UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

vDoBe — A Python tool for neutrinoless double beta decay

Scholer, O.; de Vries, J.; Graf, L.

DOI
10.1007/JHEP08(2023)043

Publication date
2023

Document Version
Final published version

Published in
The Journal of High Energy Physics

License
CC BY

Link to publication

Citation for published version (APA):
Scholer, O., de Vries, J., & Graf, L. (2023). vDoBe — A Python tool for neutrinoless double beta decay. *The Journal of High Energy Physics*, *2023*(8), Article 43. https://doi.org/10.1007/JHEP08(2023)043

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:18 Nov 2026

Published for SISSA by Springer

Received: *April 24, 2023*
Accepted: *July 17, 2023*
Published: *August 9, 2023*

νDoBe — A Python tool for neutrinoless double beta decay

Oliver Scholer,[a] **Jordy de Vries**[b,c] **and Lukáš Gráf**[d,e]

[a] *Max-Planck-Institut für Kernphysik,*
Saupfercheckweg 1, 69117 Heidelberg, Germany
[b] *Institute for Theoretical Physics Amsterdam and Delta Institute for Theoretical Physics, University of Amsterdam,*
Science Park 904, 1098 XH Amsterdam, The Netherlands
[c] *Nikhef, Theory Group,*
Science Park 105, 1098 XG, Amsterdam, The Netherlands
[d] *Department of Physics, University of California,*
Berkeley, California 94720, U.S.A.
[e] *Department of Physics, University of California,*
San Diego, La Jolla, CA 92093-0319, U.S.A.

E-mail: scholer@mpi-hd.mpg.de, j.devries4@uva.nl, lukas.graf@berkeley.edu

Abstract: We present νDoBe, a Python tool for the computation of neutrinoless double beta decay ($0\nu\beta\beta$) rates in terms of lepton-number-violating operators in the Standard Model Effective Field Theory (SMEFT). The tool can be used for automated calculations of $0\nu\beta\beta$ rates, electron spectra and angular correlations for all isotopes of experimental interest, for lepton-number-violating operators up to and including dimension 9. The tool takes care of renormalization-group running to lower energies and provides the matching to the low-energy effective field theory and, at lower scales, to a chiral effective field theory description of $0\nu\beta\beta$ rates. The user can specify different sets of nuclear matrix elements from various many-body methods and hadronic low-energy constants. The tool can be used to quickly generate analytical and numerical expressions for $0\nu\beta\beta$ rates and to generate a large variety of plots. In this work, we provide examples of possible use along with a detailed code documentation. The code can be accessed through:
GitHub: https://github.com/OScholer/nudobe
Online User-Interface: https://oscholer-nudobe-streamlit-4foz22.streamlit.app/

Keywords: Automation, Baryon/Lepton Number Violation, SMEFT, Chiral Lagrangian

ArXiv ePrint: 2304.05415

Open Access, © The Authors.
Article funded by SCOAP^3.

https://doi.org/10.1007/JHEP08(2023)043

JHEP08(2023)043

Contents

1 Introduction 1

2 Physics formalism — The $0\nu\beta\beta$ master formula framework 4

3 Examples of νDoBe applications 4
- 3.1 An analysis of recent KamLAND-Zen limits 5
  - 3.1.1 The light-neutrino-exchange mechanism 5
  - 3.1.2 Higher dimensional mechanisms 7
- 3.2 The minimal left-right symmetric model 8
- 3.3 A leptoquark mechanism 13

4 Conclusion & outlook 15
- 4.1 Online tool 15

5 Documentation: νDoBe— A Python tool for neutrinoless double beta decay 16
- 5.1 List of functions and classes 16
- 5.2 Installation 17
  - 5.2.1 Requirements 17
  - 5.2.2 Setting up νDoBe 17
- 5.3 Parameters and constants 18
  - 5.3.1 Physical constants 18
  - 5.3.2 Units 18
  - 5.3.3 Nuclear matrix elements 19
  - 5.3.4 Phase-space factors 19
  - 5.3.5 Low energy constants 19
- 5.4 Setting up a model 21
  - 5.4.1 EFT model classes 21
  - 5.4.2 Importing the model classes 23
  - 5.4.3 Initiating a model 23
  - 5.4.4 List of Wilson coefficients 24
  - 5.4.5 Example: light neutrino exchange mechanism in LEFT 24
  - 5.4.6 RGE running 25
- 5.5 Getting the half-life 27
- 5.6 Decay-rate formula 29
- 5.7 Plot half-lives — varying a single Wilson coefficient 33
  - 5.7.1 Line-plots 33
  - 5.7.2 Scatter plots 38
- 5.8 Half-life ratios 41
- 5.9 Phase-space observables 45

JHEP08(2023)043

5.10 Limits on Wilson coefficients 50
5.10.1 Single operator dominance 50
5.10.2 Two-operator scenario 55

A List of operators 61

B Operator matching 64

C Phase space factors 66
C.1 Calculation of PSFs 66
C.2 Related observables 66

D Nuclear matrix elements 67

1 Introduction

Neutrinoless double beta decay ($0\nu\beta\beta$) searches are the most sensitive probe of the violation of lepton number (LNV). The process takes place in a nucleus and converts two neutrons to two protons and two electrons, but no outgoing anti-neutrinos. So far only upper limits have been set on $0\nu\beta\beta$ rates [1–11], but next-generation experiments hope to make a first detection by probing $0\nu\beta\beta$ lifetimes reaching 10^{27}–10^{28} years [12–16]. A nonzero signal would have profound implications. It would imply lepton number is violated by two units. Following the famous black-box theorem [17] a positive observation of $0\nu\beta\beta$ would indicate that neutrinos are indeed Majorana mass eigenstates. Additionally, it would give important clues towards the mechanism of the neutrino mass and also be a strong hint for leptogenesis scenarios to explain the absence of anti-matter in our universe. Detailed reviews about $0\nu\beta\beta$ can be found in refs. [18–20].

While tremendous experimental effort is going towards the first $0\nu\beta\beta$ detection, we must keep in mind that it is a complicated process involving particle, hadronic, nuclear, and atomic physics. The interpretation of a signal (or lack thereof) requires care. First of all, even if a nonzero decay rate is measured this does not immediately point towards the underlying source of LNV. Typically, $0\nu\beta\beta$ experiments are interpreted in terms of the effective neutrinos mass, $m_{\beta\beta}$, that enters through the exchange of light Majorana neutrinos (either active or potential sterile neutrinos). In a broad class of BSM models there can be competing mechanisms from the exchange of heavy particles such as heavy right-handed neutrinos or double charged scalars [21, 22]. Studies of $0\nu\beta\beta$-decay in multiple isotopes can be utilized to identify or reject certain types of BSM models [23–26].

As $0\nu\beta\beta$ decay is a low-energy process, the typical Q value of the reaction is a few MeV, it can be efficiently described through the use of effective field theory (EFT) techniques. In particular, EFTs provide a useful method to systematically categorize and order the possible LNV mechanisms [27–32]. If the origin of LNV happens at scales well above the scale of $0\nu\beta\beta$ decay, this allows for a model-independent description. EFT techniques are

JHEP08(2023)043

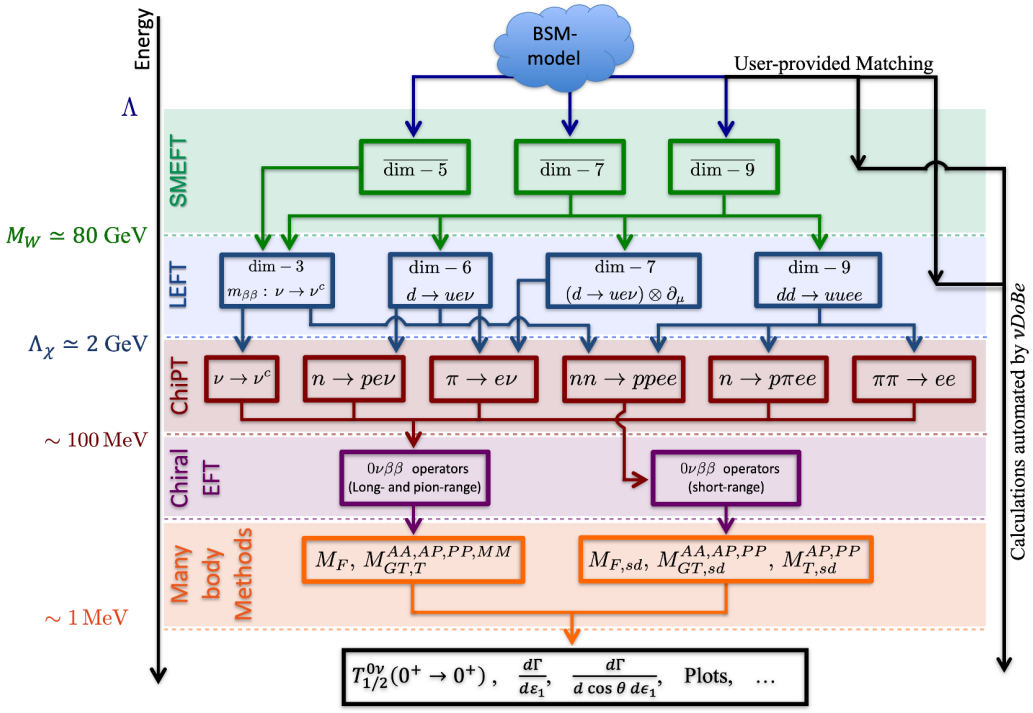


Figure 1. The EFT approach applied in νDoBe. Given a certain lepton number violating BSM-model, all $0\nu\beta\beta$ observables of interest can be calculated by following a subsequent chain of EFTs. In νDoBe the user is required to provide the matching conditions at either SMEFT or LEFT level. The calculations of $0\nu\beta\beta$ observables, plots, etc. are then automated by νDoBe. Figure modified from ref. [32].

also very powerful in the description of the associated hadronic and nuclear physics that is required to connect data to the underlying LNV mechanism that is typically written at the level of elementary particles.

νDoBe is based on an end-to-end EFT framework, from particle to nuclear physics, that has been developed in refs. [31, 32] and is illustrated in figure 1. Assuming that LNV originates at scales well above the electroweak scale,[1] $v \simeq 246\,\text{GeV}$, possible LNV mechanisms can be described by local effective LNV operators among Standard Model fields. The resulting EFT, often called the Standard Model Effective Field Theory (SMEFT), contains an infinite number of operators that can be ordered by their dimension. The higher the dimension of the operator, the more suppressed its low-energy effects are by additional powers of Q/Λ where Q is a low-energy scale and $\Lambda \gg v$ the scale where the SMEFT operators are generated. LNV operators appear with odd dimension [34] and νDoBe includes operators with dimension up-to-and-including dimension nine [35–37].

The SMEFT operators are then evolved from Λ to the electroweak scale, where heavy Standard Model fields (W, Z, Higgs, top) are integrated out and the theory is matched to an EFT that is often-called the low-energy EFT (LEFT). The LEFT operators are then

[1] This excludes the possibility of light sterile neutrinos which require a modified framework [33]. νDoBe cannot (yet) handle such scenarios.

JHEP08(2023)043

evolved to the GeV scale and matched to hadronic LNV operators by using chiral EFT which provides an expansion in m_π/Λ_χ where m_π denotes the pion mass, around 100 MeV, and $\Lambda_\chi \sim 2\,\text{GeV}$, the chiral-symmetry-breaking scale. The chiral perturbation theory Lagrangian is used to calculate the so-called neutrino transition operator that mitigates $nn \to pp + ee$ transitions. This step is non-trivial as strong interactions among nucleons lead to deviations from naive dimensional analysis expectations typically used to power count different contributions [38, 39]. The neutrino transition operator is then inserted into nuclear wave functions to obtain the relevant nuclear matrix elements (NMEs) that can be combined with atomic calculations to obtain $0\nu\beta\beta$ decay rates.

The EFT approach calculates $0\nu\beta\beta$ decay rates in a systematic expansion $(v/\Lambda)^\alpha(\Lambda_\chi/v)^\beta(m_\pi/\Lambda_\chi)^\gamma$, where α, β, and γ are exponents that depend on the LNV mechanism under consideration. The decay rate is expressed in terms of phase space factors (PSFs), NMEs, low-energy constants (LECs), evolution factors, and the Wilson coefficient of the SMEFT operators, as outlined in detail in refs. [31, 32]. The actual procedure is, however, somewhat challenging and, frankly, tedious for the expert and non-expert alike because of the large number of steps and the sizeable number of QCD, nuclear, and atomic matrix elements. νDoBE is constructed to automate this procedure: the user chooses which LNV SMEFT operators appear at the high-energy scale Λ (or alternatively which LEFT operators at the electroweak scale), and the tool presents the resulting $0\nu\beta\beta$ decay observables. In this way, any high-scale model of LNV can be quickly analyzed.

In this work we describe νDoBe (Neutrinoless DOuble-BEta calculator) in detail and give some explicit examples of its use. Before doing so let us list the most important applications of νDoBe:

- An automatic calculation of $0\nu\beta\beta$ decay rates and electron kinematics of all isotopes of experimental interest in terms of LNV SMEFT operators up to and including dimension nine. Alternatively, the decay rate can be expressed in terms of LNV LEFT operators.
- The tool computes differential decays rates based on state-of-the-art information regarding renormalization-group factors, hadronic and nuclear matrix elements, and atomic phase space factors.
- It is possible to choose various sets of NMEs (based on different many-body nuclear-structure methods) to assess the theoretical uncertainty of the predictions. Users can define their own set of NMEs if necessary.
- The expression for $0\nu\beta\beta$-decay rates includes recently identified short-range contributions associated to hard-neutrino-exchange processes [38, 39]. The code uses current best estimates for the associated LECs but different values can be specified.
- νDoBe can be used to generate interesting plots of half-lives or various different distributions as function of neutrino masses or LNV Wilson coefficients. This can be used to quickly analyse specific BSM scenarios.

JHEP08(2023)043

- For a given BSM model or a set of Wilson coefficients νDoBe can generate analytical expressions of the decay half-life and output them in latex (or html) form.

- Given an experimental bound on the half-life of certain isotope, νDoBe can be used to extract limits on the effective Majorana neutrino mass or higher-dimensional LNV LEFT or SMEFT operators.

2 Physics formalism — The $0\nu\beta\beta$ master formula framework

The effects of LNV physics appearing at some high-energy scale $\Lambda \gg M_W$ on low-energy phenomena such as $0\nu\beta\beta$ can be described by a chain of EFTs. On the high-energy end of this chain we employ the Standard Model EFT (SMEFT) [36, 37, 40–45] describing the effects of new physics above the electroweak scale. At the scale of electroweak symmetry breaking SMEFT is matched onto the low-energy EFT (LEFT) [46–50] respecting the $SU(3)_C \times U(1)_{em}$ gauge group. At lower scales we match subsequently onto lepton-extended chiral EFT to incorporate the non-perturbative nature of QCD below $\Lambda_\chi \sim 2\,\text{GeV}$. νDoBe follows the procedure developed in refs. [31, 32] which is illustrated in figure 1. Below, we present the relevant expressions and how they are used in νDoBe and refer to [31, 32] for more details.

The $0\nu\beta\beta$ half-life equation can be written in very short-hand notation

$$T_{1/2}^{-1} = g_A^4 \sum_k G_{0k} |\mathcal{A}_k(\{C_i\})|^2, \tag{2.1}$$

where conventionally g_A^4, with $g_A \simeq 1.27$, is factored out. The G_{0k} denote atomic phase space factors. Most of the physics is captured by the so-called subamplitudes $\mathcal{A}_k(\{C_i\})$ which depend on NMEs, LECs, and the Wilson coefficients of LNV higher-dimensional operators labeled by $\{C_i\}$. The full expressions of the subamplitudes are rather lengthy and can be found in ref. [32], but we provide some details in appendices. Specifically, appendix A gives the full lists of SMEFT and LEFT operators included in νDoBe. The matching of SMEFT to LEFT is summarized in appendix B where the matching of dim-9 LNV SMEFT operators to LEFT is performed for the first time. Appendices C and D discuss the phase space factors and NMEs, respectively.

3 Examples of νDoBe applications

We now present three analyses revisited from the literature and re-performed with νDoBe to illustrate the practical use of the tool. We begin by analyzing the KamLAND-ZEN upper limit on the $0\nu\beta\beta$ decay of ^{136}Xe and the implications for the effective neutrino Majorana mass. We then turn to the analysis of $0\nu\beta\beta$ in specific beyond-the-Standard Model models including the minimal left-right symmetric extension to the Standard Model as well as possible realizations of leptoquarks.

JHEP08(2023)043

	M_ν	$M_{F,sd}$	$M_{F,sd}/M_\nu$	$m_{\beta\beta}$ [meV]	$\tilde{m}_{\beta\beta}$ [meV]
QRPA [51]*	3.009	$-61.8\frac{m_e m_p}{m_\pi^2}$	−0.51	57	27
QRPA [52]	3.384	–	–	51	35
QRPA [53]	2.460	–	–	70	48
QRPA [54]	1.89	–	–	91	63
QRPA [55]	1.18	$-28.8\frac{m_e m_p}{m_\pi^2}$	−0.60	146	62
EDF [56]	4.773	–	–	36	25
EDF [57]	4.32	–	–	40	28
EDF [58]	4.20	–	–	41	28
IBM2 [59]*	3.387	$-29.8\frac{m_e m_p}{m_\pi^2}$	−0.22	51	34
IBM2 [60]	3.05	$-29.7\frac{m_e m_p}{m_\pi^2}$	−0.24	57	37
SM [61]	2.39	–	–	72	50
SM [62]	1.76	–	–	98	68
SM [63]	1.77	–	–	98	67
SM [64]*	2.45	$-52\frac{m_e m_p}{m_\pi^2}$	−0.52	71	32

Table 1. Different nuclear matrix elements in ^{136}Xe. The NMEs are given assuming $g_A \sim 1.271$. Appropriate rescaling has been applied to the IBM2 NMEs. The sets labeled with an asterisk (*) are pre-installed in νDoBe. In the columns labeled with $m_{\beta\beta}$ and $\tilde{m}_{\beta\beta}$ we show the resulting limits on the effective Majorana mass for each set of NMEs without ($m_{\beta\beta}$) and including ($\tilde{m}_{\beta\beta}$) the short-range contribution proportional to g_ν^{NN}.

3.1 An analysis of recent KamLAND-Zen limits

3.1.1 The light-neutrino-exchange mechanism

Recently, the KamLAND-Zen experiment obtained a new lower limit on the half-life in ^{136}Xe of 2.3×10^{26} yr [10]. We will analyze the resulting limits on the effective Majorana mass $m_{\beta\beta}$ using νDoBe. In particular, we will apply different sets of NMEs and study the potential impact of the recently introduced short-range contributions originating from hard neutrino-exchange [38, 39] which were not included in ref. [10]. We briefly describe the necessary inputs to the code here while a Jupyter notebook with all details is accessible on GitHub.

If only the standard mass mechanism is considered, the half-life is expressed as

$$T_{1/2}^{-1} = g_A^4 \left|\frac{m_{\beta\beta}}{m_e}\right|^2 G_{01} V_{ud}^4 \left|\left(-\frac{1}{g_A^2} M_F + M_{GT} + M_T + 2\frac{m_\pi^2 g_\nu^{NN}}{g_A^2} M_{F,sd}\right)\right|^2 , \tag{3.1}$$

where the various M_i denote NMEs, while g_ν^{NN} is an LEC associated with hard-neutrino exchange. To replicate the results from [10] we apply the same value for the phase space factor $V_{ud}^4 G_{01}(^{136}\text{Xe}) = 1.458 \times 10^{-14}\,\text{y}^{-1}$ (the factor V_{ud}^4 is included because of a slightly different definition of G_{0k}). νDoBe has the NMEs split into various components, but most

JHEP08(2023)043

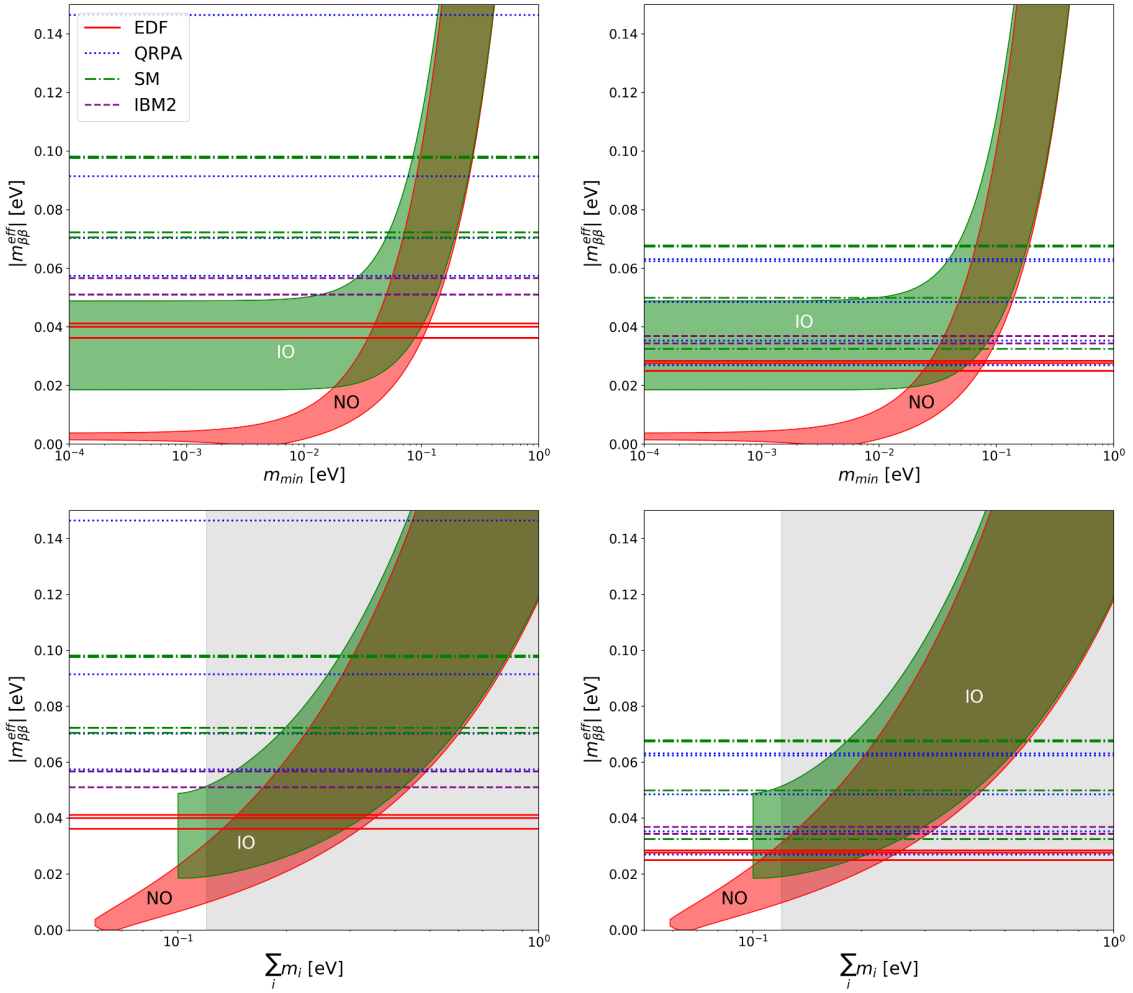


Figure 2. Limits on the effective Majorana mass $m_{\beta\beta}$ obtained from the recent half-life limit by the KamLAND-Zen experiment [10]. On the left, the limits are shown without including the short-range contribution proportional to g_ν^{NN} as in [10], while on the right we do include it (see text for details). In the lower plots we show the current limit on the sum of the neutrino masses $\sum_i m_i \lesssim 0.12\,\text{eV}$ [65] as a gray band. The allowed parameter regions for normal (NO) and inverse (IO) mass ordering are displayed as red and green bands, respectively. They are obtained by variation of the unknown Majorana phases utilizing the `plot_m_eff()` function of νDoBe.

literature is based on the effective combination

$$M_\nu = -\frac{1}{g_A} M_F + M_{GT} + M_T \,, \tag{3.2}$$

while setting $g_\nu^{NN} = 0$. In table 1 we show the different sets of NMEs studied here.[2]

[2]For some NME calculation the magnetic factor has been calculated incorrectly through $g_M = \kappa_1 g_V$, where κ_1 is the isovector anomalous magnetic moment of the nucleon, instead of the correct $g_M = (1+\kappa_1) g_V$. For more details see the appendices or refs. [31, 32]. If the individual NMEs $M_{GT,T}^{MM}$ are given explicitly in the original publication, we correct M_ν appropriately. In case that both CD Bonn and Argonne potentials are given we take the values corresponding to the CD Bonn potential.

JHEP08(2023)043

$m_{\beta\beta}$	$C_S^{(6)}$	$C_T^{(6)}$	$C_{VL}^{(6)}$	$C_{VR}^{(6)}$	$C_V^{(7)}$	$C_{S1}^{(9)}$	$C_{S2}^{(9)}$	$C_{S3}^{(9)}$	$C_{S4}^{(9)}$	$C_{S5}^{(9)}$	$C_V^{(9)}$	$C_{\tilde{V}}^{(9)}$
$m_{\beta\beta}$	$C_{SL}^{(6)}$	$C_T^{(6)}$	$C_{VL}^{(6)}$	$C_{VR}^{(6)}$	$C_{VL}^{(7)}$	$C_{1L}^{(9)}$	$C_{2L}^{(9)}$	$C_{3L}^{(9)}$	$C_{4L}^{(9)}$	$C_{5L}^{(9)}$	$C_6^{(9)}$	$C_7^{(9)}$
	$C_{SR}^{(6)}$				$C_{VR}^{(7)}$	$C_{1R}^{(9)}$	$C_{2R}^{(9)}$	$C_{3R}^{(9)}$	$C_{4R}^{(9)}$	$C_{5R}^{(9)}$	$C_8^{(9)}$	$C_9^{(9)}$
						$C_{1L}^{(9)\prime}$	$C_{2L}^{(9)\prime}$	$C_{3L}^{(9)\prime}$			$C_6^{(9)\prime}$	$C_7^{(9)\prime}$
						$C_{1R}^{(9)\prime}$	$C_{2R}^{(9)\prime}$	$C_{3R}^{(9)\prime}$			$C_8^{(9)\prime}$	$C_9^{(9)\prime}$

Table 2. Groups of LEFT operators that result in the same half-lives.

In general, to apply νDoBe we need the individual NMEs $M_{F,GT,T,Fsd}$. However, often only the combined NME $M_\nu = -\frac{1}{g_A}M_F + M_{GT} + M_T$ is given in the literature. Within νDoBe we can work around this by simply defining $M_{GT} = M_\nu$ while putting all other NMEs (including $M_{F,sd}$) to zero. By utilizing the internal `get_limits()` function we then arrive at the limits on $m_{\beta\beta}$ in the range $[36, 146]$ meV. The individual limits on $m_{\beta\beta}$ derived from each NME method are listed in table 1 and the left column of figure 2. Whenever NMEs calculated using both CD Bonn and Argonne nuclear potentials are available we choose to show the CD-Bonn ones. If, instead, the Argonne potential NMEs are considered the upper limit moves to 156 meV as in [10].

Now we want to study the effect of the additional short-range contribution proportional to g_ν^{NN} which has not been considered in [10]. For those NME calculations where the corresponding short-range NME $M_{F,sd}$ is available we simply add this contribution. However, for many NMEs no short-range contributions have been calculated. Here, we take a very simplistic approach by approximating the short-range NME $M_{F,sd}$ from the ratios $M_{F,sd}/M_\nu$ found in those NME calculations with $M_{F,sd}$ available. In ^{136}Xe we find that $|M_{F,sd}/M_\nu| > 0.2$, hence, we take $M_{F,sd} = -\frac{1}{5}M_\nu$ as a lower limit approximation of the resulting half-life (Note, again, that g_ν^{NN} is negative and, hence, the contribution of a negative $M_{F,sd}$ is constructive). In this approximation we arrive at limits on $m_{\beta\beta}$ in the range $[25, 68]$ meV which are more stringent, illustrating the importance of fixing the short-distance contributions. The limits from all different NMEs with short-range contributions are shown in the right column of figure 2 and table 1.

We stress that these limits are only indicative as the short-distance contributions must be calculated for each nuclear many-body method consistently with the long-distance contributions, see section (5.3.5) for a more detailed discussion. This has recently been done in refs. [66, 67].

3.1.2 Higher dimensional mechanisms

LEFT: we can use νDoBe to put limits on the higher-dimensional LNV operators. Considering operators from the low-energy EFT we can put these into 13 different groups of operators that result in the same half-lives (see table 5 in appendix A) because of parity conservation in QCD. For convenience we summarize these operator groups here again in table 2. We require the corresponding NMEs to calculate the limits on the higher dimensional ($d \geq 6$) operators of interest, which means we can only use a few NME calculation methods namely those from the shell model [64], QRPA [51] and IBM2 [59]. The resulting

JHEP08(2023)043

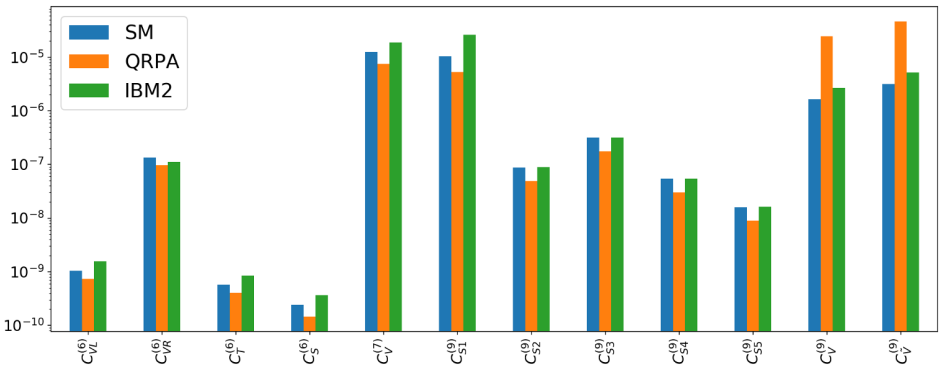


Figure 3. Limits on the higher dimensional ($d \geq 6$) LEFT operators obtained from the recent limit on the half-life of ^{136}Xe by the KamLAND-Zen experiment in combination with different NMEs.

limits on the higher dimensional operators are given in figure 3. The limits are extracted at the SMEFT $\rightarrow$ LEFT matching scale m_W and acquire a broad range of values from $\sim 10^{-10}$ (for $C_S^{(6)}$) to $\sim 10^{-5}$ (for $C_V^{(7)}$, $C_{S1}^{(9)}$, $C_V^{(9)}$, and $C_{\tilde{V}}^{(9)}$), thus illustrating the fact that different operators can have very different $0\nu\beta\beta$ efficiencies.

We can estimate the scale of new physics Λ through

$$\Lambda_i \simeq \frac{v}{C_i^{1/(d-4)}} \tag{3.3}$$

with $d \in [6, 7, 9]$ being the operator's LEFT dimension and the Higgs vacuum expectation value (vev) $v \simeq 246\,\text{GeV}$. This translates to limits on the scale of new physics in the range of $\sim 2\,\text{TeV}$–$20\,\text{PeV}$.

SMEFT: alternatively, we can use the new limit on the half-life in ^{136}Xe from KamLAND-Zen to provide limits on the dimensionfull Wilson coefficients of the SMEFT operators. These limits are given in figure 4 and correspond to scales in the range of $\sim 1\,\text{TeV}$–$400\,\text{TeV}$ For more details, we refer to the provided Jupyter notebook on GitHub as well as section 5.10.1.

3.2 The minimal left-right symmetric model

The minimal left-right symmetric model (mLRSM) [68–71] is a well-studied extension to the Standard Model. It enlarges the Standard Model gauge group by adding the right-handed $SU(2)$; hence, the mLRSM gauge group reads $SU(3)_C \times SU(2)_L \times SU(2)_R \times U(1)_{B-L}$. At the same time, this additional symmetry requires the existence of new fermionic and bosonic degrees of freedom. Typically, the mLRSM incorporates two scalar triplets $\Delta_L \in (1, 3, 1, 2)$ and $\Delta_R \in (1, 1, 3, 2)$ and a scalar bidoublet $\Phi \in (1, 2, 2*, 0)$. Fermions are grouped into left- and right-handed $SU(2)_{L,R}$ doublets requiring the introduction of right-

JHEP08(2023)043

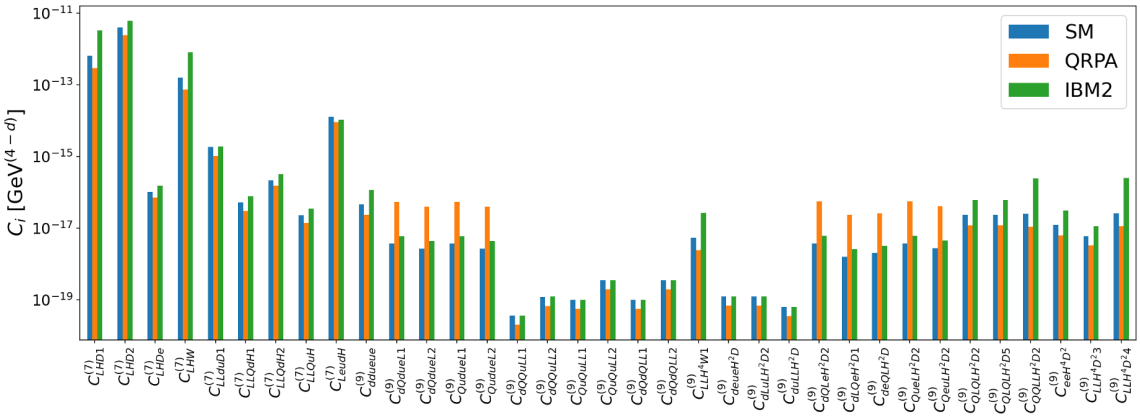


Figure 4. Limits on the higher dimensional ($d \geq 7$) SMEFT operators obtained from the recent KamLAND-Zen results assuming different NMEs.

handed neutrinos ν_R

$$L_L = \begin{pmatrix} \nu_L \\ e_L \end{pmatrix} \in (1, 2, 1, -1)\,, \qquad Q_L = \begin{pmatrix} u_L \\ d_L \end{pmatrix} \in (3, 2, 1, 1/3)\,, \tag{3.4}$$

$$L_R = \begin{pmatrix} \nu_R \\ e_R \end{pmatrix} \in (1, 1, 2, -1)\,, \qquad Q_R = \begin{pmatrix} u_R \\ d_R \end{pmatrix} \in (3, 1, 2, 1/3)\ . \tag{3.5}$$

In this section, we revisit the mLRSM as studied in [25, 32] (conventions are equal to [25]) utilizing νDoBe and the shell model NMEs [64] provided therein. Lepton number violation in the mLRSM is introduced via Yukawa interactions,

$$\mathcal{L}_y = \sum_{ij} \left[Y^l_{ij} \overline{L_{Li}} \Phi L_{R,j} + \tilde{Y}^l_{ij} \overline{L_{Li}} \tilde{\Phi} L_{R,j} + Y^L_{ij} L^T_{L,i} C i\tau_2 \Delta_L L_{L,j} + Y^{R\dagger}_{ij} L^T_{R,i} C i\tau_2 \Delta_R L_{R,j} \right] + \text{h.c.} \tag{3.6}$$

When the neutral components of the scalar multiplets acquire non-zero vacuum expectation values,

$$\langle \Phi \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} \kappa & 0 \\ 0 & \kappa' e^{i\alpha} \end{pmatrix}, \qquad \langle \Delta_L \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 \\ v_L e^{i\theta_L} & 0 \end{pmatrix}, \qquad \langle \Delta_R \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 \\ v_R & 0 \end{pmatrix}, \tag{3.7}$$

they give rise to neutrino mass matrices

$$\begin{aligned} M^\nu_{D,ij} &= \frac{1}{\sqrt{2}} \left[Y^l_{ij} \kappa + \tilde{Y}^l_{ij} \kappa' \exp -i\alpha \right], \\ {M^\nu_{L,ij}}^\dagger &= \sqrt{2} Y^L_{ij} v_L \exp i\theta_L, \qquad\qquad M^\nu_{R,ij} = \sqrt{2} Y^R_{ij} v_R\ . \end{aligned} \tag{3.8}$$

The ratio of the bidoublet's vevs $\xi = \frac{\kappa'}{\kappa}$ describes the mixing between the left-handed and right-handed W-bosons. When the right-handed triplet gains a non-zero vev, the mLRSM gauge group is broken down to the Standard Model gauge structure. Matching the mLRSM

JHEP08(2023)043

onto the SMEFT results in the following LNV Lagrangian,

$$\begin{aligned}\mathcal{L}_{\Delta L=2} =& C^{(5)}\left(\left(L^T C i\tau_2 H\right)\left(\tilde{H}^\dagger L\right)\right)\\ &+\left(L^T\gamma^\mu e_R\right) i\tau_2 H\left[C^{(7)}_{Leu\bar{d}\Phi}\overline{d_R}\gamma_\mu u_R + C^{(7)}_{L\Phi De}H^T i\tau_2(D_\mu\Phi_{SM})\right]\\ &+\overline{e_R}e_R^c\left[C^{(9)}_{eeud}\overline{u_R}\gamma^\mu d_R\overline{u_R}\gamma_\mu d_R + C^{(9)}_{ee\Phi ud}\overline{u_R}\gamma^\mu d_R\left([iD_\mu H]^\dagger\,\tilde{H}\right)\right.\\ &\left.+\,C^{(9)}_{ee\Phi D}\left([iD_\mu H]^\dagger\,\tilde{H}\right)^2\right],\end{aligned} \tag{3.9}$$

where H is the Standard Model Higgs with the vev $v=\sqrt{\kappa^2+\kappa'^2}$ and the SMEFT Wilson coefficients are given by

$$\begin{aligned}C^{(5)} &= \frac{1}{v^2}\left(M_D^{\nu\,T}M_R^{\nu\,-1}M_D^\nu - M_L^\nu\right),\\ C^{(7)}_{Leu\bar{d}\Phi} &= \frac{\sqrt{2}}{v}\frac{1}{v_R^2}\left(V_R^{ud}\right)^*\left(M_D^{\nu T}M_R^{\nu\,-1}\right)_{ee}, \qquad C^{(7)}_{L\Phi De} = \frac{2i\xi\exp i\alpha}{(1+\xi^2)\,V_R^{ud*}}C^{(7)}_{Leu\bar{d}\Phi},\\ C^{(9)}_{eeud} &= -\frac{1}{2v_R^4}V_R^{ud\,2}\left[\left(M_R^{\nu\dagger}\right)^{-1}+\frac{2}{m^2_{\Delta_R}}M_R^\nu\right], \quad C^{(9)}_{ee\Phi ud} = -4\frac{\xi\exp -i\alpha}{(1+\xi^2)\,V_R^{ud}}C^{(9)}_{eeud},\\ C^{(9)}_{ee\Phi D} &= 4\frac{\xi^2\exp -2i\alpha}{(1+\xi^2)^2\,V_R^{ud\,2}}C^{(9)}_{eeud}.\end{aligned} \tag{3.10}$$

The Wilson coefficients are completely described by fixing the vevs of the triplet scalars $v_{L,R}$, the heavy right-handed triplet mass m_{Δ_R}, the heavy neutrino masses $m_{\nu_{R,i}}, i=1\ldots3$, the lightest active neutrino mass $m_{\nu_{\min}}$, the neutrino mixing matrices $U_{L,R}$, the complex vev phases θ_L and α as well as the left-right mixing parameter ξ.

To apply νDoBe, we need to translate these Wilson coefficients to the correct operator basis (see appendix A) via

$$\begin{aligned}C^{(5)}_{LH} &= \left(C^{(5)}\right)_{ee},\\ C^{(7)}_{LeudH} &= C^{(7)}_{Leu\bar{d}\Phi},\\ C^{(7)}_{LHDe} &= C^{(7)}_{L\Phi De},\\ C^{(9)}_{ddueue} &= 4\left(C^{(9)}_{eeud}\right)^*,\\ C^{(9)}_{deueH^2D} &= -2\left(C^{(9)}_{ee\Phi ud}\right)^*,\\ C^{(9)}_{eeH^4D} &= -\left(C^{(9)}_{ee\Phi D}\right)^*.\end{aligned} \tag{3.11}$$

With the help of νDoBe we can use these Wilson coefficients to generate a SMEFT model class to study different parametric scenarios of mLRSM realizations in Python. As an example, we revisit four different realizations of the mLRSM studied in ref. [32], given by

$$\begin{aligned}&m_{\nu_{R1}} = 10\,\text{TeV}, \quad m_{\nu_{R2}} = 12\,\text{TeV}, \quad m_{\nu_{R3}} = 13\,\text{TeV}\\ &m_{\Delta_R} = 4\,\text{TeV}, \qquad v_R = 10\,\text{TeV}, \qquad V_{ud}^R = V_{ud}^L, \qquad U_R = U_L,\end{aligned} \tag{3.12}$$

JHEP08(2023)043

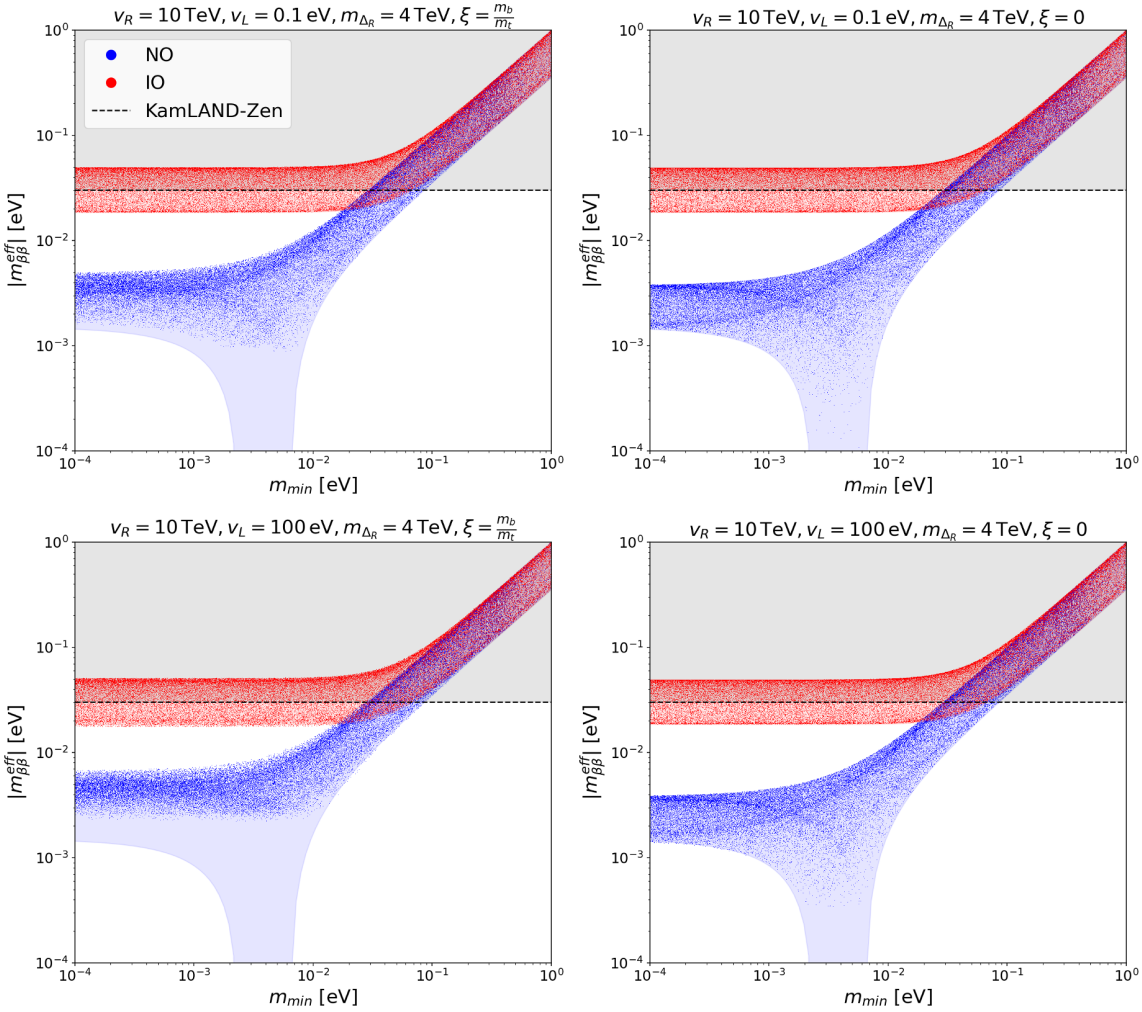


Figure 5. The effective Majorana mass $m_{\beta\beta}^{\text{eff}}$ in dependence on the minimal light neutrino mass for the four parameter settings listed in (3.13). The shaded areas show the Majorana mass $m_{\beta\beta}$ arising from the light-neutrino-exchange mechanism alone, while the scattered points show the resulting effective mass for the different mLRSM realizations when varying the unknown Majorana phases, θ_L and α. Both the normal (blue) and the inverted (red) mass orderings are displayed. The gray area shows the current limit given by KamLAND-Zen.

and by setting the left-handed triplets vev v_L and the ratio of the bidoublet vevs ξ to one of the following values

$$\begin{aligned}
&\text{Model 1}: \quad v_L = 0.1\,\text{eV}\,, \quad \xi = \frac{m_b}{m_t}\,,\\
&\text{Model 2}: \quad v_L = 0.1\,\text{eV}\,, \quad \xi = 0\,,\\
&\text{Model 3}: \quad v_L = 100\,\text{eV}\,, \quad \xi = \frac{m_b}{m_t}\,,\\
&\text{Model 4}: \quad v_L = 100\,\text{eV}\,, \quad \xi = 0\,.
\end{aligned} \tag{3.13}$$

JHEP08(2023)043

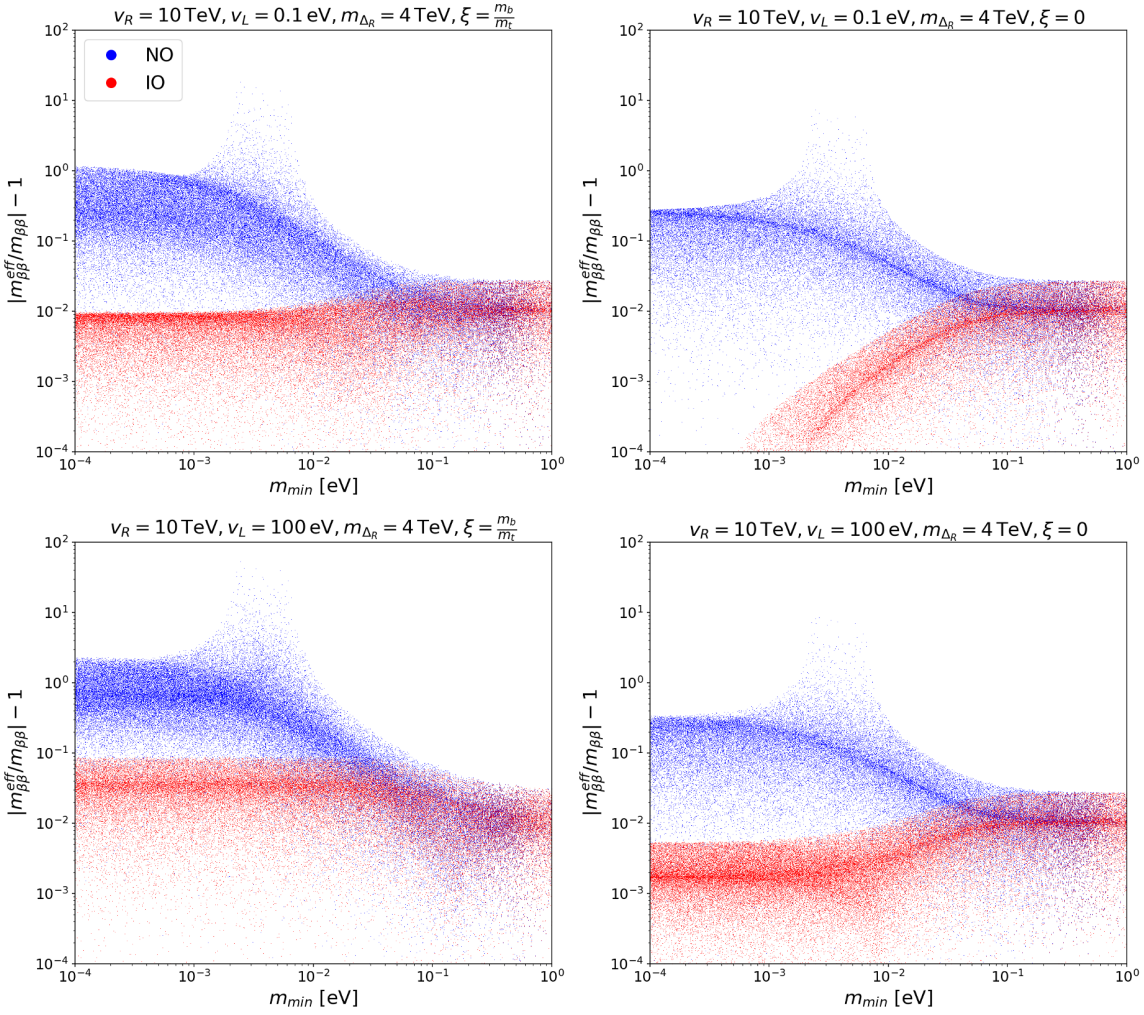


Figure 6. Similar to figure 5 but normalized to the standard mass mechanism induced by $m_{\beta\beta}$.

In figure 5 we show the evolution of the effective neutrino mass parameter,

$$m^{\text{eff}}_{\beta\beta} = \frac{m_e}{g_A^2 V_{ud}^2 \mathcal{M}_3^{(\nu)} G_{01}^{1/2}} T_{1/2}^{-1/2}, \tag{3.14}$$

where $\text{M}_3^{(\nu)}$ is the NME for the light-neutrino-exchange mechanism (LνEM)(see [32]). Figure 6 shows similar information but now we normalized to $m_{\beta\beta}$, the effective mass in the LνEM. The scattered points show variations of the unknown Majorana phases in the neutrino mixing matrix as well as θ_L and α. We observe that in all four parameter settings the case of inverted mass ordering is hardly influenced by the additional higher dimensional interactions introduced in the mLRSM. This would change for smaller values of the left-right symmetry breaking scale. On the contrary, for normal mass ordering the higher dimensional $d \geq 7$ contributions do increase the expected decay rate when compared to the LνEM. In addition, the case of normal mass ordering does no longer show the typical funnel.

JHEP08(2023)043

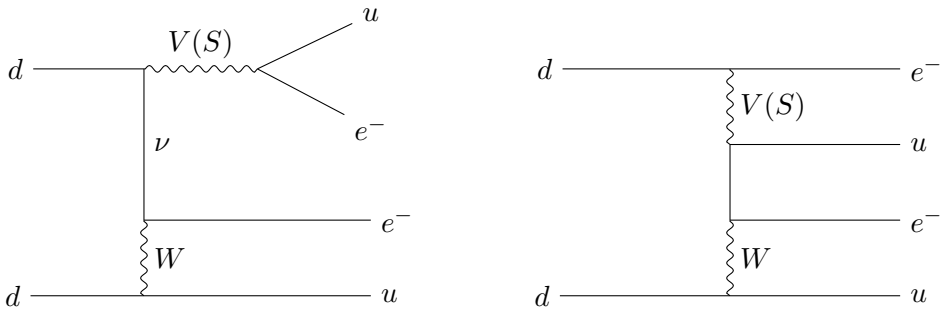


Figure 7. Feynman diagrams of the vector (V) and scalar (S) leptoquark interactions contributing to $0\nu\beta\beta$.

We refrain from a more detailed analysis of the parameter space as our main goal is to illustrate how νDoBE can be readily applied for $0\nu\beta\beta$ analyses of complicated models. Again, a detailed Jupyter notebook is accessible via GitHub.

3.3 A leptoquark mechanism

In this section we revisit the $0\nu\beta\beta$ decay induced by new leptoquark (LQ) interactions first studied in ref. [72]. Assuming the Standard Model without right-handed neutrinos, one can add up to 10 scalar or vector LQs with non-trivial couplings to the Standard Model at tree level. These are summarized in table 3. LQs with $Q = \pm 1/3$ and $Q = \pm 2/3$ then generate new decay channels for $0\nu\beta\beta$ decay different from the usually considered LνEM with the corresponding Feynman diagrams displayed in figure 7. The matching procedure involves a minor subtlety when compared to the previous case study of the mLRSM as the LNV interactions relevant for $0\nu\beta\beta$ decay are generated when electroweak symmetry breaking introduces non-diagonal mixing between different LQs. The analysis is more straightforward if one matches the BSM model directly onto LEFT instead of SMEFT.

After doing so, the parts of the low-energy Lagrangian relevant for $0\nu\beta\beta$ decay are given by [72]

$$\begin{aligned}\mathcal{L}_{LQ} = &[\bar{e}P_L\nu^c]\left\{\frac{\epsilon_S}{M_S^2}[\bar{u}P_Rd] + \frac{\epsilon_V}{M_V^2}[\bar{u}P_Ld]\right\} \\ &- [\bar{e}\gamma^\mu P_L\nu^c]\left\{\left(\frac{\alpha_S^R}{M_S^2} + \frac{\alpha_V^R}{M_V^2}\right)[\bar{u}\gamma_\mu P_Rd] - \sqrt{2}\left(\frac{\alpha_S^L}{M_S^2} + \frac{\alpha_V^L}{M_V^2}\right)[\bar{u}\gamma_\mu P_Ld]\right\} + \text{h.c.}.\end{aligned} \tag{3.15}$$

JHEP08(2023)043

LQ (Ω)	$SU(3)_C$	$SU(2)_L$	$U(1)_Y$	Q
S_0	3	1	-2/3	-1/3
$\tilde{S}_0$	3	1	-8/3	-4/3
$S_{1/2}$	$\overline{3}$	2	-7/3	$(-2/3, -5/3)$
$\tilde{S}_{1/2}$	$\overline{3}$	2	-1/3	$(1/3, -2/3)$
S_1	3	3	-2/3	$(2/3, -1/3, -4/3)$
V_0	$\overline{3}$	1	-4/3	-2/3
$\tilde{V}_0$	$\overline{3}$	1	-10/3	-5/3
$V_{1/2}$	3	2	-5/3	$(-1/3, -4/3)$
$\tilde{V}_{1/2}$	3	2	1/3	$(2/3, -1/3)$
V_1	$\overline{3}$	3	-4/3	$(1/3, -2/3, -5/3)$

Table 3. List of possible scalar and vector leptoquarks and their transformation properties under the Standard Model symmetries [72].

Hence, we obtain the following LEFT Wilson coefficients

$$\begin{aligned} C_{SL}^{(6)} &= \frac{v^2}{M_V^2}\epsilon_V\,, \\ C_{SR}^{(6)} &= \frac{v^2}{M_S^2}\epsilon_S, \\ C_{VL}^{(6)} &= \sqrt{2}v^2\left(\frac{\alpha_S^L}{M_S^2}+\frac{\alpha_V^L}{M_V^2}\right), \\ C_{VR}^{(6)} &= -v^2\left(\frac{\alpha_S^R}{M_S^2}+\frac{\alpha_V^R}{M_V^2}\right). \end{aligned} \tag{3.16}$$

Now we can use the `get_limits_LEFT()` function of νDoBe to evaluate the limits on the parameters $\epsilon_i, \alpha_i^{S,V}, i \in [L, R]$ imposed by the recent KamLAND-Zen results when assuming only one parameter at a time to be non-vanishing. In doing so, using the IBM2 NMEs [59] provided with νDoBe we find that

$$\begin{aligned} \epsilon_i &\lesssim 5.80\times10^{-9}\left(\frac{M_i}{1\,\text{TeV}}\right)^2, \\ \alpha_i^L &\lesssim 1.81\times10^{-8}\left(\frac{M_i}{1\,\text{TeV}}\right)^2, \\ \alpha_i^R &\lesssim 1.86\times10^{-6}\left(\frac{M_i}{1\,\text{TeV}}\right)^2. \end{aligned} \tag{3.17}$$

In comparison with the limits obtained in the original work [72] the above bounds are somewhat more stringent, which is mainly due the improved KamLAND-Zen results [10]. This aspect is partially compensated by using a different set of NMES and PSFs reflecting the improved status of theoretical calculations. A more detailed comparison with ref. [72] can be found in the provided Jupyter notebook.

JHEP08(2023)043

4 Conclusion & outlook

Neutrinoless double beta decay experiments provide the most stringent tests of lepton number violation. The experimental prospects are excellent and it will play an important role in the resolution of the neutrino mass puzzle and the search for beyond-the-Standard-Model physics in general.

That being said, $0\nu\beta\beta$ is a complicated process involving particle, nuclear, and atomic physics and its interpretation requires a great care. The main goal of νDoBe is to help the community with these difficulties. Being a low-energy process, effective field theory methods can be used efficiently to describe $0\nu\beta\beta$ for a large class of LNV scenarios stemming from possibly rich physics living at UV scales. νDoBe computes $0\nu\beta\beta$ (differential) decay rates in terms of Wilson coefficients of LNV operators constructed within the Standard Model Effective Field Theory (SMEFT). It uses the state-of-the-art renormalization-group evolution factors and QCD, nuclear, and atomic matrix elements and comes with a large number of built-in plotting and analysis tools. Naturally, νDoBe can help to easily extract limits on the $m_{\beta\beta}$ parameter associated with the LνEM, but also on all the other beyond-the-Standard-Model Wilson coefficients of interest, imposed by current and upcoming experimental results. We hope it will be used by the community to quickly and accurately analyze $0\nu\beta\beta$ predictions and constraints related to beyond-the-Standard-Model models involving lepton number violation. We encourage users of νDoBe to share their BSM analyses with the community by adding a Jupyter notebook to νDoBe's GitHub.

In the upcoming updates we aim to extend νDoBe in several directions, some of which are work in progress. Most importantly, we will include the possibility of relatively light sterile neutrinos, which means addition of effective operators arising in the SMEFT framework extended by three right-handed singlet neutrinos, so called νSMEFT. Several works have computed the impact of light sterile neutrinos on $0\nu\beta\beta$ observables, but the analysis is complicated and essentially done on a case-by-case basis [22, 33, 73–77]. An automated tool would be very helpful in this regard. In addition to the currently available approximate calculations of the phase-space factors described in appendix C we plan to add the option of employing the exact numerical solutions for the radial electron wave functions using a broader variety of nuclear potentials and including other subtle effects such as the electron screening along with and beyond the treatments adopted in refs. [78, 79]. Last but not least, we envision to incorporate predictions for and constraints from other lepton-number-violating processes, such as kaon decays.

4.1 Online tool

Finally, for quick analyses we created an online user-interface using Streamlit. The online user-interface is aimed towards delivering an easy and fast to use possibility for the most relevant use-cases like, e.g., calculating the decay observables (half-life, spectra, angular correlation) from a given model or studying limits on the different Wilson coefficients given half-life limits from experiments.

JHEP08(2023)043

Acknowledgments

The authors are grateful to Vaisakh Plakkot, Jacob Spisak and Nele Volmer for testing the code and to Wouter Dekens and Xiao-Dong Ma for helpful discussions and clarifications of encountered discrepancies. LG acknowledges support from the National Science Foundation, Grant PHY-1630782, and the Heising-Simons Foundation, Grant 2017-228. JdV acknowledges support from the Dutch Research Council (NWO) in the form of a VIDI grant.

5 Documentation: νDoBe — A Python tool for neutrinoless double beta decay

In the following we provide a detailed description of all the different functionalities of νDoBe including various explicit code examples.

5.1 List of functions and classes

1. nudobe.EFT:
   (a) `EFT.SMEFT`
   (b) `EFT.LEFT`
2. nudobe.EFT.model: (model = SMEFT or LEFT)
   (a) `model.run`
   (b) `model.half_lives`
   (c) `model.t_half`
   (d) `model.generate_formula`
   (e) `model.generate_matrix`
   (f) `model.plot_WC_variation`
   (g) `model.plot_t_half`
   (h) `model.plot_t_half_inv`
   (i) `model.plot_m_eff`
   (j) `model.plot_WC_variation_scatter`
   (k) `model.plot_t_half_scatter`
   (l) `model.plot_t_half_inv_scatter`
   (m) `model.plot_m_bb_scatter`
   (n) `model.ratios`
   (o) `model.plot_ratios`
   (p) `model.spectrum`
   (q) `model.angular_corr`

JHEP08(2023)043

(r) `model.plot_spec`

(s) `model.plot_corr`

(t) `model.get_limits`

3. nudobe.functions:

   (a) `functions.generate_formula`

   (b) `functions.generate_matrix`

   (c) `functions.get_limits_LEFT`

   (d) `functions.get_limits_SMEFT`

   (e) `functions.get_contours`

4. nudobe.plots:

   (a) `plots.limits_LEFT`

   (b) `plots.limits_SMEFT`

   (c) `plots.contours`

5.2 Installation

5.2.1 Requirements

Using νDoBe requires the following packages to be installed (the versions we used are given in [brackets], though more recent version should work, too.):

1. `NumPy` [80] [v. 1.19.2]
2. `Pandas` [81, 82] [v. 1.1.3]
3. `Matplotlib` [83] [v. 3.3.2]
4. `SciPy` [84] [v. 1.5.2]
5. `mpmath` [85] [v. 1.1.0] (only necessary if both PSF schemes are required)

Additionally, Python 3 is required while 3.6 or higher is recommended. If you use Python 2 the code will give wrong results! Besides internal usage of pandas, νDoBe generally uses pandas' DataFrame class to output results in a table format. Pandas DataFrames provide a convenient framework for further analyses similar to numpy with additional features like, e.g., latex exports or plotting functions.

To ensure that there are no conflicts with other third-party python modules we recommend setting up a dedicated virtual environment when using νDoBe.

5.2.2 Setting up νDoBe

To use νDoBe simply download the code from GitHub and copy it into your project's main directory. Your projects directory structure should look similar to the example given in figure 8.

JHEP08(2023)043

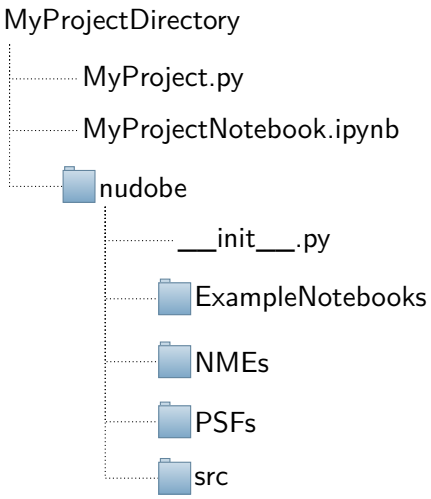


Figure 8. An example of how you should include νDoBe in your project. The `src` folder contains all the νDoBe sub-modules, the `NMEs` and `PSFs` folders contain the different sets of phase-space factors and nuclear matrix elements stored as `.csv` files and the `ExampleNotebooks` folder contains some Jupyter notebook examples of how to use νDoBe.

5.3 Parameters and constants

5.3.1 Physical constants

Physical constants such as particles masses and mixing angles are defined in the `constants.py` file and can be changed if that is required.

5.3.2 Units

Generally, all dimensionful parameters in νDoBe are defined in GeV, i.e., we set GeV = 1, numerically. The only exception to this is the minimal neutrino mass in plotting functions which is taken in eV. To make it easier to not get confused with units, the numerical values of different energy units can be accessed in the `nudobe.constants` module.

[1]:
```
from nudobe import constants
from constants import *
```

[2]:
```
#We can define a scale Lambda = 100TeV directly as
Lambda = 1e+5
print(Lambda)

#or by using the TeV parameter defined in constants
Lambda = 100*TeV
print(Lambda)
```

```
100000.0
100000.0
```

νDoBe includes parameters for the energy units of `TeV, GeV, MeV, keV, eV` and `meV`.

JHEP08(2023)043

5.3.3 Nuclear matrix elements

$0\nu\beta\beta$ decay rates depend on a set of nuclear matrix elements that involve complicated nuclear structure calculations. As of today, the NMEs are computed within different nuclear many-body methods and the results, unfortunately, tend to differ. νDoBe comes with three sets of NMEs that one can choose from; the *interacting boson model 2* ("IBM2") [59], the *quasi random-phase approximation* ("QRPA") [51] and the *shell model* ("SM") [64]. These particular sets of NMEs are chosen because within the corresponding method all NMEs required to analyze $0\nu\beta\beta$ from dimension-9 SMEFT operators have been computed. The corresponding files are stored in the `NMEs/` folder as `.csv` files. Other NME approximation methods can be studied by, simply, adding a new `.csv` file to the `NMEs/` folder. If you do so, please make sure to follow the definitions of the NMEs described in appendix D.

Within ν`DoBe` there are two possibilities to set NME methods:

1. You can set the method right at the start when initiating a model as will be shown in the following section. Every calculation you will do within this model will then use the corresponding NMEs.

2. Alternatively, most functions have a parameter called `method` that will reset the NMEs temporarily when calling the function (see the functions definitions).

Additionally, within the model classes the NMEs can be accessed and changed in the `model.NMEs` dictionary.

5.3.4 Phase-space factors

ν`DoBe` includes two approximation schemes for the calculation of PSFs and electron wave functions [86]. The PSF-scheme is defined when initiating a model class via the `PSF_scheme` parameter. Scheme "A" includes PSFs calculated from approximate wave functions in a uniform charge distribution while scheme "B" gives PSFs calculated exactly assuming a point-like nucleus. The Phase-Space factors for the approximation schemes A and B are stored in the `PSFs/` folder as `PSFs_A.csv` and `PSFs_B.csv`. Again, similar to the NMEs you can use custom PSFs by replacing entries in the `.csv` files accordingly. Note, however, that this will only change the overall magnitude of the PSFs while the spectra and angular correlations are still calculated from the electron wave functions defined in the `PSFs.py` file and by the choice of the PSF-scheme. Additional methods for calculating the electron wave functions will be added in a future update. Note, however, that while different approximation schemes do have a noticeable effect on the magnitude of the PSFs, the general shape of the electron spectra and angular correlation is not expected to be influenced largely by a different choice of wave functions [86]. Within a model class, you can also access and change the PSFs by replacing the corresponding entries in the `model.PSFpanda` DataFrame.

5.3.5 Low energy constants

The choice of low energy constants (LECs) strongly influences the resulting half-lives. The values of the known LECs as well as order of magnitude estimates have been summarized in [32] and are shown in table 4 for convenience.

JHEP08(2023)043

$n \to pe\nu$, $\pi \to e\nu$			$\pi\pi \to ee$		
g_A	1.271 ± 0.002	[87]	$g_1^{\pi\pi}$	0.36 ± 0.02	[88]
g_S	0.97 ± 0.13	[89]	$g_2^{\pi\pi}$	2.0 ± 0.2 GeV^2	[88]
g_M	4.7	[87]	$g_3^{\pi\pi}$	$-(0.62 \pm 0.06)$ GeV^2	[88]
g_T	0.99 ± 0.06	[89]	$g_4^{\pi\pi}$	$-(1.9 \pm 0.2)$ GeV^2	[88]
$\lvert g_T' \rvert$	$\mathcal{O}(1)$		$g_5^{\pi\pi}$	$-(8.0 \pm 0.6)$ GeV^2	[88]
B	2.7 GeV		$\lvert g_{\mathrm{T}}^{\pi\pi} \rvert$	$\mathcal{O}(1)$	
$n \to p\pi ee$			$nn \to pp\,ee$		
$\lvert g_1^{\pi N} \rvert$	$\mathcal{O}(1)$		$\lvert g_1^{NN} \rvert$	$\mathcal{O}(1)$	
$\lvert g_{6,7,8,9}^{\pi N} \rvert$	$\mathcal{O}(1)$		$\lvert g_{6,7}^{NN} \rvert$	$\mathcal{O}(1)$	
$\lvert g_{\mathrm{VL}}^{\pi N} \rvert$	$\mathcal{O}(1)$		$\lvert g_{\mathrm{VL}}^{NN} \rvert$	$\mathcal{O}(1)$	
$\lvert g_{\mathrm{T}}^{\pi N} \rvert$	$\mathcal{O}(1)$		$\lvert g_{\mathrm{T}}^{NN} \rvert$	$\mathcal{O}(1)$	
			$\lvert g_\nu^{NN} \rvert$	$-92.9\,\mathrm{GeV}^{-2} \pm 50\%$	[66, 67, 90]
			$\lvert g_{VL,VR}^{E,m_e} \rvert$	$\mathcal{O}(1)$	
			$\lvert g_{2,3,4,5}^{NN} \rvert$	$\mathcal{O}((4\pi)^2)$	

Table 4. The low-energy constants as used in νDoBe. The table is taken from [32] with an updated g_ν^{NN}. When varying LECs νDoBe uses a 50% uncertainty for g_ν^{NN}. Additionally, the unknown LECs are varied within their order of magnitude estimates (i.e. in the range $\pm[1/\sqrt{10}, \sqrt{10}] \times \mathcal{O}(|g|)$ while all other LECs are kept constant.

The preset values of the LECs are defined in the `constants.py` file. Global changes of the LECs are best taken care of in this file. Locally, within the model classes we have implemented two generic settings that can be applied when doing calculations.

1. In the first setting we use the known LECs and set the unknown LECs to null except keeping $g_{6,7}^{NN} = g_V^{\pi N} = \tilde{g}_V^{\pi N} = 1$ such that we still allow for short-range vector operators in the LEFT framework to contribute to the overall amplitude. This setting can be applied by calling `unknown_LECs = False` when initiating a model.

2. In the second setting we take the unknown LECs to be equal to their positive order of magnitude estimate. This option can be chosen by applying `unknown_LECs = True`.

You can also manually adjust each LEC. For each initiated model class, the LECs are stored within a dictionary. You can simply change the values for each LEC by replacing the corresponding values in the `model.LEC` dict.

We stress that the value of short-distance LECs, such as g_ν^{NN} are only meaningful within a given renormalization scheme that also affects the corresponding nuclear forces and thus the long- and short-distance NMEs. Refs. [66, 90] outlined a strategy how this can be achieved. The idea is that the amplitude of the process $nn \to pp + ee$, A_ν, is observable and should therefore be regulator independent. While the process itself cannot be measured in any practical way, the amplitude has been computed in [66, 90] by relating

JHEP08(2023)043

it to momentum integral of a known kernel (proportional to the neutrino propagator) multiplied by the generalized forward Compton scattering amplitude $nnW^+ \to ppW^-$. The low- and high-energy regime of this integral can be described model independently through, respectively, chiral EFT and the operator product expansion for perturbative QCD. The full amplitude is then obtained by interpolating between the two regimes using appropriate form factors for single-nucleon and nucleon-nucleon interactions. Once A_ν is obtained at some kinematic point, it becomes possible to fix g_ν^{NN} for an EFT of nucleon-nucleon interactions using any regularization scheme. Crucially, the value of g_ν^{NN} is not fixed but depends on the applied EFT and the associated regularization scheme, however, the observable amplitude A_ν is always correctly described. This procedure was followed in ref. [67] which applied chiral EFT nucleon-nucleon interactions to extract g_ν^{NN} and then performed ab initio computations of the NMEs related to $0\nu\beta\beta$ of ^{48}Ca. The value of g_ν^{NN} in the table is the value taken from ref. [67]. However, strictly speaking this value is not compatible with the NME sets of νDoBe for heavier isotopes which were obtained with different many-body methods. For that reason we have assigned a 50% uncertainty on the value of g_ν^{NN}.

5.4 Setting up a model

The goal of the tool is to directly connect BSM models that contain additional LNV sources to $0\nu\beta\beta$ decay rates and electron-kinematics measured in experiments. The main task of the user ("you", from now on) is to provide the matching relations between the specific BSM model to the LNV SMEFT or LEFT operators. Currently, the tool contains all $\Delta L = 2$ operators relevant for $0\nu\beta\beta$ involving first-generations quarks and leptons up-to-and-including dim-9 operators (all $\Delta L = 2$ SMEFT operators have odd dimension [34]). The basis of dim-5, dim-7, and dim-9 SMEFT operators has been derived in refs. [35–37, 45] and we follow the notation of these references. A full list of the relevant operators is given in appendix A.

5.4.1 EFT model classes

The code allows you to set up an EFT model consisting of LNV operators that trigger $0\nu\beta\beta$ decay. You can set up models either above the scale of electroweak symmetry breaking (EWSB) as a SMEFT model or below EWSB as a LEFT model. The most important classes are

JHEP08(2023)043

nudobe.EFT.SMEFT(WC, scale = m_W, name = **None**, unknown_LECs = **False**, method = "IBM2", PSF_scheme = "A")

nudobe.EFT.LEFT(WC, name = **None**, unknown_LECs = **False**, method = "IBM2", PSF_scheme = "A")

Parameter	**Type**	**Description**
WC	dictionay	Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. You only need to define the non-vanishing WCs here.
scale	float	Optional - Sets the scale of new physics Λ for the WCs. Needs to be set larger than or equal to 80 i.e. higher than the mass of the W boson. If `scale>m_W` ν`DoBe` will automatically run the provided WCs down to `m_W` when initiating a model.
name	string	Optional - Defines a name for the model. This will show up in plots.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". The preset value is "IBM2".
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge

Both are mostly identical in what they can do. In fact, the SMEFT class generally does all its calculations by matching the SMEFT Wilson coefficients onto LEFT to internally generate a LEFT class which then performs the desired calculations. See section 5.4.4 and appendix A for the different Wilson coefficients. The SMEFT→LEFT matching is provided in appendix B.

JHEP08(2023)043

5.4.2 Importing the model classes

To get started you first need to import the EFT classes which allow you to set up a model in either SMEFT or LEFT:

```
[ ]: from nudobe import EFT
```

5.4.3 Initiating a model

You can set up a model by defining the corresponding non-vanishing Wilson coefficients in a dictionary style.

```
[ ]: #define Wilson coefficients
Lambda = 1e+16
SMEFT_WCs = {"LH(5)" : 1/Lambda,
             "LH(7)" : 0.5/Lambda**3,
              .
              .
              .
            }
LEFT_WCs = {"m_bb"  : 1e-9, #Remember that 1eV = 1e-9GeV
            "SL(6)" : 1e-6,
             .
             .
             .
           }
```

Afterwards you can initiate a model with the specific choice of the Wilson coefficients. When initiating LEFT models you have to set the WCs at the matching scale m_W. The LEFT model class will then automatically run the Wilson coefficients to the scale relevant for neutrinoless double-β decay and save them internally to the parameter `model.WC` while the original input WCs are stored to the `model.WC_input` parameter. SMEFT models can be defined at an arbitrary scale $\Lambda \geq m_W$. See section 5.4.6 for more details on the relevant scales and RGE evolution.

All LEFT Wilson coefficients except for $m_{\beta\beta}$ are dimensionless while the SMEFT WCs are of mass dimension $4 - d$, where d is the operators dimension. Masses and scales are, generally, taken in GeV. That is, m_bb = 1e−9 corresponds to $m_{\beta\beta} = 1\,\text{eV}$. You can initiate the models by typing:

```
[ ]: #generate models
SMEFT_model = EFT.SMEFT(SMEFT_WCs, scale = Lambda)

LEFT_model = EFT.LEFT(LEFT_WCs)
```


JHEP08(2023)043

If you know you only want to work within SMEFT or LEFT or you do not want to put the "EFT." in front each time you initiate a new model class, you can also directly import the corresponding classes via

```
[ ]: from nudobe import EFT
     from EFT import LEFT, SMEFT
     .
     .
     .
     SMEFT_model = SMEFT(SMEFT_WCs, scale = 1e+16)
     LEFT_model  = LEFT(LEFT_WCs)
```

After having set up a model class you can call the Wilson coefficients at m_W (SMEFT) or Λ_χ (LEFT) via e.g.

```
[ ]: SMEFT_model.WC
```

5.4.4 List of Wilson coefficients

The relevant Wilson coefficients under consideration are given in tables 5 (LEFT) and 6 (SMEFT). Within `nudobe` they are defined as dictionaries. The dictionary key for each WC is given in the column "Code Label" in each table. After importing the EFT module you can access the WC dictionaries directly via

```
[ ]: from nudobe import EFT
```

```
[ ]: EFT.LEFT_WCs
```

```
[ ]: EFT.SMEFT_WCs
```

Alternatively, the list of WCs are also stored in the `nudobe.constants` module and can be accessed there.

5.4.5 Example: light neutrino exchange mechanism in LEFT

You can generate a model class that represents the $0\nu\beta\beta$-decay induced by the exchange of light Majorana neutrinos with $m_{\beta\beta} = 1\,\text{eV}$ using NMEs calculated in the IBM2 framework via

```
[1]: from nudobe import EFT
     from EFT import LEFT

     mass_mechanism = LEFT({"m_bb" : 1e-9}, method = "IBM2")
```


JHEP08(2023)043

5.4.6 RGE running

When initiating either a LEFT or a SMEFT model the different Wilson coefficients should generally be defined at the scale $\Lambda = m_W$. νDoBe will then take care of the running and matching procedure down to the chiral EFT scale $\Lambda_\chi = 2\,\text{GeV}$. The different RGEs are defined in the `RGE.py` file and the running of the LEFT WCs between m_W and Λ_χ is calculated and stored as a matrix when importing the `EFT` module. If you want to define WCs at a different scale than m_W you can run them down to m_W (SMEFT) or Λ_χ (LEFT) - or generally any other scale - by using the `RGE.run()` function that is also built into the model classes. Note that the matching scale SMEFT→LEFT is m_W.

---

nudobe.EFT.LEFT.run(WC = **None**, initial_scale = m_W, final_scale = lambda_chi, inplace = **False**)

nudobe.EFT.SMEFT.run(WC = **None**, initial_scale = **None**, final_scale = m_W, inplace = **False**)

Parameter	**Type**	**Description**
WC	dictionay	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. You only need to define the non-vanishing WCs here.
initial_scale	float	Optional - Scale the WCs are defined at initially.
final_scale	float	Optional - Scale the WCs should be evolved to.
inplace	bool	Optional - If **True** the resulting WCs after running will replace the models WCs.

The RGE running of the LEFT operators is given in [32] (eqs. 14-16) while the RGEs for the dimension 7 SMEFT operators can be found in [42] (eqs. 21-24). The RGEs of the SMEFT dimension 9 operators will be included in a future update. Currently, νDoBe does not evolve the SMEFT dimension 9 operators when using the `run()` function. Therefore, for reasons of consistency, we recommend defining SMEFT models directly at the matching scale m_W if both dimension 7 and dimension 9 operators are to be studied. Alternatively, you can use the SMEFT models `.run()` function to run the dimension 7 SMEFT operators from any arbitrary scale to the LEFT matching scale.

JHEP08(2023)043

Example: RGE evolution of a specific SMEFT model.

[1]:
```
from nudobe import EFT, constants
from EFT import SMEFT
from constants import *
```

After importing the SMEFT class we define a model with two non-vanishing dimension 7 operators. To show the impact of the RGE running we first define the model class without setting a scale explicitly such that the SMEFT class assumes the scale is m_W. This time we use the unit parameters defined in the `constants` module to define scales and masses.

[2]:
```
#Scale of new physics @ 50TeV
Lambda = 50*TeV

#Generate SMEFT model
model = SMEFT({"LeudH(7)" : 0.05/Lambda**3, "LHD1(7)" : 5/Lambda**3})
```

[3]:
```
#Non-vanishing SMEFT WCs before RGE running
for WC in model.WC:
    if model.WC[WC] != 0:
        print(WC, model.WC[WC])
```

[3]:
```
LHD1(7) 4e-14
LeudH(7) 4.0000000000000004e-16
```

We now calculate the half-life in ^{136}Xe before running the SMEFT WCs from 50 TeV down to m_W

[4]:
```
#Half-Life in 136Xe
model.t_half("136Xe")
```

[4]:
```
1.7039062688609342e+29
```

Now we use the `run()` function of the SMEFT class to run the WCs down to m_W and recalculate the half-life

[5]:
```
#Run the WCs down to the matching scale m_W
model.run(initial_scale = Lambda, inplace = True)

#Recalculate Half-Life
model.t_half("136Xe")
```

[5]:
```
1.7973515786734228e+29
```

To see how the WCs change when running them from 50 TeV down to m_W we can print them again

JHEP08(2023)043

[6]:
```
#Non-vanishing SMEFT WCs after RGE running
for WC in model.WC:
    if model.WC[WC] != 0:
        print(WC, model.WC[WC])
```

```
LHD1(7)  3.105078059915025e-14
LHD2(7)  4.0238127608753095e-15
LHW(7)  -3.3074147823810323e-16
LeudH(7) 3.915696874279481e-16
```

5.5 Getting the half-life

There are two ways you can get the half-lives from a specific model. To get the half-lives in all isotopes that are included in the chosen NME method you can simply run

---

model.half_lives(WC = **None**, method = **None**, vary_LECs = **False**, n_points = 1000)

Parameter	**Type**	**Description**
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.
vary_LECs	bool	Optional - If set to **True** the unknown LECs will be varied within their NDA estimates (see table 4). If set to **False** the unknown LECs will stay fixed.
n_points	integer	Optional - Number of variations.

---

with `model` initiated via either `nudobe.EFT.SMEFT` or `nudobe.EFT.SMEFT`. This will output a pandas DataFrame containing the half-lives of all isotopes included. It works for both SMEFT and LEFT models. If you only want to get the half-life for a single isotope you can run the function

JHEP08(2023)043

model.t_half(isotope, WC = **None**, method = **None**)

Parameter	Type	Description
isotope	string	Optional - Defines the isotope to be studied.
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.

Example: Light neutrino exchange mechanism in LEFT.
As an example we study once more the standard mass mechanism induced by the exchange of light Majorana neutrinos. We want to find the expected half-lives in the case of an effective Majorana mass $m_{\beta\beta} = 100\,\text{meV}$:

[1]:
```
from nudobe import EFT, constants
from EFT import LEFT
from constants import *
```

The resulting range of values can be used as an estimate for the theoretical uncertainty of the predictions arising from QCD matrix elements (that is, the LECs).

[2]:
```
#Define Wilson coefficients for the standard mass mechanism
#with m_bb = 100meV
WC = {"m_bb" : 100*meV}

#NME method
method = "IBM2"

#initiate model
mass_mechanism = LEFT(WC, method = method)
```

[3]:
```
#half-life for 76Ge
mass_mechanism.t_half("76Ge")
```


JHEP08(2023)043

[3]:
```
4.754252877417845e+25
```

[4]:
```
#half-lives in all isotopes
mass_mechanism.half_lives()
```

[4]:

	76Ge	82Se	...	232Th	238U
0	4.754253e+25	1.578798e+25	...	9.476641e+24	2.565784e+24

[5]:
```
#half-lives in all isotopes with varied LECs
mass_mechanism.half_lives(vary_LECs = True, n_points = 100)
```

[5]:

	76Ge	82Se	...	232Th	238U
0	4.108211e+25	1.360687e+25	...	7.861367e+24	2.130089e+24
1	3.791413e+25	1.254053e+25	...	7.105329e+24	1.925981e+24
2	5.139999e+25	1.709430e+25	...	1.048842e+25	2.838449e+24
...	...	...	...	...	...
97	5.193175e+25	1.727460e+25	...	1.063069e+25	2.876775e+24
98	3.979882e+25	1.317465e+25	...	7.552259e+24	2.046653e+24
99	5.260296e+25	1.750227e+25	...	1.081124e+25	2.925407e+24

[100 rows x 18 columns]

5.6 Decay-rate formula

In case you want to get an analytical expression of the decay rate in terms of the different Wilson coefficients in your model you can use the function `generate_formula()`

---

nudobe.functions.generate_formula(WC, isotope = "136Xe", method = "IBM2", decimal = 2, output = "latex", unknown_LECs = **False**, PSF_scheme ="A")

model.generate_formula(isotope, WC = **None**, method = **None**, decimal = 2, output = "latex", unknown_LECs = **None**, PSF_scheme = **None**)

JHEP08(2023)043

Parameter	Type	Description
isotope	string	Defines the isotope to be studied.
WC	list	Optional - List of non-zero Wilson coefficients that should contribute to the half-life ["WCname1" ..., "WCnameN"]. If **None** the models non-zero WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If called via the `functions` module the preset value is "IBM2". If called via a `model` class he models method will be used if **None**.
decimal	integer	Optional - Sets the numbers of decimals used in rounding.
output	string	Optional - Get output in "latex" or "html" format.
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge

to generate an expression for the decay rate

$$T_{1/2}^{-1} = \sum_{ij} M_{ij} C_i^\dagger C_j \tag{5.1}$$

either in latex or html format. This equation represents a matrix equation of the form

$$T_{1/2}^{-1} = C^\dagger M C \tag{5.2}$$

and you can obtain the matrix coefficients from

nudobe.functions.generate_matrix(WC, isotope = "136Xe", method = "IBM2",
unknown_LECs = **False**, PSF_scheme ="A")

model.generate_matrix(isotope, WC = **None**, method = **None**,
unknown_LECs = **False**, PSF_scheme = **None**)

JHEP08(2023)043

Parameter	Type	Description
isotope	string	Defines the isotope to be studied.
WC	list	Optional - List of non-zero Wilson coefficients that should contribute to the half-life ["WCname1" ..., "WCnameN"]. If **None** the models non-zero WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If called via the `functions` module the preset value is "IBM2". If called via a `model` class he models method will be used if **None**.
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge

Both functions will only consider those Wilson coefficients which are non-zero in your model.

Example: the light-neutrino-exchange mechanism with an additional LNV right-handed current

Let's assume a LEFT model with two lepton number violating terms in the Lagrangian

$$\mathcal{L}_{\Delta L=2} = -\frac{1}{2} m_{\beta\beta} \left[\overline{\nu^C_{L,e}} \nu_{L,e}\right] + \frac{1}{v^2} C^{(6)}_{\mathrm{VR}} \left[\overline{u_R} \gamma^\mu d_R\right] \left[\overline{e_R} \gamma_\mu \nu^C_{L,e}\right] \tag{5.3}$$

[1]:
```
import numpy as np
from nudobe import EFT, constants
from EFT import LEFT
from constants import *
```

[2]:
```
#define Wilson coefficients and model
WC = {"m_bb" : 100*meV,
      "VR(6)" : 1e-7
     }

model = LEFT(WC, method = "IBM2")
```

JHEP08(2023)043

[3]:
```
#get the decay rate R in 76Ge
R = 1/model.t_half("76Ge")
R
```

[3]:
```
2.418831433432257e-26
```

[4]:
```
#get decay rate formula for 76Ge in latex form
print(model.generate_formula("76Ge"))
```

```
$T_{1/2}^{-1} = +2.1\times
10^{-6}\left|\frac{m_{\beta\beta}}{1\mathrm{GeV}}\right|^2+1.07\times
10^{-13}\left|C_{VR}^{(6)}\right|^2+2.09\times 10^{-10}
\mathrm{Re}\left[\frac{m_{\beta\beta}}{1\mathrm{GeV}}
({C_{VR}^{(6)}})^*\right]$
```

which if we put it into latex reads

$$T_{1/2}^{-1} = +2.1 \times 10^{-6} \left|\frac{m_{\beta\beta}}{1\text{GeV}}\right|^2 + 1.07 \times 10^{-13} \left|C_{VR}^{(6)}\right|^2 + 2.09 \times 10^{-10} \text{Re}\left[\frac{m_{\beta\beta}}{1\text{GeV}}(C_{VR}^{(6)})^*\right] \tag{5.4}$$

[5]:
```
#get the decay rate matrix M
M = model.generate_matrix("76Ge")
M
```

[5]:
```
array([[2.10337991e-06, 1.04362734e-10],
       [1.04362734e-10, 1.06726053e-13]])
```

[6]:
```
#get the decay rate from the matrix M
C  = np.array(list(WC.values()))
R2 = C.T.conj() @ M @ C
R2
```

[6]:
```
2.418831433432201e-26
```

Comparing the results on the decay rate $T_{1/2}^{-1}$ R and R2 from the functions model.half_lives() and model.generate_matrix(), respectively, we see that they are equal up to some small accumulated rounding errors.

[7]:
```
(R-R2)/R
```

[7]:
```
2.313607087198588e-14
```


JHEP08(2023)043

5.7 Plot half-lives — varying a single Wilson coefficient

5.7.1 Line-plots

You can plot the half-lives $T_{1/2}$ (`model.plot_t_half(...)`), the decay-rate $T_{1/2}^{-1}$ (`model.plot_t_half_inv(...)`) or the effective Majorana mass m_{eff} (`model.plot_m_eff(...)`) as a line-plot while varying a single Wilson coefficient via the functions:

---

model.plot_WC_variation(xaxis = "m_min", yaxis = "t", isotope = "76Ge",
x_min = 1e-4, x_max = 1e+0, y_min = **None**, y_max = **None**, xscale = "log",
yscale = "log", n_points = 100, cosmo = **False**, m_cosmo = 0.15, limits = **None**,
experiments = **None**, ordering = "both", dcp = 1.36, numerical_method = "Powell",
show_mbb = **False**, normalize = **False**, colorNO = "b", colorIO = "r",
legend = **True**, labelNO = **None**, labelIO = **None**, autolabel = **True**,
alpha_plot = 0.5, alpha_mass = 0.05, alpha_cosmo = 0.1, vary_phases = **True**,
alpha = [0,0], savefig = **False**, file = "variation.png", dpi = 300)

model.plot_t_half() = model.plot_WC_variation(yaxis ="t", ...)

model.plot_t_half_inv() = model.plot_WC_variation(yaxis ="1/t", ...)

model.plot_m_eff() = model.plot_WC_variation(yaxis ="m_eff", ...)

Parameter	Type	Description
xaxis	string	Optional - Wilson coefficient varied on the x-axis.
yaxis	string	Optional - Choose from "t", "1/t", "m_eff" to get the half-life, inverse half-life or effective neutrino mass.
isotope	string	Optional - Defines the isotope to be studied.
x_min	float	Optional - Minimal x value.
x_max	float	Optional - Maximal x value.
y_min	float	Optional - Minimal y value.
y_max	float	Optional - Maximal y value.

JHEP08(2023)043

xscale	string	Optional - Sets scaling of x-axis e.g. "log" or "lin".
yscale	string	Optional - Sets scaling of y-axis e.g. "log" or "lin".
n_points	integer	Optional - Number of datapoints on the x-axis.
cosmo	bool	Optional - If **True** show cosmology limit.
m_cosmo	integer	Optional - Limit from cosmology.
limits	dictionary	Optional - Plots limits from experiments {Name : {limit, color, linestyle, linewidth, alpha, fill, label}}. The Isotope is assumed to be the same as the isotope parameter for the plot.
experiments	dictionary	Optional - The same as limits but instead of giving the y-axis limit you set the half-life limit.
ordering	string	Optional - "both", "NO" or "IO" sets neutrino mass ordering.
dcp	float	Optional - Dirac CP phase.
numerical_method	string	Optional - Optimization method. See scipy.optimize.minimize
show_mbb	bool	Optional - If **True** plot mass mechanism for comparison.
normalize	float	Optional - If **True** normalize to mass mechanism.
colorNO	string	Optional - Set color for NO plot.
colorIO	string	Optional - Set color for IO plot.
legend	bool	Optional - If **True** a legend is shown.
labelNO	string	Optional - Legend label of NO plot.
labelIO	string	Optional - Legend label of IO plot.
autolabel	bool	Optional - If **True**, legend labels are generated automatically.

JHEP08(2023)043

alpha_plot	string	Optional - Set alpha for filled areas.
alpha_mass	string	Optional - Set alpha for mass mechanism if `show_mbb=`**True**.
alpha_cosmo	string	Optional - Set alpha for mass mechanism if cosmo=**True**.
vary_phases	string	Optional - If **True**, vary the unknown phases of the x-axis WC.
alpha	float	Optional - 2-entries float array that defines the unknown Majorana phases or float that defines the complex WC phase. Only necessary if phases are not varied on the x-axis WC.
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

These functions will plot the allowed y-axis range of values by minimizing and maximizing the half-life (effective mass) with respect the phase of the Wilson coefficient given in the `xaxis` parameter while varying the absolute value in the range given by the parameters `x_min` and `x_max`. You can turn off the variation of the unknown phases by setting `vary_phases=`**False**. All other Wilson coefficients will be kept constant as well as the unknown LECs. Note that this assumes that all Wilson coefficients are independent from each other. See the provided Jupyter notebook on the mLRSM for an example of how to approach studying a model with interdependent Wilson coefficients.

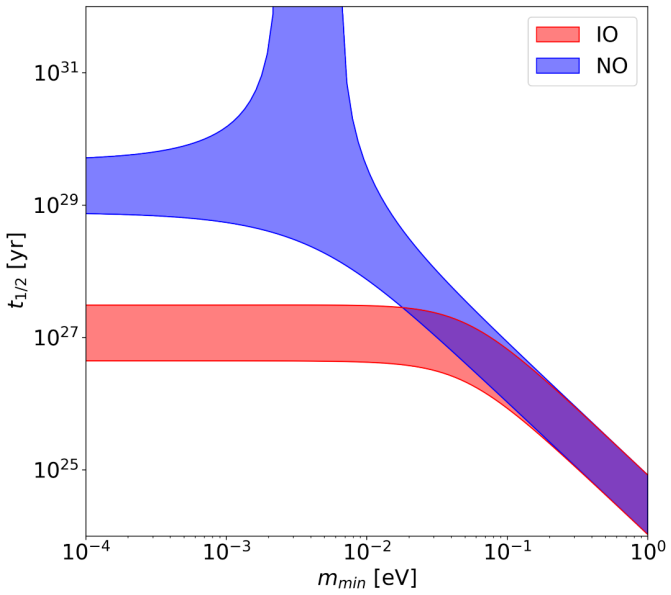
Example: the light-neutrino-exchange mechanism with an additional LNV right-handed current once more

Let us again consider the previous example of an LNV right-handed current appearing in addition to the effective neutrino Majorana mass. First we plot the half-life for the LνEM only and save it to `t_half_mass.png` by typing

[1]:
```
from nudobe import EFT
from EFT import LEFT
```

[2]:
```
fig = LEFT({}, method = "SM").plot_t_half(savefig = True,
                                          file    = "t_half_mass.png")
```


JHEP08(2023)043



This time we chose NMEs generated in the shell model ("SM"), including the g_ν^{NN} contribution. Then we want to plot the half-lives for our model of interest which we first have to initialize via

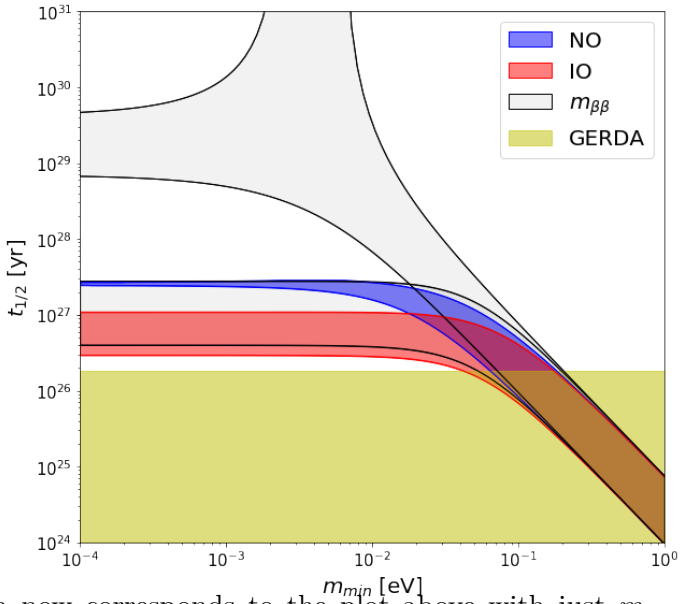
[3]:
```
WC = {"m_bb" : 0.1e-9,
      "VR(6)" : 1e-7
     }
model = LEFT(WC, method = "SM")
```

In our half-life figure we want to show the current experimental limit from GERDA [2] in yellow. Hence, we choose ^{76}Ge as the isotope and define GERDA in the parameter `experiments`. We plot the standard mass mechanism alongside by putting `show_mbb=True`. Finally, we save the generated figure as a `.png` file to `t_half_model.png`

[4]:
```
isotope = "76Ge"
experiments = {"GERDA" : {"limit"     : 1.8e+26,
                          "color"     : "y",
                          "linestyle" : "-",
                          "linewidth" : 1,
                          "alpha"     : 0.5,
                          "fill"      : True,
                          "label"     : "GERDA"}}

fig = model.plot_t_half(isotope = isotope, experiments = experiments,
                        show_mbb = True, y_min = 1e+24, y_max = 1e+31,
                        savefig = True, file = "t_half_model.png")
```

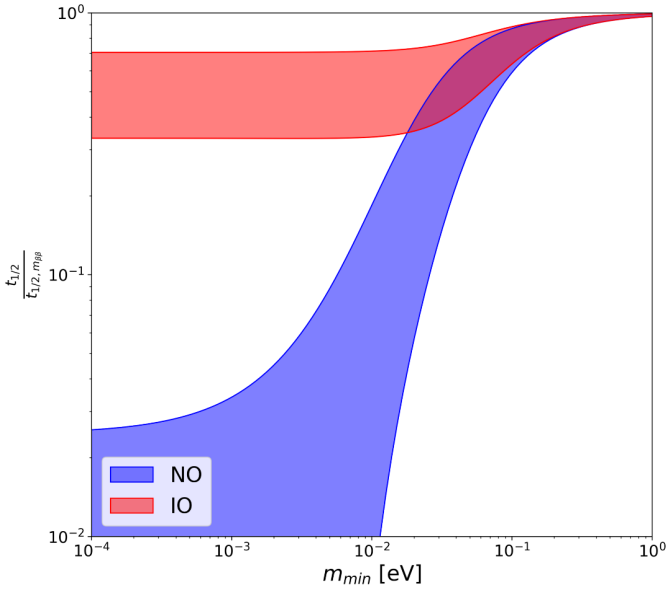
JHEP08(2023)043



The gray area now corresponds to the plot above with just $m_{\beta\beta}$, i.e., it shows the expected half-life from the LνEM. We see that due to the new LNV operator, the $0\nu\beta\beta$ rates are significantly enhanced for the normal ordering for small m_{min} but this effect is less pronounced in the inverted ordering. The yellow area shows the regions of the parameter space excluded by the GERDA experiment [2].

Additionally, instead of plotting the mass mechanism alongside our BSM model we can normalize the half-life generated by our model to the LνEM and plot it via

[5]:
```
model.plot_t_half(isotope   = isotope,
                  normalize = True,
                  savefig   = True,
                  file      = "t_half_normalized.png")
```



JHEP08(2023)043

From these plots we can easily see that while the case of inverse mass hierarchy is almost unaffected by the additional higher dimensional contribution, the normal mass hierarchy case results in significantly higher decay rates for minimal neutrino masses below 0.1 eV.

5.7.2 Scatter plots

The above plots show all allowed values when varying the unknown phase(s) of the Wilson coefficient on the x-axis. If, instead, you want to additionally vary the unknown LECs, get a graphical representation of the probability distribution when doing a variation, or simply want a different design you can generate scatter plots through

---

plot_WC_variation_scatter(xaxis = "m_min", yaxis = "t", isotope = "76Ge",
vary_phases = **True**, vary_LECs = **False**, alpha = [0,0], n_points = 10000,
markersize = 0.5, x_min = 1e-4, x_max = 1, y_min = **None**, y_max = **None**,
xscale = "log", yscale = "log", limits = **None**, experiments = **None**,
ordering = "both", dcp = 1.36, show_mbb = **False**, normalize = **False**,
cosmo = **False**, m_cosmo = 0.15, colorNO = "b", colorIO = "r", legend = **True**,
labelNO = **None**, labelIO = **None**, autolabel = **True**, alpha_plot = 1,
alpha_mass = 0.05, alpha_cosmo = 0.1, savefig = **False**, file = "var_scat.png",
dpi = 300)

model.plot_t_half_scatter(...) = model.plot_WC_variation(yaxis ="m_eff", ...)

model.plot_t_half_inv_scatter(...) = model.plot_WC_variation(yaxis ="m_eff", ...)

model.plot_m_eff_scatter(...) = model.plot_WC_variation(yaxis ="m_eff", ...)

Parameter	Type	Description
xaxis	string	Optional - Wilson coefficient varied on the x-axis.
yaxis	string	Optional - Choose from "t", "1/t", "m_eff" to get the half-life, inverse half-life or effective neutrino mass.
isotope	string	Optional - Defines the isotope to be studied.
vary_phases	string	Optional - If **True**, vary the unknown phases of the x-axis WC.
vary_LECs	string	Optional - If **True**, vary the unknown LECs within their NDA estimate (see table 4).

JHEP08(2023)043

alpha	float	Optional - 2-entries float array that defines the unknown Majorana phases or float that defines the complex WC phase. Only necessary if phases are not varied on the x-axis WC.
n_points	integer	Optional - Number of scattered datapoints.
markersize	float	Optional - Markersize of scattered points.
x_min	float	Optional - Minimal x value.
x_max	float	Optional - Maximal x value.
y_min	float	Optional - Minimal y value.
y_max	float	Optional - Maximal y value.
xscale	string	Optional - Sets scaling of x-axis e.g. "log" or "lin".
yscale	string	Optional - Sets scaling of y-axis e.g. "log" or "lin".
limits	dictionary	Optional - Plots limits from experiments {Name : {limit, color, linestyle, linewidth, alpha, fill, label}}. The Isotope is assumed to be the same as the isotope parameter for the plot.
experiments	dictionary	Optional - The same as limits but instead of giving the y-axis limit you set the half-life limit.
ordering	string	Optional - “both”, “NO” or “IO” sets neutrino mass ordering.
dcp	float	Optional - Dirac CP phase.
numerical_method	string	Optional - Optimization method. See scipy.optimize.minimize
show_mbb	bool	Optional - If **True** plot mass mechanism for comparison.
normalize	float	Optional - If **True** normalize to mass mechanism.

JHEP08(2023)043

cosmo	bool	Optional - If **True** show cosmology limit.
m_cosmo	integer	Optional - Limit from cosmology.
colorNO	string	Optional - Set color for NO plot.
colorIO	string	Optional - Set color for IO plot.
legend	bool	Optional - If **True** a legend is shown.
labelNO	string	Optional - Legend label of NO plot.
labelIO	string	Optional - Legend label of IO plot.
autolabel	bool	Optional - If **True**, legend labels are generated automatically.
alpha_plot	string	Optional - Set alpha for filled areas.
alpha_mass	string	Optional - Set alpha for mass mechanism if `show_mbb`=**True**.
alpha_cosmo	string	Optional - Set alpha for mass mechanism if cosmo=**True**.
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

Example: light-neutrino-exchange mechanism with variation of g_ν^{NN}
Again, we import the module and define the model by setting the NME method and WCs

[1]:
```
from nudobe import EFT
from EFT import LEFT
```

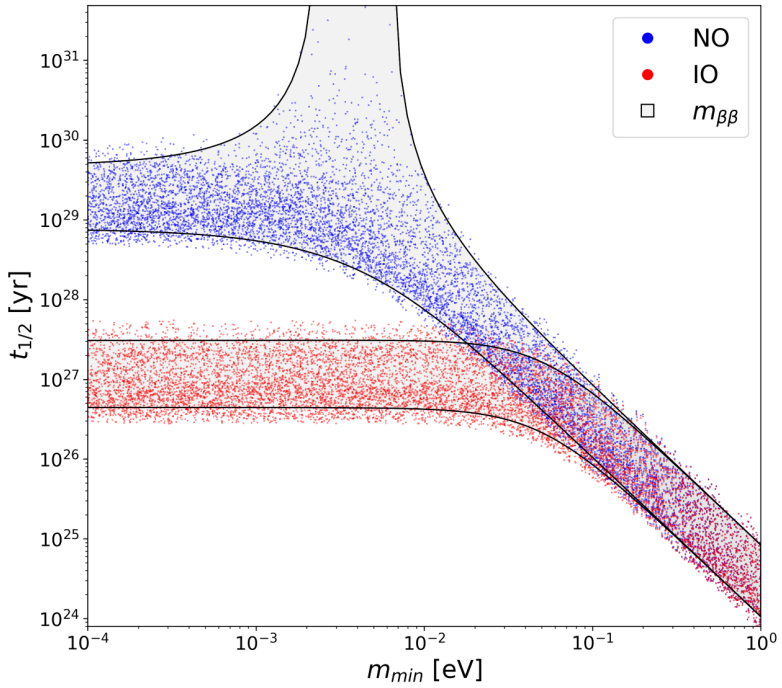
[2]:
```
WC = {"m_bb" : 0.1e-9}
method = "SM"

model = LEFT(WC, method = method)
```

We can generate a scatter-plot comparing the standard mass mechanism with and without variation of the unknown LEC g_ν^{NN} in a range of $\pm 50\%$ around the central value

JHEP08(2023)043

```
[3]: model.plot_t_half_scatter(vary_LECs = True, show_mbb = True,
                               savefig = True, file = "mbb_scatter.png",
                               dpi = 300)
```



In the above plot, the scattered points show the resulting half-lives when varying both the Majorana phases as well as g_ν^{NN} while the shaded regions show the allowed half-life region when only the Majorana phases are varied.

5.8 Half-life ratios

To distinguish different models among each other it can be interesting to look at the ratios of half-lives compared to a reference isotope [25], e.g. ^{76}Ge,

$$R = \frac{T_{1/2}(^{A}Z)}{T_{1/2}(^{76}\mathrm{Ge})}. \tag{5.5}$$

You can return the ratios in all isotopes of interest in a tabular form by using

JHEP08(2023)043

model.ratios(reference_isotope = "76Ge", normalized = **True**, WC = **None**,
method=**None**, vary_LECs = **False**, n_points = 100)

Parameter	Type	Description
reference_isotope	string	Optional - Sets the reference isotope for the half-life ratios.
normalized	bool	Optional - If **True** the ratios will be normalized to the standard mass mechanism $R/R_{m_{\beta\beta}}$.
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.
vary_LECs	bool	Optional - If set to **True** the unknown LECs will be varied within their NDA estimates (see table 4). If set to **False** the unknown LECs will stay fixed
n_points	integer	Optional - Number of variations.

which will output a pandas DataFrame containing the half-life ratios. You can depict the ratio R or, if you set `normalized=`**True**, $R/R_{m_{\beta\beta}}$. Additionally, you can generate a scatter plot of these ratios through

JHEP08(2023)043

model.plot_ratios(reference_isotope = "76Ge", normalized = **True**, WC = **None**, method=**None**, vary_LECs = **False**, n_points = 100, color = "b", alpha = 0.25, addgrid = **True**, savefig = **False**, file = "ratios.png", n_dpi = 300)

Parameter	Type	Description
reference_isotope	string	Optional - Sets the reference isotope for the half-life ratios.
normalized	bool	Optional - If **True** the ratios will be normalized to the standard mass mechanism $R/R_{m_{\beta\beta}}$.
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.
vary_LECs	bool	Optional - If set to **True** the unknown LECs will be varied within their NDA estimates. If set to **False** the unknown LECs will stay fixed
n_points	integer	Optional - Number of variations.
color	string or list	Optional - Sets the color of the plot either in RGB or via string. See Matplotlib for details.
alpha	float	Optional - Sets the alpha value for the variational points.
addgrid	bool	Optional - If **True** a grid is plotted.
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

JHEP08(2023)043

Example: a LNV right-handed current (aka the λ-mechanism)

[1]:
```
from nudobe import EFT
from EFT import LEFT
```

[2]:
```
#Set WCs
WC = {"VR(6)": 1e-7}

#Initiate model
model = LEFT(WC, method = "SM")
```

[3]:
```
#Define reference isotope
reference_isotope = "76Ge"
```

[4]:
```
#Calculate half-life ratios R/R_mbb
ratios = model.ratios(reference_isotope)
ratios
```

[4]:

	48Ca	76Ge	82Se	124Sn	130Te	136Xe
0	0.313404	1.0	0.4864	0.842622	0.721081	0.73654

[5]:
```
#Calculate ratios R/R_mbb with varied LECs
ratios_vary = model.ratios(reference_isotope, vary_LECs = True)
ratios_vary
```

[5]:

	48Ca	76Ge	82Se	124Sn	130Te	136Xe
0	0.188201	1.0	0.466779	0.704380	0.562058	0.607119
1	0.184358	1.0	0.490324	0.691977	0.551269	0.600332
2	0.293439	1.0	0.584830	0.768171	0.655088	0.691217
3	0.346657	1.0	0.791642	0.748110	0.655176	0.705475
4	0.209962	1.0	0.495675	0.707698	0.573777	0.616179
..	...	...	...	...	...	...
95	0.208918	1.0	0.462122	0.721153	0.581308	0.620530
96	0.268060	1.0	0.587650	0.739712	0.622569	0.664289
97	0.235483	1.0	0.510691	0.762411	0.630302	0.671020
98	0.153871	1.0	0.513988	0.642975	0.502340	0.559727
99	0.343899	1.0	0.771410	0.817641	0.727843	0.772934

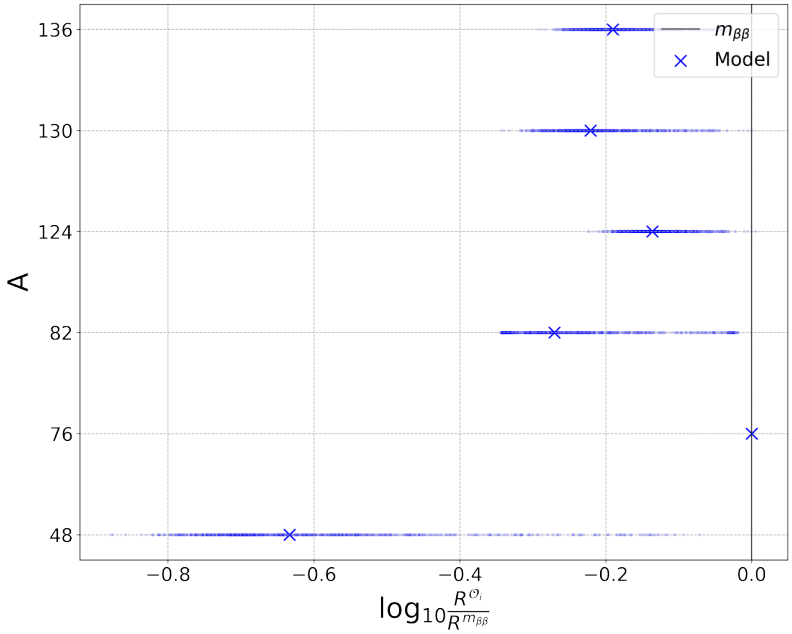
[100 rows x 6 columns]

[6]:
```
#Get central ratio values of the LEC variation
ratios_vary.median()
```

JHEP08(2023)043

[6]:
```
48Ca     0.229494
76Ge     1.000000
82Se     0.526431
124Sn    0.725552
130Te    0.594985
136Xe    0.637997
dtype: float64
```

[7]:
```
#Plot Ratios
model.plot_ratios("76Ge", vary_LECs = True, n_points = 1000, alpha = 0.1,
                  show_central = True, savefig = True,
                  file = "CVR6_ratios.png", dpi = 300)
```



Here, the x-axis shows the range of ratio values while the y-axis gives the isotope mass number A.

5.9 Phase-space observables

The code can generate both the single electron spectra $\mathrm{d}\Gamma/\mathrm{d}\epsilon$ defined in (C.3) as well as the energy dependant angular correlation coefficients a_1/a_0 defined in (C.5) for any given model. To calculate the numerical values for a given normalized electron energy

$$\bar{\epsilon} = \frac{\epsilon - m_e}{Q} \tag{5.6}$$

you can use the functions

JHEP08(2023)043

model.spectrum(Ebar, isotope = "76Ge", WC = **None**, method = **None**)

model.angular_corr(Ebar, isotope = "76Ge", WC = **None**, method = **None**)

Parameter	Type	Description
Ebar	float	Normalized electron energy $0 <$ Ebar < 1. Either float or array of floats.
isotope	string	Optional - Defines the isotope to be studied.
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.

Plots are then generated through

model.plot_spec(isotope="76Ge", WC = **None**, method=**None**, print_title = **False**, addgrid = **True**, show_mbb = **True**, normalize_x = **True**, savefig=**False**, file = "spec.png", dpi = 300)

model.plot_corr(isotope="76Ge", WC = **None**, method=**None**, print_title = **False**, addgrid = **True**, show_mbb = **True**, normalize_x = **True**, savefig=**False**, file = "angular_corr.png", dpi = 300)

Parameter	Type	Description
isotope	string	Optional - Defines the isotope to be studied.
WC	dictionary	Optional - Defines the Wilson coefficients as {"WCname1" : WCvalue1 ..., "WCnameN" : WCvalueN}. If **None** the models WCs will be used.

JHEP08(2023)043

method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If **None** the models method will be used.
print_title	string	Optional - Put a title into the plot.
addgrid	bool	Optional - If **True** a grid is plotted.
show_mbb	bool	Optional - If **True** plot mass mechanism for comparison.
normalize_x	bool	Optional - If **True** normalize x-axis to $\bar{\epsilon} \in [0, 1]$. If **False** show $T = \epsilon - m_e$ on the x-axis.x
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

Example: the light-neutrino-exchange mechanism with an additional LNV right-handed current

Let's look at our previous example again which combines a lepton number violating right-handed vector current with the standard mass mechanism. This time we choose $m_{\beta\beta} = 0.1\,\text{eV}$ and $C_{\text{VR}}^{(6)} = 3 \times 10^{-7}$. We initiate the model through

[1]:
```
import numpy as np
from nudobe import EFT
from EFT import LEFT
```

[2]:
```
WC = {"m_bb" : 0.1e-9,
      "VR(6)" : 3e-7
     }

model = LEFT(WC, method = "SM",
             name = r"$m_{\beta\beta} + C_\mathrm{VR}^{(6)}$")
```

Then we define an array `Ebar` that represents the normalized kinetic energy of the outgoing electrons via $\overline{E} = (E_e - m_e)/Q$. Additionally, we need to define a parameter `e` which helps us to avoid poles in the calculation of phase-space quantities g_{0k} and h_{0k} (see appendix C)

JHEP08(2023)043

[4]:
```
e = 1e-5 #avoid poles

Ebar = np.linspace(0+e, 1-e, 10)
```

We can then calculate the single electron spectrum

[5]:
```
model.spectrum(Ebar)
```

[5]:
```
array([4.35812489e-27, 5.46776633e-27, 5.60235186e-27, 5.22051437e-27,
       4.87200447e-27, 4.87200447e-27, 5.22051437e-27, 5.60235186e-27,
       5.46776633e-27, 4.35812489e-27])
```

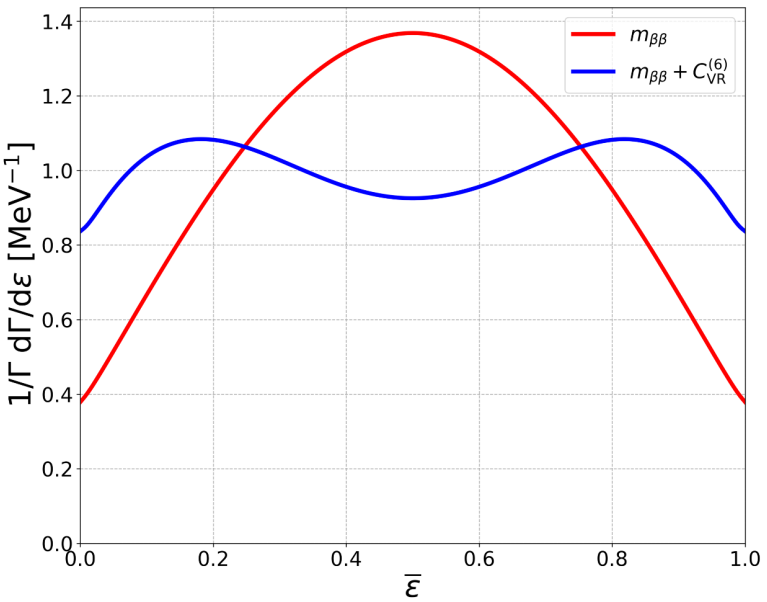
and the angular correlation coefficient

[6]:
```
model.angular_corr(Ebar)
```

[6]:
```
array([ 5.24575925e-04, -8.91742707e-02, -3.31779066e-01, -6.
↪33496492e-01,
       -8.64038720e-01, -8.64038720e-01, -6.33496492e-01, -3.
↪31779066e-01,
       -8.91742707e-02,  5.24575925e-04])
```

Plots can be generated via

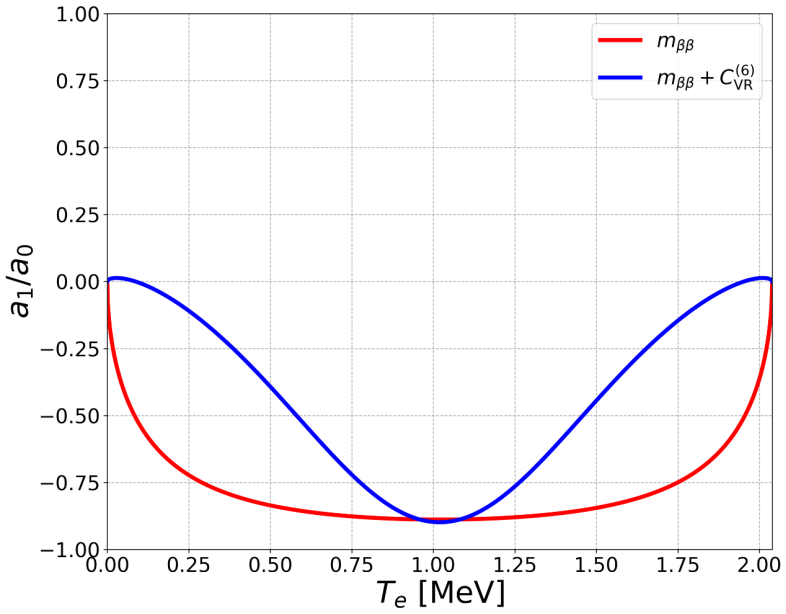
[7]:
```
model.plot_spec(linewidth = 4, normalize_x = True, savefig = True,
                file = "spec.png")
```



JHEP08(2023)043

for the single electron energy spectrum and via

```
[8]: model.plot_corr(linewidth = 4, normalize_x = False, savefig = True,
                    file = "corr.png")
```



for the angular correlation. In the latter case we refrained from normalizing the x-axis to present both options.

While in case of the LνEM, the energy distribution for a single electron peaks, as expected, at $Q/2$, the exotic contribution proportional to $C_{\mathrm{VR}}^{(6)}$ features a dip around that value (see e.g. ref. [25]), and therefore modifies the combined spectrum substantially, resulting in two equally high peaks on the sides of the distribution, i.e. for small and large energies of the electron, while there is a local minimum at $Q/2$. The negative angular correlation of the electrons obtained for the mass mechanism corresponds to the fact that they are preferably emitted back-to-back. The additional contribution from $C_{\mathrm{VR}}^{(6)}$ distorts the shape, such that there are small ranges of energies, for which the correlation becomes positive and the dip around the middle of the distribution is more pronounced. This is caused, again, by a fairly distinctive shape of the correlation for this higher dimensional $0\nu\beta\beta$ mechanism, see e.g. ref. [25].

Measuring the spectra of the individual electrons as well as the angular correlation between the emitted electrons can help to identify BSM scenarios with non-standard vector interactions [25]. Experiments like NEMO-3 [4, 11] and its next-generation upgrade SuperNEMO [91] are equipped to resolve the individual electron kinematics such that both the spectra and the angular correlation can be studied. In fact, these observables are avail-

JHEP08(2023)043

able for the ordinary double beta decay observed by NEMO-3 and also in this case they can be used to probe new physics [92].

5.10 Limits on Wilson coefficients

5.10.1 Single operator dominance

Given experimental limits on the half-life of the $0\nu\beta\beta$-decay one can translate these into limits on the different Wilson coefficients assuming the contributions from a single operator dominates the $0\nu\beta\beta$ amplitude. Under this assumption and using the LEFT operators, the decay rates scale as

$$t_{1/2}^{-1}(C_i) \propto |C_i|^2 \ . \tag{5.7}$$

We can calculate limits on the different Wilson coefficients from a given half-life limit, $t_{1/2}^{\lim}({}^A Z)$, through

$$|C_i| = \sqrt{\frac{t_{1/2}(C_i = 1, {}^A Z)}{t_{1/2}^{\lim}({}^A Z)}} \ . \tag{5.8}$$

Additionally, one can translate limits on the different Wilson coefficients into limits on the expected new-physics scale Λ. Assuming that LEFT operators of dimension d scale as

$$C_i \simeq \left(\frac{v}{\Lambda}\right)^{(d-4)} \tag{5.9}$$

we can estimate the scale of new physics as

$$\Lambda \simeq v \, (C_i)^{1/(4-d)} \tag{5.10}$$

Because of operator mixing induced by the running of the Wilson coefficients from $m_W \to \Lambda_\chi$ it is important to clearly define the scale at which one wants to apply the assumption of single operator dominance.

Similarly, when considering SMEFT the limits on the WCs can be obtained directly at the matching scale m_W. Here, limits are taken with respect to the dimensionful WC

$$C_i \equiv \tilde{C}_i / \Lambda^{(d-4)} \tag{5.11}$$

and correspondingly the new physics scale can be estimated assuming $\tilde{C}_i = 1$ as

$$\Lambda \simeq (C_i)^{1/(4-d)} \tag{5.12}$$

Both the limits on the different Wilson C_i coefficients as well as the corresponding new physics scales can be accessed from the model classes or directly via

JHEP08(2023)043

LEFT_model.get_limits(half_life, isotope = "76Ge", method = **None**,
groups = **False**, basis = **None**, scale = "up")

functions.get_limits_LEFT(..., unknown_LECs = **False**, PSF_scheme ="A")

SMEFT_model.get_limits(...)

functions.get_limits_SMEFT(..., unknown_LECs = **False**, PSF_scheme ="A")

Parameter	**Type**	**Description**
half_life	float	Experimental half-life limits
isotope	string	Optional - Defines the corresponding isotope the half-life limit is obtained from.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". If called via the `functions` module the preset value is "IBM2". If called via a `model` class he models method will be used if **None**.
groups	bool	Optional - If set to **True** operators with equal limits will be put into groups. Limits are then only shown for these operator groups. If **False** all operator limits will be shown.
scale	string	Optional - Only for LEFT models. Use "up" to get limits at m_W or "down" to get limits at Λ_χ.
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge

Plots of either the operator limits or the corresponding scales can be generated via

JHEP08(2023)043

plots.limits_LEFT(experiments, method = **None**, unknown_LECs = **False**,
PSF_scheme = "A"groups = **False**, plottype = "scales", savefig = **True**,
file = "limits.png", dpi = 300)

plots.limits_SMEFT(...)

Parameter	Type	Description
experiments	dict	Experimental half-life limits. Dictionary of the form {name : { "half-life" : x, "isotope" : x, "label": x}}
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". The preset value is "IBM2".
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge
groups	bool	Optional - If set to **True** operators with equal limits will be put into groups. Limits are then only shown for these operator groups. If **False** all operator limits will be shown.
plottype	string	Optional - Either "scales" or "limits" to plot on the y-axis
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

JHEP08(2023)043

Example: operator Limits obtained from the recent KamLAND-Zen result

[1]:
```
import nudobe
```

[2]:
```
LEFT_lims = nudobe.functions.get_limits_LEFT(2.3e+26, "136Xe",
                                             groups = True)
LEFT_lims
```

[2]:

	Limits	Scales [GeV]
m_bb	3.023437e-11	2.001563e+15
VL(6)	1.556091e-09	6.236166e+06
VR(6)	1.124938e-07	7.334505e+05
T(6)	8.488927e-10	8.443232e+06
S(6)	3.655193e-10	1.286708e+07
V(7)	1.873385e-05	9.262454e+03
S1(9)	2.637535e-05	2.026261e+03
S2(9)	8.871995e-08	6.328938e+03
S3(9)	3.182765e-07	4.902010e+03
S4(9)	5.466169e-08	6.972660e+03
S5(9)	1.609433e-08	8.904315e+03
V(9)	2.697231e-06	3.197065e+03
Vtilde(9)	5.138768e-06	2.810369e+03

[3]:
```
SMEFT_lims = nudobe.functions.get_limits_SMEFT(2.3e+26, "136Xe",
                                               method = "QRPA")
SMEFT_lims
```

[3]:

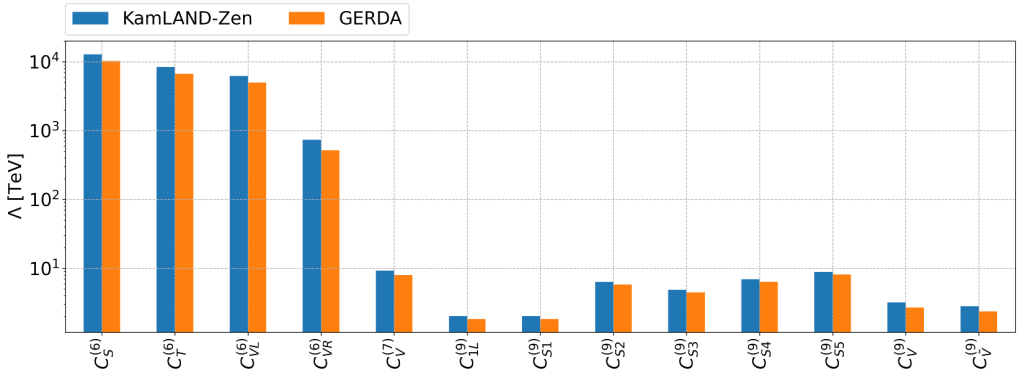
	Limits [GeV^{4-d}]	Scales [GeV]
LH(5)	3.923345e-16	2.548845e+15
LH(7)	1.296631e-20	4.256584e+06
LHD1(7)	2.964838e-13	1.499684e+04
LHD2(7)	2.441054e-12	7.426899e+03
LHDe(7)	7.199730e-17	2.403779e+05
LHW(7)	7.370338e-14	2.385087e+04
LLduD1(7)	1.041539e-15	9.865252e+04
LLQdH1(7)	3.052282e-17	3.199817e+05
LLQdH2(7)	1.530423e-16	1.869517e+05
LLQuH(7)	1.387754e-17	4.161301e+05
LeudH(7)	9.166365e-15	4.778236e+04
ddueue(9)	2.362703e-17	2.115038e+03
dQdueL1(9)	5.435754e-17	1.790392e+03
dQdueL2(9)	3.930148e-17	1.910373e+03

JHEP08(2023)043

```
QudueL1(9)            5.435754e-17  1.790392e+03
QudueL2(9)            3.930148e-17  1.910373e+03
dQQuLL1(9)            1.978746e-20  8.724127e+03
dQQuLL2(9)            6.720479e-20  6.831561e+03
QuQuLL1(9)            5.453917e-20  7.122924e+03
QuQuLL2(9)            1.956554e-19  5.516983e+03
dQdQLL1(9)            5.453917e-20  7.122924e+03
dQdQLL2(9)            1.956554e-19  5.516983e+03
LLH4W1(9)             2.435831e-18  3.331736e+03
deueH2D(9)            6.898672e-20  6.795899e+03
dLuLH2D2(9)           6.898672e-20  6.795899e+03
duLLH2D(9)            3.440686e-20  7.810359e+03
dQLeH2D2(9)           5.579882e-17  1.781045e+03
dLQeH2D1(9)           2.341447e-17  2.118864e+03
deQLH2D(9)            2.567379e-17  2.080185e+03
QueLH2D2(9)           5.579882e-17  1.781045e+03
QeuLH2D2(9)           4.034355e-17  1.900401e+03
QLQLH2D2(9)           1.212675e-17  2.416858e+03
QLQLH2D5(9)           1.212675e-17  2.416858e+03
QQLLH2D2(9)           1.086507e-17  2.470549e+03
eeH4D2(9)             6.224144e-18  2.761748e+03
LLH4D23(9)            3.303900e-18  3.134690e+03
LLH4D24(9)            1.115316e-17  2.457652e+03
```

[4]:
```
nudobe.plots.limits_LEFT(experiments = {"KamLAND" : {"half-life":2.3e+26,
                                                     "isotope" : "136Xe",
                                                     "label":
↪"KamLAND-Zen"
                                                    },
                                        "GERDA"   : {"half-life" : 1.
↪8e+26,
                                                     "isotope" : "76Ge",
                                                     "label" : "GERDA"
                                                    }
                                       },
                         savefig = True,
                         file = "LEFT_limits.png"
                        )
```


JHEP08(2023)043



5.10.2 Two-operator scenario

If we drop the assumption of a single LNV operator the analysis becomes a bit more complicated. Assuming two non-vanishing real WCs $C_{x,y}$ with a relative phase ϕ the half-life is given as

$$T_{1/2}^{-1} = C_x^2 M_{xx} + C_y^2 M_{yy} + 2\mathrm{Re}[C_x C_y \exp(i\phi)] M_{xy} \tag{5.13}$$

Now, instead of calculating the half life in dependency on $C_{x,y}$, $M_{xx,xy,yy}$ and ϕ we can also solve for C_y

$$C_y = -C_x \frac{\cos(\phi) M_{xy}}{M_{yy}} \pm \sqrt{C_x^2 \frac{\cos(\phi) M_{xy}^2 - M_{xx} M_{yy}}{M_{yy}^2} + \frac{1}{T_{1/2} M_{yy}}} \tag{5.14}$$

Given an experimental limit on the half-life, we can use this relation to find the allowed parameter space. Within `nudobe` this is done with the function

---

functions.get_contours(WCx, WCy, half_life, isotope, method = "IBM2",
phase = 3/4*np.pi, n_points = 5000, x_min = **None**, x_max = **None**,
unknown_LECs = **False**, PSF_scheme ="A")

Parameter	Type	Description
WCx	string	Independent Wilson coefficient. Can be from SMEFT or LEFT.
WCy	string	Dependent Wilson coefficient to be calculated from WCx and the half-life limit. Should be from the same EFT as WCx.
half_life	float	Experimental half-life limits

JHEP08(2023)043

isotope	string	Defines the corresponding isotope the half-life limit is obtained from.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". The preset value is "IBM2".
n_points	float	Optional - Sets the number of points to be calculated.
phase	float	Optional - Sets the relative phase ϕ.
x_min	float	Optional - Sets the minimum value of WCx to be studied.
x_max	float	Optional - Sets the maximum value of WCx to be studied.
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge

The resulting contours can be plotted via

```
plots.contours(WCx, WCy,
        limits = {"KamLAND-Zen": {"half-life" : 2.3e+26, "isotope" : "136Xe",
                  "color" : "b", "label" : "KamLAND-Zen", "linewidth" : 1,
                  "linealpha" : 1, "linestyle" : "-", "alpha" : None} },
        method = "IBM2", unknown_LECs = False, PSF_scheme ="A",
        n_points = 5000, phase = 0, varyphases = False, n_vary = 5, x_min = None,
        x_max = None, savefig = False, file = "contour_limits.png", dpi = 300 )
```

Parameter	Type	Description
WCx	string	Independent Wilson coefficient. Can be from SMEFT or LEFT.
WCy	string	Dependent Wilson coefficient to be calculated from WCx and the half-life limit. Should be from the same EFT as WCx.

JHEP08(2023)043

limits	dictionary	Experimental half-life limits. Given as a dictionary of the type {Name : [half-life limit, isotope, color, label, linewidth, linealpha, linestyle, alpha]}. The color will decide the color of the plot while the label sets the label for the legend. Linealpha sets the alpha channel for the outer line of the contour. If it is not defined it will be set equal to alpha which defines the alpha channel of the filled area. If alpha is not set or set to **None** it is set automatically.
method	string	Optional - Sets the NME calculation method. You can choose from "IBM2", "SM" and "QRPA". The preset value is "IBM2".
unknown_LECs	bool	Optional - If set to **True** the unknown LECs will be set to their NDA estimates (see table 4). If set to **False** the unknown LECs will be turned off i.e. set to 0.
PSF_scheme	string	Optional - Choose PSFs and electron wave functions - "A": approximate solution to a uniform charge distribution. "B": exact solution to a point-like charge
n_points	float	Optional - Sets the number of points to be calculated.
phase	float	Optional - Sets the relative phase ϕ.
varyphases	bool	Optional - If **True** the relative complex phase will be varied and multiple plots will be generated.
n_vary	integer	Optional - Number of variations of the complex phase.
x_min	float	Optional - Sets the minimum value of WCx to be studied.
x_max	float	Optional - Sets the maximum value of WCx to be studied.
savefig	bool	Optional - If **True** save figure as file
file	string	Optional - Filename to save figure to.
dpi	float	Optional - sets the resolution in dots per inch when saving the figure.

JHEP08(2023)043

Example: the light-neutrino-exchange mechanism with additional lepton number violating right-handed current

Again, we take a look at our standard example and want to study the combined limits on $m_{\beta\beta}$ and $C_{VR}^{(6)}$. We can get the lower and upper limits on $C_{VR}^{(6)}$ for different values of $m_{\beta\beta}$ via

[1]:
```
import nudobe
```

[2]:
```
contours = nudobe.functions.get_contours("m_bb", "VR(6)", 2.3e+26,
                                         "136Xe", phase = 0)
contours
```

[2]:

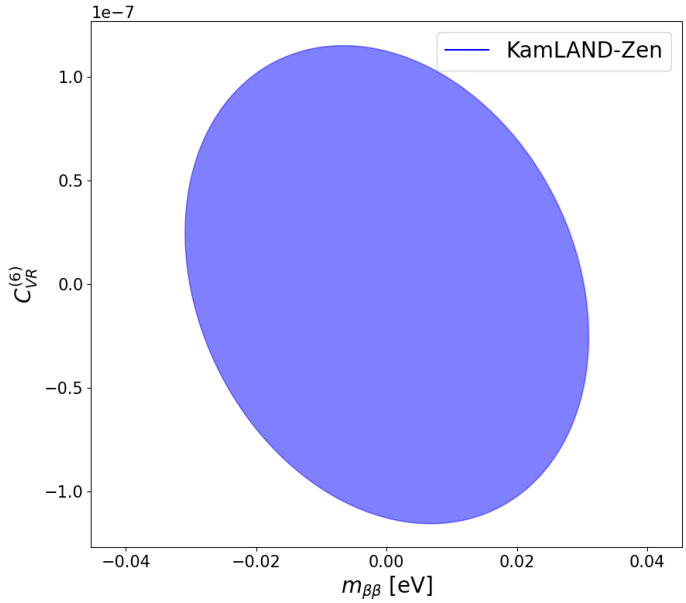
	m_bb	VR(6) min	VR(6) max
0	-3.096316e-11	2.304978e-08	2.678719e-08
1	-3.094502e-11	2.062428e-08	2.918349e-08
2	-3.092687e-11	1.913349e-08	3.064508e-08
3	-3.090873e-11	1.795119e-08	3.179817e-08
4	-3.089058e-11	1.693975e-08	3.278041e-08
...	...	...	...
3409	3.089058e-11	-3.278041e-08	-1.693975e-08
3410	3.090873e-11	-3.179817e-08	-1.795119e-08
3411	3.092687e-11	-3.064508e-08	-1.913349e-08
3412	3.094502e-11	-2.918349e-08	-2.062428e-08
3413	3.096316e-11	-2.678719e-08	-2.304978e-08

[3414 rows x 3 columns]

And a plot showing the contours of the allowed parameter space resulting from the recent KamLAND-Zen result is generated via

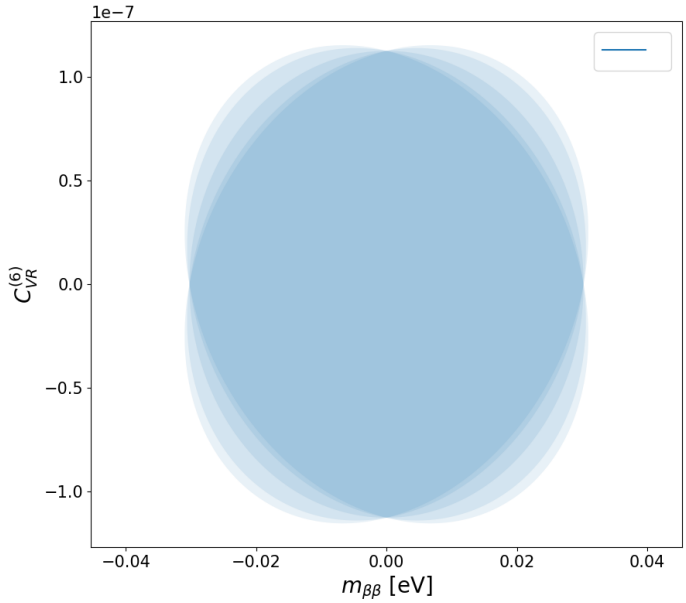
[3]:
```
nudobe.plots.contours("m_bb", "VR(6)", phase = 0,
                      limits = {"KamLAND-Zen" : {"half-life" : 2.3e+26,
                                "isotope"    : "136Xe",
                                "color"      : "b",
                                "label"      :"KamLAND-Zen"}})
```


JHEP08(2023)043



With variations of ϕ the resulting allowed parameter space can be plotted via

[4]:
```
nudobe.plots.contours("m_bb", "VR(6)",
                      limits = {"KamLAND-Zen" : {"half-life" : 2.3e+26,
                                "isotope"   : "136Xe"}
                               },
                      varyphases = True)
```

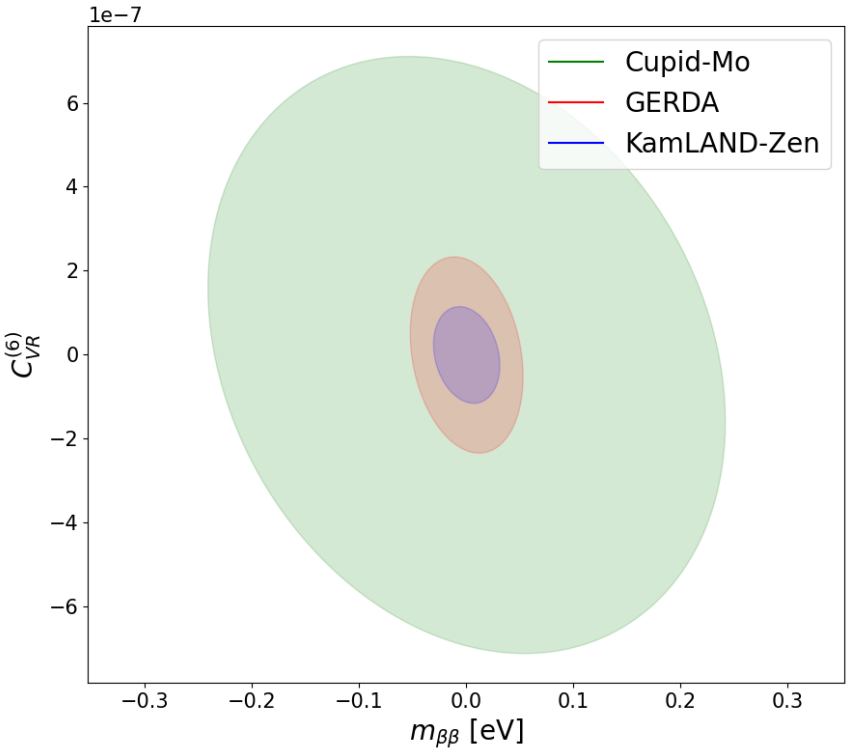


Now, instead of one area, the plot shows 5 different contour areas each corresponding to a different choice of the complex relative phase Φ.

JHEP08(2023)043

And limits from multiple different experiments in different isotopes can be plotted via

[5]:
```
nudobe.plots.contours("m_bb", "VR(6)",
phase = 0,
limits = {"KamLAND-Zen" : {"half-life" : 2.3e+26,
                           "isotope"   : "136Xe",
                           "color"     : "b",
                           "label"     :"KamLAND-Zen"},
          "GERDA"       : {"half-life" : 1.8e+26,
                           "isotope"   : "76Ge",
                           "color"     : "r",
                           "label"     : "GERDA"},
          "Cupid-Mo"    : {"half-life" : 1.5e+24,
                           "isotope"   : "100Mo",
                           "color"     : "g",
                           "label"     : "Cupid-Mo"}})
```



Because of different NMEs related to the different isotopes, the contours not only have different radii but also different orientations and shapes.

JHEP08(2023)043

A List of operators

In total, νDoBe contains 32 different LEFT operators and 36 different SMEFT operators that contribute to $0\nu\beta\beta$. The relevant LEFT operators up to dimension 9 have been obtained in [31, 32] and are summarized in table 5. The general LEFT Lagrangian can be written as

$$\mathcal{L}_{\text{LEFT}} = -\frac{1}{2} m_{\beta\beta} \overline{\nu_L^C} \nu_L + \frac{1}{v^2} \sum_i C_i^{(6)} \mathcal{O}_i^{(6)} + \frac{1}{v^3} \sum_j C_j^{(7)} \mathcal{O}_j^{(7)} + \frac{1}{v^5} \sum_k C_k^{(9)} \mathcal{O}_k^{(9)} \tag{A.1}$$

In the SMEFT we consider all LNV operators up to dimension 7 and all dimension 9 operators that match onto LEFT dimension 9 [37]. The relevant SMEFT operators are summarized in table 6 and the SMEFT Lagrangian can be written as

$$\mathcal{L}_{\text{SMEFT}} = C_{LH}^{(5)} \mathcal{O}_{LH}^{(5)} + \sum_i C_i^{(7)} \mathcal{O}_i^{(7)} + \sum_j C_j^{(9)} \mathcal{O}_j^{(9)} \tag{A.2}$$

Therefore, we define the Wilson coefficients of LEFT as dimensionless numbers while the Wilson coefficients of the SMEFT are dimensionful, i.e., $C^d_{i,\text{SMEFT}} \propto \Lambda^{4-d}$. We choose this somewhat awkward convention because it is done in most of the $0\nu\beta\beta$ literature.

In defining the operator basis, we use the following conventions: the partial derivative acting to the left and right is defined as

$$\overline{\Psi_1} \overleftrightarrow{\partial}_\mu \Psi_2 = \overline{\Psi_1} \left(\partial_\mu \Psi_2\right) - \left(\partial_\mu \overline{\Psi_1}\right) \Psi_2 \tag{A.3}$$

The covariant derivative used to define the SMEFT operators (ignoring $SU(3)$) is given by

$$D_\mu = \left(\partial_\mu - i\frac{g_2}{2} \tau^I W_\mu^I - i g_1 Y B_\mu\right) , \tag{A.4}$$

with the Pauli matrices labeled via $\tau^I, I = 1, 2, 3$. The generators of $SU(3)$, that are present in the definition of several LEFT operators, are denoted by

$$t_a = \frac{\lambda_a}{2} , \tag{A.5}$$

where $\lambda_a, a = 1, \ldots, 8$ represent the Gell-Mann matrices.

JHEP08(2023)043

Class	Op. Name	Code Label	Op. Structure	Half-Life Group	SMEFT Dim.
			Dim 3		
Ψ^2	$\mathcal{O}_{m_{\beta\beta}}$	m_bb	$-\frac{1}{2}m_{ee}\overline{\nu_{L,e}^C}\nu_{L,e}$	$m_{\beta\beta}$	5
			Dim 6		
Ψ^4	$\mathcal{O}_{SL}^{(6)}$	SL(6)	$[\overline{u_R}d_L][\overline{e_L}\nu_L^C]$	$C_S^{(6)}$	7
	$\mathcal{O}_{SR}^{(6)}$	SR(6)	$[\overline{u_L}d_R][\overline{e_L}\nu_L^C]$	$C_S^{(6)}$	7
	$\mathcal{O}_{VL}^{(6)}$	VL(6)	$[\overline{u_L}\gamma^\mu d_L][\overline{e_R}\gamma_\mu\nu_L^C]$	$C_{VL}^{(6)}$	7
	$\mathcal{O}_{VR}^{(6)}$	VR(6)	$[\overline{u_R}\gamma^\mu d_R][\overline{e_R}\gamma_\mu\nu_L^C]$	$C_{VR}^{(6)}$	7
	$\mathcal{O}_{T}^{(6)}$	T(6)	$[\overline{u_L}\sigma^{\mu\nu}d_R][\overline{e_L}\sigma_{\mu\nu}\nu_L^C]$	$C_T^{(6)}$	7
			Dim 7		
$\Psi^4\partial$	VL(7)	$\mathcal{O}_{VL}^{(7)}$	$[\overline{u_L}\gamma^\mu d_L][\overline{e_L}\overleftrightarrow{\partial}_\mu\nu_L^C]$	$C_V^{(7)}$	7
	$\mathcal{O}_{VR}^{(7)}$	VR(7)	$[\overline{u_R}\gamma^\mu d_R][\overline{e_L}\overleftrightarrow{\partial}_\mu\nu_L^C]$	$C_V^{(7)}$	7
			Dim 9		
Ψ^6	$\mathcal{O}_{1L}^{(9)}$	1L(9))	$[\overline{u_L}\gamma_\mu d_L][\overline{u_L}\gamma^\mu d_L][\overline{e_L}e_L^C]$	$C_{S1}^{(9)}$	7
	$\mathcal{O}_{1R}^{(9)}$	1R(9)	$[\overline{u_R}\gamma_\mu d_R][\overline{u_R}\gamma^\mu d_R][\overline{e_R}e_R^C]$	$C_{S1}^{(9)}$	9
	$\mathcal{O}_{1L}^{(9)\prime}$	1L(9)prime	$[\overline{u_R}\gamma_\mu d_R][\overline{u_R}\gamma^\mu d_R][\overline{e_L}e_L^C]$	$C_{S1}^{(9)}$	11
	$\mathcal{O}_{1R}^{(9)\prime}$	1R(9)prime	$[\overline{u_R}\gamma_\mu d_R][\overline{u_R}\gamma^\mu d_R][\overline{e_R}e_R^C]$	$C_{S1}^{(9)}$	9
	$\mathcal{O}_{2L}^{(9)}$	2L(9)	$[\overline{u_R}d_L][\overline{u_R}d_L][\overline{e_L}e_R^C]$	$C_{S2}^{(9)}$	9
	$\mathcal{O}_{2R}^{(9)}$	2R(9)	$[\overline{u_R}d_L][\overline{u_R}d_L][\overline{e_R}e_R^C]$	$C_{S2}^{(9)}$	11
	$\mathcal{O}_{2L}^{(9)\prime}$	2L(9)prime	$[\overline{u_L}d_R][\overline{u_L}d_R][\overline{e_L}e_L^C]$	$C_{S2}^{(9)}$	9
	$\mathcal{O}_{2R}^{(9)\prime}$	2R(9)prime	$[\overline{u_L}d_R][\overline{u_L}d_R][\overline{e_R}e_R^C]$	$C_{S2}^{(9)}$	11
	$\mathcal{O}_{3L}^{(9)}$	3L(9)	$[\overline{u_R}^\alpha d_L^\beta][\overline{u_R}^\beta d_L^\alpha][\overline{e_L}e_L^C]$	$C_{S3}^{(9)}$	9
	$\mathcal{O}_{3R}^{(9)}$	3R(9)	$[\overline{u_R}^\alpha d_L^\beta][\overline{u_R}^\beta d_L^\alpha][\overline{e_R}e_R^C]$	$C_{S3}^{(9)}$	11
	$\mathcal{O}_{3L}^{(9)\prime}$	3L(9)prime	$[\overline{u_L}^\alpha d_R^\beta][\overline{u_L}^\beta d_R^\alpha][\overline{e_L}e_L^C]$	$C_{S3}^{(9)}$	9
	$\mathcal{O}_{3R}^{(9)\prime}$	3R(9)prime	$[\overline{u_L}^\alpha d_R^\beta][\overline{u_L}^\beta d_R^\alpha][\overline{e_R}e_R^C]$	$C_{S3}^{(9)}$	11
	$\mathcal{O}_{4L}^{(9)}$	4L(9)	$[\overline{u_L}\gamma^\mu d_L][\overline{u_R}\gamma_\mu d_R][\overline{e_L}e_L^C]$	$C_{S4}^{(9)}$	7
	$\mathcal{O}_{4R}^{(9)}$	4R(9)	$[\overline{u_L}\gamma^\mu d_L][\overline{u_R}\gamma_\mu d_R][\overline{e_R}e_R^C]$	$C_{S4}^{(9)}$	9
	$\mathcal{O}_{5L}^{(9)}$	5L(9)	$[\overline{u_L}^\alpha\gamma^\mu d_L^\beta][\overline{u_R}^\beta\gamma_\mu d_R^\alpha][\overline{e_L}e_L^C]$	$C_{S5}^{(9)}$	9
	$\mathcal{O}_{5R}^{(9)}$	5R(9)	$[\overline{u_L}^\alpha\gamma^\mu d_L^\beta][\overline{u_R}^\beta\gamma_\mu d_R^\alpha][\overline{e_R}e_R^C]$	$C_{S5}^{(9)}$	11
	$\mathcal{O}_{6}^{(9)}$	6(9)	$[\overline{u_L}\gamma_\mu d_L][\overline{u_L}d_R][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{V}^{(9)}$	9
	$\mathcal{O}_{6}^{(9)\prime}$	6(9)prime	$[\overline{u_R}\gamma_\mu d_R][\overline{u_R}d_L][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{V}^{(9)}$	9
	$\mathcal{O}_{7}^{(9)}$	7(9)	$[\overline{u_L}t^A\gamma_\mu d_L][\overline{u_L}t^A d_R][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{\tilde{V}}^{(9)}$	9
	$\mathcal{O}_{7}^{(9)\prime}$	7(9)prime	$[\overline{u_R}t^A\gamma_\mu d_R][\overline{u_R}t^A d_L][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{\tilde{V}}^{(9)}$	9
	$\mathcal{O}_{8}^{(9)}$	8(9)	$[\overline{u_L}\gamma_\mu d_L][\overline{u_R}d_L][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{V}^{(9)}$	9
	$\mathcal{O}_{8}^{(9)\prime}$	8(9)prime	$[\overline{u_R}\gamma_\mu d_R][\overline{u_L}d_R][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{V}^{(9)}$	9
	$\mathcal{O}_{9}^{(9)}$	9(9)	$[\overline{u_L}t^A\gamma_\mu d_L][\overline{u_R}t^A d_L][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{\tilde{V}}^{(9)}$	9
	$\mathcal{O}_{9}^{(9)\prime}$	9(9)prime	$[\overline{u_R}t^A\gamma_\mu d_R][\overline{u_L}t^A d_R][\overline{e}\gamma^\mu\gamma_5 e^C]$	$C_{\tilde{V}}^{(9)}$	9

Table 5. LEFT operators contributing to $0\nu\beta\beta$-decay at tree level. The primed dimension 9 operators represent a parity flip in the quark currents. The "Code Label" column shows the name of the operators used in νDoBe. Additionally, the "Half-Life Group" column identifies operators that result in the same half-life. Finally, the "SMEFT Dim." column shows the lowest SMEFT dimension an operator can originate from at the SMEFT→LEFT matching scale m_W. When evolving the operators down to the chiral scale Λ_χ, mixing between the different LEFT operators can occur such that, e.g., the operators $\mathcal{O}_{5L,R}^{(9)}$ receive contributions from SMEFT dim 7 and 9 at Λ_χ.

JHEP08(2023)043

Class	Op. Name	Code Label	Op. Structure	LEFT Matching
			Dim 5	
Ψ^2H^2	$\mathcal{O}_{LH}^{(5)}$	LH(5)	$\epsilon_{ij}\epsilon_{kl}[\overline{L_i^C}L_k][H_jH_n]$	$m_{\beta\beta}$
			Dim 7	
Ψ^2H^4	$\mathcal{O}_{LH}^{(7)}$	LH(7)	$\epsilon_{ij}\epsilon_{kl}[\overline{L_i^C}L_k][H_jH_n][H^\dagger H]$	$m_{\beta\beta}$
Ψ^2H^3D	$\mathcal{O}_{LHDe}^{(7)}$	LHDe(7)	$\epsilon_{ij}\epsilon_{kl}[\overline{L_i^C}\gamma^\mu e][H_jH_k][D_\mu H]_l$	$\mathcal{O}_{VL}^{(6)}$
$\Psi^2H^2D^2$	$\mathcal{O}_{LHD1}^{(7)}$	LHD1(7)	$\epsilon_{ij}\epsilon_{kl}[\overline{L_i^C}(D_\mu L)_j][H_k(D_\mu H)_l]$	$\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$
	$\mathcal{O}_{LHD2}^{(7)}$	LHD2(7)	$\epsilon_{ik}\epsilon_{jl}[\overline{L_i^C}(D_\mu L)_j][H_k(D_\mu H)_l]$	$\mathcal{O}_{SL}^{(6)},\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VL}^{(7)}$
Ψ^2H^2W	$\mathcal{O}_{LHW}^{(7)}$	LHW(7)	$\epsilon_{ij}\epsilon_{km}\tau^I_{ml}g_2[\overline{L_i^C}\sigma^{\mu\nu}L_k][H_jH_l]W^I_{\mu\nu}$	$\mathcal{O}_{VL}^{(6)},\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$
Ψ^4D	$\mathcal{O}_{LLduD1}^{(7)}$	LLduD1(7)	$\epsilon_{ij}[\overline{d}\gamma^\mu u][\overline{L_i^C}(D_\mu L)_j]$	$\mathcal{O}_{VR}^{(7)},\mathcal{O}_{4L}^{(9)}$
Ψ^4H	$\mathcal{O}_{LLQdH1}^{(7)}$	LLQdH1(7)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}L_i][\overline{Q^C}_jL_k]H_l$	$\mathcal{O}_{SR}^{(6)},\mathcal{O}_{T}^{(6)}$
	$\mathcal{O}_{LLQdH2}^{(7)}$	LLQdH2(7)	$\epsilon_{ij}\epsilon_{kl}[\overline{d}L_i][\overline{Q^C}_jL_k]H_l$	$\mathcal{O}_{T}^{(6)}$
	$\mathcal{O}_{LLQuH}^{(7)}$	LLQuH(7)	$\epsilon_{ij}[\overline{Q}_ku][\overline{L^C}_kL_i]H_j$	$\mathcal{O}_{SL}^{(6)}$
	$\mathcal{O}_{LeudH}^{(7)}$	LeudH(7)	$\epsilon_{ij}[\overline{L^C}_i\gamma^\mu e][\overline{d}\gamma_\mu u]H_j$	$\mathcal{O}_{VR}^{(6)}$
			Dim 9	
Ψ^6	$\mathcal{O}_{ddueue}^{(9)}$	ddueue(9)	$[\overline{d^\alpha}d^{C\beta}][\overline{u^{C\alpha}}e][\overline{u^{C\beta}}e]$	$\mathcal{O}_{1R}^{(9)\prime}$
	$\mathcal{O}_{dQdueL1}^{(9)}$	dQdueL1(9)	$\epsilon_{ij}[\overline{d}Q_i][\overline{d}\gamma^\mu u][\overline{e^C}\gamma_\mu L_j]$	$\mathcal{O}_{8}^{(9)\prime}$
	$\mathcal{O}_{dQdueL2}^{(9)}$	dQdueL2(9)	$\epsilon_{ij}[\overline{d^\alpha}Q_i^\beta][\overline{d^\beta}\gamma^\mu u^\alpha][\overline{e^C}\gamma_\mu L_j]$	$\mathcal{O}_{8}^{(9)\prime},\mathcal{O}_{9}^{(9)\prime}$
	$\mathcal{O}_{QudueL1}^{(9)}$	QdueL1(9)	$[\overline{Q}u][\overline{d}\gamma^\mu u][\overline{e^C}\gamma_\mu L_j]$	$\mathcal{O}_{6}^{(9)\prime}$
	$\mathcal{O}_{QudueL2}^{(9)}$	QdueL2(9)	$[\overline{Q^\alpha}u^\beta][\overline{d^\beta}\gamma^\mu u^\alpha][\overline{e^C}\gamma_\mu L_j]$	$\mathcal{O}_{6}^{(9)\prime},\mathcal{O}_{7}^{(9)\prime}$
	$\mathcal{O}_{dQdQLL1}^{(9)}$	dQdQLL1(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}Q_i][\overline{d}\gamma^\mu Q_j][\overline{L_k^C}\gamma_\mu L_l]$	$\mathcal{O}_{2L}^{(9)\prime}$
	$\mathcal{O}_{dQdQLL2}^{(9)}$	dQdQLL2(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d^\alpha}Q_i^\beta][\overline{d^\beta}\gamma^\mu Q_j^\alpha][\overline{L_k^C}\gamma_\mu L_l]$	$\mathcal{O}_{3L}^{(9)\prime}$
	$\mathcal{O}_{dQQuLL1}^{(9)}$	dQQuLL1(9)	$\epsilon_{ij}[\overline{d}Q_i][\overline{Q}u][\overline{L^C}L_j]$	$\mathcal{O}_{5L}^{(9)}$
	$\mathcal{O}_{dQQuLL2}^{(9)}$	dQQuLL2(9)	$\epsilon_{ij}[\overline{d^\alpha}Q_i^\beta][\overline{Q^\beta}u^\alpha][\overline{L^C}L_j]$	$\mathcal{O}_{4L}^{(9)}$
	$\mathcal{O}_{QuQuLL1}^{(9)}$	QuQuLL1(9)	$[\overline{Q_i}u][\overline{Q_j}u][\overline{L_i^C}L_j]$	$\mathcal{O}_{2L}^{(9)}$
	$\mathcal{O}_{QuQuLL2}^{(9)}$	QuQuLL2(9)	$[\overline{Q_i^\alpha}u^\beta][\overline{Q_j^\beta}u^\alpha][\overline{L_i^C}L_j]$	$\mathcal{O}_{3L}^{(9)}$
Ψ^2H^4W	$\mathcal{O}_{LLH^4W1}^{(9)}$	LLH4W1(9)	$\epsilon_{ij}\epsilon_{km}\tau^I_{ml}g_2[\overline{L_i^C}\sigma^{\mu\nu}L_k][H_jH_l]W^I_{\mu\nu}[H^\dagger H]$	$\mathcal{O}_{VL}^{(6)},\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$
$D^2\Psi^2H^4$	$\mathcal{O}_{eeH^4D^2}^{(9)}$	eeH4D2(9)	$\epsilon_{ij}\epsilon_{kl}[\overline{e^C}e][H_i(D_\mu H)_j][H_k(D^\mu H)_l]$	$\mathcal{O}_{1R}^{(9)}$
	$\mathcal{O}_{LLH^4D^23}^{(9)}$	LLH4D23(9)	$\epsilon_{ik}\epsilon_{jl}[(D_\mu\overline{L^C})_i(D^\mu L)_j][H_kH_l][H^\dagger H]$	$\mathcal{O}_{SL}^{(6)},\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$
	$\mathcal{O}_{LLH^4D^24}^{(9)}$	LLH4D24(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{L^C}_i(D^\mu L)_j][(D_\mu H)_kH_l][H^\dagger H]$	$\mathcal{O}_{SL}^{(6)},\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$
$D\Psi^4H^2$	$\mathcal{O}_{deueH^2D}^{(9)}$	deueH2D(9)	$\epsilon_{ij}[\overline{d}\gamma^\mu e][\overline{u^C}e][H_i(iD_\mu H)_j]$	$\mathcal{O}_{4R}^{(9)}$
	$\mathcal{O}_{dQLeH^2D2}^{(9)}$	dQLeH2D2(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}Q_i][\overline{L_j^C}\gamma^\mu e][H_k(iD_\mu H)_l]$	$\mathcal{O}_{6}^{(9)}$
	$\mathcal{O}_{dLQeH^2D1}^{(9)}$	dLQeH2D1(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}L_i][\overline{Q_j^C}\gamma^\mu e][(iD_\mu H)_kH_l]$	$\mathcal{O}_{6}^{(9)},\mathcal{O}_{7}^{(9)}$
	$\mathcal{O}_{dLuLH^2D2}^{(9)}$	dLuLH2D2(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}L_i][\overline{u^C}\gamma_\mu L_j][\tilde{H}_k(iD^\mu H)_l]$	$\mathcal{O}_{4L}^{(9)}$
	$\mathcal{O}_{duLLH^2D}^{(9)}$	duLLH2D(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}\gamma_\mu u][\overline{L^C}_i(iD^\mu L)_j][\tilde{H}_kH_l]$	$\mathcal{O}_{SL}^{(6)},\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VR}^{(7)},\mathcal{O}_{4L}^{(9)}$
	$\mathcal{O}_{deQLH^2D}^{(9)}$	deQLH2D(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{d}\gamma^\mu e][\overline{Q_i^C}(iD_\mu L)_j][H_kH_l]$	$\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VL}^{(6)}\mathcal{O}_{VR}^{(6)},\mathcal{O}_{T}^{(6)},$ $\mathcal{O}_{6}^{(9)},\mathcal{O}_{7}^{(9)},(\mathcal{O}_{VR}^{(7)\prime})$
	$\mathcal{O}_{QueLH^2D2}^{(9)}$	QueLH2D2(9)	$\epsilon_{jk}[\overline{Q_i}u][\overline{u^C}\gamma^\mu L_j][H_i(iD_\mu H)_k]$	$\mathcal{O}_{8}^{(9)}$
	$\mathcal{O}_{QeuLH^2D2}^{(9)}$	QeuLH2D2(9)	$\delta_{ik}\epsilon_{jl}[\overline{Q_i}e][\overline{u^C}\gamma^\mu L_j][H_k(iD_\mu H)_l]$	$\mathcal{O}_{8}^{(9)},\mathcal{O}_{9}^{(9)}$
	$\mathcal{O}_{QLQLH^2D2}^{(9)}$	QLQLH2D2(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{Q}\gamma^\mu L][\overline{Q_i^C}\gamma^\mu L_j][H_k(iD_\mu H)_l]$	$\mathcal{O}_{1L}^{(9)}$
	$\mathcal{O}_{QLQLH^2D5}^{(9)}$	QLQLH2D5(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{Q}\gamma^\mu L_i][\overline{Q_j^C}\gamma^\mu L][(iD_\mu H)_kH_l]$	$\mathcal{O}_{1L}^{(9)}$
	$\mathcal{O}_{QQLLH^2D2}^{(9)}$	QQLLH2D2(9)	$\epsilon_{ik}\epsilon_{jl}[\overline{Q}\gamma^\mu Q_i][\overline{L^C}(iD_\mu L)_j][H_kH_l]$	$\mathcal{O}_{SL}^{(6)},\mathcal{O}_{SR}^{(6)},\mathcal{O}_{VL}^{(7)},\mathcal{O}_{1L}^{(9)}$

Table 6. Here we show the different SMEFT dimension 9 operators that we considered. Again, the "Code Label" column shows the name of the operators as used in νDoBe. The "LEFT Matching" column provides a list of LEFT operators each SMEFT operator matches onto with the precise matching given in appendix B.

JHEP08(2023)043

B Operator matching

The transition amplitudes and correspondingly all $0\nu\beta\beta$ observables calculated by νDoBe are evaluated at LEFT level. Hence, models which are generated at SMEFT level first have to be matched down onto LEFT. At SMEFT level lepton number violation by two units without baryon number violation occurs only at odd dimensions [34]. Here we provide the explicit matching conditions at the matching scale $E = m_W$

$$
\begin{aligned}
m_{\beta\beta} =& -v^2 C_{LH}^{(5)} - \frac{v^4}{2} C_{LH}^{(7)} \,, \\
C_{SL}^{(6)} =& v^3 \left(\frac{1}{\sqrt{2}} C_{LLQuH1}^{(7)*} + \frac{m_u}{v} \frac{V_{ud}}{2} C_{LHD2}^{(7)*} \right) \\
&+ v^4 \Bigg(+ m_u \frac{V_{ud}}{2} C_{LLH^4D^23}^{(9)*} - m_u \frac{V_{ud}}{4} C_{LLH^4D^24}^{(9)*} \\
&\qquad - m_u \frac{1}{4} C_{QQLLH^2D2}^{(9)*} - m_d \frac{1}{4} C_{duLLH^2D}^{(9)*} \Bigg) \,, \\
C_{SR}^{(6)} =& v^3 \left(\frac{1}{2\sqrt{2}} C_{LLQdH1}^{(7)*} - \frac{V_{ud}}{2} \frac{m_d}{v} C_{LHD2}^{(7)*} \right) \\
&+ v^4 \Bigg(- m_d \frac{V_{ud}}{2} C_{LLH^4D^23}^{(9)*} + m_d \frac{V_{ud}}{4} C_{LLH^4D^24}^{(9)*} + \frac{m_d}{4} C_{QQLLH^2D2}^{(9)*} \\
&\qquad + \frac{m_u}{4} C_{duLLH^2D}^{(9)} + \frac{m_e}{8} C_{deQLH^2D}^{(9)*} \Bigg) \,, \\
C_{VL}^{(6)} =& v^3 \left(- \frac{i}{\sqrt{2}} V_{ud} C_{LHDe}^{(7)*} + 4 \frac{m_e}{v} C_{LHW}^{(7)*} \right) \\
&+ v^4 \left(2 m_e V_{ud} C_{LLH^4W1}^{(9)*} - \frac{m_d}{4} C_{deQLH^2D}^{(9)*} \right) \,, \\
C_{VR}^{(6)} =& v^3 \frac{1}{\sqrt{2}} C_{LeudH}^{(7)*} - v^4 \frac{m_u}{4} C_{deQLH^2D}^{(9)*} \,, \\
C_{T}^{(6)} =& v^3 \left(\frac{1}{8\sqrt{2}} C_{LLQdH1}^{(7)*} + \frac{1}{4\sqrt{2}} C_{LLQdH2}^{(7)*} \right) + v^4 \frac{m_e}{16} C_{deQLH^2D}^{(9)*} \,, \\
C_{VL}^{(7)} =& v^3 \left(V_{ud} C_{LHD1}^{(7)*} - \frac{V_{ud}}{2} C_{LHD2}^{(7)*} + 4 V_{ud} C_{LHW}^{(7)*} \right) \\
&+ v^5 \left(2 V_{ud} C_{LLH^4W1}^{(9)*} + \frac{V_{ud}}{2} C_{LLH^4D^23}^{(9)*} - \frac{V_{ud}}{4} C_{LLH^4D^24}^{(9)*} - \frac{1}{4} C_{QQLLH^2D2}^{(9)*} \right) \,, \\
C_{VR}^{(7)} =& v^3 \left(- i C_{LLduD1}^{(7)*} \right) + v^5 \left(\frac{1}{4} C_{duLLH^2D}^{(9)*} \right) \,,
\end{aligned}
\tag{B.1}
$$

for the mass-mechanism and the long-range operators and

JHEP08(2023)043

$$
\begin{aligned}
C_{1L}^{(9)} =& v^3\Big(2V_{ud}^2 C_{LHD1}^{(7)*} + 8V_{ud}^2 C_{LHW}^{(7)*}\Big) \\
&+ v^5\Big(4V_{ud}^2 C_{LLH^4W1}^{(9)*} - V_{ud}^2 C_{LLH^4D^23}^{(9)*} - V_{ud}^2 C_{LLH^4D^24}^{(9)*} - V_{ud} C_{QQLLH^2D2}^{(9)*} \\
&\quad - \frac{V_{ud}}{2} C_{QLQLH^2D2}^{(9)*} - \frac{V_{ud}}{2} C_{QLQLH^2D5}^{(9)*}\Big) , \\
C_{1R}^{(9)} =& - v^5 V_{ud}^2 C_{eeH^4D^2}^{(9)*} , \quad C_{1R}^{(9)\prime} = \frac{v^5}{4} C_{ddueue}^{(9)*} , \\
C_{2L}^{(9)} =& - v^5 C_{QuQuLL1}^{(9)*} , \quad C_{2L}^{(9)\prime} = -v^5 C_{dQdQLL1}^{(9)*} , \\
C_{3L}^{(9)} =& - v^5 C_{QuQuLL2}^{(9)*} , \quad C_{3L}^{(9)\prime} = -v^5 C_{dQdQLL2}^{(9)*} , \\
C_{4L}^{(9)} =& - v^3 i2V_{ud} C_{LLduD1}^{(7)*} + v^5\Big(V_{ud} C_{duLLH^2D}^{(9)*} - \frac{V_{ud}}{2} C_{dLuLH^2D2}^{(9)*} - \frac{1}{2} C_{dQQuLL2}^{(9)*}\Big) , \\
C_{4R}^{(9)} =& - v^5 \frac{V_{ud}}{2} C_{deueH^2D}^{(9)*} , \\
C_{5L}^{(9)} =& - \frac{1}{2} v^5 C_{dQQuLL1}^{(9)*} , \\
C_{6}^{(9)} =& v^5\Big(- \frac{2}{3} V_{ud} C_{dLQeH^2D1}^{(9)*} + \frac{V_{ud}}{2} C_{dQLeH^2D2}^{(9)*} - \frac{5}{12} V_{ud} C_{deQLH^2D}^{(9)*}\Big) , \\
C_{6}^{(9)\prime} =& v^5\Big(\frac{1}{6} C_{QudueL2}^{(9)*} + \frac{1}{2} C_{QudueL1}^{(9)*}\Big) , \\
C_{7}^{(9)} =& v^5\Big(- V_{ud} C_{dLQeH^2D1}^{(9)} - V_{ud} C_{deQLH^2D}^{(9)*}\Big) , \quad C_{7}^{(9)\prime} = v^5 C_{QudueL2}^{(9)*} , \\
C_{8}^{(9)} =& v^5\Big(- \frac{V_{ud}}{2} C_{QueLH^2D2}^{(9)*} + \frac{V_{ud}}{6} C_{QeuLH^2D2}^{(9)*}\Big) , \\
C_{8}^{(9)\prime} =& v^5\Big(\frac{1}{6} C_{dQdueL2}^{(9)*} + \frac{1}{2} C_{dQdueL1}^{(9)*}\Big) , \\
C_{9}^{(9)} =& v^5 V_{ud} C_{QeuLH^2D2}^{(9)*} , \quad C_{9}^{(9)\prime} = v^5 C_{dQdueL2}^{(9)*} ,
\end{aligned}
\tag{B.2}
$$

for the short-range LEFT operators. The dim-9 SMEFT operator $\mathcal{O}_{deQLH^2D}$, additionally, matches onto the low-energy $\Delta_L = 2$ operator $\mathcal{O}_{VR}^{(7)\prime} = [\overline{e_R}\gamma^\mu \nu_L^C][\overline{u_L} i\overset{\leftrightarrow}{\partial}_\mu d_R]$. To study the relevance of $\mathcal{O}_{VR}^{(7)\prime}$ we can be express it in terms of $\mathcal{O}_{VL}^{(6)}, \mathcal{O}_{VR}^{(6)}$ and an additional tensor contribution

$$
\begin{aligned}
\mathcal{O}_{VR}^{(7)\prime} &= \frac{2G_F}{\sqrt{2}v} C_{VR}^{(7)\prime}\Big[\overline{e_R}\gamma^\mu \nu_L^C\Big]\Big[\overline{u_L} i\overset{\leftrightarrow}{\partial}_\mu d_R\Big] \\
&= \frac{2G_F}{\sqrt{2}v} C_{VR}^{(7)\prime}\Big[\overline{e_R}\gamma^\mu \nu_L^C\Big]\Big(m_d\Big[\overline{u_L}\gamma_\mu d_L\Big] + m_u\Big[\overline{u_R}\gamma_\mu d_R\Big]\Big) \\
&\quad + \frac{2G_F}{\sqrt{2}v} C_{VR}^{(7)\prime}\Big(\Big[\partial_\nu \overline{e_R}\gamma_\mu \nu_L^C\Big] + \Big[\overline{e_R}\gamma_\mu \partial_\nu \nu_L^C\Big]\Big)\Big[\overline{u_L}\sigma^{\mu\nu} d_R\Big]
\end{aligned}
\tag{B.3}
$$

With the chiral power counting $\epsilon_\chi = m_\pi/\Lambda_\chi$ and $\Lambda_\chi \sim m_N \sim 1\,\text{GeV}$ we find that the tensor operator's $\Big(\Big[\partial_\nu \overline{e_R}\gamma_\mu \nu_L^C\Big] + \Big[\overline{e_R}\gamma_\mu \partial_\nu \nu_L^C\Big]\Big)\Big[\overline{u_L}\sigma^{\mu\nu} d_R\Big]$ leading contribution to the decay amplitude is at $\mathcal{O}(\epsilon_\chi^5)$ as it is proportional to the lepton momenta $k \sim \epsilon_\chi^3 \Lambda_\chi$, the neutrino

JHEP08(2023)043

momentum $q \sim \epsilon_\chi \Lambda_\chi$ and requires the inclusion of NLO nuclear currents $\sim \epsilon_\chi$. We can therefore savely ignore it. With $m_d \sim m_u \sim \epsilon_\chi^2 \Lambda_\chi$ and the leading order contributions from $C_{VL}^{(6)} \sim \Lambda_\chi \epsilon_\chi^2$ and $C_{VR}^{(6)} \sim \Lambda_\chi \epsilon_\chi^3$, the leading order contribution from the remaining parts of $\mathcal{O}_{VR}^{(7)\prime}$ is at $\sim \frac{\Lambda_\chi^2}{v} \epsilon_\chi^4$ from $C_{VR}^{(7)\prime} m_d \left[\overline{e_R} \gamma^\mu \nu_L^C\right] \left[\overline{u_L} \gamma_\mu d_L\right]$. Therefore, compared to the other dim-7 and dim-9 operators [32] $\mathcal{O}_{VR}^{(7)\prime}$ is suppressed by an additional factor of ϵ_χ^2 and we ignore it.

C Phase space factors

C.1 Calculation of PSFs

The different phase-space factors G_{0k} that enter the half-life formula are defined as

$$\begin{aligned} G_{0k} = C_k \frac{G_F^4 m_e^2}{64\pi^5 \ln 2 R^2} \int \delta\Big(\epsilon_1 + \epsilon_2 + E_f - E_i\Big) \\ \times \Big(h_{0k}(\epsilon_1, \epsilon_2, R) \cos\theta + g_{0k}(\epsilon_1, \epsilon_2, R)\Big) \\ \times p_1 p_2 \epsilon_1 \epsilon_2 \, \mathrm{d}\epsilon_1 \, \mathrm{d}\epsilon_2 \, \mathrm{d}(\cos\theta), \end{aligned} \tag{C.1}$$

where ϵ_i, p_i denote the energy and momentum of the $i-$th. single electron, θ is the angle between the emitted electrons, we defined

$$C_k = \begin{cases} 9/2 & , k = 4 \\ m_e R/2 & , k = 6 \\ (m_e R/2)^2 & , k = 9 \\ 1 & , \text{else} \end{cases}, \tag{C.2}$$

and h_{0k}, g_{0k} were introduced in ref. [86]. The constants C_k take care of the different conventions used in refs. [86] and [32]. νDoBe takes care of this re-scaling internally, such that no action is required from the user and $C_k = 1$ can be chosen when calculating PSFs to use in νDoBe. The quark-mixing parameter V_{ud} is not part of G_{0k} in contrast to the usual literature convention. This is to account for the fact that higher dimensional typically do not exchange two W bosons. Instead, factors of V_{ud} enter the sub-amplitudes $\mathcal{A}_k$ (see ref. [32]). νDoBe includes two approximation schemes for the calculation of PSFs and electron wave functions, i.e., an approximate solution to a uniform charge distribution (Scheme A) as well as an exact solution to a point-like nuclear potential (Scheme B).

C.2 Related observables

Apart from the total decay rate, we are also interested in the differential rates, i.e., the energy and angular electron spectra [86]. The normalized single electron spectrum can be determined through

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}\epsilon_1}\left(\{C_i\}, \tilde{\epsilon}\right) \propto \sum_k g_{0k}\left(\epsilon, \Delta M - \epsilon, R\right) \left|A_k(\{C_i\})\right|^2 p_1 p_2 \epsilon \left(\Delta M - \epsilon\right) , \tag{C.3}$$

JHEP08(2023)043

Obviously, the electron spectra depend on the precise EFT model, i.e., the chosen set of Wilson coefficients $\{C_i\}$. This dependency is represented in the sub-amplitudes $\mathcal{A}_k(\{C_i\})$. We defined the mass difference of the mother and daughter isotope $\Delta M = E_i - E_f$ and the normalized electron energy $\tilde{\epsilon}_i = (\epsilon_i - m_e)/Q$ in terms of the Q-value $Q = \Delta M - 2m_e$. The nuclear radius R is determined from the mass number A

$$R = 1.2\,\mathrm{fm} \times A^{1/3}\,. \tag{C.4}$$

Similarly, the energy-dependent angular correlation coefficient a_1/a_0 is given by

$$\frac{\mathrm{d}\Gamma}{\mathrm{d}\cos\theta \mathrm{d}\tilde{\epsilon}_1} = a_0\left(1 + \frac{a_1}{a_0}\cos\theta\right)\,, \qquad \frac{a_1}{a_0}\left(\{C_i\},\tilde{\epsilon}\right) = \frac{\sum_j h_{0j}(\epsilon, \Delta M - \epsilon, R)\left|A_j(\{C_i\})\right|^2}{\sum_k g_{0k}(\epsilon, \Delta M - \epsilon, R)\left|A_k(\{C_i\})\right|^2}\,. \tag{C.5}$$

D Nuclear matrix elements

The different NMEs that enter the calculation of the sub-amplitudes $\mathcal{A}_k$ are defined via

$$h_K^{ij}(r) = \frac{2}{\pi} R_A \int_0^\infty d|\mathbf{q}| h_K^{ij}(\mathbf{q}^2) j_\lambda(|\mathbf{q}|r), \qquad h_{K,sd}^{ij}(r) = C_{sd}\frac{2}{\pi}\frac{R_A}{m_e m_p}\int_0^\infty d|\mathbf{q}|\mathbf{q}^2 h_K^{ij}(\mathbf{q}^2) j_\lambda(|\mathbf{q}|r) \tag{D.1}$$

with

$$\begin{aligned} &h_{GT,T}^{AA}(\mathbf{q}^2) = \frac{g_A^2(\mathbf{q}^2)}{g_A^2}, \qquad h_{GT}^{AP}(\mathbf{q}^2) = \frac{g_P(\mathbf{q}^2) g_A(\mathbf{q}^2)}{g_A^2}\frac{\mathbf{q}^2}{3m_N}, \quad h_{GT}^{PP}(\mathbf{q}^2) = \frac{g_P^2(\mathbf{q}^2)}{g_A^2}\frac{\mathbf{q}^4}{12m_N^2} \\ &h_{GT}^{MM}(\mathbf{q}^2) = g_M^2(\mathbf{q}^2)\frac{\mathbf{q}^2}{6g_A^2 m_N^2}, \quad h_F(\mathbf{q}^2) = g_V(\mathbf{q}^2) \end{aligned} \tag{D.2}$$

and

$$\begin{aligned} g_V(\mathbf{q}^2) &= \left(1 + \frac{\mathbf{q}^2}{\Lambda_V^2}\right)^{-2}, \quad g_A(\mathbf{q}^2) = g_A\left(1 + \frac{\mathbf{q}^2}{\Lambda_A^2}\right)^{-2}, \\ g_M(\mathbf{q}^2) &= (1+\kappa_1) g_V(\mathbf{q}^2), \quad g_P(\mathbf{q}^2) = -\frac{2m_N g_A(\mathbf{q}^2)}{\mathbf{q}^2 + m_\pi^2} \end{aligned} \tag{D.3}$$

as

$$\begin{aligned} M_{F,(sd)} &= \left\langle 0^+ \middle| \sum_{m,n} h_{F,(sd)}(r)\tau^{+(m)}\tau^{+(n)} \middle| 0^+ \right\rangle, \\ M_{GT,(sd)}^{ij} &= \left\langle 0^+ \middle| \sum_{m,n} h_{GT,(sd)}^{ij}(r)\sigma^{(m)}\cdot\sigma^{(n)}\tau^{+(m)}\tau^{+(n)} \middle| 0^+ \right\rangle, \\ M_{T,(sd)}^{ij} &= \left\langle 0^+ \middle| \sum_{m,n} h_{T,(sd)}^{ij}(r) S^{mn}(\hat{\mathbf{r}})\tau^{+(m)}\tau^{+(n)} \middle| 0^+ \right\rangle, \end{aligned} \tag{D.4}$$

with the position-space tensor

$$S^{mn}(\hat{\mathbf{r}}) = 3(\sigma^{(m)}\cdot\hat{\mathbf{r}})(\sigma^{(n)}\cdot\hat{\mathbf{r}}) - \sigma^{(m)}\cdot\sigma^{(n)} \tag{D.5}$$

Again, the re-scaling factor

$$C_{sd} = \frac{m_e m_p}{m_\pi^2} \tag{D.6}$$

JHEP08(2023)043

takes care of different conventions used in the common literature and [32]. νDoBe takes care of this re-scaling internally when importing NMEs from the corresponding `.csv` files. That is, the NMEs defined in the `.csv` files are multiplied by C_{sd} when initiating a model class such that, similar to the case of C_k in the PSFs, users can choose $C_{sd} = 1$ when calculating NMEs to use in νDoBe.

Open Access. This article is distributed under the terms of the Creative Commons Attribution License (CC-BY 4.0), which permits any use, distribution and reproduction in any medium, provided the original author(s) and source are credited.

References

[1] S. Umehara et al., *Neutrino-less double-beta decay of Ca-48 studied by Ca F(2)(Eu) scintillators*, *Phys. Rev. C* **78** (2008) 058501 [arXiv:0810.4746] [INSPIRE].

[2] GERDA collaboration, *Final Results of GERDA on the Search for Neutrinoless Double-β Decay*, *Phys. Rev. Lett.* **125** (2020) 252502 [arXiv:2009.06079] [INSPIRE].

[3] CUPID-0 collaboration, *First Result on the Neutrinoless Double-β Decay of ^{82}Se with CUPID-0*, *Phys. Rev. Lett.* **120** (2018) 232502 [arXiv:1802.07791] [INSPIRE].

[4] NEMO-3 collaboration, *Measurement of the two neutrino double beta decay half-life of Zr-96 with the NEMO-3 detector*, *Nucl. Phys. A* **847** (2010) 168 [arXiv:0906.2694] [INSPIRE].

[5] CUPID collaboration, *New Limit for Neutrinoless Double-Beta Decay of ^{100}Mo from the CUPID-Mo Experiment*, *Phys. Rev. Lett.* **126** (2021) 181802 [arXiv:2011.13243] [INSPIRE].

[6] F.A. Danevich et al., *Search for double beta decay of ^{116}Cd with enriched $^{116}CdWO_4$ crystal scintillators (Aurora experiment)*, *J. Phys. Conf. Ser.* **718** (2016) 062009 [arXiv:1601.05578] [INSPIRE].

[7] C. Arnaboldi et al., *A Calorimetric search on double beta decay of Te-130*, *Phys. Lett. B* **557** (2003) 167 [hep-ex/0211071] [INSPIRE].

[8] CUORE collaboration, *Improved Limit on Neutrinoless Double-Beta Decay in ^{130}Te with CUORE*, *Phys. Rev. Lett.* **124** (2020) 122501 [arXiv:1912.10966] [INSPIRE].

[9] EXO-200 collaboration, *Searches for double beta decay of ^{134}Xe with EXO-200*, *Phys. Rev. D* **96** (2017) 092001 [arXiv:1704.05042] [INSPIRE].

[10] KamLAND-Zen collaboration, *Search for the Majorana Nature of Neutrinos in the Inverted Mass Ordering Region with KamLAND-Zen*, *Phys. Rev. Lett.* **130** (2023) 051801 [arXiv:2203.02139] [INSPIRE].

[11] NEMO collaboration, *Measurement of the Double Beta Decay Half-life of Nd-150 and Search for Neutrinoless Decay Modes with the NEMO-3 Detector*, *Phys. Rev. C* **80** (2009) 032501 [arXiv:0810.0248] [INSPIRE].

[12] E. Armengaud et al., *The CUPID-Mo experiment for neutrinoless double-beta decay: performance and prospects*, *Eur. Phys. J. C* **80** (2020) 44 [arXiv:1909.02994] [INSPIRE].

[13] LEGEND collaboration, *The Large Enriched Germanium Experiment for Neutrinoless Double Beta Decay (LEGEND)*, *AIP Conf. Proc.* **1894** (2017) 020027 [arXiv:1709.01980] [INSPIRE].

JHEP08(2023)043

[14] LEGEND collaboration, *The Large Enriched Germanium Experiment for Neutrinoless $\beta\beta$ Decay: LEGEND-1000 Preconceptual Design Report*, arXiv:2107.11462 [INSPIRE].

[15] nEXO collaboration, *nEXO: neutrinoless double beta decay search beyond 10^{28} year half-life sensitivity*, *J. Phys. G* **49** (2022) 015104 [arXiv:2106.16243] [INSPIRE].

[16] SNO+ collaboration, *The SNO+ experiment*, 2021 *JINST* **16** P08059 [arXiv:2104.11687] [INSPIRE].

[17] J. Schechter and J.W.F. Valle, *Neutrinoless Double-β Decay in $SU(2) \times U(1)$ Theories*, *Phys. Rev. D* **25** (1982) 2951 [INSPIRE].

[18] M.J. Dolinski, A.W.P. Poon and W. Rodejohann, *Neutrinoless Double-Beta Decay: Status and Prospects*, *Ann. Rev. Nucl. Part. Sci.* **69** (2019) 219 [arXiv:1902.04097] [INSPIRE].

[19] M. Agostini et al., *Toward the discovery of matter creation with neutrinoless $\beta\beta$ decay*, *Rev. Mod. Phys.* **95** (2023) 025002 [arXiv:2202.01787] [INSPIRE].

[20] V. Cirigliano et al., *Neutrinoless Double-Beta Decay: A Roadmap for Matching Theory to Experiment*, arXiv:2203.12169 [INSPIRE].

[21] F.F. Deppisch, M. Hirsch and H. Pas, *Neutrinoless Double Beta Decay and Physics Beyond the Standard Model*, *J. Phys. G* **39** (2012) 124007 [arXiv:1208.0727] [INSPIRE].

[22] G. Li, M. Ramsey-Musolf and J.C. Vasquez, *Left-Right Symmetry and Leading Contributions to Neutrinoless Double Beta Decay*, *Phys. Rev. Lett.* **126** (2021) 151801 [arXiv:2009.01257] [INSPIRE].

[23] F. Deppisch and H. Pas, *Pinning down the mechanism of neutrinoless double beta decay with measurements in different nuclei*, *Phys. Rev. Lett.* **98** (2007) 232501 [hep-ph/0612165] [INSPIRE].

[24] V.M. Gehman and S.R. Elliott, *Multiple-Isotope Comparison for Determining 0 nu beta beta Mechanisms*, *J. Phys. G* **34** (2007) 667 [*Erratum ibid.* **35** (2008) 029701] [hep-ph/0701099] [INSPIRE].

[25] L. Gráf, M. Lindner and O. Scholer, *Unraveling the $0\nu\beta\beta$ decay mechanisms*, *Phys. Rev. D* **106** (2022) 035022 [arXiv:2204.10845] [INSPIRE].

[26] M. Agostini, F.F. Deppisch and G. Van Goffrier, *Probing the mechanism of neutrinoless double-beta decay in multiple isotopes*, *JHEP* **02** (2023) 172 [arXiv:2212.00045] [INSPIRE].

[27] H. Pas, M. Hirsch, H.V. Klapdor-Kleingrothaus and S.G. Kovalenko, *Towards a superformula for neutrinoless double beta decay*, *Phys. Lett. B* **453** (1999) 194 [INSPIRE].

[28] H. Pas, M. Hirsch, H.V. Klapdor-Kleingrothaus and S.G. Kovalenko, *A Superformula for neutrinoless double beta decay. 2. The Short range part*, *Phys. Lett. B* **498** (2001) 35 [hep-ph/0008182] [INSPIRE].

[29] G. Prezeau, M. Ramsey-Musolf and P. Vogel, *Neutrinoless double beta decay and effective field theory*, *Phys. Rev. D* **68** (2003) 034016 [hep-ph/0303205] [INSPIRE].

[30] L. Graf, F.F. Deppisch, F. Iachello and J. Kotila, *Short-Range Neutrinoless Double Beta Decay Mechanisms*, *Phys. Rev. D* **98** (2018) 095023 [arXiv:1806.06058] [INSPIRE].

[31] V. Cirigliano et al., *Neutrinoless double beta decay in chiral effective field theory: lepton number violation at dimension seven*, *JHEP* **12** (2017) 082 [arXiv:1708.09390] [INSPIRE].

[32] V. Cirigliano et al., *A neutrinoless double beta decay master formula from effective field theory*, *JHEP* **12** (2018) 097 [arXiv:1806.02780] [INSPIRE].

JHEP08(2023)043

[33] W. Dekens et al., *Sterile neutrinos and neutrinoless double beta decay in effective field theory*, *JHEP* **06** (2020) 097 [arXiv:2002.07182] [INSPIRE].

[34] A. Kobach, *Baryon Number, Lepton Number, and Operator Dimension in the Standard Model*, *Phys. Lett. B* **758** (2016) 455 [arXiv:1604.05726] [INSPIRE].

[35] S. Weinberg, *Baryon and Lepton Nonconserving Processes*, *Phys. Rev. Lett.* **43** (1979) 1566 [INSPIRE].

[36] L. Lehman, *Extending the Standard Model Effective Field Theory with the Complete Set of Dimension-7 Operators*, *Phys. Rev. D* **90** (2014) 125023 [arXiv:1410.4193] [INSPIRE].

[37] Y. Liao and X.-D. Ma, *An explicit construction of the dimension-9 operator basis in the standard model effective field theory*, *JHEP* **11** (2020) 152 [arXiv:2007.08125] [INSPIRE].

[38] V. Cirigliano et al., *New Leading Contribution to Neutrinoless Double-β Decay*, *Phys. Rev. Lett.* **120** (2018) 202001 [arXiv:1802.10097] [INSPIRE].

[39] V. Cirigliano et al., *Renormalized approach to neutrinoless double- β decay*, *Phys. Rev. C* **100** (2019) 055504 [arXiv:1907.11254] [INSPIRE].

[40] B. Grzadkowski, M. Iskrzynski, M. Misiak and J. Rosiek, *Dimension-Six Terms in the Standard Model Lagrangian*, *JHEP* **10** (2010) 085 [arXiv:1008.4884] [INSPIRE].

[41] B. Henning, X. Lu, T. Melia and H. Murayama, *2, 84, 30, 993, 560, 15456, 11962, 261485,...: Higher dimension operators in the SM EFT*, *JHEP* **08** (2017) 016 [*Erratum ibid.* **09** (2019) 019] [arXiv:1512.03433] [INSPIRE].

[42] Y. Liao and X.-D. Ma, *Renormalization Group Evolution of Dimension-seven Operators in Standard Model Effective Field Theory and Relevant Phenomenology*, *JHEP* **03** (2019) 179 [arXiv:1901.10302] [INSPIRE].

[43] H.-L. Li et al., *Complete set of dimension-eight operators in the standard model effective field theory*, *Phys. Rev. D* **104** (2021) 015026 [arXiv:2005.00008] [INSPIRE].

[44] C.W. Murphy, *Dimension-8 operators in the Standard Model Eective Field Theory*, *JHEP* **10** (2020) 174 [arXiv:2005.00059] [INSPIRE].

[45] H.-L. Li et al., *Complete set of dimension-nine operators in the standard model effective field theory*, *Phys. Rev. D* **104** (2021) 015025 [arXiv:2007.07899] [INSPIRE].

[46] E.E. Jenkins, A.V. Manohar and P. Stoffer, *Low-Energy Effective Field Theory below the Electroweak Scale: Operators and Matching*, *JHEP* **03** (2018) 016 [arXiv:1709.04486] [INSPIRE].

[47] E.E. Jenkins, A.V. Manohar and P. Stoffer, *Low-Energy Effective Field Theory below the Electroweak Scale: Anomalous Dimensions*, *JHEP* **01** (2018) 084 [arXiv:1711.05270] [INSPIRE].

[48] W. Dekens and P. Stoffer, *Low-energy effective field theory below the electroweak scale: matching at one loop*, *JHEP* **10** (2019) 197 [*Erratum ibid.* **11** (2022) 148] [arXiv:1908.05295] [INSPIRE].

[49] Y. Liao, X.-D. Ma and Q.-Y. Wang, *Extending low energy effective field theory with a complete set of dimension-7 operators*, *JHEP* **08** (2020) 162 [arXiv:2005.08013] [INSPIRE].

[50] H.-L. Li et al., *Low energy effective field theory operator basis at $d \leq 9$*, *JHEP* **06** (2021) 138 [arXiv:2012.09188] [INSPIRE].

JHEP08(2023)043

[51] J. Hyvärinen and J. Suhonen, *Nuclear matrix elements for $0\nu\beta\beta$ decays with light or heavy Majorana-neutrino exchange*, *Phys. Rev. C* **91** (2015) 024613 [INSPIRE].

[52] J. Terasaki, *Strength of the isoscalar pairing interaction determined by a relation between double-charge change and double-pair transfer for double- β decay*, *Phys. Rev. C* **102** (2020) 044303 [arXiv:2003.03542] [INSPIRE].

[53] F. Šimkovic, V. Rodin, A. Faessler and P. Vogel, *$0\nu\beta\beta$ and $2\nu\beta\beta$ nuclear matrix elements, quasiparticle random-phase approximation, and isospin symmetry restoration*, *Phys. Rev. C* **87** (2013) 045501 [arXiv:1302.1509] [INSPIRE].

[54] M.T. Mustonen and J. Engel, *Large-scale calculations of the double-β decay of ^{76}Ge,^{130}Te,^{136}Xe, and ^{150}Nd in the deformed self-consistent Skyrme quasiparticle random-phase approximation*, *Phys. Rev. C* **87** (2013) 064302 [arXiv:1301.6997] [INSPIRE].

[55] D.-L. Fang, A. Faessler and F. Simkovic, *$0\nu\beta\beta$ -decay nuclear matrix element for light and heavy neutrino mass mechanisms from deformed quasiparticle random-phase approximation calculations for ^{76}Ge, ^{82}Se, ^{130}Te, ^{136}Xe , and ^{150}Nd with isospin restoration*, *Phys. Rev. C* **97** (2018) 045503 [arXiv:1803.09195] [INSPIRE].

[56] N. López Vaquero, T.R. Rodríguez and J.L. Egido, *Shape and pairing fluctuations effects on neutrinoless double beta decay nuclear matrix elements*, *Phys. Rev. Lett.* **111** (2013) 142501 [arXiv:1401.0650] [INSPIRE].

[57] J.M. Yao et al., *Systematic study of nuclear matrix elements in neutrinoless double-β decay with a beyond-mean-field covariant density functional theory*, *Phys. Rev. C* **91** (2015) 024316 [arXiv:1410.6326] [INSPIRE].

[58] T.R. Rodriguez and G. Martinez-Pinedo, *Energy density functional study of nuclear matrix elements for neutrinoless $\beta\beta$ decay*, *Phys. Rev. Lett.* **105** (2010) 252503 [arXiv:1008.5260] [INSPIRE].

[59] F.F. Deppisch, L. Graf, F. Iachello and J. Kotila, *Analysis of light neutrino exchange and short-range mechanisms in $0\nu\beta\beta$ decay*, *Phys. Rev. D* **102** (2020) 095016 [arXiv:2009.10119] [INSPIRE].

[60] J. Barea, J. Kotila and F. Iachello, *$0\nu\beta\beta$ and $2\nu\beta\beta$ nuclear matrix elements in the interacting boson model with isospin restoration*, *Phys. Rev. C* **91** (2015) 034304 [arXiv:1506.08530] [INSPIRE].

[61] L. Coraggio et al., *Calculation of the neutrinoless double-β decay matrix element within the realistic shell model*, *Phys. Rev. C* **101** (2020) 044315 [arXiv:2001.00890] [INSPIRE].

[62] A. Neacsu and M. Horoi, *Shell model studies of the ^{130}Te neutrinoless double-beta decay*, *Phys. Rev. C* **91** (2015) 024309 [arXiv:1411.4313] [INSPIRE].

[63] J. Menendez, A. Poves, E. Caurier and F. Nowacki, *Disassembling the Nuclear Matrix Elements of the Neutrinoless beta beta Decay*, *Nucl. Phys. A* **818** (2009) 139 [arXiv:0801.3760] [INSPIRE].

[64] J. Menéndez, *Neutrinoless $\beta\beta$ decay mediated by the exchange of light and heavy neutrinos: The role of nuclear structure correlations*, *J. Phys. G* **45** (2018) 014003 [arXiv:1804.02105] [INSPIRE].

[65] Particle Data Group collaboration, *Review of Particle Physics*, *PTEP* **2022** (2022) 083C01 [INSPIRE].

JHEP08(2023)043

[66] V. Cirigliano et al., *Toward Complete Leading-Order Predictions for Neutrinoless Double β Decay*, *Phys. Rev. Lett.* **126** (2021) 172002 [arXiv:2012.11602] [INSPIRE].

[67] R. Wirth, J.M. Yao and H. Hergert, *Ab Initio Calculation of the Contact Operator Contribution in the Standard Mechanism for Neutrinoless Double Beta Decay*, *Phys. Rev. Lett.* **127** (2021) 242502 [arXiv:2105.05415] [INSPIRE].

[68] J.C. Pati and A. Salam, *Lepton Number as the Fourth Color*, *Phys. Rev. D* **10** (1974) 275 [INSPIRE].

[69] R.N. Mohapatra and J.C. Pati, *A Natural Left-Right Symmetry*, *Phys. Rev. D* **11** (1975) 2558 [INSPIRE].

[70] G. Senjanovic and R.N. Mohapatra, *Exact Left-Right Symmetry and Spontaneous Violation of Parity*, *Phys. Rev. D* **12** (1975) 1502 [INSPIRE].

[71] P. Duka, J. Gluza and M. Zralek, *Quantization and renormalization of the manifest left-right symmetric model of electroweak interactions*, *Annals Phys.* **280** (2000) 336 [hep-ph/9910279] [INSPIRE].

[72] M. Hirsch, H.V. Klapdor-Kleingrothaus and S.G. Kovalenko, *New leptoquark mechanism of neutrinoless double beta decay*, *Phys. Rev. D* **54** (1996) R4207 [hep-ph/9603213] [INSPIRE].

[73] M. Blennow, E. Fernandez-Martinez, J. Lopez-Pavon and J. Menendez, *Neutrinoless double beta decay in seesaw models*, *JHEP* **07** (2010) 096 [arXiv:1005.3240] [INSPIRE].

[74] J. Barea, J. Kotila and F. Iachello, *Limits on sterile neutrino contributions to neutrinoless double beta decay*, *Phys. Rev. D* **92** (2015) 093001 [arXiv:1509.01925] [INSPIRE].

[75] T. Asaka, S. Eijima and H. Ishida, *On neutrinoless double beta decay in the νMSM*, *Phys. Lett. B* **762** (2016) 371 [arXiv:1606.06686] [INSPIRE].

[76] J. de Vries, G. Li, M.J. Ramsey-Musolf and J.C. Vasquez, *Light sterile neutrinos, left-right symmetry, and $0\nu\beta\beta$ decay*, *JHEP* **11** (2022) 056 [arXiv:2209.03031] [INSPIRE].

[77] W. Dekens et al., *Neutrinoless double-beta decay in the neutrino-extended Standard Model*, arXiv:2303.04168 [INSPIRE].

[78] J. Kotila and F. Iachello, *Phase space factors for double-β decay*, *Phys. Rev. C* **85** (2012) 034316 [arXiv:1209.5722] [INSPIRE].

[79] S. Stoica and M. Mirea, *New calculations for phase space factors involved in double-β decay*, *Phys. Rev. C* **88** (2013) 037303 [arXiv:1307.0290] [INSPIRE].

[80] C.R. Harris et al., *Array programming with NumPy*, *Nature* **585** (2020) 357 [arXiv:2006.10256] [INSPIRE].

[81] J. Reback et al., *pandas-dev/pandas: Pandas 1.1.3*, DOI:10.5281/ZENODO.4067057.

[82] W. McKinney, *Data Structures for Statistical Computing in Python*, in the proceedings of the *9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman eds., (2010), pp. 56–61 [DOI:10.25080/majora-92bf1922-00a] [INSPIRE].

[83] J.D. Hunter, *Matplotlib: A 2D Graphics Environment*, *Comput. Sci. Eng.* **9** (2007) 90 [INSPIRE].

[84] P. Virtanen et al., *SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python*, *Nature Meth.* **17** (2020) 261 [arXiv:1907.10121] [INSPIRE].

JHEP08(2023)043

[85] F. Johansson et al., *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 1.1.0)* (2013), http://mpmath.org/.

[86] D. Stefanik, R. Dvornicky, F. Simkovic and P. Vogel, *Reexamining the light neutrino exchange mechanism of the $0\nu\beta\beta$ decay with left- and right-handed leptonic and hadronic currents*, *Phys. Rev. C* **92** (2015) 055502 [arXiv:1506.07145] [INSPIRE].

[87] PARTICLE DATA GROUP collaboration, *Review of Particle Physics*, *Chin. Phys. C* **40** (2016) 100001 [INSPIRE].

[88] A. Nicholson et al., *Heavy physics contributions to neutrinoless double beta decay from QCD*, *Phys. Rev. Lett.* **121** (2018) 172501 [arXiv:1805.02634] [INSPIRE].

[89] T. Bhattacharya et al., *Axial, Scalar and Tensor Charges of the Nucleon from 2+1+1-flavor Lattice QCD*, *Phys. Rev. D* **94** (2016) 054508 [arXiv:1606.07049] [INSPIRE].

[90] V. Cirigliano et al., *Determining the leading-order contact term in neutrinoless double β decay*, *JHEP* **05** (2021) 289 [arXiv:2102.03371] [INSPIRE].

[91] R. Arnold et al., *Probing new physics models of neutrinoless double beta decay with SuperNEMO*, *Eur. Phys. J. C* **70** (2010) 927.

[92] F.F. Deppisch, L. Graf and F. Šimkovic, *Searching for New Physics in Two-Neutrino Double Beta Decay*, *Phys. Rev. Lett.* **125** (2020) 171801 [arXiv:2003.11836] [INSPIRE].

JHEP08(2023)043