ELTE
EÖTVÖS LORÁND
TUDOMÁNYEGYETEM

ML-based Event Reconstruction Development for the DUNE ND-GAr

Mahmoud Ibrahim

Eötvös Loránd University

Faculty of sciences

The framework to improve reconstruction algorithm

The effort:

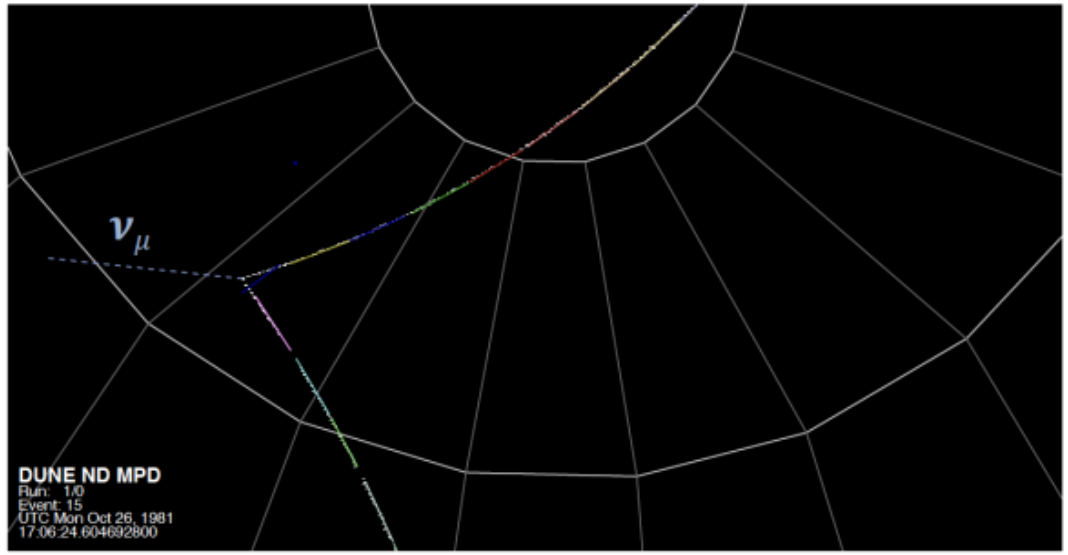
- to continue/extend Thomas Campbell's low energy proton reconstruction work.
- Thomas finish his development in 2019 (4 years ago)
- to introduce some improvements in primary vertex finding and selection
- the reconstruction algorithm employs a ML technique.

The goal:

Machine learning opportunity:

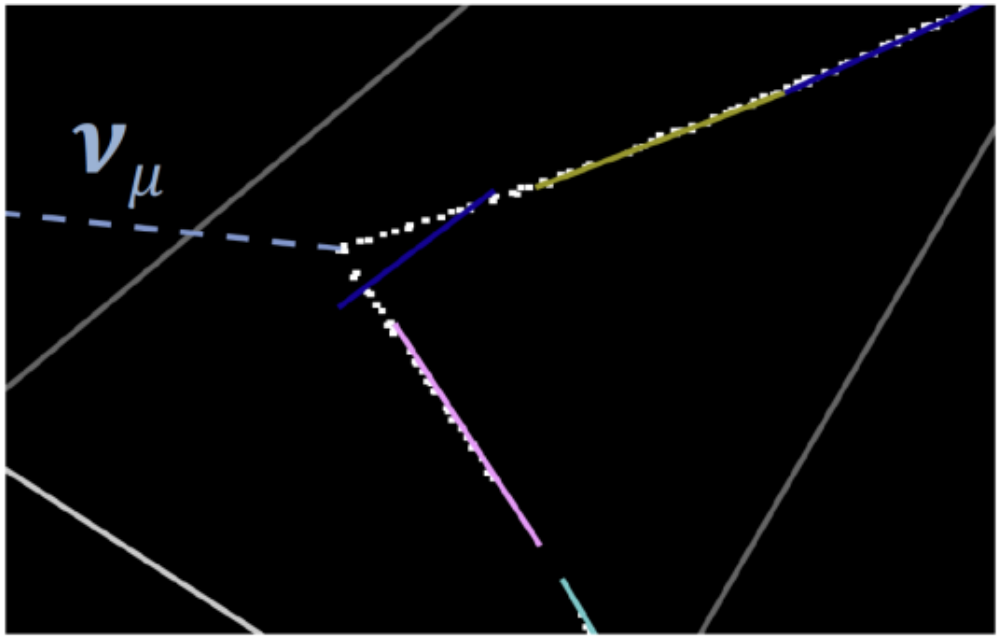
- Look at tracks near the primary vertex
- A 3D CNN with raw digits or TPC clusters as input and 3D reconstructed vertices
- Can extend this to estimate the number of short protons stubs at the primary.

A Need: Better Vertex Reco



A two-track ν_μCC event. Tiered pattern recognition in GArSoft
TPC Clusters shown in white, Vector hits shown in colors. Tracks (not shown) built out of vector hits.

The Problem Near the Primary Vertex



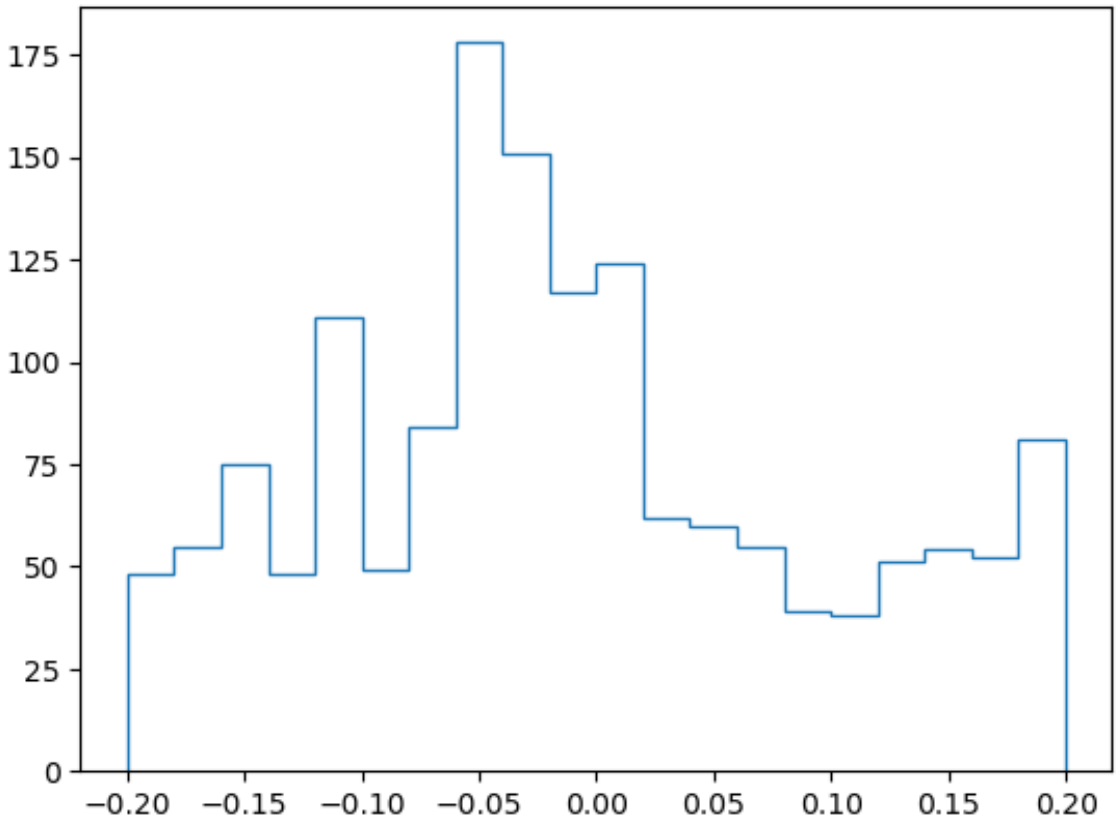
Same event, zoomed in. Blue vector hit contains some contamination from the pink one.
Tracks stop or bend near the primary, worsening efficiency and resolution.

By Tom Junk at DUNE CM in September 2022

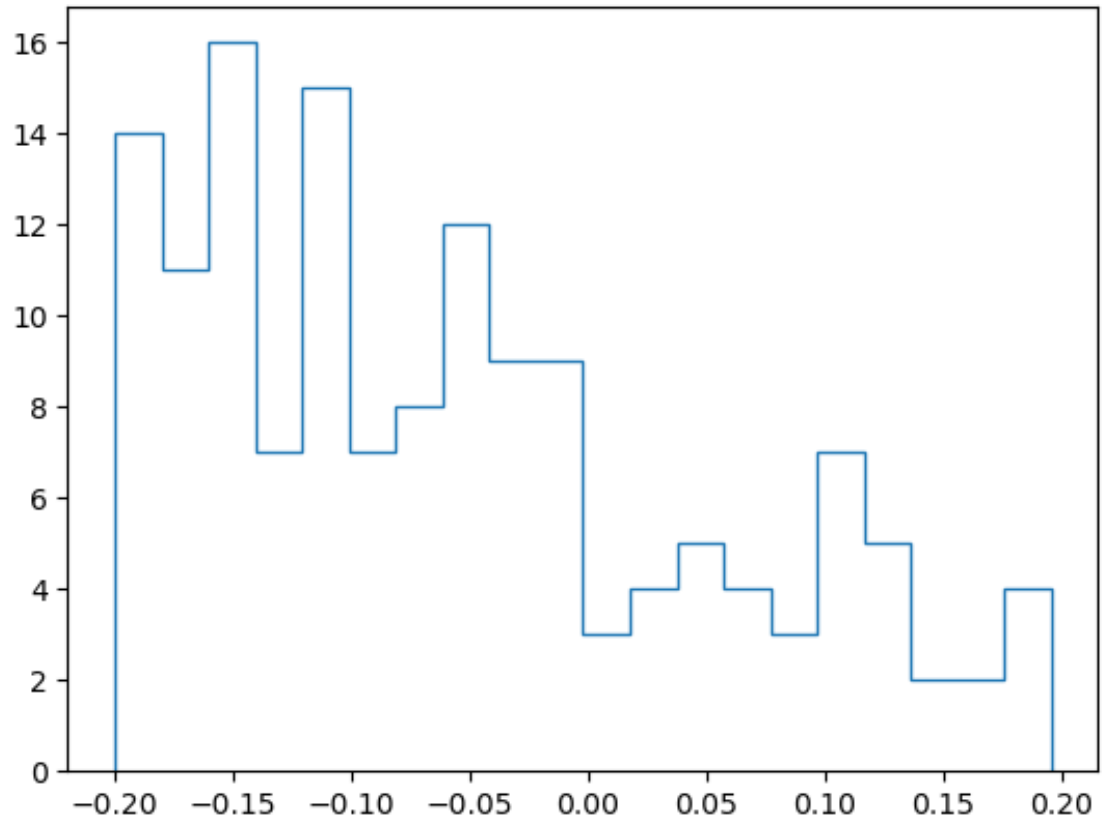
Setting up the environment and Running ND-GAr software：

- Built the FNAL Computing Environment
- Have used the ND-GAr software for event generation (1000 event) and detector simulations and event reconstructions.
- Made an analysis flat tree and used to produce histograms that can be used as an input for the machine learning .

The Histograms produced from the analysis flat tree :

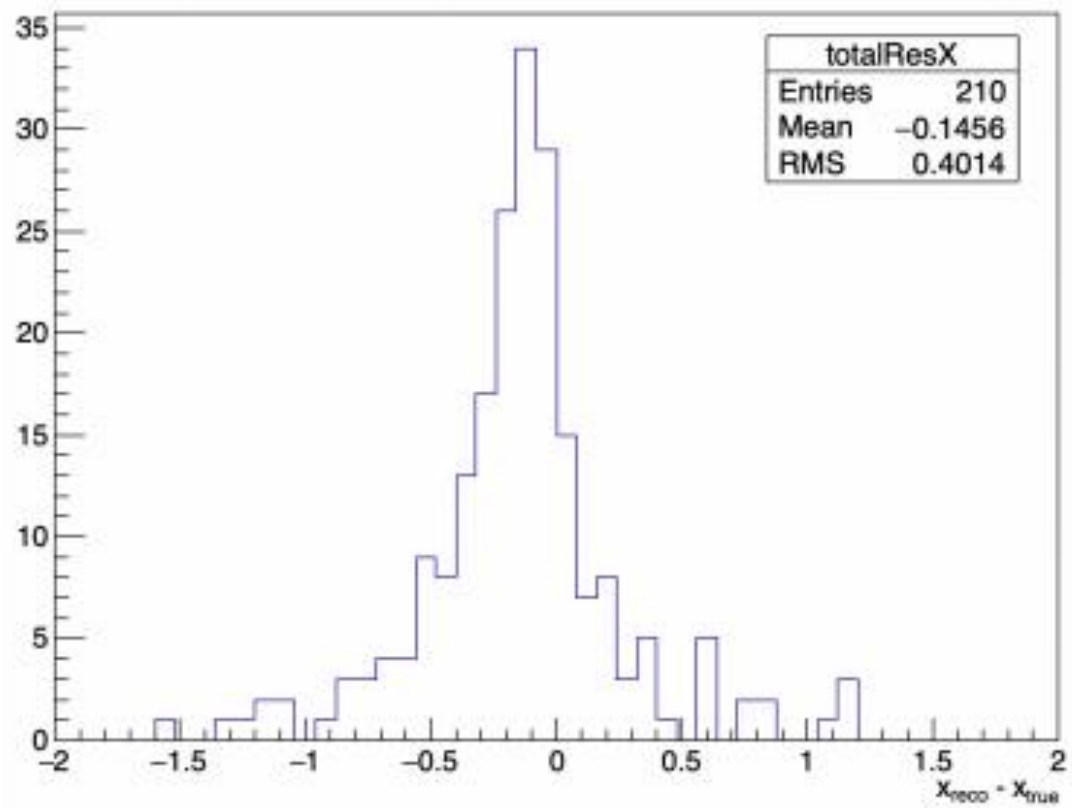


Momentum difference between MC (simulated position) and reco (reconstructed position)

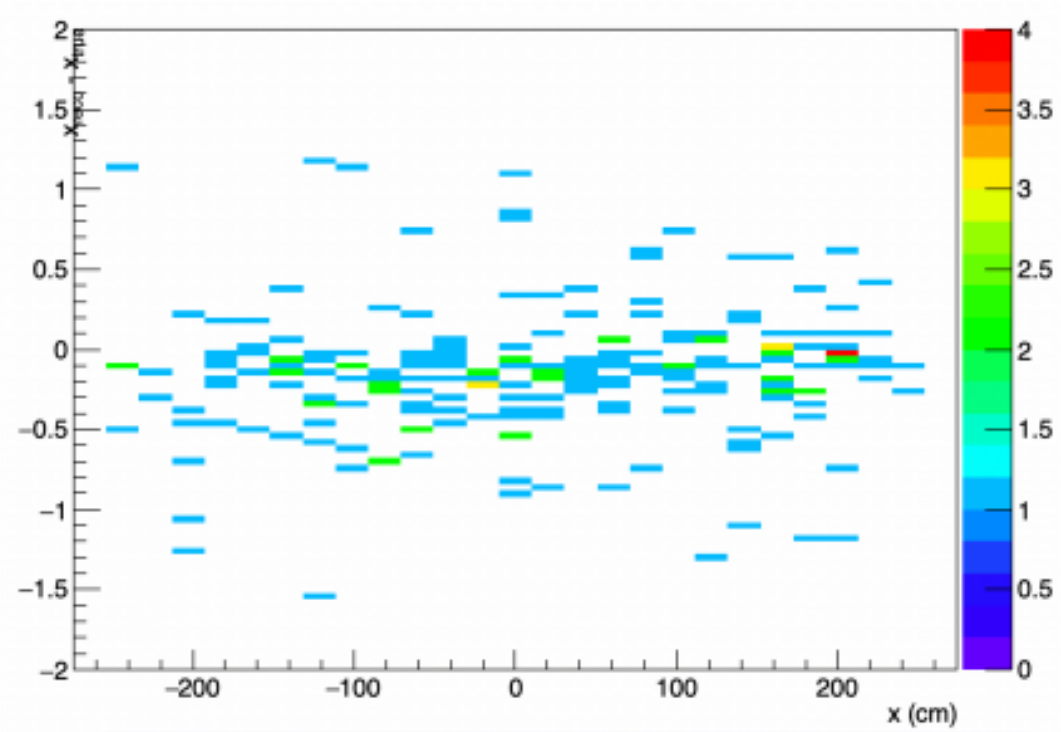


Start position difference between MC (simulated position) and reco (reconstructed position)

The Histograms produced from the analysis flat tree :



Vertex position difference between MC (simulated position) and reco (reconstructed position)



Vertex position difference between MC (simulated position) and reco (reconstructed position) as a function of the detector X position

The Framework prepared by Thomas Campbell

Particle Identification Learning for Low Energy Argon Gas Events (PILLAGE). A combination of a Random Sample Consensus (RANSAC) based clustering algorithm for identifying short, linear tracks and a Neural Net for particle ID classification and observable regression on the RANSAC's tracks.

It consists of two main codes

- Particle ID (PILLAGE): https://github.com/dr-thomas/PILLAGE
- Tracking (RANSAC): https://github.com/dr-thomas/RANSAC
- Both are designed to use ND-GAr output information as the inputs for ML-training

Note:

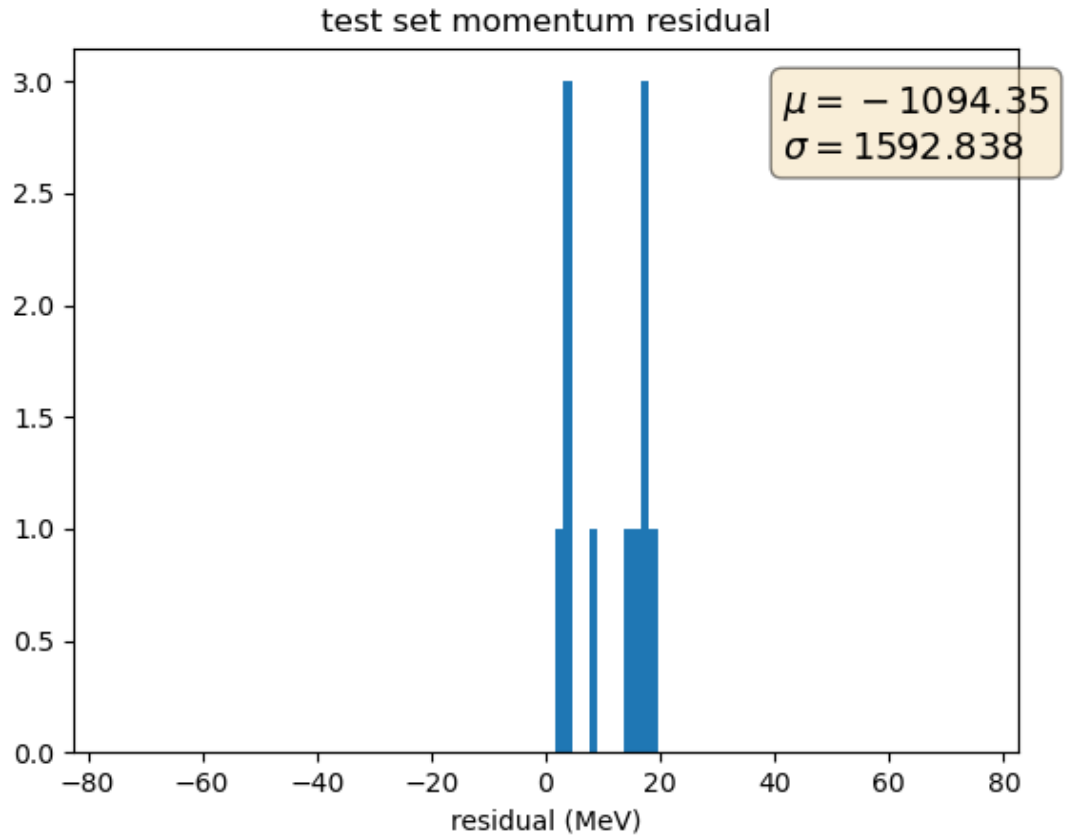
- It was developed in 2019, because of that it is not compatible with the latest ND-Gar software.

- Running the Framework prepared by Thomas Campbell:
  - Since the code was developed back in 2019, the variables names were no longer compatible with the produced anatree so all the codes had to be modified so they process the correct and desired data.
  - The codes then were run to produce the train data as a csv file
  - The train data was then divided into test/train data by 1:9 ratio.
  - The model was trained using the MLPRegressor class from the sklearn.neural_network module.

MLPRegressor is an artificial neural network model that uses backpropagation to adjust the weights between neurons in order to improve prediction accuracy

The outcome of the neural network training:

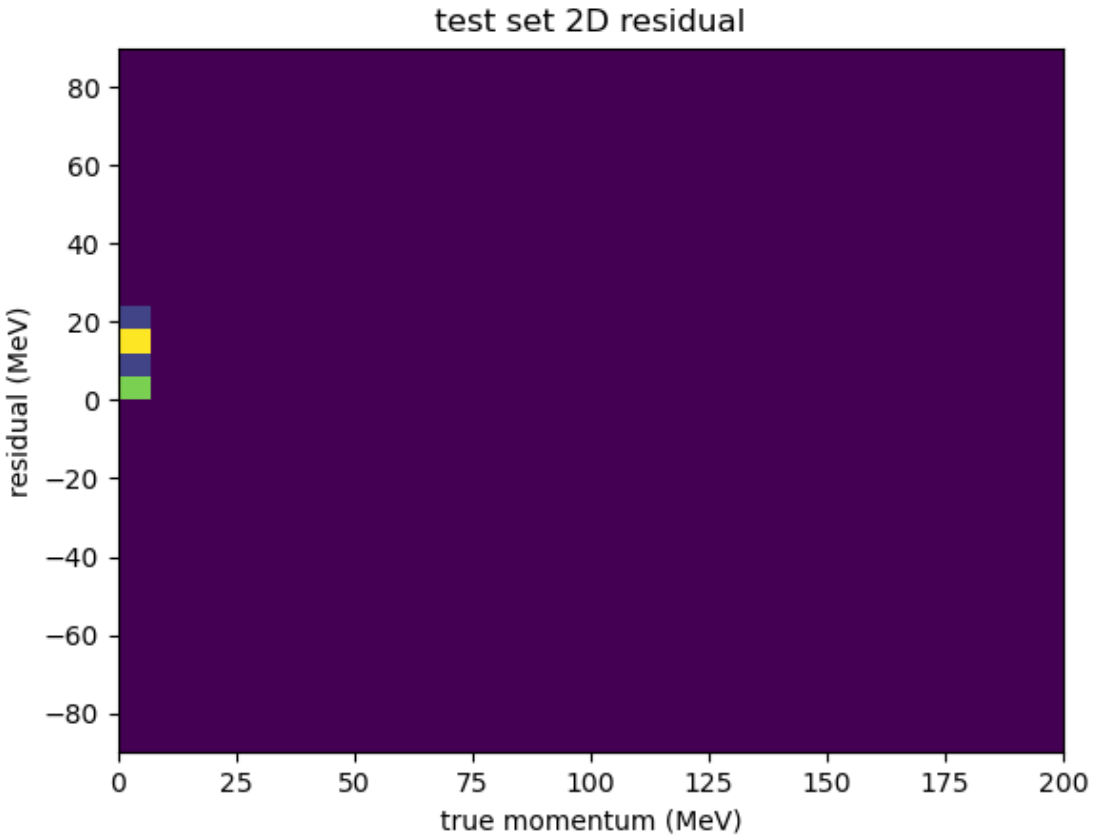
- After the neural network is trained using the fit method, the model's predictions on the testing set are computed using the predict method. The residuals (the difference between the predicted and actual target values) are then computed and saved.
- A warning message was received that the fitting method class did not converge during training to a minimum.
- This could be due to a variety of reasons, including insufficient training data, or the presence of noisy or complex data that is difficult to learn.



"mu" refers to the mean of the residual distribution, which is the average value of the residuals. It is calculated as the sum of all the residuals divided by the number of residuals.

"Sigma" refers to the standard deviation of the residual distribution

The outcome of the neural network training:



"mu" refers to the mean of the residual distribution, which is the average value of the residuals. It is calculated as the sum of all the residuals divided by the number of residuals.

"Sigma" refers to the standard deviation of the residual distribution

Summary:

- Used the ND-GAr software to generate 1000 events and simulated detector behavior for each event
- Analyzed the data by creating a flat tree and producing histograms to use as input for machine learning
- Modified Thomas Campbell's framework developed in 2019 to process desired data
- Produced a train data CSV file by running the modified codes
- Split train data into test and train data with a ratio of 1:9
- Used the MLPRegressor class from the sklearn.neural_network module to train the model
- Used the predict method to compute the model's predictions on the testing set
- Saved the residuals (the difference between the predicted and actual target values)
- Received a warning message during training that the fitting method class did not converge to a minimum
- Possible reasons for the warning include insufficient training data or noisy or complex data that is difficult to learn.

Next Steps:

- I have a script that will create several arrays and histograms for the data after we collect better data. It iterates over each event in the loaded data, and for each event, it appends the predicted momentum and true momentum to two different arrays. It then calculates the residual between the predicted momentum and true momentum for each pair of momenta within an event and appends the residual and predicted momentum to two different arrays. It also calculates the kinetic energy associated with each momentum and creates histograms for the residual in kinetic energy, the predicted kinetic energy, and the true kinetic energy.
- Eventually, another script will produce the fractional occupancy of the detector.

Question

- Are there any additional pre-processing or feature selection to reduce noise and complexity in the data ?

- Are there an official large statistic MC that we can use instead of private production?