Shallow Neural Networks as Particle Classifiers for the CHANDLER Neutrino Detector

Paul Rose[1]

[1] *Bard College at Simon's Rock*

The use of neural networks to classify detector events by particle type and evaluate the strength of variables as classifiers is explored. Two types of perceptron network are trained on a computer-simulated monte carlo dataset that identifies the event as a neutron or a neutrino, then gives a list of data values for the event. The first network– which optimized a single hyperplane cut– achieved a significance of 100, and the second network– a more sophisticated model with four ReLu processing neurons– achieved a significance of 146, outperforming the decision tree used previously for the CHANDLER detector. Subsequently, separate iterations of these networks were trained on 1-gamma and 2-gamma events to allow them to isolate features individual to these classes of events. This procedure found that 1-gamma events were classified very poorly by both networks, so a new variable that measures the escape probability of the second gamma was introduced, achieving a small improvement at selecting for 1-gamma IBDs. An alternative method of separating 1-gamma IBDs from 2-gamma IBDs was also found through the analysis of 2-d histograms, which improved the classification rate further. Finally, a new reward function that optimized for significance directly was introduced to train the neural network, greatly reducing the amount of manual tuning and re-learning needed to train an effective network. The combination of all of these achieved a significance of 170, outperforming all prior classification methods.

I. INTRODUCTION

Neutrino detection is a difficult endeavor in experimental physics. The only standard model force these particles interact through is the weak nuclear force, which has an extremely short range, meaning that detectable neutrino interactions are exceptionally rare. Thus, most neutrino detectors are buried deep underground to eliminate cosmic ray backgrounds that would otherwise swamp the neutrino signal. However, the MiniCHANDLER detector, which weighs only 80 kg and can be moved anywhere by a small trailer [1], can discern the electron antineutrino spectrum from a nuclear fission reactor without any overburden to eliminate cosmic ray and other backgrounds. The analysis of this neutrino spectrum can discriminate between uranium fission reactions used for civilian energy and plutonium breeding used for nuclear weapons manufacturing without any foreknowledge of what went into the reactor. Thus, improved iterations of MiniCHANDLER– like the 1-ton CHANDLER detector currently in development– have the potential to revolutionize nonproliferation diplomacy.

The shieldless detection of neutrinos is possible through specialized detector hardware and data analysis procedures that leverage the particular signature of inverse beta decay (IBD), the interaction where an electron antineutrino collides with a proton yielding a positron and a neutron.

$$\overline{v}_e + p \rightarrow e^+ + n$$

The MiniCHANDLER detector has a 3-dimensional layered structure which vertically alternates between a lattice of plastic scintillator cubes and a thin neutron capture sheet. When an energetic charged particle ionizes molecules in the plastic scintillator, it gives off a short pulse of visible range photons. Conversely, when a neutron thermalizes in the detector, its capture on Lithium 6 in the neutron sheet results in the emission of a long pulse of photons which is easily discernible from the plastic scintillator signal. The light is then channeled down the row and column where the event occurred by total internal reflection in the lattice and detected by an array of photomultiplier (PMT) tubes. This channeling allows the positions of energy depositions to be reconstructed from PMT data. Because an IBD event produces a positron and a neutron, it will always be marked by a short plastic pulse followed by a nearby long neutron pulse. This allows for the elimination of background gamma rays, charged particles, and thermal neutrons, as these events display only one of the two components of an IBD event.

Fast neutrons from cosmic rays, however, pose a problem for this method of IBD discrimination. When a high-energy neutron enters the detector, it can collide with and scatter protons, which produce short pulses, and then capture in a neutron sheet, producing a long pulse. This can falsely mimic IBD coincidence and hamper the spectral analysis needed for nonproliferation applications of this detector. Thus, a method for discriminating between protons and positrons is needed. When a positron finishes depositing its energy in the detector, it annihilates with an electron and produces two anti-parallel 511 KeV gamma rays, whereas a proton quietly assimilates with the material and produces no gamma rays. These gamma rays can then Compton scatter in the detector, producing additional clusters of short pulses. To leverage this distinction between protons and positrons, an algorithm must be made that can discriminate between a cluster of energy depositions from Compton scattering and a cluster of energy depositions from scattered protons.

The CHANDLER team broke this algorithm into two

steps. First, the energy deposition data is processed into variables that capture important geometric information. Such variables include the distance between the positron candidate and the neutron capture, the energy deposited by the Compton cluster candidates, the distances between the positron candidate and the Compton cluster candidates, and the angle between the two Compton cluster candidates with the positron candidate as the vertex. These variables must then be run through a program that makes a decision on the event's identity. This second step is difficult for a human to program intuitively, as it involves the statistics of how correlated the different variables are to the event type. To evaluate these correlations, a machine learning algorithm can be trained on computer-simulated Monte Carlo data. While particle identification can never be perfectly accurate due to the huge variety of possible particle events, a machine learning algorithm trained on strong indicator variables is likely the best approach to dealing with this variety.

II. CLASSIFICATION BACKGROUND

A. Decision Trees vs. Neural Networks

Currently, the CHANDLER team uses a decision tree to optimize the separation of neutrons and IBDs. At each fork in the tree, a single variable is measured, and events whose value of that variable fall below the cut point go one way in the tree, and those with a value above the cut point go the other way [2]. Through a series of these cuts, the data is divided into many groups, and each of these groups is labeled as neutrons or IBDs. If the dataset is visualized as a many-dimensional hyperspace where the variables give the coordinates of events in that hyperspace, the decision tree cuts the space into many boxes, where the hyperplanes making up the sides of the boxes are orthogonal to the basis vectors of the space. Each box is then designated to contain neutron events or IBD events by the decision tree. Through the combination of many boxes, any more complicated identity boundary can be approximated. However, the approximation will have a toothed edge, where the corners of the boxes cut back and forth across the ideal division boundary. This means that many points near the boundary can be misclassified. A perceptron neural network, however, uses hyperplanes as the building block of decision boundaries [3]. These can take any orientation, unlike the decision tree boundaries which are limited to cuts perpendicular to basis vectors. This allows complex shapes to be approximated as polygonal solids, reducing the error near the boundary. Additionally, the hyperplanes don't have to be binary cuts- they can give a score to each event based on how far it is from the surface, resulting in a soft boundary that gives the probability of the event's identity. These many soft boundaries can be combined into rounded polygons that can effectively approximate identity boundaries.

B. Quantifying Classifier Performance: Significance

The metric used to evaluate the performance of a given classification method is based on the scenario where the CHANDLER detector is placed next to an array of modular reactors, and the classification method tries to determine whether the reactor closest to the detector has turned off. Significance is the square of the z-score of the null hypothesis that the reactor's operation has not changed. While this method of performance evaluation is somewhat arbitrary, it captures how well a classification method parses the IBD signal in an environment where noise vastly outnumbers signal. An evaluation of how well a classification method determines the energy spectrum would be truer to the end goal of this enterprise, but also far more difficult to quantify and optimize on in practice. As long as the classifier doesn't use the event's total energy as a determining condition, the significance parameter should correlate positively with good spectrum measurement. Significance can be calculated using the following formula:

$$significance = \frac{signal^2}{(signal + 2 * noise)}$$

Where signal is the number of correctly identified IBDs expected per 24 hours and noise is the number of neutrons incorrectly identified as IBDs per 24 hours. At the start of this process, the decision tree achieved a significance of 130, which means that a statistically significant difference could be detected within 45 minutes of a reactor unit's shutoff. This will serve as a benchmark to evaluate the success of neural networks as classification methods.

III. NEURAL NETWORKS AND 1 VS 2 GAMMA EVENTS

A. Conventionally Optimized Networks

In this analysis, neural networks were coded in Python using the NumPy library. The first network tested was a perceptron with 25 input neurons and one output neuron. This network classifies based on a single hyperplane cut through the dataset space- the input vector is dot multiplied by the weight vector and added to a bias. To optimize classification, the square of the difference between the event's identity (1 for IBDs and 0 for neutrons) and the predicted probability of being an IBD (float between 0 and 1) was minimized. This maximizes the percent accuracy of the network, but does not take into account the fact that false positives are far costlier than false negatives in a high noise environment. To account for this, the network was trained on 5 neutron events for every IBD event to skew it towards negative identification. This method achieved 92 significance, which is good for a single planar cut. To improve the performance, the redundancy and relevancy of variables was explored. The

25 variables used as input are educated guesses at what qualities of an event could indicate its identity, so they are not guaranteed to improve classification. To investigate the variables, 1000 1-hyperplane networks were trained on the dataset, and the mean distance of each weight from the average initialization value was divided by the standard deviation of the weight over the 1000 networks to obtain a z-score for the weight's training. If a given weight always trained to close to the same value, its corresponding variable is likely an important classifier. Conversely, if the weight is left close to the initialization value or is trained to a pseudo-random number, it is likely an unimportant or unclear classifier of variables. Upon running this test, two variables– capture time and cluster multiplicity– had by far the lowest z-scores, at 0.15 and 0.07 respectively. When these variables were removed from the training set, reducing the number of inputs to 23, the 1-plane perceptron improved to 100 significance. This makes sense because capture time is known to be a poor indicator of particle identity, and cluster multiplicity is redundant because it is the sum of cluster 1 multiplicity and cluster 2 multiplicity. However, when three other variables whose z-score ranged from 0.45 to 0.65 were removed, the network's performance worsened. Thus, these variables are likely weak but useful classifiers for the network. Possibly, the identity boundary in the space of these variables is sufficiently nonlinear that many different weight combinations are similarly effective classifiers.

The next perceptron had four processing neurons in between the inputs and outputs, each of which used a ReLu activation function. This network draws four hyperplanes in the dataset space, giving a positive score proportional to distance on one side of the hyperplane and a score of 0 on the other side. This allows each processing neuron to specialize on classifying some types of events and turn off for other types. This network achieved 146 significance, outperforming the decision tree by 16. Training the network was somewhat labor intensive, as to achieve optimum performance the network needed to be trained at a certain learning rate, saved, and then trained at a lower learning rate to fine-tune. This significance was achieved after four rounds of training at learning rates that were manually selected for optimum performance. The reduced set of 23 inputs was used, as the 25 input training performed worse on first training runs, indicating that the removed variables were poor classification contributors beyond the 1-plane network. Networks with 2 to 20 processing neurons were tried, but 4 neurons performed the best. This is likely because the crude inaccuracy-based method of training, which had to be manually slanted to maximize significance, failed to guide the more complex networks to a minimum due to the reduced proportion of IBD training data.

B. 1-Gamma 2-Gamma Sorting

For some events, both annihilation gammas Compton scatter in the detector. For others, however, one (or occasionally both) of the gammas escapes the detector without leaving an energy trace. The latter group of events is much harder to classify because it leaves less geometrical information in the detector. The set of particle events can be partitioned into 2-gamma and 1-gamma candidate groups based on the SoLid cluster algorithm, which draws a plane through the positron candidate that is perpendicular to the line connecting the highest energy Compton candidate with the positron candidate [4]. The barycenter of the hits on the max-energy side of the plane is calculated by taking the weighted average of their positions, and the line connecting this barycenter to the positron candidate defines the normal for the final plane. If energy is seen on both sides of this plane, both annihilation gammas could have scattered in the detector. If energy is only seen on one side of the plane, it is presumed that one of the annihilation gammas must have escaped if the event is a IBD. When the 146 significance parameters were tested individually on SoLid-classified 2 and 1-gamma events, they attained a significance of 138 on the 2-gamma events, but only reached 10.6 on 1-gamma events. This is because it is more likely for a neutron to mimic the geometry of a 1-gamma event than a 2-gamma event, as less information is available when one of the gammas escapes. As a result, high-energy neutrons outnumber IBDs by 118 to 1 in the 1-gamma regime, but only 19 to 1 in the 2-gamma regime. 2-gamma events also have two other parameters that can aid in classification– the distance between the positron candidate and the cluster 2 candidate and the angle between the two clusters with the positron at the vertex. Thus, it is much easier to parse signal from 2-gamma events than 1-gamma events.

Due to the differing geometry of 1-gamma and 2-gamma events, sorting the events into two separate datasets by this criterion and training separate networks on the two sets could allow the networks to specialize on particular qualities of the two categories. Additionally, because the 2-gamma set has two more variables than the 1-gamma set, training on the combined set required the substitution of dummy variables for 1-gamma events. However, pre-sorting did not immediately yield the expected success. The 1-plane perceptron performed about the same on 2-gamma events as on the combined set but classified abysmally on the 1-gamma set with significance no better than the unclassified set. The 4-plane ReLu perceptron performed about the same on 2-gamma events whether it was trained on just 2-gamma events or the combined set, but could not gain a foothold on the 1-gamma set, achieving only significances around 3. This is somewhat surprising, given that the network trained on the combined set achieved a significance of 10.6. One explanation is that 2-gamma and 1-gamma events have some similar distinguishing characteristics between neutrons and IBDs, but these identity boundaries are far

weaker classifiers in the 1-gamma dataset, making them harder for the network to find. If an identity boundary in a dataset is weak, the gradient forces guiding the network towards the identity boundary are weaker, making it more likely that the network becomes caught in false minima that are suboptimal. Thus, the presence of 2-gamma events could guide a network to a more optimal classification of 1-gamma events.

C. 1-Gamma Variable Attempts

A few attempts were made at creating additional geometric variables to distinguish 1-gamma IBDs from 1-gamma neutrons. The first variable was the angle between the line connecting the positron candidate and the highest Compton hit and the shortest path from the positron candidate to the face of the detector (Figure 1). This parameter assumes that the highest energy Compton hit is the first Compton scatter of the trapped gamma and that the missing gamma probably escaped perpendicular to the nearest face along the shortest path from the positron to the detector edge. A large angle indicates an IBD, as the missing gamma is more likely to be anti-parallel to the first. Another variable attempt measured the length of the path the second gamma would have to take to escape the detector (Figure 2). It was again assumed that the highest energy Compton hit was the first scatter.

FIG. 1. θ = escape angle

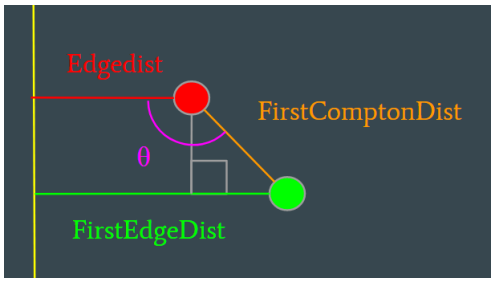
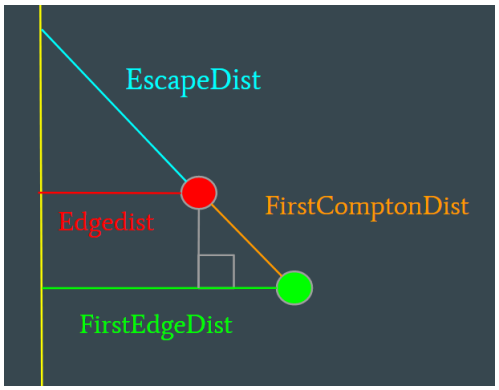
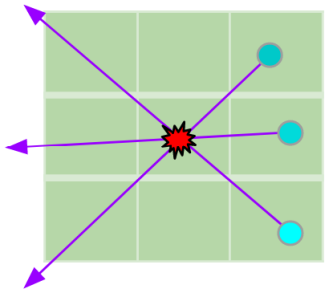


FIG. 2. Escape distance



When tested individually, optimal cuts to both variables improved the signal-to-noise ratio a small amount, but did not improve significance. This indicates that these variables are weakly correlated with 1-gamma event identity. Putting these variables into the training set did not improve the network performances, however. The limited success of these variables is likely caused by two factors– the highest energy compton scatter often isn't the first and the location of the hits within a scintillator cube is not known. These two factors make estimates of the paths of the first and second gammas imprecise, which smears estimates of how likely the second gamma is to escape. To deal with the first issue, an escape score was created that sums the probability of escape for all Compton hits. It is first assumed that the highest energy hit is the first Compton scatter, and then for each other hit the chance of that hit being the first scatter instead is calculated. If the max energy hit is not the first scatter, the gamma must have enough energy left after the actual first scatter to deposit the max energy. Thus, the maximum allowed scatter following the hit being tested can be compared to the actual maximum hit to obtain a first Compton probability. A second adjustment compared the expected scattering angle with the measured scattering angle to obtain another probability. The product of these two probabilities approximates the chance that the given hit instead of the max hit is the first scatter. Each hit's chance of being the first scatter is then multiplied by the chance a gamma traveling anti-parallel to that hit would escape. This was calculated by dividing each escape distance by the 10.6 cm attenuation length of 511 KeV gammas and raising 1/2 to the power of the result. Finally, these products were summed for all hits, and to obtain the final escape score, the sum was normalized by dividing by the sum of all first Compton probability scores. Thus, this parameter accounts for and weights all possible ways the second gamma could have escaped.

FIG. 3. Escape probability weighted by first Compton probability



This parameter performed somewhat better than the first two, with cuts improving the significance by about 3% and the signal/noise ratio by about 20%. Interestingly, the score calculated without the scattering angle adjustment boosted significance twice as much as the score calculated with this adjustment. This is because ac-

counting for scattering angle lowered the probability that low-energy hits could be the first hit, making it harder for an event to achieve a high score. This reduced the number of neutrons and IBDs that achieved a higher score, and because the prism size made angle uncertainty quite high, this reduction didn't improve the signal/noise ratio enough to make up for the overall loss of events. Below are escape probability histograms of 1-gamma events calculated with various methods. These events are also selected to have a best pairing angle below 90 degrees. This eliminates L-shaped 2-gamma events where the two first scatters occurred in perpendicular neighbor cubes, which the SoLid clustering algorithm often mistakes for 1-gamma events.

FIG. 4. IBD Escape Score: assuming highest energy scatter is first

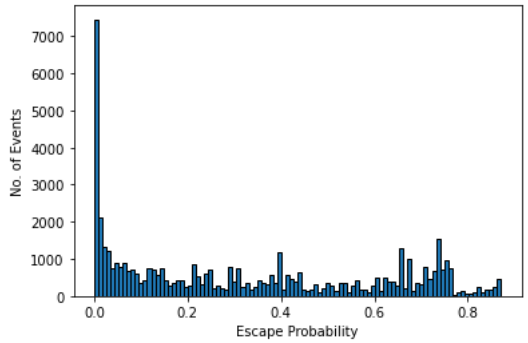
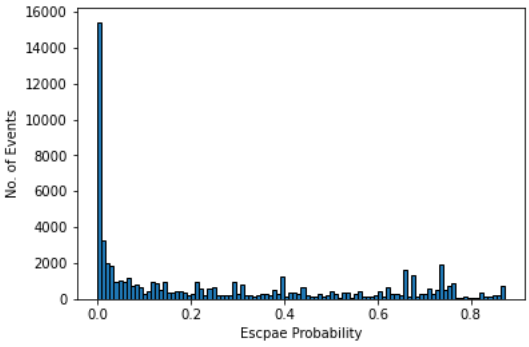


FIG. 5. Neutron Escape Score: assuming highest energy scatter is first Compton scatter



Although IBDs are more likely to be classified with a high escape score than Neutrons, many neutrons still make it into the high escape score regime, and many IBDs are classified with an escape score near 0. The former issue is likely due to the sheer number and variety of neutron noise events, which allows for a significant number to look like 1-gamma IBDs by random chance. The latter issue, however, is likely due to the uncertainty in event position. The detector can reconstruct which plastic prism an energy deposition occurred in, but is blind to the exact location within the prism. This allows for cases where the central approximation escape score

FIG. 6. IBD Escape Score: energy-based first Compton probability

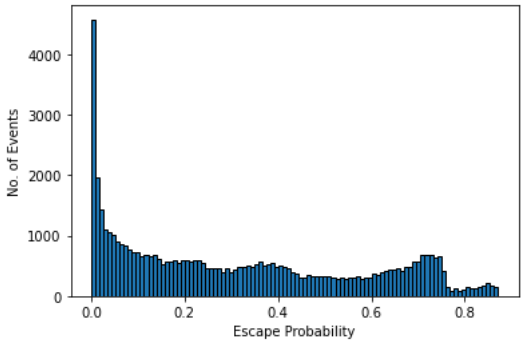
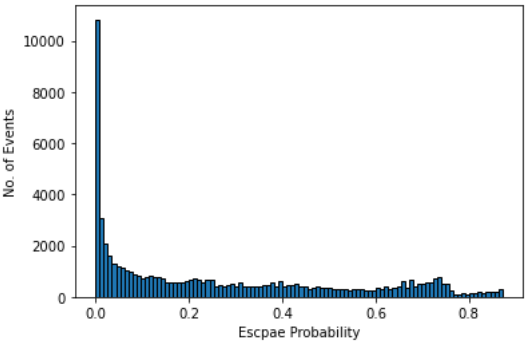


FIG. 7. Neutron Escape Score: energy-based first Compton probability



vastly underestimates the actual probability of escape.

Unfortunately, adding the parameter to the neural networks did not improve their 1-gamma performance. To improve the effectiveness of this variable, all possible locations of the positron and Compton hits in the prisms could be accounted for by averaging the escape probability of all possible paths through the two hit prisms with a 6-dimensional integral. This integral can be approximated numerically by simulating a large number of random location combinations in every possible pair of prisms and recording the average escape probability of the anti-parallel gamma in a dataset. This dataset could then be referenced whenever the escape score of a particular pair of prisms is needed. Because of the detector's planar symmetries, the positron hit would only have to be varied over one octant of the detector to capture all possibilities. Additionally, the number of events decays as the first Compton distance increases, and the number of random positional pairs that needs to be tried to accurately capture the escape score decreases as the angular variation of the path decays with distance.

Thus, the number of random positions simulated per prism pair could exponentially decay from 10000 for adjacent pairs to 1000 for pairs 10 cube lengths apart. A program with these cuts was written in Python, and using small sampling runs it was estimated that the full dataset would take 150 hours to create. This runtime could be

FIG. 8. IBD Escape Score: energy and angle-based first Compton probability

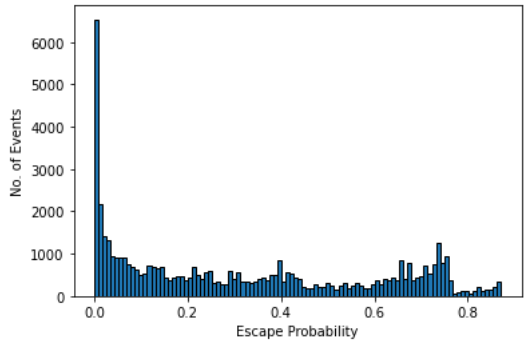


FIG. 9. Neutron Escape Score: energy and angle-based first Compton probability

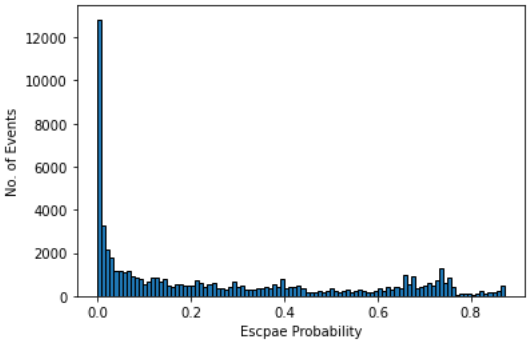


FIG. 10. A case where the center approximation predicts a much longer escape path

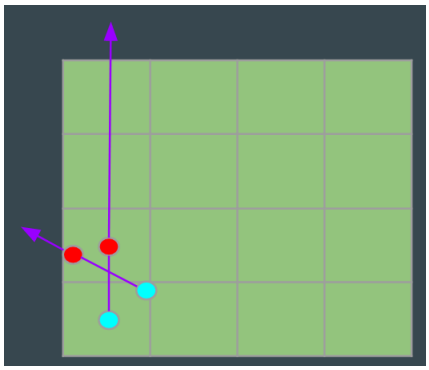
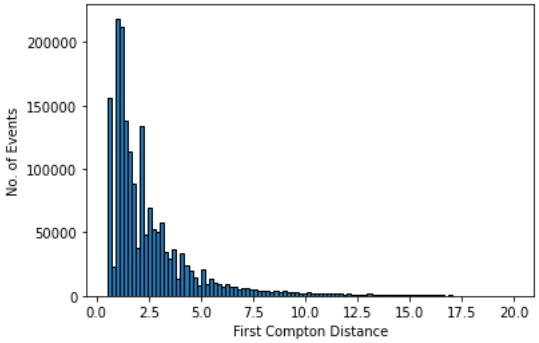


FIG. 11. Distance between Positron Candidate and First Compton Candidate



reduced by coding the program in a faster language and cutting the tested Compton hits further by how much the random average escape score is expected to differ from the center approximation escape score. The principal case in which these differ is one where positional variance within the prisms allows for a drastically shorter escape path, as depicted in figure 3. Thus, these cases- which are likely responsible for most of the 1-gamma IBDs given a very low escape score- could be isolated with a geometrical cut to further reduce computation time.

D. Significance Optimization

All of the networks thus far have been trained to minimize the sum of squares error between the particle identity and predicted probability, which incentivizes improving the percent accuracy. This is not the end goal however– the objective is to draw statistically significant conclusions about reactor operation in an extremely high-noise environment. Thus, optimizing the network for significance with this cost minimization method entails manually biasing the network towards negative classification. This crude method means finding parameters with good significance has a large element of luck, which makes training tedious. Additionally, it was found that networks with more than four input neurons did not train as well as the 4-plane network. This is likely because larger networks need more training data to perform well, and the artificial reduction of the IBD data's contribution to training hampered the larger networks' ability to discern distinguishing characteristics of IBDs. Thus, an optimization that maximizes significance directly could speed up the training process and allow the complexity of larger networks to be exploited fully.

Significance is based on the binary classification of events, so it is not a continuous function of the network parameters and cannot be optimized via gradient descent. To remedy this, a continuous version of significance was created based on the probability prediction from 0 to 1 for each event:

$$softsig = \frac{(at)^2}{(at + 2bf)}$$

t (for true positive) is the sum of all predicted probabilities for IBD events divided by the number of IBD events tested, and f (for false positive) is the sum of all predicted probabilities for neutron events divided by the number of neutron events tested. t and f are effectively continuous versions of the false positive proportion and true positive proportion used in the regular significance calculation. a and b are constants found by dividing the daily natural rate of neutrons and IBDs by the number of those events in the initial simulation set. These parameters convert

from the rate each type of events occur in the computer simulation data to the rate each type occur in real life. To maximize significance, the gradient function for the parameters needs to be calculated based on the partial derivative of softsig with respect to t and f. This calculation yields the following reward functions for signal and noise events:

$$R_S = a\frac{(at)^2 + 4atbf}{(at + 2bf)^2}$$

$$R_N = -2b\frac{(at)^2}{(at + 2bf)^2}$$

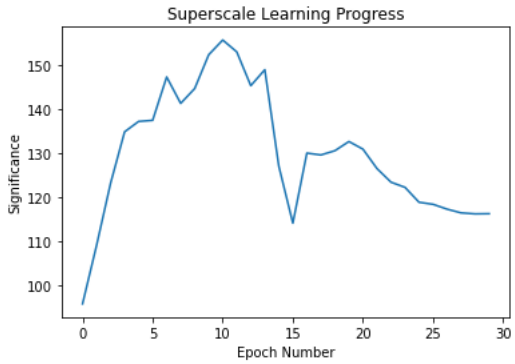
When a network trains in the traditional manner– to minimize a cost function– it reads through each example in a training batch and calculates how it needs to change the prediction for each example to reduce the cost. For the sum of squared errors cost, the derivative of cost with respect to predicted probability is given by:

$$dC = 2(Probability - Correct)$$

where Correct is 1 for IBDs and 0 for neutrons and Probability ranges from 0 to 1. This is the starting value for the back-propagation that follows. Thus, the degree to which error is reduced does not depend on event type. However, while augmenting signal and reducing noise are always good things, doing one compromises the ability for a network to do the other, so a proper balance between the two should be found. By taking into account the inherent rate disparity between signal and noise as well as the current values of signal and noise, softsig optimization finds the balance between augmenting signal and cutting noise that increases significance the most. To train network parameters with softsig, the network first has to read a training batch and calculate the signal and noise it reads from that batch, and from those calculate R_S and R_N. It then reads the batch again and adjusts parameters through backpropagation, using R_S as the starting value for IBD examples and R_N as the starting value for neutron examples.

When this method was first tested, the significance climbed quite quickly and then peaked and crashed. This is likely because the constant learning rate became too high for the more delicate steps required to fine-tune in later stages of training. Additionally, R_S and R_N tend to become higher for batches with higher significance and signal-to-noise ratio, so the change requests of worse batches likely got washed out. This means the network trained at the expense of the batches it needed to improve on most. To address this issue, the learning rate was multiplied by an exponential decay function of significance. This naturally reduces the learning rate as the network hones in on the minimum and prioritizes improvement on batches the network performs the worst on. Training a 4-plane ReLu network with these methods yielded a significance of 146 on 2-gammas alone, which was achieved in a single run with no manual retraining. This outperformed the previous 4-plane ReLu network which only achieved 138 on the 2-gamma set. Furthermore, a 20-plane network using an arctangent-based activation function achieved a significance of 156 in one training run on the 2-gamma set. This outperformed any other classification method used for the CHANDLER data. Even with the above tweaks, however, the significance on the validation set still crashed after the peak.

FIG. 12. The parameters achieved a peak significance and then wandered out of the maximum



When this was investigated, a bug was found in the Python code that failed to reset the parameter change arrays to 0 between training batches. This means that the reward gradient for each batch acted as the second derivative of the network parameters, not the first as previously thought. Thus, the hyperplanes possessed inertia as they moved through the state space, with gradient calculations from early in the training process having a continuing influence on how they moved. Additionally, a separate bug was found that inflated soft signal and deflated soft noise by a factor of 1.166, which skewed the calculation of R_S and R_N. When either one or both of these bugs were fixed, however, the network failed to train at all. No matter how high or low the learning rate was set, the network always became stuck in false maxima where all events were classified as IBDs. Thus, inertia and gradient warping were necessary for the network to get off the ground. Nonetheless, when the 156-sig parameters were trained with the bugs fixed, significance climbed to 169 for the 2-gamma set. As expected, the inertia and gradient warping bugs were responsible for the significance crash in the later stages of training. To train a new set of parameters of this caliber, the old code with inertia and warped gradient bugs must be used until the network exceeds approximately 150 significance, and then switched out for the fixed code to train it the rest of the way.

Training this network on the 1-gamma dataset, however, produced a significance of at most 3. Thus, it may be profitable to include 2-gamma events in the training set for the 1-gamma network because they may help guide it towards identity boundaries, as this network underperformed the 4-plane ReLu network from earlier. When the

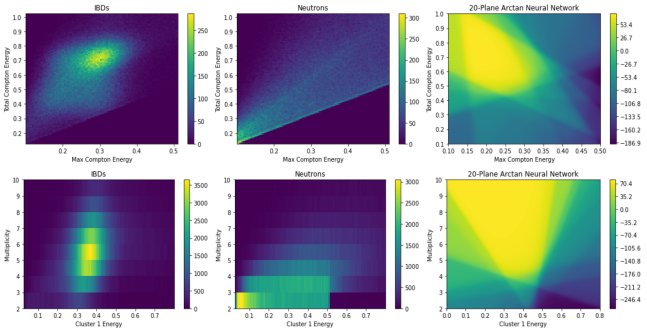
20-plane arctan network's 2-gamma results and 4-plane ReLu's 1-gamma results were merged, a significance of 174 was achieved. Notably, to achieve this gain, the final bias of the latter network had to be manually reduced. This is because significances only add when the signal-to-noise ratio of the two outputs are the same, and the addition becomes worse as the signal-to-noise ratio differs. To optimize data merging, the output bias of the 4-plane ReLu network was manually adjusted downward until the signal-to-noise ratio of the 1-gamma output was comparable to that of the 2-gamma output. In an attempt to optimize the cooperation of the two networks, the network training on 1-gamma events used the final soft signal and noise of the 2-gamma network added to its own current soft signal and noise in the reward function calculations in order to optimize the combined significance. This approach led nowhere, however, because the 1-gamma network trained in this way could not find a classification strong enough to contribute to significance, so it resorted to classifying all 1-gamma events as neutrons. Thus, the 4-plane ReLu network trained using crude cost minimization was still the best performer on 1-gamma events.

E. Justification for Neural Networks

A common criticism of neural networks is that they are black box classifiers– it just magically works, and it is impossible to know what is going on under the hood of the network. For neural networks with deep architectures, this is often the case. For example, when a network is trained to recognize handwritten digits, one may expect the first layer to pick up on small edges, the second to pick up on larger loops and lines, and the third to put those together into the digit's identity. When one looks at a weight map of a first layer neuron projected onto the writing space, however, it looks like a random, unintelligible mess [3]. Thus, the mechanism by which the network classifies digits is hidden from human interpretation. For shallow networks like the ones discussed here, however, the weight maps are easy to understand– it is simply including parts of the variable space with a high ratio of signal to noise, and excluding parts of the variable space with a ratio that is too low. This can be seen by picking two variables to analyze, plotting 2-d histograms of the IBD and Neutron data, and comparing these to a 2-d plot of the score given by the network to different parts of the phase space. Upon doing this, one can see that the network is drawing boundaries in the phase space that are similar to what a person manually looking at the dataset and hand-drawing lines would make. Additionally, because noise outweighs signal by an extreme ratio, the network is sufficiently averse to including noise that it will exclude a significant amount of signal to avoid it.

A criticism of neural networks for particle physics in particular is that the network can pick up on patterns in the computer training data that do not exist in real life. However, this is true of any type of classification informed by computer simulations. Both decision trees and perceptrons approximate curved classification boundaries in the phase space through a sum of simpler components– the major difference is the decision tree's boundaries have a saw-tooth edge from the restriction that all classification planes are perpendicular to a basis vector. Thus, there is no reason why a shallow neural network is more vulnerable to computer simulation patterns than a decision tree. If the neural network was deep enough to allow for many space transformations, it would be possible for the network to pick up on patterned fluctuations from simulation irregularities. However, because all networks discussed here have only one processing layer, they are restricted to drawing simple boundaries and are thus no more vulnerable to computer simulation deception than a decision tree.

FIG. 13. Histograms demonstrating neural network cuts in different variables. Left column: IBD histogram. Middle column: neutron histogram. Right column: network score plot.



IV. SPECTRUM-BASED EVALUATION AND OPTIMIZATION

The final goal of the CHANDLER project is to determine the contents of a nuclear reactor by measuring its neutrino spectrum. The current significance parameter does not directly quantify how effective a detector and its classification software are at spectral measurement. A simple way to do this is to make histograms of the true and predicted IBD spectra and compare. To make the predicted spectrum, run a set of IBDs and neutrons with the expected spectra of a real deployment through the processing pipeline, scale both event types by the real-life rates of IBDs and neutrons, and add them together. The result is the predicted antineutrino spectrum of the reactor. This can be compared to the truth spectrum of the input IBD events to discern how effectively the classification pipeline preserved the spectrum through the noise. An important question is whether an optimization algorithm should take into account positron energy in the parsing of IBDs from neutrons or remain blind to this metric to prevent the classification process from

influencing the spectrum. To answer this question, the 20-plane architecture mentioned above was trained with and without positron energy as an input variable and the predicted spectra of the resulting parameters were plotted.

FIG. 14. Truth Spectrum

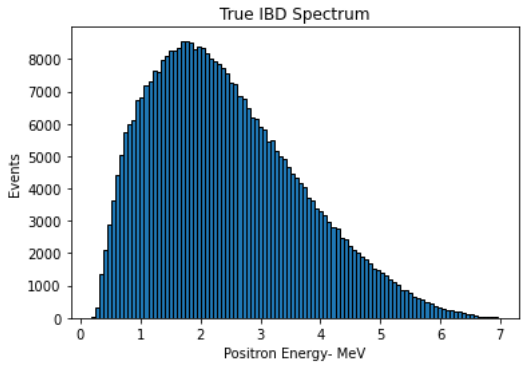


FIG. 15. Measured spectrum without positron energy training

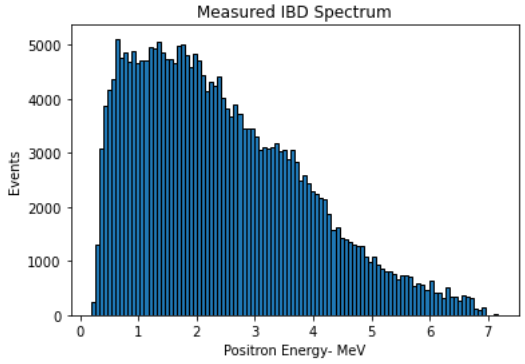
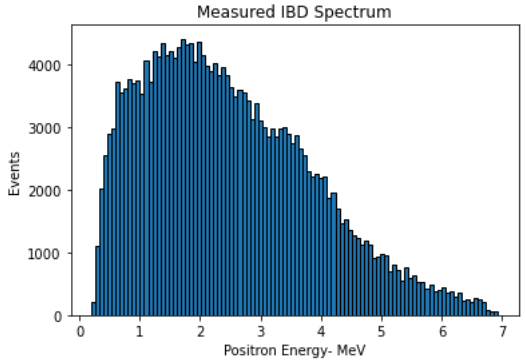


FIG. 16. Measured spectrum with positron energy training



Even though both networks achieved similar significance (156 with positron energy and 155 without), the network that uses positron energy as a classifying condition produces a more accurate spectrum shape. This is because the network with positron energy can make tighter cuts in the energy regimes where neutrons are most common, reducing their ability to distort the spectrum. This can be seen when the false positive neutrons are graphed alone, as the low-energy peak is much less prevalent in the positron-influenced output.

FIG. 17. Neutron contribution to measured spectrum without positron energy training

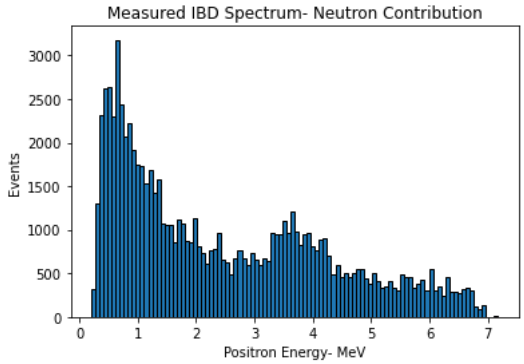
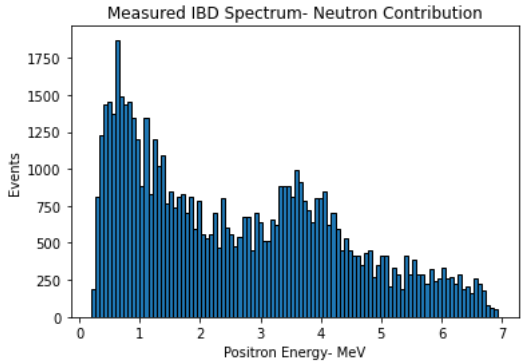


FIG. 18. Neutron contribution to measured spectrum with positron energy training



Thus, including positron energy as a classifying variable prevents spectrum distortion as opposed to causing it. Nonetheless, neither spectrum is perfect, and a way to quantify and optimize on spectral accuracy directly could be useful.

A chi squared score can be calculated between the predicted and truth spectra after both are normalized by dividing the events per bin of both by the total number of events in each histogram. The chi squared or sum of squared difference between the actual and predicted histogram can then be used as a cost function for machine learning to minimize. If the number of events in a certain energy bin needs to be reduced by x amount to match the actual and predicted spectra, this value can serve as the starting cost derivative for backpropagation of training examples in that energy range. To execute this, a similar procedure could be used as with the softsig optimization: run a training batch and compare the true and predicted spectra, record the cost derivative for each energy bin, and run backpropagation for the events in each energy

bin using the cost derivative as the starting value. To avoid training the algorithm to always produce the same spectrum, a variety of spectra must be used as a training set. If one wants the algorithm to accurately identify an arbitrary neutrino spectrum, a wide variety of random spectra can be used as training data. However, if one cares the most about identification of a few select spectra– say, the spectrum of uranium fission vs that of plutonium breeding– then the training set can be limited to those spectra. This would result in an algorithm that is very good at distinguishing between these two possibilities at the expense of spectral accuracy in more general situations. Before energy spectrum training, however, the algorithm should be trained to maximize significance. This is because the above method of spectrum training does not take into account the type of event making up the spectrum, so it should be ensured beforehand that the algorithm has learned an effective way to separate IBDs from neutrons.

V. FUTURE WORK

A. Obtaining Real Training Data

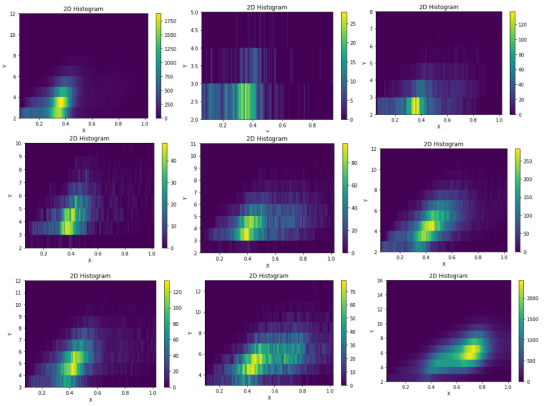
If one desires to eliminate the possibility of computer simulation inaccuracies entirely, real-life test data can be gathered of signal and noise alone. To sample isolated IBD data, CHANDLER could be deployed underneath a commercial nuclear reactor so the cooling water could shield from cosmic ray neutrons. Alternatively, the detector could be placed deep underground along with a small experimental fission reactor or even a strong beta decay source. To sample neutrons alone, CHANDLER can be deployed anywhere away from a fission reactor with a lead overburden to eliminate gamma and charged particle backgrounds. Unfortunately, this requires the detector to be currently built and operational, which is not the case for CHANDLER. Thus, obtaining real training data at this stage is simply not possible. However, once CHANDLER is operational, obtaining real-life training data for analysis algorithms will be an important step in preparation for nonproliferation deployments.

B. Training Improvements

The most obvious way to improve the classification aptitude of the above networks is to add more processing layers, which would allow the network to draw decision boundaries that are smooth curves. This could improve the isolation of signal significantly because IBDs often collect in ellipsoidal "islands" in the phase space histograms. Through the making of network score histograms like figure 13, the inner workings of the network can remain transparent as more layers are added.

Another area of improvement lies in the 1-gamma events, as no network was able to train successfully on that set alone. Firstly, the implementation of a monte-carlo escape probability could reduce the number of 1-gamma IBDs falsely given a low score, improving discrimination between event types. Additionally, some 2-gamma events could be introduced at the start of gradient ascent to guide the parameters to the best maximum and then removed later to optimize on 1-gammas specifically. Optimizing on signal/noise ratio or significance multiplied by signal/noise ratio might also train the 1-gamma network to collaborate better with the 2-gamma network. Furthermore, the 1 and 2-gamma datasets can be segmented based on maximum angle between Compton hits and whether a line passing through two Compton hits and a positron hit in between can be drawn. These different segments of data have different properties, so training separate networks on each could allow them to exploit the unique clustering characteristics of IBDs in each regime.

FIG. 19. Multiplicity vs Total Compton Energy: IBD distributions of different segmentations by SoLid clustering, best pairing angle, and geometry passes.



Finally, a significance parameter that evaluates how effectively an analysis algorithm can detect a spectral change should be created. This would allow for a training procedure that directly optimizes the ability to discriminate between uranium and plutonium neutrino spectra, which is the end goal of the CHANDLER project.

ACKNOWLEDGMENTS

Dr. Link- mentor and project director. Thank you for guidance and the suggestion of a monte carlo approach to escape score.

Keegan Walkup- graduate student advisor. Thank you for guidance and creating the decision tree and the monte carlo training data for the networks.

his work was funded by the National Science Foundation under grant No. PHY-2149165.

[1] A. Haghighat, P. Huber, S. Li, J. M. Link, C. Mariani, J. Park, and T. Subedi, Phys. Rev. Applied **13**, 034028 (2020), arXiv:1812.02163 [physics.ins-det].

[2] "Plot the decision surface of a decision tree on the iris dataset," (2010), tutorial Page.

[3] G. Sanderson, "Neural Networks," (2017), video series.

[4] M. Verstraeten, *Search for sterile neutrinos in the eV and MeV mass range with the SoLid detector*, Ph.D. thesis, Antwerp U. (2021).