UC San Diego

DSC291 Final Project: MJD Data Release Analysis

Akbota Assan[1] (A69037121), Devana Perupurayil[1] (A69034326), Melissa Medina-Peregrina[2] (A59016508)
[1]*Halicioğlu Data Science Institute at UC San Diego.*
[2]*Department of Physics at UC San Diego.*

Introduction

Double beta decay (2νββ), was first proposed by Maria Goeppert-Mayer in 1935. 2νββ decay occurs in a nucleus with mass number A and proton number Z when single beta decay is energetically forbidden and decaying to a Z+2 proton number daughter nucleus is possible [1]. Neutrinoless double beta decay ($0\nu\beta\beta\ or\ NLDBDB$) is a theoretical rare nuclear process whereby a nucleus decays into another one and emits two electrons and no antineutrinos. For each nuclear decay, only a small amount of energy is released and available for detection. The goal is to protect this small signal from the background [2].

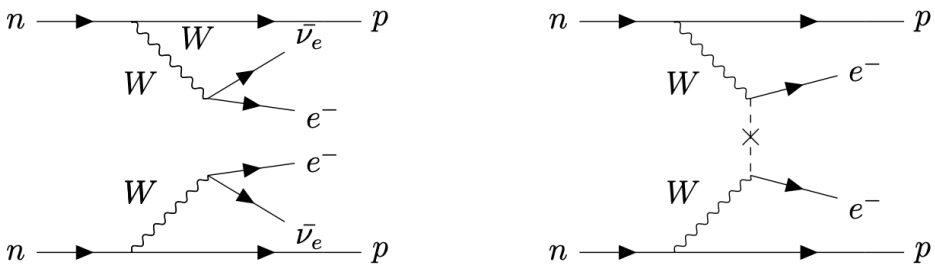

Fig. 1: Feynman diagrams of the two double beta decay modes. In these processes two neutrons decay into two protons, two electrons and either two neutrinos as shown in Figure a) or no neutrinos as shown in Figure b).

The Majorana Demonstrator experiment searched for $0\nu\beta\beta$ of ^{76}Ge using modular arrays of 56 high-purity Ge detectors. The MJD dataset represents a portion of its detector waveforms with corresponding analysis labels released to support the training and test of AI/ML algorithms to get an unambiguous identification of $0\nu\beta\beta$ events.

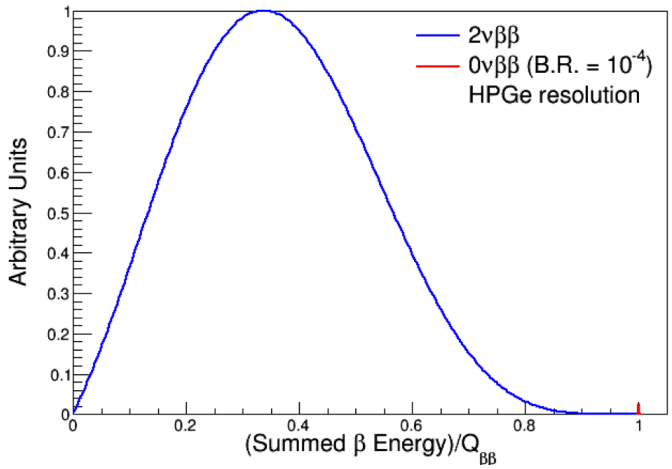

Fig. 2: The curves represent the electron sum energy spectrum of the two double beta decay modes.

Current experimental results indicate that to achieve sensitivity to such a rare decay, any experiment aiming to find 0νββ must have a very low background rate since our signal is very small, as shown in figure 2.

Methodology

We analyzed the MJD dataset to extract relevant information from the energy spectrums, evaluate classification performance, and fit the unknown parameter θ_c for Detector C in the following way:

1. Our first step consisted in analyzing the distribution of detected events in terms of their energy levels. Using the NumPy and Pandas libraries, we loaded the four CSV files (DetectorA.csv, DetectorB.csv, DetectorC.csv, DetectorTarget.csv) that conforms the data set into a structured data format. To visualize the energy spectrum of each detector, we used the associated energy value of each event.

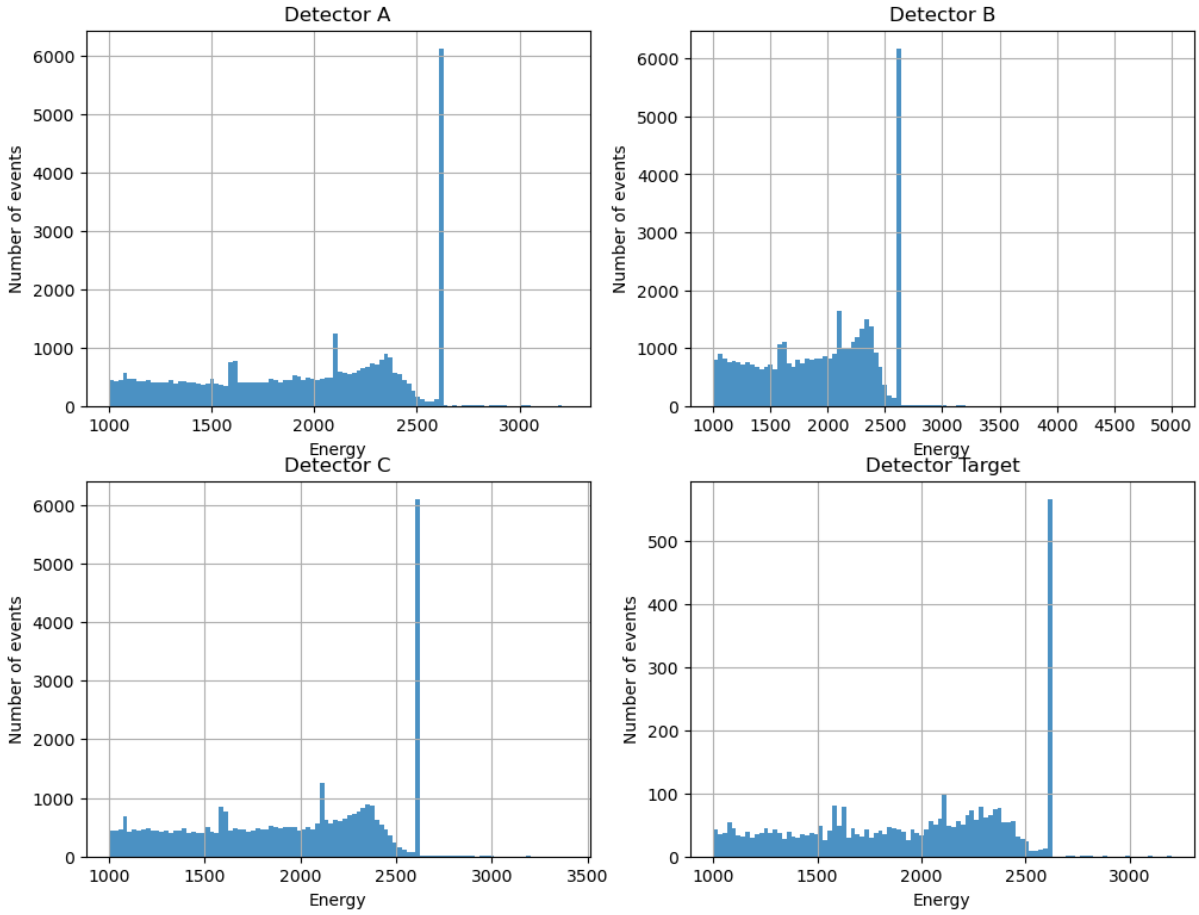

Fig.3: Energy spectrum of each detector.

The binning is determined by the energy range for each detector. We find the minimum and maximum energy values in each detector's dataset and we then generate four histograms with 100 bins each to represent the distribution of energy values for each detector as shown in figure 3.

2. We computed the true positive rate to quantify how effectively the cut retains signal events by setting a classification score threshold based on the 1592 keV peak in detector A to maximize the signal events. We deduced from [2] that the energy range around the 1592 keV peak should be ± 3 keV, which means that our algorithm only selects events in Detector A whose energy values fall within the range of 1589 keV to 1595 keV. These events represent the subset of data around the expected signal peak. We set the threshold at 20% of cnn_score values within the selected energy range. This means that approximately half of the events around the peak in detector A will be classified as signal. We found how well the threshold retains signal events through the True Positive Rate (TPR):

$$TPR = \frac{TP}{TP + FN}$$

Where TP is the amount of True Positive counts (signal events correctly retained with cnn_score greater than or equal to the threshold) and FN are the False Negatives (events with cnn_score below the threshold being rejected by mistake). After calculations, our TPR = 0.801.

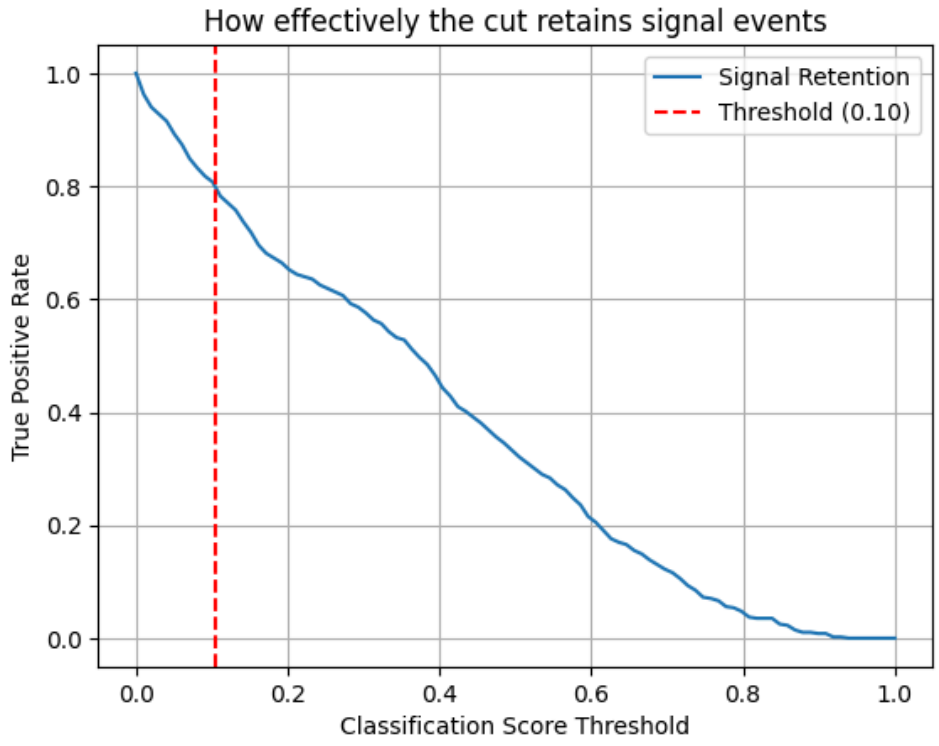

Fig.4: Plot of the signal retention as a function of threshold to get an estimate of how effectively the cut retains signal events.

We iterated over different cnn_score thresholds to generate a signal retention curve. For each threshold value, the proportion of retained signal events (cnn_score ≥ t) is computed and plotted as shown in figure 4. This plot quantifies how many signal events are retained when applying this threshold and then we get information on how effectively the cut retains these signal events.

3. For this step, our goal was to evaluate how well our classification model separates signal from background using the Receiver Operating Characteristic (ROC) curve and calculating the Area Under the Curve (AUC). We also computed the False Positive Rate (FPR):

$$FPR = \frac{FP}{FP + TN}$$

Where FP is the amount of False Positive counts (background events with cnn_score greater than or equal to the threshold being retained by mistake) and TN are the True Negatives (events with cnn_score below the threshold). This quantifies how many background events there are after the cut based on the 2103 keV peak in Detector B, which represents background events. After calculations, our FPR = 0.190. We then compare this value to the signal events from Detector A to evaluate the impact of the classification score threshold applied previously, which is represented in figure 5.

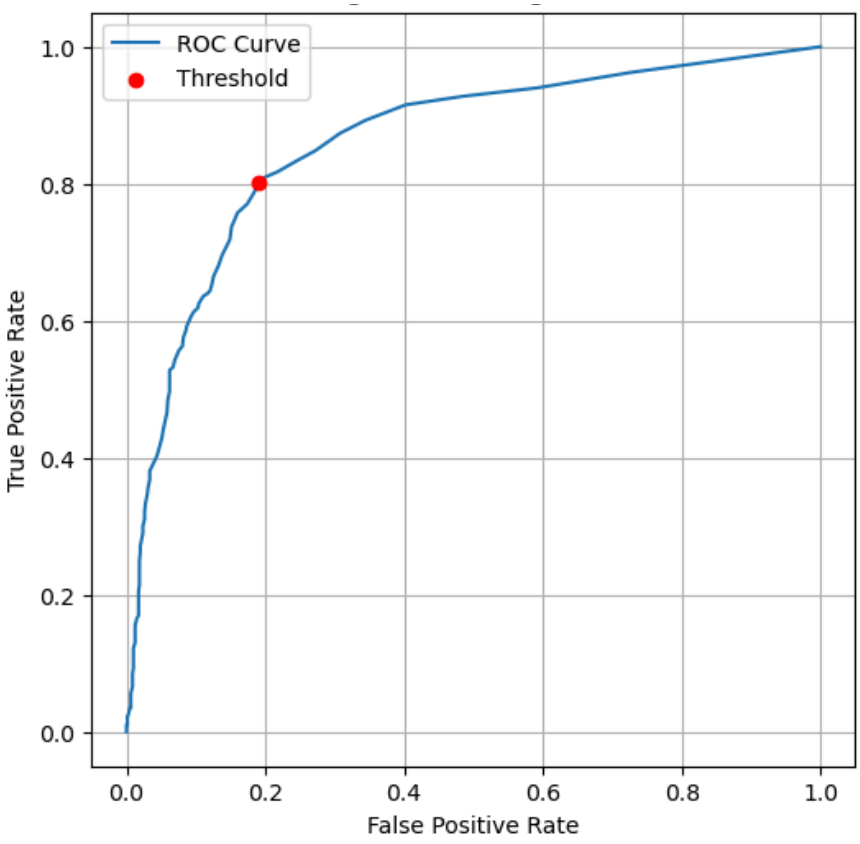

Fig.5: Plot of the signal retention as a function of threshold to get an estimate of how effectively the cut retains signal events.

When we calculated the AUC, our result was 0.86, which is an acceptable value since if compared to perfect classification (AUC = 1) we get that our model performs a good classification.

4. Using the SciPy library, we built a probability density function which is the expected signal for $0\nu\beta\beta$ decay. This was modeled as a Gaussian distribution { N(2039,1) } with width $\sigma = 1\ keV$ and a peak at $\mu = 2039$ keV:

$$f(E) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(E-2039)^2}{2}}$$

As observed in figure 6, we created an array of 100 energy values uniformly spaced from 2036 keV to 2042 keV (equivalent to $\mu \pm 3\ keV$ [2]) to define the energy range over which the Gaussian function will be evaluated.

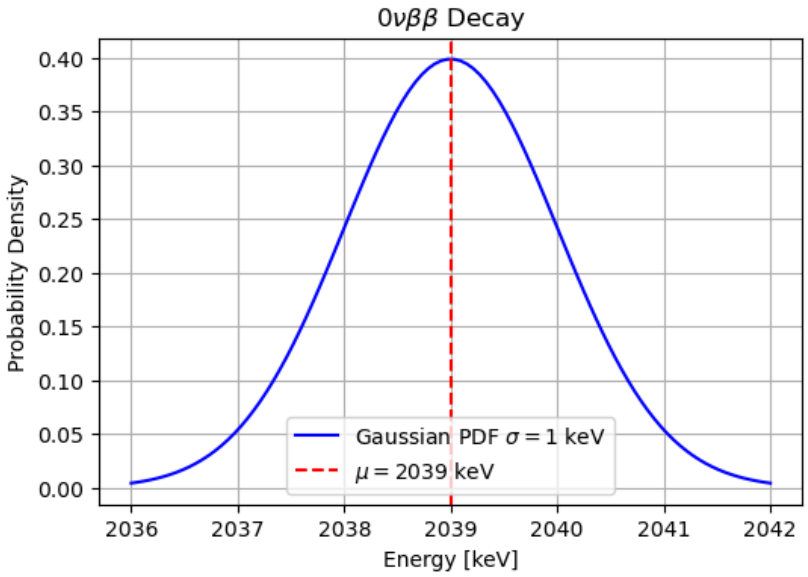

Fig.6: Visualization of the expected energy distribution of signal events for $0\nu\beta\beta$.The dashed line shows the mean value for a $0\nu\beta\beta$ signal, i.e. 2039 keV. The blue plot shows the probability density for detecting an event $\pm$ 3 keV around the mean.

5. In our analysis, we applied an energy cut to neutrinoless double beta decay data to distinguish signal from noise. Our algorithm selects only the events where cnn_score is greater than or equal to our threshold for each detector to keep the ones classified with a higher probability to be a true signal. The plots representing these cuts are shown in figure 7. This shows how the filtering affects the number of events at different energy levels.

We modeled the energy spectrum of the surviving events as a Gaussian distribution. For each detector, we obtained the mean and standard deviation from the filtered dataset and then we created a Gaussian distribution centered around these mean values. In a range of ±3σ we generated 100 points and found the probability density.

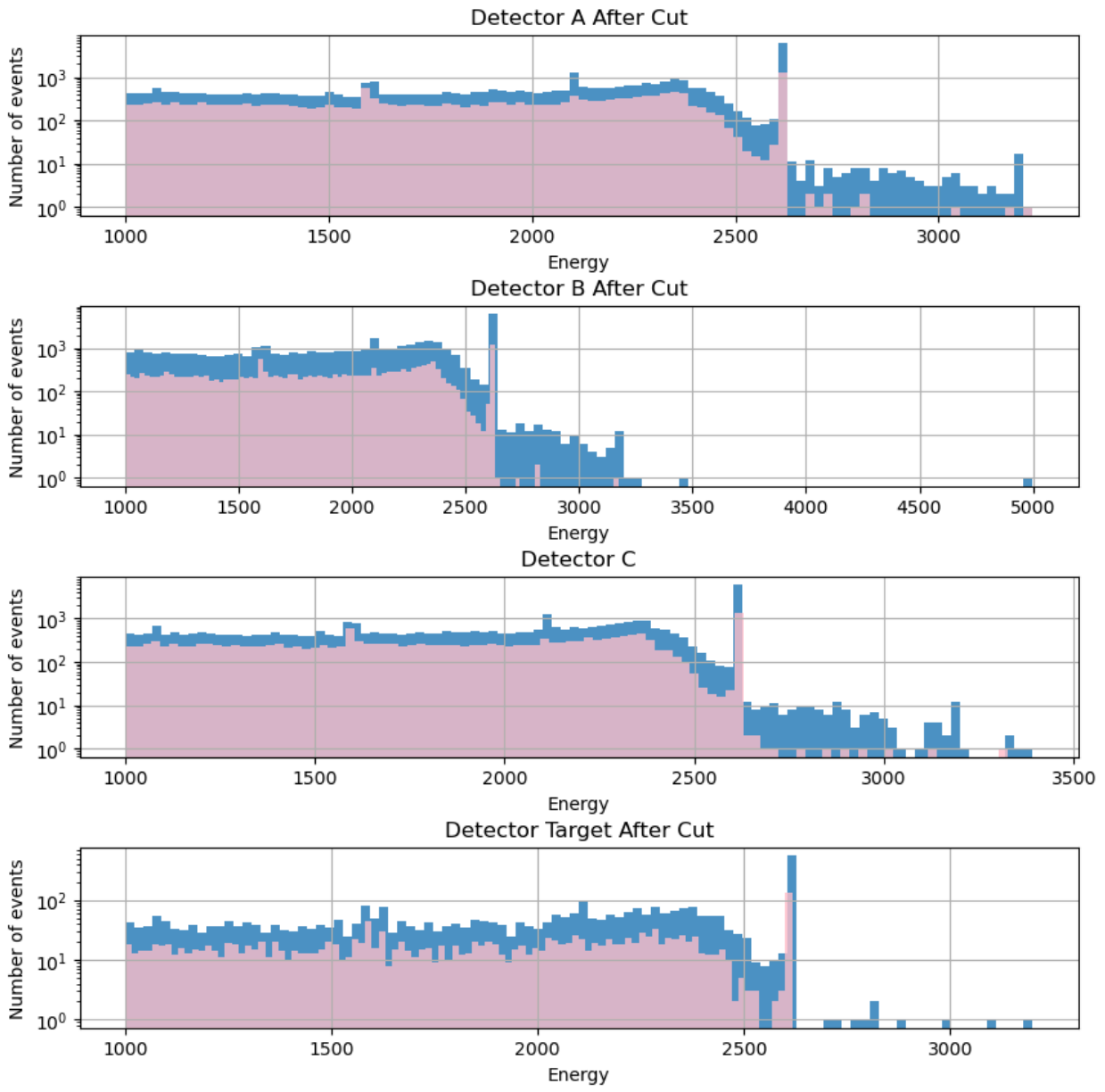

Fig.7: Energy spectrums before and after applying the threshold. The full energy spectrum is shown in blue, and the after-cut energy distribution is shown in pink.

This helps in understanding the efficiency of the classification cut and how it impacts the energy spectrum of each detector.

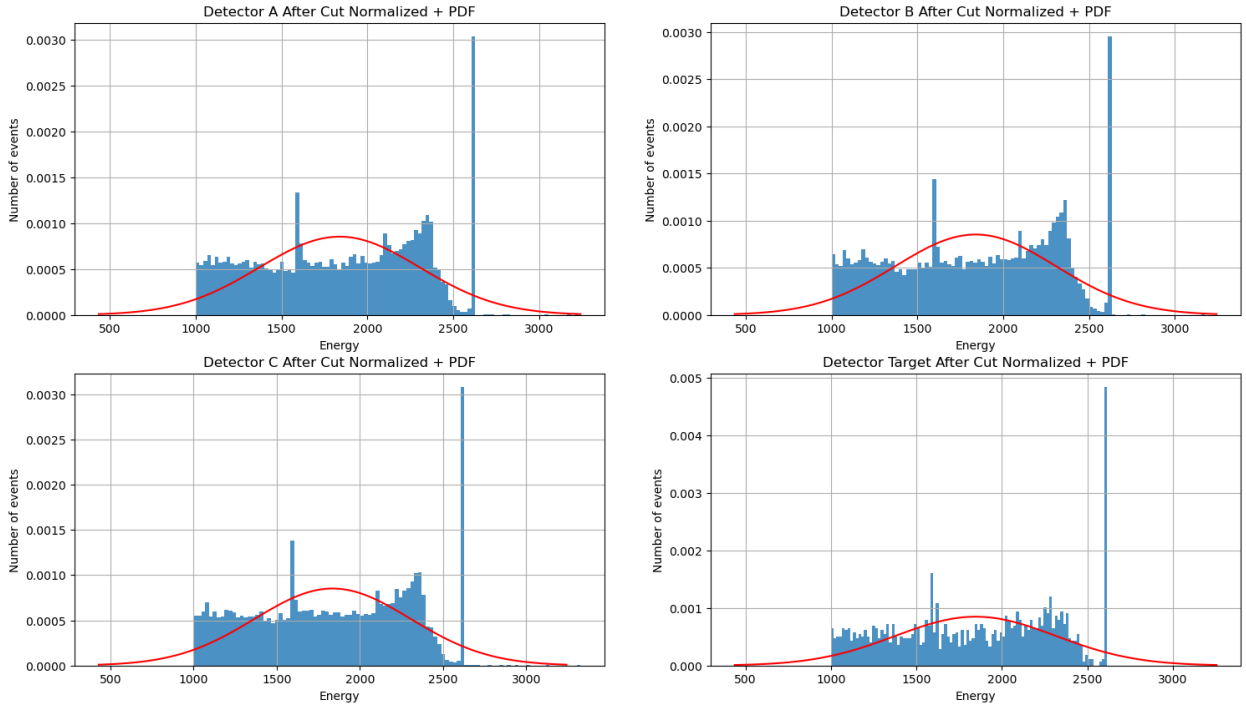

Fig.8: The Gaussian fit (red curve) overlaying the histograms of the filtered datasets (blue data).

6. The Bayesian approach was applied to estimate the parameters θ_A, θ_B, θ_C, and θ_{NLDBD} using the given detector data. The goal was to fit the observed energy distribution using a model constructed from individual detector signals and a potential signal from neutrinoless double-beta decay (NLDBD). The total model was defined as:

$$\mu_i = \theta_A PDF_i(A) + \theta_B PDF_i(B) + \theta_C PDF_i(C) + \theta_{NLDBD} N_i(2039, 1)$$

Where PDF(A), PDF(B), and PDF(C) are the probability density functions of energies for each detector, and $N_i(2039, 1)$ is the Gaussian PDF representing the NLDBD signal.

The Bayesian analysis requires defining prior distributions to encode any prior knowledge about the parameters. The following priors were used:

- $\theta_A \sim N_i(1350, 100^2)$, as this data was provided to us.
- $\theta_B \sim N_i(770, 270^2)$, as this data was also provided to us.
- $\theta_C \sim N_i(1836.19, 468.7^2)$. To be able to use this prior, we used the statistics package in Python to find the best fitting normal distribution for a given sample.
- $\theta_{NLDBD} \sim$ Uniform(0, 100), as we have no prior information.

One thing that should be noted is that the Gaussian distributions were normalized as they might not be properly scaled.

The likelihood function was modelled using a Poisson distribution to describe the observed energy counts. The log-likelihood is defined as:

$$Log\ L(n,\theta) = \sum_i n_i(logu_i) - u_i - n_i logn_i + n_i$$

where n_i represents the observed counts in each energy bin. The posterior distribution was calculated using the Bayes' theorem:

$$P(\theta|n) = P(n|\theta)\frac{P(\theta)}{P(n)}$$

The log posterior was implemented as the summation of the log prior and log likelihood.

The Markov Chain Monte Carlo Sampling (MCMC) method using the emcee library was employed to sample from the posterior distribution. The chain was initialized with 50 walkers and a run of 10000 steps. A burn-in period of 1000 steps was applied to remove initial bias. After the sampling process, the values we got were as follows:

θ_A = 1275.39 ± 98.98
θ_B = 226.70 ± 233.79
θ_C = 209.85 ± 246.97
θ_{NLDBD} = 3.46 ± 3.47

Since our value for TPR = 0.801, then the number of NLDBDs is obtained from:

$$\frac{\theta_{NLDBD}}{TPR} = 4.319$$

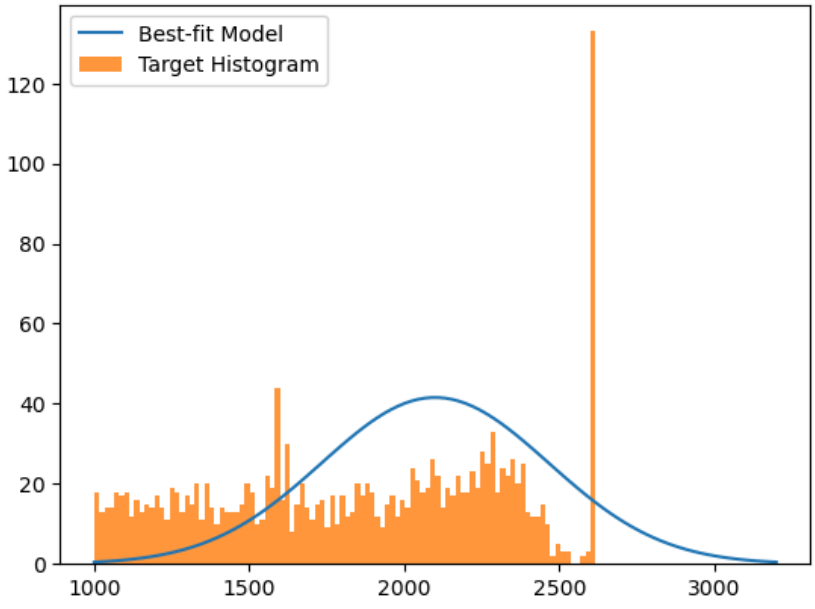

Fig.9: Representation of the best-fit model over the observed energy spectrum. The best-fit parameters θ_A ,θ_B , θ_C and θ_{NLDBD} are shown above.

The Bayesian method is chosen for this analysis because:

- Bayesian inference allows us to integrate prior information from Detectors A and B into the fitting process, making use of previously known calibration data.
- Bayesian inference provides full posterior distributions, allowing for a more comprehensive assessment of uncertainties through credible intervals.
- Bayesian methods use Markov Chain Monte Carlo (MCMC) to explore the parameter space efficiently, even in cases where the likelihood function is complex or non-Gaussian.

7. To determine the 90% confidence level upper limit for θ_{NLDBD}, the MCMC samples from the posterior distribution are analyzed. The 4th parameter in the sampling results corresponding to θ_{NLDBD}, represents the potential signal count of NLDBD. The upper limit was computed using the 90th percentile of the θ_{NLDBD} posterior samples. The percentile function is a straightforward and effective method for deriving credible intervals in a Bayesian context.

   The upper limit of 7.97 for θ_{NLDBD} at a 90% confidence level (CL) means that, based on the data and the Bayesian analysis, there is a 90% probability that the true value of θ_{NLDBD} lies below 7.97.

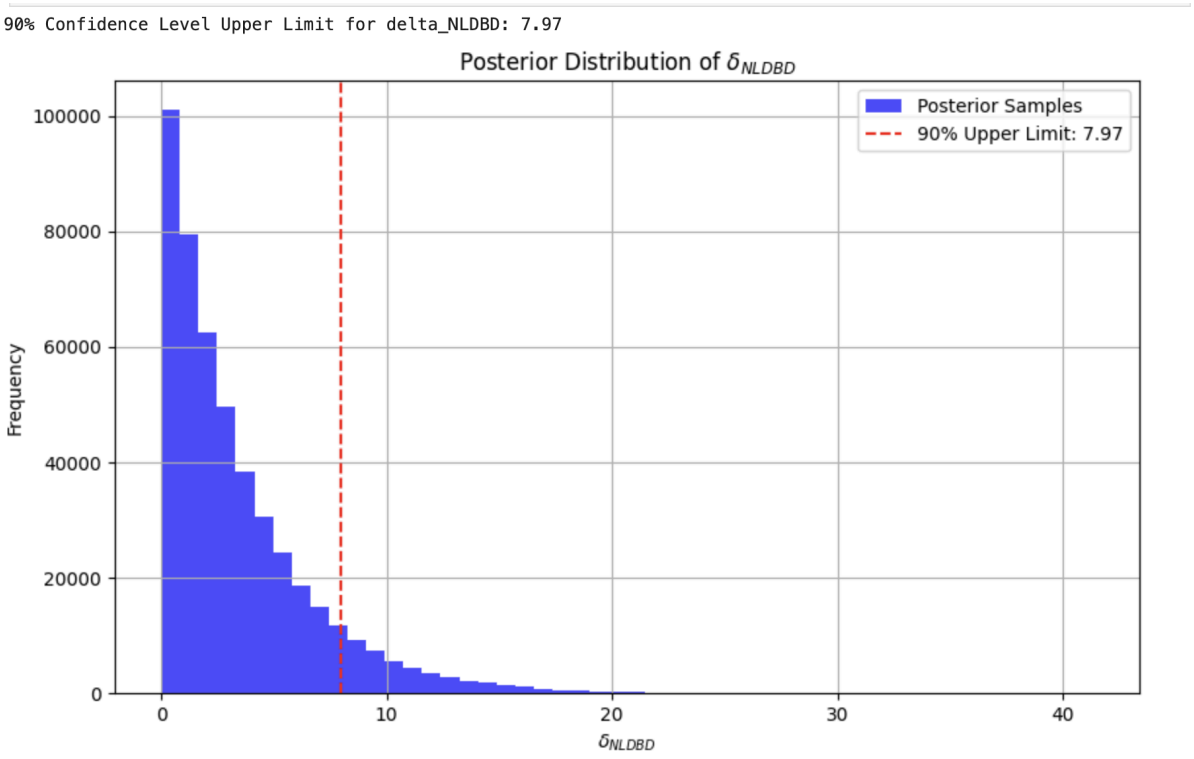

Fig.10: Representation of the posterior distribution of θ_{NLDBD}

8. Sensitivity refers to our analysis's ability to detect or constrain a potential signal, even if one isn't actually observed. To estimate this, we used what's called an Asimov dataset—a smooth, idealized version of our data generated using the best-fit background parameters, with the signal strength set to zero. This represents a scenario where the signal doesn't exist, and we want to see how much signal our model would still allow before confidently saying "there's something there." We ran MCMC on this Asimov data and looked at the posterior distribution of the signal strength parameter (`delta_NLDBD`). The 90% credible interval we got is 7.97, and the median from the MCMC fit on Asimov toy data is 2.46. This means we observed an over fluctuation—basically, the result looks stronger than what we'd expect if there were no NLDBD events, according to the toy Asimov dataset. But that makes sense, since some NLDBD events were injected into the data.

GitHub:

https://github.com/melissamp03/DSC291_FINAL-PROJECT/

Presentation slides:

DSC291_Final_Project_Presentation

References

[1] Medina-Peregrina, M. (2020). Development of an in-Xe-gas Laser Ablation Source for the Ba-tagging technique for nEXO [MSc dissertation, McGill University]. Experimental Particle Physics at McGill University Resources. https://www.physics.mcgill.ca/xhep/en/resources/thesis/2020_Medina_MSc_nEXO_Laser.pdf

[2] Arnquist, I. J. et al (MAJORANA Collaboration)(2023). *Majorana demonstrator data release for AI/ML Applications*. arXiv preprint arXiv:[2308.10856]

[3] *The MAJORANA DEMONSTRATOR*. The MAJORANA Neutrinoless Double-beta Decay Experiment | MAJORANA. (n.d.). https://www.npl.washington.edu/majorana/majorana-experiment