Article

Data Augmentation for Neutron Spectrum Unfolding with Neural Networks

James McGreivy [1,2,*], Juan J. Manfredi[3], and Daniel Siefman[2]

[1] Department of Physics; University of California, Berkeley; Berkeley, CA
[2] Nuclear Criticality Safety Division; Lawrence Livermore National Laboratory; Livermore, CA
[3] Department of Engineering Physics; Air Force Institute of Technology; Wright-Patterson Air Force Base, OH
[*] Correspondence: jamesmcgreivy@berkeley.edu

Abstract: Neural networks require a large quantity of training spectra and detector responses in order to learn to solve the inverse problem of neutron spectrum unfolding. In addition, due to the under-determined nature of unfolding, non-physical spectra which would not be encountered in usage should not be included in the training set. While physically realistic training spectra are commonly determined experimentally or generated through Monte Carlo simulation, this can become prohibitively expensive when considering the quantity of spectra needed to effectively train an unfolding network. In this paper, we present three algorithms for the generation of large quantities of realistic and physically motivated neutron energy spectra. Using an IAEA compendium of 251 spectra, we compare the unfolding performance of neural networks trained on spectra from these algorithms, when unfolding real-world spectra, to two baselines. We also investigate general methods for evaluating the performance of and optimizing feature engineering algorithms.

Keywords: Detector Response Unfolding; Neutron Spectrum Unfolding; Machine Learning; Neural Network; Feature Engineering

1. Introduction

Accurate measurements for the energy spectrum of a neutron radiation source are important across widely-ranging applications, such as monitoring workplace radiation exposure, medical imaging, and tracking rates of nuclear reactions for fission or fusion. In many ways, neutron energy spectra act as fingerprints to identify the nuclear composition of a given neutron radiation source. Thus, these measurements are also crucial in national security applications such as detecting the smuggling of illicit nuclear materials or nuclear warhead treaty verification [1,2].

There are several radiation detection methods used to determine the energy spectrum of a neutron radiation source, including Bonner sphere detectors [3,4] and scintillation detectors [5–7]. Almost all neutron detectors, however, involve the use of materials with characteristic responses to radiation which preserve information about the energy of the incoming radiation. That characteristic response does not directly contain the energy information; in order to determine the incident neutron energy spectrum, this response must first be processed by spectrum unfolding algorithms.

Unfolding algorithms rely on the ability to construct a linear transformation, known as the detector response function, which gives a mapping between neutron energy spectra and the corresponding measured characteristic detector responses. This linear transformation is generally determined explicitly through experiment, or by using Monte Carlo radiation transport codes [6,8,9] such as Geant4 [10] and MCNP [11]. The act of unfolding thus involves inverting this linear transformation in order to determine the neutron energy spectrum which is most likely to have given rise to a known detector response. Since the number of detector response bins is usually less than the number of energy bins to be unfolded, attempting to directly invert the detector response function would yield an under-determined system of linear equations with potentially infinite solutions. Thus, to unfold the correct neutron energy spectra from a given detector response, constraints must be placed on the possible solution space to filter out extraneous and non-physical solution spectra.

Many commonly used unfolding algorithms employ iterative methods [4,7,12] to converge on the unfolded spectrum, given some *a priori* guess of the solution spectrum. The constraints are usually baked into iterative unfolding algorithms - often through initializing the iteration from a realistic-looking neutron energy spectrum or by only considering neutron energy spectra which follow from theoretical models [12]. However, iterative methods can become prohibitively computationally expensive for fine energy bin structure or when unfolding must be performed *in situ*, i.e. when *a priori* knowledge is for some reason not available.

More recently, neural network based unfolding algorithms have been explored [4,13, 14]. Neural networks have the advantage that, once trained, they are extremely quick to evaluate on any input detector response regardless of the size of the network or number of energy bins to be unfolded. However, the accuracy and generalizability of a neural network is dependent upon the quantity of data used to train it. In addition, the under-determined nature of neutron spectrum unfolding means that no single function exists, which an unfolding network can learn to approximate, that will reliably unfold the correct neutron energy spectra for all possible detector responses.

Including non-physical spectra within a training set may potentially decrease unfolding performance by forcing an unfolding network to learn mappings between detector responses and spectra which would never arise in the real world. However, by restricting the types of neutron energy spectra which an unfolding network has been trained on, one may constrain the solution space through exposure to only physical spectra. Thus, there are two key considerations for effectively training an unfolding neural network: first, for the purposes of generalizability a network should be trained on a sufficiently large quantity of diverse spectra. Second, these training spectra should only come from within the distribution of spectra likely to be seen in usage.

This is an essential problem in the application of neural networks to spectrum unfolding - how should one quickly obtain large training sets, which contain only spectra which could be found in the field? Due to the quantity of spectra required, it is prohibitively expensive to train networks on energy spectra and detector responses which have been determined experimentally. Instead, it is common to use Monte Carlo simulations to calculate realistic neutron energy spectra using common neutron source types, such as ^{252}Cf and Am-Be, in combination with a variety of moderator materials and geometries [3,5]. However, performing these simulations can be computationally demanding given the several hundred or more training spectra needed by neural networks. Regardless of the detector type or neural network architecture used, it is desirable to be able to generate a large quantity of realistic neutron energy spectra without the need for explicit simulation or measurement.

In this paper we investigate this problem through the use of algorithms for automatic training data generation. The goal of this work is to outline spectra generation algorithms which optimally constrain the types of neutron energy spectra an unfolding neural network is exposed to during training, and thus improve unfolding performance on real-world neutron energy spectra. The tested generation methods include random spectra, random perturbations from real-world spectra, superpositions of random Gaussians, and random perturbations from a parameterized representation of common kinds of neutron energy spectra. We test these methods using open data (in particular Bonner sphere detector response and corresponding neutron spectra) [8] with which readers can validate our approach and test their own.

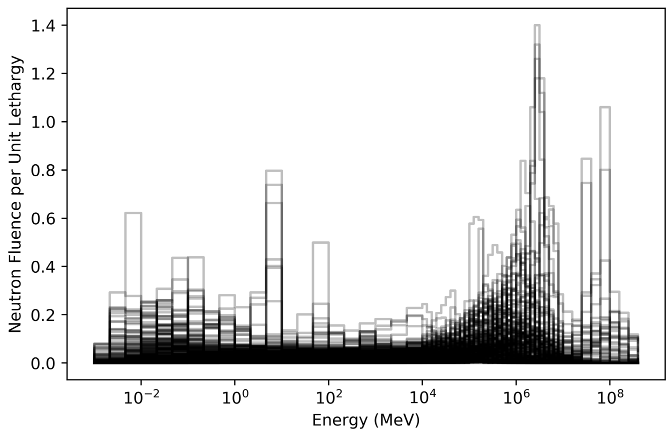

Figure 1. All 251 real-world neutron energy spectra provided by the IAEA technical report [8] plotted on top of each other.

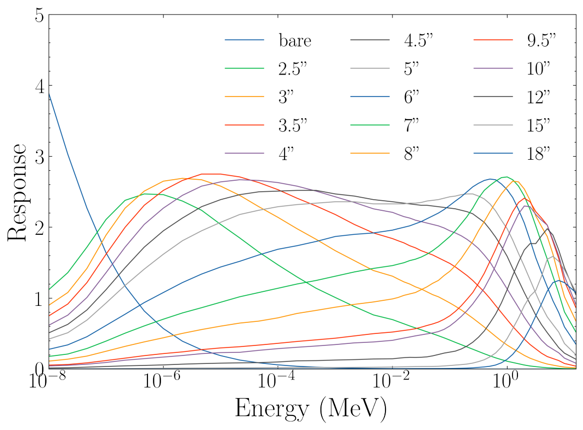

Figure 2. The IAEA Bonner sphere response function for 15 different moderator thicknesses (in inches).

Spectra Kind	Number Present	Number Used by **PSA**
reference	11	2
moderated reference	13	4
fission	102	30
moderated fission	12	4
workplace	27	6
fusion	4	1
high energy	17	4
accelerator	47	7
cosmic ray	5	1
boron therapy	13	3

Table 1. The kinds of neutron energy spectra present in the IAEA data set are shown in column 1. The number of spectra for each kind is shown in column 2, and the number of spectra present in the 62 IAEA spectra designated for use by the **PSA** algorithm (Section 2.2.1) is shown in column 3.

2. Materials and Methods

To evaluate the effectiveness of a neutron energy spectra generation algorithm, a data set of real-world neutron energy spectra was needed on which to test unfolding performance. For this work, we used an IAEA technical report [8] containing 251 measured and simulated neutron energy spectra which range from 0.001 eV to 15.8 MeV. All 251 spectra are plotted in Figure 1, and the kinds of spectra present in this set are shown in Table 1. These spectra cover a wide range of applications that include calibration sources, evaporation fields, cosmic ray background, and reactor spectra. Our neural networks seek optimal performance on this wide variety of spectra, but we believe these methods would be applicable to constrained neutron energy ranges as well. This report also contains a fully solved Bonner sphere detector response matrix (Figure 2), with which we may calculate an expected detector response given any discretized neutron energy spectrum:

$$d_i = \sum_j R_{ij}\, \phi(E_j). \tag{1}$$

Here, d_i is the expected response of the ith Bonner sphere; R is the detector response matrix, which characterizes the response of the ith Bonner sphere to neutrons in the jth energy bin E_j; and $\phi(E)$ is the neutron energy spectrum. Neutron energy spectra are always normalized to unity:

$$\sum_j \phi(E_j) = 1. \tag{2}$$

2.1. *Unfolding Neural Network Architecture*

For this paper, the unfolding neural network used is a densely-connected network with two hidden layers. Both hidden layers use the leaky rectified linear unit (Leaky ReLU) activation function, and the final layer uses a linear activation function. Dropout layers are placed between every pair of densely connected layers in the hope that they will prevent over-fitting and improve unfolding generalizability [15].

The tunable hyper-parameters of the network include the training batch size, the slope of the Leaky ReLU activation function, the neuron dropout rate, and the number of neurons in each hidden layer. Optimal hyper-parameters were determined through Bayesian hyper-parameter tuning [16]. This hyper-parameter tuning was performed against the IAEA spectra, however these hyper-parameters are used for all neural networks throughout this work regardless of the training data used. This was done to ensure consistency in the complexity of the unfolding network so that differences in unfolding performance may be primarily attributed to the quality of the training data. All networks will have identical architecture, but the weights and biases of each layer which are determined from back propagation depend entirely upon the training set used.

2.2. *Random Neutron Spectra Generation Algorithms*

2.2.1. Baseline Methods: Perturbed IAEA Spectra (**PSA**) and Random Spectra (**RAND**)

We select baseline neutron energy spectra generation algorithms in order to evaluate the effectiveness of our other spectra generation algorithms. The first algorithm uses random spectra (**RAND**) where each energy bin is given a random value between 0 and 1. Then, the entire spectrum is properly normalized according to Equation 2. Because this method places zero constraints on the types of neutron energy spectra our neural network will be exposed to during training it will be used as a baseline to evaluate the effectiveness of non-physical training data. Example random spectra are shown in Figure 3.

We would also like to be able to approximate a near-perfectly constrained solution space. While we could train directly on IAEA spectra, the 251 spectra contained in the IAEA report are not enough to effectively train an unfolding network on - we were unable to get performance comparable to our other unfolding networks using a 80-20 training-validation split of only IAEA spectra. Instead, perturbations were performed on a random subset of

25% of the 251 IAEA spectra - this technique will be referred to as the perturbed spectra algorithm (**PSA**). This was done so that the remaining 75% of the IAEA spectra may be quarantined and used only for final analysis. Since this method is only capable of producing neutron energy spectra which closely resemble the IAEA data, a neural network trained on this algorithm would likely not generalize as well to neutron energy spectra which are not already present in the IAEA data set. This algorithm is also highly specific to the exact energy bin structure used by the IAEA report. Therefore, it is used only as an upper-bound baseline for training data specificity, against which we may compare other neutron spectra generation algorithms.

For the **PSA** algorithm, one spectrum is randomly selected from the 62 IAEA spectra which were previously designated for use (listed in Table 1). In addition, a scaling factor α is chosen from a normal distribution, with a mean 1 and a width of σ_α:

$$\alpha \sim \mathcal{N}(1, \sigma_\alpha). \tag{3}$$

The chosen spectrum is interpolated, and this scaling factor is used to create a new neutron energy spectrum:

$$\tilde{\phi}(E) = \phi(\alpha E). \tag{4}$$

After this perturbation, the spectrum $\tilde{\phi}(E)$ is properly normalized (Eq 2). This algorithm has a single tunable feature parameter σ_α which will be optimized using the method outlined in section 2.3.2. Example perturbed spectra are shown in Figure 3.

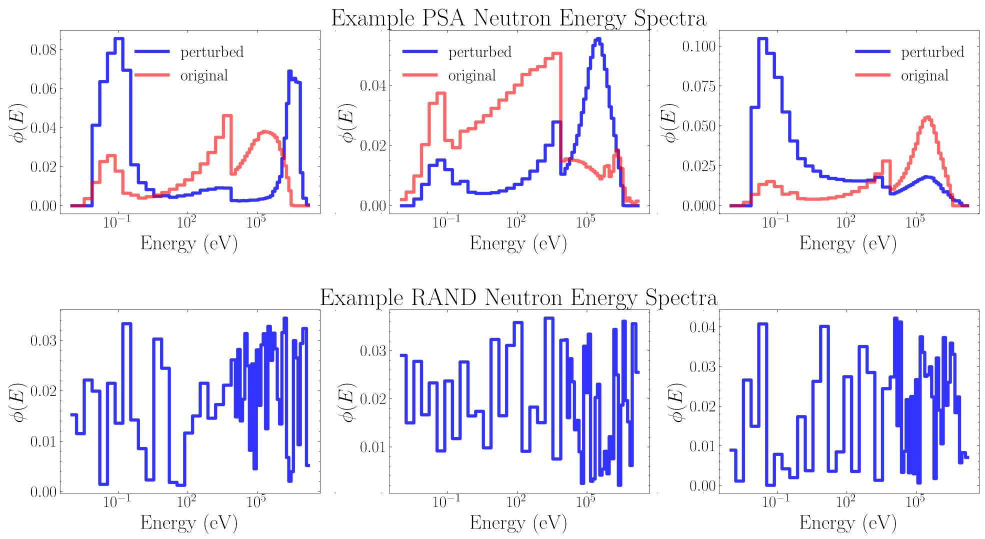

Figure 3. Example neutron energy spectra, generated by **PSA** (top) and **RAND** (bottom). For **PSA** the original unperturbed spectra is plotted in red. Both of these algorithms are only intended as baselines, for the purposes of comparison against.

2.2.2. Method 1: **GAUSS1**

This algorithm relies on the qualitative observation that most real-world neutron energy spectra, when viewed on a logarithmic energy scale, consist primarily of one or more overlapping peaks. At greater than 1 MeV energies, neutrons resulting from fission or evaporation may show up as distinct peaks in the energy distribution. At thermal neutron energies of less than 1 eV, these structures are generally smaller in amplitude and result from thermalization physics. As a rough approximation to the shape of these peaks when viewed on a logarithmic scale, Gaussians were chosen due to their mathematical simplicity and smoothness.

This spectrum generation algorithm works by adding a random number of Gaussian peaks to a blank neutron energy spectrum. The process is controlled by several tunable parameters:

- μ_{center}, σ_{center} - parameters between 0 and 1, which correspond to the mean and standard deviation of the normal distribution of means of the Gaussian peaks.
- μ_{width}, σ_{width} - parameters between 0 and 1, which correspond to the mean and standard deviation of the normal distribution of widths of the Gaussian peaks.
- A_{decay} - controls the amount by which to cumulatively suppress the amplitude of subsequent Gaussian peaks. In general, this gives rise to one central energy peak, surrounded by smaller amplitude structures.
- σ_A - controls random deviations of the amplitude of the Gaussian.
- p_{peak} - the probability to add an extra Gaussian peak to the spectra.

These are the feature parameters of the spectra generation algorithm, and must be tuned to reasonable values by comparison to the IAEA neutron energy spectra. Processes for tuning each of these parameters is outlined in section 2.3.

To generate a neutron energy spectrum we begin with a blank spectrum $\phi_{i=0}(E) = 0$ in which all energy bins are initialized to zero. Then, an iterative process is carried out in which one or more Gaussian-shaped peaks are added on top of this blank spectrum. Starting with $i = 0$,

1. Parameters for the width and position of the Gaussian peak, on a logarithmic scale, are sampled from their corresponding normal distributions. The values -3 and 10 correspond to the logarithms of the lowest energy bin value and the number of orders of magnitude which our energy bins cover (10^{-3} eV to 10^7 eV) respectively.

$$\begin{aligned} \bar{x} &= -3 + \mid 10\,\mathcal{N}(\mu_{center}, \sigma center) \mid \\ \sigma &= \mid 10\,\mathcal{N}(\mu_{width}, \sigma width) \mid \end{aligned} \tag{5}$$

2. The mean height A_i is perturbed according to σ_A:

$$A = A_i\,\mathcal{N}(1, \sigma_A) \tag{6}$$

3. A logarithmic Gaussian is added to the neutron energy bins:

$$\phi_i(E) = \phi_{i-1}(E) + A\,\exp\left(-\frac{1}{2}\left(\frac{\log_{10}(E) - \bar{x}}{\sigma}\right)^2\right) \tag{7}$$

4. The mean amplitude for the next iteration is suppressed according to A_{decay}:

$$A_{i+1} = A_i * A_{decay} \tag{8}$$

5. $i \rightarrow i + 1$ and this entire process is repeated with a probability of p_{peak}.

After this iterative process has terminated, the final spectrum is properly normalized (Eq 2). Some example spectra, with properly tuned feature parameters, are shown in Figure 4 below.

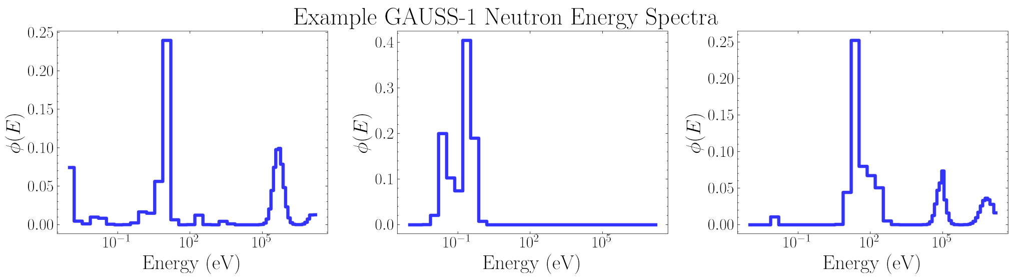

Figure 4. Example spectra generated by **GAUSS1**. The feature parameters used to generate these plots were tuned according the method in section 2.3.2

2.2.3. Method 2: **GAUSS2**

This algorithm is very similar to method 1: however it further constrains the types of possible neutron energy spectra by coupling the width and height of the Gaussian peaks to their central energy. This is based on the observation that, due to the physics of neutron thermalization, Gaussian structures at lower energies tend to have smaller amplitudes. Structures with a larger energy spread will also tend to have smaller amplitudes than those which are highly concentrated around some central energy.

The tunable parameters for this algorithm are:

- μ_{width}, σ_{width} - parameters between 0 and 1, which control the statistics of the width of the Gaussian peaks.
- ω_{width} - controls how strongly a Gaussian peak's width is suppressed by its energy scale.
- ω_{amp} - controls how strongly a Gaussian peak's amplitude is increased by its energy scale.
- A_{decay} - controls the amount by which to cumulatively suppress the amplitude of subsequent Gaussian peaks.
- p_{peak} - the probability to add an extra Gaussian peak to the energy spectra.

Similarly to method 1, these feature parameters must be tuned to reasonable values prior to usage (Section 2.3).

To generate a neutron energy spectrum, we will once again start with a blank spectrum $\phi_{i=0}(E) = 0$ in which all energy bins are initialized to zero. Then, the following iterative process is carried out, starting with $i = 0$,

1. A value between 0 and 1 is randomly chosen:

$$s \sim \mathcal{R}(0, 1) \tag{9}$$

2. Parameters for the width and position of the Gaussian peak, as seen on a logarithmic scale, are determined:

$$\begin{aligned} \bar{x} &= -3 + 10\,s \\ \sigma &= |\,10\,\mathcal{N}(\mu_{width}, \sigma_{width})\,|\,(1 + \omega_{width}\,s)^{-1} \end{aligned} \tag{10}$$

3. The amplitude of the Gaussian is suppressed according to ω_{amp}:

$$A = A_i\,(1 + \omega_{amp}\,s) \tag{11}$$

4. The Gaussian is added to the logarithm of the neutron energy bins:

$$\phi_i(E) = \phi_{i-1}(E) + A\,\exp\left(-\frac{1}{2}\left(\frac{\log_{10}(E) - x_i}{\sigma_i}\right)^2\right) \tag{12}$$

Model	Thermal $\phi_{th}(E,...)$	Epithermal $\phi_e(E,...)$	Fast $\phi_f(E,...)$	High Energy $\phi_{hi}(E,...)$
Fission	$\left(\frac{E}{T_0^2}\right)e^{-E/T_0}$	$\left[1-e^{-(E/E_d)^2}\right]E^{b-1}e^{-E/\beta'}$	$E^{\alpha}e^{-E/\beta}$	0
Evaporation	$\downarrow$	$\downarrow$	$\left(\frac{E}{T_{ev}^2}\right)e^{-E/T_{ev}}$	0
Gaussian	$\downarrow$	$\downarrow$	$exp\left(-\frac{1}{2}\left(\frac{E-E_m}{\sigma E_m}\right)\right)$	0
High Energy	$\downarrow$	$\downarrow$	$\left(\frac{E}{T_{ev}^2}\right)e^{-E/T_{ev}}$	$\left(\frac{E}{T_{hi}^2}\right)e^{-E/T_{hi}}$

Table 2. The thermal, epithermal, fast, and high energy component spectra for each of the four FRUIT models [12]. Note the inconsistent units across each component spectra - this problem is remedied through proper normalization, which is done before the weighted sum of each component is performed.

5. The amplitude for the next iteration is suppressed according to A_{decay}:

$$A_{i+1} = A_i * A_{decay} \tag{13}$$

6. $i \rightarrow i+1$ and this entire process is repeated with a probability p_{peak}. 203
After this process has terminated the spectrum is properly normalized (Eq 2). Example 204
spectra, with properly tuned feature parameters, are shown in Figure 5 below. 205

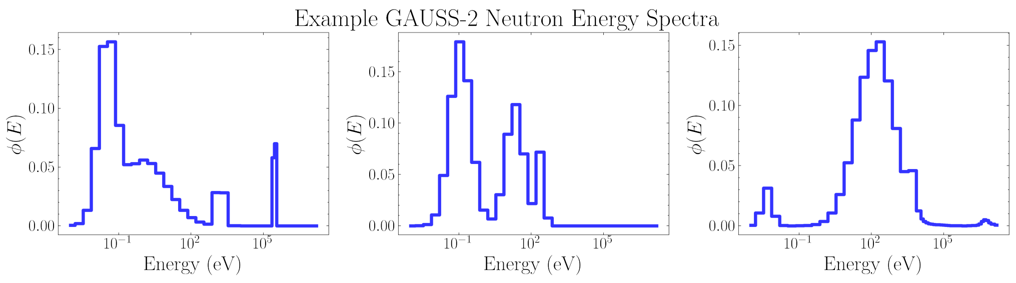

Figure 5. Example random spectra generated by GAUSS2. The feature parameters used to generate these plots were tuned according to the method in section 2.3.2

2.2.4. Method 3: FRUIT Spectra Generation Algorithm 206

This spectrum generation algorithm is inspired by the ***F****rascati* ***U****nfolding* ***I****nteractive* ***T****ool* (FRUIT) [12]. FRUIT iteratively unfolds neutron energy spectra by assuming that they are well described by one of four theoretical models: fission, evaporation, Gaussian, or high energy spectra. These four models describe a wide variety of common neutron energy spectra, including fission sources, radionuclide neutron sources, medical cyclotrons, and hadron accelerators [12]. 207 208 209 210 211 212

The fission, evaporation, Gaussian, and high energy models consist of a weighted sum of four component spectra, describing the physics of thermal, epithermal, fast, and high energy neutrons. The component spectra for each model have been reprinted in Table 2. Importantly for this paper, each of these component spectra takes tunable parameters. By properly selecting well motivated parameters, we generate physically accurate spectra for the given model type. The tunable parameters for each model are: 213 214 215 216 217 218

- Fission: P_{th}, P_e, P_f, b, β', α, β 219
- Evaporation: P_{th}, P_e, P_f, b, β', T_{ev} 220
- Gaussian: P_{th}, P_e, P_f, b, β', E_m, σ 221
- High Energy: P_{th}, P_e, P_f, P_{hi}, b, β', T_{ev}, T_{hi} 222

Bedogni et al. [12] interpret each parameter taken by these models, and provide definite values for $T_0 = 2.53 * 10^{-8}$ MeV, the conventional thermal neutron energy, and $E_d = 7.07 * 10^{-8}$ MeV, the lower energy range of epithermal neutrons. The parameters 223 224 225

P_{th}, P_e, P_f, and P_{hi} control the proportion of each model's spectrum that comes from the thermal, epithermal, fast, and high energy components respectively.

To randomly generate neutron energy spectra from these models, we must first determine reasonable distributions for each model parameter. To do this the fission, Gaussian, and high energy models were fitted to each individual IAEA neutron energy spectrum. Since the high energy model is identical to the evaporation model but with extra degrees of freedom from the high energy component, the evaporation model was not considered for fitting; the high energy model is always guaranteed to provide an equivalent or better fit. Due to the high degree of non-linearity within each model's fitting parameters, the accuracy of the fit depends heavily upon the initial guesses for the fit parameters. To find a globally optimal fit, Bayesian optimization was used to search for appropriate initial guesses. For each IAEA spectrum, only the model that had the minimal mean squared fitting error was considered.

Using these distributions for each model parameter, which should cover the possible values taken by realistic neutron energy spectra, the procedure of generating a random neutron energy spectra is as follows:

1. Randomly select between the fission, evaporation, Gaussian, or high energy models.
2. Randomly sample model parameters from the distributions determined from IAEA spectra. For an evaporation spectrum, parameters are sampled from the relevant high energy parameter distributions.
    - P_{th}, P_e, P_f, and P_{hi} are always selected from the same fit so that they sum to 1, ensuring normalization.
3. Create the final spectrum as a weighted sum of the relevant component spectra:

$$\begin{aligned} \phi(E) = {} & P_{th}\,\phi_{th}(E,\{\text{params}\}) + P_e\,\phi_e(E,\{\text{params}\}) \\ & + P_f\,\phi_f(E,\{\text{params}\}) + P_{hi}\,\phi_{hi}(E,\{\text{params}\}) \end{aligned} \tag{14}$$

Example spectra for each model type are shown in Figure 6 below.

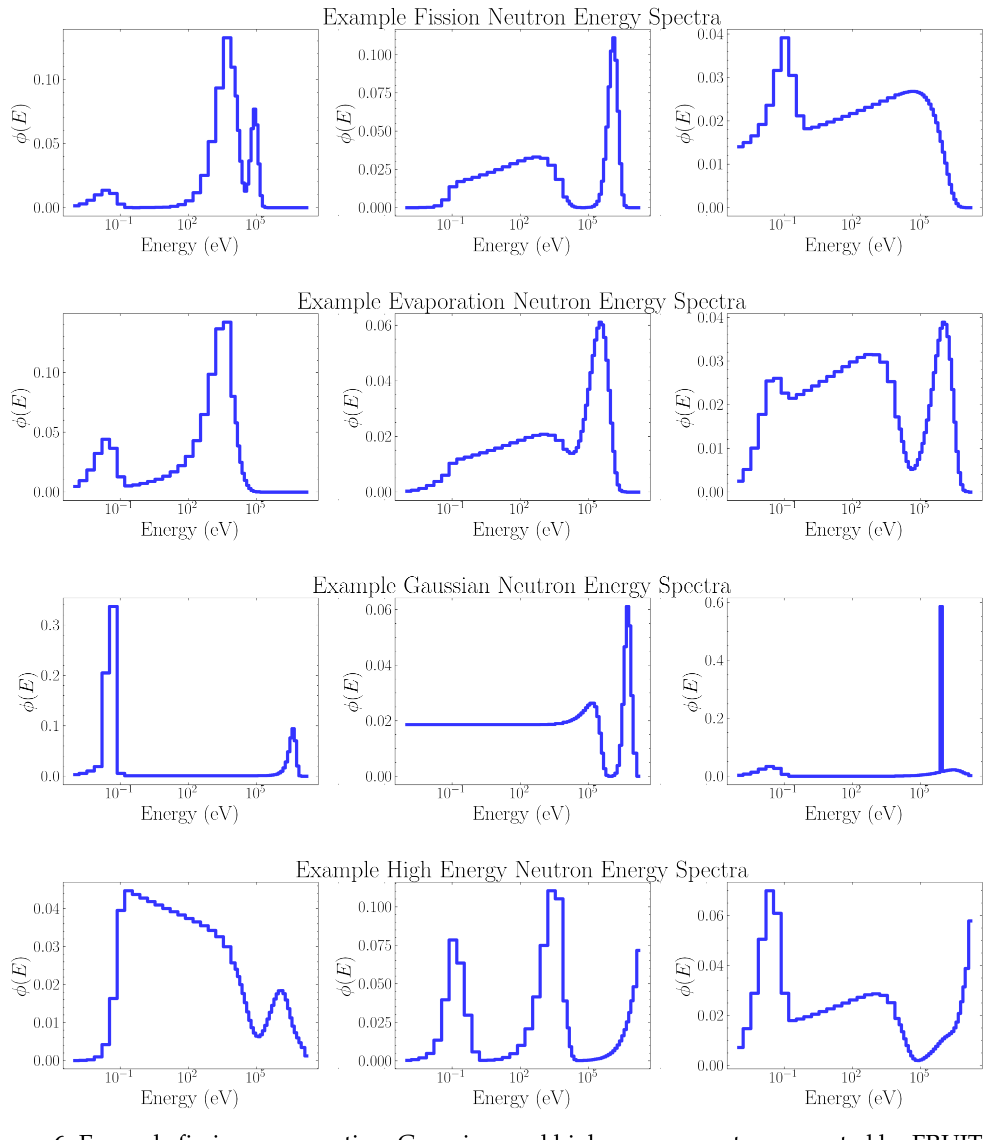

Figure 6. Example fission, evaporation, Gaussian, and high energy spectra generated by FRUIT.

2.3. Tuning the Feature Parameters of a Spectra Generation Algorithm 249

The **PSA**, **GAUSS-1**, and **GAUSS-2** algorithms all take one or more feature parameters 250
which must be tuned in order to maximize the unfolding performance of a neural network 251
trained on the produced spectra. To evaluate the effectiveness of a given choice of feature 252
parameters we must first construct an objective function to optimize against. This objective 253
function should take a given set of feature parameters and return the effectiveness of those 254
feature parameters at generating realistic spectra. In particular, we would like our objective 255
function to capture the idea that neural networks which have been trained on data from a 256
properly tuned algorithm will have the best possible unfolding performance when shown 257
real-world neutron energy spectra. This feature parameter tuning process is outlined in 258
Figure 7. We consider two candidate objective functions. 259

2.3.1. Feature Parameter Tuning by Evaluation on Neural Networks 260

The first objective function takes a brute force approach; the feature parameters are 261
used to create a training set which is then used to train an unfolding neural network. This 262
is done four times, to create four separate unfolding networks trained on four distinct sets 263
of randomly generated neutron energy spectra with the same feature parameters. The 264
cost function returns the average mean squared error unfolding performance of these four 265
networks when evaluated on IAEA neutron energy spectra. 266

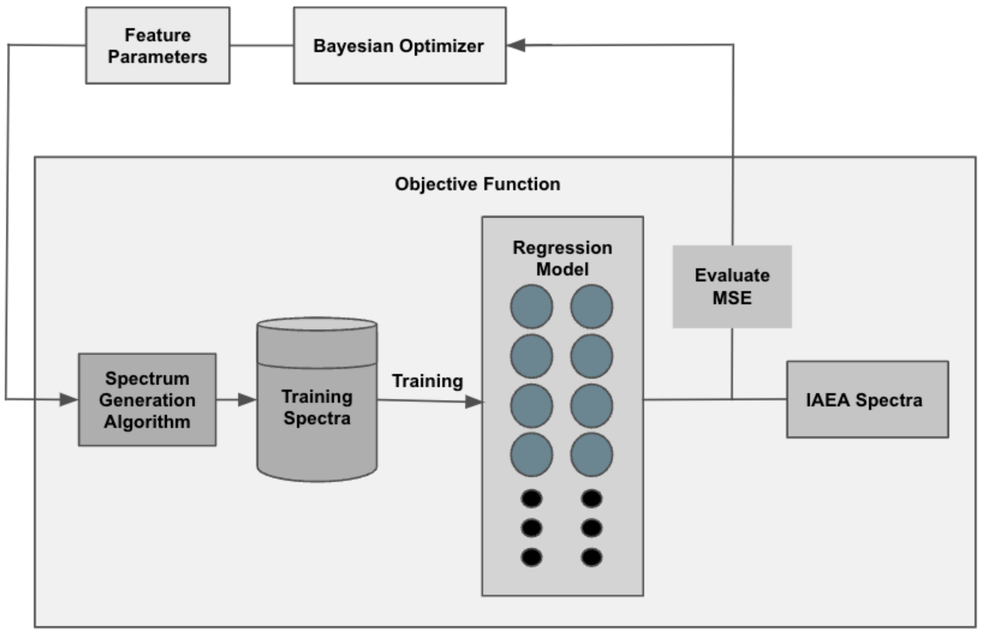

Figure 7. The process for tuning the feature parameters of a spectrum generation algorithm. The regression model may be a neural network (Section 2.3.1) or a linear regression model (Section 2.3.2).

2.3.2. Feature Parameter Tuning by Evaluation on Linear Regression Models 267

For the second objective function, the unfolding neural network is replaced with a linear regression model,

$$\phi(E_i) = w_{ij}\, d_j, \tag{15}$$

where w_{ij} is a matrix which maps the jth detector response bin to the ith energy bin in a neutron energy spectra.

Although far simpler than a neural network, linear regression models still involve a training step in which the model weights w_{ij} are tuned to minimize unfolding error against a training set. We do not expect a linear regression model to perform as well as an unfolding network, however both types of regression will still benefit from a properly constrained solution space - just like a neural network, a linear regression model which has been trained on realistic neutron energy spectra will far outperform one which has been trained on purely random data when unfolding real-world spectra.

The second objective function evaluates the effectiveness of a given set of feature parameters by training four linear regression models on four distinct sets of neutron energy spectra, randomly generated from the given feature parameters. The objective function then returns the mean squared unfolding error of each linear regression model when evaluated on the IAEA data set.

3. Results

When attempting to evaluate the effectiveness of a spectrum generation algorithm for the purposes of training an unfolding neural network, there is a great deal of inherent uncertainty. This uncertainty comes primarily from the randomly initialized neuron weights and biases, which can lead two separate networks trained on identical data to find two completely different loss function minima. In order to account for this uncertainty when evaluating a given spectra generation algorithm, an ensemble of thirty networks with identical architecture is trained on the same 3000 randomly generated spectra. Analysis of mean squared unfolding error and standard deviation unfolding error is performed on individual neural networks and then averaged across the thirty distinct networks.

3.1. *Optimal Feature Parameters for* ***PSA****,* ***GAUSS-1****, and* ***GAUSS-2***

The feature parameters of **GAUSS-1** and **GAUSS-2** were tuned by means of Bayesian optimization, with the neural network and linear regression objective functions both utilized. All objective function optimizations were performed over the same computation time. When compared to the neural network objective function, Bayesian optimization performed on the linear regression objective function was able to search much more of the parameter space since evaluations of its objective function were dramatically quicker. This yielded two separate optimal feature parameters, according to each objective function. Table 3 shows the average unfolding performance for networks trained on the **GAUSS-1** and **GAUSS-2** algorithms, for both sets of optimal feature parameters, when evaluated on IAEA spectra.

Spectra Generation	Feature Parameter Tuning Method	MSE
GAUSS-1	Neural Network	$3.42 \pm 6.11 * 10^{-4}$
GAUSS-1	Linear Regression	$2.58 \pm 4.89 * 10^{-4}$
GAUSS-2	Neural Network	$2.51 \pm 4.51 * 10^{-4}$
GAUSS-2	Linear Regression	$2.14 \pm 4.16 * 10^{-4}$

Table 3. The mean squared unfolding error, averaged across thirty separate neural networks, all trained on the same spectra generation algorithm and evaluated on the IAEA real-world neutron energy spectra.

For both **GAUSS-1** and **GAUSS-2** the feature parameters found through optimization of the linear regression objective function gave better unfolding performance, when training a neural network to unfold IAEA spectra, compared to the feature parameters found through optimization of the neural network objective function. For the rest of our analysis, unless otherwise specified the feature parameters found through linear regression will be used due to their superior performance.

3.2. *Unfolding Performance of Each Algorithm*

Table 4 shows the average mean squared unfolding error for networks trained on each of the five spectra generation algorithms, when evaluated on IAEA spectra.

Spectra Generation	MSE
PSA	$1.74 \pm 3.07 * 10^{-4}$
GAUSS-1	$2.58 \pm 4.89 * 10^{-4}$
GAUSS-2	$2.14 \pm 4.16 * 10^{-4}$
FRUIT	$1.92 \pm 4.27 * 10^{-4}$
RAND	$5.83 \pm 8.39 * 10^{-4}$

Table 4. The averaged mean squared unfolding error of thirty separate neural networks, all trained on the same spectra generation algorithm, when evaluated on IAEA neutron energy spectra.

Figures 8, 9, 10, 11, and 12 show the three IAEA spectra which were unfolded with the least MSE and the three IAEA spectra which were unfolded with the most MSE, for neural networks trained on the the **PSA**, **RAND**, **GAUSS-1**, **GAUSS-2**, and **FRUIT** algorithms respectively.

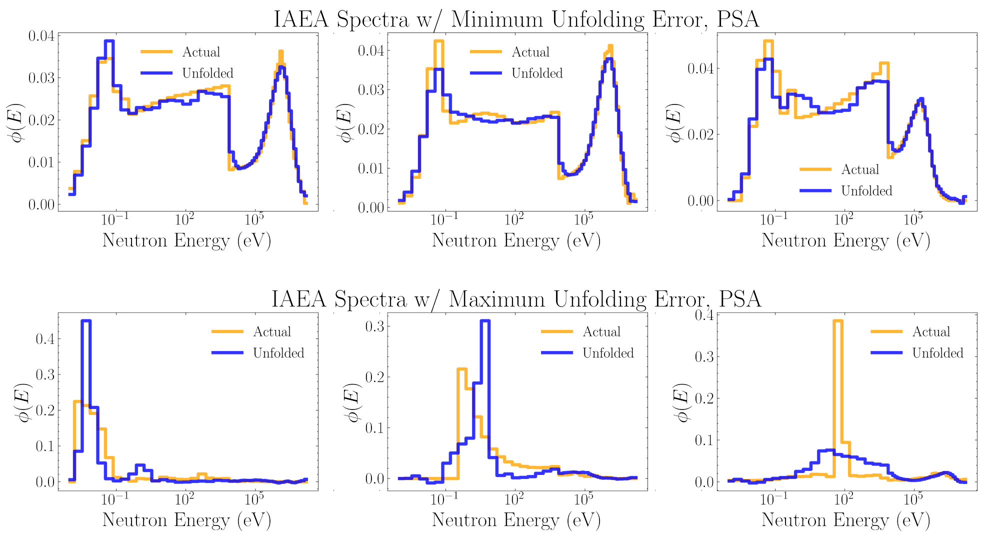

Figure 8. The three IAEA spectra which were unfolded with the least MSE (top) and most MSE (bottom) for neural networks trained on **PSA** data. Each plotted unfolded spectrum is the average of the spectra unfolded by the thirty distinct neural networks trained on the same data.

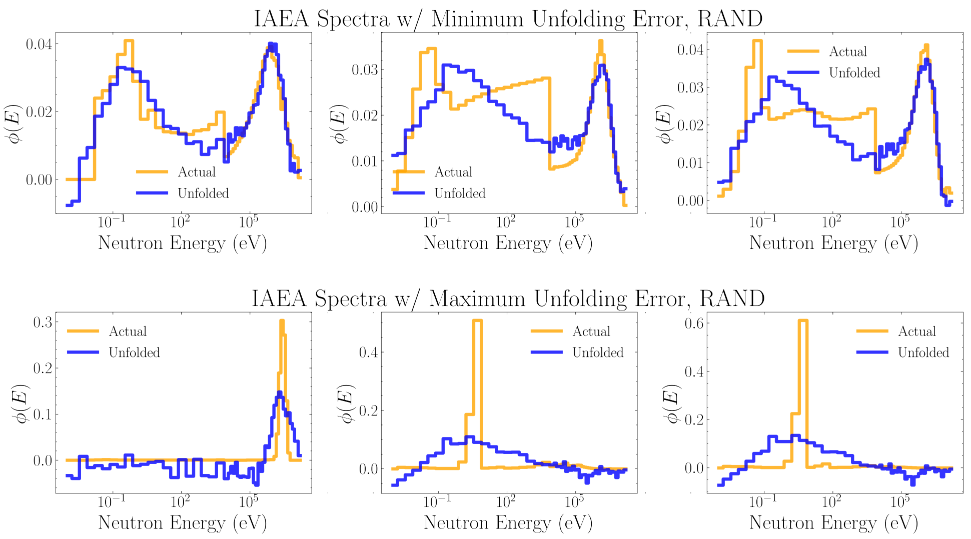

Figure 9. The three IAEA spectra which were unfolded with the least MSE (top) and most MSE (bottom) for neural networks trained on **RAND** data. Each plotted unfolded spectrum is the average of the spectra unfolded by the thirty distinct neural networks trained on the same data.

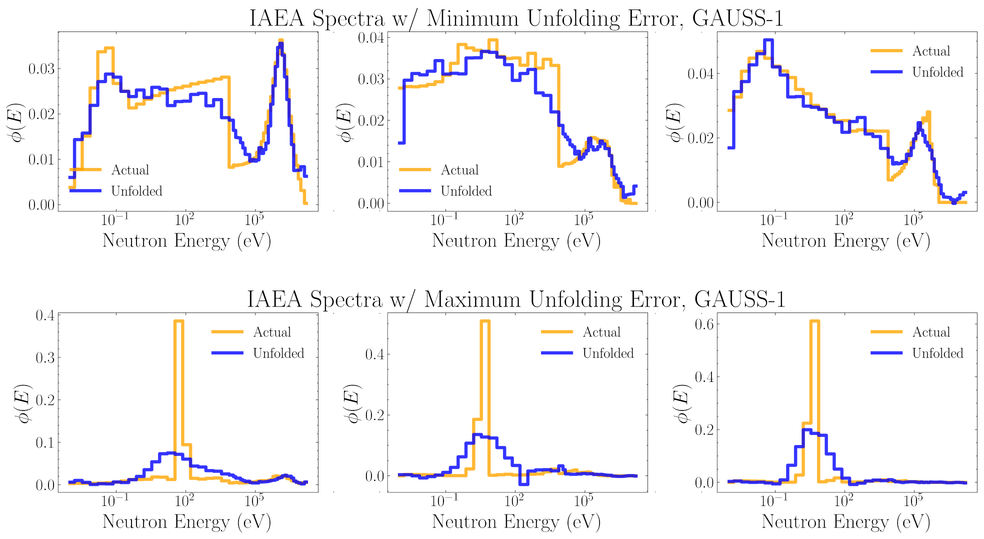

Figure 10. The three IAEA spectra which were unfolded with the least MSE (top) and most MSE (bottom) for neural networks trained on **GAUSS-1** data. Each plotted unfolded spectrum is the average of the spectra unfolded by the thirty distinct neural networks trained on the same data.

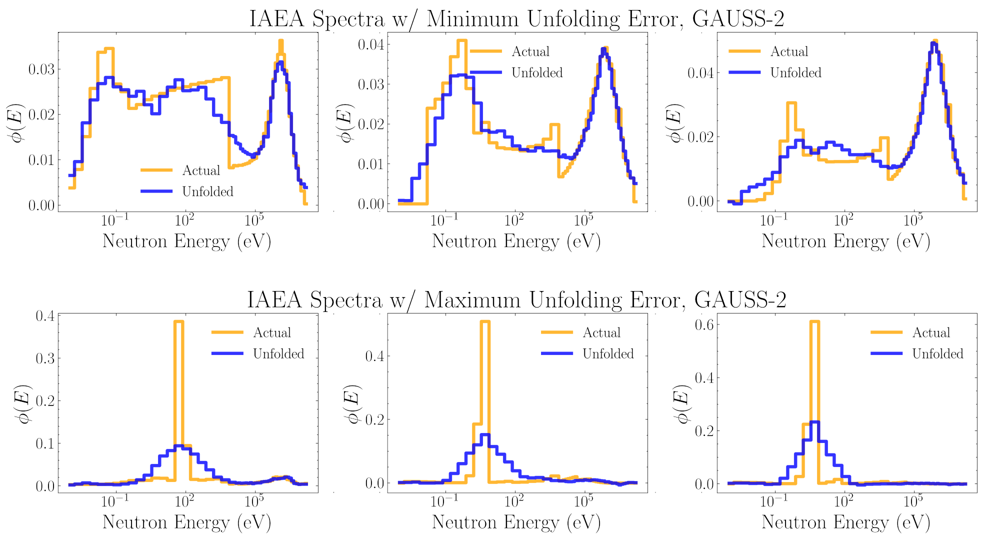

Figure 11. The three IAEA spectra which were unfolded with the least MSE (top) and most MSE (bottom) for neural networks trained on **GAUSS-2** data. Each plotted unfolded spectrum is the average of the spectra unfolded by the thirty distinct neural networks trained on the same data.

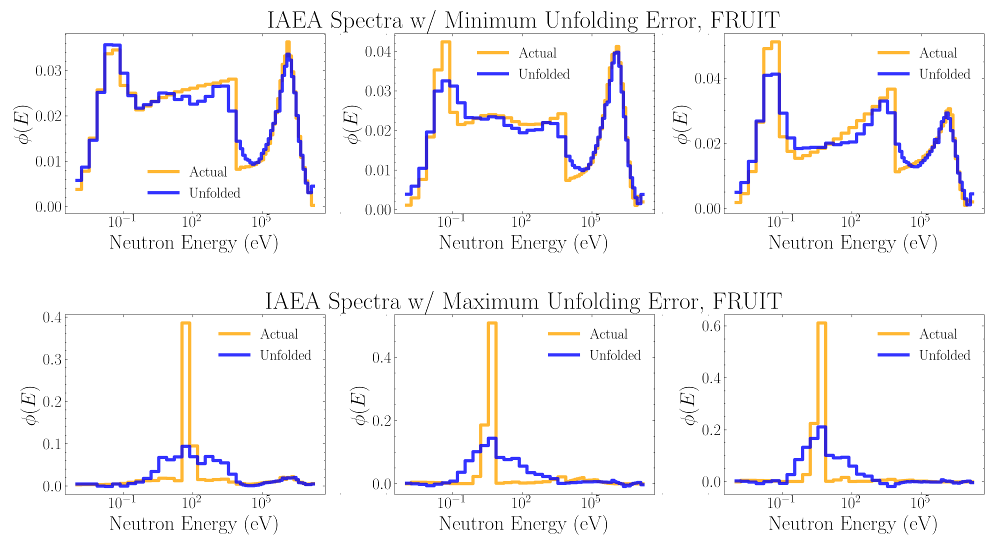

Figure 12. The three IAEA spectra which were unfolded with the least MSE (top) and most MSE (bottom) for neural networks trained on **FRUIT** data. Each plotted unfolded spectrum is the average of the spectra unfolded by the thirty distinct neural networks trained on the same data.

The IAEA neutron energy spectra set contains several kinds of commonly encountered 316
neutron sources, such as fission, fusion, high energy, etc. Some of these sources also include 317
moderators, giving rise to lower energy thermalized neutrons. Figure 13 shows the mean 318
squared unfolding error of Table 4, sorted according to the kind of spectra being unfolded. 319

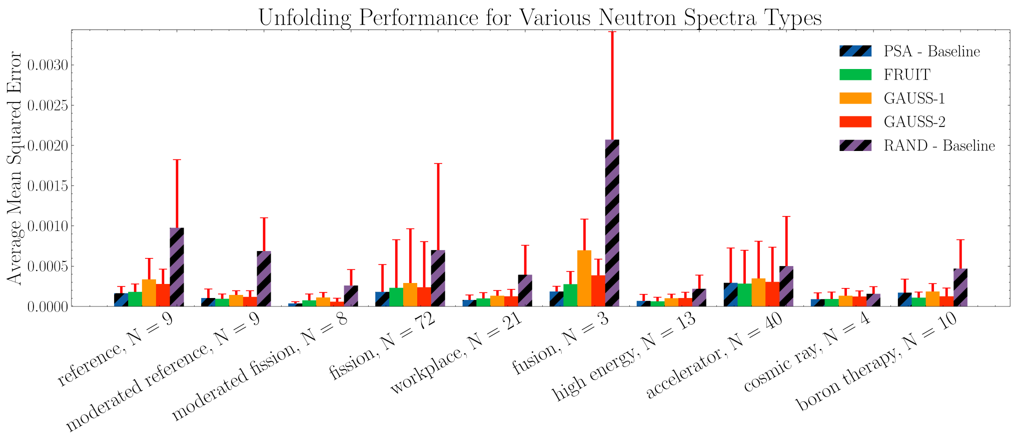

Figure 13. The mean squared unfolding error for neural networks trained on each of the five spectra generation algorithms, when evaluated on IAEA spectra of a given kind. The red line shows the standard deviation squared unfolding error across spectra of that kind, and N is the number of spectra of that type present within the IAEA evaluation data.

3.3. Unfolding Accuracy for Radiation Dosimetry 320

For some applications, such as radiation dosimetry, the precise shape of the unfolded spectra is not the primary objective. Although an unfolded spectrum might oscillate around the true neutron energy spectrum, when this spectrum is convoluted with equivalent dose weighting factors these oscillations may average out yielding an accurate dose estimate. To perform this analysis, neutron group fluence to ambient dose equivalent coefficients d_i provided by the IAEA technical report [8] were used. The calculated dose equivalent H^* is the dose equivalent in soft tissue at a depth of 10 mm. The neutron group fluence φ_i for a given neutron energy spectrum bin $\phi(E_i)$ is calculated according to

$$\varphi_i = (ln(E_{i+1}) - ln(E_i))\ \phi(E_i). \tag{16}$$

The ambient dose equivalent is then calculated as

$$H^*(10\text{mm}) = n \sum_i \varphi_i \, d_i, \tag{17}$$

where n is a scaling factor which depends upon the emission rate of the neutron source. For analysis only the percent error in dose is considered, and so n does not affect the results.

Table 5 shows the percent error between the dose calculated by the unfolded spectra and the dose calculated using the known neutron energy spectra. The average and standard deviation in percent dose error across all IAEA spectra for a given unfolding network is calculated and then this value is averaged over all thirty networks.

Spectra Generation	Percent Error (%)
PSA	7.9 ± 9.2
GAUSS-1	11.6 ± 10.9
GAUSS-2	8.3 ± 7.6
FRUIT	7.1 ± 7.0
RAND	60.8 ± 84.1

Table 5. The averaged percent error in dose prediction, across thirty separate neural networks all trained on the same spectra generation algorithm, when evaluated on IAEA neutron energy spectra.

Figure 14 shows the mean percent error in dose prediction, sorted according to the kind of spectra being unfolded.

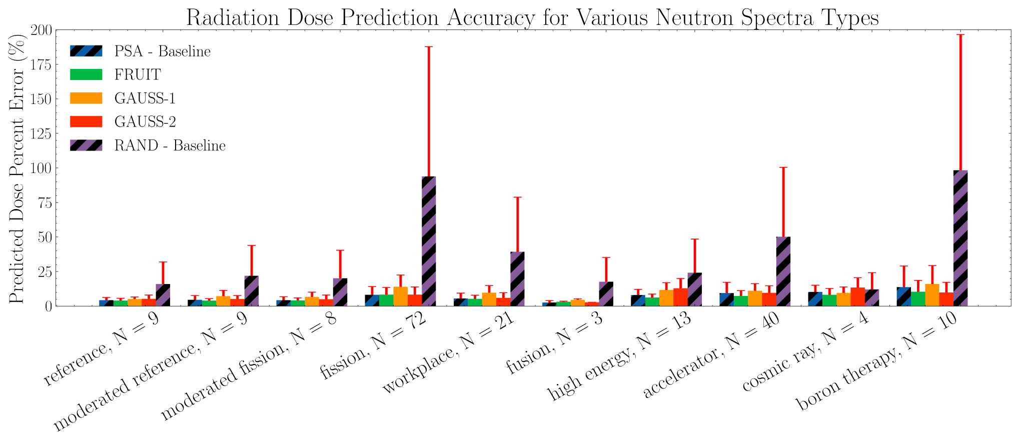

Figure 14. The averaged percent error in radiation dose determined from unfolded neutron energy spectra for neural networks trained on each of the five spectra generation algorithms and evaluated on IAEA spectra of a given kind. The red lines show one standard deviation in dose error and N is the number of spectra of that type present within the IAEA evaluation data.

4. Discussion

The results shown in Table 3 demonstrate that linear regression models may be used as an effective proxy when performing feature engineering on the training data of a neural network. Although a neural network based objective function will provide the most accurate metric of expected unfolding performance, it will also be extremely computationally expensive. Each objective function evaluation will involve fully training multiple unfolding networks, in order to account for deviations between networks, which can take several minutes. Since an objective function is called several hundred times or more while searching for a minimum, this will prevent any optimization algorithm from performing an extensive search across the parameter space in a reasonable time frame. The linear regression objective function sacrifices accuracy in calculating expected mean squared unfolding error in order to dramatically reduce computational complexity. This allows optimization algorithms such as Bayesian optimization to perform a much wider search over the parameter space.

Although the kinds of neural networks used for unfolding and the specific architecture of those networks vary widely throughout the literature, this work was performed in the hope that our optimal training spectra would remain optimal across a wide range of regression models. There are several properties, such as the hyper-parameters of an unfolding network, which are highly specific to the exact neural network architecture being employed. However, the fact that the optimal feature parameters for a linear regression model remained optimal when being used to train a neural network provides context to the potential for the same spectra generation algorithms outlined in this work to be employed across a wide variety of network architectures. However, when unfolding onto a different energy bin structure we are far less certain that these spectra generation algorithms would be optimal. This is why procedures for feature parameter tuning have been discussed - so that this work may be reproduced for other energy ranges and resolutions of interest.

In Table 4 we can see that the **PSA**-trained networks and the **RAND**-trained networks had the least and greatest mean squared unfolding error respectively. This is not surprising, as these algorithms were intended as upper and lower bounds on the degree of solution space constraint. On average the mean squared unfolding error of the **PSA**-trained networks, with their properly constrained training domain, was more than three times less than the **RAND**-trained networks. In addition, when using these unfolded networks for dosimetry purposes (Table 5), the **RAND**-trained networks gave dose estimates with more than 7 times as much error as the **PSA**-trained networks. This provides a rough sense of scale to the difference in unfolding performance expected between a network trained on physical vs non-physical spectra.

It is important to note, however, that mean squared error is a flawed metric for evaluating unfolding performance which undervalues how much more valuable the **PSA**-unfolded spectra would be when used for dosimetry or source characterization. This is illustrated when comparing the best-case and worst-case **PSA**-trained unfolding (Figure 8) to that of the **RAND**-trained unfolding (Figure 9). We can see that **RAND**-trained networks were incapable of confidently unfolding important qualitative features of the energy spectra. If these unfolded spectra were to be used for source characterization, even the best-case unfolding would be unable to correctly identify the radiation source. On the other hand, the worst-case unfoldings for the **PSA**-trained networks were still able to identify the rough shape and energy scale of structures within the spectra. Although the average mean squared unfolding error of networks trained on these upper and lower bound baselines of training domain constraint differ by only a factor of three, within the context of unfolding use cases a properly constrained training domain can mean the difference between highly accurate or entirely useless unfolding.

Table 4 also shows that the **GAUSS-2**-trained networks had 20% less mean squared unfolding error on average than the **GAUSS-1**-trained networks. This difference in performance may also be seen in Table 5, where **GAUSS-2**-trained networks predicted doses with 30% less mean percent dose error than **GAUSS-1**-trained networks. This trend is likely due to the heavier constraints which the **GAUSS-2** algorithm places on the training domain through coupling the width and height of the added Gaussian peaks to the magnitude of their central energy. This means that lower energy Gaussian peaks will be smaller in amplitude and broader in width, resembling the physics of thermalized neutrons. In particular, we expected that this would translate into better unfolding performance of **GAUSS-2**-trained networks on moderated sources or spectra with more thermal neutrons present. This is partially demonstrated in Figures 13 and 14, where the **GAUSS-2**-trained networks have less mean squared unfolding error and percent dose error than the **GAUSS-1**-trained networks when evaluated on fission, moderated fission, and moderated reference spectra. On the other hand, **GAUSS-2**-trained networks see a decrease in unfolding performance compared to **GAUSS-1**-trained networks for spectra with no thermalization such as high energy and cosmic ray. The dosage predictions for these spectra types from **GAUSS-2**-

trained networks are worse than those from **GAUSS-1**-trained networks, and occasionally worse than **RAND**-trained networks. This difference in performance is likely because the **GAUSS-2** algorithm attempts to enforce low-energy neutron thermalization, thus poorly representing any non-thermal low energy structures. However, these differences in unfolding performance are not dramatic.

Out of the three spectra generation algorithms to be investigated, the unfolding neural networks trained on **FRUIT** data performed the best, on aggregate, in both mean squared unfolding error (Table 4) and dosage estimation (Table 5). In fact, the dosage estimates of the **FRUIT**-trained networks were more accurate on average than the estimates made by **PSA**-trained networks, even though **PSA** was intended to be an upper-bound on performance. Since the **FRUIT** algorithm generates spectra from one of four theoretical models - fission, evaporation, high energy, and Gaussian - we would expect networks trained on it to perform particularly well when unfolding real spectra of these categories. On the other hand, **FRUIT**-trained networks should not perform nearly as well when unfolding accelerator, cosmic ray, or medical spectra as these are not modeled for by the algorithm. However, this trend is not seen in Figures 13 or 14, and the **FRUIT**-trained networks seem to be capable of generalizing to many types of neutron spectra.

Certain trends in the mean squared unfolding error across spectra types align with what we would anticipate from the design of each spectra generation algorithm. However, there are no spectra types for which the mean squared unfolding error of networks trained on a specific algorithm clearly stands out. Instead, it seems that there are particular kinds of spectra which are generally easier to unfold with a lower mean squared error. Looking at Figures 8-12, unfolding mean squared error was universally lower for spectra which consisted of a single peak. Due to the normalization constraint, these single-peaked spectra necessarily have an extremely high amplitude. If the unfolding networks do not match the small number of large amplitude bins, even if they are correct about the rough energy range, the MSE will be very large due to the squaring of the residuals. On the other hand, a large number of small discrepancies within spectra consisting of multiple structures, such as for the best-case unfolding of **RAND**-trained networks (Figure 9), will contribute much less to the MSE even if the quality of the unfolding is very poor. This could be avoided by using other loss functions, such as mean absolute error, which do not penalize outliers as heavily. However, due to the wide-spread use of MSE as a metric for spectrum unfolding, and to facilitate easy comparison of our unfolding performance to other works, we have chosen to use MSE in our analysis.

5. Conclusions

In this paper we presented three methods for the random generation of large quantities of realistic and physically motivated neutron energy spectra, without the need for expensive experiment or Monte-Carlo simulations. This work was based on the observation that, regardless of the specifics of an unfolding network's architecture, the performance and generalizability of any neural network based unfolding algorithm would be dramatically impacted by the quality and quantity of neutron energy spectra used to train it.

Our analysis was performed on 189 of the 251 IAEA neutron energy spectra (with the other 62 spectra being used by the **PSA** algorithm), which were quarantined during the training of the unfolding networks in order to prevent any data leakage. We demonstrated that these neutron energy spectra generation algorithms far outperformed random training data which were not physically motivated. Out of the three spectra generation algorithms investigated, neural networks trained on the **FRUIT** spectrum generation algorithm (section **??**) had the greatest overall unfolding performance on IAEA spectra. This is because it relied on theoretical models which are able to accurately capture the physics of fission, high energy, and evaporative spectra - all of which are present in large quantities in the IAEA data. Building off of the **FRUIT** algorithm presented in this paper, future work could improve the generalizability of networks trained on **FRUIT** spectra by constructing more theoretical models for a wider range of spectra types and performing further analysis on how to optimally select parameters for each model.

We also determined that linear regression models could be used in proxy of an unfolding neural network to evaluate the performance of a particular set of training data. The technique outlined in this paper may be applied to the feature engineering stage of future work on neural network based unfolding, with different energy ranges of interest or radiation types.

All code and relevant analysis have been made publicly available. Our hope for this work is that any project investigating neural network based neutron spectrum unfolding, regardless of the neural network architecture or kind of detector used, will be able to insert similar algorithms into the training data generation step in order to achieve optimal unfolding performance on the problem under investigation.

Author Contributions: Conceptualization, James McGreivy, Juan Manfredi and Daniel Siefman; Data curation, Daniel Siefman; Formal analysis, James McGreivy; Investigation, James McGreivy, Juan Manfredi and Daniel Siefman; Methodology, James McGreivy, Juan Manfredi and Daniel Siefman; Software, James McGreivy and Daniel Siefman; Supervision, Juan Manfredi and Daniel Siefman; Visualization, James McGreivy; Writing – original draft, James McGreivy; Writing – review editing, James McGreivy, Juan Manfredi and Daniel Siefman.

Funding: This material is based upon work supported in part by the Department of Energy National Nuclear Security Administration through the Nuclear Science and Security Consortium under Award Numbers DE-NA0003180 and DE-NA0003996. This material has been approved for release by LLNL under document number LLNL-JRNL-842794.

Data Availability Statement: The relevant code may be found at https://github.com/JamesMcGreivy/NeutronSpectraGeneration.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

IAEA	The International Atomic Energy Agency
MSE	The Mean Squared Error Loss Function

References

1. Glodo, J.; et al. New Developments in Scintillators for Security Applications. *Physics Procedia* **2017**, *90*, 285–290. https://doi.org/10.1016/j.phpro.2017.09.012.
2. Stange, S.; et al. A fissionable scintillator for neutron flux monitoring. In Proceedings of the SPIE Proceedings; Grim, G.P.; Schirato, R.C., Eds. SPIE, 2011. https://doi.org/10.1117/12.894708.
3. Mohammadi, N.; Hakimabad, H.M.; Motavalli, L.R. Neural network unfolding of neutron spectrum measured by gold foil-based Bonner Sphere. *Journal of Radioanalytical and Nuclear Chemistry* **2014**. https://doi.org/10.1007/s10967-014-3650-8.
4. Ortiz-Rodriguez, J.M.; Reyes Alfaro, A.; Reyes Haro, A.; Solis Sanches, L.O.; Miranda, R.C.; Cervantes Viramontes, J.M.; Vega-Carrillo, H.R. Evaluating the performance of two neutron spectrum unfolding codes based on iterative procedures and artificial neural networks. *AIP Conference Proceedings* **2013**, *1544*. https://doi.org/10.1063/1.4813468.
5. Sharghi Ido, A.; Bonyadi, M.; Etaati, G.; Shahriari, M. Unfolding the neutron spectrum of a ne213 scintillator using artificial neural networks. *Applied Radiation and Isotopes* **2009**, *67*, 1912–1918. https://doi.org/10.1016/j.apradiso.2009.05.020.
6. Kelley, R.P.; Rolison, L.M.; Lewis, J.M.; Murer, D.; Massey, T.N.; Enqvist, A.; Jordan, K.A. Neutron response function characterization of 4He scintillation detectors. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2015**, *793*, 101–107. https://doi.org/https://doi.org/10.1016/j.nima.2015.04.011.
7. Febbraro, M.; Becker, B.; deBoer, R.; Brandenburg, K.; Brune, C.; Chipps, K.; Danley, T.; Fulvio, A.D.; Jones-Alberty, Y.; Macon, K.; et al. Performance of neutron spectrum unfolding using deuterated liquid scintillator. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2021**, *989*, 164824. https://doi.org/https://doi.org/10.1016/j.nima.2020.164824.
8. Compendium of Neutron Spectra and Detector Responses for Radiation Protection Purposes. Technical report, 2001.

9. Bai, H.; Wang, Z.; Zhang, L.; Jiang, H.; Lu, Y.; Chen, J.; Zhang, G. Simulation of the neutron response matrix of an EJ309 liquid scintillator. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2018**, *886*, 109–118. https://doi.org/https://doi.org/10.1016/j.nima.2017.12.072.
10. Agostinelli, S.; Allison, J.; Amako, K.; Apostolakis, J.; Araujo, H.; Arce, P.; Asai, M.; Axen, D.; Banerjee, S.; Barrand, G.; et al. Geant4—a simulation toolkit. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2003**, *506*, 250–303. https://doi.org/https://doi.org/10.1016/S0168-9002(03)01368-8.
11. Goorley, J.T.; James, M.R.; Booth, T.E.; Brown, F.B.; Bull, J.S.; Cox, L.J.; Durkee, Jr., J.W.; Elson, J.S.; Fensin, M.L.; Forster, III, R.A.; et al. Initial MCNP6 Release Overview - MCNP6 version 1.0 **2013**. https://doi.org/10.2172/1086758.
12. Bedogni, R.; Domingo, C.; Esposito, A.; Fernández, F. Fruit: An operational tool for multisphere neutron spectrometry in workplaces. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2007**, *580*, 1301–1309. https://doi.org/10.1016/j.nima.2007.07.033.
13. Mukherjee, B. A high-resolution neutron spectra unfolding method using the Genetic Algorithm technique. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **2002**, *476*, 247–251. Int. Workshop on Neutron Field Spectrometry in Science, Technolog y and Radiation Protection, https://doi.org/https://doi.org/10.1016/S0168-9002(01)01440-1.
14. Hosseini, S.A. Neutron spectrum unfolding using artificial neural network and modified least square method. *Radiation Physics and Chemistry* **2016**, *126*, 75–84. https://doi.org/https://doi.org/10.1016/j.radphyschem.2016.05.010.
15. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* **2014**, *15*, 1929–1958.
16. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms, 2012. https://doi.org/10.48550/ARXIV.1206.2944.