

# Search for Heavy Neutral Leptons with IceCube DeepCore

## **Dissertation**

zur Erlangung des akademischen Grades  
doctor rerum naturalium  
(Dr. rer. nat.)

im Fach: Physik  
Spezialisierung: Experimentalphysik

eingereicht an der  
Mathematisch-Naturwissenschaftlichen Fakultät  
der Humboldt-Universität zu Berlin

von

**Leander Fischer M. Sc.**  
geboren am 24. Oktober 1992  
in Heidelberg

Präsidentin der Humboldt-Universität zu Berlin  
Prof. Dr. Julia von Blumenthal

Dekanin der Mathematisch-Naturwissenschaftlichen Fakultät  
Prof. Dr. Caren Tischendorf

**Copyright Notice**

This book is released into the public domain using the CC-BY-4.0 code.

To view a copy of the CC-BY-4.0 code, visit:

<https://creativecommons.org/licenses/by/4.0/>

**Colophon**

This document was typeset with the help of KOMA-Script and L<sup>A</sup>T<sub>E</sub>X using the open-source kaobook template class.

The source code of this thesis is available at:

[https://github.com/LeanderFischer/phd\\_thesis](https://github.com/LeanderFischer/phd_thesis)

## **Zusammenfassung**

Zusammenfassung ...

## **Abstract**

Abstract ...



# Todo list

Make my own DC string positions/distances plot . . . . .	2
fix ic/dc string positions plot caption . . . . .	2
fix caption for simplistic sampling distris . . . . .	3
fix caption for realistic sampling distris . . . . .	4
Select interesting plots (energy length) and move the others to the appendix? . . . . .	4
Fix caption for simplified gen distris . . . . .	4
Fix caption for realistic gen distris . . . . .	4
Re-write the following parts.. . . .	8
add varied total cross-section for a few background HNL events . . . . .	8
Add comparions of SM cross-sections between NuXSSplMkr and genie . . . . .	8
correct information about filenumbers, number of events and how many that were at Gen . . . . .	10
add information about the matter profile used . . . . .	15
add information about the oscillation probability calculation and the software used for it . . . . .	15
get correct final level rates from my pipeline(s) . . . . .	15
add rate and Poisson error for HNL samples . . . . .	15
maybe just pick one mixing? . . . . .	15
add 3D expectation and/or $S/\sqrt{B}$ plots . . . . .	16
Do I want/need to include the description of the KDE muon estimation? . . . . .	16
Add table with all systematic uncertainties used in this analysis (in the analysis chapter). . . . .	16
add final level effects of varying the axial mass parameters (or example of one) . . . . .	16
add final level effects of varying the DIS parameter (or example of one) . . . . .	16
Do I want additional plots for this (fit diff, LLH distr, minim. stats, param. fits)? . . . . .	17
Add bin-wise TS distribution? Add 3D TS maps? . . . . .	17



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Contents</b>	<b>vii</b>
<b>1 Heavy Neutral Lepton Signal Simulation</b>	<b>1</b>
1.1 Model Independent Simulation . . . . .	1
1.1.1 Generator Functions . . . . .	1
1.1.2 Simplistic Sets . . . . .	2
1.1.3 Realistic Set . . . . .	3
1.1.4 Generation Level Distributions . . . . .	4
1.2 Model Dependent Simulation . . . . .	7
1.2.1 Custom LeptonInjector . . . . .	7
1.2.2 Sampling Distributions . . . . .	10
1.2.3 Weighting Scheme . . . . .	11
1.2.4 Generation Level Distributions . . . . .	12
<b>2 Detecting Low Energetic Double Cascades</b>	<b>13</b>
2.1 Reconstruction . . . . .	13
2.1.1 Table-Based Minimum Likelihood Algorithms . . . . .	13
2.1.2 Double Cascade Hypothesis . . . . .	13
2.1.3 Modification to Low Energy Events . . . . .	13
2.2 Cross Checks . . . . .	13
2.2.1 Simplistic Sets . . . . .	13
2.3 Performance . . . . .	14
2.3.1 Energy/Decay Length Resolution . . . . .	14
2.3.2 Double Cascade Classification . . . . .	14
<b>3 Search for an Excess of Heavy Neutral Lepton Events</b>	<b>15</b>
3.1 Final Level Sample . . . . .	15
3.1.1 Expected Rates/Events . . . . .	15
3.1.2 Analysis Binning . . . . .	16
3.2 Statistical Analysis . . . . .	16
3.2.1 Test Statistic . . . . .	16
3.2.2 Systematic Uncertainties . . . . .	16
3.3 Analysis Checks . . . . .	17
3.3.1 Minimization Robustness . . . . .	17
3.3.2 Ensemble Tests . . . . .	17
3.4 Results . . . . .	18
3.4.1 Best Fit Parameters . . . . .	18
3.4.2 Upper Limits . . . . .	18
3.4.3 Post-Fit Data/MC Agreement . . . . .	18
3.4.4 Likelihood Coverage . . . . .	18
<b>APPENDIX</b>	<b>19</b>
<b>A Analysis Checks</b>	<b>21</b>
A.0.1 Minimization Robustness . . . . .	21
A.0.2 Ensemble Tests . . . . .	21
<b>Bibliography</b>	<b>23</b>





# List of Figures

1.1	xx . . . . .	2
1.2	Simplistic set . . . . .	5
1.3	Realistic set . . . . .	6
1.4	HNL branching ratios . . . . .	7
1.5	Custom HNL total cross-sections for the four target masses compared to the total ( $\nu_\tau/\bar{\nu}_\tau$ NC) cross-section used for SM neutrino simulation production with GENIE. . . . .	9
3.1	Asimov inject/recover test (0.6 GeV) . . . . .	18
3.2	Pseudo-data trials TS distribution (0.6 GeV) . . . . .	18
A.1	Asimov inject/recover test (0.3 GeV, 1.0 GeV) . . . . .	21
A.2	Pseudo-data trials TS distribution (0.3 GeV, 1.0 GeV) . . . . .	21



# List of Tables

1.1	xx . . . . .	3
1.2	xx . . . . .	4
1.3	xx . . . . .	9
1.4	xx . . . . .	10
3.1	Final level event/rate expectation . . . . .	15
3.2	Analysis binning . . . . .	16
3.3	Staged minimization settings . . . . .	17



# Heavy Neutral Lepton Signal Simulation

# 1

After the SM simulation generation and the default low energy event selection and processing chain were introduced in the previous Chapter ??, the focus will now be on the central part of this thesis - the HNL signal simulation. Since this is the first attempt of performing a search for HNLs with IceCube DeepCore, there was no prior knowledge of the expected performance nor the event expectation, and the simulation had to be developed from scratch. Two avenues of simulation generation were pursued in parallel; a collection of model independent simulation sets was realized and is explained in Section 1.1 and the physically accurate model dependent simulation is described in Section 1.2.

1.1	Model Independent Simulation	1
1.2	Model Dependent Simulation	7

## 1.1 Model Independent Simulation

To investigate the potential of IceCube to detect HNLs by identifying the unique double cascade morphology explained in Section ??, it is very valuable to have simulation sets where the double cascade kinematics can be controlled directly. In a realistic model the decay kinematics and the absolute event expectation all depend on the specific model parameters chosen (see Section ??). To decouple the simulation from a specific model, a model independent double cascade generator was developed and will be explained in the following sections. Based on the generator a few simulation sets were produced to investigate the performance of IceCube DeepCore to detect low energetic double cascades, dependent on their properties, which will be discussed in Chapter 2.

### 1.1.1 Generator Functions

In order to produce the model independent simulation sets a series of generator functions was implemented in PYTHON [1]. The collection of functions can be found in [this public repository](#). A few independent functions are needed to perform the sampling based on a random variable between 0 and 1 as input. There is a simple function to return a random sign (+1/−1) and two functions to sample from a power law and an exponential distribution. The inputs are the wanted sampling range and the power law index or the exponential decay constant, respectively. They both apply the inverse transformation method.

[1]: Van Rossum et al. (2009), *Python 3 Reference Manual*

Additionally, there are some functions that are IceCube specific. Two functions are implemented to transform a direction in zenith/azimuth angles to a direction vector and vice versa. There is a function to create an EM cascade particle from a position, direction, energy, and time and another to produce an arbitrary list of EM cascades, with the previous function, given the list of input parameters, and then adding it to the current IceCube event. Based on these, any specific simulation set can be produced by choosing the sampling distributions and number of cascades

to be placed in each event and then calling the generator functions with the input parameters based on these sampling distributions.

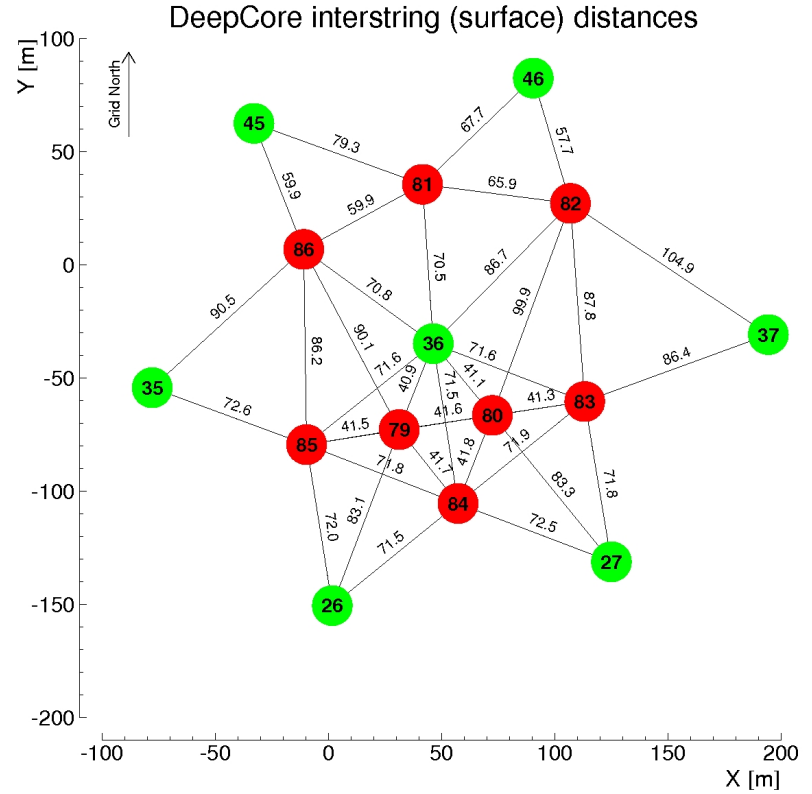
#### IceCube software framework

The functions described above are based on the (public) `icetray` software project and the EM particles are defined as type `I3Particle`, while the object to store the MC particles is called `I3MCTree` and each IceCube event information is in on `I3Frame` object.

### 1.1.2 Simplistic Sets

To test the implemented generator functions and investigate some idealistic double cascade event scenarios, two sets are produced for straight up-going events that are centered on a string and horizontal events located inside DeepCore.

Make my own DC string positions/distances plot



**Figure 1.1:** Horizontal positions and distances between DeepCore strings. Red strings are instrumented more densely (vertically) and partially with higher quantum efficiency (HQE) DOMs.

fix ic/dc string positions plot caption

The first set is used to investigate one of the best possible cases to detect a double cascade, where both cascades are placed on a DeepCore string (namely string 81) and the directions are directly up-going. The horizontal positions and distances of all DeepCore fiducial volume strings are shown in Figure 1.1 and string 81 is at a medium distance of  $\sim 60$  m to its neighboring strings. As already mentioned in Section ??, DeepCore strings have higher quantum efficiency DOMs and a denser vertical spacing. The  $x, y$  position is fixed to the center of string 81 while the  $z$  position of each cascade is sampled uniformly along the strings  $z$  elongation and the energies are sampled uniformly between

Set	Variable	Distribution	Range/Value
Up-going			
	energy	uniform	0.0 GeV to 60.0 GeV
	zenith	fixed	180.0°
	azimuth	fixed	0.0°
	$x, y$ position	fixed	(41.6, 35.49) m
	$z$ position	uniform	-480.0 m to -180.0 m
Horizontal			
	energy	uniform	0.0 GeV to 60.0 GeV
	zenith	fixed	90.0°
	azimuth	uniform	0.0° to 360.0°
	$x, y$ position	uniform (circle)	$c=(46.29, -34.88)$ m, $r=150.0$ m
	$z$ position	fixed	-330.0 m

**Table 1.1:** Sampling distributions of up-going, string 81 centered and horizontal simulation generation.

0.0 GeV and 60.0 GeV. The specific sampling distributions/values for the cascades are listed in Table 1.1. The order of the cascades is chosen such that the lower one is first ( $t_0 = 0.0$  ns) and the upper one is second ( $t_1 = L/c$ ), assuming the speed of light  $c$  as speed of the heavy mass state, traveling between the two cascades.

The second set is used to investigate the effect of the tilt of the double cascade and the reconstruction performance for horizontal events. The cascades are placed uniformly on a circle centered in DeepCore. The direction is always horizontal and azimuth is defined by the connecting vector of both cascade positions. The energies are again sampled uniformly between 0.0 GeV and 60.0 GeV and the detailed sampling distributions/values are also listed in Table 1.1.

fix caption for simplistic sampling distris

### 1.1.3 Realistic Set

To thoroughly investigate the potential of IceCube DeepCore to detect double cascade events, a more realistic simulation set is produced that aims to be as close as possible to the expected signal simulation explained in Section 1.2, while still allowing some freedom to control the double cascade kinematics. For this purpose the total energy is sampled from an  $E^{-2}$  power law, mimicking the energy spectrum of the primary neutrinos as stated in Section ?? . Although in the realistic process described in Section 1.2 the energy is distributed in a more complex way into the two cascades and secondary particles, it is a good approximation to simply divide the total energy into two parts. This is done by randomly assigning a fraction between 0 % and 100 % to one cascade and the remaining part to the other cascade. In this way the whole sample covers various cases of energy distributions between the two cascades. To efficiently generate events in a way that produces distributions similar to what would be observed with DeepCore, one of the cascade positions is sampled inside the DeepCore volume by choosing its coordinates randomly on a cylinder that is centered in DeepCore. This is similar to a trigger condition of one cascade always being inside the DeepCore fiducial volume. By choosing the direction of the event sampling zenith and azimuth uniformly between 70° and 180° and 0° and 360°, respectively, the position of the other cascade can be inferred for a given decay length. The length is sampled from an exponential distribution, which would be expected for the decaying heavy mass state. Based on the direction and the decay length, the position of the other cascade is found, assuming a

Table 1.2: xx

Variable	Distribution	Range/Value
energy (total)	power law $E^{-2}$	1 GeV to 1000 GeV
decay length	exponential $e^{-0.01L}$	0 m to 1000 m
zenith	uniform	70° to 180°
azimuth	uniform	0° to 360°
$x, y$ (one cascade)	uniform (circle)	$c=(46.29, -34.88)$ m, $r=150$ m
$z$ (one cascade)	uniform	-480.0 m to -180.0 m

travel speed of  $c$  and randomly choosing whether the cascade position that was sampled is the first cascade or the second and then assigning the other cascade position accordingly. The sampling distributions/values are listed in Table 1.4.

fix caption for realistic sampling distris

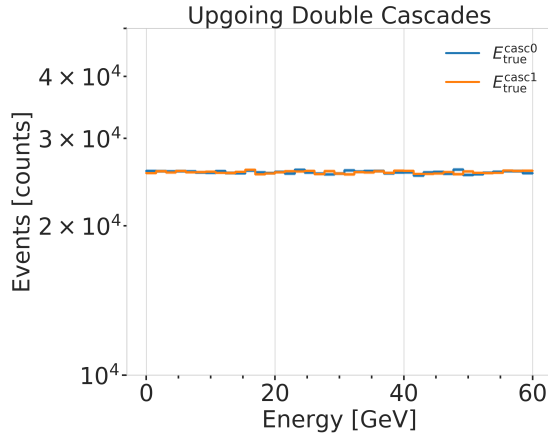
Select interesting plots (energy length) and move the others to the appendix?

Fix caption for simplified gen distris

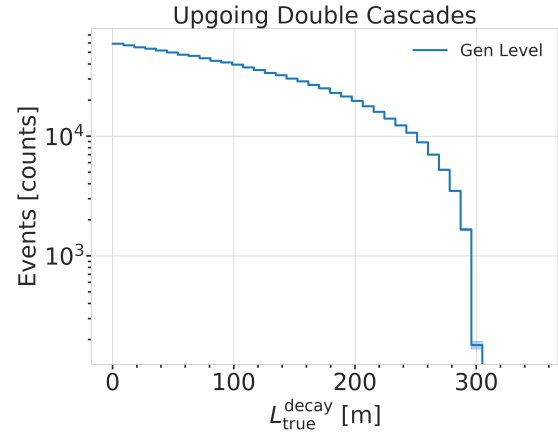
Fix caption for realistic gen distris

### 1.1.4 Generation Level Distributions

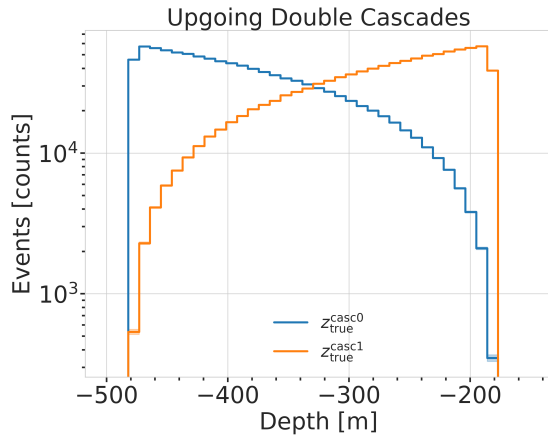




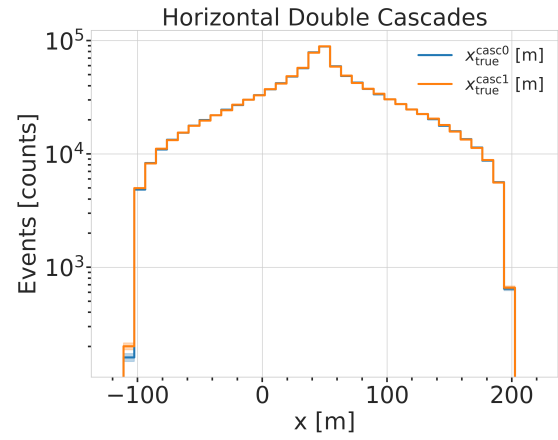
(a) Cascade and total energies (Up-going and horizontal)



(b) Decay lengths (Up-going and horizontal)

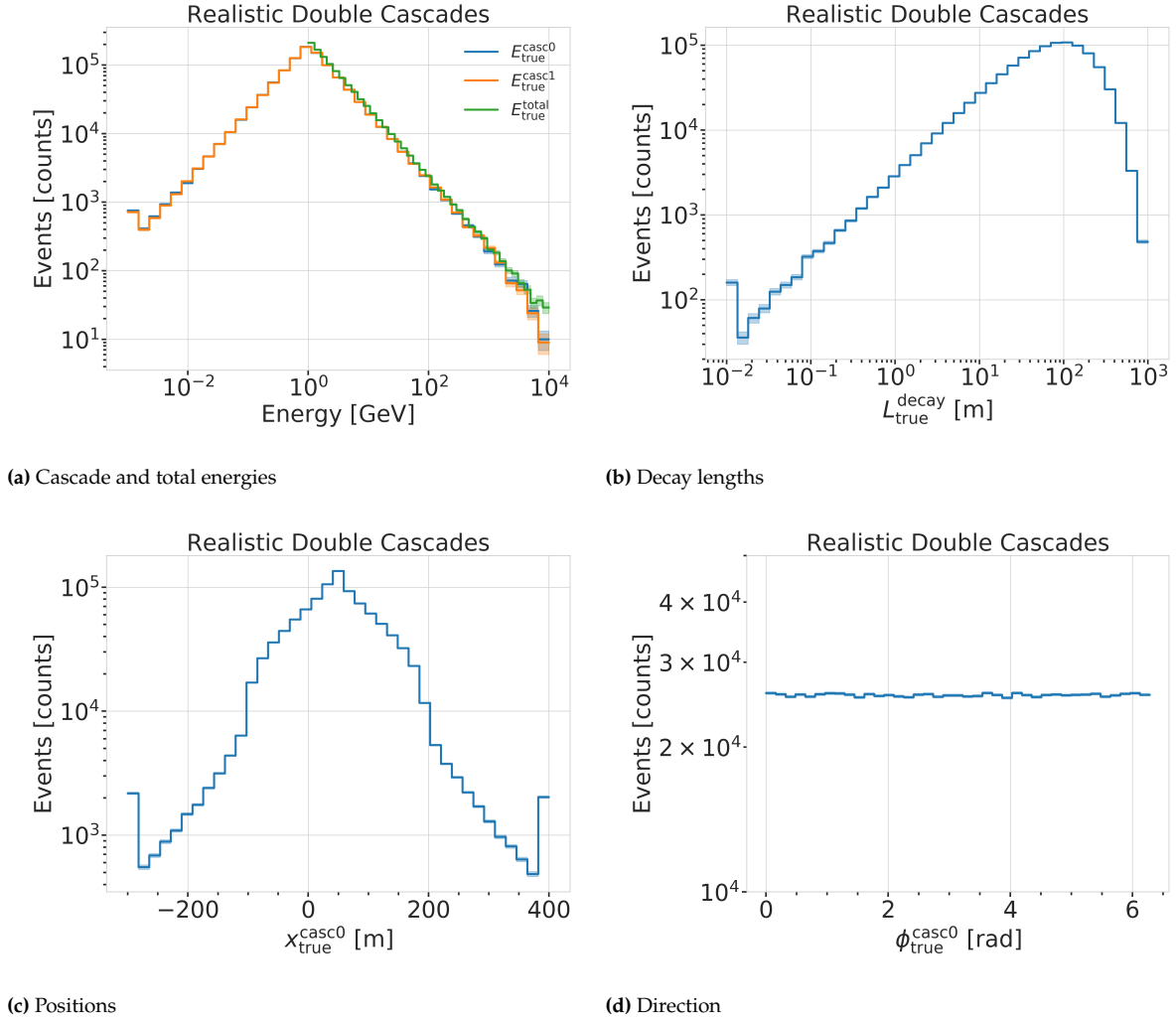


(c) Vertical position (Up-going)



(d) Horizontal position (Horizontal)

**Figure 1.2:** Generation level distributions of the simplistic simulation sets of the 1000 files, with 1000 events each. Only the parameters that are not fixed to a certain value are shown.



**Figure 1.3:** Generation level distributions of the simplistic simulation sets for the 1000 files, with 1000 events each. Only the parameters that are not fixed to a certain value are shown.

## 1.2 Model Dependent Simulation

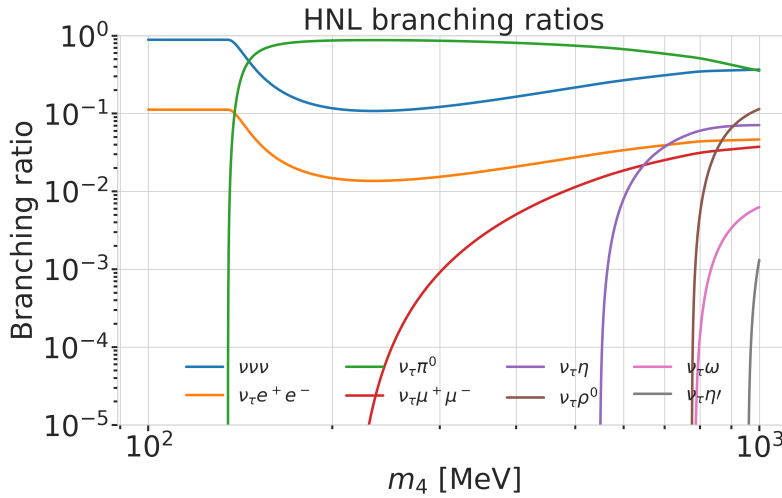
To get a realistic estimate of the HNL event expectation in IceCube DeepCore, depending on the specific model parameters, a generator was developed that is based on the HNL theory introduced in Section ?? . For this work, only the interaction with the  $\tau$ -sector was taken into account ( $|U_{\alpha 4}^2| = 0$ ,  $\alpha = e, \mu$ ), which reduces the physics parameters of interest and relevant for the simulation to the fourth heavy lepton mass,  $m_4$ , and the mixing,  $|U_{\tau 4}^2|$ . The generator uses a customized *LEPTONINJECTOR* (LI) version to create the events and *LEPTONWEIGHTER* (LW) to weight them [2]. The modified LI and the essential components needed for the HNL simulation are described in the next sections, followed by the description of the weighting scheme and the sampling distributions chosen for the simulation generation.

[2]: Abbasi et al. (2021), “LeptonInjector and LeptonWeighter: A neutrino event generator and weighter for neutrino observatories”

### 1.2.1 Custom LeptonInjector

In its standard version, the LI generator produces neutrino interactions by injecting a lepton and a cascade<sup>1</sup> at the interaction vertex of the neutrino, where the lepton is the charged (neutral) particle produced in a CC (NC) interaction and the cascade is the hadronic cascade from the breaking nucleus. Both objects are injected with the same  $(x, y, z, t)$  coordinates and the kinematics are sampled from the differential and total cross-sections that are one of the inputs to LI.

1: The cascades are defined as icetray I3Particles with type *Hadrons*.



**Figure 1.4:** Branching ratios of the HNL within the mass range considered in this work, only considering  $|U_{\tau 4}^2| \neq 0$ , calculated based on the results from [3].

In the modified version, the lepton at the interaction vertex is replaced by the HNL, where the interaction cross-sections are replaced by custom, mass dependent HNL cross-sections. The HNL is forced to decay after a chosen distance to produce secondary SM particles, where the decay mode is randomly chosen based on the mass dependent branching ratios from the kinematically accessible decay modes shown in Figure 1.4. The cross-section and decay width calculations were implemented for this purpose and will be explained in more detail in the following. Another needed addition to LI is that the decay products of the HNL are also added to the list of MC particles in each event. They are injected with the correctly displaced position and delayed time from the interaction vertex, given the HNL decay length. These HNL daughter particles form

[4]: Alwall et al. (2014), "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations"

the second cascade, not as a single cascade object, but as the explicit particles forming the shower. The kinematics of the two-body decays are computed analytically, while the three-body decay kinematics are calculated with MADGRAPH5 [4], which will also be explained further below. Independent of the number of particles in the final state of the HNL decay, the kinematics are calculated/simulated at rest and then boosted along the HNL momentum.

Any number of files can be produced, where the number of events per file is also variable. The injection is done using the LI *volume mode*, for the injection of the primary particle on a cylindrical volume, adding 50 % of the events with  $\nu_\tau$  and the other half with  $\bar{\nu}_\tau$  as primary particle type. The generator takes the custom double-differential/total cross-section splines described below and the parameters defining the sampling distributions as inputs.

add varied total cross-section for a few background HNL events

[5]: Argüelles (),

### Cross-Sections

The cross-sections are calculated using the NuXSPLMKR [5] software, which is a tool to calculate neutrino cross-sections from *parton distribution functions (PDFs)* and then produce splines that can be read and used with LI/LW. The tool was customized to produce the custom HNL cross-sections, where main modification to calculate the cross-sections for the  $\nu_\tau$ -NC interaction into the new heavy mass state, is the addition of a kinematic condition to ensure that there is sufficient energy to produce the heavy mass state. It is the same condition that needs to be fulfilled for the CC case, where the outgoing charged lepton mass is non-zero. Following [6] (equation 7), the condition

$$(1 + x\delta_N)h^2 - (x + \delta_4)h + x\delta_4 \leq 0, \quad (1.1)$$

is implemented for the NC case in the NuXSplMkr code. Here  $\delta_4 = \frac{m_4^2}{s-M^2}$ ,  $\delta_N = \frac{M^2}{s-M^2}$ , and  $h \stackrel{\text{def}}{=} xy + \delta_4$ , with  $x, y$  being the Bjorken variables,  $m_4$  and  $M$  the mass of the heavy state and the target nucleon, respectively, and  $s$  the center of mass energy squared.

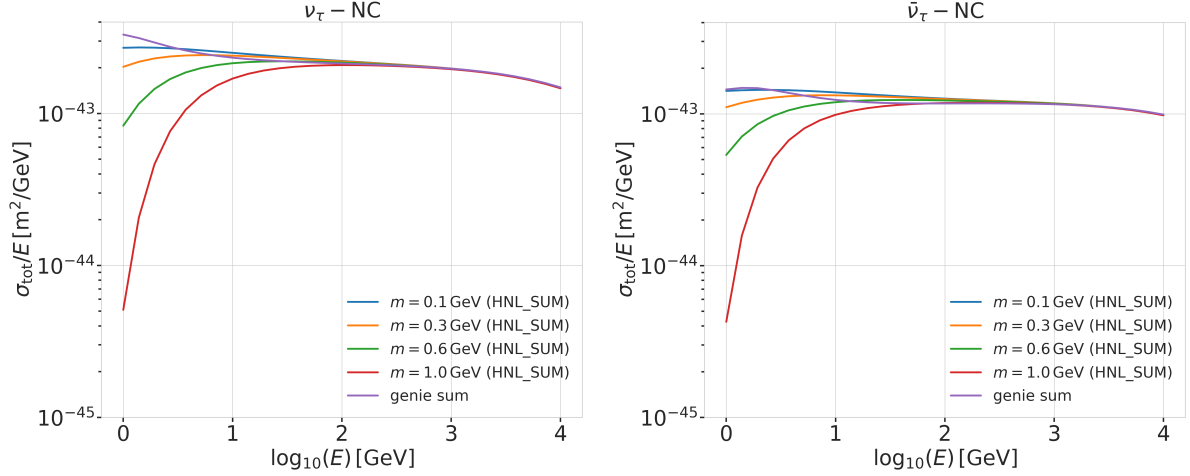
Re-write the following parts..

Since the SM neutrino background simulation explained in was created using GENIE (version 2.12.8), interfaced through the IceCube software package *genie-icetray*, with the GRV98LO PDFs, those were added as *GRV98lo\_patched* to the cross section spline maker, to ensure the best possible agreement. Double-differential ( $dsdx dy$ ) and total ( $\sigma$ ) cross-sections were produced for the four target HNL masses and then splined. The produced cross section splines are stored in the resources of the custom *LeptonInjector* module. Figure 1.5 shows the total cross-sections that were produced compared to the cross section used for the production of the SM  $\nu_\tau/\bar{\nu}_\tau$  NC background simulation .

Add comparisons of SM cross-sections between NuXSplMkr and genie

### Decay Channels

The accessible decay channels are dependent on the mass of the HNL and the allowed mixing. For this analysis, where only  $|U_{\tau 4}|^2 \neq 0$ , the considered decay channels are listed in Table 1.3 and the corresponding branching ratios are shown in Figure ?? . The individual branching ratio



**Figure 1.5:** Custom HNL total cross-sections for the four target masses compared to the total ( $\nu_\tau/\bar{\nu}_\tau$  NC) cross-section used for SM neutrino simulation production with GENIE.

for a specific mass is calculated as  $\text{BR}_i(m_4) = \Gamma_i(m_4)/\Gamma_{\text{total}}(m_4)$ , where  $\Gamma_{\text{total}}(m_4) = \sum \Gamma_i(m_4)$ . The formulas to calculate the decay width show up in multiple references, but we chose to match them to [3], which also discusses the discrepancies in previous literature.

[3]: Coloma et al. (2021), “GeV-scale neutrinos: interactions with mesons and DUNE sensitivity”

**2-Body Decay Widths** The decay to a neutral pseudoscalar mesons is

$$\Gamma_{\nu_4 \rightarrow \nu_\tau P} = |U_{\tau 4}|^2 \frac{G_F^2 m_4^3}{32\pi} f_P^2 (1 - x_P^2)^2, \quad (1.2)$$

with  $x_P = m_P/m_4$  and

$$f_{\pi^0} = 0.130 \text{ GeV}, \quad f_\eta = 0.0816 \text{ GeV}, \quad C_2 = f_{\eta'} = -0.0946 \text{ GeV}, \quad (1.3)$$

while the decay to a neutral vector meson is given by

$$\Gamma_{\nu_4 \rightarrow \nu_\tau V} = |U_{\tau 4}|^2 \frac{G_F^2 m_4^3}{32\pi} \left( \frac{f_V}{m_V} \right)^2 g_V^2 (1 + 2x_V^2)(1 - x_V^2)^2, \quad (1.4)$$

with  $x_V = m_V/m_4$ ,

$$f_{\rho^0} = 0.171 \text{ GeV}^2, \quad f_\omega = 0.155 \text{ GeV}^2, \quad (1.5)$$

and

$$g_{\rho^0} = 1 - 2 \sin^2 \theta_w, \quad g_\omega = \frac{-2 \sin^2 \theta_w}{3}, \quad \sin^2 \theta_w = 0.2229 \quad (1.6)$$

[7].

Channel	Opens	BR [%]
$\nu_4 \rightarrow \nu_\tau \nu_\alpha \bar{\nu}_\alpha$	0 MeV	100.0
$\nu_4 \rightarrow \nu_\tau e^+ e^-$	1 MeV	?
$\nu_4 \rightarrow \nu_\tau \pi^0$	135 MeV	?
$\nu_4 \rightarrow \nu_\tau \mu^+ \mu^-$	211 MeV	?
$\nu_4 \rightarrow \nu_\tau \eta$	548 MeV	?
$\nu_4 \rightarrow \nu_\tau \rho^0$	770 MeV	?
$\nu_4 \rightarrow \nu_\tau \omega$	783 MeV	?
$\nu_4 \rightarrow \nu_\tau \eta'$	958 MeV	?

**Table 1.3:** xx

**3-Body Decay Widths** The (invisible) decay to three neutrinos is

$$\Gamma_{\nu_4 \rightarrow \nu_\tau \nu_\alpha \bar{\nu}_\alpha} = |U_{\tau 4}|^2 \frac{G_F^2 m_4^5}{192\pi^3}, \quad (1.7)$$

[7]: Tiesinga et al. (2021), “CODATA recommended values of the fundamental physical constants: 2018”

**Table 1.4:** Sampling distributions of HNL simulation generation.

Variable	Distribution	Range/Value
energy	$E^{-2}$	[2 GeV, $1 \times 10^4$ GeV]
zenith	uniform (in $\cos(\theta)$ )	[180°, 80°]
azimuth	uniform	[0°, 360°]
vertex $x, y$	uniform	$r=600$ m
vertex $z$	uniform	[-600, 0]m
$m_4$	fixed	[0.3, 0.6, 1.0]GeV
$L_{\text{decay}}$	$L^{-1}$	[0.0004, 1000]m / [1, 1000]m

while the decay to two charged leptons (using  $x_\alpha = (m_\alpha/m_4)^2$ ) of the same flavor reads

$$\Gamma_{\nu_4 \rightarrow \nu_\tau l_\alpha^+ l_\alpha^-} = |U_{\tau 4}|^2 \frac{G_F^2 m_4^5}{192\pi^3} [C_1 f_1(x_\alpha) + C_2 f_2(x_\alpha)], \quad (1.8)$$

with the constants defined as

$$C_1 = \frac{1}{4}(1 - 4s_w^2 + 8s_w^4), \quad C_2 = \frac{1}{2}(-s_w^2 + 2s_w^4), \quad (1.9)$$

the functions as

$$f_1(x_\alpha) = (1 - 14x_\alpha - 2x_\alpha^2 - 12x_\alpha^3)\sqrt{1 - 4x_\alpha} + 12x_\alpha^2(x_\alpha^2 - 1)L(x_\alpha), \quad (1.10)$$

$$f_2(x_\alpha) = 4[x_\alpha(2 + 10x_\alpha - 12x_\alpha^2)\sqrt{1 - 4x_\alpha} + 6x_\alpha^2(1 - 2x_\alpha + 2x_\alpha^2)L(x_\alpha)], \quad (1.11)$$

and

$$L(x) = \ln\left(\frac{1 - 3x_\alpha - (1 - x_\alpha)\sqrt{1 - 4x_\alpha}}{x_\alpha(1 + \sqrt{1 - 4x_\alpha})}\right). \quad (1.12)$$

### MadGraph5 3-Body Decays

The code to produce the 3-body decay kinematics is [MadGraph4 v3.4.0](#) based on the decay diagrams calculated with [FeynRules 2.0](#) using the Lagrangians derived in [3]. The Universal FeynRules Output (UFO) from `effective_HeavyN_Majorana_v103` were used for our calculation. For each meass and corresponding decay channel, we produce  $1e06$  decay kinematic variations (rest frame) and store those in a text file.

To do so, we randomly pick an event from a list that we generated for each three-body decay mode

[3]: Coloma et al. (2021), “GeV-scale neutrinos: interactions with mesons and DUNE sensitivity”

correct information about  
filenumbers, number of  
events and how many that  
were at Gen

### 1.2.2 Sampling Distributions

This is the description of the signal simulation generator used to (re-)start simulation production in December 2023. The underlying sampling distributions are listed in Table ?? . Judging from how the generation/processing efficiency was for the 190607 set, we target  $1e04$  files per set with  $5e05$  events per file at generation, resulting in a maximum of  $5e09$  events per set at generation level. Note here that the actual number of events per set at generation might be a little lower since some events won't be allowed if they don't have enough energy to produce the HNL.

### 1.2.3 Weighting Scheme

The weighting for the HNL signal simulation happens in a **custom stage of PISA**. The only input is the stored OneWeight and the variable physics parameter  $|U_{\tau 4}|^2$ , which is the mixing strength of the new heavy mass state and the tau sector. The custom re-weighting is needed to go from the used sampling PDF (1/L with fixed range in lab frame decay length) to the target PDF (exponential defined by proper lifetime of the HNL). For each event the re-weighting factor is calculated using the gamma factor

$$\gamma = \frac{\sqrt{E_{\text{kin}}^2 + m_{\text{HNL}}^2}}{m_{\text{HNL}}}, \quad (1.13)$$

with the HNL mass  $m_{\text{HNL}}$  and it's kinetic energy  $E_{\text{kin}}$ . The speed of the HNL is calculated as

$$v = c \cdot \sqrt{1 - \frac{1}{\gamma^2}}, \quad (1.14)$$

where  $c$  is the speed of light. With these the lab frame decay length range can be converted into the rest frame lifetime range for each event

$$\tau_{\text{min/max}} = \frac{s_{\text{min/max}}}{v \cdot \gamma}. \quad (1.15)$$

The proper lifetime of each HNL event can be calculated using the total decay width  $\Gamma_{\text{total}}$  shown in Figure ?? and the chosen mixing strength  $|U_{\tau 4}|^2$  as

$$\tau_{\text{proper}} = \frac{\hbar}{\Gamma_{\text{total}}(m_{\text{HNL}}) \cdot |U_{\tau 4}|^2}, \quad (1.16)$$

where  $\hbar$  is the reduced Planck constant. Since the decay length/lifetime of the events is sampled from an inverse distribution instead of an exponential as it would be expected from a particle decay we have to re-weight accordingly to achieve the correct decay length/lifetime distribution. This is done by using the wanted exponential distribution

$$\text{PDF}_{\text{exp}} = \frac{1}{\tau_{\text{proper}}} \cdot e^{\frac{-\tau}{\tau_{\text{proper}}}}, \quad (1.17)$$

and the inverse distribution that was sampled from

$$\text{PDF}_{\text{inv}} = \frac{1}{\tau \cdot (\ln(\tau_{\text{max}}) - \ln(\tau_{\text{min}}))}. \quad (1.18)$$

The lifetime re-weighting factor is calculated as

$$w_{\text{lifetime}} = \frac{\text{PDF}_{\text{exp}}}{\text{PDF}_{\text{inv}}} = \frac{\Gamma_{\text{total}}(m_{\text{HNL}}) \cdot |U_{\tau 4}|^2}{\hbar} \cdot \tau \cdot (\ln(\tau_{\text{max}}) - \ln(\tau_{\text{min}})) \cdot e^{\frac{-\tau}{\tau_{\text{proper}}}}. \quad (1.19)$$

Adding another factor of  $|U_{\tau 4}|^2$  to account for the mixing at the interaction vertex the total re-weighting factor becomes

$$w_{\text{total}} = |U_{\tau 4}|^2 \cdot w_{\text{lifetime}}, \quad (1.20)$$

which can be applied on top of flux and oscillation weight to get the final HNL weight for a given mixing (and mass).

### **1.2.4 Generation Level Distributions**



# Detecting Low Energetic Double Cascades

# 2

## 2.1 Reconstruction

2.1 Reconstruction . . . . . 13

### 2.1.1 Table-Based Minimum Likelihood Algorithms

2.2 Cross Checks . . . . . 13

### 2.1.2 Double Cascade Hypothesis

2.3 Performance . . . . . 14

### 2.1.3 Modification to Low Energy Events

## 2.2 Cross Checks

### 2.2.1 Simplistic Sets

After generation the events are processed with standard Photon, Detector, L1, and L2 processing and then Taupede+MuMillipede is run on top of the L2 files. Multiple versions with different parameters were produced, some with the OscNext baseline parameters, some without detector noise (in Det level) and some with h2-50cm holeice model, to match the holeice model that was used to generate the photonics tables.

**BrightDom Cleaning** To investigate the effect of the BrightDom cleaning cut the 194601 set without detector noise (and baseline hole ice model) is used. The BrightDom cleaning is needed to stop a few DOMs with many photon hits to drive the reconstruction because this leads to large biases in the energy estimations. Historically, the BrightDom cleaning was removing all DOMs that had a charge larger than 10 times the mean charge. After quickly checking some charge distributions and how the mean behaves it was clear that the cut should better be defined based on a metric that is less affected by outliers, like the median. Figure ?? shows where the mean and the median are located for an example event. The cut was re-defined to use the median instead of the mean and 10% of the simulation were processed with Taupede using 30x and 100x the median as BrightDom cutoff. Figure ?? shows where these values fall for the same example event.

As a quick check of the performance of both cuts the decay length resolution/bias and the resolutions/biases of all energies were checked. The reconstructed decay length is almost not affected by applying this cut, which is as expected, because it is mostly dependent on the arrival time of the photons. The effect on the reconstructed energy is much stronger, where a looser cut (100x) shows a significantly larger bias than the tighter cut at (30x). Even though this was not a highly sophisticated optimization of the BrightDom cut, an improvement was achieved by changing from mean to median and selecting the tighter cut (of the two tested). It's hard to tell how this would perform for high energy events, but I'm quite certain that a definition based on the median would be more reliable than on the mean.

## **2.3 Performance**

### **2.3.1 Energy/Decay Length Resolution**

### **2.3.2 Double Cascade Classification**

# Search for an Excess of Heavy Neutral Lepton Events

The measurement performed in this thesis is the search for an excess of HNL events in the 10 years of IceCube DeepCore data. In principle the two physics parameters to be probed are the mass of the HNL,  $m_4$ , and the mixing between the fourth heavy mass state and the SM  $\tau$  sector,  $|U_{\tau 4}|^2$ . Since the mass itself influences the production and decay kinematics of the event and the accessible decay modes, individual mass sets were produced as described in Section 1.2. The mass slightly influences the energy distribution, while the mixing both changes the overall scale of the HNL events and the shape in energy and PID. IceCube DeepCore is suited to measure the excess which appears around and below 20 GeV, due to its production from the atmospheric tau neutrinos, although a reduced lower energy threshold could improve the analysis. The measurement will be performed for the three mass sets individually, while the mixing is the parameter that can be varied continuously and will be measured in the fit.

- 3.1 Final Level Sample . . . . 15
- 3.2 Statistical Analysis . . . . 16
- 3.3 Analysis Checks . . . . . 17
- 3.4 Results . . . . . 18

## 3.1 Final Level Sample

The final level sample of this analysis always consists of the neutrino and muon MC introduced in Section ?? and one of the three HNL samples explained in Section 1.2. All of those simulation sets and the 10 years of IceCube DeepCore data are processed through the full processing and event selection chain described in Section ?? leading to the final level sample. Since applying the last cuts from Section ?? leaves an insignificant amount of pure noise events in the sample, the noise simulation is not included in the analysis and won't be listed here.

- add information about the matter profile used
- add information about the oscillation probability calculation and the software used for it
- get correct final level rates from my pipeline(s)
- add rate and Poisson error for HNL samples
- maybe just pick one mixing?

### 3.1.1 Expected Rates/Events

The rates and the expected events in 10 years are shown in Table 3.1. For the HNL the expectation depends on the mass and the mixing. Shown here are two example mixings for all the three masses. A mixing of 0.0 would result in a rate of 0.0 and therefore no HNL events.

Type	Rate [mHz]	Events (in 10 years)	
$\nu_\mu^{\text{CC}}$	0.3522	$103063 \pm 113$	
$\nu_e^{\text{CC}}$	0.1411	$41299 \pm 69$	
$\nu_\tau^{\text{CC}}$	0.0348	$10187 \pm 22$	
$\nu_{\text{NC}}$	0.0667	$968 \pm 57$	
$\mu$	0.0033	$19522 \pm 47$	
HNL		$ U_{\tau 4} ^2 = 10^{-3}$	$ U_{\tau 4} ^2 = 10^{-1}$
$m_4=0.3 \text{ GeV}$	x.xxx	2.5	1342.5
$m_4=0.6 \text{ GeV}$	x.xxx	9.0	1207.0
$m_4=1.0 \text{ GeV}$	x.xxx	9.6	966.5

**Table 3.1:** Final level rates and event expectation of the SM background particle types and the HNL signal for all three masses and two example mixing values.

**Table 3.2:** Three dimensional binning used in the analysis. All variables are from the FLERCNN reconstruction explained in Section ??.

Variable	$N_{\text{bins}}$	Edges	Step
$P_\nu$	3	[0.00, 0.25, 0.55, 1.00]	linear
$E$	12	[5.00, 100.00]	logarithmic
$\cos(\theta)$	8	[-1.00, 0.04]	linear

### 3.1.2 Analysis Binning

[8]: Yu et al. (2023), “Recent neutrino oscillation result with the IceCube experiment”

The identical binning to the analysis performed in [8] is used. It was chosen such that the track-like bin has the largest  $\nu_\mu$ -CC fraction. Extend the binning towards lower energies or increasing the number of bins did not improve the HNL sensitivities significantly. It also has to be considered that sufficient data events need to end up in the individual bins to result in a good fit, which was already investigated in the previous analysis. To mitigate the low data statistics, a few bins were not taken into account in the analysis. There are three bins in PID (cascade-like, mixed and track-like), 12 bins in reconstructed energy, and 8 bins in cosine of the reconstructed zenith angle as specified in Table 3.2. Originally, there were two more bins in  $\cos(\theta)$ , which were removed to reduce muons coming from the horizon and some low energy bins in the cascade-like bin are removed due to the low event expectation.

add 3D expectation and/or S/sqrt(B) plots

## 3.2 Statistical Analysis

### 3.2.1 Test Statistic

The measurements are performed by comparing the weighted MC to the data. Through variation of the nuisance and physics parameters that govern the weights, the best matching set of parameters can be found. The comparison is done using a modified  $\chi^2$  defined as

$$\chi_{\text{mod}}^2 = \sum_{i \in \text{bins}} \frac{(N_i^\nu + N_i^\mu + N_i^{\text{HNL}} - N_i^{\text{obs}})^2}{N_i^\nu + N_i^\mu + N_i^{\text{HNL}} + (\sigma_i^\nu)^2 + (\sigma_i^\mu)^2 + (\sigma_i^{\text{HNL}})^2} + \sum_{j \in \text{syst}} \frac{(s_j - \hat{s}_j)^2}{\sigma_{s_j}^2}, \quad (3.1)$$

as the test statistic (TS), where  $N_i^\nu$ ,  $N_i^\mu$ , and  $N_i^{\text{HNL}}$  are the expected number of events in bin  $i$  from neutrinos, atmospheric muons, and HNL, while  $N_i^{\text{obs}}$  is the observed number of events in bin  $i$ . The expected number of events from each particle type is calculated by summing the weights of all events in the bin  $N_i^{\text{type}} = \sum_i^{\text{type}} \omega_i$ , with the statistical uncertainty being  $(\sigma_i^{\text{type}})^2 = \sum_i^{\text{type}} \omega_i^2$ . The expected Poisson error is calculated using the combined expectation of neutrinos, atmospheric muons, and HNL events. The additional term in Equation 3.1 is included to apply a penalty term for prior knowledge of the systematic uncertainties of the parameters where they are known.  $s_j$  are the systematic parameters that are varied in the fit, while  $\hat{s}_j$  are their nominal values and  $\sigma_{s_j}$  are the known uncertainties.

Do I want/need to include the description of the KDE muon estimation?

Add table with all systematic uncertainties used in this analysis (in the analysis chapter).

add final level effects of varying the axial mass parameters (or example of one)

add final level effects of varying the DIS parameter (or example of one)

### 3.2.2 Systematic Uncertainties

### 3.3 Analysis Checks

Fitting to data will be performed in a *blind* manner, where the analyzer does not immediately see the fitted physics and nuisance parameter values, but first checks that a set of pre-defined *goodness of fit* (GOF) criteria are fulfilled. If those criteria are met to satisfaction the fit results are unblinded and the full result can be revealed. Before these blind fits to data are run, the robustness of the analysis method is tested using pseudo-data that is generated using the MC sets.

#### 3.3.1 Minimization Robustness

To find the set of parameters that describes the data best, a staged minimization routine is used. In the first stage, a fit with coarse minimizer settings is performed to find a rough estimate of the *best fit point* (BFP). In the second stage, the fit is performed again in both octants<sup>1</sup> of  $\theta_{23}$ , starting from the BFP of the coarse fit. For each individual fit the MIGRAD routine of MINUIT [9] is used to minimize the  $\chi^2$  TS defined in Equation 3.1. Iminuit is a fast, python compatible minimizer based on the MINUIT2 C++ library [10]. The individual minimizer settings are shown in Table 3.3.

To test the minimization routine and to make sure it consistently recovers any injected physics parameters, pseudo-data sets are produced from the MC by choosing the nominal nuisance parameters and specific physics parameters, without adding any statistical or systematic fluctuations to it. These so-called *Asimov*<sup>2</sup> data sets are then fit back with the full analysis chain. This type of test is called *Asimov inject/recover test*. A set of mixing values between  $10^{-3}$  and  $10^0$  is injected and fit back. Even though this range is well within the excluded regions by other experiments, discussed in Section ??, this covers the current sensitive region of the analysis in IceCube DeepCore. Without fluctuations the fit is expected to always recover the injected parameters (both physics and nuisance parameters). The fitted mixing values from the Asimov inject/recover tests are compared to the true injected values in Figure 3.1 for the 0.6 GeV set. As expected, the fit is always able to recover the injected physics parameter and the nuisance parameters. Additional plots for the other mass sets can be found in Section A.0.1.

#### 3.3.2 Ensemble Tests

To estimate the goodness of fit, pseudo-data is generated from the MC by injecting the BFP parameters as true parameters and then fluctuating the expected bin counts using Poisson fluctuation. The resulting pseudo-data sets are then fit back with the analysis chain. By comparing the distribution of TS values from this *ensemble* of pseudo-data trials to the TS of the fit to real data, a p-value can be calculated. The p-value is the probability of finding a TS value at least as large as the one from the data fit. Figure 3.2 shows the TS distribution from the ensemble tests for the 0.6 GeV mass set and the observed TS value from the fit, resulting in a p-value of 1.23 %<sup>3</sup>. Plots for the additional two mass sets are shown in Section A.0.2.

1: There is a degeneracy between the lower octant ( $\theta_{23} < 45^\circ$ ) and the upper octant ( $\theta_{23} > 45^\circ$ ), which can lead to TS minima (local and global) at two positions that are mirrored around  $45^\circ$  in  $\theta_{23}$ .

[9]: Dembinski et al. (2022), *scikit-hep/iminuit: v2.17.0*

[10]: James et al. (1975), “Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations”

Fit	Err.	Prec.	Tol.
Coarse	1e-1	1e-8	1e-1
Fine	1e-5	1e-14	1e-5

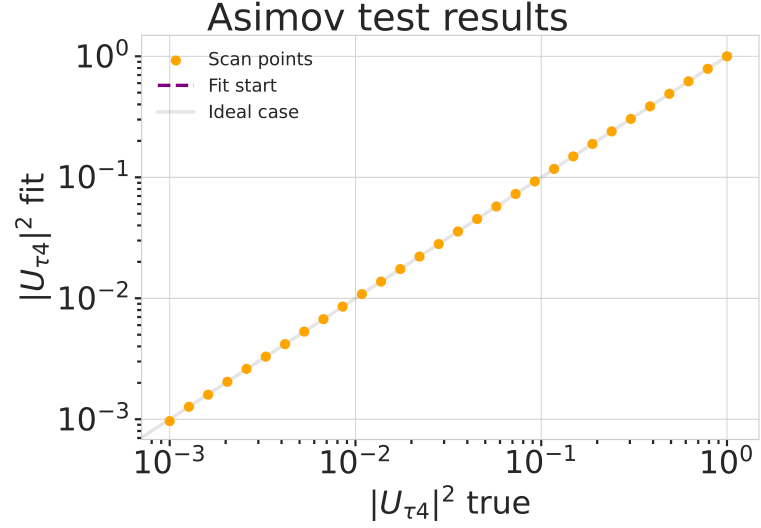
**Table 3.3:** Migrad settings for the two stages in the minimization routine. *Err.* are the step size for the numerical gradient estimation, *Prec.* is the precision with which the LLH is calculated, and *Tol.* is the tolerance for the minimization.

2: A pseudo-data set without statistical fluctuations is called Asimov data set.

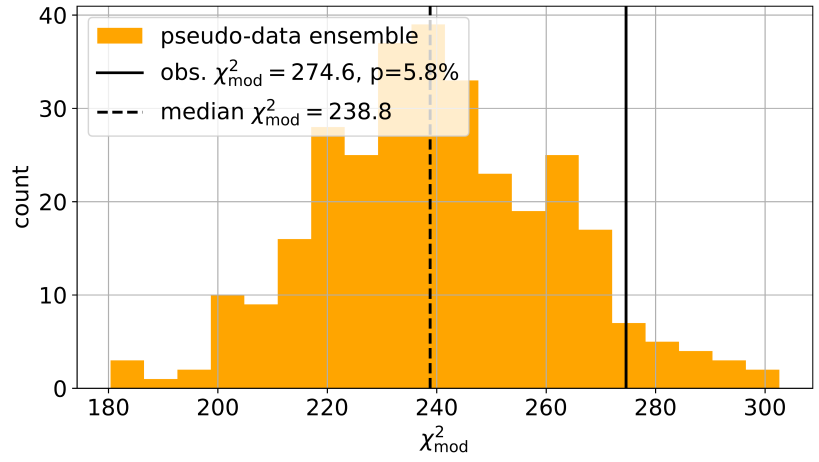
Do I want additional plots for this (fit diff, LLH distr, minim. stats, param. fits)?

Add bin-wise TS distribution? Add 3D TS maps?

3: The p-values for the 0.3 GeV and 1.0 GeV are 1.23 % and 1.23 %, respectively



**Figure 3.1:** Asimov inject/recover test for the 0.6 GeV mass set. Mixing values between  $10^{-3}$  and  $10^0$  are injected and fit back with the full analysis chain. The injected parameter is always recovered within the statistical uncertainty.



**Figure 3.2:** Observed fit TS and TS distribution from pseudo-data trials for the 0.6 GeV mass set.

## 3.4 Results

### 3.4.1 Best Fit Parameters

### 3.4.2 Upper Limits

### 3.4.3 Post-Fit Data/MC Agreement

### 3.4.4 Likelihood Coverage

# APPENDIX



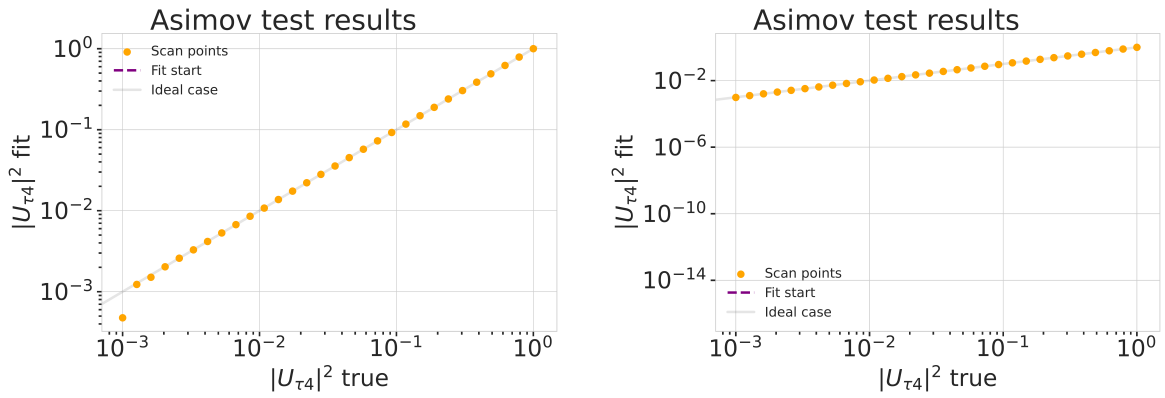


# A

## Analysis Checks

### A.0.1 Minimization Robustness

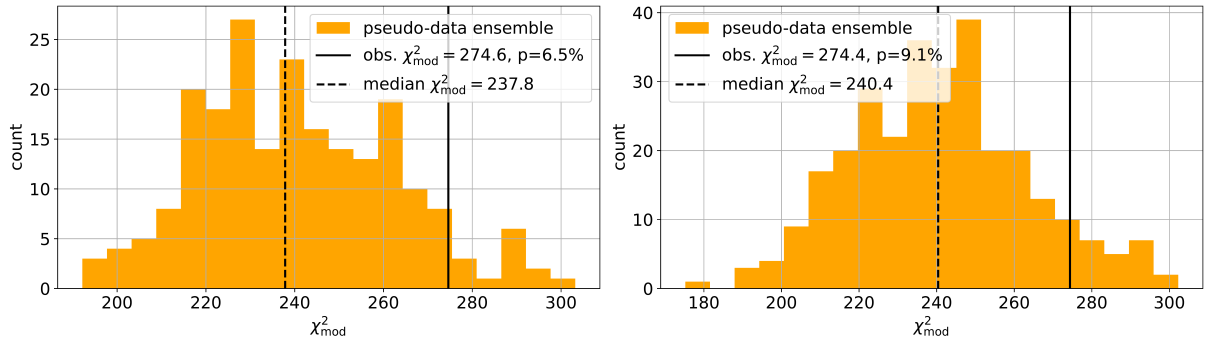
Figure A.1 shows additional Asimov inject/recover tests for the 0.3 GeV and 1.0 GeV mass sets. The tests were described in Section 3.3.1.



**Figure A.1:** Asimov inject/recover test for the 0.3 GeV (left) and 1.0 GeV (right) mass sets. Mixing values between  $10^{-3}$  and  $10^0$  are injected and fit back with the full analysis chain. The injected parameter is always recovered within the statistical uncertainty.

### A.0.2 Ensemble Tests

Figure A.2 shows additional TS distributions from pseudo-data trials and the observed TS from the fit to the data for the ensemble for the 0.3 GeV and 1.0 GeV mass sets. The tests were described in Section 3.3.2.



**Figure A.2:** Observed fit TS and TS distribution from pseudo-data trials for the 0.3 GeV (left) and 1.0 GeV (right) mass set.



# Bibliography

Here are the references in citation order.

- [1] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009 (cited on page 1).
- [2] R. Abbasi et al. “LeptonInjector and LeptonWeighter: A neutrino event generator and weighter for neutrino observatories”. In: *Comput. Phys. Commun.* 266 (2021), p. 108018. doi: [10.1016/j.cpc.2021.108018](https://doi.org/10.1016/j.cpc.2021.108018) (cited on page 7).
- [3] P. Coloma et al. “GeV-scale neutrinos: interactions with mesons and DUNE sensitivity”. In: *Eur. Phys. J. C* 81.1 (2021), p. 78. doi: [10.1140/epjc/s10052-021-08861-y](https://doi.org/10.1140/epjc/s10052-021-08861-y) (cited on pages 7, 9, 10).
- [4] J. Alwall et al. “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations”. In: *JHEP* 07 (2014), p. 079. doi: [10.1007/JHEP07\(2014\)079](https://doi.org/10.1007/JHEP07(2014)079) (cited on page 8).
- [5] C. Argüelles. <https://github.com/arguelles/NuXSplMkr> (cited on page 8).
- [6] J.-M. Levy. “Cross-section and polarization of neutrino-produced tau’s made simple”. In: *J. Phys. G* 36 (2009), p. 055002. doi: [10.1088/0954-3899/36/5/055002](https://doi.org/10.1088/0954-3899/36/5/055002) (cited on page 8).
- [7] E. Tiesinga et al. “CODATA recommended values of the fundamental physical constants: 2018”. In: *Rev. Mod. Phys.* 93 (2 July 2021), p. 025010. doi: [10.1103/RevModPhys.93.025010](https://doi.org/10.1103/RevModPhys.93.025010) (cited on page 9).
- [8] S. Yu and J. Micallef. “Recent neutrino oscillation result with the IceCube experiment”. In: *38th International Cosmic Ray Conference*. July 2023 (cited on page 16).
- [9] H. Dembinski et al. *scikit-hep/iminuit: v2.17.0*. Version v2.17.0. Sept. 2022. doi: [10.5281/zenodo.7115916](https://doi.org/10.5281/zenodo.7115916) (cited on page 17).
- [10] F. James and M. Roos. “Minuit: A System for Function Minimization and Analysis of the Parameter Errors and Correlations”. In: *Comput. Phys. Commun.* 10 (1975), pp. 343–367. doi: [10.1016/0010-4655\(75\)90039-9](https://doi.org/10.1016/0010-4655(75)90039-9) (cited on page 17).