

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^AT_EX using the open-source kaobook template class.

The source code of this thesis is available at:

https://github.com/LeanderFischer/phd_thesis

Zusammenfassung

Zusammenfassung ...

Abstract

Abstract ...

Todo list

Make my own DC string positions/distances plot version.	2
Re-make plot with all energies (cascades and total, both sets (they are the same))	3
Re-make plot with all decay lengths (both sets)	3
Re-make plot with 3 target masses and better labels	6
Add comparions of SM cross-sections between NuXSSplMkr and genie?	7
add varied total cross-section for a few background HNL events (for QE/RES variations?!?)	7
Calculate max BRs	7
where is PISA introduced? is there a reference?	9
add table with number of gen level files? mention the event number is smaller becasue of kinemtic condition?	10

Contents

Abstract	iii
Contents	vii
1 Heavy Neutral Lepton Signal Simulation	1
1.1 Model Independent Simulation	1
1.1.1 Generator Functions	1
1.1.2 Simplistic Sets	2
1.1.3 Realistic Set	3
1.2 Model Dependent Simulation	5
1.2.1 Custom LeptonInjector	5
1.2.2 Sampling Distributions	8
1.2.3 Weighting Scheme	9
1.2.4 Generation Level Distributions	10
APPENDIX	11
A Heavy Neutral Lepton Signal Simulation	13
A.1 Model Independent Simulation Distributions	13
A.2 Model Dependent Simulation Distributions	14
B Analysis Checks	15
B.1 Minimization Robustness	15
B.1.1 Ensemble Tests	15
Bibliography	17

List of Figures

1.1	DeepCore string spacing	2
1.2	Simplified model independent simulation generation level distributions	3
1.3	Realistic model independent simulation generation level distributions	4
1.4	HNL branching ratios	5
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.	6
1.6	Model dependent simulation generation level distributions	10
A.1	Simplified model independent simulation generation level distributions	13
A.2	Realistic model independent simulation generation level distributions	14
A.3	Model dependent simulation generation level distributions	14
B.1	Asimov inject/recover test (0.3 GeV, 1.0 GeV)	15
B.2	Pseudo-data trials TS distribution (0.3 GeV, 1.0 GeV)	15

List of Tables

1.1	Simplified model independent simulation sampling distributions	3
1.2	Realistic model independent simulation sampling distributions	4
1.3	HNL mass dependent decay channels	7
1.4	Model dependent simulation sampling distributions	9

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	5

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 parameter choice, 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. The results of this study will be discussed in Chapter ??.

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 from IceCube zenith/azimuth angles to a direction vector and vice versa. There is a function to create an EM cascade particle from 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 functions, 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 one `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 version.

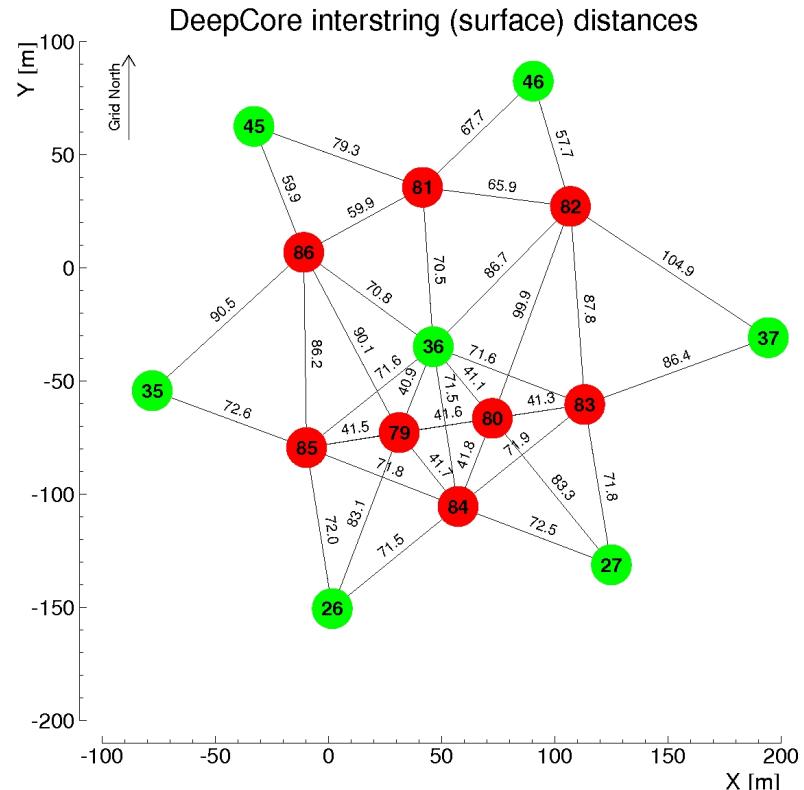


Figure 1.1: Horizontal positions and distances between DeepCore strings. Red strings are instrumented more densely (vertically) and partially have higher quantum efficiency (HQE) DOMs.

The first set is used to investigate one of the potentially 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 ~ 70 m to its neighboring strings. As already mentioned in Section ??, DeepCore strings have higher quantum efficiency DOMs and a denser vertical spacing, making them better to detect low energetic events that produce little light. To produce the events, the x, y position of the cascades is fixed to the center of string 81 while the z position is

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: Generation level sampling distributions and ranges/values of up-going and horizontal model independent simulation.

sampled uniformly along the strings z elongation and the energies are sampled uniformly between 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.

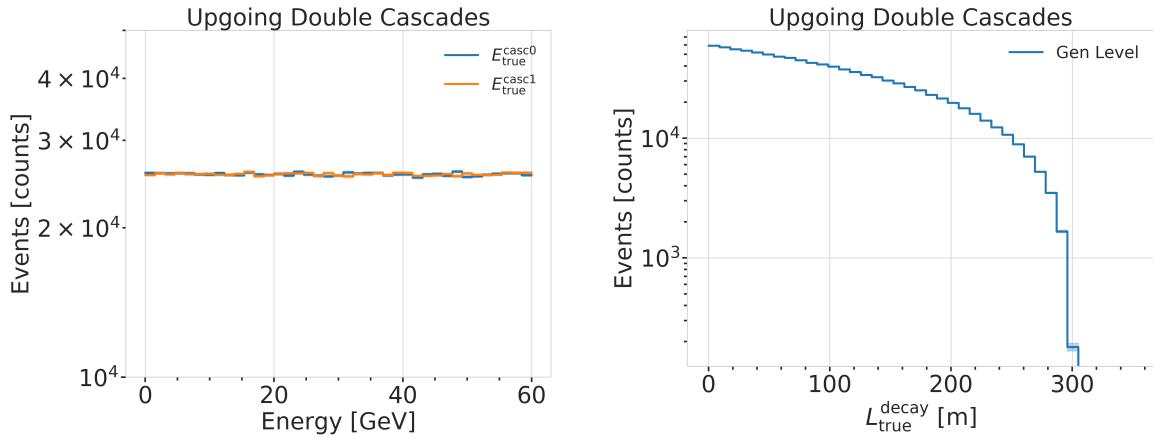


Figure 1.2: Generation level distributions of the simplistic simulation sets. Cascade and total energies (left) and decay lengths (right) of both sets are shown.

The second set is used to investigate the reconstruction performance for horizontal events, where the spacing between DOMs is much larger. 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. Some examples of the generation level distributions of the simplified sets are shown in Figure 1.2, while further distributions can be found in Figure A.1. The variables that are uniformly sampled or fixed to a certain value are not presented.

Re-make plot with all energies (cascades and total, both sets (they are the same))

Re-make plot with all decay lengths (both sets)

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

Table 1.2: Generation level sampling distributions and ranges/values of the realistic model independent simulation.

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

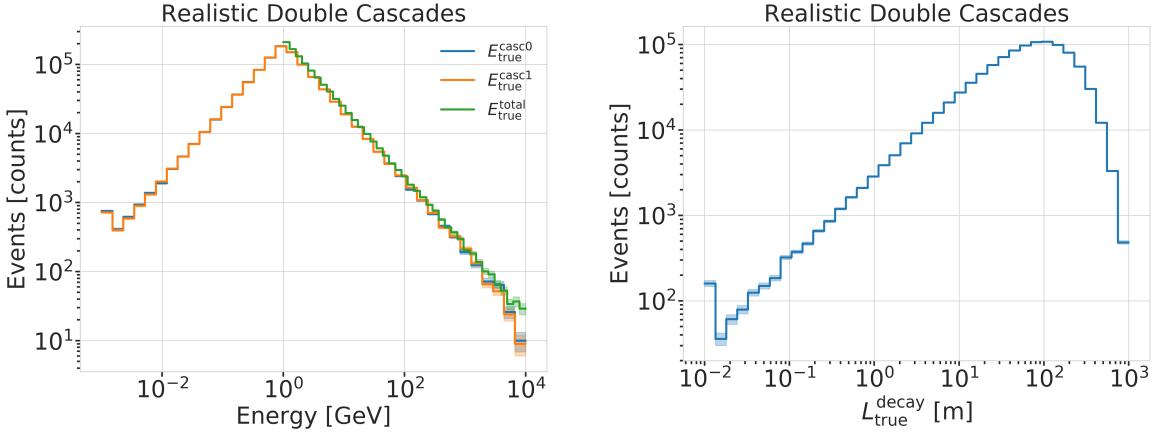


Figure 1.3: Generation level distributions of the simplistic realistic set. Shown are the cascade and total energies (left) and decay lengths (right).

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 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. Example distributions of the generation level variables are shown in Figure 1.3, while further distributions can be found in Figure A.2.

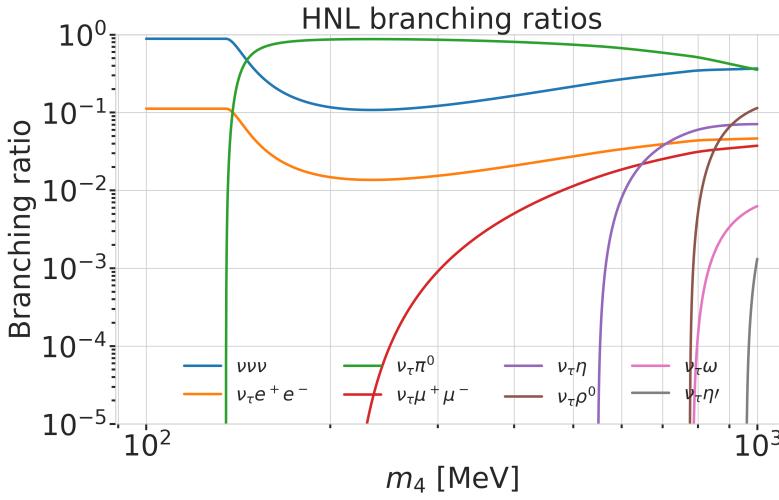
1.2 Model Dependent Simulation

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 τ -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 cascade¹ 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 **MADGRAPH** [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 ν_τ 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.

Cross-Sections

The cross-sections are calculated using the **NuXSSplMkr** [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 the main modification to calculate the cross-sections for the ν_τ -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 **NuXSSplMkr** 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-make plot with 3 target masses and better labels

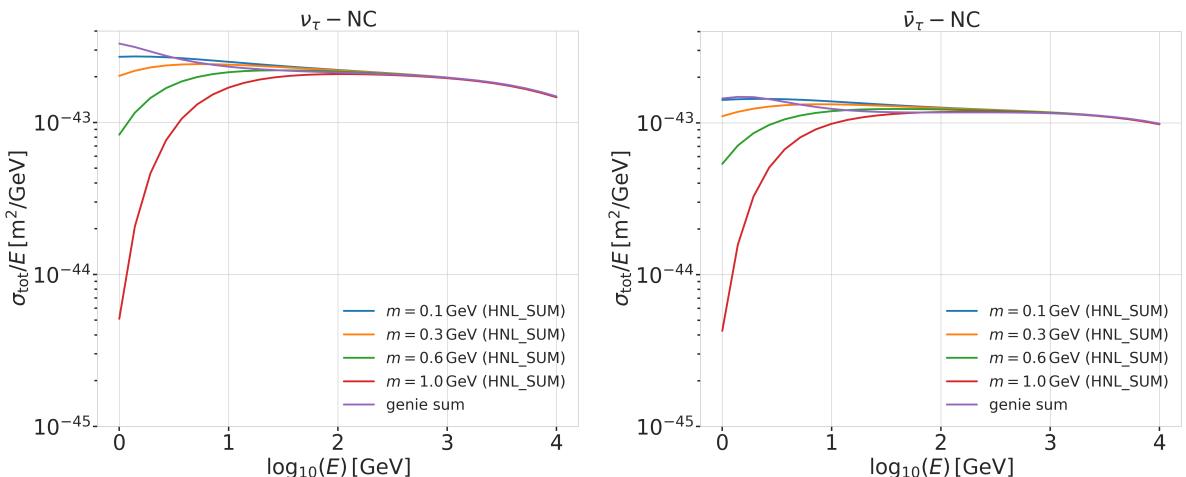


Figure 1.5: Custom HNL total cross-sections for the four target masses compared to the total ($\nu_\tau/\bar{\nu}_\tau$ NC) cross-section used for SM neutrino simulation production with GENIE.

As already described in Section ??, the SM neutrino background simulation is created using the GRV98LO PDFs. These PDFs also had to be added to the cross-section spline maker, to ensure good agreement between the background and signal cross-sections. The double differential ($ds/dx dy$) and total (σ) cross-sections were produced for the chosen target HNL masses and then splined. The produced cross-section are added to the custom LI version and used for the simulation generation and weighting. 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. Above $\sim 2 \times 10^2$ GeV they match, which is the wanted result of using the identical input PDFs.

Add comparisons of SM cross-sections between NuXSSplMkr and genie?

add varied total cross-section for a few background HNL events (for QE/RES variations?!)

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 1.4. The individual branching ratio for a specific mass is calculated as $BR_i(m_4) = \Gamma_i(m_4)/\Gamma_{\text{total}}(m_4)$, where $\Gamma_{\text{total}}(m_4) = \sum \Gamma_i(m_4)$. The formulas to calculate the decay widths 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"

Channel	Opens	$\hat{BR} [\%]$
$\nu_4 \rightarrow \nu_\tau \nu_\alpha \bar{\nu}_\alpha$	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: Possible decay channels of the HNL, considering only $|U_{\tau 4}|^2 \neq 0$. Listed is the mass at which each channel opens and the maximum branching ratio.

Calculate max BRs

2-Body Decay Widths The decay to a neutral pseudoscalar meson 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].

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

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

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

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

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

MadGraph 3-Body Decays

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

The specific MadGraph version used to produce the 3-body decay kinematics is [MadGraph4 v3.4.0](#). As input, the decay diagrams calculated with [FeynRules 2.0](#) using the Lagrangians derived in [3] are taken. The Universal FeynRules Output (UFO) from `EFFECTIVE_HEAVYN_MAJORANA_-v103` were used for our calculation. For each mass and corresponding decay channels, we produce 1×10^6 decay kinematic variations in the rest frame and store those in a text file. During event generation, we randomly pick an event from that list, to simulate the decay kinematics of a 3-body decay.

1.2.2 Sampling Distributions

In principle, the generation level sampling distributions should be chosen such that at final level of the selection chain, the for the analysis relevant phase space is covered with sufficient statistics to make a reasonable estimate of the event expectation. Insufficiently large initial distributions lead to an underestimate of the expected rates, which is still correct as in being the more conservative estimate, but limiting the analysis potential. Three discrete simulation sets were produced with HNL masses of 0.3 GeV, 0.6 GeV and 1.0 GeV. Each set consist of a part that is generated for very short decay lengths and one for long decay lengths, because during development it became clear that the low lengths component is crucial to get a reasonable event estimate. To remaining sampling distributions are identical for all sets and are listed in Table ???. The target file number for each set was 5000, with 5×10^5 events per file.

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

Table 1.4: Generation level sampling distributions and ranges/values of the model dependent simulation sets.

1.2.3 Weighting Scheme

To produce physically correct event distributions based on the arbitrary generation sampling distributions for the HNL simulation, the forward folding method that was already introduced for the SM simulation in Section ?? is used again. How this will be applied in the analysis is discussed in Section ??.

The weighting scheme that will be explained in the following is implemented in the IceCube low energy analysis framework PISA in [this custom stage](#). The only needed input is the mixing strength $|U_{\tau 4}|^2$, which is the variable physics parameter in this analysis. This weighting is needed to go from the used decay length sampling distribution (inverse $1/L$ with fixed range in lab frame) to the target distribution (exponential defined by proper lifetime of the HNL). For each event the additional weight is calculated using the gamma factor

$$\gamma = \frac{\sqrt{E_{\text{kin}}^2 + m_4^2}}{m_4}, \quad (1.13)$$

with the HNL mass m_4 and it's kinetic energy E_{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 $[s_{\min}, s_{\max}]$ can be converted into the rest frame lifetime range $[\tau_{\min}, \tau_{\max}]$ for each event

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

The proper lifetime of each HNL event can be calculated using the total decay width Γ_{total} from Section 1 and the chosen mixing strength $|U_{\tau 4}|^2$ as

$$\tau_{\text{proper}} = \frac{\hbar}{\Gamma_{\text{total}}(m_4) \cdot |U_{\tau 4}|^2}, \quad (1.16)$$

where \hbar is the reduced Planck constant. Since the decay lengths/lifetimes of the events are 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

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

where is PISA introduced? is there a reference?

This re-weighting factor is then calculated as

$$w_{\text{lifetime}} = \frac{\text{PDF}_{\text{exp}}}{\text{PDF}_{\text{inv}}} = \frac{\Gamma_{\text{total}}(m_4) \cdot |U_{\tau 4}|^2}{\hbar} \cdot \tau \cdot (\ln(\tau_{\max}) - \ln(\tau_{\min})) \cdot e^{\frac{-\tau}{\tau_{\text{proper}}}} . \quad (1.19)$$

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

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

If this additional weighting factor is multiplied to a generation weight (like in Equation ??), the livetime, and the oscillated primary neutrino flux, it results in the number of expected events in the detector for this particular MC event for a chosen mixing (and mass).

add table with number of gen level files? mention the event number is smaller because of kinematic condition?

1.2.4 Generation Level Distributions

Figure 1.6 shows some selected generation level distributions. Additional distributions can be found in Figure A.3.

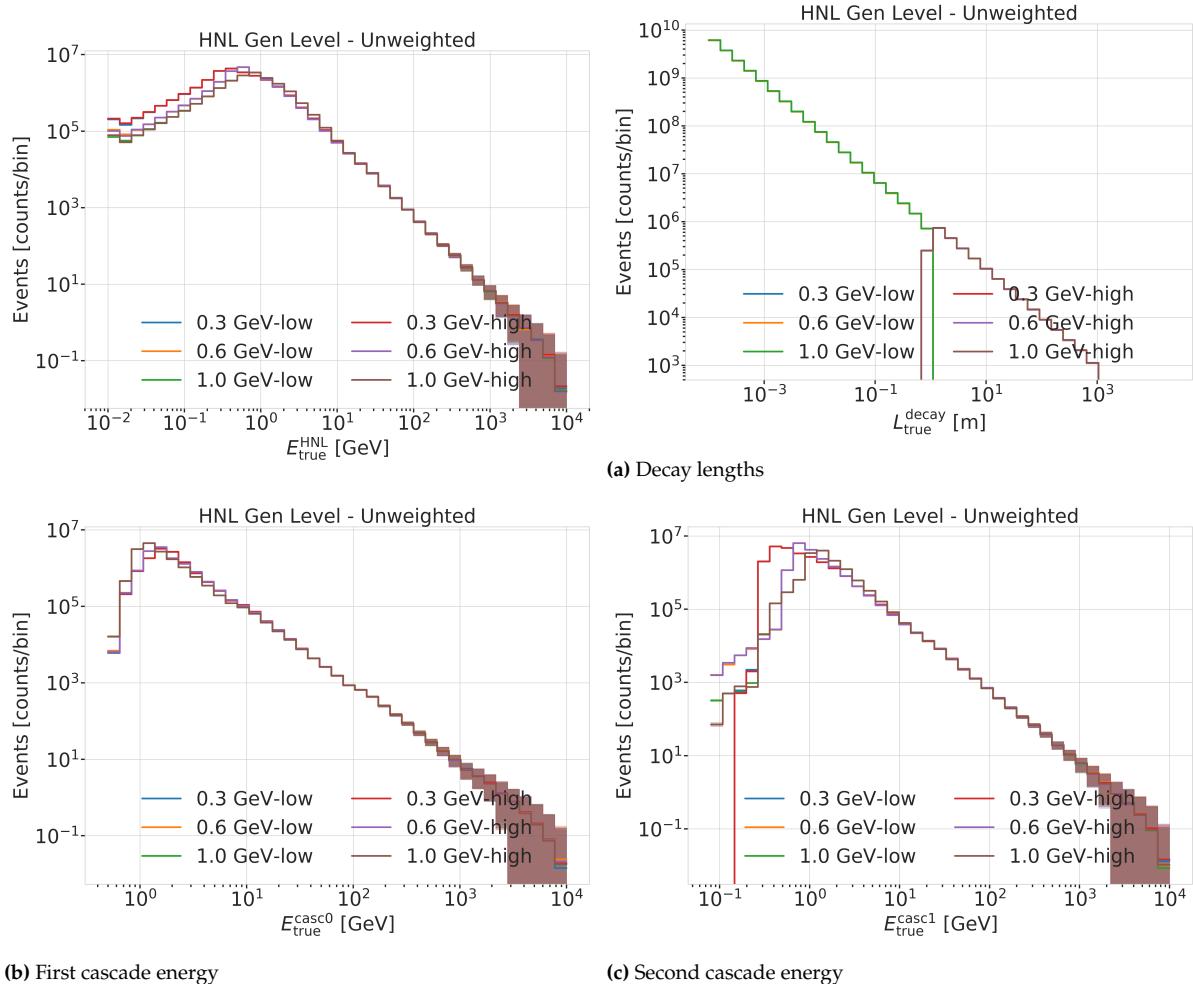


Figure 1.6: Generation level distributions of the model dependent simulation.

APPENDIX

A

Heavy Neutral Lepton Signal Simulation

A.1 Model Independent Simulation Distributions

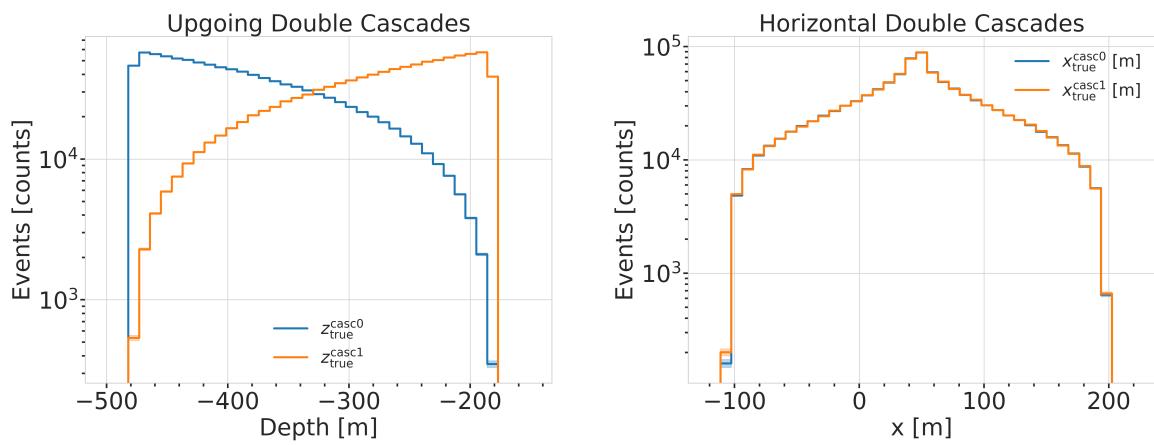


Figure A.1: Generation level distributions of the simplistic simulation sets. Vertical positions (left) and horizontal positions (right) of both sets are shown.

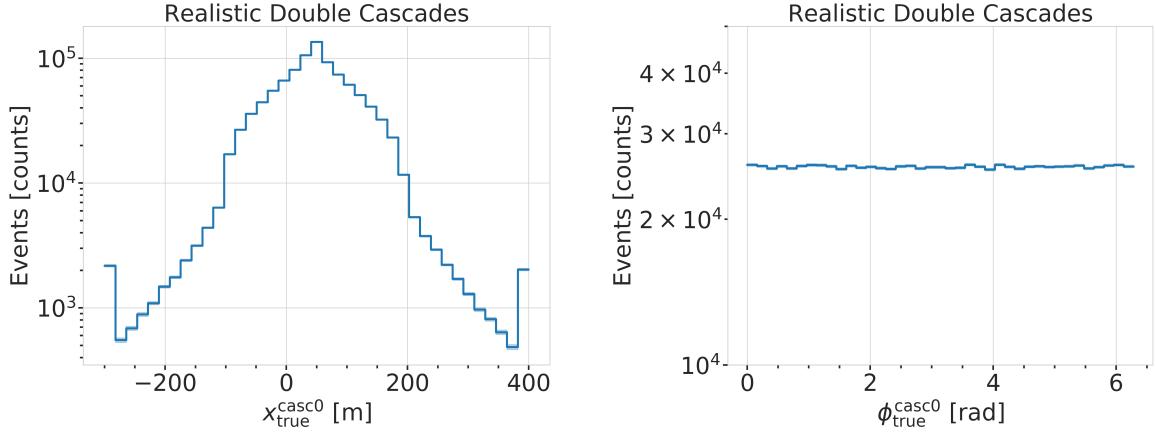


Figure A.2: Generation level distributions of the simplistic realistic set. Shown are the cascade x, y, z positions (left) and direction angles (right).

A.2 Model Dependent Simulation Distributions

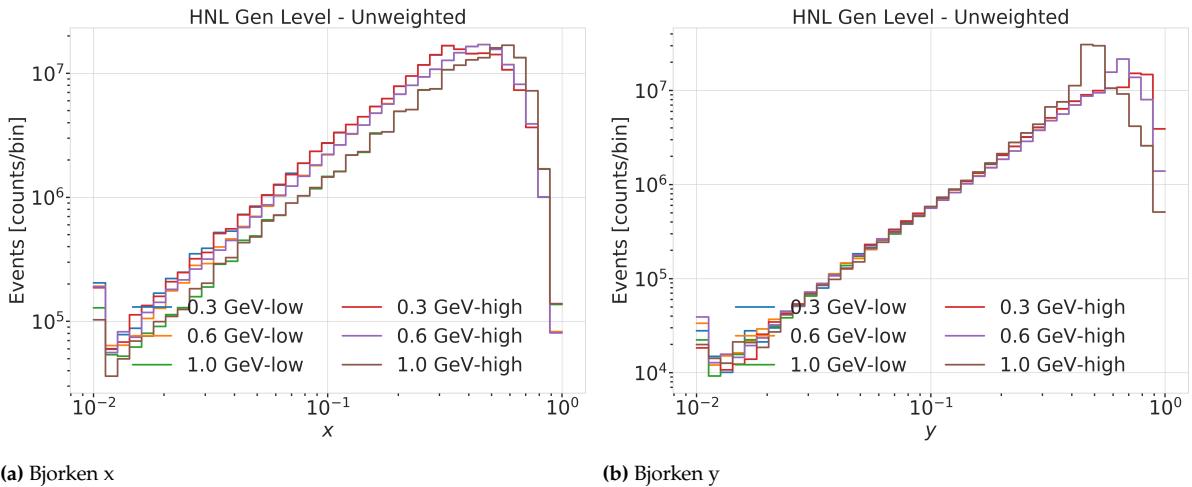


Figure A.3: Generation level distributions of the model dependent simulation.

B

Analysis Checks

B.1 Minimization Robustness

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

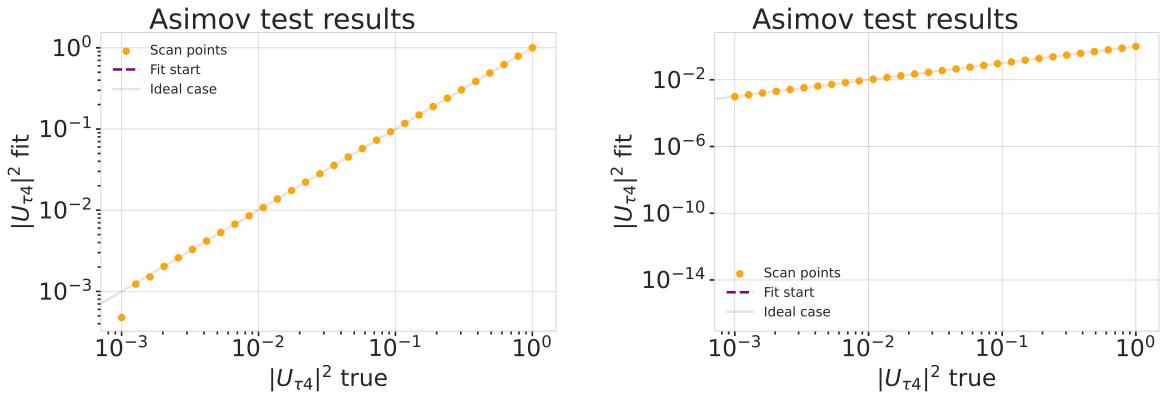


Figure B.1: Asimov inject/recover test for the 0.3 GeV (left) and 1.0 GeV (right) mass sets. Mixing values between 10^{-3} and 10^0 are injected and fit back with the full analysis chain. The injected parameter is always recovered within the statistical uncertainty.

B.1.1 Ensemble Tests

Figure B.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 ??.

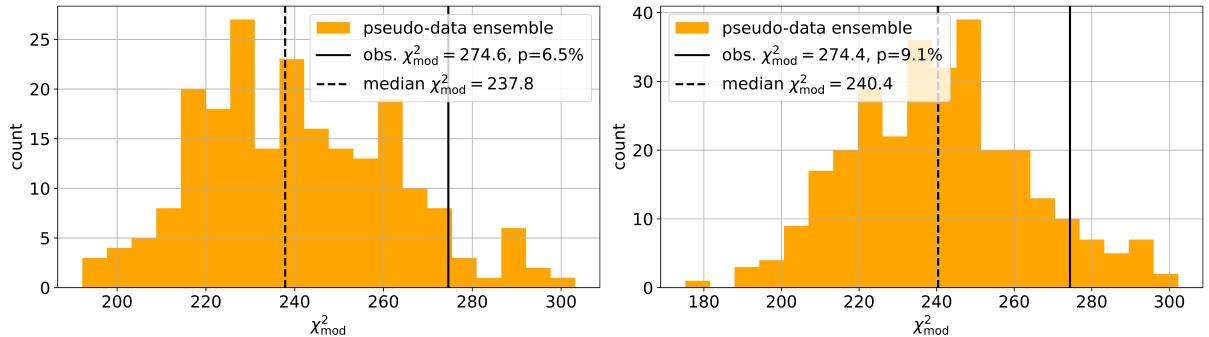


Figure B.2: Observed fit TS and TS distribution from pseudo-data trials for the 0.3 GeV (left) and 1.0 GeV (right) mass set.

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 5).
- [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 5, 7, 8).
- [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 6).
- [5] C. Argüelles. <https://github.com/arguelles/NuXSplMkr> (cited on page 6).
- [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 6).
- [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 7).