

Resonance Capture Widths for the Bayesian Resonance Reclassifier

I. Snider

Department of Physics, Truman State University, Kirksville, MO 63501, USA

G. P. A. Nobre

National Nuclear Data Center, Brookhaven National Laboratory, Upton, NY 11973-5000, USA

(Dated: August 11, 2023)

An accurate description of the neutron interaction cross sections for the most abundant isotopes of lead are essential for application in practical nuclear systems. These nuclear reaction cross sections demonstrate resonance phenomena in the presence of low-energy neutrons. The cross section shape is obtained by classifying the associated angular-momentum quantum numbers for each resonance. These classifications, however, are often subjective and not fully reproducible. This leads to incorrect assignments. In this project, we attempt to rectify assignments for ^{206}Pb by using a machine learning (ML) approach. ML is a process that attempts to learn patterns and make predictions based on the given training data and set of features. Quality ML training requires abundant and diverse training data. The real data is often incomplete and contains errors, so instead, we build synthetic data that mimics the statistical properties of real resonances. For neutron capture reactions, we have found that a realistic distribution of resonance decay widths contributes substantially to the success of the ML algorithm. To accomplish this, we provide our synthetic data with capture widths sampled from a Porter-Thomas distribution, where the number of degrees of freedom ν is fit from the real data.

I. INTRODUCTION

A. Neutron resonances

A neutron resonance reaction occurs when a target nucleus absorbs an incident neutron forming a *compound* nucleus. The compound nucleus then decays by emitting a gamma ray (capture channel) or by emitting a neutron (elastic channel); there are other types of channels as well, such as inelastic, but for this project we will focus on capture cross sections. An important metric for a neutron capture experiment is the capture cross section. A cross section can be thought of as the effective area of a target as seen by a projectile. This is useful for describing the probability of interaction between a given nucleus and free neutrons. This makes an accurate understanding of an isotope's neutron capture cross section of paramount importance for designing nuclear fission and fusion reactors, as well as nuclear security and nonproliferation. Neutron reactions with incident neutron energies from 10^{-5} eV to 20 MeV exhibit resonance phenomena, this complicates describing the cross section. Instead of observing smooth sloping changes in the cross section, we observe sudden and frequent resonance peaks. These “neutron resonances” are observed when an excitation state of a compound nucleus approaches the energy of the incident neutron. In this way, the cross section resonances map out the excited states of the compound nucleus. For only the lightest nuclei, the cross section, including the resonances, can be found deterministically. For a heavier nucleus, we are faced with a fermionic many-body system, so statistical methods are needed to fill the gaps.

A given resonance will have associated J , L , and S quantum numbers, referring to the total angular momen-

tum, incident neutron orbital angular momentum, and spin respectively, where S can of course be determined by knowing both L and J . Together L and J constitute a “spingroup.” A Random Matrix Theory (RMT) method uses the spingroups of each resonance to create a Gaussian Orthogonal Ensemble (GOE) Hamiltonian to fit a cross section of the experimental data. From this cross section, we obtain the resonance parameters needed for a nuclear evaluation. A complete characterization of resonance parameters will include the spingroup, incident neutron energy, spacings between resonances, and the decay widths for each of the observed reaction channels. This process is aptly called “shape analysis.” An example is shown in fig. 1. The resonance peaks shown in the figure, represent different types of waves: s-waves, p-waves, etc., corresponding to angular momenta values of $L = 0, 1$, etc., respectively.

B. Errors in the data

There are many issues associated with shape analysis. The first consideration is the experimental method. If there are resonances with too small of decay widths, then they may not be detected; this is a detector resolution issue. In fig. 1, I have illustrated resolution issues with two cross sections of ^{206}Pb . The first, on the left, is a cross section collected in 2007 [1], and the second, on the right, is a cross section collected in 2003 [2] and is the most current ^{206}Pb evaluation contained in ENDF/B-VIII.0. The cross section on the left, and indeed, many high resolution cross section measurements were taken at the GELINA experiment in Geel, Belgium. In the 2003 ENDF data, there are clearly missing resonances. This creates a problem when extracting in-

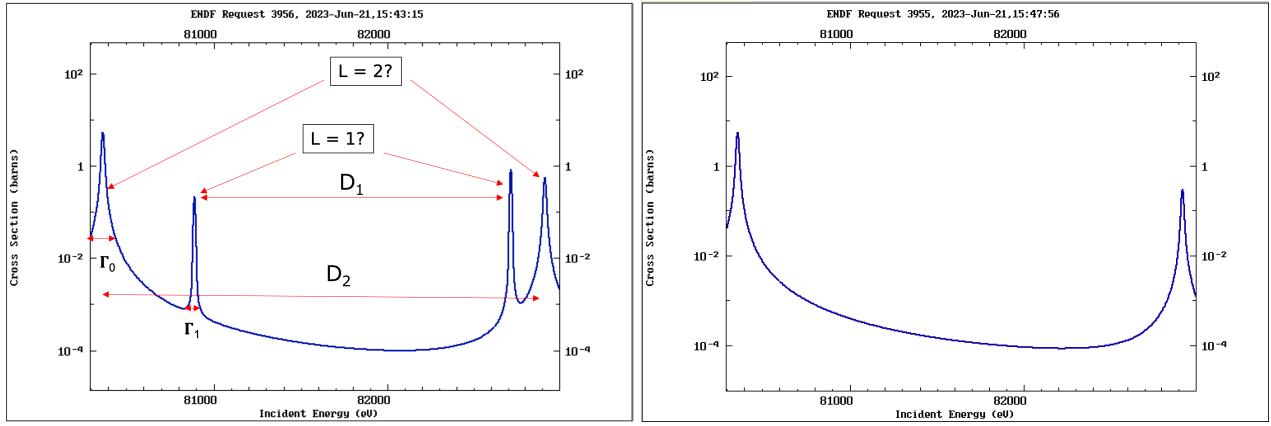


FIG. 1: Cross section resonances fit from a neutron TOF spectroscopy with ^{206}Pb as the target nucleus. D and Γ are the resonance spacing and capture width respectively. Shown left is the 2007 EXFOR entry from Borella [1]. Shown right is the ENDF/B-VIII.0 entry (taken using data from 2003) [2]. There are clearly missing resonances in the 2003 cross section.

teraction probabilities from cross section data since the method used, an RMT method, relies on accurate experimental descriptions of resonance spingroups to find spacings between resonances of like spingroup and their capture decay width distributions. Missing resonances will skew these distributions and thus create an imprecise probability of interaction for the isotope. So, increasing the resolution of the neutron time-of-flight (TOF) spectroscopy experiment is valuable for an evaluation.

The other main issue with shape analysis is much less systematic. Quite simply, shape analysis is a manual and labor intense process. This means a resonance classification is subjective and multiple evaluators may arrive at different conclusions. Evaluated Nuclear Data Files (ENDF), typically contain hundreds or even thousands of neutron resonances. Because of this, many resonances will be assigned the mean capture width for its decided spingroup. This occurs when the width is difficult to measure or as a method of saving time.

We can very quickly determine that errors are present by graphing a cumulative level distribution (CLD) of the resonance spacings. A recent evaluation of ^{206}Pb conducted at the Naval Nuclear Laboratory (NNL) [3], is a good example of this (see fig. 2). The CLD is known to be linear in the low-energy regime [4], as the energy range of the spacing distribution is small compared to the neutron separation energy. For large energy, the slope of the CLD will decrease as the capture widths enlarge and the spacings shrink. But for our energy region, we can assume the slope to be linear with small deviations according to RMT.

The CLD for spingroup $(2, \frac{3}{2})$ demonstrates what we would probably hope to obtain from a good shape analysis, as the graph is approximately linear with only small deviations as expected. The rest of the CLDs are not so flattering for shape analysis. We clearly see large deviations with deficits in one spingroup being filled by another. This is clear evidence of misclassifications. The CLD for spingroup $(3, \frac{5}{2})$ is not numerically well-founded

since there are only 3 resonances with that spingroup in the dataset. Currently, such kinds spingroups, are difficult for our machine learning model to find, so we exclude them from the training data.

II. MACHINE LEARNING & BRR

A. Machine Learning

The abundance of errors found in ENDF files has plagued the nuclear data community for decades. So, we have attempted to build a machine learning model to aid nuclear data evaluations and correct errors in resonance classifications. The first inspiration for such a project came from S.F. Mughabghab's *Atlas of Neutron Resonances* [5]. Mughabghab's book is a compilation of evaluated nuclear data for many isotopes found in the National Nuclear Data Center's (NNDC) repository of ENDF files. Unfortunately, the Atlas was found to be filled with errors, including gaps in data, typos, bibliographic insanity, and resonances that were simply misclassified [6]. Because the Atlas is a valuable resource for research at the NNDC, we saw this as an opportunity to develop a new approach.

Machine learning (ML) is a metaheuristic computing process that attempts to learn patterns from a data set in order to make predictions. At the NNDC, we devised a machine learning method to identify and reclassify misassigned neutron resonances: the Bayesian Resonance Reclassifier (BRR) [6]. BRR uses a Python ML library called “scikit-learn.” scikit-learn is a collection of ML classifiers that offer various methods for learning patterns [7].

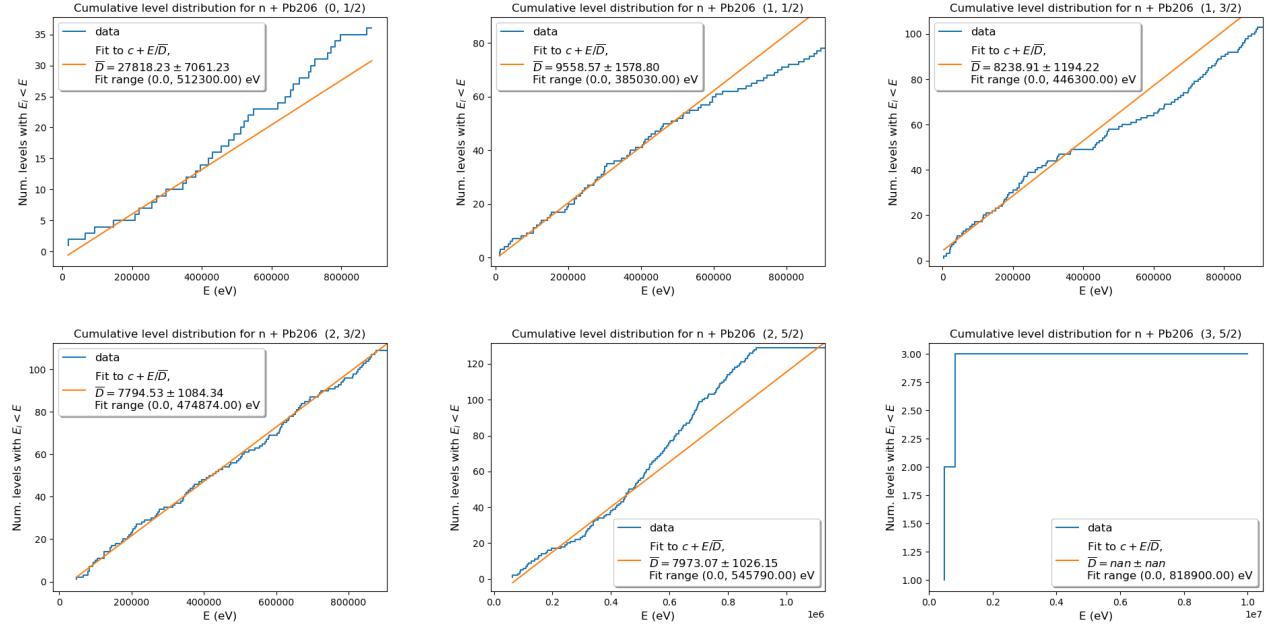


FIG. 2: Cumulative level distributions (CLD) of spingroups $(0, \frac{1}{2}), (1, \frac{1}{2}), (1, \frac{3}{2}), (2, \frac{3}{2}), (2, \frac{5}{2}),$ and $(3, \frac{5}{2})$ in Peter Brain's ${}^{206}\text{Pb}$ neutron resonance data [3]. Each point is the number of resonances with energy less than E . The CLD is known to be linear in the low-energy regime [4]. A perfectly harmonic oscillator would produce a staircase along the 45° line. For our data, we expect an approximately linear line with small deviations according to RMT. However, this is not what we see. The resonances often diverge from linearity, and we observe deficits in spingroups being filled by other spingroups. This is clear evidence of misclassifications.

B. Classifiers

The classifier is essentially the method by which the ML algorithm is trained. A classifier builds a function $f(\cdot) : R^m \rightarrow R^o$ where m is the number of features and o is the number of labels. There are many classifiers to choose from, but for this project we use the classic Neural Network. These classifiers have functions that can be altered using “hyperparameters,” values that can be adjusted to optimize the effectiveness of training an algorithm on our data. One such hyperparameter is the solving method. The solver hyperparameter provides the user with a variety nonlinear programming options such as BFGS and gradient descent methods. This project is not a review of hyperparameter options, but we will be using the LBFGS solving method, as we have found it to be the fastest and most effective. This is important since training on our datasets is computationally intense.

1. Features

Features are the physical properties of neutron resonance that we give to the ML classifier. These include capture width distributions, resonance spacing distributions, L and J priors, neutron width distributions, and many other theoretical models. In this project, we will mostly be discussing capture widths, so it is worth briefly

mentioning the capture width features.

For ${}^{206}\text{Pb}$ we allow 5 spingroups, mentioned in the caption of fig. 2, and I will include a table of ${}^{206}\text{Pb}$ parameters later (table. I) when discussing fits of the Porter-Thomas. But, the feature we give the ML classifier is referred to as the “quadratic difference.” This is the quadratic difference between the n th capture width and the average capture width [8], represented as:

$$\text{qdiff} = \frac{|\Gamma_{\gamma,n} - \bar{\Gamma}_\gamma|^2}{\bar{\Gamma}_\gamma^2} \quad (1)$$

where $\Gamma_{\gamma,n}$ is the n th capture width and $\bar{\Gamma}_\gamma$ is the average capture width. This feature is then automatically applied to ${}^{206}\text{Pb}$ ’s 5 allowed spingroups. The quadratic difference is not a perfect feature, but it is one of the few numerically well-founded options at this time. A feature we are also considering due to recent developments (explained in section III), is a PDF of the Porter-Thomas, which previously was not numerically well-founded due to $\nu \rightarrow \infty$ because we could only assign mean capture widths. But now, we could potentially use this feature because $\nu \rightarrow \infty$ is no longer a problem. However, this would be the scope of another project and part of a major feature and hyperparameter optimization mission.

2. Labels

As mentioned in section II B, a classifier builds a function $f(\cdot) : R^m \rightarrow R^o$ where m is the number of features and o is the number of labels. The labels then, are the output of a trained algorithm. For BRR, the labels are L and J. In other words, the classifier must choose to reclassify the spingroup of a given resonance to one of the 5 allowed spingroups for ^{206}Pb , we call this a “reclassification.” We also explore reclassification only by L, where we disregard the classification status of the J-value. In ML terms, this is “label mode: L,” and when classifying both L and J, we say: “label mode: spingroup.” So, when using label mode: L, $o = 1$, and when using label mode: spingroup, $o = 2$.

C. Synthetic Data

An ML classifier requires abundant and diverse training data. The real data is full of errors (the ones we are trying to fix) and often too short to be used for ML training. So instead, we use synthetic data. Synthetic data mimics the statistical properties of real data in a way that is scalable to a large data set. We do this by fitting distributions to the real data then sampling our resonance parameters from these distributions. This includes Wigner’s Surmise, describing the nearest neighbor spacing distribution [9]; and the Porter-Thomas, which describes a distribution of decay widths for the capture and neutron widths [10]. Previously, instead of using the Porter-Thomas, we simply used mean capture widths for each spingroup, but in this project, we expand and demonstrate the capabilities of using the Porter-Thomas distribution. The stochastic resonance generator for populating synthetic resonance data is contained in the FUDGE processing system [11].

With this method for generating synthetic data we can conduct several types of tests to evaluate the performance of our ML training. First, we create two realizations of synthetic data; one for training the reclassifier algorithm and one for validation. After we collect a training and validation accuracy from this test. We apply the trained reclassifier algorithm to our real data. We use a computer cluster to perform 1000 training events, and with each training event we apply the trained reclassifier algorithm to the real data. From this we can collect a reclassification frequency for each resonance to determine which resonances were incorrectly assigned. This method is further explored in section IV.

D. Random Misassignment Fraction (RMF)

In addition to generating synthetic data, we also apply a random misassignment fraction (RMF) to mimic errors in the real data [6]. This is a process where we purposely misassign a certain percentage of the resonances. If we

misassign 20% of the resonances that is an RMF value of 20%. The RMF is applied in proportion to the number of resonances in each spingroup of the original data set. So the training set with RMF applied will still have the same proportion of each spingroup when compared with the original data set. For ideal training, we need to predict an appropriate RMF value. The RMF value is found by determining the number of incorrectly assigned resonances in the real data (essentially: what is the RMF of the real data?). However, we can only approximate this value, and the method for doing so is still a work in progress (see section VIA). So for now, we try a range of training RMFs from 1% to 99%. From this we can observe trends in the training and validation accuracy.

III. CAPTURE WIDTHS

“Capture width” refers to the decay width of a neutron capture reaction. A decay width is essentially the lifetime of the particle, or rate of decay of the particle, expressed simply as

$$\Gamma = \frac{\hbar}{\tau} \quad (2)$$

where \hbar is Planck’s constant and τ is the particle’s lifetime. On a cross section, this is the width of a given resonance peak. Obviously, experimentally obtaining a value for τ is nearly impossible, so instead we fit the cross section and find the width of a resonance in order to determine Γ . As stated in the abstract, a key element of this project is making our capture width features more realistic. There were a few ideas such as distributing capture widths around the mean width according to a Gaussian distribution, but what we correctly decided upon was fitting then sampling capture widths from a Porter-Thomas distribution. The trouble here is that this method requires the given real data to have measured capture widths. Oftentimes, evaluators will measure a few capture widths then dismissively assign the rest the mean. When this happens, the number of degrees of freedom $\nu \rightarrow \infty$. The Porter-Thomas then becomes a δ -function. This is not numerically well-founded. So, we look for data where the capture widths have been (mostly) measured. With this in mind we were able to find two candidates: ^{103}Rh and ^{206}Pb . A new ^{206}Pb evaluation from the Naval Nuclear Lab was recently made available [3], so we decided to use this isotope. ^{103}Rh will hopefully be the subject of a future BRR evaluation.

A. The Porter-Thomas

The Porter-Thomas follows from RMT and describes a distribution of channel decay widths [10] for a given spingroup. A PDF of the Porter-Thomas conveys the

probability of a given width with number of degrees of freedom ν and is represented as follows:

$$P_{pt}(x|\nu) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\nu/2-1} e^{-x/2} \quad (3)$$

This is, of course, just a χ^2 distribution where $x = \Gamma/\bar{\Gamma}$ and $\bar{\Gamma}$ is the average width of a given spingroup. Physically, ν is the number of states which an excited compound nucleus can decay to, usually by emitting a γ -ray, however, this is complicated by γ -cascades. As mentioned above, we would like to create more realistic capture widths for our synthetic data. In this way, we will be able to use the capture width features in BRR. To do this, we will need to fit the real data to a Porter-Thomas and find a value of ν for each spingroup.

B. Fitting the Porter-Thomas

Instead of fitting a PDF of the Porter-Thomas, we fit the integral. The Porter-Thomas, like a χ^2 distribution, will rapidly approach a Gaussian when increasing the degrees of freedom. This makes the Porter-Thomas challenging for numerical curve fitting methods, so instead, we fit the CDF (mainly this eliminates the exponential terms). The CDF of the Porter-Thomas is as follows:

$$CDF(x) = \int_{x_{min}}^x dx' P(x') \quad (4)$$

$$= \frac{\Gamma(\frac{\nu}{2}, \frac{x_{min}}{2}) - \Gamma(\frac{\nu}{2}, \frac{x}{2})}{\Gamma(\frac{\nu}{2})} \quad (5)$$

where $\Gamma(s, x)$ is the incomplete gamma function [12]. For more intuitive interpretation, the Survival Function (SF) is:

$$SF(x) = \int_x^{x_{max}} dx' P(x') \quad (6)$$

$$= 1 - CDF(x) \quad (7)$$

Basically, just the CDF upside down. We then fit the SF to find the number of degrees of freedom ν for the capture width distribution of a given spingroup. I have shown the SF fit to Peter Brain's ^{206}Pb data [3] in fig. 3. One can see that the fits are not perfect, and there are vertical bars where capture widths have been assigned approximately the same width. These vertical bars are somewhat of a problem for obtaining an accurate value of ν , but we can still extract quite a bit from this data. In the next section (IV), I will provide a list resonance parameters extracted from the data (table. I), including ν 's fit from the capture width SF.

IV. APPLICATION TO ^{206}Pb

To reiterate, in this project, we are evaluating the success of BRR on ^{206}Pb data obtained from a resonance evaluation at the Naval Nuclear Laboratory [3]. We would also like to compare the success of BRR both with and without the use of capture width features. In order to use the capture width features, we had to populate synthetic data with a realistic distribution of capture widths (sampled from a Porter-Thomas). To do this, we had to fit a value of ν_γ for each spingroup. The resonance parameters, including the number of degrees of freedom, for each spingroup are shown in table. I.

The tests performed are as follows:

1. We collect the resonance parameters from the real data [3] and generate 2 realizations of synthetic data.
2. We then train on the 1st realization, which will be split into a training and testing set (convention in ML), and from this we will obtain a training accuracy with capture width features on by testing the trained algorithm on the testing set. Furthermore, we do this for RMF values from 1% to 99% and automate the process using a computer cluster.
3. With the trained algorithm now obtained (for each RMF), we then validate the algorithm on our 2nd realization of synthetic data. Additionally, we apply an RMF of 1%, 50%, and 80% to compare the algorithm's performance on data with varying amounts of misclassifications.
4. For steps 2 & 3 we retrain the algorithm 400 times at each RMF to obtain a reliable average. See section IV B for more about averaging training events.
5. Now that we have some idea of the training quality of BRR, we apply our trained algorithm to the real data. We do this for 1000 training events for each training RMF, so we can average outcomes.
6. With these 1000 reclassification events collected we plot the reclassification frequency of each resonance in the data set to determine which resonances the trained algorithm is certain need to be reclassified.
7. We then repeat steps 2, 3, 4, 5, & 6 without using capture width features and compare.
8. Note: Each step is performed using both label modes: L & spingroup.

A. Validating on synthetic data

In tables. II & III I have listed the average resonance parameters collected from the synthetic data. Notice that

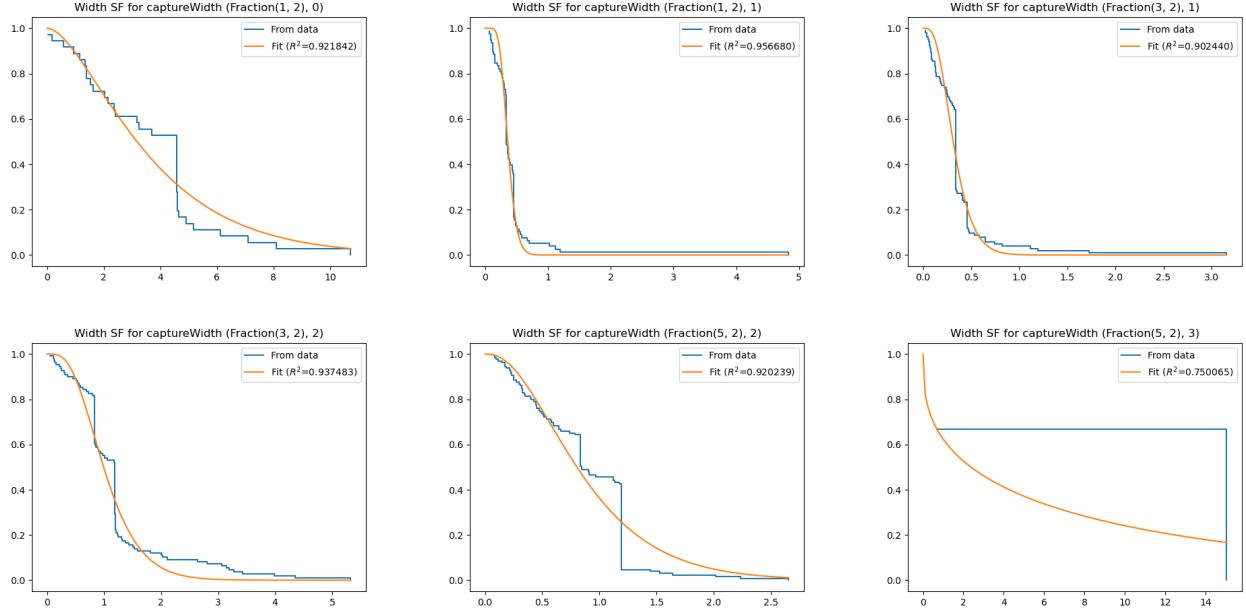


FIG. 3: Survival Function fits for Peter Brain's ^{206}Pb data [3]. Spingroups: $(0, \frac{1}{2})$, $(1, \frac{1}{2})$, $(1, \frac{3}{2})$, $(2, \frac{3}{2})$, $(2, \frac{5}{2})$, and $(3, \frac{5}{2})$. We see that the quality of the fit varies dramatically between spingroups. Additionally, we can see where capture widths have been assigned approximately the same value. Small fluctuations are expected from RMT, but this follows the trend of misclassifications. However, we can still extract a useful value of ν from this data, and I have included these values in table I. For spingroup $(3, \frac{5}{2})$, there wasn't enough data for a good fit, so the value for ν may be left out.

TABLE I: Average resonance parameters extracted from Peter Brain's ^{206}Pb resonance evaluation [3]. Values for capture width number of degrees of freedom ν_γ obtained from fitting SF (using `check-sequence.py`). Currently, our model ignores uncertainties (maybe this will be a future feature).

L	J	Num. res.	D (eV)	$\bar{\Gamma}_n$ (eV)	ν_n	$\bar{\Gamma}_\gamma$ (eV)	ν_γ
0	1/2	36	$25046 \pm 3.3\text{e}2$	$1384 \pm 1.2\text{e}2$	0.853 ± 0.058	3.84 ± 0.14	3.71 ± 0.53
1	1/2	78	$4.88\text{e}4 \pm 8.4\text{e}3$	97.7 ± 4.1	1.003 ± 0.052	$0.3716 \pm 2.9\text{e}-3$	21.6 ± 2.0
1	3/2	103	$3.06\text{e}4 \pm 4.0\text{e}3$	115.9 ± 4.0	0.762 ± 0.024	$0.3392 \pm 5.1\text{e}-3$	8.8 ± 1.2
2	3/2	109	$2.85\text{e}4 \pm 3.7\text{e}3$	140.4 ± 4.5	0.824 ± 0.026	1.074 ± 0.012	8.57 ± 0.78
2	5/2	129	$2.15\text{e}4 \pm 2.4\text{e}3$	124.3 ± 3.2	0.700 ± 0.013	0.916 ± 0.014	5.30 ± 0.42

the parameters deviate slightly from the data in table I but are still approximately the same. Also, we do not include uncertainties in the synthetic data because they are currently of no use to our classifier. Additionally, we see that the size of the data in the synthetic file is much greater than the data in the real file. This is good because the ML classifier needs an abundance of training data for quality training. So, now that we have two realizations of synthetic data we can examine the success of the BRR algorithm.

For this first batch of data, I will examine the BRR performance using only label mode: L. For label mode: L, there are only 3 possible outcomes: $L = 0, 1, \& 2$, in theory this means the classifier should have a higher chance of classifying correctly than when classifying by label mode: spingroup. Shown in fig. 4 are the training accuracies and validation accuracies for many training RMFs and validation RMFs of 1%, 50%, & 80%. The top

TABLE II: 1st realization average resonance parameters extracted from data synthesized from Peter Brain's ^{206}Pb resonance evaluation [3]. Values for capture width number of degrees of freedom ν_γ obtain from fitting SF (using `check-sequence.py`). Recall that the original BRR paper had $\nu_\gamma = \infty$ [6].

L	J	Num. res.	D (eV)	$\bar{\Gamma}_n$ (eV)	ν_n	$\bar{\Gamma}_\gamma$ (eV)	ν_γ
0	1/2	392	25337.6	1426.4	1.0324	3.8809	3.5459
1	1/2	199	50010	103.2	0.948	0.35994	20.12
1	3/2	299	33222.2	122.10	0.9795	0.32281	7.812
2	3/2	343	28933.3	141.51	0.9558	1.0552	7.796
2	5/2	432	23044.3	134.68	1.0124	0.91147	5.292

row is classification without using capture width features, this should and indeed does reflect the performance of the previous version of BRR explored in Ref. [6]. Overall, we observe about a 10% increase in accuracy when using

TABLE III: 2nd realization average resonance parameters.

L	J	Num. res.	\bar{D} (eV)	$\bar{\Gamma}_n$ (eV)	ν_n	$\bar{\Gamma}_\gamma$ (eV)	ν_γ
0	1/2	373	26621.6	1274.6	0.8979	3.8675	3.38
1	1/2	183	53737	108.64	1.083	0.37103	20.32
1	3/2	330	30114.4	115.01	0.9956	0.32242	8.129
2	3/2	354	28145.7	145.77	0.9828	1.02475	8.046
2	5/2	473	21088.5	123.67	0.9533	0.91372	6.025

realistic capture width features.

Shown in fig. 5 is BRR's performance on ^{206}Pb when using label mode: spingroup. When classifying by spin-group we allow 5 possible outcomes (see table. I) since we must consider combinations of L and J. The accuracy for classification by spingroup isn't quite as good as L, but there is still noticeable improvement between with and without capture width features.

For both L and J we also plot the starting accuracy on the validation plot. This value is simply 1 - validation RMF. We see general improvement over this line for each validation RMF except for val. RMF = 1%. For val. RMF = 1, the accuracy decreases, but for all other val. RMFs the accuracy increases after just one classification. Perhaps an iterative reclassification process could incrementally improve the validation accuracy after successive reclassifications. A resonance sequence with any number of misclassifications may then be able to incrementally improve until close to fully correct. And, at this moment, there is nothing to suggest that an iterative method would not work for both label modes. This is something we hope to research soon.

B. Reclassifying real resonance data

The next step is using BRR to reclassify ^{206}Pb neutron resonances, is to actually apply a trained algorithm to real resonance data. In this case, that real resonance data is Peter Brain's ^{206}Pb data [3].

As a first step, we estimate how many training events are required to obtain a reliable average reclassification assignment. The results of this test are shown in fig. 6. For both label mode L and spingroup, as well as with and without capture widths, we see that the average fraction of resonances reclassified converges around 400 training events. In our actual reclassification attempt (see fig. 7) we use an average from 1000 training events, so we will be well within the average convergence range.

With a recommended lower limit for training events in mind, we can now explore the reclassification frequency of each of the resonances in our sequence. For the real data, we cannot simply print out a validation accuracy because we do not know exactly which resonances need reclassifying and what they need reclassifying to. We can, however, observe the decisions our trained algorithm is making and make an informed decision from the collected data. This is why we find a reclassification frequency. For

this test, we train an algorithm 1000 times, and with each training event, we apply a trained algorithm to the real data. We can then count whether a resonance was reclassified then collect resonance's reclassification status. After, a 1000 training events we will have a reclassification count for each resonance. Then, we divide this value by the total number of training events (1000) and obtain our reclassification frequency. We then sort by energy and plot each of the resonances to obtain fig. 7.

The main takeaway from fig. 7 is the dramatic difference between classification with and without capture width features. When using capture width features, for both label mode L and spingroup, we see that the algorithm is much more selective and certain in its reclassification choices. There are resonances that the algorithm is choosing to reclassify almost 100% of the time and resonances where the algorithm is clearly not as likely to reclassify. This is in stark contrast to the reclassifications without capture width features. We do however see that there is a specific energy region where the algorithm is less certain about reclassification: the very low energy and higher energy regimes that occur at the edge of our cross section. **For a future test, we need to take the resonances reclassified over say 90% of the time, make a new data set, and refit a cross section. Also, we need to collect cumulative level distributions and survival function fits of the data with selected reclassified resonances.**

As a preliminary test of our reclassification validity, we plot observed, reclassified, and expected fraction of resonances in fig. 8. In this test, we want to see if the reclassified fraction of resonances for a given assignment resembles what is expected. However, this test is somewhat flawed because the expected fraction of resonances is calculated using distributions from the given real resonance data [3]. Nevertheless, when using label mode L, the expected fraction of resonances over a given ΔE (a bin), is calculated using the expected number of resonances $N_L = \Delta E / D_L$ then the expected fraction is $N_L / N_{tot} = D / D_L$; where N_{tot} is the total number of resonances in the bin, D is the average spacing in the bin, D_L is the spacing of resonances with label L. Similarly, for label mode spingroup, the expected fraction is $N_{sg} / N_{tot} = D / D_{sg}$; where D_{sg} is given in table. I. Between with and without capture widths, using capture width features is much more likely to elicit a meaningful response from the trained classifier. This data is taken as a result of 1000 training events. Ideally, we would want to see that the solid line is moving away from the dashed (observed) line and toward the expected (dotted) line. In the case of label mode L, this is only the case for L = 2, for the other resonances, there seems to be no correlation. For spingroup, each of the reclassified spingroups are more or less moving in the right direction, so this is an interesting discrepancy. Additionally, the difference between with and without capture width features is also seen here. When not using capture width features the reclassified fraction of resonances barely changes from the

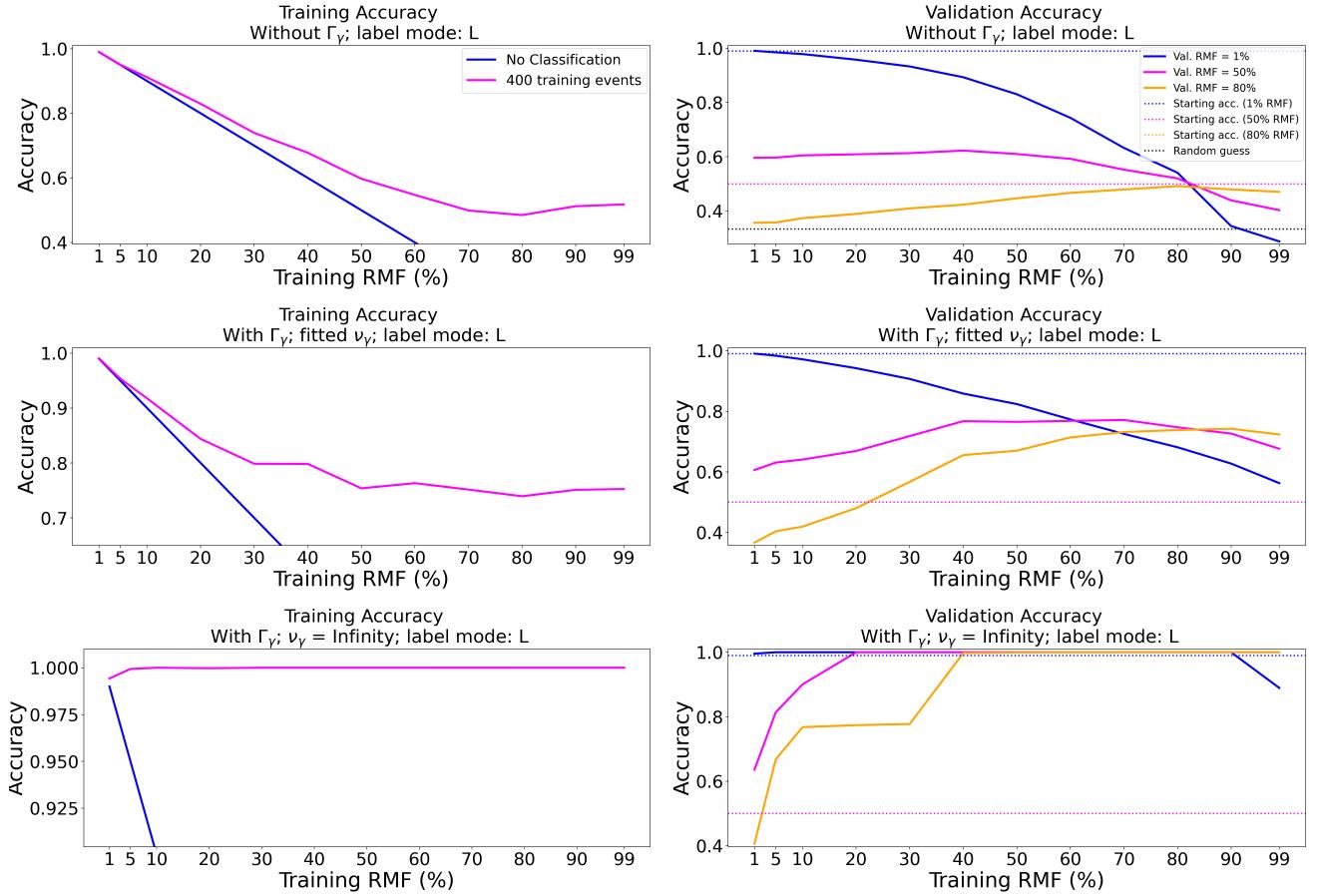


FIG. 4: Classification by LABEL MODE: L, where each accuracy is an average of 400 training events. On the left are the training accuracies, and on the right: the validation accuracies. The top row of plots is classification without using capture width features. This data has capture widths distributed according to a Porter-Thomas, the classifier is just ignoring them. The middle row of plots is classification with realistic capture widths, and, in this case, the classifier is in fact using the capture width features. Here we see about a 10% increase in accuracy! And finally, on the bottom row, we classify using capture width features, but the capture width distribution is unrealistic. Each resonance is assigned the mean capture width for that spingroup. This is very unrealistic and not applicable to the real data. In this case, we basically have a linear programming problem that we are solving with metaheuristics, very silly. The no classification line is simply what one would get from randomly guessing, also known as 1 - RMF.

observed. Fortunately, when using capture widths, there is a much more noticeable adjustment in the fraction of reclassified resonances.

V. SUMMARY AND CONCLUSIONS

BRR could potentially become a valuable tool for neutron resonance evaluation. An accurate description of neutron interaction cross sections is of paramount importance to nuclear fission and fusion reactors. And as these fields become increasingly populous, the need for accurate nuclear data will continue to grow.

In this project, we have shown that BRR, a machine learning based reclassifier, can be used to reclassify neutron resonance sequence with an increasing level of reliability. With the introduction of realistic capture width

features, we were able to increase the accuracy and degree of certainty by which a trained algorithm reclassifies neutron resonances. In the near future, we would like to compile a new resonance sequence from the reclassified resonances and test our data by fitting a cross section. Additionally, we need to collect cumulative level distributions and survival functions for our reclassified sequence. By performing these tests, in addition to the fraction of resonances changed graph, we can determine the efficacy of our approach and find new ways to improve.

Another interesting area of future research is iterative reclassifications. We want to explore how successive reclassifications to a given resonance sequence can iteratively improve over time. With this method, it may be possible to fully converge a given resonance sequence to 100% accuracy. We have some preliminary data available from intern projects of summer 2022 and summer

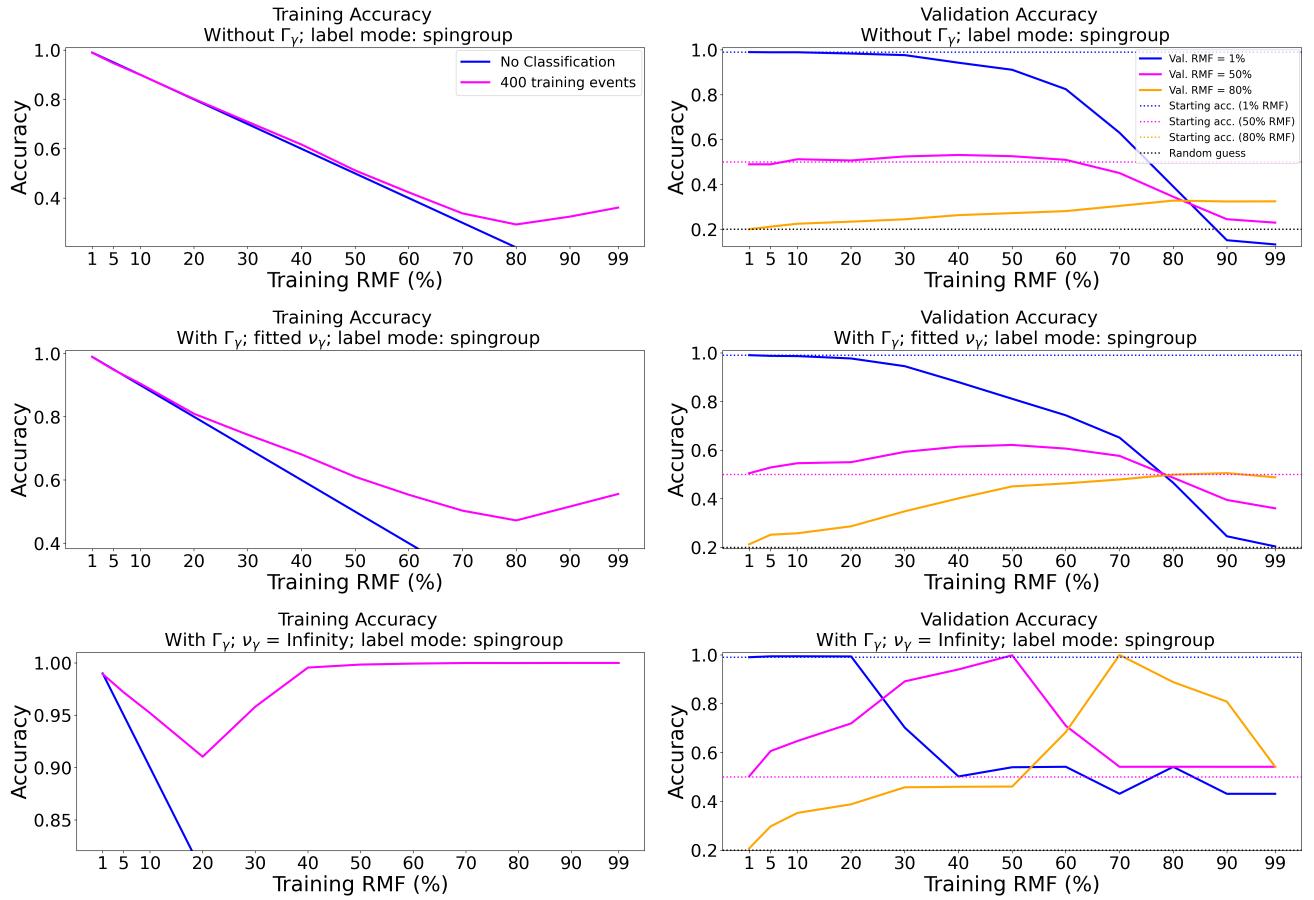


FIG. 5: Classification by LABEL MODE: spingroup, where each accuracy is an average of 400 training events. On the left are the training accuracies, and on the right: the validation accuracies. The top row of plots is classification without using capture width features. This data has capture widths distributed according to a Porter-Thomas, the classifier is just ignoring them. The middle row of plots is classification with realistic capture widths, and, in this case, the classifier is in fact using the capture width features. Here we see about a 10% increase in accuracy! And finally, on the bottom row, we classify using capture width features, but the capture width distribution is unrealistic. Each resonance is assigned the mean capture width for that spingroup. This is very unrealistic and not applicable to the real data. The no classification line is simply what one would get from randomly guessing, also known as 1 - RMF.

2023 that suggests even just a few successive reclassification could have a huge impact on the overall reclassification accuracy. So, we will perform a full analysis of this method soon.

Finally, there are other isotopes out there! The ^{206}Pb was a useful isotope because it was a recent evaluation with quality capture width assignments. We are also interested in the more abundant isotope: ^{208}Pb , as we may be able to create high quality training data from ^{208}Pb resonance data. Additionally, we briefly considered ^{103}Rh earlier in this project, but spent most of our time analyzing ^{206}Pb , so in the future we would also like to use this isotope for a BRR analysis.

VI. APPENDIX

A. Determining an appropriate training RMF when validating on real data

The question all the evaluators want to know the answer to is: how wrong is my data? Well, here on the BRR project, we are working on finding an answer. Outlined in the following is a general procedure that we are still working on for finding a suitable training RMF. This training RMF ideally will be approximately equivalent to the percentage of resonances that were misclassified in the real data, the trick, of course, is figuring out what percentage of resonances were reclassified in the real data. So the process we have for now is as follows:

1. Find the approximate cut off energy for the linear regime of the cumulative level distribution for the real data. We do this by finding the derivative

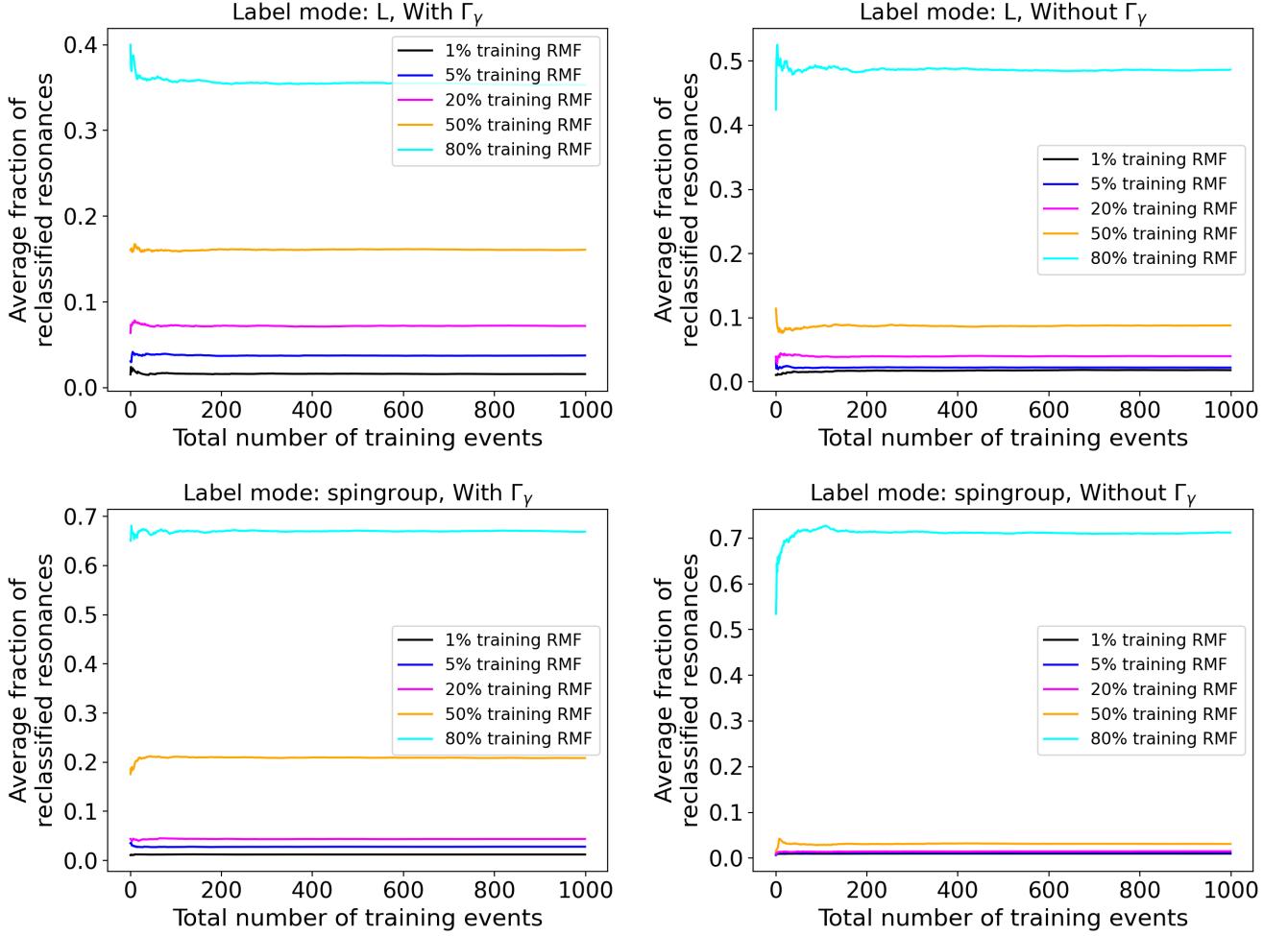


FIG. 6: In this plot, I show the average fraction of resonances reclassified vs the number of training events. Each label mode and with and without capture widths are plotted. These plots serve to show how many training events are needed to obtain a repeatable prediction. We see that each of the lines completely converges around 400 training events. So any more than 400 training events is excessive, at least for ^{206}Pb that is, other isotopes will likely behave differently.

of the staircase of the real data. Then, we determine where the slope of the staircase derivative begins converging to 0. Interestingly, this derivative plot exhibited periodicity, so I used an approximate Fourier Cosine series fit to find a suitable cut off.

2. We fit a line to the staircase plot from the resonances at 0 energy to the fit cut-off energy found in the previous step.
3. We find the difference between the integral of the line and the staircase plot computed after the energy cut-off.
4. The % RMF is then this difference divided by the value of integral of the line after the cut-off region.
5. This integral is discretized to each point in the graph, so we have a % RMF for each point on our plot. The RMF selected for a given spingroup is

then the average of all the % RMFs after the cut-off energy.

6. The suggested RMF is then a weighted average of the average % RMFs for each spingroup. For clarity, this value is calculated as the following:

Recommended Training RMF =

$$\sum_n^{\# \text{ of labels}} |\langle \text{RMF} \rangle_n| \frac{N \text{ with } E_n > E_{\text{cut},n}}{\sum_n^{\# \text{ of labels}} N \text{ with } E_n > E_{\text{cut},n}}$$

where N is the number of resonances. There is a figure for each step of this process and this is shown in fig. 10. This figure shows a CLD at the fit line, the derivative of the staircase, and a graph of the RMF. This is shown for each label.

For ^{206}Pb we concluded that a good training RMF would be between 10% and 30%. Somewhat of a large

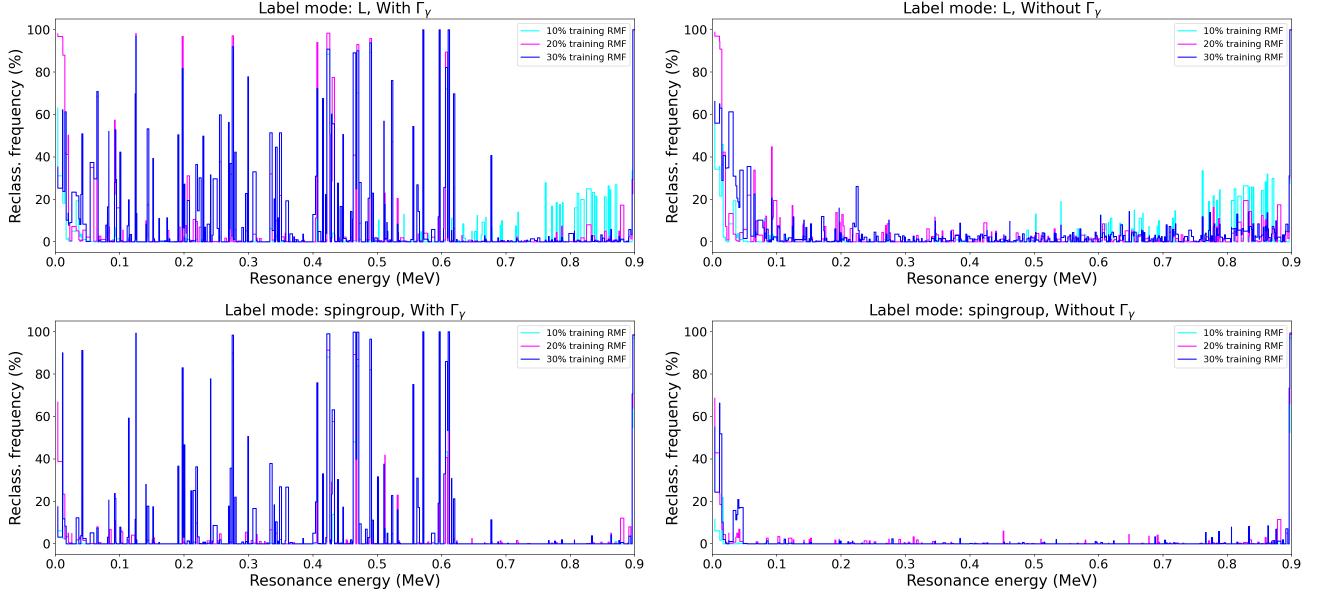


FIG. 7: This is a plot of the reclassification frequency of each resonance over 1000 training events. When using capture width features with more realistic capture width distributions in the training data, we see that the algorithm is much more selective and consistent than without the capture widths; and this is true for both label modes. From this, we can determine which resonances are good candidates for reclassification. The reclassification frequency is even more striking with label mode spingroup.

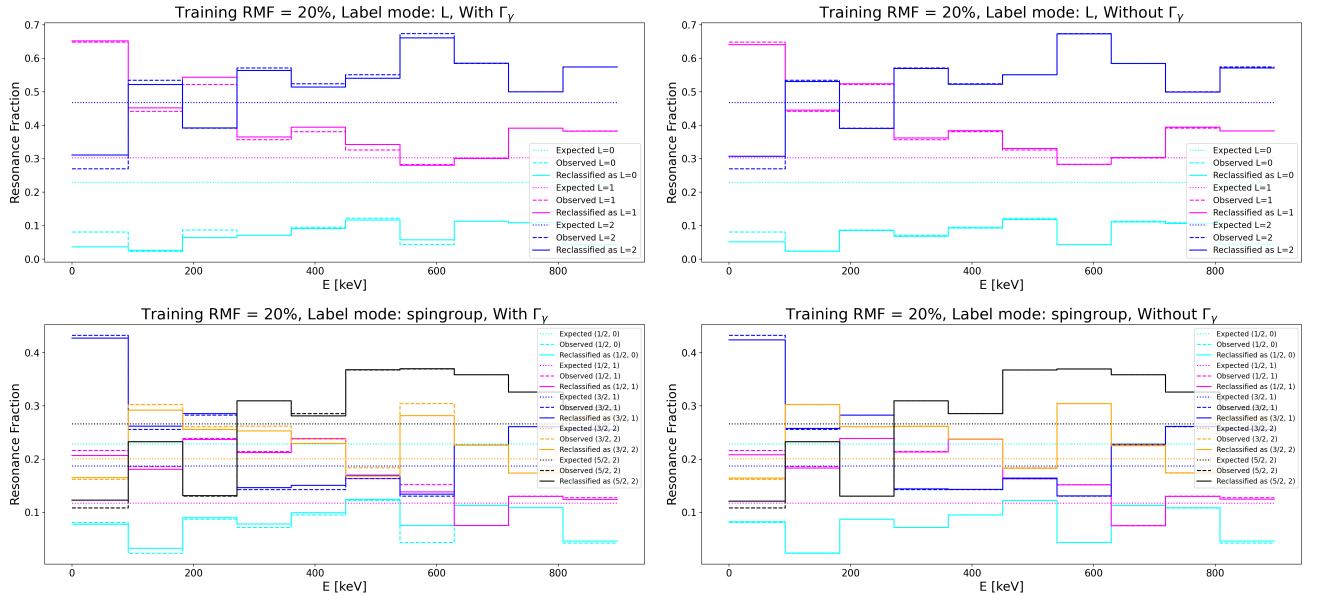
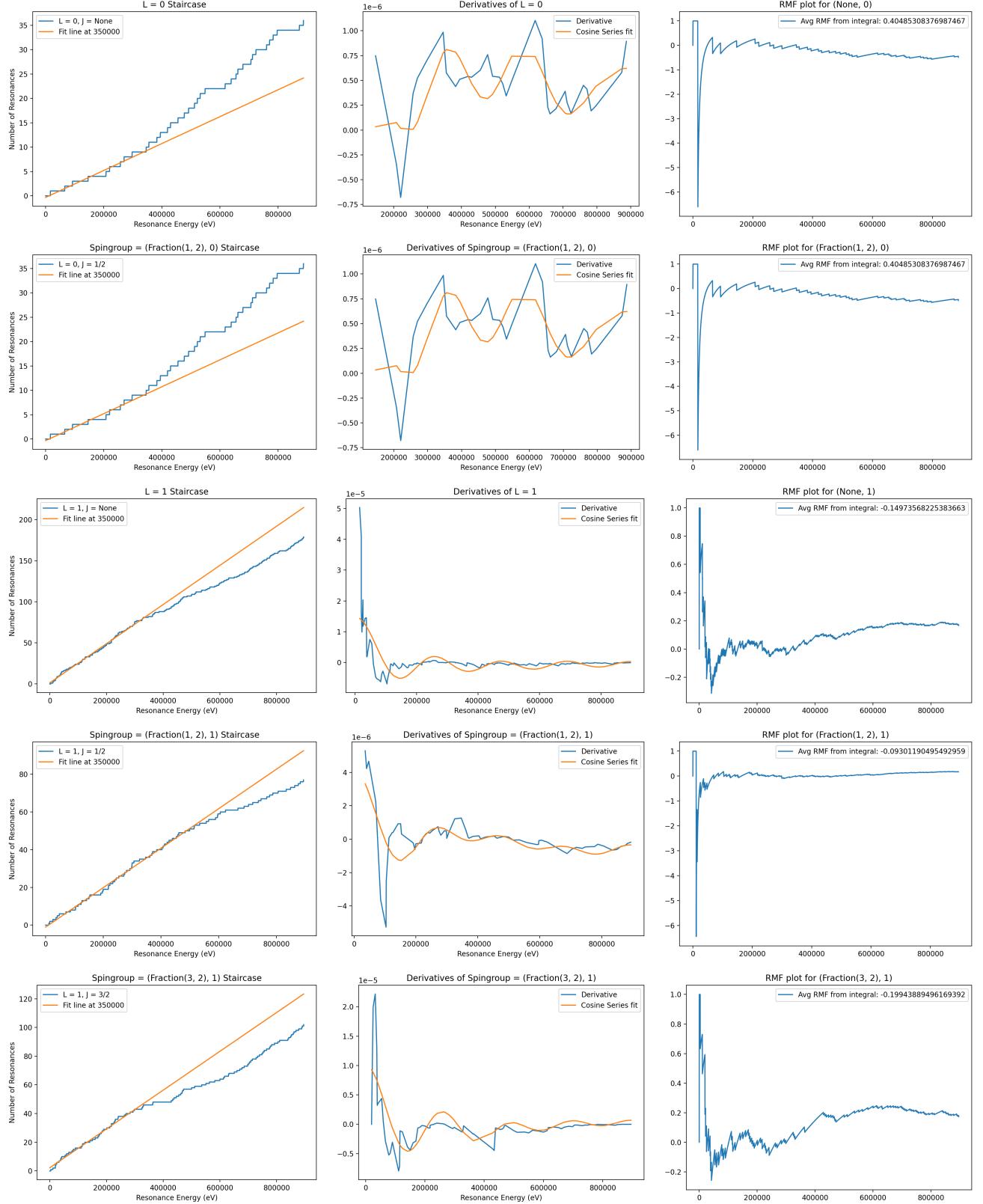


FIG. 8: Plots showing the observed, reclassified, and expected fraction of resonances for each allowed label of ^{206}Pb . Additionally, we compare between with and without training using capture width features. The resonances are separated into 10 bins for ease of viewing. The expected fraction of resonances is calculated from Peter Brain's ^{206}Pb data [3]. When using label mode L, the expected fraction of resonances over a given ΔE (a bin), is calculated using the expected number of resonances $N_L = \Delta E / D_L$ then the expected fraction is $N_L / N_{tot} = D / D_L$; where N_{tot} is the total number of resonances in the bin, D is the average spacing in the bin, D_L is the spacing of resonances with label L. Similarly, for label mode spingroup, the expected fraction is $N_{sg} / N_{tot} = D / D_{sg}$; where D_{sg} is given in table I. Between with and without capture widths, using capture width features is much more likely to elicit a meaningful response from the trained classifier. This data is taken as a result of 1000 training events.

FIG. 9: *Continued on next page...*

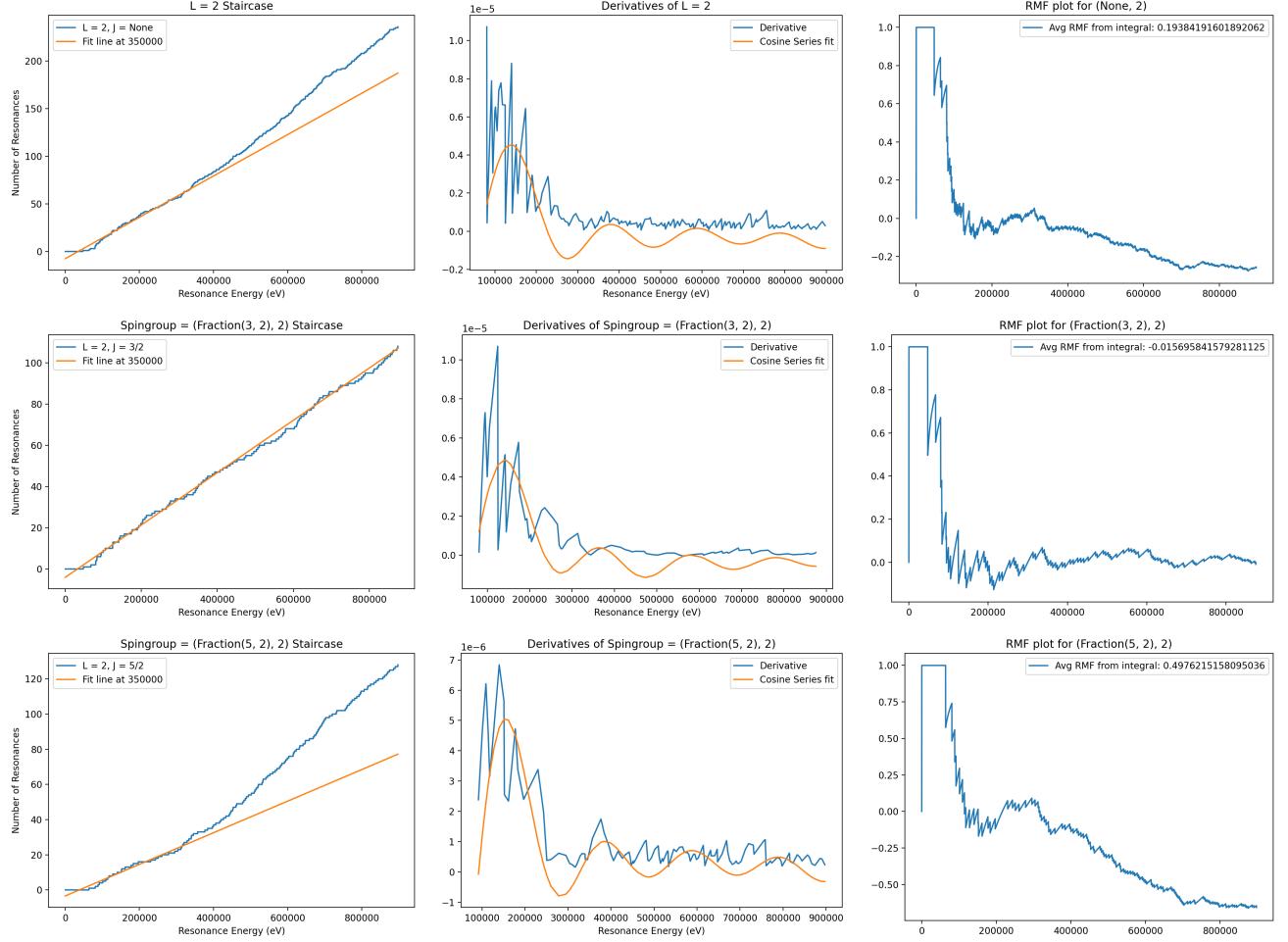


FIG. 10: Process for determining a recommended training RMF. First we find a cut-off energy for the CLD, then we find the difference between the staircase integral and linear fit integral above the cut-off energy. We then divide this value by the fit integral to obtain a % RMF.

range, but we are still refining the method. When testing training RMFs of 10, 20, and 30% we obtained similar results, so this may be something to consider **when making a generalized resonance sequence: taking resonances not only above a certain cut-off frequency, but also only taking resonances when all RMFs (10, 20, and 30) reclassify a given resonance above a certain cut-off frequency**. This may not end up entirely beneficial, but it might be an idea worth researching.

B. SHAP Values

SHapley Additive exPlanations (SHAP) is a metric for determining the efficacy of ML training features [13]. If any single feature is too dominant in the feature set, our trained algorithm is subject to biasing and thus, overfitting. This makes the resulting trained algorithm not sufficiently generalizable. We can check for overfitting

using a validation score, but the validation test will not tell us which features are introducing non-negligible bias. An in-depth feature test is still needed for BRR, but that will be the purview of another project. A short feature review was published in a lab report [8]. Right now, we are just interested in seeing how the SHAP values change with and without using realistic capture widths.

In fig. 11, I show what is called a “SHAP summary plot.” In this graph, I train an algorithm for reclassifications by label mode: L and show the reliance of our training on each of the features used. In this graph, we can see that for low training RMF values, our reclassifier algorithm is heavily reliant on the priors. Then as the RMF increases, the distribution of features begins to even out a bit and the capture with features take over, but not excessively though. This is a good argument for using a training RMF that is reflective of the real data’s RMF, as we can see that when we use low RMF, we introduce quite a bit of bias into our algorithm.

In figs. 12 & 13, we plot SHAP decision trees for various training RMFs. The decision tree ranks features by

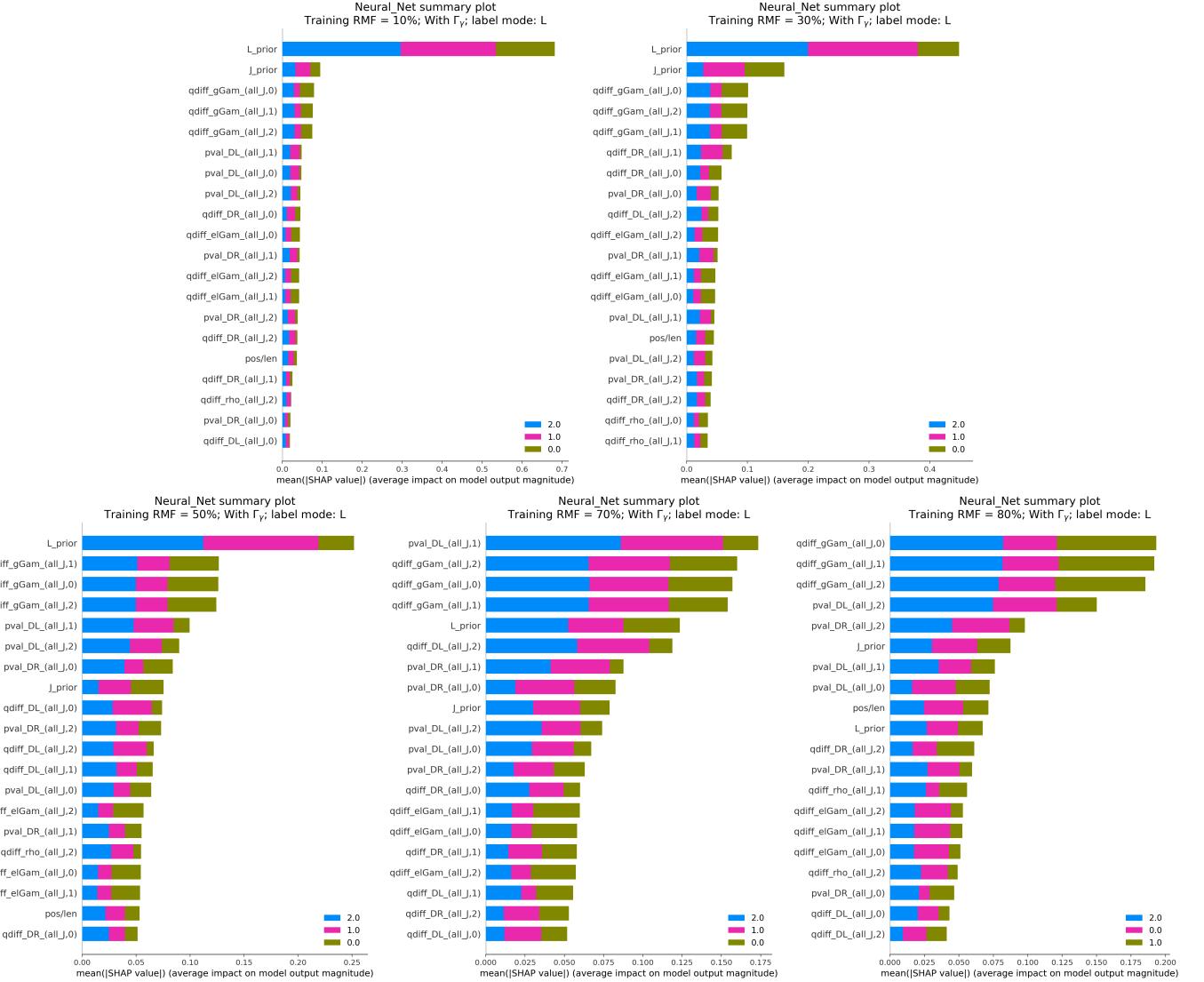


FIG. 11: SHAP summary for each of the features used in reclassifications by label mode: L. We see that for low training RMFs our algorithm is heavily reliant on the priors, and as we make the RMF more realistic, our algorithm becomes less biased toward just one feature. Note: these graphs do not show all the features, just the most prominent ones.

importance, then cumulates the SHAP values for each of the features, drawing a line between them. So, the difference on the x-axis for adjacent features is the SHAP value for the end feature. In fig. 12, we do not use capture width features. Generally, we see that the cumulative SHAP values converge to similar values regardless of training RMF. This could be a sign of biasing and overfitting in our algorithm. This biasing problem is potentially remedied, as shown in fig. 13, when training an algorithm with capture width features. We see that usually the final SHAP value is more diverse than without capture widths. This could be a sign that our algorithm is less biased, but, still, an in-depth feature analysis is still needed and will no doubt be the focus of a future project.

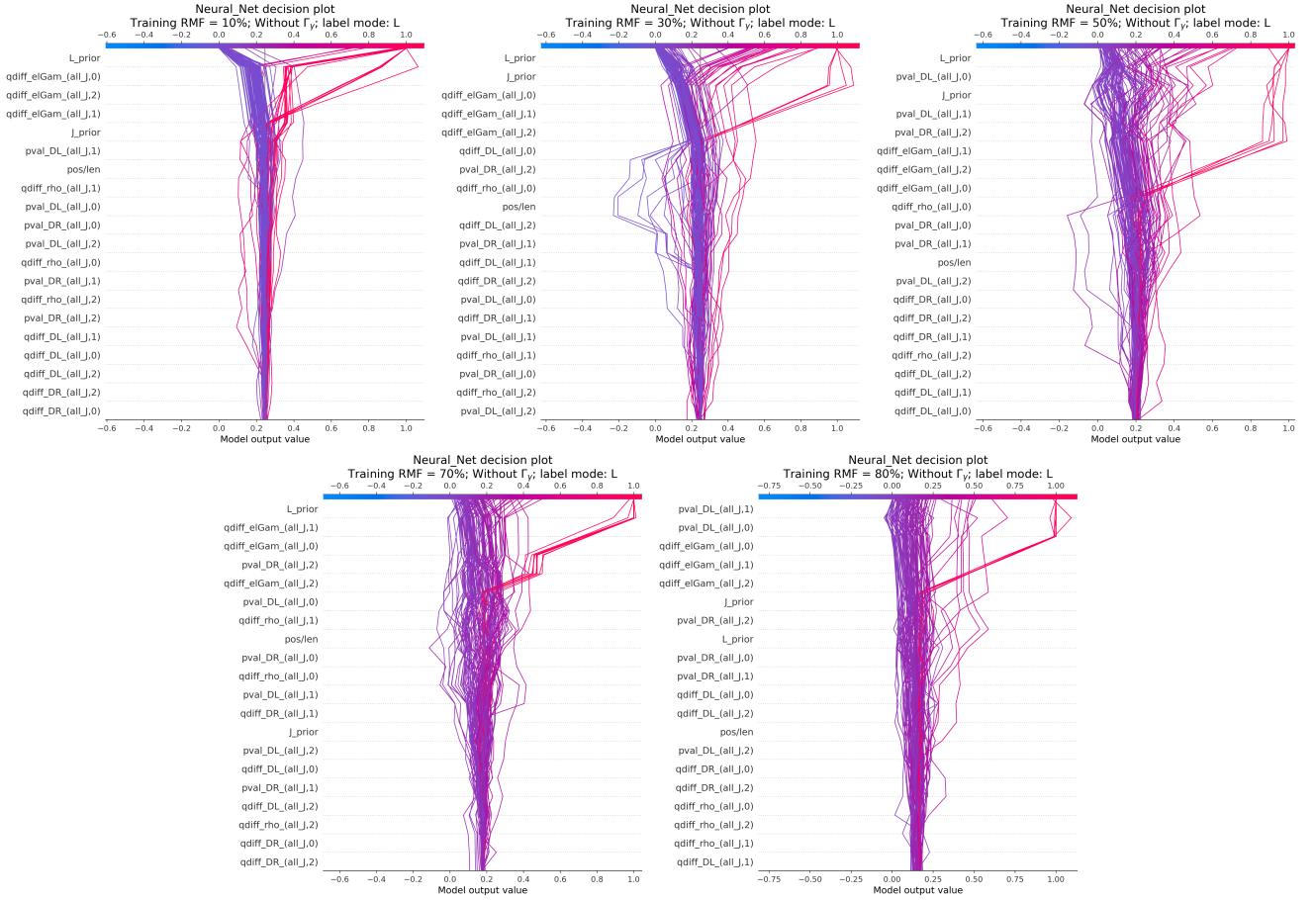


FIG. 12: SHAP decision tree for label mode: L, without capture width features. The decision tree ranks features by importance, then cumulates the SHAP values for each of the features, drawing a line between them. So, the difference on the x-axis for adjacent features is the SHAP value for the end feature.

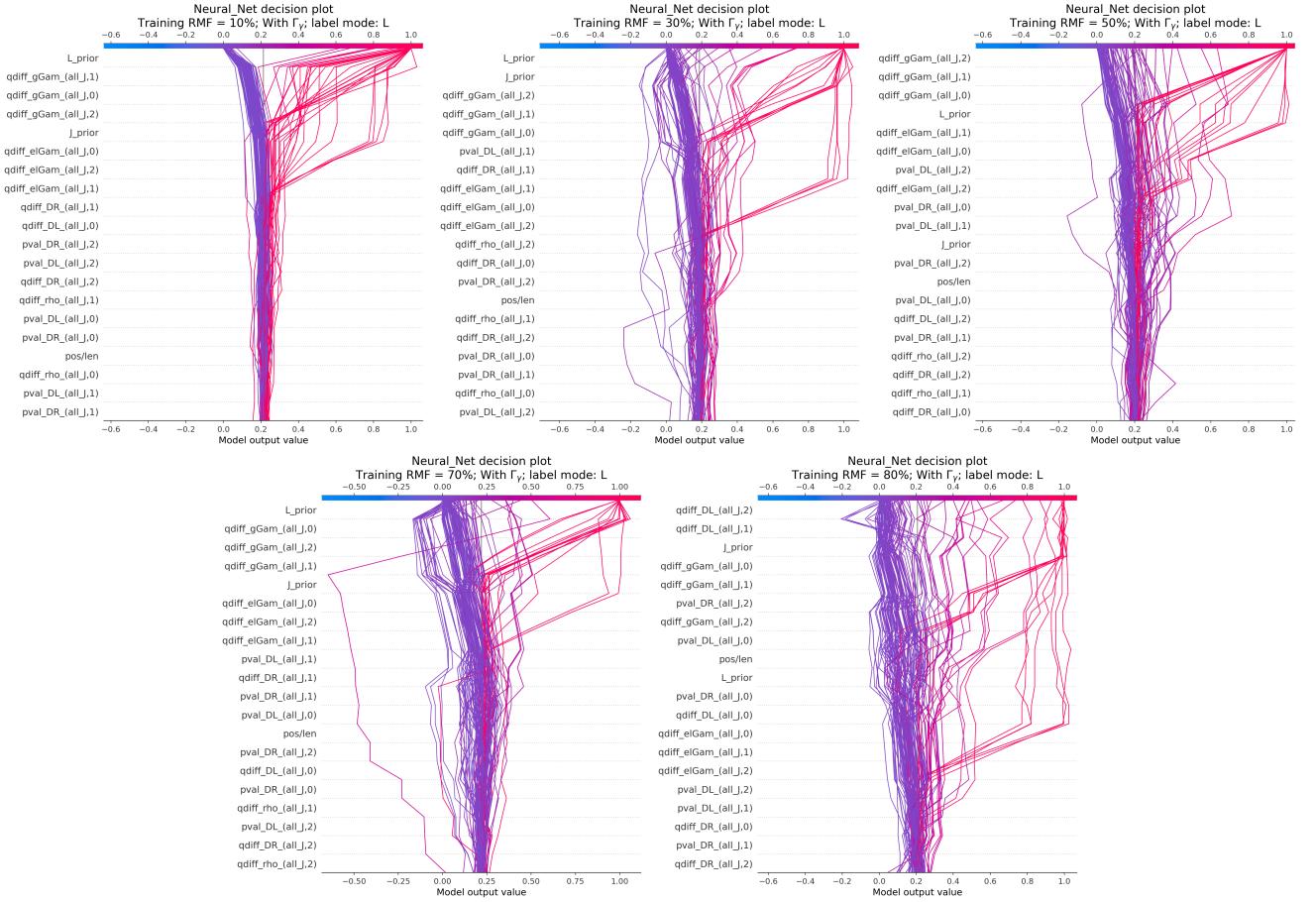


FIG. 13: SHAP decision tree for label mode: L, with capture width features. We see, that when compared to without capture width features, our algorithm converges to a wider range of values. This is potentially a sign that our algorithm is less biased and less likely to overfit when using capture width features.

-
- [1] A. Borella, F. Gunsing, M. Moxon, P. Schillebeeckx, and P. Siegler. High-resolution neutron transmission and capture measurements of the nucleus ^{206}Pb . *Phys. Rev. C*, 76:014605, Jul 2007.
- [2] D.A. Brown, M.B. Chadwick, R. Capote, A.C. Kahler, A. Trkov, M.W. Herman, A.A. Sonzogni, Y. Danon, A.D. Carlson, M. Dunn, D.L. Smith, G.M. Hale, G. Arbanas, R. Arcilla, C.R. Bates, B. Beck, B. Becker, F. Brown, R.J. Casperson, J. Conlin, D.E. Cullen, M.-A. Descalle, R. Firestone, T. Gaines, K.H. Guber, A.I. Hawari, J. Holmes, T.D. Johnson, T. Kawano, B.C. Kiedrowski, A.J. Koning, S. Kopecky, L. Leal, J.P. Lestone, C. Lubitz, J.I. Márquez Damián, C.M. Mattoon, E.A. McCutchan, S. Mughabghab, P. Navratil, D. Neudecker, G.P.A. Nobre, G. Noguere, M. Paris, M.T. Pigni, A.J. Plompen, B. Pritychenko, V.G. Pronyaev, D. Roubtsov, D. Rochman, P. Romano, P. Schillebeeckx, S. Simakov, M. Sin, I. Sirakov, B. Sleaford, V. Sobes, E.S. Soukhovitskii, I. Stetcu, P. Talou, I. Thompson, S. van der Marck, L. Welser-Sherrill, D. Wiarda, M. White, J.L. Wormald, R.Q. Wright, M. Zerkle, G. Žerovník, and Y. Zhu. Endf/b-viii.0: The 8th major release of the nuclear reaction data library with cielo-project cross sections, new standards and thermal scattering data. *Nuclear Data Sheets*, 148:1 – 142, 2018. Special Issue on Nuclear Reaction Data.
- [3] Peter Brain, Yaron Danon, David Brown, Devin Barry, Amanda Lewis, and Toshihiko Kawano. Pb evaluation - endf/b-viii.1. *unpublished*, 2023.
- [4] Declan Mulhall. Using the Δ_3 statistic to test for missed levels in mixed sequence neutron resonance data. *Phys. Rev. C*, 80:034612, Sep 2009.
- [5] S. F. Mughabghab. *Atlas of Neutron Resonances, Vol. 1 and 2*. Elsevier, Amsterdam, 2018.
- [6] G. P. A. Nobre, D. A. Brown, S. J. Hollick, S. Scoville, and P. Rodríguez. Novel machine-learning method for spin classification of neutron resonances. *Phys. Rev. C*, 107:034612, Mar 2023.
- [7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [8] Gustavo Nobre, D. A. Brown, S. Scoville, M. Fucci, S. Ruiz, R. Crawford, A. Coles, and M. Vorabbi. Expansion of machine-learning method for classifying neutron resonances. 10 2021.
- [9] M.L. Mehta. *Random Matrices and the Statistical Theory of Energy Levels*. Academic Press, New York, NY, 1967.
- [10] C. E. Porter and R. G. Thomas. Fluctuations of nuclear reaction widths. *Phys. Rev.*, 104:483–491, Oct 1956.
- [11] For Updating Data and Generating Evaluations (FUDGE): LLNL code for managing nuclear data. <https://github.com/LLNL/fudge>, 2021.
- [12] George B. Arfken, Hans J. Weber, and Frank E. Harris. Chapter 13 - gamma function. In George B. Arfken, Hans J. Weber, and Frank E. Harris, editors, *Mathematical Methods for Physicists (Seventh Edition)*, pages 599–641. Academic Press, Boston, seventh edition edition, 2013.
- [13] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.