used for the training and validation. It also can also contain paths to DL files on which the network should be applied on in inference mode after the training is complete. This is typically used for the test set, or for predictions on measured data.

config.toml: Parameters that will be used during the training, like the learning rate or the batchsize.

model.toml: Definition of the architecture of the model, using predefined blocks of layers which are often used together in common architectures. In principle, this file is optional, as any compiled keras model can be used with OrcaNet. However, the use of a configuration file guarantees reproducibility and makes sharing an architecture easy.

Often times, small modifications need to be made to the samples or labels that are read from a DL file before it can be handed to the network. For example, classifications typically require the label to be in the so-called one-hot encoding, i.e. the classes $\{1, 2, 3\}$ need to be encoded as vectors $\{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$. This can be done in OrcaNet using small, user-defined functions referred to as `modifiers`. Several commonly used functions, like the one-hot encoding mentioned above, are already implemented in OrcaNet and can be used by specifying them as a string in the config.toml file. For the example of the one-hot encoding, the corresponding string is `ClassificationLabels`.

In order to start the training of a new instance of a model, the three configuration files described above can be moved in a new, empty directory. Then, the command line tool `orcanet train name_of_directory` is called on this directory, which per default will train for an unlimited number of epochs. If the training gets interrupted, for example because the walltime of the job in a computing cluster ran out, it can be resumed from the last auto-generated checkpoint by running the same command again. Finally, the prediction of the network can be saved to an hdf5 file by running the command `orcanet inference name_of_directory`. For this, the network with the lowest validation loss across the entire training history will be loaded automatically. All the OrcaNet configuration files used for this thesis are listed in the appendix section A.2.

Chapter 6.

Graph neural networks

Deep learning has seen a large surge in popularity for image recognition tasks over the last decade, making use of model architectures involving discrete convolutions on the rectangular grid that images are made of. These designs have also been successfully applied in the context of high-energy particle physics, for example in KM3NeT [80], IceCube [88] or CTA [89].

However, the nature of the data measured at experiments like KM3NeT differs significantly from the images or videos these architectures were originally designed for. These high dimensional and sparse signals bear much closer resemblance to point clouds, which can be well represented by the general structure of a graph. Graph neural networks are deep learning algorithms working in the domain of graphs, which have the potential to improve the performance of convolutional networks, as demonstrated by IceCube [90].

This chapter explores strategies for using graph neural networks in KM3NeT, and discusses basic choices like the architecture, as well as technical details of the implementation developed for this thesis.

6.1. Basics of graph theory

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is composed of a set of N nodes $\mathcal{V}$ and up to N^2 edges $\mathcal{E}$ [91]. Each edge connects two nodes, and can also have a direction, in which case $\mathcal{G}$ is called a directional graph. It can also connect the same node back to itself, which is called a self connection. The information of which nodes in a graph are connected to each other can be represented by the adjacency matrix $\boldsymbol{A}$. The entry at position (i, j) in the adjacency matrix has a non-zero value if the nodes i and j are connected, and is zero otherwise. In the simplest case, the non-zero values of the entries in the adjacency matrix are one, but they can also chosen to be any real number, in which case they represent the weight of each edge. In Figure 6.1, these definitions are visualized for a simple directional graph with unweighted edges.

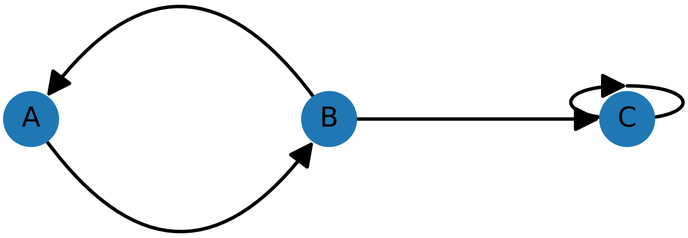
The degree matrix $\boldsymbol{D}$ of a directional graph is a diagonal matrix, whose entry $D_{i,i}$ is the degree of node i, i.e. the number of edges that point towards that node. Another important property of a graph is its Laplacian matrix $\boldsymbol{L}$, which is defined as the difference between the degree matrix $\boldsymbol{D}$ and the adjacency matrix $\boldsymbol{A}$:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{6.1}$$

It is the discrete analogue for the Laplacian operator of continuous functions [92].

6.2. Convolutional graph networks - spectral and spatial

Convolutions have proven to be a powerful tool in deep learning, and are found in virtually all modern image recognition architectures. It is therefore quite natural to look for a similar approach in graph based networks. However, transferring the concept of convolutions from images to the more general structure of graphs is not straight-forward, as neighborhoods are not as easy to define, and the computational effort can quickly become unreasonably large.



(a) Nodes and edges.

$$\begin{array}{c|ccc} & A & B & C \\ \hline A & 0 & 1 & 0 \\ B & 1 & 0 & 1 \\ C & 0 & 0 & 1 \end{array}$$

(b) Adjacency matrix.

Figure 6.1.: Example of a simple directional graph featuring 3 nodes and 4 edges. Nodes A and B are connected to each other in both directions. B is connected to C, but not the other way around. C also has a self connection. This structure is visualized in the plot on the left. Note that the positions of the nodes can be chosen freely in this representation, as the graph is defined only by the number of nodes and the connections between them. The adjacency matrix of the graph is shown on the right.

A possible strategy for tackling this issue in graph networks are spectral methods, which perform convolutions on spectral representations of graphs [93]. For this, we can first look at the well known definition of a continuous Fourier transformation of a function f:

$$\mathcal{F}\{f\}(\xi) = \int_{-\infty}^{\infty} e^{-2\pi i x \xi}\ f(x)\, dx. \tag{6.2}$$

The exponential term in Equation 6.2 is identical to the eigenfunctions $u_\xi(x) = e^{2\pi i x \xi}$ of the Laplacian operator Δ:

$$\Delta e^{2\pi i x \xi} = \frac{\partial^2}{\partial x^2} e^{2\pi i x \xi} = -(2\pi\xi)^2 e^{2\pi i x \xi}. \tag{6.3}$$

By using these eigenfunction, we can write the Fourier transformation in Equation 6.2 as:

$$\mathcal{F}\{f\}(\xi) = \int_{-\infty}^{\infty} u_\xi^*(x)\ f(x)\, dx. \tag{6.4}$$

In a similar way, we can also express the discrete Fourier transformation $\mathcal{F}_\mathcal{G}$ on a graph $\mathcal{G}$ in terms of the eigenvectors u_l and eigenvalues λ_l of the Laplace matrix $\boldsymbol{L}$ defined in Equation 6.1 [94]:

$$\mathcal{F}_\mathcal{G}\{f\}(\lambda_l) = \sum_{i=1}^{N} u_l^*(i)\ f(i). \tag{6.5}$$

Here, $f(i)$ is a discrete function on the N nodes of the graph. Using matrix notation, this can be compactly written as:

$$\mathcal{F}_\mathcal{G}\{\vec{f}\} = \boldsymbol{U}^T \vec{f}. \tag{6.6}$$

The basic motivation behind spectral graph methods is rooted in the convolutional theorem. In continuous space, a convolution $*$ between two functions f and g is defined as:

$$(f * g)(x) \triangleq \int_{-\infty}^{\infty} f(\tau) g(x-\tau)\, d\tau = \int_{-\infty}^{\infty} f(x-\tau) g(\tau)\, d\tau. \tag{6.7}$$

The convolutional theorem then states that

$$f * g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} \cdot \mathcal{F}\{g\}\}, \tag{6.8}$$

i.e. the convolution of two functions is equivalent to the point-wise multiplication of these functions in the Fourier transformed space. This is also true for the discrete version of the Fourier transformation on a graph as defined above:

$$f * g = \boldsymbol{U}(\boldsymbol{U}^T\vec{f} \cdot \boldsymbol{U}^T\vec{g}), \tag{6.9}$$

with the inverse graph Fourier transformation $\boldsymbol{U}$. The matrix $\boldsymbol{f_\theta} := \boldsymbol{U}^T\vec{f}$ can be seen as a filter kernel in spectral space, which can be defined and learned by a neural network with a set of parameters θ. A basic approach is to construct $\boldsymbol{f_\theta}$ as a non-parametric filter, i.e. as a diagonal matrix in spectral space [95]. However, this definition proves to be difficult to use in practice due to its computational inefficiency, as the resulting filter is not localized in space. Different approaches exist to solve this issue, for example by parametrizing $\boldsymbol{f_\theta}$ using an expansion with Chebyshev polynomials [96].

An alternative to the spectral approach explained above is the spatial approach. Here, the kernel that is learned by the network is simply directly defined in space in a localized way, which also removes the need to perform any Fourier transformations of the graph. In this sense, it is similar to the traditional approach of convolutional neural networks on images. While this is an easy way to solve the issue of localization, it poses the problem of matching the different neighborhoods of each node [95]. This is naturally much more difficult for graphs as compared to images, since graphs in general have an irregular structure, as opposed to the highly regular grid and therefore neighborhoods found in images. As will be shown in section 6.4, this spatial strategy is also adopted for the network architecture of this work.

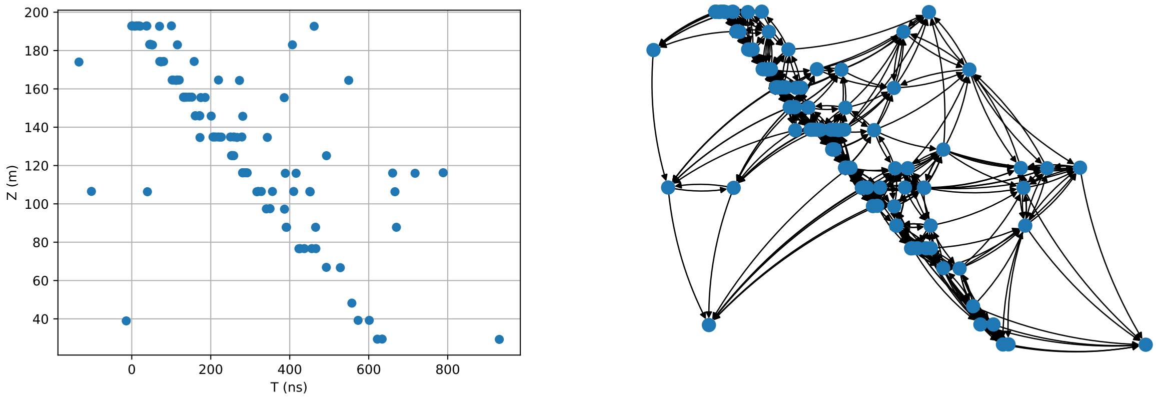
6.3. Transforming KM3NeT data to graphs

A KM3NeT event contains a set of hits, each with its own set of coordinates. For example, these coordinates include the x-, y- and z-position of where a particular hit was measured in the detector, the time t, as well as the pointing direction of the PMT that recorded the hit. A natural way of encoding the information of an event as a graph is to view each hit as a node, and the features of each node as the aforementioned coordinates of that particular hit. This way, the full data of each hit can be fed into the network as float numbers, avoiding any loss of information in the process.

However, it is not clear a priori how the nodes should be connected to each other in the graph. A straight-forward approach is to define some sort of distance between two nodes, and base the adjacency on that. Reasonable choices for such a distance can be the norm of the distance in the euclidean space $d^2 = c^2\delta t^2 + \delta x^2 + \delta y^2 + \delta z^2$, or in the Minkowski spacetime $d^2 = c^2\delta t^2 - \delta x^2 - \delta y^2 - \delta z^2$. Here, δx, δy, δz and δt denote the differences in space and time between the two hits that are represented by the corresponding nodes. For this work, the euclidean distance was chosen, but it is also worth to try out other metrics in future projects.

Even with the distance defined, there are still multiple ways to connect the nodes. A possible approach using the euclidean distance is to define a fixed maximum radius r, and then connect each node to all the nodes that are within that radius. While easy in theory, this poses multiple problems in practice: It is not clear how large or small r should be, making this a problem dependent hyperparameter. Furthermore, since the number of nodes that are within the radius is not limited, the resulting graph could have many connections. In the worst case, it might even happen that every node in a particular event gets connected to every other one. This could lead to memory issues, since the required VRAM would be proportional to the squared total number of nodes in the graph.

A different approach is to connect every node to its k nearest neighbors (knn) in terms of the chosen distance. This way, the total number of connections in the graph is always linear in the number of nodes, and thus also the memory usage. For this reason, the graphs in this work are



(a) z-t-plot of the event.

(b) Graph representation of the same event.

Figure 6.2.: Example of the transformation of a KM3NeT event into a graph, in this case an atmospheric muon event simulated with Mupage for Orca4. The plot on the left shows the z- and t-coordinate of each hit in its respective detection unit. On the right, each node in the graph represents one of the hits, and the edges connect to the eight nearest neighbors of each hit, calculated using the euclidean distance in x, y, z and t. The arrows are pointing away from the neighbors, and towards the central node. Note that the nodes are plotted in the same z-t position as in the left plot. This was chosen only to aid the visualization, as the position of the nodes can be chosen arbitrarily when plotting the graph.

constructed using the knn method based on the euclidean distance. Such a transformation into a graph is illustrated on the example of an atmospheric muon event in Figure 6.2.

Efficient data structures

Memory efficiency is crucial, not only when calculating the adjacency, but also when handling the data of the nodes. As established above, an event consist of N hits, each with a set of F hit features like the space and time coordinates of that hit. Overall, the data of an event can thus be stored efficiently as a two dimensional array with the shape (N, F).

In practice, we will often work with batches of events at a time, for example with multiple events in a run that are saved in the same file. This results in a more complex data structure: If we have M different events, each of them will typically have a different number of hits. So in total, we have a list of M arrays with shapes $(N_1, F), ..., (N_M, F)$. Even though the number of hits N_i differs from event to event, the number of hit features F is the same for each hit in each event.

We can therefore think of this batch of events as a single array of shape $(M, ?, F)$, where the question mark denotes an axis with variable length $\{N_1, ..., N_M\}$. Such an array is typically referred to as a ragged array. Having data in a ragged structure is a common occurrence, especially when dealing with graphs, so there are several efforts in the Python community to treat them efficiently and easy at the same time.

An efficient way to save such a ragged array to disk is to transform it into a single, big array with shape (N_{tot}, F) of the hits in all the events, and additionally a small vector of length M that contains the number of hits N_i in each event. A scheme of this strategy is shown in Figure 6.3.

In the TensorFlow framework, a special class exists to handle data with ragged properties: a TensorFlow ragged tensor. It can be initialized by giving the large array of all hits and the small vector described above, and can then be used directly as an input to networks.

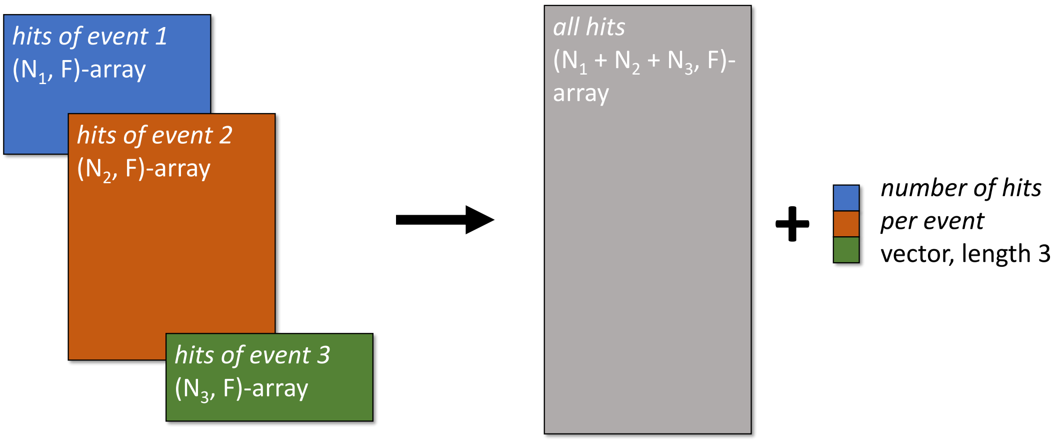


Figure 6.3.: Three events with a varying number of hits N_1, N_2, N_3 each, but the same number of hit features F. They can be stored efficiently in a single regular two-dimensional array of shape $(N_1 + N_2 + N_3, F)$, together with a small vector of length 3 that contains the number of hits per event.

6.4. ParticleNet and the Edge Convolutional block

The model architecture used in this work is based on the ParticleNet model by Qu and Gouskos [97], which makes use of the Edge Convolutional layer block proposed by Wang et al. [98]. Originally designed for point clouds, this block was adapted for jet tagging in LHC, and was shown to outperform the image convolutional based networks significantly in this task. Since KM3NeT data with its high dimensionality and sparse signal also resembles a point cloud, it is a promising choice of architecture here as well.

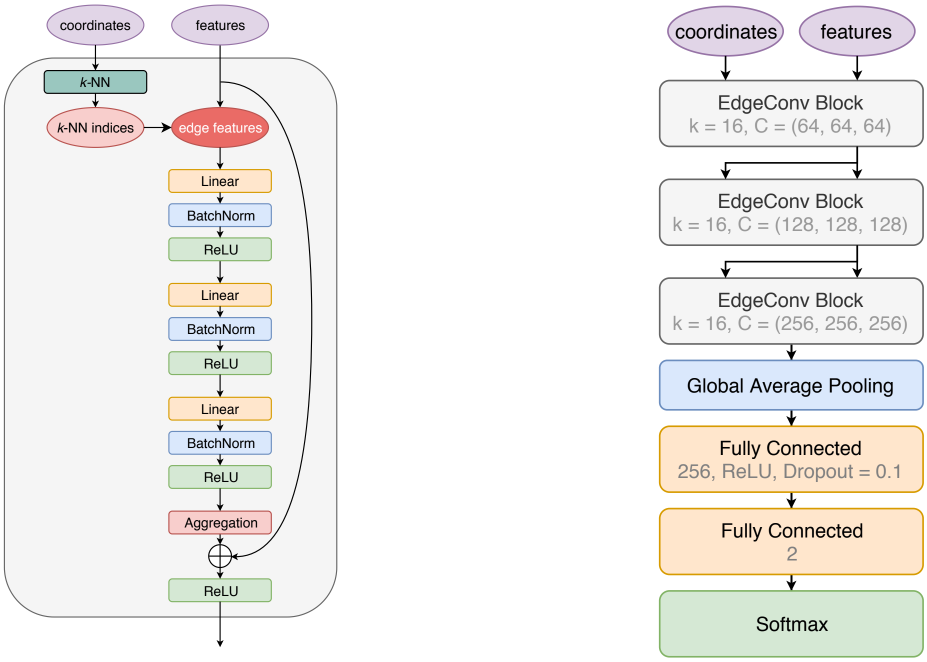
The Edge Convolutional block has two separate inputs: The first input contains the coordinates of each node and is used to determine which nodes are connected to each other. For this, each node is connected to its k nearest neighbors based on the euclidean distance of the coordinates. The second input is then used to calculate the node and edge features of the graph (see Figure 6.4a).

While the authors of ParticleNet provide the TensorFlow code used in their study, it is not sufficiently optimized to allow for the use with KM3NeT data. An optimized solutions exists for the PyTorch framework, but this can not be used within the context of TensorFlow. Since the deep learning efforts and tools developed so far in KM3NeT rely exclusively on TensorFlow, an efficient implementation of the Edge Convolutional block using the TensorFlow backend was designed for this work[1]. and is described in the following.

The k nearest neighbors algorithm

The calculation of the k nearest neighbors is done in parallel on a batch of events. Since each of the events has in general a varying number of hits, a knn-operation that works on a ragged tensor is required. Since no such operation exists in the TensorFlow framework, a custom kernel with CUDA [99] acceleration was developed. The number of operations performed in the k nearest neighbors search for a graph with N nodes is $\mathcal{O}(N^2)$, since we need to calculate for each node the distance to each other node. However, it is possible to reduce the required VRAM to a linear scaling of $\mathcal{O}(N \cdot k)$. For this, consider the knn calculation of a single node. A naive approach would be to calculate the distance between this node and all other N nodes in the graph first, and then pick the k smallest values. However, this would mean having N values in memory at the same time, and since this is done in parallel for all nodes in the graph, it could result in a

[1] https://github.com/StefReck/MEdgeConv



(a) Structure of an EdgeConv block.

(b) Architecture of ParticleNet.

Figure 6.4.: The structure of ParticleNet and the EdgeConv block as published by Qu and Gouskos [97], where it was used for a classification task. The original model shown on the right was slightly changed for this work, by removing the dropout in the first layer after the pooling, and by replacing the last layer depending on the specific reconstruction task at hand.

VRAM requirement of $\mathcal{O}(N^2)$. This can become prohibitively large for bigger graphs. A more efficient approach is to keep only the smallest k distances encountered for each node in memory. For each new node, we can then compare the new distance to the k saved distances, and if it is smaller then any of them, replace the largest saved distance with the new one. This means we will never have more then $\mathcal{O}(N \cdot k)$ values in memory, which allows for using much larger graphs as an input.

This is especially useful for applications in KM3NeT. While most events tend to have a comparable number of hits, we also occasionally encounter events with a very large number of hits. But since these unusually large events happen increasingly rarer the more hits they contain, they will only have a reduced impact on the overall running time.

For example, consider a reconstruction of 10,000 events, with the average event taking a time t to process. If one of them happens to be large and consequently take 100 times longer to process than the average event, the total elapsed time will increase to 10,099 t, an increase of merely one percent. A $\mathcal{O}(N^2)$ scaling of the memory usage on the other hand is more problematic, since exceeding the available VRAM at any point will lead to a crash, independent of whether such events are rare or not.

The knn calculation can be efficiently parallelized: Each node in a graph and each graph in a batch can be treated independently of each other. Therefore, a large speed boost can be acquired by writing and compiling the code using CUDA, similar to many other operations in TensorFlow and deep learning in general. The compiled CUDA kernel is then linked with a corresponding CPU implementation of the same algorithm and wrapped in a Python TensorFlow operation, which can be used in custom keras layers.

Edge Convolution

After defining the start and end points of the edges using the k nearest neighbors, we can then proceed to calculate the features of these edges. Following the strategy in ParticleNet, they are derived from the features of the nodes (i.e. the coordinates input). Given two nodes $\vec{n}_i$, $\vec{n}_j$ with F features each, the features of the edge $\vec{e}_{i,j}$ going from node i to node j are defined as:

$$\vec{e}_{i,j} = (\vec{n}_i, \vec{n}_j - \vec{n}_i) \tag{6.10}$$

This is a vector of length $2 \cdot F$, which consist of the features of the node itself, and the difference to the neighboring node. To perform the convolution, we use a multi-layer perceptron consisting of three dense layers with a batch normalization layer in each of them. This so called kernel network takes the $2 \cdot F$ features of an edge $\vec{e}_{i,j}$, and produces an update vector $\vec{u}_{i,j}$. The length of the vectors $\vec{u}_{i,j}$ is given by the number of nodes in the last of the three dense layers. Then, the update vectors of the k nearest neighbors of each node are averaged to retrieve the new node features $\vec{n}_i^*$:

$$\vec{n}_i^* = \frac{1}{k} \sum_{j=1}^{k} \vec{u}_{i,j}. \tag{6.11}$$

Here it is important to note that the same kernel network is used for every single edge in the graph. In other words, the kernel network is slid over the edges of the graph, similar to how the kernel of an image convolutional layer is slid over the pixels.

In the spirit of popular image recognition networks like ResNet [100], ParticleNet also makes use of shortcut connections between the input and output of the EdgeConv block. For this, the updated node features $\vec{n}_i^*$ from Equation 6.11 are added to the original nodes features $\vec{n}_i$ that were used as the input to the EdgeConv block. However, the length of the new nodes $\vec{n}_i^*$ is not generally the same as the length of the old nodes $\vec{n}_i$, since the length of the vector that the kernel network produces is a hyperparameter and can therefore be of any size. So to make it possible to add them together, a single dense layer and batch normalization is used on $\vec{n}_i$, whose number of units is the same as in the last dense layer of the kernel network, thus producing a vector with the same length as $\vec{n}_i^*$. The result can then be added element-wise to $\vec{n}_i^*$.

All of the operations and layers described above are grouped into a higher level unit called EdgeConv block. In summary, this more abstract object has two inputs: the coordinates and the features, which are ragged tensors of shape $(N, ?, C)$ and $(N, ?, F)$. Here, C and F are the number of coordinate and nodes features. The output is again a ragged tensor of shape $(N, ?, F^*)$, that contains the new node features. F^* is the number of features of the new nodes, which is given by the number of units in the last layer of the kernel network.

Since the inputs and output of the EdgeConv block all are three-dimensional ragged tensors, we can stack these blocks by using the output of one block as the input of the next. We can even use the output for both of the inputs - the coordinates and the features. The adjacency of the graph is constructed using the coordinates input at the start of every EdgeConv block. This means that the connections in the graph are recalculated based on the output of the previous block, which of course depends on the free parameters of the network. This is a called a dynamic graph, and it is a powerful property of the Edge Convolution. Essentially, it allows the network to redefine adjacency in the graph, for example to group nodes together that share properties useful for the task at hand.

ParticleNet architecture

Figure 6.4b shows the architecture of the entire ParticleNet model. It consist of three EdgeConv blocks, with a dynamic update of the graph between them. The adjacency is constructed by finding the $k = 16$ nearest neighbors of each node. The units in the dense layers of the kernel network double in each layer, from (64, 64, 64) in the first layer to (256, 256, 256) in the third.

The feature input in the first block is seven dimensional: For each hit, it gets the x-, y- and z-position, the time t with respect to the first triggered hit of each event, as well as the pointing direction of the PMT as an euclidean vector (dx, dy, dz). The coordinate input is supplied with x, y, z and c · t (i.e. the time multiplied by the speed of light in water). The difference of these coordinates between nodes is used for the calculation of the nearest neighbors. The second and third block get the output of the previous block in both inputs.

After the third EdgeConv block, a global average pooling is performed. It averages the nodes in the graph feature-wise, i.e. it takes a set of node features of shape $(N, ?, F)$, averages the nodes along the ragged axis, and returns a regular tensor of shape (N, F). After that, two dense layers are added, which produce the final output of the network.

This architecture is reminiscent of popular image recognition networks like VGG [71]. These are built by stacking blocks of image convolutional layers with batch normalizations, while the number of filters per layer is doubled after each block. In the end, they often perform a filter-wise global pooling as well, followed by some dense layers.

A main difference between these architectures is the presence of intermediate pooling layers between the image convolutional blocks, which do not exists in ParticleNet. Apart from the fact that pooling operations are less straight forward to define for graphs than for highly regular images,Wang et al. [98] argue that the dynamic graph updates extend the receptive field of the Edge Convolution. In that sense, they perform part of the role of pooling layers in image convolutional networks. However, a reduction of the number of nodes using some sort of graph pooling could still be beneficial, as it could reduce the computational burden and consequently shorten the runtime or allow for handling larger graphs.

6.5. Advantages and limitations compared to convolutional networks

Several advantages of graph convolutional networks compared to the previously published image based methods in KM3NeT ([80], [84]) are apparent when comparing the way that the input information is encoded. Images of events are generated by defining a detector-specific grid in space and time, and then summing up the number of hits in each bin. The exact position within a bin of the DOMs in the water, which can also change over time due to the sea current, is therefore lost in this step.

The grid also limits the time resolution to the size of a time bin. While it is possible in principle to increase the number of time bins to any desired time resolution, this would increase the size of the images and consequently the training and reconstruction time. It would also further raise the sparsity of the signal, and thus potentially require the use of more convolution layers to increase the receptive field.

Finally, convolutional layers in TensorFlow can only convolve over three or less dimensions. Since our data has at least six dimensions (x, y, z, t, PMT direction), we can not convolve a kernel over all of them simultaneously and have to resort to different methods like using the color channel of convolutions or building models with multiple towers.

The graph network approach discussed in this work elegantly solves all of these issues. Since the information of each hit is represented as float numbers in the node features, it is not required to design a grid for each specific detector layout, and none of the hit information is lost in the encoding step. This might be especially advantageous considering the positions of DOMs in the water can change over time due to sea current. However, dedicated simulations including a shift in the position might be necessary to fully understand the implications of this effect [101]. Furthermore, the graph convolutions are by design n-dimensional and well suited for point clouds, which allows us to treat our sparse high-dimensional data efficiently.

The fixed grid used in image convolutional networks also has advantages though: The resulting images always have the same size independent of the number of hits in an event. So even very large events with thousands of hits do not lead to slower or more memory intensive calculations. In graph networks this is not the case. As established in section 6.4, the memory requirement is expected to be $\mathcal{O}(N)$ of the number of hits N in an event, and the k nearest neighbor algorithm requires $\mathcal{O}(N^2)$ operations.

However, first investigations for simulations with ARCA115 suggest that even events with an average of many thousand hits can still be processed using current GPUs, so this limitation might not be too troublesome for the use in KM3NeT. Additionally, technological advancements of the employed hardware in the coming years, parallelizing networks among multiple GPUs, or reducing the number of hits per graph via some sort of preprocessing might be viable options to reduce the memory burden.

Since the graphs are constructed using exclusively the hits, the information about which PMTs did *not* measure any hits is not directly provided on an event-by-event basis. It is likely though that the network will learn the number of PMTs over the course of the millions of events it gets presented with during the training, which all share this property. But the exact position of where each DOM is might still be helpful for the convergence or performance, especially when used in conjunction with the positioning system that keeps track of the movement of DOMs in the water.

Chapter 7.

Muon bundle reconstruction with Graph Neural Networks

Even though KM3NeT/ORCA is primarily a neutrino detector, most of the triggered events contain an atmospheric muon, or a bundle of muons traversing the detector at the same time. While these events usually pose as background for neutrino measurements, they are interesting subjects of research as well, since they can be used to indirectly investigate cosmic rays.

Various observables can be used to characterize such a bundle of muons: The incident direction and number of muons contained within the bundle (also called the muon multiplicity) as well as the energy and the lateral position of each muon. The classical track reconstruction algorithm of KM3NeT as described in section 4.4 can be used to estimate the direction and energy of atmospheric muons, but since it was developed for events with a single track, it turns out to be suboptimal in its current state for the case of multiple muons.

This chapter describes a deep learning based approach of reconstructing the direction, multiplicity and lateral diameter of muon bundles, which aims to improve the existing reconstruction, and extract additional information beyond what is currently available. It also details the first ever application of deep learning algorithms in KM3NeT on measured data [86][102]. Since the detector is currently under construction, data from a small version of it is used, which consists of four lines out of the 115 lines that will be in place in the future. This detector setup is referred to as ORCA4 in the following. Due to the high detection rate of muons, precise statistical comparisons of the simulations and measured data are feasible, even with relatively short windows of observation. This can greatly improve the understanding of the algorithm, and is a crucial step in building up trust in the powerful method of deep learning.

7.1. Model design and training setup

The architectures of the models used for the different reconstruction tasks in the following sections are in large parts identical, as they follow the ParticleNet structure introduced in section 6.4. The inputs are the x-, y-, and z-coordinates and the pointing direction of the PMT that recorded each hit as well as its time. The time is given relative to the first triggered hit of the event: For example, a hit time of 50 nanoseconds means that the hit was recorded 50 nanoseconds after the time of the first triggered hit. The pointing direction is encoded as the three components of a unit vector, so in total, each hit has seven features in the input layer.

For technical reasons, only up to 2000 hits are supplied to the network for each event. If there are more, hits are randomly removed until the threshold is reached. In order to reduce the number of hits, a time cut is applied before this: All hits that are outside of a time window of -250 ns to +1000 ns of the first triggered hit are removed. Using the simulations, it can bee seen that the vast majority of signal hits coming from muons are inside of this interval.

The coordinate input from which the edges are constructed are x, y, z and $t \cdot c$ of each hit for the first EdgeConv block, and the output of the previous block for the second and third EdgeConv block. After the global pooling, a dense layer with 256 neurons is added before the final output layer. This final layer is the only part of the architecture that is different for each reconstruction

Table 7.1.: Properties of the datasets used in this chapter, all of which are for the four-line setup of the KM3NeT/ORCA detector (ORCA4). The MUPAGE dataset with the KM3 light simulator (see section 4.2) is only used for training the multiplicity network of section 7.4. Everywhere else, the MUPAGE dataset with the JSirene light simulator is used as simulations.

MUPAGE JSirene dataset	
data version	5.4
light simulation	JSirene
events train	20.0 million
events validation	1.1 million
events test	5.3 million

measured dataset	
data version	5.4
runs	40
livetime	9.5 days
time period	08/2019 - 01/2020
events	5.4 million

MUPAGE KM3 dataset	
data version	5.4
light simulation	KM3
events train	4.9 million
events validation	1.2 million

task. Otherwise, the configurations of the models are identical. The structure of the output layer for each reconstruction task is described in detail in each of the following sections. By training and comparing several models with different hyperparameters, the proposed architecture was found to be suited decently for all the given tasks.

All presented networks have about 300,000 free parameters in their three Edge Convolutional blocks. In addition, they have about 70,000 free parameters in the dense layers, though this does depend slightly on the reconstruction task at hand. The models are trained with a batchsize of 64 and the Adam optimizer with the default parameters. The properties of the datasets used in the following sections are listed in Table 7.1.

The labels for the training are derived from the properties of the muon events. For example, one interesting observable is the muon multiplicity, i.e. the number of atmospheric muons in an event. However, it might not be ideal for the reconstruction to simply count the total number of simulated muons in the simulation can, since not all of these muons necessarily produce measurable hits in the detector. Consequently, it would be impossible to directly reconstruct such a muon. Similarly, if a muon merely produces a single hit, it might not be distinguishable from the background.

For this reason, only *visible* muons, defined as producing ten McHits or more, are counted towards the multiplicity. A McHit is a hit in the simulations (also referred to as Monte Carlos) before the simulation of the detector response. Together with the simulated PMT efficiency of 50%, ten McHits result in an average of about five actual hits. The number of five hits was chosen somewhat arbitrarily, but there was little difference in the result when choosing a slightly higher or lower value. Whenever the number of muons is mentioned in the following sections, it always refers only to the visible muons, unless specified otherwise.

The preprocessing of the data is done using the OrcaSong framework (see subsection 5.7.1) in version 4.7.1. The results can be easily reproduced by using the configuration files listed in section A.1 for MUPAGE and measured data. All the models presented in the following sections were built and trained using OrcaNet v0.16.3 (see subsection 5.7.2) and MEdgeConv v2.1 (see section 6.4), together with the respective configuration files found in section A.2.

7.2. Reconstruction of the incident direction

For this task, the goal is to reconstruct the incident direction of an atmospheric muon or a bundle of muons. The case of just one muon in an event is the most common one, making up about 77% of the triggered simulated events, while events with higher multiplicities occur increasingly more rare.

In measured data, the multiple muons of a high multiplicity event do not generally all have the same incident direction, as they can receive different transversal momenta during their creation in the air shower, and can also be deflected along the way. However, this variation is typically small and is therefore not part of the Mupage simulation. Instead, all muons in Mupage arrive from the same direction, and consequently there is also only this one observable to reconstruct.

The classical reconstruction described in section 4.4 was primarily designed for single track events, as it makes use of a track hypothesis and PDFs developed from single muons. These events provide an excellent opportunity to evaluate the performance of the deep learning algorithm, and cross check results on measured data. Deep learning can be especially powerful for multi muon events. As these events are part of the training dataset, the network learns to properly reconstruct them, too.

7.2.1. Performance evaluation on simulations

The architecture of the network for the directional reconstruction is as described in section 7.1. The output of the network are the three components of a vector (`dir_x`, `dir_y`, `dir_z`), from which the direction can be inferred. As described in section 5.6, a normal distribution is assumed for each of these components in order to obtain a measure for the uncertainty of the reconstruction. Since this results in two values (μ and σ) per component, the network has six output neurons in total. The μ neurons have a linear activation, while the σ neurons have an exponential one, which ensures that their output is always positive. The loss is the negative log-likelihood of a unit vector pointing in the true direction given the prediction, calculated individually for each component and then added together:

$$\begin{aligned}\text{loss} = &- \log L(\text{true}_{\texttt{dir\_x}} | \mu_{\texttt{dir\_x}}, \sigma_{\texttt{dir\_x}}) \\ &- \log L(\text{true}_{\texttt{dir\_y}} | \mu_{\texttt{dir\_y}}, \sigma_{\texttt{dir\_y}}) \\ &- \log L(\text{true}_{\texttt{dir\_z}} | \mu_{\texttt{dir\_z}}, \sigma_{\texttt{dir\_z}})\end{aligned}$$

The training of the network was done on a GTX 1080ti GPU, and finished after about four days. As shown in Figure 7.1, the learning rate was kept constant at the default value 0.001 of the Adam optimizer until no improvement in the loss was observed. At this point, the learning rate was decayed by a factor of 10 for an epoch, and again for the final epoch. No big change in the loss was observed after the second decay, indicating that the network has converged sufficiently. After every quarter of an epoch, the loss was checked on a validation dataset that was not used otherwise during the training. All following plots in this section are generated using a third test dataset, which was not used for the training or validation.

In Figure 7.2, the zenith and azimuth angle of the reconstructed vector is compared to the true values. For now, only the central values μ of the distributions are investigated, while the uncertainties σ will be explored separately in subsection 7.2.2. The reconstructed values are scattered tightly around the true values. For the zenith, a bias in the positive direction (i.e. towards cosine zenith of 1, muons traveling straight down) can be observed. This is likely caused by the shape of the true distribution on the training data: Less and less atmospheric muons arrive from directions closer to the horizon (cosine zenith = 0), where they have to travel longer distances through the atmosphere and the sea water and therefore lose more energy before arriving at the detector.

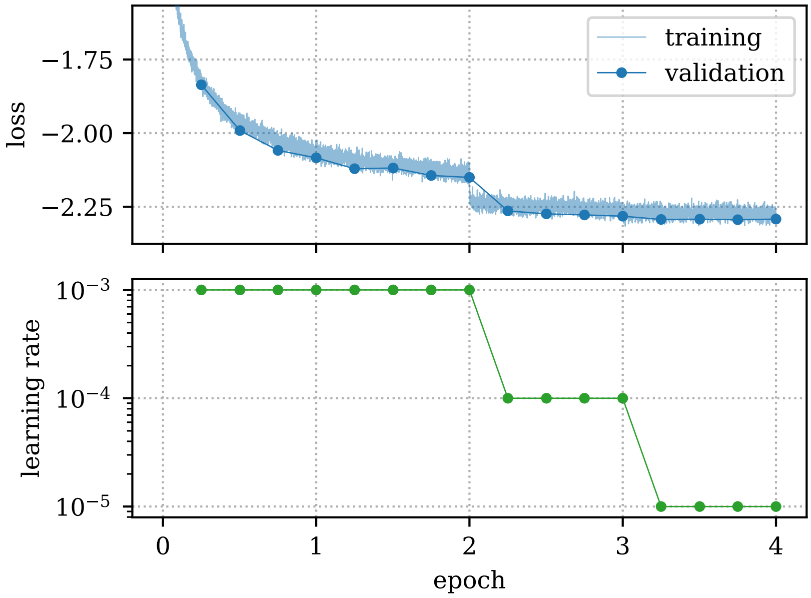


Figure 7.1.: Training history of the neural network used for the directional reconstruction. Shown is the loss (top) and the learning rate (bottom) over the epoch, i.e. the number of iterations over the training dataset.

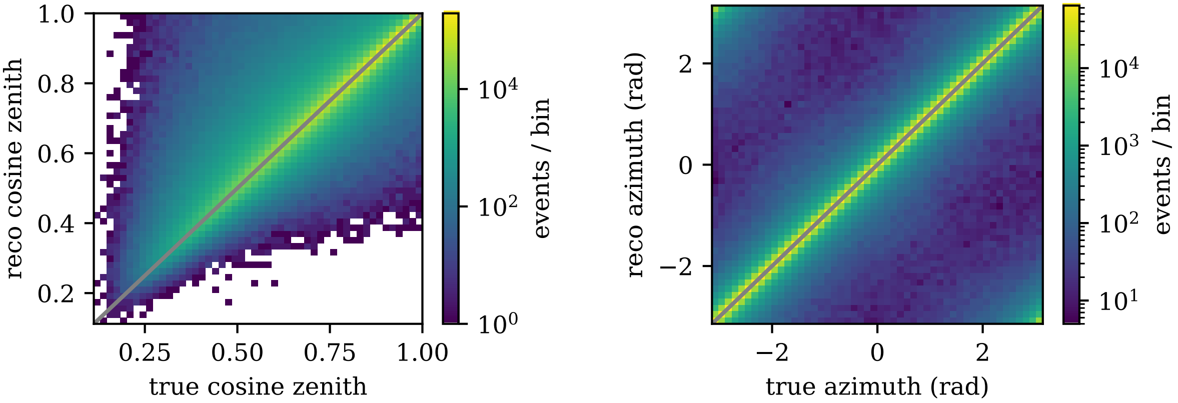


Figure 7.2.: 2D histograms of the directional reconstruction of muon bundles with the neural network. Shown is the reconstruction versus the truth, for the cosine of the zenith angle (left) and the azimuth (right). The straight grey lines through the origin show the ideal case where the reconstruction is equal to the truth.

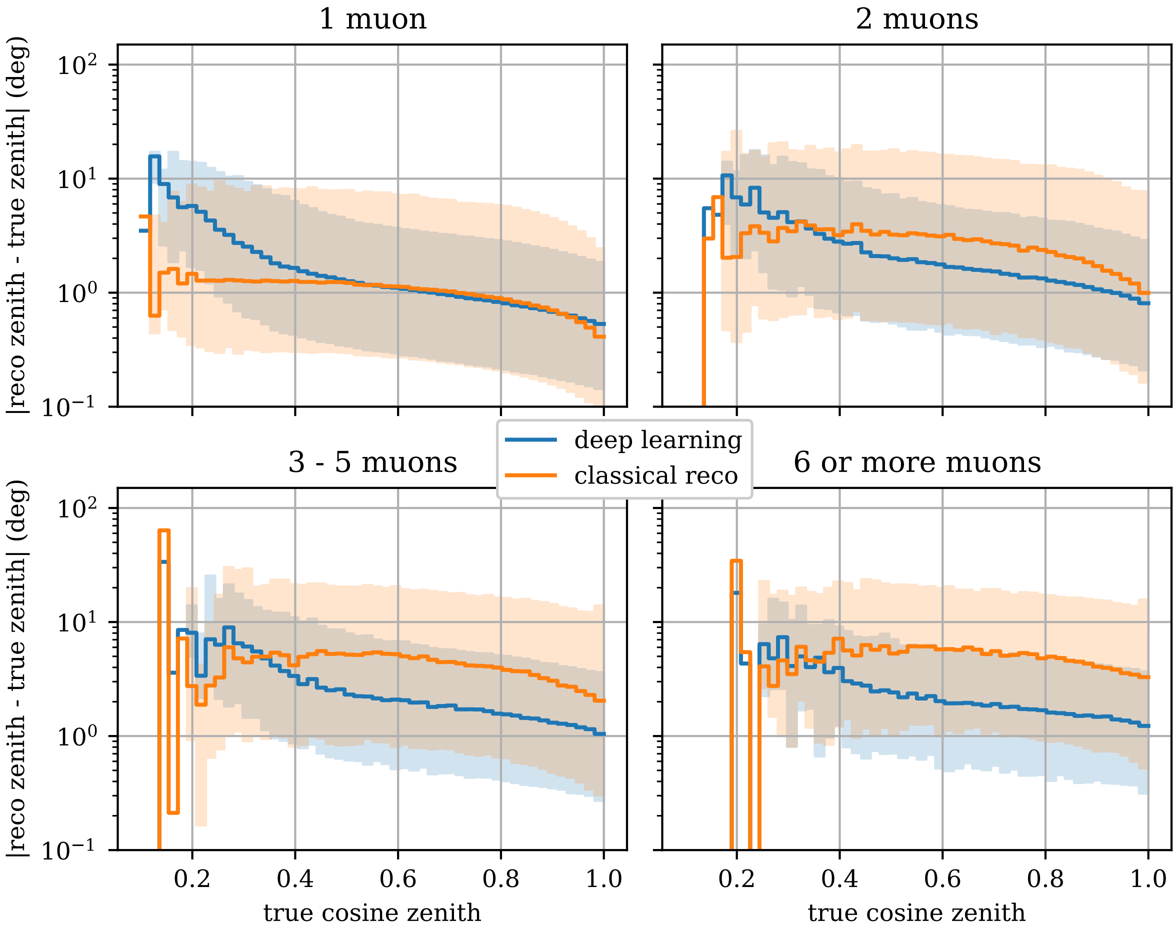


Figure 7.3.: Error in the zenith reconstruction for deep learning (blue) and the classical reconstruction (orange) over the true cosine zenith angle. Each plot shows only events with a specific number of simulated muons, as indicated in the title. The solid line is the median of the distribution for each bin, while the colored regions give the 68% interval around the median. Only the 93% of events that were reconstructed as down-going by the classical reconstruction were used in the plots.

The network takes on the information about the shape of the true distribution during the training, and incorporates it in its prediction via the observed bias. Since such a distribution of arrival directions is also observed in measured data, it can be considered as a legitimate way to improve the performance. However, it also adds a dependency on the shape of this reduction of the flux. If a bias-free reconstruction is desired, the true distribution should be made flat, either by producing a dedicated simulation, or by weighting the samples accordingly during the training.

As mentioned above, there is also a classical approach for reconstructing the direction of a single muon track in KM3NeT. In Figure 7.3, the error in the zenith angle reconstruction is compared between the deep learning and the classical approach. Since these plots show the true zenith angle on the x-axis, the bias of the deep learning approach is clearly visible as an increase for the more horizontal events below a true cosine zenith of about 0.4. The classical approach is unbiased and therefore flat in this region.

In order to make this comparison a bit fairer, any events that are reconstructed as up-going by the classical approach were removed here. The network never reconstructs events as up-going as such events were not part of the training dataset, so it would have an advantage for them.

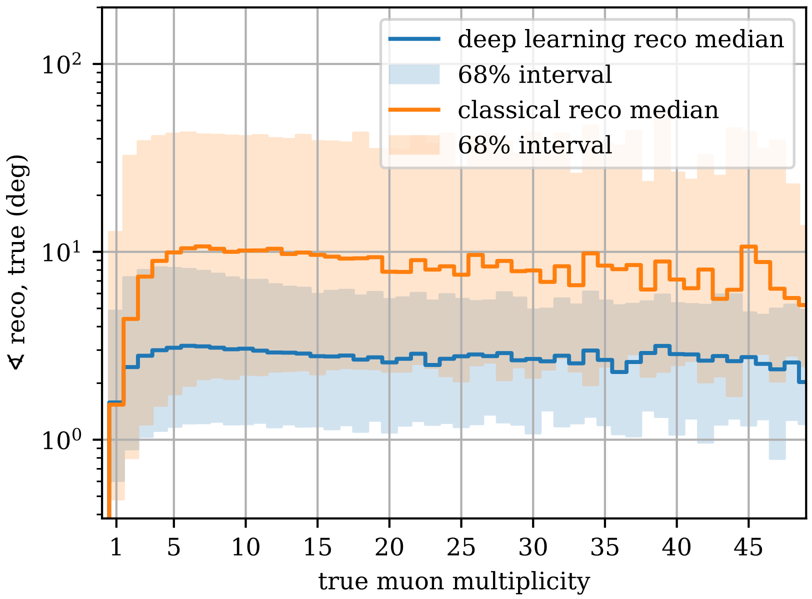


Figure 7.4.: Angle between the reconstructed direction and the true direction as a function of the true muon multiplicity for deep learning (blue) and the classical reconstruction (orange). The solid line is the median of the distribution for each bin, while the colored regions give the 68% interval around the median. Only the 93% of events that were reconstructed as down-going by the classical reconstruction were used in the plot.

As seen in Figure 7.3, the deep learning approach achieves a similar performance in the zenith reconstruction for single muons, but greatly outperforms the classical approach for higher multiplicities.

This is shown in greater detail in Figure 7.4. The precision of the directional reconstruction is best for single muons for both approaches, and quickly climbs up to a plateau for events with about three or more muons. This plateau is at a median error of about three degrees for deep learning, and eight to ten degrees for the classical approach.

It is not too surprising that the classical approach performs suboptimal for higher multiplicities, as it was designed for single muon tracks produced by neutrino interactions. Its performance for atmospheric muons could potentially be improved by re-parametrizing the PDF for the multi muon case. Additionally, only downward directions could be sampled during the prefit stage, which would prevent upward going solutions in the first place, as they are not required for reconstructing atmospheric muons. For deep learning, no such manual adjustments are necessary: The optimization happens autonomously as the training dataset consists of downward going muons and bundles.

7.2.2. Uncertainty estimation

Since the outputs of the network were parametrized to be normal distributions for each component of a three dimensional vector, they can be used not only as an estimator for the direction, but also for the quality of the reconstruction on an event-by-event basis. This quality is given by the widths σ of the normal distributions. Cuts on this quantity allow to discard events for which the network was not able to reconstruct the incident direction well. This is illustrated in Figure 7.5 for the zenith angle by selecting events of increasingly higher quality.

It is not inherently clear whether the choice of independent normal distributions for the outputs of the network are a reasonable choice given the reconstruction task at hand. As mentioned

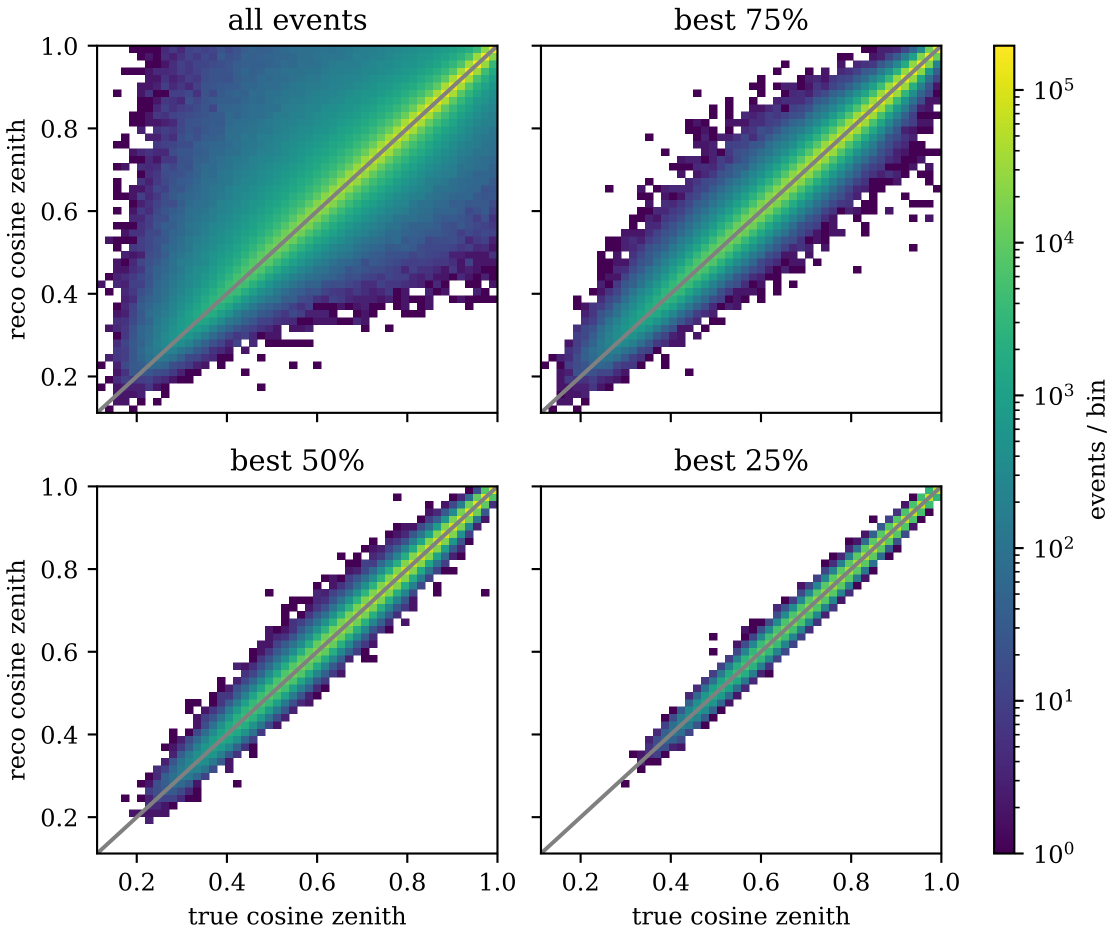


Figure 7.5.: Histograms of the reconstructed cosine zenith angle over the true cosine zenith angle for different cuts on the reconstruction quality, indicated by the title of each plot. For example, the top right plot shows only the events that are among the best 75% of events in terms of the reconstructed sigma zenith.

in section 5.6, this choice can be highly problem specific, and is de facto an additional hyperparameter of the architecture.

A good way to investigate this is to compare the pull distribution of the reconstruction to the ideally expected standard normal distribution. The pull is defined for each event as the difference between the true value $\mathrm{y}_{\mathrm{true}}$ and the reconstructed μ_{reco}, divided by the reconstructed uncertainty σ_{reco}:

$$\mathrm{pull} = \frac{\mathrm{y}_{\mathrm{true}} - \mu_{\mathrm{reco}}}{\sigma_{\mathrm{reco}}}.$$

In other words, the pull describes how many sigmas the true value is away from the reconstruction. If the error estimation works correctly, 68% of the true values should be within the 1σ interval around the reconstruction, 95% within the 2σ interval, and so on. In short: the pull should follow a standard normal distribution. The pull distributions for the components of the reconstructed direction are shown in Figure 7.6.

In the interval $\pm 2\sigma$, the pulls closely follow the normal distribution. For the few events in higher intervals, non-Gaussian tails are observable. It is intuitively clear that a normal distribution can not be a perfect posterior distribution for the output of the network, because the components of

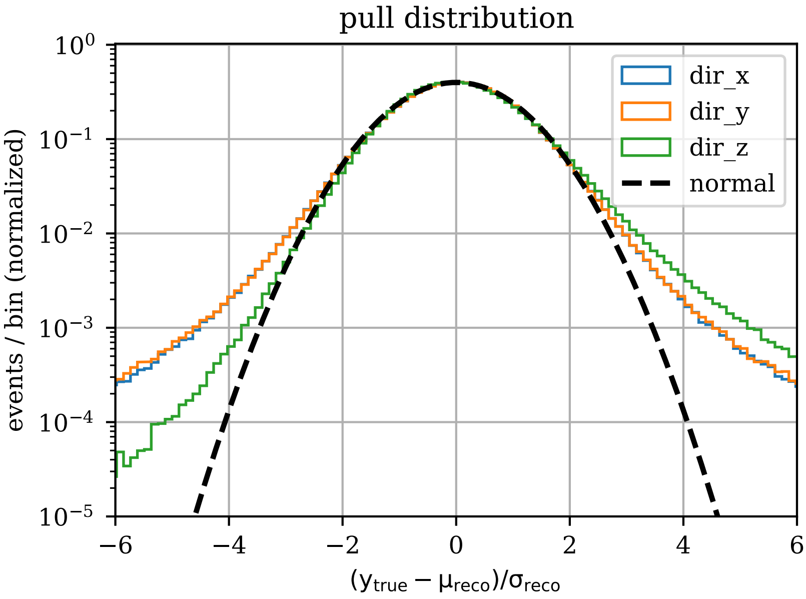


Figure 7.6.: Pull distributions of the directional reconstruction. The solid lines are the pulls for the components of the reconstructed vector, and the dashed line is the ideally expected case of a standard normal distribution.

a unit vector are restricted to the interval $[-1, 1]$, whereas a normal distribution is unbounded. However, for the 95% of events in the two sigma interval, the estimation of the error is accurate.

This could potentially be improved by adjusting the output distribution, for example by including correlation terms between the components, or choosing a distribution that is directly defined on the surface of a sphere. An interesting candidate for this could be the von-Mises-Fisher distribution for a spherical surface, which can be compared to a two dimensional normal distribution that respects the periodicity of the angles. However, no satisfying results using this distribution could be achieved within the scope of this work.

While the pulls are symmetrical for `dir_x` and `dir_y`, the distribution of `dir_z` is asymmetrically slanted towards the positive direction. This is caused by the fact that while the true `dir_z` is restricted from above to always be ≤ 1 (since the truth is defined as a unit vector), it only has a soft restriction from below: The flux of atmospheric muons gradually decreases when looking closer to the horizon, reaching practically zero for `dir_z` < 0.

If the network would reconstruct `dir_z` $= 1$, it would essentially put half of the normal distribution outside of the interval in which the true value can occur. To maximize the likelihood, it will therefore *shy away* from the border `dir_z` $= 1$ and place the normal distribution at a slightly smaller value, depending on σ. This means that the true values will be systematically above the μ in these cases, leading to the positively slanted pull distribution. As this affects the `dir_x` and `dir_y` components for both their borders -1 and $+1$ in opposite ways but with the same frequency, their pulls are symmetric.

7.2.3. Data-Monte Carlo comparison

Deep learning is a relatively new approach to reconstruction in high energy physics. Previous studies in KM3NeT have only looked at the performance on simulations [84], but have not investigated the application on measured data. Since the neural networks are sometimes considered to be a *black box*, it is not a priori clear how potential differences present in measured data as compared to the simulations affect their behavior.

It is therefore of critical importance to thoroughly investigate how networks behave when using

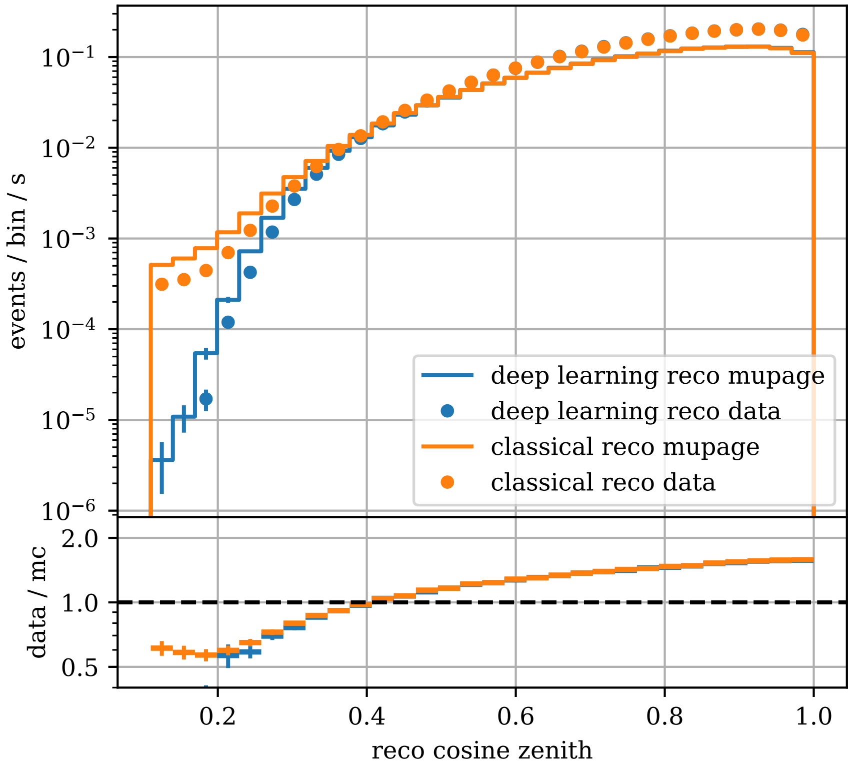


Figure 7.7.: Comparison of the reconstructed cosine zenith angle on simulations (solid lines) and data (dots) for deep learning (blue) and the classical reconstruction (orange). The bottom plot shows the ratio between data and the simulation for each bin. As described in the text, only events with a high quality of the classical reconstruction are used for this plot.

them on measured data. Since atmospheric muons are measured at high rates, large statistics of events are available even within relatively short time periods and a detector in an early stage of construction.

A good way to check if deep learning works as expected on data is to compare it to the established classical reconstruction. For this, a strict cut is made on the classical reconstruction quality to ensure that only well understood events are selected, for which the PDF could be very accurately fit to a muon track. Only events with a likelihood per hit ratio of more than 2, at least 40 hits involved in the fit, and which are reconstructed as down-going are considered for this. The described cut keeps 27% of simulated events and 36% of data events, and also removes virtually all multi muon events. The distribution of the reconstructed cosine zenith angle of these selected events is shown in Figure 7.7 for simulations and data.

The rates in the top part of the plot are calculated by dividing the number of events in each bin by the total livetime of the simulation or the measured data. The error bars show the standard deviation of the Poisson distribution for each bin, i.e. the square root of the number of events in a bin divided by the livetime. The bottom part shows the ratio $r_\text{data}/r_\text{mc}$ between the rates on data and simulations (also referred to as Monte Carlos, or MC). The uncertainty of this ratio is calculated from the errors of the rates via propagation of error:

$$\sigma_\text{data/mc} = \frac{r_\text{data}}{r_\text{mc}} \sqrt{\left(\frac{\sigma_\text{data}}{r_\text{data}}\right)^2 + \left(\frac{\sigma_\text{mc}}{r_\text{mc}}\right)^2} \tag{7.1}$$

Both algorithms show excellent agreement with each other for cosine zenith angles above 0.4 (zenith angles between zero and 66 degrees from straight down-going). Below that value, the absolute rates differ increasingly, as the bias towards down-going events of deep learning becomes apparent compared to the unbiased classical reconstruction. However, the ratio between data and simulation agrees down to a cosine zenith of 0.2 (78 degrees), below which the statistics on data are too sparse.

It is also worth noting that the data-MC ratio is quite far from one overall. This is in part due to the aforementioned quality cut, as there is a significant data-MC disagreement of the likelihood per hit of the classical reconstruction. The reason for this disagreement is currently under investigation in KM3NeT. On the other hand, the parametrization of MUPAGE from which the arrival directions are drawn is also not fully accurate, and it is an ongoing effort in KM3NeT to improve the simulation by tuning the MUPAGE parameters [103].

On the left side of Figure 7.8, a similar data-MC comparison as before is shown, but this time including all events. It is noticeable that there is a significant excess of events reconstructed as relatively straight down (cosine zenith > 0.9) by the neural network. This increase is not visible for the classical reconstruction, so these events are likely wrongly reconstructed by the deep learning algorithm. This highlights another very important use case of the uncertainty estimation of the network: Improving the data-MC agreement.

For this, consider the data-MC comparison of the reconstructed sigma of the zenith angle in Figure 7.9. While the agreement is good for low uncertainties $\sigma < 0.16$ rad, there is a large excess visible for the rarer, high uncertainty events. The 3% of MC events and 8% of data events above that threshold can be easily removed with a cut. The right side of Figure 7.8 shows that the previously observed excess in the zenith angle reco vanishes with this cut, resulting in good agreement of the data-MC ratio between deep learning and the classical approach. A slight disagreement is visible for cosine zenith below about 0.5, caused by the bias of the network compared to the bias free classical reco.

Unlike in Figure 7.7, where a strict cut on the classical fit quality was used, the absolute rates also differ slightly above a cosine zenith of 0.5. This means that for events where the classical approach has a lower quality, the network reconstructs events with a slightly different direction. Such events include for example multi muon events, for which it was shown in Figure 7.4 that deep learning produces a slightly different and more precise directional reconstruction. However, the fact that the ratio matches suggests that this difference in the reconstruction of the zenith is consistent on both data and simulations.

But why is the deep learning algorithm not capable of reconstructing the events that ended up with a high sigma more precisely? For this, it is useful to look at Z-T plots of events that got removed with the sigma cut (Figure 7.10). These plots visualize the data recorded in the events in question. and can help to understand the strategy of the network. The two Z-T plots in the first row show typical events that get cut. They have a very low number of hits, and there is no track clearly visible, suggesting that these are events that got randomly triggered by background noise (see section 4.2). As there is no muon to reconstruct, it is not surprising that the directional reconstruction does not produce a precise reconstruction. In fact, it is very useful that the network assigns these events a high uncertainty, even though such events are not part of the MUPAGE dataset used during the training. This suggests that a deep learning reconstruction together with a sigma cut can be robust against deviations between data and simulations.

This point is further strengthened by specifically looking at the rare high uncertainty events that also have a high number of hits, meaning that they are unlikely to be triggered by background noise. The second and third row of Figure 7.10 show typical events with a high uncertainty and with more than 200 hits, which make up only about 0.4% of all the events in the cut. The two Z-T plots in the second row exhibit atypical horizontal patterns, again without a clear track visible. These events are likely to be either microsecond afterpulses or sparking PMTs. Afterpulses can occur after a bright event goes through the detector, and cause PMTs to record additional hits

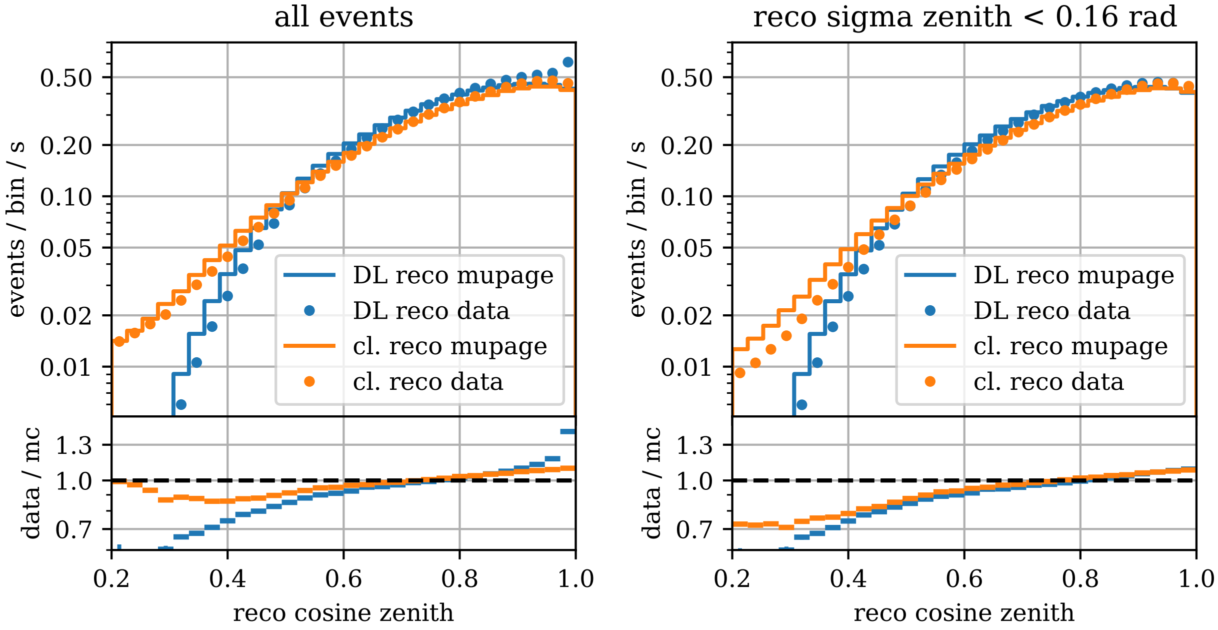


Figure 7.8.: Data-MC comparison of the reconstructed cosine zenith angle, similar to Figure 7.7. All events are used on the left side, and only events with a reconstructed sigma zenith below 0.16 rad are used on the right.

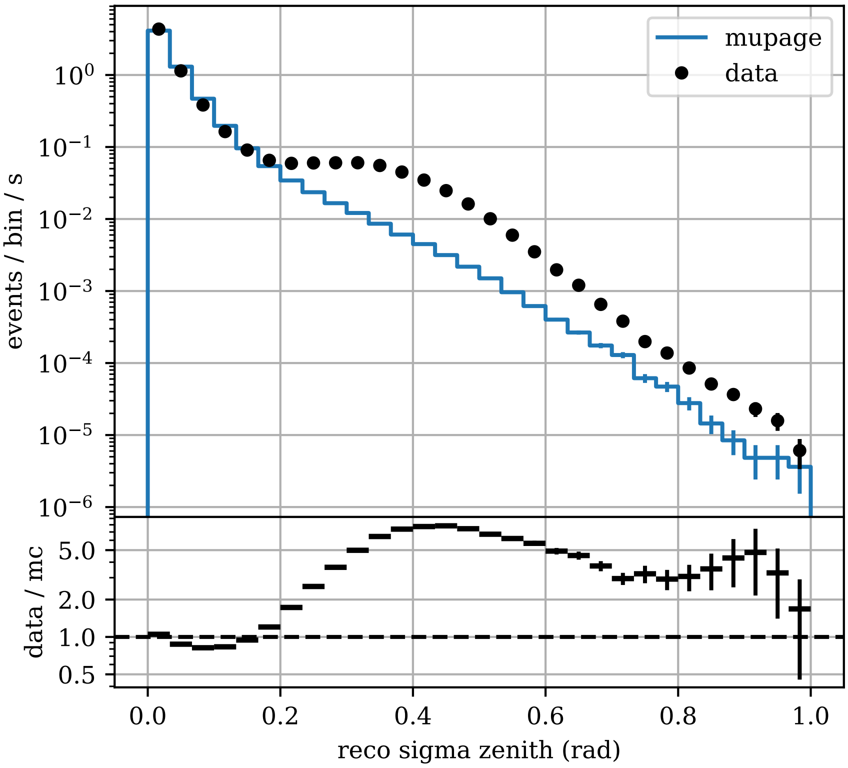


Figure 7.9.: Data-Monte Carlo comparison of the reconstructed sigma zenith of the neural network.

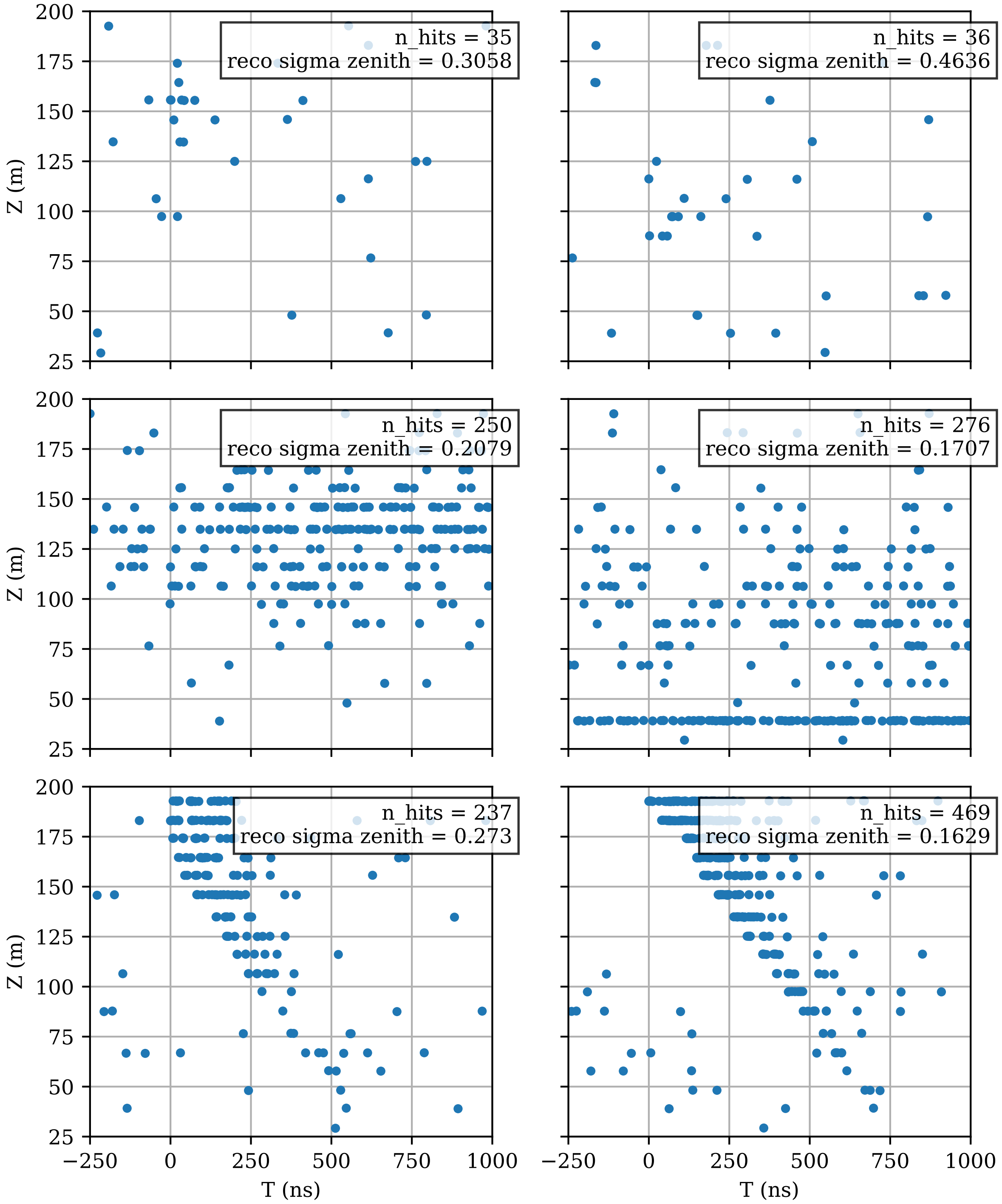


Figure 7.10.: Z-T plots of selected events that have a high uncertainty in the deep learning directional reconstruction (reconstructed sigma zenith above 0.16 rad). The plots show the z coordinate and the time at which each hit is measured in any of the four lines of the detector.

due to ionized residual gases in the photomultiplier tube long after the particle is gone [104]. This can then trigger another event, without a particle being present. Sparking PMTs repeatedly register hits with a high frequency for a short time, that did not originate from a passing particle either. Both of these effects are not part of the detector response simulation and therefore do not occur in the training dataset. Still, the network can properly assign these events a high uncertainty, which allows for easily removing them from a selection.

Some of the events with a high number of hits also look like a legitimate track, as shown in the third row of Figure 7.10. These muons might have hit the detector far from the center, as their track appears to barely reach down to the lowest story. As such events are more difficult to reconstruct, it is expected that their estimated uncertainty is also high.

7.3. Reconstruction of the bundle diameter

7.3.1. Definition

When multiple muons traverse the detector at the same time, one of the reconstructable observables of this bundle is the lateral spread of the individual muons. It arises mainly from the transversal momentum the muons receive when they are produced in the air shower. As they travel down to the depth of the detector, this momentum leads to a spatial separation of the muons. With its dense instrumentation, KM3NeT/ORCA is an especially promising candidate for analyzing this observable. This lateral spread is correlated to several properties of the primary particle of the extensive air shower. For example, heavier primaries can produce more detectable muons at the depth of KM3NeT further out from the primary axis [105].

In order to reconstruct the spatial configuration of a bundle, it is useful to first simplify the problem from the three dimensional trajectories down into a two dimensional plane. For this, we can choose a plane that is perpendicular to the prolonged primary trajectory of the air shower, and consider the points in which the muon trajectories hit the plane.

This plane can be referred to as the *shower plane* of the bundle. The individual muons can travel at slightly different angles in CORSIKA and in reality, so the plane should be chosen to be close to the instrumented volume. For this work, the point (0, 0) of the shower plane is set to be the center of the instrumented volume of the ORCA4 detector. The x- and y-axes of the plane are set to be in the xz- and yz-planes of the 3D coordinate system used in the simulations.

In the MUPAGE simulations used to train the network, no information about the primary particle is available. But since all muons travel in parallel there, the trajectory of any muon can be used to construct the shower plane.

The result of this transformation is a set of shower plane (x, y) coordinates for each muon in the bundle. But how can this spatial configuration be reconstructed efficiently using deep learning? A direct approach would be to reconstruct the coordinate of every single muon, and thus obtain the full information. However, this proved to be difficult due to the varying number of muons per event, which causes problems for a static model architecture with a fixed number of output neurons.

Instead, we can break down the spatial information even further to a single scalar number describing the diameter of the bundle. This makes a deep learning based reconstruction very straight forward, as it can then be done using the same approach as for example with the directional reconstruction. One option to define the bundle diameter is to look at the distance of the muon that is furthest from the primary. Since the MUPAGE simulation does not contain the primary trajectory, this approach is not possible.

Other options include replacing the primary position with the energy-weighted average position of all the muons in the bundle, or by calculating the smallest enclosing circle around all muon positions. But for this work, the diameter was defined as the maximum distance between any pair of visible muons in the bundle (see Figure 7.11). In practice, little difference was observed between the different definitions.

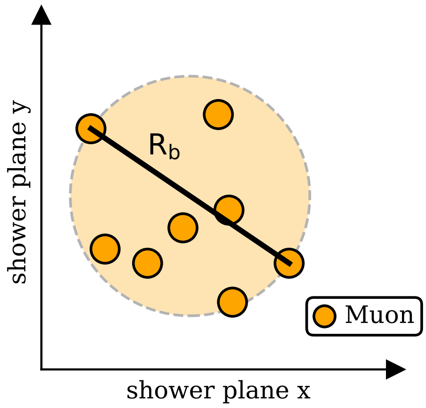


Figure 7.11.: Scheme showing the positions of the atmospheric muons of a bundle in the shower plane. The bundle diameter $\mathrm{R_b}$ is defined in this work as the maximum perpendicular distance between any two visible muons in the bundle.

7.3.2. Performance evaluation on simulations

The architecture of the network up to the last layer is as described in section 7.1. The output of the model consists of two neurons (μ, σ) parametrizing a normal distribution, with a linear and an exponential activation function, respectively. The label is chosen to be the $\log_{10}$ of the bundle diameter, which is functionally identical to letting the network parametrize a lognormal distribution for the diameter, but easier to execute in practice. The lognormal distribution was used, since the uncertainty is expected to be proportional to the central value. For example, it can be argued that a reconstruction uncertainty of ±1 m for an event with a 10 m diameter is similar to an uncertainty of ±10 m for a 100 m event. This is properly reflected in the lognormal distribution, in which both of these cases could have a σ of about 0.1. Similar to the approach used in subsection 7.2.1, the loss for the training is then defined as the negative log-likelihood of the true value given the distribution parametrized by the network.

As usual, the data is split threefold in a training, validation and a test dataset. Since the bundle diameter can only be defined for events with more than one muon, all single muon events are removed from the training and validation set. The training takes about 25 hours to complete on a GTX 1080ti GPU, spanning over close to nine epochs (see Figure 7.12). The learning rate was decayed from its original value of 0.001 by a factor of 0.9 after every quarter of an epoch. Some fluctuations in the loss curve can be observed in the first two epochs, but it stabilizes in the later stages of the training where the learning rate is lower.

In Figure 7.13, the reconstruction of the network is plotted over the true diameter for events from the test set with more than one muon. Since the distribution of the true diameter spans over multiple orders of magnitude, the number of events in the column of each true bin were normalized to the same value. When looking at the left plot in Figure 7.13, multiple components can be identified:

- Events close to the diagonal, for which the reconstruction works well.
- Events in a *box* below a true diameter of about $\log_{10}\mathrm{R_b} \approx 0.5$ ($\mathrm{R_b} \approx 3$ m). For these events, the reconstruction of the network is more or less a guess between 0.3 and 1 (2 - 10 meters). The muons in these events might be too close together to allow for any sort of separation. This is likely also the reason that the network never reconstructs events with a diameter below about 0.2 (1.6 meters).
- Events in a horizontal band located at a reconstructed diameter of $\log_{10}\mathrm{R_b} \approx 1$ ($\mathrm{R_b} \approx 10$ meters), with a slight shift upwards for higher true diameters. These are mostly events

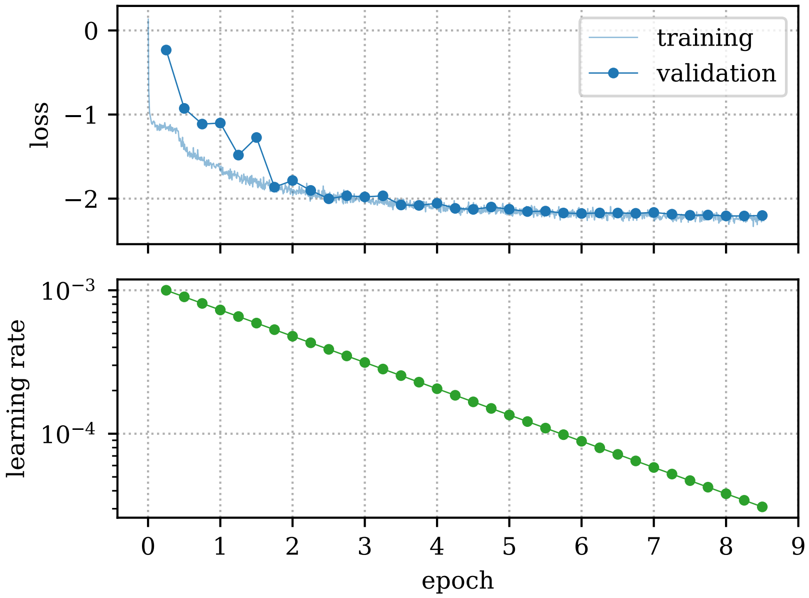


Figure 7.12.: Training history of the neural network used for the bundle diameter reconstruction. Shown is the loss (top) and the learning rate (bottom) over the epoch, i.e. the number of iterations over the training dataset.

where one or more muons are far from the detector: If only the 44% of bundles where all muons are within 50 meters to the center of the detector are plotted, this band disappears (right of Figure 7.13). For these events, the localization of the far away muon might be more difficult. When such a muon is missed, it can lead to a misreconstruction of the true diameter.

Since the cut used on the right hand side of Figure 7.13 uses MC truth information, it can not be used on data, even though it clearly improves the performance. Instead, we can make use of the reconstructed σ from the network, i.e. the width of the (log) normal distribution that is the output of the network. In Figure 7.14, the pull distribution of the reconstruction in log space is compared to a normal distribution, just like in subsection 7.2.2. Apart from non-Gaussian tails above $\approx 2.5\sigma$, the distributions match well. Thus, σ can be used as an event-by-event measure of the reconstruction quality.

In Figure 7.15, the same 2D plot as before is shown, except that only the best 44% of events with the smallest values of σ have been used. While Figure 7.15 and Figure 7.13 (right) both use 44% of the events, the cut based on the DL reco quality provides a better performance. This is quite natural, as not every event with a muon further than 50 meters from the detector is necessarily an event with a bad reconstruction. The reco quality allows for identifying well reconstructed events even for bundles with a far away muon. Additionally, most of the events in the *box* below a true diameter of ≈ 3 m were correctly assigned a high uncertainty as well, and were consequently also removed with the cut.

The median of the relative error of the diameter reconstruction for the different cuts is compared in Figure 7.16. The relative error is defined as the absolute difference between the true and the reconstructed value, divided by the reconstruction. When using the reco quality cut, the error has a plateau at about 15% between a true diameter of 5 and 40 meters, and ramps up below and above these values. It is worth noting that the diameter of the instrumented volume of the ORCA4 detector is roughly 40 meters, the distance up to which the plateau of the error extends. This could imply that future larger versions of the detector could measure much larger diameters with good precision, as the full KM3NeT/ORCA detector is expected to have a

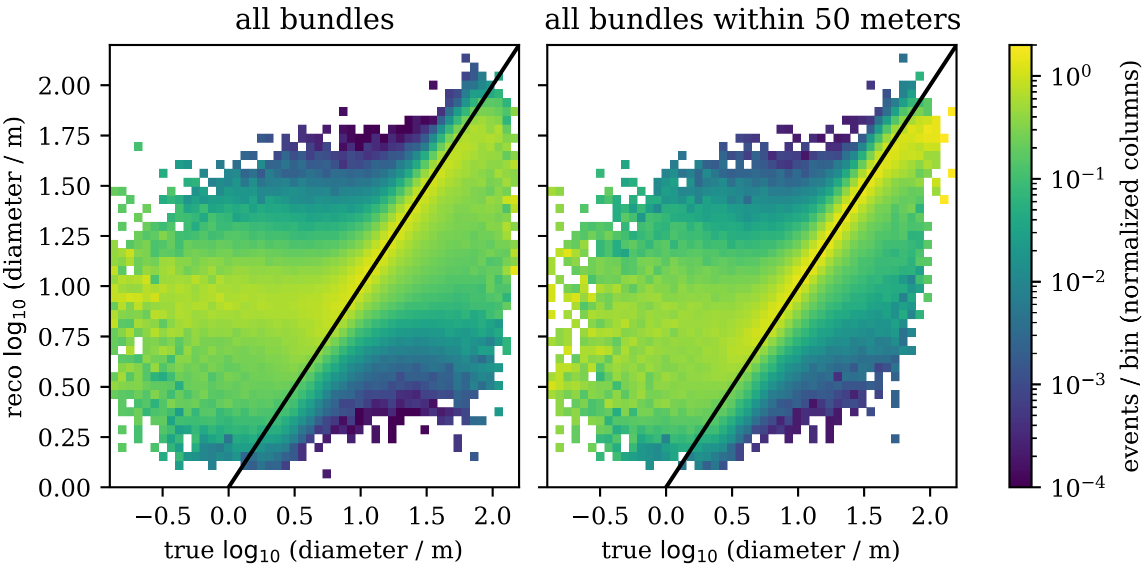


Figure 7.13.: 2D histograms of the bundle diameter reconstruction with the neural network. Shown is the reconstructed versus the true bundle diameter, for all bundles (left) and only the 44% of bundles in which the true trajectories of all muons come to within 50 meters of the center of the detector (right). The straight black lines through the origin show the ideal case where the reconstruction is equal to the truth. The bins in each column were normalized to have an area of one.

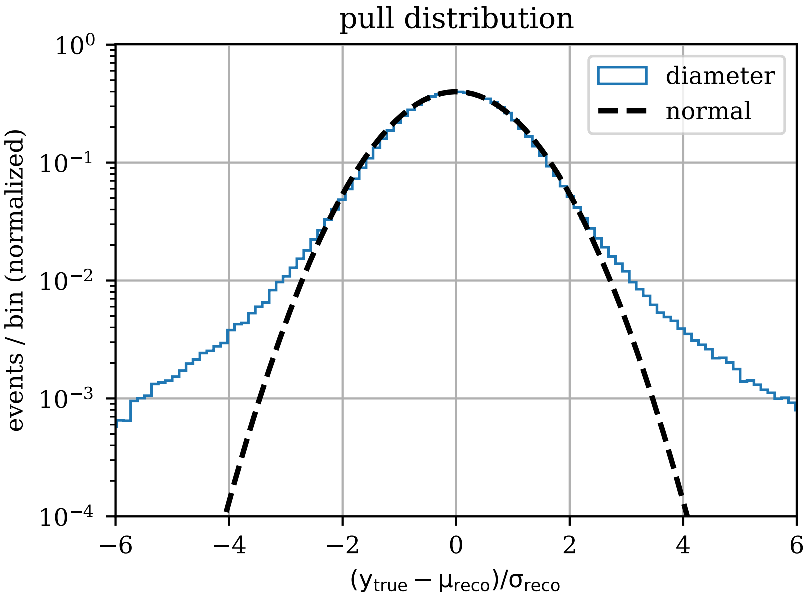


Figure 7.14.: Pull distribution of the bundle diameter reconstruction. The solid line is the pull for the reconstructed diameter, and the dashed line is the ideally expected case of a standard normal distribution.

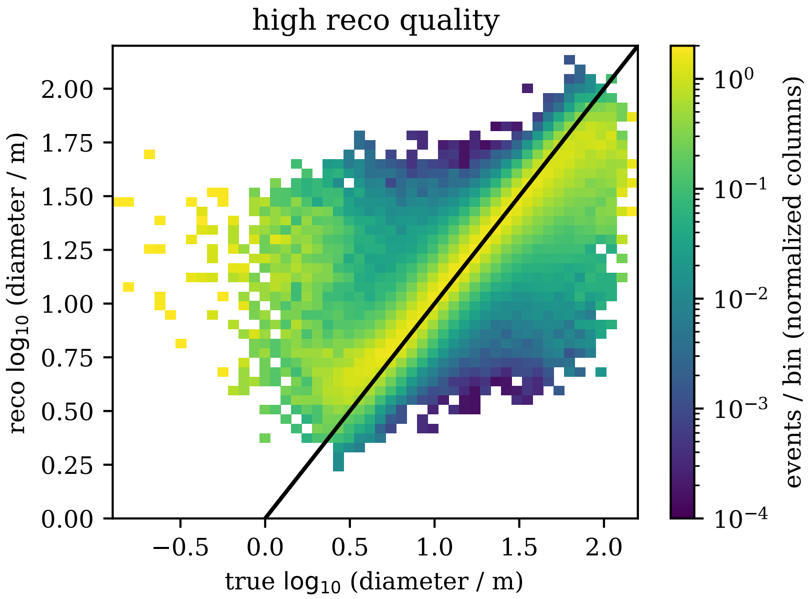


Figure 7.15.: 2D histogram of the bundle diameter reconstruction with the neural network, similar to Figure 7.13. Only the best 44% of muon bundles according to the reconstructed uncertainty of the diameter are used for this plot (events with $\sigma < 0.15$). This is the same fraction of remaining events as when using the distance cut in the right plot of Figure 7.13.

diameter of more than 200 meters.

Heavier and more energetic primaries can lead to higher bundle diameters. It is therefore important to check how the reconstruction of the diameter depends on the energy of the bundle. For example, it would be suboptimal if the diameter reco would be proportional to the energy and therefore not add additional information beyond that. The bundle energy is calculated here as the sum of the energies of each visible muon in the bundle at the height of the detector.

Figure 7.17 shows the performance of the reconstruction for different intervals of the bundle energy. The reconstruction works well for all intervals, so the neural network can indeed distinguish between events with a small and wide diameter, even if their energy is similar. The distribution of the true bundle energies, along with the intervals used in the aforementioned performance plot, are shown in Figure 7.18.

7.3.3. Biases in reconstructions

During the training, the network can implicitly learn the true distribution of the diameters, and tune the bias of its reconstruction in order to optimize its performance. This is similar to the situation of the zenith reconstruction, where the network picked up the fact that the flux of muons decreases towards the horizon. As can be seen in the following, the autonomously learned bias can be quite powerful, but it also bears the risk of model dependency: If the underlying true distribution changes, for example when applying the network to data, this can have implications for the accurateness of the reconstruction as well. Cross-checks using a different true distribution could be used to investigate this.

In Figure 7.19, a 2D histogram of the reconstruction is shown similar to the ones from before, except that now each row is normalized to the same area. So instead of showing what the distribution of reconstructed values are for a given true interval, it is now the opposite way around: Each row shows the distribution of true values for a given reconstructed value. Or, in other words, if we look at events that are reconstructed with a specific diameter, it shows what

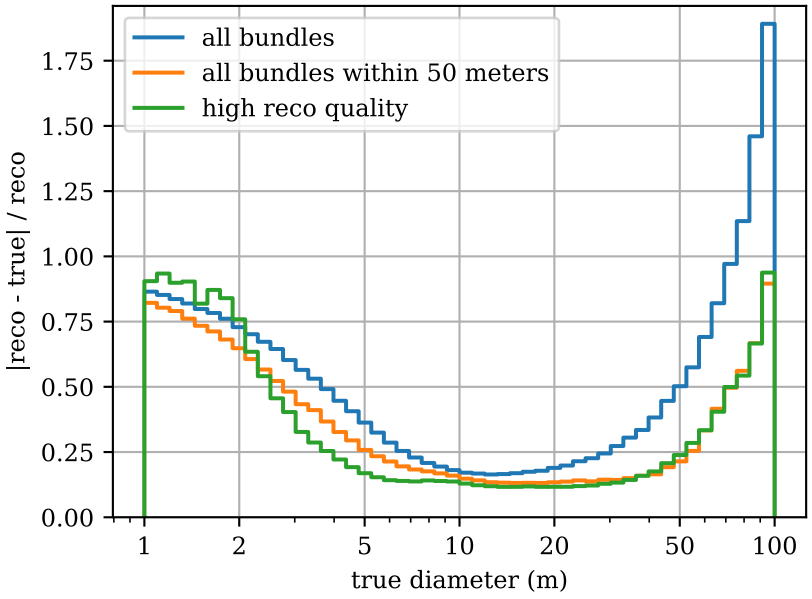


Figure 7.16.: The median relative error of the diameter reconstruction plotted over the true diameter. The orange and green curves are using the different cuts introduced in Figure 7.13 and Figure 7.15.

the true diameters of these events are. Even though no cuts were used in the plot, the events are very close to the diagonal across the board, from the lowest reconstructed diameters of 2 meters or so, up to 100 meters and beyond.

Components like the band of bad reconstructions at a reco diameter of 10 meters discussed for Figure 7.13 are not present in the row-normalized plot. This is because the highest flux of muons is located at a true diameter of about 10 meters (see Figure 7.20). Events with a higher or lower true diameter appear much more rarely. Thus, when such events get wrongly reconstructed as 10 meters, they only pose a small background to the much larger amount of correctly reconstructed true 10 meter events. In fact, this is arguably the reason for why this band of misreconstructions is even located at a reco diameter of 10 meters.

Similarly, if the network encounters one of the many difficult to reconstruct events with a very small or very large diameter, it can shift the reconstruction of these events upward or downward, closer to the maximum of the underlying true distribution. This way, the reconstruction is closer to the truth for a given reco value, even for events with an extreme diameter. On the flip side, this also means that the efficiency at which such events are reconstructed is relatively small, as many events are shifted away from their true value due to this bias. This can also be seen in Figure 7.20, where the number of events of the reco distribution is much lower than that of the truth for big and small diameters.

Considering the diameter of the instrumented volume of the ORCA4 detector is only about 40 meters, it is quite remarkable that the network can accurately reconstruct bundles with a diameter of up to 100 meters. This poses the question of what the approach of the network is for this task. While it is not possible to deduce the network's strategy analytically from its complex internal structure, we can still try to get some insight by looking at the events for which the reconstruction works well, and at those for which it does not.

In Figure 7.22, the positions of the individual muons in a bundle are plotted for several randomly selected events, together with a rough indicator of the instrumented volume of the ORCA4 detector. Figure 7.22a shows random events with a large reconstructed diameter. As was established in Figure 7.19, the true diameter is close to the reco on average. In Figure 7.22b, random misreconstructed events with a large true diameter are plotted, for which the network

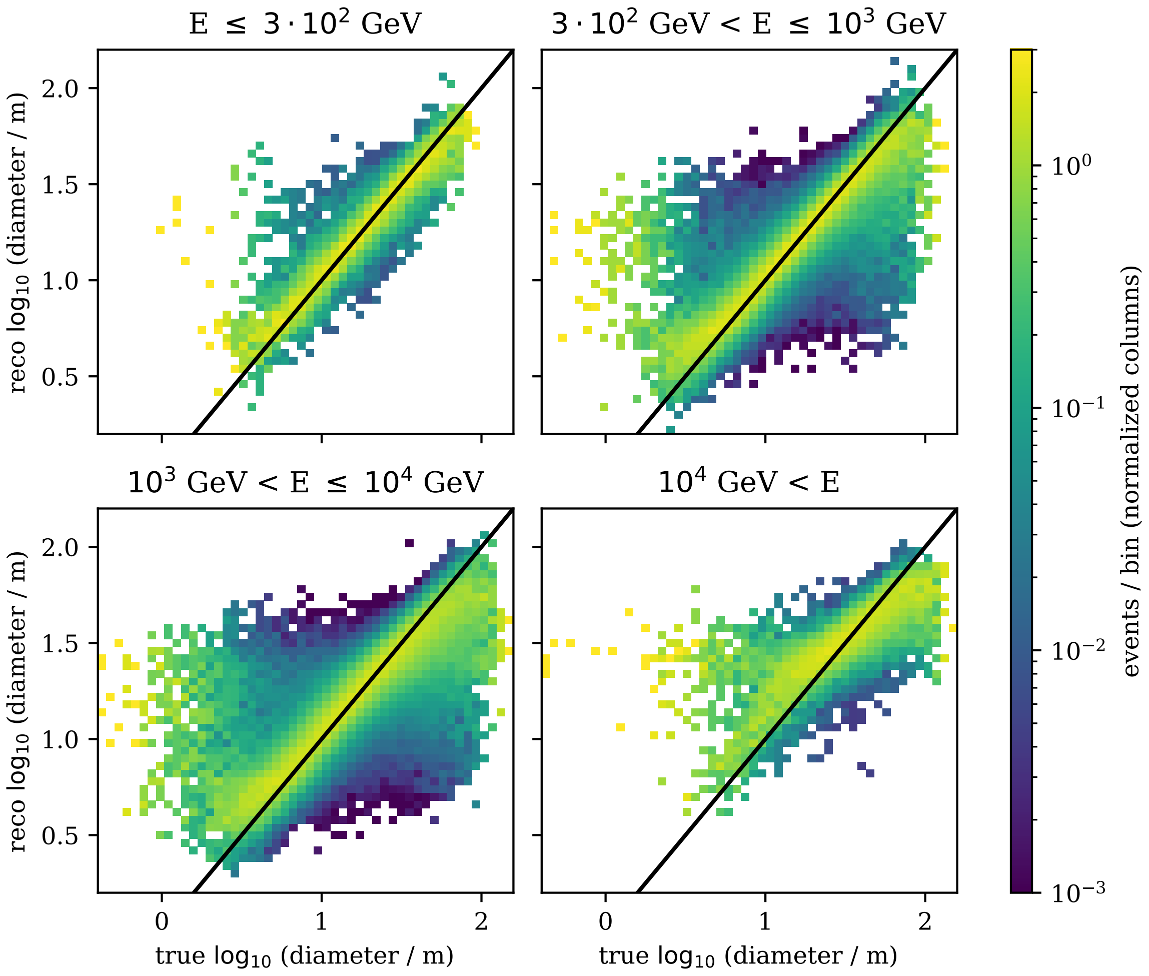


Figure 7.17.: 2D histograms of the bundle diameter reconstruction for the 44% of events with the highest reco quality, similar to Figure 7.15. Each plot only shows events in the specific true energy interval indicated in the titles. These intervals are also shown in Figure 7.18.

wrongly reconstructed a small diameter.

Events in Figure 7.22a often exhibit a typical spatial configuration, in which one or more muons are close to the instrumented volume, while the other muon or muons are in a distant separate group and close together. In cases like these, the network could try to measure the time distance between the arrival of the photons from the central muon(s) and the group that is far away. From this time difference, the diameter could be calculated.

Naturally, this strategy only works if the groups of muons have a large difference in the arrival time of their photons. In particular, it does not work if the muons have a similar distance to the center of the detector. Looking at the badly reconstructed events in Figure 7.22b, they generally seem to exhibit this exact pattern: The muons are often on a tangent to the instrumented volume. In conclusion, it seems plausible that the network actually follows the aforementioned strategy, at least for larger diameters.

7.3.4. Data-Monte Carlo comparison

In all plots in the previous subsections, we were only ever looking at muon bundles, that is, events with more than one muon. In fact, single muon events were even excluded from the dataset on which the network was trained, because the used label, the $\log_{10}$ of the diameter, can not be

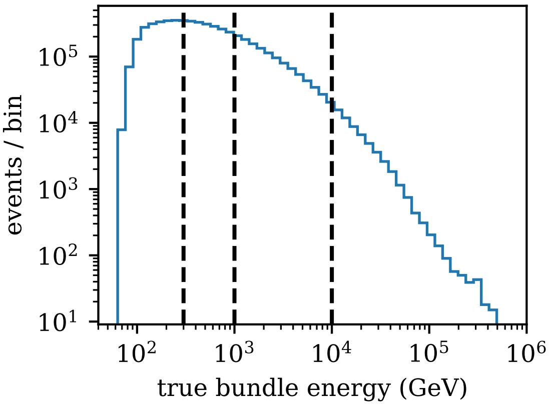


Figure 7.18.: Distribution of the true bundle energy, defined as the sum of the true energies of all visible muons in the bundle. The vertical black lines indicate the four intervals plotted in Figure 7.17.

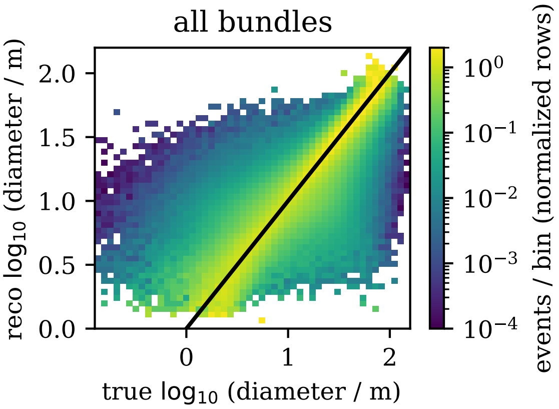


Figure 7.19.: 2D histogram of the bundle diameter reconstruction with the neural network. Similar to Figure 7.13 (left), except that now the bins in each *row* were normalized to have an area of one, instead of in each *column* as before.

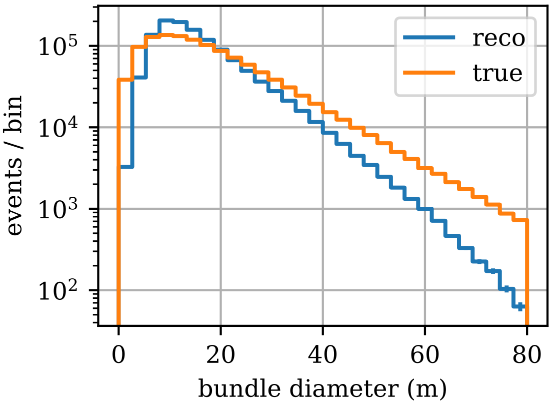


Figure 7.20.: Distribution of the true and reconstructed bundle diameter for all events with two or more muons.

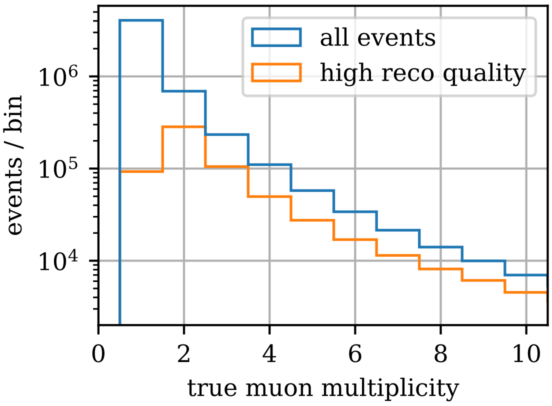
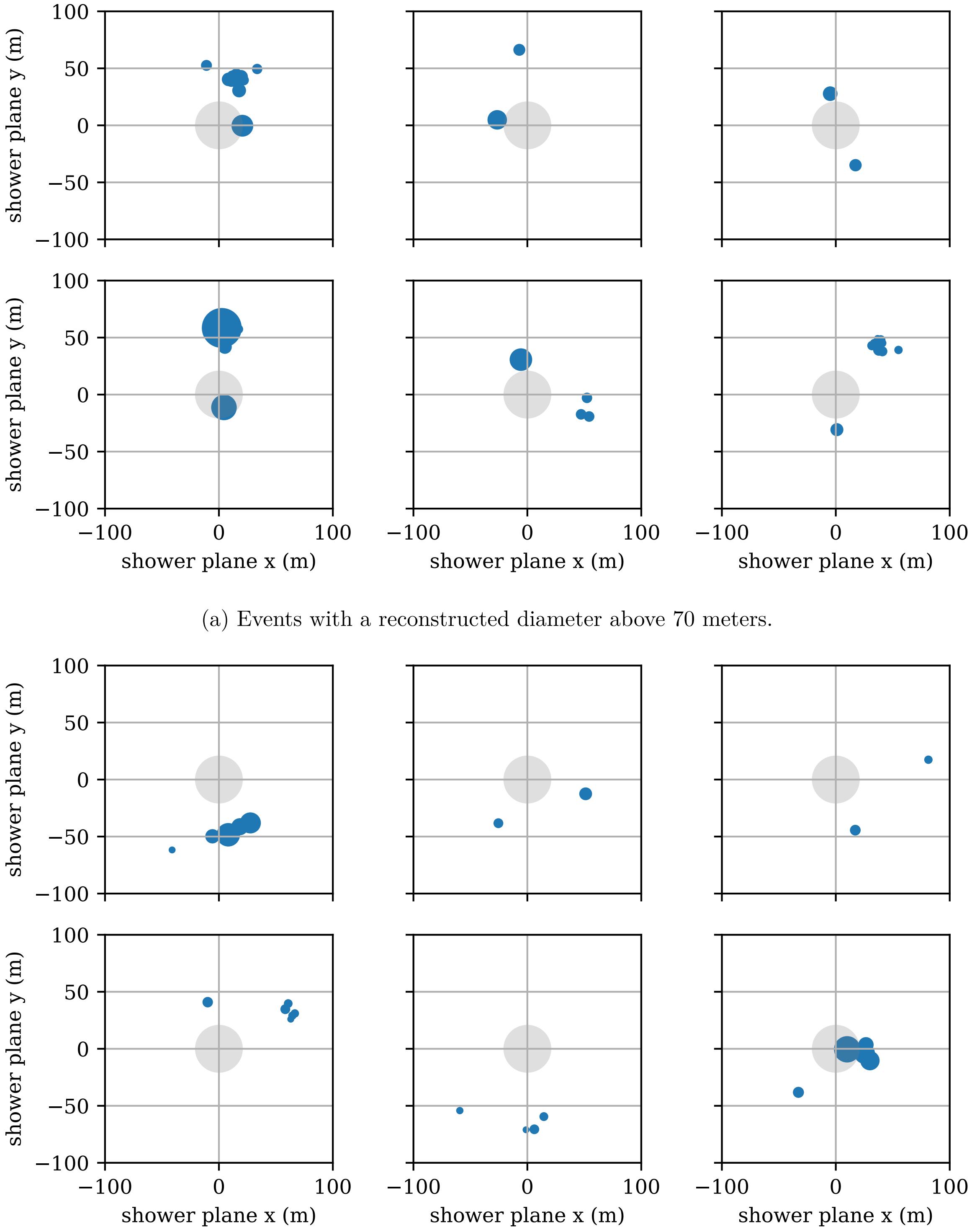


Figure 7.21.: Distribution of the true muon multiplicity for all events (blue) and only the best 44% of events according to the reconstruction quality (orange).



(a) Events with a reconstructed diameter above 70 meters.

(b) Events with a true diameter above 70 meters, but a reconstructed diameter below 30 meters.

Figure 7.22.: True muon positions in the shower plane of 12 random events. Each blue circle represents one muon, while the size of the circle is proportional to the number of McHits that the given muon deposited in the detector (i.e. the bigger the circle, the brighter the muon). A grey circle centred at the origin with a diameter of 40 meters is also shown, which roughly represents the instrumented volume of the Orca4 detector.

properly defined for an event with just one muon. However, most events encountered in practice are events with just one muon.

Ideally, the network would recognize that something is wrong with these single muon events, and reconstruct them with a high uncertainty. We could then easily remove these events by a cut on the reco quality. In Figure 7.21, this is investigated using the cut from Figure 7.15 that leaves only the best 44% of muon bundles (events with $\sigma < 0.15$). Unlike before, *all* events are plotted here, not just the bundles.

For a multiplicity of two or more, the number of events is reduced by a factor of about 2 - 3 as expected (since $1/44\% \approx 2.3$). The cut leaves slightly more events for higher multiplicities, probably due to the higher number of hits and energies of these events, which can make the reconstruction more precise. The number of events with just one muon is reduced by factor of about 40, which is much higher than for the other multiplicities. This means that the network is clearly capable of separating the single muon events, even though it was never trained on them. Naturally, not all single muon events are removed with the cut, for example because some of them might look very similar to a double muon event. Similarly, some double muon events in the training might have looked like as if there was just one muon, so the network might not be completely unfamiliar with event signatures of single muons.

Using the quality cut, which removes most of the single muon events, a comparison of the reconstructed diameter between the simulations and measured data can be made. This is shown in Figure 7.23. The distributions of the reconstructed diameter on both simulations and data were normalized here to have an area of one, so only the shapes of the distributions and not their overall scale is compared. As will be described in section 7.4, the reason for this is that Mupage has a shallower multiplicity distribution than what is seen in the measured data, so there is a significant excess of muon bundles in the simulations. Since the discrepancy in the multiplicity is discussed separately later, the distributions are normalized here.

The agreement between data and simulations is decent, the ratio being close to one between $\log_{10} R_b = 0.8$ and 1.5 ($R_b = 6$ meters to 30 meters). A slight excess of data events at the flanks of the distribution with diameters of about 5 meters and 50 meters is visible. This could be caused by an incorrect modeling of the radial distribution of the muons in Mupage. However, it could also arise from the shallower multiplicity distribution, since the lateral parametrization depends on the multiplicity [62]. In subsection 7.4.3, a reweighting of the Mupage events is proposed, which adjusts the simulated muon multiplicity distribution in order to match it with the measurement on data. Using this adjusted weighting for the data-MC comparison improves the agreement for the left flank, but not for the right one. It will be interesting to see if the agreement improves in upcoming productions of Mupage, which feature and adjusted parametrization.

7.4. Reconstruction of the muon multiplicity

The muon multiplicity N_μ describes the number of muons produced in an atmospheric air shower. As the muons propagate through water and air, they lose energy and may eventually decay before reaching the detector. In this work, the muon multiplicity therefore always refers to the number of muons at the detector. More specifically, as has been described in the introduction to chapter 7, only muons that deposit ten or more McHits ($\approx$ 5 hits) in the detector are counted towards the multiplicity. For example, if a bundle consists of three muons with 1, 10 and 20 McHits respectively, it would be assigned a muon number of two. The reasoning of this definition is that muons with very few hits are increasingly difficult to reconstruct, apart from possibly a statistical statement. For future studies, however, it might be worthwhile to change the definition to something that is not dependent on the detector response, for example by basing it on the path length of the muon in the instrumented volume.

The muon multiplicity is an important parameter for studying cosmic rays, as it is one of the

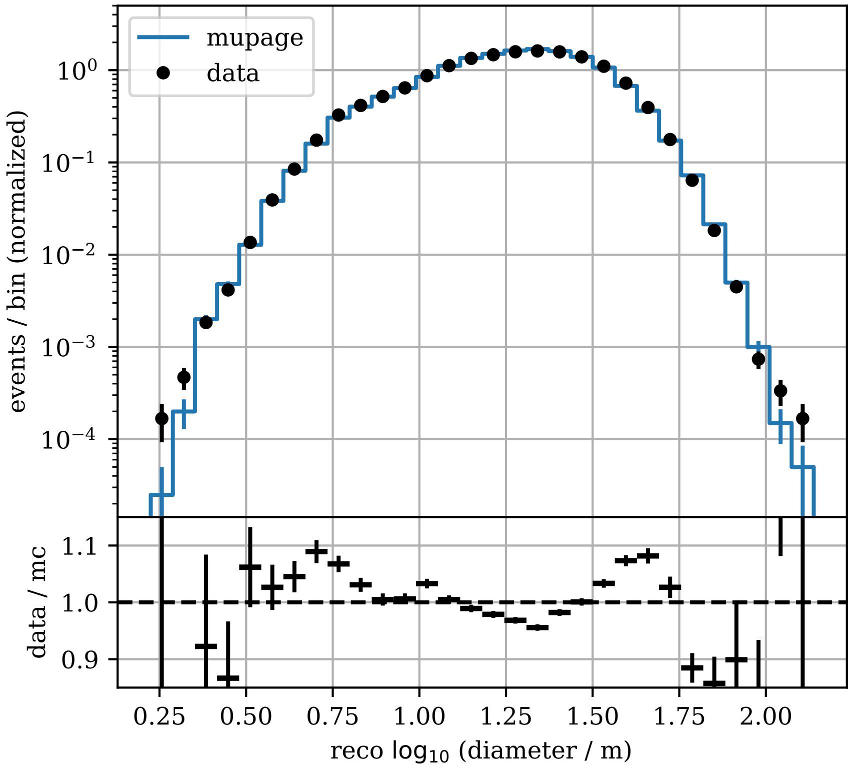


Figure 7.23.: Comparison of the reconstructed bundle diameter on simulations (solid line) and data (dots). The bottom plot shows the ratio between data and the simulation for each bin. Only events with a reco $\sigma < 0.15$ (best 44% of bundles on MC according to the reconstruction quality) are used in this plot.

main ways to estimate the mass of the primary particle [15]. Another commonly used observable for this is the depth of the air shower maximum X_{max}. This observable is typically reconstructed by measuring the electromagnetic component of the air shower. Since KM3NeT is only directly sensitive to the muonic (and neutrino) component of the shower, the muon multiplicity is crucial for studying cosmic ray induced air showers.

7.4.1. Architecture

A regression of the muon multiplicity is quite different from the other observables presented in section 7.2 and section 7.3, since it is a discrete quantity and bound from below to be greater than zero. It is worth noting that in very rare cases, simulated events can also have a multiplicity of zero (if all muons have less than ten McHits). For technical reasons, the label for the network is still set to one for these events. The distribution of the muon number roughly follows a power law, and has its maximum at $N_\mu = 1$. About 77% of the simulated events consist of only a single muon, 13% have two and 4% have three.

As before, the regression of the multiplicity can also provide an estimate of the event-by-event uncertainty. For this, an appropriate distribution needs to be chosen for the output of the network in order to follow the likelihood approach introduced in section 5.6. Due to the discrete nature of the muon number, this choice is not as straight forward as with continuous observables. A promising candidate is the negative binomial distribution, which is both discrete and bound from below. However, no satisfying results with this distribution could be obtained within the scope of this thesis.

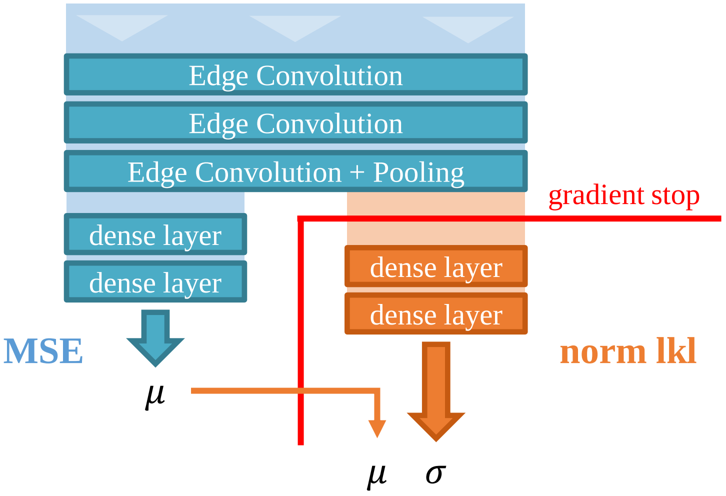


Figure 7.24.: Architecture of the network used for the multiplicity reconstruction. It consists of two separate output towers trained with different loss functions (mean squared error and normal likelihood).

Instead, the distribution of choice is yet again the normal distribution using the log-transformed multiplicity as a label, which in this combination can be thought of as letting the network output a log-normal distribution. While this distribution properly reflects the positivity of the multiplicity, it is continuous and not discrete. In practice, a network with such an output was observed to suffer from high instabilities during the training, preventing it from converging. On the other hand, using the simple and robust mean squared error as a loss function works fine, but does not provide an estimate on the uncertainty. So how can we get the best out of both worlds?

In section 5.6, it was shown that the MSE is a specialization of the normal likelihood loss with a constant $\sigma = 1$. To increase the stability of the training, we can therefore make use of a trick that has been used previously in KM3NeT [80]: The model is trained as usual with the MSE as a loss, resulting in a stable convergence. It outputs an estimator μ for the muon number, assuming a fixed $\sigma = 1$.

Additionally, a separate output tower of dense layers is added after the pooling layer of the last Edge Convolution, which is separated from the rest of the network via a gradient stop (see Figure 7.24). This prevents gradient updates from this part of the network to reach into the Edge Convolutional layers and the other output tower. The second tower is trained with the normal likelihood loss, using the μ as a constant from the first tower. It is not limited any more to the value of $\sigma = 1$, and instead can adjust the uncertainty on an event-by-event basis. Even though this part of the network can be very unstable, sudden sharp increases in its loss do not affect the training progress of the main part of the network, the Edge Convolutional layers. Thus, as these layers get closer and closer to convergence over time, the σ tower can stabilize as well.

In summary, this trick can essentially be thought of as fitting the μ of the normal distribution while keeping σ constant, and then fitting σ separately while keeping μ constant. While it is quite obvious that this can increase the stability of the fit, it can also lead to a different result. After all, it is not infeasible that μ would have a different - possibly better - minimum if it was fit together with a variable σ at the same time. Such a behavior was actually observed for the directional reconstruction: Here, the performance slightly increases when using the normal likelihood approach instead of the MSE. However, the trick is at least strictly superior to the MSE-only approach, as it additionally provides an uncertainty estimate without affecting the reconstruction of μ.

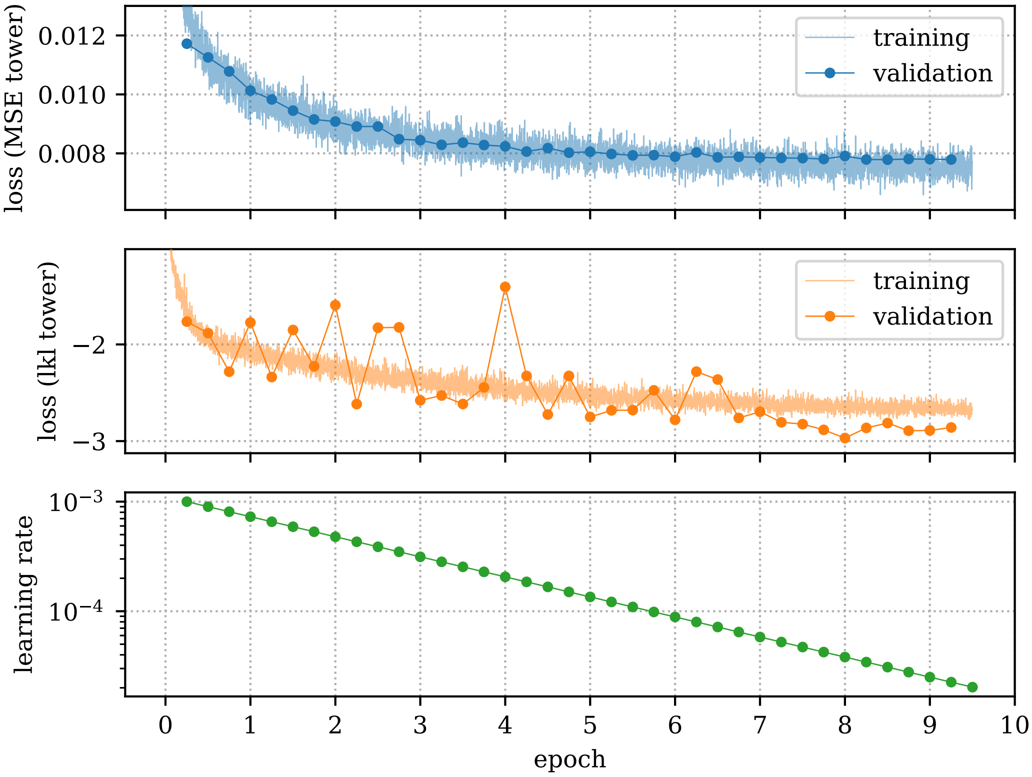


Figure 7.25.: Training history of the neural network used for the multiplicity reconstruction. Shown are the losses of the two output towers of the model (top and middle) and the learning rate (bottom) over the epoch, i.e. the number of iterations over the training dataset.

7.4.2. Performance evaluation on simulations

Unlike the previous networks, the model for the reconstruction of the muon number is trained and validated on a dataset generated using the KM3 light simulator, instead of JSirene. This was done because a slight degradation of the performance was noticeable when training the model on the JSirene dataset. Updates to JSirene released in the meantime might have fixed this effect, though. The degradation was only observed for the reconstruction of the muon number, and not for the diameter or the direction. The data with which all the plots in this section are generated is still the same JSirene test set used in the previous sections.

The training of the network was done on a GTX 1080ti GPU, and finished after about three days. In Figure 7.25, the course of the loss for the two output towers of the network is shown. The learning rate was decayed from its original value of 0.001 by a factor of 0.9 after every quarter of an epoch, until the losses showed no further improvement. While the curve of the MSE tower is very stable, large fluctuations are visible for the normal likelihood tower. However, the curve gets more and more stable as the training progresses.

The performance of the reconstruction is visualized as 2D histograms in Figure 7.26. Since the event rate of muon bundles falls off steeply as the multiplicity increases, each column was normalized to sum up to the same value in these plots. In the left of the two histograms, it can be seen that the reconstruction is close to the truth up to true multiplicities of about 50, with a slight tendency to underestimate the truth. Above 50, the underestimation becomes a lot stronger. Apart from that, the distribution of the reconstruction around the truth widens up toward a higher muon number. This is expected, as it is plausible to separate a two from a three muon event, but more difficult to tell apart events with 30 muons from those with 31.

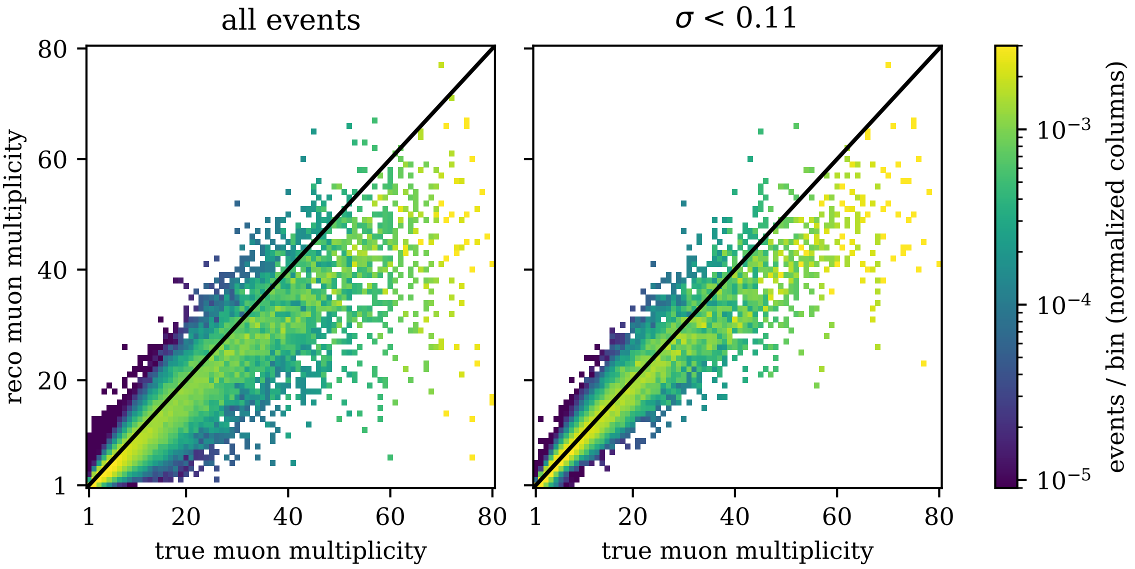


Figure 7.26.: 2D histograms of the muon multiplicity reconstruction versus the truth. The straight black lines through the origin show the ideal case where the reconstruction is equal to the truth. The plot on the left shows all events, while the one on the right shows only the 82% of events with a reconstructed $\sigma < 0.11$. Each column has been normalized to have an area of one.

This underestimation at high muon numbers might have multiple reasons: The number of events in the region of high multiplicity is so low in the used MUPAGE simulation, that it might negatively impact the performance. Using a weighted simulation with a flatter distribution would address this issue. On the other hand, it could also be caused by the selection of hits. As mentioned in the introduction of chapter 7, only events within a time window of -250 ns to +1000 ns around the first triggered hit were used for generating the input to the network, and only up to 2000 hits per event. This selection might negatively impact the reconstruction of events with a high multiplicity, since they also have more hits than the average event. Relaxing these conditions might address the observed bias.

It is more difficult here to judge how well the estimation of the reconstruction quality works than it was for the direction or the diameter reco task. For these, a pull plot and a comparison to the standard normal distribution was reasonable. However, the pull distribution of the muon multiplicity reco does not follow a normal distribution due to its discrete nature, especially for lower multiplicities close to the minimum multiplicity of one where the event rates are highest. Instead, the error estimation can be visualized by observing the effect of cuts on the reco quality.

As an example, the right plot of Figure 7.26 uses an arbitrary cut on the reconstruction quality $\sigma < 0.11$, which leaves 82% of the original events. Many of the events with an underestimation were removed by this cut, as they were assigned a high uncertainty by the network. This suggests that the bias might have been purposefully introduced by the network in order to improve its performance. Similar to what is described in subsection 7.3.3 for the reconstruction of the diameter, this could be achieved by taking into account the steeply falling true distribution of the muon number. This learned bias could probably be prevented by training on a flat true distribution. However, since the power law of the muon multiplicity is also observed on real data, it might actually be advantageous to let the network have this bias.

As mentioned before, the cut $\sigma < 0.11$ leaves 82% of events. It is important to note that, since higher multiplicities appear so rarely, it is not immediately clear how this cut affects the underlying distribution of the muon number. In Figure 7.27, this impact is visualized. Cuts on

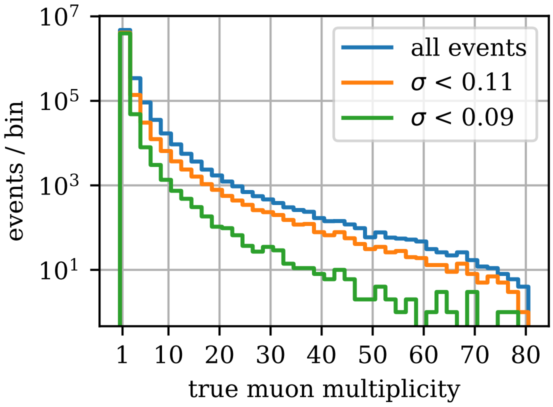


Figure 7.27.: Distribution of the true muon multiplicity for all events (blue) and using different cuts on the reconstruction quality (orange and green).

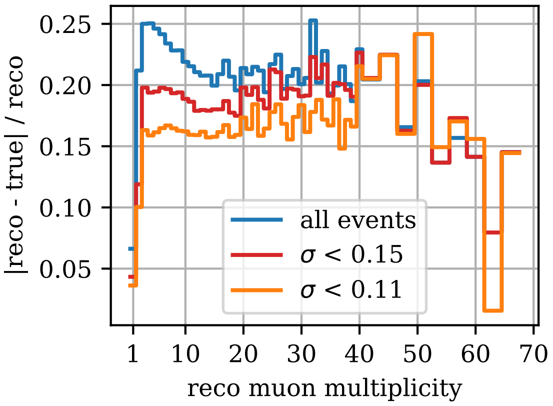


Figure 7.28.: Mean relative error over the reconstructed muon multiplicity, using different cuts on the reconstruction quality.

σ generally affect events with just a single muon much less strongly than bundles. Apparently, it is easier for the network to separate one muon from multiple than it is to distinguish bundles from each other.

In Figure 7.28, the relative error of the muon multiplicity reconstruction is plotted over the reco. Single muon events can be reconstructed much more precisely in relative terms than higher multiplicities. A plateau with an average error over all events of $\approx$ 20% is reached for reco $N_\mu > 10$. In the interval between reco $N_\mu = 2$ and 10, the error has a spike up to about 25 % caused by events with a low reconstruction quality, which can be removed with a corresponding cut.

The muon number is of course correlated to the number of hits in an event, as each additional muon can further deposit hits in the detector. But is the network doing more than simply counting up the hits? This is investigated in Figure 7.29, where the performance is shown for different intervals of the number of hits. As can be seen there, even for events with a similar number of hits, the network still manages to separate low multiplicities from higher ones.

7.4.3. Data-Monte Carlo comparison

Unlike in Mupage, not all events in the measured data contain a muon or muon bundle. Apart from the extremely rare neutrinos, a few percent are triggered randomly by noise and, to a much smaller extent, by afterpulses or sparking PMTs. Since these events are not part of the Mupage dataset on which the network was trained, they should be removed from the selection before a comparison between data and simulations. As shown in subsection 7.2.3, a cut on the quality of the zenith angle reconstruction can be used to exclude the majority of these events.

Even though the number of events triggered by these effects is small in comparison to the high rate of muon events overall, it is still important to investigate the impact of this on the reconstruction of the muon multiplicity. The flux of muons is decreasing quickly towards higher muon numbers, so if these background events were to be reconstructed with a high muon multiplicity, even a small fraction of such events remaining in the selection could severely limit the validity of the reconstruction.

We can get an estimate of which muon number pure noise events get reconstructed as by making use of a cut on the uncertainty of the reconstructed zenith angle. As described in subsection 7.2.3, most of the events with a $\sigma_{\text{zenith}} > 0.16$ rad contain only noise and no muons. Figure 7.30a shows

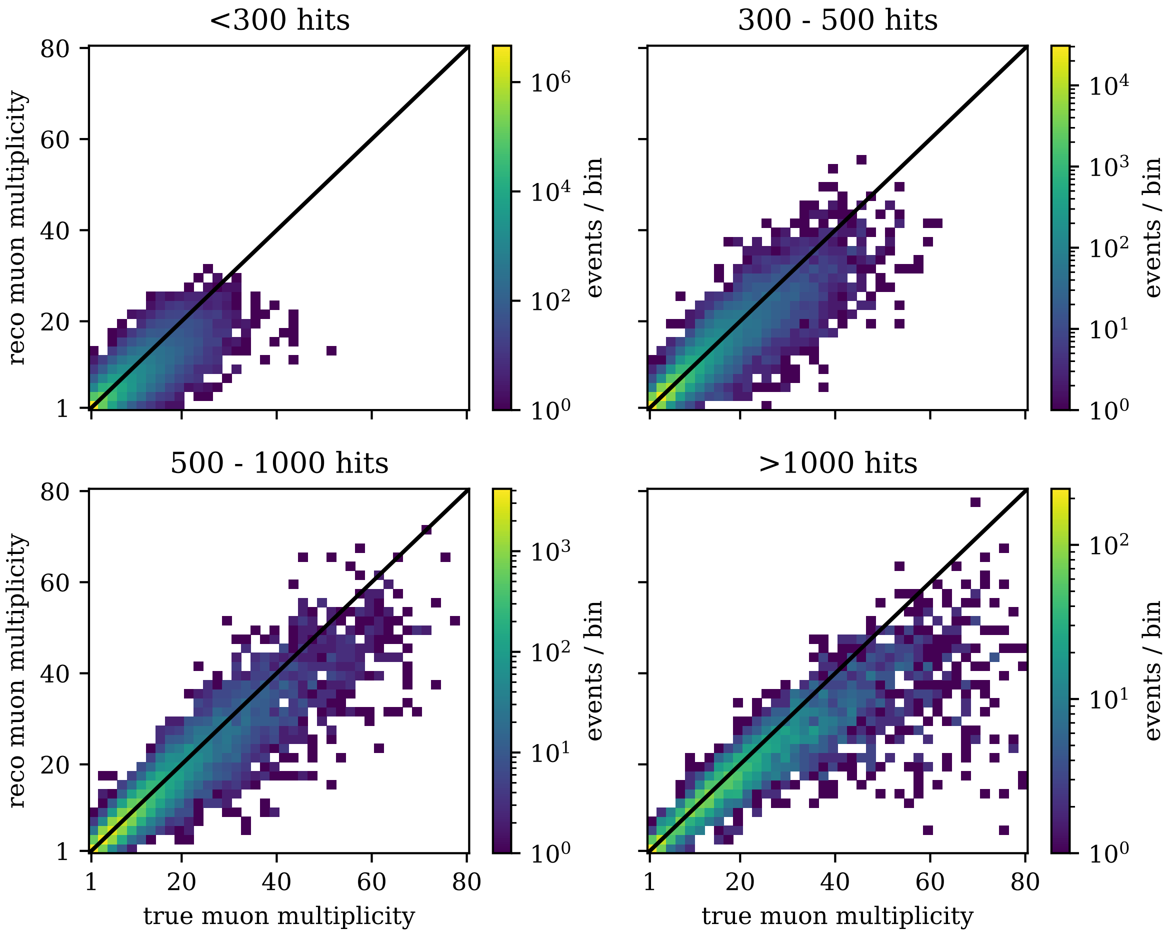


Figure 7.29.: 2D histograms of the muon multiplicity reconstruction, similar to Figure 7.26. Only events with a specific number of hits were used for each plot, as indicated by the titles.

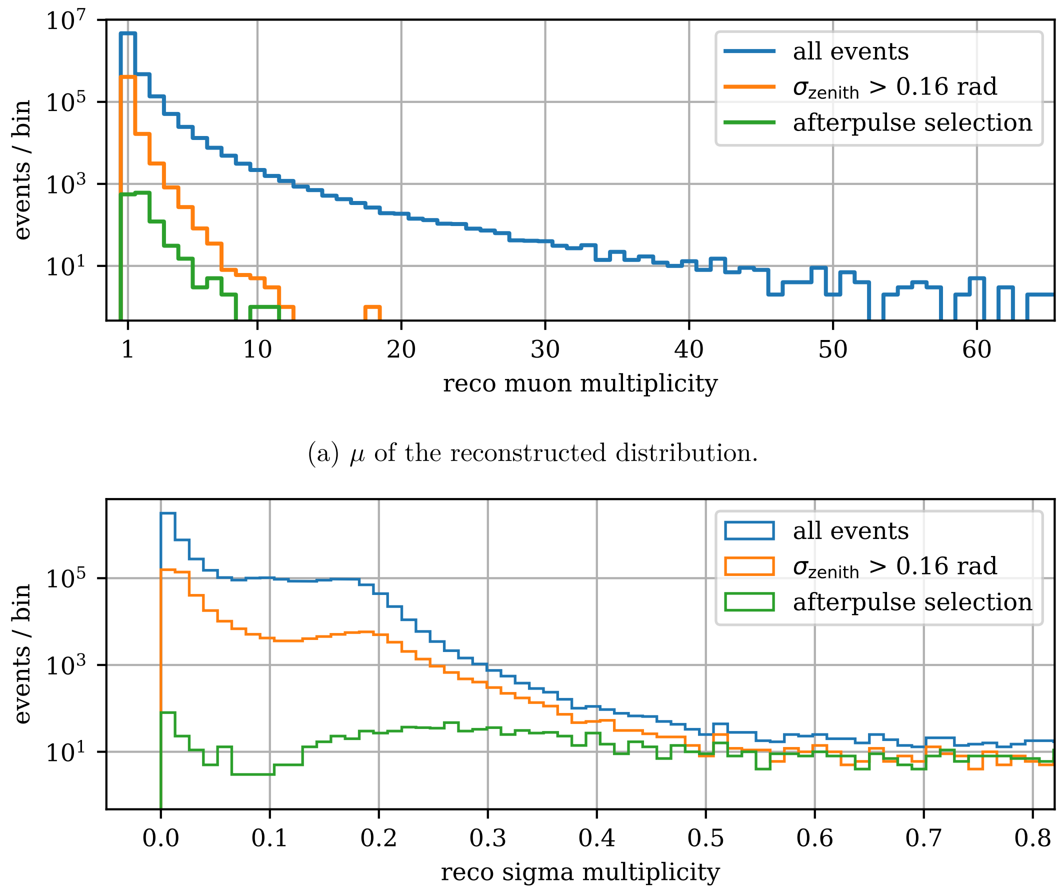
the distribution of the reconstructed muon multiplicity for these events, which are clustered at low multiplicities. This is very advantageous, as the overall flux of muons is highest there, so any noise events possibly not removed by a cut would result in a neglectable contamination.

Intuitively, it seems reasonable that pure noise events with their low number of hits get reconstructed mostly as one- or few-muon events, as higher multiplicities would be expected to lead to many more hits. However, this does not apply for the events triggered by afterpulses or sparking PMTs. As was seen in Figure 7.10, these events often have high numbers of hits, albeit typically in a different, non-track-like arrangement.

To investigate how the network reacts to these events, a set of cuts is used to isolate a small sample which consists to a large degree of afterpulses and sparking PMTs. These cuts are:

- the event triggered directly before the current one had at least 500 hits
- less than 30% of the hits in the current event are triggered hits
- the time between first and last triggered hit is more than 400 ns

1345 events out of the 5.4 million events in the measured dataset satisfy these conditions. By looking at Z-T plots of random events in this afterpulse cut, it can be estimated that about 75% of the selected events actually show afterpulses or sparking PMTs, with the rest being normal tracks. Similar to pure noise events, Figure 7.30a shows that they get reconstructed with a low



(a) μ of the reconstructed distribution.

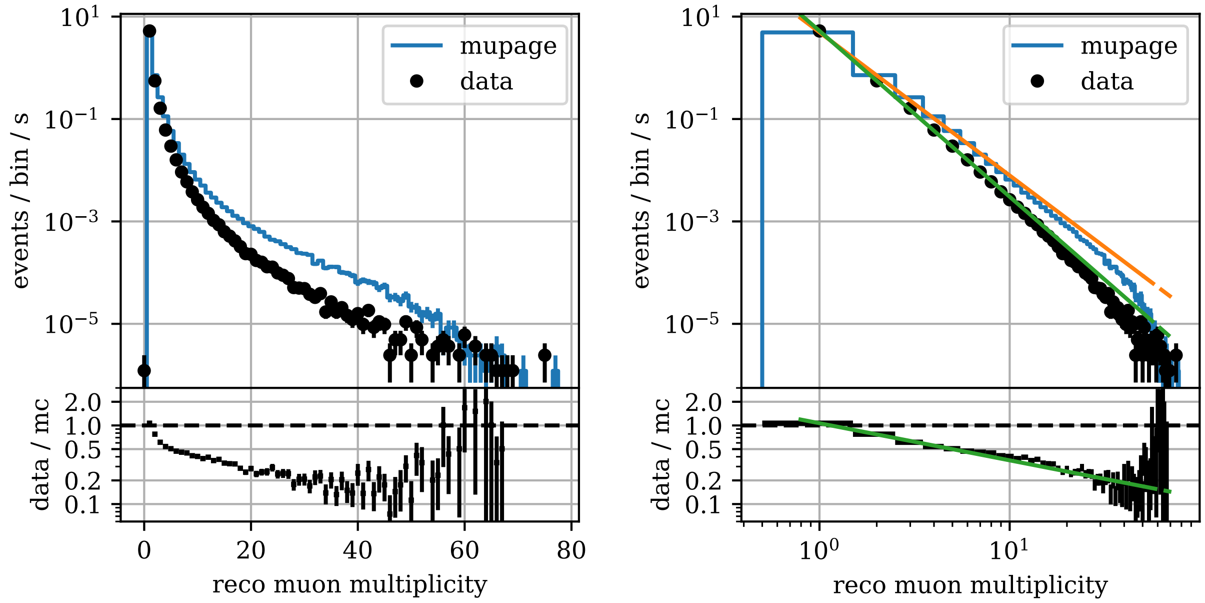
(b) σ of the reconstructed distribution.

Figure 7.30.: Distribution of the reconstructed muon multiplicity and uncertainty on measured data. The different curves show all events (blue), events with a high uncertainty in the directional reco (orange, mostly noise events), and events that are likely to be afterpulses (green).

muon multiplicity. While it is likely that the proposed cut criteria do not identify all of the desired events, they pose very little background at these low muon numbers. Additionally, they get reconstructed with a very high σ on average, so even a moderate cut on the reconstruction quality can be used to reduce the contamination further (see Figure 7.30b).

For the following, only the 97% of events in MUPAGE and the 92% of events on data with $\sigma_{\text{zenith}} < 0.16$ rad are used for the plots, which eliminates many of the pure noise events present on data but not the simulation. In Figure 7.31a, the distributions of the reconstructed muon number are compared between MUPAGE and measured data. While the agreement is good for $N_\mu = 1$, the distribution is much steeper on data and reaches an event rate as low as only 0.1 of what is expected from the simulations around $N_\mu = 45$ to 50. The rates are closer to each other again for multiplicities above 50, but low statistics do not allow for a definitive evaluation.

Is such a large discrepancy between MUPAGE and measured data feasible, or is this simply a sign that the network does not work properly when confronted with data that is potentially different from the simulations it was trained on? Unfortunately, no classical reconstruction of the muon number exists yet, so it is not possible to directly perform a cross check of the deep learning reconstruction. But as established earlier, many low-level observables, like the number of hits or triggered DOMs in an event, are proportional to the muon multiplicity. So, if there is indeed such a large discrepancy in the muon number, it should be visible in these quantities, too.



(a) Linear scale.

(b) Log scale with fitted powerlaws.

Figure 7.31.: Data-Monte Carlo comparison of the reconstructed muon multiplicity, using only events with $\sigma_{\text{zenith}} < 0.16$ rad.

The event rates on data and MUPAGE for a selection of several such observables are shown in Figure 7.32. Many of them indeed show an excess in MUPAGE with a similar order of magnitude as the reconstruction of the muon number.

It is therefore interesting to see to which degree these differences can be explained by the mismatch of the muon number. This can be investigated by reweighting the MUPAGE events so that the data-Monte Carlo ratio of the multiplicity reconstruction is close to one. For this, the data-MC ratio shown in the bottom plot of Figure 7.31a is used. Each MUPAGE event is then weighted according to this ratio and its true muon multiplicity. For example, the data-MC ratio is about 0.5 for a reconstructed muon multiplicity of 5. Consequently, each MUPAGE event with a true muon multiplicity of 5 is assigned a weight of 0.5. Since the ratio gets increasingly unstable for $N_\mu > 30$, all events with a true muon multiplicity above 30 get the same weight as an event with 30 muons.

These reweighted distributions are shown as the orange curves in Figure 7.32. For many observables, like the number of PMTs that recorded a hit in an event, the number of triggered hits or triggered DOMs, the data-Monte Carlo agreement is substantially improved. Also for parameters related to the classical reconstruction (JGandalf) a clear improvement can be observed, even though some differences in the fit quality (likelihood/n_hits) remain.

In Figure 7.31b, the same data-Monte Carlo comparison as in Figure 7.31a is shown, but on a log scale for the muon number. As was described in subsection 4.1.2, the dependency of the muon bundle flux Φ on the multiplicity m can be described by a power law

$$\Phi(m, h, \theta) = K(h, \theta) \cdot m^{-\nu(h,\theta)}. \tag{7.2}$$

The parameters K and ν depend on the height h and the zenith angle θ. Figure 7.31b shows the flux integrated over these quantities, but they still seem to follow a power law up to multiplicities of about 20, after which the flux decreases more strongly. This reduction is much more prominent on MUPAGE than on data for muon numbers above 50. Two power laws have been fitted to the

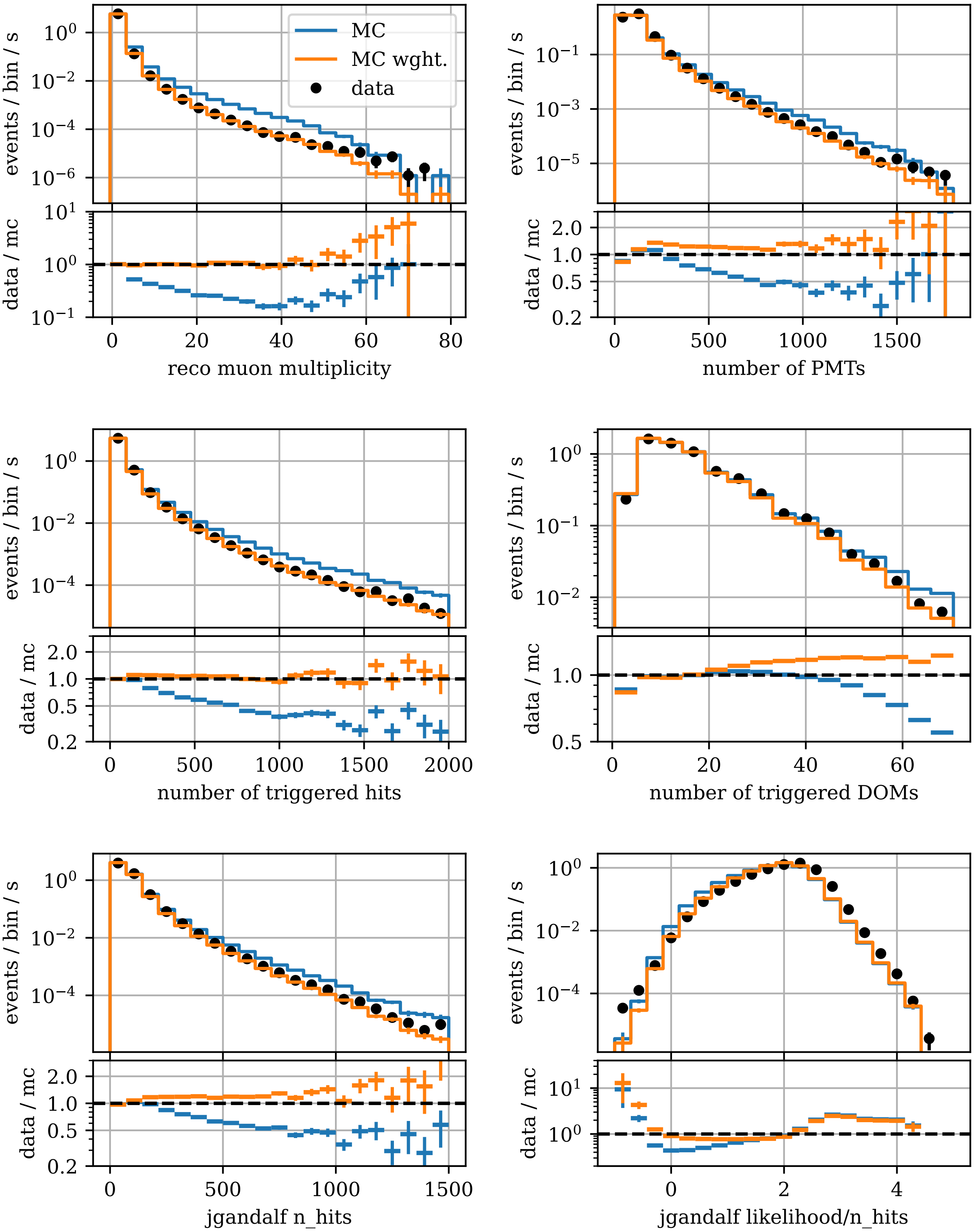


Figure 7.32.: Data-Monte Carlo comparison of various quantities, using only events with $\sigma_{\text{zenith}} < 0.16$ rad. The blue curves are for the default MUPAGE weighting, whereas the orange curves were reweighted in such a way that the distributions of the reconstructed multiplicity (top left) on data and simulations match up to $N_\mu = 30$.

distributions in Figure 7.31b via a χ^2 fit in the interval between one and 50 muons, which takes the form of a straight line in the double log plot.

Interestingly, the distribution of the ratio between data and simulation follows a power law more closely than the muon numbers themselves. This suggests that the effect that causes the reduction in flux for multiplicity above 20 is present to the same multiplicative extent in both the data and the simulations. By using Equation 7.2, the following formula for the expected distribution of the ratio between data and simulation can be obtained:

$$\frac{\Phi_{\text{data}}(m,h,\theta)}{\Phi_{\text{mc}}(m,h,\theta)} = \frac{K_{\text{data}}(h,\theta)}{K_{\text{mc}}(h,\theta)} \cdot m^{-(\nu_{\text{data}}(h,\theta)-\nu_{\text{mc}}(h,\theta))}$$

The height- and zenith-angle integrated quantities were determined via the fit to the data-MC ratio of Figure 7.31b in the interval between one and 50 muons to be:

$$\begin{aligned}\frac{K_{\text{data}}}{K_{\text{mc}}} &= 1.0671 \pm 0.0020 \\ \nu_{\text{data}} - \nu_{\text{mc}} &= 0.472 \pm 0.005. \end{aligned} \tag{7.3}$$

This indicates that a tuning of the original MUPAGE parameters could fix the observed differences. However, the parameters obtained in Equation 7.3 might not suffice for this, as they do not respect the zenith and depth dependency of the muon multiplicity. There are various ongoing efforts in KM3NET that pursue alternative strategies, like making use of CORSIKA simulations to update the MUPAGE parametrization, or perform a multi dimensional optimization based on various other observables [103].

Chapter 8.

Cosmic ray composition measurement

In the previous chapter 7, the reconstruction and evaluation of the different bundle properties has been performed using exclusively Mupage simulations. Due to the relatively low computational effort to simulate events in Mupage, vast numbers of atmospheric muon bundles are available, which is especially helpful during the training of the deep neural networks. However, these simulations do not contain information about the primary that produced the air shower from which the muons originate. Corsika simulations do provide this information, but they are much more expensive to generate and therefore typically only available in fewer numbers.

The strategy followed in this chapter is to use the networks that were trained on the large Mupage dataset, and then apply them to the much smaller Corsika dataset for measuring the primary composition. This way, the network can make use of large statistics for learning the features of bundles, while the entire Corsika production with its information about the primary can be used for the measurement, since none of its events were used during the training. Additionally, this approach doubles as a cross check of the reconstruction, since the training of the network is done on a different set of simulations than the measurement itself. Furthermore, this also allows for a systematic study to assure that the network retains its performance when switching from Mupage training set to Corsika.

As before, the preprocessing of the data is done using OrcaSong (see subsection 5.7.1). Since Corsika contains additional information compared to Mupage, the OrcaSong configuration file given in the appendix in section A.1 was slightly adjusted accordingly.

8.1. Corsika dataset

First studies with the Corsika dataset provided by the Collaboration have shown satisfying agreement to Mupage simulations and measured data [106]. It was simulated with Corsika version 7.6400 using Sibyll 2.3c [49] for high energy hadronic interactions, and Geisha 2002d [53] for the lower energies. In total, 2.5 million atmospheric showers were simulated with primary energies between 10^3 and 10^9 GeV, resulting in about 242,000 triggered events in the KM3NeT/ORCA detector. Five different nuclei have been used as the primary particles of the shower: proton, helium, carbon, oxygen and iron. The number of triggered events is roughly even among the different primaries, with about 48,000 events for each of them. The flux of events is calculated by weighting each event according to its primary type and energy using the GST-3 cosmic ray flux model [55].

The networks used in this section are the exact same ones that have been introduced in chapter 7. As described there, these models have been trained on Mupage simulations, and are applied in inference mode (i.e. with their free parameters being fixed) on the Corsika simulations in the following. Similar to the Mupage test set, the Corsika data has therefore not been seen by the network at any point during the training or validation. For comparisons with Mupage and measured data, the corresponding datasets from chapter 7 described in Table 7.1 are used.

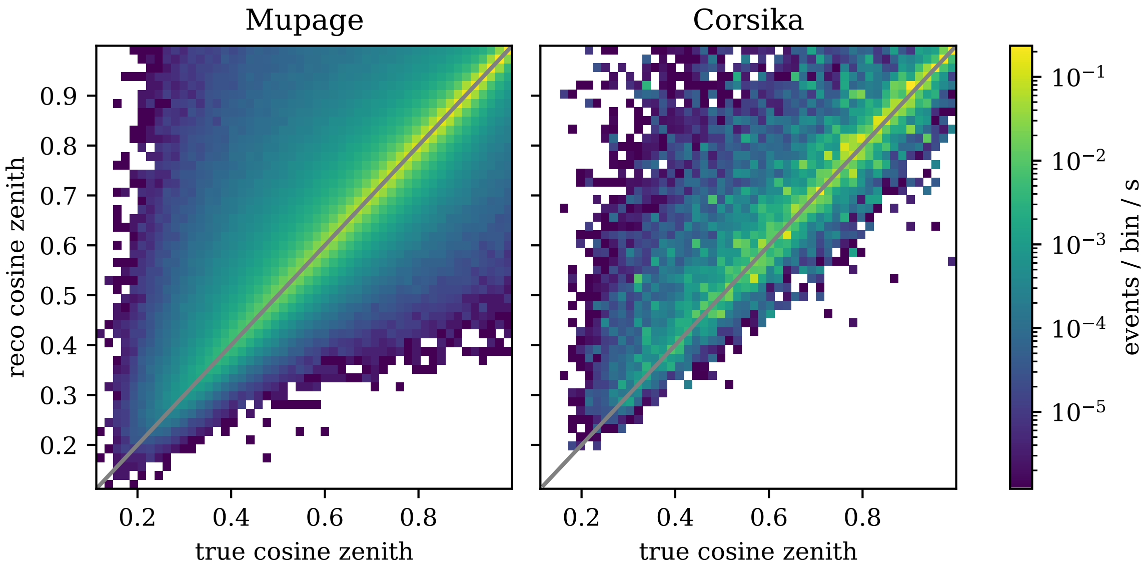


Figure 8.1.: 2D histograms of the directional reconstruction of muon bundles with the neural network, similar to Figure 7.2. The plots show data from the Mupage (left) and the Corsika (right) datasets. Bins below the minimal rate of the colorbar are set to the same dark blue color, or to white if if they are below 20% of the minimum.

8.2. Performance of the GNNs on Corsika

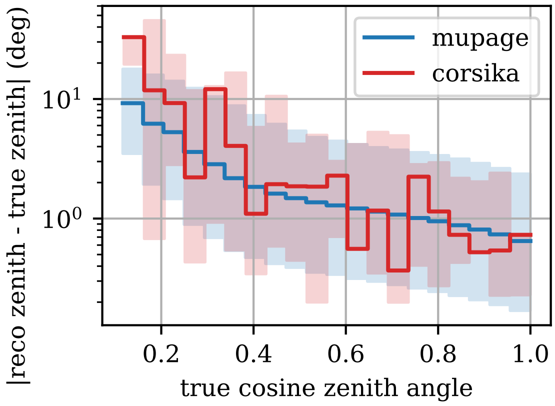
8.2.1. Direction

As described in section 7.2, this network reconstructs the incident direction of an atmospheric muon or muon bundle. Unlike in Mupage, the muons in bundles of Corsika are in general not fully parallel, so the incident direction is defined as the direction of the primary in this case instead. In Figure 8.1, the reconstructed cosine zenith angles on both the Mupage and Corsika simulations are compared to the truth.

The color indicates the rate of events in each bin. For Mupage, this can be calculated by dividing the number of events in a bin by the total simulated livetime (in this case 9.6 days). For Corsika, the flux is computed using the `weight_w3` included from the simulation chain divided by the total number of simulated air showers per primary (five million). In the case of Mupage, a bin cannot have a non-zero flux below $r_{\mathrm{min}} = 1$ / 9.6 days $\approx 1.2 \cdot 10^{-6}$ s^{-1}, which is reached if there is only one event in a bin. For Corsika, however, the flux can be much lower due to the weighting. In order to make the plots comparable, all bins with a flux between r_{min} and $0.2 \cdot r_{\mathrm{min}}$ are plotted in the same dark blue color, and bins with a rate below $0.2 \cdot r_{\mathrm{min}}$ are set to white.

The performance of the network is very similar on Mupage and Corsika, as the zenith reconstruction is close to the truth in both cases. The counts per bin are fluctuating much more strongly in the case of Corsika, though. This is because the Corsika dataset consists of only 242,000 events - much fewer than the 5.3 million events of the Mupage JSirene test set. Additionally, many of these events contribute only little to the overall flux, as they originate from the rarer, heavier primaries, or have a large energy. For example, the proton events alone account for about 69% of the flux, but only for 18% of the simulated events. On the other hand, oxygen, carbon and iron together make up 7% of the total event rate, but 63% of the simulated events. The larger fluctuations from bin to bin seen in Figure 8.1 are therefore mainly caused by the small subset of events with a high weight from the overall already quite small Corsika dataset.

In Figure 8.2, the error in the zenith reconstruction is compared between Mupage and Cor-



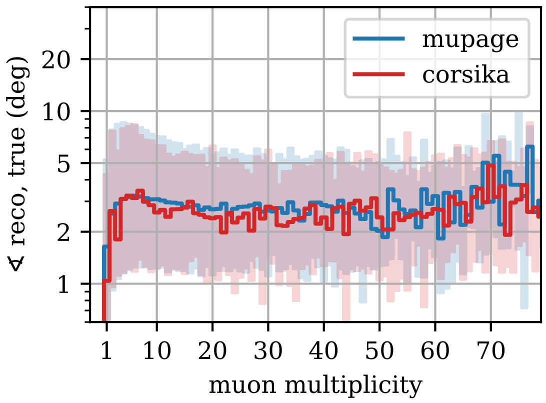


Figure 8.2.: Error in the zenith reconstruction for events simulated with Mupage (blue) and Corsika (red) over the true cosine zenith angle, similar to Figure 7.3. The solid line is the weighted median of the distribution for each bin, while the colored region is the 68% interval around the median.

Figure 8.3.: Angle between the reconstructed direction and the true direction as a function of the true muon multiplicity for Mupage (blue) and Corsika (red), similar to Figure 7.4. The solid line is the weighted median of the distribution for each bin, while the colored region is the 68% interval around the median.

sika. The low statistics lead to large fluctuations here as well, but it can still be seen that the performance is very similar on the two datasets. Since muons with higher multiplicities are much more rare, they carry little weight in this graph. In Figure 8.3, the performance of the directional reconstruction is therefore compared as a function of the muon number. As before, there is little to no difference between Mupage and Corsika, even for larger bundles. Since heavier primaries tend to produce bundles with more muons, the fluctuations from bin to bin are less prominent in this plot.

A data-Monte Carlo comparison of the reconstructed cosine zenith angle is shown in Figure 8.4. In order to remove events that have been triggered by noise instead of muons, events with a high uncertainty in the deep learning reconstruction of more than $\sigma_{\text{zenith}} = 0.16$ rad have been cut on both data and simulations for these plots. This is the same approach that was introduced in subsection 7.2.3, which leaves about 97% of the total event rate on both Mupage and Corsika. For reference, the comparison is additionally also done with the result from the classical reconstruction using the same cut.

For Mupage and measured data, the rates shown in the top part of the plot are calculated as described in subsection 7.2.3 by dividing the number of events in each bin by the overall livetime. For Corsika, the approach is slightly more complicated, as the weight of each event has to be taken into account. The rate r_{corsika} of the n events in a bin is therefore given by the sum of the weights w_i of all the events contained in it. As described above, w_i is calculated by taking into account the total number of simulated air showers. The uncertainty σ_{corsika} per bin is given by the square root of the squared sum of each weight w_i:

$$r_{\text{corsika}} = \sum_{i=1}^{n} w_i, \quad \sigma_{\text{corsika}} = \sqrt{\sum_{i=1}^{n} w_i^2}. \tag{8.1}$$

For both techniques, the agreement between Mupage, Corsika and data is decent in the interval between a cosine zenith of 1 and 0.4 (0 to 67 degrees from vertical down going), although

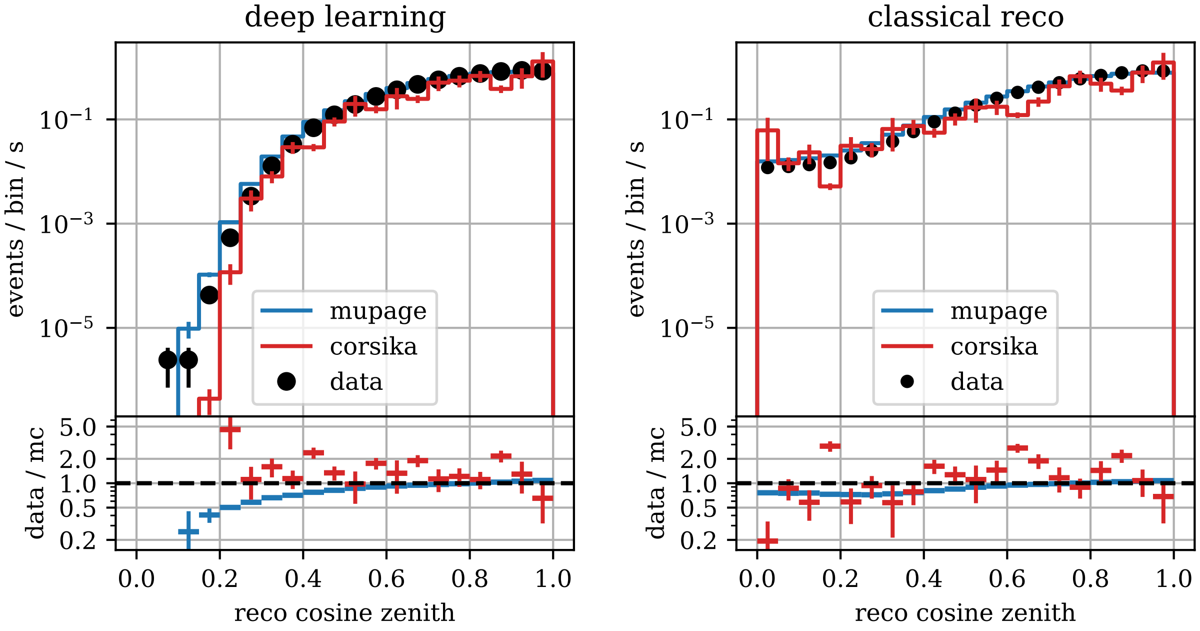


Figure 8.4.: Data-Monte Carlo comparison of the reconstructed cosine zenith angle, similar to Figure 7.8. The plots only show events from the Mupage (blue) and Corsika (red) datasets that have an uncertainty of the zenith angle reconstruction σ_{zenith} below 0.16 rad. The plot on the left shows the result of the deep learning reconstruction, and the one on the right shows the results from the classical reconstruction for reference.

the low statistics only allow for evaluating this down to a factor of about two. The disagreement between deep learning and the classical reconstruction for cosine zenith angles below 0.4 is caused by the fact that the classical algorithm is bias-free toward the horizon, and produces different results for bundles. This was discussed to greater extent in subsection 7.2.3.

The deep learning reconstruction shows a much more pronounced decrease of the rate for cosine zenith angles below 0.25 on Corsika than on Mupage. The classical reconstruction does not exhibit this behavior. However, the distribution of the classical reconstruction does not match with the true distribution of the zenith angle in this interval at all (cf. Figure 4.3), since it is dominated by misreconstructed events there. The deep learning reconstruction should therefore be seen as evidence for a mismatch of the zenith distribution between Mupage and Corsika for close-to-horizontal events.

8.2.2. Diameter

As described in section 7.3, the bundle diameter is defined as the maximum perpendicular distance between any two visible muons in a bundle. In Mupage, all muons are simulated as traveling in parallel, so their lateral positions can be easily defined in a plane that is perpendicular to all of them. In Corsika, however, each individual muon travels at a slightly different angle depending on the transversal momentum it received during its production in the air shower. Their lateral positions are therefore instead defined using a plane that is perpendicular to the trajectory of the primary particle. In both cases, the origin (x, y) = (0, 0) of the coordinate system of this plane is set to the center of the Orca4 detector, and the positions of the muons in the plane are the points at which their trajectories intersect with it.

Figure 8.5 compares the reconstructed distributions of the diameter on Mupage and Corsika using two-dimensional histograms. Only events with at least two muons are used for this plot, since the diameter cannot be defined for events with just one muon. As before, the small

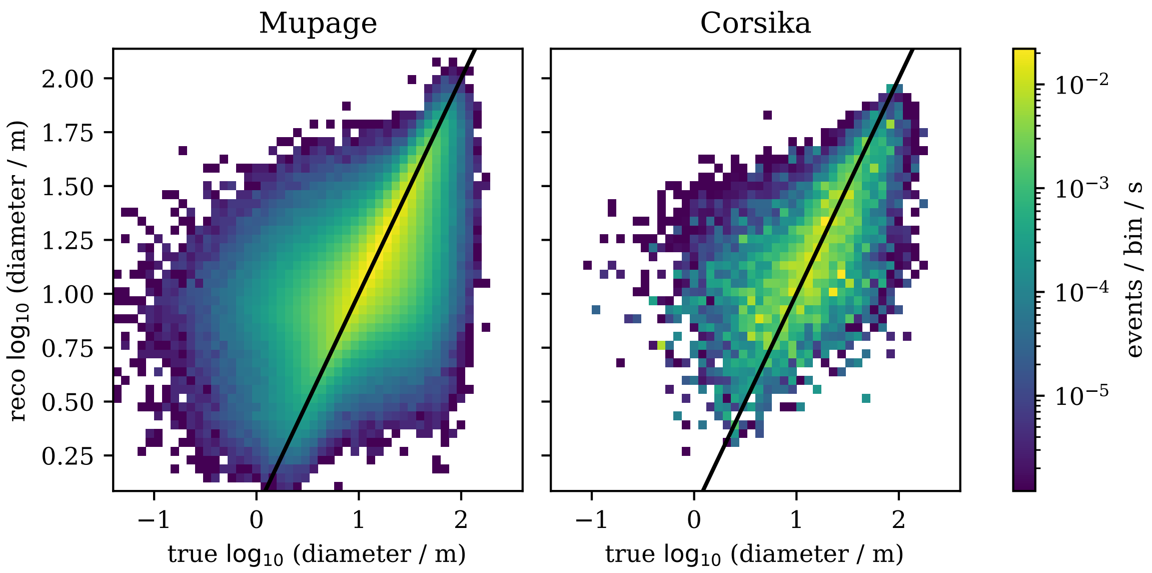


Figure 8.5.: 2D histograms of the bundle diameter reconstruction. Shown is the reconstructed versus the true bundle diameter, using bundles simulated with Mupage (left) and Corsika (right). Bins below the minimal rate of the colorbar are set to the same dark blue color, or to white if if they are below 20% of the minimum.

statistics of the Corsika dataset lead to much higher fluctuations of the rate between bins compared to Mupage. Apart from that, the distributions are similar, and the events are scattered predominantly around the truth in both cases.

A more detailed look at the performance of the reconstruction is given in Figure 8.6. For this, the relative error of the reconstruction is plotted over the true diameter. Similar to what was shown in Figure 7.16, the curves represent the median of the distribution in each bin. However, the fact that the events in Corsika are weighted has to be taken into account when calculating the median. For this reason, a weighted variation of the median is used. Given a distribution of values $\{x_1, x_2, ..., x_N\}$ in a bin, it is defined as the value x_m for which the summed up weights of events smaller than x_m make up half of the total sum of weights. As there usually is no value that fulfills this criterion exactly, an interpolation is used instead. In order to remove events for which the diameter could not be properly reconstructed, a cut on the reconstruction quality $\sigma_{\text{diameter}} < 0.15$ is used. This is the cut referred to as the high reco quality cut introduced in subsection 7.3.2, which leaves 44% of bundles for Mupage and 41% for Corsika. The slight deviation in the fraction of remaining events between the simulation programs is caused by the different distributions of the true multiplicity and the diameter (see Figure 4.3). On both Mupage and Corsika, a stable plateau in the performance is visible between 5 and 40 meters, for which the diameter can be reconstructed down to about 15%.

In Figure 8.7, the distributions of the reconstructed diameter are compared between Mupage, Corsika and measured data. The cut on the reconstruction quality $\sigma_{\text{diameter}} < 0.15$ is used here as well, as it removes most of the single muon events encountered in data. Within the relatively large uncertainty of a factor of about two, Corsika shows decent agreement with Mupage and the measurement. It is noticeable that the bins with a reco log$_{10}$ diameter below 0.5 and above 2 are empty for Corsika, but not for Mupage and data. This is caused by the limited statistics of the Corsika dataset, especially for events with exceptionally large or small diameters.

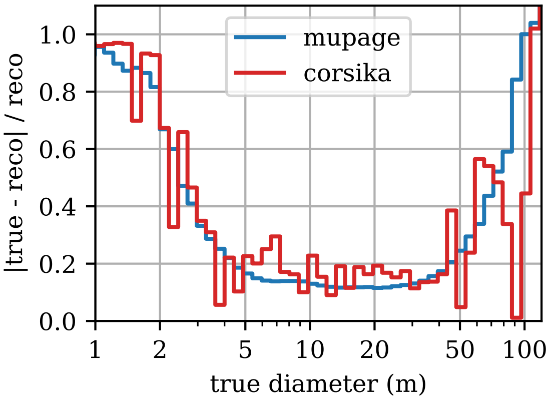


Figure 8.6.: The weighted median of the relative error of the diameter reconstruction plotted over the true diameter, similar to Figure 7.16. The curves show events simulated with Mupage (blue) and Corsika (red). Only events with $\sigma_{\text{diameter}} <$ 0.15 are used for the plot.

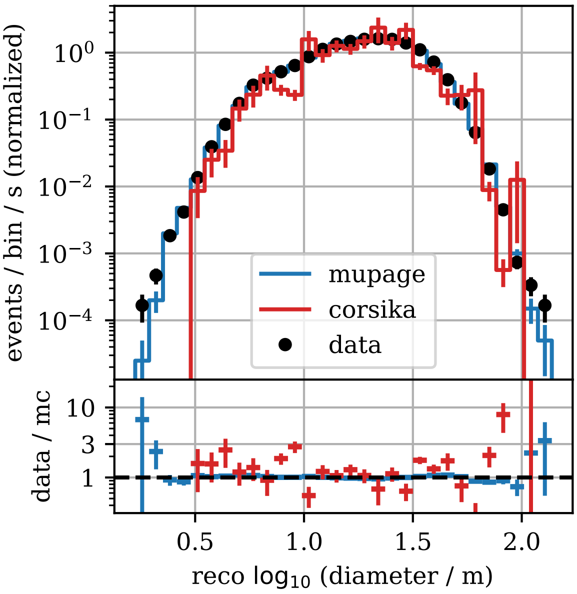


Figure 8.7.: Comparison of the reconstructed bundle diameter on Mupage (blue), Corsika (red) and data (black), similar to Figure 7.23. Only events with $\sigma_{\text{diameter}} < 0.15$ are used for the plot.

8.2.3. Multiplicity

As mentioned before, the muon multiplicity of an event in this work is defined as the number of visible muons in the bundle. A muon is assumed to be visible if it produces ten or more McHits in the detector. In Mupage, events are simulated with up to 100 muons, which leads to a maximum visible muon number of about 80. In the Corsika dataset, bundles with up to 2500 muons are simulated, corresponding to a maximum visible multiplicity of about 2000. However, events with such a high muon number occur at an extremely low rate, and therefore have little influence on the performance at lower multiplicities.

In Figure 8.8, the reconstruction of the network is compared to the truth for both Mupage and Corsika. The plots show a high degree of agreement, all the way up to the highest multiplicities that have been simulated in Mupage. As before, bins with a rate lower than 20% of the minimum that can be encountered in Mupage dataset are set to white. Therefore, most of the events with a muon number above 80 would not be visible in the Corsika plot and therefore not part of the depicted interval of true multiplicities.

The performance as a function of the reconstructed multiplicity is investigated to a closer degree in Figure 8.9. In order to make the datasets more comparable, events with more than 100 simulated muons were removed from the Corsika set for this plot. The curves are on top of each other for most of the interval, but a slight deviation is visible at low multiplicities: The average relative error in Corsika is a few percent lower for single muon events, but a few percent higher for events with two muons.

This is because the network was observed to very slightly overestimate the true muon multiplicity on Corsika for lower muon numbers. In other words, in Corsika there are fewer two

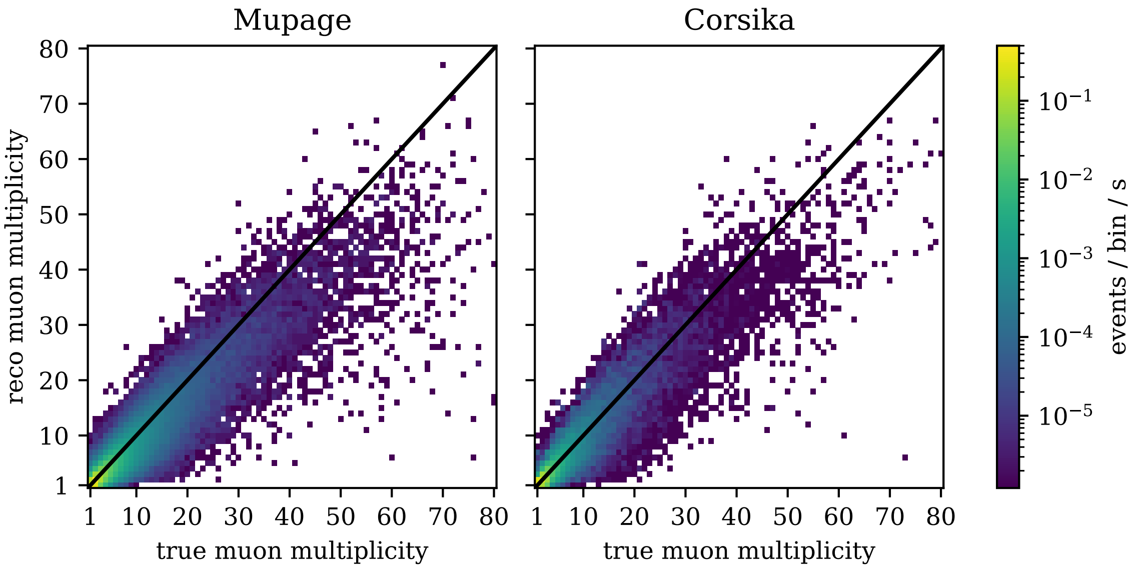


Figure 8.8.: 2D histograms of the muon multiplicity reconstruction. Shown is the reconstructed versus the true multiplicity, using events simulated with MUPAGE (left) and CORSIKA (right). Bins below the minimal rate of the colorbar are set to the same dark blue color, or to white if if they are below 20% of the minimum.

muon events wrongly reconstructed as one muon, and more single muons events wrongly reconstructed as two. The reason for this is not fully clear, but as can be seen in Figure 8.10, muon bundles have a slightly higher median energy in CORSIKA as compared to MUPAGE. Since the network was trained on MUPAGE, it might sometimes wrongly interpret the higher muon energy in CORSIKA as an additional muon, leading to the observed overestimation. Another potential reason is the strong difference in the distribution of the muon multiplicity between the MUPAGE and CORSIKA dataset. A bias acquired during the training on the MUPAGE set could lead to a slight degradation of the performance, especially in the low multiplicity region where the distribution falls steeply. The difference in the distributions is discussed in greater detail at the end of this section, and shown in Figure 8.12. In upcoming productions of MUPAGE, alternative distributions of the muon number will be provided, which can be used to test this. Alternatively, the observed overestimation might also merely be an artifact caused by the low statistics of the CORSIKA set, which are especially noticeable in the proton-dominated low multiplicity region. Future CORSIKA productions with improved statistics and updates might therefore resolve this issue.

Since the network was never trained on events with a true muon multiplicity above 80, it is interesting to see how the algorithm performs on this type of events in the CORSIKA dataset. In Figure 8.11, the mean reconstructed multiplicity is plotted as a function of the truth. As explained, the MUPAGE curve ends at a muon number of 80, as no events with a multiplicity above that are part of the dataset. The average reconstruction is significantly below the truth at that point, which is caused by the bias that the network put in place to make use of the steeply falling distribution of the muon multiplicity (see subsection 7.4.2). Consequently, the bias is not visible in plots like Figure 8.9, which show the performance as a function of the reconstructed value.

On CORSIKA, the curve can be extended well beyond a true muon number of 80. In this region, the network is forced to extrapolate its reconstruction, which can be seen to result in a plateau around the highest values it predicted on MUPAGE. For true multiplicities above 120, there is even a slight reduction visible in the mean reconstructed muon number. However, this

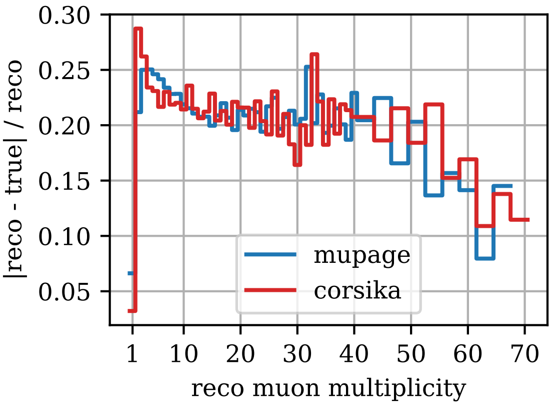


Figure 8.9.: Average relative error over the reconstructed muon multiplicity, similar to Figure 7.28. Only events with less than 100 simulated muons were used for this plot.

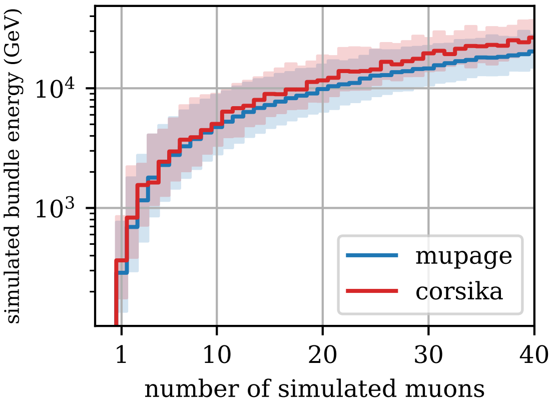


Figure 8.10.: Mean bundle energy over the number of simulated muons. The solid line is the weighted median, and the shaded region indicates the 68% interval around it. The bundle energy is defined here as the sum of the energies of all muons in the bundle.

effect is not only caused by the limited distribution of the multiplicity in the training dataset. As mentioned in section 7.1, only up to 2000 hits are handed to the network for each event. While this has a relatively small impact on the events from Mupage, this approach can severely limit the information contained in events with very high multiplicities. For future iterations of the algorithm, the number of hits should therefore not be limited. Since events with a very high multiplicity are extremely rare, this will have only a negligible effect on the runtime of the network for an average event, but could prove useful for studies on Corsika datasets.

The distributions of the reconstructed muon multiplicity on Mupage, Corsika and measured data are compared in Figure 8.12. In order to remove events triggered by noise, the cut on the reconstruction quality of the zenith angle introduced in subsection 7.2.3 is used. While Corsika does not perfectly match the distribution on data, the agreement is a lot better than with Mupage across the entire multiplicity interval, and is very close to one for events with ten or less muons. Both simulations reach the highest difference between data and Monte Carlo in the range of 40 to 50 muons. For Corsika, the rate on data in that region is lower by a factor of about two, compared to a factor of five to eight for Mupage. A potential explanation for the remaining observed disagreement between Corsika and measured data could be an incorrect description of the primary mass composition of the cosmic rays. While the GST-3 model was used for the analysis so far, multiple different flux models exist, which can have a strong influence on the expected multiplicity distribution. Several different alternative flux models are discussed and investigated in subsection 8.4.1.

8.3. Connection between bundle observables and the primary mass

Even though the properties of the primary particle, like its mass or energy, cannot directly be measured in the detector, various features of the observable atmospheric muon bundles have a correlation to them. In Figure 8.13, the rate of simulated Corsika events is plotted over the muon multiplicity as reconstructed by the graph neural network. In order to evaluate the connection to the primary mass, events are plotted in separate curves depending on the type of primary that caused the air shower they originate from. Since five different primaries were

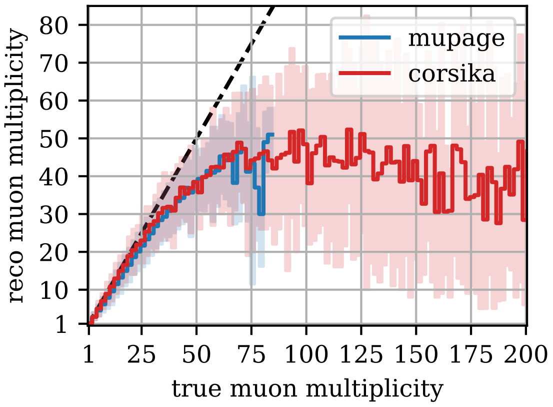


Figure 8.11.: Reconstructed versus true muon multiplicity for Mupage (blue) and Corsika (red). The solid line is the weighted median of the reconstruction for each true bin, while the colored region is the 68% interval around the median.

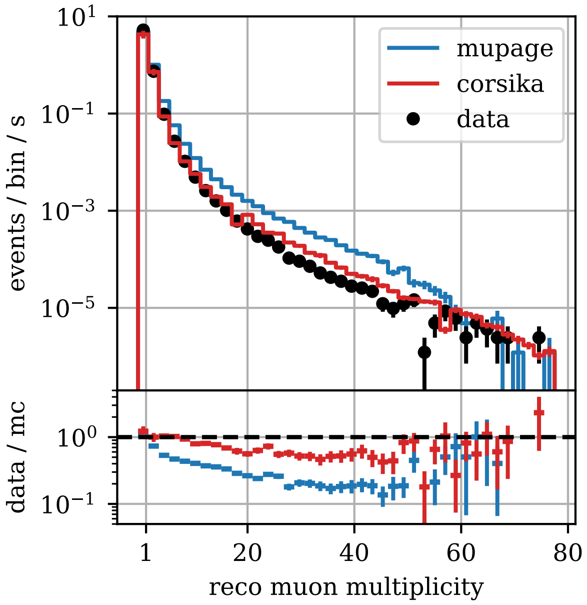


Figure 8.12.: Data-Monte Carlo comparison of the reconstructed muon multiplicity for Mupage (blue) and Corsika (red). Just like in Figure 7.31, only events with $\sigma_{\text{zenith}} < 0.16$ rad are used, which are often triggered by noise.

simulated in the Corsika dataset used, there are also five different curves ranging from light (proton) over medium (helium, carbon, oxygen) to heavy (iron) cores.
As can be seen in the plot, the distributions vary significantly between the different primary masses:

- The rate of proton events is dominant only in the left-most bin with a multiplicity of one and two, and drops steeply afterwards.
- Helium events become dominant in the low multiplicity range between two and ten.
- Carbon and oxygen exhibit a very similar distribution, which makes it difficult to separate them from each other based on the multiplicity alone. Their rate is never dominant in any region of the reconstructed multiplicity, so their contribution to the shape of the total rate is smaller than that of the other primaries.
- Iron events replace helium as the dominant nucleus at multiplicities higher than 15. At the highest muon numbers of 40 or more, the flux consists almost completely of iron events.

The curves in Figure 8.13 show events of all primary energies. As explained in subsection 2.1.3, the elemental composition undergoes drastic changes depending on the primary energy. Multiple different models exist, which differ significantly in their predictions for the flux of the various primaries, especially at higher energies beyond the knee. So fat, the cosmic ray particle flux model used to weight the Corsika simulations has been the GST 3-gen model. As can be seen in Figure 8.14, the flux of this model shows a correlation between the energy and the mass of primaries, as heavier nuclei tend to become dominant at higher energies. Similarly, heavier nuclei also become dominant at higher multiplicities, so there is a correlation between the primary

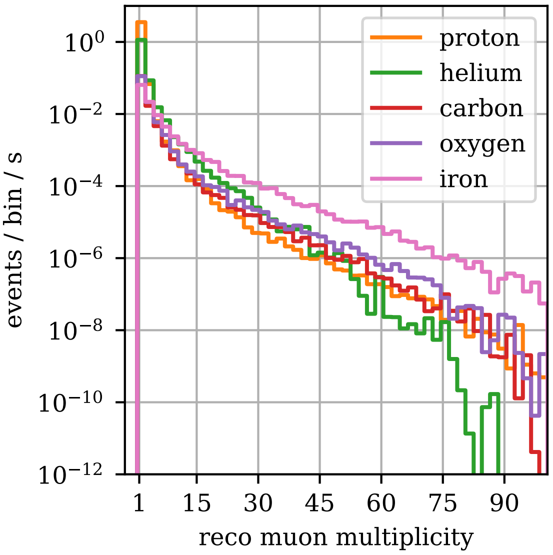


Figure 8.13.: Rate of Corsika events over the reconstructed muon multiplicity. Each curve shows the events produced by air showers that have been induced by the given primary particle.

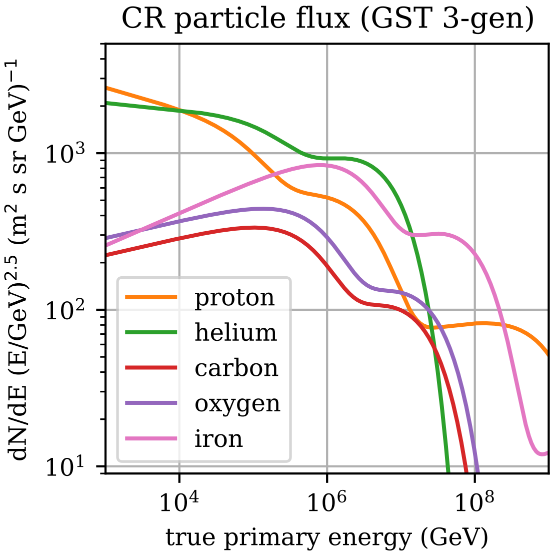


Figure 8.14.: Flux of cosmic ray particles according to the GST 3-gen [55] model plotted over the true primary energy.

energy and the muon number as well. As a consequence, the features of the muon multiplicity distributions of the individual primaries described above can also be encountered to some extent in the primary flux.

The muon number is not the only observable that is connected to the primary composition. In Figure 8.15, the distribution of the reconstructed zenith angle of muon bundles is plotted for the different primaries. Proton is dominant for events going close to straight down with cosine zenith angles above 0.8 (i.e. zenith angles $<$ 37 degrees). The contribution of helium increases when looking closer to the horizon, while even heavier nuclei start to become increasingly dominant at cosine zenith angles below 0.3 (i.e. zenith angles $>$ 73 degrees). This behavior is likely caused by the aforementioned correlation between the primary mass and their energy. Muons that arrive close to horizontally have to travel through more of the Earth's atmosphere and sea before reaching the detector. The additional energy loss along the way means that muons need a higher initial energy on average in order to be measured by the detector, which they are more likely to have if they have been produced in the air shower of a highly energetic primary.

The bundle diameter also exhibits a connection to the primary type. In Figure 8.16, the diameter of events with two or more muons originating from showers produced by proton and iron nuclei is plotted. At low diameters, proton is dominant, while iron induced events become the majority above diameters of about 30 meters. However, it is difficult to quantify the diameter of this transition exactly, as low statistics of the Corsika simulations lead to large fluctuations in the rates and spikes in some of the bins. This is despite the overall flux at 50 meters being at about 10^{-3} events per second. In contrast, the statistics for the muon multiplicity are still sufficiently high even at muon numbers of 75 or more, with rates of about 10^{-6} events per second (see Figure 8.13). It is unclear whether an appropriate adjustment of the weighting during the simulation could reduce the fluctuations and thus improve the feasibility of estimating the primary mass with the bundle diameter. However, a larger detector can improve the statistics even further and likely provide a better reconstruction of events with large diameters. The four line detector used for this work has a diameter of about 40 meters, while the finished 115 line

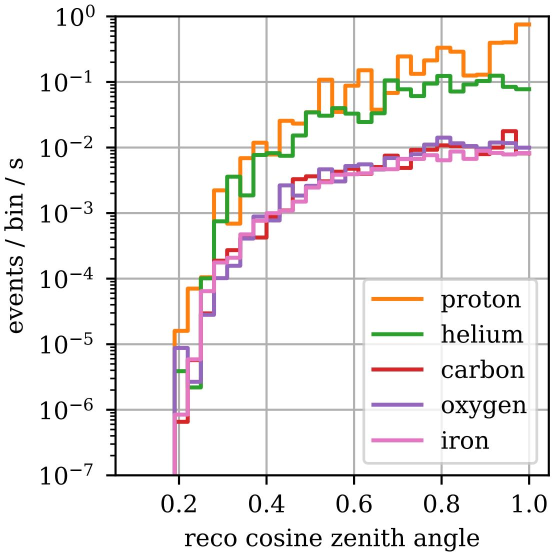


Figure 8.15.: Rate of CORSIKA events over the reconstructed bundle zenith angle. Each curve shows the events produced by air showers that have been induced by the given primary particle.

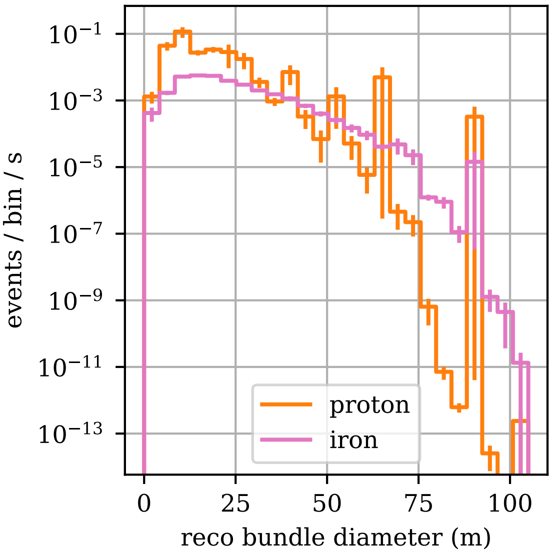


Figure 8.16.: Rate of CORSIKA proton and iron events over the reconstructed bundle diameter. Only events with more than one muon are used for this plot.

setup will span more than 200 meters across. This is well in the region in which events from iron induced showers are dominant, so this observable will become increasingly important for future studies.

8.4. Composition measurement

8.4.1. Muon multiplicity and zenith angle

It was shown in the previous section that one of the most important observables for estimating the primary composition is the muon number. As can be seen in Figure 8.17, heavier primaries tend to produce events with higher multiplicities on average. This correlation can be exploited to make a statistical statement about the composition given the measured distribution of the muon multiplicity on data.

The data-Monte Carlo comparison in Figure 8.17 shows that there is a significant difference between the distribution of the reconstructed muon number on measured data and simulations. The agreement can be improved by increasing or decreasing the fraction of the individual primaries in the total flux. In a logarithmic plot like Figure 8.17, this scaling corresponds to shifting the distributions of the primaries up or down.

As can be seen in the plot, the contribution of carbon and oxygen induced events to the overall event rate is small compared to other primaries over the entire range of reconstructed multiplicities. For this study, they are therefore not needed in order to describe the observed distribution of the muon number and are not considered for the measurement. This leaves only the three primaries proton, helium and iron, which are dominant in the low, medium and high multiplicity range, respectively. In Figure 8.18, the distribution of the reconstructed muon number of the CORSIKA simulation with three primaries is compared to what is measured on data. Each of the three primaries is assigned a scale s_p, s_{he} and s_{fe} with an initial value of 1. These are the free variables in the least squares fit that matches the sum of the three primary

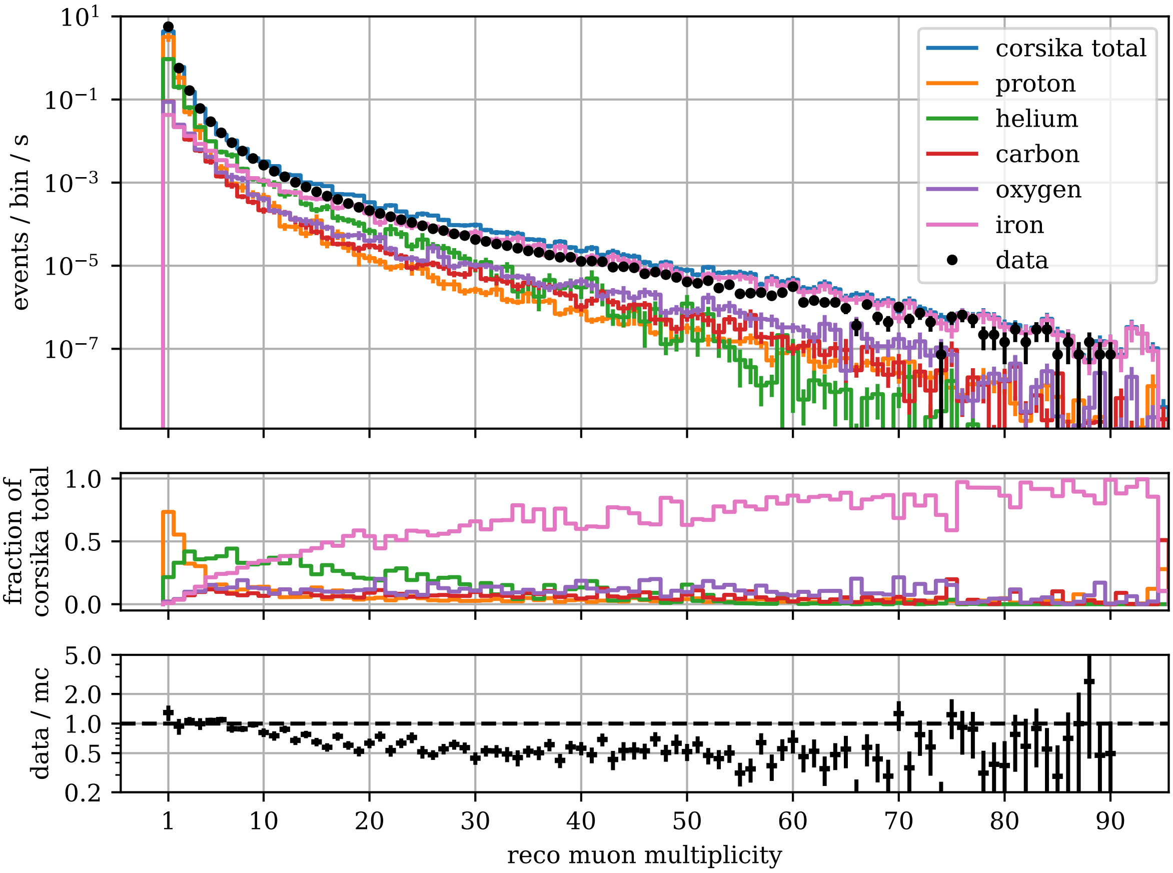


Figure 8.17.: Data-MC comparison of 6 months of ORCA4 line data versus the CORSIKA simulated dataset. The top plot shows the distributions of the reconstructed muon multiplicity for data, CORSIKA, and the five individual primaries that make up the CORSIKA dataset. The middle plot gives the fraction that each primary contributes to the total flux in CORSIKA, depending on the reconstructed multiplicity. At the bottom, the ratio between CORSIKA total and measured data is shown.

curves to the distribution on measured data. The quantity which is minimized in the fit is r^2, the sum of the squared weighted residuals $\delta_i^2 \ / \ \sigma_i^2$ over all the bins M in the histogram:

$$r^2 = \sum_{i=1}^{M} \frac{\delta_i^2}{\sigma_i^2},$$

$$\delta_i = n_{data,i} - \sum_{prim} s_{prim} \cdot n_{prim,i},$$

$$\sigma_i = \sqrt{\sigma_{data,i}^2 + \sigma_{MCtotal,i}^2}.$$

In order to calculate the residual δ_i between the fitted simulation and the measured data for a bin i, the bin counts of the individual primary curves $n_{prim,i}$ are multiplied with the corresponding scale s_{prim}, summed up, and then subtracted from the bin count on data $n_{data,i}$. δ_i is then divided by its uncertainty σ_i, which is calculated using a propagation of errors from the uncertainties of the bin counts $n_{data,i}$ and $n_{MCtotal,i}$. As before, the uncertainty of the bin count is given by the standard deviation of a Poisson distribution with the expectation value of the observed bin count.

The above calculation yields the weighted residuals $\delta_i \ / \ \sigma_i$ for each bin i. The squared sum over

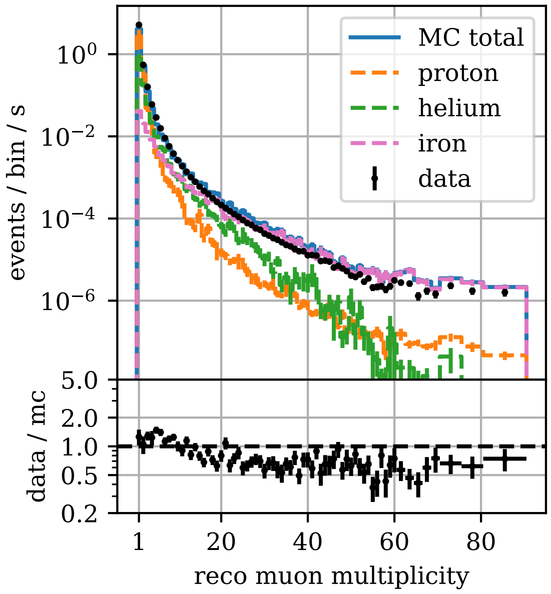


Figure 8.18.: Data-MC comparison of the reconstructed muon multiplicity using the GST 3-gen weights, before any fitting is done.

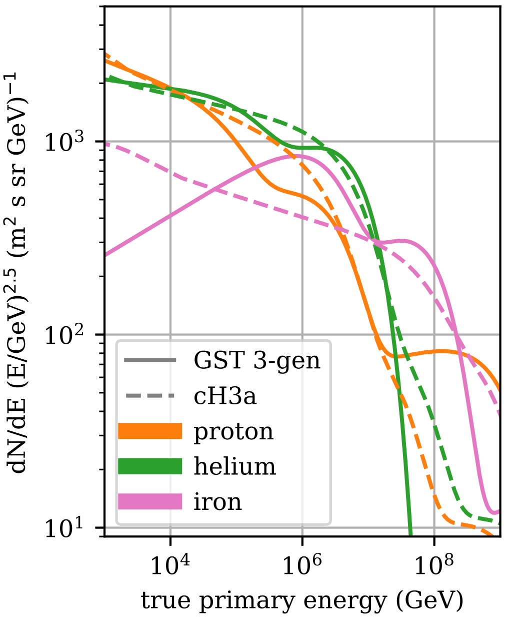


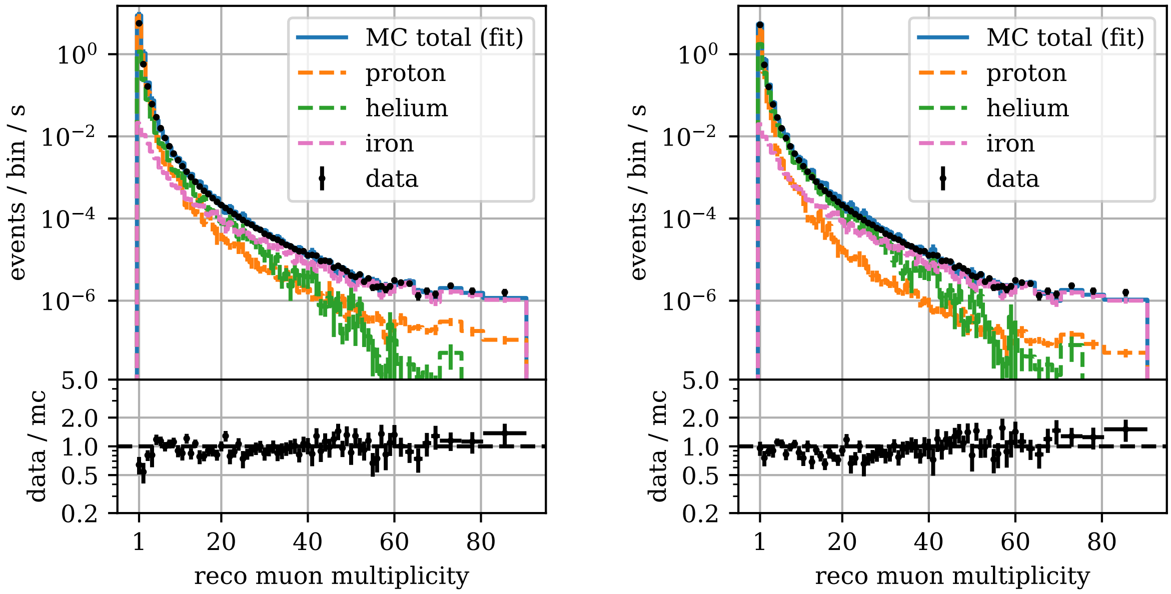
Figure 8.19.: Cosmic ray particle flux over the true primary energy. The GST 3-gen model is shown as the solid lines, while the cH3a model is plotted with dashed lines.

all bins results in r^2, which is minimized with respect to the scales s_{prim} using the Levenberg-Marquardt algorithm [107]. In addition to the fitted scales, an estimation of their covariance can be obtained from this as well. The squared diagonal entries of the covariance matrix are used to estimate the standard deviation of the fitted scales. It is important to note that the uncertainties σ_i are dominated by the uncertainty from the simulations up to muon numbers of about 60. So even though the detection rates of muons on data are high at low multiplicities, low statistics of the Corsika dataset in this range prevent a precise measurement of the proton and helium component.

The result of the fit is shown in Figure 8.20a. In comparison to the situation before the fit shown in Figure 8.18, the data-MC agreement is improved significantly for multiplicities above four. At the lowest muon numbers, the fit produces a severe overestimation of the rates due to an overly large proton fraction. This is likely caused by the low Corsika statistics leading to large uncertainties per bin, in combination with the fact that the proton curve is essentially constrained by only two bins (i.e. events with one or two muons). Starting at muon numbers as low as three, helium replaces proton as the main contribution to the overall flux (see Figure 8.17).

A strategy to mitigate this problem is to make use of the zenith dependence of the primary composition shown in Figure 8.15. Proton events are encountered more often than those from other primaries close to the vertically downward direction. The fit procedure described above can be conducted for two zenith intervals at the same time. For this, the events are split in two groups, one each for events above or below a reconstructed cosine zenith angle of 0.8. Several thresholds were tested, and this value was found to work best in practice. The result is two sets of histograms, for which a combined fit can be done like described above in order to better constrain the proton curve. For this, the sum of squared weighted residuals is calculated over the bins in both histograms, and then minimized using the Levenberg-Marquardt algorithm as before.

This will also provide additional bins in which helium is dominant, further contributing to the



(a) Fit done using all events independent of their reconstructed zenith angle.

(b) Fit done using two zenith bins for events above and below a reconstructed cosine zenith angle of 0.8. An improvement of the data-MC agreement is visible for low multiplicities.

Figure 8.20.: Result of the least squares fit of the reconstructed muon multiplicity distributions from CORSIKA to data. The curve of each primary is multiplied by the corresponding fitted scale. Both plots only show events that have an uncertainty of the zenith angle reconstruction σ_{zenith} below 0.16 rad.

overall stability of the algorithm. In the future, larger event numbers in the CORSIKA simulation at low multiplicities could stabilize the fit and make this procedure unnecessary. The result of the fit using the two zenith bins is shown in Figure 8.20b. Compared to the previous approach without the split in the zenith angle (Figure 8.20a), the data-MC agreement is improved for low muon numbers, but very slight worse in the 10 to 30 range. In Table 8.1, the fitted fractions as well as the correlation matrices of the fitted parameters are compared between the approaches with and without using the zenith angle. Since the proton and helium contribution are anti-correlated, the improved lower proton fraction of the two bin fit also results in a higher helium fraction as in the one bin fit, and a slightly reduced contribution for iron. The table also shows the energy-integrated composition of the GST 3-gen model. Compared to it, the two bin fit shows a similar proton fraction, but a significant increase of the flux for helium and reduction for iron. However, since these values are integrated over the entire energy range, it is not possible to determine in which energy region the differences occur.

Energy dependence

It is worth noting that the described fit procedure does not change the dependence of the flux on the energy, as it still uses the parametrization of the GST 3-gen model for that (see Figure 8.21). The fit merely changes the overall normalization of the individual primaries or, in other words, the spectra in Figure 8.21 are shifted up or down, but keep their shape. However, it is very useful to measure the composition not only for the integrated flux, but for different energy intervals, as the primary composition changes significantly over the range of the cosmic ray energy spectrum.

A first look into how the energy dependence influences the fit can be obtained by varying

Table 8.1.: Results of the composition measurements which do not use the zenith angle (one bin), and which make use of two zenith bins to improve the fit. Both use the GST 3-gen model for the energy dependence of the flux.

Fractions	proton	helium	iron
default GST 3-gen	73%	25%	2.15%
fit (one bin)	140% ± 25%	24.8% ± 2.7%	0.87% ± 0.05%
fit (two bins)	68% ± 15%	38% ± 3%	0.83% ± 0.05%

Correlation (one bin)	proton	helium	iron
proton	0.18	-0.05	-0.0021
helium		0.019	-0.0007
iron			0.0007

Correlation (two bins)	proton	helium	iron
proton	0.07	-0.026	-0.0009
helium		0.02	-0.0012
iron			0.0009

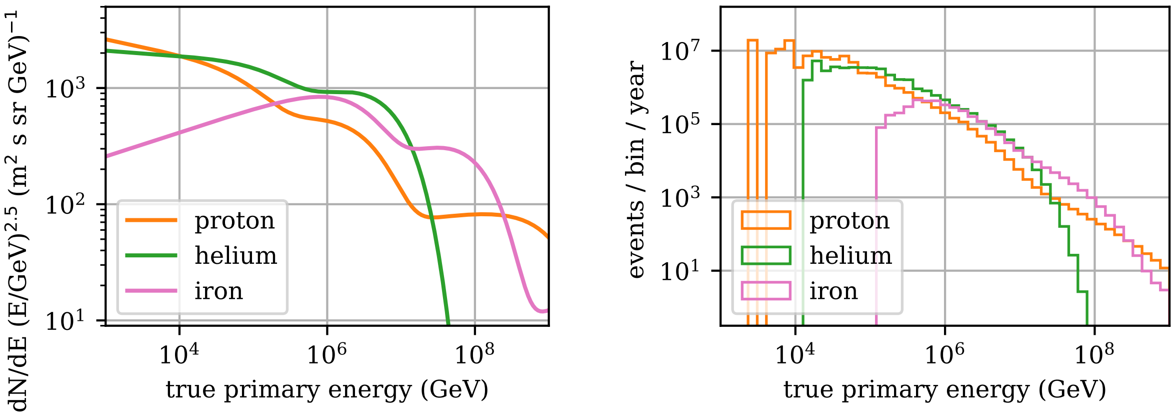


Figure 8.21.: Cosmic ray particle flux over the true primary energy using the GST 3-gen model. The left plots shows the flux of primaries arriving at Earth according to the formulas of the GST 3-gen model. The right plots shows the rate of atmospheric muon events produced in showers of the corresponding primary at trigger level using the Corsika simulations.

the flux model of the Corsika simulation. This can easily be done at trigger level without the need to simulate any additional events, as the flux model can be applied to each event using the `weight_w2` parameter of the Corsika simulation, together with the primary type and energy. The Python package `cr_flux`[1] contains the functions of several flux models, and can be used to conveniently re-weight the existing dataset. The models can differ substantially, especially at higher energies. A selection of these models is summarized in the following:

- The **poly-gonato** model [54] proposed in the early 2000s provides a comparatively simple description of the primary spectra by parametrizing them with powerlaws and a cutoff depending on their gyroradius (see subsection 2.1.4). This model was only intended to describe the spectrum up to and at the knee.
- **GST 3-gen** [55] assumes three populations of cosmic rays, representing supernova remnants below the knee, galactic sources above the knee, as well as extragalactic sources at the ankle. Each population contains a mixture of five different nuclei (proton, helium, carbon, oxygen, iron). Similar to poly-gonato, the nuclei in each population exhibit a cutoff depending on their gyroradius. **GST 4-gen** is an extension to GST 3-gen with an additional fourth population of protons at high energies of about 10^{10} GeV. The model parameters were constrained via a fit to the measured low-energy flux of individual primaries from CREAM, as well as the total cosmic ray flux measured by several experiments at higher energies.
- **H3a** and **H4a** [56] (sometimes referred to as HGm and HGp) follow a similar idea as the GST 3-gen model, as they describe the cosmic ray spectrum with three populations of particles. However, the nuclei species used in the model are different (proton, helium, carbon-nitrogen-oxygen, magnesium-silicon, iron). H4a is a variation of H3a which has only protons in the third component.
- **cH3a** and **cH4a** [16] combine the Gaisser Honda (GH) model [108] for lower energies below 10^4 GeV with H3a and H4a for higher energies.

In Figure 8.19, a comparison of the flux according to the GST 3-gen model used for the Corsika simulations so far, and the cH3a model is plotted. The models differ especially at energies beyond the knee and for iron nuclei. Using the cH3a model for the simulations, a composition measurement with two zenith bins like described above can be conducted (see Figure 8.22). The lower proton flux at high energies in cH3a is reflected by the smaller proton contribution at high multiplicities. The data-MC agreement is decent after the fit, showing a slight improvement in the 10 to 30 multiplicity range compared to the fit with the GST 3-gen model in Figure 8.20.

After the fit, the measured event rate for each primary can be calculated by integrating the scaled distribution over the multiplicities. In order to quantify the composition, these rates can be divided by the overall rate on data. The uncertainty is obtained via propagation of error of the uncertainty of the fitted primary flux and of the total flux on data. As the latter is comparatively small, the resulting uncertainty is dominated by the uncertainty from the fit. For example, the fit using the GST-3 model results in a rate of 4.1 ± 0.9 proton events per second. Since the overall rate of triggered events on measured data is 6.0130 ± 0.0007 events per second, the proportion of proton in the composition is given by 68% ± 15%.

The result of the composition fit using all of the flux models listed above is plotted in Figure 8.23 and summarized in Table 8.2. The flatter iron curve of the non-GST models leads to a better quality of the fit and a reduced iron contribution to the overall flux. This suggests that the energy spectrum of these models can better describe the cosmic ray flux as measured by Km3NeT/Orca. Since there is a significant overlap and therefore anti-correlation between the different primaries, a lower rate of iron induced events leads to a slight increase of the rate for helium and decrease

[1] https://pypi.org/project/crflux/

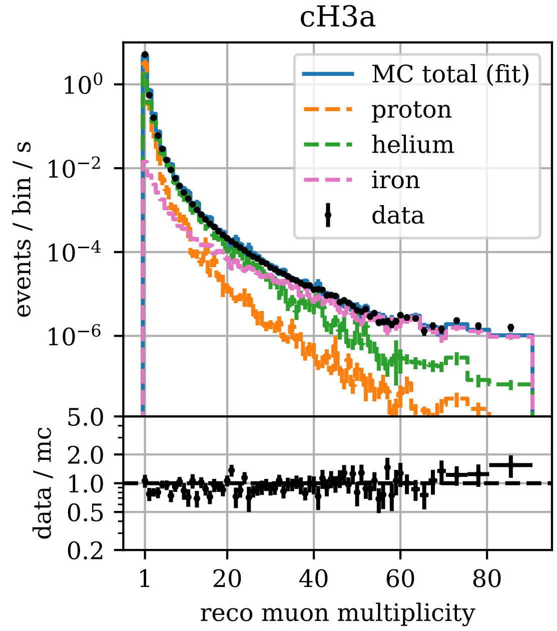


Figure 8.22.: Result of the least squares fit similar to Figure 8.20b, except that the cH3a flux model was used for the energy spectrum of the cosmic rays.

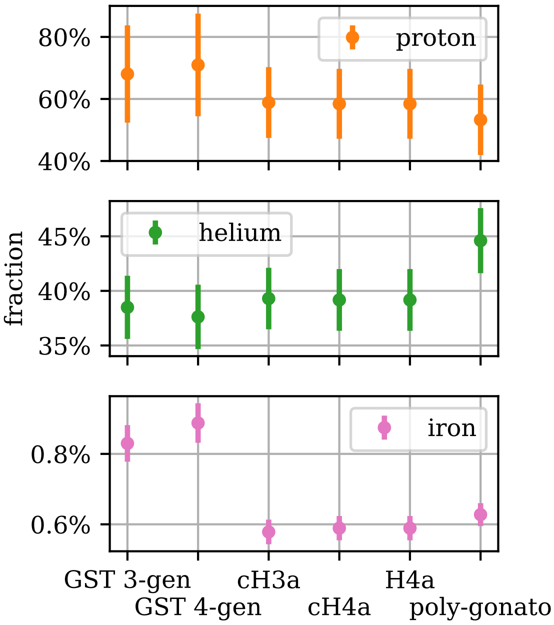


Figure 8.23.: Measured composition of the cosmic ray flux using different models for the energy spectrum. The error bars are obtained from the standard deviation of the least squares fit.

for proton. For future studies, it will be very interesting to measure the energy dependence of the composition directly instead of assuming one of the described models. However, this likely requires a reconstruction of the primary energy, which poses multiple issues and could not be successfully completed within the scope of this thesis. These problems are discussed to greater detail in subsection 8.4.4.

8.4.2. Bundle diameter

Next to the muon number, the diameter of the bundle also has a strong connection to the mass of the primary. In Figure 8.24, a data-Monte Carlo comparison of the reconstructed bundle diameter for proton, helium and iron induced events is shown. The cut on the reconstruction quality of the diameter introduced in subsection 7.3.2 ensures that most of the single muon events are removed from the selection.

flux model	proton	helium	iron	reduced χ^2
GST 3-gen	68% ± 15%	38% ± 3%	0.83% ± 0.05%	1.72
GST 4-gen	71% ± 17%	38% ± 3%	0.89% ± 0.06%	1.75
cH3a	59% ± 11%	39% ± 3%	0.58% ± 0.03%	1.51
cH4a	58% ± 11%	39% ± 3%	0.59% ± 0.03%	1.50
H4a	58% ± 11%	39% ± 3%	0.59% ± 0.03%	1.50
poly-gonato	53% ± 11%	45% ± 3%	0.63% ± 0.03%	1.45

Table 8.2.: Result of the primary composition measurement using different models for the energy spectrum. The reduced χ^2 gives the total squared sum of residuals of the fit divided by the degrees of freedom, and is thus a measure of the quality of the fit.

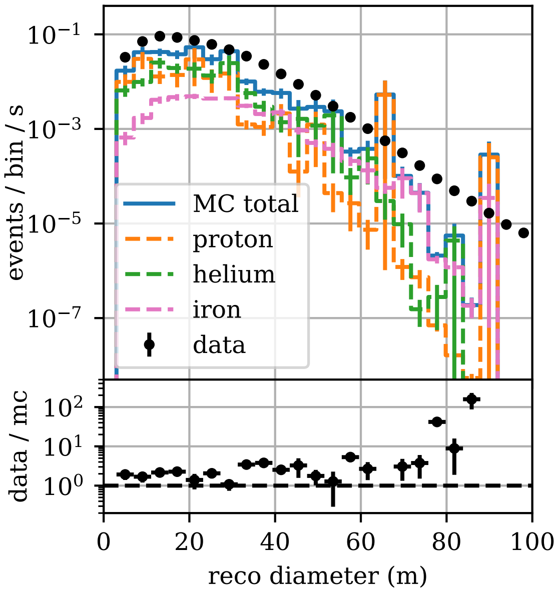


Figure 8.24.: Data-MC comparison of the reconstructed bundle diameter for events with σ_{diameter} < 0.15.

In the region below a diameter of 70 meters, the rate on data is higher than the rate in Corsika by a factor of about two. This is caused by the flux of single muons remaining after the quality cut. They make up about 14% of the total rate, and as mentioned in subsection 8.2.3, the rate of single muons is underestimated by the used Corsika production relative to what is measured.

For events with a diameter of more than 70 meters, the simulated distributions appear to fall off much more rapidly than what is measured on data. While the low statistics in this region make a definitive statement difficult, this could be indicative of an underestimation of the iron contribution to the cosmic ray flux in the simulations, as iron induced events are most common in this interval. Interestingly, this seemingly goes against the result based on the muon multiplicity, which suggests that the iron contribution is overestimated in the simulations (see Figure 8.18). As it turns out, both of these observations are true at the same time, as the excess of iron events is present for events which are reconstructed with a high multiplicity *and* a low diameter.

For this, consider the data-MC comparison of the diameter for events with a low or high multiplicity shown in Figure 8.25. The large excess of events with a data/MC ratio above 100 is present only for events with a high diameter and a multiplicity below 30. For events with a high diameter but a multiplicity above 30, no such increase is visible. High multiplicity events barely affect the overall distribution of the diameter due to their low flux, and high diameter events in turn barely affect the overall multiplicity distribution. Therefore, both effects can be present at the same time. In summary, the iron component in Corsika is underestimated for events with a high diameter and low multiplicity, and overestimated for events with a high multiplicity but low diameter at the same time.

As can be seen in Figure 8.24, heavier primaries tend to become more prominent at higher diameters: The most common primary is proton at diameters below 10 meters, helium between 10 and 40 meters, and iron above 40 meters. The diameter can therefore provide an additional way of measuring the cosmic ray composition, and improve upon results when used in conjunction with the muon number.

Similar to before, the individual primary curves can be scaled using a least squares fit in order to match the bin counts on measured data to the ones on the total simulation. The result from this is a measurement of the composition of the primary cosmic rays. However, no successful

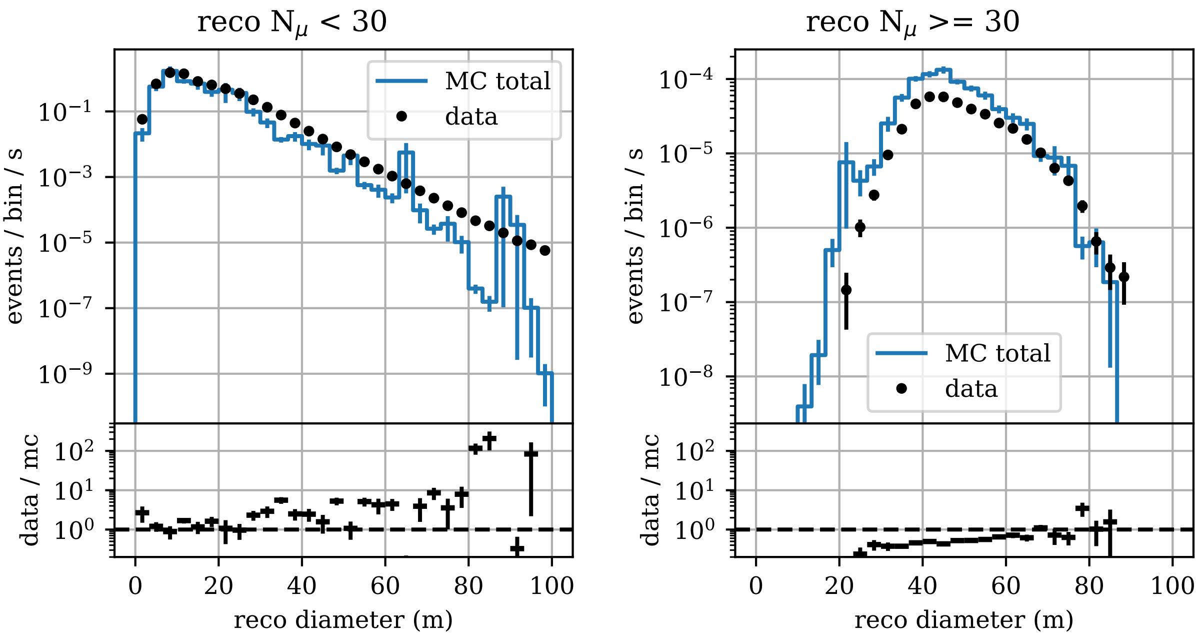


Figure 8.25.: Data-MC comparison of the reconstructed bundle diameter using the Corsika simulations. The plots use only events with a reconstructed muon multiplicity of below (left) or above (right) 30.

attempts could be made to bring the fit to convergence. A combined fit of the diameter together with the multiplicity and zenith angle did not produce satisfying results either. This is likely caused by the low statistics of events with high weights, which results in the large error bars seen in Figure 8.24. While the errors for six months of measured data are smaller than the dots visible in the plot, the fluctuations in the simulated distributions make it difficult to achieve a meaningful separation between the three different primaries. Larger Corsika productions are likely to improve this situation in the future.

8.4.3. Seasonal variation

It has been established by various experiments like Minos [109], Opera [110], Nova [111] and IceCube [112] that the rate of atmospheric muons at the surface and below varies periodically over the course of a year. In [113], an overview over the results from various experiments over the last decades is presented.

This effect is mainly caused by temperature variations of the atmosphere from season to season, which in turn affect the density profile of the air in which the cosmic ray air showers develop [5]. For example, the highest overall rate of single atmospheric muons is typically observed across experiments during the summer, when the temperature is highest. The hot air expands compared to the colder seasons, which reduces the air density and thus makes it more likely for mesons in the hadronic component of the air showers to decay into muons instead of interacting with an air molecule. The opposite effect is visible during the winter, where the rate at which atmospheric muons can be measured is lowest.

Different amplitudes of the variation of the muon rate are observed for the different experiments, typically in the range of a few percent. For example, Opera reports a seasonal change of the single muon rate of 1.55 ± 0.08 % compared to the average [110]. On the other hand, IceCube reports an annual variation of about 8% over the course of the seven years between 2011 and 2017 [112]. A cause for the difference in amplitudes is the locations of the detectors, which come with stronger or weaker annual changes of the air temperature. Additionally, they have higher or lower energy detection thresholds for measured muons due to their different depths. As a consequence, deeper detectors are expected to measure a larger seasonal variation of the muon flux [113].

Several experiments have reported that events with higher muon multiplicities show a stronger seasonal variation in their rate than single muon events. Nova reports an amplitude of the variation of 5.11 ± 0.10 % for events with 15 to 19 muons, and 6.73 ± 0.13 % for events with 39 to 100 muons [111]. Similarly, Minos also has measured a stronger seasonal change of the muon rate for events with multiple muons [109]. Since the muon multiplicity is the main observable used for the cosmic ray composition measurement presented in this thesis, the seasonal fluctuation has to be considered as a source of systematic uncertainty. The goal of this section is to investigate its impact on the presented study.

Interestingly, both Nova and Minos have found that events with multiple muons have the maximum of their rate in the winter, as opposed to the summer maximum of single muon events. The reason for this is still under debate, but a potential explanation is that multi muon events at lower energies could mostly get produced by interactions of mesons in the air shower [113]. On the other hand, single muon events originate primarily from decays of the mesons. Since the air density is lower during the summer and the probability of decay is therefore increased, this would lead to a higher single muon rate, but a reduced multi muon rate. Similarly, the opposite would be true for the winter months.

An unpublished study with the Macro detector has not found a winter maximum for the multi muon rate, and instead measured the maximum to be in the summer just like for single muons [113]. The reason for this could be the large difference in terms of the depth of the detectors: Minos lies at a depth of 225 meters water equivalent (w.e.), while Nova has merely 3.6 meters w.e.. In contrast, Macro has a rock overburden of 3.8 kilometers w.e., being in a similar order of magnitude as the 2.4 kilometers w.e. of the KM3NeT/Orca detector. Due to the larger depths of these detectors, only muons with high energies can be detected. If these high energy muons get produced from meson decays directly even for multi muon events, it would explain why no phase shift was measured with a deep detector like Macro.

Muon rate in KM3NeT/ORCA

In Figure 8.26, the rate of triggered events for six months of Orca4 data between August 2019 and February 2020 is plotted for each run. When using all runs and events, as is shown in blue, significant variations in the rate can be observed from run to run on top of a slowly changing baseline. A cause for these upward and downward spikes is the variation of the rate at which the bioluminescence background in the water triggers the detector. In order to filter out runs with a particularly high background activity, the so-called high rate veto fraction can be used.

If a photomultiplier records hits at a rate higher than a set threshold (more than 2000 hits in a 100 ms period), some of the data it has recorded is deleted due to the high rate veto [114]. The average fraction of time in a run that the photomultipliers in the detector have undergone a high rate veto is called the high rate veto fraction. The source of these high rates is typically background in the form of bioluminescence. Consequently, if the veto was active for a large fraction of the photomultipliers, this is indicative of a high noise background. In order to select runs with a high quality, only physics runs with a high rate veto fraction smaller than 0.1 and a livetime of more than 20,000 seconds are chosen for this study.

Since the goal is to investigate the trigger rate of atmospheric muon events, a cut is used to remove events triggered by background sources instead of a muon. For this, the cut on the reconstruction quality of the zenith angle described in subsection 7.2.3 is used. With this cut, only events that have an uncertainty σ_{zenith} of less than 0.16 rad are added to the muon rate. The trigger rate of events that pass the described selection criteria for both the run and the event quality is shown as the orange curve in Figure 8.26. Compared to the rates calculated from all events shown in blue, the selection is much more stable, and makes the downward trend in the

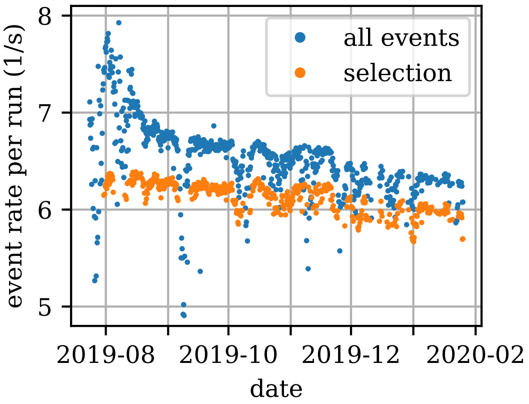


Figure 8.26.: Overall rate of triggered events on measured data in the given time period, for all events (blue) and selected high quality events and runs (orange, see definition in the text). Each dot represents a single run.

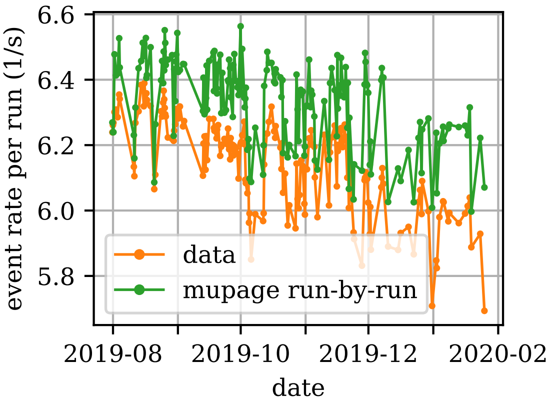


Figure 8.27.: Rate at which selected high quality events and runs are triggered. The orange curve represents the rate on measured data (same as in Figure 8.26), while the green curve uses the MUPAGE run-by-run simulation instead.

rate toward the winter months clearly visible.

However, it is important to note that the observed change in rate over time is not only caused by the variation of the atmospheric density. The bioluminescence background rate decreases in the winter, similar to the rate of atmospheric muons. While the described cut on the zenith reconstruction quality can be used to remove events without a muon track, a higher background rate still has an impact on the measured muon rate. For example, a higher background can let the trigger fire for barely visible tracks that would have gone unnoticed otherwise, or lead to hits being randomly recorded in just the correct place and time to make it seem like they are part of a track.

In order to determine what fraction of the change of the muon rate is caused by air density variations instead of the background, the procedure described by Mulder [115] can be used. It makes use of the fact that the MUPAGE run-by-run simulations include the variation of the background, but not of the atmospheric density. By applying the aforementioned selection criteria on the run-by-run simulations, the change in the muon rate due to the background alone can therefore be estimated. It is important to note that three quarters of the run-by-run simulation were used for the training and validation of the network, and can thus not be used anymore for this comparison. As a consequence, the livetime for MUPAGE runs has to be scaled up according to how many of their events are in the test set, and some runs even have to be removed entirely as all of their events were used during training.

The result of the selection on both measured data and the MUPAGE simulation is shown in Figure 8.27. Jumps in the rate from run to run are present in a similar fashion in both measured data and the simulations, as they are not caused by the varying temperature and are therefore correctly modeled as part of the run-by-run MUPAGE dataset. Apart from the short scale jumps, the overall rate is decreasing toward the winter months for both curves. However, the rate is decreasing more strongly on data. This is because the rate on measured data is not only impacted by the change of the background rate, but additionally also by the change of the air density, which is not part of the MUPAGE simulation. By dividing the rates of the two curves with each other, an estimate of the variation of the muon rate due to the atmosphere alone can be obtained.

This is shown in Figure 8.28. While the muon rate on measured data is about 99% of the rate

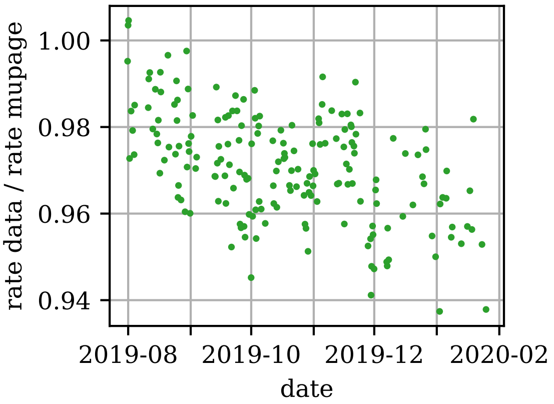


Figure 8.28.: Ratio between the trigger rate of high quality events and runs on measured data and the MUPAGE run-by-run simulation for each run in the given time period. Calculated by dividing the curves from Figure 8.27 by each other. This estimates the change in the muon rate due to seasonal variations of the air density.

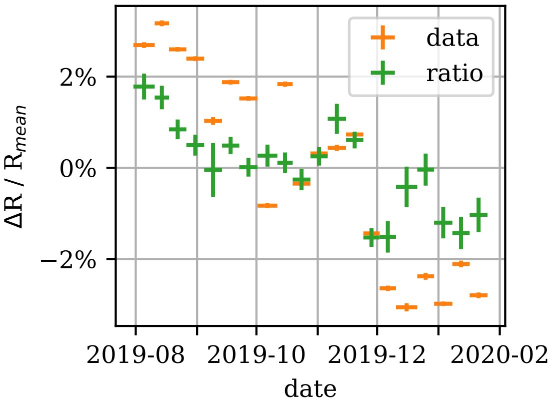


Figure 8.29.: Change in the trigger rate of atmospheric muon events compared to the average over the entire time period. The orange markers show the total change on measured data, while the green markers show the change compared to MUPAGE run-by-run simulations.

in the run-by-run simulations during the summer, this ratio drops to about 95% in the winter. This yearly variation with an amplitude of about ±2% matches the result obtained by Mulder [115] for the ORCA6 detector in the time period between March and December 2020.

In Figure 8.29, the change of the muon rate compared to the average over the course of the given six month period is plotted for 20 time bins. The plot shows both the overall change of the rate for selected events and runs on measured data, as well as the estimate for the change due to the variation in the atmospheric density, obtained from the ratio to the MUPAGE run-by-run simulation. The rate varies with an amplitude of about 3% between August and February (orange dots), from which about 2% are caused by the variation of the atmospheric density (green dots), and the remaining 1% are caused by the bioluminescence background.

The muon rate jumps significantly between the different bins in Figure 8.29. Part of this is caused by the variation of the bioluminescence background. But jumps are still visible even for the green dots which take the background into account. A potential reason for this is that the variation in the muon rate is caused by changes of the air temperature and density, as described above. While the temperature generally decreases towards the winter months, periods with hot or cold weather can have an influence on the atmospheric density on a short time scale. This effect is not captured in the presented analysis, as the time of the year is used merely as an approximation of the air temperature. In order to incorporate these short scale effects, the dependence of the muon rate on the temperature and air density profile can be investigated explicitly in future studies (see e.g. [112] for an example in IceCube, or [115] for a first study in KM3NET/ORCA).

Influence on the composition measurement

In Figure 8.30, the change in the overall trigger rate of high quality runs and events on measured data is shown, depending on the reconstructed muon multiplicity. Events with a higher muon number show a stronger change in the muon rate. The amplitude of the variation reaches from

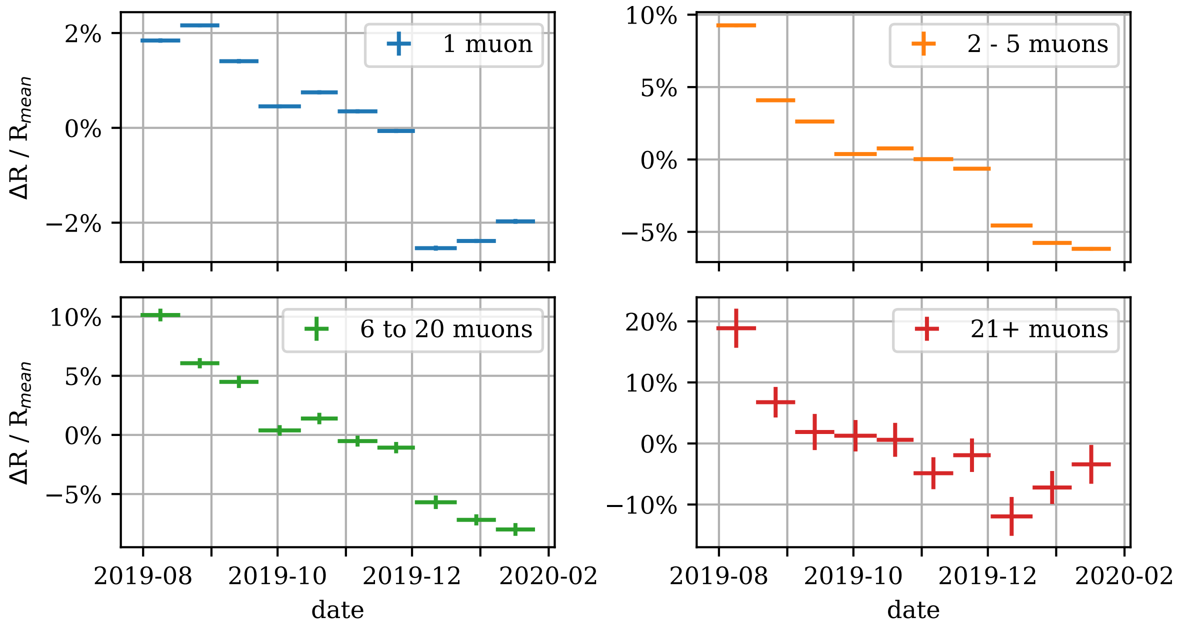


Figure 8.30.: Change in the rate of atmospheric muons compared to the average over the entire six month period on measured data. Each plot shows events with a different number of reconstructed muons. Note the different ranges on the y-axis for each of the plots. The same selection as in Figure 8.26 was applied to the runs and events.

about 2% for single muon events, up to 10% or more for events with more than 20 muons. This variation includes both the change in background as well as the atmospheric air density. Since events with higher multiplicity consist of more hits on average, the change in background is likely to be less important for these events. Consequently, the observed seasonal variation of multi muon events is expected to be mostly caused by the atmospheric density. In contrast, single muon events are expected to be most affected by the background. In the future, this study could be repeated on Mupage run-by-run simulations to determine how big of a fraction of this variation is caused by the change in the air density, similar to the approach discussed above.

The maximum of the muon rate is found during the summer months, for both single and multi muon events alike. A winter maximum of the multi muon rate like it was measured by Nova and Minos is not present in the given data. As described above, it is likely that the winter peak for multi muon events can only be observed in detectors with a significantly lower energy detection threshold than KM3NeT/Orca.

The muon multiplicity is the main observable used for determining the cosmic ray mass composition in this thesis. The given Corsika dataset does not include the seasonal variation of the density profile of the atmosphere or the background rate. This would be necessary in order to respect the significant dependence of the muon number on the time of the year. Thus, the described seasonal variation of the muon rate has to be considered as a source of systematic uncertainty. Its impact on the composition measurement can be estimated by performing the fit using measured data from different time periods over the course of the examined six month period. Apart from that, the measurement is done in the same way as explained in subsection 8.4.1.

Figure 8.31 shows the result of this analysis. The impact of the seasonal variation on the proton rate is negligible given the current substantial uncertainties from the limited statistics of the Corsika set. Protons are dominant in the low multiplicity range where the seasonal change was observed to be the smallest. The impact of the seasonal variation on the proton fraction is therefore expected to be the smallest of all the primaries. On the contrary, iron and in particular helium show significant changes over the year, as they are determined by events with higher

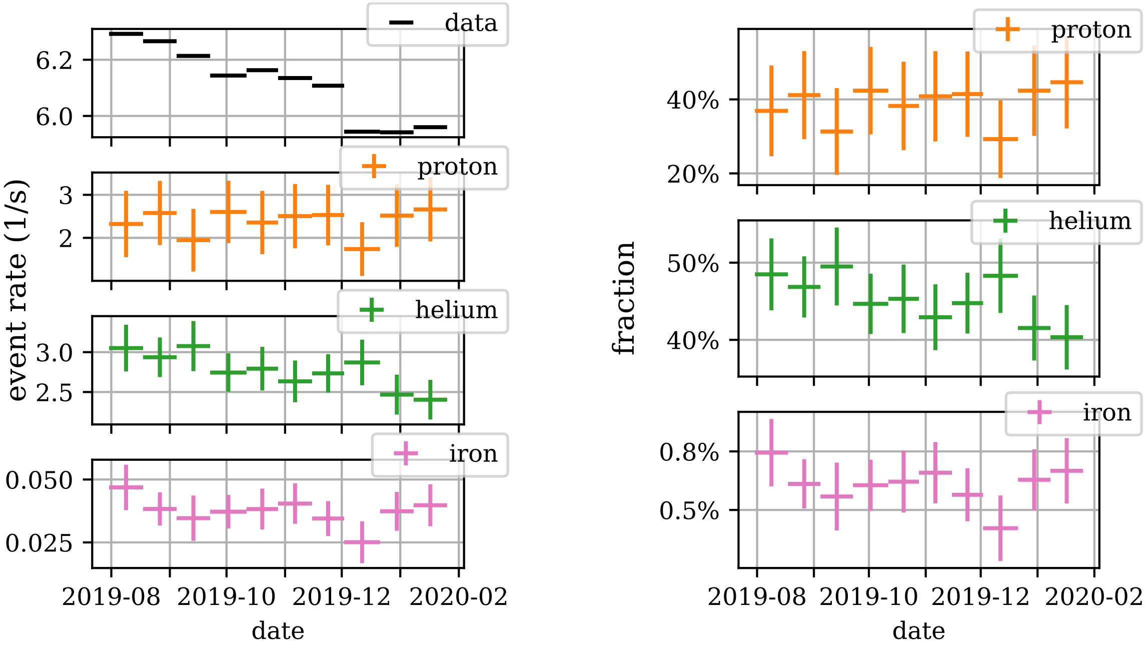


Figure 8.31.: Measured composition of the cosmic ray flux over the time period in which the corresponding events were recorded. The left plot shows the overall rate of triggered events with $\sigma_{zenith} < 0.16$ at the top, as well as the rates of the different primaries below it as measured by the fit. The plot on the right shows the fraction each primary contributes to the overall rate on data. It was calculated using the values from the left plot by dividing the rate of each primary by the overall rate.

muon numbers. Their fraction changes from a maximum of about 50% ± 5% to a minimum of 40% ± 5% for helium and from 0.8% ± 0.2% to 0.4% ± 0.2% for iron between summer and winter. In the likely case that the composition can be determined with a higher precision in the future, dedicated run-by-run CORSIKA simulations for different periods of the year could be necessary in order to eliminate the seasonal variation as a source of systematic errors. These simulations would need to respect both the variation of the background rate as well as of the air temperature and density to most accurately describe the data. However, it is unclear how large the uncertainty from the variation of the atmospheric density is in comparison to the one caused by the different high energy interaction models, which could not be tested within the scope of this thesis.

8.4.4. Discussion

Even though the available statistics of the CORSIKA dataset are small, and the detector is still in an early stage of construction, a fit using the reconstructed muon multiplicity can already be used to measure the composition, and even shows sensitivity to the choice of the energy spectrum. It is shown in subsection 8.4.2 that the bundle diameter is also sensitive to the primary mass. However, the number of simulated events in the present dataset is not enough in order to determine the composition using the diameter with the presented method. With sufficient statistics, the precision of the composition measurement could likely be improved by making use of both the diameter and the multiplicity. A straight forward approach for this is to perform the fit of the primary distributions using the distributions for both these observables simultaneously in a single chi squared fit. This approach is successfully used in this work to incorporate the reconstruction of the zenith angle into the fit. For this, the multiplicity distribution is fitted in two different zenith bins at the same time. This improves the result, as light primaries contribute increasingly little to the overall flux closer to the horizon.

Apart from increasing the statistics available in simulations, a different strategy could be applied to improve the stability of the fit. In this thesis, the composition measurement is conducted using the distribution of observables like the multiplicity on a set of CORSIKA simulations. In some regions of the phase space, like for low multiplicities or moderate to high bundle diameters, the low statistics of the simulations posed a significant problem for the stability of the fit. It could be beneficial to instead follow a different approach for the fit, which has been used by multiple experiments in the past [116][117]. For this, a theory curve is fitted to the distribution on simulations for each primary first. Then, the sum of these theory curves is fitted to data like before. This could eliminate the problem of low statistics in some of the regions. On the other hand, it requires choosing an appropriate function for the theory curves.

In addition to measuring the primary composition, it is very useful to also reconstruct the energy of the primary. By combining both of these measurements, the composition can be determined as a function of the energy. This allows for additional cross checks of primary flux models and interaction models. A strategy for estimating both the mass and energy of the primary is presented by the IceCube Collaboration [116]. For this, a small neural network was trained to estimate the energy and mass of the primary using several lower level observables. As described above, parametrized theory curves were then fitted to the resulting distributions for the mass, which in turn were scaled and fitted to the measured data. The estimate for the primary energy was used to split the data into multiple energy bins, and the composition analysis was performed for each of them. For KM3NeT, the direct observables could be the zenith angle of a muon bundle, the muon number, the diameter, as well as an estimate of the summed up energy of all the muons. While there is currently no established way of reconstructing the bundle energy in KM3NeT, a previous study suggests that this task can be solved very efficiently using a graph neural network as well [118].

Instead of directly reconstructing the primary energy, the dependence of the composition on the energy can also be investigated by adding free variables to the chi square fit. These variables could describe the slopes and cutoffs present in the spectra of the individual primaries, and would be tuned during the fit exactly like the normalization factors described in this chapter. Using the available CORSIKA dataset and its limited statistics, no successful attempt could be made with this strategy so far.

It is worth noting that KM3NeT does not have a way of measuring the energy of cosmic rays independently of the mass, as both of these quantities are reconstructed using the muonic component of air showers. This is different compared to IceCube, which can additionally make use of the IceTop detector to reconstruct the energy from the electromagnetic component. It has been reported that experiments which can only perform a correlated measurement of the energy show a much weaker effect of the muon puzzle on the rate of high energy events with a high muon number [15] (see section 2.1.3 for details on the muon puzzle). If this is correct, KM3NeT/ORCA will likely also exhibit this property.

A source of systematic uncertainty for the measurement of the composition is the high-energy hadronic interaction model used in the CORSIKA simulations (see subsection 4.1.1). The dataset used in this work was simulated with the post-LHC model SIBYLL 2.3c. Alternatives include EPOS-LHC [119] and QGSJET II-04 [120]. The different models can affect observables like the muon number or the bundle diameter significantly [17], and are therefore an important source of systematic errors. However, investigating this requires the production of additional sets of CORSIKA simulations, as the interaction model affects the entire evolution of the air shower. It can thus not be simply changed at trigger level like it is the case with the energy spectrum.

In the presented study, only proton, helium and iron nuclei were used to describe the observed distribution of the muon multiplicity. The two additional primaries that are present in the CORSIKA dataset (oxygen and carbon) could not be used in the fit, as their scaling factors would get set to zero during the least squares fit. This is because their overall rates are too small and their distributions do not differ significantly enough in shape from those of the other primaries in order to be reliably identified. The large uncertainty per bin which is caused by the low statistics of the Coriksa dataset is a potential reason for this. Furthermore, it is likely that the carbon and oxygen component can be better identified as the Km3NeT/Orca detector grows in size and can measure wider bundles more precisely. Other spectral models, like the HXa and cHXa parametrizations, feature different primary groups in addition to proton, helium and iron than the GST models. So in order to be able to switch the spectrum to these models for cross checks, additional primaries would need to be simulated in Corsika.

Chapter 9. Summary and Outlook

Graph networks are especially efficient network architectures for analyzing data recorded with the KM3NeT detectors. The model design used in this work uses the euclidean distances between the space and time coordinates of hits to construct a graph, and applies Edge Convolutional layers on it. While this approach was shown to function well on the given data from the ORCA4 detector, it is not the only available design for graph based models. Other architectures might prove superior, both in terms of performance as well as memory efficiency, which could be especially important for larger detector layouts which result in much larger graphs as an input. A study comparing multiple different architectures is necessary in order to find the one that is best suited for the given data.

The first application of deep learning algorithms on measured data in KM3NeT presented in this work has shown satisfying agreement across the board, suggesting that this method is robust against deviations between simulations and measurements. In future measurements, KM3NeT additionally includes an acoustic positional calibration system, which takes into account the influence of the movement of seawater on the positions of the photomultipliers. It will be interesting to see how this affects the agreement between data and simulations of the graph based approach, and whether dedicated simulations including this movement are necessary to train the network.

Reconstructions of the incident direction, the muon multiplicity as well as the diameter of muon bundles have been developed using these graph based networks within the scope of this thesis. This showcases one of the big advantages of deep learning: Different observables can be reconstructed using a very similar approach, meaning this method is fast and requires little resources compared to traditional approaches. Additional reconstructions can easily be included in the future if need be. For example, another relevant property that can be reconstructed using graph networks is a measure of the total energy of the bundle, which was successfully developed within the scope of a separate Bachelor's thesis [118].

Using the deep learning reconstructions, a measurement of the primary composition was performed for the first time in KM3NeT. It makes use of the correlation between the muon multiplicities of events recorded in the detector and the masses of the primary particles that have produced them. Despite the limited statistics of the available set of simulations, the overall fractions averaged over the entire energy spectrum were determined with a relative statistical uncertainty between 22% for proton, down to as low as 6% for iron. An energy dependent measurement of the composition will allow for a more detailed investigation of the highly energetic particles beyond the knee, whose comparatively low flux makes them contribute little to the overall rate. While no such measurement could be successfully completed within the scope of this thesis, larger sets of CORSIKA simulations and a potentially improved reconstruction quality from a larger detector are likely to make this possible for future studies. For this, an estimator for the energy of the primary is useful. This could be either determined by making use of the lower level reconstructions of the deep learning and classical approaches, or alternatively by training a network on CORSIKA simulations to let it directly predict the energy. However, training a network from scratch requires a large set of samples, which are costly to simulate with CORSIKA.

Bibliography

[1] Julia Becker Tjus and Lukas Merten. "Closing in on the origin of Galactic cosmic rays using multimessenger information". In: *Physics Reports* 872 (2020), pp. 1–98. ISSN: 0370-1573. DOI: https://doi.org/10.1016/j.physrep.2020.05.002. arXiv: 2002.00964 [astro-ph.HE]. URL: https://www.sciencedirect.com/science/article/pii/S0370157320301927.

[2] Markus Ahlers and Philipp Mertsch. "Origin of small-scale anisotropies in Galactic cosmic rays". In: *Progress in Particle and Nuclear Physics* 94 (May 2017), pp. 184–216. DOI: 10.1016/j.ppnp.2017.01.004. URL: https://doi.org/10.1016%2Fj.ppnp.2017.01.004.

[3] M. Ageron et al. "Dependence of atmospheric muon flux on seawater depth measured with the first KM3NeT detection units". In: *The European Physical Journal C* 80.2 (Feb. 2020). DOI: 10.1140/epjc/s10052-020-7629-z. URL: https://doi.org/10.1140%2Fepjc%2Fs10052-020-7629-z.

[4] Alessandro De Angelis and Mário Pimenta. *Introduction to Particle and Astroparticle Physics*. 2nd ed. Springer, 2018. DOI: 10.1007/978-3-319-78181-5.

[5] Thomas K. Gaisser, Ralph Engel, and Elisa Resconi. *Cosmic Rays and Particle Physics*. 2nd ed. Cambridge University Press, 2016. DOI: 10.1017/CBO9781139192194.

[6] Kenneth Greisen. "End to the Cosmic-Ray Spectrum?" In: *Phys. Rev. Lett.* 16 (17 Apr. 1966), pp. 748–750. DOI: 10.1103/PhysRevLett.16.748. URL: https://link.aps.org/doi/10.1103/PhysRevLett.16.748.

[7] Carmelo Evoli. *The Cosmic-Ray Energy Spectrum*. Dec. 2020. DOI: 10.5281/zenodo.4396125. URL: https://doi.org/10.5281/zenodo.4396125.

[8] Mathieu De Naurois. "The H.E.S.S. experiment : current status and future prospects". In: *PoS* ICRC2019 (2019), p. 656. DOI: 10.22323/1.358.0656.

[9] M. Aguilar et al. "Precision Measurement of the Proton Flux in Primary Cosmic Rays from Rigidity 1 GV to 1.8 TV with the Alpha Magnetic Spectrometer on the International Space Station". In: *prl* 114.17, 171103 (May 2015), p. 171103. DOI: 10.1103/PhysRevLett.114.171103.

[10] O. Adriani et al. "Ten years of PAMELA in space". In: *Rivista del Nuovo Cimento* 40 (Oct. 2017), pp. 473–522. DOI: 10.1393/ncr/i2017-10140-x.

[11] Y. S. Yoon et al. "Proton and Helium Spectra from the CREAM-III Flight". In: *The Astrophysical Journal* 839.1 (Apr. 2017), p. 5. ISSN: 1538-4357. DOI: 10.3847/1538-4357/aa68e4. URL: http://dx.doi.org/10.3847/1538-4357/aa68e4.

[12] R. D. Monkhoev et al. "The Tunka-Grande experiment". In: *Journal of Instrumentation* 12.6 (June 2017), p. C06019. DOI: 10.1088/1748-0221/12/06/C06019.

[13] In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 798 (Oct. 2015), pp. 172–213. ISSN: 0168-9002. DOI: 10.1016/j.nima.2015.06.058. URL: http://dx.doi.org/10.1016/j.nima.2015.06.058.

[14] M. Aguilar et al. "Towards Understanding the Origin of Cosmic-Ray Electrons". In: *prl* 122.10, 101101 (Mar. 2019), p. 101101. DOI: 10.1103/PhysRevLett.122.101101.

[15] Johannes Albrecht et al. *The Muon Puzzle in cosmic-ray induced air showers and its connection to the Large Hadron Collider*. 2021. arXiv: 2105.06148 [astro-ph.HE].

[16] Anatoli Fedynitch, Julia Becker Tjus, and Paolo Desiati. "Influence of hadronic interaction models and the cosmic ray spectrum on the high energy atmospheric muon and neutrino flux". In: *Physical Review D* 86.11 (Dec. 2012). ISSN: 1550-2368. DOI: 10.1103/physrevd.86.114024. URL: http://dx.doi.org/10.1103/PhysRevD.86.114024.

[17] Tanguy Pierog. "Air Shower Simulation with a New Generation of post-LHC Hadronic Interaction Models in CORSIKA". In: *PoS* ICRC2017 (2017), p. 1100. DOI: 10.22323/1.301.1100.

[18] Dembinski, H.P. et al. "Report on Tests and Measurements of Hadronic Interaction Properties with Air Showers". In: *EPJ Web Conf.* 210 (2019), p. 02004. DOI: 10.1051/epjconf/201921002004. URL: https://doi.org/10.1051/epjconf/201921002004.

[19] Hans Dembinski et al. "Data-driven model of the cosmic-ray flux and mass composition from 10 GeV to 10^{11} GeV". In: *PoS* ICRC2017 (2017), p. 533. DOI: 10.22323/1.301.0533.

[20] J. Matthews. "A Heitler model of extensive air showers". In: *Astroparticle Physics* 22.5 (2005), pp. 387–397. ISSN: 0927-6505. DOI: https://doi.org/10.1016/j.astropartphys.2004.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0927650504001598.

[21] Maria Vasileiou. "Strangeness production with ALICE at the LHC". In: *Physica Scripta* 95 (Apr. 2020). DOI: 10.1088/1402-4896/ab85fc.

[22] J. A. Bellido et al. "Muon content of extensive air showers: Comparison of the energy spectra obtained by the Sydney University Giant Air-shower Recorder and by the Pierre Auger Observatory". In: *Phys. Rev. D* 98 (2 July 2018), p. 023014. DOI: 10.1103/PhysRevD.98.023014. URL: https://link.aps.org/doi/10.1103/PhysRevD.98.023014.

[23] A.G. Bogdanov et al. "Investigation of very high energy cosmic rays by means of inclined muon bundles". In: *Astroparticle Physics* 98 (2018), pp. 13–20. ISSN: 0927-6505. DOI: https://doi.org/10.1016/j.astropartphys.2018.01.003. URL: https://www.sciencedirect.com/science/article/pii/S0927650517302566.

[24] S. Müller et al. "Impact of muon detection thresholds on the separability of primary cosmic rays". In: *Astroparticle Physics* 97 (2018), pp. 174–185. ISSN: 0927-6505. DOI: https://doi.org/10.1016/j.astropartphys.2017.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0927650517301305.

[25] Dennis Soldin. "High pT muons from cosmic ray air showers in IceCube ". In: *PoS* ICRC2015 (2016), p. 256. DOI: 10.22323/1.236.0256.

[26] C. Corti et al. "Solar modulation of the local interstellar spectrum with VOYAGER 1, AMS-02,PAMELA, ANDBESS". In: *The Astrophysical Journal* 829.1 (Sept. 2016), p. 8. DOI: 10.3847/0004-637x/829/1/8. URL: https://doi.org/10.3847/0004-637x/829/1/8.

[27] A. M. Hillas. "The Origin of Ultra-High-Energy Cosmic Rays". In: 22 (1984), p. 425. DOI: 10.1146/annurev.aa.22.090184.002233.

[28] Abraham Achterberg et al. "Particle acceleration by ultrarelativistic shocks: theory and simulations". In: 328 (Dec. 2001), p. 393. DOI: 10.1046/j.1365-8711.2001.04851.x. eprint: astro-ph/0107530.

[29] M. Ackermann et al. “Detection of the Characteristic Pion-Decay Signature in Supernova Remnants”. In: *Science* 339.6121 (2013), pp. 807–811. DOI: 10.1126/science.1231160. eprint: https://www.science.org/doi/pdf/10.1126/science.1231160. URL: https://www.science.org/doi/abs/10.1126/science.1231160.

[30] A. Aab et al. “Observation of a large-scale anisotropy in the arrival directions of cosmic rays above 8 x 10 18 eV”. In: *Science* 357.6357 (2017), pp. 1266–1270. DOI: 10.1126/science.aan4338. eprint: https://www.science.org/doi/pdf/10.1126/science.aan4338. URL: https://www.science.org/doi/abs/10.1126/science.aan4338.

[31] A. Aab et al. “An Indication of Anisotropy in Arrival Directions of Ultra-high-energy Cosmic Rays through Comparison to the Flux Pattern of Extragalactic Gamma-Ray Sources”. In: *The Astrophysical Journal* 853.2 (Feb. 2018), p. L29. ISSN: 2041-8213. DOI: 10.3847/2041-8213/aaa66d. URL: http://dx.doi.org/10.3847/2041-8213/aaa66d.

[32] Karl-Heinz Kampert and Michael Unger. “Measurements of the cosmic ray composition with air shower experiments”. In: *Astroparticle Physics* 35.10 (2012), pp. 660–678. ISSN: 0927-6505. DOI: https://doi.org/10.1016/j.astropartphys.2012.02.004. URL: https://www.sciencedirect.com/science/article/pii/S0927650512000382.

[33] Robert M. Wagner. “Measurement of Very High Energy Gamma-Ray Emission from Four Blazars Using the MAGIC Telescope and a Comparative Blazar Study”. In: *Publications of the Astronomical Society of the Pacific* 119 (2006), pp. 1201–1203.

[34] P.A. Zyla et al. “Review of Particle Physics”. In: *PTEP* 2020.8 (2020), p. 083C01. DOI: 10.1093/ptep/ptaa104.

[35] S. Aiello et al. “Characterisation of the Hamamatsu photomultipliers for the KM3NeT Neutrino Telescope”. In: *Journal of Instrumentation* 13.05 (May 2018), P05035–P05035. DOI: 10.1088/1748-0221/13/05/p05035. URL: https://doi.org/10.1088/1748-0221/13/05/p05035.

[36] S. Adrián-Martínez et al. (KM3NeT Collaboration). “Letter of Intent for KM3NeT 2.0”. In: *Journal of Physics G: Nuclear and Particle Physics, 43 (8), 084001, 2016* (Jan. 27, 2016). DOI: 10.1088/0954-3899/43/8/084001. arXiv: 1601.07459v2 [astro-ph.IM].

[37] John David Jackson. *Klassische Elektrodynamik*. 3rd ed. De Gruyter, 2002.

[38] Mathieu de Naurois and Daniel Mazin. “Ground-based detectors in very-high-energy gamma-ray astronomy”. In: *Comptes Rendus Physique* 16.6 (2015). Gamma-ray astronomy / Astronomie des rayons gamma, pp. 610–627. ISSN: 1631-0705. DOI: https://doi.org/10.1016/j.crhy.2015.08.011. URL: https://www.sciencedirect.com/science/article/pii/S1631070515001462.

[39] Jannik Hofestädt. “Measuring the neutrino mass hierarchy with the future KM3NeT/ORCA detector”. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2017.

[40] Javier Tiffenberg. *UHE Neutrino searches with the Pierre Auger Observatory*. At the NUSKY 2011. URL: http://users.ictp.it/~smr2246/monday/tiffenberg-NUSKY.pdf (visited on 11/26/2021).

[41] IceCube Collaboration et al. “IceCube Data for Neutrino Point-Source Searches Years 2008-2018”. In: (Jan. 2021). DOI: 10.21234/CPKQ-K003. arXiv: 2101.09836 [astro-ph.HE].

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

[43] R. Abbasi et al. “IceTop: The surface component of IceCube”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 700 (Feb. 2013), pp. 188–220. ISSN: 0168-9002. DOI: 10.1016/j.nima.2012.10.067. URL: http://dx.doi.org/10.1016/j.nima.2012.10.067.

[44] M. Ageron et al. “ANTARES: The first undersea neutrino telescope”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 656.1 (Nov. 2011), pp. 11–38. ISSN: 0168-9002. DOI: `10.1016/j.nima.2011.06.103`. URL: `http://dx.doi.org/10.1016/j.nima.2011.06.103`.

[45] Fedor Šimkovic et al. “Neutrino telescope in Lake Baikal: Present and Future”. In: *PoS* ICRC2019 (2019), p. 1011. DOI: `10.22323/1.358.1011`.

[46] Rastislav Dvornický et al. “The Baikal-GVD neutrino telescope as an instrument for studying Baikal water luminescence”. In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)* (July 2021). DOI: `10.22323/1.395.1113`. URL: `http://dx.doi.org/10.22323/1.395.1113`.

[47] S. Aiello et al. “gSeaGen: The KM3NeT GENIE-based code for neutrino telescopes”. In: *Computer Physics Communications* 256 (Nov. 2020). ISSN: 0010-4655. DOI: `10.1016/j.cpc.2020.107477`. URL: `http://dx.doi.org/10.1016/j.cpc.2020.107477`.

[48] D. Heck et al. “CORSIKA: A Monte Carlo code to simulate extensive air showers”. In: *FZKA-6019* (Feb. 1998).

[49] Ralph Engel et al. “The hadronic interaction model Sibyll – past, present and future”. In: *EPJ Web of Conferences* 145 (Jan. 2017), p. 08001. DOI: `10.1051/epjconf/201614508001`.

[50] Klaus Werner. “The hadronic interaction model EPOS”. In: *Nuclear Physics B - Proceedings Supplements* 175-176 (2008). Proceedings of the XIV International Symposium on Very High Energy Cosmic Ray Interactions, pp. 81–87. ISSN: 0920-5632. DOI: `https://doi.org/10.1016/j.nuclphysbps.2007.10.012`. URL: `https://www.sciencedirect.com/science/article/pii/S0920563207007736`.

[51] J. Ranft. “Dual parton model at cosmic ray energies”. In: *Phys. Rev. D* 51 (1 Jan. 1995), pp. 64–84. DOI: `10.1103/PhysRevD.51.64`. URL: `https://link.aps.org/doi/10.1103/PhysRevD.51.64`.

[52] S.A. Bass et al. “Microscopic models for ultrarelativistic heavy ion collisions”. In: *Progress in Particle and Nuclear Physics* 41 (1998), pp. 255–369. ISSN: 0146-6410. DOI: `https://doi.org/10.1016/S0146-6410(98)00058-1`. URL: `https://www.sciencedirect.com/science/article/pii/S0146641098000581`.

[53] H. Fesefeldt. *Report PITHA-85/02*. Tech. rep. RWTH, Aachen, 1985.

[54] Joerg Rudolf Hoerandel, N. N. Kalmykov, and A. I. Pavlov. “The Knee in the Energy Spectrum of Cosmic Rays in the Framework of the Poly-Gonato and Diffusion Models”. In: *International Cosmic Ray Conference*. Vol. 1. International Cosmic Ray Conference. July 2003, p. 243.

[55] Thomas K. Gaisser, Todor Stanev, and Serap Tilav. “Cosmic Ray Energy Spectrum from Measurements of Air Showers”. In: *Frontiers of Physics* (Apr. 2013). DOI: `10.1007/s11467-013-0319-7`. arXiv: `1303.3565v1 [astro-ph.HE]`.

[56] Thomas K. Gaisser. “Spectrum of cosmic-ray nucleons, kaon production, and the atmospheric muon charge ratio”. In: *Astroparticle Physics* 35.12 (July 2012), pp. 801–806. ISSN: 0927-6505. DOI: `10.1016/j.astropartphys.2012.02.010`. URL: `http://dx.doi.org/10.1016/j.astropartphys.2012.02.010`.

[57] P. Antonioli et al. “A three-dimensional code for muon propagation through the rock: MUSIC”. In: *Astroparticle Physics* 7.4 (Oct. 1997), pp. 357–368. ISSN: 0927-6505. DOI: `10.1016/s0927-6505(97)00035-2`. URL: `http://dx.doi.org/10.1016/S0927-6505(97)00035-2`.

[58] J.-H. Koehne et al. “PROPOSAL: A tool for propagation of charged leptons”. In: *Computer Physics Communications* 184.9 (2013), pp. 2070–2090. ISSN: 0010-4655. DOI: `https://doi.org/10.1016/j.cpc.2013.04.001`. URL: `https://www.sciencedirect.com/science/article/pii/S0010465513001355`.

[59] G. Carminati, A. Margiotta, and M. Spurio. “Atmospheric Muons from Parametric formulas: a fast Generator for neutrino telescopes (MUPAGE)”. In: *Comput. Phys. Commun. 179:915-923, 2008* (Feb. 5, 2008). DOI: `10.1016/j.cpc.2008.07.014`. arXiv: `0802.0562v2 [physics.ins-det]`.

[60] C. Forti et al. “Simulation of atmospheric cascades and deep-underground muons”. In: *Phys. Rev. D* 42 (11 Dec. 1990), pp. 3668–3689. DOI: `10.1103/PhysRevD.42.3668`. URL: `https://link.aps.org/doi/10.1103/PhysRevD.42.3668`.

[61] Jörg R. Hörandel. “On the knee in the energy spectrum of cosmic rays”. In: *Astroparticle Physics* 19.2 (May 2003), pp. 193–220. ISSN: 0927-6505. DOI: `10.1016/s0927-6505(02)00198-6`. URL: `http://dx.doi.org/10.1016/S0927-6505(02)00198-6`.

[62] Y BECHERINI et al. “A parameterisation of single and multiple muons in the deep water or ice”. In: *Astroparticle Physics* 25.1 (Feb. 2006), pp. 1–13. ISSN: 0927-6505. DOI: `10.1016/j.astropartphys.2005.10.005`. URL: `http://dx.doi.org/10.1016/j.astropartphys.2005.10.005`.

[63] Ch. Berger et al. “Experimental study of muon bundles observed in the Fréjus detector”. In: *Phys. Rev. D* 40 (7 Oct. 1989), pp. 2163–2171. DOI: `10.1103/PhysRevD.40.2163`. URL: `https://link.aps.org/doi/10.1103/PhysRevD.40.2163`.

[64] M. Ambrosio and MACRO. “Measurement of the residual energy of muons in the Gran Sasso underground Laboratories”. In: *Astropart.Phys.19:313-328,2003* (July 2002). DOI: `10.1016/S0927-6505(02)00217-7`. arXiv: `hep-ex/0207043 [hep-ex]`.

[65] G. Carminati et al. “MUPAGE: a fast atmospheric MUon GEnerator for neutrino telescopes based on PArametric formulas”. In: (July 2009). arXiv: `0907.5563 [astro-ph.IM]`.

[66] Maarten de Jong. “Multi-dimensional interpolations in C++”. In: (July 2019). arXiv: `1907.02597 [cs.MS]`.

[67] Steffen Hallmann. “Sensitivity to atmospheric tau-neutrino appearance and all-flavour search for neutrinos from the Fermi Bubbles with the deep-sea telescopes KM3NeT/ORCA and ANTARES”. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2021.

[68] S. Aiello et al. (KM3NeT Collaboration). “Determining the Neutrino Mass Ordering and Oscillation Parameters with KM3NeT/ORCA”. In: (Mar. 2021). arXiv: `2103.09885 [hep-ex]`.

[69] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[70] David Silver et al. “Mastering the game of Go without human knowledge”. In: *Nature* 550 (Oct. 2017), p. 354. URL: `http://dx.doi.org/10.1038/nature24270`.

[71] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: (Sept. 4, 2014). arXiv: `1409.1556v6 [cs.CV]`.

[72] Stefan Reck. “Investigating systematics for KM3NeT/ORCA using unsupervised Deep Learning”. MA thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2018.

[73] Stanford University. *CS231n syllabus*. URL: `http://cs231n.github.io/neural-networks-1/`.

[74] Kurt Hornik. “Approximation capabilities of multilayer feedforward networks”. In: *Neural Networks* 4.2 (1991), pp. 251–257. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(91)90009-T. URL: http://www.sciencedirect.com/science/article/pii/089360809190009T.

[75] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0.

[76] Nicolas Brunel, Vincent Hakim, and Magnus JE Richardson. “Single neuron dynamics and computation”. In: *Current Opinion in Neurobiology* 25 (2014). Theoretical and computational neuroscience, pp. 149–155. ISSN: 0959-4388. DOI: https://doi.org/10.1016/j.conb.2014.01.005. URL: http://www.sciencedirect.com/science/article/pii/S0959438814000130.

[77] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS’12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105. URL: http://dl.acm.org/citation.cfm?id=2999134.2999257.

[78] Bing Xu et al. “Empirical Evaluation of Rectified Activations in Convolutional Network”. In: (May 2015). arXiv: 1505.00853 [cs.LG].

[79] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (Dec. 22, 2014). arXiv: 1412.6980v9 [cs.LG].

[80] Sebastiano Aiello et al. “Event reconstruction for KM3NeT/ORCA using convolutional neural networks”. In: *JINST 15 P10005 (2020)* (Apr. 2020). DOI: 10.1088/1748-0221/15/10/P10005. arXiv: 2004.08254 [astro-ph.IM].

[81] Christopher M. Bishop. “Mixture density networks”. Technical Report. Birmingham, 1994. URL: https://publications.aston.ac.uk/id/eprint/373/.

[82] Ivan Kobyzev, Simon J. D. Prince, and Marcus A. Brubaker. “Normalizing Flows: An Introduction and Review of Current Methods”. In: (Aug. 2019). DOI: 10.1109/TPAMI.2020.2992934. arXiv: 1908.09257 [stat.ML].

[83] Thorsten Glüsenkamp. “Unifying supervised learning and VAEs – automating statistical inference in high-energy physics”. In: (Aug. 2020). arXiv: 2008.05825 [cs.LG].

[84] Michael Moser. “Sensitivity studies on tau neutrino appearance withKM3NeT/ORCA using Deep Learning Techniques”. PhD thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.

[85] Stefan Reck et al. *OrcaNet*. Nov. 2021. DOI: 10.5281/zenodo.5996080.

[86] S. Reck et al. “Graph neural networks for reconstruction and classification in KM3NeT”. In: *Journal of Instrumentation* 16.10 (Oct. 2021), p. C10011. DOI: 10.1088/1748-0221/16/10/c10011. URL: https://doi.org/10.1088/1748-0221/16/10/c10011.

[87] Nicole Geißelbrecht. “Event Classification and Energy Reconstruction for ANTARES using Convolutional Neural Networks”. MA thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2021.

[88] R. Abbasi et al. “A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory”. In: *JINST 16 (2021) P07041* (Jan. 2021). DOI: 10.1088/1748-0221/16/07/P07041. arXiv: 2101.11589 [hep-ex].

[89] Thomas Vuillaume et al. “Analysis of the Cherenkov Telescope Array first Large-Sized Telescope real data using convolutional neural networks”. In: (Aug. 2021). arXiv: 2108.04130 [astro-ph.IM].

[90] Nicholas Choma et al. “Graph Neural Networks for IceCube Signal Classification”. In: (Sept. 2018). arXiv: 1809.06166 [cs.LG].

[91] Robin J. Wilson. *Introduction to Graph Theory.* Prentice Hall, 2010. ISBN: 978-0273728894.

[92] Alexander J. Smola and Risi Kondor. “Kernels and Regularization on Graphs”. In: *Learning Theory and Kernel Machines.* Ed. by Bernhard Schölkopf and Manfred K. Warmuth. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 144–158. ISBN: 978-3-540-45167-9.

[93] Jie Zhou et al. “Graph Neural Networks: A Review of Methods and Applications”. In: (2018). arXiv: 1812.08434. URL: http://arxiv.org/abs/1812.08434.

[94] David I Shuman et al. “The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains”. In: (Oct. 2012). DOI: 10.1109/MSP.2012.2235192. arXiv: 1211.0053 [cs.DM].

[95] Joan Bruna et al. “Spectral Networks and Locally Connected Networks on Graphs”. In: (Dec. 2013). arXiv: 1312.6203 [cs.LG].

[96] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems 29 (2016)* (June 2016). arXiv: 1606.09375 [cs.LG].

[97] Huilin Qu and Loukas Gouskos. “ParticleNet: Jet Tagging via Particle Clouds”. In: *Phys. Rev. D 101, 056019 (2020)* (Feb. 22, 2019). DOI: 10.1103/PhysRevD.101.056019. arXiv: 1902.08570v3 [hep-ph].

[98] Yue Wang et al. “Dynamic Graph CNN for Learning on Point Clouds”. In: (2018). arXiv: 1801.07829. URL: http://arxiv.org/abs/1801.07829.

[99] John Nickolls et al. “Scalable Parallel Programming with CUDA”. In: *Queue* 6 (Mar. 2008), pp. 40–53. DOI: 10.1145/1401132.1401152.

[100] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: (Dec. 2015). arXiv: 1512.03385.

[101] Daniel Guderian. “Data-driven calibration studies and data analysis based on graph neural networks for oscillation research in KM3NeT/ORCA”. PhD thesis. Westfälische Wilhelms-Universität Münster, 2022.

[102] Stefan Reck, Thomas Eberl, and Ulì Katz. “Muon bundle reconstruction with KM3NeT/ORCA using graph convolutional networks”. In: *PoS* ICRC2021 (2021), p. 1048. DOI: 10.22323/1.395.1048.

[103] Brían O Fearraigh. “Tuning parametric models of the atmospheric muon flux in MUPAGE to data from the KM3NeT detector”. In: *PoS* ICRC2021 (2021), p. 1176. DOI: 10.22323/1.395.1176.

[104] Hamamatsu Photonics. *Photomultiplier Tubes - Basics and Applications.* 3a. 2007.

[105] Elise Lotte de Waardt. “Signatures of cosmic ray interactions in the KM3NeT neutrino detector”. MA thesis. University of Amsterdam, NIKHEF, 2018.

[106] Piotr Kalaczyński et al. “Comparison of the measured atmospheric muon rate with Monte Carlo simulations and sensitivity study for detection of prompt atmospheric muons with KM3NeT”. In: *Proceedings of 37th International Cosmic Ray Conference — PoS(ICRC2021)* (July 2021). DOI: 10.22323/1.395.1112. URL: http://dx.doi.org/10.22323/1.395.1112.

[107] Jorge J Moré. *The Levenberg-Marquardt algorithm: implementation and theory.* Springer, 1978, pp. 105–116.

[108] T. K. Gaisser and M. Honda. "FLUX OF ATMOSPHERIC NEUTRINOS". In: *Annual Review of Nuclear and Particle Science* 52.1 (2002), pp. 153–199. DOI: 10.1146/annurev.nucl.52.050102.090645. eprint: https://doi.org/10.1146/annurev.nucl.52.050102.090645. URL: https://doi.org/10.1146/annurev.nucl.52.050102.090645.

[109] P. Adamson et al. "Observation of seasonal variation of atmospheric multiple-muon events in the MINOS Near and Far Detectors". In: *Physical Review D* 91.11 (June 2015). DOI: 10.1103/physrevd.91.112006. URL: https://doi.org/10.1103%2Fphysrevd.91.112006.

[110] N. Agafonova et al. "Measurement of the cosmic ray muon flux seasonal variation with the OPERA detector". In: *Journal of Cosmology and Astroparticle Physics* 2019.10 (Oct. 2019), pp. 003–003. DOI: 10.1088/1475-7516/2019/10/003. URL: https://doi.org/10.1088%2F1475-7516%2F2019%2F10%2F003.

[111] M. A. Acero et al. "Seasonal variation of multiple-muon cosmic ray air showers observed in the NOvA detector on the surface". In: *Physical Review D* 104.1 (July 2021). DOI: 10.1103/physrevd.104.012014. URL: https://doi.org/10.1103%2Fphysrevd.104.012014.

[112] Serap Tilav et al. *Seasonal variation of atmospheric muons in IceCube.* 2019. DOI: 10.48550/ARXIV.1909.01406. URL: https://arxiv.org/abs/1909.01406.

[113] Stefano Castro Tognini. "Observation of multiple-muon seasonal variations in the NOvA Near Detector". PhD thesis. 2018. DOI: 10.2172/1468447. URL: https://www.osti.gov/biblio/1468447.

[114] S. Aiello et al. "Implementation and first results of the KM3NeT real-time core-collapse supernova neutrino search". In: *The European Physical Journal C* 82.4 (Apr. 2022). DOI: 10.1140/epjc/s10052-022-10137-y. URL: https://doi.org/10.1140%2Fepjc%2Fs10052-022-10137-y.

[115] Jelmer Mulder. "Seasonal variation of the muon rate at KM3NeT- The influence of the atmospheric temperature". Bachelor's Thesis. National Institute for Subatomic Physics - Nikhef, 2021.

[116] M. G. Aartsen et al. "Cosmic ray spectrum and composition from PeV to EeV using 3 years of data from IceTop and IceCube". In: *Phys. Rev. D* 100.8 (2019), p. 082002. DOI: 10.1103/PhysRevD.100.082002. arXiv: 1906.04317 [astro-ph.HE].

[117] Fahim Varsi et al. "Cosmic ray energy spectrum and composition measurements from the GRAPES-3 experiment: Latest results". In: *PoS* ICRC2021 (2021), p. 388. DOI: 10.22323/1.395.0388.

[118] Julia Häfner. "Reconstruction of the energy of muons and muon bundles via Graph Neural Networks in the ORCA detector". Bachelor's Thesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2021.

[119] T. Pierog et al. "EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider". In: *Phys. Rev. C* 92 (3 Sept. 2015), p. 034906. DOI: 10.1103/PhysRevC.92.034906. URL: https://link.aps.org/doi/10.1103/PhysRevC.92.034906.

[120] S. Ostapchenko. "Monte Carlo treatment of hadronic interactions in enhanced Pomeron scheme: QGSJET-II model". In: *Phys. Rev. D* 83 (1 Jan. 2011), p. 014018. DOI: 10.1103/PhysRevD.83.014018. URL: https://link.aps.org/doi/10.1103/PhysRevD.83.014018.

List of Figures

2.1. The cosmic ray energy flux multiplied by E^2 . . . 9
2.2. Selected cosmic ray composition measurements up to the ankle. . . . 11
2.3. The z scale measuring the logarithmic muon multiplicity plotted over the calibrated primary energy. . . . 12
2.4. The classical Hillas plot with potential candidates for the acceleration of cosmic rays. . . . 15
2.5. Mean logarithmic mass $\langle \ln A \rangle$ of cosmic rays plotted over their energy [15]. . . . 17
2.6. Scheme of an extensive air shower induced by a cosmic ray particle. . . . 19
2.7. Feynman diagrams of the decay of a positively and negatively charged pion. . . . 20
2.8. The energy loss of muons in matter for different energy regions. . . . 21

3.1. Map of relevant locations of the KM3NeT experiment. . . . 24
3.2. Photos of a a prototype of a Hamamatsu photomultiplier, and a KM3NeT digital optical module (DOM). . . . 24
3.3. Artists impression of a fully constructed KM3NeT building block with 115 installed detection strings, each of which contains 18 DOMs (courtesy KM3NeT). . . . 25
3.4. Scheme of the propagation of Cherenkov radiation. . . . 26
3.5. Scheme of the different interactions between a neutrino and a nucleon [40]. . . . 27

4.1. Scheme of the processing chain for atmospheric muons and data in KM3NeT. . . . 30
4.2. Scheme of the can, in which the light of particles is simulated. . . . 30
4.3. Distribution of various observables of atmospheric muon events after the trigger stage of the simulation chain. . . . 34

5.1. Scheme of a shallow, fully connected feed-forward network. . . . 37
5.2. Example of three different, commonly used activation functions: The sigmoid function, the rectified linear unit (ReLu) and the Softmax function. . . . 40

6.1. Example of a simple directional graph featuring 3 nodes and 4 edges. . . . 49
6.2. Example of the transformation of a KM3NeT event into a graph, in this case for an atmospheric muon event simulated with MUPAGE for ORCA4. . . . 51
6.3. Scheme showing how graphs can be stored efficiently in memory. . . . 52
6.4. The structure of ParticleNet and the EdgeConv block as published by Qu and Gouskos [97]. . . . 53

7.1. Training history of the neural network used for the directional reconstruction. . . . 60
7.2. 2D histograms of the directional reconstruction of muon bundles with the neural network. . . . 60
7.3. Error in the zenith reconstruction for deep learning and the classical reconstruction over the true cosine zenith angle. . . . 61
7.4. Angle between the reconstructed direction and the true direction as a function of the true muon multiplicity for deep learning and the classical reconstruction. . . . 62

7.5. Histograms of the reconstructed cosine zenith angle over the true cosine zenith angle for different cuts on the reconstruction quality. 63
7.6. Pull distributions of the directional reconstruction. 64
7.7. Comparison of the reconstructed cosine zenith angle on simulations (solid lines) and data (dots) for deep learning and the classical reconstruction. 65
7.8. Data-MC comparison of the reconstructed cosine zenith angle, ... 67
7.9. Data-Monte Carlo comparison of the reconstructed sigma zenith of the neural network. 67
7.10. Z-T plots of selected events that have a high uncertainty in the deep learning directional reconstruction. 68
7.11. Scheme showing the positions of the atmospheric muons of a bundle in the shower plane. The bundle diameter $\mathrm{R_b}$ is defined in this work as the maximum perpendicular distance between any two visible muons in the bundle. 70
7.12. Training history of the neural network used for the bundle diameter reconstruction. 71
7.13. 2D histograms of the bundle diameter reconstruction with the neural network. . 72
7.14. Pull distribution of the bundle diameter reconstruction. 72
7.15. 2D histogram of the bundle diameter reconstruction with the neural network. . . 73
7.16. The median relative error of the diameter reconstruction plotted over the true diameter. 74
7.17. 2D histograms of the bundle diameter reconstruction for the 44% of events with the highest reco quality. 75
7.18. Distribution of the true bundle energy, defined as the sum of the true energies of all visible muons in the bundle. 76
7.19. 2D histogram of the bundle diameter reconstruction with the neural network. . . 76
7.20. Distribution of the true and reconstructed bundle diameter for all events with two or more muons. 76
7.21. Distribution of the true muon multiplicity for all events (blue) and only the best 44% of events according to the reconstruction quality (orange). 76
7.22. True muon positions in the shower plane of 12 random events. 77
7.23. Comparison of the reconstructed bundle diameter on simulations and data. . . . 79
7.24. Architecture of the network used for the multiplicity reconstruction. It consists of two separate output towers trained with different loss functions (mean squared error and normal likelihood). 80
7.25. Training history of the neural network used for the multiplicity reconstruction. . 81
7.26. 2D histograms of the muon multiplicity reconstruction versus the truth. 82
7.27. Distribution of the true muon multiplicity for all events and using different cuts on the reconstruction quality. 83
7.28. Mean relative error over the reconstructed muon multiplicity, using different cuts on the reconstruction quality. 83
7.29. 2D histograms of the muon multiplicity reconstruction. 84
7.30. Distribution of the reconstructed muon multiplicity and uncertainty on measured data. 85
7.31. Data-Monte Carlo comparison of the reconstructed muon multiplicity, using only events with $\sigma_{\mathrm{zenith}} < 0.16$ rad. 86
7.32. Data-Monte Carlo comparison of various quantities using the default Mupage weighting, and a weighting that is derived from correcting the measured multiplicity distribution. 87

8.1. 2D histograms of the directional reconstruction of muon bundles with the neural network. 90
8.2. Error in the zenith reconstruction for events simulated with Mupage and Corsika over the true cosine zenith angle. 91

8.3. Angle between the reconstructed direction and the true direction as a function of the true muon multiplicity for Mupage and Corsika 91
8.4. Data-Monte Carlo comparison of the reconstructed cosine zenith angle. 92
8.5. 2D histograms of the bundle diameter reconstruction for Mupage and Corsika. 93
8.6. The weighted median of the relative error of the diameter reconstruction plotted over the true diameter. 94
8.7. Comparison of the reconstructed bundle diameter on Mupage, Corsika and data. 94
8.8. 2D histograms of the muon multiplicity reconstruction for Mupage and Corsika. 95
8.9. Average relative error over the reconstructed muon multiplicity. 96
8.10. Mean bundle energy over the number of simulated muons. 96
8.11. Reconstructed versus true muon multiplicity for Mupage and Corsika. 97
8.12. Data-Monte Carlo comparison of the reconstructed muon multiplicity for Mupage and Corsika 97
8.13. Rate of Corsika events over the reconstructed muon multiplicity. 98
8.14. Flux of cosmic ray particles according to the GST 3-gen [55] model plotted over the true primary energy. 98
8.15. Rate of Corsika events over the reconstructed bundle zenith angle. Each curve shows the events produced by air showers that have been induced by the given primary particle. 99
8.16. Rate of Corsika proton and iron events over the reconstructed bundle diameter. Only events with more than one muon are used for this plot. 99
8.17. Data-MC comparison of the reconstructed muon multiplicity of 6 months of Orca4 line data versus the Corsika simulated dataset and its different primaries. . . . 100
8.18. Data-MC comparison of the reconstructed muon multiplicity using the GST 3-gen weights, before any fitting is done. 101
8.19. Cosmic ray particle flux over the true primary energy. The GST 3-gen model is shown as the solid lines, while the cH3a model is plotted with dashed lines. . . . 101
8.20. Result of the least squares fit of the reconstructed muon multiplicity distributions from Corsika to data. 102
8.21. Cosmic ray particle flux over the true primary energy using the GST 3-gen model, arriving at Earth and at trigger level. 103
8.22. Result of the least squares fit similar to Figure 8.20b, except that the cH3a flux model was used for the energy spectrum of the cosmic rays. 105
8.23. Measured composition of the cosmic ray flux using different models for the energy spectrum. The error bars are obtained from the standard deviation of the least squares fit. 105
8.24. Data-MC comparison of the reconstructed bundle diameter for events with $\sigma_{\text{diameter}} < 0.15$. 106
8.25. Data-MC comparison of the reconstructed bundle diameter using the Corsika simulations. 107
8.26. Overall rate of triggered events on measured data in the given time period, for all events and selected high quality events and runs. 109
8.27. Rate at which selected high quality events and runs on data and Mupage run-by-run are triggered. 109
8.28. Ratio between the trigger rate of high quality events and runs on measured data and the Mupage run-by-run simulation for each run in the given time period. . . 110
8.29. Change in the trigger rate of atmospheric muon events compared to the average over the entire six month time period on Mupage and data. 110
8.30. Change in the rate of atmospheric muons compared to the average over the entire six month period on measured data, depending on the reconstructed muon multiplicity. 111

8.31. Measured composition of the cosmic ray flux over the time period in which the corresponding events were recorded. 112

Appendix A. Configuration files

This chapter lists the configuration files used in OrcaSong and OrcaNet in order to produce the presented results. They are also included in the respective git repositories.

A.1. OrcaSong

The configuration files can also be found under the given name in the OrcaSong repository in the `configs` directory.

Mupage ORCA4

Filename in OrcaSong: `bundle_ORCA4_mupage_v5-40.toml`

```
# produce graphs as samples
mode = "graph"
# produce labels for atmospheric muon bundles
extractor = "bundle_mc"
# only hits within this time window around the first triggered hit
# are used for the graphs
time_window = [−250, 1000]

[extractor_config]
# DU1 was part of the simulation, but only DU 2, 3, 4 and 5 are active
inactive_du = 1
# the center of the detector for calculating the shower plane
plane_point = [17, 17, 111]
# save the best downgoing solution of the classical reconstruction
only_downgoing_tracks = true
```

Corsika ORCA4

Filename in OrcaSong: `bundle_ORCA4_corsika_sibyll_2-3c.toml`

```
mode = "graph"
extractor = "bundle_mc"
# coordinate system in corsika is different from the one in mupage
center_hits_to = [0.020, −0.040, 111.186]
time_window = [−250, 1000]

[extractor_config]
inactive_du = 1
plane_point = [17, 17, 111]
only_downgoing_tracks = true
# store information from the primary
is_corsika = true
```

Measured data ORCA4

Filename in OrcaSong: `bundle_ORCA4_data_v5-40.toml`

```
mode = "graph"
extractor = "bundle_data"
time_window = [-250, 1000]
correct_mc_time = false

[extractor_config]
only_downgoing_tracks = true
```

A.2. OrcaNet

The configuration files can also be found under the given names in the OrcaNet repository in the `orcanet_contrib/configs` directory.

Direction

Config

Filename in OrcaNet: `direction_config.toml`

```
[config]
batchsize=64
learning_rate = "lr.csv"
train_logger_flush = 10
verbose_train = 2
validate_interval = 1
shuffle_train = true
cleanup_models = true

sample_modifier="GraphEdgeConv"
label_modifier = {name="RegressionLabels", columns=['dir_x', 'dir_y', 'dir_z
    '], model_output='dir', stacks=2}
dataset_modifier = "as_recarray_dist"
```

The following file `lr.csv` was used to schedule the learning rate:

```
# epoch  fileno   set learning rate to
1        1        1e-3
3        1        1e-4
4        1        1e-5
```

Model

Filename in OrcaNet: `direction_model.toml`

```
[model]
type = "DisjointEdgeConvBlock"
next_neighbors = 16
shortcut = true

blocks = [
    {units=[64, 64, 64], batchnorm_for_nodes=true},
    {units=[128, 128, 128]},
    {units=[256, 256, 256], pooling=true},
    {type="OutputRegNormal", output_neurons=3, output_name="dir", unit_list
        =256, sigma_activation="exponential"},
]

[compile]
optimizer = 'keras:Adam'

[compile.losses]
dir = {function = 'lkl_normal', metrics=["mae_err_reco", "mse_err_reco"]}
```

Diameter

Config

Filename in OrcaNet: `diameter_config.toml`

```
[config]
batchsize=64
learning_rate = [0.001, 0.33]
train_logger_flush = 10
verbose_train = 2
validate_interval = 1
shuffle_train = true
cleanup_models = true

sample_modifier="GraphEdgeConv"
label_modifier = {name="RegressionLabels", columns=['max_pair_dist'],
    model_output='max_pair_dist', stacks=2, log10=true}
dataset_modifier = "as_recarray_dist"
```

Model

Filename in OrcaNet: `diameter_model.toml`

```
[model]
type = "DisjointEdgeConvBlock"
next_neighbors = 16
shortcut = true

blocks = [
    {units=[64, 64, 64], batchnorm_for_nodes=true},
    {units=[128, 128, 128]},
    {units=[256, 256, 256], pooling=true},
    {type="OutputRegNormal", output_neurons=1, output_name="max_pair_dist",
        unit_list=[256,]},
]

[compile]
optimizer = 'keras:Adam'

[compile.losses]
max_pair_dist = {function = 'lkl_normal'}
```

Multiplicity

Config

Filename in OrcaNet: `muon_mutli_config.toml`

```
[config]
batchsize=64
learning_rate = [0.001, 0.1]
train_logger_flush = 10
verbose_train = 2
validate_interval = 1
shuffle_train = true
cleanup_models = true

sample_modifier="GraphEdgeConv"
label_modifier = {name="RegressionLabelsSplit", columns="n_muons_10_mchits",
     model_output="n_muon_reg", log10=true}
dataset_modifier = 'as_recarray_dist_split'
```

Model

Filename in OrcaNet: `muon_multi_model.toml`

```
[model]
type = "DisjointEdgeConvBlock"
next_neighbors = 16
shortcut = true

blocks = [
    {units=[64, 64, 64], batchnorm_for_nodes=true},
    {units=[128, 128, 128]},
    {units=[256, 256, 256], pooling=true},
    {type="OutputRegNormalSplit", unit_list=256, output_name="n_muon_reg",
        output_neurons=1, sigma_unit_list=[256, 256], sigma_activation="
        softplus"},
]

[compile]
optimizer = 'keras:Adam'

[compile.losses]
n_muon_reg = {function = 'mean_squared_error'}
n_muon_reg_err = {function = 'lkl_normal'}
```

Acknowledgments

I'm incredibly grateful of the fantastic atmosphere and supportive working environment I have encountered during my years of research in KM3NeT. First of all, I therefore want to thank everyone at ECAP and in the KM3NeT Collaboration for the amazing time, the interesting discussions and the joint scientific effort! Furthermore, I want to express my special thanks to the following people (in alphabetical order):

Ronald Bruijn For organizing the Cosmic Ray group and helping me with all questions in this regard.

Thomas Eberl For the long lasting support ever since I started my Master's thesis, and introducing me to the topic of neutrino physics.

Tamás Gál For endless knowledge and support with everything regarding IT.

Nicole Geißelbrecht For the incredibly productive atmosphere devoid of any distractions in our office. And for great quotes which unfortunately didn't make it into my thesis.

Kay Graf For keeping ECAP running, and being always eager to help me with any issues that came up.

Steffen Hallmann For the great pre-corona time in our office, and somehow literally knowing everything.

Jannik Hofestädt For somehow also literally knowing everything (are you and Steffen secretly twins?).

Piotr Kalaczynski For the tireless effort to produce the CORSIKA simulations that were crucial for my thesis.

Uli Katz For being the best and most supportive professor anyone could think of!

Michael Moser For the great pre-corona time in our office, and for great discussions about Machine Learning as well as starting the deep learning software in KM3NeT.

Rodrigo Gracia Ruiz and Daniel Guderian For the extensive reviews of my thesis.