JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

Likelihood-free inference for IceCube low energy reconstruction

ICECUBE
SOUTH POLE NEUTRINO OBSERVATORY

Dissertation zur Erlangung des Grades
„Doktor der Naturwissenschaften“

vorgelegt dem Fachbereich Physik, Mathematik und Informatik
der Johannes Gutenberg-Universität Mainz

von

Jan Christopher Weldert

geboren am 20 November 1993 in Lahnstein

Mainz, 25. Juli 2022

Tag der müdlichen Prüfung: 26. Oktober 2022

Jan Christopher Weldert
ETAP
Institut für Physik
Staudingerweg 7
Johannes Gutenberg-Universität D-55099 Mainz
weldert@uni-mainz.de

Abstract

Many neutrino detectors, like for example the IceCube Neutrino Observatory, consist of several tons of light transparent media instrumented with hundreds to thousands of individual photosensors. They detect light emitted following a neutrino interaction with the respective medium. An important step and big challenge in these detectors is to infer (or reconstruct) the parameters of a model, describing the neutrino interaction, based on the detected photons. The entire process from photon emission via photon propagation to photon detection is too complicated to be fully known analytically. Therefore, statistical inference methods such as the maximum likelihood method rely on approximations of the photon arrival distributions at the sensors to reconstruct neutrino event parameters.
In contrast to the reconstruction, the simulation of neutrino events includes more details about the involved physics. Machine learning methods trained on simulation can gain access to this detailed information, but do not provide the same level of versatility and interpretability as, for example, a likelihood function. Most machine learning techniques provide only a single information per event, while a likelihood function can be used for e.g. parameter inference, the determination of confidence regions, and the calculation of likelihood ratios.

This work introduces a hybrid machine learning-likelihood method, called FreeDOM, where the likelihood is directly learned from simulation. It can be used to reconstruct events in a detector consisting of an array of individual photosensors. The high flexibility of this method makes it easily applicable to IceCube's low energy extension DeepCore, the IceCube Upgrade or in fact any desired geometry. It can handle different sensor types which is indispensable for the application to the IceCube Upgrade and likely any other future IceCube extension. In addition, the method is about 100 times faster than current likelihood-based reconstructions in DeepCore without sacrificing the benefits of using a likelihood, like determining confidence regions or calculating likelihood ratios. The reconstruction speed is an important point because DeepCore MC event samples contain many millions of events that need to be reconstructed and current likelihood-based reconstructions need $\mathcal{O}(10s)$ per event.

Contents

1. Introduction **1**

2. Physics background **3**
- 2.1. Neutrinos . . . 3
  - 2.1.1. Flavor eigenstates and interactions . . . 4
  - 2.1.2. Mass eigenstates and their ordering . . . 7
  - 2.1.3. Atmospheric neutrinos . . . 10
  - 2.1.4. Neutrino detection via Cherenkov radiation . . . 11
- 2.2. Neutrino oscillations . . . 13
  - 2.2.1. Vacuum oscillations . . . 15
  - 2.2.2. Matter effects (MSW effect) . . . 19

3. Parameter estimation **24**
- 3.1. Probability . . . 24
- 3.2. Likelihood function . . . 26
- 3.3. Machine learning . . . 27
  - 3.3.1. Categories . . . 28
  - 3.3.2. Methods . . . 30
- 3.4. Likelihood-free inference . . . 32

4. The IceCube Neutrino Observatory **36**
- 4.1. The detection medium - deep Antarctic ice . . . 37
- 4.2. The Digital Optical Module . . . 39
  - 4.2.1. Definition of a DOM hit . . . 40
- 4.3. Low energy instrumentation . . . 42
  - 4.3.1. DeepCore . . . 42
  - 4.3.2. The IceCube Upgrade . . . 44
- 4.4. Low energy reconstruction . . . 46
  - 4.4.1. Measurement and model parameters in the IceCube low energy context . . . 46
  - 4.4.2. Particle IDentification (PID) . . . 48
  - 4.4.3. Current reconstructions . . . 50
  - 4.4.4. Potential . . . 51

5. Implementation of likelihood-free inference - FreeDOM **53**
- 5.1. Decomposing the likelihood . . . 53
- 5.2. Learning the likelihood . . . 56
  - 5.2.1. Training set . . . 58

    - 5.2.2. Neural networks and training process . . . 61
    - 5.2.3. Changes for the IceCube Upgrade . . . 64
  - 5.3. Verifying the likelihood . . . 65
    - 5.3.1. Toy detectors . . . 65
    - 5.3.2. IceCube simulation . . . 71
  - 5.4. Minimizing the likelihood . . . 75

6. Applications of FreeDOM 82
  - 6.1. Toy experiment studies . . . 82
    - 6.1.1. One-string toy experiment . . . 82
    - 6.1.2. Spherical toy experiment . . . 86
    - 6.1.3. Detector optimization . . . 88
  - 6.2. DeepCore studies . . . 89
    - 6.2.1. Reconstruction performance . . . 90
    - 6.2.2. Physics sensitivities . . . 103
    - 6.2.3. MCMC posterior sampling . . . 107
    - 6.2.4. Estimating the influence of training sample size on sensitivities 109
  - 6.3. IceCube Upgrade studies . . . 113
    - 6.3.1. Reconstruction performance . . . 114
    - 6.3.2. mDOM noise study . . . 121

7. Uncertainty estimations 123
  - 7.1. Estimate per event parameter uncertainties . . . 123
  - 7.2. Weighting events . . . 126
  - 7.3. Impact of weights on sensitivities . . . 129

8. Summary and outlook 133

A. Toy experiment setup 137

B. Likelihood formulation rearrangement 138

C. Supplementary material 139

List of Figures

2.1. Standard model of particle physics 4
2.2. Neutrino cross-sections at GeV energies 6
2.3. DIS Feynman 6
2.4. Mixing angles 8
2.5. Neutrino Mass Ordering 9
2.6. Atmospheric ν_μ/ν_e ratio 11
2.7. Atmospheric neutrinos 12
2.8. Cherenkov radiation 13
2.9. Super-Kamiokande oscillation evidence 15
2.10. Neutrino interactions in matter 20
2.11. Atmospheric ν oscillation in Earth 23

3.1. Artificial neuron 31
3.2. Multilayer perceptron 31
3.3. Likelihood-free training algorithm 33
3.4. Data preparation for likelihood-free training 35

4.1. IceCube detector sketch 37
4.2. Optical properties of the deep Antarctic ice 38
4.3. Sketch of a PhotoMultiplier Tube (PMT) 39
4.4. Sketch of the Digital optical Module (DOM) 40
4.5. Waveform unfolding 41
4.6. Low energy instrumentation in IceCube 45
4.7. New optical modules for the Upgrade 46
4.8. Low energy example event displays 47
4.9. PID in IceCube 49
4.10. Sensitivity potential for improved resolutions and PID 52

5.1. Decomposition of the likelihood 57
5.2. oscNext selection levels 58
5.3. True oscNext parameter distributions 61
5.4. Network structure 64
5.5. One-string toy LLH scans 67
5.6. One-string toy MCMC sampling 68
5.7. Spherical toy reweighted hit time distributions 69
5.8. Spherical toy reweighted charge distributions 70
5.9. Comparison to other likelihoods 72
5.10. Reweighting for a “real” MC event 73

5.11. Hit time distributions at different DOMs 74
5.12. mDOM PMT segmentation likelihood scans 76
5.13. More minimizer livepoints 79
5.14. Additional local minimization 80
5.15. Resolutions for training vs new event 81

6.1. One-string toy parameter resolutions 83
6.2. Likelihood approximation quality depending on number of training events 84
6.3. Likelihood approximation quality for different training sets 85
6.4. Spherical toy parameter resolutions 87
6.5. Detector optimization study 89
6.6. Best-fit distributions 90
6.7. Reonstruction time for DeepCore events 92
6.8. Reconstructed vs true parameter values for DeepCore 93
6.9. Energy and zenith resolution for DeepCore 94
6.10. Resolution versus true energy for DeepCore 95
6.11. Resolution per flavor and interaction 97
6.12. Particle identification in DeepCore 98
6.13. Energy resolution for systematic sets 100
6.14. Zenith resolution for systematic sets 101
6.15. Data-MC agreement 102
6.16. Atmospheric $\nu_\mu \to \nu_\tau$ oscillation probabilities 104
6.17. Sensitivity to atmospheric oscillation parameters 105
6.18. Sensitivity to neutrino mass ordering 107
6.19. MCMC posterior sampling for DeepCore MC event 108
6.20. MCMC posterior sampling for DeepCore data event 109
6.21. Deepcore KS values 110
6.22. KS value vs sensitivity 112
6.23. Sensitivity vs training sample size 112
6.24. Best fit distributions Upgrade 114
6.25. Reconstructed versus true parameter values for the Upgrade 116
6.26. Upgrade parameter resolutions versus true energy 118
6.27. Parameter resolutions using only one module type 119
6.28. Resolutions for events not reconstructable with DeepCore 119
6.29. PID in the Upgrade 120
6.30. mDOM noise study 122

7.1. Correlation between estimated uncertainty and true residuum 125
7.2. Data-MC agreement for uncertainty estimations 126
7.3. Impact of weights on resolutions and PID 128
7.4. Sensitivities to atmospheric oscillation parameters using weighted samples 130
7.5. NMO sensitivity using weighted samples 131

8.1. Combination of different machine learning methods 135

C.1. Spherical toy reweighted charge per module distributions 139
C.2. True parameter distributions IceCube Upgrade 140
C.3. Likelihood scans DeepCore event 140
C.4. Uncertainty estimation methods 141
C.5. z vertex position of events with estimated uncertainty 0 141
C.6. Resolution ratios for L2 and L6 trained classifiers 142
C.7. oscNext event histogram example 142

List of Tables

6.1. The considered systematic sets and their difference to the nominal set. 99

C.1. Used systematics 143

1. Introduction

A central step for many particle detectors is to estimate or reconstruct the event-level parameters of interest $\boldsymbol{\theta}$ (e.g. energy, direction of an incoming particle) based on the actual measurement $\boldsymbol{x}$ (e.g. a time-dependent electrical current in a detector unit). A frequently used statistical method is the definition of a so-called likelihood function that calculates how likely it is to observe $\boldsymbol{x}$ for a given $\boldsymbol{\theta}$. However, a likelihood is not a probability density, since it is not normalized correctly. Ratios between likelihood values can be used to compare different hypotheses [1], where a higher likelihood value means that the hypothesis is more preferred by the data. Finding the maximum of the likelihood yields the so-called best-fit parameter values, which represent an estimator for the true values. The likelihood function can also be used to calculate confidence level regions (uncertainties) for the parameter values, based on the likelihood landscape around the maximum. In addition, likelihood ratios between hypotheses with different sets of parameters $\boldsymbol{\theta}$ show how important single parameters are to explain the measurement. Thus, a minimal model to describe the measurement $\boldsymbol{x}$ can be found. Furthermore, in the Bayesian interpretation of probability (see sec. 3.1), the likelihood can be used to sample posterior probability distributions for the parameters.

However, defining the correct likelihood function is challenging and actually intractable for many modern physics experiments. These experiments rely on approximations and simplifications to compute a likelihood function and reconstruct events. Often, simulating events includes more details than reconstructing them. For example, propagating photons from a certain start point through a transparent medium can be done including many details about the involved medium [2], although the likelihood function is unknown. Inverting the problem and asking about the origin of a detected photon, as needed for a reconstruction, is more difficult. The probabilities for all possible light sources to produce a photon reaching the specific sensor would have to be known which is not the case if the likelihood is unknown.

Machine learning offers techniques for universal function approximation. These techniques can make use of the detailed simulation and directly extract the relation between $\boldsymbol{x}$ and $\boldsymbol{\theta}$ from the simulated data. While they only require forward $\boldsymbol{\theta} \rightarrow \boldsymbol{x}$ simulation to train on, most machine learning based approaches work as point regressors, meaning that they only give a point estimation of the parameter values $\boldsymbol{\theta}$ without the additional benefits of a likelihood function described earlier.

The IceCube Neutrino Observatory at the South Pole, introduced in chapter 4, faces the problems described above. Its low energy extension DeepCore detects atmospheric neutrinos at a rate in the order of mHz (see fig. 5.2) but has an

intractable likelihood. Thus, assumptions must be made about the propagation of photons in ice in order to reconstruct events. The high number of detected events means that the speed of a reconstruction method is an important factor too.
Since IceCube will be upgraded in the next years, which involves completely new sensors in a new arrangement, a reconstruction must be flexible in both the type of sensors and their positioning.

Combining likelihood and machine learning methods keeps the likelihood benefits and can make use of detailed simulations. Likelihood-free inference [3] is an example for such a hybrid method. In this approach, the likelihood function is learned directly from the simulation, without the need to make explicit assumptions about the function in advance.
This work introduces a hybrid machine learning-likelihood reconstruction framework, called FreeDOM, which is geometry flexible and therefore suitable for IceCube DeepCore, the IceCube Upgrade, and in fact any detector consisting of an array of photosensors.
Chapter 2 gives an overview of the physics background that is important to understand what needs to be reconstructed, i.e. what $\boldsymbol{\theta}$ is in the IceCube low energy context. The concept of neutrino oscillation is described as well as the neutrino mass ordering, both are used to demonstrate use cases of the reconstruction later on.
Chapter 3 introduces different concepts of parameter estimation. The likelihood formalism is explained, followed by an overview of machine learning. Finally, the technique of likelihood-free inference, which is the basis of the reconstruction framework presented in this thesis, is introduced.
In chapter 4 the IceCube detector as well as its low energy extension and upgrade are described. The chapter familiarizes with the installed hardware of the detector and what exactly is measured with it, i.e. what $\boldsymbol{x}$ is in IceCube. Also the used set of model parameters $\boldsymbol{\theta}$ that should be reconstructed is introduced. The chapter closes with an overview of the currently used reconstructions.
The implementation of FreeDOM is the focus of chapter 5. It starts with a derivation of the expected form of the likelihood function which will be approximated. Details on the training process are given and the quality of the likelihood approximation is verified. It is also described how the minimization process of the negative log-likelihood function is performed.
Reconstruction results are shown in chapter 6. Parameter resolutions for IceCube DeepCore and the IceCube Upgrade as well as their impact on sensitivities to neutrino oscillation parameter and mass ordering analyses are presented. In addition, further applications of FreeDOM are highlighted to demonstrate its flexibility and versatility.
As mentioned before, to be able to determine per event parameter reconstruction uncertainties is an advantage of knowing the likelihood. Chapter 7 discusses ways to extract these uncertainties from the networks. It also shows the potential of using uncertainty weights in an analysis as well as some achieved improvements.
Finally, chapter 8 summarizes the results and gives an outlook on necessary next steps and possible future applications of the technique presented in this thesis

2. Physics background

To be able to construct a model whose parameters $\boldsymbol{\theta}$ can then be reconstructed, a solid understanding of the underlying physics is essential. Reconstructing neutrino induced events therefore requires detailed knowledge about these particles, how they are produced and how they interact.

For different physics analyses different parameters might be of importance. For example an analysis of a neutrino oscillation phenomenon depends on other aspects of the neutrino interaction than a neutrino mass measurement. While it might be of advantage to reconstruct as much information as possible from a measurement, the focus will usually be on just a few parameters. Identifying these parameters also requires knowledge about the physics background involved.

2.1. Neutrinos

The existence of neutrinos was first postulated by Wolfgang Pauli in 1930 [4] to explain the continuous energy spectrum of electrons emitted in a β-decay. If only two particles (electron and proton) would be the product of this decay just one specific energy value would be allowed for each of them to conserve energy, momentum, and spin. Only the existence of a third involved particle, unknown at that time, could explain a continuous energy spectrum. The name neutrino goes back to Enrico Fermi (Pauli suggested neutron as name) and means something like small neutral particle. Fermi developed a theory of the β-decay that includes a neutrino in 1933 [5]. This was the first attempt to theoretically describe the particle and its interaction. While the theoretical description followed rather quick, it took 26 years to confirm Pauli's prediction with a direct measurement. In 1956 Clyde L. Cowan and Frederick Reines detected electron anti-neutrinos via the inverse β-decay with their so called *poltergeist project* [6]. It was the first measurement of a neutrino, but just confirmed the existence of the electron neutrino. Measurements at the Alternating Gradient Synchrotron located at the Brookhaven National Lab performed by Leon Lederman, Melvin Schwartz, and Jack Steinberger proved the existence of a second neutrino type, the muon neutrino, in 1962 [7]. It was produced in pi meson decays. This also demonstrated the doublet structure of leptons. The third neutrino type, the tau neutrino, was finally detected by the DONUT collaboration in 2000 [8].

Since then neutrinos from different sources (e.g. solar [9], atmospheric [10] or reactor [6]) were observed. Today neutrinos are an important part of the current research.

Neutrinos belong to the leptons of the Standard Model (SM, fig 2.1) of particle physics and thus to the most fundamental components of the universe. As leptons they are fermions with a spin of $\frac{1}{2}$. One can distinguish between three different neutrino eigenstates (either flavor described in sec. 2.1.1 or mass introduced in sec. 2.1.2), one for each family of the SM. Measurements performed at the LHC at CERN on the Z^0 resonance confirmed the existence of $N_\nu = 2.9840 \pm 0.0082$ [11] neutrinos with less than half the Z^0 mass. This does not exclude the existence of additional sterile neutrinos which do not interacting weakly but could interact with the "active" ones via neutrino oscillation 2.2.

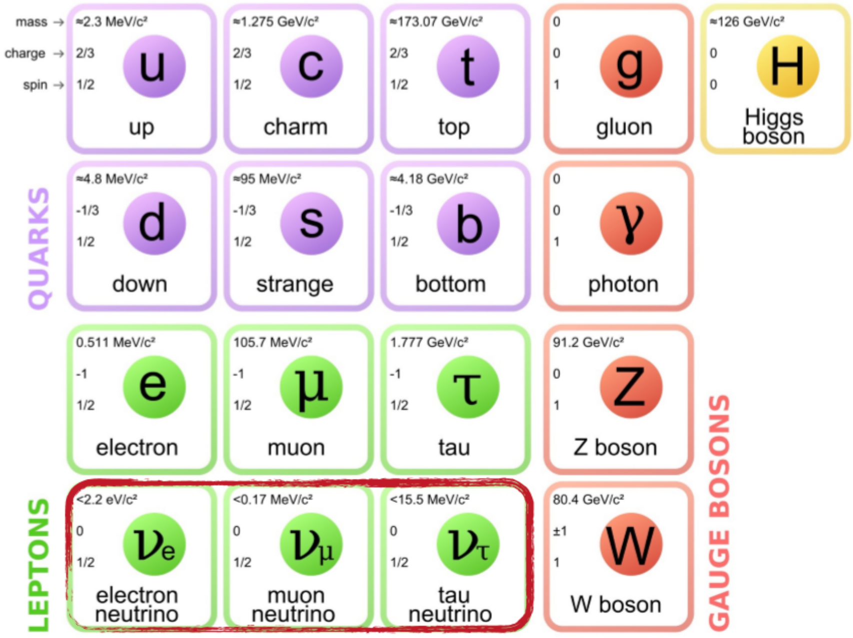


Figure 2.1.: The standard model (SM) of particle physics including the Higgs boson. The SM contains all fundamental building blocks (particles) of our universe that are known. Mass, electrical charge, and spin of all particles are given. The three neutrinos are highlighted with a red border.

2.1.1. Flavor eigenstates and interactions

Figure 2.1 shows the neutrino eigenstates in their flavor representation which is used to describe how neutrinos interact. They have no electric and no color charge, hence they only interact via the weak force and one could also interpret the flavor eigenstates as eigenstates of the weak interaction.
Two different interaction types are possible in the weak interaction:

1. **Charged current (CC)** interactions mediated by the exchange of an electrically charged $W^\pm$-boson. Here the neutrino is converted to the SM lepton with

the corresponding flavor or vice versa. Consequentially, for neutrinos in the initial state, there is an energy threshold for the interaction type which is the rest mass of the respective lepton.

2. **Neutral current (NC)** interactions mediated by the exchange of an electrically neutral Z^0-boson. All flavor eigenstates with any initial energy can interact.

Like for all SM fermions, for each neutrino flavor eigenstate there is a corresponding anti-neutrino state. Neutrinos and anti-neutrinos differ in their helicity, neutrinos are always left-handed (spin vector points in opposite direction to the momentum vector) while anti-neutrinos are right-handed. Right-handed neutrinos have not been observed so far but are a candidate for sterile neutrinos. If there are more differences than the helicity of the particles is subject of current research. Since it does not carry electrical charge, the neutrino could also be its own anti-particle, a so called Majorana particle.

That they can only interact weakly leads to the fact that neutrino interactions have very low cross-sections [12]. Depending on the neutrino energy different processes are important. At low energies, starting at a few MeV, the inverse β-decay has the highest cross-section, but only electron anti-neutrinos can undergo this process. For the other flavors elastic and quasi-elastic neutrino nucleon scattering are the dominant processes (for energies of just a few MeV elastic neutrino electron scattering is also important). The difference between these two processes is that elastic scattering refers to a neutral current interaction while quasi-elastic refers to a charged current interaction. At a few GeV resonance production becomes possible. Here the neutrino excites a nucleon to a baryonic resonance state. For higher energies of several GeV, the region interesting for IceCube (chapter 4), so-called deep inelastic neutrino nucleon scattering becomes the dominant process. The neutrino energy is high enough to break up the nucleon and create a shower of secondary particles.
Figure 2.2 shows the result of cross-section measurements for neutrino and anti-neutrino nucleon interactions at GeV energies. Individual cross-sections for quasi-elastic scattering (QE), resonance production (RES), and deep inelastic scattering (DIS) are displayed as well.
It can be seen that from about 5-10 GeV deep inelastic scattering is the dominant process. If the neutrino energy is high enough, its de Broglie wavelength [13] becomes short enough to resolve individual quarks in a nucleon and the neutrino can directly interact with them. In this process momentum is transferred from the neutrino to the quark which is kicked out of the nucleon as a result. This leads to a hadronic cascade because due to confinement [14] quarks can not exist freely. In case of a CC interaction the neutrino is converted into its leptonic partner which additionally leads to an electromagnetic cascade. Depending on the energy and type of lepton, this cascade is located closer or further away from the interaction point.
Figure 2.3 shows the Feynman diagrams [15] for a CC (left) and NC (middle) deep inelastic scattering. The right plot in the figure illustrates why the interaction cross-section is higher for neutrinos compared to anti-neutrinos. Matter consists mainly

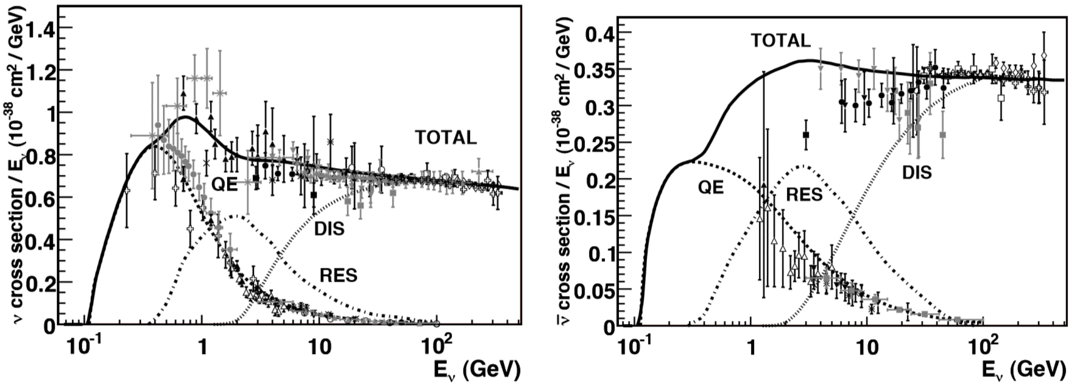


Figure 2.2.: Neutrino (left) and anti-neutrino (right) interaction cross-sections at GeV energies [12]. The individual cross-sections for quasi-elastic scattering (QE), resonance production (RES), and deep inelastic scattering (DIS) are given as well as their sum (TOTAL). One can see that the neutrino cross-section is about twice as high as the one for anti-neutrinos.

of quarks and not of anti-quarks. In a neutrino quark interaction there is no overall spin s and all scattering angles θ are possible. However, in a anti-neutrino quark interaction there is a overall spin of 1. The spin has to be conserved and consequently high scattering angles are suppressed. Therefore, the final state phase space is reduced and thus the cross-section for the process. This is called helicity suppression.

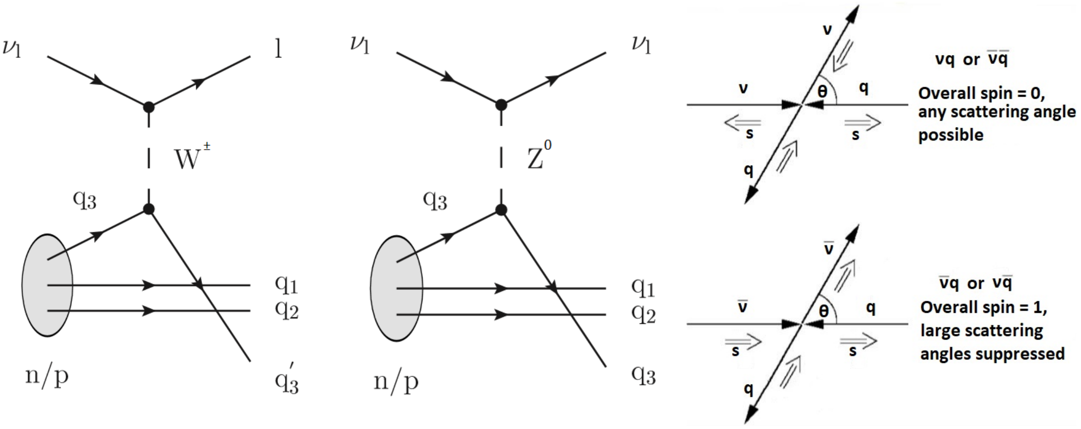

ν_l
l
W^±
q_3
q_1
q_2
q'_3
n/p
ν_l
ν_l
Z^0
q_3
q_1
q_2
q_3
n/p
v
θ
q
s
s
q
vq or v̄q̄
Overall spin = 0, any scattering angle possible
v̄
θ
q
s
s
q
v̄q or vq̄
Overall spin = 1, large scattering angles suppressed

Figure 2.3.: Feynman diagram of a CC (left) and NC (middle) DIS interaction. The right plot illustrates the helicity suppression for anti-neutrino interactions. See text for more details.

2.1.2. Mass eigenstates and their ordering

The following abbreviations are used in this and all subsequent sections of this thesis to keep equations short and improve clarity:

$$\begin{aligned} s_{ij} &\equiv \sin(\theta_{ij}) \\ c_{ij} &\equiv \cos(\theta_{ij}) \\ \Delta m^2_{ij} &\equiv m^2_i - m^2_j \\ \Delta_{ij} &\equiv \Delta m^2_{ij} \cdot \frac{L}{E} \,, \end{aligned}$$

where θ_{ij} is one of the mixing angles defined in this section, m_i the mass of the eigenstate ν_i also defined in this section, E the energy and L the traveled distance of the neutrino.

In the original SM formulation neutrinos were assumed to be masseless. The observation of neutrino oscillation 2.2 implied that this can not be true for all of them. As can be seen by looking at a simplified two flavor oscillation probability defined in equ. 2.16, oscillation is only possible if the different neutrino eigenstates have different masses $\Delta m^2 \neq 0$. Therefore, only a maximum of one can have a mass of zero and all other have to have non zero masses.
There is no well defined mass for each of the flavor eigenstates. They are mixtures of the different neutrino masses, which themselves are represented by mass eigenstates. Each mass eigenstate has a well defined mass, but no well defined way to interact (as the flavor eigenstates have). To change between the three known flavor (ν_e, ν_μ, ν_τ) and the respective mass eigenstates (ν_1, ν_2, ν_3) of a neutrino, a three dimensional matrix is needed. This matrix is called Pontecorvo–Maki–Nakagawa–Sakata (PMNS [16]) matrix and it is similar to the CKM matrix [17] in the quark sector.

$$\begin{pmatrix} \nu_e \\ \nu_\mu \\ \nu_\tau \end{pmatrix} = \begin{pmatrix} \mathcal{U}_{e1} & \mathcal{U}_{e2} & \mathcal{U}_{e3} \\ \mathcal{U}_{\mu 1} & \mathcal{U}_{\mu 2} & \mathcal{U}_{\mu 3} \\ \mathcal{U}_{\tau 1} & \mathcal{U}_{\tau 2} & \mathcal{U}_{\tau 3} \end{pmatrix} \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} . \tag{2.1}$$

The PMNS matrix $\mathcal{U}$ is the product from three rotation matrices and basically a base change of the space describing the eigenstate composition of a neutrino. The respective rotation angles, also called mixing angles, are named θ_{12}, θ_{13} and θ_{23}. Figure 2.4 illustrates the relation between the eigenstates shown as vectors which can be mapped onto each other by rotation through the mixing angles.
Writing out the elements of the PMNS matrix by explicitly multiplying the three individual rotations leads to:

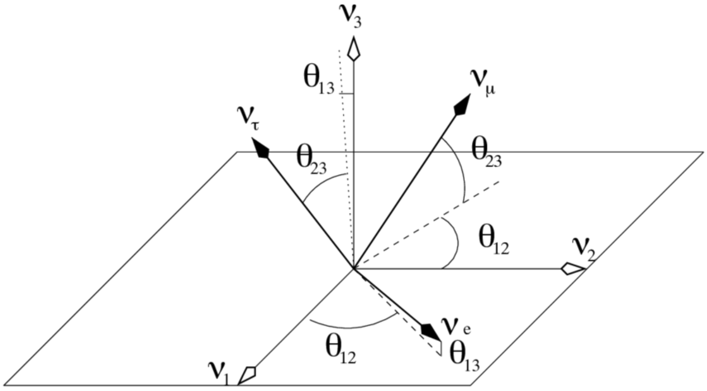


Figure 2.4.: Mixing between the three flavor and mass eigenstates. The eigenstates are represented as vectors and the mixing via Euler angles [18]. The plot does not include CP violation.

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

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

Here, δ_{CP} is a charge-parity (CP) violating phase introduced to be able to account for possible CP violation of the weak interaction. Measurements of the PMNS matrix elements [19] yield that they are comparatively close together, meaning that a strong mixing is present for neutrinos. This is different for quarks, where the CKM matrix is rather diagonal and the magnitudes of its elements differ by more than two orders.

While the mixing is strong, the values of the neutrino masses are orders of magnitudes smaller compared to all other particles of the standard model. The exact masses are still unknown but the current upper limit, set by the KATRIN collaboration, is $m_\nu < 0.8\,\mathrm{eV}$ [20]. This would be at least five orders below the second lightest particle, the electron with a rest mass of about 0.5 MeV.

Not only the values of the three different masses are unknown, their ordering (NMO for Neutrino Mass Ordering) is also undetermined. From measurements of solar neutrinos [21] it is known that $m_1 < m_2$ and $\Delta m^2_{21} \approx 7.5 \cdot 10^{-5}\,\mathrm{eV}^2$. However, the relative ordering of the third mass compared to the other two is still unknown. Only the absolute distance $|\frac{\Delta m^2_{31} + \Delta m^2_{32}}{2}| \approx 2.4 \cdot 10^{-3}\,\mathrm{eV}^2$ could be determined from atmospheric neutrino measurements [10]. This leaves two different possible orderings of the masses. The so-called "normal" ($m_1 < m_2 < m_3$, NO) and the so-called "inverted" ($m_3 < m_1 < m_2$, IO) ordering. The normal ordering is illustrated in the left part of fig. 2.5 and the inverted ordering in the right part of the same figure.

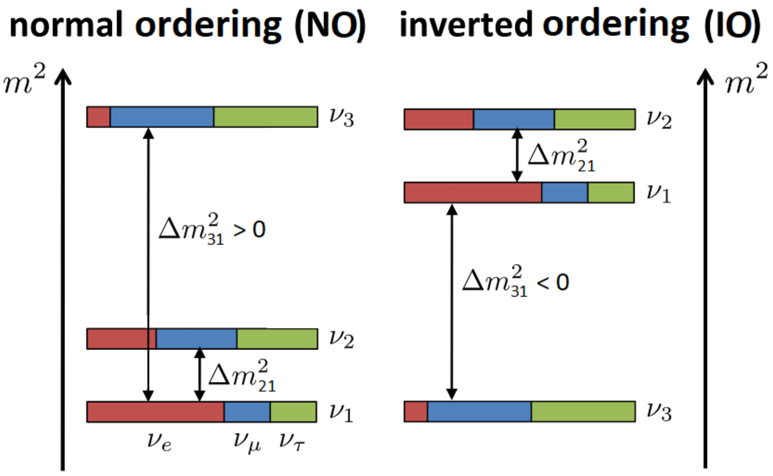


Figure 2.5.: The two different possible realizations of the neutrino mass ordering, so the ordering of the (squared) masses of the three neutrino mass eigenstates. In the normal ordering (left) the eigenstates are ordered like their names suggest $m_1 < m_2 < m_3$ (where m_i is the mass of eigenstate ν_i). In the inverted ordering (right) m_3 is the lightest mass, followed by m_1 and m_2. The colors represent the fractions of the flavor eigenstates to the respective mass eigenstate. Δm^2_{31} refers to the difference of the squared masses (not the square of the mass difference) and can therefore be negative. The sign of Δm^2_{31} (or Δm^2_{32}) is usually used to distinguish the orderings.

The main difference between the two possible orderings (apart from the sum of the masses of the three mass eigenstates) is the sign of Δm^2_{31} or Δm^2_{32}, which is positive for the normal and negative for the inverted ordering. As discussed in sec. 2.2 Δm^2_{ik} is a neutrino oscillation frequency and consequently neutrino oscillation properties have to be studied to determine the NMO. However, the sign of Δm^2_{ik} is not easily accessible. This can be seen by looking at formula 2.6 which shows that

$$P_{\alpha\rightarrow\beta} \propto \sin^2(\Delta m^2_{ik}) = \sin^2(|\Delta m^2_{ik}|)\,. \tag{2.2}$$

Nevertheless, there are multiple ways to determine the NMO. The approach used by IceCube (and similar detectors) is to explore matter effects (sec. 2.2.2) on oscillations as exploited for the "solar" ordering. More details about the IceCube NMO determination follow in sec. 6.2.2.

2.1.3. Atmospheric neutrinos

Since neutrinos only interact weakly, they can only be produced as a result of a weak interaction process. The β-decay, which lead Pauli to the postulation of the neutrino, is an example of such a process. In general, particles decaying via the weak interaction nearly always produce neutrinos to conserve the lepton family number. Also fusion processes involving the weak interaction, like the ones in the sun, produce neutrinos. These two types of processes lead to a wide range of neutrino sources: man made sources like particle accelerators or nuclear power plants, as well as natural sources like cosmic accelerators or radioactive decays in the Earth. Even most living beings are neutrino sources because of radioactive decays of ^{40}K in their bodies.
Another natural source of neutrinos, important for IceCube, is the Earths atmosphere. Uninterruptedly the Earth is bombarded by electrically charged particles with unknown origin. These particles are called (primary) cosmic radiation [22] and most of them are hadrons (mainly protons). In a height between 10-20 km [23] these particles interact with the different nucleons that make up the Earths atmosphere and as a result of these interactions particle showers are produced. The left part of fig. 2.7 shows a sketch of such a shower production. This is the so-called secondary cosmic radiation, which mainly consists of photons. Neutrinos, which are also part of this secondary radiation, are produced in the following interactions:

$$\begin{gathered} p + N \to \pi^\pm + X \\ \pi^- \to \mu^- + \overline{\nu}_\mu \\ \mu^- \to e^- + \overline{\nu}_e + \nu_\mu \end{gathered} \qquad \begin{gathered} p + N \to K^\pm + X \\ K^- \to \mu^- + \overline{\nu}_\mu \quad (63.56\%) \\ K^- \to \pi^- + \pi^0 \quad (20.67\%) \\ K^- \to e^- + \overline{\nu}_e + \pi^0 \quad (5.07\%) \end{gathered}$$

Kaon branching ratios are taken from [24].
Charged pions and muons from the kaon decays subsequently also decay like it is shown on the left side. The dominant decay channel for the neutral pion (π^0) with about 98.8% [25] is $\pi^0 \to 2\gamma$. Here the decay chains for the negatively charge particles are shown, the reactions for π^+ and K^+ are identical but with neutrinos and anti-neutrinos exchanged (and all charges positive).

Most of the decay chains end with a muon that further decays into an electron. In the muon production one muon neutrino and in the muon decay one muon and one electron neutrino is created. If all muons decay the fraction of muon and electron neutrinos would be

$$R \equiv \frac{\#(\nu_\mu + \overline{\nu}_\mu)}{\#(\nu_e + \overline{\nu}_e)} \approx 2\,. \tag{2.3}$$

However, if the muon energy is high enough it may not decay on its way to the detector on the ground. This means that no electron neutrino is produced and the ratio R defined above increases for higher energies, which can be seen in fig. 2.6.

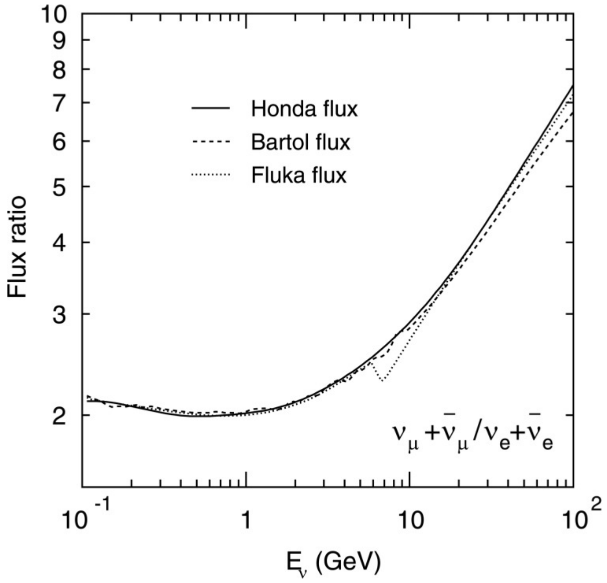


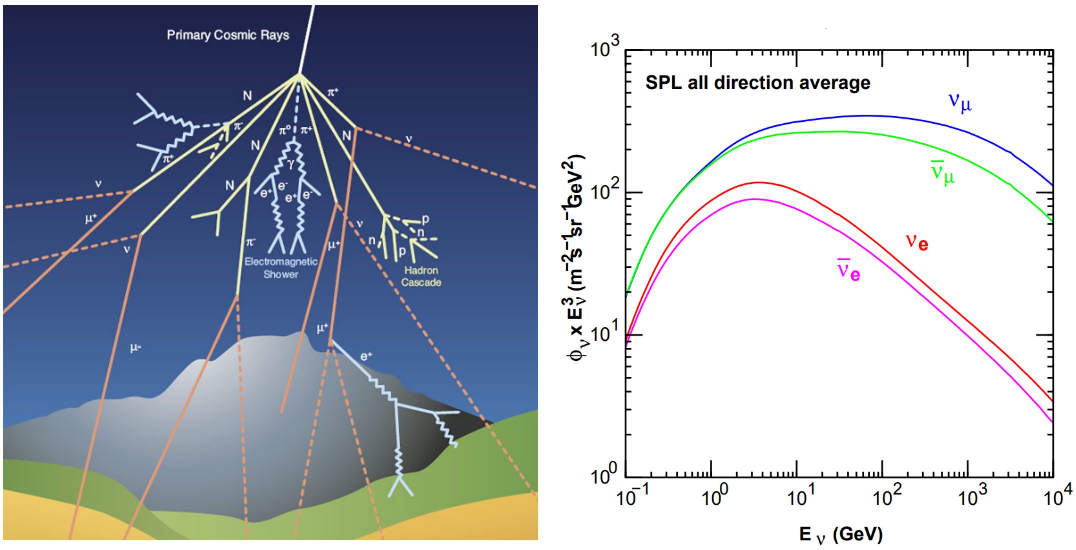
Figure 2.6.: Ratio of the atmospheric muon and electron neutrino flux versus the neutrino energy for three different flux models [26]. The sum of neutrino and anti-neutrino flux is used in each case. The ratio increases because at higher energies less atmospheric muons decay on their way to the ground.

Above a certain energy threshold ε interactions of the pions ($\varepsilon_\pi \approx 115\,\mathrm{GeV}$ [27]) and kaons ($\varepsilon_K \approx 850\,\mathrm{GeV}$ [27]) are favored over their decay. The ratio between the number of produced pions and kaons also depends on the energy. At low energies more pions are produced because their mass is smaller than the kaon mass which leads to a larger phase space for their production. Consequently, at low energies more neutrinos are created in pion decays, while at high energies more neutrinos occur from kaon decays. The right part of fig. 2.7 shows the measured flux of atmospheric electron and muon neutrinos and anti-neutrinos over their energy. The plot was made for the South Pole and averages over all directions and one year in time. One can see that more neutrinos are produced at lower energies and that R increases for higher energies.

2.1.4. Neutrino detection via Cherenkov radiation

Neutrinos are extremely difficult to detect. As mentioned several times before they only interact weakly and therefore have very low cross-sections. This makes a direct detection practically not feasible. A neutrino will not interact more than once in a detector nor emit any detectable radiation itself. However, there are multiple ways to indirectly detect these elusive particles. The one important for this thesis is Cherenkov radiation [29] produced by charged particles that are created following an neutrino interaction.

In deep inelastic scatter processes charged particles are produced (see fig. 2.3). If a



(a) Particle shower after cosmic ray interaction in the atmosphere of the Earth. Atmospheric neutrinos are produced as well as atmospheric muons. Figure taken from [28].

(b) Measured energy spectra of the atmospheric neutrino fluxes at the South Pole. The plot shows the averaged data over one year. Figure adapted from [23].

Figure 2.7.: Production and flux of atmospheric neutrinos.

charged particle flies through a dielectric medium (like ice) faster than the speed of light in this medium, Cherenkov radiation is emitted.

The speed of light in a medium is given by the vacuum speed of light c divided through the refractive index n of the medium. n is a dimensionless number which indicates the light diffractivity of the medium. It is always greater than 1, so light is never faster than c. For water respectively ice the refractive index varies between 1.35 and 1.30 for photon wavelengths between 250 nm and 900 nm [30].
On its way the charged particle polarizes the atoms in the dielectric medium. This is illustrated in the left part of fig. 2.8. If the velocity of the particle is faster than $\frac{c}{n}$ the polarization is asymmetric (case b in the plot) because the atoms can not rearrange fast enough. When the atoms depolarize, they emit electromagnetic waves. Constructive inference of this waves leads to a cone of light as can be seen in the right plot of fig. 2.8. This constructive inference is only possible starting from a asymmetric polarization. The principle is similar to the sonic boom "emitted" by a jet aircraft flying faster than the speed of sound.
The emission angle θ of the radiation relative to the direction of the charged particle can be calculated by:

$$\cos(\theta) = \frac{\frac{c}{n} \cdot t}{\beta \cdot c \cdot t} = \frac{1}{\beta n} . \tag{2.4}$$

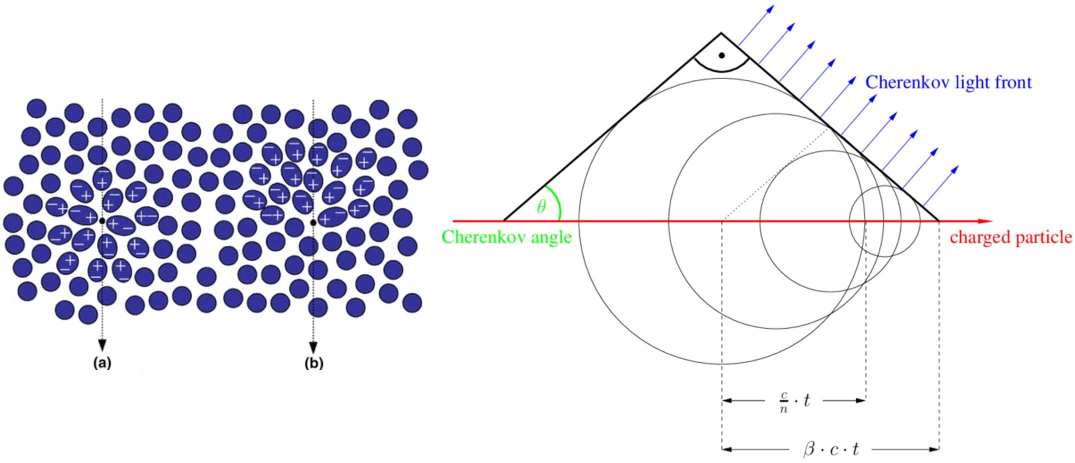


Figure 2.8.: Cherenkov radiation emergence. The left plot shows the polarization of atoms by a charged particle flying slower (a) and faster (b) than light in that medium. In the right plot the formation of a Cherenkov light cone is visualized. The left plot is based on [31], the right plot is taken from [32].

Equation 2.4 also shows that no radiation is emitted for $v < \frac{c}{n}$, where v is the speed of the charged particle. Assuming a refractive index of 1.33 this leads to a kinetic energy threshold of $E_{th,\mu} \approx 55\,\mathrm{MeV}$ for muons and $E_{th,e} \approx 0.26\,\mathrm{MeV}$ for electrons in water and ice [33]. Both energies are well below one GeV.

The number of emitted photons N per path length ∂x and wavelength interval $\partial\lambda$ is given by the Frank-Tamm formula [34]:

$$\frac{\partial^2 N}{\partial x \partial \lambda} = \frac{2\pi\alpha z^2}{\lambda^2}\left(1 - \frac{1}{(\beta n(\lambda))^2}\right), \tag{2.5}$$

where α is the fine-structure constant [35], β the speed of the charged particle divided by the (vacuum) speed of light c, z the electric charge of the particle, and $n(\lambda)$) the wavelength-dependent refractive index of the traversed material.

These photons can be detected by photosensors and thus lead to an indirect detection of neutrinos.

2.2. Neutrino oscillations

The first experiment that measured neutrinos produced in the sun was the Homestake experiment [9] by Raymond Davis Jr. in 1968. It had the goal to basically look inside a star and directly verify the hypothesis of nuclear energy generation in stars. Calculations of the number of nuclear fusion processes to explain the intensity of solar radiation on the Earth's surface lead to an expected number of neutrino interactions in the detector. The experiment successfully detected solar neutrinos, but only about one-third of the predicted number. Since this discrepancy was way too large to be explained by uncertainties and neither in the experiment nor in the theory errors could be found, it got the name solar neutrino problem. The solution to this problem, proposed by Bruno Pontecorvo [36] before the actual measurement and even earlier by Ziro Maki, Masami Nakagawa and Shoichi Sakata [16], builds on the fact that the Homestake detector was only sensitive to electron neutrinos. The sun only produces electron neutrinos, so it should be sufficient to only search for them. However, in case neutrinos have mass they could change their flavor on their way to the detector after being produced and therefore be invisible for the detector. This phenomenon is called neutrino oscillation.

The solar neutrino problem was seen as the first experimental hint for neutrino oscillation, but actually solar neutrinos do not really oscillate on their way to Earth. Solar neutrinos are produced in the electron flavor state which is a coherent superposition of all three mass eigenstates. However, in very dense media, like the core of the sun, the ν_e eigenstate is nearly identical to the ν_2 eigenstate. That means most of the neutrinos leave the sun as ν_2, which is a decoherent superposition of the three flavor states. ν_2 has a well defined mass and therefore can not change to another mass eigenstate with a different mass. When it arrives at the detector, ν_2 has a certain probability to be detected in each flavor. This probability is about a third for each flavor (c.f. 2.5) and explains the number of ν_e events being only a third of the prediction. The solar neutrino non-oscillation is explained in more detail in [37].

Actual neutrino oscillation was measured (and confirmed) with atmospheric neutrinos by the Super-Kamiokande experiment in 1998 [38]. The Earths atmosphere is a spherical source with the detector inside of it. According to Gauss's law [39] one would expect the same number of neutrinos arriving from each direction, even if the detector sits very close to one side of the sphere. Atmospheric neutrinos from each direction have the same energy (spectrum) but not the same distance to travel. So neutrinos from certain directions will oscillate more than others and one would observe for example less up-going muon neutrinos than down-going muon neutrinos. That is what Super-Kamiokande found. Figure 2.9 shows the number of observed neutrino events compared to the no oscillation prediction in grey and to the prediction with oscillations in black. The measurement can best be explained with the presence of oscillation.

Neutrino oscillation is another perfect example for the nature of quantum mechanics. To explain it neutrinos have to exist in different states at the same time and only when they interact they have to take one specific. While this seems unintuitive, it can be described mathematically. In Vacuum, the probability that a neutrino with energy E that was created in the flavor state α will interact in the flavor state β after traveling the distance L is given by:

$$P_{\alpha\to\beta} = |\sum_i \mathcal{U}_{\alpha i}\mathcal{U}^*_{\beta i}\,\exp\left(-i\frac{\Delta m^2_{ik}}{2E}L\right)|^2 \tag{2.6}$$

with $\Delta m^2_{ik} = m^2_i - m^2_k$ and $\mathcal{U}$ is defined in (2.1.2).

Note that this is only the probability that if the neutrino interacts it will be in that

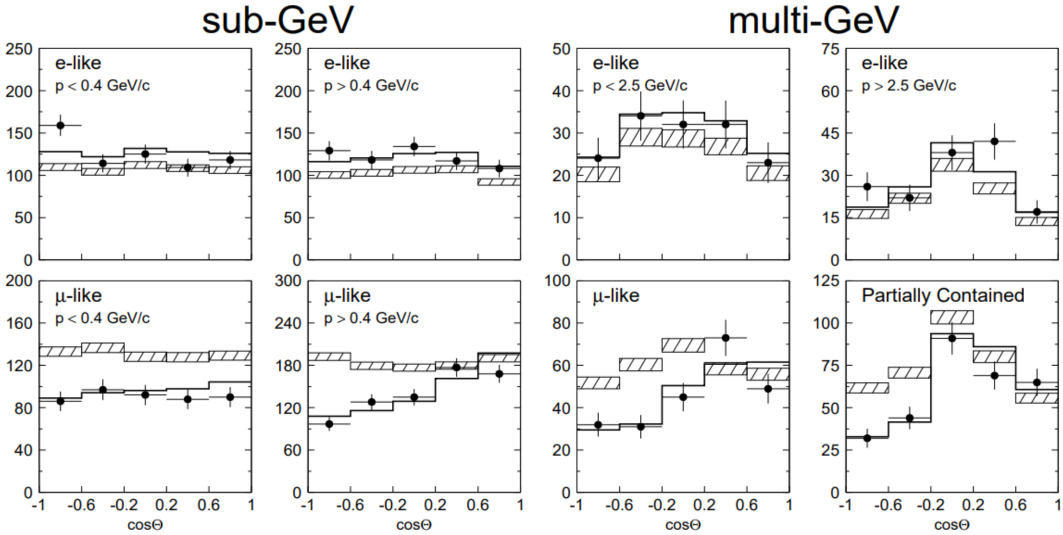


Figure 2.9.: Super-Kamiokande neutrino oscillation evidence [38]. The plot shows the zenith angle distribution of μ-like and e-like events in the detector. It is subdivided into different energy and momentum regions. The hatched region marks the expectation without neutrino oscillations, while the solid line shows the best-fit including oscillations. It can be clearly seen that the scenario with oscillations fits much better.

specific flavor. The probability that the neutrino interacts at all is much smaller and given by the cross-section of the specific interaction. So to get the probability of a specific interaction one has to multiply cross-section and oscillation probability.

2.2.1. Vacuum oscillations

The derivation of the vacuum oscillation probability is based on lecture notes by Hubert Spiesberger which are not publicly available anymore.

There are several ways to derive equation 2.6, but most of them do not get along without some inconsistencies. Since neutrino oscillation is a quantum mechanical phenomenon it is not possible to fully describe it with classical arguments. Elementary particles, as neutrinos are, do not travel along well-defined trajectories but well defined path and time are needed for a classical approach. So assumptions about energy and momentum of the neutrino states would be needed to compute the oscillation probability. An attempt to better describe the neutrino propagation is the wave package approach, where (for fixed energy or momentum) plane waves $\psi(\vec{x},t) = e^{i(\vec{p}\vec{x}-Et)}$ are assumed for path and time[1]. These waves are not localized and not moving but describe the location of the neutrino at every point in space-time. While this approach

---

[1] The full neutrino state in the flavor space would then be given by $|\nu_\alpha(\vec{x},t)\rangle = \sum_i \mathcal{U}_{\alpha i}\psi_i(\vec{x},t)\,|\nu_i\rangle$, where $|\nu_i\rangle$ are the mass eigenstates

describes the propagation well, it can not fully describe the production and detection mechanism. The approach that is considered to be most complete is the Quantum Field Theory (QFT) formalism (even though in practice it also needs approximations). Three steps are needed to derive the oscillation probability in the QFT framework:

1. First one has to define the initial and final state for the quantum mechanical system which consists of the particle (neutrino), the source and the detector state.
2. Next the interaction Hamiltonian $\mathcal{H}_{int}$ has to be determined. It basically transforms the initial to the final state. This includes defining field operators that act on the neutrinos (usually they create or annihilate the particle).
3. Finally, the oscillation probability $P_{\alpha\to\beta}$ can be calculated from transition amplitude for going from the initial to the final state.

Define the initial and the final state

The quantum mechanical system considered basically consists of the neutrino, its source (S) and the detector (D). So both the initial and final state have to describe these three components. For the source and the detector a excited (E) as well as a ground (G) state exists. The excited state is the one before emitting (for the source) respectively after absorbing (for the detector) the neutrino, while the ground state is the respective state after emitting/before absorbing the neutrino. In the following, the states are written as: $|S_E\rangle$, $|S_G\rangle$, $|D_E\rangle$, $|D_G\rangle$.
The source and the detector are assumed to be very heavy which means that the uncertainties on position and velocity given by Heisenberg‘s uncertainty principle are very small for them:

$$\Delta p \cdot \Delta x \geq \frac{1}{2} \xrightarrow[\text{large M}]{} \Delta v \cdot \Delta x \geq \mathcal{O}\left(\frac{1}{M}\right) .$$

For $M \to \infty$ S and D can therefore be considered as localized ($\Delta x \approx 0$) and with a precisely known velocity $\Delta v \approx 0$. That makes it possible to express source and detector in eigenstates of the position operator in the following way:

$$\langle \vec{x}' | \, e^{-iH_0\tau} \, |\vec{x}\rangle = \delta^{(3)}(\vec{x} - \vec{x}')e^{-iM\tau} + \mathcal{O}\left(\frac{1}{M}\right) .$$

Here $H_0 \simeq M + \frac{p^2}{2M}$ was used as well as the previously made assumption that the states are very heavy. τ denotes the time in the rest frame.

Ground and exited state for the source and the detector have different energies (and masses). The source looses energy $-\Delta_S$ by going from S_E to S_G, while the detector gains energy Δ_D when getting excited ($D_G \to D_E$). This leads to a phase difference of $e^{-i\Delta_S t}$ for the source and $e^{i\Delta_D t}$ for the detector.

Combining source, detector and neutrino yields the full state of the system. The initial $|i\rangle$ respectively final $|f\rangle$ state are defined as

$$\begin{aligned} |i\rangle &= |S_E\rangle \otimes |D_G\rangle \otimes |0_\nu\rangle \ , \\ |f\rangle &= |S_G\rangle \otimes |D_E\rangle \otimes |0_\nu\rangle \ . \end{aligned} \tag{2.7}$$

Both states do not actually contain a neutrino (represented as $|0_\nu\rangle$), because the initial state is defined as the state before the neutrino was emitted and the final state is the state after the absorption of the neutrino.

Determine the interaction Hamiltonian $\mathcal{H}_{int}$

After defining the initial and final state, the next step is to define how the former is transformed to the latter. The interaction responsible for that is described by the interaction Hamiltonian, which basically describes the time evolution of the states.

In the spectral representation quantum mechanical operators can be decomposed in their eigenvalues. For the Hamiltonian that reads

$$\begin{aligned} \text{for source:} \quad & e^{-i\Delta_S t} \, |S_G\rangle \langle S_E| \ , \\ \text{for detector:} \quad & e^{i\Delta_D t} \, |D_E\rangle \langle D_G| \ . \end{aligned} \tag{2.8}$$

Simplified speaking, because of the orthogonality of the eigenstates $\langle S_i|S_j\rangle = \delta_{ij}$, the source operator for example basically replaces the state $|S_E\rangle$ with the state $|S_G\rangle$ and adds the energy (or mass) difference $-\Delta_S$ (which means it subtracts the energy Δ_S).

The interaction leads to a neutrino production. As mentioned before neutrinos interact in their flavor state and therefore the neutrino is produced in a flavor eigenstate $|\nu_\alpha\rangle$ at the source. Then the neutrino propagates to the detector, but not as flavor but as mass eigenstates. The mixing between the flavor and the mass eigenstates $|\nu_i\rangle$ is represented by the PMNS matrix (see sec. 2.1.2). At the detector the neutrino is absorbed in the flavor state $|\nu_\beta\rangle$ (because absorption is an interaction again), which can be but does not have to be the same flavor state as $|\nu_\alpha\rangle$. So the neutrino in the flavor and mass eigenstate representation is given by

$$\begin{aligned} \text{for source:} \quad & |\nu_\alpha\rangle = \sum_i \mathcal{U}_{\alpha i} \, |\nu_i\rangle \ , \\ \text{for detector:} \quad & |\nu_\beta\rangle = \sum_i \mathcal{U}^*_{\beta i} \, |\nu_i\rangle \ . \end{aligned} \tag{2.9}$$

Emission ($S_E \rightarrow S_G + \nu_\alpha$) and absorption ($D_G + \nu_\beta \rightarrow D_E$) of the neutrino happen via the weak interaction, because neutrinos do not participate in other interactions. So common weak coupling with coupling constant g can be assumed for both.

Now only the neutrino propagation has to be added to the Hamiltonian. This can be done by using Dirac field operators ϕ for the mass eigensates ($|\nu_i\rangle \to \phi_i(t,\vec{x})$). With the "rotating wave approximation" that separates creation and annihilation of the neutrino, the field operators can be written as

$$\begin{aligned}\phi_i^{(-)}(t,\vec{x}) &= \frac{1}{(2\pi)^3}\int \frac{d^3p}{2E_p}\, a^{+}_{\vec{p},i}\, e^{-i(E_p t-\vec{p}\vec{x})}\,,\\ \phi_i^{(+)}(t,\vec{x}) &= \frac{1}{(2\pi)^3}\int \frac{d^3p}{2E_p}\, a_{\vec{p},i}\, e^{i(E_p t-\vec{p}\vec{x})}\,.\end{aligned} \tag{2.10}$$

Here a and a^{+} are the quantum mechanical particle annihilation and creation operator for the neutrino.

The complete Hamiltonian is then given by

$$\begin{aligned}\mathcal{H}_{int}^{\alpha\to\beta}(t) = g\sum_i \{&\mathcal{U}_{\alpha i}\,\phi_i^{(-)}(t,0)\, e^{-i\Delta_S t}\,|S_G\rangle\langle S_E| + \\ &\mathcal{U}^*_{\beta i}\,\phi_i^{(+)}(t,\vec{L})\, e^{i\Delta_D t}\,|D_E\rangle\langle D_G|\}\,.\end{aligned} \tag{2.11}$$

Calculate $P_{\alpha\to\beta}$ from transition amplitude

Now that the initial and final state as well as the Hamiltonian are defined, the transition probability can be calculated. The first step is to calculate the transition amplitude $A_{\alpha\to\beta}$ from the S-matrix ($S = \mathcal{T}e^{-i\int dt\mathcal{H}_{int}(t)}$. Here $\mathcal{T}$ is the time ordered product (sometimes also called time-ordering operator). The transition amplitude is simply given by

$$A_{\alpha\to\beta} = \langle f|\, S\, |i\rangle\ . \tag{2.12}$$

To perform the calculation the Taylor expansion of the e-function ($\exp(x) = 1 + x + \frac{x^2}{2} + ...$) is used. At second order[2] this reads

$$A_{\alpha\to\beta} = \frac{(-i)^2}{2}\int_{\tau}^{\tau+T} dt_1 \int_0^{t_1} dt_2\, \langle f|\,\mathcal{H}_{int}(t_1)\mathcal{H}_{int}(t_2)\,|i\rangle\ . \tag{2.13}$$

The first time integral represents the detector which is active in the time interval [τ, τ+T] and the second time integral represents the activity of the source at all times $t_2 < t_1$.

Considering the orthogonality of the quantum mechanical states ($\langle K_i|L_j\rangle = \delta_{ij}\delta_{KL}$ with $K, L \in [S, D]$ and $i, j \in [G, E]$) the bracket simplifies to

---

[2] The zeroth and first order term vanish because of the orthogonality of the eigenstates.

$$\begin{aligned}\langle f|\,...\,|i\rangle &= g^2\sum_i \mathcal{U}_{\alpha i}\mathcal{U}^*_{\beta i}\, e^{-i\Delta_S t_2}e^{i\Delta_D t_1}\,\langle 0_\nu|\,\phi_i^{(+)}(t_1,L)\phi_i^{(-)}(t_2,0)\,|0_\nu\rangle \\ &= g^2\sum_i \mathcal{U}_{\alpha i}\mathcal{U}^*_{\beta i}\, e^{-i\Delta_S t_2}e^{i\Delta_D t_1}\int \frac{d^3p}{(2\pi)^3}\frac{1}{2E_p}e^{-i(t_1-t_2)E_{\vec{p}}+i\vec{p}\vec{L}}\,.\end{aligned} \tag{2.14}$$

For the same reason the first two terms of the Taylor expansion vanish, because one has to replace the source and detector part in the initial state to not get a 0 but the Hamiltonian contains only one bra (and ket) vector.

The transition probability is the square of the absolute transition amplitude

$$P_{\alpha\to\beta} = |A_{\alpha\to\beta}|^2\,.$$

At this point all necessary parts are defined and only the mathematical calculation is left. The result of the two 1-dimensional time integrals and the 3-dimensional momentum integral is

$$P_{\alpha\to\beta} = \frac{g^4}{\pi L^2}\,|\sum_i \mathcal{U}_{\alpha i}\,\mathcal{U}^*_{\beta i}\, e^{-i\frac{m_i^2 L}{2\Delta_S}}\,|^2\,. \tag{2.15}$$

The only difference between this result and the oscillation formula shown in equation 2.6 is the (geometric) suppression term $\frac{g^4}{\pi L^2}$. This suppression occurs because here the probability to measure a $|\nu_\beta\rangle$ at the detector starting with a $|\nu_\alpha\rangle$ at the source is shown, which is the product of the probability to measure the neutrino times the oscillation probability. Δ_S, which for reasons of energy conservation has to be equal to Δ_D, can be identified as the neutrino energy.

The principle of neutrino oscillation does not change for two (or more than three flavor). For many purposes it is sufficient to look at the simpler two flavor case, where only one mixing angle (θ) and one mass difference (Δm^2) exist. The oscillation probability formula then is reduced to

$$P_{\alpha\to\beta} \approx \sin^2(2\theta)\cdot\sin^2\left(\frac{\Delta m^2 L}{4E}\right)\,. \tag{2.16}$$

Here $\sin(x) = \frac{e^{ix}-e^{-ix}}{2i}$ was used.
This simplified form demonstrates the effect of the oscillation parameters. The mixing angle represents the amplitude and the squared mass difference the frequency of the oscillation.

2.2.2. Matter effects (MSW effect)

As mentioned before, neutrinos only interact weakly and therefore have very small cross-sections. Consequentially, neutrinos traverse large amounts of matter without being absorbed or deflected. Nevertheless, neutrinos “feel” matter effects even if they cross it apparently undisturbed. Elastic forward scattering of neutrinos does not change their momentum or direction, but has an impact on neutrino oscillations in matter. This effect is called MSW effect, after Stanislav Michejew, Alexei Smirnow and Lincoln Wolfenstein, and was first introduced by Lincoln Wolfenstein in 1978 [40]. Elastic forward scattering changes the oscillation properties of neutrinos because it influences the different flavor in a different way. Normal matter contains electrons (mostly in bound states), but no muons or taus. So for the electron neutrino flavor charged current interactions are possible in matter. All neutrino flavor can scatter via the exchange of a Z-boson (neutral current, right Feynman diagram in fig. 2.10). In addition, electron neutrinos can scatter via a W-boson (charged current, middle Feynman diagram in fig 2.10) and electron anti-neutrinos can be annihilated and recreated by a W-boson interaction with an electron (charged current, left Feynman diagram in fig 2.10).

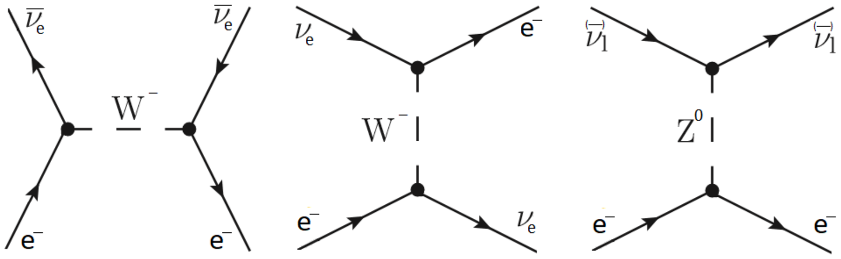


Figure 2.10.: Neutrino interactions in matter. **Left**: CC annihilation of an electron anti-neutrino. **Middle**: CC scattering of an electron neutrino. **Right**: NC scattering of all neutrino and anti-neutrino types.

This leads to higher cross-sections for $\nu_e e \rightarrow \nu_e e$ scattering compared to $\nu_{\mu,\tau} e \rightarrow \nu_{\mu,\tau} e$ [41].

So the electron neutrino component “feels” a different matter potential compared to the other neutrino flavors. The principle does not change considering the three or two flavor case, so the rest of this section will focus on the two flavor case as shown in [42]. Only the electron flavor sees a effective matter potential, so in the flavor base it can be expressed through:

$$V_{matter} = \begin{pmatrix} \sqrt{2}G_F n_e & 0 \\ 0 & 0 \end{pmatrix} \tag{2.17}$$

where G_F is the Fermi coupling constant and n_e the electron density in the traversed matter. In the following $V = \sqrt{2}G_F n_e$ is used.

To get the neutrino Hamiltonian in matter, this potential has to be added to the Vacuum Hamiltonian. For the two flavor example the matter Hamiltonian then looks like

$$\begin{aligned} H =& \mathcal{U} \begin{pmatrix} 0 & 0 \\ 0 & \dfrac{\Delta m^2}{2E} \end{pmatrix} \mathcal{U}^\dagger + \begin{pmatrix} V & 0 \\ 0 & 0 \end{pmatrix} \\ =& \frac{\Delta m^2}{2E} \begin{pmatrix} 1 - \cos(2\theta) + 2EV/\Delta m^2 & \sin(2\theta) \\ \sin(2\theta) & 1 + \cos(2\theta) \end{pmatrix} . \end{aligned} \tag{2.18}$$

The oscillation probability can be calculated with the S-matrix $S = e^{-iH}$ in the same way as before

$$P_{\alpha\to\beta} = |S_{\beta\alpha}|^2 . \tag{2.19}$$

The detailed calculations are shown in [42] for two and three flavor and [43] for the three flavor case. The result looks very similar to the Vacuum result

$$P^{matter}_{\alpha\to\beta} = \frac{\sin^2(2\theta)}{\sin^2(2\theta) + (\cos(2\theta) - \frac{2EV}{\Delta m^2})^2} \sin^2\left(\frac{\Delta m^2 L}{4E} \sqrt{\sin^2(2\theta) + \left(\cos(2\theta) - \frac{2EV}{\Delta m^2}\right)^2} \right) .$$

Introducing the effective squared mass difference and mixing angle in matter

$$\begin{aligned} \Delta m^2_{eff} &= \Delta m^2 \sqrt{\sin^2(2\theta) + \left(\cos(2\theta) \mp \frac{2VE}{\Delta m^2}\right)^2} \\ \sin(2\theta_{eff}) &= \frac{\sin(2\theta) \cdot \Delta m^2}{\Delta m^2_{eff}} \end{aligned} , \tag{2.20}$$

where the minus occurs for neutrinos and the plus for the same calculation with anti-neutrinos, allows to rewrite equation 2.2.2 to

$$P^{matter}_{\alpha\to\beta} = \sin^2(2\theta_{eff}) \cdot \sin^2\left(\frac{\Delta m^2_{eff} L}{4E}\right) . \tag{2.21}$$

It now has the exact same structure as equation 2.16. This basically shows that matter does not fundamentally change neutrino oscillations but it changes the oscillation parameters Δm^2 (frequency) and $\sin(2\theta)$ (amplitude).

The calculation for the three flavor case is very similar. The Hamiltonian for that case is

$$H = \mathcal{U} \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta m^2_{21}}{2E} & 0 \\ 0 & 0 & \frac{\Delta m^2_{31}}{2E} \end{pmatrix} \mathcal{U}^\dagger + \begin{pmatrix} V & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} . \tag{2.22}$$

More details on the 3-flavor calculation can be found in [43] and [42].

The strength of the matter potential V (and therefore the MSW effect itself) depends on the electron density n_e of the traversed medium. The electron density on the other hand depends on the matter density ρ

$$n_e = Y \cdot \frac{\rho}{m_N} \tag{2.23}$$

where Y is the electron fraction in the material and m_N the nucleon mass.

All these quantities vary for different materials. For example, according to the Preliminary Reference Earth Model (PREM, [44]) the matter density in the Earth varies between roughly 1 and 13 $\frac{g}{cm^3}$ for the other crust and inner core. Atmospheric neutrino experiments rely on a precise knowledge of the density profile of the Earth to calculated the expected oscillation probabilities.

Figure 2.11 shows the probability for a muon neutrino which traversed nearly the whole Earth (11.500 km) to oscillate into an electron neutrino. Different models for the matter density (ρ) distribution in the Earth are included: A model with constant ρ (grey), a simple 3 layer model with two constant ρ layers for the mantel and one constant ρ layer for the core (green), a optimized 3 layer model where ρ_{mantle} and ρ_{core} are calculated specifically for the chosen baseline (red) and the accurate treatment using small slabs of constant density (black). For a baseline of $L = 2R_{Earth}$ the red and the green curve would be identical. The vertical lines indicate the positions of the resonances ($\sin^2(2\theta_{eff}) = 1$) for the 3 layer model, on the left the core and on the right the two mantle layers.

The plot shows the strong dependence of neutrino oscillations on the traversed density profile.

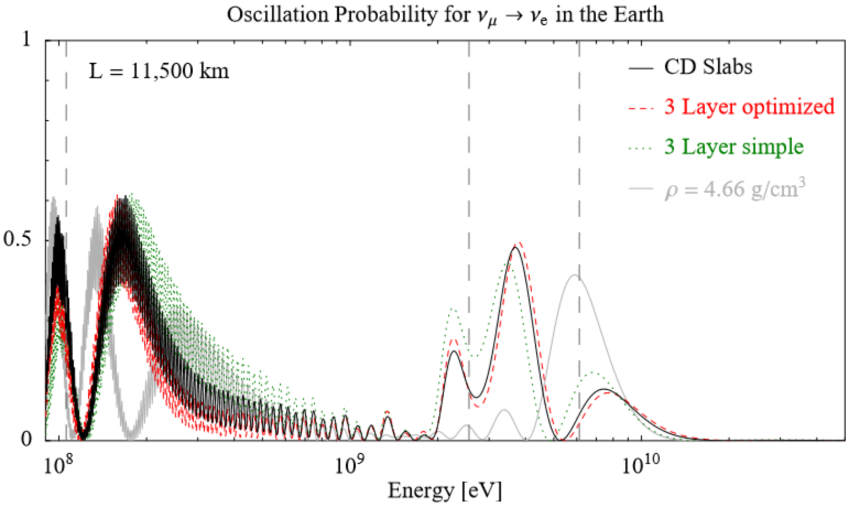


Figure 2.11.: $\nu_\mu \rightarrow \nu_e$ oscillation probability for neutrinos which traversed nearly the whole Earth [42]. Different Earth matter density models are compared. The oscillation parameter values used for this calculation are: $\sin^2(2\theta_{12}) = 0.79$, $\sin^2(2\theta_{23}) = 1.0$, $\sin^2(2\theta_{13}) = 0.05$, $\Delta m^2_{21} = 8.1 \cdot 10^{-5}\,\mathrm{eV}^2$, $\Delta m^2_{31} = 2.2 \cdot 10^{-3}\,\mathrm{eV}^2$ and $\delta_{CP} = 0$.

3. Parameter estimation

In (particle) detectors measured data $\boldsymbol{x}$ are observed. For example, the change over time of an electric current. More interesting than the data itself, however, is the underlying cause that produced them (e.g. a particle interacting inside the detector volume). The properties of this underlying cause have to be described by a model using a well defined set of parameters $\boldsymbol{\theta}$ (e.g. energy of that particle and point of the interaction). The challenge is now to determine the most probable values for all or some $\boldsymbol{\theta}$ parameters based on the measurement $\boldsymbol{x}$. This process is called parameter estimation or reconstruction. In addition to the most probable values, confidence levels or posterior distributions for the parameters are also of interest.

A frequently used reconstruction method is the so-called maximum likelihood estimation. For this method, a likelihood function (sec. 3.2) has to be constructed that connects $\boldsymbol{x}$ and $\boldsymbol{\theta}$ via probability distributions. Maximizing this function yields the best-fit $\boldsymbol{\theta}$ values based on $\boldsymbol{x}$. Another inference method are direct parameter estimators which use machine learning (sec. 3.3) to gain knowledge about the relation between $\boldsymbol{x}$ and $\boldsymbol{\theta}$. They are trained on large amounts of simulated data and directly predict the most probable $\boldsymbol{\theta}$ values without the need of a minimization (in the inference process). It is also possible to combine likelihood and machine learning efforts, for example using likelihood-free inference (sec. 3.4). Here, machine learning is used to learn the likelihood function.

3.1. Probability

The concept of probability is essential for parameter inference in particle detectors. Because of the randomness of the involved processes, like for example photon propagation or particle decays, the exact same $\boldsymbol{\theta}$ can produce different $\boldsymbol{x}$ if repeated multiple times. Conversely, two different $\boldsymbol{\theta}$ can produce the same $\boldsymbol{x}$ because of the limited detector resolution. Therefore, even the perfect reconstruction method will not always return the true $\boldsymbol{\theta}$ values but the most probable values given the data $\boldsymbol{x}$. Thus, it has to be defined what most probable means.

There are two main interpretations of probability, the frequentist and the Bayesian interpretation. Both are shortly introduced in the following.

Frequentist interpretation

In the frequentist interpretation, probability is defined as the relative frequency of occurrence in the limit of infinite trials. For example the probability $P(1)$ to role a one with a dice would be obtained by

$$P(1) = \lim_{N \to \infty} \frac{N_1}{N}\,, \tag{3.1}$$

where N is the number of trials and N_1 the number of ones rolled.
This concept works for repeatable observations but not if the probability should be defined for a unique event, e.g. what the probability is for a specific team to win the championship this year. In the frequentist interpretation this probability would either be 0 or 1, but it is unknown which.

Bayesian interpretation

In the Bayesian interpretation, probability is defined as the degree of belief based on personal prior knowledge. This is a subjective definition because there is no general prescription for priors. It is possible to assign a probability even to unique events like the question about the probability for a specific team to win the championship this year. However, this probability value will differ depending if a person is asked that knows a lot about sports compared to a person without this knowledge.
The name Bayesian interpretation goes back to Bayes' theorem for conditional probabilities $P(\boldsymbol{\theta}|\boldsymbol{x})$. In general, the probability for $\boldsymbol{\theta}$ and $\boldsymbol{x}$ to occur together is:

$$P(\boldsymbol{x} \cap \boldsymbol{\theta}) = P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})\,, \tag{3.2}$$

so the probability for $\boldsymbol{\theta}$ to be true multiplied with the conditional probability observing $\boldsymbol{x}$ given $\boldsymbol{\theta}$. Bayes' theorem states that $P(\boldsymbol{\theta} \cap \boldsymbol{x}) = P(\boldsymbol{x} \cap \boldsymbol{\theta})$ and that this can be rewritten as:

$$P(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\boldsymbol{x})}\,, \tag{3.3}$$

where $P(\boldsymbol{\theta}|\boldsymbol{x})$ is the so-called posterior, $P(\boldsymbol{x}|\boldsymbol{\theta})$ is the likelihood (see sec. 3.2), $P(\boldsymbol{\theta})$ is the prior, and $P(\boldsymbol{x})$ is called evidence.
Bayes' theorem basically defines, for certain prior probabilities, how these probabilities should change given the data.

For both interpretations the Kolmogorov axioms [45] of probability apply:

- $P(A) \in \mathbb{R}, P(A) \geq 0$: Probabilities are real numbers greater or equal to 0.
- $P(\Omega) = 1$: The total probability is 1.
- $P(\bigcup_{i=1}^{N} A_i) = \sum_{i=1}^{N} P(A_i)$: The probability for mutual exclusive events to happen together is the sum of the individual probabilities.

The difference between the two interpretations is what they consider as random variable. For Bayesians $\boldsymbol{\theta}$ is treated as random variable to which probabilities can be attributed, while for a Frequentist $\boldsymbol{\theta}$ is fixed but unknown and the parameters of the $\boldsymbol{x}$ distributions are random variables.

If probabilities should be assigned to a continuous spectrum x, a so-called Probability Density Function (PDF) $p(x)$ is used. It defines the probability of $a \leq x \leq b$ as follows:

$$P(a \leq x \leq b) = \int_a^b p(x)dx\,. \tag{3.4}$$

3.2. Likelihood function

The likelihood function $L(\boldsymbol{\theta}|\boldsymbol{x})$ is used to calculate the conditional probability of the assumed model parameters $\boldsymbol{\theta}$ given the observed data $\boldsymbol{x}$. The basic idea of the concept goes back to Ronald Fisher [46, 47].

Let us assume that $p(\boldsymbol{x}|\boldsymbol{\theta})$ is the probability distribution (probability mass function for discrete and probability density function for continuous variables) for observing a specific value of $\boldsymbol{x}$ given $\boldsymbol{\theta}$. The likelihood for a concrete measurement $\boldsymbol{x}_1$ is then $L(\boldsymbol{\theta}|\boldsymbol{x}_1) = p(\boldsymbol{x}_1|\boldsymbol{\theta})$. Here only $\boldsymbol{\theta}$ is variable, while $\boldsymbol{x}_1$ is fixed. In case of multiple measurements $\vec{\boldsymbol{x}} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ the likelihood is calculated by the product of the individual probabilities

$$L(\boldsymbol{\theta}|\vec{\boldsymbol{x}}) = \prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\theta})\,. \tag{3.5}$$

To find the optimal values of the model parameters $\boldsymbol{\theta}_{opt}$, $L(\boldsymbol{\theta}|\vec{\boldsymbol{x}})$ has to be maximized with respect to $\boldsymbol{\theta}$.

Note that while $p(\boldsymbol{x}|\boldsymbol{\theta})$ is a PDF, $L(\boldsymbol{\theta}|\vec{\boldsymbol{x}})$ is not, since it is not normalized to 1 and therefore violates Kolmogorov's second axiom. The correct normalization is only given by Bayes' theorem.

Because the values of L are usually very small, it is common to use the logarithm of the likelihood $\mathcal{L} = \ln(L)$ (or LLH) to ensure numerical stability. Instead of maximizing $\mathcal{L}$, depending on the optimization algorithm of choice, minimizing $-\mathcal{L}$ can be preferred.

According to the *likelihood principle* [48, 49], the correct likelihood function yields the full information that can be extracted from the measurement. That does not mean that a reconstruction based on the true likelihood will always yield the true values of the $\boldsymbol{\theta}$ parameters, but the most probable given the information provided by the detector. How close these values are to the truth depends on the detector.

An advantage of having a likelihood function is that it can be evaluated everywhere in the $\boldsymbol{\theta}$ space. Since it strongly depends on the number of individual measurements, it is difficult to interpret the absolute value of the likelihood function. However, ratios between likelihood values (equivalent to differences of the log likelihood) can be used to compare how much the data support different $\boldsymbol{\theta}$. This is sometimes referred to as *law of likelihood* [1]. In fact, the likelihood ratio is the most powerful test-statistic to

compare two $\boldsymbol{\theta}$ [50] in the light of a measurement $\boldsymbol{x}$. Different parameter values can be compared as well as different numbers of free parameters.
The former is used to find the optimal values $\boldsymbol{\theta}_{opt}$ for the model parameters. In addition, it is possible to calculate the extent to which certain parameter values are preferred over others, and consequently to calculate confidence levels for the parameters. The later can be used to test the model. It opens the possibility to find out how important a specific model parameter is to explain the data. While additional free parameters will always lead to an equal or better modeling, the important question is how much they improve the optimal likelihood value compared to the other parameters. In general, an attempt should be made to use a model that is as simple as possible (has as few parameters as necessary) [51].

A not insignificant disadvantage of the likelihood approach on the other hand is that one has to exactly know $p(\boldsymbol{x}|\boldsymbol{\theta})$ to compute the likelihood. While this might be possible for simple experiments, it is not for any real world particle physics detector. The probability distributions for a real detector are just too complicated to be fully known. That means $p(\boldsymbol{x}|\boldsymbol{\theta})$ has to be approximated, which can be associated with high computational effort and high effort on the part of physicists who must model all the physical phenomena involved. In addition, approximations lead to worse reconstruction results.
Therefore, although the probability approach is in principle the best possible, its implementation is often very difficult.

3.3. Machine learning

Machine learning describes a brought set of techniques where computer algorithms or more generally artificial systems learn in order to improve performance on a task by extracting knowledge from data. For a specific task it is not necessary to explicitly tell the system how to solve it. Rather the system will learn a solution by iterating through the data multiple times. In a successful learning process, the data are not just memorized, but patterns and relations in the data are "understood". The quality of a trained system is defined by its performance on data that were not part of the training process.
Often machine learning is seen as part of or equated with artificial intelligence, but while there are intersections, machine learning in general does not require acting independently, drawing conclusions or extrapolate the learned.

A simple example for machine learning would be an affine regression. Let us assume a set of noisy data (x_i, y_i) generated from $y_i = a \cdot x_i + b + \Delta_n$, where Δ_n is random (e.g. Gaussian) noise. The true values of a and b are unknown, but it can be assumed that the functional relation between x and y can be described by $y = \alpha \cdot x + \beta$. That is the model (function) and it has two free parameters namely α and β. The next step is to define a so-called loss function that describes how well

the model predicts the data. Here, for example the squared difference of the model prediction and the actual value $\sum_i (y_i - \alpha \cdot x_i + \beta)^2$ can be used. Minimizing this loss with respect to α and β over the data points is nothing but learning a and b. With an infinite amount of data, a and b could be learned perfectly.
For most tasks the learning process is to minimize a loss function which expresses the difference of the expected and the actual system behavior. Another example would be a binary classification where the loss function could be the mean of the absolute difference between the predicted and the actual label. Minimizing this function over the whole training set will lead to the optimal possible classifier.

Machine learning is used for example for medical applications [52, 53], speech recognition [54,55], in autonomous systems [56], or for playing a game [57,58]. It is also widely used in particle physics. Noise rejection [59], particle identification [60], and also parameter estimation [61] are suitable tasks for machine learning.

What makes machine learning so useful for particle physics is the fact that it only requires simulated data to train on. As mentioned in 3.2, $p(\boldsymbol{x}|\boldsymbol{\theta})$ is usually untraceable. However, forward simulation, for example Monte Carlo (MC [62]) simulation, is feasible with acceptable computational effort. So while it is not possible to analytically write down $p(\boldsymbol{x}|\boldsymbol{\theta})$, it is possible to effectively draw from the distribution and show the results to a machine learning algorithm. Thus the algorithm has access to $p(\boldsymbol{x}|\boldsymbol{\theta})$ without the need to provide the function explicitly. In addition, a trained model is fast to evaluate (especially on GPUs [63]).
Disadvantages are that it is not trivial to access the internal representation of the data in the artificial systems. That makes it difficult to optimize machine learning models as it is not clear which part prevents the model from a better understanding of the data. Machine learning algorithms are also bad at extrapolating beyond what they have learned and prone to biases [64]. It must therefore be ensured that the training data cover the whole parameter space of interest as uniformly as possible. Another problem might be the amount of available training data. Although having training data at all is sufficient to train a model, a large amount may be required to get a good model. From the point of view of reconstruction, it is disadvantageous that for a specific $\boldsymbol{x}$ the model only returns a specific $\boldsymbol{\theta}$, without providing any other information. For parameter uncertainties or parameter importance measures, additional machine learning models would be required.

3.3.1. Categories

Depending on the available data (or better the information that can be extracted from them) and the given task (e.g. regression, clustering or playing a game) one can roughly distinguish between three main categories of machine learning: Supervised learning, unsupervised learning and reinforcement learning.

Supervised learning

Supervised learning is probably the most common category. Here the artificial system is trained on labeled data. For each step in the training process the desired output (true label y_{true}) is known and compared to the actual output of the system (predicted label y_{pred}). This comparison is done with a loss function, which yields smaller values if y_{true} and y_{pred} match better. Minimizing the loss function will lead to better predictions of the model and is called training. The minimization is done via so-called backpropagation [65]. First, the model evaluates the training data and calculates the loss based on this prediction and the truth. Then, the derivative of the loss function is taken with respect to all free parameters of the model, so a gradient-based minimizer can change these parameters to reduce the loss value. This is repeated until no more improvement is achieved on an independent data set that was not part of the training and is used to avoid overfitting the training data.

Supervised learning problems can be divided into two main types:

- **Regression:** learning to predict the output of a continuous function. For example predicting the energy of a particle based on some detector data.
- **Classification:** sorting the input into different classes, which is basically mapping onto a discrete function. Noise or particle identification are examples for this type.

Parameter estimation, in the context of a particle detector, falls into this category.

Unsupervised learning

If the data are unlabeled, there is no desired output which the artificial system can learn to predict. In unsupervised learning it searches for similarities or anomalies in the data. Thus, the system tries to identify clusters or hidden patterns that are not predefined by its creator. This is done by trying to mimic the data and consequently learning about their structure. Therefore, unsupervised learning is well suited for generative tasks or image and pattern recognition. It can also be used for feature reduction, for example with a principal component analysis [66] or a singular value decomposition [67].

Reinforcement learning

Reinforcement learning does not require data but an environment in which the artificial system, called agent, cant take actions and earns a reward for it. The learning process consist of multiple tries of the agent to gain the highest possible reward from the environment. Actions that lead to a higher reward at the end are given preference. In that way the agent learns the optimal behavior in the given environment. Reinforcement learning can be seen as the part of machine learning that is closest to artificial intelligence.

Machine learning is a constantly evolving field and there are applications that do not perfectly fit in any of these categories.

3.3.2. Methods

Many different methods can be used in machine learning. As mentioned before an affine regression is (a rather simple but still) a machine learning method. The same is true for a linear and a logistic regression. More general methods include for example random forests [68], (boosted) decision trees (e.g. [69]), or support vector machines [70]. This list is far from being complete, and new methods are constantly being developed.

Some methods are mainly used in one machine learning category. For example clustering methods like k-means clustering [71] solve unsupervised learning problems, while Q-learning [72] is a method used in reinforcement learning.

Neural networks

Artificial neural networks have become the most successful machine learning method in recent years. They are inspired by the biological neuron that forms the brain. The basic idea goes back to McCulloch and Pitts in 1943 [73]. Rosenblatt succeeded in making the first simple implementation (called perceptron) in 1957 [74].

An artificial neuron basically represents the function shown in fig. 3.1. It gets n input values $x_1, x_2, ..., x_n$, where n can be any integer 1 or higher. The input values are provided by other neurons or are the inputs of the network itself. Each input is assigned a weight $w_1, w_2, ..., w_n$ that represents how important the connection between the two neurons is. Everything the neuron does is to sum up all its weighted inputs and add a bias b. The result is then inserted into an activation function f, which calculates the output of the neuron. The purpose of the activation function is to only pass important information and to make the calculation non-linear. A famous example is the Rectified Linear Units (ReLU) function which is 0 for negative values and the identity elsewhere. Thus important information encoded as high positive output values can pass through while less important information is filtered out. For more details on activation functions, see [75].

The weights as well as the bias are trainable parameters. That means that during the training (or learning) process they are changed to achieve the optimal result. Similar to the learning process in the human brain some connections between neurons become more important (high weight) and also some neurons get a higher chance to be activated (high bias).

There are many types of neural networks but they mainly differ in the way the neurons are connected to each other and the structure of the input. The most interesting for this thesis is the multilayer perceptron (MLP), often referred to as densely or fully connected neural network. It consists of multiple layers of neurons and connects each neuron in a layer to every neuron of the previous layer. The first and the last layer

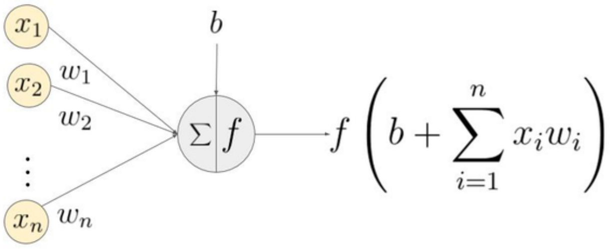


Figure 3.1.: Function represented by an artificial neuron. It gets n inputs $(x_1, ..., x_n)$ and returns a single (float) output value. Each input x_i is multiplied by a weight w_i and added up. A bias b is added to this sum. Finally, the result is inserted into an activation function f. Weights and bias are the free trainable parameters of the artificial neuron. This is the fundamental building unit of most neural networks.

are called input respectively output layer. All layers in between are hidden layers. At least one hidden layer is needed in a MLP. Figure 3.2 shows a sketch of a MLP.

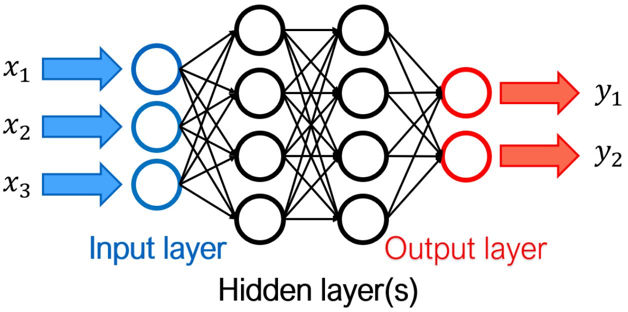


Figure 3.2.: Sketch of a multilayer perceptron or fully connected artificial neural network with two hidden layers.

A fully connected network with only a single hidden layer can already approximate any Borel measurable function if it consist of enough hidden neurons ("Universal Approximation Theorem" [76]). However, deep neural networks, i.e. networks with many hidden layers, are usually used because they need fewer neurons to achieve acceptable results. Training deep networks is called deep learning [77].
How a network is trained depends on the task it is trained for. For this thesis supervised learning is the most important category of machine learning. As explained in sec. 3.3.1, backpropagation is used to calculate the gradient of the loss function with respect to the network weights over the training data. Changing the weights opposite to the gradient reduces the loss function. This is called gradient descend. In case of large training samples and many network weights, this calculation can become computationally expensive. To avoid this, a stochastic approximation [78] of gradient descent is used, called stochastic gradient descent. Instead of calculating the gradient

over the entire training sample, randomly chosen small batches of training data are used to iteratively update the network parameters. The size of the batches has to be balanced between the speed of the calculation, which is faster for smaller batches, and the quality of the gradient approximation, which is better for a lager batch size.

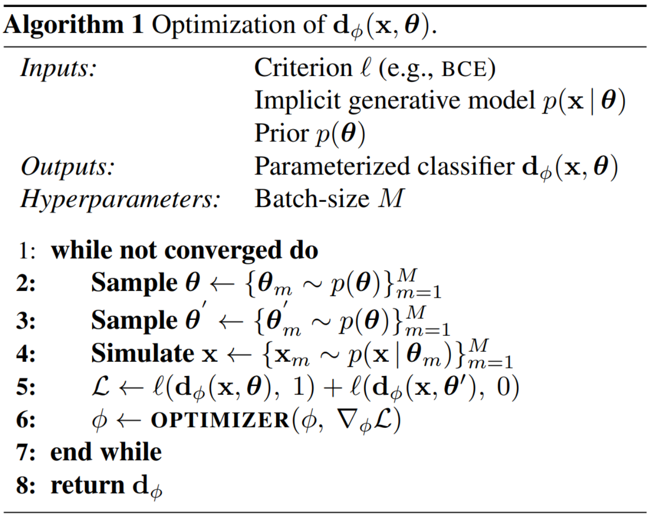
In practice it can be very difficult to construct and train a network to achieve sufficient results. Since there is no clear mathematical way to optimize its structure for a given task, the best practice is often trial and error. Also the amount and distribution of available training data are crucial numbers for the quality of the network approximation of the data.

3.4. Likelihood-free inference

One approach to combine the advantages of having a likelihood function and using machine learning techniques is likelihood-free inference with amortized approximate ratio estimators introduced in [3]. Likelihood-free in this context means that it is not necessary to construct a likelihood function in advance, rather it is learned from simulated data. No prior knowledge of the likelihood is required for this, it must only be possible to create a forward simulation (e.g. MC) to train on.

In general, learning to predict the output of a function would be a regression task. However, in the likelihood-free approach a binary classifier $d(\boldsymbol{x}, \boldsymbol{\theta})$ is trained to distinguish dependent correlated combinations of measurement $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$ from independent random combinations of $\boldsymbol{x}$ and $\boldsymbol{\theta}$. The dependent combinations originate from forward simulation and are basically drawn from $p(\boldsymbol{x}, \boldsymbol{\theta})$. They get the class label 1. The independent combinations are randomly drawn from the product of their individual probability distributions $p(\boldsymbol{x})p(\boldsymbol{\theta})$. They are assigned a class label of 0. Using the Binary Cross-Entropy (BCE, see equ. 3.6) as loss function and the sigmoid activation (see equ. 3.10) in the final layer of the classifier, it can be used to approximate the likelihood-to-evidence ratio $\frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})} := r(\boldsymbol{x}, \boldsymbol{\theta})$. The evidence, sometimes also called marginal likelihood, is the probability $p(\boldsymbol{x})$ of observing the measured data $\boldsymbol{x}$. For a specific measurement, $p(\boldsymbol{x})$ is constant and consequently $r(\boldsymbol{x}, \boldsymbol{\theta})$ is proportional to the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$.
Figure 3.3 illustrates the basic idea of the algorithm used to train $d(\boldsymbol{x}, \boldsymbol{\theta})$. Note that it proposes to newly simulate a batch of $\boldsymbol{x}$ in every iteration. This would take very long, and besides, the simulation data were already created in advance. Figure 3.4 shows a way to prepare pre existing MC simulation for the algorithm.

In the following, it is shown why the proposed training procedure leads to the fact that the classifier $d(\boldsymbol{x}, \boldsymbol{\theta})$ can be used to approximate the likelihood-to-evidence ratio $r(\boldsymbol{x}, \boldsymbol{\theta})$.
The binary cross-entropy loss H for the training is given by:



Algorithm 1 Optimization of $\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta})$.

```
Inputs:            Criterion $\ell$ (e.g., BCE)
                   Implicit generative model $p(\mathbf{x} \,|\, \boldsymbol{\theta})$
                   Prior $p(\boldsymbol{\theta})$
Outputs:           Parameterized classifier $\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta})$
Hyperparameters:   Batch-size $M$

while not converged do
    Sample $\boldsymbol{\theta} \leftarrow \{\boldsymbol{\theta}_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
    Sample $\boldsymbol{\theta}' \leftarrow \{\boldsymbol{\theta}'_m \sim p(\boldsymbol{\theta})\}_{m=1}^{M}$
    Simulate $\mathbf{x} \leftarrow \{\mathbf{x}_m \sim p(\mathbf{x} \,|\, \boldsymbol{\theta}_m)\}_{m=1}^{M}$
    $\mathcal{L} \leftarrow \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}),\ 1) + \ell(\mathbf{d}_\phi(\mathbf{x}, \boldsymbol{\theta}'),\ 0)$
    $\phi \leftarrow$ OPTIMIZER$(\phi,\ \nabla_\phi \mathcal{L})$
end while
return $\mathbf{d}_\phi$
```

Figure 3.3.: The algorithm used for the likelihood-free training process. ϕ refers to the trainable parameters of the classifier $d(\boldsymbol{x}, \boldsymbol{\theta})$, BCE stands for binary cross-entropy. Image from [3].

$$H = -p(\boldsymbol{x}, \boldsymbol{\theta}) \ln(d(\boldsymbol{x}, \boldsymbol{\theta})) - p(\boldsymbol{x}) p(\boldsymbol{\theta}) \ln(1 - d(\boldsymbol{x}, \boldsymbol{\theta})) \tag{3.6}$$

In the process of training the classifier this loss is minimized and the optimal classifier $d^*(\boldsymbol{x}, \boldsymbol{\theta})$ sits at the minimum of the loss function.

$$\left.\frac{\delta H}{\delta d}\right|_{d^*} = 0 \Rightarrow -\frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{d^*(\boldsymbol{x}, \boldsymbol{\theta})} + \frac{p(\boldsymbol{x}) p(\boldsymbol{\theta})}{1 - d^*(\boldsymbol{x}, \boldsymbol{\theta})} = 0 \tag{3.7}$$

Rearranging equation 3.7 yields:

$$\frac{d^*(\boldsymbol{x}, \boldsymbol{\theta})}{1 - d^*(\boldsymbol{x}, \boldsymbol{\theta})} = \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{x}) p(\boldsymbol{\theta})} = \frac{p(\boldsymbol{x} | \boldsymbol{\theta})}{p(\boldsymbol{x})} = r(\boldsymbol{x}, \boldsymbol{\theta}) \tag{3.8}$$

So the optimal classifier yields the likelihood-to-evidence ratio. A non optimal classifier $d(\boldsymbol{x}, \boldsymbol{\theta})$ will therefore represent the ratio estimator $\hat{r}(\boldsymbol{x}, \boldsymbol{\theta})$

$$\frac{d(\boldsymbol{x}, \boldsymbol{\theta})}{1 - d(\boldsymbol{x}, \boldsymbol{\theta})} = \hat{r}(\boldsymbol{x}, \boldsymbol{\theta}) \approx r(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{p(\boldsymbol{x} | \boldsymbol{\theta})}{p(\boldsymbol{x})} \,. \tag{3.9}$$

This ratio estimator can be extracted from a trained classifier by taking its decision function X, so the input of the final sigmoid activation function.
The sigmoid function is defined as:

$$sigmoid(X) = \frac{e^X}{e^X + 1} = d(\boldsymbol{x}, \boldsymbol{\theta}) \tag{3.10}$$

Rearranging the definition of $\hat{r}(\boldsymbol{x},\boldsymbol{\theta})$ yields:

$$d(\boldsymbol{x},\boldsymbol{\theta}) = \frac{\hat{r}(\boldsymbol{x},\boldsymbol{\theta})}{\hat{r}(\boldsymbol{x},\boldsymbol{\theta})+1} \tag{3.11}$$

From equation 3.10 and 3.11 it follows

$$X = \ln(\hat{r}(\boldsymbol{x},\boldsymbol{\theta})) \approx \ln\left(\frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})}\right) \tag{3.12}$$

So the classifier output before the final sigmoid activation approximates the logarithm of the ratio estimator. That also means that for a fixed measurement $\boldsymbol{x}$ it is the logarithm of the likelihood plus a constant term, which can be neglected for minimization tasks or likelihood ratios.
An additional advantage of extracting the logarithm of the likelihood-to-evidence ratio from the classifier is that it is numerically more stable.

As mentioned before, the simulation data usually already exist. The recommended way to prepare existing MC simulation to train a classifier for likelihood-free inference is illustrated in fig. 3.4. Instead of creating new simulation in each iteration, the combinations of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ labeled with 0 are obtained by randomly shuffling $\boldsymbol{\theta}$ (or $\boldsymbol{x}$). That is equal to randomly drawing from $p(\boldsymbol{x})$ and $p(\boldsymbol{\theta})$. Accordingly, the combination of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ obtained in this way will have the probability $p(\boldsymbol{x})p(\boldsymbol{\theta})$. The shuffling can be repeated in each iteration to show the classifier more different combinations. Another important feature of the shuffling is that it yields balanced class labels (same number of 1 and 0 labeled combinations). So the classifier will not be biased towards one of the classes.

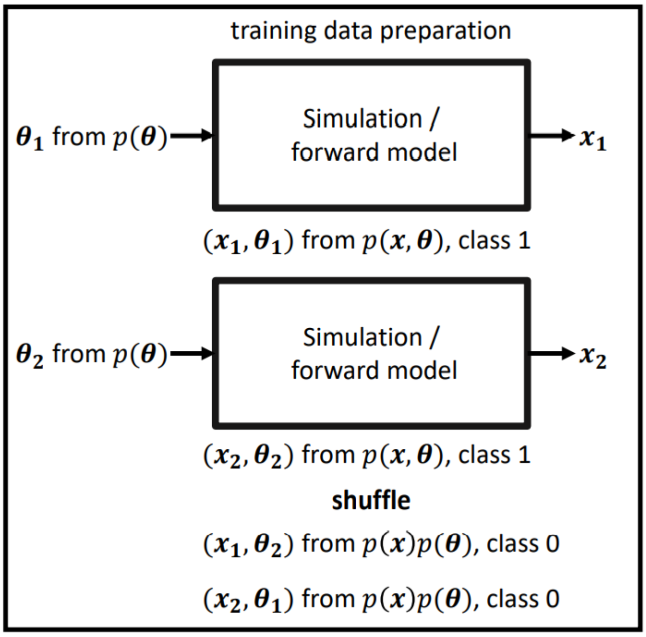

training data preparation
θ₁ from p(θ)
Simulation / forward model
x₁
(x₁, θ₁) from p(x, θ), class 1
θ₂ from p(θ)
Simulation / forward model
x₂
(x₂, θ₂) from p(x, θ), class 1
shuffle
(x₁, θ₂) from p(x)p(θ), class 0
(x₂, θ₁) from p(x)p(θ), class 0

Figure 3.4.: The recommended way to prepare existing MC simulation data for a likelihood-free inference training process. Combinations of $\boldsymbol{\theta}$ and $\boldsymbol{x}$ that come from simulation are labeled as 1, while random combinations obtained from shuffling $\boldsymbol{\theta}$ (or $\boldsymbol{x}$) get the class label 0. Plot made by Aaron Fienberg.

4. The IceCube Neutrino Observatory

The IceCube Neutrino Observatory [79] located at the Geographic South Pole is a particle detector specialized in the measurement of neutrinos. This chapter introduces the detector and explains what exactly it measures. It is also described how these measurements can be used in a reconstruction.

IceCube is among the largest detectors in the world, featuring an instrumented volume of about one cubic kilometer. It consists of 5160 photosensors frozen into the deep antarctic glazier mainly detecting Cherenkov radiation (see sec. 2.1.4) emitted from charged particles produced after neutrino interactions with the surrounding ice. IceCube is designed as a neutrino telescope with the primary goal of observing high energetic (TeV or higher) neutrinos from outside our galaxy to enable inferences about their sources. However, it is also able to observe lower energetic (GeV) atmospheric neutrinos to study their properties.

86 holes were melted into the ice sheet down to a depth of about 2450 m, which is roughly 400 m above the Antarctic bedrock. Each of them is filled with photosensors attached to a cable called string. The last 8 strings belong to the low-energy infill of IceCube, called DeepCore, discussed in 4.3.1. For the others the instrumented region reaches from a depth of 1450 m to 2450 m. With 60 sensors per string this results in a vertical sensor spacing of roughly 17 m. The horizontal spacing between these strings is about 125 m. All strings are connected to the so-called IceCube laboratory at the surface which provides the power and collects the data from the strings. After the (pre-processed) data is collected, it is sent to the "north" via satellite.

Figure 4.1 shows a schematic view of the detector including its sub-arrays.

IceTop

IceTop [80] is an extensive air shower array build at the surface above the IceCube main detector. It consists of two Cherenkov tanks per in-ice IceCube string separated by 10 m from each other. The tanks are filled with clear ice and two DOMs each.

The original idea for IceTop was to use it as veto for the in-ice detector. If there is a coincident signal in IceTop and the in-ice detector, there is a high probability that the observed event is either a atmospheric neutrino or muon and not of galactic origin. However, IceTop only covers a small area of the solid angle and no analysis actually used it as a veto [81]. IceTop is used to study cosmic rays [82].

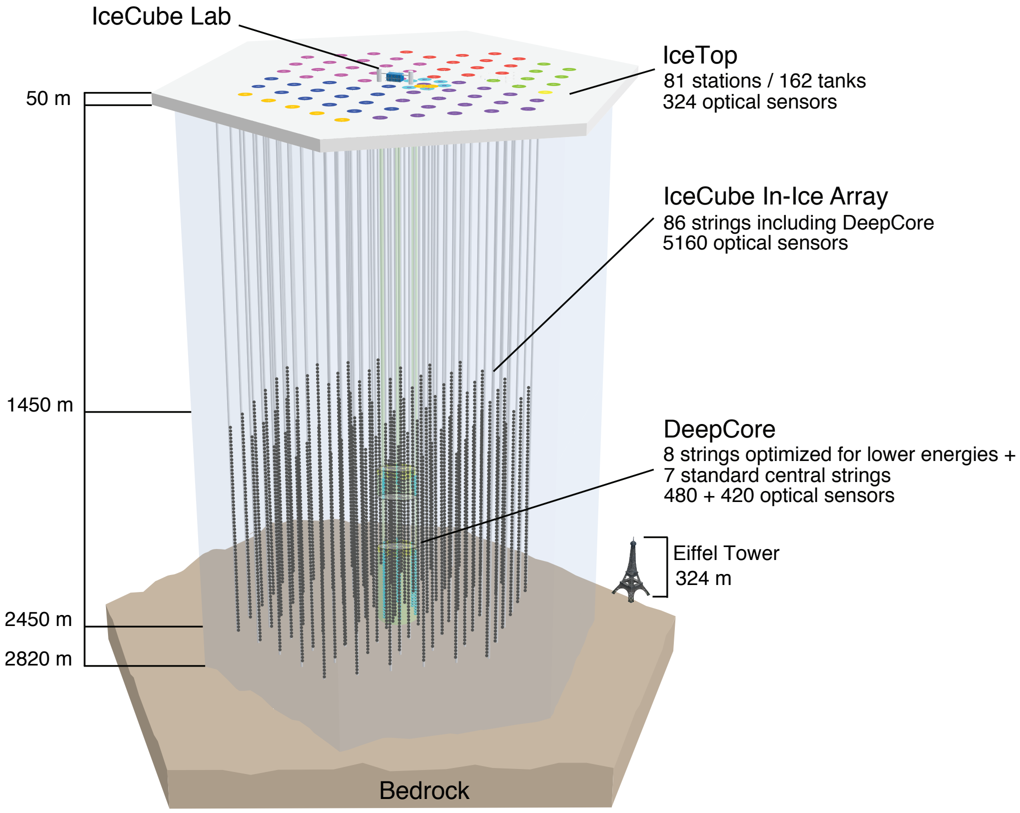


Figure 4.1.: Sketch of the IceCube neutrino telescope located at the geographic South Pole. The antarctic glazier ice is displayed in a transparent light blue color. IceCube strings and DOMs are shown as grey lines respectively black dots. On the left side the depth below the surface and on the right side the numbers of deployed modules are shown. The cosmic ray detector IceTop (colored) and the low energy sub-detector DeepCore (green) are also part of the sketch. In addition, the Eiffel Tower is used to demonstrate the size of the detector. Plot taken from [79].

4.1. The detection medium - deep Antarctic ice

The deep Antarctic glazier ice is the detection medium for IceCube and can therefore be considered as part of the detector. In the following, the optical properties of this medium are briefly discussed.

At the instrumented depths this ice is under a enormous pressure. Ice has a density of about $0.92\,\mathrm{g\,cm^{-3}}$ [83]. So the pressure increases about 0.92 bar per 10 m depth and reaches about 190 bar to 230 bar in the DeepCore volume. Impurities, for example air bubbles, in the ice are compressed to a point where they only contribute very slightly to photon absorption. Therefore this ice is extremely clear. The absorption length for 400 nm photons ranges from about 50 to more than 250 m [84]. Only at

a depth around 2050 m the absorption is larger due to a dust layer in the ice. The high absorption lengths make it possible to use a sensor spacing of tens or hundreds of meters.
However, light scattering in ice plays a more important role for IceCube. Figure 4.2 shows the absorption and scattering properties of the Antarctic glazier ice at different depths.

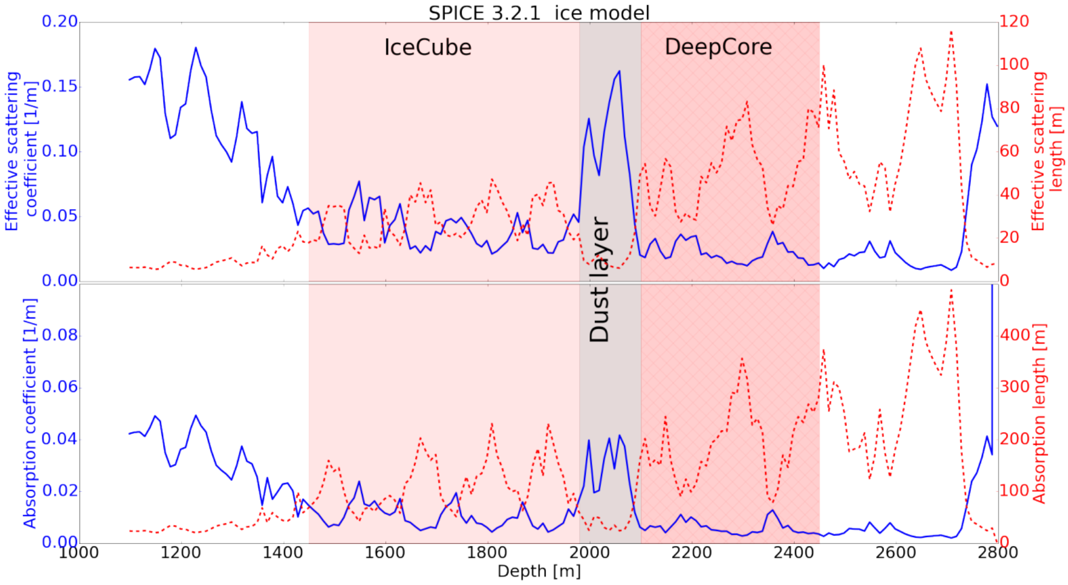


Figure 4.2.: Optical properties of the deep Antarctic ice. Plot taken from [85].

It can be see that the scattering length is only about 20 m to 80 m in the DeepCore region. So while photons fly long paths through the ice, these paths are not straight lines. This makes a reconstruction difficult because a photon arriving from a certain direction was not necessarily produced in that direction. In addition, scattering is a stochastic process and photon trajectories may look very different even if the initial photon started under the exact same conditions. An example of the result of photon scattering can be seen in fig. 5.11 where the arrival time distributions of photon hits at different DOMs are shown.

Scattering and absorption are the most prominent optical properties of a medium. However, there are more effects to be considered. The Antarctic glazier is moving with a speed of about 10 m per year at the Pole [86]. This leads to an optical anisotropy in the flow direction [85]. In addition, the bedrock of the Antarctic continent is not completely flat. While the ice is moving it fills the bedrock valleys and the ice depth layers are tilted to form the same structure as the ground. Ice anisotropy and tilt further complicate the photon propagation. They make the glazier ice a inhomogeneous medium.
More about the optical properties of the deep Antarctic glazier ice can be read in [85]

in great detail.

4.2. The Digital Optical Module

The installed photosensors are called Digital Optical Modules (DOMs) [87]. They mainly consist of a photomultiplier tube, a pressure vessel as well as readout and supply electronics. The DOM is the fundamental building block of the IceCube detector.

PhotoMultiplier Tube

The actual photon detection unit of a DOM is a PhotoMultiplier Tube (PMT). PMTs are light sensors that are sensitive enough to detect single photons.
The important components of a PMT are a photocathode and several metal plates, called dynodes, in an evacuated tube. A high voltage is applied to the plates which increases from plate to plate. Figure 4.3 shows a sketch of the different components of a PMT and how they are arranged.

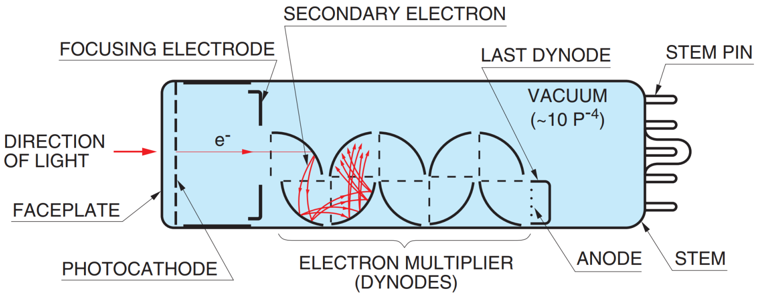


Figure 4.3.: A sketch of the essential components of a PhotoMultiplier Tube (PMT). Plot taken from [88].

The photon detection principle is based on the photoelectric effect [89]. A photon whose energy is high enough can release an electron, called PhotoElectron (PE), out of the photocathode. This electron is then accelerated towards the first dynode where it releases multiple secondary electrons. These electrons are accelerated towards the second dynode and the electron multiplication continues. At each dynode the number of electrons is multiplied and at the final dynode or anode a current is measurable. Since an electron does not always release exactly the same number of electrons from a dynode, this current varies even for the same number of initial PEs. Usually, the current is translated into a corresponding charge. The average current produced by a single initial photoelectron is assigned a charge value of 1 PE.
More details about PMTs can be found in [88].

DOM composition

The DOM PMT is a 25 cm diameter Hamamatsu R7081-02 [90] operated with the photocathode grounded. It is surrounded by a glass pressure vessel with a thickness

of 13 mm, which is able to withstand the high pressure at the instrumentation depths. PMT and pressure vessel are optically connected via an optical silicon (Room Temperature Vulcanization - RTV) gel ensuring an sufficient optical coupling of the two components. To protect the PMT from external magnetic fields (e.g. the Earth's magnetic field), a mu-metal grid is placed between PMT and pressure vessel. Several electronic components are installed to make the modules work as independently as possible. A modular high voltage power supply provides 2 kV for the PMT. Each DOM has a mainboard incorporated, which digitizes the measured signals and performs first processing steps before sending the recorded data to the surface. The mainboard contains a Field Programmable Gate Array (FPGA) and a Central Processing Unit (CPU). The installed digitizers are described in sec. 4.2.1. A Light-Emitting Diode (LED) flasher board can be used for calibration purposes or to study ice properties by emitting short light pulses that can be detected by other DOMs. After assembling, the DOMs are filled with dry nitrogen till their inner pressure reaches half an atmosphere to make them mechanically stable.
Figure 4.4 illustrates the components of a DOM and how they compose the module.

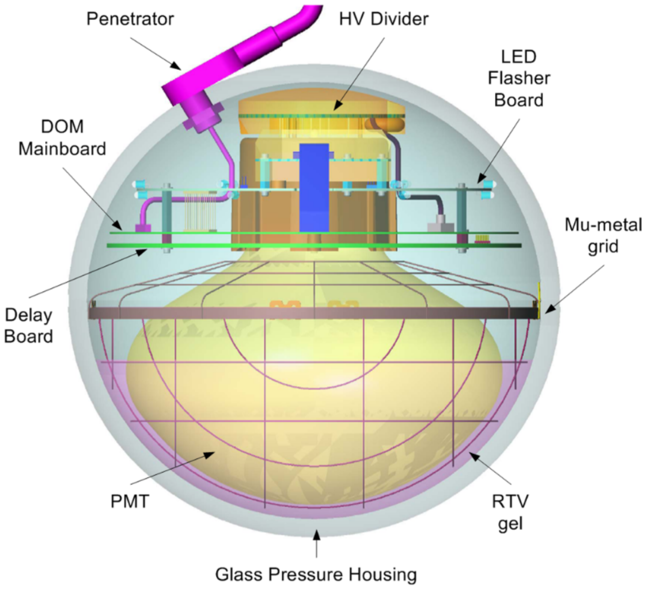


Figure 4.4.: A sketch of the essential components of a Digital Optical Module (DOM). Plot taken from [87].

4.2.1. Definition of a DOM hit

The DOMs are operated continuously. Every time the PMT sees a signal height of more than 25% of the value that an average single photon would produce (0.25 PE), the DOM gets triggered.
The analog PMT signal is digitized by two different devices:

- **ATWD:** A Analog Transient Waveform Digitizer (ATWD) [91] with a sampling rate of 300 megasamples per second. It has three channels with different amplification levels (x16, x2 and x0.25). The most amplified channel is read out first. If any sample in a channel exceeds a count of 768 the channel "saturates" and the next lower amplified channel is digitized. 128 samples are stored, which together with the sampling period of 3.3 ns leads to a coverage of an interval of about 422 ns by the ATWD. The readout process takes about 29 μs. No other waveform can be digitized during this time, so it can be considered as deadtime. Two ATWD chips are installed per DOM to reduce the deadtime.
- **FADC:** A Fast-Analog-to-Digital-Converter (FADC) with a sampling period of 25 ns (40 megasamples per second). It records 256 samples and therefore covers an interval of 6400 ns. The FADC introduces no additional deadtime to the readout.

Waveform unfolding [92] is performed on the digitized signal to determine a time and a PE charge value for each hit. The hit time is an indicator of the arrival time of the Cherenkov photons. In principle, a PMT can detect single photons. However, due to a limited timing resolution multiple photons arriving promptly one after the other can be combined in one hit with a higher charge. Figure 4.5 shows two example waveforms and the result of their unfolding. The left one is a simple waveform likely caused by only one photon. The right one seems to contain four photons. All of them could be resolved into individual hits here.

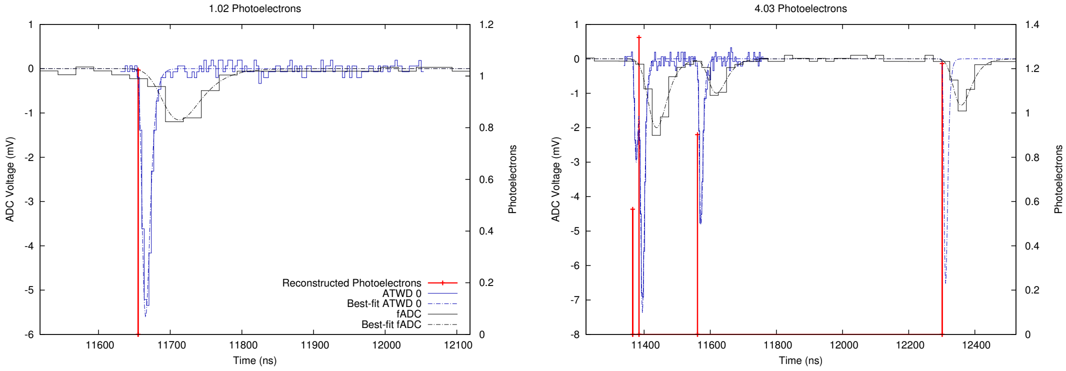


Figure 4.5.: Examples of hit time and charge determination via waveform unfolding [92]. The solid histograms show the recorded signals digitized by the ATWD (blue) and FADC (black). The dashed lines are the result of fits to the respective histogram. Red lines indicate the hit time and charge (right y-axis) obtained by waveform unfolding.

A distinction is made between two types of hits. Every time a DOM sees a hit it is checked if there is also a hit in the two neighboring DOMs above or below on the same string (so 4 DOMs are checked) within a time of ±1 μs. If this is the case, the hit is labeled as Hard Local Coincidence (HLC). HLC leads to a full readout of the

FADC and ATWD channels. If there are no other hits in the time window the hit is labeled as Soft Local Coincidence (SLC). The name is a bit misleading because there is actually no coincident hit at all. For an SLC hit the readout is reduced to only 3 out of the first 25 samples of the FADC (the highest amplitude bin and its two neighbors). No ATWD readout is done for SLC hits.

Background hits

Not all hits recorded by a DOM originate from Cherenkov photons emitted by charged particles. These additional hits do not provide any physics information and are therefore called background hits.

One source of background hits is so-called noise. For PMTs, dark noise is the dominant background source [79]. Dark noise means that a PMT signal was produced although no photon was present. It has many possible sources including for example thermal motion that can release electrons from the PMT's photocathode or radioactive decays in the glass of the PMT and pressure vessel. Fortunately, some of these effects are suppressed at low temperatures. When a DOM is cooled down, until reaching a temperature of -5 °C, its noise rate decreases, but increases again for lower temperatures. This behavior is not completely understood. The overall DOM noise rate is about 560 Hz [79].

Another type of non-signal hits observed in DOMs are so called late hits. They occur when photoelectrons are scattered back from and accelerated again to the first dynode. Around 4% of all hits are late hits, most of them arriving 26-65 ns after the primary signal [93].

A third type of background hits are afterpulses. The PMT is not perfectly evacuated. When electrons are accelerated between the dynodes they can interact with residual gas molecules and produce ions. These ions are accelerated onto the photocathode where they emit electrons which then produce delayed signals. The time delay depends on the ion type and ranges from a few hundred ns to a few μs [94].

4.3. Low energy instrumentation

The IceCube string and sensor spacing was optimized to detect and contain TeV-PeV neutrino events. However, the IceCube detection principle is also ideally suited for the detection of neutrinos with lower energies. A subset of the DOMs was arranged in a closer spacing to be able to also adequately measure atmospheric neutrinos with energies in the GeV range. This existing IceCube low energy instrumentation is presented in this section as well as a future extension of it.

4.3.1. DeepCore

The last 8 strings (79-86) deployed for IceCube have a closer horizontal spacing of between 40 m and 90 m instead of 125 m. Also their vertical DOM spacing is reduced from 17 m to 7 m. The lower 50 DOMs on these strings only cover depths between

2100 m and 2450 m, where the ice is extremely clear and the atmospheric muon background strongly reduced. The upper 10 DOMs cover depths between 1750 m and 1850 m just above the dust layer. DOMs on these 8 strings have a Higher Quantum Efficiency (HQE), so a higher probability to detect a photon. They form the DeepCore [95] sub-array whose purpose is to measure atmospheric neutrinos with energies on the GeV scale. Note that sometimes the (bottom 22 DOMs of the) 7 adjacent IceCube strings are also considered as part of DeepCore. The DeepCore energy threshold is considered to be ∼5 GeV.
Figure 4.1 shows the position of DeepCore in IceCube and fig. 4.6 its sensor spacing and instrumentation depth.

The DeepCore event extraction

The IceCube detector runs without major breaks (>99% uptime [79]). A procedure is needed to extract individual events from this continuous data stream. This includes both triggering and vetoing. Hit cleaning, which targets background hits, is performed as part of the event extraction process. Many different algorithms "filter" the IceCube data stream for interesting events. It depends on the analysis which one to use. For low energy (∼GeV) analyses there is the so-called DeepCore filter. If all conditions of this filter are fulfilled a DeepCore event is recorded.

The goal of the DeepCore filter is to keep/catch atmospheric neutrino events (initial rate ∼4 mHz) while rejecting atmospheric muons (initial rate ∼3 kHz).
As trigger condition a Simple Multiplicity Trigger (SMT) is defined. It requires three HLC hits in DeepCore DOMs within a time window of $2.5\mu s$. This trigger condition is called SMT3. All hits, HLC and SLC, occurring in this $2.5\mu s$ time window are included as part of the triggered event. It is designed to accept neutrino events, but does not explicitly reject atmospheric muons. The rate of events passing the SMT3 trigger is ∼280 Hz and thus significantly higher than the expected atmospheric neutrino rate.
So a veto algorithm is needed to reduce the number of muon events. First, a "fiducial" and a "veto" region is defined in the detector. The fiducial DOMs are all high quantum efficiency DOMs as well as the DOMs on strings immediately adjacent to the HQE strings. All other DOMs form the veto region. A hit cleaning, which targets background hits, is performed for all hits in fiducial DOMs. It removes DOMs with the first hit occurring more than one standard deviation away from the mean hit time. Then, a so-called SeededRT cleaning is performed, which iteratively looks for hits that could be causally connected. It starts with a subset of hits, typically all HLC hits, and checks for all other hits if they are within a radius of 125 m and a time of 500 ns to one of the hits in this subset. All hits that meet these criteria are added to the subset, and the remaining hits are checked again against the now larger subset. This is repeated until no hits are added to the subset anymore.
For the cleaned fiducial hits the Center of Gravity (CoG) is calculated:

$$\vec{x}_{CoG} = \frac{\sum_i^N \vec{x}_i}{N}\,, \tag{4.1}$$

where N is the number of all hits and $\vec{x}$ the vertex position.
An average hit time is calculated assuming a isotropic light emission at $\vec{x}_{CoG}$ and neglecting any scattering:

$$t_{CoG} = \frac{\sum_i^N t_i - \frac{|\vec{x}_i - \vec{x}_{CoG}|}{c/n_{ice}}}{N}\,, \tag{4.2}$$

with t being the time of a hit, c the vacuum speed of light, and n_{ice} the refractive index of ice.
For all hits in veto DOMs an effective particle speed v is calculated based on the CoG of the fiducial hits:

$$v = \frac{|\vec{x}_{CoG} - \vec{x}_{hit}|}{t_{CoG} - t_{hit}}\,. \tag{4.3}$$

Atmospheric muons traverse the whole detector approximately at c (0.3 m/ns). They often deposit (nearly) unscattered hits first in veto and then in fiducial DOMs. This leads to positive v values of about 0.3 m/ns. Neutrino events on the other hand occur inside the detector and will illuminate fiducial DOMs first (and sometimes exclusively). This leads to negative v values. If more than one veto hit gets an effective speed of $0.25\,\text{m/ns} \leq v \leq 0.4\,\text{m/ns}$, the event is rejected.
The algorithm reduces the atmospheric muon rate from 280 Hz after the SMT3 trigger to about 17 Hz while keeping 99.4% of neutrino events in DeepCore [96].

4.3.2. The IceCube Upgrade

The IceCube Upgrade [97] is a planned (and completely funded) low energy extension to the existing instrumentation. 7 additional strings will be deployed featuring approximately 700 optical sensors in total. The sensor spacing is further reduced compared to DeepCore and is about 20 m (horizontal) and 3 m (vertical). The new sensors will be located in the clearest ice region inside the DeepCore volume.
Figure 4.6 shows the Upgrade string layout compared to the existing IceCube and DeepCore strings.
As indicated in fig. 4.6, new types of optical modules have been developed for the Upgrade. The main difference to the standard IceCube DOM is that the new modules consist of more than one PMT. The two modules the Upgarde will essentially consist of are the mDOM [98] and the D-Egg [99].
The multi-PMT Digital Optical Module or short mDOM hosts 24 PMTs in a spherical pressure vessel. This design is based on the module used by the KM3Net collaboration [100, 101]. The installed mDOM PMTs have about one-third the diameter of a DOM PMT. Their different orientations nearly cover the entire solid angle and provide directional information about the incoming photons. In addition, hit coincidences in a single module can be used for triggering.

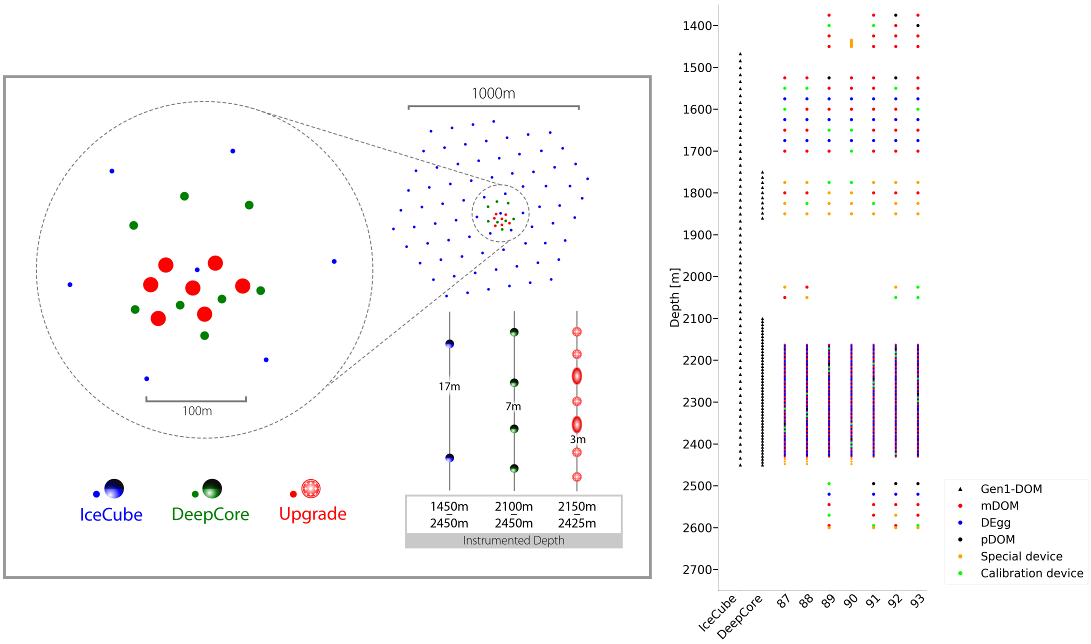
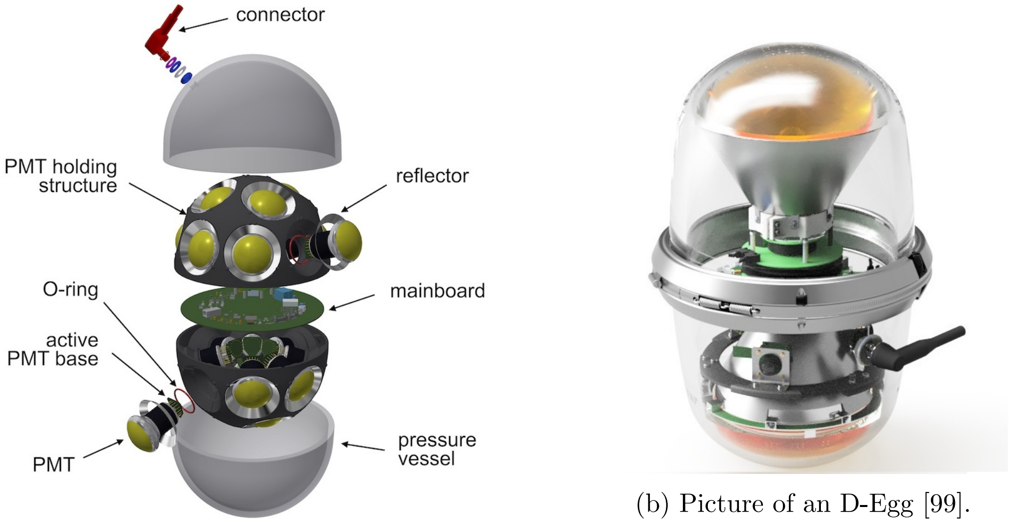


Figure 4.6.: Position and instrumentation of the Upgrade strings. The left plot shows the position of the strings in the horizontal plane relative to the existing IceCube and DeepCore strings. The instrumented depth and sensor spacing is also compared to the existing strings. More details on individual string instrumentation can be seen in the right plot. Plots taken from [97] and [98].

The D-Egg incorporates two PMTs and is oval shaped to fit into thinner bore holes. One of the PMTs looks straight down (towards Earth) and the other straight up (towards the surface). They are HQE PMTs with about 80% of the size of a DOM PMT. Its design is based on the original DOM, but the D-Egg has a higher photon collection efficiency while being cost-effective.
Figure 4.7 shows an image of both new module types.

The Upgrade will lower IceCube's energy threshold from about 5 GeV to approximately 1 GeV. This will increase the number of detected atmospheric neutrinos [97]. In addition, a improved atmospheric neutrino event selection can be expected with the Upgrade (see sec. 6.3.1). The additional information, more hits per event and the different PMT orientations, will moreover allow to better reconstruct events at low energies. All this will increase the sensitivity of IceCube to the atmospheric oscillation parameters [97]. Figure 6.16 illustrates that most of the atmospheric neutrino oscillation happens below the DeepCore threshold, the Upgrade will give better access to this region.



(b) Picture of an D-Egg [99].

(a) Rendering of an (exploded) mDOM [98].

Figure 4.7.: Images of the two optical module types that will mainly be deployed for the IceCube Upgrade.

4.4. Low energy reconstruction

Reconstructing GeV neutrino events in IceCube is challenging. Due to the large sensor spacing, after noise cleaning, there are not many hits left in an event. A median DeepCore GeV event contains 17 hits, distributed over 14 DOMs and 6 strings, most of which have a charge of about 1 PE [102]. This is only a small fraction of the modules in the detector. Moreover, the number of events to be reconstructed is usually on the order of several millions, so a fast reconstruction algorithm is preferred.
In the following it is explained what information is available in an IceCube low energy reconstruction and what is expected as its result. In addition, the reconstructions currently in use are introduced.

4.4.1. Measurement and model parameters in the IceCube low energy context

It is important to understand what the available observations $\boldsymbol{x}$ are that can be used by a reconstruction algorithm, as well as what knowledge $\boldsymbol{\theta}$ should be inferred from these observations.

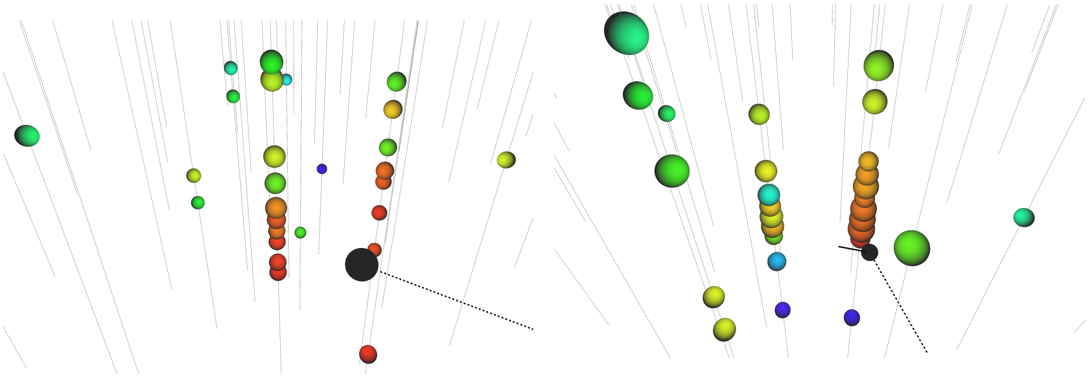
Measurement $\boldsymbol{x}$

Speaking of a measurement in the context of a single DOM means a hit with a time and a charge value. In addition, the position of the DOM can also be counted as measurement. More details about DOM hits can be found in sec. 4.2.1. For the

Upgrade modules the orientation of the hit PMT can also be added to the measurement variables.

An IceCube measurement or event consists of a collection of DOM hits. These hits come from multiple modules, but a single DOM can contribute more than one hit to this collection. Because the detector runs continuously, a trigger system is needed to determine when an interesting event happened and which hits should be part of it. This task is fulfilled by the DeepCore filter described in sec. 4.3.1.

Figure 4.8 shows two example displays of simulated DeepCore events. Both events contain a higher number of hits and hit DOMs compared the median of 17 respectively 14.



(a) Simulated event display of a 25 GeV ν_e CC neutrino interaction with the Antarctic ice inside the DeepCore sub-volume.

(b) Simulated event display of a 25 GeV ν_μ CC neutrino interaction inside DeepCore. As result of the interaction a 8 GeV μ is produced (solid black line).

Figure 4.8.: Example event displays of two simulated IceCube low energy events. The colored spheres represent hit DOMs. The size of a sphere corresponds to the summed charge of all hits and the color to the first hit time in a DOM. The black dashed line indicates the true neutrino direction, and the black dot marks the interaction vertex. IceCube strings are shown as grey lines. Plots are taken from [102].

With regard to machine learning applications it is important to note that the number of hits, and hit DOMs, as well as where they appear in the detector varies between different events. Thus, a machine learning method must be able to handle inputs of different lengths, if it should be used to reconstruct e.g. DeepCore events.

Model parameters $\boldsymbol{\theta}$

An event model has to describe the properties of the incoming neutrino as well as the interaction itself. At GeV energies the neutrino interaction with the highest cross-section is deep inelastic neutrino nucleon scattering (see fig. 2.2). To include the interaction in the model is important, because, for example, a muon neutrino with the exact same properties can produce different event signatures depending on if it

interacts via a charged or neutral current.
The following 8 parameters are used to describe a neutrino interaction with the Antarctic ice:

Position (x, y, z)
Three dimensional position of the interaction vertex in the detector.

Time t
Time of the interaction relative to the first trigger time.

Direction $(\phi^{azimuth}, \theta^{zenith})$
Azimuth and zenith angle of the direction the incoming neutrino.

Cascade energy $E^{cascade}$
Energy of all particles produced after the neutrino interaction that are neither a neutrino nor the highest energetic muon.

Track energy E^{track}
Energy of the highest energetic muon produced following the the neutrino interaction. If no muon is produced, the track energy is 0.

For most IceCube low energy analyses the total deposited energy $E^{deposited} = E^{cascade} + E^{track}$, which is a proxy for the incoming neutrino energy, and the cosine of the zenith angle, which translates to the traveled distance of an atmospheric neutrino [103], are the most important parameters. This can be illustrated by looking at equ. 2.16 which shows the two flavor oscillation probability of a neutrino. Therefore, it can be useful to reparameterize the model. Cascade energy and track energy can be replaced by the total deposited energy and the track energy fraction $(E^{track}/E^{deposited})$ or inelasticity $(E^{cascade}/E^{deposited})$. The advantage of this parameterization is that both parameters of interest are directly included, which makes it easier to calculate uncertainties for them.
The other parameters of the event model can be of interest for an analysis too. They are used for background and containment cuts as well as data-MC comparisons.

4.4.2. Particle IDentification (PID)

Particle identification is another important aspect for IceCube low energy analyses. Although in this case no specific types of particles are identified, but a distinction is made between two event signatures called tracks and cascades.
For an neutrino oscillation analysis, ideally the different neutrino flavors could be distinguished, but this is not possible with DeepCore. All neutrino interactions produce a similar sphere-like event signature called a cascade. In principle, there are two different types of cascades namely electromagnetic cascades, produced by electrons or taus after a ν_e CC respectively ν_τ CC interaction, and hadronic cascades, produced by quarks after all interaction types and for all neutrino flavors. The light yield is different for the two cascade types. In electromagnetic cascades more light

per energy is deposited in the detector than in hadronic cascades [104], but they are indistinguishable because it is always possible to scale the cascade energy to match either type. However, there is one interaction that produces a different event signature compared to all others. In a ν_μ CC interaction, a muon is produced which flies about 5 m per GeV [105] through the ice before decaying. At a few GeV muons can travel far enough to leave the cascade and produce a elongated event signature which is called a track. A second important point is that muons travel with nearly the vacuum speed of light c trough the ice, while photons only have a speed of $\sim 0.75\,c$ in ice. That means Cherenkov light emitted by muons can produce earlier hits than the Cherenkov photons from the cascade, but only in one direction.
Figure 4.9 illustrates the different interactions and what signatures they produce.

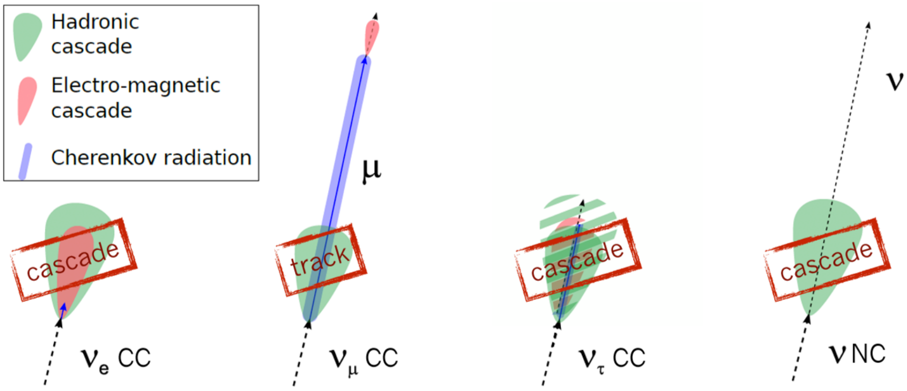


Figure 4.9.: Different neutrino interaction signatures in IceCube. All NC interactions only produce a hadronic cascade, while CC interactions also produce a electromagnetic cascade. A muon neutrino CC interaction produces a muon which can travel long distances ($\sim 5\,\frac{m}{GeV}$ [105]) through the detector. This is called a track. In IceCube particle identification (PID) means distinguishing tracks ($\nu_{\mu CC}$) from cascades (all other interactions).

The primary goal of the IceCube low energy PID is to distinguish ν_μ CC interactions from all others. Most atmospheric neutrinos are produced as ν_μ (see fig. 2.6) and for DIS the CC cross-section is higher than the NC cross-section [106]. Consequently, even though they are only produced in one interaction, more than half of the detected neutrino events are tracks (see e.g. fig 6.15).
A machine learning classifier, like for example a boosted decision tree, is suitable for PID [102]. However, according to the Neyman-Pearson lemma [107], a likelihood ratio would be the most efficient test statistic. The ratio between a fit including a free track energy and a fit with track energy fixed to 0 gives information about how important the track is to explain the measurement. Thus, a likelihood based reconstruction algorithm directly provides a PID.

4.4.3. Current reconstructions

All advanced reconstruction methods used so far in IceCube low energy analyses are based on a maximum likelihood (or minimum χ^2) approach. As mentioned in 3.2, $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ is unknown for the DeepCore detector, therefore the distribution must be approximated. A general concept in IceCube reconstruction is the use of so-called photon lookup tables [2]. These tables contain the expected photon arrival time distributions at a DOM, for a given event hypothesis, as well as the expected number of detected photons.

To build a photon table, repeated forward $\boldsymbol{\theta} \rightarrow \boldsymbol{x}$ simulation of the same light source (e.g. a cascade or a muon) is used. The table contains the light flux in spacetime around the source and is binned in 6 dimensions: 3 spatial coordinates and a time relative to the photon emission point, the photon emission angle from the light source principle axis, and the photon zenith direction angle. To avoid binning effects and be able to evaluate the table at any point in the 6 dimensional space, the table is spline interpolated and only the resulting splines are used. Because not only the photon arrival time distribution is of interest, but also in the expected number of photons, the table is split up into a part giving the arrival time PDF and a part giving the normalization. The first part can be evaluated for individual hits, while the second part is evaluated once per DOM.

Since a single table only describes one specific source, a set of tables is needed for reconstruction. As explained in sec. 4.1, the Antarctic ice is not homogeneous. That means properties like the position, especially the depth, and direction of the light source make a difference and require a new table.

While there are more reconstructions that were developed for DeepCore, this section will focus on the two used in the latest round of IceCube low energy analyses, called oscNext. The first are actually two algorithms. The Single-string Antares-inspired Analysis (SANTA, [102]) and the so-called LEERA algorithm. The second reconstruction is called RetroReco [102].

SANTA/LEERA

SANTA focuses on the reconstruction of the neutrino direction angles. It is a fast (∼0.15 s per event) reconstruction method. The algorithm is designed to only use hits produced by unscattered photons, so-called direct hits. To achieve this, a cleaning routine which aims to remove hits produced by light that has undergone significant scattering is used in advance. The expected observation time of direct hits from Cherenkov photons can be calculated geometrically using the Cherenkov angle in ice and the direction of the neutrino respectively the charged particles produced following the neutrino interaction. With this expected time and the measured time a χ^2-fit can be performed.

LEERA is a table-based energy reconstruction algorithm. It uses SANTA and another fast reconstruction output to determine the direction and vertex of the event. The

algorithm then scales the cascade energy and the track energy until the hit-no-hit probability is maximized across all DOMs. The reconstruction time per event is at the order of seconds.
The combination of SANTA and LEERA is fast, but only about 40% of the events in the final oscNext selection (see sec. 5.2.1) fulfill the criteria to be reconstructed with SANTA [102].

RetroReco

RetroReco is a table-based maximum likelihood approach. It attempts to reconstruct all 8 parameters of the neutrino interaction model (see sec. 4.4.1) at the same time. The likelihood function used in RetroReco has the following form:

$$\mathcal{L}(\boldsymbol{\theta}|\vec{q},\vec{t}) = \sum_i q_i \ln(\lambda(t_i|\boldsymbol{\theta}) + n) - \int \lambda(t|\boldsymbol{\theta})dt - N \tag{4.4}$$

where q_i and t_i are the charges respectively times of all detected hits. $\lambda(t_i|\boldsymbol{\theta}) + n$ is the time dependent, unnormalized probability of registering a hit, consisting of the contribution from the neutrino interaction $\lambda(t_i|\boldsymbol{\theta})$ and the noise rate n. $\int \lambda(t|\boldsymbol{\theta})dt$ is the total number of expected physics hits and N the total number of noise hits.
The reason for weighting the hits with their charge is that the reconstruction is based on a collection of photons. As described in 4.2, due to a limited timing resolution of the DOM, multiple photons can be combined to a single hit with a higher charge.

RetroReco uses a different approach to generate its photon tables. Instead of simulating light sources and propagate the emitted photons, the sensors themselves are treated as emitter. Hence the name **Retro**Reco. A track hypothesis is modeled as a series of colinear, constant luminosity emitters placed every ∼0.9 m along the trajectory.
RetroReco can handle scattered photons and therefore reconstruct all events. It is the state of the art IceCube low energy reconstruction and shows superior performance compared to SANTA and previous algorithms [102]. However, the RetroReco reconstruction is significantly slower (∼40 s per event).

Another disadvantage of the table-based approach taken by RetroReco is that it is difficult to transfer to the IceCube Upgrade. The tables for the new sensor types would need more memory, because the PMT orientation has to be added as dimension. Also the progress made in determining the ice properties would require larger tables that include more details about the ice. In addition, the higher number of hits will further slow down the reconstruction.

4.4.4. Potential

Finally, it is shown how potential improvements in parameter resolutions and PID would affect the sensitivity of DeepCore to different analyses. As example analyses

the determination of the atmospheric oscillation parameters and the neutrino mass ordering are used. Both analyses are describe in sec. 6.2.2.
The artificial improvement is realized by shifting the reconstructed RetroReco values of a parameter p by a fraction of the difference between reconstructed and true values:

$$p_{reco}^{new} = p_{reco}^{old} + a \cdot (p_{true} - p_{reco}^{old}) \,, \tag{4.5}$$

where a is the improvement fraction, p_{true} the true parameter value, and $p_{reco}^{old/new}$ the reconstructed parameter value before respectively after the improvement.
Figure 4.10 shows the change in sensitivity depending on a change in resolution for deposited energy, cosine zenith, or PID. For the PID, the BDT described in [102] is used. It returns values between 0, the true value for cascade-like events, and 1, the true value for track-like events. The left plot shows the 90% contour of the sensitivity to the atmospheric oscillation parameters. The right plot shows the median sensitivity to the neutrino mass ordering for different relative improvements.

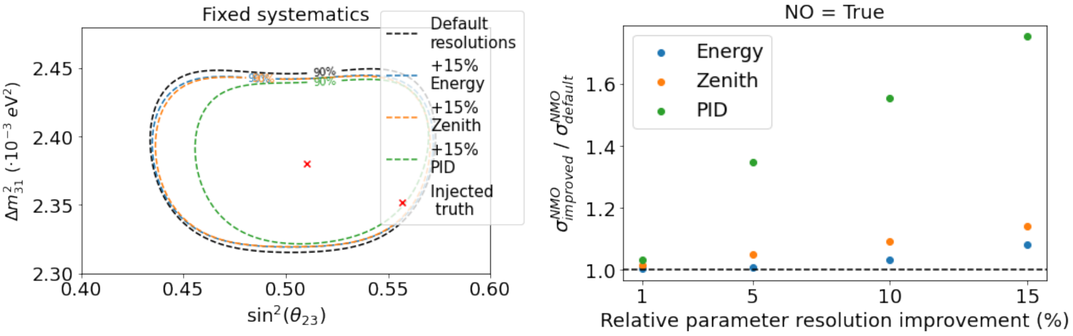


Figure 4.10.: Potential gain in sensitivity due to improved parameter resolutions and PID. Atmospheric oscillation parameters (left) and neutrino mass ordering (right) are shown. Only one resolution is changed at a time.

The PID has the strongest effect on both analyses. A 15% relative PID improvement, $a = 0.15$ in equ. 4.5, would nearly double DeepCore's NMO sensitivity. The other two parameters have a less strong effect, but would still lead to an improvement.
Note that for this study, the binning of the analyses was not changed. With improved resolutions, a finer binning could be used to better resolve oscillation signals imprinted in the detected neutrino rates per bin. This would increase the sensitivity further. However, it is not clear whether the detector provides enough information to actually achieve these improvements.

5. Implementation of likelihood-free inference - FreeDOM

Light propagation in ice is a stochastic process. Even the exact same neutrino interaction will produce different hit pattern if repeated multiple times. In addition, the inhomogeneous glazier ice prevents translation equivariance, so the exact same neutrino interaction will produce different hit patterns depending on the position of the interaction in the detector. Learning the likelihood, so the conditional probability of the model parameters $\boldsymbol{\theta}$ depending on the observation, therefore has advantages over the direct prediction of $\boldsymbol{\theta}$. The likelihood does not only encode information about the most likely $\boldsymbol{\theta}$ values, but also about the confidence in those values.
While in principle each binary classifier can be used for likelihood-free inference, deep neural networks have dramatically improved the state-of-the-art in many domains of machine learning [77]. So neural networks are used as classifier here.
DeepCore is not the ideal detector for many types of neural networks. The strings are not aligned in a symmetrical arrangement. This will also be the case for future detector extensions like the IceCube Upgrade. Thus, an approach that is flexible with respect to the position of the sensors/observations is required. This is a problem for e.g. convolutional neural networks which assume their inputs to be on a regular grid when performing the convolution. Moreover, the number of hits strongly varies between different events. That means it is important for the classifier to be flexible about the length of its input data.

In this chapter, the implementation of a likelihood-free inference based reconstruction method for particle detectors consisting of individual photosensors is presented. This includes IceCube and its low energy facilities, but is also applicable to any other detector of this type like for example KM3NeT [108] or Hyper-Kamiokande [109].
First, assumptions about the likelihood function that should be learned are discussed, followed by a description of how to learn the function. Then, the accuracy of how well the function is learned is verified. Finally, a minimization process is presented, that can minimize the likelihood.

5.1. Decomposing the likelihood

The first step is to think about what actually should be learned by the network. Although the exact function is unknown, it is possible to make reasonable assumptions about the composition of the likelihood.

IceCube, as well as its sub-detectors and extensions, consists of a large number of individual photosensors. To detect a particle interaction, observed hits in some of these sensors is required. However, all sensors will never be illuminated and for low energy events the vast majority will see no light. In addition, a single sensor can register multiple hits in the same event. As mentioned in 4.4.1, a measurement (event) $\boldsymbol{x}$ is basically a variable-length collection of hits, each of which provides a time, a charge and a position. Each sensor contributes zero or more times to this hit collection, so the total number of hits in the detector is an important component of the measurement in addition to the information provided by the individual hits.

A likelihood suited for problems where the number of observations itself is a relevant part of the measurement is the Extended Maximum Likelihood (EML) [110]. For a single sensor, the function is written as:

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \left[\prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})\right] \mathcal{N}(\boldsymbol{\theta})^N e^{-\mathcal{N}(\boldsymbol{\theta})}\,, \tag{5.1}$$

where $p(\boldsymbol{x}|\boldsymbol{\theta})$ is the Probability Density Function (PDF) for a single hit, N the observed number of hits, and $\mathcal{N}(\boldsymbol{\theta})$ the expected number of hits given the hypothesis parameters $\boldsymbol{\theta}$.

The likelihood for multiple different sensors/PMTs is composed of the individual as follows,

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{s=1}^{N_{\text{sens}}} \left[\prod_{i=1}^{N_s} p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta})\right] \mathcal{N}_s(\boldsymbol{\theta})^{N_s} e^{-\mathcal{N}_s(\boldsymbol{\theta})}\,, \tag{5.2}$$

where s is the sensor index and N_{sens} is the total number of sensors. The hit PDF $p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta})$, expected number of per sensor hits $\mathcal{N}_s$, and observed number of per sensor hits N_s is different for each sensor. Therefore, each quantity must be subscripted with an additional s. In this treatment of the different sensors, it is assumed that the statistical variations are independent for each of them. That is the case, because individual photons propagating through the ice do not affect each other and also the noise in a sensor is independent of other sensors.
Equation 5.2 involves a product over hits that depends on $p_s(\boldsymbol{x}|\boldsymbol{\theta})$ and a scaling term $P_s(N_s|\boldsymbol{\theta}) = \mathcal{N}_s(\boldsymbol{\theta})^{N_s} e^{-\mathcal{N}_s(\boldsymbol{\theta})}$ that depends on $\mathcal{N}_{\mathrm{s}}(\boldsymbol{\theta})$. So the extended likelihood can be expressed in terms of two probability distributions:

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \prod_{s=1}^{N_{\text{sens}}} \left[\prod_{i=1}^{N_s} p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta})\right] P_s(N_s|\boldsymbol{\theta})\,. \tag{5.3}$$

Consequently, knowledge of two different probability distributions is required. The likelihood formulation presented in 5.3 will be called *per-sensor* formulation in the following.

While equ. 5.2 provides a complete description of the problem, it comes with the practical disadvantage that while there may be only a small number of hits in a given event, one must determine $\mathcal{N}_s$ for every sensor, including those with no hits. This scales with the number of individual sensors that form the detector and can become computationally expensive in cases where many thousands of sensors are involved. To avoid that problem, equ. 5.2 can be rearranged defining the expected number of hits in the total detector $\mathcal{N}_{\text{tot}} = \sum_{s=1}^{N_{sens}} \mathcal{N}_s$ and the observed number of hits in the total detector $N_{\text{tot}} = \sum_{s=1}^{N_{sens}} N_s$

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \left[\prod_{i=1}^{N_{\text{tot}}} \frac{\mathcal{N}_{s_i}(\boldsymbol{\theta})}{\mathcal{N}_{\text{tot}}(\boldsymbol{\theta})} p_{s_i}(\boldsymbol{x}_i|\boldsymbol{\theta})\right] \mathcal{N}_{\text{tot}}(\boldsymbol{\theta})^{N_{\text{tot}}} e^{-\mathcal{N}_{\text{tot}}(\boldsymbol{\theta})} \,. \tag{5.4}$$

The details of this rearrangement can be found in sec. B in the appendix.
Now there is a term for each observed hit, with s_i indicating the sensor for hit i, and an overall factor relating the total number of observed hits N_{tot}, to its expectation. In this formulation, the non-hit sensors contribute solely through $\mathcal{N}_{\text{tot}}$.

Finally, introducing $p_{s_i}^{\text{tot}}(\boldsymbol{x}|\boldsymbol{\theta}) := \frac{\mathcal{N}_{s_i}(\boldsymbol{\theta})}{\mathcal{N}_{\text{tot}}(\boldsymbol{\theta})} p_{s_i}(\boldsymbol{x}|\boldsymbol{\theta})$ and a similar probability distribution for the scaling term as in the previous formulation $P_{tot}(N_{\text{tot}}|\boldsymbol{\theta}) = \mathcal{N}_{\text{tot}}(\boldsymbol{\theta})^{N_{\text{tot}}} e^{-\mathcal{N}_{\text{tot}}(\boldsymbol{\theta})}$,

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \left[\prod_{i=1}^{N_{\text{tot}}} p_{s_i}^{\text{tot}}(\boldsymbol{x}_i|\boldsymbol{\theta})\right] P_{\text{tot}}(N_{\text{tot}}|\boldsymbol{\theta}) \,. \tag{5.5}$$

This has the exact same form as equ. 5.1. In this formulation, s has taken the role of an observable quantity rather than a label. Indeed, $p_s^{\text{tot}}(\boldsymbol{x}|\boldsymbol{\theta})$ is a joint probability over $\boldsymbol{x}$ and s. If $P_{\text{tot}}(N_{\text{tot}}|\boldsymbol{\theta})$ is a Poisson distribution, then equ. 5.3 differs from equ. 5.4 only through the proportionality constant $N_{\text{tot}}!$, which is independent of $\boldsymbol{\theta}$ and has no effect on parameter inference.
This formulation, written in equation 5.5, will be refereed to as *all-sensor* formulation in the following.

The likelihood formulations described above assume that every single photon can be resolved, but as described in 4.2 the timing resolution of the DOMs is limited and photons that arrive within a short time can be combined to a hit with higher charge. Both probability distributions, in both formulations, have to be modified to include the charge information. The charge of a hit indicates approximately the number of photons that are part of the hit. So the time pdf must be multiplied by the charge of the respective hit, representing the different photons arriving at roughly the same time, and in the scaling part the charge substitutes the number of hits. So for example equ. 5.5 takes the following form including hit charges

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = \left[\prod_{i=1}^{N_{\text{tot}}} q_i \, p_{s_i}^{\text{tot}}(\boldsymbol{x}_i|\boldsymbol{\theta})\right] P_{\text{tot}}(Q_{\text{tot}}|\boldsymbol{\theta}) \,, \tag{5.6}$$

where q_i is the charge of a individual hit and Q_{tot} the complete charge observed in the total detector.
For equ. 5.6 it does not make a difference if a single hit with charge 2 or two hits with charge 1 are observed, assuming all other hit information are the same, and thus the likelihood is less depended on the ability of the readout system to resolve individual photons.

Illustration of the likelihood decomposition

To verify the validity of the likelihood decomposition discussed above and to illustrate the difference between the *per-sensor* formulation and the *all-sensor* formulation of the extended likelihood, a simple toy experiment can be used. In the toy experiment, cascades are modeled as isotropic point-like light sources and tracks as sequence of cascades. Analytic approximations of the PDFs that describe light propagation are used. More details about the toy model settings can be found in chapter A in the appendix.
The toy detector used here consists of a single string of five photosensors located in the x-y plane. They all have a y-coordinate of 0 and their x-coordinates are linearly spaced between -5 and 5, resulting in a distance of 2.5 m between the sensors. The example event is located in the x-y plane. It has a total energy of 2 GeV distributed evenly between cascade and track energy with the track pointing perpendicular to the x-y plane.
Figure 5.1 shows the LLH (see sec. 3.2) contours corresponding to the $n\sigma$ levels ($n \in 1, 2, 3, 4, 5$) of x-y likelihood scans for this toy example. The different sub terms in equ. 5.3 (first column) and equ. 5.5 (second column) formulation are shown as well as the complete likelihood. The third plot in each row shows the difference between the first two. The capital T marks the true position of the light source.
In the *per-sensor* formulation, $p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta})$ can not strongly constrain the position, because the probability for a photon to reach the respective sensor is part of $P_s(N_s|\boldsymbol{\theta})$ and $p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta})$ is only responsible for the timing of the hits. In the *all-sensor* formulation on the other hand, $P_{tot}(N_{\text{tot}}|\boldsymbol{\theta})$ does not contain information about individual sensors. Therefore, the photon survival probability is part of $p^{\text{tot}}_{s_i}(\boldsymbol{x}_i|\boldsymbol{\theta})$, which allows the position to be better constrained.
The two formulations conceptualize and decompose the likelihood function differently but ultimately yield the same result. However, the second formulation in terms of $\mathcal{N}_{\text{tot}}$ does not require an explicit sum over all sensors. Hence, the *all-sensor* formulation is considered to be faster to evaluate and less computing resource (e.g. memory) intensive. It is used for most applications in this thesis. A comparison of networks trained for both formulations is shown in sec. 6.1.2.

5.2. Learning the likelihood

The decomposed likelihood gives an idea of what should be approximated (learned) by a neural network. The next step is to implement a process that allows neural net-

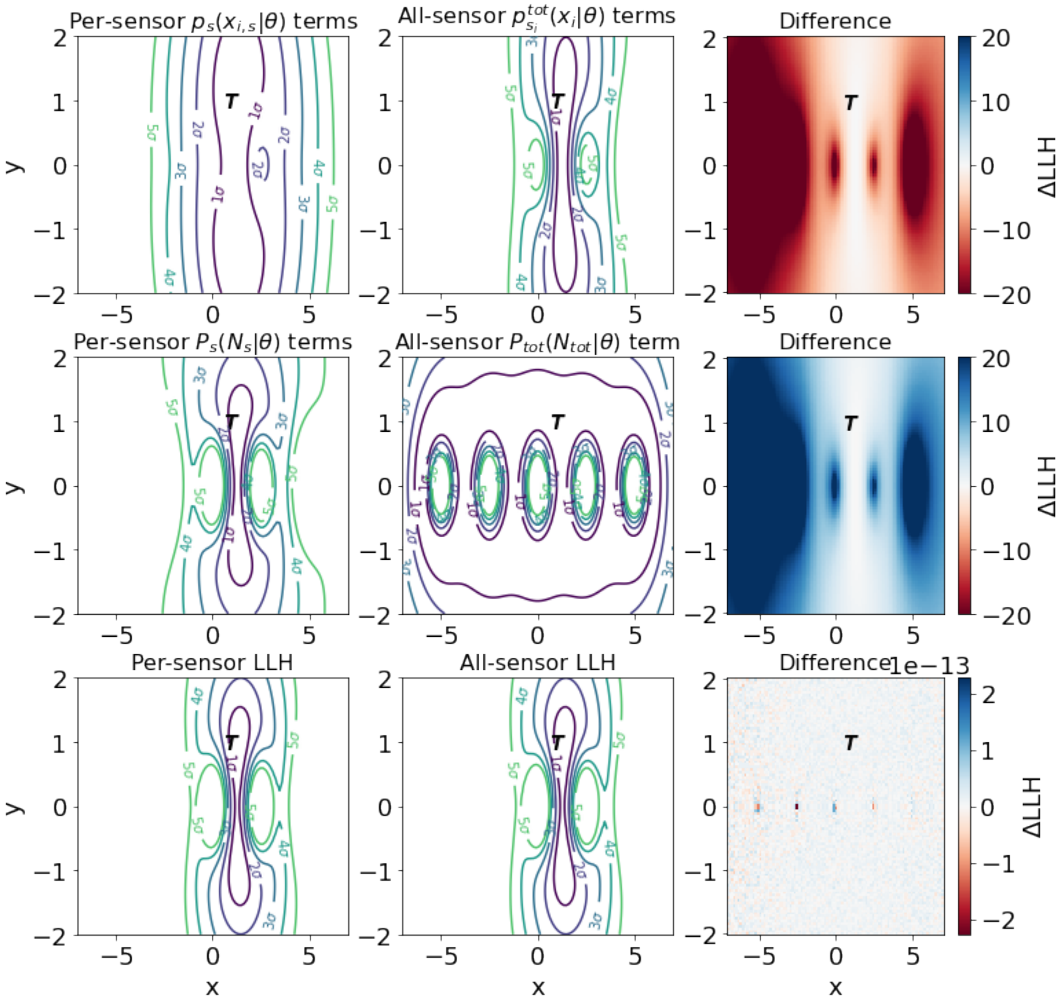


Figure 5.1.: Two formulations of the extended likelihood for an event from the illustrative toy example described in the text. Contours in the negative log-likelihood rather than the likelihood itself are shown. The first column depicts terms from equ. 5.3, and the second depicts terms from equ. 5.5. The respective terms are written in the title. In addition, their difference is shown in the third column. The black T indicates the true location of the interaction vertex.

work(s) to actually learn the likelihood function of the detector in question. Essential components of this process are the training set from which the likelihood should be learned, the structure of the neural networks and the training procedure itself.

5.2.1. Training set

This section will focus on the MC simulation that was used to train the DeepCore networks, whose reconstruction performance is presented in sec. 6.2. For all toy experiment results presented in this thesis the training set was uniformly generated in all considered parameters of the neutrino interaction model, namely $[x, y, z, t, \phi^{azimuth}, \cos(\theta^{zenith}), E, I]$. The IceCube Upgrade training event samples are described in sec. 5.2.3.
For DeepCore the current IceCube low energy MC simulation sample was used to train the networks. The sample is called *osNext* and goes through several selection levels introduced in the following.

oscNext

The oscNext MC simulation and event selection is the most advanced for IceCube low energy studies. It is used for (nearly) all analyses working with GeV energy atmospheric neutrinos. The selection is subdivided into different levels representing different cuts applied to the data. Figure 5.2 shows the expected neutrino rates for different flavors in the sample at the different selection levels together with the background rates from atmospheric muon and sensor noise. The goal of the cuts performed at each level are also indicated.

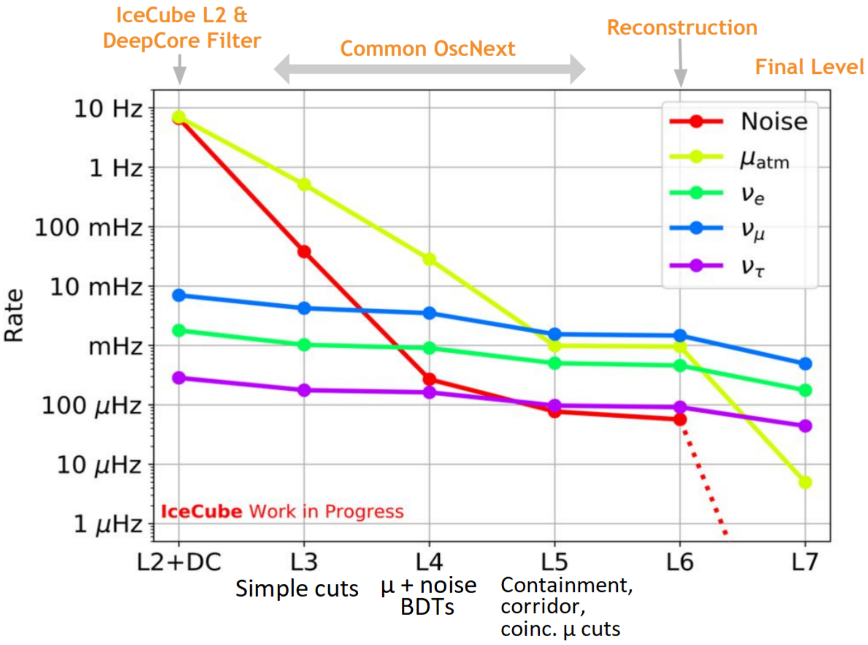


Figure 5.2.: Rates of atmospheric neutrinos, muons, and noise at the different oscNext selection levels. The plot is taken from [111].

Level 2 (L2) is a very basic selection/processing level used in all IceCube analyses. It

contains all events that meet any of the trigger conditions defined for the detector. Several low level quantities have been calculated at L2, like the calibrated waveforms of the PMT signals. L2 represents the data acquisition described in [79]. The DeepCore (DC) filter specifically selects events that show activity in the sub-detector. It is described in sec. 4.3.1.
The specific oscNext selection starts after L2 and the DeepCore filter. It is shortly described here. The reason for using different levels is that higher levels use more complicated cuts that require more computing power. Applying these cuts to events that can be removed with simpler cuts would be a waste of resources.

L3 uses simple cuts on variables that are fast to calculate, like the number of hit DOMs or the time difference between the first and last hit, to remove noise only[1] events, atmospheric muons, and coincident events. It cuts out regions of bad data-MC agreement resulting from event types that are not simulated, such as atmospheric muon bundles or coincident events. Several different cuts are used which will not be described here in detail. In summary, it can be said that muons are identified by searching for temporally and spatially coincident hits in the veto regions around the DeepCore detector (mainly above). Noise only events are identified by looking at the number of hits or the hit times relative to each other. Coincident muon events are usually longer in duration. Overall, the cuts significantly reduce the noise only and muon rates while keeping most neutrino events (see fig. 5.2).

L4 uses Boosted Decision Trees (BDTs) to further reduce the number of muon and noise only events in the sample. Two BDTs were trained (one for each type of background) to distinguish neutrino (signal) events from the respective background. The LightGBM [112] algorithm is used for both classifiers. Only cuts on the two classifier outputs are used in L4.

L5 mainly aims to reject the remaining atmospheric muons. The muons that have survived to this level do not have a clear muon-like signature (otherwise they would have been caught by the BDT). At L5 two types of cuts are used. *Starting Containment* cuts are designed to remove events that occur outside or at the edge of the DeepCore fiducial volume. *Corridor* cuts should remove muons that pass through the outer DeepCore veto region along specific corridors in the hexagonal IceCube grid without producing light in any of the IceCube strings used as veto. Starting from L5, the sample is neutrino dominated.

L6 involves the high level reconstructions described in sec. 4.4.3. The number of events in the sample is now reduced to a rate that can be fully reconstructed with a reasonable computational effort. Cuts based on reconstructed quantities are performed at this level.

---

[1] events where no particle was present

L7 is the final event selection level. More cuts are applied based on reconstructed quantities from the previous level. These cuts can depend on the analysis the sample is used for.

Only neutrino MC events, so events where a neutrino interaction was simulated, are used in the training. The reason is that the likelihood of a neutrino interaction should be learned, no muon or noise only model. Noise is also present in simulated (and of course real) neutrino events and it is possible that for some only noise hits are registered. However, even in that case information about the neutrino interaction is provided, namely that it was to faint to cause more hits.
In order to maximize the event statistics, all MC events of the oscNext simulation that pass the DeepCore filter (L2) were used to train the networks. This corresponds to about 37.6 million events (~8 million ν_e, ~20 million ν_μ, ~9.6 million ν_τ), while at the level where reconstructions are applied only about 8 million (~2 million ν_e, ~4 million ν_μ, ~2 million ν_τ) MC events are left. Higher selection level only cut out entire events but no further hit cleaning (sec. 4.3.1) is performed, which would change the average hit pattern observed from an event. Thus, it is save to reconstruct higher level events with networks trained on L2+DC events. Figure C.6 in the appendix shows that networks trained on L2+DC events can reconstruct L6 events better or equally well than networks trained on L6 events.

However, not all L2+DC events are used in the training. It has been seen that the track energy is the most difficult parameter to learn for the classifier (see e.g. sec. 6.1.1). On the one hand, the light emission from a track is more complicated to model than from a cascade. On the other hand, more than half of the events are cascades, i.e. they have a track energy of 0, but all other track energy values occur much less frequently. If half of the events in a training sample have the exact same parameter value, but the distribution of the parameter is actually continuous, this could lead to a bias towards this value. Therefore, all ν_μ and only 20% of the ν_e events are used in the training. Thus, the networks have the chance to learn each event signature but the track energy distribution is more uniform. ν_τ events are not part of the training because at GeV energies their signature is identical to one of ν_e events. This still leaves a bit less than 22 million events for the training.

Figure 5.3 shows the true simulated distributions of all 8 parameters of the neutrino interaction model at L2. The total deposited energy-inelasticity parameterization is used. The x and y vertex position are combined in a single plot to better visualize the distribution around the strings. Also the total deposited energy and inelasticity are combined in a two dimensional plot. Its uppermost bins contain the cascades and give an impression about the cascade energy distribution.

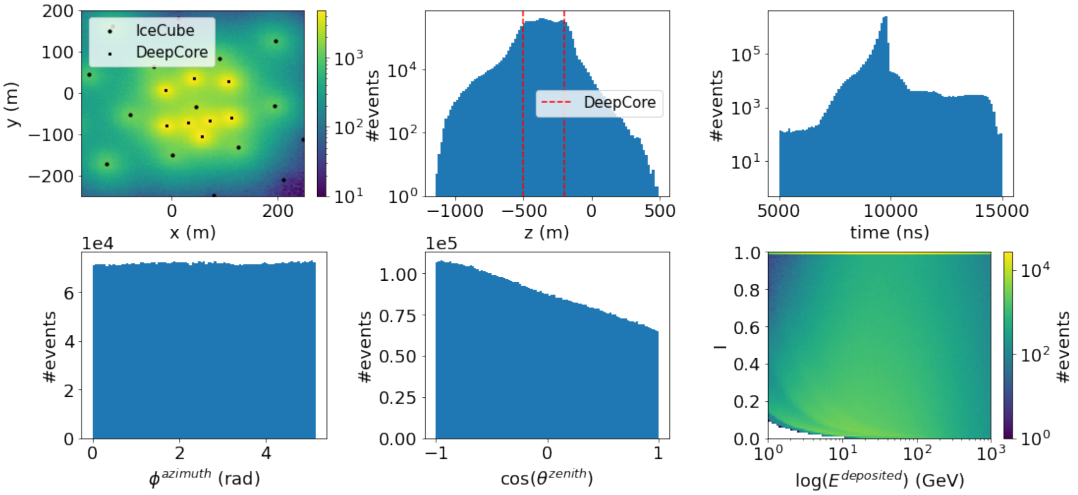


Figure 5.3.: True parameter distributions for the used training set. The x and y vertex as well as the total deposited energy and inelasticity are shown in a two dimensional histogram. This visualizes the higher probability for an event to be triggered close to a DeepCore string as well as the cascade energy distribution. The positions of IceCube and DeepCore strings are marked in the x-y plot as black dots respectively squares. The instrumented depth of DeepCore strings is indicated as red dashed lines in the z plot.

5.2.2. Neural networks and training process

This section focuses on the specific application of likelihood-free inference on a DeepCore-like detector, so a detector consisting of individual light sensors. A general introduction to the method is given in sec. 3.4.

Following the approach discussed in [3] fully connected artificial neural networks are used to approximate the likelihood discussed in sec. 5.1. These networks can not handle a variable-length input, such as a variable-length collection of hits, because they have a fixed number of input nodes. Fortunately, due to negligible photon-photon interactions, the photon propagation in ice does not dependent on other photons traveling through the ice. Consequently, hits are independent from other hits in the same event. In order to solve the variable-length input issue, the individual hit information, time and position of a single hit, for only one hit is given to a network at a time. This network is called HitNet. This is done for all hits in an event one after another and the network output is added up to build up the likelihood. This way the network always sees the same input structure no matter how many hits are part of the event. The number of hits just changes how often the network is evaluated. However, information about the total number of hits is lost in this process. Therefore, a second fully connected network, which is called ChargeNet, has to be trained which gets the number of hits in an event as input. The output of this second network has to be added to the output of the first one to obtain the complete likelihood. Figure C.3 in the appendix illustrates this construction of the likelihood. The *all-sensor* formulation of the likelihood was used for the figure, the difference between the networks in the different likelihood formulations are explained in the following.

As described in 5.1, there are two different ways decomposing the extended maximum likelihood. In both cases one gets a individual hit probability part (covered by the HitNet) and an expected number of hits part (covered by the ChargeNet). One formulation is using the expected number of hits per sensor equ. 5.2 and the other is using the total number of expected hits for the total detector equ. 5.4. The training process differs between the different versions of the networks. In general, the training process introduced in [3] requires showing the networks combinations of measurement $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$ as they come from the simulation labeled with 1 and random combinations, obtained by randomly shuffling $\boldsymbol{x}$ to different $\boldsymbol{\theta}$, labeled with 0 (see fig. 3.4). What changes for the different versions of the networks is the observation $\boldsymbol{x}$ in case of the ChargeNets and the way to shuffle $\boldsymbol{x}$ and $\boldsymbol{\theta}$ for the HitNets.

In the following, the networks are described in more detail. Both networks get the full set of neutrino interaction model parameters $\boldsymbol{\theta}$ as input, but they get different observations $\boldsymbol{x}$.

HitNet

The main purpose of the HitNet is to learn the photon arrival time distribution in a specific DOM (at a specific DOM position) depending on $\boldsymbol{\theta}$. It gets time and position of a hit as $\boldsymbol{x}$ input.

In the *per-sensor* formulation the HitNet $\boldsymbol{x}$ is shuffled in-sensor, meaning that only observations from the exact same sensor are used for the shuffling. Thereby the network learns $p_s(\boldsymbol{x}|\boldsymbol{\theta})$. For the HitNet in the *all-sensor* formulation $\boldsymbol{x}$ is shuffled between all sensors, which allows the network to learn $p_{s_i}^{\mathrm{tot}}(\boldsymbol{x}|\boldsymbol{\theta})$. Noise hits are treated in the same way as signal hits in the shuffling, because they can not be distinguished if they survived the noise cleaning.

Two additional steps have to be done for the HitNet training. Firstly, all MC events are simulated at the same absolute time. This time is then shifted depending on the event trigger time because this also defines the time scale for real data where the true time of the interaction is unknown. This explains the spread in fig. 5.3. However, even with this time spread the distribution is very narrow. To let the network also experience larger time differences for shuffled $\boldsymbol{x} - \boldsymbol{\theta}$ combinations in the training, the true event and corresponding hit times are smeared by a Gaussian. Since only relative time differences are important for the likelihood this is not a problem for true combinations but allows to see larger time differences for shuffled combinations. Without this time smearing, the network can not properly learn to exclude large time differences and instead starts to guess the likelihood. This can be a problem for events with large time differences between the hits, like for example long tracks.

Secondly, each hit has a charge expressed in Photo Electrons (PEs). The charge includes the limited time resolution of the sensors which sometimes combines multiple photons into a single hit with a higher charge. To be independent of the hit charge, a hit with charge N is split up into N identical hits for the training. However, charge values are not integer. A single photon will only on average produce a charge of 1 in a PMT. So the modulo 1 value of the charge is interpreted as probability to create another hit. For example a hit with charge 3.7 has a 70% probability to be split up into 4 hits and a 30% chance to be split up into 3 hits. Consequently, in the reconstruction the likelihood contribution of each hit is multiplied (weighted) by its charge.

ChargeNet

The task of the ChargeNet is to learn the charge distribution in a specific DOM respectively the whole detector depending on $\boldsymbol{\theta}$. Its measurement input $\boldsymbol{x}$ depends on the likelihood formulation it is trained for.

The ChargeNet in the *per-sensor* formulation gets the total charge in and the position of a single sensor as input and approximates $P_s(N_s|\boldsymbol{\theta})$ [2] as introduced in equ. 5.2. Similar to the HitNet there are multiple observations per event, one per sensor, that have to be added up to get the likelihood. The *all-sensor* formulation ChargeNet gets the total charge in the detector as input to approximate $P_{\mathrm{tot}}(N_{\mathrm{tot}}|\boldsymbol{\theta})$ as shown in equation 5.4, respectively $P_{\mathrm{tot}}(Q_{\mathrm{tot}}|\boldsymbol{\theta})$ as shown in equation 5.6. In addition, its gets the number of DOMs seeing light. The latter is not necessary but improved the network performance.

These are the essential procedures to learn the complete likelihood function $\mathcal{L}(\boldsymbol{\theta}|\vec{\boldsymbol{x}})$. In addition, to further improve the network performance a transformation layer is used as first network layer. Quantities the likelihood should depend on, like for example the distance between sensor and interaction vertex, can be calculated in that layer and passed to the network instead of directly passing $\boldsymbol{x}$ and $\boldsymbol{\theta}$. Pre-processing of the network input distributions can be performed in this layer too. The use of pre-processing, for example normalizing the input distributions, can improve the network performance [113]. Since HitNet and ChargeNet get different inputs, different transformation layers are required.

In contrast to the activation in the last layer, the activation functions in all other layers can be chosen freely. However, it is advantageous to use a smooth activation function, otherwise the learned likelihood may not be smooth.

It is also recommended to reshuffle the independent combinations of $\boldsymbol{x}$ and $\boldsymbol{\theta}$, which are labeled with 0, after each epoch of the classifier training. In this way, more combinations can be used without having unbalanced class labels in the training.

Once the networks are trained, to obtain the logarithm of the likelihood from the them, the activation function in the last layer is changed from sigmoid to linear, as described in sec. 3.4.

---

[2] Actually, in the presence of charge it rather approximates $P_s(Q_s|\boldsymbol{\theta})$

Figure 5.4 shows the structure of a network from its input to output including a transformation layer. The plot shows two red output nodes even though there is actually just one. The first one represents a node with a linear activation, the second with a sigmoid activation. The sigmoid is used in the training process, the linear activation during reconstruction.

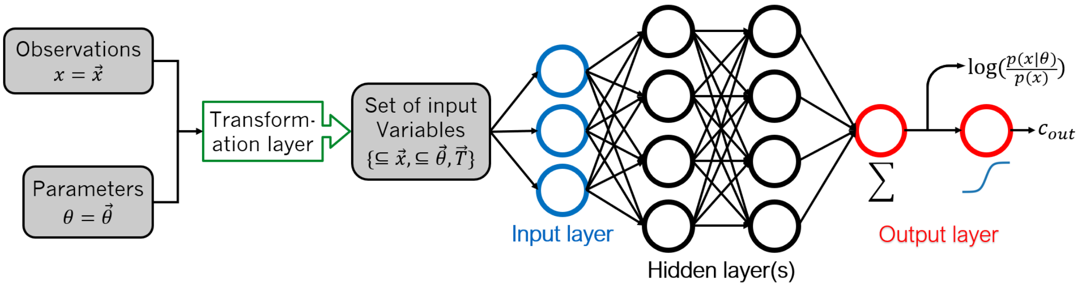


Figure 5.4.: The structure of the neural networks used for likelihood-free inference. $\vec{T}$ represents variables calculated from $\boldsymbol{x}$ and $\boldsymbol{\theta}$. c_{out} is the classifier output and lies between 0 and 1.

As indicated in fig. 5.4, the number of nodes in each hidden layer is the same. Other structures, like an inverted hour glass, were tested but did not improve over this simple structure.

5.2.3. Changes for the IceCube Upgrade

Upgrade event refers to an event that occurs after the deployment of the IceCube Upgrade, so DeepCore and the new strings can detect light. For the Upgrade two new sensor types (mDOM and D-Egg) will be installed in the Antarctic ice. Together with the already installed DOMs there are three module types involved in an Upgrade event. The big difference between the module types is that the new ones consist of multiple PMTs oriented in different directions. In addition, the PMTs have different sizes and therefore different effective areas and dark noise rates. Also the glass used for the pressure vessel is different for each of the module types changing the noise from radioactive decays. All these information could be included in the HitNet transformation layer while the ChargeNet transformation layer could pass the total charge for each module type. But this would make the task for the networks even more complicated. So to not add additional features the networks have to learn, three different Hit- and ChargeNets (one per module type) were trained for the Upgrade reconstruction. This does not slow down the reconstruction since the individual hits are anyway processed on after another. The three module types can be handled independently of each other for the same reason as the individual DOMs, photon propagation in ice is independent from other photons.
The training process does not differ for the different module types and is identical to the DeepCore training discussed in the previous section.

The upgrade MC simulation and event selection process is currently under active development. Implementing the detector response of the newly developed module types is difficult because the modules are still under development themselves and the electronic readout system is not yet finished. Another big uncertainty is the noise rate of the new modules. The noise rate of the mDOM PMTs strongly differs from its original assumption. More about this can be read in sec. 6.3.2.

So to test FreeDOMs performance on Upgrade events a noise free MC set was simulated using simplified but conservative assumptions about the module readout. This set contains about 1 million ν_μ events located in the Upgrade fiducial volume. In addition, 30 thousand test events have been simulated that are not part of the network training. Depending on their interaction muon neutrinos can produce a cascade or track event signature. So only this flavor was simulated. A plot of the true interaction model parameter distributions for this set can be found in fig. C.2 in the appendix.

The MC set described here was used for most results presented in sec. 6.3, except for the mDOM noise study discussed in sec. 6.3.2 which uses MC simulation specifically produced for this study. More details about this MC set can be found in sec. 6.3.2.

5.3. Verifying the likelihood

The next step is to verify that the neural networks trained as described in the previous section really approximate the underlying likelihood function of the (MC) data used in the training. To be able to verify the network outputs is a big advantage of the likelihood-free technique compared to direct regressors.

First, networks trained on toy data are verified. There, it is possible to directly compare the network output to the true likelihood and thus proof the functionality of the method. The general toy experiment setup used for all detector configurations in this thesis is described in sec. A in the appendix. Then, *real* detector MC simulation is used to verify that the method is also capable to approximate the likelihood function of non-toy detectors.

5.3.1. Toy detectors

The most obvious way to check if likelihood-free inference can be used to learn a likelihood function (or more accurately a function proportional to it) is to use a toy experiment where the true likelihood function can be described analytically, and thus is known. This allows for a direct comparison of the network learned function with the truth.

Two toy detector configurations are presented here. The first one is the same configuration that was used to demonstrate the validity of the extended maximum likelihood decomposition in sec. 5.1. It only consists of **one string** of five photosensors. The second toy detector consists of a **spherical** arrangement of 162 photosensors with a radius of 10 m. For the positioning of the individual sensors, the *icosphere* python package [114] was used, which generates points uniformly distributed over the sphere. The goal here to use a sensor placement, like it could be used for a liquid scintilator or water-filled detector.

One-string toy detector

This simple detector is used as a proof of concept. To also keep the problem simple, all events used here are located in the x-y plane and point upwards. Thus, only five parameters of the event model are important for this detector. Namely the x and y coordinate of the interaction vertex, the time of the interaction, the total energy (brightness) of all light sources, and the fraction of energy that goes into the first cascade, called inelasticity. A Hit- and a ChargeNet for the *all-sensor* formulation of the decomposed likelihood were trained for this example.
First, two dimensional likelihood scans are used to give an impression about the quality of the likelihood approximation. In fig. 5.5 scans of the true likelihood are compared to the function learned by the networks. The first row shows scans through x and y, while for the second row total energy and inelasticity were scanned. Each plot shows contour levels for the ΔLLH values above the minimum of the scan. So it does not matter that the networks actually learn $\frac{p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x})}$, because $p(\boldsymbol{x})$ is a constant term in the scan and thus only provides an offset in the log-likelihood. This was done for three different events. The true parameter values are marked with a capital T in each plot. For all three example events the results are in a good agreement. Remaining differences are due to the limited amount of training events (about 2 million events here) and the imperfect training process. A more detailed study about the required amount of training data follows in sec. 6.1.1.

Another way to verify and visualize the quality of the likelihood approximation is Markow-Chain-Monte-Carlo (MCMC) [115] sampling. It is a Bayesian inference technique to access the so-called posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$, which encodes information about the model parameters $\boldsymbol{\theta}$ for a given observation $\boldsymbol{x}$. The normalization of the provided input density is not important for the Markow chains, so it does not matter that the networks actually learn a function proportional to the likelihood.
Figure 5.6 shows a comparison of the posterior distribution obtained by sampling based on the true analytic likelihood and a distribution obtained by sampling based on the network learned function. The *emcee* python package [116] was used for the sampling.
Similar to the likelihood scans, the network and true result agree well. The correlations between the parameters as well as the shape of the distribution are modeled by the networks in a similar way as it is for the truth.

This can be seen as a proof of concept: In principle, it is possible to learn the decomposed likelihood described in sec. 5.1 with the likelihood-free method

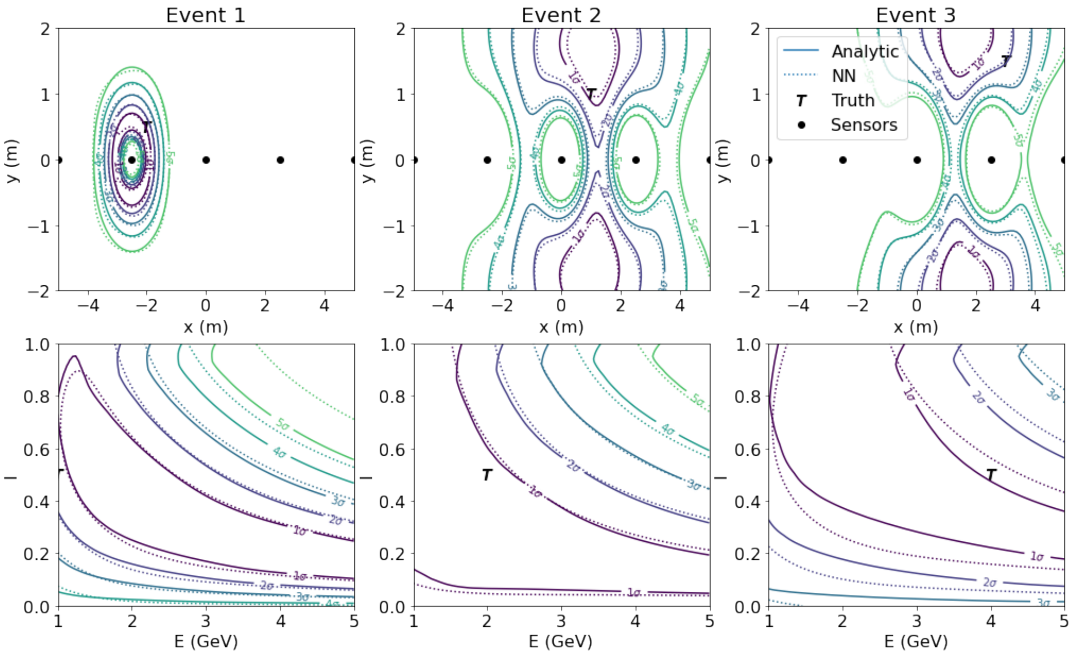


Figure 5.5.: Comparison of true (Analytic) and neural network learned (NN) likelihood for three example events in the one-string toy detector. The 1-5 sigma contours of likelihood scans are shown. For each scan all other parameter values remain at their respective truth. The capital T marks the truth of the scanned parameters.

introduced in sec. 3.4 and its realization explained in sec. 5.2.

Spherical toy detector

This toy detector features a geometry similar to water-filled neutrino detectors like the Sudbury Neutrino Observatory (SNO) [117] or liquid scintilator-filled detectors like the Jiangmen Underground Neutrino Observatory (JUNO) [118]. The training set only consists of events that are completely contained in the detector. Contained means that the interaction vertex and the last track element are inside the detector volume. Four networks were trained for this toy detector: one Hit- and one ChargeNet each for the *all-sensor* and the *per-sensor* formulation.

For this toy detector all eight parameters of the neutrino interaction model (sec. 4.4.1) are varied in the event generation process. Therefore, more two dimensional likelihood scans, compared to the one-string detector, would be needed to demonstrate the quality of the learned likelihood.

MCMC sampling would still be possible, but another way to verify the network outputs, suggested in [3], is to reweight events with the ratio estimator $\hat{r}(\boldsymbol{x}, \boldsymbol{\theta})$ learned by the networks. It approximates the likelihood-to-evidence ratio $r(\boldsymbol{x}, \boldsymbol{\theta})$

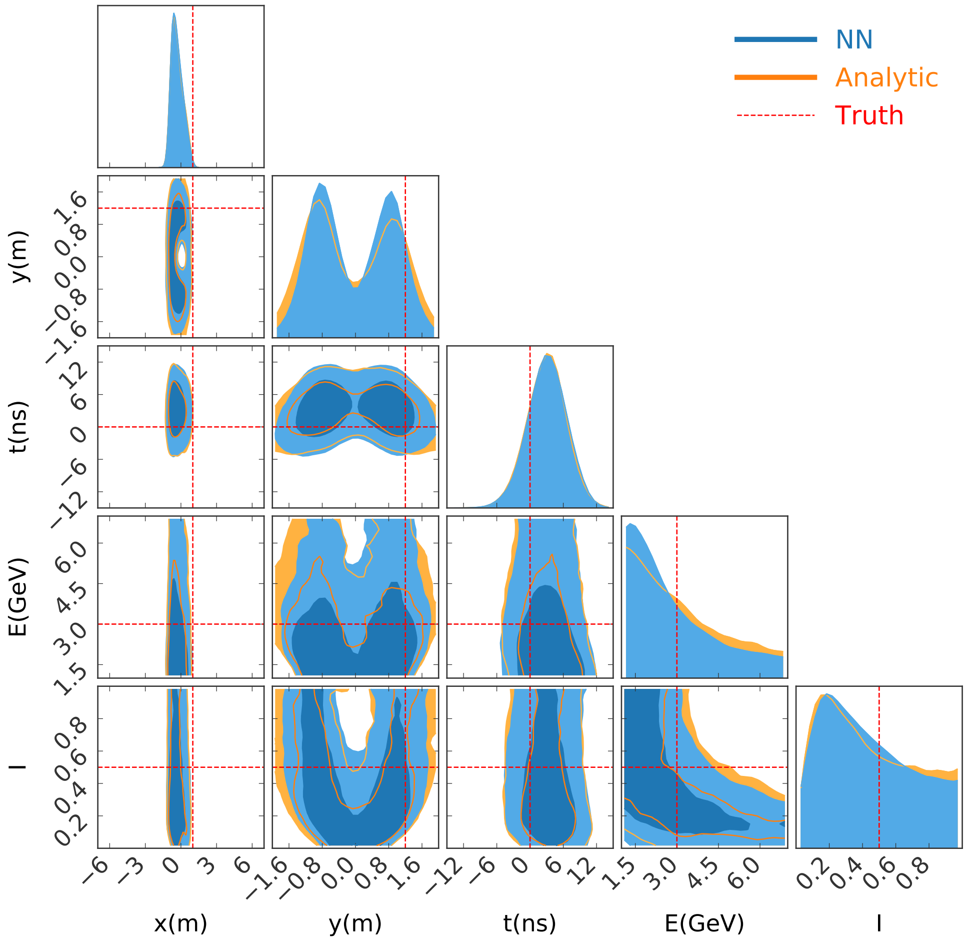


Figure 5.6.: MCMC sampling for an event in the one-string toy detector. The result based on the true likelihood (orange) is compared to the results based on the network learned function (blue). The 2D 68% and 95% contours are shown as well as the individual 1D distributions. The red lines mark the true parameter values. $2.5 \cdot 10^5$ samples are used each.

and is defined in equ. 3.9. This is achieved by reweighting the distribution $p(\boldsymbol{x})$ of measurement variables from the training set to match $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ for specific values of the model parameters $\boldsymbol{\theta}_i$

$$p(\boldsymbol{x})\hat{r}(\boldsymbol{x},\boldsymbol{\theta}_i) \approx p(\boldsymbol{x})r(\boldsymbol{x},\boldsymbol{\theta}_i) = p(\boldsymbol{x})\frac{p(\boldsymbol{x},\boldsymbol{\theta}_i)}{p(\boldsymbol{x})} = p(\boldsymbol{x},\boldsymbol{\theta}_i)\,. \tag{5.7}$$

To realize this, the networks are evaluated for each $\boldsymbol{x}$ in the training set in combination with the specific $\boldsymbol{\theta}_i$. The result yields the weight for the respective $\boldsymbol{x}$. The measurement variables are time and vertex of a hit (mainly used by the HitNets) as well as the charge of the hit (used by the ChargeNets). If the networks learned the

correct likelihood-to-evidence ratio, the reweighted $p(\boldsymbol{x})$ distribution should look like $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ so the true analytic PDF for $\boldsymbol{x}$ at $\boldsymbol{\theta}_i$.
In case $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ is unknown, the reweighted $p(\boldsymbol{x})$ distribution can also be compared to repeated MC simulation of $\boldsymbol{\theta}_i$. Thus, this reweighting method can be used to verify the likelihood even if the true function is intractable. This is done in sec. 5.3.2 for DeepCore MC simulation.

The quality of the HitNets can be visualized by looking at the time distributions of the hits. Figure 5.7 shows the times of all hits in all events of (a sample similar to) the training sample. This is $p(\boldsymbol{x})$, where $\boldsymbol{x}$ is the time of a hit. For three different sets of event parameters $\boldsymbol{\theta}_{1,2,3}$, the true analytic hit time PDF $p(\boldsymbol{x}|\boldsymbol{\theta}_{1,2,3})$ is shown as dashed lines. $p(\boldsymbol{x})$ is reweighted for each of the three $\boldsymbol{\theta}_i$ which yields the distribution in the respective color.
In the left panel an *all-sensor* HitNet was used for the reweighting and in the right panel a *per-sensor* HitNet. They describe different parts of the decomposed likelihood and consequently different PDFs. In the *all-sensor* formulation the ChargeNet does not contribute individual sensor information, so the HitNet has to learn the arrival time distribution times the probability for the photon to arrive at the respective module. In the *per-sensor* formulation the ChargeNet provides information about individual sensors and the HitNet only has to learn the time distribution. That is the reason why it is trained by only shuffling within a sensor.

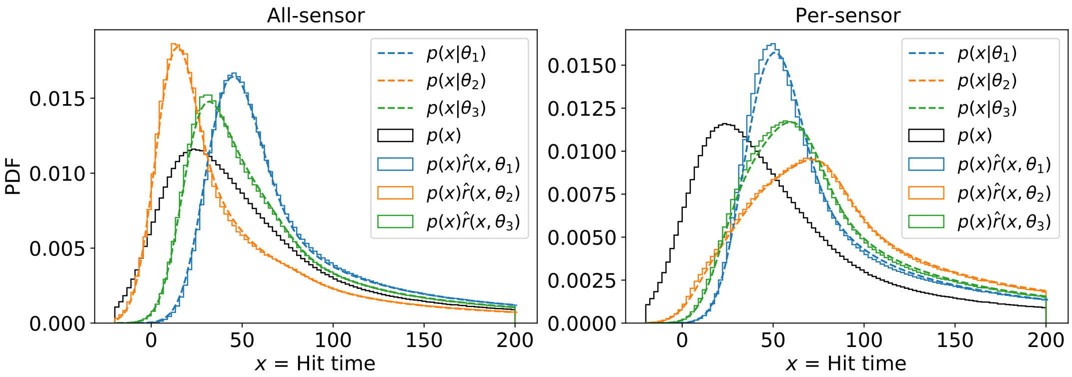


Figure 5.7.: Likelihood verification based on event reweighting. The black distributions show the times of all hits in the sample the network was trained on. The dashed lines represent the true hit time PDFs for three example events. The colored distributions are obtained by reweighting the black distributions according to the respective event parameter values. In the left plot the *all-sensor* formulation is used and in the right the *per-sensor* formulation.

All reweighted distributions look similar to the respective true PDFs. The maximum differences between true PDFs and reweighted distributions of the blue, orange, and green event are $[5.1, 4.8, 3.9] \cdot 10^{-4}$ for the *all-sensor* formulation respectively

$[6.0, 1.4, 3.2] \cdot 10^{-4}$ for the *per-sensor* formulation. So the networks have approximated the likelihood well at these points in the $\boldsymbol{\theta}$ space.

The same reweighting can be done for the charge of a hit and the ChargeNets. Figure 5.8 shows the total detector charge (left) and per-sensor charge (right) distributions. Again the distributions for the event sample similar to the training sample are shown in black. The reweigthed distributions for three example sets of event parameters are shown in blue, orange, and green. The expected charge PDFs (Poisson) are drawn as dashed lines.

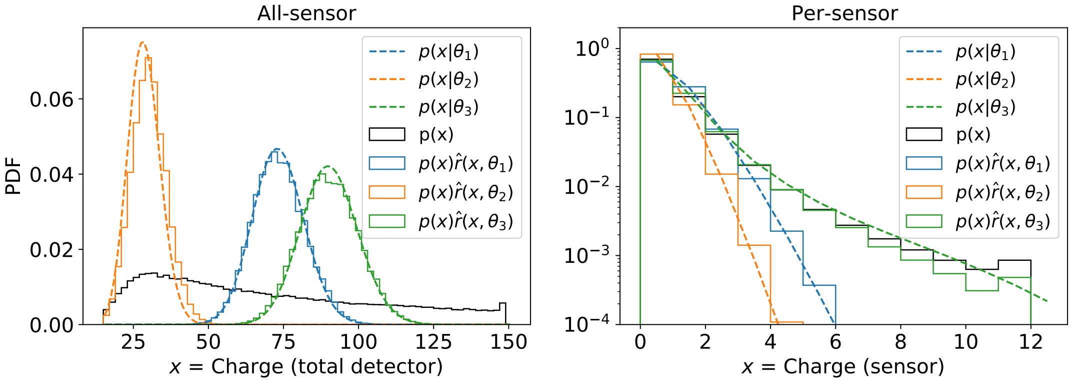


Figure 5.8.: Likelihood verification based on event reweighting. In the left plot, the black distribution shows the charge deposited in the complete detector for all events in the training sample. The right plot shows the charges for (all) individual sensors. The dashed lines represent the respective charge PDFs for three example events. The colored distributions are obtained by reweighting the black distributions according to the respective event parameter values.

Also the ChargeNets learned the likelihood well at the tested event model parameter values. Here the maximum differences between the true PDFs and the reweighted distributions of the blue, orange, and green event are $[2.9, 18.5, 4.2] \cdot 10^{-3}$ for the *all-sensor* formulation respectively $[10.6, 11.1, 2.3] \cdot 10^{-3}$ for the *per-sensor* formulation. The event shown in orange has an energy of 2 GeV, while the two other events have an energy of 5 GeV.
Figure C.1 in the appendix shows the per-sensor charge distributions for a few individual example sensors.

The reweighting method shows the quality of the likelihood approximation for three points in the event model parameter space. This method is a complementary test to a likelihood scan, because it verifies $p(\boldsymbol{x}|\boldsymbol{\theta}_i)\forall\boldsymbol{x}$, while a scan verifies $p(\boldsymbol{x}_i|\boldsymbol{\theta})\forall\boldsymbol{\theta}$. Also a reconstruction (minimization) can be used to verify the likelihood function. If the likelihood is approximated well, the resolutions obtained from a reconstruction using the learned function should match the ones using the true likelihood. A comparison between a reconstruction with the true analytic and one using the network learned likelihood is shown in sec. 6.1.2.

5.3.2. IceCube simulation

That the likelihood-free inference method works for a toy experiment is no guaranty that it will also work for the real IceCube detector. As mentioned before, the true likelihood is untraceable for this detector. Therefore, it is not possible to directly compare the network outputs to the true function.

However, one way to still get an impression of whether the networks have learned something useful is to compare likelihood scans between them and other established likelihood based reconstructions used in IceCube. In fig. 5.9 two dimensional likelihood scans of the likelihood learned by the FreeDOM networks are compared to scans with the current reconstruction RetroReco (described in sec. 4.4.3) and its predecessor PegLeg [119]. Both reconstructions use photon tables to build up their likelihood.

For each scan all other parameters are set to their true value. The blue lines mark the respective true parameter values and the white star the minimum of the scan. The color encodes the likelihood difference to the minimum of the scan.

All three likelihoods look similar, both in shape and scale of variations. This is a good sign for FreeDOM because, even though they are just approximating the true likelihood function, the other likelihoods are not expected to look completely different from the truth.

A few noticeable differences are visible in these scans. In general the FreeDOM likelihood is smoother than the others. While this is of course only a good thing if the true likelihood is also smooth, it makes the minimization easier for some optimizers (including the one used here).

In the first (x-y) scan there are likelihood maxima (in blue) visible on both sides of the minimum. They are at locations of DeepCore strings. If an event sits close to a string a high charge is expected in that string. So a lower charge event was likely not close to a string. These maxima are less prominent in the likelihood learned by the FreeDOM networks. A reason could be that these maxima represent fast changes in the likelihood which are difficult to approximate.

The fourth scan (time-z) shows an acausal region of the parameter space in the lower right corner. At these times the event would happen too late to cause most of the hits. For PegLeg and RetroReco this region is highly excluded and only a small constant noise hit probability is assigned here. However, FreeDOM shows some structure in this region which is not necessarily a bad thing. Noise hits have a different probability to survive noise cuts depending on their position in time and space relative to other hits. So there is actually no constant noise hit probability, as assumed by PegLeg and RetroReco, but a more complicated structure that could have been learned by FreeDOM, which is trained on a sample containing noise.

Thanks to the reweigthing described in [3], which was already used for the

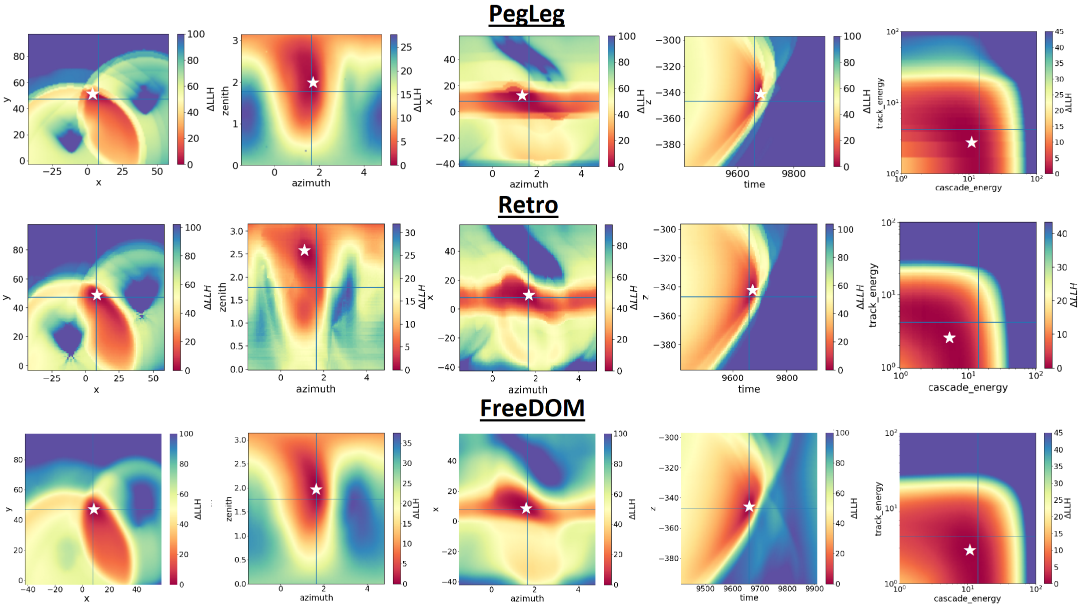


Figure 5.9.: Comparison between the function learned by the FreeDOM networks and other likelihood based reconstructions used in IceCube. Five two dimensional scans of the same event but different scanned parameters are shown. All other parameters are fixed to their true value for the scan. The blue lines mark the true parameter values and the white star the minimum of the scan. The first row shows the reconstruction used in previous DeepCore analyses (PegLeg), the second the current reconstruction (RetroReco), and the last the FreeDOM results.

spherical toy detector in fig. 5.7 and fig. 5.8, there is another way to verify the learned function. This time it is not possible to compare the reweighted $p(\boldsymbol{x})$ distributions to an analytic truth. However, it is possible to resimulate the tested event model parameters multiple times, so draw multiple measurements $\boldsymbol{x}$ based on the same model parameter values $\boldsymbol{\theta} = \boldsymbol{\theta}_i$. The obtained distribution for a measurement $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ can then be compared to the distribution $p(\boldsymbol{x})$ in the training set weighted by the ratio estimator $\hat{r}(\boldsymbol{x}|\boldsymbol{\theta}_i)$ learned by the networks. If the networks learned to correct likelihood-to-evidence ratio, the reweighted $p(\boldsymbol{x})$ distribution should approximate $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$.

Figure 5.10 shows, similar to 5.7 and 5.8, a comparison of reweighted $p(\boldsymbol{x})$ and the distributions obtained by repeated simulation for three example events. $p(\boldsymbol{x})$ is shown in black again, the reweighted distributions as dashed, and the resimulated $p(\boldsymbol{x}|\boldsymbol{\theta}_i)$ as solid histograms. As before, the colors represent different events (different $\boldsymbol{\theta}_i$) with energies of 6.37 GeV (blue), 11.71 GeV (orange), and 18.19 GeV (green).

The left plot was made for the hit time and a HitNet, the right plot for the total detector charge and a ChargeNet. The *all-sensor* formulation is used for the networks. The agreements between the reweighted and resimulated distributions show that, even

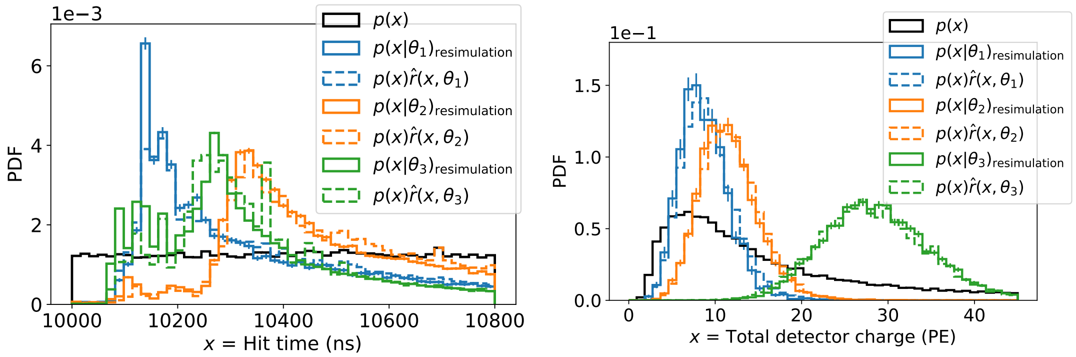


Figure 5.10.: Reweighting of hit time (left) and total detector charge (right) distribution. The black distributions are from all events in the training set. The solid colored distributions are obtained by resimulating the same event (parameters) multiple times. Reweighting the black distributions by the network output for these parameters (and the respective measurement) yields the dashed colored distributions. The *all-sensor* formulation of the decomposed likelihood is used here.

for the realistic IceCube MC, FreeDOM is able to approximate the true likelihood function, at least for these specific points in the parameters space.

Another way to verify the learned function, based on repeated MC simulation of the same $\boldsymbol{\theta}_i$, is to look at photon arrival distributions in individual DOMs. Here $\hat{r}(\boldsymbol{x}, \boldsymbol{\theta}_i)$ is not used to reweight hits but the ratio estimator is evaluated for different hit times in combination with $\boldsymbol{\theta}_i$, similar to a likelihood scan. Since evaluating $\hat{r}(\boldsymbol{x}, \boldsymbol{\theta}_i) \approx \frac{p(\boldsymbol{x}, \boldsymbol{\theta}_i)}{p(\boldsymbol{x})}$ for different hit times would change $p(\boldsymbol{x})$, the interaction time is changed instead. This does not make a difference for the likelihood, because the function depends only on the time difference between interaction and hit time, but it keeps $p(\boldsymbol{x})$ constant for the scan. Therefore, normalizing the scan yields $p(\boldsymbol{x}, \boldsymbol{\theta}_i)$ for a DOM.

Figure 5.11 shows, for a selection of six DOMs, the photon arrival time PDF predicted by FreeDOM compared to the actual hit times obtained by the event resimulation. In each histogram the hits are weighted by their charge to actually match photon counts. This is necessary because in the network training hits are split up by their charge to achieve the same effect.

Network prediction and actual hit time distribution agree well for the used event and selected DOMs. For all six DOMs the network correctly identified the causality edge, so the time where the first (not noise) photons arrive, within a few nanoseconds. Also the falling edges, which are the result of the photon scattering in ice, match the actual distributions.

It may be surprising that a closer distance to the interaction vertex does not necessarily lead to more arriving photons. The reason why that is not the case is

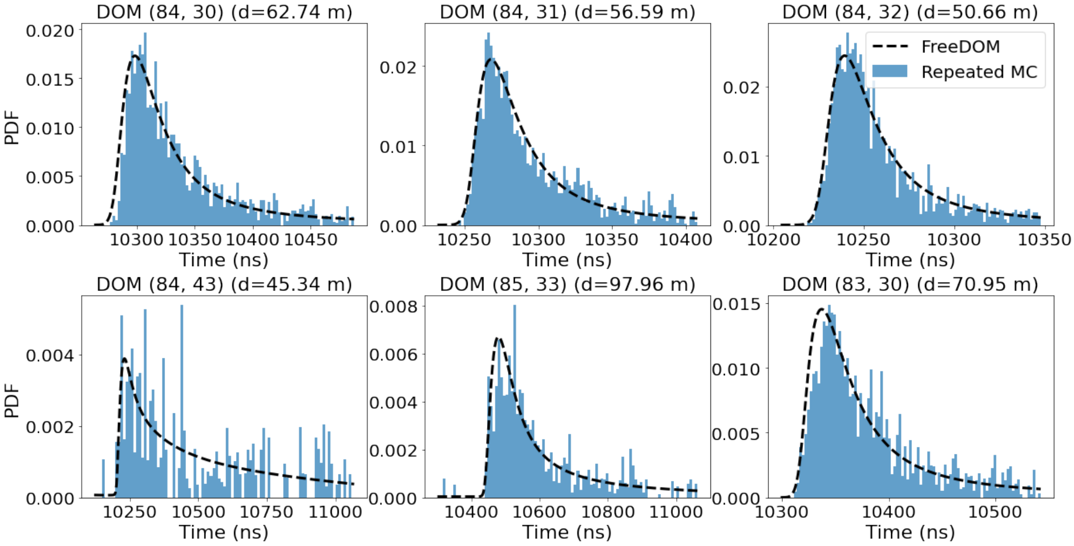


Figure 5.11.: Hit time distributions at different DOMs for the resimulation (blue) compared to the network predicted PDFs (black dashed). Each entry is weighted by the charge of the hit. The DOM position (string index, DOM index) as well as its distance (d) to the event vertex are given in the respective title of each plot.

the direction of the initial neutrino. Due to momentum conservation, most shower particles fly in similar directions to the neutrino. These particles constantly emit light while moving through the ice, so more light is emitted in regions that lie in the direction of flight of the neutrino from the interaction point. In addition, light is emitted under the Cherenkov angle which also leads to photons being directed towards the original neutrino direction.

All methods discussed above, apart from the likelihood scans, only verify the quality of the learned function at one point in the $\boldsymbol{\theta}$ parameter space. To prove the quality of the network output across more points of the parameter space (aka more events), for example a minimizer can be used. For the toy detectors it is possible to compare parameter resolutions to get a better idea of the goodness of the learned likelihood. This is done in sec. 6.1.
Also MCMC sampling includes different points in the $\boldsymbol{\theta}$ parameter space. An example of a DeepCore MCMC sampling is shown in sec. 6.2.3.

Upgrade PMT segmentation

For this part, the changes to the network training described in sec. 5.2.3 were applied and the training set described in this section was used.
In contrast to a DOM, the new optical modules deployed in the IceCube Upgrade host more than one PMT per module. The mDOM for example incorporates 24 PMTs

with different orientations. As a consequence, the optical efficiency of an mDOM is not symmetric around its azimuth (as it is for a DOM). This adds an additional dimension to the photon tables used in other likelihood reconstructions (like RetroReco) and significantly increases their size/memory usage. So far these tables just assume that the modules are equally effective around their azimuth. FreeDOM's memory usage is not affected by different PMT orientations, just the likelihood it learns.
To ensure that the FreeDOM networks correctly learned the meaning of the different PMT orientations, the following artificial test case involving an mDOM is constructed. A 10 GeV cascade at a distance of 20 m to the mDOM is used. The direction of the cascade points towards the center of the mDOM. The measurement $\boldsymbol{x}$ is a single hit in only one of the PMTs in the mDOM 110 ns after the event time. So there are 24 cases to test here, one for each PMT.
Figure 5.12 shows the HitNet part of a likelihood scan where the cascade is moved around the mDOM in a sphere keeping the distance and always pointing to the module. The angles ϕ and θ in the plots refer to the relative position of the cascade with respect to the mDOM. The difference between the 24 plots in the figure is the PMT where the hit was deposited. The black dot in each plot marks the direction of view of this PMT.
The black dot in each plot lies very close to the respective likelihood minimum. This makes sense because if only one PMT sees light the most probable position of the light source is right in front of this PMT. FreeDOM correctly modeled this behavior and consequently can make use of the additional information provided by the different PMT orientations. Parameter resolutions of reconstructions including the new modules are presented in sec. 6.3.

5.4. Minimizing the likelihood

To get a complete reconstruction method, it is not enough to construct a likelihood function, it also has to be minimized. The minimizer used to find the minimum of the FreeDOM likelihood is the same that is used for RetroReco. However, for FreeDOM all 8 parameters of the neutrino interaction model are used in the minimization while for RetroReco the energy parameters are minimized separately.
The minimization is performed by a custom version of a derivative free, global optimization algorithm. The simplex based controlled random search with local mutation algorithm crs2 described in [120], is modified to correctly treat the direction angles $(\phi_{azimuth}, \theta_{zenith})$.

Simplex based optimization algorithm refers to so-called downhill simplex methods, based on the Nelder-Mead [121] method, which can be used to minimize nonlinear functions. These methods evaluate the objective function at a number of points, called livepoints, and iteratively try to substitute the livepoint(s) with the worst function value. This is done until a convergence criterion is met, e.g. no improvement in a certain number of iterations or the standard deviation of all current livepoints

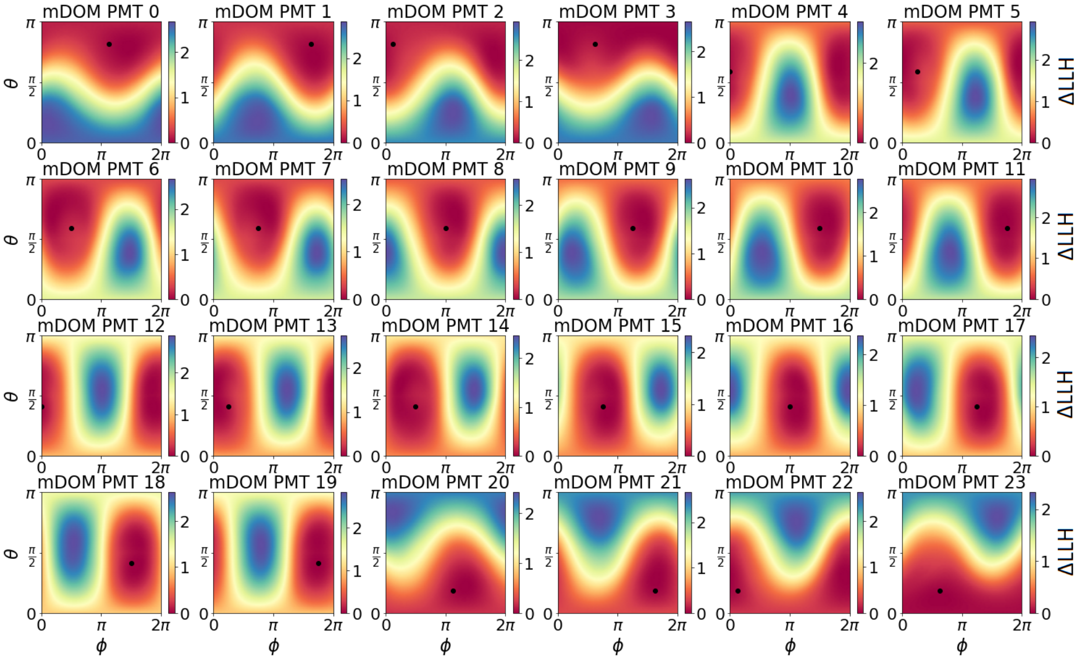


Figure 5.12.: Likelihood scans demonstrating the PMT orientations of an mDOM and that FreeDOM correctly learned them. For each of the individual plots a single hit was placed in one of the 24 PMTs. The black dot marks the direction of view for this PMT while the title gives the PMT number. Then, a light source was moved in a sphere around the mDOM. The scanned parameters ϕ and θ refer to the relative orientation between mDOM and light source.

is below a certain threshold. No gradient information are necessary and, depending on the number and initial distribution of livepoints, global optimization can be performed with simplex methods.

Crs2 is chosen because it can handle multi-modal target functions. Like most simplex based optimization algorithms, crs2 uses geometric point reflections and centroid calculations. A Euclidean geometry is assumed for these operations, which is not the case for angles. To correctly treat the azimuth and zenith angle, the calculations for these two dimensions are performed on a sphere. This changes the mentioned operations in the following way:

Centroid Calculation: The centroid of a given set of points in spherical coordinates is calculated by transforming the points to Cartesian coordinates and place them on the unit sphere. Then the element-wise average in x, y and z is calculated. The result will be somewhere inside the unit sphere, so it is renormalized to sit on the surface of the sphere again. Finally, the coordinates are transformed back to spherical coordinates to obtain the centroid in zenith and azimuth. This procedure can be done for any number of points.

Point Reflection: Performing correct geometric point reflections in spherical coordinates needs more modification than the centroid calculation. For a point reflection around the centroid (ϕ_c, θ_c), a coordinate transformation is performed that moves the centroid to the North Pole of the unit sphere. This is done in two steps. First a rotation about the z-axis ($R_z^T(\phi_c)$) such that the centroid point will be on the x-z plane, then a rotation around the y-axis ($R_y^T(\theta_c)$) until it reaches the Pole. Now a reflection about the centroid is nothing but taking the negative values of the x and y value of the respective point in its Cartesian representation. At the end the points have to be rotated back by using the inverse rotations. The full operation is shown in equation 5.8 for a point p in its Cartesian representation.

$$\begin{aligned}\overline{\boldsymbol{p}} &= R_z(\phi_c) \cdot R_y(\theta_c) \cdot \mathrm{diag}(-1,-1,1) \cdot R_y^T(\theta_c) \cdot R_z^T(\phi_c) \cdot \boldsymbol{p} \\ &= \begin{pmatrix} c_\phi(-c_\phi c_\theta^2 + c_\phi s_\theta^2) - s_\phi^2 & c_\phi s_\phi + s_\phi(-c_\phi c_\theta^2 + c_\phi s_\theta^2) & 2c_\phi c_\theta s_\theta \\ c_\phi s_\phi + c_\phi(-s_\phi c_\theta^2 + s_\phi s_\theta^2) & -c_\phi^2 + s_\phi(-s_\phi c_\theta^2 + s_\phi s_\theta^2) & 2s_\phi c_\theta s_\theta \\ 2c_\phi c_\theta s_\theta & 2s_\phi c_\theta s_\theta & c_\theta^2 - s_\theta^2 \end{pmatrix} \cdot \boldsymbol{p},\end{aligned} \tag{5.8}$$

where $c_\phi = \cos\phi_c$, $s_\phi = \sin\phi_c$, $c_\theta = \cos\theta_c$ and $s_\theta = \sin\theta_c$.
An open-source implementation of the modified algorithm is available[3].

Seeding

The crs2 algorithm needs as many seed points to start as should be used in the minimization. They are uniformly drawn from a predefined initialization range, which is different for each of the 8 parameters involved. For the vertex coordinates the range is centered around the charge weighted center of gravity of all hits in the respective event and symmetrically expands around its centers. The charge weighted center of gravity is also used to get the time range. But in contrast to the vertex the time range only expands to earlier times, because the neutrino interaction time should be significantly earlier than the average hit time. For both angles the full range of possible values is used. However, the zenith angle is sampled uniformly in its cosine instead of the angle itself, to get a uniform distribution of directions. Also the different energies are sampled over the full range allowed during the fit, but in contrast to all other parameters with a exponentially decreasing probability to make sure more points start at low energies. This should reflect the true distribution of atmospheric neutrino energies (see fig. 2.7).
Unless otherwise specified, all results presented in this thesis were achieved with 97 points present in the minimization process, of which 12 are moved simultaneously in each iteration.

---

[3] https://pypi.org/project/spherical-opt/

Boundaries

Boundaries are implemented in order to prevent the minimizer from leaving the area of the parameter space that is considered to be physically possible. An uncorrelated minimum and maximum value is specified for each parameter, resulting in an eight dimensional box the minimizer is bound to. This box is defined as:

- $-500\,m \leq x, y \leq 500\,m$
- $-1000\,m \leq z \leq 700\,m$
- $800\,ns \leq t \leq 20000\,ns$
- $0 \leq \phi^{azimuth} \leq 2\pi$
- $0 \leq \theta^{zenith} \leq \pi$
- $0.1\,GeV \leq E^{cascade} \leq 1000\,GeV$
- $0\,GeV \leq E^{track} \leq 1000\,GeV$

Using a (hard) boundary is especially important when working with neural networks, which usually extrapolate very poorly. When the networks are asked to evaluate the likelihood value of a point outside the physically possible range or more general a point far away from the parameter space covered by the respective training set, they start to extrapolate. This can result in preferred likelihood values at point that should be excluded with a high confidence.

Convergence

Three stopping criteria are defined for the minimizer. They are tested after each iteration, and if any of them is satisfied, the minimization process is terminated. For two of the criteria the fit is considered successful: if the standard deviation of the log-likelihood values of all current minimizer points is ≤ 0.1. Or if the number of consecutive iterations without finding a better log-likelihood value is ≥ 1000. If the number of iterations reaches 10000, the fit terminates unsuccessfully. Note that there is another case where the minimization is considered unsuccessful. Namely, when all seeding points are outside of the allowed region. In that case, the fit terminates after one iteration.

The minimizer successfully converges for nearly all events, but that does not necessarily mean the global minimum was found. It is very difficult to prove that the global minimum was found; in principle a full eight dimensional parameter scan would be needed, which is a substantial computational effort. There are some tests that can be performed to verify the quality of the optimizer.
A simple test is to check if the final point chosen by the minimizer yields a better likelihood value than the truth. This is possible for simulated events and the case for

approximately 97% of all fits.
It is also possible to look at parameter resolutions for different minimizer settings to verify if the used settings are sufficient. For the following plots in this section, showing parameter resolutions for different minimizer tests, oscNext L2 events were used. This explains the difference in resolution compared to fig. 6.9 which shows resolutions for a final level event selection sample.

Optimizing minimizer configuration

The first test that can be done for simplex based minimizer is to increase the number of points used by the minimizer. Thereby it will explore more of the likelihood space at the same time. This of course will make the centroid less "mobile" meaning that the change of one point has less of an effect on it. To compensate for that effect the number of points moved in each iteration has to be increased by the same factor as the total number of minimizer points. While an increased number of minimizer points will improve the resolution, it also increases the minimization time respectively the computational effort by about the factor of additional points. For fig. 5.13 the number of points was increased by a factor of 5 compared to the default.

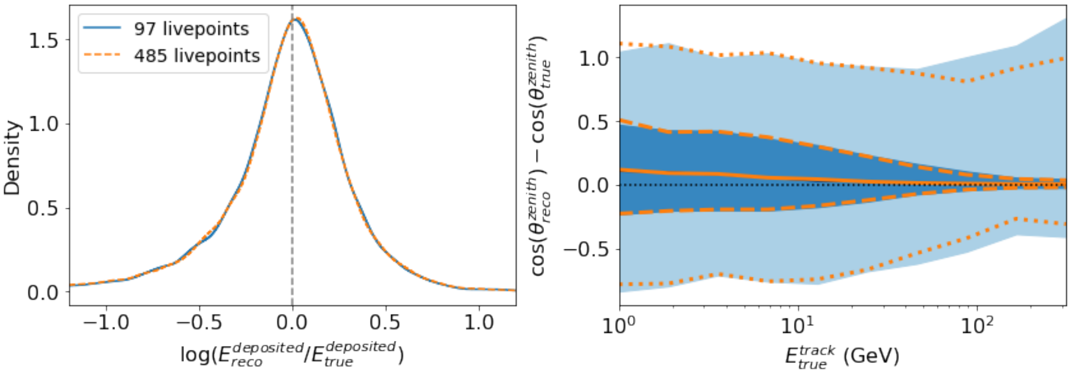


Figure 5.13.: Change in energy and zenith resolution achieved by increasing the number of crs2 points by a factor of 5 (orange) compared to default the value (97, blue). The drawn distributions are the result of a Kernel Density Estimation (KDE) [122, 123].

Figure 5.13 shows that the total deposited energy resolution is virtually unaffected by the increase of minimizer points. The zenith resolution for low track energies is also very similar using significantly more points. However, for higher track energies there is a slight increase in angular resolution. This is the reason for showing the zenith angle versus the true track energy and can be explained by the fact that, for long tracks, the likelihood minima become very narrow for the angles. These narrow minima might be missed by a minimizer using a lower point density which consequently samples the likelihood space more coarsely. Since there is only a slight increase in performance, in a region that is less interesting for most low energy analyses, but a increase in reconstruction time of about a factor of 5, 97 minimizer points are assumed to be sufficient.

The global crs2 minimizer can be followed by a gradient based local minimizer like *minuit* [124] to better locate the exact position of the minimum. Assuming the global minimization at least roughly identified the global minimum but did not perfectly converge on it. In fig. 5.14 the performance of crs2 only is compared to crs2 followed by minuit. The local minimizer is seeded with the best fit point of the global one.

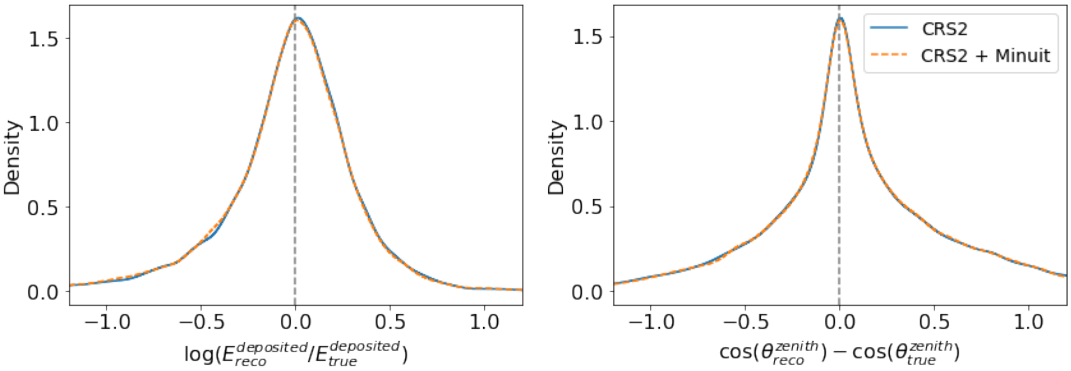


Figure 5.14.: Energy and zenith resolution for a global fit with crs2 only (blue) compared to a global followed by a local fit with cr2 and minuit (orange). The drawn distributions are the result of a KDE.

No significant improvement was achieved in either of the parameters but the fitting procedure now takes about twice as long. So crs2 identifies the position of the minimum with a sufficient precision and no second round of minimization is needed.

A nice feature of the likelihood free inference technique used in this thesis is that training on a set of events and reconstructing the very same events involves different evaluations of the networks. It is very unlikely that the exact true point in the parameter space is hit during the minimization, so the networks see different combinations of $\boldsymbol{x}$ and $\boldsymbol{\theta}$ compared to the training and can not overfit the event sample as fast as it would be possible in a direct regression. As a result the resolutions of events that were part of the training process and those that were not should be the same and it would not necessary to produce a specific training set that is independent from the set that needs to be reconstructed.
Figure 5.15 compares the energy and zenith resolution of events that were part of the network training process with events that were not.
No significant difference in energy and zenith resolution can be seen between the two event samples. This means there is no overfitting present in the likelihood learning process.

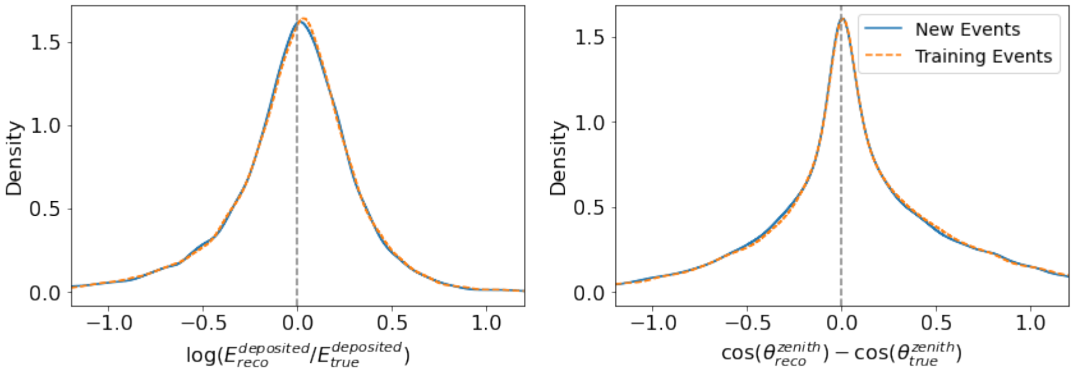


Figure 5.15.: Energy and zenith resolution for events that were part of the network training (orange) and new events that were never seen by the networks (blue). The drawn distributions are the result of a KDE.

Parameter transformations

As mentioned in sec. 4.4.1 it can be advantageous to use another representation of the neutrino interaction model parameters. Instead of passing the cascade $E^{cascade}$ and track energy E^{track} to the minimizer, the total deposited energy $E^{deposited}$ and the track or cascade energy fraction (the latter is called inelasticity $I = \frac{E^{cascade}}{E^{deposited}}$) can be used. This does not change the underlying likelihood function but the way the minimizer has access to it. In chapter 7 the minimizer points visited during the minimization process are used to calculate parameter uncertainties. Since the total energy is important for most analyzes, the uncertainty on it is needed. In the parameterization including $E^{deposited}$ this uncertainty can directly be extracted from the minimizer points.

Also networks can in principle be trained in each parameterization. To be able to use different parameterizations for networks and minimizer, a transformation feature was added to the minimizer. However, networks trained with either parameterization showed similar performances.

6. Applications of FreeDOM

As seen in the previous chapter, FreeDOM can be used for any IceCube-like detector, i.e. any array of photosensors in any configuration, without major changes. This chapter discusses the results of the application of FreeDOM to different detectors.
First, different toy experiment configurations are used to compare the parameter resolutions achieved with the learned function to the results of a reconstruction based on the true likelihood (sec. 6.1). Then, the performance for DeepCore (MC) events is evaluated and compared to RetroReco. Finally, results for IceCube Upgrade events are discussed.

6.1. Toy experiment studies

In this section the reconstruction performance of FreeDOM on the different toy experiment configurations is shown. The toy detectors are the same that were used in sec. 5.3.1 to verify the validity of the likelihood-free approach. In addition, a FreeDOM-based detector optimization study is presented.

6.1.1. One-string toy experiment

The first configuration is the simple one-string toy detector, that was used to verify the likelihood decomposition. Ten thousand events were reconstructed. Just as in sec. 5.3.1, they are restricted to be located in the x-y plane and pointing upwards. The events were simulated uniformly in x, y, time, total energy, and inelasticity, while the z coordinate and the angles are fixed. The *all-sensor* formulation of the decomposed likelihood was used for the networks.
Figure 6.1 shows the parameter resolutions for a reconstruction using the true analytic likelihood function together with the results of a reconstruction using the function learned by the FreeDOM networks. For each distribution the 50 percent Inter Quantile Range (IQR), the difference between the 0.25 and 0.75 quantile, is included.
The reconstruction performance achieved with the FreeDOM learned likelihood is similar to the one using the true analytic likelihood. There are nearly no visible biases and the parameter resolutions are comparable. However, there are small differences for example in the total energy resolution. These differences can (among other things) be attributed to the limited amount of available training data. The networks used in fig. 6.1 were trained with about 2.5 million events.

This raises the question how many events are needed to *perfectly* learn the likelihood, so how the network quality depends on the number of training events.

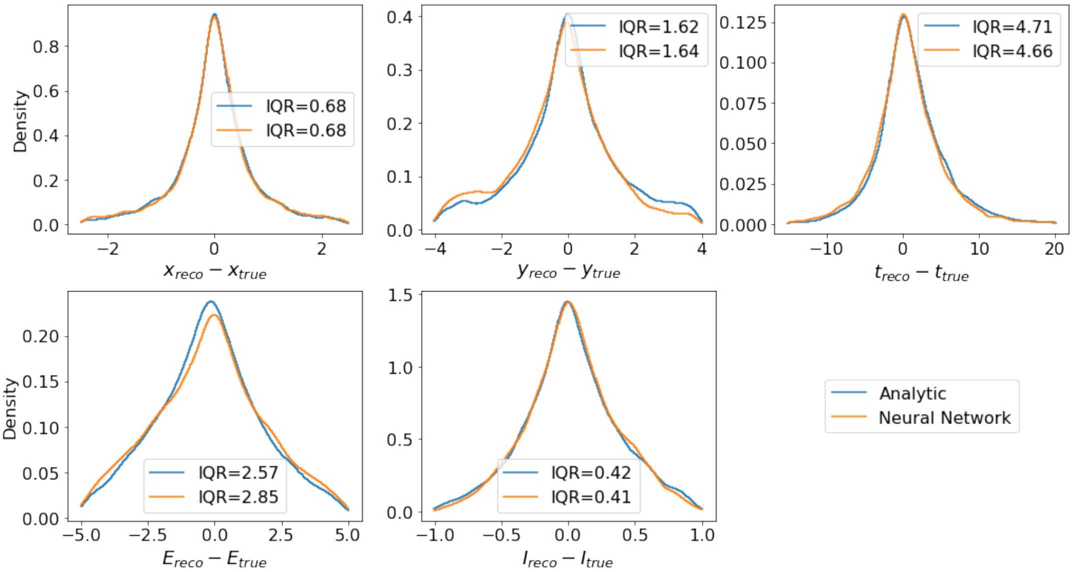


Figure 6.1.: Parameter resolutions for the one-string toy experiment. The results of a reconstruction with the true analytic likelihood (blue) are compared to those with the function learned by the networks (orange). For each distribution the IQR is given in the legend. The drawn distributions are the result of a KDE.

To address this question, first the network quality has to be defined. Ideally, there would be no difference between the true analytic likelihood and the one learned by the networks. As a consequence, also the distributions showing the difference of the reconstructed and true parameter values should be identical. Comparing these distributions includes many events and is much faster to do than comparing likelihood scans. One way to quantify how similar two (one-dimensional probability) distributions are is the Kolmogorow-Smirnow (KS) test [125], which looks at the cumulative distributions. This test has to be done for each of the five parameters individually. The average of the five resulting KS values can then be used as a measure of how well the networks have approximated the true likelihood.
Now that the network quality is defined, it is possible to study the likelihood approximation depending on the number of events in the training sample. First, a Hit- and a ChargeNet is trained with a certain number of training events. Afterwards, always the same set of test events is reconstructed with these networks. KS values are calculated for the resulting parameter resolution distributions. Finally, the average of these KS values can be determined. The reconstruction itself (minimizer, boundaries, seeding, ...) is not changed here, only the used likelihood function.

Figure 6.2 shows the result of the procedure described above. In the left plot the individual parameter KS values as well as their averages are shown. The used number of events to train the networks is given by the lower x-axis, while the upper x-axis displays the memory usage of the respective training set. In addition, the black dashed line gives the average KS value of a comparison of the analytic results with a second round of reconstruction using the true likelihood. Since the crs2 minimizer is seeded randomly, even two fits with the same settings might not yield the same result. Therefore, it is not possible to reach perfect agreement (KS value of 0), but the dashed line marks the limit where differences in the likelihood do not matter anymore compared to the minimizer.

The right plot shows the average KS value at each point for three different network sizes. Both the network depth and width have been changed. The smaller networks have about 60% of the number of free parameters compared to the "normal" ones, while the big networks have about 1.66 times the number of free parameters.

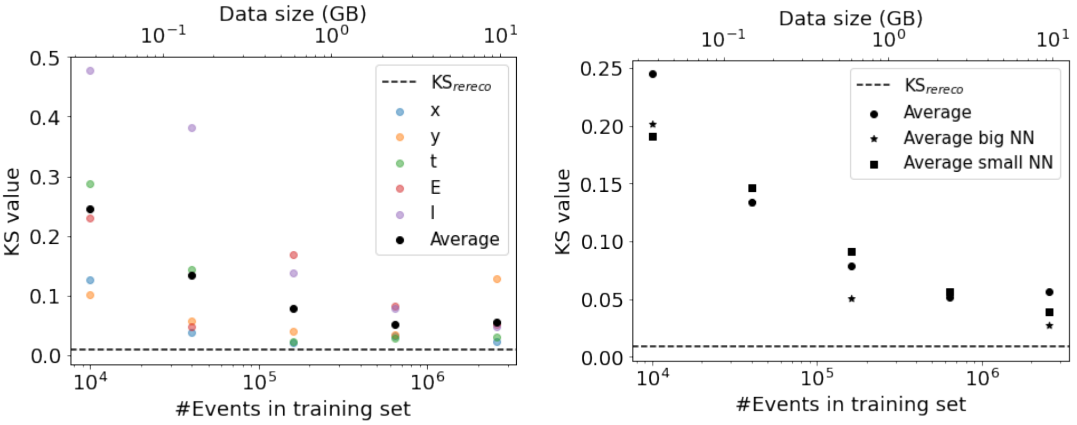


Figure 6.2.: Quality of the likelihood approximation depending on number of events in the training sample. KS values between the resolution distributions obtained with the true and the network learned likelihood are shown. In the left plot, the individual KS values for all five parameters (colored) are plotted together with their average (black). The right plot shows average KS values for three different networks sizes. The dashed black line is the average KS value of two reconstructions with the true likelihood. The upper x-axis gives the memory usage of the training sample.

There is a strong fluctuation of the individual KS values, but their average decreases with more training events before it saturates after about half a million events. Also the spread of the individual KS values becomes smaller. The inelasticity I appears to be the most difficult parameter to learn with low numbers of training events. It has higher KS values than the average for all but the last tested number of training events.

The number of events in training is more important than the size of the networks here. The fluctuation in the average KS values for different network sizes is ∼0.012 and thus smaller than the trend for more training events. As measure for the KS fluctuation, the mean of the standard deviations of the three KS values at the different numbers of training events was used.

In the previous plot one set of events was used for each number of training events. However, especially at low numbers it can make a difference which events are included in the training. Also the network quality will be different even if the networks are trained twice on the same data. To take the variability of the training set and process into account, five different training sets were simulated for each number of events.
Figure 6.3 shows the KS results of the sets. The plot on the left shows the KS values averaged over all model parameters for the different training sets together with their average value. The error bar of the black line is the standard deviation of the colored dots. So the blue dots here are equivalent to the black dots in the previous plot and the black line represents the average of average KS values.
The right plot shows an extrapolation of the KS values. A ad-hoc fit to the KS values is used to estimate the average KS value at higher numbers of training events. It just assumes the most simple function that could describe the data and has no theoretical motivation.

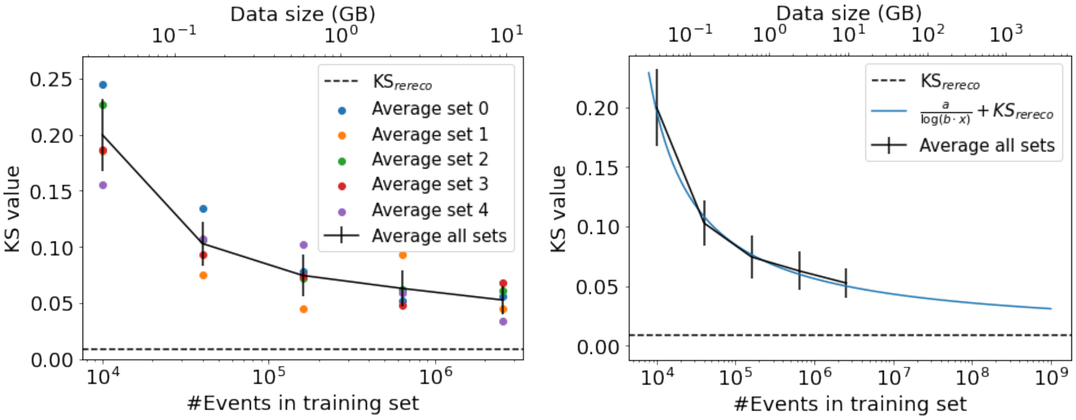


Figure 6.3.: Estimation of the number of events needed to *perfectly* learn the likelihood. KS values between the resolution distributions obtained with the true likelihood and network learned likelihood are shown. The left plot shows the average KS value for five different training sets (colored) together with their average (black). In the right plot, the same black line is used. In addition, a fit to the black data points is included.

The average (average) KS values monotonically decrease for networks that saw more events during the training process. This is expected because more events provide more information about the likelihood landscape that should be learned. However, the KS value only approaches the optimal line asymptotically and does not reach the optimum even for the highest number of training events.

To estimate how many event would be needed to finally get close to the dashed black line, an extrapolation is used. It indicates that a memory usage of several terabyte would be necessary to finally approach the optimal line. However, the likelihood might not have to be learned perfectly, depending on what it is supposed to be used for. In sec. 6.2.4 the KS value is compared to physics sensitivities obtained with the reconstruction based on the respective likelihood for a DeepCore-like toy detector in order to evaluate this question.

6.1.2. Spherical toy experiment

As explained in sec. 5.3.1, networks in both likelihood formulations have been trained for the spherical toy detector configuration. Now their reconstruction performances can be directly compared to each other (and the true likelihood). Similar to the training set, the set of reconstructed events only contains tracks that are completely contained in the detector.

Figure 6.4 shows the resolutions for both formulations together with the resolutions based on the true analytic likelihood. In contrast to the one-string toy detector, all eight parameter of the neutrino interaction model are shown here. For the total energy the difference between the reconstructed and true values was replaced by the difference of their logarithms.

The networks in the different formulations perform similar. This is not surprising because they are trained on similar numbers of events and ultimately represent the same function. The resolutions obtained from the reconstruction with the true likelihood are slightly better than the network results. Especially for the angular resolution there is still potential to improve the networks in the future. The limited number of training events as well as an imperfect training process[1] are reasons for the difference. Also the combination of only considering events completely contained in a spherical detector and the isotropic light emission implemented in the toy experiment makes it difficult for the networks to learn everything about the event direction.

While the parameter resolutions are similar, there is a speed difference between the two formulations. In the *per-sensor* formulation the ChargeNet has to be evaluated once per sensor to calculate the likelihood at a specific θ. However, in the *all-sensor* formulation it has to be evaluated only once. The number of HitNet evaluations is the same in both formulations and is equal to the number of hits.

The mean reconstruction time $\bar{t}_{\textit{per-sensor}}$ for the *per-sensor* networks is 59.0 s, while the mean reconstruction time $\bar{t}_{\textit{all-sensor}}$ for the *all-sensor* networks is only 27.1 s. For their ratio, the following relation holds:

$$\frac{\bar{t}_{\textit{per-sensor}}}{\bar{t}_{\textit{all-sensor}}} \approx \frac{\bar{N}_{hits} + N_{sensors}}{\bar{N}_{hits} + 1} \,, \tag{6.1}$$

---

[1] Imperfect training means that not the best possible values for all network weights were found by the training process.

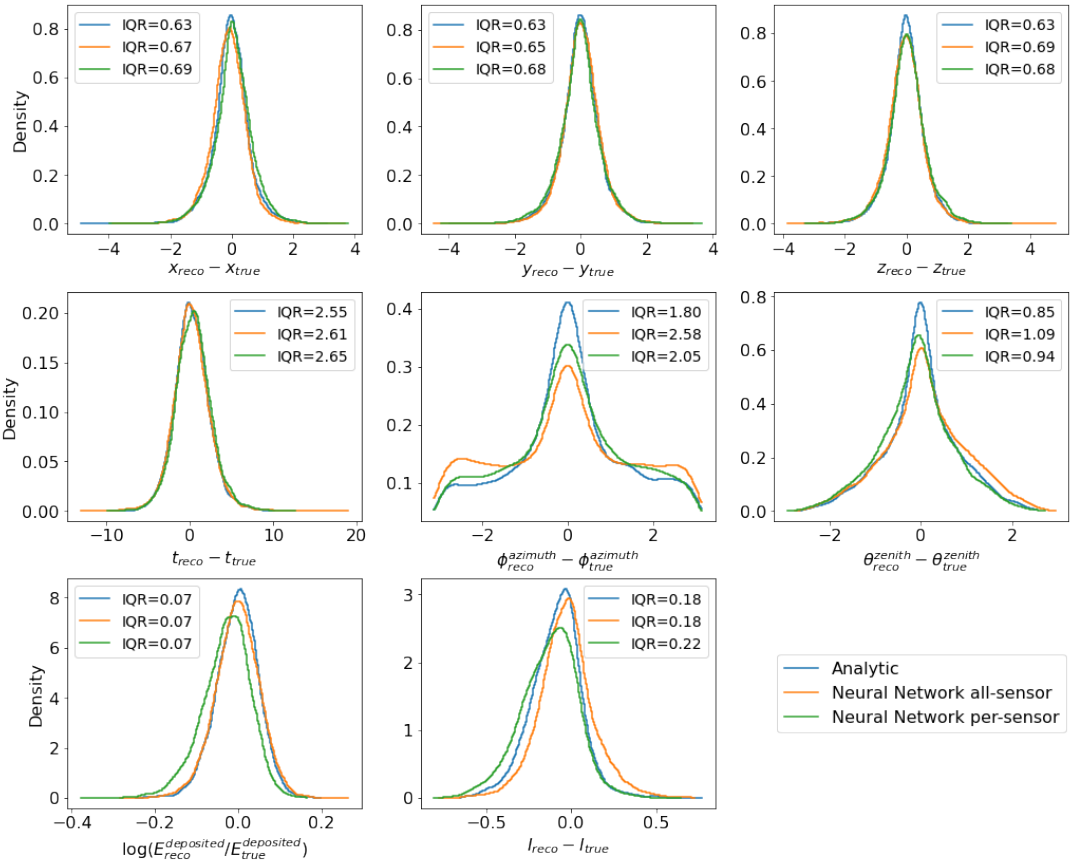


Figure 6.4.: Parameter resolutions of networks trained in both likelihood formulations compared to the resolutions obtained with the true analytic likelihood (which is the same in both formulations). The drawn distributions are the result of a KDE.

where $\bar{N}_{hits}$ is the average number of hits in the reconstructed event sample and $N_{sensors}$ the number of sensors in the detector.

The difference in reconstruction speed will be larger for IceCube which consists of more sensors. That is the reason why, for all following studies including the IceCube/DeepCore detector, networks trained for the *all-sensor* formulation of the decomposed likelihood are used. They should yield the same parameter resolutions with much faster reconstruction times. In addition, the memory required to train a *per-sensor* ChargeNet is higher by a factor of $N_{sensors}$ compared to a *all-sensor* ChargeNet. With $N_{sensors}$ being the number of sensors in the detector.

6.1.3. Detector optimization

A big advantage of the likelihood-free inference technique implemented in FreeDOM is its high flexibility. In the *per-sensor* formulation the HitNet and the ChargeNet only receive per sensor information. If the transformation layer only passes relative and no absolute positional information, the networks basically become independent from the actual detector configuration. So networks trained on a specific detector configuration can be used to reconstruct every possible configuration. This can be used to test the impact of different configurations on reconstruction resolutions without changing or retraining the networks. As a result detector optimization studies can be performed.

An example of such a study in the toy model framework is presented here. A HitNet and a ChargeNet were trained on a simulation sample featuring only a single sensor. The events in the training sample were placed randomly in a sphere with a radius of 50 m around the sensor to cover all possible sensor-event combinations. This simple detector configuration also makes it possible to simulate and train on a huge number of events ($\mathcal{O}(10^7)$).
The detector configuration that should be tested is a 5x5x5 cubic grid. The distance between neighboring sensors is the same in all three dimensions. For the optimization study the sensor density is varied without changing the number of sensors, only their distance to one another. The reconstructed event sample consists of tracks that are also generated in a cubic volume of the size of the smallest detector configuration tested here. So the generation volume is always completely contained in the detector, but the tracks are not necessarily. A minimum number of 4 hits was required for each detector configuration, to only get reconstructable events. This has the effect that smaller detectors reconstruct more low energetic events (that do not deposit 4 hits in larger detectors), even though the events for all detectors were drawn from uniform distributions in all parameters using the same parameter ranges.
Figure 6.5 shows the relative change in parameter resolution (quantified by the IQR) for all eight parameters describing a neutrino interaction. The resolutions are normalized to the best one in the plot.
For most parameters, a more densely instrumented detector is preferred. The three vertex variables show a virtually identical behavior and their resolutions get better with higher sensor densities. Both findings are in agreement with expectations, because the three dimensions are identical in this toy detector and a closer sensor spacing can better resolve positions. The same is true for the reconstruction of the time variable. For all other parameters, the fact that longer tracks have a higher probability not being completely contained in a smaller detector plays an important role. This can be seen most clearly in the energy resolution. If a track is not completely contained in the detector there is essentially light missing and the event seems to be less energetic. That also affects the angular resolutions, because the pointing of the track might not be visible. The angular resolutions start to decrease at lower detector volumes, compared to the energy, because for the directional reconstruction the track does not have to be completely contained in the detector to get a good estimate.

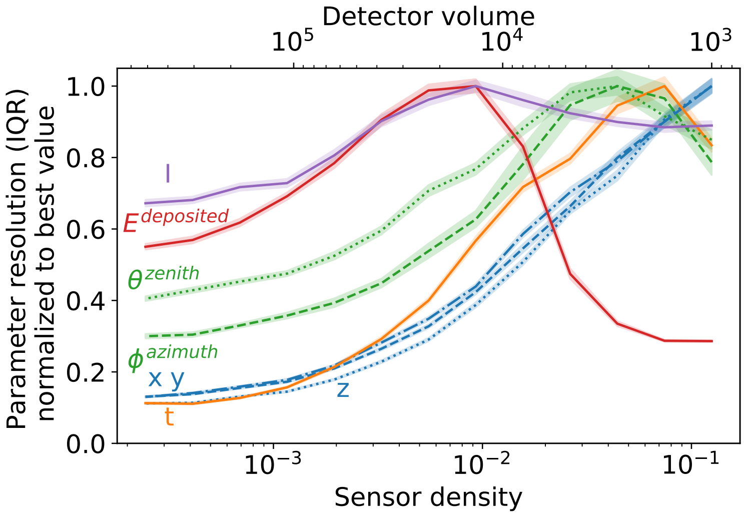


Figure 6.5.: The influence of sensor density on parameter resolutions for a fixed number of 125 sensors in a cubic geometry (with varying sensor distances). Parameter resolutions are quantified by the IQR. The relative resolutions compared to the best one are shown. The error ranges are determined via one-sample bootstrapping [126].

Thus, for a given true energy range, the physics signal region, and a given number of sensors, limited by the funding of the detector, there is an optimal sensor density where the best resolutions can be achieved. FreeDOM can be used to identify this optimal value, thanks to its high flexibility. It is also possible to test completely different (non-cubic) sensor arrangements to find the optimal one.

In this study, only the effect of sensor density on parameter resolutions was investigated. Additional effects of different effective volumes and trigger rates were not considered because they are independent from the reconstruction.

6.2. DeepCore studies

All DeepCore studies presented in this section use the networks described in sec. 5.2, so a ChargeNet and a corresponding HitNet in the *all-sensor* formulation of the decomposed likelihood.

The reconstructed event set consists of 612517 neutrino events (140018 ν_e^{CC}, 259250 ν_μ^{CC} , 124989 ν_τ^{CC}, and 67586 ν^{NC}). That is ∼10% of the nominal oscNext MC set at L6 described in sec. 5.2. None of the events was part of the network training. The true parameter distributions for the set can be seen in fig. 6.6.

The reconstruction performance (sec. 6.2.1) and physics sensitivities (sec. 6.2.2) are compared to the current sate-of-the-art reconstruction RetroReco, which is introduced in sec. 4.4.3. In addition, it is estimated how the size of the training set affects the sensitivities (sec. 6.2.4).

6.2.1. Reconstruction performance

First, the performance of the reconstruction itself is demonstrated.
In fig. 6.6 the one-dimensional best-fit parameter distributions of the two reconstructions are compared to the true “injected” distributions for each parameter of the neutrino interaction model described in sec. 4.4.1. In addition, the logarithm of the total deposited energy is shown.

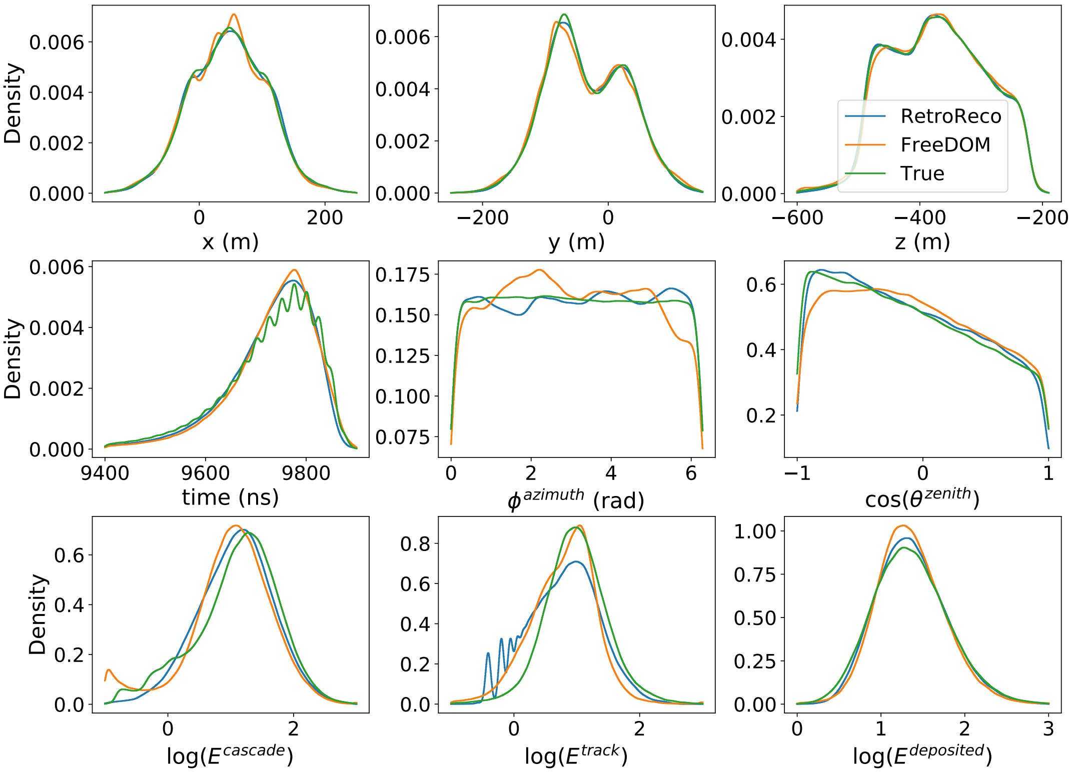


Figure 6.6.: Best-fit distributions for FreeDOM (orange) and RetroReco (blue) compared to the true injected parameter distributions (green). All eight parameters of the neutrino interaction model are included together with the total deposited energy. KDEs of the distributions are shown.

All three y and z vertex distributions look similar. For the x vertex, a few peaks are visible for FreeDOM compared to the truth, due to a bias towards string positions. This bias is less pronounced in the y-distribution because there are essentially only two positions around which the strings cluster. The true time distribution shows a wiggle structure that is not present in the reconstructed distributions. This structure is not completely understood but probably results from the way events are triggered in IceCube. The time period of these wiggles is 25 ns which is the FADC sampling

period.
The reconstructed azimuth distributions also show an oscillation structure. This is likely the result of the irregular detector grid which makes it more difficult to reconstruct certain directions in the horizontal plane. In addition, FreeDOM reconstructs less very up-going events while RetroReco reproduces the zenith distribution well.
A bias towards lower cascade energies is visible for both reconstructions, slightly stronger for FreeDOM. The small peak in the FreeDOM distribution at very low cascade energies are events that hit the minimizer boundary. Also for the track energy both reconstructions return more small and less high values than the truth. RetroReco returns discrete track energy values which explains its structure at low values. FreeDOM reconstructs a continues value for the track energy but nearly never returns high track energy values. This might be the case because of the low number of long tracks in the training sample. The total energy distributions look similar. The reconstructed distributions are biased towards the medium energy range, again FreeDOM more than RetroReco.

Timing

The per event reconstruction time becomes important when millions of events have to be reconstructed. In the following the reconstruction speed achievable with FreeDOM is compared to the time RetroReco needs to reconstruct events.
Figure 6.7 gives an impression of the speed that can be achieved with FreeDOM. The upper plot shows the time needed to reconstruct an event for FreeDOM and RetroReco without paralellization on a Graphics Processing Unit (GPU). FreeDOM is ∼35% faster than RetroReco an average. The lower left plot shows the number of likelihood (LLH) calls needed to minimize the FreeDOM likelihood.
Neural networks can be evaluated on GPUs very efficiently [127]. This allows to reconstruct many events in parallel and reduce the per event reconstruction time. In the lower right plot the effective time per likelihood evaluation is shown versus the number of parallel evaluations in a GPU batch. In case of no paralellization, represented by the first blue dot, it takes about 5 ms per likelihood evaluation. Multiplying this with the average number of evaluations yields the average time shown in the upper plot. However, the time per likelihood evaluation can be strongly reduced increasing the GPU batch size. If several hundreds of events are reconstructed in parallel this time is $\sim 50\,\mu$s, i.e. about a factor of 100 faster. Consequently, the average reconstruction time per event is reduced to about 0.2 s, which is more than a hundred times faster than RetroReco. The GPU used in this example is a NVIDIA Tesla V100 [128].

Parameter resolutions

The quality of a reconstruction method is mainly determined by its parameter resolutions, i.e. how close the reconstructed parameter values are to their true values. To illustrate the different parameter resolutions depending on the true parameter values, fig. 6.8 shows the reconstructed versus true values for the same choice of parameters

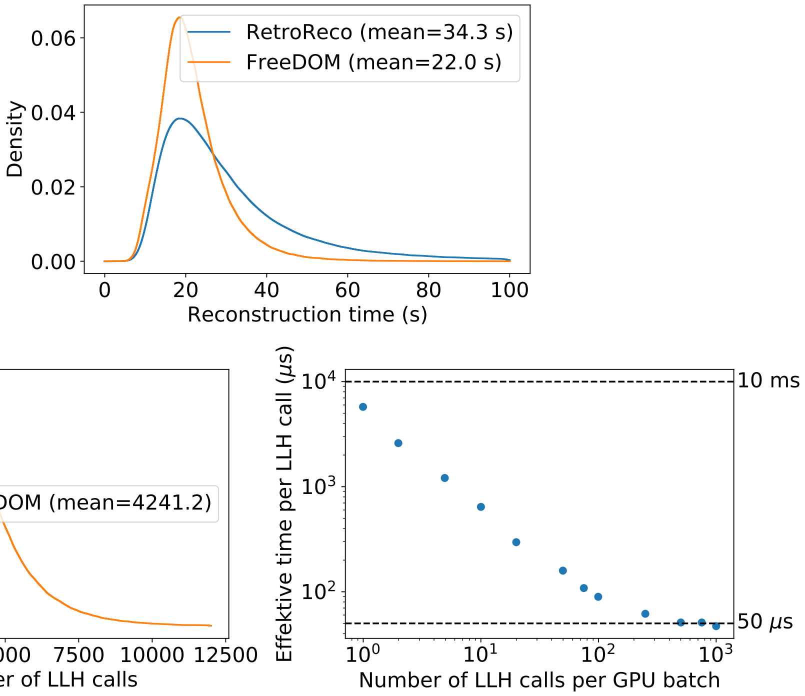


Figure 6.7.: Time to reconstruct oscNext DeepCore events. The upper plot shows the time needed to minimize the likelihood per event (in case of no GPU paralelization). The two lower plots show the number of likelihood calls per event for FreeDOM and the effective time per call for different GPU batch sizes. KDEs of the distributions are shown

as in fig. 6.6.
For the vertex distributions the one to one (1:1) line is always inside the respective IQR. The maximum distance of the median line to the 1:1 line is $[6.6, 4.2, 1.6]\,m$ for RetroReco and $[10.5, 4.9, 2.0]\,m$ for FreeDOM for x, y, and z respectively. The slight curve structure in x and y arises from the non-homogeneous string placement and is similarly visible in both reconstructions. The time experiences a bias to values between roughly 9700 ns and 9800 ns. The absolute hit time is defined by the trigger time and most of the hits in each event have time values in that range. This pulls reconstructions towards these values.
For high azimuth values, the median RetroReco line is closer to the 1:1 line than the FreeDOM median line, while for low azimuth values this difference is smaller. This is not expected because 0 and 2π are identical for azimuth and it suggests that the angle was not modeled correctly by the networks. The equality of 0 and 2π is also the reason for the bending of the azimuth lines at these values. A “symmetric” distribution with mean 0 for example would result in a median of π in this plot. The form of the zenith curves looks similar for both reconstructions, but RetroReco’s IQR for up-going events is 0.076 rad smaller than FreeDOM’s, while for down-going the

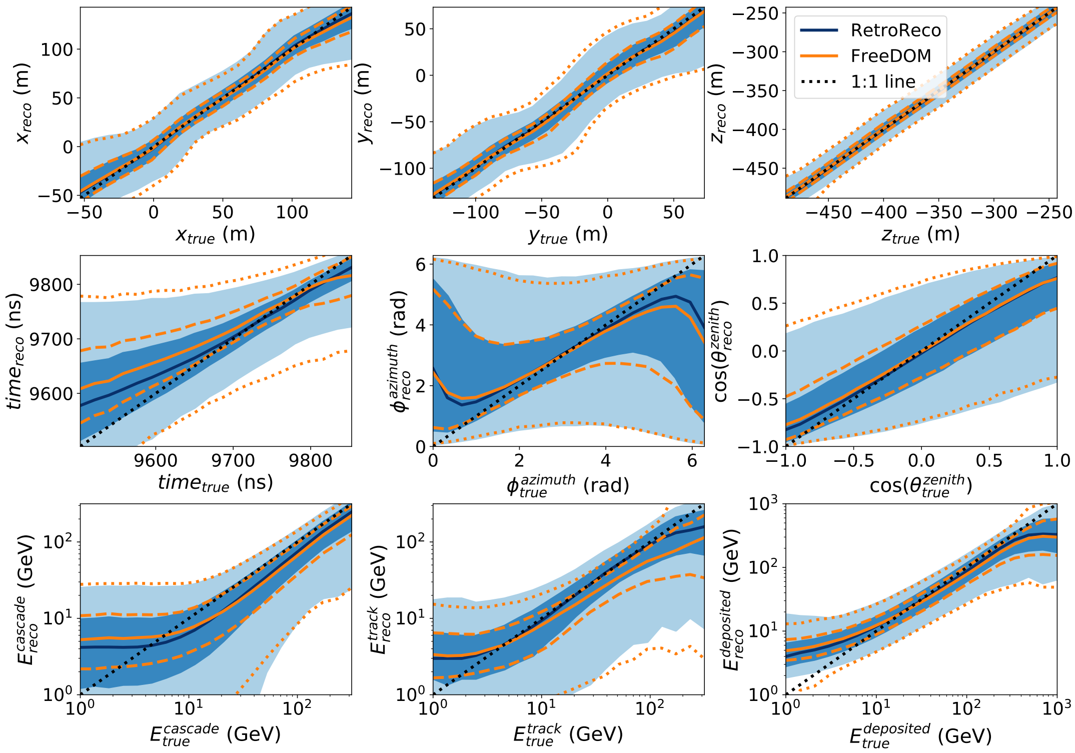


Figure 6.8.: Reconstructed vs true parameter values for FreeDOM (orange lines) and RetroReco (blue surfaces). The median as well as the IQR and 90 percent inter quantile range are shown. In addition, the one to one line is represented as dotted black line.

difference is only 0.028 rad.
As seen before the cascade energy is underestimated in both methods. That is the case for nearly all values except for those below 5 GeV. In addition, FreeDOM underestimates the track energy starting at about 10 GeV of true track energy. The total deposited energy is reconstructed similarly by both methods, the maximal relative difference in IQR is 16% of the (RetroReco) IQR value. All three energies show a bending at the edges of the shown energy ranges. The reason for over-predicting low energy values is that low energetic events usually do not contain many hits. Only events with more hits than expected on average pass the event selection and the low energy events in this plot can be considered as “over-fluctuated”. For high-energy events, it is exactly the opposite. Figure 5.10 shows the different charges the same event can deposit in the detector if resimulated. This also gives an impression about the variation of the number of hits.

As explained in sec. 4.4.1, the two most important parameters for atmospheric neutrino oscillation analyses with IceCube are the total deposited energy and the cosine of the zenith angle. One dimensional resolutions for these parameters are shown in fig. 6.9. The plot contains events with a true deposited energy between 5 GeV and 50 GeV. As can be seen in fig. 6.8, events below 5 GeV can not be reconstructed well with DeepCore and above 50 GeV there is nearly no atmospheric oscillation (see fig. 6.16).

For the energy, the logarithm of the ratio between the reconstructed and the true value is used to quantify the resolution. It shows the relative energy difference which is important because it makes a difference if a 1 GeV event is reconstructed with a precision of 1 GeV compared to a 50 GeV event. Furthermore, a bias would only shift the entire distribution along the x-axis, not change its width. For the cosine of the zenith angle, the difference of the reconstructed and true value is used.

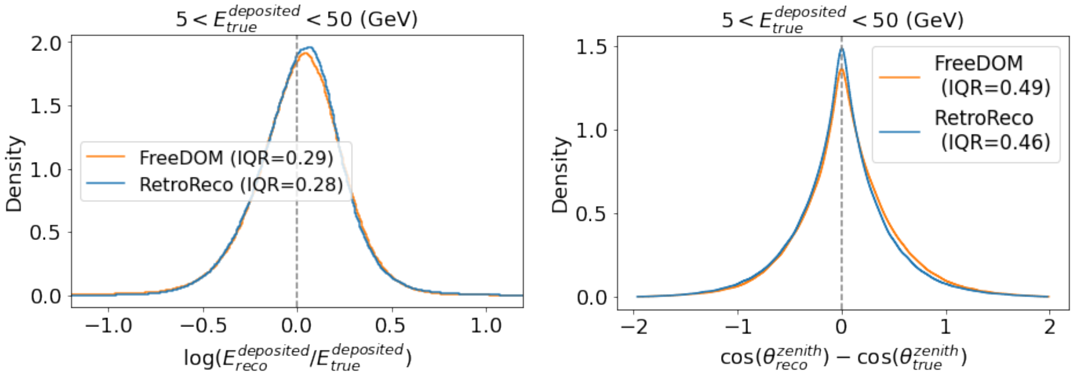


Figure 6.9.: OscNext resolutions for total energy and cosine zenith angle. FreeDOM (orange) is shown together with RetroReco (blue). KDEs of the distributions are shown.

The total energy resolution in the considered energy range is similar for both reconstructions. The IQR values differ by about 3%. Both methods also experience the same bias towards higher values. This might seem surprising because fig. 6.8 shows that the energy is underestimated for most of the considered energy region. However, there are more low energy events in the sample, which can be seen in fig. 6.6. These events tend to be reconstructed at higher energies, so the overall bias here is positive.

The difference in zenith resolution is about 6% with RetroReco being better than FreeDOM. The FreeDOM distribution is skewed to the right which likely results from the fact that FreeDOM has difficulties reconstructing very up-going events (see fig. 6.6).

At higher energies more light is deposited in the detector and therefore more information is available for a reconstruction algorithm. Consequently, higher energetic events should also be better reconstructed.

Figure 6.10 shows the parameter resolutions versus the true deposited energy for the vertex position and the deposited energy. For the angular resolutions, the true track energy is used on the x-axis, because the direction of long tracks is expected to be reconstructed better. In addition to azimuth and cosine zenith resolutions, the angle between the reconstructed and true direction ($\Delta\Psi^{reco}_{true}$) is included. The cascade and track energy resolutions are shown versus their respective truth. The time resolution is not included, because its absolute value is less important for an analysis compared to the other parameters.

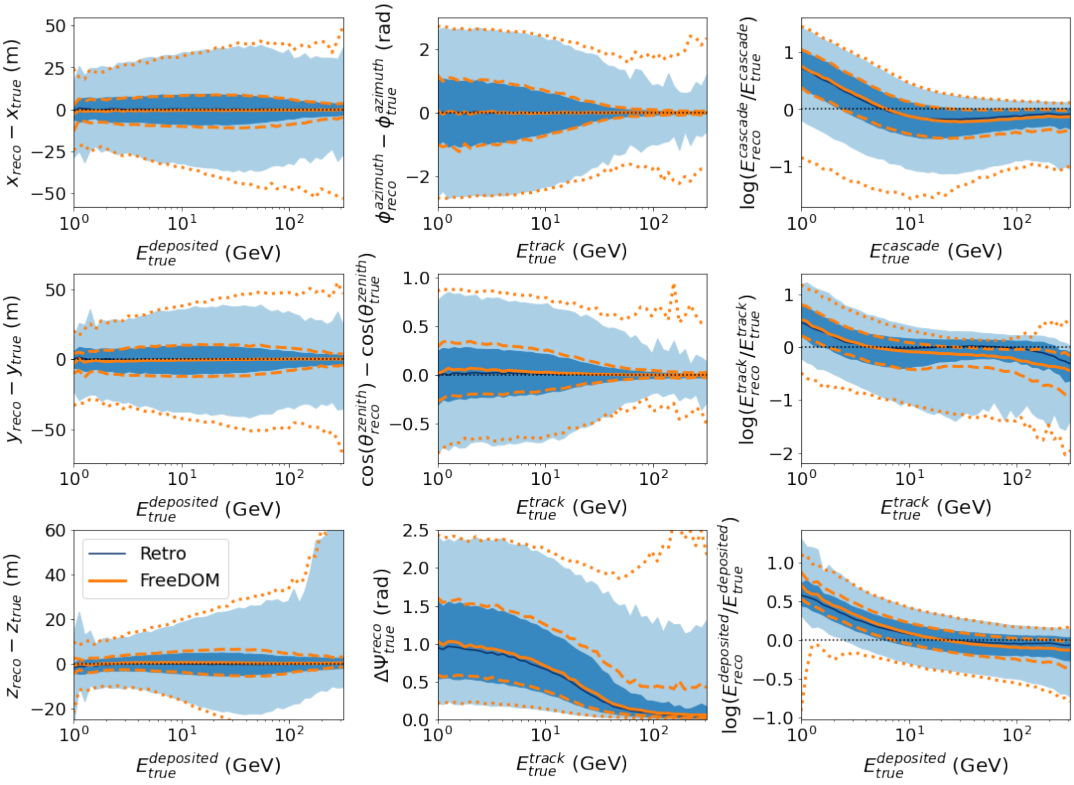


Figure 6.10.: Parameter resolutions versus true energy for FreeDOM (orange lines) and RetroReco (blue surfaces). Depending on the parameter, different energies are used on the x-axis. The median as well as the IQR and 90 percent inter quantile range are shown.

Up to about 100 GeV vertex resolutions get worse before they improve again. A reason for this is that low energetic events only consist of a few hits rather close to the interaction vertex. So their resolution is better than the resolution for medium energy events, which more often contain hits further away from the vertex. At high energies the large number of hits allows to better locate the vertex again. Above 100 GeV some events are reconstructed more then 30 m above their actual position. These events are sitting below the detector but are bright enough to deposit light in lower DeepCore DOMs. As a result, they are reconstructed closer to the detector. This only happens for events below DeepCore because it is located at the bottom of

IceCube. That means there are DOMs above DeepCore that can be used to properly locate the event.
All angular resolutions improve with higher track energies, i.e. longer tracks. These tracks typically produce more elongated signatures that allow for a better estimation of the orientation of the event. The median $\Delta\Psi_{true}^{reco}$ value for RetroReco goes from about 1 rad at 1 GeV track energy down to about 0.15 rad at 300 GeV, while for FreeDOM it goes from about 1 rad to about 0.39 rad. Internal studies [129] showed that this behavior is less strong for reconstructions based purely on neural networks.
The energy plots show the same as in fig. 6.8 and are only included for completeness. It is noticeable that, for all parameters, the 90 percent inter quantile range for FreeDOM does not improve at high energies but the IQR does. This indicates that the likelihood space at high energies is not well modeled for every event in the sample. This is not unexpected given that the networks are mainly trained on low energetic events.

Finally, looking at the resolutions for the different flavors and interactions is important, because the networks were mainly trained with muon neutrinos and no tau neutrino events were present in the training set. It has to be tested, in comparison with RetroReco, if there is any unexpected behavior for the FreeDOM reconstruction of tau events. It is also interesting how the reconstruction performs for the different event topologies described in sec. 4.4.2, so if there are differences between tracks and cascades.
Figure 6.11 shows the energy and zenith resolutions for the different flavors and interactions. The same cut on the true energy is used as in fig. 6.9.
The different interactions show slightly different energy biases and resolutions. The energy of ν_e^{CC} events is overestimated, while the energy of ν^{NC} events is underestimated. That is because of the different light yields of electromagnetic and hadronic cascades (see sec. 4.4.2). This behavior is identical for both FreeDOM and RetroReco. The ν_μ^{CC} energy resolution shows the larges difference between the reconstructions.
In agreement with the expectation, the angular resolution for ν_μ^{CC} (tracks) is better than for all other interactions (cascades). Again, both reconstructions show similar relations between the interactions, with RetroReco showing better resolutions.

FreeDOMs performance for tau neutrinos is similar to RetroRecos. So even there were no taus present in the training, they can be reconstructed. It is surprising that the energy resolution for ν_μ^{CC} is not as good as it should be, looking at the RetroReco resolutions, even though it was the most frequent interaction in the training sample. This again suggests, similar to the toy studies in sec. 6.1.1, that the track energy is the most difficult parameter to model.

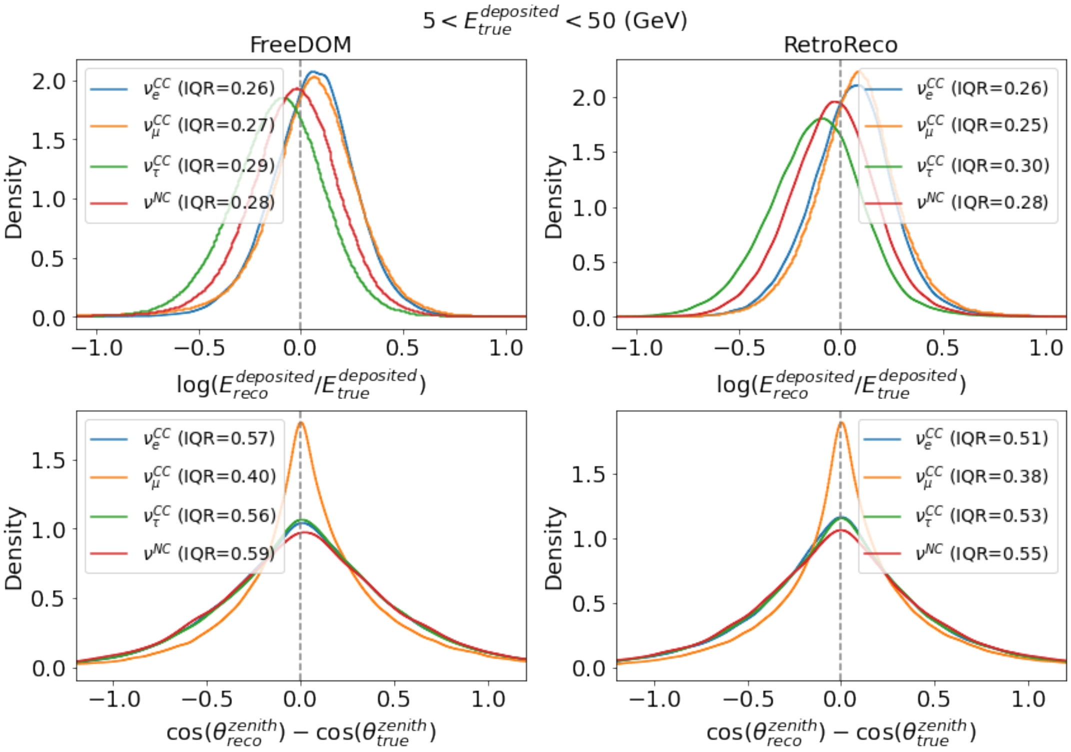


Figure 6.11.: Energy and zenith resolution per flavor and interaction. All neutral current (NC) events are combined in one distribution. FreeDOM (first row) and RetroReco (second row) are shown. KDEs of the distributions are shown.

Particle IDentification (PID)

The particle identification has a significant impact on many of IceCube's atmospheric neutrino oscillation analyses (see fig. 4.10). That makes it important for a reconstruction to also provide a PID. As explained in sec. 4.4.2, a likelihood function can also be used for particle identification in the IceCube low energy context.

For each event a second fit is performed for which the track energy is fixed to zero. This fit is seeded at the best-fit result of the full eight dimensional fit and uses the same minimizer settings. The FreeDOM PID score is defined as:

$$PID = \frac{min|_{E_{track}=0}(-\mathcal{L}) - min(-\mathcal{L})}{Q_{tot}}\,, \tag{6.2}$$

where $\mathcal{L}$ is the logarithm of the likelihood and Q_{tot} the total charge in the event. FreeDOM actually does not provide the likelihood but the likelihood-to-evidence ratio. However, since the evidence is identical for both fits (same $\boldsymbol{x}$), it cancels out in the calculation. The reason for including Q_{tot} is that it improved the PID performance by about 5%.

Based on this PID score the Receiver Operating Characteristic (ROC) curves shown in fig. 6.12 could be calculated. These curves visualize, for different cut values on the PID score, how many tracks are identified correctly (true positive) versus how many cascades are falsely classified as track (false positive). Two pairs of ROC curves are shown. One containing all events above the DeepCore energy threshold, the other focusing on high energy events which are easier to distinguish, because high energy tracks are longer and differ more from cascades. The Area Under the Curve (AUC) quantifies the goodness of the particle identification. Respective curves for FreeDOM and RetroReco are included.

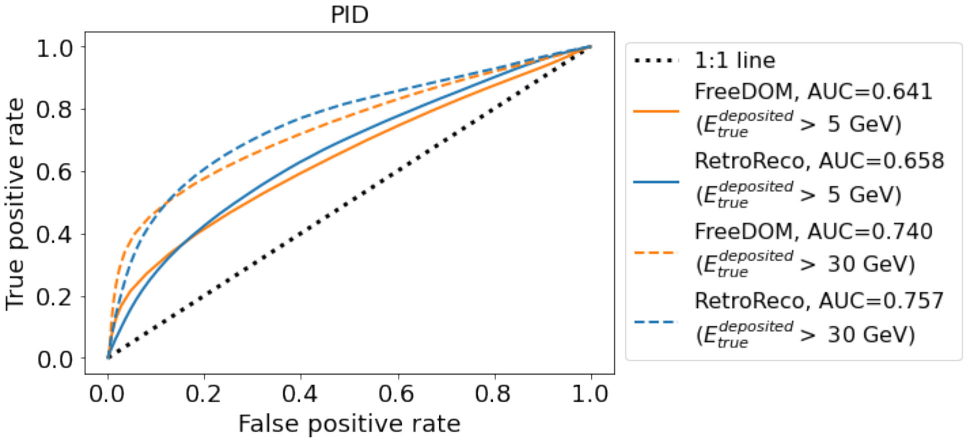


Figure 6.12.: Particle identification based on likelihood ratios for different energy cuts. ROC curves for FreeDOM (orange) and RetroReco (blue) are shown. For the solid lines all events with true energies above 5 GeV are used, while for the dashed lines only higher energetic events are included. The area under each curve is given in the legend. The black dots mark the one to one line.

At low false positive rates, the FreeDOM curves raise faster, which allows to extract a purer track sample, compared to RetroReco, that still contains about 20% of all tracks. Overall, the two PIDs are comparable with RetroReco being about 3% better in the AUC.

This demonstrates that the likelihood learned by the FreeDOM networks can be used for a PID. No separate networks or classifiers have to be trained. Considering FreeDOMs previously seen difficulties modeling long tracks, there should also be some potential in its PID performance.

Stability for systematic sets

The MC simulation used to train the networks was produced with specific values for the different systematics that affect an IceCube atmospheric oscillation analysis. These values are referred to as nominal in the following. Important systematic parameters are the DOM efficiency, the optical properties of the hole ice, and the bulk ice scattering and absorption length. All of which are shortly introduced in the following.

The DOM efficiency determines how effective the modules detect photons. A increased DOM efficiency leads to more hits in an event and vice versa. Hole ice refers to the ice in the string holes which were drilled during the deployment of the detector. These holes refreeze after the strings are deployed, producing an hole ice cylinder around each string. This ice is not as transparent as the surrounding deep Antarctic glazier ice (which is called bulk ice) and effectively changes the angular acceptance of the deployed modules. The optical properties of the bulk ice, mainly the photon scattering and absorption, also affect the times and number of expected hits in an event.

For the purpose of oscNext, different systematic MC sets were produced. Each with one or more systematic parameters changed compared to the nominal set. FreeDOM has to be robust against changes of the systematic parameters to be able to reconstruct real data, because it is unlikely that the nominal assumptions about this parameters correspond to the truth. To test FreeDOMs stability against systematics, events from eight systematic sets are reconstructed and compared to the performance for the nominal set. RetroReco is used to estimate the expected reconstruction differences for each systematic set.

Table 6.1 shows the considered systematic changes with respect to the nominal set.

Set number	change to nominal
0	no change (nominal set)
1	-10% DOM efficiency
4	+10% DOM efficiency
100	angular acceptance of PMT
106	-3% DOM efficiency, angular acceptance of PMT
150	angular acceptance of PMT
152	angular acceptance of PMT
500	+5% scattering and absorption in bulk ice
503	-5% scattering and absorption in bulk ice

Table 6.1.: The considered systematic sets and their difference to the nominal set.

About 14000 events (~20% ν_e and ~80% ν_μ) per set are used for this comparison. Figure 6.13 shows the total energy resolution for all systematic sets compared to the nominal.

The standard deviation of the different IQR values is about 1.3% of the IQR value of

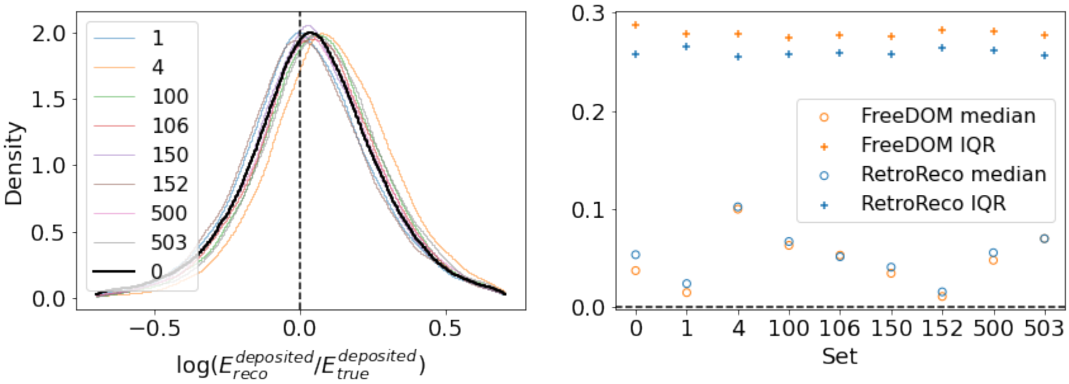


Figure 6.13.: FreeDOM energy resolutions for systematic sets compared to RetroReco. The left plot shows the FreeDOM energy resolution distributions for all sets on top of each other. The drawn distributions are the result of KDEs. The right plot shows their median and IQR compared to the respective RetroReco values.

the nominal set. So FreeDOM's energy resolution is similar to the nominal one for all systematic sets.

The median values show a stronger fluctuation for different systematic sets. Their standard deviation is about 9% of the nominal IQR value. The standard deviation of the medians must also be compared to the IQR because it is important how much the distributions are shifted compared to their width not to their absolute position. The stronger fluctuation of the medians is to be expected and RetroReco shows the same behavior. For example looking at the DOM efficiency sets, a higher efficiency will lead to more hits in the detector and consequently to higher reconstructed energies. Therefore, it is consistent that the median for this systematic set is higher than for the nominal set, while the median value for the systematic set with a lower DOM efficiency is lower than the one for the nominal set.

Figure 6.14 shows the same comparison as fig. 6.13 for the cosine of the zenith angle.

Similar to the energy resolution, the standard deviation of the different IQR values is only about 2.1% of the IQR value of the nominal set. The zenith resolution is not strongly affected for the systematic sets.

In contrast to the energy, there are no strong biases visible for the angle. Here, the standard deviation of the medians is about 3.7% of the nominal IQR value. Again, the biases are strongly correlated between FreeDOM and RetroReco.

In summary, FreeDOM is stable against systematic changes in the MC sets. This shows that it did not overfit the nominal set, which is important for its application to real data.

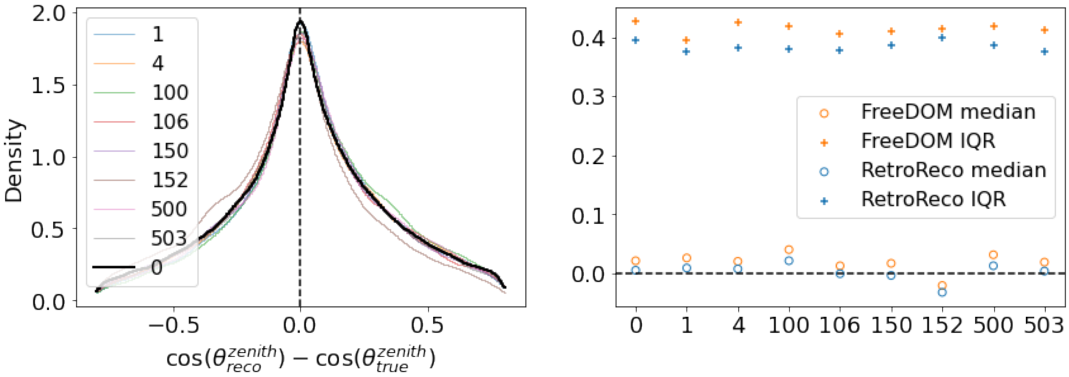


Figure 6.14.: Zenith resolution for systematic sets compared to RetroReco. The left plot shows the FreeDOM zenith resolution distributions for all sets on top of each other. The drawn distributions are the result of KDEs. The right plot shows their median and IQR compared to the respective RetroReco values.

Data-MC agreement (pre-fit)

Finally, the FreeDOM performance on real detector data has to be compared to the MC performance. This ensures that the networks did not learn MC specific features and are applicable to data.
The oscNext data set consist of 9.5 years of DeepCore data, but is not unblinded[2] yet. Therefore, only 1% of all available data are used here, which corresponds to 10698 data events. After applying the final level cuts that are used for oscNext analyses, but based on the FreeDOM reconstruction, this number is reduced to 3142 events.

It is not possible to compare parameter resolutions for data events, but the best-fit distributions between MC and data events can be compared. Figure 6.15 shows the reconstructed distribution for the nominal MC set and the data. Since atmospheric muons are the largest background in the detector, muon MC simulation is also included in the plot. This corresponds to 640 events after final cuts, which were also reconstructed with FreeDOM. The included uncertainty for the data points is the Poisson uncertainty. The MC events are weighted by an atmospheric flux model (Honda [23]), their oscillation probability, and interaction cross-section. The small plots at the bottom show the ratio between data and MC events in each bin compared to the uncertainty on the data.
Figure 6.15 shows data-MC comparisons for the IceCube low energy analysis variables for FreeDOM and RetroReco.
These plots were made pre-fit, which means that, for example, the oscillation parameter values used to calculate the MC event weights are the nominal ones and not

---

[2] Blind analyses are performed in IceCube. That means it is not allowed to look at analysis variable distributions for data events.

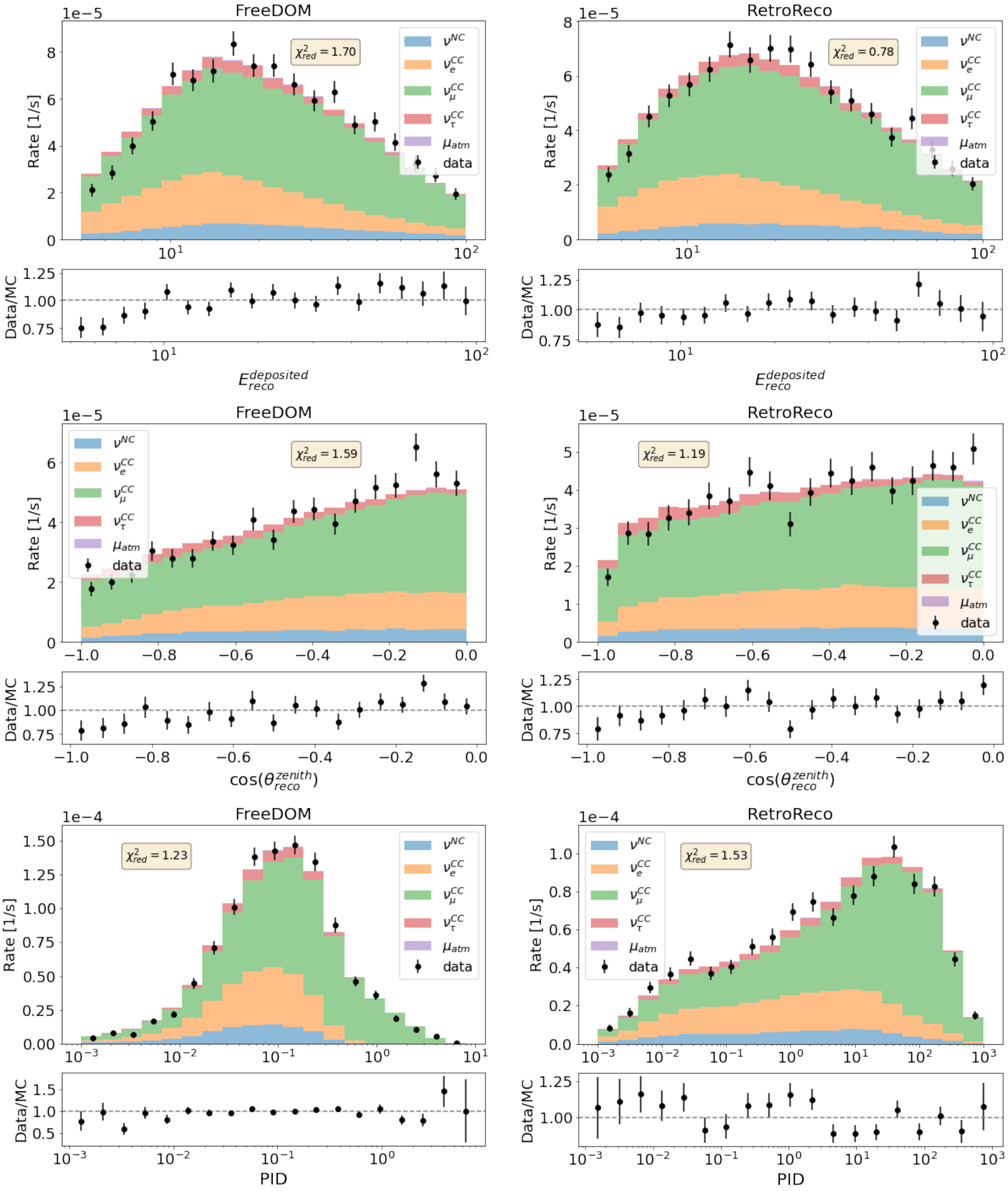


Figure 6.15.: Pre-fit data-MC agreement for total deposited energy (first row), cosine zenith (second row), and PID score (third row). The reduced χ^2 value for the data (black dots) and MC expectations (colored bars) is given in each plot. FreeDOM (first column) and RetroReco (second column) data-MC comparisons are shown.

extracted from a fit to the data. Using post-fit values would violate the blindness and is not allowed at this point. Considering this fact, FreeDOM shows an acceptable data MC agreement for all three variables. None of the reduced χ^2 values is significantly above one. The existing differences are similar to RetroReco, e.g. the lack of low energy events in data.
Even though FreeDOM was exclusively trained on the nominal MC set, it can be used to reconstruct real detector data.

6.2.2. Physics sensitivities

In this section, the impact of the reconstruction on different analyses is presented. The PISA [130] software framework, which was developed for IceCube low energy analyses, is used to calculate the sensitivities.

Atmospheric neutrino oscillations are mainly ν_μ to ν_τ oscillations [10]. Figure 6.16 shows this oscillation probability depending on the neutrino energy and the cosine of its zenith angle relative the IceCube detector. A cosine of 1.0 means that the event is down-going (from the detector point of view), so the neutrino interaction happened in the atmosphere above the South Pole. Down-going events only travel a few kilometers and do not traverse any Earth matter. A cosine of -1.0 means that the neutrino came from the direction of the Earth and is up-going. These events traversed the entire Earth. A neutrino production height of 20 km [23] was assumed to translate $\cos(\theta^{zenith})$ into a traveled distance.
The interesting region of the oscillogram illustrated in fig. 6.16 is the lower left quarter. At GeV energies, there are no oscillations for down-going to horizontal events because their traveled distance is too short. Also above ~60 GeV no ν_μ to ν_τ oscillation is visible. Sub-GeV energies are very interesting for atmospheric oscillations, but DeepCore can not detect or reconstruct such low energetic neutrinos (as demonstrated in the previous section). IceCube's sensitivity to atmospheric oscillation physics comes from precisely examining this oscillation pattern based on a high number of detected atmospheric neutrino events.
For an analysis, events are binned in the three dimensions total deposited energy, cosine of the zenith angle, and PID score. It is not yet possible to identify GeV tau neutrinos in IceCube, but the PID introduced in sec. 4.4.2 aims to separate ν_μ^{CC} from all other flavors and interactions. Each analysis bin contains a number of detected events with the respective reconstructed quantities. Figure C.7 in the appendix shows an example of such an event histogram.

Definition of the sensitivity

All sensitivities presented in this thesis are so-called "Asimov" sensitivities. The Asimov approach is introduced e.g. in [131, 132]. It assumes that the sensitivity of an experiment can be calculated from the mean experimental outcome. Therefore, instead of sampling pseudo experiments, the exact model predictions for the number

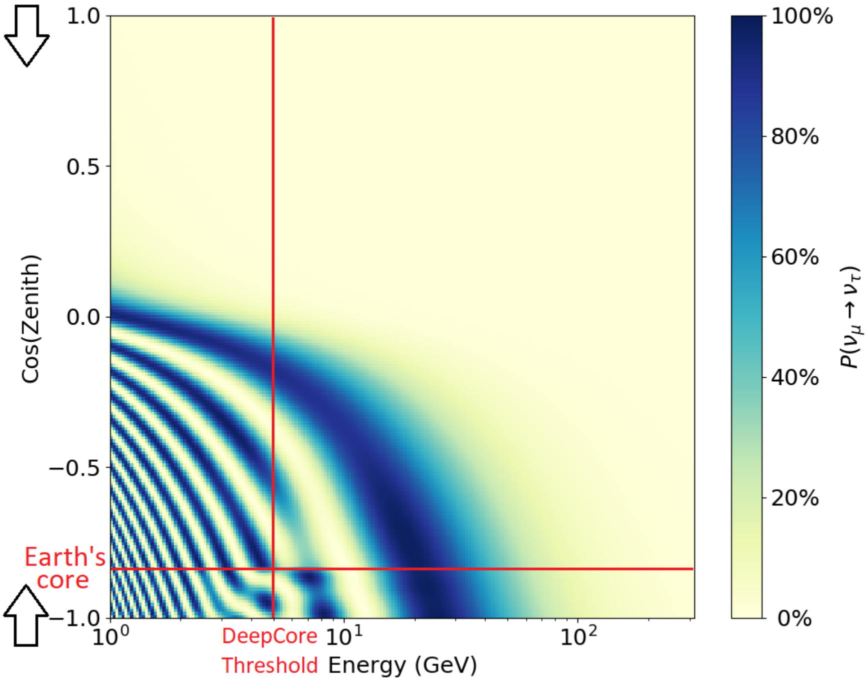


Figure 6.16.: Atmospheric $\nu_\mu \rightarrow \nu_\tau$ oscillation probability depending on neutrino energy and cosine of its zenith angle relative to the detector. For zenith values below the horizontal red line neutrinos traverse the inner iron core of the Earth. The vertical red line indicates DeepCore's energy threshold.

of events in each analysis bin are used in the sensitivity calculation.
As metric to quantify the sensitivities a modified χ^2 is used, which takes into account the uncertainties arising from the finite amount of MC simulation. It is defined as follows:

$$\chi^2_{mod} = \sum_{i}^{N_{Bins}} \frac{(N_i - n_i)^2}{n_i + \sigma_i^2}, \tag{6.3}$$

where N_{Bins} is the number of analysis bins, N_i the observed number of events in bin i, n_i the expected number of events in bin i, and σ_i the uncertainty on n_i.

The sensitivities of two IceCube low energy analyses are presented in the following. Once using the state-of-the-art reconstruction RetroReco and once using FreeDOM.

Oscillation parameters

The first analysis aims to determine the atmospheric neutrino oscillation parameters. Oscillation parameters refers to the neutrino mixing angles and the differences of the squared masses of the neutrino mass eigenstates. As illustrated in equ. 2.16, for the simplified two flavor case, they define the amplitude (mixing angle) and frequency (mass difference) of the oscillation. This analysis is performed by determining the position and shape of the oscillation maxima and minima shown in fig. 6.16.
As mentioned before, atmospheric neutrino oscillation is mainly ν_μ to ν_τ oscillation. Hence, DeepCore looks for ν_μ disappearance respectively ν_τ appearance in the data and is mainly sensitive to the mixing angle θ_{23} and the mass difference Δm^2_{31}. The latest published DeepCore oscillation parameter studies can be found in [133] (ν_τ appearance) and [134] (ν_μ disappearance).

The sensitivity to θ_{23} and Δm^2_{31} is shown in fig. 6.17. A χ^2 scan with fixed systematic parameters is used to demonstrate the impact of FreeDOM on the analysis. The 90% contour level is shown which corresponds to a $\Delta\chi^2$ value of 4.61 above the minimum for two degrees of freedom [135].

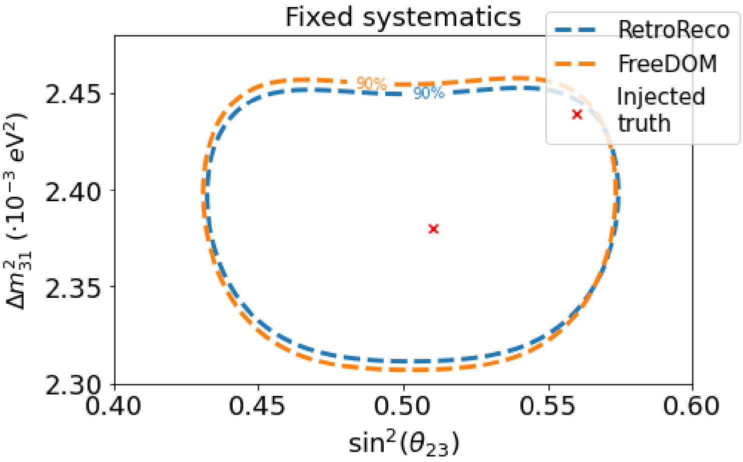


Figure 6.17.: Sensitivity to the atmospheric oscillation parameters θ_{23} and Δm^2_{31}. The 90% contour level ($\Delta\chi^2 = 4.61$) is shown using the RetroReco reconstruction (blue) and using FreeDOM (orange). The red cross marks the injected true parameter values. All systematic parameters were fixed to their true value for the scan.

The constraint on the mixing angle θ_{23} is nearly identical for both reconstructions, with the FreeDOM curve shifted slightly to the left.
The mass difference Δm^2_{31} can be better constraint with RetroReco. The constraining power for this parameter mainly comes from tracks with energies around 15-20 GeV. So FreeDOMs difficulties to correctly model higher energetic tracks is likely the cause for the visible difference here.

NMO

The second analysis is the determination of the neutrino mass ordering introduced in sec. 2.1.2. IceCube's NMO sensitivity comes from the different impact of matter (sec. 2.2.2) on neutrinos and anti-neutrinos.
Matter effects only become important in very dense media like for example Earth's core. In fig. 6.16 these effects are visible as distortion of the oscillation pattern for very up-going neutrinos, so neutrinos that traversed the inner core of the Earth. In the normal ordering this distortion only affects neutrinos, while in the inverted ordering it only affects anti-neutrinos. IceCube can not distinguish between neutrinos and anti-neutrinos, but they have different interaction cross-sections ($\sigma(\nu) \sim 2\sigma(\bar{\nu})$, see fig. 2.2) and the atmospheric neutrino flux is higher than the anti-neutrino flux ($\phi(\nu) > \phi(\bar{\nu})$ [23]). Therefore, IceCube detects more than twice as many neutrinos than anti-neutrinos and the NMO can be determined based on the strength of the distortion in the measured data. The matter effects have the strongest influence between 5 and 10 GeV, which is just above DeepCore's energy threshold. As a consequence, DeepCore's sensitivity to the NMO is below a discovery threshold of 3σ. Please refer to [136] for the latest published DeepCore NMO study.

NMO sensitivity refers to the median sensitivity, which is defined by the probability that a test statistic result of an experiment assuming the wrong ordering is below (NO=True)/above (IO=True) the median test statistic result of the true ordering assumption. As test statistic the metric difference between a fit assuming the NO and a fit assuming the IO is used. In this thesis, a χ^2 is used as metric so the test statistic would be $\chi^2_{NO} - \chi^2_{IO}$.
More information about the NMO determination can for example be found in [137] or [138].

Figure 6.18 shows test statistic distributions of 300 pseudo experiments for RetroReco and FreeDOM. The true parameter values for the inverted ordering were obtained from an Asimov fit, assuming the inverted ordering, to the normal ordering with injected true parameter values. For this plot only the atmospheric oscillation parameters θ_{23} and Δm^2_{31} were free for the fit while all systematic parameters were fixed to their nominal values. In addition, the Asimov sensitivity values are shown to demonstrate that the Asimov assumption is (approximately) correct in this case.
The one-sided sensitivities calculated from the Asimov values are $\sigma_{\text{RetroReco}} = 1.4$ and $\sigma_{\text{FreeDOM}} = 1.3$.
Similar to the first analysis, both reconstructions lead to similar sensitivities with slight advantages for RetroReco. RetroReco's better PID performance is expected to be the main reason for the difference here, because the PID has the strongest influence on the NMO (see fig. 4.10).

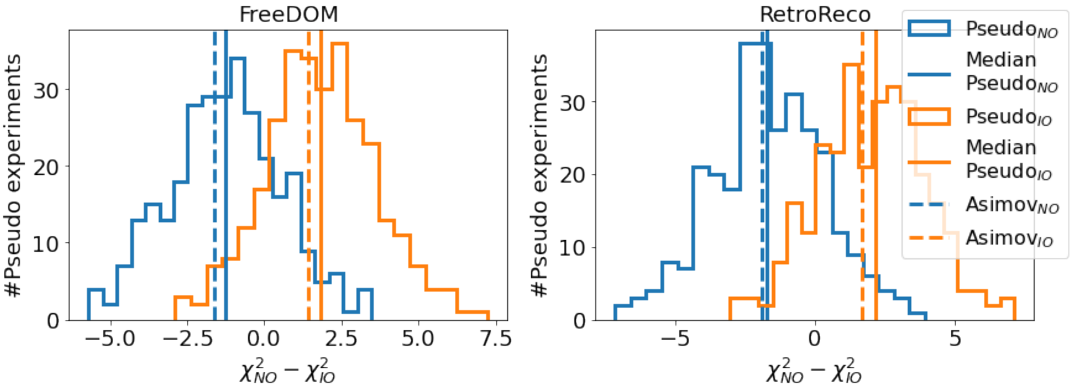


Figure 6.18.: Test statistic values of 300 pseudo experiments drawn from each mass ordering together with the respective Asimov value. Only θ_{23} and Δm^2_{31} were free in the fits, all other parameters (systematics) were fixed to their true values.

6.2.3. MCMC posterior sampling

As demonstrated in sec. 5.3.1, the function learned by the neural networks can be used for MCMC sampling of the posterior distribution (defined in sec. 3.1). Thus, contours for the parameters of the neutrino interaction model can be determined, including correlations between the different parameters. This can be used, for example, for pointing studies in IceCube similar to [139], which looks at higher energies (>1 TeV). Estimating the contours of the neutrino direction is more important for extragalactic than for atmospheric neutrinos. However, in this section, using atmospheric neutrinos as an example, it is shown that FreeDOM can be used for such studies. To transfer this to higher energies, an adapted training set would be required.

Figure 6.19 shows the sampled posterior distribution for an example DeepCore MC event.

The truth is inside the 68% contour for all parameter combinations. Expectable parameter correlations like z vertex position and zenith angle (vertical alignment) or x and y vertex position and azimuth angel (horizontal alignment) can be seen. The inelasticity is not constraint for this event because the event only has a true track energy of roughly 4 GeV, meaning that the track is too short to be resolved by the DeepCore sensor spacing. As a consequence, the angular 95% contour covers about 2 rad in azimuth and 1.8 rad in zenith.

Figure 6.20 shows the sampled posterior distribution for a real data event. Since the truth is unknown, RetroReco is shown as comparison. This time a higher energetic track, about 127 GeV according to RetroReco, was chosen.

The two reconstruction methods agree for this event, meaning that the RetroReco best-fit values are inside the FreeDOM 68% contours for nearly all parameter combinations. The angular contours for this event are much narrower than for the

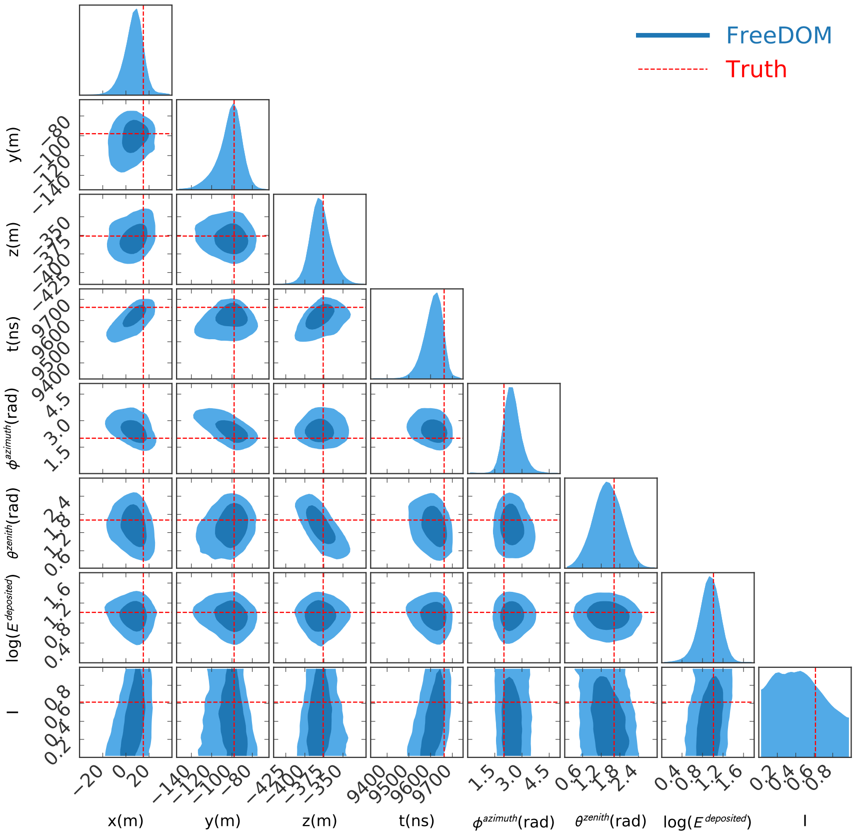


Figure 6.19.: MCMC posterior sampling for a DeepCore MC event. The 2D 68% and 95% contours are shown as well as the individual 1D distributions. The red lines mark the true parameter values. $1.65 \cdot 10^5$ samples are used.

first event. The 95% contours spans about 0.1 rad in zenith and 0.1 rad in azimuth. So the direction of a long track can be estimated with a higher confidence than for a short track. The total energy region expands to higher values, which can be explained by the fact that the track is not fully contained in the DeepCore detector. Thus, making the track even longer is only constraint by non-DeepCore strings which have a less dense spacing.

In summary, FreeDOM shows the expected behavior for longer tracks and parameter correlations. The sampled posterior distributions agree with the truth respectively RetroReco values. FreeDOM is a promising candidate for pointing or, in fact, any MCMC-based analysis.

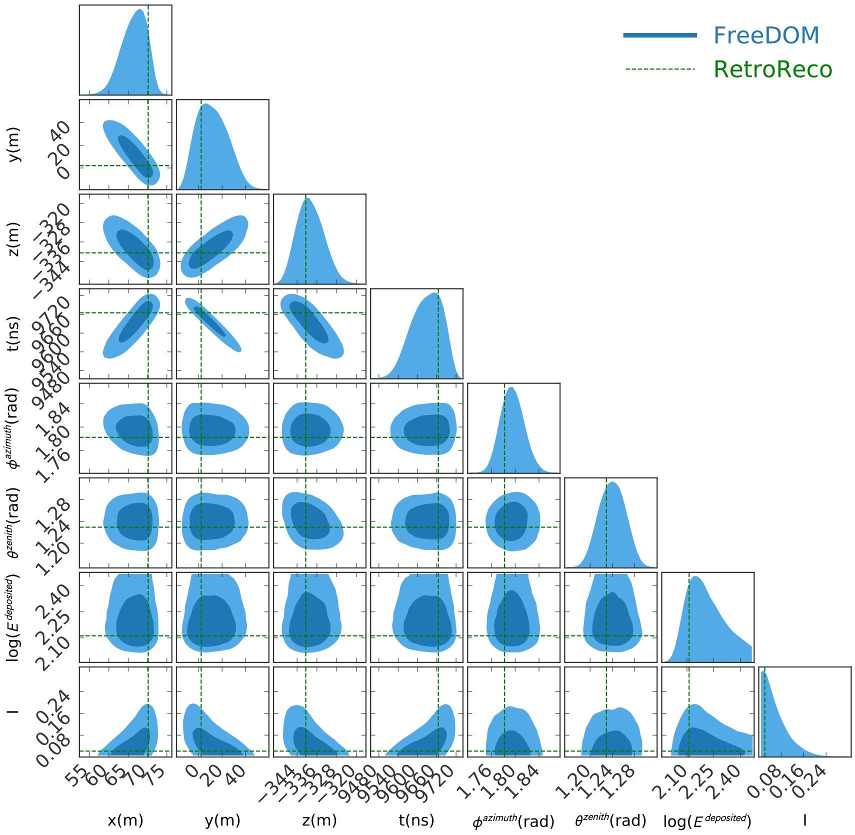


Figure 6.20.: MCMC posterior sampling for a DeepCore data event. The 2D 68% and 95% contours are shown as well as the individual 1D distributions. The green lines mark the RetroReco best-fit parameter values. $2.5 \cdot 10^5$ samples are used.

6.2.4. Estimating the influence of training sample size on sensitivities

As shown in sec. 6.1.1, even for a simple detector, huge amounts of training events are needed to approximate the true likelihood function to the point where the influence of the minimizer on the best-fit points is dominant.

The goal of this section is to estimate how the physics sensitivities of a DeepCore-like detector are affected by the number of training events. This is done in three steps. First, a DeepCore-like toy detector is used to study the relation between the training sample size and the KS values between the parameter resolution distributions achieved with the true and learned likelihood function. This is identical to the study shown in sec. 6.1.1. Then, the parameter resolutions of the oscNext sample are changed to obtain the relation between the KS values of the original and changed parameter resolution distributions and the relative change in sensitivity. Finally, the training sample size and the relative change in sensitivity are connected via the respective KS values.

Relation between KS values and training sample size from toy detector

A DeepCore-like toy detector is used to estimate the relation between KS values and training sample size. The general toy experiment setting is described in sec. A. The toy detector consist of 480 sensors located at the same positions as the HQE DOMs on the last 8 IceCube strings.
The KS value calculation is done in the same way as in sec. 6.1.1. For each number of training events, five different training sets have been simulated and used to train networks. These networks were used to reconstruct the always same set of test events. Thus, five KS values per parameter could be calculated for each number of training events. This time the KS value is not averaged over all parameters but the individual parameter results are used, averaged over the five training sets per number of training events.
Figure 6.21 shows the mean KS values for the total energy, cosine zenith, and inelasticity. The latter is used as PID proxy. The errorbar represents the standard deviation of the respective five KS values. The same fit function as in sec. 6.1.1 was used to extrapolate the data sets. The horizontal dashed lines are the individual KS values resulting from a second reconstruction using the true likelihood.

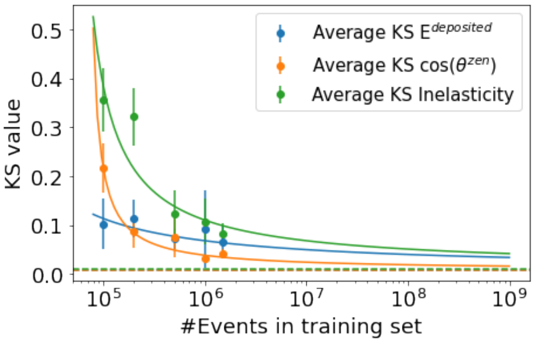


Figure 6.21.: KS values between the parameter resolution distributions obtained with the true and network learned likelihood function for a DeepCore-like toy detector depending on number of training events.

Based on the inverse functions of these three fits a conversion from KS values to number of training events can be made for each parameter.

Relation between KS values and sensitivity from scaled oscNext

The next step is to estimate how much a change in parameter resolution, represented by the KS value between the old and new parameter resolution distributions, affects the sensitivity of an analysis. The same oscNext events as in sec. 6.2.2 are used to calculate the sensitivities for the oscillation parameter and NMO determination. To

be able to calculate a relative change in sensitivity, the sensitivity to the oscillation parameters is quantified by the area of the 90% contour (as shown in fig. 6.17).
As *optimal* resolutions 15% improved RetroReco resolutions are used. The reason for using improved resolutions is to emulate a detector that is sensitive to both analyses presented before. With the actual resolutions the NMO sensitivity is to weak to allow conclusions about the quality of a reconstruction.
Now two different methods are used to change the resolutions. The new reconstructed parameter values p_{new} are calculated based on the *optimal* $p_{optimal}$ in the following way:

- $p_{new} = p_{optimal} + a \cdot (p_{true} - p_{optimal}) \cdot \mathcal{N}(1, 0.15)$: Similar to the estimation of the potential in sec. 4.4.4, the reconstructed values are moved by a certain fraction of their distance to the truth. Here the values are moved further away from the truth ($a < 0$). The KS value is increased by increasing the absolute value of a. $\mathcal{N}(1, 0.15)$ is a random Gaussian smearing of the parameter shift.
- $p_{new} = p_{optimal} + s \cdot \mathcal{N}(0, b)$: A simple Gaussian smearing of the reconstructed values. s is a scaling factor which is different for each parameter. For the energy $s = E_{true}^{deposited}$, for the cosine zenith $s = 2$, and for the PID $s = 1$. The KS value is increased by increasing the standard deviation of the Gaussian smearing b.

The first method assumes a correlation between the $p_{optimal}$ and new reconstruction, while the second only assumes that they will roughly find the same best-fit values.

To demonstrate the impact of the different parameters, the resolution of only one of them is changed each time. This would result in six different relations between KS value and relative change in sensitivity (three parameters times two methods to change the resolution). However, the first method is not applicable to the PID score, because at some point shifting away the classification score from the truth improves the classifier. Basically, a classifier that is always wrong is just as useful as a classifier that is always right. Therefore, fig. 6.22 only shows five curves. The curves labeled with an a used the first method to change the parameter resolution, while the curves labeled as b used the second. The vertical dotted lines show the KS value of the last point in fig. 6.21 for each parameter.
except for the green curve in the NMO plot, all curves show a almost linear relation between KS values and relative change in sensitivity. Therefore, a linear interpolation is used to convert KS values into a relative change in sensitivity.

Relation between training sample size and sensitivity

Now the number of training events and the relative change in sensitivity can be connected via their respective relation to the KS values. This is done with the fits shown in fig. 6.21 and the linear interpolations shown in fig. 6.22. Figure 6.23 shows the relative change in sensitivity depending on the number of training events. The vertical black line shows the number of events that were used to train the oscNext networks.

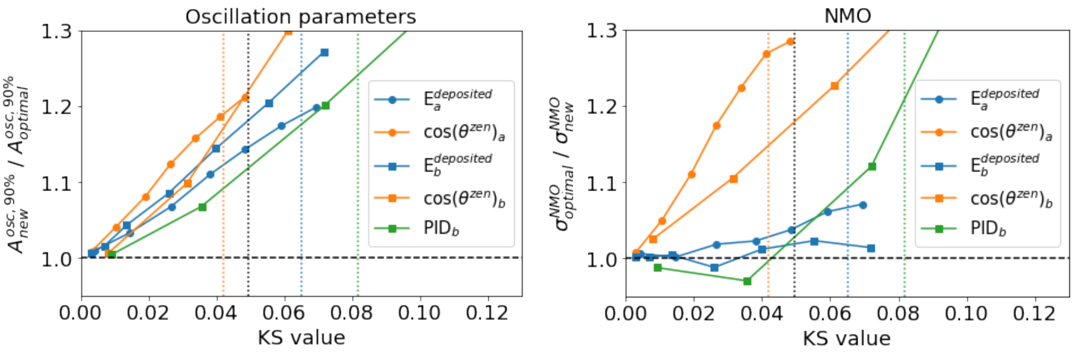


Figure 6.22.: Relation between the KS value of changed parameter resolution distributions and the relative change in sensitivity resulting from the changed resolutions.

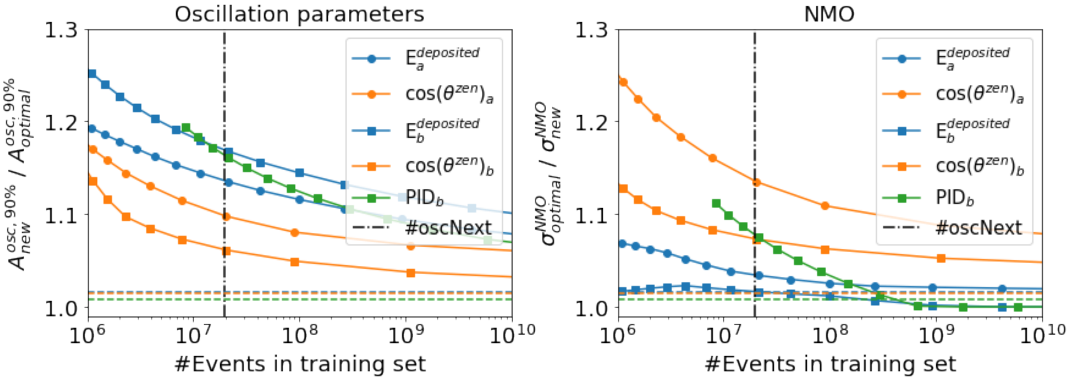


Figure 6.23.: Relative change in sensitivity to the oscillation parameters (left) and NMO (right) for different numbers of events in the training sample.

The plot suggests that for the oscillation parameters, several orders of magnitude more events than in the oscNext sample would be needed to approximate the likelihood to a level where there is no significant reduction in sensitivity. The analysis would benefit from a better modeling of all three parameters Also the NMO sensitivity would benefit from more events than there are in the oscNext sample. In contrast to the oscillation parameters, with one order of magnitude more data, energy and PID would be modeled well enough to not significantly reduce the NMO sensitivity.

Note that this study makes some assumptions, e.g. about the change in resolution, and should only be considered as rough estimation of the relation between training sample size and sensitivity. It is also not made for DeepCore but a DeepCore-like detector because the optimal resolution as well as the true likelihood for the real DeepCore detector are unknown.

6.3. IceCube Upgrade studies

There currently is no likelihood based reconstruction method for the IceCube Upgrade. The photon tables used for RetroReco make symmetry assumptions about the optical modules that are no longer valid for the mDOM and the D-Egg. Dropping these assumptions would result in prohibitively large memory and CPU requirements for the reconstruction. In addition, the reconstruction time would increase significantly. For reference, the reconstruction of the oscNext DeepCore MC sets with RetroReco takes more than two million CPU hours.

FreeDOM, on the other hand, requires almost no changes to work with Upgrade data because it is trained without such assumptions. The only change, described in sec. 5.2.3, is the use of a separate Hit- and ChargeNet for each module type. This makes FreeDOM a promising candidate for the future of IceCube low energy reconstruction, while RetroReco will not be developed further to work with Upgrade data.

It should be mentioned that there is also a Graph Neural Network (GNN) based reconstruction under development that shows promising performance on Upgrade simulation. However, the GNN can only estimate a single quantity (e.g. best-fit values, uncertainties, PID) at a time and does not provide a likelihood function.

Most studies presented in this section were performed with the noise free muon only Upgrade MC set described in sec. 5.2.3. Only the mDOM noise study section uses different MC sets specifically generated for the purpose of this study. These sets are described in sec. 6.3.2.

The Upgrade parameter resolutions in this section are not compared to another reconstruction, simply because there is none that estimates all eight parameters of the neutrino interaction model. Instead, they are compared to DeepCore only resolutions to demonstrate the benefit of the Upgrade. The DeepCore only resolutions are obtained in a second fit to the same MC events, which includes only DOMs and uses the DOM networks that are also part of the Upgrade fits. That is an advantage of having a separate pair of networks for each optical module type. However, making a fair comparison between DeepCore and the Upgrade is difficult. There are events with too few hits in DOMs to be reconstructed with DeepCore but enough hits in the new module types to be reconstructed with the Upgrade. Having these events in the test sample would decrease DeepCore's resolutions, but only including events that can be well reconstructed with DeepCore would underestimate the improvement through the Upgrade. For the following plots, events are used where the minimizer has successfully converged for both detectors and where there is at least one hit in the DeepCore detector. In addition, a paragraph was added to show the Upgrade performance on events that have to few hits in DeepCore to be selected for a DeepCore event sample.

Note that, unless otherwise specified, Upgrade always includes new strings and

DeepCore strings. Furthermore, the shown parameter resolutions are considered optimistic because no noise is present in the events. The presence of noise worsens the resolutions, especially for low energy events. So the focus should be on the relative improvement to the DeepCore resolutions.

6.3.1. Reconstruction performance

First, the reconstruction performance of the Upgarde (including DeepCore strings) is compared to the DeepCore only resolutions.
Figure 6.24 shows the one-dimensional best-fit distributions of the Upgrade and DeepCore compared to the true "injected" parameter distributions. The same selection of parameters was made as in fig. 6.6.

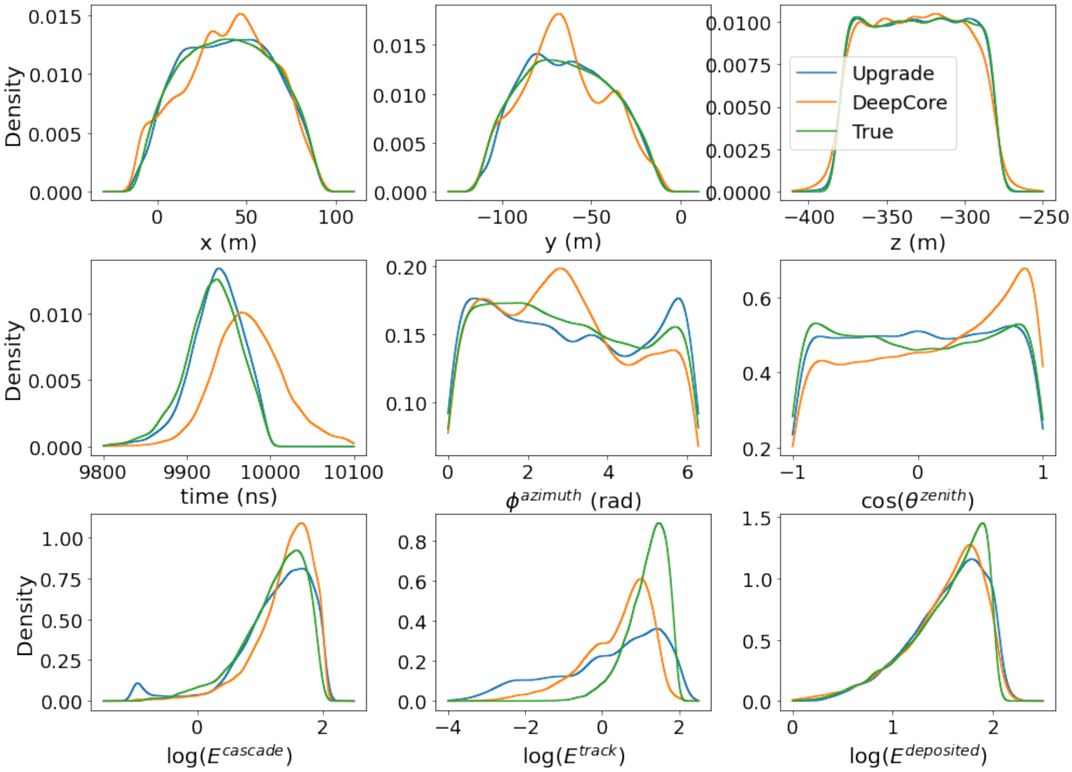


Figure 6.24.: Best fit distributions of the Upgrade (blue) and DeepCore only (orange) compared to true (injected) parameter distributions (green). All eight parameters of the neutrino interaction model are included together with the total deposited energy. KDEs of the distributions are shown.

The vertex distributions recovered by the Upgrade look similar to the true distributions, while DeepCore shows biases towards its string positions. Events that only illuminate a single DeepCore string are pulled towards this string in the reconstruction. This also explains the bias towards later times for DeepCore. If an event is

reconstructed closer to a string its reconstructed time is shifted later to produce the same hit times. The Upgrade time distribution does not show this bias.
DeepCore also shows some biases for the angles. For events that provide nearly no information about the azimuth, the likelihood should be flat. However, due to statistical fluctuations in the training set and process, the function learned by the networks is not perfectly flat. In the minimization process, the minimizer finds these artificial minima in the function. The trend towards down-going was also visible in fig. 6.6 but not as strong as here. The lower number of training events, compared to the oscNext set, leads to a worse approximation of the likelihood (see sec. 6.1.1) and could amplify biases.
The track energy shows a significant difference between the true and reconstructed distribution for both detectors. The toy experiments (sec. 6.1) showed that, especially for low numbers of training events, the division in track and cascade energy is the most difficult parameter to learn for the networks. The track energy is shifted to lower values while the cascade energy is biased towards higher energies. This leads to a stronger difference for the track energy in fig. 6.24 because of the logarithmic x-axis. The reason for the track energy to be biased towards lower energies is that it is possible to have no track but not to have no cascade. The reconstructed total energy distributions look more like the truth, which also indicates that the division in cascade and track energy was not learned well.

Timing

The Upgrade adds nearly ten thousand additional PMTs to the current instrumentation, which leads to more recorded hits per event. While this provides more information for the reconstructions, it also requires more likelihood evaluations since the likelihood is called once per hit. Thus, each minimizer iteration needs more time and the reconstruction takes longer.
For the tested MC sample, the median of the ratio of reconstruction times $median\left(\frac{t^{Upgrade}}{t^{DeepCore}}\right)$ and the median of the ratio of hits $median\left(\frac{N_{hits}^{Upgrade}}{N_{hits}^{DeepCore}}\right)$ are both ~7.47. Therefore, the difference in reconstruction times can be explained by the higher number of hits.
The mean Upgrade reconstruction time is 79.6 s. Taking into account the speedup from event paralellization on a GPU, as shown in sec. 6.2.1, this is still less than one second per event. So the Upgrade reconstruction with FreeDOM is much faster than the DeepCore reconstruction with RetroReco, which takes about 40 s. This will save computing resources and allows for either larger MC sets or more systematic sets.

Parameter resolutions

The important question for IceCube with respect to the Upgrade is how much the parameter resolutions will benefit from the additional photosensor instrumentation. To answer this question fig. 6.25 shows the reconstructed versus true values for all eight parameters of the neutrino interaction model plus the total energy.

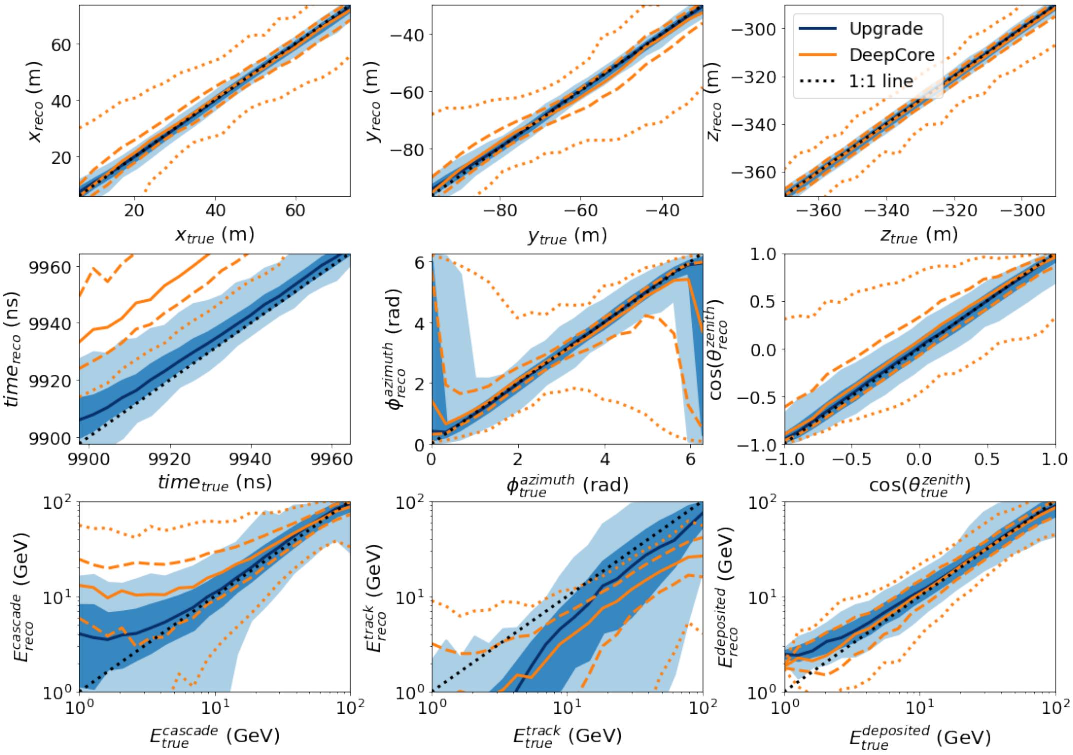


Figure 6.25.: Reconstructed versus true parameter values for using the Upgrade (blue surfaces) and DeepCore only (orange lines) hits of an event in the reconstruction. The median as well as the IQR and 90 percent inter quantile range are shown. In addition, the one to one line is indicated by the dotted black line.

For the vertex resolutions, the Upgrade 90 percent inter quantile ranges are comparable to the DeepCore IQRs. The more dense module spacing and different PMT orientations help locating the interaction vertex. The latter helps locating the relative position of the vertex with respect to the optical module. The time bias for the Upgrade is only a few ns, while it is more than 30 ns for DeepCore, which is similar to fig. 6.8. The time values are different compared to fig. 6.8 because the absolute time depends on the trigger time. The trigger here is based on all strings including the Upgrade ones.

Also the angular resolutions strongly benefit from the additional PMTs installed in the ice. Different PMT orientations, as arranged in the mDOM, also contribute to the improved directional resolutions. Even a single mDOM can provide information about the relative position and orientation of a light source to the module, as it has multiple PMTs. However, it should be mentioned that the directional resolution is likely to deteriorate in the presence of noise.

The energy biases, which could already be seen in fig. 6.24, are stronger for lower energies. This behavior can be seen for both detectors but the bias, represented by the distance from the median to the 1:1 line, is less pronounced for the Upgrade. Since a different track energy behavior was seen in fig. 6.8, it can be assumed that more training events change the picture here too. At low energies the total energy resolution of the Upgrade, defined by the inter quantile ranges, supersedes DeepCore while at higher energies they are equal. DeepCore's median is closer to the 1:1 line at low energies, but this does not mean it can better reconstruct low energy events. Many of these events only have very few hits in DeepCore and consequently are reconstructed at low energy values. This lowers the median but widens the inter quantile ranges which are a better indication of the actual parameter resolution. Note that the resolutions at low energies will change when including noise. More about that can be found in sec. 6.3.2.

The dependence of the resolutions on the true energy are of special interest here because it shows if the Upgrade really provides better resolutions at low energies. Figure 6.26 shows the parameter resolutions versus true energy. For all vertex parameters the true deposited energy is used. The angular resolutions, including the angle between reconstructed and true direction ($\Delta\Psi_{true}^{reco}$), are plotted versus the true track energy, because they are expected to be reconstructed better for longer tracks. Energy resolutions are shown versus their respective truth.
The Upgrade can locate the interaction vertex of events at deposited energies down to 1 GeV within less than 3 m IQR, while the Deepcore vertex resolution at 1 GeV is about 60 m. The DeepCore value is different than in sec. 6.2 because of the different event sample (selection process) used here. The Upgrade vertex resolution is better than DeepCore's throughout the entire tested energy range, but the strongest difference can be seen below 10 GeV.
The same applies to angular resolutions. The median $\Delta\Psi_{true}^{reco}$ for track energies around 100 GeV is 5.6° for DeepCore and 3.1° for the Upgrade, while for 1 GeV tracks it is 49° for DeepCore and 21° for the Upgrade.
The energy behavior was already seen in fig. 6.25.

One advantage of having a separate pair of networks for each type of optical module is that it is possible the look at their individual contributions to the overall likelihood. It is also possible to only use one module type in the reconstruction and get an impression of their individual contribution to the Upgrade performance. This is similar to the DeepCore only resolutions which represent the DOM contribution.
Figure 6.27 shows the total deposited energy and cosine zenith resolutions of reconstructions only using one optical module type respectively compared to the full Upgrade.
The overall energy resolution of mDOMs and D-Eggs are virtually the same and both are worse than the DeepCore (DOM) result. The reason for this is likely the higher number of DOMs in the detector and their more even distribution, as can be seen in sec. 4.3.2. But even though their individual energy resolutions are worse, in

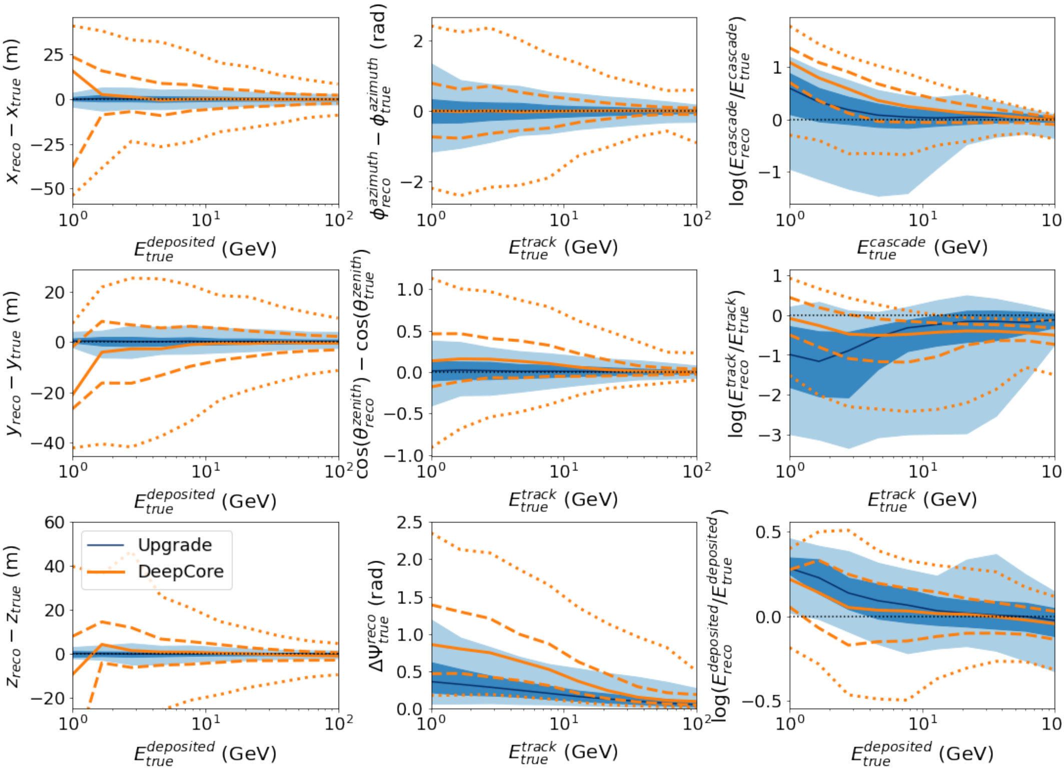


Figure 6.26.: Parameter resolutions versus true energy for the Upgrade (blue surfaces) and DeepCore only (orange lines). Depending on the parameter, different energies are used on the x-axis. The median as well as the IQR and 90 percent inter quantile ranges are shown.

combination with the DOMs the new modules improve the energy resolution of the detector.

The cosine zenith resolution shows the benefit of the different PMT orientations in the new modules for angular reconstruction. Both individual resolutions clearly exceed the DOM resolution. That different PMT orientations play an important role here can be seen from the fact that the mDOM (24 PMT orientations) zenith resolution is significantly better than the one for D-Eggs (2 PMT orientations). The Upgrade further improves upon the individual mDOM result.

Upgrade performance for non DeepCore reconstructable events

The Upgrade does not only provide more hits for events that could already be reconstructed with DeepCore, it also allows to reconstruct events that do not produce enough hits in DeepCore strings. This will lower the energy threshold of the detector and result in larger data samples.

Figure 6.28 shows the total deposited energy and cosine zenith resolutions for events

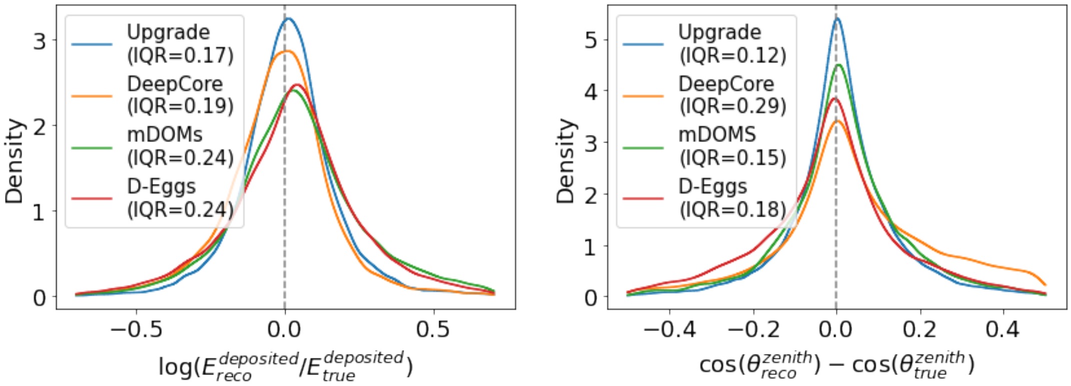


Figure 6.27.: Energy and zenith resolutions of reconstructions using only hits from one optical module type compared to the full Upgrade resolutions. DeepCore (orange) is identical to the DOM only resolutions. KDEs of the distributions are shown.

that can be reconstructed with both detectors and such that can only be reconstructed with the Upgrade. Events are considered as not reconstructable with DeepCore if they have less than eight hits in DOMs. For the oscNext event sample, this cut was used because there are eight parameters that need to be estimated.

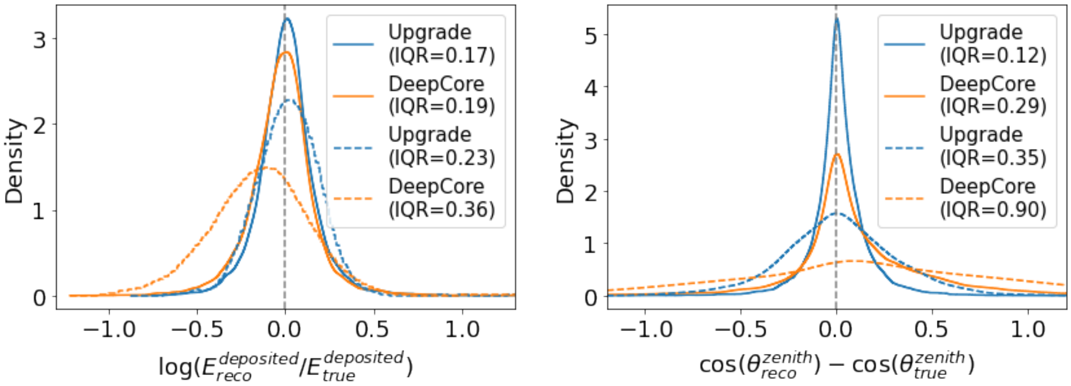


Figure 6.28.: Energy and zenith resolutions for events with less than eight hits in DOMs (dashed lines) compared to all events (solid lines). For each of the two cases Upgrade (blue) and DeepCore only (orange) resolutions are shown.

The dashed DeepCore lines confirm that these events can not be reconstructed using only DeepCore strings. The low number of DOM hits results in a strong underestimation of the energy and nearly no sensitivity to the zenith angle.
The Upgrade, on the other hand, can estimate energy and zenith of these events nearly unbiased and with much better resolutions. Both resolutions are worse compared to

the Upgrade resolutions including all events, because if there are fewer hits in DOMs, there is also less information available for the Upgrade. However, the important point is that having less than eight hits in DOMs is no longer a reason to remove an event from a data sample.

Particle IDentification (PID)

The Upgrade likelihood function can be used for particle identification in the IceCube low energy context in the same way as for DeepCore before.
Figure 6.29 shows PID ROC curves obtained with the full Upgrade likelihood and the DeepCore only part of it. A cut was made on the true deposited energy of the events. The solid lines only contain events with energies above 15 GeV, while the dashed lines are for low energetic events below 15 GeV. A 15 GeV track has a length of roughly 70 m, while the horizontal spacing of DeepCore strings is between 40 m and 90 m (see sec. 4.3.1).

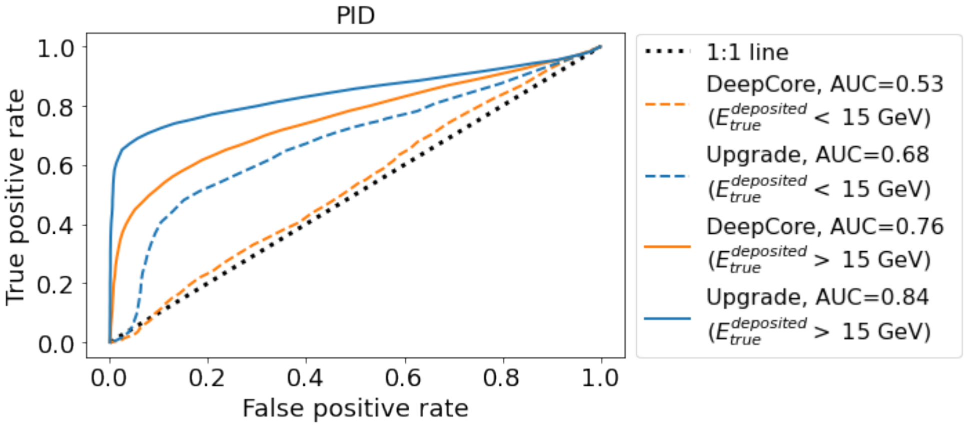


Figure 6.29.: The PID performance based on the Upgrade likelihood (blue) compared to the DeepCore only part (orange). A likelihood ratio is used as PID score. The solid lines only include high energetic events above 15 GeV, while the dashed lines only include low energy events below this value. The area under each curve is given in the legend. The black dots indicate the one to one line.

For low energetic events below 15 GeV, DeepCore has nearly no distinguishing power (AUC$\approx$ 0.5). These tracks are too short to be resolved by the DOM spacing. The Upgrade can distinguish tracks and cascades even at energies below 15 GeV. It basically lowers the PID threshold of the detector which will be important for example for the NMO sensitivity. At energies above 15 GeV, both detectors can separate tracks from cascades. However, the Upgrade AUC is about 10% better than DeepCore's. In addition, the Upgrade ROC curve reaches a true positive rate of 60% at a false positive rate of roughly 0. This allows for a pure ν_μ sample that contains about 60%

of the tracks, while less than 2% of the cascades.
Again, it has to be mentioned that including noise will worsen these performances.

6.3.2. mDOM noise study

As seen is the previous section, the mDOM is a important component of the IceCube Upgrade. More than half of the newly deployed modules will be mDOMs. A single mDOM hosts 24 PMTs so it has a higher noise rate (~ 2700 Hz specified, extrapolated from [140]) per module compared to other modules (e.g. ~ 560 Hz [79] for a DOM). Therefore, it is important to use PMTs with a noise rate as small as possible. In the production of these PMTs a new kiln was used. This lead to a higher radioactive contamination in the PMT glass than specified by the manufacturer and consequently to a higher noise rate. The PMT noise rate was increased by roughly a factor of 3 (from 111 Hz to 360 Hz [140]). Since this was a failure of the producing company, they agreed to either give a discount on the already delivered PMTs or replace them which however would take more than a year. The question that had to be answered now was how much the increased mDOM noise decreases the Upgrade parameter resolutions and if it would be acceptable to keep the high noise PMTs.
To answer this question two MC sets were produced. One with the specified noise rates and the other with the observed increased rates. Each of the sets consists of about two million ν_μ events. A noise cleaning based on a graph neural network [141] was applied to both sets. FreedDOM networks were trained on 90% of each set respectively. Then, the remaining 10% were reconstructed by the respective networks.

Figure 6.30 shows the relative change in resolution for the analysis variables for the new higher noise rate. The resolution is defined by the IQR of $\ln\left(\frac{E_{reco}}{E_{true}}\right)$ for the deposited energy and $\cos\left(\theta_{reco}^{zenith}\right) - \cos\left(\theta_{true}^{zenith}\right)$ for the zenith angle. An event selection based on the number of hits per event was applied, which removes the same fraction of noise only events in both sets. This leads to a selection of events with at least 10 hits for the low noise rate and at least 20 for the high rate after cleaning, where the first set contains about twice as many events as the second one.
For high energies the energy resolution is basically unchanged because higher energetic events deposit more light in the detector. This leads to more hits that are actually caused by arriving photons in the modules. For low energies, a higher fraction of hits are noise hits. Below 10 GeV the energy resolution significantly drops for the increased noise scenario. Events in that energy region become more and more noise dominated with the increased noise rates.
The angular resolution for short tracks is up to 20% worse with the higher noise rate, for the same reason as the energy resolution, but it becomes better for long tracks. Getting better angular resolutions with more noise sounds counter intuitive, even for long tracks. The main reason for this behavior is the applied event selection. For the higher noise rate simulation, only events with at least 20 hits are considered, while for the low noise rate simulation, at least 10 hits were required. Long tracks

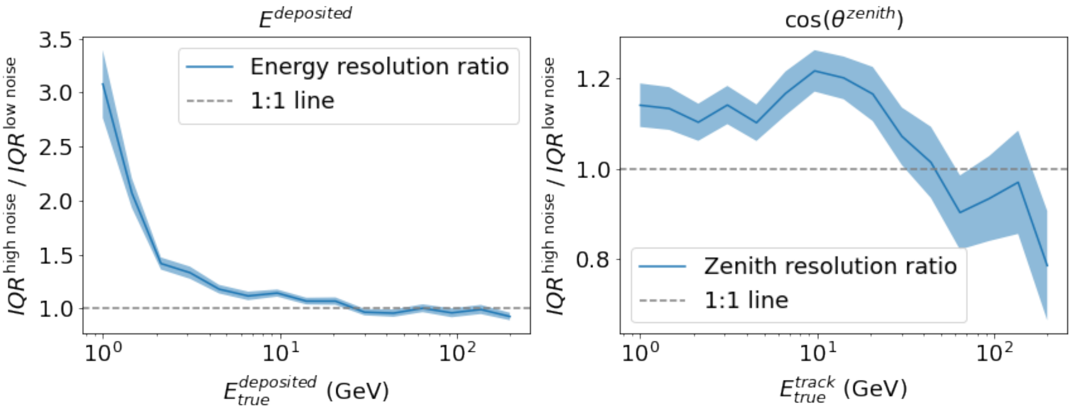


Figure 6.30.: Effect of the increased mDOM noise on energy and zenith resolution. A reconstruction of events simulated with a higher noise rate is compared with a reconstruction of events with a lower noise rate. The ratios of the resolution IQRs are shown versus the true deposited energy (energy) and true track energy (zenith) respectively. The error ranges are determined via one-sample bootstrapping. The grey dashed line represents the point where both reconstructions are equally good.

are rather elongated events which makes it easier for the noise cleaning to remove hits that do not align with the track. Having more hits in an event helps with the directional resolution. Considering only events with more than 20 hits consequently leads to better resolutions.

In summary, an increased noise rate degrades the energy and zenith resolution at a few GeV. For higher energetic events, starting at about 30 GeV, there is no difference for the energy and an improvement for the zenith angle.

7. Uncertainty estimations

A big advantage of knowing the likelihood landscape is that it is possible to calculate per event parameter uncertainties based on the width of the likelihood minimum[1]. This additional knowledge can be used to weight events in an analysis. Events with high uncertainties are expected to be reconstructed with larger residuums on average than events with low uncertainties. Assigning low weights to these low-quality events should reduce their influence on an analysis and consequently improve its sensitivity. In this chapter, the two physics analyses described in the previous chapter are repeated with weighted events. First, it is described how to extract per event parameter uncertainties from the reconstruction. Then, it is shown how these uncertainties are converted into event weights. Finally, the impact of these weights on the physics sensitivities is discussed.

7.1. Estimate per event parameter uncertainties

Uncertainties can be calculated for each parameter that affects the likelihood function in the minimization process, i.e. for each of the eight parameters which describe the neutrino interaction. However, since IceCube low energy analyses are binned in the three dimensions of total deposited energy, cosine of the zenith angle (see sec. 4.4.1), and particle identification (sec. 4.4.2), uncertainties on these parameters are considered to be of more interest than others.

There are multiple methods to estimate the width or curvature of a likelihood minimum, even without explicitly differentiating the function. The use of a simplex based optimization algorithm, such as crs2, enables the possibility to use the points visited during the minimization process. Such methods considered during this study include:

Likelihood weighted standard deviation

For all points with a $\Delta LLH <= 2$ to the minimum, calculate the standard deviation of the difference of their respective parameter value to the best-fit. Each point is weighted with $e^{-\Delta LLH}$ in this calculation.

Furthest point within $\Delta LLH = 2$

For all points with a $\Delta LLH <= 2$ to the minimum, return the largest absolute difference between the value for the respective parameter and its best-fit result.

Fitting a parabolic envelope

A parabolic fit to the minima in some slices of the parameter is performed. The size of a slice is defined by the standard deviation described above.

---

[1] Minimum of the negative log-likelihood

Area of convex hull
The area of the convex hull that includes all points with a $\Delta LLH <= 2$.

Figure C.4 in the appendix illustrates the different methods of uncertainty estimation using the z-vertex coordinate as an example.
The likelihood weighted standard deviation was chosen as uncertainty measure for two reasons. First, it never fails (never returns a NAN value), which is only true for the standard deviation and furthest point method. Second, it showed slightly higher correlations with the true absolute residuum, so the absolute difference between the reconstructed and the true value of a parameter, than the furthest point method.
Since it is possible to extract gradient information from neural networks, in principle the Fisher information, i.e. the curvature of the log likelihood function at the best-fit point, could be used to determine parameter uncertainties. However, unlike the other methods, this uncertainty estimate would be based on only a single point of the approximated function. The likelihood function is not approximated perfectly and methods based on multiple points are less affected by local fluctuations of the approximation.

Figure 7.1 shows the correlation between the uncertainty estimated based on the standard deviation of likelihood points and the true absolute residuum. It contains the same oscNext final level DeepCore events that are used in sec. 6.2. Ideally, the solid blue line would increase monotonically and the spread represented by the two dashed blue lines would be 0. The title of each subplot contains the Pearson product-moment correlation coefficient ρ [142] between uncertainty and residuum. A higher ρ value means that the uncertainty is a better estimation of the true residuum.
The $E^{deposited}$-E^{track} fraction parameterization of the neutrino interaction model parameters was used in the minimization. The uncertainty on the logarithm of the total deposited energy is calculated by:

$$\Delta \log\left(E_{reco}^{deposited}\right) = \log\left(\frac{E_{reco}^{deposited} + \Delta E_{reco}^{deposited}}{E_{reco}^{deposited}}\right). \tag{7.1}$$

It is noticeable that some events get estimated uncertainties of zero for nearly all parameters, but they are not perfectly reconstructed. Indeed, they are reconstructed worse than events with a small but nonzero estimated uncertainty. Events with estimated uncertainties of ~ 0 are events that were pulled to sit exactly at the bottom of the allowed range during the minimization. Figure C.5 in the appendix shows the z vertex position of these events. Usually, events that are fitted below the DeepCore volume (z< -500) are excluded from an analysis. So events with an estimated uncertainty of nearly zero for any parameter are thrown away.
Apart from that, for all parameters a positive correlation between uncertainty and residuum can be seen. Low estimated uncertainties really mean that the parameter is

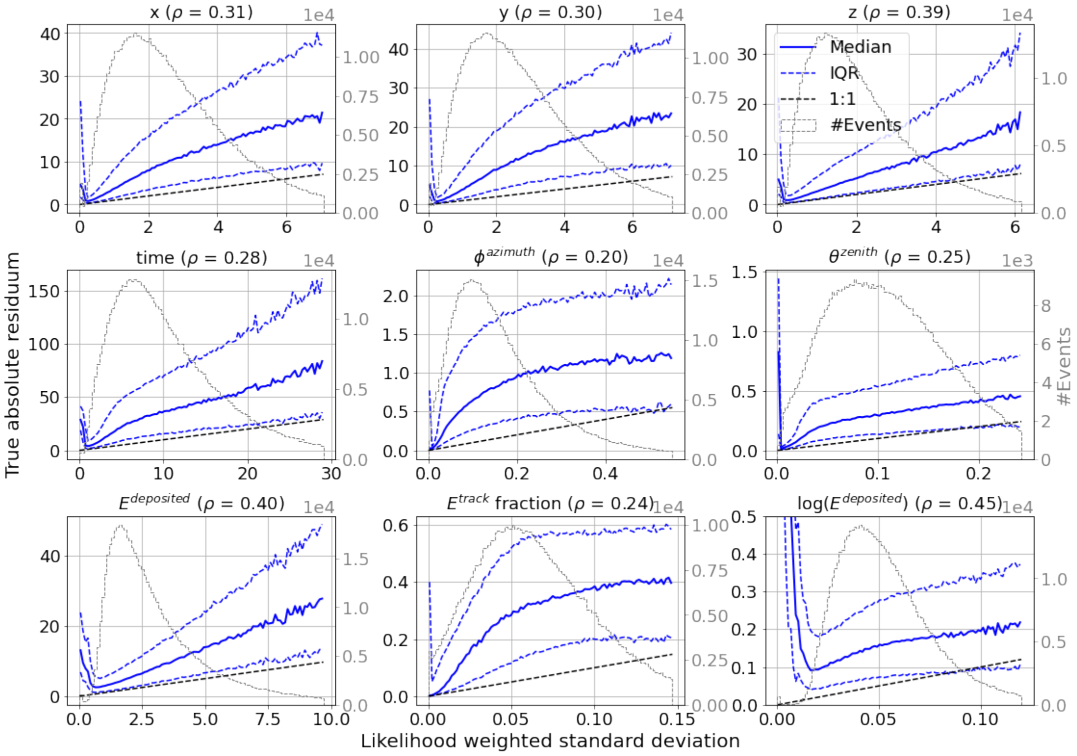


Figure 7.1.: Correlation between the per event estimated uncertainty and the true absolute residuum. The uncertainty estimation is based on the likelihood weighted standard deviation of the points visited during the minimization process. The solid blue line shows the median residuum for all events with the uncertainty given on the x-axis, the dashed blue lines the corresponding IQR. Parameter name and Pearson coefficient ρ are given in the titles. The grey histograms show the number of events with this estimated uncertainty and belong to the right y-axes.

likely reconstructed well, which is a requirement for using the uncertainties to weight events.

Data-MC agreement

Estimated uncertainties can only be used safely in an analysis if they describe real detector data in the same way that they describe MC simulation. Again it is not possible to compare the estimated uncertainty to the true residuum for real data events, but the distribution of the uncertainty estimations for a set of events can be compared.

In fig. 7.2 the distributions of the uncertainty estimations for the logarithm of the total deposited energy, the zenith angle, and the PID are shown. The same MC and data events are used as before. As PID uncertainty, the uncertainty on the track energy fraction (third row second column in fig. 7.1) is used.

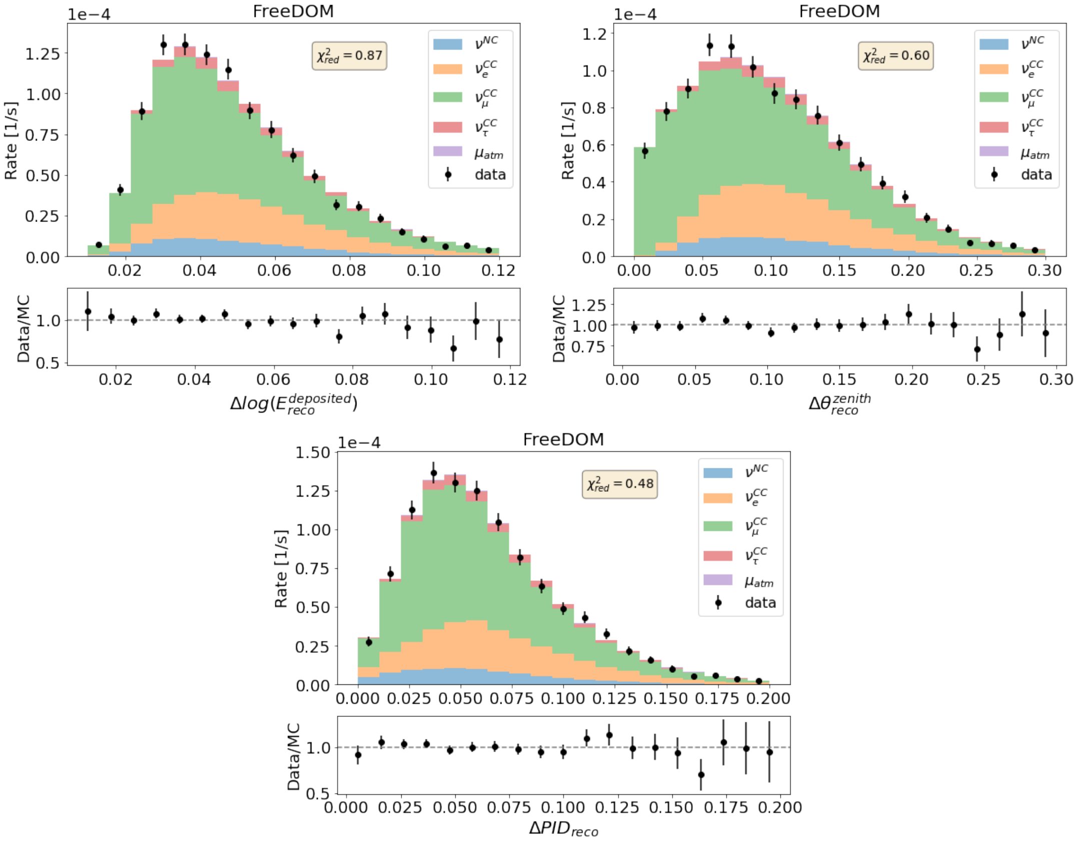


Figure 7.2.: Data-MC agreement for the logarithm of the total energy, zenith, and PID uncertainty estimations. The reduced χ^2 value for the data and MC expectations is given in each plot.

All χ^2_{red} values are below 1 which means data and MC agree within the uncertainty of the data points. So the estimated uncertainties can be used in an analysis. It is worth mentioning that the events with the lowest estimated uncertainty on the zenith angle are almost all ν_μ^{CC} events, so events containing a muon track. This makes sense because the direction of tracks can be reconstructed better as the direction of cascades (see fig. 6.11), especially if the tracks have a high energy (see fig. 6.10).

7.2. Weighting events

Once parameter uncertainties are determined for each event and parameter, they can be used to weight the events in an analysis histogram. In principle, there are infinite possibilities to turn the uncertainties into event weights. The optimal way depends on the analysis that should benefit from weighting the events and is not obvious without thorough testing.

The essential questions are what uncertainties should be used and how should the weight be calculated based on them. As mentioned in the previous section, typical IceCube low energy analyses are binned in the logarithm of the total deposited energy, cosine of the zenith angle and, PID. For the energy and the zenith angle, uncertainties can directly be extracted from the points visited during the minimization. For the PID, the uncertainty on the track energy fraction is used as a proxy. The three quantities are shown in fig. 7.2.

The only constrains to the function that calculates the event weights are that it should give higher values for smaller uncertainties and that it should only return positive weights. In addition, it can be required that a uncertainty of zero gives a weight of one. This suggests for example a function based on exponential functions of the following form for an event weight W:

$$W = e^{-a \cdot \Delta \log\left(E_{reco}^{deposited}\right)} \cdot e^{-b \cdot \Delta \theta_{reco}^{zenith}} \cdot e^{-c \cdot \Delta PID_{reco}} \, , \tag{7.2}$$

where a, b, and c are scaling factors adjusting the relative strength of the individual uncertainties on the weight. The scaling factors as well as the uncertainties always have positive values, so the weights calculated with function 7.2 are always between zero and one.

As mentioned earlier, finding the right values for the scaling factors depends on the analysis that should benefit from the event weighting. This is similar to cuts on events, which are also optimized for specific analyses. An analysis specific optimization of the weight calculation is needed.

Impact on resolutions

The main reason why the uncertainty weights described above can improve the sensitivity in different analyses is that they should effectively improve the parameter resolutions and PID classification performance.

Figure 7.3 shows the normalized resolutions to the analysis binning variables in IceCube low energy studies. It contains the unweighted FreeDOM and RetroReco results as well as two weighted FreeDOM results. For the first weight W_{unc} the estimated reconstruction uncertainties are used, while for the second weight W_{resi} the true residuum values are used instead of the uncertainties. The latter demonstrates the theoretical potential of weighting events for the analyses, because it shows what happens for an ideal uncertainty estimation. The two weight distributions are shown in the lower right plot. In this figure a, b, and c are chosen in a way that the weight distributions look similar, so the weights have similar magnitudes. This makes a almost fair comparison of the impact of two weights possible. For W_{unc} this results in $a, b, c = 3$ and for W_{resi} $a, b, c = 1$ was chosen arbitrarily. In the legend of the upper two plots the inter quantile ranges of the weighted distributions are given.

The weights improve FreeDOM's resolutions for all parameters important for an analysis. It was clear that this is the case for the residuum weights, but also the weights calculated from the estimated uncertainties give a boost in resolution. The gain in

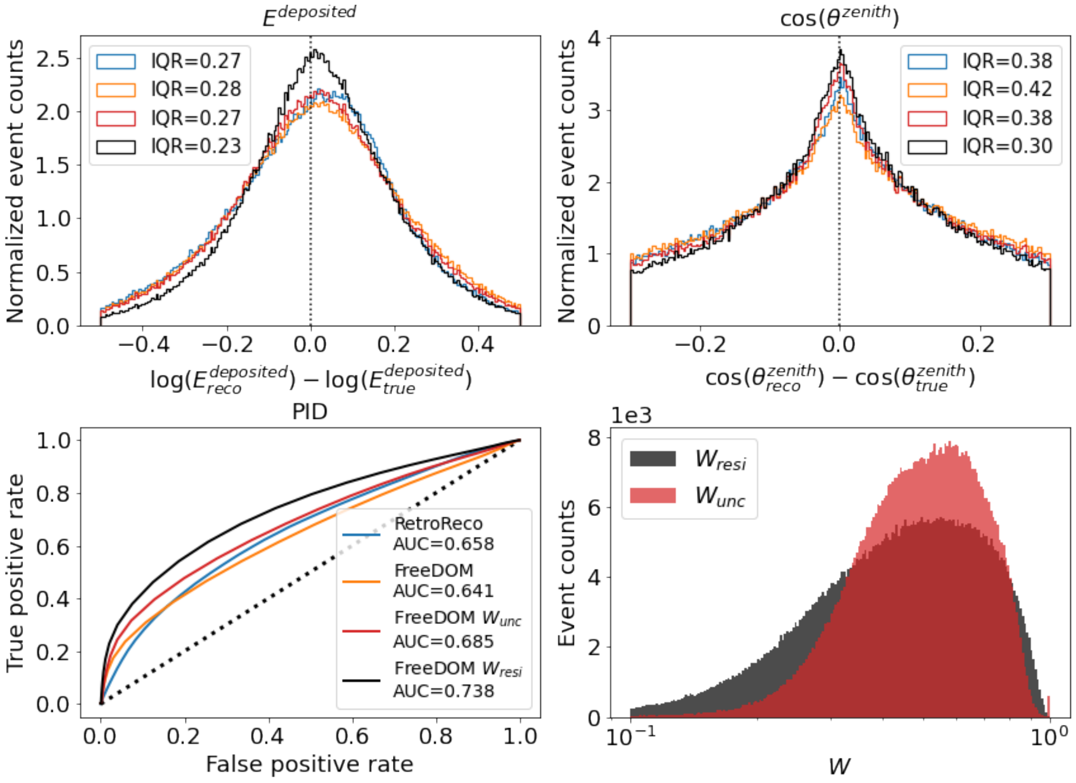


Figure 7.3.: Impact of weighting events on energy and zenith resolution as well as PID. The unweighted RetroReco (blue) and FreeDOM (orange) results are compared to FreeDOM weighted with the estimated uncertainty (red) and FreeDOM weighted with the true residuum (black). The last plot shows the distribution of weights for the two cases and chosen weight scaling factors mentioned in the text.

energy is about 5%, in terms of IQR reduction, and the gain in zenith ∼10%. The PID AUC is increased by about 7%.

It would be possible to get even better resolutions with more extreme weights, i.e. higher values of a, b, and/or c. However, this would result in only a few events with much higher weights than all others. These events would likely be reconstructed well, but an analysis has to take the fact into account that only a few events really contribute to a bin (in an event histogram). That means the used metric to calculate sensitivities has to be adjusted accordingly. This is done in the next section. Here it can be said in advance that extreme weight distributions, even though they yield better resolutions, lower the sensitivity to both analyses investigated in this thesis.

7.3. Impact of weights on sensitivities

Finally, the impact of the weights calculated based on the estimated uncertainties on the physics sensitivities discussed in the previous chapter can be examined. Atmospheric oscillation parameters and NMO sensitivity were calculated again using weighted events in the analysis histograms. To be able to do this the used metric has to be adjusted to the use of weights.

How to modify the χ^2 metric

The χ^2 metric defined in equ. 6.3 assumes that the bin counts represent a number of events. That is not the case anymore for weighted events. The number of expected events n_i in the denominator actually represents the square of the uncertainty of the bin count. As long as the bins simply contain numbers of events they are described by Poisson statistics and the uncertainty is the square root of the bin count. Consequently, the square of the uncertainty of the bin count is the bin count itself. However, in the case of weighted events the bin uncertainty changes to the square root of the sum of all weights in the bin squared [143]. So equ. 6.3 has to be rewritten to

$$\chi^2_{weight} = \sum_i^{Bins} \frac{(W_i - w_i)^2}{\Delta w_i^2 + \sigma_i^2}, \tag{7.3}$$

where W_i is the observed sum of weights of all events in bin i, w_i the expected sum of weights, Δw_i the uncertainty on the expected sum of weights and σ_i the MC uncertainty on the expected number of events in the bin weighted with the respective event weights of all events in the bin.
Each event basically has two different weights. The MC weight w_{MC} that represents a number of events and the uncertainty weight w_{unc} from a reconstruction which would be applied to real data in the same way. The difference between the (Poisson) MC uncertainty and the uncertainty of the sum of the bin weights is that $\Delta w_i = \sqrt{\sum_j w_{MC,j} \cdot w_{unc,j}^2}$ while $\sigma_i = \sqrt{\sum_j w_{MC,j}^2 \cdot w_{unc,j}^2}$. In case of no uncertainty weights ($w_{unc,j} = 1$) Δw_i^2 would just be $\sum_j w_{MC,j}$ which represents the number of events in the bin and is identical to the Poisson uncertainty (squared). So in that case χ^2_{weight} is identical to χ^2_{mod} defined in equ. 6.3.

Oscillation parameters

For the atmospheric oscillation parameters (θ_{23} and Δm^2_{31}), new sensitivity scans were made with weighted FreeDOM results. The same true values for θ_{23} and Δm^2_{31} as before were used. They are indicated as red cross in fig. 7.4. The analysis binning was not changed for any of the variables.
The left plot in the figure shows the 90% contours for RetroReco and FreeDOM with unweighted events. These are the same curves as shown in fig. 6.17. In addition, a FreeDOM curve made with events weighted by their estimated uncertainties and a

curve made with events weighted by their true residuum values are included. Both curves use values of $a, b, c = 5$ for the scaling factors introduced in equ. 7.2. Different values for the weight scaling factors have been tested but did not lead to significant improvements. However, it might still be possible that a specific combination of a, b, and, c is better suited here.
The right plot in fig. 7.4 shows sensitivity curves made with FreeDOM results weighted by the true residuum. The differences between the curves are the used weight scaling factors a, b, c. Only one value was used for all three factors to keep it simple. So the black and orange curves in the two plots are identical.

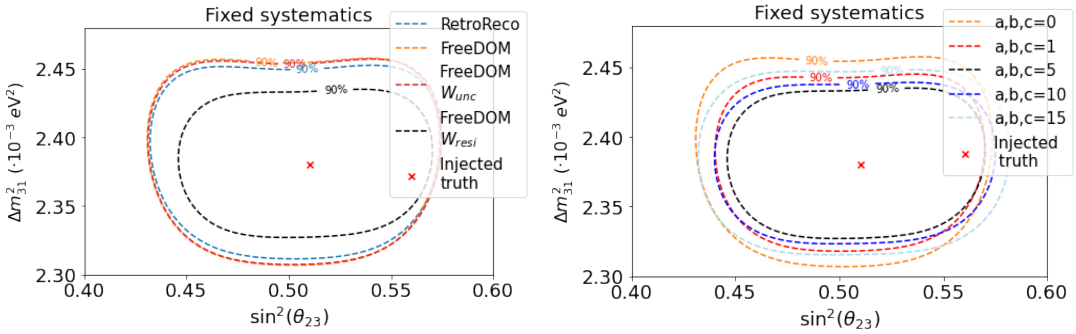


Figure 7.4.: Sensitivities to the atmospheric oscillation parameters θ_{23} and Δm^2_{31} using weighted event samples. In the left plot, the RetroReco (blue) and unweighted FreeDOM (orange) curve are identical to the ones in fig. 6.17. The red curve shows the FreeDOM result for events weighted by their estimated uncertainties, while the black curve uses true residuum weights. The right plot shows the results including residuum weights for different weight factors. Black and orange curves are identical in both plots.

For this specific choice of weight scaling factors, the area enclosed by the 90% contour line shrinks by only less than 1% using the estimated parameter uncertainties for the weights. The sensitivity to the oscillation parameters mostly stems from medium long tracks between 20 GeV and 50 GeV, which are not modeled well by the FreeDOM likelihood. That could explain why a weight calculated based on the shape of this likelihood does not improve the sensitivity much.
However, weighting with the true residuum shows the potential of this idea. A better estimation of the reconstruction error would lead to a significant improvement of the sensitivity to the atmospheric oscillation parameters θ_{23} and Δm^2_{31}.
As mentioned before, higher values of the weight scaling factors a, b, c lead to better resolutions but also to less events really contributing to the sensitivity calculation. Too high values of these scaling factors will consequently reduce the sensitivity. This can be seen in the right plot of fig. 7.4 (where for reasons of simplicity $a = b = c$). Increasing the weight scale up to a factor of $\sim$5 also increases the sensitivity to θ_{23} and Δm^2_{31}. For higher values of a the sensitivity decreases again.

NMO

Also for the neutrino mass ordering analysis new DeepCore sensitivities with weighted FreeDOM results were calculated. Again, the only thing that was changed compared to the results in sec. 6.2.2 is the introduction of event weights.
Figure 7.5 shows Asimov NMO sensitivities for RetroReco, FreeDOM, and FreeDOM weighted with its estimated uncertainties. Fit results where only θ_{23} and Δm^2_{31} were free in the fit, labeled as "Physics only", are shown as well as fits including systematic parameters, labeled as "Systematics". A table containing all systematic parameters can be found in the appendix (tab. C.1).
The left plot of the figure simply shows the calculated sensitivities for each of the cases. The weight scaling factors used here are $a = b = c = 12$. The impact of the scaling factors on the NMO sensitivity can be seen in the right plot of fig. 7.5. Again, for reasons of simplicity the same value was used for all three factors. It is given as a on the x-axis.

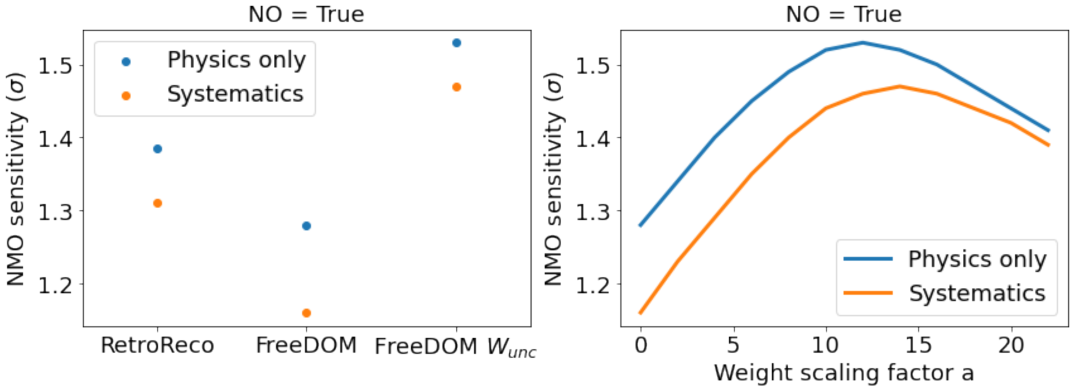


Figure 7.5.: NMO sensitivities using weighted event samples. Fits where only θ_{23} and Δm^2_{31} were free ("Physics only", blue) are shown as well as fits including systematic parameters ("Systematics", orange). In the left plot the unweighted RetroReco and FreeDOM results are compared to FreeDOM using events weighted by their estimated uncertainty. The right plot shows the sensitivity for the latter case and different weight scaling factors a.

In contrast to the oscillation parameter sensitivity, FreeDOM's NMO sensitivity can be improved using weights calculated from estimated parameter uncertainties. This is the case with and without including systematic parameters in the NMO fit. IceCube's NMO sensitivity comes from a lower energy region and not predominately from track-like events. So it is not surprising that the weighting has a different impact on this analysis. The weighted FreeDOM result supersedes the (unweighted) RetroReco result. In principle, it is possible to do the same weighting with RetroReco. So it could be that also this results can be improved further. However, this would require reconstructing all events again, since the information needed to calculate the weights was not stored for RetroReco.

Again, the right choice of the weight scaling factors is important. In this thesis, always the same value for all three scaling factors was used. A value of 12 appears to be the optimal choice for the “physics only” case, while the “systematics” fit yields the best results for a value of 14. For higher values of a (and equivalently b and c) the sensitivity decreases again. Using different values for a, b, and c might further increase the sensitivity.

In summary, weighting events with their parameter uncertainties is a promising way to increase the sensitivity of different analyses. For the NMO better results were achieved with the current choice of weights and quality of uncertainty estimation. With a better estimation of the parameter uncertainties also the sensitivity to the atmospheric oscillation parameters θ_{23} and Δm^2_{31} can be improved. There are many options which combination of parameter uncertainties should be used to calculate the event weights and what the individual weight factors should be. The best choice strongly depends on the analysis in question and the used likelihood function. It has to be studied for each analysis individually.

8. Summary and outlook

This thesis introduced a hybrid machine learning-likelihood method, called FreeDOM, which is capable of approximating the likelihood-to-evidence ratio of arbitrary configurations of individual photosensors. It is based on likelihood-free inference [3] and solely requires forward $\boldsymbol{\theta} \rightarrow \boldsymbol{x}$ MC simulation to train on. Therefore, it can even learn likelihood functions which are actually intractable. The method can be verified by comparing its predictions to an known truth, for toy detectors, or repeated MC simulation, for real detectors. In contrast to pure machine learning based methods, which act as point estimators, FreeDOM provides information about the entire likelihood and not only about its maximum value.

The presented work focused on a specific form of a decomposed likelihood [110]. It is designed for problems where the measurement consists of a variable length of observations and the number of observations itself also provides information. This is the case with a detector consisting of a number of individual photosensors. To tackle this type of problem, two artificial neural networks are trained. One of them, called HitNet, learns to use the information provided by single observations, which is mainly information about the propagation of photons. The other network, called ChargeNet, learns about the information provided by the number of observations, which is mainly information about the brightness of the light source. The combination of these two networks then yields the complete function.

Toy detectors show that the implementation of the method works for detectors consisting of an array of individual photosensors. It is very flexible and can be used for different detector configurations without major changes. Preliminary detector optimization studies can be run quickly to examine the influence of sensor arrangements on parameter resolutions. Moreover, not only was the minimum of the likelihood learned but also the likelihood landscape around it. Therefore, confidence levels or posterior distributions can also be determined.
However, it also became clear that a huge amount of training data is needed to approximate the true likelihood. Even for a simple detector, consisting of only five sensors aligned in a row, the likelihood could not be approximated to a level where the best-fit results of a reconstruction are dominated by the minimizer and not the likelihood approximation. Other network structures, changes in the training process, or the inclusion of more information about the underlying physics could reduce the number of events needed. This needs to be further investigated. Possible helpful physics information would be, for example, the velocity, lifetime, and lightyield of muons. However, care should be taken not to oversimplify the problem or provide

too much information. Finally, the true likelihood should be learned, not a simplified version.

The application of FreeDOM to DeepCore MC demonstrates that it also works for a realistic detector simulation. The achieved parameter resolutions are comparable to the state of the art likelihood-based reconstruction RetroReco, but FreeDOM is about 100 times faster. The learned likelihood is stable to changes in systematic parameters and applicable to real data. In addition, it can be used to determine a particle identification score by calculating the likelihood ratio of a track hypothesis and a no track hypothesis. Similar to the parameter resolutions, the sensitivities obtained with FreeDOM are comparable, but slightly worse than with RetroReco.
The fact that better sensitivities can be obtained with RetroReco shows that FreeDOM has not approximated the correct likelihood to the point where it is superior to a simplified, human-constructed function. The most probable explanation why this is not the case is a lack of training data. About 20 million events were used in the training. For an eight dimensional problem this results in a bit more than 8 points per dimension, which is low given the complexity of the problem.

FreeDOM is the first likelihood-based reconstruction successfully used for the IceCube Upgrade. Reconstructing Upgrade events poses challenges for methods building a likelihood with photon tables. The new optical module types developed for the Upgrade violate symmetry assumptions made by the photon tables. Dropping these assumptions would lead to prohibitively large memory and CPU requirements. FreeDOM on the other hand only needs MC simulation including the new module types to approximate the likelihood function. Likelihood scans as well as parameter resolutions showed that the additional information provided by the new module types could be used by the reconstruction. Other machine learning techniques can also be used to reconstruct Upgrade events, but they do not provide a Upgrade likelihood function.
In addition, FreeDOM also proved useful for the detector development process. It can estimate the effects of detector changes, such as an increased noise rate, on the reconstruction performance. If further maintained, FreeDOM has the potential to play an important role in the Upgrade and any future IceCube low energy reconstruction.
It should be noted that most of the presented Upgrade results are preliminary obtained with a noise-free simulation. Further studies with updated simulation are needed to determine FreeDOM's true potential for the Upgrade.

Access to the likelihood landscape makes it possible to estimate per event parameter uncertainties, which can be used to weight individual events in an analysis and improve its sensitivity. Using the true residuum shows that the potential gain in sensitivity can be high for a perfect estimation of the uncertainty. FreeDOM can estimate per event parameter uncertainties based on the simplex minimizer points visited. The estimated uncertainties correlate with the true residuum and show

a good data-MC agreement. They do not significantly improve the sensitivity to the atmospheric oscillation parameters, but they do improve the sensitivity to the neutrino mass ordering.
Since only sensitivity studies were conducted in this thesis, the impact of uncertainty weighting has yet to be tested in a full analysis.

There are many possibilities for future applications of FreeDOM. Firstly, it will be developed further alongside the IceCube Upgrade to reconstruct Upgrade events once the detector is taking data. Furthermore, machine learning reconstructions are very fast, so it is computationally feasible to run several of them on the same data and combine their results. FreeDOM can make an important contribution to this, as it differs from other reconstructions based purely on machine learning. Initial internal studies already indicated the potential of such a combination. Figure 8.1 shows energy and zenith resolutions of three methods involving machine learning, namely FreeDOM, a CNN-based method [144], and a GNN-based method [145], compared to RetroReco. In addition, the resolutions of two fully connected neural networks that get the best fit values of the three machine learning methods as input are included. These fully connected networks are called GAUNNTLET (Grand Algorithm Unifying Neural Networks To Leverage Effective Traits).

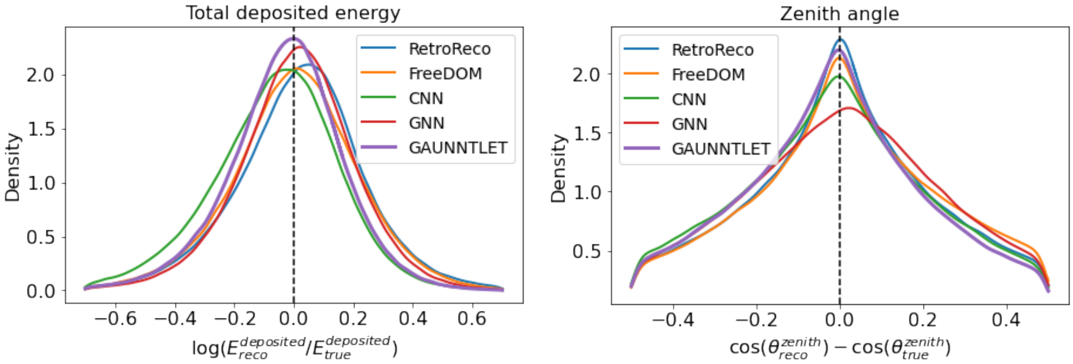


Figure 8.1.: Energy and zenith resolutions for different machine learning reconstructions and their combination (GAUNNTLET) compared to RetroReco.

The combined GAUNNTLET resolutions supersede the individual resolutions of the three included methods. The energy resolution is also better than RetroReco's. Although three reconstructions and one additional network evaluation must be performed, getting a GAUNNTLET result is much faster than RetroReco. RetroReco needs $\mathcal{O}(10\,\mathrm{s})$ per event for a reconstruction, FreeDOM $\mathcal{O}(100\,\mathrm{ms})$ and everything else $\mathcal{O}(\mu\mathrm{s})$.
It should be noted that improved versions of all three machine learning methods have been developed since this study was conducted and that the GAUNNTLET networks were just a initial proof of concept and not strongly optimized.

The application of FreeDOM is not only limited to the low energy part of IceCube. In principle, the method is also applicable to the reconstruction of TeV or higher energetic events in IceCube. MCMC sampling, for example, can be used in pointing analyses to estimate parameter contours for the neutrino direction. However, the way the information about the measurement is passed to FreeDOM needs to be modified to efficiently handle large numbers of hits. Summary statistics, like time of the first hit or median hit time in a DOM, have to be used instead of feeding individual hits to lower the computational effort.

Finally, the presented method can also be used outside IceCube. It has already been tested for the EOS detector [146] and helps with detector development, e.g. how high the concentration of liquid scintillator should be. As shown using the different toy configurations, all detectors consisting of individual photosensors can use FreeDOM without major changes. Moreover, the likelihood-free inference method itself can be transferred to other detectors as well. As long as a sufficient amount of simulation is available, the method can be used to learn any likelihood function.

A. Toy experiment setup

To verify the validity of the likelihood-free method and to demonstrate some of its use-cases, a toy model was developed. It was aimed to be as realistic as possible, but one should still be able to analytically write down the true likelihood function. This allows to directly compare the function learned by the neutral networks to the truth. Furthermore, the achieved parameter resolutions can be compared to the optimal ones. The toy model makes it possible to test different detector configurations.

The detector response implements a segmented interaction model with parameterized time and distance functions. The event model itself consists of the same eight parameters as used for the real IceCube detector (described in sec. 4.4.1). The cascade part is modeled as an isotropic light emission with 12819 Cherenkov photons per GeV [84]. The track part is modeled as a line of cascade emissions placed every 1 m along the track, to a total length of 4.5 m/GeV, and a constant 2451 photons of Cherenkov light emitted each meter [84].

The expected amount of photons arriving at a sensor scales with $1/r^2$ geometrically, plus an additional absorption of e^{-r/λ_a}, where the absorption length was set to $\lambda_a = 100\,\mathrm{m}$ (see fig. 4.2). The actual number of photons is drawn from a Poisson distribution. The time arrival distribution at sensors is modeled via a convolved Pandel function [147], for which the absorption length was set to the same value $\lambda_a = 100\,\mathrm{m}$ and the scattering length to $\lambda_s = 30\,\mathrm{m}$ (also see fig. 4.2). The refractive index of the surrounding medium was assumed to be $n = 1.3$ (ice, [30]). The Gaussian smearing of the convolution was set to 10 ns, representing the non-perfect time resolution of the modules.

Note that different detector configurations (sensor alignments) are used in this thesis.

B. Likelihood formulation rearrangement

In this chapter the rearrangement of the *per-sensor* formulation of the decomposed likelihood, shown in equ. 5.2, that leads to the *all-sensor* formulation, shown in equ. 5.4, is presented in more detail.

$$
\begin{aligned}
L(\boldsymbol{\theta}|\boldsymbol{x}) &= \prod_{s=1}^{N_{\mathrm{sens}}} \left[\prod_{i=1}^{N_s} p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta}) \right] \mathcal{N}_s(\boldsymbol{\theta})^{N_s} e^{-\mathcal{N}_s(\boldsymbol{\theta})} \\
&= \prod_{s=1}^{N_{\mathrm{sens}}} \left[\prod_{i=1}^{N_s} \mathcal{N}_s(\boldsymbol{\theta}) p_s(\boldsymbol{x}_{i,s}|\boldsymbol{\theta}) \right] e^{-\mathcal{N}_s(\boldsymbol{\theta})} \\
&= \left[\prod_{i=1}^{\sum_{s=1}^{N_{sens}} N_s} \mathcal{N}_{s_i}(\boldsymbol{\theta}) p_{s_i}(\boldsymbol{x}_i|\boldsymbol{\theta}) \right] e^{-\sum_{s=1}^{N_{sens}} \mathcal{N}_s} \\
&= \left[\prod_{i=1}^{N_{\mathrm{tot}}} \mathcal{N}_{s_i}(\boldsymbol{\theta}) p_{s_i}(\boldsymbol{x}_i|\boldsymbol{\theta}) \right] \frac{\mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})^{N_{\mathrm{tot}}}}{\mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})^{N_{\mathrm{tot}}}} e^{-\mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})} \\
&= \left[\prod_{i=1}^{N_{\mathrm{tot}}} \frac{\mathcal{N}_{s_i}(\boldsymbol{\theta})}{\mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})} p_{s_i}(\boldsymbol{x}_i|\boldsymbol{\theta}) \right] \mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})^{N_{\mathrm{tot}}} e^{-\mathcal{N}_{\mathrm{tot}}(\boldsymbol{\theta})}
\end{aligned}
\tag{B.1}
$$

C. Supplementary material

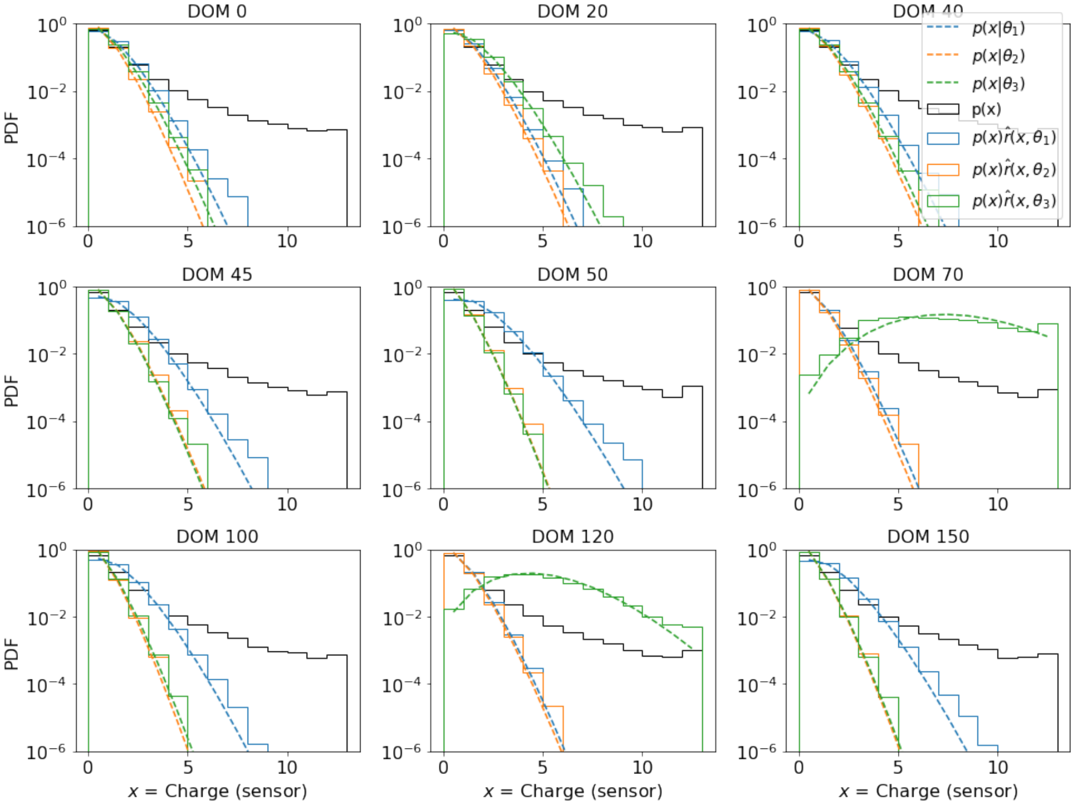


Figure C.1.: Likelihood verification based on event reweighting. The black distribution shows $p(x)$ for individual sensors. The dashed lines represent the respective charge PDFs for three example events. The colored distributions are obtained by reweighting the black distributions according to the respective event parameter values.

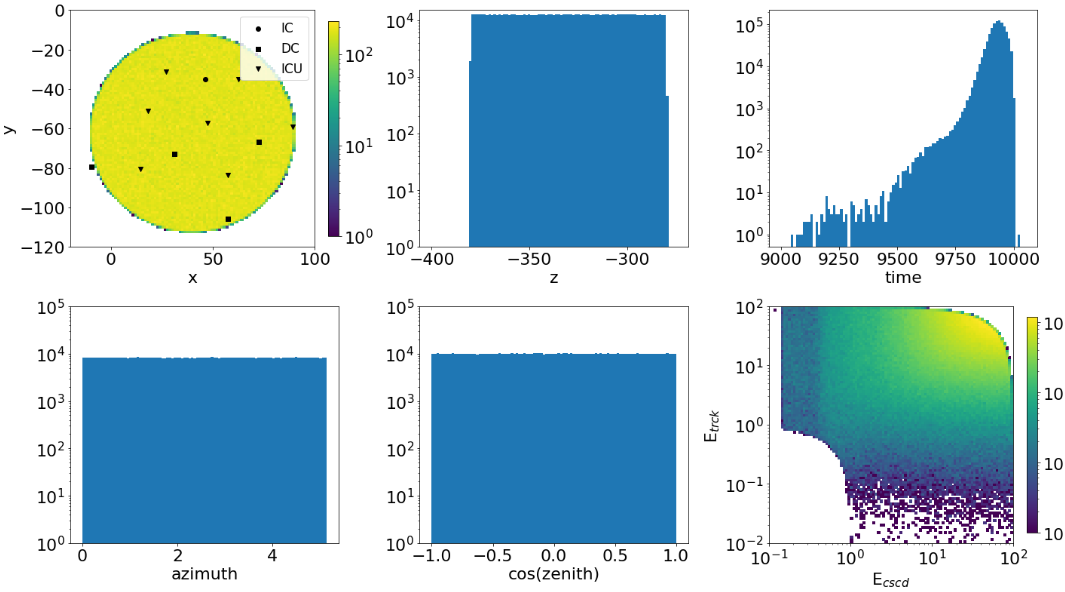


Figure C.2.: True parameter distributions for the noise free muon only MC simulation that is used to test FreeDOMs Upgrade performance.

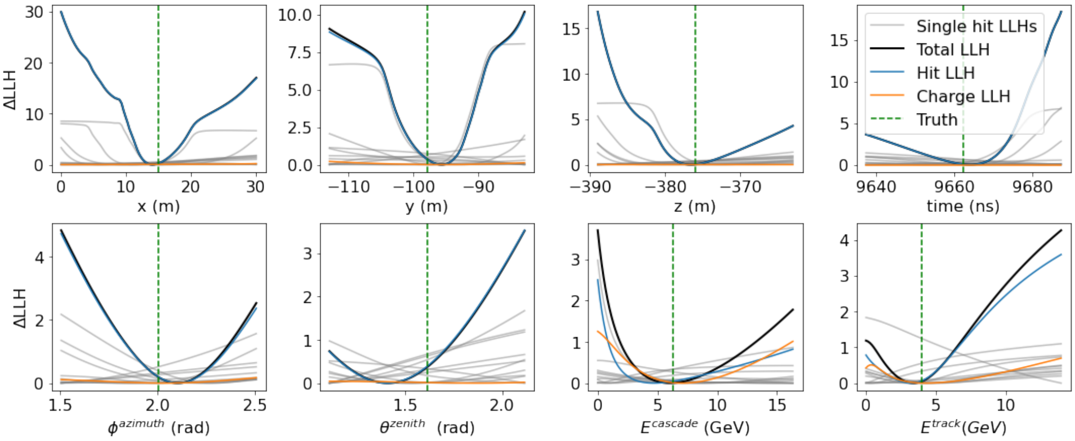


Figure C.3.: Likelihood scans for an example DeepCore MC event. The *all-sensor* formulation of the likelihood is used.

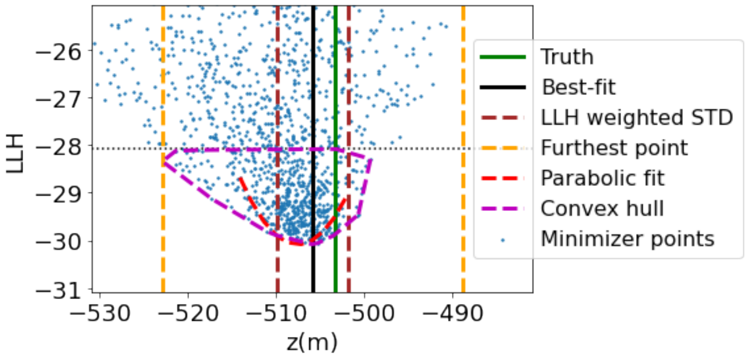


Figure C.4.: All tested parameter uncertainty estimation methods demonstrated at the example of the z vertex coordinate.

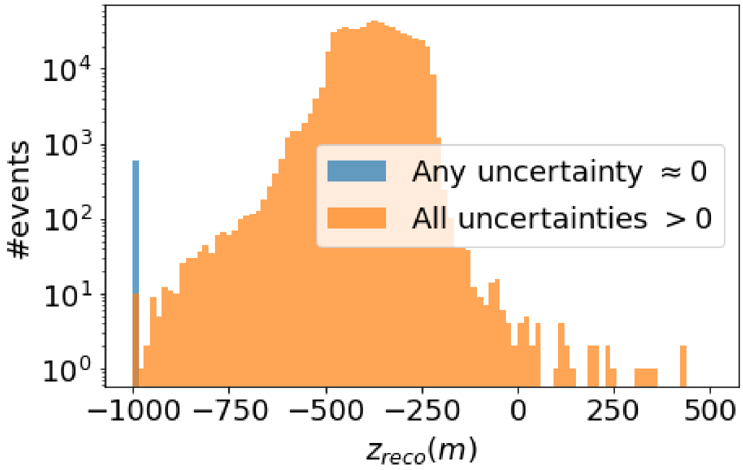


Figure C.5.: The best-fit z vertex position of events that got at least one estimated parameter uncertainty of ∼0 (blue) compared to all other events (orange).

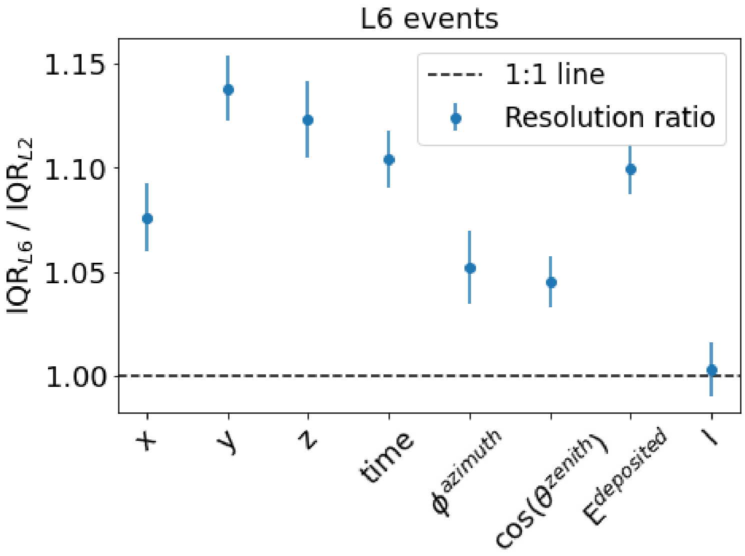


Figure C.6.: Parameter resolution ratios for networks trained on L6 events and networks trained on L2 events. L6 events were used in the reconstruction. The included uncertainties were obtained by one-sample bootstrapping and only include statistical uncertainties from the reconstructed event set, no possible uncertainties from the network training.

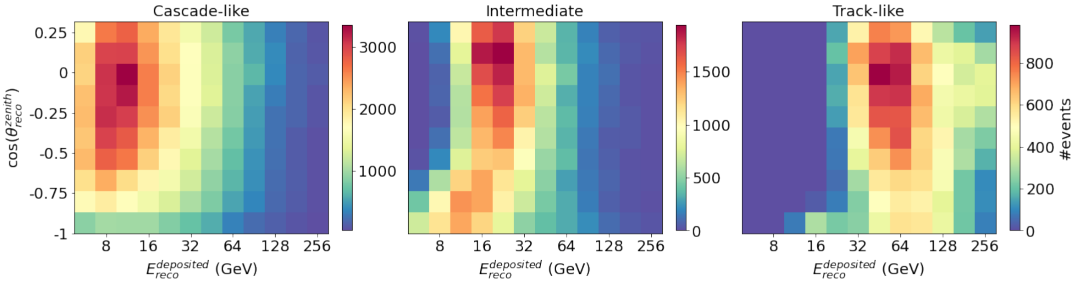


Figure C.7.: Example event histogram used in IceCube atmospheric neutrino oscillation analyses. Twelve logarithmic bins from 5 GeV to 300 GeV in energy, ten linear bins from -1 to 0.3 in cosine zenith, and three PID bins are used. oscNext final selection level MC events are shown reconstructed with RetreReco.

Systematics	
$\frac{\nu_e}{\nu_\mu}$ ratio	Covers uncertainity on ratio of atmospheric electron and muon neutrino production.
Barr Up/Hor ratio	Covers uncertainity on ratio of atmospheric neutrinos produced at the Poles (Up) and close to the horizon (Hor). Parameterization based on [148].
Barr $\frac{\nu}{\bar{\nu}}$	Covers uncertainity on ratio of atmospheric neutrino and anti-neutrino production. Parameterization based on [148].
Atmospheric index	Covers uncertainty on the energy spectrum of the atmospheric neutrino flux. Simply changes assumed power law by its value.
DIS CSMS	Covers uncertainty on deep inelastic scattering cross section. Based on [149].
A_{eff} scale	Effective area scale. Scales number of events in all analysis bins.

Table C.1.: Systematic parameters used in the NMO analysis.

Bibliography

[1] I. Hacking, *Logic of Statistical Inference.*
Cambridge, England: Cambridge University Press, 1965.

[2] J. Lundberg, P. Miočinović, K. Woschnagg, T. Burgess, J. Adams, S. Hundertmark, P. Desiati, and P. Niessen, “Light tracking through ice and water—scattering and absorption in heterogeneous media with photonics,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 581, pp. 619–631, nov 2007.

[3] J. Hermans, V. Begy, and G. Louppe, “Likelihood-free mcmc with amortized approximate ratio estimators,” 2020.

[4] “Offener brief an die gruppe der radioaktiven bei der gauvereins-tagung zu tübingen.” `https://cds.cern.ch/record/83282/files/meitner_0393.pdf`.
Accessed: 2022-03-17.

[5] E. Fermi, “Tentativo di una teoria dell’emissione dei raggi beta,” *Ric. Sci.*, vol. 4, pp. 491–495, 1933.

[6] C. L. Cowan, F. Reines, F. B. Harrison, H. W. Kruse, and A. D. McGuire, “Detection of the free neutrino: A Confirmation,” *Science*, vol. 124, pp. 103–104, 1956.

[7] G. Danby, J.-M. Gaillard, K. Goulianos, L. M. Lederman, N. Mistry, M. Schwartz, and J. Steinberger, “Observation of high-energy neutrino reactions and the existence of two kinds of neutrinos,” *Phys. Rev. Lett.*, vol. 9, pp. 36–44, Jul 1962.

[8] K. Kodama *et al.*, “Observation of tau neutrino interactions,” *Physics Letters B*, vol. 504, pp. 218–224, apr 2001.

[9] R. Davis, D. S. Harmer, and K. C. Hoffman, “Search for neutrinos from the sun,” *Phys. Rev. Lett.*, vol. 20, pp. 1205–1209, May 1968.

[10] T. Kajita, E. Kearns, and M. Shiozawa, “Establishing atmospheric neutrino oscillations with super-kamiokande,” *Nuclear Physics B*, vol. 908, pp. 14–29, 2016.
Neutrino Oscillations: Celebrating the Nobel Prize in Physics 2015.

[11] ALEPH Collaboration, DELPHI Collaboration, L3 Collaboration, OPAL Collaboration, SLD Collaboration, LEP ElectroweakWorking Group, and SLD Electroweak and Heavy Flavour Groups, “Precision electroweak measure-

ments on the z resonance," *Physics Reports*, vol. 427, no. 5, pp. 257–454, 2006.

[12] J. A. Formaggio and G. P. Zeller, "From ev to eev: Neutrino cross sections across energy scales," *Reviews of Modern Physics*, vol. 84, p. 1307–1341, Sep 2012.

[13] L. de Broglie, "Waves and Quanta," *Nature*, vol. 112, p. 540, Oct. 1923.

[14] K. G. Wilson, "Confinement of quarks," *Phys. Rev. D*, vol. 10, pp. 2445–2459, Oct 1974.

[15] R. P. Feynman, "Space-time approach to quantum electrodynamics," *Phys. Rev.*, vol. 76, pp. 769–789, Sep 1949.

[16] Z. Maki, M. Nakagawa, and S. Sakata, "Remarks on the Unified Model of Elementary Particles," *Progress of Theoretical Physics*, vol. 28, pp. 870–880, 11 1962.

[17] T. GERSHON, "Overview of the cabibbo–kobayashi–maskawa matrix†," *Pramana*, vol. 79, pp. 1091–1108, nov 2012.

[18] S. F. King, "Discrete symmetries and models of flavour mixing," *Journal of Physics: Conference Series*, vol. 631, p. 012005, jul 2015.

[19] C. Giganti, S. Lavignac, and M. Zito, "Neutrino oscillations: The rise of the PMNS paradigm," *Progress in Particle and Nuclear Physics*, vol. 98, pp. 1–54, jan 2018.

[20] M. Aker *et al.*, "Direct neutrino-mass measurement with sub-electronvolt sensitivity," *Nature Phys.*, vol. 18, no. 2, pp. 160–166, 2022.

[21] B. Aharmim *et al.*, "Combined analysis of all three phases of solar neutrino data from the sudbury neutrino observatory," *Physical Review C*, vol. 88, Aug 2013.

[22] T. Gaisser, *Cosmic Rays and Particle Physics.*
Cambridge University Press, 1991.

[23] M. Honda, M. Sajjad Athar, T. Kajita, K. Kasahara, and S. Midorikawa, "Atmospheric neutrino flux calculation using the NRLMSISE-00 atmospheric model," *Phys. Rev. D*, vol. 92, no. 2, p. 023004, 2015.

[24] K. A. Olive *et al.*, "(particle data group)," *Chin. Phys. C*, vol. 38, p. 090001, 2014.

[25] M. Tanabashi *et al.*, "(particle data group)," *Phys. Rev. D*, vol. 98, p. 030001, 2018.

[26] T. KAJITA, "Atmospheric neutrinos and discovery of neutrino oscillations," *Proceedings of the Japan Academy, Series B*, vol. 86, no. 4, pp. 303–321, 2010.

[27] T. K. Gaisser, "Atmospheric neutrinos in the context of muon and neutrino radiography," *Earth Planets Space*, vol. 62, pp. 195–199, 2010.

[28] `https://home.web.cern.ch/science/physics/cosmic-rays-particles-outer-space`. accessed: 2022-07-22.

[29] P. A. Cherenkov, "Visible luminescence of pure liquids under the influence of γ-radiation," *Dokl. Akad. Nauk SSSR*, vol. 2, no. 8, pp. 451–454, 1934.

[30] S. G. Warren, "Optical constants of ice from the ultraviolet to the microwave," *Appl. Opt.*, vol. 23, pp. 1206–1225, Apr 1984.

[31] M. Grootendorst, M. Cariati, A. Kothari, D. Tuch, and A. Purushotham, "Cerenkov luminescence imaging (cli) for image-guided cancer surgery," *Clinical and Translational Imaging*, vol. 4, 05 2016.

[32] T. Gluesenkamp, *On the Detection of Subrelativistic Magnetic Monopoles with the IceCube Neutrino Observatory.*
PhD thesis, 01 2010.

[33] L. Rädel and C. Wiebusch, "Calculation of the cherenkov light yield from low energetic secondary particles accompanying high-energy muons in ice and water with geant4 simulations," *Astroparticle Physics*, vol. 38, pp. 53–67, oct 2012.

[34] I. M. Frank and I. E. Tamm, "Coherent visible radiation of fast electrons passing through matter," *Compt. Rend. Acad. Sci. URSS*, vol. 14, no. 3, pp. 109–114, 1937.

[35] L. Morel, Z. Yao, P. Cladé, and S. Guellati-Khélifa, "Determination of the fine-structure constant with an accuracy of 81 parts per trillion," *Nature*, vol. 588, no. 7836, pp. 61–65, 2020.

[36] B. Pontecorvo, "Neutrino Experiments and the Problem of Conservation of Leptonic Charge," *Zh. Eksp. Teor. Fiz.*, vol. 53, pp. 1717–1725, 1967.

[37] A. Y. Smirnov, "Solar neutrinos: Oscillations or no-oscillations?," 2017.

[38] Y. Fukuda *et al.*, "Evidence for oscillation of atmospheric neutrinos," *Physical Review Letters*, vol. 81, pp. 1562–1567, aug 1998.

[39] C. F. Gauss, *Theoria attractionis corporum sphaeroidicorum ellipticorum homogeneorum, methodo nova tractata*, pp. 279–286.
Berlin, Heidelberg: Springer Berlin Heidelberg, 1877.

[40] L. Wolfenstein, "Neutrino oscillations in matter," *Phys. Rev. D*, vol. 17, pp. 2369–2374, May 1978.

[41] W. J. Marciano and Z. Parsa, "Neutrino–electron scattering theory," *Journal of Physics G: Nuclear and Particle Physics*, vol. 29, pp. 2629–2645, oct 2003.

[42] J. Kopp, "Phenomenology of Three-Flavour Neutrino Oscillations," diploma, Technische Universität München, München, 2006.

[43] H. W. Zaglauer and K. H. Schwarzer, "The Mixing Angles in Matter for Three Generations of Neutrinos and the Msw Mechanism," *Z. Phys. C*, vol. 40, p. 273, 1988.

[44] A. M. Dziewonski and D. L. Anderson, “Preliminary reference earth model,” *Physics of the Earth and Planetary Interiors*, vol. 25, no. 4, pp. 297–356, 1981.

[45] A. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung.* Ergebnisse der Mathematik und ihrer Grenzgebiete, J. Springer, 1933.

[46] R. A. Fisher, “On the ”probable error” of a coefficient of correlation deduced from a small sample.,” 1921.

[47] R. A. Fisher, *Statistical Methods for Research Workers*, pp. 66–70. New York, NY: Springer New York, 1992.

[48] A. Birnbaum, “On the foundations of statistical inference,” *Journal of the American Statistical Association*, vol. 57, no. 298, pp. 269–306, 1962.

[49] G. A. Barnard, G. M. Jenkins, and C. B. Winsten, “Likelihood inference and time series,” *Journal of the Royal Statistical Society. Series A (General)*, vol. 125, no. 3, pp. 321–372, 1962.

[50] J. Neyman and E. S. Pearson, “On the problem of the most efficient tests of statistical hypotheses,” *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, pp. 289–337, 1933.

[51] B. Duignan, “Occam’s razor.” `https://www.britannica.com/topic/Occams-razor`. Accessed: 2022-06-18.

[52] S. S. Yadav and S. M. Jadhav, “Deep convolutional neural network based medical image classification for disease diagnosis,” *Journal of Big Data*, vol. 6, pp. 1–18, 2019.

[53] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, “Cardiologist-level arrhythmia detection with convolutional neural networks,” 2017.

[54] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, “Deep speech: Scaling up end-to-end speech recognition,” 2014.

[55] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” 2018.

[56] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access*, vol. 8, p. 58443–58469, 2020.

[57] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[58] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, 01 2016.

[59] M. Abbas, A. Khan, A. S. Qureshi, and M. W. Khan, "Extracting signals of higgs boson from background noise using deep neural networks," 2020.

[60] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle, "A convolutional neural network neutrino event classifier," *Journal of Instrumentation*, vol. 11, p. P09001–P09001, Sep 2016.

[61] R. Abbasi *et al.*, "A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory," 1 2021.

[62] N. Metropolis and S. Ulam, "The monte carlo method," *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[63] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking state-of-the-art deep learning software tools," 2016.

[64] G. Hooker, *Diagnostics and Extrapolation in Machine Learning.*
PhD thesis, Stanford, CA, USA, 2004.
AAI3145521.

[65] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct 1986.

[66] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[67] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, p. 403–420, apr 1970.

[68] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[69] C. G. Tianqi Chen, "XGBoost: A Scalable Tree Boosting System," 2016.

[70] A. Y. C. V. N. Vapnik, "A class of algorithms for pattern recognition learning," *Avtomat. i Telemekh.*, vol. 25, no. 6, pp. 937–945, 1964.

[71] H. Steinhaus, "Sur la division des corps matériels en parties," *Bull. Acad. Pol. Sci., Cl. III*, vol. 4, pp. 801–804, 1957.

[72] C. Watkins, "Learning from delayed rewards," 01 1989.

[73] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, Dec 1943.

[74] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton," Tech. Rep. 85-460-1, Cornell Aeronautical Laboratory, Ithaca, New York, January 1957.

[75] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018.

[76] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[77] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[78] H. E. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 2007.

[79] M. G. Aartsen *et al.*, "The IceCube Neutrino Observatory: Instrumentation and Online Systems," *JINST*, vol. 12, no. 03, p. P03012, 2017.

[80] R. Abbasi *et al.*, "Icetop: The surface component of icecube," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 700, p. 188–220, Feb 2013.

[81] M. Rongen. internal communication.

[82] K. Andeen and M. Plum, "Latest cosmic ray results from icetop and icecube," *EPJ Web of Conferences*, vol. 210, p. 03005, 01 2019.

[83] V. Petrenko and R. Whitworth, *Physics of Ice.*
OUP Oxford, 1999.

[84] M. G. Aartsen *et al.*, "Measurement of South Pole ice transparency with the IceCube LED calibration system," *Nucl. Instrum. Meth. A*, vol. 711, pp. 73–89, 2013.

[85] M. Rongen, *Calibration of the IceCube Neutrino Observatory.*
PhD thesis, RWTH Aachen U., 2019.

[86] D. Lilien, T. Fudge, M. Koutnik, H. Conway, E. Osterberg, D. Ferris, E. Waddington, and C. Stevens, "Holocene ice-flow speedup in the vicinity of south pole," *Geophysical Research Letters*, vol. 45, 06 2018.

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

[88] H. P. K.K., *Photomultiplier Tubes: Basics and Applications.*
Hamamatsu Photonics K.K.

[89] A. Einstein, "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt," *Annalen der Physik*, vol. 322, pp. 132–148, Jan. 1905.

[90] Hamamatsu Photonics K.K., *Large photocathode area photomultiplier tubes*, 3 2019.
Rev. 1.

[91] S. Kleinfelder, "Gigahertz waveform sampling and digitization circuit design and implementation," *IEEE Transactions on Nuclear Science*, vol. 50, no. 4, pp. 955–962, 2003.

[92] M. G. Aartsen *et al.*, "Energy Reconstruction Methods in the IceCube Neutrino Telescope," *JINST*, vol. 9, p. P03009, 2014.

[93] R. Abbasi *et al.*, "Calibration and characterization of the icecube photomultiplier tube," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 618, p. 139–152, Jun 2010.

[94] K. Ma *et al.*, "Time and amplitude of afterpulse measured with a large size photomultiplier tube," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 629, pp. 93–100, feb 2011.

[95] R. Abbasi *et al.*, "The design and performance of icecube deepcore," *Astroparticle Physics*, vol. 35, p. 615–624, May 2012.

[96] M. J. Larson, *A Search for Tau Neutrino Appearance with IceCube-DeepCore.* PhD thesis, 03 2018.

[97] A. Ishihara, "The icecube upgrade – design and science goals," 2019.

[98] L. Classen, A. Kappes, and T. Karg, "A multi-pmt optical module for the icecube upgrade," 2019.

[99] C. Hill, M. Meier, R. Nagai, K. Kin, N. Shimizu, A. Ishihara, S. Yoshida, T. Anderson, J. Braun, A. Fienberg, and J. Weber, "Performance of the d-egg optical sensor for the icecube upgrade," 2021.

[100] H. Löhner *et al.*, "The multi-PMT optical module for KM3NeT," *Nucl. Instrum. Meth. A*, vol. 718, pp. 513–515, 2013.

[101] S. Aiello *et al.*, "The KM3NeT multi-PMT optical module," 3 2022.

[102] R. Abbasi *et al.*, "Low Energy Event Reconstruction in IceCube DeepCore," 3 2022.

[103] C. Giunti and C. W. Kim, *Fundamentals of Neutrino Physics and Astrophysics.* 2007.

[104] M. P. Kowalski, *Search for neutrino-induced cascades with the AMANDA-II detector.*
PhD thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I, 2004.

[105] M. Vehring, *Measurement of neutrino oscillations with IceCube-DeepCore.* Dissertation, RWTH Aachen University, Aachen, 2017.

Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2017.

[106] R. Gandhi, C. Quigg, M. H. Reno, and I. Sarcevic, "Ultrahigh-energy neutrino interactions," *Astropart. Phys.*, vol. 5, pp. 81–110, 1996.

[107] J. Neyman and E. S. Pearson, "On the problem of the most efficient tests of statistical hypotheses," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 231, pp. 289–337, 1933.

[108] A. Margiotta, "The KM3NeT deep-sea neutrino telescope," *Nucl. Instrum. Meth. A*, vol. 766, pp. 83–87, 2014.

[109] K. Abe *et al.*, "Letter of Intent: The Hyper-Kamiokande Experiment — Detector Design and Physics Potential —," 9 2011.

[110] R. J. Barlow, "Extended maximum likelihood," *Nucl. Instrum. Meth. A*, vol. 297, pp. 496–506, 1990.

[111] A. Trettin, "Oscnext." `https://events.icecube.wisc.edu/event/154/contributions/8171/`.
IceCube spring meeting: 2022-05-19.

[112] "Lightgbm documentation." `https://lightgbm.readthedocs.io/en/latest/`.
Accessed: 2022-06-10.

[113] C. M. Bishop, *Neural Networks for Pattern Recognition.*
USA: Oxford University Press, Inc., 1995.

[114] "Icosphere python package." `https://pypi.org/project/icosphere/`.
Accessed: 2022-05-30.

[115] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine Learning*, vol. 50, pp. 5–43, Jan 2003.

[116] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, "emcee: The MCMC hammer," *Publications of the Astronomical Society of the Pacific*, vol. 125, pp. 306–312, mar 2013.

[117] A. Bellerive, J. R. Klein, A. B. McDonald, A. J. Noble, and A. W. P. Poon, "The Sudbury Neutrino Observatory," *Nucl. Phys. B*, vol. 908, pp. 30–51, 2016.

[118] A. Abusleme *et al.*, "JUNO Physics and Detector," 4 2021.

[119] M. Leuermann, *Testing the neutrino mass ordering with IceCube DeepCore.*
Dissertation, RWTH Aachen University, Aachen, 2018.
Veröffentlicht auf dem Publikationsserver der RWTH Aachen University 2019; Dissertation, RWTH Aachen University, 2018.

[120] P. Kaelo and M. M. Ali, "Some variants of the controlled random search algorithm for global optimization," *Journal of Optimization Theory and Applications*, vol. 130, pp. 253–264, Aug 2006.

[121] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, 01 1965.

[122] M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832 – 837, 1956.

[123] E. Parzen, "On Estimation of a Probability Density Function and Mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065 – 1076, 1962.

[124] F. James, "MINUIT Function Minimization and Error Analysis: Reference Manual Version 94.1," 1994.

[125] F. J. Massey, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[126] B. Efron, "Bootstrap Methods: Another Look at the Jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1 – 26, 1979.

[127] "Nvidia cudnn." `https://developer.nvidia.com/cudnn`.
Accessed: 2022-06-23.

[128] "Nvidia v100 tensor core gpu." `https://images.nvidia.com/content/technologies/volta/pdf/volta-v100-datasheet-update-us-1165301-r5.pdf`.
Accessed: 2022-05-26.

[129] M. H. Minh, "Overview of machine learning efforts for low-energy event reconstruction." `https://events.icecube.wisc.edu/event/143/contributions/7796/`.
IceCube fall meeting: 2021-09-22.

[130] IceCube Collaboration, "Computational techniques for the analysis of small signals in high-statistics neutrino oscillation experiments," 2018.

[131] G. Cowan, K. Cranmer, E. Gross, and O. Vitells, "Asymptotic formulae for likelihood-based tests of new physics," *Eur. Phys. J. C*, vol. 71, p. 1554, 2011.
[Erratum: Eur.Phys.J.C 73, 2501 (2013)].

[132] P. Collaboration, "Letter of intent: The precision icecube next generation upgrade (pingu)," 2017.

[133] M. G. Aartsen *et al.*, "Measurement of Atmospheric Tau Neutrino Appearance with IceCube DeepCore," *Phys. Rev. D*, vol. 99, no. 3, p. 032007, 2019.

[134] M. G. Aartsen *et al.*, "Measurement of Atmospheric Neutrino Oscillations at 6–56 GeV with IceCube DeepCore," *Phys. Rev. Lett.*, vol. 120, no. 7, p. 071801, 2018.

[135] Y. Avni, "Energy spectra of X-ray clusters of galaxies.," *The Astrophysical Journal*, vol. 210, pp. 642–646, Dec. 1976.
Provided by the SAO/NASA Astrophysics Data System.

[136] M. G. Aartsen *et al.*, "Development of an analysis to probe the neutrino mass ordering with atmospheric neutrinos using three years of IceCube DeepCore data," *The European Physical Journal C*, vol. 80, jan 2020.

[137] M. Blennow, P. Coloma, P. Huber, and T. Schwetz, "Quantifying the sensitivity of oscillation experiments to the neutrino mass ordering," *Journal of High Energy Physics*, vol. 2014, mar 2014.

[138] E. Ciuffoli, J. Evslin, and X. Zhang, "Sensitivity to the neutrino mass hierarchy," 2013.

[139] R. Abbasi *et al.*, "A muon-track reconstruction exploiting stochastic losses for large-scale cherenkov detectors," *Journal of Instrumentation*, vol. 16, p. P08034, aug 2021.

[140] L. Halve, "Internal report," 2022.

[141] T. Kozynets, "Studies of the mdom pmt noise and event cleaning." `https://events.icecube.wisc.edu/event/154/contributions/8152/`.
IceCube spring meeting: 2022-05-18.

[142] K. Pearson, "Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia," *Philosophical Transactions of the Royal Society A*, vol. 187, pp. 253–318.

[143] "Errors in weighted histograms." `https://www.zeuthen.desy.de/~wischnew/amanda/discussion/wgterror/working.html`.
Accessed: 2022-06-15.

[144] J. Micallef, "Fast low energy reconstruction using convolutional neural networks (flercnn)." `https://github.com/jessimic/LowEnergyNeuralNetwork`.
accessed: 2022-07-16.

[145] M. H. Minh, "Reconstruction of neutrino events in icecube using graph neural networks," 2021.

[146] "Eos detector." `https://nino.lbl.gov/eos/detector.html`.
Accessed: 2022-06-29.

[147] N. van Eijndhoven, O. Fadiran, and G. Japaridze, "Implementation of a gauss convoluted pandel pdf for track reconstruction in neutrino telescopes," *Astroparticle Physics*, vol. 28, p. 456–462, Dec 2007.

[148] G. D. Barr, S. Robbins, T. K. Gaisser, and T. Stanev, "Uncertainties in atmospheric neutrino fluxes," *Physical Review D*, vol. 74, nov 2006.

[149] A. Gazizov and M. Kowalski, "Anis: High energy neutrino generator for neutrino telescopes," *Computer Physics Communications*, vol. 172, no. 3, pp. 203–213, 2005.