FERMILAB-PUB-22-594-T

DarkNews: a Python-based event generator for heavy neutral lepton production in neutrino-nucleus scattering

Asli M. Abdullahi,[1, *] Jaime Hoefken Zink,[2, 3, †] Matheus Hostert,[4, 5, 6, ‡] Daniele Massaro,[2, 3, 7, §] and Silvia Pascoli[2, 3, 8, ¶]

[1] *Theoretical Physics Department, Fermi National Accelerator Laboratory, Batavia, IL 60510, USA*
[2] *Dipartimento di Fisica e Astronomia, Università di Bologna, via Irnerio 46, 40126 Bologna, Italy*
[3] *INFN, Sezione di Bologna, viale Berti Pichat 6/2, 40127 Bologna, Italy*
[4] *Perimeter Institute for Theoretical Physics, Waterloo, ON N2J 2W9, Canada*
[5] *School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA*
[6] *William I. Fine Theoretical Physics Institute, School of Physics and Astronomy, University of Minnesota, Minneapolis, MN 55455, USA*
[7] *Centre for Cosmology, Particle Physics and Phenomenology (CP3), Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium*
[8] *CERN, Theoretical Physics Department, Geneva, Switzerland*

(Dated: July 12, 2022)

We introduce DarkNews, a lightweight Python-based Monte-Carlo generator for beyond-the-Standard-Model neutrino-nucleus scattering. The generator handles the production and decay of heavy neutral leptons via additional vector or scalar mediators, as well as through transition magnetic moments. DarkNews samples pre-computed neutrino-nucleus upscattering cross sections and heavy neutrino decay rates to produce dilepton and single-photon events in accelerator neutrino experiments. We present two case studies with differential distributions for models that can explain the MiniBooNE excess. The aim of this code is to aid the neutrino theory and experimental communities in performing searches and sensitivity studies for new particles produced in neutrino upscattering.

I. INTRODUCTION

The discovery of physics beyond the Standard Model (BSM) is among the highest priority goals of modern experimental particle physics. While searches for heavy new physics continue at the Large Hadron Collider, other low-energy experiments continue to make progress in searching for lighter, but weakly-coupled particles [1, 2]. Among these, neutrino experiments stand out for their high intensity and detector capabilities. The sheer number of protons delivered on target produces an intense neutrino beam, and possibly large numbers of hypothetical particles. Together with the fact that neutrinos interact very little with matter, accelerator neutrino experiments are an ideal environment to look for the feeble interactions of hidden sector particles.

In this context, light dark sectors are particularly relevant as they may contain particles that propagate freely between the proton target and the neutrino detector before depositing some amount of energy either through decays or interactions with matter. Numerous studies in the literature have explored these features to set limits on light dark matter particles, light scalar and vector bosons, and heavy neutral leptons [3–7] (for recent overviews, see Refs. [8–10]). Several neutrino detectors have come online in the past decade, and several more will become operational in the near future. These include the near and far detectors of the short-baseline program at Fermilab [11], the upgraded near detector of T2K [12], and, on a more distant time scale, Hyper-Kamiokande [13] and DUNE [14]. Several new proposals are also currently under study [15]. Exploring their capabilities to probe light new physics will require efforts from both the experimental and theoretical particle physics communities. On the theory side, the provision of theoretical targets and tools to simulate the effects of light new physics at experiments will be crucial.

New interactions shared between neutrinos and dark sector particles are usually also shared by other charged particles in the SM. One notable exception is given by the neutrino portal

$$\mathcal{L} \supset \overline{L}\widetilde{H}N + \mathcal{L}_{N-\text{int}}, \tag{1}$$

where new interactions felt by a heavy neutrino N can be shared with the Standard Model neutrinos ν via mixing. In the absence of new forces, N interacts with SM particles via the Weak interactions, further suppressed by mixing elements $U_{\alpha N}$ which are experimentally constrained to be very small. However, even very feeble interactions between N and other dark sector states can dominate over the "weaker-than-Weak" couplings between N and the SM W and Z bosons. In this case, it is possible that BSM interactions of N can dominate its decay modes and lead to new production mechanisms.

One new production mechanism that has gained interest recently is the production of heavy neutrinos by neutrinos scattering off of ordinary matter. Their subsequent

[*] asli@fnal.gov; (0000-0002-6122-4986)
[†] jaime.hoefkenzink2@unibo.it; (0000-0002-4086-2030)
[‡] mhostert@perimeterinstitute.ca; (0000-0002-9584-8877)
[§] daniele.massaro5@unibo.it; (0000-0002-1013-3953)
[¶] silvia.pascoli@unibo.it; (0000-0002-2958-456X)

decays to charged particles via the new force have been used in a number of phenomenological papers. This new force may be mediated by, for example, a dark photon or dark scalar, or even by the SM photon itself if transition magnetic moments between light and heavy neutrinos exist. These models have a relatively simple phenomenology in neutrino experiments: neutrino scattering with a nuclear target and potentially displaced decay signatures with photons, charged lepton pairs, or meson final states.

```
pip install DarkNews
```

FIG. 1. The `DarkNews` logo and the installation command in systems where pip is linked to the desired version of `Python`.

Currently, to model new physics in neutrino-nucleus scattering the community often relies on custom-made generators, or the manual addition of new interactions and particles to existing neutrino generators e.g. `Genie` [16, 17], `NuWro` [18], `Neut` [19], and `GiBUU` [20]. This is to be contrasted with collider physics, where decades-long efforts have produced complete and automated Monte-Carlo tools that go from a user-defined Lagrangian input, e.g. FeynRules [21, 22], to matrix elements in the standard UFO file format [23] with a universal interface with event generators (see Ref. [24] for a recent overview of Monte-Carlo generators). Similarly, in the intensity frontier, exploration of dark matter models has followed a similar route with programs like `MicrOMEGAs` [25–28], `MadDM` [29–32], as well as the more recent `MadDump` [33] for dark sectors in beam dump-style experiments.

In neutrino physics, these efforts are rather recent. Tools like `LeptonInjector` [34] specialize in providing a detailed description of neutrino interactions in arbitrary and complex detector geometries at neutrino telescopes as well as at accelerator neutrino experiments. Implementing new physics models in `LeptonInjector` is possible (as done recently in Ref [35]), but also relies on a case-by-case approach. On the other hand, the recently-developed tool `Achilles` [36, 37] aims to combine a state-of-the-art description of nuclear physics in lepton-nucleus interactions with automated particle physics tools. These tools have recently gained great interest from both the theoretical and experimental community for their interface with tools like FeynRules. In view of upcoming large-scale projects like DUNE and Hyper-Kamiokande, they will be of great relevance for the phenomenology community studying new physics models at GeV neutrino experiments. Nevertheless, there is a need for simpler tools that can provide ready-to-use simulations of specific new physics models. This is especially true when it comes to probing new-physics solutions to the short-baseline puzzle. Such generators are also of great importance for phenomenologists studying the sensitivity of upcoming experiments to models of new physics and for whom the use of more sophisticated generators is time-consuming. In this article, we present a simple tool that aims to do just this for a range of heavy neutral lepton (HNL) models.

This article is divided as follows: in Section II, we introduce the generator and describe its pipeline, inputs, and outputs. In Section III, we discuss the new physics models already implemented in the code, providing a few example simulations with DarkNews in Section IV. Finally, we conclude with Section V. Tables of input and output formats are given in Appendix A.

II. THE GENERATOR

`DarkNews` provides a simple, fast, and `Python`-based generator that simulates neutrino-induced production and decay of heavy neutrinos inside neutrino detectors. It builds on `vegas` [38, 39] to sample from pre-computed, analytical differential cross sections and decay rates, using standard algorithms for adaptive and stratified sampling. The main scope of the generator is in the modeling of coherent neutrino-nucleus scattering followed by the decay of HNLs endowed with additional interactions. We also provide cross sections for incoherent scattering on nucleons, but we do not model any nuclear physics. The models that are currently implemented are discussed in Section III. Implementation of other new physics scenarios in DarkNews-generator can be pursued on a case-by-case basis.

The code takes as input: neutrino fluxes, model parameters, and nuclear targets, and provides as an output a set of weighed events with all particles involved, their 4-momenta, event rate weights, weights for flux-averaged cross sections, weights for decay rates, and miscellaneous event information. `DarkNews` does not model detector geometries (except for its default implementation of MiniBooNE and MicroBooNE). For all other experiments, scattering on the list of desired targets is placed at $(t, x, y, z) = (0, 0, 0, 0)$, and the decay position is calculated based on the propagating heavy neutrino. All units in `DarkNews` are in GeV for energies and momenta, cm for positions, and s for time.

The code is publicly available on GitHub [1], and can be used in a variety of accelerator neutrino experiments. `DarkNews` was developed using `Python` 3.9 and makes use of a few `Cython` functions for improved performance.

[1] github.com/mhostert/DarkNews-generator.

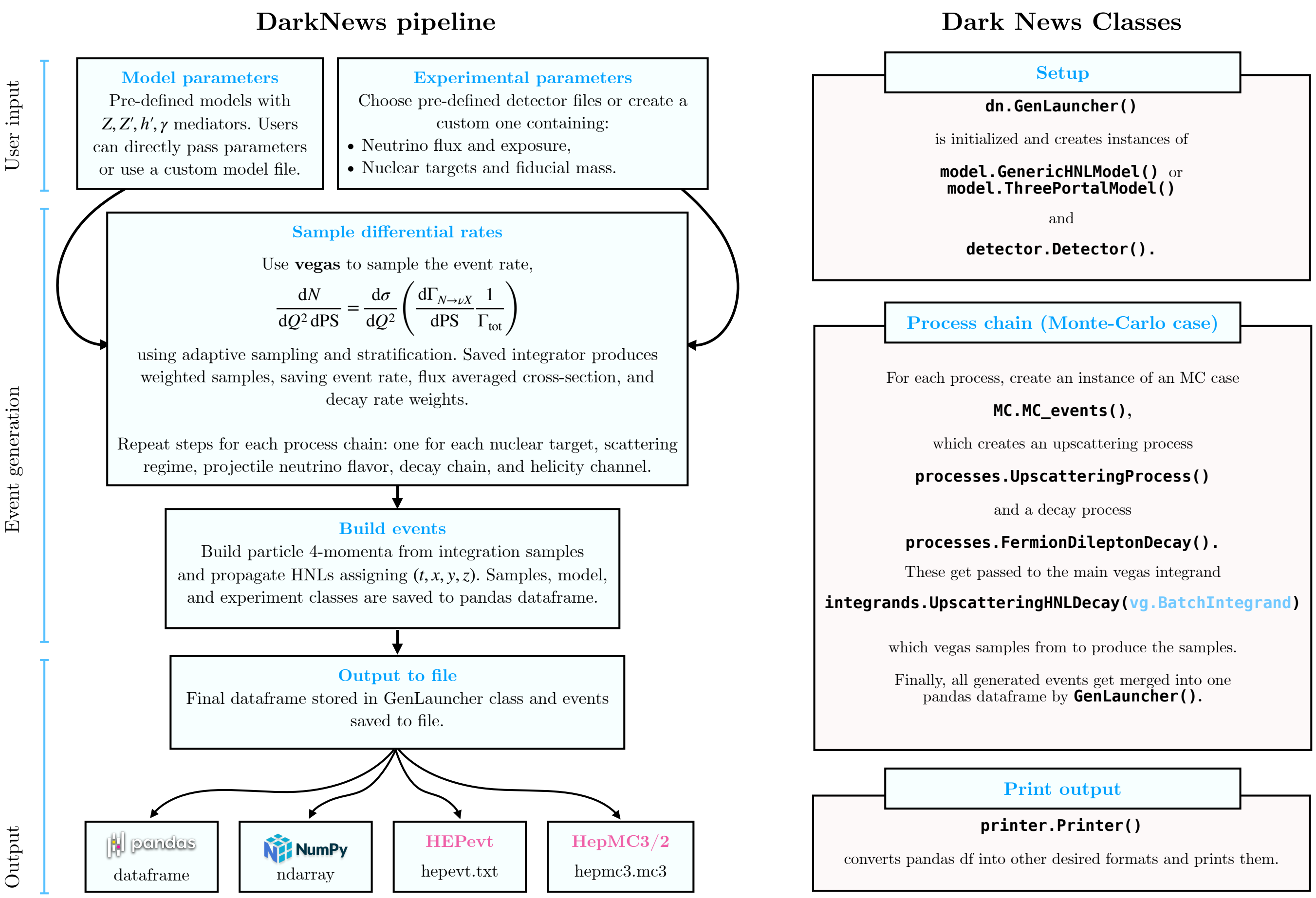
DarkNews pipeline
User input
Model parameters
Pre-defined models with Z, Z', h', γ mediators. Users can directly pass parameters or use a custom model file.
Experimental parameters
Choose pre-defined detector files or create a custom one containing:
• Neutrino flux and exposure,
• Nuclear targets and fiducial mass.
Event generation
Sample differential rates
Use **vegas** to sample the event rate,
$$\frac{\mathrm{d}N}{\mathrm{d}Q^2\,\mathrm{dPS}} = \frac{\mathrm{d}\sigma}{\mathrm{d}Q^2}\left(\frac{\mathrm{d}\Gamma_{N\to\nu X}}{\mathrm{dPS}}\frac{1}{\Gamma_{\rm tot}}\right)$$
using adaptive sampling and stratification. Saved integrator produces weighted samples, saving event rate, flux averaged cross-section, and decay rate weights.
Repeat steps for each process chain: one for each nuclear target, scattering regime, projectile neutrino flavor, decay chain, and helicity channel.
Build events
Build particle 4-momenta from integration samples and propagate HNLs assigning (t, x, y, z). Samples, model, and experiment classes are saved to pandas dataframe.
Output
Output to file
Final dataframe stored in GenLauncher class and events saved to file.
pandas
dataframe
NumPy
ndarray
HEPevt
hepevt.txt
HepMC3/2
hepmc3.mc3
Dark News Classes
Setup
dn.GenLauncher()
is initialized and creates instances of
model.GenericHNLModel() or model.ThreePortalModel()
and
detector.Detector().
Process chain (Monte-Carlo case)
For each process, create an instance of an MC case
MC.MC_events(),
which creates an upscattering process
processes.UpscatteringProcess()
and a decay process
processes.FermionDileptonDecay().
These get passed to the main vegas integrand
integrands.UpscatteringHNLDecay(vg.BatchIntegrand)
which vegas samples from to produce the samples.
Finally, all generated events get merged into one pandas dataframe by GenLauncher().
Print output
printer.Printer()
converts pandas df into other desired formats and prints them.

FIG. 2. A schematic chart of the DarkNews event generation. On the left, we provide a big-picture explanation of the generator in terms of the underlying physics, and on the right, in parallel, we show the DarkNews classes responsible for each action.

As indicated in Figure 1, the easiest way to install DarkNews is via **pip**, which will also install all requisite requirements automatically. Requirements include **numpy**, **pandas**, **vegas**, and other smaller packages, including particle and pyhepmc_ng, which are part scikit-hep [40]. With the same Python environment used in the installation, DarkNews can be easily imported in your Python script as `import DarkNews as dn`, defining our preferred shorthand **dn**.

Getting started with event generation is very simple. We provide both a Python interface as well as a command-line functionality. From a Python script, one can simply run

```python
import DarkNews as dn
my_gen = dn.GenLauncher(mzprime=1.25, m4=0.140,
↪   neval=1000, noHF=True, HNLtype="dirac",
↪   exp="microboone",
↪   nu_flavors=["nu_mu","nu_mu_bar"])
df = my_gen.run()
```

to generate approximately 1000 upscattering events per process chain (in this case 1000 for ν_μ and 1000 for $\overline{\nu}_\mu$ initiated upscattering) at MicroBooNE for a dark photon mass of 1.25 GeV and a Dirac heavy neutrino of mass 140 MeV, excluding the sub-dominant, helicity-flipping (HF) chain. The last line collects the main pandas dataframe with all the events and metadata, which is now also an attribute of the `GenLauncher` instance, `my_gen.df`. The code above will also save the dataframe to a file at a default location set by the model parameter names. Alternatively, it is possible to achieve the same result by using the command line,

```
dn_gen --mzprime 1.25 --m4 0.140 --neval 1000 --noHF
↪   --HNLtype dirac --exp microboone --nu_flavors
↪   nu_mu nu_mu_bar
```

For single-valued arguments, an = sign may be used, but for multi-valued arguments like in --nu_flavors above, it cannot.

The two previous examples will use the default choices for couplings, but these can also be specified by the user, as we explain below. A quick view into some of the kinematics and properties of the generation can be achieved with the `--make-summary-plots` option, which will produce a few histograms of energies and angles of the particles involved in the same path as the outputs are saved.

A. Inputs

As shown above, it is possible to pass all model parameters as arguments to the `GenLauncher` class or to the `dn_gen` command. A list of available arguments is provided in Table III. For the most up-to-date information, please see the `README` file. The same list can also be accessed via `dn_gen --help` in the terminal prompt. Similarly, experimental definitions, which encompass a choice for neutrino flux, nuclear targets, and exposure, can also be set as arguments. In this case, the user can choose from a set of pre-defined experiments listed in Table I, together with our source for the neutrino fluxes used. The given detector fiducial mass is an estimate that will vary from analysis to analysis. Note that detector geometries are not specified within `DarkNews`, and should be built by the user after event generation on the different nuclear targets is complete. This means that the position of scattering is not implemented at the `DarkNews` generation level (except for the MiniBooNE and MicroBooNE experiments) and that the decay position of the heavy neutrino is calculated from the origin at time $t = 0$.

The user may define their own models and experimental configurations with input files. We provide two types of file interfaces for that, one for setting experimental conditions and one for setting model parameters.

a. Model definition files Input parameters can be specified via text file, which could be passed as a parameter to the `GenLauncher` class. The following code would parse the parameters contained in the file `parameters.txt` (the search path is relative to the current working directory), loading their values prior to the arguments given to the `GenLauncher` constructor: in this example, the parameter `mzprime` is first read from the file `parameters.txt` and then from the parameter list.

```python
from DarkNews.GenLauncher import GenLauncher
gen_object = GenLauncher(param_file="parameters.txt",
↪   mzprime=1.4)
```

This is exactly equivalent to running the following line through the binary `dn_gen`:

```
dn_gen --param-file parameters.txt --mzprime 1.4
```

The `parameters.txt` file may look like the following.

parameters.txt

```
hbar = 6.582119569e-25 # GeV s
c = 299792458.0 # m s^-1
gD = 2.0
alphaD = gD^2 / (4*PI)
epsilon = 1e-2
# chi = 0.0031
Umu5 = 1e-3
UD5 = 35.0
m4 = 0.080
m5 = 0.140
mzprime = 1.4
HNLtype = "majorana"
neval = 1000
```

The following rules apply when writing an input parameter file:

1. every line should be in the form of an assignment statement;

2. a line is commented out by prepending it with `#`;

3. the following input types are allowed:

   integers: e.g. `5`;

   floats: written with decimals or in exponential notation, e.g. `5.0` or `5e10` (`5.` will result in an error);

   strings: encapsulated with either single ‘ or double " quotation marks, e.g. `"DarkNews"`, or `‘DarkNews’`; escape character is the backslash \;

   booleans: by explicitly writing `True` or `False`. These are case-insensitive;

   lists: `Python`-like by encapsulating a comma-separated list of elements in square brackets; multi-line lists are also allowed;

4. it is possible to define any kind of variable, using also names not included among the allowed input parameters; variables can be defined according to the usual `Python` conventions for the variable name (only letters, digits and underscores) and with the assignment operator `=`. It is not allowed to name variables after program-defined names;

5. inputs may be specified using mathematical expressions, according to the following rules:

   (i) `+`, `-`, `*`, `/` are the usual mathematical operators;

   (ii) `^` is used for exponentiation (do not use `**`);

   (iii) round brackets `(` and `)` can be used;

   (iv) `e` (case-insensitive, isolated: not inside float numbers) is understood as `Python` `math.e = 2.718281828`;

   (v) `pi` (case-insensitive) is understood as `Python` `math.pi = 3.1415926535`;

(vi) `sin(<expr>)`, `cos(<expr>)`, `tan(<expr>)` are the usual trigonometric functions;

(vii) `exp(<expr>)` is the usual exponential function;

(viii) `abs(<expr>)` is the absolute value;

(ix) `sgn(<expr>)` $= -1$ if `<expr>` $< -10^{-100}$, $+1$ if `<expr>` $> 10^{100}$, 0 otherwise;

(x) `trunc(<expr>)` returns the truncated float `<expr>` to integer;

(xi) `round(<expr>)` is the integer part of the float number `<expr>`;

(xii) `sum(<list>)` will sum each element of the list `<list>`, returning a float number;

(xiii) any other string is interpreted as a variable which must already have been defined (the file is scanned from top to bottom).

b. Experiment definition files `DarkNews` provides the user with a range of pre-defined experiments, which may be accessed with a keyword through the `exp` input parameter:

```python
from DarkNews.GenLauncher import GenLauncher
gen_object = GenLauncher(exp="microboone")
```

equivalent to:

```
dn_gen --exp microboone
```

Keywords are listed along the experiments' features in Table I.

Alternatively, it is possible to define custom experiments via an experiment input file, e.g. `custom_experiment.txt`, passed as a parameter to the `GenLauncher` class or to the binary `dn_gen` in the same way as the parameter input file:

```python
from DarkNews.GenLauncher import GenLauncher
gen_object = GenLauncher(exp="custom_experiment.txt")
```

equivalent to:

```
dn_gen --exp custom_experiment.txt
```

The search path of the file is also relative to the current working directory. Notice that `DarkNews` first looks for an experiment with a keyword equal to those defined in Table I to then afterward search for an experiment input file. The experiment input file may look like the following:

custom_experiment.txt

```
name = "MicroBooNE"
fluxfile =
    "../fluxes/MicroBooNE_BNB_numu_flux.dat"
flux_norm = 1.0
erange = [
    0.05,
    9
]
nuclear_targets = [
    'Ar40'
]
fiducial_mass = 85.0 # tons
fiducial_mass_per_target = [
    fiducial_mass
]
POTs = 6.8e+20
```

In this example, we define a custom MicroBooNE experiment that has a different number of POTs with respect to the default. Notice that in this case, we wanted to keep the same definition for the neutrino fluxes file, so we defined the parameter `fluxfile` according to the path of the same file used by the default MicroBooNE experiment definition and listed in Table I. In case we wanted to change the flux file, we could have set the path of a file relative to the directory where the experiment input file is stored. Flux files are formatted as the following: `Enu nue numu nutau nuebar numubar nutaubar`, the same one used in GLoBES [41, 42], where E_ν is in GeV and fluxes in $\nu/\text{cm}^2/\text{POT}/\text{GeV}$. In principle, they can have any units, which is normalized with the parameter `flux_norm`.

A list of the possible parameters that can be set is shown in Table V.

B. Output formats

We now discuss the outputs of the event generation. Internally, `DarkNews` uses a pandas dataframe containing all event information and generation setup. The format is detailed in Appendix A.

a. Pandas dataframe `DarkNews` creates a pandas dataframe with all event information and 4-momenta built from the `vegas` samples of each process chain. It then combines all of these into one final dataframe, which is pickled and saved to a file. It contains all the 4-momenta of the incoming and outgoing particles, including any intermediary particles, as well as weights and additional information about the events. The dataframe index is a MultiIndex with columns and sub-columns. That is to say, the dataframe is a 3-dimensional object with rows representing different events and columns representing an event property e.g. the 4-momenta of the particles, or the event rate weight. Sub-columns represent a Lorentz index (0,1,2,3) to denote t, x, y, z coordinates, or E, p_x, p_y, p_z momentum components.

Detector	POT (10^{20})	fiducial mass (t)	material	neutrino flux	keyword	flux file
MiniBooNE FHC/RHC	18.75 (11.27)	818	CH_2	[43] (data release)	`miniboone_fhc` `miniboone_rhc`	`MiniBooNE_FHC.dat` `MiniBooNE_RHC.dat`
MicroBooNE	12.25	85	Ar	[43] (data release)	`microboone`	`MicroBooNE_BNB_numu_flux.dat`
MINERνA LE FHC	3.43	6.10 (in [44])	C_8H_8	[45] (data release)	`minerva_le_fhc`	`NUMI_FHC_LE.dat`

MINERνA ME FHC/RHC	11.6 (12.2)	6.10 (in [46])	C_8H_8	[47] (digitized)	`minerva_me_fhc`
					`minerva_me_rhc`
						`NUMI_FHC_ME.dat`
						`(NUMI_RHC_ME.dat)`

NOνA ND LE	13.6	193 (in [48])	H (10.8%), C (66.7%), O (3.00%), Cl (16.1%), Ti (3.20%)	[49] (data release)	`nova_le_fhc`	`NOvA_FHC.dat`
MINOS LE	10.56	28.6 (in [50])	C (20%), Fe (80%)	[47] (digitized)	`minos_le_fhc`	`NUMI_FHC_LE.dat`
T2K ND280	19.7	22.34 (in [51])	H (54%), C (3.6%), O (1.8%), Cu (15%), Zn (24%), Pb (1.9%)	[52] (digitized)	`nd280_fhc`	`ND280_FHC.dat`

DUNE LArND (1.2 MW, 120 GeV)	100	30	Ar	[53] (data release)	`dune_nd_fhc`
					`dune_nd_rhc`
						`DUNE_ND_FHC.dat`
						`(DUNE_ND_RHC.dat)`

FASERν	units of 1/150fb^{-1}	1.2	W	[54] (data release)	`fasernu`	`FASERnu.dat`
NuTeV	0.05	690	C (2%), Fe (98%)	[55] (digitized)	`nutev_fhc`	`NuTeV_FHC.dat`

TABLE I. The default parameters assumed by `DarkNews` for each different experiment. Note that the values above do not represent the full useful detector mass for each detector, which may be much larger depending on the analysis under consideration. Flux files can be found in `"src/DarkNews/include/fluxes"` together with their original sources.

Also included within the attributes of the pandas dataframe is the information used in the event generation. The attributes contain the model and detector classes used at the generation stage, the name of the generation (which can be specified by the user), and the lifetimes of all heavy neutrinos used. The detector and model classes can be accessed via `df.attrs["model"]` and `df.attrs["experiment"]`, and contain all information regarding the model parameters, neutrino fluxes, target material, exposure, fiducial mass, etc.

b. sparse It is also possible to remove less crucial information from the dataframe if one is not interested in non-observable particles or metadata, e.g. the four-momenta of invisible particles, or model parameters. This "sparse" dataframe can be useful when running a large number of simulations on a grid to save storage space.

c. Other `pandas`/`numpy` formats Other output formats are also available. One example is the `.parquet` format, popular for its compression and fast read-and-write operations. The pandas dataframe saved to the parquet file is stripped of its `attrs` (no metadata). Another possibility is to print all the floats in the dataframe to a `numpy` array. This is not necessarily advantageous since pandas already work rather seamlessly with `numpy`, but may be useful if one wants to avoid loading a pandas object when reading the event files. The column names are not included in the `.npy` files, so the user has to be careful to correctly identify the column indices.

d. HEPevt Output formats that are more standard in particle physics can also be used. The first one is `HEPevt`, a legacy `Pythia` format that is also used by several other generators. There are two implementations of HEPevt formats: one legacy one that includes the event weight next to the event number and the number of particles in the event, and a more standard one implemented with the help of the scikit-hep package `pyhepmc` [56]. The latter format cannot be used with weighted events.

e. HepMC Finally, we also implement the more modern format `HepMC3` [57, 58] as well as the older `HepMC2` format. These are also written using the `pyhepmc` package [56]. This is a different `Python` binding for `HepMC3` from the one implemented in [58], and allows

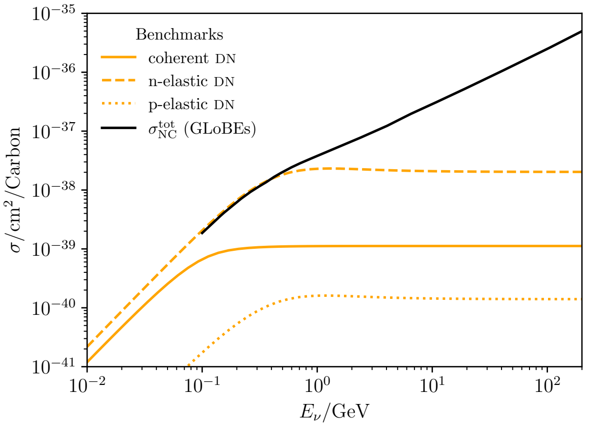

FIG. 3. A comparison between the neutrino-neutron neutral current cross section calculation and the cross sections in Refs [66, 67]. Neutron-elastic cross sections agree with the total neutral-current cross section up until the region where resonant and deep-inelastic-scattering regimes dominate.

us to stay fully inside Python, without the need for a local installation of HepMC3. The event weights are saved in both cases with the same notation defined in Appendix A.

III. MODELS

There are two model implementations in DarkNews. The first follows a completely generic parametrization of the interaction vertices, called a GenericHNLModel. The second, and more physical, implementation follows the "three-portal" model in Refs. [59, 60] and is referred to as the ThreePortalModel(). The latter option is preferred when quoting results in terms of a physical and self-consistent model, but the former may be advantageous when the user is interested in implementing a model of their own. For example, in the three-portal model, new scalar interactions between the heavy neutrinos and matter are rather suppressed and strongly constrained due to the m_f/v_{EW} suppression of the interactions. In that case, non-minimal scalar sectors like those discussed in Refs [61–65] can be implemented using the generic-model approach.

A. Model parameters

a. Generic model To assign interaction vertices, this model definition makes use of the following simplified interactions terms,

$$\begin{aligned}\mathcal{L}_{\mathrm{neutral}} \supset\ & c_{ij}\, Z_\mu \overline{N_i}\gamma^\mu N_j + d_{ij}\, Z'_\mu \overline{N_i}\gamma^\mu N_j \\ & + s_{ij}\, h' \overline{N_i} N_j + \frac{\mu^{\mathrm{tr}}_{ij}}{2}\overline{N_i}\sigma_{\mu\nu}F^{\mu\nu}N_j,\end{aligned} \tag{2}$$

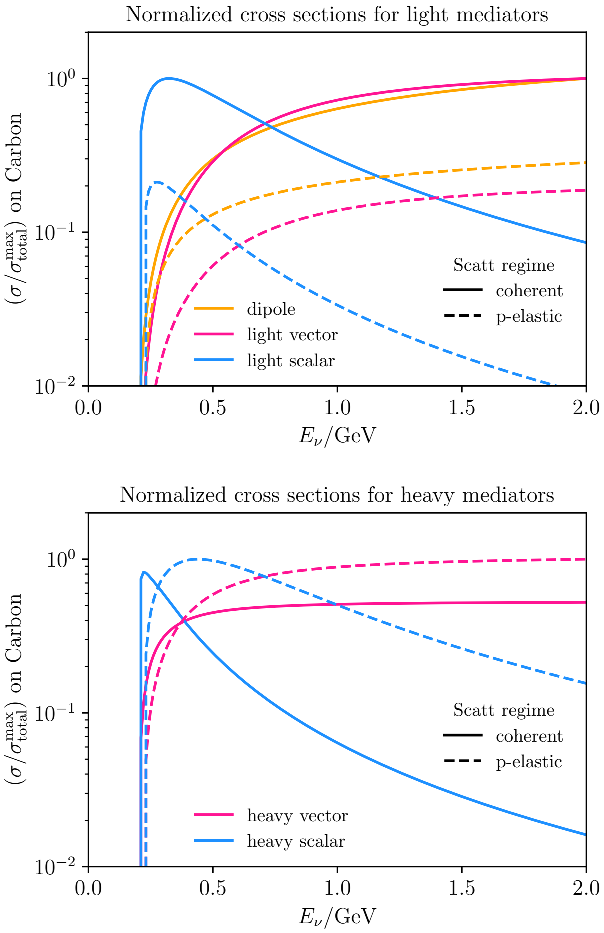

FIG. 4. The shape of the total upscattering cross sections for different types of upscattering processes in DarkNews, summed over final state helicities. The top panel shows cases with light mediators ($m_{Z'} = m_{h'} = 30$ MeV, including a transition magnetic moment), and the bottom one heavy mediators ($m_{Z'} = m_{h'} = 1.25$ GeV).

where N_i is some neutral lepton, light or heavy, in the set $\{\nu_e, \nu_\mu, \nu_\tau, N_4, N_5, N_6\}$. Note that for light neutrinos, we work with the flavor index, rather than the mass index. Here, c_{ij}, d_{ij}, s_{ij} and μ^{tr}_{ij} parametrize the couplings of the neutral leptons with the SM Z boson, the dark photon, the dark scalar, and transition magnetic moment terms. The mediators are also allowed to have couplings to other SM particles. For the couplings to charged particles, we include

$$\begin{aligned}\mathcal{L}_{\mathrm{charged}} \supset\ & Z_\mu \bar{f}\left(c_V^f\gamma^\mu + c_A^f\gamma^\mu\gamma^5\right) f \\ & + Z'_\mu \bar{f}\left(d_V^f\gamma^\mu + d_A^f\gamma^\mu\gamma^5\right) f \\ & + h'\bar{f}\left(d_S^f + d_P^f\gamma^5\right) f,\end{aligned} \tag{3}$$

where $f = \{u, d, e\}$. It is always assumed that the direct couplings to electrons and muons are the same.

fermion flavor		vector coupling	axial-vector coupling
Z_μ	charged leptons, $l^\pm$	$c^l_V = \frac{g}{2c_W}\left[c_\beta\left(-\frac{1}{2}+2s^2_W\right)+\frac{3}{2}s_\beta s_W t_\chi\right]$	$c^l_A = \frac{g}{2c_W}\left[-\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	up-type quarks, q^u	$c^u_V = \frac{g}{2c_W}\left[c_\beta\left(\frac{1}{2}-\frac{4}{3}s^2_W\right)-\frac{5}{6}s_\beta s_W t_\chi\right]$	$c^u_A = \frac{g}{2c_W}\left[\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	down-type quarks, q^d	$(c^d_{ij})_V = \frac{g}{2c_W}\left[c_\beta\left(-\frac{1}{2}+\frac{2}{3}s^2_W\right)+\frac{1}{6}s_\beta s_W t_\chi\right]$	$(c^d_{ij})_A = \frac{g}{2c_W}\left[-\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	heavy neutral lepton, N	$c^N_V = \frac{g}{4c_W}U^\dagger_{\alpha i}U_{\alpha j}\left(c_\beta+s_\beta s_W t_\chi\right) + \frac{s_\beta\, g_D}{2c_\chi}U^\dagger_{Di}U_{Dj}$	$c^N_A = -c^N_V$
Z'_μ	charged leptons, $l^\pm$	$d^l_V = \frac{g}{2c_W}\left[c_\beta\left(-\frac{1}{2}+2s^2_W\right)+\frac{3}{2}s_\beta s_W t_\chi\right]$	$d^l_A = \frac{g}{2c_W}\left[-\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	up-type quarks, q^u	$d^u_V = \frac{g}{2c_W}\left[c_\beta\left(\frac{1}{2}-\frac{4}{3}s^2_W\right)-\frac{5}{6}s_\beta s_W t_\chi\right]$	$d^u_A = \frac{g}{2c_W}\left[\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	down-type quarks, q^d	$d^d_V = \frac{g}{2c_W}\left[c_\beta\left(-\frac{1}{2}+\frac{2}{3}s^2_W\right)+\frac{1}{6}s_\beta s_W t_\chi\right]$	$d^d_A = \frac{g}{2c_W}\left[-\frac{c_\beta+s_\beta s_W t_\chi}{2}\right]$
	heavy neutral lepton, N	$(d^N_{ij})_V = -\frac{g}{4c_W}U^\dagger_{\alpha i}U_{\alpha j}\left(s_\beta-c_\beta s_W t_\chi\right) + \frac{c_\beta\, g_D}{2c_\chi}U^\dagger_{Di}U_{Dj}$	$(d^N_{ij})_A = -d^N_V$

TABLE II. Vector and axial-vector couplings of the SM charged fermions and heavy neutral leptons (which include the SM light neutrinos) with the modified Z_μ and new Z'_μ gauge bosons. Note that $c_x = \cos x$ and $s_x = \sin x$, where s_W is the sine of the Weinberg angle, $\tan 2\beta \equiv (2s_W s_\chi c_\chi)/(c^2_\chi - s^2_W s^2_\chi - \mu^2)$, $\mu = m_{Z'}/m_Z$, and $\chi = \epsilon/c_W$. In these expressions, we have assumed that dark neutrinos have a unit charge under the dark gauge group.

b. Three-portal model It is also possible to set the dark photon and dark scalar interaction vertices using the three portal model of Refs. [59, 60], which features a dark neutrino, ν_D, in addition to the standard sterile neutrino, ν_N, and the dark photon and dark scalar mediators. In this case, the interaction vertices defined in eq. (3) may be computed with the parameters, $U_{\alpha i}$, U_{Di}, ϵ, g_D, and θ, as featured in the code. Here, $U_{\alpha i}$ refers to the neutrino mixing matrix elements in the light neutrino sub-block, U_{Di} to the elements in the dark neutrino sub-block, ϵ to the kinetic mixing parameter, g_D to the coupling of the dark photon to the dark neutrino, and θ to the mixing between the SM Higgs and new dark scalar mediator.

In the language of the couplings introduced in the generic model, the full expressions for the couplings in the three-portal model are given in Table II (see also Ref. [68]). The scalar couplings to heavy neutrinos is always set in a model-independent way, as it would otherwise be necessary to specify the Yukawa couplings and mixing elements between the sterile and the heavy neutrino, U_{Ni}. This approach is justified given that for most applications, interesting phenomenology can only be achieved in non-minimal scalar models for which the generic model approach is more appropriate.

In the limit of small portal couplings, we have that

$$\mathcal{L}_{\text{charged}} \supset e\epsilon q_f\, Z'_\mu \bar{f}\gamma^\mu f + \theta\frac{m_f}{v_{\text{EW}}}h'\bar{f}f, \tag{4}$$

where e is the electromagnetic (EM) coupling, q_f the EM charge of the fermion f, g the electroweak gauge coupling, c_W the cosine of the Weinberg angle, m_f the mass of the fermion f, and v_{EW} the Higgs vev. Scalar-nucleon interactions are described according to Ref. [69]. In this model, charged-current diagrams in the HNL decays are also taken into account, and scale proportionally to $U_{\alpha i}$.

B. Cross sections

We now discuss how the analytical expressions for the upscattering cross sections are calculated. First, let us consider the process of a normalized low-energy neutrino

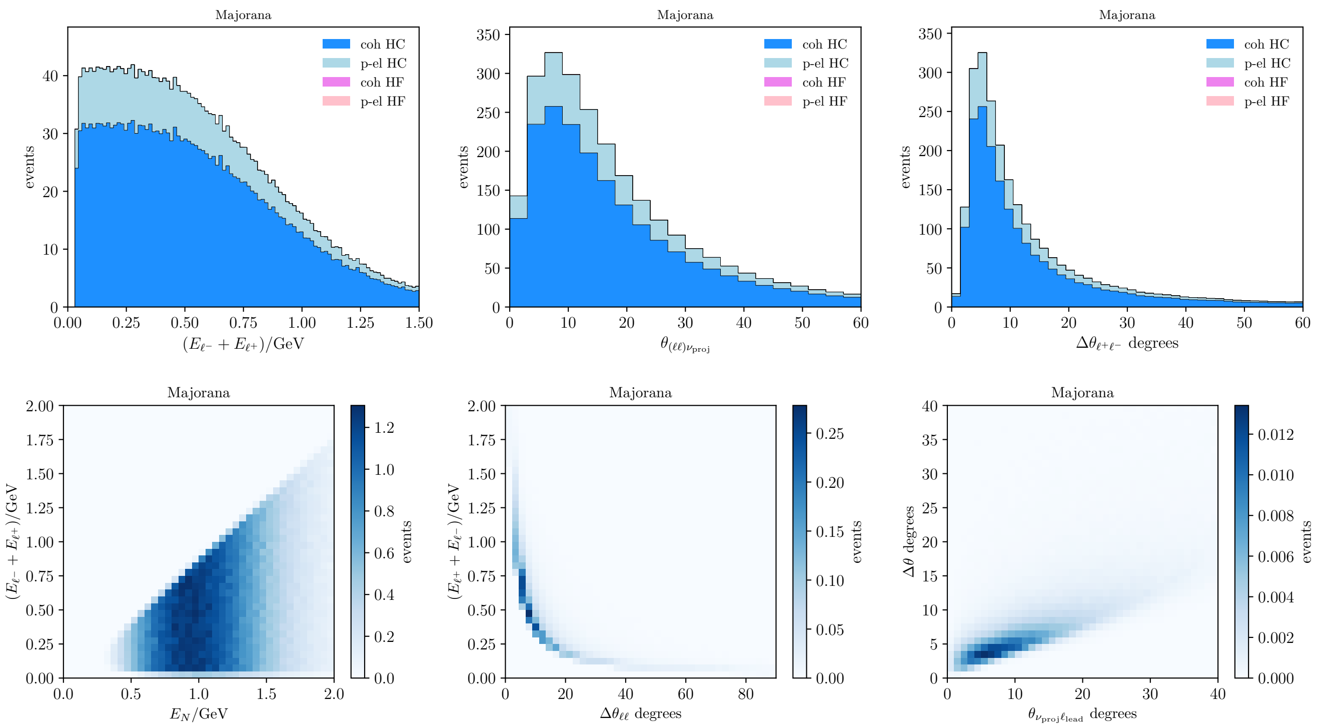

FIG. 5. Stacked histograms of the kinematical distributions of the final states produced in the neutrino-nucleus upscattering. Heavy neutrino is a Majorana particle with model parameters taken to be: $g_D = 1$, $\epsilon = 10^{-4}$, $m_5 = 300$ MeV, $m_4 = 100$ MeV, and $m_{Z'} = 30$ MeV. See text for definitions.

flavor state

$$|\hat{\nu}_\alpha\rangle = \frac{\sum_{i=1}^{3} U_{\alpha i} |\nu_i\rangle}{\sqrt{\sum_i^3 |U_{\alpha i}|^2}} \tag{5}$$

scattering on a hadronic target $\mathcal{H}$, either a nucleus or free nucleon, producing a heavy neutral lepton N_j with helicity h,

$$\hat{\nu}_\alpha(k) + \mathcal{H}(p) \to N_j^{h=\pm1}(k') + \mathcal{H}(p+q), \tag{6}$$

where $q = k - k' = (\nu, \vec{q})$ is the momentum transfer to the hadronic target. We define $\nu = E_\nu - E_N$ and $Q^2 \equiv -q^2$.

For convenience of notation and to keep the connection to the indexing used in the code, we will assign $i = \{e, \mu, \tau\} \equiv \{1, 2, 3\}$. This effectively means that we are considering a diagonal mass matrix for the three light neutrinos, and assuming that heavy neutrinos with $i \geq 4$ always decohere from the light states.

For the sake of illustration, we will assume that neutrino upscattering is mediated only by the Z boson and the dark photon. For a massless projectile incident on a nucleon of energy E_p, we can write the single differential cross section as

$$\frac{\mathrm{d}\sigma}{\mathrm{d}Q^2}(\nu_i\mathcal{H} \to N_j^h\mathcal{H}) = \frac{1}{64\pi s|\vec{p}_\nu^{\,\rm CM}|^2}\mathrm{L}^h_{\mu\nu}\left[\frac{(d_{ij}d_V^{\mathcal{H}})^2}{(Q^2+M_{Z'}^2)^2}\mathrm{H}^{\mu\nu}_{\rm EM} + \frac{(c_{ij}c_V^{\mathcal{H}})^2}{(Q^2+M_Z^2)^2}\mathrm{H}^{\mu\nu}_{\rm NC} + \frac{2c_{ij}d_{ij}c_V^{\mathcal{H}}d_V^{\mathcal{H}}}{(Q^2+M_Z^2)(Q^2+M_{Z'}^2)}\mathrm{H}^{\mu\nu}_{\rm int}\right], \tag{7}$$

where the leptonic tensor is given by

$$\begin{aligned}\mathrm{L}^h_{\mu\nu} = -\frac{g^{\mu\nu}}{2}&\left[m_\nu^2 + m_N^2 + Q^2 - 2h(k\cdot s)m_N\right] \\ &-(k^\mu s^\nu + k^\nu s^\mu)h\, m_N + (k^\mu k'^\nu + k^\nu k'^\mu) \\ &+ i\epsilon^{\mu\nu\rho\sigma}\left[k^\rho k'^\sigma - hm_N k^\rho s'^\sigma\right],\end{aligned} \tag{8}$$

which in the limit of $m_\nu = m_N = 0$ is the usual weak leptonic tensor $\mathrm{L}_{\mu\nu} = -(Q^2/2)g^{\mu\nu} + (k^\mu k'^\nu + k^\nu k'^\mu) + i\epsilon^{\mu\nu\rho\sigma}k^\rho k'^\sigma$. The hadronic tensors are determined for each scattering regime assuming a spin-1/2 nucleus and nucleon.

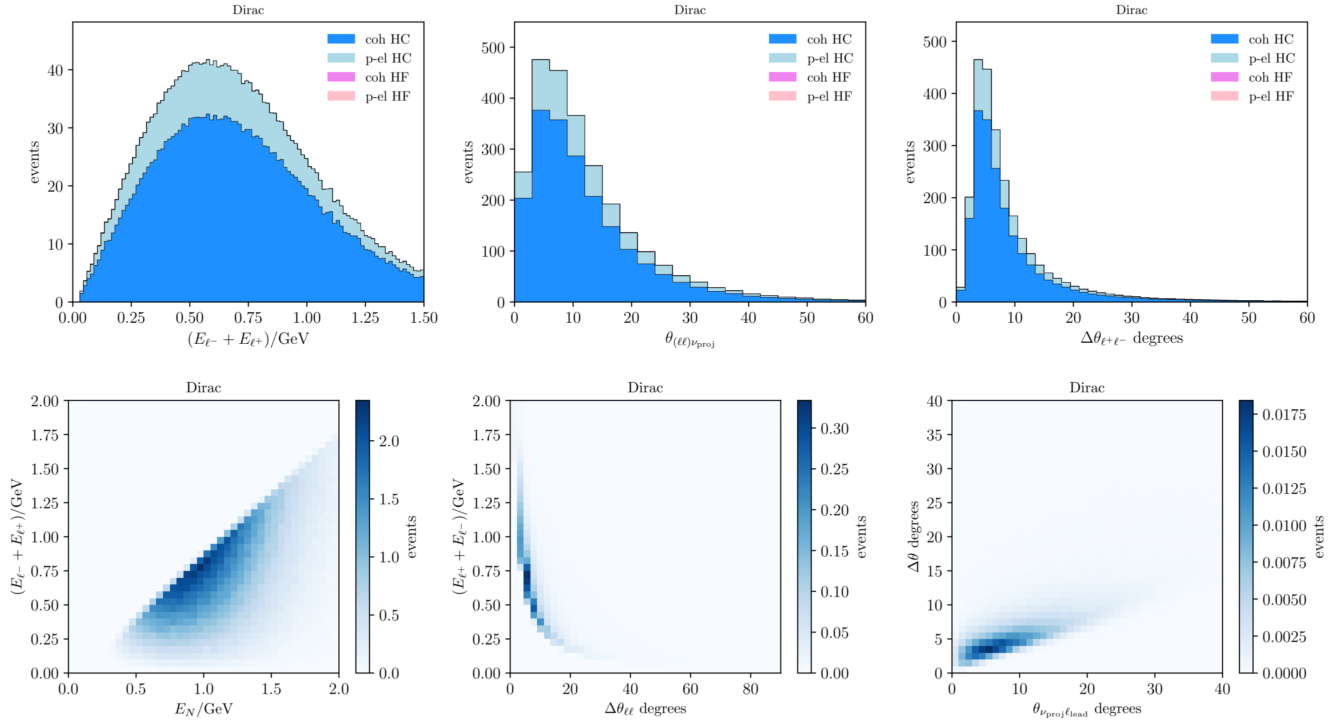

FIG. 6. Stacked histograms of the kinematical distributions of the final states produced in the neutrino-nucleus upscattering. Heavy neutrino is a Dirac particle with model parameters taken to be: $g_D = 1$, $\epsilon = 10^{-4}$, $m_5 = 300$ MeV, $m_4 = 100$ MeV, and $m_{Z'} = 30$ MeV. See text for definitions.

a. Form factors For coherent scattering, we use only the electric form factor for γ, Z, and Z' exchange (with the appropriate interaction vertex), while for a scalar exchange we implement the formalism in Ref. [69]. For the proton and neutron elastic cases, we implement the relevant Dirac, Pauli, and Axial form factors. Nuclear form factors are given by either the Fourier-Bessel parametrization [70–72] when nuclear data is available (given in machine-friendly format by Ref. [73]), or by a Fermi-Symmetrized function when they are not. These are described in more detail in the Appendix of Ref. [35].

b. Helicity As we assume the neutrino beam is fully polarized, the only helicity that we keep track of is that of the produced HNL. The scattering can then be helicity flipping, where the outgoing HNL has the opposite helicity of the initial projectile, or helicity-conserving, where they are the same. Depending on the interaction, one of these channels dominates over the other. For instance, scalar interactions and transition magnetic moments couple two separate chiralities and, therefore, lead to dominance in the helicity flipping channels. The opposite is true for vectorial interactions mediated by the dark photon. The decay of the HNL is then performed using the relevant helicity state produced. This allows for a proper comparison of the kinematics in the case of Majorana and Dirac neutrinos.

c. Additional contributions If one, or two, of the heavy neutrinos are long-lived with respect to the baseline of the experiment, their flux may also be non-negligible at the detector. In addition to their decays-in-flight (which are not included in `DarkNews`), they can also initiate upscattering processes. It is worthwhile to estimate such processes since the upscattering cross section involving only heavy neutrinos is no longer proportional to portal parameters but rather to the large couplings between N_i and N_j pairs. While `DarkNews` does not consider such processes, the user can obtain an estimate of this rate by reweighing light neutrino upscattering.

For instance, in models with N_4 and N_5 and a dark photon, the N_4-initiated process $N_4\mathcal{H} \to N_5\mathcal{H} \to \nu e^+e^-\mathcal{H}$ can be estimated by re-weighting the process $\hat{\nu}_\mu\mathcal{H} \to N_5\mathcal{H}$. Taking into account that the flux of N_4 would depend on its parent meson and including only one polarization structure, the ratio of N_4- and ν_μ-initiated upscattering events is

$$R = \sum_M \left(\frac{d_{45}}{d_{\mu5}}\right)^2 \rho\left(\frac{m_4^2}{m_M^2}, \frac{m_\mu^2}{m_M^2}\right) \frac{\Phi_M(E_\nu)}{\Phi(E_\nu)} P_{\rm surv}, \qquad (9)$$

where $\rho(a,b) = |U_{\mu4}|^2(a + b - (a-b)^2)\sqrt{\lambda(1,a,b)}$, $\lambda(1,a,b)$ is the Källén function, $\Phi_M(E_\nu)$ is the flux of neutrinos from the meson M, $\Phi(E_\nu)$ is the total neutrino flux used in the simulation, and $P_{\rm surv}$ is the survival prolbability of N_4. Note that this ratio scales as $|U_{\mu4}|^2/|U_{\mu5}|^2$ and so this additional contribution is not fully determined by the parameters that enter into the

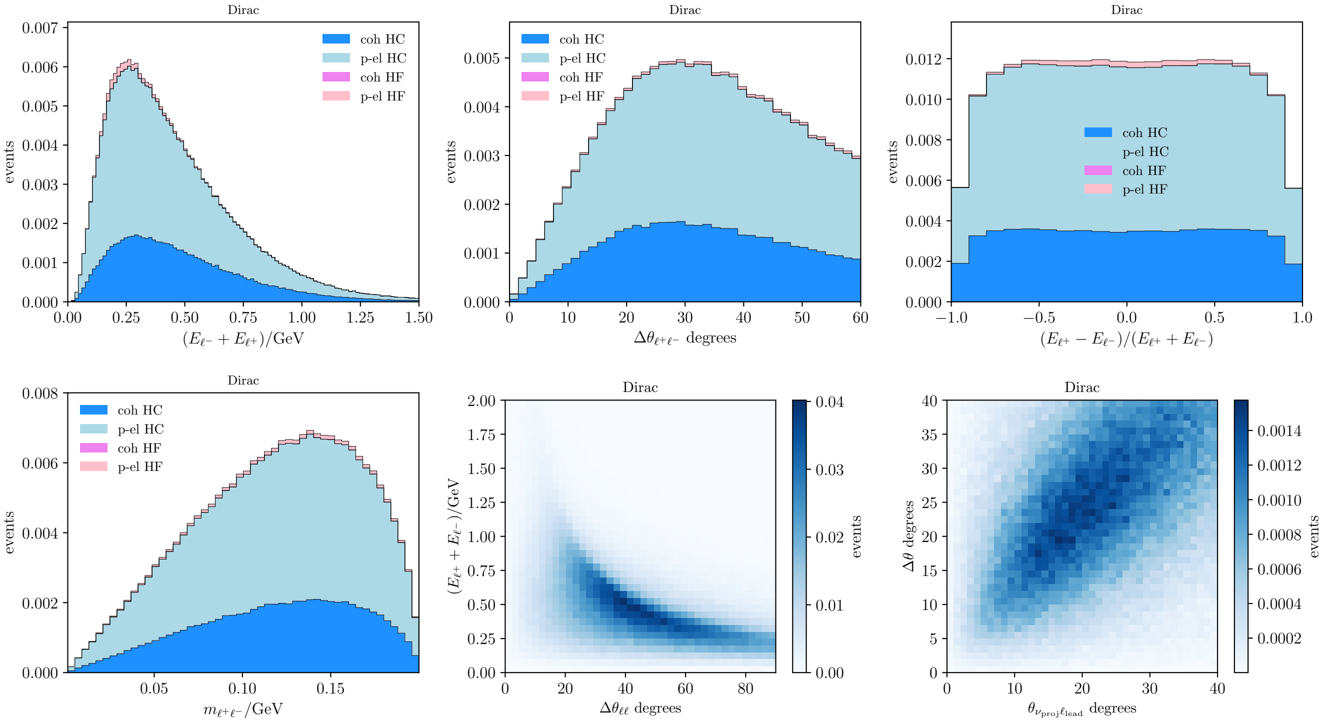

FIG. 7. Stacked histograms of the kinematical distributions of the final states produced in the neutrino-nucleus upscattering. Heavy neutrino is a Dirac particle with model parameters taken to be: $g_D = 1$, $\epsilon = 10^{-4}$, $m_5 = 300$ MeV, $m_4 = 100$ MeV, and $m_{Z'} = 1.25$ GeV. See text for definitions.

ν_μ upscattering included in DarkNews. In addition, in many cases $P_{\rm surv}$ is negligible, or $|U_{\mu 4}|^2$ is constrained to be very small by decay-in-flight searches.

C. Event rate

The differential experimental event rate for dileptons or single photons produced in the subsequent decay of the HNL is given by

$$\frac{\mathrm{d}N(E_\nu)}{\mathrm{d}Q^2\,\mathrm{dPS}} = \frac{\mathrm{d}\sigma(E_\nu)}{\mathrm{d}Q^2}\left(\frac{\mathrm{d}\Gamma_{N\to\nu X}}{\mathrm{dPS}}\frac{1}{\Gamma_{\rm tot}}\right), \qquad (10)$$

where $\mathrm{d}\Gamma_{N\to\nu X}/\mathrm{dPS}$ is the differential decay rate of the HNL into a neutral lepton and $X = e^+e^-, \mu^+\mu^-$ or γ, and $\Gamma_{\rm tot}$ is the total decay rate of the HNL. The whole quantity in parentheses can be thought of as a differential branching ratio.

In the final dataframe, DarkNews stores the left-hand side of Equation (10) as "w_event_rate", the differential cross section as "w_flux_avg_xsec", and the differential decay rate $\mathrm{d}\Gamma_{N\to\nu X}/\mathrm{dPS}$ as "w_decay_rate_i", where $i = 0$ or 1, depending on the process. The event rate is properly normalized to the experimental exposure and number of scattering targets, such that the total event rate can be obtained by `df.w_event_rate.sum()` . The total decay rate of the HNLs is stored in the attributes of the dataframe.

IV. EXAMPLES

In this section, we show some simple examples of the usage of DarkNews. First, we note that earlier versions of this code have been used in the following studies: Refs. [59, 60] with MiniBooNE predictions, Ref. [74] studying MINERvA, and Ref. [75] studying ND280, the near detector of T2K. For this version, we have provided comprehensive examples in the Examples/ folder of the repository in the form of documented Jupyter notebooks.

We now show how to compute cross sections and generate events, showing some kinematical distributions.

A. Cross sections

It is possible to get differential and total cross sections from DarkNews. Differential cross sections are always given in terms of the momentum exchange Q^2, with limits given by `upscattering_Q2min(Enu, mHNL, M)` and `upscattering_Q2max(Enu, mHNL, M)` , where Enu, mHNL, and M stand for the neutrino energy, heavy neutrino mass, and nuclear/nucleon mass, respectively.

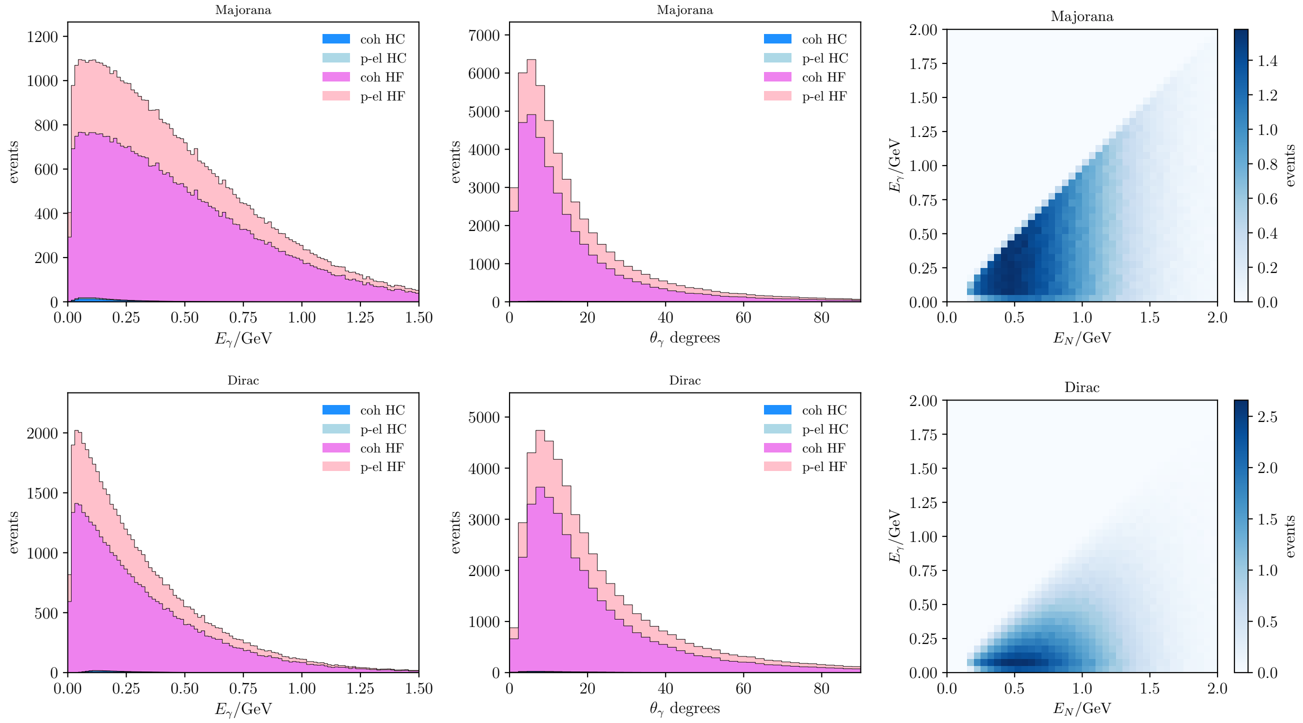

FIG. 8. Stacked histograms with the kinematical distributions of the final states produced in neutrino-nucleus upscattering for a transition magnetic moment model. For this point, we use a Majorana heavy neutrino for the top row and a Dirac heavy neutrino for the bottom row with $m_4 = 150$ MeV and $\mu^{\rm tr}_{\mu 4} = 1$ PeV^{-1}. See the text for definitions.

```
import numpy as np
import pandas as pd

import DarkNews as dn
from DarkNews import GenLauncher
from DarkNews import phase_space
from DarkNews import pdg

# targets
C12     = dn.detector.NuclearTarget("C12")

# model
SM_like = dn.model.ThreePortalModel(m4 = 0.001,
↪  Umu4=1e-3, epsilon=0, gD=0)

# xsec calculators
calc_NC_coh    = dn.UpscatteringProcess(
TheoryModel = SM_like,
scattering_regime = 'coherent',
nu_projectile=pdg.numu,
nu_upscattered=pdg.neutrino4,
helicity='conserving',
nuclear_target=C12)

# calculating it
xsec_NC = calc_NC_coh.total_xsec(Enu =
↪  np.geomspace(0.01,2e2,50), NEVAL=1000, NINT=10)
```

B. Upscattering at MiniBooNE

We now generate events for the MiniBooNE experiment in neutrino mode for a $3+2$ heavy neutrino model. The resulting kinematic distributions for a model with a light dark photon mediator can be found in Figure 5 and Figure 6 for Majorana and Dirac HNLs, respectively, and in Figure 8 for a transition magnetic moment model with Dirac neutrinos. In all cases, the plots are easily obtained with the `--make-summary-plots` argument.

For the light Z' cases in Figure 6 and Figure 5 we show the sum of the e^+e^- energy, the angle between the projectile neutrino (in the $\hat{z}$-direction) and the sum of the e^+e^- momenta referred to as $\theta_{(\ell\ell)\nu_{\rm proj}}$, the opening angle between the two charged lepton $\theta_{\ell\ell}$, the angle between the projectile neutrino and the leading (highest energy) charged lepton $\theta_{\nu_{\rm proj}\ell}$, and the energy of the parent HNL E_N. For the heavy Z' in Figure 7, we also show the energy asymmetry $(E_{\ell^+} - E_{\ell^-})/(E_{\ell^+} + E_{\ell^-})$ and the invariant mass $m_{\ell^+\ell^-}$. The Dirac-Majorana differences in the heavy Z' case are much less pronounced and appear mostly in the energy asymmetry – for Dirac, the center region is slightly concave, while for Majorana it is slightly convex. For the invariant mass, it leans backward for Majorana, and forward for Dirac. The total energy and opening angles remain very similar.

For the transition magnetic moment case in Figure 8, we show the photon energy E_γ, the angle of the photon with respect to the projectile neutrino θ_γ, and the energy of the parent heavy neutrino E_N. The Dirac-Majorana differences in this case are also important for the total energy and angle.

In Figure 5 and Figure 6, we can see that the dominant mode of upscattering is the helicity-conserving one due to the vector nature of the dark photon interactions with matter. The dilepton angles with respect to the neutrino beam are also rather forward, as expected. Dirac HNLs lead to less forward angles and lower energy signatures, as well as other subtle effects. In Figure 8, on the other hand, the dominant upscattering mode is the helicity-flipping one since the transition magnetic moment interaction is of a similar chiral structure to the scalar case. The photon energy is lower energy for the same reason: the photon is preferentially emitted in the backward direction and tends to be lower energy and less forward.

V. CONCLUSIONS

We have presented DarkNews, an easy-to-install, easy-to-use, Python-based Monte-Carlo generator for new physics in neutrino-nucleus scattering. The code models neutrino upscattering to heavy neutrinos and their subsequent decays into dilepton pairs or single photons at neutrino experiments. New physics models may be defined by the user and can contain up to three heavy neutrinos. Additional interactions may be defined for the heavy neutrinos including e.g. a dark photon, a dark scalar particle, or a transition magnetic moment. We model coherent nuclear scattering, as well as free proton- and neutron-elastic scattering (including bound nucleons without nuclear modifications). The generator is able to output events to several formats, including the internally-used pandas dataframe and numpy ndarrays. Events may also be saved to HEPevt and HepMC2/3 files, both of which are frequently used by experimental collaborations.

We have provided examples of how to obtain differential and total cross sections, as well as kinematical distributions of e^+e^-, $\mu^+\mu^-$, and single-photon events.

In recent years, interest in the development of new event generators for neutrino experiments has grown and a number of sophisticated new tools join the existing event generators. While the implementation of new physics models in these newer tools exceeds the traditional generators in ease, there is a need for lighter-weight generators to be used by both the phenomenology and experimental communities. The power of DarkNews lies in its simplicity and speed and we believe it will be a valuable tool in the search for new physics in current and future neutrino experiments.

ACKNOWLEDGMENTS

We are grateful to G. Peter Lepage for correspondence and for implementing new features in vegas. We would like to thank Nicòlo Foppiani, Joshua Isaacson, Georgia Karagiorgi, and Mark Ross-Lonergan for suggestions that helped improve the code. The research of M.H. was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Research, Innovation and Science. The research of J.H.Z. has received funding / support from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860881-HIDDeN. The research of A.A is supported by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

Appendix A: Formats, input, and outputs

The input parameters for the generator are shown in Table III, while the input parameters that define the underlying new physics model to be used are shown in Table IV for the generic model of Equation (2) and Equation (3), as well as for the three portal model of [60]. The experiment definition parameters are listed in Tables I and III.

The format of the output pandas dataframe is shown in Appendix A. This dataframe is the main result of the generator and is used internally in DarkNews.

Parameter	Type	Default	Description
Experiment			
`exp`	*string*	`"miniboone_fhc"`	The experiment to consider: see text for more information
Monte-Carlo scope			
`nopelastic`	*bool*	`False`	Do not generate proton elastic events
`nocoh`	*bool*	`False`	Do not generate coherent events
`noHC`	*bool*	`False`	Do not include helicity conserving events
`noHF`	*bool*	`False`	Do not include helicity flipping events
`decay_products`	*string*	`"e+e-"`	Decay process of interest, choose among `["e+e-","mu+mu-","photon"]`
`nu_flavors`	*list of string*	`"nu_mu"`	Neutrino projectile, a list with elements among `["nu_e","nu_mu","nu_tau","nu_e_bar","nu_mu_bar","nu_tau_bar"]`
`sample_geometry`	*bool*	`True`	Whether to sample the detector geometry using DarkNews. If False or a geometry function is not found, the upscattering position is assumed to be $(0, 0, 0, 0)$.
`enforce_prompt`	*bool*	`False`	Force particles to decay promptly
`vegas` **integration options**			
`neval`	*int*	10000	Number of evaluations of integrand
`nint`	*int*	20	Number of adaptive iterations
`neval_warmup`	*int*	1000	Number of evaluations of integrand in warmup
`nint_warmup`	*int*	10	Number of adaptive iterations in warmup
`seed`	*int*	`None`	`numpy` random number generator seed used in `vegas`
Output format options			
`pandas`	*bool*	`True`	Save `pandas.DataFrame` as `.pckl` file
`parquet`	*bool*	`False`	Save `pandas.DataFrame` as `.parquet` file (`engine=pyarrow`)
`numpy`	*bool*	`False`	Save events in `.npy` files
`hepevt`	*bool*	`False`	Save events in HEPEVT-formatted text files
`hep_unweight`	*bool*	`False`	Unweight events when printing in any of the HEP formats (needs large statistics)
`unweighted_hep_events`	*int*	100	Unweight events when printing in any of the HEP formats (needs large statistics)
`sparse`	*bool*	`False`	if True, save only the neutrino energy, charged lepton or photon momenta, and weights. Not supported for HEP formats.
`make_summary_plots`	*bool*	`False`	if True, generates summary plots of kinematics in the ‘path‘
`path`	*string*	`"./"`	Path where to save run’s outputs
Output verbosity			
`loglevel`	*string*	`"INFO"`	Logging level, choose among `["INFO", "WARNING", "ERROR", "DEBUG"]`
`verbose`	*bool*	`False`	Verbose for logging
`logfile`	*string*	`None`	Path to log file; if not set, use std output

TABLE III: The input parameters that can be set in `DarkNews`. We separate them into broad categories regarding the behavior of the generator.

[1] If specified while $g_{\rm D}$ is not, ignore default $g_{\rm D}$ and compute it from this parameter.

[2] If specified while ε is not, ignore default ε and compute it from this parameter.

[3] It is also possible to specify couplings to `f` = `proton` and `f` = `neutron`, e.g. `cprotonV`. In that case, the nucleon couplings override the calculation based on the sum of quark couplings such as in `cprotonV` = 2`cuV` + `cdV`.

Parameter	Description
Dark sector spectrum	
`m{i}`	Mass of the heavy neutrinos ($i = 4, 5, 6$, default $m_4 = 150$ MeV)
`mzprime`	Mass of dark photon Z' (default $m_{Z'} = 1.25$)
`mhprime`	Mass of dark scalar h'
`HNLtype`	Dirac or Majorana HNLs? (default ”dirac”)
Common model-independent parameters	
`mu_tr_{ij}`	Transition magnetic moment defined in Equation (2) ($i = e, \mu, \tau, 4, 5, 6$)
`s_{ij}`	Scalar couplings defined in Equation (2) ($i = e, \mu, \tau, 4, 5, 6$)
Generic model parameter setting	
`c_{ij}`	Neutral lepton couplings of the SM Z boson ($i = e, \mu, \tau, 4, 5, 6$)
`d_{ij}`	Neutral lepton couplings of the dark photon Z' ($i = e, \mu, \tau, 4, 5, 6$)
`c{f}V`	SM fermions vector couplings of the SM Z boson ($f = e, u, d$) [c]
`c{f}A`	SM fermions axial-Vector couplings of the SM Z boson ($f = e, u, d$) [c]
`d{f}V`	SM fermions vector couplings of the dark photon Z' ($f = e, u, d$) [c]
`d{f}A`	SM fermions axial-vector couplings of the dark photon Z' ($f = e, u, d$) [c]
`d{f}S`	SM fermions scalar couplings of the dark scalar h' ($f = e, u, d$) [c]
`d{f}P`	SM fermions pseudo-scalar couplings of the dark scalar h' ($f = e, u, d$) [c]
Three-portal model parameter setting	
`ue{i}`	Mixing element between electron flavor and heavy neutrinos
`umu{i}`	Mixing element between muon flavor and heavy neutrinos ($\lvert U_{\mu 4}\rvert = 10^{-3}$)
`utau{i}`	Mixing element between tau flavor and heavy neutrinos
`ud{i}`	Mixing element between dark flavor and heavy neutrinos ($\lvert U_{D4}\rvert = 1$)
`gD`	$\mathrm{U}(1)_{\mathrm{d}}$ dark coupling ($g_{\mathrm{D}} = 1$)
`alphaD`	$\alpha_{\mathrm{D}} = g_{\mathrm{D}}^2 \,/\, (4\pi)$ [a]
`epsilon`	Kinetic mixing parameter ($\varepsilon = 10^{-3}$)
`epsilon2`	ε^2 [b]
`alpha_epsilon2`	$\alpha\varepsilon^2$ [b]
`chi`	$\chi = \varepsilon \,/\, c_w$ [b]
`theta`	Scalar mixing θ between the Higgs and h'

TABLE IV. Model input parameters that can be set in `DarkNews`. If a generic input parameter is set, the chosen model class is GenericHNLModel(), while if a three-portal model parameter is set, the chosen model class is ThreePortalModel(). It is not possible to mix the two. All parameters are assumed to be real numbers, and satisfy $X_{ij} = X_{ji}$. Inputs are given by X_{ij} with $i < j$.

Parameter	Type	Description
`name`	*string*	Name of the experiment (capital letters allowed);
`fluxfile`	*string*	Path of the fluxes file with respect to either the experiment file directory or the program internal flux files directory;
`flux_norm`	*float*	Flux normalization factor: all fluxes should be normalized so that the units are $\#\nu/(\mathrm{cm}^2\,\mathrm{GeV\,POT})$;
`erange`	*list of float*	Neutrino energy range $E_{\min} \leq E \leq E_{\max}$ in GeV;
`nuclear_targets`	*string*	Detector materials in the form of `"<element_name><mass_number>"`, e.g. `"Ar40"`;
`fiducial_mass`	*float*	Fiducial mass in tons;
`fiducial_mass_per_target`	*list of float*	Fiducial mass for each target in the same order as specified in the `nuclear_targets` parameter.

TABLE V. The experiment parameters that can be set to define an experiment. These can only be set via the experimental definition files discussed in Sec. II A 0 b

Column	Subcolumn	Type	Description
`P_projectile`	`0, 1, 2, 3`	*float*	4-momenta of beam neutrino
`P_decay_N_parent`	`0, 1, 2, 3`	*float*	4-momenta of HNL_parent
`P_target`	`0, 1, 2, 3`	*float*	4-momenta of nucleus
`P_recoil`	`0, 1, 2, 3`	*float*	4-momenta of recoiled nucleus
`P_decay_ell_minus`	`0, 1, 2, 3`	*float*	4-momenta of e-
`P_decay_ell_plus`	`0, 1, 2, 3`	*float*	4-momenta of e+
`P_decay_N_daughter`	`0, 1, 2, 3`	*float*	4-momenta of HNL_daughter / nu_daughter
`pos_scatt`	`0, 1, 2, 3`	*float*	upscattering position
`pos_decay`	`0, 1, 2, 3`	*float*	decay position of primary parent particle (N_parent) – secondary decay positions are not saved.
`w_decay_rate_0`		*float*	Weight of the decay rate of primary unstable particle: $\sum_i w_i = \Gamma_N$
`w_decay_rate_1`		*float*	Weight of the decay rate of secondary unstable particle: $\sum_i w_i = \Gamma_X$
`w_event_rate`		*float*	Weight for the event rate: $\sum_i w_i =$ event rate
`w_flux_avg_xsec`		*float*	Weight of the flux averaged cross section: $\sum_i w_i =$ exposure× $\int(\sigma\Phi_\nu)$
`target`		*string*	Target object, it will typically be a nucleus
`target_pdgid`		*int*	PDG id of the target
`scattering_regime`		*string*	Regime can be coherent, p-el, or n-el.
`helicity`		*string*	Helicity process: can be flipping or conserving; flipping is suppressed
`underlying_process`		*string*	String of the underlying process, e.g, `"nu(mu) + proton_in_C12 -> N4 + proton_in_C12 -> nu(mu) + e+ + e- + proton_in_C12"`

TABLE VI. The format of the main `pandas` dataframe used by `DarkNews`. To access the negative lepton energy, for example, one can simply do `df["P_decay_ell_minus","0"]` .

[1] J. Alexander et al. (2016), 1608.08632.

[2] P. Agrawal et al., Eur. Phys. J. C **81**, 1015 (2021), 2102.12143.

[3] B. Batell, M. Pospelov, and A. Ritz, Phys. Rev. D **80**, 095024 (2009), 0906.5614.

[4] R. Essig, R. Harnik, J. Kaplan, and N. Toro, Phys. Rev. D **82**, 113008 (2010), 1008.0636.

[5] P. deNiverville, M. Pospelov, and A. Ritz, Phys. Rev. D **84**, 075020 (2011), 1107.4580.

[6] S. N. Gninenko, Phys. Rev. D **85**, 055027 (2012), 1112.5438.

[7] P. deNiverville, D. McKeen, and A. Ritz, Phys. Rev. D **86**, 035022 (2012), 1205.3499.

[8] L. Buonocore, C. Frugiuele, and P. deNiverville, Phys. Rev. D **102**, 035006 (2020), 1912.09346.

[9] C. A. Argüelles et al., Rept. Prog. Phys. **83**, 124201 (2020), 1907.08311.

[10] A. M. Abdullahi et al., in *2022 Snowmass Summer Study* (2022), 2203.08039.

[11] M. Antonello et al. (MicroBooNE, LAr1-ND, ICARUS-WA104) (2015), 1503.01520.

[12] K. Abe et al. (T2K) (2019), 1901.03750.

[13] (2016).

[14] B. Abi et al. (DUNE) (2020), 2002.03005.

[15] M. Toups et al., in *2022 Snowmass Summer Study* (2022), 2203.08102.

[16] C. Andreopoulos et al., Nucl. Instrum. Meth. A **614**, 87 (2010), 0905.2517.

[17] J. Tena-Vidal et al. (GENIE), Phys. Rev. D **104**, 072009 (2021), 2104.09179.

[18] T. Golan, C. Juszczak, and J. T. Sobczyk, Phys. Rev. C **86**, 015505 (2012), 1202.4197.

[19] Y. Hayato and L. Pickering, Eur. Phys. J. ST **230**, 4469 (2021), 2106.15809.

[20] O. Buss, T. Gaitanos, K. Gallmeister, H. van Hees, M. Kaskulov, O. Lalakulich, A. B. Larionov, T. Leitner, J. Weil, and U. Mosel, Phys. Rept. **512**, 1 (2012), 1106.1344.

[21] N. D. Christensen and C. Duhr, Comput. Phys. Commun. **180**, 1614 (2009), 0806.4194.

[22] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr, and B. Fuks, Comput. Phys. Commun. **185**, 2250 (2014), 1310.1921.

[23] C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer, and T. Reiter, Comput. Phys. Commun. **183**, 1201 (2012), 1108.2040.

[24] J. M. Campbell et al., in *2022 Snowmass Summer Study* (2022), 2203.11110.

[25] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput. Phys. Commun. **176**, 367 (2007), hep-ph/0607059.

[26] G. Belanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput. Phys. Commun. **185**, 960 (2014), 1305.0237.

[27] G. Bélanger, F. Boudjema, A. Pukhov, and A. Semenov, Comput. Phys. Commun. **192**, 322 (2015), 1407.6129.

[28] G. Bélanger, F. Boudjema, A. Goudelis, A. Pukhov, and B. Zaldivar, Comput. Phys. Commun. **231**, 173 (2018), 1801.03509.

[29] M. Backovic, K. Kong, and M. McCaskey, Physics of the Dark Universe **5-6**, 18 (2014), 1308.4955.

[30] M. Backović, A. Martini, O. Mattelaer, K. Kong, and G. Mohlabeng, Phys. Dark Univ. **9-10**, 37 (2015), 1505.04190.

[31] F. Ambrogi, C. Arina, M. Backovic, J. Heisig, F. Maltoni, L. Mantani, O. Mattelaer, and G. Mohlabeng, Phys. Dark Univ. **24**, 100249 (2019), 1804.00044.

[32] C. Arina, J. Heisig, F. Maltoni, D. Massaro, and O. Mattelaer (2021), 2107.04598.

[33] L. Buonocore, C. Frugiuele, F. Maltoni, O. Mattelaer, and F. Tramontano, JHEP **05**, 028 (2019), 1812.06771.

[34] R. Abbasi et al. (IceCube), Comput. Phys. Commun. **266**, 108018 (2021), 2012.10449.

[35] N. W. Kamp, M. Hostert, A. Schneider, S. Vergani, C. A. Argüelles, J. M. Conrad, M. H. Shaevitz, and M. A. Uchida (2022), 2206.07100.

[36] J. Isaacson, S. Höche, D. Lopez Gutierrez, and N. Rocco, Phys. Rev. D **105**, 096006 (2022), 2110.15319.

[37] J. Isaacson, W. I. Jay, A. Lovato, P. A. N. Machado, and N. Rocco (2022), 2205.06378.

[38] G. P. Lepage, J. Comput. Phys. **27**, 192 (1978).

[39] G. P. Lepage, J. Comput. Phys. **439**, 110386 (2021), 2009.05112.

[40] E. Rodrigues et al., EPJ Web Conf. **245**, 06028 (2020), 2007.03577.

[41] P. Huber, M. Lindner, and W. Winter, Comput. Phys. Commun. **167**, 195 (2005), hep-ph/0407333.

[42] P. Huber, J. Kopp, M. Lindner, M. Rolinec, and W. Winter, Comput. Phys. Commun. **177**, 432 (2007), hep-ph/0701187.

[43] A. A. Aguilar-Arevalo et al. (MiniBooNE), Phys. Rev. D **79**, 072002 (2009), 0806.1449.

[44] J. Park et al. (MINERvA), Phys. Rev. D **93**, 112007 (2016), 1512.07699.

[45] L. Aliaga et al. (MINERvA), Phys. Rev. D **94**, 092005 (2016), [Addendum: Phys.Rev.D 95, 039903 (2017)], 1607.00704.

[46] E. Valencia et al. (MINERvA), Phys. Rev. D **100**, 092001 (2019), 1906.00111.

[47] L. Aliaga Soplin, Ph.D. thesis, William-Mary Coll. (2016).

[48] M. A. Acero et al. (NOvA), Phys. Rev. D **102**, 012004 (2020), 1902.00558.

[49] *NOvA flux histograms for 2017 analysis*, https://publicdocs.fnal.gov/cgi-bin/ShowDocument?docid=8, accessed: 2022-05-02.

[50] P. Adamson et al. (MINOS), Phys. Rev. D **94**, 072006 (2016), 1608.05702.

[51] S. Assylbekov et al., Nucl. Instrum. Meth. A **686**, 48 (2012), 1111.5030.

[52] *ND280 flux prediction for 2020 analysis*, https://t2k-experiment.org/result_category/flux/, accessed: 2022-05-02.

[53] B. Abi et al. (DUNE) (2021), 2103.04797.

[54] F. Kling and L. J. Nevay, Phys. Rev. D **104**, 113008 (2021), 2105.08270.

[55] S. E. Avvakumov, Ph.D. thesis, Rochester U. (2002).

[56] *pyhepmc-ng on github*, https://github.com/scikit-hep/pyhepmc.

[57] A. Buckley, P. Ilten, D. Konstantinov, L. Lönnblad, J. Monk, W. Pokorski, T. Przedzinski, and A. Verbytskyi, Comput. Phys. Commun. **260**, 107310 (2021), 1912.08005.

[58] *Hepmc3 on gitlab*, https://gitlab.cern.ch/hepmc/HepMC3.

[59] P. Ballett, M. Hostert, and S. Pascoli, Phys. Rev. D **101**, 115025 (2020), 1903.07589.

[60] A. Abdullahi, M. Hostert, and S. Pascoli, Phys. Lett. B **820**, 136531 (2021), 2007.11813.

[61] B. Dutta, S. Ghosh, and T. Li, Phys. Rev. D **102**, 055017 (2020), 2006.01319.

[62] A. Datta, S. Kamali, and D. Marfatia, Phys. Lett. B **807**, 135579 (2020), 2005.08920.

[63] W. Abdallah, R. Gandhi, and S. Roy, JHEP **12**, 188 (2020), 2006.01948.

[64] W. Abdallah, R. Gandhi, and S. Roy, Phys. Rev. D **104**, 055028 (2021), 2010.06159.

[65] W. Abdallah, R. Gandhi, and S. Roy (2022), 2202.09373.

[66] M. D. Messier (????), uMI-99-23965.

[67] E. A. Paschos and J. Y. Yu, Phys. Rev. **D65**, 033002 (2002), hep-ph/0107261.

[68] K. S. Babu, C. F. Kolda, and J. March-Russell, Phys. Rev. D **57**, 6788 (1998), hep-ph/9710441.

[69] J. M. Cline, K. Kainulainen, P. Scott, and C. Weniger, Phys. Rev. D **88**, 055025 (2013), [Erratum: Phys.Rev.D 92, 039906 (2015)], 1306.4710.

[70] G. Fricke, C. Bernhardt, K. Heilig, L. A. Schaller, L. Schellenberg, E. B. Shera, and C. W. de Jager, Atom. Data Nucl. Data Tabl. **60**, 177 (1995).

[71] H. De Vries, C. W. De Jager, and C. De Vries, Atom. Data Nucl. Data Tabl. **36**, 495 (1987).

[72] C. W. De Jager, H. De Vries, and C. De Vries, Atom. Data Nucl. Data Tabl. **14**, 479 (1974), [Erratum: Atom.Data Nucl.Data Tabl. 16, 580–580 (1975)].

[73] URL http://discovery.phys.virginia.edu/research/groups/ncd/index.html.

[74] C. A. Argüelles, M. Hostert, and Y.-D. Tsai, Phys. Rev. Lett. **123**, 261801 (2019), 1812.08768.

[75] C. A. Argüelles, N. Foppiani, and M. Hostert (2021), 2109.03831.