SANDIA REPORT
SAND2024-02143
Printed February 2024

Sandia National Laboratories

A Semi-Supervised Learning Method to Produce Explainable Radioisotope Proportion Estimates for NaI-based Synthetic and Measured Gamma Spectra

Alan J. Van Omen & Tyler J. Morrow

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185
Livermore, California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from

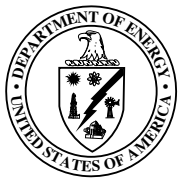
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Road
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods



ABSTRACT

Quantifying the radioactive sources present in gamma spectra is an ever-present and growing national security mission and a time-consuming process for human analysts. While machine learning models exist that are trained to estimate radioisotope proportions in gamma spectra, few address the eventual need to provide explanatory outputs beyond the estimation task. In this work, we develop two machine learning models for a NaI detector measurements: one to perform the estimation task, and the other to characterize the first model's ability to provide reasonable estimates. To ensure the first model exhibits a behavior that can be characterized by the second model, the first model is trained using a custom, semi-supervised loss function which constrains proportion estimates to be explainable in terms of a spectral reconstruction. The second auxiliary model is an out-of-distribution detection function (a type of meta-model) leveraging the proportion estimates of the first model to identify when a spectrum is sufficiently unique from the training domain and thus is out-of-scope for the model. In demonstrating the efficacy of this approach, we encourage the use of meta-models to better explain ML outputs used in radiation detection and increase trust.

This page intentionally left blank.

ACKNOWLEDGMENT

This work was funded by the U.S. Department of Energy, National Nuclear Security Administration, Office of Defense Nuclear Nonproliferation Research and Development (DNN R&D). Laboratory resources were provided by the GADRAS team and the Radiological Assistance Program (RAP) at Sandia National Laboratories. Additional subject matter expertise was generously provided by Elliott Leonard and Paul Thelen.

This page intentionally left blank.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page intentionally left blank.

ACRONYMS & DEFINITIONS

ANN Artificial neural network

BG Background, a gamma spectrum with counts only from one or more background sources

cps Counts per second

DRF Detector Response Function

FG Foreground, a gamma spectrum with counts only from one or more non-background sources

FNR False Negative Rate

FPR False Positive Rate

GADRAS Gamma Detector Response and Analysis Software

HPGe High Purity Germanium, a type of semiconductor radiation detector

IND In-Distribution, gamma spectra within the scope of a model or algorithm

JSD Jensen-Shannon Distance

LPE Label Proportion Estimation

MAE Mean Absolute Error

ML Machine Learning

MLP Multi-Layer Perceptron

NaI Sodium Iodide, a type of scintillating radiation detector

OOD Out-Of-Distribution, gamma spectra outside the scope of a model or algorithm

PVT Polyvinyltoluene, a type of plastic scintillating radiation detector

PyRIID Sandia's Open-Source Python package of RIID-related software utilities

RIID Radioisotope Identification

SNL Sandia National Laboratories

SNM Special Nuclear Material

This page intentionally left blank.

1. INTRODUCTION

Radioisotope identification (RIID) of the radioactive sources present in gamma spectra, as well as quantifying them, is a continued focus of environmental monitoring for industry, medicine, and energy production, as well as national security concerns related to border monitoring, arms control, treaty verification, emergency response, and consequence management. Measured gamma spectra from these applications can contain contributions from multiple sources simultaneously, in which case a multi-class identification model may not be sufficient to fully explain a measurement as only the most prominent source would be predicted. Furthermore, simply detecting the presence of multiple sources, while helpful, may be inadequate if the relative isotopic proportions are key to assess urgency. For human subject-matter experts (SMEs), estimating the relative contributions of different sources is often conducted in a time-intensive, software assisted process. The process, at a high level, amounts to finding a selection of source signatures which best explains the measurement, or from another perspective, finding the explanation which minimizes some goodness-of-fit metric. To assist in reducing analysis time, which is particularly important when faced with large quantities of data, machine learning (ML) is becoming more frequently employed to fit models to a space of spectra rather than perform a search of that space later to find the best fit. What remain missing from prior ML-based isotopic analysis solutions, and underpins continued criticism of ML in radiation detection, are user-facing explanatory results that are key to establishing trust between human analysts and their software. As such, we contend that incorporating traditional goodness-of-fit tests or distance metrics into ML, whether to explain or train them better, is the most constructive path forward.

To address this challenge, we fit a machine learning (ML) model to perform label proportion estimation (LPE) (where labels are radioisotopes) on a predefined space of possible gamma spectra (carried out as an optimization problem not unlike the aforementioned search procedure), along with an auxiliary model which alerts a user if an encountered gamma spectrum is sufficiently different from training, i.e., out-of-distribution (OOD). The first model which performs isotopic proportion estimation is a neural network, or more specifically a multilayer perceptron (MLP) acting as a regressor, which from this point forward we will simply refer to as the LPE model. The auxiliary model used for OOD detection is a cubic polynomial fit (via a U-spline function) which leverages the output of the LPE model, which from this point forward we will simply refer to as the OOD detection model or OOD detector. Both models are trained and tuned on a synthetically-defined problem space, and are then tested on lab-measured data. Our approach is novel as we utilize a custom learning objective to train the LPE model which constrains its estimates to be explainable in terms of a spectral reconstruction error from comparing a reconstruction to the input sample. In particular, this is done by minimizing a semi-supervised loss function consisting of a supervised term for the LPE task and an unsupervised term to promote spectra reconstruction and the ability to fit an OOD detection model. The OOD detector is based on the hypothesis that in-distribution (IND) samples and their reconstructions exhibit a trend noticeably different from OOD spectral

reconstructions given sufficient signal-to-noise, and leverages the outputs of the LPE model to make a decision.

The technique presented here is a NaI-based validation using measured data of the work conducted in [1], which, to the authors' knowledge, is the first time a semi-supervised learning technique had been applied to this problem space for increased explain-ability of radioisotopes estimates. However, the technique presented here differs in that a poorer resolution detector is used (NaI) to assess the isotopic analysis capabilities on a relatively inexpensive detector, and while significantly fewer sources are considered, a greater number of measured spectra are used in testing to validate the method. While various other machine learning (ML) methods have been applied to identifying multiple isotopes, none have attempted to unify the isotopic analysis with OOD as we have done here.

The remainder of this report is organized as follows:

- Section 2 briefly discusses gamma spectroscopy, useful tools for synthesizing physics-based gamma spectra, concepts related to the presented methods, and related works.
- Section 3 describes all methods in terms of real and synthetic detectors, environments, and data, as well as the models themselves and how they were trained.
- Section 4 presents the results of the trained models on measured spectra.
- Section 5 offers our recommendation regarding usage scenarios for the models.
- Section 6 provides commentary on conclusions to be drawn from the results.

2. BACKGROUND

2.1. Gamma Spectroscopy

Radioisotopes, which exist in an unstable or excited state, spontaneously decay to a more stable state which can include transformation into another element. During this transformation, various types of radiation can be emitted including neutrons, alphas, betas, and photons. For this study we are only concerned with high energy electromagnetic radiation from about 40 keV to 3 MeV, which includes some hard X-rays and gamma rays. Gammas can be used for radioisotope identification because when emitted, they are of discrete energies, and can typically be resolved in spectra from a gamma detector. A gamma detector, such as the sodium-iodide (NaI) detector used for this study, collects the energy deposited by gammas and converts it to an electrical signal proportional to the original deposited energy. Each energy deposition, colloquially referred to as a *count*, is then binned by energy level to build up a histogram known as a gamma spectrum.

Gamma spectra analysis can be very complicated due to the nature of photon interactions and other physical phenomena observable with modern detectors. All of these phenomena cause the detector to be sensitive to the scattering environment, source location, gain changes, and background radiation fields, among others. Each sensitivity can interact in complicated, additive and subtractive ways that confound human and algorithmic analyses alike. The details and physics around gamma spectrometry are well-described in Knoll [2].

With all of these challenges in mind, gamma spectrum analysis generally still relies on experienced subject-matter experts (SME), called gamma spectroscopists, who have, over time, become more software-assisted. Increased use of software is a necessity due to the increasing complexity of problems that gamma spectroscopists are employed to solve. Software automation has worked its way into their workflow to save time on nearly all steps of analysis such as photo-peak identification, background subtraction, and template matching [3], which has moved the field toward a traditional, artificial intelligence approach. Even optimization processes, the core of ML, are already common in radiation detection and transport software which seek to minimize some useful objective function, such as chi-squared [4].

2.2. Software

In the last decade, ML approaches to RIID have become increasingly popular in research and have shown promise in a number of problem spaces (section 2.5), potentially serving as yet another supplement to analysis workflows. Such research has been accelerated by the availability of modern software tools which can generate the large amounts of synthetic gamma spectra that ML requires.

This section discusses two such pieces of software that were used to carry out the work in this report.

2.2.1. GADRAS

The Gamma Detector Response and Analysis Software (GADRAS) uses Detector Response Functions (DRFs) to simulate the output of real gamma and neutron detectors when they are exposed to sources of photon and neutron radiation. Characterization of a physical detector to obtain a DRF is performed by fitting various detector-describing parameters to well-formed gamma spectra based on long collects of calibration sources with photopeaks spanning the full range of energy. More about DRFs can be found in [4].

2.2.2. PyRIID

PyRIID (pronounced: PIE-rid) is a Python package that facilitates gamma spectrum synthesis, model training, and visualizations for ML-based radioisotope identification [5]. Synthetic data generation occurs in three stages:

1. Seed Synthesis: the first step is to obtain source templates (without Poisson noise) on which to fit or test a model. PyRIID does this by providing a Python wrapper around the GADRAS API which accesses its Inject capabilities. While assumptions about sources, detector, and environment are all made here, the ability to vary any parameter is provided. From this stage, the user will end up with two sets of seeds: those intended as foregrounds and those intended as backgrounds. The eventual goal is to combine these seeds to form gross spectra. From this standpoint, PyRIID's terminology is that foreground is source-only and background is background-only. The authors recognize that in some circles foreground is synonymous with gross, but not here.
2. Seed Mixing (optional): the obtained seeds are then summed together in proportions randomly generated from a Dirichlet distribution. The most common use of the mixer is to construct a variety of background samples from base seeds of PotassiumInSoil, ThoriumInSoil, UraniumInSoil, and Cosmic.
3. Static Synthesis: seeds, or mixed versions of them if that was desired, are then randomly varied in terms of signal-to-noise ratio and live time for an expected background rate to obtain gross spectra. Poisson noise is applied by default. Sample-wise background spectra are also readily available to make it easy to obtain background-subtracted (AKA, foreground) samples via a simple arithmetic operation.

Throughout the synthetic process, careful ground truth tracking is performed to preserve the source or sources which have contributed to each spectrum and the precise proportions in which they have done so. Moreover, source descriptions are, at the time of writing, tracked at three levels forming a source hierarchy: category (SNM, medical, industrial, etc.), isotope, and seed. Through standard data transformations, this information can be collapsed to combine contributors in order to create

models which target a specific level of detail. Following this, preprocessing or normalization is typical before training or testing a model with the synthesized data.

2.3. Machine Learning

In the last decade, ML approaches to RIID have become increasingly popular and have shown promise in a number of problem spaces (section 2.5), potentially serving as yet another supplement to analysis workflows. The research has been accelerated by the availability of modern software tools which can generate the large amounts of synthetic gamma spectra that ML requires [4, 5].

2.3.1. Neural Networks

Machine learning, which falls under the broader field of artificial intelligence (AI), can be thought of in its most fundamental form as a set of statistical methods used to define nonlinear systems. This section serves to provide a basic understanding of the principles of machine learning models as they pertain to this study. For more detailed information we direct readers to the works of Hastie et al. [6], Murphy [7] (for a more statistical perspective), Mohri et al. [8], Bishop et al. [9], and Shalev-Shwartz et al. [10].

Any machine learning method begins with a set of observations x which are drawn i.i.d. from some domain or distribution $P(\mathcal{X})$ which is defined by a set of physical processes. These observations, which consist of sets of observed features, can be represented through various modalities such as images/videos, sets of scalar/categorical descriptors, spectra (which is the case for this study), etc. The aim of ML models is to learn a (generally nonlinear) function, f, which maps observations, $\mathcal{X}$, to an associated label space, $\mathcal{Y}$,

$$f := \mathcal{X} \rightarrow \mathcal{Y}.$$

For the problem targeted in this study, which can be thought of a type of regression problem, raw gamma spectra form the feature space ($\mathcal{X}$) and the label space ($\mathcal{Y}$) consists of a vector of the corresponding isotope proportions.

This relation between observations and labels can be learned in several fashions, including in a supervised or unsupervised manner. For supervised learning, the ML model learns using a dataset, $\mathcal{D}$, consisting of n pairs of observations and labels,

$$\mathcal{D} := \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}.$$

ML models learn through an optimization problem called empirical risk minimization (ERM). This involves minimizing the error (or risk) between the model's predictions $f(x)$ and the corresponding true labels y, which is defined via some error metric known as a loss function. For a loss function, L, this optimization problem can be expressed as a learning objective,

$$\underset{f}{\operatorname{argmin}} \sum_{i=1}^{n} L(f(x_i), y_i).$$

For this study a neural network model was used. Neural network models, which consist of multiple layers connected in various fashions, are nonlinear models which can be applied to many problem types including (but not limited to) classification and regression. Neural networks, in their simplest form, are structured in three stages.

1. An input layer receives as inputs the vector of features corresponding with an observation.
2. Subsequent hidden layers (called this because they are not directly observed) represent internal features, where each layer's features are the linear combination of features in the preceding layer.
3. A final output layer returns the prediction of the neural network which are found as a linear combination of features from the penultimate layer.

Every layer in a neural network (excluding the input layer) consists of a set of features (or nodes) modeled using the features of the previous layer. Each node, being fully connected, is derived as a linear combination of all the nodes in the preceding layer, where each input from the preceding layer is multiplied by a learned weight. Generally the output of every hidden layer is passed through an activation function to normalize the features, before they are carried forward. A neural network learns by adjusting its weights such that the loss from the aforementioned learning objective is reduced. This is done during a step called backpropagation where the weights are updated via an optimization algorithm such as gradient descent. Mathematically speaking, neural networks are simply large, nonlinear functions which are well-defined in terms of their learned weights. However, the way in which the weights are learned as well as the complex relationships between various features is mostly unknown. This mystery motivates the need of additional explain-ability for ML models.

2.3.2. Loss Functions

ML models rely on loss functions when training to estimate the quality of their predictions and in order to calculate a loss used to perform back propagation on their weights. In this section we introduce a few loss functions which are relevant to this study.

2.3.2.1. Cross-Entropy

The cross-entropy loss, also known as the logistic loss or log loss, is one of the most commonly used loss functions in ML, and is used to measure the difference between two probability distributions. The cross-entropy loss is based on the softmax activation function which maps the non-normalized outputs of a neural network (called logits) to valid probabilities (such that they are non-negative and sum to one). For a vector of logits $\mathbf{x} \in \mathbb{R}^n$, softmax activation is defined as,

$$\text{softmax}(\mathbf{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}.$$

The cross-entropy loss function aims to minimize the difference between the predicted distribution and the true distribution. Suppose that $\hat{\mathbf{y}} \in \mathbb{R}^n$ is a model's softmax-activated predictions and $\mathbf{y} \in \mathbb{R}^n$ is the true distribution. Then the cross-entropy loss is defined as,

$$L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{j=1}^{n} y_j \log(\hat{y}_j).$$

The cross-entropy loss uses a logarithmic penalty which leads to a large loss for errors close to one and a small loss for errors close to zero.

Although the cross-entropy loss is typically used for multi-class classification tasks, it can also be used for LPE as it minimizes the variance between probability distributions.

2.3.2.2. Sparsemax

The recently proposed sparsemax loss [11] for LPE is very similar to the cross-entropy loss, and is optimal when the true label proportions are known to be sparse (as in most of the proportions are exactly zero). While the cross-entropy loss relies on softmax activation, which maps logits to a dense distribution, sparsemax loss relies on sparsemax activation which maps logits to a generally sparse distribution.

For a vector of logits $\mathbf{x} \in \mathbb{R}^n$, sparsemax activation is defined as,

$$\text{sparsemax}(\mathbf{x}) = \underset{\mathbf{p} \in \Delta^{n-1}}{\text{argmin}} \parallel \mathbf{p} - \mathbf{x} \parallel^2,$$

which is the Euclidean projection of the logits onto the probability simplex. In other words, sparsemax activation will map a vector of logits to the closest valid probability distribution, which generally being on the boundary of the simplex, will be sparse.

The paper provides a closed-form solution of the sparsemax activation which is defined as,

$$\text{sparsemax}(\mathbf{x})_i = [x_i - \tau(\mathbf{x})]_+,$$

where $[\cdot]_+$ is the soft-thresholding function ($[t]_+ = \max\{0, t\}$) and $\tau(\mathbf{x})$ is defined as,

$$\tau(\mathbf{x}) = \frac{\left(\sum_{j \in S(\mathbf{x})} x_j\right) - 1}{|S(\mathbf{x})|},$$

where $S(\mathbf{x}) = \{j \in [n] \mid \text{sparsemax}_j(\mathbf{x}) > 0\}$ is the support of $\text{sparsemax}(\mathbf{x})$.

The sparsemax loss for the LPE task is based off the sparsemax activation and is defined as follows,

$$L_{\text{sparsemax}}(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbf{y}^T \hat{\mathbf{y}} + \frac{1}{2} \sum_{j \in S(\hat{\mathbf{y}})} (\hat{y}_j^2 - \tau^2(\hat{\mathbf{y}})) + \frac{1}{2} \parallel \mathbf{y} \parallel^2 .$$

2.3.2.3. Jensen-Shannon Distance

The Jensen-Shannon distance is a measure of distance between two probability distributions and can be used as a metric of similarity between them. The Jensen-Shannon distance is based off the Kullback-Leibler divergence which is also a measure of similarity between probability distributions. However, unlike the Kullback-Leibler divergence, the Jensen-Shannon distance enjoys the guarantees of being symmetric and bounded (between 0 and 1). For two distributions, P and Q, the Jensen-Shannon distance is defined as,

$$\text{JSD}(P \parallel Q) = \frac{\text{KL}(P \parallel M) + \text{KL}(Q \parallel M)}{2},$$

where KL is the Kullback-Leibler divergence and $M = \frac{1}{2}(P + Q)$ is the average mixture of P and Q.

2.4. Dirichlet Distribution

The Dirichlet distribution, a multivariate generalization of the beta distribution, can be used to randomly sample a discrete probability distribution and is useful for randomly sampling mixture proportions (section 3.4.1). The Dirichlet distribution can be thought of a distribution of distributions and is defined on the probability simplex for some $\mathbf{x} \in \Delta^{n-1}$ as,

$$Dir(\mathbf{x}|\alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{n} x_i^{\alpha_i - 1},$$

where $B(\cdot)$ is the multivariate beta function and α is a vector which controls the variance and shape of the distribution. In particular, if $\alpha_1 = \alpha_2 = \ldots = \alpha_n$ then all the mixture proportions will be sampled evenly, while $\alpha_1 = 2, \alpha_2 = 1, \alpha_3 = 1, \ldots, \alpha_n = 1$ would generate a skewed distribution with the first term having a greater proportion on average. Also, the magnitude of α is inversely related to the variance of the distribution. For example $\alpha_1 = \alpha_2 = \ldots = \alpha_n \to \infty$ will generate a distribution converging on all equal proportions. However, $\alpha_1 = \alpha_2 = \ldots = \alpha_n \to 0$ will generate a distribution with a single, random positive proportion equal to 1.

2.5. Related Works

In 2019, Kamuda et al. trained a neural network model to perform isotope identification and quantification [12]. Their model was trained with synthetic gamma spectra from a 2"x2" NaI detector which spanned 29 sources with a fixed background composition. Their primary contribution was to demonstrate that machine learning approaches could be successfully used to estimate isotope proportions in small mixtures. However, their technique differs from this report as they only consider two, smaller mixture combinations (with contributions from 2 and 3 sources), train their model with a conventional loss function (cross-entropy), test their model solely on synthetic spectra, and do not consider OOD detection.

In 2019, Kim et al. trained a neural network to perform multi-isotope identification of various isotope combinations [13]. Their model was trained with synthetic and measured gamma spectra from a PVT detector which including isotope combinations spanning 4 sources with various background measurements. Their primary contribution was to demonstrate how an ANN could be used to predict isotope mixtures with a classification approach, and to show how such a model performed on real gamma spectra. However, their technique differs from that of this report as their model performs classification and can only identify combinations of present isotopes, not their proportions. They also do not consider OOD detection.

In 2022, Ghawaly et al. characterized their Autoencoder Radiation Anomaly Detection (ARAD) model to detect anomalies from background as a binary classifier [14]. The model was trained and evaluated on a set of measured gamma spectra collected over three years using a NaI detector near the HFIR/REDC complex at Oak Ridge National Laboratory (ORNL). The primary contribution was the introduction of the first use of a deep neural autoencoder for anomaly detection from gamma spectra. While both ARAD and the OOD detector of this report are similar in that they aim to detect anomalies from a pre-defined problem space, they differ in several ways:

- ARAD focused on anomalies from background signatures while our model focuses on anomalies from a set of foreground signatures in the presence of an LPE task.
- ARAD used an autoencoder to first learn a latent feature representation of input spectra which is poorly reconstructed in the presence of anomalies. The model of this paper does not learn a latent feature space in the same way, but rather the two tasks of reconstruction and LPE are treated as complementary and learned together.
- ARAD set a JSD anomaly threshold based on a target false-positive rate (1 per 2 hours) related to a ROC curve obtained from a test dataset, whereas our OOD detector adapts its JSD threshold as a function of a sample's SNR while maintaining a false-positive rate no higher than 0.01, which was similarly learned from a test dataset.

In 2023, Khatiwada et al. explored the performance of various machine learning models (decision tree, random forest, gradient boosted trees, K-nearest neighbors regression, Gaussian process regression, MLP, and convolutional neural network) for isotope proportion estimation [15]. Their models were trained using synthetically generated mixtures of uranium and plutonium based on an HPGe detector with varied backgrounds and shielding configurations. The primary contribution of their work was to compare the performance of various machine learning algorithms for isotope proportion estimation and evaluate the effects of source/shielding geometries on model performance. Their work differs from this study as they apply a number of off-the-shelf ML approaches to isotope proportion estimation, and do not consider OOD detection. They also focus on a different problem space, predicting the relative isotope proportions of uranium and plutonium in various shielding configurations.

In 2023, Van Omen produced, as part of their master's thesis, a semi-supervised neural network that incorporated concepts from dictionary learning to perform LPE and OOD detection [1]. The LPE model was trained on synthetically-generated fission spectra based on an HPGe response to estimate the proportional contribution of up to 30 distinct radioisotopes. The primary contribution was to demonstrate the efficacy of a semi-supervised approach to radioisotope proportion estimation that incorporates a single semi-supervised loss function in the training process [1]. However, the

technique of that thesis differs from that of this report in that a lower resolution detector (NaI) is utilized, fewer sources are considered, and a greater number of measured spectra are used in testing.

In 2023, Stomps et al. apply various semi-supervised ML models to perform binary classification of SNM identification [16]. Their models were trained with data collected from the MINOS test-bed project at ORNL which monitored nuclear material transfers throughout two test sites using NaI detectors. The primary contribution of their work was to demonstrate how semi-supervised ML methods could successfully utilize both labeled and unlabeled data to improve on supervised ML techniques in cases where labeled data is expensive to obtain. Their approach and problem differ from that of this report in several key ways:

- Their paper uses a different type of semi-supervised learning. In particular, they mean semi-supervised learning in the sense that they leverage both labeled and unlabeled data when training their model. However, the model in this paper only utilized labeled data but combined a supervised and unsupervised loss term to form the learning objective.
- They performed binary classification of SNM while we are solving an LPE problem.
- They did not consider OOD detection.

3. METHOD

This study investigates the capabilities of synthetically trained ML models on real gamma spectra. The detector hardware, detection environment, and sources are introduced, for both the synthetic and real gamma spectra generation. Numerous subsections distinguish between *real* vs. *synthetic* and IND vs. OOD aspects of the problem to make clear the models' priors as well as how they were tested. In short, models are tested on all of these aspects, but trained only on synthetic, IND data.

3.1. Detector

3.1.1. Real

Real gamma spectra measurements were taken with a 3" $\times$ 3" NaI scintillation detector.

3.1.2. Synthetic

The synthetic spectral signatures for the target sources, referred to as "seeds," were obtained via GADRAS Inject [4] using the specific detector response function (DRF) associated with the detector used to collect measurements with a 10 μs dead time.

3.2. Environment

3.2.1. Real

Gamma spectra were measured with the NaI detector in a lab setting under controlled conditions. The room itself was about 4m $\times$ 8m $\times$ 2.5m with concrete flooring, aluminum walls, and a dropped ceiling. The detector was placed on its side on the floor so that its major axis aligned with the direction of the target source, as shown in figure 3-1.

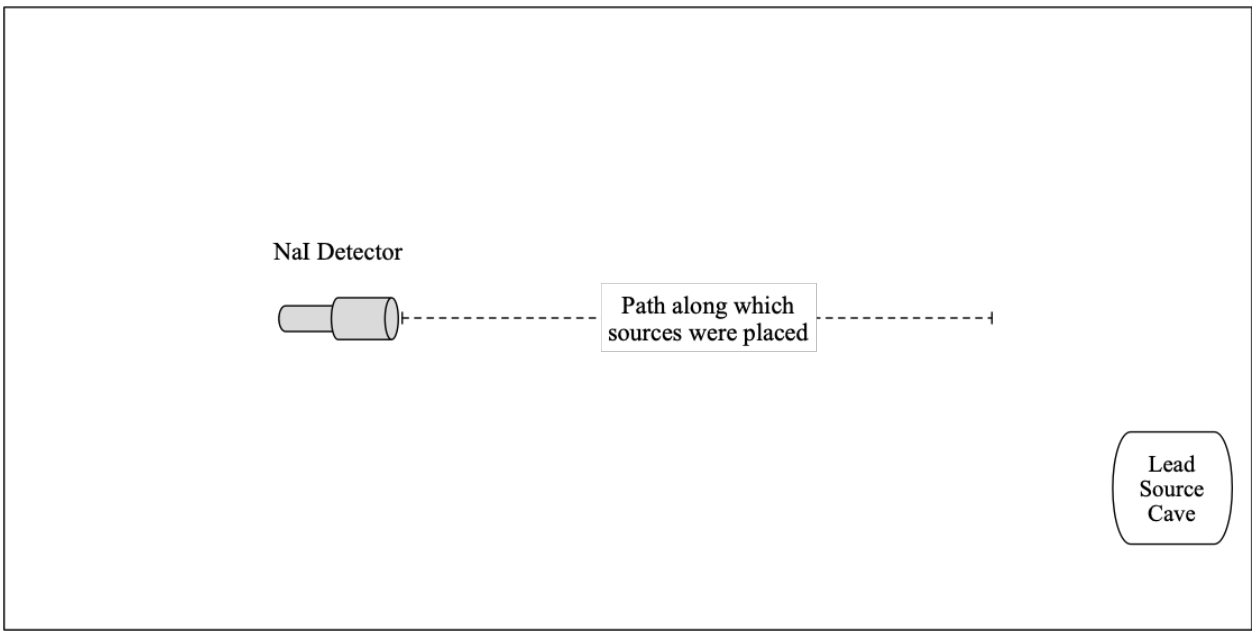


Figure 3-1. Lab floor layout

3.2.2. Synthetic

The synthetic seeds produced with GADRAS were generated for a fixed scattering environment at a distance of 50 cm distance and a height of 1 cm.

3.3. Sources

3.3.1. Real

Measured sources were held at 100 cm above the floor, except in case of Co60 which was placed on the floor to get closer to the detector due to its low activity. The distance between the detector and source was varied to understand count rate per cm, but ultimately only the highest SNR samples would be used for pseudo-measured mixtures. It is also worth noting that in some cases the closest distance did not yield the highest SNR. This is most likely because of the arm holding the source from underneath which acted as shielding between the source and the detector. Figures 3-2 and 3-3 provide a visual reference for the lab setup used to measure the radioactive sources.

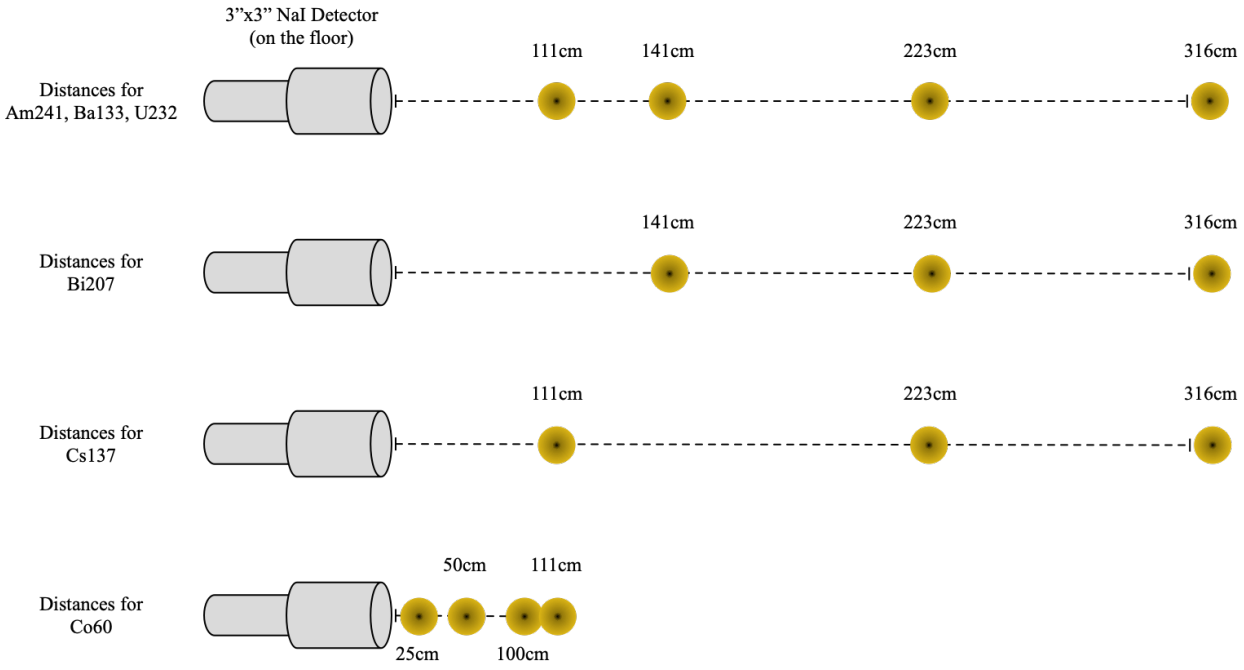


Figure 3-2. Lab setup for collecting gamma measurements of sources (viewed from above)

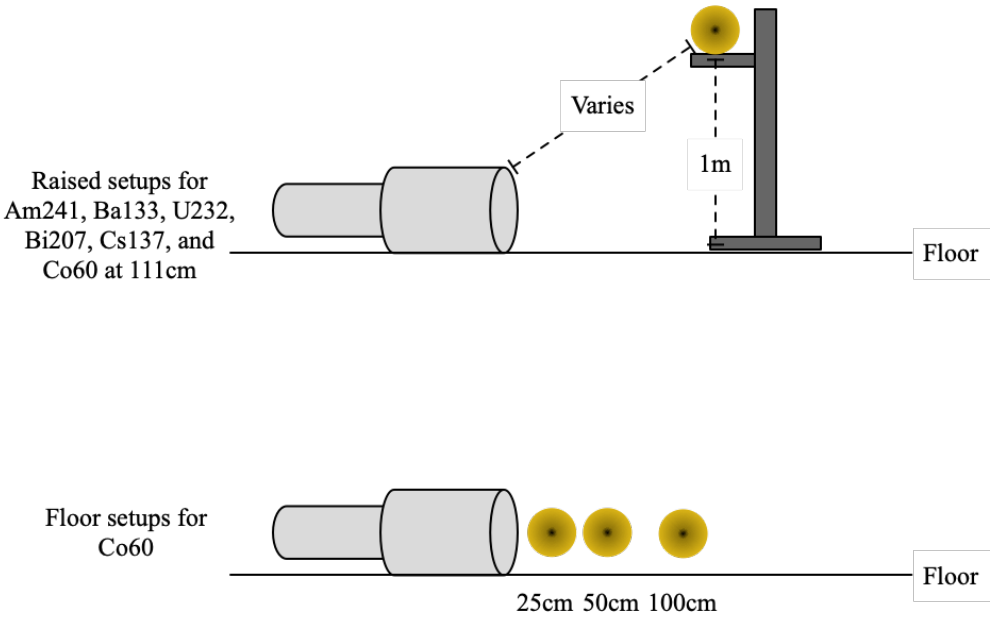


Figure 3-3. Lab setup for collecting gamma measurements of sources (viewed from side)

Sixteen measurements were taken of 4 IND sources (Am241, Ba133, Co60, U232) in the lab. The detector-source distance, live time, and SNR of each IND measurement is shown in Table 3-1. Six measurements were also taken of 2 OOD sources (Bi207, Cs137) which are shown in Table 3-2. A long-collect background measurement was also taken with a live time of 611.7 sec yielding

a background count rate of 421 counts per second. The long-collect background measurement was used to perform background subtraction on the measured spectra and produce approximate foreground measurements. All the measured spectra (IND, OOD, and BG) are plotted in Appendix A.1.

Table 3-1. Real IND source measurements

Source	Distance (cm)	Live Time (sec)	SNR
Am241	316.2	60.4	4
Am241	223.6	62.7	12
Am241	141.4	60.4	23
Am241	111.8	59.7	10
Ba133	316.2	61.0	14
Ba133	223.6	60.6	27
Ba133	141.4	62.3	58
Ba133	111.8	60.1	42
Co60	111.8	60.3	10
Co60*	100.0	60.72	16
Co60*	50.0	60.2	65
Co60*	25.0	59.8	194
U232	316.2	60.4	19
U232	223.6	59.9	30
U232	141.4	60.0	71
U232	111.8	60.2	86

* source placed on the ground in line with detector

Table 3-2. Real OOD source measurements

Source	Distance (cm)	Live Time (sec)	SNR
Bi207	316.2	59.6	24
Bi207	223.6	60.1	47
Bi207	141.4	60.6	97
Cs137	316.2	62.0	13
Cs137	223.6	61.5	25
Cs137	111.8	60.0	48

3.3.2. Synthetic

Synthetic seeds obtained via GADRAS Injects, which are used to create training data, are generated for 6 IND sources: Am241, Ba133, Co60, U232, Cf252, and Eu152. Two of these isotopes (Cf252 and Eu152) were not available to measure in the lab, but were still included in synthetic training

data. This was done in order to provide an extra challenge for the models by expanding the scope of the training data.

Synthetic seeds were also generated for 4 background sources: K40 (*PotassiumInSoil*), Ra226 (*UraniumInSoil*), Th232 (*ThoriumInSoil*), and cosmic radiation. These background seeds were randomly combined to generate representative background spectra when creating training datasets, and they were also used as synthetic OOD sources when testing the OOD detection model.

3.3.3. Comparison

The success of our approach for estimating isotope proportions from real measurements relies on the synthetic training data accurately representing the real spectra encountered. This can be challenging as numerous real-world effects, which are not accounted for when training, can change the shape of measured gamma spectra. Some of these include the specific geometry of the scattering environment in which spectra were measured, flaws or particular characteristics of the detector that were not accounted for in the DRF, changes in temperature (which can affect the detector's calibration), and differences in the background environment, just to name a few. In order to mitigate some of these discrepancies, the gain and offset values of the energy calibration were manually tuned on the measured spectra so that the spectral peaks visually aligned with the corresponding peaks in the synthetic seeds. Background subtraction was also applied to the gross measurements using the long-collect background measurement in order to obtain approximate foreground spectra. And lastly, the first four energy channels of the resulting foreground signatures were set to zero as these channels were found to only contain X-ray noise. To illustrate the degree to which the energy calibration of measurements match their corresponding synthetic seeds, overlays of the spectra are shown in Figures 3-4, 3-5, 3-6, and 3-7.

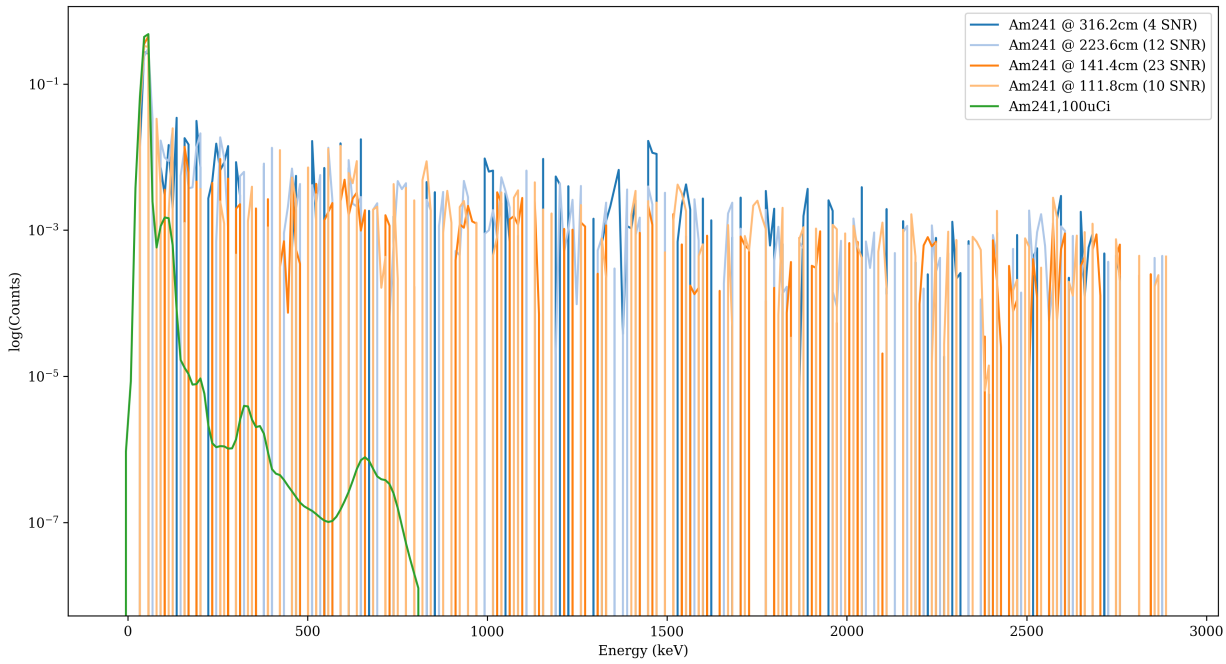


Figure 3-4. Comparison between synthetic seed and foreground measurements for Am241

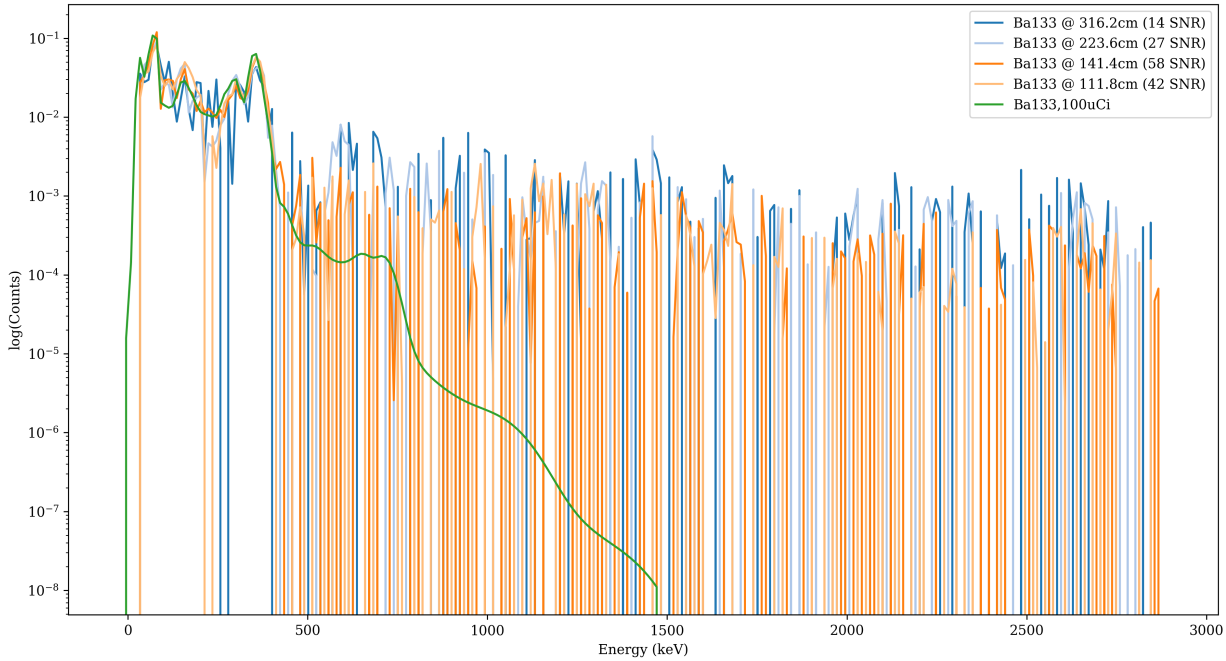


Figure 3-5. Comparison between synthetic seed and foreground measurements for Ba133

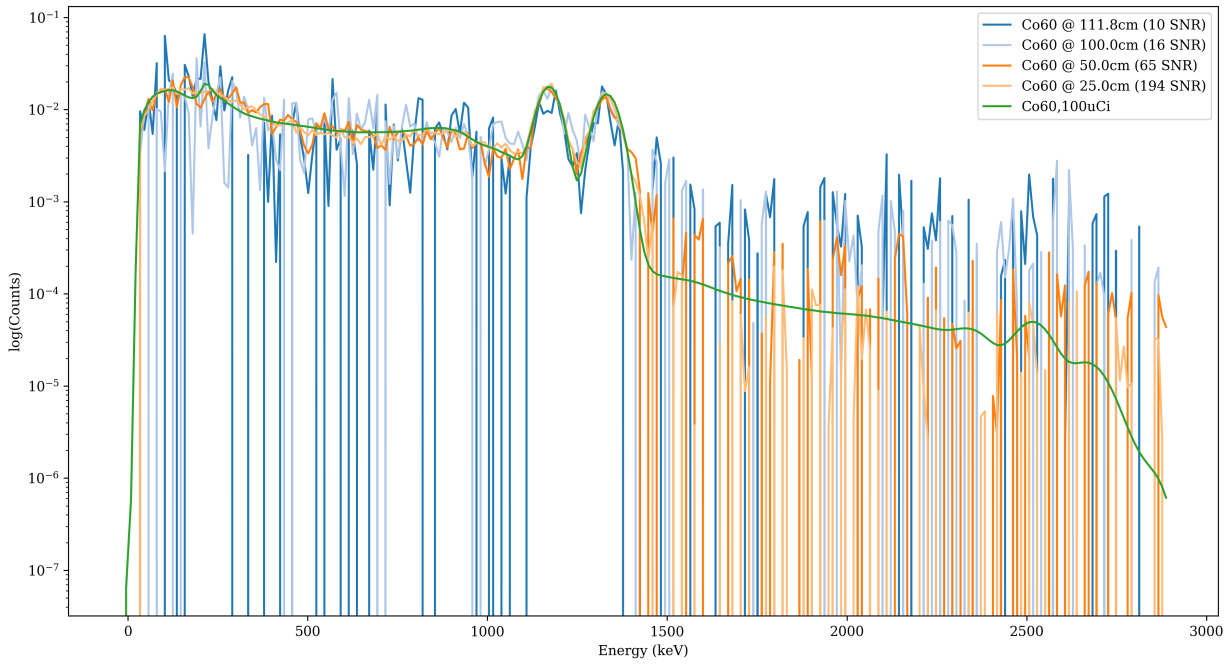


Figure 3-6. Comparison between synthetic seed and foreground measurements for Co60

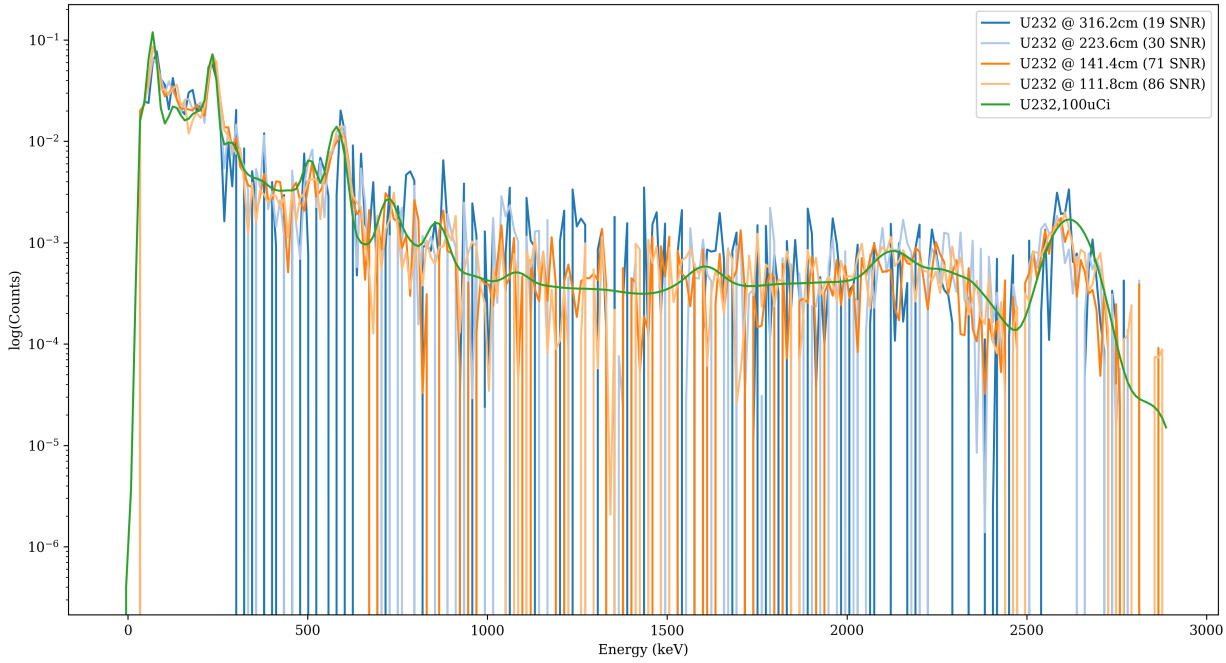


Figure 3-7. Comparison between synthetic seed and foreground measurements for U232

Understanding the relative similarity between various source signatures is important, not only to validate the quality of the synthetic seeds, but also to understand the models' behavior (Section 6). While Figure 3-7 qualitatively shows the agreement between some synthetic and measured source signatures, these similarities can also be described quantitatively using the Jensen-Shannon distance which is used to describe the similarity between two probability distributions. The spectral distances between the various normalized seeds (synthetic vs. real, IND vs. OOD) are represented using spectral distance matrices shown in Figure 3-8 and 3-9. Each element in a spectral distance matrix represents the Jensen-Shannon distance between two particular sources, where smaller values indicate closeness.

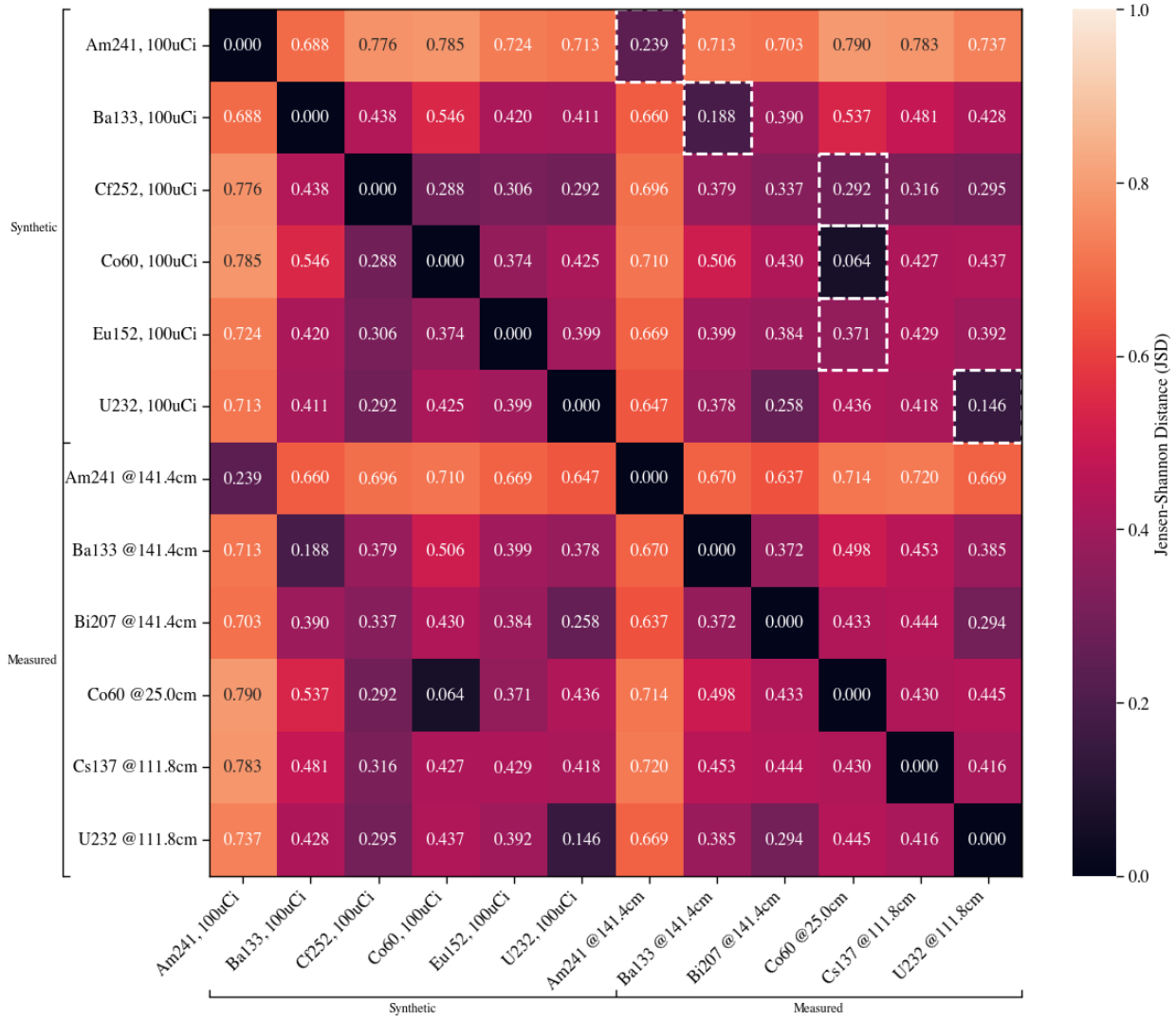


Figure 3-8. Spectral distance matrix comparing IND synthetic and measured seeds (a dashed square denotes the measured column source most similar to the synthetic row source).

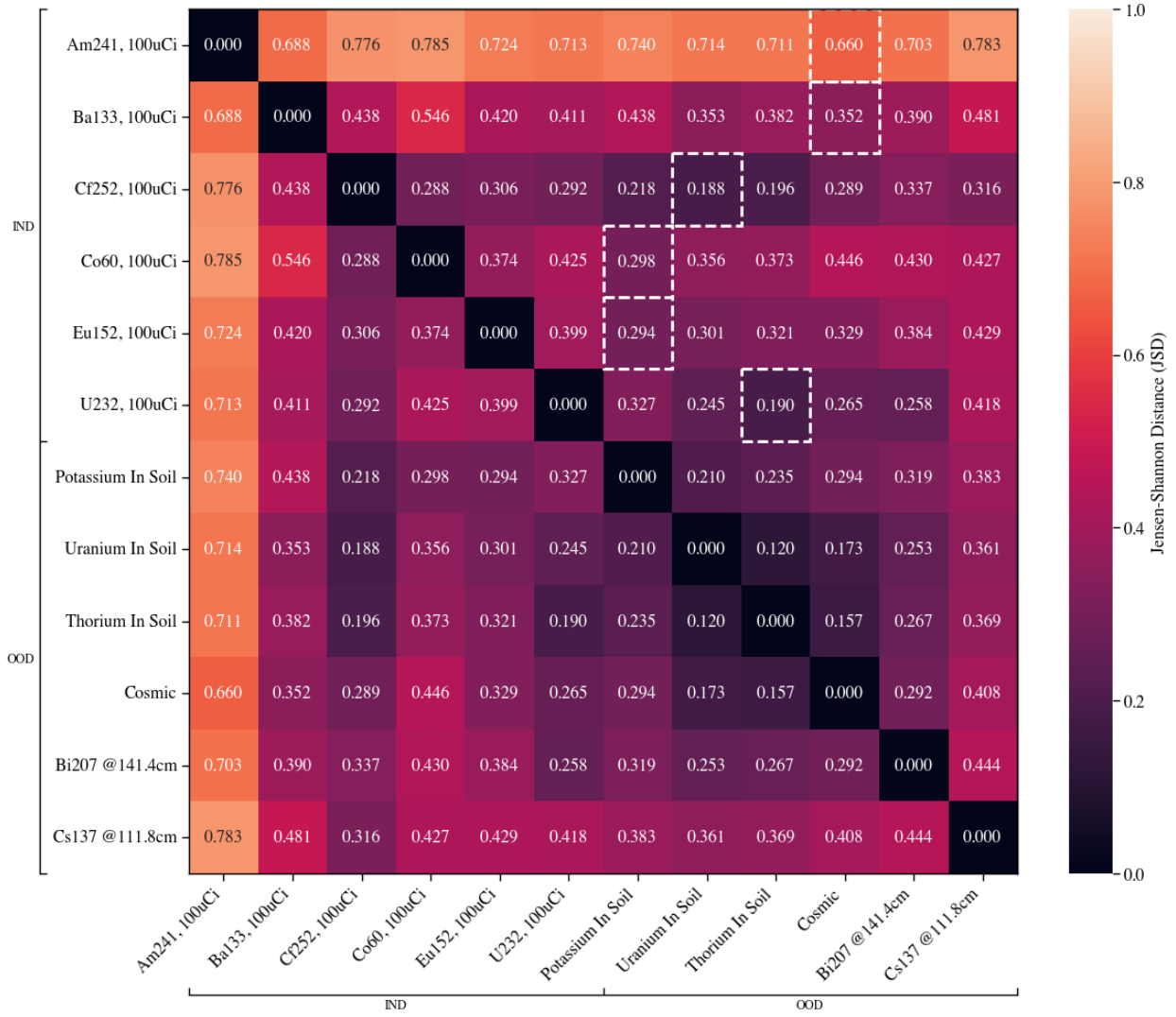


Figure 3-9. Spectral distance matrix comparing IND synthetic seeds and OOD sources (a dashed square denotes the OOD column source most similar to the IND row source).

3.4. Semi-Supervised LPE Model

3.4.1. Generating Training Data

The model used to perform isotope proportion estimation was trained and fine-tuned solely with synthetic gamma spectra. With the seeds (foreground and background) obtained from GADRAS, a synthetic training dataset was generating using PyRIID (2.2.2). The process to generate realistic synthetic spectra can be broken down into four distinct steps:

1. Down-sample Raw Spectra
   Initially, the synthetic foreground and background seeds contain 1024 distinct energy channels. The seed spectra are down-sampled to 256 channels by uniformly summing every

4 channels. This serves to (1) reduce noise contained in the spectra and (2) reduce the computation cost of synthesizing subsequent data and training models.

2. Generate Random Mixtures
   The next step is to randomly sample mixture proportions and mix the synthetic seeds together to create noiseless mixtures. Both the foreground and background seeds were mixed together separately in order to create foreground and background mixtures. This was done using the *SeedMixer* in PyRIID which randomly samples proportions using the Dirichlet distribution (section 2.4). Each background mixture contains a random mixture of all four background constituents. For the foreground mixtures, a mixture size of 3 was used so that each mixture contains 1-3 foreground sources simultaneously.

3. Static Synthesis of Mixture Spectra
   Taking the pure mixtures from the previous step, realistic gamma spectra can be simulated using the *StaticSynthesizer* in PyRIID. The *StaticSynthesizer* receives both the foreground and background mixtures as inputs and returns realistic gamma spectra by (1) generating gross spectra by randomly sampling the SNR, (2) applying Poisson noise to the counts in each energy channel, and (3) performing imperfect background subtraction using a long-collect Poisson-sampled background mixture.

4. Additional Preprocessing Steps
   Before the spectra are ready to be used for training, several preprocessing steps are applied. As a result of the imperfect background subtraction, it is common for some low-SNR foreground spectra to have negative counts in some channels. These negative values are clipped at zero, allowing the spectrum to be L1-normalized. This is done by dividing through each spectrum by its total counts such that each spectrum sums to one. This ensures that all the training data is on the same scale.

Following these steps a synthetic training dataset was generated containing 900k spectra (900k = 15 BG mixtures * 300 FG mixtures * 200 samples per seed). All the parameters used to generate the training dataset are shown in Table 3-3.

Table 3-3. Parameters for generating synthetic training data

Parameter	Value
FG sources (synthetic)	Am241, Ba133, Co60, Cf252, Eu152, U232
target bins	256
BG mixture size	4
BG mixture samples	15
BG Dirichlet alpha	1
FG mixture size	3
FG mixture samples	300
FG Dirichlet alpha	1
BG counting rate (cps)	300
samples per seed	200
SNR range	(5, 100)
SNR sampling style	log10
live time range (sec)	(60, 60)
normalization	L1

Once the training set was generated, the distribution of mixture proportions for each source can be visually inspected to ensure the dataset is balanced. Figure 3-10 shows these distributions for each source in the training dataset which verifies our training data is balanced. Note only the non-negative proportions are shown as zero proportions dominate the dataset.

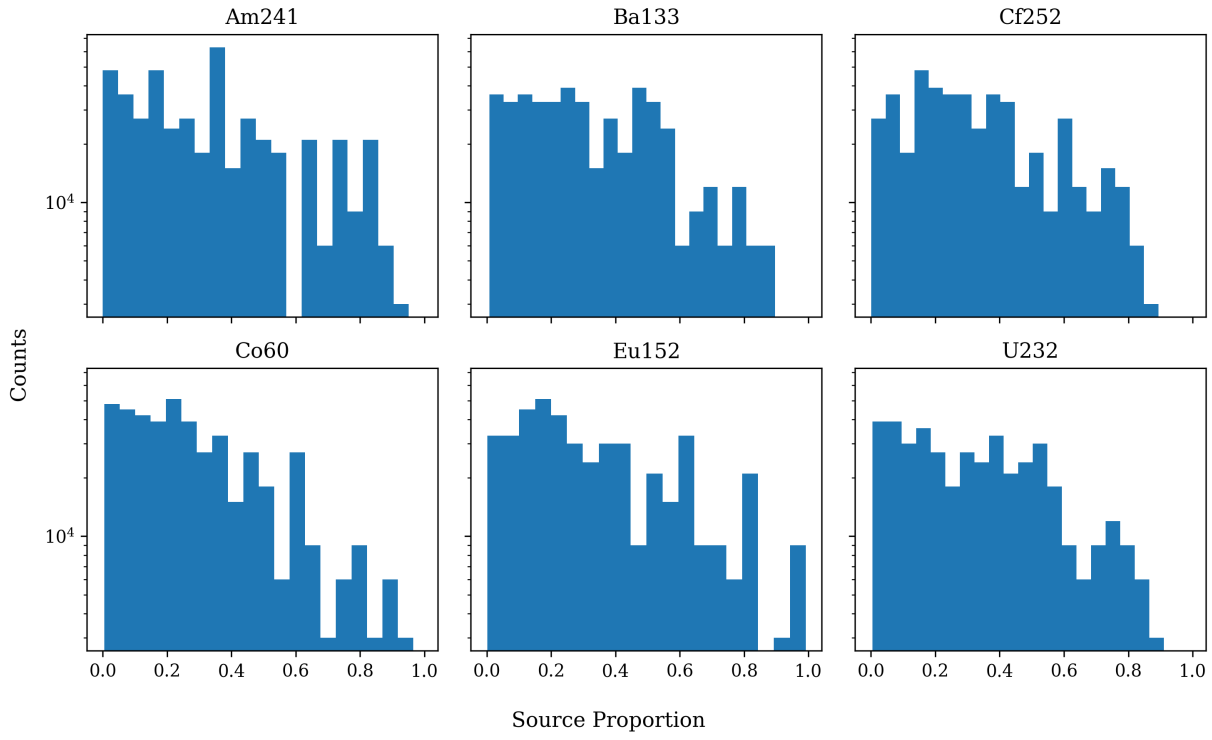


Figure 3-10. Distribution of mixture proportions for each isotope present in training spectra

Figure 3-11 shows the distribution of mixture sizes (ranging from 1 to 3) present in the training dataset. Here the mixture size of a spectrum is defined as the number of contributing sources with proportions greater than 10%. The plots show that the training dataset is dominated by mixtures of size 2 and 3, although a handful of single-isotope spectra are present. To illustrate mixtures fully, a randomly drawn spectrum from the training dataset is shown in Figure 3-12.

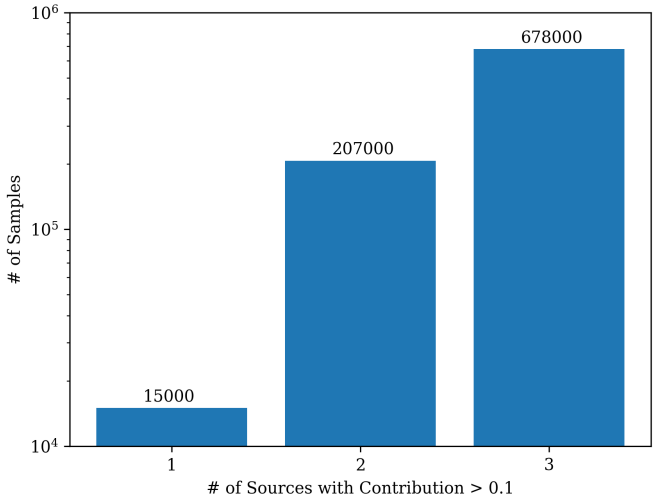


Figure 3-11. Distribution of mixture sizes in training data

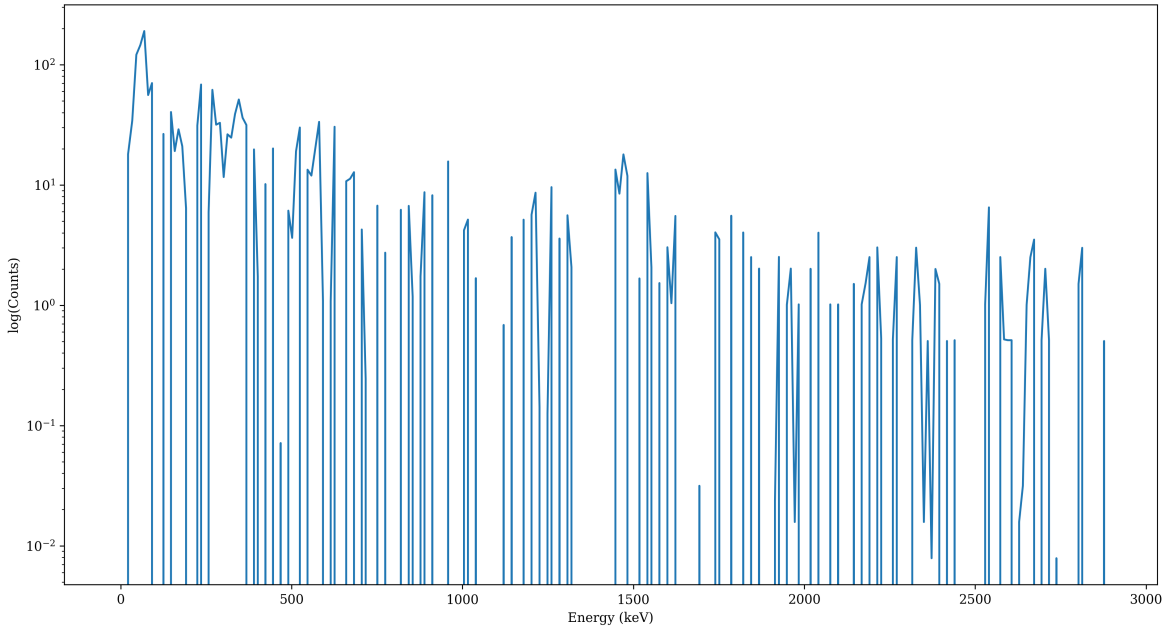


Figure 3-12. A random sample from the training dataset with an SNR of 9.5 (1273 foreground source counts) and approximately composed of 17% Am241, 40% Ba133, and 43% U232

Randomly sampling of mixture proportions was conducted consistently for each source to ensure the dataset did not favor any particular source. However, due to the nature of the Dirichlet sampling technique, the dataset was not balanced in terms of mixture size as targeting a mixture size of 3 does not guarantee that every sample will contain a significant contribution from all sources. Using a definition of mixture size that counts the number of sources contributing more than 10% of the counts in a spectral sample, the numbers of samples for each mixture size were observed as follows: 15k 1-mixture samples, 207k 2-mixture samples, and 678k 3-mixture samples.

3.4.2. Generating Testing Data

3.4.2.1. Synthetic Test Data

First a dataset of synthetic test spectra was created, which was used to evaluate performance for the hyperparameter search (Section 3.4.3.3), tune β (Section 3.4.3.4), train the OOD detector (Section 3.5.1), and compare against the models' performance on measured data (Section 4.2). In order to simulate the scenario tested in the lab, only the four seeds corresponding to the four IND lab sources (Section 3.3.2) were used to generate the synthetic test set. The test spectra were generated in the same way was as the training spectra, using the *SeedMixer* and *StaticSynthesizer* in PyRIID. To ensure that the test set is IND, the background mixtures were sampled from the background mixtures generated for the training dataset. All the parameters used to generate the test dataset are shown in Table 3-4. The synthetic test dataset contained 100k spectra (100k = 5 BG mixtures * 200 FG mixtures * 100 samples per seed).

Table 3-4. Parameters for generating synthetic test data

Parameter	Value
FG sources (synthetic)	Am241, Ba133, Co60, U232
target bins	256
BG mixture size	4
BG mixture samples	5
BG Dirichlet alpha	1
FG mixture size	3
FG mixture samples	200
FG Dirichlet alpha	1
BG counting rate (cps)	300
samples per seed	100
SNR range	(5, 100)
SNR sampling style	log10
live time range (sec)	(60, 60)
normalization	L1

Table 3-5. Parameters for generating measured test data

Parameter	Value
FG sources (real)	Am241, Ba133, Co60, U232
target bins	256
BG mixture size	4
BG mixture samples	5
BG Dirichlet alpha	1
FG mixture size	3
FG mixture samples	200
FG Dirichlet alpha	1
BG counting rate (cps)	300
samples per seed	100
SNR range	(5, 100)
SNR sampling style	log10
live time range (sec)	(60, 60)
normalization	L1

3.4.2.2. Pseudo-Measured Test Data

Collecting a sufficiently large measured test set in the lab and calculating all the true isotope proportions would have been prohibitively expensive. In order to evaluate the models' performance on measured mixture spectra, a large pseudo-measured test set was generated using the highest-SNR measurement of each source as seeds. In particular, the foreground spectra for each lab source was estimated by performing background subtraction on the highest-SNR lab samples. With these measured seeds in hand a test dataset was generated in the same way as the synthetic test dataset using PyRIID. Note we refer to these spectra as "pseudo-measured" because they were synthetically generated by randomly sampling off of real measurements treated as seeds. From this point forward we will refer to this data as measured test data to simplify terminology. The measured test data also contains the same number of samples and was generated using the same parameters as the synthetic test dataset, the only difference being that measured seeds were used in place of synthetic ones (Table 3-5).

3.4.3. Creating the Model

3.4.3.1. Semi-Supervised Loss Function

The main idea behind our approach is to train an ML model for isotopic proportion estimation with a custom semi-supervised loss function that is designed to produce accurate proportion estimates while simultaneously encouraging reliable OOD detection by constraining overconfidence inherent to neural networks. Our proposed loss function contains two terms, a supervised term and an unsupervised term, and is based off [1] except with the unsupervised loss term now being Jensen-Shannon distance. This approach leverages the fact that L1-normalized gamma spectra can be

viewed as discrete probability distributions, as all the values are non-negative and sum to one (utilizing negative clipping). Moreover, every target spectrum can be viewed as a mixture of the pure signatures associated with each class, which we call seeds.

This approach relies on the assumption that we have access to the seed associated with each class a priori. This assumption is reasonable as isotope identification algorithms typically target a specific set of isotopes, and the spectral shape for each isotope can be easily simulated with radiation transport and detector response software, such as GADRAS. The assumption can be formally stated using a dictionary matrix $\mathbf{D} \in \mathbb{R}^{c \times d}$, where each column is the seed associated with a particular class. Here $c = 256$ represents the number of energy channels in our spectra and $d = 6$ represents the number of target classes. Using this representation, any measured spectrum, $\mathbf{x} \in \mathbb{R}^c$, should be approximately represented as a mixture of the dictionary columns,

$$\mathbf{x} = \mathbf{D} * \mathbf{y} + \mathbf{n},$$

where $\mathbf{y} \in \mathbb{R}^d$ is a vector of the true source proportions and $\mathbf{n} \in \mathbb{R}^c$ is noise.

For a data pair $(\mathbf{x}, \mathbf{y})$ and LPE model $f : \mathbb{R}^c \to \mathbb{R}^d$, the semi-supervised loss function takes the following form,

$$\begin{aligned} L(\mathbf{x}, \mathbf{y}; \mathbf{D}, \beta) = (1 - \beta) * L_{sup}(\mathbf{y}, f(\mathbf{x})) + \\ \beta * L_{unsup}(\mathbf{D} * f(\mathbf{x}), \mathbf{x}), \end{aligned}$$

where L_{sup} represents a supervised loss function which compares the true isotope proportions $\mathbf{y}$ to the predicted isotope proportions $f(\mathbf{x})$, and L_{unsup} represents an unsupervised loss function which compares the reconstructed input spectrum $\mathbf{D} * f(\mathbf{x})$ to the actual input spectrum $\mathbf{x}$. Thus, the unsupervised loss function is referred to as the reconstruction error and $\beta \in [0, 1]$ represents a scalar hyperparameter which controls the trade-off between the two loss terms.

For the supervised loss a bounded version of the Sparsemax loss [11] was selected, as the true isotope proportions were known to be sparse. Alternatively, the cross-entropy loss could also be used if sparsity cannot be assumed. The Sparsemax loss in an unbounded, non-negative loss function, so to ensure it remained on the same scale as the unsupervised term the hyperbolic tangent function was used to bound its values between 0 and 1. In particular, the supervised loss function is given as follows,

$$L_{sup}(\mathbf{y}, f(\mathbf{x})) = \tanh(5 * \text{sparsemax}(\mathbf{y}, f(\mathbf{x}))),$$

where a scalar multiple of 5 was used to change the shape of the hyperbolic tangent, such that the supervised and unsupervised loss returned values approximately equal in magnitude.

For the unsupervised loss the Jensen-Shannon distance was chosen. The JSD is a bounded, symmetric version of the Kullback-Leibler divergence and is used to compare the similarity between two probability distributions, making it ideal for this use case. The unsupervised term of the loss function is given as follows,

$$L_{unsup}(\mathbf{D} * f(\mathbf{x}), \mathbf{x}) = \text{JSD}(\mathbf{D} * f(\mathbf{x}), \mathbf{x}).$$

Then putting both loss terms together we have the final semi-supervised learning objective used to train the LPE model,

$$L(\mathbf{x}, \mathbf{y}; \mathbf{D}, \beta) = (1 - \beta) * \tanh(5 * \text{sparsemax}(\mathbf{y}, f(\mathbf{x})))+ \\ \beta * \text{JSD}(\mathbf{D} * f(\mathbf{x}), \mathbf{x}).$$

3.4.3.2. Model Architecture and Implementation

The isotope proportion estimation is performed with an ML model. Various types of ML models were explored, including random forests, gradient-boosted decision trees, convolutional neural networks, and MLPs. An MLP (a shallow feedforward neural network, to be more specific) was ultimately chosen as it exhibited similar performance to the other models and can be easily embedded in various compute systems.

The neural network model was constructed in TensorFlow [17] and trained using the Adam optimizer [18]. The architecture consisted of a dense neural network with two hidden layers.

3.4.3.3. Model Hyperparameter Search

The size of the hidden layers as well a number of the training parameters were selected using an automated hyperparameter search. The hyperparameter search was conducted with the Optuna [19] framework which uses a Tree-Structured Parzen Estimator to efficiently navigate the search space and sample hyperparameters expected to provide the most improvement. The hyperparameter search was set up to minimize the MAE on the synthetic test set and was conducted with 100 trials. Table 3-6 shows the hyperparameter search space and the final hyperparameters used for the model. For this hyperparameter search, β was held constant at 0.5 as it was selected based not only on the MAE but also the reconstruction error (Section 3.4.3.4). Early stopping and learning rate annealing were also used to reduce training time.

Table 3-6. The search space and final selection of training hyperparameters

Parameter	Search Space	Final Value
hidden layer 1 nodes	(16, 256)	173
hidden layer 2 nodes	(16, 64)	63
batch size	(32, 512)	86
hidden layer activation	{mish, relu, softplus, tanh}	relu
initial learning rate	(0.005, 0.015)	0.012
epsilon (for optimizer)	(0.0, 0.05)	0.017
dropout	(0.0, 0.05)	0.04
kernel L1 regularization	(0.0, 0.001)	8.1e-4
kernel L2 regularization	(0.0, 0.001)	8.2e-5
activity L1 regularization	(0.0, 0.001)	8.4e-4
activity L2 regularization	(0.0, 0.001)	4.7e-4

3.4.3.4. Tuning β

The relative performance of the model for LPE and OOD detection can be optimized by properly tuning β, which controls the trade-off between the two loss terms in the learning objective. The supervised loss term minimizes the difference between the true and predicted isotope proportions, and thus primarily serves to minimize the MAE of the proportion estimations. The unsupervised loss term minimizes the difference between the reconstructed spectrum and the input spectrum, and thus it primarily serves to improve the reconstruction error of the proportion estimates. In the loss function, β is applied such that $\beta = 0$ only uses the supervised loss and $\beta = 1$ only uses the unsupervised loss.

In order to select an optimal value for β, 105 models were trained across a range of $\beta \in [0, 1]$ using the hyperparameters found in the previous section. Figures 3-13 and 3-14 show the performance of the models on the synthetic test set in terms of the MAE and reconstruction error as a function of β. From the figures we can visually see that by selecting $\beta = 0.85$ we can obtain low reconstruction errors without significantly raising the MAE of the predicted proportions.

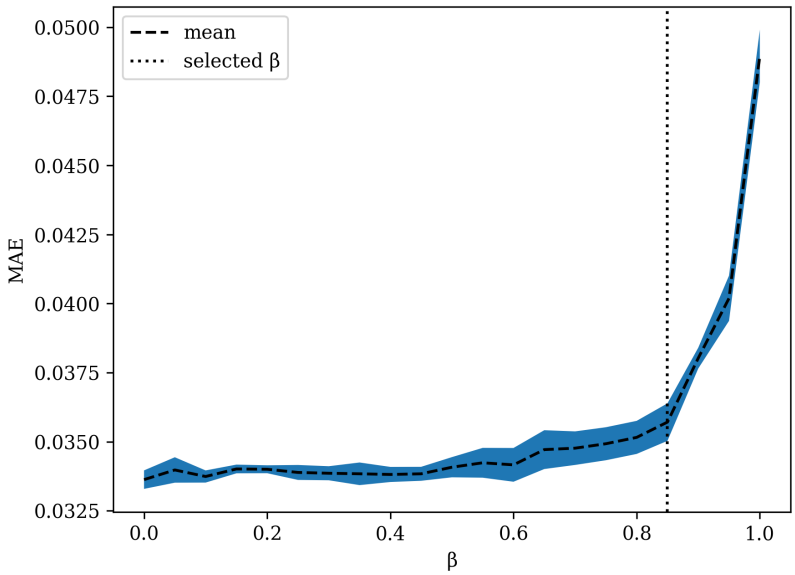


Figure 3-13. MAE as a function of β on synthetic test data, blue band represents standard deviation.

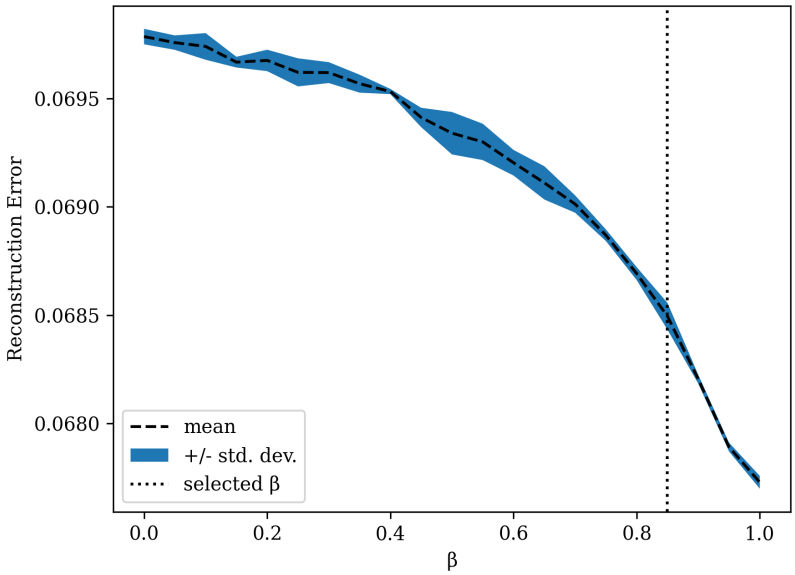


Figure 3-14. Reconstruction error as a function of β on synthetic test data.

3.4.3.5. Learning Curves

The final model was trained using the parameters found from the hyperparameter search in Section 3.4.3.3 and using $\beta = 0.85$ found in Section 3.4.3.4. The learning curves for the model are shown in Figure 3-15.

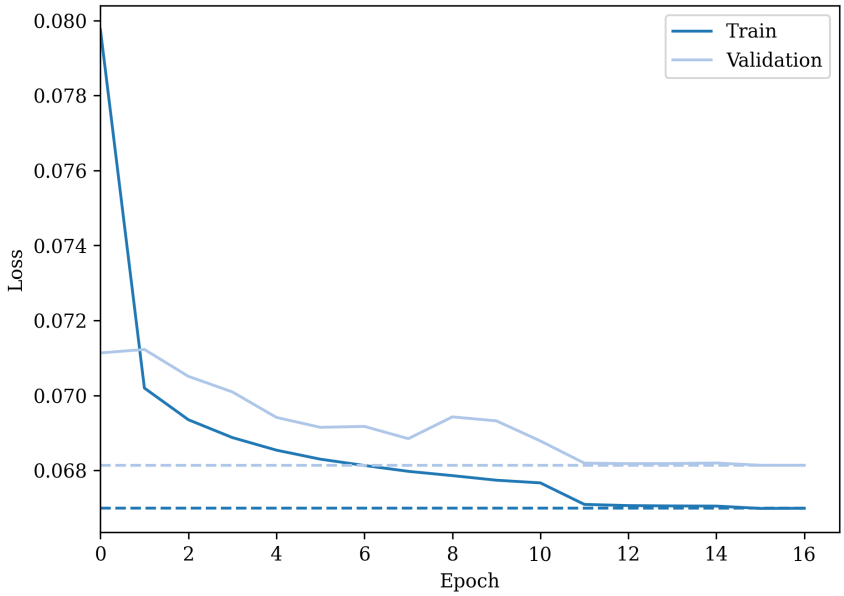


Figure 3-15. Training and validation curves from model training.

3.5. OOD Detection Model

The OOD detector is based on the fact the reconstruction error can be calculated for test spectra as it does not depend on the true label proportions. By including the reconstruction error in the loss function the LPE model is encouraged to predict proportions that yield reconstructions that are consistent with the dictionary (as in yield good reconstructions). When a spectrum is OOD due to the presence of an anomalous source, then the OOD detection model will not be able to generate a close reconstruction as the dictionary is missing the spectral signature associated with the anomaly. Thus, an OOD spectrum should yield a large reconstruction error. The more unique the anomaly is, the harder it will be for the OOD detection model to produce a close reconstruction, and the larger the reconstruction error should be. Increasing β when training the model should promote lower IND reconstruction errors, making OOD reconstruction errors easier to identify.

3.5.1. Training

The OOD detector is a decision function that classifies each input spectrum as either IND or OOD based on its reconstruction error. In practice, we have found that a simple threshold for the reconstruction error at some scalar value is insufficient for OOD detection, as the reconstruction error correlates with SNR. The reconstructed spectrum is created by mixing together the dictionary seeds with the predicted proportions. For low SNR, the LPE model does not have enough counts to accurately predict the isotope proportions, which results in a worse reconstruction. To account for this, we propose a OOD detector function which uses both the reconstruction error and the SNR to make a decision.

The exact nature of OOD reconstruction errors is unknown, a form of epistemic uncertainty introduced by such as factors as the similarity between an anomaly and IND seeds and the number of counts from the anomaly relative to other sources. As a practical matter, because one does not know the OOD sources to be encountered, OOD thresholds can only be characterized in terms of IND synthetic data. The way this is done is by identifying the region of IND reconstruction errors.

Specifically, the reconstruction error threshold is determined as a function of SNR for a desired FPR using an Interpolated Univariate Spline function (U-Spline). This was done in two steps based on the IND synthetic test set:

1. Bin SNR
   First all the spectra contained in the synthetic test set were binned by SNR into 15 equal-sized quantiles. For the spectra in each quantile, two values were calculated: (1) the average SNR and (2) the 0.99 quantile of the reconstruction errors (this corresponds to a 1% FPR).

2. Fit Spline Function
   Then using the average SNRs as the x-values and the 0.99 reconstruction error quantiles as the y-values, an Interpolated Univariate Spline function was fit to the data. A cubic spline function was used. The Interpolated Univariate Spline function guarantees that the spline will pass through each data points and be smooth.

Figure 3-16 plots the reconstruction errors as a function of SNR for the IND synthetic test spectra, along with the 0.99 reconstruction error quantile for each SNR bin, and the fitted threshold function.

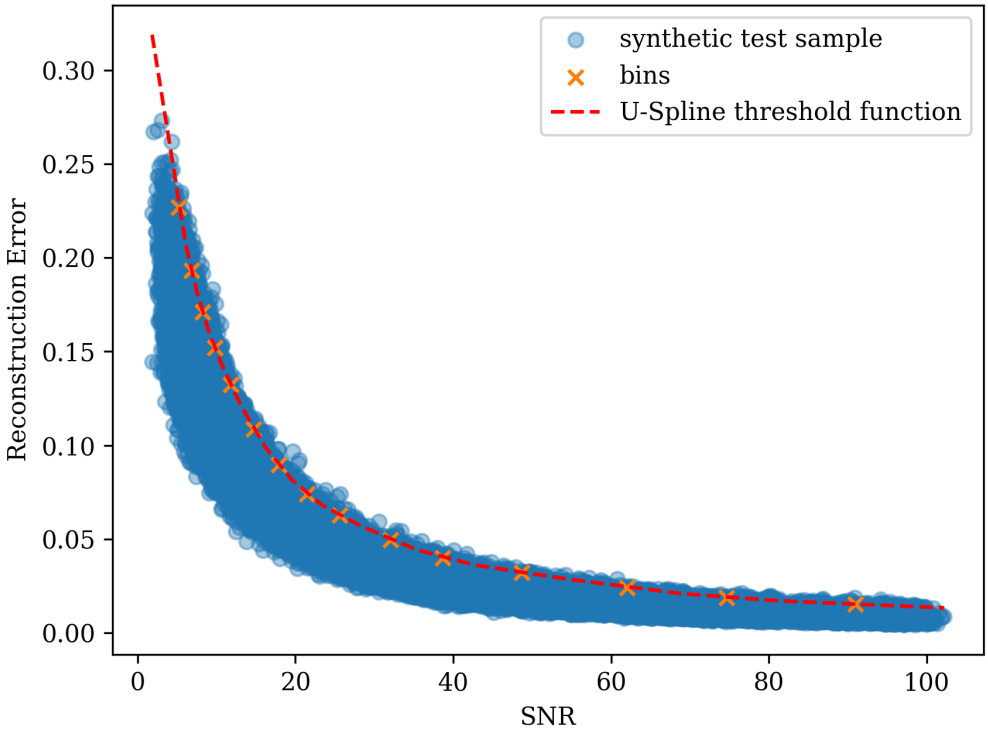


Figure 3-16. U-spline OOD threshold function for a 1% FPR along with reconstruction error vs. SNR for synthetic test samples.

3.5.2. Testing

To test the OOD detector on measured gamma spectra two OOD datasets were created using different types of OOD sources: OOD synthetic background sources (K40, Th232, Ra226, cosmic) and OOD measured lab sources (Cs137 and Bi207). Both OOD datasets were generated from the measured seeds, similar to the measured IND test data, along with one of the OOD seeds. The OOD dataset using the synthetic OOD seeds contains 4 million samples and the OOD dataset using the measured OOD seeds contains 2 million samples. The data generation parameters for these two datasets are shown in Tables 3-7 and 3-8.

Table 3-7. Parameters for generating OOD test data with synthetic BG OOD sources

Parameter	Value
FG sources (real)	Am241, Ba133, Co60, U232
OOD sources (synthetic)	K40, Th232, Ra226, cosmic
target bins	256
BG mixture size	4
BG mixture samples	5
BG Dirichlet alpha	1
FG mixture size	5
FG mixture samples	200
FG Dirichlet alpha	1
BG counting rate (cps)	300
samples per seed	100
SNR range	(5, 100)
SNR sampling style	log10
live time range (sec)	(60, 60)
normalization	L1

Table 3-8. Parameters for generating OOD test data with measured OOD sources

Parameter	Value
FG sources (real)	Am241, Ba133, Co60, U232
OOD sources (real)	Bi207, Cs137
target bins	256
BG mixture size	4
BG mixture samples	5
BG Dirichlet alpha	1
FG mixture size	5
FG mixture samples	200
FG Dirichlet alpha	1
BG counting rate (cps)	300
samples per seed	100
SNR range	(5, 100)
SNR sampling style	log10
live time range (sec)	(60, 60)
normalization	L1

3.6. Study Reproduction

The code, data, and best, pre-trained models produced in this study can be found online [20].

This page intentionally left blank.

4. RESULTS

4.1. Performance on Single-Isotope Measurements

The behavior of the LPE model can be validated by first predicting isotope proportions on the single-isotope lab measurements. The predictions of the LPE model on all 16 IND measurements and 6 OOD measurements are shown in figure 4-1. Although the LPE model is primarily trained to predict on mixture spectrum containing 2 or 3 isotopes there are a few single-isotope samples in the training dataset which is shown in figure 3-11. This result shows that the LPE model can successfully generalize to this test case, as it correctly identified the dominant source in each measurement.

The estimates from figure 4-1 also show that the LPE model seems to detect high proportions of Cf252 in the Ba133 measurements. This confusion is likely due to the fact that Ba133 and Cf252 have a similar spectral shape, which is supported by the small JSD between them shown in figure 3-8.

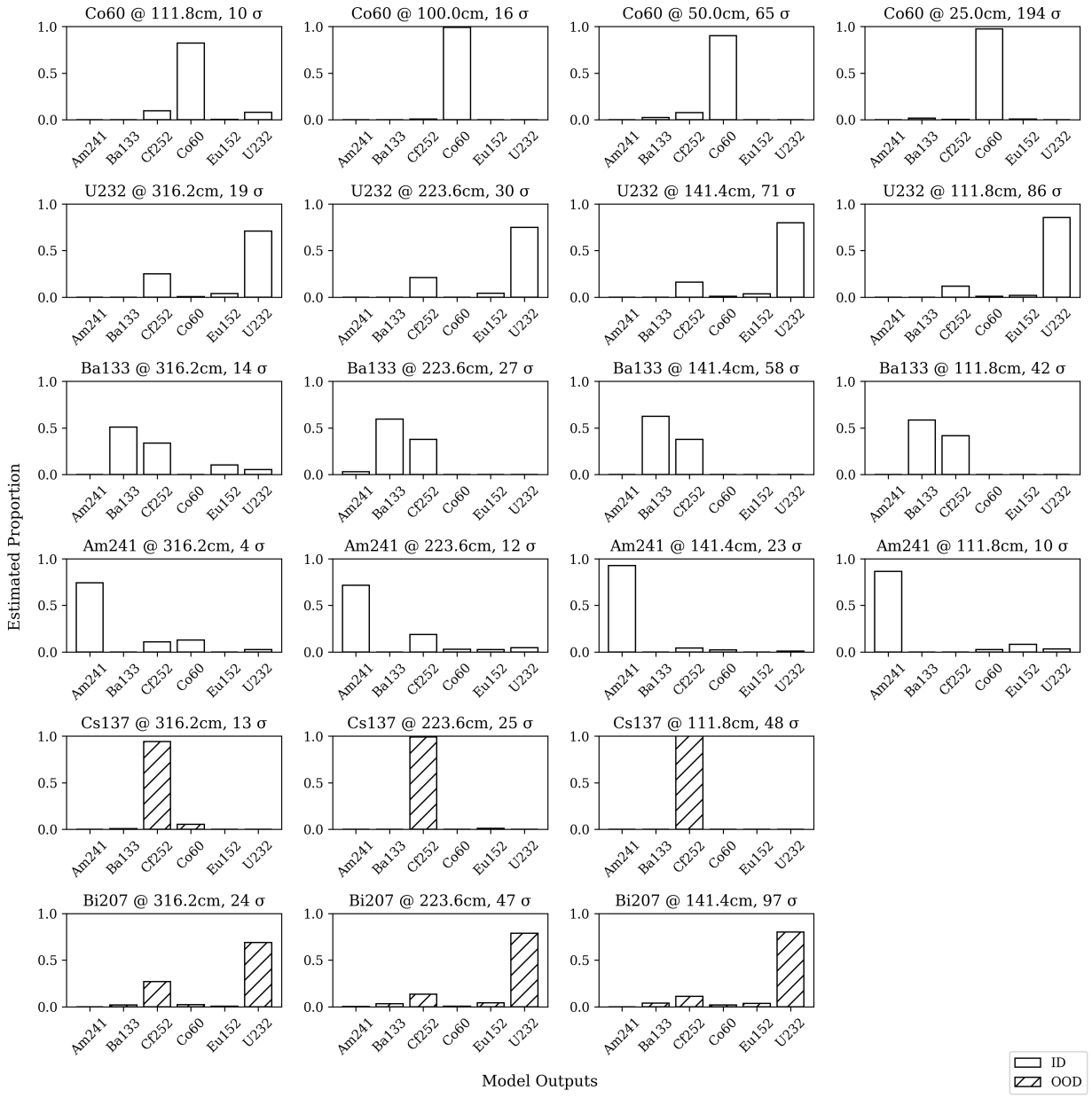


Figure 4-1. Isotope proportion estimates of single-isotope lab measurements

4.2. Performance on Multi-Isotope Spectra

4.2.1. Performance on Multi-Isotope Measurements

The LPE model was used to predict isotope proportions on both the synthetic and measured test datasets, on which it achieved a MAE of 0.035 and 0.07, respectively. Figures 4-2 and 4-3 compare the true and predicted isotope proportion estimates for a subset of both the synthetic and measured test spectra.

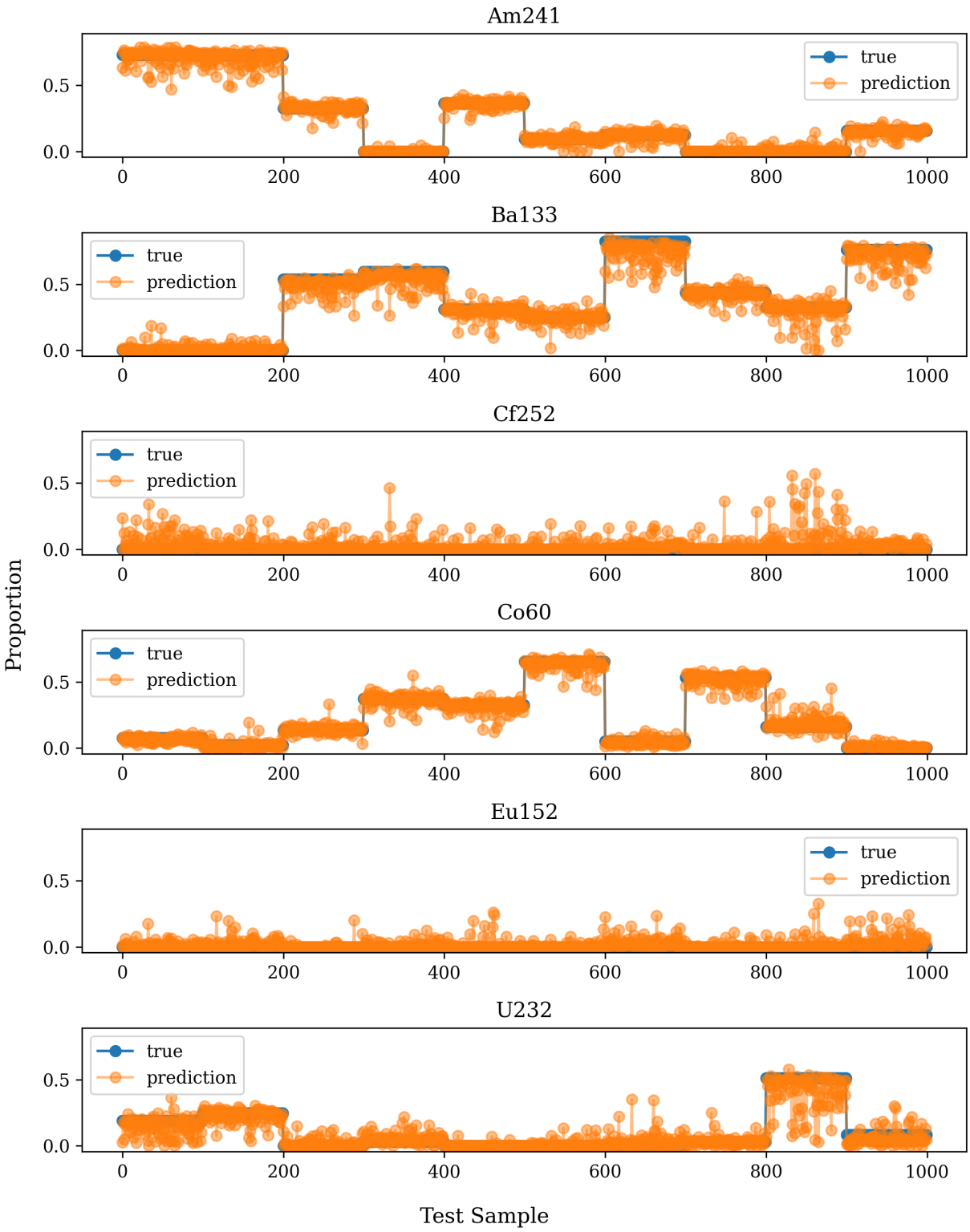


Figure 4-2. Scatter plot showing the true and predicted isotope proportions for 1000 synthetic test samples

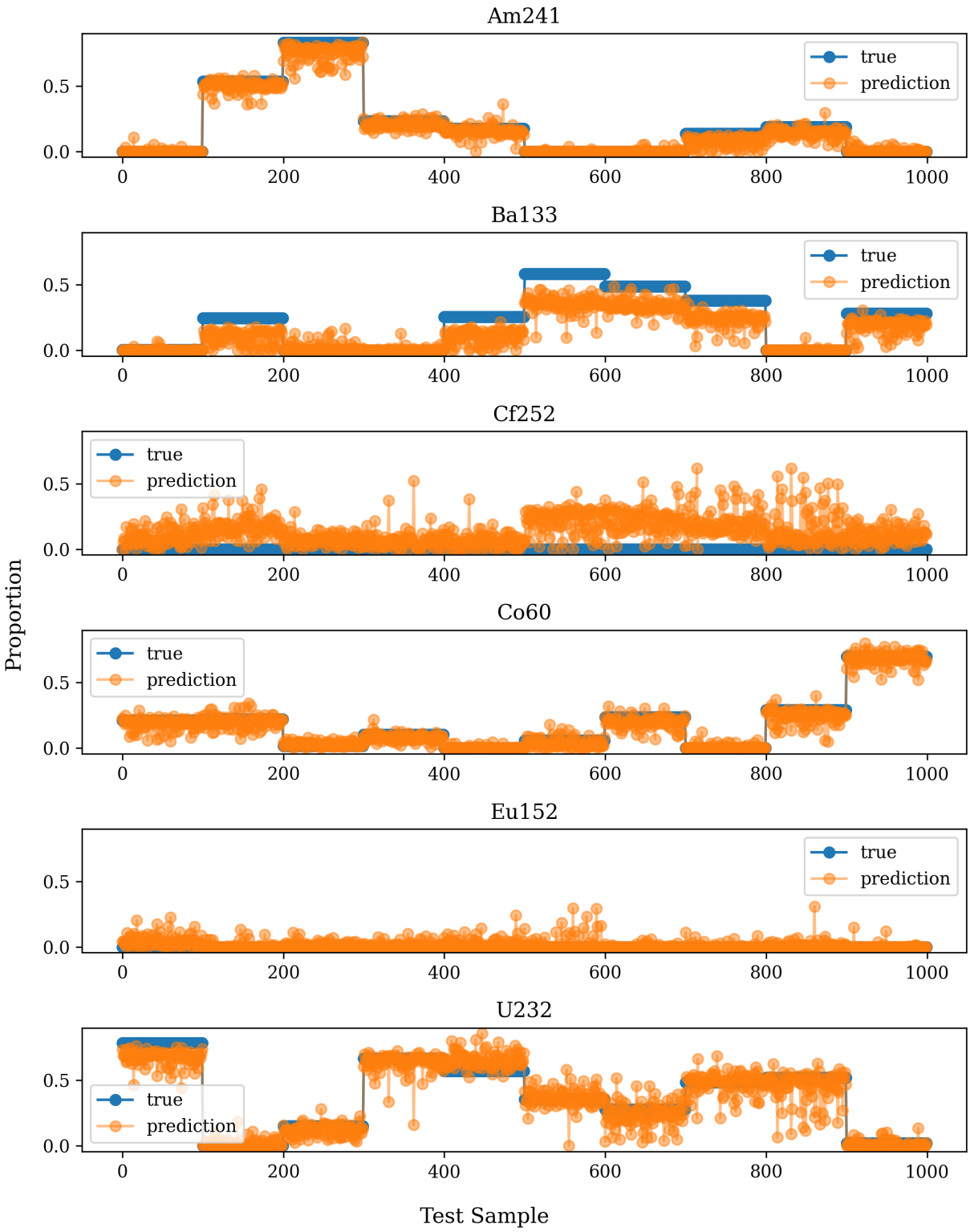


Figure 4-3. Scatter plot showing the true and predicted isotope proportions for 1000 measured test samples

4.2.2. MAE vs. SNR

The prediction errors were found to depend highly on the SNR of the test samples. Figures 4-4 and 4-5 show the MAE of the LPE model decreases as a function of SNR for both the synthetic and measured test spectra. Beyond about 50 SNR, the LPE model does not seem to gain any additional spectral information that would further lower the MAE. These plots also indicate that the LPE model performs better on synthetic spectra than real spectra, which is to be expected given the observed differences illustrated in Section 3.3.3.

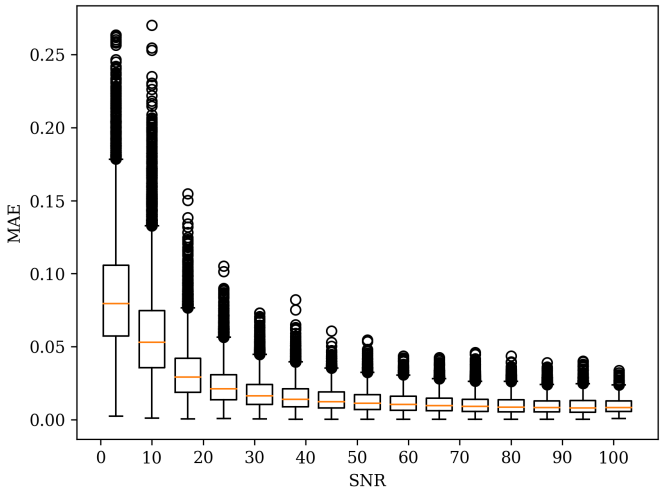


Figure 4-4. MAE vs. SNR for synthetic test spectra. Boxes represent inner quartile range (IQR), whiskers extend to the farthest sample within 1.5x the IQR, and outliers are shown as black circles.

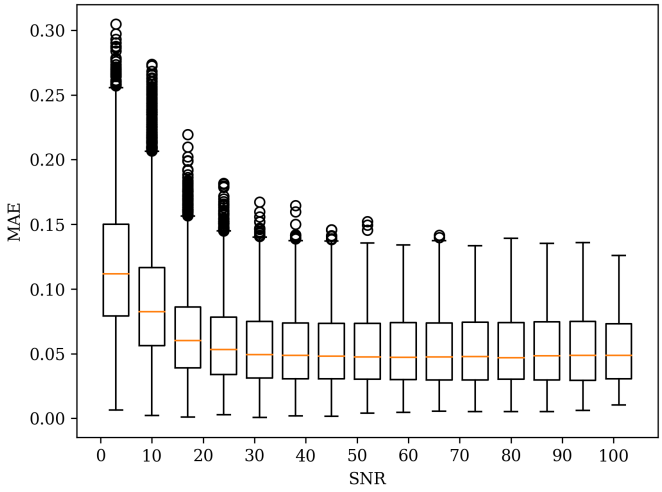


Figure 4-5. MAE vs. SNR for measured test spectra

4.2.3. Reconstruction Error and OOD Detection

Figure 4-6 shows the distribution of reconstruction errors for both the synthetic and measured test spectra. This plot shows that the reconstruction error distribution peaks are slightly offset, but still on the same scale.

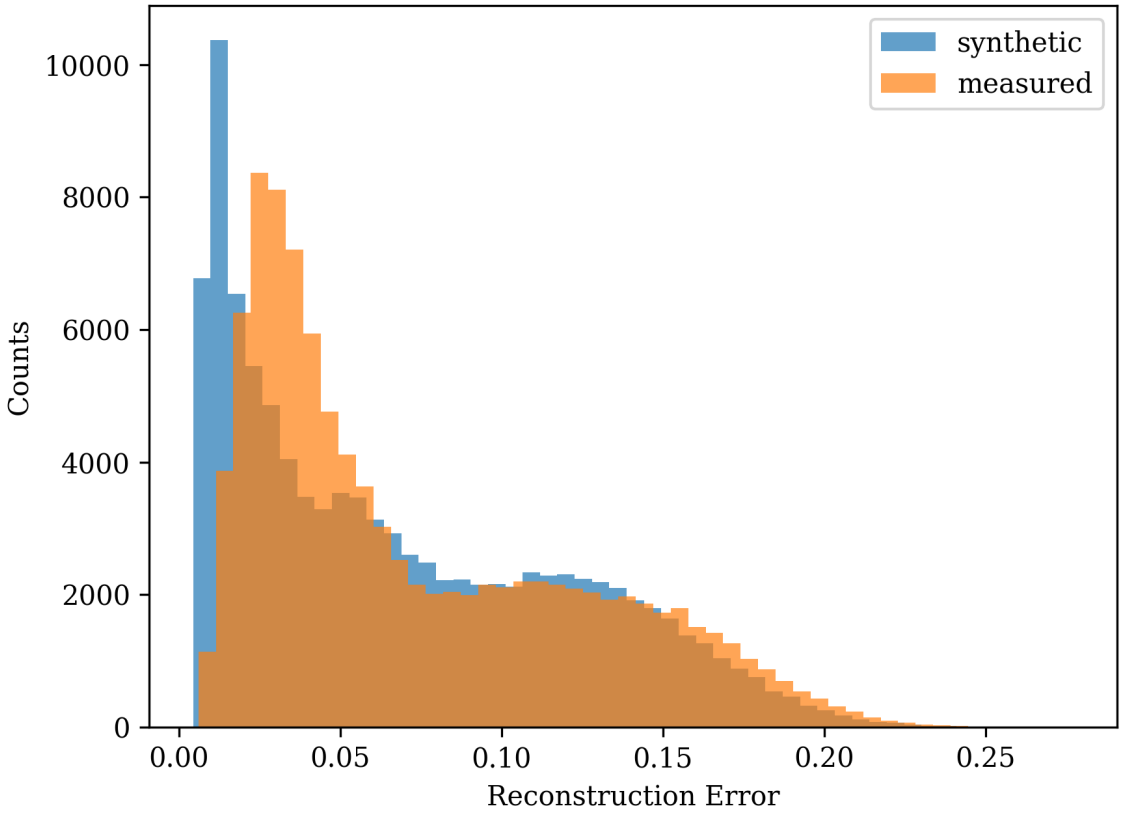


Figure 4-6. Distribution of reconstruction errors for synthetic and measured test spectra

To test the performance of the OOD detectors on measured test data the models were used with both OOD test datasets. Figure 4-7 and 4-8 show the reconstruction error for the OOD spectra as a function of the OOD contribution. From the figure it is clear that the reconstruction error grows larger with a higher OOD proportion, which was the expected behavior. However, this trend is most noticeable for the measured OOD sources (Bi207 and Cs137), and much smaller for the background OOD sources (K40, Ra226, Th232, and Cosmic). This is likely because the background signatures are relatively similar to nearly all the IND seeds which is supported by the spectral distance matrices in Figure 3-9.

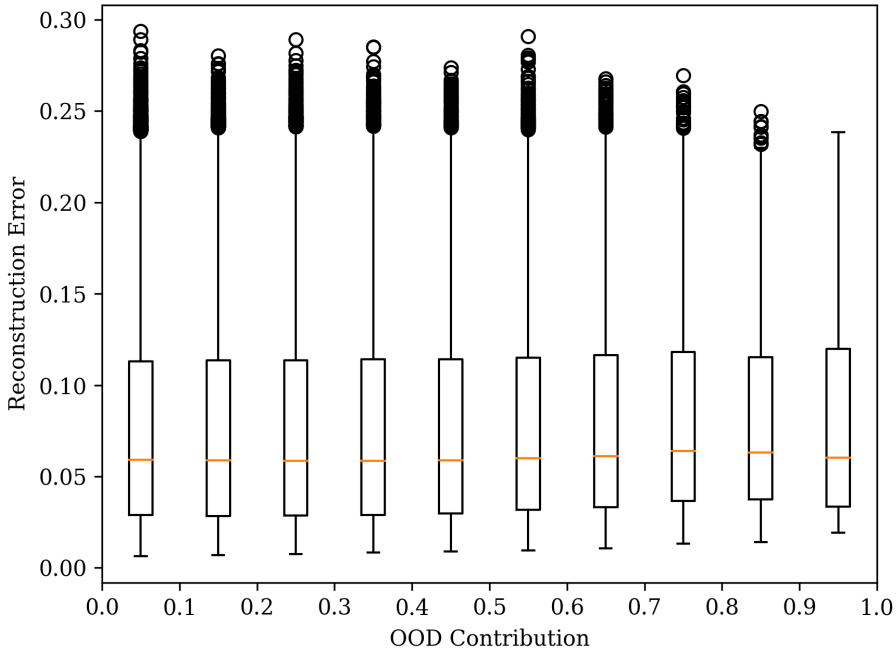


Figure 4-7. Reconstruction error vs. OOD contributions from synthetic, OOD background sources

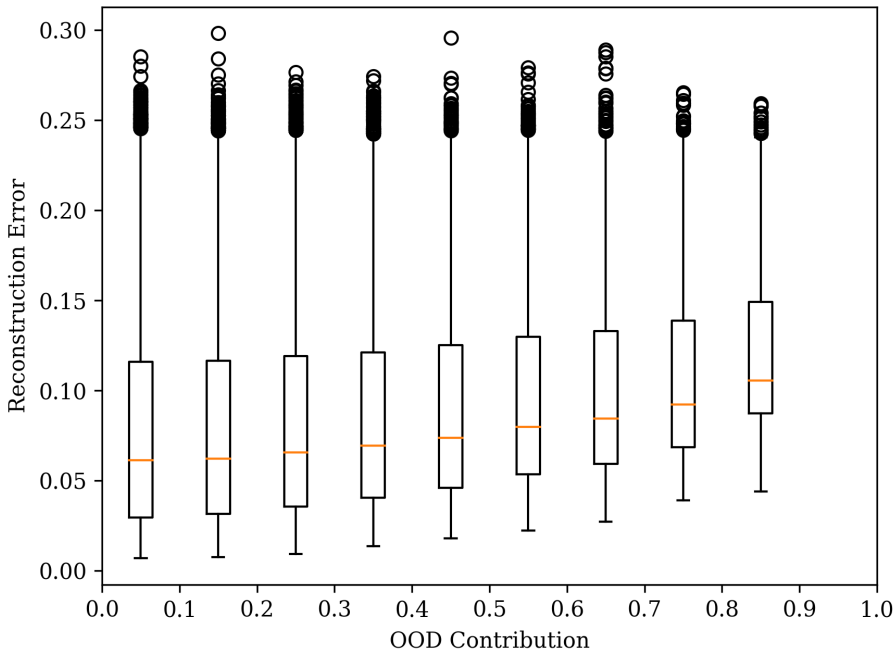


Figure 4-8. Reconstruction error vs. OOD contribution from measured OOD sources

Figure 4-9 and 4-10 demonstrate the performance of the OOD detector on both OOD datasets in terms of the FNR as a function of SNR and OOD proportion. From Figure 4-9 we can see that at 50 SNR the OOD detector can detect nearly all OOD spectra with an OOD proportion greater than 50%. From Figure 4-10 we also confirm that OOD detector has a harder time identifying the background sources as OOD, and only achieves a 0% FNR at an SNR > 80 and OOD proportion > 70%. This is again explained by the high similarity between the BG signatures and the IND training seeds (Figure 3-9).

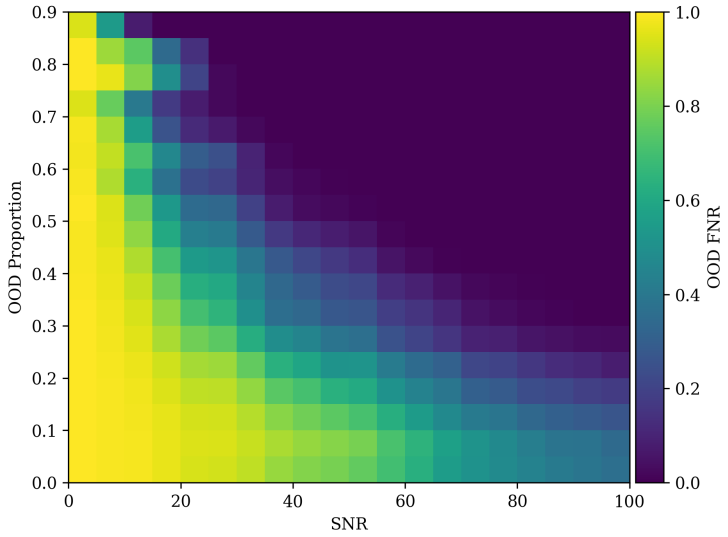


Figure 4-9. Heatmap showing the OOD FNR as a function of OOD proportion and SNR for the measured OOD test spectra generated from lab measurements (OOD sources: Bi207, Cs137)

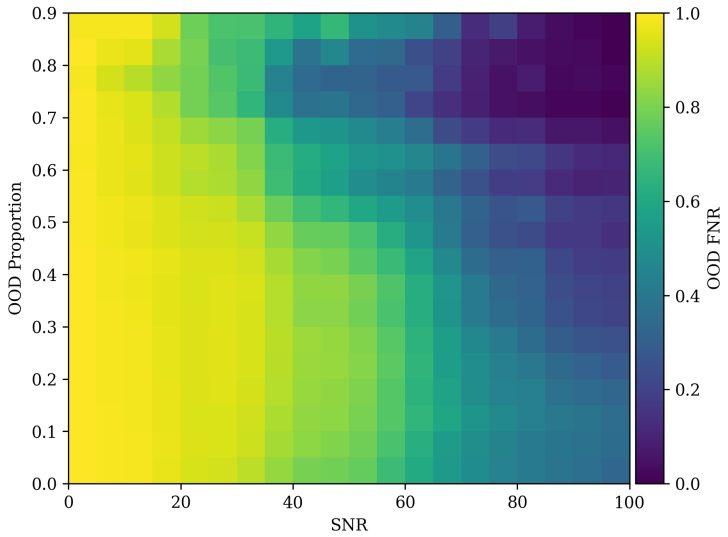


Figure 4-10. Heatmap showing the OOD FNR as a function of OOD proportion and SNR for the BG OOD test spectra generated from lab measurements (OOD sources: K40, Ra226, Th232, Cosmic)

4.3. Computational Reference

The LPE and OOD detection models are not computationally expensive and should capable of better than real-time performance on synthetic and measured spectra. The models, which were run on a 2.4 GHz 8-core Intel Core i9 processor with 32 GB of memory, were able to process spectra at speeds > 50 kHz. This is well above the required speed threshold for most realistic scenarios. The LPE model itself is relatively small (55807 parameters) and should be easily embedded in a lightweight compute system. The LPE and OOD detection models took about 10 minutes to train on the aforementioned computer.

This page intentionally left blank.

5. RECOMMENDED USAGE

In Section 3.6, the reproduction of this study is discussed which also indirectly empowers the reader to create similar models for their own problems. This is, in fact, the purpose of the PyRIID package: to enable both ML practitioners and nuclear SMEs to create ML solutions for RIID as quickly, correctly, and easily as possible. To that end, this section provides additional recommendations from the authors with respect to how such models are intended to be used.

The most straight-forward way we see to incorporate this technology is as a supplemental tool SMEs use to perform an initial, batch screening of large quantities of gamma spectra. As illustrated in Figure 5-1, this technique could be used to generate an alarm under specific scenarios (based on the OOD detector and sample SNR) which would indicate further investigation is required by a spectroscopist. In the case that the spectrum of interest is detected as IND, the radioisotope proportion estimates can serve as an initial estimate of the true isotopic proportions.

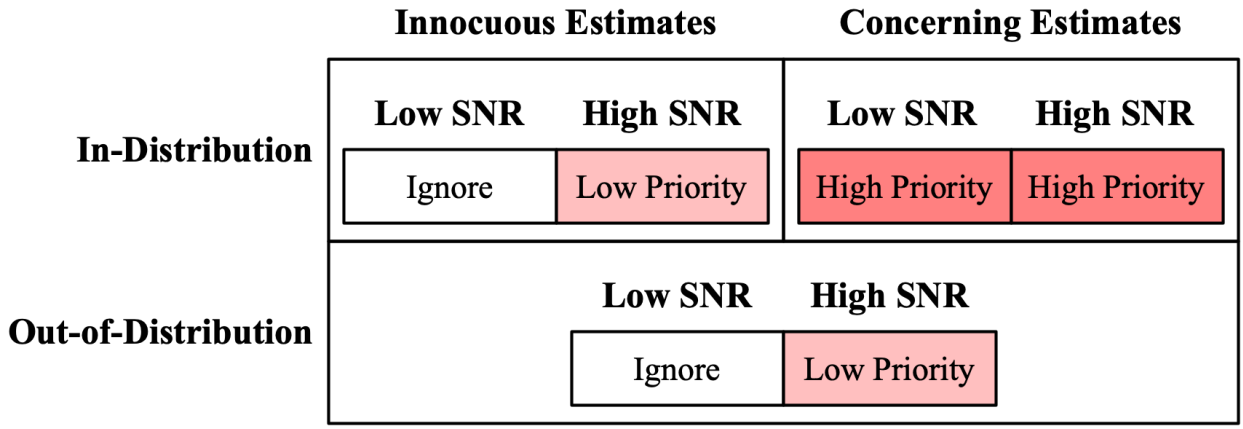


Figure 5-1. Prioritization matrix for samples analyzed by LPE and OOD models

The threshold of low versus high SNR is a simplification which can be adjusted to represent multiple levels of severity, all of which most often relate to dose-related, safety concerns. How estimates are interpreted as innocuous vs. concerning is application-specific, and likely related to post-processing of estimates into more useful quantities such as source activity or activity ratios between specific sources.

This page intentionally left blank.

6. CONCLUSION

This study demonstrates that the LPE model, trained and tuned solely using synthetic gamma spectra, successfully performed LPE on real gamma spectra. Additionally, by properly tuning the trade-off parameter (β) between the two terms of the custom, semi-supervised loss function the model was able to achieve reasonable OOD detection performance without compromising the isotope proportion estimates. Having a built-in method for OOD detection enables the model to know when to not trust its output because the input spectrum is OOD, which improves confidence in the model.

In particular, the LPE model successfully identified the dominant isotope in each of the lab measurements (Figure 4-1) even though it was primarily trained to predict on mixture spectra (Figure 3-11). The LPE model also achieved a MAE of 0.07 on the measured test set which was imbalanced towards low SNR samples. The OOD detection model correctly identified measured OOD spectra which contained the OOD sources Bi207 and Cs137 (Section 4.2.3) with a 0% FPR for OOD proportions above 50% at 50 SNR or higher.

The cases where the models seemed to perform worse are predictable and can be explained in terms of the spectral uniqueness of the sources used in this study (Section 3.3.3). For example, although correctly identifying measurements of Ba133, the LPE model consistently predicted non-zero proportions of Cf252 which was not present (Figure 4-1). This confusion, however, can be explained as Ba133 and Cf252 have a similar spectral shape and small JSD between them. The OOD detection model also had a difficult time detecting pure background components (K40, Ra226, Th232, and Cosmic) as OOD (Fig. 4-10). This too can be explained as these background sources are much more similar to the IND synthetic seeds than Bi207 or Cs137 (Section 3.3.3). Practical consideration aside, potential remedies we surmise for improving OOD detection on measurements include, but are not limited to, the following: (1) region-specific reconstruction instead of full spectrum reconstruction; (2) increasing overall measurement SNR; and (3) using a detector with better resolution.

The performance of the models (in terms of both isotope proportion estimation and OOD detection) will only be as good as the spectral signatures used to generate training data, and to be more specific, how similar those signatures are to the actual source signatures in the models' natural habitat. When generating a training dataset many assumptions must be made to create the synthetic seeds, and although effort was made to ensure they matched as close as possible for this study (such as adjusting the energy calibration and using an appropriate DRF for our detector), it would be unrealistic to expect perfect alignment with the testing environment. Moreover, there are also unpredictable, real-world effects that change the shape of measured gamma spectra (Section 2.1). This dilemma, known as the sim-to-real gap, will always be a challenge for ML approaches to RIID, which will heavily rely on synthetic training data if they are to be applied to complex problem spaces. In light of this predicament, the ability of an ML model to "know what it does not know" (OOD detection)

is absolutely crucial and should be given a reasonable degree of consideration at all times. Having built-in OOD detection capabilities enables our method to identify when a spectrum is sufficiently different from training such that it should not be used, and is critical for having a sense of confidence in the estimated proportions.

REFERENCES

[1] A. Van Omen, "A Semi-Supervised Model for Multi-Label Radioisotope Classification and Out-of-Distribution Detection," Master's thesis, University of Michigan (Ann Arbor), 2023.

[2] G. F. Knoll, *Radiation Detection and Measurement*. John Wiley & Sons, 2010.

[3] M. Rawool-Sullivan, J. Bounds, S. Brumby, L. Prasad, and J. Sullivan, "Steps Toward Automated Gamma Ray Spectroscopy," tech. rep., Los Alamos National Laboratory, Los Alamos, NM (United States), 2010.

[4] D. J. Mitchell, L. Harding, G. G. Thoreson, and S. M. Horne, "GADRAS Detector Response Function," tech. rep., Sandia National Laboratories, Albuquerque, NM (United States), 2014.

[5] T. Morrow, N. Price, and T. McGuire, "PyRIID v.2.0.0," April 2021. https://doi.org/10.11578/dc.20221017.2.

[6] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Springer, 2009.

[7] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[8] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2018.

[9] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. Springer, 2006.

[10] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[11] A. Martins and R. Astudillo, "From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification," in *International Conference on Machine Learning*, pp. 1614–1623, PMLR, 2016.

[12] M. Kamuda and C. J. Sullivan, "An Automated Isotope Identification and Quantification Algorithm for Isotope Mixtures in Low-resolution Gamma-ray Spectra," *Radiation Physics and Chemistry*, vol. 155, pp. 281–286, 2019. https://doi.org/10.1016/j.radphyschem.2018.06.017.

[13] J. Kim, K. Park, and G. Cho, "Multi-Radioisotope Identification Algorithm Using an Artificial Neural Network for Plastic Gamma Spectra," *Applied Radiation and Isotopes*, vol. 147, pp. 83–90, 2019. https://doi.org/10.1016/j.apradiso.2019.01.005.

[14] J. M. Ghawaly, A. D. Nicholson, D. E. Archer, M. J. Willis, I. Garishvili, *et al.*, "Characterization of the Autoencoder Radiation Anomaly Detection (ARAD) Model," *Engineering Applications of Artificial Intelligence*, vol. 111, p. 104761, 2022. https://doi.org/10.1016/j.engappai.2022.104761.

[15] A. Khatiwada, M. Klasky, M. Lombardi, J. Matheny, and A. Mohan, "Machine Learning Technique for Isotopic Determination of Radioisotopes Using HPGe γ-ray Spectra," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 1054, p. 168409, 2023. https://doi.org/10.1016/j.nima.2023.168409.

[16] J. R. Stomps, P. P. H. Wilson, K. J. Dayman, M. J. Willis, J. M. Ghawaly, *et al.*, "SNM Radiation Signature Classification Using Different Semi-Supervised Machine Learning Models," *Journal of Nuclear Engineering*, vol. 4, no. 3, pp. 448–466, 2023. https://doi.org/10.3390/jne4030032.

[17] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[18] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[20] A. Van Omen and T. Morrow, "sandialabs/baldr-study-07.05: v1.0.0," Dec. 2023. https://doi.org/10.5281/zenodo.10278474.

APPENDIX A. Spectral Plots

A.1. Measured

This appendix contains the plots of all the spectral measurements taken in the lab including both the gross measurements of sources and the long-collect background measurement.

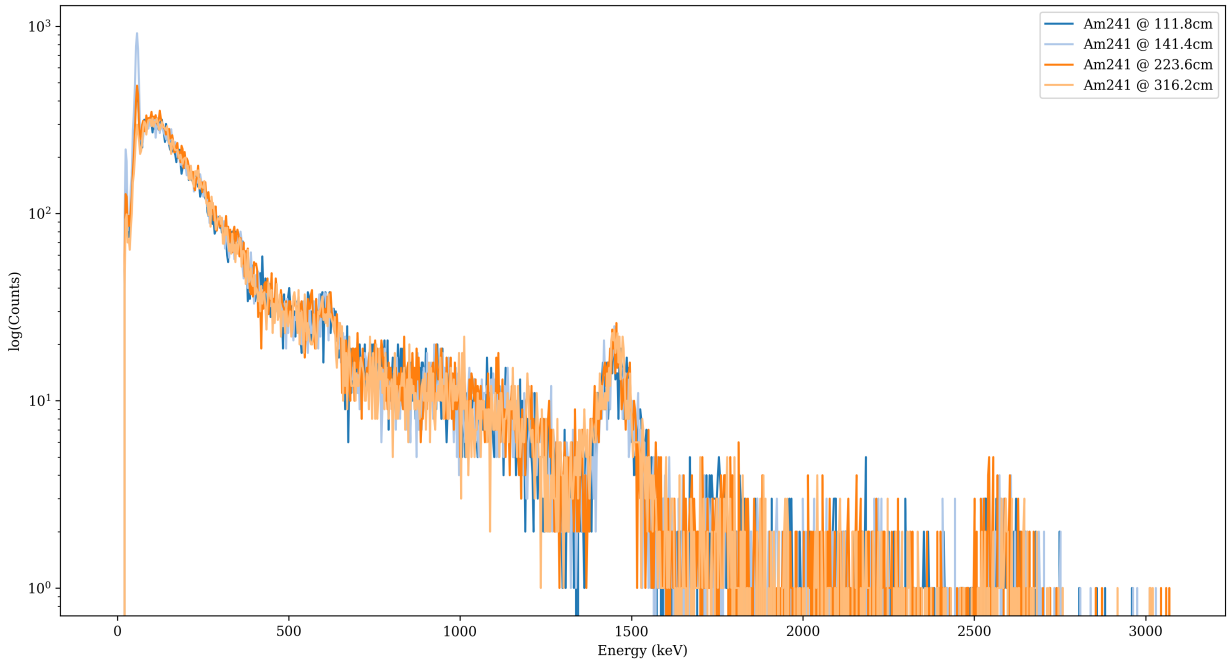


Figure A-1. Gross measurements of Am241 source

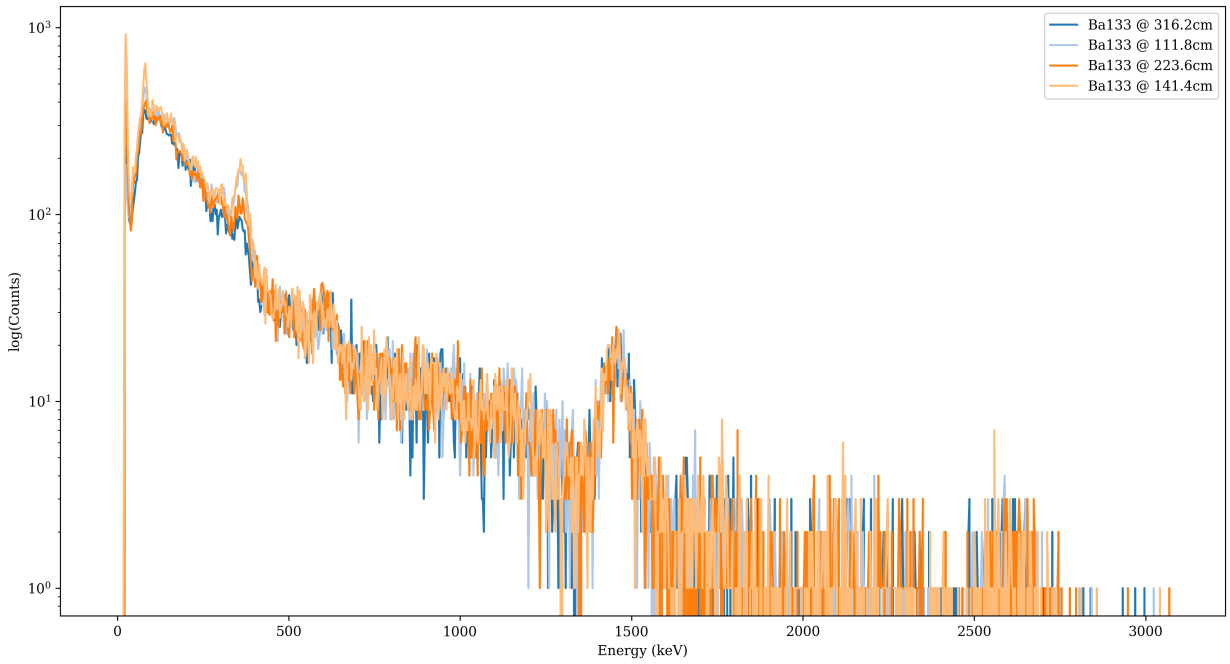


Figure A-2. Gross measurements of Ba133 source

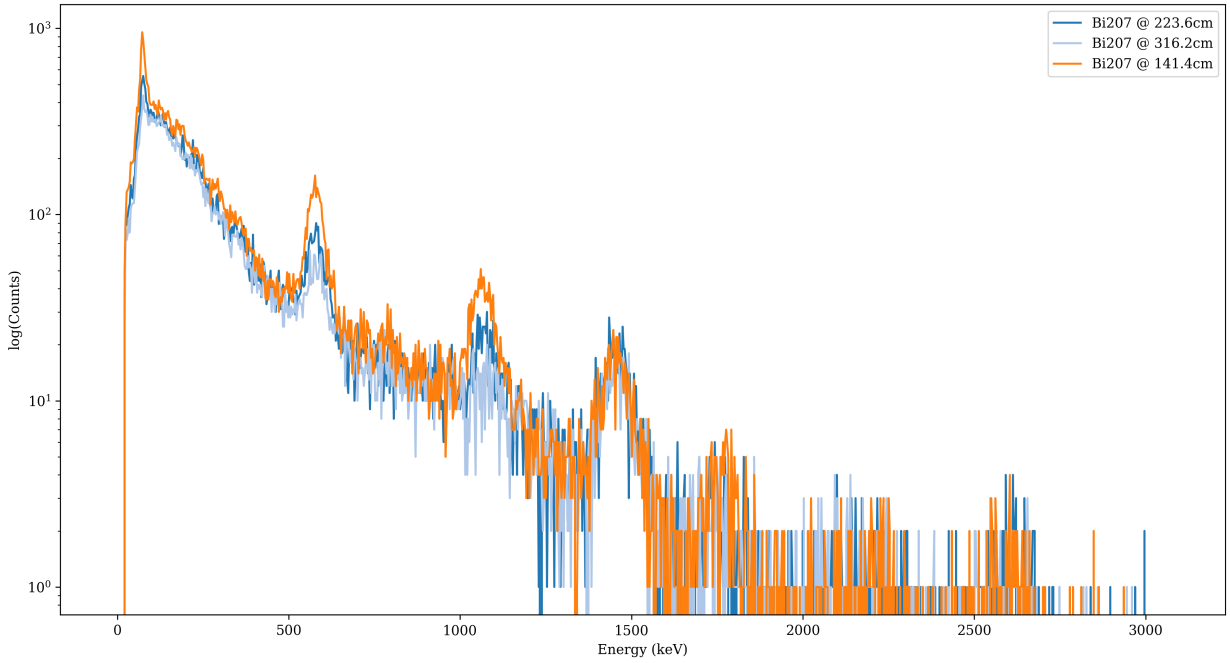


Figure A-3. Gross measurements of Bi207 source

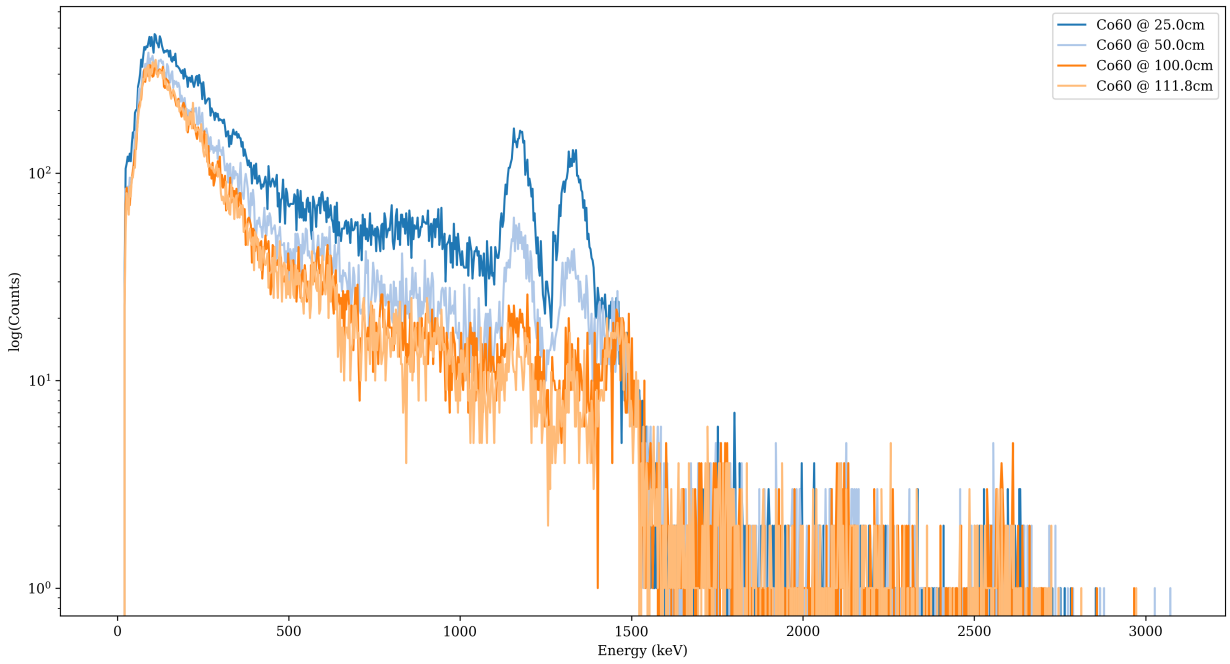


Figure A-4. Gross measurements of Co60 source

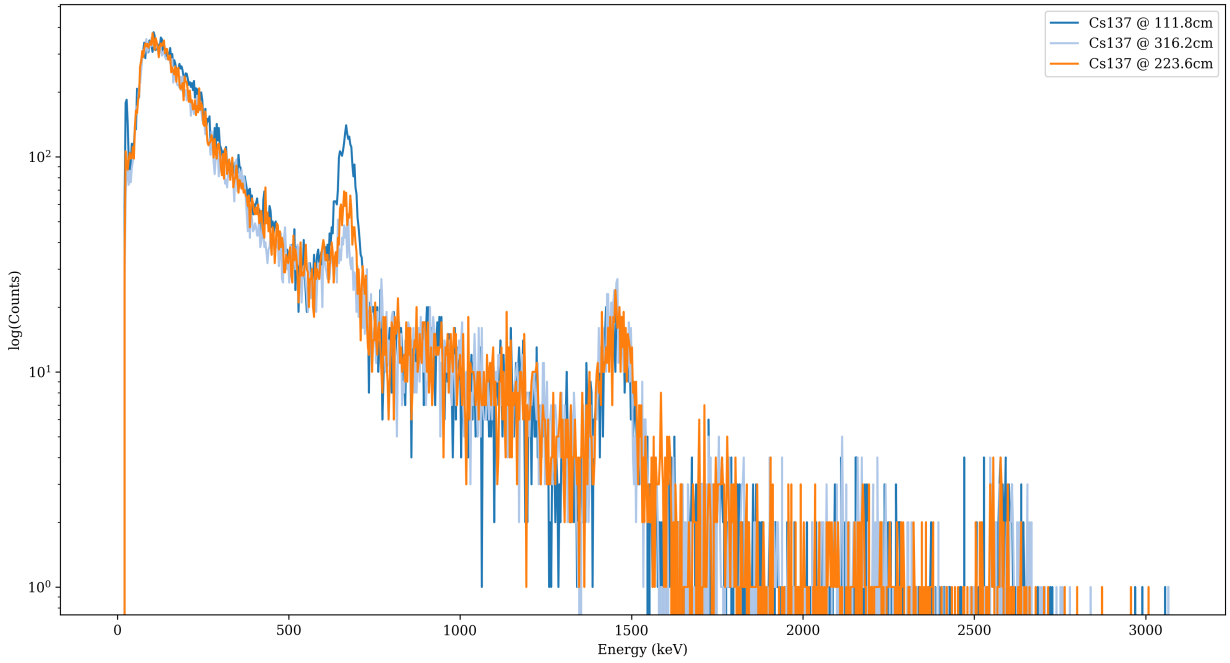


Figure A-5. Gross measurements of Cs137 source

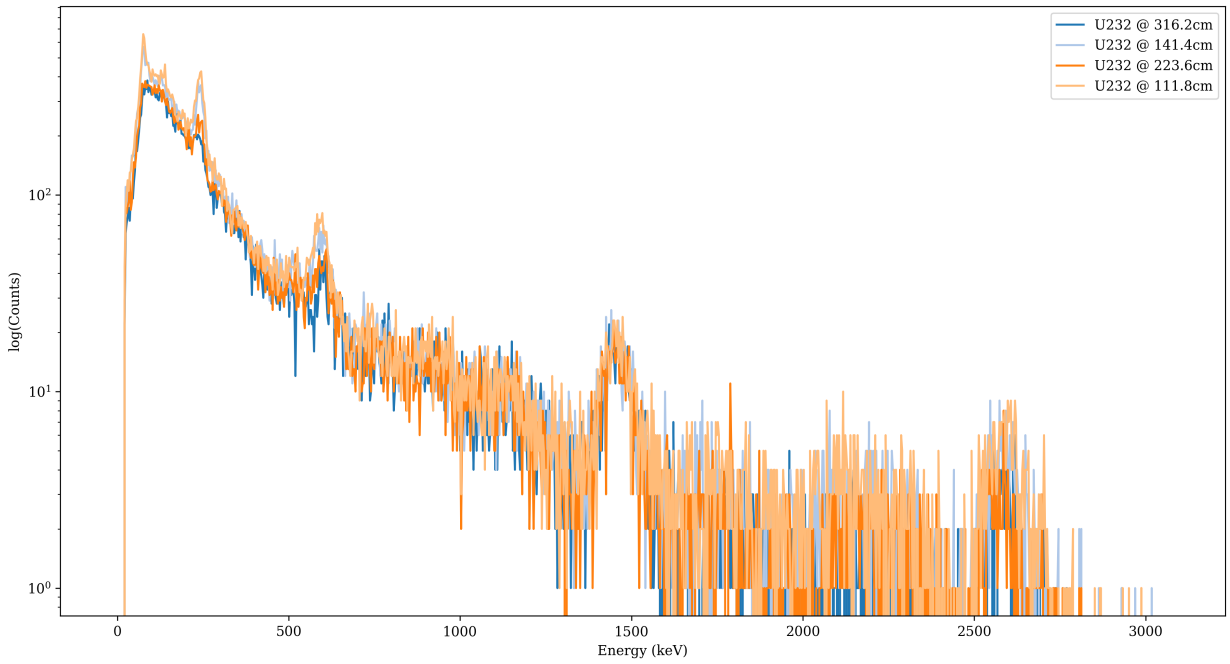


Figure A-6. Gross measurements of U232 source

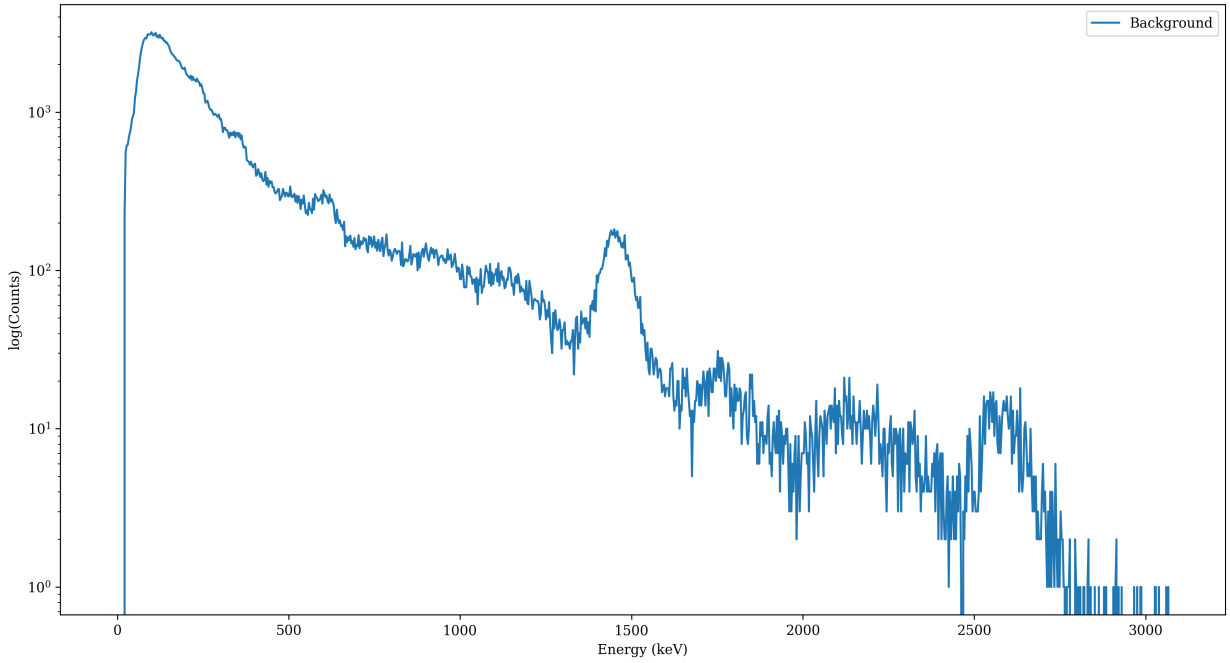


Figure A-7. Long-collect background measurement

A.2. Synthetic

This appendix contains the plots of all the synthetic seeds used in this study including the FG and BG seeds.

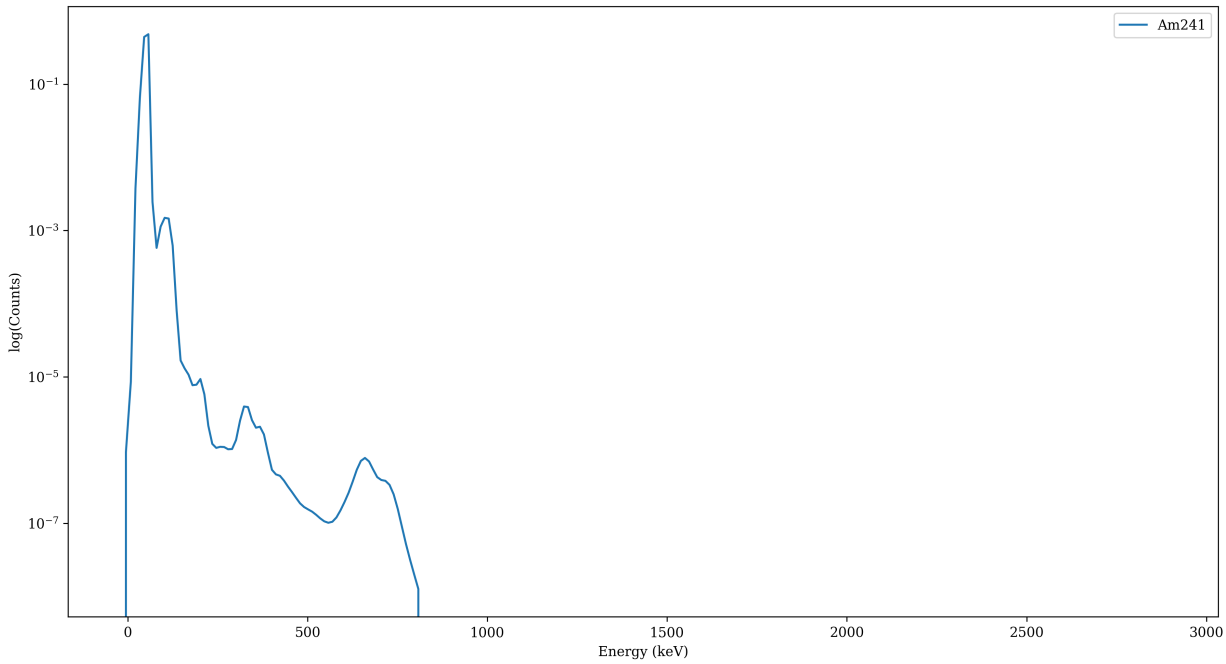


Figure A-8. Synthetic seed for Am241

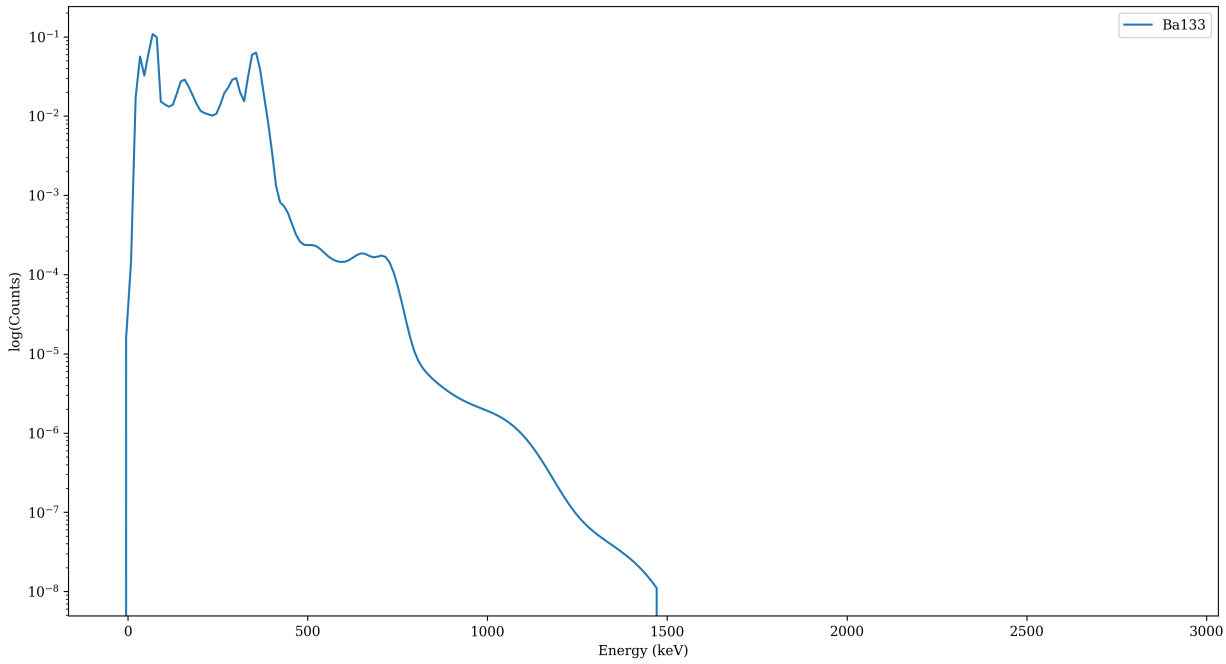


Figure A-9. Synthetic seed for Ba133

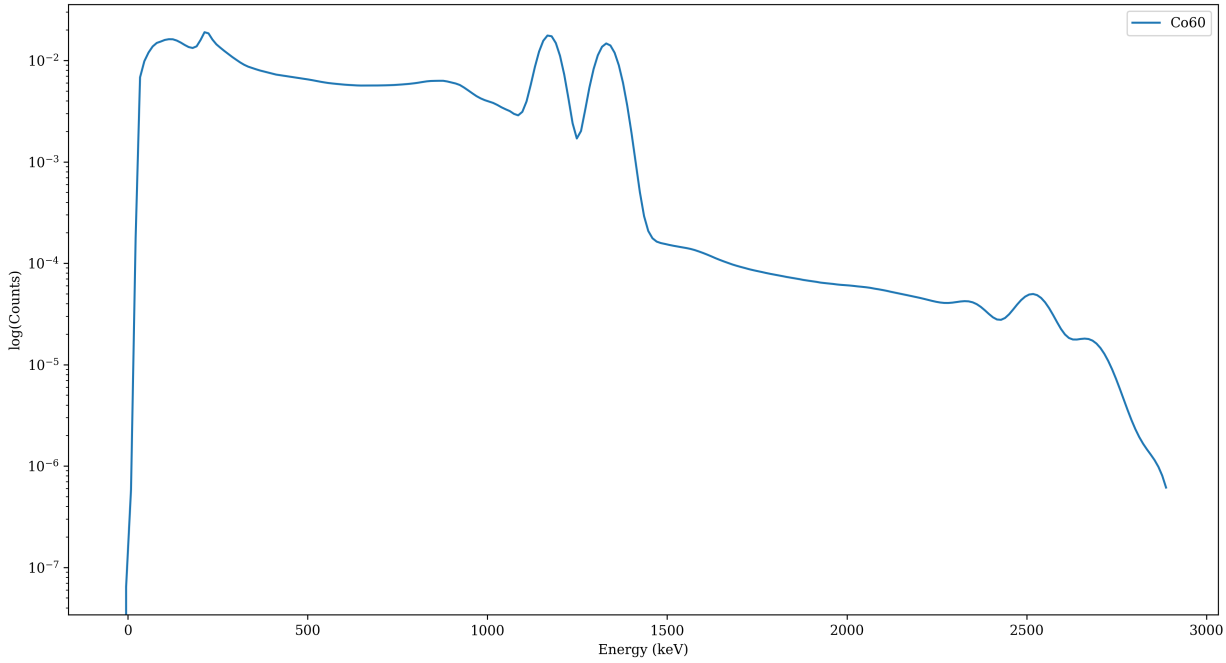


Figure A-10. Synthetic seed for Co60

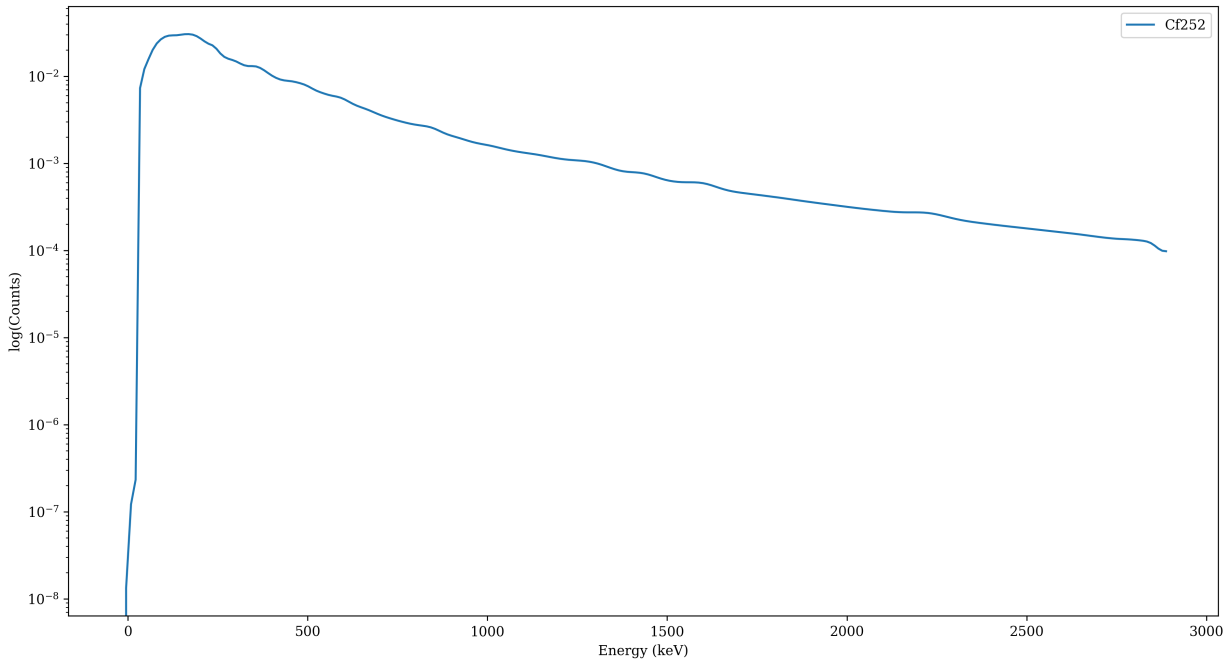


Figure A-11. Synthetic seed for Cf252

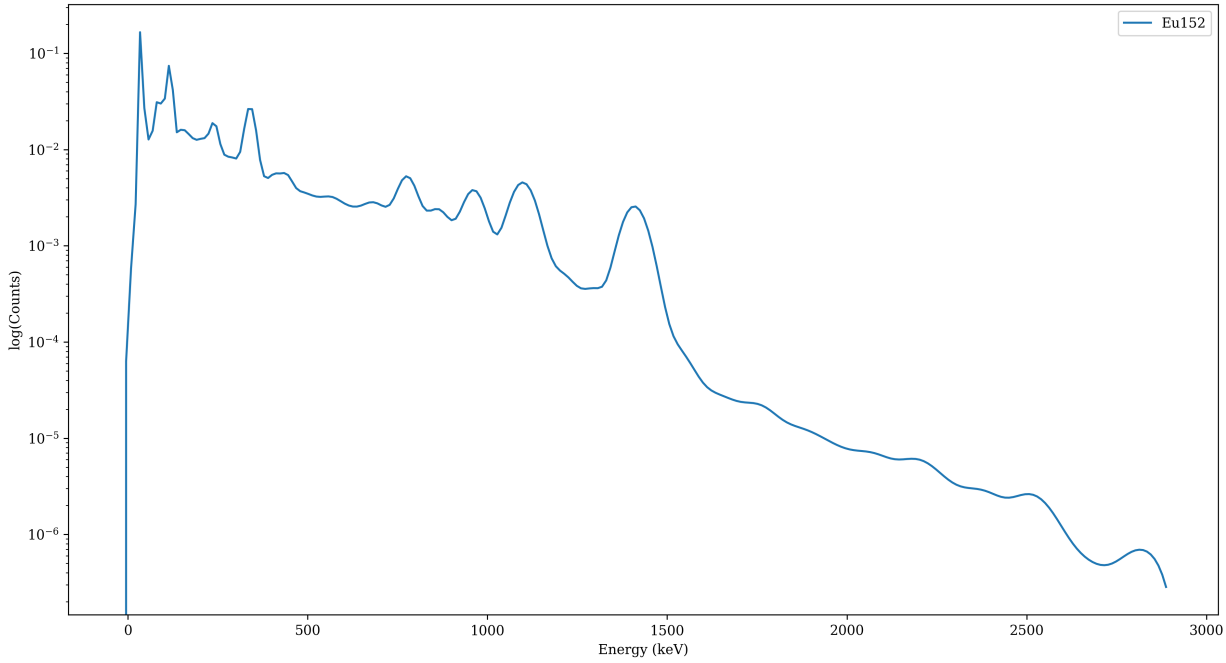


Figure A-12. Synthetic seed for Eu152

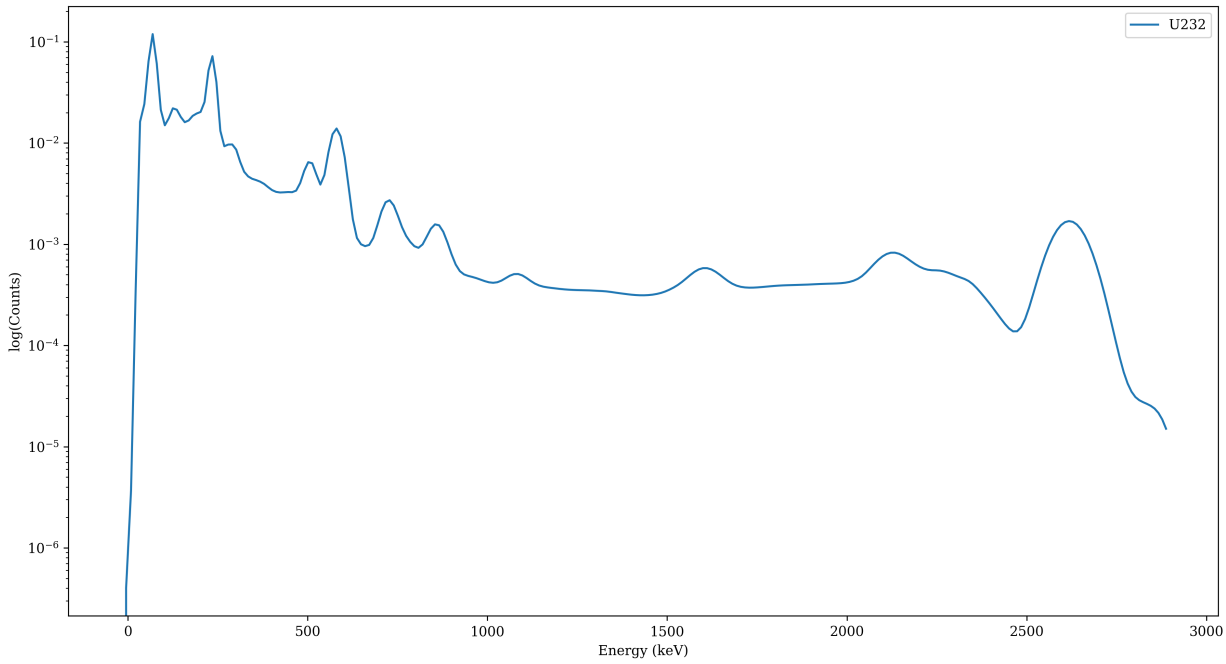


Figure A-13. Synthetic seed for U232

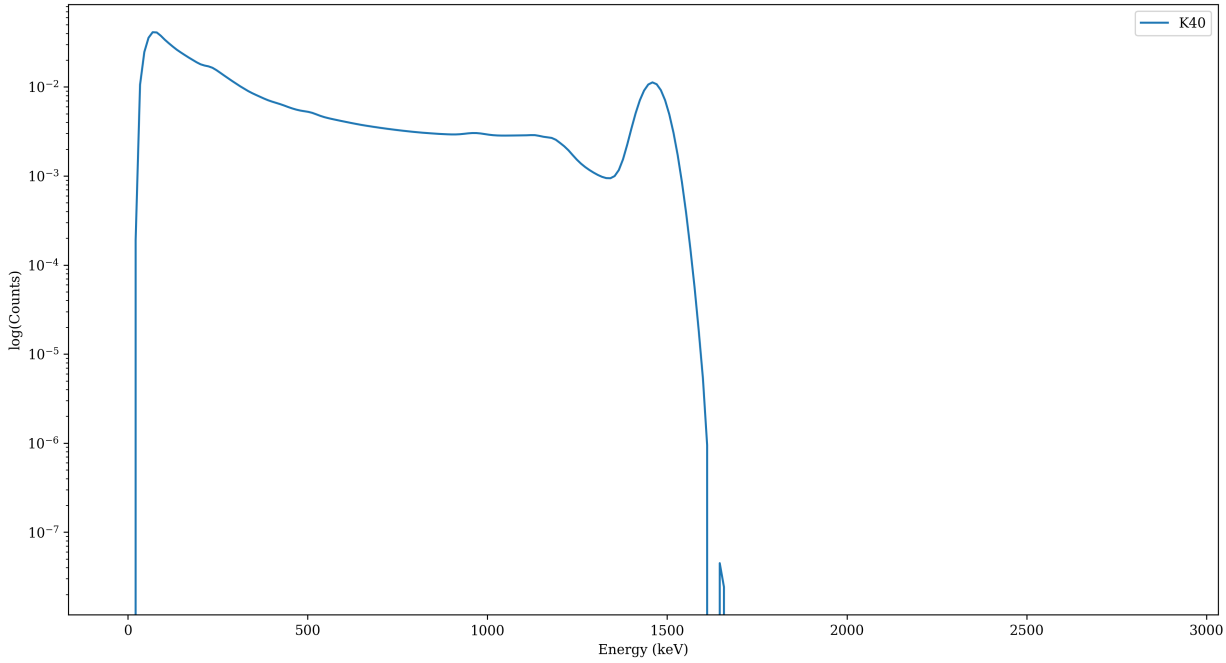


Figure A-14. Synthetic seed for K40

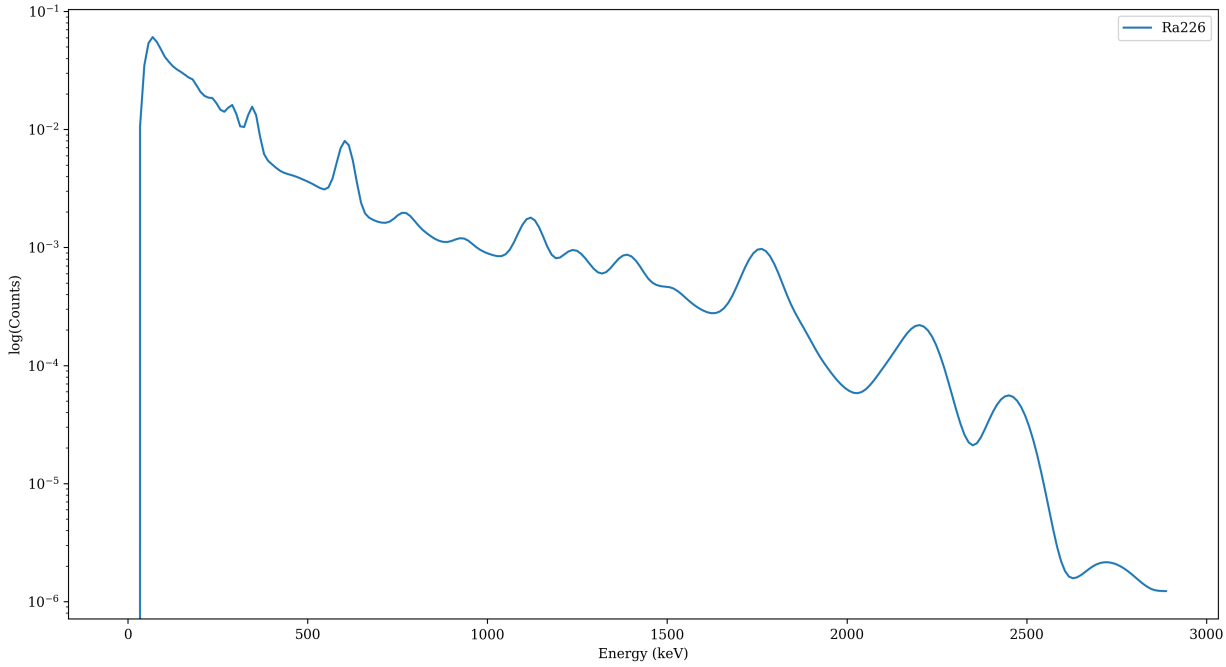


Figure A-15. Synthetic seed for Ra226

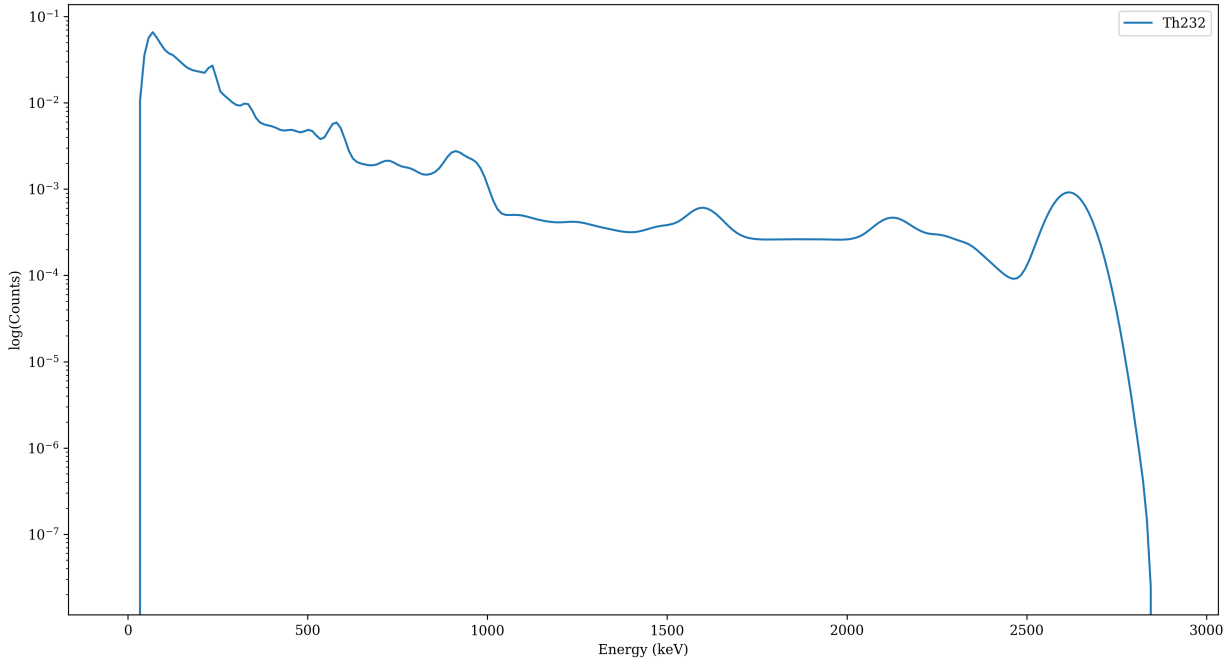


Figure A-16. Synthetic seed for Th232

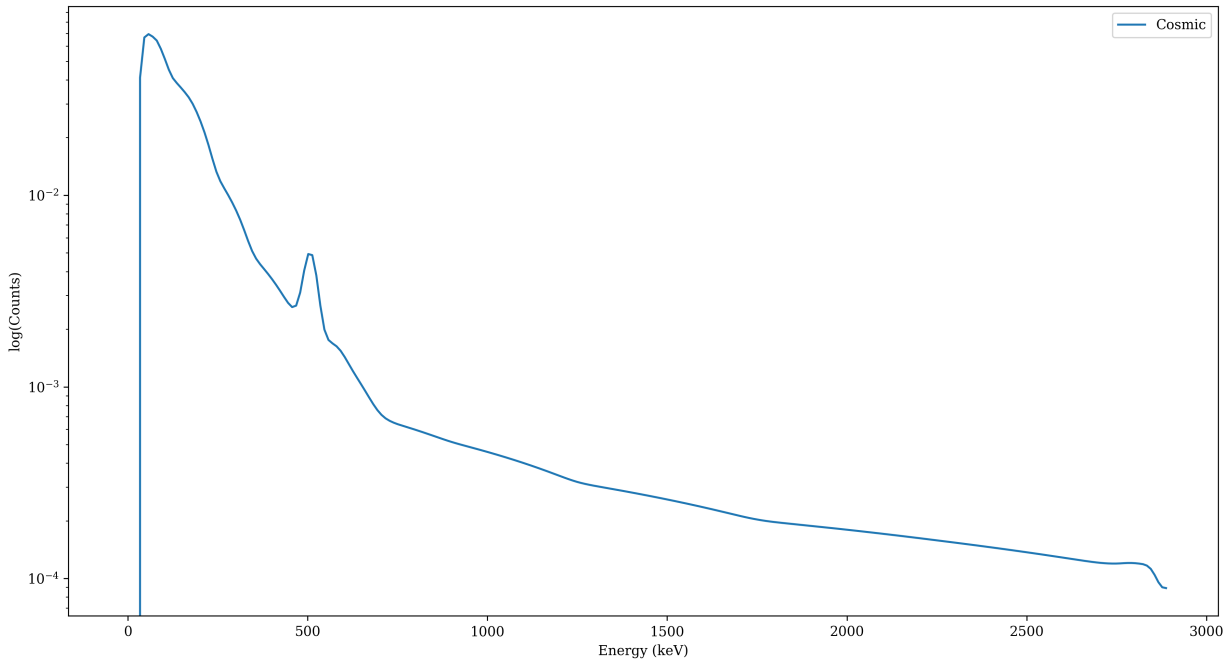


Figure A-17. Synthetic seed for Cosmic Radiation

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

Email—External

Name	Company Email Address	Company Name
Hank Zhu	hank.zhu@doe.nnsa.gov	DOE NNSA
Jake Zappala	jake.zappala@doe.nnsa.gov	DOE NNSA

This page intentionally left blank.

This page intentionally left blank.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.