BACHELOR THESIS

Training and application of a neural network to calculate PMT hit probabilities for incident photons in mDOMs for IceCube Upgrade

Supervisor:
Prof. Dr. Alexander KAPPES

Second examiner:
Prof. Dr. Anton ANDRONIC

A thesis submitted in fulfilment of the requirements for the degree of

Bachelor of Science

by

Luca BARTHELMEß

AG Kappes
Institute of Nuclear Physics
University of Münster

13th September 2025

Contents

1 Introduction

Questions such as what lies beyond Earth, how the universe works, and where it originates have occupied astronomers for many centuries. Over time, the state of research in astronomy and astrophysics has continued to evolve. The first significant advance in observable astronomy was the invention of the optical telescope, which enabled more precise observation of stars and celestial bodies than with the naked eye. Although optical telescopes were constantly being improved, the observation range was limited. Electromagnetic radiation (e.g. infrared, ultraviolet, radio, X-ray, gamma ray radiation) revolutionized research as it makes the detection of invisible astronomical phenomena possible. However, electromagnetic radiation is easily absorbed, and the universe becomes opaque to the higher energies over large distances, limiting our field of view. Nowadays, astronomers observe not only different forms of electromagnetic radiation, but also cosmic rays, gravitational waves, and *neutrinos*, allowing us to explore parts of the universe that are otherwise hidden from view. In contrast to electromagnetic radiation and cosmic rays, neutrinos are not absorbed or scattered by matter or deflected by electromagnetic fields, since they are uncharged particles, which mainly interact with matter via weak interaction. Due to these properties, neutrinos can provide useful information about their origin. However, as a consequence, their detection is challenging. In order to detect high-energy astrophysical neutrinos, a large detector volume is required, as the *IceCube Neutrino Observatory* [1], which consists of 1 km^3 antarctic ice instrumented with photon detectors called digital optical modules (DOMs). These modules equipped with photomultiplier-tubes (PMTs) can detect Cherenkov radiation which is emitted by secondary charged particles produced through the interaction of neutrinos with the ice. In the near future, a new extension called *IceCube Upgrade* [2], will be deployed with new optical modules, among which is the multi-PMT digital optical module (mDOM) [3] that will be considered in this thesis. The complex geometry and optical properties of the mDOM, compared to the standard DOM, require the development of new algorithms—for example, for assigning photons that reach the vicinity of an mDOM to the corresponding PMT in IceCube simulations—since the current IceCube simulation [4] cannot capture this complexity. As demonstrated in [5], for the case of the IceCube-Gen2DC-16 optical module [6], neural networks represent a strong candidate for this task, given their fast inference times and ability to excel in high-dimensional problems. The approach is to train a neural network that incorporates the inherent symmetries of the optical module on very detailed Geant4 simulations, in order to predict the detection probability on any of the PMTs of the optical module for a given incoming photon. This thesis takes that work as a starting point and expands it to implement the mDOM. For this purpose, the structure of this work is as follows: First a brief introduction into IceCube, with emphasis on IceCube Upgrade, and the corresponding neutrino physics will be provided in chapter 2. After that, the basics of neural networks will be discussed in chapter 3, which are necessary for the description of the neural network presented later in chapter 4. To train and evaluate the neural network, the Geant4 simulation of the mDOM [7] described in chapter 5 will be used. Finally, the performance of the neural network is presented and discussed in chapter 6, which includes the accuracy for an mDOM with and without harness and the inference time improved by several techniques.

2 Neutrino astronomy & IceCube

The purpose of neutrino astronomy is to search for nearly massless subatomic particles called neutrinos, which provide useful information on astrophysical sources and phenomena [8]. These particles can be detected with large water- or ice-based Cherenkov detectors, such as KM3NeT [9], Super-Kamiokande [10] or *IceCube* [1], which is the discussed detector of this thesis. In order to understand the detection mechanism of the *IceCube Neutrino Observatory*, it is necessary to get acquainted with the theory of neutrino physics. After a brief introduction into neutrino properties, the detection and potential sources of high-energy neutrinos, an outline of IceCube will be given. Special emphasis is placed on its extension *IceCube Upgrade* [2], which deploys mDOMs analyzed in this thesis.

2.1 Neutrino properties and interaction with matter

Neutrinos are elementary particles in the Standard Model of particle physics where they are part of the lepton family. They can appear in three flavors, being the electron neutrino ν_e, muon neutrino ν_μ and the tauon neutrino ν_τ, where each neutrino flavor also has a corresponding antiparticle. After propagation through matter or vacuum their flavor can change periodically, a phenomenon known as *neutrino oscillation* [11], which proves that neutrinos are not massless. However, this mass is almost negligible, with a reported 90% upper limit value of $0.45\ \mathrm{eV/c^2}$ [12] as measured by the KATRIN experiment[1] [13]. Neutrinos are also uncharged particles. As a result, they do not undergo electromagnetic interactions, while gravitational interactions are negligible. The most significant interaction of neutrinos is the *weak interaction*, which manifests, for example, in β-decays. In fact, neutrinos were first postulated by Wolfgang Pauli in 1930 to explain the continuous energy spectrum observed in these decays [14].

In the context of the IceCube experiment, typical neutrino energies are in the order of 10 GeV to 10 PeV [15]. At energies greater than a few GeV [16] neutrinos interact primarily through *deep-inelastic scattering* with nucleons, where the scattering process can follow two different mechanisms of weak interaction, the *charged current* and the *neutral current* [17]. The charged current is mediated by the exchange of a W^+ or a W^- boson, whereas the neutral current is mediated by the exchange of a Z^0 boson. The two different currents can be described as follows, where $l = e, \mu, \tau$ denotes the corresponding flavor, $N = p, n$ the nucleon and X a hadronic shower:

$$\begin{aligned} \text{charged current:} \quad & \nu_l + N \xrightarrow{\mathrm{W^{\pm}}} l + X, \\ \text{neutral current:} \quad & \nu_l + N \xrightarrow{\mathrm{Z^0}} \nu_l + X. \end{aligned} \tag{2.1.1}$$

According to eq. (2.1.1), deep-inelastic scattering of neutrinos results in a generation of hadronic shower consisting of charged particles and a production of a corresponding lepton or neutrino, depending on the current. Furthermore, the above equations are also valid for antiparticles $\bar{\nu}_l$ and $\bar{l}$.

[1] **KA**rlsruhe **TRI**tium **N**eutrino experiment

2.2 Neutrino detection through Cherenkov radiation

As mentioned in section 2.1, neutrinos interact mainly through the weak interaction. Thus, neutrinos cannot be detected directly. Nevertheless, neutrinos can produce charged secondary leptons according to eq. (2.1.1) during their interaction with matter. These particles can, in turn, emit photons via the *Cherenkov effect* [19], which can be detected by optical modules consisting of photomultiplier-tubes (PMTs). When charged particles with velocity v_p travel through a dielectric medium, for example ice in case of IceCube, the particles can induce dipoles in the medium along its trajectory. As the dipoles relax, they emit spherical electromagnetic waves propagating through the medium with a velocity $c' = c/n$, where c is the speed of light in vacuum and n the refractive index of the medium. If the speed of the particle is lower than the phase velocity of the medium ($v_p < c'$), the electromagnetic waves interfere destructively. However, if the particle is faster than the velocity of light in the medium, adjacent spherical waves can interfere constructively forming a conical wavefront with velocity $v_w = c/n$, which is illustrated in fig. 2.2.1. This effect is called *Cherenkov effect* and was experimentally verified by Pavel Cherenkov in 1934 [19].

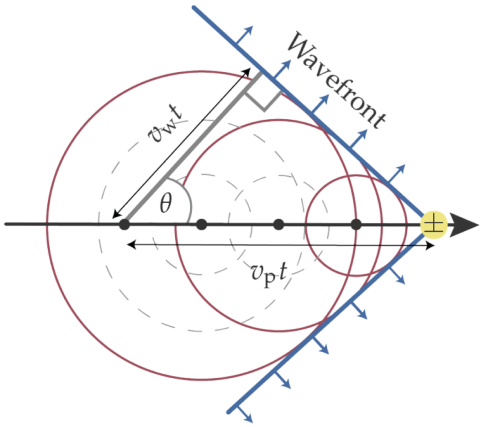


FIGURE 2.2.1: Schematic illustration of the Cherenkov effect. Spherical electromagnetic waves (red) are produced and interfere constructively, forming a wavefront (blue) at the Cherenkov angle θ with respect to the trajectory of the charged particle (black arrow), whose velocity v_p exceeds the phase velocity of light in the medium v_w. Taken from [18].

The shape of the conical wavefront can be characterized by the opening angle θ with respect to the particle trajectory. The opening angle shown in fig. 2.2.1 can be calculated by the following formula, where β is equal to v_p/c:

$$\cos(\theta) = \frac{v_w t}{v_p t} = \frac{1}{\beta n}. \tag{2.2.1}$$

2.3 Neutrino sources

Neutrinos detected on Earth originate from a wide variety of sources [20]. Typical energies of neutrinos observed by IceCube and its future extensions (see section 2.5) range from a few GeV to EeV. This implies that IceCube is mostly sensitive to *atmospheric* and *high-energy astrophysical neutrinos*. The lower energies are dominated by atmospheric neutrinos, which are mainly generated by pion decays resulting from cosmic ray interactions in the Earth's atmosphere. In contrast, the flux of high-energy astrophysical neutrinos decreases less steeply with energy than the atmospheric component and becomes dominant from about 10 to 100 TeV onward. These neutrinos are thought to originate from the same sources that accelerate cosmic rays, and their identification is therefore of great importance. As of the time of writing this thesis, the three sources with the highest significance identified by IceCube are the blazar *TXS 0506+056* [21], the active galaxy *NGC 1068* [22], and the Milky Way galaxy [23].

2.4 The IceCube Neutrino Observatory

The AMANDA[2] neutrino telescope [24], the predecessor of IceCube, built at the geographical South Pole in the mid 1990s, demonstrated that the clear Antarctic ice was suitable as a detection medium for high-energy neutrinos [1]. Based on its results, IceCube was designed and constructed at the Amundsen-Scott South Pole Station where it was completed in 2011 [25] as the first cubic-kilometer-scale neutrino detector worldwide. The primary purpose has been the discovery of astrophysical neutrinos and their sources [25]. Furthermore, it contributes e.g. to the detection of dark matter, searches for exotic particles, studies of neutrino oscillations and also plays a key role in multi-messenger astronomy, collaborating with optical, X-ray, gamma-ray, radio, and gravitational wave observatories to provide a more comprehensive view of astrophysical objects.

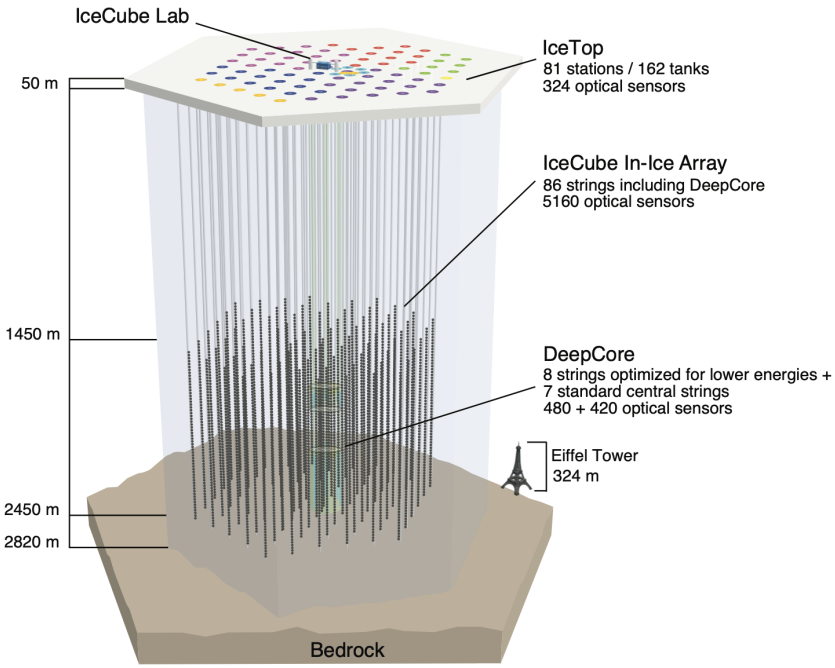


FIGURE 2.4.1: The IceCube Neutrino Observatory with its in-ice array, DeepCore sub-array and IceTop air shower array. The colors visualize different deployment seasons of according stations or strings. Taken from [25].

As illustrated in fig. 2.4.1, the IceCube detector consists of four parts, described in the following [25]. The first part is the *in-ice array*. It consists of 86 strings, each equipped with 60 Digital Optical Modules (DOMs), each containing a downward-facing 10-inch diameter PMT housed in a glass vessel [26]. 78 of the 86 strings form the primary in-ice array, which is deployed in a hexagonal grid with an average horizontal spacing of 125 m and is optimized to detect neutrinos in the energy range $\mathcal{O}(\mathrm{TeV})$–$\mathcal{O}(\mathrm{PeV})$. *DeepCore* [27] is a denser sub-array with an average string spacing of 72 m (down to 40 m in the core region). The depths between 2000 m and 2100 m are not instrumented, as optical scattering and absorption significantly affect the detection of Cherenkov radiation due to dust in the ice [28]. In total, DeepCore consists of 8 specialized strings in addition to 7 strings from the main array and is optimized for neutrino detection in the energy range $\mathcal{O}(10\,\mathrm{GeV})$–$\mathcal{O}(100\,\mathrm{GeV})$, improving sensitivity, for example, to atmospheric neutrino oscillations.
The third component is *IceTop* [29], a surface array consisting of 162 tanks filled with ice, each instrumented with two IceCube DOMs. IceTop can measure primary cosmic rays in the

[2] **A**ntarctic **M**uon **A**nd **N**eutrino **D**etector **A**rray

energy range from 100 TeV to 1 EeV [15] and can also serve as a partial veto for atmospheric muons and coincident atmospheric neutrinos in astrophysical neutrino searches in the southern sky. Finally, the *IceCube Laboratory*, also located on the surface, is the central operation building and responsible for data acquisition.

2.5 IceCube Upgrade

The *IceCube Upgrade* [2] is a planned extension of IceCube that will be deployed during the austral summer of 2025/2026 [30]. This new extension is optimized for low energies in the context of IceCube and will reduce the energy threshold to about 1 GeV. This will provide, for example, world-leading sensitivity to atmospheric neutrino oscillations [2]. Moreover, the IceCube Upgrade will allow a more precise measurement of the ice properties by means of new calibration devices.

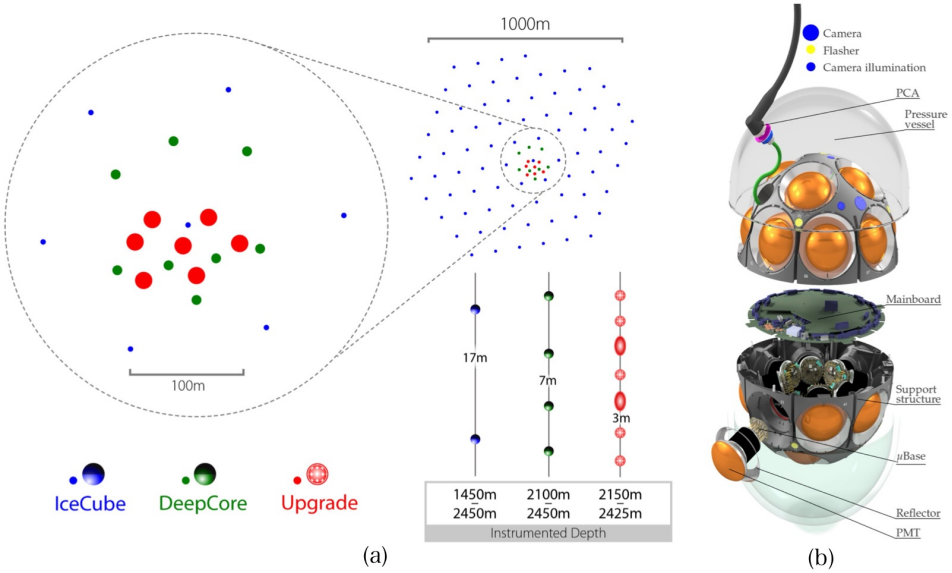


FIGURE 2.5.1: **(a)**: Schematic illustration of *IceCube Upgrade*. Taken from [2]. **(b)**: Exploded view of an mDOM with 24 PMTs embedded in a support structure and encapsulated by a pressure vessel. Taken from [18].

As illustrated in fig. 2.5.1 (a), IceCube Upgrade will consist of seven new strings with nearly 700 optical modules in total, embedded inside the already existing DeepCore. They will be placed at depths between 2150 m and 2425 m, referred to as the *physics region*, in order to take advantage of the high clarity of the glacial ice and the low atmospheric muon background. Furthermore, the spacing between two modules measures approximately 20 m in horizontal and 3 m in vertical direction, which results in a higher sensitivity to lower energetic events. In contrast to the original IceCube array, the extension incorporates mainly two new optical modules with improved photon detection efficiencies and calibration capabilities, being the **mDOM** (**m**ulti-PMT **D**igital **O**ptical **M**odule) [3] shown in fig. 2.5.2 and the D-Egg [32]. The mDOM consists of 24 PMTs with 80 mm diameter, whose orientations cover the entire solid angle and are fixed in a 3D printed support structure, where surrounding reflectors around the PMTs increase the sensitivity. This design depicted in fig. 2.5.1 (b)

FIGURE 2.5.2: mDOM with harness consisting of 24 PMTs. Taken from [31].

provides an almost homogeneous angular coverage and an effective photosensitive area that is more than twice that of a DOM [3]. Moreover, a pressure vessel secures and encloses the various components. By using a silicon-based gel layer between vessel, PMTs and support structure, the module gains increased mechanical stability and improved coupling between the glass pressure vessel and the PMTs. Deploying the modules on the strings requires a harness, which is shown in fig. 2.5.2.

3 Theory of neural networks (NN)

This chapter provides the fundamental knowledge of deep learning, neural networks, and their training, as well as quantization techniques, in order to understand the subsequent chapters where these concepts are applied to the neural networks studied in this thesis.

3.1 Machine learning and deep learning

Classical programming follows the principle that data are processed according to pre-defined rules in programs. However, sometimes it is difficult or even impossible to figure out these rules for complex problems, making it necessary to find new approaches. *Machine learning* [33] has led to a paradigm shift in which instead of data and *rules*, data and *answers* are used to learn mappings between so-called input data (features) and output data (labels/answers). These rules can then be applied to new data in order to predict the corresponding answers. In this approach, the learning process consists of automatically finding suitable transformations that turn the input data into more useful *representations* which can lead to an as high percentage of right answers as possible.
A specific subfield of machine learning is *deep learning* [33], in which the model learns successive *layers* of increasingly meaningful representations. In this context, the term "deep" comes from the fact that modern deep learning can contain tens or even hundreds of layers of representations (deep hierarchy). These layered representations are learned via models called *neural networks* which are structured as a series of layers each consisting of neurons. The next two sections explain two common examples of deep learning models, being the *Multilayer Perceptron* [33, 34] and the *Convolutional Neural Network* [33].

3.2 Multilayer Perceptrons (MLP)

Multilayer Perceptrons are feedforward neural networks and consist of an input layer, one or more hidden layers and an output layer each composed of *neurons* (also known as units). This structure is illustrated in fig. 3.2.1, where units are symbolized by circles. Each hidden layer and the output layer use linear tensor operations combined with a nonlinear activation function to capture non-linear features in the data. The output $\mathbf{y}$ of each layer will be calculated as follows, where W describes the weight matrix, $\mathbf{x}$ the input data of a *sample*[3], $\mathbf{b}$ the bias vector, and a the non-linear activation function of a specific layer:

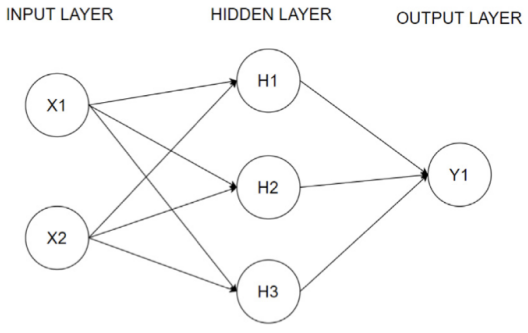


FIGURE 3.2.1: Schematic representation of a fully connected MLP consisting of three layers with a different number of neurons, represented by circles. Taken from [34].

[3] e.g. an image (sample) with several bits (features)

$$\mathbf{y} = a(W \cdot \mathbf{x} + \mathbf{b}). \tag{3.2.1}$$

The output vector $\mathbf{y}$ of a layer has a length equal to the number of units N_{units}. Consequently, the weight matrix W has dimensions $(N_{\text{neurons}} \times \text{length}(\mathbf{x}))$, and each unit has a corresponding bias term. The output of a layer is used as the input of the next layer, so that each layer transforms the input data from the network to a more abstract representation. Layers with such characteristics are referred to as *Dense Layers*, which means that each unit on a layer takes as input the output of every unit in the previous layer. This is symbolized in fig. 3.2.1 by arrows that connect neurons of different layers.

3.3 Convolutional Neural Network (CNN)

While multilayer perceptrons learn global patterns in their input data, convolutional neural networks [33] are able to learn local and translation-invariant patterns in so-called *feature maps*, such as images. This means that *convolutional layers* can recognize patterns anywhere in a feature map, whereas dense layers have to learn patterns repeatedly, if they change their location. As a result, convolutional layers are data-efficient, because they need less training data to learn the same pattern, regardless of its location in the feature map. Moreover, convolutional layers can learn spatial hierarchies of patterns. A well-known example for this characteristic is the identification of objects, like a cat, in a picture, in which the first convolutional layer recognizes small edges, whereas subsequent layers combine these patterns to more complex ones, such as an ear or an eye of a cat. Thus, they are widely used in image recognition.

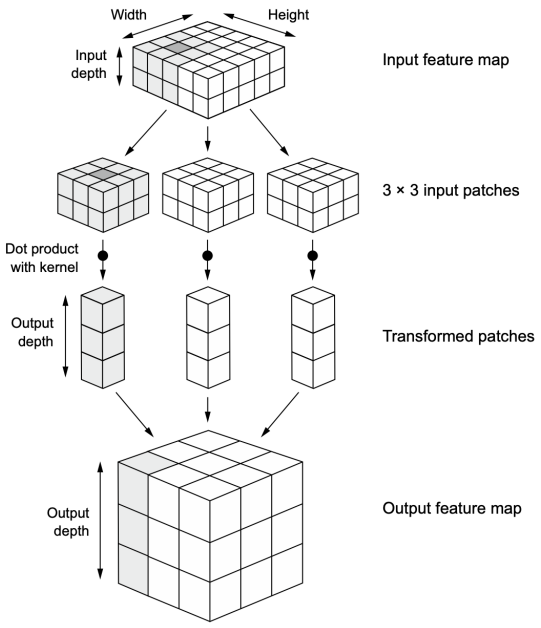


FIGURE 3.3.1: Mechanism of a convolutional layer in case of a 3D feature map with 3 kernels. Taken from [33].

The extraction of these patterns is done by means of different filters called *kernels*, which contain weights. In case of 3-dimensional input data, such as images ($height \times$ *width* $\times$ *input depth*) the mechanism of one convolutional layer is depicted in fig. 3.3.1 for three different kernels of size 3×3 sliding over a 3D tensor. Every time, when the kernel is shifted one step forward in a certain direction (height, width), the convolutional operation extracts ($3 \times 3 \times$ *input depth*) patches from the input feature map. In case of fig. 3.3.1, this results in nine patches. Subsequently, each filter is applied to each of these patches by dot products. Hence, all $3 \times 3 \times$ *input depth* patches are mapped to $1 \times 1 \times 3$ patches, where the output depth is equal to the number of kernels. In the end, the transformed patches are compressed to a new feature map of size $3 \times 3 \times 3$, which builds the output of the convolutional layer.

In CNNs, the final tensor feature map is usually flattened and fed as input to an MLP, which produces the target output.

3.4 Training of a Neural Network

At the beginning, neural network parameters are initialized randomly. If a *batch* of input samples is passed through a network with these initialized configurations, the corresponding representations are meaningless. To approximate the target values, these parameters must be tuned through an iterative process known as *training* [33]. A schematic training structure is illustrated in fig. 3.4.1, where the neural network consists of an input layer, a hidden layer and an output layer. In this thesis, the neural network training is carried out using the *mini-batch stochastic gradient descent algorithm* (SGD) [35]. First, a mini-batch of n random samples is drawn from the training dataset, which consists of pairs of input features X and corresponding true targets Y. The propagation of the input features through the network is calculated for this batch (also known as *forward pass*), and as a result, the predicted targets Y' can be obtained. After that, the difference between the predictions Y' and the corresponding true target Y can be calculated by a *loss function* that provides a *loss score* which quantifies the mismatch between predictions and true targets. The choice of an appropriate loss function is critical and depends on both the assumed likelihood distribution of the target variable and the nature of the task - for example, mean squared error usually corresponds to Gaussian regression tasks. In the next step, the loss gradient with respect to the weights and biases of k different layers, can be computed using *backpropagation* [33]. This procedure is based on the well-known chain rule from calculus. Applying this rule, backpropagation starts from the final loss score and propagates derivatives backward through the network to the first layer, as the name suggests.

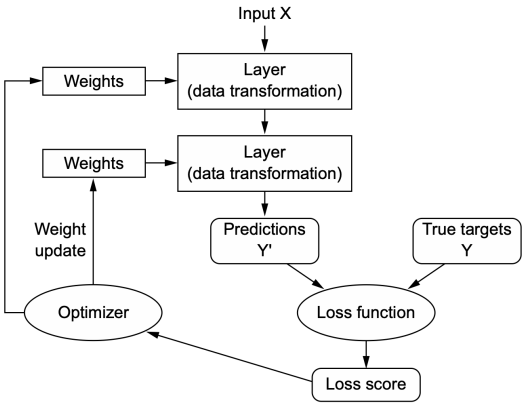


FIGURE 3.4.1: Schematic representation of the training of a neural network consisting of one input layer, one hidden layer and one output layer (explanation in text). Taken from [33].

The resulting gradients are then used by an *optimizer* to adjust the weights w_k and biases b_k of the kth layer in the opposite direction of the gradient, as shown in eq. (3.4.1), where α is the *learning rate*, m the number of mini-batches and L_{B_j} the loss of a specific mini-batch B_j [35]:

$$w_k' = w_k - \frac{\alpha}{m} \sum_j^m \frac{\partial L_{B_j}}{\partial w_k} \quad \text{and} \quad b_k' = b_k - \frac{\alpha}{m} \sum_j^m \frac{\partial L_{B_j}}{\partial b_k}. \tag{3.4.1}$$

By updating the parameters according to the above equation, the optimizer minimizes the loss score, which consequently leads to a better approximation of the target values. If the parameters are updated for all mini-batches B_j available in the training dataset, a so-called *epoch* is finished and the calculation continues for further iterations or epochs, which result in further descents of the loss.

The choice of optimization parameters strongly affects training. Larger batch sizes generally yield more stable updates but require more memory and training time, while smaller batches are faster but noisier. Similarly, the learning rate must balance convergence speed and stability: values that are too small slow down training, while values that are too large risk overshooting the minimum. In both cases, a compromise is required. Often, SGD is used with so-called *momentum* [33], which incorporates previous updates when calculating

the next one. This reduces the effect of noisy gradients and help the optimizer escape local minima.
During training, the loss on the training dataset is monitored together with that of a sample of events unseen by the network, the so-called *validation set*. This is done to avoid *overfitting* [33], which occurs when the network stops learning general features and instead begins to memorize the training dataset. Overfitting manifests itself as an increase in the loss of the validation set over successive epochs, in contrast to the continuously decreasing loss curve of the training dataset. Training is often stopped at this turning point, a technique known as *early stopping*. In order to avoid overfitting and to improve the *generalization* [33] of the model, the easiest way is to increase the number of training data or to reduce the network's capacity, particularly the number of layers and the number of units/filters per layer. Another solution is *weight regularization*. This approach put constraints on the network by forcing the weight to be small, so that the distribution is regular. It is done by adding a *cost* C to the loss function, which is proportional to the absolute value (*L1 regularization*) or squared value (*L2 regularization* also called *weight decay*) of the weight coefficients w_{ki} of k layers. The costs can be calculated with the following equations, which in addition to the weights contain a fixed regularization factor λ:

$$C_{\mathrm{L1}} = \lambda \cdot \sum_{k,i} |w_{ki}| \quad \text{or} \quad C_{\mathrm{L2}} = \lambda \cdot \sum_{k,i} |w_{ki}|^2. \tag{3.4.2}$$

Consequently, according to eq. (3.4.2), a large weight increases the loss score, which will be penalized in subsequent iterations.

3.5 Quantization of Neural Networks

Quantization [36] is a special neural network technique, which allows faster inference of neural networks by changing the datatype used for calculations from *float32* to *int8*. This reduces the memory requirements during computations significantly and the inference time. Unlike what one might expect, quantization does not simply round floating-point data directly to integers, but rather transforms the data by so-called *mapping functions*, which map an input $x \in [\alpha, \beta]$ to a quantized value $\tilde{x} \in [\tilde{\alpha}, \tilde{\beta}]$.

The quantization of an input x follows a linear transformation given by eq. (3.5.1), where s and z are quantization parameters:

$$\tilde{x} = \mathrm{round}\left(\frac{x}{s} + z\right). \tag{3.5.1}$$

The *scaling factor* s in eq. (3.5.1) is the ratio of the input range $[\alpha, \beta]$ to the quantized range $[\tilde{\alpha}, \tilde{\beta}]$, while the *zero point* z describes the position of the zero value in the quantized interval. Since the data are transformed into 8-bit integers, the quantized range is chosen as $[\tilde{\alpha}, \tilde{\beta}] = [0, 2^8 - 1] = [0, 255]$. The corresponding formulas for the quantization parameters are shown in the following:

$$s = \frac{\beta - \alpha}{\tilde{\beta} - \tilde{\alpha}}, \tag{3.5.2}$$

$$z = -\left(\frac{\alpha}{s} - \tilde{\alpha}\right). \tag{3.5.3}$$

It is important to note that although the inputs to the matrix multiplications are *int8*, the intermediate results are accumulated in *int32*. After the matrix calculations, the output is

rescaled and quantized back to *int8* before being passed to subsequent layers. This is known as integer-arithmetic-only inference [37].
In order to dequantize (convert back) the data into normal values, the quantized value has to be mapped back by the rearranged formula from eq. (3.5.1):

$$x' = (\tilde{x} - z) \cdot s. \tag{3.5.4}$$

However, the dequantized value x' is not exactly identical to the input x due to the rounding operation in eq. (3.5.1). The difference between the value x' and x constitutes the *quantization error*.

In addition to asymmetric quantization explained above, it is also possible to use symmetric quantization which maps an interval $[-\alpha, \alpha]$ to a quantized interval $[-\tilde{\alpha}, \tilde{\alpha}] = [-127, 127]$ without using a zero point z. Which quantization method is the best, depends on the range of values used in each layer of a network and can be specified for the weights of a layer and the output of a activation function separately. The process of estimating this input range $[\alpha, \beta]$ is known as *calibration* and can be done by various *observers*, which collects statistics on the input values of each layer, in order to enable the calculation of quantization parameters.

4 Neural network implementation

The goal of this thesis is to predict the PMT hit probabilities for incident photons in mDOMs. Initial efforts toward a similar task were presented in [5] for the case of the IceCube-Gen2DC-16 optical module, where the OMNNSim project [38] was implemented using the deep learning library *PyTorch* [39] of *Python* to build, train, and test a neural network for this task. The present work takes that study as a starting point and extends the OMNNSim framework to include the mDOM. In this chapter, the task of the neural network will first be outlined and explained in depth. In the next step, the network used in [5] together with the modifications introduced for the inclusion of the mDOM will be described. Finally, the focus is placed on an adaptation of the neural network to include quantization techniques.

4.1 Neural Network task

Before discussing the neural network architecture used in this thesis, it is important to clarify the following aspects: The main goal of the networks trained in this work, beyond the scope of this thesis itself, is their deployment within the IceCube-Upgrade simulation chain, more precisely after photons produced by charged particles are propagated through the ice and arrive at any of the mDOMs. As suggested in [5], the approach explored here is to propagate photons only up to spherical surfaces representing the mDOM, and then map the photon wavelength, landing position, and landing direction into detection probabilities for each PMT. From these probabilities, Monte Carlo photoelectrons can be sampled. The task presented to the neural network can therefore be summarized as follows: *given the description of a photon on a sphere, predict its detection probability for each PMT*. One of the spheres considered in this thesis is shown in green in fig. 4.1.1, enclosing the mDOM in the visualization of the Geant4 simulation used for training. Further details about this simulation can be found in chapter 5.

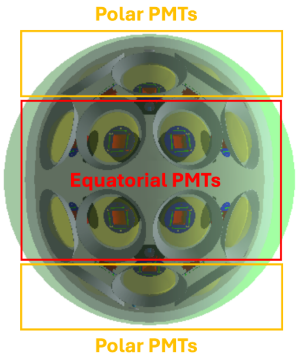


FIGURE 4.1.1: Spherical sensitive volume enclosing the mDOM used in Geant4 (see chapter 5). PMTs can be devided into polar (orange) and equatorial PMTs (red).

4.2 Neural network architecture

One of the main breakthroughs in neutrino astronomy in recent years was the discovery of neutrinos from the galactic plane of our galaxy [23]. This was made possible by new machine learning techniques that directly incorporate the symmetries of the IceCube detector. The OMNNSim project was inspired by these techniques to approximate the task described above for the case of the IceCube-Gen2DC-16 optical module, by exploiting the fact that all PMTs of the same type – polar or equatorial – within an optical module are equivalent.
A schematic view of the neural network for the mDOM is provided in fig. 4.2.1, where the data transformation follows a top-down flow. It can be divided into three parts each serving a different function. The most important part is the **CNN branch**, which calculates *relative*

inputs, in order to exploit the symmetry of the mDOM. If features that induce asymmetries are considered, such as the harness, it is also possible to include the **MLP branch**, which works directly on the *absolute inputs*. In the third part, the **prediction head**, the outputs of both branches are concatenated and mapped to PMT hit probabilities. For a better understanding of the different parts, each part will be described in the corresponding subsection.

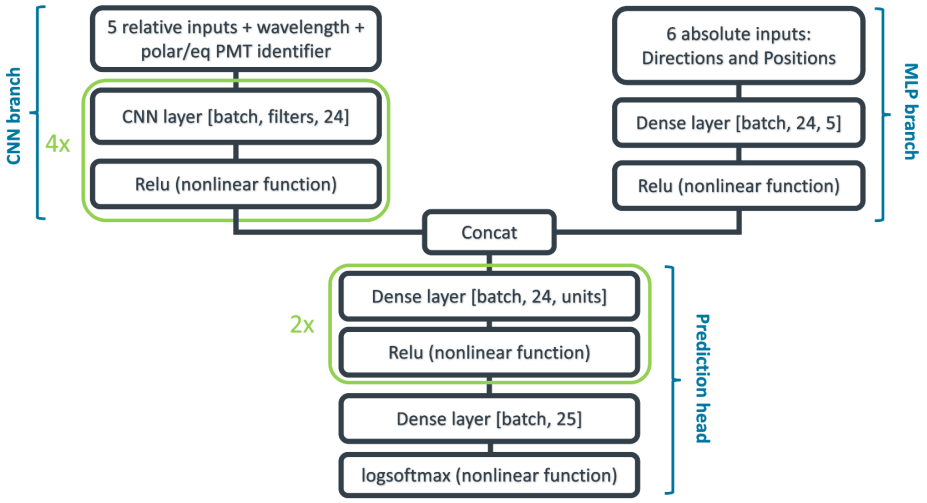


FIGURE 4.2.1: Schematic representation of the neural network used in this thesis and presented in [5].

TABLE 4.2.1: Input features used for the network, consisting of relative inputs describing symmetries and absolute inputs which consider symmetry-breaking features

relative inputs	**absolute inputs**
1. Wavelength λ	1. Photon position x_{pos}
2. Relative z position Δz_{pos}	2. Photon position y_{pos}
3. Absolute photon z-direction $z_{\text{dir}}^{\text{photon}}$	3. Photon position z_{pos}
4. Relative azimuthal position $\cos(\Delta\varphi_{\text{pos}})$	4. Photon direction x_{dir}
5. Relative azimuthal direction $\cos(\Delta\varphi_{\text{dir}})$	5. Photon direction y_{dir}
6. Convergence (binary value)	6. Photon direction z_{dir}
7. Polar/Equatorial (binary value)	

4.2.1 CNN branch

The aim of this branch is to exploit the fact that all PMTs of the same type – polar or equatorial – are equivalent. This is achieved by using a stack of four 1D convolutional layers with kernel size 1×1 on an input tensor of shape (*batch size* $\times$ *relative-input features* $\times$ 24), where 24 is the number of PMTs in the mDOM. With kernel size 1×1, the convolution is applied independently at each PMT position and the same weights are shared across all 24 positions, so an identical set of relative inputs is processed identically at every PMT index. In this way, by carefully selecting a relative description of the photon for each PMT, the symmetries can be encoded in the network.

To decide how to define the relative inputs, they should be based on the symmetries of the PMT angular acceptance within the mDOM. Unlike the IceCube-Gen2DC-16 optical module, the mDOM is approximately spherical, so a test was performed to determine whether a

spherical symmetry description of the photons would be suitable. In the end, it was found that the mDOM PMTs do not conform well to this approximation, which would negatively affect the accuracy. Therefore, the same inputs as in [5] were used, describing the photons based on cylindrical coordinates. Further details about these tests can be found in Appendix section A.1.
The relative inputs are listed in the left part of tab. 4.2.1 (a detailed description is given in Appendix section A.2). They describe the photon position and direction based on cylindrical coordinates and include a polar/equatorial PMT label to distinguish the two PMT types. These relative inputs are computed dynamically by this branch from the absolute true inputs of the network, that can be found in the right column of the same table tab. 4.2.1. The positions and orientations of the PMTs within an mDOM were included in the framework to enable this.
Overall, seven relative inputs, including wavelength, are used to form an input tensor of shape (*batch size*, 7, 24). Two inputs, being the wavelength and the relative z-position, have to be normalized before entering the network. In case of the relative z-position, the corresponding values were scaled to the range $[-1, 1]$ by division through the radius of the spherical volume, while the wavelengths were normalized to the range $[0, 1]$. In contrast to these two inputs, the cosine-based features and the binary ones are already in the correct range. The ReLU activations between layers are used to learn non-linear patterns. They can be written as [33]:

$$\mathrm{ReLU}(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases}.$$

4.2.2 MLP branch

Under certain circumstances, such as the presence of the harness, the PMT-equivalence symmetry is partially broken. In such cases, an additional MLP branch is included. This branch consists of a dense layer that maps the six normalized absolute photon inputs on the sphere (right column of tab. 4.2.1) to a five-dimensional representation, followed by a ReLU activation. The output of this branch, with shape (*batch size*, 24, 5), is concatenated with the output of the CNN branch and fed into the prediction head.

4.2.3 Prediction head

The outputs of the two preceding branches are concatenated (the CNN output is reshaped accordingly) and fed into a stack of two dense layers, interleaved with ReLU activations. After flattening, a final dense layer maps the features to shape (*batch size*, 25) to include the non-detection class as an additional index. Finally, a log-softmax activation is applied along the last dimension. This can be written as [40]:

$$\mathrm{LogSoftmax}(x_i) = \log\left(\frac{\exp(x_i)}{\sum_j \exp(x_j)}\right). \tag{4.2.1}$$

Here, the bracketed term denotes the *softmax*, which ensures the 25 outputs sum to one and can therefore be interpreted as a probability mass function over the 25 outcomes. By using *logsoftmax* instead of *softmax* a higher training stability can be achieved. Exponentiating the result of eq. (4.2.1) after training yields the detection probability for each PMT and for a failed detection.

4.2.4 Training

The training of the neural network requires the definition of a suitable loss function and optimizer. An appropriate loss function for comparing probability distributions is the *Kullback-Leibler divergence* [41] consisting of a real discrete probability mass function $p_i(x)$ and a predicted one $q_i(x)$ of a sample (photon) i for different PMTs x:

$$L_i^{\mathrm{KLD}} = \sum_{x=1}^{25} p_i(x) \log\left(\frac{p_i(x)}{q_i(x)}\right). \tag{4.2.2}$$

In case of this network, the real probability distribution $p_i(x)$ refers to the training data generated by the Geant4 simulation explained in chapter 5, while the predicted one $\log(q_i(x))$ refers to the output of the *logsoftmax* activation function. In order to get a scalar for the loss score, the losses computed for each sample in a batch according to eq. (4.2.2) will be averaged. Moreover, the loss is complemented by the *mean squared error* [42] multiplied by a factor of 0.1, which contributes to a higher stability. According to its name, this error, depending on the predicted probability $q_i(x)$ and the real probability $p_i(x)$ of a batch sample i, can be calculated as follows:

$$L_i^{\mathrm{MSE}} = \frac{1}{25} \sum_{x=1}^{25} \left[p_i(x) - q_i(x)\right]^2. \tag{4.2.3}$$

Thus, the whole loss score for a set of all probabilities $P = \{p_i(x)\}$ and $Q = \{q_i(x)\}$ within a batch can be computed with the following equation:

$$L(P,Q) = \frac{1}{n} \sum_{i=1}^{n} L_i^{\mathrm{KLD}} + 0.1 \cdot \frac{1}{n} \sum_{i=1}^{n} L_i^{\mathrm{MSE}}. \tag{4.2.4}$$

The *AdamW* [43] optimizer is used to update the trainable parameters of the network. It uses momentum and L2 regularization with weight decay, as explained in section 3.4. The learning rate was set to 0.0001, and a batch size of 10,000 was used. The models considered in this thesis use the same number N of filters in the CNN layers and hidden units in the dense layers, with a total of about 90,000 parameters for $N = 85$ and about 200,000 parameters for $N = 150$. The models were trained for 8 epochs on an NVIDIA Geforce RTX 4090 GPU, which took about 3 to 4 days, depending on the size of the model. Given that the total training set consists of 20 billion photon samples, overfitting is very unlikely. The production of these training samples will be explained in chapter 5.

4.3 Quantization with PyTorch

The previously discussed network was adapted in this work to use quantization techniques, implemented through the *PyTorch quantization* extension [36]. It is based on the theory described in section 3.5 and provides several types of quantization, which are explained in [36, 44]. In particular, *post-training static quantization* was employed in this work to the network trained for the mDOM without harness.
The implementation of the quantization approach was carried out according to [44] and follows the steps illustrated in fig. 4.3.1. The starting point of this approach is the code of the pre-trained model presented in section 4.2. Initially, the code was modified by fusing dense layers and convolutional layers with their respective *ReLU* activations into fixed modules. This leads to improved accuracy, as the fusion avoids inefficient dequantization and quantization steps between layers, which could otherwise increase the quantization error.

Subsequently, quantization and dequantization operations, so-called *stubs*, were inserted in the code to define the quantized domain in *int8*. Specifically, *QuantStubs* were placed immediately before the first layer of both the CNN branch and the MLP branch described in section 4.2, while the *DequantStub* was inserted prior to the *logsoftmax* activation.
For calibration, a dataset consisting of 10,000 photons, sampled from the original training data of the corresponding model, was used. A *MinMaxObserver*, which assigns the recorded minimum and maximum values of each module during calibration, uses the recorded values to define the input range $[\alpha, \beta]$. Furthermore, *per-channel quantization* was activated that allows independent quantization parameters (scaling factors and zero points) for each filter in a CNN layer.

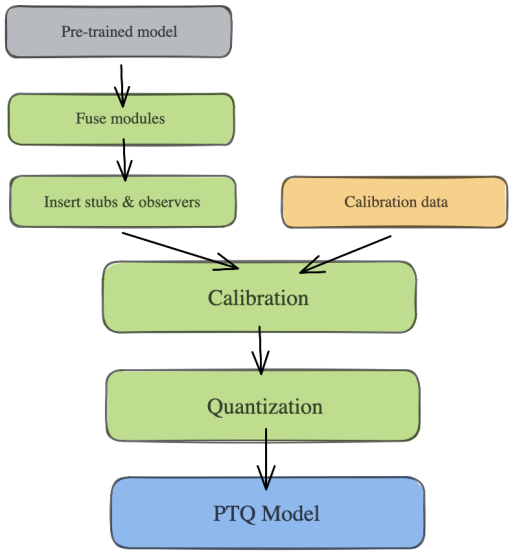


FIGURE 4.3.1: Steps in static Post-Training Quantization (explanation in text). Taken from [36].

After loading the weights and biases of the model, calibration is performed using the previously described calibration dataset and observer, resulting in quantization parameters. Finally, these parameters can be loaded to perform faster inference for calculating PMT hit probabilities.

5 Training & benchmark simulations

In this thesis, detailed Geant4 simulations of the mDOM are used both to train the neural network model that predicts the PMT hit probabilities and to benchmark its performance. For this purpose, the implementation of the mDOM within the Geant4 framework is first described. In order to evaluate the accuracy of different simulations, the concept of the effective area will be introduced. Subsequently, the generation of training data will be explained, including an analysis of a simplified harness. After a brief description into training data processing, the current simulation of the mDOM in IceCube will be discussed at the end of this chapter.

5.1 mDOM simulation in OMSim using Geant4

Geant4 [45] is a simulation toolkit, developed at CERN, that simulates the trajectory of particles through matter by means of Monte Carlo methods. It is based on the object-oriented programming language C++ and provides many different classes that enable, among other tasks, the definition of detector geometries and materials, as well as particle generation, interactions and tracking. Moreover, it allows the visualization of objects and particle trajectories.
Given the accuracy and success of the Geant4 toolkit, the IceCube Münster group developed the OMSim framework [7] in Geant4 for the simulation of different optical modules of IceCube and its future extensions. The framework enables a variety of studies, such as background estimation induced by radioactive decays, supernova detection, and assessments of optical module sensitivity via effective area.
One of these implemented modules is the mDOM, which was originally developed as part of two doctoral thesis [46, 47] and has since been refined by several subsequent theses [18, 48]. The realistically implemented mDOM consists of a pressure vessel filled with a gel. Inside of the vessel, the support structure is formed analogously and embeds 24 PMTs with conical reflectors, which are distributed over four rows in the module. If applicable, a harness encompassing the mDOM also can be placed. It consists of multiple bands, clamps, ropes, taps and the PCA cable including the plug. A screenshot of an mDOM with harness in OMSim is depicted in fig. 5.1.1.
In OMSim it is possible to simulate photons which undergo absorption as they propagate and boundary processes at interfaces between objects with different materials such as reflection and transmission. In order to characterize their interaction with the mDOM, all materials are specified by wavelength-dependent optical properties.
For transparent materials, these include the refractive index and the absorption length, while for other components, reflectivity is used. The crucial components for detecting photons within the mDOM

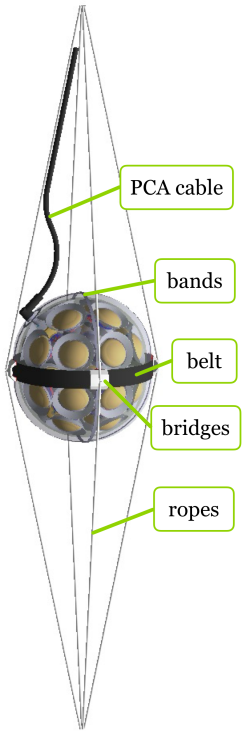


FIGURE 5.1.1: Visualization of the mDOM with harness in OMSim.

are the PMTs, or more specifically their photocathodes, which are simulated using thin-layers physics and a complex refractive index [49]. Every time a photon reaches the photocathode, the probabilities of reflection, transmission and absorption are computed. The absorption of a photon is treated as a detection. If a photon transmits through the photocathode, it can be reflected by internal PMT components and return to the photocathode, where a new detection probability is applied. Finally, the overall detection probabilities are calibrated via 2D scan measurements across the photosensitive region [18].
Furthermore, the environment outside the mDOM is set to ice, which accounts for a wavelength-dependent refractive index, while ignoring photon scattering and absorption.
Throughout this thesis, mentions of the Geant4 simulation refer exclusively to the mDOM simulation previously described within OMSim.

5.2 Effective Area

The effective area [50] is a suitable parameter for characterizing the sensitivity of the module as well as its angular acceptance. Therefore, it will be used to benchmark the performance of the neural network trained in this thesis.
A plane wave of uniformly distributed mono-energetic photons within a disk with a diameter larger than that of an mDOM is directed towards the center of the mDOM, as illustrated in fig. 5.2.1. The corresponding effective area A_{eff} for a particular incident direction described by θ and φ with wavelength λ is defined as

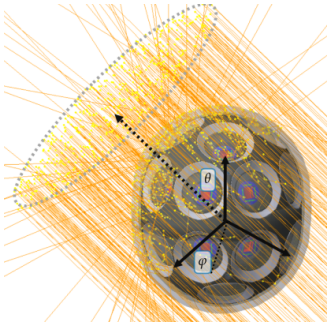


FIGURE 5.2.1: Effective area simulation for a certain direction (θ, φ). Taken from [18].

$$A_{\text{eff}}(\lambda, \varphi, \theta) = \frac{N_{\text{det}}(\lambda, \varphi, \theta)}{N_{\text{emit}}} \cdot A_{\text{beam}} \,, \tag{5.2.1}$$

where N_{emit} describes the number of emitted photons within a beam with area A_{beam}, while N_{det} corresponds to the number of detected photons in dependence of wavelength λ and direction (φ, ϑ).
In order to illustrate the effective area for different directions, the values will be plotted in *mollweide projections* using HEALPix[4] pixelization, which can be implemented with the Python package *healpy* [51]. In this package, a spherical surface is discretized into equal-sized pixels, which can be used to calculate the effective area isotropically. An exemplary mollweide projection is illustrated in fig. 5.2.2, where each pixel represents a direction and their color the corresponding value of the effective area. The resolution of the map can be defined by the NSIDE parameter, which is a power of 2. Commonly used NSIDEs in this thesis are 16 or occasionally 32 with $N_{\text{pixel}} = 3072$ or $N_{\text{pixel}} = 12288$ pixels (directions), respectively. Unless stated otherwise, the used NSIDE is 16 throughout the following sections, while NSIDE $= 32$ will be

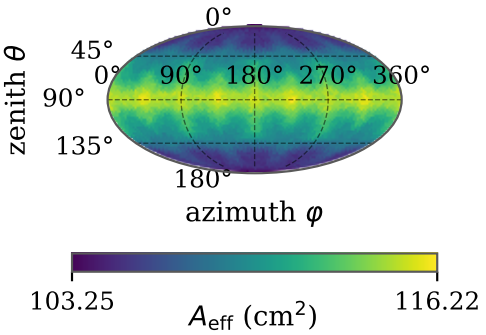


FIGURE 5.2.2: Exemplary mollweide projection of the effective area as a function of the incident direction used throughout this thesis (explanation in text).

[4] **H**ierarchical **E**qual **A**rea iso**L**atitude **Pix**elization [51]

indicated accordingly. Given the directions (θ, φ) and the number of directions N_{pixel} provided by *healpy*, the mean effective area (over solid angle) for a specific wavelength λ can be calculated as:

$$\overline{A}_{\text{eff}}(\lambda) = \frac{\sum_{\varphi,\theta} A_{\text{eff}}(\lambda, \varphi, \theta)}{N_{\text{pixel}}}. \tag{5.2.2}$$

5.3 Producing training data

The training data for the neural network was generated by a modification of the effective area simulation in OMSim that was originally used to produce training data for a neural network which calculates the detection probabilities of photons for an IceCube-Gen2DC-16 optical module [5]. By adapting this simulation to the structure and geometry of the mDOM, it also can be used to generate training data for the neural network of this thesis by the following mechanism.

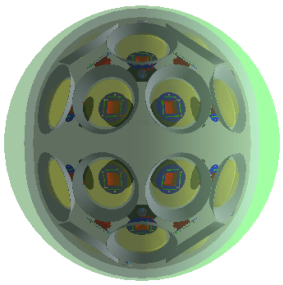


FIGURE 5.3.1: Geant4 visualization of the spherical sensitive volume (green) enclosing the mDOM, used to record photon properties at its surface.

The simulated geometry consists of a spherical sensitive volume with radius $R_{\text{sphere}} = 20.8\,\text{cm}$ that encloses the mDOM and the surrounding ice environment as shown in fig. 5.3.1. Photons are simulated independently, following the same procedure as in the effective area simulation from section 5.2, where each photon is generated uniformly over the surface of a disk, whose normal vector is sampled uniformly over the unit sphere. In addition, the wavelength of each photon is drawn from a uniform distribution in the range of 270 nm to 700 nm, where the quantum efficiency of the embedded PMTs is different from zero [18]. When a photon enters the sensitive spherical volume, its position, direction, and wavelength are recorded, thereby serving as inputs to the neural network. On the other hand, the true labels consist of the detection probabilities provided by OMSim for the different PMTs in a single-photon simulation.

5.4 Training data for mDOM with harness

In IceCube-Upgrade, once deployed, the mDOM will be surrounded by several components, such as a harness and the PCA cable, as shown in fig. 5.1.1. These elements can absorb or reflect photons, and it is therefore important to include their impact in the neural network. To capture the full influence of the harness, one would need to create a spherical sensitive volume that encompasses it and simulate photons throughout the entire volume. However, given the size of the harness, this is not feasible, since only a tiny fraction of photons would be detected by any of the PMTs, which make the inference of their probabilities challenging. To address this, only the most relevant part of the harness is considered, namely the section surrounding the mDOM. Specifically, all components within a radius of $R_{\text{sphere}} = 24.9\,\text{cm}$ from the center of the mDOM, corresponding to the radius of the new spherical sensitive detector, were kept there, while the rest were removed. The simplified version of the harness is shown in fig. 5.4.1.

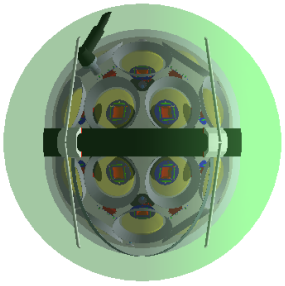


FIGURE 5.4.1: Geant4 visualization of a simplified version of the harness considered for the neural network training inside the spherical sensitive volume.

5.4.1 Influence of a simplified harness

In order to check whether the mDOM with a simplified harness provides a suitable approximation for the mDOM with the entire harness, effective area differences will be compared using Geant4 simulations. The effective areas for an mDOM without harness will also be provided for comparison. In contrast to all other sections, this section uses an NSIDE parameter of 32, which corresponds to $N_{\text{pixel}} = 12288$ directions.

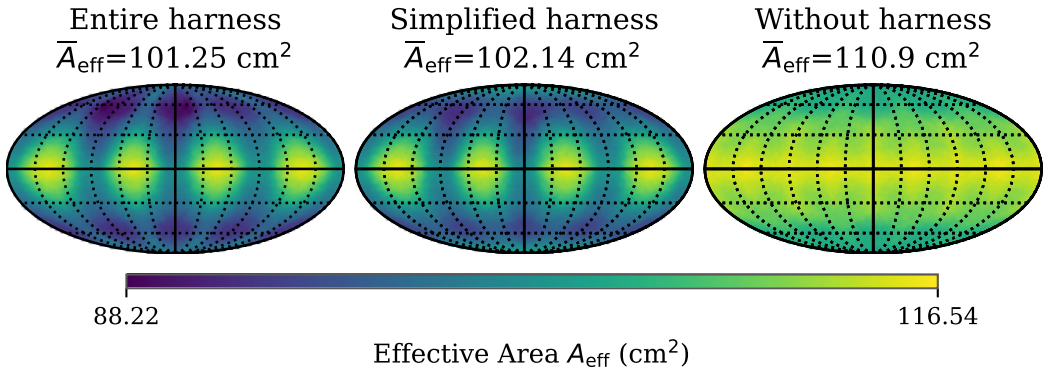
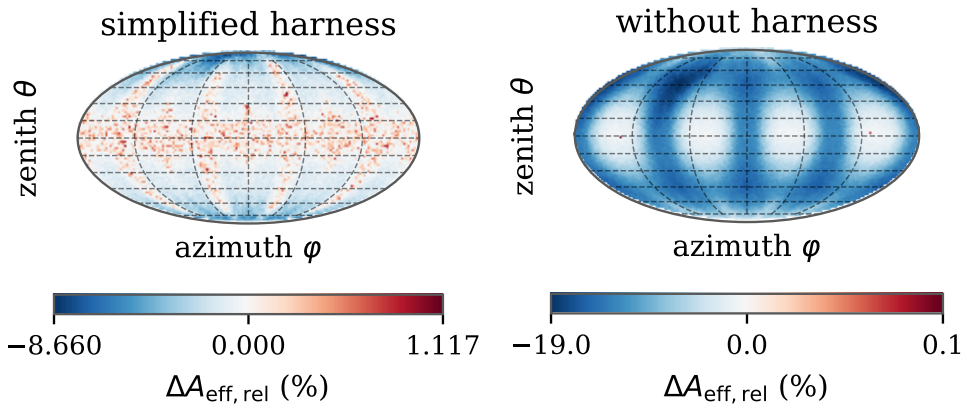


FIGURE 5.4.2: Effective area A_{eff} for different directions and harness configurations at a wavelength of $\lambda = 400\,\text{nm}$ for the whole detector.

The effective areas as a function of the incident direction at a wavelength of $\lambda = 400\,\text{nm}$ for different harness configurations is shown in fig. 5.4.2. A comparison between the simplified harness and the entire harness shows, that both effective areas are distributed similarly. In order to compare the different versions in detail, the relative differences will be calculated using the following formula:

$$\Delta A_{\text{eff,rel}}(\lambda,\theta,\varphi) = \frac{A_{\text{eff}}^{\text{harness}}(\lambda,\theta,\varphi) - A_{\text{eff}}^{\text{compared}}(\lambda,\theta,\varphi)}{\overline{A}_{\text{eff}}^{\text{harness}}(\lambda)}. \tag{5.4.1}$$

$A_{\text{eff}}^{\text{harness}}(\lambda,\theta,\varphi)$ is defined as the effective area of the mDOM with the entire harness for a certain wavelength λ and direction (θ,φ), while $A_{\text{eff}}^{\text{compared}}(\lambda,\theta,\varphi)$ refers to the mDOM with a simplified harness or without harness. After applying eq. (5.4.1) on the mean effective areas provided by fig. 5.4.2, the overall difference between an mDOM with a simplified and an entire harness yields $\Delta\overline{A}_{\text{eff,rel}} = -0.88\%$, whereas the difference between an mDOM with and without an entire harness is $\Delta\overline{A}_{\text{eff,rel}} = -9.53\%$.
The relative differences for various directions, calculated by eq. (5.4.1) are illustrated in fig. 5.4.3 (a) for a comparison with an mDOM with a simplified harness and in (b) for a comparison with an mDOM without harness.
According to fig. 5.4.3 (a), the maximum differences are on the order to 8% and occur only in a small region in the upper-left part of the plot. This can be mainly attributed to the PCA cable simplification, which strongly affects those directions. However, in the most directions the differences are small and about 1%.
By contrast, when comparing a model without a harness to one with the full harness, much larger differences can be observed, with values up to 19% in the upper-left region. This indicates that the simplified harness captures most of the full-harness shadowing and is a reasonable approximation, while enabling training within a moderately sized sensitive spherical volume.



(a) (b)

FIGURE 5.4.3: Relative difference of the effective area $\Delta A_{\text{eff,rel}}$ for different directions based on the effective area illustrated in fig. 5.4.2 for **(a)** mDOM with simplified harness and **(b)** mDOM without harness.

5.5 Processing of training data

Overall, 20 billion photons were produced for training using the Geant4 simulation for an mDOM with and without harness, respectively. The stored values for each photon consists of one wavelength, three position coordinates, three direction coordinates and 24 detection probabilities, one for each PMT. The default output file format of OMSim is ".dat"; however, this is not ideal for machine-learning training. Therefore, using the OMNNSim framework, the simulation results were converted to the '.hdf5" format, which is more suitable and offers better compression, reducing disk usage.

5.6 Current mDOM simulation in IceCube

Currently, the mDOM simulation is implemented in the context of IceCube simulations in the PPC [5] framework [4]. Simplified, this approach parametrizes the relative angular acceptance of each PMT by a function $f(\beta, \vec{m}\cdot\vec{n})$, where $\vec{m}$ denotes the PMT axis, $\vec{n}$ the photon direction, and β a shape parameter of the PMT's sensitive area (in the case of the mDOM, accounting for both the photocathode and the surrounding reflector). The current value is $\beta = 0.95$, where $\beta = 1$ would correspond to a flat surface.
If f is normalize (e.g., to unit integral or unit maximum), it can be regarded as an angular probability density. The *absolute* angular acceptance at wavelength λ is then given by $A_{\text{eff}}(\lambda, \vec{n}) = \overline{A}_{\text{eff}}(\lambda)\, f(\beta, \vec{m}\cdot\vec{n})$, where $\overline{A}_{\text{eff}}(\lambda)$ is the mean effective area determined from the previous Geant4 simulations.
This analytical approximation preserves an expected solid-angle average effective area, which characterizes the angular acceptance of the mDOM. However, it neglects some aspects of geometry and the optical properties of the mDOM. For example, it enforces a response that is symmetric with respect to the PMT axis despite the lack of alignment between the PMT axes and the normal vector of the pressure vessel. Moreover, it ignores the wavelength-dependent optical properties of the glass, gel and reflector cones, which are specifically the reflectivity, absorption length and refractive index. The corresponding effective area simulations using this method for benchmark were produced by [53].

[5] **P**hoton **P**ropagation **C**ode [52]

6 Neural network performance

In this chapter, the performance of neural networks with the architecture described in chapter 4, for mDOMs with and without harness, is presented. Given that the long-term goal of this tool is the deployment on IceCube-Upgrade simulations, the network must be not only accurate but also fast enough for large-scale simulation production. For this reason, the effective areas and computing times of OMSim simulations (Geant4), the neural networks (NN) trained in this work, and the current IceCube analytical approximation (PPC) will be compared. In addition, the performance of a test model without a harness using quantization methods, a promising technique for reducing the inference time of neural networks, will be discussed.

6.1 Effective area performance

The accuracy of the neural network, based on the effective area of the mDOM it provides, will be discussed in this section. To this end, it will be compared to that of Geant4 (the baseline) and, where applicable, to the analytical approximation. The performance of a model without the harness, with the harness, and with quantization techniques will be presented.

6.1.1 Accuracy for the mDOM without harness

The effective area of a trained neural network model, as a function of wavelength and incident direction, can be computed with the OMNNSim simulation, which includes a function that samples photons on the network's input sphere according to the spatial distribution obtained by projecting a plane wave of photons onto the sphere.

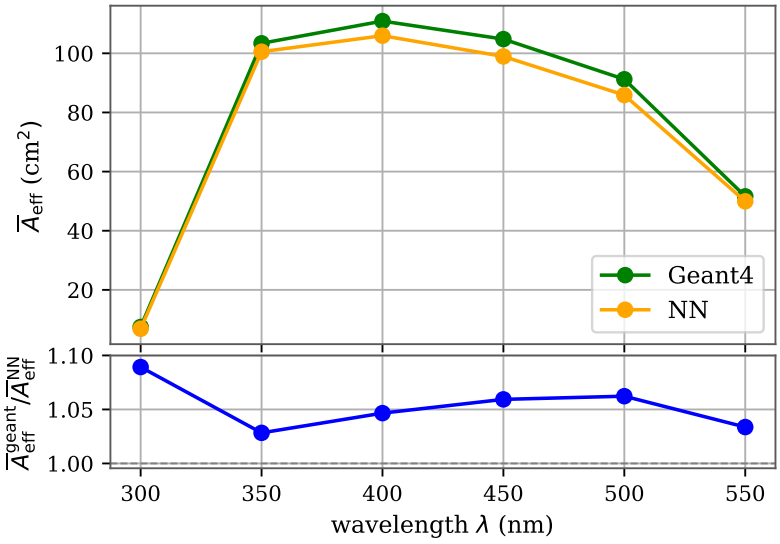


FIGURE 6.1.1: Mean effective area $\overline{A}_{\text{eff}}$ for different simulations and wavelengths together with the ratio $a(\lambda) = \overline{A}_{\text{eff}}^{\text{geant}}/\overline{A}_{\text{eff}}^{\text{NN}}$, which constitutes the scaling factor.

The mean effective area over 3072 incident directions of a neural network with 85 units per dense layer and the same number of filters per CNN layer including the MLP branch is plotted as a function of wavelength, together with the corresponding Geant4 values in fig. 6.1.1. Although the overall trend is similar, the neural network slightly underestimated the mean effective area with a difference of about 5%. This can be explained by the fact that the non-detection probability (index 25 of the true target) always has a non-zero value, at least around 0.7, whereas the other PMT indices are zero most of the time, and only occasionally does one of them take a modest value between approximately 0.2 and 0.3. This induces a small bias toward predicting relatively larger values for the last index, because doing so reduces the loss in most cases. This bias can be corrected by scaling the network outputs by the factor $a(\lambda) = \overline{A}_{\text{eff}}^{\text{geant}} / \overline{A}_{\text{eff}}^{\text{NN}}$, which corresponds to the bottom plot of fig. 6.1.1. This scaling is applied to the detection probabilities of every network in this chapter. As mentioned in section 5.6, the analytical approach is designed to reproduce the expected mean effective area and is therefore not shown in the plot.

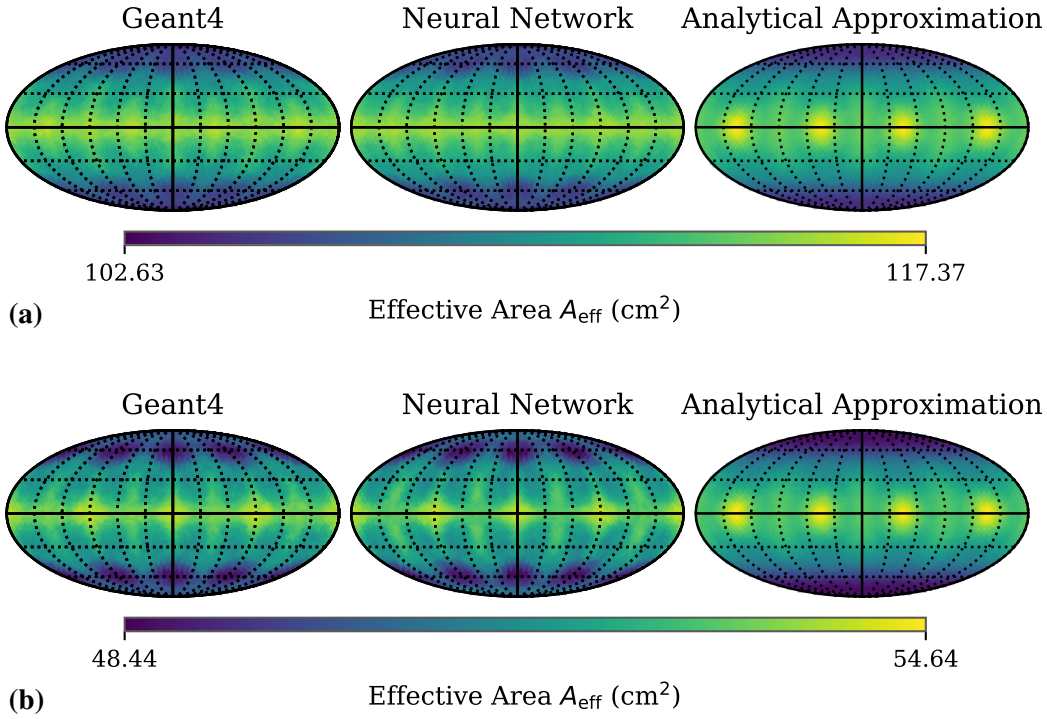
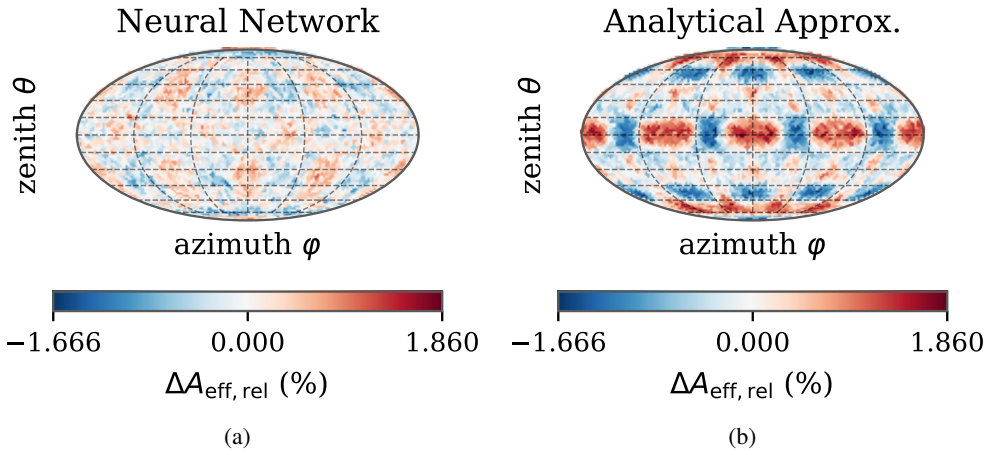


FIGURE 6.1.2: Effective area A_{eff} for the mDOM as a function of the direction for two different wavelengths **(a)** $\lambda = 400$ nm and **(b)** $\lambda = 550$ nm and from left to right for the Geant4 simulation, the neural network and the analytical approximation.

After scaling the effective areas for different directions provided by *healpy*, the values are visualized using mollweide projections, as shown in fig. 6.1.2. The effective area is illustrated for wavelengths of $\lambda = 400$ nm in fig. 6.1.2 (a) and for $\lambda = 550$ nm in fig. 6.1.2 (b). In both cases, the pattern obtained with the neural network approach matches the one from Geant4 very closely and, as expected, it is able to adapt to different patterns at different wavelengths. This behavior reflects the wavelength-dependent optical properties of the various components of the mDOM, such as the glass, gel, and conical reflectors. In contrast, the analytical approximation always produces the same pattern that clearly differs from the others. These differences are illustrated more clearly in fig. 6.1.3, which displays the relative deviations in the effective area for the neural network (a) and the analytical approximation



(a) (b)

FIGURE 6.1.3: Effective area differences $\Delta A_{\mathrm{eff,rel}}$ according to eq. (6.1.1) at directions provided by *healpy* for $\lambda = 400$ nm between Geant4 and **(a)**: neural network or **(b)**: analytical approximation; blue color means overestimation, while red color means underestimation.

(b) calculated by the following equation:

$$\Delta A_{\mathrm{eff,rel}}(\lambda,\theta,\varphi) = \frac{A_{\mathrm{eff}}^{\mathrm{geant}}(\lambda,\theta,\varphi) - A_{\mathrm{eff}}^{\mathrm{approx.}}(\lambda,\theta,\varphi)}{\overline{A}_{\mathrm{eff}}^{\mathrm{geant}}(\lambda)}. \tag{6.1.1}$$

$A_{\mathrm{eff}}^{\mathrm{geant}}(\lambda,\theta,\varphi)$ and $\overline{A}_{\mathrm{eff}}^{\mathrm{geant}}(\lambda)$ define the effective area and the mean effective area provided by the Geant4 simulation, respectively, while $A_{\mathrm{eff}}^{\mathrm{approx.}}(\lambda,\theta,\varphi)$ denotes either the effective area approximated by the neural network or the analytical approximation.
From the plots shown in fig. 6.1.3, it is evident that the main discrepancies in the analytical approximation appear in the equatorial and polar directions, while the neural network shows smaller deviations, primarily in the mid-latitudes.
The case of the single polar PMT is shown in fig. 6.1.4. Once again, a closer agreement between Geant4 and the neural network can be observed, which is even able to capture a ring-like feature at longer wavelengths around the central maximum, as well as the correct extension of the pattern across different wavelengths. Analogous results for an equatorial PMT are provided in Appendix section A.3.
To quantify performance across all wavelengths, percentiles of the distribution of absolute relative differences in effective area $\Delta A_{\mathrm{eff,rel}}$ evaluated at 3072 sky directions from Healpy are calculated according to eq. (6.1.1). The 50th percentile (median) and the 90th percentile for both the neural network and the analytical approximation as a function of the wavelength are illustrated in fig. 6.1.5 for three cases, namely, the entire module, a single polar PMT and an equtorial PMT, where the denominator in eq. (6.1.1) corresponds to the mean effective area of the respective case (corresponding distributions for this neural network configuration are provided in Appendix section A.4). Throughout this thesis, these percentiles will be denoted as *performance percentiles*.
In case of the full optical module shown in fig. 6.1.5, the neural network performs better than the analytical approximation, since its differences are always smaller than 1% (except for 300 nm), while the 90th percentile of the analytical approximation yields values between 1% and 2%. The highest differences result for the wavelength of $\lambda = 300$ nm, where a large amount of photons is absorbed before reaching the PMTs as a result of the short absorption lengths of the glass (pressure vessel) and the gel at this wavelength. Therefore, the PMT hit probabilities in the training dataset mainly contain zeros, which are insufficiently informative

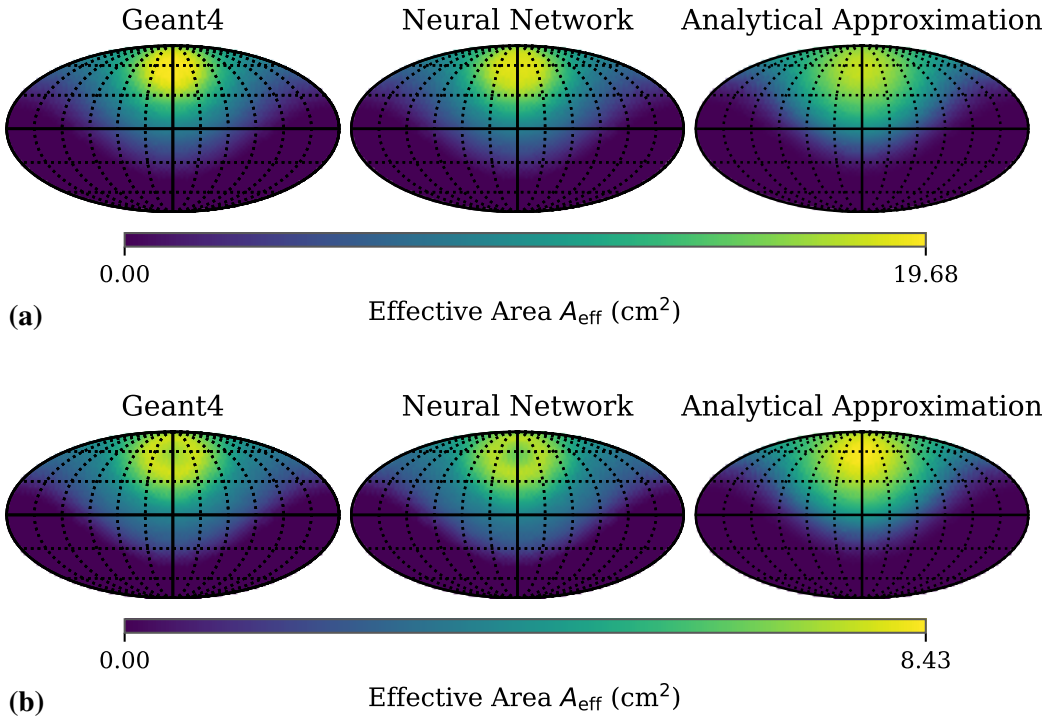


FIGURE 6.1.4: Effective area A_{eff} for a single polar PMT as a function of the direction for two different wavelengths **(a)** $\lambda = 400\,\text{nm}$ and **(b)** $\lambda = 550\,\text{nm}$ and from left to right for the Geant4 simulation, the neural network and the analytical approximation. Dots represent the calculated values and are connected with lines for better visibility.

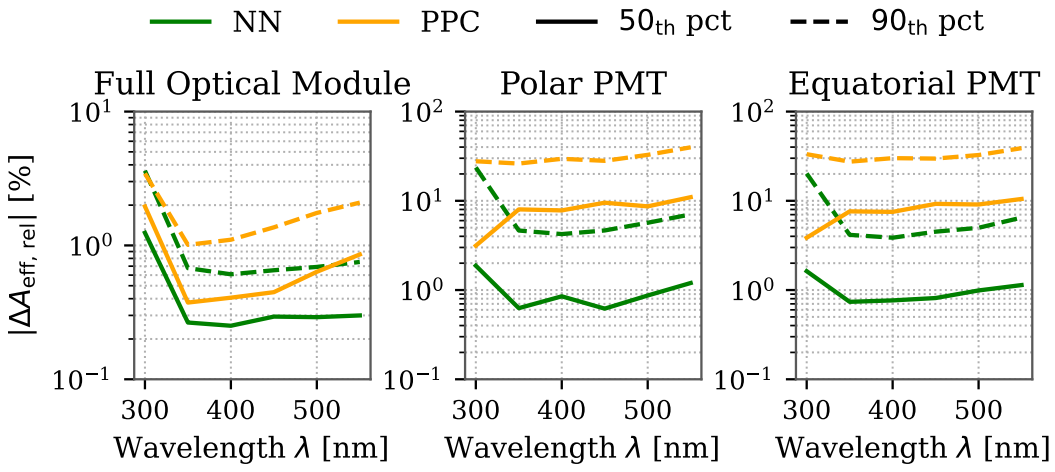


FIGURE 6.1.5: 50th (solid) and 90th (dashed) percentiles of the relative differences $|\Delta A_{\text{eff,rel}}|$ (in %) across 3072 isotropic directions, plotted versus wavelength. Absolute difference $|\Delta A_{\text{eff,rel}}|$ illustrated by the 50th and 90th percentile (pct) for different wavelengths and simulations (NN: neural network; PPC: analytical approximation).

for training. Despite twice as many photons are produced by the Cherenkov effect at $\lambda = 300\,\mathrm{nm}$ than at $\lambda = 400\,\mathrm{nm}$ ($N_{\mathrm{photon}} \sim 1/\lambda^2$ [54]), the mean effective area at $\lambda = 400\,\mathrm{nm}$ is about 20 times bigger than at $\lambda = 300\,\mathrm{nm}$. Moreover, the absorption length of the ice at $\lambda = 300\,\mathrm{nm}$ is also significantly shorter than for longer wavelengths in the IceCube simulation. As a result, the impact of photons with a wavelength of 300 nm is expected to be low and thus acceptable.
The most significant improvement, however, can be observed for single PMTs, as the medians of the neural network is about 1% and one order of magnitude lower than in case of the analytical approximation.

6.1.2 Influence of different neural network parameters

As explained in section 4.2, the neural network can be configured by several parameters, such as the number N of units (filters) per dense (CNN) layer and the consideration of the MLP branch. Moreover, it is possible to vary the amount of training data. The impact of these parameters on the accuracy of the corresponding models is depicted in fig. 6.1.6, where

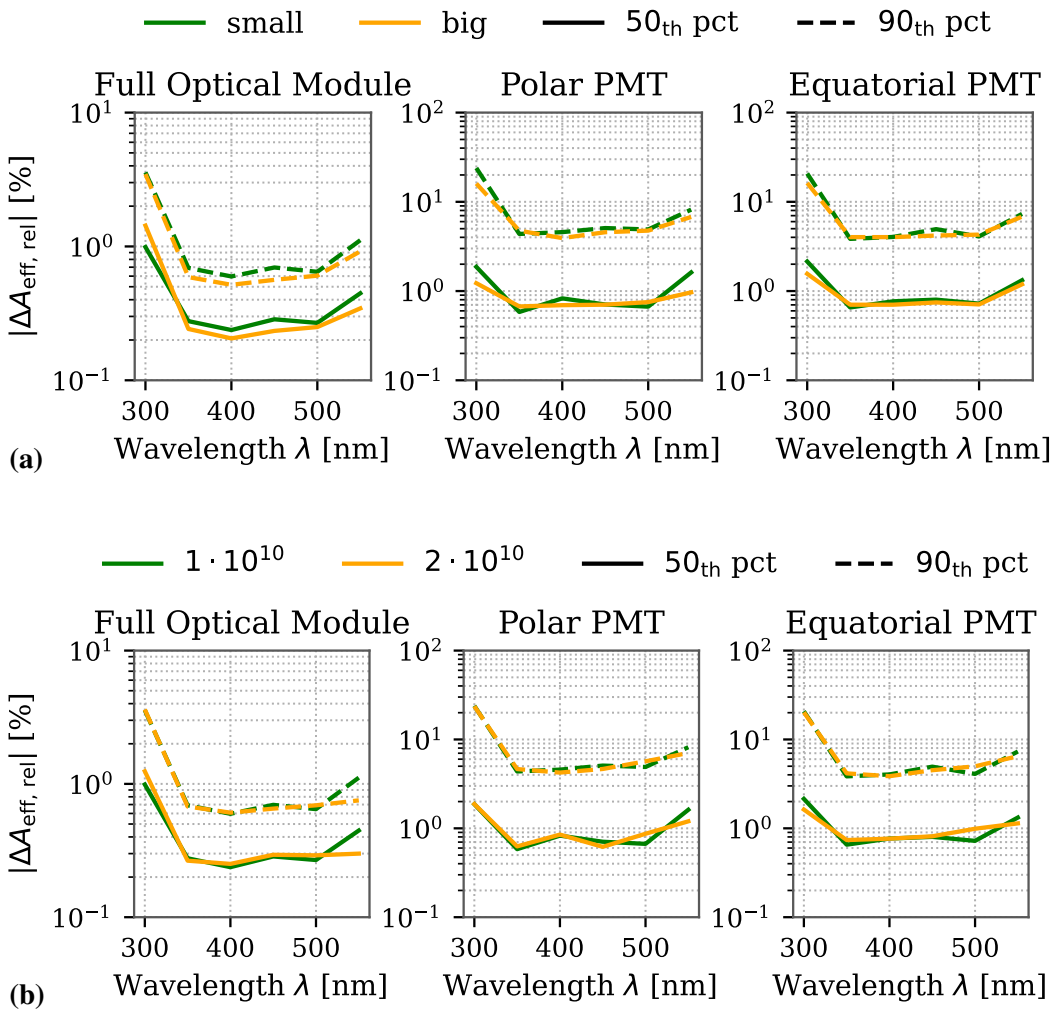


FIGURE 6.1.6: Performance percentiles plotted versus wavelength and for different neural network configurations: **(a)**: small (85 units and filters) or large model (150 units and filters) without MLP branch trained on 10 billion samples; **(b)** small model including MLP branch trained on 10 or 20 billion samples.

(a) illustrates the influence of the model size, comparing models trained with $N = 85$ or $N = 150$ units and filter, and (b) the amount of training data between 10 and 20 billion.
The influence of the model size, depicted in fig. 6.1.6 (a), demonstrates that a larger model yields slightly higher accuracy in all cases, since more parameters improve the model's ability to learn new and more complex representations. However, this improvement on accuracy is very subtle, being about 0.1 percentage points in the case of the entire optical module and even lower for single PMTs, which does not justify the increased inference time observed in section 6.2 (see tab. 6.2.1). Therefore, the focus will be placed on the smaller model with $N = 85$ units and filters.
In fig. 6.1.6 (b), the impact of the amount of training data is examined. A higher accuracy would generally be expected with a larger training dataset, since the model can learn additional cases of incident photons. However, no significant improvements were observed, suggesting that 10–20 billion training samples are sufficient. Further increasing the size of the training dataset is unlikely to yield better performance.
The impact of adding the MLP branch to this model that does not consider the harness can be observed by comparing the green lines between plot (a) and (b), as both use the same number of units, filters, and training data. As can be observed, this branch does not provide any improvement in performance for a model without symmetry-breaking features. Nevertheless, it was kept, since it adds only a negligible number of extra parameters and allows for a unified model throughout this thesis.
The standardized model for subsequent analyses consists of 85 units and filters including the MLP branch and is trained on 20 billion training data for 8 epochs. If other model configurations are used, this will be mentioned at the corresponding point.

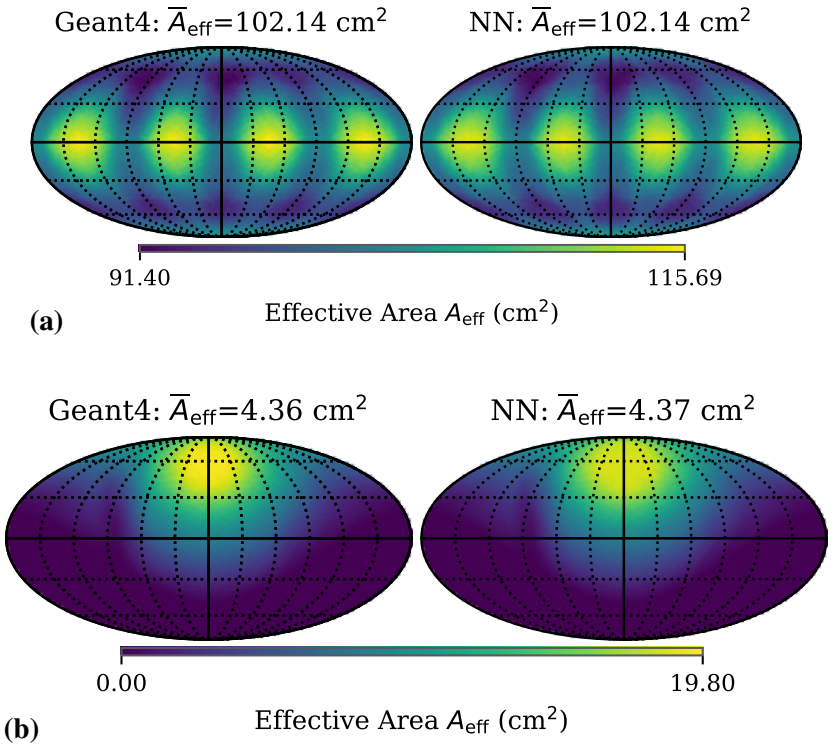


FIGURE 6.1.7: Mean effective area $\overline{A}_{\text{eff}}$ and effective area $A_{\text{eff}}(\theta, \varphi)$ at $\lambda = 400\,\text{nm}$ for a **(a)** whole detector and **(b)** polar PMT.

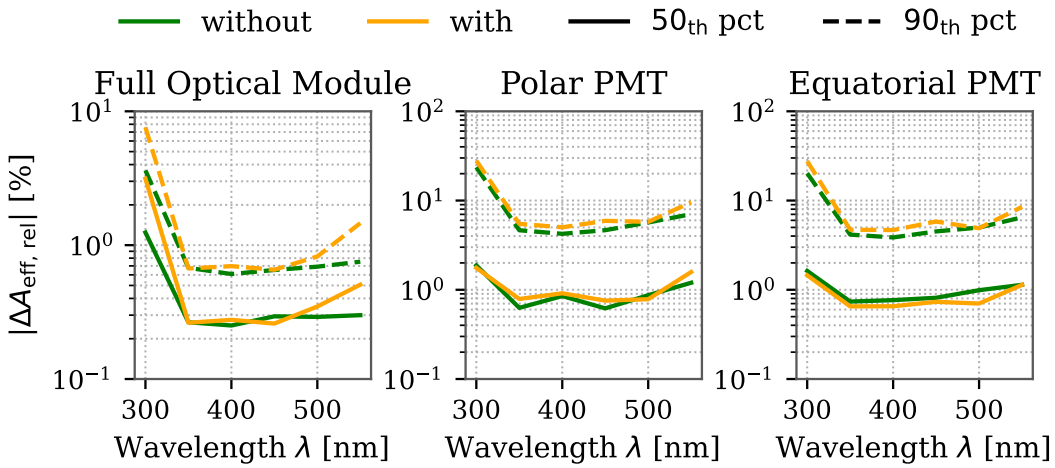


FIGURE 6.1.8: Performance percentiles plotted versus wavelength for mDOM without harness (green) and mDOM with simplified harness (orange). In case of single PMTs, one representative polar and equatorial PMT were analyzed, which are affected by the shadowing of the harness.

6.1.3 Accuracy for mDOM with harness

In section 5.4.1, it was shown that simulating the full harness is not feasible for training the neural network, and therefore a simplified version that reproduces most of the shadowing effect was considered. Given that the harness introduces asymmetries, the MLP branch explained in section 4.2 was used for this task. On the other hand, the analytical approximation does not take the effect of the harness into account and is therefore not discussed in this section.

A comparison between the Geant4 simulation and the neural network with respect to the mean effective area $\overline{A}_{\text{eff}}$ and the effective areas A_{eff} for different directions at a wavelength of $\lambda = 400\,\text{nm}$ are illustrated in fig. 6.1.7 (a) for the whole detector and in (b) for a polar PMT. According to these plots, the patterns and mean effective areas of both simulations have a good agreement.

Moreover, it becomes clear that the neural network can capture the shadowing of the new components as in the upper left part of plot (b).

To quantify the accuracy, the performance percentiles are provided in fig. 6.1.8, overlaid with the differences previously obtained for the model without harness in fig. 6.1.5 as a reference. The results show that the accuracy for an mDOM with harness is comparable to the values reported previously in the absence of the harness, with notable differences only at 300 nm, where the median of the model with the harness is 2 percentage points higher.

6.1.4 Accuracy of a neural Network with quantization

As explained in section 3.5 quantization can minimize the inference time of a neural network in a CPU. However, quantization introduces an approximation due to reduced numerical precision, which may affect accuracy.

It was observed that, in cases where a photon was expected to be detected by a single PMT, the quantized network assigned a non-zero detection probability at the correct PMT index. However, this probability consistently collapsed to 0.5, with the complementary non-detection probability also fixed at 0.5. This behavior can be explained by quantization reducing the differences between layer outputs. When these nearly identical values are passed

through the logsoftmax activation, the resulting probabilities become indistinguishable, leading to the observed 0.5/0.5 split. As a direct consequence, the calculated mean effective areas, without the appropriate scaling, failed to reflect the actual behavior. Furthermore, the logsoftmax activation function is inherently sensitive to small numerical deviations, which can further degrade overall performance. The mean effective areas and scaling factors can be found in the Appendix section A.5. A mollweide projection for the scaled effective area $A_{\text{eff}}(\theta, \varphi)$ is depicted in fig. 6.1.9 together with the effective area provided by Geant4 for a wavelength of $\lambda = 400\,\text{nm}$.
Comparing both plots, the patterns look overall similar, but a clear pixelization—absent in previous results such as fig. 6.1.2—is visible as a result of the quantization approximation. Again, to quantify the accuracy, the performance percentiles for the quantized model, the non-quantized (normal) model and the analytical approximation are collected in fig. 6.1.10.

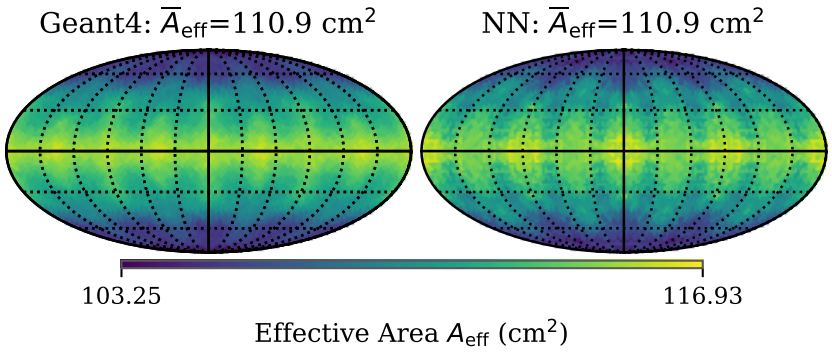


FIGURE 6.1.9: Effective area $A_{\text{eff}}(\theta, \varphi)$ at $\lambda = 400\,\text{nm}$ as a function of incident direction for the entire mDOM. Left: Geant4. Right: Quantized neural network.

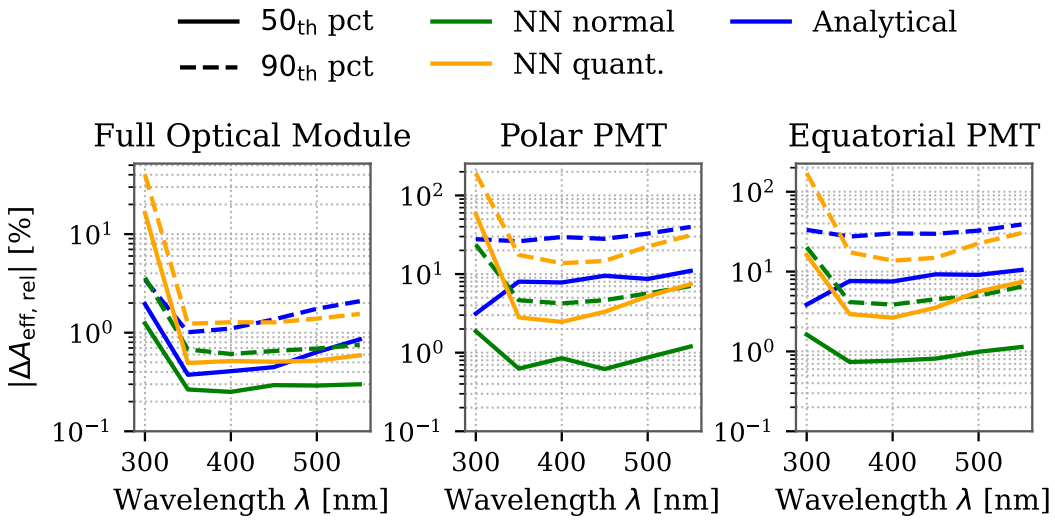


FIGURE 6.1.10: Performance percentiles plotted versus wavelength for the analytical approximation (blue), a neural network without quantization (green) and with quantization (orange).

Compared to the normal neural network, the quantized model shows lower accuracy in all cases, with the largest difference at 300 nm for single polar PMTs (about 70 percentage points). At other wavelengths, the differences remain below 6 percentage points. Regarding the analytical approach, except at 300 nm, the performance of the quantized model is comparable for the entire module and even better for single PMTs, with improvements of 3 to 6 percentage points for the median, demonstrating that the quantization approach can be a more accurate alternative to the analytical method.
In the future, different quantization techniques, such as *quantization aware-training* (QAT) presented in [36, 44], could be investigated to improve the accuracy. In contrast to the current technique, this approach quantizes the model during its training. This allows a minimization of the loss by adjusting the weights to compensate the quantization errors caused by the quantization steps.

6.2 Inference time

The neural networks trained in this thesis are intended to serve as a more accurate alternative to the current analytical approximation used in the IceCube-Upgrade simulations, while at the same time providing fast inference times and a simple deployment suitable for large-scale IceCube-Upgrade simulation production.
The higher accuracy of the neural network has already been demonstrated in section 6.1.1. Therefore, the inference time will be discussed in the following. To this end, several scenarios, such as GPU execution, multi-threaded CPU processing, and quantization techniques, will be compared to the estimated runtimes of the current analytical approximation and Geant4 simulations.
Both the neural network running on the CPU and Geant4 can speed up their inference through *multi-threading*, in which computations are parallelized across different CPU threads. The influence of multithreading on the inference time of the Geant4 simulation, the neural network and the quantized network both running on CPU is depicted in fig. 6.2.1, where the number of threads T was gradually increased from 1 to 10. To obtain the value and the standard error the inferences were performed 1000 times.
As shown in the figure, the inference times significantly decrease with more threads, where significantly reduced runtimes can already be observed at four threads. The runtimes of Geant4 and the neural network without quantization are very close in the most cases. However, by using quantization, the time can be reduced by approximately 66% which is a notable improvement.

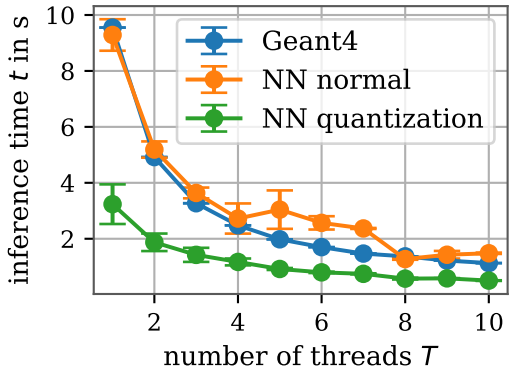


FIGURE 6.2.1: Multithreading on CPU: Inference time t (in s) for processing 100,000 photons in case of Geant4 (blue), the neural network without quantization (orange) and the quantized network (green) as a function of number of threads T.

The table shown in tab. 6.2.1 contains the inference time for several processor configurations and for a small (85 units and filters) and a large neural network (150 units and filters). The times were recorded for 30720 mini-batches, each consisting of 100,000 photons and subsequently averaged. A general behavior observed in the table is that the large model increases the run-time by twice (or even more). This agrees with the increased number of trainable parameters (80,000 vs.

200,000). As already observed in section 6.1.2 the larger model only improves the accuracy by less than 1 percentage point. Therefore, this large increase in inference time is not justified. The focus is therefore placed on the smaller model, where more threads and especially quantization techniques decreases the inference time on a CPU (*Intel Xeon Gold 6140*) substantially. The shortest time on a CPU can be observed for the quantized model with 10 threads, where the time yields $t = (0.4759 \pm 0.0019)$ s. In addition to different CPU configurations, the non-quantized neural network was run on GPUs (*Geforce RTX 4090*) which parallelize matrix-tensor operations across thousands of lightweight threads. Due to this property, the GPU provides the shortest inference time with $t = (0.0412 \pm 0.0016)$ s, which is in the same order as the run-time of the analytical approximation with an estimated time of $t = 0.02$ s [55] on a CPU.

TABLE 6.2.1: Average runtime (in s) for various configurations when processing 20 billion photons across 85 and 150 nodes.

Device and configuration	**85 nodes**	**150 nodes**
Intel Xeon Gold 6140, 1 thread	(8.58 ± 0.74) s	(14.8 ± 2.3) s
Intel Xeon Gold 6140, 10 threads	(1.251 ± 0.040) s	(2.28 ± 0.19) s
Intel Xeon Gold 6140, quantization, 1 thread	(4.09 ± 0.34) s	(7.63 ± 0.69) s
Intel Xeon Gold 6140, quantization, 10 threads	(0.4759 ± 0.0019) s	(0.987 ± 0.020) s
NVIDIA Geforce RTX 4090	(0.0412 ± 0.0016) s	(0.2066 ± 0.0018) s

Overall, the best configuration is to run the smaller neural network on a GPU, which is much more accurate than the analytical approximation while maintaining a similar runtime. If CPUs are preferable due to resource constraints, the quantized model is the fastest alternative, at the cost of accuracy, whereas the non-quantized model on the CPU is about 400 times slower than the analytical approach with 1 thread (60 times slower with 10 threads), and about three times slower than the quantized model. Although the runtime of the non-quantized neural network on the CPU is similar to that of Geant4, the neural network may still be preferable because it is easier to deploy – the saved model can simply be loaded in Python or C++ – whereas incorporating Geant4 into the IceCube-Upgrade simulation chain is more complex and may add additional runtime overhead.

7 Summary & Outlook

The main goal of this thesis was to train and characterize neutral networks that estimate PMT hit probabilities for incident photons in newly developed multi-PMT optical modules (mDOMs) for the upcoming IceCube-Upgrade, with the aim of replacing the simulation currently used in the IceCube Upgrade. The existing simulation assumes a wavelength independent and symmetric relative angular acceptance for each PMT within the mDOM. However, due to wavelength-dependent optical properties of the different materials in the mDOM such as absorption length, refractive index and reflectivity, resulting in asymmetric angular acceptance for PMTs, these assumptions provide an inaccurate representation of the actual detector response. As presented in [5] for the case of the IceCube-Gen2DC-16 optical module, the corresponding neural network represent a strong candidate for this task, given its fast inference time and ability to excel in high-dimensional problems. It combines convolutional and dense layers to capture inherent symmetries of the optical module, namely the equivalence of PMTs of the same type within the module, while also capturing symmetry-breaking features, such as those caused by the harness shadowing. Due to its advantages, the neural network was adapted for the mDOM and trained on detailed Geant4 simulations in two configurations: with and without the harness.

The accuracy of an mDOM without harness showed that the predictions of the neural network are clearly closer to that of the Geant4 simulation than those of the analytical approximation, especially regarding patterns observed in mollweide projections of the effective area. Relative effective-area differences, expressed with respect to the Geant4 mean effective area, in case of the neural network for the full optical module are often less than 1%, except at 300 nm, while in case of the analytical approximation the difference reaches values up to 2%. Generally, the highest deviations can be observed at wavelengths of $\lambda = 300\,\mathrm{nm}$. As explained, this can be attributed to increased photon absorption (by glass and gel) at that wavelength, which yields less informative training data and thus poorer predictions. Nevertheless, the discrepancies are acceptable and, given the low effective area and strong ice absorption at this wavelength, their impact is not expected to be significant. For single PMTs the accuracy of the neural network is one order of magnitude higher than in case of the analytical approximation with a general median difference over the full solid angle of below 1%, which underlines the improvement achieved by the neural network.
In order to determine the best settings for the neural network, the influence of the number of units and filters, the inclusion of the symmetry breaking MLP branch and the amount of training data was examined. The investigations indicate that a model with around 80,000 parameters achieves accuracy comparable to a model with approximately 200,000 parameter, while the former offers much faster inference. Moreover, the additional MLP branch has negligible effect for the mDOM without a harness, and increasing the dataset beyond 10 billion samples is not expected to improve accuracy.
To train the mDOM with harness, it was necessary to simplify the harness by truncating it. This simplification captures most of the shadowing and yields an approximated 1% difference in the mean effective area at 400 nm. The neural network model reproduces the harness shadowing, achieving accuracy comparable to that obtained for an mDOM without a harness.

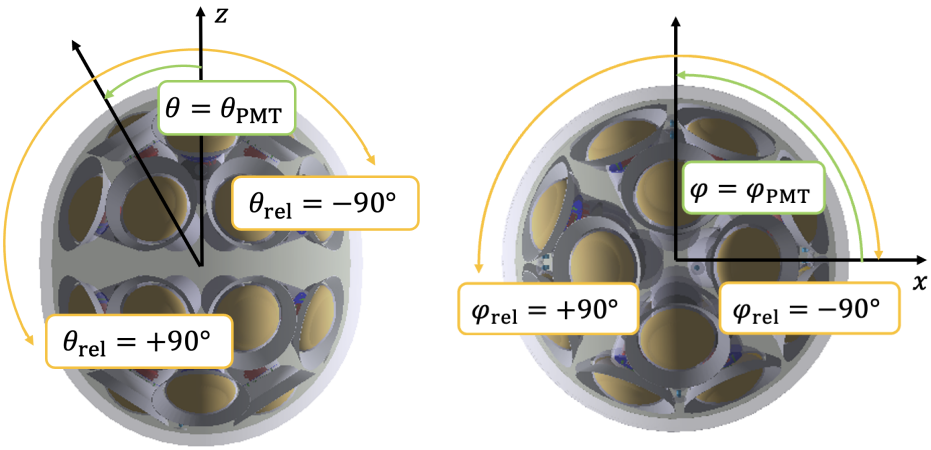
Quantization techniques were applied in this thesis to reduce the neural network's CPU inference time. The quantized model provided a reduced accuracy compared to the neural network without quantization by up to one percentage point for the full module but at the same time a similar or a slightly better accuracy compared to the analytical approximation, specifically with respect to the observed directional effective area patterns. Only for wavelengths of $\lambda = 300\ \mathrm{nm}$ the performance of the quantized model is clearly worse than both simulations, where the median reaches differences up to 10 percentage points. On the other hand, the accuracy of the quantized model for single PMTs is higher by three to six percentage points compared to the analytical approach. The worse performance can be explained by the *logsoftmax* activation, which is sensitive to small numerical deviations caused by quantization errors. Implementing other quantization techniques, such as *quantization aware-training* could result in higher accuracies, which should be evaluated in the future. Nevertheless, the inference time could be reduced by 66% compared to the neural network without quantization running on CPU.
The estimated processing time for 100,000 photons for the analytical approximation is $t = 0.02\ \mathrm{s}$ [55]. The closer running times are for the neural network model on GPU which is $t = (0.0412 \pm 0.0016)\ \mathrm{s}$. On the contrary, the inference times of the model in CPU was $t = (0.4759 \pm 0.0019)\ \mathrm{s}$ in case of the quantized model executed with 10 threads, while the corresponding time of the non-quantized model with 1 thread was $t = (8.58 \pm 0.74)\ \mathrm{s}$. Moreover, regardless of the number of threads, the neural network without quantization in CPU has similar runtimes to the Geant4 simulation. However, independent from the inference time, the neural network might be more preferable than the Geant4 simulation due to its simple deployment in the IceCube simulation.

In the future, the neural networks need to be deployed in IceCube-Upgrade simulations, and the fraction of overhead time contributed by different CPU and GPU configurations relative to the total neutrino or muon simulation time should be estimated. This will provide a clearer picture of the best settings to use. In addition, the impact of hole ice [28] needs to be considered. This ice forms around the drill hole from refrozen water and exhibits higher scattering and absorption. It is particularly important for the mDOM with harness, where the sensitive spherical volume extends well above and below the module. As a result, hole ice can alter the angular acceptance and may introduce non-negligible photon arrival-time-delays due to scattering that was neglected in this work following [5].

A Appendix

A.1 Additional sanity checks for single PMTs

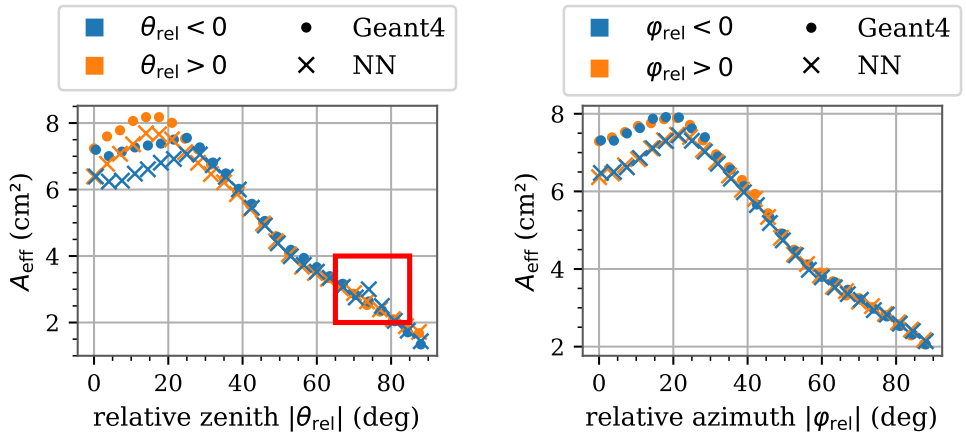
We performed a sanity check to better understand the symmetry of the PMT angular acceptance to photons and how it is modeled by the network. The motivation was to explore whether the PMT-relative inputs could be expressed more compactly using only the opening angle between the photon position vector and the PMT axis, rather than the current cylindrical description, which could result in a shorter inference time, or otherwise to confirm the agreement achieved with the present approach.



(a) (b)

FIGURE A.1.1: Sanity checks for a polar PMT: Measurement of the effective area **a)** in zenith direction $\theta_{\mathrm{rel}} = \theta - \theta_{\mathrm{PMT}}$ for a fixed azimuth $\varphi = \varphi_{\mathrm{PMT}}$ (side view) and **b)** in azimuth direction $\varphi_{\mathrm{rel}} = \varphi - \varphi_{\mathrm{PMT}}$ for a fixed zenith $\theta = \theta_{\mathrm{PMT}}$ (top view).

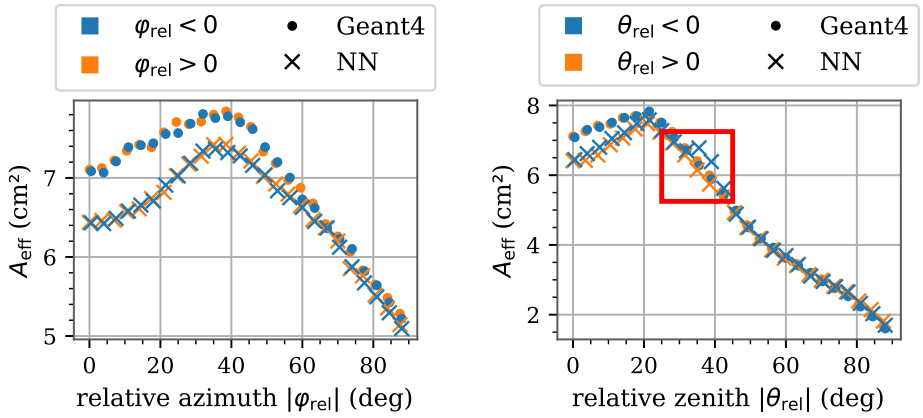
The configuration for this study was the following: First the azimuth angle of the analyzed PMT φ_{PMT} was fixed, while the relative zenith angle $\theta_{\mathrm{rel}} = \theta - \theta_{\mathrm{PMT}}$ between the PMT angle θ_{PMT} and the zenith angle θ of the beam measured from the positive z-axes were changed by θ, as illustrated in fig. A.1.1. For the PMT analyses, only angles between $-90^\circ \leq \theta_{\mathrm{rel}} \leq 90^\circ$ were considered, since the effective area on the opposite side of the PMT is most of the cases zero (some reflections in the pressure vessel can lead to small effective areas). Second, the effective area was measured for the fixed zenith angle of the PMT θ_{PMT} in dependence of the relative azimuth angle $\varphi_{\mathrm{rel}} = \varphi - \varphi_{\mathrm{PMT}}$. For a better evaluation of the symmetry, negative angles θ_{rel} were mirrored to the positive x-axes, so two curves can be compared. This PMT analysis was applied to the Geant4 simulation to examine the PMT symmetry and in addition to the neural network approach, in order to check, if the network was able to learn symmetry properties of PMTs. The resulting plots for an equatorial PMT hit by photons with a wavelength of $\lambda = 550\,\mathrm{nm}$ are depicted in fig. A.1.2, where the rings observed in previous sections can be identified.



(a) (b)

FIGURE A.1.2: (Mirrored) Effective area A_{eff} of an equatorial PMT simulated by the Geant4 simulation (points) and the neural network (crosses) for a wavelength of $\lambda = 550\,\text{nm}$ and **(a)** a fixed azimuth angle $\varphi_{\text{PMT}} = 247.5°$ as a function of the relative zenith $|\theta_{\text{rel}}|$; **(b)** a fixed azimuth angle $\theta_{\text{PMT}} = 72°$ as a function of the relative azimuth $|\varphi_{\text{rel}}|$; the red box highlights a discontinuity in case of the neural network.

In case of the zenith direction, shown in fig. A.1.2 (a), the Geant4 simulation proves that equatorial PMTs are not zenith-symmetric, as the orange and blue dotted curves deviate from each other, especially for small incident angles up to $25°$. This is a result of a lack of alignment between the normal vector of the pressure vessel and the equatorial PMT axis, which consequently means a asymmetric filling of the gel between the PMT and the glass vessel. A similar behavior can be observed for the neural network, which also shows a deviation between the orange and blue crossed curves. This means that the neural network learned the asymmetry in zenith direction. However, the neural network slightly underestimates the effective area for angles up to $25°$ despite scaling with a percentage up to approximately 10% for the normal direction of the PMT. Possible reasons could be the same as explained in section 6.1.1 for the mean effective area shown in fig. 6.1.1. For shorter wavelengths at which the observed ring structure fades, these differences decrease significantly as shown in section A.1. Therefore, in the most cases these differences are negligibly small. Another discrepancy can be observed at an angle of $\theta_{\text{rel}} = -72°$, where the blue crossed curve exhibits a discontinuity. The origin lies in the relative input variables explained in section 4.2.1, specifically in the cosinus of the relative azimuth $\cos(\Delta\varphi_{\text{dir}})$ between photon direction and PMT axis. With the current configuration, this parameter shows a discontinuity around a relative zenith angle of $72°$, which corresponds to a zenith beam angle of $\theta = 0°$. At this position, the parameter takes the value $+1$ for a slightly higher "positive" zenith and -1 for a slightly higher "negative" zenith due to an azimuth shifting of $\Delta\varphi = 180°$.
The effective area in azimuth direction is illustrated in fig. A.1.2 (b) and indicates, that equatorial PMTs are symmetric in azimuth direction (dotted curve). A similar behavior can be observed for the neural network, except for a slight underestimation within the ring with the same deviation as seen before in zenith direction. In contrast to the zenith direction, there is no discontinuity.
The corresponding plots for a polar PMT are illustrated in fig. A.1.3 and prove that polar PMTs are both zenith-symmetric and azimuth-symmetric. Moreover, the discontinuity observed for the equatorial PMT also occurs for polar PMTs in zenith direction. However,



(a) (b)

FIGURE A.1.3: (Mirrored) Effective area A_{eff} of an polar PMT simulated by the Geant4 simulation (points) and the neural network (crosses) for a wavelength of $\lambda = 550\,\text{nm}$ and **(a)** a fixed azimuth angle $\varphi_{\text{PMT}} = 0°$ as a function of the relative zenith $|\theta_{\text{rel}}|$; **(b)** a fixed azimuth angle $\theta_{\text{PMT}} = 33°$ as a function of the relative azimuth $|\varphi_{\text{rel}}|$; the red box highlights a discontinuity in case of the neural network.

the gap appears at a different angle of $\theta_{\text{rel}} = -33°$ due to a different zenith of the PMT at $\theta_{\text{PMT}} = 33°$.
In conclusion, the sanity checks disprove the zenith-symmetry of the mDOM, especially for equatorial PMTs. However, in order to reduce the PMT-relative inputs to the opening angle between the photon vector and the PMT axis, it is necessary that the module is both zenith and azimuth symmetric. Therefore, this approach cannot be applied to the mDOM. Other techniques for a reduction of the inference time are presented in section 6.2. Nevertheless, it can be confirmed that the neural network captures these symmetry properties except for small deviations at small relative angles up to 25° and at beam angles of 0°.

A.2 Explanation of the relative neural network inputs

The neural network uses relative inputs for the symmetric description of the mDOM and absolute input for symmetry breaking features of the mDOM, e.g. the harness. The relative inputs are presented in the following:

Relative z-coordinate Δz_{pos}
One input is the relative z-coordinate of the position Δz_{pos}, which is calculated by the z-coordinate of the PMT $z_{\text{pos}}^{\text{pmt}}$ and the z-coordinate of the photon $z_{\text{pos}}^{\text{photon}}$ with the following formula:

$$\Delta z_{\text{pos}} = z_{\text{pos}}^{\text{pmt}} - z_{\text{pos}}^{\text{photon}} \tag{A.2.1}$$

As shown in fig. A.2.1, the distance between the orange photon and the top PMT is equal to the distance between the blue photon and the bottom PMT, except for the sign. Because the situation is symmetric, the z-sign of the bottom PMT will be reverted. This will be done by multiplying a "1" or a "-1" depending on the upper/bottom binary value of the corresponding PMT.

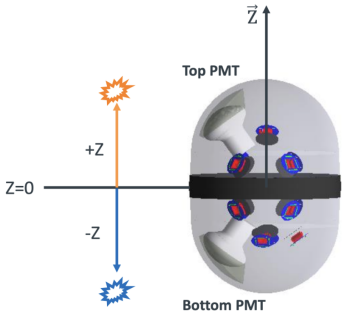


FIGURE A.2.1: Symmetric situation between the upper and bottom part of the module. Taken from [53].

Absolute z-coordinate of photon $z_{\text{dir}}^{\text{photon}}$
The first input described the *position* in z orientation. However, a desciption for the *direction* in z orientation is missing. Due to the fact, that the PMTs might not have a symmetry in zenith direction (angle measured from the z-axis) because of the oval form of the mDOM, a relative description does not make sense. Instead, the absolute z-coordinate of the photon direction $z_{\text{dir}}^{\text{photon}}$ is used, which is, according to trigonometry, equal to:

$$z_{\text{dir}}^{\text{photon}} = \cos \vartheta_{\text{dir}}^{\text{photon}} \tag{A.2.2}$$

As illustrated in fig. A.2.2, there is the same symmetry as in the previous input. Therefore, the bottom PMTs are reverted analogously.

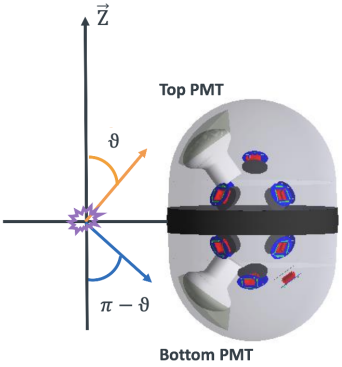


FIGURE A.2.2: Symmetry regarding to the absolute zenith of the photon direction. Modified and taken from [53].

Relative azimuth $\Delta\varphi_{\text{pos}}$
After the description of photons in z orientation, a description in the x-y-plane is needed. If the mDOM is viewed from the z-direction, the module seems to be a circle, so the mDOM has a azimuth symmetry. This allows a relative description of the photon position to each PMT axis, which has the following form:

$$\Delta\varphi_{\text{pos}} = \varphi_{\text{pos}}^{\text{pmt}} - \varphi_{\text{pos}}^{\text{photon}} \tag{A.2.3}$$

As shown in fig. A.2.3, the relative positions of the blue and orange photons are symmetric. Thus, both photons should be associated with the same value. This can be achieved by using the cosinus of the azimuth difference

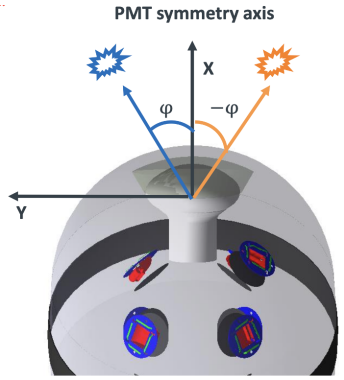


FIGURE A.2.3: Symmetry regarding to the relative azimuth angle of the positions. Modified and taken from [53].

$\cos \Delta\varphi_{\text{pos}}$.

Relative azimuth $\Delta\varphi_{\text{dir}}$ + convergence check Δy_{pos}
In addition to the position, the direction in the x-y-plane has to be characterized. The symmetry is analogously described to the relative azimuth of the position through $\cos \Delta\varphi_{\text{dir}}$. However, as depicted in fig. A.2.4, the direction is a vector, which could be located everywhere. This leads to the issue, that the same directions 1 and 3 are one time directed to the PMT and one time not. For the directions 2 and 4 it is similar. In order to find out, whether the photon points to a PMT, the difference in the y position between photon and PMT will be analyzed for two different time steps. First, the difference will be calculated as before for other parameters, according to this equation:

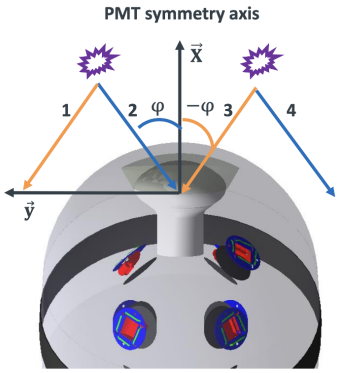


FIGURE A.2.4: Symmetry regarding to the relative azimuth angle of the direction. Modified and taken from [53].

$$\Delta y_{\text{pos}}(t_0) = y_{\text{pos}}^{\text{pmt}} - y_{\text{pos}}^{\text{photon}} \tag{A.2.4}$$

After that, this distance will be computed for a later time t_1, which can be imitated by a step (such as 0.1) of the photon in its direction $y_{\text{dir}}^{\text{photon}}$. According to this convergence strategy, the difference can be calculated as follows:

$$\Delta y_{\text{pos}}(t_1) = y_{\text{pos}}^{\text{pmt}} - (y_{\text{pos}}^{\text{photon}} + 0.1 \cdot y_{\text{dir}}^{\text{photon}}) \tag{A.2.5}$$

Consequently, the convergence can be checked by the inequation $|\Delta y_{\text{pos}}(t_0)| > |\Delta y_{\text{pos}}(t_1)|$, which returns a binary value as an input for the network.

A.3 Effective area of an equatorial PMT

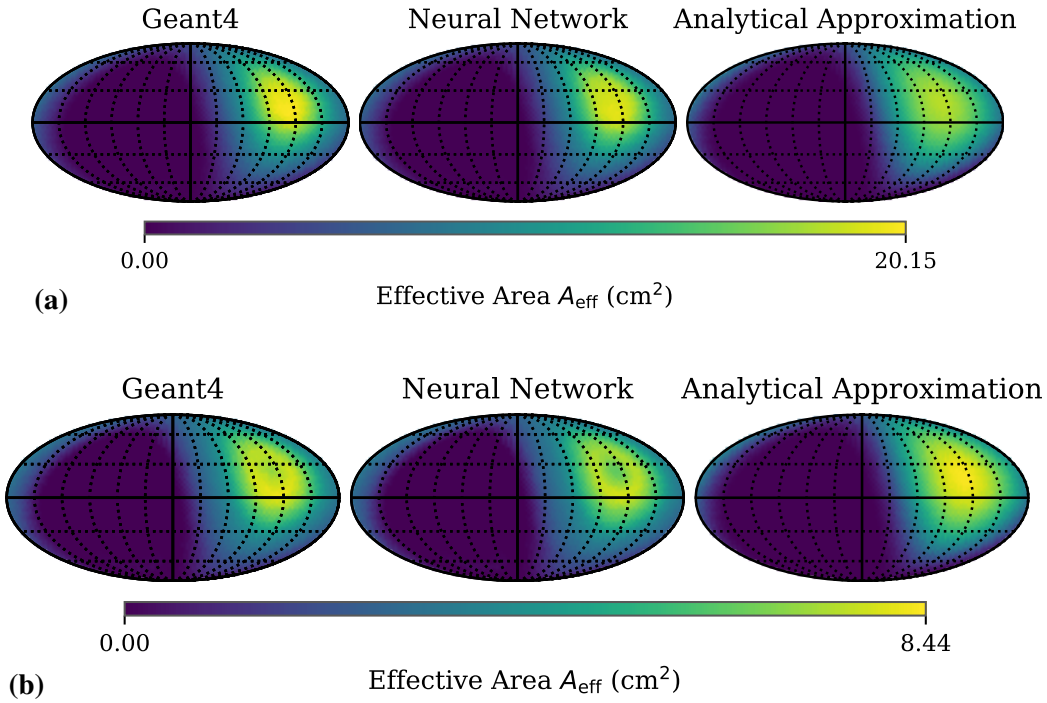


FIGURE A.3.1: Effective area A_{eff} for a **single equatorial PMT** at different directions and wavelengths: **(a)** $\lambda = 400\,\text{nm}$, **(b)** $\lambda = 550\,\text{nm}$.

A.4 Distribution of effective area difference between Geant4/NN

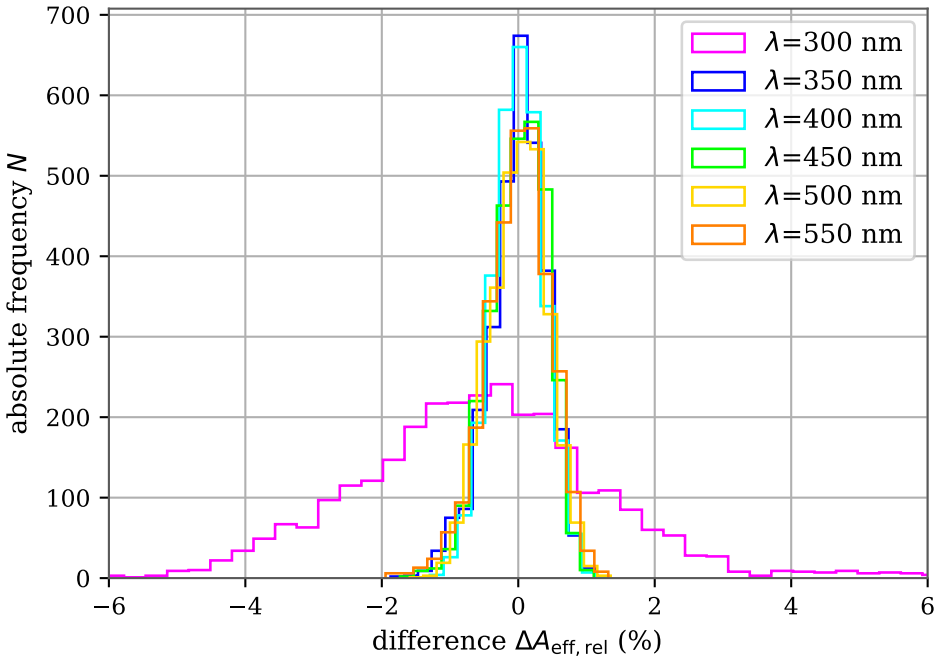
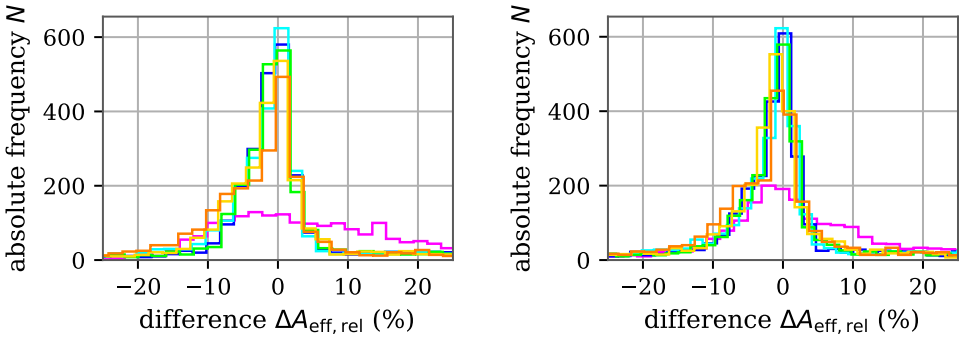


FIGURE A.4.1: Distribution of the relative effective area difference $\Delta A_{\text{eff,rel}}$ (in %) for the whole optical module.



(a) (b)

FIGURE A.4.2: Distribution of the relative effective area difference $\Delta A_{\text{eff,rel}}$ (in %) for **a)** a polar PMT and an **b)** equatorial PMT. Colors are explained in fig. A.4.1.

A.5 Mean effective area of the neural network with quantization

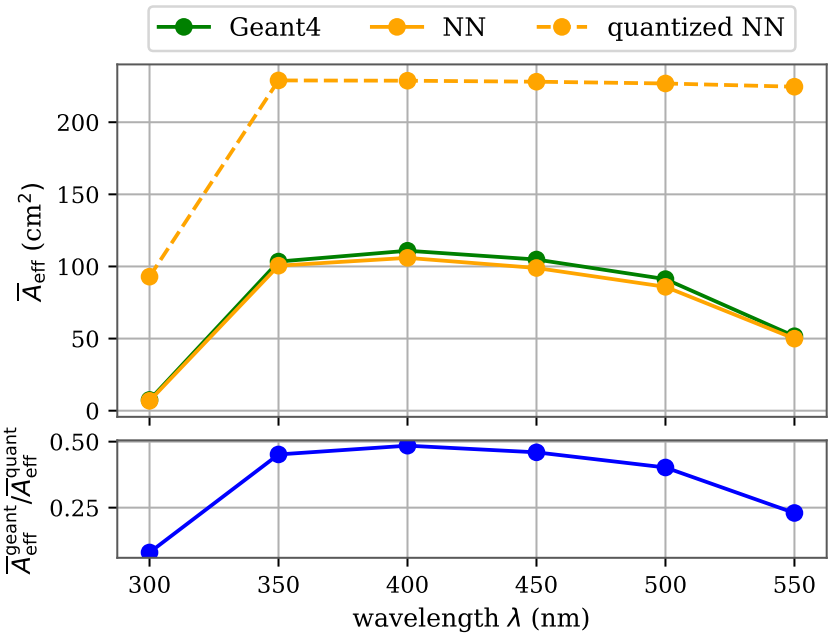


FIGURE A.5.1: mean effective area $\overline{A}_{\text{eff}}$ for different simulations and wavelengths together with the ratio $a(\lambda) = \overline{A}_{\text{eff}}^{\text{geant}}/\overline{A}_{\text{eff}}^{\text{quant}}$, which constitutes the scaling factor.

Bibliography

[1] IceCube Collaboration. *IceCube*. Accessed: 2025-07-17. URL: https://icecube.wisc.edu/science/icecube/.

[2] Aya Ishihara. *The IceCube Upgrade – Design and Science Goals*. 2019. arXiv: 1908.09441 [astro-ph.HE]. URL: https://arxiv.org/abs/1908.09441.

[3] Lew Classen et al. "A multi-PMT Optical Module for the IceCube Upgrade". In: *Proceedings of 36th International Cosmic Ray Conference — PoS(ICRC2019)*. Madison, WI, U.S.A.: Sissa Medialab, July 2019. DOI: 10.22323/1.358.0855.

[4] Dmitry Chirkin. private communication.

[5] Francisco Javier Vara Carbonell and Jonas Selter. *Machine Learning Tools for the IceCube-Gen2 Optical Array*. 2025. arXiv: 2507.07844 [astro-ph.IM]. URL: https://arxiv.org/abs/2507.07844.

[6] M G Aartsen et al. "IceCube-Gen2: the window to the extreme Universe". In: *Journal of Physics G: Nuclear and Particle Physics* 48.6 (Apr. 2021), p. 060501. ISSN: 1361-6471. DOI: 10.1088/1361-6471/abbd48. URL: http://dx.doi.org/10.1088/1361-6471/abbd48.

[7] IceCube Collaboration. *OMSim*. URL: https://github.com/icecube/OMSim (visited on 08/05/2025).

[8] IceCube Collaboration. *IceCube Overview*. Accessed: 2025-07-23. URL: https://icecube.wisc.edu/about-us/overview/.

[9] The KM3NeT Collaboration, P. Bagley, et al. *KM3NeT: Conceptual Design Report for a Deep-Sea Research Infrastructure Incorporating a Very Large Volume Neutrino Telescope in the Mediterranean Sea*. Accessed: July 23, 2025. 2008. URL: https://www.km3net.org/wp-content/uploads/2015/07/CDR-KM3NeT.pdf.

[10] Yoichiro Suzuki. "The super-kamiokande experiment". In: *Eur. Phys. J. C Part. Fields* 79.4 (Apr. 2019).

[11] The Nobel Foundation. *The Nobel Prize in Physics 2015 – Popular Science Background*. https://www.nobelprize.org/uploads/2018/06/popular-physicsprize2015-1.pdf. Accessed: 2025-08-24. 2015.

[12] KATRIN Collaboration et al. "Direct neutrino-mass measurement based on 259 days of KATRIN data". In: *Science* 388.6743 (Apr. 2025), pp. 180–185. DOI: 10.1126/science.adq9592.

[13] KATRIN Collaboration. "The design, construction, and commissioning of the KATRIN experiment". In: *Journal of Instrumentation* 16.T08015 (Aug. 2021). DOI: 10.1088/1748-0221/16/08/T08015. arXiv: 2103.04755 [physics.ins-det]. URL: https://iopscience.iop.org/article/10.1088/1748-0221/16/08/T08015.

[14] Wolfgang Pauli. *Pauli letter collection: Letter to Lise Meitner*. Typed copy. CERN Document Server. URL: https://cds.cern.ch/record/83282.

[15] IceCube Collaboration. *IceCube Neutrino Observatory – Research Highlights*. Accessed: 2025-07-23. URL: https://icecube.wisc.edu/science/research/.

[16] R. Abbasi et al. "Measurement of the high-energy all-flavor neutrino-nucleon cross section with IceCube". In: *Phys. Rev. D.* 104.2 (July 2021). DOI: 10.1103/physrevd.104.022001.

[17] J. A. Formaggio and G. P. Zeller. "From eV to EeV: Neutrino cross sections across energy scales". In: *Reviews of Modern Physics* 84.3 (Sept. 2012), 1307–1341. ISSN: 1539-0756. DOI: 10.1103/revmodphys.84.1307. arXiv: 1305.7513.

[18] M. A. Unland Elorrieta. *Development, simulation, and characterisation of a novel multi-PMT optical module for IceCube Upgrade with emphasis on detailed understanding of photomultiplier performance parameters*. 2023. DOI: 10.5281/zenodo.8121321. URL: https://doi.org/10.5281/zenodo.8121321.

[19] R. E. Jennings. "Čerenkov Radiation". In: *Science Progress (1933-)* 50.199 (1962). Accessed: 2025-07-23, pp. 364–375. ISSN: 0036-8504. URL: http://www.jstor.org/stable/43425324.

[20] C. Spiering. *Neutrinoastronomie. Blick in verborgene Welten*. Berlin, Heidelberg: Springer-Verlag GmbH, 2021, pp. 31–36. ISBN: 978-3-662-63294-9.

[21] Mark Aartsen et al. "Neutrino emission from the direction of the blazar TXS 0506+056 prior to the IceCube-170922A alert". In: *Science* 361.6398 (July 2018), 147–151. ISSN: 1095-9203. DOI: 10.1126/science.aat2890. URL: http://dx.doi.org/10.1126/science.aat2890.

[22] R. Abbasi et al. "Evidence for neutrino emission from the nearby active galaxy NGC 1068". In: *Science* 378.6619 (Nov. 2022), 538–543. ISSN: 1095-9203. DOI: 10.1126/science.abg3395. URL: http://dx.doi.org/10.1126/science.abg3395.

[23] R. Abbasi et al. "Observation of high-energy neutrinos from the Galactic plane". In: *Science* 380.6652 (June 2023), 1338–1343. ISSN: 1095-9203. DOI: 10.1126/science.adc9818. URL: http://dx.doi.org/10.1126/science.adc9818.

[24] E. Andres. "The AMANDA neutrino telescope: principle of operation and first results". In: *Astroparticle Physics* 13.1 (Mar. 2000), pp. 1–20. ISSN: 0927-6505. DOI: 10.1016/S0927-6505(99)00092-4. URL: http://dx.doi.org/10.1016/S0927-6505(99)00092-4.

[25] M.G. Aartsen et al. "The IceCube Neutrino Observatory: instrumentation and online systems". In: *Journal of Instrumentation* 12.03 (Mar. 2017), P03012–P03012. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/03/p03012. arXiv: 1612.05093 [astro-ph.IM].

[26] R Abbasi et al. "Calibration and characterization of the IceCube photomultiplier tube". In: *Nucl. Instrum. Methods Phys. Res. A* 618.1-3 (June 2010), pp. 139–152. DOI: 10.1016/j.nima.2010.03.102.

[27] R Abbasi et al. "The design and performance of IceCube DeepCore". In: *Astropart. Phys.* 35.10 (May 2012), pp. 615–624. DOI: 10.1016/j.astropartphys.2012.01.004.

[28] M.G. Aartsen et al. "Measurement of South Pole ice transparency with the IceCube LED calibration system". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 711 (May 2013), 73–89. ISSN: 0168-9002. DOI: 10.1016/j.nima.2013.01.054. URL: http://dx.doi.org/10.1016/j.nima.2013.01.054.

[29] IceCube Collaboration, R. Abbasi, et al. "IceTop: The surface component of IceCube". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 700 (2013), pp. 188–220. DOI: 10.1016/j.nima.2012.10.067.

[30] Philipp Eller et al. *Sensitivity of the IceCube Upgrade to Atmospheric Neutrino Oscillations*. 2023. arXiv: 2307.15295 [astro-ph.HE]. URL: https://arxiv.org/abs/2307.15295.

[31] IceCube Collaboration. *Successful testing of over 10,000 photomultiplier tubes for IceCube Upgrade digital optical modules*. Accessed: 2025-07-25. 2024. URL: https://icecube.wisc.edu/news/research/2024/05/successful-testing-of-over-10000-photomultiplier-tubes-for-icecube-upgrade-digital-optical-modules/.

[32] IceCube Collaboration. *D-Egg: a Dual PMT Optical Module for the IceCube Upgrade*. Accessed: 2025-09-06. 2023. URL: https://pos.sissa.it/444/1082/pdf.

[33] François Chollet. *Deep Learning with Python*. English. New York, NY: Manning Publications, Oct. 2017.

[34] Thanasis Kotsiopoulos et al. "Machine Learning and Deep Learning in smart manufacturing: The Smart Grid paradigm". In: *Comput. Sci. Rev.* 40.100341 (May 2021), p. 100341. DOI: 10.1016/j.cosrev.2020.100341.

[35] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

[36] Suraj Subramanian, Mark Saroufim, Jerry Zhang. *Practical Quantization in PyTorch*. Accessed: 2025-08-05. URL: https://pytorch.org/blog/quantization-in-practice/.

[37] Benoit Jacob et al. *Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference*. 2017. arXiv: 1712.05877 [cs.LG]. URL: https://arxiv.org/abs/1712.05877.

[38] Francisco Javier Vara Carbonell. *OMNNSim*. Accessed: 2025-09-04. URL: https://github.com/jvaracarbonell/OMNNSim/tree/main.

[39] Pytorch Foundation. *Pytorch*. Accessed: 2025-09-10. URL: https://pytorch.org/.

[40] PyTorch Contributors. *LogSoftmax*. Accessed: 2025-08-04. URL: https://docs.pytorch.org/docs/stable/generated/torch.nn.LogSoftmax.html.

[41] PyTorch Contributors. *KLDivLoss*. Accessed: 2025-08-04. URL: https://docs.pytorch.org/docs/stable/generated/torch.nn.KLDivLoss.html.

[42] PyTorch Contributors. *MSELoss*. Accessed: 2025-08-04. URL: https://docs.pytorch.org/docs/stable/generated/torch.nn.MSELoss.html.

[43] PyTorch Contributors. *AdamW*. Accessed: 2025-08-04. URL: https://docs.pytorch.org/docs/stable/generated/torch.optim.AdamW.html.

[44] PyTorch Contributors. *Quantization*. Accessed: 2025-08-05. URL: https://docs.pytorch.org/docs/stable/quantization.html#post-training-static-quantization.

[45] Geant Collaboration. *Book For Application Developers*. 2021. URL: https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/fo/BookForApplicationDevelopers.pdf (visited on 08/05/2025).

[46] L. Classen. "The mDOM - a multi-PMT digital optical module for the IceCube-Gen2 neutrino telescope". Doktorarbeit. Feb. 2017. URL: https://www.uni-muenster.de/imperia/md/content/physik_kp/agkappes/abschlussarbeiten/doktorarbeiten/1702-phd_lclassen.pdf.

[47] Björn Herold. *Simulation and Measurement of Optical Background in the Deep Sea Using a Multi-PMT Optical Module*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). 2017.

[48] Moritz Schlechtriem. *Simulationsstudien zum Einfluss der Modulaufhängung auf die Photon-Sensitivität des mDOM-Sensors im IceCube Upgrade*. 2021. URL: https://www.uni-muenster.de/imperia/md/content/physik_kp/agkappes/abschlussarbeiten/bachelorarbeiten/ba_moritz_schlechtriem.pdf.

[49] Nicolai Krybus. *Thin Layer Optics in Photomultipliers: A Geant4 Simulation and Comparison to Ellipsometric Measurements*. 2024. URL: https://www.uni-muenster.de/imperia/md/content/physik_kp/agkappes/abschlussarbeiten/masterarbeiten/ma_krybus.pdf.

[50] Lew Classen. "The mDOM - a multi-PMT digital optical module for the IceCube-Gen2 neutrino telescope". PhD thesis. Feb. 2017. URL: https://www.uni-muenster.de/imperia/md/content/physik_kp/agkappes/abschlussarbeiten/doktorarbeiten/1702-phd_lclassen.pdf.

[51] Healpy developers. *Healpy documentation site*. URL: https://healpy.readthedocs.io/en/latest/ (visited on 08/07/2025).

[52] D. Chirkin. *Photon Propagation Code (PPC)*. http://icecube.wisc.edu/~dima/work/WISC/ppc.

[53] Francisco Javier Vara Carbonell. private communication.

[54] John David Jackson. *Classical Electrodynamics*. 3rd. New York: Wiley, 1998.

[55] Yukiho Kobayashi. private communication. Chiba University.

Acknowledgement

At this point, I would like to thank all the people who supported me in this work.
I would like to thank:

- Prof. Dr. Alexander Kappes for welcoming me into his friendly research group and giving me the opportunity to write this bachelor thesis.
- Prof. Dr. Anton Andronic who kindly took the time to perform the second review of this thesis.
- especially Javi who was always there to support me with questions and continually assisted me with any issues.
- Javi and Markus who generously took the time to read and correct my thesis.
- the whole group for the warm welcome and the supportive and fun atmosphere.

Declaration of Academic Integrity

I hereby confirm that this thesis on “Training and application of a neural network to calculate PMT hit probabilities for incident photons in mDOMs for IceCube Upgrade” is solely my own work and that I have used no sources or aids other than the ones stated. All passages in my thesis for which other sources, including electronic media, have been used, be it direct quotes or content references, have been acknowledged as such and the sources cited.

Münster, 13th September 2025 _____

I agree to have my thesis checked in order to rule out potential similarities with other works and to have my thesis stored in a database for this purpose.

Münster, 13th September 2025 _____