Computer Physics Communications 277 (2022) 108387

Contents lists available at ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

NuSD: A Geant4 based simulation framework for segmented anti-neutrino detectors ☆,☆☆

Mustafa Kandemir [a,*], Emrah Tiras [b,c], Vincent Fischer [d]

[a] Department of Physics, Recep Tayyip Erdogan University, Rize, 53100, Turkey
[b] Department of Physics, Erciyes University, Kayseri, 38030, Turkey
[c] Department of Physics and Astronomy, The University of Iowa, Iowa City, IA, 52242, USA
[d] Fermi National Accelerator Laboratory, Batavia, IL, 60510, USA

ARTICLE INFO

Article history:
Received 13 January 2022
Received in revised form 7 April 2022
Accepted 24 April 2022
Available online 29 April 2022

Keywords:
NuSD
Geant4 based program
Segmented scintillation detector
Reactor anti-neutrino
Inverse beta decay simulation
Neutrino interaction
Near-field reactor monitoring

ABSTRACT

NuSD: Neutrino Segmented Detector is a Geant4-based user application that simulates inverse beta decay events in a variety of segmented scintillation detectors developed by different international collaborations. This simulation framework uses a combination of cross-programs and libraries including Geant4, ROOT and CLHEP developed and used by the high energy physics community. It will enable the neutrino physics community to simulate and study neutrino interactions within different detector concepts using a single program. In addition to neutrino simulations in segmented detectors, this program can also be used for various research projects utilizing scintillation detectors for different physics applications.

Program summary
Program title: NuSD
CPC Library link to program files: https://doi.org/10.17632/7vgfhxbzdv.1
Licensing provisions: GNU General Public License 3
Programming language: C++
External routines/libraries: Geant4, CLHEP, ROOT, CMAKE
Nature of problem: Considerable effort has been spent on small anti-neutrino detectors for various purposes by different collaborations around the globe, but there is not an interactive and user-friendly joint-simulation framework.
Solution method: To fulfill the need in the field, we developed a simulation framework to combine various segmented detector technologies. This package code will be open to public to let people simulate and test different detector concepts easily. Here, we categorized segmented anti-neutrino detectors as homogeneous and inhomogeneous detectors. Homogeneous detectors consist of PROSPECT, NULAT, and HSP type detectors. Inhomogeneous detectors consist of PANDA type detectors with single scintillator modules. CHANDLER, SOLID and SWEANY detector types are characterized as composite scintillator modules.

1. Introduction

In a global context of growing energy needs and reduction of carbon emission, the use of nuclear energy as an alternative to fossil fuels is expanding and more countries desire or need to acquire the capability to generate nuclear power. In order to limit the possible misuse of civilian nuclear power for military purposes, the International Atomic Energy Agency (IAEA) has been investigating ways to guarantee the peaceful use of nuclear reactors through inspection and monitoring [1].

Emitted in overwhelming numbers by the decay of fission products in nuclear reactors, neutrinos[1] have the unique property of being able to travel almost unhindered through matter, making them impossible to block or tamper with.

☆ The review of this paper was arranged by Prof. Z. Was.
☆☆ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).
* Corresponding author.
E-mail address: mustafa.kandemir@erdogan.edu.tr (M. Kandemir).

[1] In this study, electron anti-neutrinos are referred to as “neutrinos” for brevity.

https://doi.org/10.1016/j.cpc.2022.108387
0010-4655/

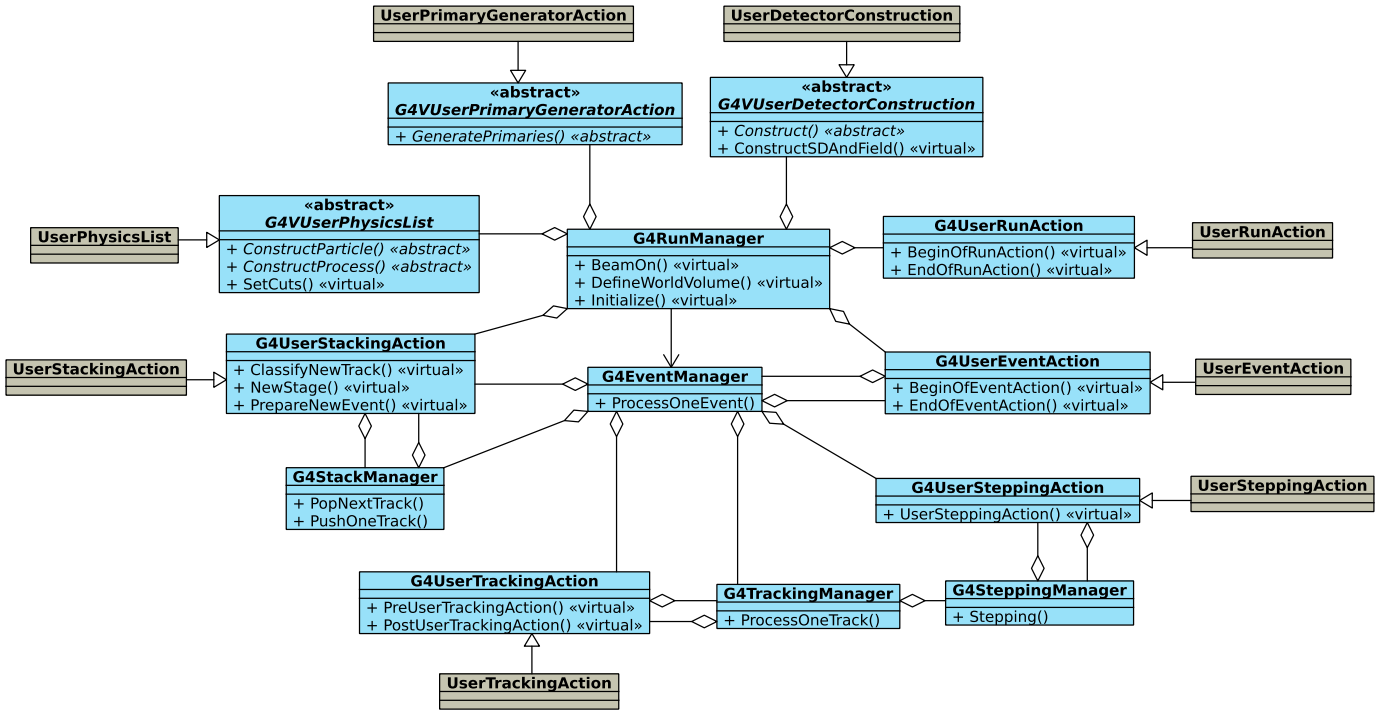

Fig. 1. Unified Modeling Language (UML) class diagram of a typical Geant4 application. The blue color shows classes provided by Geant4, and the brown color indicates concrete classes to be written by the application developers. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

When emitted at low energies (∼1-10 MeV), such neutrinos can be detected through the Inverse Beta Decay reaction (IBD) given its higher cross section on hydrogen [2]. This reaction leads to the simultaneous emission of a positron and a neutron, whose detection can be correlated both in time and in space. A neutrino detector, placed at an optimal distance from a nuclear reactor, would thus have the capability to monitor its neutrino emission flux, directly correlated to the reactor thermal power, and its fissile content in real-time, without encountering the risk of being duped.

However, a detector placed in the vicinity of a nuclear reactor faces several challenges in terms of background and deployment. When deployed within or next to a reactor building, compactness is required in order to fit in existing galleries or rooms and possibly be deployed in several locations in the same reactor complex. Time-correlated and accidental backgrounds from cosmic rays or the reactor itself are likely to mimic a neutrino interaction and bias the estimation of the neutrino flux. Such backgrounds can be mitigated by segmenting the detector volume into smaller subvolumes. This allows a better reconstruction of cosmic rays and their interaction products, as well as a better separation of accidental backgrounds, which are spatially uncorrelated in contrast to neutrino interactions.

As such, while several projects plan on detecting low energy neutrinos through IBD using monolithic water-based detectors [3–6], allowing the reconstruction of Cherenkov patterns, segmented liquid scintillator-based detectors offer a decisive advantage in terms of background reduction and detection efficiency.

Compact segmented scintillation (liquid or plastic) detectors have sparked an interest from the low energy neutrino physics and the nuclear non-proliferation communities as they are well suited to detect neutrino signals in close proximity to nuclear reactors. A global effort is currently ongoing to design and develop several of such detectors, each with their own strengths, weaknesses, and physics goals.

The physics goal of several of the detectors that will be presented and modeled in this study is the detection of short-baseline neutrino oscillations, a possible hint for the existence of additional sterile neutrino states. However, given their design and deployment location, those detectors are well suited to perform measurements of the utmost interest for the nuclear non-proliferation community. This former physics goal requires a good position resolution, hence the segmented design of these detectors, as well as a good energy resolution. While the focus of the following study is set on reactor monitoring applications, the adaptability of the simulation framework presented in the following study allows users to explore physics goals, such as neutrino oscillations, that require accurate event reconstruction in both position and energy.

We believe that a similar effort for simulation studies of such detectors by using a single framework would be useful. In this sense, NuSD brings together numerous reactor neutrino experiments under a single simulation environment, giving the neutrino physics community an easy-to-use, versatile and user-friendly program to explore diverse detector technologies developed by various international collaborations.

2. Implementations

Geant4 [7] is an open-source C++ software package composed of tools that can be used to accurately simulate the passage of particles through matter. The toolset includes all parts of the simulation process, including [8]:

- The geometry of the system and the materials involved
- The particles of interest
- The creation of primary particles
- The tracking of particles through materials
- The physics processes regulating particle interactions
- The response of sensitive detector components
- The creation of event data

Geant4 does not provide any standard program the developer or user can run. A developer must write their own C++ program based on the Geant4 user classes. Fig. 1 presents the software architecture of a typical and simple Geant4 application in the Unified

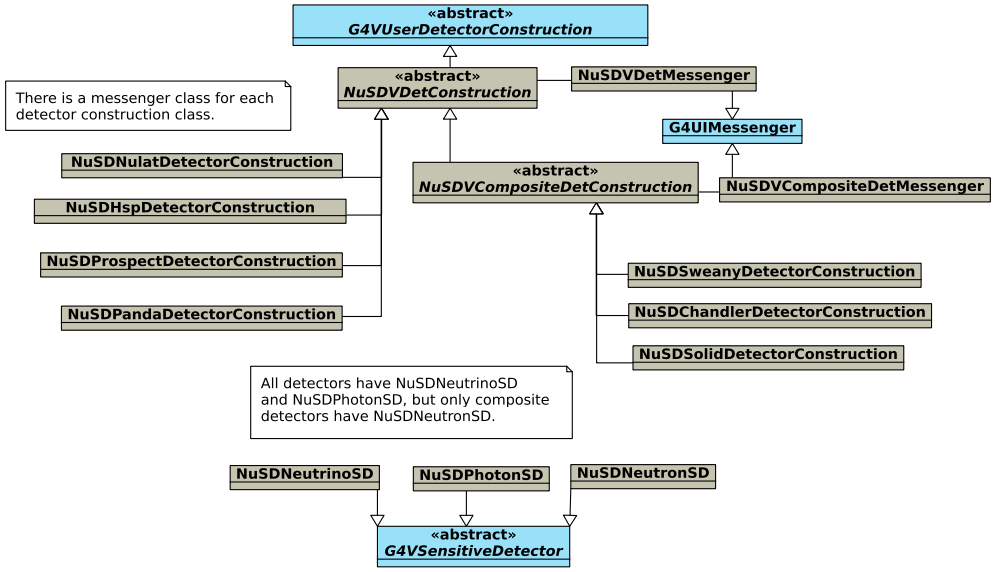

Fig. 2. The UML diagram of the classes involved in building NuSD detectors.

Modeling Language (UML) [9]. The group of classes marked as abstract represents classes that Geant4 developers should implement. Other classes are optional and used to get information and gain control of the simulation at various stages. The classes indicated in brown are the concrete classes to be written by the users.

NuSD is built based on the Geant4 user classes. Apart from the option of utilizing the standard Geant4 API (Application Programming Interface) to implement user classes, NuSD builds an additional API on top of the standard Geant4 API to facilitate the implementation process. To retain flexibility, some of the user classes in NuSD are directly derived from Geant4 user classes, while others are inherited from newly created abstract classes to avoid code redundancy. During the implementation process, the inheritance concept of Object Oriented Programming (OOP) is heavily used to share behaviors between classes. The next sections go through the NuSD implementation in depth.

2.1. Overview of NuSD

NuSD is written in the C++ language and is built based on Geant4 MC code. It requires ROOT [10] and Geant4 software packages to be installed on the system as a prerequisite to work. The code is compiled using CMake and tested on Linux operating system with the version of Geant4 10.06.

NuSD utilizes a main configuration file named *NuSD_config.h* to conditionally compile certain parts of the code. To produce this file, a file named *NuSD_config.h.in* is provided in the source directory. This file contains all of the preprocessor macro definitions used throughout the program. When NuSD is built through CMake, the *NuSD_config.h.in* file is automatically converted to *NuSD_config.h* and the generated file is copied into the build directory. The users can edit this file by commenting out or activating code lines. This file includes the following preprocessor macros:

- #define DETECTOR_NAME HSP. The user chooses one of the defined detectors. HSP is selected by default.
- #define CREATE_ROOT_FILE. The user decides whether to output a ROOT file from the simulation.
- #undef GENERIC_PRIMARY_GENERATOR. The user decides whether to simulate IBD events or any other event. By default, IBD events are simulated.
- #define NuSD_DEBUG. The user decides whether to print the results of the simulation.

NuSD provides many macro files for users to easily modify simulation parameters. A macro file is an ASCII file that contains user interface commands, allowing easy control of the simulation and associated settings, without having to interact with the source code of the program. Thus, a NuSD user only needs basic knowledge of how to control the program flow but does not necessarily have to know object-oriented programming or C++.

NuSD works in both sequential and multi-threading modes. It is very critical to be able to run in multi-threading mode, particularly in cases where optical photons are simulated. This is because when a particle deposits its energy in a scintillator, tens of thousands of photons are produced, and tracking them across the detector requires a significant amount of CPU time. As a result, on machines with numerous cores or CPUs, running in multi-threading mode dramatically improves simulation speed.

NuSD can be run in two different modes, one in interactive mode with visualization and the other in batch mode without visualization. The former is typically used in the preliminary stages of the simulation to verify the detector set-up, the model, and the typical outcome of an event. On the other hand, the latter is much faster and is generally used to accumulate statistics.

NuSD is written in such a way that it can be easily extended. The intent here is to easily incorporate a new detector concept into the program. Furthermore, the code is written in a straightforward approach without using advanced topics of C++ that anyone with basic knowledge of C++ and OOP can readily understand and make changes on it.

2.2. Construction of NuSD detectors

NuSD currently models seven different detector concepts based on the respective experiments. In addition to the G4VUserDetectorConstruction class provided by Geant4, NuSD introduces two new intermediate abstract detector construction classes, NuSDVDetConstruction and NuSDVCompositeDetConstruction, to derive a NuSD detector. The former is inherited from the G4VUserDetectorConstruction class and is used to create detector types that use a single scintillator. The latter is inherited from the first one and is used to construct composite detector types.

These two abstract classes contain common attributes and implement common functionality of all derived detector construction classes. Fig. 2 depicts the UML diagram of a detector construction process in NuSD.

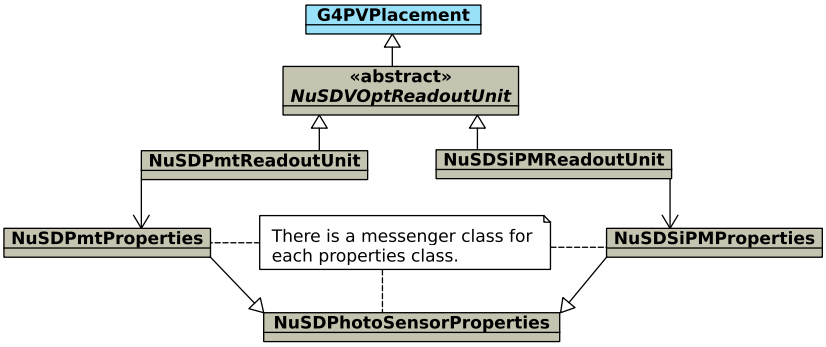

Fig. 3. The UML diagram of the classes involved in building a major detector component. In this example figure, it is an optical readout unit.

A NuSD detector roughly consists of a main volume, which is in the form of a *scintillation lattice*, and a number of identical optical readout units surrounding it. Each readout unit contains a photosensor, and optionally a light guide and accessory elements that act as glue between the scintillator and light guide or the scintillator and the photosensor.

To construct a detector in NuSD, the smallest repeating unit of the main volume of that detector is first created. Then, in a loop, these unit volumes are positioned in a certain geometrical arrangement within a mother volume that represents the entire volume of that detector. In this way, the main part of the detector is formed. The optical readout system can also be installed together with the main volume within the same loop, according to the user's request.

NuSD detectors differ from each other in various aspects. First, each detector type has its own distinct repeating units, and the techniques for combining these units may differ. The differences can be grouped under six headings:

- Optical readout system: PMT and SiPM.
- Neutron converter materials. Different reactions for neutron detection. Neutrons are converted into charged particles via Li-6 or gamma rays with Gadolinium (Gd).
- The method of incorporating neutron converter materials into detectors.
- Detector segmentation: both physical and optical.
- Type of scintillators: plastic and liquid.
- Signal identification techniques: pulse shape discrimination with appropriate material selection and topological selection cuts via segmentation.

NuSD uses two different ways to construct a detector component. If a component is small, that is, only a few lines of code are required to produce it, then it is created in its own detector construction class. If it is a major component that necessitates a significant amount of code, a new class is derived from the G4PVPlacement, and all steps connected to this component are implemented in this new class. As an example, Fig. 3 depicts the classes used to construct an optical readout unit, a key component common to all detectors, and their relationship in UML notation.

All NuSD detectors are constructed as described above. Therefore, if a new detector concept is to be incorporated into NuSD, using a similar approach is recommended in terms of compatibility.

Since each detector type may have its unique set of parameters, a messenger class is created for each detector construction class, inheriting from the G4UIMessenger. While the NuSDVDetMessenger class contains commands common to all detectors, the NuSDVCompositeDetMessenger class contains commands that are specific to composite detectors. All of the commands defined in a detector messenger class are kept in an ASCII file with the same name as that of the detector (i.e. Chandler.mac, Solid.mac...). These macro files allow a selected detector type to be set up in a desired geometric configuration. The following lists some critical parameters that can be adjusted when setting up a NuSD detector:

- Number of segments along x, y and z.
- Dimension of the neutrino scintillator.
- Thickness of coating and container materials. For example, neutron converter sheets, optical barrier materials, reflectors, and aluminium tanks.
- Dimensions of light guide and accessory elements (such as cement and grease).
- A flag to add/remove the optical readout system.
- A few true/false flags to include/remove some detector components.
- A flag to indicate whether photosensors will be installed on either side of the detector.
- Parameters related to the photosensors (PMT or SiPM) and reflector.

In NuSD, the scintillator that detects neutrinos is referred to as a neutrino scintillator, whereas the scintillator that detects neutrons is referred to as a neutron scintillator. Neutron scintillators are only available in composite detectors, whereas neutrino scintillators are employed in all detector types.

The sensitive detector class in Geant4 has the task of creating hits each time a particle traverses a sensitive volume. To retrieve information from the simulation, the user writes a concrete sensitive detector class, inherited from G4VSensitiveDetector, that produces hits, and attaches it to the volume in the detector geometry from which the information is desired.

NuSD utilizes the sensitive detector approach to retrieve information from the following three components:

- NuSDNeutrinoSD: is attached to the neutrino scintillator volume of the detector.
- NuSDNeutronSD: is created only for composite detectors and is attached to the neutron scintillator portion of the detector.
- NuSDPhotonSD: is attached to the photon detection area of the optical readout system.

As an important point, NuSDPhotonSD cannot be triggered like a normal sensitive detector since it does not allow photons to pass through it. To trigger it, the status of the G4OpBoundary is monitored whenever a photon hits the photo-sensitive area of a photosensor (i.e., photocathode or mppc). If the status is 'Detection', the NuSDPhotonSD object is retrieved from the G4SDManager and its ProcessHits function is called manually.

2.3. NuSD materials

Material definitions in Geant4 can be done in two alternative ways. The first one utilizes the G4NistManager class, which implements an interface to the National Institute of Standards and Technology (NIST) material database, allowing users to retrieve predefined isotopes, elements, and materials. Alternatively, the users can build their own custom materials from scratch using the provided G4Element, G4Isotope, and G4Material classes.

NuSD uses both approaches for material definition through the use of a singleton class named NuSDMaterials. It keeps a pointer to the G4NistManager and has a public method GetMaterial that takes a single parameter specifying the name of the material. This method internally invokes the FindOrBuildMaterial method of G4NistManager and/or GetMaterial method of G4Material class. All materials utilized in the simulation are defined within the NuSDMaterials, and when a material is needed for a detector component, the GetMaterial method of the NuSDMaterials is called from the corresponding detector construction class. In this manner, materials common to distinct detectors are defined only once, rather than being defined repeatedly in each detector construction class.

The materials that make up the components of NuSD detectors, as well as their functions in a detector, are outlined below:

- Several commercially available scintillators from Eljen Technology® such as EJ-200, EJ-260, EJ-335, and EJ-390 and two custom-made scintillators developed for Prospect and Nulat experiments, Custom-EJ-309 and Custom-EJ-254. These are referred to as neutrino scintillators in NuSD, and a neutrino scintillator is the primary component of a neutrino detector.
- Thermal neutron scintillator screen. It is composed of a matrix of ZnS:Ag and Li-6. In addition to a neutrino scintillator, composite detectors employ a neutron scintillator. It converts neutrons into charged particles.
- Gadolinium-coated Mylar films. It converts neutrons into gamma rays.
- Optical barrier materials such as air, FEP (Fluorinated ethylene propylene), and water. The main function of these materials is to direct the photons emitted from the scintillator towards the light collectors through total internal reflection. Therefore, materials with a low refractive index are preferred.
- Reflector sheets such as 3M reflector film and aluminized mylar film. These are used to improve light collection and also for optical segmentation.
- PMMA (polymethylmethacrylate), also known as acrylic, acrylic glass, perspex, or plexiglass. Light guides are fabricated from acrylic materials. A light guide is generally used to improve energy resolution of a detector.
- Accessory elements such as cement and grease. The former is used to bind a scintillator to a light guide, while the latter couples a scintillator directly to a photosensor.
- Containter materials such as a stainless steel or acrylic tank.

These are the common materials used to construct a complete neutrino detector. The users can easily define new materials in the same way as in the NuSDMaterials class.

A material is defined by specifying its density, the number of components, and the fraction of each component that makes up the material. The specification of the isotope composition of the material is especially significant for the hadronic interactions since the hadronic cross-section is a strong function of the nuclear property of the material. If the physics processes of a simulation do not involve optical interactions (or the optical processes are disabled by the user), no additional material information is required. If it involves optical properties of the medium, which are key to the implementation of the optical processes, should be defined.

To accurately simulate optical photons, the user must define all material and surface properties. Many parameters are provided by Geant4 to accomplish this. The definitions of these parameters and how these parameters are implemented are explained in detail in the Application Developer's Guide [11].

To briefly summarize, Geant4 provides the G4MaterialProperties table class to impart new properties to a pre-defined material or a surface. For this, an instance of G4MaterialPropertiesTable is created, and the desired number of properties is added to the table as entries. This table showing properties is then linked to the material or surface in question. Each property in the table may be independent of energy (denoted as "Constants") or it may be expressed as a function of the photon energy. If a property does not depend on energy, it has a single value. If it depends on energy, it contains a list of energy-value pairs.

NuSD utilizes a set of text files to store the energy-value pairs of energy-dependent quantities. The text files can be arranged in either one or two columns. The single-column text files include two discrete energy values that indicate the energy response range of the interested quantitiy. In this case, the same default value is used at all energies. Using the same value at all photon energies for a time-dependent quantity allows for directly measuring the impact of that property on the interested quantity. On the other hand, the two-column text files contain a list of energy-value pairs. A NuSD user first decides whether a property depends on energy or not, and then edits the relevant text file accordingly. In this way, a NuSD user can run the simulation program using his/her own experimental input parameters.

The provided text files are kept in three separate folders that are grouped according to their contents:

- Quantum efficiency spectra of the used photosensors (photocathode for PMT, and silicon microcells for SiPM/MCPP). NuSD uses the NuSDPhotoSensorSurface class for the implementation of a photo sensor surface.
- Reflectivity spectra of the used reflectors. NuSD uses the NuSDReflectorSurface class to implement a reflector surface.
- Refractive indices and absorption lengths of the materials, as well as the photon emission spectra of the used scintillators.

These three folders are stored under a parent directory called *data*, which is in the same path as the source code of NuSD. It is the user's responsibility to research and then define these experimental input values that are very crucial for accurate simulation. For example, Eljen technology provides data sheets for the scintillators it produces. A user can extract the relevant information from the supplied data sheets by the manufacturers.

2.4. Physics processes

Physics processes describe how particles interact with materials. Geant4 uses the concept of physics lists to handle the physics of the simulation. A physics list is a collection of physics processes, models, and cross-sections that are used to model the process of particle passage through matter.

There are three ways to build a physics list in Geant4. In the first case, the user has to manually specify all the particles to be used in the simulation as well as the physics processes assigned to each particle. Although it provides a flexible way to set up the physics environment, it requires deep knowledge of the physics of the target application. The second one is the usage of the "physics constructor" which allows grouping particles and their processes according to a physics domain. A physics constructor handles a well-defined category of physics (e.g. EM physics, hadronic physics, decay, etc). The third option is the "reference physics lists" which cover all the physics interactions that take place in the simula-

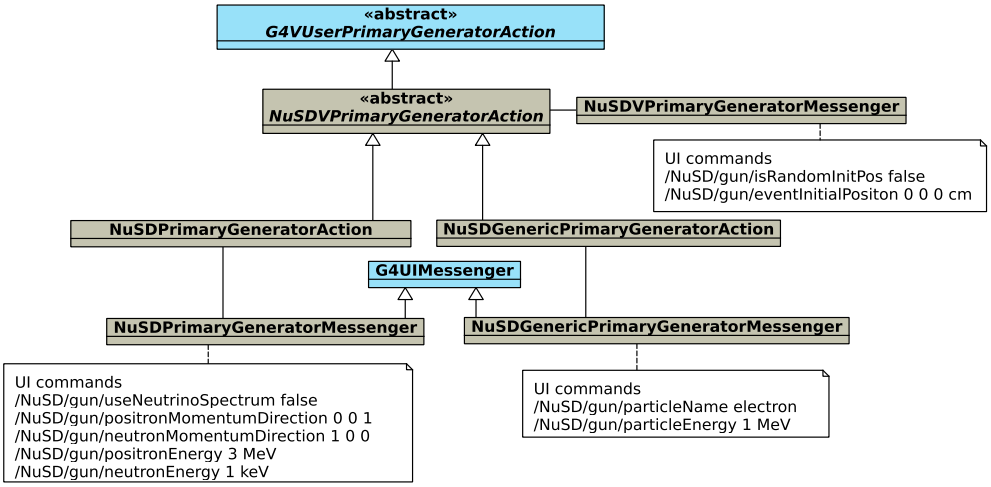

Fig. 4. The UML diagrams of the classes involved in generating primary vertex and primary particles. If the command */NuSD/gun/useNeutrinoSpectrum* is set to true, the commands below it are ignored.

tion. These are ready-to-use complete physics lists provided by the toolkit and are routinely validated and updated with each release. Geant4 provides several reference physics lists that are suited for different use cases.

NuSD uses QGSP-BERT-HP/QGSP-BIC-HP ("HP" for high-precision neutron interactions) reference physics list for the simulation of IBD events. The QGSP-BERT-HP includes standard electromagnetic and hadronic physics processes and is specifically suited for the transport of neutrons below 20 MeV down to thermal energies [12]. This modular physics list covers all the well-known processes such as ionization, Coulomb scattering, Bremsstrahlung, Compton scattering, photoelectric effect, pair production, annihilation, decay, radiative capture, fission, hadronic elastic, and inelastic scattering. This reference physics list does not include optical processes. To include them, G4OpticalPhysics, the constructor of optical processes, is instantiated and then registered in the list. This automatically adds all the optical physics processes.

2.5. Primary particles

NuSD provides two different primary particle generation classes: NuSDPrimaryGeneratorAction and NuSDGenericPrimaryGeneratorAction. The first one is used to generate IBD events, while the second one is used to simulate background events or any desired particle. These two classes are inherited from the NuSDVPrimaryGeneratorAction abstract class and are used to identify the primary particles to be generated as well as setting their initial values for energy, momentum, position, and time. The NuSDVPrimaryGeneratorAction class itself is derived from the G4VUserPrimaryGeneratorAction class and includes some additional member functions for generating random positions within the active volume of NuSD detectors. Fig. 4 shows the UML diagrams of the primary generator classes used by NuSD.

The initial position of an event can be specified in one of two ways in NuSD. The first one is to generate an event at a random point inside the detector, and the second one is to generate an event at a user-specified position. In the first option, which is accomplished by setting the command /NuSD/gun/isRandomInitPos to true, a random unit is first selected, then a random point is generated within that unit. If the command is set to false, a random unit is selected again, but this time the position is specified relative to the selected unit's center via the command /NuSD/gun/eventInitialPositon.

NuSD generates an IBD event by directly creating its products as primary particles due to the neutrino's extremely low interaction cross-section. By default, NuSD simulates reactor neutrinos (1-10 MeV energy range) and performs the following steps to determine the initial energy of the positron and neutron:

- Eq. (1) is used to sample neutrino energies. Here $P(E_{\overline{\nu}})$ is the detecting probability of a reactor neutrino at energy $E_{\overline{\nu}}$, $\Phi_i(E_{\overline{\nu}})$ is the emitted reactor neutrino energy spectrum per fission of each of the four isotopes, $\sigma(E_{\overline{\nu}})$ is the neutrino-proton interaction cross section at zeroth-order, α_i is the average fission fraction of each isotope over the reactor fuel cycle, and N is the normalization constant.
- Eq. (2) is used to sample the cosine of the scatterring positron angle [2].
- Using the ROOT analysis framework, two histograms, one for the initial neutrino energy and the other for the scattering positron angle, are generated and written into a ROOT file.
- The NuSDPrimaryGeneratorAction class uses this ROOT file to get a neutrino energy and a positron scattering angle from the written histograms for each event.
- The positron energy E_{e^+} is computed from Eq. (3). Here m_e is the electron mass, M is the neutron mass, and the $\triangle$ is the mass difference of the neutron and proton.
- The neutron energy T_n is calculated from Eq. (4).

$$P(E_{\overline{\nu}}) = \frac{1}{N} \sum_{i=5,8,9,1} \alpha_i \Phi_i(E_{\overline{\nu}}) \sigma(E_{\overline{\nu}}) \tag{1}$$

$$P(\cos\theta) = \frac{1}{N}\left[1 - 0.1\cos\theta\right] \tag{2}$$

$$E_{e^+}^{(1)} = E_{e^+}^{(0)}\left[1 - \frac{E_\nu}{M}(1-\cos\theta)\right] - \frac{\triangle^2 - m_e^2}{2M} \tag{3}$$

$$T_n = \frac{E_\nu\,(E_\nu - \triangle)}{M}(1-\cos\theta) + \frac{\triangle^2 - m_e^2}{2M} \tag{4}$$

As an alternative to the above, a NuSD user can manually set the parameters of the positron and neutron with the supplied user interface commands (see Fig. 4).

The NuSDGenericPrimaryGeneratorAction class is implemented to simulate electrons by default. The user can alter the type of particle and its initial setting using the supplied user commands.

Fig. 5 shows the initial energy distributions of the positron and neutron at the top, and their scattering angle distribution at the bottom. From the bottom of the Figure, it can be seen that the average positron direction is slightly backward and the neutron direction is purely forward.

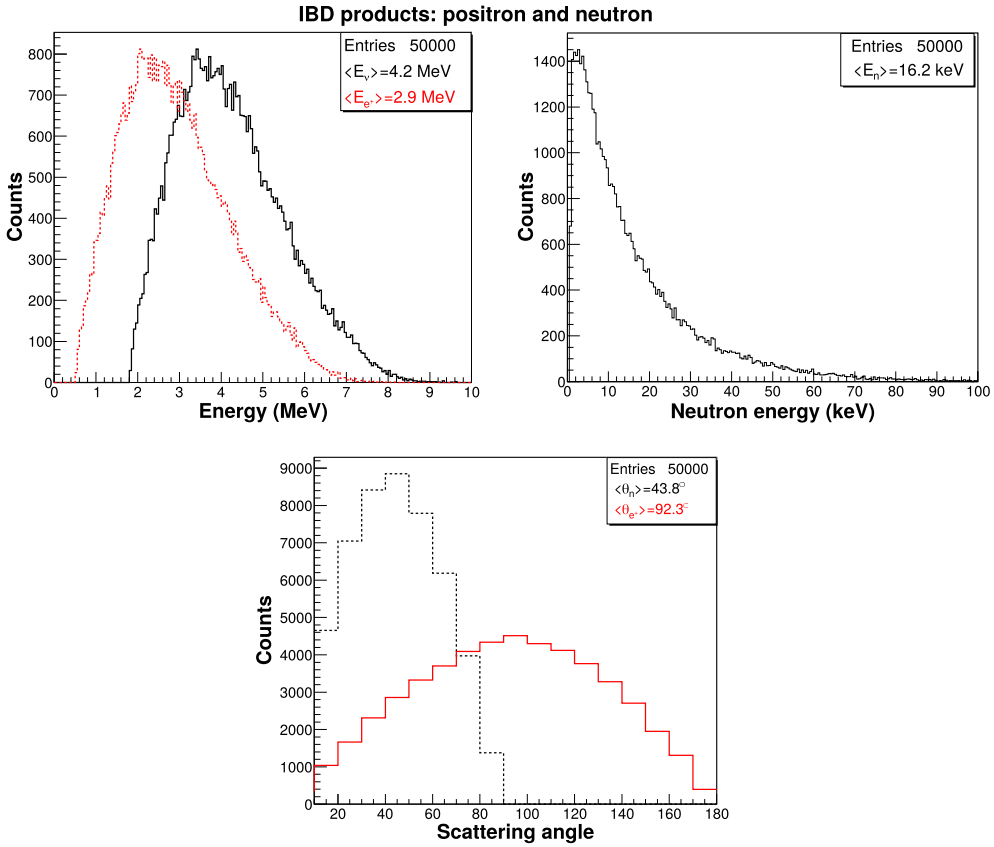

Fig. 5. The initial energy distributions of the positron (top-left) and neutron (top-right). The scattering angle distribution of the positron and neutron (bottom).

2.6. Output of NuSD

To extract information from the simulation, Geant4 provides the g4tools package and analysis manager classes that can be used to write data in several formats (ROOT, AIDA XML, CSV and HBOOK). NuSD prefers ROOT output technology since it is well-documented and is the most widely used tool for data analysis in the High Energy Physics community.

NuSD collects information broadly in two ways. Firstly, through the use of sensitive detector classes (NuSDNeutrinoSD, NuSDNeutronSD, and NuSDPhotonSD) that generates hits, which contains some desired information. The information acquired from the sensitive detectors is common for both IBD and single-particle simulation. Secondly, via user action classes, which give users complete control over the simulation and access information from various stages of the simulation.

Second, through user action classes that give users full control over the simulation and access information from the various stages of the simulation.

Both methods mentioned above are handled by the use of two singleton analysis manager classes, NuSDAnalysisManager and NuSDGenericAnalysisManager. The former is used for the outputs of the IBD events, while the latter is used for any other events to be simulated. The users of NuSD can rearrange the NuSDGenericAnalysisManager class depending on the desired outputs.

The following physics quantities are obtained from the IBD on an event-by-event basis, and each is recorded in a different column of an Ntuple.

- Event ID
- Antineutrino energy that initiates the IBD event, the initial energy of the IBD products (i.e. positron and neutron), and the momentum direction of these three.
- Event vertex position (3D vector in Cartesian coordinates).
- Neutron capture position (3D vector in Cartesian coordinates).
- Neutron capture time.
- Mass number of the neutron-capturing nucleus.
- Number of emitted scintillation photons.
- Sensitive detectors create hits from each step and store them into a hits collection. A hit object created by the NuSDNeutrinoSD or NuSDNeutronSD has the following attributes: particle PDG code, track ID, copy number, deposited energy, and energy deposition time. At the end of the event, vector is created for each member of the hit.
- The hit object created by the NuSDPhotonSD for each detected photon has two members: the detection time of the photon and the copy number of the PMT/SiPM that detects that photon. At the end of the event, a std::vector is created for both members of the hit.

Although NuSD is not an analysis framework, it provides several ready-to-use ROOT macro files to analyze raw data obtained from the simulations. These macro files plot the simulation data and calculate some important quantities that cannot be obtained directly from the simulations. Some of those are:

- Total deposited energy in each segment of the detector. In other words, an energy deposition map of a detector.
- Number of detected photons by each photosensor in an event.
- Neutron capture efficiency of nuclei such as hydrogen, carbon, Li-6, and Gd isotopes.
- Mean neutron capture time.
- Antineutrino detection efficiency.
- Light collection efficiency.

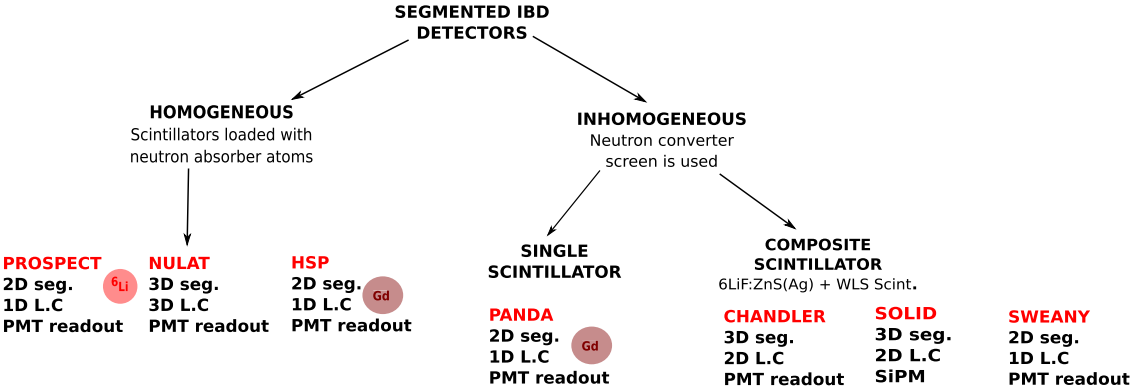

Fig. 6. Detector classification based on the selection of scintillation material and the application method for neutron detection.

The provided macro files enable users to apply several selection cuts when calculating a quantity. For example, cuts such as the amount of deposited energy and its deposition time, and detected photon number and its detected time can be applied.

3. NuSD workflow

The following summarizes the main steps a user follows when running a NuSD application:

- When the program is built, a file entitled *NuSD_config.h* is generated in the build directory. This file contains a few settings that are changed rarely. Using this file, configure the initial simulation settings that are explained in Sec. 2.1.
- There is a macro file for each detector type under the folder "macros" in the build directory. Using the relevant macro file, set up the selected detector with the desired geometric layout.
- Next, start NuSD in interactive mode to visualize the installed detector and then monitor the event to be simulated by pressing the run button. This mode is typically used at the initial stage of the simulation for the validity of the detector setup and to see the behavior of particle tracks at the surfaces. For example, one can reveal the number of photons hit on a photosensor or the number of photons absorbed by an optical surface by watching photons on those surfaces with the naked eye. Further, an event summary will be printed on the terminal at the end of the event if the NuSD_DEBUG macro is enabled. The geometry can also be altered at this stage (Idle state) using the supplied interface available on the panel.
- Once the validation stage is successfully completed, run the program with the desired number of events in batch mode to accumulate statistics. Working in multi-threading mode is recommended if optical photons are simulated.
- When the program has finished running, ROOT files will be created automatically if the CREATE_ROOT_FILE macro is enabled.
- Using the supplied ROOT files, make your analysis. Many analysis ROOT macros have already been provided for users.

4. Results

This section is divided into three parts. The first part briefly introduces and categorizes the detectors modeled using NuSD. The second part presents some simulation results over a selected detector type. The final part identifies a few key design evaluation metrics that are critical for all detector types, and findings are reported using these quantitative metrics.

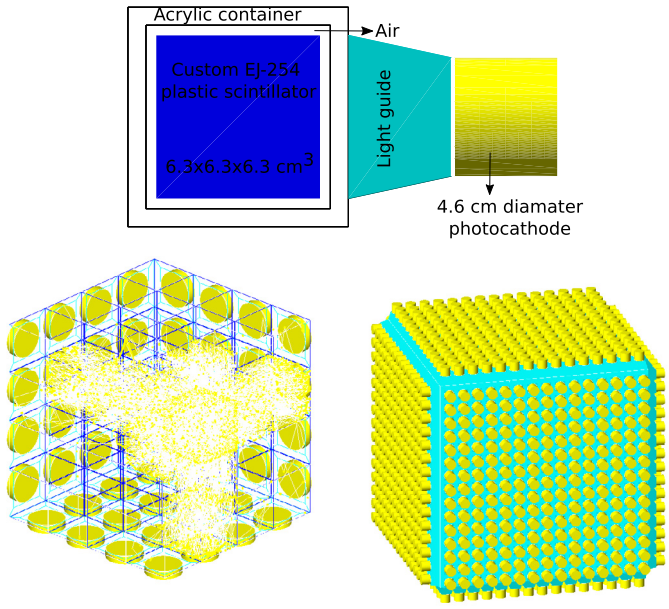

Fig. 7. NULAT-like detector concepts [13,14]. 3D segmentation and 3D light collection that is based on the Raghavan Optical Lattice (ROL) [15].

4.1. NuSD detectors

We classify segmented IBD detectors as homogeneous and inhomogeneous. In a homogeneous detector, the neutrino scintillator is uniformly loaded with a neutron capture agent, while in an inhomogeneous detector, the neutron capture material is in the form of a thin foil/sheet and is wrapped around the scintillator. Moreover, depending on whether a single type or dual type scintillator is employed, inhomogeneous detectors are split into two types. Fig. 6 shows the categorized NuSD detectors schematically.

4.1.1. Homogeneously doped detectors

NULAT-style NuLat (short for Neutrino Lattice) is a segmented detector design based on the Raghavan Optical Lattice (ROL) concept. Its first phase consists of a cubical assembly of 125 ^{6}Li-doped plastic scintillator cubes ($5\times5\times5$ cm^3) and the entire detector will be made of 3375 cubes ($15\times15\times15$). The air gaps separating each cube enhance the total internal reflection effect thus channeling the light emitted in each cube after an energy deposition towards photosensors placed on the sides of the detectors. This concept provides excellent spatial and energy resolution within the entirety of the detector and is well-suited to reject external backgrounds.

Fig. 7 shows three different layouts of a Nulat-style detector plotted with NuSD: a single module with a dimension of

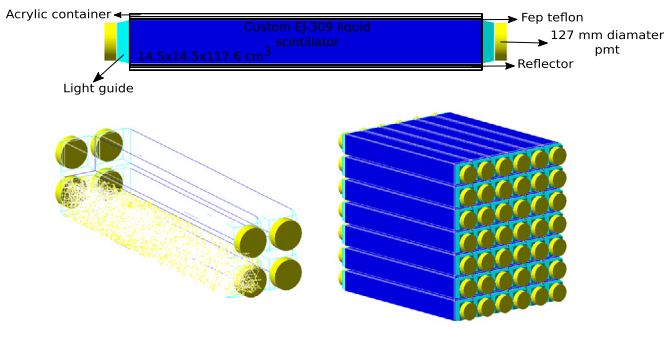

Fig. 8. PROSPECT-like detector concepts [16–19]. 2D segmentation and 1D light collection.

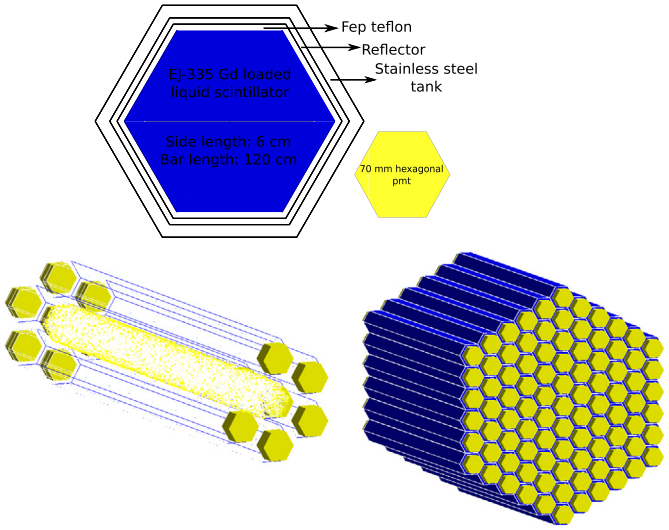

Fig. 9. HSP detector concepts [20]. 2D segmentation and 1D light collection.

$6.3\times6.3\times6.3$ cm^3 (top), an example configuration of $4\times4\times4$ modules showing how light is transported (bottom-left), and finally a version of $15\times15\times15$ modules corresponding to 0.85 m^3 active volume (bottom-right).

PROSPECT-style The Precision Reactor Oscillation and Spectrum Experiment (PROSPECT) detector design consists of optically isolated segments of ^{6}Li-doped liquid scintillators. The entire lattice is made of 11×14 segments, 1.2 m in length, for a total weight of about 4 tons. Due to the reflective internal walls of the segments, light emitted after an energy deposition in the scintillator is propagated towards both sides of a segment to be detected by PMTs.

Fig. 8 depicts three different layouts of a Prospect-style detector plotted with NuSD: a single module with a dimension of $14.5\times14.5\times117.6$ cm^3 (top), an example configuration of $2\times2\times1$ showing how light is transported (bottom-left), and finally a version of $6\times7\times1$ corresponding to 1.04 m^3 active volume (bottom-right).

HSP-style The Hexagonal Shaped Package (HSP) detector design is an assembly of 91 identical units in a hexagonal shape, as displayed in Fig. 9. Each unit is composed of a hexagonal volume, filled with liquid scintillator, with a side length of 6 cm and a height of 120 cm coupled on both sides to light guides and PMTs. The light collection efficiency is increased by wrapping the scintillator bars and the light guides with an aluminized Mylar sheet. This final assembly offers an active volume of 1 m^3 and a gadolinium concentration of 0.25% by weight.

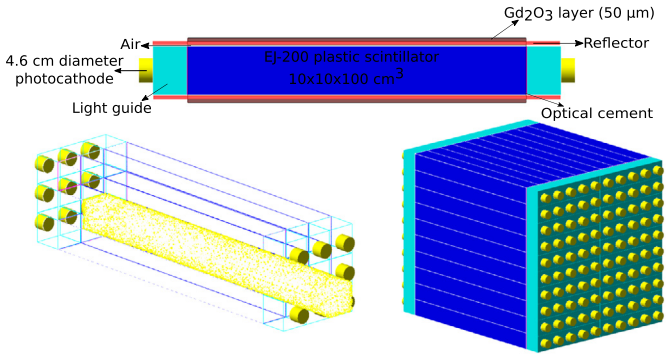

Fig. 10. PANDA-like detector concepts [21,22]. 2D segmentation and 1D light collection.

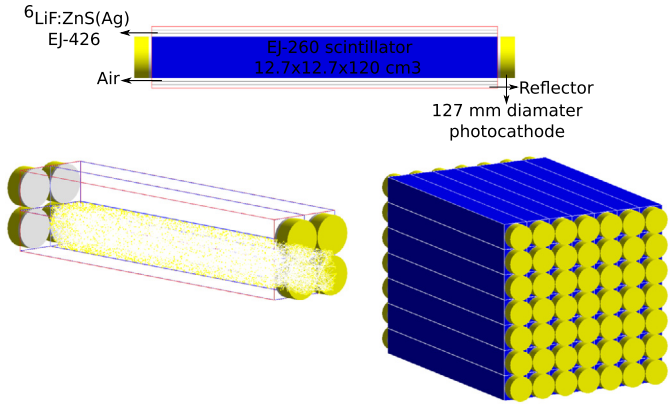

Fig. 11. SWEANY-like detector concepts [23,24]. 2D segmentation and 1D light collection.

4.1.2. Inhomogeneous detectors

PANDA-style The Plastic Anti-Neutrino Detector Array (PANDA) concept consists of an array of 100 identical modules in a 10×10 rectangular shape, as displayed in Fig. 10. Each module consists of a scintillator bar ($10\times10\times100$ cm^3) wrapped in aluminized Mylar film to increase light collection efficiency. A second layer of Mylar film is coated with a 50 μm-thick deposit of gadolinium oxide (Gd_2O_3) to enhance the neutron capture efficiency (4.9 mg of Gd per cm^2). Each bar is connected to acrylic light guides and PMTs on both sides.

4.1.3. Inhomogeneous composite detectors

SWEANY-style The SWEANY-like design, named after M. Sweany, the corresponding author of the initial concept [23], consists of a rectangular assembly of individual plastic scintillator units. Each unit is a $12.7\times12.7\times60$ cm^3 bar covered on all long sides with a 0.45 mm thick layer of ^{6}LiF:ZnS(Ag) scintillator and a PMT is coupled to each short side of the unit.

Fig. 11 shows three different layouts of a Sweany-style detector plotted with NuSD: a single module with a dimension of $12.7\times12.7\times120$ cm^3 (top), an example configuration of $2\times2\times1$ showing how light is transported (bottom-left), and finally a version of $7\times7\times1$ corresponding to 0.95 m^3 active volume (bottom-right).

CHANDLER-style Similar to NuLat, Carbon Hydrogen Anti-Neutrino Detector with Lithium Enhanced Raghavan optical lattice (CHANDLER) is a segmented detector design based on the Raghavan Optical Lattice concept. The Raghavan Optical Lattice divides the active region of the detector into identical cubic cells. Each cube is made of wavelength shifting plastic scintillators. Light is transported along rows and columns of cells by total internal reflection.

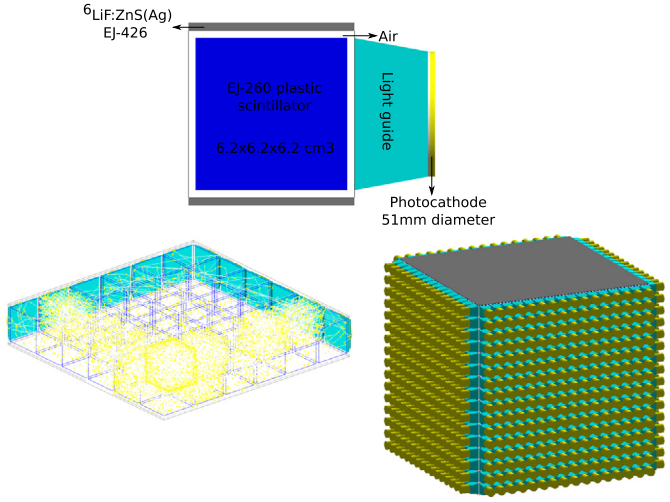

Fig. 12. CHANDLER-like detector concepts [25–28]. 3D segmentation and 2D light collection that is based on Raghavan Optical Lattice [15]. Light is transported along rows and columns of cells by total internal reflection.

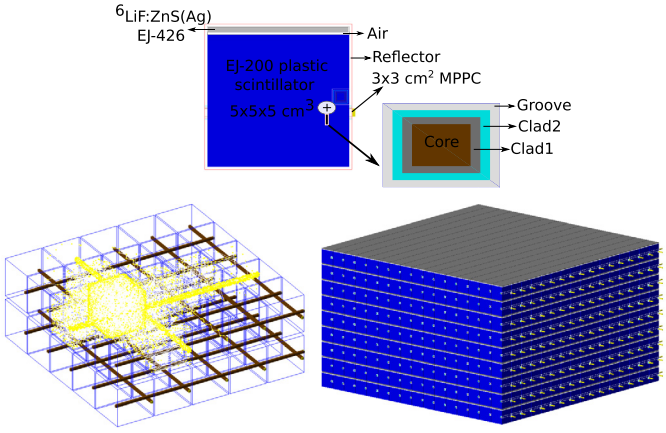

Fig. 13. SOLID-like detector concepts [29–34]. 3D segmentation and 2D light collection.

PMTs on both ends of the bars collect light generated in neutrino scintillators and neutron scintillators (^{6}LiF:ZnS). The neutron detection sheet optically isolates the cube layers, although a small amount of light can pass through the sheet.

Fig. 12 shows three different layouts of a Chandler-style detector plotted with NuSD: a single module with a dimension of $6.2\times6.2\times6.2$ cm^3 (top), an example configuration of $6\times1\times6$ showing how light is transported (bottom-left), and finally a version of $16\times16\times16$ corresponding to 0.98 m^3 active volume (bottom-right).

SOLID-style Search for Oscillations with a ^{6}Li detector (Solid) is a segmented composite scintillator technology which uses polyvinyl toluene (PVT) plastic scintillator cubes, each coupled to a neutron sensitive inorganic scintillator ^{6}LiF:ZnS(Ag) screen. Each cube is also optically isolated from its neighbors, and the scintillation signals from the two scintillators are collected via the same wavelength shifting fibers connected with silicon Multi-Pixel Photon Counter.

Fig. 13 shows three different layouts of a Solid-style detector plotted with NuSD: a single module with a dimension of $5\times5\times5$ cm^3 (top), an example configuration of $5\times2\times5$ showing how light is transported (bottom-left), and finally a version of $16\times9\times16$ corresponding to 0.29 m^3 active volume (bottom-right).

4.2. Example

We plot two important measurable quantities that are essential to the analysis of IBD data and cannot be obtained directly from the output of NuSD. The first one is an anti-neutrino signal that reveals itself with two correlated signals, the so-called prompt and delayed signal. The second one is a measurement of the total energy accumulated over all detector segments. Chandler is chosen as a representative detector and established based on the respective experiment.

Fig. 14 depicts an exemplary anti-neutrino signal observed with Chandler. The prompt and delayed pulses show the number of detected photons as a function of time. Since the decay time constants of the neutrino (EJ-260) and neutron scintillators (EJ-426) employed in this detector are different, the pulse shape discrimination technique can be utilized for discrimination of prompt and delayed signals.

Fig. 15 shows the distribution of the total energy deposited in all segments of the Chandler detector. Since composite detectors employ two distinct scintillator types, the results are shown for both scintillators. The amount of energy accumulated in the neutrino scintillators is calculated by considering the time of energy deposition (before and after 0.1 us for this example).

The macro files that plot these two graphs are also provided for users. To produce the same graphs for other detector types, simply change the detector name in the source code of the macros.

4.3. Comparison metrics

When performing an optimization study with NuSD, improving an existing design, or developing a new detector concept, the following physics quantities can be considered as a benchmark for performance evaluation:

- Intrinsic IBD detection efficiency (ϵ_{IBD}). This is the most significant parameter for an anti-neutrino detector. The following two physics quantities associated with this parameter are also widely used in design optimization and the values of these parameters are usually reported by detector-developing communities.
  - Neutron capture efficiency. This parameter sets an upper limit on the anti-neutrino detection efficiency. If the neutron escapes from the detector in an IBD event, that event is considered lost. However, the opposite is not true. That is, the neutron can be captured, but the resulting particles may escape from the detector.
  - Neutron capture time. Capturing the neutrons as fast as possible is a desirable feature for all detector types. Reducing the neutron capture time is crucial for improving uncorrelated background rejection efficiency.
- Relative energy resolution. The following equation is used as a metric for optimizing a detector's optical components and comparing the energy resolution performance of different detectors.

$$r(E)^2 = \left(\frac{\sigma_{N_e}}{<N_e>}\right)^2 + \left(\frac{\sigma_{LCE}}{<LCE>}\right)^2 + \left(\frac{1}{\sqrt{<N_{p.e}>}}\right)^2 \quad (5)$$

To calculate r(E) from the simulation, the following steps are performed:

- The G4GenericPrimaryGenerator class is selected as a primary generator by activating the line GENERIC_PRIMARY_GENERATOR in *NuSD_config.h* file. By default, this file generates a 1 MeV electron at a random position within the

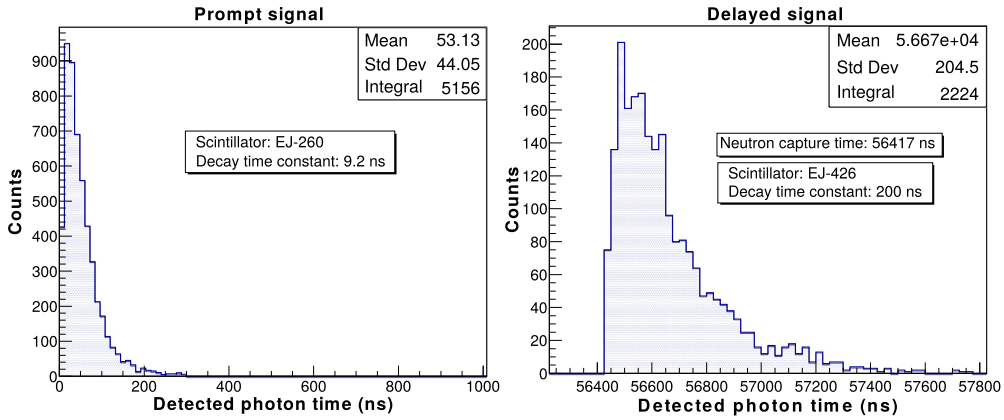

Fig. 14. Prompt and delayed components of an anti-neutrino signal obtained with Chandler-style detector. Pulse shape discrimination is an effective method for discriminating prompt and delayed signals in this type of detector.

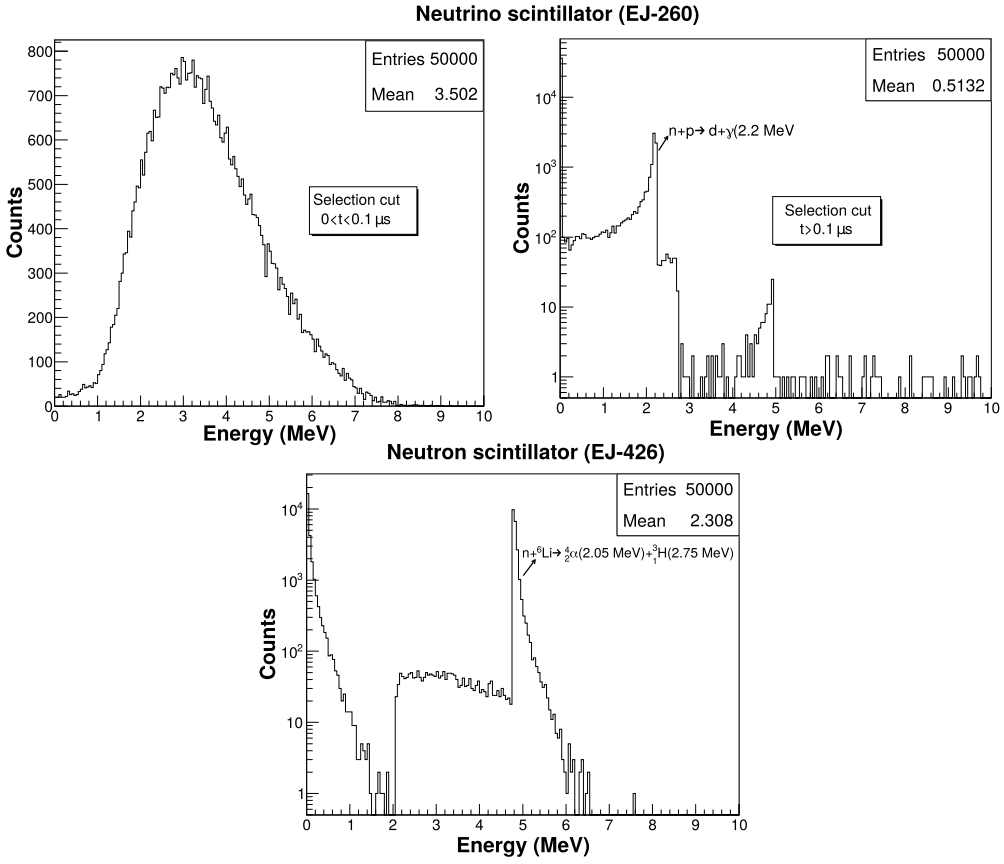

Fig. 15. Total deposited energy distribution in neutrino scintillators (top) and neutron scintillators (bottom). For the top plots, two different time cuts are applied as calculating total deposited energy in neutrino scintillators.

active volume of the detector and fires it in any direction for each event.

– During the analysis stage of the simulated data, histograms are made for the emitted photon number (N_e), total detected photon number ($N_{p.e}$), and the light collection efficiency (LCE) which is defined as the ratio of the number of photons detected to the total number of photons emitted. The mean values and the standard deviations of these distributions constitute the parameters of Eq. (5). The first term in Eq. (5) corresponds to the intrinsic energy resolution of the scintillator and includes the fluctuation of the light generation process. The second term is defined as the transfer factor and includes the fluctuation of light collection and light detection processes. The last term corresponds to the statistical contribution of the photosensors.

Table 1 comparatively presents the physics quantities defined above for all NuSD detectors. The main purpose of this table is not to interpret the values presented, but to show NuSD users that these physical quantities can be obtained with their own input parameters.

Table 2 displays some of the critical input parameters used in the simulation. The findings reported in Table 1 strongly depend on these parameters. For example, the LCE is strongly influenced by the absorption length of the scintillator. Similarly, the neutron capture efficiency is strongly dependent on the identity of the nu-

Table 1
Some selected simulation results obtained with NuSD depending on the user input parameters (see Table 2). Neutron capture efficiency strongly depends on the concentration of the neutron capturing elements such as Li and Gd. The light collection efficiency depends on several parameters, the most important of which are shown in Table 2. Here, the quantum efficiency (QE) of the PMT is set to 0.25 and the QE of the SiPM is set to 0.35.

Time window (us)	Inhomogeneous composite			Homogeneous			Inhomogeneous
	SWEANY	CHANDLER	SOLID	PROSPECT	NULAT	HSP	PANDA
	Neutron capture efficiency (%)						
0<t<50	26	21	22	45	51	68	38
50<t<100	11	11	11	15	14	2	14
100<t<150	6	6	6	5	4	0	7
150<t<200	4	4	3	2	1	0	3
200<t<250	2	3	2	1	0	0	1
250<t<500	3	3	2	0	0	0	1
Mean neutron capture time (us)	85	96	80	48	40	16	59
Delayed time window (us)	Anti-neutrino detection efficiency (%) prompt energy>2 MeV and delayed energy>3 MeV						
1<t<50	20	16	17	35	40	33	24
50<t<100	9	9	8	12	11	3	9
100<t<150	5	5	5	4	3	0	4
150<t<200	3	3	3	2	1	0	2
200<t<250	2	2	1	1	0	0	1
250<t<500	2	3	2	0	0	0	1
1<t<500	42	38	36	55	56	37	42
Optical parameters							
Mean of LCE (%)	9.18	10.56	0.51	3.80	3.92	7.37	3.19
Std deviation of LCE	1.16	0.72	0.08	0.58	1.09	0.80	0.18
Intrinsic energy resolution	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Transfer factor	0.13	0.07	0.16	0.15	0.28	0.11	0.06
Statistical contribution	0.03	0.03	0.13	0.06	0.06	0.04	0.06
Relative energy resolution	0.13	0.08	0.20	0.16	0.29	0.12	0.08

cleus used as the neutron capture agent and its concentration in the active volume. Some input parameters are set to a fixed value at all photon energies. If users wish, they can provide the discrete energy spectrum through the text files provided for them.

5. Conclusions and future plans

In this work, a new simulation framework named NuSD, which is developed in the most general sense to conduct simulation studies in segmented scintillation detectors, is introduced. It focuses on near-field reactor neutrino detectors developed for the investigation of neutrino oscillation phenomena or potential applications of reactor neutrinos. NuSD combines seven different novel detector concepts developed by various international collaborations under a joint simulation environment, giving the neutrino physics community the ability to explore and test all of these detector technologies.

The benefits of NuSD can be summarized as:

- Providing a flexible environment to simulate IBD events in seven different detector concepts. It provides a realistic neutrino signal for each detector type and it allows for simulation of any particle in different detector types.
- Providing a flexible code to perform optimization studies for a selected detector type. Users can easily optimize the detector design parameters such as dimensions of each detector component, number of segments, the concentration of neutron capture agents, the thickness of neutron converter material, reflector reflectivity, photo-cathode radius, and photo-cathode efficiency. These parameters may vary depending on the chosen detector type.
- Allowing the users to incorporate a new detector concept into NuSD using the supplied classes.
- Providing many ready-to-use ROOT macro files for the analysis of simulation data.

NuSD is a complete simulation framework and a very beneficial tool for the physics community, and we plan to develop its applications in the near-future. Our plan is to have an updated version with the specifications below:

- A more user-friendly tool: even if many critical simulation parameters can be managed with user interface commands in the NuSD, it can still be developed further. We're aiming to have an updated version, in which users can control the simulation entirely through user interface commands.
- A compatible version with the updated versions of Geant4: after the Geant4 10.0.6 version, the implementation method of optical photons has changed considerably and new functionalities have been added. We are planning to adapt these changes quickly in the next version of NuSD.
- Actively following advances in segmented detector technology: we aim to incorporate innovative detector concepts into the next version of the NuSD.
- Extending the scope of the NuSD: it is mainly developed for simulating reactor neutrinos in segmented scintillation detectors, but our goal is to extend the NuSD framework to encompass calorimetric studies.
- Integrating a detailed signal analysis package into NuSD: NuSD currently provides some ready-to-use ROOT macro files to analyze raw data, but we're planning to build and integrate a detailed signal (physics) analysis package into the next version of NuSD.

Table 2
There are a large number of input parameters available, but only the most important ones in our judgment are listed here. However, users can run the program using their own experimental input parameters.

Neutron screen material composition [35]								
Density = 2.42* g/cm^3	Capture (LiF)		Scintillation (ZnS)		Binder			
Element	Enriched Li	F	Zn	S	C	H	O	Si
Mass fraction (%)	3.93	12.31	43.58	21.37	6.57	0.17	4.38	7.69
Custom EJ-309 scintillator material composition [17]								
Density = 0.979* g/cm^3	Homogeneously doped liquid scintillator							
Element	Enriched Li	H	C	O				
Mass fraction (%)	0.11	9.52	84.14	6.23				
Custom EJ-254 scintillator material composition [13]								
Density = 1.009* g/cm^3	Homogeneously doped plastic scintillator							
Element	Enriched Li	H	C					
Mass fraction (%)	0.11	8.56	91.33					
EJ-335 scintillator material composition								
Density = 0.89* g/cm^3	Homogeneously doped liquid scintillator							
Element	Natural Gd	H	C					
Mass fraction (%)	0.25	11.59	88.16					
Some important optical parameters								
Material	Light yield 1/MeV	Refractive index	Absorption length (cm)					
EJ-200	10000	1.58	300					
EJ-260	9200	1.58	Energy dependent					
EJ-426	40000	1.7	Energy dependent					
EJ-335	8500	1.49	450					
Custom EJ-254	7500	1.58	50					
Custom EJ-309 [17]	8200	1.57	85					
Pmma	-	1.49	300					
Air	-	1	300					
Fep	-	1.34	300					
Optical cement	-	1.57	300					
Optical grease	-	1.43	300					

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] T. Akindele, et al., Nu tools: exploring practical roles for neutrinos in nuclear energy and security, https://doi.org/10.2172/1826602.

[2] P. Vogel, J.F. Beacom, Phys. Rev. D 60 (1999) 053003, https://doi.org/10.1103/PhysRevD.60.053003.

[3] M. Askins, et al., The physics and nuclear nonproliferation goals of WATCHMAN: a, arXiv:1502.01132.

[4] J.C. Anjos, et al., AIP Conf. Proc. 1222 (1) (2010) 427–430, https://doi.org/10.1063/1.3399360.

[5] V. Fischer, E. Tiras, Nucl. Instrum. Methods A 969 (2020) 163931, https://doi.org/10.1016/j.nima.2020.163931, arXiv:2001.02655.

[6] A. Bat, E. Tiras, V. Fischer, M. Kamislioglu, Low energy neutrino detection with a portable water-based liquid scintillator detector, arXiv:2112.03418.

[7] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, et al., Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 506 (3) (2003) 250–303, https://doi.org/10.1016/S0168-9002(03)01368-8.

[8] Geant4 Collaboration, Geant4: book for toolkit developers, https://geant4-userdoc.web.cern.ch/UsersGuides/ForToolkitDeveloper/fo/BookForToolkitDevelopers.pdf.

[9] M. Fowler, UML Distilled: a Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 2015.

[10] R. Brun, F. Rademakers, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 389 (1) (1997) 81–86, https://doi.org/10.1016/S0168-9002(97)00048-X.

[11] Geant4 Collaboration, Geant4: book for application developers, https://geant4-userdoc.web.cern.ch/UsersGuides/ForApplicationDeveloper/BackupVersions/V10.6c/fo/BookForApplicationDevelopers.pdf.

[12] A. Ribon, J. Apostolakis, A. Dotti, G. Folger, V. Ivanchenko, M. Kosov, V. Uzhinsky, D. Wright, in: CERN-LCGAPP, vol. 2, 2010, p. 2010.

[13] C. Lane, et al., A new type of neutrino detector for sterile neutrino search at nuclear reactors and nuclear nonproliferation applications, arXiv:1501.06935.

[14] D. Xinjian, Development and calibration of nulat, a new type of neutrino detector, PhD thesis, 2018, https://vtechworks.lib.vt.edu/handle/10919/82933.

[15] C. Grieb, J.M. Link, R.S. Raghavan, Phys. Rev. D 75 (2007) 093006, https://doi.org/10.1103/PhysRevD.75.093006.

[16] J. Ashenfelter, et al., J. Instrum. 10 (11) (2015) P11004, https://doi.org/10.1088/1748-0221/10/11/p11004.

[17] J. Ashenfelter, et al., J. Instrum. 13 (06) (2018) P06023, https://doi.org/10.1088/1748-0221/13/06/p06023.
[18] J. Ashenfelter, et al., J. Instrum. 14 (03) (2019) P03026, https://doi.org/10.1088/1748-0221/14/03/p03026.
[19] J. Ashenfelter, et al., Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 922 (2019) 287–309, https://doi.org/10.1016/j.nima.2018.12.079.
[20] M. Kandemir, A. Cakir, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 953 (2020) 163251, https://doi.org/10.1016/j.nima.2019.163251.
[21] Y. Kuroda, S. Oguri, Y. Kato, R. Nakata, Y. Inoue, C. Ito, M. Minowa, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 690 (2012) 41–47, https://doi.org/10.1016/j.nima.2012.06.040.
[22] S. Oguri, Y. Kuroda, Y. Kato, R. Nakata, Y. Inoue, C. Ito, M. Minowa, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 757 (2014) 33–39, https://doi.org/10.1016/j.nima.2014.04.065.
[23] M. Sweany, J. Brennan, B. Cabrera-Palmer, S. Kiff, D. Reyna, D. Throckmorton, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 769 (2015) 37–43, https://doi.org/10.1016/j.nima.2014.09.073.
[24] S.D. Kiff, N. Bowden, J. Lund, D. Reyna, in: Symposium on Radiation Measurements and Applications (SORMA) XII 2010, Nucl. Instrum. Methods Phys. Res., Sect. A, Accel. Spectrom. Detect. Assoc. Equip. 652 (1) (2011) 412–416, https://doi.org/10.1016/j.nima.2010.07.082.
[25] A. Haghighat, P. Huber, S. Li, J.M. Link, C. Mariani, J. Park, T. Subedi, Observation of reactor antineutrinos with a rapidly-deployable surface-level detector, arXiv:1812.02163, 2018.
[26] P. Huber, J.M. Link, C. Mariani, S. Pal, J. Park, J. Phys. Conf. Ser. 1216 (2019) 012014, https://doi.org/10.1088/1742-6596/1216/1/012014.
[27] T.P. Subedi, Observation of Reactor Antineutrinos with the Chandler Detector, PhD thesis, Virginia Tech., 2020.
[28] S. Li, Detection of Antineutrinos at the North Anna Nuclear Generating Station, PhD thesis, Virginia Tech., 2020.
[29] W.V.D. Pontseele, Characterisation and modelling of correlated noise in silicon photomultipliers for the solid experiment, PhD thesis, 2017, https://vtechworks.lib.vt.edu/handle/10919/82933.
[30] D.M. Saunders, First data reconstruction and inverse beta decay analysis at the large scale solid prototype detector, PhD thesis, 2017, http://solid-experiment.org/sites/default/files/pdf/thesis/PhD_DanielSaunders.pdf.
[31] Y. Abreu, et al., J. Instrum. 12 (04) (2017) P04024, https://doi.org/10.1088/1748-0221/12/04/p04024.
[32] L. Arnold, W. Beaumont, D. Cussans, D. Newbold, N. Ryder, A. Weber, J. Instrum. 12 (02) (2017) C02012, https://doi.org/10.1088/1748-0221/12/02/c02012.
[33] Y. Abreu, et al., J. Instrum. 13 (05) (2018) P05005, https://doi.org/10.1088/1748-0221/13/05/p05005.
[34] Y. Abreu, et al., J. Instrum. 13 (09) (2018) P09005, https://doi.org/10.1088/1748-0221/13/09/p09005.
[35] J. Ely, E.R. Siciliano, M.T. Swinhoe, A. Linte reur, Modeling and simulation optimization and feasibility studies for the neutron detection without helium-3 project, Tech. Rep., Pacific Northwest National Lab. (PNNL), 2013, https://doi.org/10.2172/1069205.