SNOwGLoBES : SuperNova Observatories with GLoBES

Joshua Albert[1], Alex Beck[1], Farzan Beroz[1], Rachel Carr[2], Huaiyu Duan[3], Alex Friedland[4], Nicolas Kaiser[5,1], Jim Kneller[6], Elise McCarthy[9], Alexander Moss[1], Diane Reitzner[7], Kate Scholberg[1*], Sebastian Torres-Lara[10], David Webber[8], and Roger Wendell[1]

[1]Department of Physics, Duke University, Durham, NC 27705

[2]Department of Physics, Columbia University, New York, NY 10027

[3]Department of Physics, University of New Mexico, Albuquerque, NM, 87131

[4]Los Alamos National Laboratory, Los Alamos, NM, 87545

[5]Department of Physics, Karlsruhe Institute of Technology, Germany

[6]Department of Physics, North Carolina State University, Raleigh, NC, 27695

[7]Fermilab, Batavia, IL, 60510-5011

[8]Department of Physics, University of Wisconsin, Madison, WI, 53706-1390

[9]Department of Physics and Astronomy, University of Rochester, Rochester, NY, 14620

[10]Department of Physics, University of Houston, Houston, TX, 77004

[*]schol@phy.duke.edu

December 7, 2021

Abstract

This document describes public software for computing interaction rates and distributions of observed quantities for supernova burst neutrinos in common detector materials. The intent is to provide a very simple and fast data package which can be used to test the observability of physics signatures in current and future detectors, and for evaluation of relative sensitivities of different detector configurations. The event estimates are made using available cross-sections and parameterized detector responses. Water, argon, xenon, germanium, scintillator, and lead-based configurations are included. The package makes use of GLoBES front-end software. SNOwGLoBES is not intended to replace full detector simulations, however
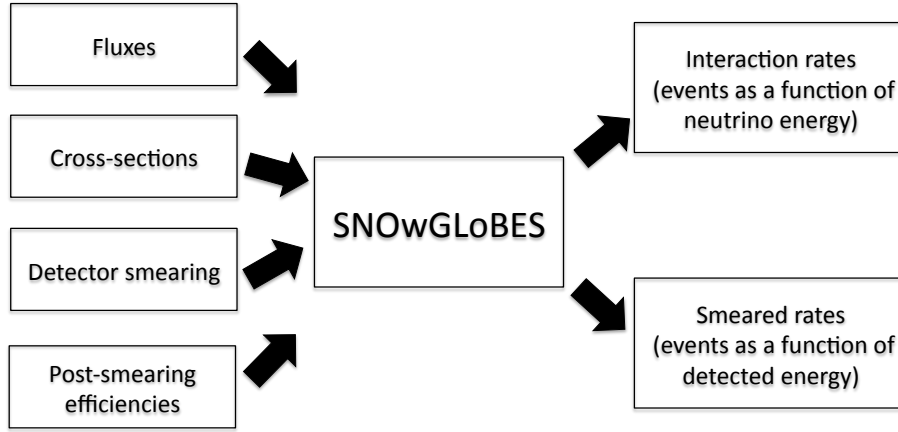
1

Figure 1: `SNOwGLoBES` data flow.

output should be useful for many types of studies. This document serves as a user's manual and provides references for the data files used.

1   Introduction

A stellar core collapse in the Milky Way or nearby will produce an enormous burst of neutrinos observable in terrestrial detectors. Such a burst will carry tremendous information about both astrophysics and particle physics: see [1] for recent reviews of physics reach and [2] for an overview of detection technology.

To enable fast, informative studies of the physics potential of the detection of supernova neutrinos, we have developed `SNOwGLoBES`, a simple software and database package to compute expected event rates by folding input fluxes with cross-sections and detector parameters. The output is in the form of interaction rates for each neutrino interaction channel as a function of incident neutrino energy, and "smeared" rates as a function of detected energy for each channel (*i.e.* the spectrum that actually be observed in a detector). We have chosen to do the event rate computation using parameterized detector responses, making use of the `GLoBES` software [3, 4]. We employ only the front-end rate engine part of `GLoBES`, and not the oscillation sensitivity part. `GLoBES` takes as input fluxes, cross sections, "smearing matrices" and post-smearing efficiencies. The smearing matrices incorporate both interaction product spectra and detector response. Fig. 1 gives a schematic overview of the approach.

Although with this approach information is lost with respect to a full event-by-event simulation using a neutrino interaction generator and detector simulation (correlations between energy and angle are lost, for

example), SNOwGLoBES nevertheless offers a fast, simple method useful for many studies.

SNOwGLoBES should enable studies from differing points of view, by allowing modification of the different kinds of input. For example:

- Using the standard cross-sections and detector parameters, a user can study whether specific physics signatures in the fluxes are observable with different detector configurations.

- Experimentalists can optimize detector configurations by using standard cross-sections and the sample fluxes, and modifying the smearing, efficiency and background files according to different configurations (for example, different PMT coverage in a water Cherenkov configuration). An early version of SNOwGLoBES was used in this way for the work described in reference [5].

- A user could also study systematic errors due to cross-section uncertainties (for example) by comparing event rates computed using cross-sections from different models or with modifications according to estimated uncertainties.

Currently, time-dependent fluxes are not supported explicitly, however time dependence can be straightforwardly handled by providing multiple files with fluxes divided into different time bins.

Users should note that *SNOwGLoBES does not employ the oscillation probability and parameter sensitivity computation functionality of GLoBES at all.* Although SNOwGLoBES can be helpful for studies of neutrino oscillation by, for example, enabling the comparison of expected oscillated and non-oscillated fluxes (for oscillation in the supernova progenitor or Earth), the user must compute the oscillation probabilities modulating the flux and evaluate sensitivities separately. SNOwGLoBES is simply an event rate calculator.

Users familiar with version 1.2 will find all functionality they are accustomed to maintained in v. 1.3. Any tools developed for v. 1.2 will require only minimal changes to run with 1.3. Computation speed is identical between versions. However, version 1.3 extends SNOwGLoBES to support low-threshold, low-background liquid noble element dark matter detectors. Version 1.3 offers three new detector materials, flexible energy ranges and binning resolution, a wealth of new supporting code, and significant quality-of-life improvements with new visualization tools and supporting scripts.

2 Software Overview

2.1 Dependencies

SNOwGLoBES v. 1.3 depends on GLoBES [4] version 3.0.15 or later (v. 3.1.6 for Mac). The environment variable GLB_DIR must be set to the GLoBES installation directory.

GLoBES itself requires `Gnu Scientific Library` (GSL), particularly access to the header files. Users of RHEL-derivative Linux distributions such as CentOS and Scientific Linux should be aware that on these operating systems, the `GSL` header files do not install with the main `GSL` library. After installing `GSL` as normal, RHEL-derivative Linux users should install the `gsl-devel` package to obtain the header files.

`Perl` is required to execute SNOwGLoBES. The cross-section file, efficiency file, and smearing matrix generation scripts require `Python`, either 2.6, 2.7, or 3.6+ depending on the script. The most current data visualization tool requires `Jupyter Notebook`, and some additional sample plotting scripts are available utilizing `ROOT` [6].

The basic SNOwGLoBES install does *not* require `ROOT`, but advanced users wishing to utilize the CEνNS-specific extension modules in the `/smearing_code/dukecevns/` subdirectory will need `ROOT` for a majority of the functions. A README and a Makefile are available in this subdirectory with more specific instructions.

2.2   Download and Installation Instructions

- Version 1.3, the most current release, is available on Github:

  `https://github.com/SNOwGLoBES/snowglobes`

- Set the environment variable `$SNOWGLOBES` to the directory where SNOwGLoBES will be installed (most commonly `/home/[user]/snowglobes`).

- Set the environment variable `$GLB_DIR` to the location of GLoBES (most commonly `/usr/local`).

- Go to `$SNOwGLoBES/src/` and type `make`, then `make install`.

- That's it. The now fully-installed package is executed from the `$SNOWGLOBES` directory.

2.3   Files

The SNOwGLoBES package has data files organized into several subdirectories:

- `$SNOWGLOBES/backgrounds` contains the additive background files, labeled by detector configuration.

- `$SNOWGLOBES/bin` contains the GLoBES executable run by the main script.

- `$SNOWGLOBES/channels` contains the files listing relevant interaction channels for each detector type.

- `$SNOWGLOBES/doc` contains documentation for installation and running, and references for the included data files.

- `$SNOWGLOBES/effic` contains the post-smearing efficiency files in `GLoBES` format, labeled by interaction type and detector configuration. A Python 3.6+ script, written to generate these files for default detector configurations, is located in the main `$SNOWGLOBES` directory.

- `$SNOWGLOBES/fluxes` contains the flux files in `GLoBES` format, labeled by flux name.

- `$SNOWGLOBES/glb` contains some template files required for building the `GLoBES` file.

- `$SNOWGLOBES/out` contains output files generated by `SNOwGLoBES`, labeled by interaction type and detector configuration, for both interaction rates and smeared rates.

- `$SNOWGLOBES/plots` contains some sample `Root` plotting scripts. A Jupyter notebook `Plotter.ipynb`, in the main `$SNOWGLOBES` directory, provides additional visualization tools for the channel files, fluxes, smearing matrices, and outfiles.

- `$SNOWGLOBES/smear` contains the smearing matrix files in `GLoBES` format, labeled by interaction type and detector configuration. The matrices available on install use the default energy ranges and binning for their interaction type. New smearing matrices generated using the `create_smearing_matrix.py` script are saved here, but will overwrite existing matrices if the user is not careful. To save new smearing matrices without overwriting existing matrices, use the `$SNOWGLOBES/smear/new/` subdirectory.

- `$SNOWGLOBES/smearing_code` contains the `create_smearing_matrix.py` script, which generates new smearing matrices with user-defined parameters (energy ranges, binning resolution, smearing function, etc.) Also located here is the `/dukecevns/` subdirectory, which is home to the CEνNS smearing matrix generation code (and a Makefile to compile the CEνNS code independently of the main `SNOwGLoBES` library).

- `$SNOWGLOBES/xscns` contains the cross-section files in `GLoBES` format, labeled by interaction channel. Supporting scripts to generate new cross-sections files are located in this directory, in a subdirectory named `/generic_xs/`.

2.4 Pre-requisites for Running `SNOwGLoBES`

Before running `SNOwGLoBES`, make sure the following files are made and named appropriately for the detector response you wish to simulate:

- An efficiency file for **each** interaction in the channel file. (see Section 7.1)

- A cross-sections file for **each** interaction in the channel file. (see Section 8.8)

- A smearing matrix for **each** interaction in the channel file. (see Section 9.1)

126 If any of these files are missing, `SNOwGLoBES` will encounter a run error. Supporting scripts are provided for

127 the user to generate any missing files, the details of which can be found in their respective sections.

2.5   Running `SNOwGLoBES`

129 The heart of `SNOwGLoBES` is a script called `supernova.pl`. This takes four arguments: the run mode, the

130 flux file name, the channel file name, and the experiment configuration name.

131 `SNOwGLoBES` is run from the terminal by calling `supernova.pl` as an executable. The general syntax is

132 ```
./supernova.pl [run mode] [flux] [channel file] [experiment name]
```

133 For example, to compute event rates in the HALO2 detector for the "Livermore" flux and the interaction

134 channels in `$SNOWGLOBES/channels/channels_lead.dat`, the command will be

135 ```
./supernova.pl 0 livermore lead halo2
```

136 The run mode argument specifies the overall state of `SNOwGLoBES` - to maintain backwards compatibility

137 with the channels available in v. 1.2, run mode 0 specifies `supernova.pl` should use the v. 1.2 default

138 energy range and binning resolution (incident neutrino energies 0.5-100 MeV, detected energies 0.5-100

139 MeV, 400 bins in each range). Run mode 1 allows the user to set their own energy range and binning. The

140 CEνNS interaction channels *require* run mode 1 - for example, a simulation of CEνNS interactions from the

141 Livermore flux in the `LUX-ZEPLIN` detector would use the command

142 ```
./supernova.pl 1 livermore xenon_coh lz
```

143 Users should see Sec. 3 for more information.

144 The channel file gives a list of interaction types, a.k.a. "channels," for which `SNOwGLoBES` should compute

145 event rates, and the appropriate target weighting factor for each of these channels. Example channel files can

146 be found in the `$SNOWGLOBES/channels` subdirectory. The user can modify these to select desired channels

147 or, if `SNOwGLoBES` is being run in "new" mode, edit the energy range and binning. See Sec. 5 for information

148 about the channel file structure, and Sec. 10 for instructions on adding new channels.

149 As of v. 1.3, an efficiency file is required to complete a simulation. However, the supporting script

150 `create_effc.py` in the top-level `$SNOWGLOBES` directory allows the user to set whatever detection efficiency

151 they like, up to and including 100% detection efficiency at all energies. See Sec. 7 for more information.

152 Simulation outputs go to the `$SNOWGLOBES/out` subdirectory, into spectrum files named by flux, channel,

153 and experiment configuration. Output text files are generated both for interaction rates (as a function of

154 neutrino energy), and for rates smeared by detector response, as a function of detected energy (the latter

are labeled "_smeared"). The first column of the output file is energy (either incident neutrino energy in GeV or, for smeared output, detected energy in GeV), and the second column is event rate in that energy bin (events per 0.5 MeV).

Files for which event rates are not weighted by target weighting factor (see Sec. 10.2) are labeled "_unweighted". As of SNOwGLoBES version 1.1, the `supernova.pl` script applies the weighting factors to create the output files.

`make_event_table.pl` is a sample script to return integrated rates for each channel. It takes the same arguments as `supernova.pl`; add an additional argument "1" if a table of non-smeared rates is desired. As of version 1.3, all outputs can be viewed in the `Plotter.ipynb` notebook, along with the flux and smearing matrix used in the simulation. Other routines are available - the plotting scripts in the `$SNOWGLOBES/plots` directory provide `Root` examples of handling the output files.

# 3    Run Mode

The purpose of the `run mode` argument is to allow the user to run SNOwGLoBES with either the default v. 1.2 binning parameters or custom binning parameters. The 1.2 defaults are to simulate interactions with incident neutrino energies between 0.5 and 100 MeV, detected energies between 0.5 and 100 MeV, 200 true energy bins and 200 sampling bins per range. For CE$\nu$NS interaction channels, this range is unsuitable. CE$\nu$NS interaction detected energies are $< 10$ keV in xenon; using the default ranges and binning clusters all events in the lowest detected energy bin.

Custom binning, by contrast, can take on whatever values the user prefers. All energy ranges with a low threshold $> 0$ are supported. The authors recommend keeping selected energy ranges close to what is physically anticipated for an interaction channel, but this is not required. In addition, the user may choose to have different numbers of incident and detected energy bins.

To enable users to toggle back and forth between custom binning and default binning (which works well for many supported interaction channels), the `run mode` argument is called in the command line when a simulation is initiated. `supernova.pl` knows which channel file format to expect based on the value of `run mode`:

1. `run mode` = 0: default binning parameters

2. `run mode` = 1: custom binning parameters

**Note:** The user must provide channel files in the correct format for the selected run mode - see Sec. 5 for details.

## 4   Neutrino Fluxes

Two example fluxes are provided with the package: the "Livermore" flux [7], and the "GKVM" flux [8]. Strictly speaking, these are not fluxes but fluences: the flux is integrated over the time of the burst. The spectra for the flavor components of these two example fluences are plotted in Fig. 2. We will include more fluxes, and flux generation tools, in the future. The supernova neutrino fluxes are given in units of neutrinos per $cm^2$ per energy bin (bin width in GeV) for a supernova occurring 10 kpc away.



Figure 2: Left: "Livermore" fluence, for different flavor components, from reference [7], assuming Fermi-Dirac spectra with the average energies indicated as a function of time and zero chemical potential, and integrated from $t = 0$ to $t = 14$ seconds. Right: "GVKM" fluence, integrated over 10 seconds.

Also included are fluxes for solar neutrinos. All major solar neutrino components are included (from pp through hep), based on flux shapes from John Bahcall's webpage [9], and magnitudes based on the BS05 (OP) model [10]. Mixing is implemented with survival probabilities taken from a figure [11], and the $\nu_\mu$ and $\nu_\tau$ probabilities set to be equal. Units of this flux (an actual flux, in this case) are neutrinos/$cm^2$/s/0.2 MeV.

In addition to the natural neutrino fluxes, we provide simplified fluxes for debugging purposes. Six monoenergetic fluxes - $1 \times 10^{12}$ $\nu_e$ all arriving with the same incident energy - are provided for incident energies 0.01, 0.02, 0.03, 0.05, 0.075, and 0.1 GeV. These fluxes are intended to verify a given detector configuration is being simulated appropriately.

So long as the units of the flux file are correct, the user may add their own neutrino fluxes locally. These files should be added to `/snowglobes/fluxes/` with the name `[flux name].dat` - the `[flux name]` parameter is the second argument used when running `SNOwGLoBES` from the command line.

# 5 Channel Files

The channel file is a list, for a given detector, of the interaction channels SNOwGLoBES should simulate. Channel files are provided on install for most detector configurations in v. 1.3; the user can employ these files as templates to construct their own channel file.

In SNOwGLoBES v. 1.3, the selected run mode impacts the structure of the channel file. For run mode 0, the channel files are structured as in Fig. 3. The columns are

1. Interaction type

2. Interaction number in channel file (indexes from 0)

3. Neutrino parity

4. Neutrino flavor

5. Target weighting factor

If it does not find this data structure, a 0-mode simulation will terminate unsuccessfully.

In 1-mode simulations, the structure of the channel files is as given in Fig. 4. The columns are

1. General identifier for GLoBES postamble file

2. Interaction type

3. Interaction number in channel file (indexes from 0)

4. Neutrino parity

5. Neutrino flavor

6. Target weighting factor

7. Number of incident neutrino flux bins

8. Low limit of incident neutrino energy [GeV]

9. High limit of incident neutrino energy [GeV]

10. Number of detected energy bins

11. Low limit of detected neutrino energy [GeV]

12. High limit of detected neutrino energy [GeV]

For CE$\nu$NS interaction simulations, running in 1-mode is mandatory. The default energy range for 0-mode simulations is inappropriate for the energy of the recoiling nucleus in CE$\nu$NS scattering. The energy range bounds in 1-mode are not restricted, but the authors suggest remaining close to what is physically reasonable for your interactions of interest.

```
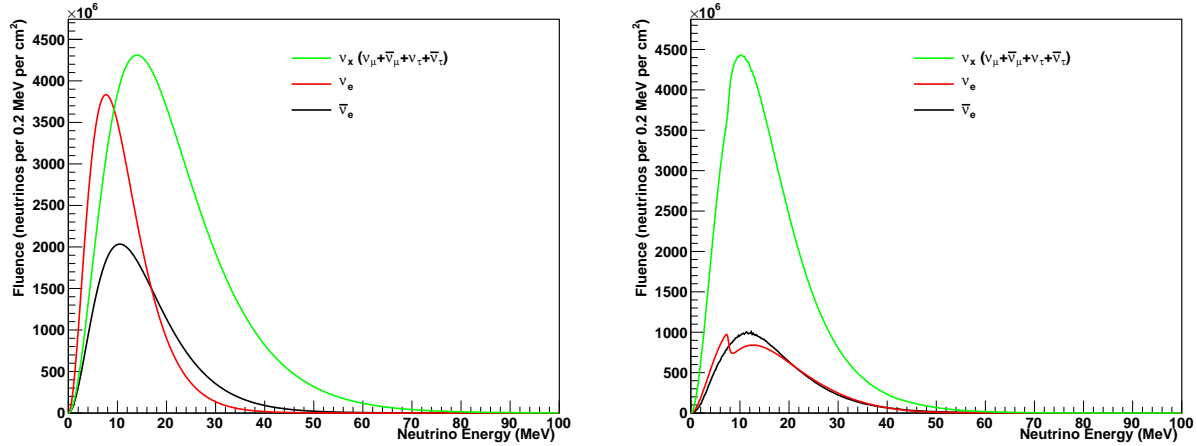nue_e 0 + e 18.
nuebar_e 1 - e 18.
numu_e 2 + m 18.
numubar_e 3 - m 18.
nutau_e 4 + t 18.
nutaubar_e 5 - t 18.
nue_Ar40 6 + e 1.
nuebar_Ar40 7 - e 1.
nc_ah_nue_Ar40 8 + e 1.
nc_ah_numu_Ar40 9 + m 1.
nc_ah_nutau_Ar40 10 + t 1.
nc_ah_nuebar_Ar40 11 - e 1.
nc_ah_numubar_Ar40 12 - m 1.
nc_ah_nutaubar_Ar40 13 - t 1.
```

Figure 3: Example of channel file with default binning

```
SN_nu nue_e 0 + e 18. 400 0.0005 .100 400 0.0005 .100
SN_nu nuebar_e 0 - e 18. 400 0.0005 .100 400 0.0005 .100
SN_nu numu_e 0 + m 18. 400 0.0005 .100 400 0.0005 .100
SN_nu numubar_e 0 - m 18. 400 0.0005 .100 400 0.0005 .100
SN_nu nutau_e 0 + t 18. 400 0.0005 .100 400 0.0005 .100
SN_nu nutaubar_e 0 - t 18. 400 0.0005 .100 400 0.0005 .100
SN_nu nue_Ar40 0 + e 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nuebar_Ar40 0 - e 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_nue_Ar40 0 + e 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_numu_Ar40 0 + m 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_nutau_Ar40 0 + t 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_nuebar_Ar40 0 - e 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_numubar_Ar40 0 - m 1. 400 0.0005 .100 400 0.0005 .100
SN_nu nc_nutaubar_Ar40 0 - t 1. 400 0.0005 .100 400 0.0005 .100
SN_nu coh_helm_Ar_nue 0 + e 1. 200 0.0005 0.1 200 0.0000000005 0.00001
SN_nu coh_helm_Ar_numu 0 + m 1. 200 0.0005 0.1 200 0.0000000005 0.00001
SN_nu coh_helm_Ar_nutau 0 + t 1. 200 0.0005 0.1 200 0.0000000005 0.00001
SN_nu coh_helm_Ar_nuebar 0 - e 1. 200 0.0005 0.1 200 0.0000000005 0.00001
SN_nu coh_helm_Ar_numubar 0 - m 1. 200 0.0005 0.1 200 0.0000000005 0.00001
SN_nu coh_helm_Ar_nutaubar 0 - t 1. 200 0.0005 0.1 200 0.0000000005 0.00001
```

Figure 4: Example of channel file with custom binning

6 Detector Configuration

The `detector_configurations.dat` file contains a list of detectors available to simulate. The first entry in each row is the detector configuration name. The second entry is the mass in kilotons. The third entry is a target normalization factor - this is reference targets per amu in the detector material. For example, for water (18 amu per molecule), the reference target is protons, and thus the target normalization factor is 2/18. For lead, the reference target is ^{208}Pb; the target normalization factor is 1/208.

17 detectors are available on install - for information about adding new configurations to SNOwGLoBES , see Sec. 10.

7 Detector Background and Efficiency

SNOwGLoBES has two mechanisms for the replication of detector effects in its simulation outputs - the detector efficiency file and the background file. The detector efficiency is a *mandatory* simulation component in v. 1.3.

The GLoBES-formatted efficiency files give the detector efficiency as a function of detected energy for a given channel and detector configuration. Generic efficiency files are provided for several detector configurations on install.

When a SNOwGLoBES simulation is initiated, supernova.pl checks the $SNOWGLOBES/effic/[experiment name] directory for an efficiency file with the name effic_[channel name]_[experiment name].dat. It must find one in order to successfully complete a simulation. To help users construct efficiency files of the correct format, we provide the script create_effc.py. The details of this script are given in Sec. 7.1.

Unlike the efficiency, including the background in a SNOwGLoBES simulation is optional. The user can provide a file of background events as a function of energy to be smeared by the indicated detector response. This is handled by creating a "fake" interaction channel and accompanying smearing file. If a GLoBES-formatted background file labeled by the detector configuration is present in the $SNOWGLOBES/backgrounds subdirectory, the background will be smeared and an additional output file created. The user is responsible for ensuring the background events in the file correspond to the same time interval as the incident neutrino signal.

7.1   Supporting Script: create_effc.py

create_effc.py is a Python 3.6+ script located in the $SNOWGLOBES main directory. It is provided to allow users to readily generate new efficiency files for their detector configuration.

create_effc.py is executed with the command python3.6 [channel file name]; it then directs the user to identify their detector configuration name. The script matches the configuration name to an efficiency function defined in get_detector_effic(). These are detection percentage as a function of detected energy in GeV, defined using lambda functions.

On install, the efficiency functions are generic - for example, create_effc.py assumes the DarkSide-20k experiment configuration has a CEνNS detection efficiency of 0% for all energies below 50 keV, and 100% detection efficiency above that threshold. The user can modify the efficiency function locally to more closely mimic their detector behavior.

New efficiency files, one per interaction channel in the channel file, are saved in a $SNOWGLOBES/effic/[detector name] subdirectory with the format effic_[interaction channel]_[detector name].dat. New files will overwrite old ones if a new savepath is not specified.

11

8 Neutrino Interaction Cross-sections

We provide cross-sections relevant for six detector materials relevant for current and near-future detectors: water, scintillator, argon, xenon, germanium, and lead. Distributions of interaction products are taken into account in the "smearing" matrices (see Sec. 9). We expect future upgrades of `SNOwGLoBES` to incorporate more channels and more materials.

Figure 5: Cross sections for relevant processes in water.

8.1 Inverse Beta Decay

Inverse beta decay $\bar{\nu}_e + p \to e^+ + n$ (IBD) is dominant for detectors with free protons, such as water and scintillator. We use the cross-section from reference [12], plotted in Fig. 5.

8.2 Neutrino-Electron Elastic Scattering

The cross-sections for elastic scattering (ES) of neutrinos on electrons $\nu_{e,x} + e^- \to \nu_{e,x} + e^-$ (both NC and CC) are known to better than percent level [13]. These cross-sections are plotted in Fig. 5. The electron ES interaction is relevant for all targets, although the scattered electrons may not be observable for some detector configurations (*e.g.* HALO).

Figure 6: Cross sections for relevant processes in scintillator.

8.3 Interactions with Oxygen

Interactions on oxygen include the CC interactions $\nu_e + {}^{16}O \rightarrow e^- + {}^{16}F$, $\bar{\nu}_e + {}^{16}O \rightarrow e^+ + {}^{16}N$. These interactions have diverse final states, including ejected nucleons and de-excitation gammas in addition to the produced lepton. For the NC interaction with ${}^{16}O$, $\nu_x + {}^{16}O \rightarrow \nu_x + {}^{16}O^*$, de-excitation gammas are in principle observable. We use cross-sections from reference [14]. Oxygen cross-sections are plotted in Fig. 5.

8.4 Interactions with Carbon

Electron flavor neutrinos will interact with carbon nuclei via the CC interactions $\nu_e + {}^{12}C \rightarrow {}^{12}N + e^-$ and $\bar{\nu}_e + {}^{12}C \rightarrow {}^{12}B + e^+$. An NC excitation interaction, $\nu_x + {}^{12}C \rightarrow \nu_x + {}^{12}C^*$ also takes place; this interaction results in a 15.1 MeV de-excitation γ-ray which can be used to tag this interaction. We use cross-sections from reference [15] for the CC interactions and the measurement from reference [16] for the NC cross-section. Carbon cross-sections are plotted in Fig. 6.

Figure 7: Cross sections for relevant processes in liquid argon.

8.5 Charged and Inelastic Neutral Current Interactions with Argon

We include the CC interactions $\nu_e + ^{40}\text{Ar} \rightarrow e^- + ^{40}\text{K}^*$, and $\bar{\nu}_e + ^{40}\text{Ar} \rightarrow e^+ + ^{40}\text{Cl}^*$. The cross sections for interactions in argon, from references [17, 18], are shown in Fig. 7. The uncertainties for the recent calculations are at around the 10-20% level. We have recently added the NC inelastic $\nu_x + ^{40}\text{Ar} \rightarrow \nu_x + ^{40}\text{Ar}^*$ channel, based on private communications with Dr. Anna Hayes. This interaction is assumed to excite the ^{40}Ar nucleus to the 9.8 MeV level, from which it will de-excite directly to ground, emitting a 9.8 MeV γ. The cross sections proved by Dr. Hayes only go up to 52 MeV, so a 2nd order polynomial fit was performed to the provided cross section curve to extrapolate this up to 100 MeV. The fit was excellent (save the lowest energy point, which was manually fixed), so the extrapolation is likely quite valid.

8.6 Interactions with Lead

For lead, we include CC and NC cross-sections for both single and double neutron ejection channels for:
$\nu_e + ^{208}\text{Pb} \rightarrow e^- + ^{208}\text{Bi}^*$, $\nu_x + ^{208}\text{Pb} \rightarrow \nu_x + ^{208}\text{Pb}^*$, $\bar{\nu}_x + ^{208}\text{Pb} \rightarrow \bar{\nu}_x + ^{208}\text{Pb}^*$.

We use cross-sections from [19]. Lead cross-sections are plotted in Fig. 8. Uncertainties on lead cross-sections are evaluated in [20].

Figure 8: Cross sections for relevant processes in lead.

8.7 Coherent Elastic Neutrino-Nucleus Scattering Interactions with Argon, Xenon, and Germanium

The CEνNS interaction channel is new to v. 1.3. First detected by the `COHERENT` collaboration in 2008 [21], CEνNS is a weak neutral current interaction which, in liquid xenon, causes interacting nuclei to recoil with $\mathcal{O}(1)$ keV of energy. CEνNS is flavor-agnostic - detectors sensitive to CEνNS are equally sensitive to all neutrino flavors. CEνNS has the highest cross-section of interaction of all neutrino scattering processes.

The CEνNS cross-section of interaction is given by [22] as:

$$\frac{d\sigma}{dE_R} = \frac{G_F^2}{2\pi} M \left[2 - \frac{2E_R}{E_I} + \frac{E_R^2}{E_I^2} - \frac{ME_R}{E_I^2} \right] \frac{Q_W^2}{4} \mathcal{F}^2(Q) \times (\hbar c)^{-4} \tag{1}$$

where G_F is the Fermi coupling constant, M the nucleus mass, the weak neutral factor is $Q_W = N - (1 - 4sin(\theta_W))Z$, and $\mathcal{F}(Q)$ is the nuclear form factor as a function of momentum transfer. The term with the most uncertainty in this expression is the form factor $\mathcal{F}(Q)$, which describes the distribution of nucleons in the target nucleus. A number of expressions for $\mathcal{F}(Q)$ are in use - for example, low-threshold, low-background dark matter experiments tend to prefer the hard-sphere based Helm form factor [23, 24]:

15

$$F(Q) = 3J_1(Qr_n)\frac{e^{-\frac{(Qs)^2}{2}}}{Qr_n}, \tag{2}$$

320 while the COHERENT collaboration utilizes the Klein-Nystrand form factor [25]:

$$F(Q) = 3\frac{J_1(QR_A)}{QR_A}\frac{1}{1+Q^2a_k^2}. \tag{3}$$

321 There is no clear consensus which form factor is preferred. To maintain consistency with the internal tools

322 of the most likely users of SNOwGLoBES - low-threshold, low-background dark matter experiments - we have

323 chosen to provide default cross-sections files for natural abundance argon, xenon, and germanium detectors

324 assuming the Helm form factor. As we will explore in Sec. 8.8, the user has the option to change the form

325 factor and abundance locally using a provided cross-section file generation script.

8.8 Supporting Scripts: Generation of Cross-Sections Files

327 In the subdirectory $SNOWGLOBES/xscns/generic_xs, we provide three scripts to generate new cross-sections

328 files. The first, compute_nu_e_cross_sections.py, runs with either Python 2.6 and 3.7. It requires no

329 arguments, and generates new cross-sections files for $\nu_{e,x} + e^- \rightarrow \nu_{e,x} + e^-$ ES interactions as detailed in

330 Sec. 8.2. This script is most useful in the event a $\nu_{e,x} + e^- \rightarrow \nu_{e,x} + e^-$ ES cross-section file is accidentally

331 deleted, but an experienced user could edit it to change how the cross-section of interaction is calculated.

332 The CEνNS cross-sections have a number of parameters the user may wish to change - to that end, we pro-

333 vide populate_CEvNS_xs.py. This script is executed using the command python3.6 populate_CEvNS_xs.py

334 [form factor name]. Allowable form factor names are Helm and Klein-Nystrand, but space has been des-

335 ignated for future implementation of the Horowitz/numerical and Hoferichter Form Factors.

336 By editing the script, the user can change the relative abundance of their detector material. Lines 161,

337 166, and 171 are lists of relative abundance weights for argon, germanium, and xenon respectively. The

338 weights are arranged from lightest to heaviest isotope. By editing this array, the user can set the relative

339 abundance of their detector material however they like, but care should be taken to ensure the sum of all

340 weights is unity to avoid normalization errors. If the relative abundance is changed here, it must also be

341 changed in the smearing matrices - for more information on that process, see Sec. 10.6.1.

342 populate_CEvNS_xs.py generates 18 files by default - one cross-section file for each neutrino flavor and

343 element - but the user can limit the script to their material of interest by commenting out the undesirable

344 Generate() calls at the end of the script. To limit the script to a smaller subset of neutrino species, edit

345 the neutrinoname list on line 142.

16

The final script provided in `$SNOWGLOBES/xscns/generic_xs` is `universal_xs.py`. This script will run with either Python 2.7 or 3.7; running the script without arguments will print instructions and produce an example file. This script is designed to assist in adding new interaction channels by producing "generic" cross-sections files for charged-current interactions on arbitrary nuclei. These generic cross-sections are based on parameterizations of cross-section shape and magnitude. The magnitude parameterization was made by fitting cross sections for various materials provided in [26]. The parameterization takes into account the Z (proton number), A (mass number), and Q-value threshold of the reaction. This parameterization matches the theoretical cross section predictions to within near 20%, though the theoretical predictions themselves may be far more uncertain than that.

The cross section parameterization is based on the cross sections provided in SNOwGLoBES for deuterium, ^{12}C, ^{16}O, and ^{40}Ar, based on Q-value threshold and mass number. Note that the thresholds used in this parameterization fit may not all be accurate. While the ground-state to ground-state mass difference (plus electron mass) can be a good estimate for the threshold, the reaction may tend to go to an excited state of the daughter nucleus.

`universal_xs.py` comes pre-tuned, although the scripts used for parameterization tuning are provided in `$SNOWGLOBES/xscns/generic_xs/parameterization` if the user wishes to use their own cross sections as input for a new parameterization. Cross-sections files produced from this script should not be assumed to be accurate. It is the authors' hope these cross sections are reasonably close (within 50% or so) to true cross sections, but the model is fairly basic, and the estimates may be off by factors of 5 or more. The cross-sections files from this script should be deployed with caution.

9   Smearing Matrices

SNOwGLoBES uses the smearing matrices to convert incident neutrino energies from the flux file to energies deposited in the detector in the simulation results. These are $i \times j$ comma-delimited matrices, where each matrix entry corresponds to the probability of some incident neutrino energy j causing an interaction in the detector with energy i. The probability is convolved with some user-defined smearing function, which mimics reconstruction inaccuracies due to detector effects. When a run of SNOwGLoBES is initiated, `supernova.pl` searches for a smearing matrix matching the selected detector name, verifies its existence, and then begins the run.

Default smearing matrices are provided on install for most of the default detector configurations, but the user should expect to generate new ones using the provided supporting script.

9.1 Supporting Script: `create_smearing_matrix.py`

The script `create_smearing_matrix.py` is provided to generate new smearing matrices according to the user's needs. This script requires Python 3.6 or higher. In the `$SNOWGLOBES/smearing_code` directory, run `python3.6 create_smearing_matrix.py` and follow the interactive prompts, which explain the inputs in detail. The script can also be run with a JSON file as the sole argument.

The script can produce smearing matrices for any of the detector configurations provided on install and the following detector materials:

^{16}O, ^{12}C, natural abundance Ar, ^{40}Ar, natural abundance Ge, natural abundance Xe,

single-neutron ^{208}Pb, double-neutron ^{208}Pb

Four interaction mechanisms - neutrino-electron elastic scattering, neutrino-nucleus charged-current scattering, neutrino-nucleus neutral-current scattering, and CEνNS - are available. It is incumbent on the user to select the appropriate material and interaction mechanism for a given detector type - the script does not prevent the user from generating a smearing matrix which is not correct for their detector configuration. The natural abundance Ar, Xe, and Ge are intended for CEνNS interaction simulations, and it is possible to modify the relative abundances of these materials - see Sec. 10.6.1 for details. The ^{40}Ar material is optimized for the neutrino-nucleus charged-current and neutral-current interactions, but *can* be used for CEνNS with some modifications to the cross-sections file (see Sec. 10.6.1).

Also settable by the user are the energy range of incident neutrinos, energy range of events in detector, and number of bins in each range, a new feature driven by the addition of the CEνNS interaction channel to v. 1.3. Be aware that when running `create_smearing_matrix.py` the script attempts to match the user-selected energy ranges and binning values to the equivalent parameters in the channel file. The channel file should be edited before `create_smearing_matrix.py` is run.

New smearing matrices generated by `create_smearing_matrix.py` save, by default, into `$SNOWGLOBES/smear` where SNOwGLoBES can find them. However, the default smearing matrices provided on install are stored in this directory, and new smearing matrices will overwrite old matrices of the same name. To prevent the default matrices from being overwritten, the user can change the savepath in `create_smearing_matrix.py` - `$SNOWGLOBES/smear/new` is recommended for this purpose.

9.2 CEνNS Smearing Matrices and the `DukeCE`ν`NS` Module

In the smearing matrix, the probability of an incident neutrino depositing some energy in the detector depends on the differential cross-section of interaction $\frac{d\sigma}{dE_R}$ (see Sec. 8.7 for discussion of the CEνNS $\frac{d\sigma}{dE_R}$ expression). $\frac{d\sigma}{dE_R}$ must be separately calculated for each entry in the smearing matrix; the CEνNS smearing

matrices, for the binning suggested on install, are 200×1600 elements. This computation is entirely too expensive for `create_smearing_matrix.py` to handle alone. We have chosen to integrate a new submodule, DukeCEνNS, as supporting code for the CEνNS smearing matrices.

DukeCEνNS is a C++ simulation package. It is capable of rapidly calculating the CEνNS differential cross-section of interaction for argon, xenon, and germanium, supports user-settable relative abundances, and has three form factors widely in use in the literature - the Helm [23], Klein-Nystrand [25], and Horowitz (numerical, data obtained via personal communication) form factors - available for the differential cross-section calculation. DukeCEνNS is also extensible in ways `create_smearing_matrix.py` is not. Available as of 2020 is a form factor based on chiral effective field theory, published by Hoferichter et al. [27]; this form factor may capture the structure of target nuclei more accurately than the hard-sphere based Helm or Klein-Nystrand form factors, which will be important for planned comparative studies. The Hoferichter form factor can readily be added to DukeCEνNS by future developers.

In DukeCEνNS, the module `cevns_recoil_response.cc` takes in the detector material (argon, xenon, or germanium, default is natural abundance), desired form factor (default is the Helm form factor), incident neutrino energy range, number of incident energy bins, recoil energy range, and number of recoil bins as inputs. It returns a whitespace-delimited $i \times j$ "response matrix," where each entry is $\frac{d\sigma}{dE_R}$ for recoil energy i and incident neutrino energy j. These response matrices are stored as `.matrix` files in `$SNOWGLOBES/smearing_code/dukecevns/recoil_response_matrices`.

A flowchart of how CEνNS smearing matrices are built is shown in Fig. 9 - the user interacts exclusively with `create_smearing_matrix.py`. Following the prompts on the terminal, the user selects their detector material, form factor, binning parameters, and the detector resolution function. `create_smearing_matrix.py` takes these inputs and searches the `/recoil_response_matrices/` subdirectory for a response matrix matching the inputs. If it finds one, it reads the file in, transforms the $\frac{d\sigma}{dE_R}$ entries to probabilities, applies the detector resolution function, and saves the resultant smearing matrix to an outfile in `$SNOWGLOBES/smear` with the proper format. If it does not find an unsmeared response `.matrix` file, it directs `cevns_recoil_response` to create one, then performs the above steps on the newly-generated response file.

Saving the unsmeared response matrices is a small sacrifice of space - each `.matrix` file is ~ 3 Mb in size - which massively improves the speed at which CEνNS smearing matrices are generated. It is occasionally necessary to alter the detector resolution function of an existing smearing matrix - with the unsmeared response matrix saved, `create_smearing_matrix.py` does not have to waste time building a new file from scratch for this change.

Figure 9: A flowchart of how the CEνNS smearing matrices are made. The user should never need to interact with any of the DukeCEνNS code; all interactions are done using the `subprocess` module in `create_smearing_matrix.py`.

10 Adding New Stuff

It is straightforward to add new fluxes, channels, targets or detector configurations to SNOwGLoBES.

If you add new data files or configurations, the SNOwGLoBES team would appreciate if you would contact us so that the new information can be made available to all users.

10.1 Adding New Fluxes

Fluxes can be added by preparing a file according to the specifications in the GLoBES manual. The flux file must be placed in the `$SNOWGLOBES/fluxes` directory and named `[flux name].dat`. The file should have 501 rows with flux (or fluence) per 0.2 MeV for the different flavors in each row, for neutrino energies ranging from 0.0001 to 0.1001 GeV.

10.2 Adding New Channels

A new channel for a particular material can be added as a new row in the channel file. Each channel needs a name. Any text may be used to identify the channel, but we use a channel naming convention as follows:

- Channels are called `[neutrino flavor]_[target name]`, where `[neutrino flavor]` is `nue`, `numu`, `nutau`, `nuebar`, `numubar`, or `nutaubar`. If the neutrino target in the detector is a single nuclear isotope, it's written as the element abbreviation followed by the mass number A, e.g. ^{12}C is rendered C12. An element abbreviation *not* followed by a mass number, e.g. Xe, is assumed to be a mixture

of isotopes. For example, the charged current interaction of ν_e with ^{16}O will have the channel name
`nue_O16`.

- For inelastic neutral current channels, the channel name is preceded by `nc_`. For CEνNS channels, the channel name is preceded by `coh_[form factor name]_`, where `[form factor name]` should be `helm, klein-nystrand`, or `horowitz`. Note that although neutral current channels (both elastic and inelastic) are flavor-independent, each incoming neutrino flavor must be listed as a separate channel.

- Specific exclusive final states, such as particular excitations or ejecta, are denoted with a final tag. For example, the neutral current interaction of $\bar{\nu}_\mu$ on ^{208}Pb producing double-neutron final states has the channel name `nc_numubar_Pb208_2n`.

- The $\bar{\nu}_e + p \to e^+ + n$ inverse beta decay channel is an exception to these conventions, and is simply denoted `ibd`.

The format of the channel file is explained in detail in Sec. 5 - note that two different formatting conventions are in use, depending on whether or not the user wishes to use the default energy ranges and binning parameters.

The target weighting factor - parameter 5 in the channel files for both SNOwGLoBES run modes - requires a bit more explanation; for each material, one selects one target "reference" type, for which the weighting factor is unity. The other targets in the channels file have target weighting factors indicating the relative number of targets per reference target. For example, for water, we have chosen protons to be the reference target, and inverse beta decay has weighting factor 1. There are 0.5 ^{16}O nuclei for each proton, and 5 electrons for each proton in water, so the target weighting factors for elastic scattering on electrons are 5, and for CC and NC interactions on oxygen are 0.5. CEνNS interactions have a target weighting factor of 1 - the target of a CEνNS interaction is the nucleus, of which there is 1 per atom.

If you add any new channels to the file, you must provide cross-sections for these channels and put them in the `$SNOWGLOBES/xscns` directory.

10.3   Adding New Detector Configurations

Each new detector configuration must have smearing and efficiency files associated with it for *each* channel the user wishes to calculate rates for. The smearing and post-smearing efficiency files must be prepared according to the specifications in the GLoBES manual. In 0 mode, SNOwGLoBES uses 200 true energy bins from 0.0005 to 0.100 GeV and 200 sampling bins over the same range. In 1 mode, the bin numbers and energy range can take on any value desired by the user - to prepare the smearing matrix and efficiency files in

21

484 1 mode, first edit the channel file to have the desired energy range and binning parameters. The scripts

485 `create_effc.py` and `create_smearing_matrix.py` will query the channel file for the required energy range

486 and binning, then format the new files appropriately.

487 Smearing matrices must go in the `$SNOWGLOBES/smear` directory and be named as follows: `smear_[channel`

488 `name]_[detector configuration name].dat`. Efficiency files must go in the `$SNOWGLOBES/effic/[detector`

489 `configuration name]` directory and be named as follows: `effic_[channel name]_[detector configuration`

490 `name].dat`.

10.4   Adding A New Detector Material

492 To add an entirely new detector material, the user must create a new channel file with interaction channels

493 for that material, along with appropriate cross-section files (and a detector configuration employing that

494 material, with requisite efficiency file and smearing matrix).

10.5   Adding A New Background

496 The background calculated by `SNOwGLoBES` is presumed to be the sum of all relevant backgrounds in the

497 appropriate time interval, as provided by the user. Backgrounds can be be added by creating a file in the

498 `backgrounds` subdirectory called `bg_[background channel]_[detector configuration name].dat`. If no

499 file is present in this directory for a given detector configuration, `SNOwGLoBES` will ignore the background.

500 An accompanying smearing file must be provided by the user, named `smear_[channel name]_[detector`

501 `configuration name].dat`. If the background is already smeared, this can be the unit matrix.

10.6   CEνNS Parameters

503 CEνNS interactions have a few additional degrees of freedom relative to inelastic NC channels, and can be

504 modified as follows.

10.6.1   Relative Detector Abundance

506 The primary detector materials for CEνNS-sensitive detectors, and the ones supported in v. 1.3 on install,

507 are argon, xenon, and germanium. By default, `SNOwGLoBES` assumes all three detector materials have natural

508 relative abundances of their constituent isotopes. This is, however, not necessarily true of all future CEνNS-

509 sensitive detectors. Detectors primarily designed to look for neutrinoless double-beta decay, for example, may

510 have abundances which are far from natural, but they are still capable of potentially detecting a supernova

22

neutrino signal. With an eye to supporting these detectors, we allow the user to specify the isotopic relative abundance of their detector configuration.

In addition to providing a cross-section file with the desired relative abundance (the process of generating this file is detailed in Sec. 8.8) the user will need to provide an updated smearing matrix.

The CEνNS smearing matrix code is a module called DukeCEνNS, located in the $SNOWGLOBES/smearing_code/dukecevns directory - the directory structure of the important DukeCEνNS files is shown in Fig. 10. The relative abundance is defined in the mixtures.h file. In mixtures.h, isotope abundances are defined using the general syntax

```
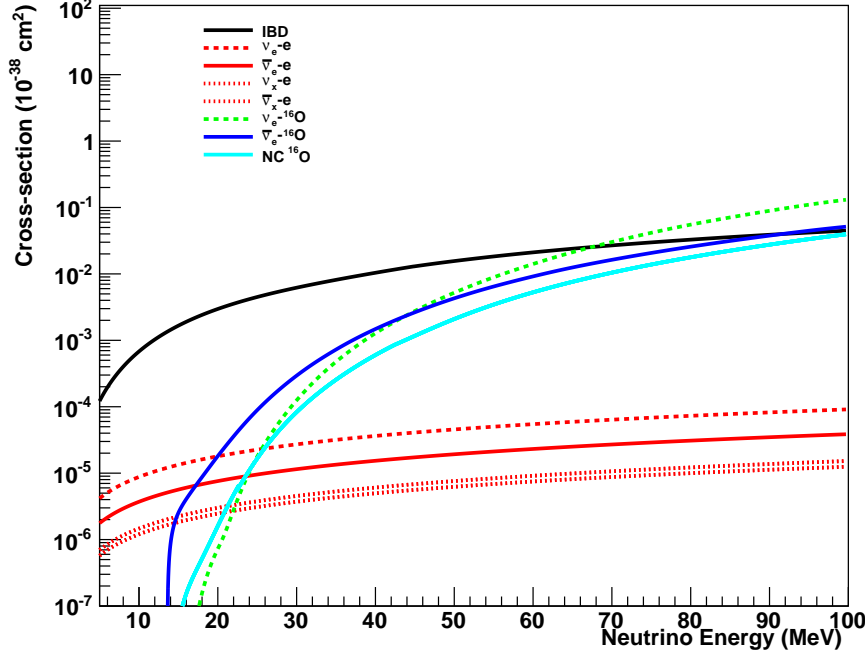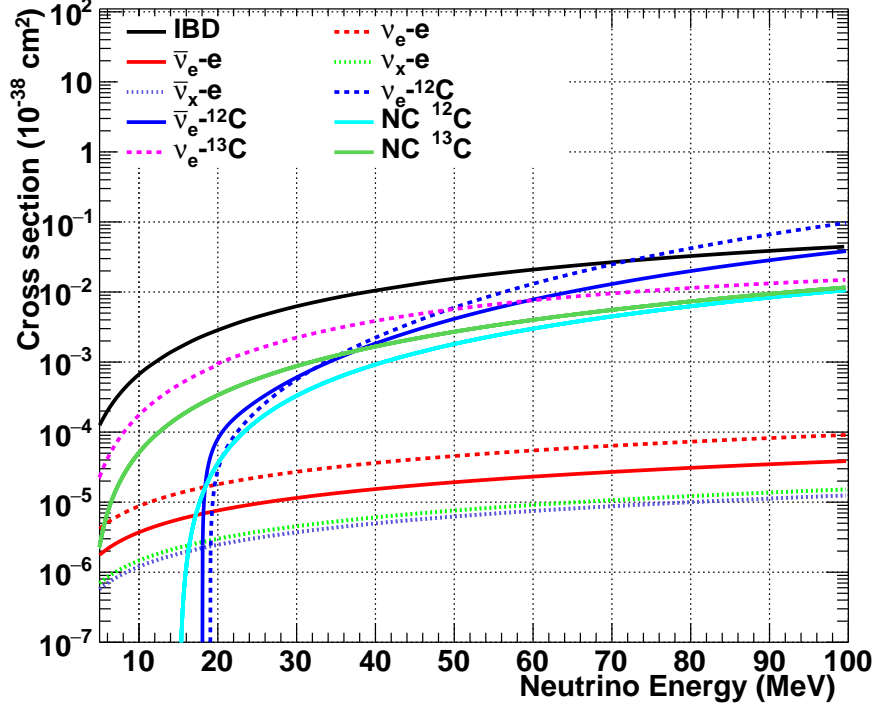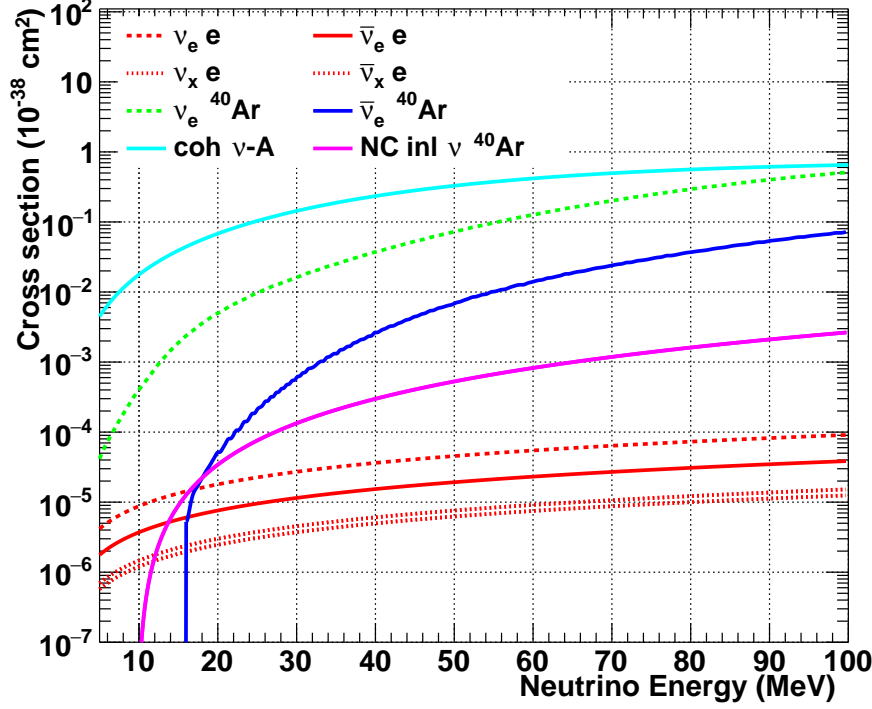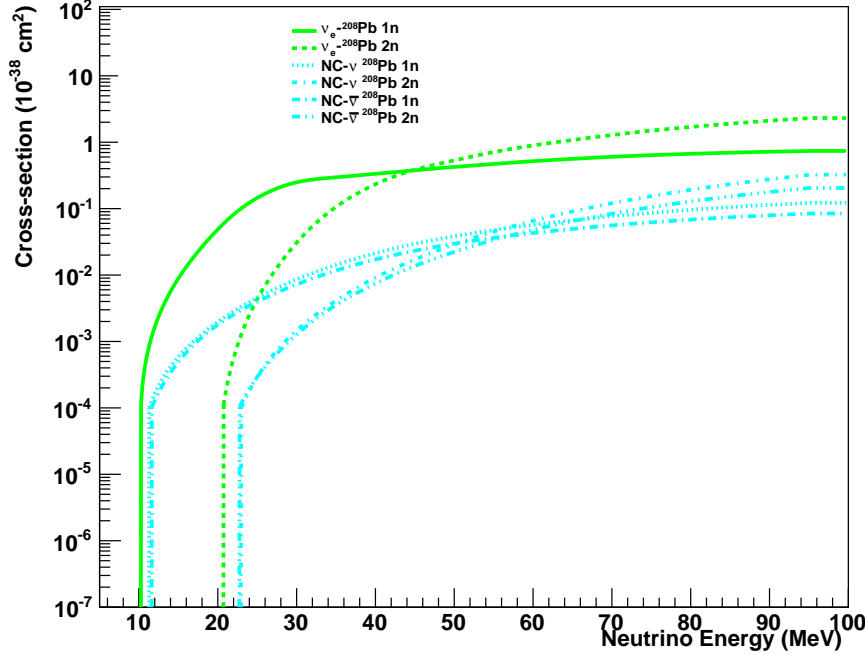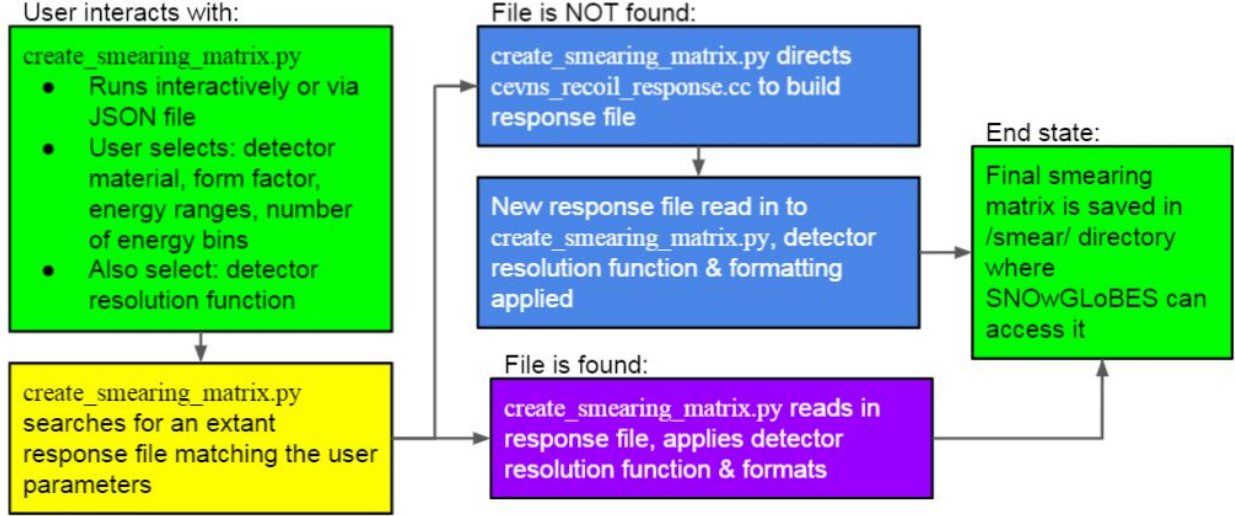isotopes["element name"].push_back["isotope"];
molar_fraction["element name"].push_back(isotope abundance);
```

where the molar fraction being pushed back is the relative abundance of the isotope on the preceding line. Up to four digits of precision are supported for the isotope abundance parameter.

To create, for example, a germanium detector with a 50% ^{70}Ge and 50% ^{73}Ge composition, locate the block where the molecular fractions of the germanium isotopes are defined. Set the molar fractions to zero - molar_fraction["Ge"].push_back(0) - for all isotopes *except* ^{70}Ge and ^{73}Ge. The molar fraction of ^{70}Ge and ^{73}Ge should be set to 0.5. The code block will look like

```
isotopes["Ge"].push_back("Ge70");
molar_fraction["Ge"].push_back(0.5);
isotopes["Ge"].push_back("Ge72");
molar_fraction["Ge"].push_back(0);
isotopes["Ge"].push_back("Ge73");
molar_fraction["Ge"].push_back(0.5);
isotopes["Ge"].push_back("Ge74");
molar_fraction["Ge"].push_back(0);
isotopes["Ge"].push_back("Ge76");
molar_fraction["Ge"].push_back(0);
```

when complete. Save the edited mixtures.h and, in $SNOWGLOBES/smearing_code/dukecevns, type make clean, then make. The DukeCEνNS code will rebuild, and the new abundance will be applied to any germanium smearing matrix generated from that point on. Take care that molar fractions of the new relative abundance sum to 1 - otherwise, you will encounter normalization errors in the smearing matrix.

Astute users will note there are some monoisotopic detector materials already defined in mixtures.h.

To use any of these pre-defined monoisotopic materials, edit line 296 of `create_smearing_matrix.py`, the `GetTargetName()` function, to add the name of the desired monoisotopic detector material to the dictionary of targets. You can then run `create_smearing_matrix.py` as normal to generate a monoisotopic smearing matrix. New monoisotopic detectors are added by editing `materials.h` to have the material defined rebuilding DukeCEνNS with `make clean` and `make`, and finally adding it as a target to `create_smearing_matrix.py`. Note that the cross-sections file must be edited accordingly in all cases.

Materials beyond argon, xenon, and germanium are available in `mixtures.h` for future implementation, if desired.

Figure 10: The general directory structure of the code used to generate CEνNS smearing matrices, including DukeCEνNS. Directories and subdirectories are shown in blue, static files are white, and executables are shown in green.

10.6.2 Form Factors

New form factors are added to the DukeCEνNS code by first editing `FormFactors.h` to define the form factor name. Form factors can be provided as closed mathematical expressions (i.e. the Helm and Klein-Nystrand form factors), as numerical parameterization files (i.e. the Horowitz/numerical form factor), or, in the future, as independent code modules which can be imported into DukeCEνNS. The authors recommend this latter approach for the chiral EFT/Hoferichter form factor.

Once the form factor namespace is defined in `FormFactors.h`, the details of the form factor are defined in `FormFactors.cc`. Finally, the new form factor must be added as an option in `cevns_recoil_response.cc`

- space is designated specifically for the Hoferichter Form Factor on line 313, but other form factors can be added in the same place.

Once all files have been edited and saved, the DukeCEνNS code should be rebuilt by typing `make clean`, then `make`. Finally, `create_smearing_matrix.py` should be edited to add the new form factor as an option on line 340 in the function `GetFormFactorName()`.

10.6.3 Advanced CEνNS Modules

By default, `SNOwGLoBES` v. 1.3 utilizes only one of several modules available in DukeCEνNS. Located in the same subdirectory are computation routines for simplified supernova neutrino fluxes, CEνNS interaction rates, and detector responses to simulated events. These are not necessary to run v. 1.3 and, unlike `cevns_recoil_response.cc`, do not compile on install.

A user wishing to develop extended functionality with these routines should utilize the Makefile located in `/smearing_code/dukecevns/` to compile the modules independent of `SNOwGLoBES`. Be aware that the vast majority of these advanced modules require `ROOT` to compile and run. The `ROOTSYS` and `PATH` variables should be set to export in the system `bashrc` file (or equivalent for non-Linux users) before attempting to compile.

11   Future Upgrades

Several future upgrades are planned for `SNOwGLoBES`:

- Inclusion of more targets, channels and fluxes.

- As knowledge of detector responses improves (for instance as detector simulation codes improve, and measurements are made) we will update the smearing matrices accordingly.

- Computation of angular distributions for channels with asymmetries.

- Inclusion of additional cross-sections to allow evaluation of cross-section-related uncertainties.

- Addition of frameworks for treatment of time-dependent fluxes, and for treating experimental and theoretical uncertainties.

- Expansion of capability to include multiple, distinct backgrounds.

- Inclusion of non-supernova fluxes, *e.g.* stopped-pion fluxes.

- Additional form factors for CEνNS channels.

Bug reports, suggestions and contributions are very welcome. Please contact Kate Scholberg at `schol@phy.duke.edu`.

12 Acknowledgements

Appendix A   Change Log

- Version 1.0 is the initial release.

- Version 1.1: The target weighting factors are applied automatically by `supernova.pl`. which creates correctly-weighted output event rate files from the raw unweighted files (the latter now designated unweighted.dat). Example plotting and event table scripts now no longer apply the weighting factor to event rates, and the user is no longer responsible for doing it.

- Version 1.2: Capacity for handling backgrounds is added; a single background rate per detector configuration is enabled, if provided by the user. A default background for the `ar17kt` configuration is included. The NC inelastic interaction cross section is added for ^{40}Ar.

- Version 1.3: The CEνNS interaction channel is added for natural abundances of Ar, Xe, and Ge detector materials, with instructions for the user to change the abundance. Detectors utilizing CEνNS, CEνNS cross-sections files, and new supporting code are added. User-settable energy ranges and binning resolution is now supported. The efficiency file is now a mandatory simulation component. New output visualization tools are included.

Appendix B   GPL

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see `https://www.gnu.org/licenses/`.

References

[1] Amol Dighe. Physics potential of future supernova neutrino observations. *J. Phys. Conf. Ser.*, 136:022041, 2008.

[2] Kate Scholberg. Supernova neutrino detection. 2007.

[3] Patrick Huber, M. Lindner, and W. Winter. Simulation of long-baseline neutrino oscillation experiments with GLoBES. *Comput. Phys. Commun.*, 167:195, 2005.

[4] http://www.mpi-hd.mpg.de/lin/globes/.

[5] M. Bass et al. A Study of the Physics Potential of the Long-Baseline Neutrino Experiment Project with an Extensive Set of Beam, Near Detector and Far Detector Configurations. 2010.

[6] http://root.cern.ch/drupal/.

[7] T. Totani, K. Sato, H. E. Dalhed, and J. R. Wilson. Future detection of supernova neutrino burst and explosion mechanism. *Astrophys. J.*, 496:216–225, 1998.

[8] Jerome Gava, James Kneller, Cristina Volpe, and G. C. McLaughlin. A dynamical collective calculation of supernova neutrino signals. *Phys. Rev. Lett.*, 103:071101, 2009.

[9] http://www.sns.ias.edu/ jnb/SNdata/sndata.html.

[10] John N. Bahcall, Aldo M. Serenelli, and Sarbani Basu. New solar opacities, abundances, helioseismology, and neutrino fluxes. *Astrophys. J.*, 621:L85–L88, 2005.

[11] Jeffrey M. Berryman, Andre de Gouvea, and Daniel Hernandez. Solar Neutrinos and the Decaying Neutrino Hypothesis. *Phys. Rev.*, D92(7):073003, 2015.

[12] Alessandro Strumia and Francesco Vissani. Precise quasielastic neutrino nucleon cross section. *Phys. Lett.*, B564:42–54, 2003.

[13] William J. Marciano and Zohreh Parsa. Neutrino-Electron Scattering Theory. *J. Phys.*, G29:2629–2645, 2003.

[14] E. Kolbe, K. Langanke, and P. Vogel. Estimates of weak and electromagnetic nuclear decay signatures for neutrino reactions in Super-Kamiokande. *Phys. Rev.*, D66:013007, 2002.

[15] E. Kolbe, K. Langanke, and P. Vogel. Weak reactions on 12C within the Continuum Random Phase Approximation with partial occupancies. *Nucl. Phys.*, A652:91–100, 1999.

[16] B. Armbruster et al. Measurement of the weak neutral current excitation C- 12(nu(mu) nu'(mu))C*-12(1+,1,15.1-MeV) at E(nu(mu)) = 29.8- MeV. *Phys. Lett.*, B423:15–20, 1998.

[17] Ines Gil Botella and A. Rubbia. Decoupling supernova and neutrino oscillation physics with LAr TPC detectors. *JCAP*, 0408:001, 2004.

[18] E. Kolbe, K. Langanke, G. Martinez-Pinedo, and P. Vogel. Neutrino nucleus reactions and nuclear structure. *J. Phys.*, G29:2569–2596, 2003.

[19] J. Engel, G. C. McLaughlin, and C. Volpe. What can be learned with a lead-based supernova-neutrino detector? *Phys. Rev.*, D67:013005, 2003.

[20] N. Paar, T. Suzuki, M. Honma, T. Marketin, and D. Vretenar. Uncertainties in modeling low-energy neutrino induced reactions on iron group nuclei. 2011.

[21] A.A. Aguilar-Arevalo, C.E. Anderson, A.O. Bazarko, S.J. Brice, B.C. Brown, L. Bugel, J. Cao, L. Coney, J.M. Conrad, D.C. Cox, A. Curioni, Z. Djurcic, D.A. Finley, B.T. Fleming, R. Ford, F.G. Garcia, G.T. Garvey, C. Green, J.A. Green, T.L. Hart, E. Hawker, R. Imlay, R.A. Johnson, G. Karagiori, P. Kasper, T. Katori, T. Kobilarcik, I. Kourbanis, S. Koutsoliotas, E.M. Laird, S.K. Linden, J.M. Link, Y. Liu, Y. Liu, W.C. Louis, K.B.M. Mahn, W. Marsh, P.S. Martin, G. McGregor, W. Metcalf, P.D. Meyers, F. Mills, G.B. Mills, J. Monroe, C.D. Moore, R.H. Nelson, V.T. Nguyen, P. Nienaber, J.A. Nowak, S. Ouedraogo, R.B. Patterson, D. Perevalov, C.C. Polly, E. Prebys, J.L. Raaf, H. Ray, B.P. Roe, A.D. Russell, V. Sandberg, R. Schirato, D. Schmitz, M.H. Shaevitz, F.C. Shoemaker, D. Smith, M. Soderberg, M. Sorel, P. Spentzouris, I. Stancu, R.J. Stefanski, M. Sung, H.A. Tanaka, R. Tayloe, M. Tzanov, R. Van

667   de Water, M.O. Wascko, D.H. White, M.J. Wilking, H.J. Yang, G.P. Zeller, and E.D. Zimmerman. First

668   observation of coherent Ï0 production in neutrinoânucleus interactions with eÎœ¡2 gev. *Physics Letters*

669   *B*, 664(1):41–46, 2008.

670   [22] Kelly Patton, Jonathan Engel, Gail C. McLaughlin, and Nicolas Schunck. Neutrino-nucleus coherent

671   scattering as a probe of neutron density distributions. *Phys. Rev. C*, 86(2):024612, August 2012. arXiv:

672   1207.0693.

673   [23] J. D. Lewin and P. F. Smith. Review of mathematics, numerical factors, and corrections for dark matter

674   experiments based on elastic nuclear recoil. *Astroparticle Physics*, 6(1):87–112, December 1996.

675   [24] D. S. Akerib, C. W. Akerlof, D. Yu. Akimov, A. Alquahtani, S. K. Alsum, T. J. Anderson, N. Angelides,

676   H. M. Araújo, A. Arbuckle, J. E. Armstrong, M. Arthurs, H. Auyeung, X. Bai, A. J. Bailey, J. Balajthy,

677   S. Balashov, J. Bang, M. J. Barry, J. Barthel, D. Bauer, P. Bauer, A. Baxter, J. Belle, P. Beltrame,

678   J. Bensinger, T. Benson, E. P. Bernard, A. Bernstein, A. Bhatti, A. Biekert, T. P. Biesiadzinski, B. Bir-

679   rittella, K. E. Boast, A. I. Bolozdynya, E. M. Boulton, B. Boxer, R. Bramante, S. Branson, P. Brás,

680   M. Breidenbach, J. H. Buckley, V. V. Bugaev, R. Bunker, S. Burdin, J. K. Busenitz, J. S. Campbell,

681   C. Carels, D. L. Carlsmith, B. Carlson, M. C. Carmona-Benitez, M. Cascella, C. Chan, J. J. Cher-

682   winka, A. A. Chiller, C. Chiller, N. I. Chott, A. Cole, J. Coleman, D. Colling, R. A. Conley, A. Cottle,

683   R. Coughlen, W. W. Craddock, D. Curran, A. Currie, J. E. Cutter, J. P. da Cunha, C. E. Dahl,

684   S. Dardin, S. Dasu, J. Davis, T. J. R. Davison, L. de Viveiros, N. Decheine, A. Dobi, J. E. Y. Dob-

685   son, E. Druszkiewicz, A. Dushkin, T. K. Edberg, W. R. Edwards, B. N. Edwards, J. Edwards, M. M.

686   Elnimr, W. T. Emmet, S. R. Eriksen, C. H. Faham, A. Fan, S. Fayer, S. Fiorucci, H. Flaecher, I. M.

687   Fogarty Florang, P. Ford, V. B. Francis, F. Froborg, T. Fruth, R. J. Gaitskell, N. J. Gantos, D. Garcia,

688   A. Geffre, V. M. Gehman, R. Gelfand, J. Genovesi, R. M. Gerhard, C. Ghag, E. Gibson, M. G. D.

689   Gilchriese, S. Gokhale, B. Gomber, T. G. Gonda, A. Greenall, S. Greenwood, G. Gregerson, M. G. D.

690   van der Grinten, C. B. Gwilliam, C. R. Hall, D. Hamilton, S. Hans, K. Hanzel, T. Harrington, A. Har-

691   rison, C. Hasselkus, S. J. Haselschwardt, D. Hemer, S. A. Hertel, J. Heise, S. Hillbrand, O. Hitchcock,

692   C. Hjemfelt, M. D. Hoff, B. Holbrook, E. Holtom, J. Y-K. Hor, M. Horn, D. Q. Huang, T. W. Hurteau,

693   C. M. Ignarra, M. N. Irving, R. G. Jacobsen, O. Jahangir, S. N. Jeffery, W. Ji, M. Johnson, J. Johnson,

694   P. Johnson, W. G. Jones, A. C. Kaboth, A. Kamaha, K. Kamdin, V. Kasey, K. Kazkaz, J. Keefner,

695   D. Khaitan, M. Khaleeq, A. Khazov, A. V. Khromov, I. Khurana, Y. D. Kim, W. T. Kim, C. D. Kocher,

696   A. M. Konovalov, L. Korley, E. V. Korolkova, M. Koyuncu, J. Kras, H. Kraus, S. W. Kravitz, H. J.

697   Krebs, L. Kreczko, B. Krikler, V. A. Kudryavtsev, A. V. Kumpan, S. Kyre, A. R. Lambert, B. Lan-

698   derud, N. A. Larsen, A. Laundrie, E. A. Leason, H. S. Lee, J. Lee, C. Lee, B. G. Lenardo, D. S. Leonard,

R. Leonard, K. T. Lesko, C. Levy, J. Li, Y. Liu, J. Liao, F. T. Liao, J. Lin, A. Lindote, R. Linehan, W. H. Lippincott, R. Liu, X. Liu, C. Loniewski, M. I. Lopes, B. López Paredes, W. Lorenzon, D. Lucero, S. Luitz, J. M. Lyle, C. Lynch, P. A. Majewski, J. Makkinje, D. C. Malling, A. Manalaysay, L. Manenti, R. L. Mannino, N. Marangou, D. J. Markley, P. MarrLaundrie, T. J. Martin, M. F. Marzioni, C. Maupin, C. T. McConnell, D. N. McKinsey, J. McLaughlin, D. M. Mei, Y. Meng, E. H. Miller, Z. J. Minaker, E. Mizrachi, J. Mock, D. Molash, A. Monte, M. E. Monzani, J. A. Morad, E. Morrison, B. J. Mount, A. St. J. Murphy, D. Naim, A. Naylor, C. Nedlik, C. Nehrkorn, H. N. Nelson, J. Nesbit, F. Neves, J. A. Nikkel, J. A. Nikoleyczik, A. Nilima, J. O'Dell, H. Oh, F. G. O'Neill, K. O'Sullivan, I. Olcina, M. A. Olevitch, K. C. Oliver-Mallory, L. Oxborough, A. Pagac, D. Pagenkopf, S. Pal, K. J. Palladino, V. M. Palmaccio, J. Palmer, M. Pangilinan, S. J. Patton, E. K. Pease, B. P. Penning, G. Pereira, C. Pereira, I. B. Peterson, A. Piepke, S. Pierson, S. Powell, R. M. Preece, K. Pushkin, Y. Qie, M. Racine, B. N. Ratcliff, J. Reichenbacher, L. Reichhart, C. A. Rhyne, A. Richards, Q. Riffard, G. R. C. Rischbieter, J. P. Rodrigues, H. J. Rose, R. Rosero, P. Rossiter, R. Rucinski, G. Rutherford, D. Rynders, J. S. Saba, L. Sabarots, D. Santone, M. Sarychev, A. B. M. R. Sazzad, R. W. Schnee, M. Schubnell, P. R. Scovell, M. Severson, D. Seymour, S. Shaw, G. W. Shutt, T. A. Shutt, J. J. Silk, C. Silva, K. Skarpaas, W. Skulski, A. R. Smith, R. J. Smith, R. E. Smith, J. So, M. Solmaz, V. N. Solovov, P. Sorensen, V. V. Sosnovtsev, I. Stancu, M. R. Stark, S. Stephenson, N. Stern, A. Stevens, T. M. Stiegler, K. Stifter, R. Studley, T. J. Sumner, K. Sundarnath, P. Sutcliffe, N. Swanson, M. Szydagis, M. Tan, W. C. Taylor, R. Taylor, D. J. Taylor, D. Temples, B. P. Tennyson, P. A. Terman, K. J. Thomas, J. A. Thomson, D. R. Tiedt, M. Timalsina, W. H. To, A. Tomás, T. E. Tope, M. Tripathi, D. R. Tronstad, C. E. Tull, W. Turner, L. Tvrznikova, M. Utes, U. Utku, S. Uvarov, J. Va'vra, A. Vacheret, A. Vaitkus, J. R. Verbus, T. Vietanen, E. Voirin, C. O. Vuosalo, S. Walcott, W. L. Waldron, K. Walker, J. J. Wang, R. Wang, L. Wang, Y. Wang, J. R. Watson, J. Migneault, S. Weatherly, R. C. Webb, W. Z. Wei, M. While, R. G. White, J. T. White, D. T. White, T. J. Whitis, W. J. Wisniewski, K. Wilson, M. S. Witherell, F. L. H. Wolfs, J. D. Wolfs, D. Woodward, S. D. Worm, X. Xiang, Q. Xiao, J. Xu, M. Yeh, J. Yin, I. Young, and C. Zhang. The LUX-ZEPLIN (LZ) experiment. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 953:163047, February 2020.

[25] Spencer Klein and Joakim Nystrand. Exclusive Vector Meson Production in Relativistic Heavy Ion Collisions. *Phys. Rev. C*, 60(1):014903, June 1999. arXiv: hep-ph/9902259.

[26] M. Sajjad Athar, Shakeb Ahmad, and S. K. Singh. Neutrino nucleus cross sections for low energy neutrinos at SNS facilities. *Nucl. Phys.*, A764:551–568, 2006.

[27] Martin Hoferichter, Javier Menéndez, and Achim Schwenk. Coherent elastic neutrino-nucleus scattering: Eft analysis and nuclear responses. *Physical Review D*, 102(7), Oct 2020.