Machine Learning for Neutrino Physics at DUNE – Final Report

Student: u2201924

Department of Physics, University of Warwick, Coventry CV4 7AL, United Kingdom

13/03/25

Physicists at the Deep Underground Neutrino Experiment are constructing liquid argon time projection chambers to image, measure and classify neutrino interactions. This project aims to employ various machine learning approaches to contribute in classification methods of neutrino interactions. These include a first principles likelihood and an extreme gradient boosting decision tree. This ultimately provides an avenue of understanding matter and antimatter neutrino oscillations, how the two differ and the resultant evidence for CP violation. The project pair developed a first principles likelihood approach of track/shower separation on Cheated reconstructions before converging on a boosted decision tree that classifies a given interaction with a true positive accuracy of $78.1\% \pm 0.45\%$.

I. Motivation

A particular aim of the physicists at DUNE (Deep Underground Neutrino Experiment) is understanding why our universe was created with more matter than antimatter after the big bang, known as the matter/antimatter asymmetry [1]. Studying neutrino interactions provides critical insights into this question.

For the purposes of this project, neutrino interactions can be divided into three types: $CC\nu_e$, $CC\nu_\mu$ and $NC\nu_x$. Charged current (CC) interactions will release a characteristic lepton (e, μ) once the incident neutrino hits an argon nucleus in the detector, the lepton will indicate the neutrino flavour, thus the interaction is a charged current electron or muon neutrino interaction, $CC\nu_e$ or $CC\nu_\mu$. If no lepton is detected, then imaging detectors will not be able to distinguish the flavour and hence

the neutral current interactions are labeled $NC\nu_x$.

In this case, measuring differing flavour oscillation behaviour between matter and antimatter neutrinos $(\nu, \bar{\nu})$ gives evidence for a break in charge conjugation and parity symmetry, equivalently known as CP violation. Convincingly showing proof that CP symmetry has been violated allows clearer discussions of matter/antimatter asymmetry, ultimately leading to the beginning of an explanation of leptogenesis [2].

Specifically, the end goal for this project is to implement machine learning methods that accurately separates neutrino interactions into the three categories, $CC\nu_e$, $CC\nu_\mu$ and $NC\nu_x$. Accurate classification means a high chance of correctly classifying the flavour of the incident neutrino. As DUNE starts, the detector will be running for an extensive amount of time. Annotating interactions manually is neither as accurate as machine learning models, nor as time efficient. There is a resounding demand on machine learning implementations to detect neutrino flavour, and this project aims to outline methods that aid this.

II. Theory

Building on this motivation, it is vital to understand and measure neutrino oscillations to detect CP violation. Additionally, an understanding of the neutrino beam composition and the principles behind detector operation will help contextualise the project measurements. This section also covers key theoretical concepts, including the various topologies to be studied, their unique physical characteristics, and their significance.

A. ν_e and $\bar{\nu}_e$ Appearance

In principle, DUNE will fire a beam of neutrinos, ideally composed of solely ν_μ or $\bar{\nu}_\mu$, depending on the study DUNE will carry out. The real composition of the beam can be predicted using the beam composition of the MicroBooNE detector. So for a ν_μ experiment, DUNE will work with a beam will likely contain 92.9% ν_μ, 5.8% $\bar{\nu}_\nu$ and 1.3% $\nu_e/\bar{\nu}_e$ [3].

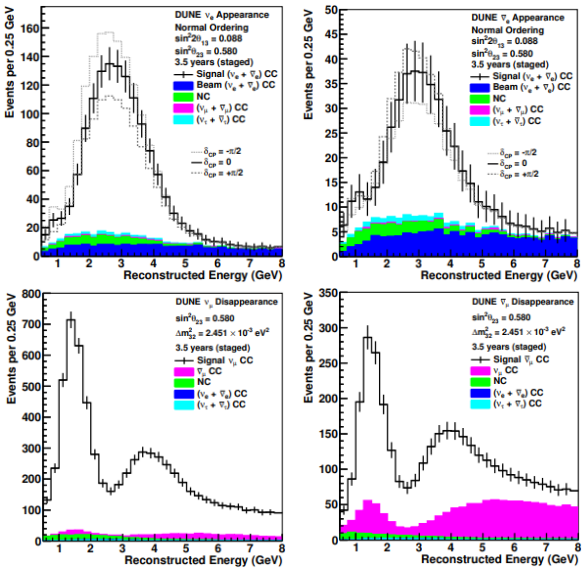

Figure 1: Simulated appearance and disappearance spectra for matter and antimatter ν_e and ν_μ, adapted from Abi, B et al [4].

To look at matter and antimatter neutrino oscillation characteristics, DUNE will

measure a phenomenon called 'ν_μ disappearance' (Fig. 1). In the lower two subplots, it is clear that the $\nu_\mu/\bar{\nu}_\mu$ spectra experience a loss of event counts in the reconstructed energy bins between values of 1-4 GeV. This is due to ν_μ 'disappearing', or more accurately, oscillating into ν_e. This can be seen in the 'ν_e appearance' plots above, where ν_e are detected in the corresponding reco energy bins.

Fig. 1 is a simulated plot, and shows the bounds for discovering CP violation through the CPV phase parameter δ_{CP}. The main idea is detecting ν_e appearance that places δ_{CP} somewhere in the continuum of $\pm\frac{\pi}{2}$, observing appearance spectra near the maximal values provides the strongest evidence for CP violation.

An important note on the plots in Fig. 1 is that the beam composition is impure, and will have intrinsic (1.3%) electron flavour. The figures account for this and display the "Beam ($\nu_e + \bar{\nu_e}$) CC" energy.

B. Charged and Neutral Current Interactions

Section I referred to neutrino interactions with respect to the charged lepton; in this case electrons or muons. $CC\nu_e$ events will have a primary electron nestled at the 'neutrino vertex' where the ν_e strikes the argon nucleus, and $CC\nu_\mu$ events a muon under the same criteria. These are the direct daughter particles of the neutrino. These e/μ are referred to as 'candidate leptons' throughout the project report; detecting these would indicate the flavour of neutrino that strikes the argon nucleus, and thus, classify an event into $CC\nu_e/CC\nu_\mu$.

C. Neutrino Oscillations

The three known neutrino flavours correspond to the lepton generations (e, μ, τ). A neutrino in transit has a mass described by a superposition of mass eigenstates, commonly noted ν_k (k = 1, 2, 3). Upon interaction, neutrinos will assume a distinct flavour. These distinct flavour masses are labeled by their lepton: ν_e, ν_μ, ν_τ. The probability of a neutrino oscillation between flavours, $P(\nu_\alpha \rightarrow \nu_\beta)$ is a function of various mixing matrices and is a concept beyond the scope of this project.

$$P(\nu_\alpha \rightarrow \nu_\beta) = \sin^2(2\theta)\sin^2(\frac{\Delta m^2 L}{4E_\nu}) \quad (1)$$

Equation 1: *The two-flavour oscillation transition probability, where θ is the mixing angle (evaluated experimentally); $\Delta m^2 = m_2^2 - m_1^2$ is the mass-squared difference; L is the distance traveled; E_ν is the neutrino energy.*

A 'two flavour approximation' (Eq. 1) for this project will be assumed hereforth, which suffices due to kinematic restrictions on tau neutrinos. $\nu_\mu \rightarrow \nu_\tau$ oscillations are

more common with a ν_μ beam, but the ν_τ will likely not meet the threshold energy to create a charged current interaction - around 3.5 GeV is required to create a stationary τ in the first place. Not only are $CC\nu_\tau$ interactions not at all common, detecting them is also a significant challenge and is beyond the scope of this project. DUNE are also not currently actively researching $CC\nu_\tau$ detection, but may look into it in the future. Tau leptons, in the case of a $CC\nu_\tau$, will decay very with a mean lifetime of $\approx 2.9 \times 10^{-13}$ s. These decays can be leptonic, $\tau \to e/\mu$, which look like events that are already discussed regardless. If the tau does not decay leptonically, then they could decay hardonically into a range of particles, mainly pions. This topology will not appear any different in the detector. DUNE aims to detect ν_τ in the future, but it is not a short term priority.

D. Neutrino Beam and the Detectors

Fermilab in Illinois, USA, is a facility located about 1300 km from where the DUNE far detector is being constructed in South Dakota [1] (Fig. 2). Neutrino beams are created at Fermilab by firing protons from their proton accelerator at a target material. This target material will produce many charged $\pi^\pm$, $K^\pm$ that can be steered into a beam and subsequently decay, predominantly via the weak interaction, into neutrinos. The magnetic horn can change the field direction to toggle the ν_μ beam or $\bar{\nu}_\mu$ beam. Positive pions and kaons will decay into matter neutrinos, whereas negative pions and kaons will provide the antimatter neutrino beam.

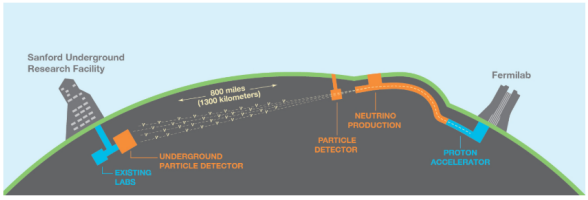

Figure 2: A diagram of DUNE, on the macro scale. Adapted from dunescience.org [5].

The Neutrino beam that will be studied at DUNE is, ideally, a muon beam with a composition estimated in section II.A. Oscillations are expected, with $L = 1300$ km (Eq. 1), with a two flavour $P(\nu_\mu \to \nu_e)$ seen in ν_e appearance plots.

Since neutrinos interact so weakly, the beam is simply fired through the Earth's crust towards the far detector. Each of the detectors are liquid argon time projection chambers (LArTPC). A near detector, depicted as "particle detector" in figure 2, is placed near the start of the neutrino beam to inspect the beam composition. Due to the high neutrino flux however, imaging will be cluttered so the near detector will not be used for examining neutrino interactions. The beam composition is necessary to fully analyse neutrino oscillations, and

incidentally present evidence for CP violation in the long run.

The far detector, depicted as "underground particle detector" in figure 2, can get a clearer picture of an interaction, since after traveling 1300 km the neutrino beam has a much lower intensity, the beam radius is now in the order of kilometers. Due to the drop in beam intensity, the interactions imaged from the detector are less cluttered and will only show the particles from a single neutrino interaction.

E. LArTPC Operating Principles

The detector works by using highly pure liquid argon (LAr) to extract ionisation electrons released by charged particles within the chamber. These electrons, once emitted, are subjected to a uniform electric field and translated by a known constant drift velocity. The field will drag the electrons through 3 sets of tightly packed parallel wires, typically called U, V and W wires (Fig. 3).

Each wire will detect electrons as a digital signal. The distance from the wires can be calculated as a function of translation time and the drift velocity.

Pure LAr provides a clean, inert backdrop to drag the ionised electrons through. Charged particles from the interaction create a free $e^- + Ar^+$ pair. Since LAr is stable and pure, the free electrons are not affected in their travel to the anode [6]; the shape of the event can be maintained over a large drift length to a high degree of accuracy.

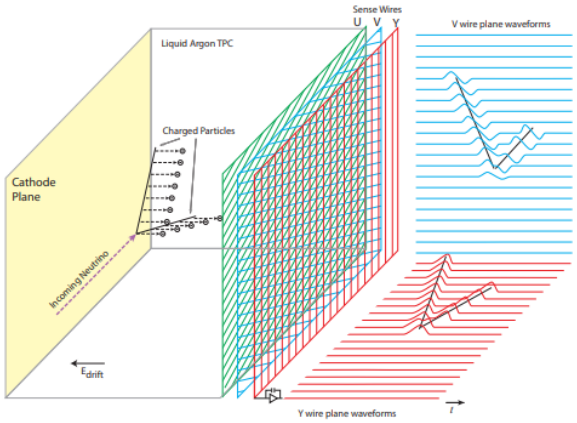

Figure 3: A LArTPC diagram, adapted from Acciarri, R. [3].

For this project, the set of W wires will be the sole focus, represented in figure 3 as the Y wire plane. This is the convenient choice since they are oriented in a way that the w wires give z co-ordinates. Additionally, the xw plane provides the highest quality set of 2D images, known as the 'collection plane'. The U and V wires do not capture the ionised electrons, and hence provide a bipolar pulse as a signal when the electrons pass across the mesh (V wire plane, Fig. 3). Since the W wires provide a single Gaussian pulse, this plane has the most favourable signal to noise ratio. The resultant projection of the event onto the xw plane called the 'W view', and is where all of the following analysis will be held.

The wires do not only provide spatial information, but reconstructed energy ADC

values too. An event with a higher energy will ionise more electrons, a W wire will detect a larger pulse which is then integrated and given as the ADC, a quantity directly proportional to the deposited energy of the neutrino event. Calibrators and corrections give a scalar that converts between ADC and energy (Eq. 2),

$$1 \text{ ADC} = 0.0075 \text{ MeV}. \tag{2}$$

When presenting energy information, that is the transformation applied to the ADC values in the events files.

F. Interaction Topologies

Before introducing the computational methodology underlying the project, it is essential to first introduce the interactions DUNE will be looking at with more detail, and how each topology provides its own differing characteristics.

Section II.B introduced candidate leptons for each charged current interaction, and why detecting a candidate lepton is an important area of study. In summary, charged current ($CC\nu_{e/\mu}$) interactions display a primary e/μ within the event, neutral current ($NC\nu_x$) interactions do not have a candidate to indicate neutrino flavour. Electrons and muons themselves have differing characteristics that are detected in the LArTPC.

Electrons undergo a series of interactions called an electromagnetic shower, a cascade of particles primarily driven by Bremsstrahlung radiation and pair production [7]. The general process is as follows: High energy electrons or positrons will release a Bremsstrahlung photon accompanied by a lower energy electron or positron respectively: $e^{\pm} \rightarrow e^{\pm} + \gamma$. These photons will then interact with a nearby nucleus, pair producing an electron positron pair: $\gamma \rightarrow e^{+} + e^{-}$. This process will continue at an exponential rate until the electrons, positrons and photons do not meet energy thresholds to fuel the subsequent decays, and instead either ionise nearby atoms instead of pair producing, or electrons and positrons annihilate into lower energy photons that are not detected.

Muons are about 200 times more massive than electrons, and do not decay quickly or interact with the strong force. As a result, they have a much higher penetration power and will retain their momentum for a greater distance through LAr. Muons are also unlikely to decay or emit Bremsstrahlung photons. These characteristics give muons a signature track in a straight, steady line in an event display.

Tracks are frequently imperfect and do not always show a perfect, rigid line. Delta rays can be folded into the track reconstruction, these are small electromagnetic showers when a muon has knocked an electron from an argon atom in passing. Michel

electrons are typically seen at the end of a muon track, if it has not escaped the detector. These 'Michel electrons' are small showers after a muon decays into an electron.

III. Methodology

This section will outline how the data is provided, what can be extracted from it, the machine learning approaches employed, methods used for performance evaluation, and the challenges encountered during the process.

A. Computational Background

DUNE is still under construction, so experimental data does not exist yet. This is why there are two reconstructions provided by DUNE that simulate the event data that is expected from the LArTPC, named 'Pandora' and 'Cheated'. Both file types are based on a series of simulations that recreate what is expected from real neutrino events. Distinctively, Cheated files are a true reverse engineering of those simulations providing a perfect reconstruction of the events, whereas Pandora files are reconstructed using the Pandora pattern recognition, consisting of over 70 algorithms that DUNE will use to reconstruct real LArTPC readings [8].

There exist two large data processing challenges at DUNE, one being the event classification, introduced in section I, the other being accuracy of reconstructions. The project pair decided to opt into exclusively using Cheated reconstructions. This avoids conflating the barriers of classification with impure or incomplete reconstructions in order to provide insight into the potential ceiling of event classifications.

"CheatedRecoFile_n.root" (n = 0, 1, ... 10) files are handled by the python package "uproot_io.py" [9], a documentation module providing easy access to event data. Data for each particle within the files are accessed with a unique particle index, confusingly called an "event index" at times. An important distinction to make early on is the difference between an event index, unique to each particle, and an "event number" that allows the grouping of multiple particles under one interaction event. For example, you may have 12 particles, each with unique index [5, 6, 7, ... 16], but they may have the same "self.event_number[index]" in the root files. This means that all 12 of those particles are part of the same event. This distinction is useful to keep in mind for later sections when making predictions from machine learning models on a particle or event basis.

B. Event Display

Figure 4 gives a clear example of what an event in the collection plane from the Cheated files will look like. Since it is a NC event, a primary candidate lepton evidently cannot be seen.

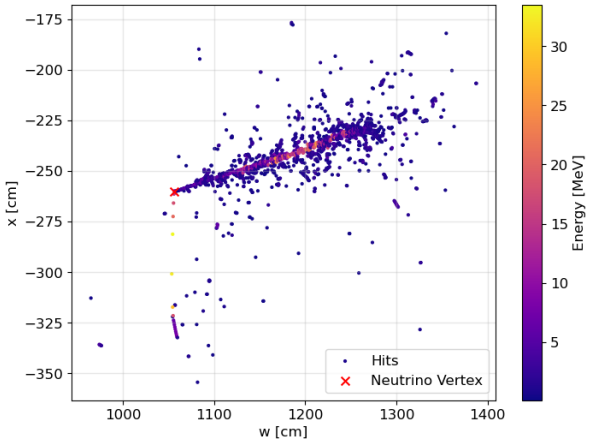

Figure 4: A CheatedReco $NC\nu_x$ event, showing the hits (w, x) and a colourmap of the energy deposition in MeV (Eq. 2).

For clarity, a 'hit' is the position of ionisation electrons that the LArTPC has imaged; the term 'ADC' refers to the quantity proportional to the energy of the hit in an event (Fig. 4), calculated by integrating over the pulse given by the wires. The reconstructions will also infer neutrino and particle vertices, which are given with perfect accuracy in Cheated reconstructions. The neutrino vertex is where the incident neutrino struck the argon atom, and each particle vertex is where a child particle was born. A candidate is a child of the neutrino, Michel electrons are child particles of a muon for example. Due to the topology of DUNE and the LArTPCs, the neutrino beam is incident in the w_+ direction. The xw plane is in effect a birdseye view of the event.

C. Defining Separation

Studies in this project are frequently motivated by the necessity to find measurable characteristics that differentiate particles within the event display. These measurable, numerically quantifiable characteristics are to be called 'features'.

An important separation axis that the project pair study is 'track-like' and 'shower-like' particles. The imperative behind track/shower separation is simply the fact that the fundamental difference between $CC\nu_e$ and $CC\nu_\mu$ events lies in the detection of a primary candidate lepton.

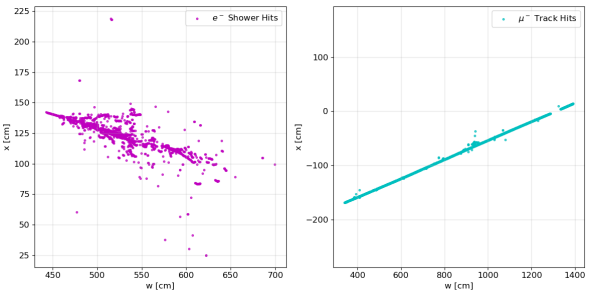

Figure 5: Figures demonstrating a characteristic e^- shower (left) and a μ^- track (right).

The mechanisms of an electromagnetic shower and a signature track explained in section II.F can be seen in figure 5. The electron shower shows a chaotic cascade of Bremsstrahlung radiation and pair production near the first half, and a gradual halt of the decays as the energy per particle dwin-

dles, while the muon track retains a steady, inert path. An example feature to separate tracks and showers would be the absolute root mean squared error of a linear regression model. Upon inspection, a shower would have a significantly higher value than a track, which would predictably maintain a near zero RMSE. The probability density functions (PDFs) of the absolute RMSE for true tracks and showers (Fig. 6) demonstrate a feature that separates tracks and showers numerically.

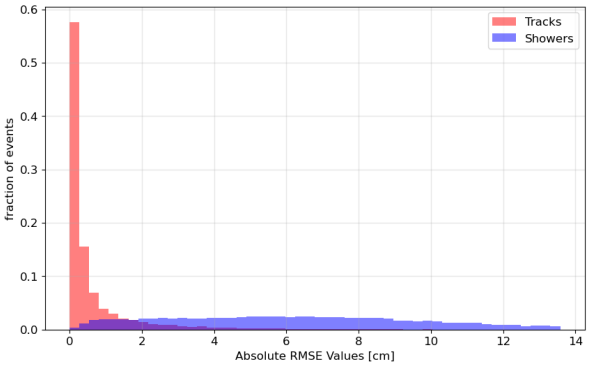

Figure 6: Two PDFs of track/shower RMSE values, right hand side is cut at the 95th percentile.

The second separation study develops the classification of electromagnetic showers further, into electron showers and photon showers. Quantifying e-like and γ-like showers is significantly more challenging than tracks and showers, where only two major distinguishing features can be created. Photon showers, contrary to electron showers, have what the project is calling a 'step' between the neutrino vertex and the start of the shower. This is a short distance, typically a few centimeters, between the neutrino vertex and the detected hit with the shortest euclidean distance to the vertex. The rate of energy deposition with respect to distance over the start of electron showers and photon showers are measured to be different, photon showers expel $\sim$ 4.2 MeV/cm whereas electron showers expel $\sim$ 2.1 MeV/cm.

D. The Workflow

In the first part, the project explored an introduction to a first principles machine learning Likelihood approach to distinguish track-like events and shower-like events. Cheated files provide enough reconstructed information to be able to tell tracks and showers apart from the events class in the uproot file, using the Monte Carlo PDG convention [10], a universal numbering scheme that attributes an integer to each particle, with antiparticles having a negative PDG code. Showers are composed of γ, e^-, e^+, respective PDG codes: 22, 11, -11, this is how the truth mask for showers was created.

Secondly, the project pair develop two features for separating electron and photon showers. For truth values, showers were separated further via their respective PDG codes, $e = \pm 11$, $\gamma = 22$. For completeness, tracks are separated further into muon tracks and non-muon tracks using the same features for track/shower separation.

Additional features are not developed since the track/shower features already separate μ and $!\mu$ tracks highly effectively. As before, $\mu/!\mu$ separation was handled via PDG codes, $\mu = \pm 13$, whereas $!\mu$ is a blanket term for all other tracks, commonly protons and charged pions.

The next step is to implement features into a boosted decision tree. Unlike the likelihood, the boosted decision tree will not be built from the ground up, but imported through a python module XGBoost.

Choosing training data and sample data is simple, the criteria are: a) a sufficiently large training sample size to form complete PDFs that span the typical feature values, doubly ensuring that a wide range of values for each feature are covered; and b) the training and sample data are disjoint. The second is important since it is possible to overtrain machine learning models, having the model look at event files that it has seen before is not productive and does not provide any new information.

For the likelihood, CheatedRecoFile_0.root is the training data whereas CheatedRecoFile_5.root will serve as the sample/unseen data. During the training and testing phases of the BDT, a systematic error was found in the event truths data in the Cheated files, causing some NC events to be classified as CC. Due to this, new "CheatedRecoFile_0_new.root" and "CheatedRecoFile_1_new.root" were generated, and are the training/testing data respectively for the BDT. The likelihood will not be affected by this error since the truth data was collected via MC PDG codes, not boolean 'is_nue' arrays for example.

To affirm that the selected datasets fulfil the criteria given above, it is important to note that all files are statistically similar, but entirely independently generated by the simulation. There are no repeat events in any of the files, hence they are disjoint. Each file contains $\sim 500,000$ particles across $\sim 9,500$ events. After quality cuts and anomalous particles that return excused errors or NoneTypes, the training and testing data will both contain $\sim 49,500$ well-defined feature vectors for each model to train and test. Taking an approximate 90% hit to the training data seems significant, but a large majority of particles in the Cheated files are events containing 1 or 2 photon hits, which are not giving much information on the event class. Looking at the larger particles and their nature is the effective convention to train models on.

E. Python Packages

A short list of the python packages used, and a brief description of the utility provided by each.

- **uproot, Awkward.** uproot [9] and uproot_io.py used to extract informa-

tion from the reco files. Awkward for array documentation within the uproot tree [11].

- **NumPy.** For fast array calculations and formatting, useful numerical functions (np.pi, np.sqrt, np.trapz etc.), additionally helpful to save large arrays [12].

- **Matplotlib, seaborn, Pandas.** For data analysis and presentation. The two former are helpful tools for presenting and plotting data [13], the latter is used for dataframes, specifically the correlation attribute [14].

- **scikit-learn (sklearn).** DBSCAN, accuracy_score, classification_report, confusion_matrix. For use in features and calculating performance figures [15]. RandomisedSearchCV to optimise hyperparameters for a boosted decision tree.

- **SciPy.** ConvexHull, stats. Former used in a booster feature in a future section, latter for helpful statistics calculations [16].

- **XGBoost.** The boosted decision tree package, trains and tests data. Findamentally takes an input of features and truth labels [17].

- **Joblib.** To store the boosted decision tree once it has been trained, saves time taken recalculating training data and training the bdt [18].

F. Features

Features are a qualitative measure of a particle or event that provide empirical separation between classes of events, per the principles discussed in section III.B One feature may, for example, have high values for shower-like particles and near zero values for tracks, that would mean the feature is strongly separating the tracks and showers. When incorporating multiple features into a model, a higher degree of performance is achieved when features are individually strong and show as low a correlation as possible. Orthogonal features will provide new angles of information to the classification that highly correlated features will not, resulting in a more informed, subsequently accurate, overall result.

G. The Likelihood

To briefly recap independent probability, given two independent events A and B, $P(A \cap B) = P(A) \times P(B)$ (events not in the context of neutrino physics for this case). The likelihood model works similarly. For this project, the model was not a black box python package, such as GaussianNB from sklearn.naive_bayes for example, and hence provides full exposure to the machine learning process. Only track/shower separation

was studied by use of a likelihood model.

Once PDFs for the 'track vs shower' features have been created on the training data (Cheated_0), the likelihood will, for each particle in the testing data (Cheated_5), calculate the feature values. For each value, PDFs are read and they return probabilities of a track and shower for each ith feature, $p_{track,i}$ and $p_{shower,i}$. Finally, the probability of that particle being a track P_{track} and shower P_{shower} is calculated (Eq. 3).

$$P = \prod_{i=1}^{N} p_i \quad (3)$$

$$\mathcal{L}_{track/shower} = \frac{P_{track/shower}}{P_{track} + P_{shower}} \quad (4)$$

Now, the likelihood can be calculated (Eq. 4). After the implementation of many features, almost all events in the sample data should return $\mathcal{L}$ values close to 0 or 1, a true shower should have $\mathcal{L}_{shower} \approx 1$ and $\mathcal{L}_{track} \approx 0$, and vice versa. For use of this model, a sample particle is said to be track like if $\mathcal{L}_{track} > \mathcal{L}_{shower}$, and vice versa.

H. Boosted Decision Trees

Using a BDT is a step up from the previous first principles likelihood approach exploring track/shower separation, in the sense that the decision tree will be an imported python module (XGBoost).

H..1 Key Details

BDTs take features, just as in the likelihood, to gain understanding on testing data and classify on training data. A large upside to using a BDT in comparison to the likelihood is it will decide how much each feature should weigh in on the final classification, potentially deciding to use features more than once during its decision routes. The one caveat to this lies in the fact that the XGB package makes its own decisions with little feedback to the user. This makes troubleshooting difficult if unexpected results are obtained.

H..2 Booster Features

Since boosted decision trees make decisions accounting for how performant individual features are, they benefit from an abundant number of features that provide new insight. In other words, even if there are many features, as long as only a small subset of the features is uncorrelated and provides a new angle on the training data, a BDT will only benefit from that subset, and there are diminishing returns on both performance increase and a risk of overfitting data (Section B4 [19] implies).

H..3 Producing a Model

The model works fundamentally using 4 arrays. x_train, y_train, x_test, y_test. The x arrays are $n \times m$ feature matrices F_{ij} where

the ith row corresponds to a feature vector, composed of m features over n samples. Whereas the y arrays are $n \times 1$ arrays of truth values for event classification, the convention is: $CC\nu_\mu = 2$, $CC\nu_e = 1$ and $NC\nu_x = 0$. Truth values are obtained through the Events class in the uproot files, where each particle has boolean values for "self.is_numu" and "self.is_nue" for each respective interaction.

The XGBoost model is trained on x_train and y_train, test classifications are made on x_test, before the model is evaluated on the truth values y_test. Predictions are made on a per-particle basis on whether each particle in the files is an electron, muon, photon or non-muon track. After that, some logic is applied to every event to find the primary particle in each event, look at whether the BDT has classified this particle as an electron or a muon. If it has, then the event class is either of $CC\nu_e$ or $CC\nu_\mu$. If not, then the event is classed as $NC\nu_x$.

Boosted decision trees can be hyperparemetrised, which is just a method of setting and optimising configuration variables. The main reason to tune hyperparameters is that it improves model performance across the board, if you are going to be performing an XGBoost, a vast majority of the literature suggests there is little sacrifice for a lot of performance gain [20]. RandomizedSearchCV is a time effective method of trying and testing parameters over a small parameter grid to squeeze out some extra performance. It will randomly search for the best configuration to optimise learning behaviour, without learning from the training data itself. This is vital since over-training models will lead to inconsistent results. In general, a parameter grid is given to RandomizedSearchCV, which will go through them and find a good balance between underfitting and overfitting these variables, an example of which is 'n_estimators', which is just the number of trees the BDT creates. More trees result in a more respectable performance, but there is a greater risk of overfitting. On the contrary, too few estimators will lead to an underwhelming performance since there is not a high enough capacity to capture the nuance that the features provide, and the data will be underfitted. Most hyperparameters follow the same balancing act, trading too large an increase in performance for a risk of overtraining a model.

H..4 BDT Structure

A final note on the structure of the BDT. Initially, the plan was to separate particles into tracks and showers, and subsequently separate those tracks into μ and $!\mu$, and showers into e and γ. The workflow makes perfect sense, but there is a pressing issue. The track/shower separation in the initial split is imperfect, so the two second stage BDTs that split tracks and showers further

are being trained on impure data - a shower BDT being trained on a small percentage of muon tracks for example. Bootstrapping the three BDTs together permanently reduces the accuracy ceiling at every step, and opens up more potential room for error. Decisions such as whether to train secondary models on pure or impure data are needed for example. This is not representative of what a machine learning model should be dealing with, so the decision to switch to a more complete model where the BDT only has a single stage is the optimal one going forward.

I. Evaluating Performance

Gaining an overall accuracy at the very end of the models is a concise and efficient way of summarising performance, but there is a much larger story to tell. Evaluating the effectiveness of the machine learning models shall be divided primarily into two sections, input quality and output performance.

Feature correlation for every feature can be plotted to visualise input quality. More heavily correlated features add less depth to the ability to make a classification for both the likelihood and BDT. An addition of a highly correlated feature in either machine learning approach will add minimal performance, even if the feature is highly performant individually. Pearson correlation coefficient (PCC) is a standard statistical method of calculating linear correlation between two variables [21]. Varies range between -1 and 1. Variables with PCC $= -1$ would indicate a perfectly negatively correlation, PCC $= 1$ are perfectly positively correlated and PCC ≈ 0 indicates highly uncorrelated variables.

The transparent nature of the likelihood allows for a wider exploration of various output quality metrics. Efficiency-Purity (E-P) curves are plotted for all likelihood features, and subsequent area under curve (AUC) values of each feature is a metric that quantifies how well the feature simultaneously provides both efficiency and purity at every selection. AUC is a metric that quantifies how effectively the chosen feature preserves efficiency and purity simultaneously across the PDF. The AUC value is obtained by integrating the E-P curve. A perfect classifier has an AUC value of 1.0, a random (uncorrelated purity and efficiency) curve has an AUC of 0.5. A poor classifier is counterproductive and has values between 0.0 and 0.5.

The input quality of the BDT can be directly extracted from the model. This will consist of a bar graph of the 'feature importances', which is simply a normalised metric of how heavily each feature contribution weighs across the span of the decision tree

for each prediction made on y_test.

$$\epsilon = \frac{k}{n} \quad (5)$$

$$\sigma = \sqrt{\frac{\epsilon(1-\epsilon)}{n}} \quad (6)$$

Output quality of the BDT shall be assessed as a confusion matrix of the predictions made by the model, and the truth values. Efficiencies (Eq. 5) and uncertainties (Eq. 6) of each classifications will be calculated assuming binomial statistics, given that k events were correctly assigned out of a total n events per truth category.

There are two distinct matrices that are important here. Firstly the direct BDT output, that is the particle classification model and will be a 4x4 confusion matrix. The second is the final class confusion matrix after the CL logic has made the decisions on which event belongs to any of $CC\nu_e$, $CC\nu_\mu$ or $NC\nu_x$.

J. Edge Cases and Challenges

A challenge experienced in the early stages is instances where the files contains particles containing little to no hits, typically a single photon hit that the Cheated reco created. A short study can be done to see what proportion of individual particles have hits within the 5-10 range, and around half of all particles return less than 5 hits. A few issues arise from this. Firstly, the qualify of the data will drop due to poorly conditioned np.polyfit() functions and inconsistencies with filtering methods. Feature data would be polluted with redundant values since looking at these photons and small hit events are not providing useful input when looking at areas of study previously listed, namely tracks/showers, e/γ separation. Demanding higher hits on occasion is also useful when performing by-hand debugging. If one can tell little about the particle by eye, then there is little troubleshooting that can really be done.

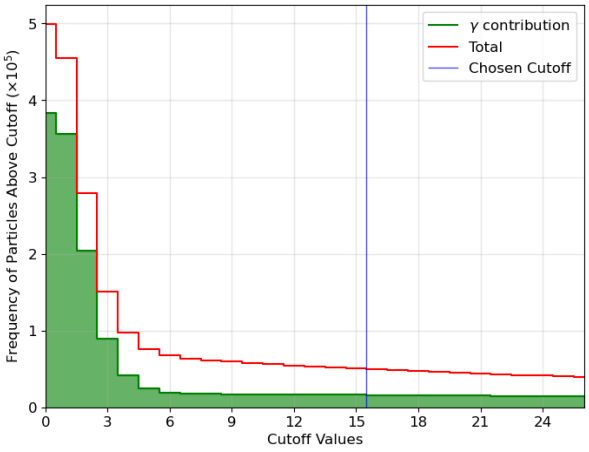

Figure 7: A plot demonstrating the loss of particles as the cutoff point is moved, and the contribution to the particle tally by photons.

It is due to all of this that an arbitrary quality cut of 15 hits was made to all likelihood features, and most BDT features. A feature will completely ignore a particle if it contains 15 or less hits. Figure 7 indicates that beyond around 4-5 hits, as long as the cutoff is within a reasonable range, the loss per increase in cutoff draws extremely thin, which justifies any selection between around 5-20 hits for data collec-

tion.

Another consideration is what happens if a feature returns a value from sample data that lies outside the span of the training data. Choosing p values based on nearby probability bins also potentially resolves this case. If there is a bin with no counts for p_{track} and/or p_{shower} nested inside data, it may be reasonable to take a mean of the two bins either side. At extremities, where p_{track} & $p_{shower} = 0$, p values of the closest bin are used. Fortunately, the training data is large enough such that these cases are few to none, and if there are no counts, taking $p_{track/shower} = 0$ suffices.

Something that poses problems when looking at events is the dimensions of the detector. A handful of events escape the detector before completing their path, thus truncating the information gained from the event.

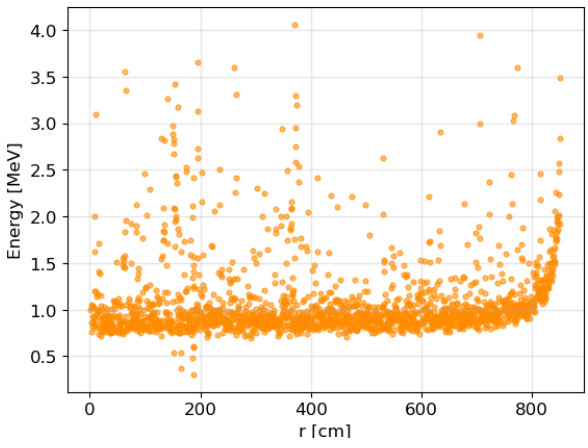

Figure 8: Energy deposition for a track with a Bragg peak, the x-axis shows the distance from the particle vertex.

Tracks will show a characteristic rise in energy towards the end of its trail in the detector, this is called a Bragg peak [22] and can be seen in figure 8.

A non-negligible percentage of tracks escape the detector before it can pick up on the Bragg peak. Experimentally, approximately 8% of tracks above 15 hits have at least some of their hits outside the detector. This is significantly challenging when engineering ADC based features for track/shower separation since there are now 2 distinct types of track to analyse.

Cheated reconstructions, being perfect reverse engineered simulation data, do not include Michel electrons in the muon track. This is why the Bragg peak is the end of the tracks that do not escape the detector.

IV. Results Obtained

Using all of the principles in section III with reference to the physics in section II, the results and performance of the machine learning models will be described and presented. Firstly, a brief overview of all features including the nature and purpose of each feature is given, followed by an assessment of correlation for each feature. Then a discussion of the likelihood performance followed by the boosted decision tree.

A. Features

Six track/shower features are included from the likelihood model, and the final BDT

motivated by e/γ separation spawned two additional features for event classification. Various booster features, for which the concept was explained in section III.H..2, were added to improve performance. This section will include the four most impactful ones that were used in the twelve feature BDT.

For track/shower separation, the likelihood model contributed:

1. **Correlation:** Takes the NumPy np.corrcoef() and np.polyfit() to get correlation and rms error scores respectively. The feature returns $0.7 \times corr + 0.3 \times error$.

2. **DBSCAN Noise:** Uses DBSCAN clustering to calculate clusters and noise points of the particle hits in the xw plane, and simply returns the number of noise points + the number of clusters. Showers will tend to have more noise points and clusters since the hits of a track comparatively has the hits in a solid line, one that DBSCAN recognise typically a single cluster.

3. **RMSE:** A feature that takes the linear line of best fit over a particle and returns the root mean squared error (RMSE) of x hits. Tracks typically have a low RMSE due to their nonvolatile nature, whereas showers are a tree of pair production and bremsstrahlung decays that create a disperse and chaotic event and show higher values.

4. **Angle:** Takes the furthest point from a line of best fit of a particle, and returns the angle of the cone created between the line and the point at the neutrino vertex. This feature is similarly motivated by the disperse nature of showers compared to tracks, where showers will return higher angles compared to tracks.

5. **Line:** Takes a sum of the line integrals between each hit in the xw plane over the total number of hits in the event. Tracks will return lower line integral per hit, whereas electromagnetic showers will return higher values.

6. **ADC Q4 Ratio:** Motivated by escaped Bragg peaks, this feature evaluates the ratio of the sum of the reconstructed ADC values in the final quartile of the particle over the sum of all ADC values of the particle. Showers deposit their energy earlier on, closer to the neutrino vertex, and hence will have lower ADC Q4 values, whereas the characteristic Bragg peak leads tracks to have a higher value.

For e/γ separation, the only two viable features that can be measure from the LArTPC files are:

1. **Step Length:** Taking the Euclidean distance between the neutrino vertex and the start of an electromagnetic shower. Electron showers will show hits straight away, and hence characteristically have a very low step length, whereas in a photon shower, the γ particle typically travels through the LAr, before producing an electromagnetic shower through various scattering mechanisms.

2. **Initial $\frac{dE}{dr}$:** Filter out the start of the shower before the first cluster of dispersion labeled the 'branch vertex', and over these hits, look at the rate of spatial energy deposition from the neutrino vertex. Typical values of $\frac{dE}{dr}$ for photon and electron showers are 4.2MeV and 2.1MeV respectively (section II.B).

Plenty of booster features are tested, but here are the ones worth including in the BDT, based on their contribution to overall classification accuracy:

1. **Hit Count**: The number of hits for a sample particle.

2. **ADC Sum**: The sum of ADC values for a sample particle.

3. **Hull Density:** Find the volume of a convex hull of the sample particle, then return the number of hits for the particle over the volume as the density.

4. **Curvature:** A combination of the sum of absolute angle differences of particle hits per unit length, and the RMS of the angle differences. 70% curvature, 30% angle difference RMS.

5. **ADC Per Hit:** Simply $\frac{\text{ADC Sum}}{\text{Hit Count}}$.

6. **Max ADC Norm:** Normalised position of the highest ADC density across an event over its length.

7. **Scatter Momentum:** Estimated momentum [MeV/c] of the particle, derived from a simplified version of the Highland formula (Eq. 7).

Simplified Highland formula ($\beta \approx 1$) [23]:

$$\theta_{rms} \approx \frac{E_s}{pc}\sqrt{\frac{L}{L_r}} \tag{7}$$

$E_s = 13.6\ MeV$, $L = X_0$ the radiation length in liquid argon, $X_0 \sim 14\ cm$.

Ultimately the most important feature, at least on paper, for the BDT is the **Primary Candidate Lepton (CL)** feature. The end goal for the project is to end up with an event classification BDT. Training and testing the BDT will be on a per-particle basis, this feature completes the classification pipeline by iterating over particles in y_test, grouping them into events via "self.event_number" and making a large decision on which candidate lepton has been found in the event. The primary particle is said to be the one with the parti-

cle vertex closest to the neutrino vertex (in most cases they are the same, but closest ensures the feature is well defined).

The CL classifier has a truth accuracy of $87.7\% \pm 0.34\%$, so if the particle BDT is at its ceiling with 100% accuracy, the overall classification accuracy is bottle-necked by the CL logic, and the workflow will only have an accuracy to match.

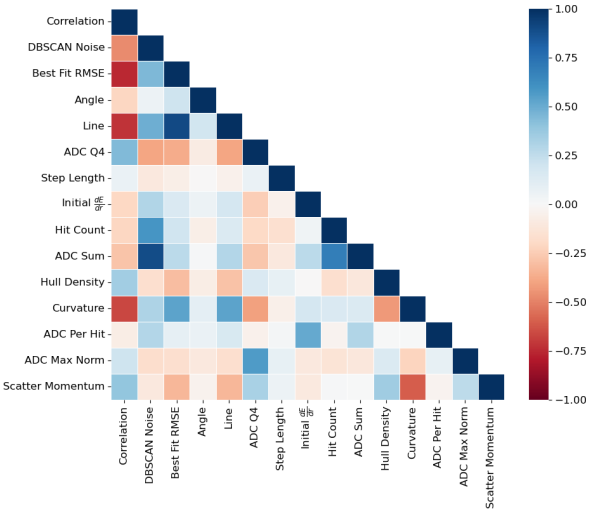

Figure 9: PCC correlation colourmap for all 15 features.

B. Feature Correlation

As it has been mentioned, it is an important criterion to have features in any machine learning model that are as uncorrelated as possible, both likelihoods and BDTs benefit from this. PCC values of the features in section III.A are calculated for all 15 features and plotted in a correlation matrix, seen in figure 9. A deeper blue indicates a strong positive correlation, which is why the leading diagonal corresponds to perfectly correlated variables. A deeper red colour indicates a stronger negative correlation.

Line and RMSE seem to have a high positive correlation. This would indicate that each feature is providing similar information to the training for each model. Intuitively, that would make sense. Both are designed to measure the 'straightness' of a particle in relation to the distance between the hits.

The importance of visualising feature correlation stands out in the next figure. PCC gives values for positive and negative correlation, but the sign of the correlation does not matter when considering input performance for machine learning models. The absolute correlation values of each element are calculated, the mean of each row, excluding the leading diagonal, was found and plotted on a bar graph (Fig. 10)

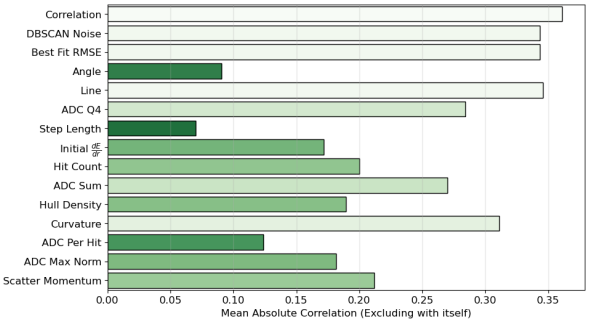

Figure 10: Bar chart of the mean absolute correlation for each feature with every other feature.

This graph provides a baseline for the quality of input features for both the likelihood model and the BDT. From the data, it can be inferred that features such as An-

gle, Step Length, and ADC Per Hit are likely to offer the most distinctive contributions to the models. This is not indicative of performance though. While these features may exhibit lower correlations compared to others, this does not necessarily imply they are the most informative or impactful for the performance of the model. The BDT may place significant weight on a high-performing feature with a large mean correlation, potentially disregarding less correlated features that may still provide valuable information for the decision-making process. One of the many advantages of boosted decision trees, and by extension, hyperparameter tuning, is that these many input and output metrics are evaluated and balanced to provide a model that is not overfitted yet still providing optimal performance.

C. Likelihood Performance Overview

PDFs (eg. Fig. 6) for each feature have been trained on Cheated_0 and stored for testing the model. Likelihood values are computed (Eq. 3, 4) and two overlapping histograms for true tracks and showers are plotted (Fig. 11a).

To get a clearer view of the ambiguity zone, a secondary semi-log plot (Fig. 11b) is created to amplify $\mathcal{L}$ bins with lower relative frequency, mainly those in the neighborhood of $\mathcal{L} \approx 0.5$.

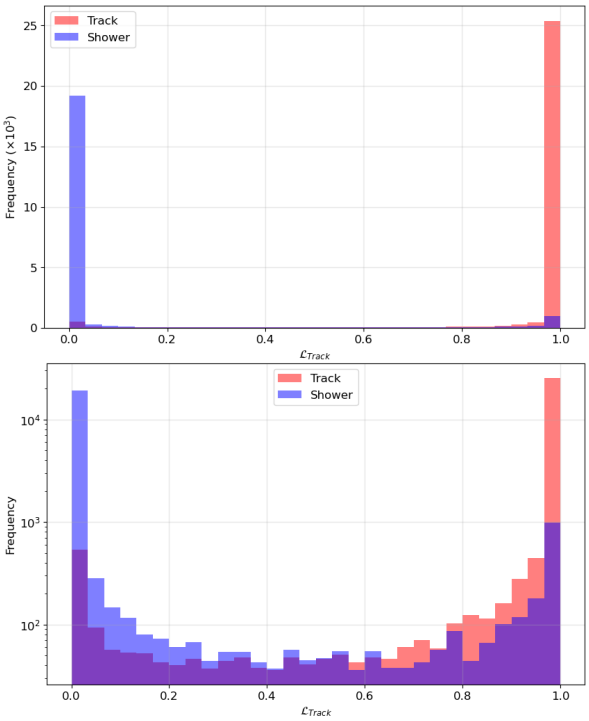

Figure 11: Likelihood plots for classifying a track, classifications peak at extremities where $\mathcal{L}_{track}$ is close to 0 or 1.

Efficiency-purity curves are plotted for each individual track vs shower feature, along with a curve for the overall likelihood (Fig. 12)

The efficiency and purity of each function for each cut is very clear, giving an indication of which features are providing a strong separation of tracks and showers, and which ones are not as helpful.

A bar graph of the area under the P-E curves is plotted (Fig. 13) showing which features best maximised purity and efficiency over the range of feature values. Interestingly, the line feature has identical AUC performance to the likelihood. It is also made clear that the two features that

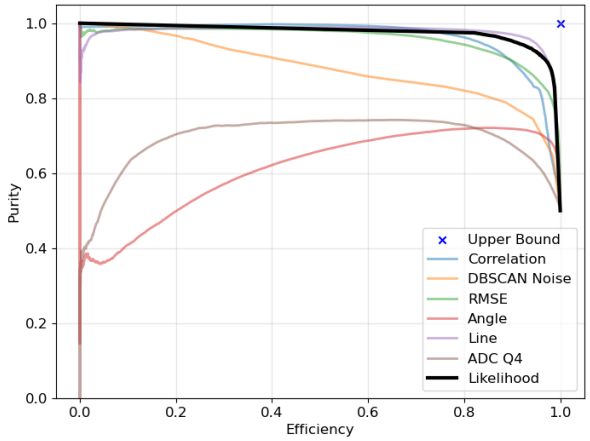

Figure 12: Efficiency-purity curves for track/shower features and the likelihood model.

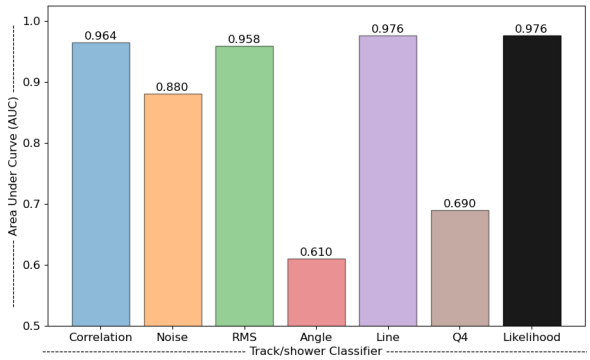

Figure 13: Area Under Curve (AUC) values of the P-E plot (Fig. 9), adjusted for AUC values above 0.5.

provide the least input performance, but still better than a random variable, are Angle and Q4.

The likelihood model achieved an accuracy of $93.77\% \pm 0.11\%$ (Eq. 5, 6) which is considered highly successful.

D. BDT Performance Overview

The BDT was trained on Cheated_0 to classify all particles in Cheated_1 into one of four categories: μ, $!\mu$, e, γ, Following this, the individual particle classifications were filtered by event number. Each individual event was assessed by the CL feature to classify into $CC\nu_e$, $CC\nu_\mu$, $NC\nu_x$.

It is useful to assess model performance in terms of feature importances. These importances can be obtained using the 'feature_importances_' XGBoost attribute, which returns normalized values (as shown in Fig. 14) that indicate the relative contribution of each feature to the decision-making process in the XGBoost model.

The features have been colour coded based on intended purpose. The BDT is making decisions and is not under the impression that any one feature has been engineered to classify any category well or not. There is a small chance that the electron and photon predictions are being made by relying on the booster features almost entirely.

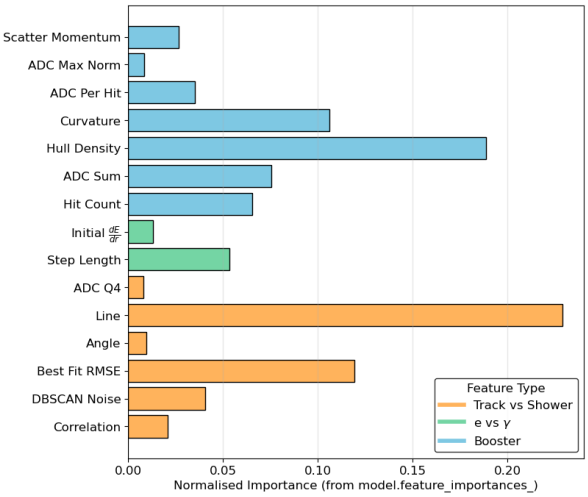

Figure 14: A normalised bar graph showing feature contribution to the particle classification, after randomised search hyperparameters were found.

Unsurprisingly, the BDT favoured the 'Line' feature the most of all track/shower

features, a conclusion reached before when it was shown to be the most performant from the likelihood. It is doubly unsurprising that overall the track/shower features contribute the most in total when you consider that they are the same features that separate muon tracks and non-muon tracks.

With a distinct reference to the nuanced point in section III.H..2 relating diminishing returns on XGBoost performance when selecting correlated features for a BDT, groups of correlated features from figure 10 can be seen to follow a similar pattern. Take a case study of the feature subset: Noise, RMSE, Line and Correlation. The individual feature correlations (Fig. 9) are all high between those 4. What the BDT appears to have done is selected the highest performing feature from that subset, Line (Fig. 13) by a small margin, and then dismissing lots of the contributions of the other features, since they are providing a lot of the same but lower quality data.

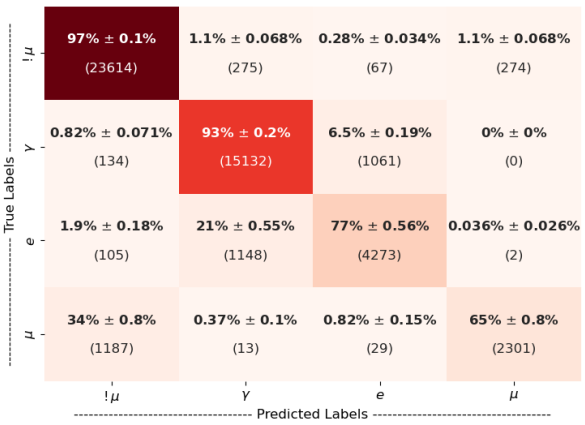

Figure 15: Particle classification BDT confusion matrix: $[\epsilon \pm \sigma \text{ (tally)}]_{ij}$.

The overall performance of the per-particle classification BDT is initially assessed as a ceiling for the performance of the event-level classification. Calling on Eq. 5, 6 for the accuracy of the particle BDT, the model achieved 92.0% ± 0.12% true positive efficiency on the particle classification.

A few of these elements are interesting for many reasons (Fig. 15). It is unsurprising that the BDT had trouble separating μ tracks with $!\mu$ tracks since the physics of both are identical. μ and charged $\pi^{\pm}$ are a recognised confusion topology. Real reconstructions have an especially tough time differentiating the two since they present awfully similar given no truth data. It comes to no surprise then that the true μ as $!\mu$ element contributes to about a third of the true μ. The BDT had no trouble however separating true $!\mu$ and μ, a vast majority (97%) of true $!\mu$ are correctly predicted.

Another confusion topology is e/γ showers. The same story is being told within this separation axis as the $\mu/\pi^{\pm}$ above. 21% of true electrons are being classed as photons, but only 6% to the contrary.

A likely hypothesis for the over-classification of photons is purely based on the number of photons included in the Cheated files. A possible danger exists with a large number of noisy photon particles (section II.J, Fig. 7) for particles with a

low number of hits. A cut any lower than 5 for example would overwhelmingly pollute the data and effectively train the BDT on largely scattered photons that real reconstructions would not pick up on, and are a product of the perfect reconstruction of the Cheated simulation. If that were the case, the overall efficiency would likely be vast due to the majority of the training and testing data being noisy photons. In the binomial efficiency (Eq. 5) calculation the photon true positive rate would dominate, leading to an accuracy that misrepresents the aims of the BDT pipeline. With the cut of 15 hits per particle discussed previously, this issue is avoided.

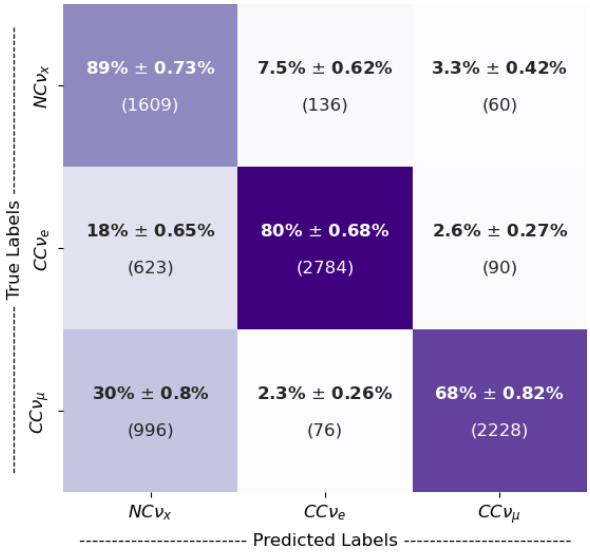

Figure 16: Event classification BDT confusion matrix.

After implementing the candidate lepton (CL) as above and comparing the BDT + CL predictions with the truth values gives an overall accuracy (Eq. 5, 6) of $78.1\% \pm 0.45\%$. The upper bound has been given by the model in Fig. 10 to be a 92% accuracy, so classifying an event through some basic python logic atop the BDT predictions to 78% is not surprising, this is a strong result that is heavily bottle-necked by a reconstruction.

E. Further Discussion

Before concluding the report, there are a few studies completed that provided helpful understanding of the machine learning process along the way.

E..1 Track/Shower BDT

After completing the likelihood model, the project pair implemented the exact same features over the same training and testing set for various boosted decision tree packages for further analysis of the likelihood workflow. Some BDT types tested include: AdaBoost, XGBoost and GradientBoosting. XGBoost was minimally more performant than the others in the track/shower model, so that was the chosen boosting algorithm for all proceeding BDTs. The XGB model achieved track/shower separation with $97.71\% \pm 0.07\%$ accuracy.

E..2 Artificial Neutral Networks (ANN)

The track/shower separation problem was also handled by a simple sequential tensor flow ANN [24]. The overall test ac-

curacy for true track/shower efficiency was 97.63% ± 0.07%, using 3 layers and 20 epochs.

The result is promising here, but the idea of ANNs was quickly dismissed as they can be difficult to tune compared to a BDT. Seeing as the ANN and BDT performed to a near identical level, the industry standard and more forgiving route was taken over to the the event classification route.

V. Conclusion

The track/shower likelihood ended with an overall accuracy of 93.77% ± 0.11%. Branching off into event classifications, the particle BDT classified μ, $!\mu$, e, γ with a 92.0% accuracy, combined with the CL logic at 87.7% ± 0.34% , the final two-stage event classifier pipeline can classify an event with a 78.1% ± 0.45% accuracy.

The strongest features for both track/shower and overall particle separation was 'line', performing highly in all regards. 'Step length' was the more dominant of the e/γ features, contributing more decision weight by a factor of 3.

An important point to revisit is the limits of real reconstructions. Cheated files are near perfect simulated reconstructions that, to reiterate, avoid conflating two large challenges within one. Dealing with inaccurate reconstructions is not favourable, and Cheated files provide a ceiling of what is possible when reconstructions improve. It is equally important to mention however that while the study of perfect reconstructions provide important insight, the studies are not necessarily representative of what will be seen when DUNE is first turned on. Since so many neutrino events will be observed every day, it is neither practical, accurate nor efficient to have specialists annotating the events manually. DUNE will then place a firm demand on accurate and time efficient reconstructions, such as Pandora.

What needs to be kept in mind in the case of improving reconstruction algorithms is the models sensitivity to the training data. Models need to be trained and tested on statistically similar data. This may require refinements of the simulation when the data arrives. Careful use of data from LArTPCs when applying machine learning models is to be encouraged at DUNE.

References

[1] Abi, B., Acciarri, R., Acero, M.A., Adamowski, M., Adams, C., Adams, D., Adamson, P., Adinolfi, M., Ahmad, Z., Albright, C.H. and Soplin, L.A., (2018). *The DUNE far detector interim design report volume 1: physics, technology and strategies.* arXiv preprint arXiv:1807.10334.

[2] Buchmüller, W., Peccei, R.D. and Yanagida, T., (2005). *Leptogenesis as the origin of matter.* Annu. Rev. Nucl. Part. Sci., 55(1), pp.311-355.

[3] Acciarri, R., Adams, C., An, R., Aparicio, A., Aponte, S., Asaadi, J., Auger, M., Ayoub, N., Bagby, L., Baller, B. and Barger, R., (2017). *Design and construction of the MicroBooNE detector.* Journal of Instrumentation, 12(02), p.P02017.

[4] Abi, B., Acciarri, R., Acero, M.A., Adamov, G., Adams, D., Adinolfi, M., Ahmad, Z., Ahmed, J., Alion, T., Monsalve, S.A. and Alt, C., (2020). *Deep underground neutrino experiment (DUNE), far detector technical design report, volume II: DUNE physics.* arXiv preprint arXiv:2002.03005.

[5] Dunescience.org. (2018). *Deep Underground Neutrino Experiment.* [online] Available at: https://www.dunescience.org.

[6] Abud, A.A., Abi, B., Acciarri, R., Acero, M.A., Adames, M.R., Adamov, G., Adamowski, M., Adams, D., Adinolfi, M., Aduszkiewicz, A. and Aguilar, J., (2022). *Scintillation light detection in the 6-m drift-length ProtoDUNE Dual Phase liquid argon TPC.* The European Physical Journal C, 82(7), pp.1-29.

[7] Blumenthal, G.R. and Gould, R.J., (1970). *Bremsstrahlung, synchrotron radiation, and compton scattering of high-energy electrons traversing dilute gases.* Reviews of modern Physics, 42(2), p.237.

[8] Marshall, J.S., Blake, A.S.T., Thomson, M.A., Escudero, L., De Vries, J. and Weston, J., (2017, September). *The Pandora multi-algorithm approach to automated pattern recognition in LArTPC detectors.* In Journal of Physics: Conference Series (Vol. 888, No. 1, p. 012142). IOP Publishing.

[9] Pivarski, J., Schreiner, H., Hollands, A., Das, P., Kothari, K., Roy, A., Ling, J., Smith, N., Burr, C. and Stark, G., (2017). *Uproot.* Zenodo.

[10] Krauss, F., Lin, C.J., Navas, S., Richardson, P. and Sjöstrand, T., (Revised 2012). *39. MONTE CARLO PARTICLE NUMBERING SCHEME.*, pp.3-5.

[11] Osborne, I. and Pivarski, J., (2023). *Awkward to RDataFrame and back.* arXiv preprint arXiv:2302.09860.

[12] Harris, C.R., Millman, K.J., Van Der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J. and Kern, R., (2020). *Array programming with NumPy.* Nature, 585(7825), pp.357-362.

[13] Bisong, E. and Bisong, E., (2019). *Matplotlib and seaborn.* Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners, pp.151-165.

[14] McKinney, W., (2011). *pandas: a foundational Python library for data analysis and statistics.* Python for high performance and scientific computing, 14(9), pp.1-9.

[15] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., (2011). *Scikit-learn: Machine learning in Python.* the Journal of machine Learning research, 12, pp.2825-2830.

[16] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. and Van Der Walt, S.J., (2020). *SciPy 1.0: fundamental algorithms for scientific computing in Python.* Nature methods, 17(3), pp.261-272.

[17] Brownlee, J., (2016). *XGBoost With python: Gradient boosted trees with XGBoost and scikit-learn.* Machine Learning Mastery.

[18] joblib.readthedocs.io. *Joblib: running Python functions as pipeline jobs — joblib 1.3.2 documentation.* [online] Available at: https://joblib.readthedocs.io/en/stable/.

[19] Han, C., Rao, N., Sorokina, D. and Subbian, K., (2020, June). *Scalable feature selection for (multitask) gradient boosted trees.* In International Conference on Artificial Intelligence and Statistics (pp. 885-894). PMLR.

[20] Putatunda, S. and Rama, K., (2018, November). *A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost.* In Proceedings of the 2018 international conference on signal processing and machine learning (pp. 6-10).

[21] Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y. and Cohen, I., (2009). *Pearson correlation coefficient.* Noise reduction in speech processing, pp.1-4.

[22] Brown, A. and Suit, H., (2004). *The centenary of the discovery of the Bragg peak.* Radiotherapy and Oncology, 73(3), pp.265-268.

[23] Highland, V.L., (1975). *Some practical remarks on multiple scattering.* Nuclear Instruments and Methods, 129(2), pp.497-499.

[24]