2022/2023 Internship Report

Gamma Spectrometry using AI on the Segmented nFacet3D Detector

Laboratoire de physique corpusculaire de Caen

11/09/2023

Özgür Özer | *Erasmus Mundus Master's*

ozgurozer@tutanota.com | *NucPhys Programme*

Supervisor: Antonin Vacheret

Contents

Abstract

Conventional nuclear spectrometry requires the expertise of a physicist, expensive detector materials and a commercial sofware programme. This study investigates a way to conduct spectrometry sans a physicist and proprietary software programmes using a cost-effective scintillator material and AI. Geant4 simulation is used to acquire the probability density function (PDF) of energy deposition spectra from 12 nuclides on nFacet3D detector system that had been developed within this project previously. The response of the nuclides on the detector is calculated from the energy spectra. Two individual samples are created from up to three randomly chosen nuclides for detector response and energy spectrum with random sample count. The initial model was a replicate of a model from a previous research to recreate the energy spectrum from the detector response. Then two more AI models with A feedforward neural network (FNN) and Convolutional neural network (CNN) were devised in this project to modernise and enhance the initial model. Another model was engineered on top of the better performing model to predict whether each of 12 nuclides is present in the sample from the detector response using multilabel binary classification. Reconstruction of the energy spectrum exhibit up to 0.96 R2 score. In average, identification of a single nuclide reached the accuracy of 0.99. Probability of finding all the nuclides present in a mix of nuclides correctly and not finding the ones absent is 0.91

Keywords: Nuclear Spectrometry, Artificial Intelligence, Geant4 Simulation, PVT Scintillator, Neural Networks, Multilabel Classification, Radiation Monitoring, Nuclide Identification.

Introduction

Nuclear spectrometry has long been a cornerstone in the field of radiation dosimetry, providing critical insights into the energy distribution of ionizing radiation. Traditionally, this intricate science required the expertise of physicists well-versed in the complex mathematics and data analysis techniques necessary to interpret spectra. Additionally, the utilization of commercial software programs often incurred substantial costs, limiting accessibility to this crucial technology. In addition, the intersection of artificial intelligence (AI) and cost-effective materials like Polyvinyl Toluene (PVT) scintillators presents a promising avenue to reduce the costs and the expert involvement.

Nuclear spectrometry has evolved significantly over the years, with advances in detector technology and data analysis techniques enabling increasingly accurate measurements of radiation energy spectra. Historically, the analysis of radiation energy distributions required the application of complex mathematical algorithms, frequently beyond the capabilities of non-experts. The process typically involved the identification of spectral peaks, followed by energy calibration and deconvolution to extract meaningful information. This inherently specialized and labor-intensive process limited the adoption of nuclear spectrometry. Machine learning algorithms, specifically neural networks, have demonstrated remarkable abilities in pattern recognition and data analysis. This has opened the door to automating many of the intricate tasks traditionally performed by physicists. In the context of nuclear spectrometry, AI algorithms can be trained to identify spectral features, perform energy calibration, and even extract dosimetric parameters with minimal human intervention. One of the most significant advantages of AI-driven nuclear spectrometry is its ability to learn and adapt continuously. Through iterative training, these algorithms can refine their accuracy, improving their performance over time. This adaptability is particularly valuable in the context of dosimetry, where changing radiation sources and conditions necessitate constant calibration and adjustments.

In parallel with the enhancements by AI, the affordability and accessibility of sensing materials have become paramount. Polyvinyl Toluene (PVT) are a cost-effective alternative to inorganic scintillation materials or silicon detectors. Their low cost and ease of handling make them an attractive choice for researchers seeking to reduce the financial burden associated with dosimetry experiments. However, PVT scintillators exhibit poor energy resolution, making them inadequate for nuclear spectrometry applications.

This research article delves into the approach that harnesses the power of AI, specifically machine learning algorithms, to perform nuclear spectrometry for dosimetry with higher precision and efficiency. Additionally, it explores the utilization of low-cost PVT scintillators as the sensing medium, thereby minimizing the financial barriers to entry. This innovative combination has the potential to reshape the methodology of radiation dosimetry, enabling faster and cheaper measurements with fewer data meanwhile reducing the burden on physicists.

Methods and Materials

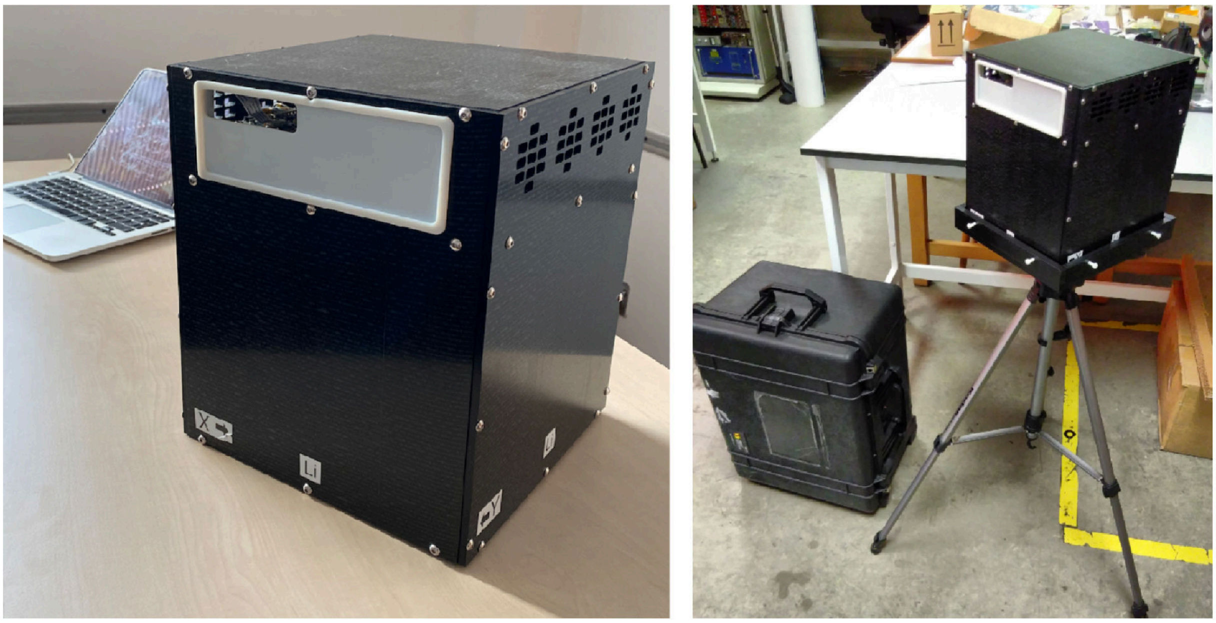

Figure 1: M1 detector system

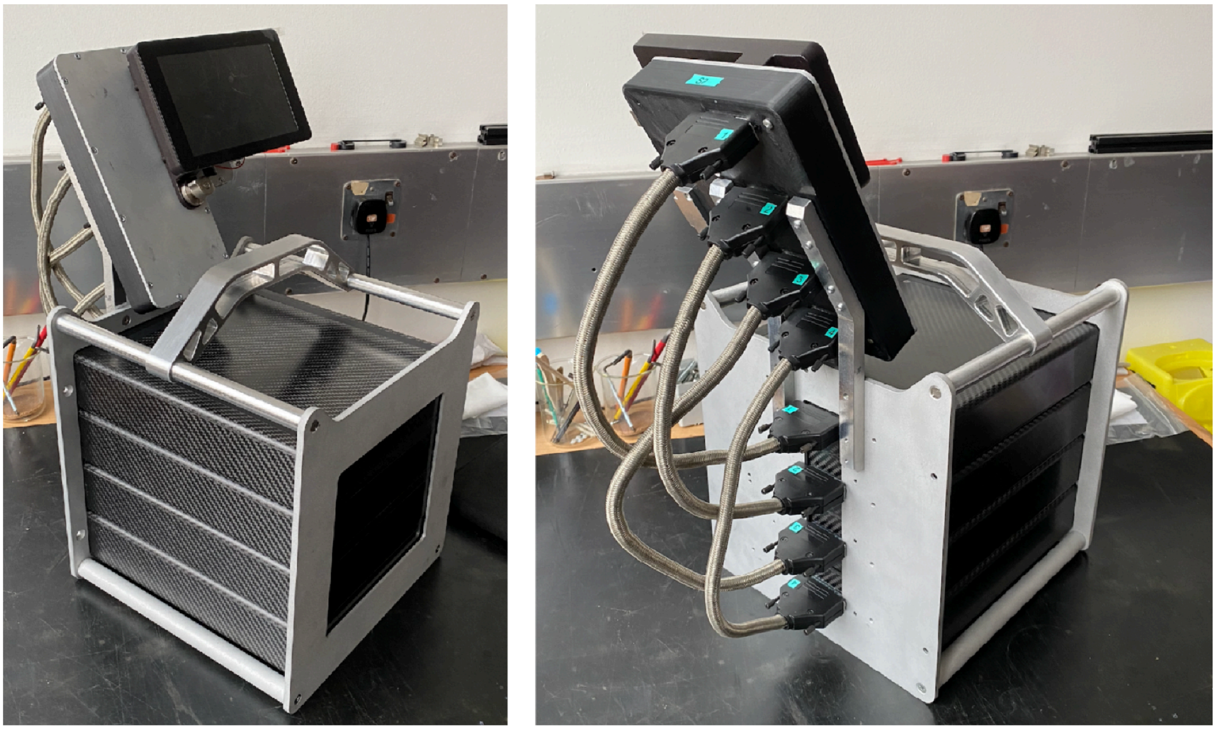

Figure 2: M2 detector system

nFacet3D project has two detector prototypes: M1 and M2. Both of these prototypes are composed of 4 × 4 voxels of polyvinyl toluene (PVT) scintillator, EJ-200. They are cubes of size 5 cm × 5 cm × 5 cm. M1 system has a 0.25 mm LiF:ZnS layer on three side of the voxels and M2 system has one layer thereof on the top of the voxels. That lithium has 95% ^{6}Li and 5% ^{7}Li. LiF:ZnS layer and the PVT is contained inside a Tyvec high-density polyethylene fibres to confine the scintillated light inside. The scintillated light inside the Tyvec is transferred to the MPPC sensor using wave length shifting (WLS) fibres.

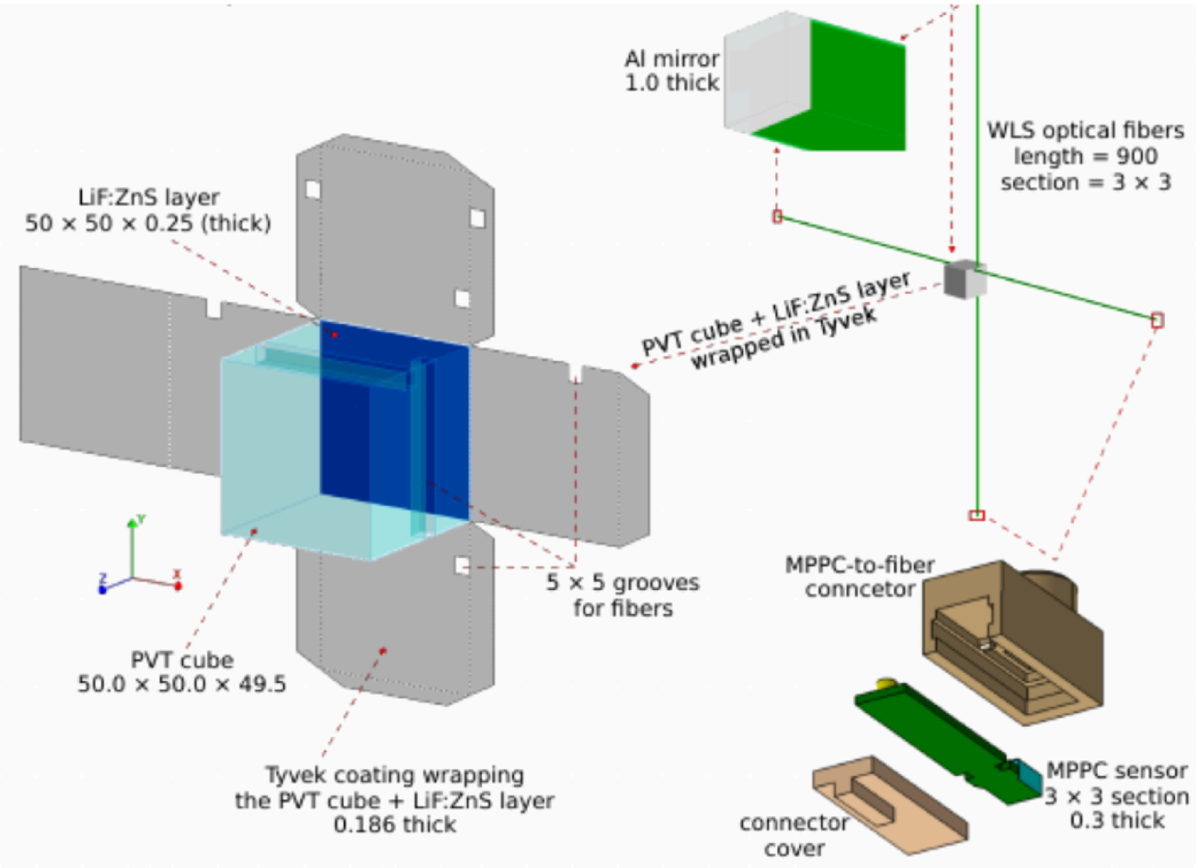

Figure 3: Structure of a single voxel

Simulation

To obtain optimal energy spectra from diverse sources, a Geant4 simulation is implemented using Geant4 version 11.1.2[1], [2], [3]. The choice of employing GDML technology for geometry is justified due to the intricate nature of the detector's geometry, which comprises numerous intricate components. Moreover, the simulation benefits from the availability of the detector's geometry in the .STEP file format. This Geant4 simulation encompasses the modelling of neutrons and muons, in addition to gamma particles, to support future research endeavours and the collaborative efforts of other project members. This multifaceted simulation is graphically represented in the Figure 4 demonstrating the distinct separation of gamma and neutron events.

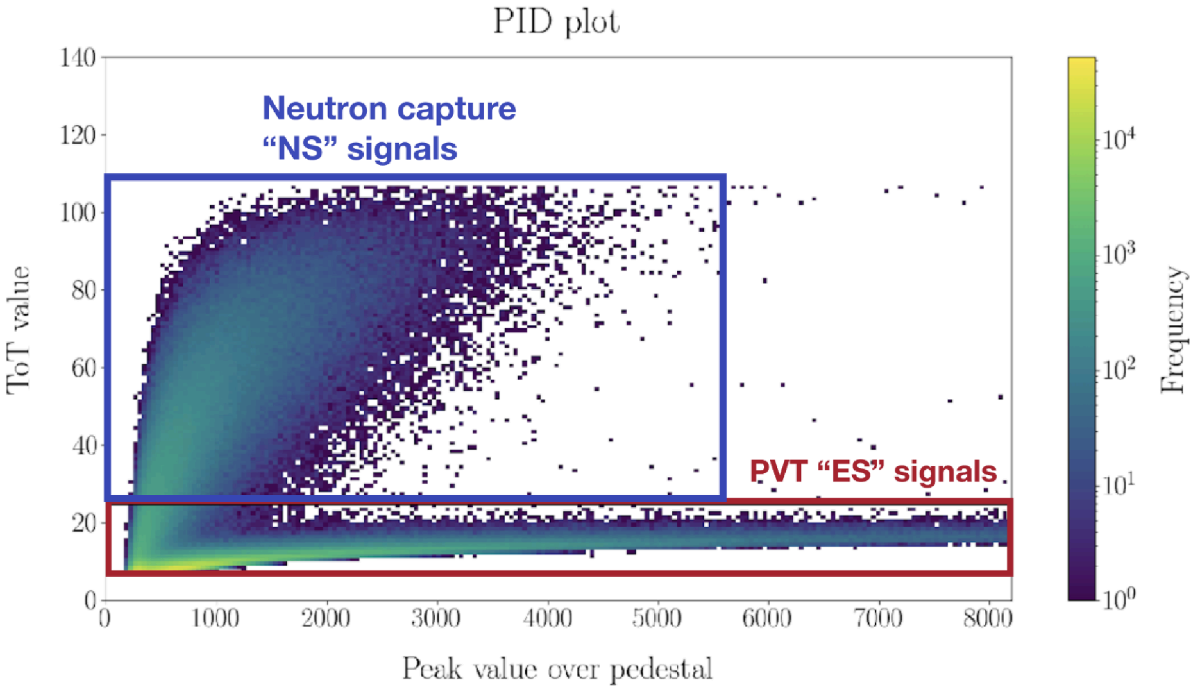

Figure 4: Time-Over-Threshold and Peak value over pedestal for particle identification (PID).

NS: Nuclear scintillation, ES: electron scintillation

The simulation had the detector mentioned above and 12 different sources are created 1 m away from the centre of the detector. The sources were ^{60}Co, ^{241}Am, ^{137}Cs, ^{85}Kr, ^{192}Ir, ^{226}Ra, ^{22}Na, ^{131}Ba, ^{68}Ga, ^{152}Eu, ^{131}I, ^{210}Po, ^{252}Cf, AmLi, AmBe. From this point on, these nuclides will be referred without the mass number because there is no other isotope. Each source was simulated at 25 different positions, starting from the φ angle of the spherical coordinates 0 ° to that of 90 °. My colleague in the same project, Eliot Martin, worked on the conversion from .STEP to .gdml and the script creating macros for the configuration mentioned. ROOT 6.28.04 [4] is used to record the results and inspect them. In the end, the following variables are recorded in the designated .root files for gamma, neutron, and muon:

- event ID
- the cube in which the energy deposition takes place
- the face of the detector from which the initial particle enters
- the kinetic energy of the initial particle when it enters the detector

- energy of the particle when it is created
- whether the neutron is captured by the ^{6}Li
- energy deposited

The plastic scintillator used in this detector doesn't show the Compton edge properly due to the nature of organic scintillators. The simulation of this detector response required a convolution of the ideal spectrum probability density function (PDF) with a special kernel [5]:

$$S^{*}(E,a,b,c) = f\left(E_0,a,b,c\right) * S\left(E_0\right) \tag{1}$$

where:

$$f\left(E_0,a,b,c\right) = Ae^{-\left(\frac{2\sqrt{2\ln 2}(E-E_0)}{\mathrm{FWHM}(E_0,a,b,c)}\right)^2} \tag{2}$$

$$\mathrm{FWHM}\left(E_0,a,b,c\right) = a + b\sqrt{E_0 + cE_0^2} \tag{3}$$

S^{*} is the spectrum with Gaussian energy broadening (GEB), S is the ideal spectrum and f is the convolution kernel. When normalised with A, f function also gives the PDF of E undergone GEB.

To simplify the a and c values were cancelled to have a Gaussian kernel in our work:

$$\bar{f}\left(E_0,a,b,c\right) = Ae^{-\left(\sigma\frac{E-E_0}{\sqrt{E}}\right)^2} \tag{4}$$

The sigma value for the nFacet detector was taken 0.1 as a sensible placeholder.

The .root file of each source-position file is read via uproot 4.3.5 [6]. For each energy deposition value, the f distribution is formed and a random variable is drawn from it. Because of noise values below a certain threshold is discarded in the detector readout – this is takes as .1 MeV in our case as a placeholder. As a result, only the energy values above this threshold is present in the spectrum. This newly created and the original energy distribution are put in a histogram with 500 bins with limits 0.1 and 1.6 MeV. Then the value in each bin is divided by the total sum of bins to acquire the PDF.

Two datasets are created for two different tasks: energy spectra reconstruction and nuclide identification. The datasets were created in regard to possible stochastic changes in real life scenarios. The process is as follows for the dataset of reconstruction task. First, a random integer between 1 and 3 inclusively is chosen for the number of sources mixed in the spectrum. That many sources and their ratios are then chosen randomly again. The number of counts in those spectra is kept between 40000 and 100000. With these sources and according to their ratios two spectra are sampled for the ideal energy deposition and detector response from their PDF functions. These are designated as input and target data for the ML model. For the source identification task, the input is the same; the target is the the sources present in the sample in the binary multilabel target format. All these are stored in .npy binary files using NumPy 1.25.1 [7].

There were various models to reconstruct the energy deposition from the detector response. The models are built using Lightning AI 2.0.6 [8] with PyTorch 2.0.1 [9] with the help of Tensorboard 2.13.0 [10] and MLflow analysis. Seed is taken as 42 for all of them. The rig utilised in this project was a computer with two AMD EPYC 7763 64-Core CPU and eight Nvidia A100 80 GB GPU along with 503 GB RAM.

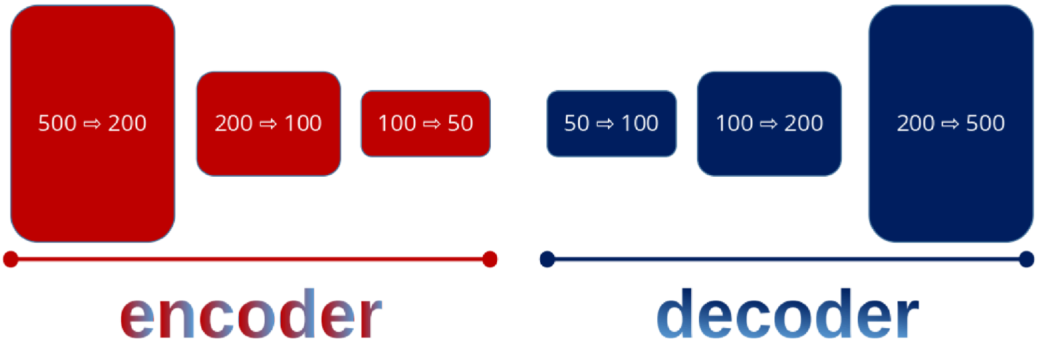

Figure 5: Default Model with Relu activation. Rounded rectangles are linear layers

First one was almost exactly the same as the one in the article *Reconstruction of Compton Edges in Plastic Gamma Spectra Using Deep Autoencoder* [11]: 500 dense, ReLU [12], 250 Dense, ReLU, 100 Dense, ReLU as encoder and 100 Dense, ReLU, 250 Dense, ReLU, 500 Dense as decoder. Here numbers specify the output of the linear layer. As loss function, first **mean absolute percentage error** (MAPE) [13] was tried just as in the article. However, there is a division by the target in this loss function and close to the end of the target histogram most values are 0. Because of division by 0, this method was unstable so this loss function was replaced by **mean absolute error** (MAE) [14]. The optimiser used was *adadelta* [15] with weight decay 0.01. This will be called default model from now on.

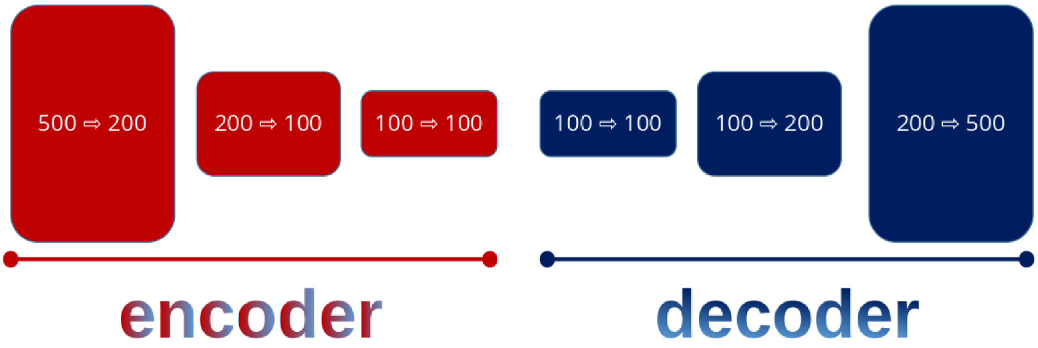

Figure 6: Default Model with Swish activation. Rounded rectangles are linear layers

The second model was an improved version of the first one keeping the model architecture the same: ReLU was replaced with swish [16]. Adadelta optimisation was replaced with AdaBelief [17] from pytorch-optimizers 2.11.1 [18]. Finally latent dimension (the dimension of the last layer of encoder or first layer of decoder) is increased to 100. This will be referred as improved model in this report.

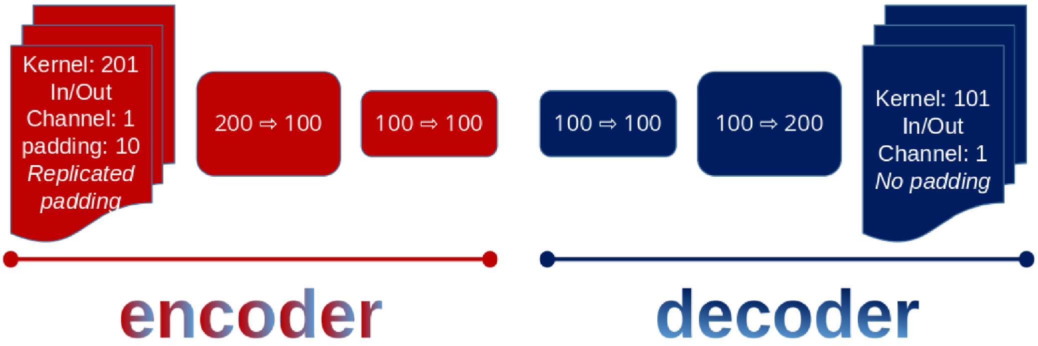

Figure 7: Default Model with Swish activation. Rounded rectangles are linear layers and the first and last shapes are convolutional layers

On top of this upgrade of algorithms to the original paper, the third model introduced convolutional neural networks (CNN) modules instead of linear layers since it can be seen that Equation 4 is a Gaussian kernel. As a matter of course, linear layers can be simplified by just enabling Gaussian and inverse Gaussian convolution kernels. In the encoder, the resulting model had a 1 dimensional CNN layer with 1 input channel, 1 output channel, 1 stride, 10 padding that replicates the edges, 1 dilation and kernel size of 201. This was followed by a swish activation, a linear layer with 100 output, a swish activation, a linear layer with 50 output, and a swish activation. The decoder part consisted of a linear layer with 100 output, a swish activation, linear layer with 600 output, a swish activation and deconvolution layer with 1 input channel, 1 output channel, 1 stride, 50 padding, 1 dilation and kernel size of 201. With this model the number of differentiable parameters in the model decreased from 250000 to 103000 with a minute sacrifice in accuracy. This model is dubbed convolutional model throughout this text.

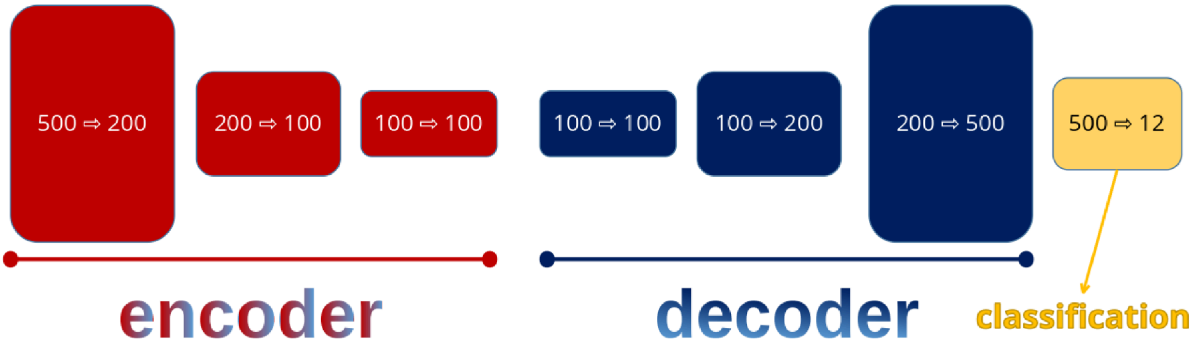

Figure 8: Default Model with Swish activation. Rounded rectangles are linear layers

The last model was a bald trial to check the feasibility of identifying sources from the detector response. At the end of the improved model, a linear layer with 12 dimensions –for each nuclide– and Sigmoid activation is added. R2 score is replaced with multilabel accuracy with micro average. This model will be named identification model. Another accuracy metric is formulated to properly evaluate the performance for the use case of this model:

$$\frac{1}{N}\sum_{i}^{N} 1\left(y_1 \Leftrightarrow y'_1 \wedge y_2 \Leftrightarrow y'_2 \wedge y_3 \Leftrightarrow y'_3 \wedge \cdots \wedge y_{12} \Leftrightarrow y'_{12}\right) \tag{5}$$

where $\vec{y}$ is the vector prediction, $\vec{y}'$ is the target vector and N is the number of samples.

This accuracy metric is unique in the sense that it is formulated to account for the intricacies of the source identification task. In essence, this metric is the probability of the model to identify all the nuclides that are present and absent in one sample correctly.

Results

The PDFs (probability density function) of nuclides from the Geant4 simulation are given below:

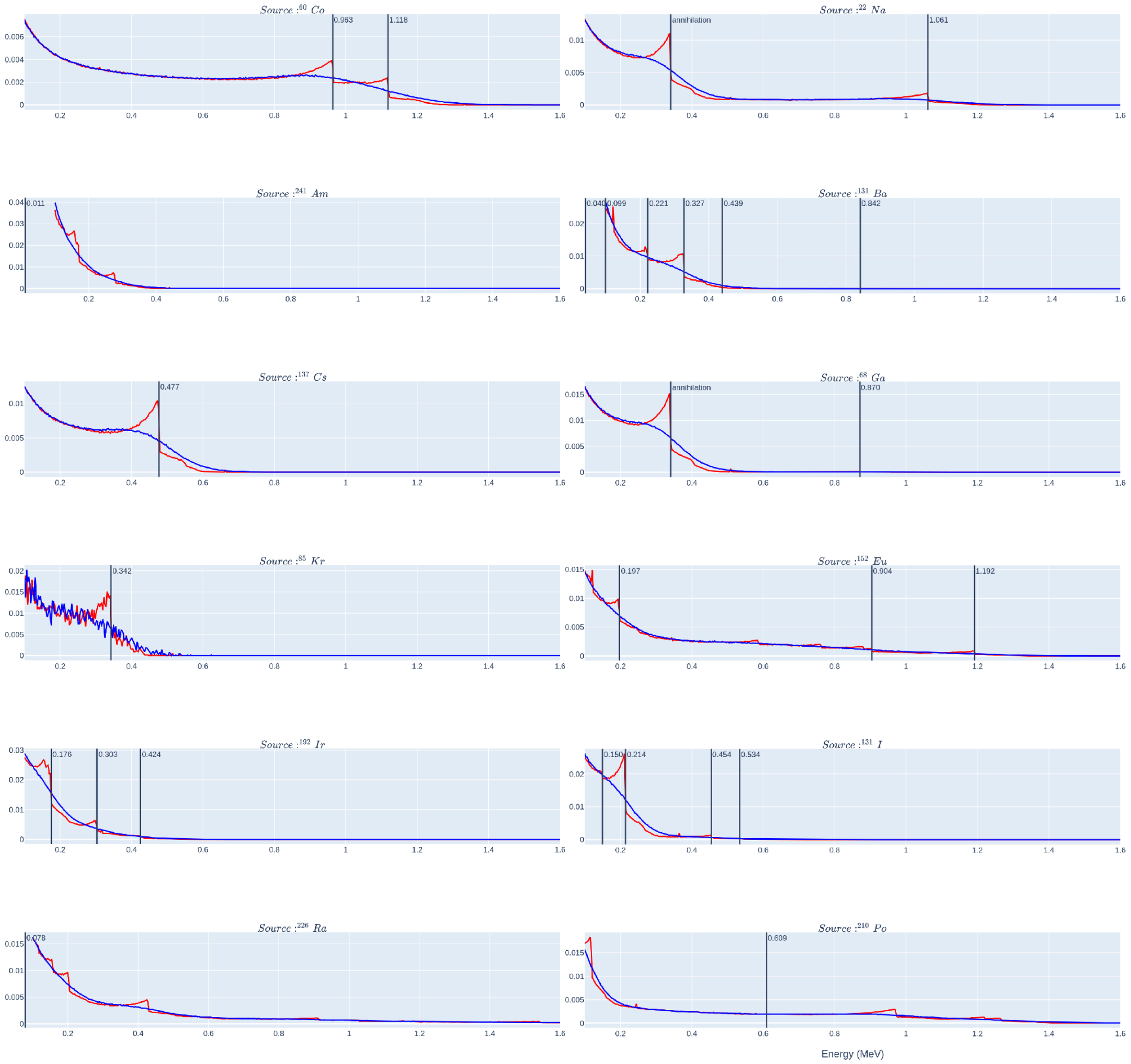

Figure 9: PDF of each chosen nuclide

If the source or one of its decay products has a B^+ decay, gammas from e^+ annihilation are also shown. Krypton has a jagged PDF graph because the branching ratio of the decay resulting in gamma ray was quite low. Since the results were not affected terribly, it was left as is. The compton edges for the decay products are deliberately left unmarked to clearly state that even when a source like ^{226} Ra emits gammas below the threshold, they can be identified because of its decay products.

Below are the training statistics of the default model:

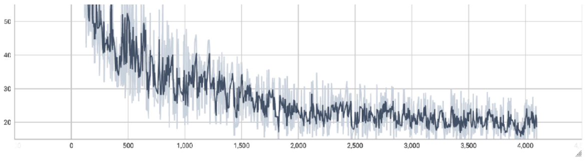

Figure 10: Training loss on each epoch for the default model

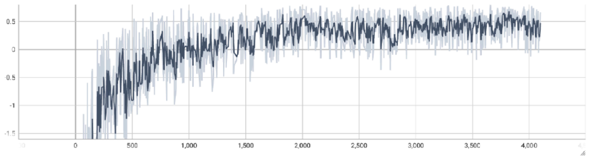

Figure 11: Training R2 score on each epoch for the default model

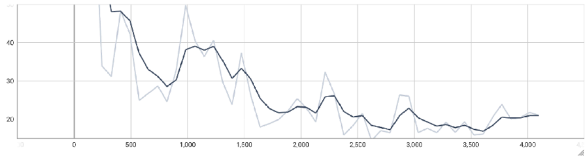

Figure 12: Validation loss on each epoch for the default model

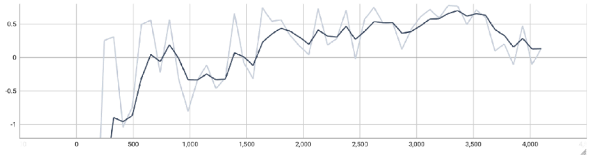

Figure 13: Validation R2 score on each epoch for the default model

Below are the training statistics of the convolutional model:

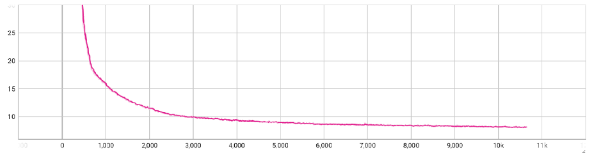

Figure 14: Training loss on each epoch for the convolutional model

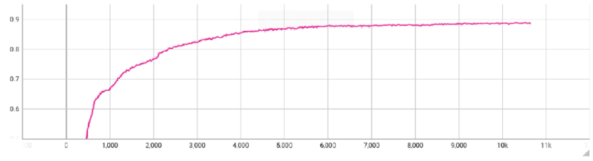

Figure 15: Training R2 score on each epoch for the convolutional model

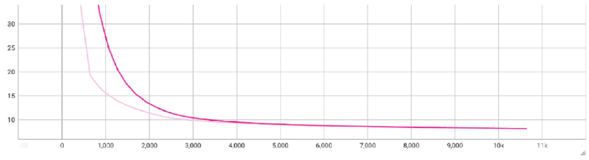

Figure 16: Validation loss on each epoch for the convolutional model

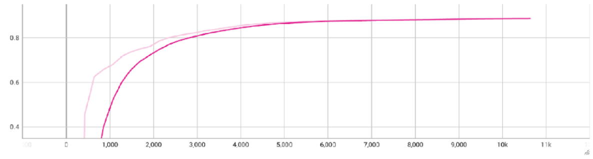

Figure 17: Validation R2 score on each epoch for the convolutional model

Convolutional model has the fastest and most stable convergence.

Below are the training statistics of the improved model:

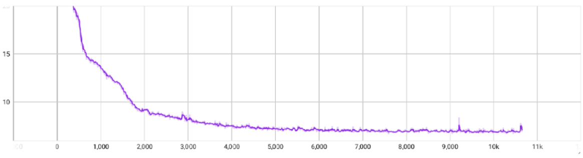

Figure 18: Training loss on each epoch for the improved model

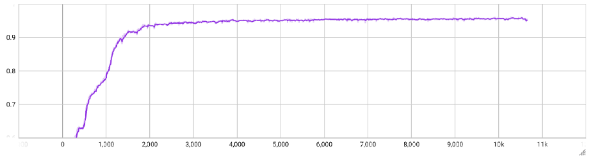

Figure 19: Training R2 score on each epoch for the improved model

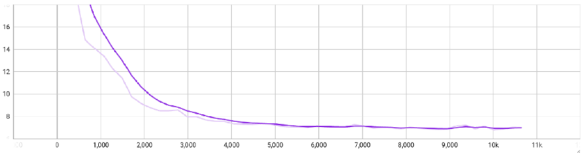

Figure 20: Validation loss on each epoch for the improved model

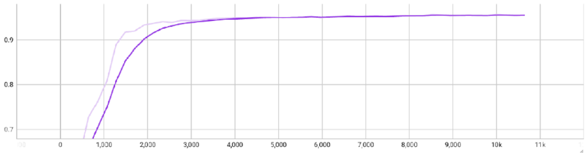

Figure 21: Validation R2 score on each epoch for the improved model

Below are the training statistics of the identification model:

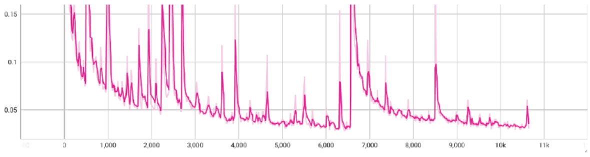

Figure 22: Training loss on each epoch for the identification model

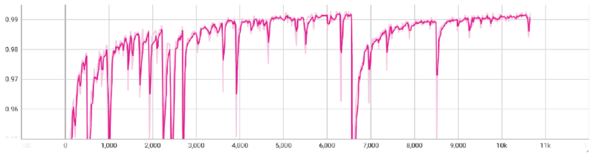

Figure 23: Training R2 score on each epoch for the identification model

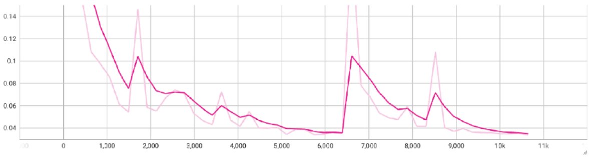

Figure 24: Validation loss on each epoch for the identification model

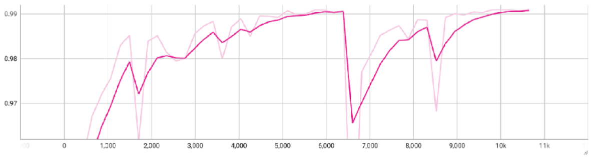

Figure 25: Validation R2 score on each epoch for the identification model

In identification model training, there are certain jumps.

Below are the training performance of each model:

Model	**CPU time** (ms)	**GPU time** (ms)
default	21.352	3.134
improved	47.679	1.227
convolutional	14.609	1.629
identification	42.020	1.006

If the thermal design power of GPU is higher than that of CPU, identification model and improved model would be the most cost-effective models due to low GPU time. The smaller convolutional model did not necessarily result in lower power consumption.

Below one can observe a few examples of practical applications of the reconstruction models:

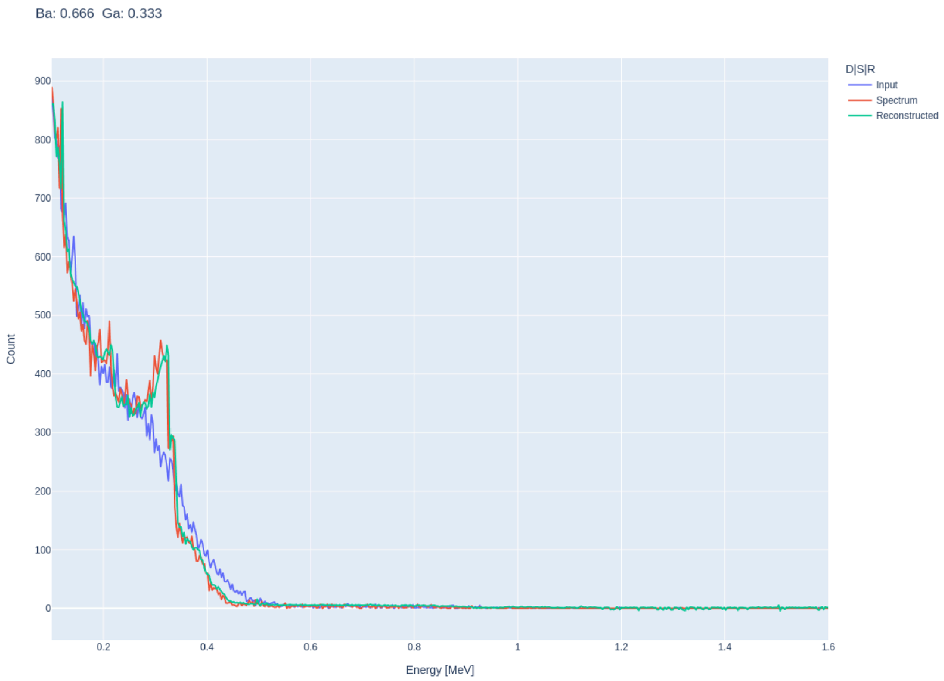

Figure 26: Reconstruction of the spectrum from the detector response or input using default model

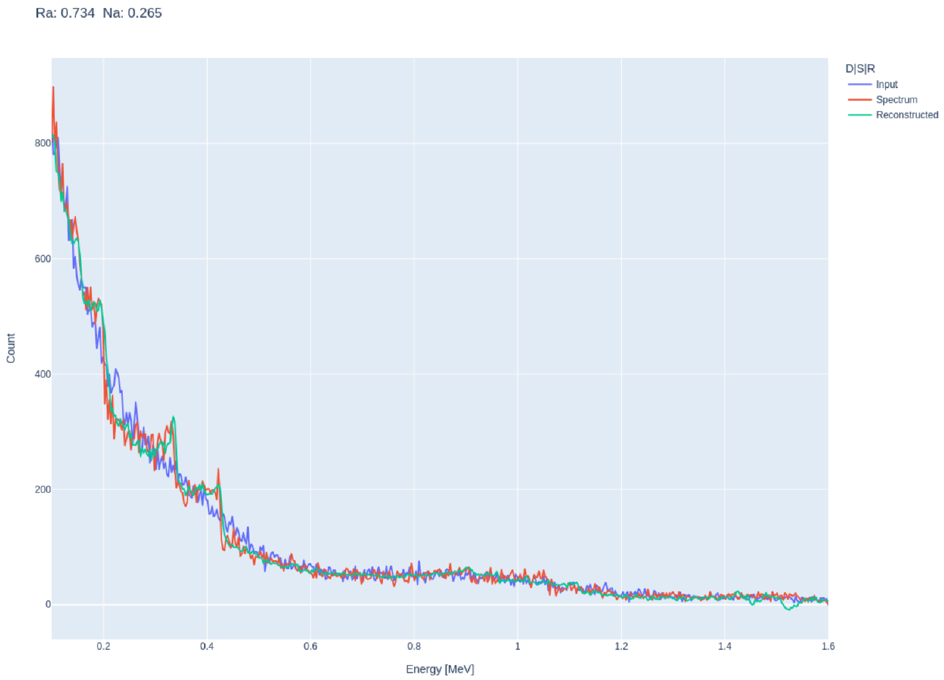

Figure 27: Reconstruction of the spectrum from the detector response or input using convolutional model

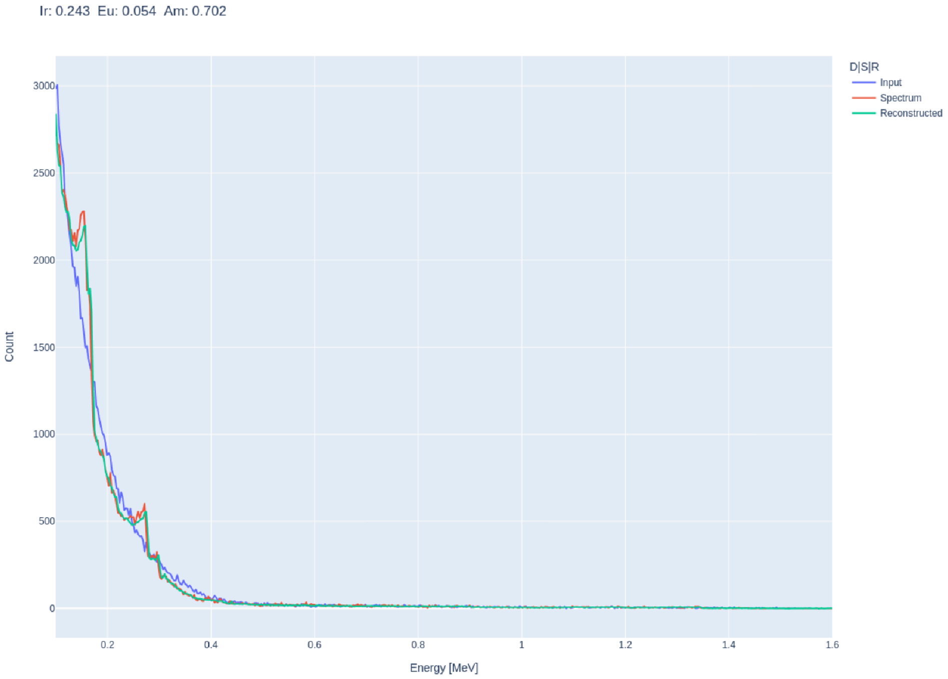

Figure 28: Reconstruction of the spectrum from the detector response or input using improved model

Below are the statistics on the test dataset:

Model	Loss	R2 Score
default	14.450	0.74742
improved	6.77516	0.95698
convolutional	8.12998	0.88706

Improved model has the best quantitative performance in spectra reconstruction.

Model	Loss	Accuracy Score
identification	0.03292	0.90570

This is the special accuracy metric from the Equation 5.

Source	Accuracy	Source	Accuracy	Source	Accuracy	Source	Accuracy
Co	0.99188	Am	0.99122	Cs	0.99590	Kr	0.99164
Ir	0.98861	Ra	0.99652	Na	0.98736	Ba	0.98558
Ga	0.98462	Eu	0.99358	I	0.992170	Po	0.99542

In average 0.99121 accuracy to identify a nuclide.

Discussion

When choosing the ratios of nuclides in a mix, no lower limit is set i.e. the ratio 0.0001 is equally likely to the ratio 0.5. Consequently, there are cases the ratio of a nuclide is so low that its effect on the detector response is not discernible from the noise. As a result, ML models fail to reconstruct or classify those samples. Especially, the custom metric for multilabel classification.

The detector response function had some placeholder parameters. Two of them were omitted and the last one was assumed 0.1. This value was 0.07 in the article *Reconstruction of Compton Edges in Plastic Gamma Spectra Using Deep Autoencoder* [11] even though the detector in this research had a larger version of the same scintillator material and a better photo multiplier. Even in the case the placeholder parameters were chosen with some tolerance, the results here were quite promising.

In the identification model training, the jumps may be due to the stochastic search of the AdaBelief algorithm.

Though this research worked with only 12 nuclide PDFs and up to 3 different nuclides in one sample, any number of nuclides are possible for both PDFs and samples. Since this research was conducted to explore the feasibility. The sources are selected to represent a vast range of scenarios. Adding new sources to the whole process is as easy as adding the atomic and mass number of the source to the macro generating and dataset building scripts.

When there is both a nuclide and its decay product are present in the same sample, this may reduce the performance or even confuse the model thoroughly.

Though Geant4 simulation has the capability to produce a very fine graph of PDF for each chosen nuclide, the sample mixes still lack the cosmic muon interaction and background gamma radiation. Muon interaction was not distinguishable in the PID graph yet its track in the detector may signal its identity. The background radiation may be hard to simulate, starting from the sources in Geant4. Instead, a mathematical approach may be used. The angular and energy distribution of cosmic muons are formulated very well [19]. These can be transcribed to Geant4 macro. Background radiation can benefit from a similar approach.

As a continuation of this research, a variety of tasks may be conducted.

- Instead of placeholders for the GEB, actual parameters a,b,and c in Equation 1 can be measured.
- The effect of muons and background radiation can be accounted for.

The following require very minor changes in the source code but mind that data wrangling takes around an hour, the simulation takes around 49 hours, and AI training for each model takes around 15 mins on the computer used in this project.

- By setting lower limit of ratio, it can be tested whether the accuracy, 0.91, for the mixes improve in realistic scenarios.
- Instead of 12 nuclides, more may be specified. Especially, nuclides combined with their decay products can be put in the same mix to assess how well the model can distinguish them.
- Performance with more than 3 nuclides in a single mix can be tested.
- The minimum number of counts required for a predetermined accuracy, say 0.90, can be measured for all models.

Conclusion

Thanks to this model, spectrometry no longer require expensive detector materials, commercial software products, and physicist's input. This project proves that cheap PVT scintillator can replace expensive detector materials such as HPGe in spectrometry tasks. This task is achieved without utilising any proprietary computer applications, eliminating the task of software calibration, a financial burden as well. Furthermore, the identification model doesn't require the expertise of a human operator to interpret the spectrum and calculate the maths of finding the nuclides. Notwithstanding, there are still some tests that must be conducted before being production ready.

Bibliography

[1] S. Agostinelli *et al.*, “Geant4—a simulation toolkit”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, no. 3, pp. 250–303, 2003, doi: https://doi.org/10.1016/S0168-9002(03)01368-8.

[2] J. Allison *et al.*, “Recent developments in Geant4”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, pp. 186–225, 2016, doi: https://doi.org/10.1016/j.nima.2016.06.125.

[3] J. Allison *et al.*, “Geant4 developments and applications”, *IEEE Transactions on Nuclear Science*, no. 1, pp. 270–278, 2006, doi: 10.1109/TNS.2006.869826.

[4] R. Brun and F. Rademakers, “ROOT — An object oriented data analysis framework”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, no. 1, pp. 81–86, 1997, doi: https://doi.org/10.1016/S0168-9002(97)00048-X.

[5] B. Jeon, J. Kim, M. Moon, and G. Cho, “Parametric optimization for energy calibration and gamma response function of plastic scintillation detectors using a genetic algorithm”, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, pp. 8–14, 2019, doi: https://doi.org/10.1016/j.nima.2019.03.003.

[6] J. Pivarski *et al.*, “Uproot”. [Online]. Available: https://doi.org/10.5281/zenodo.8122179

[7] C. R. Harris *et al.*, “Array programming with NumPy”, *Nature*, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.

[8] W. Falcon and The PyTorch Lightning team, “PyTorch Lightning”. [Online]. Available: https://github.com/Lightning-AI/lightning

[9] A. Paszke *et al.*, “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[10] T. Developers, “TensorFlow”. [Online]. Available: https://doi.org/10.5281/zenodo.8118033

[11] B. Jeon, Y. Lee, M. Moon, J. Kim, and G. Cho, “Reconstruction of Compton Edges in Plastic Gamma Spectra Using Deep Autoencoder”, *Sensors*, no. 10, 2020, doi: 10.3390/s20102895.

[12] A. F. Agarap, “Deep Learning using Rectified Linear Units (ReLU)”. 2019.

[13] A. de Myttenaere, B. Golden, B. L. Grand, and F. Rossi, “Mean Absolute Percentage Error for regression models”, *Neurocomputing*, pp. 38–48, Jun. 2016, doi: 10.1016/j.neucom.2015.12.114.

[14] C. Willmott and K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance”, *Climate Research*, pp. 79–82, 2005, doi: 10.3354/cr030079.

[15] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method”. 2012.

[16] P. Ramachandran, B. Zoph, and Q. V. Le, “Swish: a Self-Gated Activation Function”, *arXiv: Neural and Evolutionary Computing*, 2017, Available: https://api.semanticscholar.org/CorpusID:196158220

[17] J. Zhuang *et al.*, “AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients”. 2020.

[18] H. Kim, “pytorch_optimizer: optimizer & lr scheduler & loss function collections in PyTorch”. [Online]. Available: https://github.com/kozistr/pytorch_optimizer

[19] P. Shukla and S. Sankrith, “Energy and angular distributions of atmospheric muons at the Earth”. 2018.