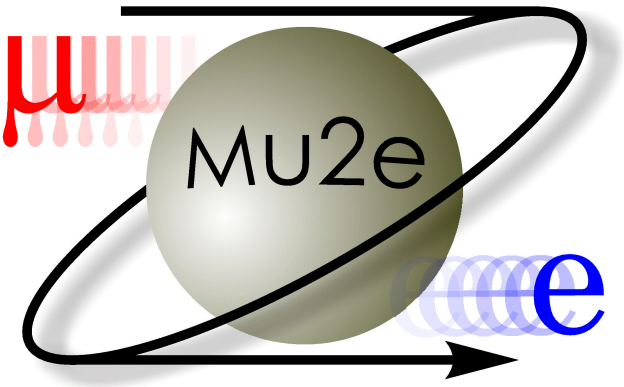
Using Machine Learning to Improve the Performance of the Mu2e Cosmic Ray Veto

Mohit V. Srivastav
University of Virginia
Distinguished Majors Thesis in Computer Science

January 19, 2022



Abstract

The Mu2e experiment at Fermilab is searching for the direct neutrinoless conversion of a muon to an electron. The experiment requires an extremely efficient Cosmic Ray Veto to detect cosmic muons and ignore electrons produced by them that can be confused with real direct conversions. The current Cosmic Ray Veto algorithm suffers from increasing deadtime with beam intensity. This study seeks to explore using a deep neural network to maintain similar cosmic rejection efficiencies while minimizing the experimental deadtime. We found that using machine learning improved upon the current Cosmic Ray Veto algorithm in terms of both the cosmic background and the deadtime, yielding much promise for future exploration.

Contents

1 The Mu2e Experiment

Although the Standard Model of particle physics is well-tested in many areas, it appears to be incomplete. Even though Charged Lepton Flavor Violation (CLFV) (a transition between taus, muons, and electrons that does not conserve lepton family number) is not explicitly forbidden in the Standard Model of particle physics, it is greatly suppressed. Beyond the Standard Model, there exist predictions for observable CLFV rates, and searches of incidences of this have greatly increased in the past few years [1].

One example of CLFV is the neutrinoless conversion of a muon to an electron within the Coulomb field of a nucleus. The Mu2e experiment, expected to start in 2025, will be looking for evidence of such a conversion. The experiment is a multinational project consistent of multiple labs and universities, mounted at the Fermi National Accelerator Laboratory near Chicago [1].

To search for the direct neutrinoless conversion of a muon to an electron, a proton pulse hits a production target every 1.7 μs, where it will produce a beam of low-energy negatively charged muons which will be transported to and stopped at a series of thin foils known as the stopping target by the transport solenoid, which selects the particle's momentum and avoids a direct line of sight from production to the stopping target. At the stopping target, the individual muons will be captured in atomic orbits. The produced conversion electron's momentum, energy, and a variety of other attributes will be recorded by the tracker and calorimeter [2]. The apparatus is pictured in Fig. 1, and the production of a conversion electron from a muon hitting the stopping target in Fig. 2.

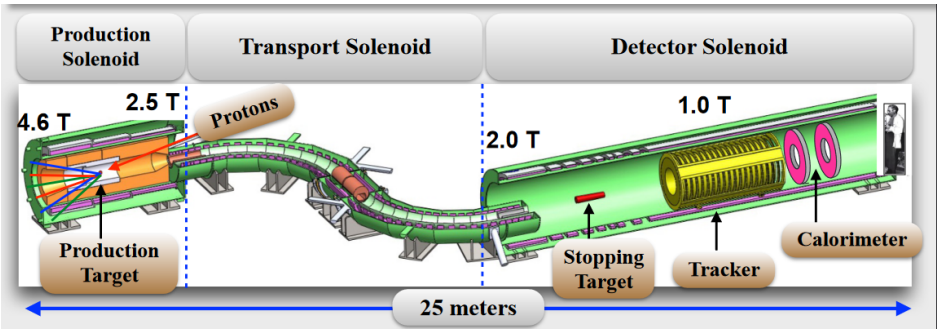


Figure 1: The Mu2e apparatus, with the proton beam, transport solenoid, and detector solenoid separated and pictured. To prevent magnetic bottles which trap muons, the magnetic field of the solenoids is graded (hence the magnetic field strengths in Teslas in the figure). [2]

The signal window for conversion is approximately 1000 ns, with 700 ns to the next proton pulse. Within this time window, if the muon converts to an electron without emitting a neutrino, and the experiment detects it, then a direct muon-to-electron conversion has been found.

2 The Cosmic Ray Veto

The occurrence of a neutrinoless conversion of a muon to an electron is extremely rare, if it occurs at all. A large barrier to achieving the desired sensitivity is the cosmic-ray background,

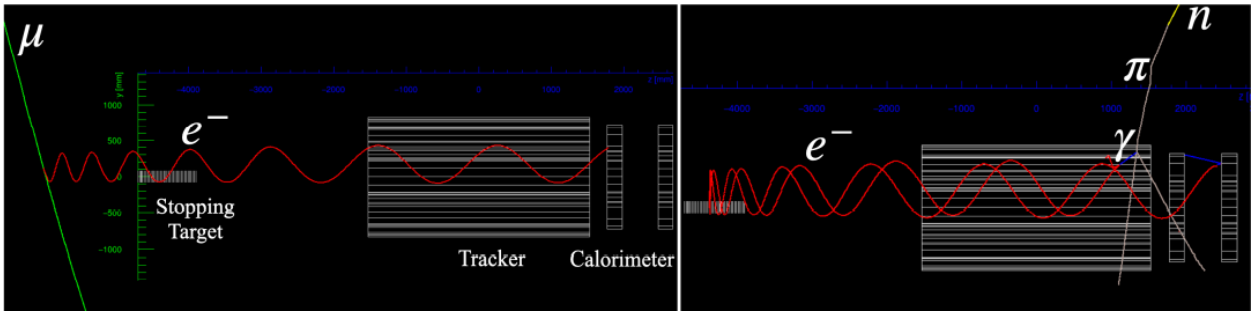


Figure 2: Left: An event produced by a cosmic ray muon that knocks out a conversion-like electron in the Detector Solenoid. Right: A cosmic-ray neutron is incident from the upper right and interacts in the apparatus to produce an upstream-going electron. This electron reverses direction in the Detector Solenoid magnetic mirror and passes again through the tracker. This event is not vetoed by the CRV, because the neutron is a neutral particle, but can be vetoed by the tracker.

induced by cosmic-ray muons. Each minute, approximately one cosmic-ray muon hits the Earth's surface per square centimeter. These muons are expected to produce, on average, one event a day (in this case, event means an electron that has the same characteristics as a real conversion electron) that cannot be distinguished from a successful conversion electron [3].

The rate of such an occurrence has to be reduced by a factor of 10,000 in order to reduce the background to less than one event [3]. The solution to this problem is to surround the Mu2e detection apparatus with a detector that identifies cosmic-ray muons and rejects, or "vetoes", time windows around cosmic-ray muons that produce conversion-like backgrounds during the offline analysis.

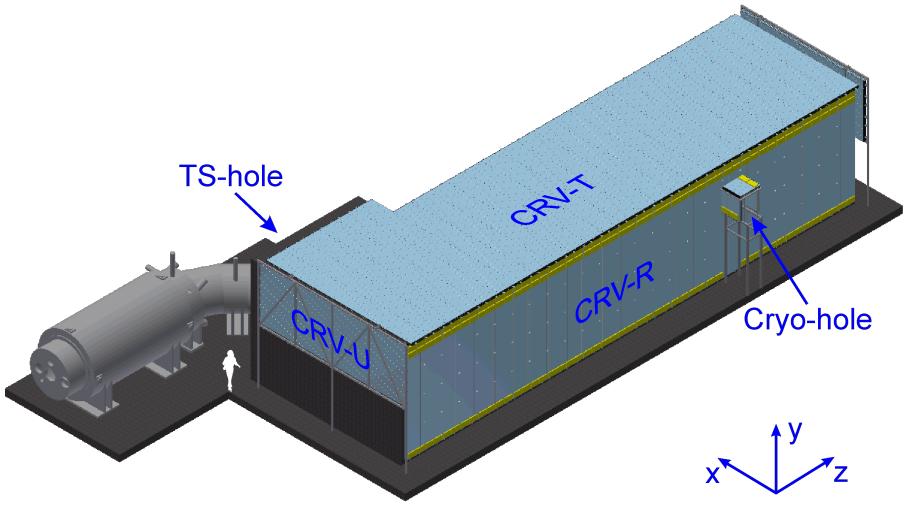


Figure 3: A drawing of what the CRV will look like, along with its coordinate axes. The white object next to the CRV-U is a human, shown for scale.

The Cosmic Ray Veto (CRV), displayed in Fig. 3, consists of four layers of extruded polystyrene scintillators (a material that "scintillates", or emits light, when excited by ion-

izing radiation) counters with embedded wavelength shifting fibers, read out with Silicon Photomultiplier (SiPM) photodetectors [3]. These counters range from 900 to 6600 mm long, and have a cross section of 50 $\times$ 20 mm^2 [3]. These detectors sense when charged particles enter the CRV, and will be used to detect muons and veto the events associated.

An track stub consists of at least three adjacent strips of the CRV with signals over a certain threshold within a 5 ns time window, signifying a real track localized in both space and time. These tracks are reconstructed using an algorithm developed by Dr. Ralf Ehrlich. Once a track stub is recorded within the CRV, the CRV reconstructs various numeric variables for the track stub using the algorithm by Dr. Ehrlich (such as position, light yield, and time), and the tracker itself also records variables pertaining to the interaction of the conversion electron, (such as the momentum, track quality, and the time recorded). Note that particles can enter the tracker without hitting the CRV and still be considered electron events.

Given a good track stub, a 200ns time window around the respective event recorded by the tracker will be vetoed during offline analysis to prevent consideration of any possible electrons produced by a cosmic-ray muon.

3 Study Goals and Strategy

The CRV currently employs a "time window cut" to veto potential conversion electrons produced by cosmic-ray muons. All electron events outside this time window are classified as possible conversion electron (CE) events, and all those within the window are classified as background. With a primary scintillator light yield of 17,000 (see the discussion on light yield in Section 4), the CRV has a very high efficiency rate in identifying cosmic-ray muons.

However, the CRV produces false track stubs from random coincidences due to the beam-induced background in the scintillator counters at high beam intensities. Deadtime, or the fraction of the time that the CRV spends vetoing events, maxes out at approximately 50% at the highest expected beam intensity with the current algorithm. Such behavior is not ideal, as the greater the deadtime the greater the running time of the experiment to achieve a given sensitivity.

The aging of the CRV will also reduce the capability of the CRV to suppress the background of cosmic-ray muons, as the light yields of the SIPMs decrease over time in the scintillator counters, which is shown in Fig. 3. The aging rate has been higher than expected. A new algorithm that is both able to decrease the deadtime and reduce reliance on the light yield of the CRV is being investigated from many angles. Even though the current CRV veto algorithm is quite good at correctly identifying cosmic-ray muons, it is quite crude. It is the goal of this study to make a more sophisticated veto algorithm using machine learning.

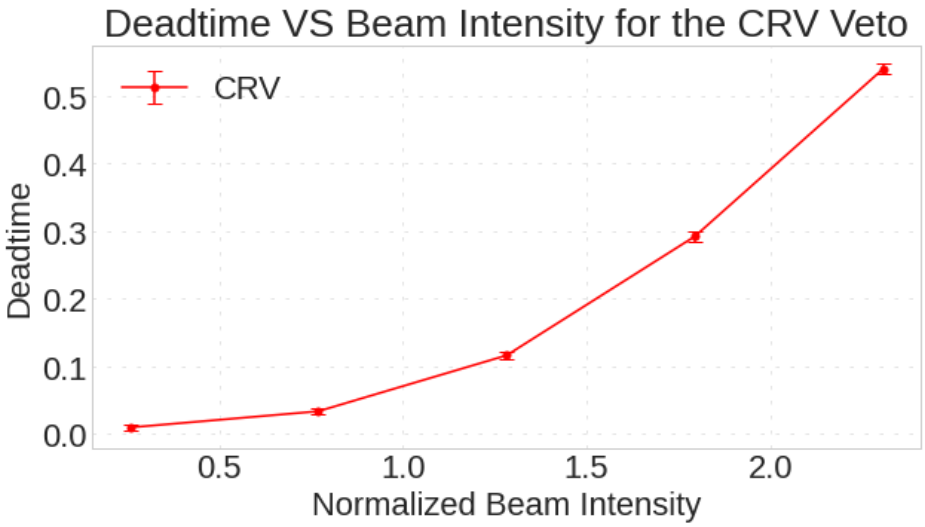


Figure 4: The deadtime produced by the CRV vs the beam intensity. Beam intensity is relative to the nominal value of 3.9×10^7 protons per pulse.

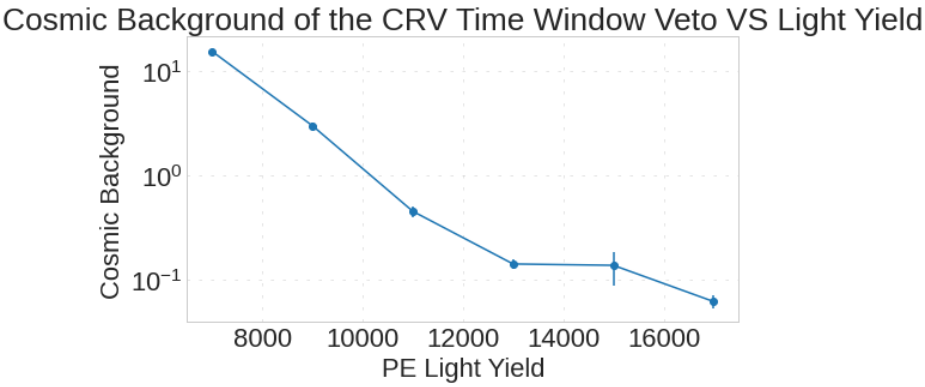


Figure 5: The cosmic background of the CRV Veto vs the scintillator lightyield on the CRY4 Sample

This study seeks to lessen the deadtime, while keeping a similar cosmic-ray induced background, by using a deep neural network instead of a time window cut. Keeping the induced background rate low relative to the light yield was not a primary objective of the study, but it was recognized that a deep neural network could also help to mitigate issues related to aging, so that was studied as well.

The general strategy for the study was to do three things:

1. Generate a sample of conversion electrons overlaid with beam-induced noise. This would be the sample used to determine the deadtime. Ideally, none of the events would be vetoed. This sample is called the CE/Noise sample, as it would consist of conversion electrons overlaid with beam-induced noise.

2. Generate two independent samples of cosmic-ray muons which produce conversion-like events, one for training and one for testing. These would be samples for which, ideally,

all of the events would be vetoed.

3. Optimize an algorithm so as many events as possible in the CE/Noise sample are not vetoed while still maintaining an adequately low induced cosmic-ray background from the cosmic-ray muon samples.

Two conversion electron samples for training and testing are not needed because the traits for the CE/Noise sample are relatively similar; simulating another dataset of CE/Noise events would simply produce similar events to the original dataset, making it unnecessary. The cosmic-ray muons, however, need to have two separate datasets because each cosmic-ray event can be very different, and using an overlapping sample makes for a much less rigorous testing method. The algorithm that was chosen to optimize the background with the deadtime in this study was the current algorithm developed by Dr. Ralf Ehrlich, supplemented by a deep neural network at the last step instead of the time window cut that is used currently.

4 Simulated Datasets

Data were generated using GEANT4 (GEometry And Tracking Iteration 4), a tool for modeling the passage of elementary particles through matter [4]. The software for Mu2e instantiates a solid model of the Mu2e apparatus within the GEANT4 framework [5]. Three different data sets were generated by Dr. Yuri Oksuzian, a scientist at Argonne National Laboratory. These simulated datasets mimicked the behavior of their respective particles through the CRV and the tracker, and had simulated numeric variables corresponding to what the tracker/CRV would record with real data. The distributions in Figs. 6, 7, and 8 have the same coordinate axis as the one shown in Fig. 3.

Lightyield will be referenced throughout this study. It is a numeric variable when simulating datasets that represents the number of PEs in the primary scintillator of the CRV. The conversion factor between the light yield and the number of PEs is 0.001726. The conversion between the lightyields used and PE count is in Table 1.

Lightyield	PE Count
7,000	12.08
9,000	15.53
11,000	18.99
13,000	22.44
15,000	25.89
17,000	29.34

Table 1: A table of lightyields on the left with their corresponding PE count on the right.

The CE/noise dataset was produced by simulating conversion electrons, overlaid with noise produced by the Mu2e beam, which consists of neutrons and gammas, and was produced at a light yield of 17,000. Overall, this dataset represented events that should not be vetoed, lest they contribute to the deadtime of the experiment. Figure 6 shows the distribution of noise events that were successfully reconstructed. The sample corresponds to "Run

1", or a simulation of the first run of the Mu2e experiment, which would run for 3.46×10^6 seconds. There were 2,385,473 Noise/CE events in the sample, and the event distribution is in Fig. 6.

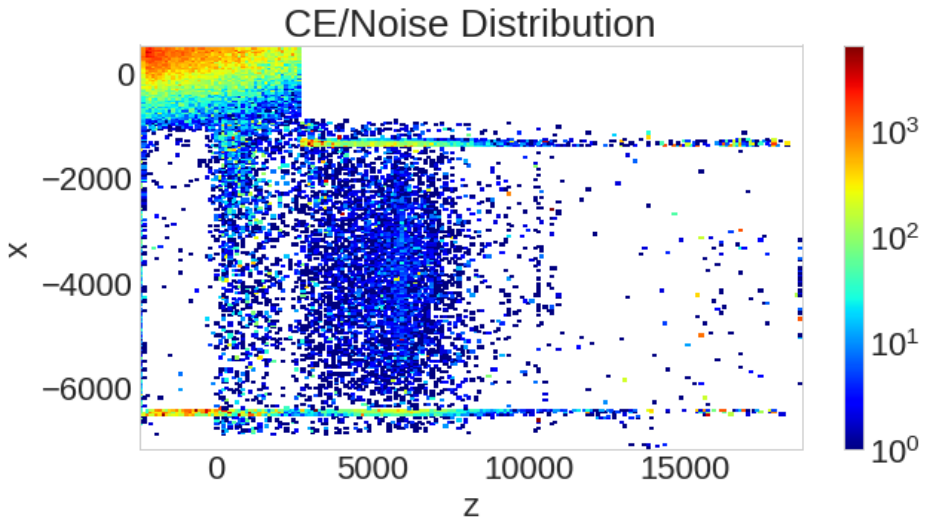


Figure 6: The xz distribution of the CE/Noise sample. These are only CE/Noise events that produced a track stub in the CRV; there were a large number of CE/Noise events that were reconstructed by the tracker but did not produce a track stub in the CRV that are not pictured.

The CRY3 dataset corresponds to cosmic-ray muons that produced an electron-like track that was successfully reconstructed in the tracker. The CRY3 sample was a mix of high-energy and low-energy muons, as defined in Eq. 1.

$$\text{Class}_{\text{muon}} = \begin{cases} \text{High Energy} & \text{Energy Deposition} > 14\text{MeV} \\ \text{Low Energy} & \text{Energy Deposition} < 14\text{MeV} \end{cases} \tag{1}$$

The scintillator response assumed a primary scintillator lightyield of 17,000, just like the CE/Noise Sample. There are 1,842,456 reconstructed muon events in the CRY3 dataset, and the event distributions are in Fig. 7, separated into high and low energy muon components. The reason for the distinction between high and low energy muons was so that any possible performance differences could be identified, in case the difference in energy deposition created a difference in performance.

The CRY4 dataset was produced in nearly exactly the same way as the CRY3 dataset. However, there were differences between the two samples. The CRY4 sample used the most up-do-date iteration of the generation algorithm, and had greater statistics, along with an updated shielding geometry. The sample also contained samples at differing light yields, in order for the algorithm to be tested on multiple light yields.

The light yields that were generated were at 17,000, 15,000, 13,000, 11,000, 9,000, and 7,000. There were 15,157,304 events total in the CRY4 muon sample. Fig. 8 shows the position distribution of the CRY4 sample, separated into high and low energy components.

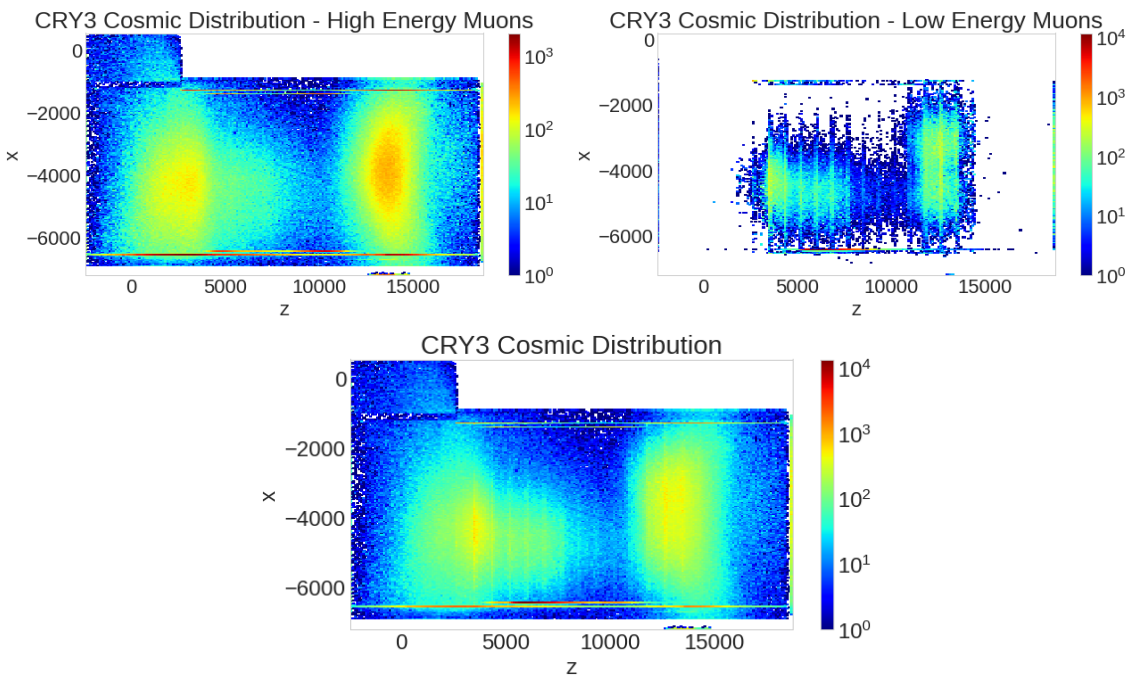


Figure 7: Position distributions of the CRY3 sample. The distribution in the top left is the xz distribution of the high energy muons, the top right the xz distribution of low energy muons, and the bottom the combined xz distribution. The two areas of higher density are the tracker and calorimeter, on the left and the right.

5 Important Definitions

The following section serves to define terms used for the rest of the paper.

5.1 Variables

Table 2 defines the variables used throughout the study. The variables in bold were selected for use in the machine learning model, whereas the others are simply variables referenced throughout the paper. These variables were within every sample produced. However, events that did not hit the CRV (and were only reconstructed by the tracker) did not have any of the variables that began with "crvinfo", as they were attributes provided by the CRV. The variables were used at the recommendation of Dr. Yuri Oksuzian.

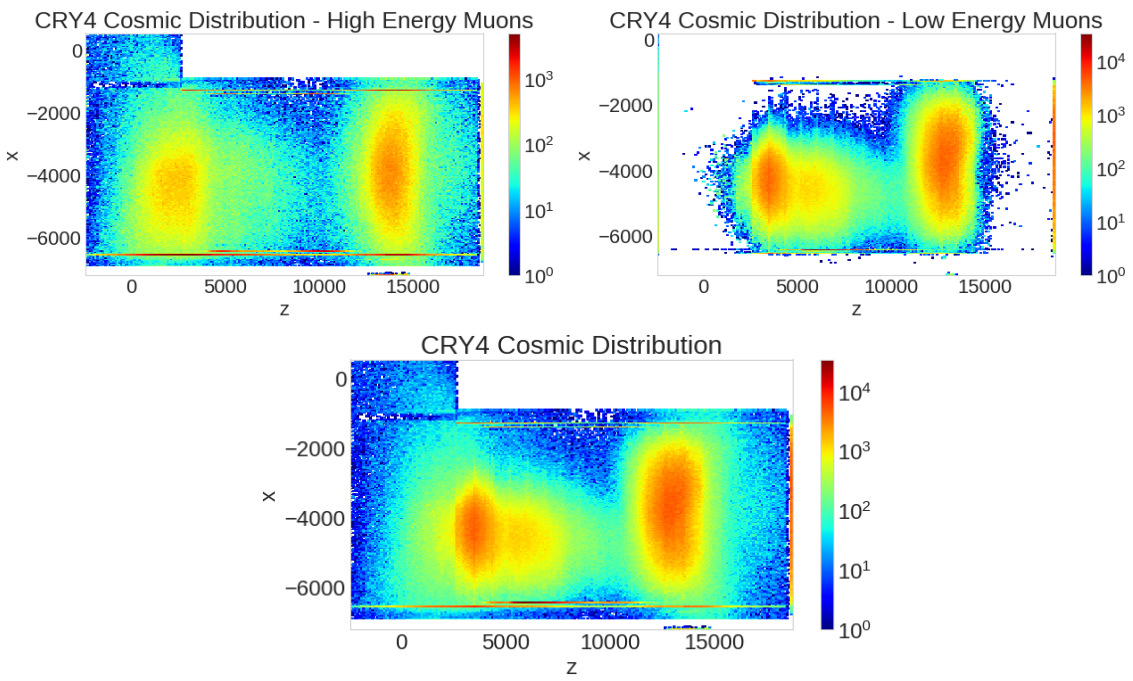


Figure 8: Position distributions of the CRY4 Sample. The distribution in the top left is the xz distribution of the high energy muons, the top right the xz distribution of low energy muons, and the bottom the combined xz distribution.

Variable Name	Corresponding Attribute
crvinfo__x	recorded X position of the track stub inside CRV
crvinfo__y	recorded Y position of the track stub inside CRV
crvinfo__z	recorded Z position of the track stub inside CRV
de_nhits	Number of tracker hits for particles moving "downstream" towards the tracker
ue_nhits	Number of tracker hits for particles moving "upstream" towards the stopping target
dequal_TrkPID	Particle ID
dequal_TrkQual	Track Quality
deent_td	Pitch Angle
deent_z0	Starting z value of track
deent_d0	Track's minimum displacement from the z axis
deent_om	Minimum transverse radius
crvinfo__PEs	PE yield of primary scintillators
de_t0	$\text{Time}_{\text{Tracker}}$
crvinfo__timeWindowStart	Time_{CRV}
crvinfo__dT	$\Delta\text{T} = \text{Time}_{\text{CRV}} - \text{Time}_{\text{Tracker}}$
deent_mom	Momentum
deent_d0_om	Maximum transverse radius
is_cosmic	Cosmic status of event

Table 2: Definitions of the variable names. Those in bold were used for the machine learning model defined in Section 8. Variables starting with "crvinfo__" are variables recorded in the CRV; everything else is recorded in the tracker, with the exception of "is_cosmic", which was a user-defined variable meant to distinguish between cosmic-ray muon and CE/noise events.

The Particle ID and Track Quality variables were produced from machine learning out-

puts from track-finding algorithms used for the tracker. They output a confidence level corresponding to the following:

- The Particle ID is a machine learning produced number that indicates how confident the event is a conversion electron or not, ranging on a scale from 0 to 1, where 1 indicates full confidence.
- The Track Quality is a machine learning produced number that indicates how good of a track an event has, ranging on a scale from 0 to 1, where 1 indicates a very good track.

Similarly, the pitch angle is a unitless tangent ratio between the solenoid axis and particle direction that helps to remove slow moving particles [6]. Movement upstream and downstream are towards the stopping target from the calorimeter and the calorimeter from the stopping target, respectively. Some events in the tracker move in some combination of upstream and downstream, so checking whether an event's track is moving downstream is not the same as checking if an event's track is not moving upstream, hence the further restriction. For example, the electron produced by the cosmic-ray neutron in Fig. 2 moves upstream, then reverses direction and moves downstream as well.

5.2 Cut Terminology

There were a variety of cuts used in the study as well. These cuts, based off of the attributes of particles that enter the tracker, have been optimized over the course of multiple Mu2e studies to both help suppress cosmic-ray muons alongside other backgrounds, and to separate noise from real conversion electrons.

The different cut sets are utilized for different purposes the stricter they are, and both the stricter and looser cuts were used throughout the study. The cuts are presented in a hierarchy going from least restrictive to most because sometimes looser cuts were used for different datasets, and it helps to understand what the exact differences are between cuts to further understand some of the steps further into the study, since they all build off one another. Table 3 displays the number of events for the CRY3, CE/Noise, and CRY4 samples after given cuts. The code for the cuts is in Appendix C.

Loose Box Cuts

These were the least restrictive cuts, simply checking whether an event was triggered or not, as well as some checks on basic attributes: the pitch angle must be greater than 0.577350 and less than 1, the minimum displacement from the z-axis must be greater than -80 and less than 205 mm, and the sum of the minimum displacement from the z-axis and twice the minimum transverse radius must be greater than 450 mm.

Loose Cuts

The Loose Cuts add restrictions for various variables that are indicative of whether an event is a good track or not. The event's track quality must be greater than 0.8, and the

particle ID greater than 0.95. The upstream status, or whether the event's track was moving upstream, needs to be less than or equal to 0 (which indicates that it was not moving upstream). These are all applied on top of the Loose Box Cuts. The xz distribution of the cuts applied to the CRY4 dataset is in the top-left image of Fig. 9.

The Loose Cuts are more restrictive than either the Box Cuts or the Loose Box Cuts, but less restrictive than every other cut that checks for track quality, hence the name.

Box Cuts

The Box Cuts are a more restrictive version of the Loose Box Cuts, with the following variable restrictions added to the Loose Box Cuts: the downstream status, or whether the event was moving downstream, needs to be greater than 0, the minimum displacement from the z-axis must be less than 105 mm, and the sum of the minimum displacement from the z-axis and twice the minimum transverse radius must be less than 680 mm. These are applied after the Loose Box Cuts.

Noticeably, the upper restriction on the distance from the z-axis became more restrictive, and there is now an upper bound to the sum of the distance from the z-axis plus twice the minimum transverse radius, and a necessary downstream status.

Quality Cuts

The Quality Cuts are cuts on the quality of the conversion electrons found in the tracker. These cuts are the same as those added in the Loose Cuts (track quality greater than 0.8 and particle ID greater than 0.95), with the only difference between the Quality Cuts and the Loose Cuts being that the Quality Cuts are applied on top of the Box Cuts, whereas the Loose Cuts are applied on top of the Loose Box Cuts.

Kinematical Cuts - Extended Momentum Cut

The Extended Momentum Cut (or Cut_extmom) is the looser of the two Kinematical Cuts, or the strictest cut available to events. The momentum must be greater than 100 and less than 115 MeV. These cuts are applied on top of the Quality Cuts.

This cut was often used in place of the Physical Momentum Cut mentioned below when there weren't enough events to have sufficient statistics. This was usually the case with the CRY4 Cosmic Sample, as the Physical Momentum Cut usually eliminated too many cosmics for the results to be statistically significant. The xz distribution of the cuts applied to the CRY4 dataset is in the top-right image of Fig. 9.

Kinematical Cuts - Physical Momentum Cut

The Physical Momentum Cut (or Cut_phymom) is the strictest cut available, and ensures that almost anything that passes, if a part of the CE/noise sample, is a CE, and mitigates the cosmic background by almost 99%. The momentum must be greater than 103.85 and less than 105.1 MeV, and is applied on top of the Quality Cuts. The xz distribution of the cuts applied to the CRY4 dataset is in the bottom image of Fig. 9.

In the actual experiment analysis, the Physical Momentum Cut would be applied to the dataset in order to discern what events were cosmics, CEs, etc.

Cut	CRY3	CE/Noise	CRY4
No Cut	1,842,456	2,385,473	15,157,304
Loose Box Cuts	220,918	1,814,739	2,059,156
Box Cuts	133,307	1,806,498	1,280,300
Loose Cuts	75,175	1,579,431	786,184
Extended Momentum Cut	11,577	1,457,507	130,903
Physical Momentum Cut	847	893,575	9,863

Table 3: A table of how many events remain for the three datasets after the given cuts defined in Section 5.2. Each cut is more restrictive than the last going from top to bottom, as they all build off of each other. Note the small number of remaining events in the CRY3 and CRY4 samples after the Physical Momentum Cut relative to the original number of events without cuts, and the larger number of CE/Noise events after the Physical Momentum Cuts.

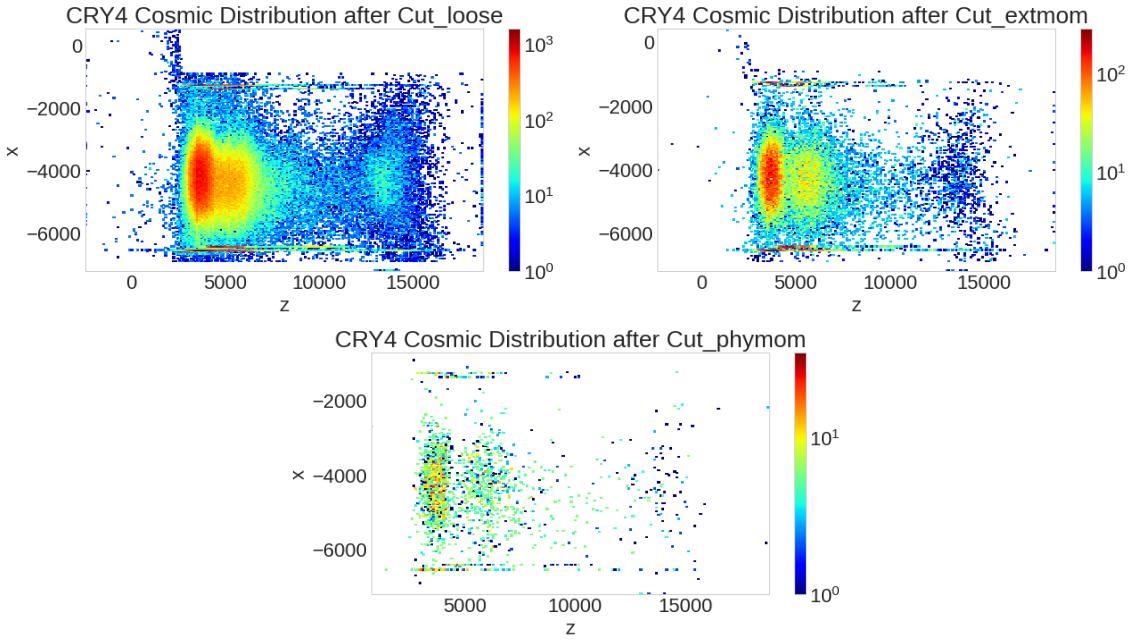


Figure 9: The position distributions for track stubs from the CRV in the CRY4 sample after various cuts. The top left is after the Loose Cuts, the top right after the Extended Momentum Cut, and the bottom after the Physical Momentum Cut. Note the number of cosmic-ray events eliminated after each cut.

CRV Time Window Cut

The CRV Time Window Cut is the cut used in the current algorithm to veto events in coincidence with a track stub in the CRV. It is applied after either of the Kinematical Cuts (for the actual experiment it would be the Physical Momentum Cut, but in this case it could be either of the cuts, depending on how good the statistics are). It is a 200 ns window

determined by the time recorded in the CRV and the time recorded in the tracker. The cut applied is that ΔT must be greater than -50 ns and less than 150 ns.

5.3 Deadtime

The deadtime of the experiment is the fraction of data removed by the action of CRV Veto. Part of the deadtime comes from the vetoing of cosmic-ray muons, whereas the other portion comes from false coincidences in the CRV induced by neutrons and gammas from the exposed beam. The latter of the two dominates. Real conversion electron events that the experiment is looking for may be vetoed if they look similar to a cosmic-ray muon. Similarly, the deadtime that was viewed for this study was the "harmful" deadtime, or the time spent vetoing false coincidences. Due to this, every reference to "deadtime" in this study refers to the deadtime induced by conversion electrons and noise. The expression for the deadtime and its uncertainty can be expressed as:

$$\text{deadtime} = 1 - \frac{\text{Identified}_{\text{CE}}}{\text{Total Possible}_{\text{CE}}}, \tag{2}$$

$$\Delta\text{deadtime} = \frac{\sqrt{\text{Identified}_{\text{CE}}}}{\text{Total Possible}_{\text{CE}}} \tag{3}$$

The denominator of the fraction in the equations above is all the CE/Noise events that passed Kinematical Cuts with the Physical Momentum Cut.

This value must be minimized, as the greater the deadtime, the longer the experiment's anticipated runtime for a given sensitivity. Since the CRV vetoes the entire 200 ns window surrounding an event that it believes to be a cosmic muon, a deadtime of, for example, 50% would correspond to a 100% increase in the time required for the experiment, as the CRV would be unable to detect anything else during that window. Decreasing the deadtime of the experiment would then decrease the anticipated experimental runtime.

5.4 Cosmic-Ray Induced Background

The cosmic-ray induced background should be reduced to well less than one expected event over the course of the experiment. Cosmic-ray muons that pass through the CRV can produce a background in the form of a cosmic-ray induced electron. Due to the simulated nature of the data produced, the number of events had to be normalized to the expected "livetime" for one run of the Mu2e experiment, since the overall background over a run is heavily dependent on the livetime of the experiment. Run One of Mu2e's livetime is equal to 3.46×10^6 seconds.

The simulated livetime for the low energy muons was $1.36 \times 10^8 + 5.09 \times 10^7 = 1.869 \times 10^8$ seconds (the livetime was divided into two portions because the low energy muons were simulated in two different batches). The simulated livetime for the high energy muons was 3.64×10^6 seconds. The expected cosmic background, and its uncertainty, over the course of

one run of the experiment is below:

$$\text{bkg} = \frac{\text{Number of cosmic events not vetoed}}{\text{livetime}_{\text{sample}}} \times \text{livetime}_{\text{Mu2e Run}}, \tag{4}$$

$$\Delta\text{bkg} = \frac{\sqrt{\text{Number of cosmics not vetoed (after event selection criteria)}}}{\text{livetime}_{\text{sample}}} \times \text{livetime}_{\text{Mu2e Run}} \tag{5}$$

6 The Current Mu2e Veto Algorithm

The current Mu2e veto algorithm finds track stubs in the CRV using an algorithm developed by Dr. Ralf Ehrlich, then uses the CRV Time Window Cut Veto, where CE-like events in the tracker that pass the CRV Time Window Cut are assumed to be cosmic-ray induced.

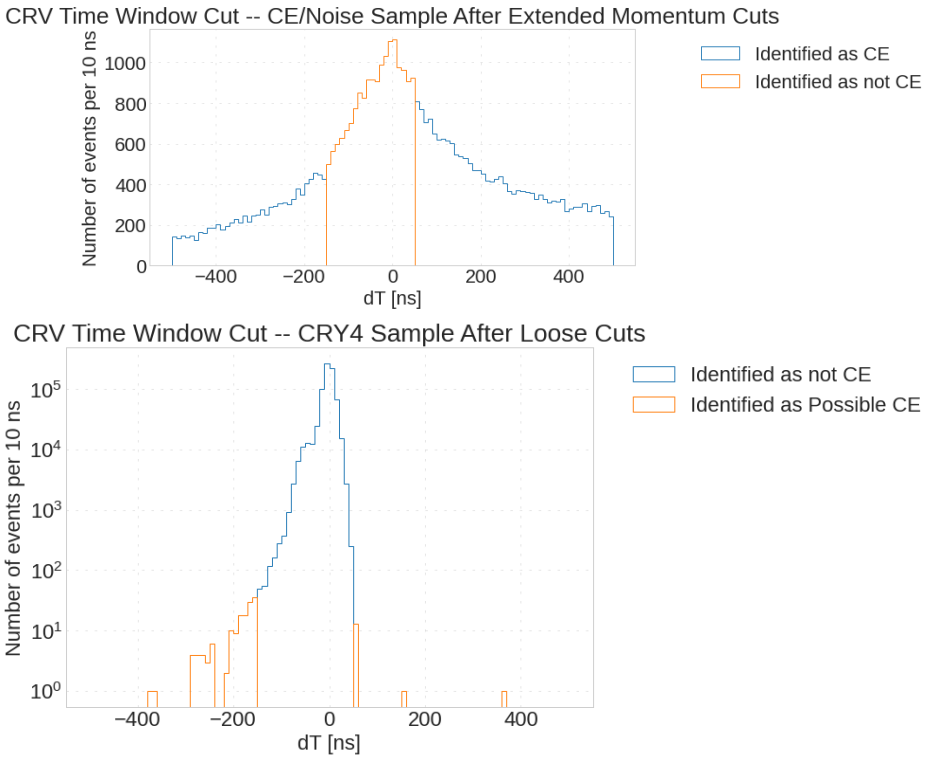


Figure 10: The ΔT distribution and identification of the CRV Time Window Veto for two different datasets. The top plot is the Time Window Cut applied after the Extended Momentum Cut on the CE/Noise sample, and the bottom plot the Time Window Cut applied to the CRY4 sample before any other cuts. Everything identified as "not CE" would be vetoed, and everything identified as "possible CE" a possible CE event (pending the Physical Momentum Cut).

Figure 10 displays how the CRV Time Window Cut works with respect to the ΔT value. Note how everything within the window is identified as a cosmic-ray induced event, and everything outside a possible real CE event in Fig. 10.

7 Preparing the Data

In order to be used for machine learning, as well as data analysis in general, the data generated had to be prepared. This section outlines the steps taken to prepare the data for input into the machine learning model.

All of the data had to be labelled in a binary fashion for proper classification into 2 classes. Since the data for the CE/Noise data was intermixed with the CRY3 and CRY4 data for the training and validation of the machine learning model (as shown later on), an easy way to differentiate them was needed, hence the labelling. As such, another variable was added to the dataset called "is_cosmic", and assigned a 1 if the event was from the CRY3 or CRY4 samples, and a 0 if it was from the CE/Noise dataset.

The CE/Noise data, due to issues with its simulation, led to a relatively unphysical distribution of the recorded z position of the track stubs in the CRV due to the peaks in the dataset. As such, the z positions were smeared such that the peaks were still represented, but the distributions made smoother, making them slightly more realistic, as shown in Fig. 9 below.

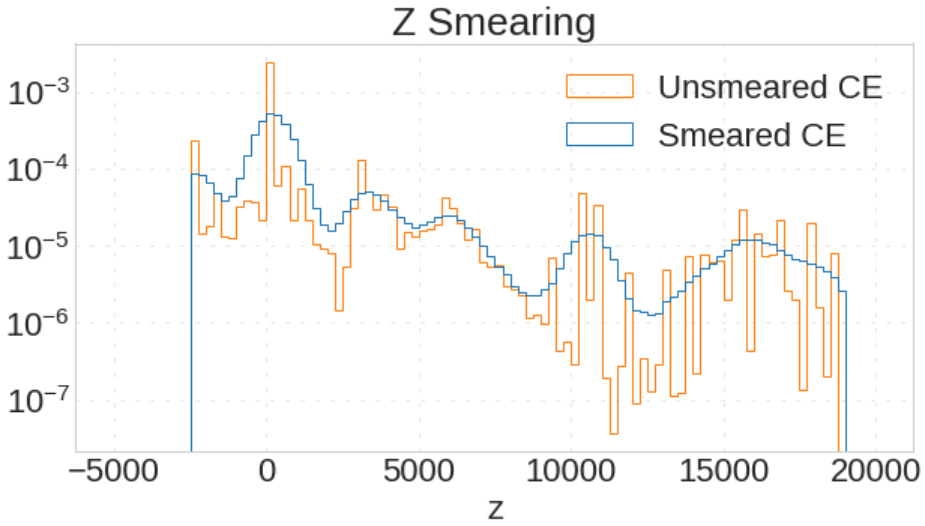


Figure 11: The z-position of CE/Noise track stubs in the CRV before and after smearing.

8 Building the Model

The Machine Learning model was built using the Keras [7] sequential programming interface for deep neural networks, which is a package for Python, the programming language used. This package allows for easily customizeable deep neural networks to be built and used. NumPy and Pandas, packages for data management and manipulation were also used throughout the study. The variables selected to be input into the model, referenced in Table 2, were at the recommendation of Dr. Yuri Oksuzian.

Neural networks are a method of machine learning where patterns are recognized using sophisticated mathematical modeling by inputting data through "neurons", known as perceptrons, that exist in different layers of varying input widths. A model is the structure of

these neurons used for machine learning.

The way that Keras constructs deep neural networks, or neural networks with more than one layer, is by adding them sequentially (hence the "sequential" interface). Layers are added one after the other with each one inputting into the next, where the output of the last layer added is the output of the entire machine learning model itself.

8.1 The Reasoning Behind the Existence of Two Models

There are two types of events within all of the simulated data: events that produce hits in the CRV, and those that do not. Events that do not produce hits do not have CRV variables, but are still recorded by the tracker, and thus still have the variables from the tracker.

At the beginning of the study, there was only one model - no matter the type of event. However, the existence of CRV variables correlated with whether an event was a CE/Noise event or a cosmic muon, diluting the actual correlations of the variable within the dataset. As seen in Fig. 12, the three CRV variables (z position, PE Yield, and ΔT) are all perfectly correlated with one another, and have the same correlation value to cosmic status as well. This behavior was due to the fact that the large number of events without CRV variables was assigned a "dummy" value of -999,999, and so the natural correlations between variables was washed out. This phenomenon ended up producing suboptimal results, and so a change was necessary.

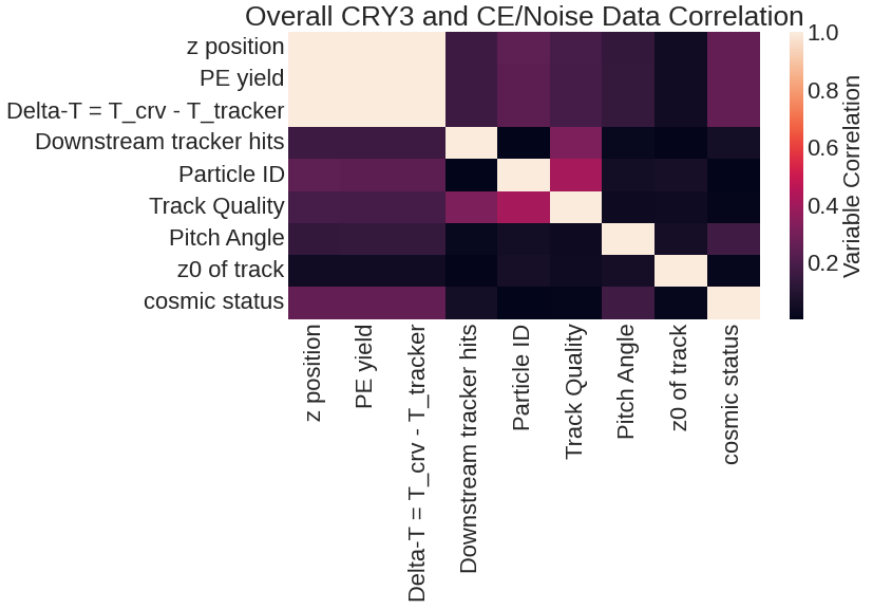


Figure 12: The correlation of variables for a mixture of the CRY3 and CE/Noise dataset.

Every dataset was then split in two, known as the "noCRV" and "CRV" datasets, with their corresponding models named the same. The "noCRV" dataset is that without CRV variables, and the "CRV" dataset is the one with CRV variables. An event without CRV variables, for example, could be an electron produced by a muon that went through the TS-Hole, depicted in Fig. 3, as the CRV does not cover there.

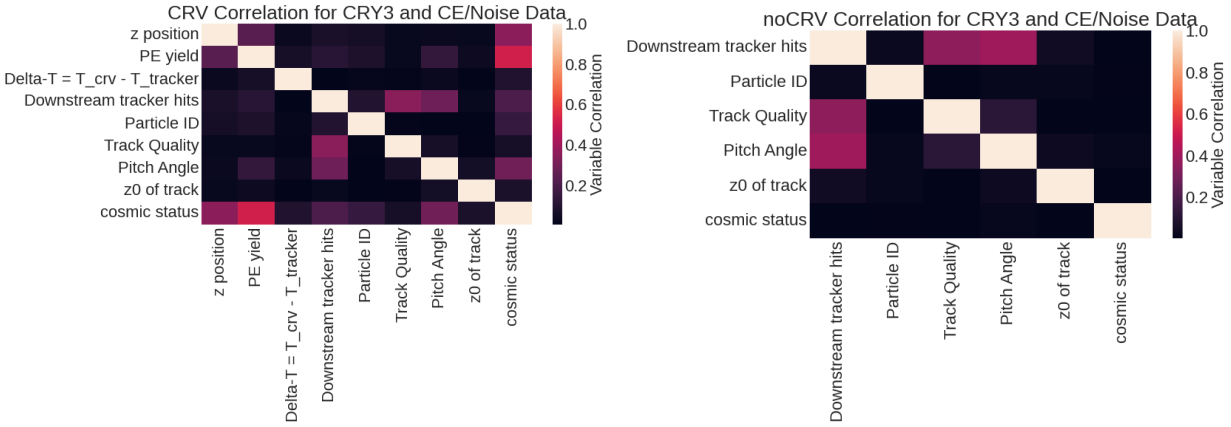


Figure 13: Correlations for the CRV (left) and noCRV (right) data, which consists of the mixed CRY3 and CE/Noise data from Fig. 12.

The attributes in question are relatively uncorrelated with each other, as seen in Fig. 13, as the correlation from CRV variables existing did not override the existing natural correlations within the CRV data. This was the goal for these variables, as when variables are correlated, one or more variables are either unnecessary or even harmful to model generalization. The reason for this is that the model should generalize as much as possible, if there are two variables that offer similar information, it tends to make the model less generalizeable. This would be because that attribute would tell the same information as another, yet be weighted similarly, leading to a weaker ability to generalize without making the model unnecessarily deep/wide. Overall, the two types of events were better encapsulated using two different models than by using one overarching model.

However, the noCRV data had few useful variables to discriminate between Cosmic and CE/Noise events, as can be seen in the low correlation coefficients with the cosmic status of the attributes in Fig. 14. This is a recurring theme for the noCRV dataset, it is quite difficult to discriminate between cosmics and CE events without CRV variables throughout this study. Much of the classification power resides in the CRV data, as evidenced by the much stronger correlations for that dataset in Fig. 14.

8.2 Data Cleaning and Separation

For a neural network, there need to exist three independent datasets: training, validation, and testing. The training dataset trains the model, hence the name. The data in the training set is entered into the model as it learns. The validation dataset is the dataset that the model checks against at each training interval. The validation set serves as a sanity check for the model within training, and the independence of this dataset is important to protect against overtraining, which is defined in Section 8.3. The testing dataset is the dataset used for testing the model after the training process is done. It is important that every dataset be independent of one another, so that any machine learning model created isn't simply looking at the same data it was trained upon, as that would not test the model's ability to generalize.

For all the data input into the model for training, validation, and testing, the data were standardized such that the mean and the standard deviation of the data for each input

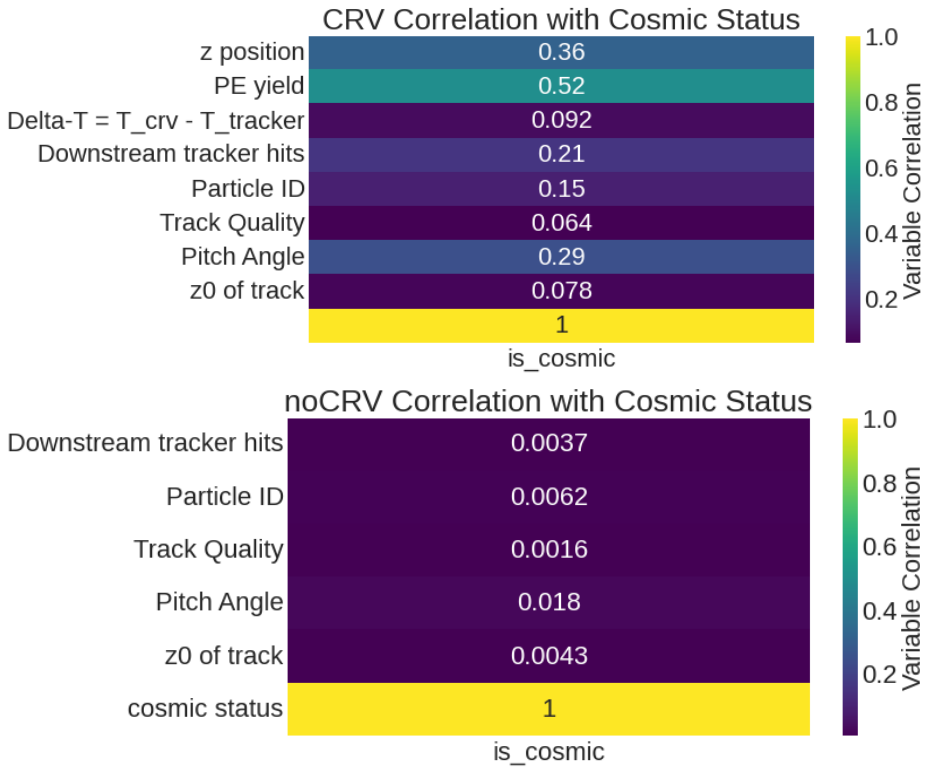


Figure 14: The correlation between variables used and cosmic status. The top plot is for the CRV data, and the bottom plot for the noCRV data. Correlation values vary from 0 to 1, with 0 being no correlation, and 1 being a perfect correlation.

variable was 0 and 1, respectively. This was done so that each variable had the same numeric weight assigned to it within the model's weight matrices. An example of such is the difference between the values for PE yield and Particle ID. While both are important variables, the numeric values of the PE yield can reach into the thirties, while the Particle ID ranges from 0 to 1. These variables, if not normalized, would lead to a very wrong weight matrix within the model.

Afterwards, the data were separated into events with and without CRV hits. The CRY3 and the CE/Noise dataset were treated separately for both cases, making 4 separate datasets (CRY3 CRV, CRY3 noCRV, CE/Noise CRV, and CE/Noise noCRV). Then the common datasets were merged, to make 2 datasets (CRV and noCRV) for usage in the training, testing, and validation datasets.

For the CRV data, the CE/Noise sample had the Extended Momentum Cut applied to it, while the CRY3 sample had loose cuts applied to it. The reason for this was that any of the harsher cuts, such as the momentum cuts, severely decreased statistics for model training since they would eliminate a large number of cosmic events. The cuts for the CE/Noise data, however, still had good statistics after the cuts, and there was no point training the model on events that it would never had to evaluate (because they would fail the cuts anyways).

A fraction of 80% of the CRY3 data went to training, with 20% going to validation. None of the CRY3 dataset went to testing, as the CRY4 dataset made up the entirety of the cosmic-ray portion of the testing data. A fraction of 60% of the CE/Noise data went to

training, 20% to validation, and the other 20% to testing.

The CRY4 sample was purely used for testing. The reason for this was that the CRY4 sample is easier to test robustness on, due to the different light yield values and greater statistics, and is also independent of the CRY3 sample, allowing for an unbiased analysis by the model. The difference between the CRY3 and CRY4 datasets is minor as it only involves the simulation of the shielding, and the input variables behave and act the same, so the differences in the dataset are negligible.

The noCRV dataset, using the same dataset breakdown between training, testing, and training as the CRV data, led to a problem, as there were a lot more CE events than there were cosmic events. This led the model to predict everything as a CE event. As such, the data processing for the noCRV dataset was slightly different. Due to the very low number of cosmics without CRV variables that pass Loose Cuts, only Box Cuts were used for the CRY3 dataset. Then, a subset of the CE/Noise events that passed the Extended Momentum Cut, randomly sampled down to be of size $2 \times \text{size}(\text{noCRV}_{\text{cosmic set}})$, was split up into the same fractions from the CRV dataset in regards to ML training (60% to training, 20% to validation), with the remainder sent to the testing sample. This made it such that the number of events in the training and validation datasets was much more equal, about 1:1.2 CRY3:CE/Noise, as opposed to a much more unequal ratio. The data breakdowns for the sizes of the training, testing, and validation data are depicted in Fig. 15 below.

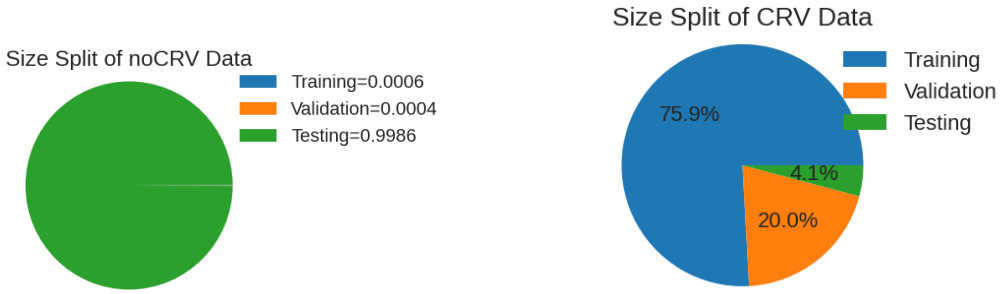


Figure 15: The final datasize breakdown for the dataset

The size of the training sample was 335,939 events for the CRV model, and 2,148 events for the noCRV model. The size of the validation sample was 106,985 events for the CRV model, and 628 events for the noCRV model. The breakdown of what data samples made up each dataset is in Fig. 16.

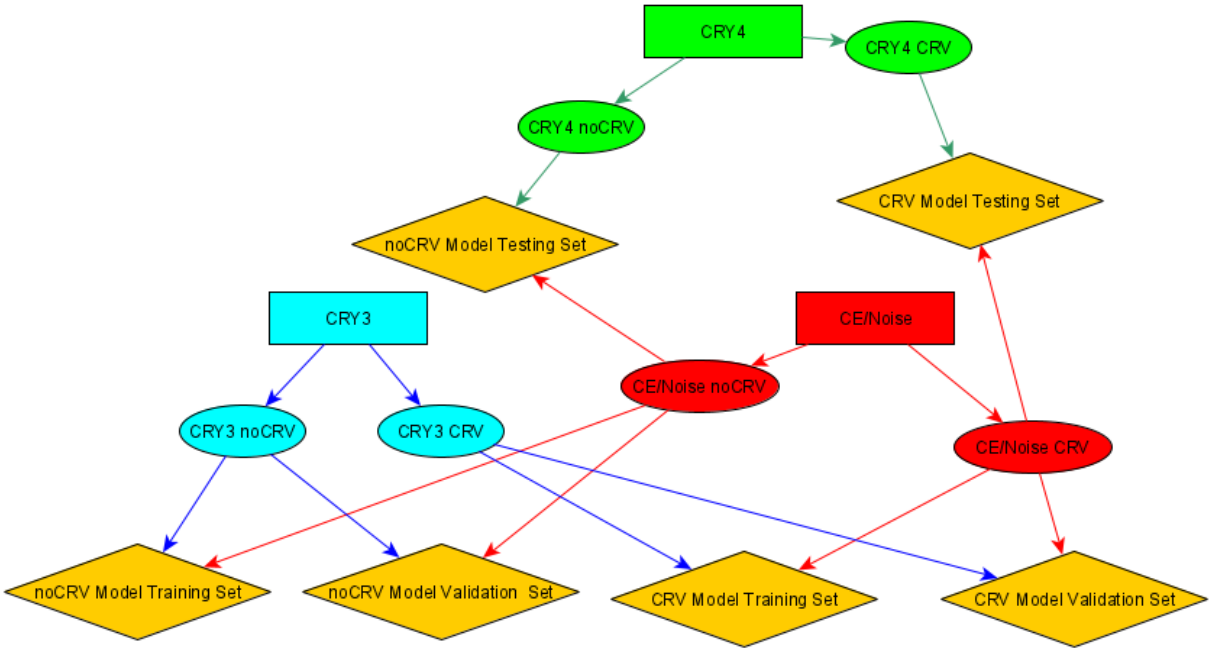


Figure 16: The different data samples and how they fit into the datasets used for the study. Blue marks the CRY3 sample, red the CE/Noise sample, and green the CRY4 sample. Simulated samples are rectangles, subsets of those samples are ovals, and the model datasets are yellow diamonds. Arrows indicate where the datasets were distributed to.

8.3 Model Parameters and Metric Definitions

The batch size is arguably the most important hyperparameter, or an attribute of the machine leaning model itself, of any given machine learning model, because the batch size directly affects how the weight matrices are updated. The batch size defines the number of data points that the model sends through before updating its weight matrices during training [8].

The epoch number is the number of times the model will run through the entire dataset when training. An epoch contains one or more batches, and models can have a large number of epochs [8]. Usually, however, the number of epochs (if one wants to prevent overtraining), correlates with the batch size (the smaller the batch size, the smaller the number of epochs). The number of epochs to train for can be manually changed, but the number is usually set to something very high and set to stop before the number of epochs set by the user (called early-stopping) due to a given user-defined metric. Checking model metrics for early stopping happens either at the beginning or at the end of every epoch.

For example, suppose there is a training sample of T events, with a batch size of B, and a number of epochs E. The dataset will be divided into a number of batches defined by Eq. 6 below. If the training sample size and the batch size do not evenly divide, then the final batch will simply be the remainder of data samples [8]. Each epoch will then have that number of updates to the weight matrices per epoch. The number of epochs trained for is purely up to the user.

$$\text{BatchNumber} = \left\lceil \frac{\text{T}}{\text{B}} \right\rceil \tag{6}$$

There are two complimentary model metrics that are displayed when talking about training data: loss and accuracy. Loss is the metric that the machine learning model is trying to minimize, and is usually based in some form of entropic calculation. There are many different loss functions that work best depending on what one is trying to achieve with the machine learning model they are using; binary cross-entropy is good for binary classification, the type of classification being done for the CRV. It is defined as:

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i)), \tag{7}$$

where y_i is the label of the event (1 if the event is a cosmic-ray muon induced event, 0 otherwise), $p(y_i)$ the calculated output of the machine learning model for that event (which is between 0 and 1, and thus treated as a probability), and N the number of events in the dataset. On the other hand, accuracy is simply the number of events that the model gets right after the internal model matrices/parameters have been updated for each epoch after setting the classification cutoff at 0.5 (see Section 10.1 for a discussion of the classification cutoff). While accuracy has little to nothing to do with the actual training of the machine learning model, it is a more intuitive metric than loss, and is therefore usually included when analyzing model training.

Overtraining occurs when the model cannot generalize to data outside of those in the training dataset, yet performs quite well on the training set itself. The hallmark of an overtrained model is when the model improves its accuracy on the training dataset at the expense of the validation dataset [9]. An analogy for machine learning overtraining is fitting a high-order polynomial order to a curve. While the polynomial definitely fits the data better, it doesn't extend itself to data points away from the dataset.

The hallmark of overtraining is an improvement to the metrics of the training dataset with a simultaneous worsening of the metrics to the validation dataset, as shown in Fig. 17 below. This is also one of the reasons for good, independent validation and testing datasets - they have to be properly independent from the training sample in order to be a good indicator of performance.

8.4 Optimizing Model Parameters

The hyperparameters of batch size, model width, and model depth were varied using a grid search, and the best option selected. A grid search entails looking through all possible combinations of given hyperparameters, where the selected choices are supplied manually. The choice of "best" model is determined by its model accuracy on the validation dataset at the end of training.

A larger batch size tends to wash out generalizations in the model [10], so a smaller batch size relative to the size of the dataset was chosen. However, a batch size that is too small would be just as bad, as given the relatively small size of the neural network, the network

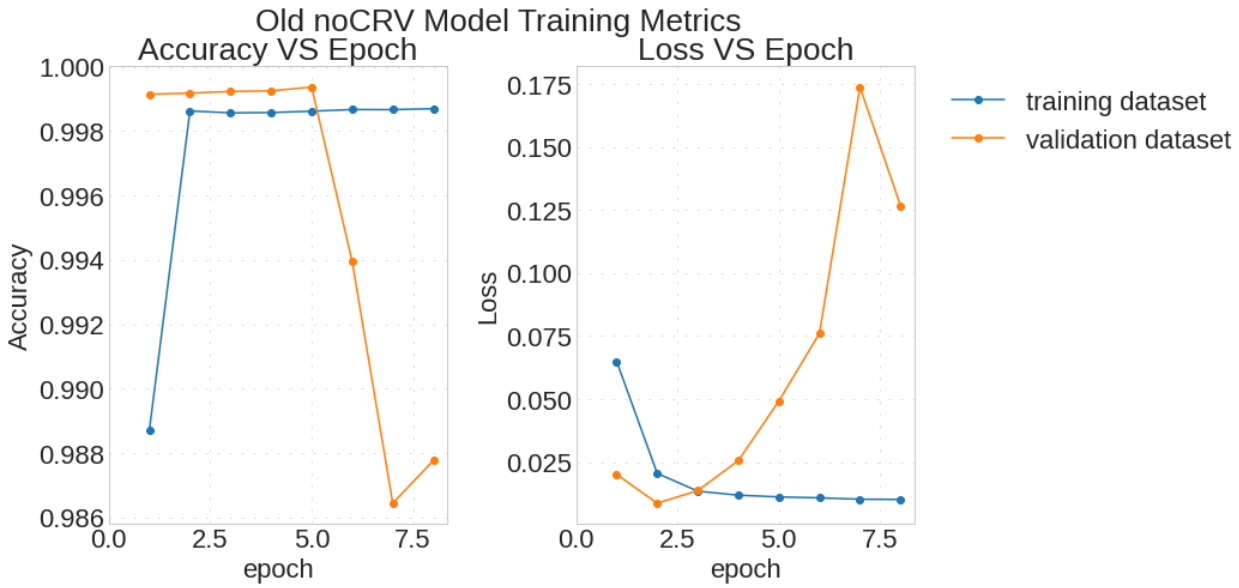


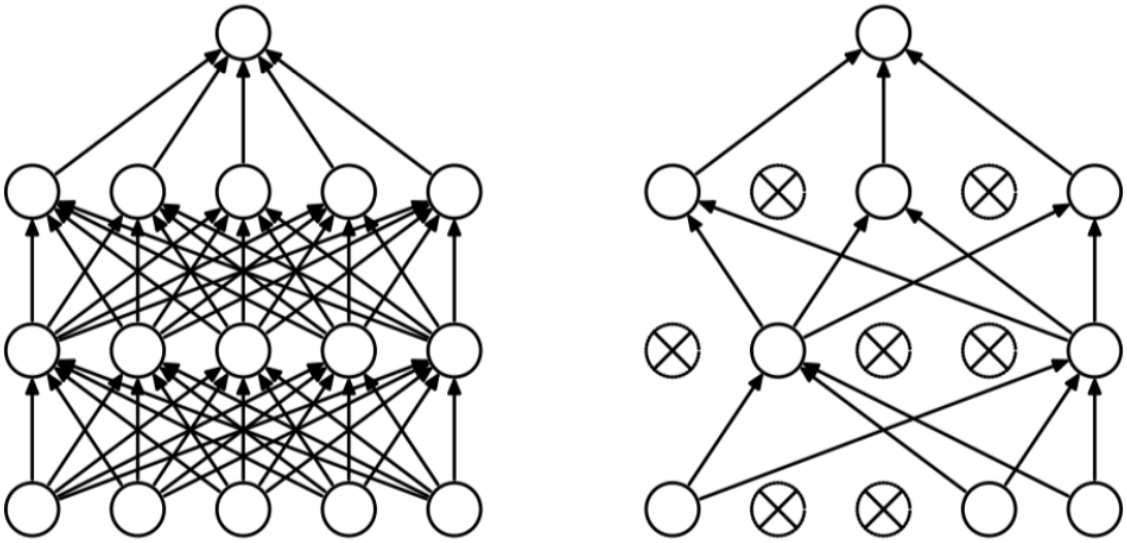
Figure 17: The training metrics from an older version of the noCRV model which was overtrained. The left plot depicts model accuracy vs epoch, and the right model loss vs epoch. Note the decreasing efficacy of the model with regards to the validation dataset, while the training dataset continues to improve.

would be overtrained within just a single epoch, making for a worse model overall. As such, a middle ground was chosen, which was reflected in the final choices made in Section 8.8.

Dropout within a model entails each neuron within a layer, per epoch, having a chance to be randomly "dropped out", or not included, in the training of the model, with the rate of this dependent on the dropout rate selected for each layer. For instance, a dropout rate of 0.2 would mean a random fifth of the neurons in each layer would be selected and dropped out. A rendition of what dropout entails is depicted in Fig. 18.

Dropout during training helps to prevent overfitting in the model, as it forces the model to consider different types of correlations, rather than falling into the trap of relying too much on one attribute or neural network combination thereby helping its ability to generalize [11].

The model structure that was chosen was a width of 4 times the number of input variables, with 8 hidden layers (layers that are neither input nor output layers) no matter the number of input variables. Inside of the middle 6 hidden layers there was a dropout rate of 0.2. The first hidden layer did not have dropout functionality because it was directly connected to the input layer, and the last one did not have dropout because it connected directly to the output. Both of these layers should have all their neurons intact as the connections between the input and output layers are too important to ever be dropped out. The dropout rate of 0.2 helped to prevent overfitting [11] by the model, yet was not so large as to make training overly difficult due to excessive dropout. Different rates between 0.2 and 0.5 were tested, with 0.2 being selected as the best. However, model performance across all these rates was similar, indicating a level of stability within the model. The batch size chosen for the CRV model was 100 events per batch, which was both small enough to help with generalization and large enough to prevent overtraining within epochs. The batch size chosen for the



(a) Standard Neural Net (b) After applying dropout.

Figure 18: A rendition of what dropout entails. The first and third layers have a dropout rate of 0.4, and the second layer a dropout rate of 0.6 [11].

noCRV model was 10, as the training sample size was much smaller for that sample.

The variables used for each model are detailed in Table 4, and a graphical rendition of the two models in Fig. 19.

CRV Model Variables	**noCRV Model Variables**
Downstream number of hits	Downstream number of hits
Particle ID	Particle ID
Track quality	Track quality
Starting z-value of track	Starting z-value of track
Distance from z-axis	Distance from z-axis
Recorded CRV z position	N/A
Recorded CRV ΔT	N/A
PE Yield of primary scintillators	N/A

Table 4: A table of variables used for the models. The CRV model used 3 more variables than the noCRV model.

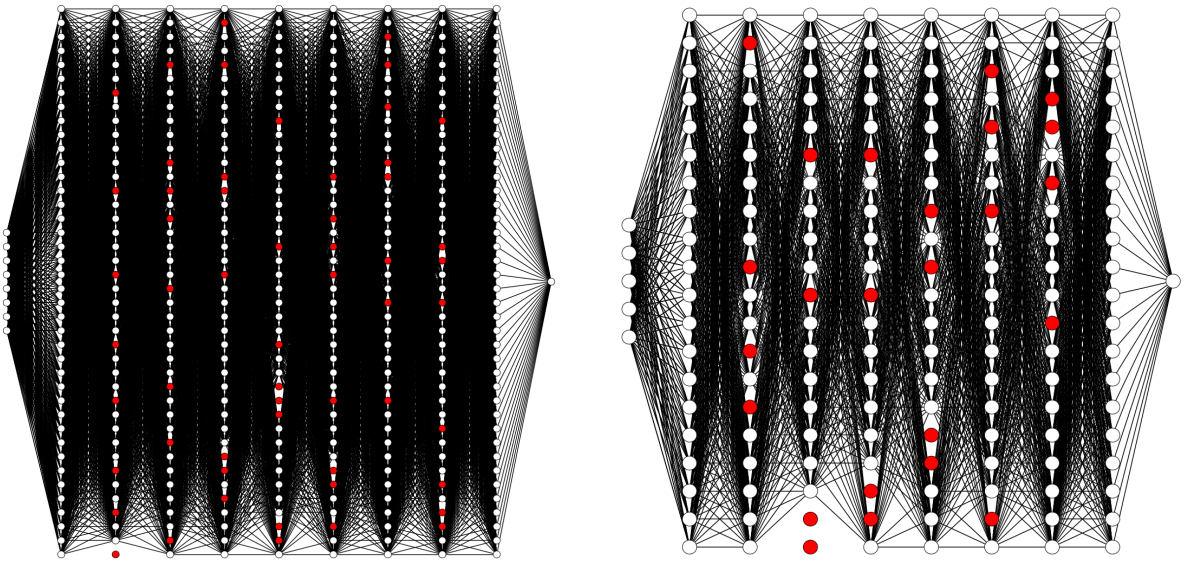


Figure 19: The models for the CRV(left) and the noCRV(right) models. The red dots represent dropped out neurons, as a representation of how many neurons would be dropped out. The input is on the left and output on the right for each model. Note that the first and last hidden layer do not have dropout functionality. The dropout rate for the hidden layers with dropout is 0.2.

9 Training the Model

9.1 The Alpha Metric

Neural networks consist of weight matrices for every neuron, as a way to actually evaluate the inputs they are given. These matrices can be analyzed to provide insight into the model itself, without needing access to the training data [12].

The α metric is an exponent determined by a model's weights such that the spectral density of the weight matrices, $\rho(\lambda)$, is approximately equal to λ^{α} for a given layer, where λ is an eigenvalue of the layer's weight matrix. It can effectively be a measure for data correlation [13]. This power law exponent's "healthy range" is usually between 2 and 6, where smaller is better (when within that range), and indicates a better correlation for that layer.

The α metric was also used during training, as an α value less than 2 corresponds to the overtraining of a layer. As such, the average α over the every layer of the model was examined at the end of every epoch, and if the value of α_{avg} was less than 2.1, training would stop, as the model was close to being overtrained. The reason for stopping at 2.1 as opposed to exactly two was to provide a safety buffer. The analysis of this metric was done using the "WeightWatcher" package for Python [14].

However, α_{avg} is not a perfect metric. The overtraining plot from section 8.3 had an α_{avg} value greater than 2, but was still clearly overtrained. While α serves as a good warning for overtraining, a manual examination is still always required.

9.2 Training Procedure

Both the noCRV and CRV models were trained on the CRY3 cosmic dataset alongside the training/validation subset of the CE/Noise sample, as stated before. The batch sizes were also the same as stated before for each of the models, and so were the training and validation dataset sizes and procedures for creating them. The optimizer used was the "adam" optimizer, which is a standard optimizer from the Keras package, and the kernel was initialized to a normal distribution (the weight matrices in the model were initially a randomized normal distribution) [7]. The training set accuracy/loss, validation set accuracy/loss, and alpha over the course of model training for both models is depicted in Fig. 20 and Fig. 21.

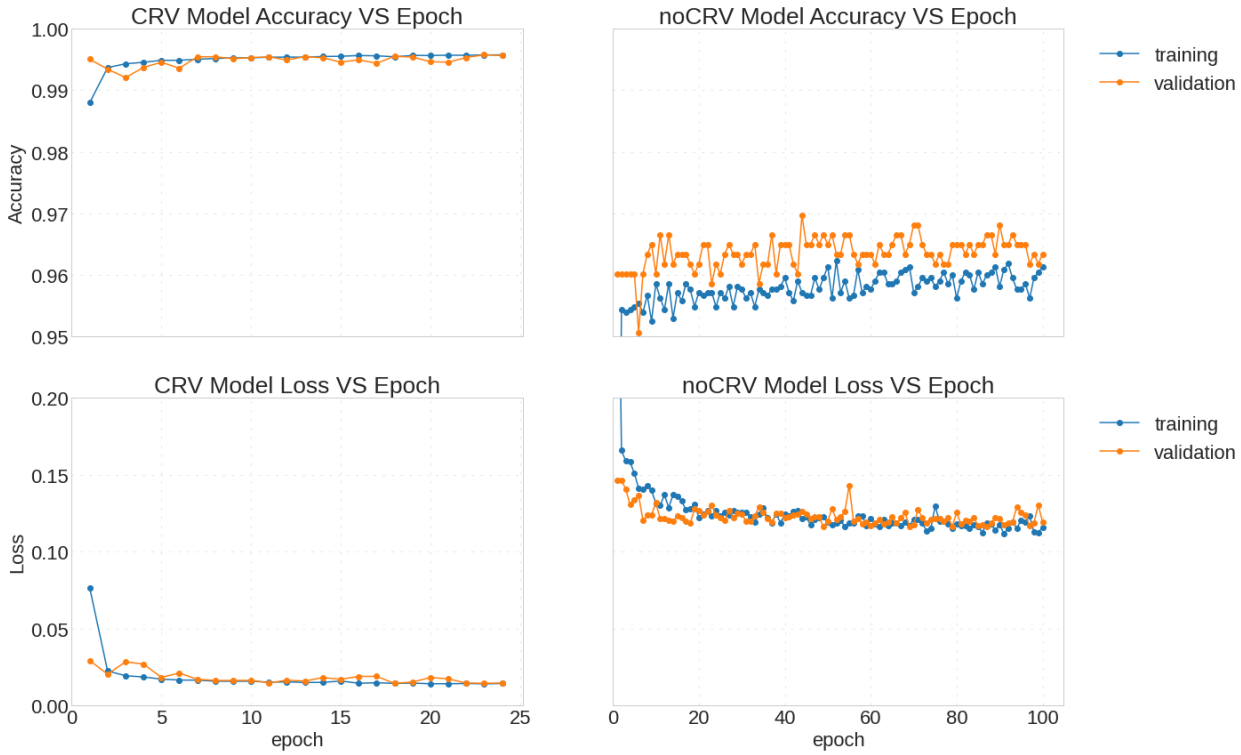


Figure 20: The accuracy and loss for the CRV model on the left, and the noCRV model on the right. Vertical axes are shared for both rows. Note the higher loss and lower accuracy value for the CRV model as opposed to the noCRV model.

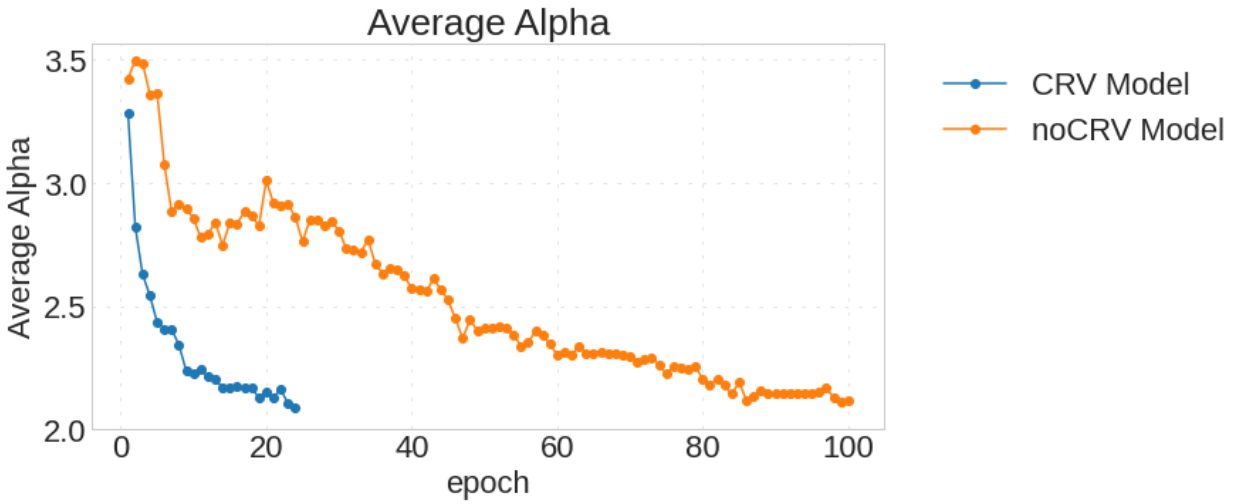


Figure 21: The average α over all layers of both models versus epoch.

The small batch size of the CRV model, relative to the size of the input dataset, allows for a much smaller number of epochs, whereas the batch size of the noCRV model relative to the size of its input dataset was larger, and is reflected in the increased number of epochs in Fig. 20 for the noCRV model.

The relatively smooth curve towards an alpha value of two for both models also reflects a better convergence towards a good model in Fig. 21, as opposed to a jagged curve[12]. This is because since lower alpha correlates to better correlation for the weight matrices, a smoother curve means that the matrix was correlated well through training, and didn't encounter many hitches along the way.

10 Model Predictions

10.1 Prediction procedure

The testing data, which was separated into the entire CRY4 dataset and the testing subset of the CE/Noise dataset, was scaled using the same values used to normalize the data from Section 8.2 to having a mean of 0 and a standard deviation of 1. These scalers put the testing data in the same relative scale, meaning that while the testing data would not have a mean of 0 and a standard deviation of 1, the data would be relative in those metrics to the data the model was trained on.

Afterwards, Loose Cuts were applied to the CRY4 dataset. This was done to conserve on computing resources, as the entire CRY4 dataset was quite large, and the loose cuts cut eliminated a large number of the events in the dataset, yet did not affect the final outcome. This step was not done for the testing portion of the CE/Noise dataset, as it had already passed Extended Momentum Cuts, as stated in Section 8.2. Predictions proceeded for both datasets after this step.

All of the predictions from the model are between 0 and 1, and represent the model's confidence that an event was a cosmic event. For example, if an event that the model

analyzes is output at a value of 0.64, the model believes that there is a 64% chance it is a cosmic event. A cutoff has to then be defined, and everything greater than the cutoff is classified as a cosmic, and everything less than the cutoff was classified as a CE.

10.2 CRY4 Sample Predictions

Figures 22 through 27 display each light yield in the CRY4, or cosmic, sample. They will all have the following:

1. An overall prediction plot of events input into the CRV and noCRV models before and after the Extended Momentum Cuts.

2. A breakdown for events input into the noCRV model before and after the Extended Momentum Cuts, split between high and low energy muons.

3. A breakdown for events input into the CRV model before and after the Extended Momentum Cuts, split between high and low energy muons, alongside a zoomed in portion to help determine behavior in the region where a strict cutoff would be placed (see Section 11.3 for a discussion on the classification cutoff). This zoomed in portion is differentiated between events that exist after Loose Cuts, and those after the Extended Momentum Cut.

The predictions for the CE/Noise sample are also plotted at the end in Fig. 28.

Figure 22: Predictions for the subset of the CRY4 sample that has a lightyield of 7,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

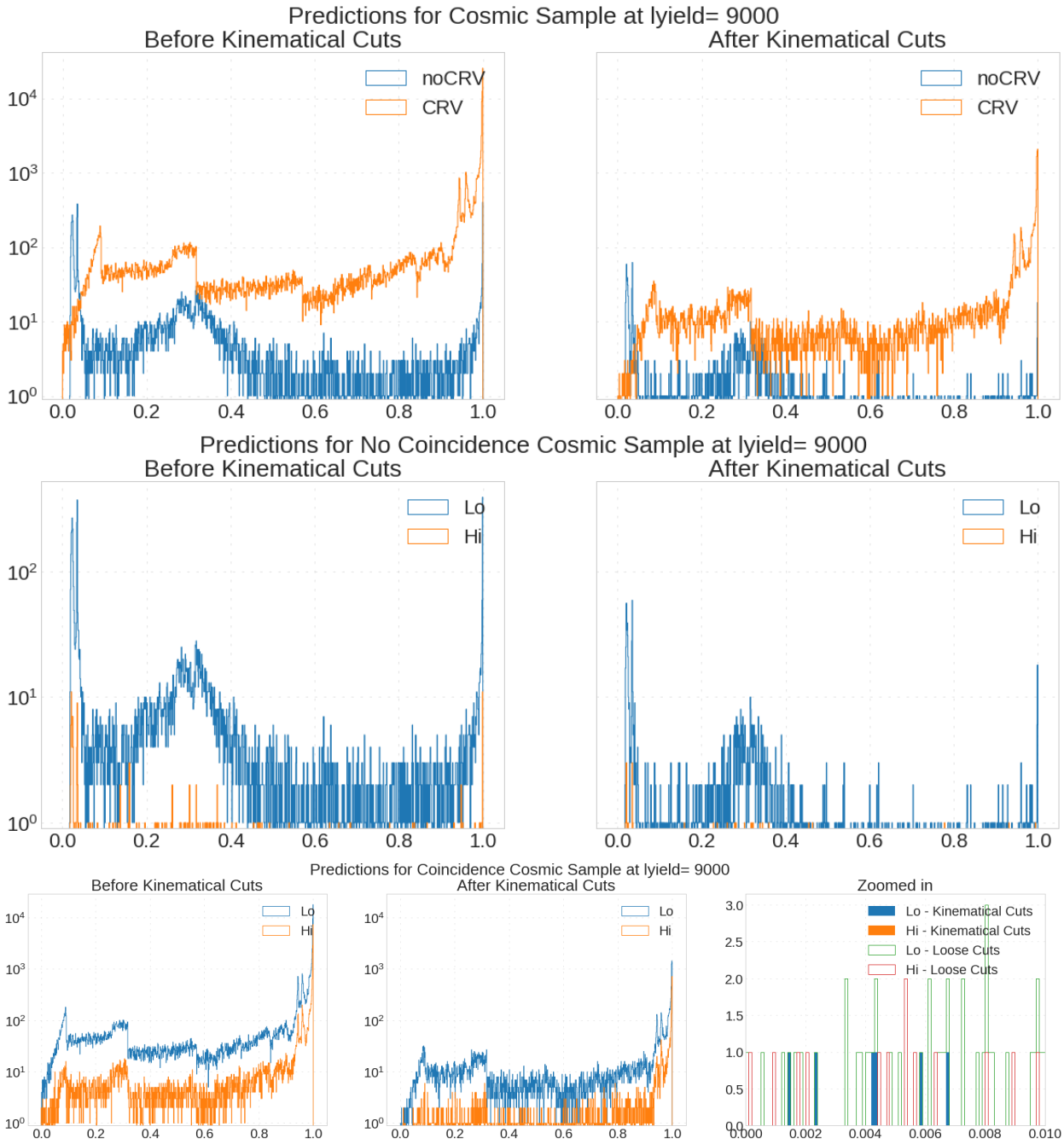


Figure 23: Predictions for the subset of the CRY4 sample that has a lightyield of 9,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

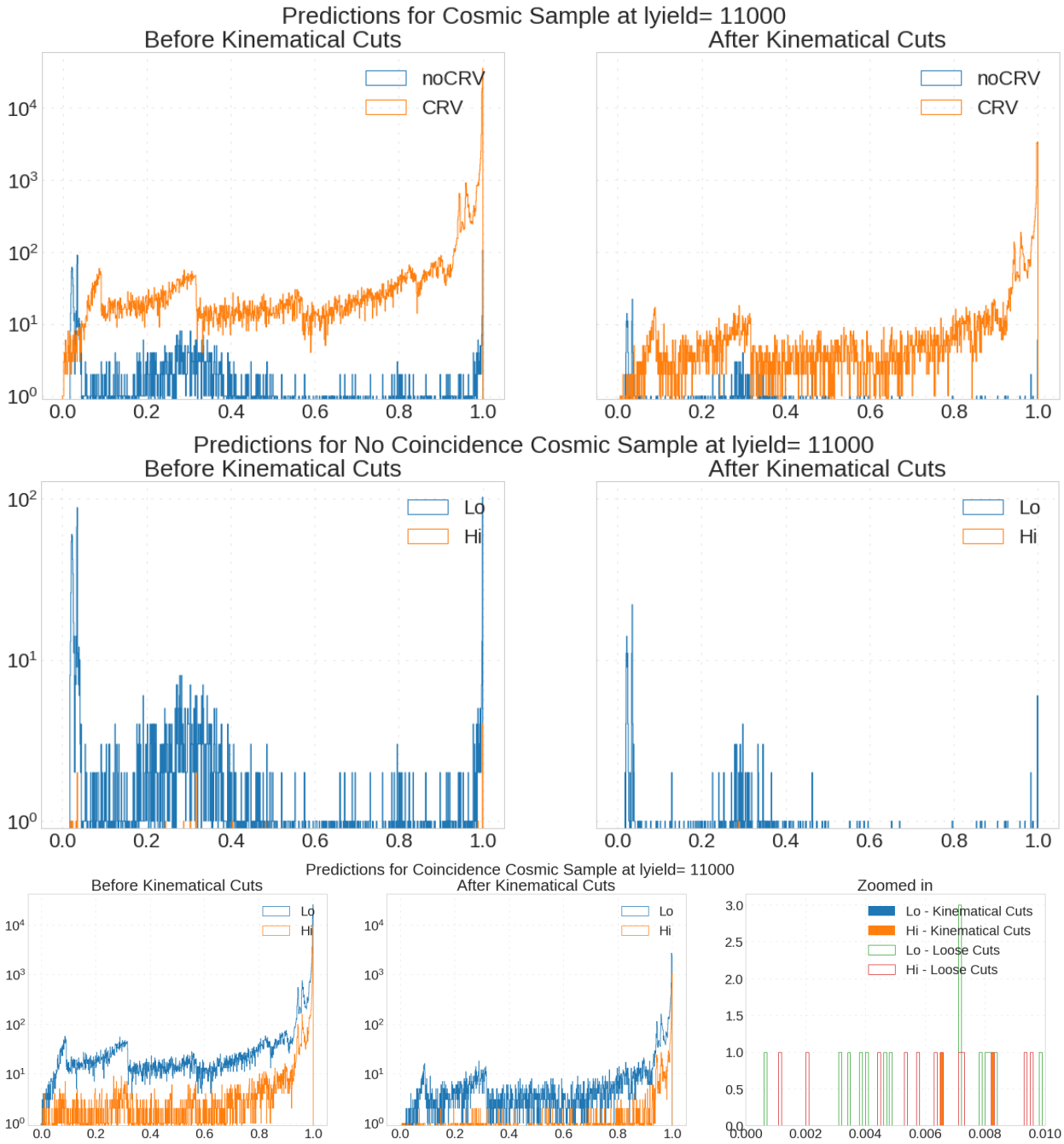


Figure 24: Predictions for the subset of the CRY4 sample that has a lightyield of 11,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

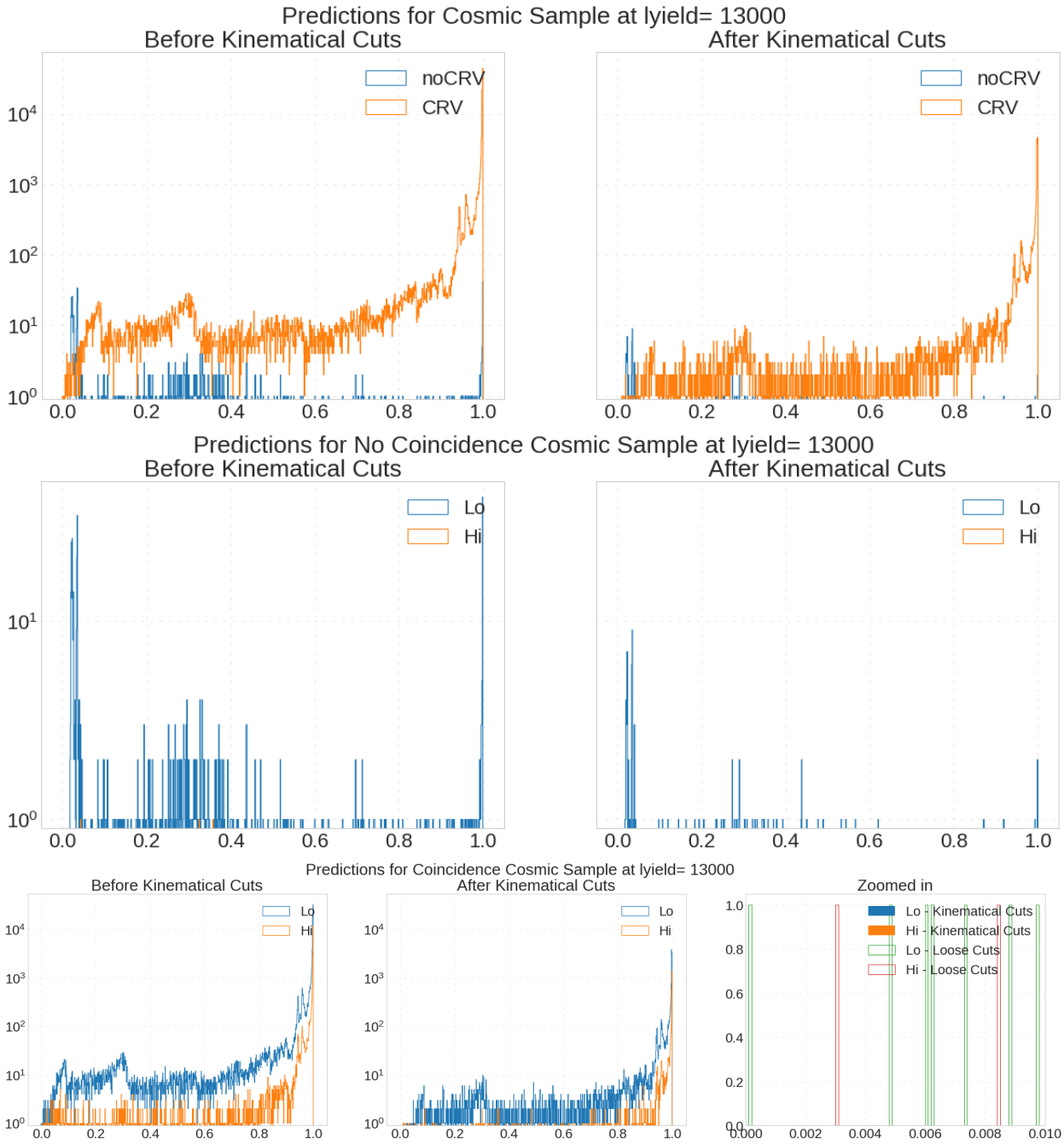


Figure 25: Predictions for the subset of the CRY4 sample that has a lightyield of 13,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

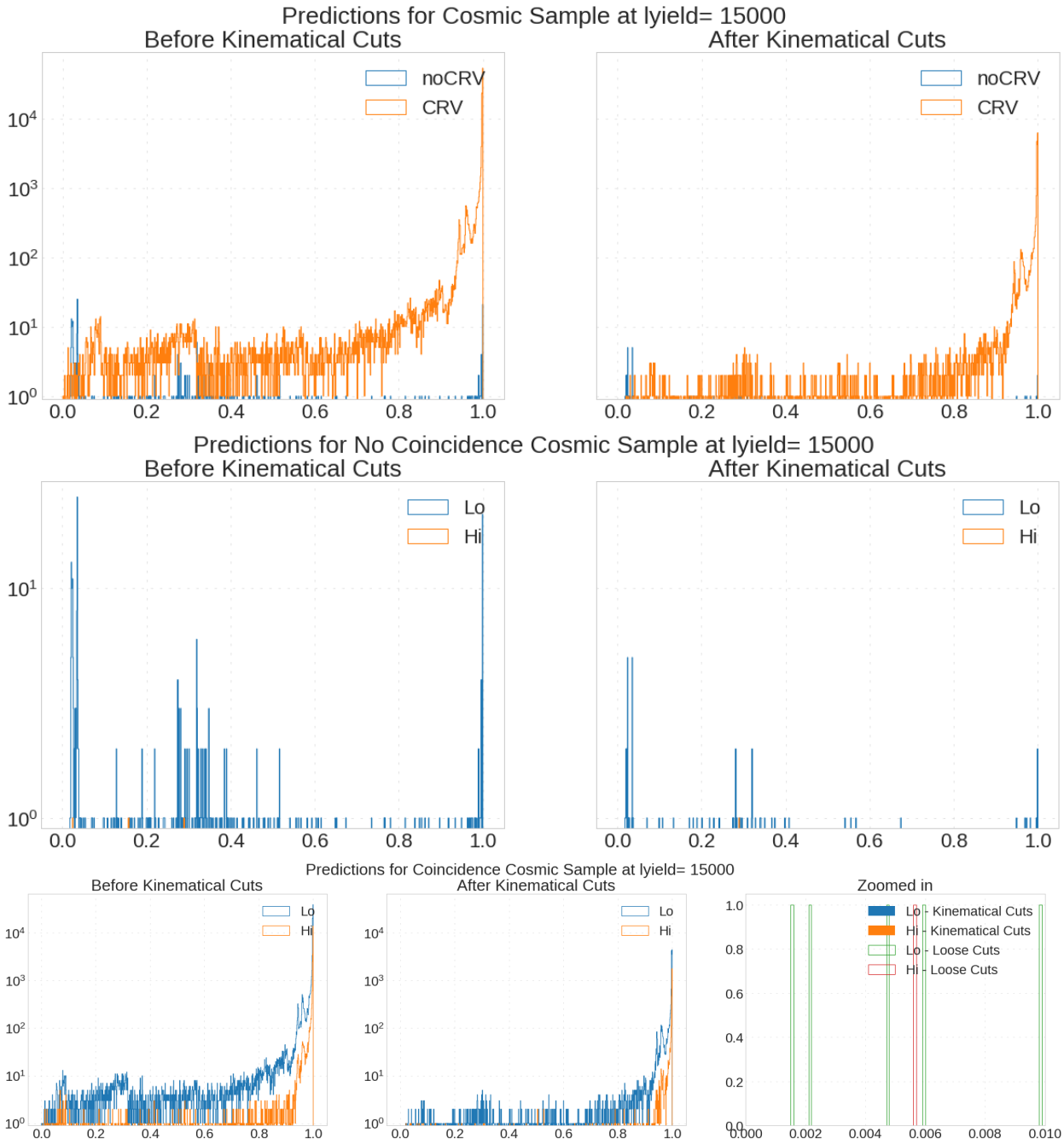


Figure 26: Predictions for the subset of the CRY4 sample that has a lightyield of 15,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

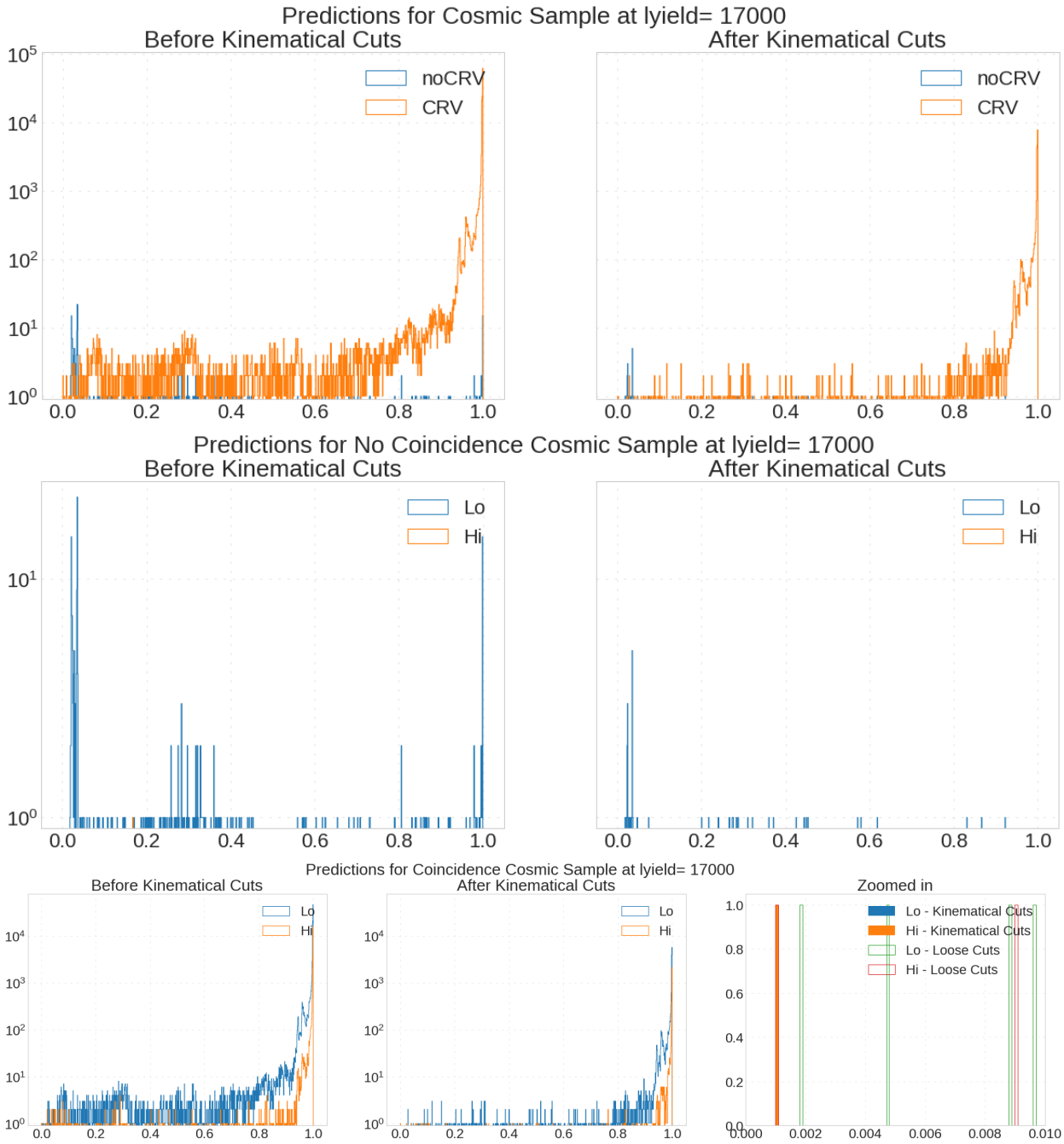


Figure 27: Predictions for the subset of the CRY4 sample that has a lightyield of 17,000. "Lo" (in blue) indicates low energy muons, and "Hi" (in orange) indicates high energy muons for the plots in the middle and bottom.

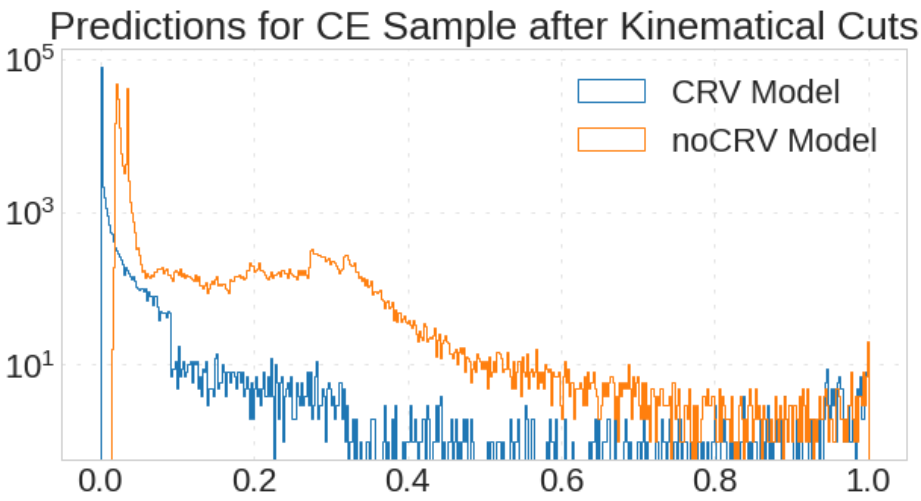


Figure 28: Predictions for the testing subset of the CE/Noise dataset for both the CRV and noCRV models. Blue indicates the prediction values of the CRV model, and orange the prediction values of the noCRV model.

11 Prediction Analysis

11.1 Calculating the Cosmic-Ray Muon Induced Background and Deadtime

The cosmic background was calculated using the procedure described in Section 5.4. However, it was not done after the Kinematical Cuts with the Physical Momentum Cut, but instead the Kinematical Cuts with the Extended Momentum Cut, in order to increase the statistics of the sample. In order to correct the background to the actual livetime of the experiment, the cosmic background was scaled by Eq. 9 below, which allowed for a ballpark measurement of what would occur during the actual experiment while maintaining high statistics.

$$\text{scale} = \frac{\#\text{ of events remaining after extended momentum cut}}{\#\text{ of events remaining after physical momentum cut}}, \tag{8}$$

$$\text{bkg}_{\text{corrected}} = \frac{\text{bkg}}{\text{scale}}, \tag{9}$$

$$\Delta\text{bkg}_{\text{corrected}} = \frac{\Delta\text{bkg}}{\text{scale}} \tag{10}$$

Whenever the cosmic background is said to be "normalized", it has been both scaled and measured relative to the livetime of one run of the Mu2e experiment, as detailed in Section 5.4.

The deadtime for the sample was calculated using the Physical Momentum Cut the same way as Section 5.3.

11.2 The Performance of the Two Models

The true positive rate compared to the false positive rate of the model is shown in the Receiver Operating Characteristic (ROC) curves in Fig. 29. The true positive rate is the rate at which cosmic-ray muon events are correctly identified, and the false positive rate is the rate at which CE/Noise events are identified as cosmic-ray muon events. Their formulas are given below:

$$\text{True Positive Rate} = \frac{\text{TC}}{\text{TC} + \text{FCE}}, \tag{11}$$

$$\text{False Positive Rate} = \frac{\text{FC}}{\text{FC} + \text{TCE}}, \tag{12}$$

where TC is the number of cosmic-ray muon events correctly identified by the model, FCE the number of cosmic-ray muon events misidentified as CE/Noise events by the model, FC the number of CE/Noise events misidentified as cosmic-ray muon events by the model, and TCE the number of CE/Noise events correctly identified by the model. A ROC Curve varies the cutoff value, and calculates the false positive rate and determines the true positive rate from that.

The AUC (Area Under the Curve) should be as close to 1 as possible, since an AUC of 1 indicates 100% accuracy with no false positives. It is a good indicator for model performance, as it shows how many false positives a model requires for a given sensitivity.

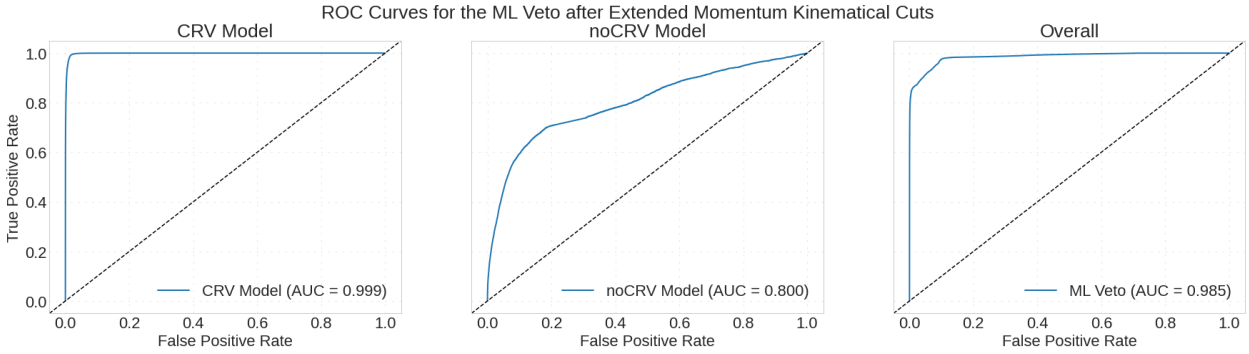


Figure 29: The ROC curves for both models, with the CRV model on the left and the noCRV model in the middle. The ROC curve for the models combined is on the right.

The performance of the noCRV model is significantly worse, as pointed out in section 8.1. None of the variables in the noCRV model really correlate to cosmic status, making prediction quite difficult. The role of the noCRV model was simply to separate at least some of the cosmic events from the rest, not necessarily to optimize the separation. The real classification power came from the CRV model, where a majority of the cosmic muons were.

11.3 Providing a Classification Cutoff

The value of the classification cutoff is the largest factor in determining whether an event is a cosmic muon or a CE, as it is the final determining cut for the ML Veto Algorithm.

The ROC Curve does not provide information on the cosmic-ray muon induced background, however, and so the direct relationship between the classification cutoff and the cosmic-ray muon induced value was investigated. The plots below depict the cosmic-ray induced background and deadtime for varying cutoffs, and were used to determine what cutoff to use. Since the goal was to be either consistently the same or better than the CRV Time Window Veto Algorithm, in terms of cosmic background and deadtime, the cutoff determination was made using these metrics. Figures 30 through 35 display the cosmic-ray muon induced background versus the cutoff value, with the uncertainty produced by Eqns. 5 and 10. Figure 36 displays the deadtime versus the cutoff, with the uncertainty produced by Eq. 3.

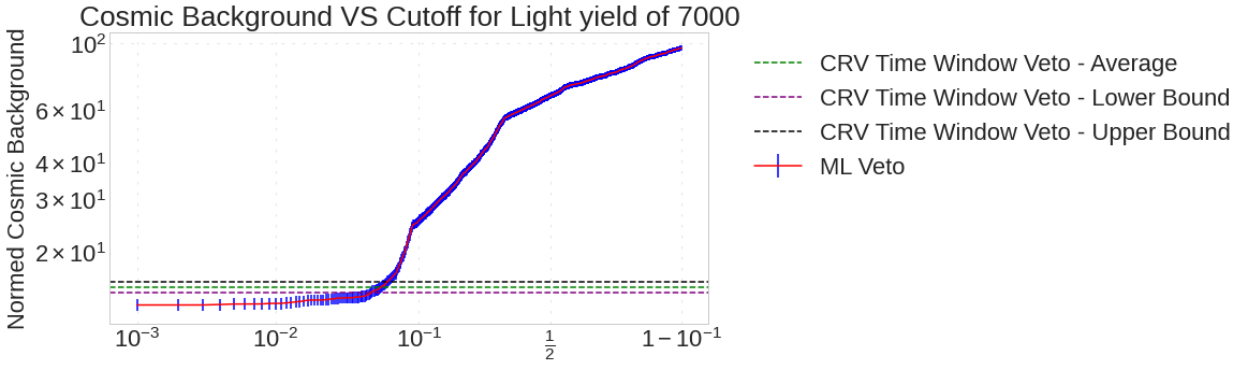


Figure 30: The cosmic-ray induced background versus prediction cutoff value for a light yield of 7,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

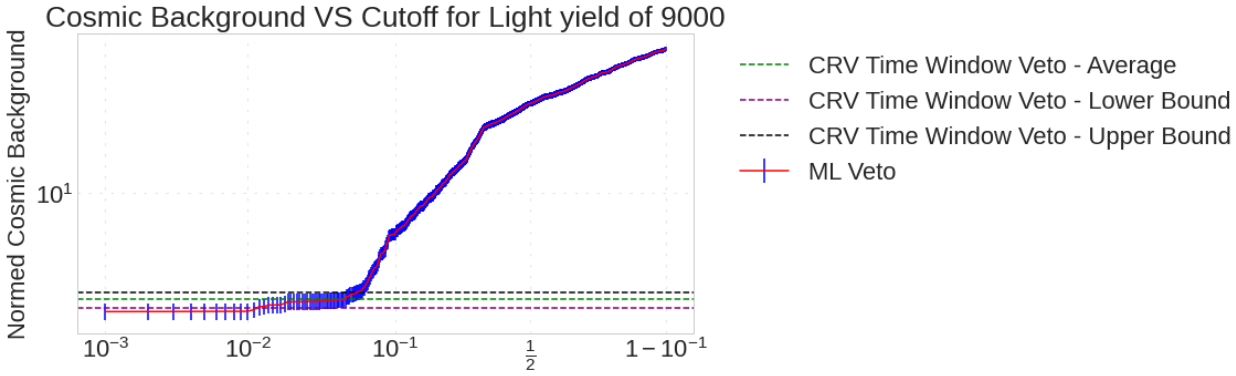


Figure 31: The cosmic-ray induced background versus prediction cutoff value for a light yield of 9,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

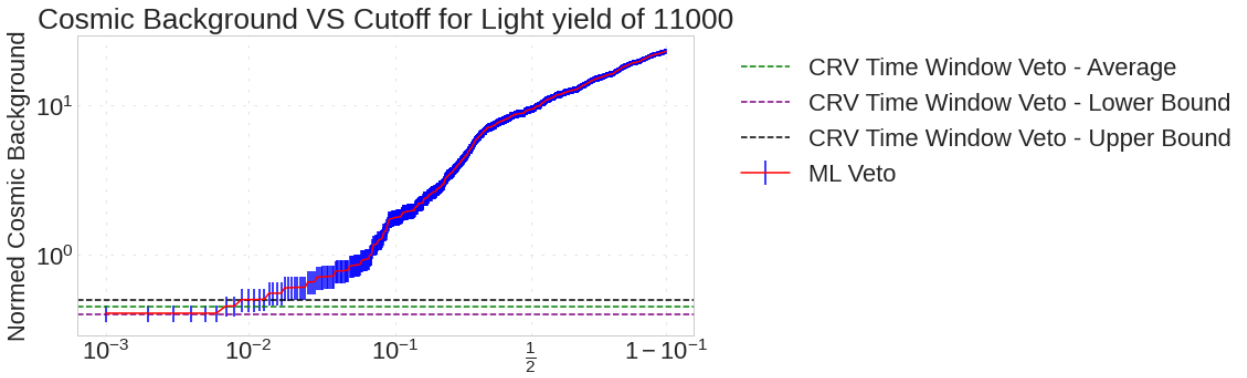


Figure 32: The cosmic-ray induced background versus prediction cutoff value for a light yield of 11,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

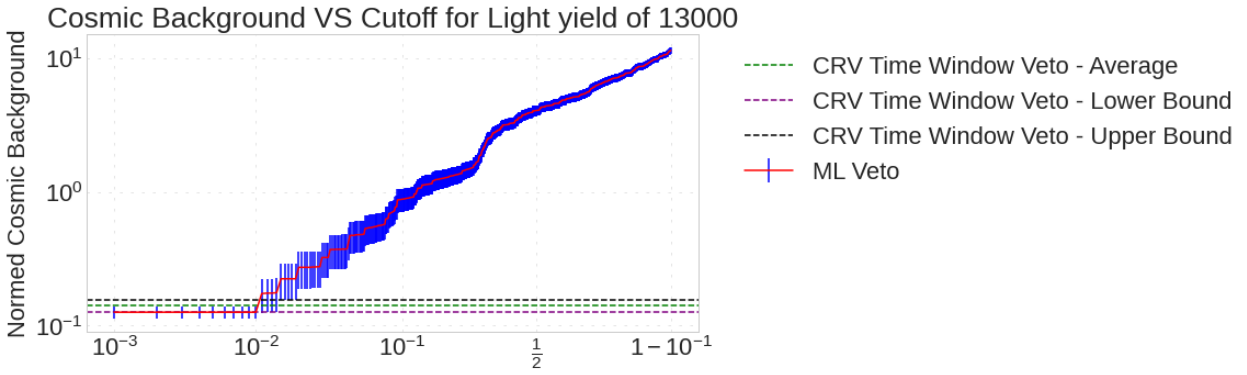


Figure 33: The cosmic-ray induced background versus prediction cutoff value for a light yield of 13,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

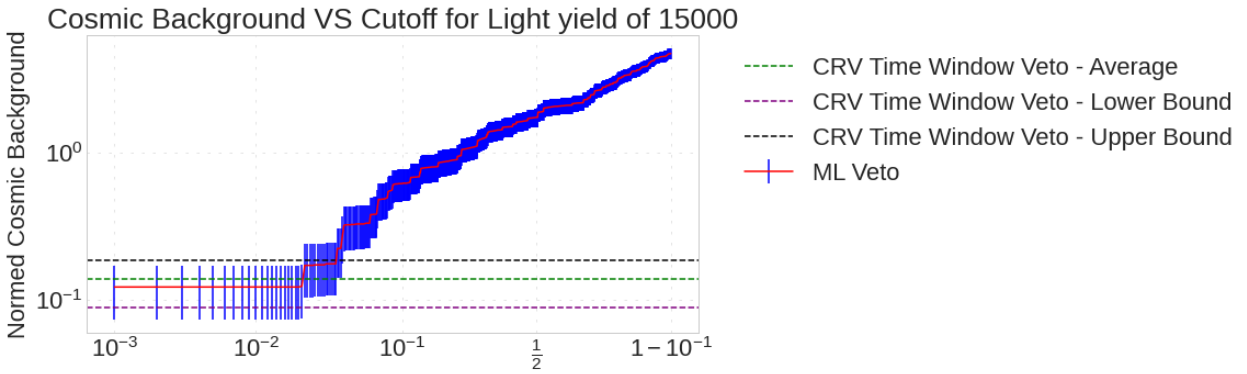


Figure 34: The cosmic-ray induced background versus prediction cutoff value for a light yield of 15,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

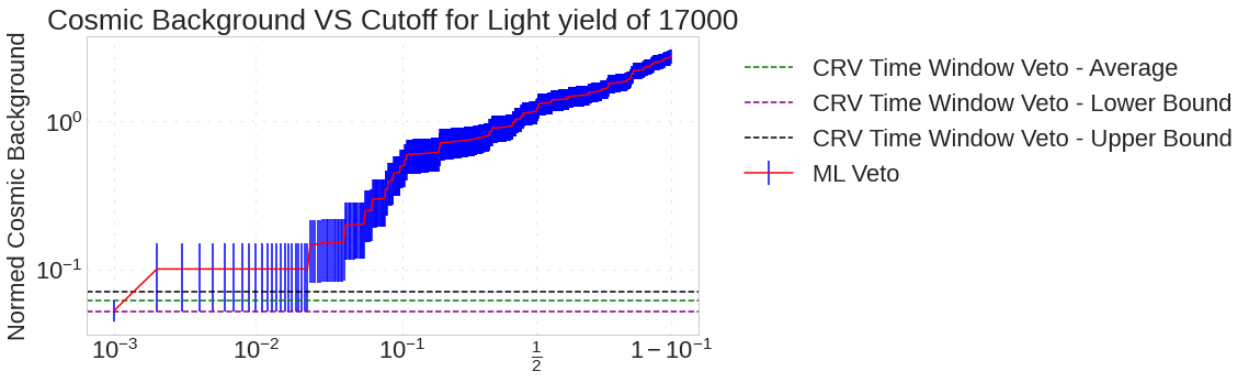


Figure 35: The cosmic-ray induced background versus prediction cutoff value for a light yield of 17,000. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

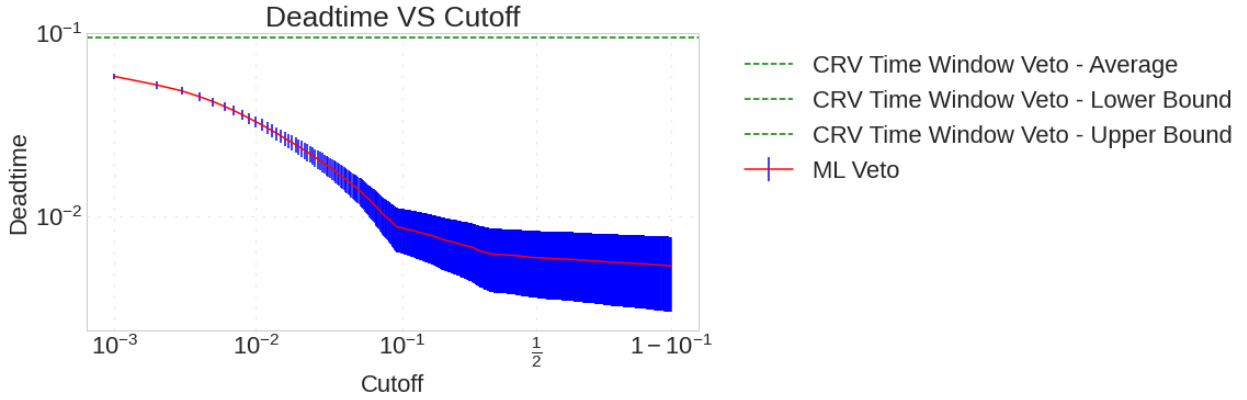


Figure 36: The deadtime versus prediction cutoff value. The only cutoff value that was changed was the cutoff for the CRV model. The cutoff for the noCRV model was kept at 0.5.

The final classification cutoff was crafted in such a way that the cosmic background would be minimized. Since the vast majority of the cosmic-ray muon events are handled by the CRV model, and a large number of CE/Noise events are handled by the noCRV model, the cutoff was established at 0.5 for the noCRV model. Other cutoff values for the noCRV model were explored, but it was settled that a value of 0.5 was the best.

For the CRV model, the cutoff value was set to match or improve upon the CRV Time Window Veto in terms of performance with regards to the cosmic-ray induced background. The three cutoffs looked at in depth were at values of 0.001, 0.005, and 0.010. A cutoff greater than 0.010 resulted in a cosmic background that was too large.

12 Physical Metrics

The output of both the noCRV and CRV models were combined to form one output dataset, which was what was analyzed for this study. This combined output of these two models formed the "ML Veto" seen moving forwards.

12.1 Cosmic Background

The ML Veto performs consistently better than the CRV Time Window Veto at the cutoff value of 0.001 in terms of both cosmic-ray induced background and deadtime. For every light yield except the highest at 17,000, the other two cutoff values selected does either better or within one Δ of the CRV Time Window Veto, with Δ being calculated using Eqs. 5 and 10. The cosmic-ray induced background vs light yield for the three cutoffs alongside the CRV Time Window Veto are plotted in Fig. 37 below.

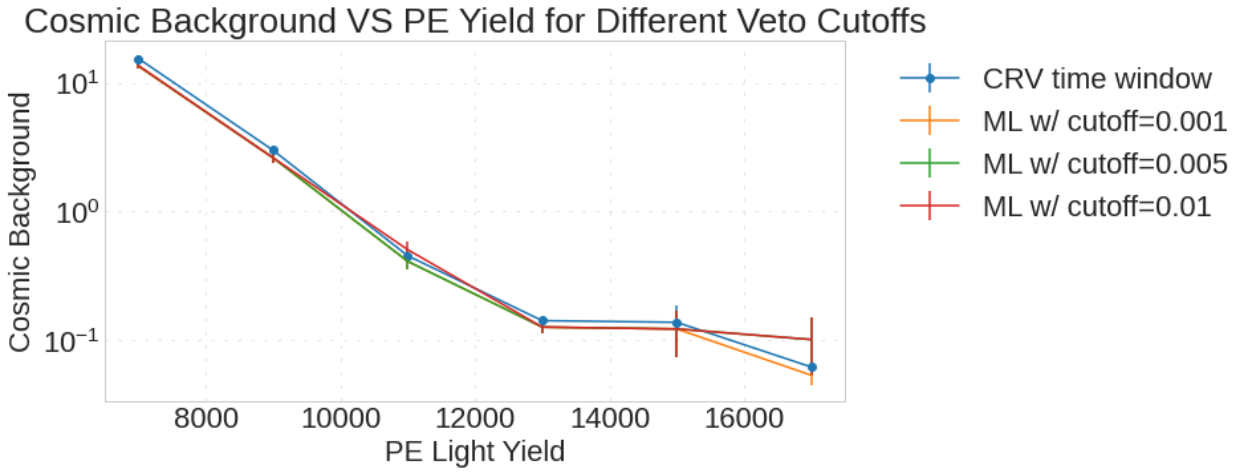


Figure 37: The cosmic-ray induced backgrounds for different light yields. The noCRV model cutoff was kept at 0.5, with only the CRV cutoff changing.

One notable trait of the ML Veto is its success in regards to low and high energy muons. The ML Veto often performs worse than the CRV Veto for high energy muons for the higher cutoff values at higher light yields, and the ML Veto performs consistently better than the CRV Time Window Veto for low energy muons at that scale. This can be seen in the plots in Appendix B.

There is also a structure evident in the prediction distributions, as seen by the peaks between 0 and 1 for the distributions of the CRY4 dataset in Section 10.2. It is not currently clear why this is, but it is a point of further study.

12.2 Deadtime

The deadtimes for the CRV Time Window Veto and the ML Veto Cutoff values are detailed below in Table 5 and Fig. 38. These deadtimes are over the testing subset of the CE/Noise dataset with no separation between beam intensities, or the number of protons per pulse from the production solenoid.

Method	Deadtime
CRV Time Window Veto	0.09419 ± 0.00225
ML Veto (Cutoff=0.001)	0.05775 ± 0.00229
ML Veto (Cutoff=0.005)	0.04211 ± 0.00231
ML Veto (Cutoff=0.010)	0.03271 ± 0.00233

Table 5: The deadtimes for the CRV Time Window Veto and the ML Veto.

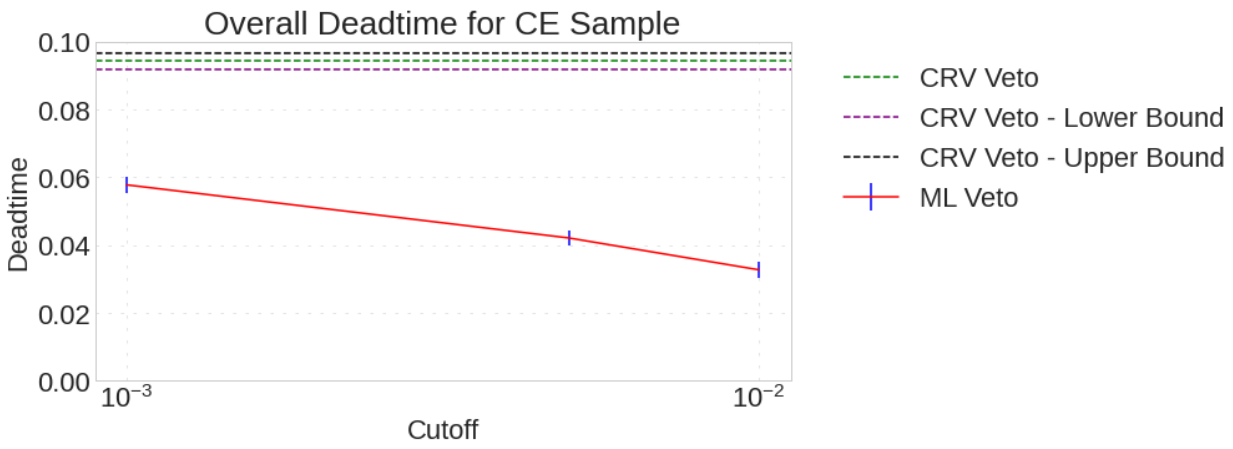


Figure 38: The overall deadtime for the three different cutoffs of the ML Veto compared with that of the CRV Time Window Veto over the entire testing portion of the CE/Noise dataset irrespective of beam intensity.

No matter the cutoff value, including the extremely strict cutoff at a value of 0.001 for the CRV model, the overall deadtime of the ML Veto performs much better than that of the CRV Time Window Veto, as shown in Fig. 38.

In reality, the beam produced by the production solenoid will have a set beam intensity, and Fig. 39 displays the deadtime for the ML Veto Algorithm versus the beam intensity. The ML Veto has less deadtime than the CRV Time Window Veto for every beam intensity except the lowest one. Even for the lowest beam intensity all the deadtimes are within each other's uncertainty bound.

There is a relatively large difference between the deadtime of the CRV Time Window Veto and the ML Veto for every cutoff available. This difference can be explained the by ΔT distribution for the ML Veto versus the current CRV Veto in Fig. 40.
Unlike the CRV Time Window Veto, the ML Veto lets through a number of CE events that are inside of the CRV Veto's time window, while still identifying and vetoing enough cosmic muons to remain relevant as a viable veto for the CRV itself, as displayed in Fig. 40 above. This produces a much lower deadtime for the ML Veto, even at the strictest cutoff value available.

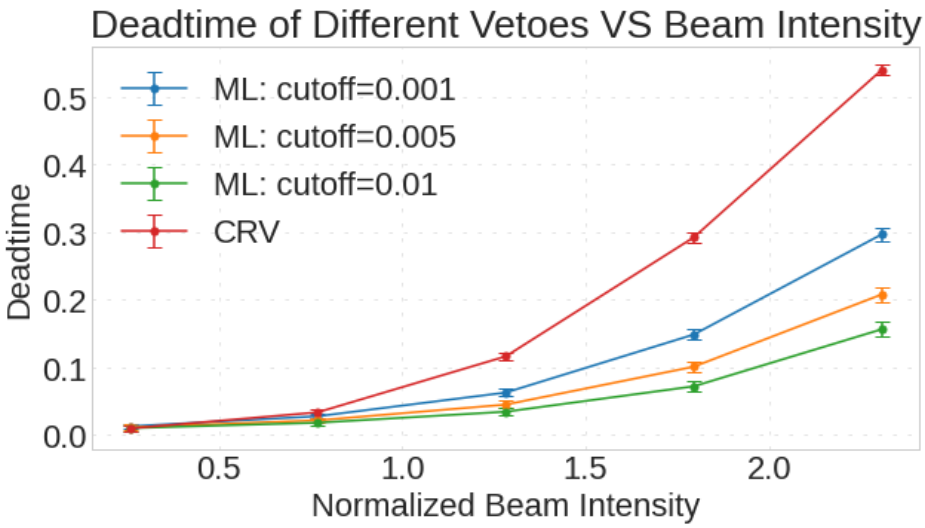


Figure 39: The deadtime of the ML Veto and CRV Time Window Veto versus the beam intensity. Beam intensity is relative to the nominal value of 3.9×10^7 protons per pulse.

13 Concluding Thoughts

13.1 Study Conclusion

This study is an investigation into whether a deep-neural network could perform better than the current CRV Time Window Veto. The ML Veto performs either the same or better than the CRV Time Window Veto in terms of both reducing the cosmic-ray muon induced background and the deadtime. For the strictest cutoff of 0.001 for the ML Veto, both the cosmic background and deadtime were improved upon at every light yield, which is very promising for the future of using machine learning for the Mu2e CRV.

13.2 Further Study

There are unexplained behaviors in the neural network currently, and it would do well to further study these behaviors. One such behavior is in the noCRV model's predictions for the CE Sample, where the entire prediction distribution never reaches zero, as shown in Fig. 28. Another such behavior is the structure in the predictions for the CRY4 sample in Section 10.2. There are peaks at certain values, and troughs at others in the prediction distribution that are currently unexplained. Exploring the cause of these oddities in the distribution would do much to help further understand how to optimize the machine learning model for the CRV Veto. Behaviors like these need to be understood and looked into thoroughly before moving forward.

Any model developed needs to be further tested for robustness and further tested against other backgrounds, such as the DIO (muons which Decay In Orbit) background, and other such backgrounds. Even though the cuts applied remove a large portion of these backgrounds, it would still be pertinent to check. This would be the only way to actually beat the CRV Time Window Veto, as the current veto algorithm is also optimized for these backgrounds.

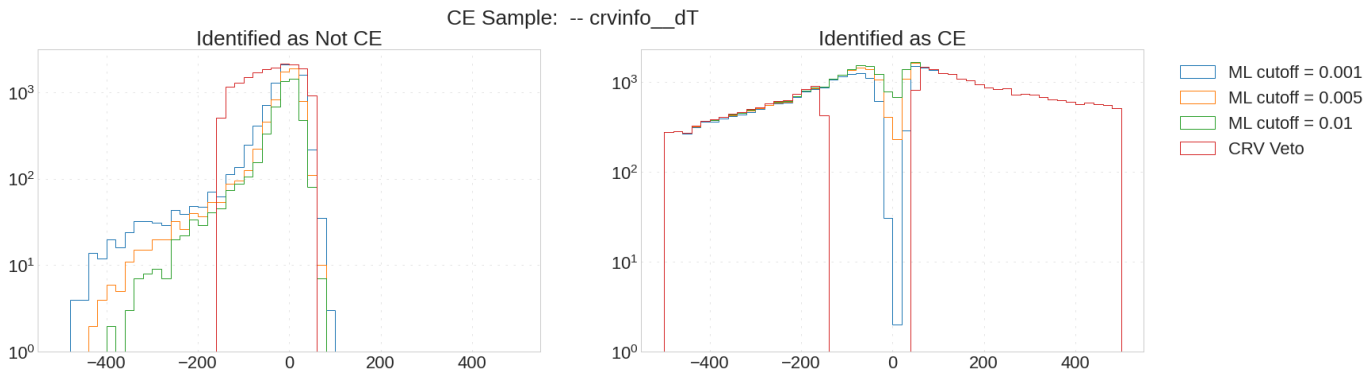


Figure 40: The ΔT distributions for the testing subset of the Noise/CE sample using different vetoes and their cutoffs. The left plot is a distribution of all events that were identified as "not CE", or set to be vetoed, and the right plot a distribution of all events identified as a CE. Note the increased number of Noise/CE events that are vetoed using the current CRV Veto (marked as "CRV Veto" in the plots).

Similarly, it would possibly be prudent to see a machine learning model's reaction to changing the reconstruction thresholds: for instance, changing the threshold for a valid muon track stub to 2/4 layers hit instead of 3/4, to allow for a higher efficiency. This could help gain more cosmic-ray induced CEs, but may also backfire, so it would be an interesting test to do in the future.

Currently the performance of the ML Veto, similar to the CRV Time Window Veto, is very dependent on the light yield of the sample. This should be improved upon due to aging concerns in the CRV (see Appendix A). Looking into methods making the machine learning model more resilient towards changes in the light yield, possibly through removing the PE yield from the model or other similar measures, is a point for further study.

It would also be prudent to look into different types of machine learning models, namely LSTM (Long Term Short Memory) models, and SVM (Support Vector Machine) models. CNNs (Convolutional Neural Networks) were tried in the past on images of events without the reconstruction algorithm alongside numerics, and remain of interest to the Mu2e Collaboration [5].

14 Acknowledgements

This work was done under the supervision and guidance of Dr. Yuri Oksuzian and Prof. E. Craig Dukes. Feedback and guidance was also provided by members of the Mu2e Collaboration over the course of the study.

15 References

[1] Fermi National Accelerator Laboratory, "Mu2e for physicists," https://mu2e.fnal.gov/public/index.shtml, 2021, [Online; accessed 2021].

[2] Y. Oksuzian, "A cosmic ray veto detector for the mu2e experiment at fermilab," https://indico.cern.ch/event/361123/contributions/856188/attachments/1135881/1625309/DPF2015_CRV_Overview.pdf, 2017, [Online; accessed 2022].

[3] E. C. Dukes, "Cosmic rays are a pain: The mu2e cosmic ray veto," http://muse.lnf.infn.it/wp-content/uploads/2017/08/mu2e_crv_seminar_2017_08_fnal.compressed.pdf, 2017, [Online; accessed 2021].

[4] CERN, "Geant4: A simulation toolkit," https://geant4.web.cern.ch/, 2022, [Online; accessed 2022].

[5] Y. Oksuzian, "Report on the 2020 LDRD expedition project: "improvements to cosmic muon identification at mu2e," 2020, [Online; accessed 2021].

[6] R. Coleman, "The mu2e muon beamline," *AIP Conference Proceedings*, vol. 1222, no. 1, pp. 387–390, 2010. [Online]. Available: https://aip.scitation.org/doi/abs/10.1063/1.3399349

[7] Keras, "Keras: the python deep learning api," https://keras.io/, 2021, [Online; accessed 2021].

[8] J. Brownlee, "Difference between a batch and an epoch in a neural network," https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/, 2019, [Online; accessed 2021].

[9] S. Teufel and A. Copestake, "Overtraining and cross-validation," https://www.cl.cam.ac.uk/teaching/1617/MLRD/slides/slides5.pdf, 2017, [Online; accessed 2021].

[10] C. Martin and M. Mahoney, "Statistical mechanics methods for discovering knowledge from production-scale neural networks," https://www.stat.berkeley.edu/~mmahoney/talks/dnn_kdd19_fin.pdf, 2019, [Online; accessed 2021].

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, 2014.

[12] C. H. Martin, T. Peng, and M. W. Mahoney, "Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data," *CoRR*, vol. abs/2002.06716, 2020. [Online]. Available: https://arxiv.org/abs/2002.06716

[13] C. Martin, "Weightwatcher: Empirical quality metrics for deep neural networks," https://calculatedcontent.com/2020/02/16/weightwatcher-empirical-quality-metrics-for-deep-neural-networks/, 2020, [Online; accessed 2021].

[14] ——, "weightwatcher 0.5.5," https://pypi.org/project/weightwatcher/, 2021, [Online; accessed 2021].

Appendices

A Aging of the CRV

Over time the scintillators within the CRV will age; their light yield will decrease. This aging has been happening at a faster rate than expected. This was studied in tandem with the main focus of this study, as the light yield is an important factor in determining the performance of the CRV veto. Over the course of about 2.5 years, the aging of different types of production counters was measured. These aging values were temperature corrected, following the equation below:

$$I_{\text{corrected}} = I(1 - m(T_0 - T)), \tag{13}$$

where I is the input current, m the slope, T_0 was 21 degrees C, the reference temperature, and T the temperature recorded of the channel at the date.

The data was taken by a test counter, with data collected every month over two years. The current produced by a source was recorded. It is directly proportional to the light yield, and is a more easily measureable quantity, so that is what was used to determine the aging rate.

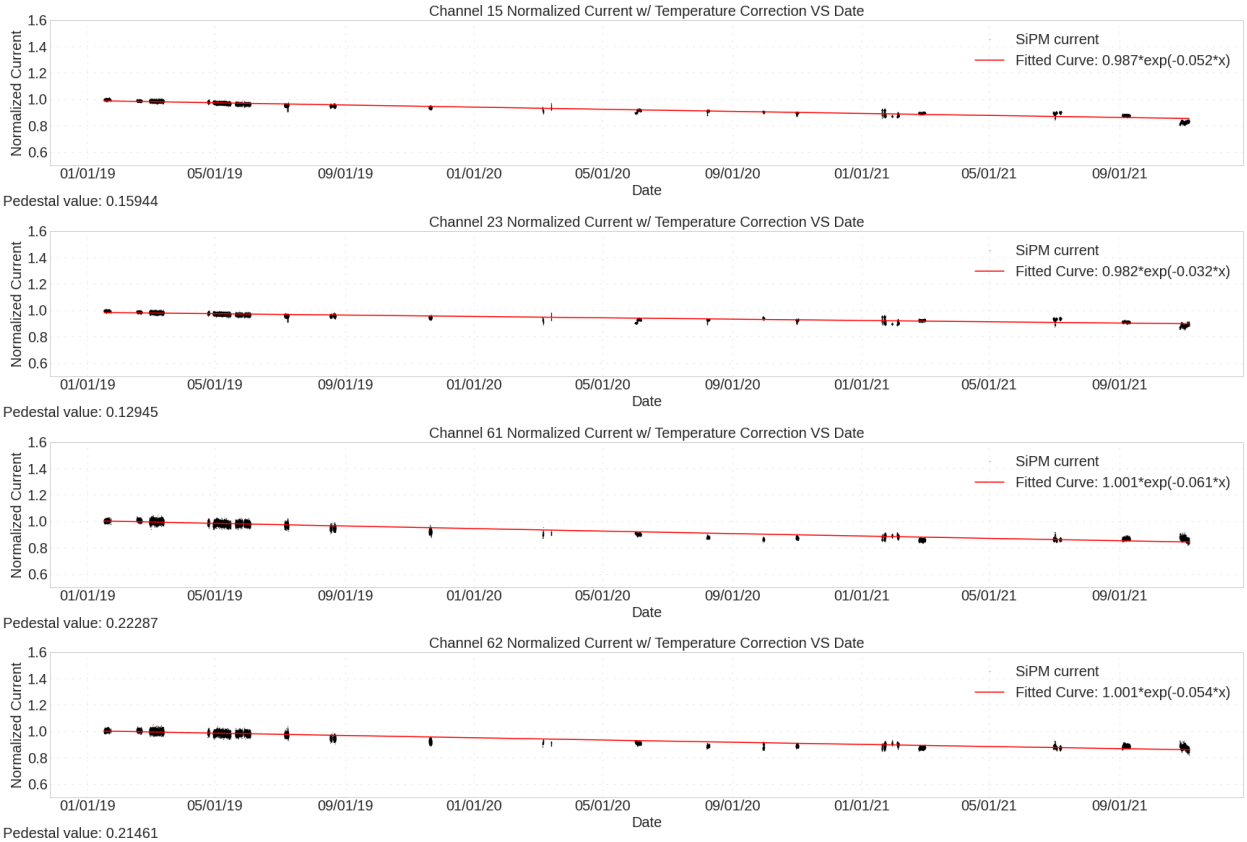


Figure 41: The temperature-corrected aging rates of different channels. The aging rate is the value of the fitted exponential.

As the light yield of the CRV decreases, anything related to the CRV will have to adjust to that accordingly, making aging an important issue for the Mu2e Collaboration.

B In-Depth Cosmic Background per Lightyield

The following plots are in-depth per cutoff breakdowns of Fig. 37 at cutoff values of 0.001, 0.005, and 0.010. All the uncertainties were determined using Eqs. 5 and 10. The "CRV Veto" indicates the current CRV Time Window Veto Algorithm (which was the same per light yield irrespective of the cutoff), and "ML Veto" indicates the ML Veto Algorithm developed in the study. They all contain the following, moving from left to right.

1. The average cosmic-ray muon induced background versus the three cutoff values for both low and high energy muon induced events. The CRV Time Window Veto Algorithm performance is also given as a straight line with uncertainty (as it is not dependent on the machine learning cutoff point.)

2. The cosmic-ray muon induced background versus the three cutoff values for high energy muon induced events. The CRV Time Window Veto Algorithm performance is also given as a straight line with uncertainty (as it is not dependent on the machine learning cutoff point.)

3. The cosmic-ray muon induced background versus the three cutoff values for low energy muon induced events. The CRV Time Window Veto Algorithm performance is also given as a straight line with uncertainty (as it is not dependent on the machine learning cutoff point.)

As can be seen in the plots, the ML Veto performs consistently either the same or better than the Time Window Veto in regards to the background due to low-energy muon induced background, and always performs either the same or better than the Time Window Veto at the lowest cutoff value for both the overall and high-energy muon induced background.

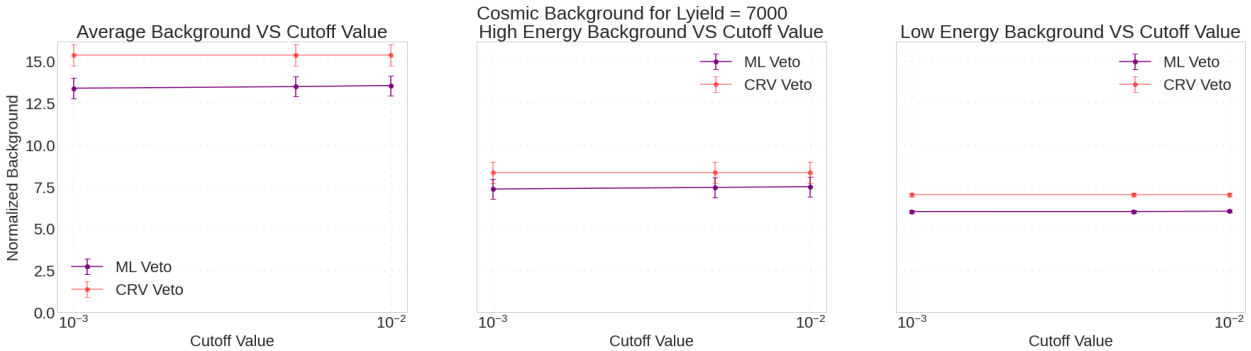


Figure 42: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 7000.

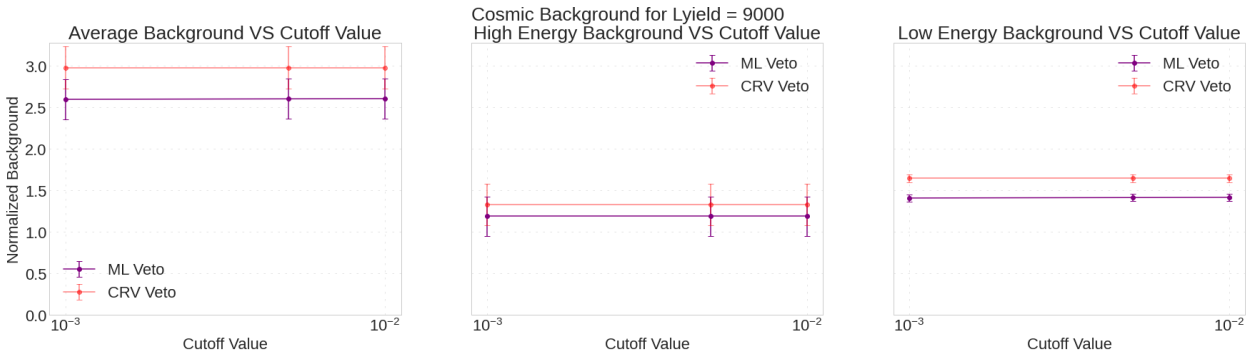


Figure 43: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 9000.

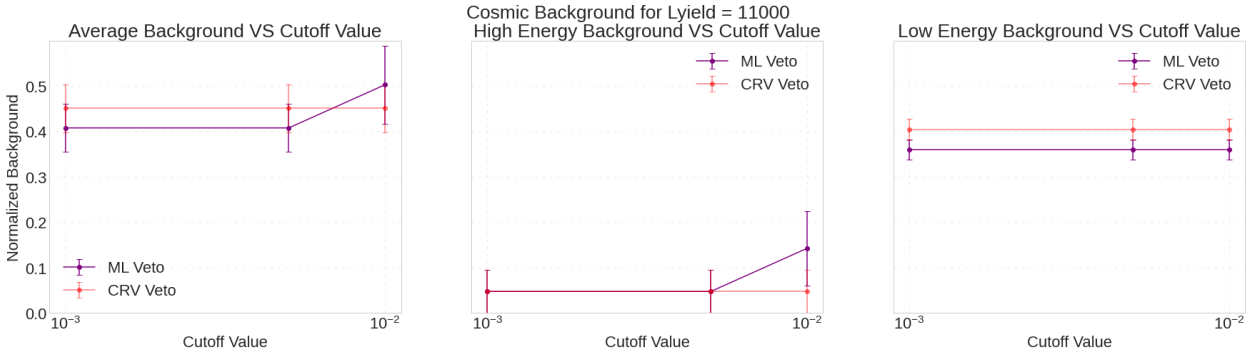


Figure 44: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 11000.

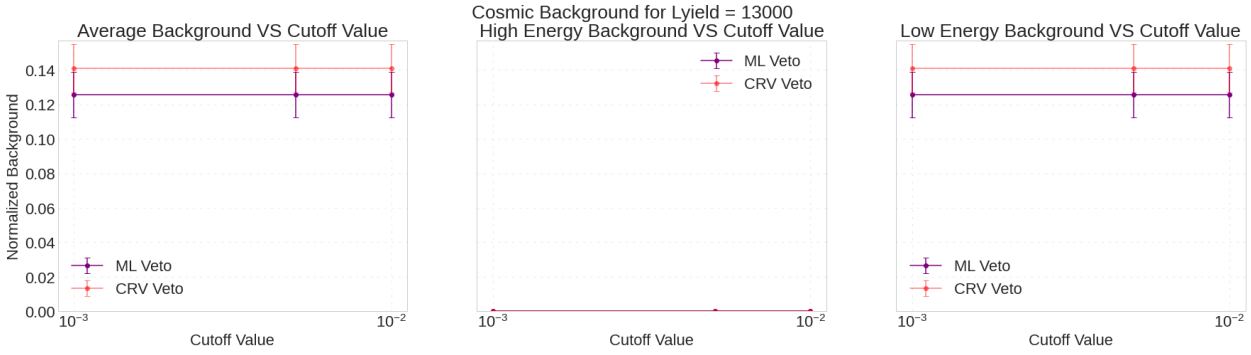


Figure 45: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 13000.

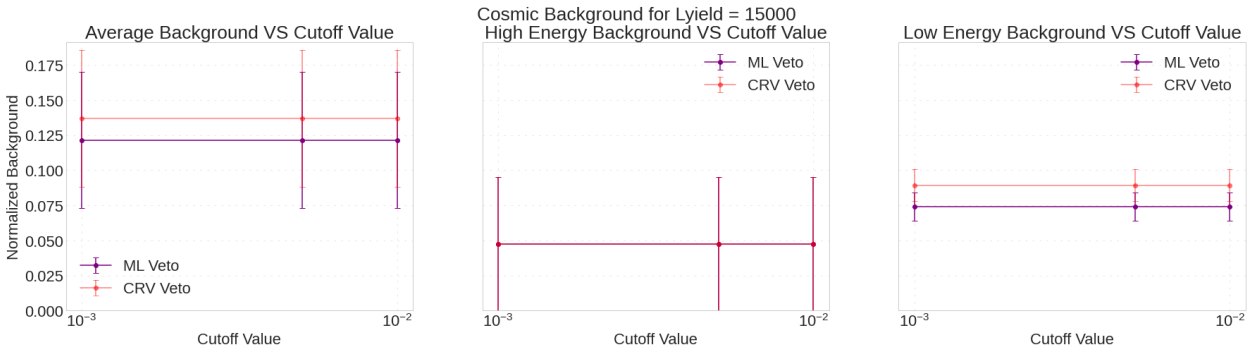


Figure 46: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 15000.

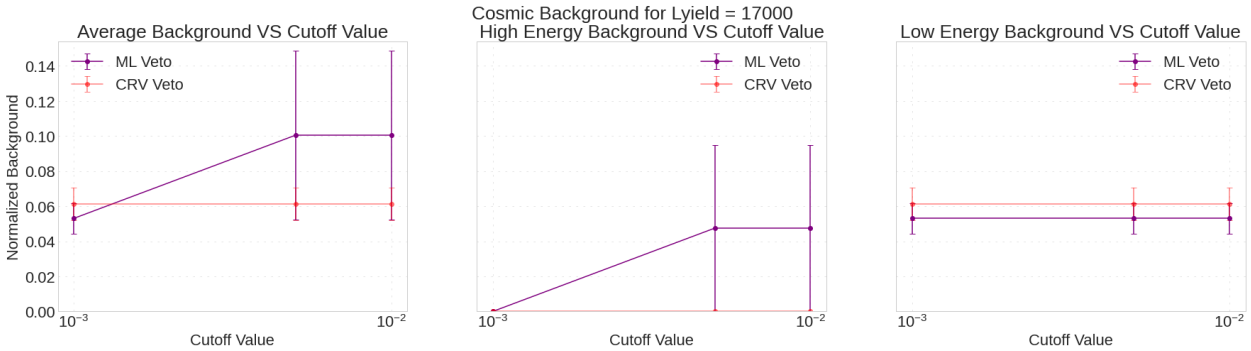


Figure 47: In-depth breakdowns of the cosmic background using different cutoffs, as well as the CRV Time Window Veto, for elements of the CRY4 sample with a light yield of 17000.

C The Code for Various Cuts

The following section contains the code forms of the cuts described in Section 5.2. Whenever a cut builds off of a previous one, there is an addition sign and the name of that set of cuts. Variable definitions can be found in Table 2, however some variables are not in Table 2 and are defined in this section.

Loose Box Cuts

```
    cut_lbox = [
    "(deent_td > 0.577350)",
    "(deent_td < 1.000)",
    "(deent_d0 > -80)",
    "(deent_d0 < 205)",
    "((deent_d0 + 2./deent_om) > 450)",
    "(is_triggered)"
]
```

The variable "is_triggered" checks for whether an event was triggered at all.

Loose Cuts

```
    cut_loose = [
    "(dequal_TrkQual > 0.8)",
    "(dequal_TrkPID > 0.95)",
    "(ue_status <= 0)"
] + cut_lbox
```

The variable "ue_status" checks whether the event is upstream, with a value less or equal to 0 indicating it is not, and a value greater than 0 indicating that it is. The Loose Cuts build off of the Loose Box Cuts, as evidenced by the addition in the code.

Box Cuts

```
    cut_box = [
    "(de_status > 0)",
    "(deent_td > 0.577350)",
    "(deent_td < 1.000)",
    "(deent_d0 > -80)",
    "(deent_d0 < 105)",
    "((deent_d0 + 2./deent_om) > 450)",
    "((deent_d0 + 2./deent_om) < 680)",
    "(is_triggered)"
]
```

Quality Cuts

```
    cut_qual = [
    "(dequal_TrkQual > 0.8)",
    "(dequal_TrkPID > 0.95)",
    "(ue_status <= 0)"
] + cut_box
```

Kinematical Cuts - Extended Momentum Cut

```
    cut_extmom = [
    "(deent_mom > 100)",
    "(deent_mom < 115)"
] + cut_qual
```

Kinematical Cuts - Physical Momentum Cut

```
    cut_phymom = [
    "(deent_mom > 103.85)",
    "(deent_mom < 105.1)"
] + cut_qual
```

CRV Time Window Cut

```
crv_time = """( ((-de_t0 + crvinfo__timeWindowStart) > 50)
OR ((de_t0 - crvinfo__timeWindowStart) > 150.0))"""
```

The CRV Time Window Cut is checking for whether $\Delta T = \text{Time}_{\text{CRV}} - \text{Time}_{\text{Tracker}}$ is both less than 50 and greater than -150 ns. Time_{CRV} is represented by "crvinfo__timeWindowStart", and $\text{Time}_{\text{Tracker}}$ is represented by "de_t0."