The Astrophysical Journal, 925:107 (17pp), 2022 February 1 https://doi.org/10.3847/1538-4357/ac350f

SNEWPY: A Data Pipeline from Supernova Simulations to Neutrino Signals

Amanda L. Baxter[1], Segev BenZvi[2], Joahan Castaneda Jaimes[3], Alexis Coleiro[4], Marta Colomer Molla[5], Damien Dornic[6], Tomer Goldhagen[7], Anne Graf[8], Spencer Griswold[2], Alec Habig[9], Remington Hill[10], Shunsaku Horiuchi[11,12], James P. Kneller[8], Rafael F. Lang[1], Massimiliano Lincetto[13], Jost Migenda[14], Ko Nakamura[15], Evan O'Connor[16], Andrew Renshaw[17], Kate Scholberg[18], Christopher Tunnell[19], Navya Uberoi[2], and Arkin Worlikar[20]
The SNEWS Collaboration
[1] Department of Physics and Astronomy, Purdue University, West Lafayette, IN, USA
[2] University of Rochester, Rochester, NY, USA
[3] Caltech, Pasadena, CA, USA
[4] Université de Paris, CNRS, AstroParticule et Cosmologie, Paris, France
[5] Inter-university Institute for High Energies, Université libre de Bruxelles, Brussels, Belgium
[6] Aix Marseille Univ, CNRS/IN2P3, CPPM, Marseille, France
[7] University of North Carolina—Chapel Hill, Chapel Hill, NC, USA
[8] NC State University, Raleigh, NC, USA; jim_kneller@ncsu.edu
[9] University of Minnesota Duluth, Duluth, MN, USA
[10] Laurentian University, Sudbury, ON, Canada
[11] Virginia Tech, Blacksburg, VA 24061, USA
[12] Kavli IPMU (WPI), UTIAS, The University of Tokyo, Kashiwa, Chiba 277-8583, Japan
[13] Astronomisches Institut, Ruhr-Universitaet Bochum, D-44780 Bochum, Germany
[14] King's College London, London, UK
[15] Fukuoka University, Fukuoka, Japan
[16] The Oskar Klein Centre, Department of Astronomy, Stockholm University, AlbaNova, SE-106 91 Stockholm, Sweden
[17] University of Houston, Houston, TX, USA
[18] Duke University, Durham, NC, USA
[19] Rice University, Houston, TX, USA
[20] Georgia Tech, Atlanta, GA, USA
Received 2021 September 15; revised 2021 October 25; accepted 2021 October 27; published 2022 January 31

Abstract

Current neutrino detectors will observe hundreds to thousands of neutrinos from Galactic supernovae, and future detectors will increase this yield by an order of magnitude or more. With such a data set comes the potential for a huge increase in our understanding of the explosions of massive stars, nuclear physics under extreme conditions, and the properties of the neutrino. However, there is currently a large gap between supernova simulations and the corresponding signals in neutrino detectors, which will make any comparison between theory and observation very difficult. SNEWPY is an open-source software package that bridges this gap. The SNEWPY code can interface with supernova simulation data to generate from the model either a time series of neutrino spectral fluences at Earth, or the total time-integrated spectral fluence. Data from several hundred simulations of core-collapse, thermonuclear, and pair-instability supernovae is included in the package. This output may then be used by an event generator such as sntools or an event rate calculator such as the SuperNova Observatories with General Long Baseline Experiment Simulator (SNOwGLoBES). Additional routines in the SNEWPY package automate the processing of the generated data through the SNOwGLoBES software and collate its output into the observable channels of each detector. In this paper we describe the contents of the package, the physics behind SNEWPY, the organization of the code, and provide examples of how to make use of its capabilities.

Unified Astronomy Thesaurus concepts: Supernova neutrinos (1666); Particle astrophysics (96); Supernovae (1668)

1. Introduction

The neutrino signal from a supernova in the Milky Way will be a golden opportunity to advance our understanding of how stars explode and probe the properties of the neutrino. For reviews of the physics potential of a supernova neutrino signal we refer the reader to several recent reviews (Scholberg 2012; Janka et al. 2016; Mirizzi et al. 2016; Horiuchi & Kneller 2018). Many different detectors worldwide will record events from the supernova and naturally there is much interest from experimenters about what they might observe. Present-day detectors, such as Super-Kamiokande (Ikeda et al. 2007), Borexino (Monzani 2006), Kamioka Liquid-scintillator Anti-Neutrino Detector (KAMLAND; Eguchi et al. 2003), Large Volume Detector (LVD; Agafonova et al. 2008, 2007), or the dedicated supernova burst detector Helium And Lead Observatory (HALO; Duba et al. 2008), are expected to record tens to hundreds of thousands of neutrino events each from a Galactic or nearby core-collapse supernova (CCSN). Such a neutrino burst would also be recorded with very high statistics in detectors such as IceCube (Halzen et al. 1994, 1996; Abbasi et al. 2011) and KM3NeT (Aiello et al. 2021) but without event-by-event reconstruction. Ever larger and more ambitious detectors are under construction or proposed, including Deep Underground Neutrino Experiment (DUNE; Acciarri et al. 2016a, 2015, 2016b), Hyper-Kamiokande (Abe et al. 2018), and Jiangmen Underground Neutrino Observatory (JUNO; An et al. 2016). The physical and

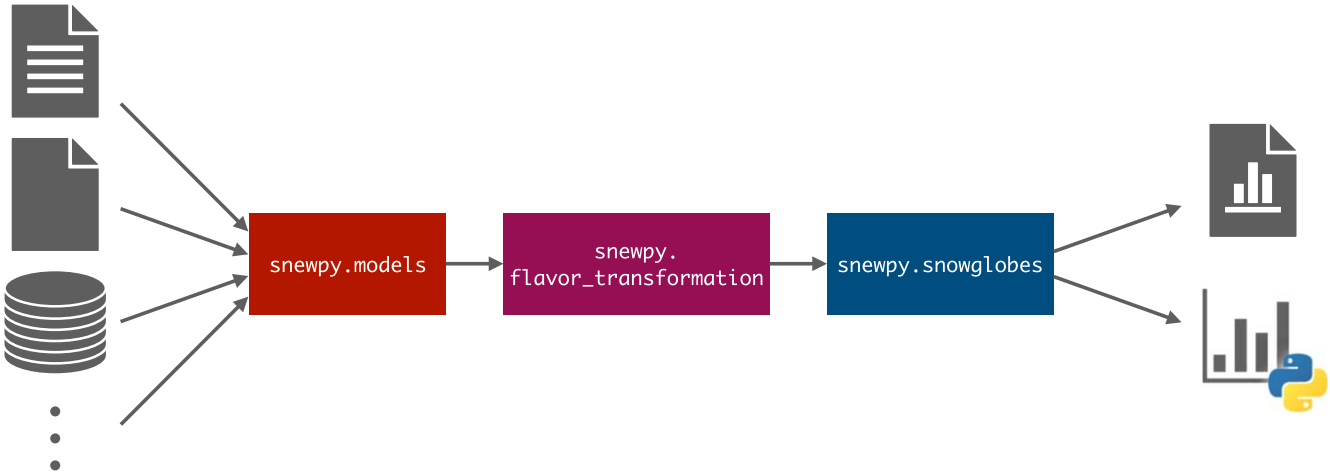

Figure 1. Flowchart showing the complete SNEWPY pipeline. SNEWPY supports a wide variety of input formats and can output results as plots or as a Python dictionary for further analysis.

astronomical potential of megaton-scale detectors has been considered (Suzuki 2001; Kistler et al. 2011). In addition, dark matter detectors are becoming of sufficient size that they too are capable of detecting a significant number of neutrinos from a CCSN burst via coherent elastic neutrino–nucleus scattering (Horowitz et al. 2003; Lang et al. 2016; Lai 2021). Other types of supernovae or compact object mergers involving neutron stars also produce neutrino bursts that are detectable if they occur sufficiently nearby (Odrzywolek & Plewa 2011; Wright et al. 2016, 2017b, 2017a; Rosswog & Liebendörfer 2003; Caballero et al. 2009; Lin & Lunardini 2020).

With so many detectors capable of detecting the next Galactic or near-Galactic supernova, there is a need for theoretical predictions of neutrino emission from supernova simulations and the resulting neutrino signals in detectors on Earth. To determine how well these detectors can provide quantitative information about the supernova explosion mechanism and nuclear physics under extreme conditions, we need a suite of neutrino signal templates to compare observations with theory. However, there is presently a substantial gulf between the data from a supernova simulation and the neutrino spectra at Earth. This gap has been bridged on a few occasions (Kneller et al. 2008; Reid et al. 2011; Wu et al. 2015; Gava et al. 2009)—see also Wright et al. 2016, 2017b, 2017a—but there are not enough such data sets to cover the wide range of available supernova models and they are time consuming to generate.

The SNEWPY code[21] has been written with the intent of bridging this gap. SNEWPY provides a consistent interface to hundreds of supernova simulation data sets to extract the neutrino emission. It can then convolve this with a prescription for the flavor transformation to generate the spectral fluence (time-integrated flux) reaching a detector on Earth either in a set of time bins or over the entire simulation time window. This output can be processed with an event generator such as sntools (Migenda et al. 2021) or an event rate calculator such as the SuperNova Observatories with General Long Baseline Experiment Simulator (SNOwGLoBES; Scholberg 2021). SNEWPY also includes routines that automate the processing of the data with SNOwGLoBES and collate its output to determine the total event rates in the observable channels of each detector.

In this paper we give an overview of the SNEWPY package, its capabilities, and the large library of simulation data sets. This paper is a complement to the companion paper published in the *Journal of Open Source Software* (Baxter et al. 2021). The overall organization of the code is described in Section 2 before proceeding to discuss in detail SNEWPY's interfaces to simulation data in Section 3, the flavor transformations prescriptions in Section 4, and the SNOwGLoBES interface in Section 5. To demonstrate the capabilities of the software and how it is used in practice, in Section 6 we provide two examples of applications of SNEWPY: the first example is a complete data pipeline using SNEWPY's interface with SNOwGLoBES, and the second example shows how it can be integrated into the event generator sntools.

2. The Structure of SNEWPY

SNEWPY is an open-source package written in Python. It is built upon NumPy (Harris et al. 2020) and SciPy (Virtanen et al. 2020) and uses Astropy (Astropy Collaboration et al. 2013, 2018) for model I/O and unit conversions. Figure 1 shows the high-level structure of a complete supernova data pipeline implemented in SNEWPY. This pipeline takes an input file from a supernova simulation, then

1. extracts neutrino fluxes produced in the supernova as a function of time, energy, and flavor;
2. applies a flavor transformation prescription to determine the fluxes reaching Earth; and
3. runs them through the detector response software SNOwGLoBES,

before providing the computed event rates per detector and interaction channel either in the form of figures or as structured data for further processing.

Matching this set of steps, SNEWPY is divided into three main modules[22]: `snewpy.models`, which interfaces with the simulation data that comes with SNEWPY; `snewpy.flavor_transformation`, which implements the different flavor transformation prescriptions; and `snewpy.snowglobes`, which integrates with SNOwGLoBES and provides functions to generate SNOwGLoBES-formatted data files, runs them through SNOwGLoBES for a chosen set of

[21] SNEWPY is available for download at https://github.com/SNEWS2/snewpy.

[22] Notation: `monospace` font is used to refer to modules and file names, **bold** font is used for class names, and *italic* font indicates class members.

neutrino detectors, and finally collates the resulting outputs. SNEWPY was designed in this way so that the user can insert or extract data at the interfaces between the components. For example, the user may have an alternative method (such as an analytic formula) for generating the neutrino spectra at Earth and therefore does not need to generate a time series from a simulation. Another user could use SNEWPY to provide a consistent interface to different supernova models or a large library of flavor transformations, without having to run the SNOwGLoBES software. In what follows, we describe these three main modules of SNEWPY in more detail.

3. Supernova Neutrino Models

Core-collapse supernova models depend on many different factors—both physical parameters of the progenitor (e.g., its mass and metallicity) and implementation parameters of the simulation (e.g., the degree of spatial symmetry and the equation of state). The choice of parameters strongly affects the predicted neutrino emission and thus what we detect at Earth. To best estimate the sensitivity of a detector (or set of detectors) to a wide variety of explosion models and parameter values, it is common practice to use a broad sample of simulation models. It is better still if those models are provided by different modeling groups using different numerical algorithms and approaches. Unfortunately, this often means that simulation data is provided in different formats. Rather than attempt to reduce all simulations to a common format, SNEWPY solves this problem by providing an extensible module called `models` which contains an abstract **SupernovaModel** base class with the absolute minimal functionality to undertake the calculation of the neutrino spectra at Earth from the simulation. The **SupernovaModel** class defines just three member functions:

1. *get_time*: access the list of snapshot times from the simulation;
2. *get_initial_spectra*: obtain the neutrino spectra as a function of time, energy, flavor, and angle at the surface of emission (neutrinosphere) within the progenitor; and
3. *get_transformed_spectra*: obtain neutrino spectra as a function of time, energy, and flavor after some flavor transformation (see Section 4).

Specific supernova models are implemented as subclasses of **SupernovaModel**. Each subclass must contain custom implementations of *get_time* and *get_initial_spectra*, since these depend on the format of the input files. At the present time, `snewpy.models` contains 13 **SupernovaModel** subclasses, which are named according to the modeling group or publication describing the simulations. A list of available models and their properties is provided in Table 1 and two example models are shown in Figure 2.

Across these models, neutrino fluxes from hundreds of core-collapse simulations are available and can be downloaded by SNEWPY on demand. In addition, SNEWPY has data from two pair-instability supernovae (PISNe) from Wright et al. (2017a), and two Type Ia supernovae (Wright et al. 2016, 2017a).[23] Given so many models we do not attempt to describe them all here and refer the reader to their associated literature.

[23] Note that the PISNe and Type Ia data sets differ from the core-collapse simulations in that (a) flavor transformations have already been included, (b) for the Type Ia data sets there are multiple lines of sight, and (c) the data is already in SNOwGLoBES format.

Each **SupernovaModel** subclass is able to construct a neutrino spectrum Φ_α at the neutrinosphere for the neutrino species ν_e, ν_x, $\bar{\nu}_e$, and $\bar{\nu}_x$. Here, x stands for the μ or τ flavors, which are treated as identical in almost all current supernova simulations, including those in Table 1.[24] Once simulations that treat all six flavors—three neutrino plus three antineutrino flavors—as separate become more frequent, we expect to update SNEWPY to accommodate this.

The customized *get_initial_spectra* member function of each model class extracts the neutrino spectra $\Phi(E_\nu)$ from the simulation data if it is provided but, more commonly, the method constructs a spectrum according to a parameterization. A common parameterization of this spectrum—though not required by SNEWPY—is in terms of the luminosity L_ν, the mean energy $\langle E_\nu \rangle$, and the spectral shape parameter α (Keil et al. 2003; Tamborra et al. 2012):

$$\Phi(E_\nu) = \frac{L_\nu}{\langle E_\nu \rangle} \frac{(\alpha+1)^{\alpha+1}}{\langle E_\nu \rangle \, \Gamma(\alpha+1)} \left(\frac{E_\nu}{\langle E_\nu \rangle} \right)^\alpha \times \exp\left(-\frac{(\alpha+1)E_\nu}{\langle E_\nu \rangle} \right). \quad (1)$$

This shape parameter can be calculated from the energy moments $\langle E_\nu^k \rangle$ via

$$\frac{\langle E_\nu^k \rangle}{\langle E_\nu^{k-1} \rangle} = \frac{k+\alpha}{1+\alpha} \langle E_\nu \rangle. \quad (2)$$

If we use $k = 2$, the most common additional moment output from a simulation, we find

$$\alpha = \frac{2\langle E_\nu \rangle^2 - \langle E_\nu^2 \rangle}{\langle E_\nu^2 \rangle - \langle E_\nu \rangle^2}. \quad (3)$$

Whatever the method used to construct the initial spectra, the initial spectra generally evolve with time as the simulation proceeds.

The last method of the **SupernovaModel** class is *get_transformed_spectra*. This method convolves the initial spectra with a flavor transformation prescription, which we now describe.

4. The Flavor Transformation Prescriptions

The second component of SNEWPY is the flavor transformation prescriptions which relate the neutrino fluxes produced in the supernova to those arriving on Earth. This library of prescriptions in SNEWPY accounts for the effects of propagation through the outer layers of the star, neutrino decay during vacuum propagation to Earth, or mixing with sterile neutrinos. The user may easily create new prescriptions to include additional scenarios or vary oscillation parameters for one of the existing prescriptions to test the sensitivity of a detector. In this section we first describe the general form of these transformations and then their implementation in SNEWPY.

4.1. General Form of Flavor Transformations

The spectral fluxes of the three neutrino flavors at Earth can be arranged into a column vector, $F_F(r_\oplus) = (F_e(r_\oplus), F_\mu(r_\oplus), F_\tau(r_\oplus))^T$, where we use the subscript F to

[24] Many current simulations go further and do not distinguish between ν_x and $\bar{\nu}_x$, so there are just three distinct spectra.

Table 1
CCSN Simulation Models with Neutrino Emission Tables Included in SNEWPY

Model	Masses ($M_\odot$)	Time Range (s)	Comment
Nakazato_2013	13, 20, 30, 50	−0.05–20.0 (SN) −0.14–0.84 (BH)	Binned spectra from Nakazato et al. (2013) reparameterized in terms of $\langle E \rangle$ and α. Uses Shen et al. (1998, 2011) and LS220 equations of state (EOS) and includes a black hole (BH) formation scenario
OConnor_2013	12–33 (1 $M_\odot$ steps), 35, 40, 45, 50, 55, 60, 70, 80, 100, 120	−0.015–0.5	32 progenitors from Woosley & Heger (2007) using both the LS220 and H. Shen EOS for 64 models in total studying the accretion phase in spherically symmetric simulations (O'Connor & Ott 2013)
Tamborra_2014	20, 27	0.006–0.338 (20) 0.011–0.552 (27)	3D models from Tamborra et al. (2014), using emission direction with maximum Standing Accretion Shock Instability (SASI) signal
OConnor_2015	40	−0.378–0.537	BH-forming simulation (O'Connor 2015) using a 40 $M_\odot$ progenitor from Woosley & Heger (2007) and LS220 EOS
Sukhbold_2015	9.6, 27	−0.35–15.44	PROMETHEUS-VERTEX simulation data presented in Rampp & Janka (2002) and Sukhbold et al. (2016), with LS220 and SFHo EOS
Bollig_2016	11.2, 27	−0.17–7.60 (11) −0.34–7.60 (27)	The s11.2 c and s27.0 c models shown in Figure 17 of Mirizzi et al. (2016), using the LS220 EOS
Walk_2018	15	0.01–0.33	Models from Walk et al. (2018) demonstrating the effect of progenitor rotation on SASI oscillations
Walk_2019	40	0.01–0.57	BH-forming simulation from Walk et al. (2020) with strong SASI features prior to BH formation
Fornax_2019	9, 10, 12, 13, 14, 15, 16, 19, 25, 60	0.01–1.04 (9) 0.01–0.40 (60)	Full 3D simulation of Vartanyan et al. (2019) produced using the FORNAX code (Skinner et al. 2019)
Warren_2020	9.0-100.0 (200 in total)	−0.22–4.7, depending on the simulation	1D FLASH simulations with Supernova Turbulence In Reduced-dimensionality presented in Warren et al. (2020)
Kuroda_2020	20	0.00–0.476, depending on rotation and magnetic field	3D simulation of a magnetized rotating star from Kuroda (2021) using a 20 $M_\odot$ progenitor by Woosley & Heger (2007) and SFHo EOS
Fornax_2021	12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27	−0.21–4.49 (12) −0.31–4.59 (27)	Axisymmetric models simulated to 4.5 s post-bounce using the FORNAX code, described in Burrows & Vartanyan (2021)
Zha_2021	16, 17, 18, 19, 19.89, 20, 21, 22.39, 23, 24, 25, 26, 30, 33	−0.21–2.02 (16) −0.30–0.35 (33)	Failing CCSN simulations with a hybrid EOS including a hadron-quark phase transition (Zha et al. 2021)

Note. User-defined CCSN models can be created by subclassing **SupernovaModel** in the module `snewpy.models`.

indicate the neutrino flavor basis, and $r_\oplus$ indicates the location of Earth. When we refer to a generic element of the flavor basis, we shall use Greek subscripts α and β; the subscripts e, μ, and τ are the specific flavor of the neutrinos, while the subscript x indicates either μ or τ. The neutrino spectra at Earth are not those that were emitted at the neutrinosphere inside the progenitor, but have been mixed on their way to Earth by flavor transformation effects. The flavor transformation of neutrinos is a quantum mechanical phenomenon that occurs due to the mismatch between the flavor states of the neutrinos and the eigenstates of the free Hamiltonian. For a complete discussion of the phenomenon we refer the reader to the reviews mentioned previously (Mirizzi et al. 2016; Horiuchi & Kneller 2018). Flavor transformation of the neutrinos occurs while they are within the supernova and then at some point on their trip to Earth the neutrinos decohere and arrive in their mass eigenstates. We shall denote quantities which are in the mass basis by the subscript M and when we wish to refer to a generic mass state we shall use the italic subscripts i, j; the subscripts 1, 2, and 3 are specific mass states.

The spectral flux of a particular flavor at Earth is the incoherent sum of the parts of the spectral flux of each mass state with a given flavor, which can be written as

$$F_F(r_\oplus) = D\, F_M(r_\oplus). \tag{4}$$

The elements of the matrix D are given by $D_{\alpha i} = ||U_{V,\alpha i}||^2$, where $U_{V,\alpha i}$ are the elements of the vacuum mixing matrix for neutrinos. The antineutrino fluxes in the flavor basis are related to the antineutrino fluxes in the mass basis by the same matrix D. We shall denote antineutrino quantities by an overbar. The relationship for antineutrinos is thus $\bar{F}_F(r_\oplus) = D\bar{F}_M(r_\oplus)$. The column vector of the spectral fluxes of the mass states are the diagonal elements of the spectral flux matrix $\mathcal{F}_M$, constructed by

$$\mathcal{F}_M = c \int \rho_M \cos\theta \, d\Omega, \tag{5}$$

where c is the speed of light, ρ_M is the density matrix in the mass basis, θ is the angle with respect to the line from the supernova to Earth, and the integral is over all the neutrino propagation angles. The conversion of the flux matrix to a

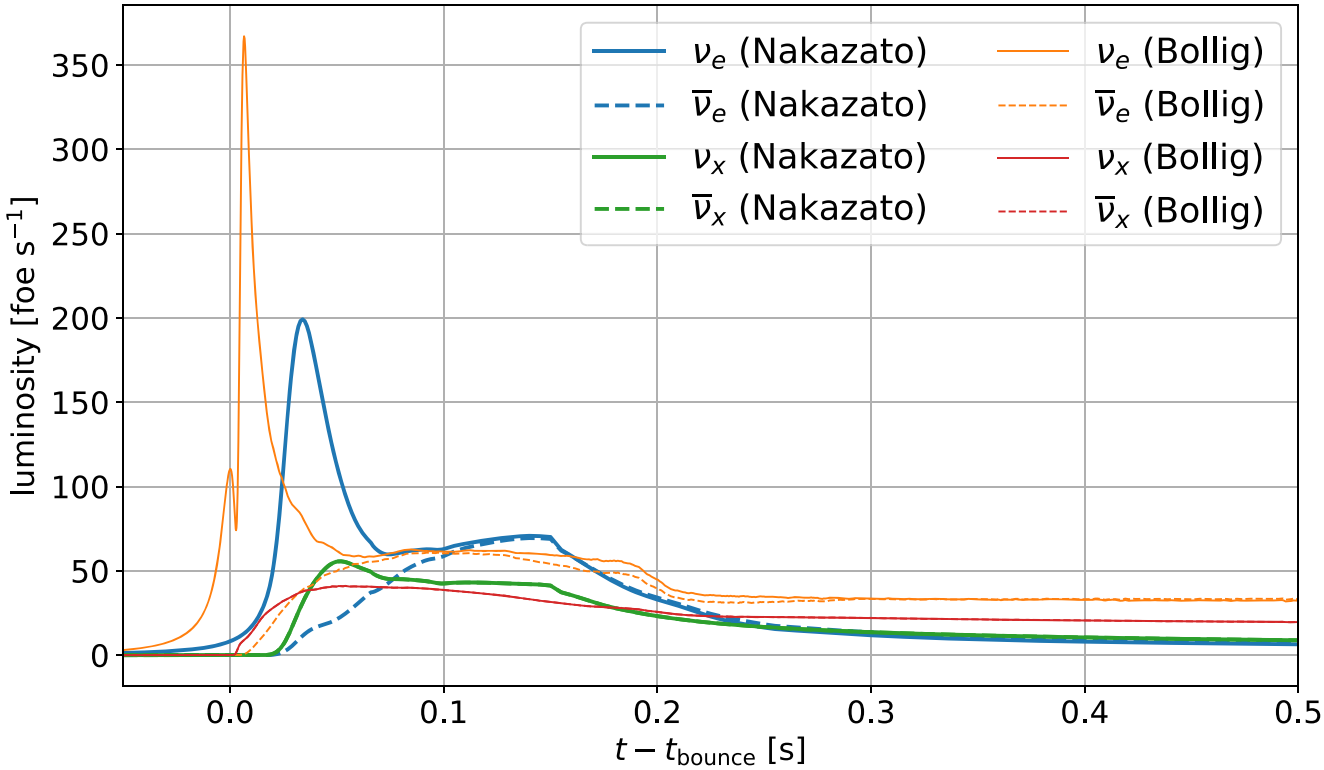

Figure 2. Luminosity of different neutrino flavors in units of foe s^{-1} (1 foe = 10^{51} erg) as a function of time for the **nakazato-shen-z0.004-trev100ms-s20.0** model (thick lines, blue and green) and the **Bollig_2016/s27.0c** model (thin lines, orange and red). Both models come with SNEWPY and were originally presented in Nakazato et al. (2013) and Mirizzi et al. (2016), respectively.

column vector—ignoring the off-diagonal elements of the flux matrix—is the effect of decoherence. Accounting for decoherence, F_M can be written as

$$F_M(r_\oplus) = \sum_j |\nu_j\rangle \langle\nu_j|\mathcal{F}_M|\nu_j\rangle \tag{6}$$

where the $\{|\nu_j\rangle\}$ are basis vectors in the mass basis. The same conversion occurs for the antineutrinos. The matrix $\mathcal{F}_M$ is

$$\mathcal{F}_M(r_\oplus) = \frac{1}{4\pi d^2}\,\Phi_M(r_\oplus), \tag{7}$$

where d is the distance of the supernova from Earth, and $\Phi_M(r_\oplus)$ is the spectral number luminosity matrix in the mass basis at Earth. The matrix Φ_M at Earth is related to the matter basis spectral matrix at the neutrinosphere $\Phi_M(R_\nu)$—the matter basis becomes the mass basis in the vacuum—by

$$\Phi_M(r_\oplus) = S_M(r_\oplus, R_\nu)\,\Phi_M(R_\nu) S_M^\dagger(r_\oplus, R_\nu) \tag{8}$$

with $S_M(r_\oplus, R_\nu)$ being the propagator of the matter/mass states and R_ν indicating the neutrinosphere. Note that this formula does not account for the absorption, emission, or scattering of neutrinos. Finally, the matrix $\Phi_M(R_\nu)$ is related to the spectral matrix in the flavor basis at the neutrinosphere by

$$\Phi_M(R_\nu) = U^\dagger\,\Phi_F(R_\nu)\,U \tag{9}$$

with U as the so-called matter mixing matrix at the neutrinosphere. The matrix U and the equivalent matrix for the antineutrinos $\bar{U}$ depend upon the mass ordering and the initial density. The spectral matrix at the neutrinosphere in the flavor basis is taken to be pure diagonal:

$$\begin{aligned}\Phi_F(R_\nu) &= \begin{pmatrix} \Phi_e(R_\nu) & 0 & 0 \\ 0 & \Phi_\mu(R_\nu) & 0 \\ 0 & 0 & \Phi_\tau(R_\nu)\end{pmatrix} \\ &= \textstyle\sum_\beta |\nu_\beta\rangle\langle\nu_\beta|\Phi_\beta(R_\nu)\end{aligned} \tag{10}$$

where the $\{|\nu_\beta\rangle\}$ are the basis vectors in the flavor basis. Note that in practice the spectra $\Phi_\beta(R_\nu)$ are usually taken to be the spectra at the largest radius in the simulation and not the actual neutrinosphere. Putting all this together so as to relate $F_F(r_\oplus)$ to $\Phi_F(R_\nu)$ and $\bar{F}_F(r_\oplus)$ to $\bar{\Phi}_F(R_\nu)$, we obtain

$$\begin{aligned}F_F(r_\oplus) = \frac{1}{4\pi d^2}\,D\sum_i\sum_\beta \Phi_\beta(R_\nu)|\nu_i\rangle\langle\nu_i| \\ S_M(r_\oplus, R_\nu)U^\dagger|\nu_\beta\rangle\langle\nu_\beta|\,U\,S_M^\dagger(r_\oplus, R_\nu)|\nu_i\rangle\end{aligned} \tag{11}$$

$$\begin{aligned}\bar{F}_F(r_\oplus) = \frac{1}{4\pi d^2}\,D\sum_i\sum_\beta \bar{\Phi}_\beta(R_\nu)|\bar{\nu}_i\rangle\langle\bar{\nu}_i| \\ \bar{S}_M(r_\oplus, R_\nu)\bar{U}^\dagger\,|\bar{\nu}_\beta\rangle\langle\bar{\nu}_\beta|\,\bar{U}\,\bar{S}_m^\dagger(r_\oplus, R_\nu)\,|\bar{\nu}_i\rangle.\end{aligned} \tag{12}$$

In these equations, the quantities $\langle\nu_i|S_M(r_\oplus, R_\nu)U^\dagger\nu_\beta$ have the physical interpretation of being the probability amplitudes that a neutrino emitted as a particular flavor β reaches Earth in mass state i. The propagators S_M and $\bar{S}_M$ and the mixing matrices U and $\bar{U}$ are the quantities that can vary depending upon the evolution of the matter states through the mantle of the supernova, and the flavor transformation scenario.

Table 2
Flavor Transformation Models Included in SNEWPY

Prescription Name	Hierarchy	Comment
NoTransformation	⋯	No flavor transformation applied.
CompleteExchange	⋯	Electron flavors completely swapped with a heavy lepton flavor.
AdiabaticMSW	NMO, IMO	Adiabatic neutrino evolution for normal or inverted hierarchy.
NonAdiabaticMSWH	NMO, IMO	The H resonance is nonadiabatic and the L resonance is adiabatic.
TwoFlavorDecoherence	NMO, IMO	50% mixing between whichever states mix at the H resonance.
ThreeFlavorDecoherence	⋯	33% mixing between all flavors and both neutrinos and antineutrinos.
NeutrinoDecay	NMO, IMO	Adiabatic evolution through the supernova mantle followed by decay of the heaviest neutrino mass state to the lightest during vacuum propagation to Earth. Uses the approximation that the energy of the neutrino does not change.
AdiabaticMSWes	NMO, IMO	Mixing for four neutrino flavors, where the fourth mass state is the heaviest and the new "es" Mikheyev–Smirnov–Wolfenstein (MSW) resonance is adiabatic. The mass ordering refers to the three lightest neutrinos.
NonAdiabaticMSWes	NMO, IMO	Mixing for four neutrino flavors, where the fourth mass state is the heaviest and the new "es" MSW resonance is nonadiabatic. The mass ordering refers to the three lightest neutrinos.

Note. Explanations and probabilities for each transformation prescription are provided in the Appendix. User-defined transformations can be created by subclassing FlavorTransformation in the module `snewpy.flavor_transformation`.

Equations (11) and (12) can be simplified considerably by summarizing the effect of transformations as

$$F_e = \frac{1}{4\pi d^2}[p_{ee}\Phi_{\nu_e} + p_{ex}\Phi_{\nu_x}], \tag{13}$$

$$F_x = \frac{1}{4\pi d^2}[p_{xe}\Phi_{\nu_e} + p_{xx}\Phi_{\nu_x}], \tag{14}$$

$$\bar{F}_e = \frac{1}{4\pi d^2}[\bar{p}_{ee}\bar{\Phi}_{\nu_e} + \bar{p}_{ex}\bar{\Phi}_{\nu_x}], \tag{15}$$

$$\bar{F}_x = \frac{1}{4\pi d^2}[\bar{p}_{xe}\bar{\Phi}_{\nu_e} + \bar{p}_{xx}\bar{\Phi}_{\nu_x}], \tag{16}$$

where the quantities $\{p_{\alpha\beta}\}$ and $\{\bar{p}_{\alpha\beta}\}$ are the various survival and transition probabilities that a neutrino emitted as a particular flavor β is detected as flavor α at Earth. Equations (13)–(16) are generalizations of those found in Dighe & Smirnov (2000), Horiuchi & Kneller (2018), and elsewhere, since we shall consider cases where $p_{\alpha\beta} \neq p_{\beta\alpha}$.

4.2. Implementation of Flavor Transformations in SNEWPY

SNEWPY implements neutrino flavor transformations in its `flavor_transformation` module. This module contains an abstract base class FlavorTransformation, which defines a minimal interface for computing neutrino survival and transformation probabilities used in Equations (13) to (16). For example, the survival probability $p_{ee} = \Pr(\nu_e \to \nu_e)$ is computed using the member function *prob_ee*, and the transition probability $p_{ex} = \Pr(\nu_x \to \nu_e)$ is computed using the member function *prob_ex*. For forward compatibility, SNEWPY has been written such that the transition probabilities can depend on time and neutrino energy.

Using this interface, it is easy to support many different kinds of flavor transformation scenarios with subclasses that inherit the interface of FlavorTransformation. Currently, the module `snewpy.flavor_transformation` supports 15 transformation scenarios: 6 for the normal mass ordering (NMO), 6 for the inverted mass ordering (IMO), and 3 that are independent of the mass ordering. The list of transformations included in SNEWPY is provided in Table 2. The values used for the neutrino mixing parameters are contained inside two instances of a class called MixingParameters which is also part of SNEWPY: one instance for the NMO and another for the IMO. They are set by default to the values from Esteban et al. (2020) but may be modified by the user.

With both the model and flavor transformation prescriptions now defined, the previously mentioned *get_transformed_spectra* member function of each model class is seen to be the implementation of Equations (13)–(16). For the purposes of illustration, we show in Figure 3 an example of how flavor transformations affect the neutrino spectra. This example uses the ThreeFlavorDecoherence prescription, which mixes the initial spectra for the neutrinos so that the flux of each flavor at Earth is equal (and the same for antineutrinos).

5. SNEWPY's Interface with SNOwGLoBES

The module `snewpy.snowglobes` contains functions to interact with the SNOwGLoBES software. This interaction typically occurs in three steps: generating the data files of the neutrino fluences at Earth in the SNOwGLoBES format, processing those files through the SNOwGLoBES software, and then collating the data output from SNOwGLoBES.

5.1. Generating SNOwGLoBES Input Files

The `snewpy.snowglobes` module contains two functions, *generate_time_series* and *generate_fluence*. The *generate_time_series* function constructs a set of data files at a set of snapshot times from a chosen simulation. For each snapshot, the code extracts (or constructs) the neutrino spectra at the neutrinosphere. It then applies a user-selected flavor transformation prescription to those spectra and scales with a user-selected supernova distance to generate the flux at Earth. Finally, the function writes the fluence—the flux multiplied by the time bin width—for the snapshot to a SNOwGLoBES-formatted file and collates the files for different snapshot times into a single compressed output file. A number of options are

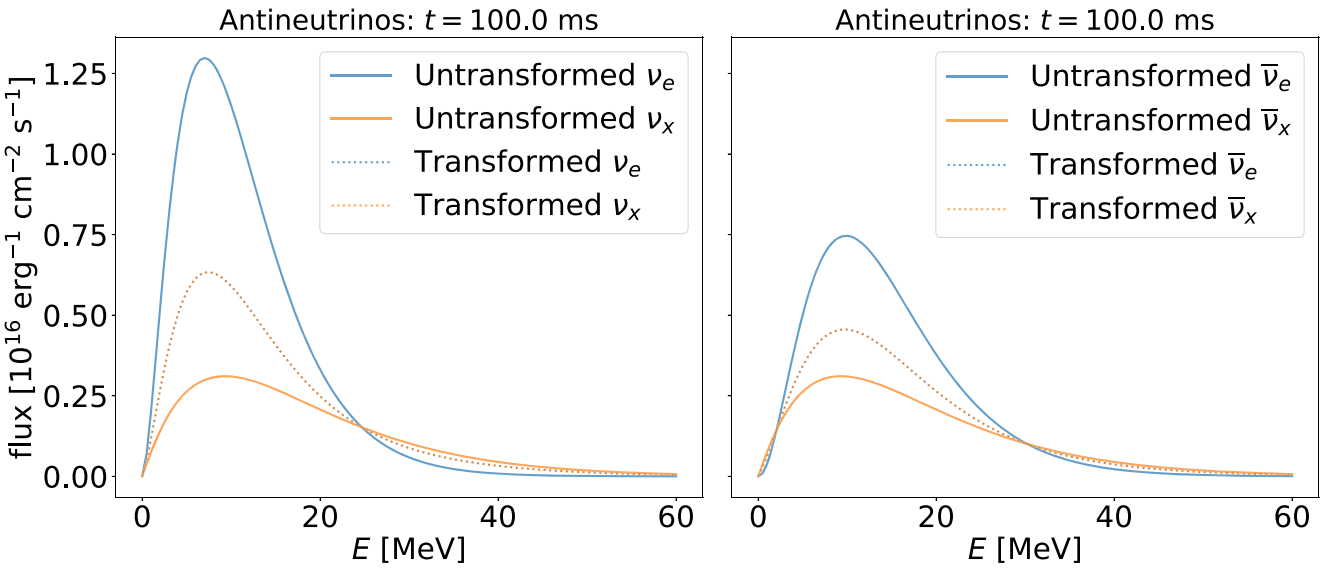

Figure 3. The initial (solid lines) and transformed (dashed lines) spectral flux for neutrinos (left panels) and antineutrinos (right panels) using the **ThreeFlavorDecoherence** prescription and a distance to the supernova of 10 kpc. Both mass orderings are equivalent in this prescription. We show the nakazato-shen-z0.004-trev100ms-s20.0 model (Nakazato et al. 2013) that comes with SNEWPY at the simulation time of 100 ms and at a supernova distance of 10 kpc.

available to the user to control how many snapshots are created and their spacing in time.

The *generate_fluence* function works similarly but applies a time integration of the fluence in each time bin to compute the total fluence.

5.2. Simulating Detector Effects with SNOwGLoBES

The `snewpy.snowglobes` module contains a function named *simulate* which takes a compressed file, such as those generated by *generate_time_series* or *generate_fluence*, and runs the contents through the SNOwGLoBES software. The bulk of the *simulate* function is a translation into Python of the script `supernova.pl` that comes with SNOwGLoBES; *simulate* also creates a `supernova.glb` file then invokes `supernova`, the executable in SNOwGLoBES. The mandatory arguments passed to *simulate* are the location of the SNOwGLoBES installation directory and the path to the compressed input file. Another, optional argument is the name of a detector. If this argument is not supplied, SNOwGLoBES is repeatedly invoked for all supported detectors. There is no output from the *simulate* function itself; instead, upon successful execution of the function a large set of files (each containing the event rate in a given set of energy bins for a particular interaction channel due to a particular flavor) is generated in the SNOwGLoBES output folder. We refer the reader to the SNOwGLoBES documentation (Scholberg 2021) for details about the interaction channels that SNOwGLoBES computes and how the calculation is done.

5.3. Collating Information from SNOwGLoBES

Finally, the `snewpy.snowglobes` module contains a function named *collate*, which collates the output generated by SNOwGLoBES into the observable channels of each detector in the four combinations of weighted and unweighted event rates and with and without applying detector energy smearing.

Like the *simulate* function, the mandatory arguments passed to *collate* are the location of the SNOwGLoBES installation directory and the name of the compressed input file. Again, an optional final argument is the name of the detector that was chosen for the SNOwGLoBES event rate calculation and if this argument is not supplied, it is assumed all detectors in the SNOwGLoBES suite were used. The *collate* function writes the collated data to a new compressed file, whose name is based on the name of the file that was run through SNOwGLoBES, and returns a dictionary of the collated results. Additional optional arguments to the function will cause it to generate simple Matplotlib (Hunter 2007) histogram figures from the collated data, which will be also placed in the compressed file, or to delete SNOwGLoBES output files.

6. SNEWPY Usage Examples

Included in the SNEWPY repository are many Jupyter notebooks containing many instances of using the software for various purposes. Here we provide two short examples.

6.1. Using SNEWPY with SNOwGLoBES

In this section we provide a simple example of using SNEWPY with the SNOwGLoBES software. This particular script reproduces the Super-Kamiokande entry in Table 4 of Al Kharusi et al. (2021) for the 11.2 $M_\odot$ model assuming adiabatic MSW oscillations with the normal mass ordering (4045 neutrinos). A script that calculates the entire Table 4 in Al Kharusi et al. (2021) is one of the previously mentioned Jupyter notebooks that are included in the package.

```
from snewpy import snowglobes

SNOwGLoBES_path = "/path/to/snowglobes/"# directory where SNOwGLoBES is located
SNEWPY_model_dir = "/path/to/snewpy/models/"# directory with model input files

distance = 10# Supernova distance in kpc
detector = "wc100kt30prct"# Name of SNOwGLoBES detector model to use
```

(Continued)

```
modeltype = 'Bollig_2016' # Model type from snewpy.models
model = 's11.2c' # Name of model
transformation = 'AdiabaticMSW' # Desired flavor transformation

# Construct file system path of model file and name of output file
model_path = SNEWPY_model_dir + "/" + modeltype + "/" + model
outfile = modeltype + "_" + model + "_" + transformation

# Now, do the main work:
print("Generating fluence files ...")
tarredfile = snowglobes.generate_fluence(model_path, modeltype, transformation, distance, outfile)

print("Simulating detector effects with SNOwGLoBES ...")
snowglobes.simulate(SNOwGLoBES_path, tarredfile, detector_input = detector)

print("Collating results ...")
tables = snowglobes.collate(SNOwGLoBES_path, tarredfile, detector_input = detector,
     skip_plots = True)
# Use results to print the number of events in different interaction channels
key = f"Collated_{outfile}_{detector}_events_smeared_weighted.dat"
total_events = 0
for i, channel in enumerate(tables[key]['header'].split()):
  if i == 0:
    continue
  # Scale to Super-K inner volume (32 kt)
  n_events = 0.32 * sum(tables[key]['data'][i])
  total_events += n_events
  print(f"{channel:10}: {n_events:.3f} events")

print("Total events in Super-K-like detector:", total_events)
```

6.2. Using SNEWPY as Part of the Event Generator sntools

sntools (Migenda et al. 2021) is an event generator that takes neutrino fluxes from supernova simulations and generates a list of resulting neutrino interactions in a detector with a realistic distribution of event time as well as energy and direction of outgoing particles. This list can then be used with a full detector simulation and event reconstruction toolchain for situations in which the approximate treatment of the detector effects in SNOwGLoBES is insufficient. In a recent update, sntools integrated SNEWPY to benefit from the large number of supernova models and flavor transformations it implements.

For example, generating a set of neutrino events in the Hyper-Kamiokande detector (Abe et al. 2018) for the first 500 ms of one of the models included in SNEWPY at a supernova distance of 50 kpc can be done as follows:

```
# install sntools (this automatically installs SNEWPY as a dependency)
pip install sntools
# download supernova model files that are part of SNEWPY
python -c 'import snewpy; snewpy.get_models("Bollig_2016")'
# run sntools using an input file from SNEWPY
sntools SNEWPY_models/Bollig_2016/s27.0c --format SNEWPY-Bollig_2016 --distance 50
  --detector HyperK --starttime 0 --endtime 500
```

SNEWPY's modular design also makes it possible to use its flavor transformations with unsupported input fluxes. The following example shows how to apply SNEWPY's **ThreeFlavorDecoherence** flavor transformation to an input file format that is natively supported by sntools but not by SNEWPY.[25]

```
# download sample input file
curl https://raw.githubusercontent.com/JostMigenda/sntools/v1.0b2/fluxes/intp2001.data
 -o intp2001.data
# run sntools using a flavor transformation from SNEWPY
sntools intp2001.data --format nakazato --detector HyperK --distance 50
  --starttime 0 --endtime 500 --transformation SNEWPY-ThreeFlavorDecoherence
```

[25] This particular file is from the simulations by Nakazato et al. (2013). Note: Unlike the Nakazato_2013 class in `snewpy.models`, which uses reparameterized and reformatted files, sntools natively supports the original file format from http://asphwww.ph.noda.tus.ac.jp/snn/index.html.

At the moment, sntools does not support all supernova models and flavor transformations listed in Sections 3 and 4.2, respectively, since some require additional physical parameters such as sterile neutrino mixing angles. Improvements are expected in future versions of sntools.

7. Summary

The SNEWPY software package connects supernova simulations with detector response software such as SNOwGLoBES and sntools, allowing users to calculate the expected event rates in various neutrino detectors for each model. We expect that SNEWPY will prove useful to modelers and theorists interested in what detectors will observe given some new piece of physics in a simulation, and to experimentalists wishing to evaluate the sensitivity of their detector to supernova neutrinos. In the future we plan to enhance the capabilities of SNEWPY and suggestions from the community about the features we should add are warmly welcome. Modelers interested in adding their simulations to the model library are encouraged to contact us.

This work is supported by the National Science Foundation "Windows on the Universe: the Era of Multi-Messenger Astrophysics" Program: "WoU-MMA: Collaborative Research: A Next-Generation SuperNova Early Warning System for Multimessenger Astronomy" through grant Nos. 1914448, 1914409, 1914447, 1914418, 1914410, 1914416, and 1914426. This work is also supported at NC State by DOE grant DE-FG02-02ER41216, at King's College London by STFC, and at Stockholm University by the Swedish Research Council (Project No. 2020-00452).

Software: Astropy (Astropy Collaboration et al. 2013, 2018), Matplotlib (Hunter 2007), NumPy (Harris et al. 2020), SciPy (Virtanen et al. 2020), SNOwGLoBES (Scholberg 2021), sntools (Migenda et al. 2021).

Appendix
Derivation of the Probabilities

In this appendix we provide the equations for the survival and transition probabilities that appear in Equations (13)–(16) for all 15 flavor transformation prescriptions that SNEWPY currently implements.

A.1. The Extreme Cases

The first two transformations implemented in SNEWPY are NoTransformation and CompleteExchange. For No Transformation, the set of probabilities is given by

$$\begin{aligned} p_{ee} &= 1,\ p_{ex} = 0 \\ p_{xx} &= 1,\ p_{xe} = 0 \\ \bar{p}_{ee} &= 1,\ \bar{p}_{ex} = 0 \\ \bar{p}_{xx} &= 1,\ \bar{p}_{xe} = 0, \end{aligned} \tag{A1}$$

while CompleteExchange corresponds to

$$\begin{aligned} p_{ee} &= 0,\ p_{ex} = 1 \\ p_{xx} &= 0.5,\ p_{xe} = 0.5 \\ \bar{p}_{ee} &= 0,\ \bar{p}_{ex} = 1 \\ \bar{p}_{xx} &= 0.5,\ \bar{p}_{xe} = 0.5. \end{aligned} \tag{A2}$$

A.2. The Three Flavor Mixing Prescriptions

For nontrivial cases of mixing between three active flavors, we only need to compute the elements of the D matrix from the electron flavor row in terms of the vacuum mixing angles θ_{12} and θ_{13}. The expressions in the electron flavor row of the matrix are:

$$D_{e1} = \cos^2\theta_{12}\cos^2\theta_{13}, \tag{A3}$$

$$D_{e2} = \sin^2\theta_{12}\cos^2\theta_{13}, \tag{A4}$$

$$D_{e3} = \sin^2\theta_{13}. \tag{A5}$$

Using Equation (15) from Kneller & McLaughlin (2009), we evaluate the matter mixing angles given in their Equations (16) to (16f) in the limit where the MSW potential becomes large. Using their notation, we find that for NMO, $\tilde{\theta}_{12} \to \pi/2$, $\tilde{\theta}_{13} \to \pi/2$ for neutrinos, while for the antineutrinos $\tilde{\theta}_{12} \to 0$, $\tilde{\theta}_{13} \to 0$. The angle $\tilde{\theta}_{23}$ can be set to zero with no loss of generality if the spectra of μ and τ flavor neutrinos are taken to be equal, because any mixing between them is not observable. In the high-density limit the β and δ phases are also irrelevant and we can pick the Majorana phases α_i to give positive definite values in the U matrix because these phases were shown not to be observable (Galais et al. 2012). Thus we find

$$U = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \bar{U} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{A6}$$

For IMO in the same high-density limit, $\tilde{\theta}_{12} \to \pi/2$, $\tilde{\theta}_{13} \to 0$ for the neutrinos, and $\tilde{\theta}_{12} \to 0$, $\tilde{\theta}_{13} \to \pi/2$ for the antineutrinos. Again, the angle $\tilde{\theta}_{23}$ can be set to zero with no loss of generality and again, in the high-density limit, the phases are also irrelevant. This gives for the IMO

$$U = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \bar{U} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{A7}$$

With U and $\bar{U}$ now defined, we turn our attention to the S matrices. The various prescriptions depend on the details of the flavor transformations.

A.2.1. Adiabatic MSW Effect

The **AdiabaticMSW** transformation assumes adiabatic neutrino evolution for both mass orderings. Given the values of the mixing angle θ_{12}, θ_{13}, and θ_{23} from experiment and typical supernova density profiles, the neutrino propagation through the supernova is adiabatic in the absence of neutrino self-interactions. For adiabatic evolution, both S_M and $\bar{S}_M$ are diagonal unitary matrices

$$S_M = \begin{pmatrix} e^{\imath\xi_{11}} & 0 & 0 \\ 0 & e^{\imath\xi_{22}} & 0 \\ 0 & 0 & e^{\imath\xi_{33}} \end{pmatrix}, \quad \bar{S}_M = \begin{pmatrix} e^{\imath\bar{\xi}_{11}} & 0 & 0 \\ 0 & e^{\imath\bar{\xi}_{22}} & 0 \\ 0 & 0 & e^{\imath\bar{\xi}_{33}} \end{pmatrix}, \tag{A8}$$

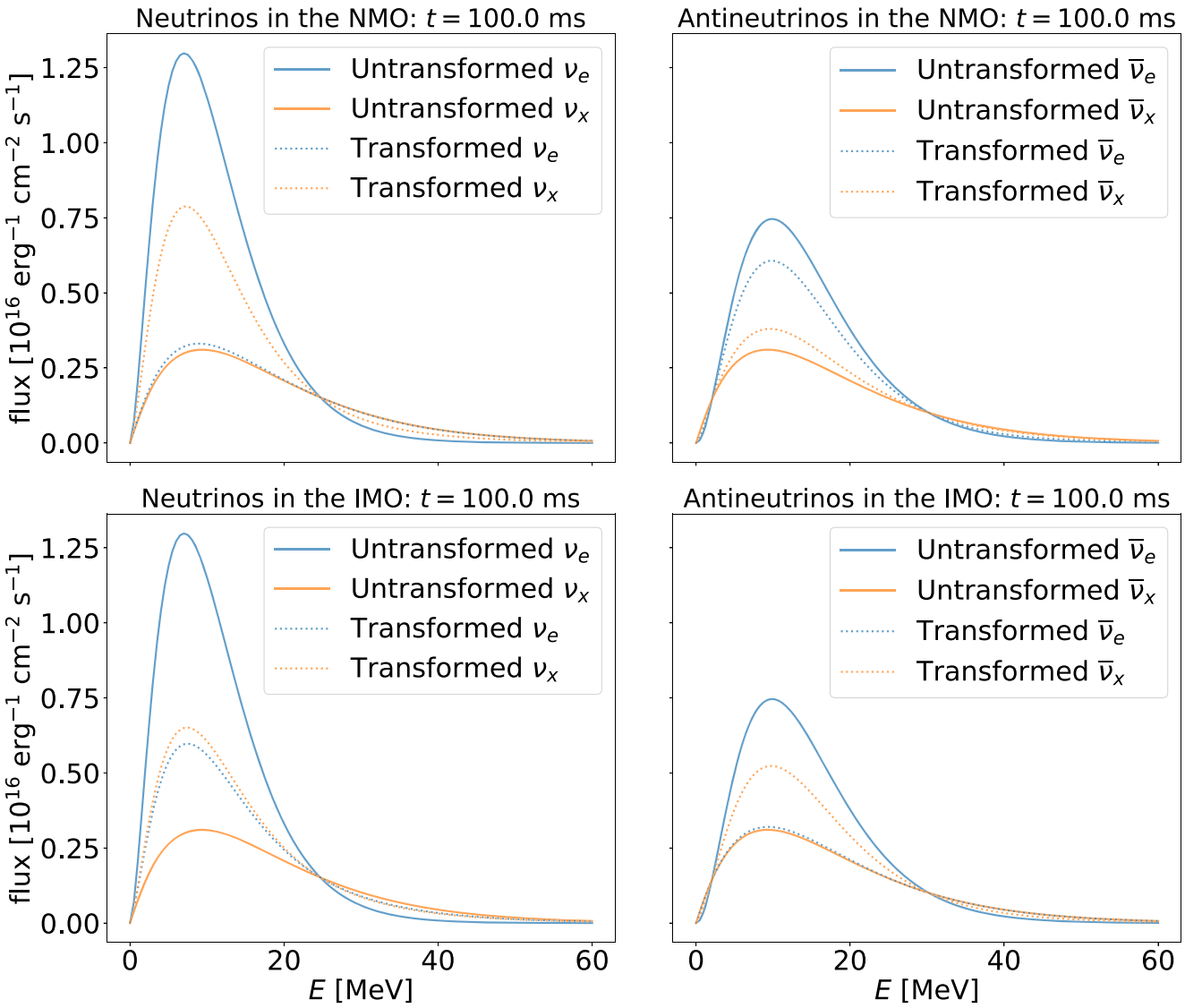

Figure 4. The untransformed (solid lines) and transformed (dashed lines) spectral flux for neutrinos (left panels) and antineutrinos (right panels) using the **AdiabaticMSW** prescription. The NMO is used in the top pair of panels, and the IMO is used in the bottom pair. We show the nakazato-shen-z0.004-trev100ms-s20.0 model (Nakazato et al. 2013) that comes with SNEWPY at the simulation time of 100 ms and at a supernova distance of 10 kpc.

where the ξ's and $\bar{\xi}$ are phases which will turn out not to enter the final formulae. Assuming the NMO, this leads to

$$\begin{aligned} p_{ee} &= D_{e3}, \\ p_{ex} &= 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2, \\ p_{xe} &= (1 - p_{ee})/2 \\ \bar{p}_{ee} &= D_{e1}, \\ \bar{p}_{ex} &= 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2, \\ \bar{p}_{xe} &= (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A9}$$

In the IMO, S_M and $\bar{S}_M$ are again diagonal unitary matrices as in Equations (A8), but the final formulae are different due to the altered structure of the U and $\bar{U}$ matrices at the neutrinosphere. In this mass ordering we find

$$\begin{aligned} p_{ee} &= D_{e2}, \\ p_{ex} &= 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2, \\ p_{xe} &= (1 - p_{ee})/2 \\ \bar{p}_{ee} &= D_{e3}, \\ \bar{p}_{ex} &= 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2, \\ \bar{p}_{xe} &= (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A10}$$

In Figure 4 we show the effect of these two prescriptions on example neutrino spectral fluxes at Earth.

A.2.2. Nonadiabatic MSW Effect

The **NonAdiabaticMSWH** transformation assumes that the H resonance mixing is nonadiabatic while the L resonance is

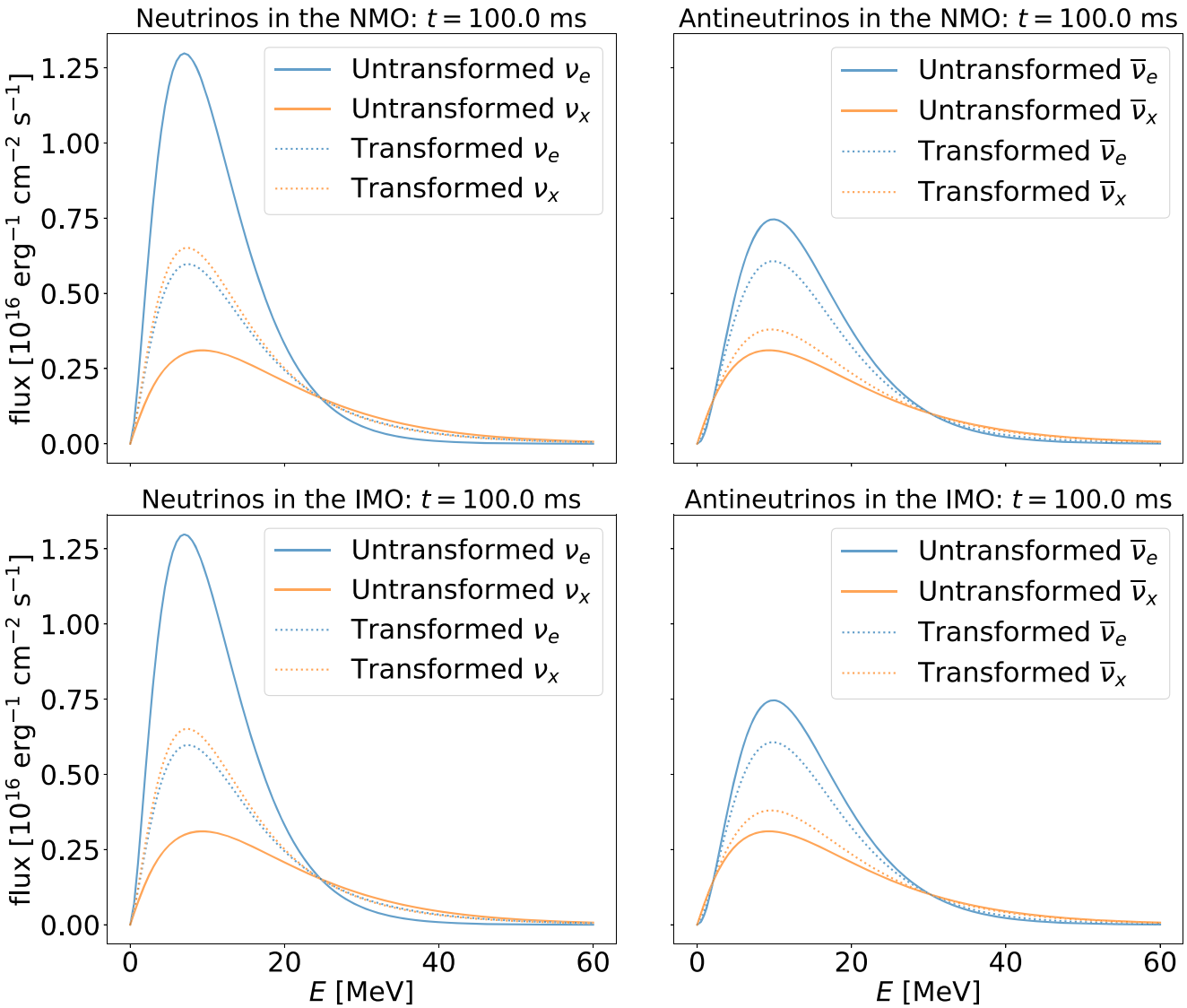

Figure 5. The same as Figure 4 but for the **NonAdiabaticMSWH** flavor transformation prescription.

adiabatic. This case is relevant when a shock is present at the H resonance densities (Schirato & Fuller 2002). For the NMO the H resonance occurs in the neutrinos (Kneller & McLaughlin 2009) between matter states ν_2 and ν_3, which means the matrix S_M is altered while $\bar{S}_m$ has the same diagonal structure as in Equation (A8). The new structure for S_m in the NMO is

$$S_M = \begin{pmatrix} e^{\imath\xi_{11}} & 0 & 0 \\ 0 & 0 & e^{\imath\xi_{23}} \\ 0 & e^{\imath\xi_{32}} & 0 \end{pmatrix}, \tag{A11}$$

and with this new matrix we derive

$$\begin{aligned} p_{ee} &= D_{e2},\ p_{ex} = 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2,\ p_{xe} = (1 - p_{ee})/2 \\ \bar{p}_{ee} &= D_{e1},\ \bar{p}_{ex} = 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2,\ \bar{p}_{xe} = (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A12}$$

In the IMO the H resonance mixes the antineutrino matter states $\bar{\nu}_1$ and $\bar{\nu}_3$. S_M is a diagonal unitary matrix as in Equation (A8) while $\bar{S}_M$ becomes

$$\bar{S}_M = \begin{pmatrix} 0 & 0 & e^{\imath\bar{\xi}_{13}} \\ 0 & e^{\imath\bar{\xi}_{22}} & 0 \\ e^{\imath\bar{\xi}_{31}} & 0 & 0 \end{pmatrix}. \tag{A13}$$

For this case we find

$$\begin{aligned} p_{ee} &= D_{e2},\ p_{ex} = 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2,\ p_{xe} = (1 - p_{ee})/2 \\ \bar{p}_{ee} &= D_{e1},\ \bar{p}_{ex} = 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2,\ \bar{p}_{xe} = (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A14}$$

In Figure 5, we show the effect of these two prescriptions on example neutrino spectra.

A.2.3. Two-flavor Decoherence

The **TwoFlavorDecoherence** transformation is relevant when there is $\lesssim$10% the amount of turbulence in the vicinity of the H resonance—see Kneller (2010) and Kneller & Mauney (2013). This prescription models 50% mixing

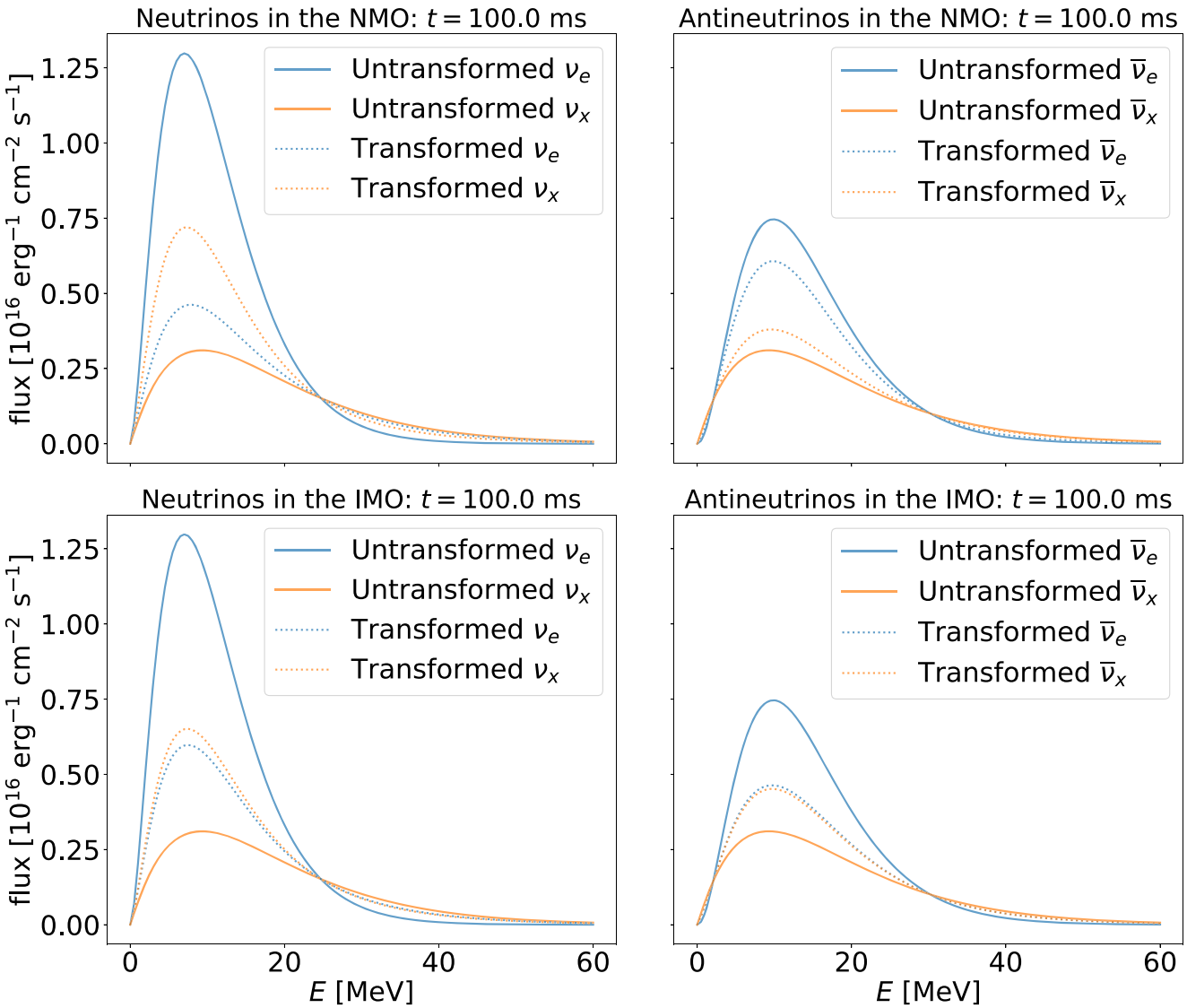

Figure 6. The same as Figure 4 but for the **TwoFlavorDecoherence** flavor transformation prescription.

between the matter states that participate in the H resonance. In the NMO this is ν_2 and ν_3, and S_M is

$$S_M = \frac{1}{\sqrt{2}}\begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & e^{\imath\xi_{22}} & e^{\imath\xi_{23}} \\ 0 & e^{\imath\xi_{32}} & e^{\imath\xi_{33}} \end{pmatrix}, \tag{A15}$$

while $\bar{S}_M$ is again a diagonal unitary matrix in this scenario. Thus we find

$$\begin{aligned} p_{ee} &= (D_{e2} + D_{e3})/2 p_{ex} = 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2 p_{xe} = (1 - p_{ee})/2 \\ \bar{p}_{ee} &= D_{e1}\bar{p}_{ex} = 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2\bar{p}_{xe} = (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A16}$$

For the IMO, the H resonance occurs in the antineutrinos between antineutrino matter states $\bar{\nu}_1$ and $\bar{\nu}_3$, so in this prescription the $\bar{S}_M$ matrix has the general form of

$$\bar{S}_M = \frac{1}{\sqrt{2}}\begin{pmatrix} e^{\imath\xi_{11}} & 0 & e^{\imath\xi_{13}} \\ 0 & \sqrt{2} & 0 \\ e^{\imath\xi_{31}} & 0 & e^{\imath\xi_{33}} \end{pmatrix} \tag{A17}$$

while S_M is diagonal. This leads to

$$\begin{aligned} p_{ee} &= D_{e2} p_{ex} = 1 - p_{ee} \\ p_{xx} &= (1 + p_{ee})/2 p_{xe} = (1 - p_{ee})/2 \\ \bar{p}_{ee} &= (D_{e1} + D_{e3})/2\bar{p}_{ex} = 1 - \bar{p}_{ee} \\ \bar{p}_{xx} &= (1 + \bar{p}_{ee})/2\bar{p}_{xe} = (1 - \bar{p}_{ee})/2. \end{aligned} \tag{A18}$$

In Figure 6 we show the effect of these two prescriptions on example neutrino spectra.

A.2.4. Three-flavor Decoherence

The **ThreeFlavorDecoherence** transformation leads to 33% mixing between all neutrino matter states and antineutrino matter states. Every element of the S_M and $\bar{S}_M$ matrices has a magnitude of $1/\sqrt{3}$. This case is relevant when there are large amounts of turbulence in the vicinity of the H resonance no matter the mass ordering—see Kneller & Mauney (2013). With this structure for the S_m and $\bar{S}_m$ matrices we obtain

$$
\begin{aligned}
p_{ee} &= 1/3 p_{ex} = 1 - p_{ee} \\
p_{xx} &= (1 + p_{ee})/2 p_{xe} = (1 - p_{ee})/2 \\
\bar{p}_{ee} &= 1/3 \bar{p}_{ex} = 1 - \bar{p}_{ee} \\
\bar{p}_{xx} &= (1 + \bar{p}_{ee})/2 \bar{p}_{xe} = (1 - \bar{p}_{ee})/2.
\end{aligned} \tag{A19}
$$

The effect of this prescription on example neutrino spectra was shown in Figure 3.

A.3. Neutrino Decay

SNEWPY also considers two cases of neutrino decay for three active neutrino flavors. The two prescriptions provided with SNEWPY for neutrino decay assume adiabatic evolution through the mantle of the supernova followed by the decay of the heaviest mass neutrino state to the lightest in the vacuum after they have decohered. To account for neutrino decay we must insert another matrix G between the D matrix and the mass state flux vector in Equation (4), i.e., $D \rightarrow D\,G$. The matrix that accounts for the decay of the antineutrinos is the same. Note that we use the approximation that the energy of the neutrino does not change—future versions of SNEWPY will correct this assumption.

If the heaviest neutrino has a mass m and a lifetime τ, we define the inverse decay length as $\Gamma = mc/(E\tau)$. If the distance to the supernova is d, then the flux of the heaviest neutrino mass state decays by the factor $e^{-\Gamma d}$ and the lightest mass state increases by the amount $1 + e^{-\Gamma d}$. The structure of the matrix G depends upon the mass ordering.

NeutrinoDecay: adiabatic evolution through the mantle of the supernova followed by neutrino decay of the heaviest mass state to the lightest without changing the neutrino energy. For the normal mass ordering the G matrix is

$$
G = \begin{pmatrix} 1 & 0 & 1 - e^{-\Gamma d} \\ 0 & 1 & 0 \\ 0 & 0 & e^{-\Gamma d} \end{pmatrix} \tag{A20}
$$

which gives for the NMO:

$$
\begin{aligned}
p_{ee} &= D_{e1}[1 - e^{-\Gamma d}] + D_{e3} e^{-\Gamma d} \\
p_{ex} &= D_{e1} + D_{e2} \\
p_{xx} &= 1 - p_{ex}/2 \\
p_{xe} &= (1 - p_{ee})/2 \\
\bar{p}_{ee} &= D_{e1} \\
\bar{p}_{ex} &= D_{e1}[1 - e^{-\Gamma d}] + D_{e2} + D_{e3} e^{-\Gamma d} \\
\bar{p}_{xx} &= 1 - \bar{p}_{ex}/2 \\
\bar{p}_{xe} &= (1 - \bar{p}_{ee})/2.
\end{aligned} \tag{A21}
$$

For the IMO, the G matrix changes to become

$$
G = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-\Gamma d} & 0 \\ 0 & 1 - e^{-\Gamma d} & 1 \end{pmatrix} \tag{A22}
$$

and using this new matrix we find for the IMO:

$$
\begin{aligned}
p_{ee} &= D_{e2} \exp(-\Gamma d) + D_{e3}[1 - \exp(-\Gamma d)] \\
p_{ex} &= D_{e1} + D_{e3} \\
p_{xx} &= 1 - p_{ex}/2 \\
p_{xe} &= (1 - p_{ee})/2 \\
\bar{p}_{ee} &= D_{e3} \\
\bar{p}_{ex} &= D_{e1} + D_{e2} e^{-\Gamma d} + D_{e3}[1 - e^{-\Gamma d}] \\
\bar{p}_{xx} &= 1 - \bar{p}_{ex}/2 \\
\bar{p}_{xe} &= (1 - \bar{p}_{ee})/2.
\end{aligned} \tag{A23}
$$

In Figure 7, we show the effect of this prescription on example neutrino spectra.

A.4. The Four Neutrino Mixing Prescriptions

We have also included prescriptions for neutrino mixing of three active flavors and a fourth undetectable, sterile flavor. To keep things simple, we have made a number of reasonable assumptions or approximations for these prescriptions. First, we added just one more mixing angle θ_{14} and assumed that the fourth mass eigenstate is the heaviest. In terms of the mixing angles, the required elements of D we will require are:

$$D_{e1} = \cos^2\theta_{12} \cos^2\theta_{13} \cos^2\theta_{14} \tag{A24}$$

$$D_{e2} = \sin^2\theta_{12} \cos^2\theta_{13} \cos^2\theta_{14} \tag{A25}$$

$$D_{e3} = \sin^2\theta_{13} \cos^2\theta_{14} \tag{A26}$$

$$D_{e4} = \sin^2\theta_{14} \tag{A27}$$

$$D_{s1} = \cos^2\theta_{12} \cos^2\theta_{13} \sin^2\theta_{14} \tag{A28}$$

$$D_{s2} = \sin^2\theta_{12} \cos^2\theta_{13} \sin^2\theta_{14} \tag{A29}$$

$$D_{s3} = \sin^2\theta_{13} \sin^2\theta_{14} \tag{A30}$$

$$D_{s4} = \cos^2\theta_{14}. \tag{A31}$$

In these active-sterile mixing scenarios we must consider the effect of the neutral current contribution to the MSW potential which has the effect that the initial density becomes more of an issue in determining the structure of the U and $\bar{U}$ matrices. For an electron fraction $Y_e < 1/3$, the sterile flavor maps to the heaviest matter eigenstate while for $Y_e > 1/3$ it is the electron flavor which maps to the heaviest. However, in supernovae the point where $Y_e = 1/3$ is typically very close to the edge of the protoneutron star where the density gradients are large. The current experimental limits on θ_{14} are such that it is difficult to find a supernova density profile where the $Y = 1/3$ resonance is adiabatic. Thus, we have assumed that the only active-sterile mixing channel that could be either adiabatic or nonadiabatic is the ν_e-ν_s resonance that is set by the mass splitting δm_{14}^2. The MSW "es" resonance is sometimes called the outer "es" resonance or H' resonance (Nunokawa et al. 1997; McLaughlin et al. 1999; Fetter et al. 2003; Beun et al. 2006; Keränen et al. 2007; Esmaili et al. 2014). Any other new resonance introduced by adding the sterile flavor—Esmaili et al. (2014) calls them H'' resonances—is completely nonadiabatic. The

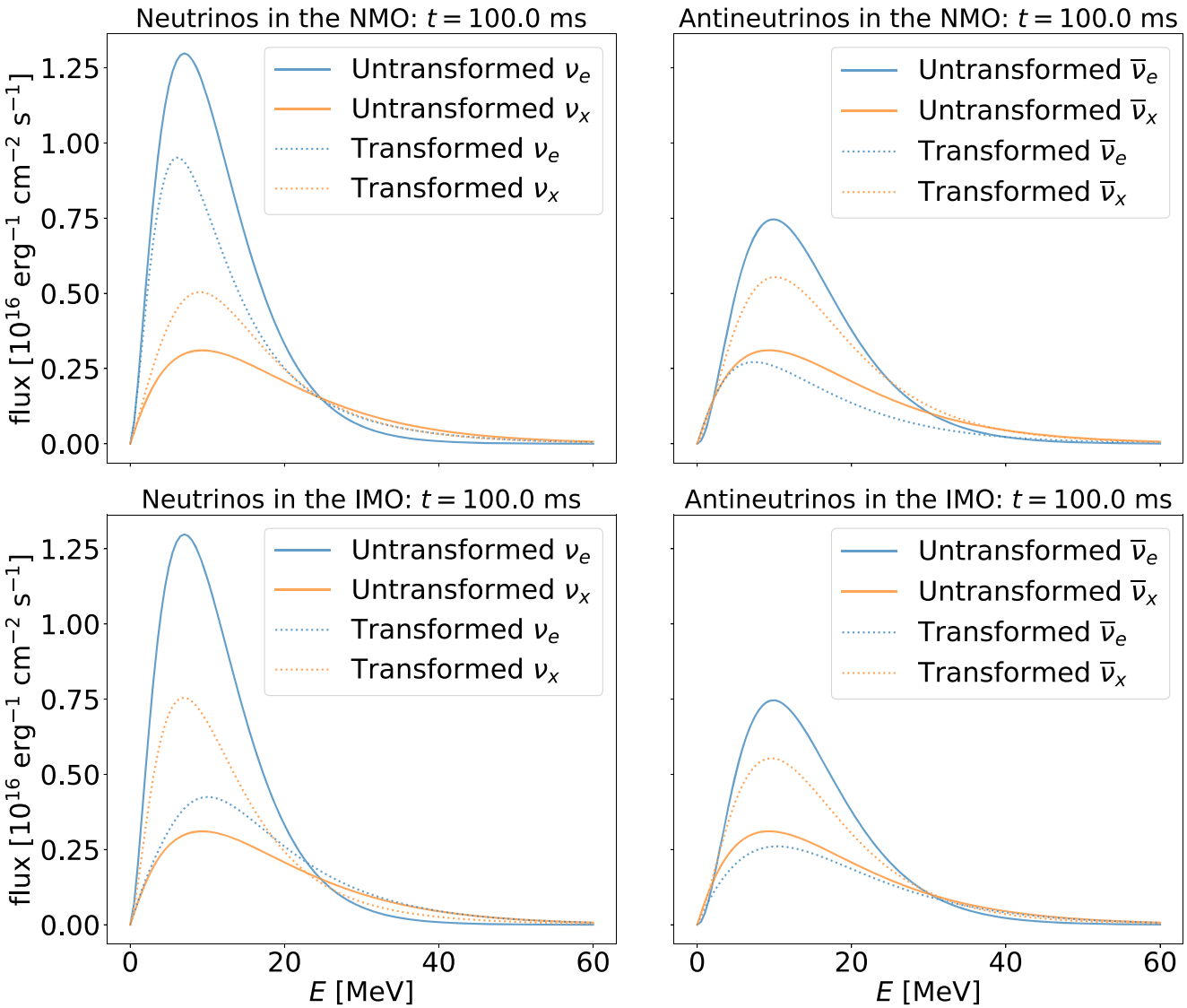

Figure 7. The same as Figure 4 but for the **NeutrinoDecay** flavor transformation prescription. The mass of the neutrino was set to 1 eV and the mean lifetime to 1 day.

mass ordering among the active flavors also needs to be specified, which gives us four cases total.

If we place the neutrinosphere between the $Y_e = 1/3$ and ν_e–ν_s MSW resonance then the structure of the matrices U and $\bar{U}$ for the NMO are

$$U = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \bar{U} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{A32}$$

In the IMO, the matrices are

$$U = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad \bar{U} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{A33}$$

Putting this all together we find the following formulae in the four cases we consider.

A.4.1. AdiabaticMSWes

For the NMO

$$\begin{aligned} p_{ee} &= D_{e4}, \\ p_{ex} &= D_{e1} + D_{e2} \\ p_{xx} &= (2 - D_{e1} - D_{e2} - D_{s1} - D_{s2})/2, \\ p_{xe} &= (1 - D_{e4} - D_{s4})/2 \\ \bar{p}_{ee} &= D_{e1}, \\ \bar{p}_{ex} &= D_{e3} + D_{e4} \\ \bar{p}_{xx} &= (2 - D_{e3} - D_{e4} - D_{s3} - D_{s4})/2, \\ \bar{p}_{xe} &= (1 - D_{e1} - D_{s1})/2. \end{aligned} \tag{A34}$$

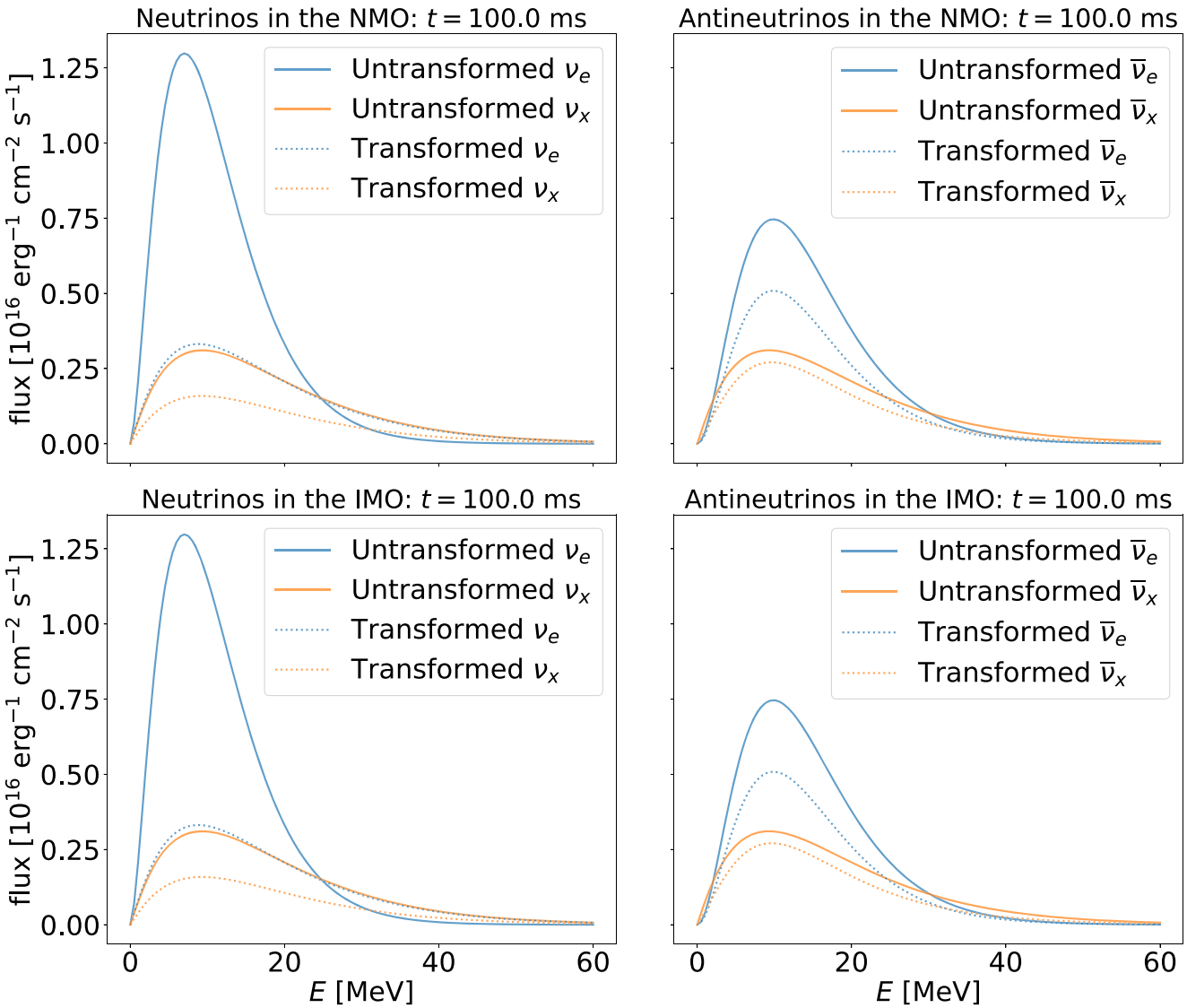

Figure 8. The same as Figure 4 but for the **AdiabaticMSWes** flavor transformation prescription. The mixing angles θ_{12}, θ_{13}, and θ_{23} are set to the values from Zyla et al. (2020), and the additional mixing angle is $\theta_{14} = 10°$.

For the IMO

$$\begin{aligned}
p_{ee} &= D_{e4}, \\
p_{ex} &= D_{e1} + D_{e3} \\
p_{xx} &= (2 - D_{e1} - D_{e3} - D_{s1} - D_{s3})/2, \\
p_{xe} &= (1 - D_{e4} - D_{s4})/2 \\
p_{ee} &= D_{e4}, \\
p_{ex} &= D_{e1} + D_{e3} \\
p_{xx} &= (2 - D_{e1} - D_{e3} - D_{s1} - D_{s3})/2, \\
p_{xe} &= (1 - D_{e4} - D_{s4})/2.
\end{aligned} \tag{A35}$$

In Figure 8, we show the effect of these prescriptions on example neutrino spectra.

A.4.2. NonAdiabaticMSWes

For the NMO, the ν_e-ν_s resonances in the neutrinos, and the two resonances in the antineutrinos—see the eigenvalue diagram in Esmaili et al. (2014)—are all nonadiabatic. The structure of the S and $\bar{S}$ matrices are thus

$$S_M = \begin{pmatrix} e^{\iota\xi_{11}} & 0 & 0 & 0 \\ 0 & e^{\iota\xi_{22}} & 0 & 0 \\ 0 & 0 & 0 & e^{\iota\xi_{34}} \\ 0 & 0 & e^{\iota\xi_{43}} & 0 \end{pmatrix}$$
$$\bar{S}_M = \begin{pmatrix} e^{\iota\xi_{11}} & 0 & 0 & 0 \\ 0 & 0 & e^{\iota\xi_{23}} & 0 \\ 0 & 0 & 0 & e^{\iota\xi_{34}} \\ 0 & e^{\iota\xi_{42}} & 0 & 0 \end{pmatrix}, \tag{A36}$$

which interestingly leads to the following formulae:

$$\begin{aligned}
p_{ee} &= D_{e3}, \\
p_{ex} &= D_{e1} + D_{e2} \\
p_{xx} &= (2 - D_{e1} - D_{e2} - D_{s1} - D_{s2})/2, \\
p_{xe} &= (1 - D_{e3} - D_{s3})/2 \\
\bar{p}_{ee} &= D_{e1}, \\
\bar{p}_{ex} &= D_{e2} + D_{e3} \\
\bar{p}_{xx} &= (2 - D_{e2} - D_{e3} - D_{s2} - D_{s3})/2, \\
\bar{p}_{xe} &= (1 - D_{e1} - D_{s1})/2.
\end{aligned} \tag{A37}$$

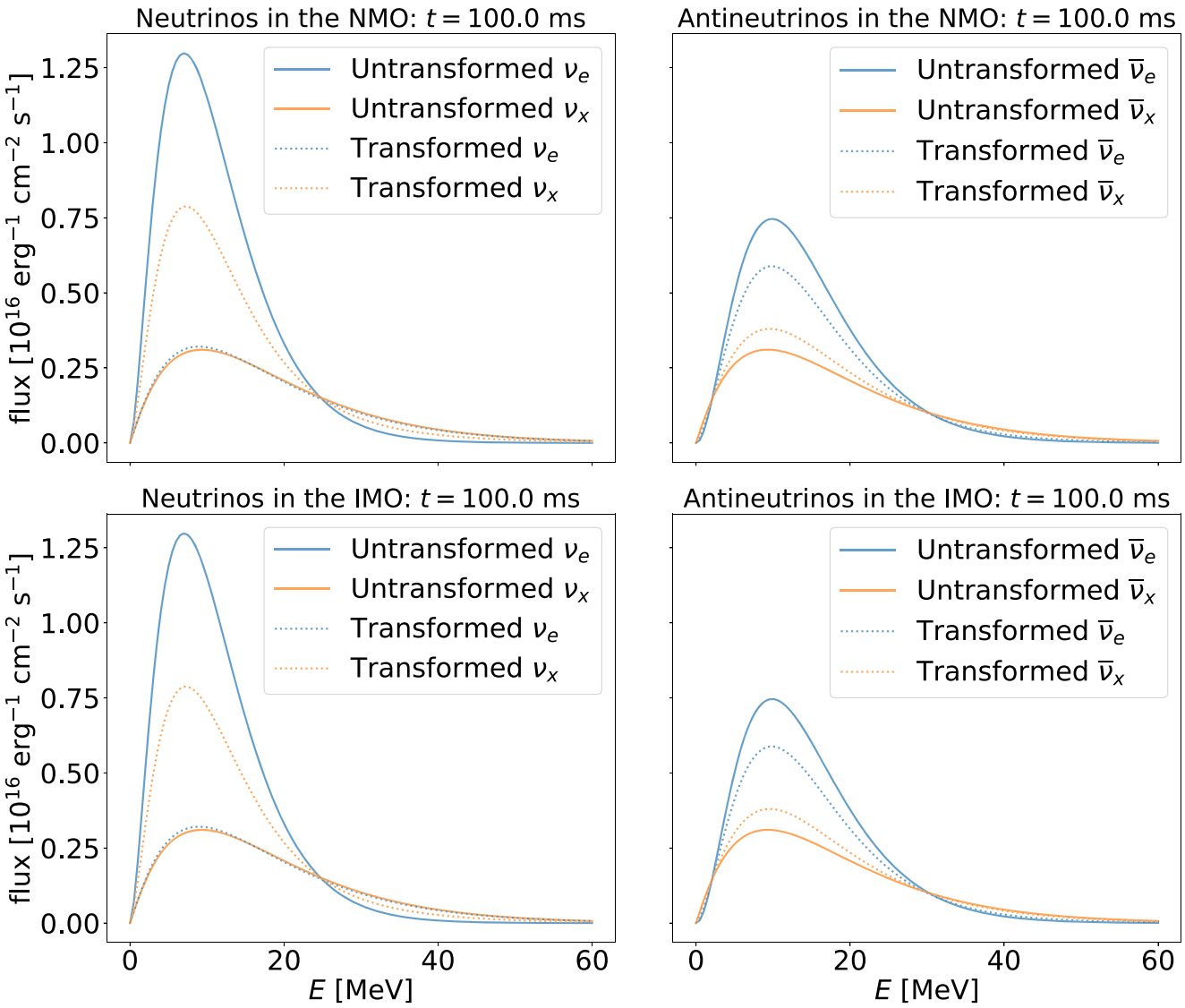

Figure 9. The same as Figure 4 but for the **NonAdiabaticMSWes** flavor transformation prescription. The mixing angles θ_{12}, θ_{13}, and θ_{23} are set to the values from Zyla et al. (2020), and the mixing angle is $\theta_{14} = 10°$.

For the IMO the ν_e-ν_s resonance in the neutrinos and the two resonances in the antineutrinos are all nonadiabatic, and the mass ordering of the active flavors is inverted. The S and $\bar{S}$ matrices are

$$S_M = \begin{pmatrix} e^{\imath\xi_{11}} & 0 & 0 & 0 \\ 0 & 0 & 0 & e^{\imath\xi_{24}} \\ 0 & 0 & e^{\imath\xi_{33}} & 0 \\ 0 & e^{\imath\xi_{42}} & 0 & 0 \end{pmatrix}$$
$$\bar{S}_M = \begin{pmatrix} 0 & e^{\imath\xi_{12}} & 0 & 0 \\ 0 & 0 & 0 & e^{\imath\xi_{24}} \\ 0 & 0 & e^{\imath\xi_{33}} & 0 \\ e^{\imath\xi_{41}} & 0 & 0 & 0 \end{pmatrix}, \tag{A38}$$

which leads to

$$\begin{aligned} p_{ee} &= D_{e2}, \\ p_{ex} &= D_{e1} + D_{e3} \\ p_{xx} &= (2 - D_{e1} - D_{e3} - D_{s1} - D_{s3})/2, \\ p_{xe} &= (1 - D_{e2} - D_{s2})/2 \\ \bar{p}_{ee} &= D_{e3}, \\ \bar{p}_{ex} &= D_{e1} + D_{e2} \\ \bar{p}_{xx} &= (2 - D_{e1} - D_{e2} - D_{s1} - D_{s2})/2, \\ \bar{p}_{xe} &= (1 - D_{e3} - D_{s3})/2 \end{aligned} \tag{A39}$$

for both mass orderings. In Figure 9, we show the effect of these prescriptions on example neutrino spectra.

ORCID iDs

Segev BenZvi https://orcid.org/0000-0001-5537-4710
Marta Colomer Molla https://orcid.org/0000-0003-1801-8121
Spencer Griswold https://orcid.org/0000-0002-7321-7513
Alec Habig https://orcid.org/0000-0002-1018-9383
Shunsaku Horiuchi https://orcid.org/0000-0001-6142-6556
James P. Kneller https://orcid.org/0000-0002-3502-3830
Rafael F. Lang https://orcid.org/0000-0001-7594-2746
Massimiliano Lincetto https://orcid.org/0000-0002-1460-3369
Jost Migenda https://orcid.org/0000-0002-5350-8049
Ko Nakamura https://orcid.org/0000-0002-8734-2147
Evan O'Connor https://orcid.org/0000-0002-8228-796X
Andrew Renshaw https://orcid.org/0000-0003-2913-8057
Kate Scholberg https://orcid.org/0000-0002-7007-2021
Christopher Tunnell https://orcid.org/0000-0001-8158-7795

References

Abbasi, R., Abdou, Y., Abu-Zayyad, T., et al. 2011, A&A, 535, A109
Abe, K., Abe, Ke., Aihara, H., et al. 2018, arXiv:1805.04163
Acciarri, R., Acero, M.-A., Adamowski, M., et al. 2015, arXiv:1512.06148
Acciarri, R., Acero, M.-A., Adamowski, M., et al. 2016a, arXiv:1601.05471
Acciarri, R., Acero, M.-A., Adamowski, M., et al. 2016b, arXiv:1601.02984
Agafonova, N. Y., Aglietta, M., Antonioli, P., et al. 2007, APh, 27, 254
Agafonova, N. Y., Aglietta, M., Antonioli, P., et al. 2008, APh, 28, 516
Aiello, S., Albert, A., Alves Garre, S., et al. 2021, EPJC, 81, 445
Al Kharusi, S., BenZvi, S. Y., Bobowski, J. S., et al. 2021, NJPh, 23 031201
An, F., An, G., An, Q., et al. 2016, JPhG, 43, 030401
Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, AJ, 156, 123
Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33
Baxter, A., BenZvi, S., Jaimes, J., et al. 2021, JOSS, 6, 3772
Beun, J., McLaughlin, G. C., Surman, R., & Hix, W. R. 2006, PhRvD, 73, 093007
Burrows, A., & Vartanyan, D. 2021, Natur, 589, 29
Caballero, O. L., McLaughlin, G. C., & Surman, R. 2009, PhRvD, 80, 123004
Dighe, A. S., & Smirnov, A. Y. 2000, PhRvD, 62, 033007
Duba, C. A., Duncan, F., Farine, J., et al. 2008, JPhCS, 136, 042077
Eguchi, K., Enomoto, S., Furuno, K., et al. 2003, PhRvL, 90, 021802
Esmaili, A., Peres, O. L. G., & Serpico, P. D. 2014, PhRvD, 90, 033013
Esteban, I., Gonzalez-Garcia, M. C., Maltoni, M., Schwetz, T., & Zhou, A. 2020, JHEP, 2020, 178
Fetter, J., McLaughlin, G. C., Balantekin, A. B., & Fuller, G. M. 2003, APh, 18, 433
Galais, S., Kneller, J., & Volpe, C. 2012, JPhG, 39, 035201
Gava, J., Kneller, J., Volpe, C., & McLaughlin, G. C. 2009, PhRvL, 103, 071101
Halzen, F., Jacobsen, J. E., & Zas, E. 1994, PhRvD, 49, 1758
Halzen, F., Jacobsen, J. E., & Zas, E. 1996, PhRvD, 53, 7359
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357
Horiuchi, S., & Kneller, J. P. 2018, JPhG, 45, 043002
Horowitz, C. J., Coakley, K. J., & McKinsey, D. N. 2003, PhRvD, 68, 023005
Hunter, J. D. 2007, CSE, 9, 90
Ikeda, M., Takeda, A., Fukuda, Y., et al. 2007, ApJ, 669, 519
Janka, H.-T., Melson, T., & Summa, A. 2016, ARNPS, 66, 341
Keil, M. T., Raffelt, G. G., & Janka, H.-T. 2003, ApJ, 590, 971
Keränen, P., Maalampi, J., Myyryläinen, M., & Riittinen, J. 2007, PhRvD, 76, 125026
Kistler, M. D., Yüksel, H., Ando, S., Beacom, J. F., & Suzuki, Y. 2011, PhRvD, 83, 123008
Kneller, J. P. 2010, arXiv:1004.1288
Kneller, J. P., & Mauney, A. W. 2013, PhRvD, 88, 045020
Kneller, J. P., & McLaughlin, G. C. 2009, PhRvD, 80, 053002
Kneller, J. P., McLaughlin, G. C., & Brockman, J. 2008, PhRvD, 77, 045023
Kuroda, T. 2021, ApJ, 906, 128
Lai, M. 2021, PhD thesis, Cagliari Univ.
Lang, R. F., McCabe, C., Reichard, S., Selvi, M., & Tamborra, I. 2016, PhRvD, 94, 103009
Lin, Z., & Lunardini, C. 2020, PhRvD, 101, 023016
McLaughlin, G. C., Fetter, J. M., Balantekin, A. B., & Fuller, G. M. 1999, PhRvC, 59, 2873
Migenda, J., Cartwright, S., Kneale, L., et al. 2021, JOSS, 6, 2877
Mirizzi, A., Tamborra, I., Janka, H.-T., et al. 2016, NCimR, 39, 1
Monzani, M. E. 2006, NCimC, 29, 269
Nakazato, K., Sumiyoshi, K., Suzuki, H., et al. 2013, ApJS, 205, 2
Nunokawa, H., Peltoniemi, J. T., Rossi, A., & Valle, J. W. F. 1997, PhRvD, 56, 1704
O'Connor, E. 2015, ApJS, 219, 24
O'Connor, E., & Ott, C. D. 2013, ApJ, 762, 126
Odrzywolek, A., & Plewa, T. 2011, A&A, 529, A156
Rampp, M., & Janka, H. T. 2002, A&A, 396, 361
Reid, G., Adams, J., & Seunarine, S. 2011, PhRvD, 84, 085023
Rosswog, S., & Liebendörfer, M. 2003, MNRAS, 342, 673
Schirato, R. C., & Fuller, G. M. 2002, arXiv:astro-ph/0205390
Scholberg, K. 2012, ARNPS, 62, 81
Scholberg, K., joshuabalbert, Vasel, J., bjfmbeck, & Conley, E. 2021, SNOwGLoBES code: SuperNova Observatories with GLoBES, GitHub, https://github.com/SNOwGLoBES/snowglobes
Shen, H., Toki, H., Oyamatsu, K., & Sumiyoshi, K. 1998, NuPhA, 637, 435
Shen, H., Toki, H., Oyamatsu, K., & Sumiyoshi, K. 2011, ApJS, 197, 20
Skinner, M. A., Dolence, J. C., Burrows, A., Radice, D., & Vartanyan, D. 2019, ApJS, 241, 7
Sukhbold, T., Ertl, T., Woosley, S. E., Brown, J. M., & Janka, H. T. 2016, ApJ, 38
Suzuki, Y. 2001, arXiv:hep-ex/0110005
Tamborra, I., Muller, B., Hudepohl, L., Janka, H.-T., & Raffelt, G. 2012, PhRvD, 86, 125031
Tamborra, I., Raffelt, G., Hanke, F., Janka, H.-T., & Müller, B. 2014, PhRvD, 90, 045032
Vartanyan, D., Burrows, A., Radice, D., Skinner, M. A., & Dolence, J. 2019, MNRAS, 482, 351
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Walk, L., Tamborra, I., Janka, H.-T., & Summa, A. 2018, PhRvD, 98, 123001
Walk, L., Tamborra, I., Janka, H.-T., Summa, A., & Kresse, D. 2020, PhRvD, 101, 123013
Warren, M. L., Couch, S. M., O'Connor, E. P., & Morozova, V. 2020, ApJ, 898, 139
Woosley, S. E., & Heger, A. 2007, PhR, 442, 269
Wright, W. P., Gilmer, M. S., Fröhlich, C., & Kneller, J. P. 2017a, PhRvD, 96, 103008
Wright, W. P., Kneller, J. P., Ohlmann, S. T., et al. 2017b, PhRvD, 95, 043006
Wright, W. P., Nagaraj, G., Kneller, J. P., Scholberg, K., & Seitenzahl, I. R. 2016, PhRvD, 94, 025026
Wu, M.-R., Qian, Y.-Z., Martínez-Pinedo, G., Fischer, T., & Huther, L. 2015, PhRvD, 91, 065016
Zha, S., O'Connor, E. P., & da Silva Schneider, A. 2021, ApJ, 911, 74
Zyla, P., Barnett, R. M., Beringer, J., et al. 2020, PTEP, 2020, 083C01