

PAPER • OPEN ACCESS


## FlameNEST: explicit profile likelihoods with the Noble Element Simulation Technique

To cite this article: R.S. James *et al* 2022 *JINST* 17 P08012

View the [article online](#) for updates and enhancements.

You may also like

- [Effects of noble metal doping on hydrogen sensing performances of monolayer MoS<sub>2</sub>](#)  
Zheng Zhang, Kai Chen, Qiang Zhao *et al.*
- [Fabrication of Pt-Nanoparticle-Loaded Mesoporous Alumina Coating through Anodizing of an Al-Pt Alloy](#)  
Yanlong Ma, Hua Lin and Yi Liao
- [Plasmonic photocatalysis](#)  
Xuming Zhang, Yu Lim Chen, Ru-Shi Liu *et al.*



**ECS** The Electrochemical Society  
Advancing solid state & electrochemical science & technology


### 242nd ECS Meeting

Oct 9 – 13, 2022 • Atlanta, GA, US

Presenting more than 2,400 technical abstracts in 50 symposia

 **ECS Plenary Lecture featuring M. Stanley Whittingham,** Binghamton University Nobel Laureate – 2019 Nobel Prize in Chemistry

 Register now!



RECEIVED: March 16, 2022

REVISED: June 1, 2022

ACCEPTED: June 27, 2022

PUBLISHED: August 11, 2022

# FlameNEST: explicit profile likelihoods with the Noble Element Simulation Technique

R.S. James,<sup>a,\*</sup> J. Palmer,<sup>b</sup> A. Kaboth,<sup>b,c</sup> C. Ghag<sup>a</sup> and J. Aalbers<sup>d,e</sup>

<sup>a</sup>*Department of Physics and Astronomy, University College London,  
Gower St, London, United Kingdom*

<sup>b</sup>*Department of Physics, Royal Holloway University of London,  
Egham Hill, Egham, United Kingdom*

<sup>c</sup>*Particle Physics Department, Rutherford Appleton Laboratory,  
Harwell Campus, Didcot, United Kingdom*

<sup>d</sup>*Kavli Institute for Particle Astrophysics and Cosmology, Stanford University,  
Serra Mall, Stanford, U.S.A.*

<sup>e</sup>*Particle Astrophysics and Cosmology Division, SLAC National Accelerator Laboratory,  
Sand Hill Rd, Menlo Park, U.S.A.*

E-mail: [robert.james.19@ucl.ac.uk](mailto:robert.james.19@ucl.ac.uk)

**ABSTRACT:** We present FlameNEST, a framework providing explicit likelihood evaluations in noble element particle detectors using data-driven models from the Noble Element Simulation Technique. FlameNEST provides a way to perform statistical analyses on real data with no dependence on large, computationally expensive Monte Carlo simulations by evaluating the likelihood on an event-by-event basis using analytic probability elements convolved together in a single TensorFlow multiplication. Furthermore, this robust framework creates opportunities for simple inter-collaboration analyses which will be fundamental for the future of experimental dark matter physics.

**KEYWORDS:** Analysis and statistical methods; Noble liquid detectors (scintillation, ionization, double-phase); Time projection Chambers (TPC); Simulation methods and programs

\*Corresponding author.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Dual-phase Liquid Xenon Time Projection Chambers</b>	<b>3</b>
<b>3</b>	<b>Technical implementation</b>	<b>4</b>
3.1	Block structure	5
3.1.1	Pre-quanta	6
3.1.2	Post-quanta	7
3.2	Performance features	8
3.2.1	Generalising bounds computations	8
3.2.2	Variable tensor stepping	10
3.2.3	Variable energy stepping	10
3.2.4	Model-dependent approximations	10
<b>4</b>	<b>Validations</b>	<b>11</b>
4.1	Mono-energetic sources	11
4.2	Full energy spectra	14
<b>5</b>	<b>Conclusion</b>	<b>17</b>
<b>A</b>	<b>Model details</b>	<b>17</b>
A.1	Model parameters	17
A.2	Pre-quanta models	19
A.3	Post-quanta models	20
<b>B</b>	<b>Modified skew Gaussian to implement NEST constraint</b>	<b>21</b>
<b>C</b>	<b>Manual ion bound computation in FlameNEST</b>	<b>22</b>

---

## 1 Introduction

Observations on both galactic and cosmological scales have found that dark matter constitutes approximately 85% of the matter density in the universe [1, 2]. Over the past decade, time projection chambers (TPCs) containing liquefied noble elements have become the leading technology in the search for the medium of dark matter [3–6]. Rare event searches such as these often choose to use frequentist hypothesis testing to present their results [7]. The central object of such tests is the likelihood which may be obtained via computation of a differential event rate  $R^j(\{O_i\})$ . This is the number of expected events from the  $j^{\text{th}}$  signal or background source producing a given set of observables  $\{O_i\}$ , when integrated over observable space. Experiments today estimate

such differential event rates by filling multi-dimensional histograms (templates) in a binned space of observables using Monte Carlo (MC) techniques. Underlying ‘nuisance’ parameters may be incorporated by creating multiple templates and interpolating between them — these are parameters which influence the event probability model but are of secondary interest to the experiment. Filling these templates to the requisite statistical accuracy scales exponentially with both the number of observables and the number of nuisance parameters, making such analyses computationally unwieldy. A common compromise is to restrict the number of observables and limit the number of underlying nuisance parameters, the former reducing the signal/background discrimination of the detector and the latter making the analysis less robust. Even in previous experiments where spatial and temporal information has been needed due to highly variable detector conditions [3], coarse binning has been used and few underlying detector response nuisance parameters have been fitted. Using a neural network to project observations onto a single dimension can limit the loss of information coming from removing observables [8], but at the cost of some interpretability. Finally, while projecting or restricting the number of observables makes templates simpler to generate, exponentially many templates are still needed to treat correlated nuisance parameters.

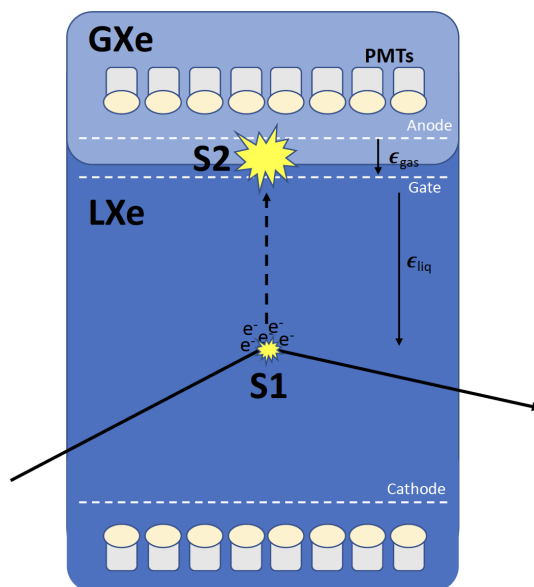
Flamedisx is an open-source Python package allowing for likelihood evaluation scaling approximately linearly rather than exponentially in the number of nuisance parameters. Further to this, there is no scaling with the inclusion of certain additional observables, making the inclusion of many more such dimensions computationally feasible [9]. This is achieved by calculating likelihoods on an event-by-event basis using real experimental data. Flamedisx computes a sum over ‘hidden variables’ where each term is a product of conditional probabilities calculated from the analytic probability density/mass function (PDF/PMF) of one part of the detector response model — the distinction here comes in modelling continuous versus discrete variables. The computation is performed using TensorFlow [10], which allows for automatic differentiation to facilitate likelihood maximisation. TensorFlow is greatly accelerated when run on a GPU, increasing computation speed roughly hundred-fold in the case of Flamedisx [9].

The detector response models originally implemented within Flamedisx, as described in [9], are inspired primarily by the XENON1T detector [11]. To extend the Flamedisx framework to be more detector-agnostic, we have incorporated the xenon models of the Noble Element Simulation Technique (NEST) into Flamedisx. NEST is a precise, detector-agnostic parameterisation of excitation, ionisation, and the corresponding scintillation and electroluminescence processes in liquid noble elements as a function of both energy and electric field [12]. These models are constantly being scrutinised and validated against real data collected by a variety of world-leading noble element experiments. In addition to improving the accuracy of and extending the reach of analyses done using Flamedisx, we believe that using the community’s gold-standard collection of noble element response models encapsulated in NEST will allow for Flamedisx to be used for future inter-collaboration data analyses between different noble element experiments, further extending physics reach.

This paper outlines the technical challenges of incorporating the NEST models into Flamedisx, a framework henceforth referred to as FlameNEST. We also present the results of a series of validations and discuss the resulting speed implications of our work. The focus throughout will be on dual-phase liquid xenon (LXe) TPCs; however, NEST contains additional models for single-phase gaseous xenon detectors along with liquid argon detectors, incorporation of which into FlameNEST is a future goal.

## 2 Dual-phase Liquid Xenon Time Projection Chambers

A general schematic of a dual-phase LXe TPC is shown in figure 1. These detectors are typically cylindrical and filled with LXe with a thin layer of gaseous xenon (GXe) above. A traversing particle will scatter off either the electrons or nucleus of the xenon atoms, producing electronic recoils (ER) or nuclear recoils (NR), respectively. These recoils produce xenon excimers, which emit UV photons as they de-excite. This prompt scintillation signal, S1, is detected by arrays of photon detectors — typically photomultiplier tubes (PMTs) — located at the top and bottom of the detector. Electron/ion pairs can also be produced from a recoil. These ionisation electrons will drift in an electric field,  $\epsilon_{\text{liq}}$ , towards the liquid-gas interface. Some electrons may be absorbed onto impurities within the LXe before reaching the top of the detector. This can be quantified by the electron lifetime, which reduces the average size of the signal towards the bottom of the detector. Once the electrons reach the liquid-gas interface, they will experience a much stronger electric field,  $\epsilon_{\text{gas}}$ , designed to extract them from the liquid into the gas and to produce a larger secondary signal, S2, via electroluminescence. The PMTs used in LXe TPCs have a probability,  $P_{\text{dpe}}$ , of producing two photoelectrons rather than one from a single detected LXe scintillation photon. This process must be accounted for when modelling the detector response [13].



**Figure 1.** Schematic of a dual-phase LXe TPC showing the signal processes from an interaction in the detector.

The spatial hit pattern of S2 photons in the top and bottom PMT arrays provides  $(x, y)$  position information in the radial plane of the detector. Due to approximately constant electron drift velocity in the liquid, the vertical  $z$  coordinate can be inferred from the time difference between the arrivals of S1 and S2 signals. This gives the full set of observables of an interaction event as  $(S1, S2, x, y, z, t)$ , where  $t$  is the time at the start of the event. It is beneficial to include  $t$  in the list of observables to capture time-varying effects of background sources, WIMP modulation, and possible changes in detector conditions like electric field.

Here, S1 and S2 refer to the integrated pulse area, in terms of detected PMT photoelectrons. FlameNEST currently operates only at this level, which is the observable of choice for statistical inference using xenon-based detectors. Extension to include pulse shape information is something which may be explored in the future. The relative size of the S1 and S2 signals provides information on the underlying interaction type of the event. Signal and background sources of interest in rare event searches can be classified as inducing either nuclear recoil (NR) or electronic recoil (ER) interactions. For the same energy deposited in the scintillation and ionisation channels, discarding that lost to quenching, NR interactions produce a lower S2/S1 ratio than ER interactions; therefore, the ratio of the two can be used as a discrimination metric.

To overcome the aforementioned difficulties in filling high-dimensional Monte Carlo templates, current statistical analyses typically opt to eliminate position dependence of the S1 and S2 values, normalising them to a reference position in the detector. Detector conditions such as temperature and electric field, which can vary throughout the lifetime of an experiment, are typically taken to be constant and data during periods of fluctuation discarded. Likelihood evaluations using Monte Carlo templates often neglect position and time dependence in certain signal and background sources, such as spatially-dependent TPC wall background and the temporal dependence of galactic WIMP annual modulation. This reduces the dimensionality of the observable space from  $(S1, S2, x, y, z, t)$  to ‘corrected’ S1 and S2 values,  $(S1c, S2c)$ .

A significant drawback of such a dimensionality reduction is that signal/background discrimination is reduced. This is particularly the case towards the top of the detector, where S2 signals are large and the relative fluctuations in the inferred charge yield are smaller. Thus, a dimensionality reduction leads to sub-optimal ER/NR discrimination in certain regions of the detector. Furthermore, not correctly accounting for the spatial and temporal dependence of the interaction rates of relevant signal or background sources further reduces signal/background discrimination.

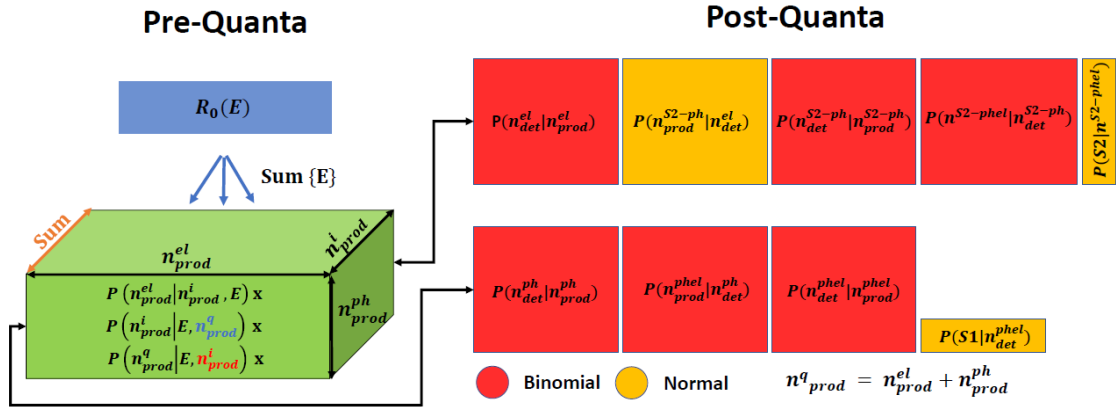
The probability distributions describing each stage of this detector response have parameters which are often functions of many other underlying nuisance parameters — these are specific to the models of the different physical processes constituting the detector response. Whilst auxiliary measurements can constrain them to some degree, a truly robust analysis will allow them to float during inference. Enabling this with a Monte Carlo template likelihood evaluation would lead to exponential scaling in the template generation as more nuisance parameters are included, whereas the Flamedisx computation scales instead approximately linearly with nuisance parameters as seen in figure 6 in [9].

### 3 Technical implementation

NEST fully models the production of ionised and excited xenon atoms (ions/excitons), including recombination fluctuations, which is subsequently used in modelling the ionisation electron and scintillation photon yields. In contrast, the original Flamedisx models did not feature this extra degree of freedom — the splitting of quanta between electrons and photons was modelled directly as a function of energy. Additionally, the detector response models translating produced quanta distributions into observable signal distributions in NEST feature a number of extra steps compared to the original Flamedisx models. Consequently, it was not possible to incorporate the NEST models directly into the original tensor structure of Flamedisx. Therefore, the underlying tensor structure of Flamedisx was extended to incorporate the NEST models in full generality. In this section we outline this new structure.

### 3.1 Block structure

The FlameNEST block structure is shown in figure 2, which may be compared with the original Flamedisx block structure in figure 3 of [9]. The pre-quanta stage maps between the differential rate spectrum of the interaction of the  $j^{\text{th}}$  source with the liquid xenon,  $R^j(E, x, y, z, t)$ , where  $E$  is the energy of the interaction, and produced quanta (photon/electron) distributions. The post-quanta stage maps between these produced quanta distributions and the distributions of signals, S1 and S2. The models depend on the type of interaction of the source with the xenon atoms — whether an ER or NR occurs. Variables unique to the ER source are written in blue in the central green block and the variables unique to the NR source are written in red. Event position and time additionally enter at the level of the model functions used in the probabilistic detector response model.



**Figure 2.** FlameNEST block structure. The blocks are categorised by whether they model pre-quanta processes (production of electrons and photons from an energy deposition) or post-quanta processes (detection of quanta and translation to final signals). The dimensions of each block are indicated graphically. Every block has an additional dimension, not depicted here, over events within a computation batch. The probability distributions for the post-quanta blocks are indicated by their colour — see section 3.1.1 for details of the pre-quanta distributions. In the green pre-quanta block, the colour of the text indicates variables that are used for ER (blue) or NR (red) only.

As outlined in [9], Flamedisx computes bounds on any non-observable dimensions of the blocks for each observed event. Each block then has (conditional) probability elements evaluated within those bounds, based on some probability distribution and model functions determining its parameters. Many relevant approximations to speed up evaluation of probability elements are handled automatically by TensorFlow Probability. The blocks are then multiplied together for different values of energy  $E$ , multiplied by  $R^j(E)$  and the results summed together. In FlameNEST, this sum has the following form:

$$\sum_{E, e, \gamma, i, j, k, l, m, n, \dots} P(S1|i)P(i|j)P(j|\dots) \dots P(k|\gamma)P(e, \gamma|E)R^j(E)P(l|e) \dots P(m|\dots)P(n|m)P(S2|n). \quad (3.1)$$

By evaluating this sum, we obtain the differential event rate  $R^j(S1, S2, x, y, z, t)$ . Here,  $e$  and  $\gamma$  are hidden variables representing the number of produced electrons and photons respectively, whilst  $i, j, k, l, m, n, \dots$  represent other hidden variables in the detector response model such as the



number of electrons/photons detected, for example. The bounds are chosen such that each computed probability element will contribute non-negligibly to the sum over probabilities. The procedure for doing this is outlined in section 3.2.1, and validations of this are presented in section 4.1.

It should be noted that, in some places, NEST uses continuous distributions to model discrete variables, rounding each sampled value during MC simulation. This choice means that the FlameNEST computation needs to include a continuity correction: instead of evaluating  $P(X = x)$ , we evaluate

$$P(X \leq x + 0.5) - P(X \leq x - 0.5). \quad (3.2)$$

### 3.1.1 Pre-quanta

The pre-quanta stage encapsulates the conversion from an energy deposit to a number of produced photons and electrons. The model functions determining the probability distribution parameters are obtained from v2.2.2 of the NEST code [12], and we direct the reader to the references therein for further details of the physics. Here we summarise the probability distributions used in each block, and will direct the reader to appendix A for detailed model descriptions. For these models, we assume a cylindrical TPC with a fixed fiducial volume of liquid, and only consider ER and NR events within the volume.

Incorporation of these NEST yield models into the Flamedisx framework was not possible with a simple modification of the existing blocks coupled with a linear extension to additional blocks, as for the post-quanta models detailed in section 3.1.2. Instead, two substantial modifications were made to the block performing this computation, shown in green in figure 2. Firstly, its dimensionality was increased by one internally contracted dimension, capturing the splitting into ions and excitons before recombination occurs: the sum of ions and excitons is equal to the sum of electrons and photons under the assumptions made in deriving the NEST models [14]. Secondly, a number of these tensors are summed together over a set of relevant energies for each event, reflecting the parameterisation of NEST’s yield models by ‘true’ energy deposition. This is in contrast to the original Flamedisx models, where the yields are parameterised in terms of some pre-computed number of net electrons and photons produced. Both of these modifications introduce memory usage and performance challenges, discussed further in section 3.2.

Let us consider the pre-quanta model block for the ER case. A normal distribution is used to model the fluctuations on the mean yields, producing  $n_{\text{prod}}^{\text{q}}$  total quanta. From this, a binomial process models a number of produced ions  $n_{\text{prod}}^{\text{i}}$ . Finally, a skew normal distribution models the recombination fluctuations leading to a produced number of electrons,  $n_{\text{prod}}^{\text{el}}$ , such that we can then obtain a produced number of photons,  $n_{\text{prod}}^{\text{ph}}$  by subtracting  $n_{\text{prod}}^{\text{el}}$  from the number of quanta produced,  $n_{\text{prod}}^{\text{q}}$ . Both the normal and skew normal distributions have continuity corrections accounted for, as NEST uses continuous distributions and rounding to model discrete random variables. When dealing with the skew normal distribution, we need to account for the additional constraint the NEST models impose, that  $n_{\text{prod}}^{\text{el}} \leq n_{\text{prod}}^{\text{i}}$ . This is done at the level of the distribution, and is detailed fully in appendix B.

In the NR case, a normal distribution models the production of  $n_{\text{prod}}^{\text{i}}$  ions based on the mean yield, with a further normal distribution modelling the difference between the produced number of total quanta  $n_{\text{prod}}^{\text{q}}$  and the value of  $n_{\text{prod}}^{\text{i}}$ . We can now obtain  $n_{\text{prod}}^{\text{el}}$  which is modelled identically to the ER case, with just the forms of the model functions determining the parameters being different. Continuity corrections are applied here for all three distributions.



We construct the green tensor in figure 2 over suitable hidden variable values of the 3 dimensions  $(n_{\text{prod}}^{\text{el}}, n_{\text{prod}}^{\text{ph}}, n_{\text{prod}}^{\text{i}})$ . A fourth dimension is included if events are grouped into batches. This tensor is constructed for a specific value of the energy,  $E$ . Each element is then the product of the 3 probability elements:  $P(n_{\text{prod}}^{\text{el}}|E)$ ,  $P(n_{\text{prod}}^{\text{i}}|E)$ , and  $P(n_{\text{prod}}^{\text{ph}}|E)$  for either ER or NR sources, where we indicate the explicit dependence on energy but not the other conditional dependencies seen in figure 2, which are different for ER and NR. Energy dependence enters at the level of the mean electron, photon, exciton and ion yields, which are used in calculating distribution parameters, outlined more clearly in appendix A.2.

Contracting each of these tensors internally over the  $n_{\text{prod}}^{\text{i}}$  dimension results in a tensor over  $(n_{\text{prod}}^{\text{el}}, n_{\text{prod}}^{\text{ph}})$  which is constructed of probability elements  $P(n_{\text{prod}}^{\text{el}}, n_{\text{prod}}^{\text{ph}}|E)$ , defined as the probability of a certain ER or NR energy deposit to produce  $n_{\text{prod}}^{\text{el}}$  electrons and  $n_{\text{prod}}^{\text{ph}}$  photons, given the energy  $E$  of the deposit. For each event, we multiply this at each energy by the value of the interaction rate spectrum of the  $j^{\text{th}}$  source,  $R^j(E)$ , which may also be a function of event position and time for certain sources. We henceforth refer to this quantity as the energy spectrum. We then multiply this with the post-quanta blocks and repeat over  $R^j(E)$ . By summing these results together, we obtain  $R^j(S1, S2, x, y, z, t)$ . This can be repeated for all events, and all relevant signal/background sources, to allow for computation of the likelihood of the dataset. More detail on this is given in [9].

### 3.1.2 Post-quanta

The post-quanta stage encapsulates the detection of the produced electrons/photons, as described in section 2. We currently seek only to emulate NEST’s ‘parametric’ S1 calculation mode, where a detection threshold is not applied to individual PMT hits; rather, the DPE effect and a parametric detection efficiency is applied to the sum of detected photons. This leads to a marginally less accurate calculation at very low S1 signal sizes. We intend to incorporate the full calculation in a future version of FlameNEST, though encapsulating it within the tensor framework is not straightforward.

The first block in the lower row of the post-quanta blocks in figure 2 represents the binomial process which describes the number of photons detected,  $n_{\text{det}}^{\text{ph}}$ , given the number of photons produced,  $n_{\text{prod}}^{\text{ph}}$ , with a position-dependent detection probability. Detector threshold effects are also applied at this stage by introducing a minimum photon cut. It should be noted that the minimum photon cut is applied to the total number of detected photons, not accounting for the expected distribution of photons across PMTs, a feature modelled more fully by NEST and used in many experimental analyses. This will be implemented in future FlameNEST versions. The next block describes the binomial process by which the DPE effect may lead to a single detected photon producing two photoelectrons. The total number of photoelectrons is denoted  $n_{\text{prod}}^{\text{phel}}$ . This is followed by a binomial process which links  $n_{\text{prod}}^{\text{phel}}$  to a number of detected S1 photoelectrons,  $n_{\text{det}}^{\text{phel}}$ . Finally, we apply a Gaussian smearing to  $n_{\text{det}}^{\text{phel}}$  to obtain S1, representing the PMT single photoelectron resolution coupled with additional terms to approximate other PMT effects and electronics noise. Acceptance cuts can then be applied to the final S1 signal.

The first block in the upper row of the post-quanta blocks in figure 2 represents the binomial electron survival process during drift, whereby an electron may be lost due to interactions with impurities in the LXe. The number of electrons extracted to the gas region from the  $n_{\text{prod}}^{\text{el}}$  produced electrons in the liquid region is denoted  $n_{\text{det}}^{\text{el}}$ . The efficiency of extraction is calculated within the

NEST models from the gas field; additional nuisance parameters could be introduced here if the user wishes to include uncertainties on this. As previously discussed, these extracted electrons produce electroluminescence in the xenon gas. The number of photons produced from this process is denoted  $n_{\text{prod}}^{\text{S2-ph}}$ , with the process being described by a normal distribution with a continuity correction applied. We use another binomial, again with position-dependent detection efficiencies, to model the detection of a number  $n_{\text{det}}^{\text{S2-ph}}$  of these photons. We introduce the DPE effect identically to the S1 case, leading to  $n^{\text{S2-phel}}$  photoelectrons. A Gaussian smearing is applied to model the final S2 signal, in the same vein as for S1, before acceptance cuts can be applied.

### 3.2 Performance features

The modifications made to Flamedisx to fully capture the NEST models introduced a substantial speed penalty to the computation, necessitating the implementation of a number of additional features to mitigate this. This section details these performance features.

#### 3.2.1 Generalising bounds computations

As discussed in section 3.1, for each data event Flamedisx must compute bounds on each hidden variable, determining the size of the tensors constructed. These must be large enough that all probability elements contributing non-negligibly to the sum in equation 3.1 are included, but not so large as to be redundantly including elements contributing close to 0. Flamedisx’s original implementation of this needed improvement for two reasons: firstly, the calculations did not fully account for fluctuations in all distributions, and so the bounds had to be made particularly wide to ensure that full range of relevance of each hidden variable was captured; secondly, the calculation to produce the bounds needed to be reproduced each time a new model block was added, which in the case of some of the additional blocks added for FlameNEST was non-trivial.

We generalised the bounds computation procedure in Flamedisx to calculate the bounds for each block’s input hidden variable,  $I$ , based on already calculated bounds for each block’s output hidden variable,  $O$ . Bayes’ theorem states

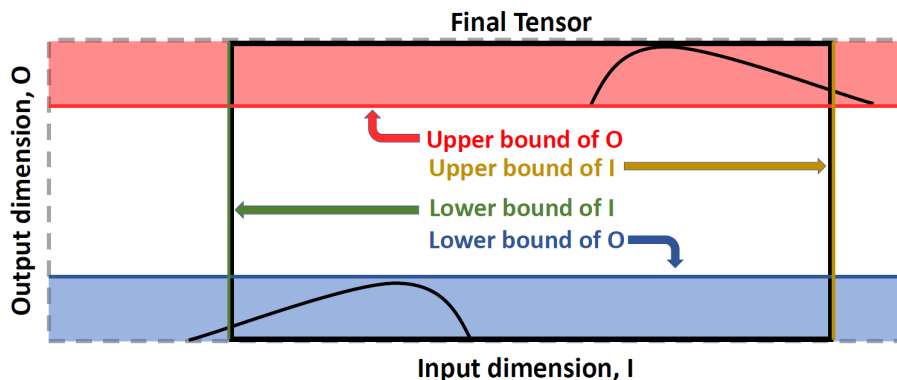
$$P(I = i|O = o) = \frac{P(O = o|I = i)P(I = i)}{P(O = o)}, \quad (3.3)$$

where the probability  $P(O = o|I = i)$  is evaluated across the support of the input hidden variable, or some sensible restriction of this domain, for the already calculated bound values of the output hidden variable; that is, to calculate the lower bound on  $I$ , the lower bound of  $O$  would be used, taking the converse for the upper bound. The prior probability  $P(I = i)$  is by default flat, but certain blocks can override this when it improves the bound calculation procedure to do so. The prior is estimated via drawing values of the hidden variable  $I$  from a large MC reservoir generated once during FlameNEST’s runtime, filtering as appropriate based on already computed bounds. An example of this for the FlameNEST block structure is given shortly.

Bounds on  $I$  can then be obtained by constructing the cumulative distribution function of the posterior probability  $P(I = i|O = o)$ , here denoted  $f(i)$ , over the support of  $I$ ,  $i \in \{\text{support}(f)_{\text{min}}, \text{support}(f)_{\text{max}}\}$ ,

$$F(x) = \frac{1}{\mathcal{N}} \sum_{i=\text{support}(f)_{\text{min}}}^x f(i), \quad (3.4)$$

with an appropriate normalisation factor  $\mathcal{N}$  chosen such that  $f(i)$  is normalised to 1 and we can set the denominator in equation 3.3 to unity. The lower and upper bounds are then taken as the values of  $x$  where  $F(x)$  evaluates to some user-defined low and high values of probability, where taking more extreme values corresponds to calculating wider bounds. This is depicted pictorially in figure 3.



**Figure 3.** Pictorial demonstration of the bounds computation for a block. The lower and upper bounds on the output dimension,  $O$ , are used to determine the input distributions,  $P(I = i, O = o_{\max})$  and  $P(I = i, O = o_{\min})$ , respectively, represented here as the black curves. We can determine the lower and upper bounds on the input dimension using these distributions depending on the max sigma chosen by the user. The final tensor is shown as a black box.

The method proceeds by computing the bounds for each block recursively — bounds on the outermost hidden variables are computed based on the observables, then the procedure outlined is repeated for each preceding block in turn until bounds are computed on all hidden variables. In the case of the FlameNEST block structure, we make two modifications to the above procedure, made to improve the accuracy of the tensor and energy stepping outlined in sections 3.2.2 and 3.2.3.

The first is making a manual calculation of the ion bounds. As we construct the central quanta tensor for various values of the energy, contracting over the ion dimension for each before summing them together, it is possible to choose the ion bounds to be different for each summed energy. Therefore the ion bounds are estimated directly as a function of energy for each summed tensor, as outlined in appendix C. Whilst in principle the Bayesian procedure could be used instead, it was found that a manual calculation in this case substantially improved performance, being of reliable accuracy due to the proximity of this hidden variable to the input dimension, energy.

The second change is that an additional bounds estimation is made for the energy values to be summed over when constructing the central quanta tensor. This is done by filtering an MC reservoir of electrons and photons produced from the source whose differential rate is being computed within the calculated electron/photon production bounds for each event. The distribution of energies from the remaining events is then used to estimate energy bounds by taking user-defined quantiles.

One can summarise the bounds computation for the FlameNEST block structure as follows. We use the Bayesian inversion procedure to calculate bounds for all hidden variables in the post-quanta blocks of figure 2, taking flat priors in each case. We then compute preliminary bounds on electrons and photons produced using the same procedure, taking a flat prior. Once these have been obtained, energy bounds can be obtained for each event using the procedure detailed above. These energy

bounds are then used together with the bounds on the outermost hidden variables — S1 and S2 photoelectrons detected — to obtain priors on electrons and photons produced. These are then used to obtain a second, tighter set of bounds on electrons and photons produced. Finally, ion bounds are computed using the procedure outlined in appendix C.

### 3.2.2 Variable tensor stepping

Originally, Flamedisx would construct each hidden variable dimension in integer steps of 1 between the computed bounds. This size of the tensors for high energy events, even for the original Flamedisx models, would thus become too large to fit in memory on many GPUs. For FlameNEST, the introduction of a number of additional post-quanta model blocks, as well as the pre-quanta block with an internally contracted dimension, greatly compounded this problem. In order to allow TensorFlow to hold all the tensors for the computation in memory and to speed up the Flamedisx computation, we implemented a variable stepping over the hidden variables.

A maximum size may be specified for any set of hidden variables, and if the difference between the upper and lower bounds for any events is greater than this, the tensors constructed for that event batch will have hidden variable dimensions increasing in integer steps greater than 1. These steps are chosen such that no hidden variable dimension goes above its maximum dimension size. Provided that all distributions computed over a stepped hidden variable are sufficiently smoothly varying over the stepped values, each calculated probability element may simply be re-scaled by the step size of its domain, with the overall computation then returning a result approximately the same as if no stepping had been done. Distributions of hidden variables can be inspected using FlameNEST’s MC generation tools to verify this, with quantitative validations such as that in section 4.1 confirming that appropriate maximum dimension sizes have been chosen.

### 3.2.3 Variable energy stepping

As detailed in sections 3.1 and 3.2.1, the green quanta tensor in figure 2 is constructed across energies between the energy bounds for each source/event pair. Provided the energy bounds are chosen to be wide enough, terms outside of the bounds will contribute negligibly to the sum over  $E$  in equation 3.1.

To further accelerate the computation, provided that the shape of the source’s energy spectrum is smoothly varying within these bounds, it is possible to obtain an accurate value of  $R^j(S1, S2, x, y, z, t)$  by taking larger steps in  $E$  in the sum, re-weighting each  $R^j(E)$  by the step size taken relative to the energy granularity of the spectrum. This is analogous to the variable tensor stepping described in section 3.2.2. Quantitative validations such as that in section 4.2 can verify that this has been done appropriately.

### 3.2.4 Model-dependent approximations

As discussed in sections 3.1.1 and 3.1.2, it is necessary to apply continuity corrections and account for the constraint that  $n_{\text{prod}}^{\text{el}} \leq n_{\text{prod}}^{\text{i}}$  to ensure good matching between the FlameNEST model implementation and the NEST MC models. However, above certain energy thresholds this becomes redundant, and has little effect on the accuracy of the computation. Therefore, both of these aspects are ignored when calculating quanta tensors above 5 keV for ER sources and 20 keV for NR sources. For current and future detectors with conditions different from the LUX defaults, the user may wish to verify that these thresholds remain sensible choices.

## 4 Validations

For the performance features outlined in section 3.2 to be used in practise, it must first be verified that they still produce accurate computed values of  $R^j(S1, S2, x, y, z, t)$  for all sources  $\{j\}$  of interest at a range of energies, whilst providing ample speedup to the computation. This section presents the results of a series of such validations.

### 4.1 Mono-energetic sources

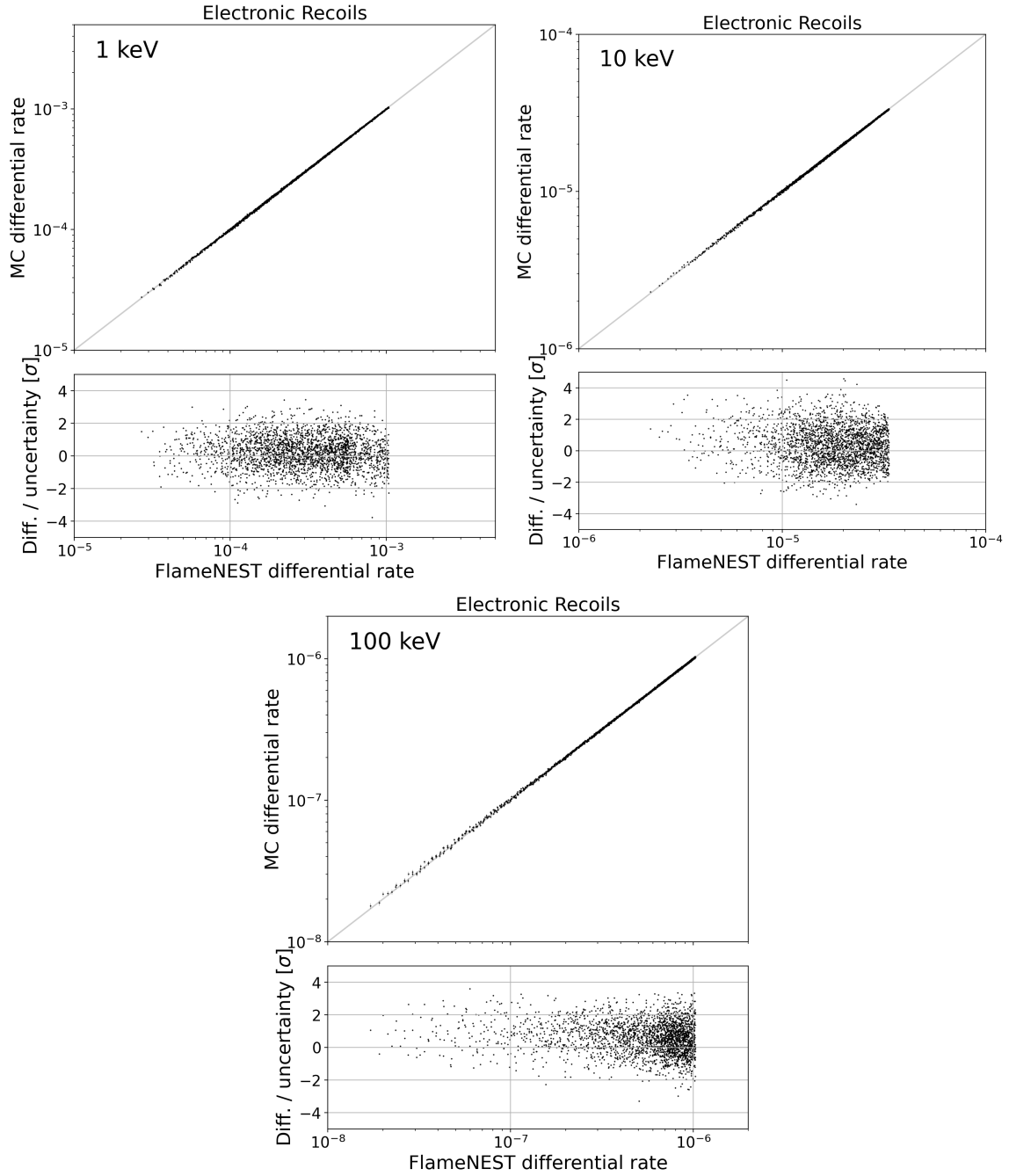
In order to validate the FlameNEST computation, we compare the differential rate computed with FlameNEST directly with an estimation from a finely binned, high statistics MC simulation using NEST v2.2.2 at 1, 10, and 100 keV energies. We use NEST to fill a two-dimensional histogram of  $(S1, S2)$  using mono-energetic sources at a fixed event position (the detector centre) and time, to avoid the computational cost of achieving sufficient simulation statistics with a 6-dimensional template, a reminder of why the Flamedisx computation is superior to a template computation. We set all parameters to the NEST defaults, which are based on the LUX detector's third science run [15]. The histogram is filled with  $1 \times 10^8$  NEST events with 50 logarithmically-spaced bins in both dimensions.

We pass the central  $(S1, S2)$  values for each bin along with the fixed position and time to the FlameNEST differential rate computation, giving us the differential rate at the centre of each bin. We then estimate the differential rate for each bin using the NEST Monte Carlo. We do this by taking the fraction of total events simulated falling within each bin, dividing by the bin volumes, and scaling by the expected number of events before selection cuts, which in our case is scaled to be 1.

We then calculate the difference between the differential rate estimated from the NEST-filled histogram and the FlameNEST differential rate, normalised by an estimate of the error in calculating the differential rate using a histogram. This includes an estimation of the (Poisson) error from finite simulation statistics in each bin, assuming bins are uncorrelated, and an estimation of the binning/discretisation error, obtained from the variation of the FlameNEST differential rate between the corners of each bin.

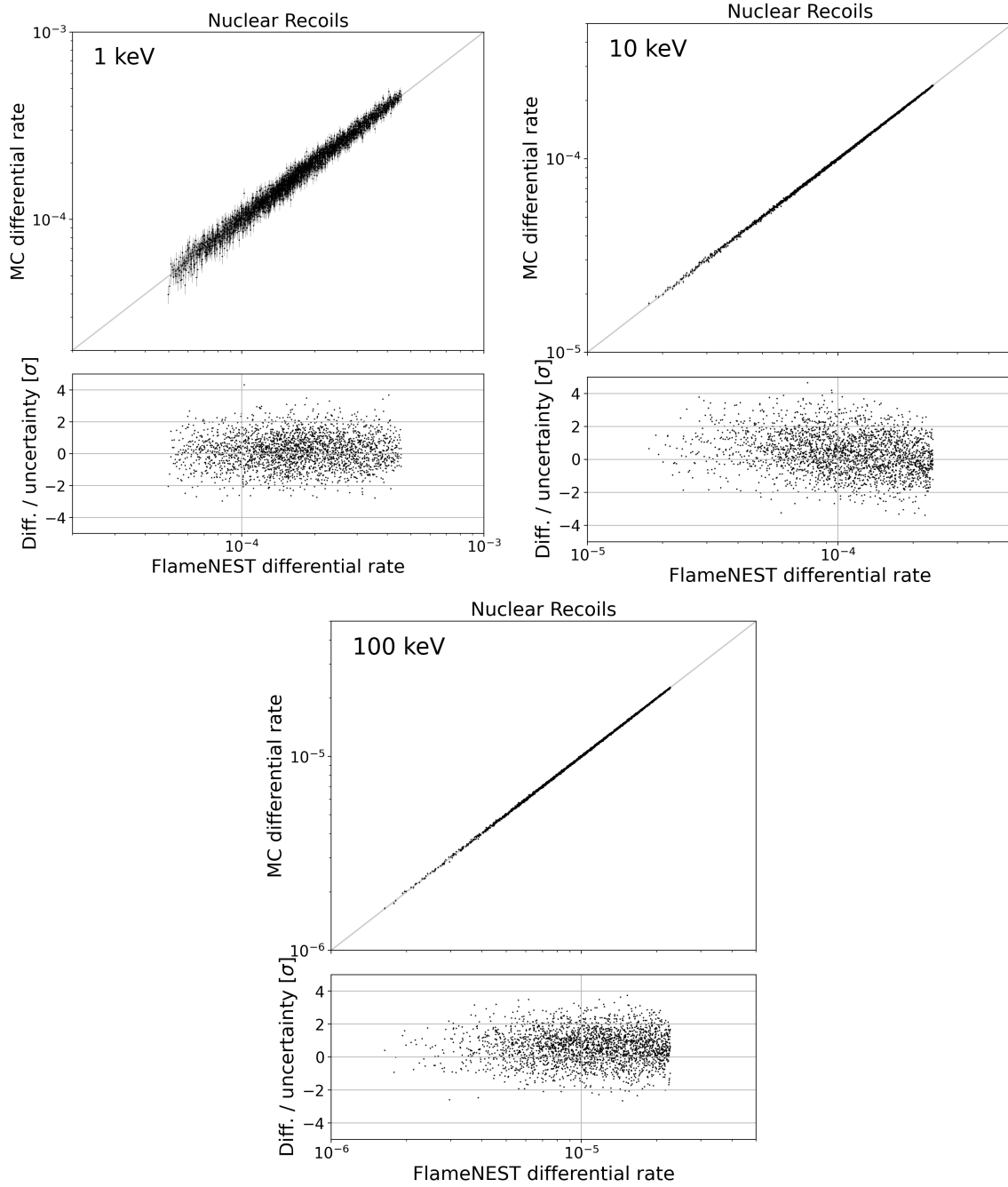
In the FlameNEST computation we take  $3\sigma$  bounds, such that the Bayesian bounds procedure uses probability corresponding to the  $3\sigma$  quantile of a Gaussian distribution, and choose all tensors to have a maximum dimension size of 70. Whilst these approximations will introduce some error in the calculation compared to the idealised case of infinite bounds and no stepping, if the difference between the FlameNEST result and a Monte Carlo template-estimated differential rate is sufficiently small, this can be accepted. The reason for this is twofold; firstly, parameters in the NEST models come with, in some cases, very large errors, and shifts in the differential rate coming from approximations in the FlameNEST computation can be absorbed by small shifts in these parameters. Secondly, MC templates come with their own errors: errors from finite simulation statistics, binning, and template interpolation as nuisance parameters are floated, meaning small errors in likelihood evaluation are not unique to FlameNEST.

Figure 4 and 5 show the comparison described above for mono-energetic ER and NR sources, respectively. Both ER and NR sources at all energies show a good agreement. Any small offsets or shape to the distributions are a result of the finite tensor bounds and the tensor stepping outlined in section 3.2, however they are within the errors inherent to template-based likelihood evaluation.



**Figure 4.** Difference between the FlameNEST and MC template differential rate for bins in S1/S2 space for 1, 10 and 100 keV mono-energetic ER source, fixed at the centre of the LUX detector, presented in terms of the estimated Poisson statistics + binning error from the MC template calculation.

We recommend this validation process is repeated when further model changes are implemented in FlameNEST. Smaller changes to models might not carry the same significance at all energies so we also recommend a wide scan in energy space. Whilst some of the NEST models have been validated up to the O(MeV) energies relevant for other physics searches with liquid xenon TPCs



**Figure 5.** Difference between the FlameNEST and MC template differential rate for bins in S1/S2 space for 1, 10 and 100 keV mono-energetic NR source, fixed at the centre of the LUX detector, presented in terms of the estimated Poisson statistics + binning error from the MC template calculation.

including neutrinoless double beta decay [16], validation of the FlameNEST computations at these energies is deferred for future work. Our current focus has been on validating that the framework can be used at energies relevant WIMP search, likely the first science to come out of current-generation experiments.



## 4.2 Full energy spectra

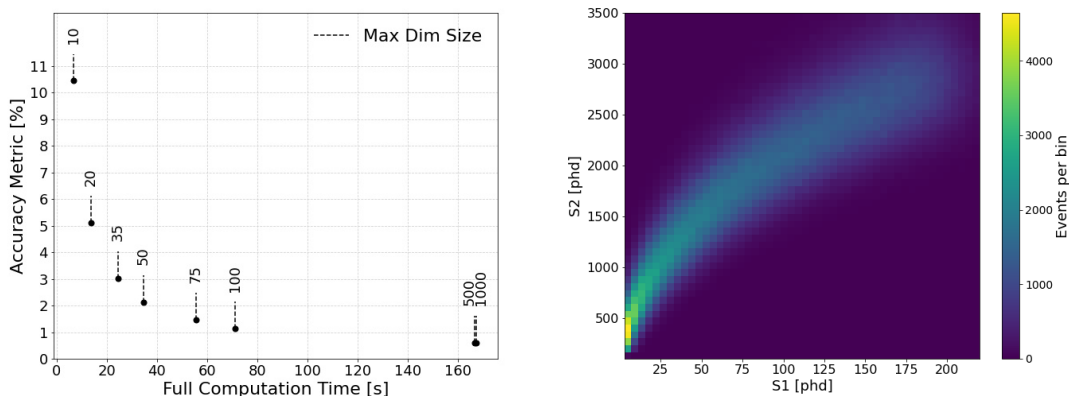
As described in section 3.2.3, FlameNEST will step over the energies remaining once the input spectrum of the source has been trimmed between the calculated energy bounds for each event (or batch of events). Here, we demonstrate how this stepping impacts the speed and accuracy of the computation. For ER and NR sources we run the same computation as in section 4.1, this time simulating a flat energy spectrum between 0–100 keV using NEST. When doing the FlameNEST computations we vary the maximum energy dimension size — this caps the size of the trimmed spectrum between the energy bounds, applying a stepping if the size of the trimmed spectrum is above the specified maximum. We set the full flat spectra used in the FlameNEST computation to be a comb of delta functions at 1000 points uniformly spaced in the energy range. All other parameters are the same as described in section 4.1, except for the maximum dimension size of the ions produced dimension, which is now capped at 30. We found that the resulting speed increase justified the minimal loss in accuracy of the FlameNEST computation, especially when the effects on accuracy of the energy stepping are accounted for.

To quantify the overall accuracy at different maximum energy dimension sizes, we define an accuracy metric,  $\Delta$ , over the template bins to be a weighted average over all bins of the percentage difference in differential rate between the FlameNEST computation and the template evaluation, weighted by the averaged differential rate of that bin, as in equation 4.1. The weights of the sum, i.e. the average differential rate between FlameNEST and the template, cancel with the denominator in the percentage difference for each bin, giving the expression a simple form. Here,  $R(S1, S2)_{\text{FN/MC}}$  denotes the differential rate at the bin with centre  $(S1, S2)$  using the FlameNEST / Monte Carlo template evaluation, and the sum is over all template bins. We chose this over the accuracy metric used in section 4.1 to mitigate the fact that for templates very large in  $S1/S2$  space, the method we used for estimating the Poisson errors begins to break down, and the correlation between bins due to discretisation begins to manifest as an offset in the residuals plot due to the inclusion of an estimate of the errors coming from this discretisation. This choice of metric also avoids the issue of most bins being empty for templates covering the full observable space when using such a broad energy spectrum.

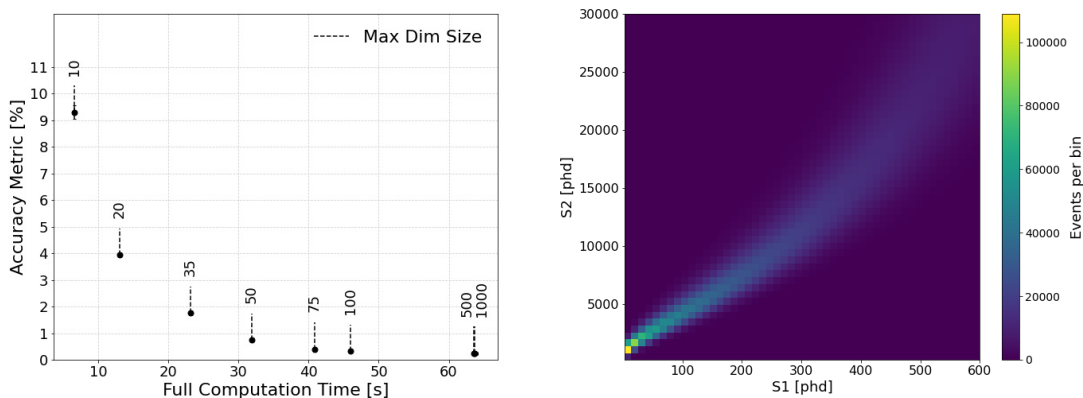
$$\Delta = \frac{\sum_{S1, S2} (R(S1, S2)_{\text{MC}} - R(S1, S2)_{\text{FN}})}{\sum_{S1, S2} \frac{1}{2} (R(S1, S2)_{\text{MC}} + R(S1, S2)_{\text{FN}})} \times 100\% \quad (4.1)$$

Figures 6 and 7 present the resulting accuracy metric value for each energy maximum dimension size, plotted against the computation time to evaluate the FlameNEST differential rate across bins for the ER and NR spectra shown. The computation is repeated for 10 separate NEST templates to estimate the variation seen. Bins with 0 MC template events are discarded from the computation; after doing so, approximately 1000 bins remained for the ER source and approximately 1750 bins remained for the NR source, out of 2500 bins used. The difference is a consequence of the different aspect ratios of the ER and NR bands. We benchmark using a Tesla P100 GPU.

Unsurprisingly the computation time increases as more energy steps are added, though perfect linearity is not seen as the number of events (bins) per computational batch is altered each time to maximise usage of the GPU memory. The accuracy metric behaves as expected; it is up to the user to decide the desired degree of accuracy, and to pay the corresponding cost in computation time.



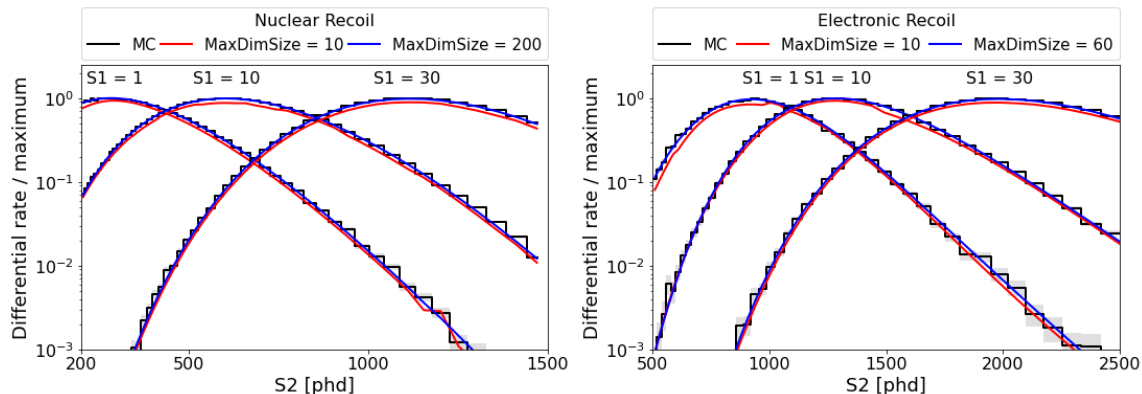
**Figure 6.** Accuracy metric vs full computation time for a range of different maximum energy dimension sizes for an NR source with a flat energy spectrum between 0.01 and 100 keV, using LUX detector parameters and fixed at the centre of this detector. The resulting (S1,S2) template used for one of the 10 comparisons is also shown. Approximately 1750 bins are used for the computation after the empty bins are removed.



**Figure 7.** Accuracy metric vs full computation time for a range of different maximum energy dimension sizes for an ER source with a flat energy spectrum between 0.01 and 100 keV, using LUX detector parameters and fixed at the centre of this detector. The resulting (S1,S2) template used for one of the 10 comparisons is also shown. Approximately 1000 bins are used for the computation after the empty bins are removed.

Saturation in time and accuracy is ultimately seen above a maximum energy dimension size; this happens when (for the majority of bins) the size of the input spectrum within the energy bounds is smaller than this maximum dimension size, rendering energy stepping redundant here. At this stage the remaining discrepancy in differential rate comes down to the other approximations made; the tensor stepping, the hidden variable and energy bounds computations and the number of terms used in the expansion of Owen’s T function, the calculation of which is necessary for the FlameNEST models (see appendix B).

The calculated accuracy metric will differ for energy spectra with more features; here, the user would likely want to implement a variable maximum energy dimension size, taking it to be larger



**Figure 8.** MC and FlameNEST differential rates over  $S_2$  bins of 3 different  $S_1$  slices of the templates shown in figures 6 and 7. We calculate the FlameNEST differential rates at two different maximum energy dimension sizes, to show the effect of this. We also depict for each bin the estimated Poisson statistics + binning error from the MC template calculation.

for events where the energy bounds cover regions of the spectrum with more features. Performing this same test would then allow them to validate that they are achieving sufficient accuracy for their source spectra.

Our choice to use a signed (weighted averaged) percentage difference as our accuracy metric has the potential to mask discrepancies if they are very large in every bin but average out across bins. To verify that this is not the case, we show in figure 8 the MC differential rate over  $S_2$  bins of 3 different  $S_1$  slices in each template, depicting also for each bin the estimated Poisson statistics plus binning error from the MC template calculation. We overlay the FlameNEST differential rates at two different maximum energy dimension sizes; a poor choice for each as well as the choice for each that takes the corresponding accuracy metric value below 1%. Clearly the discrepancy manifests visually as an overall shift, supporting our choice of accuracy metric. As expected, for the higher maximum dimension sizes, no discrepancy is visible beyond the MC errors, whereas for the low maximum dimension size (and thus greater sized energy spectrum steps), a significant shift is observed.

Finally, we wish to provide an absolute measure of the performance of FlameNEST. For a 0–10 keV ER source using a Tesla P100 GPU, we measure a differential rate computation time of 30ms per event, using a choice of 50 for the maximum energy dimension size following our findings in figure 7. It is important to reiterate that likelihood evaluation with 6 observables and multiple nuisance parameters is simply unfeasible using template methods, as the generation timescales for these templates become geological in magnitude. Whilst the evaluation of the likelihood here scales with the number of events, this method enables likelihood evaluations that would have an insurmountable barrier to overcome if template generation were required. The vastly improved accuracy and applicability of the NEST models in the Flamedisx framework enables such computations to be performed confidently in a range of experiments, justifying the slowdown in comparison to the original Flamedisx models. It should also be noted that for the time-consuming step of test statistic estimation, asymptotic estimation methods can be appealed to, and further optimisations may be possible even in the case of doing the full MC toy estimation procedure, as long as the accuracy is carefully tracked.

## 5 Conclusion

We present FlameNEST, an amalgamation of Flamedisx and NEST. The technical challenges of this union and the subsequent performance has been described in detail. FlameNEST will allow for high-dimensional likelihood evaluation, increasing the physics reach of LXe dual phase TPC experiments. Furthermore, the incorporation of the NEST models will reduce the need for involved modifications of the default models — designed for a specific detector and its conditions — when fitting real experimental data across a variety of detectors and potential variations in their operating conditions.

Inter-collaboration analyses have in the past been difficult due to software differences and the ways different experiments handle their nuisance parameters. We believe FlameNEST will make future inter-collaboration efforts much simpler by providing a robust framework which can be straightforwardly adapted to each experiment. The connections made with inter-collaboration analyses for current generation experiments will greatly facilitate the development of the next generation of noble element detection experiments, which in the case of LXe experiments will likely consist of a single, unified effort focused on one detector [17].

We point the reader to <https://github.com/FlamTeam/flamedisx>, where all of the FlameNEST code can be found within the original Flamedisx repository. Whilst FlameNEST can be run without a GPU, we do not recommend this for fitting with complex energy spectra or large numbers of events, due to the significant performance drop on a CPU. If GPUs are used, it should be noted that larger memory devices will enable larger batches of events to be computed and thus speed up the computation.

## Acknowledgments

Funding for this work is supported by the U.K. Science and Technology Facilities Council under the contract numbers ST/S000844/1, ST/S505675/1, ST/S000666/1, and ST/S555360/1. We acknowledge additional support from the Cosmoparticle Initiative at University College London, the UCL Cities Partnership programme, Stockholm University and the Kavli Institute for Particle Astrophysics and Cosmology.

We would like to thank Matthew Szydagis and Gregory Rischbieter of the University at Albany for their guidance and advice regarding the Noble Element Simulation Technique.

## A Model details

Here we provide a detailed description of the distributions and parameters in the FlameNEST block structure.

### A.1 Model parameters

In this section, we will define the parameters which are used in the FlameNEST distributions.

Table 1 lists the detector parameters which are typically measured or fixed and therefore unlikely to be floated as nuisance parameters in an analysis. It should be noted that the liquid electric field can in principle be position- and time-dependent.

**Table 1.** Physical, likely fixed, inputs to the FlameNEST model functions.

Symbol	Meaning
$T$	LXe temperature
$P$	LXe pressure
$\epsilon_{\text{liq}}(x, y, z, t)$	Liquid electric field
$\epsilon_{\text{gas}}$	Gas electric field
$z_{\text{topDrift}}$	Liquid/gas interface height
$\Delta_{\text{gas}}$	Distance between liquid/gas interface and anode
$N_{\text{PMT}}$	Number of PMTs

NEST uses some of the parameters in table 1 to calculate other fixed parameters used by the model functions. These are summarised in table 2.

**Table 2.** Calculated, likely fixed, quantities in the FlameNEST model functions.

Symbol	Meaning
$\rho_{\text{liq}}(T, P)$	Liquid xenon density
$\rho_{\text{gas}}(T, P)$	Gaseous xenon density
$v_{\text{drift}}(\epsilon_{\text{liq}}, \rho_{\text{liq}}, T)$	Electron drift velocity

Table 3 lists the parameters used by the model functions calculating the parameters of the yield probability distributions. They are all, directly or indirectly, functions of energy  $E$ , hence the need for the green tensor in figure 2 to be constructed for all relevant energies for an event and summed together.

Mean yields are calculated deterministically for both electrons and photons, along with the ratio of mean exciton yield to mean ion yield. The parameter  $\alpha$ , used as a distribution parameter for ER and NR, is defined as  $\alpha = (1 + r_{\text{ex}})^{-1}$ . The ER case calculates a ‘Fano factor’ to model over-dispersion in quanta production beyond Poisson statistics. Finally a number of parameters are calculated for modelling electron-ion recombination fluctuations. The parameters for both the ER and NR cases are functions of a number of (different) underlying nuisance parameters, which would likely be floated in a computation in the same way as the parameters in table 4.

The post-quanta model functions take a number of parameters that will likely only be determined approximately by auxiliary measurements and thus could be floated as nuisance parameters in a statistical analysis. Whilst many of these can be very well-constrained, treating them as effective parameters and allowing them to float may be necessary to achieve good fits to data when using these parametric models that don’t fully model PMT pulse production. Table 4 lists these.

A ‘Fano factor’ is used to account for an over-dispersion in S2 electroluminescence photons produced beyond Poisson statistics. The photon detection efficiencies determine the (binomial) detection probabilities for photons produced in liquid (S1) and gas (S2). Similarly the photoelectron detection efficiency determines the (binomial) detection probability for a single PMT to detect an (S1) photoelectron. The single photoelectron resolution coupled with the S1 and S2 noise terms determines the smearing of the final signals for a given number of detected photoelectrons due to

**Table 3.** Parameters for the FlameNEST yield distribution model functions.

Symbol	Meaning
$\overline{n^{\text{el}}}(E)$	Electron mean yield
$\overline{n^{\text{q}}}(E)$	Electron + photon mean yield
$r_{\text{ex}}(E)$	Ratio of mean exciton yield to mean ion yield
$\mathcal{F}_{\text{ER}}(\overline{n^{\text{q}}})$	ER Fano factor
$P_{\text{rec}}(\overline{n^{\text{el}}}, \overline{n^{\text{q}}}, r_{\text{ex}})$	Electron-ion recombination probability
$\xi(\overline{n^{\text{q}}})$	Electron-ion recombination skewness parameter
$\sigma_{\text{rec}}(\overline{n^{\text{el}}}, \overline{n^{\text{q}}}, P_{\text{rec}}, n_{\text{prod}}^{\text{i}})$	Electron-ion recombination width
$\delta\sigma(\xi)$	Electron-ion recombination width correction
$\delta\mu(\xi, \sigma, \delta\sigma)$	Electron-ion recombination mean correction

PMT effects and electronics noise.

**Table 4.** Parameters that a user may wish to float in the post-quanta FlameNEST model functions.

Symbol	Meaning
$P_{\text{dpe}}$	Double photoelectron emission probability
$\tau$	Electron lifetime
$\mathcal{F}_{\text{S2}}$	S2 Fano factor
$g1$	Photon detection efficiency in liquid at detector centre
$g1_{\text{gas}}$	Photon detection efficiency in gas
$\mu_{\text{spe}}$	Single photoelectron detection efficiency
$\sigma_{\text{spe}}$	Single photoelectron resolution
$\Delta_{\text{S1}}$	S1 noise
$\Delta_{\text{S2}}$	S2 noise

Acceptance cuts are applied to the detected signals which may be accounted for in the models in the same way as the original Flamedisx structure. Parameters determining these are summarised in table 5. The minimum photon cut currently approximates a PMT hit coincidence requirement. A better approximation to this, using an additional binomial process, will be implemented in the near future.

## A.2 Pre-quanta models

In this section, we provide the full description of the pre-quanta models implemented in FlameNEST. Equations (A.1)–(A.3) list the probability distributions used to calculate the pre-quanta model block in the ER case. Throughout this section and section A.3, we use the following notation: Normal denotes a normal distribution, Binom a Binomial distribution and SkewNormal a skew normal distribution. A tilde denotes an applied continuity correction, whilst a hat denotes the condition  $n_{\text{prod}}^{\text{el}} \leq n_{\text{prod}}^{\text{i}}$  discussed in the main text being applied at the level of the distribution. This is detailed

**Table 5.** Selection parameters.

Symbol	Meaning
$S1_{\min}$	Minimum S1 acceptance
$S1_{\max}$	Maximum S1 acceptance
$S2_{\min}$	Minimum S2 acceptance
$S2_{\max}$	Maximum S2 acceptance
$\gamma_{\min}$	Minimum photons detected

more in appendix B.

$$P(n_{\text{prod}}^{\text{q}} | \bar{n}^{\text{q}})_{\text{ER}} = \overline{\text{Normal}} \left( n_{\text{prod}}^{\text{q}} | \bar{n}^{\text{q}}, \sqrt{\mathcal{F}_{\text{ER}} \bar{n}^{\text{q}}} \right) \quad (\text{A.1})$$

$$P(n_{\text{prod}}^{\text{i}} | n_{\text{prod}}^{\text{q}})_{\text{ER}} = \text{Binom} \left( n_{\text{prod}}^{\text{i}} | n_{\text{prod}}^{\text{q}}, \alpha \right) \quad (\text{A.2})$$

$$P(n_{\text{prod}}^{\text{el}} | n_{\text{prod}}^{\text{i}})_{\text{ER}} = \overline{\text{SkewNormal}} \left( n_{\text{prod}}^{\text{el}} | (1 - P_{\text{rec}}) n_{\text{prod}}^{\text{i}} - \delta\mu, \frac{\sigma_{\text{rec}}}{\delta\sigma}, \xi \right) \quad (\text{A.3})$$

The distributions used to calculate the pre-quanta model block for NR interactions are listed in equations (A.4)–(A.6). We currently do not include Fano factors in equations (A.4) and (A.5), as allowed for within the NEST code, as both are set by default to 1, but in principle these could easily be added and included as additional nuisance parameters. As discussed in section 3.1.1, excitons do not need to be tracked as the number of these is fixed by the number of electrons, photons and ions.

$$P(n_{\text{prod}}^{\text{i}} | \bar{n}^{\text{q}})_{\text{NR}} = \overline{\text{Normal}} \left( n_{\text{prod}}^{\text{i}} | \alpha \bar{n}^{\text{q}}, \sqrt{\alpha \bar{n}^{\text{q}}} \right) \quad (\text{A.4})$$

$$P(n_{\text{prod}}^{\text{q}} | \bar{n}^{\text{q}}, n_{\text{prod}}^{\text{i}})_{\text{NR}} = \overline{\text{Normal}} \left( n_{\text{prod}}^{\text{q}} - n_{\text{prod}}^{\text{i}} | \alpha \bar{n}^{\text{q}} r_{\text{ex}}, \sqrt{\alpha \bar{n}^{\text{q}} r_{\text{ex}}} \right) \quad (\text{A.5})$$

$$P(n_{\text{prod}}^{\text{el}} | n_{\text{prod}}^{\text{i}})_{\text{NR}} = \overline{\text{SkewNormal}} \left( n_{\text{prod}}^{\text{el}} | (1 - P_{\text{rec}}) n_{\text{prod}}^{\text{i}} - \delta\mu, \frac{\sigma_{\text{rec}}}{\delta\sigma}, \xi \right) \quad (\text{A.6})$$

### A.3 Post-quanta models

In this section, we provide the precise post-quanta model descriptions implemented in FlameNEST. Equations A.7 to A.10 list the distributions describing the blocks going from produced photons to S1 signal, depicted in the lower row of the post-quanta blocks in figure 2. It should be noted that the original NEST models perform the final smearing as a two-step process, whereas we use the well-known property of two subsequent normal smearings to model this as a single step, adding the variances in quadrature.

$$P(n_{\text{det}}^{\text{ph}} | n_{\text{prod}}^{\text{ph}}) = \zeta^{\text{ph}}(n_{\text{det}}^{\text{ph}}, \gamma_{\min}) \text{Binom} \left( n_{\text{det}}^{\text{ph}} | n_{\text{prod}}^{\text{ph}}, g1f_{S1}(r, z) \right) \quad (\text{A.7})$$

$$P(n_{\text{prod}}^{\text{phel}} | n_{\text{det}}^{\text{ph}}) = \text{Binom} \left( n_{\text{prod}}^{\text{phel}} - n_{\text{det}}^{\text{ph}} | n_{\text{det}}^{\text{ph}}, P_{\text{dpe}} \right) \quad (\text{A.8})$$

$$P(n_{\text{det}}^{\text{phel}} | n_{\text{prod}}^{\text{phel}}) = \text{Binom} \left( n_{\text{det}}^{\text{phel}} | n_{\text{prod}}^{\text{phel}}, P_{\text{spe}}(\mu_{\text{spe}}, N_{\text{PMT}}) \right) \quad (\text{A.9})$$

$$P(S1 | n_{\text{det}}^{\text{phel}}) = \xi^{S1}(S1, S1_{\min}, S1_{\max}) \text{Normal} \left( S1 | n_{\text{det}}^{\text{phel}}, \sqrt{\sigma_{\text{spe}}^2 n_{\text{det}}^{\text{phel}} + \Delta_{S1}^2 (n_{\text{det}}^{\text{phel}})^2} \right) \quad (\text{A.10})$$



Equations A.11 to A.15 list the distributions corresponding to the upper row of post-quanta model blocks in figure 2, going from produced electrons to S2 signal.

$$P(n_{\text{det}}^{\text{el}} | n_{\text{prod}}^{\text{el}}) = \text{Binom} \left( n_{\text{det}}^{\text{el}} | n_{\text{prod}}^{\text{el}}, \eta^{\text{el}}(z, z_{\text{topDrift}}, v_{\text{drift}}, \tau, \epsilon_{\text{gas}}) \right) \quad (\text{A.11})$$

$$P(n_{\text{prod}}^{\text{S2-ph}} | n_{\text{det}}^{\text{el}}) = \overline{\text{Normal}} \left( n_{\text{prod}}^{\text{S2-ph}} | \mu_{\text{el}}(\epsilon_{\text{gas}}, \rho_{\text{gas}}, \Delta_{\text{gas}}) n_{\text{det}}^{\text{el}}, \sigma_{\text{el}}(\epsilon_{\text{gas}}, \rho_{\text{gas}}, \Delta_{\text{gas}}, \mathcal{F}_{\text{S2}}) \sqrt{n_{\text{det}}^{\text{el}}} \right) \quad (\text{A.12})$$

$$P(n_{\text{det}}^{\text{S2-ph}} | n_{\text{prod}}^{\text{S2-ph}}) = \text{Binom} \left( n_{\text{det}}^{\text{S2-ph}} | n_{\text{prod}}^{\text{S2-ph}}, g_{1\text{gas}} f_{\text{S2}}(r) \right) \quad (\text{A.13})$$

$$P(n^{\text{S2-phel}} | n_{\text{det}}^{\text{S2-ph}}) = \text{Binom} \left( n^{\text{S2-phel}} - n_{\text{det}}^{\text{S2-ph}} | n_{\text{det}}^{\text{S2-ph}}, P_{\text{dpe}} \right) \quad (\text{A.14})$$

$$P(\text{S2} | n^{\text{S2-phel}}) = \xi^{\text{S2}}(\text{S2}, \text{S2}_{\text{min}}, \text{S2}_{\text{max}}) \text{Normal} \left( \text{S2} | n^{\text{S2-phel}}, \sqrt{\sigma_{\text{spe}}^2 n^{\text{S2-phel}} + \Delta_{\text{S2}}^2 (n^{\text{S2-phel}})^2} \right) \quad (\text{A.15})$$

## B Modified skew Gaussian to implement NEST constraint

As discussed in the main text, NEST implements the condition that  $n_{\text{prod}}^{\text{el}} \leq n_{\text{prod}}^{\text{i}}$ . We account for this in FlameNEST by modifying the skew Gaussian PDF as follows. The PDF for a standard skew Gaussian distribution with mean  $\mu$ , standard deviation  $\sigma$  and skewness parameter  $\alpha$  takes the form

$$f(x; \mu, \sigma, \alpha) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \left(1 + \text{erf}\left[\left(\frac{\alpha}{\sqrt{2}\sigma}\right)(x-\mu)\right]\right). \quad (\text{B.1})$$

In FlameNEST, we modify this to read

$$f(x; \mu, \sigma, \alpha, l) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \left(1 + \text{erf}\left[\left(\frac{\alpha}{\sqrt{2}\sigma}\right)(x-\mu)\right]\right) & x < l \\ 1 - \left\{ \frac{1}{2} \left(1 + \text{erf}\left[\frac{x-\mu}{\sqrt{2}\sigma}\right]\right) - 2T\left(\frac{x-\mu}{\sigma}, \alpha\right) \right\} & x = l \\ 0 & x > l \end{cases} \quad (\text{B.2})$$

where  $x$  maps to  $n_{\text{prod}}^{\text{el}}$  and  $l$  to  $n_{\text{prod}}^{\text{i}}$ . The term in curly brackets in the  $x = l$  case is the cumulative distribution function (CDF) of the skew Gaussian distribution, and  $T$  is Owen's  $T$  function [18]. This has the effect of 're-dumping' all probability mass for  $x > l$  into the probability mass at  $x = l$ , once a continuity correction is applied as in equation 3.2, which is an appropriate capturing of NEST's MC behaviour, setting any sampled  $n_{\text{prod}}^{\text{el}} > n_{\text{prod}}^{\text{i}}$  to be equal to  $n_{\text{prod}}^{\text{i}}$ .

Implementing this as a TensorFlow computation required adding a custom distribution to the TensorFlow Probability library [10]. Of particular importance was an efficient evaluation of Owen's  $T$  function  $T(h, a)$ , which is the integral

$$T(h, a) = \frac{1}{2\pi} \int_0^a \frac{e^{-\frac{1}{2}h^2(1+x^2)}}{1+x^2} dx. \quad (\text{B.3})$$

In our case  $a \geq 0$ . Owen proved the relation [18]

$$T(h, a) = \frac{1}{2}\Phi(h) + \frac{1}{2}\Phi(ah) - \Phi(h)\Phi(ah) - T\left(ah, \frac{1}{a}\right), \quad (\text{B.4})$$

where  $\Phi$  is the CDF of the standard normal distribution, and so we can always recast  $T(h, a)$  to be in  $0 \leq a \leq 1$ . It is then straightforward to perform a Taylor expansion in  $a$

$$T(h, a) = \frac{1}{2\pi} \left\{ \tan^{-1}(a) + \sum_{i=1}^{\infty} C_i \frac{a^{2i-1}}{2i-1} \right\}, \quad (\text{B.5})$$

where the coefficients are obtained recursively as

$$\begin{aligned} C_1 &= e^{-\frac{h^2}{2}} - 1, \\ C_{n+1} &= -C_n + (-1)^n \frac{\left(\frac{h^2}{2}\right)^n}{n!} e^{-\frac{h^2}{2}}. \end{aligned} \quad (\text{B.6})$$

We determined that in our application of equation B.2 a sufficient degree of accuracy could be obtained for all relevant parameter values with a truncation of the series at  $C_2$  for NR sources and  $C_5$  for ER sources. This is evident from the results in sections 4.1 and 4.2.

### C Manual ion bound computation in FLameNEST

As discussed in the main text, for the FLameNEST block structure a manual calculation is done for the ion bounds, constructing different bounds for each energy summed over in the quanta tensor. In the ER case, the following quantities are first calculated, representing bounds on  $n_{\text{prod}}^q$ , coming from distribution in equation A.1,

$$n_{\text{upper}}^q = \bar{n}^q + \sigma \sqrt{\mathcal{F} \bar{n}^q} \quad (\text{C.1})$$

$$n_{\text{lower}}^q = \bar{n}^q - \sigma \sqrt{\mathcal{F} \bar{n}^q}. \quad (\text{C.2})$$

All symbols have the same meaning as in appendix A.1, and  $\sigma$  is a user-defined parameter controlling the width of the bounds. It should be noted that energy enters implicitly in  $\bar{n}^q$ . Upper and lower bounds on the mean and standard deviation of the number of ions described by the binomial of equation A.2 are then calculated as

$$\mu_{\text{upper/lower}} = n_{\text{upper/lower}}^q \alpha \quad (\text{C.3})$$

$$\sigma_{\text{upper/lower}} = \sqrt{n_{\text{upper/lower}}^q \alpha (1 - \alpha)}. \quad (\text{C.4})$$

In the NR case, the upper and lower bounds on the mean and standard deviation of the number of ions described by the normal distribution of equation A.4 can simply be calculated as

$$\mu_{\text{upper}} = \mu_{\text{lower}} = \bar{n}^q \alpha \quad (\text{C.5})$$

$$\sigma_{\text{upper}} = \sigma_{\text{lower}} = \sqrt{\bar{n}^q \alpha}. \quad (\text{C.6})$$

Then, upper and lower bounds on the number of ions can be calculated straightforwardly as

$$n_{\text{min}}^i = \mu_{\text{lower}} - \sigma \sigma_{\text{lower}} \quad (\text{C.7})$$

$$n_{\text{max}}^i = \mu_{\text{upper}} + \sigma \sigma_{\text{upper}}. \quad (\text{C.8})$$

## References

- [1] K. Garrett and G. Duda, *Dark Matter: A Primer*, *Adv. Astron.* **2011** (2011) 968283 [[arXiv:1006.2483](#)].
- [2] PLANCK collaboration, *Planck 2018 results. VI. Cosmological parameters*, *Astron. Astrophys.* **641** (2020) A6 [*Erratum ibid.* **652** (2021) C4] [[arXiv:1807.06209](#)].
- [3] LUX collaboration, *Results from a search for dark matter in the complete LUX exposure*, *Phys. Rev. Lett.* **118** (2017) 021303 [[arXiv:1608.07648](#)].
- [4] XENON collaboration, *XENONIT dark matter data analysis: Signal and background models and statistical inference*, *Phys. Rev. D* **99** (2019) 112009 [[arXiv:1902.11297](#)].
- [5] LUX-ZEPLIN collaboration, *Projected WIMP sensitivity of the LUX-ZEPLIN dark matter experiment*, *Phys. Rev. D* **101** (2020) 052002 [[arXiv:1802.06039](#)].
- [6] XENON collaboration, *Projected WIMP sensitivity of the XENONnT dark matter experiment*, *JCAP* **11** (2020) 031 [[arXiv:2007.08796](#)].
- [7] D. Baxter et al., *Recommended conventions for reporting results from direct dark matter searches*, *Eur. Phys. J. C* **81** (2021) 907 [[arXiv:2105.00599](#)].
- [8] LUX collaboration, *Fast and Flexible Analysis of Direct Dark Matter Search Data with Machine Learning*, [arXiv:2201.05734](#).
- [9] J. Aalbers, B. Pelssers, V.C. Antochi, P.L. Tan and J. Conrad, *Finding dark matter faster with explicit profile likelihoods*, *Phys. Rev. D* **102** (2020) 072010 [[arXiv:2003.12483](#)].
- [10] TensorFlow Developers, *Tensorflow*, 2021 <https://doi.org/10.5281/zenodo.5095721>.
- [11] XENON collaboration, *The XENONIT Dark Matter Experiment*, *Eur. Phys. J. C* **77** (2017) 881 [[arXiv:1708.07051](#)].
- [12] M. Szydagis, S. Andalaro, J. Balajthy, G. Block, J. Brodsky, J. Cutter et al., *Noble Element Simulation Technique*, 2021, <https://doi.org/10.5281/zenodo.5080263>.
- [13] C.H. Faham, V.M. Gehman, A. Currie, A. Dobi, P. Sorensen and R.J. Gaitskell, *Measurements of wavelength-dependent double photoelectron emission from single photons in VUV-sensitive photomultiplier tubes*, *2015 JINST* **10** P09010 [[arXiv:1506.08748](#)].
- [14] M. Szydagis, N. Barry, K. Kazkaz, J. Mock, D. Stolp, M. Sweany et al., *NEST: A Comprehensive Model for Scintillation Yield in Liquid Xenon*, *2011 JINST* **6** P10002 [[arXiv:1106.1613](#)].
- [15] LUX collaboration, *First results from the LUX dark matter experiment at the Sanford Underground Research Facility*, *Phys. Rev. Lett.* **112** (2014) 091303 [[arXiv:1310.8214](#)].
- [16] LUX collaboration, *Improved modeling of  $\beta$  electronic recoils in liquid xenon using LUX calibration data*, *2020 JINST* **15** T02007 [[arXiv:1910.04211](#)].
- [17] J. Aalbers et al., *A Next-Generation Liquid Xenon Observatory for Dark Matter and Neutrino Physics*, [arXiv:2203.02309](#).
- [18] D.B. Owen, *Tables for Computing Bivariate Normal Probabilities*, *Ann. Math. Stat.* **27** (1956) 1075.