

Predicting COMPASS/AMBER Acceptance with Neural Networks

Pedro M. P. Curvo

October 21, 2023

1 Introduction

1.1 COMPASS/AMBER

The COMPASS experiment, short for "Common Muon and Proton Apparatus for Structure and Spectroscopy," is a cutting-edge particle physics experiment conducted at CERN, the European Organization for Nuclear Research, in Geneva, Switzerland. COMPASS is a collaborative endeavor involving scientists and researchers from institutions around the world. This experiment is dedicated to unraveling the mysteries of the subatomic world by probing the structure and properties of hadrons, which are particles composed of fundamental constituents called quarks and held together by the strong nuclear force. COMPASS achieves this by utilizing a high-energy polarized muon beam, making it unique in its precision measurements of hadron physics and spin properties. With its sophisticated array of detectors, including spectrometers and tracking chambers, COMPASS captures and analyzes particles produced during high-energy collisions, providing valuable insights into the internal structure of nucleons like protons and neutrons. Additionally, COMPASS explores the production of exotic hadrons, expanding our understanding of the strong force interactions and the fundamental constituents of matter. The data generated by the COMPASS experiment contributes significantly to advancing our knowledge of the building blocks of the universe and the forces that govern them, furthering our understanding of the intricacies of particle physics.

2 Data Origin and Preprocessing

2.1 Data Origin: Generation and Reconstruction

The data used for the analysis presented in this paper was generated using a LEPTO generator, a dedicated generator for deep inelastic scattering (DIS) processes. Beyond that, the full simulation of the COMPASS spectrometer was done using the GEANT4 programme. The data was then reconstructed using the COMPASS reconstruction software. With this process, two files were generated for each type of events (Hadrons and Inclusive), one with the Monte Carlo (MC) data, and another with the reconstructed data. The goal with this data is to check if it is possible to predict the acceptance of the COMPASS/AMBER using a Neural Network, since the GEANT4 simulation is very time consuming, approximately 30 seconds per event. That might not seem like a lot, but when you have millions of events, it becomes a problem. Therefore, one could use a Neural Network to have a better idea of how changed parameters affect the data/MC agreement, and then only run the GEANT4 simulation for the events that are more likely to be accepted.

2.2 Data Variables

In this study, we used two types of events, Inclusive and Hadrons. The Inclusive events are events where the muon is detected, while the Hadrons events are events where hadrons are detected. The data used for the analysis of the Inclusive events has 11 variables:

Variable	Description
X_b	$Bjorken_x$
Y	Virtual Photon Energy Fraction of the initial momentum
Q^2	Four-momentum transfer
$Trig$	Trigger
PV_z	X-Position of interaction vertex in the target
PV_x	Y-Position of interaction vertex in the target
PV_y	Z-Position of interaction vertex in the target
Mom_{mu}	Incoming beam momentum
Mom_{mup}	Outgoing muon momentum
$d_x d_z mup$	Angle of Muon
$d_y d_z mup$	Angle of Muon

While the data used for the analysis of the Hadrons events has 16 variables:

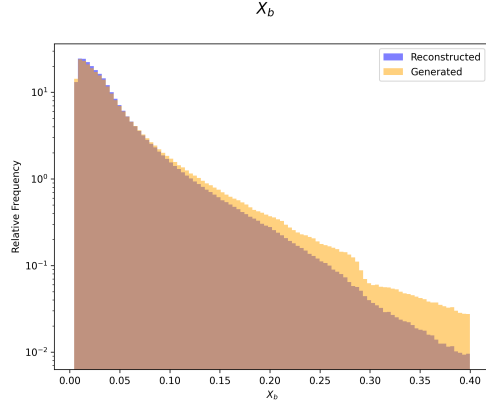
Variable	Description
X_b	$Bjorken_x$
Y	Virtual Photon Energy Fraction of the initial momentum
Z	Fraction of hadron energy to the virtual photon energy
Q^2	Four-momentum transfer
$Trig$	Trigger
PV_z	X-Position of interaction vertex in the target
PV_x	Y-Position of interaction vertex in the target
PV_y	Z-Position of interaction vertex in the target
Mom_{mu}	Incoming beam momentum
Mom_{mup}	Outgoing muon momentum
$d_x d_z mup$	Angle of Muon
$d_y d_z mup$	Angle of Muon
q	Hadron Charge
mom	Momentum of Hadron
$d_x d_z$	Angle of Hadron
$d_y d_z$	Angle of Hadron

As mentioned before, for each type of events, we have a file with the Monte Carlo data and another with the reconstructed data. Since the goal is to find a way to predict the acceptance of the COMPASS/AMBER, we created a new variable, called "Accepted", which is 1 if the event was accepted and 0 if it was rejected. Therefore, for the data with the Monte Carlo information, we added a column with the value 1 for all the events, and for the data with the reconstructed information, we added a column with the value 0. It's important to refer that the number of events for each of the files is different, since the Monte Carlo data has more events than the reconstructed data. Hence, we had to balance the data for the training and testing of the Neural Network, which will be explained in the following sections.

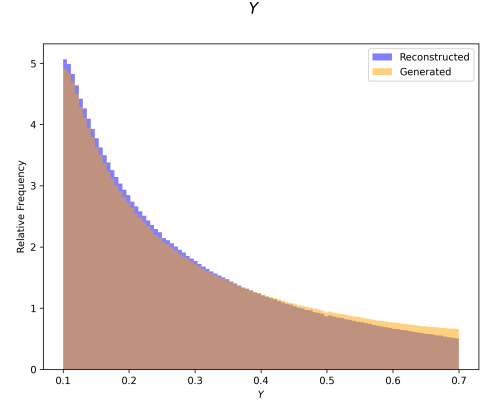
2.3 Data Visualization

In order to have a better understanding of the data, we plotted the histograms for each of the variables, for the Inclusive and Hadrons events.

The histograms for the Inclusive events are shown in the figures below:

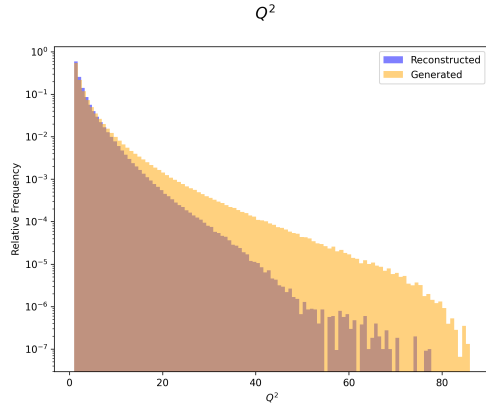


(a) Histogram for X_b

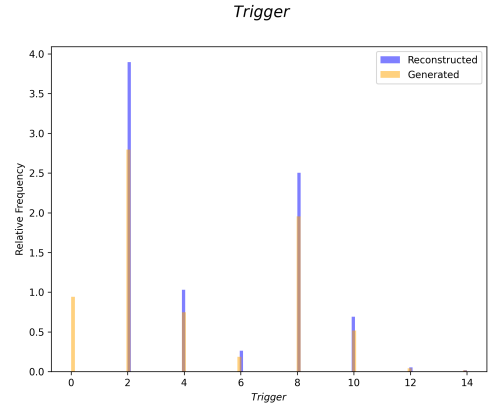


(b) Histogram for Y

Figure 1: Histograms for X_b and Y

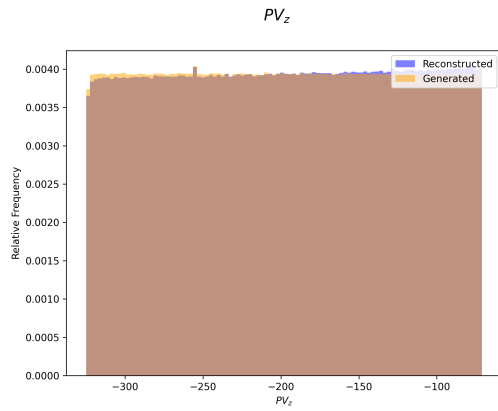


(a) Histogram for Q^2

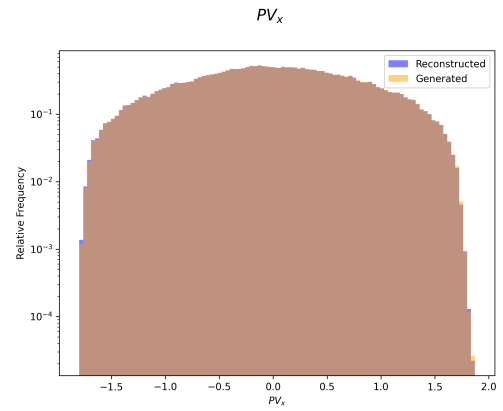


(b) Histogram for *Trigger*

Figure 2: Histograms for Q^2 and *Trig*

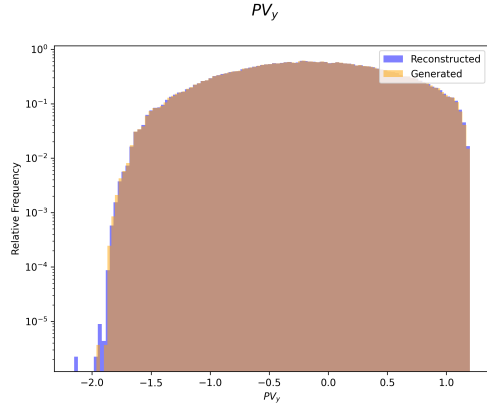


(a) Histogram for PV_z

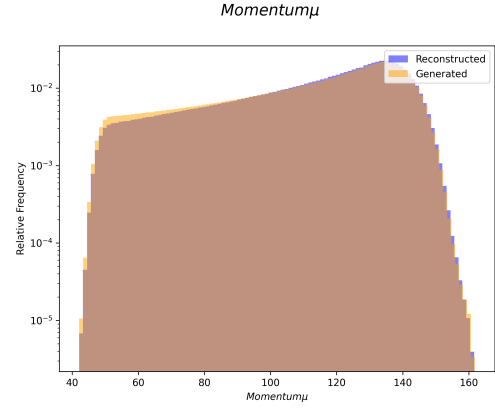


(b) Histogram for PV_x

Figure 3: Histograms for PV_z and PV_x

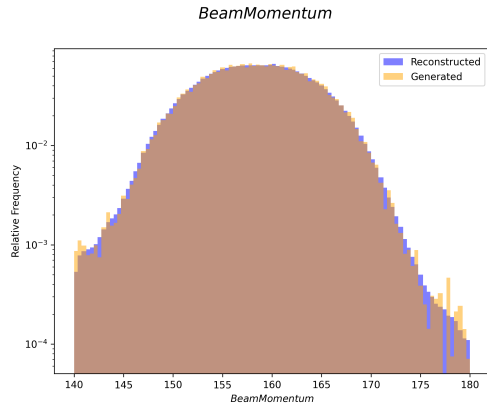


(a) Histogram for PV_y

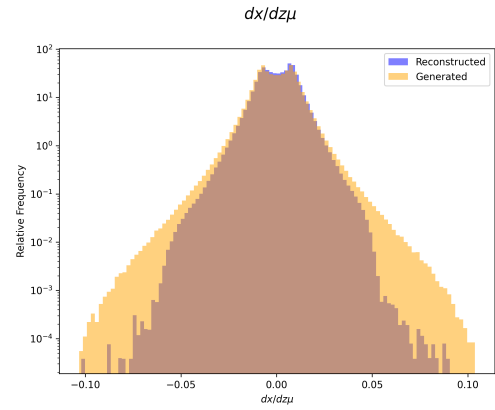


(b) Histogram for $Mom_{m\mu}$

Figure 4: Histograms for PV_y and $Mom_{m\mu}$

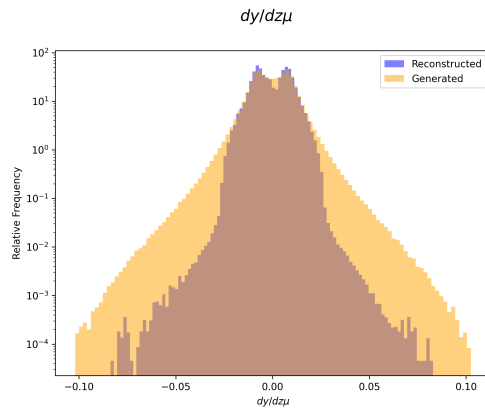


(a) Histogram for $Mom_{m\mu p}$



(b) Histogram for $d_x d_z m\mu p$

Figure 5: Histograms for $Mom_{m\mu p}$ and $d_x d_z m\mu p$



(a) Histogram for $d_y d_z m\mu p$

Figure 6: Histogram for $d_y d_z m\mu p$

Since we're going to use a 2D histogram for the chi-squared test to further evaluate the performance of the models (explained in section METER AQUI), we plotted the 2D histogram for the X and Y variables for the Inclusive:

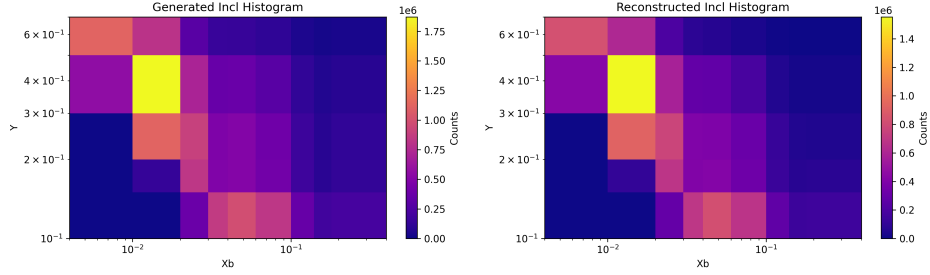


Figure 7: 2D Histogram with X and Y for the Inclusive Events

The histograms for the Hadrons events are shown in the figures below:

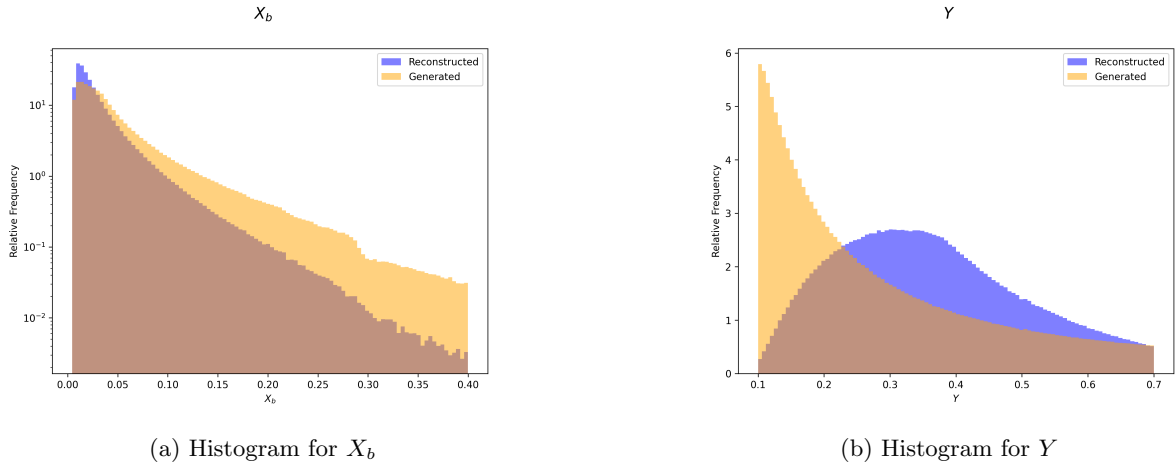
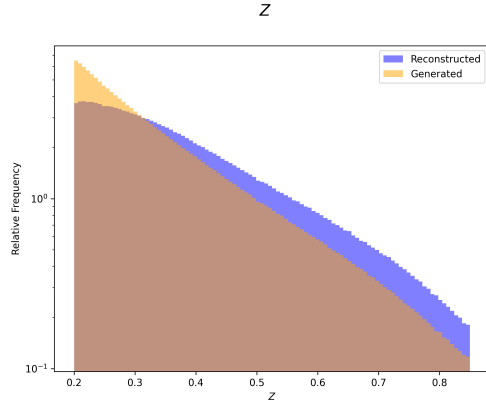
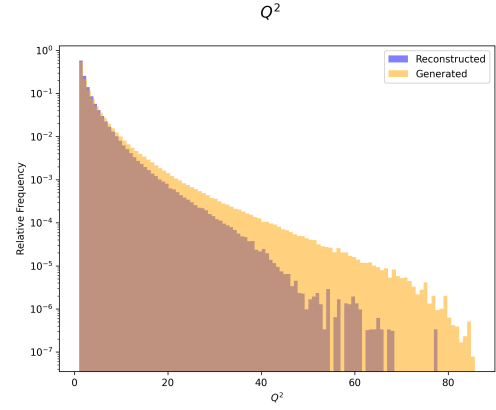


Figure 8: Histograms for X_b and Y

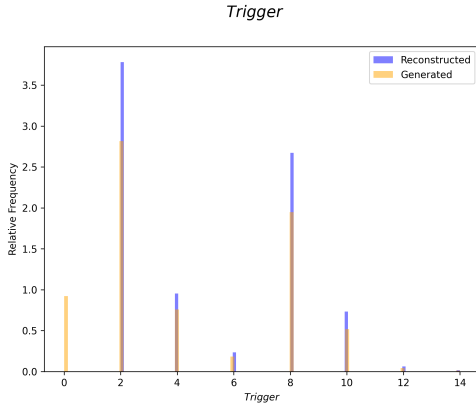


(a) Histogram for Z

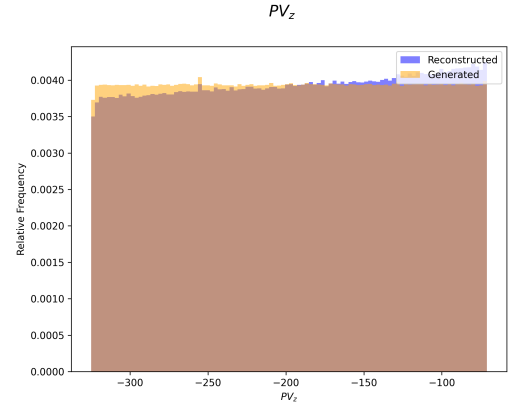


(b) Histogram for Q^2

Figure 9: Histograms for Z and Q^2

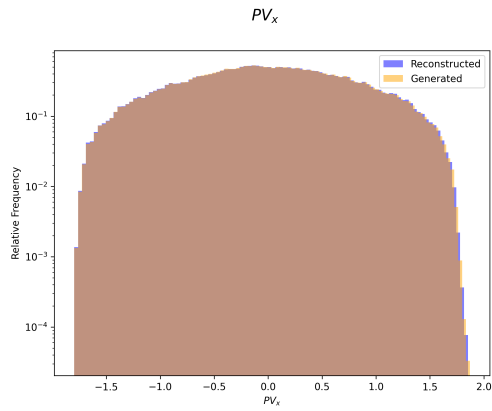


(a) Histogram for $Trig$

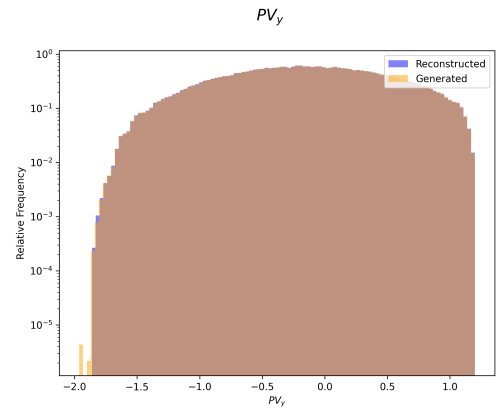


(b) Histogram for PV_z

Figure 10: Histograms for $Trig$ and PV_z

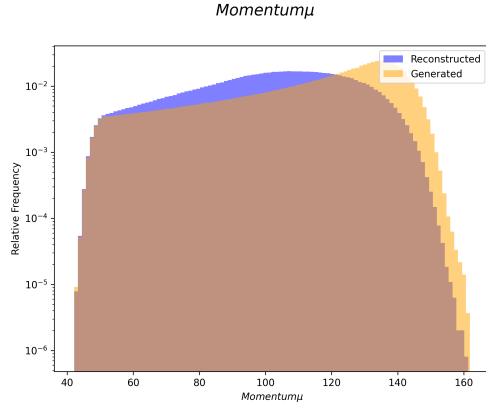


(a) Histogram for PV_x

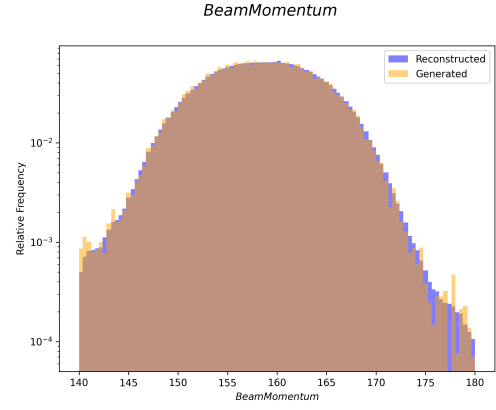


(b) Histogram for PV_y

Figure 11: Histograms for PV_x and PV_y

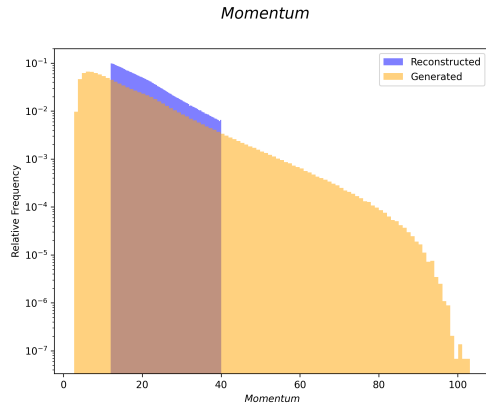


(a) Histogram for Mom_{mu}

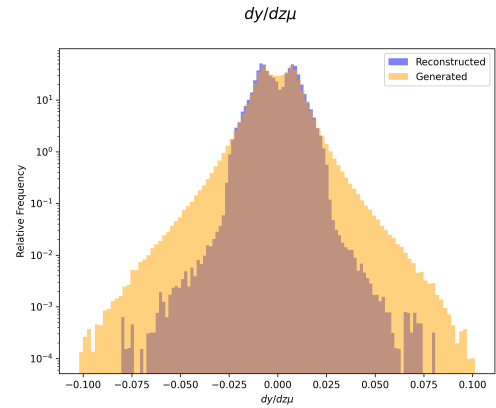


(b) Histogram for Mom_{mup}

Figure 12: Histograms for Mom_{mu} and Mom_{mup}

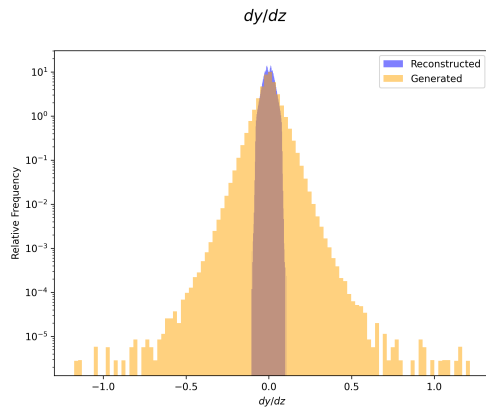


(a) Histogram for $d_x d_z mup$

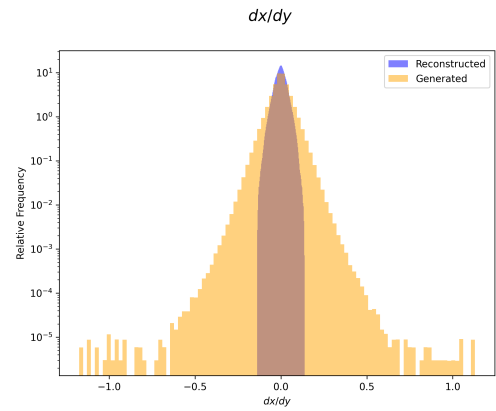


(b) Histogram for $d_y d_z \mu p$

Figure 13: Histograms for $d_x d_z mup$ and $d_y d_z \mu p$



(a) Histogram for $d_y d_z$



(b) Histogram for $d_x d_z$

Figure 14: Histograms for $d_y d_z$ and $d_x d_z$

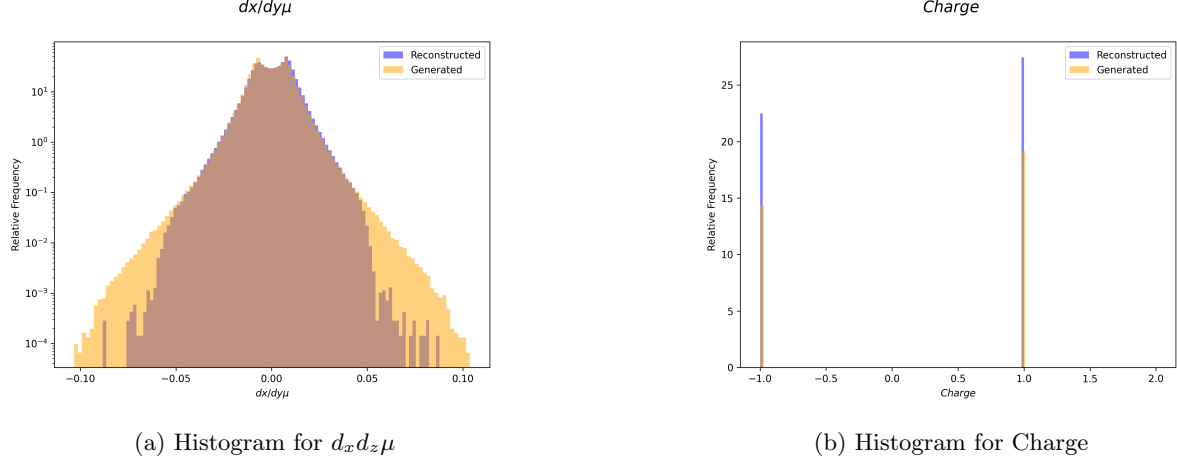


Figure 15: Histograms for $d_x d_z \mu$ and Charge

For the Hadrons events, we are going to use a 3D tensor for the X , Y and Z variables to further evaluate the performance of the models (explained in section METER AQUÍ). Since we cannot plot a histogram that allows us to visualize this '3D histogram', no image will be presented here, but it will be used in the chi-squared test. One could plot several 2D histograms, one for each Z value, but that would be very time consuming and not very practical.

2.4 Correlation between Variables

In order to check if there is any correlation between the variables, we plotted the correlation matrix with the Pearson Coefficient for the Inclusive and Hadrons events. This is important because if there is a high correlation between two variables, it means that they are redundant, and therefore, one of them can be removed, which simplifies the training of the Neural Network, reduces the number of parameters (reducing the training time) and avoids overfitting. The results are shown in the figures below:

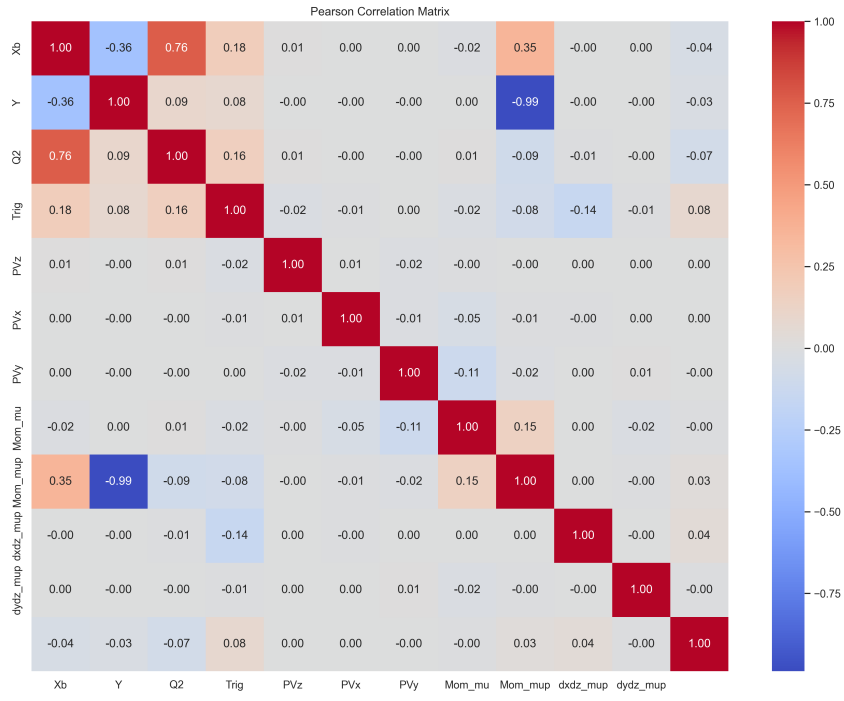


Figure 16: Correlation Matrix for the Inclusive Events

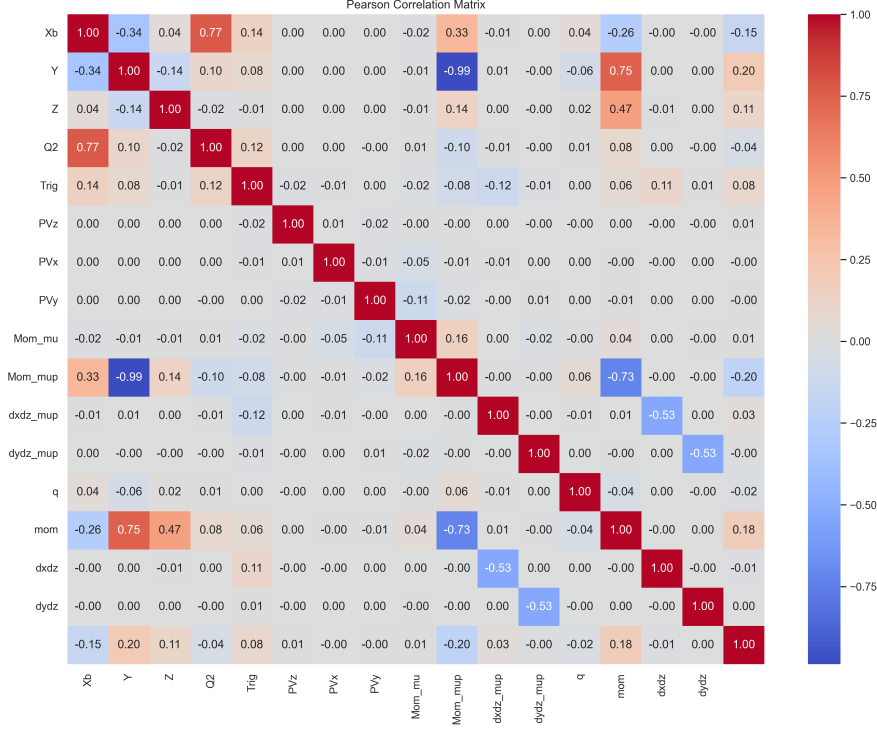


Figure 17: Correlation Matrix for the Hadron Events

As seen in the figures above, there is a high correlation between Y and Mom_{mup} , with a pearson correlation coefficient of -0.99 for both types of events. Therefore, one of them can be removed from the training of the Neural Network. Since Mom_{mup} and Y have the same correlation coefficient, we decided to remove Mom_{mup} from the training of the Neural Network, since from the physics point of view, Y is more important than Mom_{mup} , since it is the variable that gives us the virtual photon energy fraction of the initial momentum.

2.5 Weighting

As previously mentioned, the datasets from the Monte Carlo and the reconstructed data exhibit variations in the number of events. The Monte Carlo dataset contains a larger number of events compared to the reconstructed data. Consequently, we needed to address this disparity to ensure fair and effective training and testing of the Neural Network.

Our approach involved a two-step process. Initially, we trained the Neural Network using only a small percentage of the data to fine-tune the hyperparameters. Subsequently, we performed the main training using the entire dataset and tested the model with the full dataset. However, due to the differing event counts in each dataset, it was imperative to balance the data during both stages of training and testing.

While fine-tuning the hyperparameters, it was not feasible to utilize an equal count of events from the Reconstruction and Generation datasets, given their inherent numerical differences. Therefore, our solution involved using a proportional representation by employing percentages of the data. This approach is crucial because using an event count-based method would result in varying proportions of accepted and rejected events in the training and testing datasets, which could skew the model's performance evaluation.

3 Neural Network Architecture and Training

3.1 Models Architecture, Hyperparameters and Metrics

3.1.1 Architecture of the Models

For the Neural Networks models we used the PyTorch framework, which is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing. For the models architecture, two different models were used, one with 3 hidden layers, two GELU activation functions and two dropout layers, and another with 6 hidden layers, 5 GELU activation functions and 3 dropout layers. The choice of the number of hidden layers and the number of neurons in each layer was done by trial and error, and the same applies to the choice of the dropout rate. The GELU activation function was chosen because it is the one that gave the best results, and the same applies to the choice of the optimizer, which was the Adam optimizer. These two models were chosen because they are the ones that gave the best results and also because there was the need to see the difference between a not so deep model and a deeper model. The hyperparameters used for the training of the models were the following:

- Learning Rate: 0.01
- Epochs: 10
- Batch Size: 1

For the loss function, we used the Binary Cross Entropy Loss, which is the most common loss function used for binary classification problems. Despite the fact that this set of data might be familiar with a binary classification problem, since we are trying to predict if an event will be accepted or rejected, we are not doing a binary classification problem, since we are not trying to classify the events, but rather predict the acceptance of the COMPASS/AMBER. Therefore, there was no need to round the output of the Neural Network to 0 or 1, since the output of the Neural Network is a number between 0 and 1, which represents the probability of the event being accepted.

3.2 Evaluation of the Models

During the training of the models, we printed the accuracy and the loss as metrics to evaluate part of the performance of the models. But since the accuracy and loss are not the best metrics to evaluate the performance of the models in this type of problem, the chi-squared test was used to evaluate the performance of the models. The chi-squared test is a statistical hypothesis test that is used to evaluate how well a set of data fits a model. In this case, the model is the Neural Network, and the data is the data used for the training and testing of the Neural Network. For this particular case, the chi-squared was applied to:

- Hadrons: 3D histogram with X , Y and Z variables
- Inclusive: 2D histogram with X and Y variables

The chi-squared used was defined as:

$$\chi^2 = \sum_{i=1}^N \frac{(\frac{r}{g} - A_{nn})^2}{err} \quad (1)$$

where r is the number of reconstructed events in the bin, g is the number of generated events in the bin, A_{nn} is the output of the Neural Network for the bin and err is the error of the bin, which is defined as:

$$err = \sqrt{\frac{r}{g} + (\frac{r}{g})^2} \quad (2)$$

For Inclusive events, we anticipate a chi-squared value of approximately 45. It's worth noting that the expected value should theoretically be 60 since there are 60 bins. However, due to certain bins appearing to be empty, we are expecting a somewhat lower value.

In the case of Hadron events, we are looking for a chi-squared value of roughly 400. Similarly, the theoretical expected value would be 780, corresponding to the number of bins. Nonetheless, given the presence of apparently empty bins, we anticipate a reduced value.

3.3 Training, Testing and Results

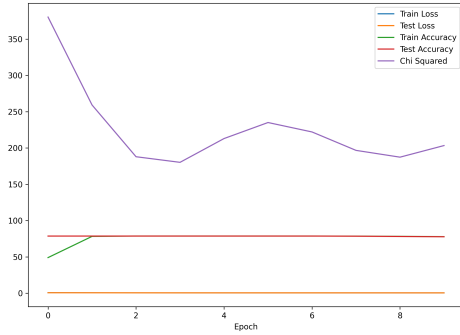
During the model training process, an 80% portion of the dataset was dedicated to training, while the remaining 20% was allocated for testing. Notably, both models underwent training and testing with the same dataset to ensure a fair basis for comparison.

Throughout the training phase, metrics like accuracy and loss were monitored to assess the models' performance. However, the primary focus was on tracking the chi-squared values, which played a crucial role in identifying the best models.

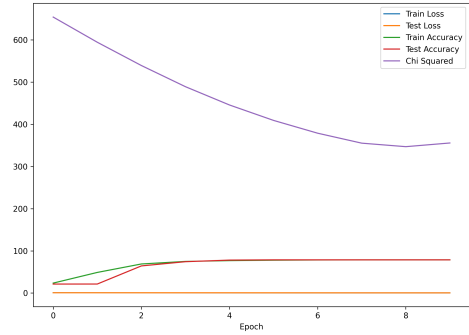
It's worth noting that the chi-squared values exhibited some instability during training. To account for this, models with chi-squared values were preserved for post-training model selection. This approach allowed for the selection of the best models after considering the chi-squared performance.

Multiple iterations of training were carried out due to the previously mentioned chi-squared instability. To gain insights into the training process, accuracy and loss metrics for both training and testing were plotted and visualized, as depicted in the figure below:

For the Hadron events:



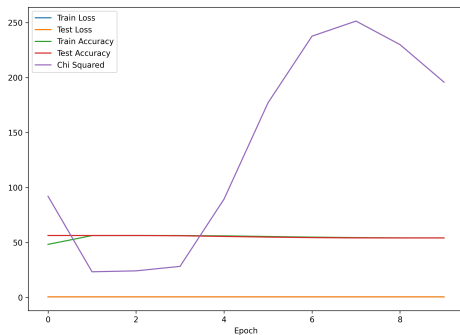
(a) Shallow Model for Hadron



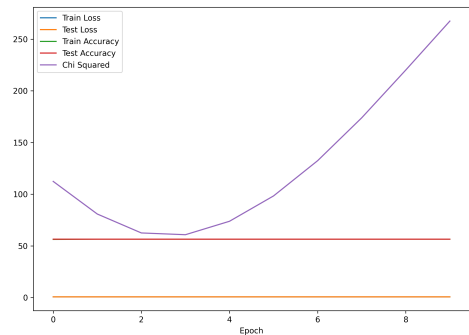
(b) Deep Model for Hadron

Figure 18: Comparison of Shallow and Deep Models for Hadron

For the Inclusive events:



(a) Shallow Model for Inclusive



(b) Deep Model for Inclusive

Figure 19: Comparison of Shallow and Deep Models for Inclusive

Please keep in mind that the image presented here represents just one of the training sessions.

As observed in the image above, the deeper model exhibited a relatively stable performance but yielded notably high chi-squared values. On the other hand, the less deep model displayed instability but delivered

lower chi-squared values. Consequently, we made the decision to discontinue the deeper model and retain the less deep one.

Following multiple training sessions and employing a transfer learning approach—initiating the training process with the weights from the best-performing model among the previous runs—we successfully obtained models that met the required chi-squared criteria.

To evaluate the models, we generated histograms illustrating the output of the Neural Network divided by the True Acceptance for both Inclusive and Hadrons events, each subjected to different rounding thresholds. For the Inclusive events we can plot the 2D histogram as seen in the figure below:

(a) Threshold of 0.5

(b) Threshold of 0.75

Figure 20: Histograms for the Output of the Neural Network for the Inclusive Events

For the Hadron events we cannot plot the 3D histogram, but one can see the values for the output of the Neural Network divided by the True Acceptance if running the code. However, was decided to present the mean value of the ratio between the output of the Neural Network and the True Acceptance for the Hadron events, which is 1 for the 0.5 threshold and 1 for the 0.75 threshold.

As evident from the figures provided above and the means, the models are making accurate predictions regarding the acceptance of COMPASS/AMBER.

3.4 Structure of the detector

Out of curiosity, we contemplated the possibility of utilizing the muon and hadron angles to attempt the reconstruction of the detector’s structure. It is well understood that when particles interact with physical objects, they may either be absorbed or deflected, leading to a loss in acceptance. As a result, we proceeded to generate histograms for the angles of the muon and the hadron, the results of which can be observed in the figures below:

Figure 21: Histogram for the Angle of the Muon

Figure 22: Histogram for the Angle of the Hadron

3.5 Generalization with Other Data

In order to assess the models’ generalization capabilities, we employed a separate dataset and generated histograms for the Neural Network’s output concerning both Inclusive and Hadron events. We then proceeded to divide the output by the True Acceptance and calculated the mean value of the ratio, which is was 1 for the Hadron events and 1 for the Inclusive events.

As the ration is close to 1, one can say the models are predicting the acceptance of the COMPASS/AMBER as desired.

4 Improvements

This problem is inherently challenging due to the complex nature of the data and the substantial volume of events involved. As such, there is considerable potential for enhancement. One avenue for improvement involves exploring alternative Neural Network architectures, such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Graph Neural Networks (GNN), among others. It’s worth noting that these architectures were not employed in this project due to constraints related to time, computational resources, and project scope.

5 Conclusion