Prepared for submission to JINST

# Particle identification in the GlueX detector using a neural network

**E. Habjan**[a,b] **R. Dube**[a] **R. Jones**[a]

[a] *University of Connecticut, Department of Physics, 196A Auditorium Road, Unit 3046, Storrs, CT 06269, USA*

[b] *Northeastern University, Department of Physics, 100 Forsyth Street 111, Boston, MA 02115, USA*

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

Abstract: Accurate Particle Identification (PID) is a crucial element for successful reconstruction of interactions measured in particle physics experiments. In the GlueX experiment at Jefferson Laboratory, PID is achieved by making cuts on the kinematic properties of tracks and showers reconstructed from hits in the detector. However, in this work we seek to improve upon these simple cuts-based procedures using machine learning with neural networks. The promise of this approach is the ability to exploit hidden correlations between PID variables in the reconstructed kinematics data. We demonstrate that both charged and neutral particles can be identified in simulated GlueX events with significantly improved accuracy using a neural network.

Keywords: Particle identification methods, Analysis and statistical methods

## Contents

## 1 Introduction

A long-standing question in modern physics is concerning the nature of the quark confinement mechanism of quarks and gluons in quantum chromodynamics (QCD). To begin unraveling this mystery, the GlueX experiment in Hall D at Jefferson Laboratory aims to better understand the quark and gluonic degrees of freedom that are predicted by QCD through gluonic excitations of hybrid mesons originating from $\gamma p$ collisions [1, 2]. A comprehensive study of this spectrum of hybrid mesons requires accurate determination of the final hadronic states through good particle identificaiton (PID). The PID capabilities of the GlueX detector is enabled by four detector systems: a forward time-of-flight wall (TOF), a barrel calorimeter with a cylindrical geometry (BCAL), a forward lead-glass calorimenter with a planar geometry (FCAL), a thin scintillator start counter (SC), and a Čerenkov detector [3, 4]. GlueX uses time-of-flight (TOF and BCAL), forward going tracks (Čerenkov detector), and dE/ds (CDC) information to classify particles using a likelihood-based PID system after reconstruction and kinematic fitting.

In this work, the aim is to demonstrate that PID accuracies for GlueX analyses can be improved through the use of machine learning (ML) [5]. The advent of ML has led to innumerable studies in a wide range of fields, including a GlueX study [6] that used a discriminating neural network (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 simulation data. The PID accuracies obtained from these NN models are directly compared to 'manual' PID methods using the same simulation dataset. This work exhibits that ML models can be a more effective alternative to the contemporary methods of PID.

## 2 Monte Carlo Simulation Dataset

The training and test data are extracted from the low-momentum GlueX particle gun simulations. In these simulations, a particle is spawned at a random location within the target and fired in a random direction with a random magnitude of momentum under 1 GeV/c. The interactions between the particle and the detector (along with any decays that may occur) are handled by Geant4 [7, 8], with the simulated detector hits being stored in a Hall D Data Model (HDDM) format. These data are then reconstructed using the `halld_recon` package to identify showers and tracks, which significantly decreases the number of features needed to describe each event. Finally, these data are converted from the hierarchical HDDM format to a tabular format that can be used in either manual or NN PID. The labels of each quantity in our final dataset are shown in Table 1.

Although the particle gun simulations allow for easy event labeling, decays and other interactions in the detector may produce tracks or showers that are not produced directly by the generated particle. To exclude events in which the generated particle decayed before interacting with the detector, we remove any event that has more than one vertex in the first 500 seconds of the simulation, as indicated by the truth information of the Monte Carlo simulation. Note that the initial spawning of the generated particle is counted as a vertex. To eliminate events where interactions with the detector produced secondary tracks or showers, cuts were placed on the number of tracks and showers per event. For events with a charged generated particle, the event was only included in the training and test samples if the reconstructed event contained exactly one track and one shower, and the shower must be associated with the track. Events with neutral generated particles must have exactly one shower and no tracks. These cuts are necessary to ensure the event label matches the particle that produced the shower or track that is included in the training or test dataset, though it may inflate the accuracy of PID techniques due to the exclusion of complicated interactions with the detector.

The training dataset consists of 80,000 events per particle type. For events with a charged generated particle, only the track hypothesis that matches the generated particle type is included in the training dataset. For events with muons, only the pion track hypothesis was added to the training dataset, as there is no muon hypothesis in the default reconstruction. This results in a training dataset in which each row of the dataset represents a different event. In contrast, a row corresponding to each hypothesis is added to the test dataset for events with charged generated particles; the event number is identified by the `group` label, which only appears in the test dataset. For neutral particles, there is only one row in the test dataset per event, as no hypotheses are used in the shower reconstruction process. The test dataset contains 40,000 events per particle type, though the number of rows is substantially larger due to the inclusion of multiple hypotheses per event for charged generated particles.

**Table 1**: Feature labels of the particle gun dataset.

| Column | Unit | Description | Overflow Value |
|---|---|---|---|
| `true ptype` | | The true generated particle type (Geant3 coding) | |
| `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/ds of track in the CDC | -5 |
| `dEdxFDC` | $keV/cm$ | Average dE/ds 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/ds 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/ds 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 | |

## 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 [9] and the creation of the idea of cross-entropy in the early years of information theory has evolved into a loss function that is ubiquitous in machine learning: the cross entropy loss function. In general, the minimization of cross entropy between two distributions is equivalent to the maximization of the log likelihood [10]. 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 $x_i$ is a given detection (or in the case of this work, a particle) and $\theta$ defines our parameter space (e.g., energy loss, momenta). By maximizing the log likelihood, we can best predict the probability of detecting a given $x_i$ when provided with $\theta$. The cross entropy $H(P_D(x), P_\theta(x))$ is also defined in terms of the probability of $x_i$ and $\theta$:

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

Thus, we see that maximizing the log likelihood is equivalent to minimizing the cross entropy, known as the cross entropy and maximum likelihood principle. In our NN models, the `Tensorflow` implementation of the cross entropy loss is used as the loss function during training.

### 3.2 Adam Optimizer

The Adam optimizer [11] is a method to efficiently optimize the subfunctions that make up the entire objective function by taking gradient steps with respect to each individual subfunction. This process also describes Stochastic Gradient Descent, however, the Adam optimizer is particularly designed to optimize parameters for stochastic subfunctions in high dimensional space, while only requiring first-order gradients. The large level of stochasticity in measured quanities within particle colliders makes the Adam optimizer an ideal choice for minimizing the cross entropy loss function.

### 3.3 Structure and Activation Functions

The structure of the NN is comprised of an input layer, one or multiple 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 1. In each of the hidden layers of the NN are make use of the Rectified Linear unit (ReLu) activation function [12, 13]. The ReLu activation function is defined as:

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

so that 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 as:

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

With an input $x$, the output $S(x)$ will always be between 0 and 1. Using the sigmoid activation function as the output of the NN models allows for a confidence-based prediction to made when classifying particles.

### 3.4 Hyperband

In order to minimize the Cross Entropy Loss function, an optimization process is carried out for 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 optimized values of each of these hyperparameters is found using `Hyperband` [14]. This choice to employ `Hyperband` as the optimization algorithm is made as it is more computationally efficient and performs better than Bayesian optimization [14]. The `Hyperband` algorithm selects a different set of hyperparameters and trains the neural network for a fixed number of epochs. A method known as *successive halving* is invoked, in which only half of the models with the largest Cross Entropy loss are allocated resources to continue training after the fixed number of epochs have passed. This procedure is repeated until only a single set of hyperparmaters remains; these hyperparmeters are then used to train the NN models.

## 4 Methods

In this section we describe our manual PID cuts and the training process for our neural network models used for PID.

### 4.1 Manual PID

In this work, we identify pions (~139.6 MeV/c$^2$) and muons (~105.7 MeV/c$^2$) as the same particle, denoted as $\pi^+ \mid \mu^+$ or $\pi^- \mid \mu^-$ for the positive and negative counterparts, respectively. This simplification is made for our PID methods in this paper since pions and muons have similar masses and the GlueX detector does not have a hadronic calorimeter, which makes discerning these particles particlularly difficult. Muons and Pions can be distinguished by investigating the momentum distributions of a given event in the FCAL, however this must be done prior to reconstruction and is outside the scope of this work.

The timing cuts implemented in this work make use of the Spring 2017 Analysis Launch Cuts [1]; each of these cuts are shown in Table 2. The measured BCAL and FCAL times are recorded as a single variable, `tShower`, in the simulation dataset, 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 BCAK (`tFlightBCAL`) or the FCAL (`tFlightFCAL`). In order to assess the quality of a given hypothesis, a chi-squared value is calculated between the mean shower time and calculated shower times. Only hypotheses with a chi-squared value of less than 0.075 are considered robust; any hypotheses that are above this threshold are labeled as no identification (no ID). There is only timing information for charged particles in the simulation dataset, thus no timing cuts can be made for the neutral particles.

In addition to timing cuts, we also implement track energy loss cuts using the `dEdxCDC` variable and the magnitude of the particle momentum. To create a decision boundary between each particle, we use the same functional form of the equations used in the Spring 2017 Analysis Launch Cuts:

$$dE/ds = e^{a \cdot p + b} + c, \tag{4.1}$$

where $p$ is the momentum of a particle in units of $GeV/c$, $dE/ds$ is the energy loss in the CDC in units of $KeV/cm$ and $e$ is Euler's number. $a$, $b$ and $c$ are constants that are varied in order to best classify each particle. Using the training dataset, the number of incorrectly identified particles is minimized by making each of these constants free parameters and utilizing the `minimize` method from the `scipy.optimize` [15] module. The dE/ds – p decision boundaries are only derived for charged particles as there is not sufficient data available for neutral particles in the simulation dataset. The optimized decision boundaries are shown here:

$$dE/ds_1 = e^{-5.095 \cdot p - 10.205} + (2.080 \cdot 10^{-6}), \tag{4.2}$$

$$dE/ds_2 = e^{-3.947 \cdot p - 12.284} + (1.936 \cdot 10^{-6}), \tag{4.3}$$

$$dE/ds_3 = e^{-0.185 \cdot p + -19.215} + (2.190 \cdot 10^{-6}). \tag{4.4}$$

Each optimized decision boundary is overlaid on the test dataset in Figure 1. 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 matches the predicted hypothesis from our manual PID. Each of these conditions for manual PID are shown in Table 2. A PID is made for every hypothesis in our test dataset that meets the chi-squared criteria, however, if a given event meets none of the cuts made, then no ID is designated. Furthermore, if an event has two or more PIDs that match different hypotheses in our test dataset, then the particle type with the highest chi-squared value is designated. The confusion matrix presenting the results of our manual PID on charged particles is shown in Figure 2a.

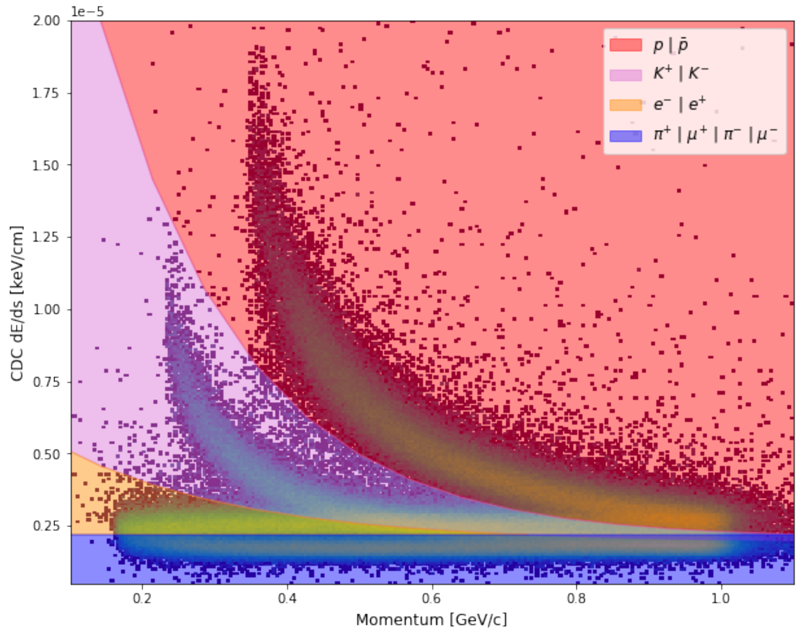
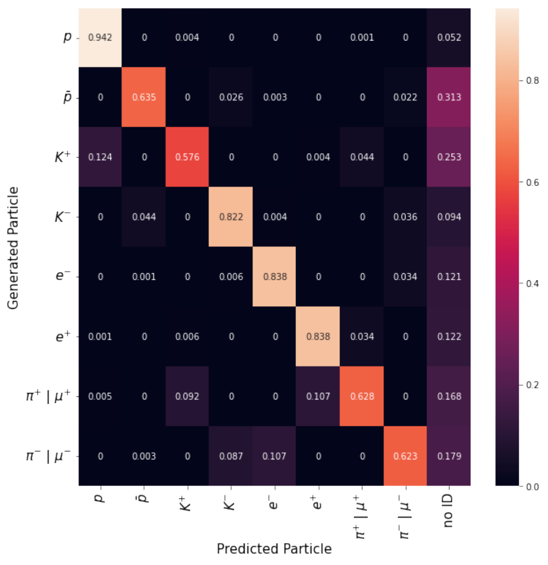


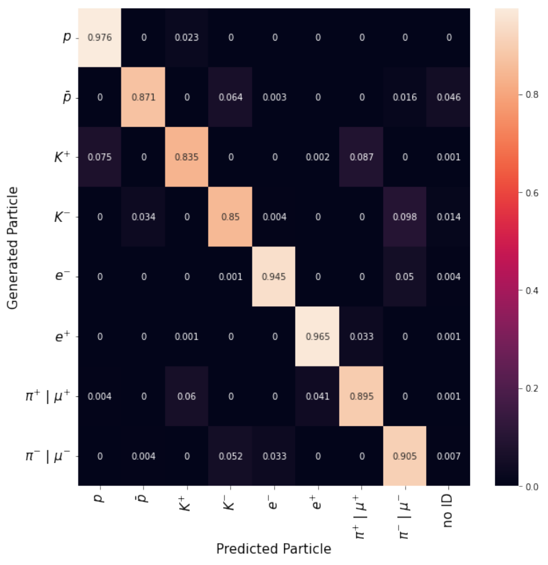
**Figure 1**: A 2-Dimensional histogram of the average track energy loss in the CDC plotted against the magnitude of momentum vectors from our test dataset. The manual PID cuts described in Section 4.1 are overlaid to show the classification boundaries; the functional form of each decision boundary is shown in Equations 4.2 – 4.4. Regions of the plot shaded in red are classified as $p$ or $\bar{p}$, purple as $K^{+|-}$, yellow as $e^{-|+}$, and blue as $\pi^{+|-}$ or $\mu^{+|-}$.

**Table 2**: Manual PID cuts. If an entry is missing, there is no cut for that particle. The $dE/ds_1$, $dE/ds_2$ and $dE/ds_3$ cuts corresponded to Equations 4.2, 4.3 and 4.4, respectively.

| Particle | $\Delta t$ BCAL [ns] | $\Delta t$ FCAL [ns] | dE/ds [keV/cm] | E/p [$c$] |
|---|---|---|---|---|
| $e^+$ | ± 1.0 | ± 2.0 | $dE/ds_3 < dE/ds$ & $dE/ds_2 > dE/ds$ | E/p > 0.83 |
| $e^-$ | ± 1.0 | ± 2.0 | $dE/ds_3 < dE/ds$ & $dE/ds_2 > dE/ds$ | E/p > 0.83 |
| $\mu^+\vert\pi^+$ | ± 1.0 | ± 2.0 | $dE/ds_3 > dE/ds$ | E/p < 0.83 |
| $\mu^-\vert\pi^-$ | ± 1.0 | ± 2.0 | $dE/ds_3 > dE/ds$ | E/p < 0.83 |
| $K^+$ | ± 0.75 | ± 2.5 | $dE/ds_2 < dE/ds$ & $dE/ds_1 > dE/ds$ | |
| $K^-$ | ± 0.75 | ± 2.5 | $dE/ds_2 < dE/ds$ & $dE/ds_1 > dE/ds$ | |
| $p$ | ± 1.0 | ± 2.0 | $dE/ds_1 < dE/ds$ | |
| $\bar{p}$ | ± 1.0 | ± 2.0 | $dE/ds_1 < dE/ds$ | |



(a) Manual PID



(b) NN PID

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

### 4.2 Neural Network PID

Prior to training the NN models, some data pre-processing steps were taken, which includes spliting the test and training datasets into charged and neutral datasets. Additionally, `Tensorflow` requires a real number input for training, thus any values in our datasets that do not have a detection are replaced by the 'Overflow Value' seen in Table 1. Lastly, only the features in Table 1 that have an Overflow Value are used to train the NN models.

The NN models make use of the `TensorFlow` implementations of the Cross Entropy Loss Function, Adam Optimizer and ReLu activation function, all of which are 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`. Variation for each hyperparameter is allowed between 1 and 6 hidden layers, 100 and 600 neurons per hidden layer and $10^{-4}$ and $10^{-2}$ for the learning rate. The optimized hyperparameters are used to train a NN model for a maximum of 50 epochs and training is stopped if the Cross Entropy Loss changes by less than 0.01 after 5 successive epochs.

The trained NN models are used to make a classification on every hypothesis in the test dataset. Each prediction made by the `predict` method from the `Tensorflow` models yield a confidence value for each possible classification (i.e. particle). The highest confidence value across all hypotheses in an event is taken as the PID. A no ID label is given for any PID that has a confidence of less than 0.4. In the same way as Section 4.1, positive pions and muons and negative pions

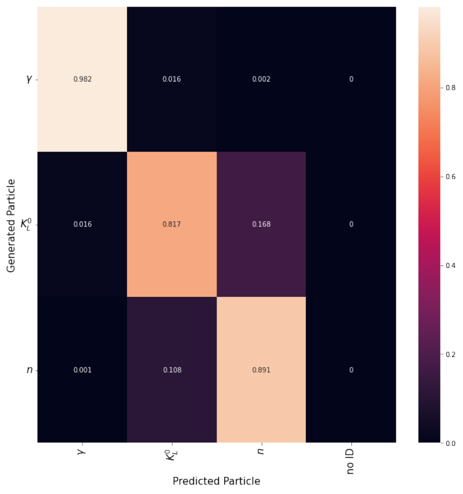


**Figure 3**: The confusion matrix for the NN PID for neutral particles. The generated particle is shown on the y-axis and the particle classified by the neutral NN model is shown on the x-axis. Particles that do not meet the confidence criteria discussed in Section 4.2 are classified with a no ID label.

and muons are classified as the same particles. The confusion matrix for the charged NN model is shown in Figure 2b and the confusion matrix for the neutral NN model is shown in Figure 3.

## 5 Results

In this section the results are presented for the traditional PID cuts and NN PID. A direct comparison between these two methods is made and the advantages of NNs in PID are discussed. The importance of each feature in the simulation dataset is determined to better understand how the NN models are making predictions.

### 5.1 Comparing PID techniques

In Figure 2a, the $p$ sample is correctly identified with an accuracy of 0.942 for the manual PID method, with only 0.052 of samples being allocated to the no ID column of the confusion matrix. The $p$ is the particle with the highest accuracy, while the $K^-$, $e^-$ and $e^+$ have PID accuracies in the range 0.82 – 0.84. The lower end of classification accuracies are $\bar{p}$, $K^+$, $\pi^+ \mid \mu^+$ and $\pi^- \mid \mu^-$ in the range 0.57 – 0.64. Many of the samples that are not correctly identified by the manual PID method are designated as no ID. The most prominent exceptions of this are the $K^+$, as 0.124 of the sample are misidentified as $p$ while 0.092 and 0.107 of the $\pi^+ \mid \mu^+$ sample are identified as $K^+$ and $e^+$, respectively. For $\pi^- \mid \mu^-$, 0.087 and 0.107 were identified as $K^-$ and $e^-$, respectively. In addition to these misclassifications, there are several cases in which the manual PID method misclassified samples more infrequently ($< 0.05$). The large number of events that do not pass the chi-squared timing cut and the instances of substantial particle misidentication discussed above highlight the areas of improvement for the manual PID method.

In Figure 2b, the confusion matrix of the charged NN model is presented. The $p$, $e^-$ and $e^+$ are correctly identified in the range 0.94 – 0.98 and the five other charged particles have accuracies in the range 0.83 – 0.91. It is found that for all particles the PID accuracy is better for the charged NN PID method in comparison to the manual PID method. $p$ and $K^-$ have an improvement of ~0.03, $e^-$ and $e^+$ have an improvement of ~0.1, and the four other particles have substantial improvements of more than 0.2. In addition to substantial improvements in PID accuracy, there are decreases in the misclassification of $K^+$ as $p$ (0.075), $\pi^+ \mid \mu^+$ as $K^+$ (0.06) and $e^+$ (0.041), as well as $\pi^- \mid \mu^-$ as $K^-$ (0.052) and $e^-$ (0.033). Despite the decreases in these particular cases, there are increases in misclassification for $\bar{p}$ as $K^-$ (0.064), $K^+$ as $\pi^+ \mid \mu^+$ (0.087), and $K^-$ as $\pi^- \mid \mu^-$ (0.098). The increase in misidentification for these specific instances are the lone shortcoming of the NN PID method in contrast to the manual PID method.

The classification scheme for the NN PID discussed in 4.2 outlines that the particle prediction with the highest confidence for a given event is chosen as the PID for that event. This is in contrast to manual PID, in which the hypothesis with the minimum timing chi-squared value is chosen. For some particles, (e.g. $e^-$) the timing information is crucial, however other particles (e.g. $p$) there are quantities that play a more significant role in PID. For this reason, using the minimum timing chi-squared as a metric for the strength of a hypothesis hinders the PID accuracy of particular particles. However, for NN PID, the sigmoid output function gives a confidence interval between 0 and 1 determined by all of the input features of the dataset, thus giving a holistic analysis of the recorded quantities in a given event.

In Figure 3, the confusion matrix for the neutral particle NN PID model is shown. Unlike the charged particles, there does not exist robust manual PID methods for classifying neutral particles in the GlueX experiment. The lone exception is the existence of timing cuts for $\gamma$, though the simulation dataset did not recover any predicted timing values (i.e. `tFlightBCAL` and `tFlightFCAL`), thus no manual PID could be carried out for neutral particles. However, it is demonstrated that neutral particles can be identified with high accuracy using NN PID methods; an accuracy of 0.982 is achieved for $\gamma$, 0.817 for $K_L^0$ and 0.891 for $n$. The model used to achieve these classification accuracies is optimized by the `HyperBand` algorithm, which found optimal validation accuracy for a NN model that had 3 hidden layers with a total of $\sim 2 \cdot 10^5$ parameters. This is compared to the single hidden layer and $\sim 4 \cdot 10^4$ parameters of the charged NN model. This near order of magnitude increase in complexity for the neutral NN PID model demonstrates the difficulty of neutral PID in the GlueX detector, though it is shown to be 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 0.168 events, and $n$ as a $K_L^0$ in 0.108 events. The confusion between neutral particles, as well as charged particles, can be minimized simply by training with more data and adding complexity to the models, which can assuredly be achieved by the GlueX collaboration for improved results.

### 5.2 Feature Importance

To analyze the PIDs made by the NN models, SHapley Additive exPlanations (SHAP) [16] is employed to assess the importance of each feature used in training. SHAP is derived from Shapley values from cooperative game theory and are 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

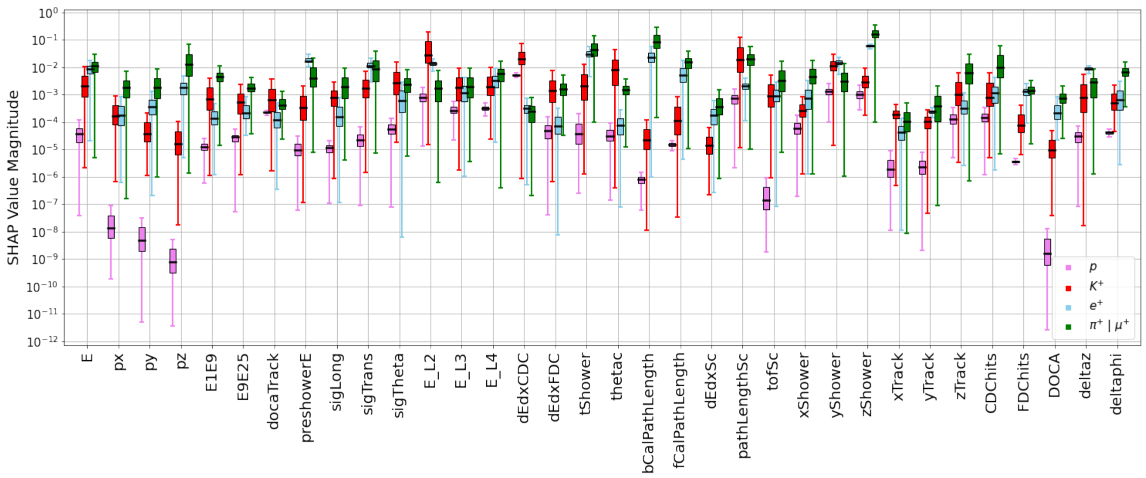


**Figure 4**: The absolute value of the SHAP values is denoted as the SHAP value magnitude. $10^3$ hypotheses from the test sample are randomly sampled for each particle type and the SHAP value [16] is calculated for each feature. Here, only the $q = +1$ particles from the charged particle NN model are shown; $p$ in pink, $K^+$ in red, $e^+$ in light blue and $\pi^+ \mid \mu^+$ 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 represents the 75th and 25th percentile of the data, respectively. Each of the feature labels used to train our charged NN model from Table 1 are shown on the x-axis unless a SHAP value of zero is calculated, then these features are omitted.

NN models with large datasets, this process is very computationally expensive, so `random.choice` from `NumPy` [17] 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 either be positive or negative the absolute value of the SHAP values is taken. The magnitude of the SHAP values for positively charged particles is shown in Figure 4, negatively charged particles in Figure 5, and neutral particles in Figure 6.

For $p$ in Figure 4, it can be seen that the most important 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 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 being necessary to perform manual PID. It is also important to highlight that, in general, the data outside of the interquartile range of a given box plot has a tendency to be skewed towards 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 SHAP values outside of the interquartile range, giving most of the features in the feature space tails towards 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 are able to consistently rely on these variables due to ubiquity

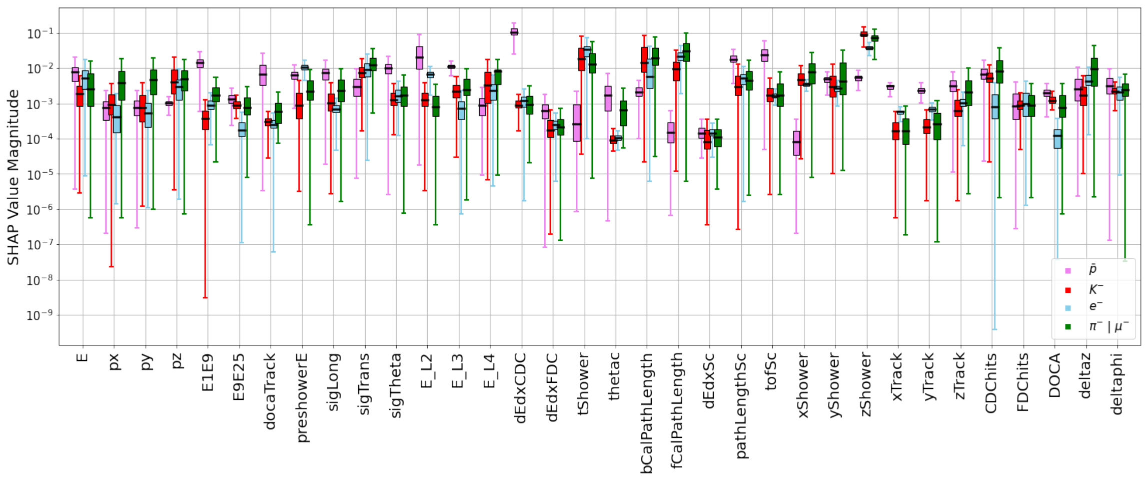


**Figure 5**: The same as Figure 4, except the SHAP value magnitudes for particles with $q = -1$ are shown. $\bar{p}$ is displayed in pink, $K^-$ in red, $e^-$ in light blue and $\pi^- \mid \mu^-$ in green.

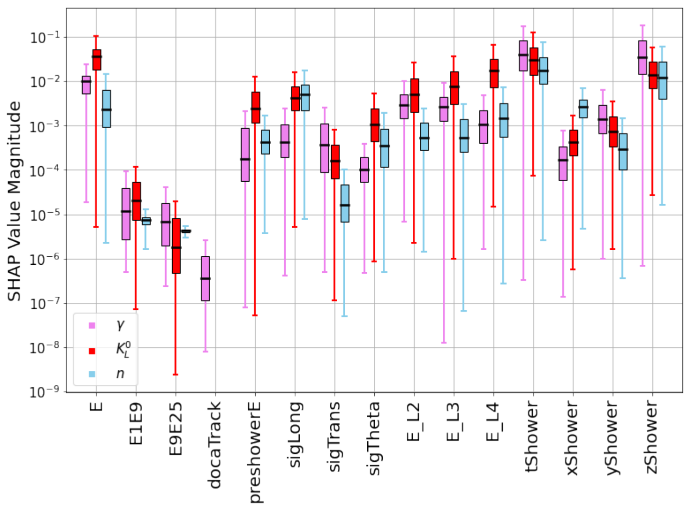


**Figure 6**: The same as Figure 4 and 5, except the SHAP value magnitudes for particles included in our neutral particle NN model are shown. $\gamma$ is displayed in pink, $K_L^0$ in red and $n$ in light blue.

and this particular feature having a characteristic distribution for a given particle. For $p$, Figure 4 demonstrates that nearly all features can be used for a given classification, and that the PID process for our NN PID model is dynamic and nonlinear.

As for $K^+$, $e^+$ and $\pi^+ \mid \mu^+$ in Figure 4, 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 $\pi^+ \mid \mu^+$. 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. Using the results shown in Figure 4, one could design manual PID cuts for one of the features that consistently has large SHAP values (e.g. `zShower`) in order to boost manual PID accuracies. However, for these three positively charged particles, all features have SHAP values within two or three orders of magnitude of each other suggesting 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 5, in which we display the SHAP value magnitudes 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 4 and 5, 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 done simply because there are no recorded values in our simulation dataset and all values for these features were replaced by the Overflow Values in Table 1, thus the model could not make use of these features. Contrarily, `q` was available in all events, but given that the dataset was separated for training into a charged and neutral dataset, the capacity of `q` to aid in classification was significantly reduced. The absence of `q` from the Figures 4 and 5 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.

In Figure 6, the SHAP values for neutral particles is shown. The most obvious property of the neutral SHAP value plot is the large number of features that are missing, and as in the charged model, the omitted features have SHAP values of zero. All these features with the exception of `q` have no detections in our simulation test dataset, thus they are always defaulted to the Overflow Value shown in Table 1 and are not crucial features for neutral PID. In the case of `q`, all particles have a charge of 0, 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 a combination of sparsity and similarity between particles. The only features that do not have SHAP magnitudes skewed to low values are `E1E9` and `E9E25` for $n$, in spite of this these features having 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 makes use of all available data to make classifications. Even with a substantial portion of the simulation dataset not having measurements, the neutral NN model still proved to be effective, suggesting that ML should be used in future GlueX PID studies.

## 6 Conclusion

In this work, the traditional PID methods used in the GlueX detector were compared with NN PID methods. The GlueX manual PID methods were adapted to best classify charged particles using the training simulation dataset while a charged and a neutral NN model were trained using this same dataset. It was found that the NN models outperformed the standard PID methods used by the GlueX collaboration for all charged particles and were able to successfully conduct 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 enable feature importance to be conducted, which gives insight into how the NN models are making these classifications and may give key insight into underlying physics when applied on experimental data.

These results are of course taken with the caveat that the data being used is simulation data, in the low energy regime (0 – 1 GeV) and only the purest (single track and/or single shower) particle gun data is used. Despite the potential biases and inflated PID accuracies invoked by the dataset used in this study, the potential of ML to contribute to PID in the GlueX experiment is made evident. The use of simulation data at higher energies (1 – 12 GeV), hit data rather than reconstructed data, or experimental data are all potential pathways to test the capabilities of ML at PID and to 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] 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].

[2] D.S. Carman, *GlueX: The Search for Gluonic Excitations at Jefferson Laboratory*, in *Hadron Spectroscopy*, A. Reis, C. Göbel, J.D.S. Borges and J. Magnin, eds., vol. 814 of *American Institute of Physics Conference Series*, pp. 173–182, AIP, Feb., 2006, DOI [hep-ex/0511030].

[3] The GlueX Collaboration, A. AlekSejevs, S. Barkanova, M. Dugger, B. Ritchie, I. Senderovich et al., *An initial study of mesons and baryons containing strange quarks with GlueX*, *arXiv e-prints* (2013) arXiv:1305.1523 [1305.1523].

[4] The GlueX Collaboration, *Mapping the spectrum of light quark mesons and gluonic excitations with linearly polarized photons presentation to pac30 – the gluex collaboration*, https://www.jlab.org/exp$_{p}rog/proposals/06/PR12-06-102.pdf$.

[5] 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) .

[6] 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].

[7] Geant4 Collaboration, "Geant4: A Simulation Toolkit for the Passage of Particles through Matter." Astrophysics Source Code Library, record ascl:1010.079, Oct., 2010.

[8] 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.

[9] 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].

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

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

[12] 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*, **405** (2000) 947.

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

[14] 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].

[15] 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.

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

[17] 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.