# Charged Track Reconstruction with Artificial Intelligence for CLAS12

*Gagik* Gavalian[1,*], *Polykarpos* Thomadakis[2,**], *Angelos* Angelopoulos[2,***], and *Nikos* Chrisochoides[2,****]

[1]Jefferson Lab, Newport News, VA, USA
[2]CRTC, Department of Computer Science, Old Dominion University, Norfolk, VA, USA

**Abstract.** In this paper, we present the results of charged particle track reconstruction in CLAS12 using artificial intelligence. In our approach, we use neural networks working together to identify tracks based on the raw signals in the Drift Chambers. A Convolutional Auto-Encoder is used to de-noise raw data by removing the hits that do not satisfy the patterns for tracks, and second Multi-Layer Perceptron is used to identify tracks from combinations of clusters in the drift chambers. Our method increases the tracking efficiency by 50% for multi-particle final states already conducted experiments. The de-noising results indicate that future experiments can run at higher luminosity without degradation of the data quality. This in turn will lead to significant benefits for the CLAS12 physics program.

## 1 Introduction

Nuclear Physics experiments have become increasingly complex over the past decades, with more complex detector systems and higher luminosities. In emerging experiments where detector occupancies are higher, there is a need for new approaches to data processing that can improve data reconstruction accuracy and speed. New developments in the Artificial Intelligence (AI) field present promising alternatives to conventional algorithms for data processing. Machine Learning (ML) algorithms are being employed in various stages of experimental data processing, such as detector data reconstruction, particle identification, detector simulations, and physics analysis.

In this paper, we present the implementation of machine learning models into the CLAS12 charged-particle track reconstruction software. Detailed analysis of the reconstruction performance are presented, comparing track reconstruction efficiency and speed improvements to conventional algorithms.

---

*e-mail: gavalian@jlab.org
**e-mail: pthom001@odu.edu
***e-mail: aangelos28@gmail.com
****e-mail: npchris@gmail.com

## 2 Track Identification with Machine Learning

### 2.1 Charged Particle Tracking

The CLAS12 [1] forward detector is built around a six-coil toroidal magnet which divides the active detection area into six azimuthal regions, called "sectors". Each sector is equipped with three regions of drift chambers [2] designed to detect charged particles produced by the interaction of an electron beam with a target. Each region consists of two chambers (called super-layers), each of them having 6 layers of wires. Each layer in a super-layer contains 112 signal wires, making a super-layer a 6x112 cell matrix. The schematic view of one region is shown on Figure 1 (right panel).
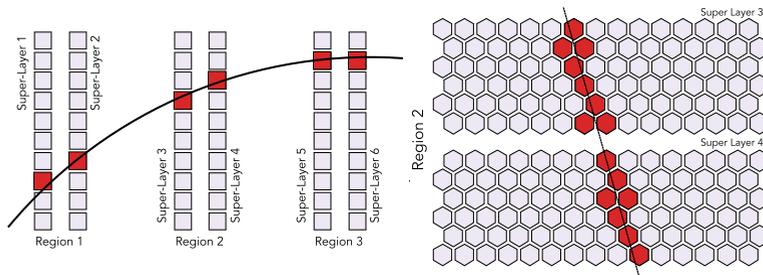


**Figure 1.** Schematic view of signals generated in the drift chambers when a particle passes through. The segments in each super-layer are shown along the trajectory of the track (left panel), and view of the activated cells in the two super-layers of one region along the track trajectory (right panel).

Particles that originate at the interaction vertex travel through the magnetic field and pass through all three regions of the drift chambers in a given sector are reconstructed by tracking algorithms. First, in each super-layer adjacent wires with a signal are grouped together into clusters (called segments), shown in Figure 1. The positions of these clusters (segments) in each super-layer are used to fit the track trajectory to derive initial parameters, such as momentum and direction. After the initial selection, good track candidates are passed through a Kalman filter [3] to further refine measured parameters.
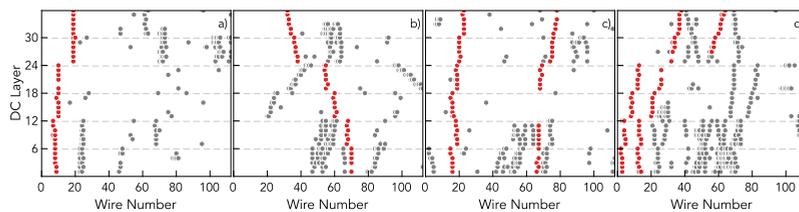


**Figure 2.** Example of signals in drift chambers for four events. Each plot represents one sector with a 36x112 matrix of wires. Background hits (in gray) are shown along with the hits of reconstructed identified tracks (in red). Dashed lines represent boundaries between super-layers.

For each beam-target interaction or "event", drift chambers produce many segments, some belonging to a track and some are background or partial trajectories of low-momentum

tracks. In Figure 2 drift chamber signals in one sector are shown for four different events, in each sector data are hits represented as a 36x112 matrix (36 layers and 112 wires per layer), showing all hits including those that were determined to be part of a track.

Due to inefficiencies in the drift chambers, it is possible to have one missing segment along the trajectory of the particle, and the track has to be reconstructed using only 5 segments. An example of a 5-segment track is shown in Figure 2 c), where super-layer 3 does not have any segment detected. For these types of tracks, candidates have to be identified from a large number of combinatorics consisting of all combinations of clusters that form 5-segment candidates.

## 2.2 Track Classifier

A Multi-Layer Perceptron neural network is used for the CLAS12 track identification [4]. The implemented architecture is shown in Figure 3, where an input layer with 6 nodes are used (each node representing the average wire position of the segment in the super-layer) and 3 output nodes for the classes "positive track", "negative track" and "false track".
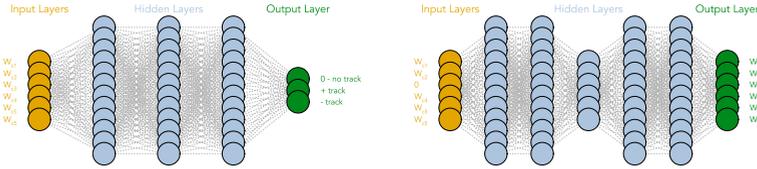


**Figure 3.** a) Architecture of Multi-Layer Perceptron used for track classification. The network has 6 input nodes, corresponding to the average wire positions of the segment in each super-layer, three hidden layers with 12 nodes in each, and 3 output nodes. b) Corruption-recovery auto-encoder architecture with 6 input nodes representing track segments mean wire values with one of the values set to 0 and 6 output nodes with the correct value for the node that has 0 in the input.

The network is trained using a sample of tracks from experimental data reconstructed by the conventional algorithm, selected with $\chi^2$ cuts to retain the highest quality ones, which are fed to the network with their respective labels (i.e. positive or negative tracks). For false tracks, a combination of segments (6 segments forming a track candidate) that was not identified as a track by the conventional algorithm is chosen.

## 2.3 Corruption Auto-Encoder

A second neural network was developed to fix the corruption in possible track candidates due to inefficiencies of the drift chambers. This network was used to identify track candidates which have one of the segments missing. We used an auto-encoder architecture to implement the corruption-recovery neural network [5]. The structure of the network can be seen in Figure 3, with 6 input nodes and 6 output nodes.

To train the corruption auto-encoder network the same sample used for the classifier training was used. The output for the network was set to the good track parameters (where all 6 segments have non-zero values) and the input was modified by setting one of the nodes (randomly) to zero. The network learns to fix the node containing zero, by assigning it a value based on the other 5 segment values.
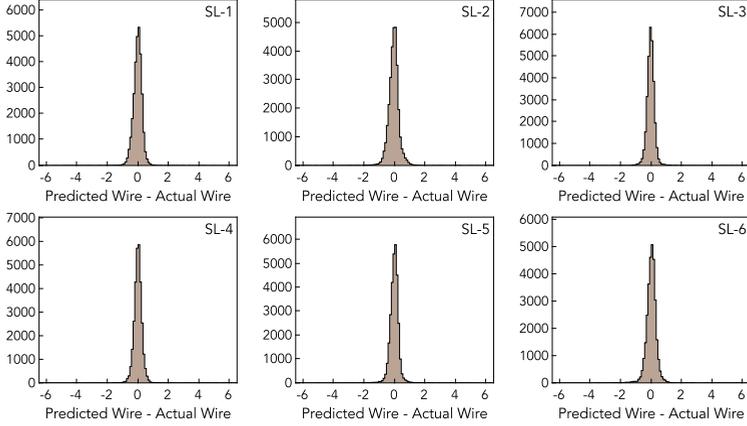
**Figure 4.** Performance of corruption recovery auto-encoder for each of six super-layers. The difference between predicted and actual wire position is shown for all super-layers. The average accuracy of wire position prediction is 0.36 wire.

The test results of the trained network are shown in Figure 4, where the difference between the true value of the segment position and the one reconstructed by the network is plotted, showing a reconstruction accuracy of 0.36 wires. In Figure 4 this difference is shown for each super-layer that was corrupted in the input. As can be seen, the performance of the network is uniform across all super-layers.

## 2.4 Track Identification Workflow

Track identification consists of two phases, programmed to be done in two passes. In the first pass over the data, signals from each sector of drift chambers are analyzed to create a track candidate list, each consisting of 6 segments. The resulting track candidates are evaluated by the classifier neural network and are assigned a probability of being either a positive or negative track.

The second stage of track identification starts by constructing a list of track candidates with combinations of 5 clusters out of 6 from all existing clusters (one per super-layer). The candidates that share a cluster with tracks identified at the first stage of classification are removed from the list. For each track candidate with a missing cluster in one of the super-layers, a pseudo-cluster is generated using the Corruption Auto-Encoder Network and the missing super-layer cluster are assigned the inferred value, hence turning all track candidates into 6 cluster track candidates. The cured (or fixed) track candidate list is finally passed to the track classifier module described above, which evaluates the list by isolating candidates with the highest probability of being a good track.

## 2.5 Physics Impact

To measure the implications of track reconstruction efficiency improvements on physics analysis, we considered two event topologies with two and three particles in the final state, respectively. The data for analysis were taken with 10.6 $GeV$ electron beam incident on 5 $cm$ liquid hydrogen target, with a beam current of 45 $nA$. We selected events where an electron
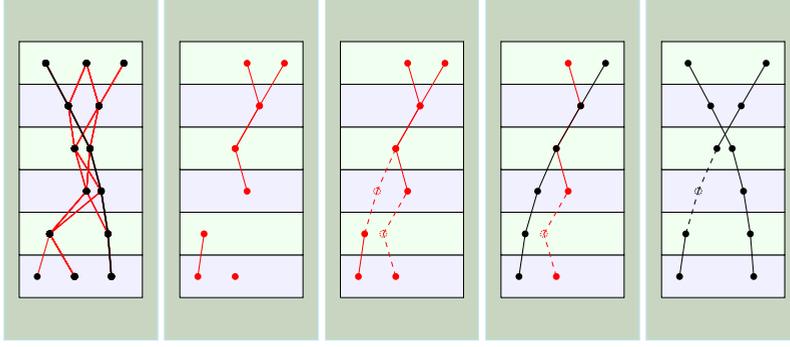
**Figure 5.** Stages of Neural Network track identification procedure. 1) identifying 6 super-layer tracks. 2) removing all hits belonging to an identified track and constructing 5 super-layer track candidates. 3) generating pseudo-clusters for 5 super-layer track candidates using corruption fixing auto-encoder. 4) identify good track candidates from the list of 6 super-layer (one of the super-layers is a pseudo-cluster) track candidates. 5) isolate both identified (6 super-layer and 5 super-layer) tracks for further fitting with Kalman-Filter.
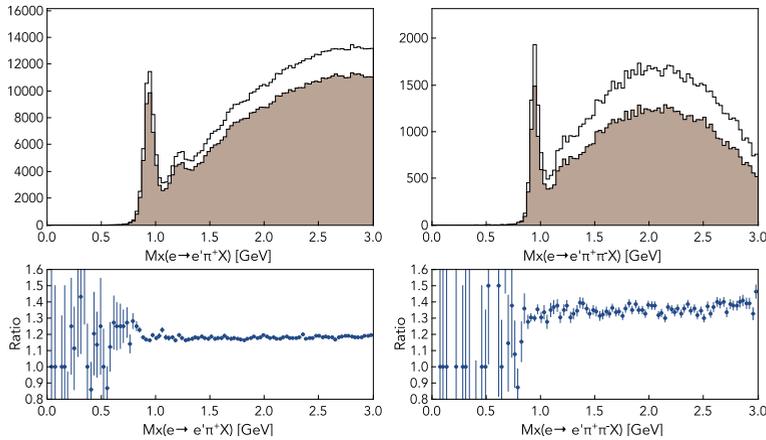


**Figure 6.** Reconstructed missing mass distribution for $H(e, e'\pi^+ X)$ and $H(e, e'\pi^+\pi^- X)$ reactions (top row) using the conventional track reconstruction algorithm (filled histogram) and AI-assisted track reconstruction (black line histogram). The ratios of the two histograms are shown on the bottom row.

was detected in the forward detector, and then isolated events where there was an additional positively charged pion ($\pi^+$) along with an electron and no other charged particle. The second topology required two pions along with the electron, one positively charged and one negatively charged. The two chosen topologies are denoted by $H(e, e'\pi^+ X)$ and $H(e, e'\pi^+\pi^- X)$. In both cases, there is a visible peak of a missing nucleon in the missing mass distribution of the detected final state, which we can use to measure the impact of efficiency on physics outcome.

The distributions of missing mass for both final state topologies are shown in Figure 6, where the plots on the top row are missing mass of $H(e, e'\pi^- X)$ and $H(e, e'\pi^+\pi^- X)$, where

the filled histogram is calculated from particles reconstructed by the conventional tracking algorithm, and the histogram with a black outline is the same distributions calculated from particles that were reconstructed using a suggestion from Artificial Intelligence. As can be seen from the figure, there is a significant increase in the number of events in the region of the nucleon peak for AI-assisted tracking. The ratios of the two histograms (AI-assisted divided by conventional) Figure 6 shows the increase in statistics is uniform over the whole range of the missing mass indicating no systematic abnormalities for AI-assisted tracking. The ratio also indicates that there is an increase in the number of events of about 15% for $H(e, e'\pi^+ X)$ final state and $30 - 35\%$ for the $H(e, e'\pi^+\pi^- X)$ final state.

## 3  Drift Chamber De-noising using Convolutional Auto-Encoders

The charged particle tracking relies on isolating clusters in each DC super-layer to identify tracks from the combination of clusters. Accidental hits in drift chambers, which increase with the luminosity, can lead to decreased efficiency of cluster identification, which in turn, leads to a decrease in track finding efficiency. A new approach with neural networks is used to clean raw data from drift chambers prior to applying the clustering algorithm.
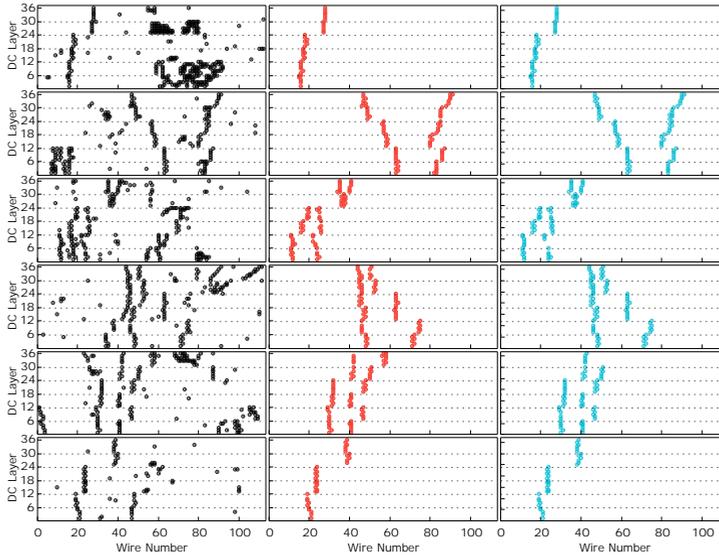


**Figure 7.** Results from the de-noising auto-encoder. The raw hits are shown in the left column for five random events, along with hits reconstructed by the CLAS12 tracking algorithm in the middle column. The resulting hits matrix from the de-noising raw hits are shown in the right column. (Systematic studies of de-noiser performance can be found here [6])

### 3.1  Convolutional Auto-Encoder Method

The Convolutional Auto-Encoder is used to de-noise raw data from the CLAS12 drift chambers [6]. The input and output for the network are matrices of size 36x112 representing hits in one sector of the drift chambers. The training data was extracted from experimental data processed with CLAS12 reconstruction software. The raw hits (converted into a matrix) are

used as an input for the neural network and a matrix constructed only from hits that belong to reconstructed tracks as an output.

An example of comparison can be seen in Figure 7 where raw data (left column) are shown along with data with hits belonging to reconstructed tracks identified by the conventional tracking algorithm (middle column) and reduced data processed by a de-noising neural network (right column).

Systematic studies [6] showed that more than 95% of the track-related hits are preserved in the output of de-noiser while background hits are significantly suppressed for normal experimental conditions of 45 *nA* incident beam current.

## 3.2 Physics Impact

The full chain of AI tools is implemented in the CLAS12 workflow. The reconstruction of tracks in drift chambers starts with a De-Noising auto-encoder which removed the hits that are determined to be noise from raw DC data, then the clustering algorithm finds clusters in the drift chambers individually for each super-layer. The clusters are further processed with AI-assisted tracking to find combinations of clusters that form a track. After the track candidates were identified the DC conventional algorithm is used to reconstruct track parameters using Kalman-Filter.

In Figure 8 the results from simulations are shown where the complete chain of AI was used to reconstruct simulated data. The simulations were done using Pythia for the final state $(e^-, \pi^+\pi^-)$, and a background was added for different experimental conditions (luminosity). The results of the number of protons in the missing mass are shown as a function of the beam current. As can be seen from the figure the efficiency of the proton reconstruction is significantly improved when AI is used in track reconstruction. The results suggest that the experiments can run efficiently at higher beam currents with AI without loss of track reconstruction efficiency.

The AI-driven track reconstruction was tested on existing experimental data (shown in Figure 9. The same final state was studied in experimental data (collected at standard experimental conditions of 45 *nA* beam incident on hydrogen target), and the results show a 48% increase in statistics.

## 4 Summary

The track reconstruction in CLAS12 currently uses AI in data reconstruction workflow. "The increased efficiency of track reconstruction (identification) efficiency, using de-noising and tracking classifiers increases yields from CLAS12 experiments up to 50% for multi-particle final states, improving physics outcomes and will allow running future experiments at higher luminosities due to better cluster identification after the de-noising procedure

## 5 Acknowledgments
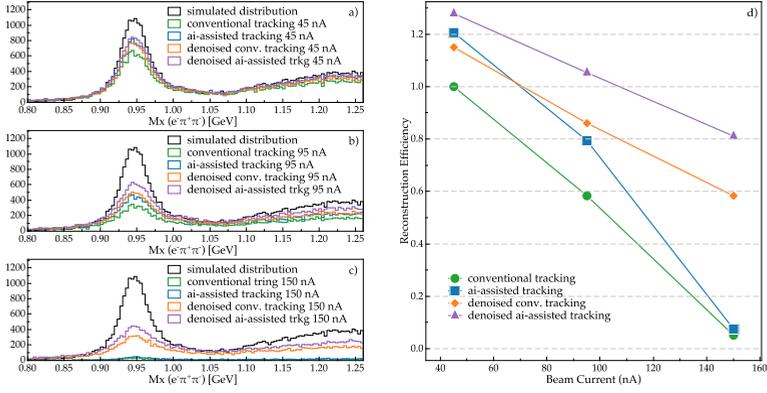
**Figure 8.** The de-noised data sample was reconstructed with an AI-assisted tracking algorithm (triangles) for 45 *nA*, 95 *nA*, and 150 *nA*. a), b) and c) reconstructed missing mass distributions for background merged data set reconstructed with conventional tracking (filled histogram) and de-noised data sample reconstructed with AI-assisted algorithm (solid line histogram). Missing mass distribution for data sample before background merging (0 *nA*) is shown (circles) for reference. The number of reconstructed protons from missing mass of $H(e \rightarrow e'\pi^+\pi^-)X$ for background merged data set reconstructed with conventional tracking (circles) compared to de-noised data sample reconstructed with conventional algorithm (diamonds) d).
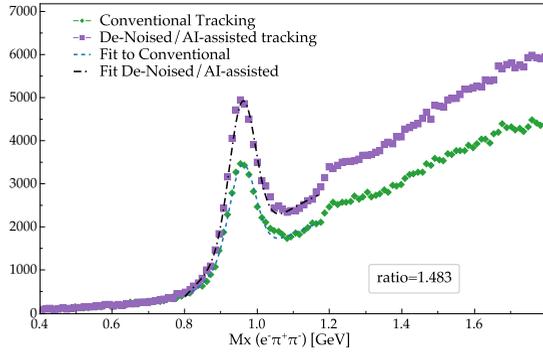


**Figure 9.** The number of reconstructed protons from missing mass of $H(e \rightarrow e'\pi^+\pi^-)X$ from experimental data , 45 nA beam incident on hydrogen target, is plotted to compare the number of reconstructed proton final stated from conventional tracking algorithm with number of protons from de-noised AI-assisted tracking.

# References

[1] Burkert, V.D. and others, The CLAS12 Spectrometer at Jefferson Laboratory, Nucl. Instrum. Meth. A **959**,163419 (2020)

[2] "Mestayer, M.D. and others", The CLAS12 drift chamber system, Nucl. Instrum. Meth. A",**959** 163518 (2020)

[3] Kalman, R. E., A New Approach to Linear Filtering and Prediction Problems, Journal of Basic Engineering, **82**, 35-45 (1960)

[4] Gavalian, Gagik and Thomadakis, Polykarpos and Angelopoulos, Angelos and Ziegler, Veronique and Chrisochoides, Nikos, Using Artificial Intelligence for Particle Track Identification in CLAS12 Detector, **2008.12860**, (2020)

[5] Gavalian Gagik,Auto-encoders for Track Reconstruction in Drift Chambers for CLAS12, **2009.05144**, (2020)

[6] Thomadakis, Polykarpos and Angelopoulos, Angelos and Gavalian, Gagik and Chrisochoides, Nikos, De-noising drift chambers in CLAS12 using convolutional autoencoders, **271**, 108201 (2022)