Prepared for submission to JINST

Particle identification in the GlueX detector using a neural network

---

E. Habjan[a,b] **R. Dube**[a,c] **J. McIntyre**[a] **M. Edo**[a] **R. Jones**[a]

[a] *University of Connecticut, Department of Physics,*
196 Auditorium Road, Unit 3046, Storrs, CT 06269, USA
[b] *Northeastern University, Department of Physics,*
100 Forsyth Street #111, Boston, MA 02115, USA
[c] *Indiana University, Department of Physics,*
727 E. Third St., Swain Hall West Room 152, Bloomington, IN 47405

E-mail: `habjan.e@northeastern.edu`

Abstract: In particle physics experiments, identifying the types of particles measured in a detector is essential for the accurate reconstruction of event data. At Thomas Jefferson National Accelerator Facility (Jefferson Lab), the GlueX experiment performs particle identification (PID) by setting specific thresholds, known as cuts, on the kinematic properties of tracks and showers obtained from detector hits. Our research aims to enhance this cut-based method by employing machine learning (ML) through neural networks (NN). This approach offers an exciting opportunity to uncover underlying correlations among PID variables in the reconstructed kinematic data. Our study illustrates that a NN can identify charged and neutral particles in Monte Carlo (MC) simulated GlueX data with significantly improved precision over the current cut-based PID method.

Keywords: Particle identification methods, Analysis and statistical methods

Contents

1 Introduction

Since the 1970s, there has been significant interest in understanding the mechanism behind the confinement of quarks and gluons within quantum chromodynamics (QCD). The confinement mechanism describes how quarks and gluons, which carry color charge, are permanently bound together to form protons, neutrons, and other color-neutral hadrons. This phenomenon prevents quarks and gluons from existing as free particles, ensuring they are always observed within larger, composite color-neutral particles.

The constituent quark model (CQM) [1] treats the simplest hadron configuration, a meson, as a bound quark-antiquark ($q\bar{q}$) state grouped into SU(N) flavor multiplets. While the CQM provides a natural framework to classify mesons, it is unable to explain quark confinement since it does not require gluons, the mediator of the strong force, which lattice QCD [2] and the Flux Tube Model [3] deems to play an essential role in QCD. The Flux Tube Model for hadrons was extracted from the strong-coupling Hamiltonian lattice formulation of QCD and contains within it the quark model. This model explains confinement and allows for pure gluon states, hybrids (particles having both quark and gluonic degrees of freedom), and other exotics (particles with quantum numbers not permitted by the CQM). Hybrid mesons ($q\bar{q}g$) can possess quantum numbers (J^{PC}) not possible from the CQM ($q\bar{q}$); therefore, the discovery of these *exotic hybrid mesons* would be confirmation of the Flux Tube Model.

The GlueX experiment located in Hall D at Jefferson Lab studies the gluon field inside the meson ($q\bar{q}g$) through the mapping of the spectrum of exotic hybrid mesons with masses up to 2.5 $^{GeV}/_{c^2}$. Identifying these mesons requires knowledge of their production mechanism, identification of their J^{PC}, and measurements of their decay modes. This is accomplished through the partial wave analysis (PWA) of exclusive final states. This technique depends in part upon having high statistics, linear polarization of the incident photon beam, excellent measurement resolution, full acceptance in decay angles, and correct decay particle identification (PID).

Particle identification (PID) in the GlueX detector [4], depicted in Figure 1, is accomplished using seven detector systems:

- **SC:** A start counter consisting of a cylindrical array of 40 scintillator paddles
- **TOF:** A forward time-of-flight wall of 44 vertical and 44 horizontal scintillator paddles
- **BCAL:** A cylindrical geometry barrel calorimeter with alternating layers of lead and scintillating fiber
- **FCAL:** A forward lead-glass block calorimeter with a planar geometry orthogonal to the photon beam axis
- **CDC:** A central drift chamber detector consisting of layers (strung in axial and stereo configurations) of cathode straw tubes, each containing an anode wire and filled with a mixture of argon and CO_2 gas
- **FDC:** A forward drift chamber detector consisting of 24 planar drift chambers with cathode strip and wire readouts, and filled with a mixture of argon and CO_2 gas
- **DIRC:** A Detection of Internally Reflected Cherenkov light detector used for forward region PID

The SC envelopes the target cell covering $\sim$90% of 4π solid angle and is the first detector to measure particles emitted from interactions in the target. Due to its proximity to the target cell, the SC provides a timing signal relatively independent of particle type and trajectory and provides a fast signal that is used in the level-1 trigger for the experiment. The thin scintillator SC is used to identify the 4 ns accelerator electron radio-frequency (RF) beam bucket associated with detected particles. Energy deposition ($^{dE}/_{dx}$) in the SC, in combination with the flight time from the TOF, is utilized for charged particle identification. The TOF provides pion and kaon separation up to a momentum of about 2 $^{GeV}/_{c}$ and is located just upstream of the FCAL, 5.5 m downstream of the target, covering polar angle $\theta \in [1^o, 11^o]$. The TOF provides PID through the measurement of a particle's velocity in the low momentum range and time of flight information with respect to the accelerator RF beam buckets.

The time of flight of decay particles is accomplished with the TOF and the BCAL. The BCAL is a barrel-shaped electromagnetic calorimeter comprised of a matrix of lead and scintillating fibers residing inside the GlueX 2.08 Tesla solenoid magnet, covering a polar angle $\theta \in [11^o, 126^o]$. The BCAL measures the time and energy deposited by charged and neutral decay particles. The FCAL comprises 2800 lead-glass modules, each coupled to a photomultiplier (PMT) and stacked in a

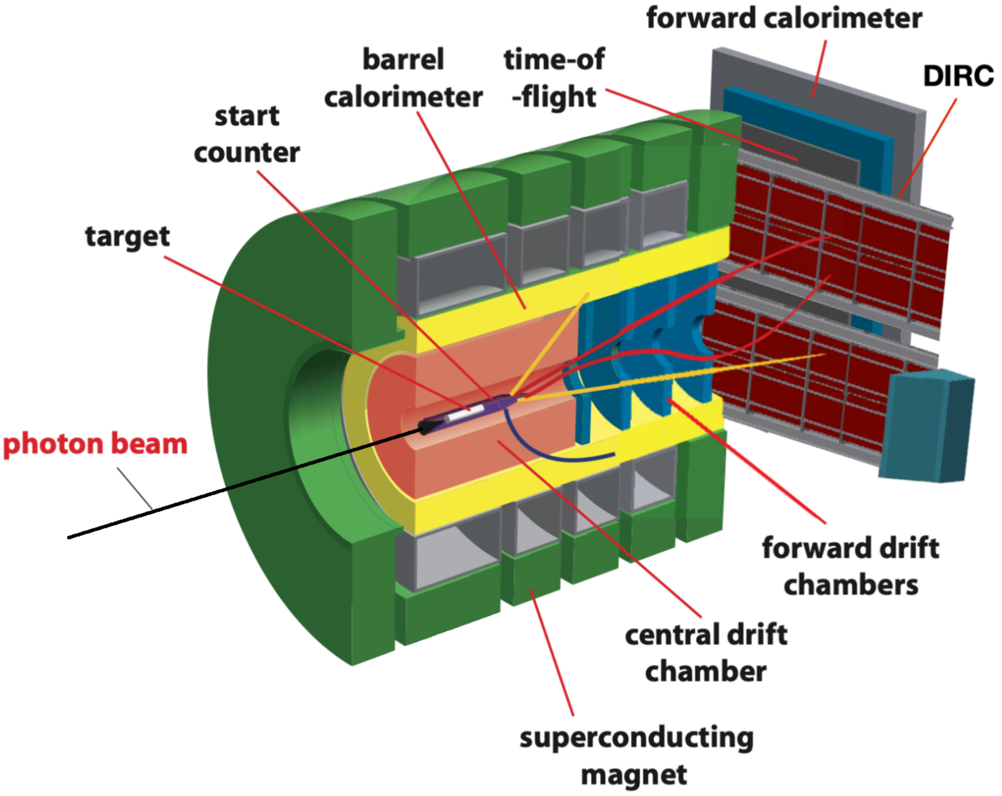

Figure 1: GlueX spectrometer schematic.

circular array inside a light-tight dark box enclosure. Decay particles that interact with the FCAL create an electromagnetic shower that is read out by the PMTs and used to reconstruct detector events. The CDC and FDC work together to provide a complete picture of the charged particle tracks, from low to high momentum and from central to forward angles.

The DIRC [5] is located about 4 m from the target and covers a polar angle range of 2^o to 11^o. As charged particles pass through the DIRC, they emit Cherenkov light while traveling inside fused silica. A fraction of this light is transported to the photon camera, where it is expanded and imaged on a pixelated photodetection plane. The resulting pattern on the pixelated photodetection plane provides information about the velocity of the charged particle, given the momentum vector reconstructed by the tracking system.

In this work, we aim to demonstrate the potential of ML [6] as a tool to improve PID accuracies for GlueX analyses. ML has broadened the range of possibilities in various fields, which includes a GlueX study [7] that used a discriminating NN to reduce background in the FCAL. Motivated by such studies and the demand for high-accuracy PID, this work makes use of NNs to classify single hadronic particles using GlueX MC simulation data. Manual PID methods currently utilized by GlueX are carried out on the same MC simulation data sets and compared to the PID accuracies obtained from the charged particle NN models.

2 Monte Carlo Simulation Data Set

The training and test data sets were extracted from low-momentum GlueX particle gun simulations; see Table 1. In these simulations, a particle is spawned at a randomized location, known as a vertex, within the 30 cm long liquid H_2 target cell. The generated particle is imparted with a random trajectory and an arbitrary momentum of less than 1 ${}^{GeV}\!/_{c}$. Detailed simulations of interactions between the generated particle and the GlueX detector and any subsequent decays and interactions are performed using a GEANT4-based software package [8, 9]. The resulting simulated detector signals (e.g., hits) are stored in Hall D Data Model (HDDM) format[1] and reconstructed using the `halld_recon` package for shower and track identification, significantly decreasing the number of features needed to describe each event. The last step of the simulation converts the data from the hierarchical HDDM format to a tabular format for use in manual or NN PID. The labels for each quantity in our final data set are shown in Table 2.

Table 1: List of the particles generated using MC particle gun simulations for training and testing of NN PID models.

Particle	Generated Events	Training Data Set	Test Data Set
e^+	120,000	80,000	40,000
e^-	120,000	80,000	40,000
π^+	120,000	80,000	40,000
π^-	120,000	80,000	40,000
μ^+	120,000	80,000	40,000
μ^-	120,000	80,000	40,000
K^+	120,000	80,000	40,000
K^-	120,000	80,000	40,000
p	120,000	80,000	40,000
$\bar{p}$	120,000	80,000	40,000
γ	120,000	80,000	40,000
n	120,000	80,000	40,000
K_L^0	120,000	80,000	40,000

MC particle gun simulations allow for easy event labeling; unfortunately, some resulting particle decays and interactions observed within the detector produce tracks and showers not directly attributed to the generated particle. To reduce these types of events, we exclude events where the MC-generated particle decays before exiting the furthest GlueX detector component. Since the simulation spawns particles at a vertex within the target cell, the training and test data sets exclude events where the initially generated particle decays, forming a second vertex, before exiting the GlueX detector. Using the truth information of the MC simulation, we remove from our data sets any event with more than one vertex occurring within the first 500 ns of the simulation.

---

[1] https://github.com/rjones30/HDDM/blob/main/docs/users_guide.pdf

To eliminate events where interactions with the detector produced secondary tracks or showers, limits (i.e., cuts) were placed on the number of tracks and showers per event in our data sets. Events generating a charged particle (e.g., $\mu^{\pm}$, $\pi^{\pm}$) were only included in the data sets if the reconstructed event contained only one track with a single track-associated shower. Meanwhile, events that generated neutral particles (e.g., n, γ, K_L^0) were required to have precisely one shower and no tracks. Although these cuts may inflate the accuracy of the manual and NN PID technique due to the exclusion of complicated interactions with the detector, these cuts were necessary to ensure the event label matched the particle producing the shower or track in the training and test data sets.

The training data set, which included 80×10^3 events per particle type, played a crucial role in the accuracy of the NN PID technique. Our study included only the track hypothesis that matched the generated particle type for simulation events generating charged particles. Since no muon hypothesis existed in the default reconstruction software, the pion track hypothesis was used in the training data set for events that generated a muon. Each row of the training data set represented a different event, while a row in the test data set corresponded to each hypothesis for events with charged generated particles. The event number was identified by the `group` label, which only appeared in the test data set. There was only one row in the test data set per event for neutral particles, as no hypotheses were used in the shower reconstruction process. The test data set contained 40×10^3 events per particle type. However, the number of rows in this data set was substantially more extensive due to the inclusion of multiple hypotheses per event for simulations where a charged particle was generated.

3 DNN model description

In this Section, we explain and justify the `Tensorflow` implementations of the Adam optimizer, the cross entropy loss function, the activation functions, and Hyperband optimization.

3.1 Cross Entropy Loss Function

The advent of logistic regression [11] and the creation of cross-entropy in the early years of information theory have evolved into a loss function ubiquitous in ML, referred to as the Cross-Entropy Loss Function. Minimizing cross-entropy between two distributions is equivalent to maximizing the log-likelihood [12]. The log-likelihood can be defined as:

$$l(\theta) = \frac{1}{N}\sum_{i=1}^{N} log(P(x_i|\theta)) \tag{3.1}$$

where θ defines our parameter space (e.g., energy loss, momenta) and x_i is a given detection (i.e., a particle). By maximizing the log-likelihood, we can best predict the probability of detecting a given x_i when provided with θ. The cross-entropy $H(P_D(x), P_\theta(x))$ is also defined in terms of the probability of x_i and θ:

Table 2: Feature labels of the particle gun data set.

Column	Unit	Description	Overflow Value
`true ptype`		The true generated particle type (GEANT3 coding [10])	
`ptype`		Particle hypothesis (GEANT3 coding)	
`group`		Event number	
`E`	GeV	Particle total energy	-5
`px`	GeV/c	Particle momentum X-component	-500
`py`	GeV/c	Particle momentum Y-component	-500
`pz`	GeV/c	Particle momentum Z-component	-500
`q`	e	Particle charge	-10
`E1E9`		E1/E9 ratio for the matched FCAL cluster	-5
`E9E25`		E9/E25 ratio for the matched FCAL cluster	-5
`docaTrack`	cm	Impact parameter of track to FCAL cluster	-5
`preshowerE`	GeV	Shower energy in the 1st layer of the BCAL	-5
`sigLong`	cm	RMS of BCAL shower along depth	-5
`sigTrans`	cm	RMS of BCAL shower along azimuth	-5
`sigTheta`	rad	RMS of BCAL shower along Z	-5
`E_L2`	GeV	Shower energy in the 2nd layer of the BCAL	-5
`E_L3`	GeV	Shower energy in the 3rd layer of the BCAL	-5
`E_L4`	GeV	Shower energy in the 4th layer of the BCAL	-5
`dEdxCDC`	keV/cm	Average dE/dx of track in the CDC	-5
`dEdxFDC`	keV/cm	Average dE/dx of track in the FDC	-5
`tShower`	ns	Mean shower time in the BCAL or FCAL	-10
`thetac`	rad	Track Cerenkov angle measured by DIRC	-5
`bCalPathLength`	cm	Track distance from vertex to BCAL entry	-5
`fCalPathLength`	cm	Track distance from vertex to FCAL entry	-5
`dEdxTOF`	keV/cm	Average track dE/dx in the TOF	-5
`tofTOF`	ns	Time from track vertex to impact on the TOF	-5
`pathLengthTOF`	cm	Distance from track vertex to impact on the TOF	-5
`dEdxSc`	keV/cm	dE/dx of track in the SC	-5
`pathLengthSc`	cm	Distance from track vertex to impact on the SC	-100
`tofSc`	ns	Time from track vertex to impact on the SC	-100
`xShower`	cm	Shower X-component	-500
`yShower`	cm	Shower Y-component	-500
`zShower`	cm	Shower Z-component	-500
`xTrack`	cm	Track X-component	-500
`yTrack`	cm	Track Y-component	-500
`zTrack`	cm	Track Z-component	-500
`CDChits`		Number of straws in the CDC producing hits	-5
`FDChits`		Number of anode wires in the FDC producing hits	-5
`DOCA`	cm	Impact parameter of track at the BCAL cluster	-5
`deltaz`	cm	Impact parameter of track at the BCAL along Z	-100
`deltaphi`	rad	Impact parameter of track at the BCAL along azimuth	-10
`tFlightSc`	ns	Calculated time from vertex to SC	
`tFlightBCAL`	ns	Calculated time from vertex to BCAL	
`tFlightTOF`	ns	Calculated time from vertex to TOF	
`tFlightFCAL`	ns	Calculated time from vertex to FCAL	

$$\begin{aligned} H(P_D(x), P_\theta(x)) &= -\frac{1}{N}\sum_{i=1}^{N} log(P(x_i|\theta)) \\ &= -l(\theta) \end{aligned} \tag{3.2}$$

Thus, maximizing the log-likelihood is equivalent to minimizing the cross-entropy, known as the cross-entropy and maximum likelihood principle. The `Tensorflow` implementation of the cross-entropy loss function is used during training for the NN models.

3.2 Adam Optimizer

The Adam optimizer [13] is a sophisticated method for optimizing complex objective functions through gradient-based parameter adjustments. By combining the strengths of the AdaGrad [14] and RMSProp [15] optimization methods, Adam excels in first-order gradient-based optimization of stochastic functions. It uses estimates of the first and second moments of gradients to compute adaptive learning rates for each parameter, making it highly effective in high-dimensional parameter spaces. Due to the high level of stochasticity in experimental particle physics data, Adam is particularly well-suited for minimizing the Cross-Entropy Loss Function.

3.3 Structure and Activation Functions

A NN structure contains an input layer, one or more hidden layers, and an output layer. The input layer of our models is comprised of 38 nodes, which is equal to the number of feature labels used in training, shown in Table 2. Each of the hidden layers of the NN makes use of the **Re**ctified **L**inear **U**nit (ReLU) activation function [16, 17], and is defined in Equation 3.3.

$$f(x) = max(0, x) \tag{3.3}$$

For any input x from a previous neuron, a non-negative output $f(x)$ will be produced from that neuron. The non-linearity of ReLU introduces sparsity and avoids saturation at large values while remaining simple. These advantages allow for computational efficiency during training and for meaningful connections to be drawn between complex relationships in the data. In the output layers of the NN models, the sigmoid activation function is used and is defined by Equation 3.4.

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.4}$$

For an input x, the output $S(x)$ will always be between 0 and 1. This property of the sigmoid activation function is particularly useful in the output layers of the NN models. It allows for a confidence-based prediction when classifying particles, as the ouput can be interpreted as an array of probabilities that correspond to each particle in the respective data set.

3.4 Hyperband Algorithm

In order to minimize the Cross-Entropy Loss Function, an optimization process is used to determine the number of hidden layers, the number of neurons in each hidden layer, and the learning rate of the Adam optimizer. In this work, the `Hyperband` algorithm [18] is implemented to determine the optimized values of each hyperparameter. Hyperband is chosen as the optimization algorithm due to its computational efficiency and superior performance compared to Bayesian optimization. This algorithm selects different sets of hyperparameters and trains the neural network for a fixed number of epochs. It employs a technique known as *successive halving*, where only half of the models with the lowest cross-entropy loss are allocated resources to continue training after a specified number of epochs have passed. This procedure is repeated until only a single set of hyperparameters remains, which are then used to train the NN models that are used for PID.

4 Methods

In this Section, a description is given of the manual PID cuts and the training process for the NN models used for PID.

4.1 Manual PID

In this work, we identify pions ($\pi^{\pm} \approx 140\,{}^{MeV}\!/\!{}_{c^2}$) and muons ($\mu^{\pm} \approx 106\,{}^{MeV}\!/\!{}_{c^2}$) as the same particle, denoted as $\pi^+|\,\mu^+$ or $\pi^-|\,\mu^-$. Our PID method introduces this simplification since pions and muons have similar masses, and the GlueX detector does not have a hadronic calorimeter, which makes distinguishing between these particles difficult. Muons can be discerned from pions by investigating the momentum distributions of a given event in the FCAL. However, this must be performed before reconstruction and is outside the scope of this paper.

The timing cuts implemented in this paper use the Spring 2017 Analysis Launch Cuts [4]; each is shown in Table 4. The measured BCAL and FCAL times were recorded as a single variable, `tShower`, in the simulation data set; thus, the BCAL and FCAL time measurements must be distinguished. If an event has a detection for `E_L2`, then `tShower` is labeled as the mean shower time in the BCAL, and if there is a detection for `E1E9`, then `tShower` is labeled as the mean shower time in the FCAL. The difference between the mean shower times in each detector is taken with the calculated time from the vertex to the BCAL (`tFlightBCAL`) or the FCAL (`tFlightFCAL`). In order to assess the quality of a given hypothesis, a χ^2 value is calculated between the mean shower time and calculated shower times. Only hypotheses with a χ^2 value of less than 0.075 are considered robust; any hypotheses above this threshold are labeled as no identification (no ID). Only timing information is available for charged particles in the simulation data set; therefore, no timing cuts were made for neutral particles.

In addition to timing cuts, track energy loss cuts are implemented using the `dEdxCDC` variable and the magnitude of the particle momentum. To create a decision boundary between each particle, the functional form of the equations utilized in the Spring 2017 Analysis Launch Cuts are used and are seen in Equation 4.1.

$$f_i(p) = \exp(a_i p + b_i) + c_i \tag{4.1}$$

In Equation 4.1, p is the momentum ($^{GeV}/_{c}$) of a particle, $^{dE}/_{dx}$ is the energy loss ($^{keV}/_{cm}$) in the CDC, and e is Euler's number. Using the training data set, the number of incorrectly identified particles is minimized by treating constants a_i, b_i, and c_i as free parameters and utilizing the `minimize` method from the `scipy.optimize` [19] module. The $^{dE}/_{dx} - p$ decision boundaries are only derived for charged particles since insufficient data is available for neutral particles in the simulation data set. The constants for the optimized decision boundaries are shown in Table 3

Table 3: The optimized decision boundary parameters.

$f_i(p)$	a_i / [$^{GeV}/_{c}$]	b_i	c_i / [$^{KeV}/_{cm}$]
$f_1(p)$	-5.095	-10.205	$2.080 \cdot 10^{-6}$
$f_2(p)$	-3.947	-12.284	$1.936 \cdot 10^{-6}$
$f_3(p)$	-0.185	-19.215	$2.190 \cdot 10^{-6}$

Each optimized decision boundary is overlaid on the test data set in Figure 2. An additional manual cut is made for electrons and muons/pions, in which the ratio of the particle's total energy `E` to the magnitude of the momentum is taken, with a decision boundary at $^{E}/_{p} = 0.83\,c$. Lastly, a hypothesis is only considered if the particle hypothesis matched the predicted hypothesis from the manual PID criteria shown in Table 4. A PID is made for every hypothesis in our test data set that meets the χ^2 criteria; however, if a given event met none of the criteria, then no ID is designated. Furthermore, if an event has two or more PIDs that match different hypotheses in the test data set, the particle type with the highest χ^2 value is designated. The confusion matrix presenting the results of our manual PID on generated charged particles is shown in Figure 3.

Table 4: Manual PID cuts. If an entry is missing, there is no cut for that particle. The $^{dE}/_{dx_1}$, $^{dE}/_{dx_2}$ and $^{dE}/_{dx_3}$ cuts corresponded to equation 4.1 with variables listed in Table 3.

Particle	Δt BCAL [ns]	Δt FCAL [ns]	$^{dE}/_{dx}$ [$^{keV}/_{cm}$]	$^{E}/_{p}$ [c]
$e^{\pm}$	$\pm$ 1.0	$\pm$ 2.0	$f_3(p) < {}^{dE}/_{dx} < f_2(p)$	> 0.83
$\pi^{\pm}$ / $\mu^{\pm}$	$\pm$ 1.0	$\pm$ 2.0	$^{dE}/_{dx} < f_3(p)$	< 0.83
$K^{\pm}$	$\pm$ 0.75	$\pm$ 2.5	$f_2(p) < {}^{dE}/_{dx} < f_1(p)$	
p / $\bar{p}$	$\pm$ 1.0	$\pm$ 2.0	$f_1(p) < {}^{dE}/_{dx}$	

4.2 Neural Network PID

Before the training of the NN models is performed, two data pre-processing steps are required. The first data pre-processing step is separating the test and training data sets into charged or neutral particle data sets, such that separate NN models are trained for neutral and charged particles. The second pre-processing step is replacing any parameter in the data sets that did not obtain a value from the simulation (e.g., no detector hits) with the 'Overflow Value' seen in Table 2. This is done

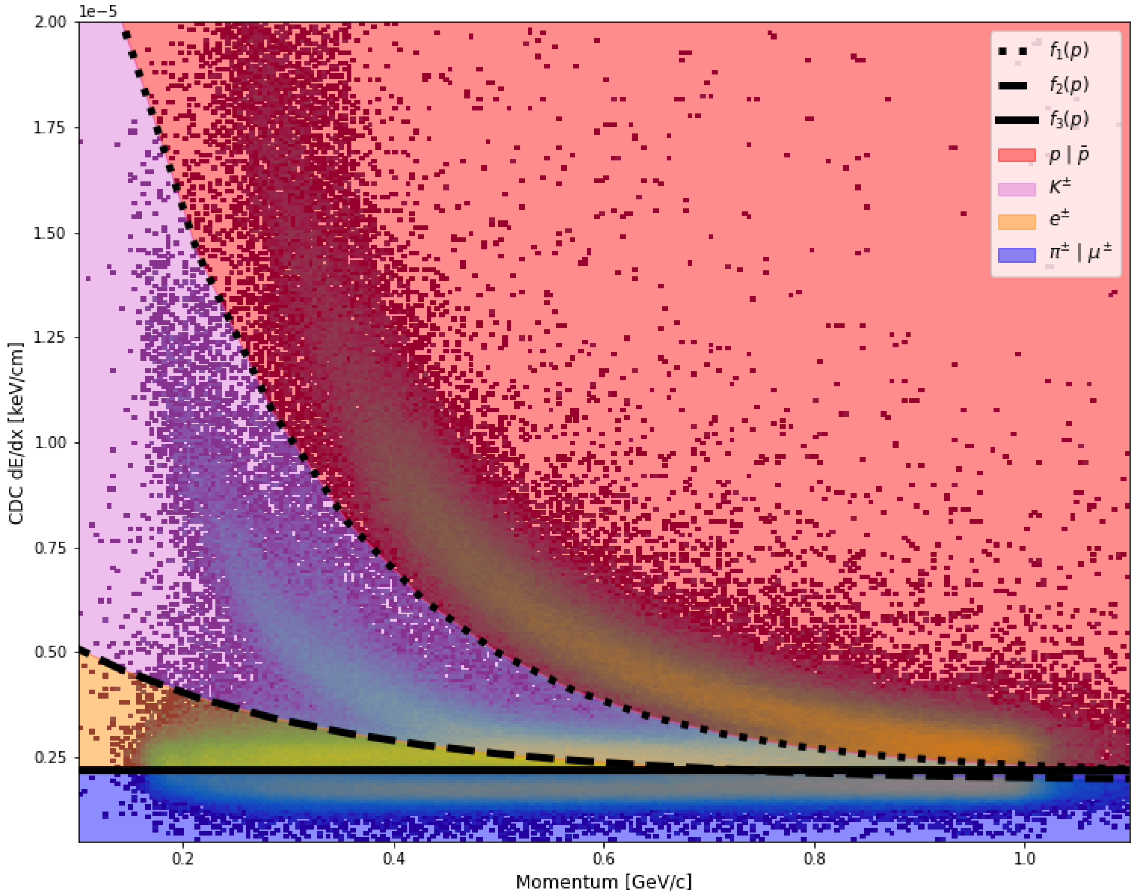

Figure 2: A 2-dimensional histogram of the average track energy loss in the CDC plotted against the magnitude of momentum vectors from the test data set. The manual PID cuts described in Section 4.1 are overlaid to show the classification boundaries; $f_1(p)$ is shown with the dotted line, $f_2(p)$ is shown with the dashed line, and $f_3(p)$ is shown with the solid line. The functional form of each decision boundary is shown in equation 4.1 and the constants for each equation in Table 3. Regions of the plot shaded in red are classified as p or $\bar{p}$, purple as $K^\pm$, yellow as $e^\pm$, and blue as $\pi^\pm$ or $\mu^\pm$.

because `Tensorflow` requires a real number input for training. The feature labels in Table 2 that have an Overflow Value are the 38 features used to train the charged and neutral particle NN models.

The NN models are trained using the `TensorFlow` implementations of the Cross-Entropy Loss Function, Adam Optimizer, and ReLU activation function, all described in Section 3. The number of neurons, the number of hidden layers, and the learning rate of the Adam optimizer are optimized to have the maximum validation accuracy by `Hyperband`. The variation permitted for each hyperparameter is listed in Table 5. The optimized hyperparameters are used to train a NN model for up to 50 epochs. The training is ended early if the Cross-Entropy Loss changes by less than 0.01 after five successive epochs.

The trained NN models are used to classify every hypothesis in the test data set. Each prediction is made by the `predict` method from the `Tensorflow` NN models and yields a confidence value

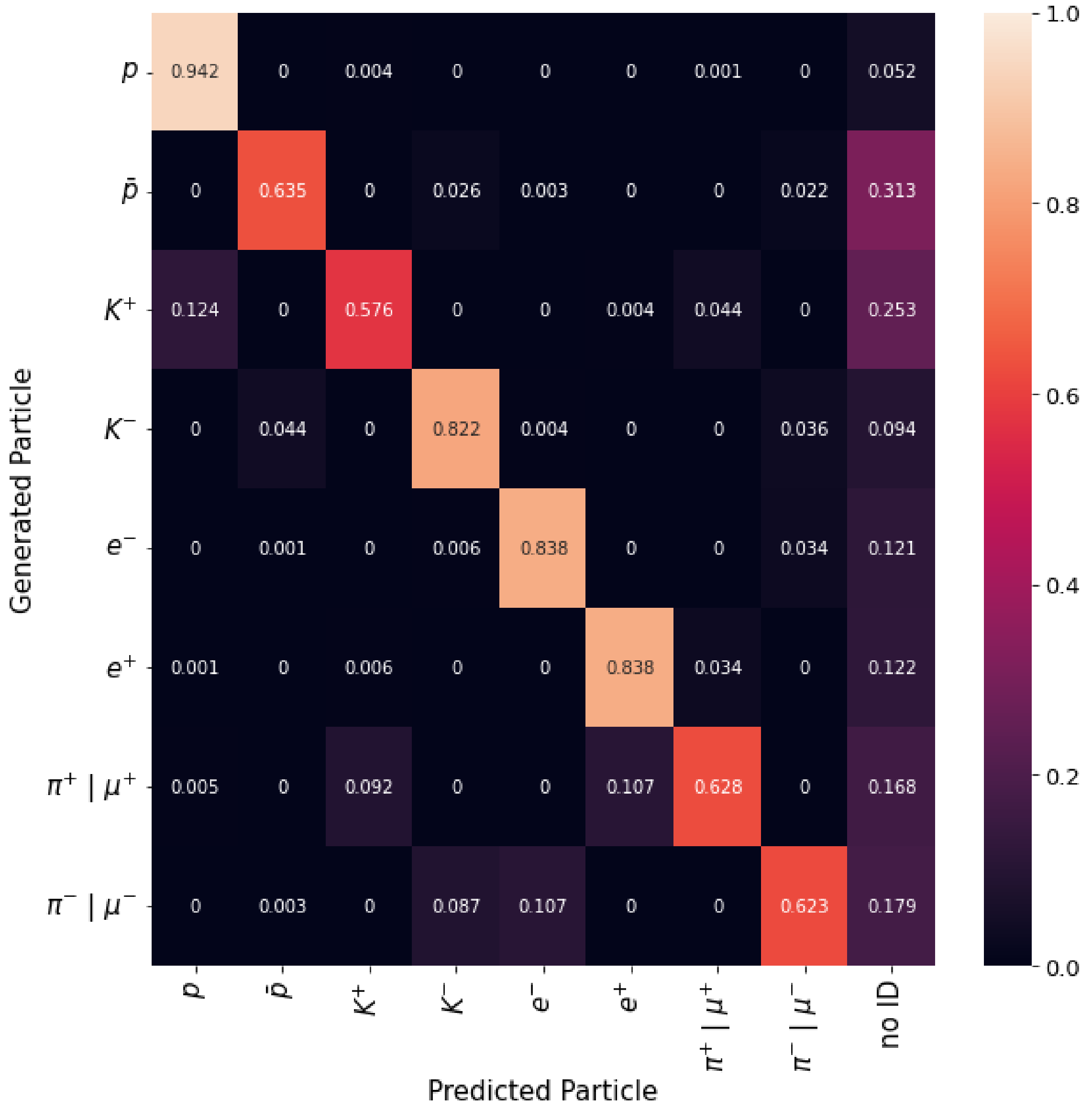

Figure 3: The confusion matrix for manual PID on charged particles is shown in Figure 3 and the confusion matrix for NN PID on charged particles is shown in Figure 4. The generated particle is shown on the y-axis, and the identified particle is shown on the x-axis. A no ID classification was given for events in the manual PID scheme that do not meet the χ^2 criteria described in Section 4.1. Similarly, a no ID classification is given for the NN PID method when the confidence criteria described in Section 4.2 is not achieved.

for each possible classification (i.e., particle). The highest confidence value across all hypotheses in an event is accepted as the PID. A no ID label is given for any PID with a confidence of less than 0.4. For the same reason discussed in Section 4.1, pions and muons with the same charge are classified as identical particles. The confusion matrix for the charged particle NN model is shown in Figure 4 and the confusion matrix for the neutral NN model is shown in Figure 5.

Table 5: Variation for each hyperparameter permitted during training.

Hyperparameter	**Variation Permitted**
Hidden layers	1 – 6
Neurons per hidden layers	100 – 600
Learning rate	10^{-4} – 10^{-2}

5 Results

This Section presents the results of the traditional PID cuts and NN PID models. It directly compares these two methods and discusses the advantages of NNs in PID. The importance of each feature in the simulation data set is presented to better understand how the NN models are making predictions.

5.1 Comparing PID techniques

The results of manual PID cuts on charged particle MC simulation data are displayed as a confusion matrix in Figure 3. The highest accuracy by the manual cuts is 94.2% for the proton (p) sample, with only 5.2% of the 40×10^3 simulated proton test events producing a no ID result. The K^-, e^-, and e^+ had manual PID accuracies range from 82% to 84%. The poorest performance of the manual PID cuts occurred in $\bar{p}$, K^+, $\pi^+ | \mu^+$, and $\pi^- | \mu^-$ test data which yield accuracies beween 57% and 64%. Many of the samples that are incorrectly identified by the manual PID method are designated as no ID. The most prominent exceptions are the K^+ test sample, which is misidentified as a p 12.4% of the time, and the $\pi^+ | \mu^+$ test sample that is misclassified as a K^+ or e^+ 9.2% and 10.7% of the time, respectively. The $\pi^- | \mu^-$ test sample is misidentified 10.7% of the time as an e^- and 8.7% as a K^-. Additionally, there are several cases in which the manual PID method misidentified test sample particles less than 5% of the time. The large number of events that did not pass the χ^2 timing cut, along with the instances of substantial particle misidentification shown in Figure 3, underscore areas where a NN PID method can enhance GlueX PID performance over the manual PID method.

Figure 4 shows the results of our NN PID method on charged particle MC simulation data displayed as a confusion matrix. Substantial improvements are achieved in particle identification for all simulated charged particle MC data samples. The most notable improvement is the reduction of events classified as no ID. All charged particle events have $< 1\%$ events classified as no ID with the exception of the $\bar{p}$ and K^- samples, which are reduced from 31.3% to 4.6% and 9.4% to 1.4%, respectively. As previously mentioned, the $\bar{p}$, K^+, $\pi^+ | \mu^+$, and $\pi^- | \mu^-$ have the lowest manual PID accuracies in the range 57% – 64%, but are correctly identified with accuracies in the range 83.5% to 90.5% by the NN models. The e^+ and e^- samples are both identified with 83.8% accuracy by the manual PID method, but are correctly identified by the NN PID method with accuracies of 96.5% and 94.5%, respectively. Compared to the manual PID method, the PID accuracy for all charged particles improved when the NN PID method was implemented. Even the highest PID accuracy in the manual PID method sample (p) is improved by 3.4% by the charged NN model. These

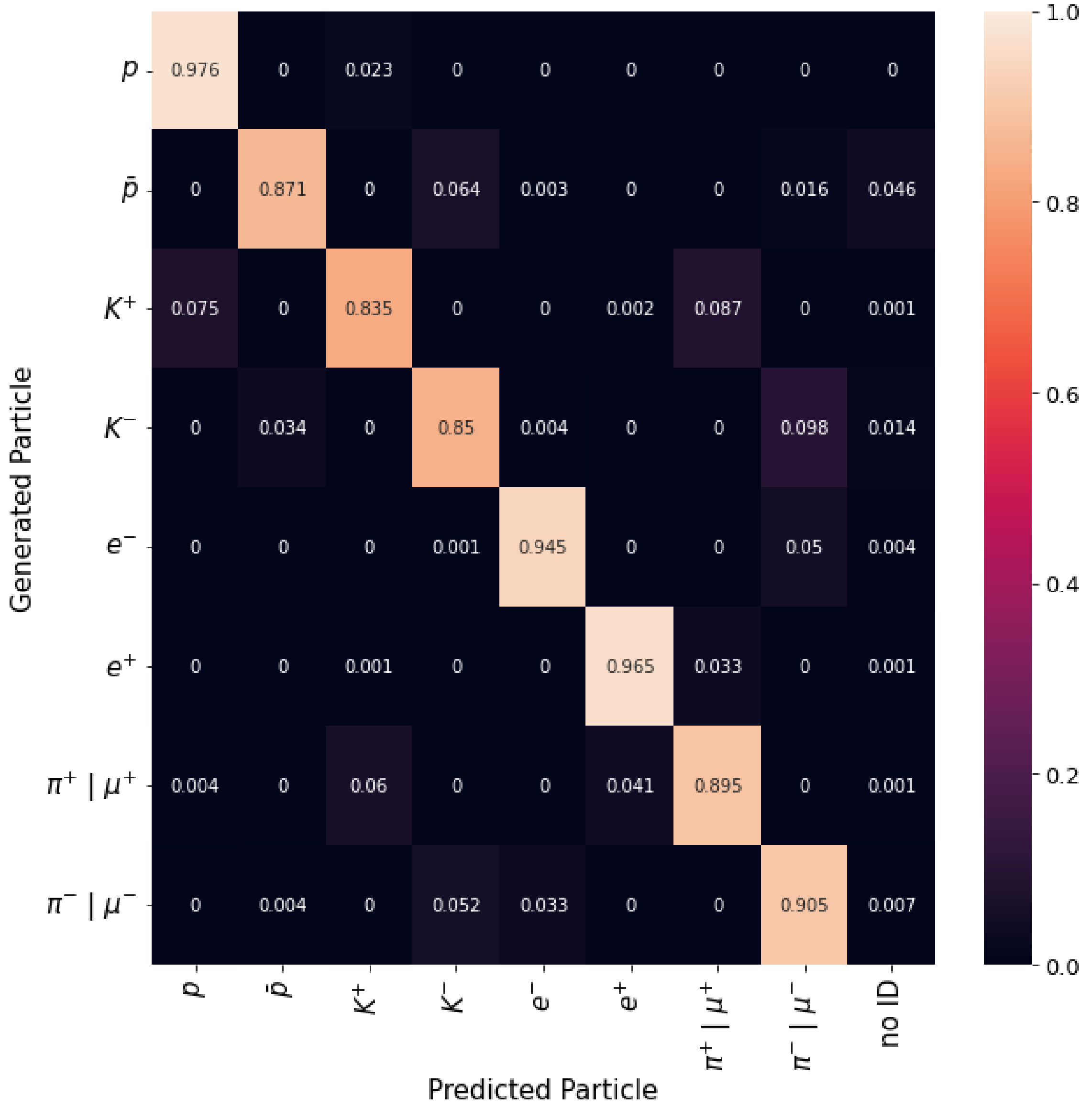

Figure 4: The confusion matrix for manual PID on charged particles is shown in Figure 3, and the confusion matrix for NN PID on charged particles is shown in Figure 4. The generated particle is shown on the y-axis, and the identified particle is shown on the x-axis. A no ID classification is given for events in the manual PID scheme that do not meet the χ^2 criteria described in Section 4.1. Similarly, a no ID classification is given for the NN PID method when the confidence criteria described in Section 4.2 is not achieved.

results show that NNs are better at PID on MC data, and that NNs have the potential to significantly increase the PID accuracies for the GlueX experiment.

Although PID performance increased for every simulated charge particle using the NN PID method, there are three increases in particle misclassification. For the $\bar{p}$, the PID accuracy increased by 23.6% and the percentage of no ID events decreased by 26.7% compared to the manual PID

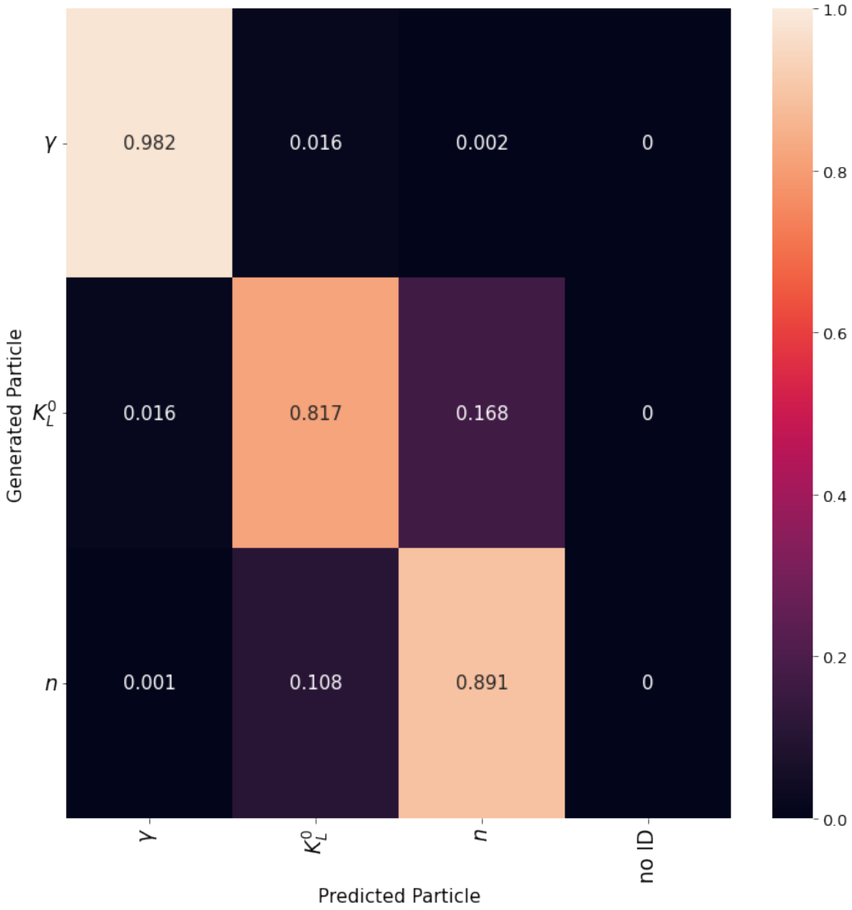

Figure 5: The confusion matrix for the NN PID for neutral particles. The generated particle is shown on the y-axis, and the particle classified by the neutral NN model is shown on the x-axis. Particles that do not meet the confidence criteria discussed in Section 4.2 are classified with a no ID label.

method, however there is a 3.8% increase in misidentification as a K^- for the NN PID method. Similarly, the K^+ sample has a 4.3% increase in being identified as a $\pi^+|\mu^+$ when using the NN PID method, but also correctly identified 25.9% more events and reduced events with a no ID classification by 25.2% when compared to manual PID. The largest increase in misidentification for the charged NN model occurred for the K^- sample in which there is a 6.2% increase in misidentification as a $\pi^-|\mu^-$, however, the K^- sample has a 2.8% increae in correclty identified events by the NN PID method and a 8% reduction in no ID classification. In summation, the only downside of the charged NN PID method is an increase of misclassification for three charged particles in three particular cases. These increases are notable, however are a necessary compromise for significant increased in PID accuracy, which can be potentially removed by using more training data and more complex ML architectures.

The classification scheme for the NN PID described in Section 4.2 specifies that the particle prediction with the highest confidence for a given event is chosen as the PID for that event. This selection criterion differs from the one used in manual PID, where the hypothesis with the lowest timing χ^2 value is selected. The timing information is crucial for some particles (i.e., e^-). However, for other particles (i.e., p), some quantities (e.g. $^{dE}\!/_{dx}$) play a more significant role in PID while the timing information is less crucial. Using the minimum timing χ^2 as a metric for the strength of a hypothesis limits the accuracy of PID for specific particles. This limitation is eliminated for the NN PID method, as the sigmoid output function provides a confidence interval between 0 and 1, which is determined by all the input features of the data set. This confidence-based decision method provides a comprehensive analysis of all the recorded quantities in a given event and avoids a bias from particles that are better characterized by any one feature.

Figure 5 shows the confusion matrix for the neutral particle NN PID model. Unlike charged particles, robust manual PID methods for classifying neutral particles do not exist in the GlueX experiment. The lone exception is the existence of timing cuts for γ. Unfortunately, the simulation data set did not recover any predicted timing values (e.g., `tFlightBCAL` and `tFlightFCAL`); thus, no manual PID could be carried out for neutral particles. Regardless, the NN PID method showed that neutral particles can be identified accurately. An accuracy of 98.2% is achieved for identifying γ, 81.7% for K_L^0, and 89.1% for n. The model that is used to achieve these classification accuracies is optimized by the `HyperBand` algorithm, which found optimal validation accuracy for a NN model with three hidden layers and a total of $\sim 2 \times 10^5$ parameters. These optimization values differ from the single hidden layer with $\sim 4 \times 10^4$ parameters of the charged NN model. The near order of magnitude increase in complexity for the neutral NN PID model illustrates the difficulty of neutral PID in the GlueX detector; however, it is demonstrated that neutral PID is possible through the use of ML. This success can still be improved upon, as there is substantial confusion between the K_L^0 and the n; with the K_L^0 identified as a n in 16.8% of events, and n as a K_L^0 10.8% of the time. The misidentifications observed between various particles, neutral as well as charged, can be reduced by adding complexity to the PID models and training with larger data sets.

5.2 Feature Importance

To analyze the PIDs made by the NN models, **Sh**apley **A**dditive ex**P**lanations (SHAP) [20] is employed to assess the importance of each feature used in training. SHAP is derived from Shapley values from cooperative game theory and is a method to measure the average contribution of a given feature across the entire feature space. A SHAP value is computed for each feature for a given classification by considering possible permutations of features and then taking the average of all marginal contributions by a feature to the resultant prediction. In our case with nonlinear NN models with large data sets, this process is computationally expensive, so `random.choice` from `NumPy` [21] is used to randomly sample 1,000 hypotheses for each particle from our test sample. Since the average marginal contribution of a given feature can be positive or negative, the absolute value of the SHAP values was taken. The magnitude of the SHAP values for positively charged particles is shown in Figure 6, negatively charged particles in Figure 7, and neutral particles in Figure 8.

For p in Figure 6, it can be seen that the most imporant feature is `dEdxCDC` by over an order of magnitude. This result is unsurprising as p has traditionally been identified using this feature in previous GlueX PID studies. Additionally, `pathLengthSc` is another pivotal variable in the p NN PID, which is used to obtain the timing information used in manual PID. Thus, this dependency is

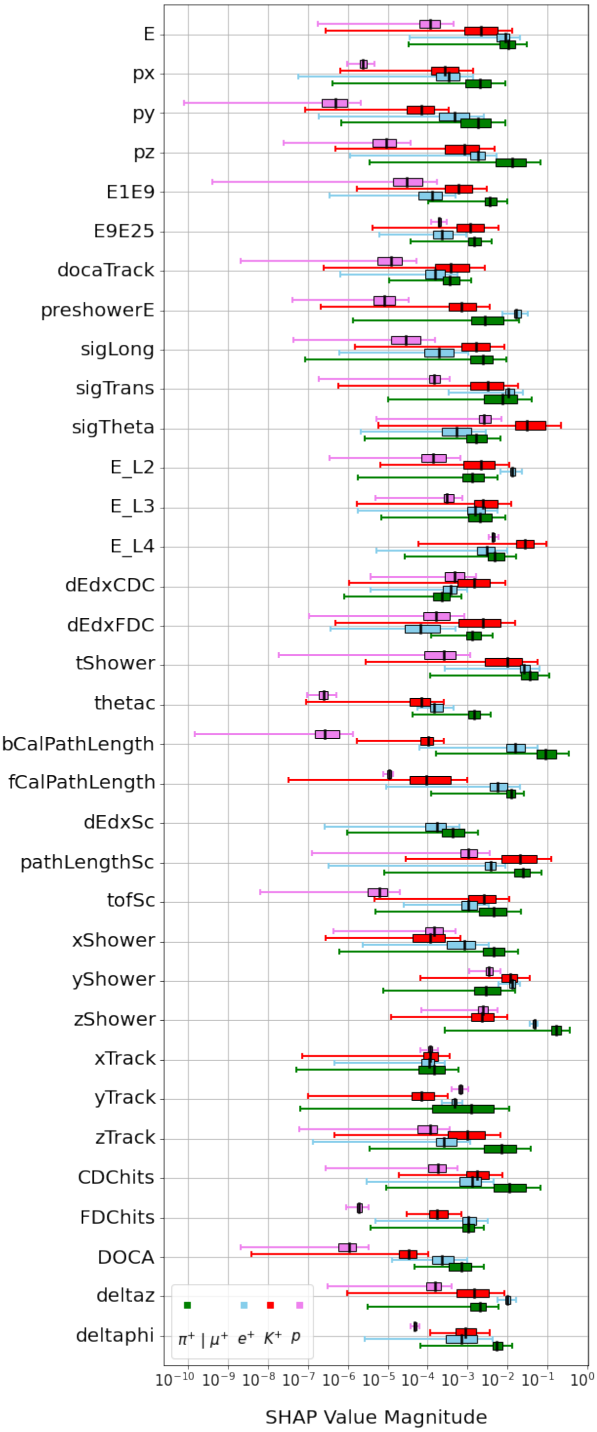

Figure 6: The absolute value of the SHAP values is denoted as the SHAP value magnitude. 10^3 hypotheses from the test sample are randomly sampled for each particle type, and the SHAP value [20] is calculated for each feature. Here, only the $q = +1$ particles from the charged particle NN model are shown; p in pink, K^+ in red, e^+ in light blue and $\pi^+ | \mu^+$ in green. The black line in each box plot represents the mean SHAP value in a given box plot. The upper and lower end of a given box plot represent the 75th and 25th percentile of the data, respectively. Each of the feature labels used to train our charged NN model from Table 2 is shown on the x-axis unless a SHAP value of zero is calculated, and then these features are omitted.

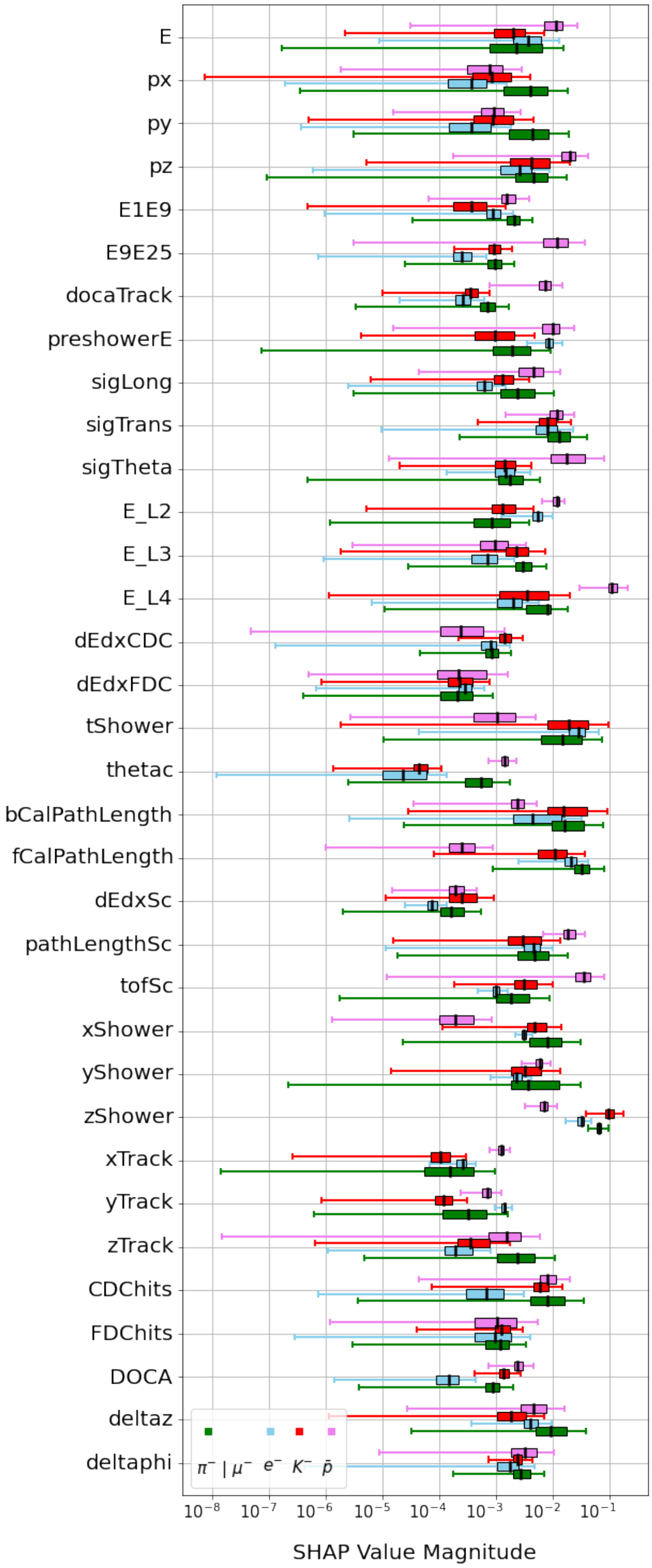

Figure 7: The same as Figure 6, except the SHAP value magnitudes for particles with $q = -1$ are shown. $\bar{p}$ is displayed in pink, K^- in red, e^- in light blue and $\pi^- \mid \mu^-$ in green.

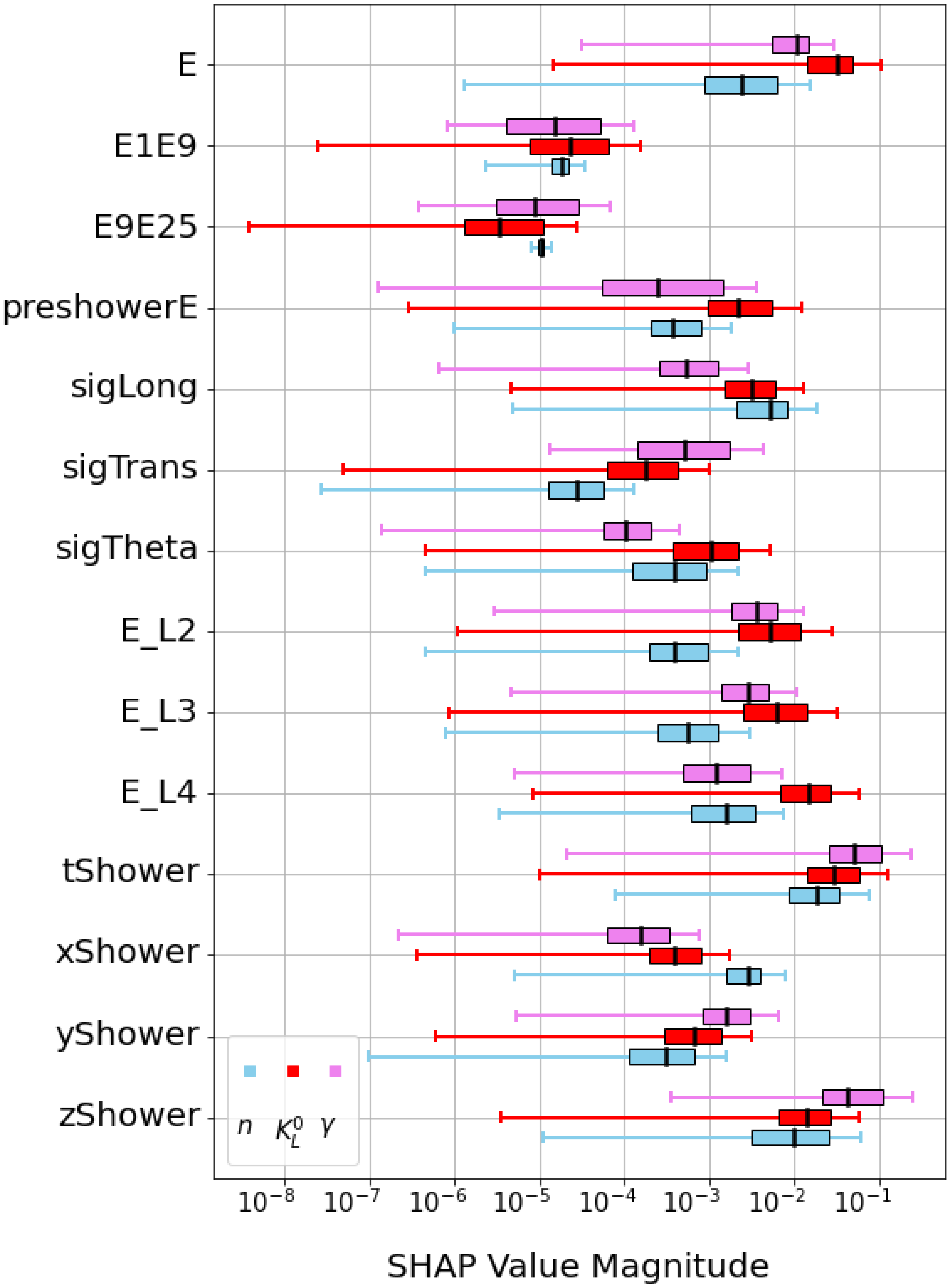

Figure 8: The same as Figure 6 and 7, except the SHAP value magnitudes for particles included in our neutral particle NN model are shown. γ is displayed in pink, K_L^0 in red and n in light blue.

also unsurprising. Variables such as `E_L2`, `E_L3`, `E_L4`, `yShower` and `zShower` each proved to be substantially beneficial in the classification of p despite not being used in manual PID. This is in comparison to each of the momenta components, which are shown to be less crucial for identifying p, although necessary to perform manual PID. It is also important to highlight that the data outside of the interquartile range of a given box plot tends to be skewed toward a minimum value. This is due to an occasional ambiguous measurement or implementation of an overflow value for a given event, which causes the models to not rely heavily on this feature for a given classification. Such a case yields very low SHAP values outside of the interquartile range, giving most features in the feature space tails toward their minimum values. Conversely, features such as `docaTrack` or `FDChits` have very constrained minimum and maximum values for the respective box plots, signifying that the models can consistently rely on these variables due to ubiquity in the data set and this particular feature having a characteristic distribution for a given particle that aids in PID. For p, Figure 6 demonstrates that nearly all features can be used for a given classification, which suggests that the PID process for our NN PID model is dynamic and nonlinear.

As for K^+, e^+ and $\pi^+|\mu^+$ in Figure 6, the noteworthy features that were used for manual PID that also have large SHAP values are `dEdxCDC` and `pathLengthSc` for K^+, `tShower` for e^+, and `tShower` as well as `pathLengthSc` for $\pi^+|\mu^+$. In contrast, each of the positive particles also rely heavily on features such as `E_L2` and `zShower` to make classifications, which are features that are generally not included in manual PID for the GlueX experiment. In the case of `zShower`, the charged NN model is most likely using the distance of an event traveled along the beam path to distinguish between a BCAL or FCAL shower. Although `zShower` has a high mean SHAP value for these three positively charged particles, all features have SHAP values within two or three orders of magnitude of each other. This suggests that in order to achieve high PID accuracies, the entire feature space should be analyzed and synthesized together. Many of the same conclusions can be drawn for Figure 7, in which the SHAP value magnitudes are displayed for the negatively charged particles. $\bar{p}$ follows similar trends as p, and the other three negatively charged particles follow the same trends for a given feature as their positively charged counterparts.

It should be noted that in Figures 6 and 7, the feature labels `q`, `dEdxTOF`, `tofTOF` and `pathLengthTOF` are omitted because all calculated SHAP values yield a value of zero. For `dEdxTOF`, `tofTOF`, and `pathLengthTOF`, this is the case because there are no recorded values in the simulation data set. Thus, all values for these features were replaced by the Overflow Values in Table 2, hence, the model could not use these features. Contrarily, `q` was available in all events, but given that the data set was separated for training into a charged and neutral data set, the capacity of `q` to aid in classification was significantly reduced. The absence of `q` from the Figures 6 and 7 reveals that charge is not an essential feature when distinguishing the positive and negative counterparts of a given particle in the GlueX detector after reconstruction.

Figure 8 shows the SHAP values for neutral particles. The most obvious property of the neutral SHAP value plot is the large number of missing features; as in the charged model, the omitted features have SHAP values of zero. All these features, except for `q`, have no detections in our simulation test data set. Thus, they always default to the Overflow Value shown in Table 2 and are not crucial features for neutral PID. In the case of `q`, all particles have no charge. Thus, this feature is not instructive for the neutral NN model.

Similarly to the SHAP values for charged particles, many of the features for the neutral particles have long whiskers extending from the 25th percentile to the minimum SHAP value magnitude. Again, this is due to sparsity of features in the data set for a given event. The only features that do not have SHAP magnitudes skewed to low values are `E1E9` and `E9E25` for n; despite this, these features have lower average SHAP magnitudes. It can also be seen that `E`, `tShower`, and `zShower` are the features that contribute the most to neutral particle classification. However, in the same manner as the charged NN model, the neutral NN model uses all available data to make classifications. Even with a substantial portion of the simulation data set not having measurements, the neutral NN model proved to be effective, which suggests that ML should be used in future GlueX PID studies.

6 Conclusion

This work compares traditional manual PID methods from the GlueX experiment with NN PID models that are trained on MC simulation data. The GlueX manual PID methods are adapted to best classify charged particles using the training simulation data set, while charged and neutral NN models are trained using the same data set. It is found that the NN models outperformed the standard manual PID methods used by the GlueX experiment for all charged particles and successfully conducted neutral particle PID, which has been a long-standing challenge for the GlueX experiment. In addition to higher PID accuracies, the NN models allow for an unbiased prediction-based PID to be performed that includes a holistic interpretation of all quantities obtained after reconstruction. The NN models also allow feature importance to be conducted, which gives insight into how the NN models make particle classifications and may provide critical insight into underlying physics when applied to experimental data.

These results underscore the significant potential of ML to enhance PID in the analysis of GlueX experimental data. Despite the potential biases and inflated PID accuracies invoked by the simulation data set used in this study, the potential of ML to contribute to PID is evident. The use of higher energy ($1 - 12\ GeV$) simulation data, training with hit-based data rather than reconstructed data, and utilizing experimental data are all potential pathways to further examine the capabilities of ML at PID and bolster the success of the GlueX experiment.

Acknowledgments

This is the most common positions for acknowledgments. A macro is available to maintain the same layout and spelling of the heading.

References

[1] M. Gell-Mann, *A schematic model of baryons and mesons*, *Physics Letters* **8** (1964) 214.

[2] NPLQCD Collaboration collaboration, *First lattice QCD study of the gluonic structure of light nuclei*, *Phys. Rev. D* **96** (2017) 094512.

[3] N. Isgur and J. Paton, *Flux-tube model for hadrons in QCD*, *Physical Review D* **31** (1985) 2910.

[4] S. Adhikari, C.S. Akondi, H. Al Ghoul, A. Ali, M. Amaryan, E.G. Anassontzis et al., *The GLUEX beamline and detector*, *Nuclear Instruments and Methods in Physics Research A* **987** (2021) 164807 [`2005.14272`].

[5] F. Barbosa, J. Bessuille, E. Chudakov, R. Dzhygadlo, C. Fanelli, J. Frye et al., *The GlueX DIRC detector*, *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **876** (2017) 69.

[6] L. Alzubaidi, J. Zhang, A. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma et al., *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, *Journal of Big Data* **8** (2021) .

[7] R. Barsotti and M.R. Shepherd, *Using machine learning to separate hadronic and electromagnetic interactions in the GlueX forward calorimeter*, *Journal of Instrumentation* **15** (2020) P05021 [`2002.09530`].

[8] GEANT4 Collaboration, "GEANT4: A Simulation Toolkit for the Passage of Particles through Matter." Astrophysics Source Code Library, record ascl:1010.079, Oct, 2010.

[9] J. Allison, K. Amako, J. Apostolakis, P. Arce, M. Asai, T. Aso et al., *Recent developments in GEANT4*, *Nuclear Instruments and Methods in Physics Research A* **835** (2016) 186.

[10] R. Brun, F. Bruyant, M. Maire, A.C. McPherson and P. Zanarini, *GEANT3*, 1987.

[11] D.R. Cox, *The Regression Analysis of Binary Sequences*, *Journal of the Royal Statistical Society: Series B (Methodological)* **20** (1958) 215 [`https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1958.tb00292.x`].

[12] Z. Shangnan and Y. Wang, *Quantum Cross Entropy and Maximum Likelihood Principle*, *arXiv e-prints* (2021) arXiv:2102.11887 [`2102.11887`].

[13] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, *arXiv e-prints* (2014) arXiv:1412.6980 [`1412.6980`].

[14] J. Duchi, E. Hazan and Y. Singer, *Adaptive Subgradient Methods for Online Learning and Stochastic Optimization*, *J. Mach. Learn. Res.* **12** (2011) 2121–2159.

[15] G. Hinton, "Neural Networks for Machine Learning, Lecture 6a Overview of mini-batch gradient descent."

[16] R.H.R. Hahnloser, R. Sarpeshkar, M.A. Mahowald, R.J. Douglas and H.S. Seung, *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*, *nat* **405** (2000) 947.

[17] A.F. Agarap, *Deep Learning using Rectified Linear Units (ReLU)*, *arXiv e-prints* (2018) arXiv:1803.08375 [`1803.08375`].

[18] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh and A. Talwalkar, *Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization*, *arXiv e-prints* (2016) arXiv:1603.06560 [`1603.06560`].

[19] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau et al., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, *Nature Methods* **17** (2020) 261.

[20] S. Lundberg and S.-I. Lee, *A Unified Approach to Interpreting Model Predictions*, *arXiv e-prints* (2017) arXiv:1705.07874 [`1705.07874`].

[21] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau et al., *Array programming with NumPy*, *Nature* **585** (2020) 357.