Muon track reconstruction in Liquid Scintillation Detectors with GCNs

Myonspurrekonstruktion im Flüßigszintillations-Detektor mit Hilfe von GCNs

by
Rosmarie Irma Wirth
date of birth
09.12.1993

Master thesis
University of Hamburg
2021

1st reviewer: Dr. Björn Wonsak
2nd reviewer: Prof. Dr. Gregor Kasieczka

Abstract

Large liquid scintillation detectors are successfully used to observe the neutrino oscillation parameters by detecting reactor neutrinos. A main and hard to identify background are cosmogenics. These are excited ^{9}Li and ^{8}He atoms, which are produced in showers along cosmic muon tracks inside the detector. While decaying the cosmogenics can mimic the inverse β-decay, which is the detection process to identify reactor neutrinos. Muon vetoes are a straight forward method to reduce this background, but it creates a lot of dead time. With the JUNO experiment 15.7 % of the reactor neutrino events are predicted to be missed due to the muon vetoes. A superior muon track and shower reconstruction method, could improve the data taking of JUNO and comparable experiments tremendously.

The here presented work studies the use of graph convolution networks to reconstruct muon tracks and corresponding showers. Graph convolution networks provide the option to include the geometric detector setup to improve the reconstruction. On TOY Monte Carlo Data showers in the detector volume could be located with an accuracy of $\pm(0.22 \pm 0.14)$ m. Additionally, the photon emission distribution was reconstructed in voxel representation. The results on TOY Monte Carlo Data are a discrimination of (99.24 ± 2.84) % of the *path* voxels to less than 7 % of the detector volume. Simulated showers are reconstructed with a mean distance of (0.36 ± 0.47) m. For 85 % of the analyzed data, the showers are reconstructed within a distance ≤ 0.5 m to the simulated position. These shower positions can be narrowed to ≤ 0.8 % of the detector volume. A veto method based on further developments of the presented work could directly increase the reactor neutrino signal at JUNO.

Kurzzusammenfassung

Um die Neutrinooszillations zu untersuchen können Reaktorneutrinos mit Flüssigzintillations-Detektoren beobachtet werden. Kosmische Myonen führen dabei zu schwer trennbaren Hintergrundsignalen. Entlang der Myonspuren im Flüßigszintillator können Schauer entstehen, in welchen Cosmogene auftauchen können. Das sind angeregte ^{9}Li und ^{8}He Atome, welche beim Zerfall ein Signal, ähnlich zum inversen β Zerfall, erzeugen können. Dieser wird als Nachweisreaktion für Reaktorneutrinos im Flüßigszintillator genutzt. Um diesen Hintergrund zu unterdrücken werden oft Myonvetos genutzt. Diese haben den Nachteil, dass sie zu Totzeiten im Detektor führen können. Beim JUNO Experiment werden voraussichtlich $15,7\,\%$ der Reaktorneutrinos wegen des Myonvetos nicht detektiert. Eine verbesserte Methode zur Identifikation von Schauern entlang von Myonspuren könnte die Menge an detektierten Reaktorneutrinos mit JUNO erheblich vergrößern.
In dieser Masterarbeit wird die Nutzung von neuronale Graphennetze zur Schaueridentifikation entlang einer Myonspur untersucht. Neuronale Graphennetze bieten die Möglichkeit die geometrische Struktur des Detektors zu nutzen. Die in dieser Arbeit genutzten Daten wurden mit der Toy Monte Carlo Methode simuliert. Mit diesen Daten konnten Schauer in einem Detektor mit einem mittleren Abstand von $\pm(0,22 \pm 0,14)\,\mathrm{m}$ lokalisiert werden. Zusätzlich wird auch die Verteilung der Photonenemission im Detektor rekonstruiert. Damit konnten $(99,24 \pm 2,84)\,\%$ der Voxel, welche die Myonspur enthalten, auf unter $7\,\%$ des Detektorvolumens eingegrenzt werden. Die simulierten Schauer konnten mit einem mittleren Abstand von $(0,36 \pm 0,47)\,\mathrm{m}$ rekonstruiert werden. Für $85\,\%$ der analysierten Daten konnte damit der Schauer mit einem Abstand von unter $0,5\,\mathrm{m}$ bestimmt werden. Diese Schauer können damit auf $\leq 0,8\,\%$ des Detektorvolumens begrenzt werden. Eine Vetomethode basieren auf einer Weiterentwicklung dieser Arbeit könnte somit die Menge an detektieren Reaktorneutrinos, in Experimenten wie JUNO, vergrößern.

Contents

1 Introduction 1

2 Neutrino Physics 3

2.1 Neutrinos in the Standard Model 3
  2.1.1 Weak Force 4
  2.1.2 Neutrino Interactions 6
2.2 Neutrino Oscillations 7
  2.2.1 Mikheyev-Smirnov-Wolfenstein Effect 10
2.3 Open Questions in Neutrino Physics 11
  2.3.1 Mass Hierarchy 11
  2.3.2 CP Violation 12
  2.3.3 Current Status of Neutrino Oscillation Parameters 13

3 Neutrino Detection with Liquid Scintillators 15

3.1 Liquid Scintillators 15
3.2 Neutrino Sources 16
  3.2.1 Reactor Neutrinos 17
3.3 JUNO 17
  3.3.1 Research Agenda 19
  3.3.2 Signal 20
  3.3.3 Background 20
    Cosmogenics 21
  3.3.4 Background suppression methods 21
    Potential of further Background suppression methods 23
3.4 Future Detectors - Theia 23
  3.4.1 Outstanding Features 24
  3.4.2 Physics Program 24

4 Graph Neural Networks 27

4.1 Introduction into Neural Networks 27
  4.1.1 Structure 27
  4.1.2 Training 28

4.1.3 Monitoring and Evaluation 31
ROC Curve 32
4.2 Graph Neural Networks (GNN) 33
4.2.1 Structure 33
4.2.2 Graph Convolution Networks (GCN) 34
Graph Convolution 35
Edge Convolution 36
4.3 Geometric Loss Function 36

5 Simulation and Data **39**
5.1 Simulation 39
5.1.1 Detector 40
5.1.2 Path 40
5.1.3 Photons 41
5.2 Generated Data 42
5.3 Classic Data Analysis 44
5.3.1 Simple Coordinate Reconstruction 44
5.3.2 Simple Voxel Reconstruction 46

6 Coordinate Reconstruction **51**
6.1 Goal 51
6.2 Implementation 52
6.2.1 Basic Idea 52
6.2.2 Optimization 53
6.3 Optimized Architecture 54
6.3.1 Loss Function 54
6.4 Results 55

7 Voxel Reconstruction **59**
7.1 Goal 59
7.2 Implementation 60
7.2.1 Basic Idea 60
7.2.2 Optimization 61
7.2.3 Loss Function 62
7.2.4 Graph 64
7.3 Architecture 65
7.3.1 Optimized Graph 66
7.3.2 Loss Function 66
Geometrical Loss 66
Cross Entropy Loss 67
Mean Squared Error Loss 67

Total Loss Function . . . 68
7.4 Results . . . 68
7.4.1 Defining the cut . . . 72
Conservative cut . . . 73
Cut with a higher Background discrimination . . . 76
7.4.2 Summary . . . 78

8 Conclusion 81
8.1 Summary and Conclusion . . . 81
8.1.1 Coordinate Reconstruction . . . 81
8.1.2 Voxel Reconstruction . . . 82
8.2 Outlook . . . 83
8.2.1 Potential Application at Theia . . . 84

Bibliography 86

List of Abbreviations 93

Appendix 95

Acknowledgments 105

Chapter 1

Introduction

Neutrinos are fundamental particles described by the Standard Model of Elementary Particles (SM). But despite its great success, the neutrinos do not behave as predicted by the SM. In the SM the neutrinos are considered massless, but the neutrino masses have been proven to be different form zero, by the observation of neutrino oscillations. The neutrino masses and the neutrino oscillations lead to the mass hierarchy problem. Additionally, the neutrino is part of many beyond the Standard Model theories, like for example Grand Unification theories. These theories try to unify three of the four fundamental forces to describe the early stages of the universe. Therefore, the neutrino is an interesting particle and the observation of its properties might lead to meaningful discoveries. The physics of the neutrino is introduced in chapter 2.
The Jiangmen Underground Neutrino Observatory (JUNO) is a large liquid scintillation detector, with 20 kt volume. Right now it is under construction and the first measurements are scheduled for 2021. Its goals are to observe the neutrino oscillations in order to determine the mass hierarchy and measure the oscillation parameters. This will be achived by measuring reactor neutrinos from two power plants in about 53 km distance. The detection reaction is the inverse β decay (IBD). At JUNO about 83 /day reactor neutrino events are predicted. This small rate, despite the large detector volume, shows how rarely the neutrino interacts. To detect rare interactions it is extremely important to understand the background and ideally have the opportunity to distinguish it from the signal. Eventhough JUNO is about 700 m under ground to shield the detector form cosmic muons, these are the biggest background at JUNO. The expected cosmic muon rate is 3.5 Hz with an average energy of 215 GeV [1]. This is a tremendous amount and makes the muon vetos mandatory. Cosmic muons do not have the characteristic signature of the IBD signal and are therefore no direct background for reactor neutrinos. But along the muon tracks showers can occur, which can result in radioactiv isotopes called cosmogenics. For JUNO these are ^{9}Li and ^{8}He. The decay of these isotopes can mimic the IBD and are thereby direct background. To suppress this background JUNO plans to use muon vetos. These have an efficiency of $\approx 98\,\%$ on the cosmogenics. But the efficiency on the IBD is only about 83 % [1]. An improved muon veto could directly improve the efficiency of JUNO for reactor neutrinos. JUNO is introduced in chapter 3.3.

One approach which tackles this problem is the Quadratic Reconstruction (QR) [2]. This method uses classical computation techniques to reconstruct the photon emission distribution inside a detector, which can be used for shower identification. The results of this method are excellent. A background suppression technique based on this method could increase the reactor neutrino signal at JUNO by approximately 10 %.

Basically, this work tackles the same problem by using machine learning methods. A full development of an improved muon veto for JUNO or a method similar to QR would by far exceed the frame of this work. Therefore, a machine learning approach is developed on simulated data with simplified constrains.

Working on the same problem as an already successful method is in no way irrelevant. On the one hand having a second method, which independently comes to the same results is a validation of measurement or reconstructions. On the other hand, the QR works with look up tables. This is a common method to fasten up computations on complicated calculations. But it has the downside, that theses tables can use a lot of memory space and need to be kept up to date or newly made for new applications. In contrast to machine learning techniques, which typically do not need much memory space. No lookup tables need to be computed or fostered. Additionally, taken data can be used to improve the algorithm along the way, improving the prediction with time of use. Furthermore, the machine learning methods are versatile and with some more new training a similar architecture can be recycled for a new application. All in all, machine learning techniques appear to be a great method for a reconstruction task like this.

As cosmogenics occur in combination with showers along muon tracks, the goal of this work is to develop a machine learning based reconstruction methods, which identify showers along muon tracks in liquid scintillation detectors. Furthermore, the developed method is supposed to reconstruct the photon emission distribution for a muon track inside the detector. To reach this goal data sets are simulated with simplified constrains. The simulation and the corresponding data sets are introduced in chapter 5. For this work graph convolution networks are used. These provide the benefit, that the structure of given data can be implemented in the network via the graph. The goal, to identify showers along muon tracks and reconstruct corresponding photon distributions is split in two steps. Firstly, the muon track is reconstructed spatially with the help of coordinate representation, presented in chapter 6. Afterwards, the muon track and the corresponding photon emission distribution are reconstructed in voxel representation, discussed in chapter 7. This approach has the advantage that in the first simpler step, a suitable methods is developed. The approved method is developed further to obtain the goal of the second step. After the presentation of the results a conclusion with an outlook for further applications is given in chapter 8.

Chapter 2

Neutrino Physics

In this chapter physics relevant for this work is introduced. Neutrino physics with focus on neutrino oscillation and mass hierarchy will be covered, as well as open questions. This chapter is mainly based on [3].

2.1 Neutrinos in the Standard Model

This section introduces the Standard Model of Elementary Particles (SM) with focus on the neutrinos. The SM describes the essential knowledge of particle physics. It classifies all known particles and describes their interactions with three of the four fundamental forces. While electromagnetism, weak and strong force are covered by the Standard Model, gravitation is not included. The elementary particles of the SM are displayed in figure 2.1. It shows the masses, charges and spins of all fundamental particles. The particles in the SM are associated with antiparticles, these have the same mass, but opposite charge. The fundamental particles are classified into two groups, the bosons and the fermions. The gauge bosons transmit the elementary forces, except for the Higgs boson, which has a special role in the SM. All elementary bosons have an integer spin and follow the Bose-Einstein statistics. The fermions follow the Fermi-Dirac statistics and have a half-integer spin. Fermions are further classified in quarks and leptons, both have three generations. The charged leptons are the electron, muon and tau, all carrying a negative electric charge of -1. Each charged lepton has an uncharged partner particle within the same generation, the electron, muon and tau neutrino. Neutrinos do not carry electromagnetic or color charge, thus they only interact via the weak force. Weak interactions are transmitted by the Z^0 boson referred to as "neutral current" and the $W^\pm$ boson as "charged current". The neutrino interactions are described in more detail in section 2.1.2. Even though in figure 2.1 the neutrinos are denoted with an upper mass limit, they are considered massless in the Standard Model. The given mass limit was set by KATRIN in 2019 [5]. The SM is a very successful theory. It predicted particles like the Z^0 boson, top and charm quark and the Higgs boson, which were later found in experimental research. Still it has some

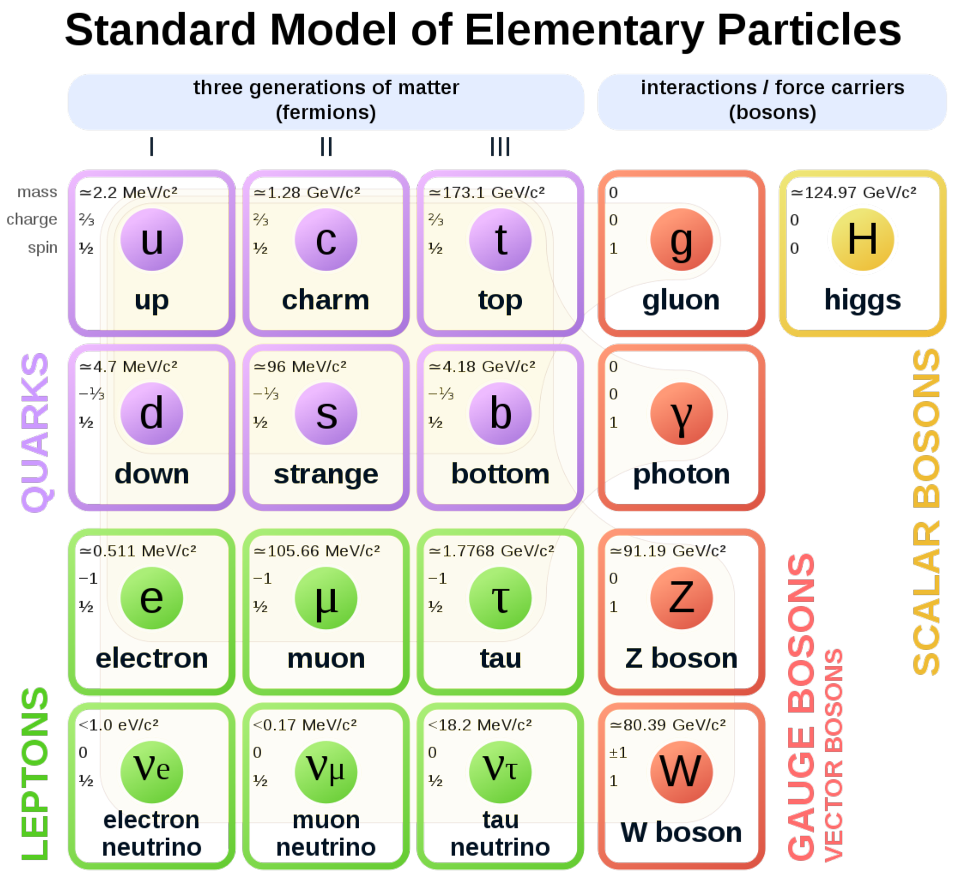

Standard Model of Elementary Particles
three generations of matter (fermions)
interactions / force carriers (bosons)
I
II
III
mass
charge
spin
≃2.2 MeV/c² ⅔ ½ u up
≃1.28 GeV/c² ⅔ ½ c charm
≃173.1 GeV/c² ⅔ ½ t top
0 0 1 g gluon
≃124.97 GeV/c² 0 0 H higgs
≃4.7 MeV/c² −⅓ ½ d down
≃96 MeV/c² −⅓ ½ s strange
≃4.18 GeV/c² −⅓ ½ b bottom
0 0 1 γ photon
≃0.511 MeV/c² −1 ½ e electron
≃105.66 MeV/c² −1 ½ μ muon
≃1.7768 GeV/c² −1 ½ τ tau
≃91.19 GeV/c² 0 1 Z Z boson
<1.0 eV/c² 0 ½ νe electron neutrino
<0.17 MeV/c² 0 ½ νμ muon neutrino
<18.2 MeV/c² 0 ½ ντ tau neutrino
≃80.39 GeV/c² ±1 1 W W boson
QUARKS
LEPTONS
GAUGE BOSONS
VECTOR BOSONS
SCALAR BOSONS

Figure 2.1: Elementary Particles of the Standard Model of Elementary Particles [4].

shortcomings. One of which are the mentioned neutrino masses, which are proven to be non-zero by neutrino oscillations, which is covered in chapter 2.2. Additionally, the Standard Model does not include the gravitation, even though it is one of four fundamental forces. Dark matter and matter-anti matter asymmetry are not adequately explained in the SM either. There are many different theories which extend the Standard Model, to explain these open problems.

2.1.1 Weak Force

The weak force is one of the four fundamental forces. Neutrinos only interact weakly. The weak force is the only fundamental force, which violates charge, parity and CP symmetry:

Parity Symmetry

Partity transformation describes the spacial transformation of coordinates, which are flipped in sign:

$$\mathbf{P}:\begin{pmatrix} t \\ x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} t \\ -x \\ -y \\ -z \end{pmatrix}. \tag{2.1}$$

In equation 2.1 a parity transformation is shown. While the spatial coordinates x, y, z change sign, the time coordinate t stays unchanged under parity transformation. The Wu experiment revealed in 1956, that parity is violated in weak interactions [6].
The parity symmetry is violated by the weak force, because its gauge bosons only interact with left-handed particles and right-handed anti-particles. The handedness or helicity is defined as the projection of the spin onto the direction of momentum of a particle. A particle has right-handed helicity, when the spin projection is parallel to the direction of momentum and has left-handed helicity in case it is anti-parallel, respectively. As the direction of momentum depends on the reference frame, the helicity can be flipped with a Lorentz boost, in case the considered particle is massive. For massless particles the helicity is Lorentz invariant. In this case the chirality is equal to the helicity. The chirality is a Lorentz invariant property for massive and massless particles. For Dirac fermions it is defined by the operator γ^5, which has the eigenvalues ± 1. This operator projects any Dirac field into its left- and right-handed components. In the SM all fermions are Dirac fermions, as they have spin 1/2 and are different form their anti-particles.
As the weak force does not interact with right-handed fermions and left-handed anti-fermions, it violates charge symmetry.

Charge Symmetry
Charge symmetry is observed, when physical laws are valid in a process which is charge conjugated as well as in the original process. Charge conjugation describes the change of every particle in a considered frame with its corresponding anti-particle and vice versa. As the weak force only couples to left-handed particles and right-handed anti-particles, it violates charge symmetry. For all other fundamental forces the charge symmetry is obtained.

CP Symmetry
The combination of parity and charge symmetry is the CP symmetry. CP symmetry in conserved when a process stays unchanged under charge and parity conjugation. In 1964 the CP violation of the weak force was discovered in the decay of neutral kaons [7]. This only affects the quark sector of the weak interaction. Whether the CP is also violated in the leptonic sector is not yet known. The open problem of CP violation in the leptonic sector is discussed in section 2.3.2.

The Glashow-Weinberg-Salam model (GWS) unifies the weak force with the electromagnetic force in the elektroweak theory. It consists of the weak isospin SU(2) and the hypercharge U(1). The generators are the weak isospin T_3 and the weak hypercharge Y, the combination of both is the electric charge Q, with $Q = T_3 + Y/2$. The weak isospin is a quantum number. It can be interpreted as the weak charge, while the hypercharge is the charge of electroweak interactions. The weak isospin is defined to be $+1/2$ for all left-handed neutrinos and $-1/2$

for the left-handed charged leptons e^-, μ^- and τ^-. All right-handed particles and left-handed anti-particles have a weak isospin of 0. The weak isospin is conserved in weak interactions. For this theory the fermions are clustered in singlets and doublets:

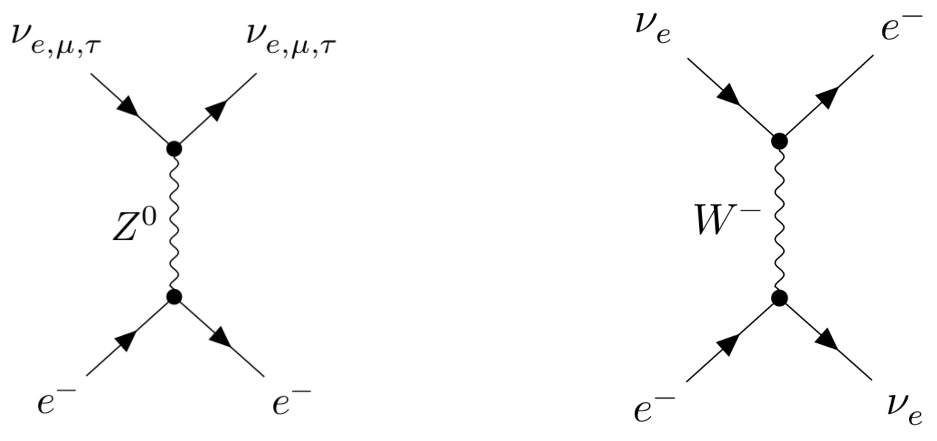
$$\begin{pmatrix} \nu_e \\ e^- \end{pmatrix}_L, \quad \begin{pmatrix} \nu_\mu \\ \mu^- \end{pmatrix}_L, \quad \begin{pmatrix} \nu_\tau \\ \tau^- \end{pmatrix}_L, \quad e_R, \quad \mu_R, \quad \tau_R \tag{2.2}$$

In equation 2.2 the leptonic, electroweak singlets and doublets are displayed. The leptonic doublets consist of a charged lepton and the neutrino from the same generation, the singlets only hold the charged leptons. As the electroweak force describes the unification of electromagnetic and weak force the charged lepton singlets are needed to describe their electromagnetic interactions. The right-handed neutrinos are neither interacting weakly nor electromagnetically and thereby not described by this theory. The quarks also arranged in singlets and doublets, they interact weakly and electromagnetically.

2.1.2 Neutrino Interactions

The neutrinos only interact weakly, therefore only left-handed neutrinos and right-handed anti-neutrinos interact.

Charged and Neutral Current
The weak force is transmitted by the $W^\pm$ and the Z^0 boson. These two possible interaction types are called charged and neutral current, displayed in figure 2.2. The charged current is transmitted by the charged $W^\pm$ boson, the Z^0 boson is neutral and transmits the neutral current.



(a) Neutral current interaction.

(b) Charged current interaction.

Figure 2.2: Examples for neutral and charged current transmitted over Z^0 or $W^\pm$ with electrons e^- and neutrinos in different flavors $\nu_{e,\mu,\tau}$.

The neutral current in transmitted by the neutral Z^0 bosons. It is an elastic scattering process.

In the displayed case an electron scatters with a neutrino of any flavor. There are many different neutral current interactions possible. All weakly interacting fundamental particles can interact in such an interaction.

The charged current is transmitted by the charged $W^{\pm}$. It couples to both particles of an electroweak doublet. Interactions with two leptons from different generations are not allowed, as this would violate lepton flavor conservation. The most common charged interaction with the neutrino is the β decay. This is also the process involved into the discovery of neutrinos [8].

β Decay

The β decay is a radioactive decay of a nucleus. It can be distinguished between the positive and the negative β-Decay, denoted with β^+- or β^-. In both cases a nucleus emits a charged lepton, either electron or positron, and its partner neutrino, the electron antineutrino or neutrino. The nucleus transforms into an isobar, which is a nucleus with the same number of nucleons, but different composition of protons and neutrons.

For the negative β^- decay a neutron n decays into a proton p, an electron e^- and an electron antineutrino $\overline{\nu_e}$:

$$n \longrightarrow p + e^- + \overline{\nu_e}. \tag{2.3}$$

For the positive β^+ decay, a proton p decays into a neutron a positron e^+ and an electron neutrino ν_e:

$$p \longrightarrow n + e^+ + \nu_e \tag{2.4}$$

For both processes the inverse direction is possible, named inverse β decay (IBD). The β decay processes only occur with the electron neutrino ν_e and the electron anti-neutrino $\overline{\nu_e}$, as they are in the electroweak doublet with the electron e^- or positron e^+, respectively.

2.2 Neutrino Oscillations

Neutrino oscillation describes the property of neutrinos to change flavor, which only occurs if they are massiv. In 1998 neutrino oscillations were first measured with the Super-Kamiokande detector [9], this delivered the first evidence for neutrino masses. The measured flux of atmospheric neutrinos did not match with the theoretical predicted neutrino flux. Neutrino oscillation had already been predicted in 1962 by the theory of Maki, Nakagawa and Sakata [10]. Even though their theory was first proposed, when only two neutrino flavors were known, the theory could be adjusted for three flavors. In 2001 the SNO collaboration published evidence for neutrino oscillations of solar neutrinos [11]. 2015 these results were honored with the Nobel Price for the discovery of neutrino oscillations.

Neutrino oscillations are comparable with mixing in the quark sector. The neutrino flavors electron, muon and tau, which are known from the Standard Model, are the interaction eigenstates.

Additionally, neutrinos have the so called mass eigenstates. When a neutrino is produced in an interaction it needs to be in one of the flavor eigenstates, otherwise it cannot interact. While traveling through time and space, the phases of the wave functions of the mass eigenstates propagate slightly different due to the mass differences. The neutrino becomes thereby a superposition of flavor eigenstates. Thereby it can be detected in an eigenstate different, to the one it has originally been produced in.
The flavor eigenstates ν_α and the mass eigenstates ν_i are mixed over a unitary matrix U, which gives the relation between mass and flavor eigenstates. Here the transformation is given:

$$|\nu_\alpha\rangle = \sum_i U_{\alpha i} |\nu_i\rangle . \tag{2.5}$$

The mass eigenstates are given by $|\nu_i\rangle$, with the corresponding masses m_i, the flavor eigenstates are denoted as $\langle \nu_\alpha|$, with the flavors electron, muon and tau. In the SM three neutrino flavors are predicted, but the formalism in principle is also valid for two or more flavors. In the following the three flavor case is considered, with Roman letter indices the mass eigenstates and with Greek letter indices the flavor eigenstates are denoted. $U_{\alpha i}$ is the (αi)th element of the transformation matrix, known as Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix. It is defined as

$$U_{PMNS} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix} \cdot \begin{pmatrix} e^{i\rho} & 0 & 0 \\ 0 & e^{i\sigma} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.6}$$

with $c_{ij} = \cos\theta_{ij}$ and $s_{ij} = \sin\theta_{ij}$. θ_{ij} represents the three real mixing angles $0 \le \theta_{12}, \theta_{13}, \theta_{23} \le \pi/2$. In total the PMNS matrix has six free parameters: three CP violating phases δ, ρ, σ and three real mixing angles θ_{ij}. The phases ρ, σ are the Majorana phases [12], these are not considered in this work and therefore set to zero. Thereby the second matrix in equation 2.6 becomes the unit matrix and drops out. For this work δ is considered to be the only CP violating phase for neutrino oscillations. The CP is maximally violated for $\delta = \pi$ and not violated for $\delta = 0 = 2\pi$.
To regard the time development of neutrinos, their wave function properties can be used

$$|\nu_\alpha(t)\rangle = e^{-iE_i t} |\nu_i(0)\rangle . \tag{2.7}$$

$|\nu_i(0)\rangle$ describes the neutrino ν_i in the production state at $t = 0$. The energy is defined as $E_i = \sqrt{\mathbf{p}^2 + m_{\nu_i}^2}$, $\mathbf{p}$ describes the momentum and m_i the mass of the considered neutrino. To conclude the time evolution of the flavor eigenstate equation 2.5 is used

$$|\nu_\alpha(t)\rangle = \sum_i U_{\alpha i} e^{-iE_i t} |\nu_i(0)\rangle . \tag{2.8}$$

Equation 2.8 shows nicely the reason for neutrino oscillation. The flavor eigenstate $|\nu_\alpha(t)\rangle$ is defined as a superposition of the three mass eigenstates, with different masses m_i. Due to the different masses, also the energies E_i differ, which results in different exponential phases for the mass eigenstates. Thereby the time development for the three summands differs, resulting in a time dependence of the superposition of the three mass eigenstates.

The probability to detect a neutrino at one point in space-time in a certain flavor β, can be described as in equation 2.9. Here the Dirac notation comes handy, which gives the probability by the squared quantum mechanical amplitude. To find the amplitude the hermitian conjugate $\langle \nu_\alpha|$ of $|\nu_\alpha\rangle$ in equation 2.5 is used. It is defined as $\langle \nu_\alpha| = \sum_j U^*_{\alpha j} \langle \nu_j|$. The probability of a neutrino, produced in flavor α, to be detected at time t in flavor β is given by

$$P(\nu_\alpha \longrightarrow \nu_\beta) = |\langle \nu_\beta|\nu_\alpha(t)\rangle|^2 = \left| \sum_{i,j=1}^{3} U_{\alpha i} U^*_{\beta j} \cdot e^{-iE_i t} \langle \nu_j|\nu_i\rangle \right|^2 . \tag{2.9}$$

For the next step two assumptions need to be made. Firstly, the eigenstates are considered to be normalized and orthogonal by definition. This allows to cancel out the double summation, because $\langle \nu_i|\nu_j\rangle = \delta_{ij}$. Secondly, neutrinos are highly relativistic particles $m_i \ll E_i$, in all known production processes. Therefore, the energy can be Taylor approximated to the first order with: $E_i = |\mathbf{p}|\sqrt{1 + (m_i/\mathbf{p})^2} \approx |\mathbf{p}| + \frac{1}{2} m_i^2/|\mathbf{p}| - \ldots$. With this knowledge equation 2.9 can be transformed and the probability expressed as

$$P(\nu_\alpha \longrightarrow \nu_\beta) = \sum_{i,j} U_{\alpha i} U^*_{\beta i} U^*_{\alpha j} U_{\beta j} \exp\left(-i \frac{m_i^2 - m_j^2}{2|\mathbf{p}|} t \right) . \tag{2.10}$$

Equation 2.10 nicely shows the importance of the neutrino masses for the neutrino oscillations. The squared mass differences $\Delta m^2_{ij} = m_i^2 - m_j^2$ between the neutrinos determine the oscillation period. Additionally, this equation reveals that the three neutrino masses are neither equal nor all zero. In this case the time dependence would vanish, leaving only the matrix elements U, which are constant over time. This concludes if neutrinos would have equal or no masses, no neutrino oscillation could be observed.

Since neutrino masses and therefore their differences are fairly small compared to their momentum $|\mathbf{p}|$, the oscillation period becomes macroscopic and can be observed over several kilometers. Figure reffig:Osci shows the observation probability of an initial electron neutrino in vacuum. The probability to detect an electron neutrino is plotted in black, the probabilities to detect a muon or a tau neutrino are displayed in blue and red. On the x axis the distance to energy relation L/E is used. This relation is handy to compare different experiments. Additionally, the relation naturally appears in the probability equation 2.10. The exponent of the exponential function in this equation is $-i\frac{t\Delta m^2_{ij}}{2|\mathbf{p}|}$. As neutrinos can be considered highly relativistic $m_i \ll E_i$, $|\mathbf{p}|$ can be approximated with $|\mathbf{p}| = \sqrt{E^2 + m^2} \approx E$. Additionally, the time of detection $t = L/v$ with the traveled distance L and the speed v, can be approximated as $t \approx L/c = L$. These approximations give the new exponent $(-i\frac{\Delta m^2_{ij}}{2} \cdot \frac{L}{E})$, which already holds the ratio L/E.

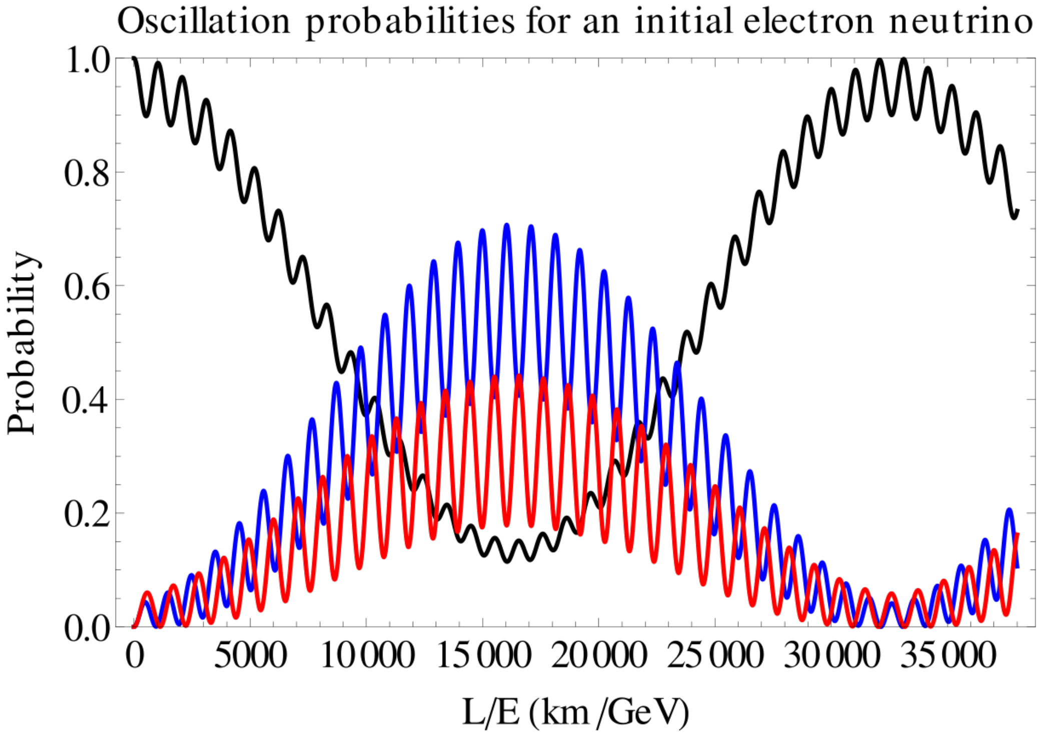


Figure 2.3: Neutrino oscillations in vacuum in regard of the distance to energy relation L/E. The propabilities to detect an electron, muon or tau neutrino are displayed in black, blue and red [13].

2.2.1 Mikheyev-Smirnov-Wolfenstein Effect

With the assumption $E^2 = \mathbf{p}^2 + m_i^2$ the neutrino has been considered to be a free particle, but this is only true in vacuum. Even though the neutrino only couples weakly, it still interacts with matter. This changes neutrino oscillation in matter compared to in vacuum. Matter contains electrons e^-, neutrons n and protons p, which all couple to the weak force, just like the neutrinos. Neutrinos interact via charged or neutral current, which is mediated with $W^\pm$ and Z^0 bosons, respectively. Neutral current is an elastic scattering process, which is possible for all the three neutrino flavors. The process is displayed in figure 2.2a, the Z^0 boson couples to any other weakly interacting particle like quarks contained in e.g. p or n, or e^-.

Charged current interactions with matter only occur for the electron neutrino. This naturally appears because the neutrino only interacts with leptons of the same generation. As matter usually contains electrons e^-, the electron neutrino ν_e can couple to it. While theoretically this process is also possible for μ and τ neutrinos, muons and taus do not naturally occur in matter. Therefore, only the elastic scattering of the electron neutrino with the electron is relevant for neutrino oscillation in matter, it is displayed in figure 2.2b. Through this process electron neutrinos ν_e experience, in contrast to the other neutrino flavors, coherent forward scattering, which changes the relationship between the phases in neutrino oscillation. This effect is called MSW effect and was firstly described in 1986 [14]. As this process strongly relies on the presence of electrons, it is especially strong in matter with a high electron density.

Here the Mikheyev-Smirnov-Wolfenstein (MSW) is introduced shortly, it is for example described in more detail in paper [15].

2.3 Open Questions in Neutrino Physics

In this section the open questions of neutrino physics are discussed. The mass hierarchy and the CP violation are introduced in detail. Afterwards, the neutrino oscillation parameters are discussed.

There are more open questions in neutrino physics, which are only briefly mentioned here, being the total neutrino masses and the Dirac property of neutrinos.

The total neutrino masses for the three known flavors are not yet known, but restricted by upper limits. One experiment working on the total mass of the electron anti-neutrino is KATRIN [5]. Neutrinos only interact weakly and the weak force only couples to left-handed neutrinos and right-handed anti-neutrinos. Thus, the right-handed neutrinos and left-handed anti-neutrinos do not interact with any known force. These neutrinos could be sterile [16]. Another option is, that the neutrinos are their own anti-particles and thereby Majorana instead of Dirac particles. More about Majorana particles can be found in reference [12].

2.3.1 Mass Hierarchy

As introduced in the last section, the squared mass differences $\Delta m_{ij}^2 = m_i^2 - m_j^2$ of the neutrino mass eigenstates are crucial for the neutrino oscillation. But with the mass differences, measurable through the oscillation probability the question arises, which mass eigenstate is the heaviest and which the lightest one. To determine the mass hierarchy is an important step for further precision measurements of neutrino oscillation parameters. As the mass hierarchy is not yet known, all nowadays experiments are analyzed for both cases. A set mass hierarchy would eliminate uncertainties on the values of the CP phase δ, the mass difference Δm_{31} and the mixing angles θ_{13} and θ_{23}. This would improve future experiments, as these are tuned to ideally measure the range in question.

The mass differences for the three mass eigenstates have been measured to be quite different in size, with $\Delta m_{21}^2/\Delta m_{31}^2 \approx 3\,\%$. Due to MSW and matter effects the sign of Δm_{21}^2 was fixed to be positive. But this was not yet possible for the much bigger Δm_{31}^2. The mass hierarchy could also be determined by precisely measuring Δm_{31}^2 in comparison to Δm_{21}^2 and Δm_{32}^2, but no experiment archived a sufficient precision yet. While it is certain, that the mass eigenstate ν_1 is lighter than ν_2, it stays unclear whether ν_3 is the heaviest or the lightest mass eigenstate. The two possibilities are:

- normal hierarchy (NH) with $m_{\nu_1} < m_{\nu_2} < m_{\nu_3}$ and $\Delta m_{32}^2 > 0$. This order is called normal as the lightest state belongs to the mass eigenstate ν_1.

- inverted hierarchy (IH) with $m_{\nu_3} < m_{\nu_1} < m_{\nu_2}$ and $\Delta m^2_{32} < 0$. Here the lightest mass eigenstate is ν_3.

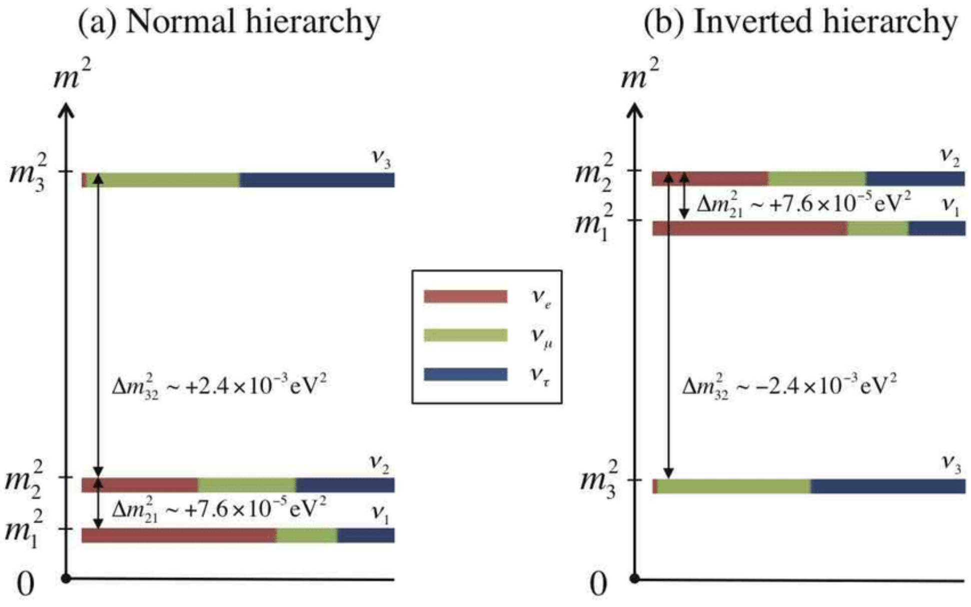

(a) Normal hierarchy
m^2
m_3^2
ν_3
$\Delta m^2_{32} \sim +2.4\times10^{-3}\text{eV}^2$
m_2^2
ν_2
$\Delta m^2_{21} \sim +7.6\times10^{-5}\text{eV}^2$
m_1^2
ν_1
0
ν_e
ν_μ
ν_τ
(b) Inverted hierarchy
m^2
m_2^2
ν_2
$\Delta m^2_{21} \sim +7.6\times10^{-5}\text{eV}^2$
m_1^2
ν_1
$\Delta m^2_{32} \sim -2.4\times10^{-3}\text{eV}^2$
m_3^2
ν_3
0

Figure 2.4: The two possible mass hierachies normal and inverted, from [17].

Both hierarchies are displayed in figure 2.4. The sign of Δm^2_{31} could in future be measured with a long or medium baseline experiment, by analyzing the oscillation propability distributions. These baselines would be around $\leq 50\,\text{km}$. For reactor neutrinos within the energy range of $2 - 8\,\text{MeV}$ the range of $\approx 6000 - 25000\,\text{km/GeV}$ [18] in figure 2.3 is covered. In this area the probability distributions for detecting a muon or tau neutrino are maximally shifted to each other and the probability to detect an electron neutrino from an initial electron neutrino has its minimum.

2.3.2 CP Violation

One open problem in cosmology, is the matter-anti-matter asymmetry. Up to nowadays knowledge the big bang should have created the same amounts of matter and anti-matter. The exact same amount of matter and anti-matter should have annihilated. But still there is matter left in the universe. This imbalance could be explained by CP violation. The CP violation in the quark sector is only very small, it cannot be alone responsible for the asymmetry. In theory there are two more sectors where CP violation is predicted. One is the strong interaction and the other one is the leptonic sector of the weak interaction.

Whether or not the CP is violated in the leptonic sector as well, is not yet known. In the PMNS matirx the CP violating phase is already incorporated as δ. Neutrino oscillation experiments can measure the CP violating phase over the difference in the oscillation spectra of neutrinos and anti-neutrinos. As displayed in section 2.3.3 the latest measurements suggest violated CP, but no sufficient significance has been reached yet.

2.3.3 Current Status of Neutrino Oscillation Parameters

The oscillation parameters are the three mixing angles, the squared mass differences and the CP violating phase. These are observed with different experiments and methods, which are shortly mentioned next.

The KamLAND experiment measured the parameters θ_{12} and Δm^2_{21} with good precision [19] and was able to measure the sign of Δm^2_{21} to be positive. These parameters do not depended on the mass hierarchy.

θ_{13} was firstly measured to be non-zero by Double Chooz in 2012, with a significance of 2.6 σ [20]. Daya Bay and Reno confirmed this results with a significance $> 5\sigma$ [21], [22]. Both experiments observe θ_{13} over the disappearance of reactor neutrinos.

θ_{23} is under observation in a number of long baseline experiments, like Noνa and T2K [23]. Noνa is additionally measuring Δm_{32} [24]. Δm_{32} is the sum of Δm_{21} and Δm_{31}, it is not explicitly listed in table 2.3. The mass difference Δm_{31} is the biggest value of the three mass differences, displayed in figure 2.4.

The JUNO experiment plans to determine the mass hierarchy. The energy resolution is planned to be able to distinguish between the neutrino probabilities for normal and inverted hierarchy [1]. JUNO is described in detail in section 3.3. Additionally, precision measurements of Δm^2_{21} and θ_{12} are predicted.

T2K and Noνa additinally observe the CP violating phase δ. To obtain measurements of δ four oscillation channels were analyzed: $\nu_\mu \longrightarrow \nu_\mu$, $\nu_\mu \longrightarrow \nu_e$, $\overline{\nu}_\mu \longrightarrow \overline{\nu}_\mu$ and $\overline{\nu}_\mu \longrightarrow \overline{\nu}_e$. The obtained distributions are fitted to a function describing the expected number of events, with degrees of freedom for each oscillation parameter and all systematic uncertainties. Additionally, the oscillation parameters were constrained by data. These analyzes were able to exclude the interval of $\delta_{CP} \in [-3.41, -0.03]$ for the normal and $[-2.54, -0.32]$ for the inverted mass hierarchy [25].

All these measurements are brought into context with analyzes and fits of the data. Here the results of the latest global fit from 2020 [26] are shown.

For this analysis the data from T2K till December 2019, the latest Noνa statistics, and the data from Daya Bay, RENO, IceCube, DeepCore, SNO, Gerda, CUORE and KamLAND-Zen have been used. The results are displayed in table 2.1. In the table the mass differences between the neutrino mass eigenstates Δm_{21} and Δm_{31}, the mixing angles θ_{12}, θ_{13} and θ_{23} and the CP violating phase δ are listed. All values despite Δm_{21} and θ_{12} are fitted for the NH and IH. This refers to the normal and inverted mass hierarchy, which is introduced in section 2.3.1. For every parameter the best fit $\pm\sigma$ and the ranges with the significance of two and three sigma are given.

As described for the CP violating phase, neutrino oscillation parameters are often measured by

Table 2.1: Oscillation parameters from the global fit 2020 [26]. NH and IH reveres to the normal and inverted mass hierarchy.

parameter	best fit $\pm\sigma$	2σ range	3σ range
$\Delta m^2_{12}[10^{-5}\text{eV}^2]$	$7.50^{+0.22}_{-0.2}$	7.11 – 7.93	6.94 – 8.14
$\lvert\Delta m^2_{31}\rvert[10^{-3}\text{eV}^2]$ (NH)	$2.56^{+0.03}_{-0.04}$	2.49 – 2.62	2.46 – 2.65
$\lvert\Delta m^2_{31}\rvert[10^{-3}\text{eV}^2]$ (IH)	2.46 ± 0.03	2.40 – 2.52	2.37 – 2.55
$\sin^2\theta_{12}/10^{-1}$	3.18 ± 0.16	2.86 – 3.52	2.71 – 3.70
$\sin^2\theta_{32}/10^{-1}$ (NH)	$5.66^{+0.16}_{-0.22}$	5.05 – 5.96	4.41 – 6.09
$\sin^2\theta_{32}/10^{-1}$ (IH)	$5.66^{+0.18}_{-0.23}$	5.14 – 5.97	4.46 – 6.09
$\sin^2\theta_{13}/10^{-2}$ (NH)	$2.225^{+0.055}_{-0.078}$	2.081 – 2.349	2.015 – 2.417
$\sin^2\theta_{13}/10^{-2}$ (IH)	$2.250^{+0.056}_{-0.076}$	2.107 – 2.373	2.039 – 2.441
δ/π (NH)	$1.20^{+0.23}_{-0.14}$	0.93 – 1.80	0.80 – 2.00
δ/π (IH)	1.54 ± 0.13	1.27 – 1.79	1.14 – 1.90

comparing the measured neutrino flux of a distant source with the predicted. Both distributions rely on the amount of neutrinos produced and the ratio of neutrinos which oscillate into a different flavor along the way. The prediction of the neutrino flux depends on the emission and the oscillation probability, which depends on the three mixing angles, the CP phase, the squared mass differences and the density of the matter along the way. Improving the precision of these parameters, would improve analyzes for all oscillation parameters, for this type of experiments.

The term $|\Delta m_{31}|$ is ≈ 30 times larger than the other mass terms. Thereby it has a comparably large impact on prediction of the probability. Additionally, determining Δm^2_{31} with precision, would determine the mass hierarchy. The CP phase defines how particles behave in comparison to anti-particles. A determined CP phase would additionally improve the calculations of the probability and thereby the analyzes on the other parameters. Upcoming experiments like JUNO [1] and DUNE [27] plan to improve the measurements on Δm^2_{31} and δ_{CP} .

Chapter 3

Neutrino Detection with Liquid Scintillators

In this section neutrino detection with liquid scintillators is described. The JUNO experiment is introduced as an example and the Theia experiment as an outlook to future neutrino experiments.

3.1 Liquid Scintillators

Scintillation describes the feature of a material to emit light when a charged particle or a high-energy photon passes through. The molecules along the path get exited due to scattering process. When they deexit the excess energy is emitted as photons. The photons are emitted in a random direction, after a random time. The probability of the deexitation of an electron is given by the decay function. It is defined with the mean lifetime τ as

$$\phi(t,\tau) = \frac{1}{\tau} \cdot \exp\left(\frac{t - t_0}{\tau}\right). \tag{3.1}$$

The light yield is defied by the luminosity per length $\frac{d\mathcal{L}}{dx}$, in Birks theory [28]

$$\frac{d\mathcal{L}}{dx} = \mathcal{L}_0 \cdot \frac{dE/dx}{1 + kB\frac{dE}{dx}}, \tag{3.2}$$

with the energy E. The variable kB depends on the used material. Equation 3.2 reveals, that the light yield depends on the energy of the particle. The light yield is not linear due to effects like quenching, which reduce the luminosity.

Scintillator materials are normally clear so the emitted photons can travel through the material. On the outer edges of the material the light can be detected with the help of Photomultiplier tubes (PMTs). In the simplest case a scintillation detector can be used as a radiation counter. More sophisticated setups allow the reconstruction of the particles path with help of several PMTs at different positions around the scintillation volume.

One common material is linear alkyl benzene (LAB) [29]. The characteristic wavelength of the emitted photons is 280 nm. LAB provides many benefits, it has a high optical transparency, a high light yield and is comparatively cheap. Additionally, other molecules can be dissolve in the scintillation liquid, which act as wavelength-shifters. In this case they absorb photons from the LAB and reemit the gained energy as photons with a different wavelength. This method is used to adjust the emitted light to best fit the sensitivity range of the used PMTs. For a combined medium the deexitation probability, introduced in equation 3.1, becomes the sum of several exponential functions. Each represents one of the components in the mixture. The summands are weighted by the ratio of these components.

The size of a liquid scintillation detector is limited by the transparency of the used medium. The transparency depends on the attenuation length, which describe the path length of a particle in a medium, after which the probability that the particle is not absorbed is $1/e$. For liquid scintillators a reasonable attenuation length is $\approx$ 20 m. Water-based liquid scintillators reach a much higher attenuation length, one application is discussed in section 3.4.

As liquid scintillation detector (LSD) detect charged particles or high-energy photons, it cannot directly detect neutrinos. Neutrinos only interact weakly and can thereby in a LSD only be detected indirectly over neutrino interactions.

3.2 Neutrino Sources

To detect neutrinos a neutrino source is needed. In experiments five different types of neutrinos are mainly used. These are:

- **Solar neutrinos** (ν_e) are emitted by the sun, which only emits electron neutrinos from fusion processes.
- **Supernova neutrinos** are emitted in supernova explosions, where all flavors of neutrinos are generated.
- **Atmospheric neutrinos** (ν_e, ν_μ, $\overline{\nu_e}$, $\overline{\nu_\mu}$) are produced in interaction of cosmic rays with atmospheric atoms.
- **Reactor electron antineutrinos** ($\overline{\nu_e}$) are produced in fission processes.
- **Accelerator neutrinos**(ν_e, ν_μ, $\overline{\nu_e}$, $\overline{\nu_\mu}$) are produced in fixed-target experiments with proton beams.

Additionally, geoneutrinos and extra-galactic neutrinos have been observed, but are not used as main source for neutrinos in experiments. Geoneutrinos come from radioactive decays in the earths interior.

In this work only the reactor neutrinos are described in detail as these are used to observe neutrino oscillations with JUNO.

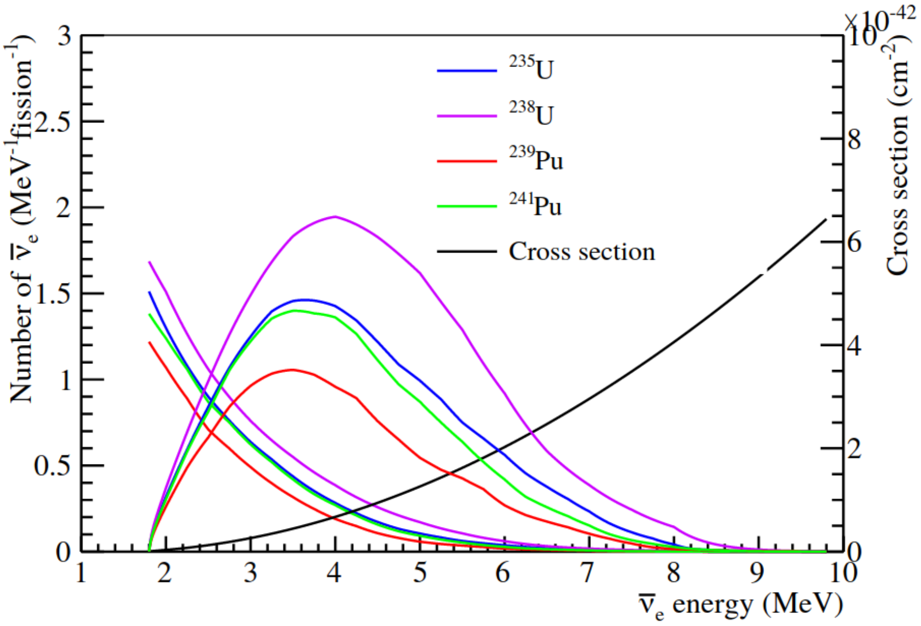


Figure 3.1: Reactor anti-electron neutrino energy distribution for the different initial atoms [31] for a commercial reactor. The decreasing distribution for all four initial atoms display the neutrino flux generated by nuclear reactors. The black curve shows the cross section for the inverse β decay. The mountain like distributions show the energy distribution of reactor neutrinos, weighted with the cross section.

3.2.1 Reactor Neutrinos

Reactor neutrinos are produced in nuclear reactors, where energy is released from fission. In nuclear reactors often neutron induction is used to trigger the decay of the fuel nuclei. In modern reactors the elements U^{235}, U^{238}, P^{239} and P^{241} are used as fuel atoms, these are all unstable. When a free neutron is absorbed by a nucleus, which brings the nucleus into an excited state, the nucleus decays shortly afterwards. Unstable fragments of the nucleus decay with the β - decay by emitting an electron anti-neutrino $\overline{\nu_e}$. For the example U^{235} approximately six electron anti-neutrinos are emitted per fission. One nuclear core of a fission reactor emits thereby in total approximately $10^{20}\overline{\nu_e}/s$ [30]. In figure 3.1 the energy distribution of reactor electron neutrinos, weighted with the cross section of the inverse β decay is displayed. The energy varies between $\approx$ 1.8 and $\approx$ 9 MeV. The peak is for the different fuel atoms between 3 and 4 MeV. The inverse β decay is used as detection reaction in LSD. Therefore, the weighted energy distribution is the energy distribution of the reactor neutrinos detected in an LSD.

3.3 JUNO

The Jiangmen Underground Neutrino Observatory (JUNO) is a liquid scintillation neutrino detector in Jiangmen, China. JUNO is currently under construction and aims to start the first

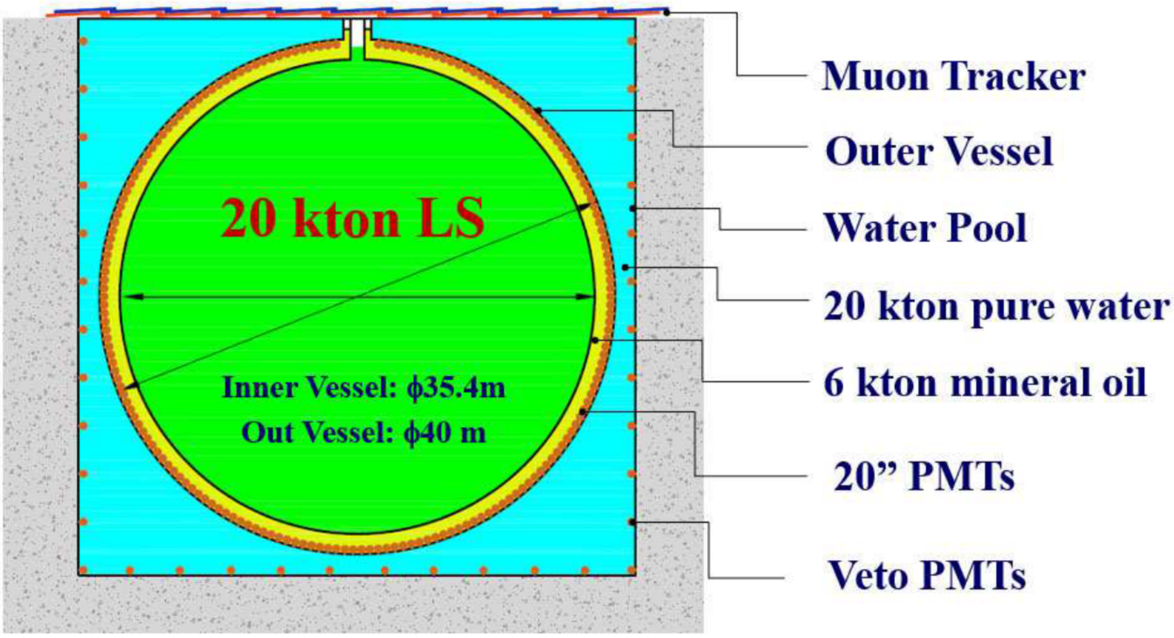


Figure 3.2: Schematic JUNO detector [1], displaying the detection volume with several detectors.

runs in 2021. This section in mainly base on JUNOs technical report[1]. The JUNO experiment consists of three different detectors: the main liquid scintillation detector (LSD), a muon tracker (MT) and a water Cerenkov detector (wCD).
The LSD itself is a spherical volume with a diameter of 35.4 m filled with 20 kt linear alkyl benzene (LAB). Around the scintillation chamber PMTs are mounted on a steel structure. Three different PMT types are used. This results in a total coverage of approximately 75 % and a designed energy resolution of $3\,\%/\sqrt{E[\mathrm{MeV}]}$. The setup is schematically displayed in figure 3.2. The scintillation volume is submerged in a the water tank, which is marked in blue. To further suppress the muon background JUNO is placed roughly 700 m under ground.
The main LSD is submerged in a cylindrical water pool. It serves two purposes, on the one hand the water shields the LSD from radioactive background naturally occurring in the surrounding rocks. On the other hand it is a Cerenkov light detector, which can detect cosmic muons entering the main detector from any direction. The Cerenkov light is detected by 1600 20-inch PMTs, which are mounted on the outer walls of the water basin. On top the muon tracker is positioned. This tracker uses plastic scintillator strips and tracks passing muons. Both muon detectors are used as vetos to reduce background.
JUNO's main goal is to determine the mass hierarchy. Therefore, electron anti-neutrinos $\overline{\nu_e}$ from two reactors are observed. The amount of electron anti-neutrinos $\overline{\nu_e}$ produced in the reactors is known. The amount of missing electron anti-neutrinos $\overline{\nu_e}$ tells how many neutrinos changed flavor and thereby the oscillation parameters can be measured. The reactors are both in approximately 53 km distance from JUNO, with the mean neutrino energy of ≈ 3 MeV. The energy spectrum of the reactor neutrinos which will be detected at JUNO is displayed in

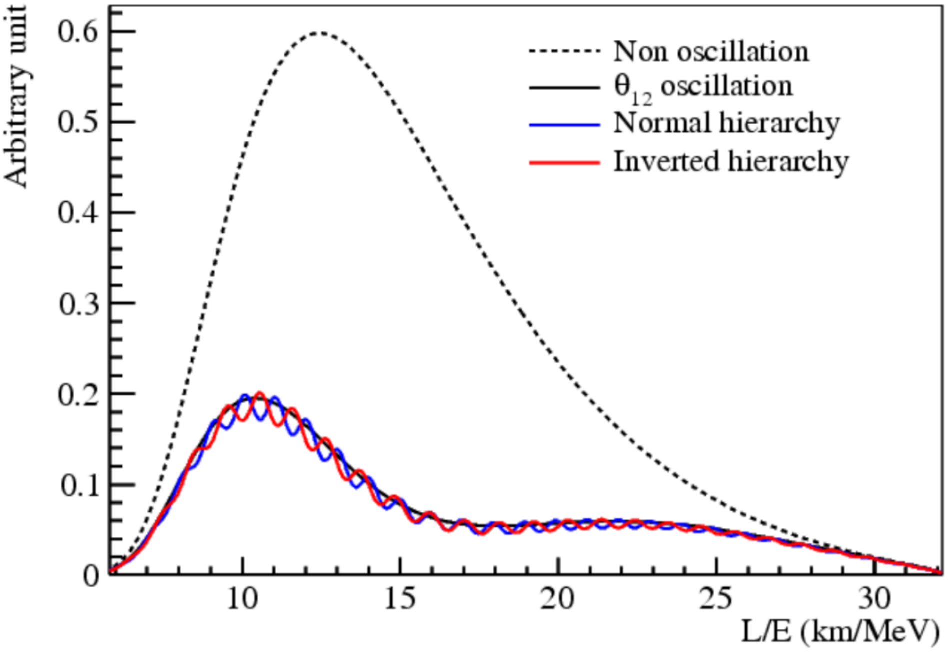


Figure 3.3: The relative shapes of the reactor anti-neutrino distribution for the two possible mass hierarchies.

figure 3.1. As JUNO uses the inverse β decay as the detection process, anti-electron neutrinos within the energy range of $\approx 2 - 9\,\mathrm{MeV}$ can be detected.

3.3.1 Research Agenda

JUNO's main research goal is to determine the neutrino mass hierarchy and measure the neutrino oscillation parameters $sin^2(\theta_{12})$ and Δm^2_{12}.

JUNO is measuring the reactor anti-neutrino spectrum within the energy range of $\approx 2 - 9\,\mathrm{MeV}$ at a medium baseline of 53 km. JUNO plans to observe the shape of the reactor anti-neutrino distribution with regards to the ratio of distance to anti-neutrino energy L/E. In figure 3.3 the shape of this distribution is plotted for the normal and inverted mass hierarchy. With JUNOs excellent energy resolution, the two possible distributions can be distinguished and the mass hierarchy thereby determined. JUNO is predicted to be determine the mass hierarchy with a $3\,\sigma$ uncertainty in approximately six years.

Additionally, JUNO is able to precisely measure a number of neutrino oscillation parameters, due to its high energy resolution. The JUNO collaboration claims to measure $sin^2(\theta_{12})$ and Δm^2_{12} with a precision better than 1 %.

Besides the reactor neutrinos, JUNO will also detect neutrinos from other sources. JUNO will be able to observe neutrino bursts from nearby supernovas, this is especially interesting for low-energy neutrino and astro-physicists. Additionally, these measurements can contribute to multi-messenger analysis of cosmic objects. Moreover, JUNO is sensitive to solar, atmospheric and geoneutrinos, giving in total a broad range of interesting observations for neutrino, astro- and geophysicists.

3.3.2 Signal

All three neutrinos and their anti-particles can be captured through different processes with different signals. Here electron anti-neutrinos $\overline{\nu_e}$ fare detected.
The electron anti-neutrino $\overline{\nu_e}$ can, as all neutrinos, only be measured over interactions inside the detector. The detection reaction for this case is the inverse β decay (IBD):

$$p + \overline{\nu_e} \longrightarrow n + e^+ \tag{3.3}$$

The electron anti-neutrino $\overline{\nu_e}$ interacts with a proton, which results in a neutron n and a positron e^+. The positron quickly deposits its energy in the detector and annihilates prompt into two 511 keV photons. The neutron scatters in the detector until being thermalized, thereby it deposits its energy inside the detector. Afterwards, the neutron is captured by a proton in a nucleus. Due to the extra neutron the nucleus is highly excited and emits discrete energy after a certain time. Both time and energy depend on the nucleus. In the JUNO detector the neutron capture emits 2.2 MeV after 200 μs. This characteristic coinciding measurements reduce the background. Most of the energy of the reactor anti-neutrino is transferred to the positron. The energy spectrum described in figure 3.3, can be obtained from the reconstructed anti-neutrino energies.
At JUNO 83 inverse β decays/day, from reactor neutrinos, are expected.

3.3.3 Background

One big background in scintillation detectors are cosmic muons. On sea level approximately 0.001 m^2/s cosmic muons pass. Muons cause ionization in the scintillator and are thereby one of the biggest backgrounds. Many scintillation detectors are placed underground to reduce the cosmic muon flux. Additionally, muon detectors can be placed around the scintillator, as at JUNO. With the timing of cosmic muons entering the detector, affected measurements can be vetoed. But the muons are no direct background for reactor neutrinos. Still they increase considerably the dead time of a scintillation detector. At JUNO the muon rate is expected to be $\approx$ 3.5 Hz with an average energy of 215 GeV.
The main backgrounds of JUNO considered in its technical report [1] are the following:

- **Accidental background**: These are independent incidents, which mimic in combination a IBD signal. These incidents can be caused by radioactivity, cosmic isotopes and spallation neutrons. The expected rate is $\approx\ 5.7 \cdot 10^4$ /day.
- **Geoneutrinos**: Electron anti-neutrinos from the earth's interior, are expected for the JUNO site with a rate of $\approx$1.5 /day.

- **Fast neutrons**: Bypassing cosmic muons produce energetic neutrons. This can lead to a fast neutron background, if they scatter off a proton and are afterwards captured. The estimated rate is ≈ 0.1 /day.

- **^{13}C(α,n) ^{16}O background**: Radioactive decays of U and Th can emit alphas, which could interact with ^{13}C in the liquid scintillator, this could lead to a correlated background, with a rate of ≈ 0.01 /day.

- **Cosmogenics**: Are explained in more detail up next, as they are relevant for the main motivation of this work.

Cosmogenics

Energetic muons often produce electromagnetic and hadronic showers along the path. Inside the showers radioactive isotopes can be produced. ^{12}C interacts with the electromagnetic and hadronic processes, thereby radioactive isotopes with $Z \leq 6$ are produced. These can decay by emitting a neutron and a positron or electron. As the neutrinos are identified over an annihilating positron and a captured neutron, the neutrino cannot be distinguished from this decay channel, if the timing matches.

The radioactive isotopes ^{9}Li and ^{8}He follow this behavior at JUNO. They can decay over a neutron and the β decay, with a half-life of 0.178 s and 0.119 s, respectively. There are more isotopes at JUNO, which lead to this type of background, but they have a much smaller contribution and are therefore not explicitly mentioned here. At JUNO the rate for this type of background is estimated to be 84 /day. This magnitude is enormous as the expected rate of IBD from the reactor neutrinos is only 83 /day.

The decay of the elements Li^{9} and He^{8} is the second biggest background for the reactor neutrinos at JUNO.

3.3.4 Background suppression methods

In the JUNO detector a huge number of background events is expected, with only a small number of reactor neutrino events, which need to be identified. To suppress the background the events have to fit into certain constraints. These describe the characteristics of the signal event, in order to cut away the background.

To keep only events with the reactor neutrino characteristics the following cuts are used:

- **fiducial volume cut**: $r < 17$ m, this cut is included to suppress the accidental backgrounds,

- **prompt energy cut**: 0.7 MeV< E_p <12 MeV,

- **delayed energy cut**: 1.9 MeV< E_d <2.5 MeV,

- **time cut**: between prompt and delayed signal: $\Delta t < 1$ ms,

- **distance cut**: between prompt and delayed signal: $R_{p-d} < 1.5\,\mathrm{m}$.

Additionally, a muon cut is used to eliminate the fast neutron and the cosmogenic background. Therefore, the surrounding muon trackers are used for a **muon veto** with the following criteria:

- muons detected in the water veto the liquid scintillator for 1.5 ms,
- well tracked muons in the water Cerenkov detectors and the central detector lead to a veto within the radius r of $r_\mu < 3\,\mathrm{m}$ around the track for $t_\mu < 1.2\,\mathrm{s}$,
- mouns which can not be tracked veto the whole sphere for 1.2 s.

The efficiency of these cuts is listed in table 3.1. As important it is for a cut to cut out as much of the background as possible, it is additionally extremely important to keep the signal events, especially for rare events like neutrino processes.

Accidental background are the main background source. Most of them are discriminated by the fiducial cut, diminishing it to about 400 events per day. The distance cut and the muon veto decrease the rate further.

Table 3.1: Different background sources and their occurrences with the different suppression methods are displayed. The prompt energy, delayed energy time and distance cut are combined here, with the name "T,E and d cut". The background source minor is a summation of geoneutrinos, fast neutrons and ^{13}C(α,n) ^{16}O background. All rates are displayed in events per day.

Method	**IBD efficiency**	**IBD [/day]**	**^{9}Li + ^{8} He [/day]**	**Accidental [/day]**	**Minor [/day]**
no	-	83	84	$\approx 5.7 \cdot 10^4$	1.7
Fiducial cut	91.8%	76	77	410	1.6
T, E and d cut	95.7%	73	71	1.1	1.5
Muon veto	83%	60	1.6	0.9	1.3
Combined	73%	60		Σ 3.8	

The cosmogenics are predicted to occur with a rate of 84 /day. With the help of all cuts this rate is lowered to 71 /day. The muon veto eliminates further cosmogenics, as these are triggered by showers made by passing cosmic muons. This lowers the expected cosmogenics rate further. After all cuts and vetos the rate is 1.6 /day.

The minor backgrounds are also lowered from ≈ 1.7 to 1.3 /day, with help of these background suppression methods.

The downside of these cuts is, that also signal events are cut of, if they fit into the given constraints. This lowers the expected IBD rate from 83 /day to only 60 /day after all cuts and vetoes. The biggest chunk here takes the muon veto with 13 events being missed due to the muon veto and corresponding dead time of the liquid scintillator detector. This is an IBD efficiency rate of 83 % for the muon veto and a combined efficiency of 73 % for all cuts and vetoes.

With the IBD rate of 83 /day and the described background and cuts, JUNO is expecting to falsely identify 3.8 background events per day as reactor neutrinos, this is a ratio of 6 % of the measured events.

Potential of further Background suppression methods

As illustrated the background suppression is a crucial part for JUNOs precision measurements. In the implemented cuts and vetoes 23 out of 83 IBD events/day, which are 27.7%, are lost. The lowest efficiency on the IBD has the muon veto. With more sophisticated suppression methods the efficiency of JUNO could be increased.

One promising method is the Quadratic Reconstruction (QR) method by D. Meyhöfer [2]. Heavily simplified speaking, this method basically reconstructs the detected scintillation photons to their point of emission. With the sum of all reconstructed emission points, showers can be identified and located inside the detector.

The QR gives the opportunity to locate over 80 % of those showers, where > 400 MeV energy was deposited, with a precision of ±0.35 m. With a background suppression based on this method, JUNO could detect ≈ 10 % more signal. This increase is tremendous.

The here presented work basically follows the same goal, developing a method to locate showers on muon tracks in a liquid scintillator, with the motivation of background suppression for reactor neutrino detection. But in contrast to the QR, which uses classical computation techniques, here machine learning methods are used. Working on the same problem as an already successful method is in no way irrelevant. On the one hand having a second method, which independently comes to the same results is a validation of measurement or reconstructions. On the other hand, the QR works with look up tables. This is a common method to fasten up computations on complicated calculations. But it has the downside, that theses tables can use lots of memory, need to be kept up to date or newly made for new applications. In contrast to machine learning techniques, these typically do not need much memory space. No lookup tables need to be computed or fostered. Additionally, taken data can be used to improve the algorithm along the way, improving the prediction with time of use. Furthermore, the machine learning methods are versatile and with some more new training one architecture can be recycled for a similar application. All in all machine learning techniques appear to have great benefits for a reconstruction task like this.

3.4 Future Detectors - Theia

Even tough JUNO is still under construction, there are plans on new neutrino detectors, with superior features. Especially the latest developments in liquid scintillators and photo detection give new possibilities for neutrino detectors. Theia [32, 33] is a planned experiment at the Sanford Underground Research Facility (SURF) as long baseline experiment. As neutrino source the LBNF neutrino beam is supposed to be used, with a baseline of 1300 km. By combining the benefits of liquid scintillation with water Cerenkov light and a new generation of advanced photo detection methods, Theia is proposed to reach a new level of sensitivity. In the publications [32, 33] different setups are being discussed, a smaller one with a total volume of 25 kt and a fiducal volume of 17 kt (Theia 25), and a bigger one with a total volume of 100 kt and a fiducial volume of 70 kt (Theia 100). Up to now Theia is a proposed future experiment, without a concrete plan for its realization.

3.4.1 Outstanding Features

The concept of Theia is to combine liquid scintillation detector (LSD) with water Cerenkov detector (wCD) benefits in a large detector of several kilotons. The Cerenkov light has the benefit that, due to its distinct pattern, the direction of charged ultra-relativistic particle tracks can be reconstructed. Additionally, water is more transparent than scintillation liquid, which makes larger detectors possible. With Liquid Scintillators a low energy threshold and a good energy resolution can be achieved. To successfully combine Liquid Scintillation and wCD methods a separation of both photon sources is important. Hit pattern, timing and wavelength can be used to separate the two photon sources. Theia proposes two methods to achieve this. On the one hand the waterbased Liquid Scintillator (wbLS) and on the other advanced photo sensors, called Large Area Picosecond Photodetectors (LAPPDs).

WbLS [34] are a combination of water and liquid scintillator. Liquid Scintillator is dissolved in water, the mixing ratio gives the opportunity to tune the medium to best fit the experimental setup and research goals. The wbLS achieve an increased transparency with a time response similar to LSD.

LAPPDs [35] with an area of $20 \times 20\,\mathrm{cm}^2$ and a time resolution of 50 ps are used. Their advantage are the short time resolution and the discrete spatial resolution. LAPPDs are based on a Multi-Channel Plate technology, which allows a spatial resolution of 5 mm. The combination of a great time and spacial resolution make LAPPDs an impressive technology, which enables great opportunities for event reconstruction.

Besides the technological innovations Theia additionally relies on strong reconstruction algorithms. Therefore, classical approaches like the Topological Track Reconstruction [36], as well as machine learning techniques could be used. Using machine learning methods to reconstruct muon tracks, as presented in this work, could become interesting for Theia as well.

3.4.2 Physics Program

Theia is designed to work in three stages. As the properties of the wbLS are dependent on the ratio of water to LS, it can be adjusted to fit the experimental setup. By changing the ratio, Theia can perform in stages observing different research questions. At first Theia is planned to work as a long baseline neutrino experiment, with distance of 1300 km from the LBNF muon neutrino beam at Fermilab. In this period Theia should measure the neutrino oscillation parameters like the CP violation phase δ_{CP} and deliver further measurements pf the mass hierarchy. The sensitivity for δ_{CP} and the mass hierarchy of the Theia 25 setup is predicted to be compatible with the sensitivity of a single DUNE module, under the assumption, that Theia reproduces the detector performance of water Cerenkov Detectors.

Afterwards, by increasing the liquid scintillator concentration Theia would increase its sensitivity in the low energy region. Thereby, Theia would be able to additionally observe solar, supernova and geo-neutrinos. Furthermore, with some adjustments and expansions [32], Theia would be sensitive to observe the neutrinoless $\beta\beta$-decay.

Chapter 4

Graph Neural Networks

Graph neural networks (GNN) are a special type of neural networks, both are introduced in this section. Firstly, neural networks are introduced as they are the basis of GNN. Many methods used for neural networks are required for GNN as well. Afterwards, the GNN are introduced. These offer the possibility to include metrics in the data and relationships between data points can be included in the architectures.

4.1 Introduction into Neural Networks

Neural networks are inspired by brains, which consist of neurons. The foundation for neural networks has already been set in the 1950's, but only with the improvement of computing power they gained the great popularity and importance they have today. This section is based on [37].

4.1.1 Structure

Neural networks consist of artificial neurons. The neuron takes some inputs x_i and produces output y. Therefore, the input is multiplied by weights w_i and summed up, an additional bias b or an activation function f can alter the output. For example absolute could be used as activation function, if only positive output is wanted. These properties are summarized in a schematic node, displayed in figure 4.1. The combination of several neurons forms a neural network. These are structured in layers. Each layer consists of at least one neuron. The output of the neurons in one layer becomes the input of the neurons in the next layer, while the output of the last layer is the output of the network. The layers between input and output layer are called hidden layers.

A straight forward approach is the *fully connected* network. In this setup all neurons from one layer make the input for all the neurons in the next one. Depending on the task different network styles are used, any combination of layers and neurons is possible. Figure 4.2 shows an example of a fully connected neural network with an input layer, two hidden layers and an

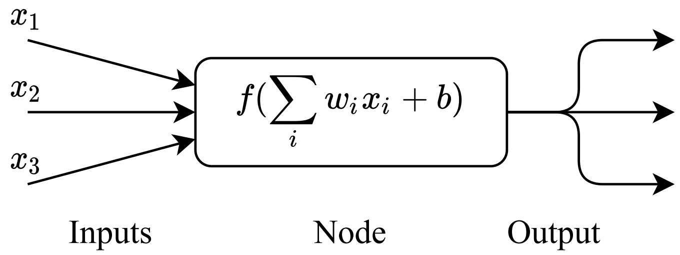


Figure 4.1: One schematic node, the inputs x_i are weighted and summed, a bias b can be added. The sum can be altered by an activation function f, this makes the output of the node. The output can be passed to several other nodes.

output layer. The hidden layers have three nodes each. For the input two nodes are used and for the output one. In the in- and output one node represents one number.

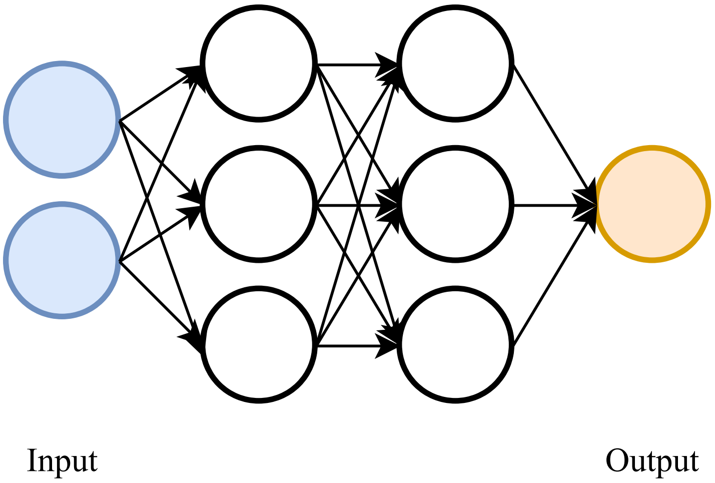


Figure 4.2: Fully connected neural network with two hidden layers, three nodes each. The input layer consists of two nodes, while the output layer has one.

4.1.2 Training

The input of a neural network propagates through the network forming the output in the last layer. While learning all weights and biases in the network are adjusted, till the output matches the label. The label defines the desired output and how well these match is defined by the loss function. During the training the output of the loss function, so called loss, is minimized by adjusting the trainable parameters. The trainable parameters of a neural network are the weights and biases of its nodes.

The loss function used in this work has several components, among them *mean squared error loss* and *cross entropy loss*. Additionally, a geometric loss function is used, which will be introduced in section 4.3.

The mean squared error loss is a widely used loss function. With y as prediction and t as label of N considered samples, it is defined as

$$L_{MSE}(y, t) = \sum_{i}^{N} \frac{(y_i - t_i)^2}{N}. \tag{4.1}$$

This loss function calculates the error of every output in comparison to the label, squares it and takes the mean.

For classification tasks the cross entropy loss can be used. In machine learning classes are often implemented with floats. For example an exclusive binary class system can be implemented with one parameter in label and prediction. This parameter can be defined as 0 for class A and 1 for class B. The network predicts the label $\in [0, 1]$ resembling the affiliation to the classes A or B. The cross entropy loss is defined as

$$L_{CrossEntropy}(y, t) = -\frac{1}{n} \sum_{i} \left[t_i \ln(y_i) + (1 - t_i) \ln(1 - y_i) \right]. \tag{4.2}$$

Here is again t the label and y the prediction, in the class form $y, t \in [0, 1]$ and n the number of samples.

To optimize the networks output to best fit the labels, the biases and weights are gradually altered, by an *optimizer*. There are different optimization methods, but most of the optimizers rely on *gradient descent*. The idea of *gradient descent* is, to alter the weights and biases with the goal of a decrease of the loss function. Or in other words in the direction of declining slope on the loss function. This can be archived with

$$w' = w - \eta \frac{\partial L}{\partial w} \tag{4.3}$$

and

$$b' = b - \eta \frac{\partial L}{\partial b}. \tag{4.4}$$

The derivation of the loss function L with respect to its variables biases b and weights w defines the slope. The new variable, marked with a dash, is given by the original one plus the negative slope times the step size η. η is the learning rate and defines how big the alterations on biases and weights are. The learning rate has to be chosen wisely, as a to big choice may over step the minimum of the loss function, while a small learning rate takes a lot of training time to reach it.

While learning, all weights and biases of all nodes need to be updated. As neural networks can easily hold some thousand nodes, the number of trainable parameters, which are the weights and biases, can be large. To efficiently calculate and apply them, the *backpropagation algorithm* is used. The output a_j^l of one node j in layer l is defined as, the weighted sum of inputs $w_j^l \cdot a_i^{l-1}$

(which are outputs of the layer before) with an additional bias b

$$a_j^l = f(\sum_i w_{j,i}^l a_i^{l-1} + b_j^l) = f(z_j^l). \tag{4.5}$$

z_j^l describes the sum inside the activation function f and will be used later on. The sum sums over all nodes i in layer $l - 1$. In vector representation $\mathbf{a}^l$ denotes all outputs of layer l. The bold variables are vector representations and W^l is the weight matrix

$$\mathbf{a}^l = f(W^l \cdot \mathbf{a}^{l-1} + \mathbf{b}^l). \tag{4.6}$$

As the output of one layer becomes the input for the next one, this equation can describe the whole neural network. The output y of a neural neural network with three layers can be described as

$$\mathbf{y} = f(W^l \cdot f(W^{l-1} \cdot f(W^{l-2} \cdot \mathbf{x} + \mathbf{b}^{l-2}) + \mathbf{b}^{l-1}) + \mathbf{b}^l). \tag{4.7}$$

As defined earlier the loss function L is a function of predicted label y and true label t. Thereby, the loss function $L(\mathbf{y}, \mathbf{t})$ can be derived with respect to the weights w, as needed in equation 4.3 and 4.4. For adapting the network every gradient for every weight and bias would need to be calculated, but the backpropagation algorithm calculates the gradients for the weighted input of each layer. The error is defined as

$$\delta_j^l = \frac{\partial L}{\partial z_j^l} = \frac{\partial L(f(z))}{\partial a_j^l} \frac{\partial f(z)}{\partial z_j^l}. \tag{4.8}$$

Which can be described as δ^1 in vector representation for the layer l

$$\delta^1 = \nabla_{a^l} L \odot f'(\mathbf{z}_j^l), \tag{4.9}$$

here $\odot$ resembles the element-wise product. With this equation the error for the next layer can be defined recursively

$$\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot f'(\mathbf{z}_j^l). \tag{4.10}$$

Back in equation 4.3 and 4.4 the adaptation of the trainable parameters w and b are defined, these can now be converted with the definitions of equation 4.8. Firstly, the adaption for the bias b

$$\frac{\partial L}{\partial b_j^l} = \frac{\partial L(f(z(b)))}{\partial b_j^l} = \frac{\partial L(f(z(b)))}{\partial f(z)} \frac{\partial f(z(b))}{\partial z} \frac{\partial z(b)}{\partial b_j^l} = \delta_j^l. \tag{4.11}$$

As $\frac{\partial z(b)}{\partial b_j^l} = 1$ the bias error resembles the derivation in 4.8. Secondly, the adaptions for the weights w

$$\frac{\partial L}{\partial w_{ij}^l} = \frac{\partial L(f(z(w_{ij}^l)))}{\partial w_{ij}^l} = \frac{\partial L(f(z(w_{ij}^l)))}{\partial f(z)} \frac{\partial f(z(w_{ij}^l))}{\partial z} \frac{\partial z(b)}{\partial w_{ij}^l} = a_j^{l-1} \delta_i^l. \tag{4.12}$$

The derivative of z with respect to w_{ij}^l, gives the factor a_j^{l-1}, which is the output of the node j in layer $l - 1$. Again here the definition of equation 4.8 is used.

The backpropagation algorithm needs the definition of these equations. In stead of calculating every gradient in a network it uses the error δ^l. To define the algorithm only four steps are needed. Firstly, the data needs to be propagated in forward direction through the network to produce the prediction, therefore all nodes in all layer need to be evaluated. Afterwards, the gradients on the prediction need to be calculated with equation 4.9. Additionally, the errors for all layers need to be calculated with equation 4.10, here the recursive definition of the layer error comes handy, as some parts of the equation can be reused from the last layer. Lastly, the weights w and biases b need to be adjusted. As the alteration is defined by the error δ^l, the values can be reused from the last step, as they are components of the error vector of the given layer. With this method much less calculations have to be done compared to calculating every alteration by itself. Thereby, a lot of computing time is saved. The backpropagation algorithm is displayed with pseudocode in algorithm 1.

Algorithm 1: Schematic backpropagation algorithm using the *gradient descent* method to adjust the learnable parameters w and b.

```
backpropagation($x$, $t$):
    Input: data $\mathbf{x}$, label $\mathbf{t}$
    Feed forward:
        calculate the outputs $z^l$ and $a^l$ for all layers $l$ and the prediction $y$
        $\mathbf{z}^l \longleftarrow \mathbf{z}^l = W^l \cdot \mathbf{a}^{l-1} + \mathbf{b}^l$
        $\mathbf{a}^l \longleftarrow \mathbf{a}^l = f(z^l)$
        $\mathbf{y} = f(W^l \cdot f(W^{l-1} \cdot \ldots f(W^0 \cdot \mathbf{x} + \mathbf{b}^0) \ldots + \mathbf{b}^{l-1}) + \mathbf{b}^l)$

    Error on prediction:
        calculate the error of the output of the last layer
        $\delta^l = \nabla_{a^l} L(y, t) \odot f'(\mathbf{z}_j^l)$

    Error on layers:
        calculate the error of the other layers
        $\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot f'(\mathbf{z}_j^l)$

    Update lernable parameters:
        adjust all weights $w$ and biases $b$
        $\mathbf{b}^l \longleftarrow \mathbf{b}^l = \mathbf{b}^l - \eta \delta^l$
        $\mathbf{w}^l \longleftarrow \mathbf{w}^l = \mathbf{w}^l - \eta \delta^l (\mathbf{a}^{l-1})^T$

end
```

4.1.3 Monitoring and Evaluation

During the training, the optimizer can optimize into a training data set specific local minimum of the loss function, which does not represent the data set as a whole. This problem is called *over fitting*. Over fitting is reached, when a network performs better on the training data set than on a comparable unknown data set. In this case, the network does not generalize, but learns specific data characteristics, which are not relevant for the prediction. To avoid this the loss is monitored during the training. To validate the results from training a second data set, the so called validation data set, is used. The network predicts labels for the validation data, which are analyzed with the loss function to obtain the *validation loss*. While the validation loss and the training loss do not diverge, the network generalizes the data.

Neural networks are trained in *epochs*. One epoch resembles working through the data set once. After every epoch loss and validation loss can be compared, to check if over fitting occurs or the loss stops decreasing. In this case, the training can be stopped, because the network generalized the data set with the given constraints.

When a model is trained the results need to be evaluated in order to analyze the prediction strength. As the loss function describes the differences between the prediction and the label, a small loss value is a hint for a good prediction. In general a prediction with a loss value of zero would be perfect for the given constraints.

Depending on the prediction and the reconstruction goals different evaluation methods are useful. For regression tasks a direct comparison of prediction and label is often used. For example, if the goal is to predict coordinates, the distance between reconstructed and label coordinate, could be an appropriate description of the prediction quality.

For classification tasks the quality of the prediction can be analyzed by the reconstruction quality of the classes. One method to evaluate classification algorithms is the ROC-Curve.

ROC Curve

The receiver operating characteristic (ROC) curve is a method , which shows how well two classes are separated. As already mentioned two classes can be described by one parameter, which is 0 for class A and 1 for class B. Neural networks usually do not predict only zero or one, but floats resembling how confident the network is with the class affiliation. Somewhere in the range between 0 and 1 a threshold needs to be set, to define where class A ends and class B begins. To produce the ROC curve the predictions are evaluated for thresholds in range $[0, 1]$, with regard of the ratio of correctly predicted events in the two classes A and B. The results are plotted with the margin of correctly assigned for class A on the x axis and correctly assigned for class B in the y axis. For a perfect separation the curve has a sharp edge in the upper right corner, which makes the area under the curve (AUC) close to 1. If the predicted values are randomly distributed $\in [0, 1]$, the area under the curve becomes 0.5.

The area under the ROC curve is a good method to compare different algorithms or networks with regards to their separation quality of two classes. In figure 4.3 an example ROC curve is shown with the corresponding AUC. The displayed ROC curve shows a rather good separation of the classes A and B.

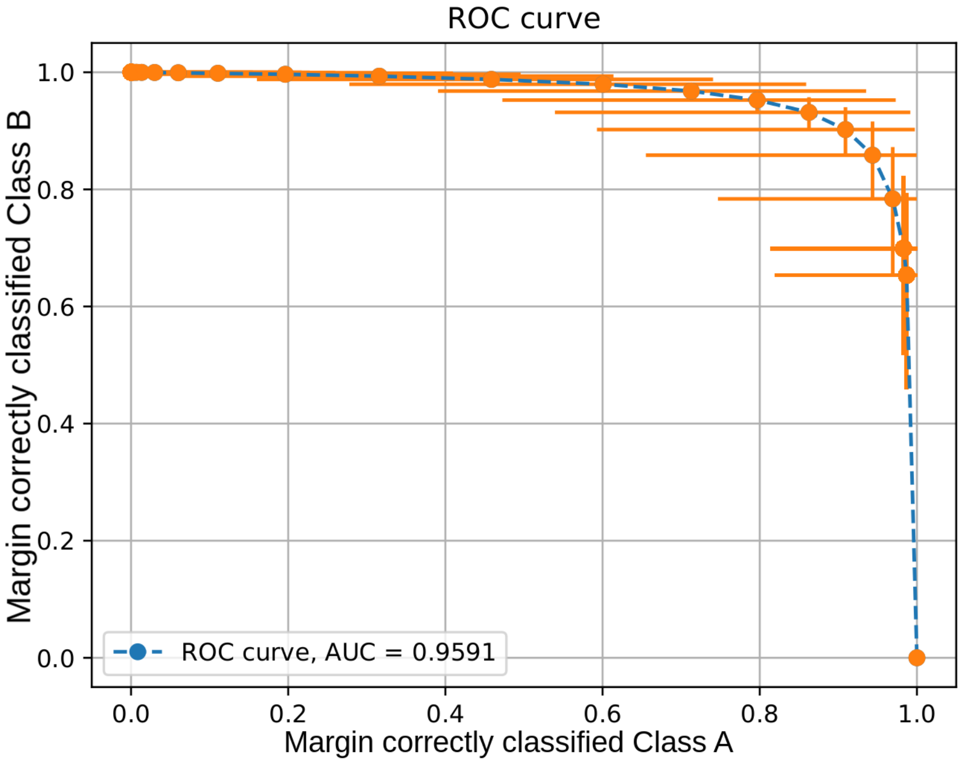


Figure 4.3: A ROC curve showing the separation between class A and B. To produce the ROC curve the predictions are evaluated for thresholds in range $\in [0, 1]$, calculating the ratio of correctly predicted for both classes. These ratios are marked in orange in the plot. The blue line shows the ROC curve which is defined by the ratios. The AUC shows how well the two classes are separated in the prediction.

4.2 Graph Neural Networks (GNN)

Graph neural networks (GNN) were first introduced in 2009 [38]. These are neural networks which use graphs instead of layers of neurons. The benefits of using graphs is their ability to use relationships in the data. This makes graph networks very useful for classification or clustering tasks.

4.2.1 Structure

A graph consists of edges e_k and nodes v_i. Undirected edges have no directional dependency, while directed edges only contribute in one direction. In figure 4.4 a graph with eight nodes is shown. Some edges are undirected, others directed, displayed by single or double headed arrows. One node has a self loop, which is an edge onto itself. This graph could for example represent a social network, with the nodes standing for persons and the edges displaying their relationships.

The data used for graph networks needs to be transferred in the graph structure beforehand. There are many different methods to operate on graphs. In general the features x_{v_i} of a node v_i

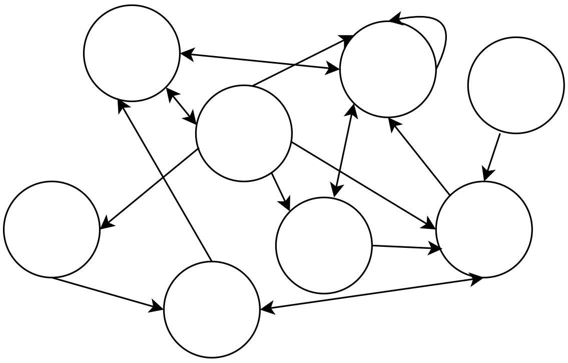


Figure 4.4: Graph with eight nodes and some edges represented by circles and arrows. This graph has directed and undirected edges and one self-loop.

are updated in regards of the connected edges and neighboring nodes. In the simplest case the node features could be updated by forming the weighted sum for the features of all connected nodes. The number of features and the graph can be altered after every iteration. Every new iteration is called layer and can follow different rules, just like with classical neural networks. The data is embedded in the graph structure. Every node holds node features, which resemble data points. A graph holds a set of data points with their connections over edges.
After some layers the data usually needs to leave the graph structure, either it directly gives the output or it is used by other neural network methods, which do not use the graph structure. Therefore, *pooling* methods are used. These need to be invariant under translation. This is important because the order of nodes can differ or be changed during training. Pooling is in general used on the feature space returning one value per feature, regardless of the total number of nodes. Typical pooling methods are maximum, minimum, standard deviation or average.

4.2.2 Graph Convolution Networks (GCN)

Graph convolution networks (GCN) are convolutional networks. Convolution is widely used for neural networks, especially in image recognition. Just like in neural networks the convolution method uses a filter to analyze several nodes, which creates the output of one. In image recognition the filter uses NxM pixels to calculate the output for one of the neurons. Graph networks use their structure in convolution, the neighboring or connected nodes are used to calculate the output.
GCNs have the goal to best reconstruct the label for a given input, after some adjustments in training. For GCNs this is done with graphs, which take as input a feature matrix X in the graph representation. The matrix has the dimensions $N \times D$ for N nodes with each D features. The feature matrix can be understood as input for the graph. Additionally, the graph structure needs to be defined, usually by the number of nodes and the edges connecting them. The

adjacency matrix A describes this. The output of the GCN is Z, with dimensions $N \times F$. F describes the amount of output features of the output nodes N. The output of any given GCN layer l can be described recursively as

$$H^{l+1} = g(H^l, A) \tag{4.13}$$

with $H^0 = X$, $H^L = Z$ and L as the total amount of layers. For different GCN propagation methods different functions g can be used. Here a very simple definition of g is considered

$$g(H^l, A) = f(AH^lW^{)} \tag{4.14}$$

with W^l as a weight matrix and f as activation function. The activation function has the same properties as for neural networks. This definition can already be used as a propagation rule for GCN, but has two shortcomings. Firstly, the adjacency matrix A does not hold self-loops. Thereby every node considers all features of all neighboring nodes, but not its own to calculate its output. This can be easily fixed by adding the identity matrix $\mathbb{I}$, thereby $A' = A + \mathbb{I}$. Secondly, the propagation is not normalized. To do so the adjacency matrix A can be multiplied to the diagonal node degree matrix D, which normalizes every row in the product to one. With $D^{-1}A$ or $D^{-1/2}AD^{-1/2}$ this task is tackled. In practice the latter method is usually used, as it does not average over neighboring nodes. With these alterations the propagation rule is

$$g(H^l, A) = f(D^{-1/2}A'D^{-1/2}H^lW^l), \tag{4.15}$$

which is introduced in [39] as Graph Convolution. There are different convolution methods on graphs, for this work Graph Convolution (GC) and Edge Convolution (EC) from the dgl library are used. Both implementations are shortly introduced up next. After propagating the input through the graph network again a backpropagation method is used to adjust all trainable parameters. As for neural networks the gradient descent method is widely used. The backpropagation thereby in principle works as for neural networks, displayed in algorithm 1.

Graph Convolution

GC is introduced in [39]. The used implementation of GC only differs in an additional bias b, from the discussed version in equation 4.15. Additionally, here the propagation is displayed in vector representation. The node v_i in layer l is updated with a weighted sum of the set of neighboring nodes $\mathcal{N}(j)$ and a bias b

$$h_{v_i}^{l+1} = f(\sum_{j \in \mathcal{N}(j)} \frac{1}{c_{ij}} h_{v_j}^l \cdot W^l + b^l). \tag{4.16}$$

The contributions of the neighbors are weighted with the weight matrix W^l for layer l and normalized with $c_{ij} = \sqrt{|\mathcal{N}(i)|}\sqrt{|\mathcal{N}(j)|}$. The weights W^l and biases b are the trainable parameters, which are adjusted during training. Additionally, an activation function f can be used. In this implementation no activation function is used by default.

Edge Convolution

The EC method is proposed in [40]. In this publication the method is used dynamically, which means the nodes of the graph are rearranged with regards to the next neighbors in feature space. The feature space is defined as the n-dimensional space of the nodes features. Being dynamic is not essential for the EC method, which can also be used statically.
The update method of the nodes $h^l_{v_i}$ is

$$h^{l+1}_{v_i} = \max_{j \in \mathcal{N}(j)}(Relu(\theta \cdot (h^l_{v_j} - h^l_{v_i}) + \phi \cdot h^l_{v_i}))) \tag{4.17}$$

The nodes are updated every layer l in consideration of all neighboring nodes $h^l_{v_j}$ in $\mathcal{N}_j$. θ and ϕ are the learnable parameters, which are adjusted during training. Neighboring refers here to a connection between two nodes with an edge. Here Rectified Linear Unit (ReLU) is used as activation function. Relu is defined as $Relu(x) = max(0, x)$, projecting all negative values to zero, while the positive values are projected onto itself.
To use this method dynamically the graph is updated after every layer with the *k-nearest neighbor* method. This method finds the k nearest neighbors of each node of the graph in feature space. Points which are close in the feature space have very similar features and are thereby similar. The nodes are connected to these k nearest neighbors over edges. This method rearranges the connections between the existing nodes in an updated version of the graph.
Here only a very brief introduction into working with graph networks is given. More detailed information on graph neural networks can be found in [41].

4.3 Geometric Loss Function

In this work a photon emission distribution is reconstructed. The already introduced loss functions do calculate the difference between label and truth element wise and thereby do not take into account the geometric structure of the problem. One method using this is the *earth mover's distance*. In simple words the earth mover's distance describes how one distribution can be rearranged into the other with a minimal "effort". This is defined by the cost martix.
In this work a loss function is used, that uses the metrics of the distributions by using the Sinkhorn divergences. This method is used as implemented in the python library geomLoss [42]. The implementation and math of the used loss function is discussed here, while the used properties and the adaption to the given setup is showed in chapter 7. The following section is mainly based on [42], as the geomLoss library and thereby the used loss function is based on the given publication.
The Sinkhorn divergences interpolates between Optimal Transport (OT) and Maximum Mean

Discrepancy (MMD). Both methods measure the distance between two probabilities by using the geometry of the underlying space. The methods are shortly introduced by using unit-mass, positive empirical distributions $\alpha, \beta \in \mathcal{M}_1^+(\mathcal{X})$ on the feature space $\mathcal{X}$.
One computationally efficient method to approximate the OT costs is the entropic regularization. The used definition of the OT is given, for $\epsilon > 0$, by

$$\begin{aligned} \mathrm{OT}_\epsilon(\alpha, \beta) &\overset{\text{def}}{=} \min_{\pi_1=\alpha, \pi_2=\beta} \int_{\mathcal{X}^2} \mathrm{C}\, \mathrm{d}\pi + \epsilon \mathrm{KL}(\pi | \alpha \otimes \beta) \\ \text{with } \mathrm{KL}(\pi|\alpha \otimes \beta) &\overset{\text{def}}{=} \int_{\mathcal{X}^2} \log\left(\frac{\mathrm{d}\pi}{\mathrm{d}\alpha \mathrm{d}\beta}\right) \mathrm{d}\pi. \end{aligned} \tag{4.18}$$

$\mathrm{C}(x, y)$ is the symmetric positive cost function. In this implementation the cost function $\mathrm{C}(x, y) = \|x - y\|^2$ on $\mathcal{X} \subset \mathbb{R}^D$ is used. The integral minimized over the couping measures $\pi \in \mathcal{M}_1^+(\mathcal{X}^2)$, with (π_1, π_2) describing the marginals. $\mathrm{KL}(\pi|\alpha \otimes \beta)$ is an entropic barrier added to the linear OT. The factor ϵ weights the entropic barrier. With ϵ set to zero, the entropic barrier is not considered. In this case the quadratic Wasserstein distance is obtained for the given definition of $\mathrm{C}(x, y)$. With the cost function defined as $\mathrm{C}(x, y) = \|x - y\|$, $\mathrm{OT}_0(\alpha, \beta)$ describes the Earth movers distance. An additional entropic barrier was firstly used in [43]. This addition makes it possible to calculate $\mathrm{OT}_\epsilon(\alpha, \beta)$ efficiently for $\epsilon > 0$. Thereby, the possibility is obtained to efficiently compute a geometric loss, that samples between measures.
The MMD method uses a simpler approach by defining a positive definite kernel $k(x, y)$ on the feature space $\mathcal{X}$. The kernel describes how the distance between two points in feature space is calculated. In euclidean space $\mathcal{X} \subset \mathbb{R}^D$ for example the Gaussian kernel could be used $k(x, y) = exp(-\|x - y\|^2/2\sigma^2)$. The kernel loss is defined as the integral of k over $\mathcal{X}$, with $\xi = \alpha - \beta$

$$\mathrm{L}_k(\alpha, \beta) \overset{\text{def}}{=} \frac{1}{2}\|\xi\|_k^2 \overset{\text{def}}{=} \frac{1}{2} \int_{\mathcal{X}^2} k(x, y) \mathrm{d}\xi(x) \mathrm{d}\xi(y). \tag{4.19}$$

In case k is universal, $\| \cdot \|_k$ metricizes the convergence in law.
The Sinkhorn divergence unites the benefits of OT and MMD. To do so the following definition is used

$$S_\epsilon(\alpha, \beta) \overset{\text{def}}{=} \mathrm{OT}_\epsilon(\alpha, \beta) - \frac{1}{2}\mathrm{OT}_\epsilon(\alpha, \alpha) - \frac{1}{2}\mathrm{OT}_\epsilon(\beta, \beta). \tag{4.20}$$

$S_\epsilon(\alpha, \beta)$ interpolates between both methods, converging to OT for $\epsilon \longrightarrow 0$ and converging to MMD for $\epsilon \longrightarrow +\inf$. Mathematically defined as follows

$$S_\epsilon(\alpha, \beta) \longrightarrow \begin{cases} \mathrm{OT}_0(\alpha, \beta) & \epsilon \longrightarrow 0 \\ \frac{1}{2}\|\alpha - \beta\|_{-C}^2 & \epsilon \longrightarrow +\inf. \end{cases} \tag{4.21}$$

With equation 4.20 the sinkhorn divergences between two identical distributions is zero, $S_\epsilon(\alpha, \alpha) = 0$. The reason why equation 4.20 has been introduced is the entropic bias. In simple words, the gradients of $\mathrm{OT}_\epsilon(\alpha, \beta)$ derive a distribution α to a shrunk measure of the target distribution β, if α is a Dirac distribution located at the mean value of β. Equation 4.20 was introduced to fix this problem, by mimicing the structure of a squared kernel norm, as used with MMDs. S_ϵ is a positive definite loss function which is convex and offers metrization of the convergence in law. These properties are displayed in detail and prooven in [42].

Chapter 5

Simulation and Data

The goal of this work is to develop machine learning methods, which can reconstruct a muon path and a corresponding shower in a liquid scintillation detector (LSD). The muon path is supposed to be reconstructed firstly with coordinates. In the second step the architecture is supposed to be adjusted to reconstruct the photon emission distribution caused by a muon and a shower in the detector. Therefore, the voxel representation is supposed to be used. The input for the machine learning methods is the light detected by the PMTs on the walls of a LSD. The needed data is simulated and the simulation and data is described in this chapter.
To train neural networks a lot of data is needed, which is already analyzed and categorized depending on the task the network is learning. In this specific case the real data from the experiment cannot be used. On the one hand the experiment is still under construction and there has no data been produced yet. On the other hand this thesis concentrates on the reconstruction techniques, not on the reconstruction of real data.
For checking whether or not a certain technique is in general suitable for a task, one approach is to start of with smaller but similar tasks and configure the problem in the progress till the whole task is handled. This approach is used in this thesis, which makes it necessary to simulate the data and brings the benefit of being able to configure the complexity of the model with every step. In this work the Monte Carlo Method is used to simulate the data. The Monte Carlo method relies on the rule of large numbers and uses random sampling to obtain numerical results. A special form of the Monte Carlo Method is used, a Toy Monte Carlo Simulation. This method is used to simulate data without the requirement to fully simulate the real process. Only the main features for the particular problem are simulated, while other effects can be neglected. This method is in particular useful for testing applications or methods.

5.1 Simulation

For training a neural network the data needs to include the key features of the events, for example coordinates defining a muon track, which the network should learn to reconstruct. Additionally, the signals which the events caused in the detector, i.e. the photon counts taken

by the PMTs of a LSD. Therefore, it is very important to keep the connection between the signal and the features of every event. This makes it necessary to simulate the whole process, rather than only the outcomes. In this work the goal is to reconstruct muon tracks in a liquid scintillation detector with showers along the track. In these showers cosmogenics can be produced, which mimic the detection signal of reactor neutrinos and are one of the biggest background in experiments like JUNO. To reach this two different reconstruction goals are set. The first one is a reconstruction of coordinates describing a track and will be referred to as *coordinate reconstruction* and the second one is to reconstruct the photon emission distribution of tracks with voxels, referred to as *voxel reconstruction*.
The simulated muon path is straight and crosses the detector from one wall to another. The scintillation light is measured by PMTs on the detectors skin. These give the signal data. The full data set includes the input data, describing the detector response and the data for the two reconstruction goals, describing the path.

5.1.1 Detector

The detector is, in contrast to the JUNO detector, cubic, not spherical. While the physical processes are all the same regardless of the geometrical shape, a cubic detector serves all the features needed for a test of the reconstruction technique. The simulated detector has an edge length of 4 m and will be considered as the interaction volume, where the events happen and produce signals. The skin of the detector is covered with simulated PMTs, which cover 100 % of the detector's skin. On each side of the cube 10×10 PMTs are simulated, this leads to a total number of 600 PMTs. Every PMT is counting the photons per time, generating a histogram for every event. The PMTs are simulated with a time resolution of 1 ns, which gives the width of the bins in the histograms and covers with the time resolution of the PMTs used in JUNO. For every event 50 ns are taken, starting with the muon entering the detector. This time frame would allow a particle to travel ≈ 15 m, under the assumption it travels with the speed of light. The signal data for one event has the shape 600×50.

5.1.2 Path

The path always starts on the top of the detector at $z = 2$, with the origin of coordinates in the center of the detector. The x- and y-coordinates of the start point are randomly distributed in range of $[-1.5, 1.5]$. This limits the minimal path length is limited to 0.5 m. The direction of the path is normally distributed with the polar coordinates $\phi \in [0, 2\pi]$ and $\theta \in [\pi/2, \pi]$. The path ends when it reaches the next detector wall. At a random position along the path a peak is simulated. At the peak 5000 photons are emitted. Start, peak and end position are the points, which are used to describe the path for *coordinate reconstruction* label. Figure 5.1 displays the distribution of start, peak and end positions in the detector. The right plot shows the start positions of the paths, which are always located on the top skin with the given constraints. Peak and end positions are displayed in the center and on the right. While the peaks are distributed

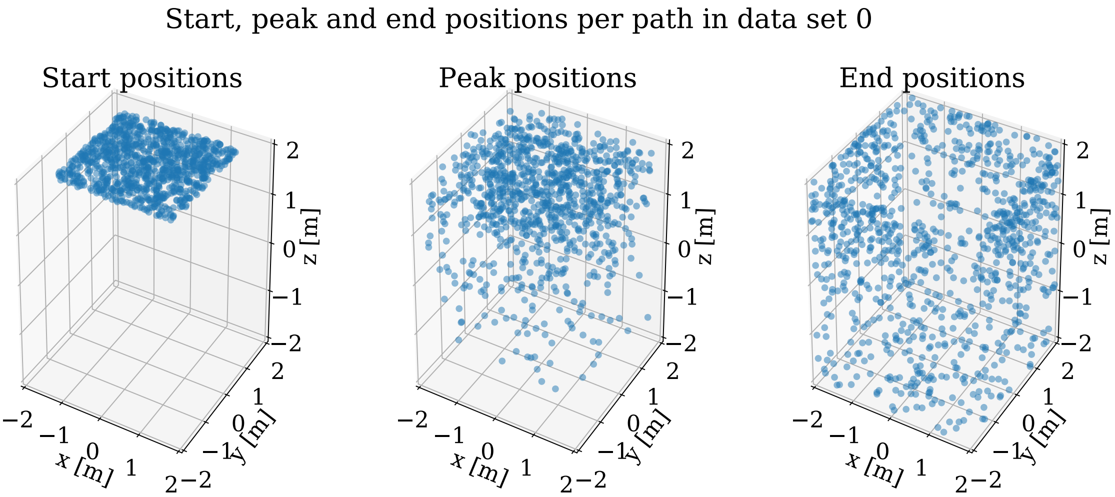


Figure 5.1: Start, peak and end positions in the detector (left, center, right) per path in data set 1.

in the detector volume, the end positions are located on the detectors skin, but not on the top skin.

5.1.3 Photons

Along the path photons are emitted. All these photons are emitted with a random direction, with the spherical coordinates $\phi \in [0, 2\pi]$ and $\theta \in [\pi/2, \pi]$. The photons travel through the detector with the speed of light, which is approximated with $3 \cdot 10^8$ m/s, till they reach a detector wall. On the way the photons can be absorbed or scattered with a mean free path of $d_{absorption} = 80$ m for absorption and $d_{scattering} = 25$ m for scattering. These values orient on the scintillation liquid used at JUNO. Due to the size of the detector both processes do not occur often. The scintillation decay distribution and the time resolution of the PMTs are considered with

$$f(s,t) = \frac{s}{2t_c} \cdot exp\left(\frac{1}{t_c}\left(t + \frac{t_{pmtr}^2}{2t_c}\right)\right) \cdot \left(1 + erf\left(-t - \frac{t_{pmtr}^2}{t_c}\right)\frac{1}{\sqrt{2}t_{pmtr}}\right) \tag{5.1}$$

from [44]. With t as the time, t_c the mean decay time of the scintillation material considered to be 0.2 ns, s the signal strength and t_{pmtr} the PMT resolution of 1 ns .

Along the path every 0.001 m 10 photons are emitted, giving a pseudo-continuous photon emission. On a random point along the path a peak is simulated where 5000 photons are emitted. All photons are simulated till they reach a detector wall. There they are mapped on the PMTs forming the PMT signals. The detector volume is split into 20x20x20 voxels leading to a total number of 8000 voxels with an edge length of 0.2 m. The emission points of all photons are mapped onto the voxel structure, to obtain the *voxel reconstruction* label. In figure 5.2 a simulated path is displayed, with the emitted photons mapped onto the detector walls in green and the path in red. Additionally, the PMT histogram of one PMT, which is sketched into the

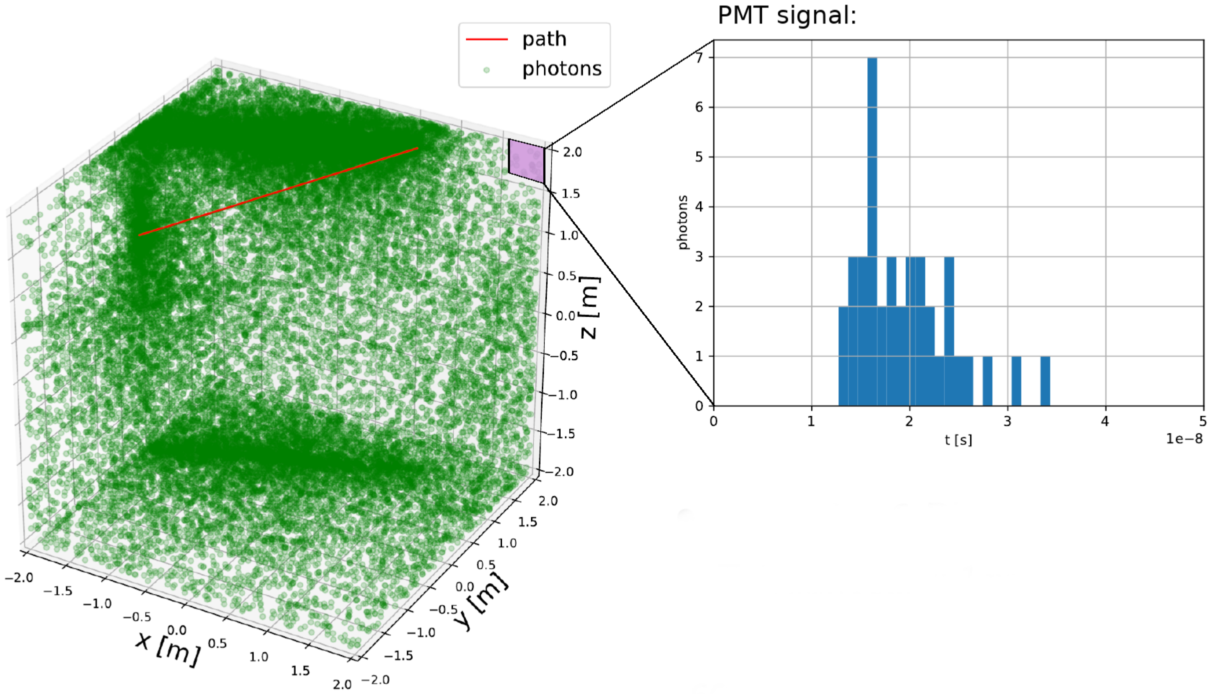


Figure 5.2: Simulated event in the detector, in green the photons are marked on the detector walls, the path is marked in red. In purple one PMT is sketched in the detector, with the related histogram. The histograms of all 600 PMTs form the input data used for the reconstruction.

detector, is displayed. In this simulation several processes are not considered:

- Cerenkov light,
- different path types like curved, stopped or with several peaks,
- reflection,
- attenuation, in this work only absorption is considered without reemission.

These processes do not have a big impact for analyzing the reconstruction qualities. They could be added in a later iteration of this work.

5.2 Generated Data

For training four data sets have been simulated, with 1000 events each. These are named from 1 to 4. Set 1 and 2 are mainly used, 3 and 4 are back-up sets, for cases more data is needed. Set 1 is used to train the networks and set 2 to validate the training. In figure 5.3 and 5.4 data sets 1 to 4 are plotted in regards of the photon number, path length and amount of filled voxels per event. In figure 5.3 the distribution of filled voxels in the sets 1 to 4 is displayed. The total number of voxels in the detector is set to 8000, with a fraction of 3 to 50 being filled. The filled voxels define the simulated muon path and are defined as voxels in which photons where emitted. The number of photons per event in data sets 1 to 4 sums up to about 10000 to 60000 photons, all data sets follow the same distribution in figure 5.4b. The path length and the amount of

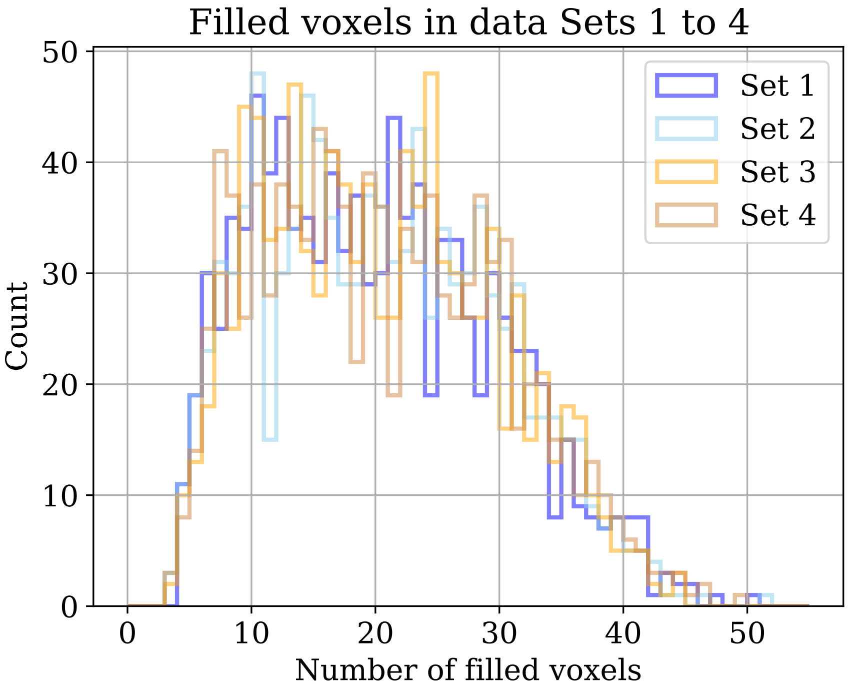
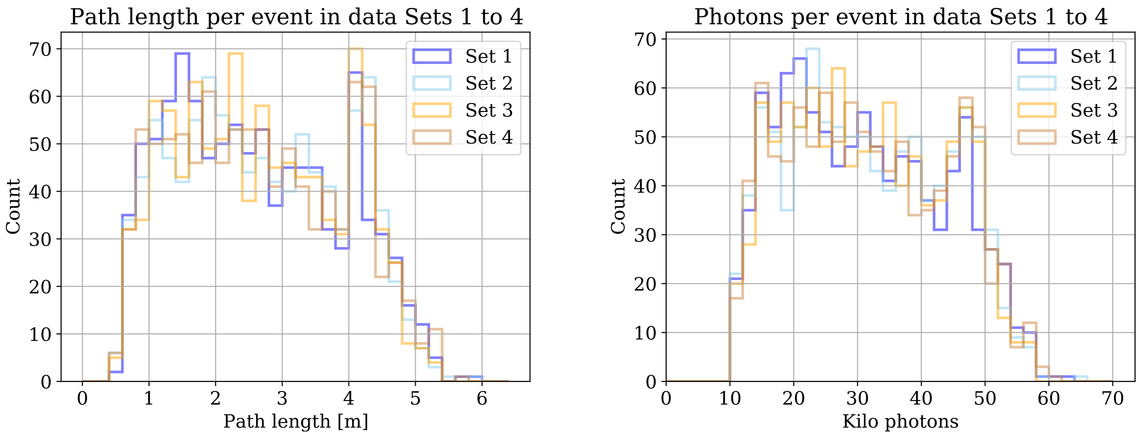


Figure 5.3: Simulated data sets 0 to 4 in comparison, with regards on filled voxels per event. The amount of filled voxels ranges between 3 and a maximum of 50.



(a) path length per event

(b) number of kilo photons per event

Figure 5.4: Simulated data sets 0 to 4 in comparison with regards on the path length per event 5.4a and the number of photons per event 5.4b. As the amount of photons is directly proportional to the path length, both plots follow the same distribution.

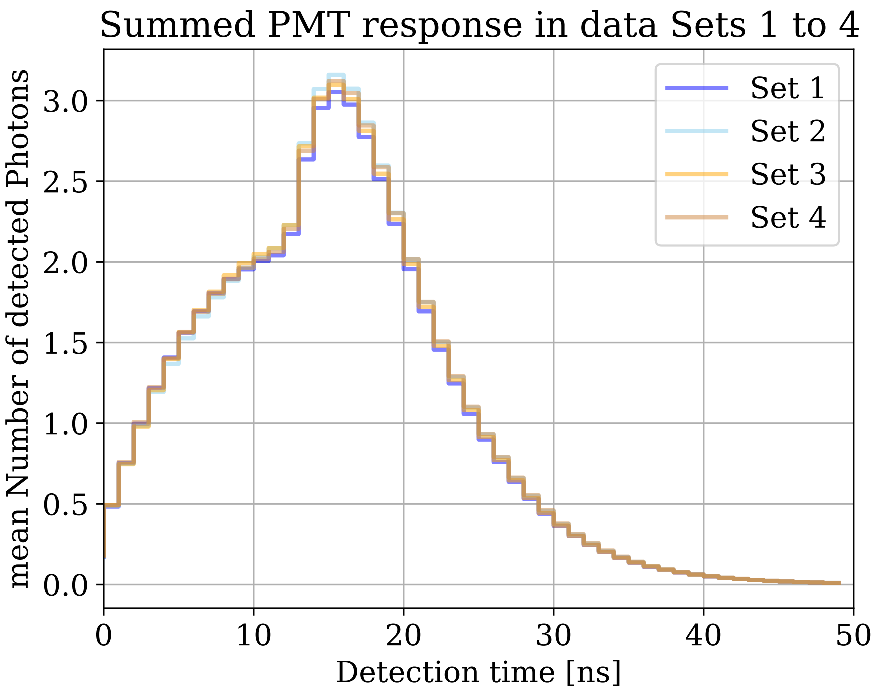


Figure 5.5: Mean PMT response for the simulated data sets 1 to 4 in comparison. All four data sets follow the same distribution, with a peak at roughly 16 ns.

filled voxels per event follow each the same distribution, as the amount of emitted photons is directly proportional to the path length. Per millimeter 10 photons are emitted, with an additional peak of 5000 photons at a random position along the track. The path lengths range between around 0.5 and 5.7 m, the distribution is displayed in figure 5.4a.

In figure 5.5 the mean PMT response is displayed. All four data sets follow the same distribution, with a peak at approximately 16 ns. After the peak the distribution converges to 0, this indicates that the measuring time of 50 ns is a valid choice.

5.3 Classic Data Analysis

The data sets are analyzed classically. A simple analysis method is used to create a minimal reconstruction goal. Both reconstruction goals are classically reconstructed.

5.3.1 Simple Coordinate Reconstruction

To find the coordinates of start, peak and end of the path the input data and the coordinates of the PMTs are used. The start point can be found easily by taking the first hit in the PMTs. This only works for this simulation, as it starts with the muon entering the detector.

The end of the path is not necessarily the last hit in the detector. As the path ends by intersecting with one of the detector walls, the PMT which holds the end point is searched. The end of the path is indicated by an increase of photons in this PMT with a sudden decline when the

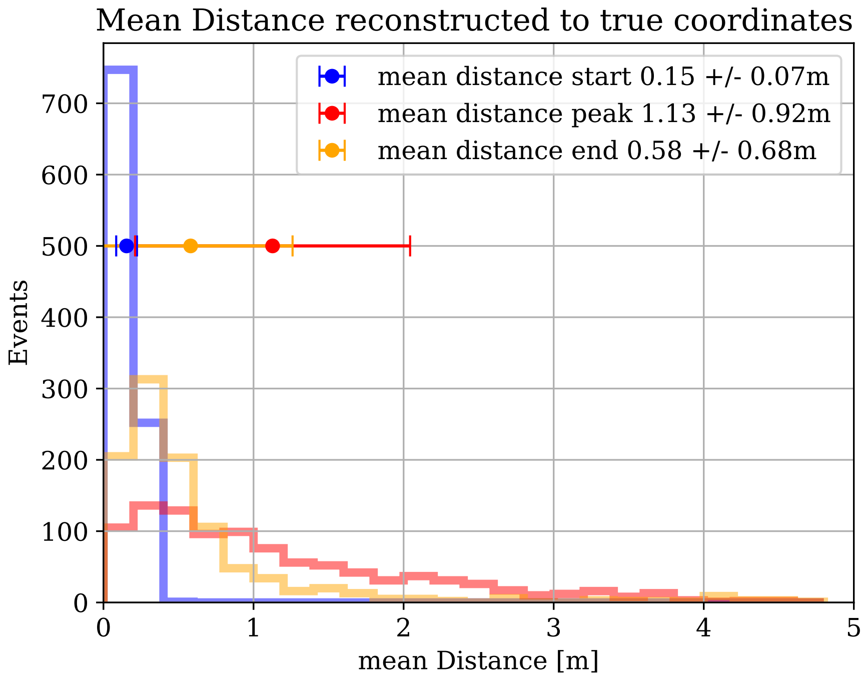


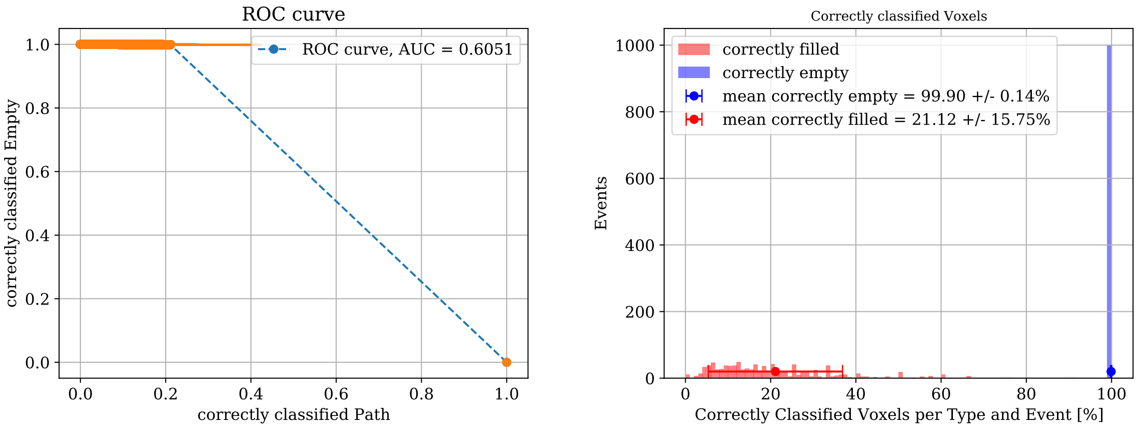
Figure 5.6: Distances of the reconstructed points start, peak and end to the true coordinate in set 1. The mean distances are (0.15 ± 0.07) m for the start, (0.58 ± 0.68) m for the end and for the peak (1.13 ± 0.92) m points.

muon leaves the detector. Therefore, the end point in this reconstruction is defined as the PMT coordinate of the PMT holding the steepest decline in the photon distribution between two neighboring bins. Lastly, the peak point needs to be reconstructed. As the path is simulated to be straight, with the peak along the path, the reconstructed coordinates need to follow the same characteristics. Therefore, the peak coordinate for this simple reconstruction is defined to be on the point on the path between start and end point, closest to the highest peak detected in the PMTs.

The reconstruction quality of these coordinates can be described by the mean distance between true and reconstructed points. The distribution of these mean distances of the characteristic points in data set 1 is displayed in figure 5.6. The start point is reconstructed with a mean distance of (0.15 ± 0.07) m. This distance is very good as the dimensions of the PMTs is 0.4x0.4 m. As the start coordinate is defined by the coordinates of the PMTs, defining their position by the center of the PMTs, a better reconstruction will probably not be archived with other methods. This method works here especially well, as the simulation starts with the muon entering the detector with no other sources of light. For real or more realistic data this approach would not work, as the time measurement would not necessarily start with the muon entering the detector.

The end points are reconstructed with a distance of (0.58 ± 0.68) m. Even though a simple method was used, in over 50 % of the analyzed events the distance is ≤ 0.4 m.

The peak points are reconstructed with a distance of (1.13 ± 0.92) m. The used reconstruction



(a) ROC curve

(b) Fractions correctly classified voxels

Figure 5.7: Plot 5.7a displays the ROC curve of the classic reconstruction of the photon distribution with voxels. In figure 5.9 the fraction of correctly filled and correctly predicted emplty voxels.

method uses the measurements from the skin of the detector to find a point inside its volume, by using the already reconstructed points and the knowledge of the simulation properties. The disadvantage of this method is that errors in the end and start coordinates add up in the reconstructed peak coordinate. This makes the peak coordinate the worst reconstructed coordinate in this analysis.

5.3.2 Simple Voxel Reconstruction

To reconstruct the photon distribution with voxels, the characteristic points from the coordinate reconstruction are used. All voxels along the line between start and end coordinate are filled with a fraction of the photon sum. The voxel containing the peak coordinate is filled with 5000 extra photons, which are subtracted from the total photon sum.

To evaluate this simple method the fraction of correctly classified as empty and full voxels, the mean number of photons in full and empty voxels and the mean distance between true and reconstructed peak and the area under the ROC curve (AUC) are used as characteristics. The AUC and the fractions of voxels correctly classified, display the separation strength of the reconstruction. The distance between the true and reconstructed peak and the mean photon sum in filled and empty voxels show, how the photons are distributed in the reconstruction.

In figure 5.7 the ROC curve and the fractions of correctly classified voxel types are displayed. The ROC curve has an AUC of 0.605, with only entries on the upper left corner. This behavior is as expected. As the predictions with this simple classic method do not vary much. Most of the voxels are here reconstructed to be zero, the next bigger values expected with this reconstruction method are already relatively big fractions of the photon sum in the reconstructed path voxels. Therefore, no cut can archive a fraction of correctly classified path voxels higher

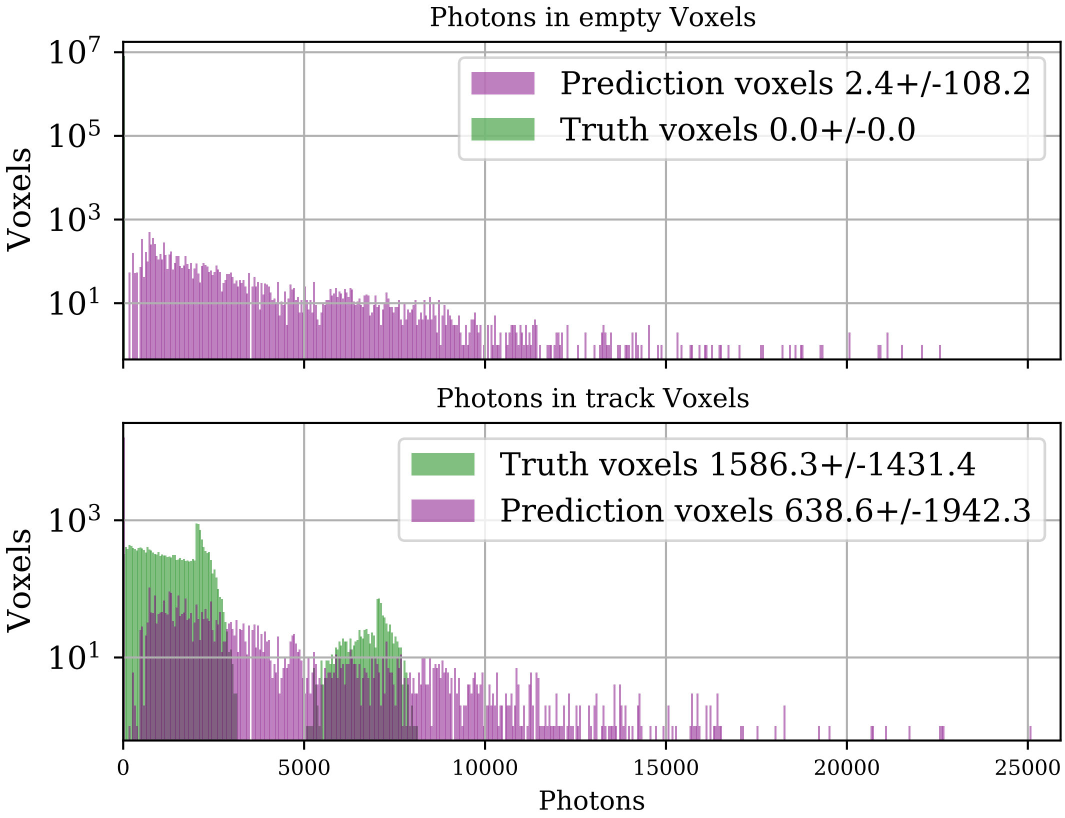


Figure 5.8: Photons in empty and filled voxels are displayed in the upper and the lower plot. In both plots the green distributions show the photons in the filled and empty voxels as simulated. The purple distributions show how many photons have been reconstructed in these voxels with this reconstruction method. Both plots share the same x axis, the ticks are labeled in the lower plot.

than 0.25 but smaller than 1.

For figure 5.9 the cut to distinguish between empty and full voxels is set to 1 photon, with less than that defined as empty. The fraction of correctly classified empty voxels is very high with (99.9 ± 0.13) %, still only (21.12 ± 15.75) % of the filled voxels are reconstructed correctly. Even though the empty voxels are mostly correctly classified, the separation between the two classes of voxels, empty and filled, is not very successful with this method.

In figure 5.8 the amount of reconstructed photons per voxel class are compared with the simulated photon distribution. The mean amount of photons in empty voxels is (2.4 ± 108.2) photons. In the filled voxels a mean photon sum of (638.6 ± 1942.3) photons is reconstructed, which is only a 40 % of the true mean photon sum for filled voxels, which is (1586.3 ± 1431.4) photons. Even tough the reconstructed mean photon sum of the filled voxels is too small, the highest entries of the distribution are much too high. The reconstructed distribution has its maximal values at around 25000 photons in one voxel, for both voxel classes. In the simulation the maximal photon sum for filled voxels is only around 8000 photons, while the simulated photon sum in empty voxels is 0. The mean distance between true and reconstructed peak is displayed in

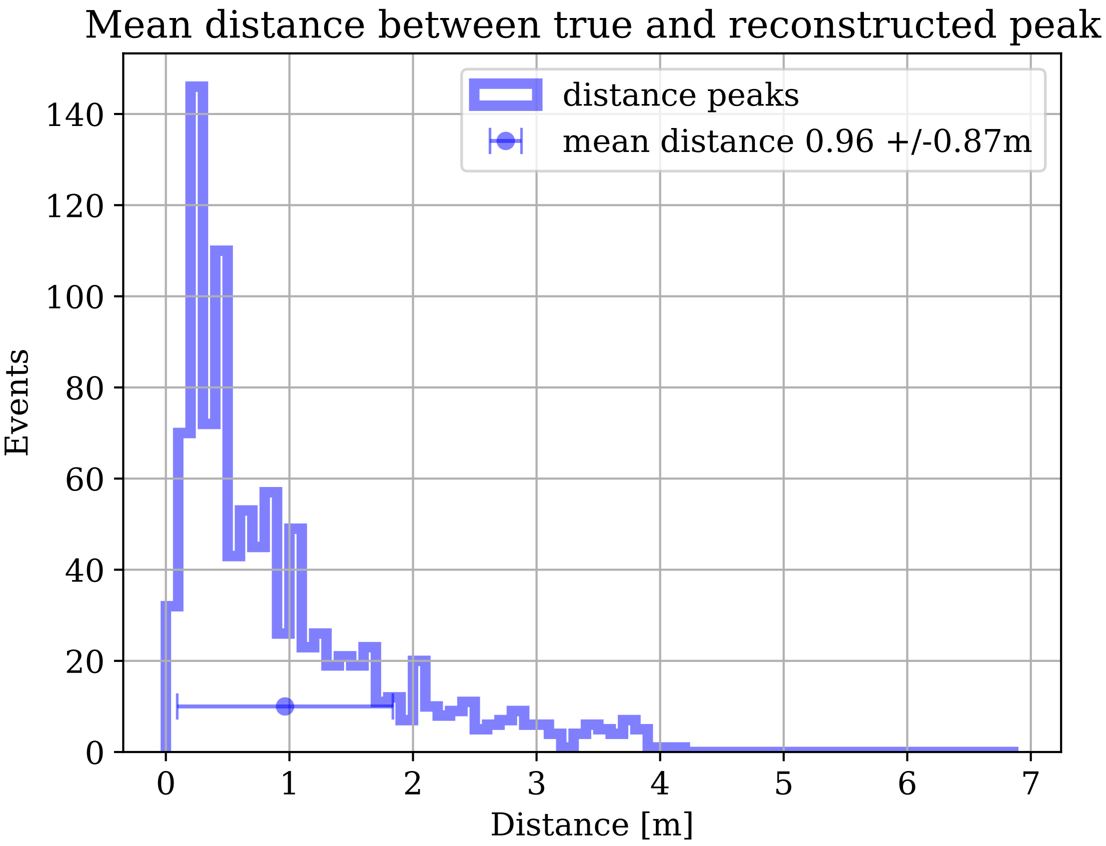


Figure 5.9: The mean distances between true and reconstructed peak position.

figure 5.9, with a mean value of (0.96 ± 0.87) m. As the peak position is used from the coordinate reconstruction a similar resolution is expected. Still the voxel reconstruction archives an around 0.17 m smaller distance. This is due to the segmentation of the detector volume in voxels, the peak position is now defined as the position of the voxel with the highest photon sum. Due to the reconstruction method the empty voxels are mostly correctly classified. Additionally, the knowledge of the path characteristics, allows this reconstruction method to reconstruct a path with the same characteristics. Still the reconstruction is not very good, as only around 20 % of the filled voxels are correctly identified. As the used method uses the results from the coordinate reconstruction, the precision is in a similar range.

The simple coordinate and the simple voxel reconstruction are used to set a minimal goal for the reconstruction method which is developed in this work. With more effort and more elaborate techniques also classic methods can reach much higher precision in path reconstruction tasks. In this work machine learning methods using the graph architecture are used. The simple reconstruction methods show that the looked for information can be found in the input data. Additionally, from the reconstruction it becomes visible how simple the data can be reconstructed. For example, the start position can be reconstructed easily by taking the coordinates of the PMT with the first hit. A neural network based reconstruction method is expected to find this connection and thereby predict the start points with a similar precision. The other points and especially the photon emission distribution are expected to be reconstructed with far more precision.

Chapter 6

Coordinate Reconstruction

In this work two different reconstruction methods are implemented. Firstly, the *coordinate reconstruction* and secondly the *voxel reconstruction*. For the coordinate reconstruction, the muon track is characterized by characteristic points along the track. The voxel reconstruction is covered in the next chapter.

6.1 Goal

The goal of the coordinate reconstruction is to reconstruct the coordinates of start, peak and end points of the muon tracks. As input the simulated data is used, which is described in detail in chapter 5. The data has the shape 600×50 representing the binned photon distribution measured by 600 simulated PMTs. Each PMT takes 50 bins with a width of 1 ns. The reconstruction is compared to the label, which was determined in the simulation, consisting of the coordinates of start, peak and end point of the muon path inside the simulated detector. Along each muon track a shower is simulated by a discrete release of photons. This is a peak in the photon emission distribution, therefore the peaks represent showers along the track.

The coordinate reconstruction is used as a simpler method to describe the path, compared to the photon emission distribution with voxels. Using a simpler reconstruction method in the beginning brings the benefit that reconstruction methods can be tested on whether or not they are in general suitable for the given task. Methods which perform well here can later on be used in more complex reconstruction approaches, as showed in chapter 7. Additionally, reconstructing a muon track and a corresponding shower with the help of coordinates could already be helpful in some experiments. For example in the JUNO experiment, a big background source is correlated with showers along muon tracks. The proposed veto discriminates a cylindrical volume around the muon track with a radius of $r = 3$ m. It has an efficiency of 83 % on the reactor neutrino detection process. The reconstructed shower position in the detector would give the opportunity to veto only parts of the detector, decreasing the dead time and increasing the efficiency on the IBD. More about JUNO and its backgrounds can be found in chapter 3.3.

6.2 Implementation

All used code for this thesis is written in Python3. For coding the network architectures "PyTorch" [45] is used as backend. It offers tensor computation and the autograd system. This is very helpful as it handles the backpropagation for neural networks. Additionally, many concepts used with neural networks are included in this library, such as pooling methods, activation and typical loss functions. To implement the graph neural network structures the Deep Graph Library (DGL) [46] is used. Here DGL operates with PyTorch as backend, which ensures simple transitions between neural network and graph network components. Both PyTorch and DGL offer the opportunity to compute on GPUs as well as CPUs. In this thesis all code runs on CPUs.

The implementation of the developed architectures provided on Github.[1] To train and validate the network the data sets 1 and 2 are used. These have each 1000 events and are discussed in detail in chapter 5.

6.2.1 Basic Idea

The goal of this reconstruction is to use the PMT measurements from the detectors skin as input to obtain coordinates for characteristic points along the muon track inside the detector. The input data has the shape 600×50 and has an underlying geometrical structur, which is not directly visible in the data. To use the geometrical structure of the data a GNN or GCN could be used. Graph Neural Networks have a great reputation in nowadays research. This method delivered great results for Jet tagging with the ParticleNet [47].

The data structure provided by the simulation suits nicely for the graph structure, which uses N dimensional point clouds as data. This structure is already provided by the detector setup, which is simulated to obtain data. One PMT can be considered as one point with N dimensions, these correspond to the bin entries in the PMT data. PMT features could be used as additional dimensions. Additionally, all PMTs form a geometrical structure representing the detector volume. The spatial relationship between the PMTs can also be implemented in the graph structure. The here presented approach uses the PMTs as nodes. The edges connect the PMT nodes. The idea is to connect spatially neighboring PMT nodes to provide the detector setup to the network.

This method has the benefit, that the data taken by a LSD can be directly used as input for the network. Additionally, the geometrical structure of the detector is implemented implicitly, which means this structure does not have to be learned by the network.

---

[1] `https://github.com/rwrth/Coordinate_Reconstruction`

6.2.2 Optimization

To optimize the network form the basic idea to the final architecture, parameters are systematically changed and resulting architectures trained. To optimize the network's hyperparameters the prediction has to be evaluated. To compare different predictions characteristic features are defined, which describe the prediction quality. For this optimization these features are the following:

- mean distance between reconstructed and label points for start, peak and end of the track,
- percentage of reconstructed end points on any detector skin,
- percentage of reconstructed start points on the top detectors skin at $z = 2$,
- percentage of reconstructed peak points on the line between reconstructed start and end point.

The mean distance between a reconstructed and the true point already tells how well the point has been reconstructed. Additionally, characteristics of the points are used to further describe the quality of the muon path reconstruction. The simulated path always starts on the top detector skin at $z = 2$ and ends on the next detector wall the muon passes. On a random point along the straight path a shower is simulated by a discrete photon emission. To check whether these characteristics are reconstructed by the network the percentages of events fulfilling these characteristics are monitored.

To optimize the the hyperparameters of the networks architecture, the hyperparameter of the architecture are varied and the resulting predictions compared in regards of the optimization features. The goal is to minimize the mean distances between reconstructed and true points and to maximize the percentage of reconstructions which fulfill the defined features.

For this optimization the following hyperparameters have been optimized:

- loss function,
- GCN method,
- pooling method,
- activation function,
- number of layers for graph convolution,
- number of layers for the fully connected network,
- number of features in graphs,
- layer width of the fully connected network,

- dropout,
- learning rate.

6.3 Optimized Architecture

The final optimized architecture is schematically displayed in figure 6.1. It consists of a GCN part and a fully connected part. The GCN uses three layers of edge convolution, which is described in detail in chapter 4. In these layers the features are stepwise increased from the initial 50 to 600. The used graph has 600 nodes, which resemble the PMTs in the detector. These are connected over edges. The method *8 nearest neighbors* builds the graph connecting a node to its eight nearest neighbors on the provided feature space. For this method the spatial coordinates of the PMTs are used. Thereby the graph structure resembles the geometrical setup of the detector.

After the *Graph Net* part of the network, the data needs to leave the graph structure. Therefore, a pooling method is used, here it is *Node Max Pooling*. The pooling method needs to be invariant under translations and is normally taken over the features of a graph. As in this setup the nodes correspond to the PMTs, the pooling method is taken over all nodes. In the optimization this method showed good results using the maximum function. Therefore, the used max pooling method takes the maximal value of every node, which is a total of 600, corresponding to the 600 PMTs.

After pooling the data is processed by the fully connected network. This part has five layers, decreasing the amount of features from 600 to 200 in the first layer and further to 9 in the last one. These nine values correspond to the (x,y,z) coordinates of the three reconstructed points. In this architecture *tanh* is used as an activation function for all layers beside the last one. In the last layer *clamp(-2,2)* is used, which sets the allowed maximum and minimum to ± 2, which are the dimensions of the detector. Additionally, a dropout is used, once in the first and once in the second part. Both times the dropout is 0.2. With this method 20 % of the features are randomly set to zero. This method increases the regularization of the training and thereby avoids overfitting.

6.3.1 Loss Function

The optimized loss function uses the distances between the reconstructed and the label points. The main goal of the reconstruction is to minimize these distances. Additionally, the collinearity of the three points is added to the loss function. To simplify the equation of the loss function the prediction and label are used in coordinate representation. The reconstruction p consists of the three points start, peak and end, labeled with s, p and e, $p = [\overrightarrow{p_s}, \overrightarrow{p_p}, \overrightarrow{p_e}]$. t represents the truth and is used with the same representation $t = [\overrightarrow{t_s}, \overrightarrow{t_p}, \overrightarrow{t_e}]$. The loss function uses the sum of

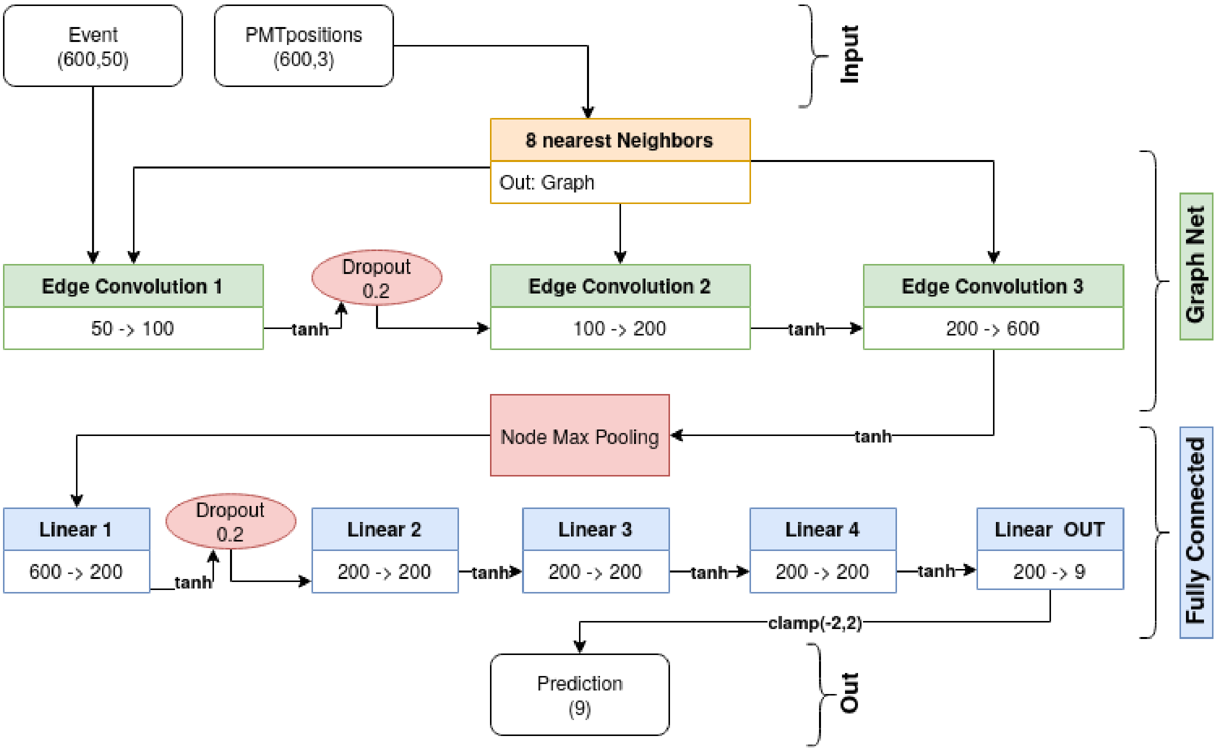


Figure 6.1: Optimized network architecture to reconstruct coordinates. The network consists of a GCN part in green and a fully connected part in blue. The numbers in the boxes below the title of the layers, display how the amount of features changes.

the distances between label and reconstructed points, divided by a factor σ and squared.

$$L(p,t) = \left[\frac{|(\vec{p_s} - \vec{t_s})| + |(\vec{p_p} - \vec{t_p})| + |(\vec{p_e} - \vec{t_e})|}{\sigma}\right]^2 + |1 - \text{coll}|^2. \tag{6.1}$$

The collinearity coll is defined as

$$\text{coll} = \frac{(\vec{p_p} - \vec{p_s}) \cdot (\vec{p_e} - \vec{p_p})}{|(\vec{p_p} - \vec{p_s})| \cdot |(\vec{p_e} - \vec{p_p})|}. \tag{6.2}$$

It is supposed to adjust the three reconstructed points to be on one straight line. The angle α between two vectors $\vec{a}$ and $\vec{b}$ is defined as $\cos(\alpha) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}|\cdot|\vec{b}|}$. For the case all three points are on one line or the two vectors $\vec{a}$ and $\vec{b}$ are collinear, $\cos(\alpha) = \text{coll} = 1$ and the last summand in the loss function vanishes.
In the optimization σ was optimized to 0.1.

6.4 Results

To obtain the results, the optimized architecture is trained and evaluated. The results are described with the same characteristics used for the optimization. For the results the validation data set has been used, as it is unknown to the network.

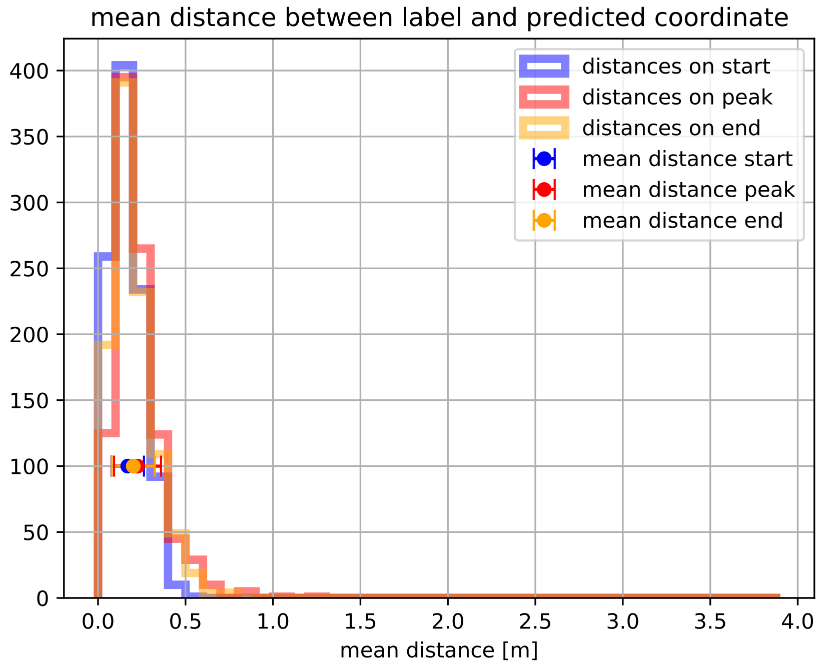


Figure 6.2: Histogram of distances between label and reconstruction for start, peak and end points. The mean distance for the start points is $d_S = (0.16 \pm 0.20)\,\mathrm{m}$, for the peak $d_P = (0.22 \pm 0.14)\,\mathrm{m}$ and for the end points $d_E = (0.21 \pm 0.11)\,\mathrm{m}$.

The most relevant criteria are the mean distances between label and reconstructed points. In figure 6.2 these distances are plotted in a histogram. The mean distance for the start coordinates is $d_S = (0.16 \pm 0.20)\,\mathrm{m}$, for the peak $d_P = (0.22 \pm 0.14)\,\mathrm{m}$ and for the end points $d_E = (0.21 \pm 0.11)\,\mathrm{m}$. Over 85 % of the reconstructed peak points in the evaluated data set 2 are reconstructed with a distance $\leq 0.4\,\mathrm{m}$.

The percentages of fulfilled characteristics are analyzed as well. 99.8 % of the predicted start coordinates are on the top detector skin, at $z = 2$. 94.5 % of the reconstructed end points are on any detector skin. And in 16.7 % of the observed events all three points are reconstructed to be on one line.

The distances between reconstructed and label points is around 0.2 m for all three points. The used PMTs have the dimensions $0.4 \times 0.4\,\mathrm{m}^2$. As the emitted photons are detected by the PMTs, the hit position of a photon on a PMT can only be determined up to an uncertainty of $\sqrt{2(d/2)^2}$, with d as the edge length of the PMT.

The start point is reconstructed with a mean distance of $d_S = (0.16 \pm 0.20)\,\mathrm{m}$ and is thereby the best reconstructed point in this reconstruction. In comparison to the simple classic method with a mean distance for the start point of $(0.15 \pm 0.07)\,\mathrm{m}$, this reconstruction performs in a similar range for the mean value, but the distribution is less spread for the classic method. The start point is the simplest to reconstruct from the given data, as the first hit is always in the PMT covering the start point.

The end and the peak points are reconstructed with this method more precisely, compared to the simple method. The peak points are reconstructed with a mean distance of $d_P = (0.22 \pm 0.14)\,\mathrm{m}$. Over 85 % of the peak points in data set 2 are reconstructed with a distance $\leq 0.4\,\mathrm{m}$. For these events the volume, which holds the peak, can be narrowed to $0.27\,\mathrm{m}^3$. Thereby 85 % of the peak points can be narrowed to less than 0.4 % of the $64\,\mathrm{m}^3$ detector volume.

A reconstruction method to find showers along muon tracks in LSD seems to be promising. Such an application could be used to find muon induced showers in a LSD to discriminate cosmogenics. With a lot of further research and improvement an identification method finding muon induced showers, could be developed from this work.

Chapter 7

Voxel Reconstruction

7.1 Goal

In this chapter the photon emission distribution is reconstructed from the input. The simulated data described in chapter 5 is used. The input is the PMT response from the 600 simulated PMTs with each 50 bins and a bin width of 1 ns. The photon emission distribution is represented by voxels. The detector volume is segmented in 20 × 20 × 20 voxels with the dimensions 0.2 × 0.2 × 0.2 m. For each voxel the sum of photons emitted in the voxels volume is supposed to be reconstructed. Besides from reconstructing characteristic points along the muon track, as showed in chapter 6, the here presented method gives detailed information on the photon emission distribution in the detector. For each voxel the number of photons inside its volume are supposed to be reconstructed. This leads to two types of voxels, filled and empty ones. The filled voxel define the track of the muon and the voxel with the highest amount of photons is the one holding the shower, which is simulated by a peak in the photon emission.
The voxels in truth data can be part of the path, referred to as *path* voxel, or not, referred to as *background*. Additionally, voxels of the reconstruction can be reconstructed as filled with photons or as empty. The reconstruction is supposed to produce semi-positive numbers for every voxel, which represents the amount oh photons emitted in the voxel. A threshold needs to be set to define the two classes filled and empty in the reconstruction. A natural choice would be a threshold at one photon, which assigns voxels with a reconstruction of at least one photon to the class filled. Filled voxels in the reconstruction can either be correctly identified *path* voxels or falsely classified *background* voxels. The empty voxels in the reconstruction can be correctly or falsely identified, respectively. In this chapter the two voxel classes in the label and the four voxel types in the reconstruction are discussed by using the introduced nomenclature.

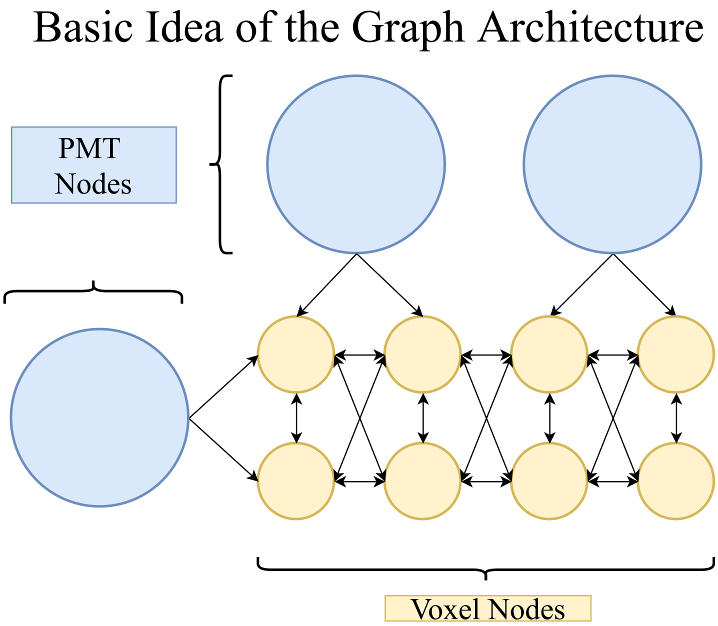


Figure 7.1: Basic idea of the architecture, showing a part of the graph. The voxel nodes are displayed in yellow and the PMT nodes in blue. The edges are visualized with arrows. The PMT voxels use the bin entries of the PMT histograms as features.

7.2 Implementation

For the implementation of this network the same libraries as in chapter 6 are used. "PyTorch" [45] is used as backend and DGL [46], the geometric loss library (geomLoss) [42] is used to determine geometric loss terms. The implementation of the developed and optimized architecture provided on Github[1].

7.2.1 Basic Idea

The used graph setup for the coordinate reconstruction already performed well by predicting three characteristic points along the track with a mean distance of $\approx 0.2\,\mathrm{m}$ to the label points. A similar architecture is used for the voxel reconstruction. In the coordinate reconstruction the graph consists of 600 nodes. These nodes represent the 600 PMTs. As input features the 50 entries from the histograms taken by the PMTs are used. These nodes are connected over edges to their eight spatially closest neighbors. In the coordinate reconstruction the data leaves the graph representation and is further processed in a fully connected network to create the output consisting of three points.

The geometrical structure of the detector should also be used in the voxel reconstruction. The basic idea is that the photons, detected in the PMTs, propagate backwards through the given structure to their origin. This progress is supposed to invert the physical progress of photon emission and propagation, which takes place in the detector when a muon passes through. To implement this idea a graph is used which is supposed to represent the structure of the

---

[1] `https://github.com/rwrth/Voxel_Reconstruction`

detector. It is made out of 8600 nodes representing the 600 PMTs and the 8000 voxels. The PMT and voxel nodes are connected over edges to their nearest spatial neighbors. The input data is used as features of the PMT nodes. It is supposed to be propagated over connected nodes and edges in the detector volume, which is represented by the voxel nodes. The output needs to have 8000 entries representing the photons emitted in the $20 \times 20 \times 20$ voxels.
In figure 7.1 a part of this graph is schematically displayed. Three PMT and eight voxel nodes are displayed in blue and yellow. The PMT nodes use the histograms taken by the PMTs as features. As displayed in the figure the PMTs are connected over edges to the spatially closest voxels. Neighboring voxels are connected over edges with each other. The updated node features are determined by using the features of all over edges connected nodes. Therefore, the features can only propagate to their direct neighbors in one layer. This means for propagating data through ten voxels, ten layers are needed to reach the core of the detector. The optimized graph is discussed in detail in section 7.3.1.

7.2.2 Optimization

To develop a well performing network from scratch a start architecture is optimized with regards to the predictions. During the optimization the hyperparameters of the architecture are varied systematically. The resulting architectures are trained and evaluated on the same data set. The reconstructions from different approaches are compared with regards to characteristic features, which describe the quality of the predictions. The here used features are on the one hand describing the classification performance of the network and on the other the regression quality. The classification performance describes how well two or more classes can be distinguished in the reconstruction. The two classes in this reconstruction are *path* and *background* voxels. The *path* voxels are filled with photons in the label and represent the muon track, while the *background* voxels are empty. In the reconstruction the *path* voxels are supposed to be filled and the *background* voxels are supposed to be reconstructed as empty. But this is not necessarily the case. In the reconstruction every voxel is represented by a float ≥ 0. A natural choice to distinguish between *path* and *background* voxels would be a threshold at one photon. Voxels predicted with less than one photon would be empty, and voxels filled with at least one photon filled. But this choice is not always the best, for example when all voxels are systematically filled with too many or too little photons. Therefore, a cut needs to be set, which distinguished between the two classes in the reconstruction.
A cut independent method to describe the classification quality is the area under the curve (AUC) using the ROC curve. This is explained in detail in chapter 4. AUC is the first characteristic feature. The second one is the percentage of voxels correctly classified as empty with the threshold defined to correctly classify all *path* voxels as filled in the used data set. As the number of *path* voxel is only $\leq 1\,\%$ of the total voxel sum, they are crucial to identify, because they describe the searched muon path. This feature displays the classification quality with

regards on the *path* voxels. Additionally, it reveals whether all *path* voxels can be correctly classified with a suitable cut and how well the *background* voxels are classified in this case. The feature is called perfect cut (pc). In a perfect reconstruction this value would reach up to 100 %. This feature is very strict, as it becomes zero, if one *path* voxel is reconstructed distinguishable from background. Still it is very helpful as it identifies setups in which all *path* voxel can be identified as such and additionally tells how well the *background* voxels are reconstructed.

The photon sums in the voxels need to be compared as well, to observe the photon distribution quality. For optimization the mean amount of photons in *path* and *background* voxels are monitored. Lastly, the peak reconstruction qualities are monitored. The peak along the muon track represents a shower. The peak reconstruction quality is measured by the mean distance between the label peak and the reconstructed peak. In the reconstruction the peak is defined as the voxel with the highest photon sum. To additionally control the amount of photons reconstructed in the peak voxel, the mean difference between the photons in the label peak and the reconstructed peak is used as an optimization feature.

To optimize the architecture the following hyperparameters have been optimized with regards of the optimization features:

- loss function,
- graph,
- layer types,
- amount of layers,
- amount of features per layer,
- drop out,
- learning rate.

As the loss function describes how the output needs to be changed to fit the label and the graph describes how the data propagates, these are considered as especially relevant to improve the prediction. The loss function and the graph have been optimized in several steps, these are listed in the following subsections.

7.2.3 Loss Function

In figure 7.2 an optimization plot is shown. This plot compares different loss functions, which are labeled by abbreviations. The loss functions Cross Entropy Loss (CE), Mean Squared Error Loss (MSE), L1 Loss, Negative Log Likelihood (NLL), geometric loss function without 3D gradients (GeomA) and geometric loss function with 3D gradients (GeomB) are used. The MSE, CE and geometric loss functions are already introduced in chapter 4. The 3D gradients

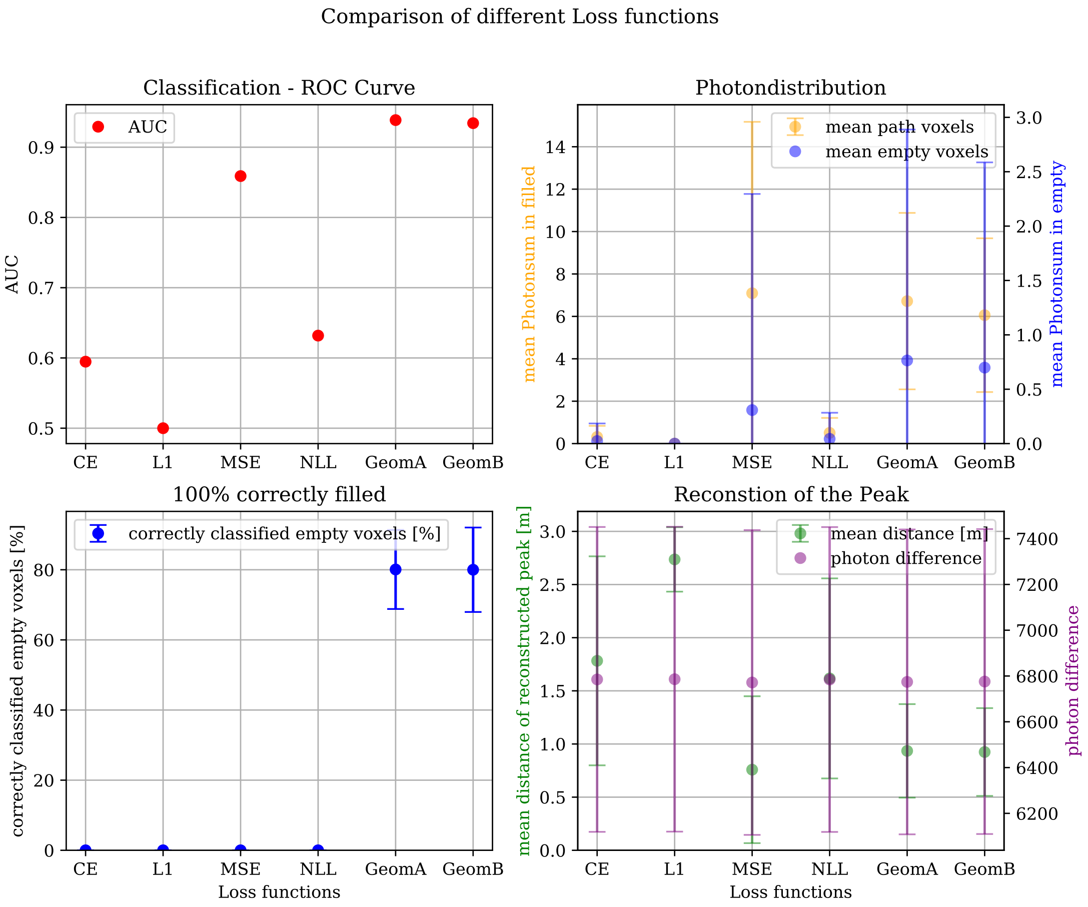


Figure 7.2: Optimization plots compares different loss functions in regard of the optimization parameters. The loss functions CE, MSE, L1 Loss, NLL, GeomA and GeomB are plotted. The different optimization features are discussed in section 7.2.2.

are discussed in the next section and L1 and NLL are not used furthermore and therefore not introduced. The figure shows the optimization features, already discussed.

For AUC[2] it sticks out, that the two geometric loss approaches perform far best. GeomB performs even better in classification than GeomA, which shows that the 3D gradients perform well in the geometric loss function. Both geometric loss functions reach up to 80 % correctly classified as empty for pc. The plots on the right hand side display the photon distribution qualities of the different loss functions. MSE performs best here, as the mean peak distance is minimal and the mean photon sum in filled voxels is maximal, in comparison to the results of the other loss functions. But all loss functions deliver far to small values for the mean photon sum in *path* voxels, as well as in the peak. This is displayed in the plot underneath. As no loss function performs well on all features, a more advanced loss function is needed. From this optimization GeomB is set as the basis for the loss function. It is combined with other loss

---

[2] For this work no errorbars for the AUC values are given, because here the AUC is only used to compare similar architectures on the same data set, with the same cuts. In the direct comparison only the AUC is considered the corresponding errors are neglected. For further ROC curves important for this works the errorbars on the different cuts are displayed.

terms to improve the prediction.

In the next step different combinations of loss terms have been tested to identify combinations which perform best. CE turned out to perform well in combination with GeomB. Additionally, MSE loss terms were included to improve the photon sum predicted in filled voxels. The optimization considered all loss terms and their specific adjustments, their weight and the data on which they work. The optimized characteristics are:

- single loss functions,
- loss term combinations,
- GeomB point cloud,
- GeomB 3D gradients,
- weight on 3D gradients,
- different approaches for weights on the points in the point cloud,
- data on which GeomB loss is used,
- CE loss weights of the two classed *path* and *background* voxels,
- CE loss cut, to distinguish between the two classes,
- data on which CE is used,
- amount of MSE terms and their purposes,
- data on which MSE terms are used,
- weights on all loss terms.

After optimization the loss function was set to a combination of five loss terms. These are GeomB, CE and three different MSE terms. The optimized loss terms are described in detail in subsection 7.3.2, where the optimized model is shown. All definitions which seem to be missing here and raise open questions are given there.

7.2.4 Graph

For a graph network the architecture of the graph is crucial, as it describes how the data propagates inside the graph. In this work, the graph is supposed to represent the geometry of the detector. Therefore, two types of components need to be implemented, the PMTs and the voxels. The input data is supposed to propagate from the PMTs into the detector volume, represented by voxels. To optimize the graph different approaches have been tested on how the edges are defined in the graph, different amounts of neighbors for the voxel nodes, the edges between PMT nodes and voxel nodes, edges between the PMTs and the initialization of

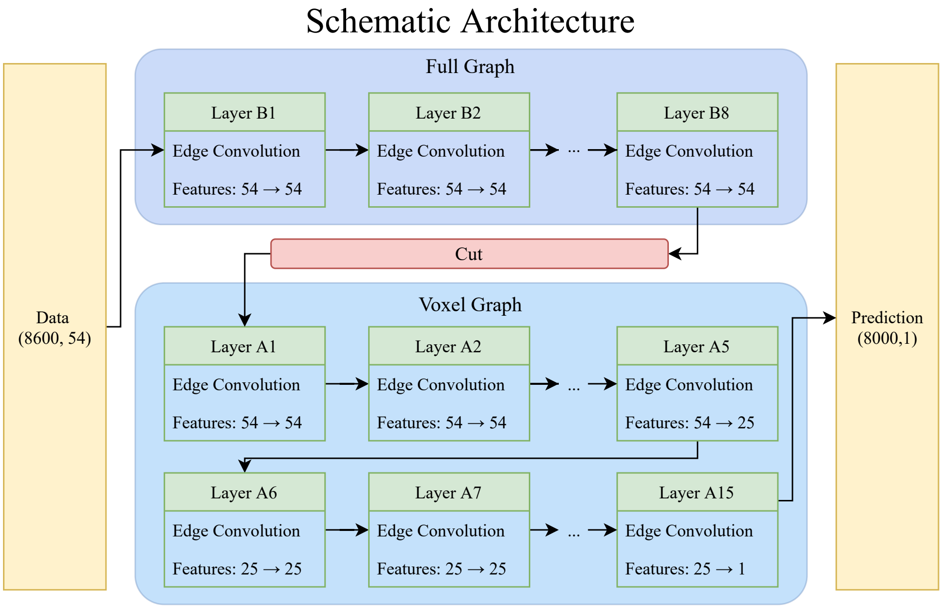


Figure 7.3: The optimized architecture is displayed schematically. The architecture is split in two parts, both using different graphs. First the Full Graph and afterwards the Voxel Graph is used. All layers are edge convolution layers, the used amount of features are denoted in the figure.

the voxel nodes have been considered as well. The optimized graph is described in detail in subsection 7.3.1.
For this architecture, the optimization showed that a second graph, without PMT nodes, after some layers of training improves the performance. This second graph only consists of voxel nodes and has also been optimized.

7.3 Architecture

In this section the optimized model is described, it is schematically displayed in figure 7.3. This architecture uses two different graphs. Firstly, the Full Graph and afterwards the Voxel Graph. The Full Graph has 600 PMT nodes and 8000 voxel nodes. The Voxel Graph only uses the voxel nodes. Which graph is used for which layer is illustrated by colored frames in figure 7.3. To transform the features from one graph into the other, the 600 PMT nodes in the graph and the 600×54 corresponding features are cut off. All layers are edge convolution layers, which are described in detail in chapter 4. The input data has 54 features[3], these are kept for the first layers B1 to B8. Afterwards, the PMT nodes are cut off. In the layers denoted with A, which stands for "after the cut", only the voxel nodes are considered. The features are decreased in layer A5 to 25 and in layer A15 to one. The output features of layer A15 are directly used as output. Additionally, drop out with rate 0.5 is used on the output features of layer B8, A3, A6 and A9.

---

[3] The input features are the 50 bin entries taken by the PMTs, the spatial coordinates of the corresponding PMT in the detector and a factor distinguishing between voxel and PMT nodes.

7.3.1 Optimized Graph

The basic idea of the graph is that the data taken by the PMT propagates backwards in the detector volume. The optimized graph consists of two types of nodes representing the PMTs and the voxels. The graph is supposed to represent the geometrical features of the detector. The 600 PMT nodes are connected over edges to the 8000 voxel nodes. Each PMT node is connected over edges to all voxel nodes within a radius of 1 m. In this way the PMT nodes are connected over edges to voxel nodes, which are up to 5 voxels deep inside the volume. Thereby, the data from the PMTs can propagate up to the 5th layer of voxels after one layer of training. The voxel nodes are connected over edges to their neighbors. Each central node has 26 neighbors. The nodes on the edge of the detector have less neighbors.
The features used in this architecture are the entries of the bins taken by the PMTs for every event. The detector has 600 PMTs, each measures for 50 ns with the resolution of 1 ns. The input has thereby the shape 600 × 50. Additionally, the coordinates of the PMTs are used. To distinguish between PMT and voxel node, the last features displays the object the node represents. Here, 1 represents PMT nodes and 0 voxel nodes. In total the number of input features is thereby 54. The voxel node features are supposed to fill with photons during training. Therefore, the first 50 features are initialized to be 0. The feature 51 to 53 are the x,y,z coordinate of the corresponding voxels and the 54th feature is 0, to represent the voxel character of the node.

7.3.2 Loss Function

The used loss function is a sum of five loss terms all serving a different purpose. The different loss terms and their weights have been optimized.

Geometrical Loss

The geometric loss function relies on the implementation in the geometric loss library [42], which is already described in chapter 4. This implementation does not work with the data structure of 20 × 20 × 20, which is used for the label in this work. For this loss term the data has been rearranged into a four dimensional point cloud. The four dimensions are the x,y,z coordinate of the voxel position and the corresponding amount of photons. To keep the computational effort low, not all 8000 voxel are transformed into points of the point cloud. Only voxels with a photon sum higher than a threshold are considered in this loss term. This threshold is optimized to 100 photons, which resembles a path length of 1 cm. This has the downside that voxels which are filled with less than 100 photons are not considered by this loss term, but it has the benefit that the computational effort decreases. This loss function has a high computational cost, as the optimal transport needs to be calculated. The here missed filled voxels are considered by other loss terms.

To calculate the loss, the points in the point clouds are weighted with the amount of photons in the resembling voxels. The returned gradients have the same shape as the input. The gradients on the reconstructed photons, can directly be applied. The gradients on the three spatial coordinates of the point cloud cannot be used directly. Therefore, the gradients need to be rearranged to operate on the $20 \times 20 \times 20$ output. The gradients for x,y,z coordinate display how the photons should be moved in space to fit the label distribution. These gradients are a key feature of geometric loss functions and thereby need to be implemented. To transform them into the output shape, the gradient is applied on the neighboring voxels in positive or negative x,y and z direction of the considered voxel. The direction correlates with the sign of the gradient. A negative gradient on the x coordinate of a point in the point cloud results in a gradient on the in negative x direction neighboring voxel. Thereby, the network retains the ability to rearrange the distribution, which is provided by a geometric loss function.

Cross Entropy Loss

The Cross Entropy Loss is defined as

$$\mathrm{CE}(y, c) = \frac{1}{\sum_c w_c} \cdot \sum_c w_c(-y[c] + \log(\sum_j \exp(y[j]))), \tag{7.1}$$

y resembles the output and c the classes of a data set. To use CE for the given output a cut needs to be set to distinguish between the two classes of voxels. The cut was optimized to be one photon. This is a very conservative threshold, which helps to minimize the number of photons reconstructed in *background* voxels. As this term only acts on classification, input in range $[0, 1]$ is used. Therefore, the output of the network is for this loss function limited to one. The two classes *background* and *path* voxels are weighted with the optimized factors $w_c = [0.001, 1]$. This factor is used for unbalanced data sets, which is the case for this reconstruction. The ratio of *path* to *background* is similar with a mean of $\approx 20/7980 \approx 0.0025/1$.

Mean Squared Error Loss

The MSE function is defined as

$$\mathrm{MSE}(y, t) = \sum_i^N \frac{(y_i - t_i)^2}{N}. \tag{7.2}$$

The prediction and label are denoted as y and t in a sample of N events.

In this model three MSE loss terms are used. All these terms have a differed purpose. As displayed in figure 7.2 MSE performs well on predicting the photon sums per voxel. This also showed during further optimizations.

MSE is used on *path*, *background* and peak voxels separately. The MSE terms *path* and *background* are supposed to improve the predicted amount of photons in the two voxel classes. The benefit of separating the two loss functions is that they can be weighted differently. If only one MSE term is used on both voxel classes, the loss function tends to decrease the predicted amount of photons in all voxels, as the label is highly unbalanced. This leads in many cases to an all zero prediction. The third MSE loss term is supposed to improve the predictions on the peak position and its photon sum. To do so only path voxels with more than 2000 photons in the label are considered. This cut has been optimized as well.

Total Loss Function

The total loss function is the weighted sum of all described loss terms

$$\text{Loss} = a\ \text{Geom} + b\ \text{CE} + c\ \text{MSE}_{\text{path}} + d\ \text{MSE}_{\text{background}} + e\ \text{MSE}_{\text{peak}}. \tag{7.3}$$

All loss terms are weighted and the weights are optimized. The magnitude of the different loss terms differ a lot, as the geometric loss is in the order of 10^8, while CE in the order of 1. Therefore, the weights on the CE and MSE terms are correspondingly big. The optimized weights are $a = 0.625$, $b = 1.1 \cdot 10^8$, $c = 10^4$, $d = 2.5 \cdot 10^4$ and $e = 10^3$.

7.4 Results

After discussing the model, the results produced by that model are discussed in this section. To train the network the data sets 1 and 3 have been used, with a total number of 2000 events. For validation the data set 2 with a total number of 1000 events has been used. The displayed results have all been produced on the validation data set. In this implementation the network does not learn from the validation data set, thereby the data set 2 can be considered as unknown. First the results on the optimization features are shown, afterwards additional analyses are displayed.

The plot in figure 7.4 shows the ROC curve for the reconstruction. The ROC curve method is described in chapter 4. The AUC displays the separation quality of two classes reached with the considered reconstruction method and is independent of cuts. The better the separation quality the more the curve moves in the upper right corner of the plot, to the coordinate $x = y = 1$. If the curve is at $x = y = 1$, a cut for the differentiation between the two classes exists, where all events for both classes are correctly classified. The classification strength of the here observed reconstruction is AUC= 0.9904.

The ROC curve is from $x = 0$ to around 0.8 approximately 1, which means in this interval all *background* voxels are classified correctly. And from $y = 0$ to around 0.9 close to 1 on the x axis, this means here around100 % of the *path* voxels are classified correctly. Along the ROC curve a cut can be chosen to obtain the preferred combination of correctly classified voxels of both classes.

Figure 7.5 shows the photon distributions of the two voxel classes. The upper plot shows the

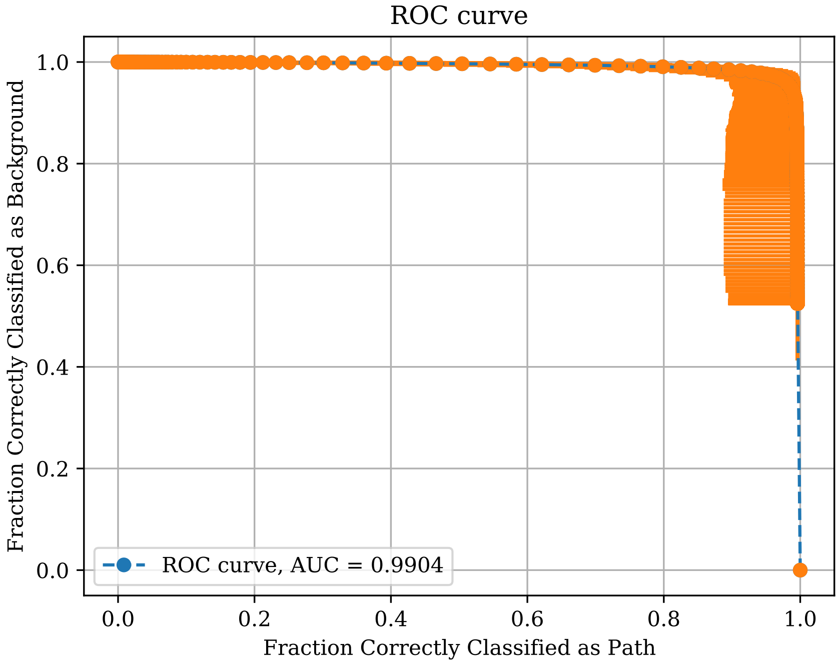


Figure 7.4: ROC curve of the optimized model to reconstruct the photon emission distribution with voxels. The AUC is 0.9904. The plot shows the ratio of correctly identified *background* voxles with regards to the rate of correctly identified *path* voxels, on the x axis. The ROC curve in blue is defined by the orange points, which are generated with different cuts. The ROC curve method is described in chapter 4.

distributions for the *background* and the lower for the *path* voxels. The distributions defined by the label and the reconstructed distributions are displayed in green and purple. In the upper plot the voxels should all be empty by definition. The mean amount of photons reconstructed in the *background* voxels is (59.5 ± 193.0) photons. The label photon distribution of the *path* voxels has a mean value of (1587.0 ± 1426.3) photons. The distribution has a second burst above 5000 photons. The voxels defining these burst are the peak voxels, where 5000 photons are emitted. The maximal path length inside one voxel is $\sqrt{3 \cdot 0.2^2} \approx 0.35\,\mathrm{m}$, for the edge length of a voxel of 0.2 m. This defines the maximal photon emission of a voxel without peak to ≈ 3500 photons. The gap between the two peaks of the distribution is between ≈ 3000 and ≈ 5000 photons. The reconstructed photon emission distribution does not follow the label distribution, but has its mean value at (1519.7 ± 634.0) photons. The maximal value in the reconstructed photon emission distribution is at around 5000 photons, while the maximal value of the label occurs at around 8000 photons. The plot shows that the second burst, defined by the peak voxels, is not matched by the reconstructed distribution.

The *pc* value for this reconstruction is 0 %. This means a cut set to identify 100 % of the path voxels correctly leads to 0 % correctly identified *background* voxels. Or in other words one or more *path* voxel in the data set are reconstructed to contain the minimal amount of photons and can thereby not be distinguished from the *background* voxels.

The mean peak distance is displayed in figure 7.6. It shows the distances between label and

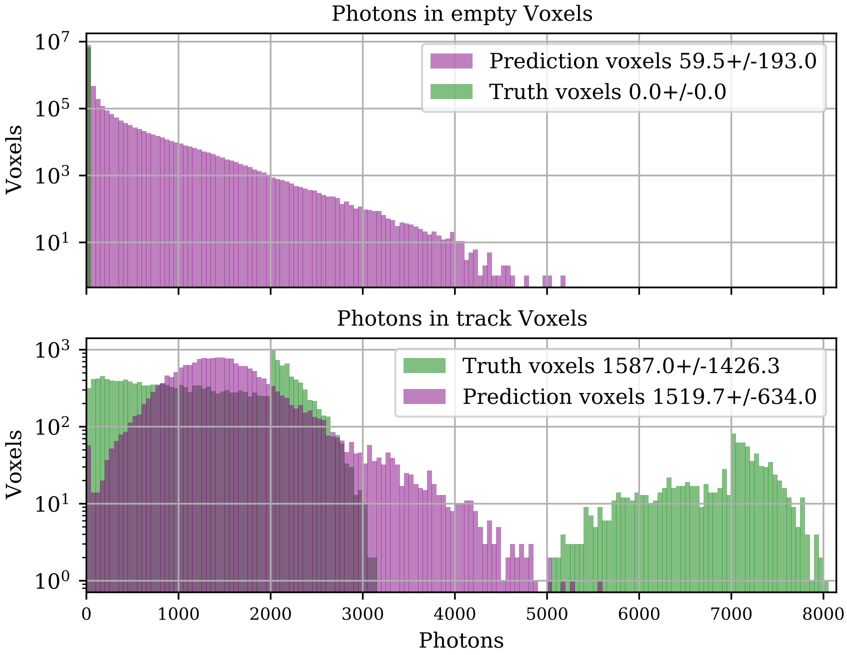


Figure 7.5: In this plot the photon distributions for the two classes of voxels are displayed. The upper plot shows the photon distribution of the *background* voxels as defined in the label in green and the reconstruction in purple. The lower plot shows the photon distribution of the *path* voxels with the same color code. Both plots share the same x axis, labeled in the lower plot.

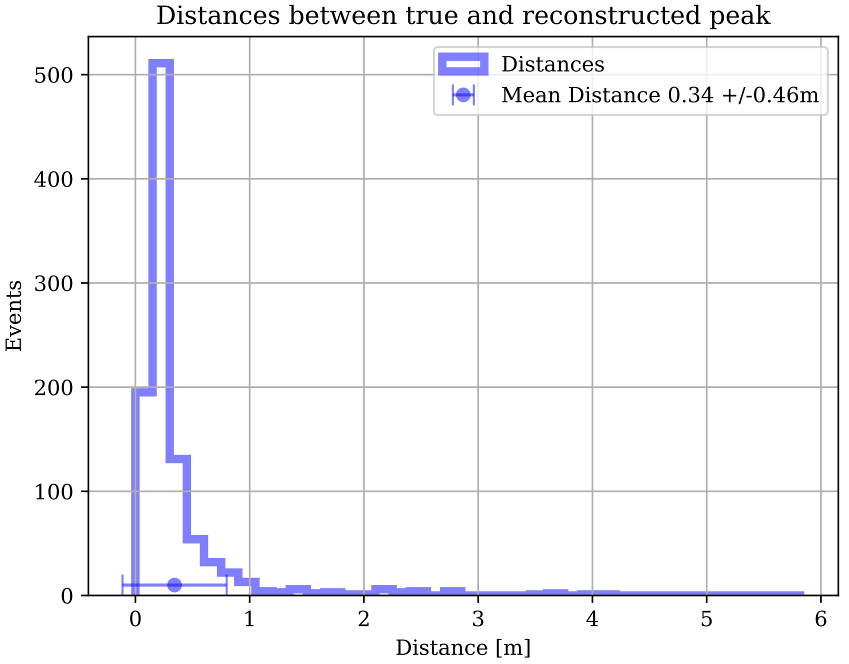


Figure 7.6: The histogram displays the distances between label peak and reconstructed peak on data Set 1. The reconstructed peak is defined as the voxel with the highest number of reconstructed photons. The distance is measured in meters. The mean distance is (0.36 ± 0.47) m. Every bin has the width of 0.15 m.

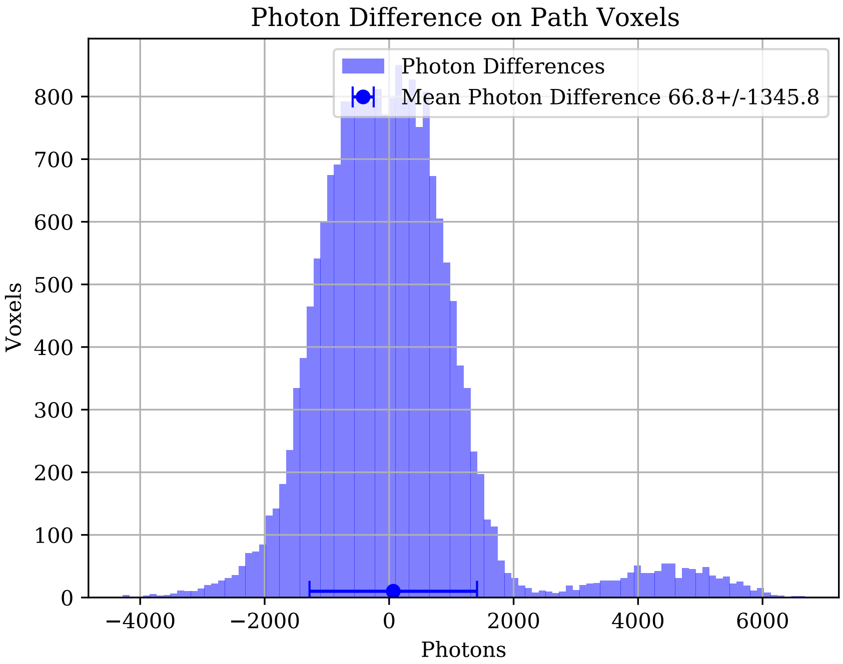


Figure 7.7: Difference between reconstructed photons and label on the *path* voxels. The intensity of the color reveals the amount of photons reconstructed in this voxel, a high intensity resembles a high number of photons.

reconstructed peak in meters. The peak coordinate is defined as the coordinate of the voxel, which holds the peak. For the reconstruction the peak is defined as the voxel for which the number of reconstructed photons is highest. Thereby, the coordinates of the peaks are discrete and their distances are discrete as well. In this reconstruction the mean distance is (0.36 ± 0.47) m. 19.5 % of the peak distances is 0 m, which means the peak is reconstructed correctly. For 74.6 % of the analyzed events the distance is below 0.35 m, these peaks are reconstructed correctly or in a neighboring voxel. In this case neighboring voxels are the 26 closest voxels, which directly surround the considered one. The bin width in the histogram is 0.15 m and the maximal bin is the second one, representing distances between 0.15 and 0.3 m. Voxels with a corresponding distance are the 6 neighbors, which share a wall with the considered voxel. In 13.7 % the peaks are reconstructed with a distance > 0.5 m and in 5.2 % the distance is > 1 m.

In figure 7.7 the photon difference between reconstruction and label on the *path* voxels is displayed. The distribution is a superposition of two Gaussian distributions. One has a large peak at around zero photons and the second one has a small peak at around ≈ 5000 photons. The smaller distribution belongs to peak voxels, these are systematically underrated. This can additionally be seen in figure 7.5. The mean of the full distribution is at (66.8 ± 1345.8) photons. The distribution could have been fitted and the reconstructed photons adjusted, so that the mean photon difference of the large Gaussian distribution becomes zero. But as the distribution seems to be only slightly of center but comparatively wide, it is abstained from this step. The mean photon difference in the peak voxel is displayed in figure 7.8. The mean difference between

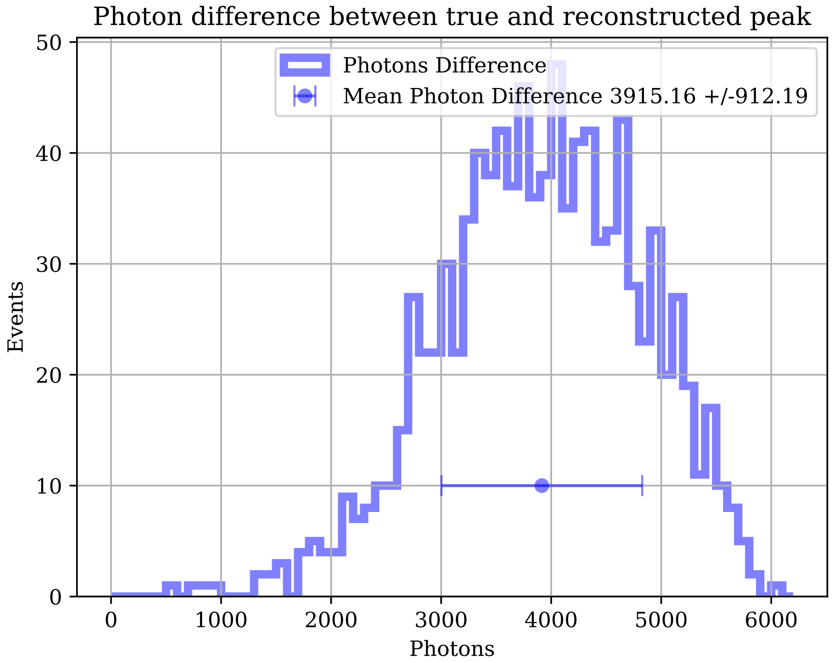


Figure 7.8: This histogram shows the difference in the number of photons in the peak voxel between reconstruction and label.

the reconstructed photons in the peak voxels and the photon peak in the label is (3915.16 ± 912.19) photons. As already displayed in figure 7.8, the amount of photons reconstructed as peak is too small by a minimum of 500 photons. This deficit is also visible in this plot.

7.4.1 Defining the cut

To further analyze the data a cut is set to distinguish between *path* and *background* voxels in the prediction. A natural choice is to set the cut to one photon. This would make every voxel in which at least one photon was reconstructed a *path* voxel and the others *background* voxels. But as visible in figure 7.5 the mean amount of photons reconstructed in the *background* voxels is around 60 photons. This means, the *background* voxels are in average filled with too many photons. A cut at one photon results in (99.68 ± 1.97) % correctly identified as *path* and (52.45 ± 10.39) % correctly identified as *background*. As visible in the ROC curve, presented in figure 7.4, other cuts allow a higher percentage of correctly identified *background* voxels, without a big loss of voxels correctly identified as *path*. To find a well performing cut the ratios of correctly identified as *path* and *background* voxels are plotted with regards to the cut. Additionally, the peak voxel is observed. As part of the reconstruction is finding the peak, a cut which would classify the peak as empty would not lead to a meaningful reconstruction. Figure 7.9 shows the described plot. The red line shows the percentage of peaks classified as *path*, as it declines for cuts above 600 photons, the cut needs to be set below. The percentage of correctly classified *path* and *background* voxels is plotted in blue and orange. The cut is set to maximize the ratio of correctly identified voxels for both classes. For balanced data the

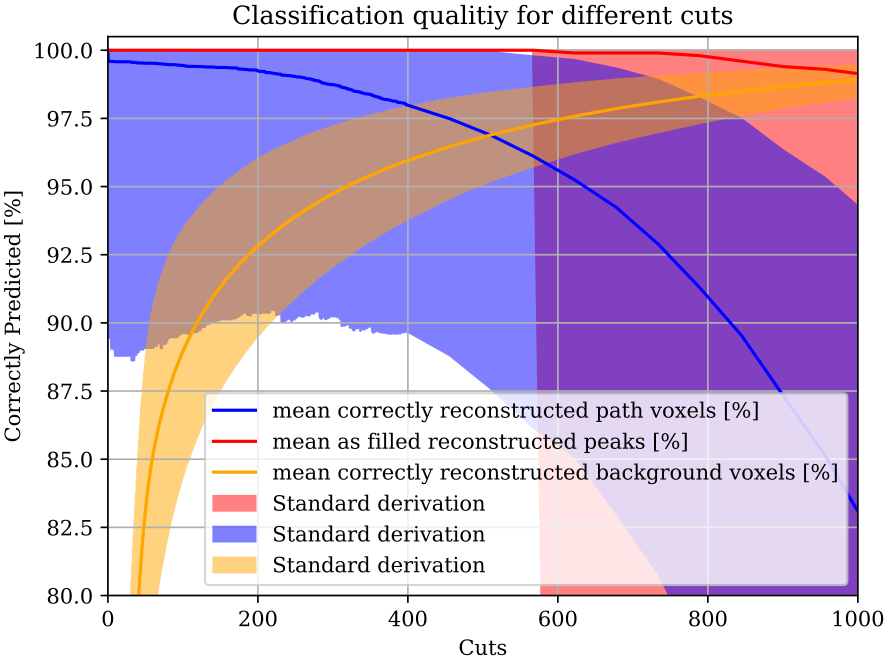


Figure 7.9: This plot shows the classification quality of the reconstructions with regards to the cut used to distinguish between the two voxel classes. The percentage of correctly filled *path* voxels is displayed in blue and of correctly identified *background* voxels in orange. Additionally, the percentage of as path classified peak voxels is displayed in red. The standard derivation are marked in the corresponding colors.

cut could be set to the intersection point of both functions. This would lead to approximately 97 % correctly classified for both classes and a cut at ≈ 500 photons. But the data is highly imbalanced as less than 1 % of the voxels are *path* voxels.

Conservative cut

The cut is set conservatively at 200 photons to correctly reconstruct more *background* voxels, with the backdraw of a minor decrease of correctly identified *path* voxels. With this cut about 93 % of the *background* voxels are correctly classified with only a small decrease in the amount of correctly identified *path* voxels. A cut at 200 photons represents a path length of 20 mm in one voxel. Voxels which have ≤ 200 photons by simulation, will be identified as *background*, if the amount of photons is reconstructed correctly.
With this cut at 200 photons, the percentage of correctly identified voxels of the two classes *path* and *background* is defined. A corresponding plot is displayed in figure 7.10. The percentage of correctly identified *path* voxel is (99.24 ± 2.84) % and for *background* voxels (92.81 ± 3.24) %. For more than 85 % of the analyzed data, all *path* voxels are correctly identified. Thereby, the path can be narrowed to 8 % of the detector volume by losing only 0.8 % of the *path* voxels. By setting this cut ≈ 0.8 % of the *path* voxels are not classified correctly as path. The emitted photons in these voxels are displayed in figure 7.11. As the cut is set to 200 photons, voxels

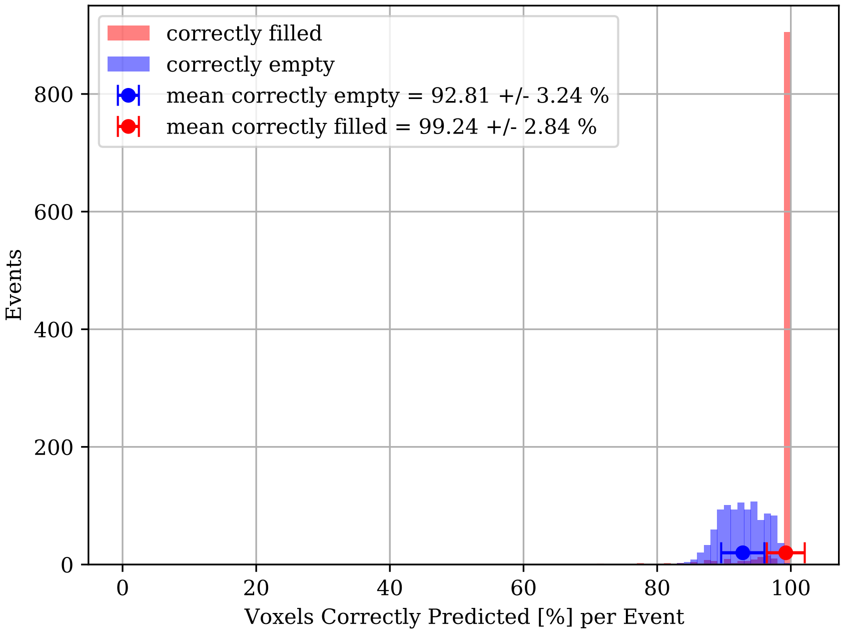


Figure 7.10: Percentage of correctly identified voxels of the two classes *path* and *background*. The *path* voxels are displayed in red and the *background* voxels in blue. Additional to the histogram the mean values are displayed.

in which less than 200 photons have been emitted are lost, if the photon sum is reconstructed correctly. In this reconstruction 29 path voxels are missed due to the cut. Additionally, voxels are missed in which too little photons have been reconstructed. These have a mean value of (822.2 ± 721.6) photons. 91 voxels are missed because of that.

In figure 7.12 the classification quality is plotted against the path length in voxels. The path lengths are shown by a histogram in green and the percentages of correctly identified voxels as *path* and *background* are displayed in orange and blue. This plot reveals that for longer tracks the percentage of correctly identified *background* voxels decreases. Additionally, some events show for shorter tracks lower percentages of correctly identified *path* voxels with a minimum at around 70 %. As these short paths only consist of less than 10 voxels, missing one already leads to a big decrease of > 10 %.

In figure 7.13 a sample reconstruction is displayed. The reconstructed filled voxels and the label *path* voxels are plotted in blue and red. The transparency of the voxels resembles the inverse photon sum in the considered voxel with regards to the other voxels. The most intense colored voxel resembles the peak. The event is evaluated with a cut at 1 photon and the discussed cut at 200 photons. Both times the reconstruction encloses the label path. On the left the plot for the cut at one photon is displayed. The reconstruction fills nearly half of the voxels. Most of these voxels are transparent, which means only a small number of photons was reconstructed in those. The cut at 200 photos, on the right, improves the number of falsely identified *background* voxels. With this cut the volume around the path is filled in the reconstruction. Additionally,

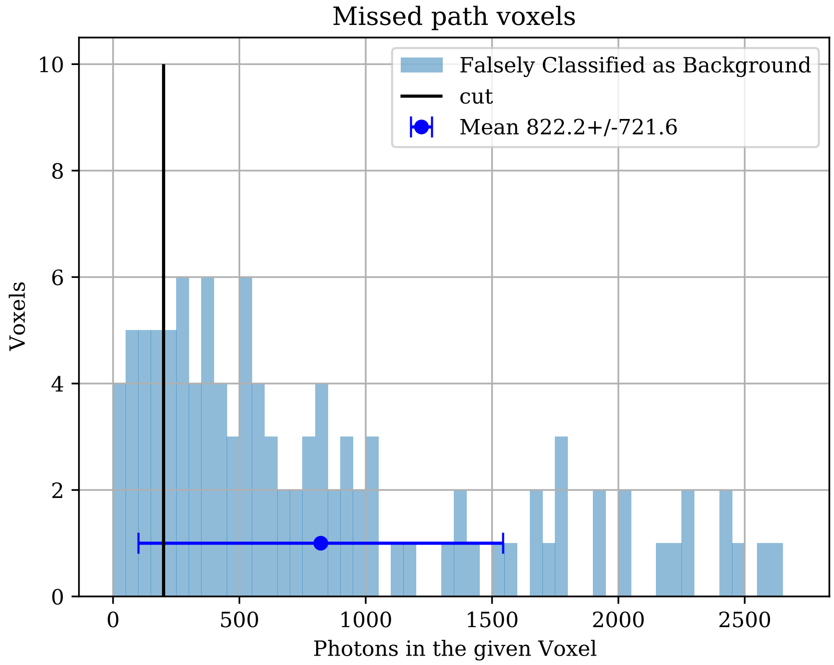


Figure 7.11: Photons emitted in voxels, which are falsely classified as *background* voxels.

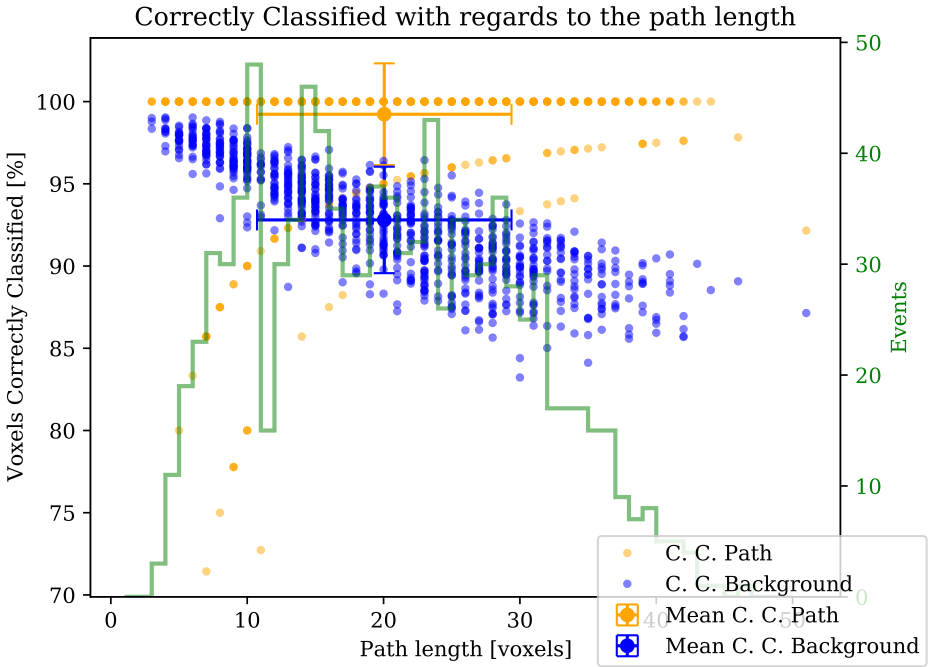


Figure 7.12: This plot shows a histogram of the path lengths in voxels in green, the corresponding y axis is on the right. Additionally the percentages of correctly identified voxels as *path* and *background* are displayed in orange and blue, with regards to the path length.

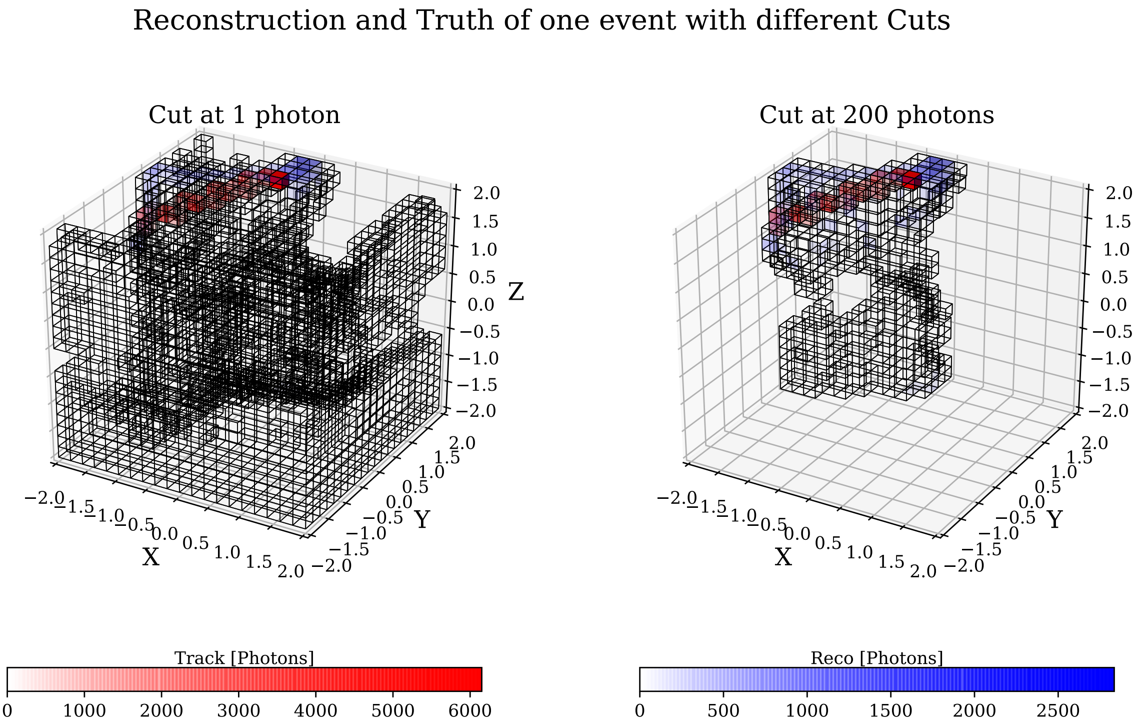


Figure 7.13: This plot shows the comparison of the same event with two different cuts at 1 and at 200 photons. The red voxels display the label and the blue ones the reconstruction. The more transparent the color is, the less photons are assigned to this voxel. The two color labels are valid for both plots.

filled voxels can be found from the path in negative z direction. As these are transparent, they are not predicted to be filled with many photons. A higher cut could further improve the amount of falsely identified *background* voxels and minimize the volume defined by filled voxels around the path. This event is only a sample, in the appendix in figures 1-9 nine more sample events are displayed. In all cases the amount of falsely identified *background* voxels decreases with the cut at 200 photons, but the filled voxels are still far more than the labels provides. Additionally a second cut is analyzed.

Cut with a higher Background discrimination

The conservative cut approach at 200 photons, tries to keep as many *path* voxels filled as possible by increasing the amount of correctly identified *background* voxels to an amount, which helps to narrow the volume polluted by the path. In this subsection a higher cut is taken to improve the amount of correctly identified *background* voxels and further analyze the reconstructed path structure.

With regards to figure 7.9, the cut is now set with the goal to discriminate a higher *background* rate. Still all peaks should be classified as *path* and the rate of correctly classified *path* voxels should be high. Therefore, the cut is set at 550 photons. With this cut about 97 % of both classes are correctly classified and all peak voxels are classified as *path*.

A cut at 550 photons results in (96.33 ± 6.58) % voxels correctly classified as *path* and (97.17 ±

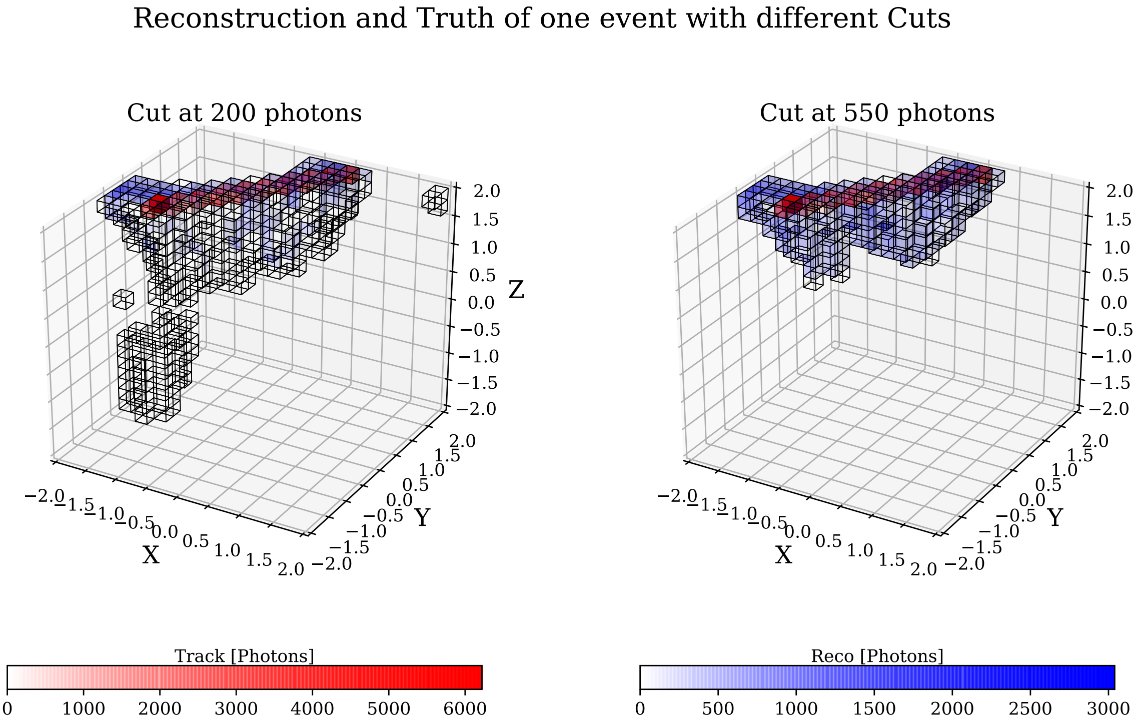


Figure 7.14: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction. The more transparent the color is, the less photons are assigned to this voxel. The two color labels are valid for both plots.

1.48) % voxels correctly classified as *background*, the corresponding plot is displayed in the appendix, as figure 10.

Figure 7.14 shows the comparison of one event evaluated with different cuts at 200 and 550 photons. The label track is displayed in red and the reconstruction in blue. For the cut at 200 photons the track is enclosed by the reconstruction. Additionally, voxels are reconstructed to be filled at negative x and negative z. These are not filled for the cut at 550 photons. For this event the discrimination of the path is done better by the cut at 550, as all *path* voxels are reconstructed correctly and less *background* voxels are reconstructed falsely. But all reconstructed voxels in the right plot have approximately the same color intensity, which shows they contain similar amounts of photons. This shows that the filled *path* and *background* voxels are hard to distinguish in the reconstruction. In the appendix nine other sample events are displayed with the two cuts at 200 and 550 photons. For all nine sample events the higher cut at 550 photons leads to a smaller amount of falsely identified *background* voxels. In some cases, like displayed in figure 11, the cut at 550 photons creates filled voxel enclaves, which are not connected to the main filled voxel cloud.

In figure 7.15 the percentages of correctly classified in dependency of the path length is displayed, with the new cut at 550 photons. In comparison the same plot for the cut at 200 photons is displayed in figure 7.12. With the new cut the percentage of correctly identified *background* voxels decreases less with the increase of path lengths. The new cut leads to a more stable

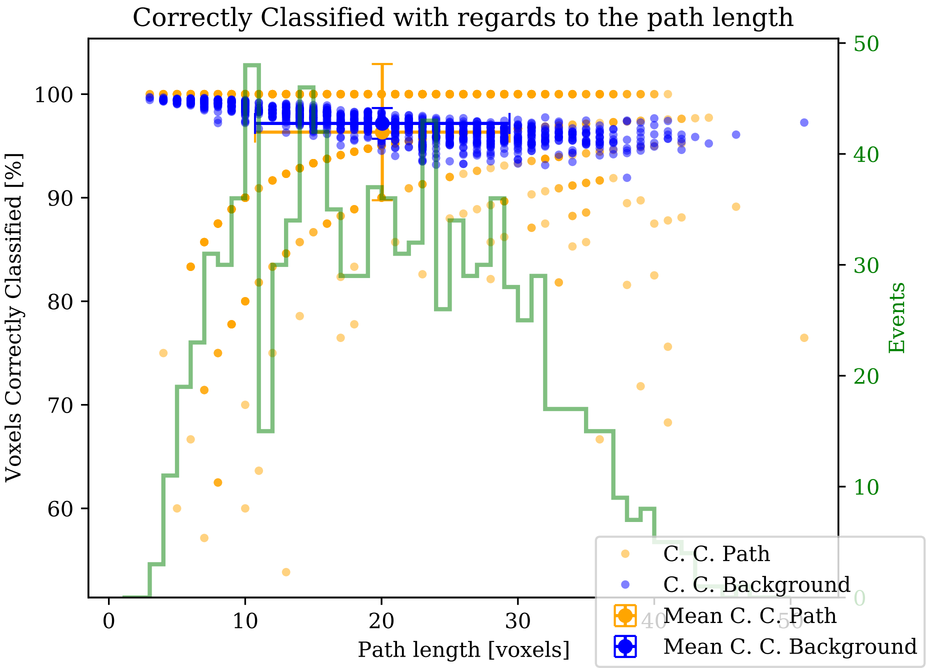


Figure 7.15: This plot shows a histogram of the path lengths in voxels in green, the corresponding y axis is on the left. Additionally the percentages of correctly identified as *path* and *background* are displayed in orange and blue, with regards to the path length.

amount of correctly identified *background* voxels. The backdraw of this cut is the decreased amount of events, for which all *path* voxels are correctly classified, to ≈ 60 %. This percentage is at ≈ 85 % with the cut at 200 photons.

The downside of the higher cut is, that more voxels are falsely identified as background. In this case 3.7 %. In figure 7.16 the photons per missed *path* voxel are displayed. As the cut is set at 550 photons, voxels which are simulated with a path length below 55 mm, are missed, when reconstructed correctly. For the analyzed data set, 263 voxels are missed, due to that, and 390 voxels because of too little photons reconstructed in the given voxel. The mean number of photons missed per voxel is (933.0 ± 730.4) photons.

7.4.2 Summary

To successfully reconstruct the photon emission distribution with voxels, the voxels need to be classified correctly in *path* and *background* and the number of photons per voxel need to match the label. The here presented results reach an AUC of 0.9904 for the analyzed data set. This is a good value for the discrimination of two classes, as the optimum is one. The number of correctly identified *path* and *background* voxels depend on the chosen cut, which distinguishes the two classes. The ROC curve displays, that combinations > 80 % *background* and > 90 % *path* correctly identified could be chosen. Less than one percent of the voxels are filled by the muon track, therefore it is important to correctly reconstruct as many *path* voxels as possible. The *background* voxels are important, to narrow down the detector volume which includes

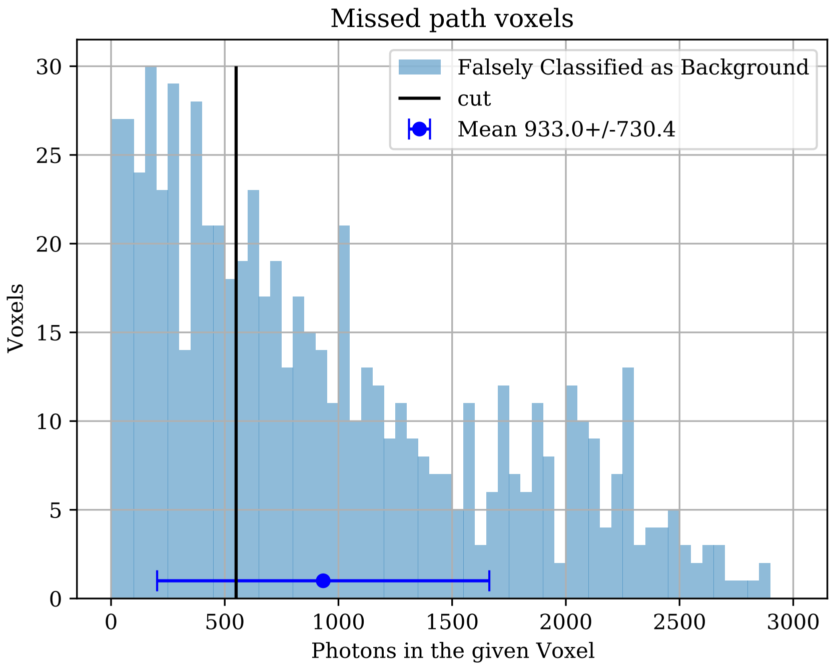


Figure 7.16: Photons per voxel in voxels, which are falsely classified as background voxels.

the track. In the last section two cuts, at 200 and at 550 photons are compared. The first one narrows (99.24 ± 2.84) % of the *path* voxels to 7 % of the detector volume, while the bigger one narrows (96.33 ± 6.58) % of the *path* voxels down to 3 % of the detector. In both cases all peak voxels in the data are correctly identified as path.

Additionally, the peak of the reconstructed distributions are analyzed. The peak positions are reconstructed with a mean distance of (0.36 ± 0.47) m, between reconstructed and label peak coordinates. The peak is reconstructed to the correct voxel in 19.5 % and for 74.6 % of the analyzed events the distance is below 0.35 m. For 85 % of the analyzed data the peak was reconstructed with a distance $\leq$ 0.5 m. In this case the peak can be enclosed to 0.8 % of the detector volume.

Additionally, the photon emission distribution has been analyzed. The here presented reconstruction underestimates the amount of photons reconstructed in the peak, in average (3915.16 ± 912.19) photons are missing. For the *background* voxels a mean of around 60 photons is reconstructed and for the *path* voxels a mean of (1519.7 ± 634.0) photons. Even tough the mean number of photons per *path* voxel is similar with (1587.0 ± 1426.3) photons, the widths of the two distributions differ by a factor of > 2. The photon distribution has two issues, on the one hand to many voxels are filled with photons, this leads to an overestimation of the total photon sum per event. On the other hand the differences of photons reconstructed in the *path* voxels and the label have a too wide spread. All in all, the reconstruction shows good results for the classification between empty and filled voxels and the peak position. But the presented reconstruction method does not deliver good photon emission distributions, as neither the photon sum per event, nor the photon sum per peak voxel is reconstructed with a bearable precision. Therefore, the provided reconstruction method needs to be developed further to use it for both spatial reconstruction and a reconstruction on the photon emission per voxel.

Chapter 8

Conclusion

In this work two reconstruction methods are developed, which reconstruct a muon track in a liquid scintillation detector. On the one hand the coordinate reconstruction, which reconstructs the track using characteristic points. On the other hand the voxel reconstruction, which reconstructs the photon emission distribution in the detector with the help of voxel representation. In chapter 6 and 7 the two developed graph convolution networks and the obtained results are presented. In this chapter these results are shortly summarized and interpreted. Afterwards, an outlook is given.

8.1 Summary and Conclusion

8.1.1 Coordinate Reconstruction

For this reconstruction method the muon track is described by three points defining the track, which are the start, peak and end point of the track inside the detector. As the simulated muon tracks are straight with a discrete peak at a random position along the track, these coordinates define the track. The used data is Toy Monte Carlo simulated. The goal of the coordinate reconstruction is firstly to reconstruct these points. Secondly, the graph convolution network architecture was tested on whether or not it is a suitable method for the given task.
The results from this reconstruction can be described by the mean distances between reconstructed and label for the three characteristic points. On the start point a mean distance of $d_S = (0.16 \pm 0.20)\,\mathrm{m}$ was reached. The start points are the simplest to reconstruct, as in the used simulation the measurement starts with the track entering the detector. Thereby the PMT with the first hit needs to be the one enclosing the start point. The end points are reconstructed with a mean distance of $d_E = (0.21 \pm 0.11)\,\mathrm{m}$. The peak points are reconstructed with a mean distance of $d_P = (0.22 \pm 0.14)\,\mathrm{m}$. This is the biggest mean distance for the three reconstructed coordinates. The peak is simulated on a random position along the track. This is typically not on one of the detector skins, where the start and end points are. Therefore, the peak coordinate has three free parameters and is thereby the most difficult to reconstruct. With this method $85\,\%$ of the peaks in the analyzed data set are reconstructed with a distance to the label $\leq 0.4\,\mathrm{m}$.

These peaks can be narrowed to 0.4 % of the detector volume, which is 64 m^3.
The motivation for the coordinate reconstruction was to find out whether or not a GCN is a suitable architecture for the reconstruction of muon tracks in a liquid scintillator. From the presented results this method was successful on the given task.

8.1.2 Voxel Reconstruction

For the voxel reconstruction the photon emission distribution is supposed to be reconstructed in voxel representation. The voxel representation displays on the one hand the spatial extent of the muon tracks. The voxels through which the tracks pass are classified as *path* voxels and thereby define the tracks. On the other hand and additional to the previous reconstruction the photon emission distribution is reconstructed. The voxel in which the most photons have been emitted is defined as the peak voxel. The same simulation as for the previous reconstruction method is used, but here the photon emission distribution in voxel representation in used as label.
The results of this reconstruction correspond either to the spatial reconstruction quality or the photon emission reconstruction quality of the used method. The spatial reconstruction quality is defined by the volume in the detector, which can be discriminated as not filled by the track. For this reconstruction (99.24 ± 2.84) % of the path voxels can be discriminated to 7 % of the detector volume, with the definition that all voxels reconstructed with $\leq$ 200 photons are considered as empty. 100 % of the peak voxels are reconstructed as path voxels. Additionally, the peak voxels are reconstructed with a mean distance to the label of (0.36 ± 0.47) m. For 85 % of the analyzed data the peak was reconstructed with a distance $\leq$ 0.5 m. These peaks can be narrowed to 0.8 % of the detector volume. Summarized, the separation between the two classes is performed well, with AUC= 0.9904. Additionally, the peak was reconstructed nicely, as it is with both provided cuts on all given data classified as path.
For the photon emission distribution the mean photon differences on *path*, *background* and peak voxels were analyzed. For the background voxels a mean of around 60 photons is reconstructed. In the *background* no photon was emitted in the simulation. For the *path* voxels a mean value of (1519.7 ± 634.0) photons was reconstructed. Even tough the mean number of photons per *path* voxel is similar with (1587.0 ± 1426.3) photons, the widths of the two distributions differ by a factor of > 2. For the peak voxel in average (3915.16 ± 912.19) photons are missing. All in all, too many voxels are filled in the reconstruction with photons. The photons per path voxel have a similar average value as the simulation, but the spread was not reconstructed correctly. Additionally, the peak voxel is drastically under estimated by the reconstruction. This makes the reconstructed photon emission distribution unreliable.
All in all, this reconstruction method has good spatial qualities, but the reconstructed photon emission distribution is not reliable. The spatial discrimination of *background* voxels could be developed further, to narrow muon tracks to path-like fractions of the detector. But to use the

photon emission distribution the presented work does not provide a sufficient precision and needs therefore further development.

In both presented reconstruction methods the usage of graph convolution networks is promising. For the presented implementations nice results for the spatial reconstruction are obtained. The coordinate reconstruction shows, that already a simple approach delivers a good spatial resolution. Therefore, the implementation of the geometrical structure in the graph seems to obtain the expected benefits.

8.2 Outlook

The motivation of this work is an improved muon veto for large liquid scintillation detectors like JUNO. In the JUNO detector a main background source for reactor neutrinos are cosmogenics. These occur in showers along muon tracks inside the detector. JUNOs muon veto approach is to veto the whole LSD, or a cylindrical volume around a muon track, depending on the quality of the muon track measurements in the Cerenkov detector and the muon trackers. It has a predicted efficiency of 83 % on the IBD, which is the detection reaction for reactor neutrinos at JUNO. This is discussed in detail in chapter 3.3. A reconstruction method based on the presented work could drastically improve the muon veto. For the voxel reconstruction approach the peak along the muon track, which resembles a shower, could be discriminated to 7 % of the detector volume, for all peaks in the considered data set. A reconstruction method like this could drastically decrease the dead time for JUNO, which would improve the efficiency on the IBDs. As neutrinos interact rarely and the expected amount of reactor neutrino events at JUNO is 84 /day, any improvement on the IBD efficiency of the muon veto increases the statistics of the experiment.

The here presented methods were trained and evaluated on Toy Monte Carlo data. The simulated detector has the dimensions of $4 \times 4 \times 4\ \mathrm{m}^3$ and is thereby very small in comparison to the JUNO detector. An optical coverage of 100 % has been implemented, while JUNO has around 75 %. Additionally, the simulation does not include, different paths, Cerenkov light, scattering on the PMTs, attenuation and disturbances by other processes in the detector. Thereby, the obtained results are not directly comparable to real detectors like JUNO and only give a promising outlook on further developments of this reconstruction method and its applications.

To further improve the presented architecture, the photon emission distribution would need to be improved. Even though the loss function was optimized with a lot of effort, the obtained loss function does deliver good results on the photon emissions. One possible idea would be a loss function specially developed for sparse data. Additionally, the geometric loss function does not work on the $20 \times 20 \times 20$ data structure. An improved geometric loss function for this data structure could also improve the results. One of the problems of the photon emission distribution is that the photon emission in the *path* voxels are systematically under estimated, while the total photon sum is too large. With a loss term on the total photon sum the reconstructed photon sum could be improved, which would result in a smaller amount of photons per filled voxel, but could after more training result in less filled voxels, with a higher photon sum. Another option is to include a some dynamic edge convolution layers after the used architecture. These rearrange the graph with every iteration by connecting the nodes to their closets neighbors in the feature space. This could improve the predictions.

With an improved photon emission distribution the network could be used on more realistic data. Firstly, further physical processes could be included, afterwards the detector could be simulated more realistically, by implementing a smaller optical coverage and random noise. With more realistic data the presented approach could be further developed and the results validated.

Even though the photon emission reconstruction did not deliver the hoped for precision, the spatial reconstruction of the muon tracks performed quite good for both reconstruction approaches. The photon emission distribution has the benefit, that it allows an energy reconstruction from the emitted photons. This is not possible for the spatial reconstruction as implemented in this work. But for muon induced shower identification of muon track identification the spatial information of the reconstruction is sufficient. Therefore, with a lot of further development and research the coordinate reconstruction could become useful for muon vetoes or comparable applications. But to validate the obtained results firstly the developed methods need to be approved on more realistic data.

8.2.1 Potential Application at Theia

The Theia experiment offers an outlook in the further future. Theia is a proposed long baseline experiment, with superior features. It combines liquid scintillation light with a water Cerenkov detector and a new generation of light detectors, the LAPPDs. Theia is described in detail in chapter 3.4.

Due to the wbLS Theia unites the benefits of wCD and LSD. Additionally, the LAPPDs have a great spatial resolution. As the scintillation light is proportional to the deposited energy, the energy can be reconstructed much better than in a wCD. In case the two light sources can be distinguished, the Cerenkov light allows a good directional reconstruction. The missing pieces are exceptional techniques analyzing the given data. With a good spacial reconstruction of the deposited energy, Theia could work as a light-based time projection chamber (TPC). TPCs are particle detectors, which detect the particle tracks inside the detector volume, over ionization processes. Thereby, detailed discrete information along the particle track is obtained, without direct interactions between the particle and the measuring elements of the detector. In contrast to LSD and wCD, were the information about the track is convoluted with the detector geometry and measured on the walls of the detector. For Theia the light-based TPC approach would be obtained by reconstruction algorithms, which return discrete information about the track inside the detector volume, using light as signal.

The here presented work is supposed to take a first step into the direction of reconstruction methods like this. The GCN approach taken in this work obtained good results on the spatial resolution. A similar approach could be used for Theia. The combination of the exceptional spatial resolution of the LAPPDs with a graph network architecture including the geometric structure of the detector, could be a promising approach. But improved results on the energy reconstruction would be necessary to obtain the desired TPC functionality. To develop graph based reconstruction methods for real detectors, more research on this topic needs to be done.

Bibliography

[1] Fengpeng An, Guangpeng An, Qi An, Vito Antonelli, E. Baussan, John Beacom, Leonid Bezrukov, Simon Blyth, Riccardo Brugnera, M. Avanzini, Jose Busto, Anatael Cabrera, Hao Cai, Xiao Cai, Antonio Cammi, Guofu Cao, Jun Cao, Yun Chang, Shaomin Chen, and Jiaheng Zou. Neutrino physics with juno. *Journal of Physics G: Nuclear and Particle Physics*, 43, 07 2015.

[2] David Meyhöfer. *3D Topological Reconstruction in JUNO applied to GeV events.* PhD thesis, Hamburg University, 2020.

[3] Frank F Deppisch. *A Modern Introduction to Neutrino Physics.* 2053-2571. Morgan and Claypool Publishers, 2019.

[4] Wikimedia Commons. Standard model of elementary particles, 2020.

[5] M. Aker, K. Altenmüller, M. Arenz, M. Babutzka, J. Barrett, S. Bauer, M. Beck, A. Beglarian, J. Behrens, T. Bergmann, and et al. Improved upper limit on the neutrino mass from a direct kinematic method by katrin. *Physical Review Letters*, 123(22), Nov 2019.

[6] C. S. Wu, E. Ambler, R. W. Hayward, D. D. Hoppes, and R. P. Hudson. Experimental test of parity conservation in beta decay. *Phys. Rev.*, 105:1413–1415, Feb 1957.

[7] J. H. Christenson, J. W. Cronin, V. L. Fitch, and R. Turlay. Evidence for the 2π decay of the k_2^0 meson. *Phys. Rev. Lett.*, 13:138–140, Jul 1964.

[8] E. Fermi. Versuch einer Theorie der β-Strahlen. I. *Zeitschrift fur Physik*, 88(3-4):161–177, March 1934.

[9] Y. Fukuda, T. Hayakawa, E. Ichihara, K. Inoue, K. Ishihara, H. Ishino, Y. Itow, T. Kajita, J. Kameda, S. Kasuga, K. Kobayashi, Y. Kobayashi, Y. Koshio, M. Miura, M. Nakahata, and S. et al. Nakayama. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, Aug 1998.

[10] Ziro Maki, Masami Nakagawa, and Shoichi Sakata. Remarks on the Unified Model of Elementary Particles. *Progress of Theoretical Physics*, 28(5):870–880, 11 1962.

[11] Q. R. Ahmad, R. C. Allen, T. C. Andersen, J. D. Anglin, G. Bühler, J. C. Barton, E. W. Beier, M. Bercovitch, J. Bigu, S. Biller, and et al. Measurement of the rate of interactions produced by b8 solar neutrinos at the sudbury neutrino observatory. *Physical Review Letters*, 87(7), Jul 2001.

[12] S T Petcov. Neutrino masses, mixing, majorana CP-violating phases and ( )0 decay. *New Journal of Physics*, 6:109–109, aug 2004.

[13] Wikimedia Commons. Oscillations electron long, 2011.

[14] A. Yu. Smirnov and S. P. Mikheev. Neutrino oscillations in matter with varying density. In *6th Moriond Workshop: Massive Neutrinos in Particle Physics and Astrophysics*, pages 355–372, 1986.

[15] A Yu Smirnov. The msw effect and matter effects in neutrino oscillations. *Physica Scripta*, T121:57–64, Jan 2005.

[16] Dmitry V. Naumov. The sterile neutrino: A short introduction. *EPJ Web of Conferences*, 207:04004, 2019.

[17] K Abe, Tetsuo Abe, Hiroaki Aihara, Y Fukuda, Y. Hayato, K Huang, A. Ichikawa, M Ikeda, Kunio Inoue, H. Ishino, Y. Itow, T. Kajita, J. Kameda, Y Kishimoto, Masayuki Koga, Y. Koshio, K Lee, A. Minamino, M Miura, and Masayuki Yokoyama. Letter of intent: The hyper-kamiokande experiment — detector design and physics potential —. 09 2011.

[18] M.V. Diwan, V. Galymov, X. Qian, and A. Rubbia. Long-baseline neutrino experiments. *Annual Review of Nuclear and Particle Science*, 66(1):47–71, Oct 2016.

[19] A. Gando, Y. Gando, K. Ichimura, H. Ikeda, K. Inoue, Y. Kibe, Y. Kishimoto, M. Koga, Y. Minekawa, T. Mitsui, T. Morikawa, N. Nagai, K. Nakajima, K. Nakamura, K. Narita, I. Shimizu, Y. Shimizu, J. Shirai, F. Suekane, A. Suzuki, H. Takahashi, N. Takahashi, Y. Takemoto, K. Tamae, H. Watanabe, B. D. Xu, H. Yabumoto, H. Yoshida, S. Yoshida, S. Enomoto, A. Kozlov, H. Murayama, C. Grant, G. Keefer, A. Piepke, T. I. Banks, T. Bloxham, J. A. Detwiler, S. J. Freedman, B. K. Fujikawa, K. Han, R. Kadel, T. O'Donnell, H. M. Steiner, D. A. Dwyer, R. D. McKeown, C. Zhang, B. E. Berger, C. E. Lane, J. Maricic, T. Miletic, M. Batygov, J. G. Learned, S. Matsuno, M. Sakai, G. A. Horton-Smith, K. E. Downum, G. Gratta, Y. Efremenko, O. Perevozchikov, H. J. Karwowski, D. M. Markoff, W. Tornow, K. M. Heeger, and M. P. Decowski. Constraints on θ_{13} from a three-flavor oscillation analysis of reactor antineutrinos at kamland. *Phys. Rev. D*, 83:052002, Mar 2011.

[20] Y. Abe, C. Aberle, J. C. dos Anjos, J. C. Barriere, M. Bergevin, A. Bernstein, T. J. C. Bezerra, L. Bezrukhov, E. Blucher, N. S. Bowden, and et al. Reactor e disappearance in the double chooz experiment. *Physical Review D*, 86(5), Sep 2012.

[21] Seon-Hee Seo. New results from reno, 2013.

[22] F. P. An, J. Z. Bai, A. B. Balantekin, H. R. Band, D. Beavis, W. Beriguete, M. Bishai, S. Blyth, K. Boddy, R. L. Brown, and et al. Observation of electron-antineutrino disappearance at daya bay. *Physical Review Letters*, 108(17), Apr 2012.

[23] Sandhya Choubey, Debajyoti Dutta, and Dipyaman Pramanik. Invisible neutrino decay in the light of nova and t2k data. *Journal of High Energy Physics*, 2018(8), Aug 2018.

[24] Tristan Blackburn. *Measurement of* Δm^2_{32} *and* $\sin^2\theta_{23}$ *using Muon Neutrino and Antineutrino Beams in the NOvA Experiment.* PhD thesis, Sussex U., 2019.

[25] K. Abe et al. Constraint on the matter–antimatter symmetry-violating phase in neutrino oscillations. *Nature*, 580(7803):339–344, 2020. [Erratum: Nature 583, E16 (2020)].

[26] P. F. de Salas, D. V. Forero, S. Gariazzo, P. Martínez-Miravé, O. Mena, C. A. Ternes, M. Tórtola, and J. W. F. Valle. 2020 global reassessment of the neutrino oscillation picture, 2020.

[27] DUNE Collaboration, B. Abi, R. Acciarri, M. A. Acero, M. Adamowski, C. Adams, D. Adams, P. Adamson, M. Adinolfi, Z. Ahmad, C. H. Albright, L. Aliaga Soplin, T. Alion, S. Alonso Monsalve, M. Alrashed, C. Alt, J. Anderson, K. Anderson, C. Andreopoulos, M. P. Andrews, R. A. Andrews, A. Ankowski, J. Anthony, M. Antonello, M. Antonova, S. Antusch, A. Aranda Fernandez, A. Ariga, T. Ariga, D. Aristizabal Sierra, E. Arrieta Diaz, and J. Asaadi et al. The dune far detector interim design report volume 1: Physics, technology and strategies, 2018.

[28] J B Birks. Scintillations from organic crystals: Specific fluorescence and relative response to different radiations. *Proceedings of the Physical Society. Section A*, 64(10):874–877, oct 1951.

[29] H. Yang, D. Cao, Z. Qian, X. Zhu, C. Loh, A. Huang, R. Zhang, Y. Yang, Y. Liu, B. Xu, and M. Qi. Light attenuation length of high quality linear alkyl benzene as liquid scintillator solvent for the JUNO experiment. *Journal of Instrumentation*, 12(11):T11004–T11004, nov 2017.

[30] Thierry Lasserre and Henry W. Sobel. Reactor neutrinos. *Comptes Rendus Physique*, 6(7):749 – 757, 2005. Neutrino physics.

[31] JUNO Collaboration, Angel Abusleme, Thomas Adam, Shakeel Ahmad, Sebastiano Aiello, and Muhammad Akram et al. Tao conceptual design report: A precision measurement of the reactor antineutrino spectrum with sub-percent energy resolution, 2020.

[32] M. Askins et al. Theia: an advanced optical neutrino detector. *Eur. Phys. J. C*, 80(5):416, 2020.

[33] D. Guffanti. Prospects for theia: an advanced liquid scintillator neutrino experiment. *J. Phys. Conf. Ser.*, 1468(1):012124, 2020.

[34] M. Yeh, S. Hans, W. Beriguete, R. Rosero, L. Hu, R.L. Hahn, M.V. Diwan, D.E. Jaffe, S.H. Kettell, and L. Littenberg. A new water-based liquid scintillator and potential applications. *Nucl. Instrum. Meth. A*, 660:51–56, 2011.

[35] Bernhard W. Adams, Klaus Attenkofer, Mircea Bogdan, Karen Byrum, Andrey Elagin, Jeffrey W. Elam, Henry J. Frisch, Jean-Francois Genat, Herve Grabas, Joseph Gregar, Elaine Hahn, Mary Heintz, Zinetula Insepov, Valentin Ivanov, Sharon Jelinsky, Slade Jokely, Sun Wu Lee, Anil. U. Mane, Jason McPhate, Michael J. Minot, Pavel Murat, Kurtis Nishimura, Richard Northrop, Razib Obaid, Eric Oberla, Erik Ramberg, Anatoly Ronzhin, Oswald H. Siegmund, Gregory Sellberg, Neal T. Sullivan, Anton Tremsin, Gary Varner, Igor Veryovkin, Alexei Vostrikov, Robert G. Wagner, Dean Walters, Hsien-Hau Wang, Matthew Wetstein, Junqi Xi, Zikri Yusov, and Alexander Zinovev. A brief technical history of the large-area picosecond photodetector (lappd) collaboration, 2016.

[36] B.S. Wonsak, C.I. Hagner, D.A. Hellgartner, K. Loo, S. Lorenz, D.J. Meyhöfer, L. Oberauer, H. Rebber, W.H. Trzaska, and M. Wurm. Topological track reconstruction in unsegmented, large-volume liquid scintillator detectors. *Journal of Instrumentation*, 13(07):P07005–P07005, jul 2018.

[37] Miroslav Kubat. *An Introduction to Machine Learning*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[38] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1), 2009.

[39] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017.

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds, 2019.

[41] Zhiyuan Liu and Jie Zhou. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14:1–127, 03 2020.

[42] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.

[43] Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport, 2013.

[44] David Meyhöfer. Vertex reconstruction in large liquidscintillator detectors. Master's thesis, Hamburg University, 2016.

[45] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alch-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[46] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

[47] Huilin Qu and Loukas Gouskos. Particlenet: Jet tagging via particle clouds. *Phys. Rev. D*, 101(5):056019, 2020.

List of Abbreviations

AUC	area under the curve
CE	Cross Entropy Loss
DGL	Deep Graph Library
EC	Edge Convolution
GC	Graph Convolution
GCN	Graph convolution networks
GeomA	geometric loss function without 3D gradients
GeomB	geometric loss function with 3D gradients
geomLoss	geometric loss library
GNN	Graph neural networks
GWS	Glashow-Weinberg-Salam model
IBD	inverse β decay
IH	inverted hierarchy
JUNO	Jiangmen Underground Neutrino Observatory
LAB	linear alkyl benzene
LAPPDs	Large Area Picosecond Photodetectors
LSD	liquid scintillation detector
MMD	Maximum Mean Discrepancy
MSE	Mean Squared Error Loss
MSW	Mikheyev-Smirnov-Wolfenstein
MT	muon tracker

NH	normal hierarchy
NLL	Negative Log Likelihood
OT	Optimal Transport
pc	perfect cut
PMNS	Pontecorvo-Maki-Nakagawa-Sakata
PMTs	Photomultiplier tubes
QR	Quadratic Reconstruction
ReLU	Rectified Linear Unit
ROC	receiver operating characteristic
SM	Standard Model of Elementary Particles
SURF	Sanford Underground Research Facility
TPC	time projection chamber
wbLS	waterbased Liquid Scintillator
wCD	water Cerenkov detector

Appendix

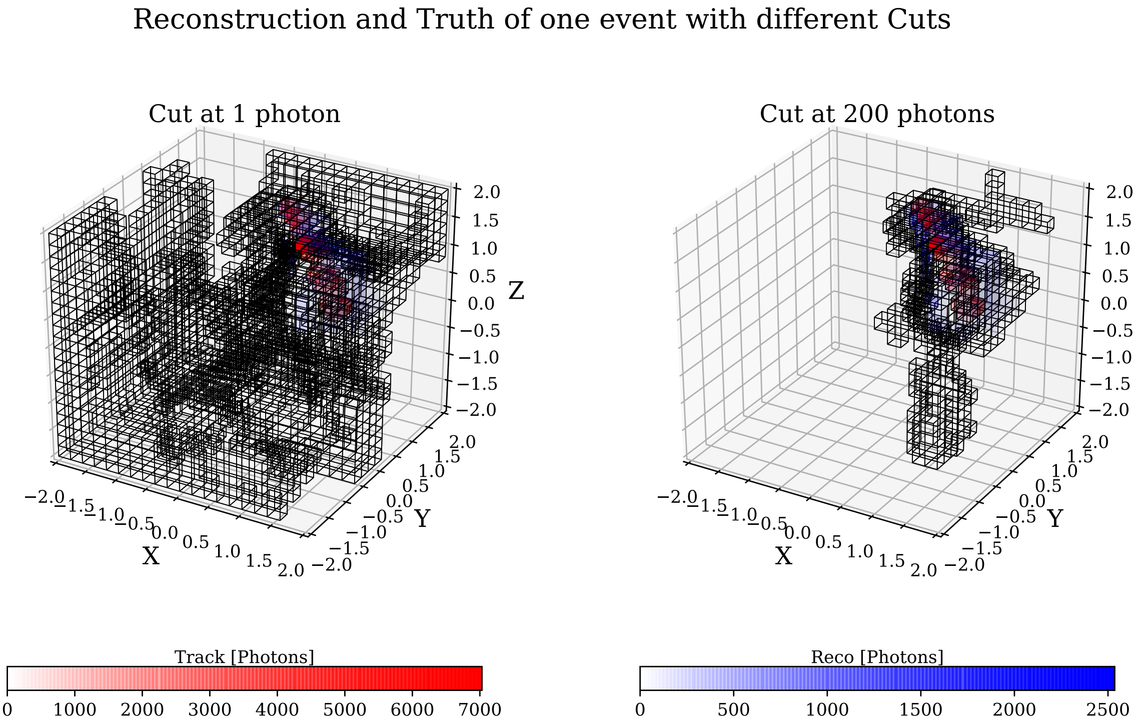


Figure 1: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

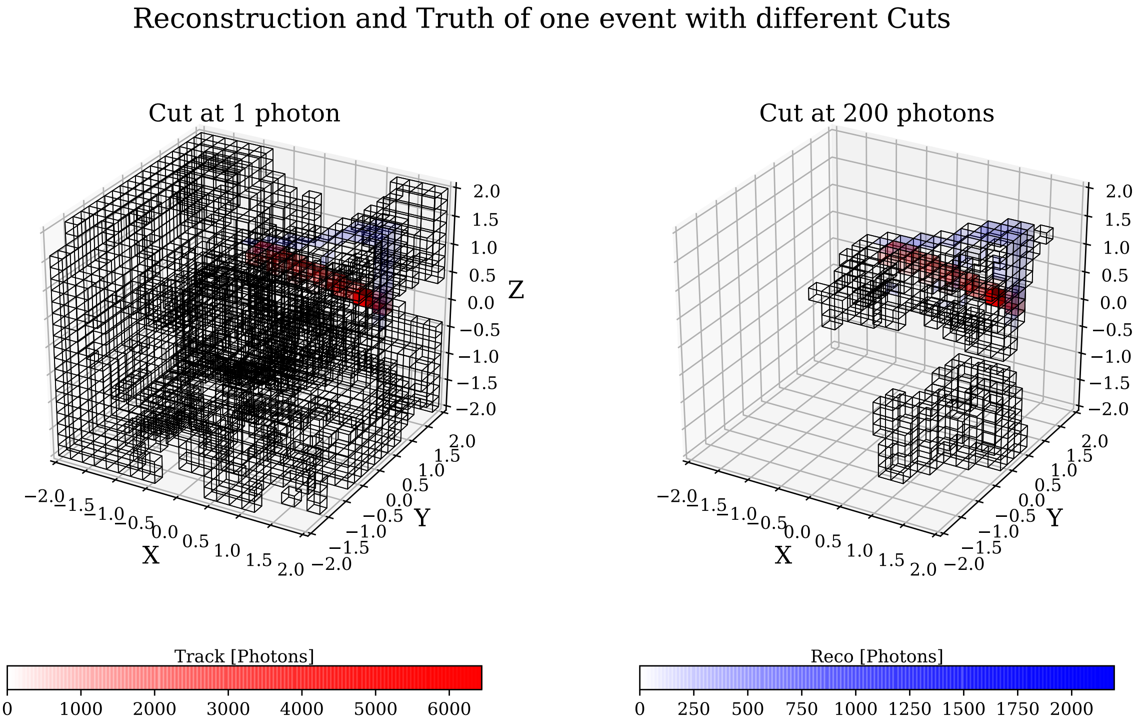


Figure 2: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

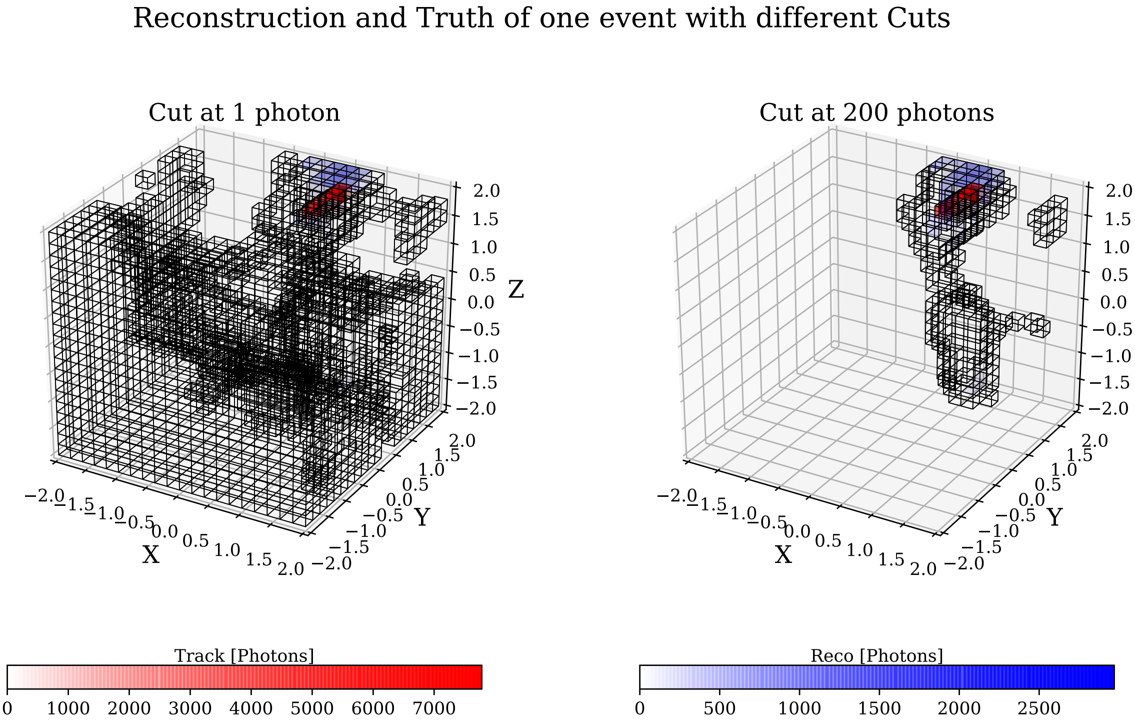


Figure 3: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

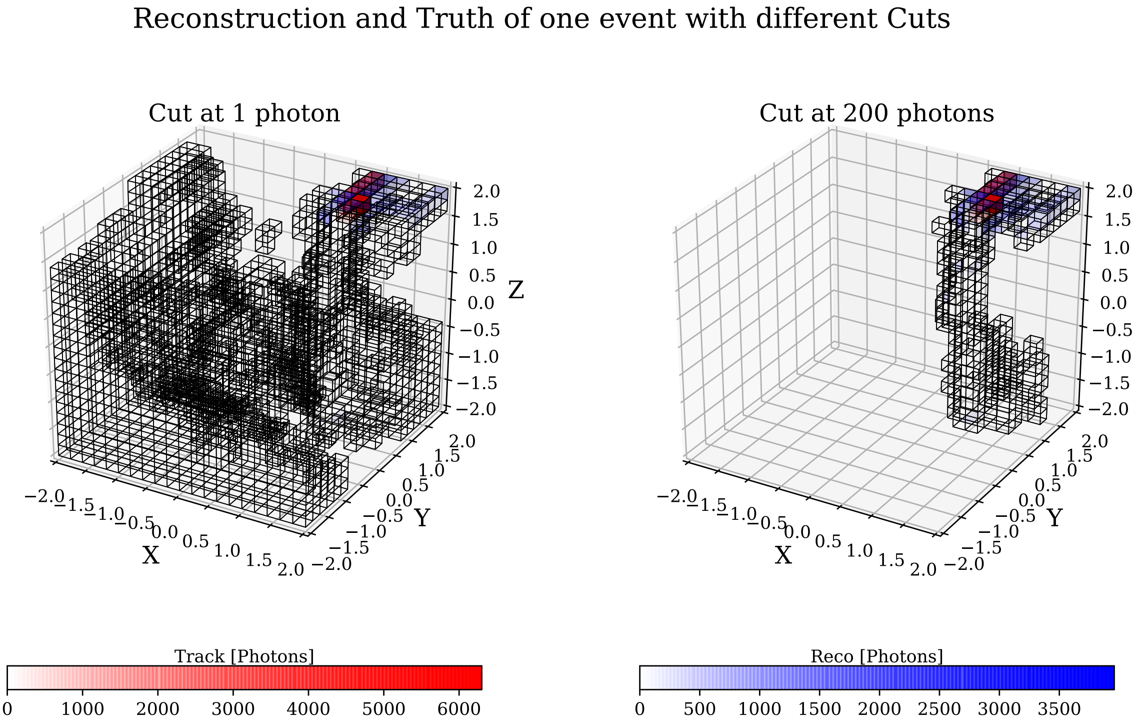


Figure 4: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

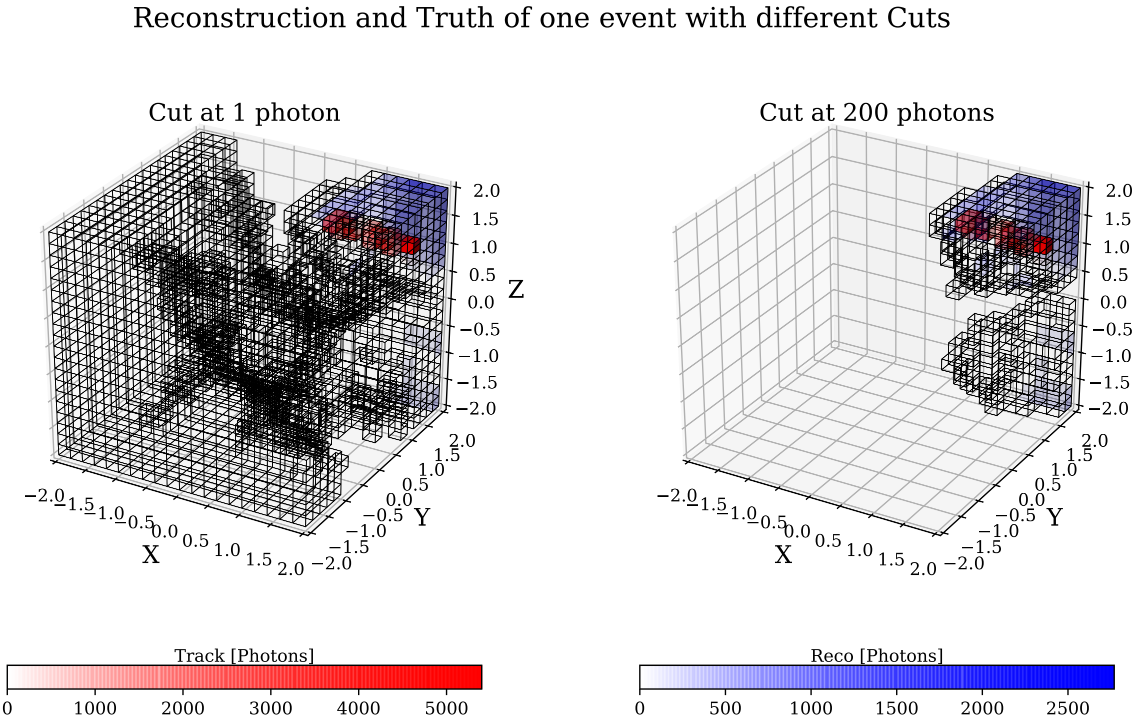


Figure 5: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

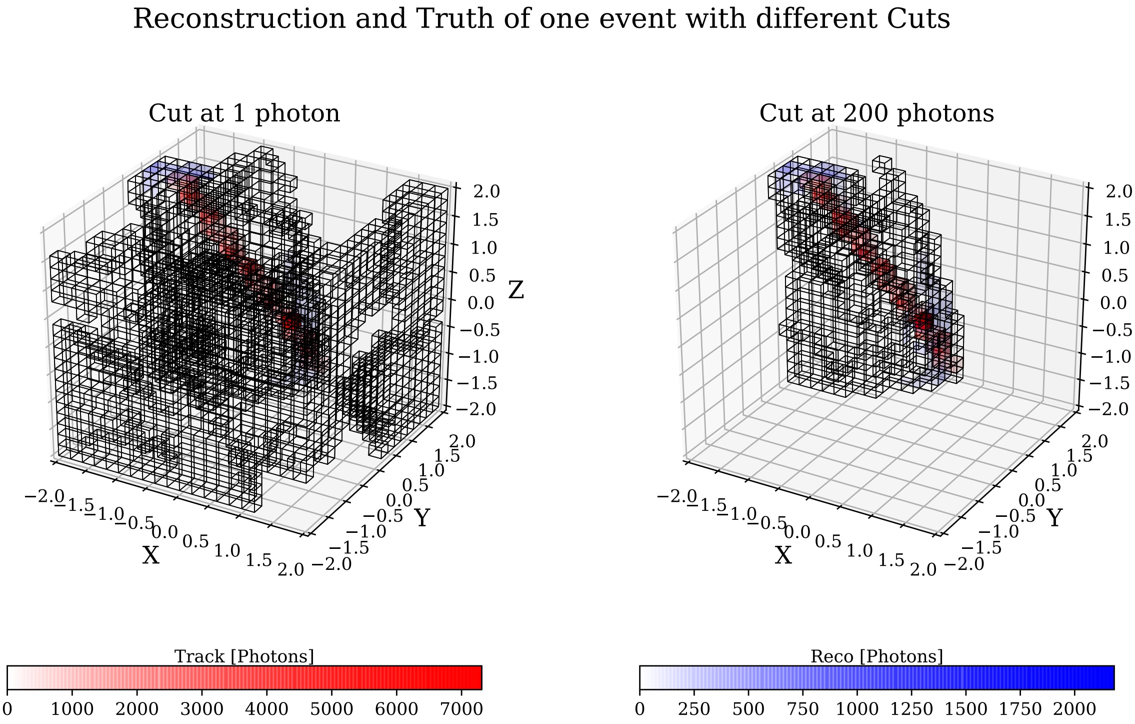


Figure 6: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

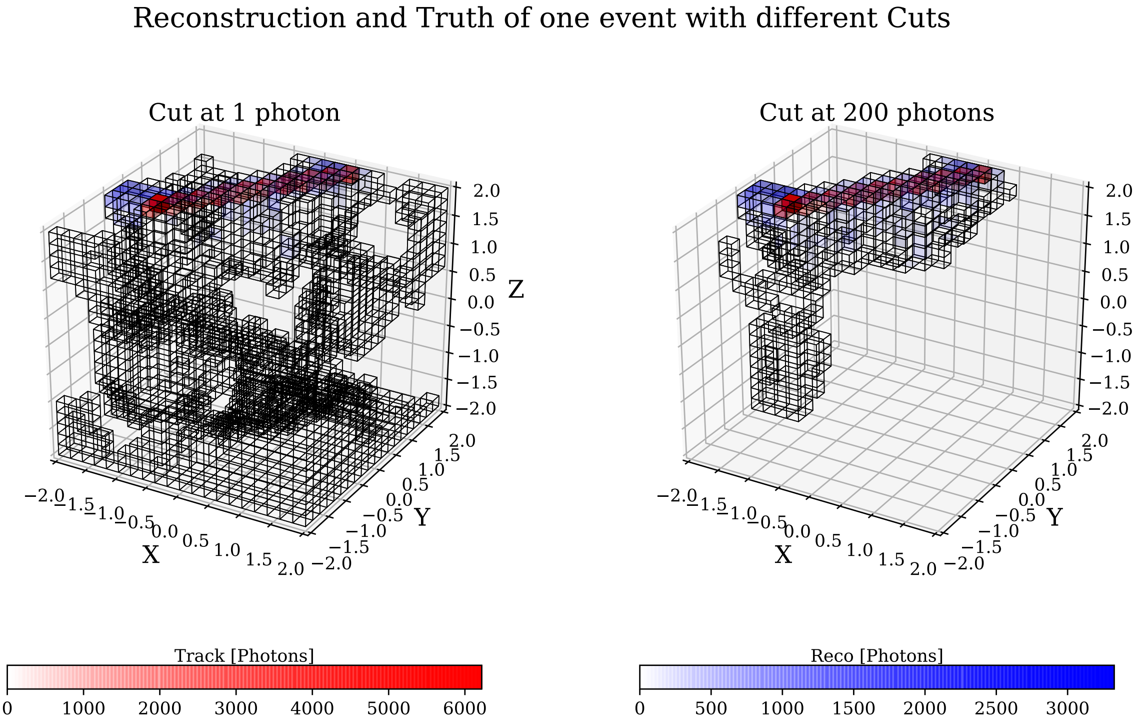


Figure 7: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

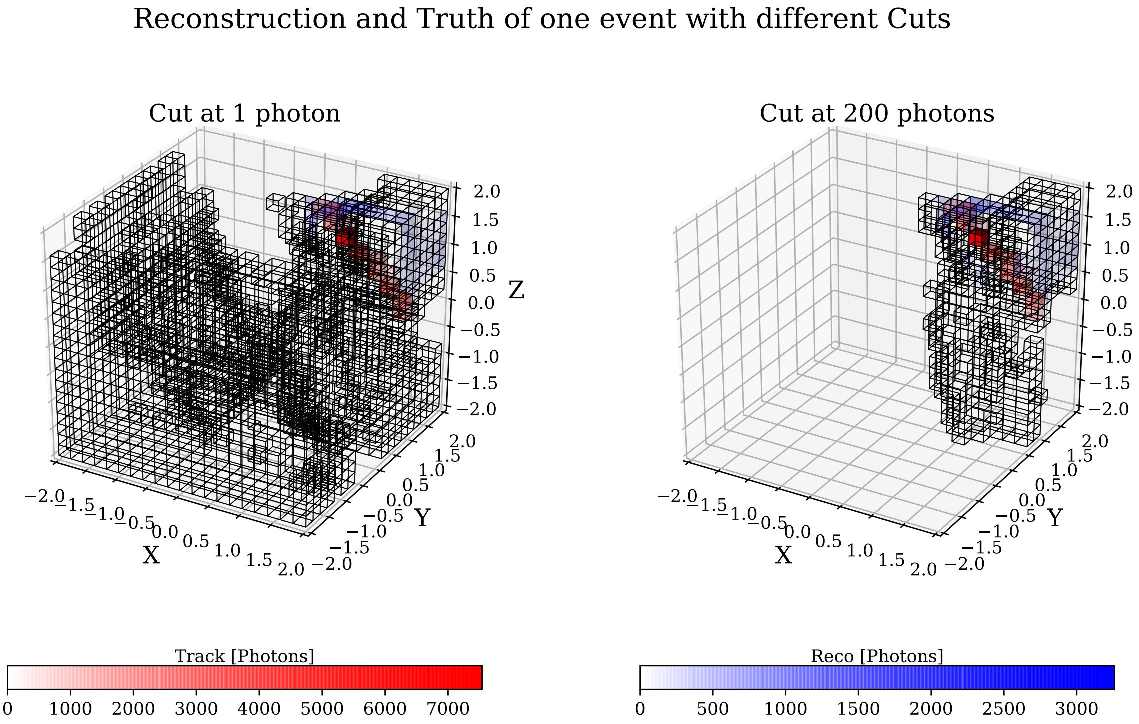


Figure 8: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

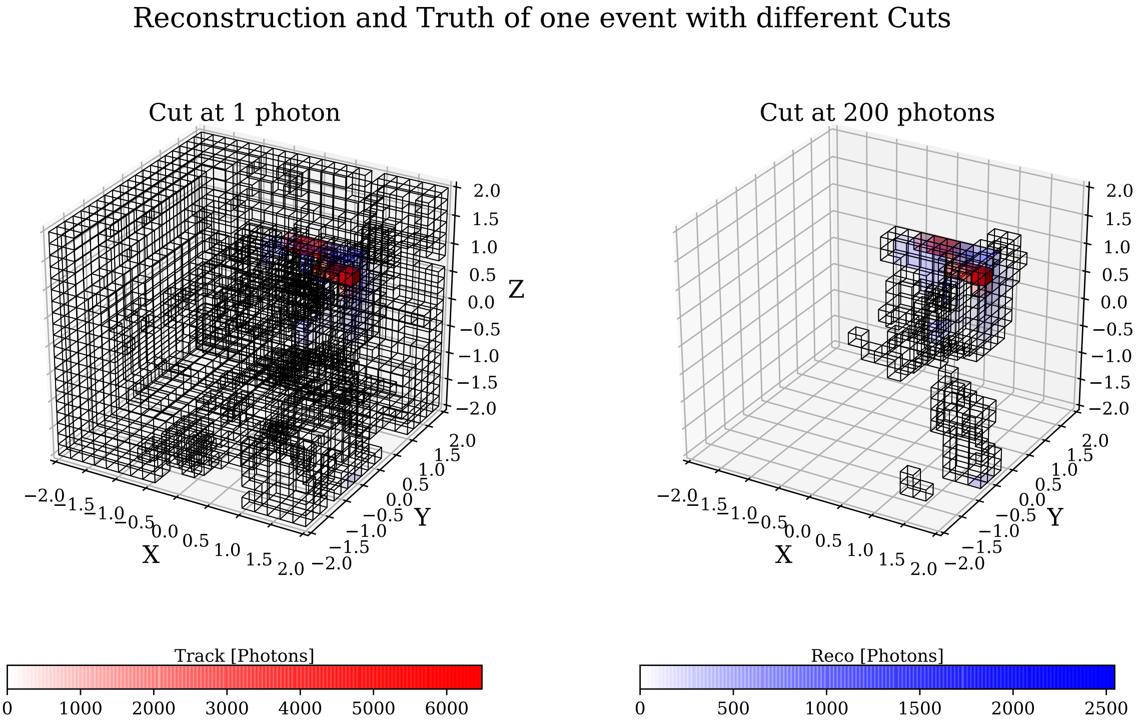


Figure 9: This plot shows the comparison of the same event with two different cuts at one and at 200 photons. The red voxels display the label and the blue ones the reconstruction.

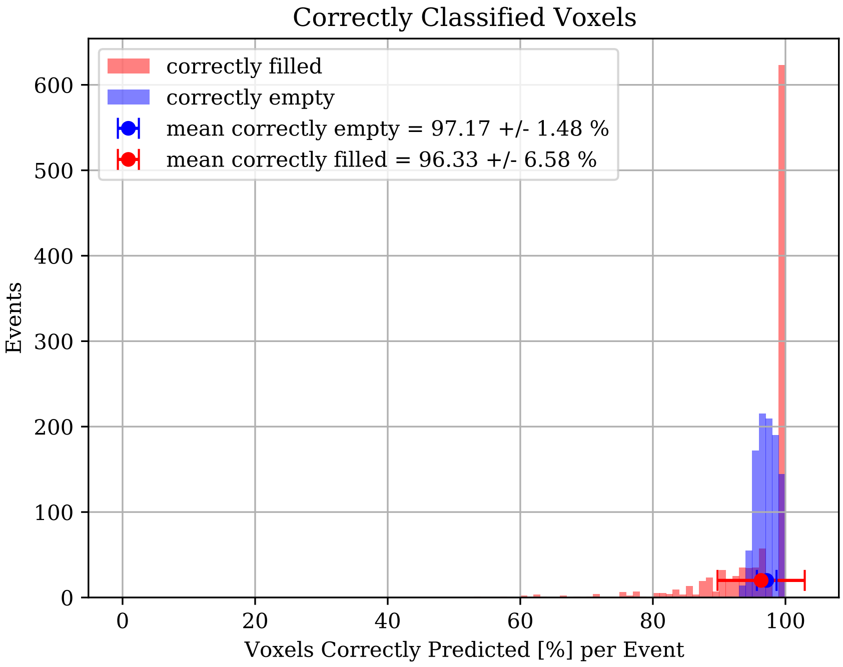


Figure 10: Percentage of correctly identified voxels of the two classes *path* and *background*. The *path* voxels are displayed in red and the *background* voxels in blue. Additional to the histogram the mean values are displayed.

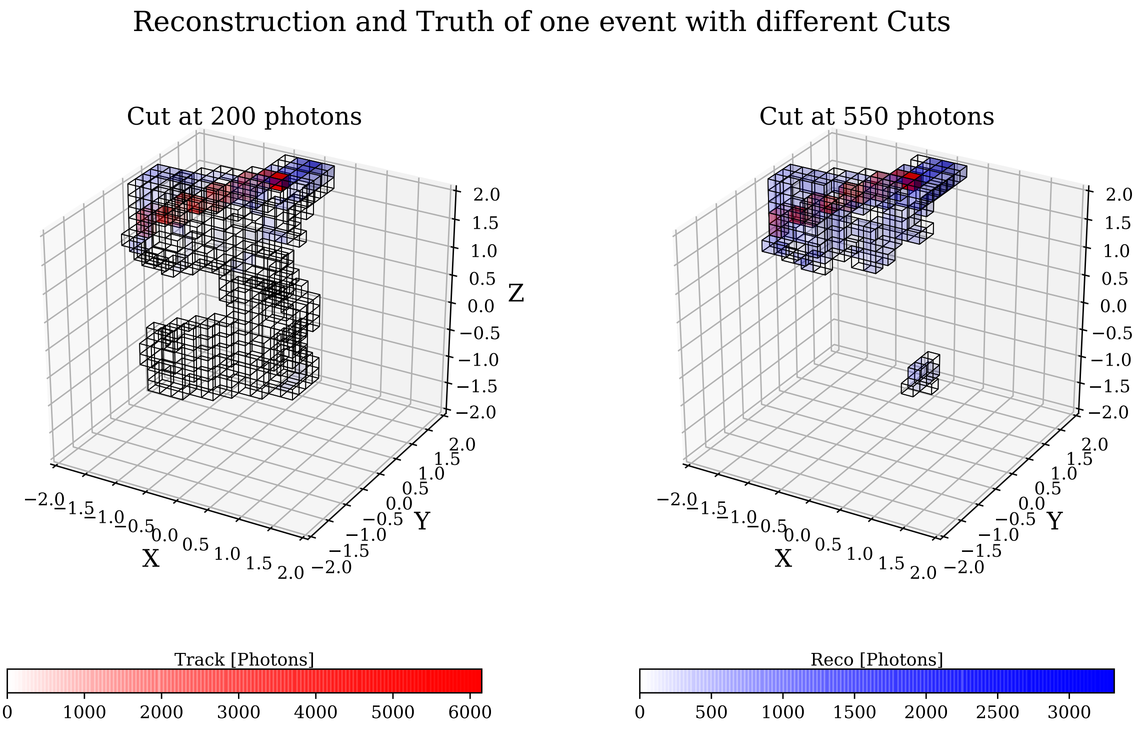
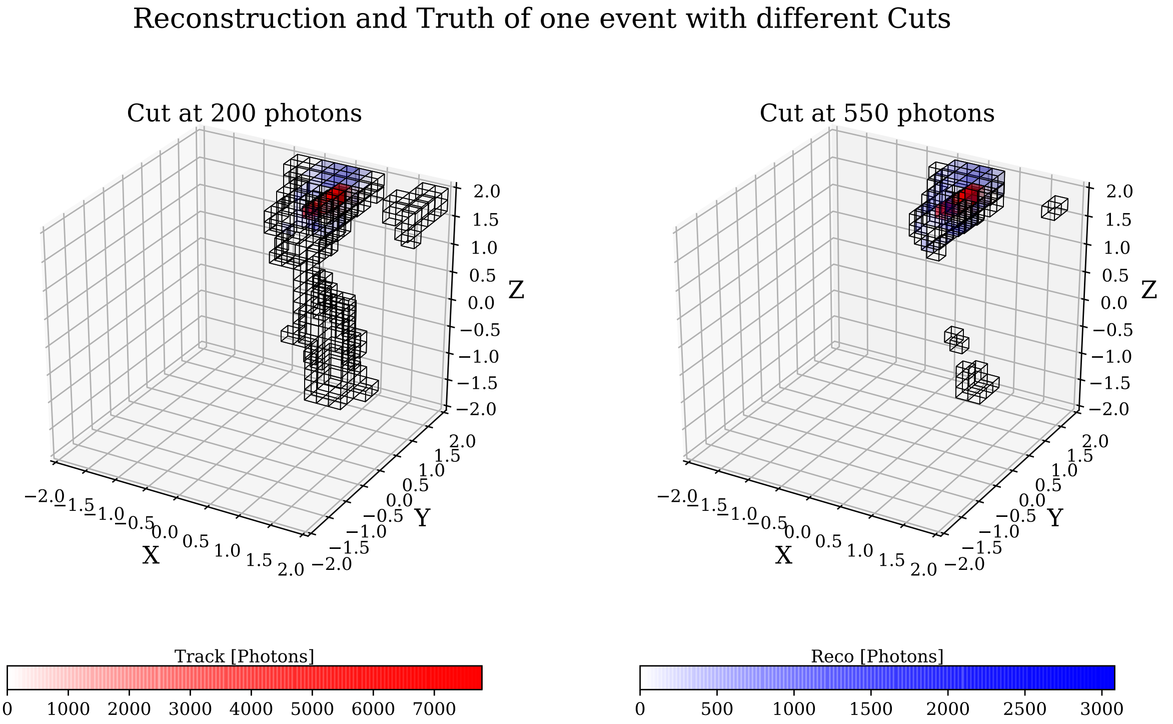


Figure 11: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

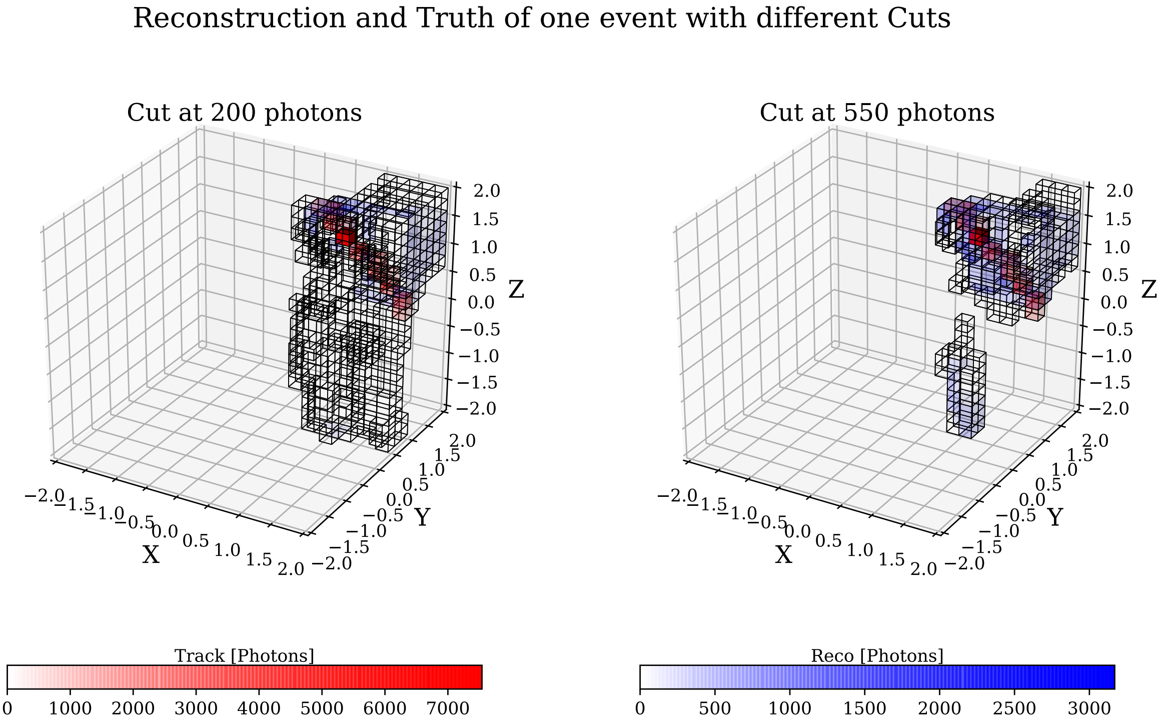


Reconstruction and Truth of one event with different Cuts

Cut at 200 photons

Cut at 550 photons

Figure 12: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.



Reconstruction and Truth of one event with different Cuts

Cut at 200 photons

Cut at 550 photons

Figure 13: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

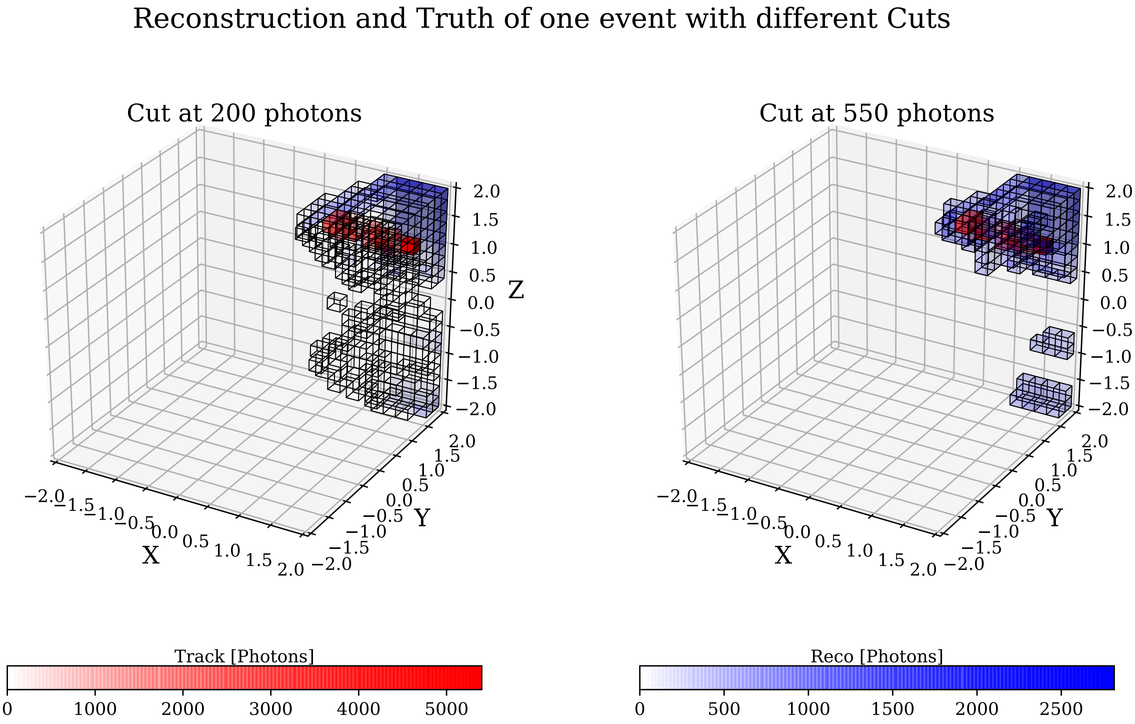


Figure 14: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

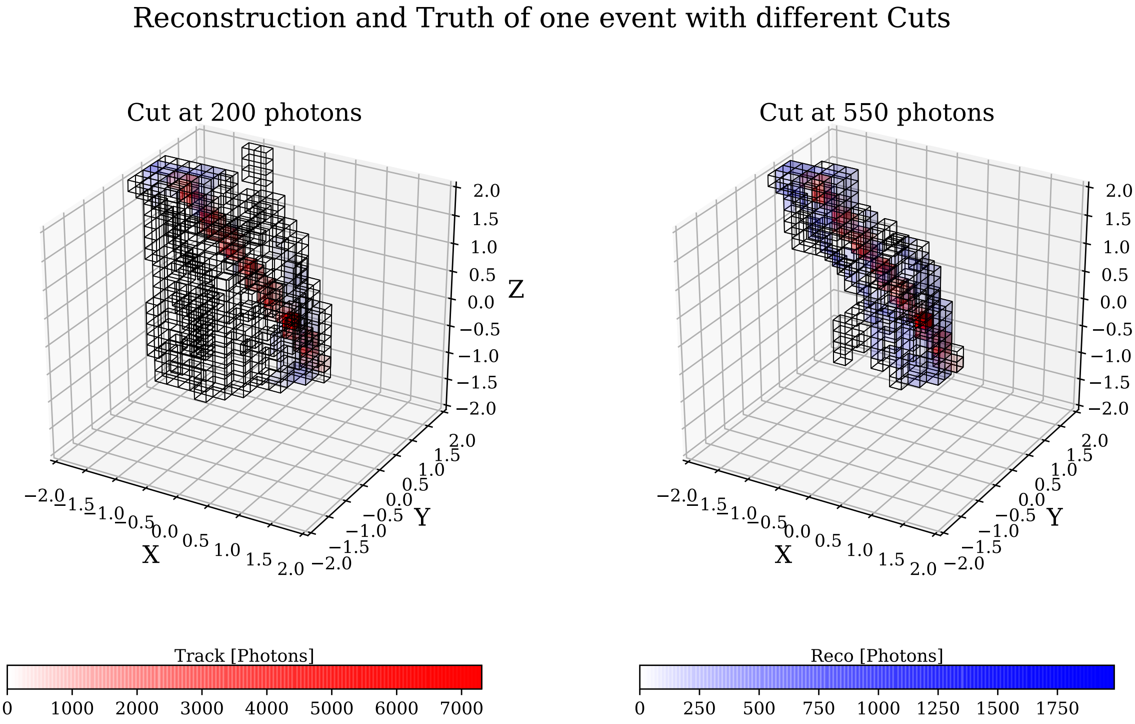


Figure 15: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

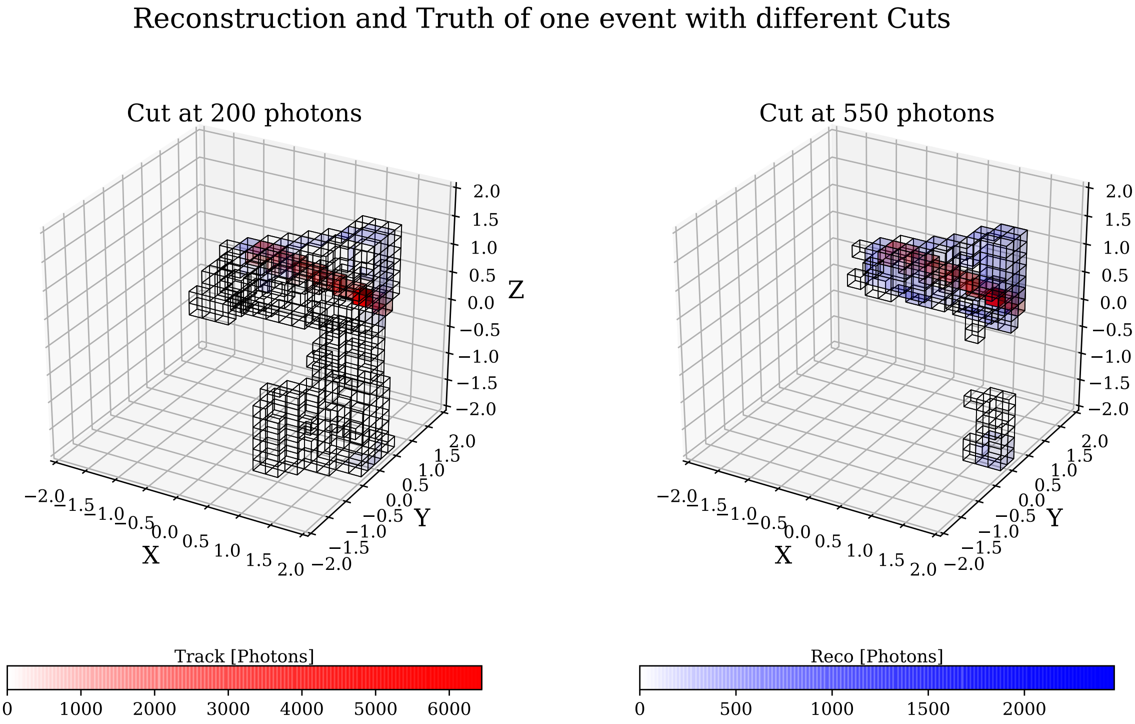


Figure 16: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

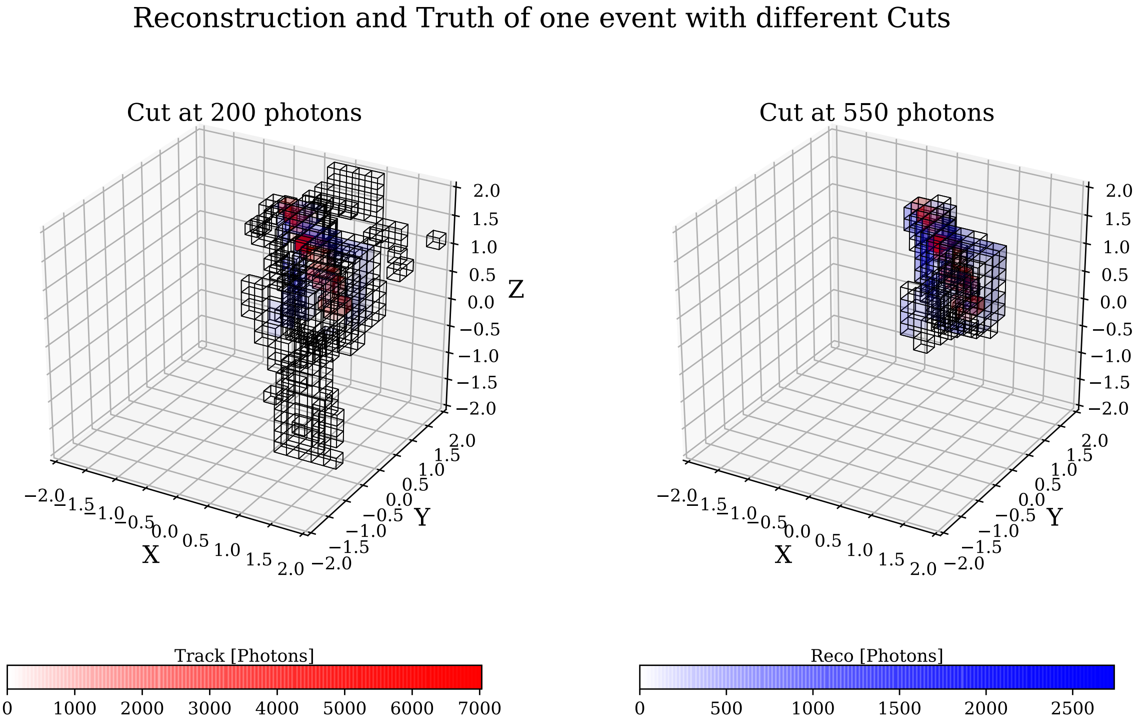


Figure 17: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

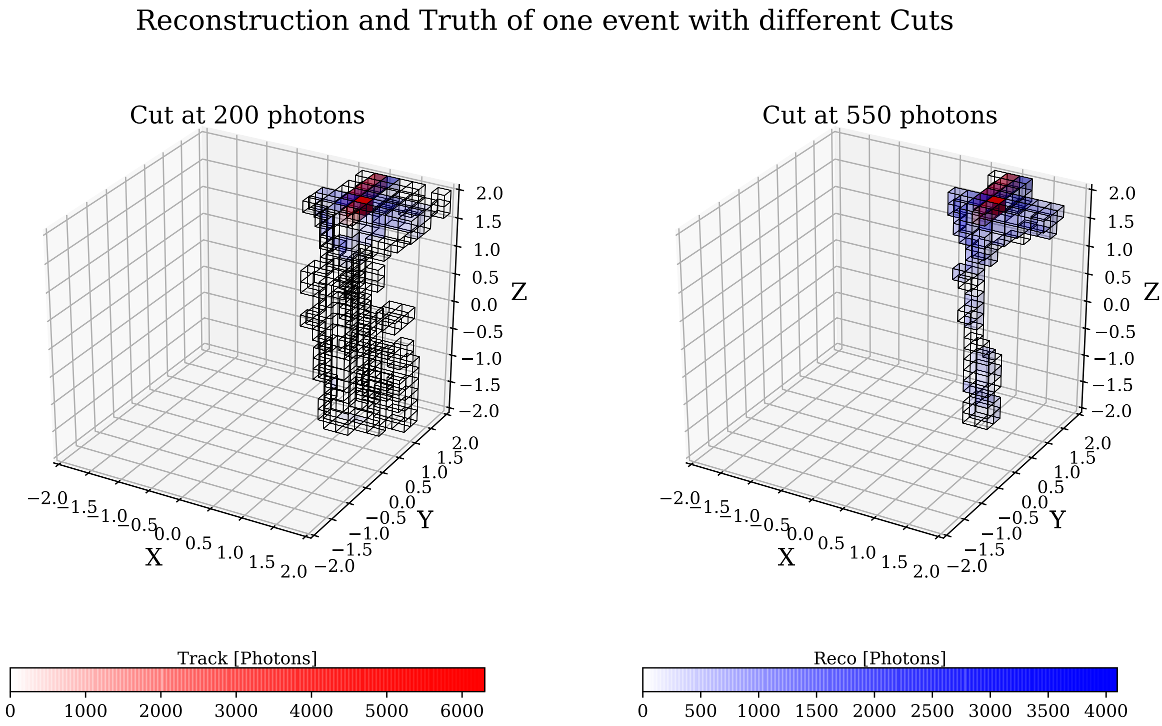


Figure 18: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

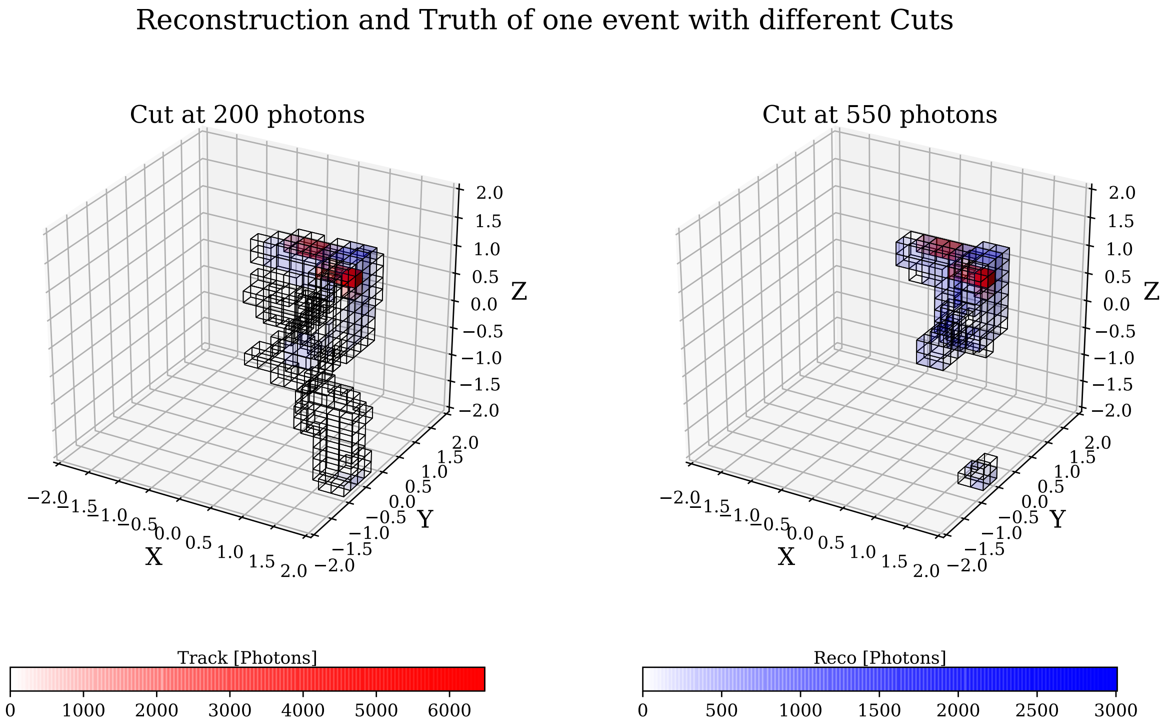


Figure 19: This plot shows the comparison of the same event with two different cuts at 200 and at 550 photons. The red voxels display the label and the blue ones the reconstruction.

Acknowledgments

Thanks to Dr. Björn Wonsak for the opportunity to write this thesis, the helpful guidance and the great support. Thanks to Prof. Dr. Gregor Kasieczka for co-supervising my work and supporting me in machine learning and GCN correlated topics.

I want to thank David Meyhöfer, who helped with technical questions. Dr. Henning Rebber and Malte Stender helped me in physical questions and a lot of reading, thank you for your time.

Thanks a lot to the *Forschungsgruppe Neutrinophysik* at the University of Hamburg, I really enjoyed my work in this group and the cake meetings, which sadly took place less frequent than expected.

Lastly, thanks to me friends and family for supporting me during this thesis and all my student life.

Erklärung

Ich versichere, dass ich die beigefügte schriftliche Masterarbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für alle Informationen, die dem Internet oder anderer elektronischer Datensammlungen entnommen wurden. Ich erkläre ferner, dass die von mir angefertigte Masterarbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung im Rahmen meines Studiums war. Die von mir eingereichte schriftliche Fassung entspricht jener auf dem elektronischen Speichermedium.

Ich bin damit einverstanden, dass die Masterarbeit veröffentlicht wird.

Ort, Datum

Rosmarie Wirth