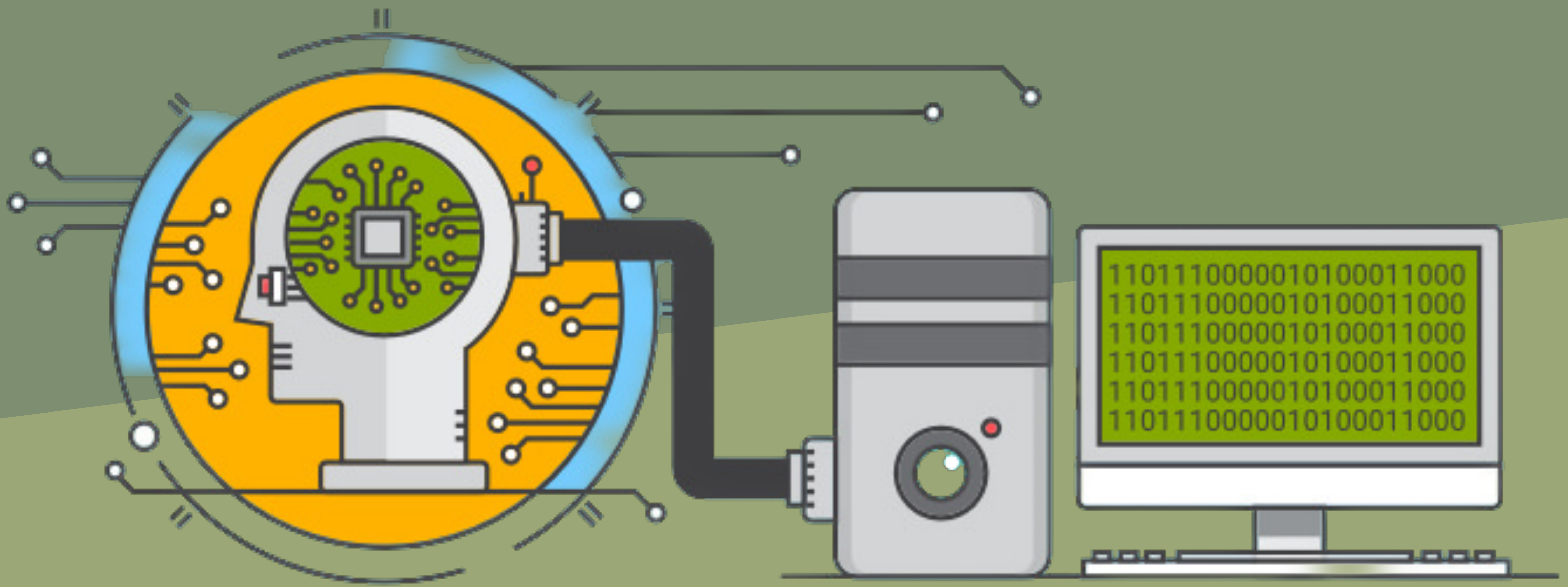
Artificial Intelligence in CLAS12

Artificial Intelligence/Machine Learning for Physics Applications

G.Gavalian (Jefferson Lab)

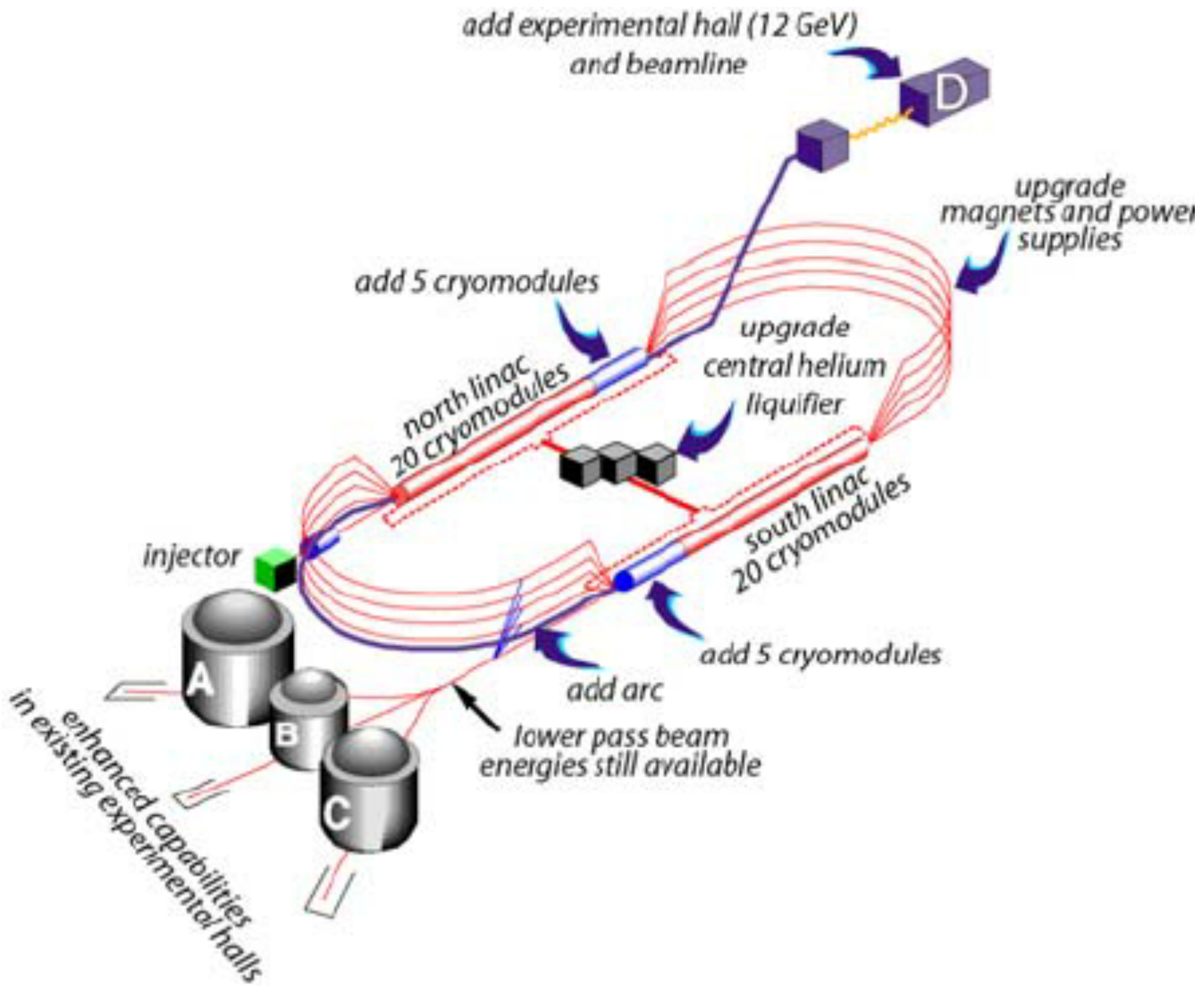


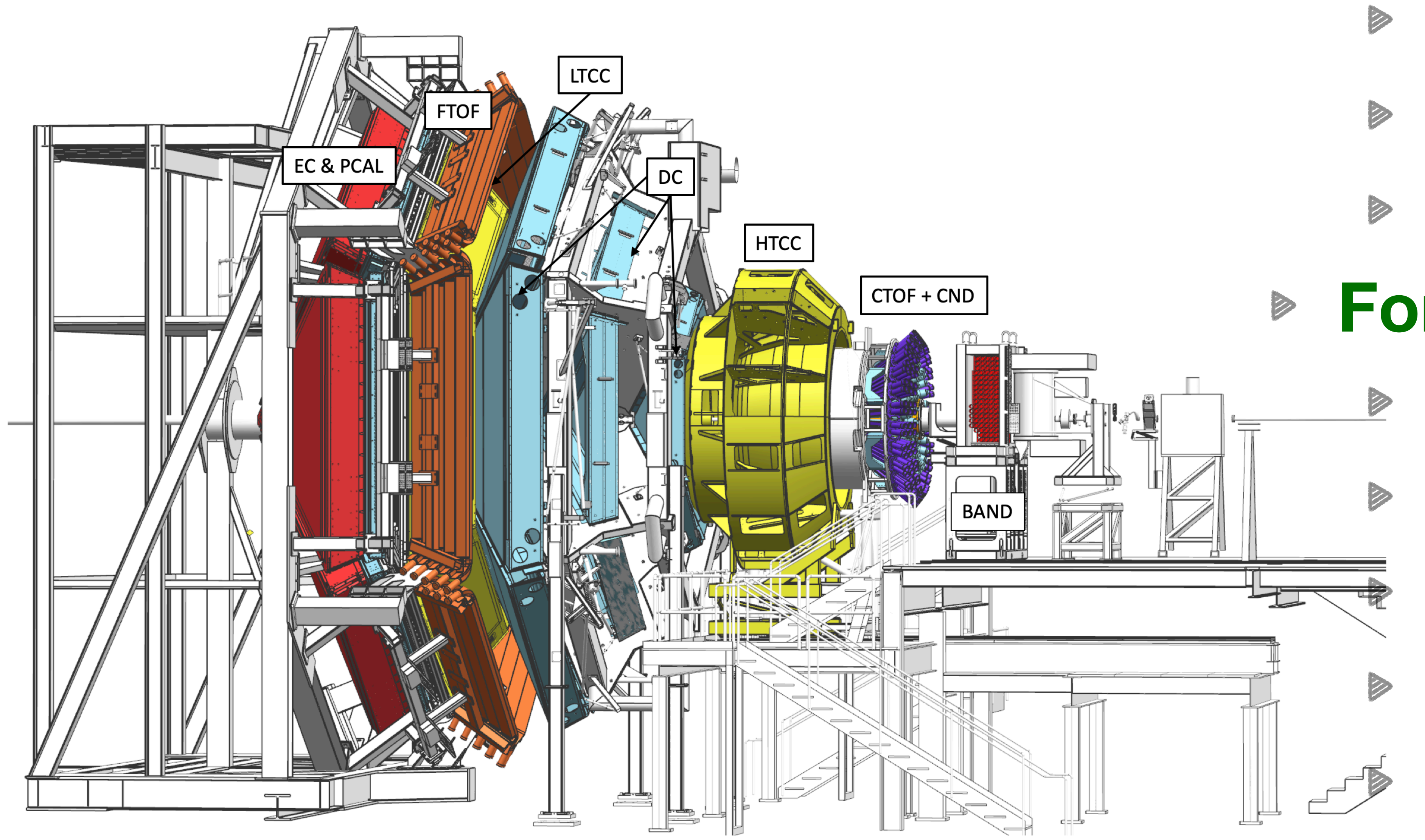
Richard Tyson (University of Glasgow)

Angelos Angelopoulos (CRTC)
Polykarpos Thomadakis (CRTC),
Nikos Chrisochoides (CRTC)
Department of Computer Science,
Old Dominion University, Norfolk, VA, 23529

AI4EIC (November 28,2023)

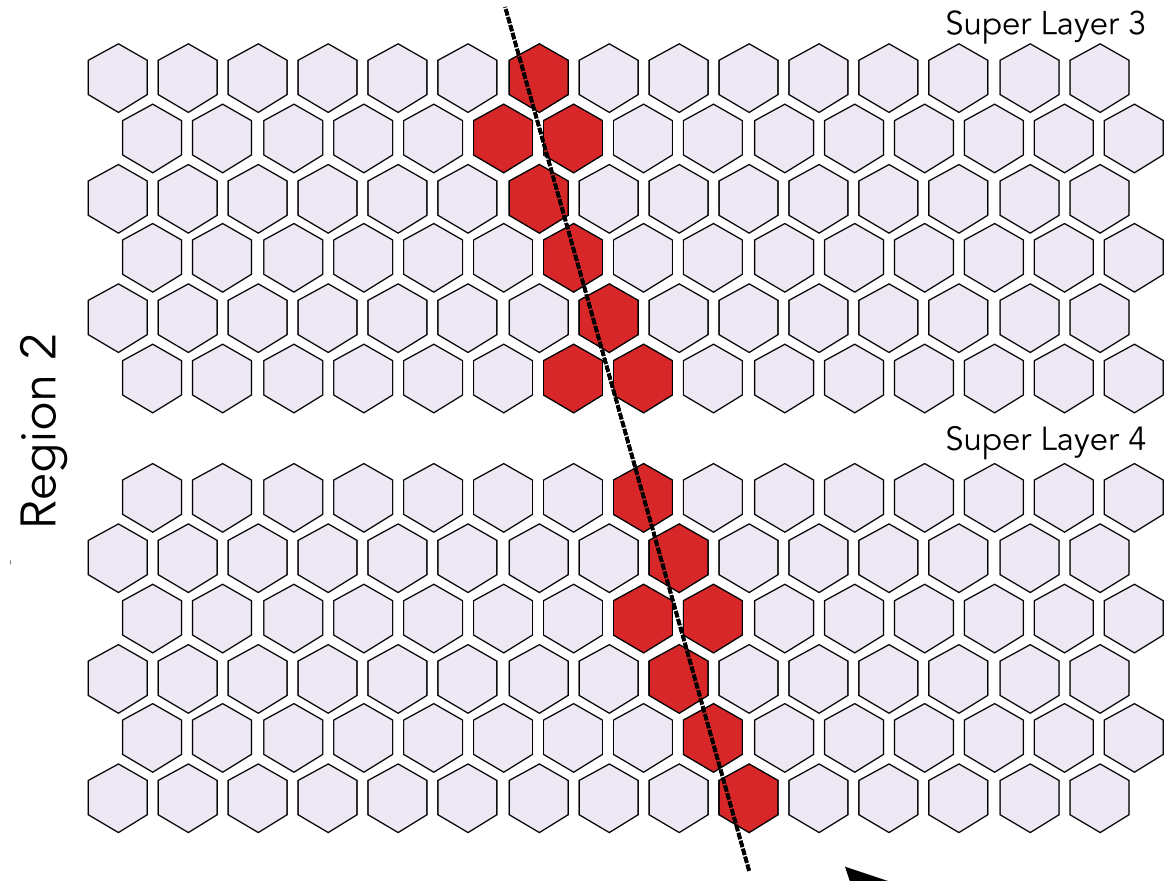
- Outline:
  - AI track identification in CLAS12
  - Implementation in the workflow and results
  - Track Parameter extraction using AI
  - Drift Chamber clustering using AI
  - Full track reconstruction online



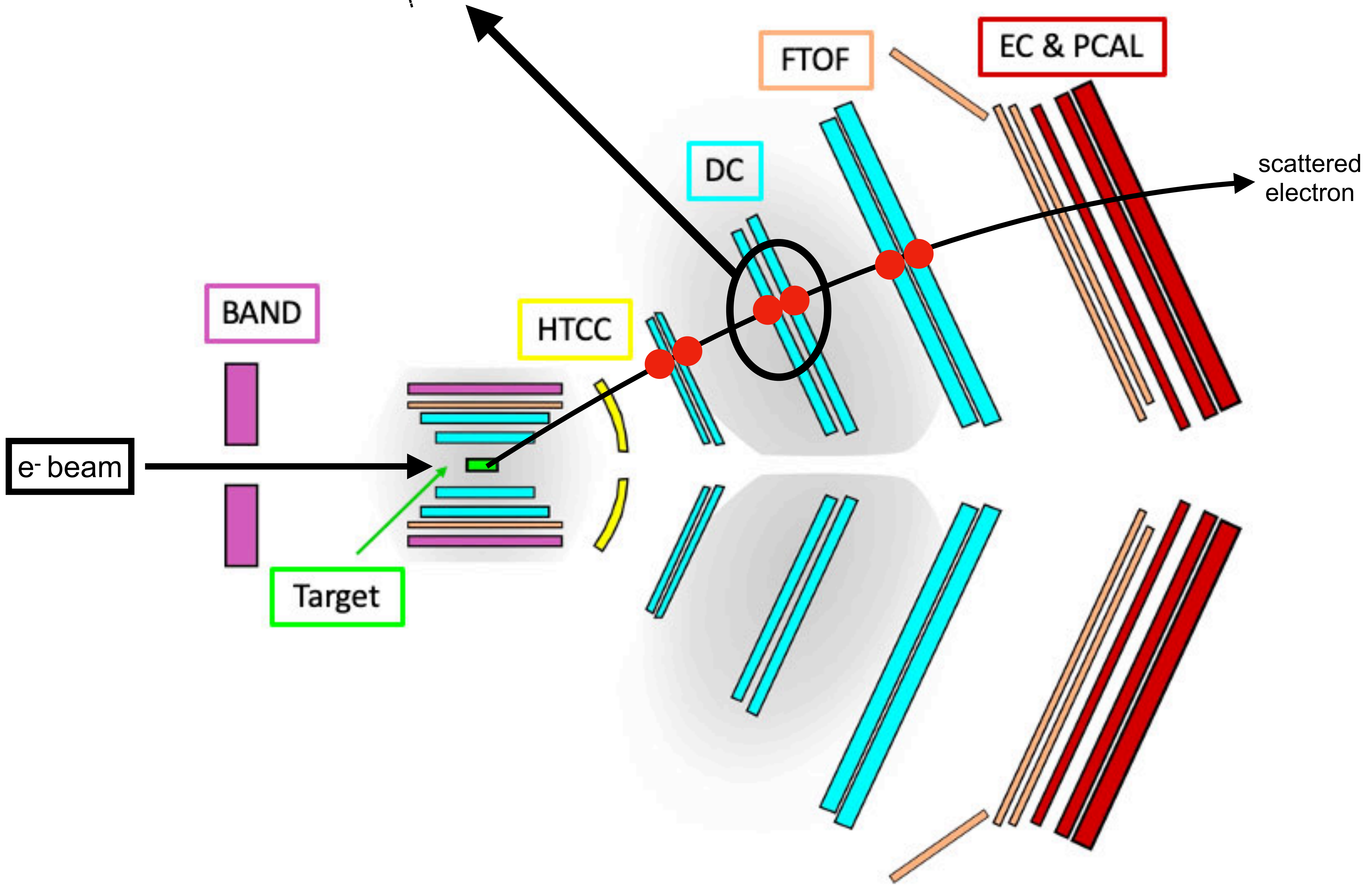


- CEBAF
  - 12 GeV electron beam distributed to 4 experimental hall
  - Each experimental hall contains a detector system for specific experiments
- **Hall-B:**
  - CEBAF Large Acceptance Spectrometer (CLAS12) Located in Hall-B
  - **Central Detector:**
    - Silicon Tracker
    - Time-Of-Flight
    - Neutron Detector
  - **Forward Detector:**
    - Drift Chambers
    - Time of Flight
    - High Threshold Cherenkov Counter
    - Ring Imaging Cherenkov Counter
    - Electromagnetic Calorimeter

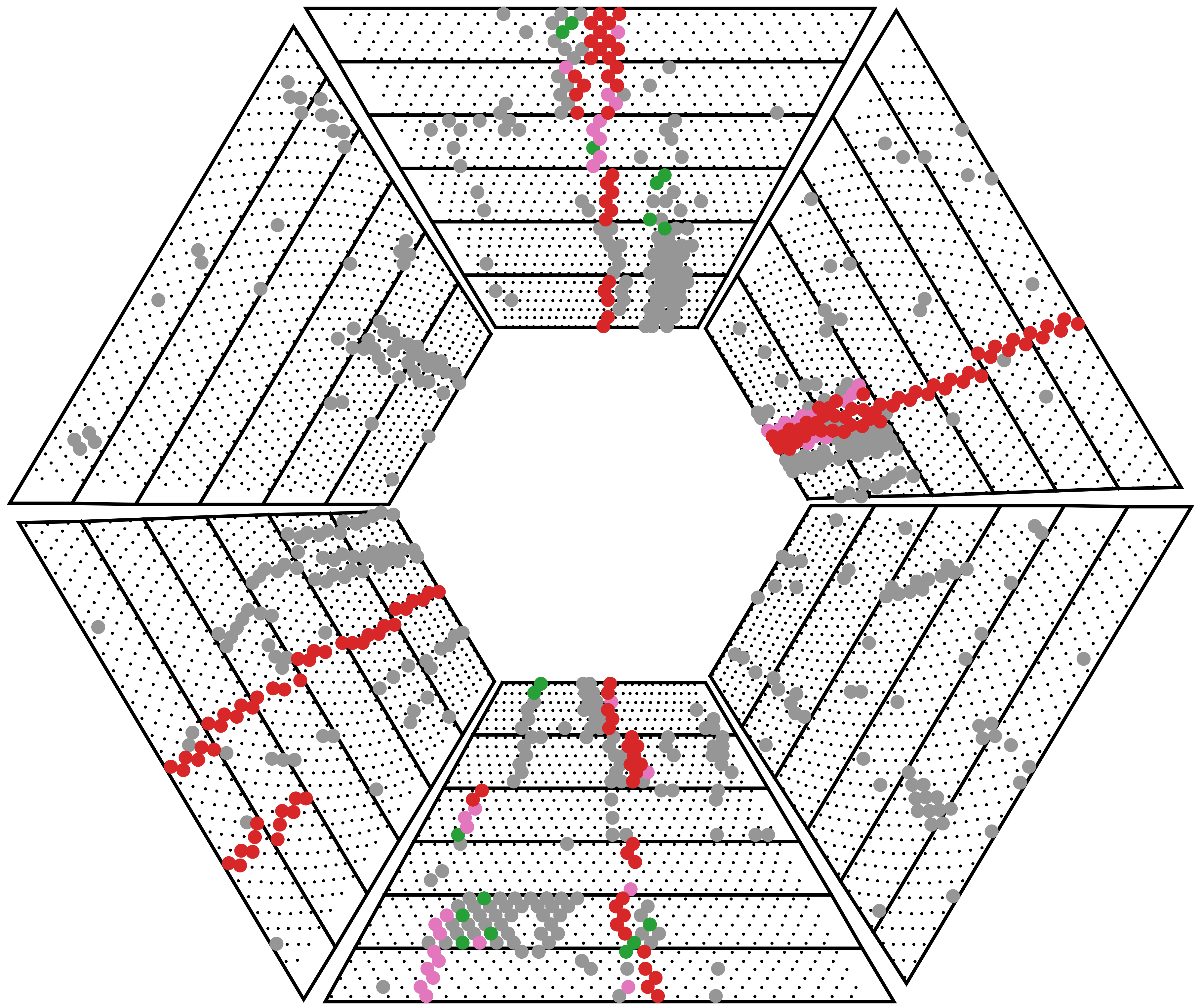
CLAS12 Detector



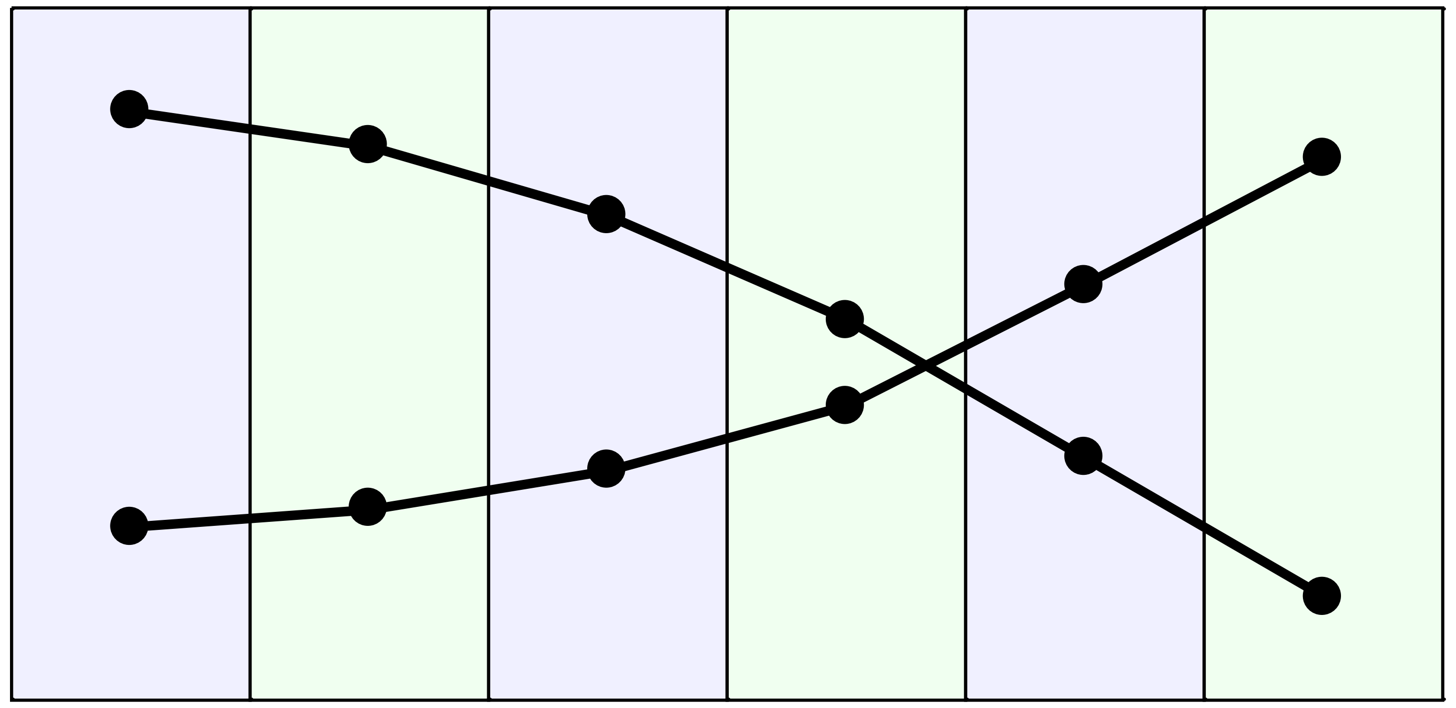
- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory



- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can speed up the tracking process and improve efficiency

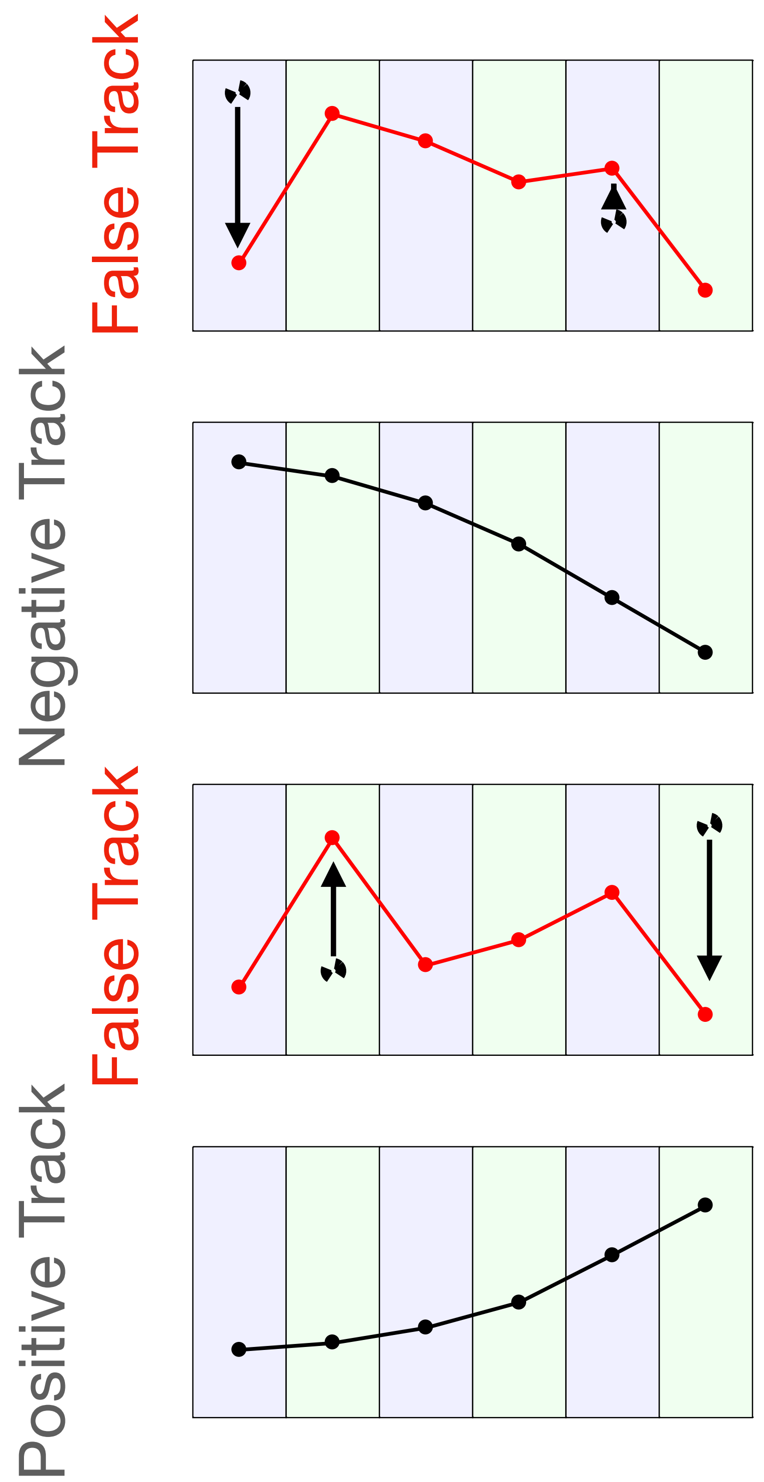


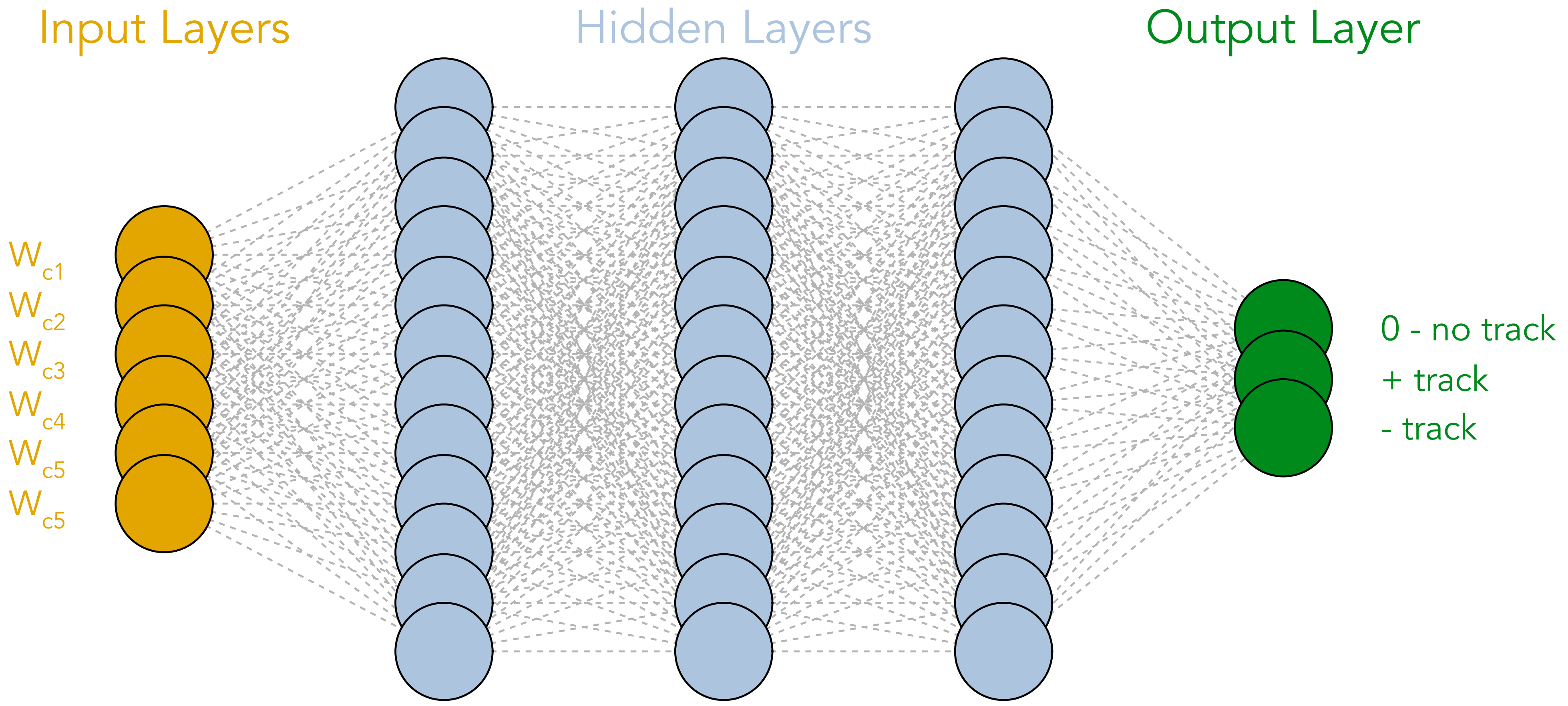
Physics Results



- True tracks are identified by conventional algorithms from real data.
- One negative and one positive track (different curvature due to magnetic field)
- False tracks are constructed by interchanging randomly one or two clusters with the clusters from the other track in the event
- Training sample balancing is done by choosing equal tracks for each momentum and angular bin.

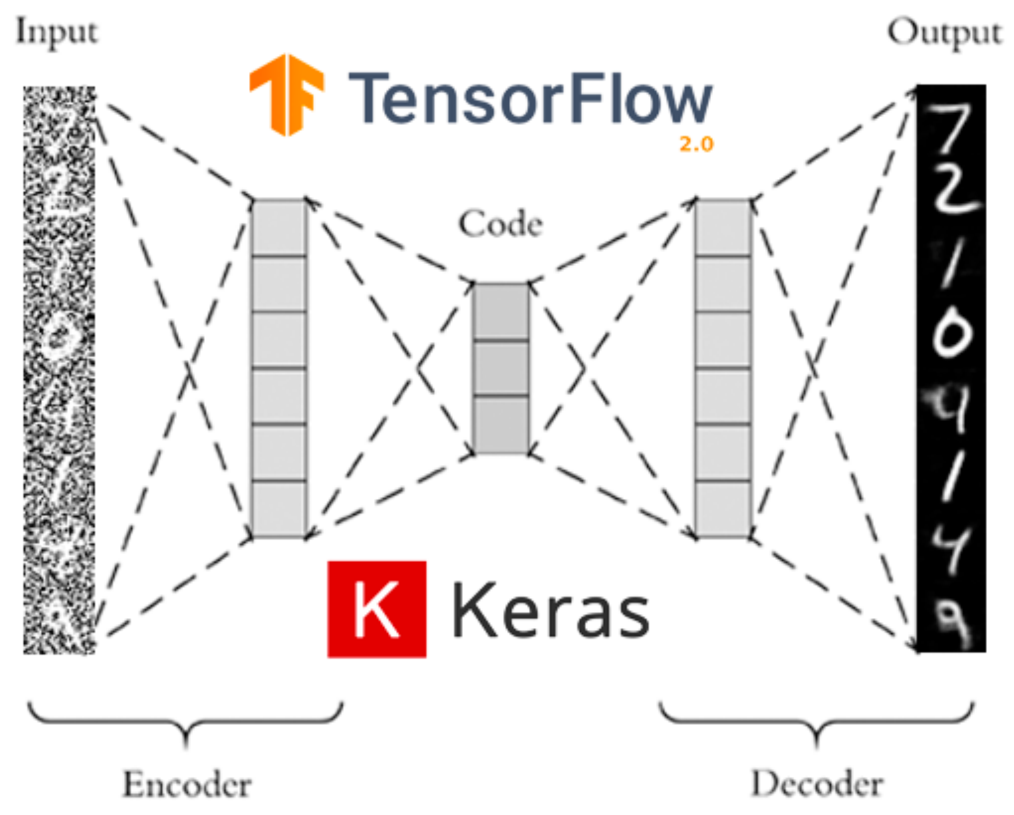
- The average wire position in each super layer is used as an input to Multi-Layer Perceptron (MLP)
- The network is trained on 6 inputs and produces three outputs:
  - False track
  - Negative Track
  - Positive Track



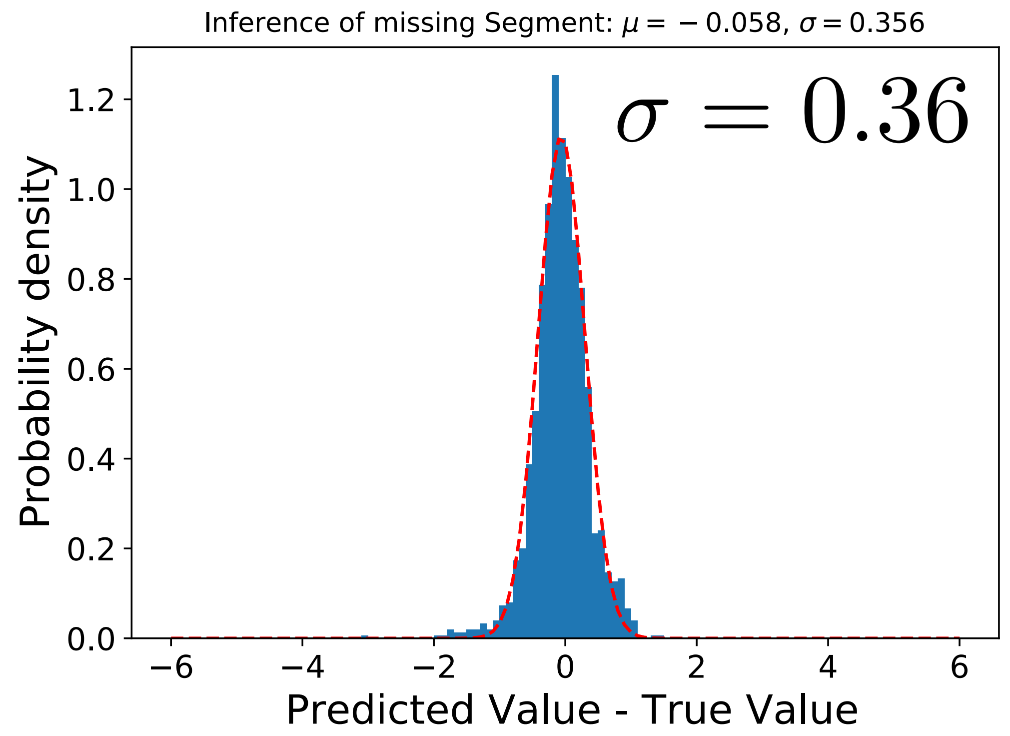


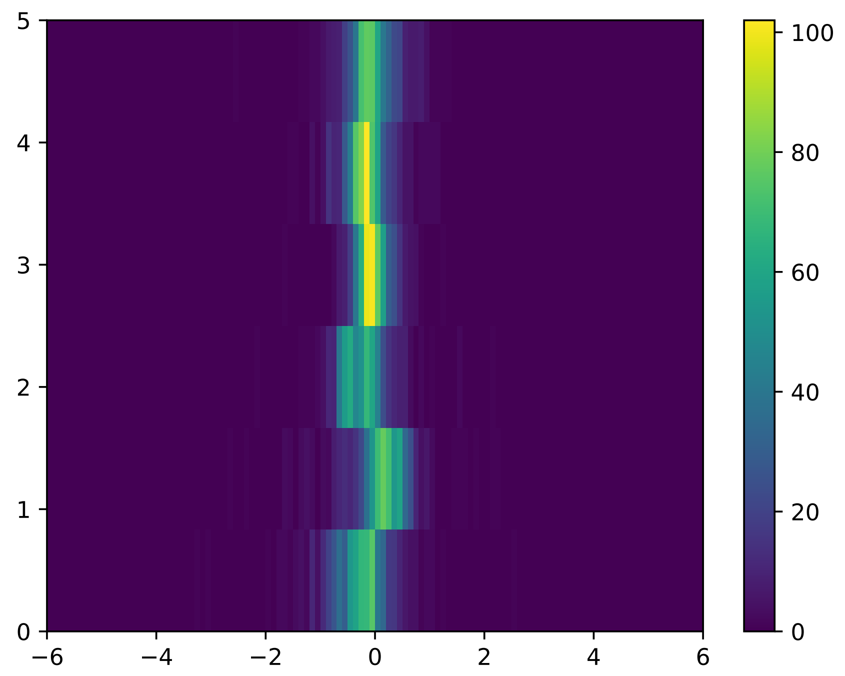
Corruption Auto-Encoder

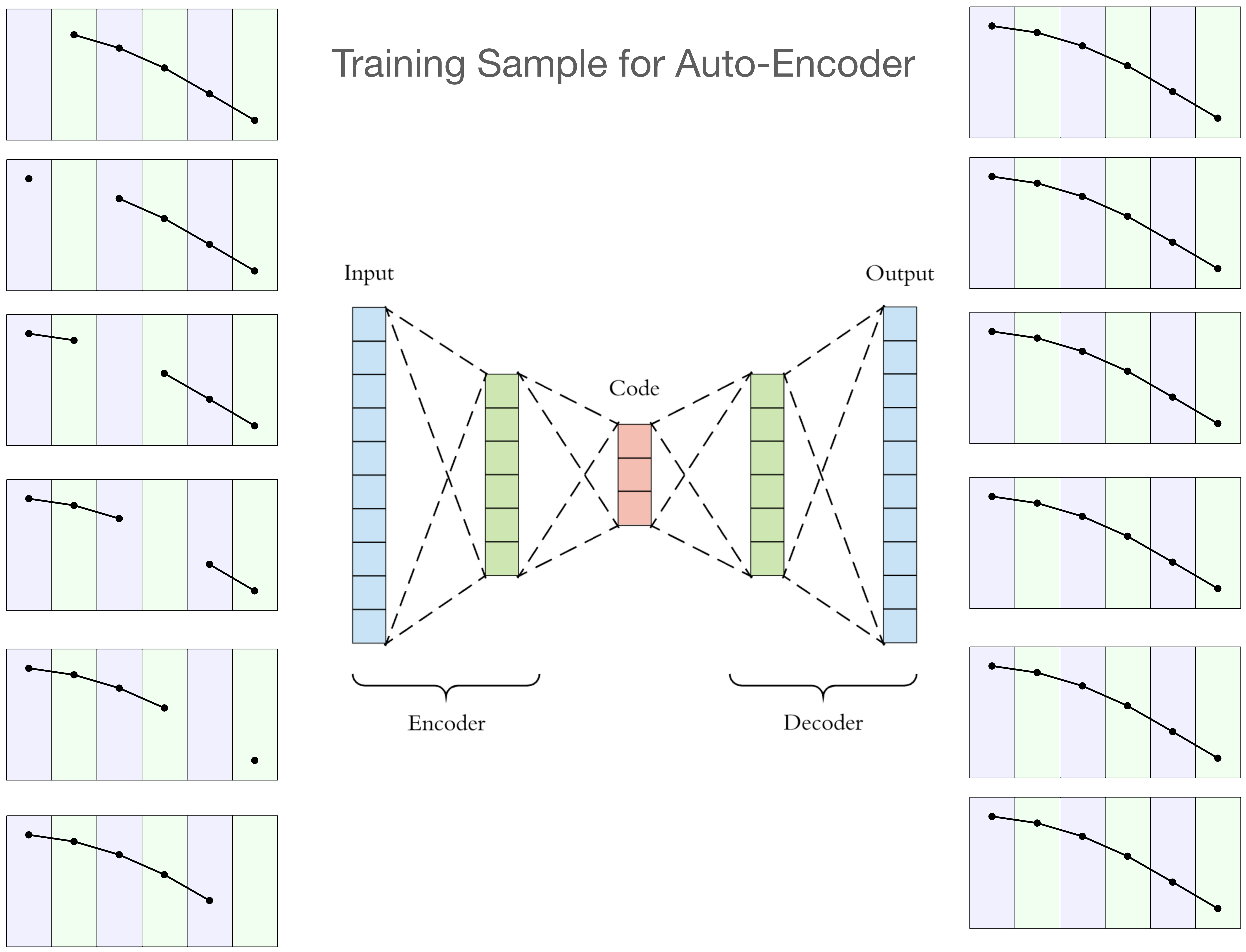
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- **Typically used for de-noising, but can be used for fixing glitches (our case).**



- The network Predicts the missing cluster position with a precision of 0.36 Wire

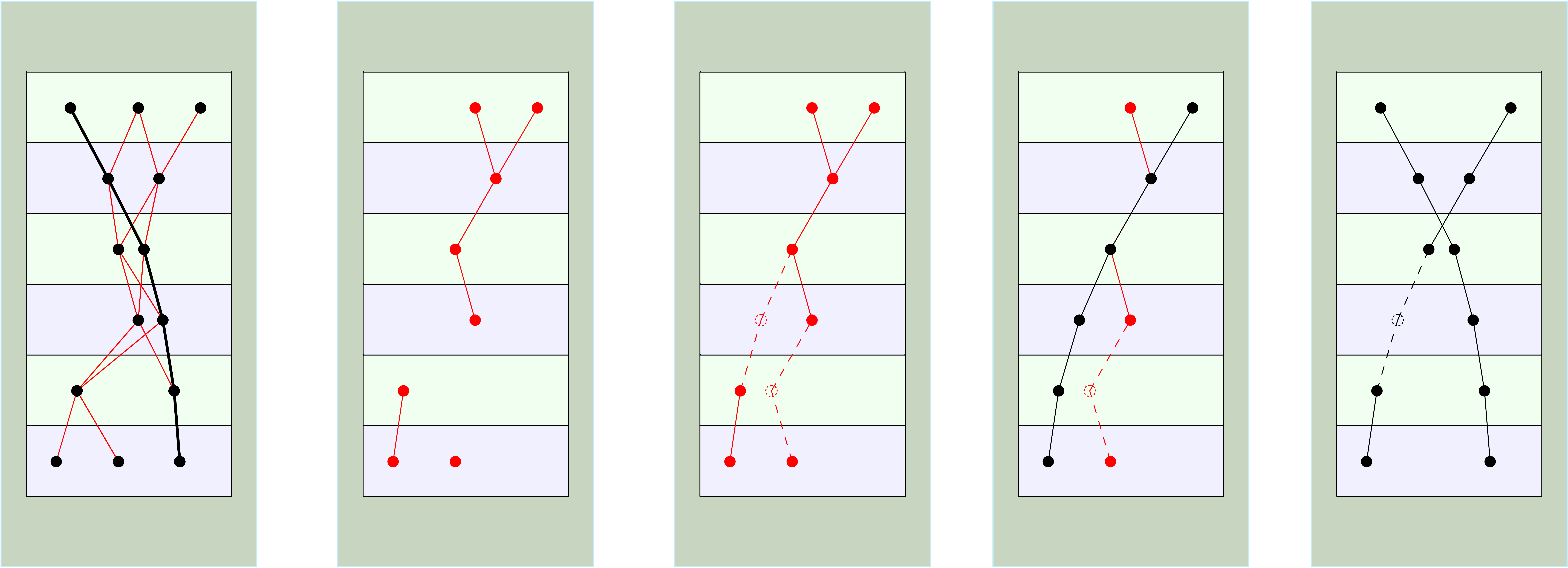






- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

Putting things together



Classifier picks the correct track from 6 super-layer combinations

Remove all clusters belonging to identified track

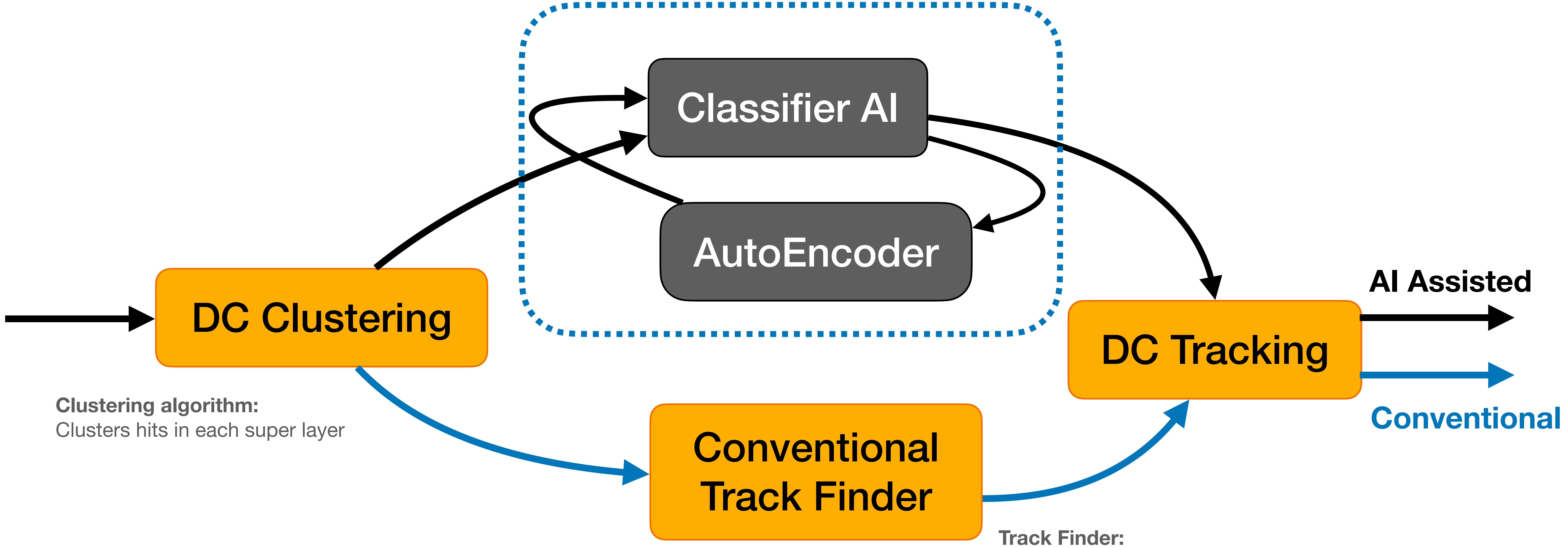
Construct pseudo-clusters for all 5 super layer combinations using Corruption Auto-Encoder

Identify tracks using 6 super layer candidates with pseudo-clusters

Voila!

The Algorithm

- CLAS12 Reconstruction Software is based on Service Oriented Architecture (SOA)
- Allows running parallel services for each algorithm producing common output.

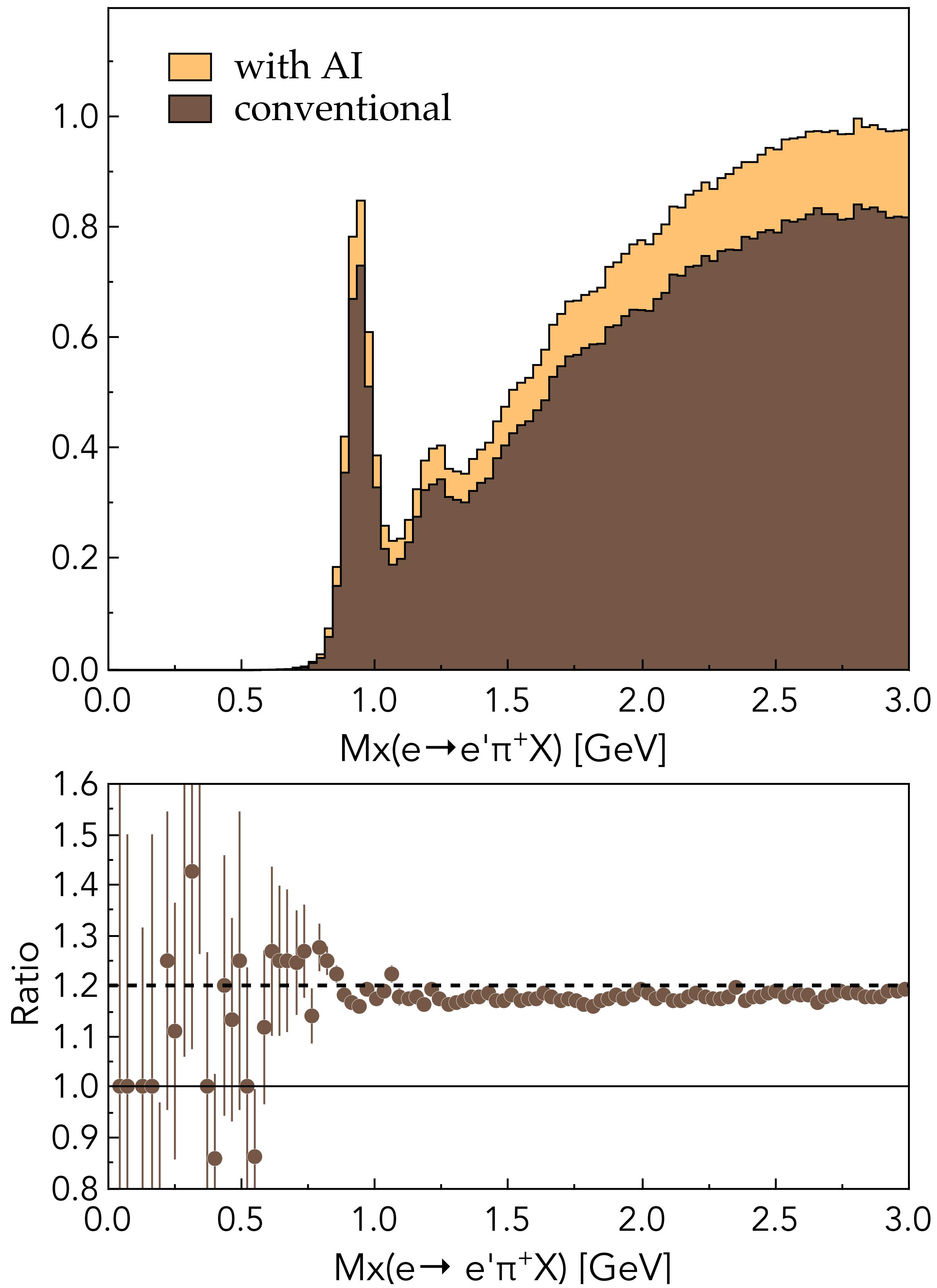


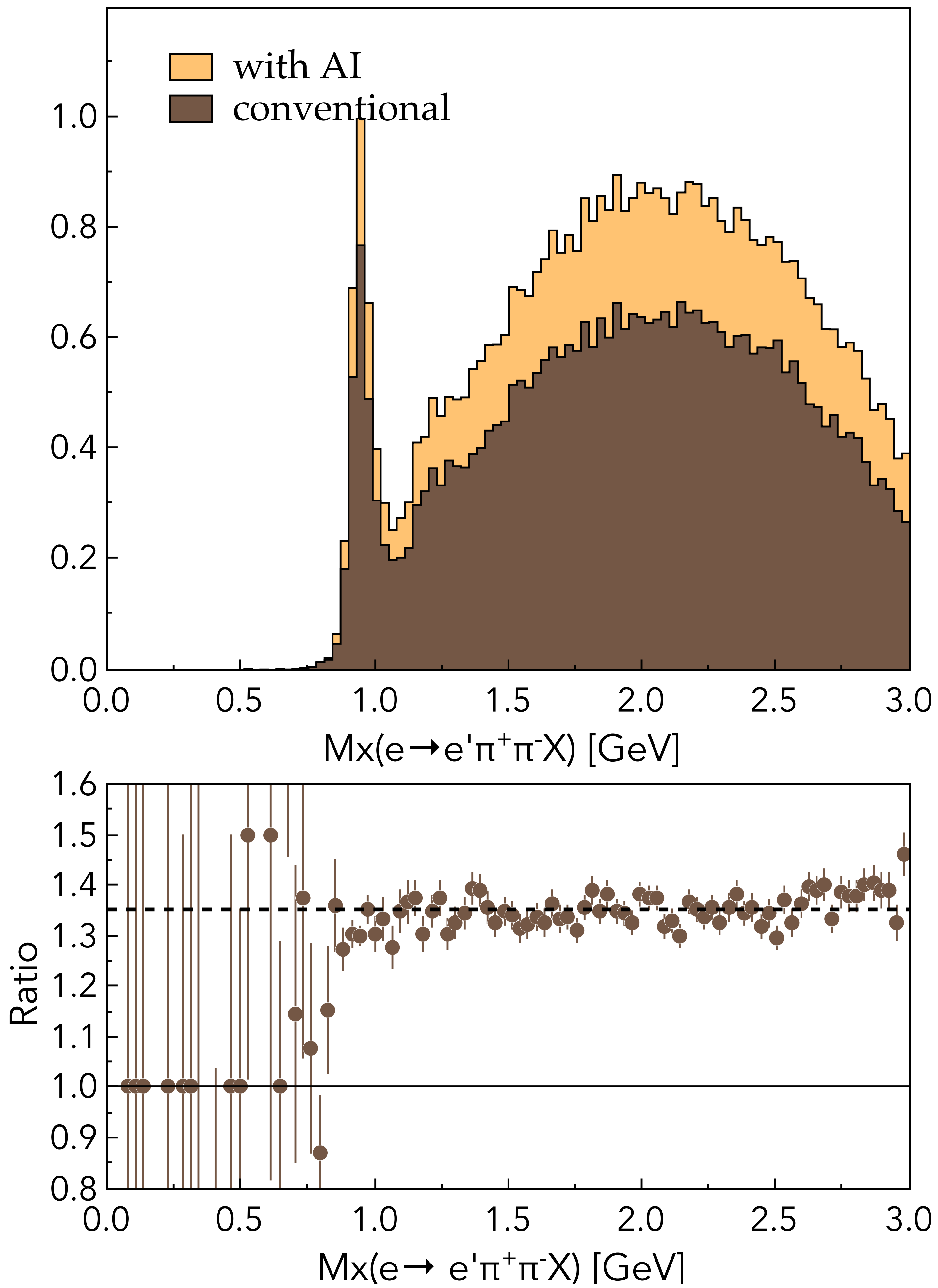
Clustering algorithm:
Clusters hits in each super layer

Track Finder:
Considers all combinations of 6 and 5 super layer track candidates Fits them and accepts/rejects them based on chi2.

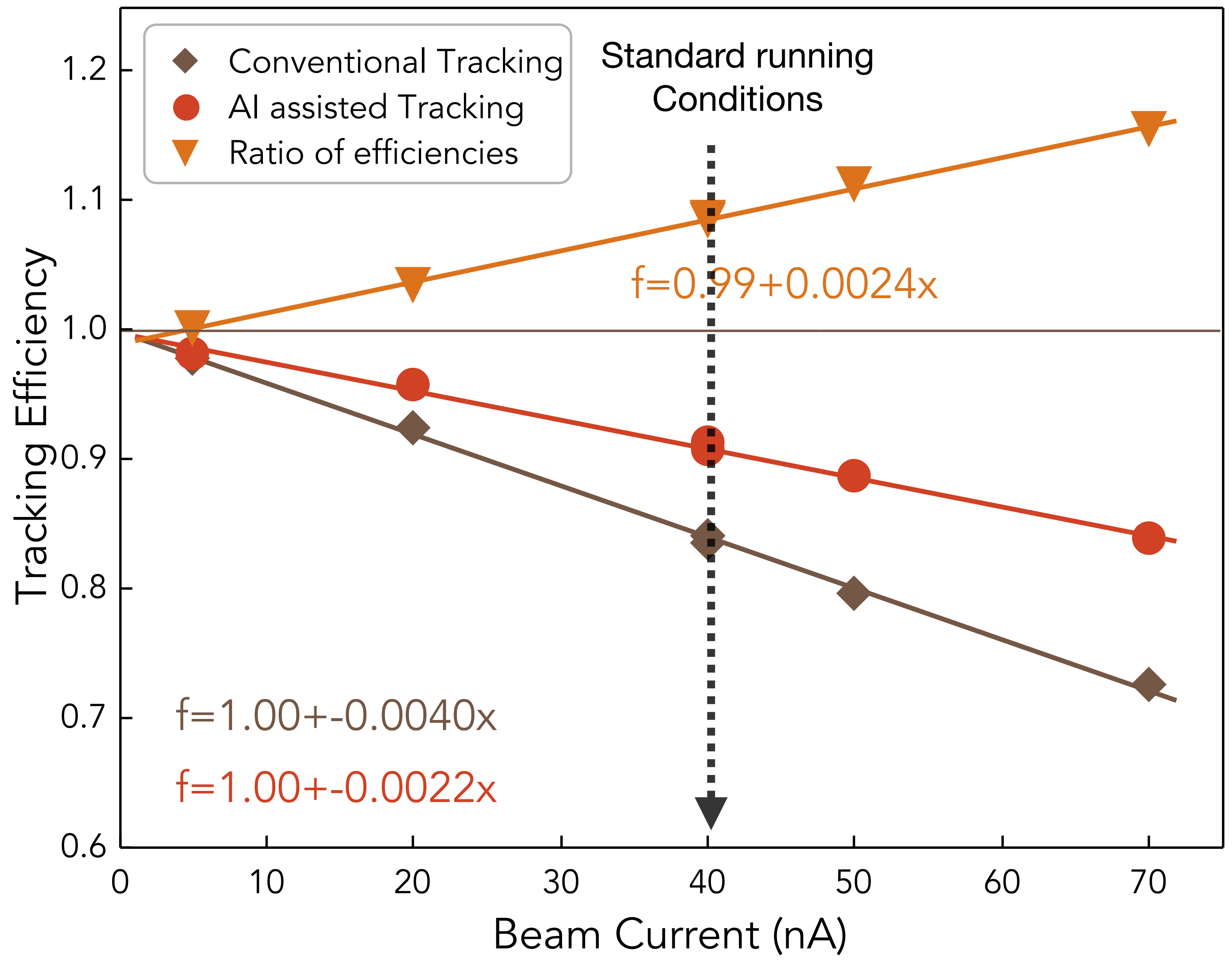
AI-assisted track candidate classification and Inefficiency Reduction Auto-Encoder

$ep \rightarrow e'\pi^+(X)$ $\qquad$ $ep \rightarrow e'\pi^+\pi^-(X)$





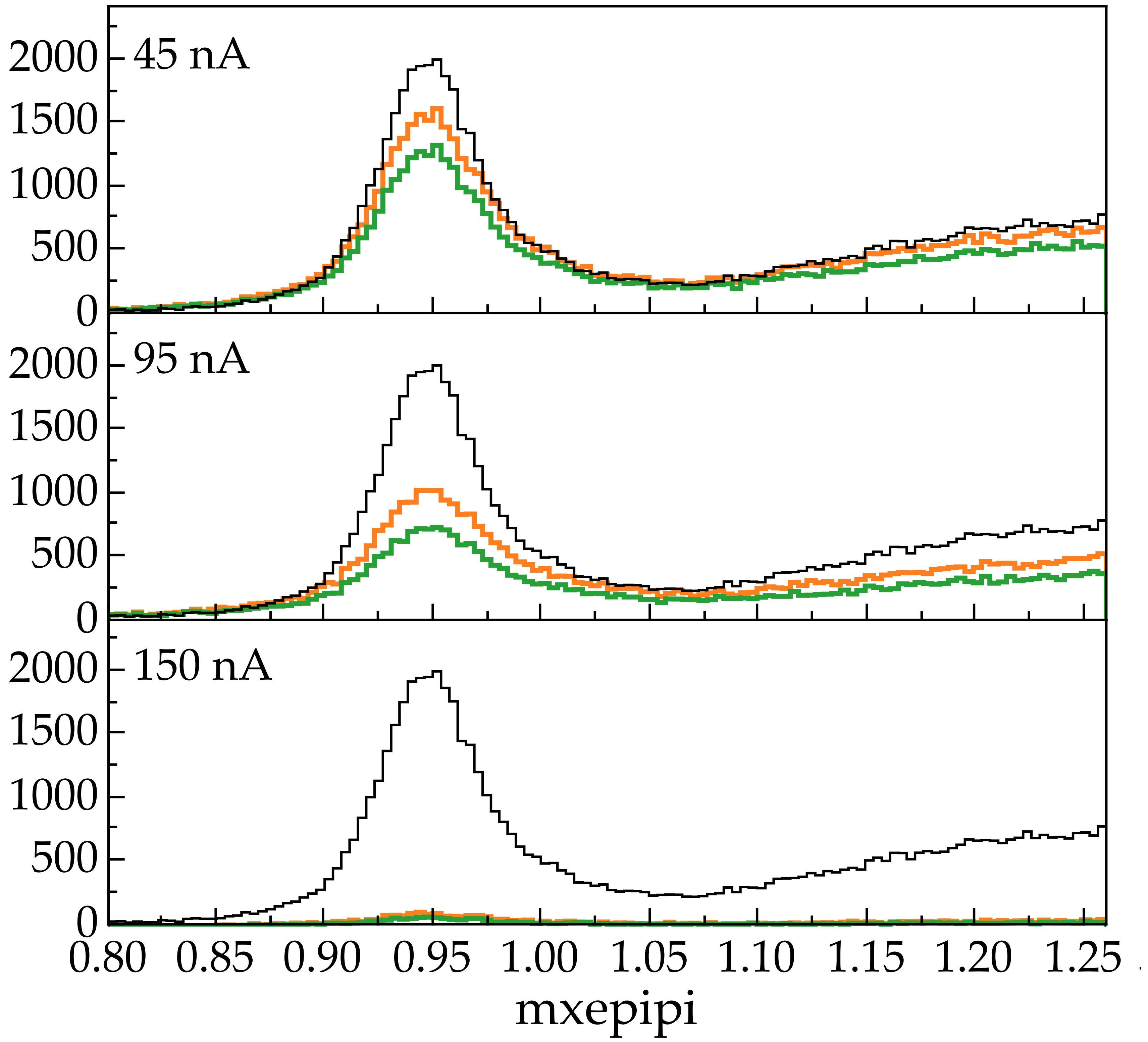
- Single particle efficiency increases by ~10%.
- The impact on physics for a multi-particle final state is dramatic (20% for the two-particle final state and ~35% for the three-particle final state)
- The tracking code speedup is ~30%.

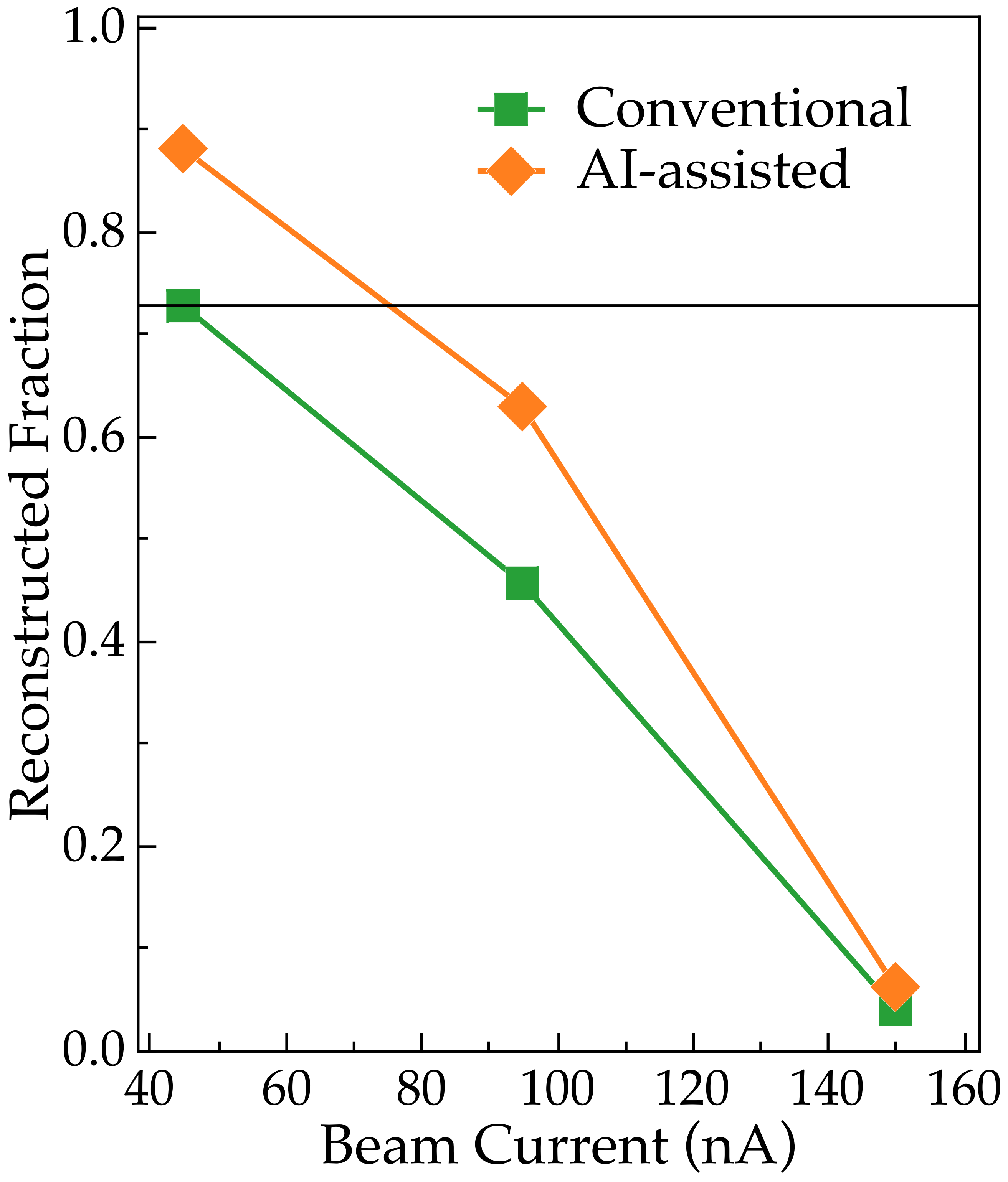


Up to ~35% gain in physics
Just using Classifiers

Moving to higher Luminosities

Performance of track identification for higher luminosity





- Pythia simulated physics reaction:

$$ep \to e'\pi^+\pi^-p$$

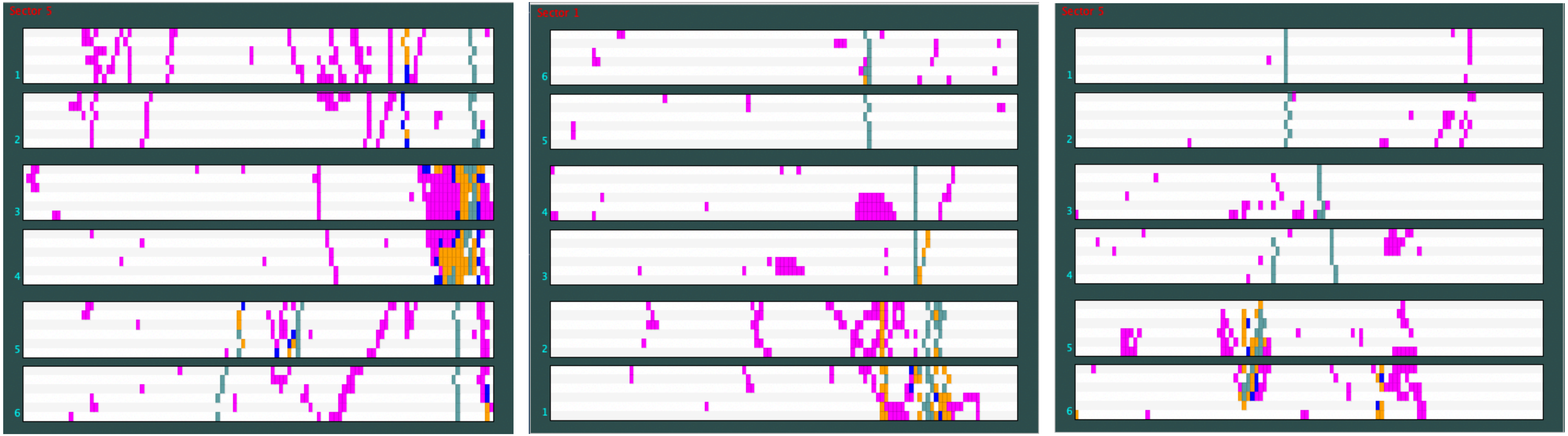
- Data for each luminosity (beam current) is created by standard background merging software.
- For each luminosity the yield of missing protons is calculated in:

$$ep \to e'\pi^+\pi^-X$$

- With increased luminosity the efficiency of reconstructed three particle final state drops sharply
- Even with the power of AI-assisted tracking (capable of resolving the combinatorics) the efficiency drop follows the same trend.

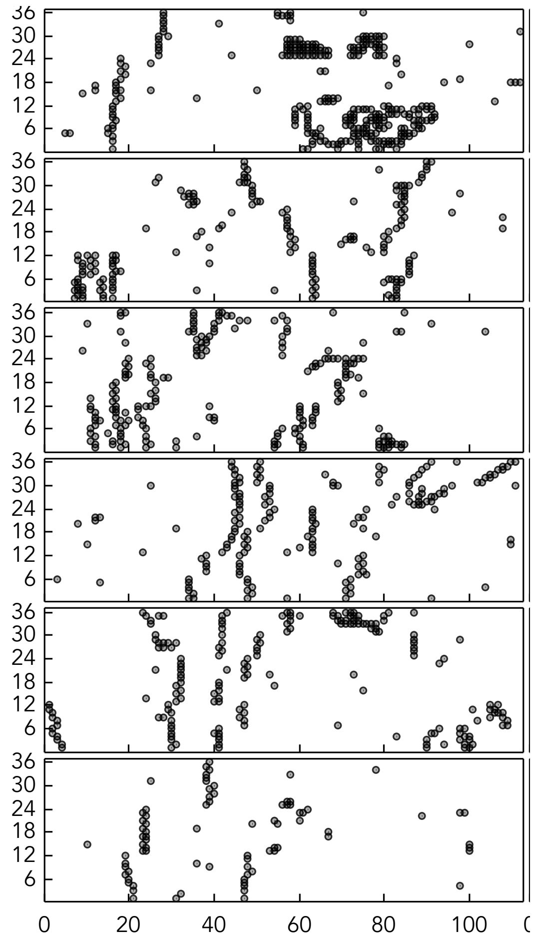
- In high luminosities the noise level increases and forming clusters (or segments in each chamber becomes challenging)
- This results in loss of clusters and AI-assited tracking can no longer help with combinatorics resolution

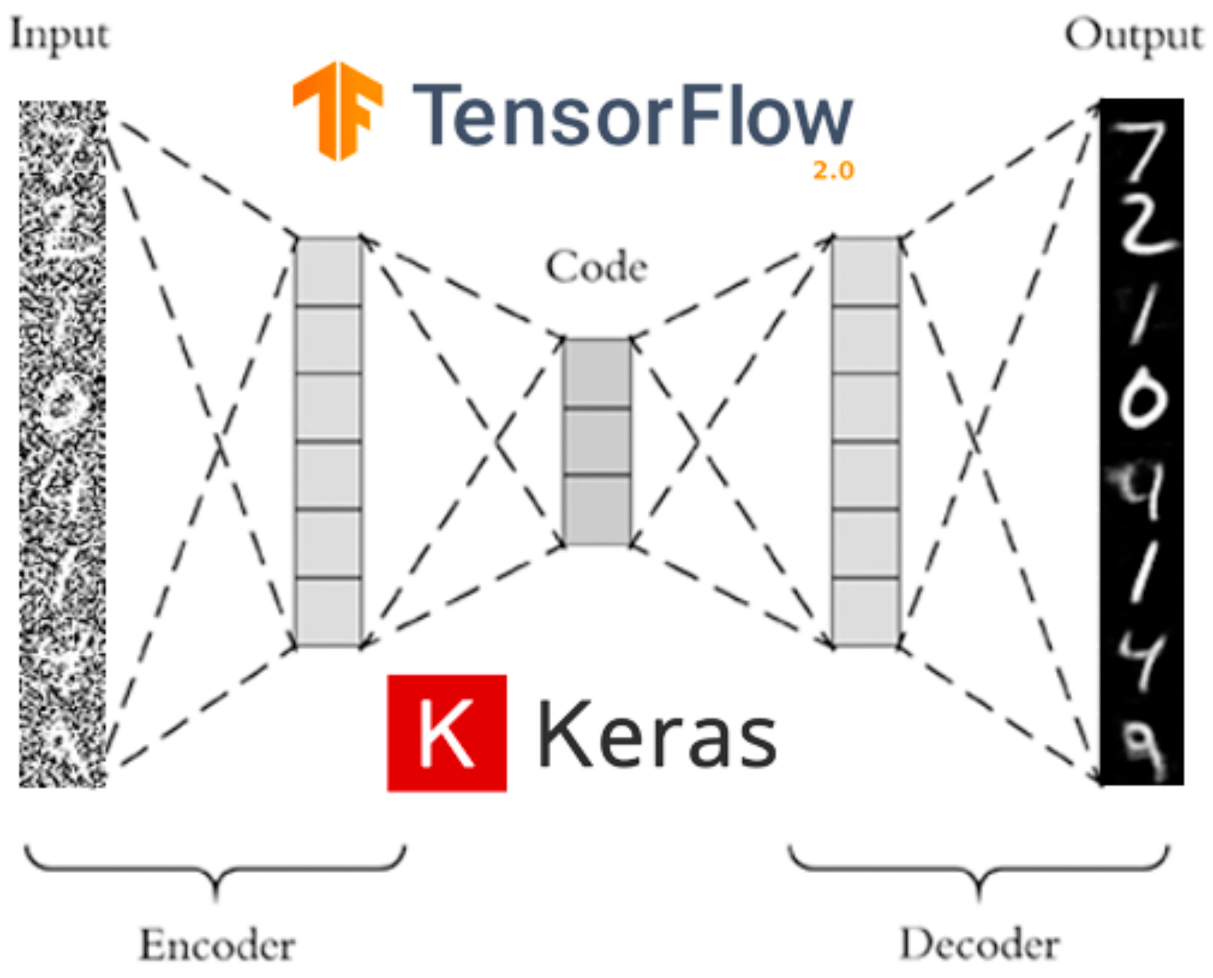
CLAS12 Event Display Examples (Drift Chambers)

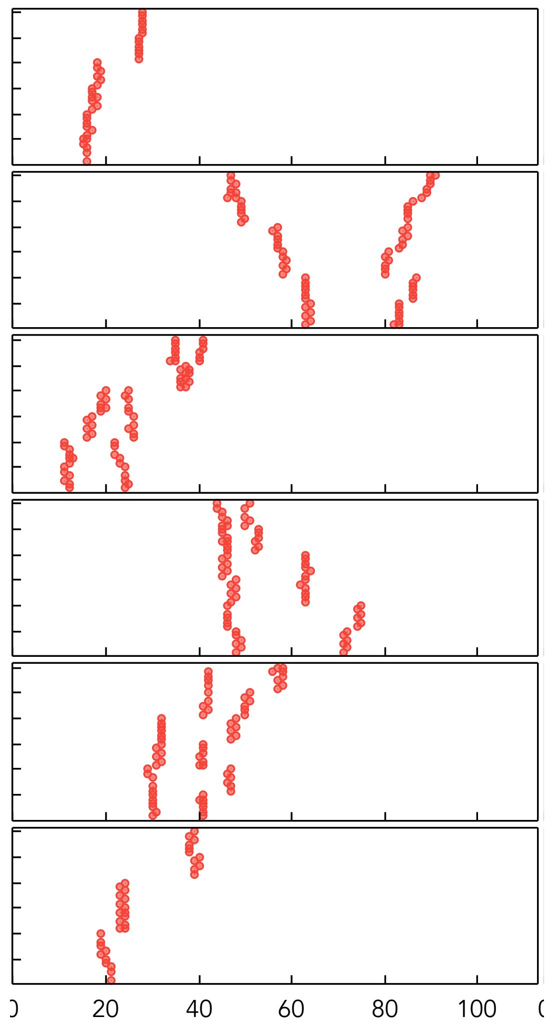


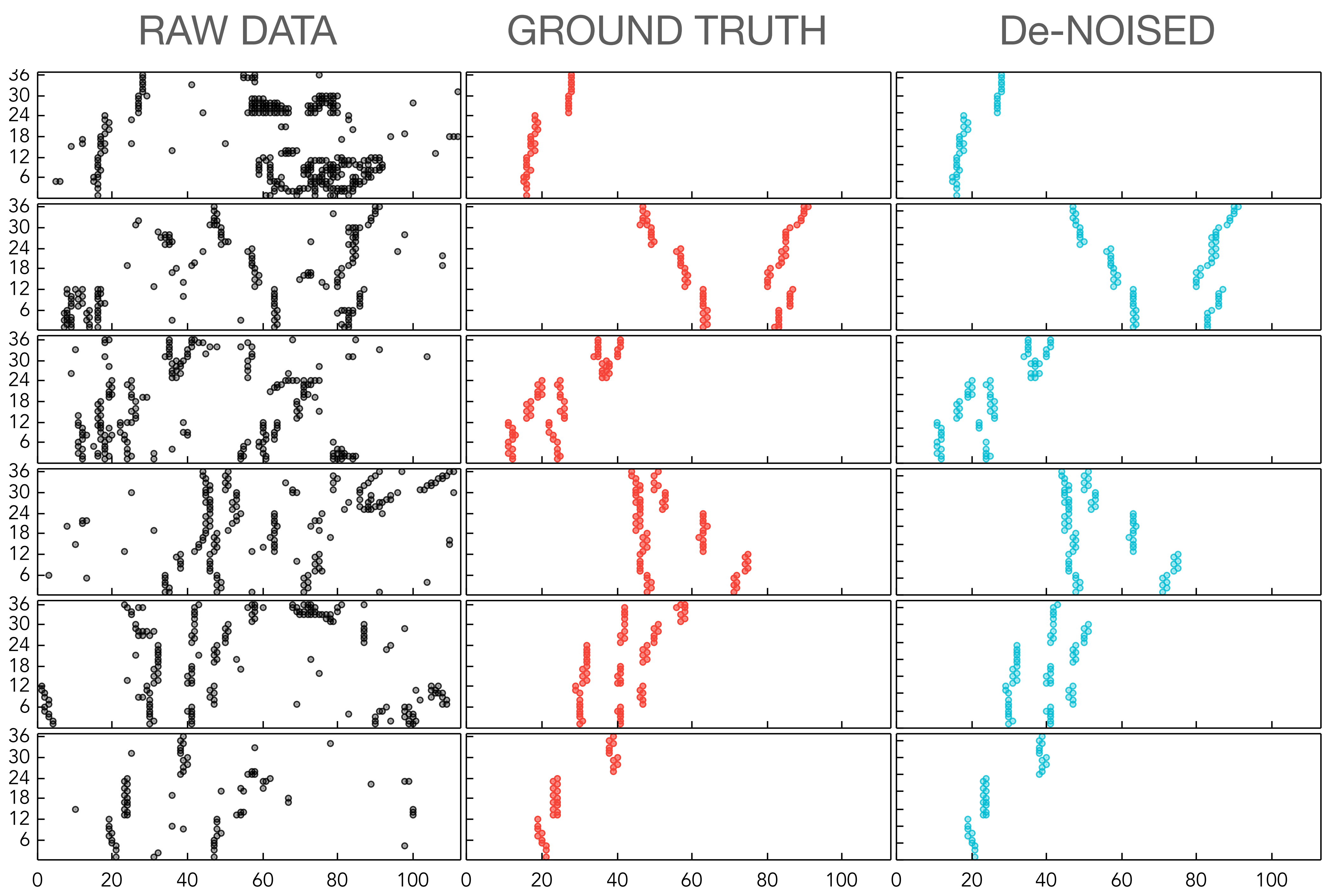
De-Noising

- Convolutional Auto-Encoder is used to de-noise raw data from drift chambers.
- The network is trained on reconstructed data with track hits isolated from raw DC hits.
- The network is able to isolate hits that potentially belong to a valid track through drift chambers

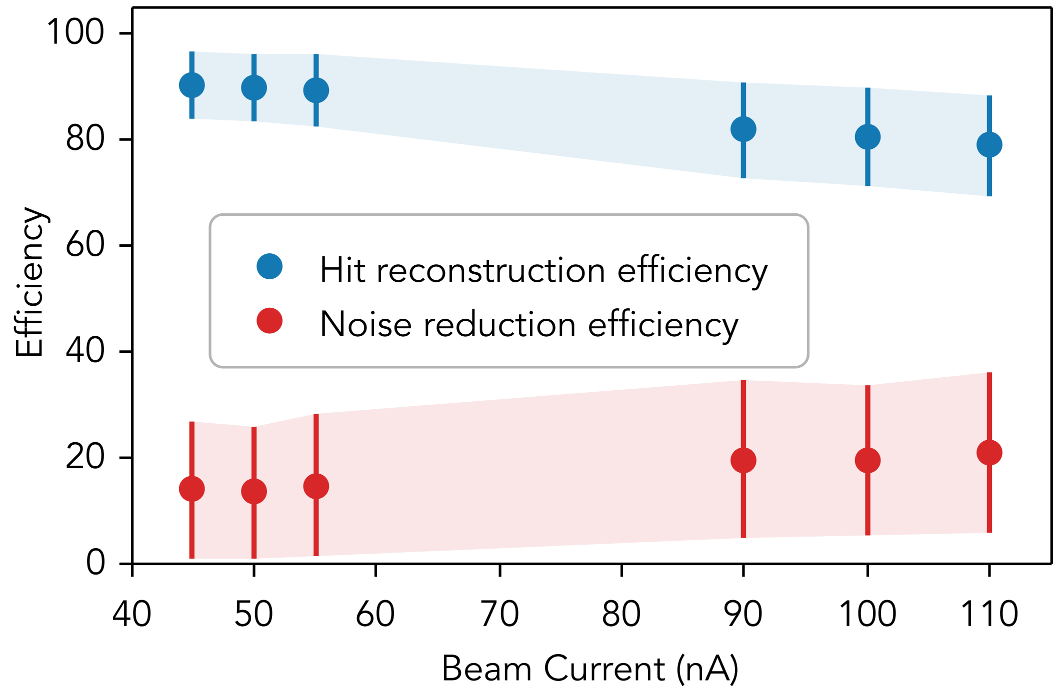


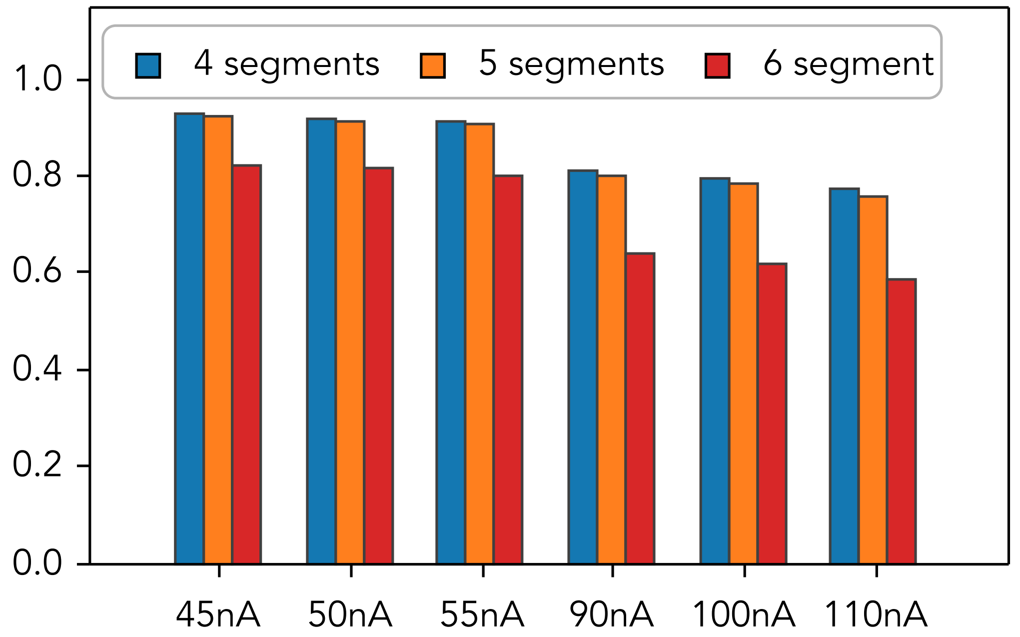






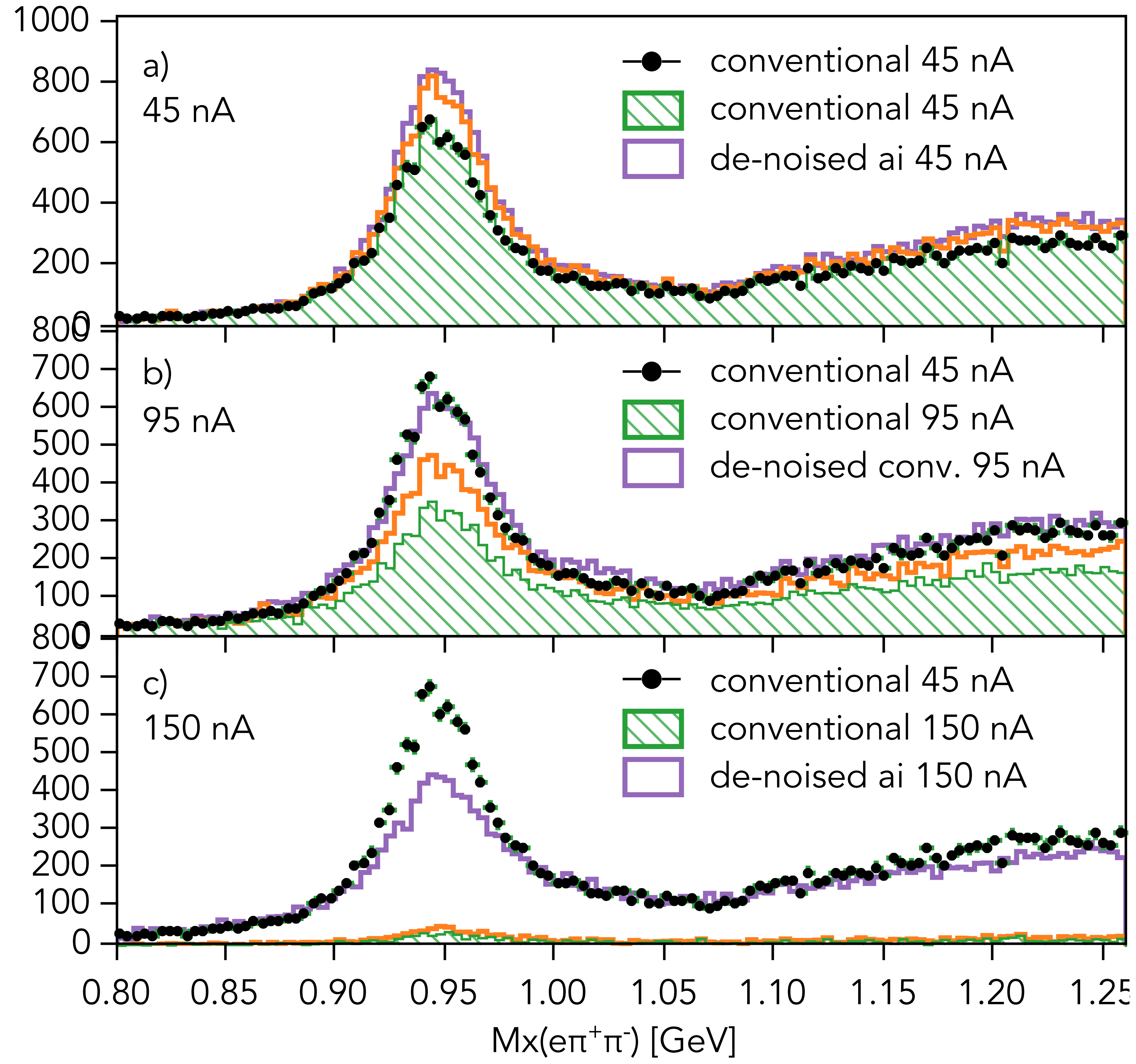
Network Performance Summary

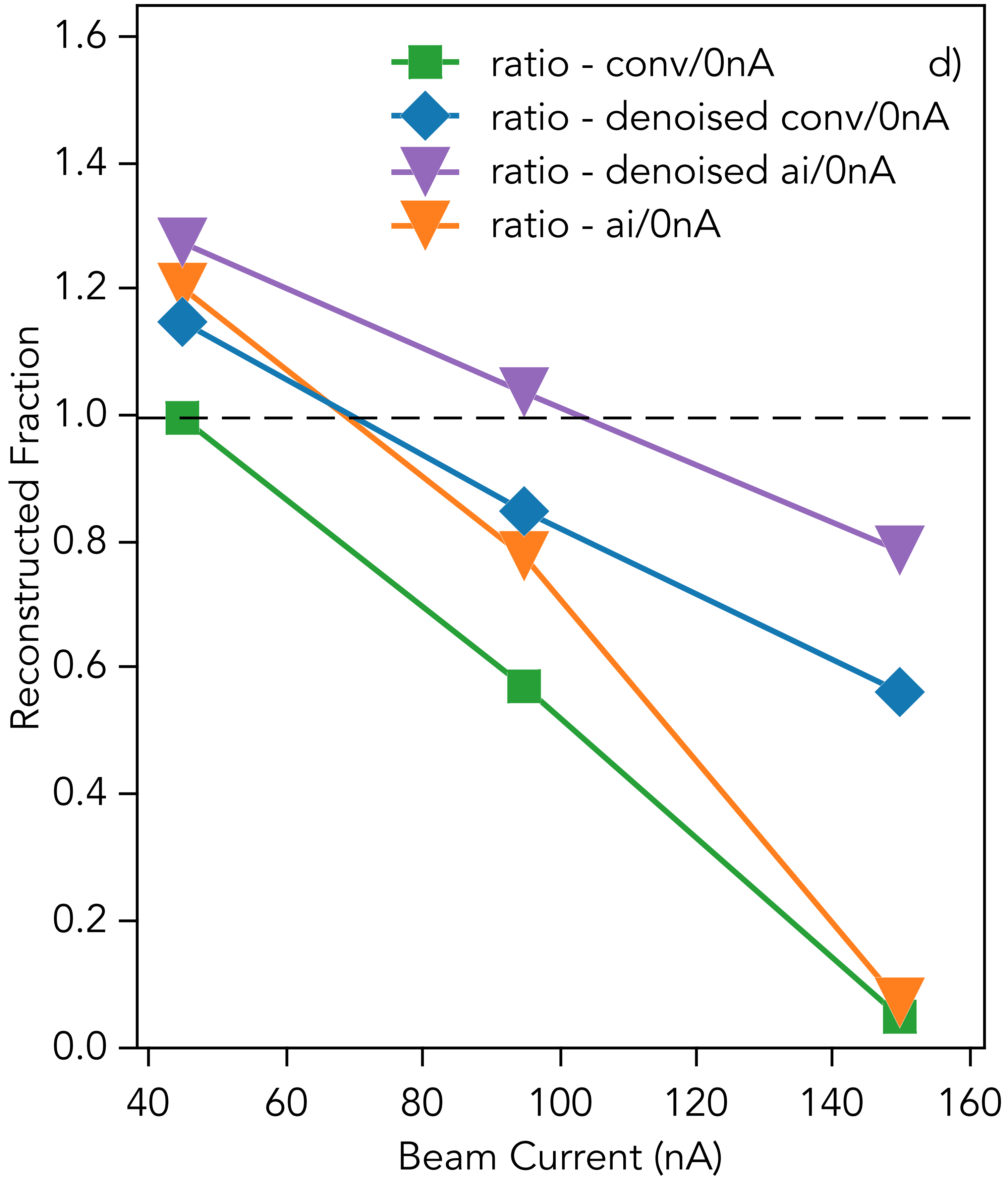




De-Noising Results (simulation)

- The reconstruction is run on simulated data with a merging background for different incident beam currents (luminosity)
- The simulated three-particle final state is analyzed to measure yield for de-noised data and for conventional

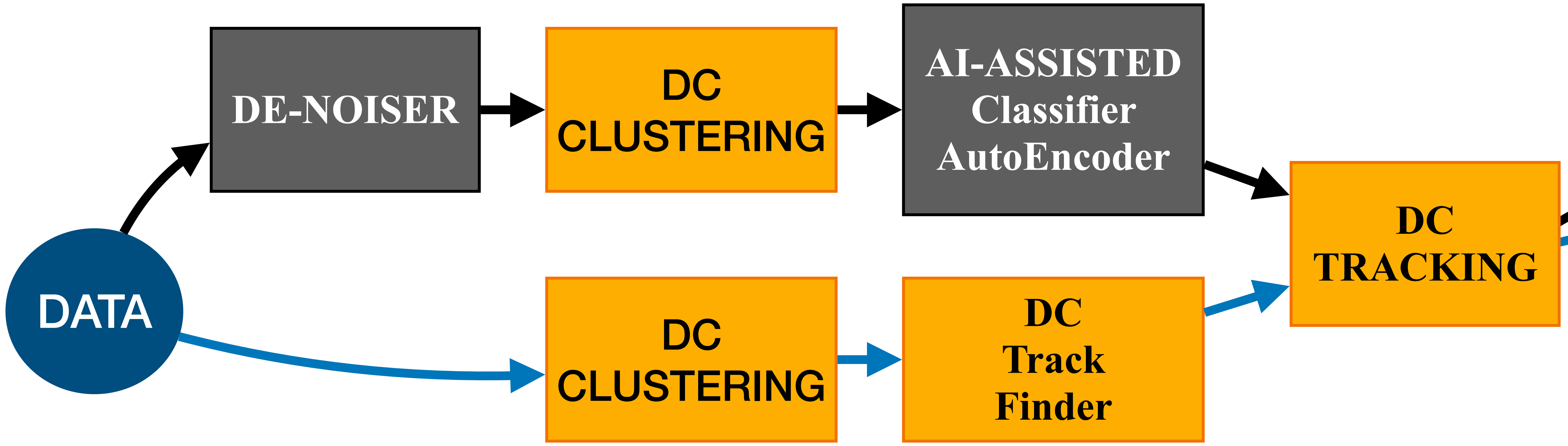


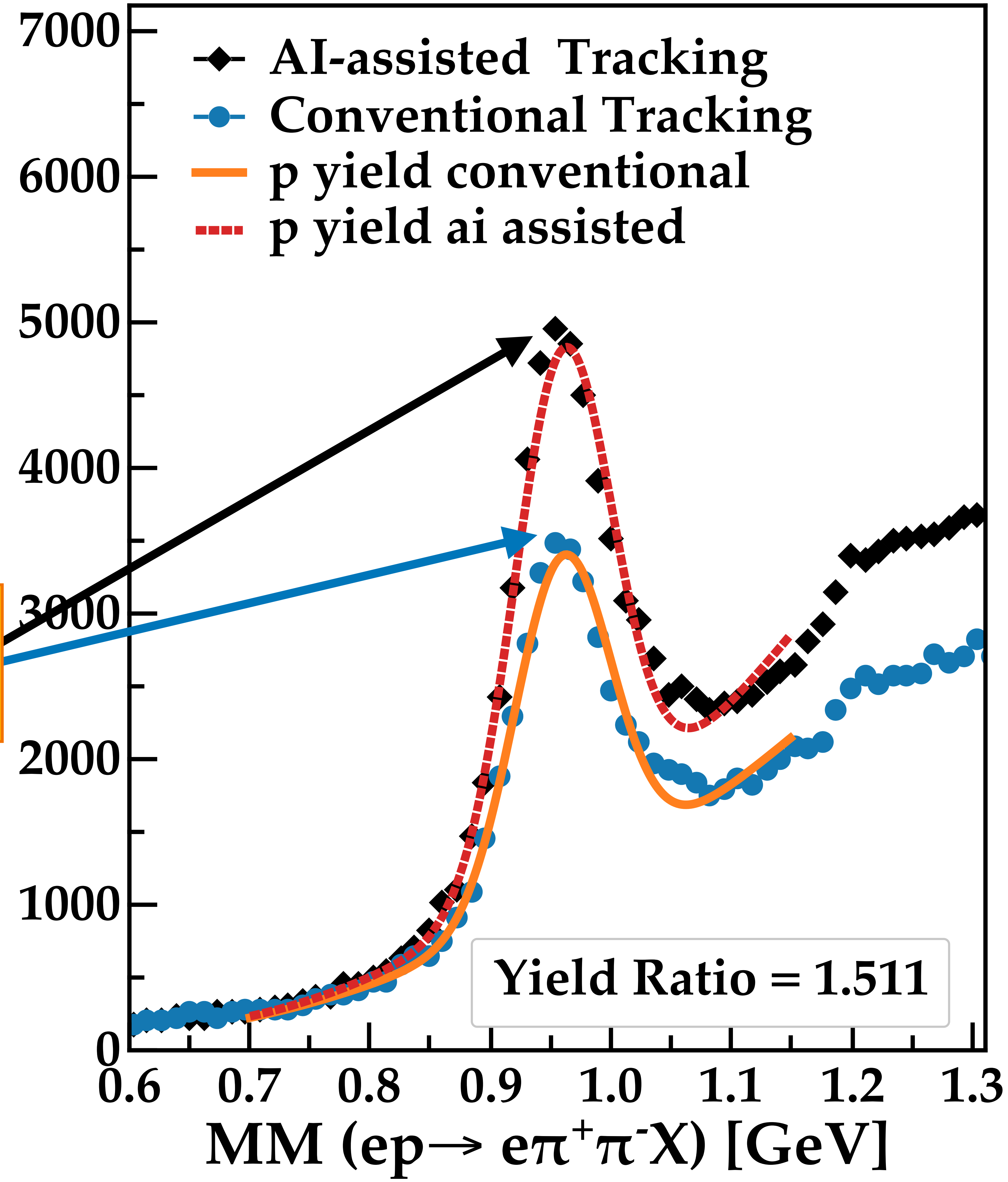


- At standard running luminosity, the de-noising slightly increases the yield compared to AI-assisted tracking.
- With increased luminosity, the de-noising helps to increase the yield significantly compared to conventional and AI-assisted tracking.
- Simulation underestimates the gain in yield significantly. In data the gain is much larger.

Current Workflow

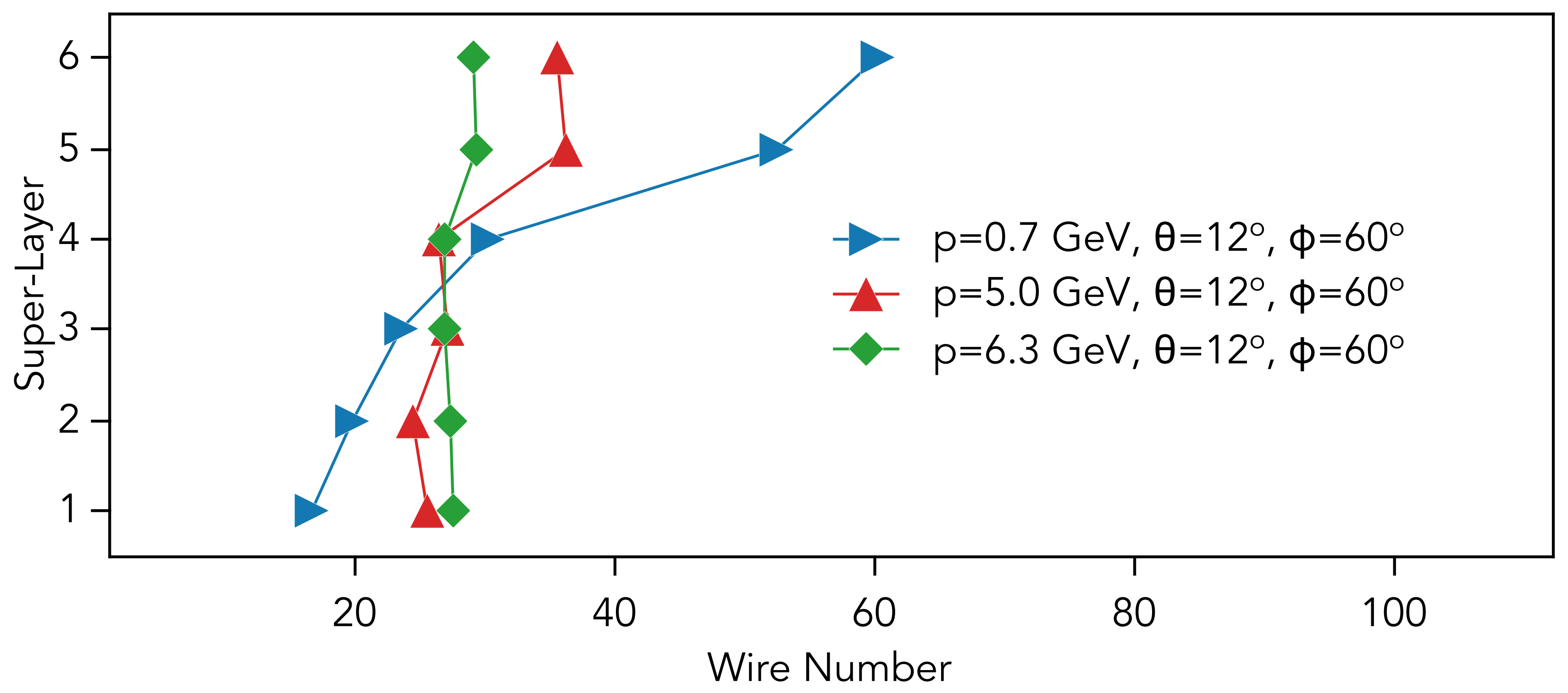
- CLAS12 Reconstruction software is based on SOA (CLARA) approach, where each detector reconstruction runs as a separate service
- The data reconstruction workflow now included de-nosier running prior to standard clustering and AI-Assisted tracking running prior to DC track finding.
- Drift Chambers code runs tracks suggested by AI-assisted tracking through Kaman-filter for final track parameter calculations.

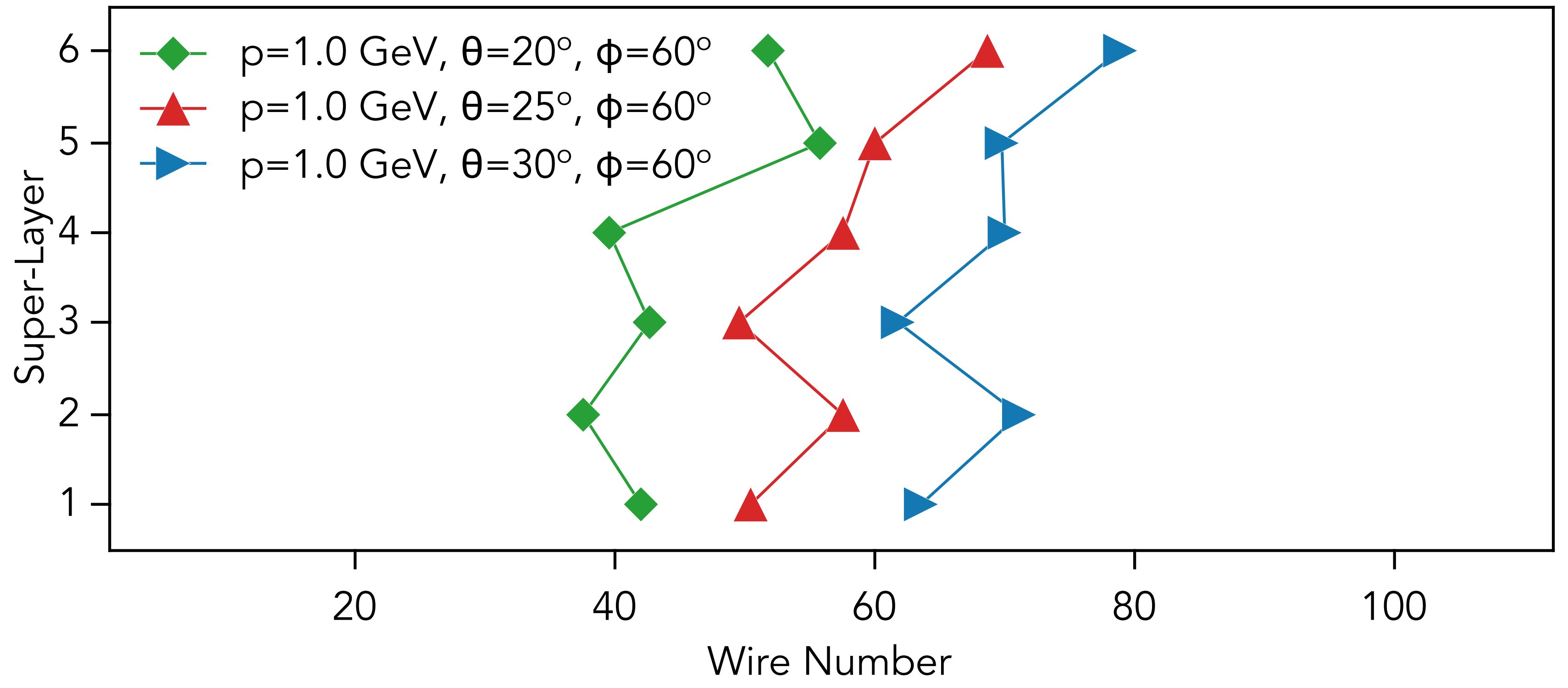


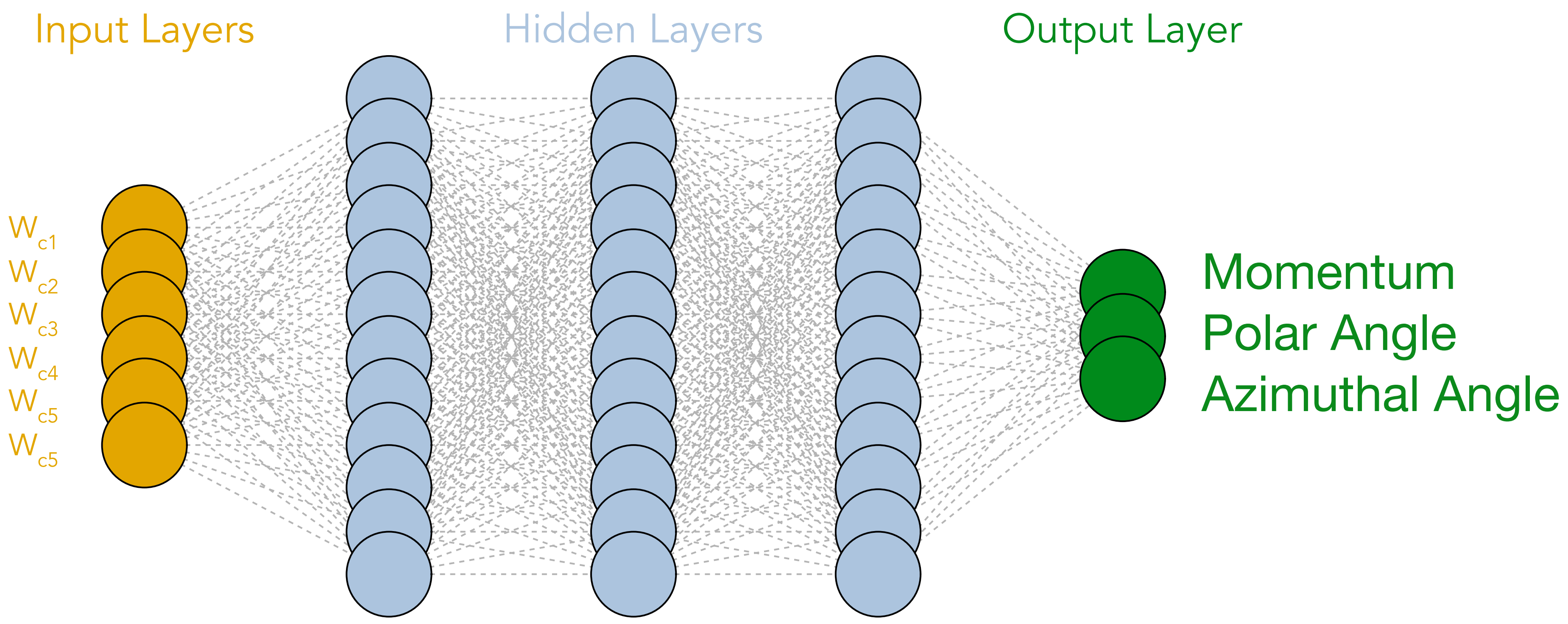


- Running at standard conditions (45 nA beam current) the AI increased the yield of missing protons by 51%.
- The improvement in yield is reaction and kinematics dependent, and for some event topologies reach 83% (J/psi with 3 particles detected final state).

Track Parameter Reconstruction

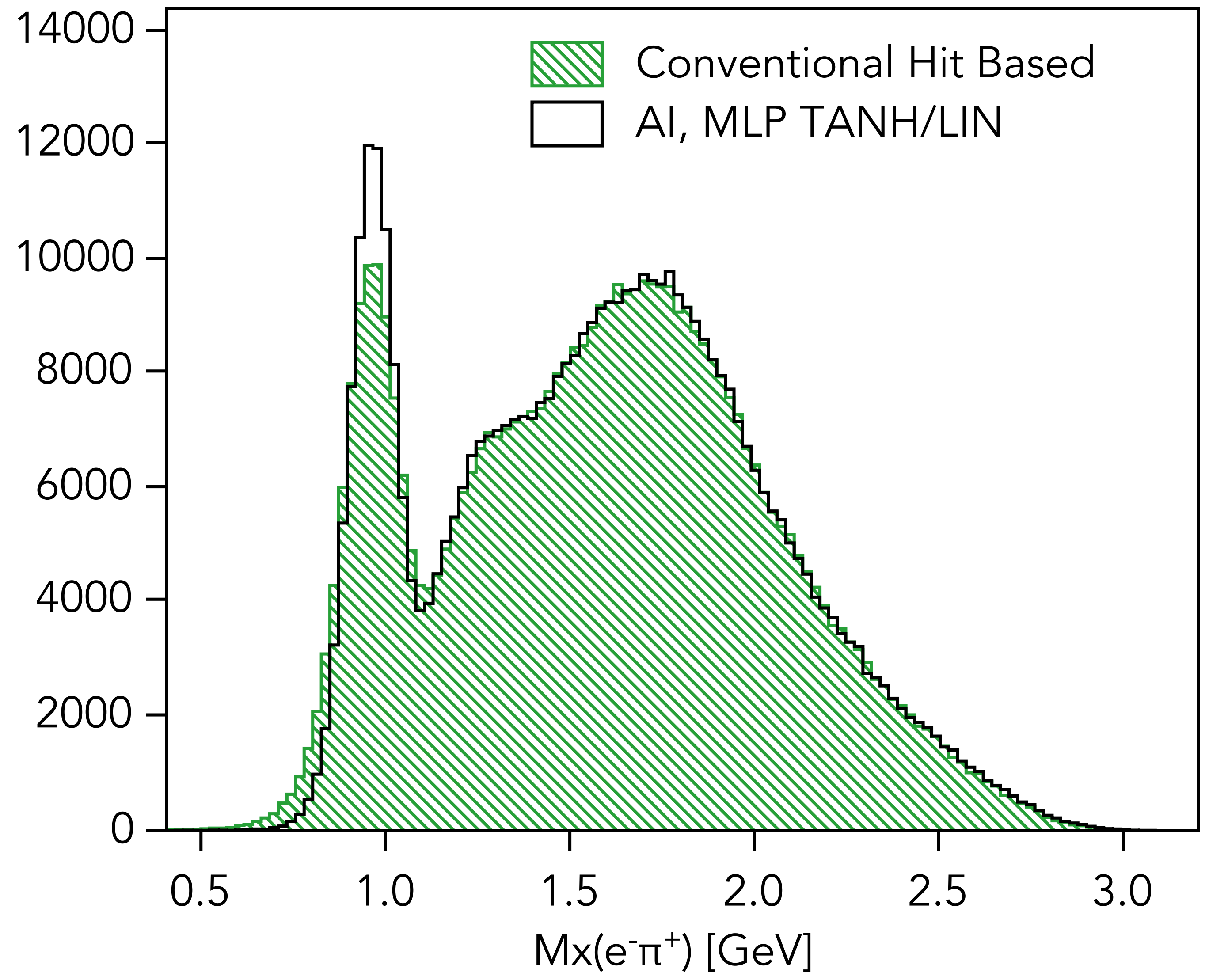






- Charge Track Parameter Inference
  - Reconstruct momentum and angles of particles based on the cluster positions of the tracks
  - Particles have distinct trajectories through drift chambers depending on their momentum, polar and azimuthal angle.
  - Design an MLP network and investigate different combinations of activation functions to derive the best network for this problem.

$$ep \rightarrow e^- n\pi^+$$

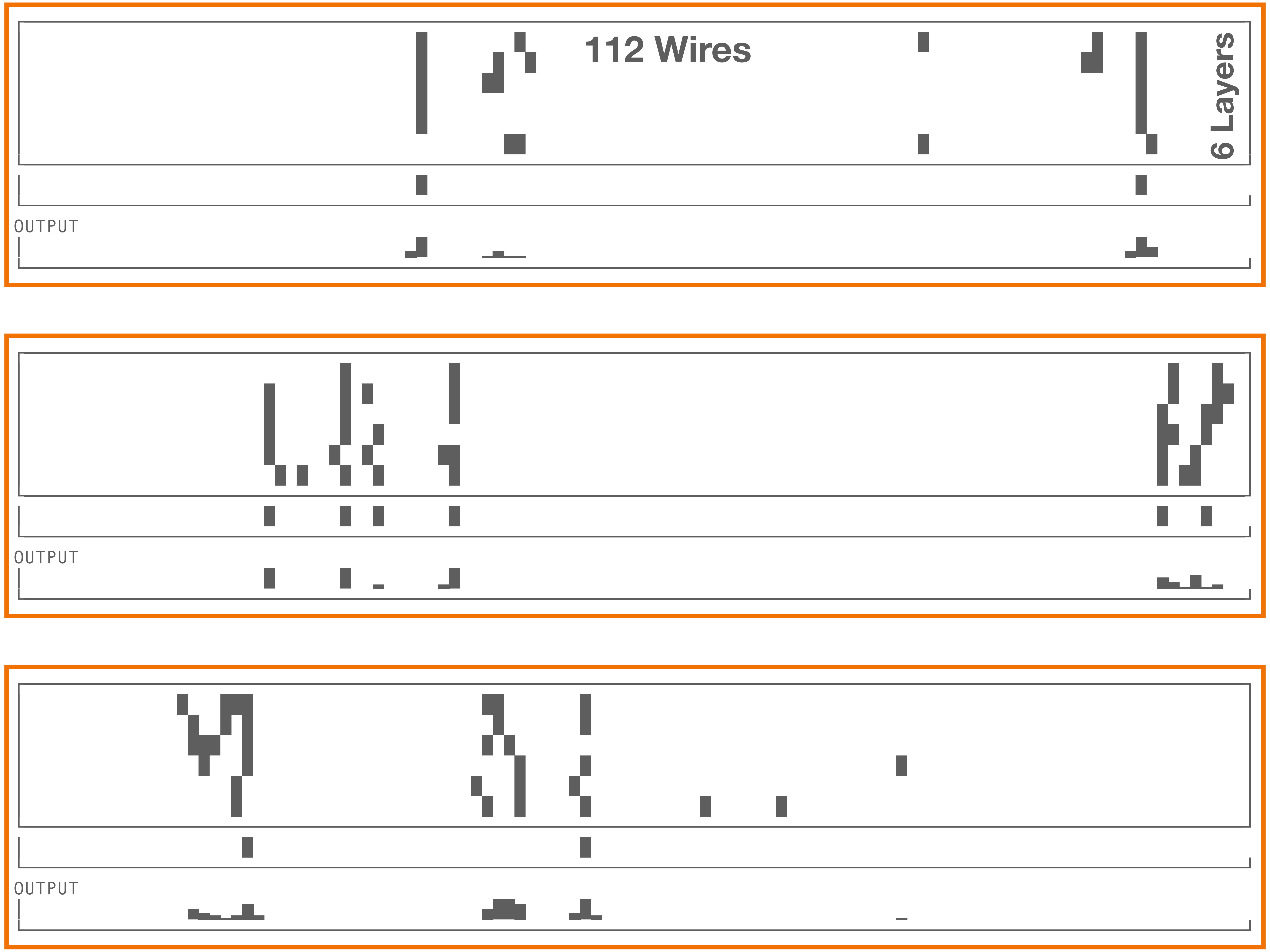


- Missing mass of two particles calculated using particle momenta from Hit-Based Tracking compared to missing mass calculated from AI particle parameter inference.
- Hit Based Tracking works ~250 ms per event
- AI reconstructs particle parameters <0.5 ms per event

Track Parameter Reconstruction

- The intention was to use reconstructed track parameters in reconstruction to speed up the code (fewer passes through Kalman-Filter), but it did not result in significant speed improvements.
- But we were one algorithm away from full track reconstruction (clustering)
- Decided to implement AI clustering algorithm (Convolutional Logistic Regression)
- With Clustering, Track Identification, and Track Parameter reconstruction we could run tracking without the Reconstruction software.
- The question is:
  - How Fast, and How Accurate?

AI-driven clustering



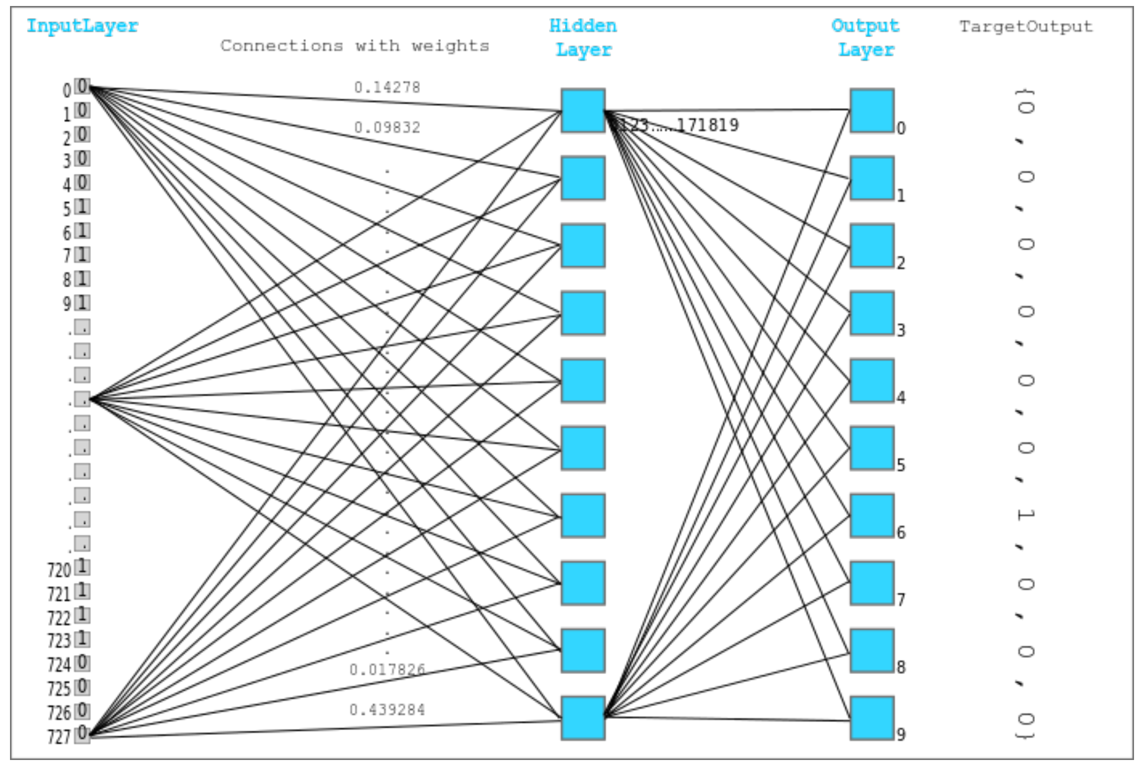
- **AI to extract clusters**
  - Convolutional Logistic Regression
  - Input:
    - **Matrix 112x6 of one Chamber (out of 6)**
  - Output:
    - **112 numbers: likelihood of a cluster in the wire position**

- **Trained on Simulation:**
  - Matrix of one Chamber from simulated tracks
  - Noise added from CLAS12 background files (background extracted from random trigger events)

Input

Truth

Output



Tools used

DEEP NETTS

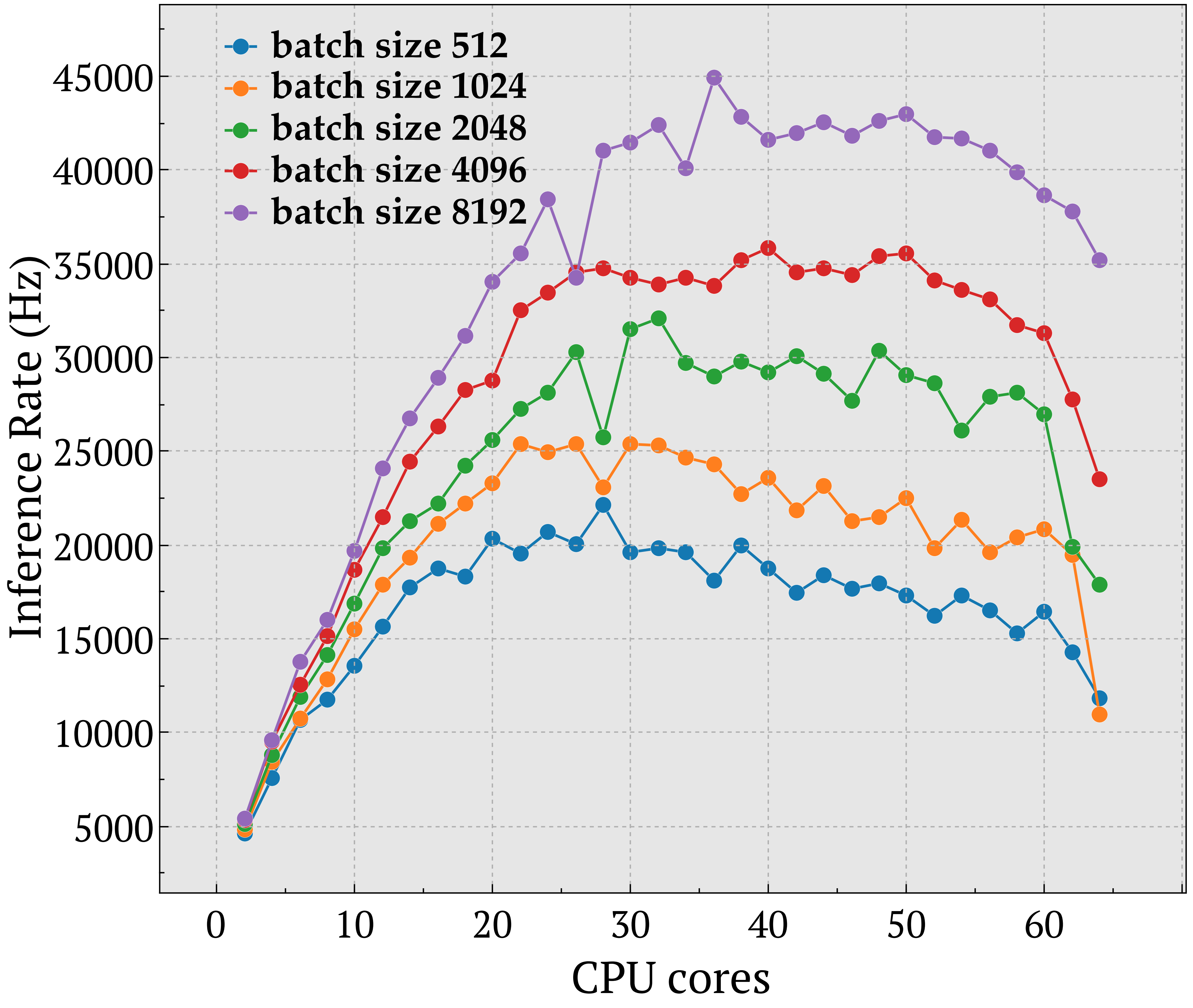
https://www.deepnetts.com

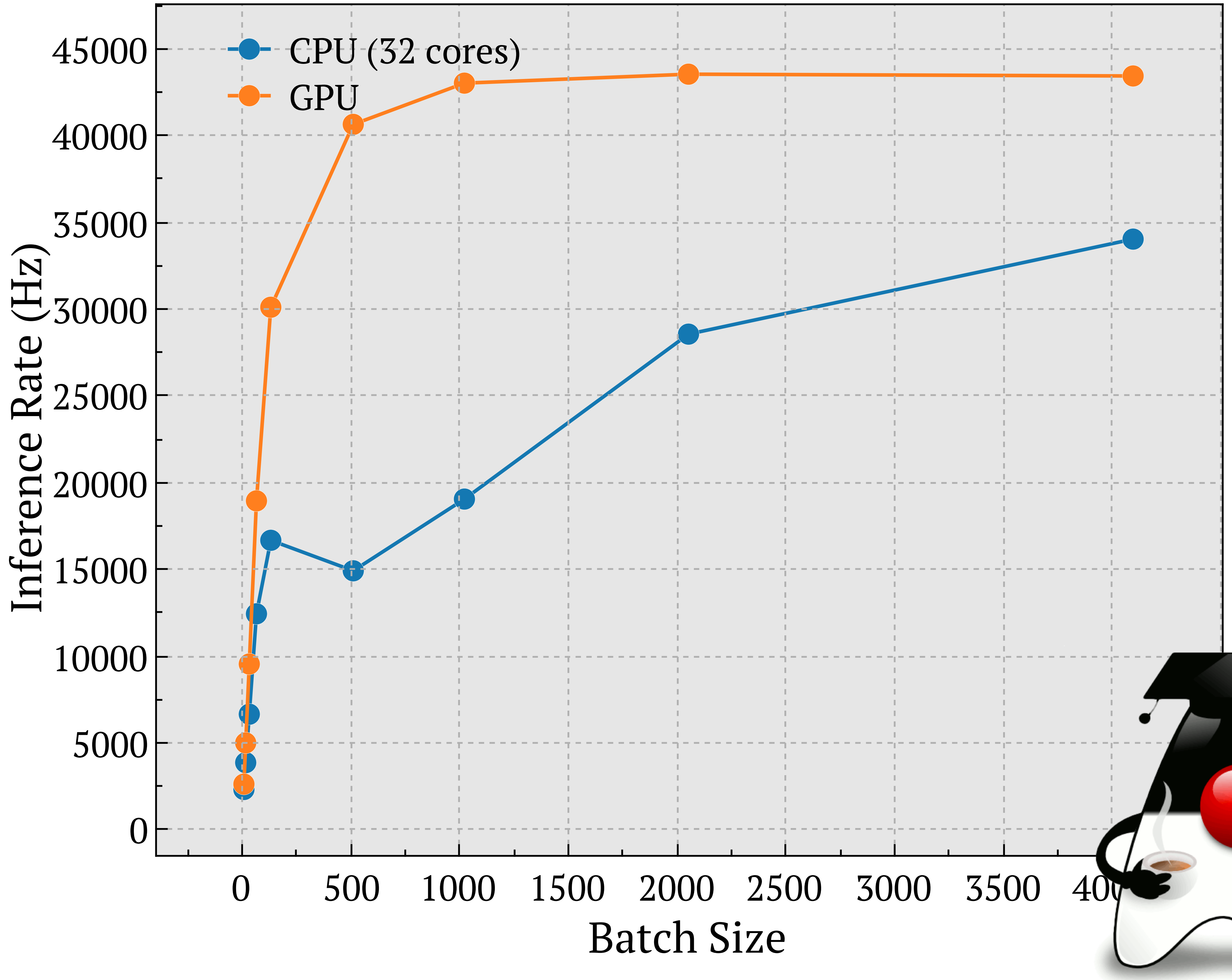
- Python is usually associated with AI development, frameworks like TensorFlow, PyTorch, SciLearn
- CLAS12 reconstruction software is a multi-threaded SOA architecture implemented in Java
- It is preferable to have Java for AI implementation for easy integration in the workflow.
- All the developments in this talk are done in Java using DeepNeets Framework, which has all the necessary networks (MLP, CNN, and more).

https://deeplearning4j.konduit.ai/

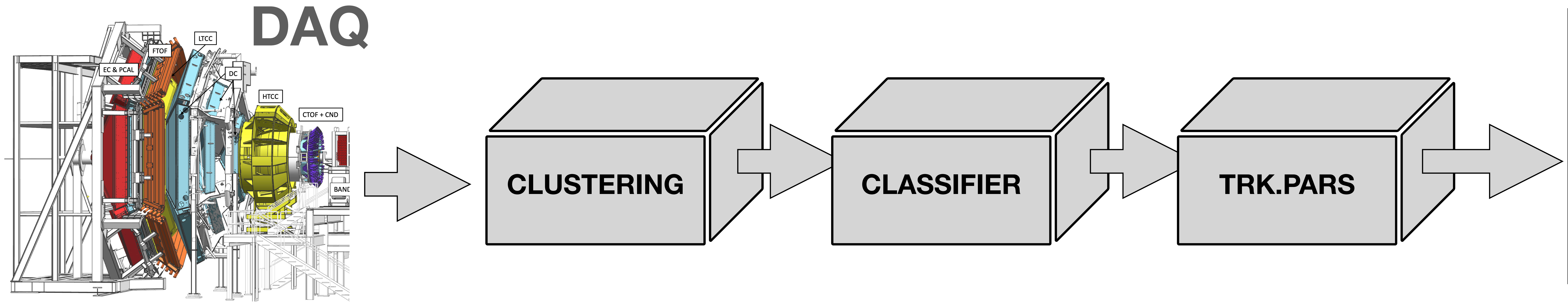
DL4J

Level-3 trigger with Convolutional Neural Graph (including Drift Chambers and Calorimeter Images)



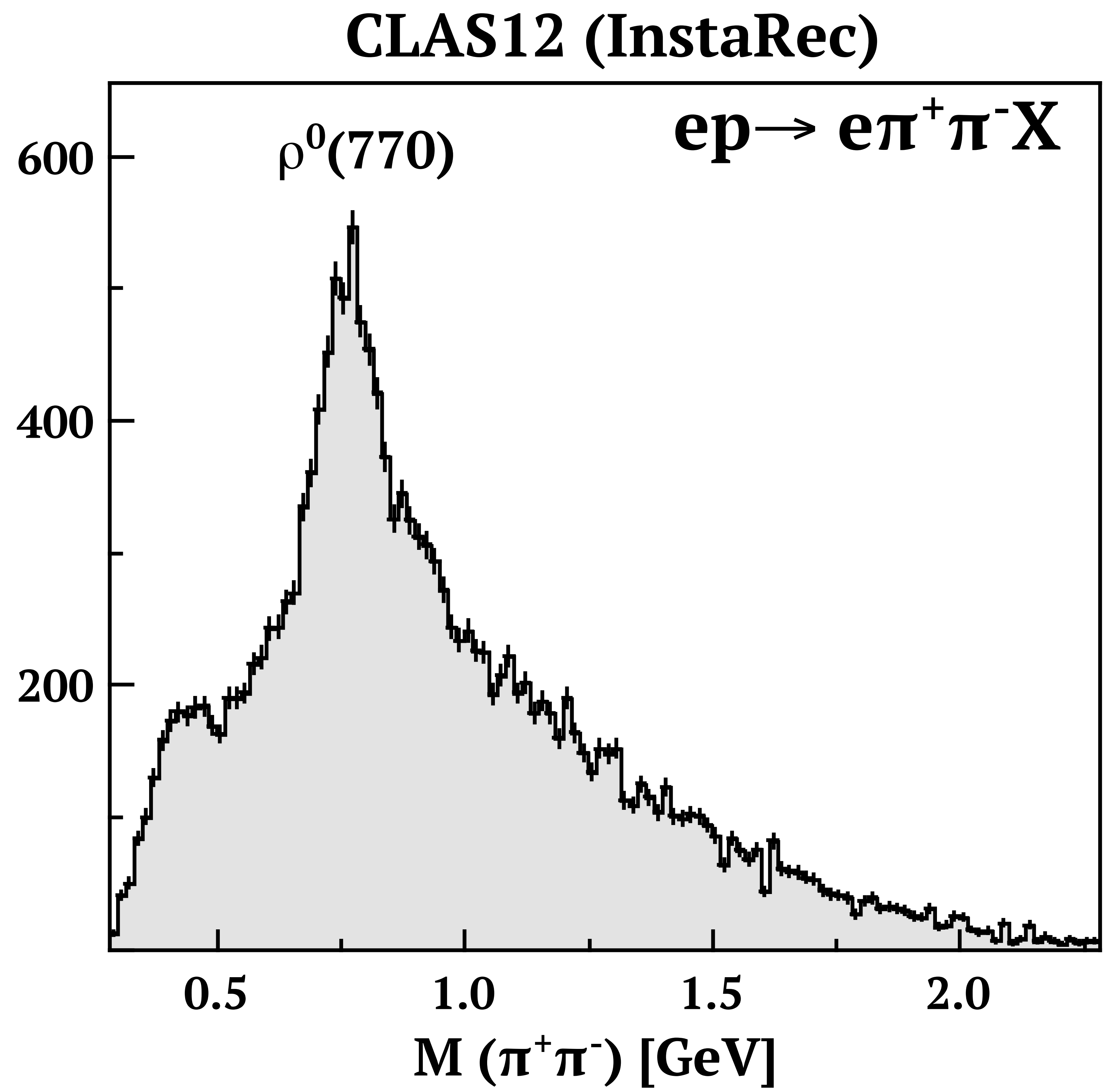


Putting all together (InstaRec)



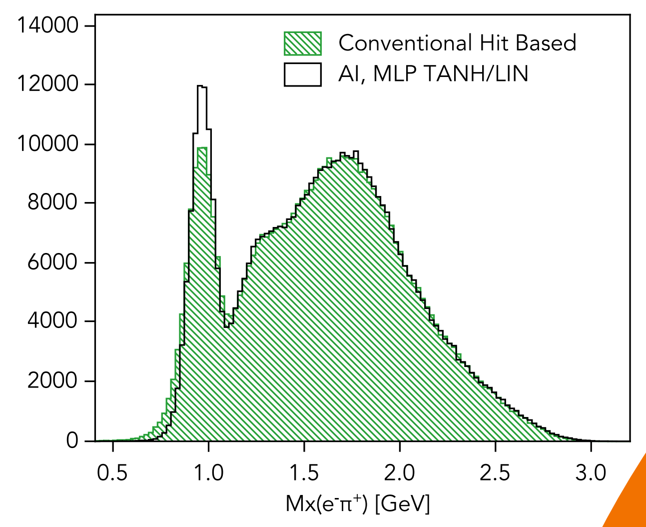
1	2	3
Px	Px	Px
Py	Py	Py
Pz	Pz	Pz
Vz	Vz	Vz

- **Online-Reconstruction:**
  - Data from DAQ is passed through 3 Neural Networks
  - The output contains particles with momenta and direction
- **Physics:**
  - Tracks reconstructed online (at 44 kHz)
  - The invariant mass of two pions, with pre-selection of electron and cuts on W and Q2.
- **Online:**
  - Online reconstruction code (about 200 Hz) takes suggestions from AI on which events to reconstruct.



Future of CLAS12-AI (ALL components exist, moving towards online)

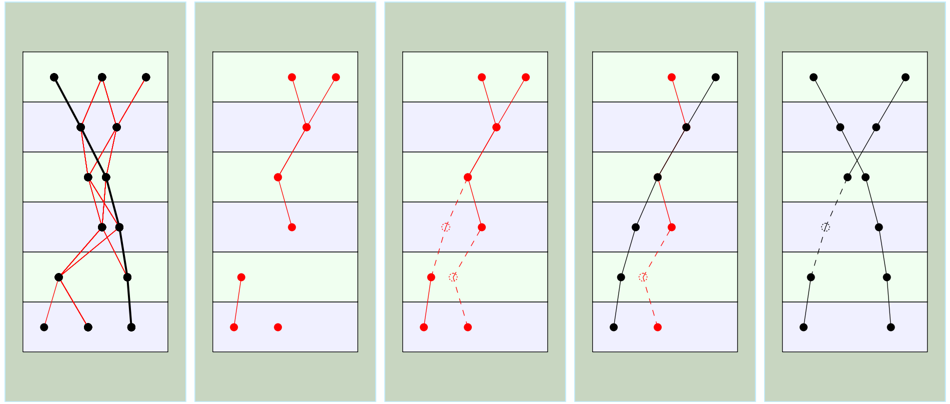
Physics Reconstruction (AI)



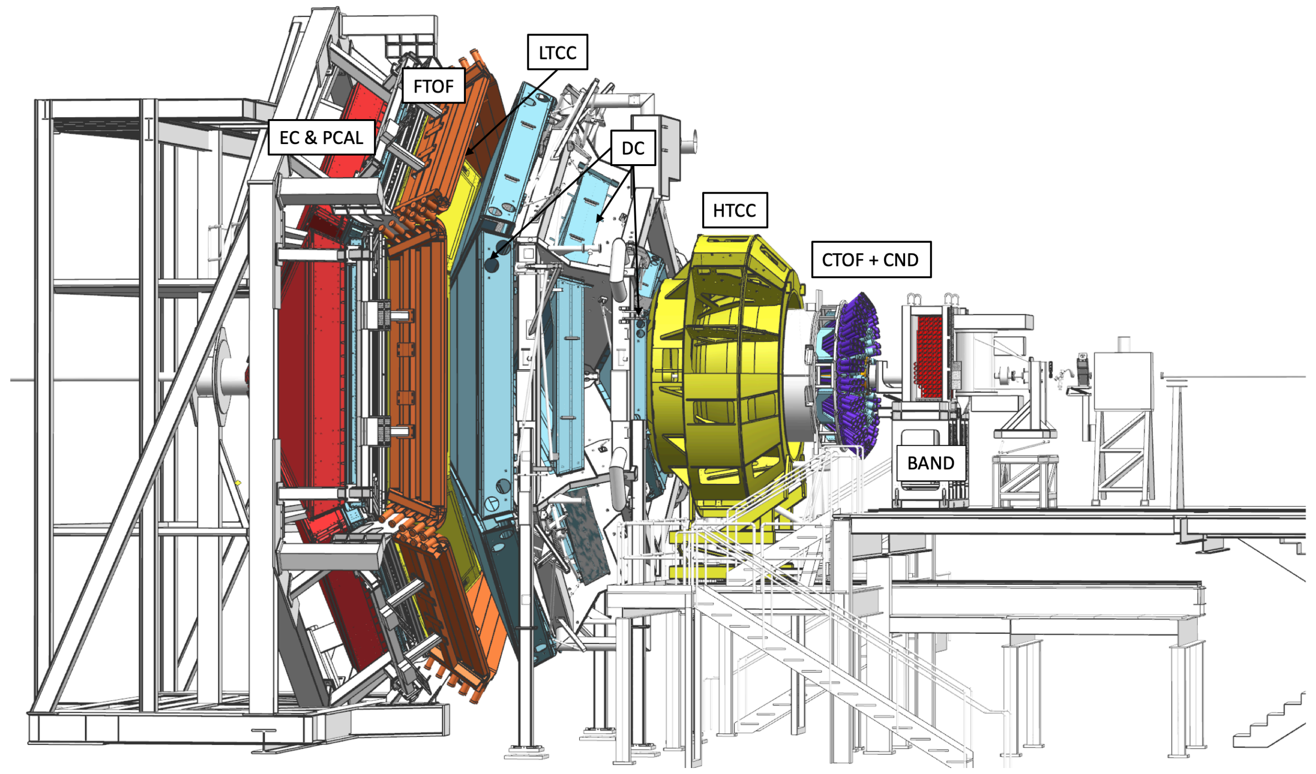
Data Persistence

Saving experimental data
Already containing tracks
And physics topologies
Identified by AI

Track Classification (AI)



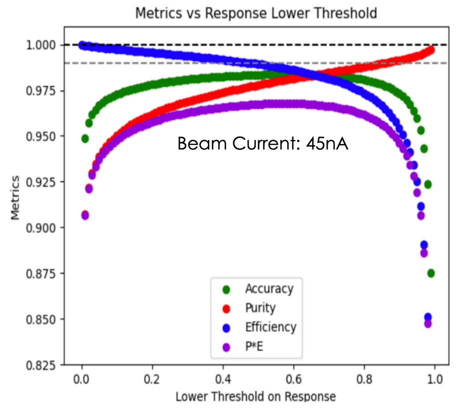
Classifying track candidates from
Reconstructed clusters
In real-time

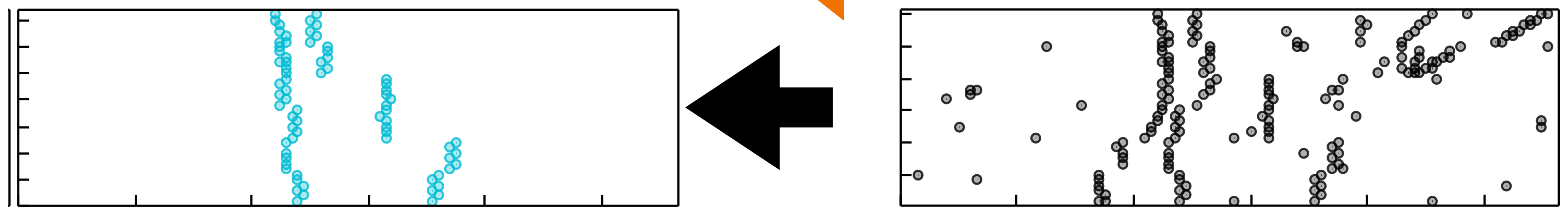


Data Acquisition

Level-3 Trigger (AI)

Threshold	Purity	Efficiency	Accuracy
0.0012	0.841	0.9999	0.906
0.03	0.930	0.999	0.962
0.47	0.977	0.99	0.983





Removing Noise signals
From tracking detectors

Data De-Noising (AI)

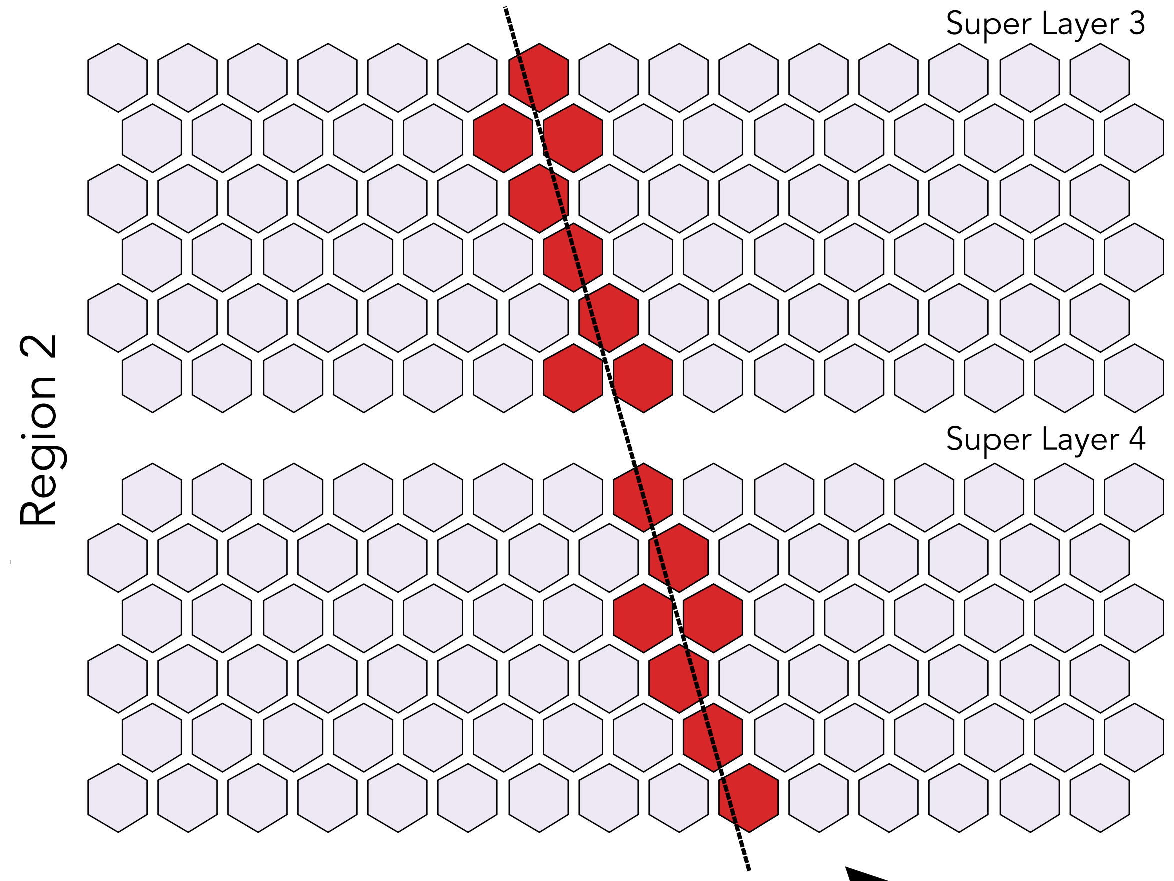
Summary

- The CLAS12 tracking AI (including De-Noising and track identification) provides a 50%-75% increase in statistics for the completed experiments (**in production since early 2023**)
- New developments in track reconstruction allow real-time physics reaction identifications, allowing the organization of the data stream into different physics samples.
- Significantly increases the efficiency of online data reconstruction and data monitoring by pre-selecting which events to process.
- **Future developments**: identify particle types with AI in real-time, including time of flight and calorimeter response processing.

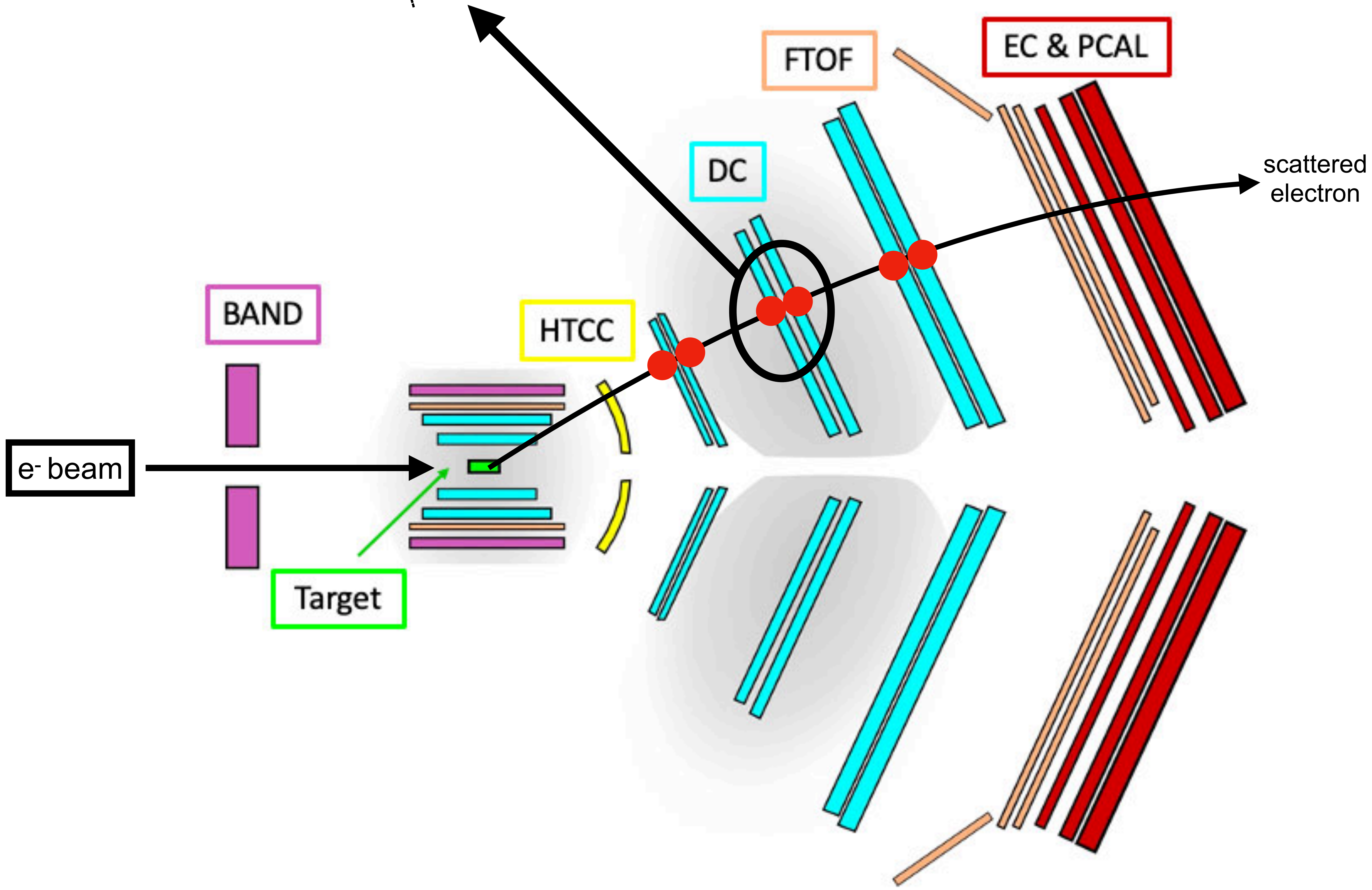
DEEP NETTS
https://www.deepnetts.com

Backup Slides

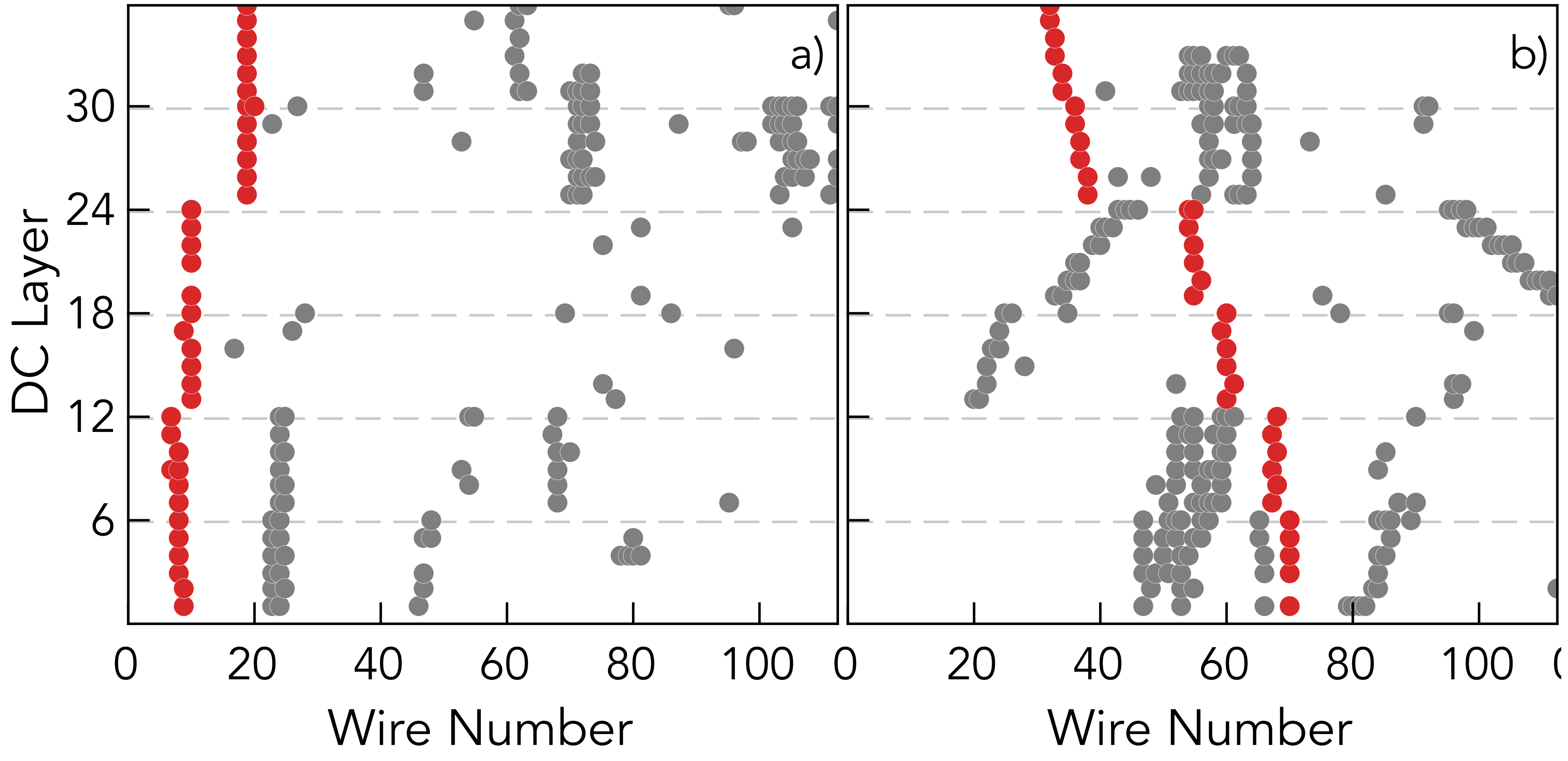
CLAS12 Detector



- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory



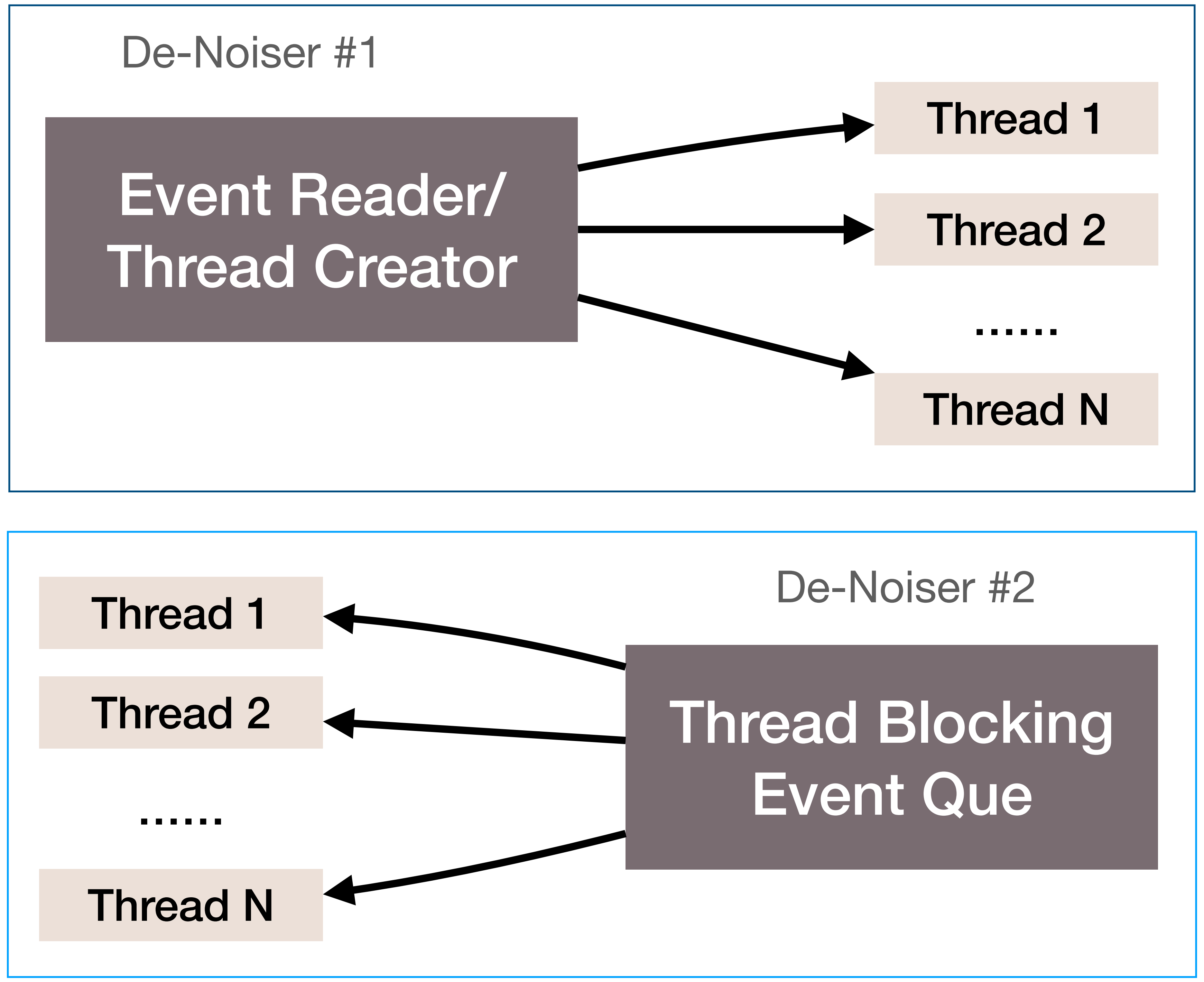
- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can speed up the tracking process and improve efficiency

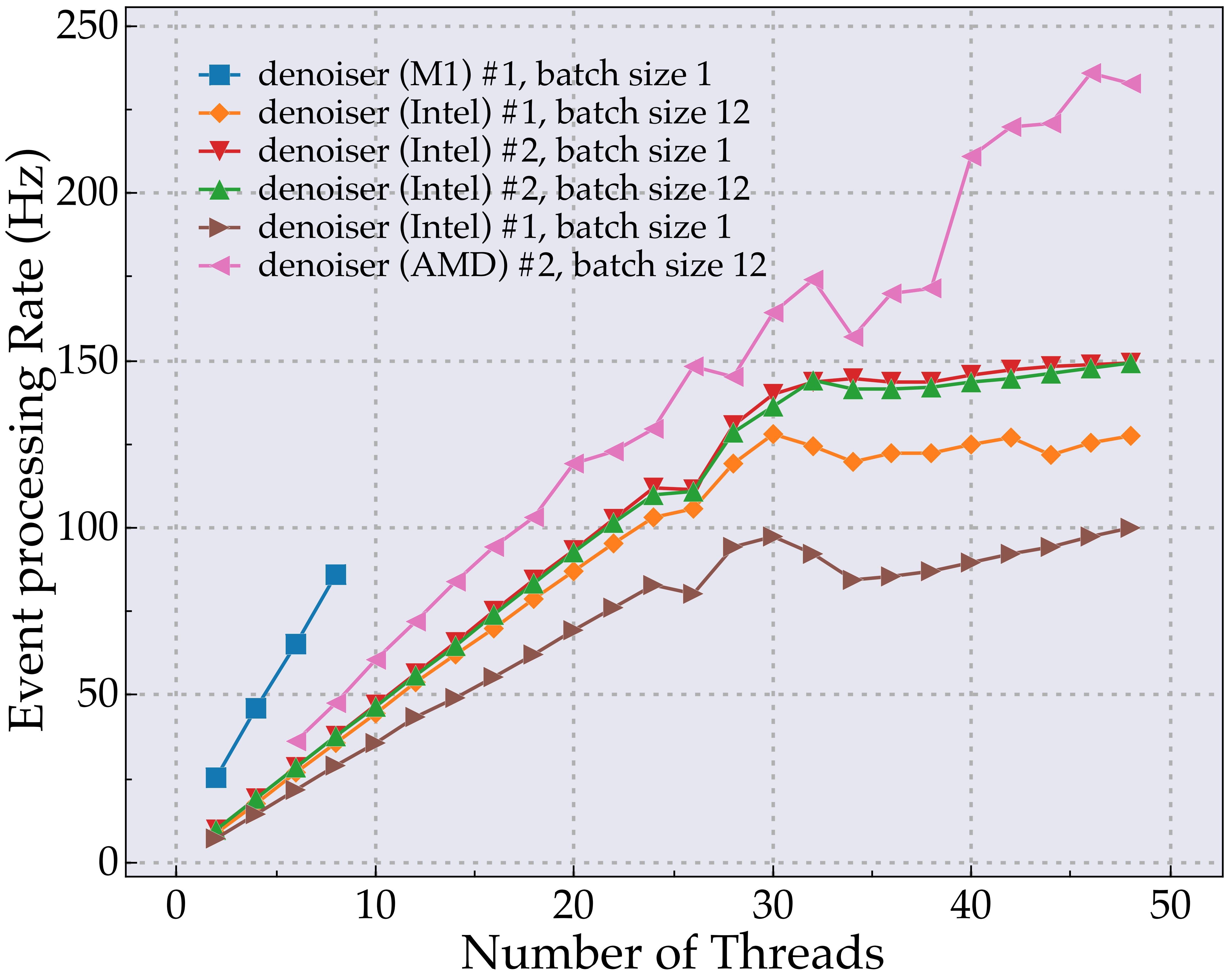


De-nosing Performance Multi-Threaded

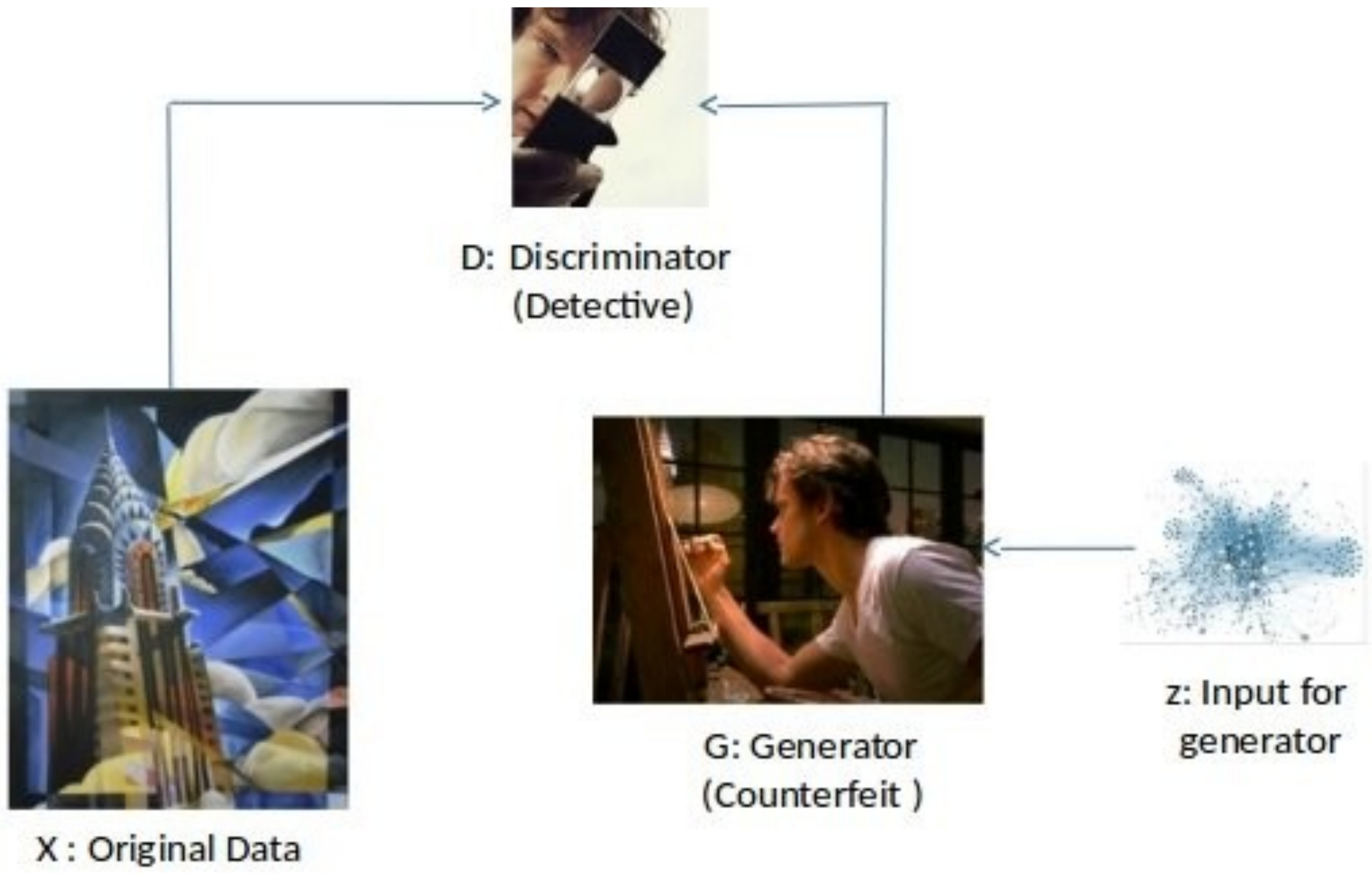
- **C++:** Keras model inference in C++ code implemented for CLAS12 de-noiser.

- **Multi-Threading**: Multi-threading implemented to process data files (using std::thread)



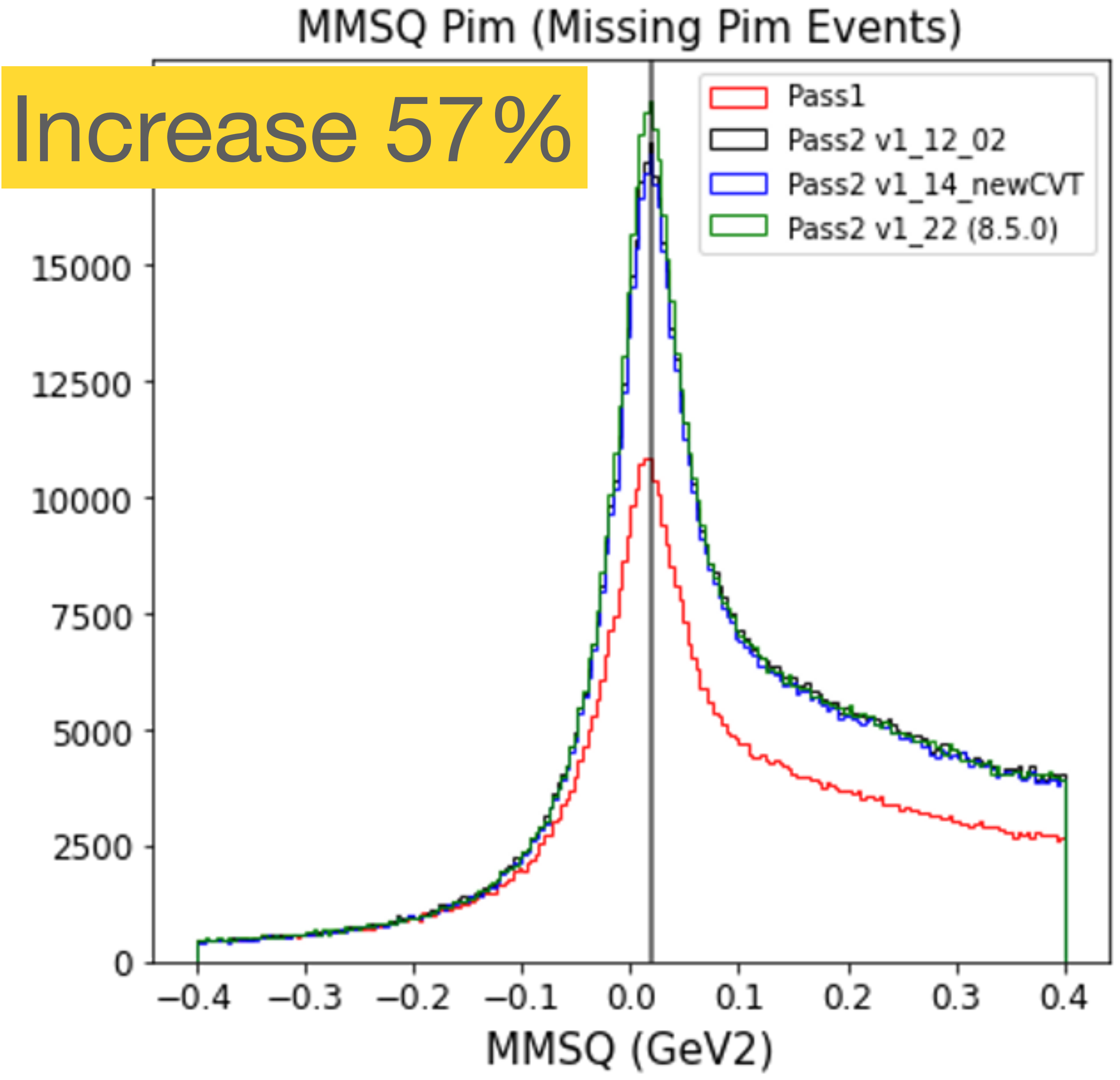


- Image Generation:
  - AI tools to generate images based on the description
  - Ability to generate images with the style of a certain painter

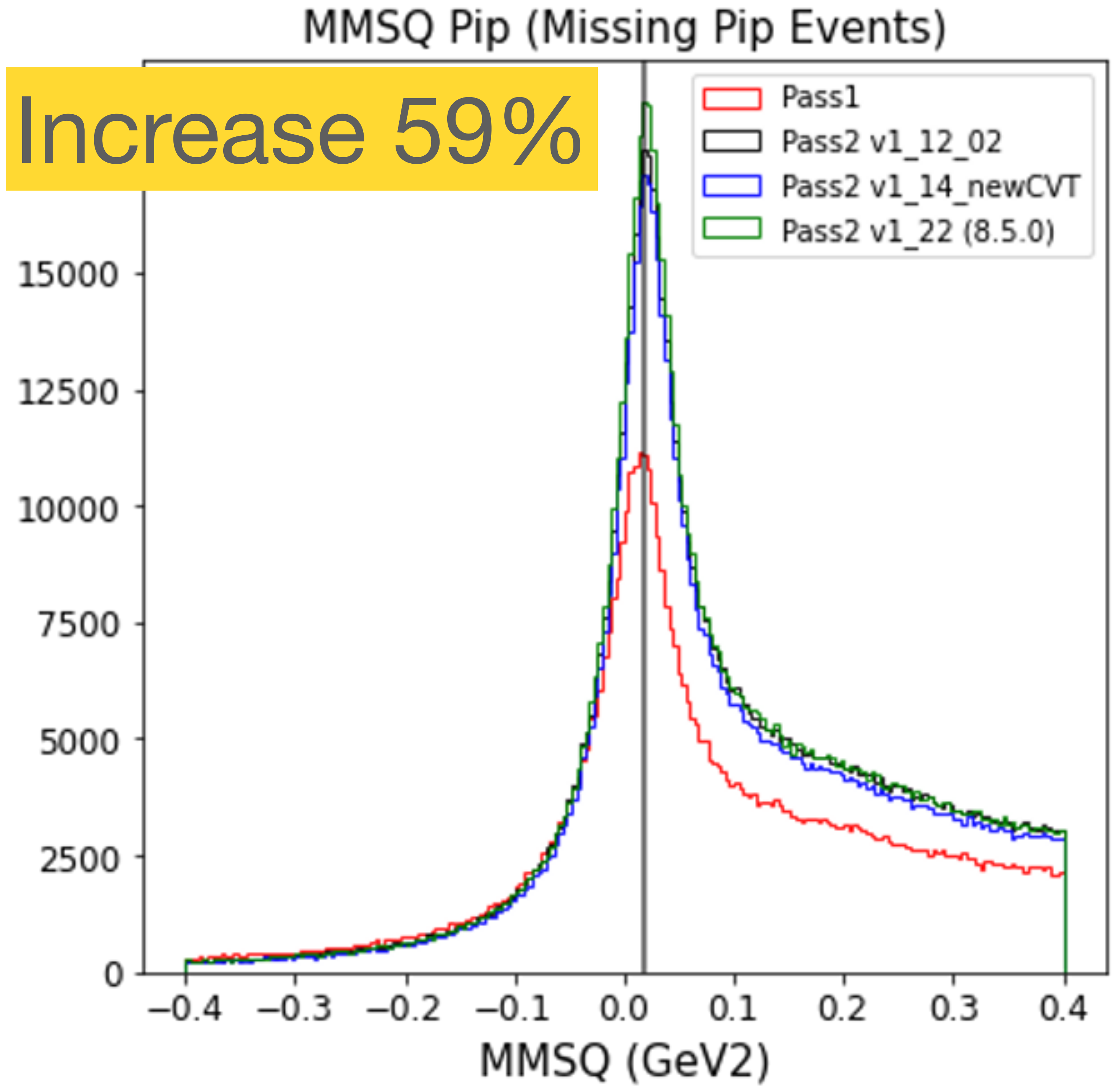


RUN GROUP-A Pass2 Validation Cooking
Includes De-Nosing and AI-assisted Tracking

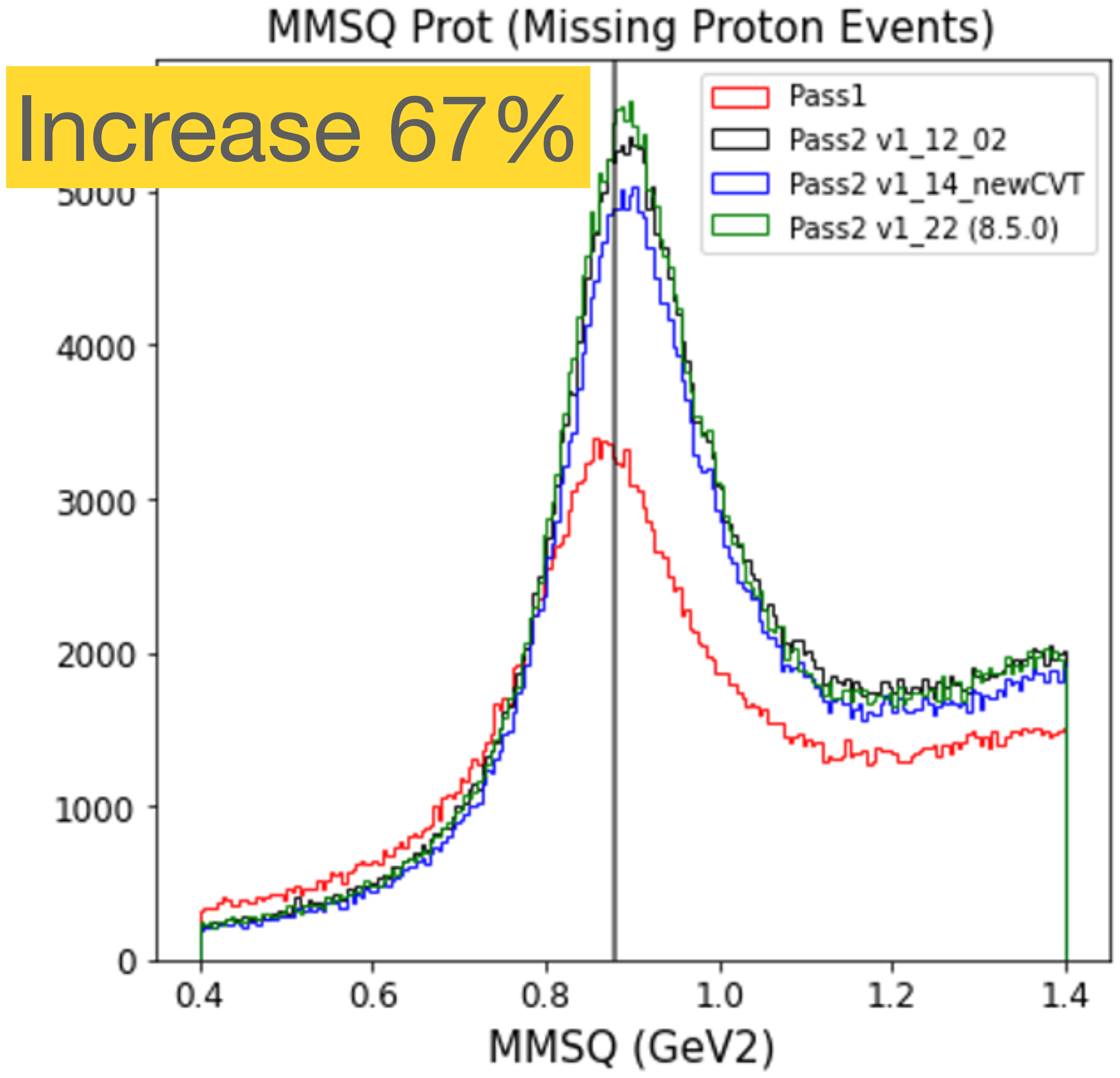
$$ep \rightarrow e'p\pi^-(X)$$



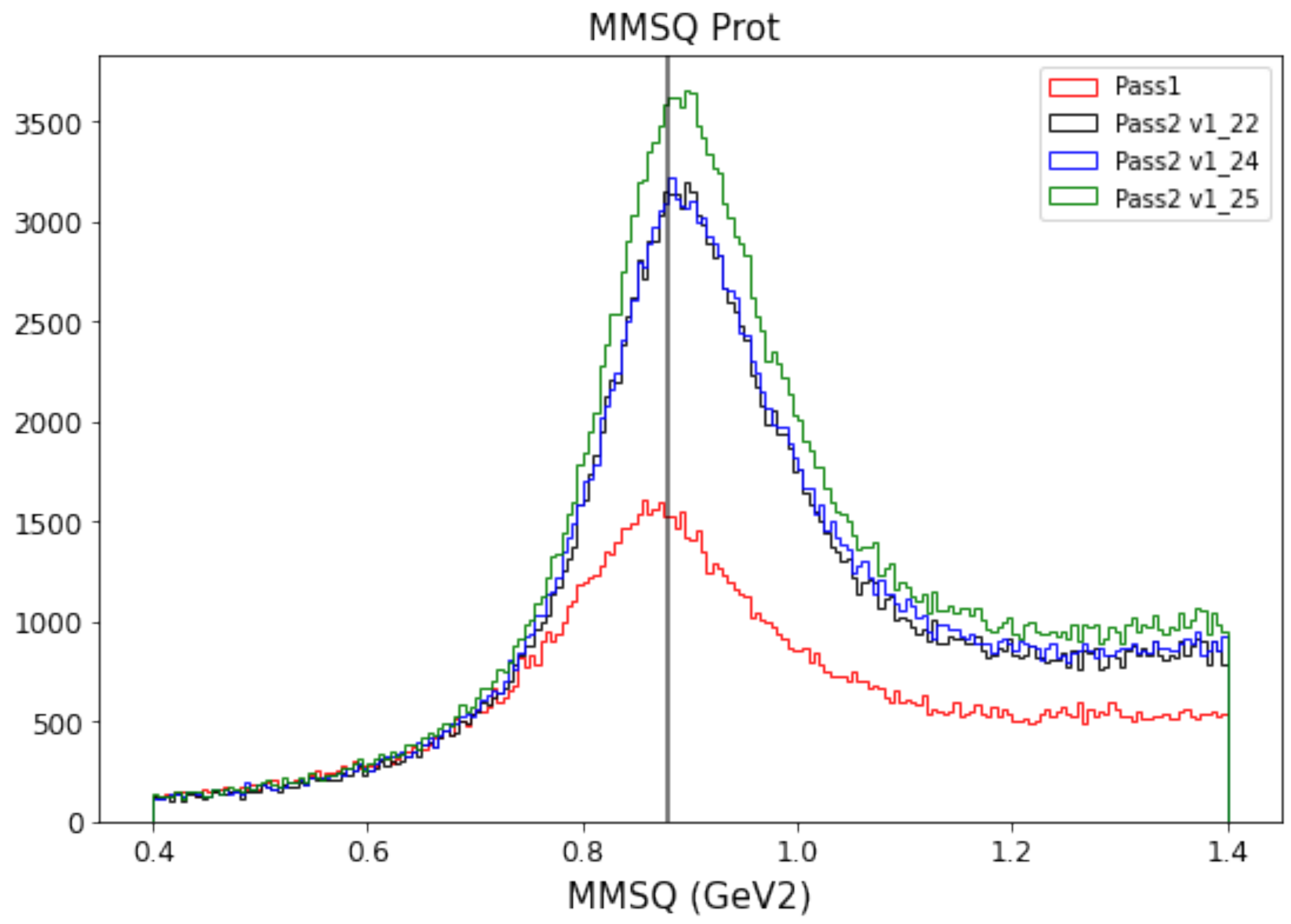
$$ep \rightarrow e'p\pi^+(X)$$

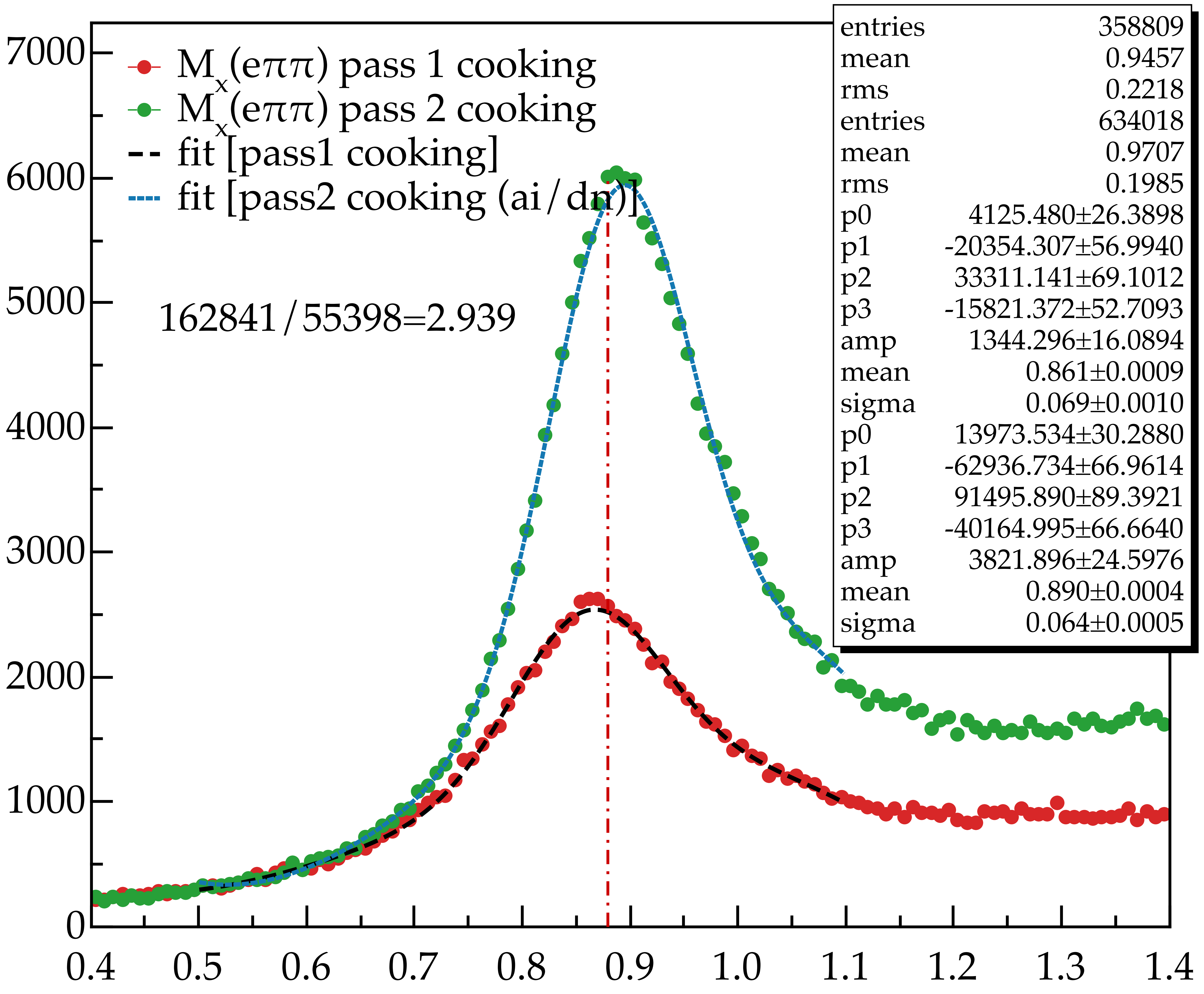


$$ep \rightarrow e'\pi^+\pi^-(X)$$



De-nosing Performance With Central Detector





pass1 = 129894
pass2 v1_22/pass1 = 1.618
pass2 v1_24/pass1 = 1.662
pass2 v1_25/pass1 = 1.866

De-nosing Performance Multi-Threaded