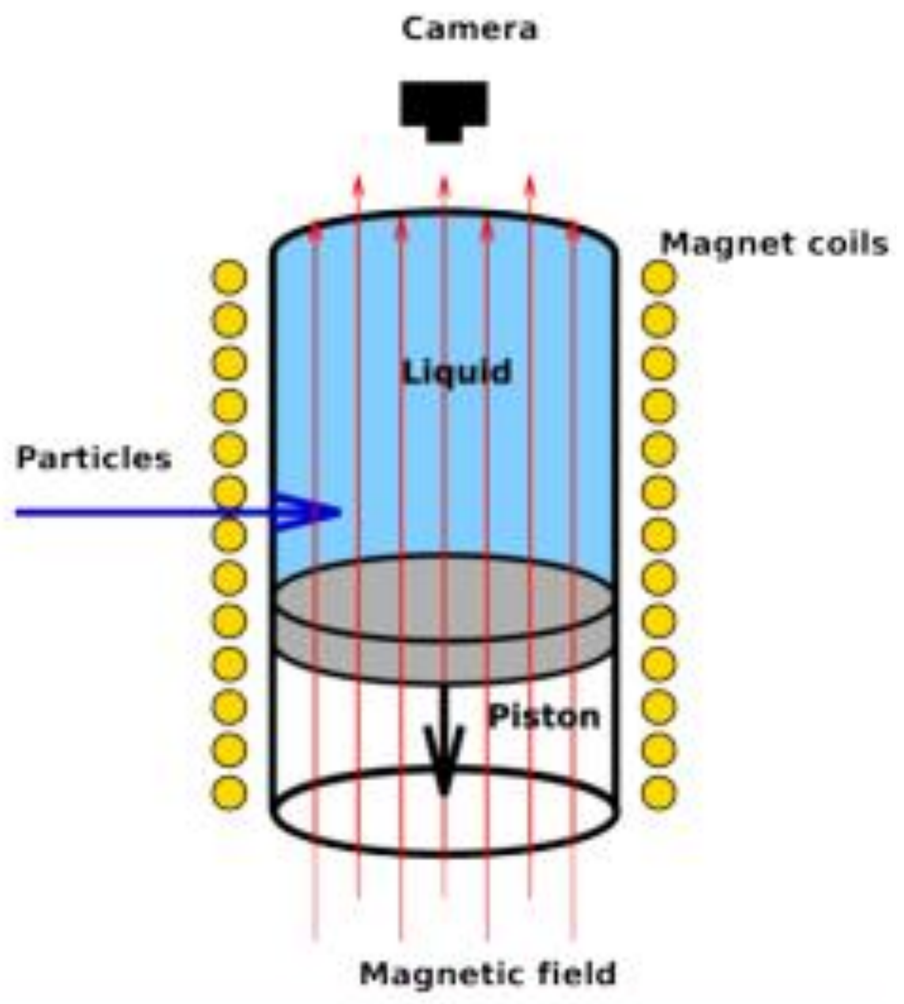
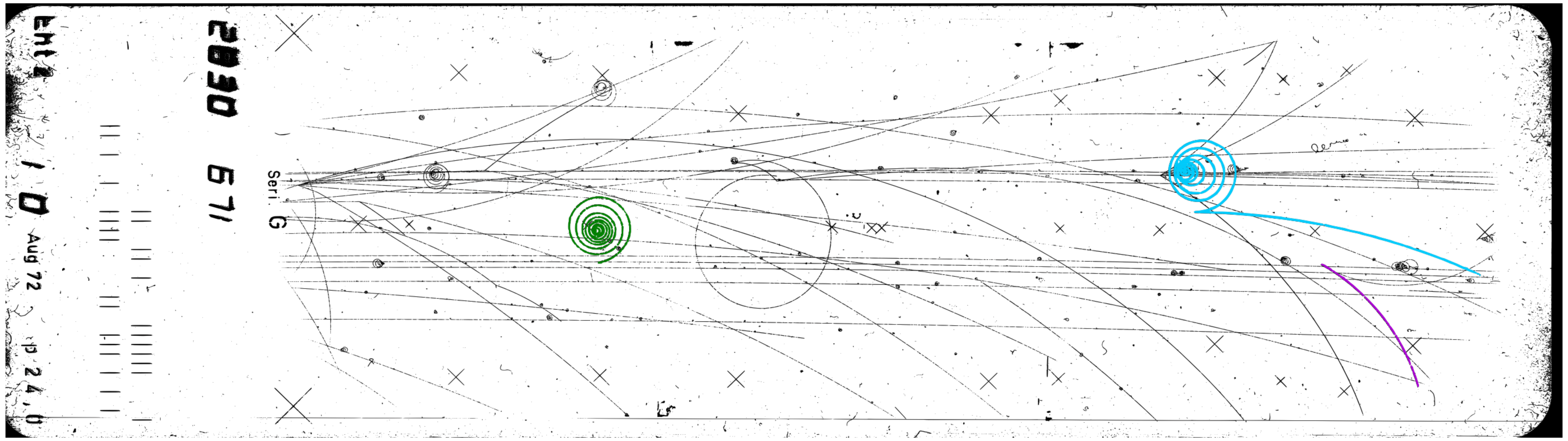
AI/ML in CLAS12 Track Reconstruction

Future of experiments with AI/ML

Gagik Gavalian (Thomas Jefferson National Lab), Richard Tyson (Thomas Jefferson National
Hall A/C Collaboration, JLAB

Bubble Chambers (early experiments)





Manual Scanning and Digitization:

In the early days of bubble chamber experiments (from the 1950s through the 1970s), photographs of particle tracks were produced at a high rate. Each image had to be examined by trained "scanners" who used optical devices and digitizing tables to measure the coordinates of the tracks. For many experiments, a single event (one photograph) could take on the order of **10–30 minutes** to analyze manually. When you multiply that by thousands or tens of thousands of photographs, it's easy to see why the overall effort could extend over many months or even years.

How Much Data Can You Analyze?

Earth Population in 1984: 4.8 billion

Average rate of processing event **0.0008 Hz** (events per second)

Earth Population in the age bracket 21-65 years old 56%

The events reconstructed (if everyone joins in) 2,135 kHz

Evolution of Data Reconstruction

How Much Data Can You Analyze?
Earth Population in 1984 - 4.8 billion
Average rate of processing event **0.0008 Hz** (events per second)
Earth Population in the age bracket 21-65 years old 56%
The events reconstructed (if everyone joins in) 2,135 kHz

Modern Advanced chips
Multi-core servers are used (64 cores per node)
Experiments collected data at 16 kHz
Tracks were reconstructed with a rate of **2 Hz** (per core)

Improvement 4 orders of magnitude →

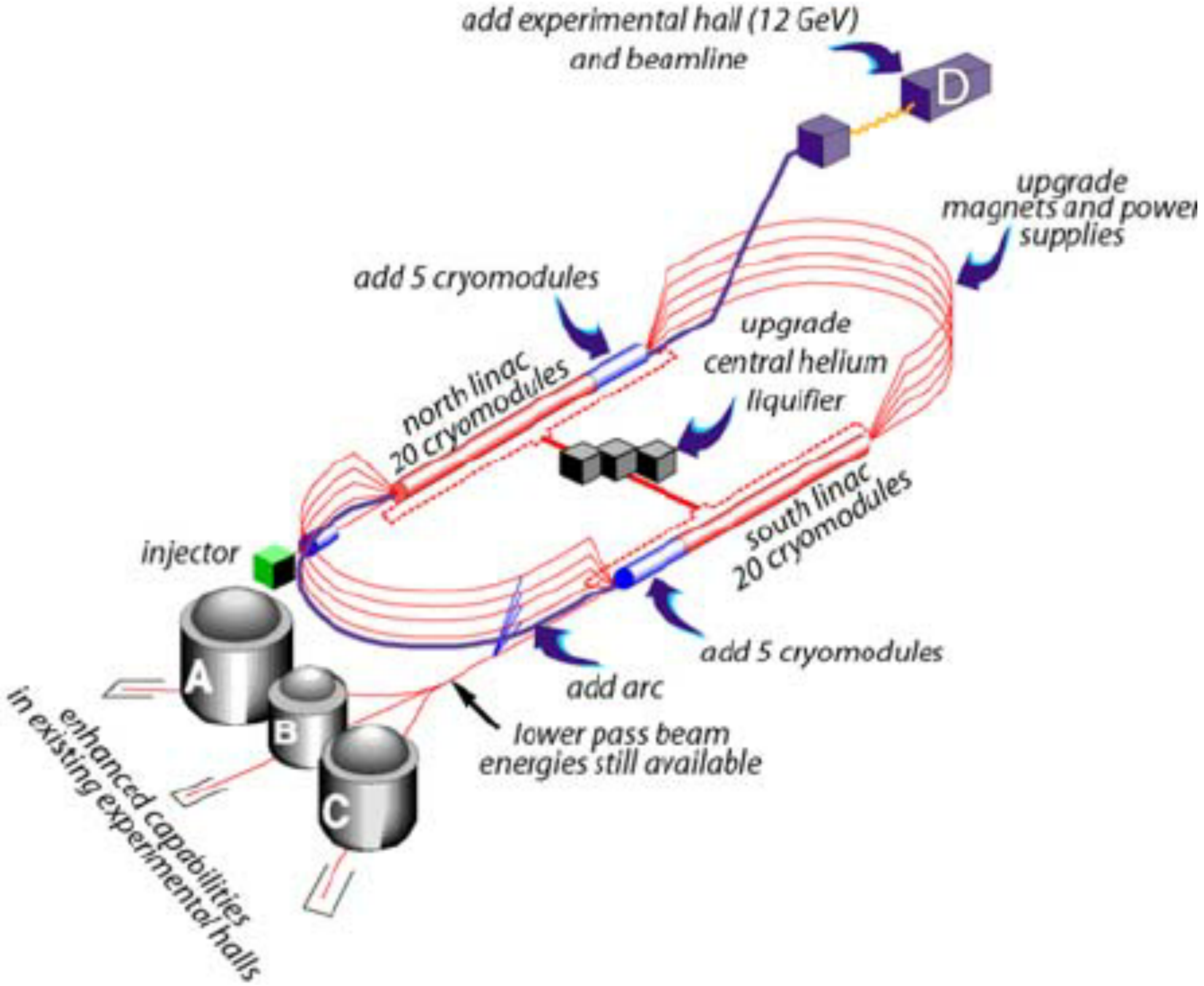
No significant speedup →

Age of AI → ?

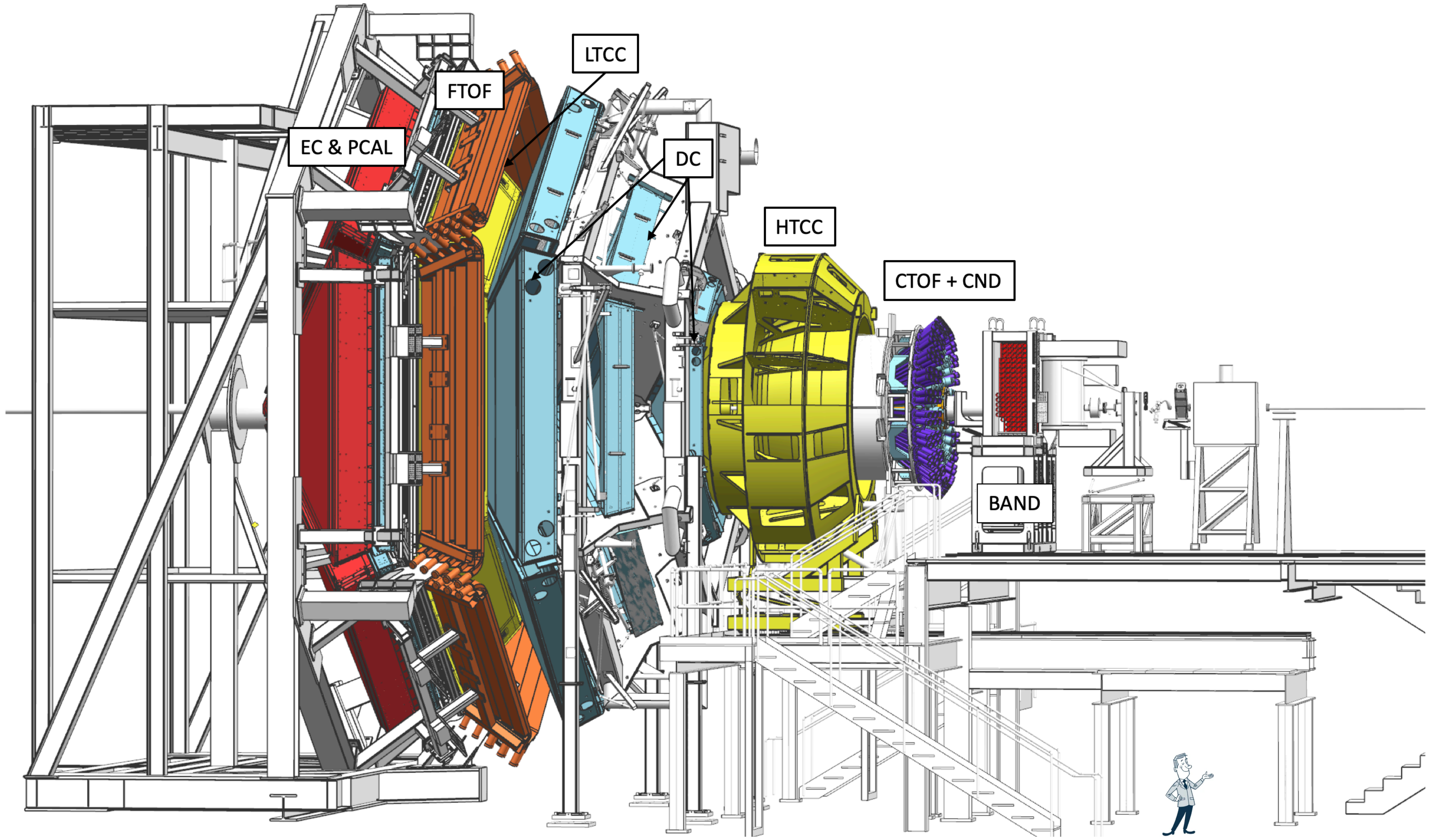
1984 — **2004** — **2024**

Introduction of Computers into Reconstruction
Computers are used to reconstruct tracks
Experiments collected data at 2 kHz
Tracks were reconstructed with a rate of **8 Hz** (per core)

Thomas Jefferson National Laboratory



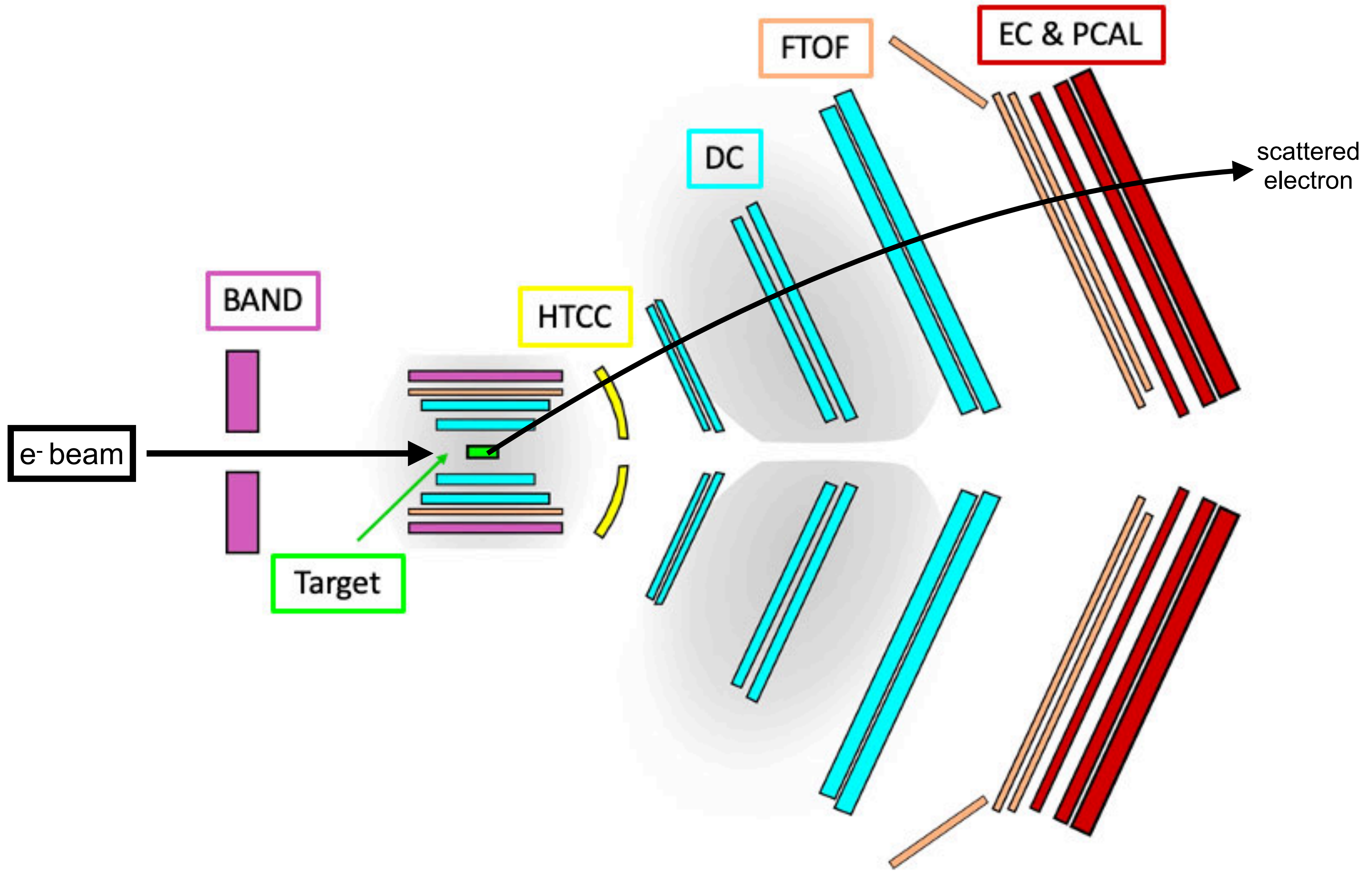
CEBAF
Continuous Electron Beam Accelerator Facility
Polarized Electron Beam (maximum energy 12 GeV)
Experiments Conducted in 4 Experimental Halls



CLAS12
Located in Hall-B
Data Acquisition Rate **~16 kHz**
Targets (Hydrogen, Helium, Carbon etc)

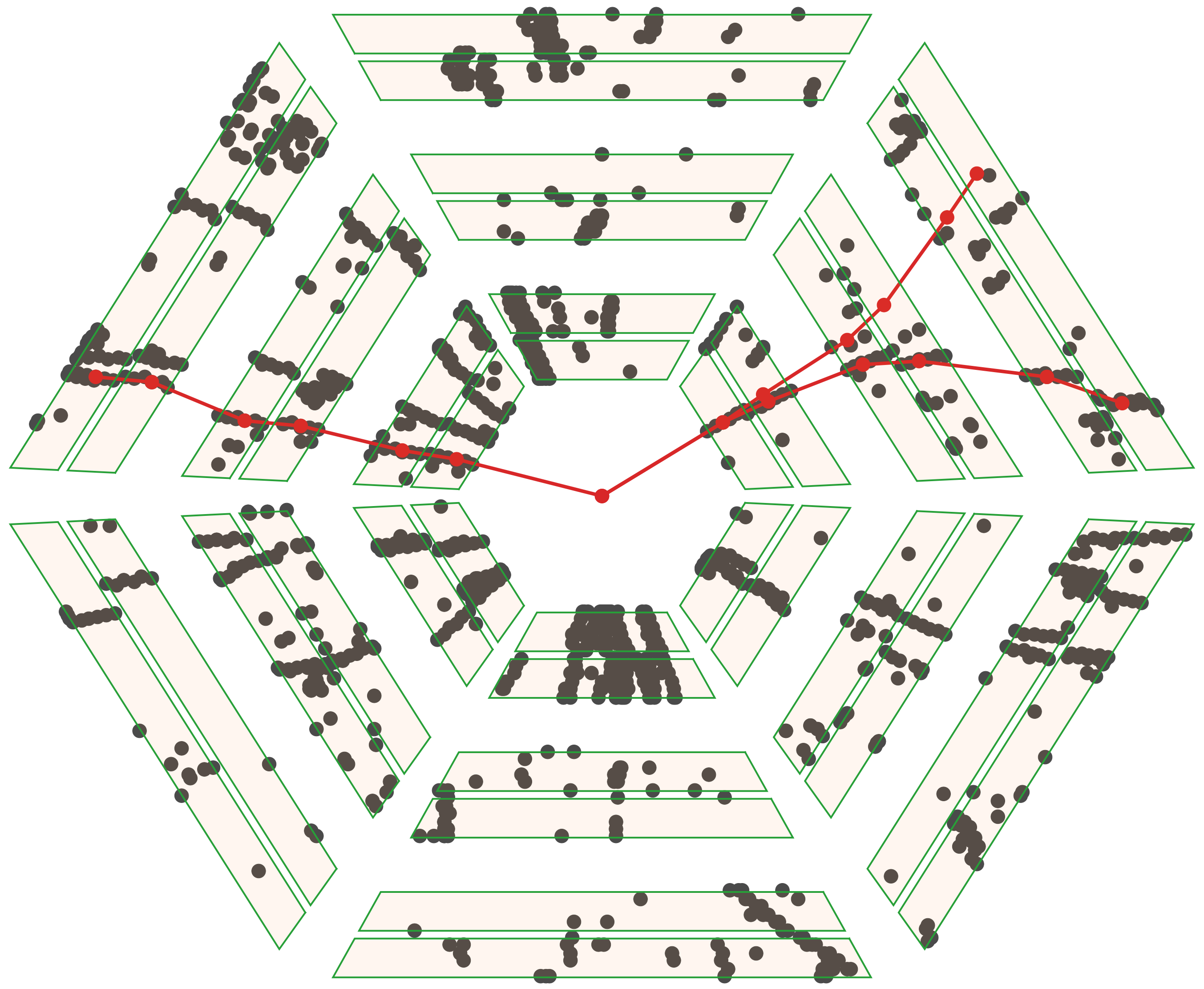
Detecting Scattered Particles with Drift Chambers

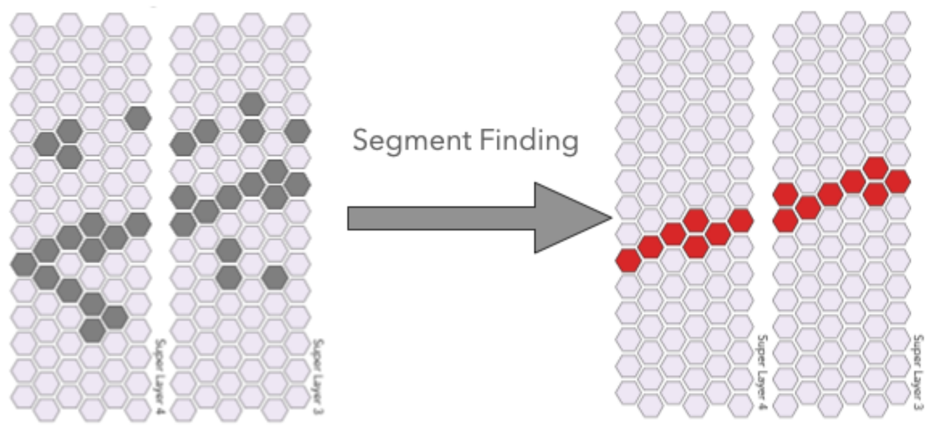
Measuring Tracks from the interaction
The particle tracks are measured in Drift Chambers (DC), embedded in a magnetic field. By measuring the curvature of the track the momentum (speed) of the particle is calculated.



Drift Chambers
There are six drift chambers surrounding the interaction point. Each drift chamber consists of six sections, each section contains six layers of wires.

Signals in Drift Chambers (Front View)
Due to the high intensity of the experiment, the drift chambers detect many signals. Some are part of a track and some are noise. Data processing procedure must identify which hits are traces of a particle.





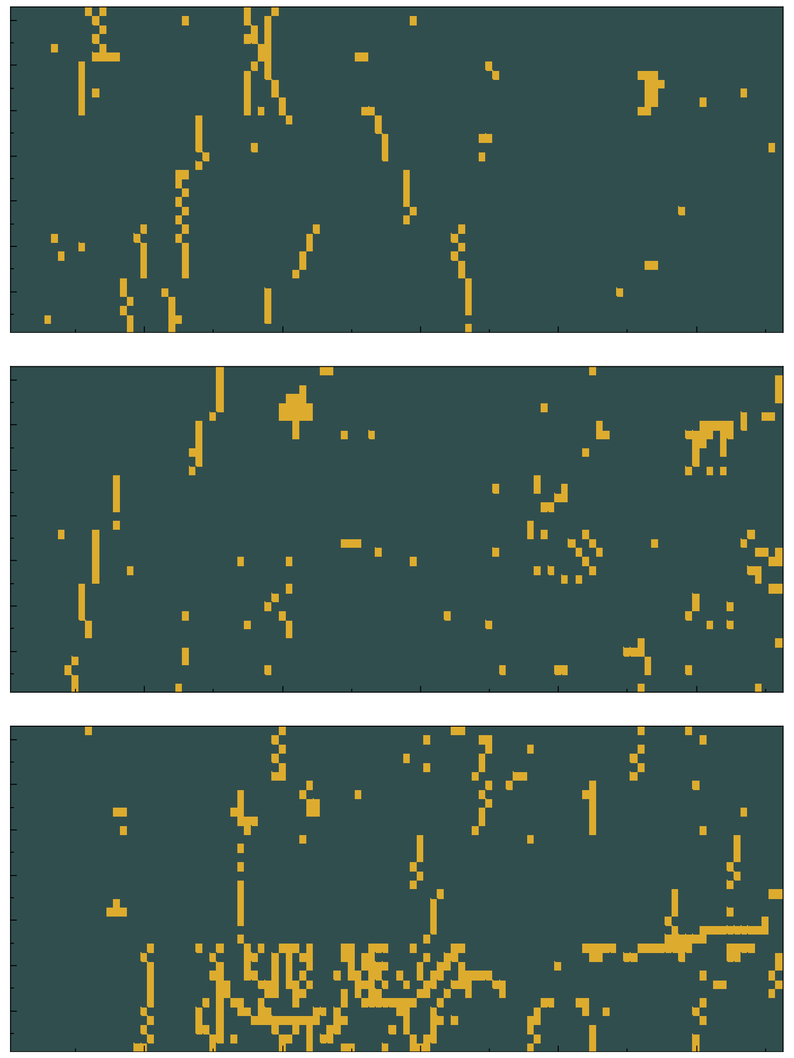
- Find segments in each super layer (remove noise)
- Combine 6 segments (one from each super layer) to make a list of possible tracks
- Identify correct combinations of segments that represent a track

Track Candidate Finding
The track candidate identification is done using segment positions in 6 superlayers as input to the network as a vector. (There are 112 wires in each layer)

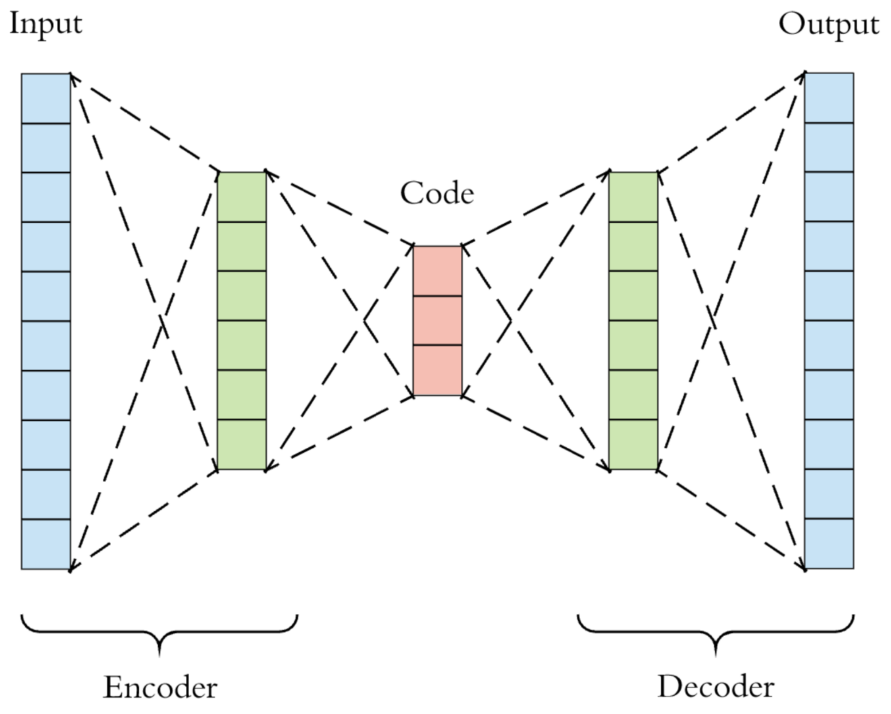
$$V = [W_{c1}, W_{c2}, W_{c3}, W_{c4}, W_{c5}, W_{c6}]\ ,\ W_{ci} = [1 - 112]$$

Using Artificial Intelligence in Reconstruction (De-Noising)

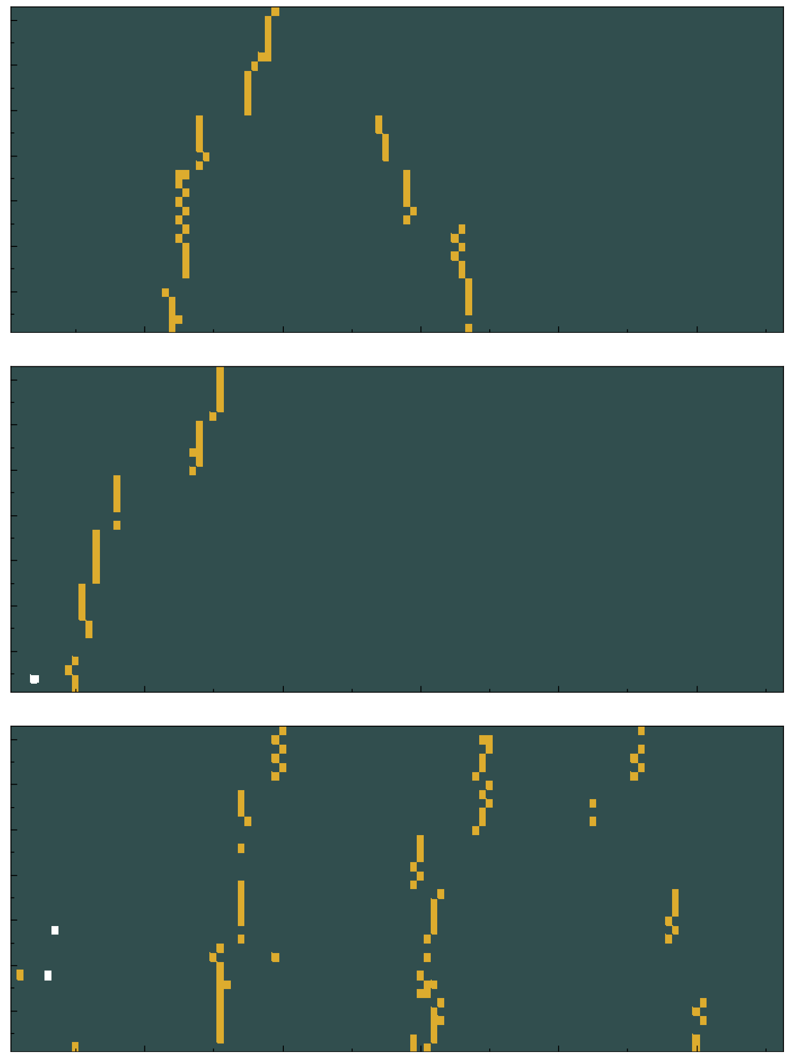
AI image denoising employs artificial intelligence and machine learning techniques to reduce or eliminate noise in images

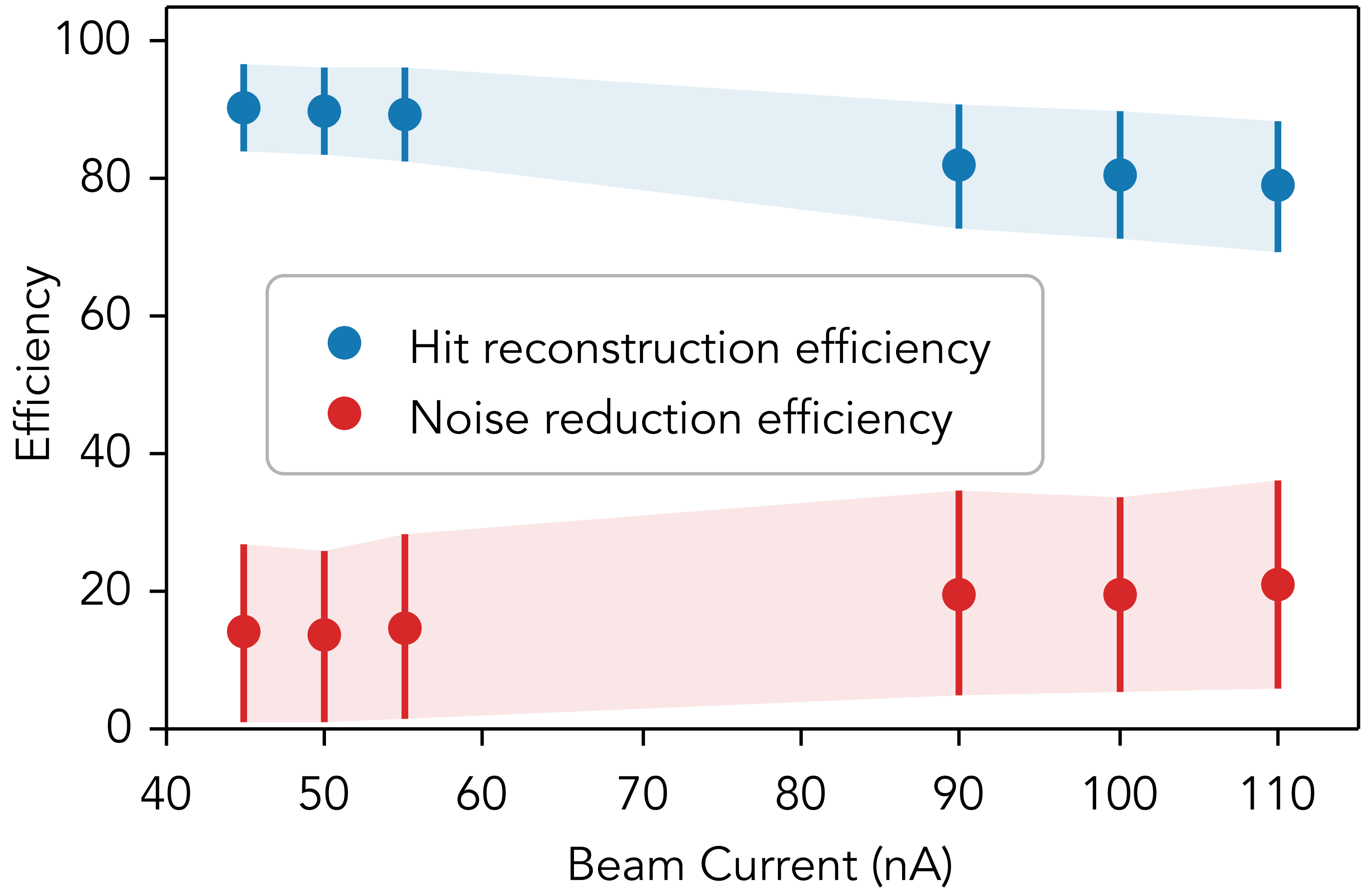


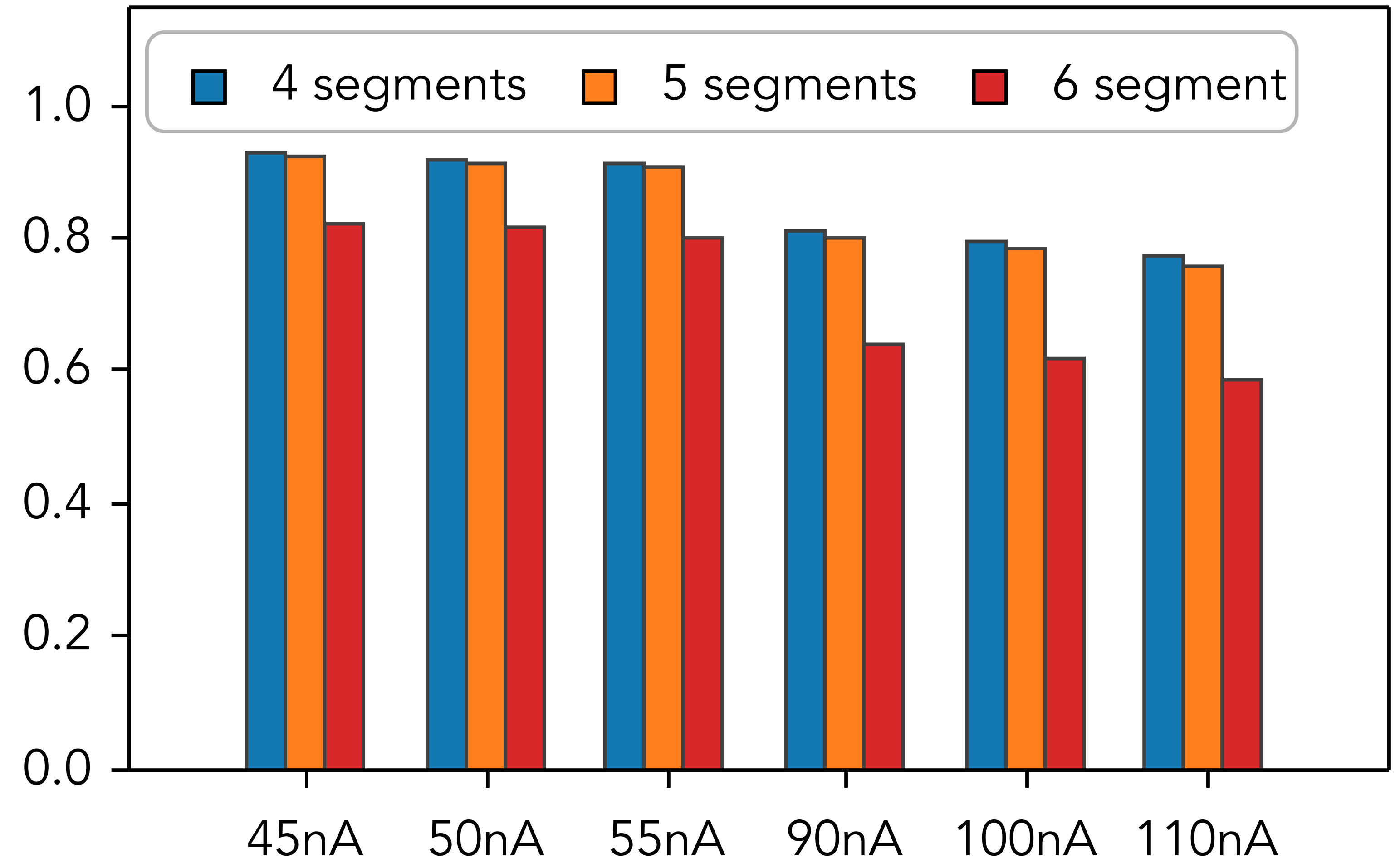
De-Noising CNN Auto-Encoder



Retains >95% of hits belonging to tracks, and removes >90% of background hits

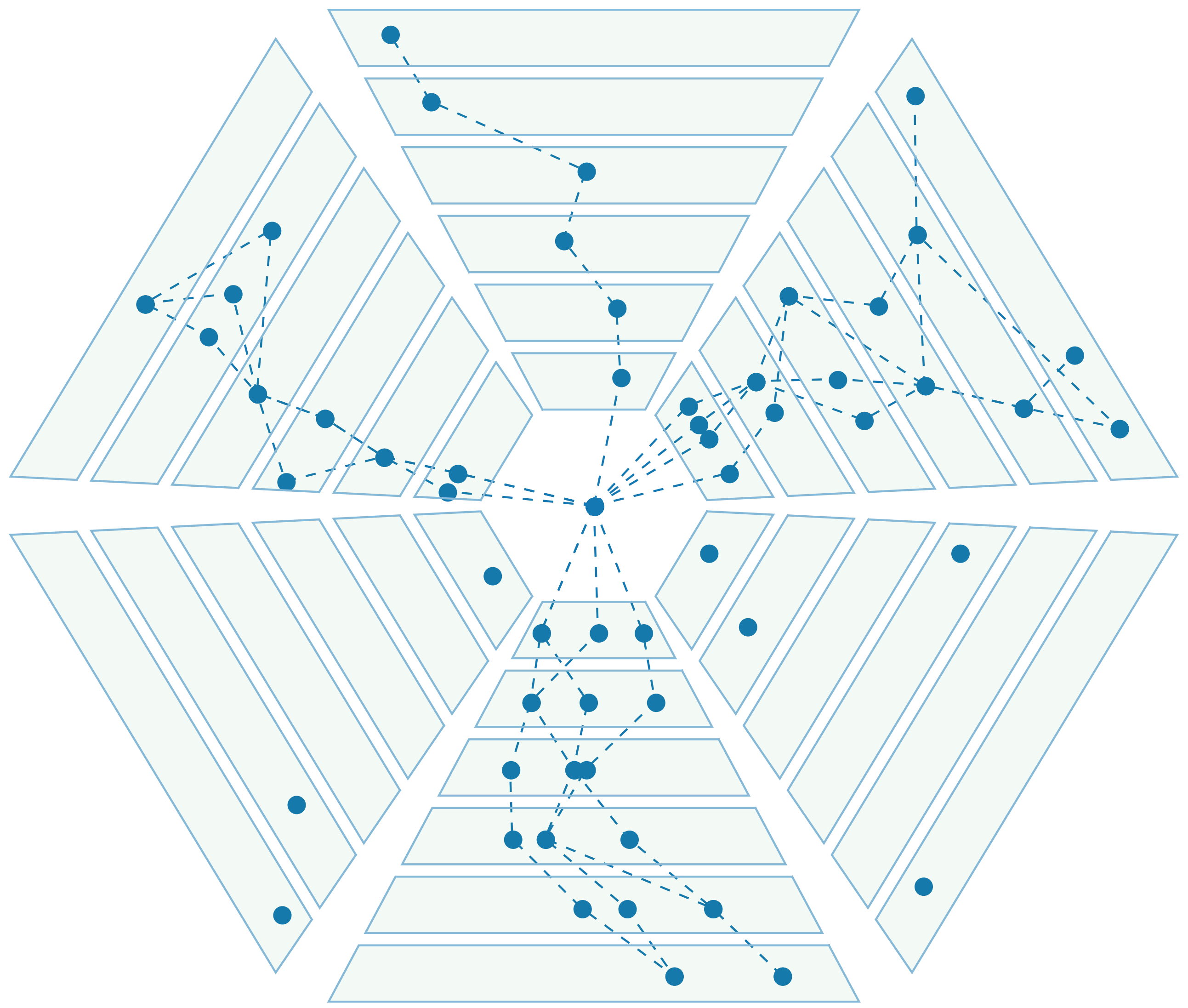


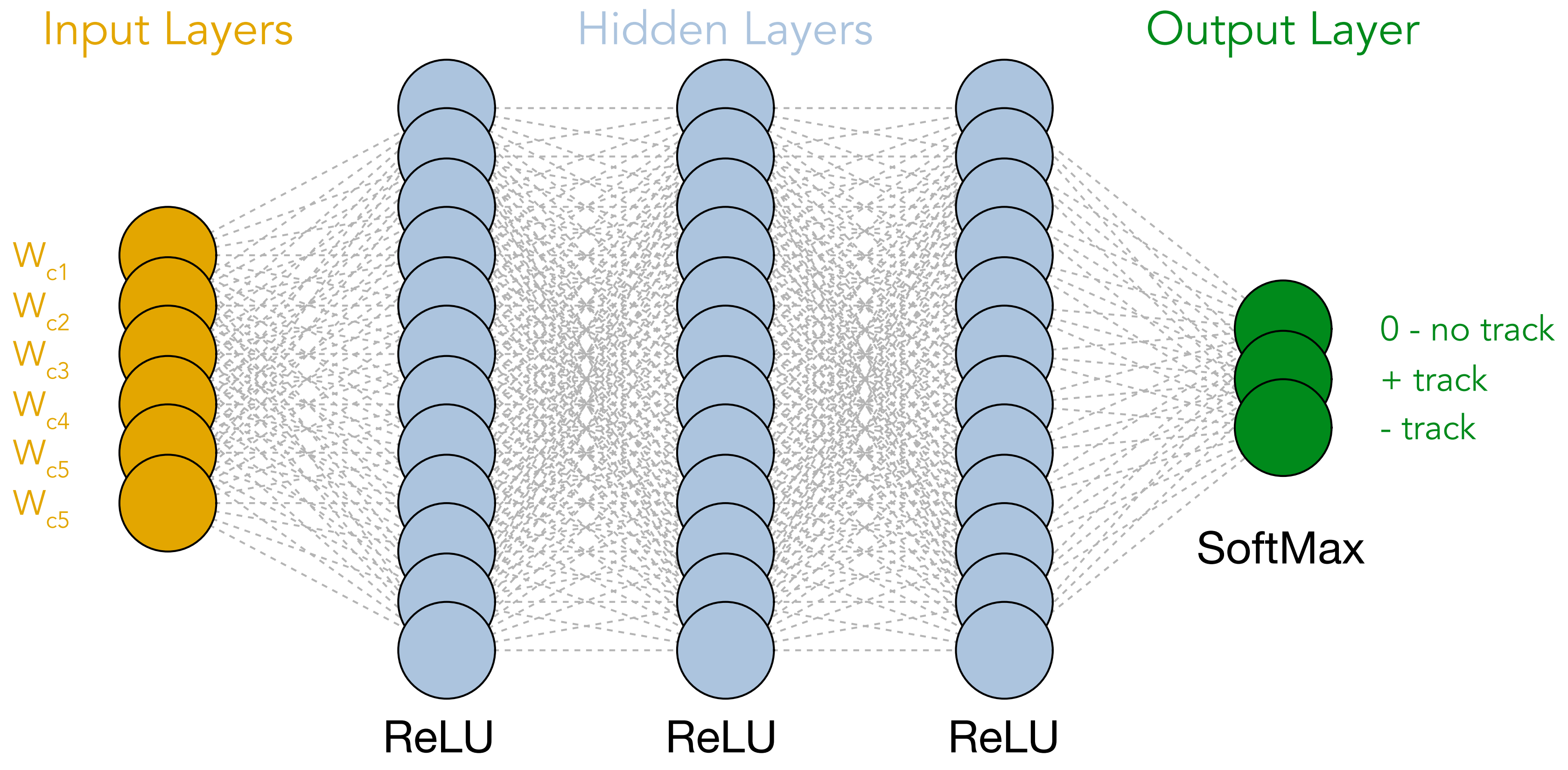


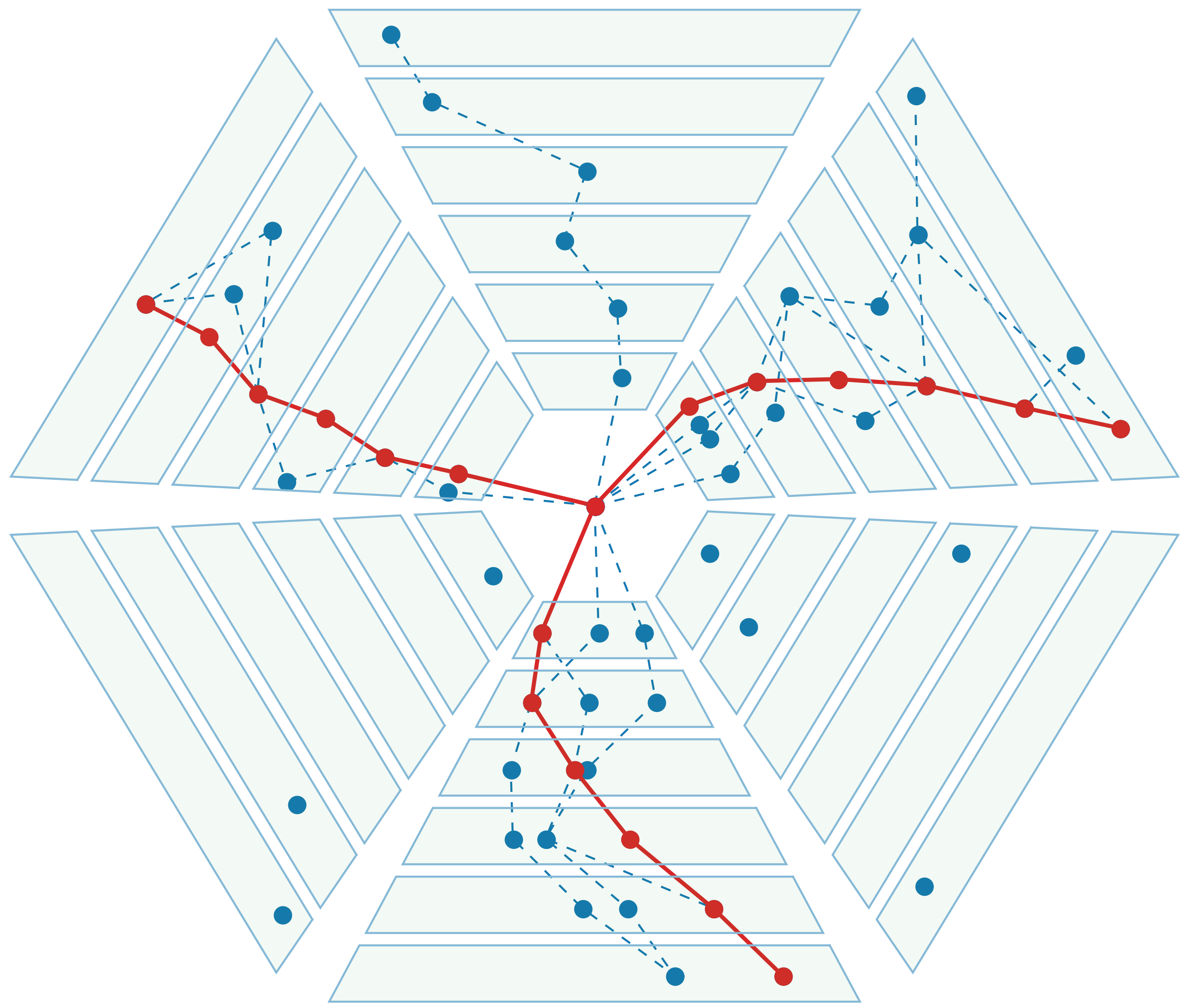


The CNN Auto-Encoder learns to remove 85% of noise hits
While preserving >90% of the signal hits.

Using Artificial Intelligence in Reconstruction (Track Classifier)

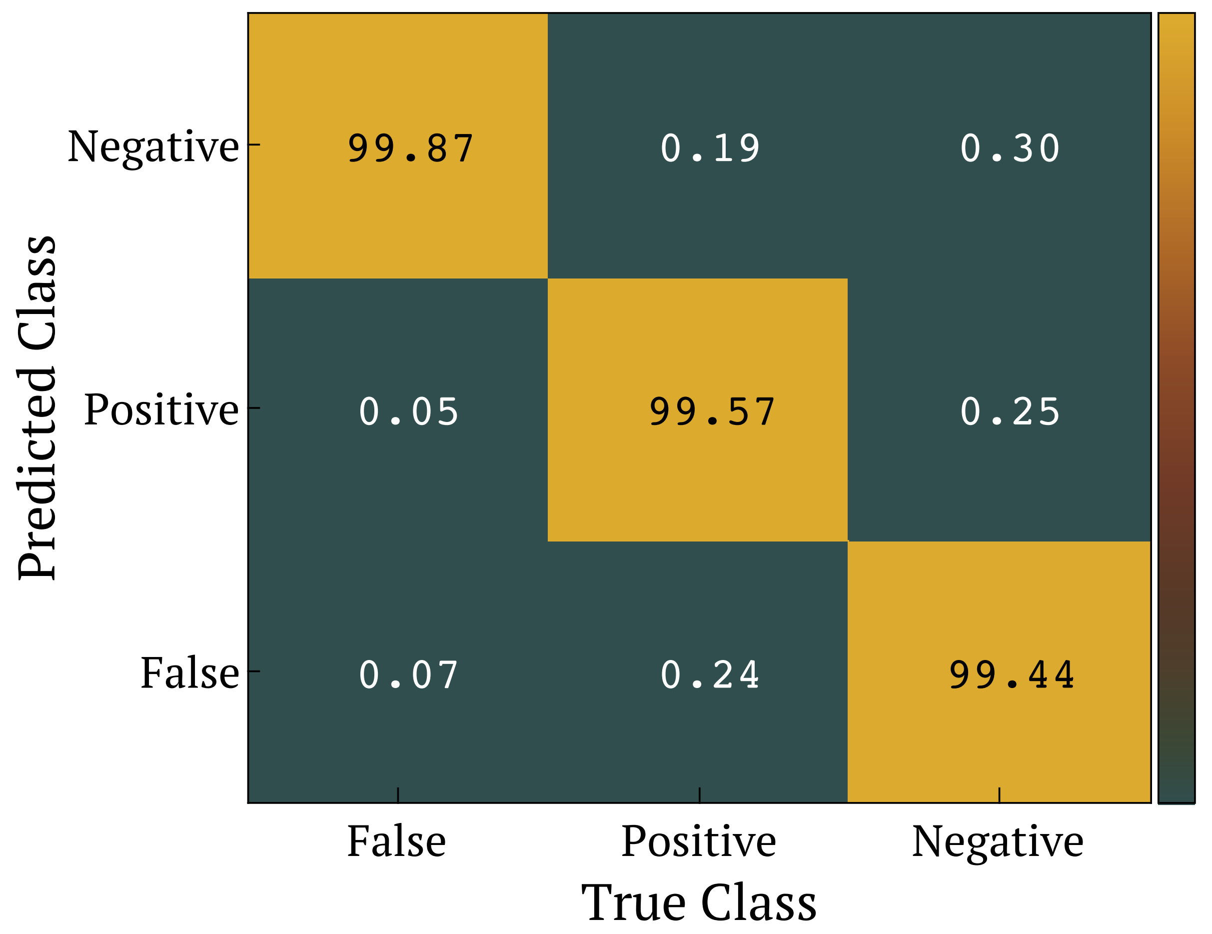






- Input: W [1..6] - average wire position of the segment
- Output: [false track, positive track, negative track]

- A Neural Network is trained to recognize patterns of segment combinations
- The track classifier assigns a probability that the track candidate is positive, negative, or a false track.
- The network is trained on reconstructed data where the right combinations are already found, and false combinations of segments are generated by interchanging clusters from a different track
- Conventional tracking takes **~250 msec** (**~4 Hz**) per event to classify the right track candidates (fitting through the magnetic field)
- The AI identifies correct cluster combinations at a rate of **~8 kHz** (depending on the number of combinations, changes slightly with luminosity)



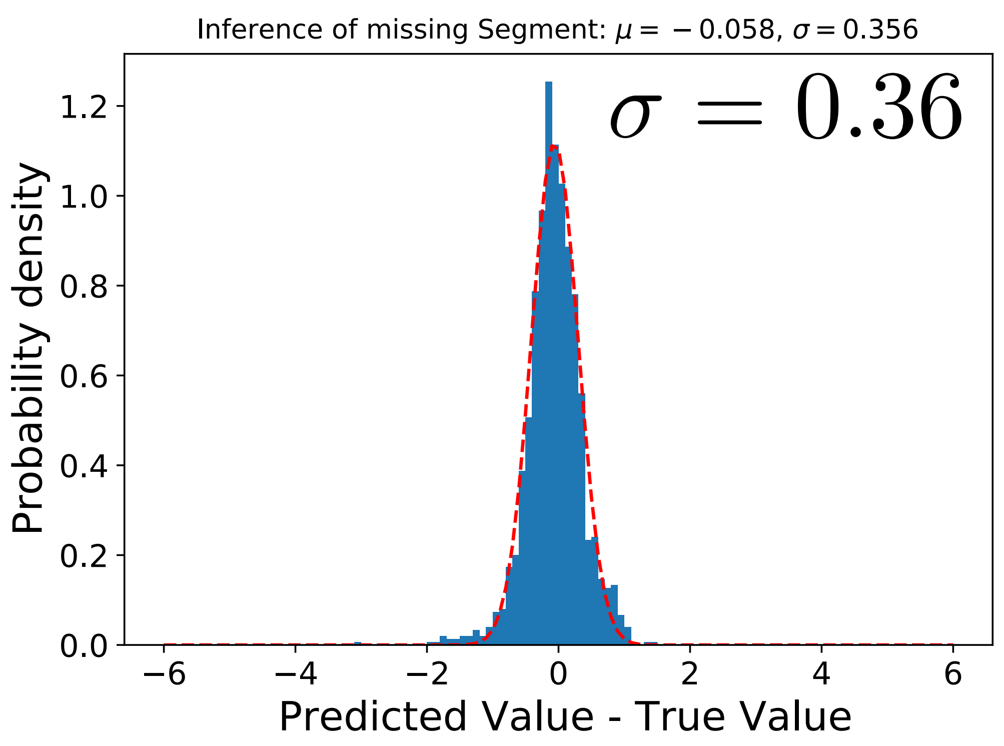
Using Artificial Intelligence in Reconstruction (Track Classifier)

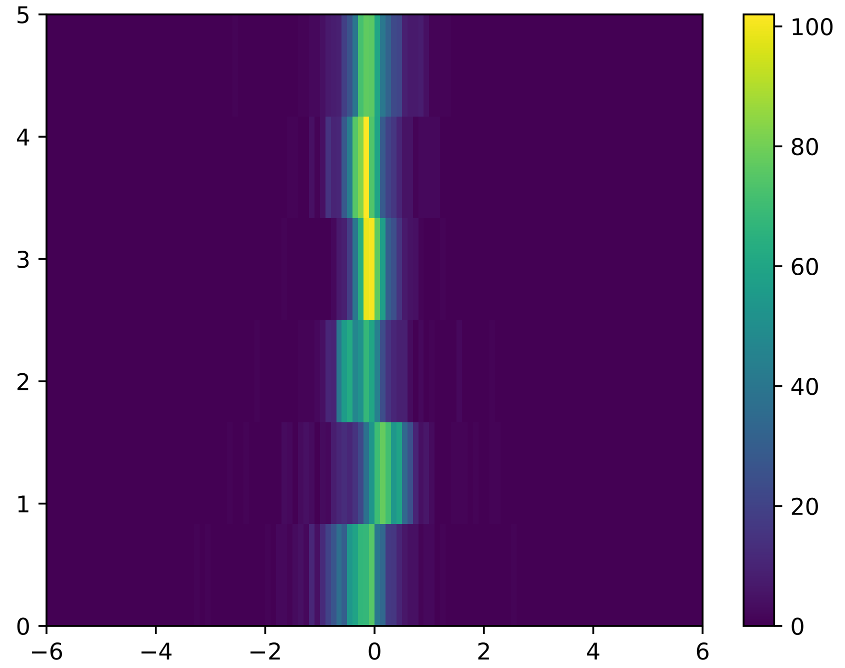
Drift Chamber Inefficiencies:

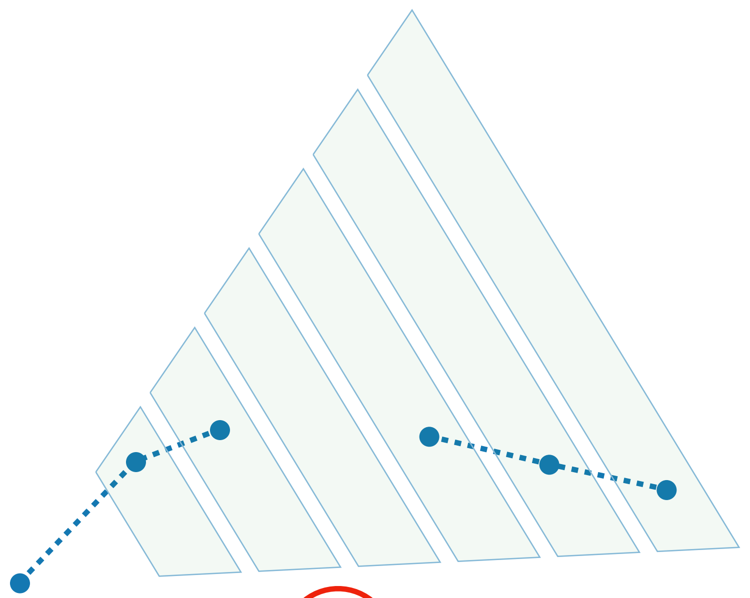
- Inefficiencies in the Drift Chamber can cause missing segments along the track trajectory
- The cause can be inactive regions of the drift chambers or the segment finding algorithm
- The conventional tracking package can fit tracks using only 5 segments with Kalman-Filter
- The classifier network can classify tracks with 6 segments

Prediction Accuracy:

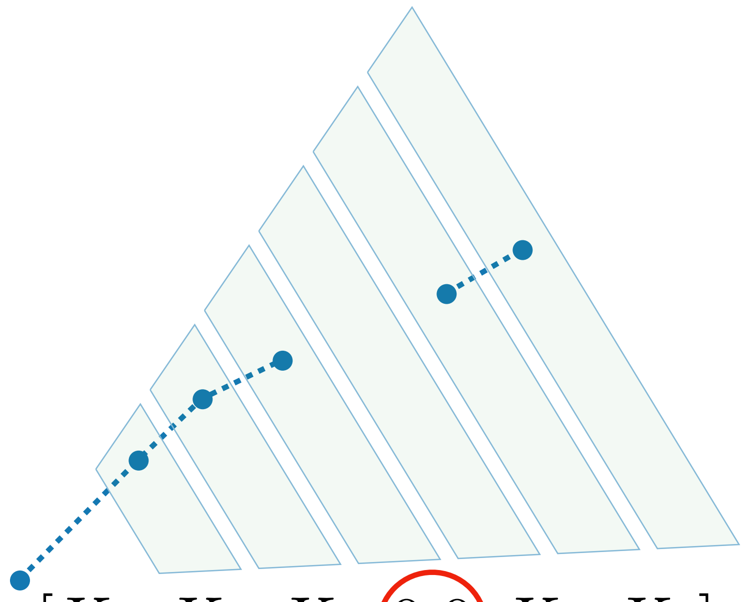
The average prediction of segment position is within 1 wire (0.36)—no significant dependence on the superlayer.



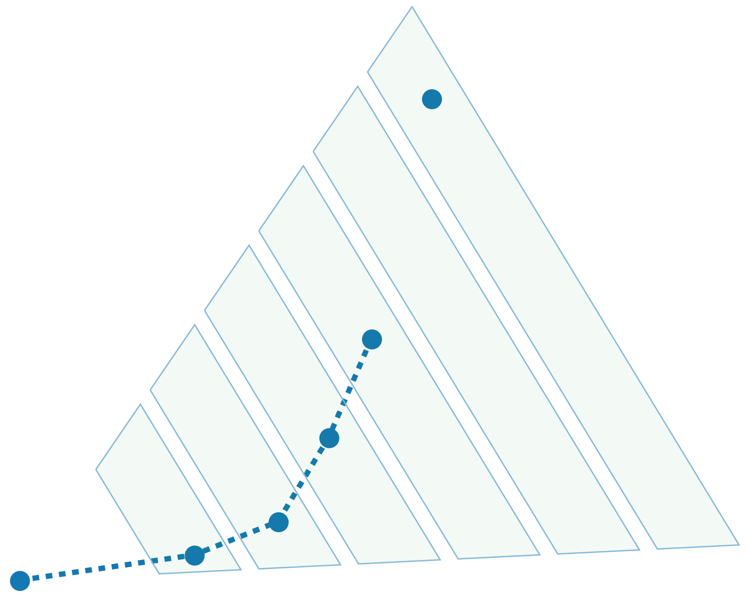




$[X_1, X_2, 0.0, X_4, X_5, X_6]$



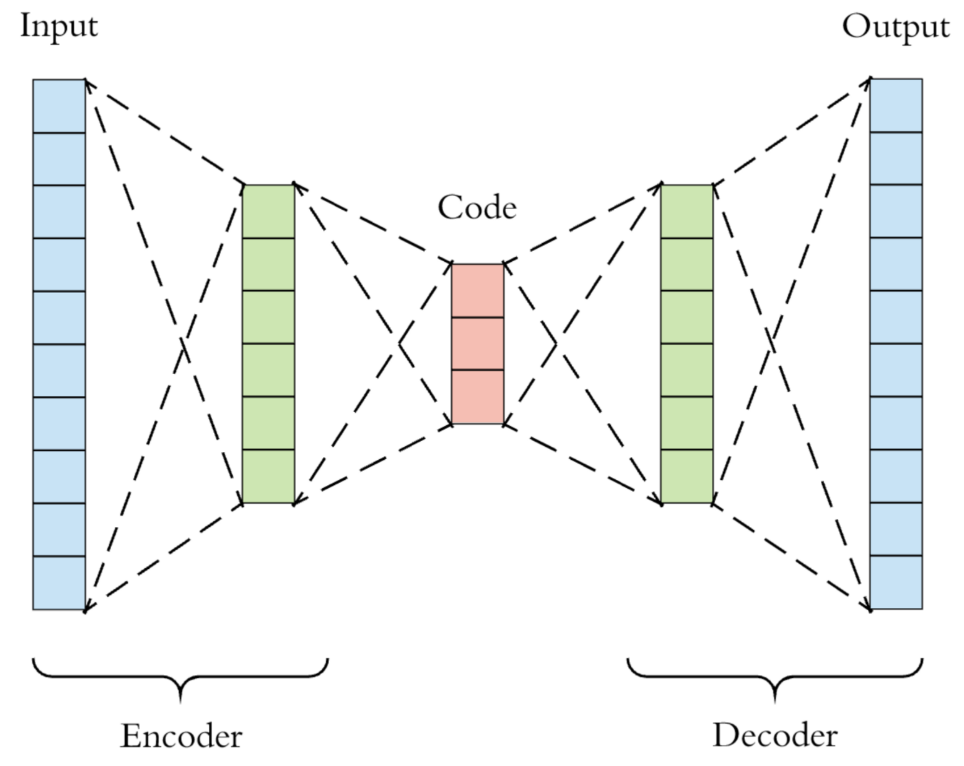
$[X_1, X_2, X_3, 0.0, X_5, X_6]$



$[X_1, X_2, X_3, X_4, 0.0, X_6]$

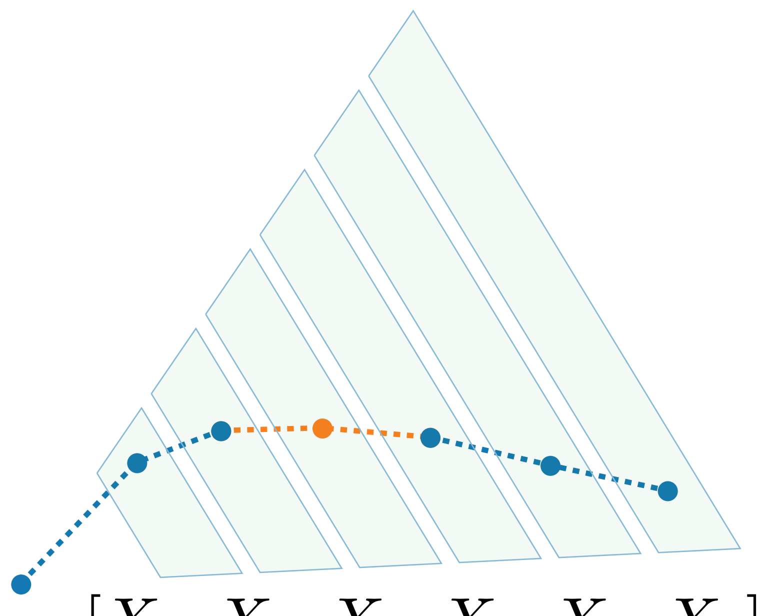
Auto-Encoder:

An autoencoder is a type of artificial neural network used to learn efficient codings of data. The input and output are the same size.

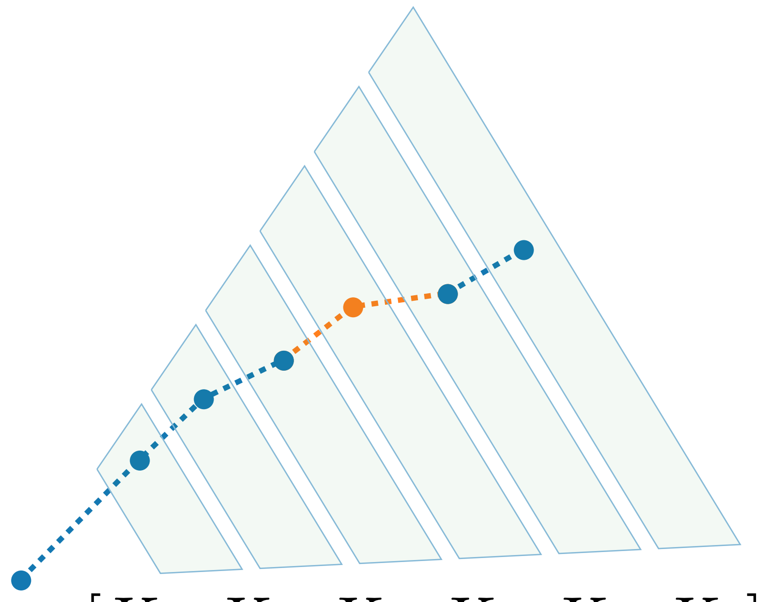


Training Procedure:

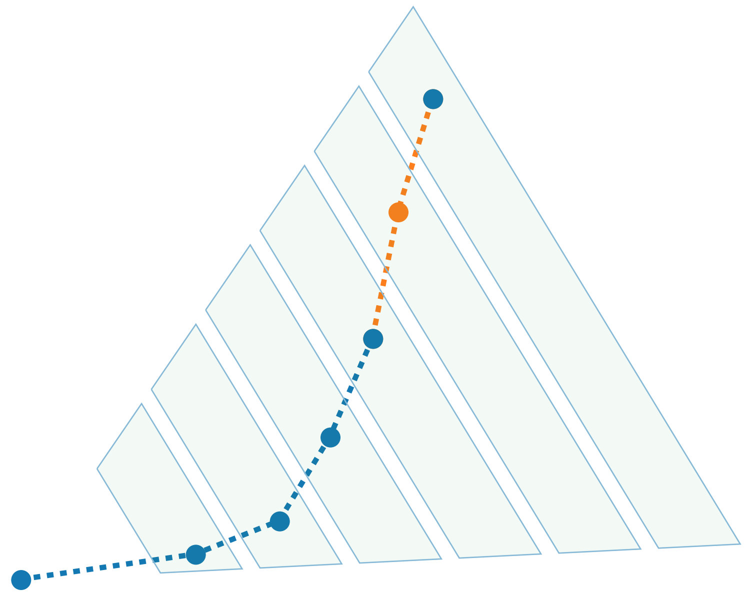
Use reconstructed tracks with 6 segments. Remove one of the segments (random superlayer) in the input data, and keep it in the output data. The network learns to predict the missing segment position.



$[X_1, X_2, X_3, X_4, X_5, X_6]$

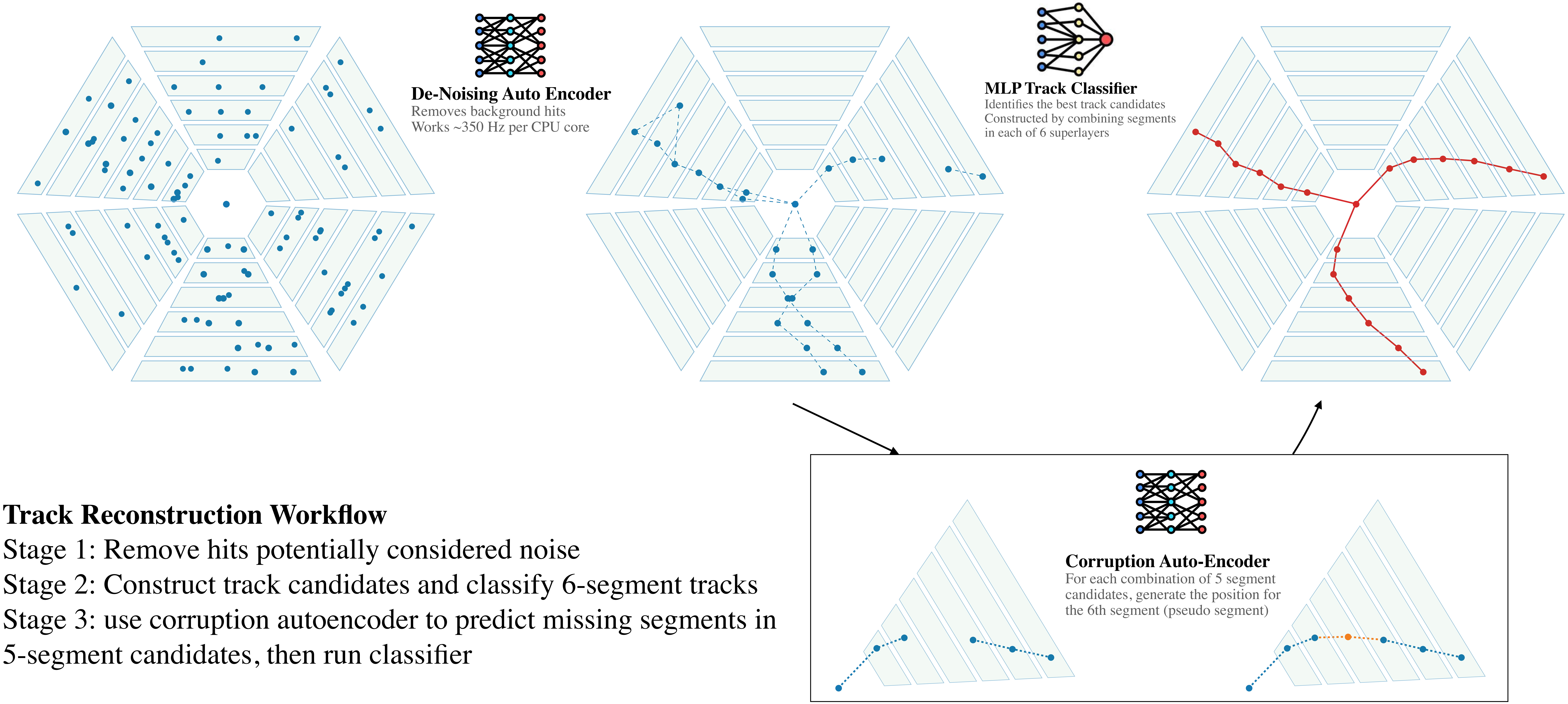


$[X_1, X_2, X_3, X_4, X_5, X_6]$



$[X_1, X_2, X_3, X_4, X_5, X_6]$

Track Reconstruction Workflow



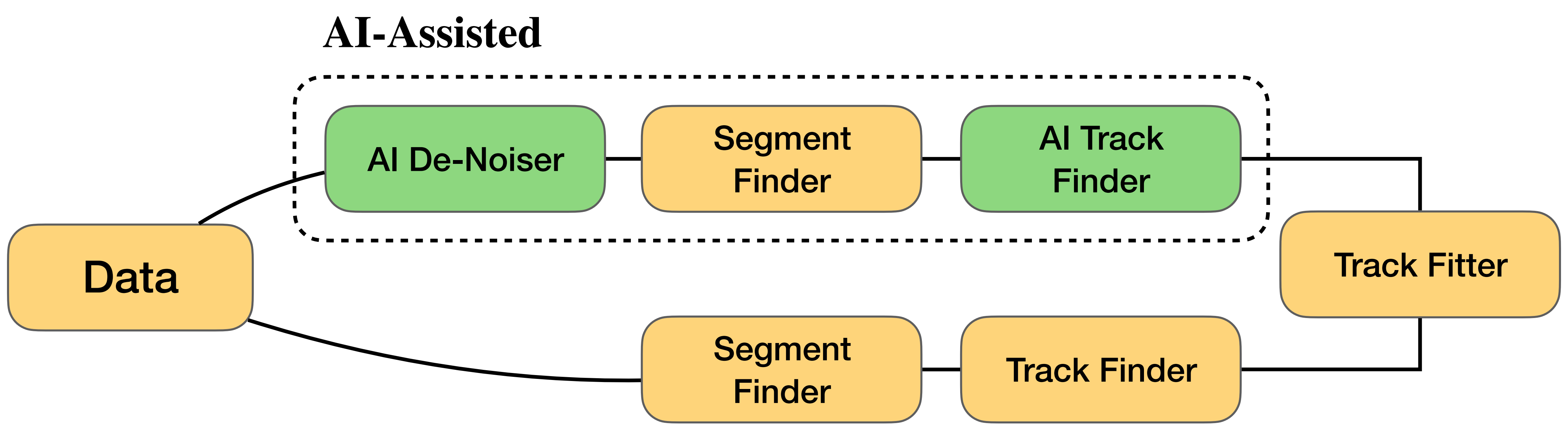
Track Reconstruction Workflow

Stage 1: Remove hits potentially considered noise

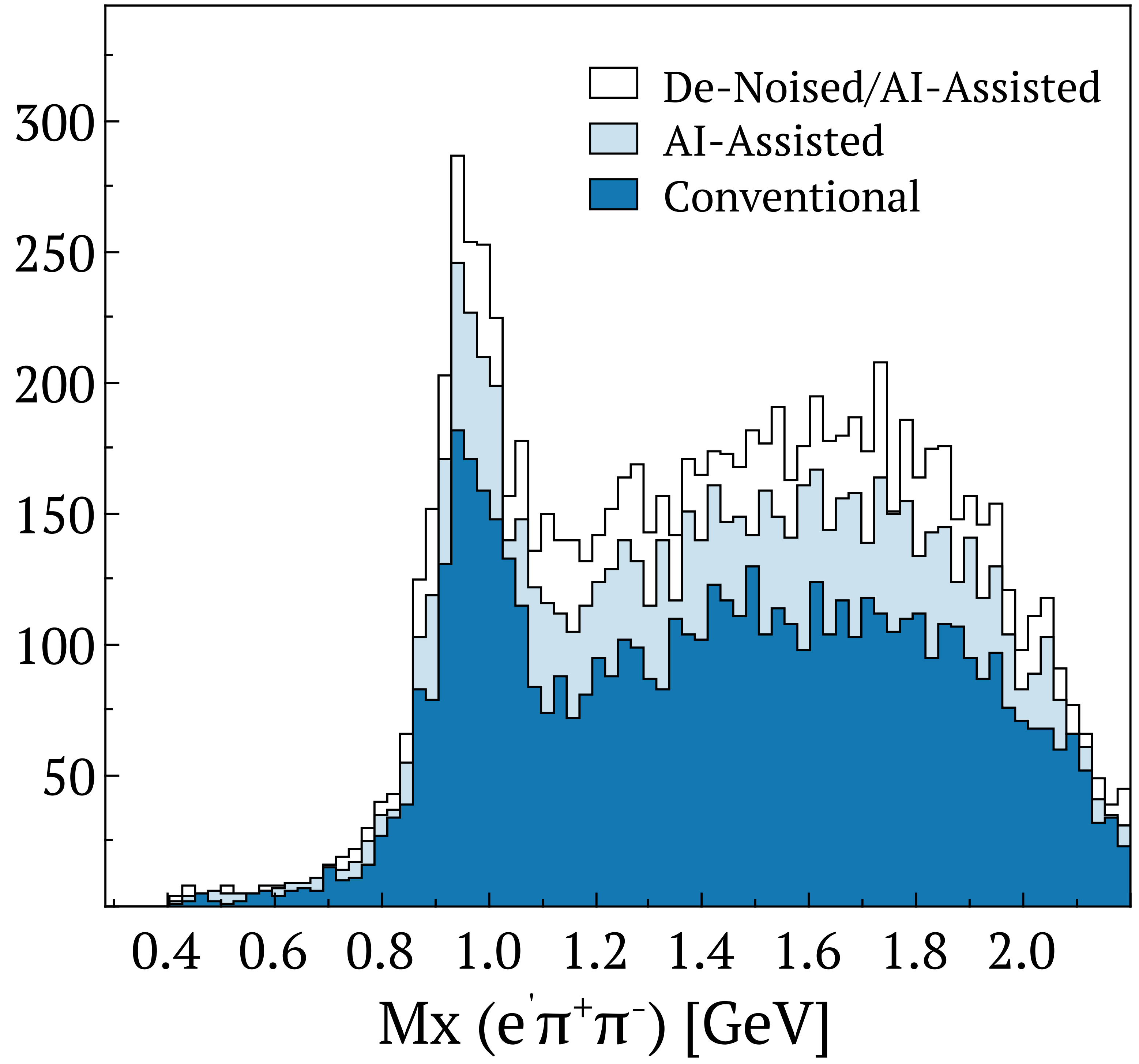
Stage 2: Construct track candidates and classify 6-segment tracks

Stage 3: use corruption autoencoder to predict missing segments in 5-segment candidates, then run classifier

AI assisted Tracking impact

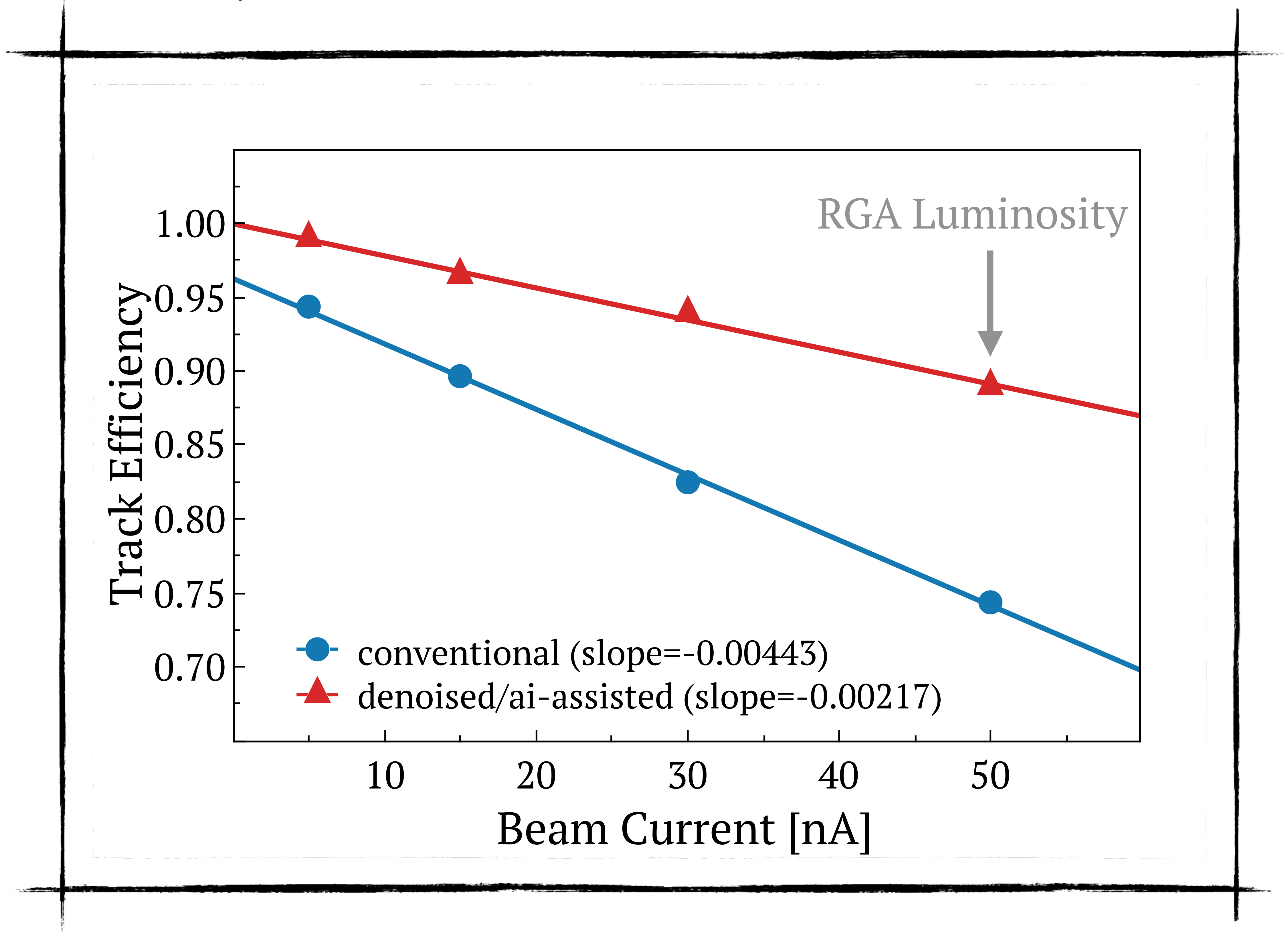


$$ep \rightarrow e'\pi^+\pi^-(p)$$

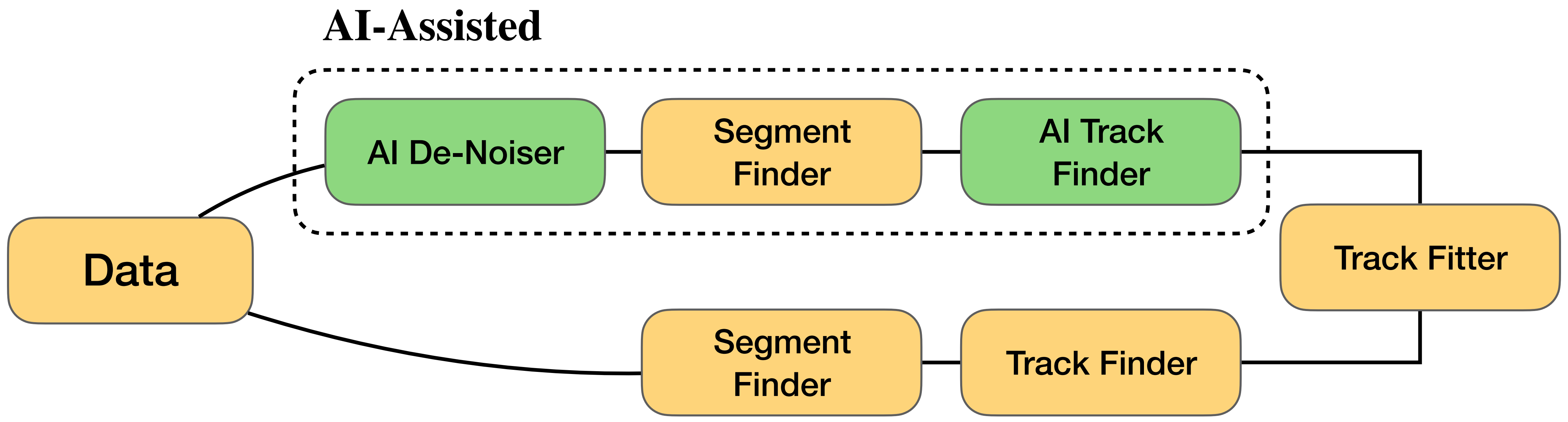


Artificial Intelligence Approach:

- Two parallel reconstruction workflows implemented for CLAS12
- The AI workflow uses de-noising and AI track candidate identification from reconstructed segments
- The AI-assisted reconstruction results in a ~60% increase in statistics for 3-particle final states
- The AI-assisted tracking efficiency degrades slower as a function of luminosity (beam current)

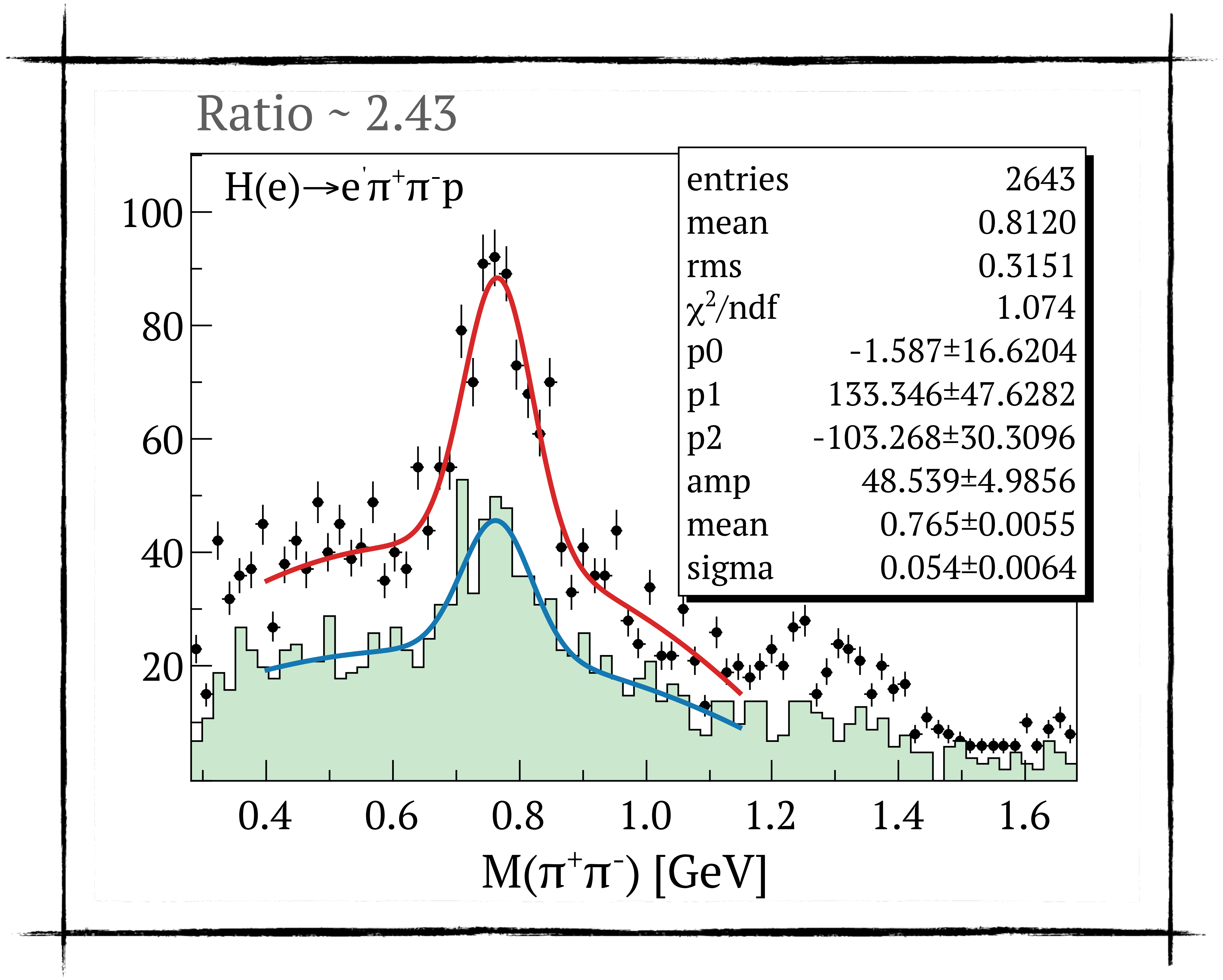


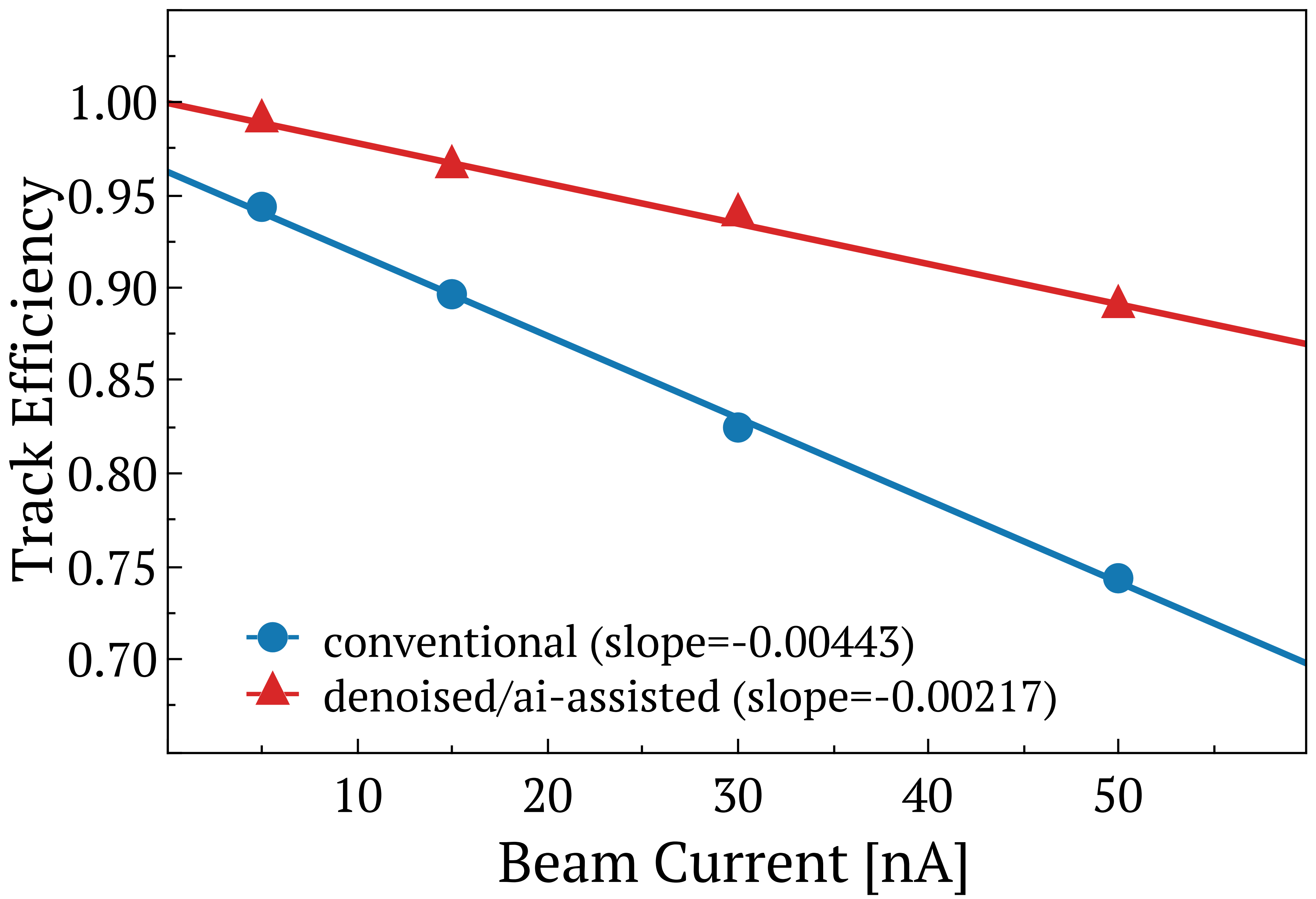
AI assisted Tracking impact



Artificial Intelligence Approach:

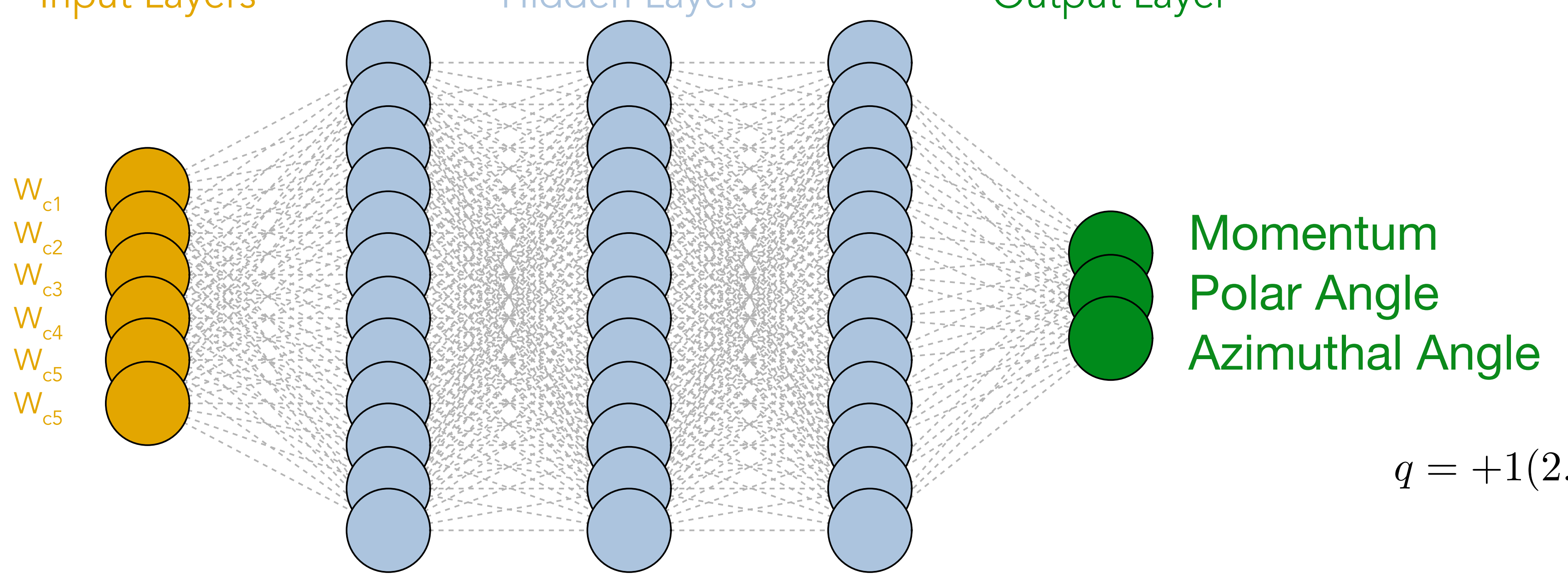
- Two parallel reconstruction workflows implemented for CLAS12
- The AI workflow uses de-noising and AI track candidate identification from reconstructed segments
- The AI-assisted reconstruction results in a ~60% increase in statistics for 3-particle final states
- The AI-assisted tracking efficiency degrades slower as a function of luminosity (beam current)

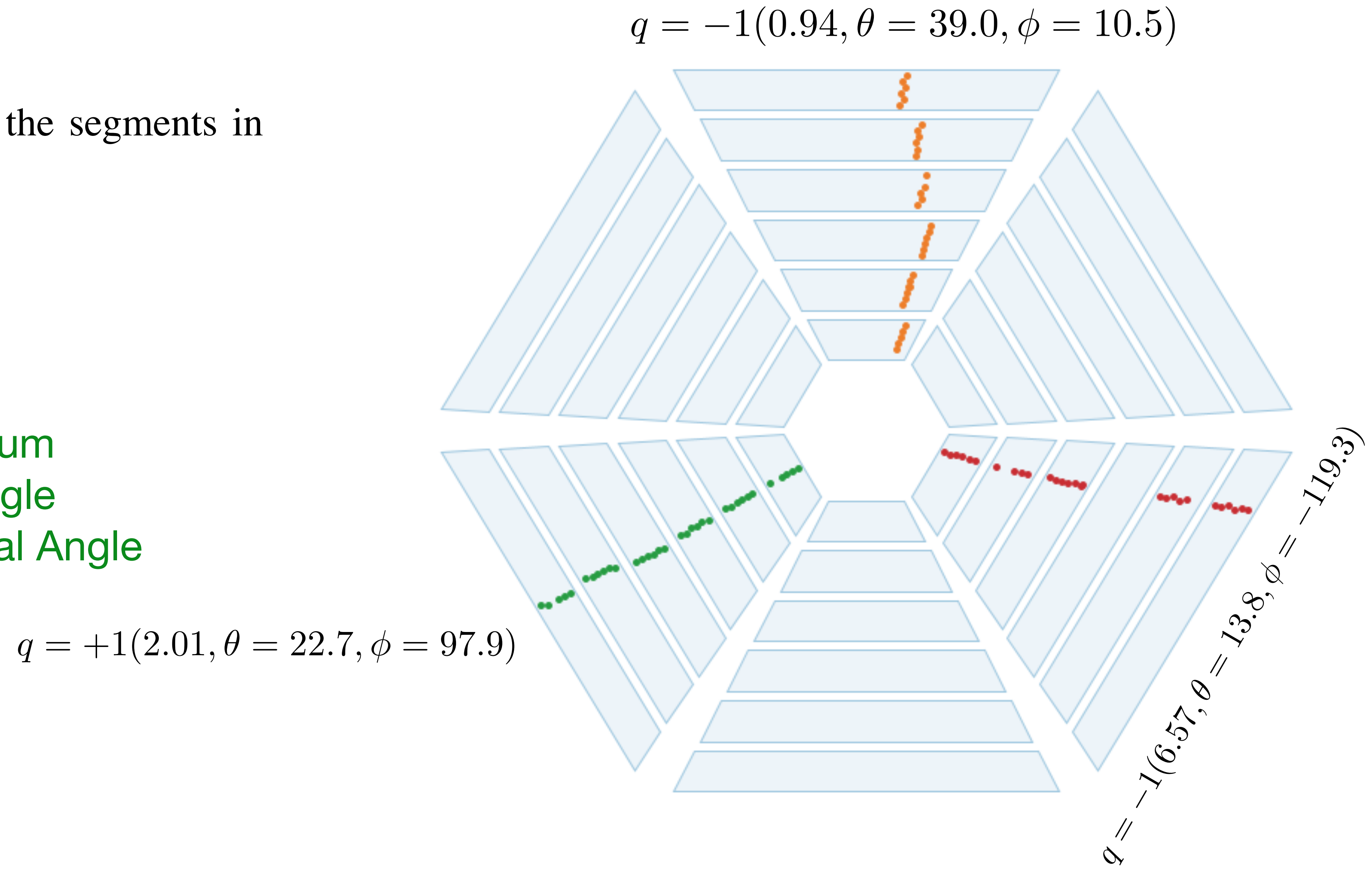


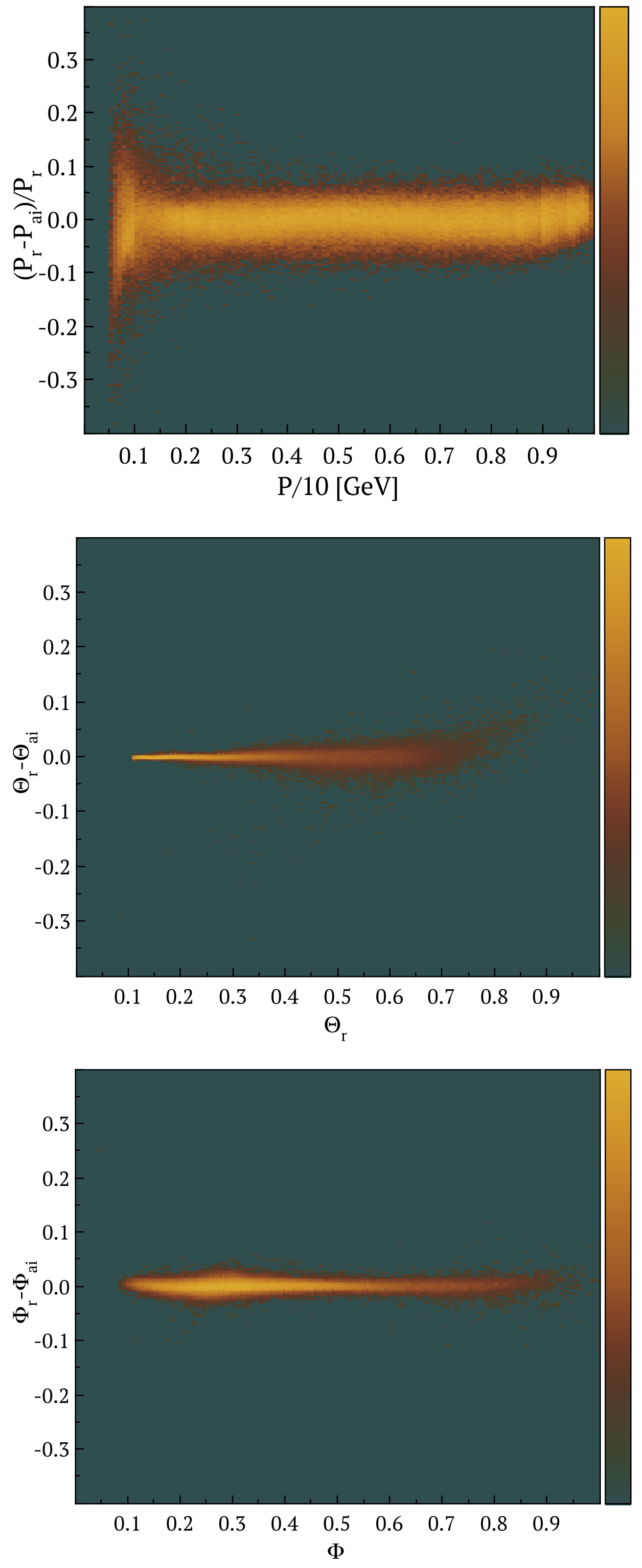


Track Parameter Estimation Network

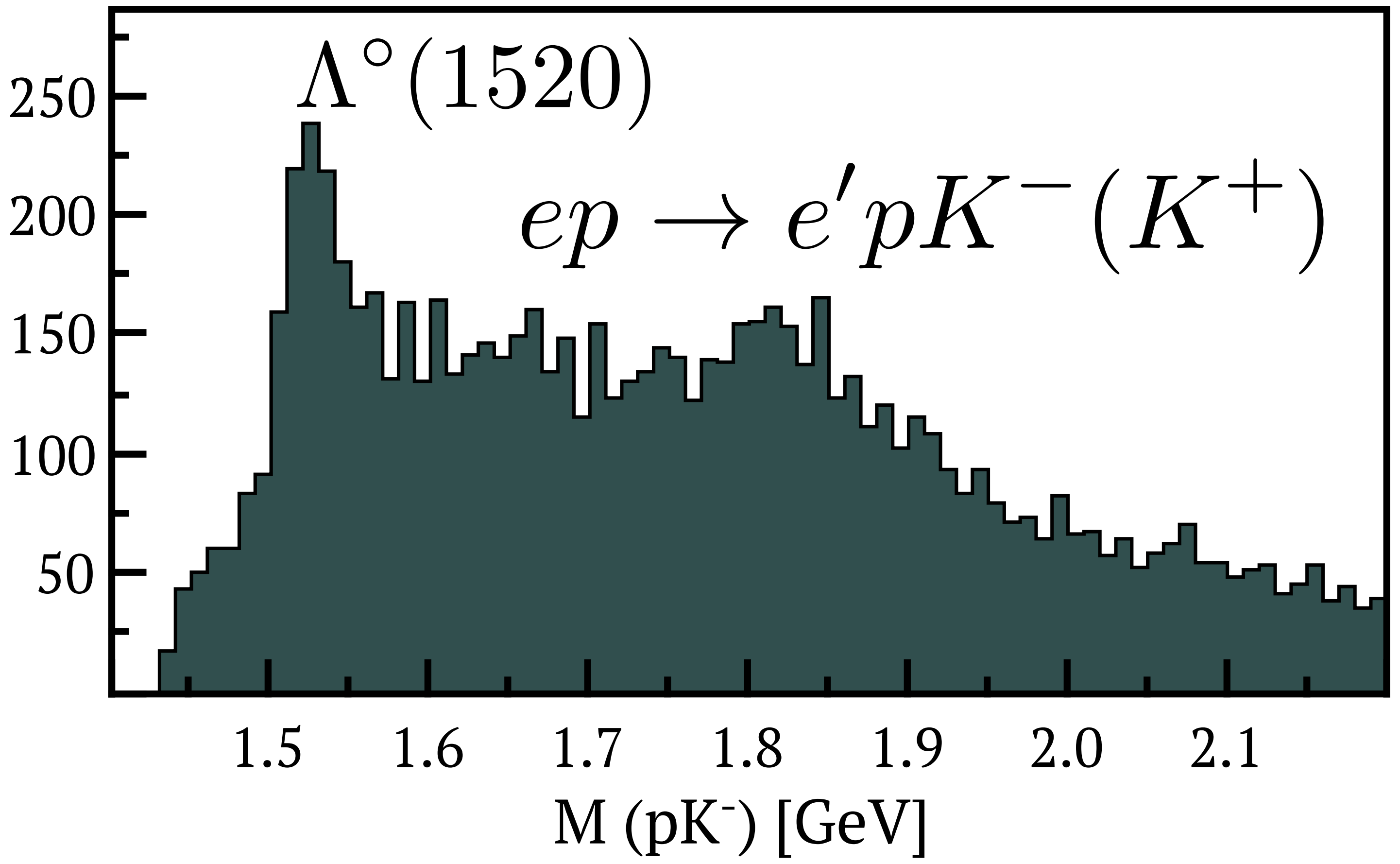
MLP Particle Reconstruction:
A Neural Network to predict particle momentum and directions based on the segments in each super-layer of the drift chamber.

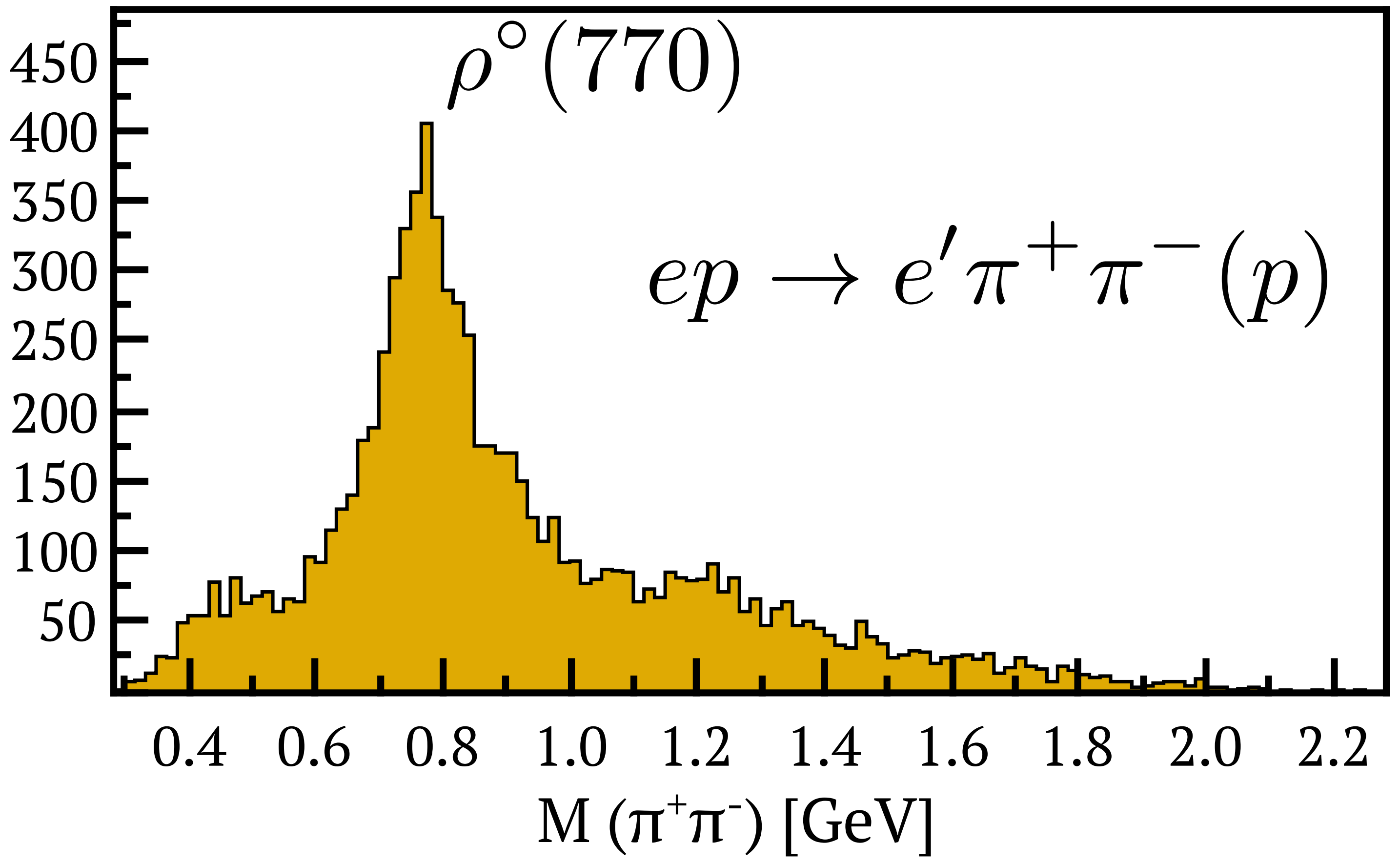






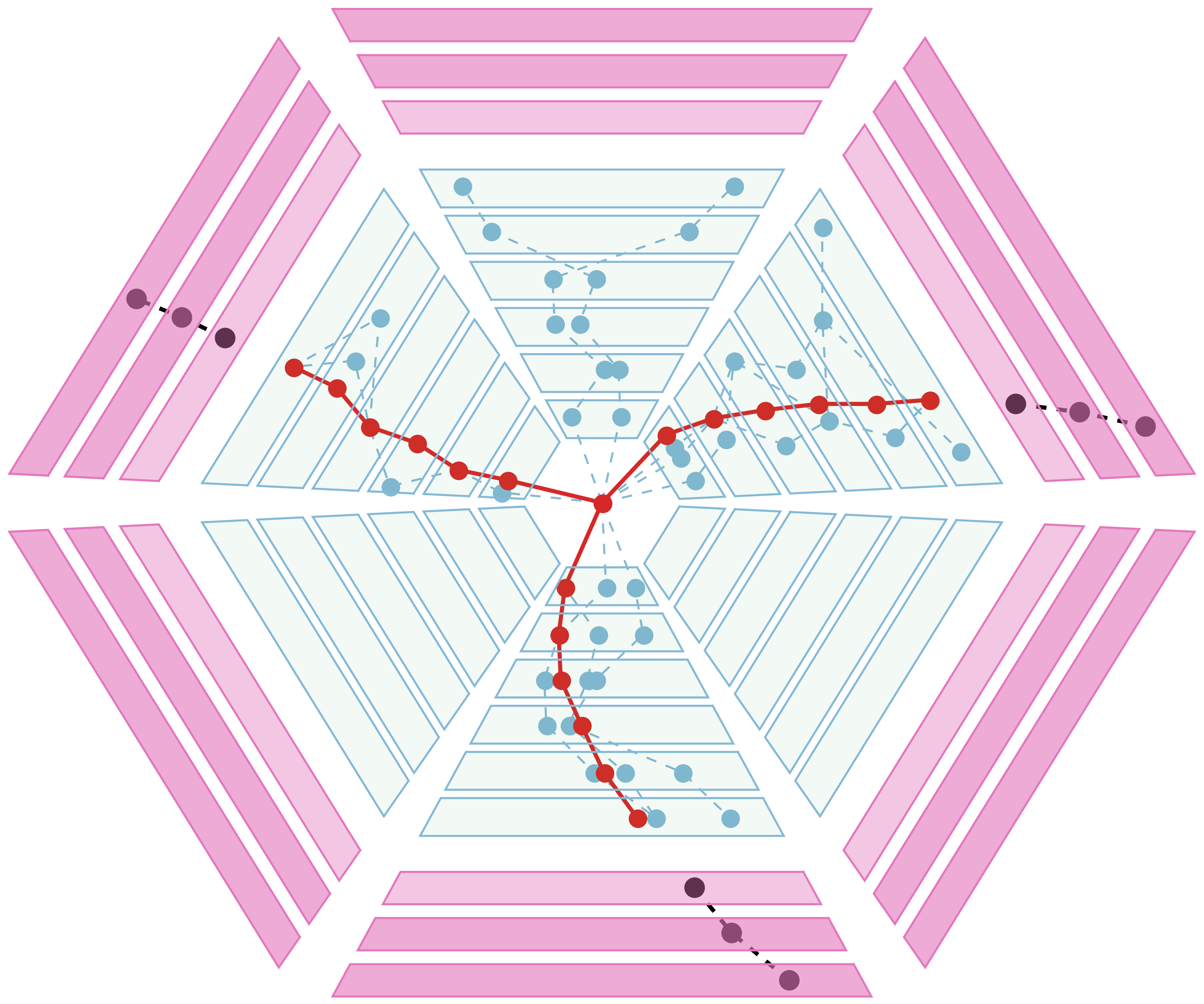
Physics Reactions:
Invariant masses calculated from particles reconstructed by AI, from RAW drift chamber hits.
Reconstruction speed ~24 kHz on a single Laptop.



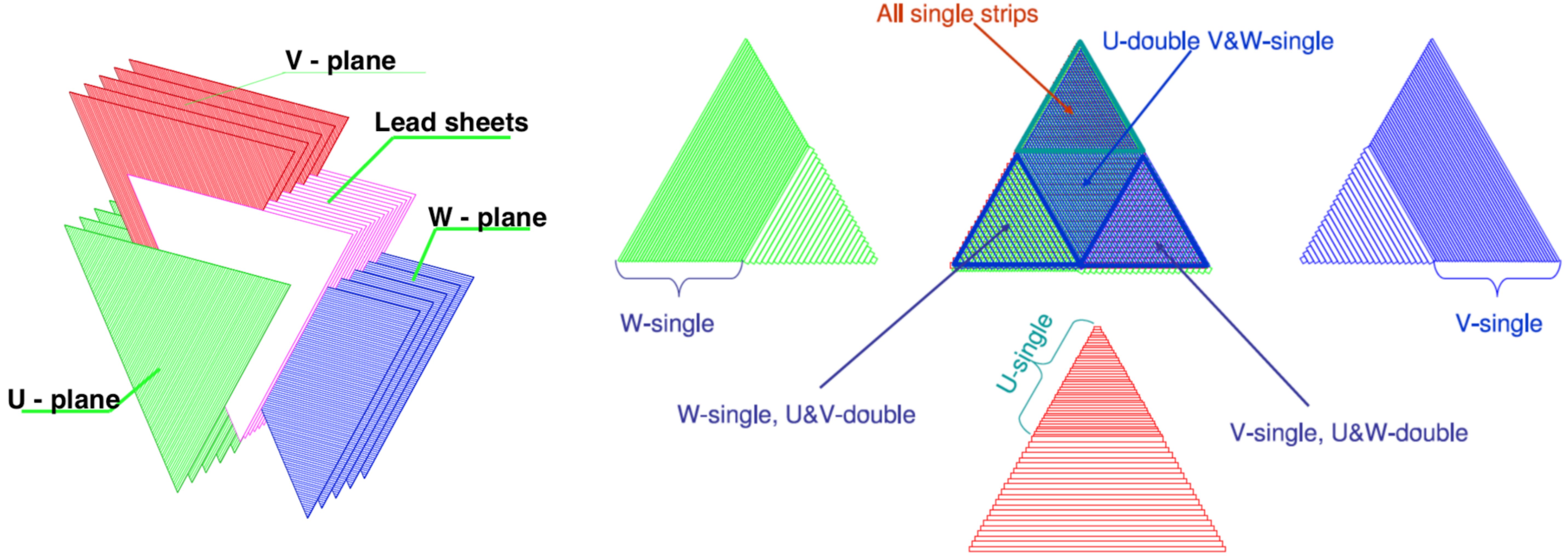


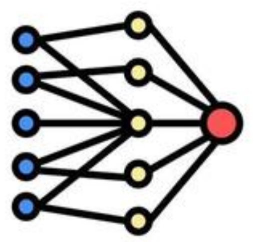
MLP Particle Reconstruction
Predicts particle momentum ~2.4%
Conventional Tracking ~0.05%

Electron Identification with AI



CLAS12 Electromagnetic Calorimeter

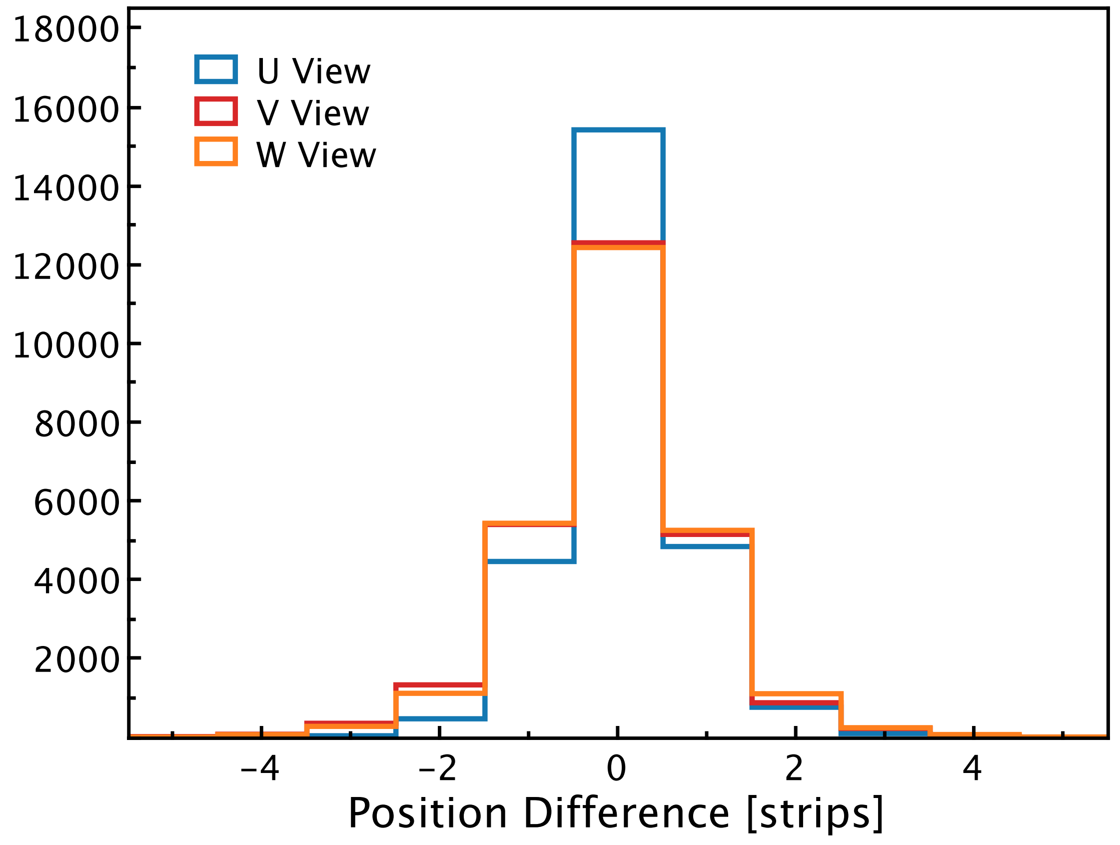


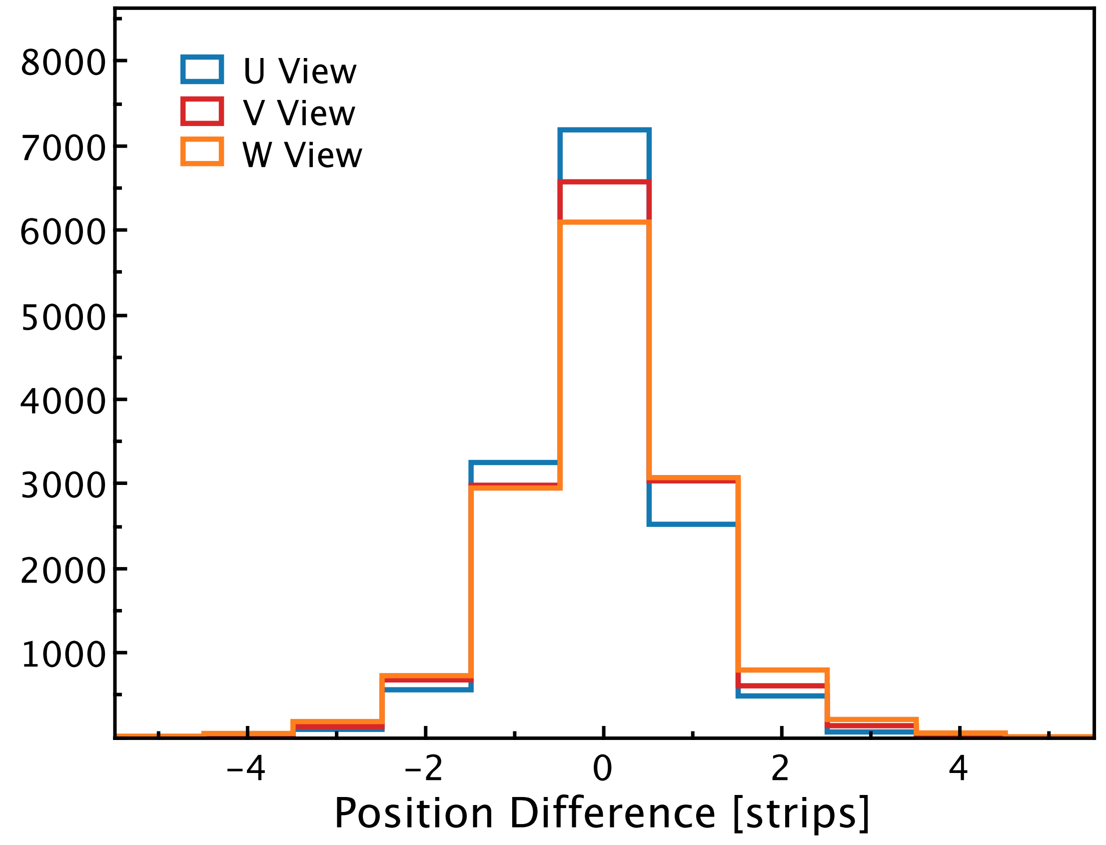


AI Position Prediction:
A Neural Network can predict the strip that the track will hit in the Calorimeter
Prediction accuracy ~1.4 strips

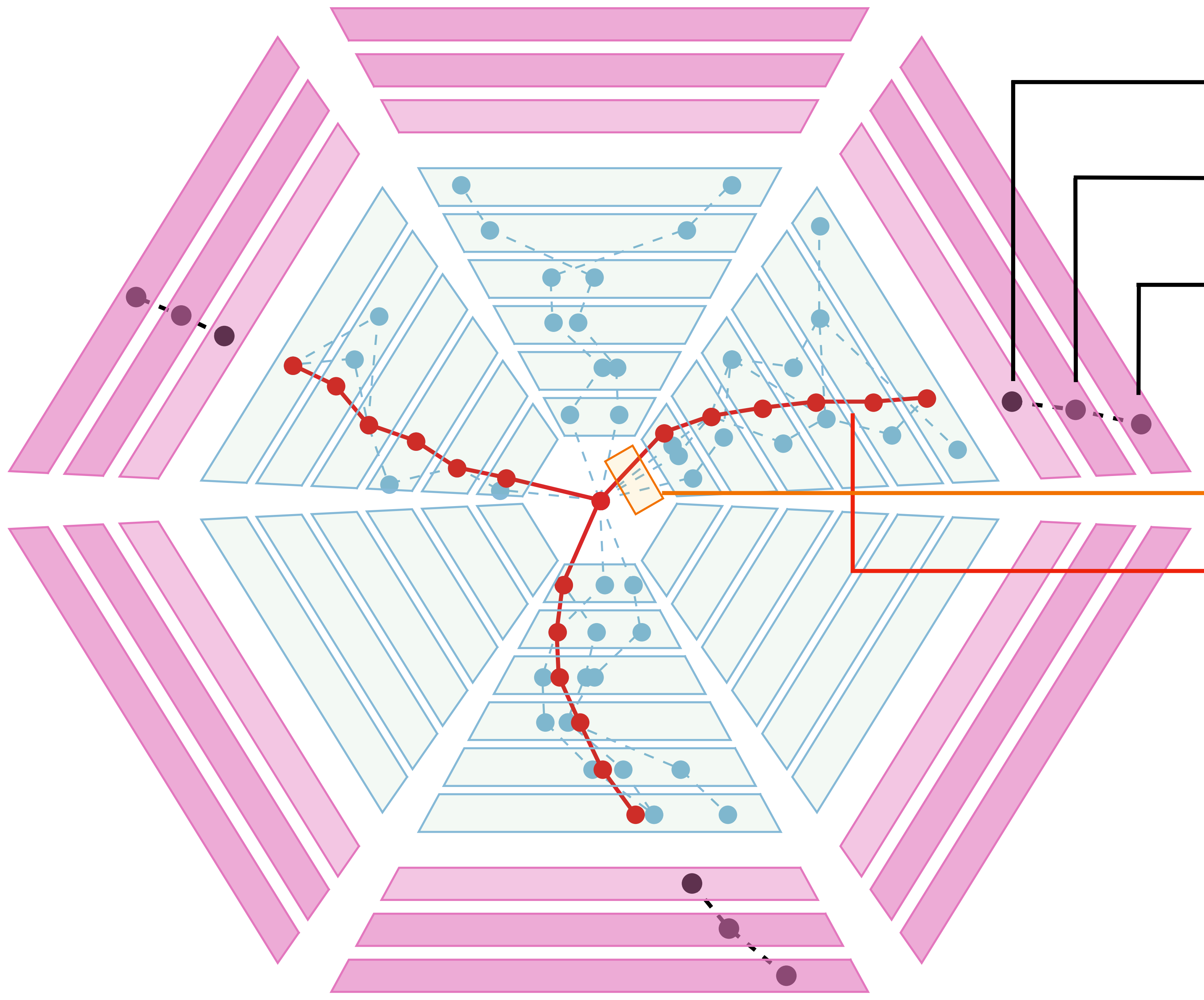
Electron Identification:

- Electrons in CLAS12 are identified using the energy deposited in ECAL combined with response from High-Threashold Cherenkov Counter (HTCC) and the track parameters.
- A Neural Network was developed to predict the track's intersection position with ECAL (all 9 layers).
- The signal from 9 layers was aggregated from the raw ECAL ADC. They used to train another neural network to classify electrons.





Electron Identification with AI



$PCAL\ [E_u, E_v, E_w]$

$ECIN\ [E_u, E_v, E_w]$

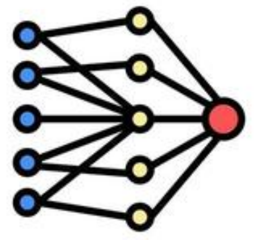
$ECOUT\ [E_u, E_v, E_w]$

$HTCC\ [A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8]$

$TRACK\ [W_1, W_2, W_3, W_4, W_5, W_6]$

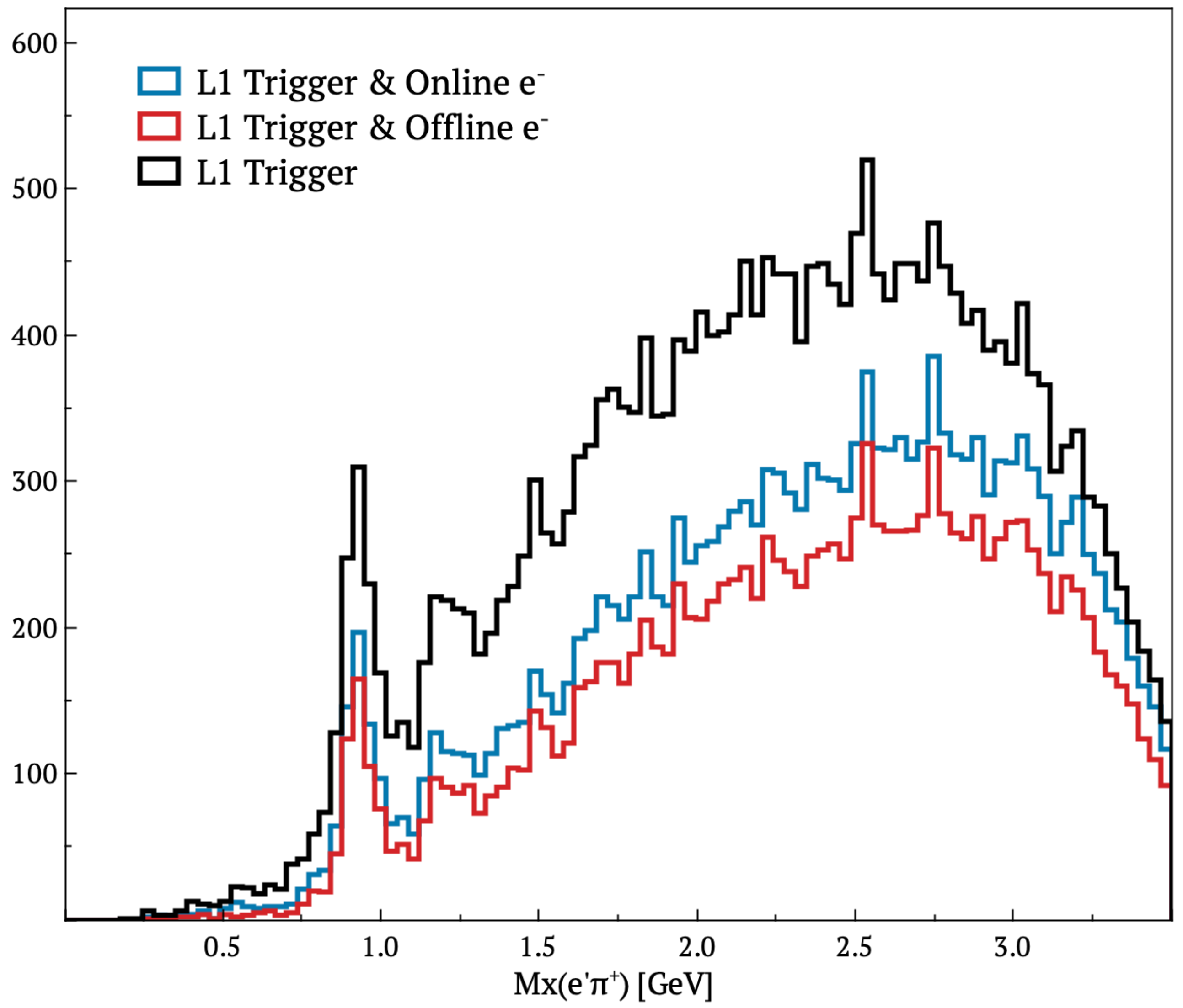
- Electrons in CLAS12 are identified using the energy deposited in ECAL combined with response from High-Threashold Cherenkov Counter (HTCC) and the track parameters.
- A Neural Network was developed to predict the track's intersection position with ECAL (all 9 layers).

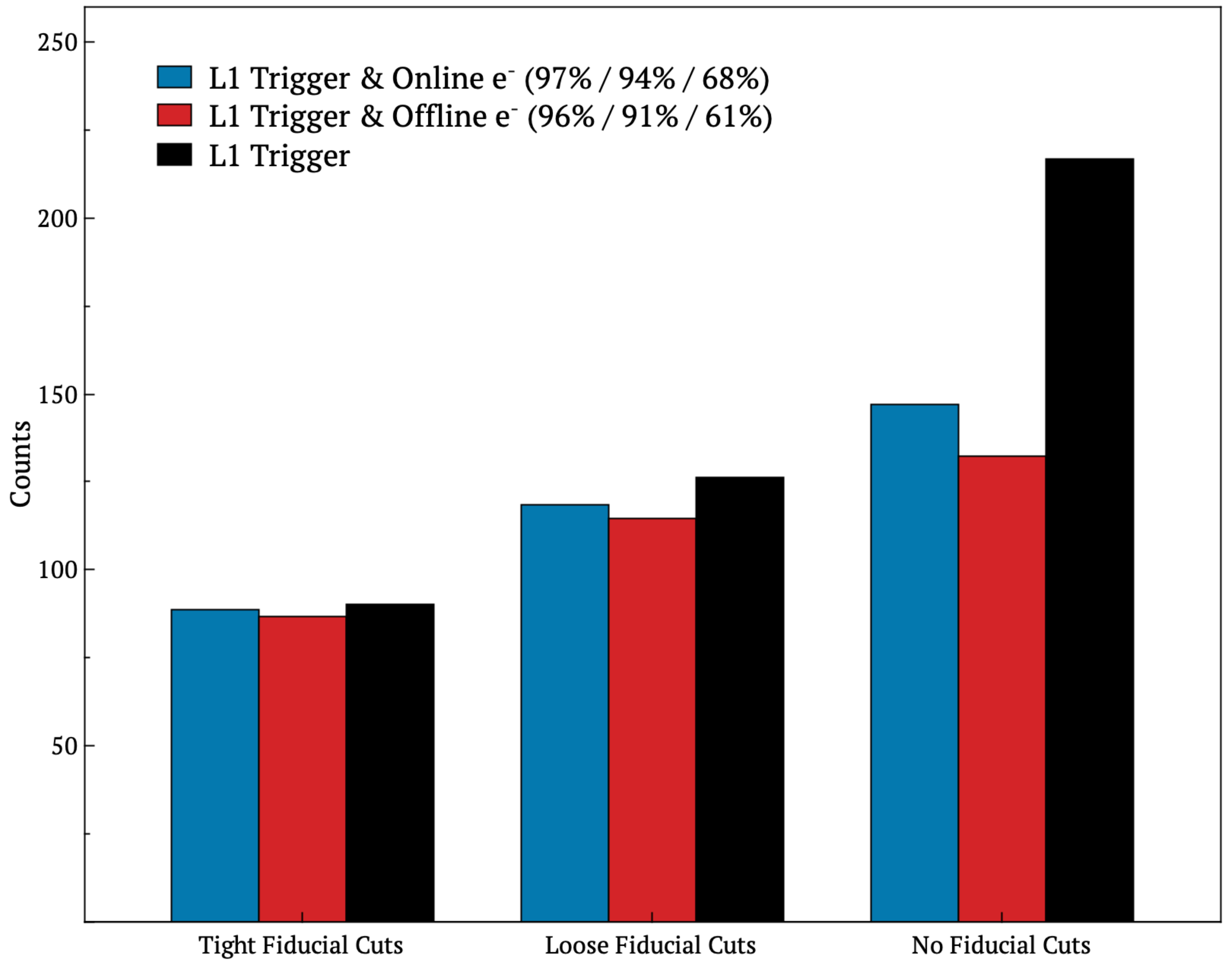
$INPUT\ [TRACK(6), HTCC(8), ECAL(9)]$



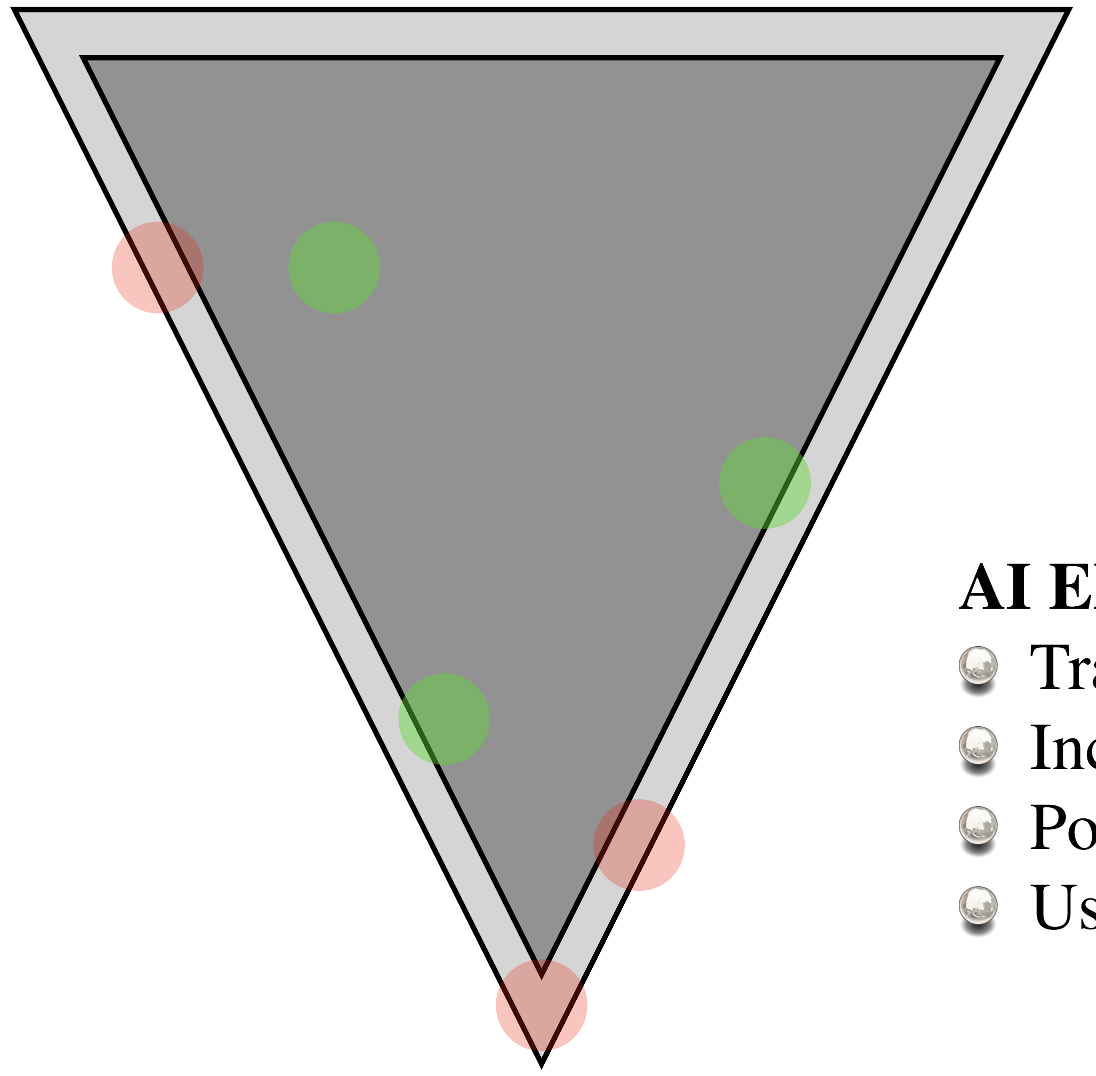
$OUTPUT\ [ELECTRON, NOTANELECTRON]$

L1 Trigger & Online e$^-$ (97% / 94% / 68%)
L1 Trigger & Offline e$^-$ (96% / 91% / 61%)
L1 Trigger





Electron Identification with AI

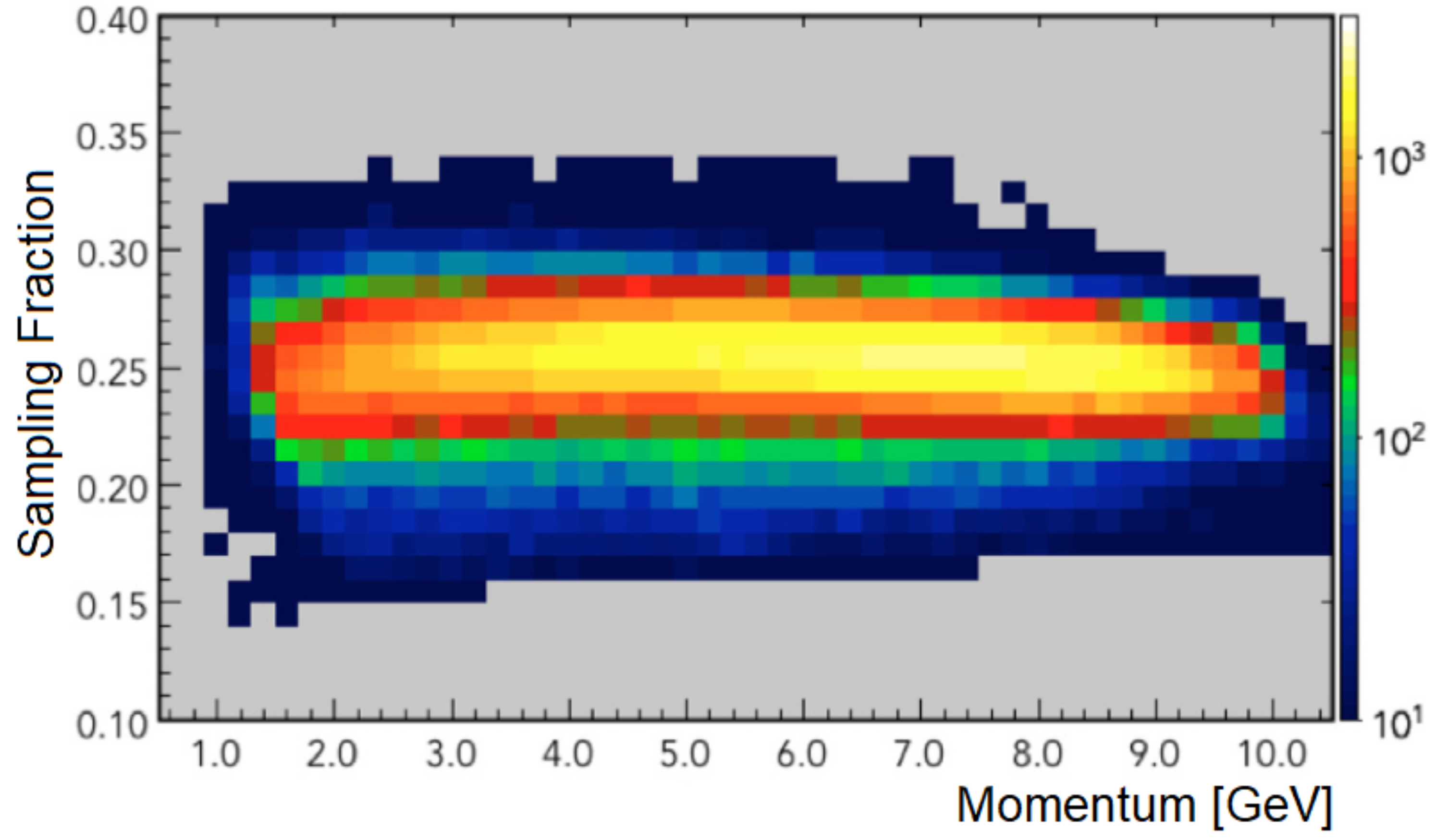


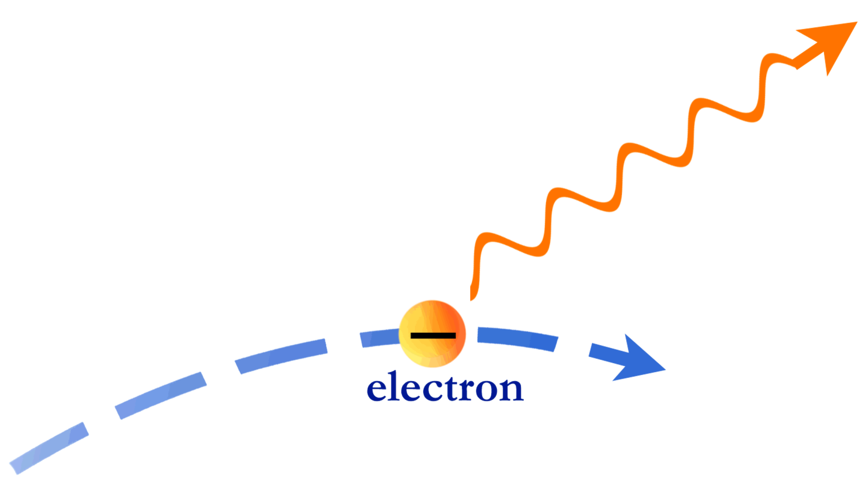
Fiducial Region: full energy deposition

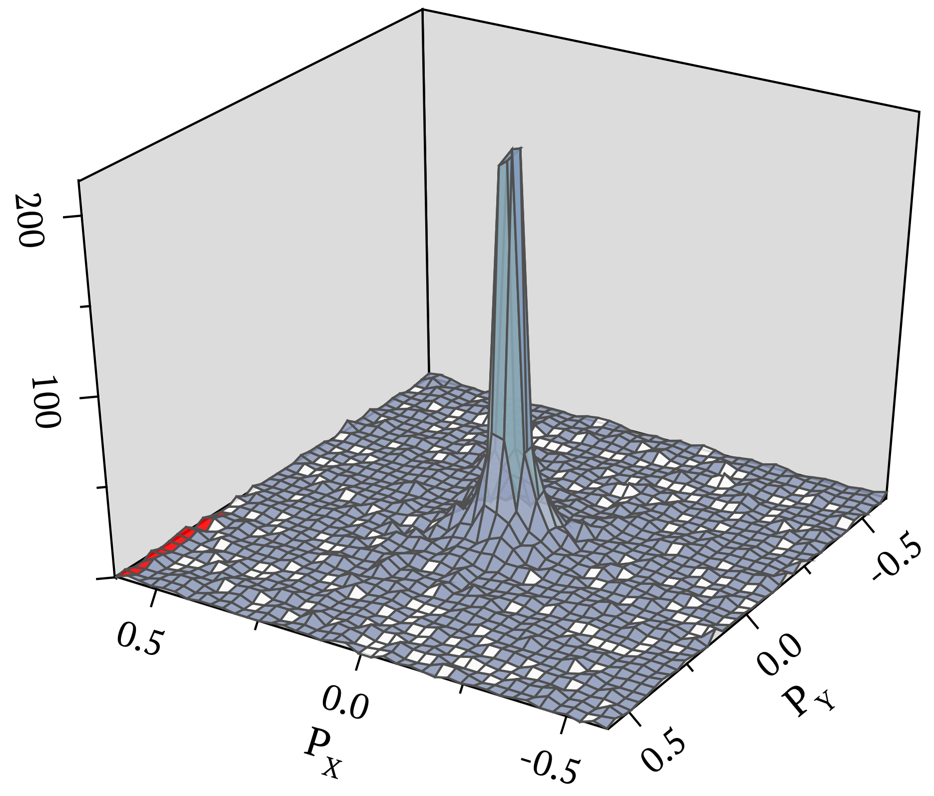
Edges: some energy escapes

AI Electron Identification Approach:
- Train a network to identify electrons in the fiducial region.
- Include edge cases in the training sample
- Possibly recover some of the electrons outside of fiducial region
- Use radiative electrons for AI training

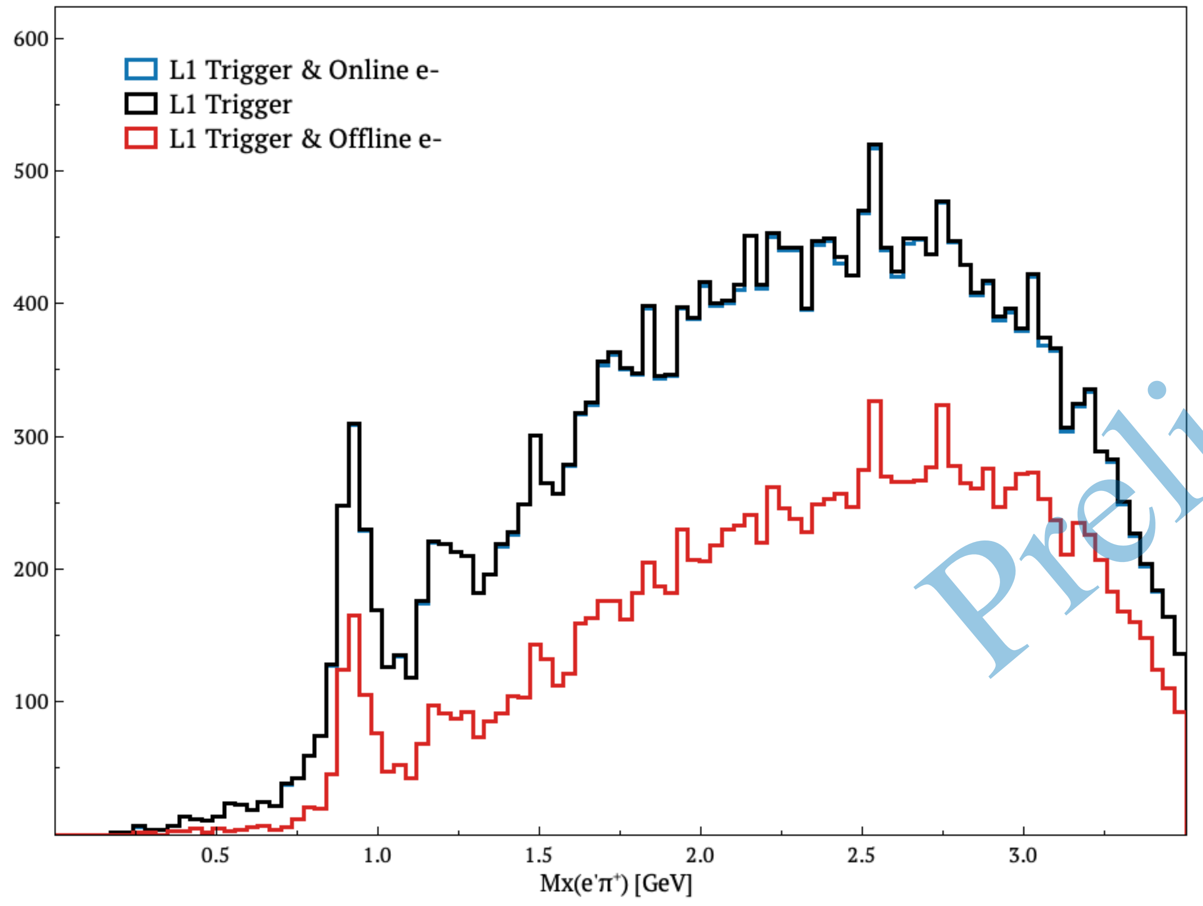
Offline Electron Identification:
A cut-based identification algorithm requires the sampling fraction (E/P) for electrons to be >0.21

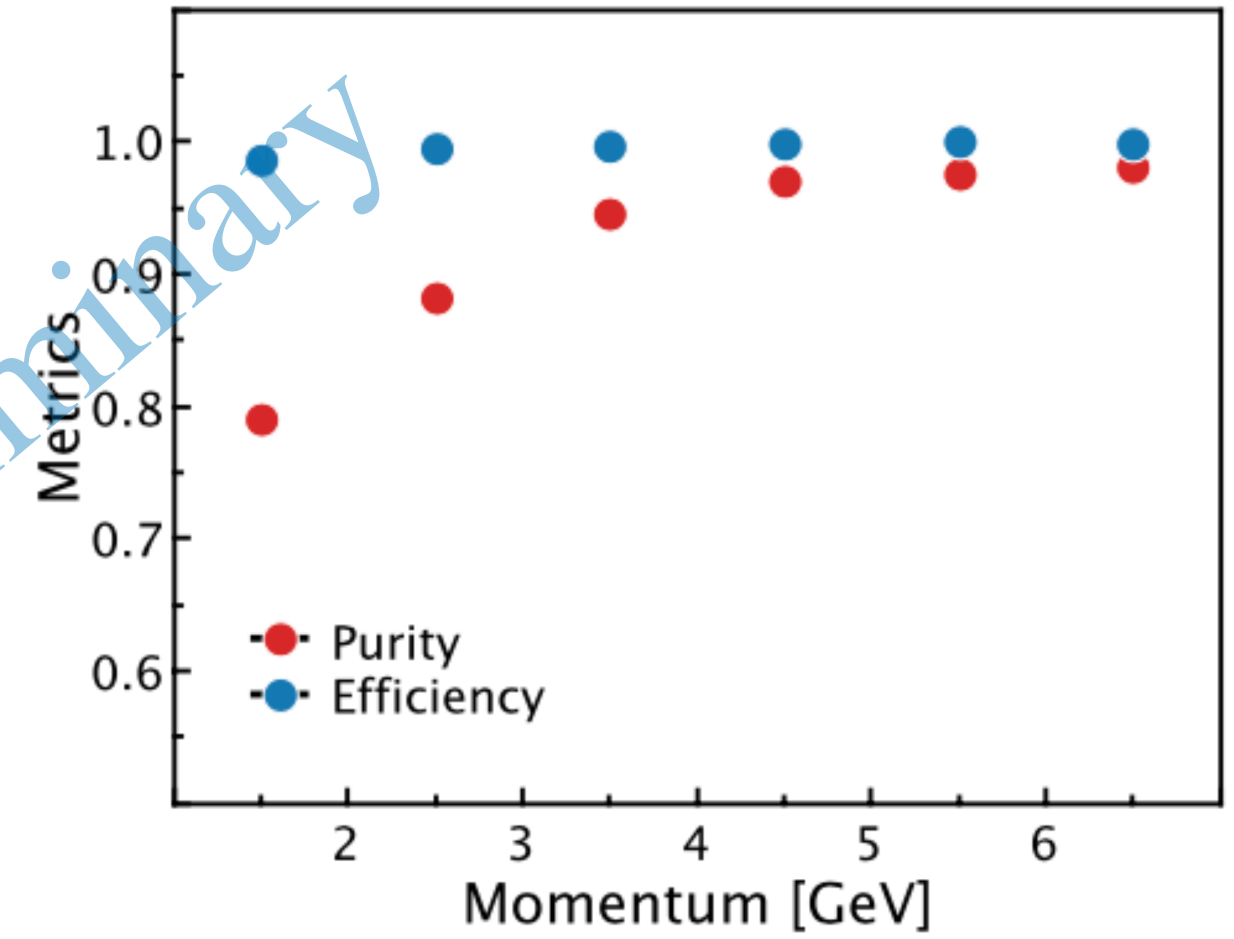






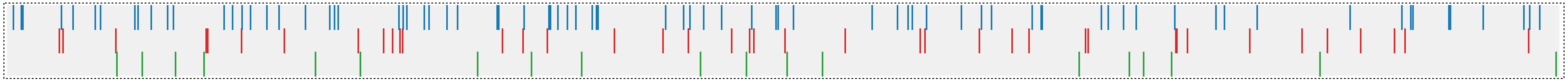
Photon Direction relative to electron



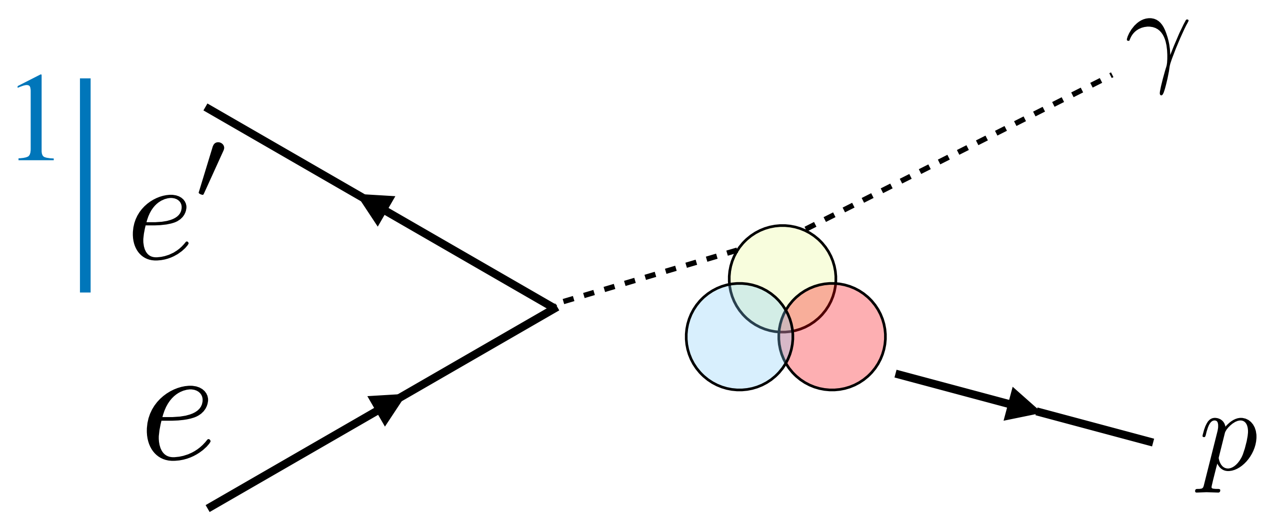


AI Revolution

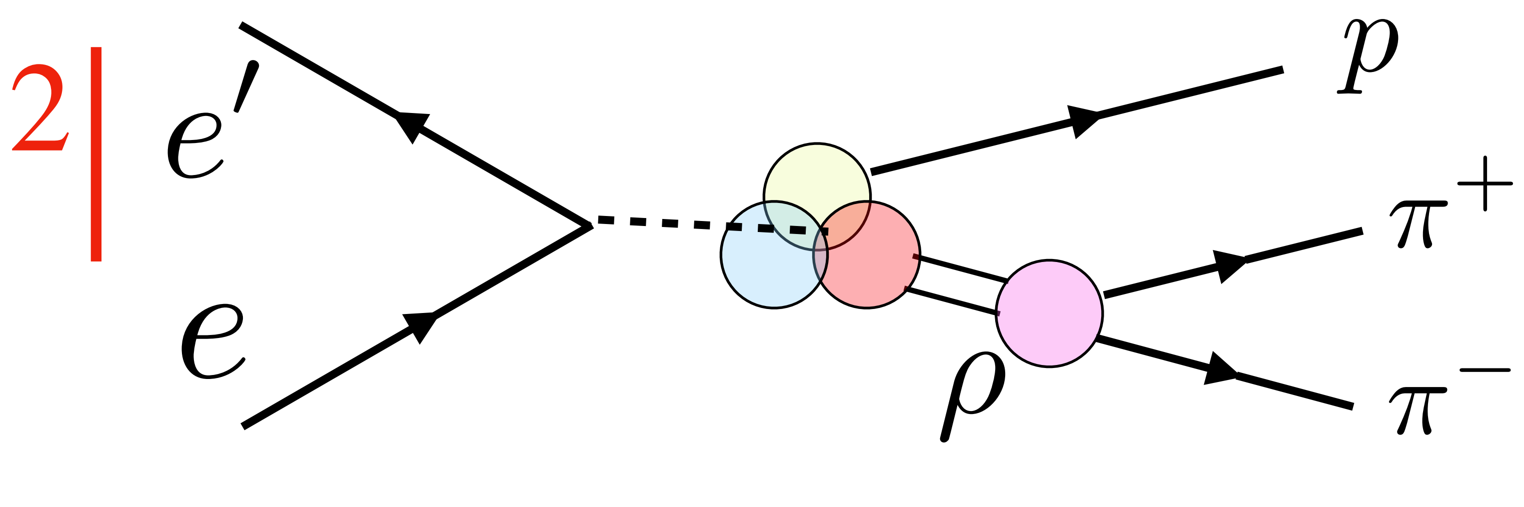
100,000,000 interactions, 4 hours



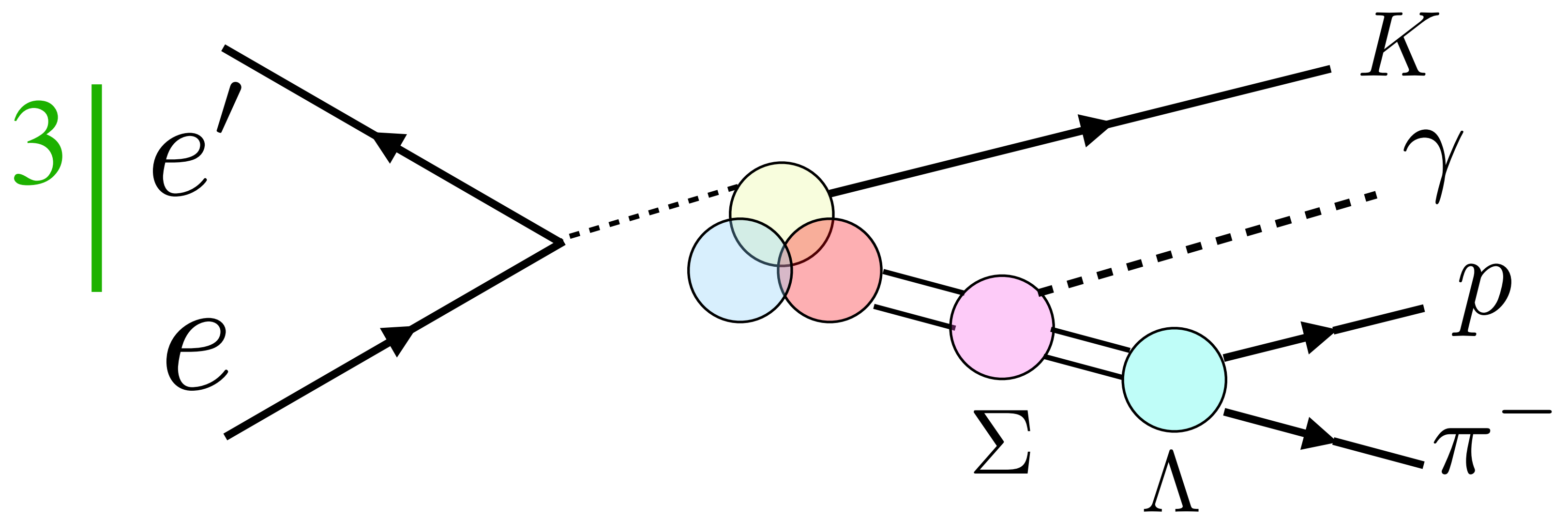
Frequency of specific interaction of electron with nuclei (hypotetical)



DVCS Probing The structure of Proton



Production of Mesons



Production of Baryons

Physics Reactions:

Electron-proton collisions yield outcomes at varying rates. Each analysis group focuses on specific event topologies, and the entire dataset must be processed with conventional methods to isolate the reactions of interest.

10,000 interactions per second
Event Topology Sorting
Processing: 2,000 CPUs

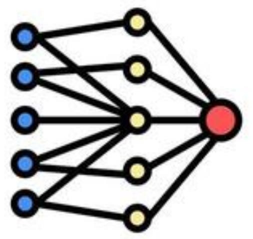
Typical experiment runs for ~2 month (50% efficiency). Collect ~180 runs (4 hour sessions). Takes 2 months to process data, making it available for physics analysis. Requires large data processing facilities.

Artificial Intelligence Approach:

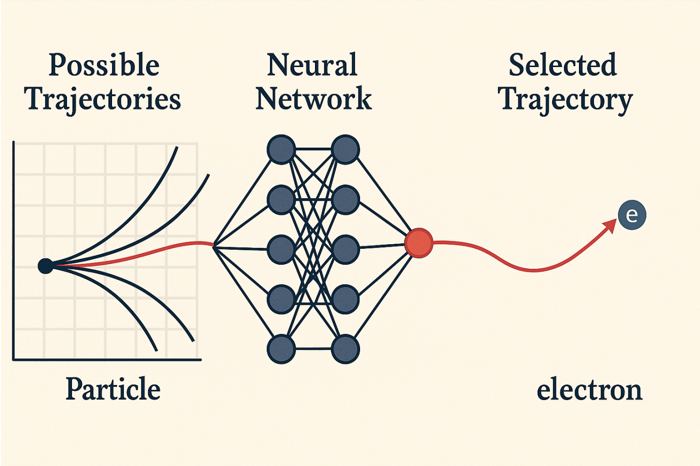
Fast AI reconstruction identifies tracks and isolates physics reactions directly from the raw data stream. This allows the interactions of interest to be processed using conventional methods much more quickly, providing scientists with the data they need for analysis.

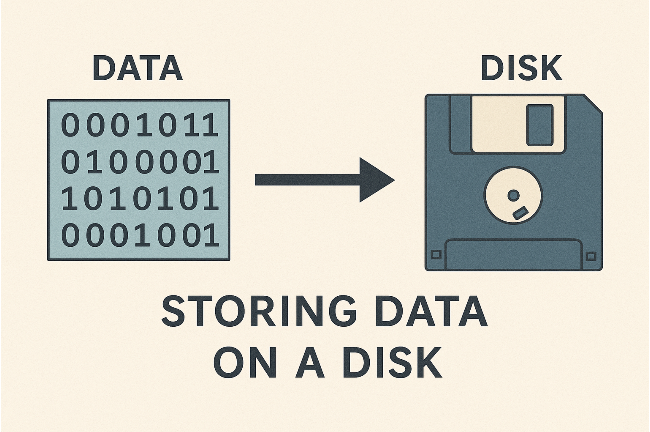
10,000 interactions per second
Event Topology Sorting
AI on Laptop: 8 CPUs

AI Data Processing in Real-Time

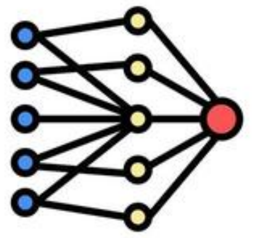


Electron Identification
Idnetifies electron tracks by combining raw data from several detectors (HTCC,DC,ECAL)

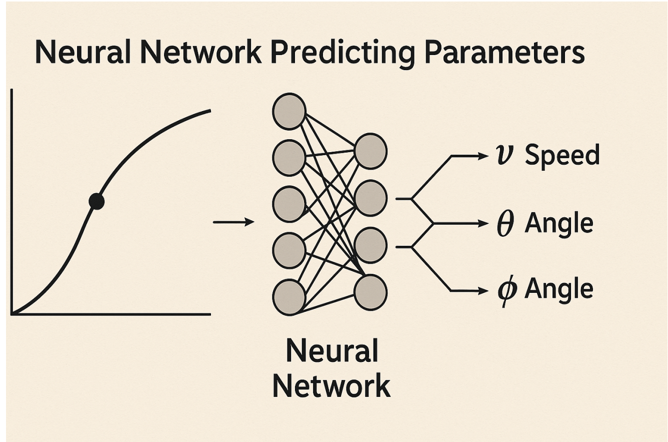


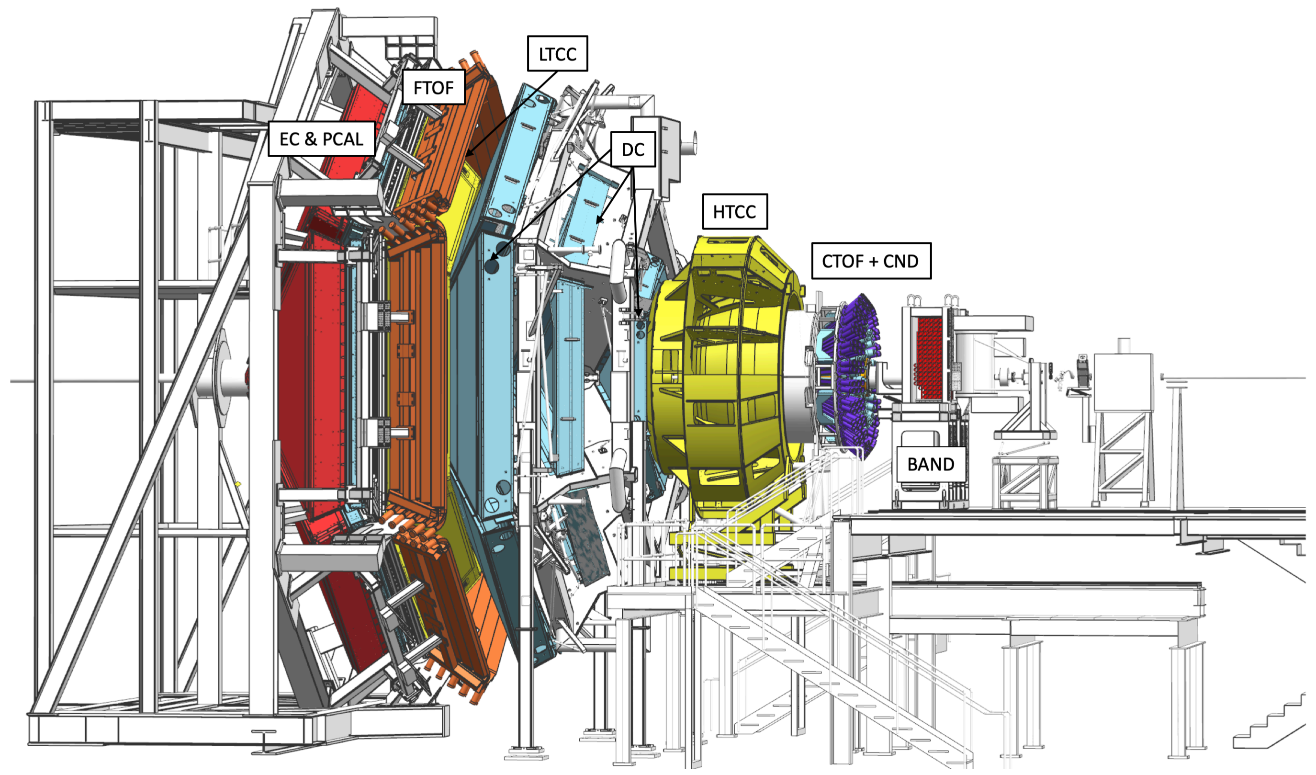


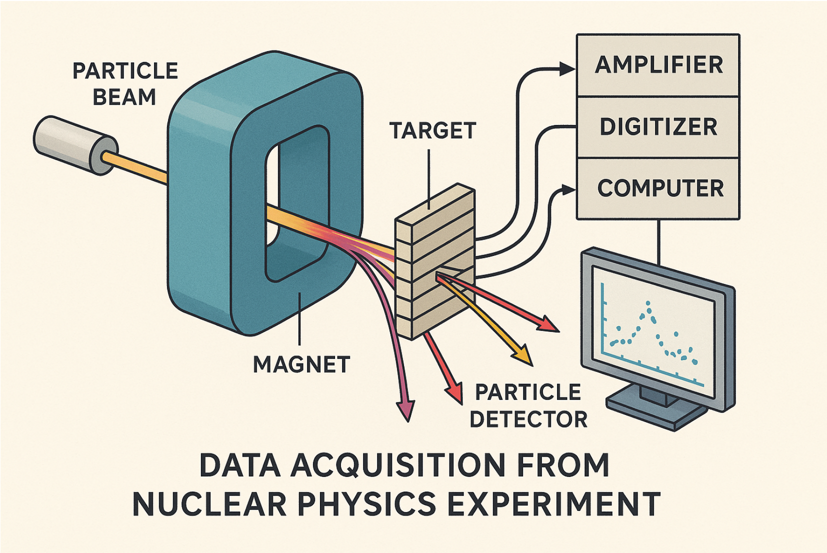
Saving experimental data
Already containing tracks
And physics topologies
Identified by AI

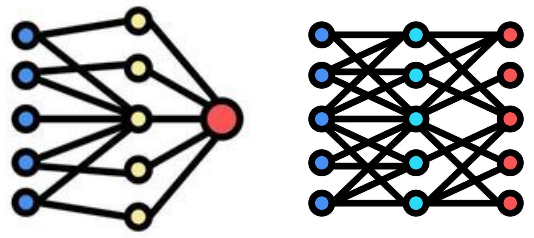


Track Parameter Predictor
A Neural Network to predict charged track parameters, momentum and direction

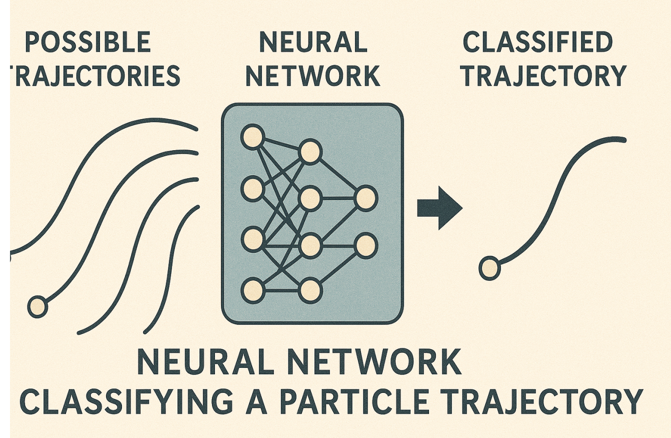






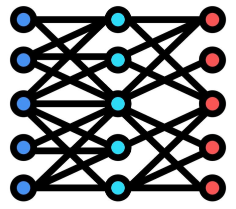


MLP Track Classifier
Identifies the best track candidates Constructed by combining segments in each of the 6 superlayers

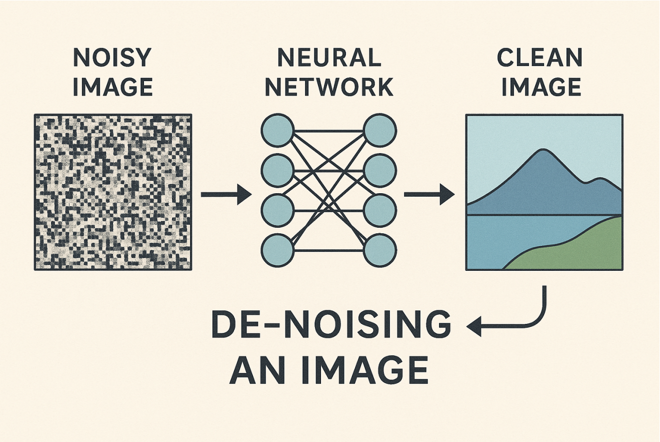


Streaming Readout:

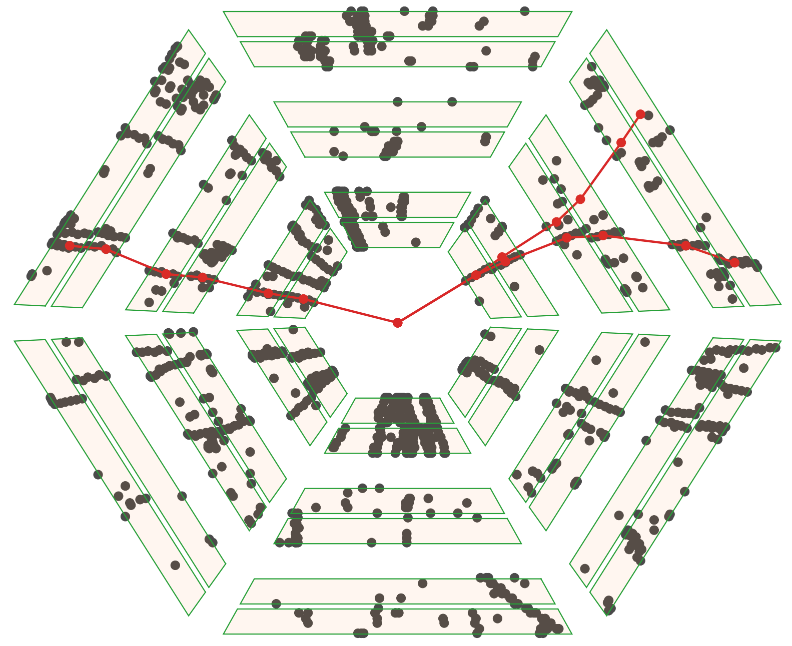
With the transition to streaming readout, AI tools will play a crucial role in data processing. Achieving comparable efficiency with conventional approaches would be difficult, especially while handling large data volumes.

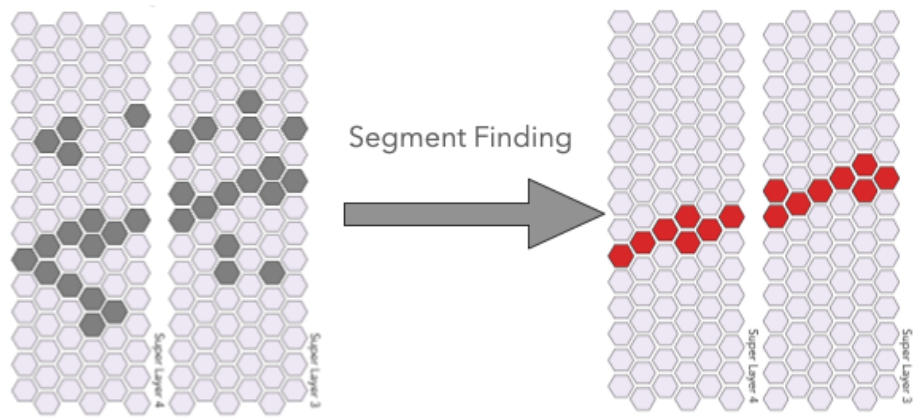


De-Noising Auto Encoder
Removes background hits
Works ~350 Hz per CPU core

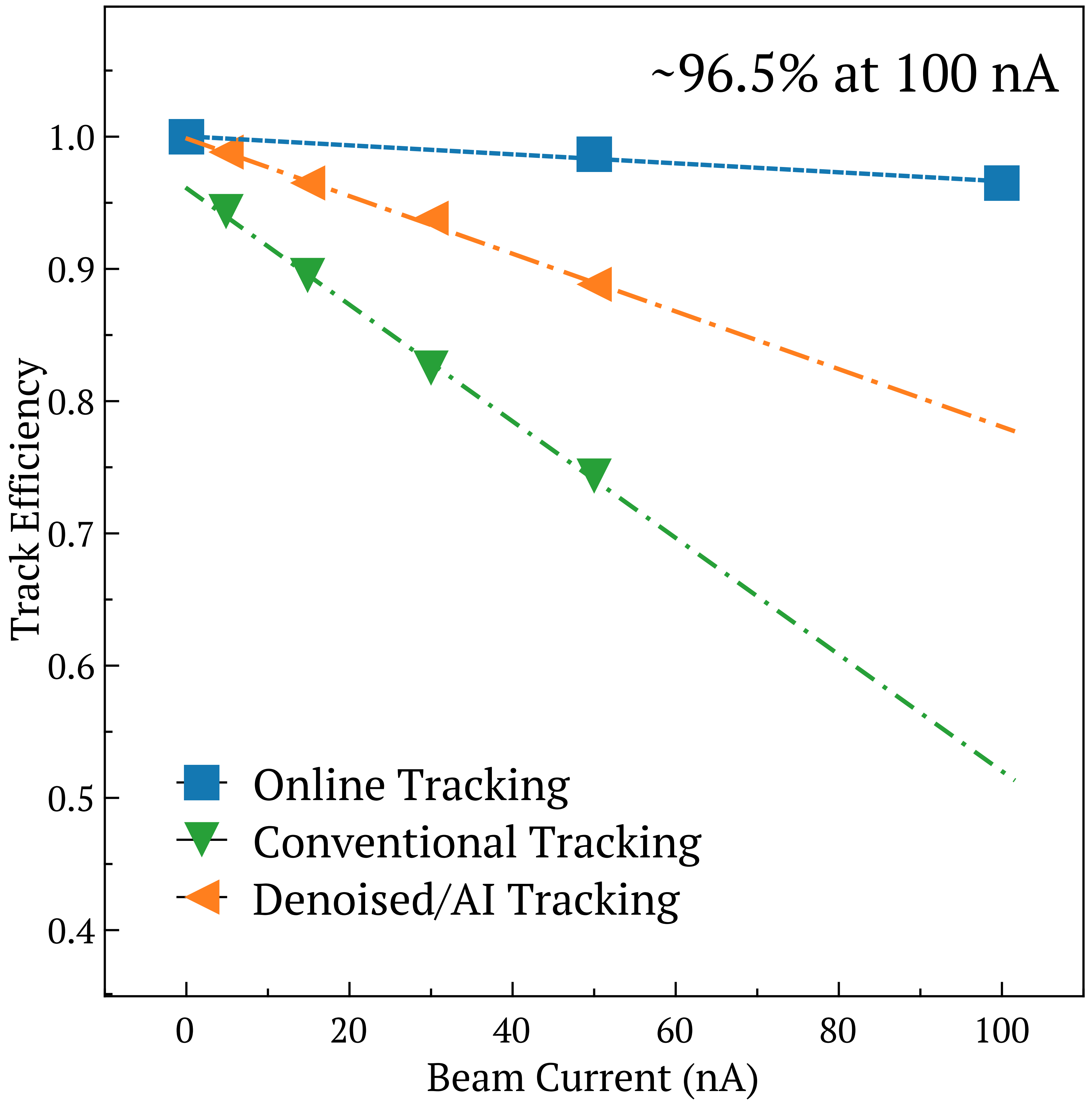


AI Data Processing in Real-Time





- Bit-wise operations to identify segments in drift chambers are implemented for online reconstruction
- Capable of working at DAQ speeds ~3.5 kHz per CPU core



Online Track Identification:

- The Online track identification relies on a faster and more efficient segment finder
- Track identification efficiency from the raw DC signals is higher compared to the offline tracking package.
- Ideal for using in real time to identify events and for triggering in streaming readout mode.
- It will be incorporated into offline algorithms to further enhance track finding efficiency.

Conclusions

Physics Event Reconstruction in CLAS12:

- The AI-assisted workflow (De-Noising CNN, Classifier MLP, and Corruption Recover AE) is integrated with standard reconstruction software and results in a significant increase in statistics (~60% for 3 particle final state)
- The improved track efficiency as a function of beam current allows running future experiments at higher luminosities, which can result in significantly more statistics for the experiments.
- The track reconstruction software speed improvement is ~30%
- The online reconstruction using only AI is capable of tagging physics reactions at speeds comparable to Data Acquisition (DAQ).
- Work is ongoing on implementing AI reconstruction as a Level-3 trigger with an electron identification AI network (work in progress)
- The AI approach to reconstructing tracks and identifying particles (electrons for triggering) is crucial for transitioning to streaming readout.
- Artificial Intelligence will change how experiments are conducted and lead to shorter times from data collection to paper publications.

The Implementation:

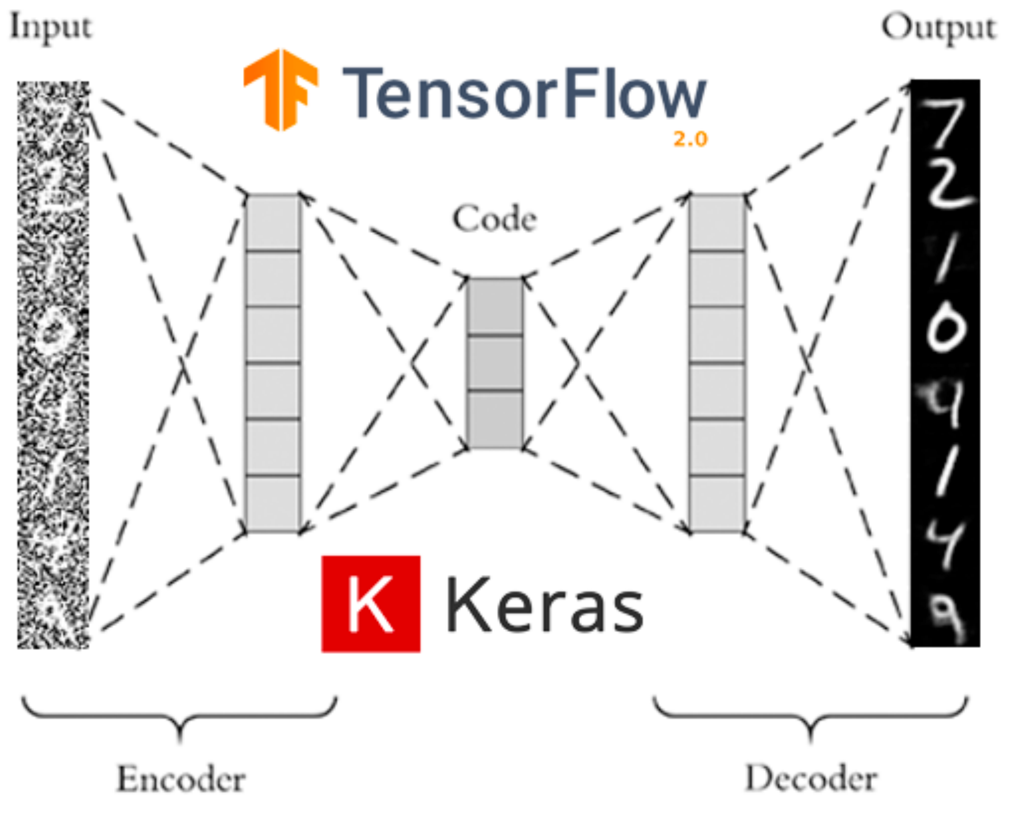
DEEP NETTS

CLAS12 reconstruction software is a Service-Oriented-Architecture (SOA), running in Java, reconstruction and monitoring software is Java-based.
The AI trackfinder is implemented in Java using DeepNetts Machine Learning library, for easy integration with reconstruction software.

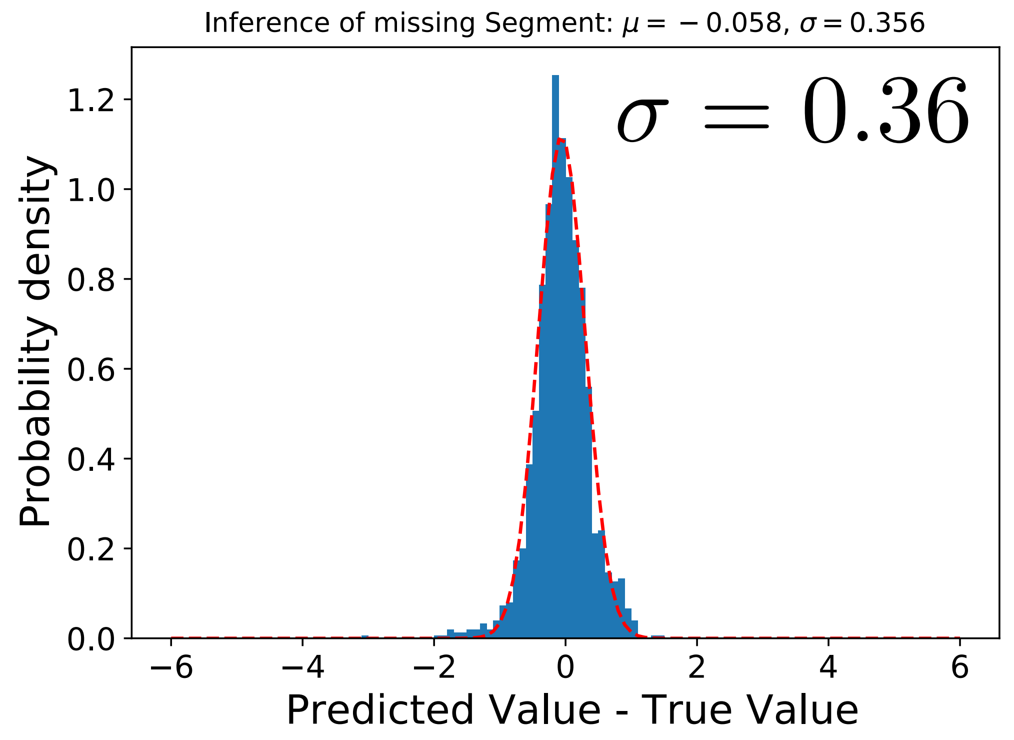
Thank you….

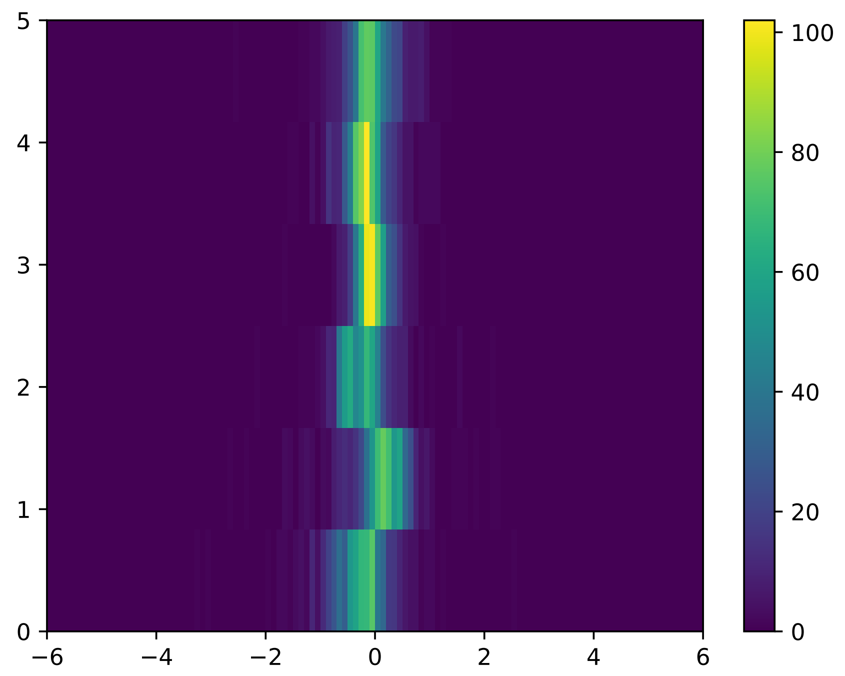
Corruption Recovery Auto-Encoder

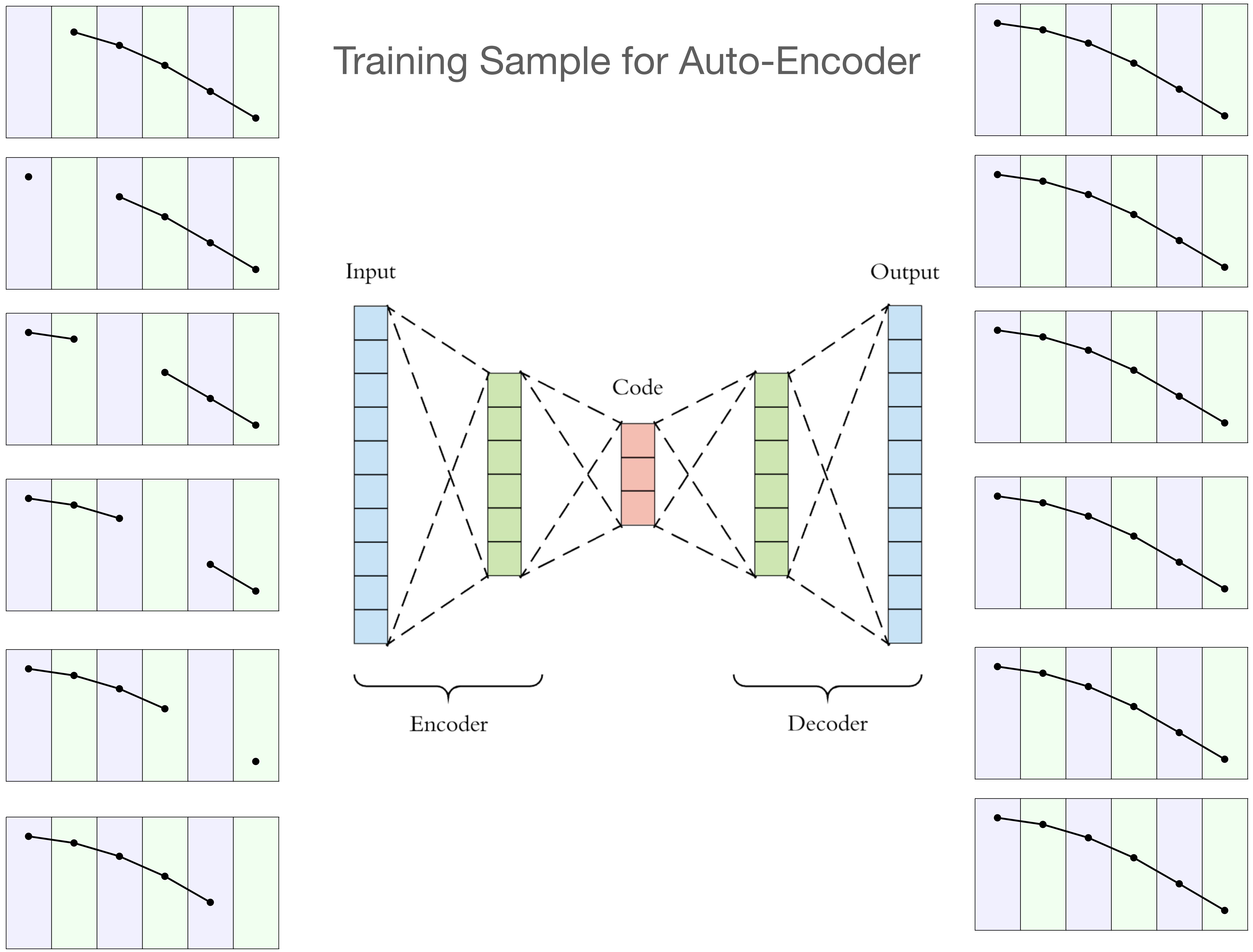
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- **Typically used for de-noising, but can be used for fixing glitches (our case).**



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





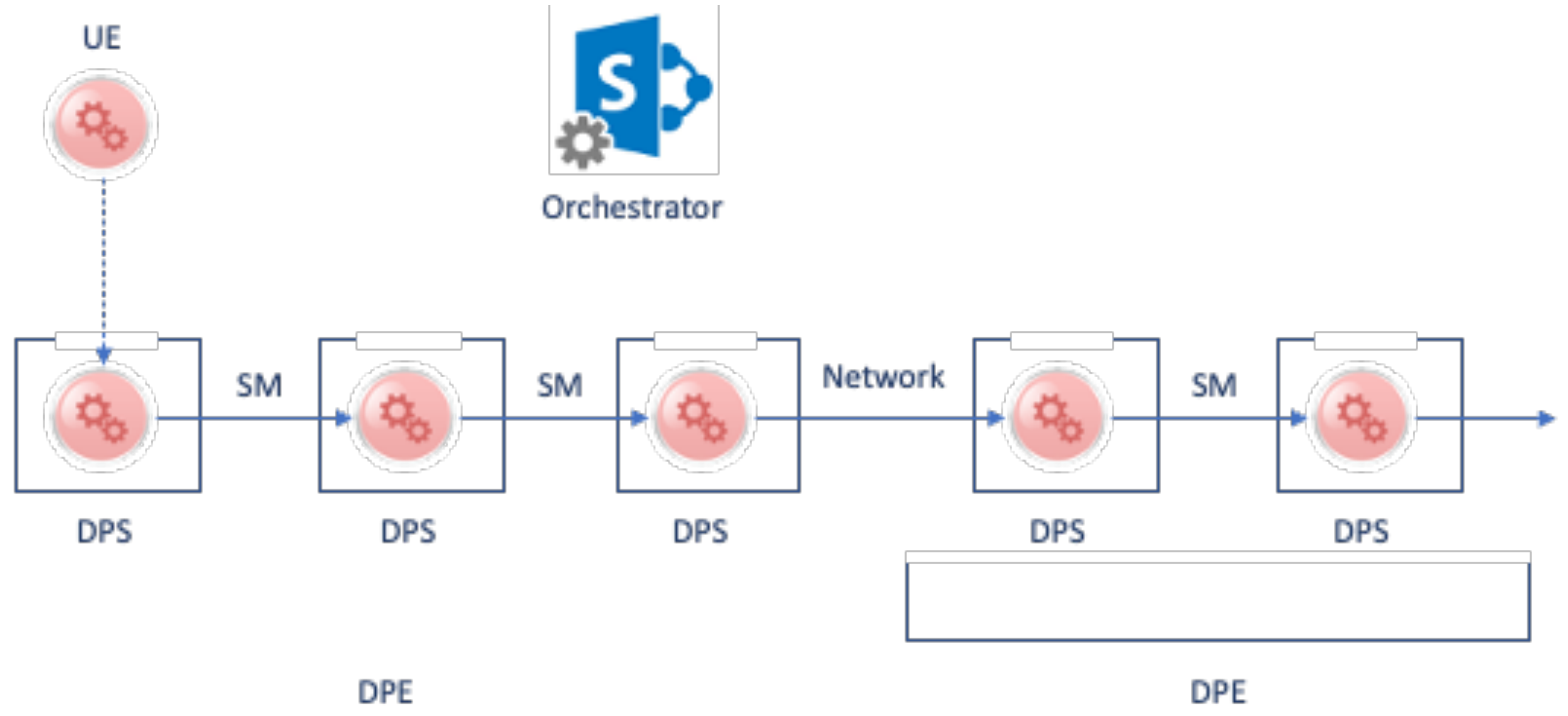


- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

Java In Nuclear Physics

- Different language bindings to present user engines in C++, C, Fortran, Java, Python..
- Reacts to:
  - Streaming-unit (message-driven and responsive)
  - Service failure (resilient)
  - Variable load conditions (elastic)
- Provides multi-threading:
  - Non blocking processing
    - Threads don't block on async execution
  - Task oriented orchestration
    - Can be altered dynamically based on running conditions
- Graphical application representation and design

- **CLARA Overview:** A reactive actor/micro-service-based framework designed for real-time processing of unbounded data streams at scale.
- **Key Features:**
  - Event-driven reactive actors
  - Networked by data pipelines for efficient data transport
  - Compositional actors with runtime-configurable conditional routing
  - Based on the flow-based programming paradigm (FBP)
- **Why CLARA Matters:**
  - Encourages application design based on software artifacts
  - Improves fault isolation
  - Easy to embrace hardware and software heterogeneity
  - Eliminates long-term commitment to a single technology stack



DPE : Data Processing Environment

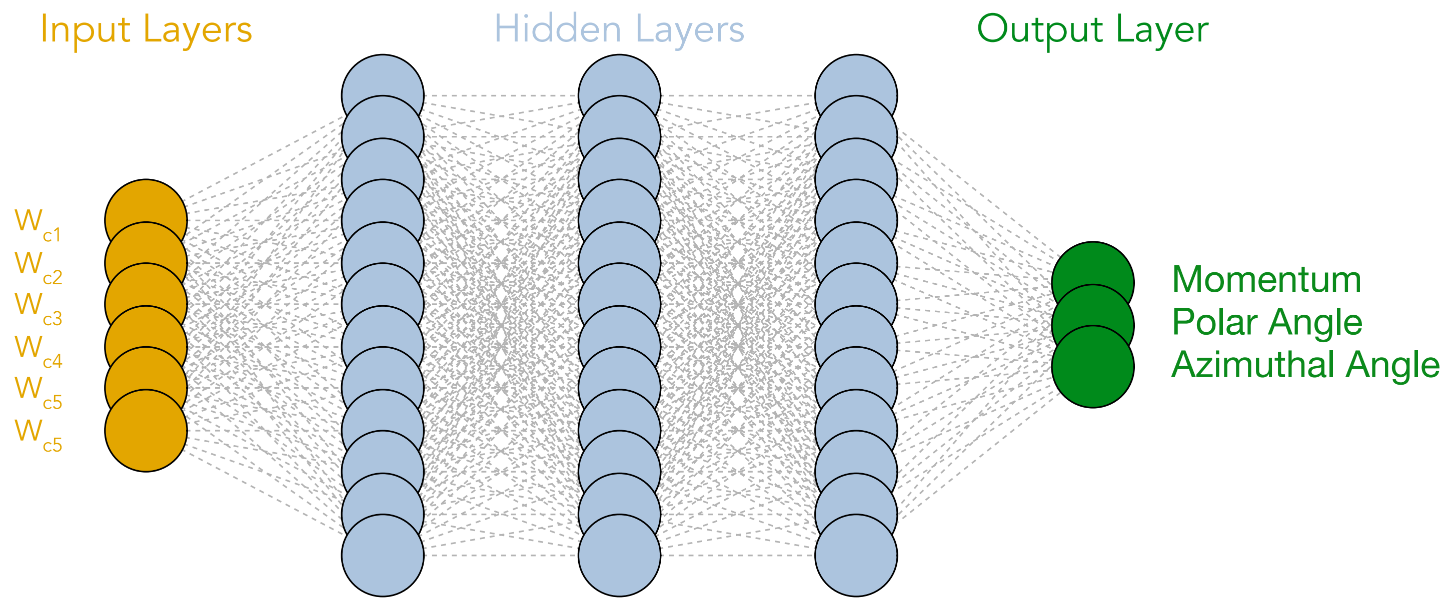
SM : Shared Memory

DPS : Data Processing Station

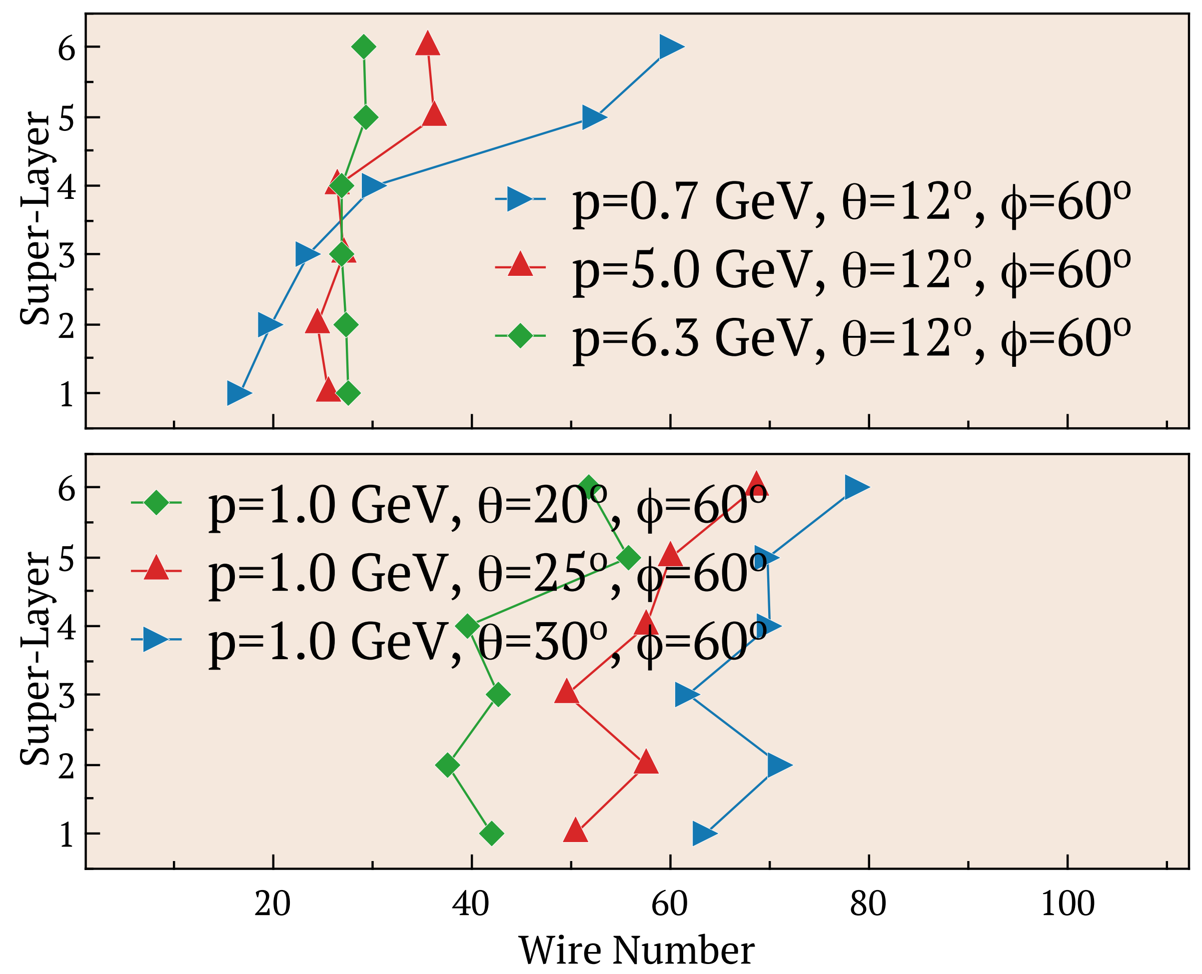
Data Processing From The Experiment:

- CLAS12 Experiment Data Processing uses CLARA (SOA based Architecture)
- Entirely written in Java
- Allows running a Modular application on heterogeneous platforms
- Deployable to Computing Data Centers

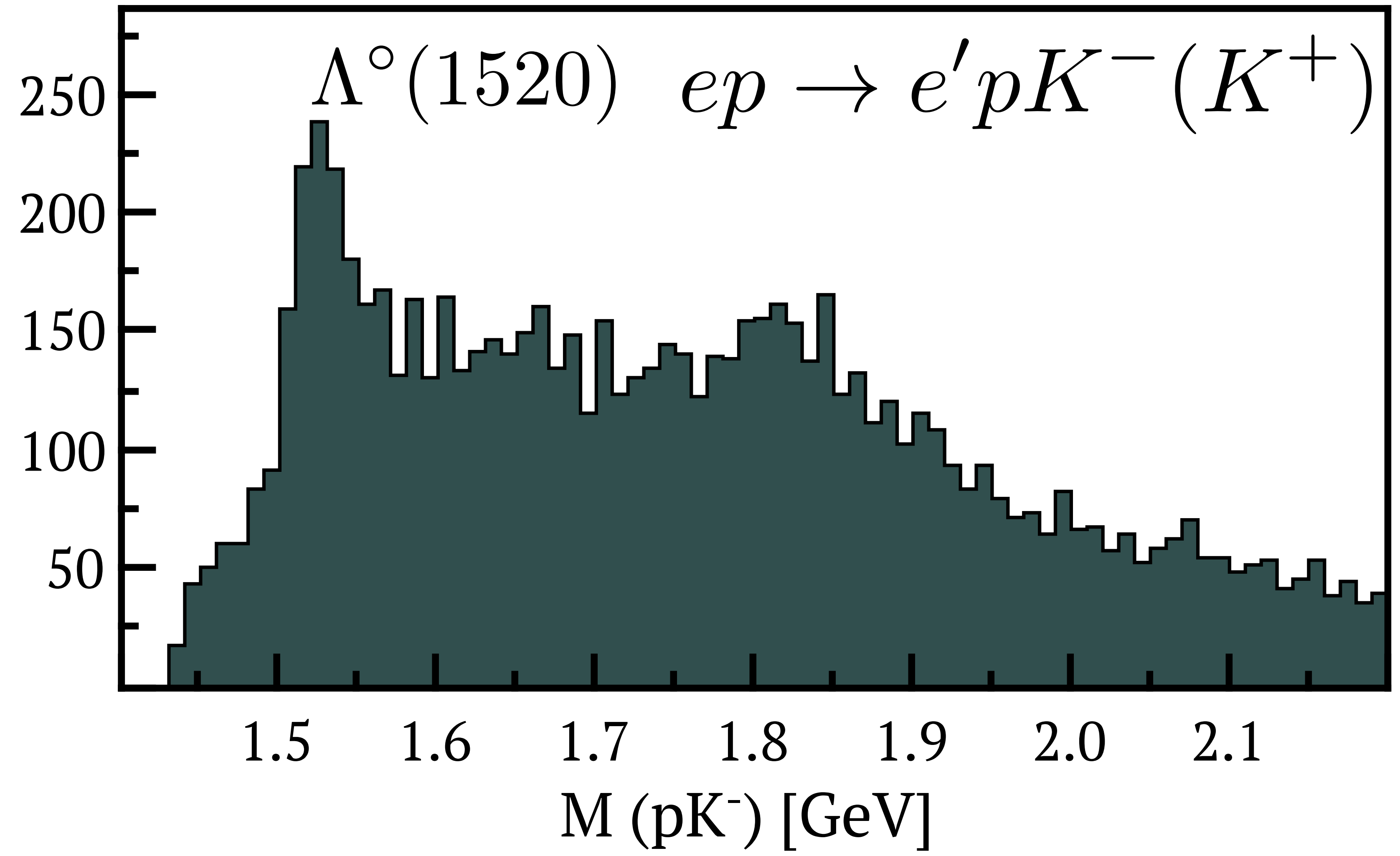
Particle Reconstruction



Regression Neural network to predict the track momentum and direction.
The track momentum is reconstructed with an accuracy of 1.4%-1.7%



Distributions calculated from track reconstruction from RAW Drift Chamber hits
Inference speed 96 kHz on a laptop (MacBook M3)

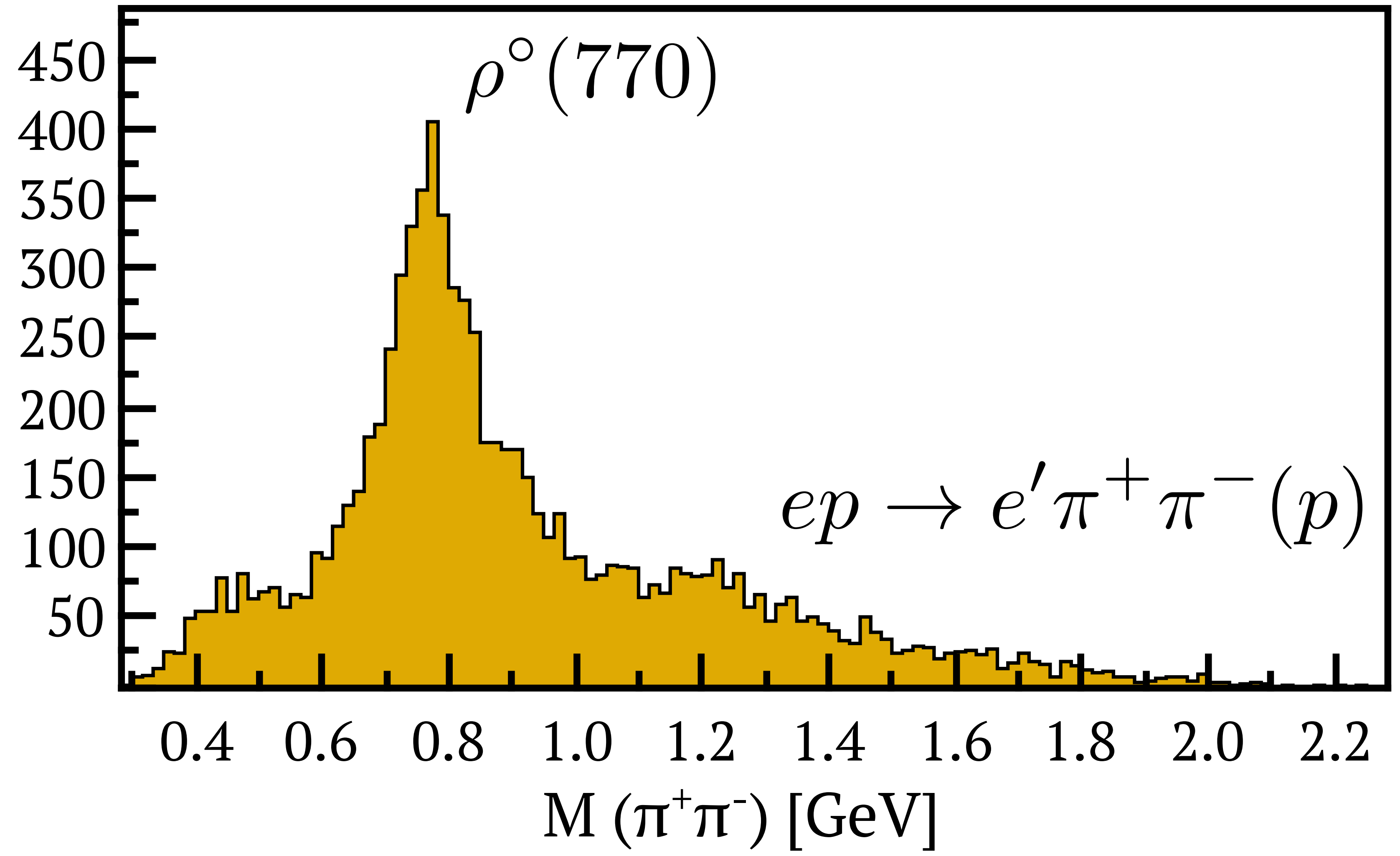


Particle Reconstruction (with AI)

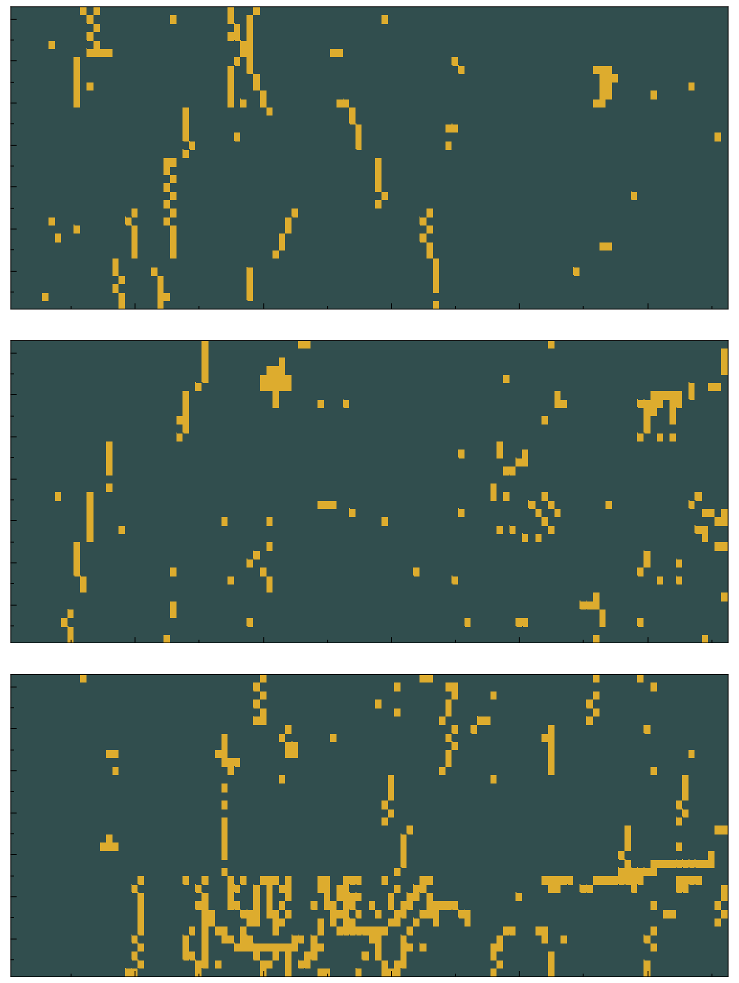
The particle final states are identified using particles reconstructed using only AI algorithms. Using Denoising, Track identification, and Parameter estimation networks

Allow full final state reconstruction.

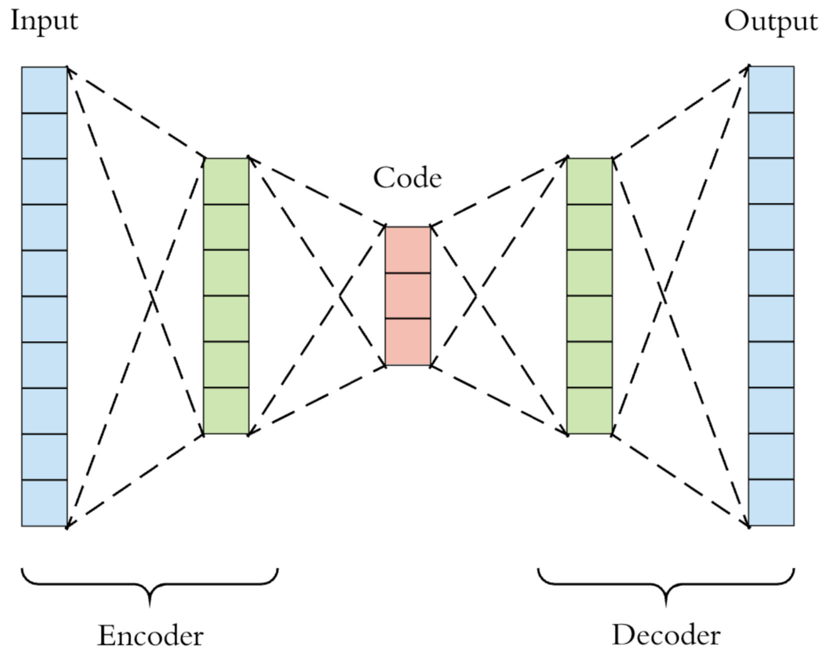
In Figures, baryon and meson states are identified when the electron beam scatters off of a proton target.

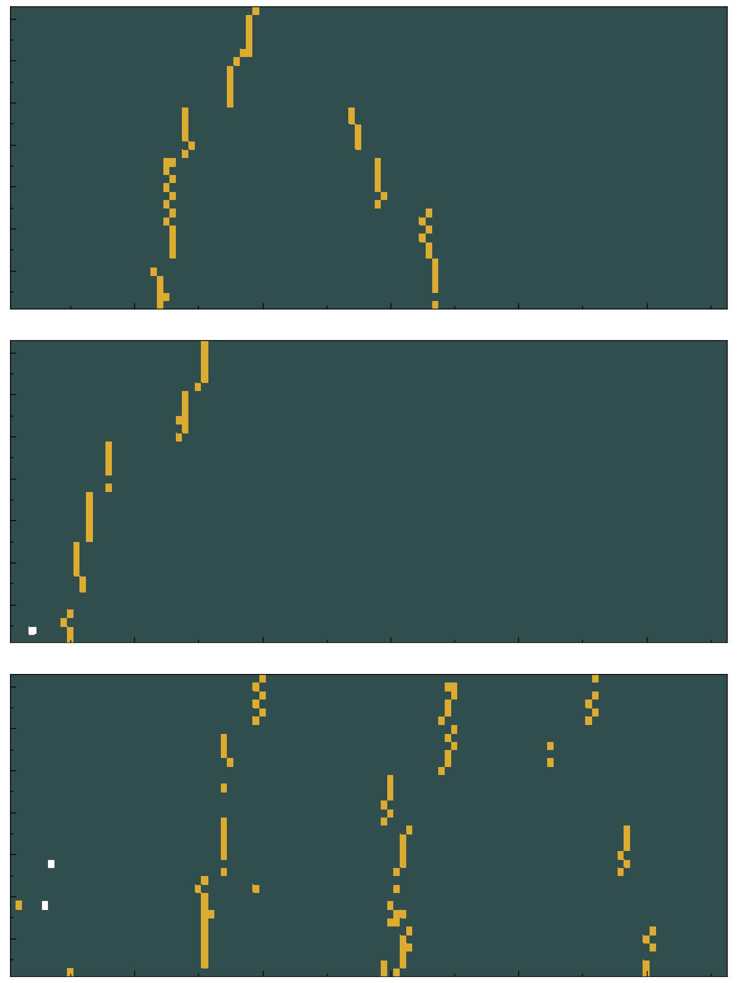


Using Artificial Intelligence in Reconstruction

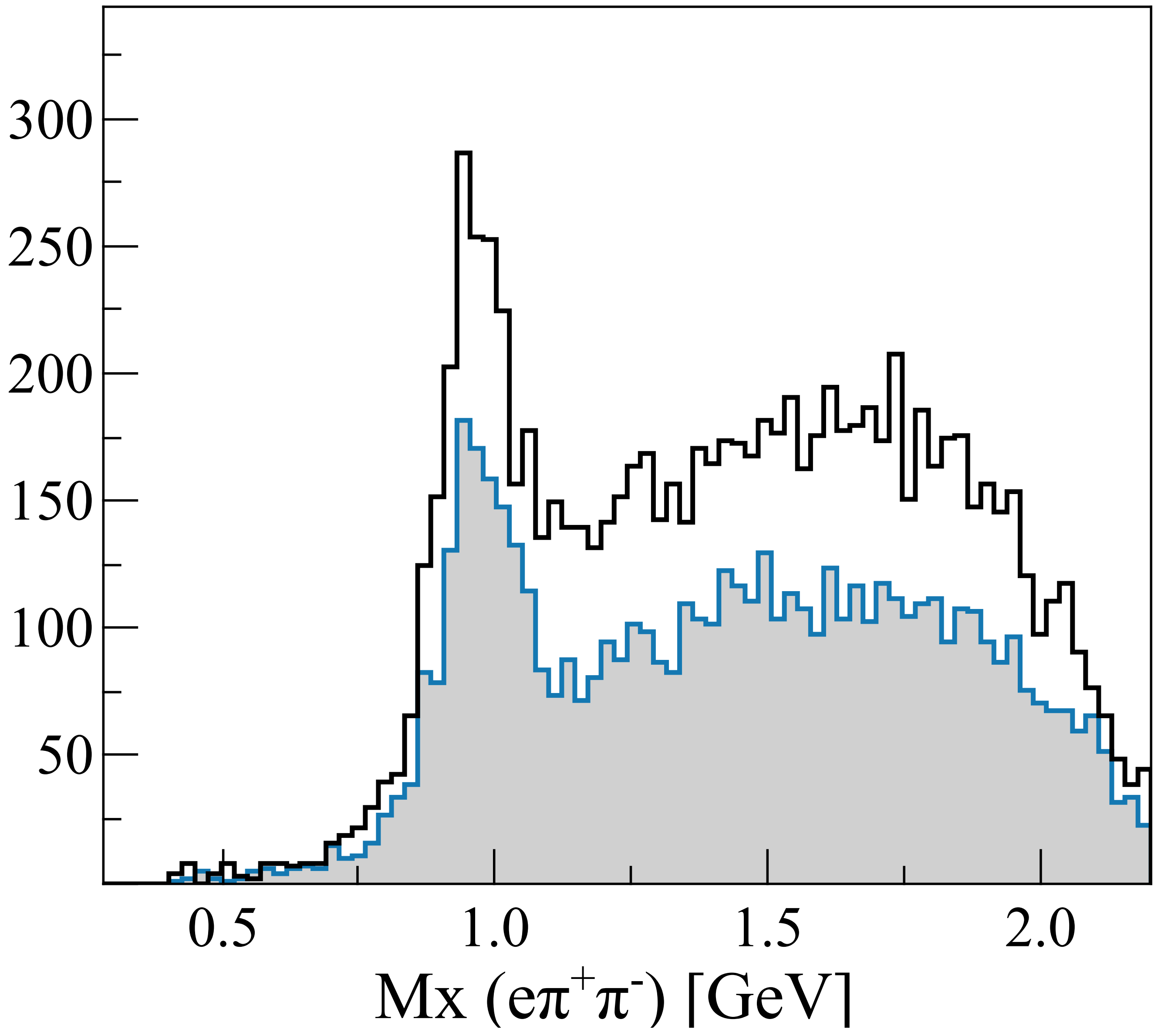


De-Noising CNN Auto-Encoder

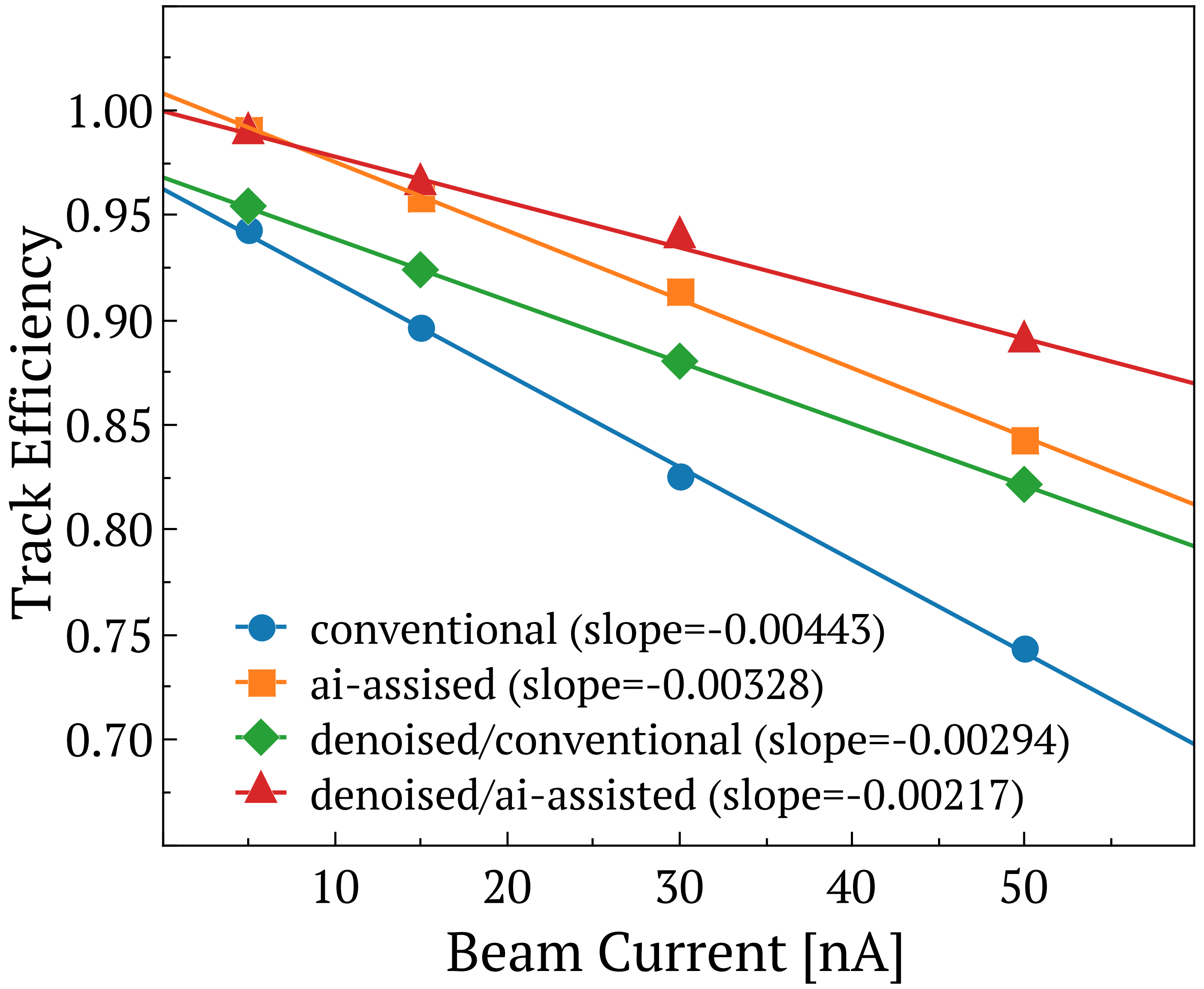


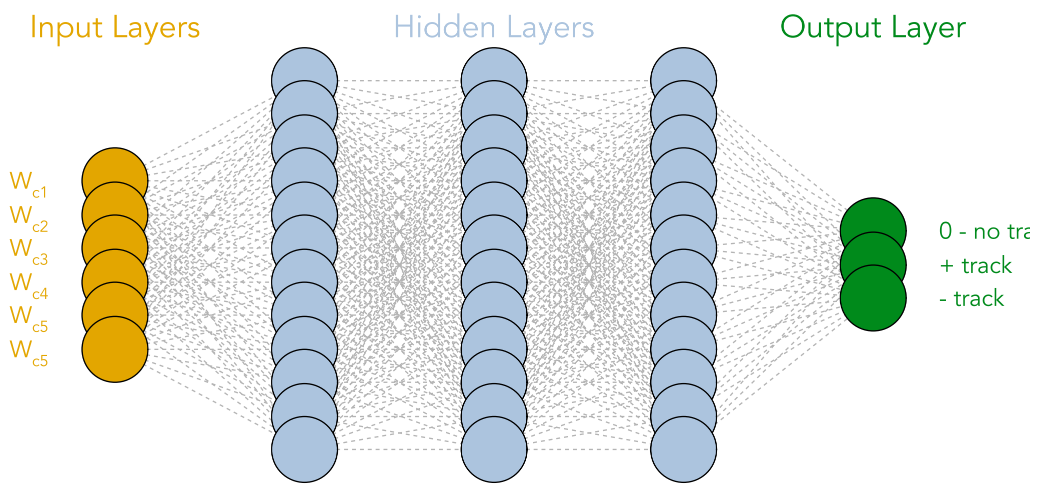


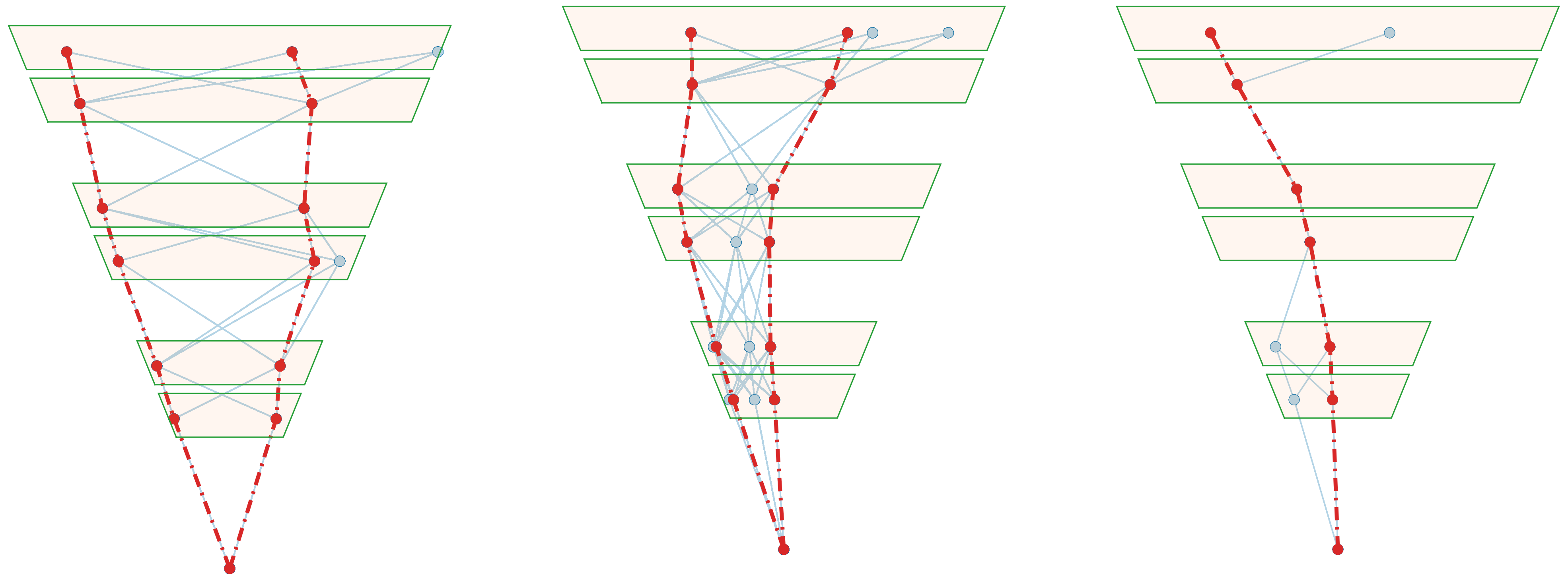
Retains >95% of hits belonging to tracks, and removes >90% of background hits



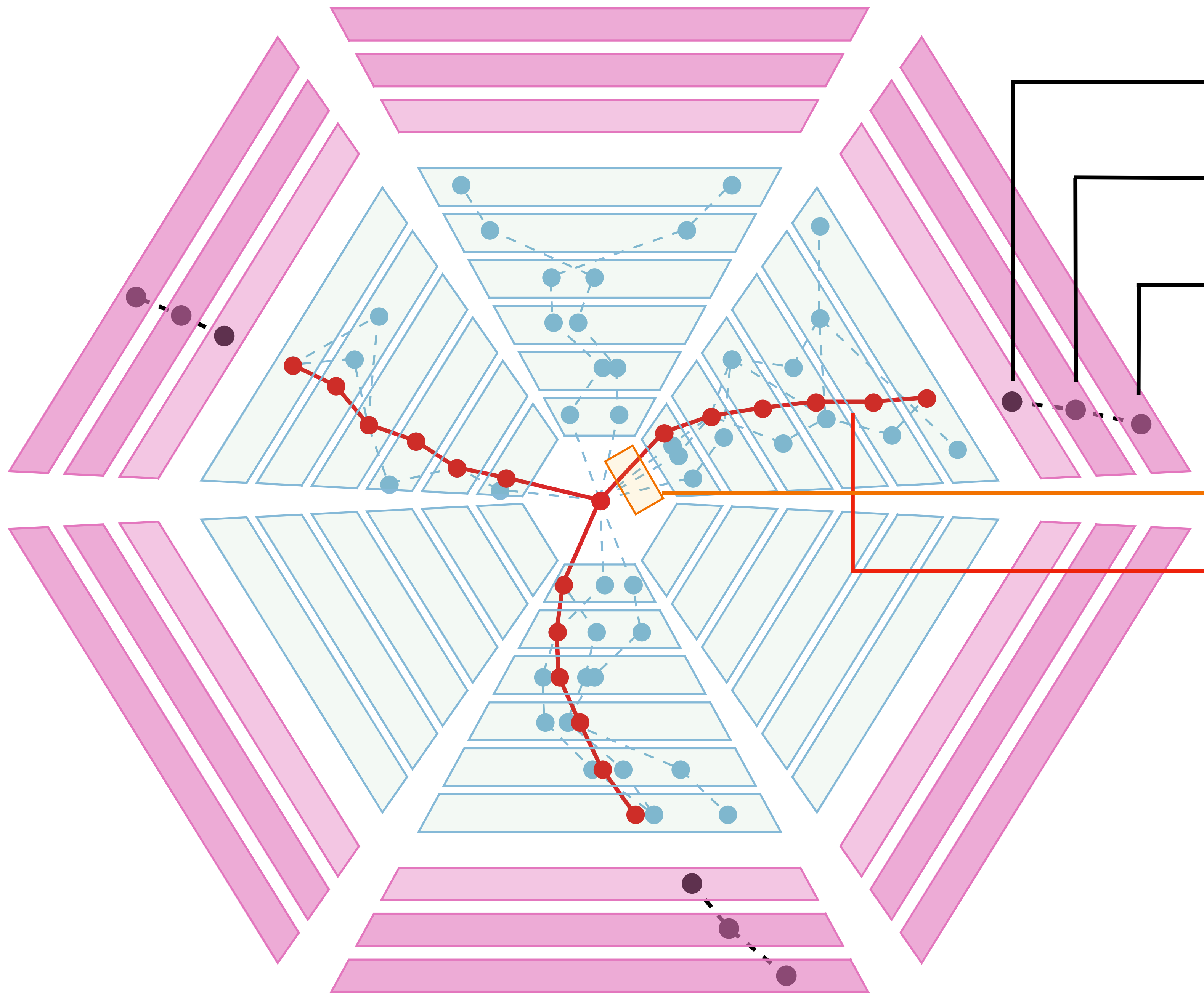
The Classifier network identifies tracks from segment combinations and identifies track charges. The AI-assisted track identification increased tracking efficiency by **15%-21%** (depending on luminosity)
Improvement of the efficiency slope as a function of luminosity.







Electron Identification with AI



$PCAL\ [E_u, E_v, E_w]$

$ECIN\ [E_u, E_v, E_w]$

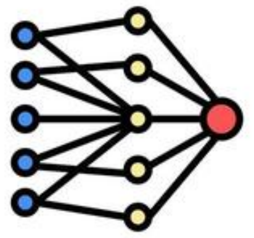
$ECOUT\ [E_u, E_v, E_w]$

$HTCC\ [A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8]$

$TRACK\ [W_1, W_2, W_3, W_4, W_5, W_6]$

- Electrons in CLAS12 are identified using the energy deposited in ECAL combined with response from High-Threashold Cherenkov Counter (HTCC) and the track parameters.
- A Neural Network was developed to predict the track's intersection position with ECAL (all 9 layers).

$INPUT\ [TRACK(6), HTCC(8), ECAL(9)]$



$OUTPUT\ [ELECTRON, NOTANELECTRON]$