# Artificial Intelligence in CLAS12

## Artificial Intelligence/Machine Learning for Physics Applications


G.Gavalian (Jefferson Lab)

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

VIA VERITAS VITA

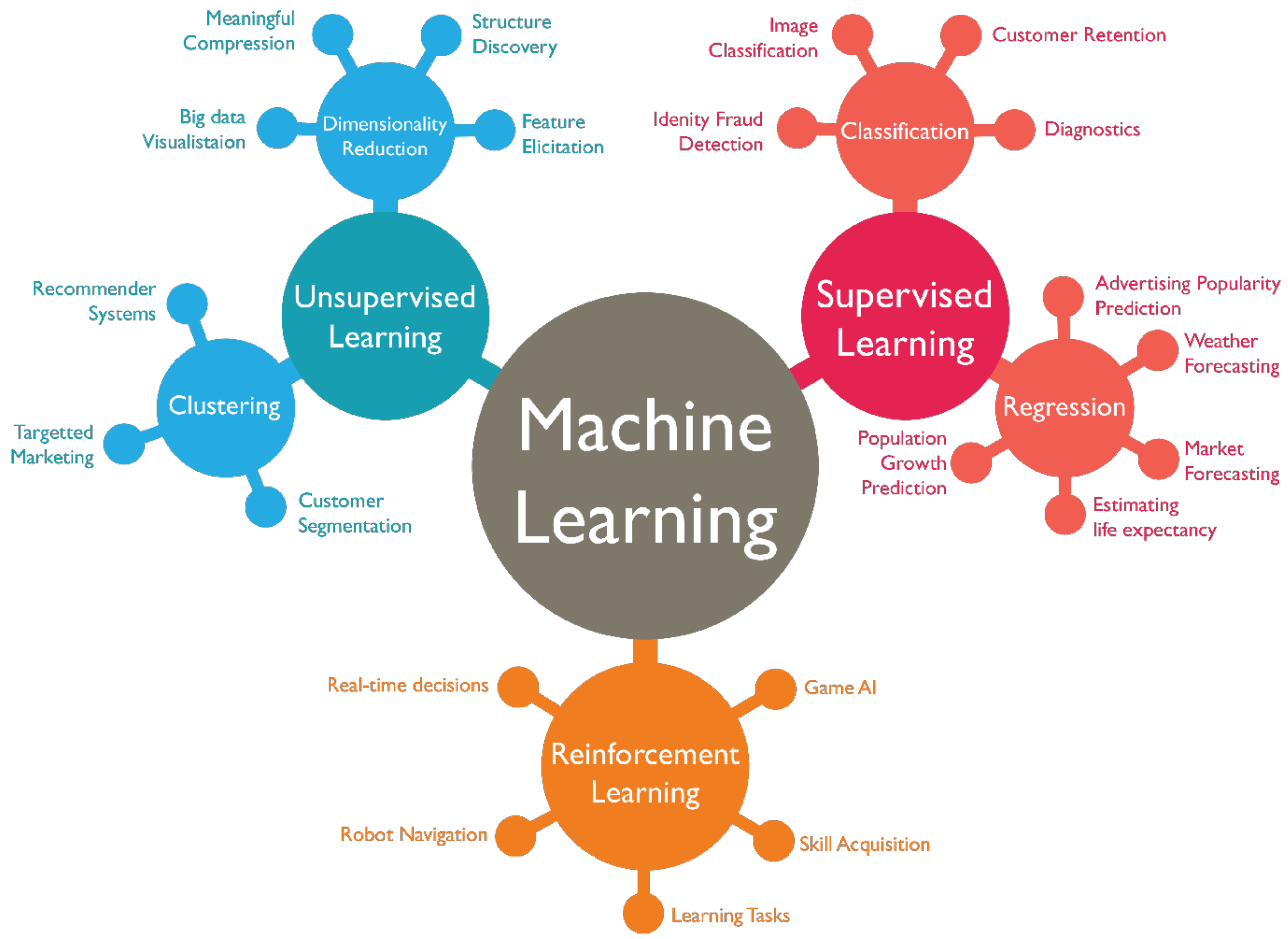
**Richard Tyson (University of Glasgow)**


Prague (June 28,2023)

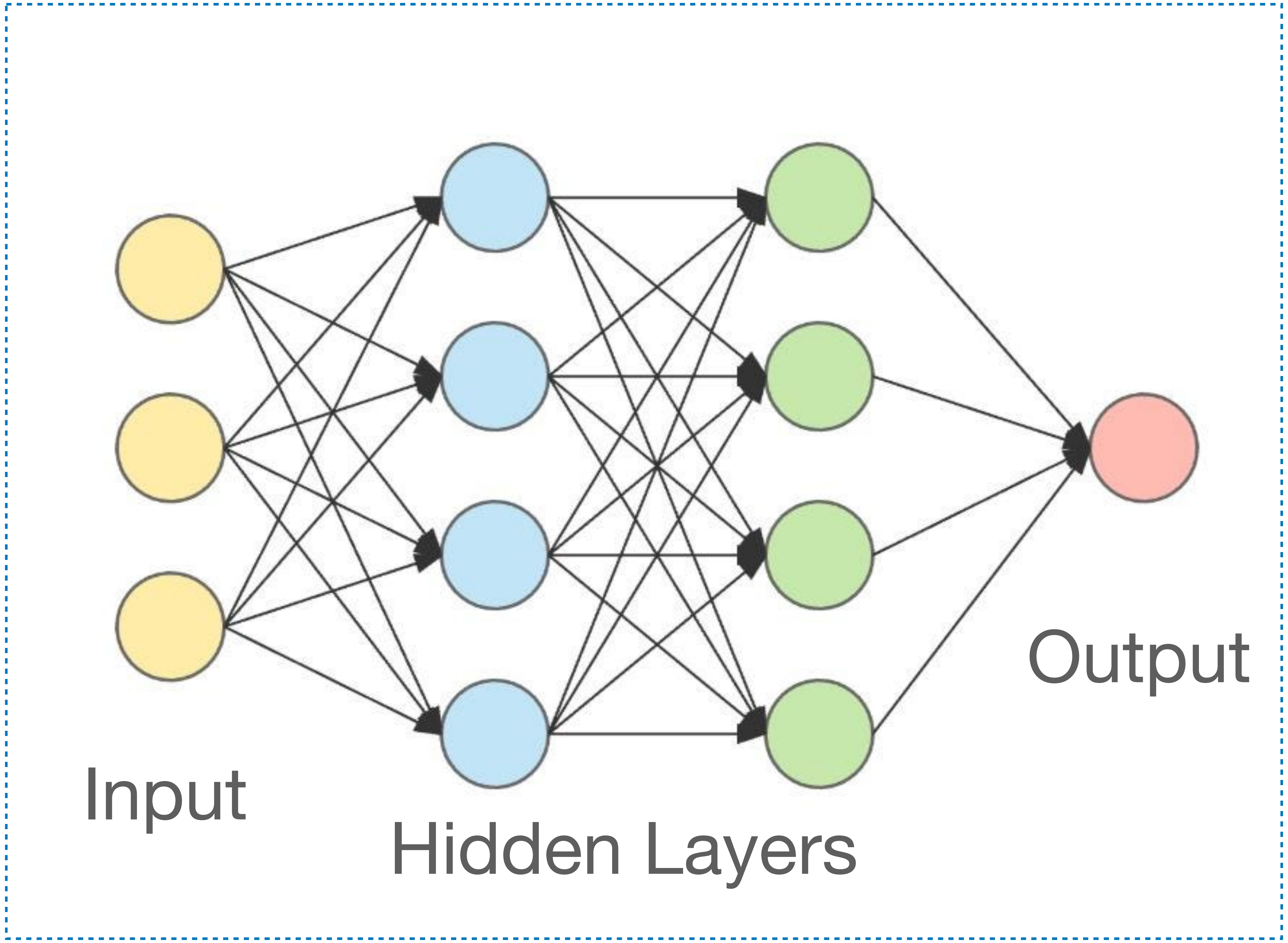
## Outlook



- Outline:
  - Machine Learning (brief introduction)
  - Track identification in Drift Chambers
  - Drift Chamber Data De-Noising
  - Impact on the experiment outcome
  - RICH (Ring Cherenkov) Particle Identification



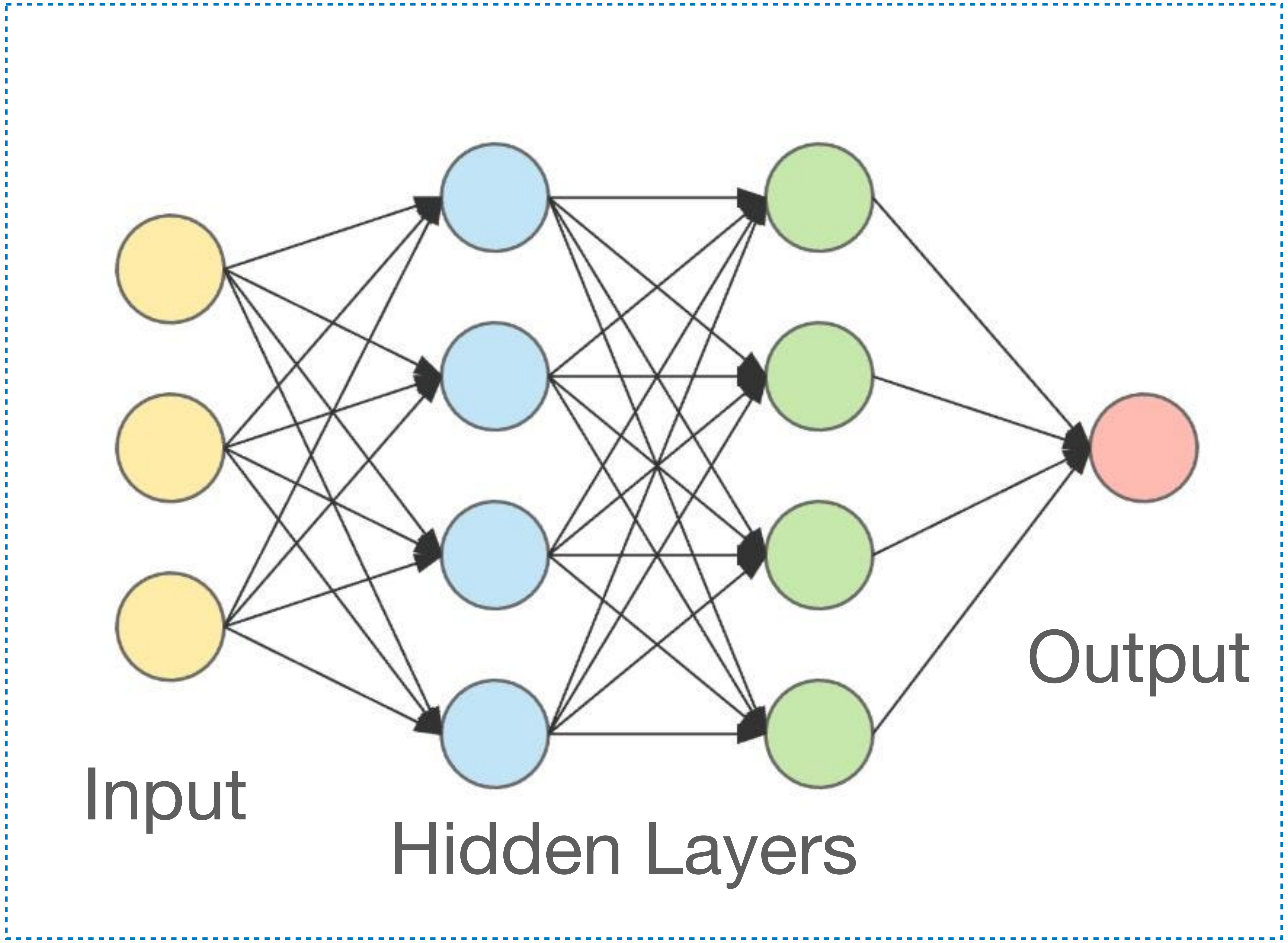

- Machine Learning (ML) is part of artificial intelligence.
- Machine learning is a field of inquiry devoted to understanding and building methods that 'learn', that is methods that leverage data to improve performance on some set of tasks.
- Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.



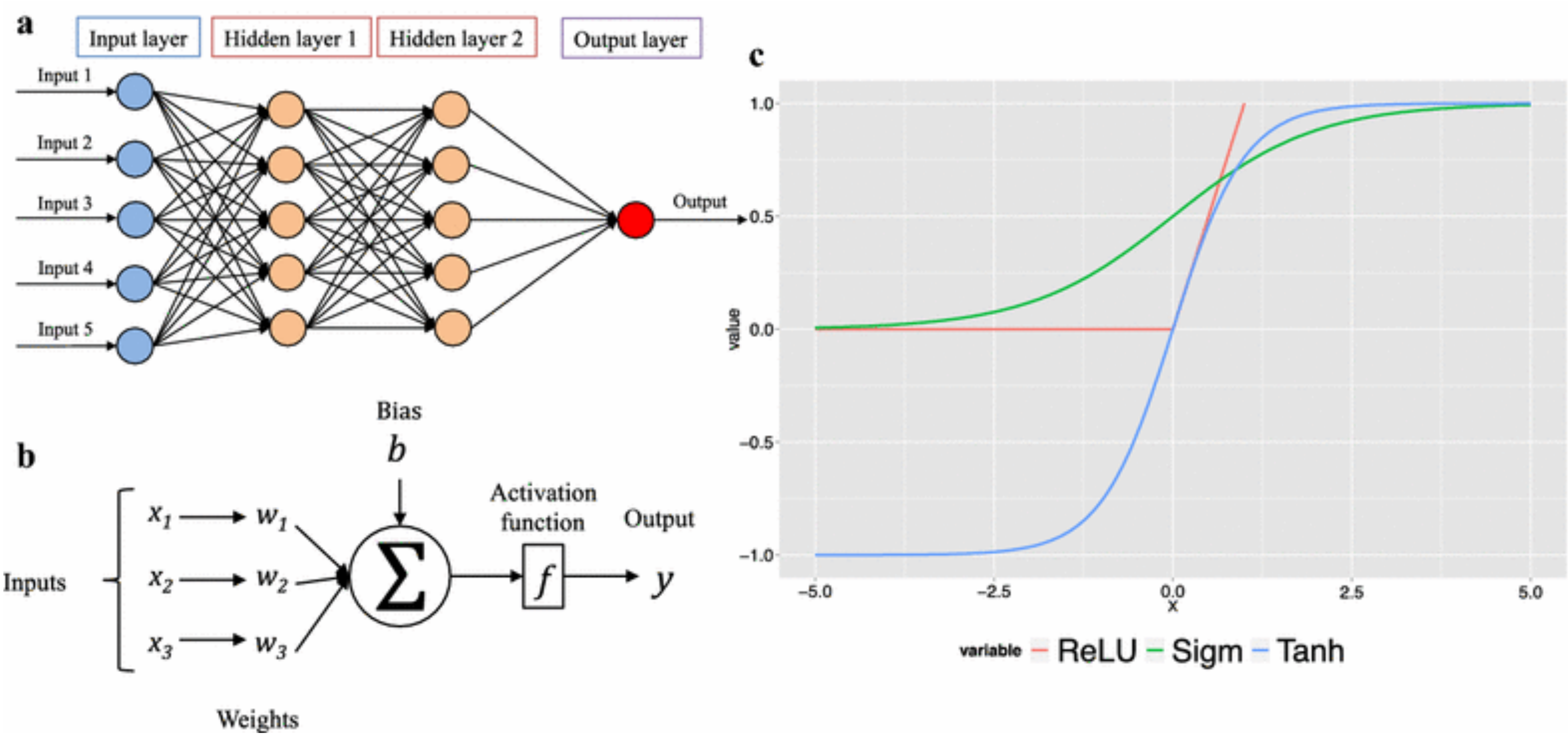

"Iron"

# Outlook

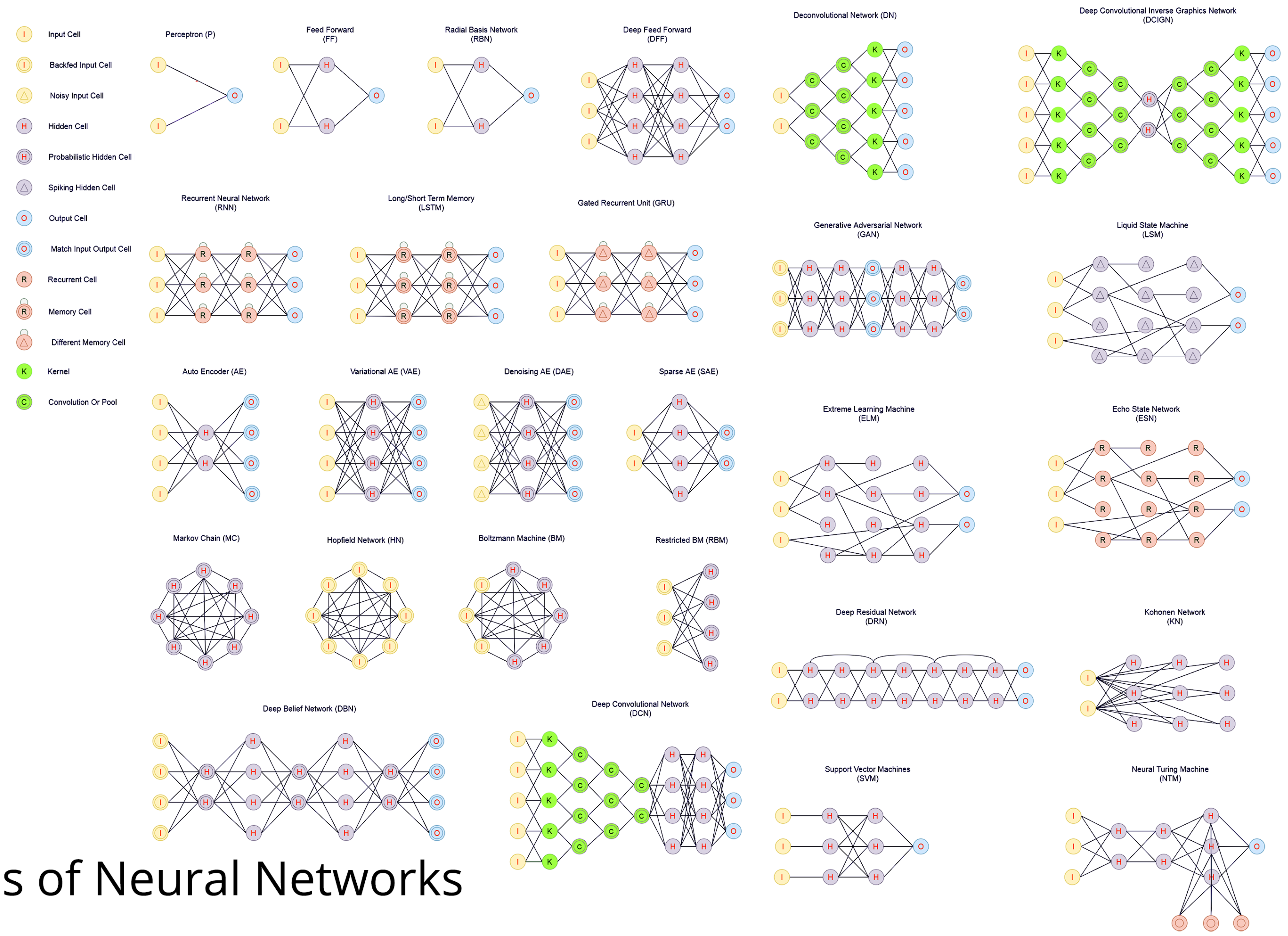





"Iron"


a
Input layer
Hidden layer 1
Hidden layer 2
Output layer
Input 1
Input 2
Input 3
Input 4
Input 5
Output
b
Bias
b
Activation function
Output
Inputs
x1 → w1
x2 → w2
x3 → w3
Σ
f
y
Weights
c
1.0
0.5
0.0
−0.5
−1.0
value
−5.0
−2.5
0.0
2.5
5.0
x
variable ReLU Sigm Tanh

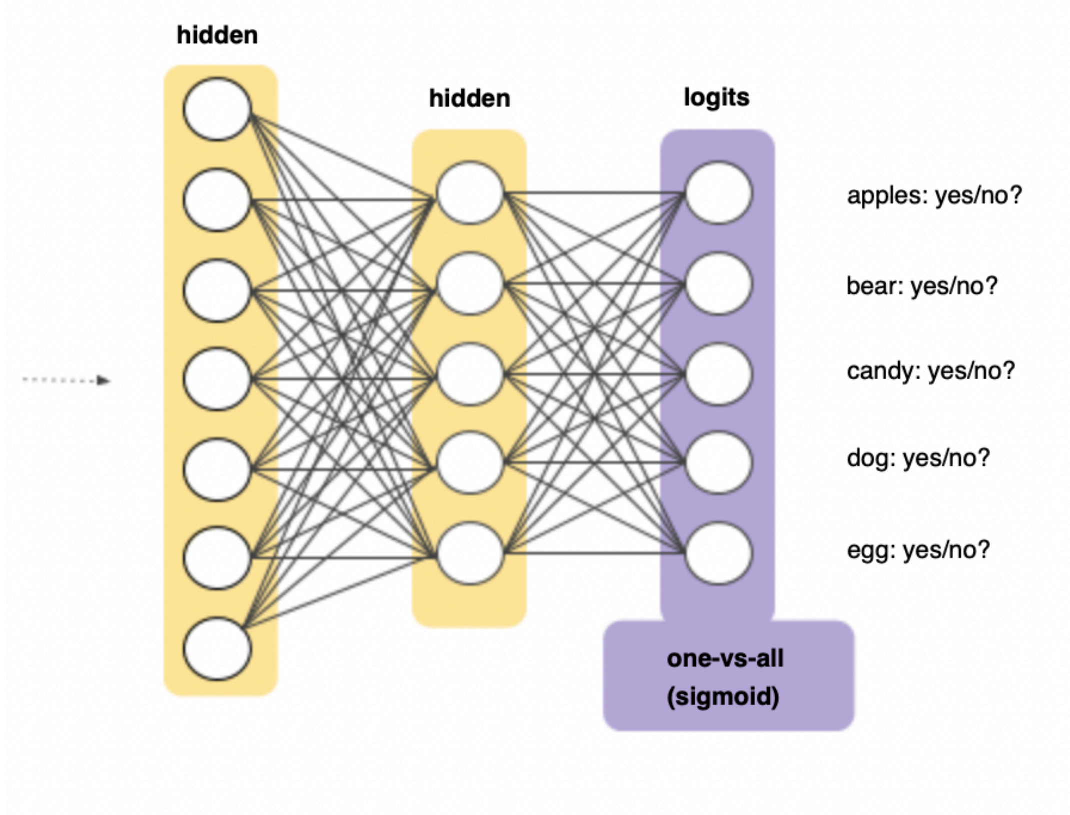
# Artificial Intelligence/Machine Learning




Main Types of Neural Networks

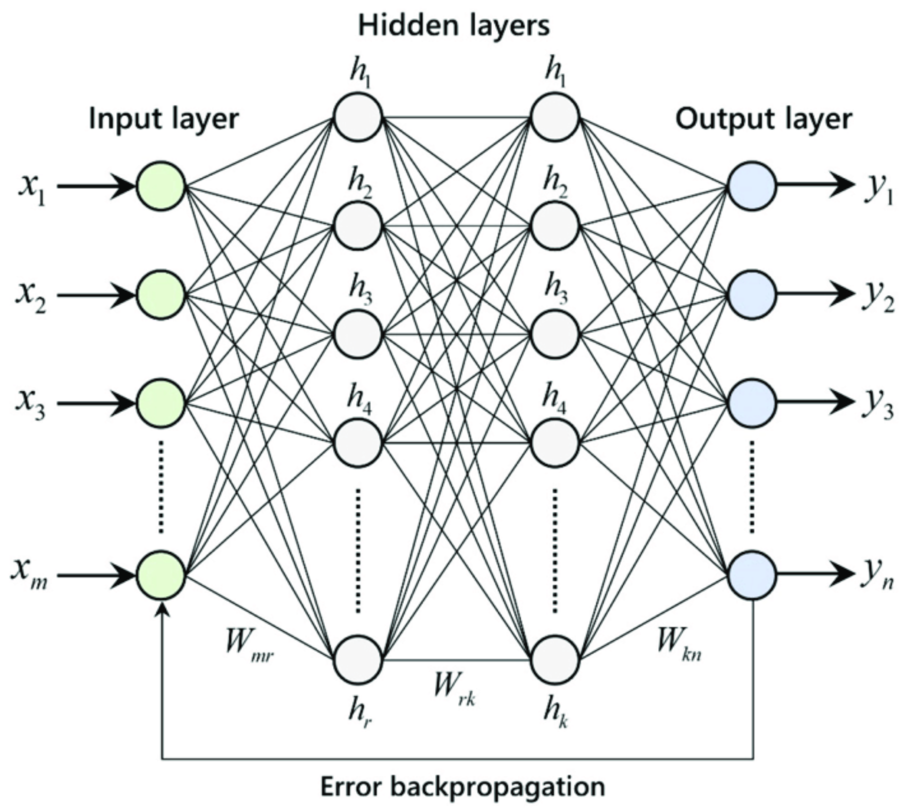
# Artificial Intelligence/Machine Learning

## Classifier Neural Network



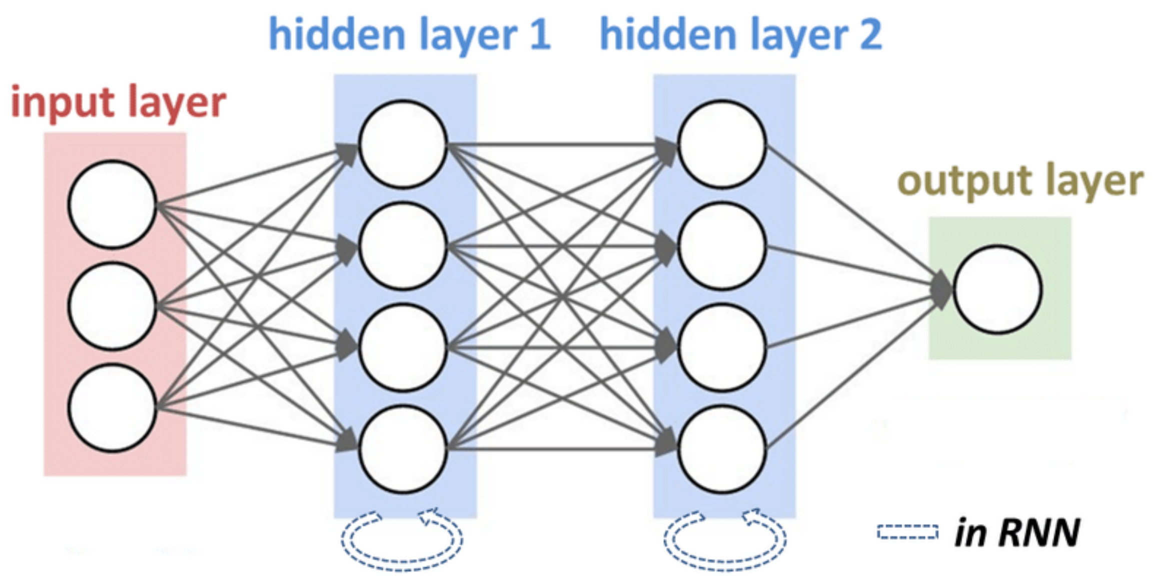

- Based on the input (image/vector) decide what type (or class) of data it is.
- Used to identify dogs/cats in the image
- Identify what king of particle it is based on the signals from detector components

## Regression Neural Network



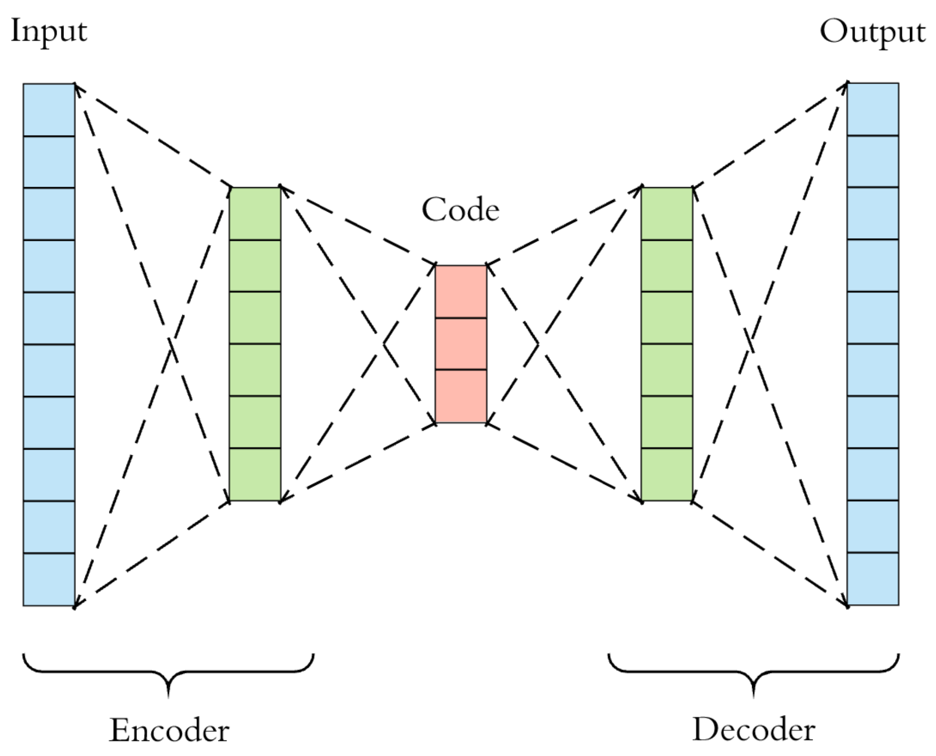

- For the given input (image/vector) calculate some values characteristic of the input
- Calculate the amplitude of a peak given points of the histogram
- Predict the speed of the object from series of measurements

## Recurrent Neural Network




- For the given series predict the next value.
- Predict the next word in the sentence.
- Predict the next point for a given trajectory

## Auto-Encoder Neural Network



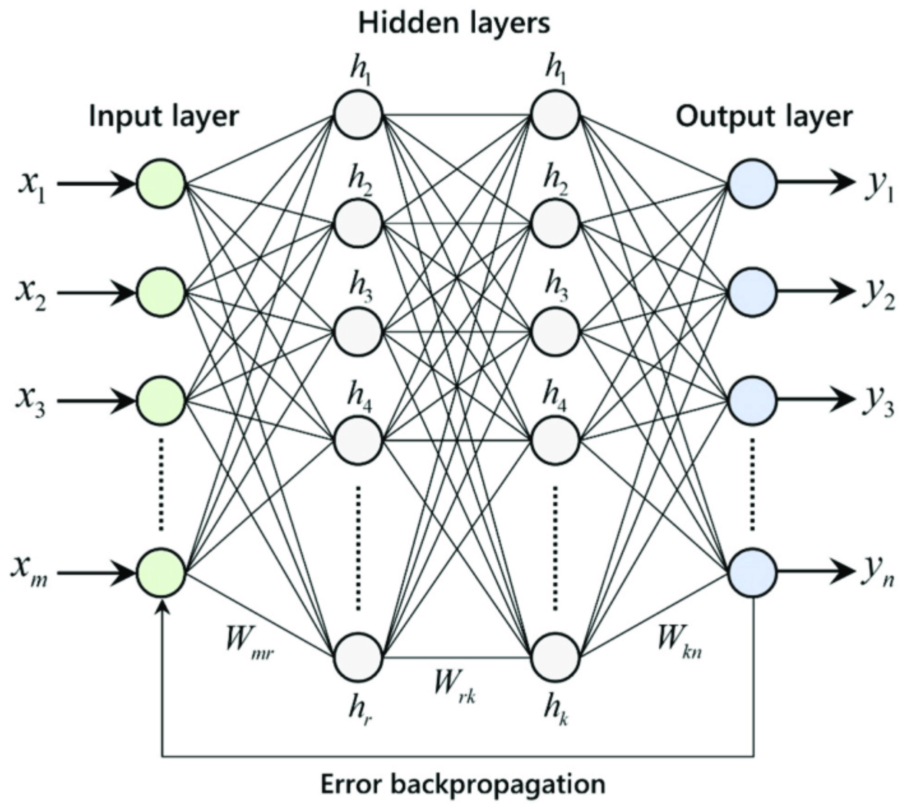

- Modify the input data to correct for some deficiencies
- Used for de-noising images
- Predicting missing information in the input

## Classifier Neural Network




- Based on the input (image/vector) decide what type (or class) of data it is.
- Used to identify dogs/cats in the image
- Identify what king of particle it is based on the signals from detector components

## Regression Neural Network




- For the given input (image/vector) calculate some values characteristic of the input
- Calculate the amplitude of a peak given points of the histogram
- Predict the speed of the object from series of measurements

## Recurrent Neural Network

- If one collects all world data and feeds it to Regression Neural Network the answer should come out 42.

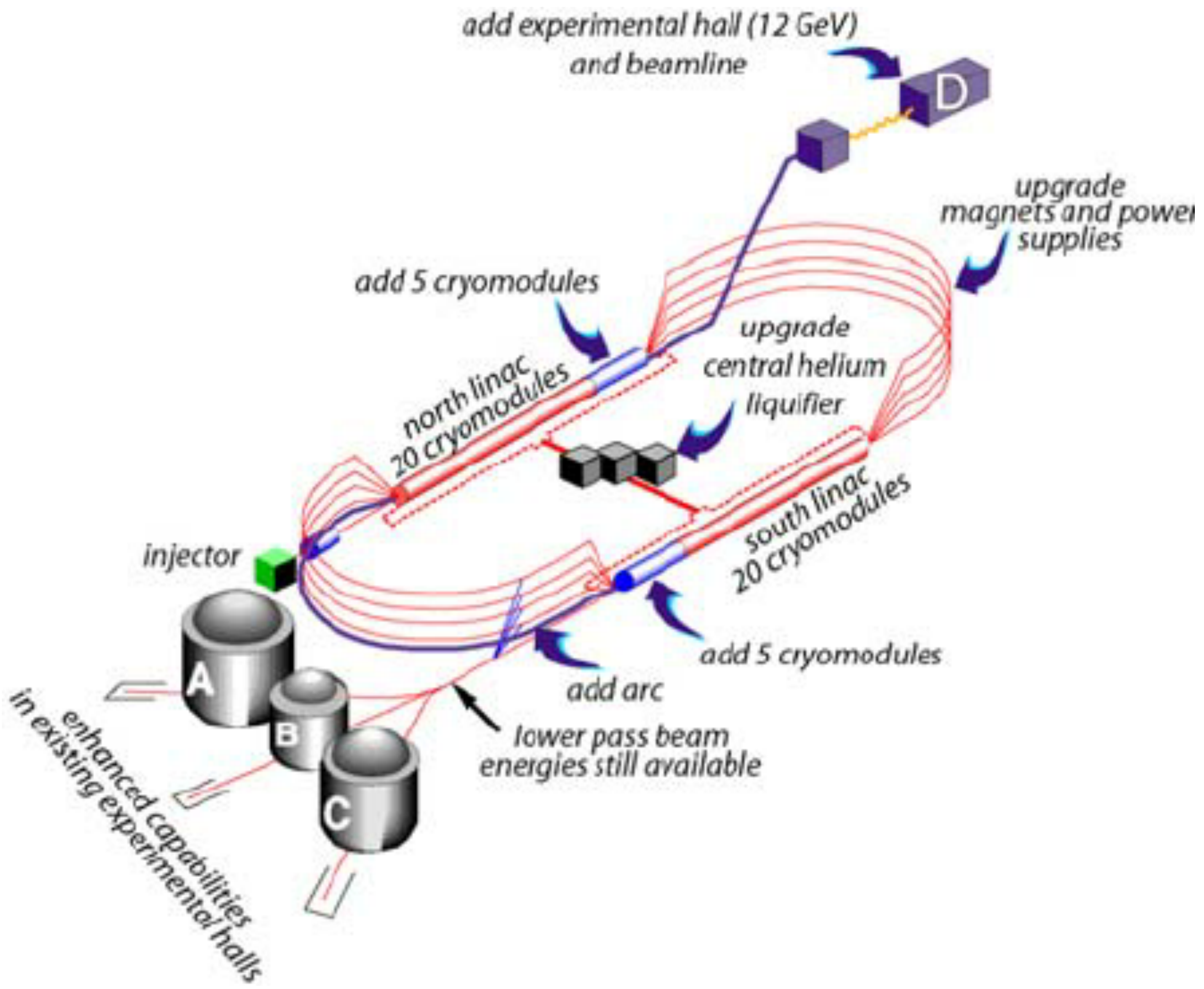


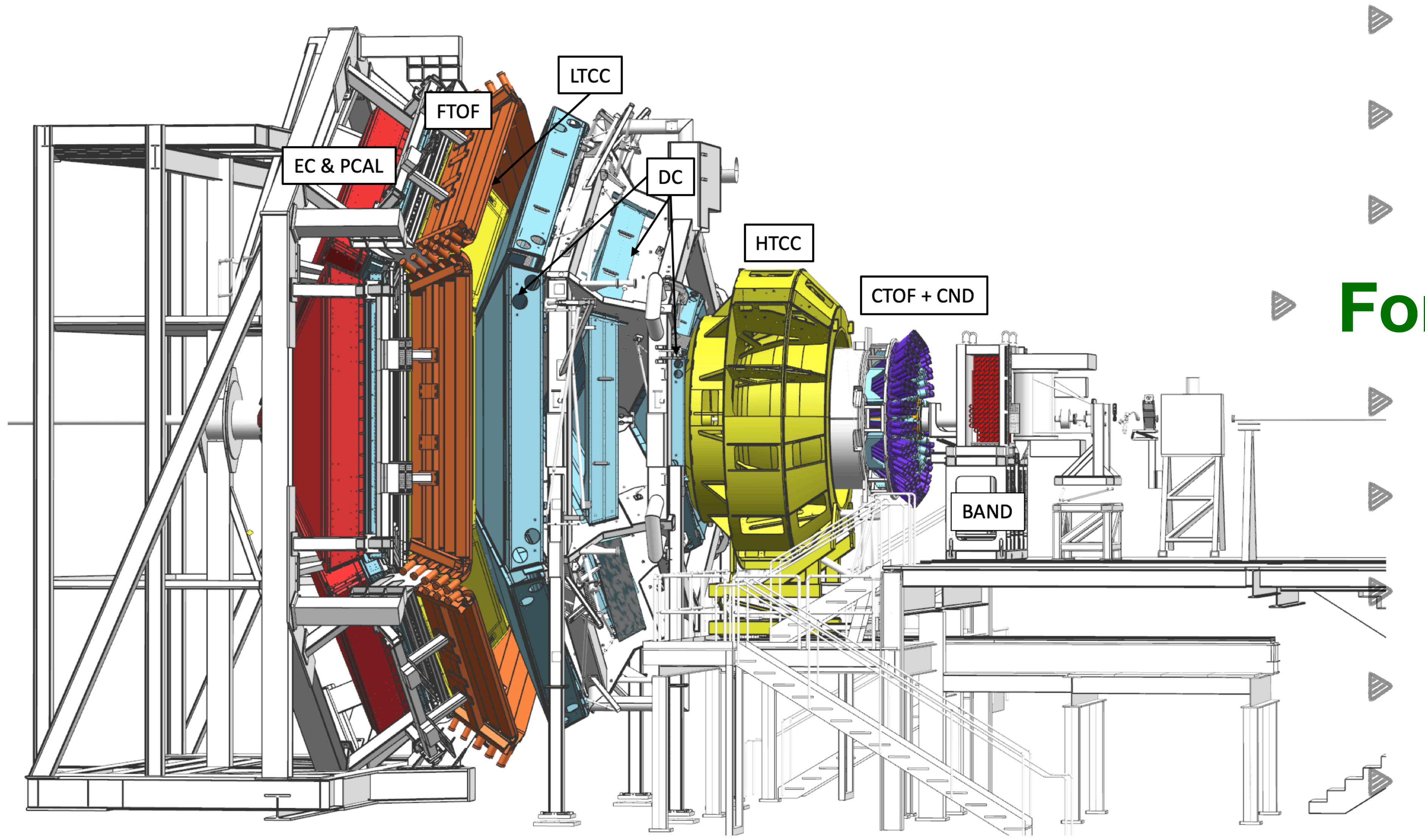

- For the given series predict the next value.
- Predict the next word in the sentence.
- Predict the next point for a given trajectory

## Auto-Encoder Neural Network



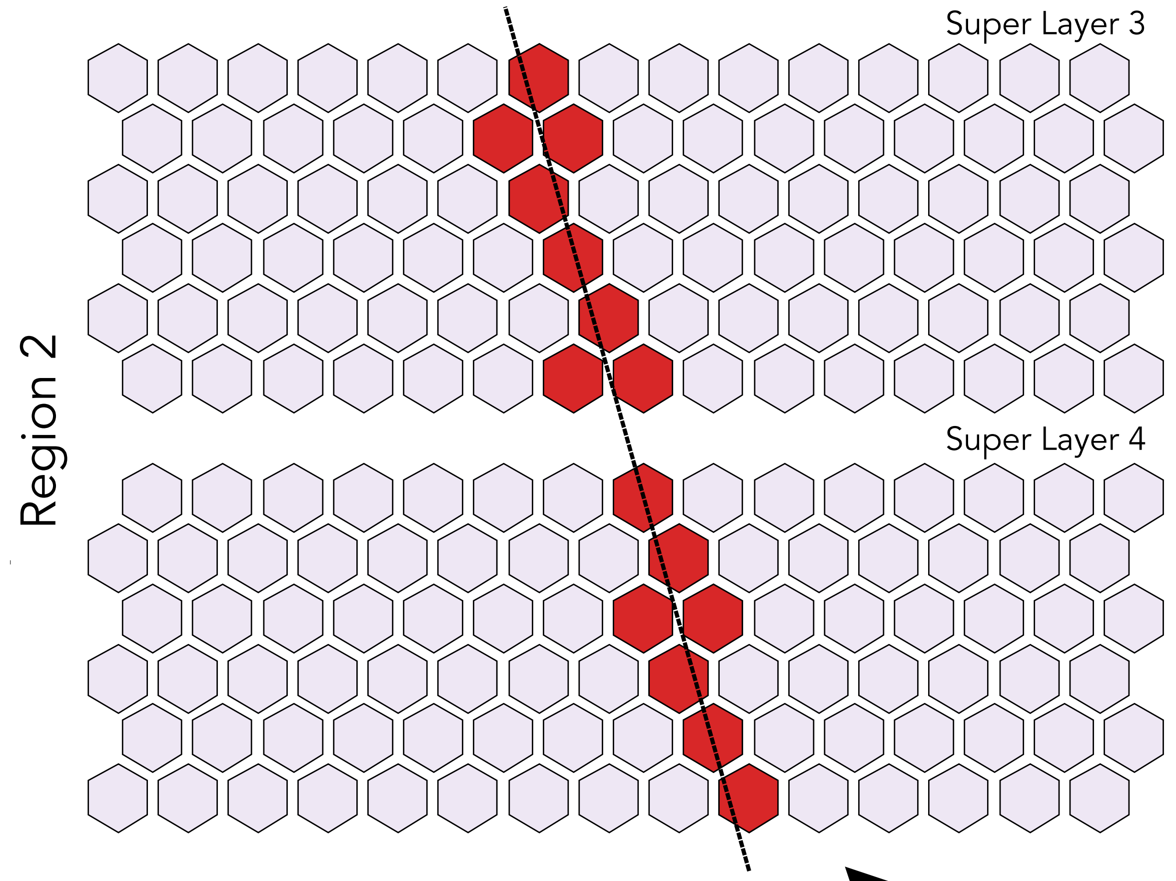

- Modify the input data to correct for some deficiencies
- Used for de-noising images
- Predicting missing information in the input

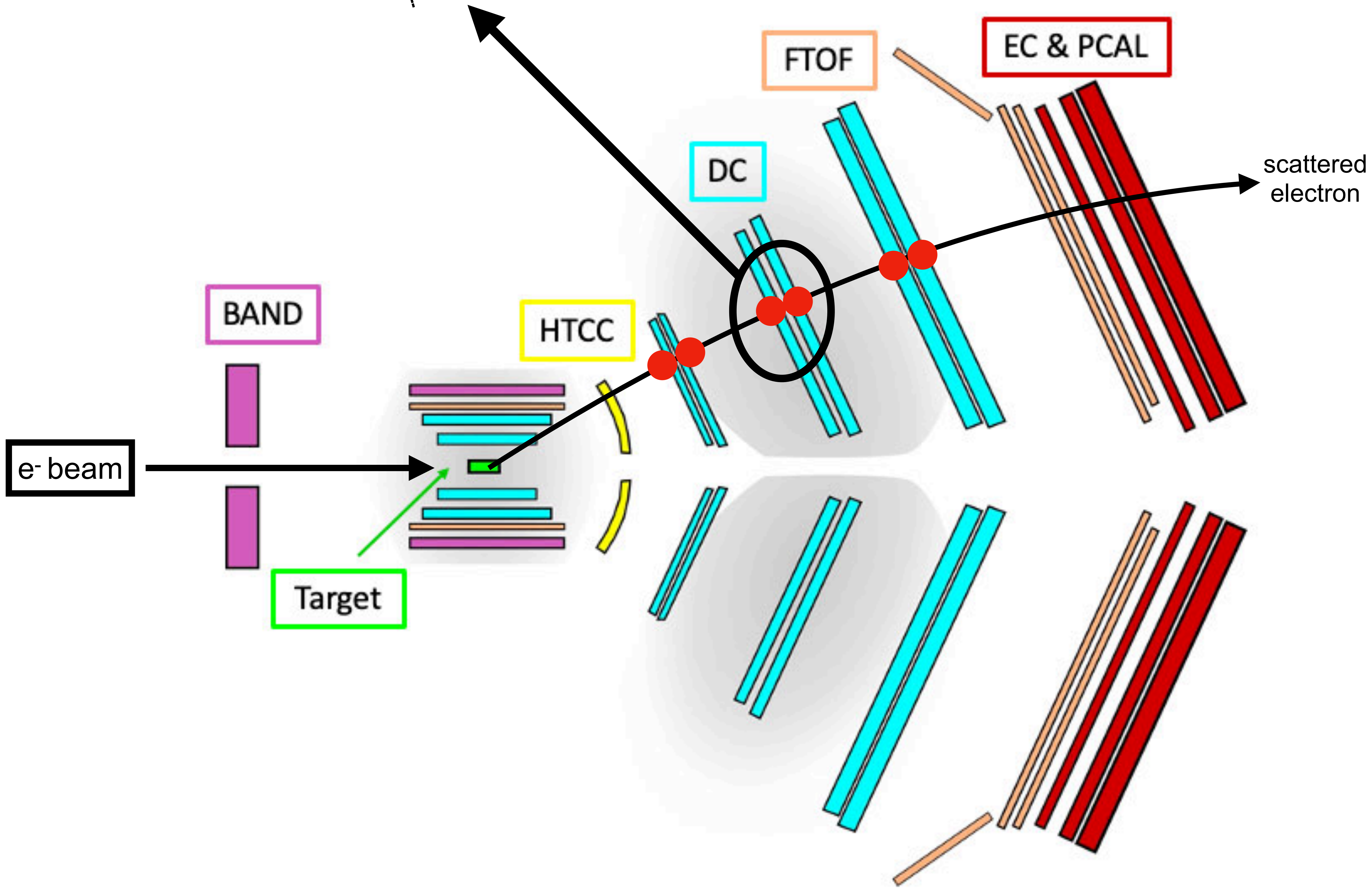




- CEBAF
  - 12 GeV electron beam distributed to 4 experimental halls
  - Each experimental hall contains a detector system for specific experiments
- **Hall-B:**
  - CEBAF Large Acceptance Spectrometer (CLAS12) Located in Hall-B
  - **Central Detector:**
    - Silicon Tracker
    - Time-Of-Flight
  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

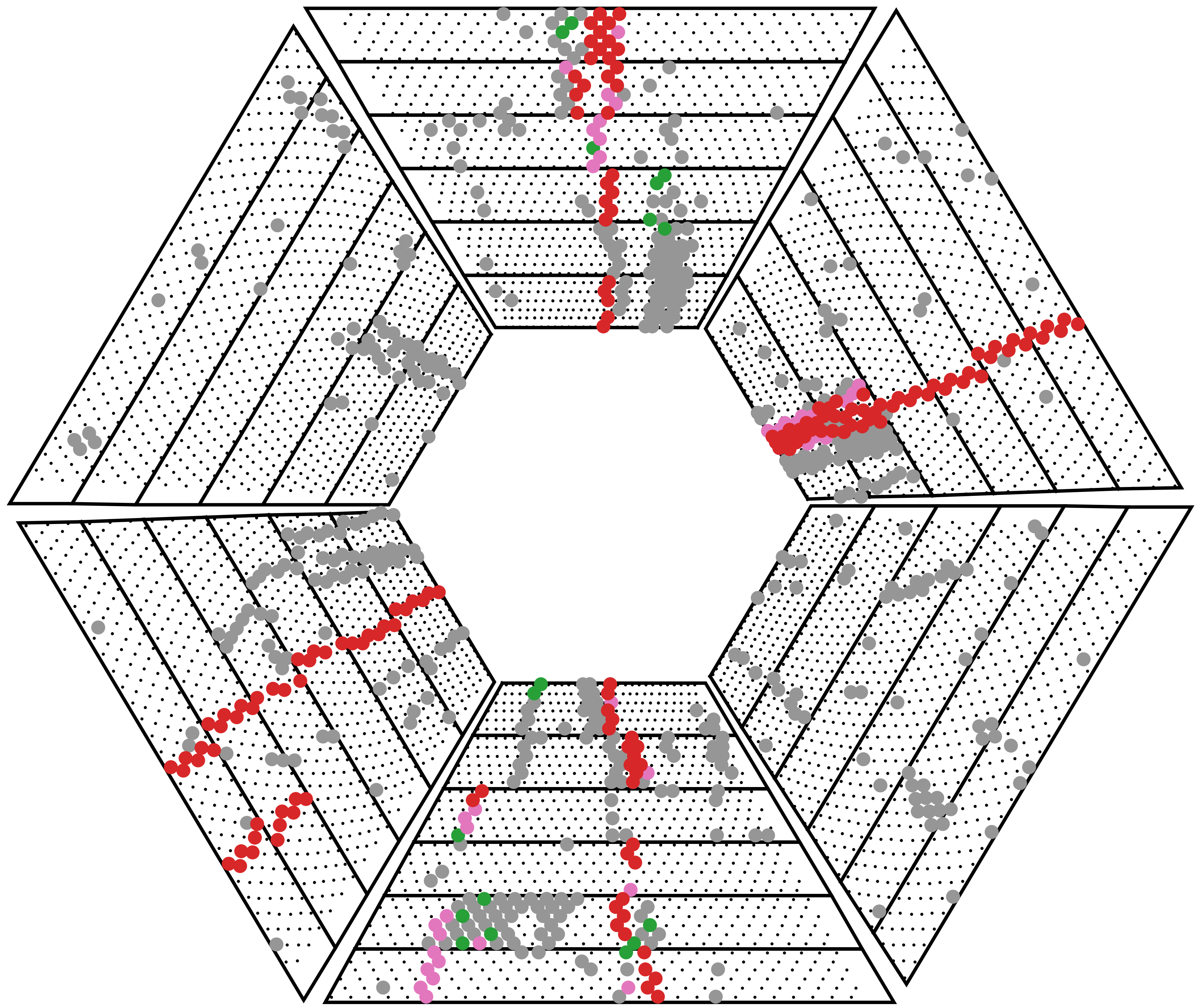
# CLAS12 Detector

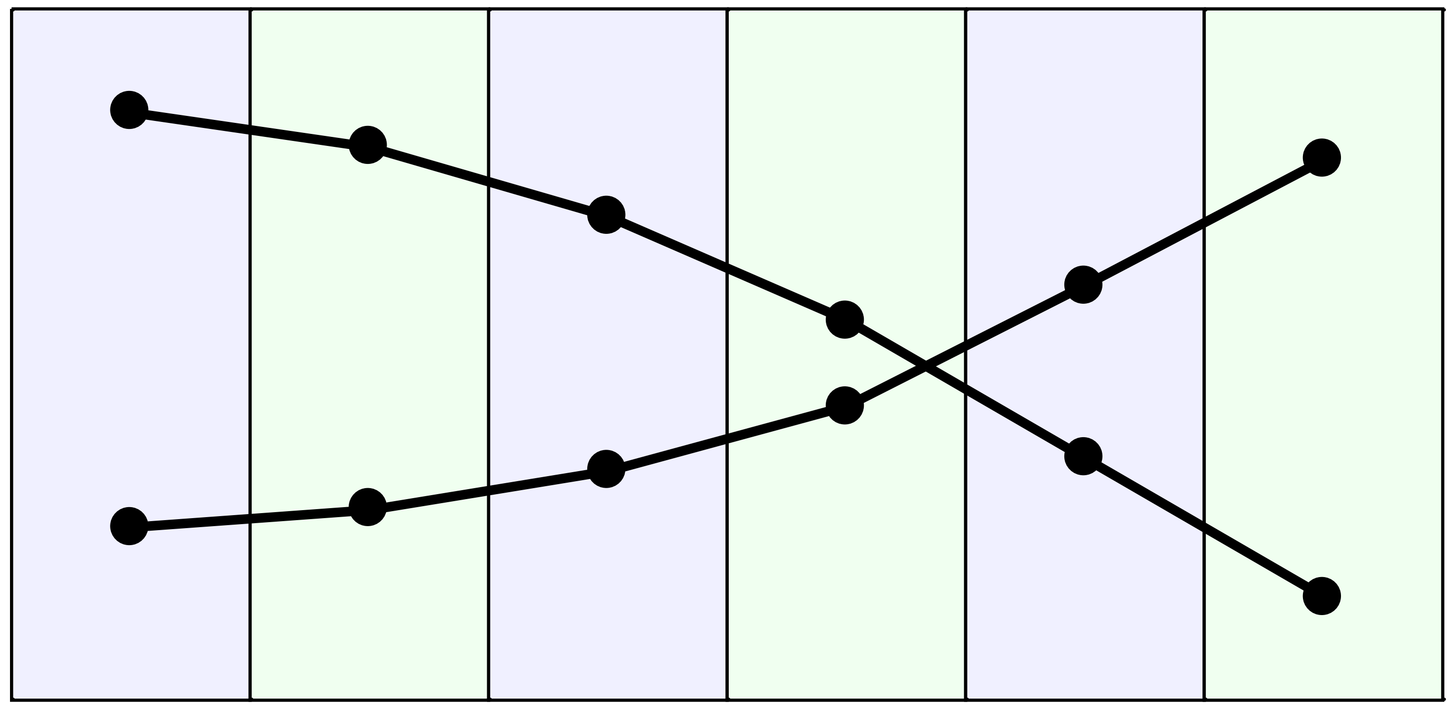



- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory



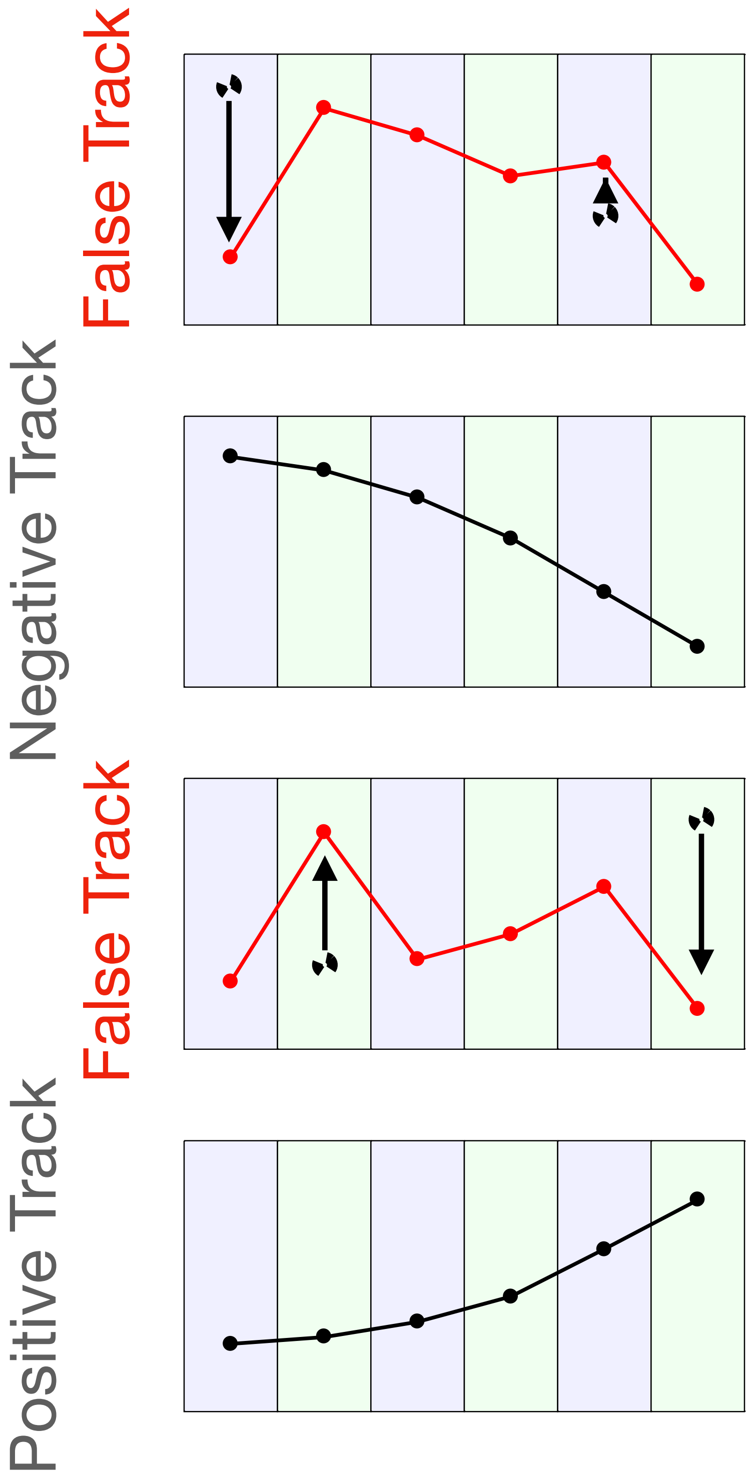

- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can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

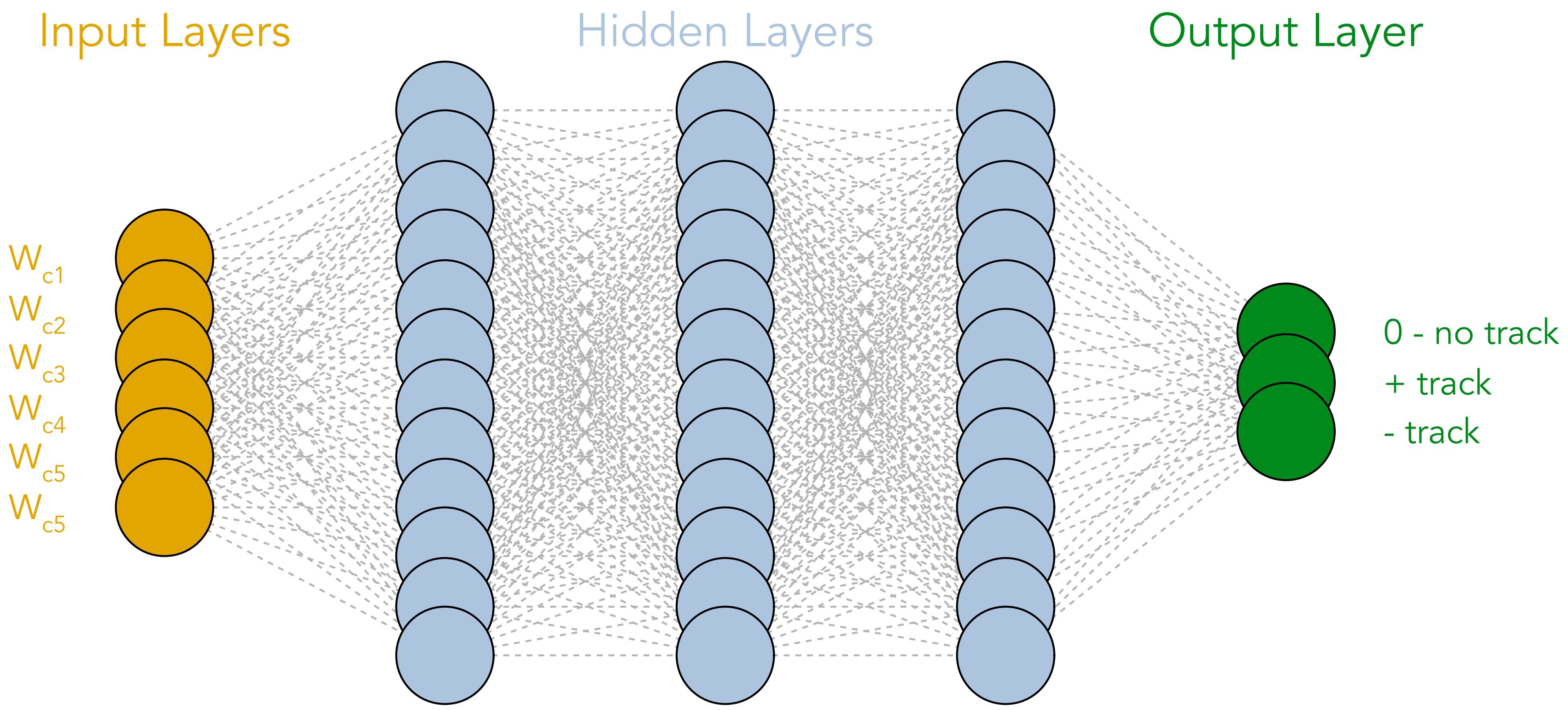




- True tracks are identified by conventional algorithms from real data.
- One negative and one positive track (different curvature due to magnetic field)
- False tracks are constructed by interchanging randomly one or two clusters with the clusters from the other track in the event
- Training sample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

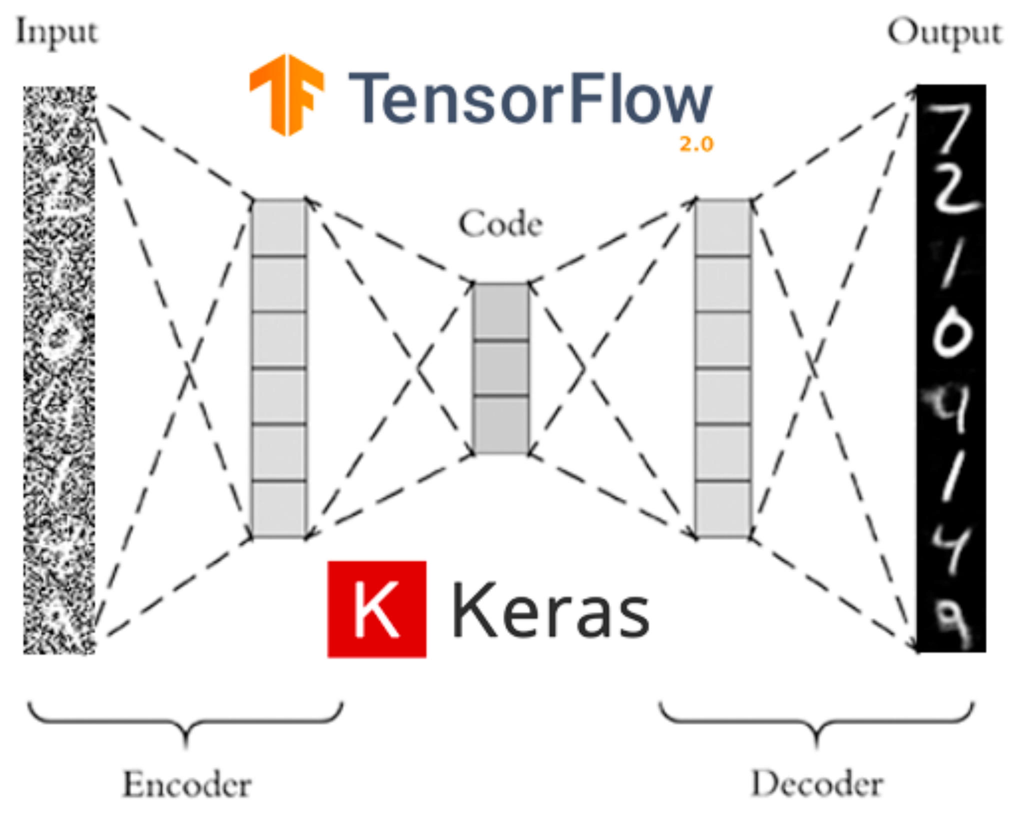
- The average wire position in each super layer is used as an input to Multi-Layer Perceptron (MLP)
- The network is trained on 6 inputs and produces three outputs:
  - False track
  - Negative Track
  - Positive Track



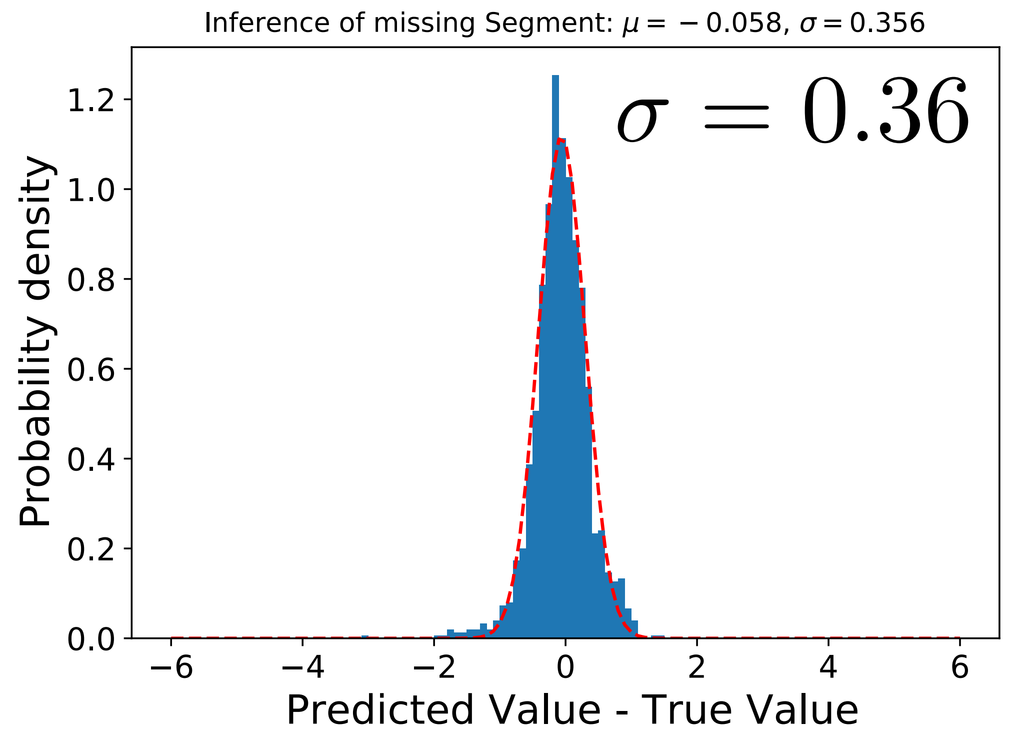



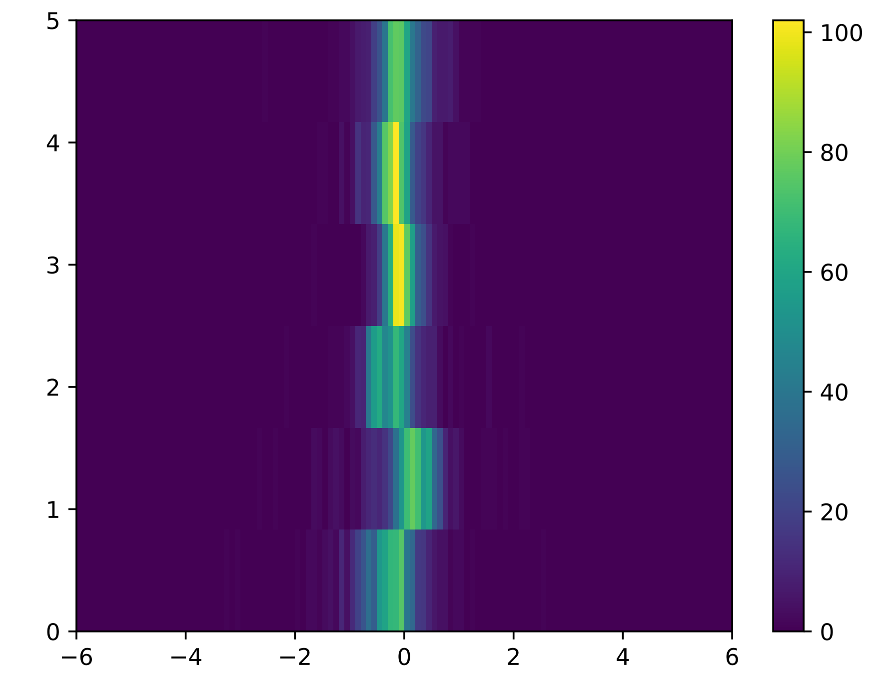
# Corruption Auto-Encoder

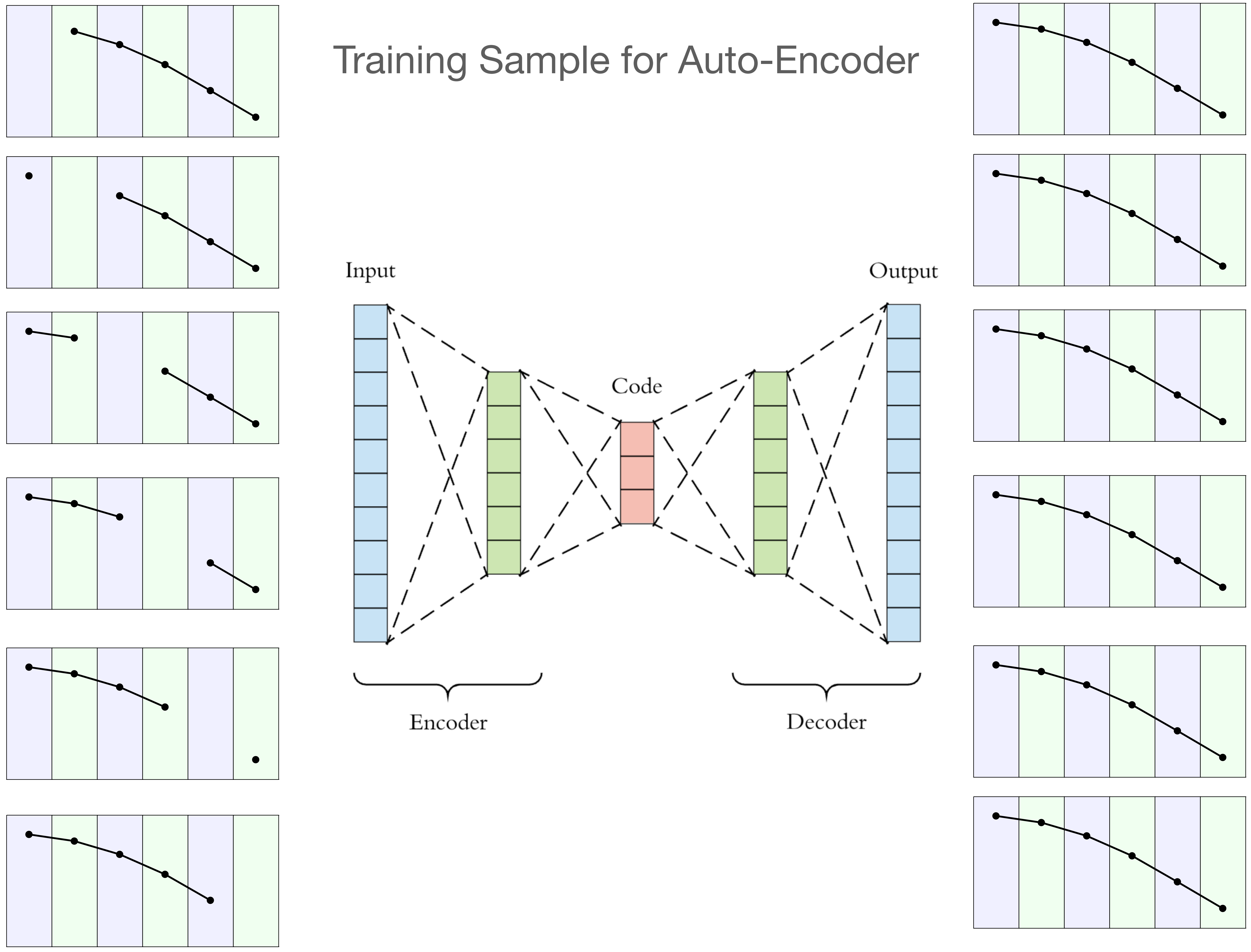
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- **Typically used for de-noising, but can be used for fixing glitches (our case).**




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

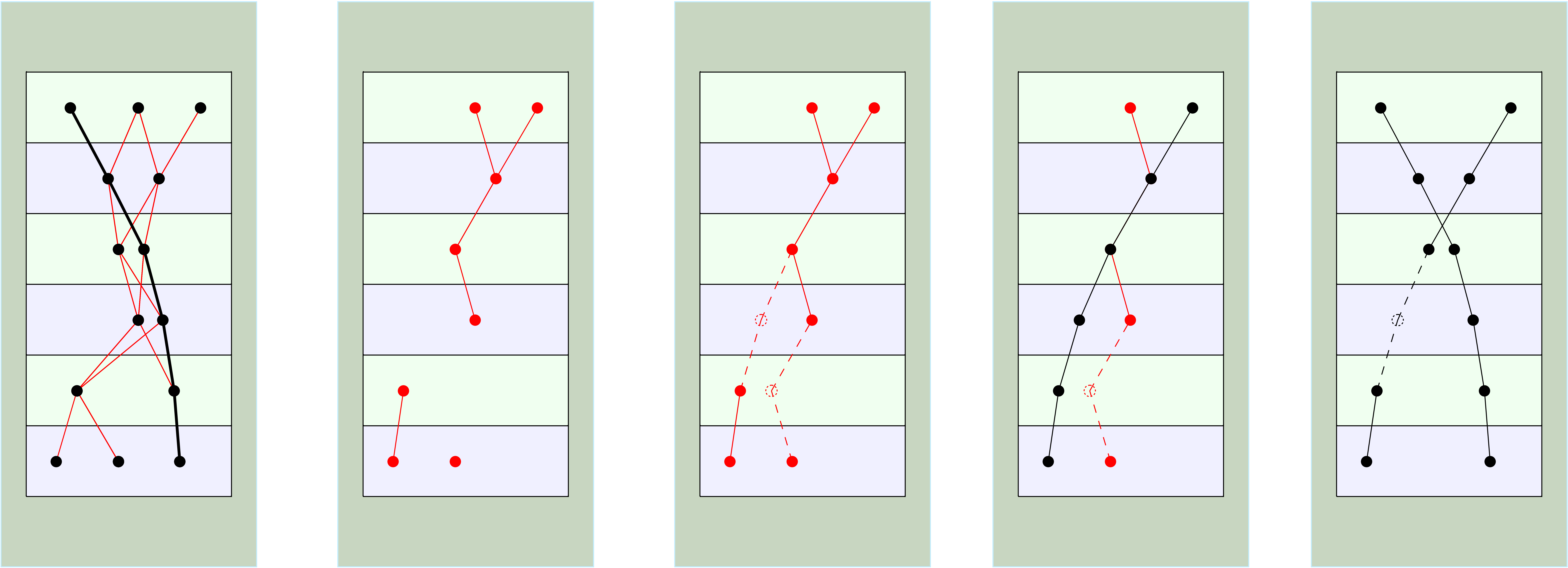





- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

# Putting things together



Classifier picks the correct track from 6 super-layer combinations

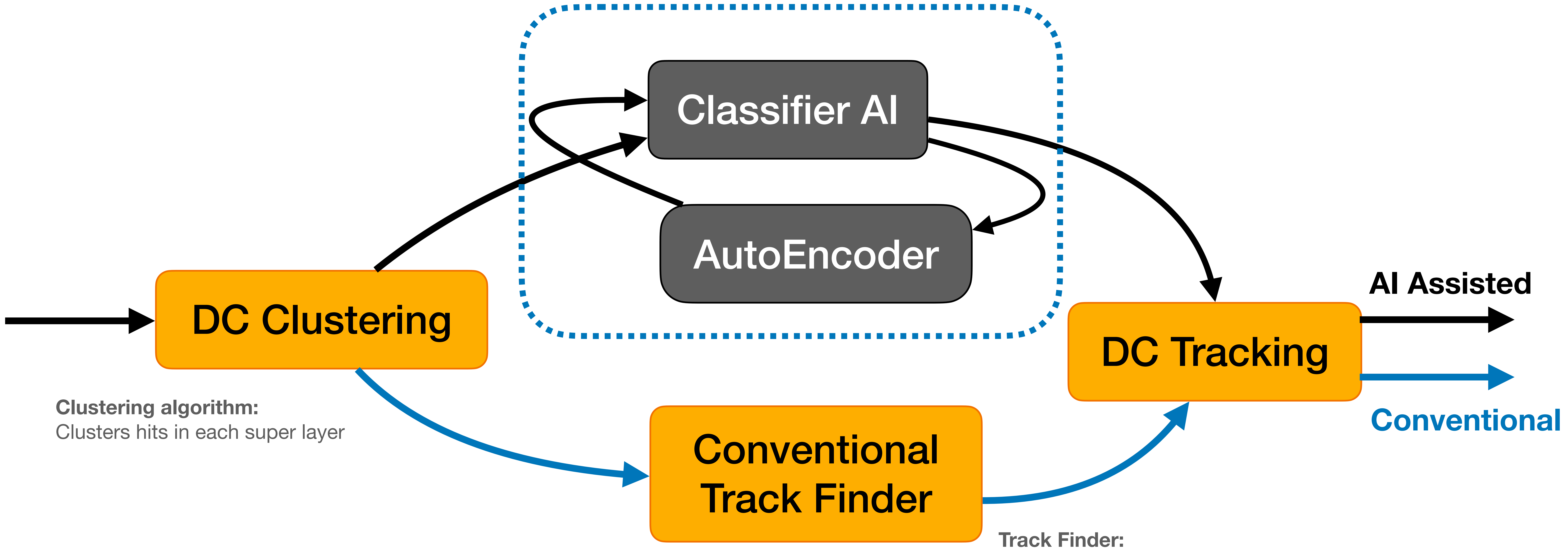
Remove all clusters belonging to identified track

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

Identify tracks using 6 super layer candidates with pseudo-clusters

Voila!

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

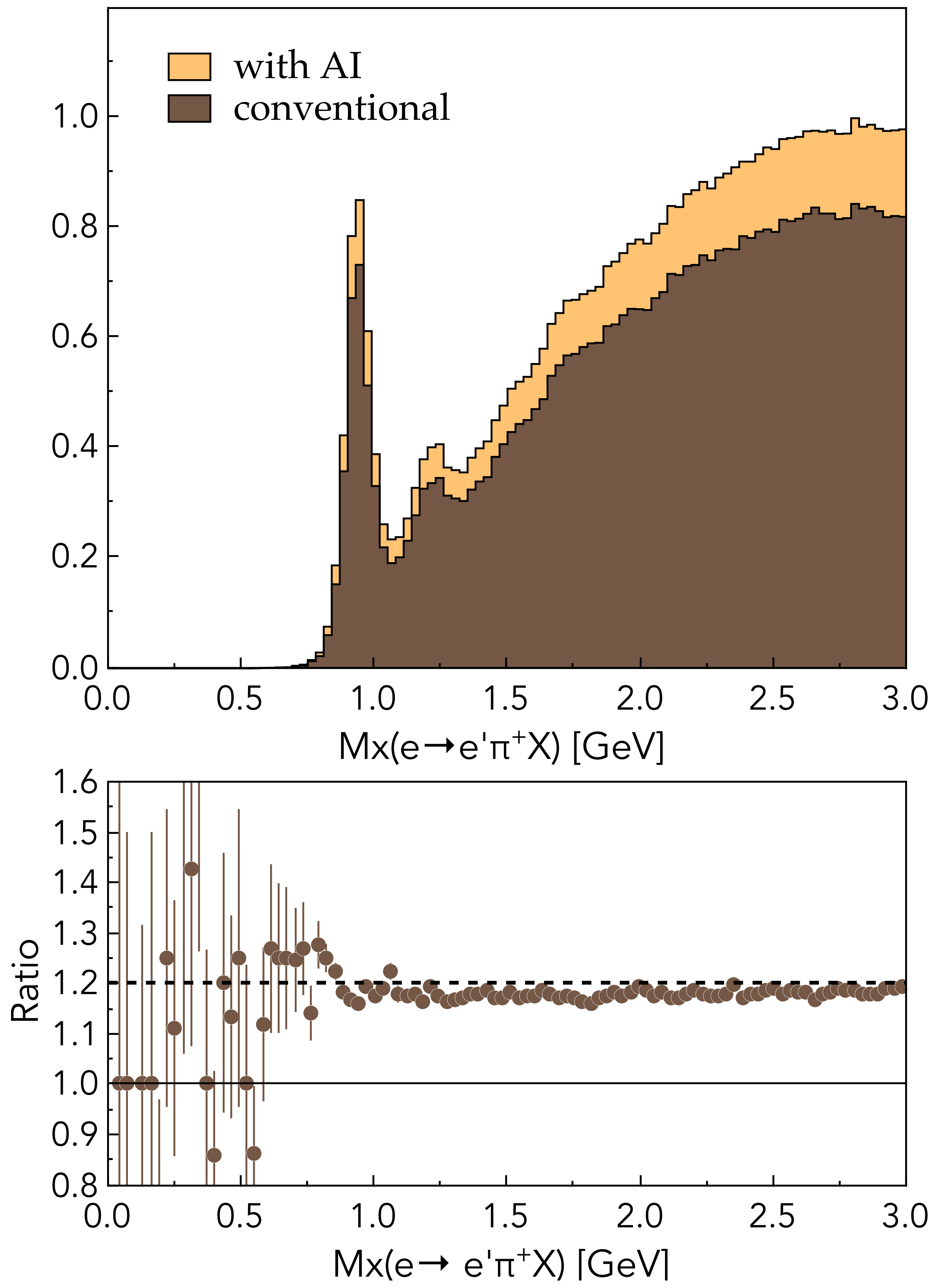



**Clustering algorithm:**
Clusters hits in each super layer

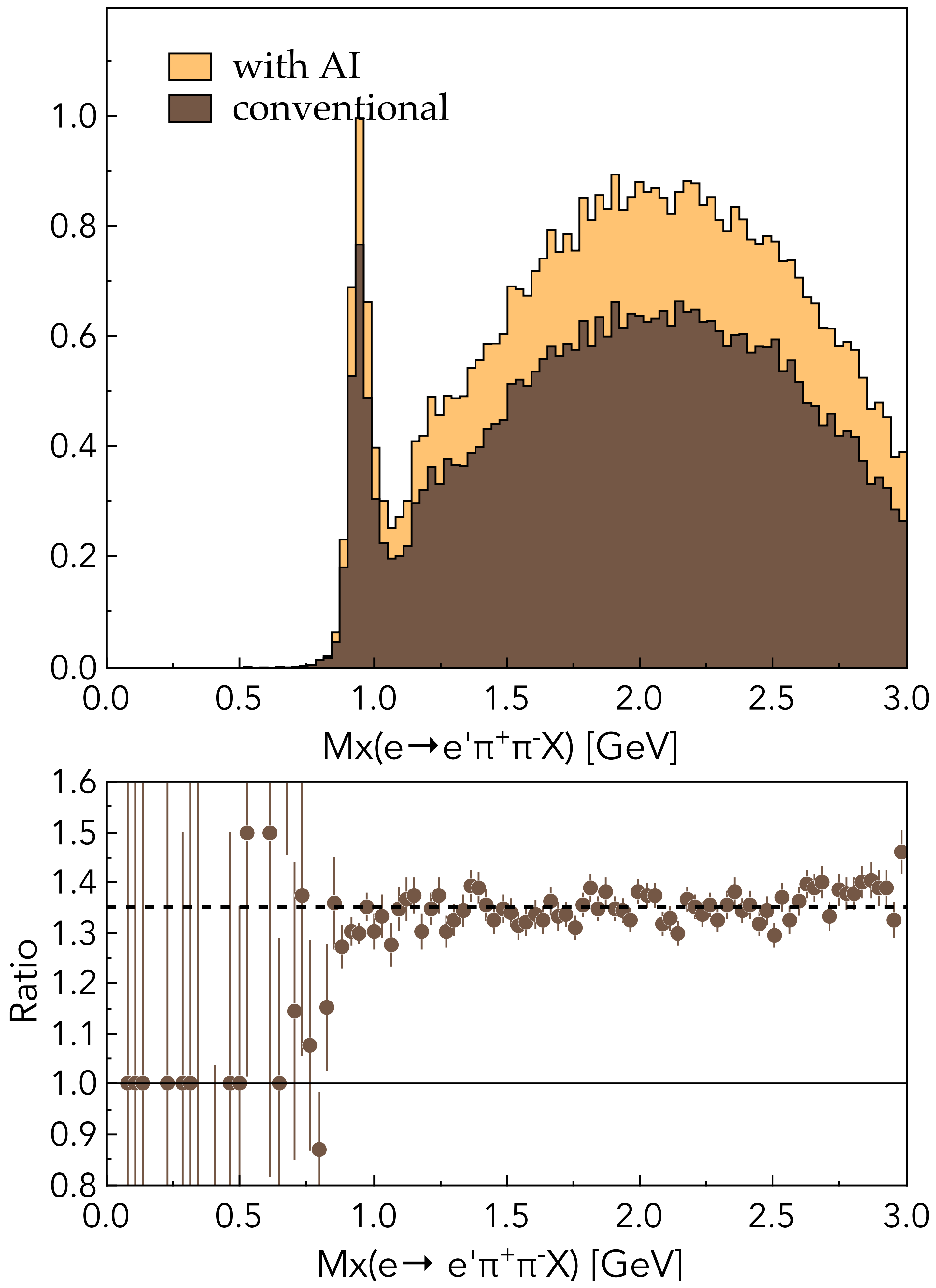
**Track Finder:**
Considers all combinations of 6 and 5 super layer track candidates Fits them and accepts/rejects them based on chi2.

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

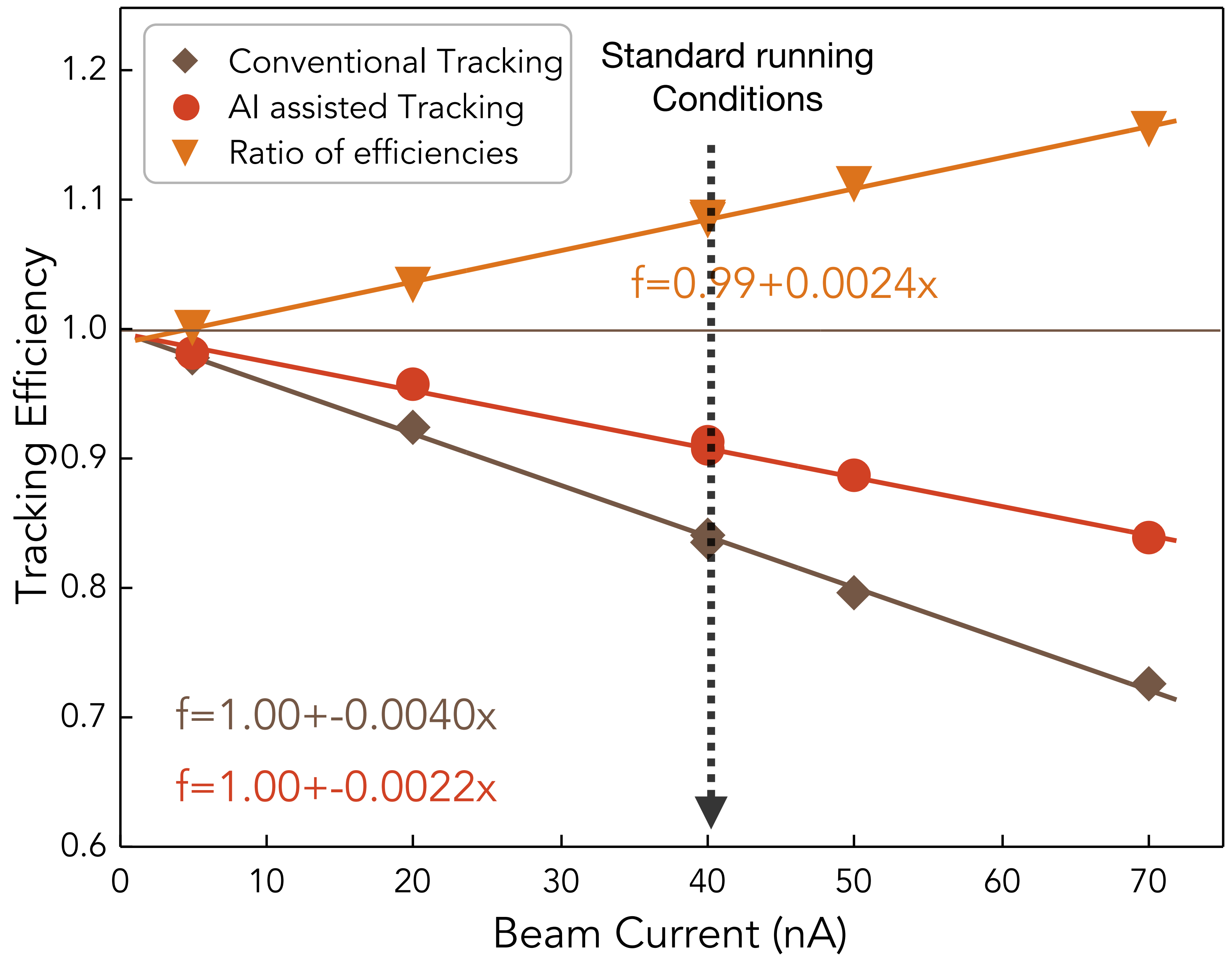
$$ep \to e'\pi^+(X)$$




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



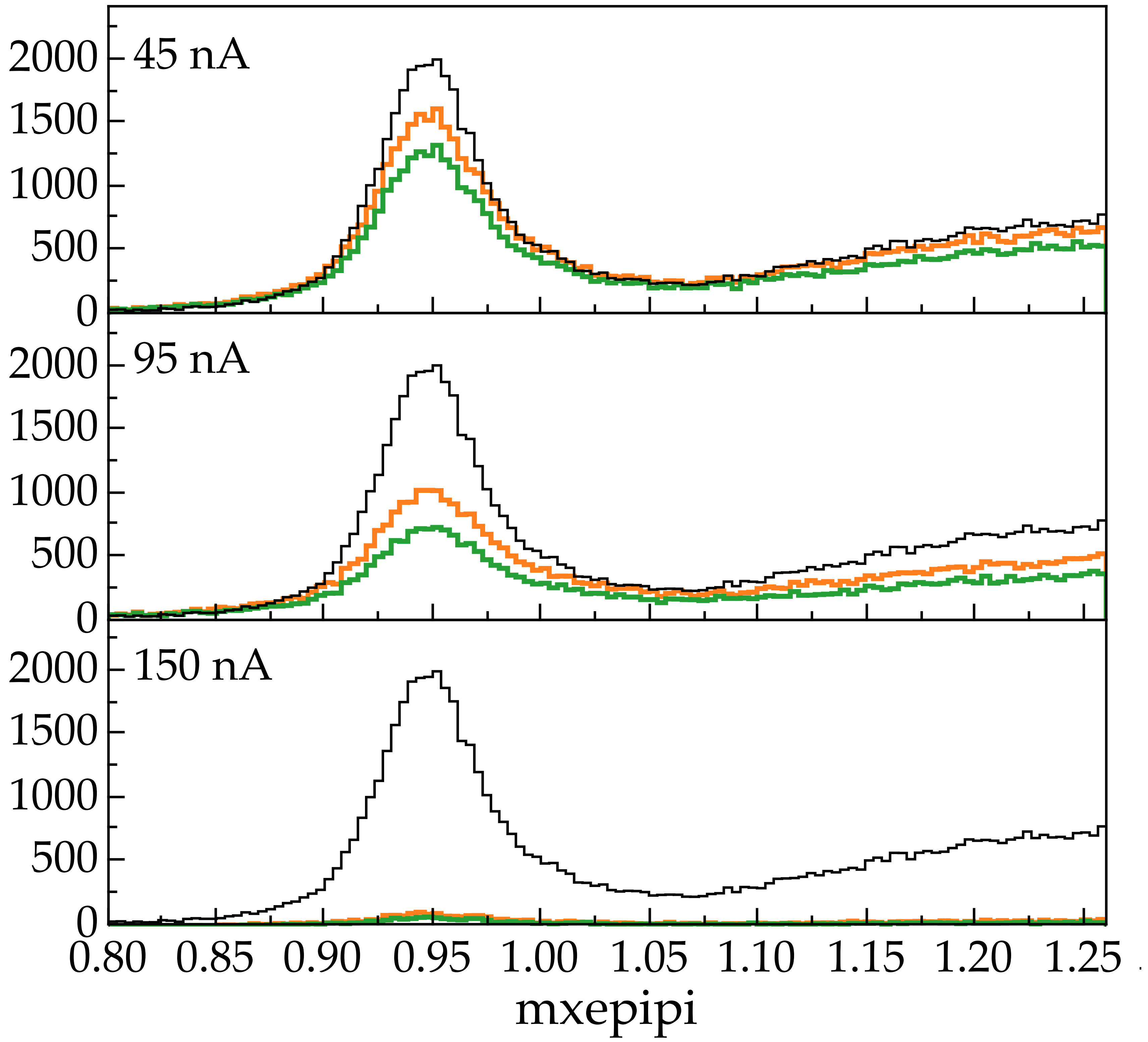

- Single particle efficiency increases by ~10%.
- The impact on physics for a multi-particle final state is dramatic (20% for the two-particle final state and ~35% for the three-particle final state)
- The tracking code speedup is ~30%.

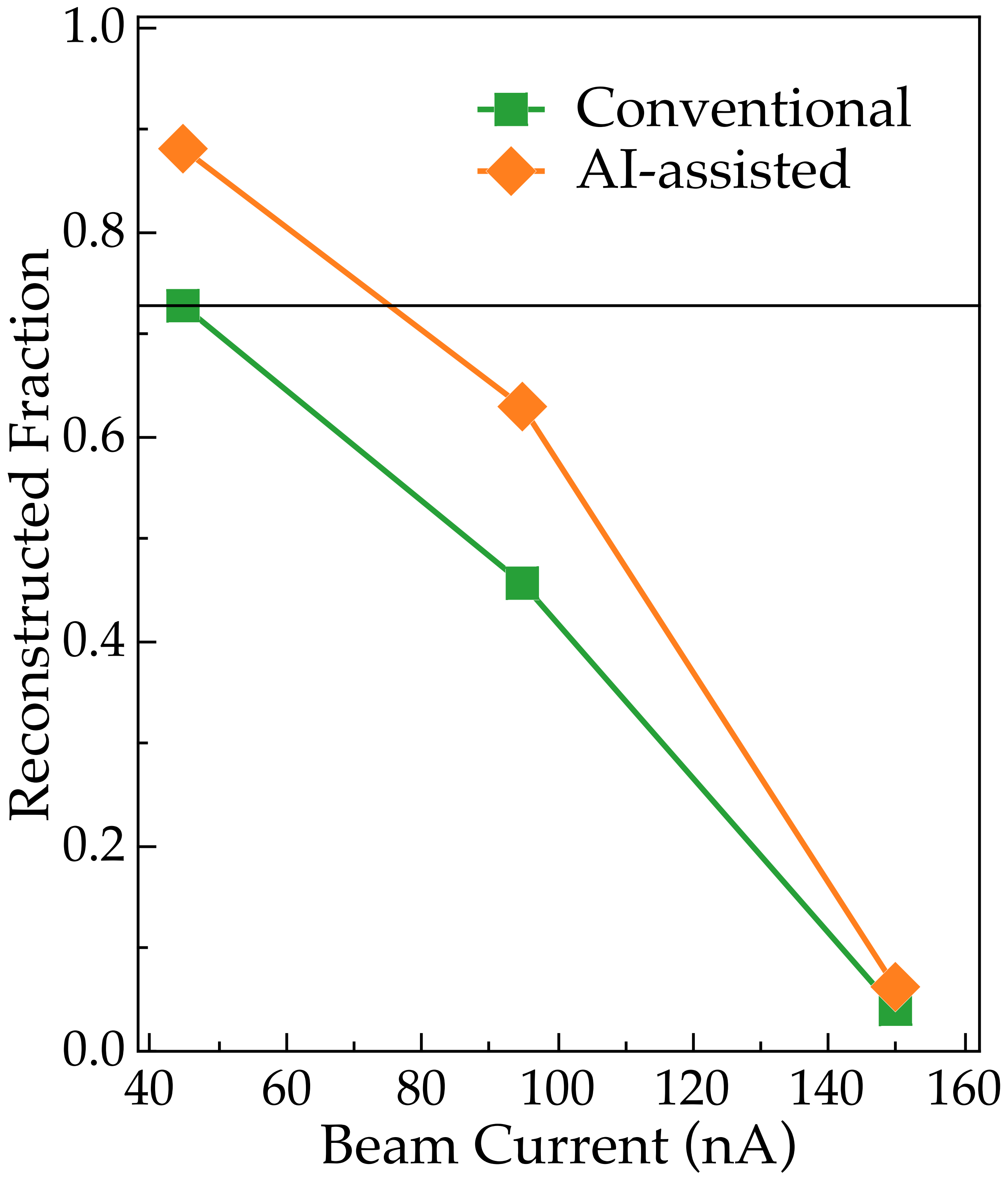


Up to ~35% gain in physics
Just using Classifiers

Moving to higher Luminosities

Performance of track identification for higher luminosity




- Pythia simulated physics reaction:

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

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

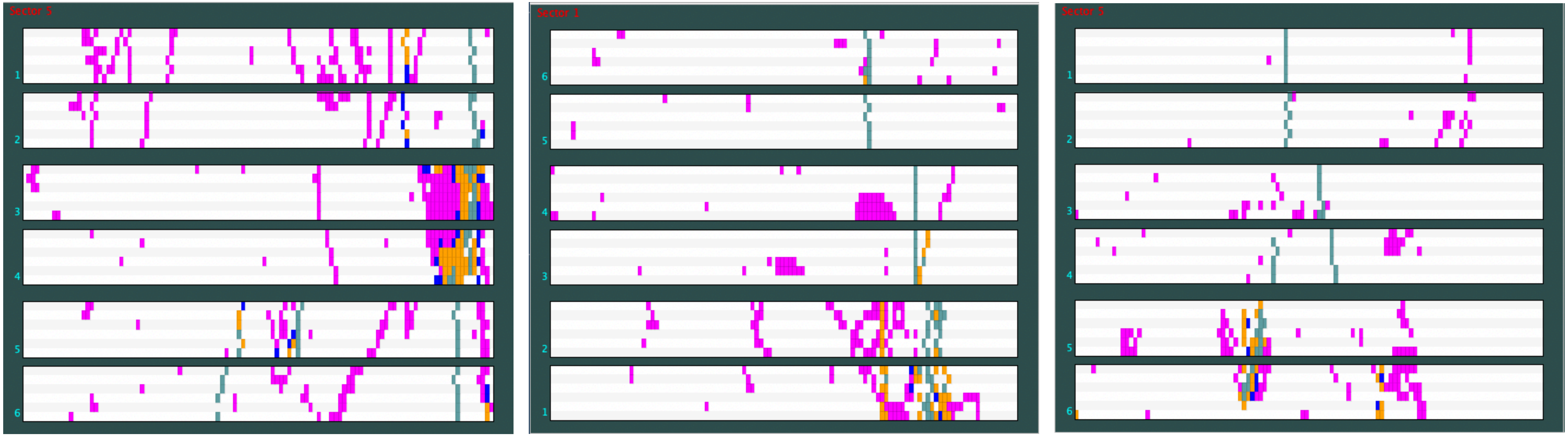
$$ep \rightarrow e'\pi^+\pi^-X$$

- 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.

# De-Noising

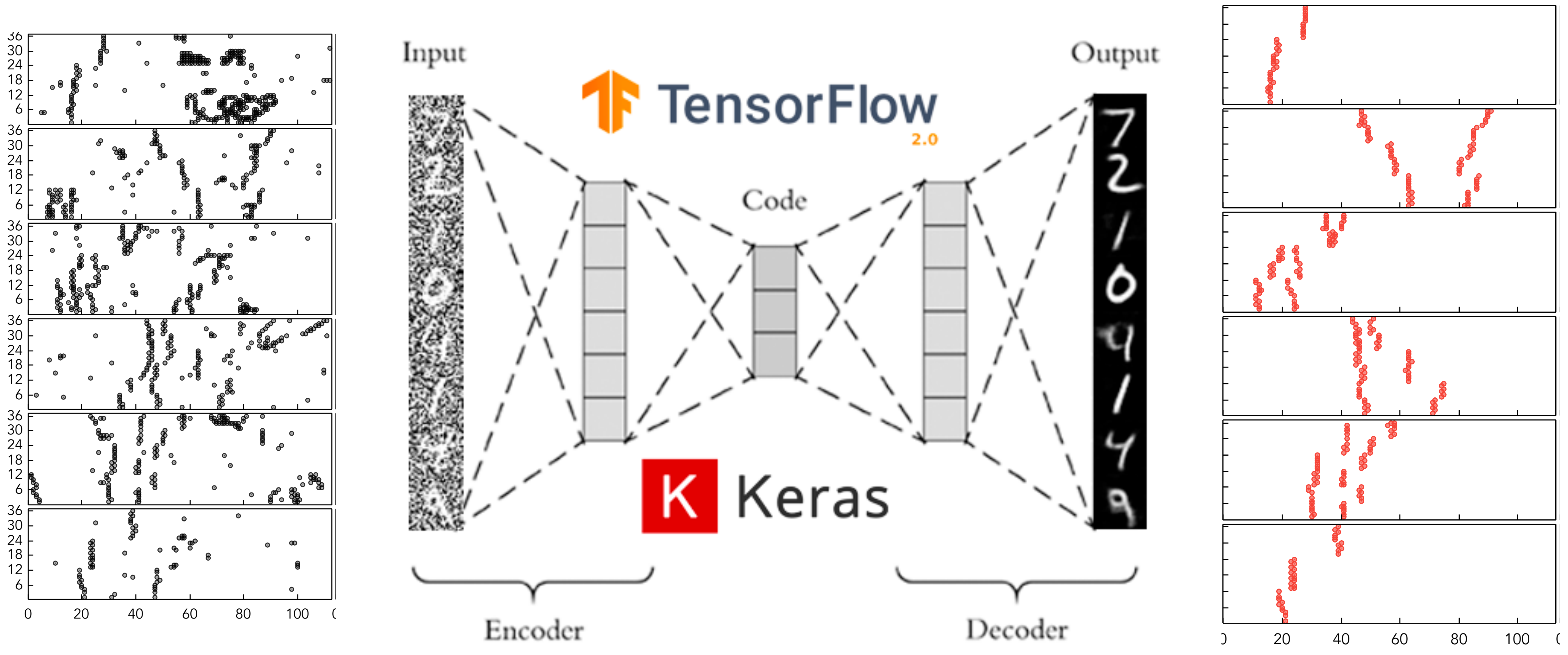
- In high luminosities the noise level increases and forming clusters (or segments in each chamber becomes challenging)
- This results in loss of clusters and AI-assited tracking can no longer help with combinatorics resolution

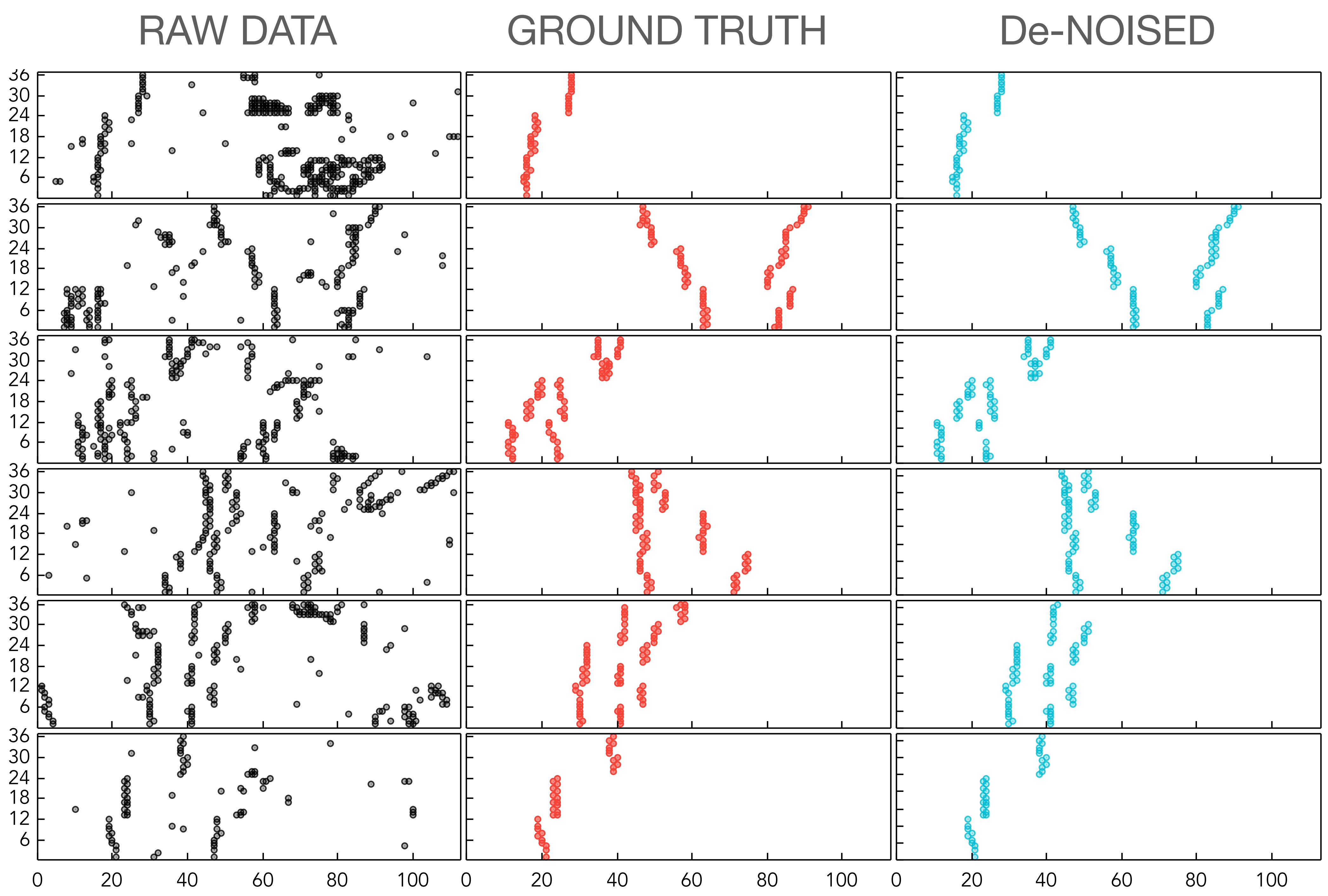
CLAS12 Event Display Examples (Drift Chambers)



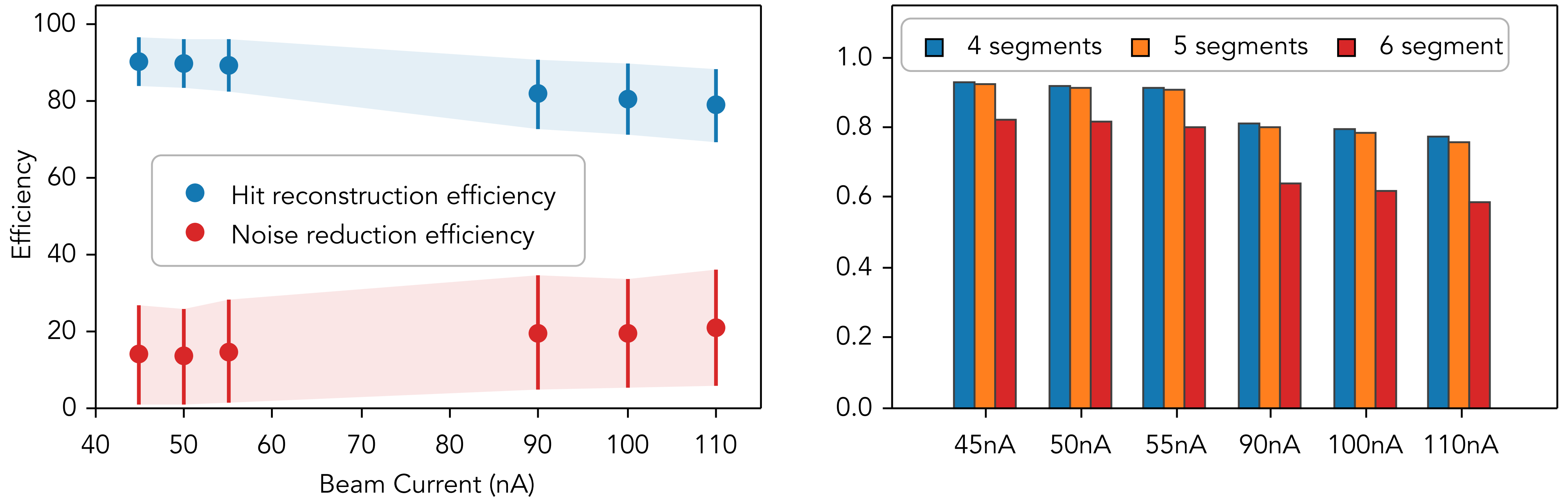
# De-Noising

- 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



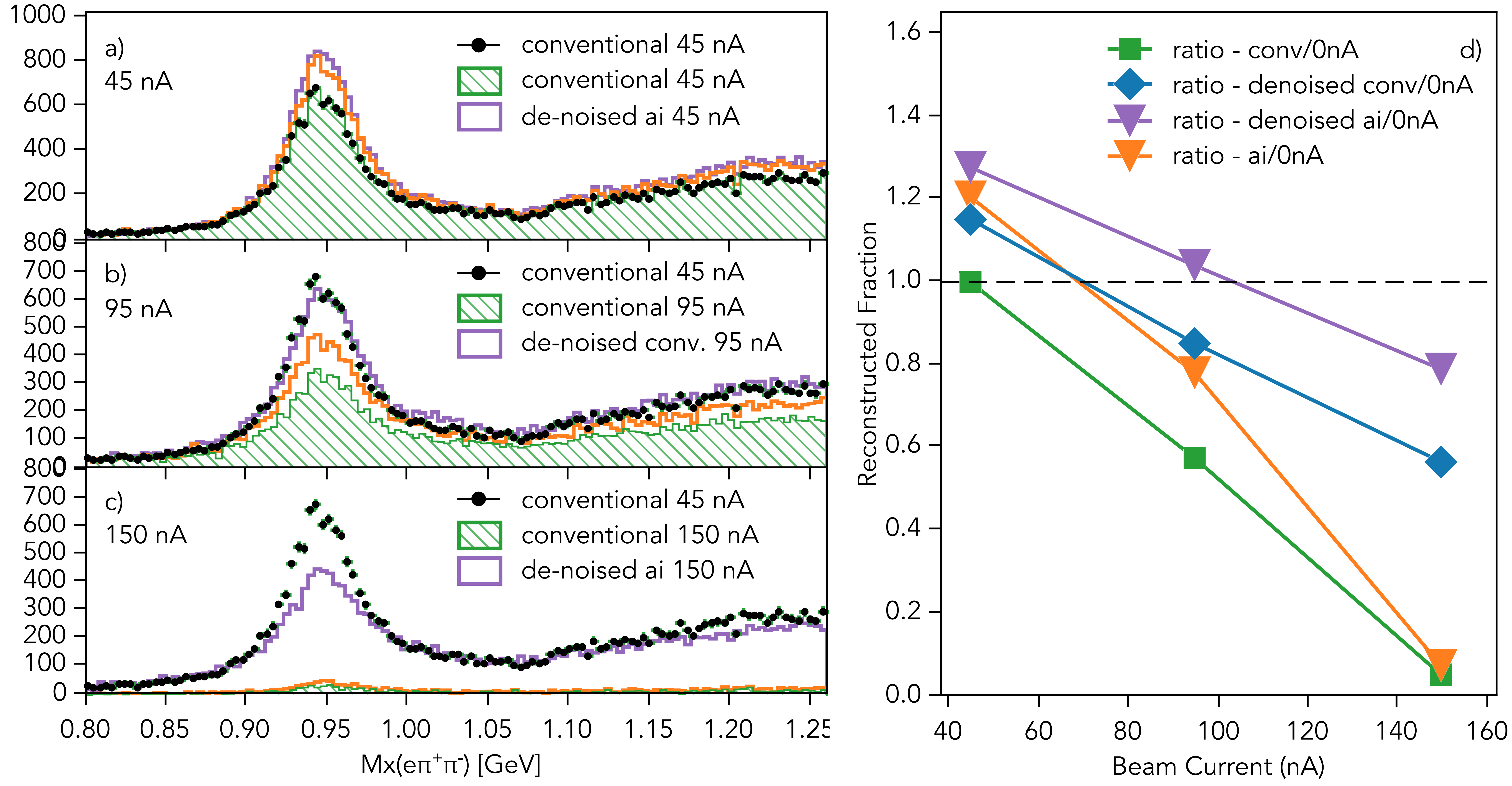




## Network Performance Summary



# 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

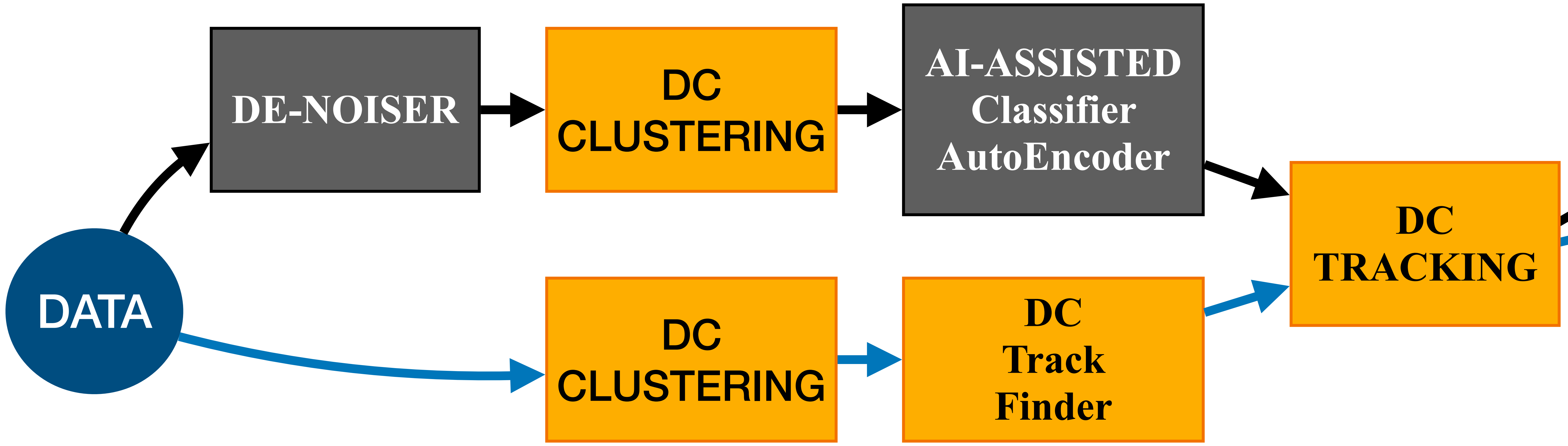


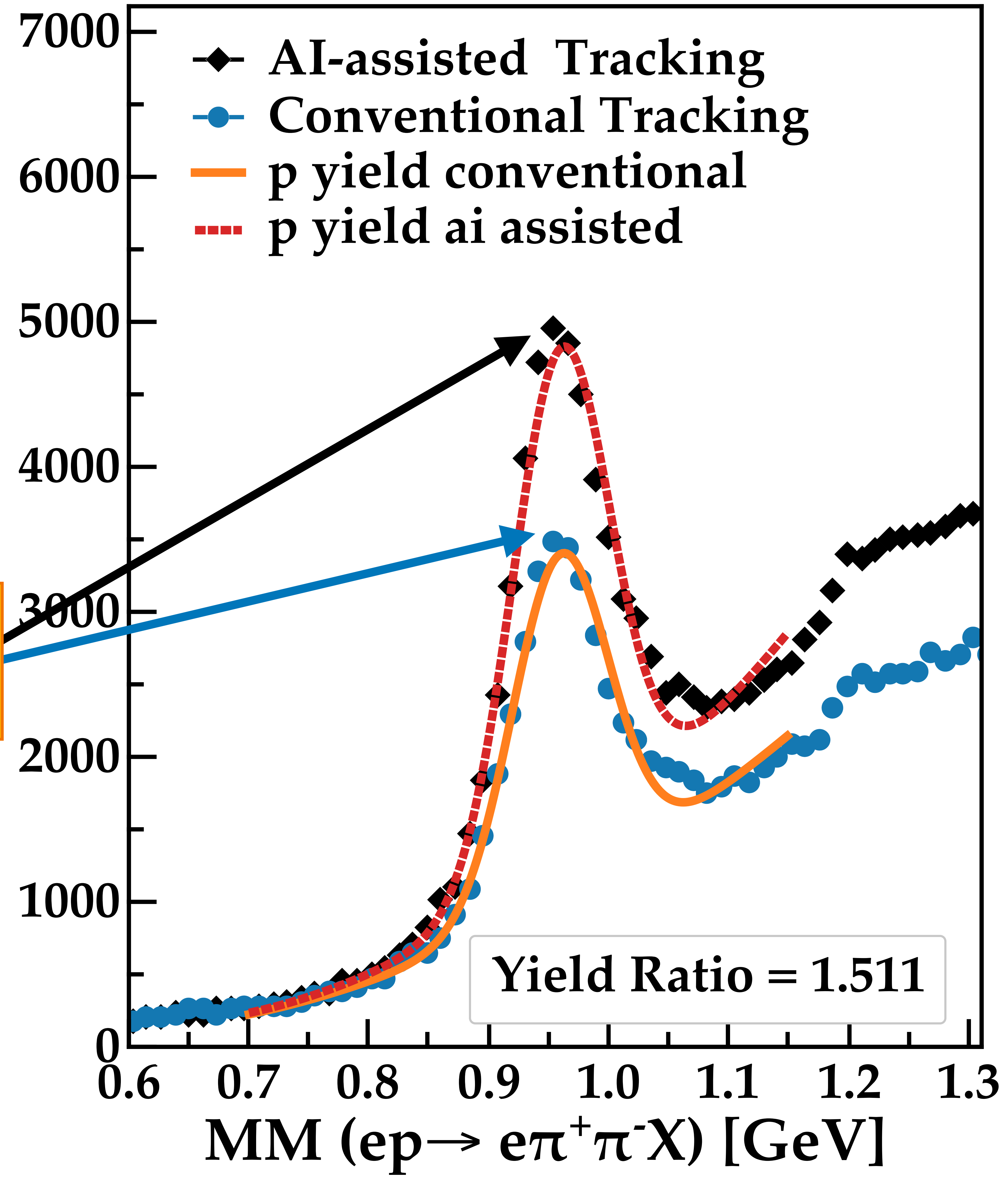

- At standard running luminosity, the de-noising slightly increases the yield compared to AI-assisted tracking.
- With increased luminosity, the de-noising helps to increase the yield significantly compared to conventional and AI-assisted tracking.
- Simulation underestimates the gain in yield significantly.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

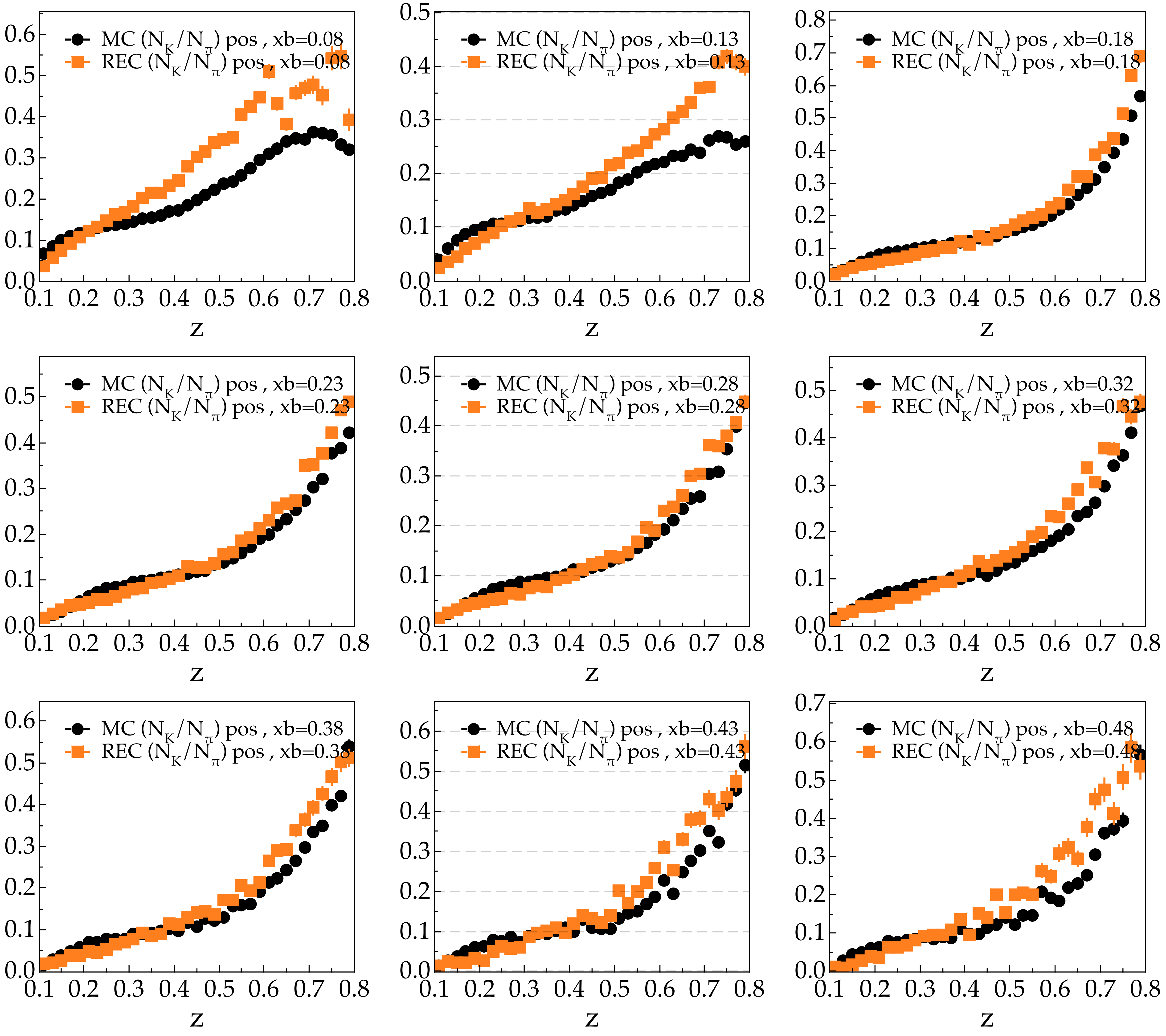


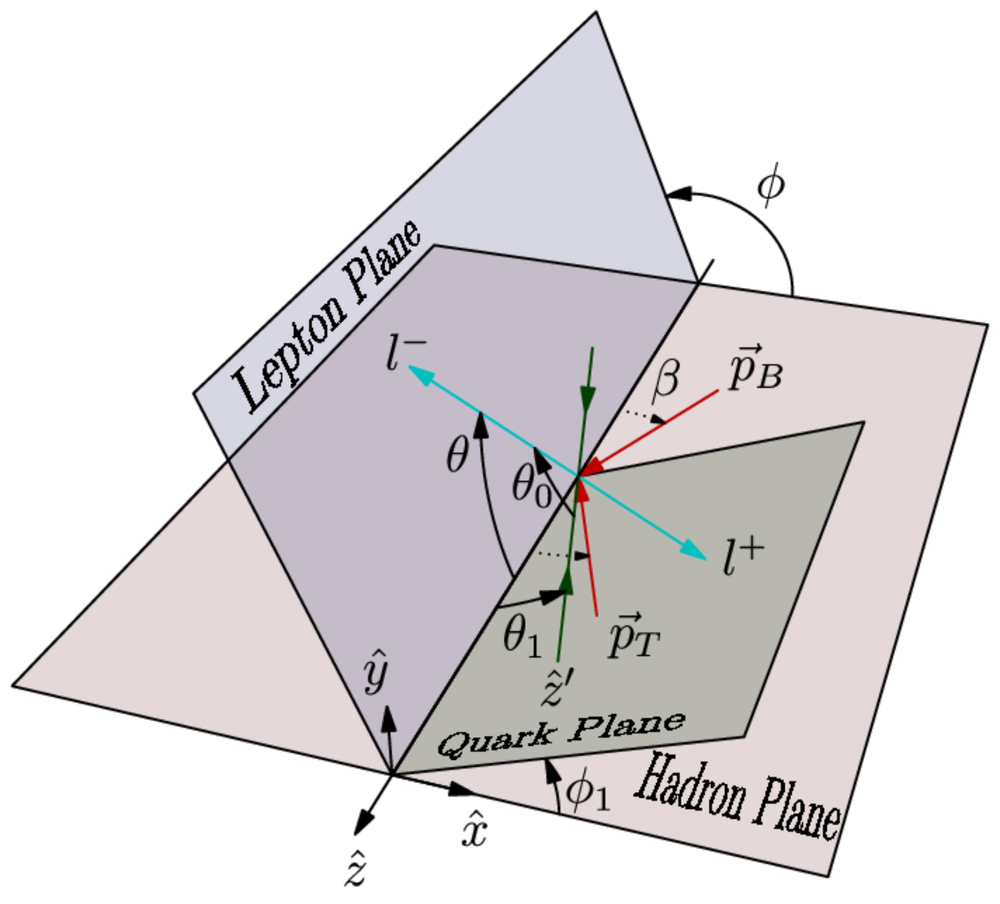




- Running at standard conditions (45 nA beam current) the AI increased the yield of missing protons by 51%.
- The improvement in yield is reaction and kinematics dependent, and for some event topologies reaches even 83% (J/psi with 3 particles detected final state).

# Particle Identification

# Physics Analysis/Particle Identification (TMDs)



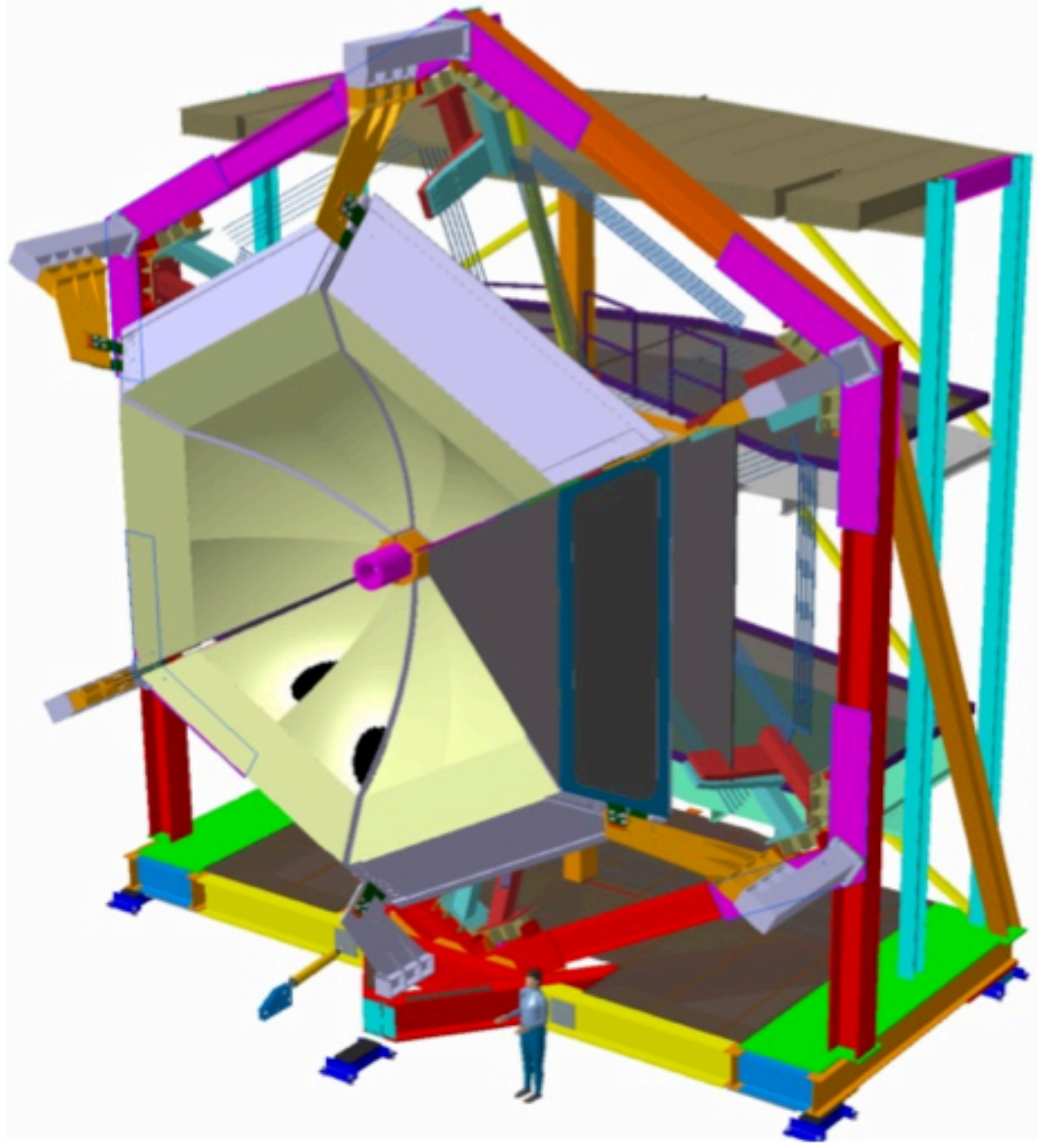

| | U | L | T |
|---|---|---|---|
| **Quarks** | $\gamma^+$ | $\gamma^+\gamma^5$ | $i\sigma^{i+}\gamma^5$ |
| U | $\boldsymbol{f_1}$ | | $h_1^\perp$ |
| L | | $\boldsymbol{g_1}$ | $h_{1L}^\perp$ |
| T | $f_{1T}^\perp$ | $g_{1T}$ | $\boldsymbol{h_1}$, $h_{1T}^\perp$ |
| LL | $\boldsymbol{f_{1LL}}$ | | $h_{1LL}^\perp$ |
| LT | $f_{1LT}$ | $g_{1LT}$ | $h_{1LT}$, $h_{1LT}^\perp$ |
| TT | $f_{1TT}$ | $g_{1TT}$ | $h_{1TT}$, $h_{1TT}^\perp$ |

- Traditional (time-of-flight) can effectively separate pi/K up to 3.5 GeV
- For full measurement of hadron multiplicities as a function of z and Pt need to separate hadrons at higher momenta to measure:
  - Hadron multiplicities
  - Single Spin Asymmetries (SSA)
  - Double Spin Asymmetries
- Map fragmentation functions:

$$D^{q\to K}(z, P_T), D^{q\to\pi}(z, P_T)$$

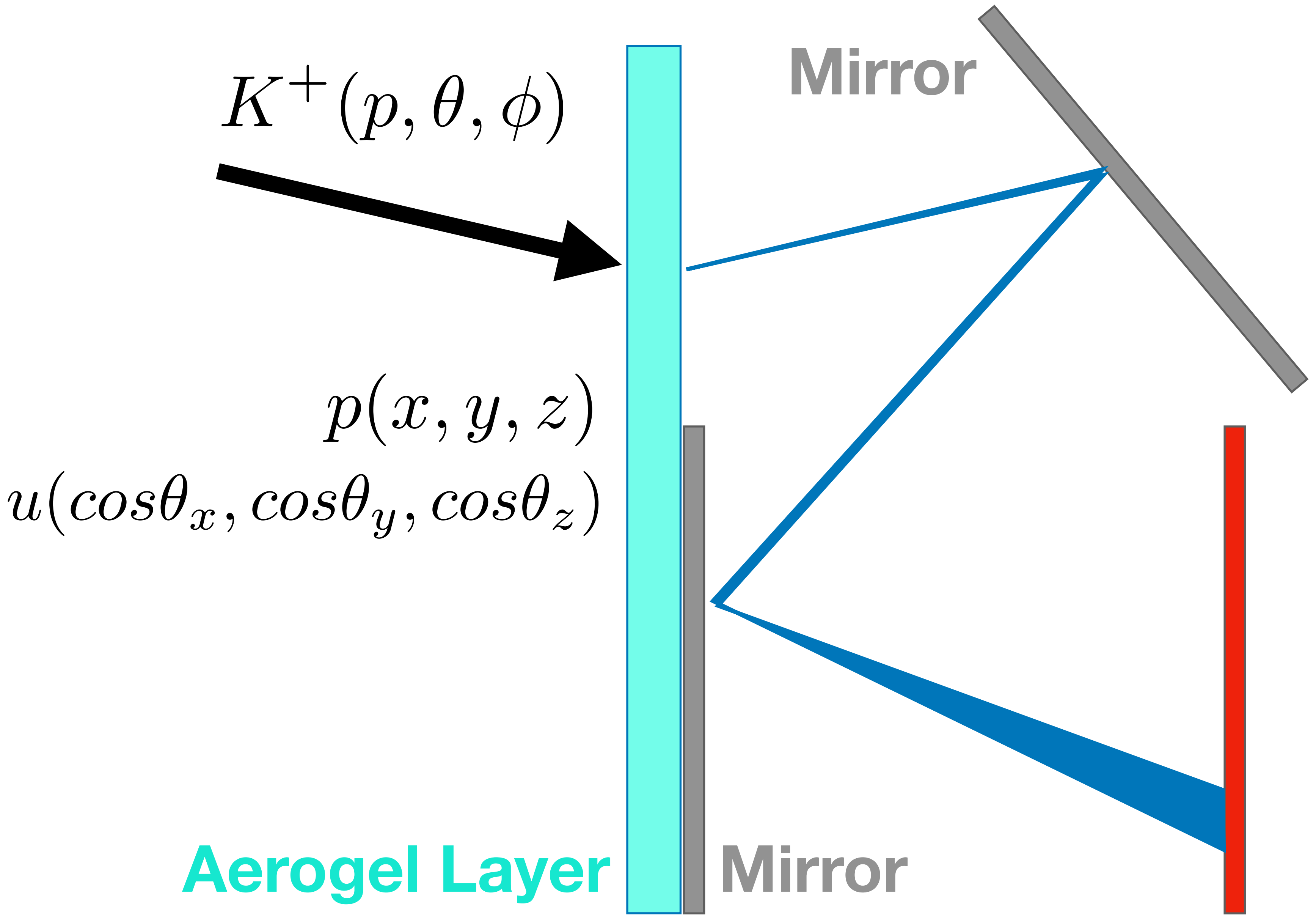
# *CLAS12* – RICH

- The Ring Imaging Cherenkov detector (RICH) is designed to improve CLAS12 particle identification in the momentum range 3-8 GeV/c and will replace one sector of the existing LTCC detector.
- The RICH design incorporates aerogel radiators, visible light photon detectors, and a focusing mirror system, which will be used to reduce the detection area instrumented by photon detectors to ~1 $m^2$ . Multi-anode photomultiplier tubes (MA-PMTs) provide the required spatial resolution and match the aerogel Cherenkov light spectrum (visible and near-ultraviolet region).
- For forward scattered particles ($\theta < 13^o$) with momenta 3 - 8 GeV/c, a proximity imaging method with thin (2 cm) aerogel and direct Cherenkov light detection will be used.
- For larger incident particle angles of $13^o < \theta < 25^o$ and momenta of 3 - 6 GeV/c, the Cherenkov light will be produced by a thicker aerogel (6 cm), focused by a spherical mirror, undergo two further passes through the thin radiator material and a reflection from planar mirrors before detection.

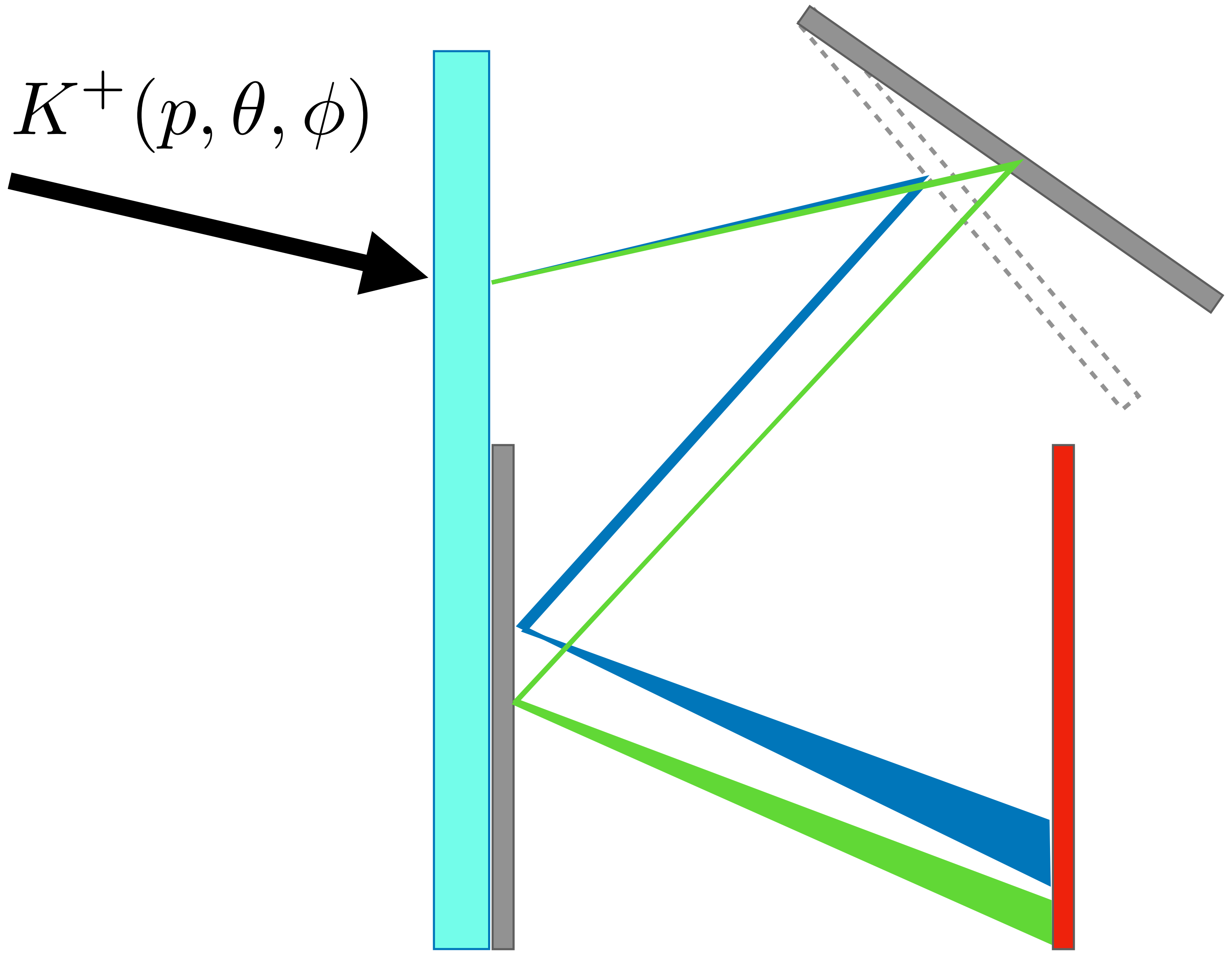


# Rich Detector (preliminary)

- RICH Ideal Geometry
  - If the ideal geometry and position of mirrors are known the ray-tracing can help recover the Cherenkov angle
  - Calculating the Cherenkov angle for each of the hits on the photomultiplier plane allows to identify the particles.

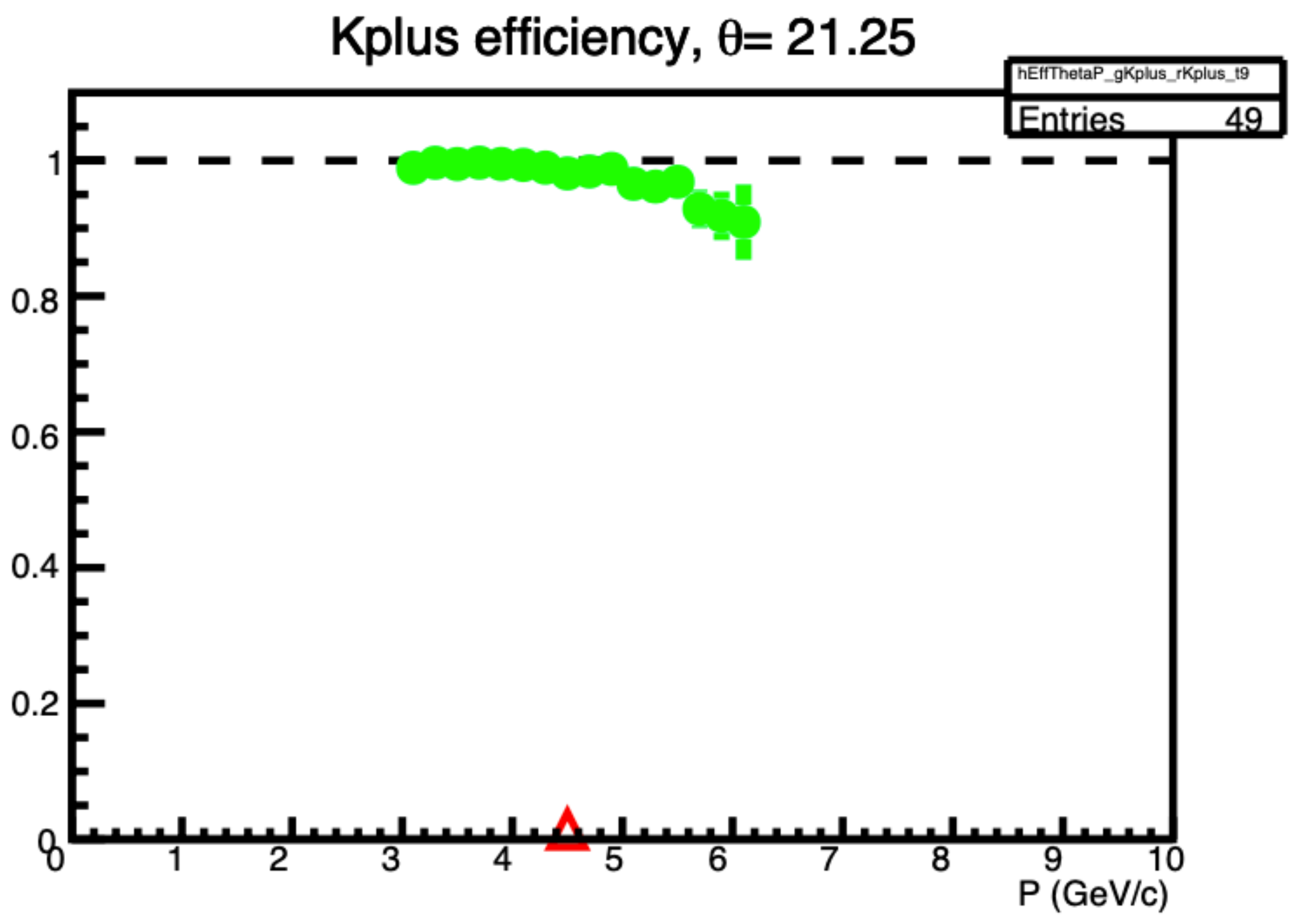


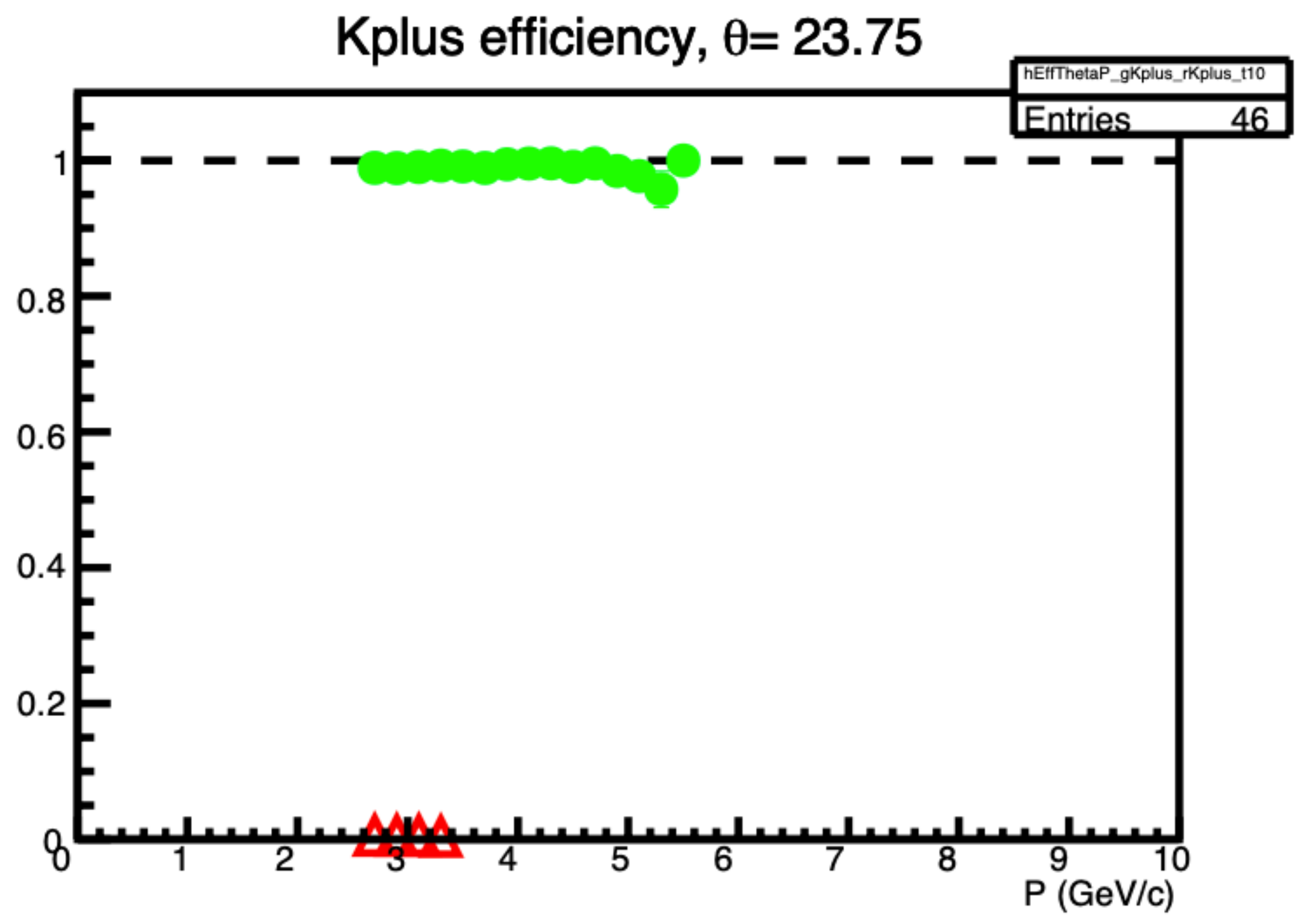

- RICH Real World Geometry
  - Ray tracing will predict an inaccurate position for the hit on the detector plane
  - This affects the efficiency of particle identification



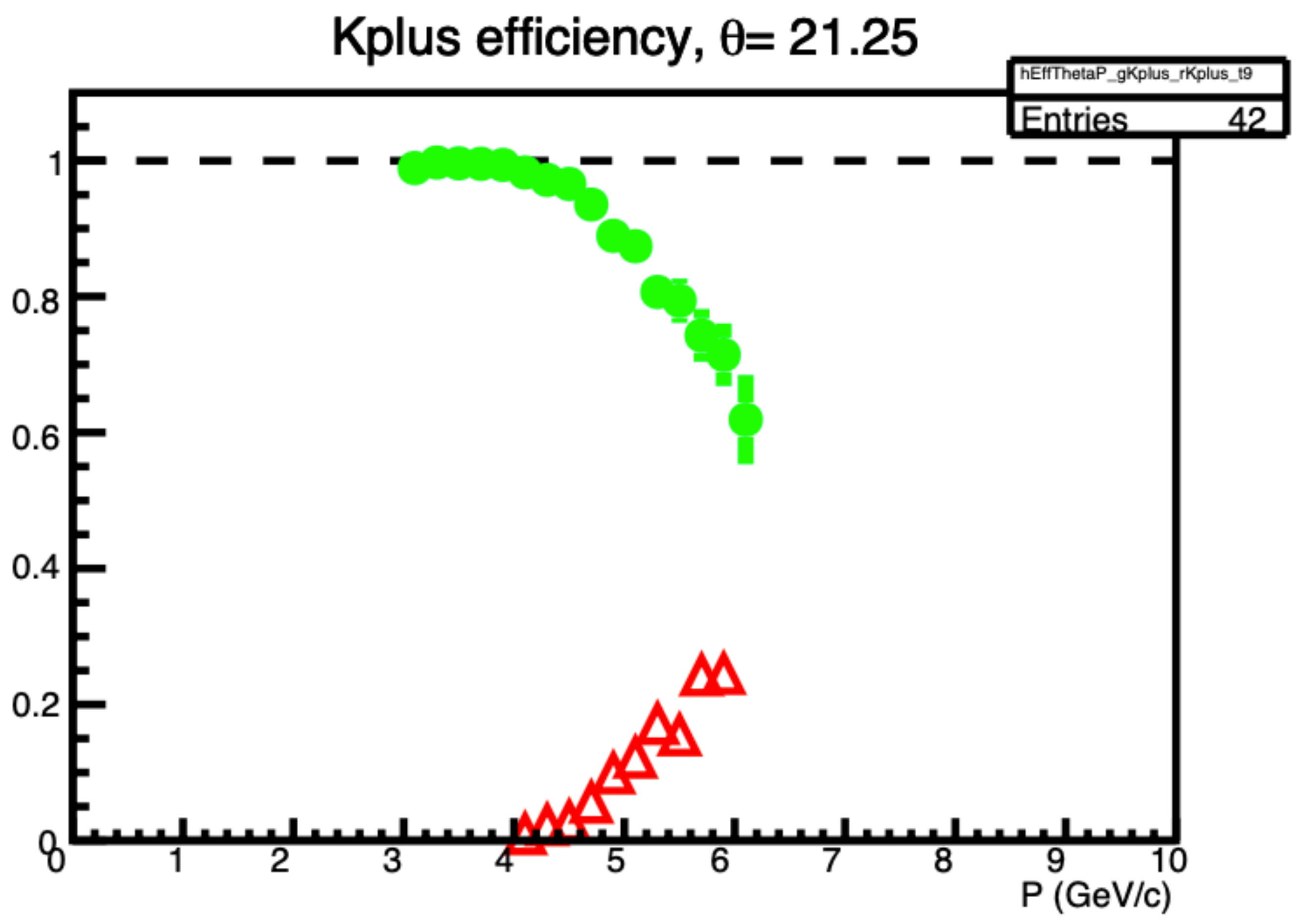

- Neural Networks can be trained on Real-World data which includes miss-alignments
- It can learn the Cherenkov ring patterns for incident particles, given interaction point and direction at crossing the aerogel layer

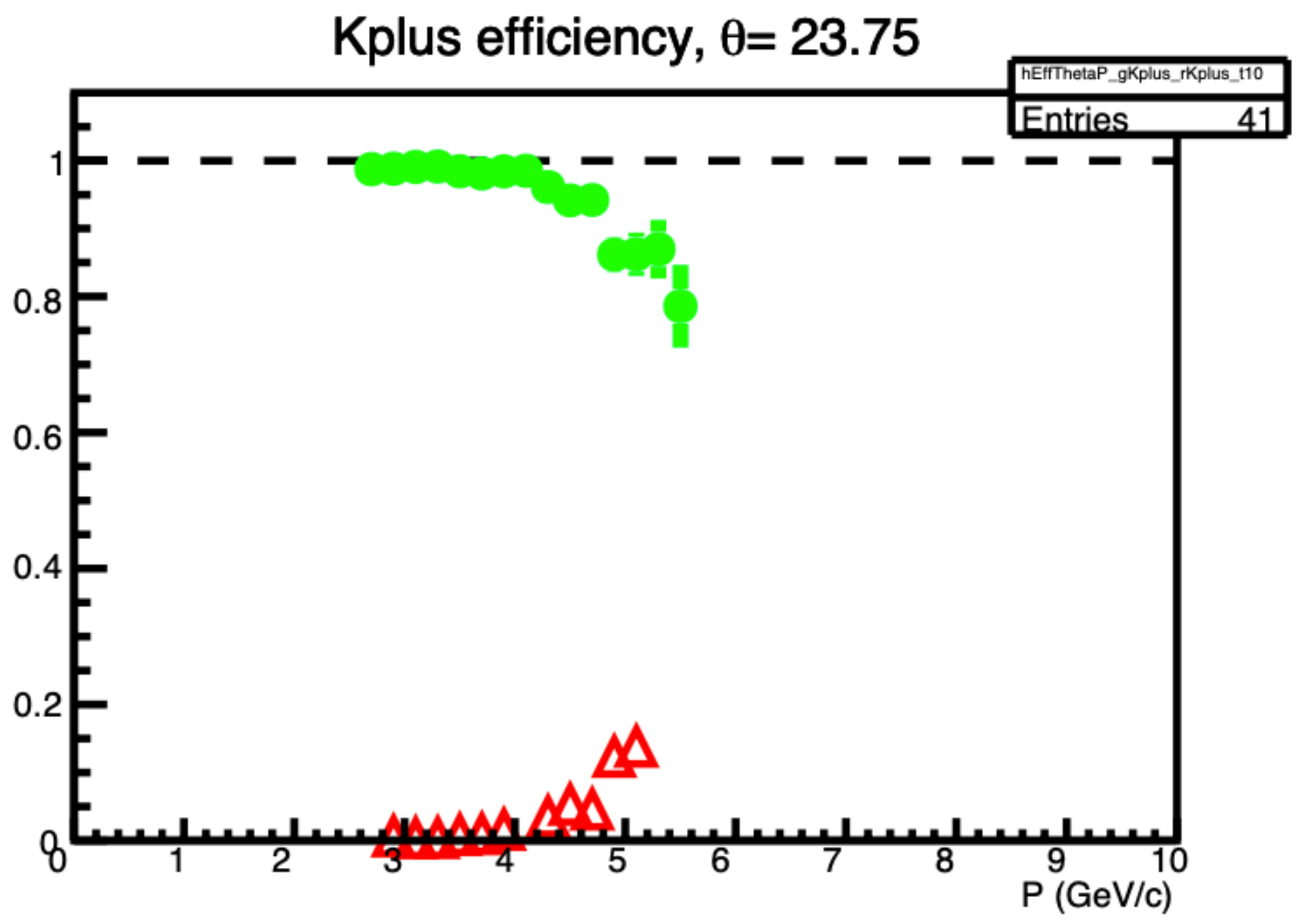
## Kaon Identification Efficiency (IDEAL GEOMETRY)

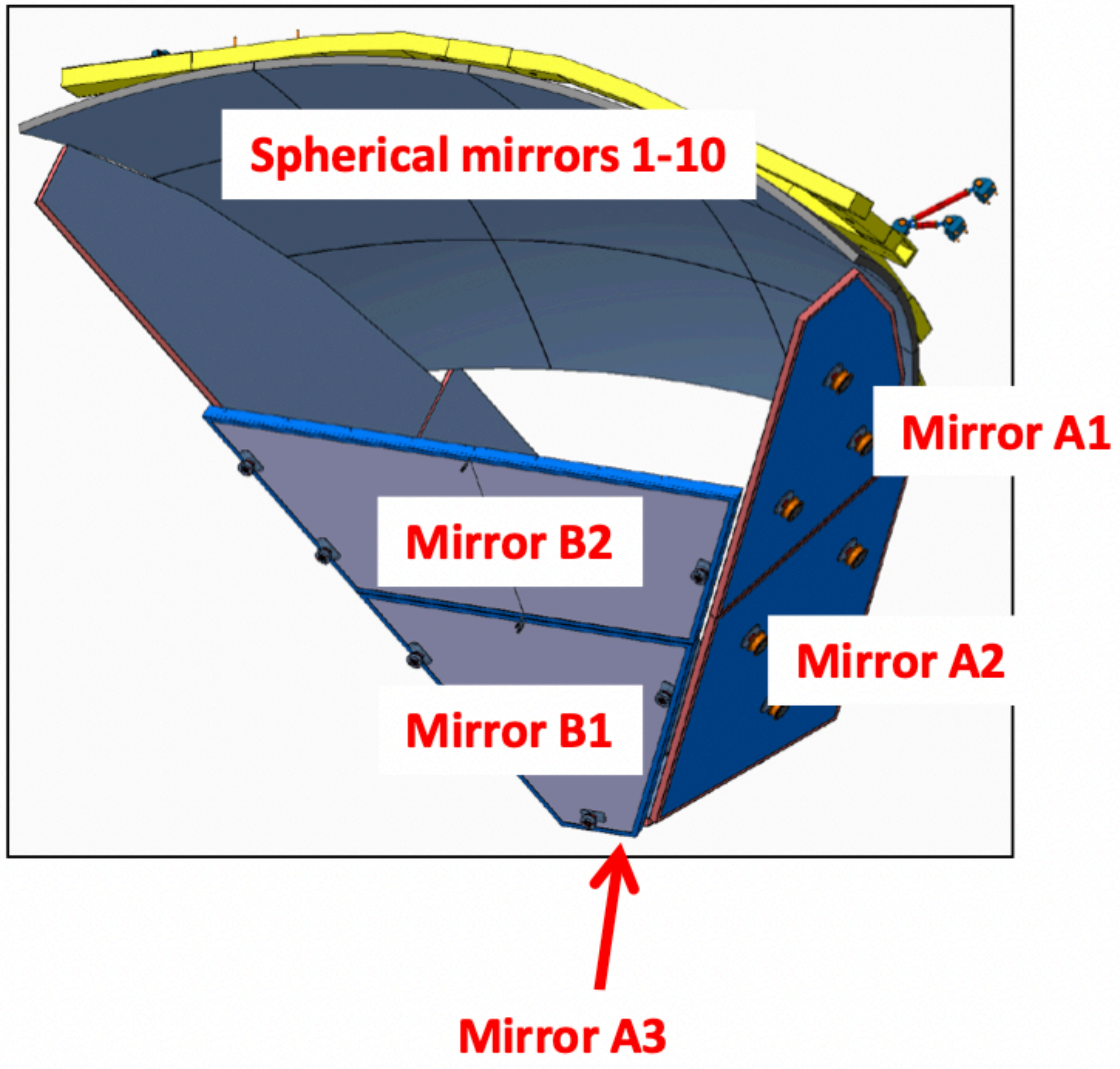


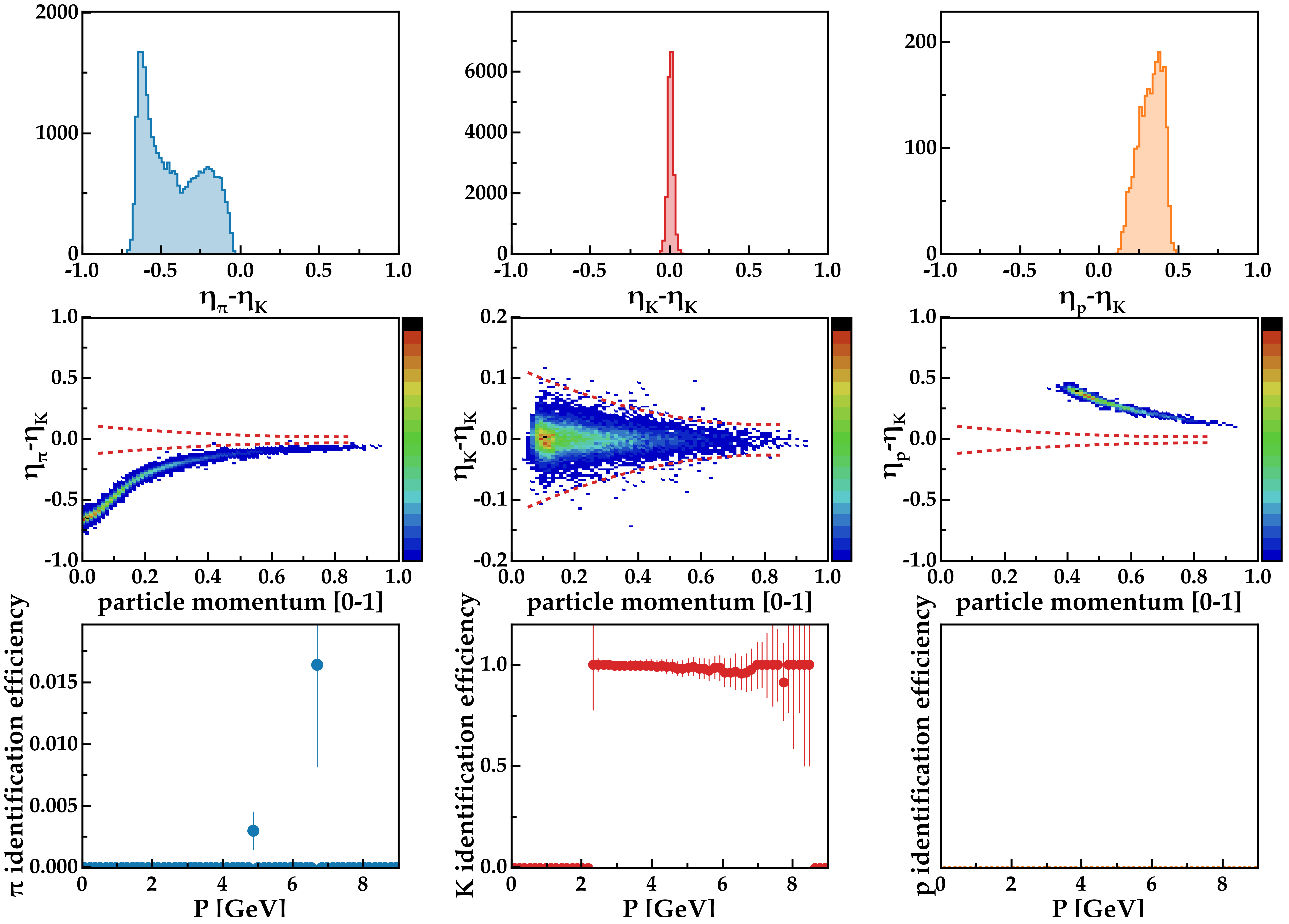

## Kaon Identification Efficiency (MIS-ALLIGNED GEOMETRY)



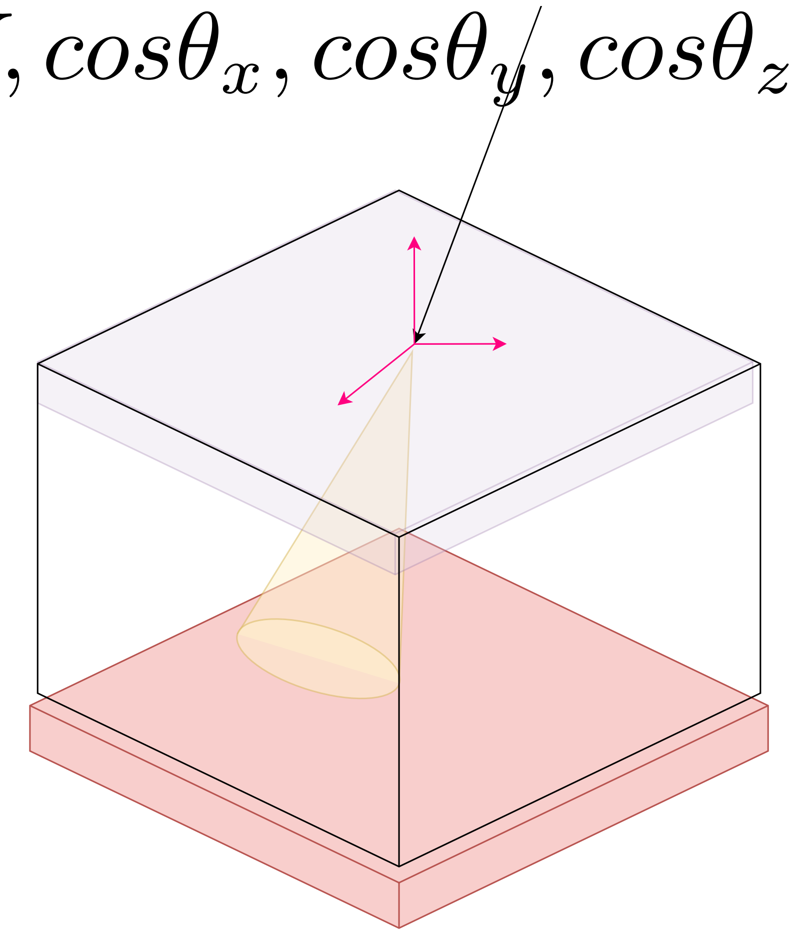
# Rich Detector (particle identification)




- Neural Network predicts Cherenkov angle for incoming particles based on the hits on the RICH photo-multipliers
- Kaon efficiency is uniform across the momentum range
- The Network is trained on misaligned data
- Kaon efficiency is calculated from misaligned data
- The detector will not need to be aligned when trained on experimental data.

**Input:**

$$X[P, X, Y, cos\theta_x, cos\theta_y, cos\theta_z, X_h, Yh]$$



**Output:** $\eta$

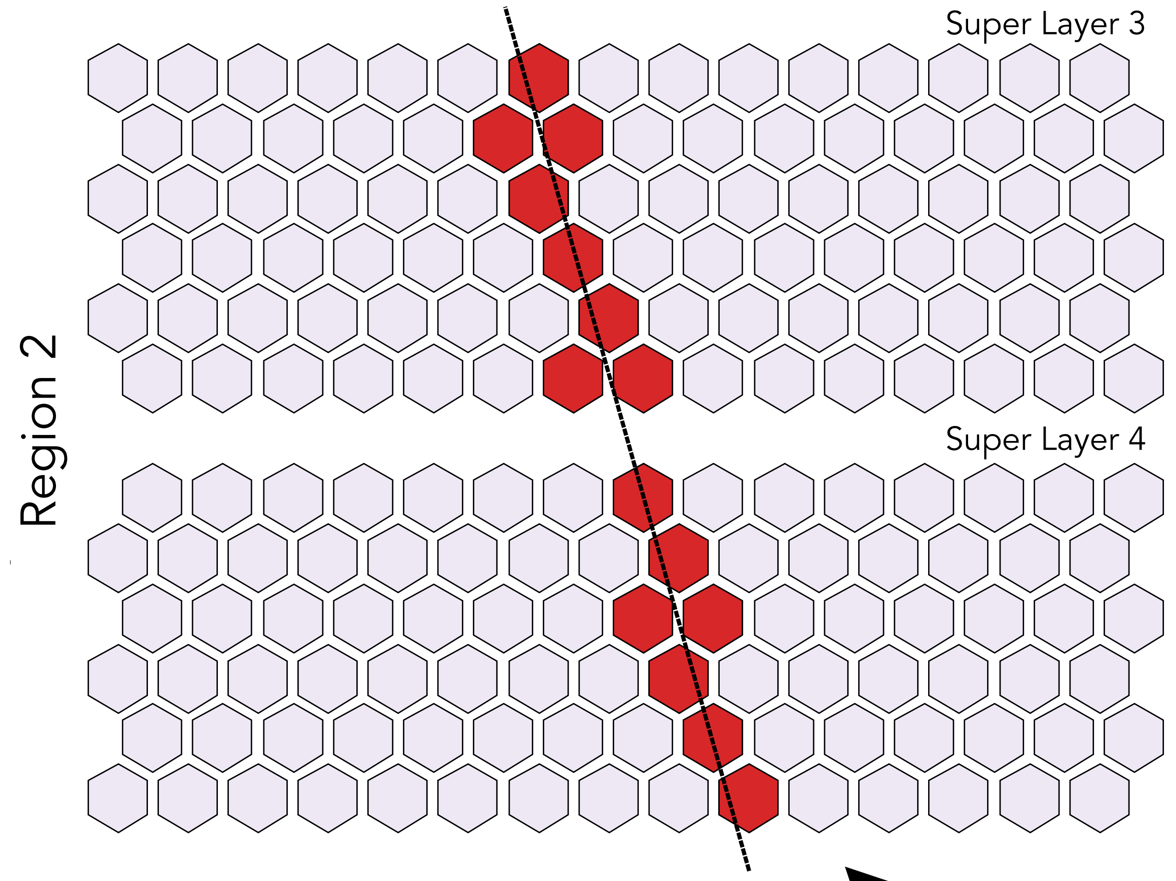
# Summary



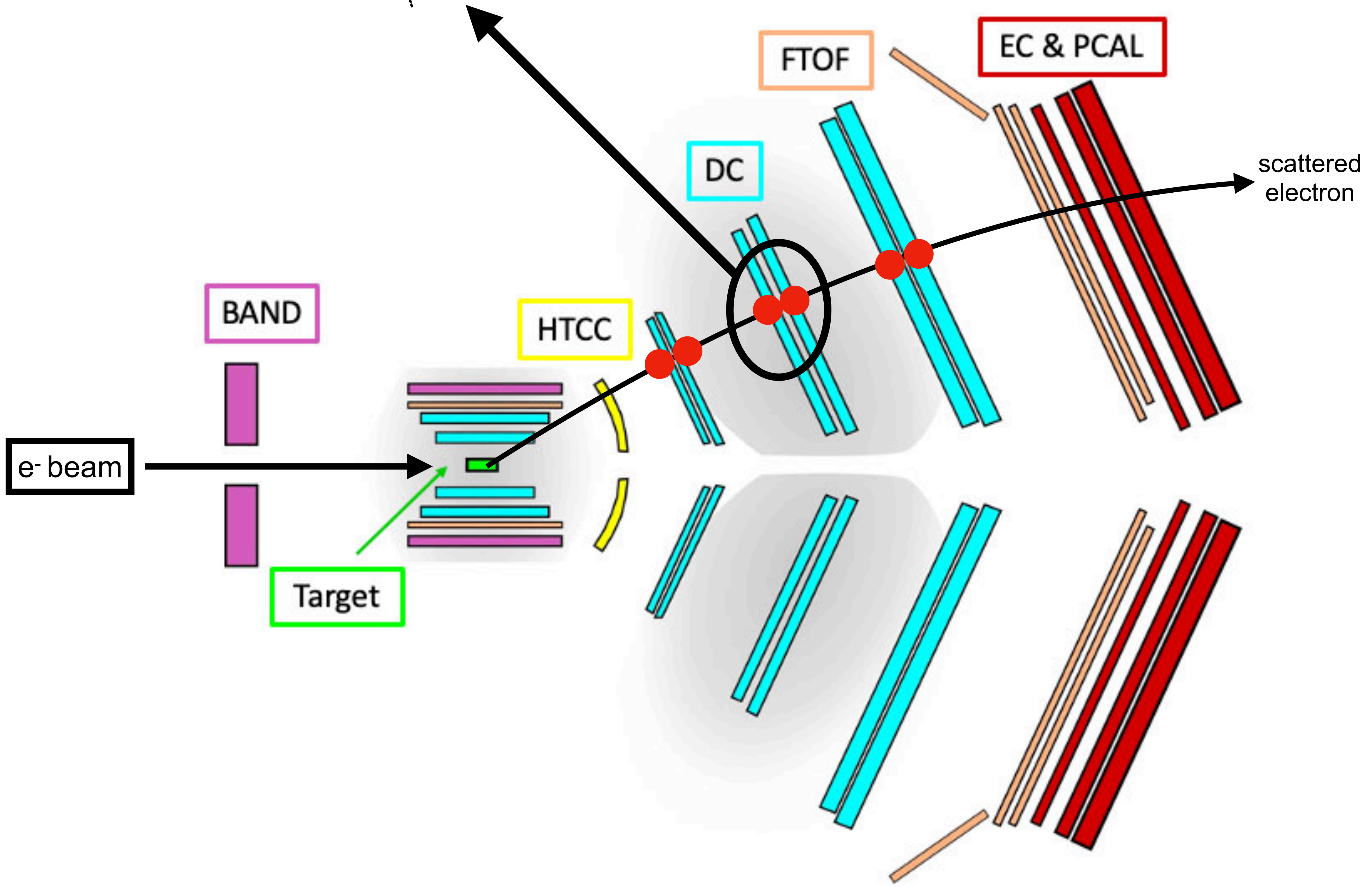
- AI/ML tracking provides significant improvements in physics yield for completed experiments in CLAS12.
- Using tracking AI/ML experiments can run at higher luminosities and collect more data in the allocated time.
- The AI/ML will be an essential component for Jlab 24 GeV upgrade (hard to imagine any experiment without utilizing machine learning), for tracking and particle identification.
- The developed particle identification methods will extend our kinematic range in physics observables.
- There are many more AI/ML projects in CLAS12 (not mentioned in this talk), such as:
  - Online track reconstruction (allows online physics monitoring)
  - Level-3 trigger (improves purity from 25% to 93%)
  - Neutral pion identification in Electromagnetic Calorimeter
  - Particle Identification (using signals from all detectors)
  - Central Detector Track classification (will improve tracking efficiency of central detector)

# Backup Slides

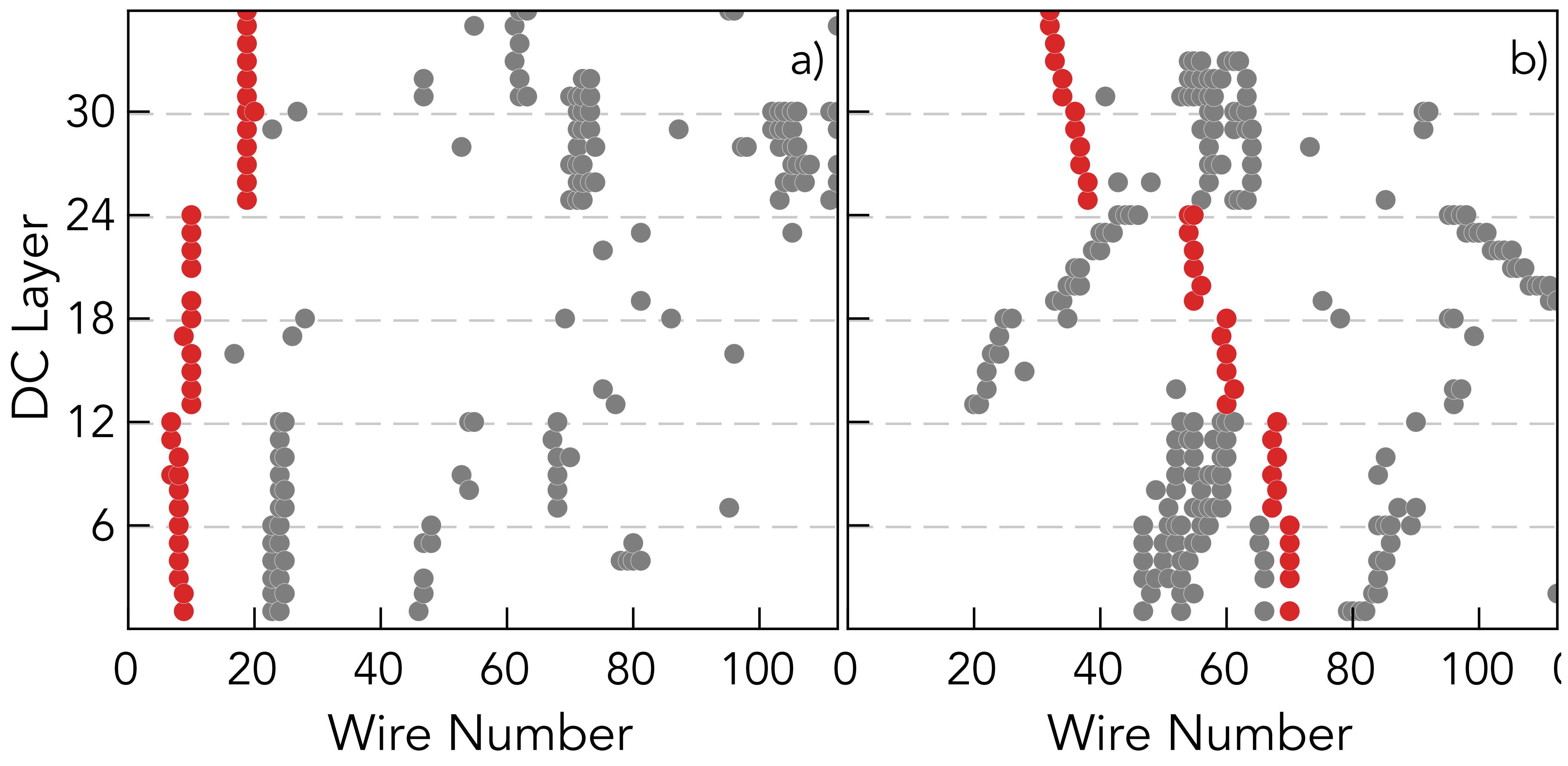
# CLAS12 Detector




- 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

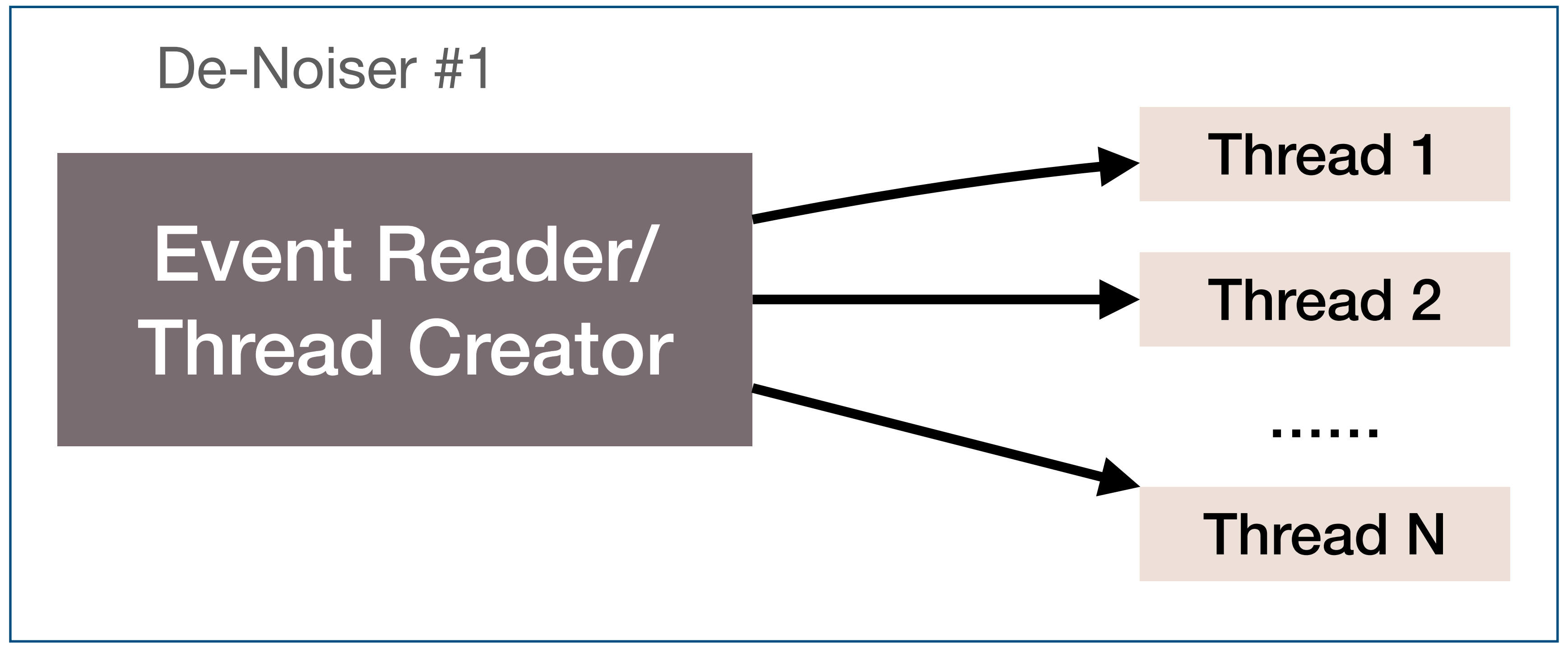


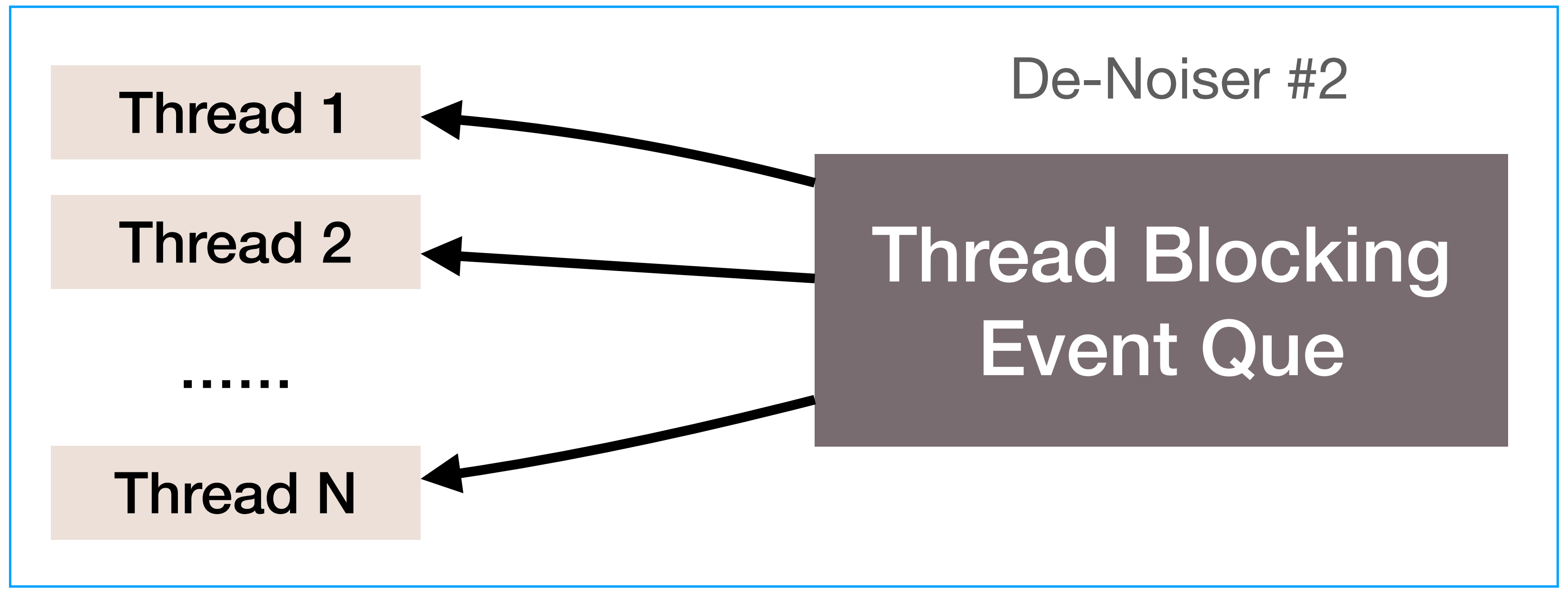

- Charged particle tracking is computationally extensive (about 80% of data processing time)
- The multi-particle final states produce numerous clusters in each sector which have to be analyzed to find the right combinations of clusters that form a track
- Identifying correct cluster combinations can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

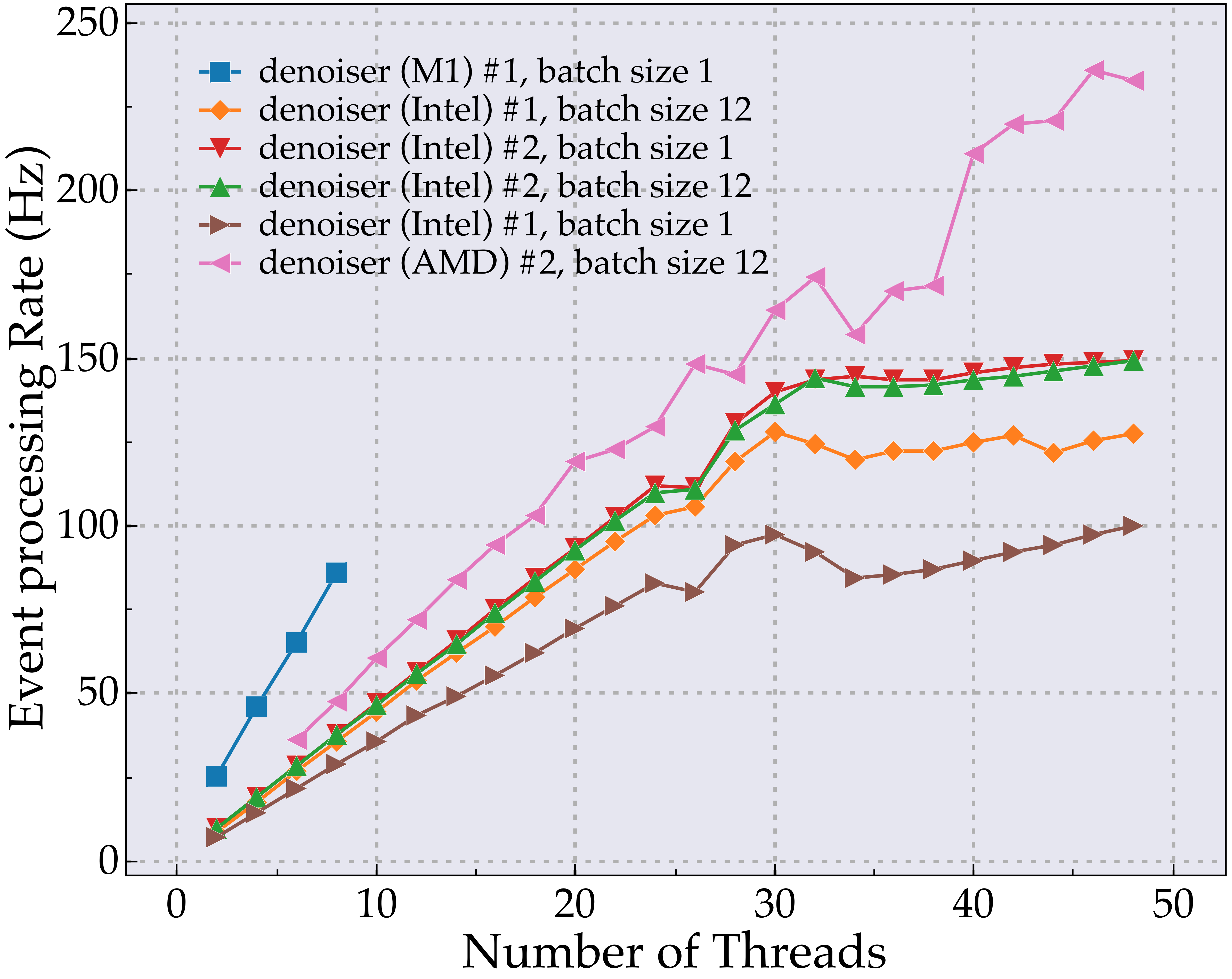


# De-nosing Performance Multi-Threaded

- **C++:** Keras model inference in C++ code implemented for CLAS12 de-noiser.
- **Multi-Threading**: Multi-threading implemented to process data files (using std::thread)

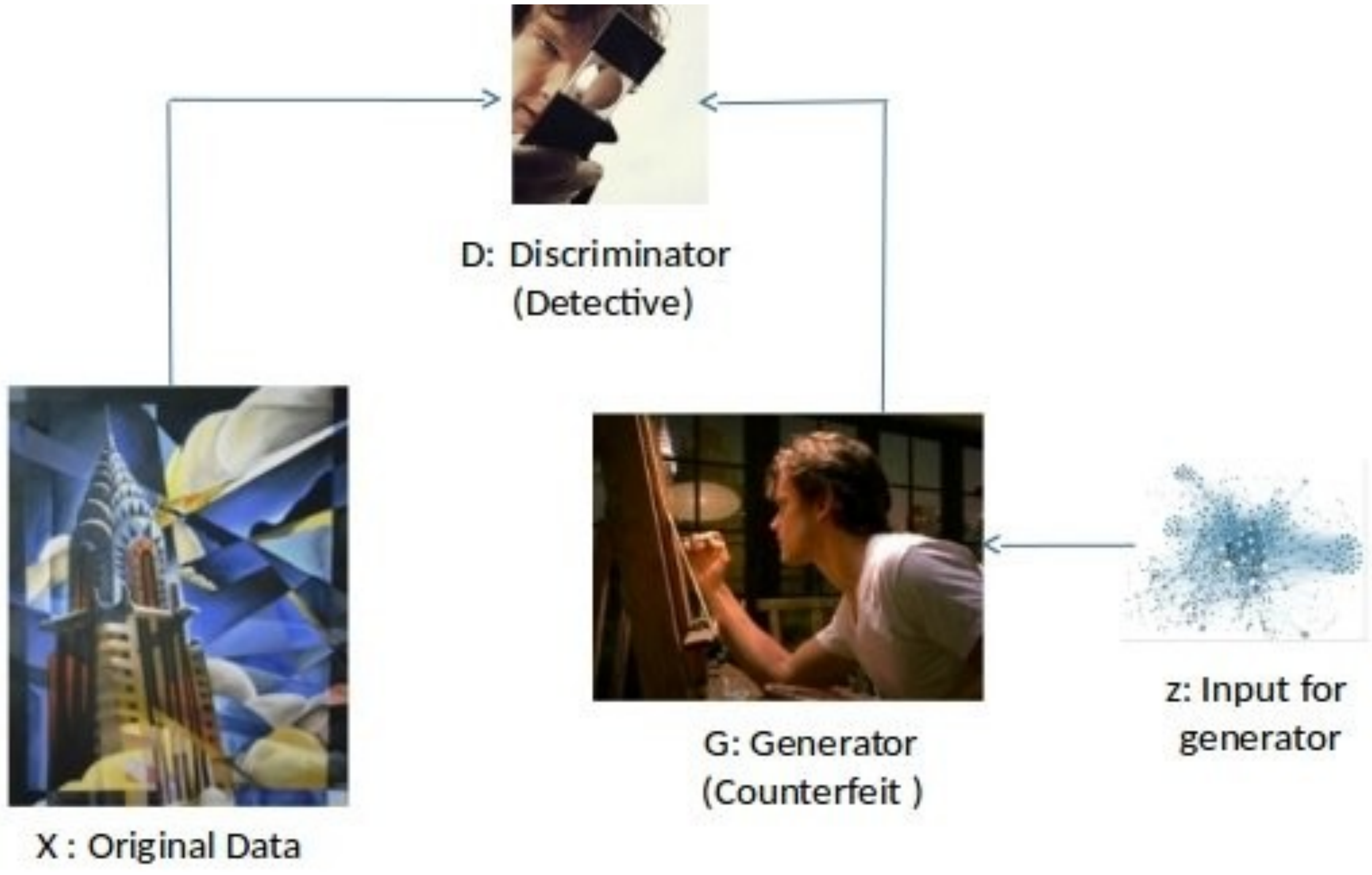


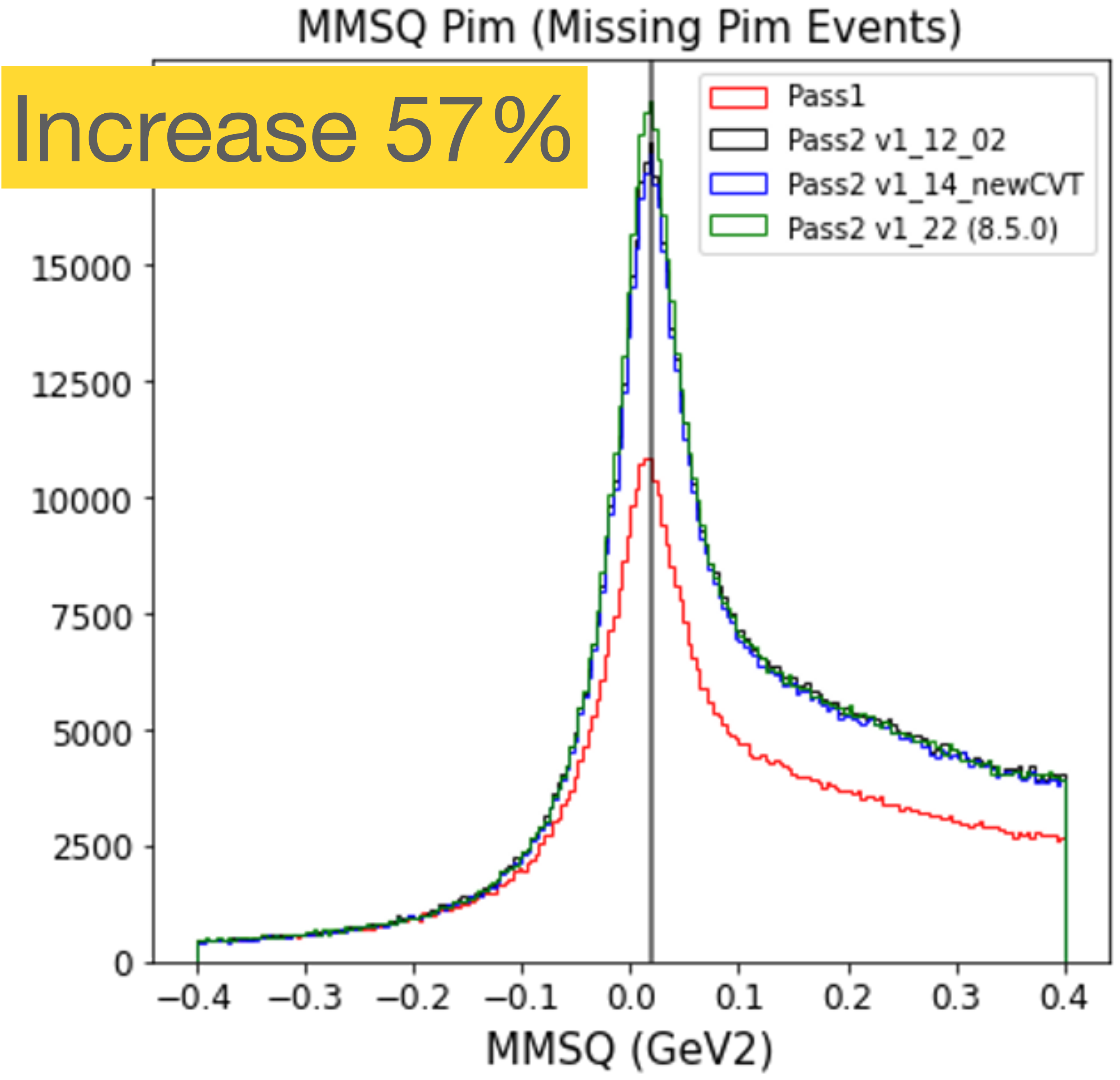
DALL.E
Open AI

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

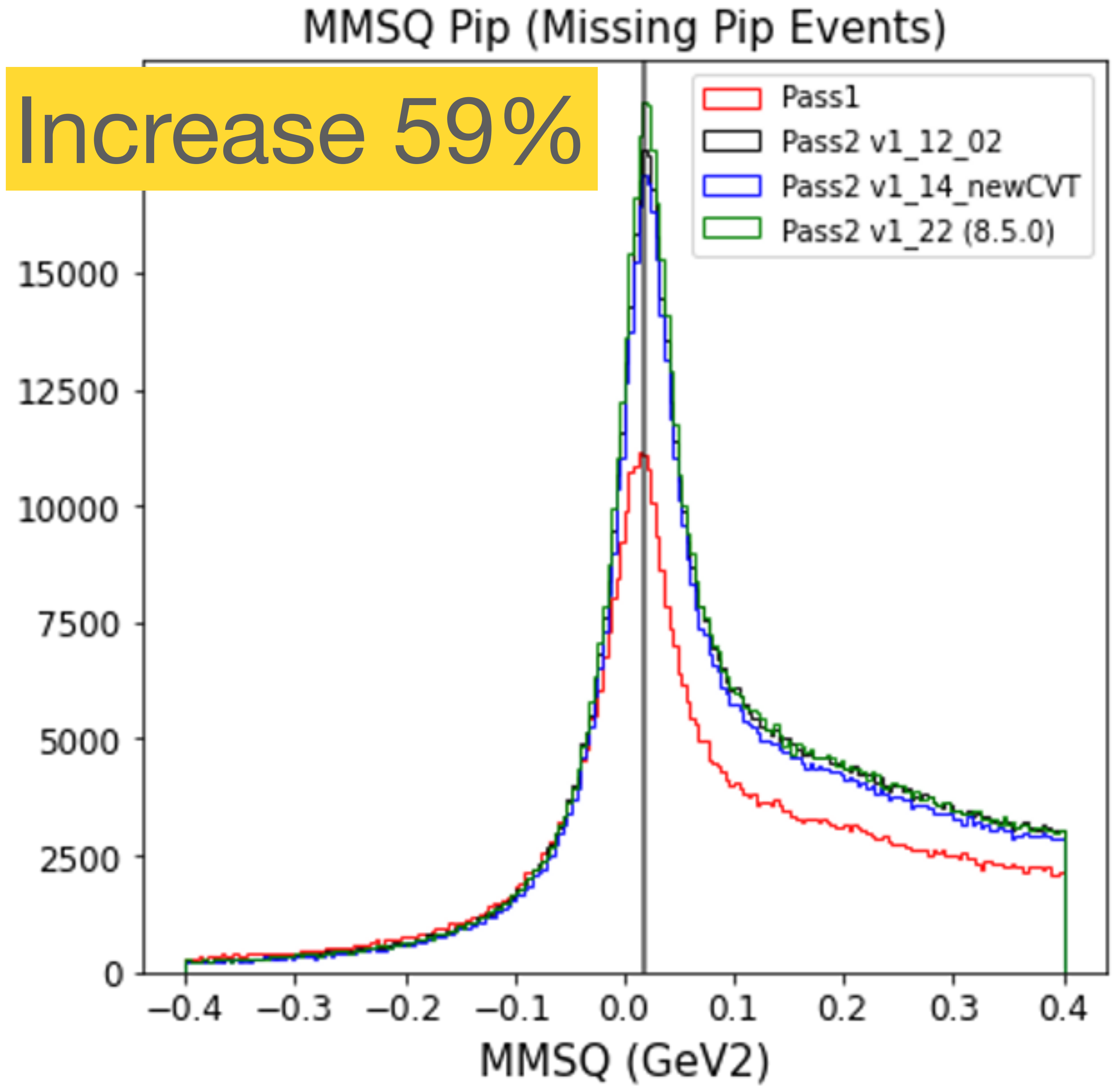



## RUN GROUP-A Pass2 Validation Cooking
## Includes De-Nosing and AI-assisted Tracking

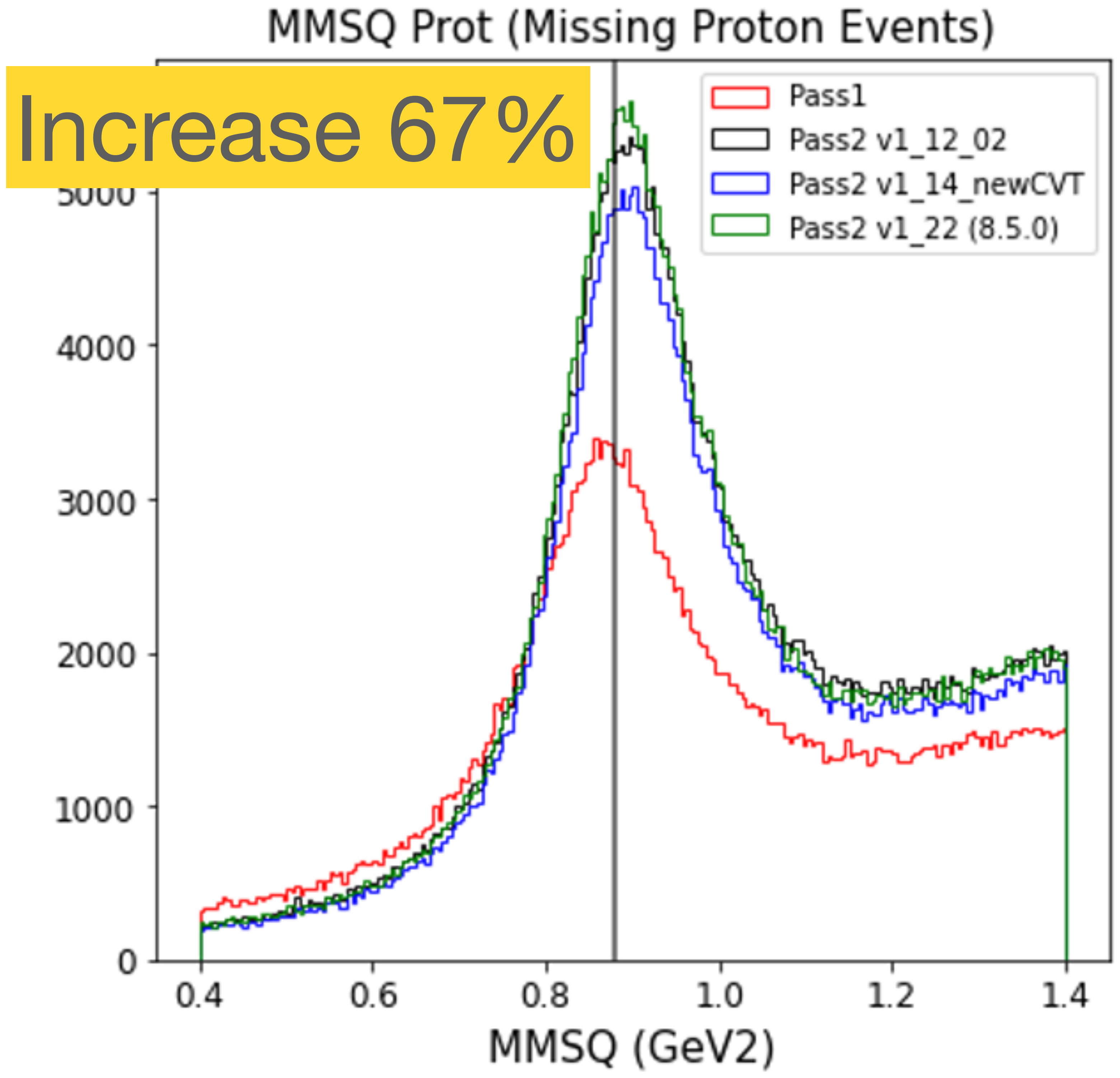
$$ep \to e' p \pi^- (X)$$



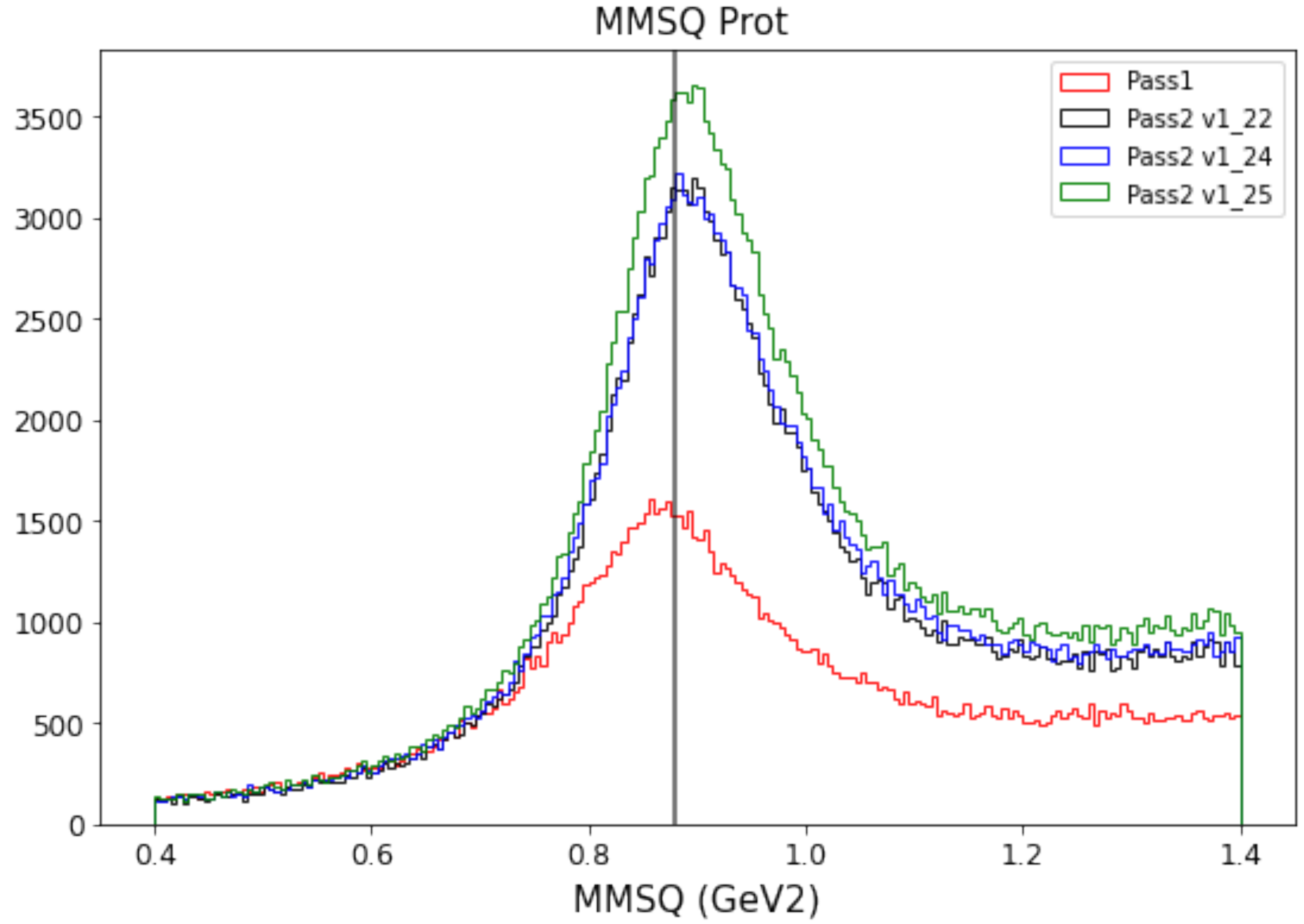

$$ep \to e' p \pi^+ (X)$$



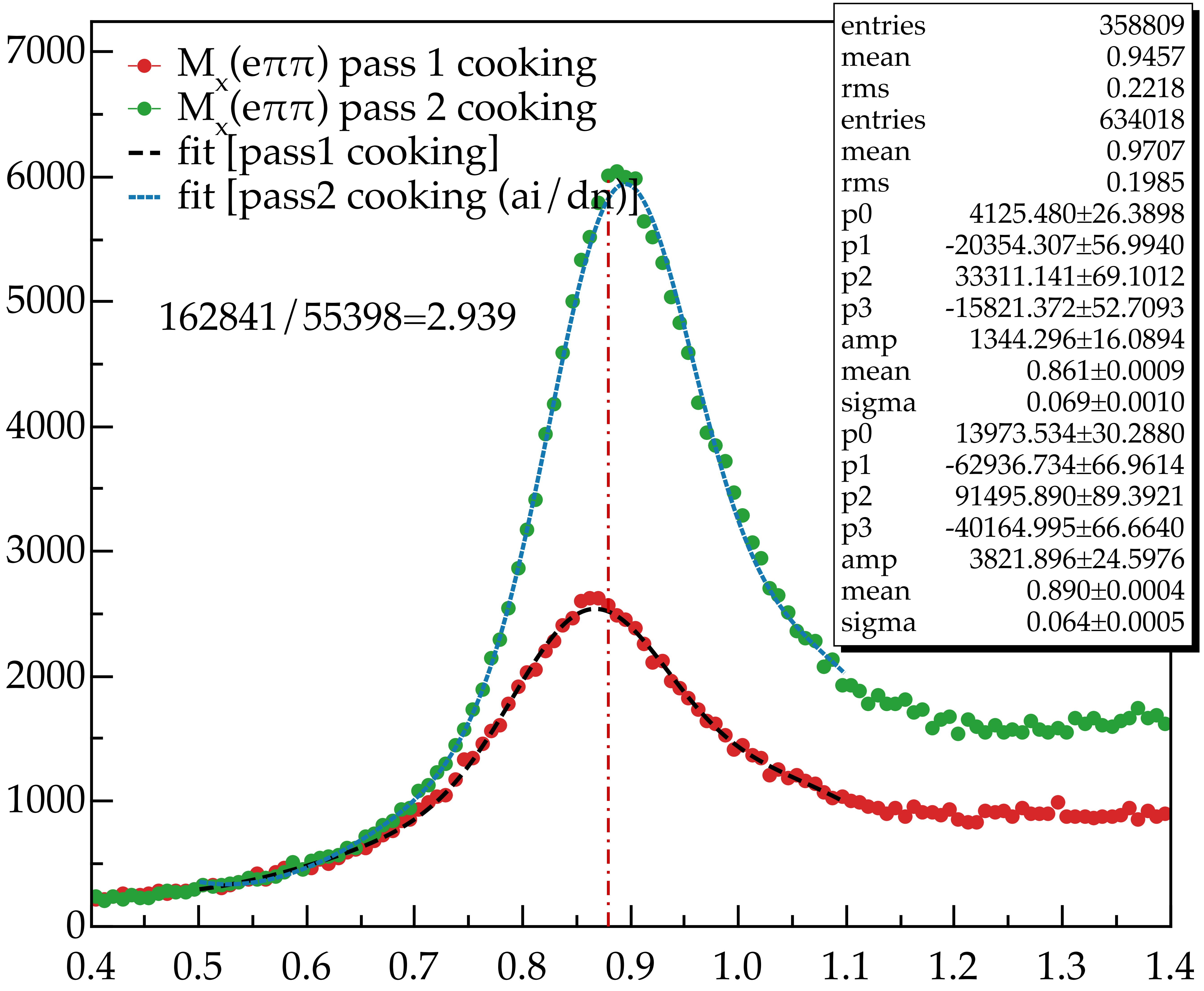

$$ep \to e' \pi^+ \pi^- (X)$$



# 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