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

Richard Tyson (University of Glasgow)

CNU (October 18,2023)

Outlook

DALL.E
Open AI

Stable Diffusion

CHAT GPT

Playground

craiyon

MIDJOURNEY
All about imagination

BARD AI

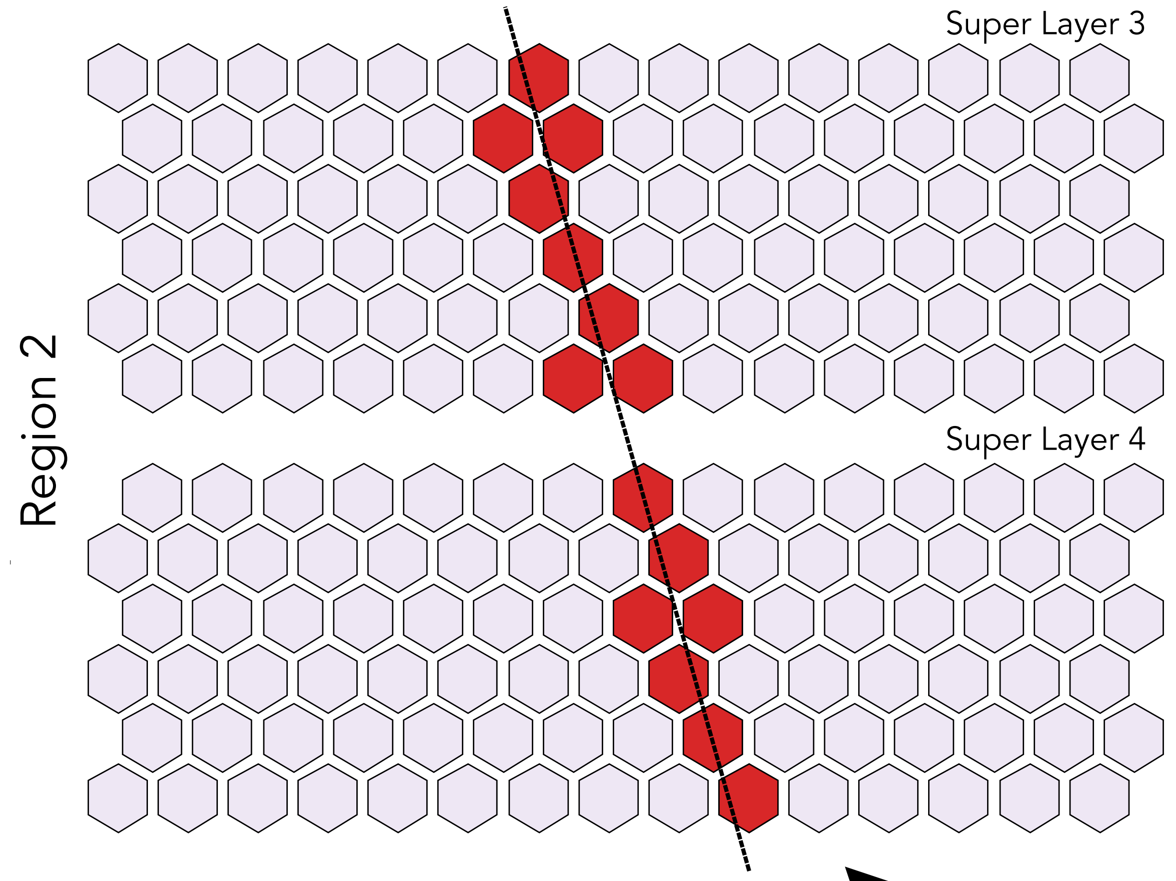
Chatsonic

GitHub Copilot

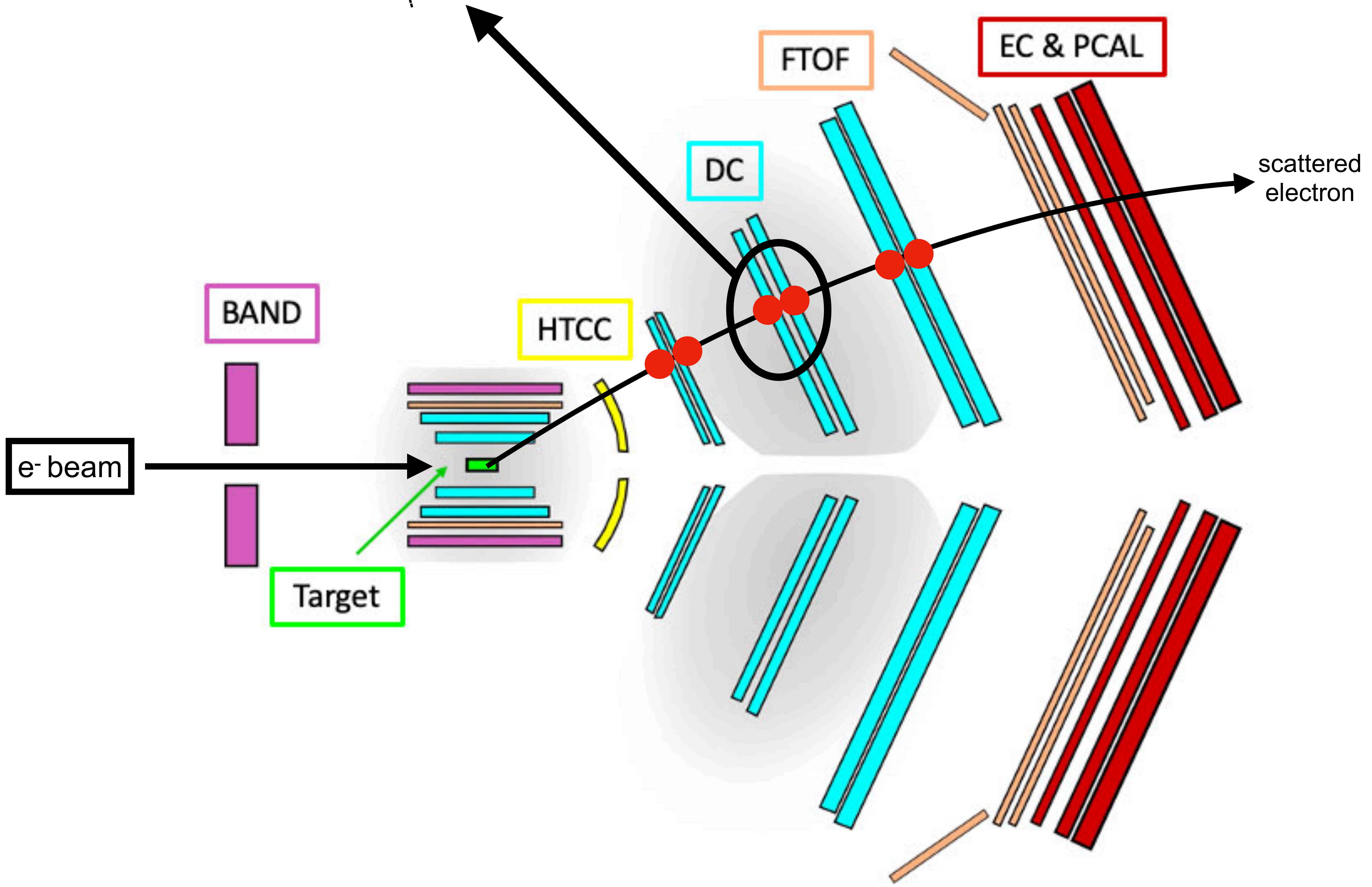
BING CHAT

- Outline:
  - What we have achieved with AI in CLAS12
  - What other ideas do we have?
  - How does this impact future developments?
  - How the data analysis will change

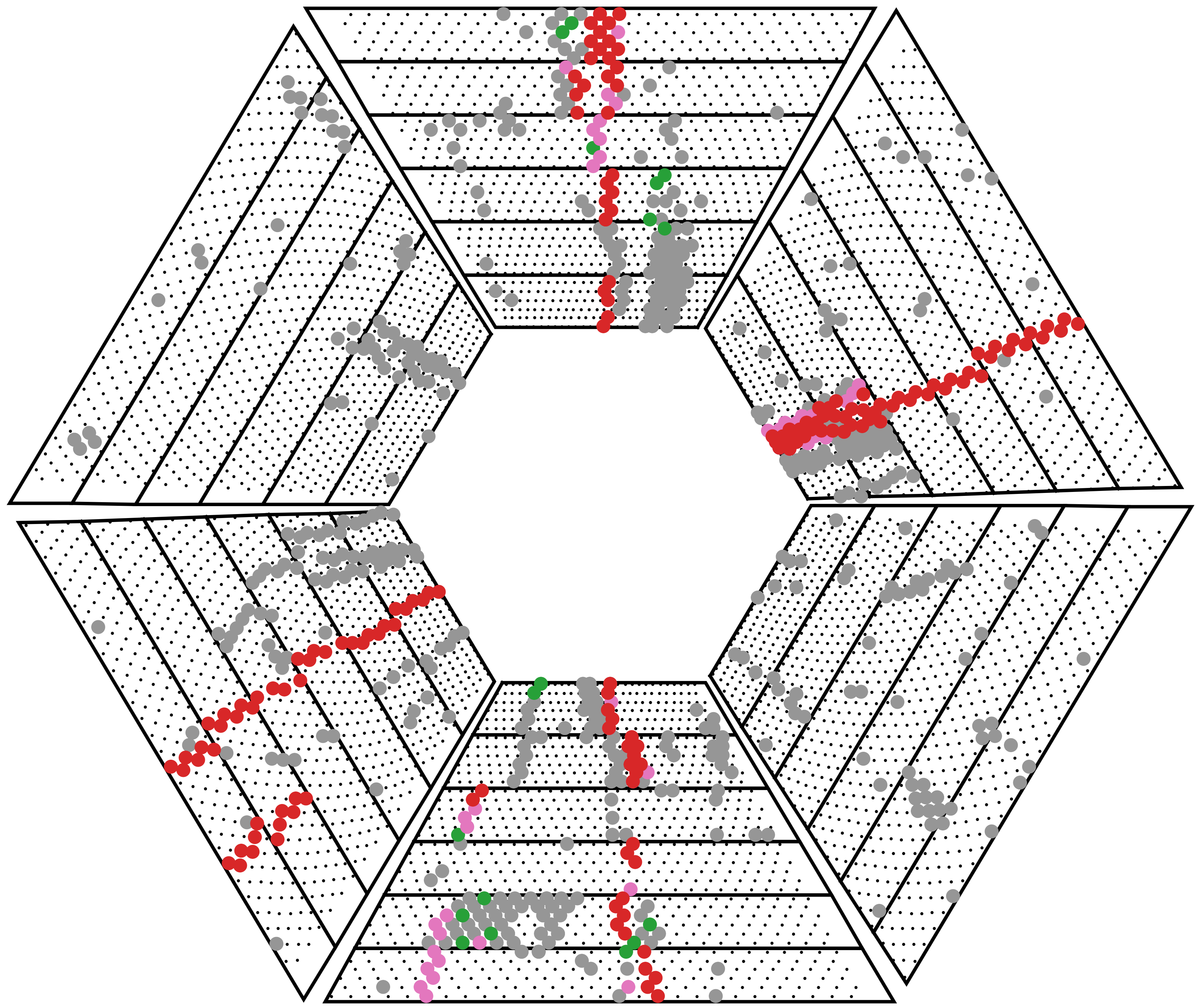
CLAS12 Detector



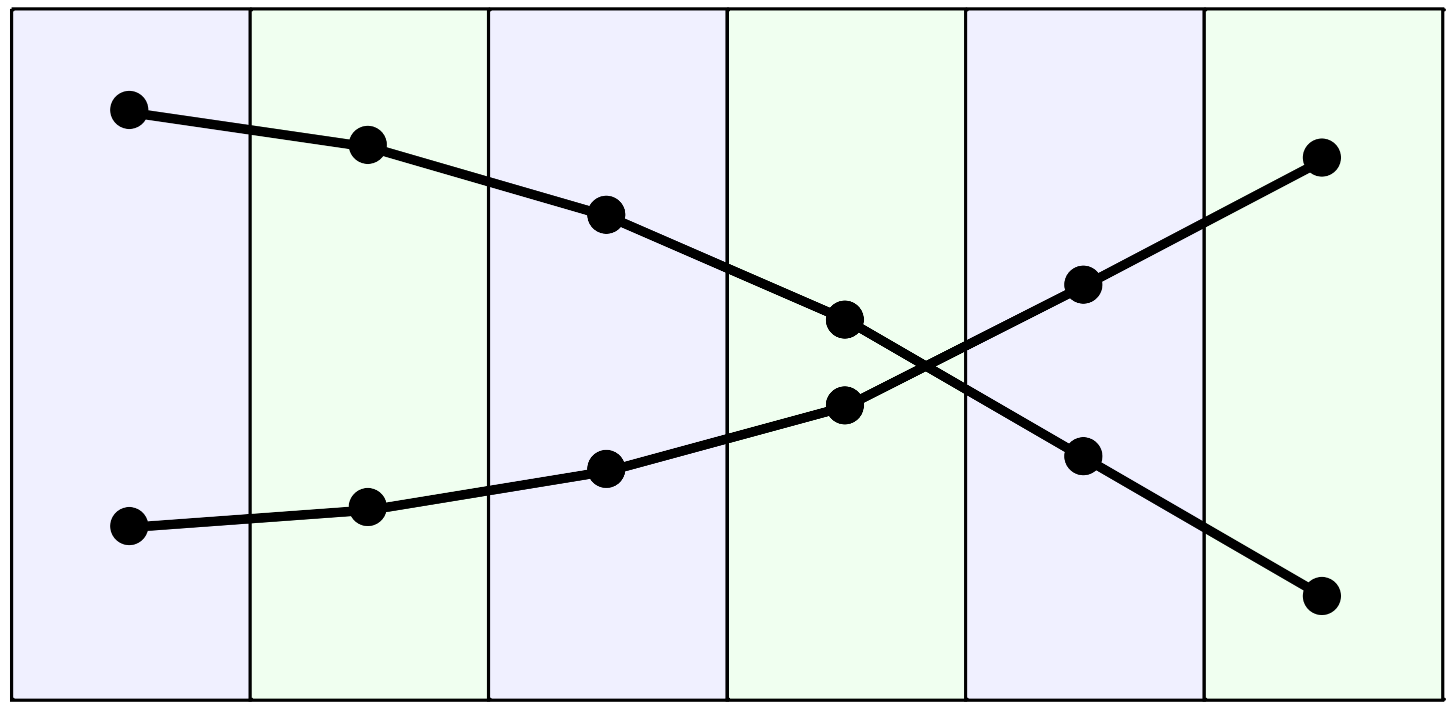
- 2 super layers in each region
- 6 wire planes in each super layer with 6-degree tilt relative to each other, (112 wires in each plane)
- Clusters in each super layer are considered part of the track trajectory



- 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

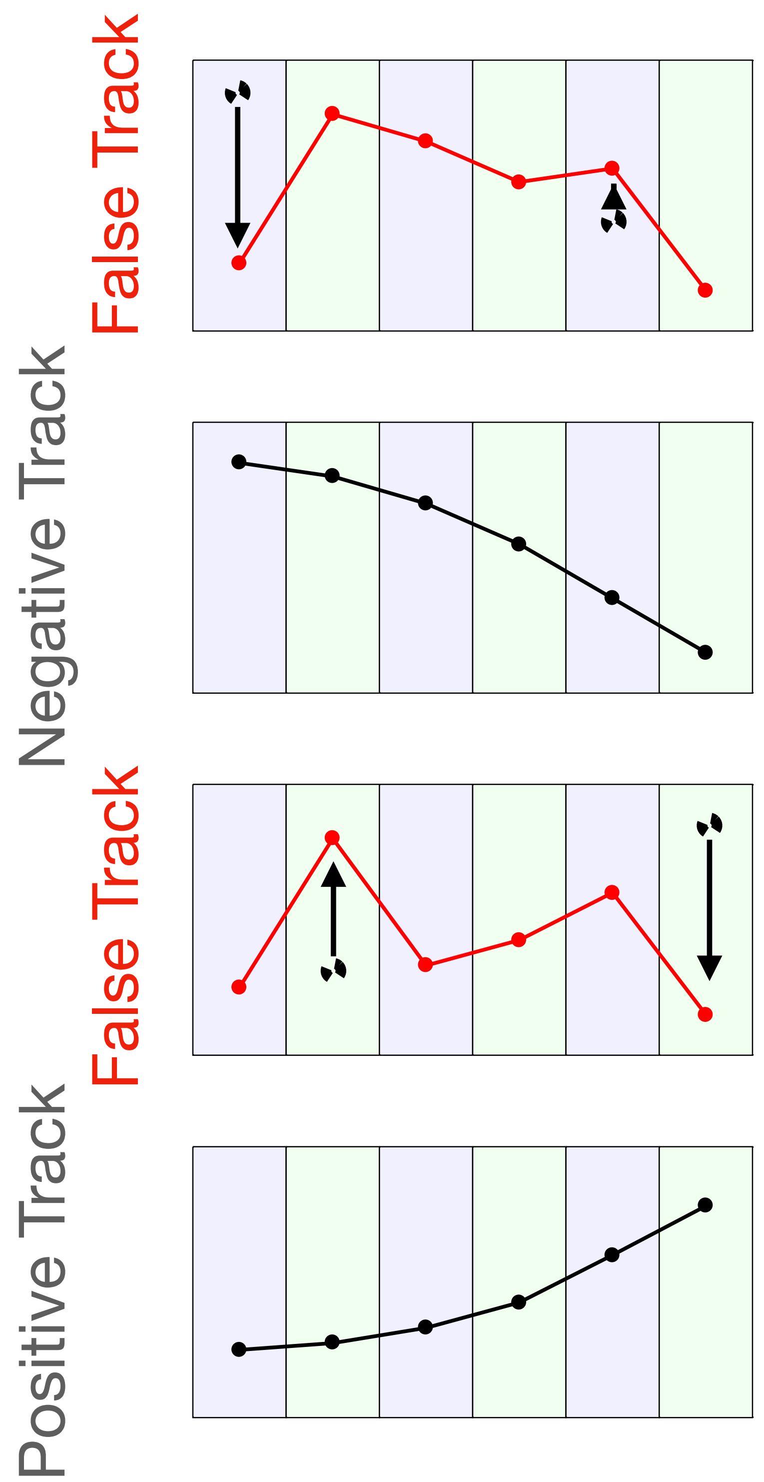


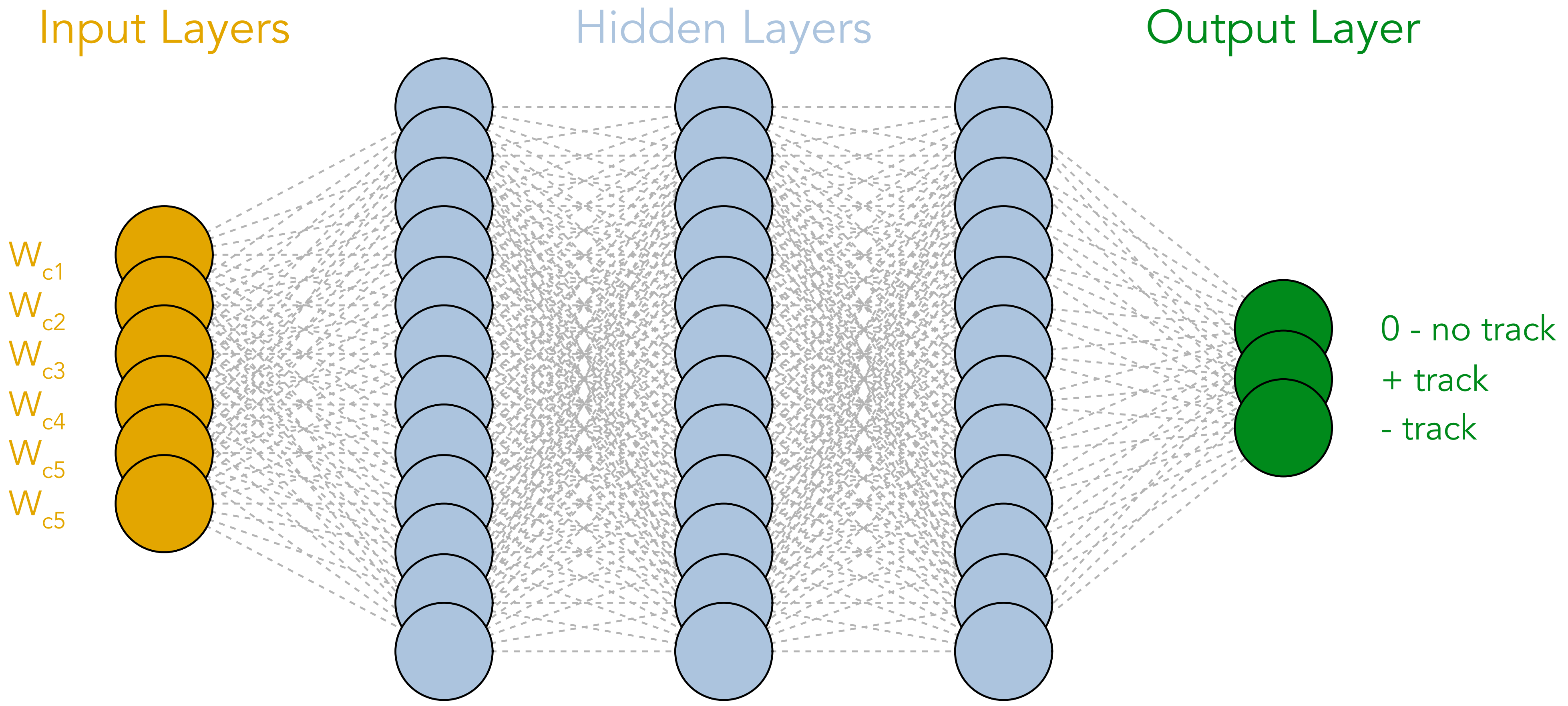
Physics Results



- 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.

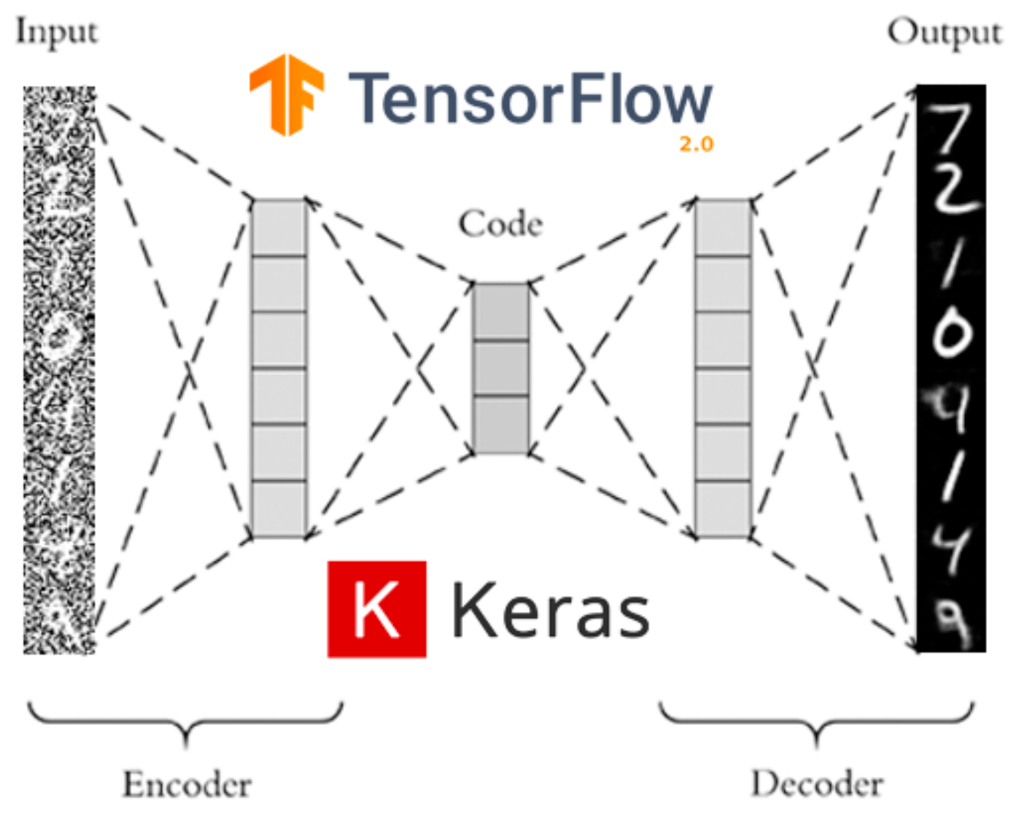
- The average wire position in each super layer is used as an input to Multi-Layer Perceptron (MLP)
- The network is trained on 6 inputs and produces three outputs:
  - False track
  - Negative Track
  - Positive Track



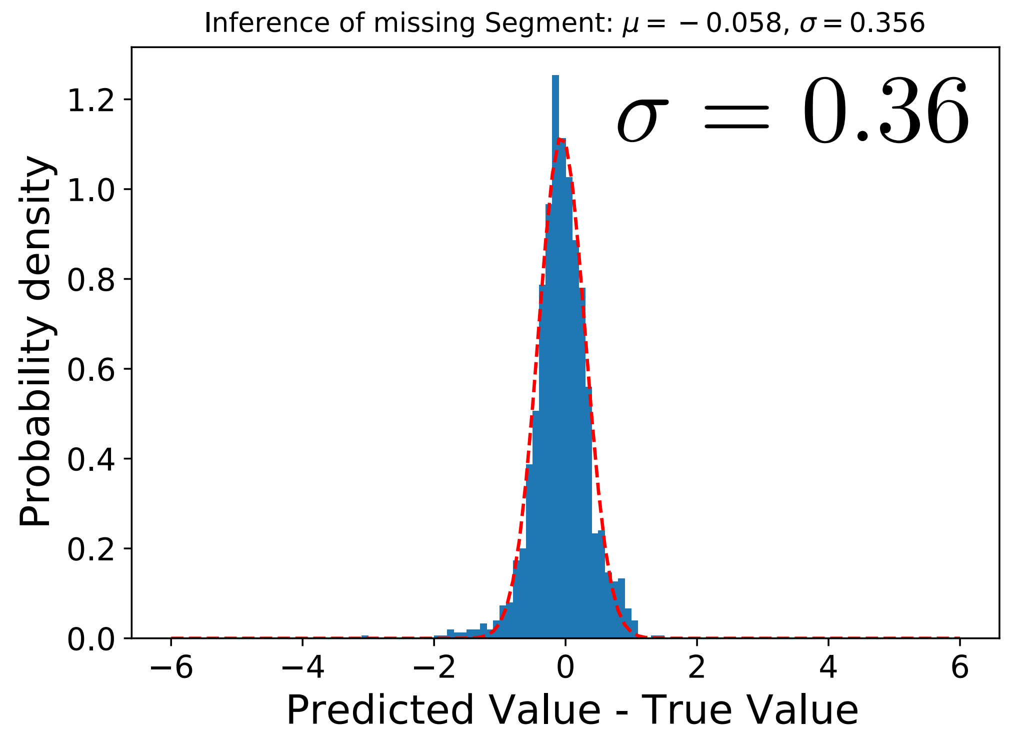


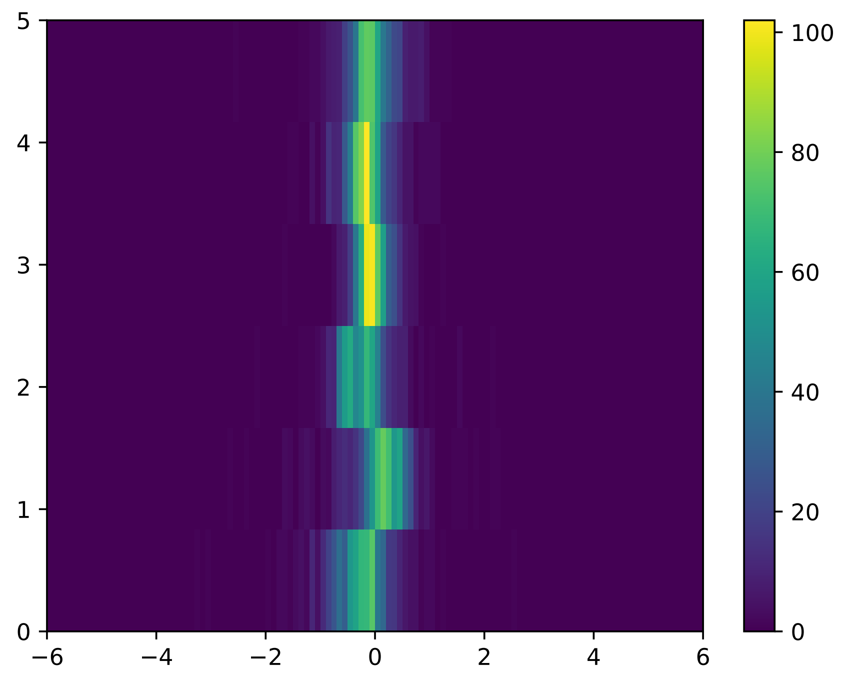
Corruption Auto-Encoder

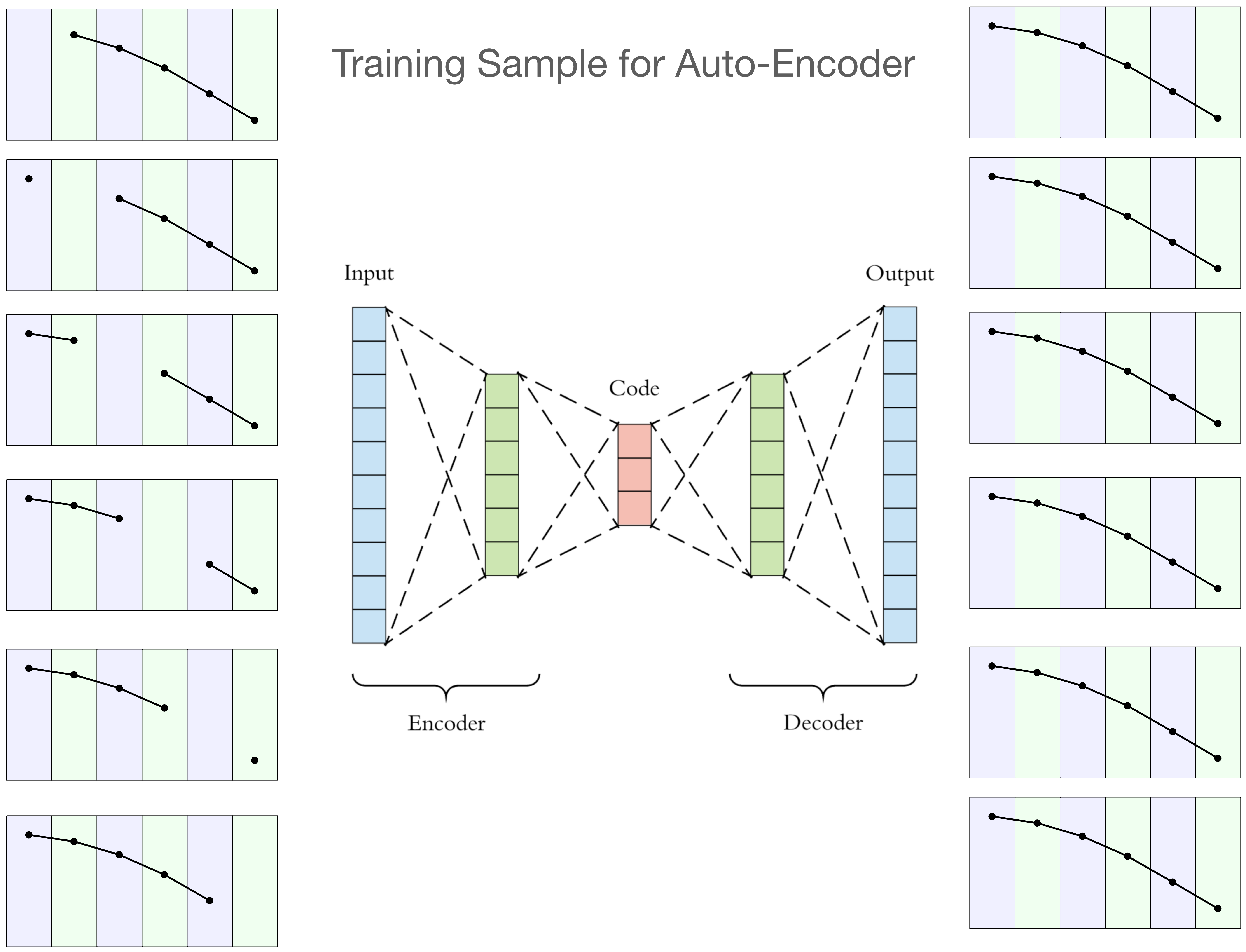
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the encoder.
- **Typically used for de-noising, but can be used for fixing glitches (our case).**



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

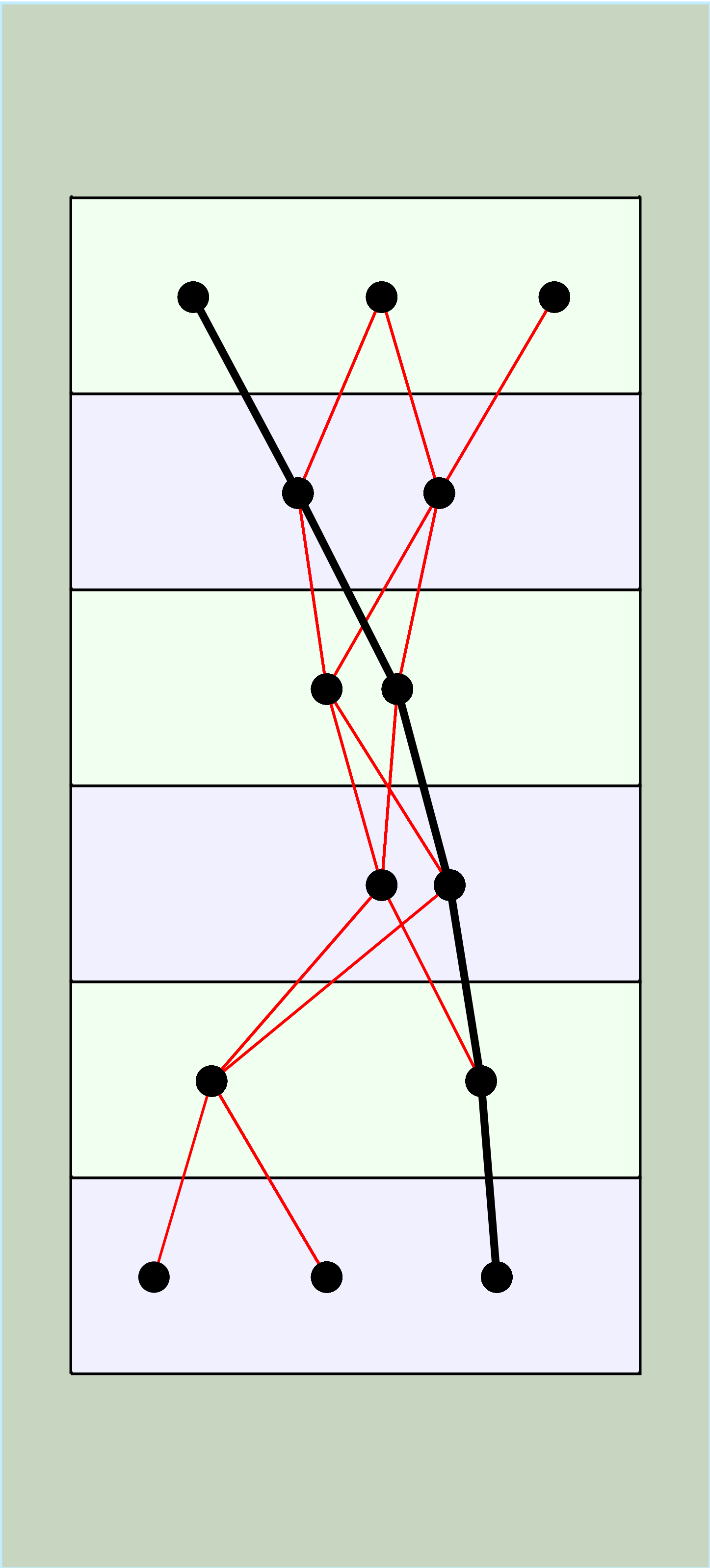




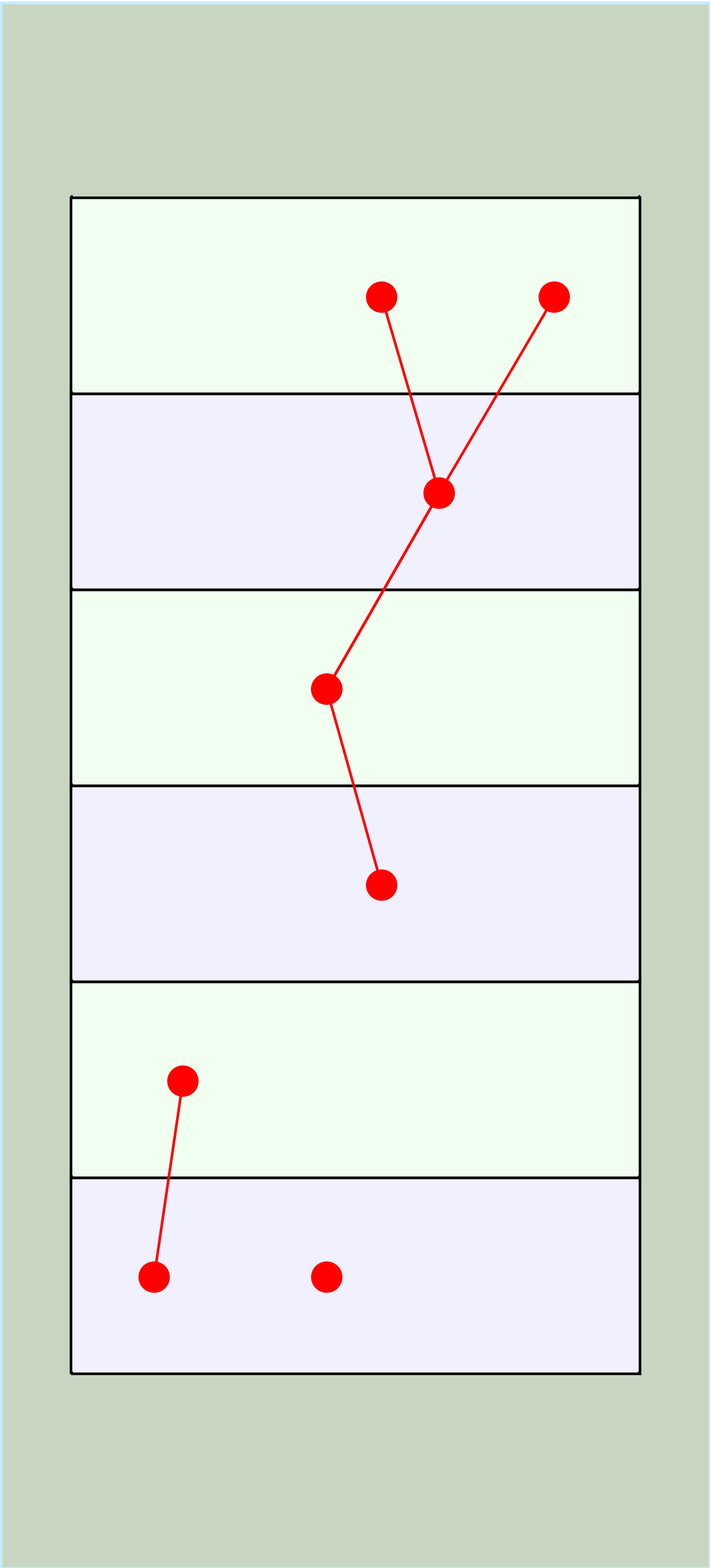


- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

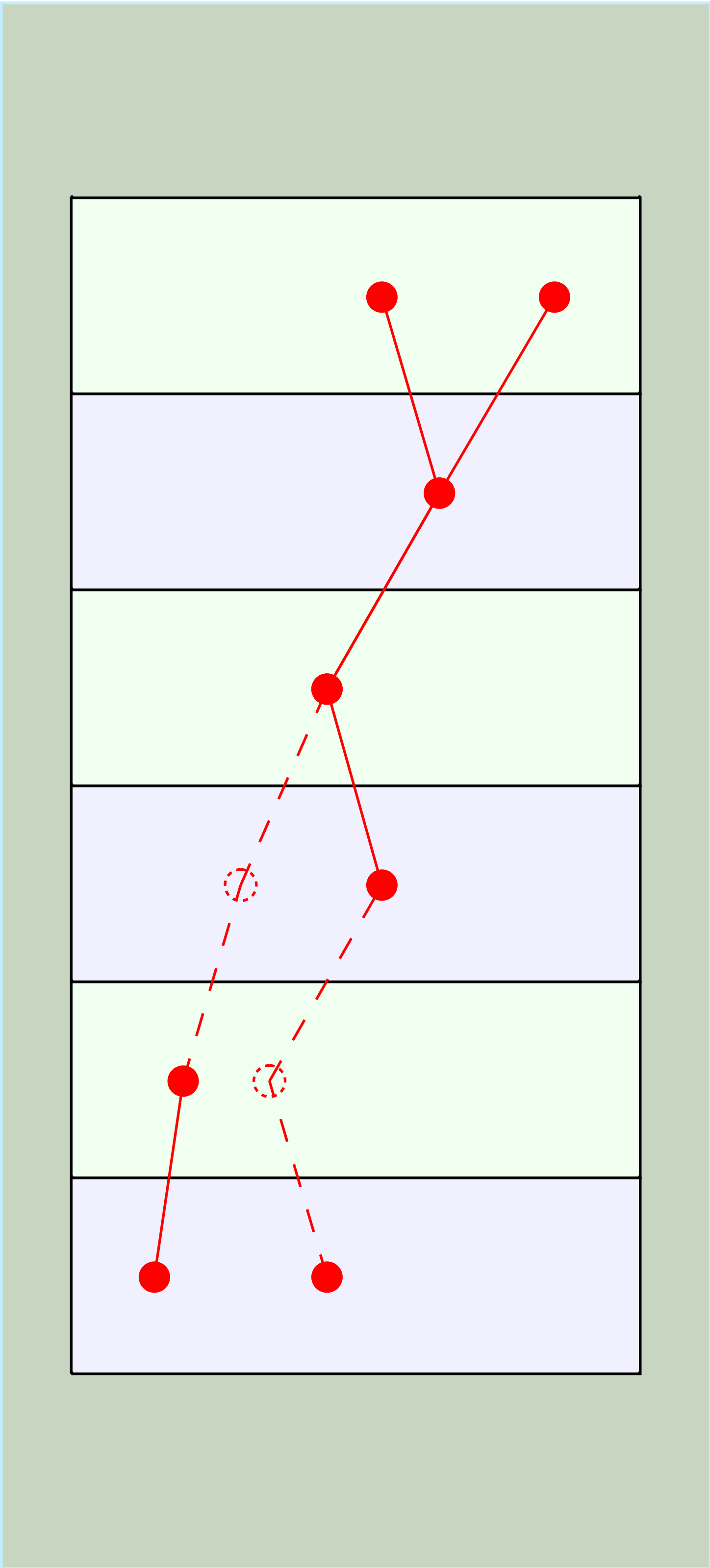
Putting things together



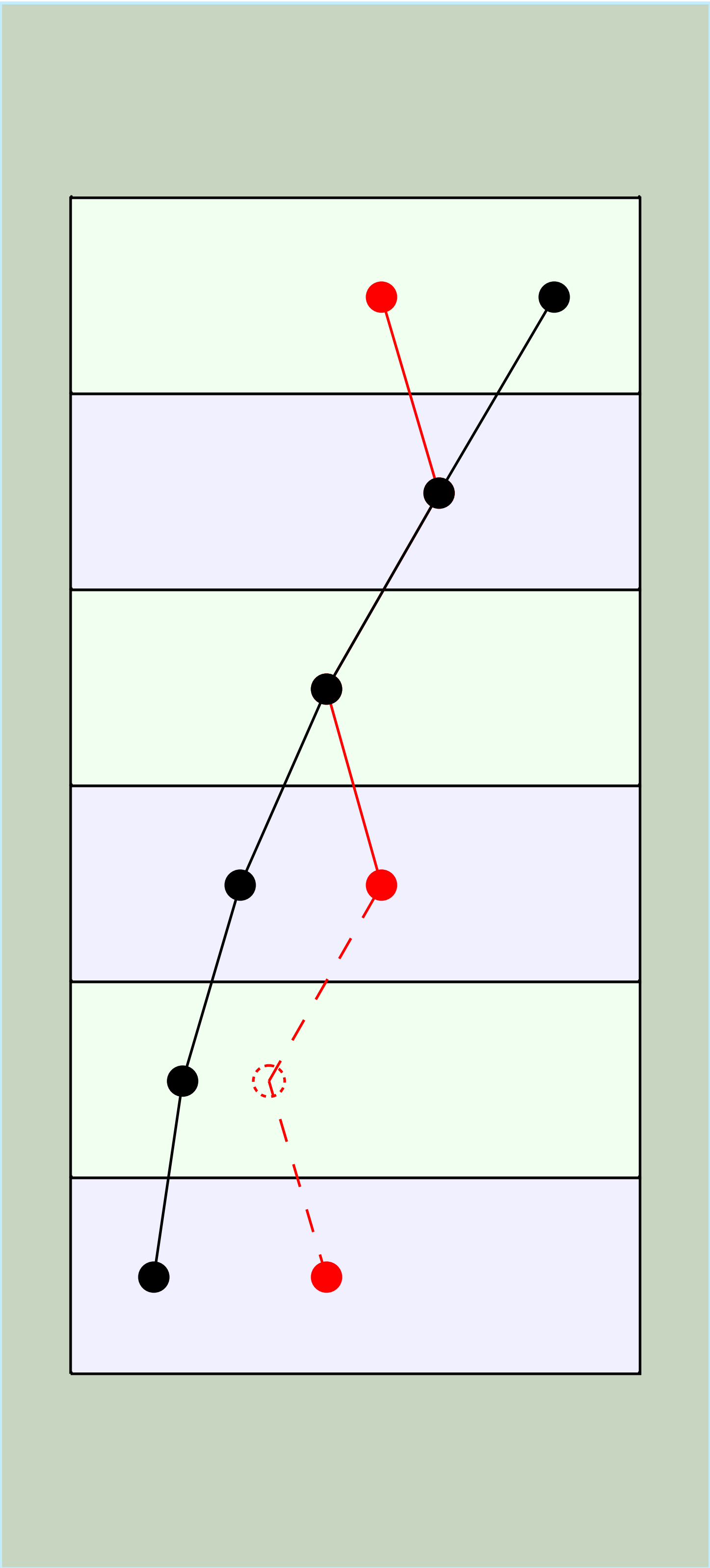
Classifier picks the correct track from 6 super-layer combinations



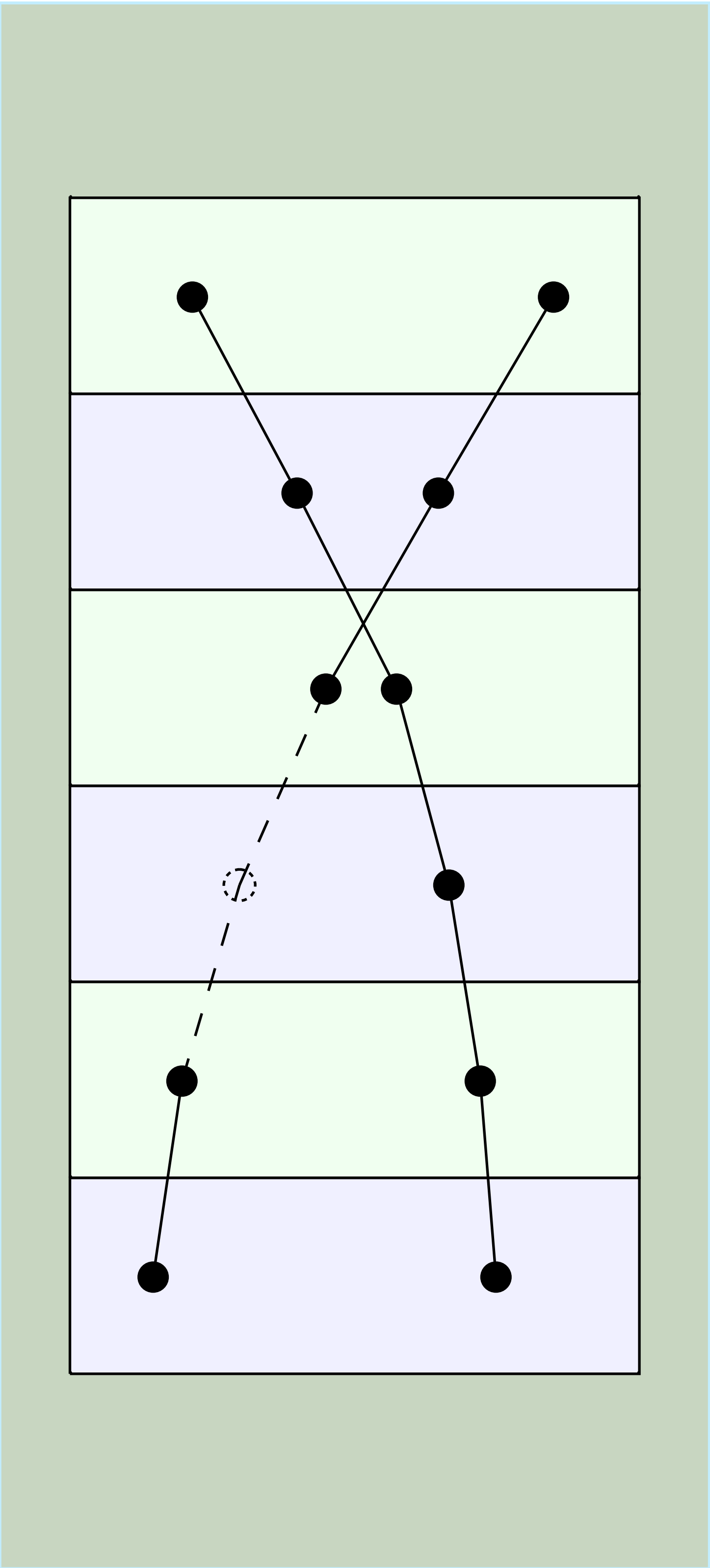
Remove all clusters belonging to identified track



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



Identify tracks using 6 super layer candidates with pseudo-clusters

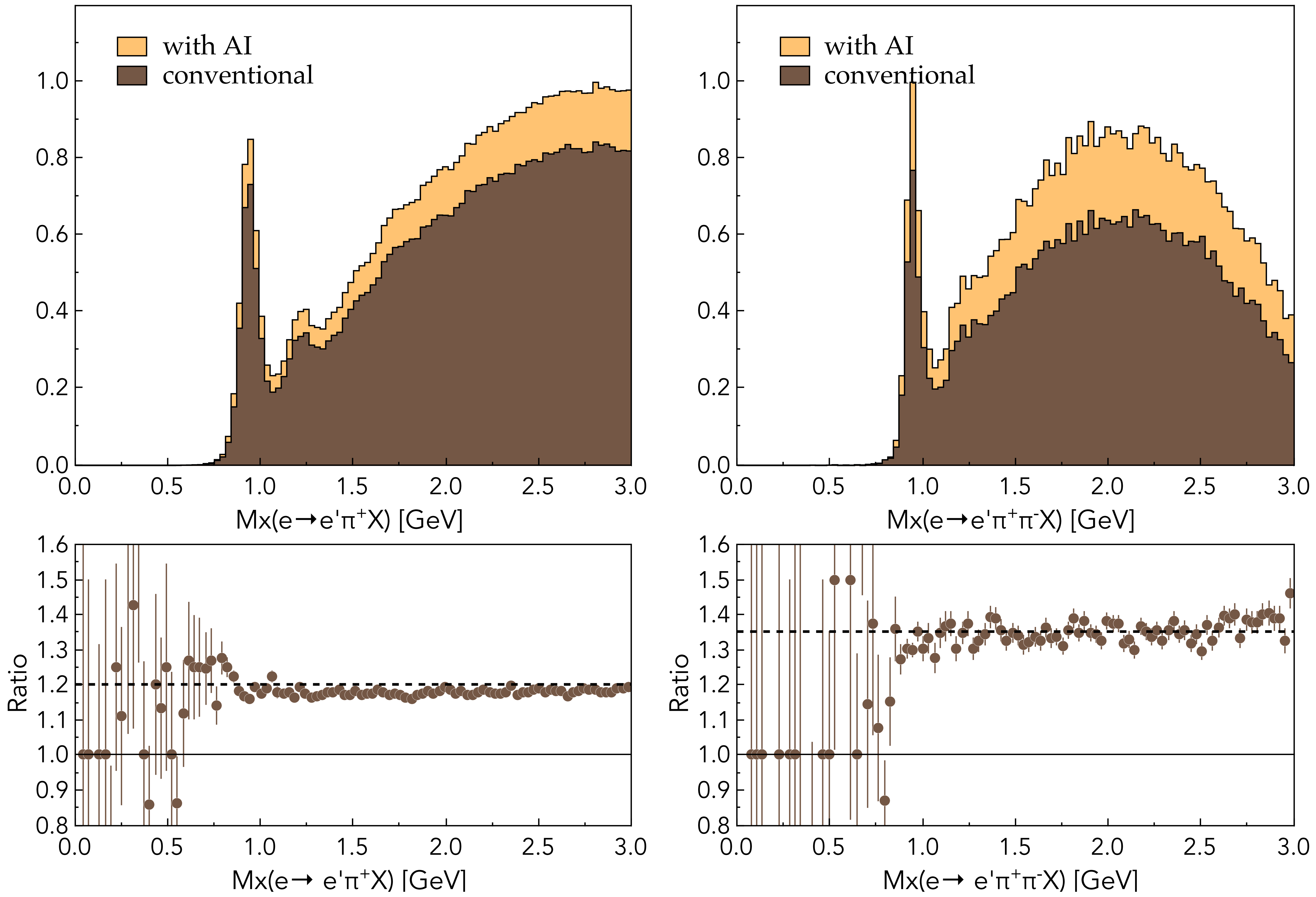


Voila!

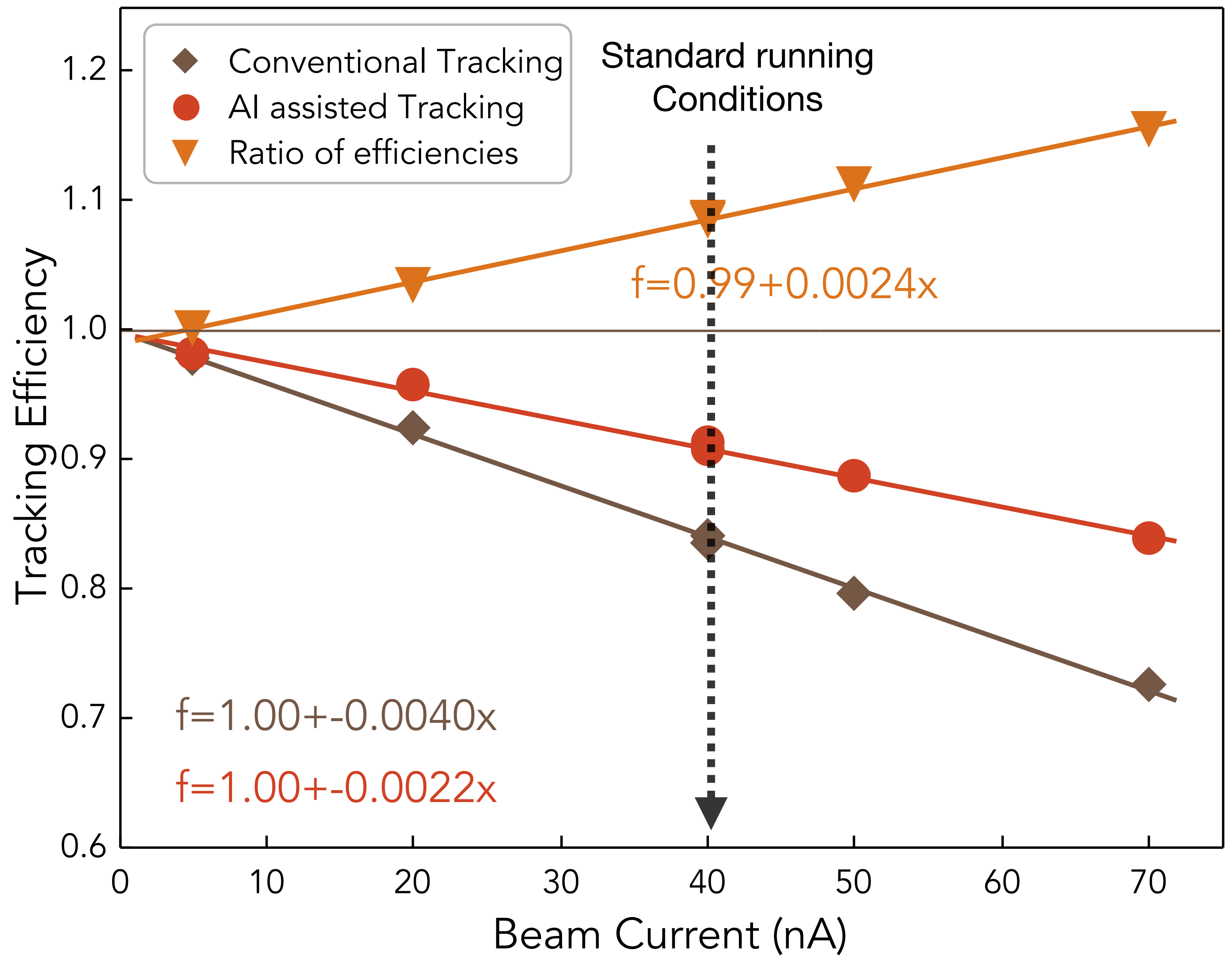
Physics Results

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

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

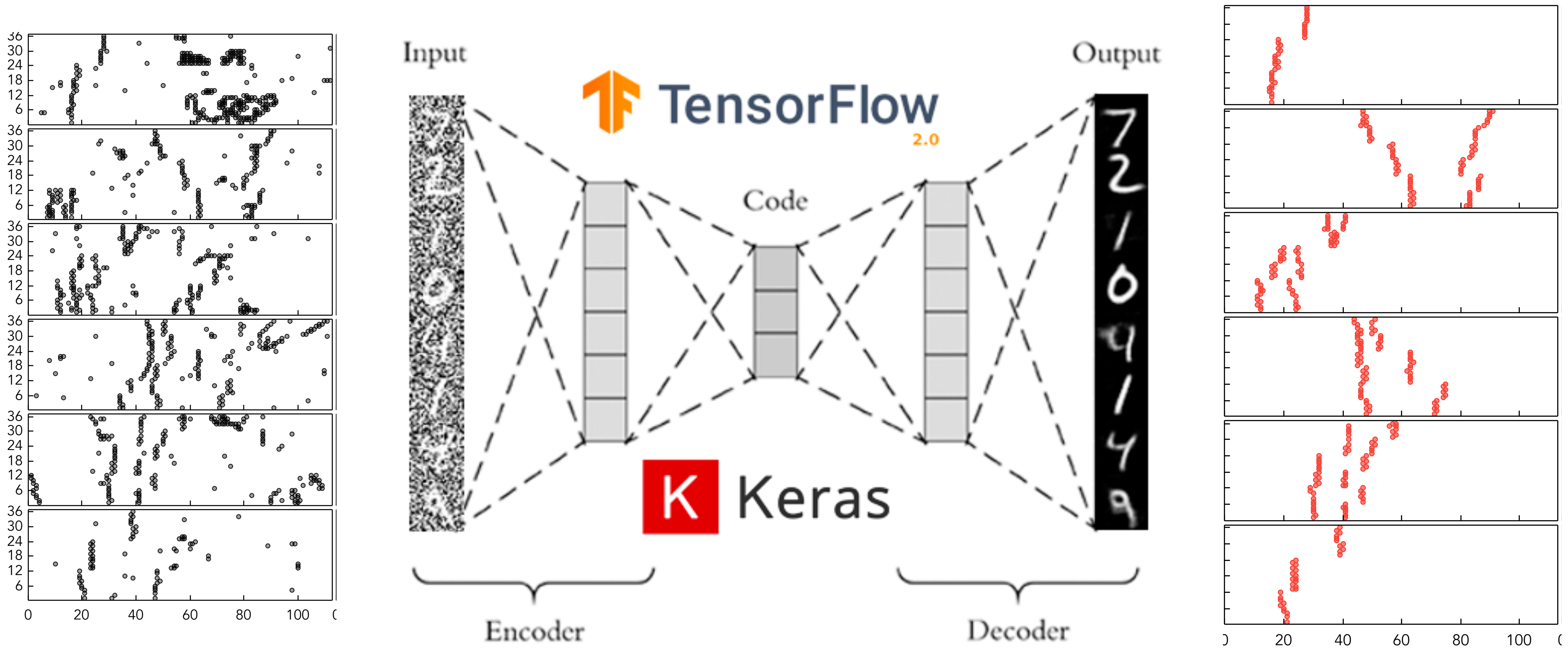


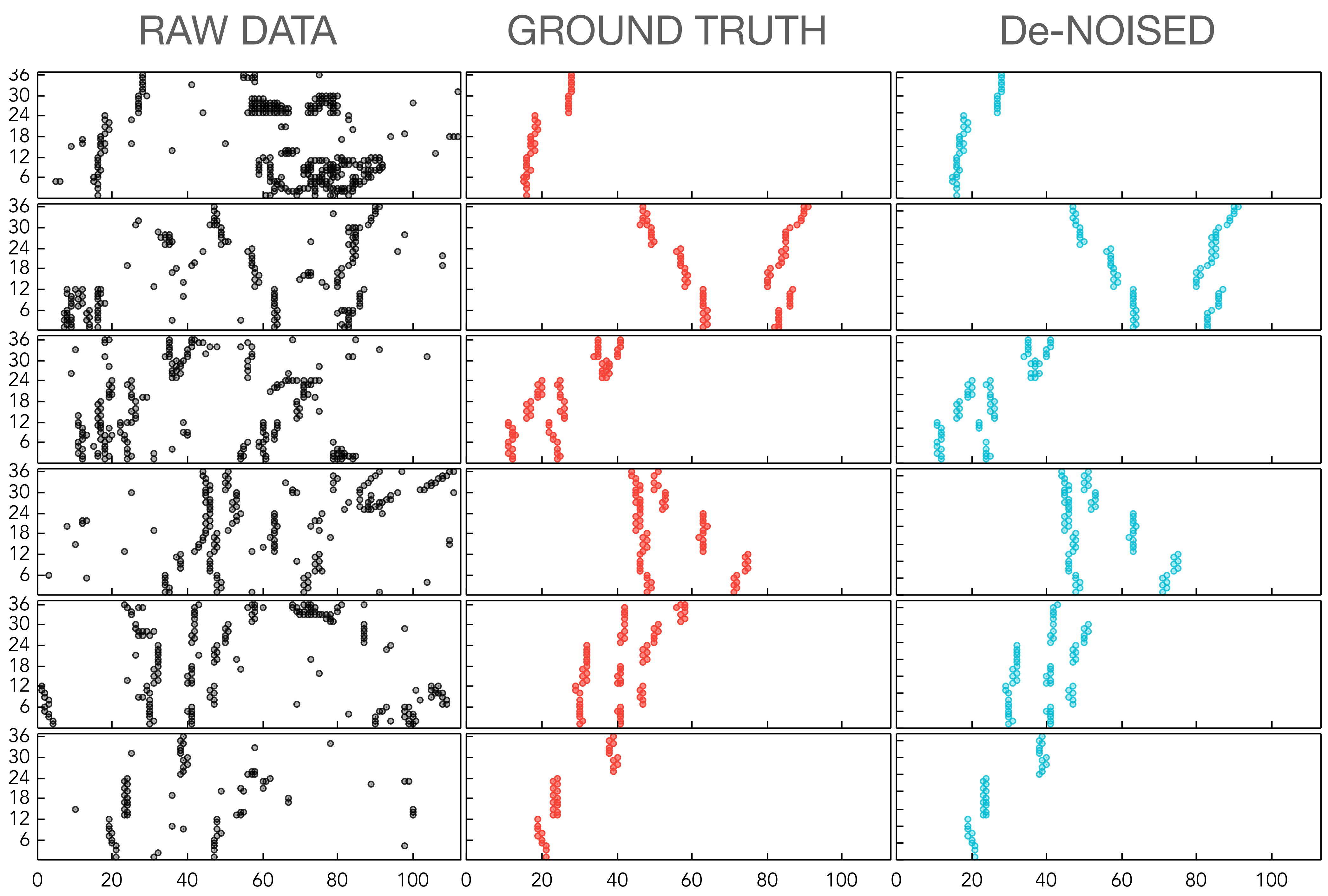
- Single particle efficiency increases by ~10%.
- The impact on physics for a multi-particle final state is dramatic (20% for the two-particle final state and ~35% for the three-particle final state)
- The tracking code speedup is ~30%.



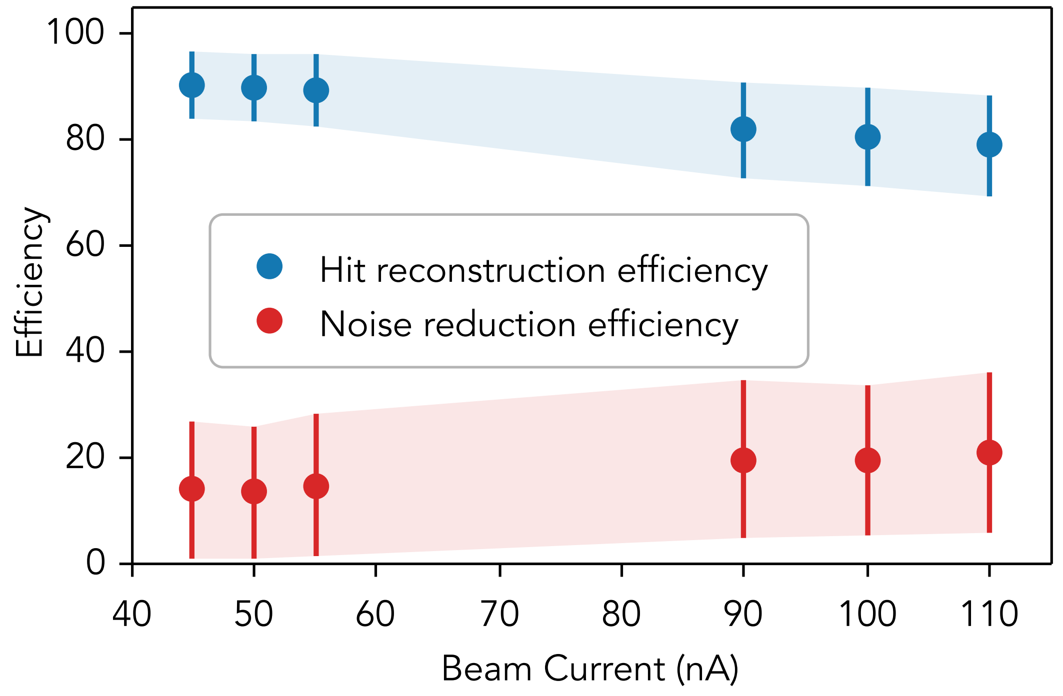
De-Noising

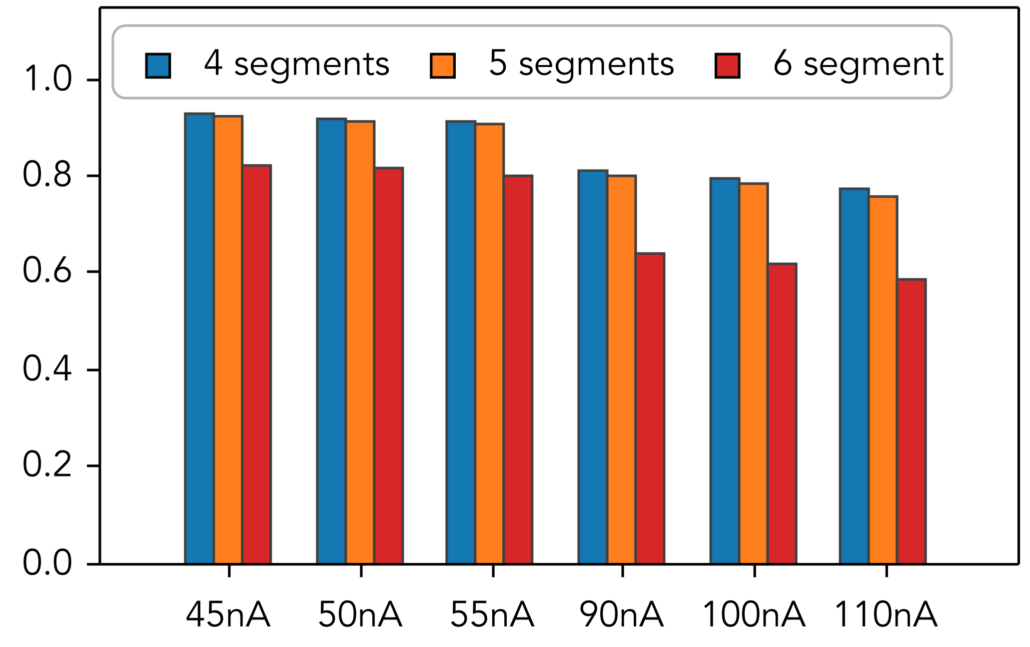
- Convolutional Auto-Encoder is used to de-noise raw data from drift chambers.
- The network is trained on reconstructed data with track hits isolated from raw DC hits.
- The network is able to isolate hits that potentially belong to a valid track through drift chambers





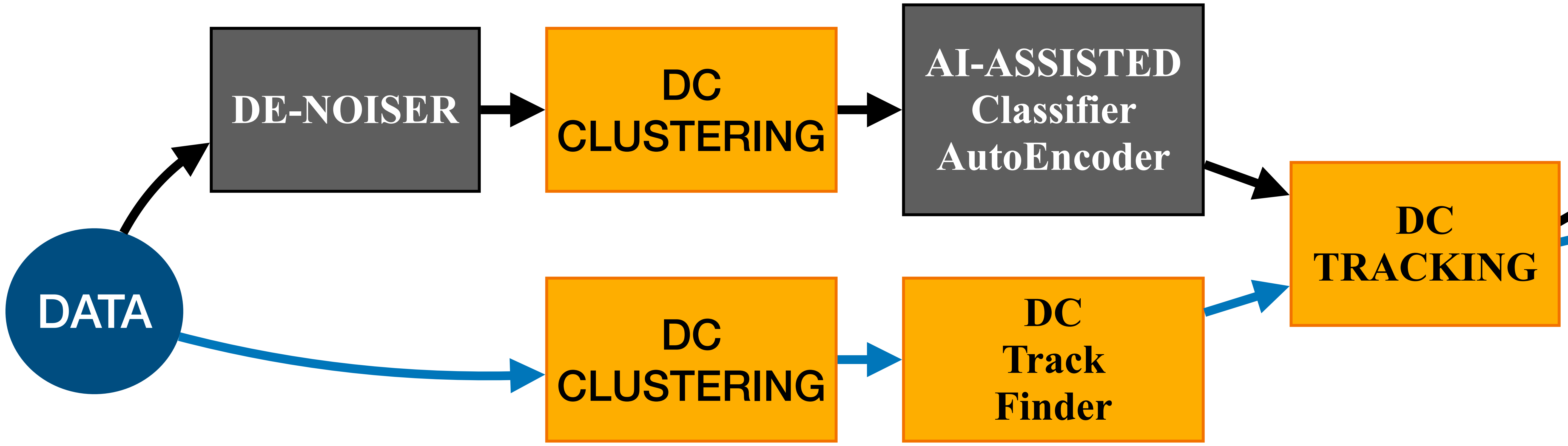
Network Performance Summary

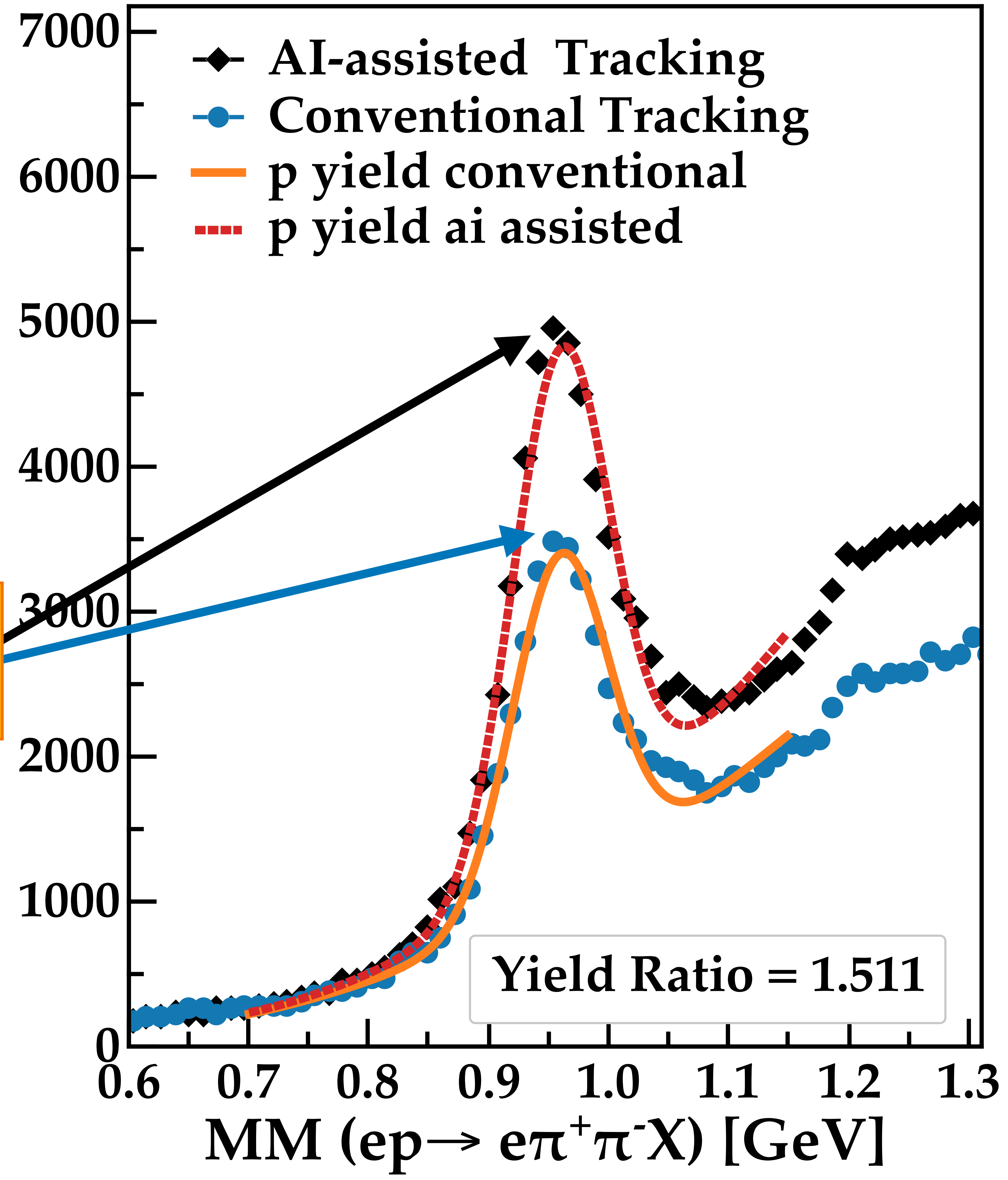




Current Workflow

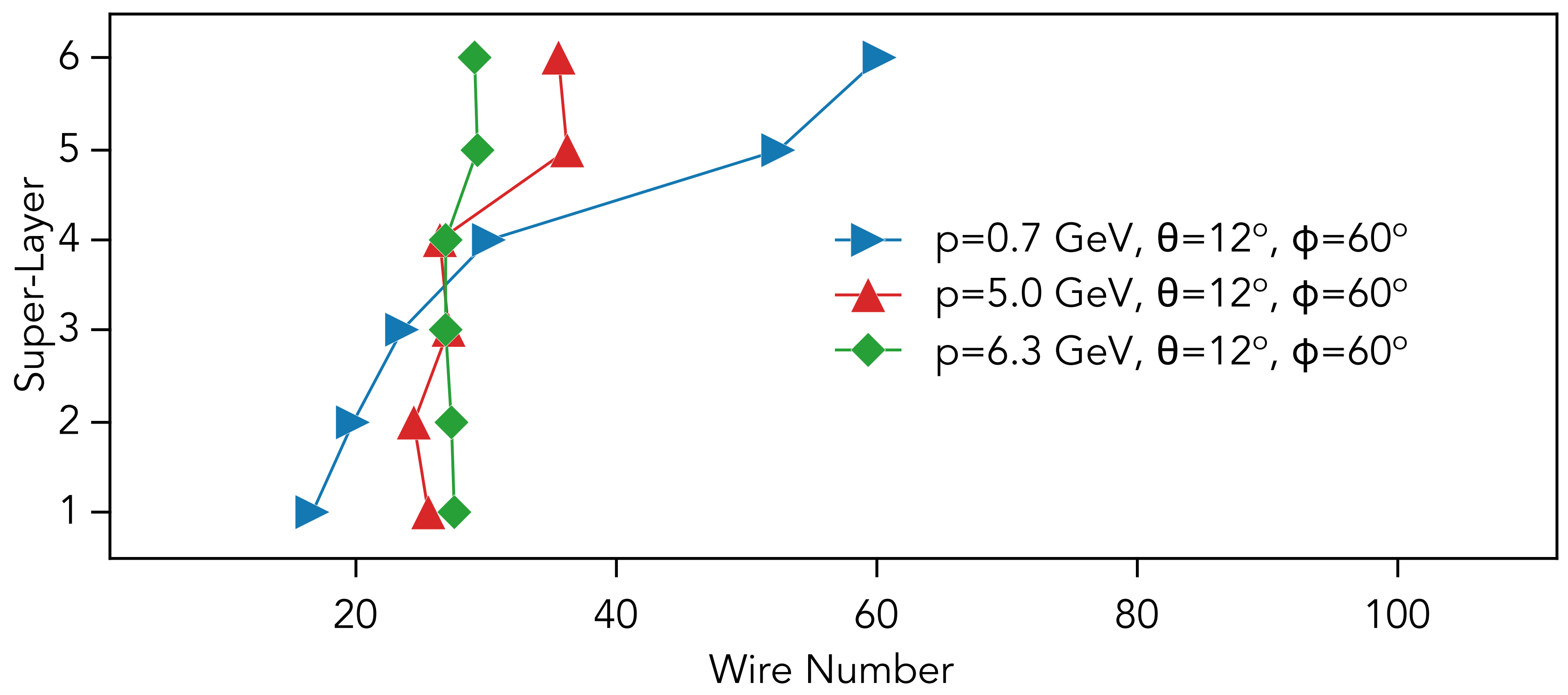
- CLAS12 Reconstruction software is based on SOA (CLARA) approach, where each detector reconstruction runs as a separate service
- The data reconstruction workflow now included de-nosier running prior to standard clustering and AI-Assisted tracking running prior to DC track finding.
- Drift Chambers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

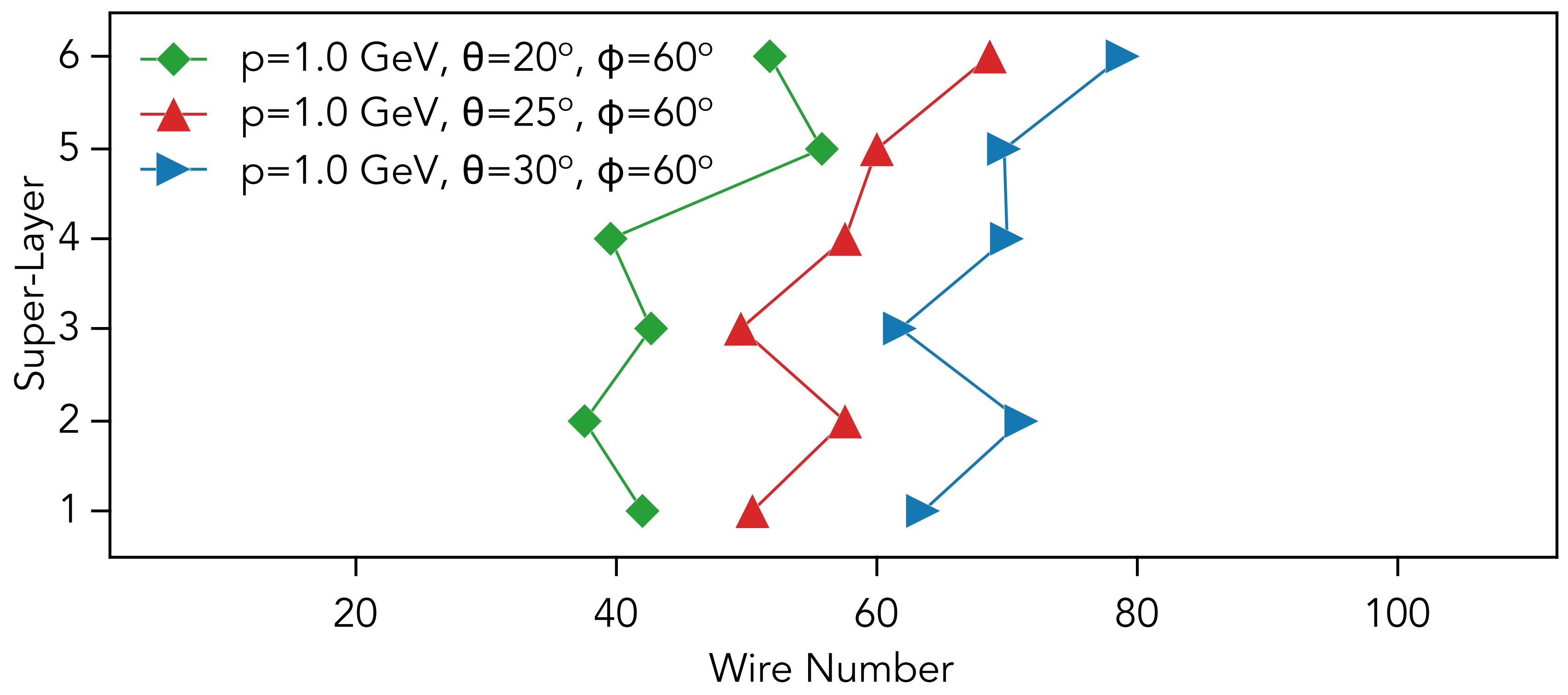


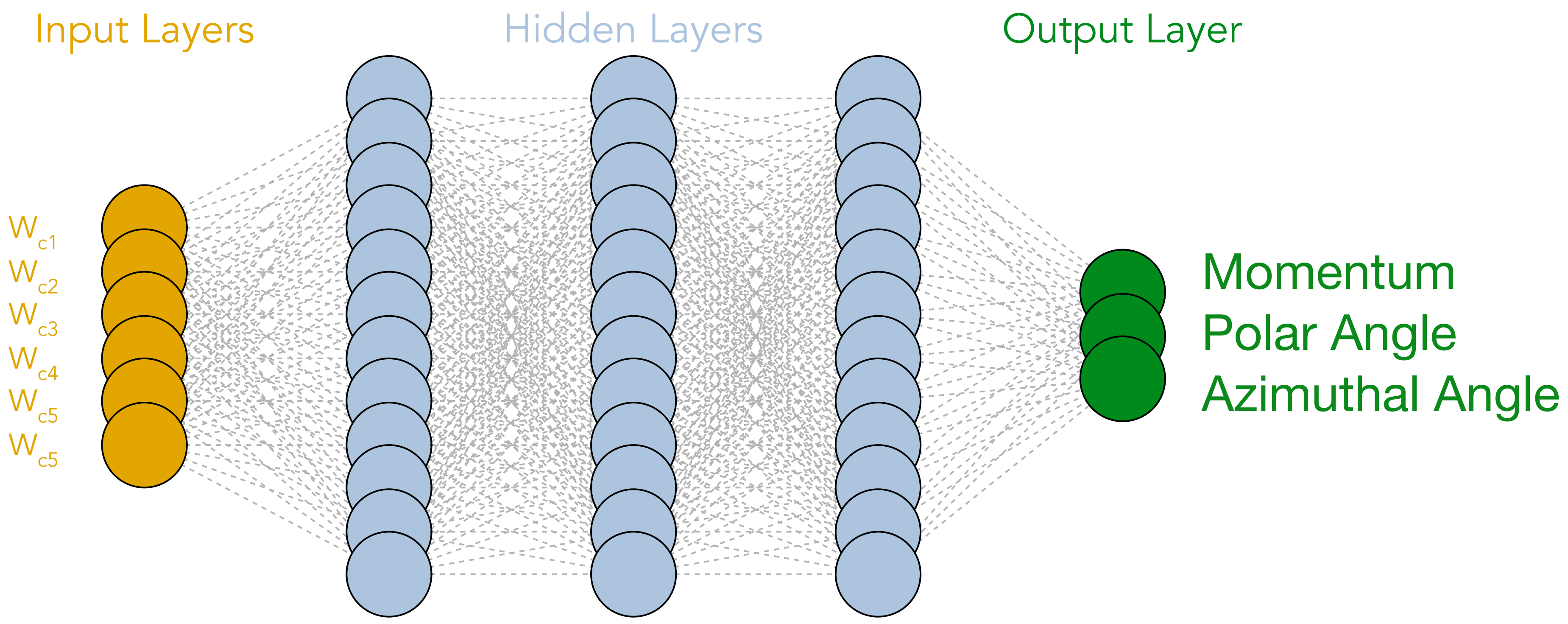


- Running at standard conditions (45 nA beam current) the AI increased the yield of missing protons by 51%.
- The improvement in yield is reaction and kinematics dependent, and for some event topologies reaches even 83% (J/psi with 3 particles detected final state).

Track Parameter Reconstruction

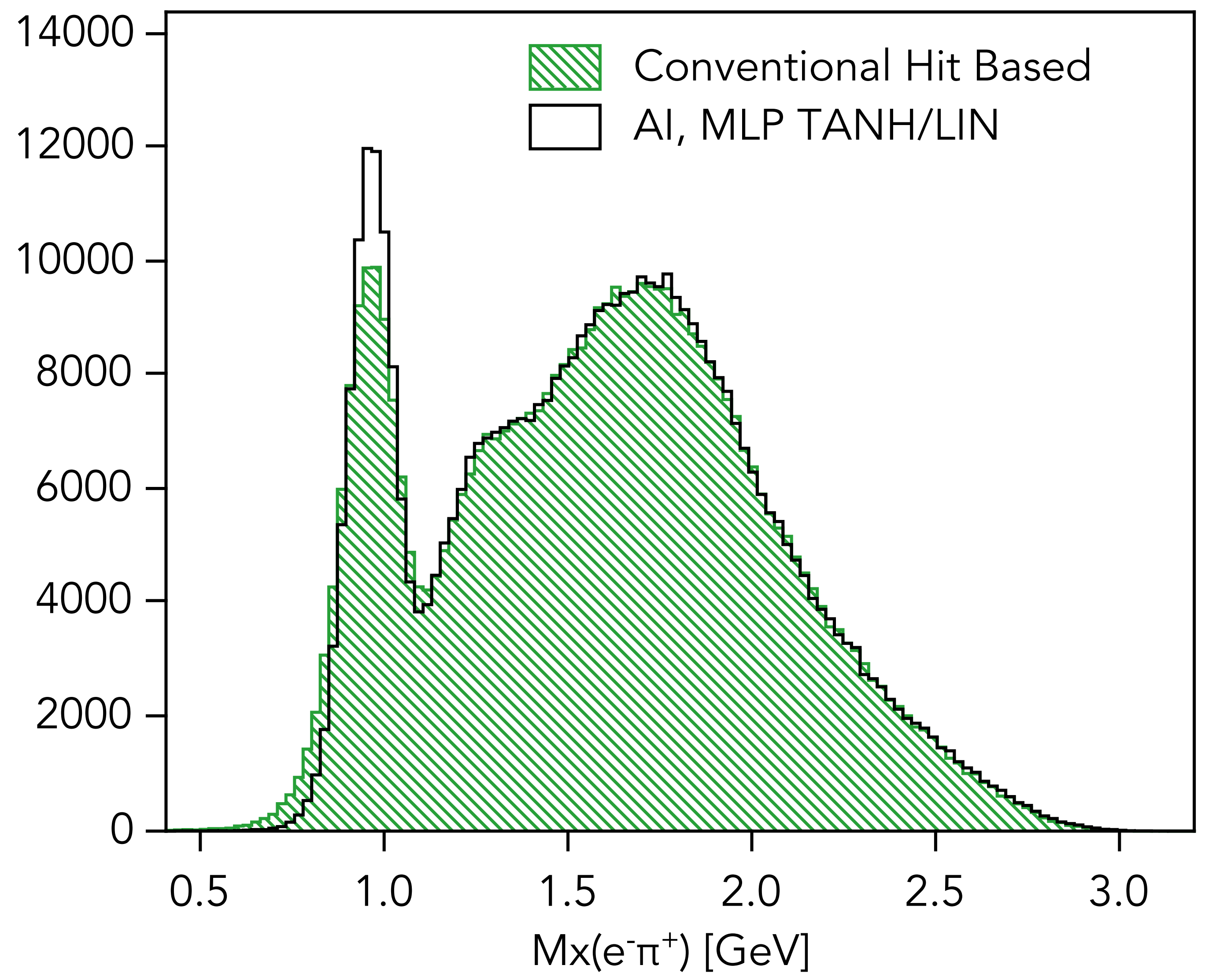






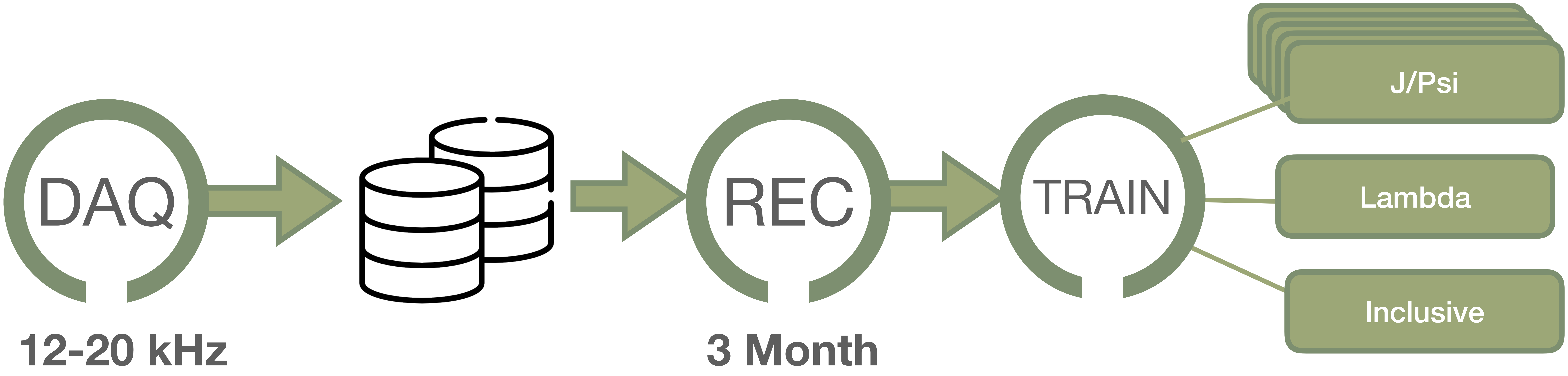
- Charge Track Parameter Inference
  - Reconstruct momentum and angles of particles based on the cluster positions of the tracks
  - Particles have distinct trajectories through drift chambers depending on their momentum, polar and azimuthal angle.
  - Design an MLP network and investigate different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

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



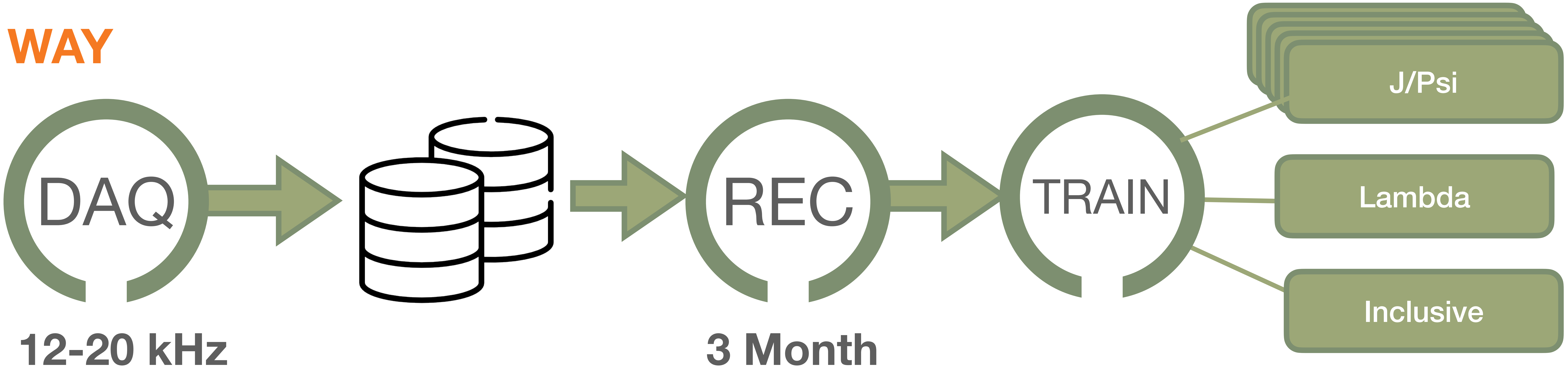
- Missing mass of two particles calculated using particle momenta from Hit-Based Tracking compared to missing mass calculated from AI particle parameter inference.
- Hit Based Tracking works ~250 ms per event
- AI reconstructs particle parameters <0.5 ms per event

Data Work Flow

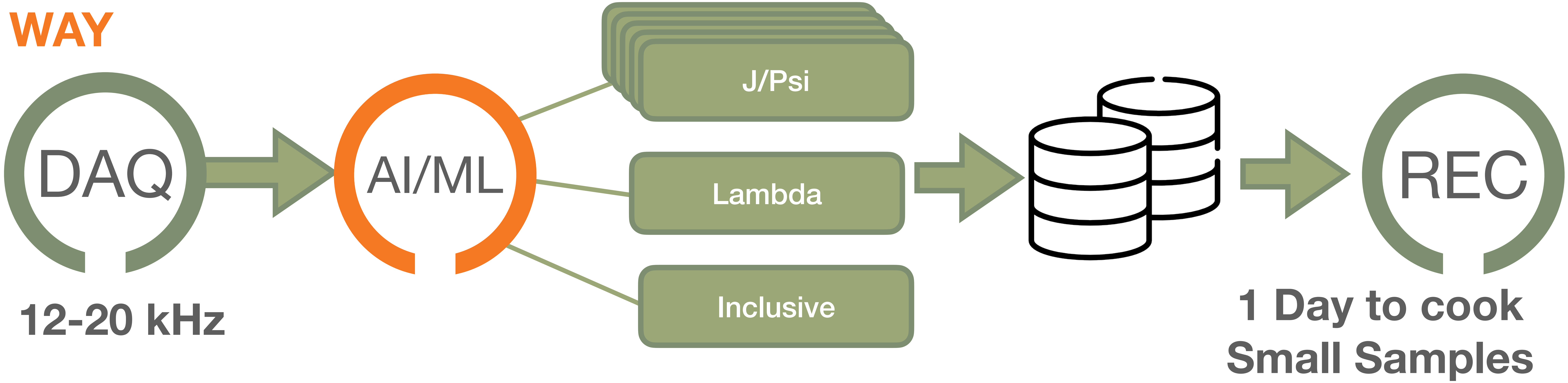


Data Work Flow

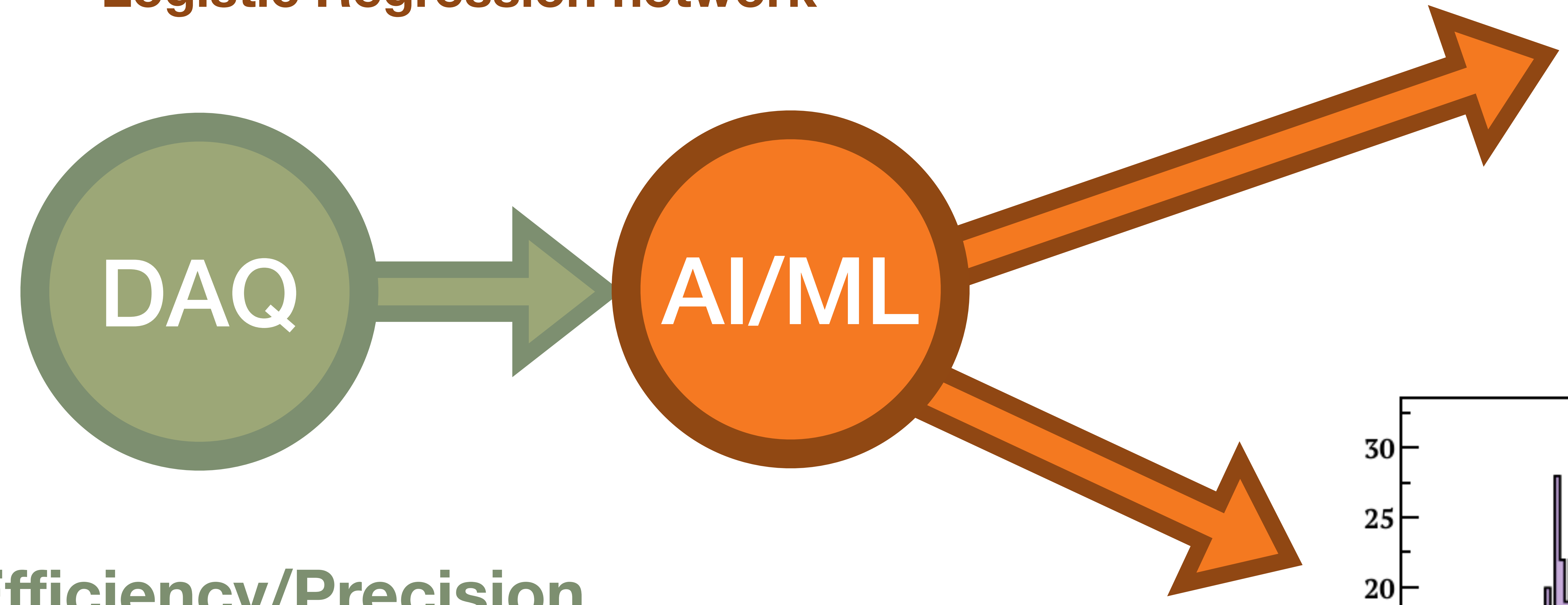
OLD WAY

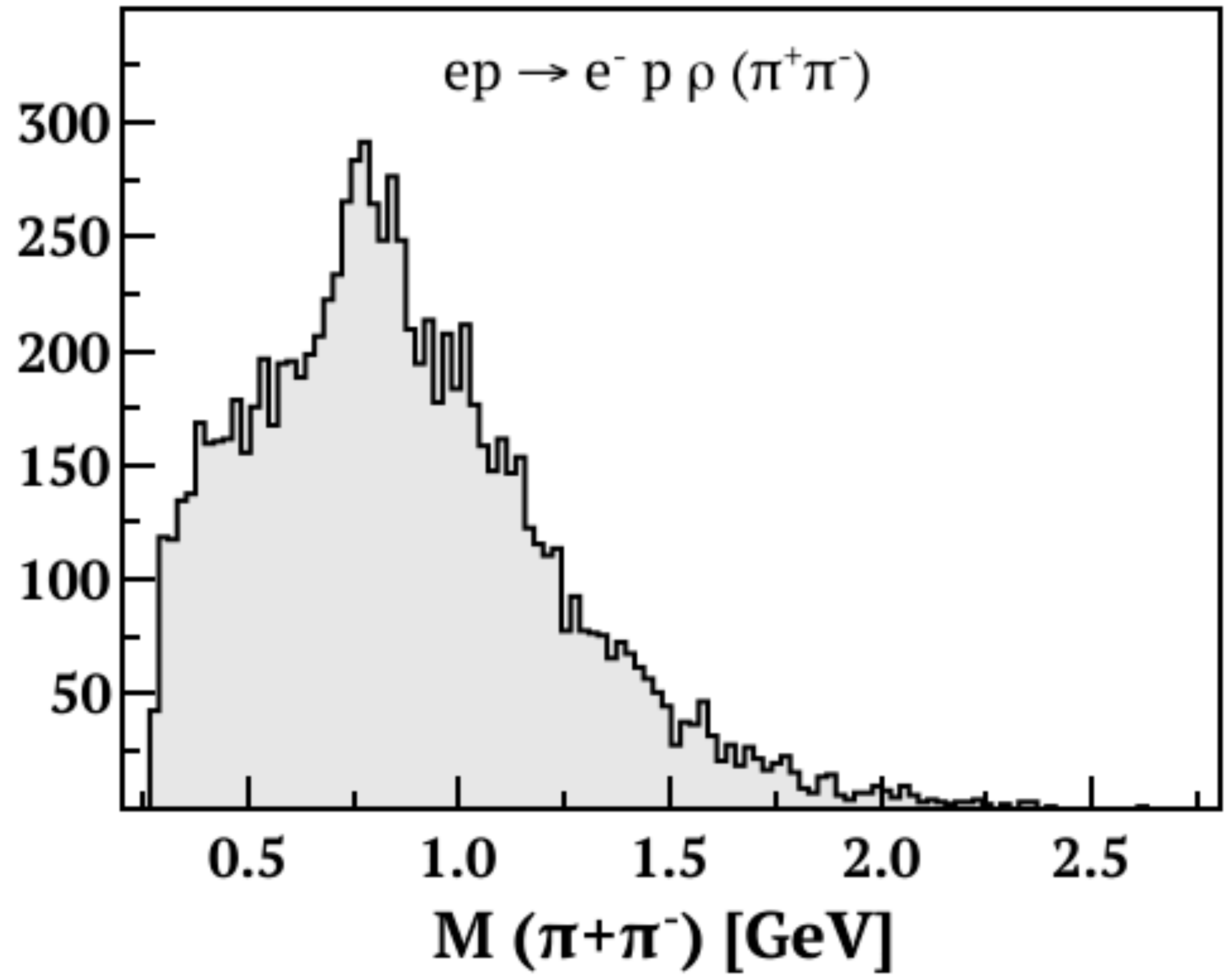


NEW WAY

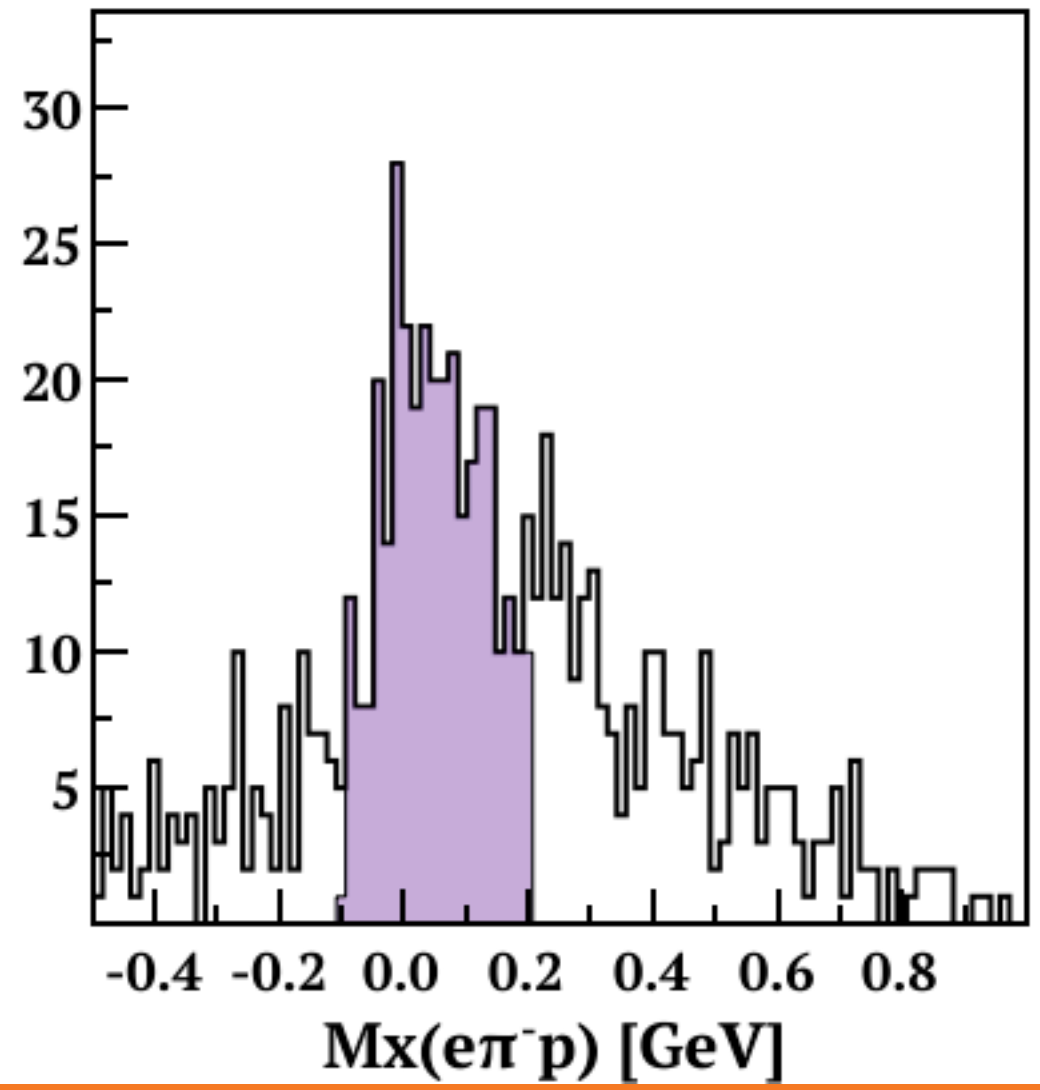


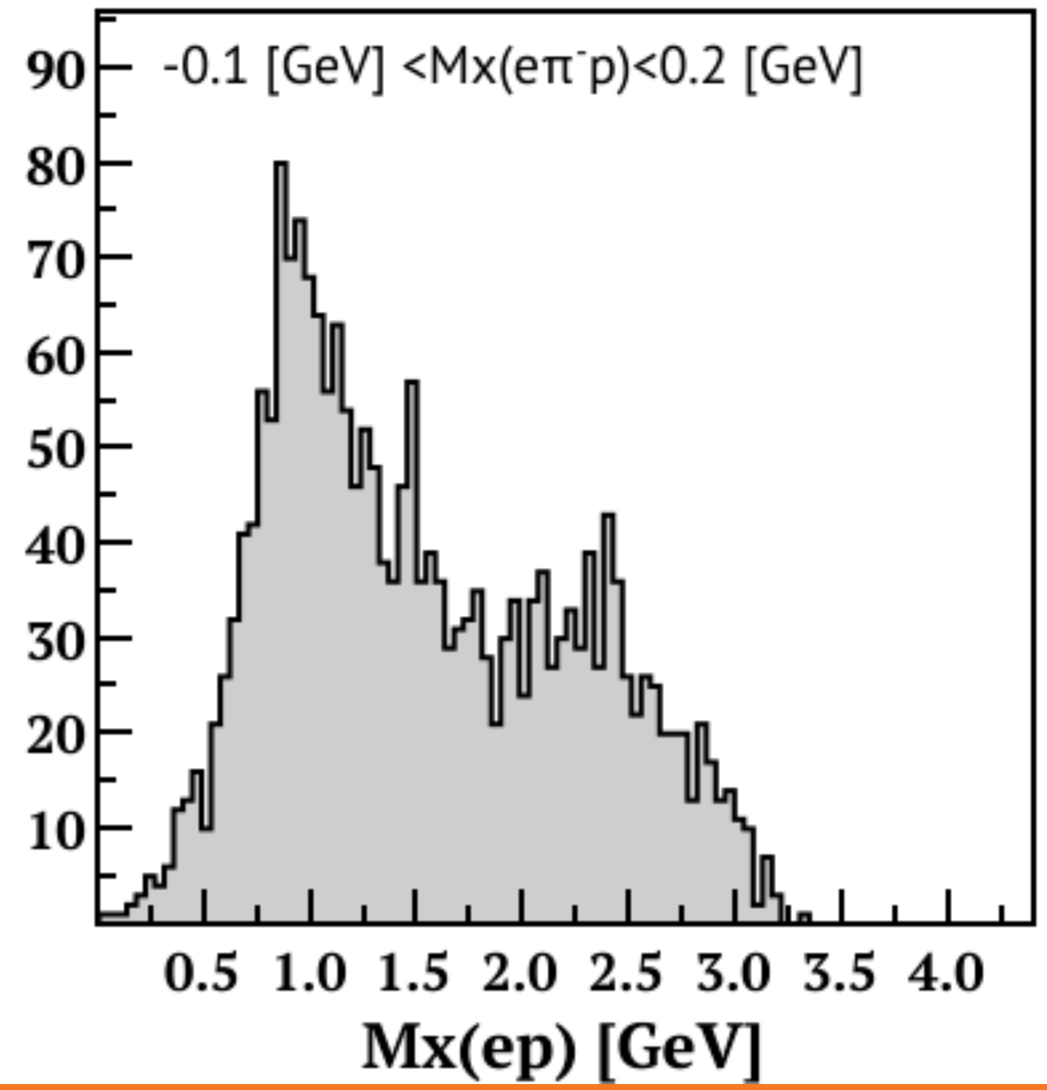
- **Online Reconstruction**
  - **Combines all neural networks developed for tracking**
  - **Includes newly developed clustering Convolutional Logistic Regression network**





- **Efficiency/Precision**
  - **Reconstruct final states by decayed particles**
  - **Capable of processing data at a rate of 42 kHz**

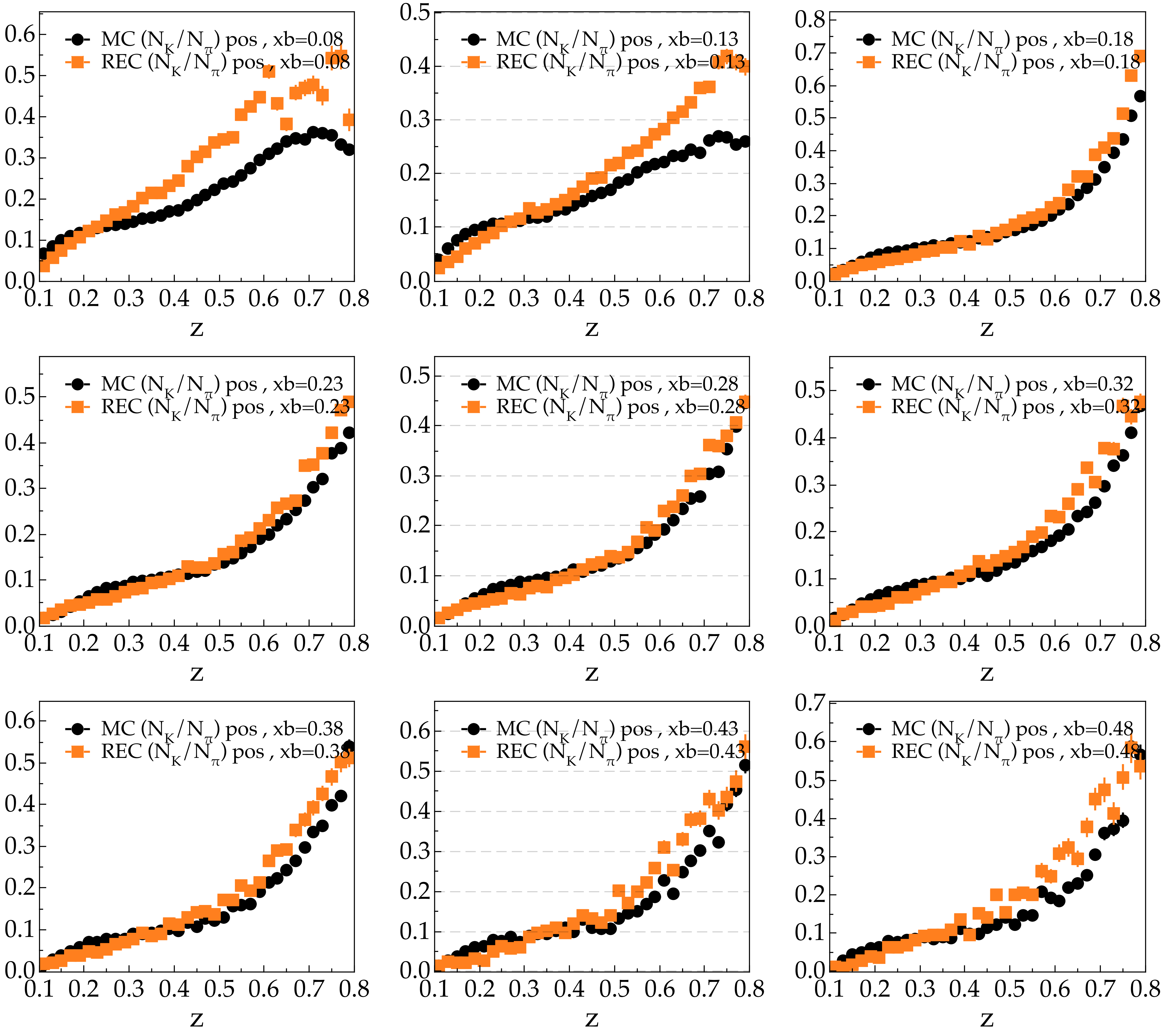


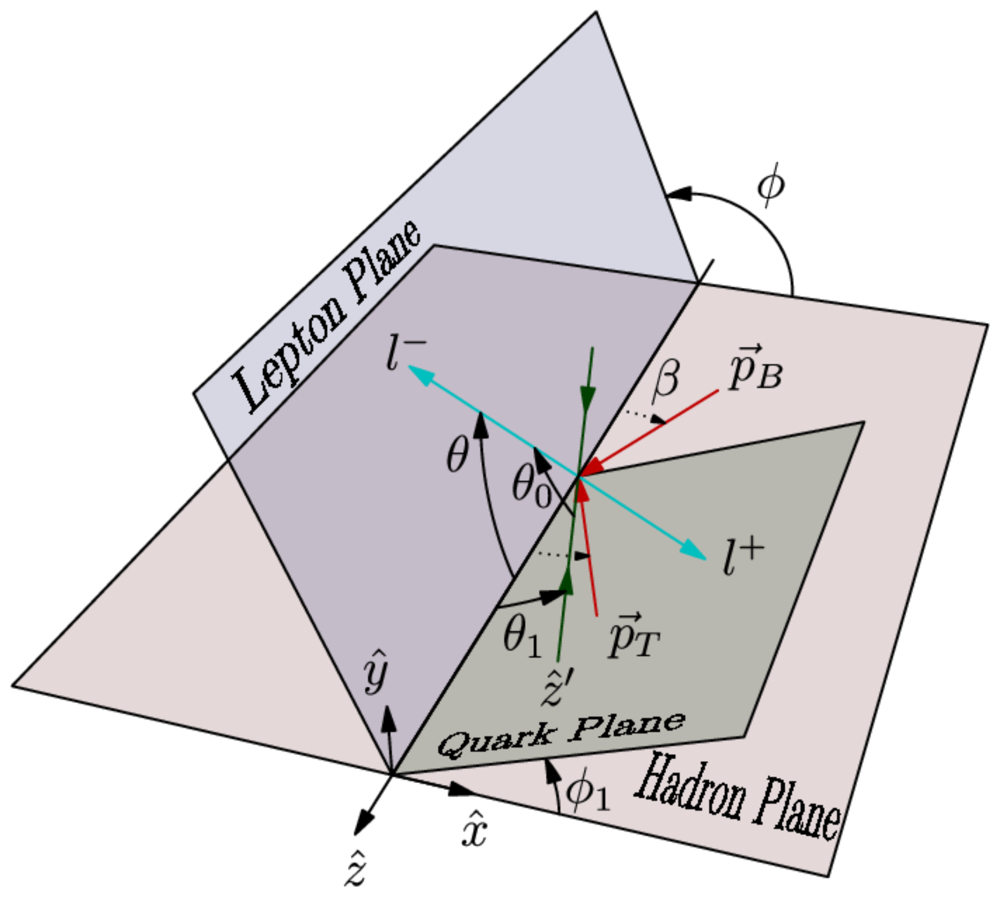


Physics Analysis

- **Particle Identification using AI/ML methods:**
  - extends the kinematics range for measurements.
- **Reaction identification with AI:**
  - reduction of background and clean signal extraction
- **Theoretical Model Validation/Extraction using AI/ML:**
  - Physics Model parameter extraction from experimental data
- **Simulations Model/Calculations using AI:**
  - Increase speed of reaction simulation needed to rigorously train Neural Networks

Physics Analysis/Particle Identification (TMDs)



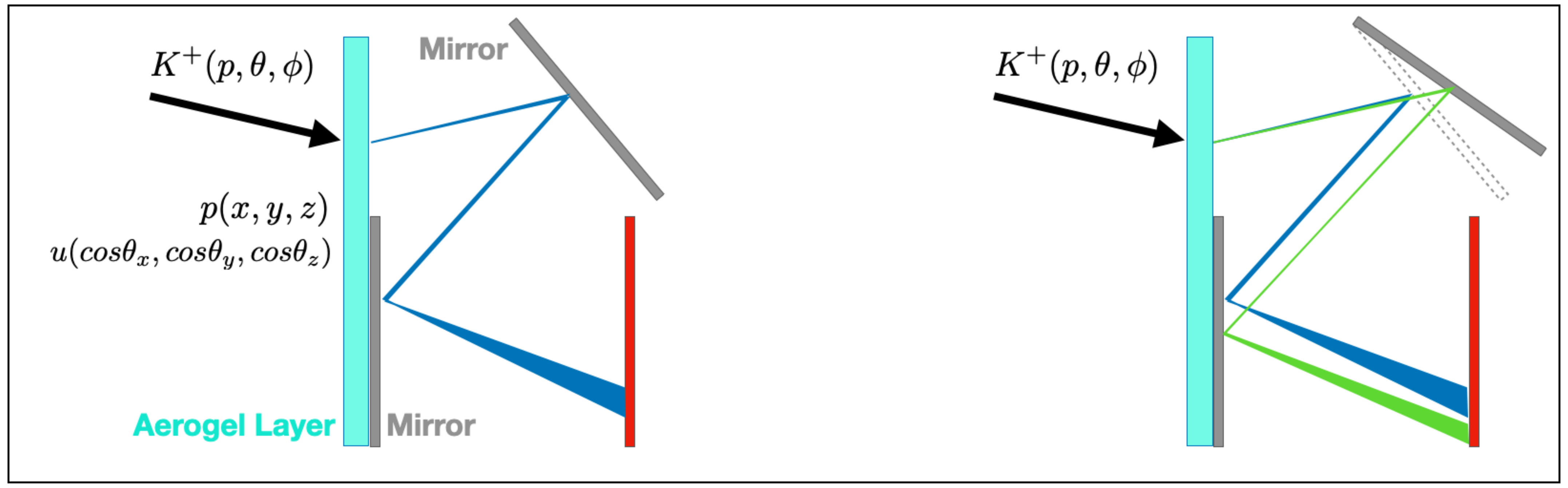


	U	L	T
Quarks	γ^+	$\gamma^+\gamma^5$	$i\sigma^{i+}\gamma^5$
U	$\boldsymbol{f_1}$		$h_1^\perp$
L		$\boldsymbol{g_1}$	$h_{1L}^\perp$
T	$f_{1T}^\perp$	g_{1T}	$\boldsymbol{h_1}$, $h_{1T}^\perp$
LL	$\boldsymbol{f_{1LL}}$		$h_{1LL}^\perp$
LT	f_{1LT}	g_{1LT}	h_{1LT}, $h_{1LT}^\perp$
TT	f_{1TT}	g_{1TT}	h_{1TT},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

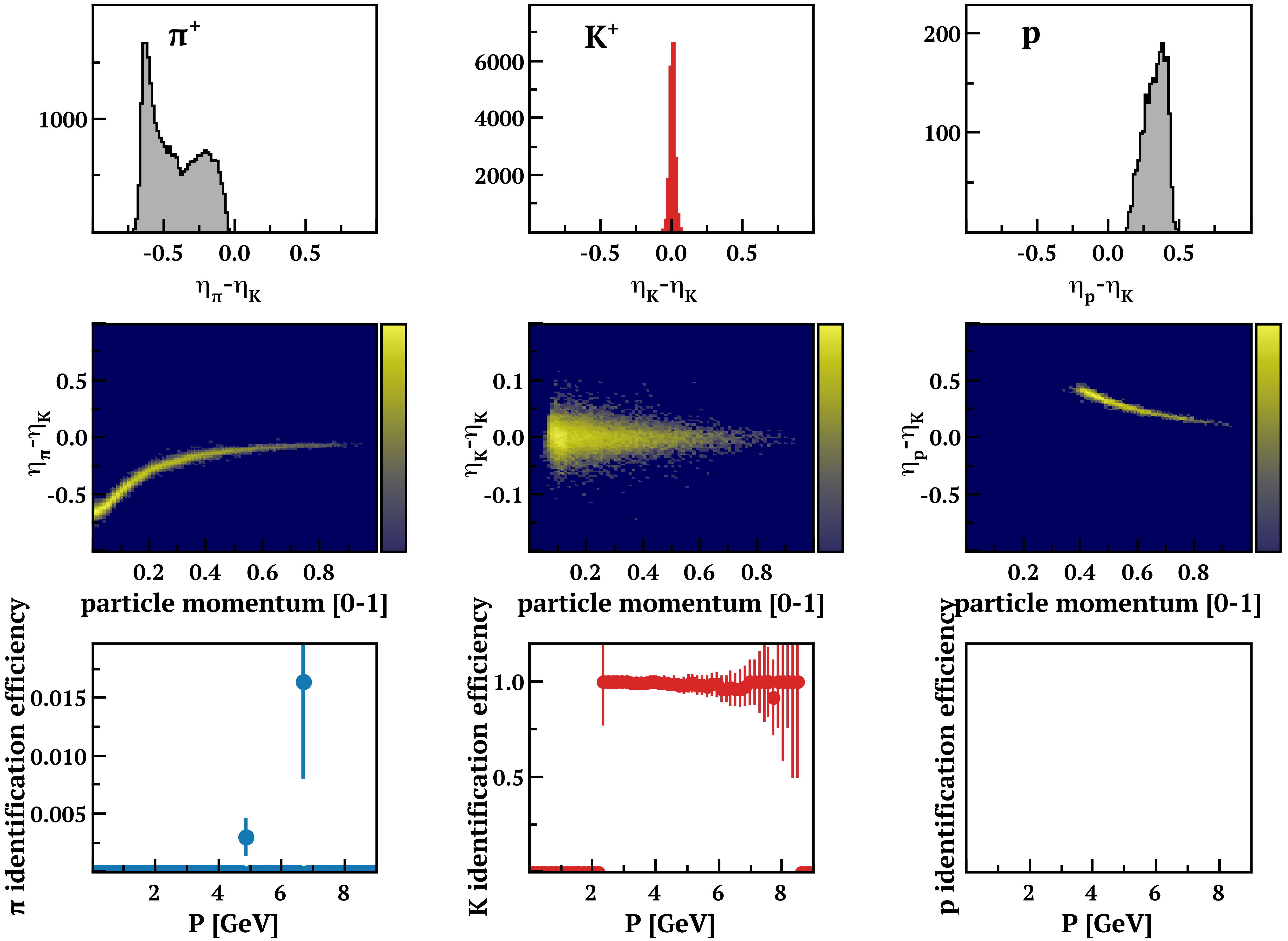
- Traditional (time-of-flight) can effectively separate pi/K up to 3.5 GeV
- For full measurement of hadron multiplicities as a function of z and Pt need to separate hadrons at higher momenta to measure:
  - Hadron multiplicities
  - Single Spin Asymmetries (SSA)
  - Double Spin Asymmetries
- Map fragmentation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

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

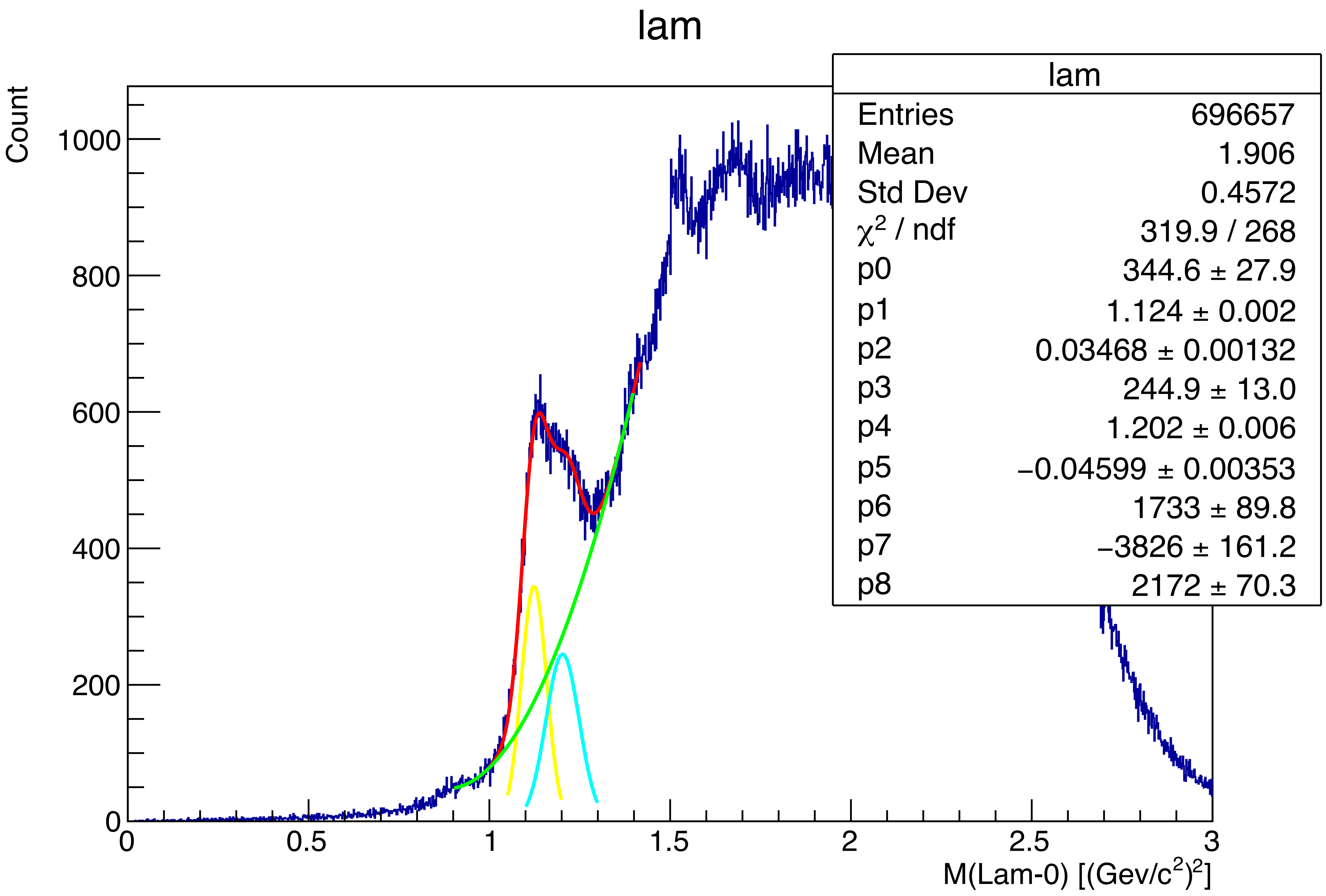
Marco Mirazita, Armen Gyurjinyan (INFN)



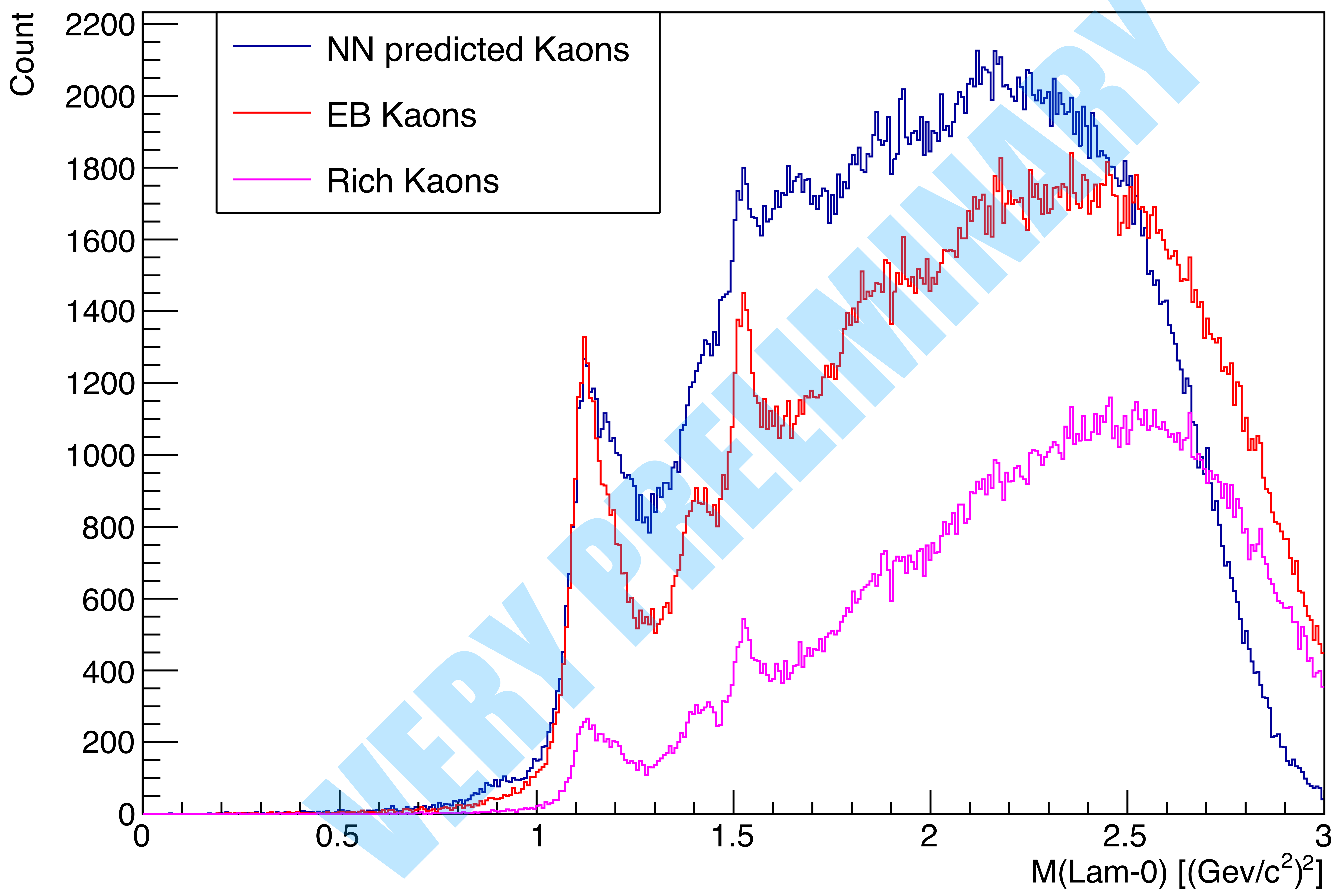
- **RICH Ideal Geometry**
  - If the ideal geometry and position of mirrors are known the ray-tracing can help recover the Cherenkov angle
  - Calculating the Cherenkov angle for each of the hits on the photomultiplier plane allows to identify the particles.
- **RICH Real World Geometry**
  - Ray tracing will predict an inaccurate position for the hit on the detector plane
  - This affects the efficiency of particle identification



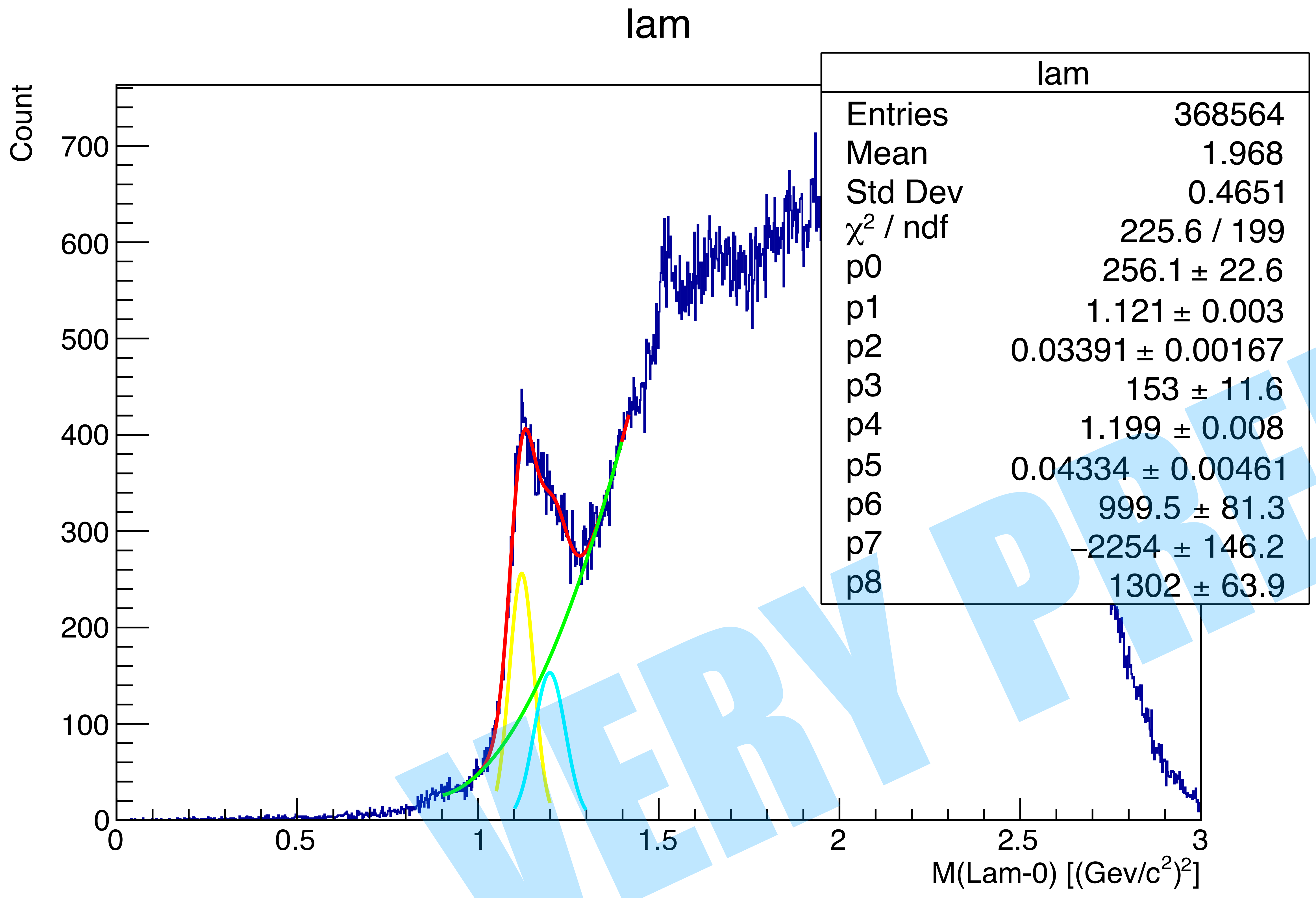
Marco Mirazita, Armen Gyurjinyan (INFN)

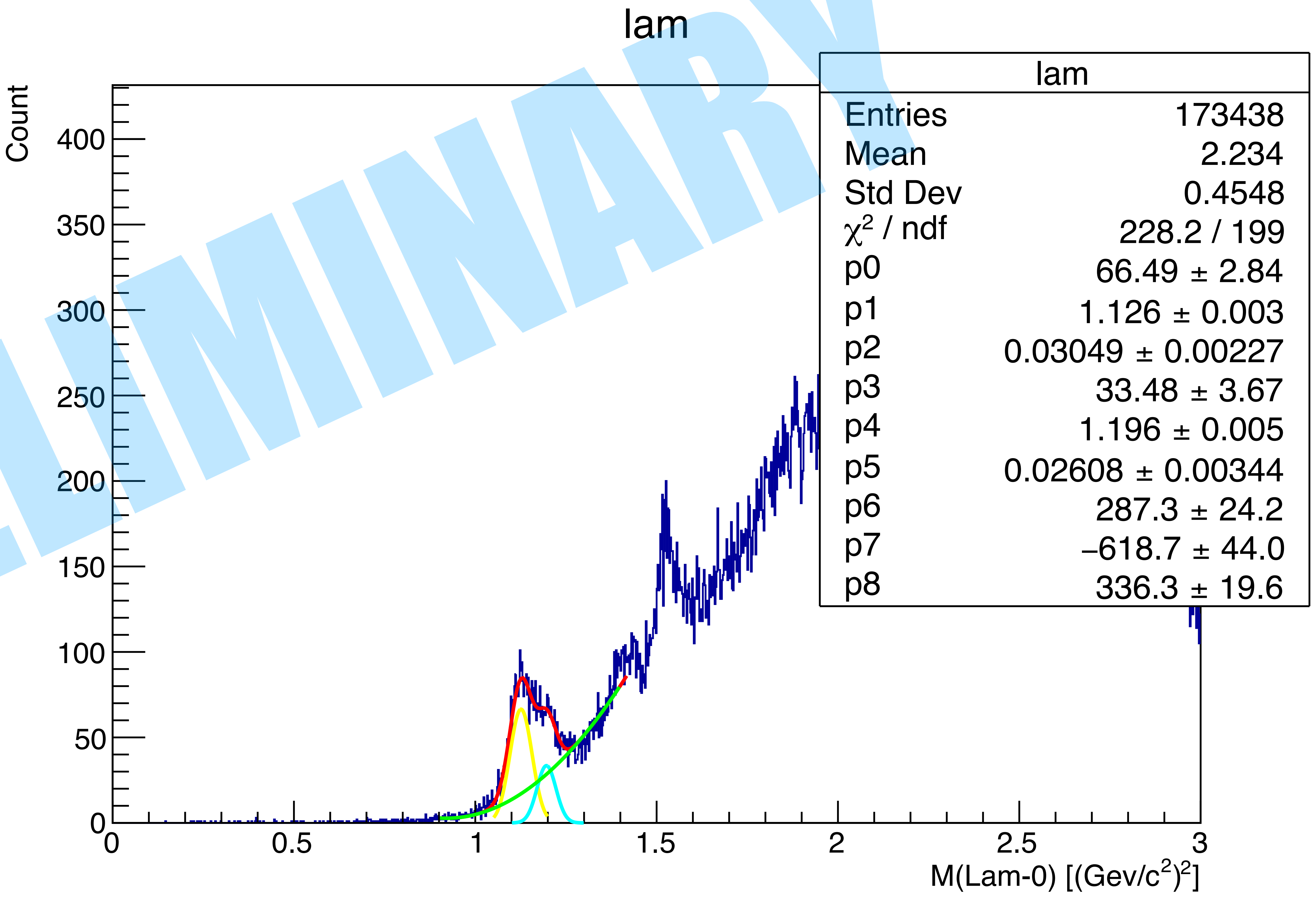


Marco Mirazita, Armen Gyurjinyan (INFN)



Marco Mirazita, Armen Gyurjinyan (INFN)

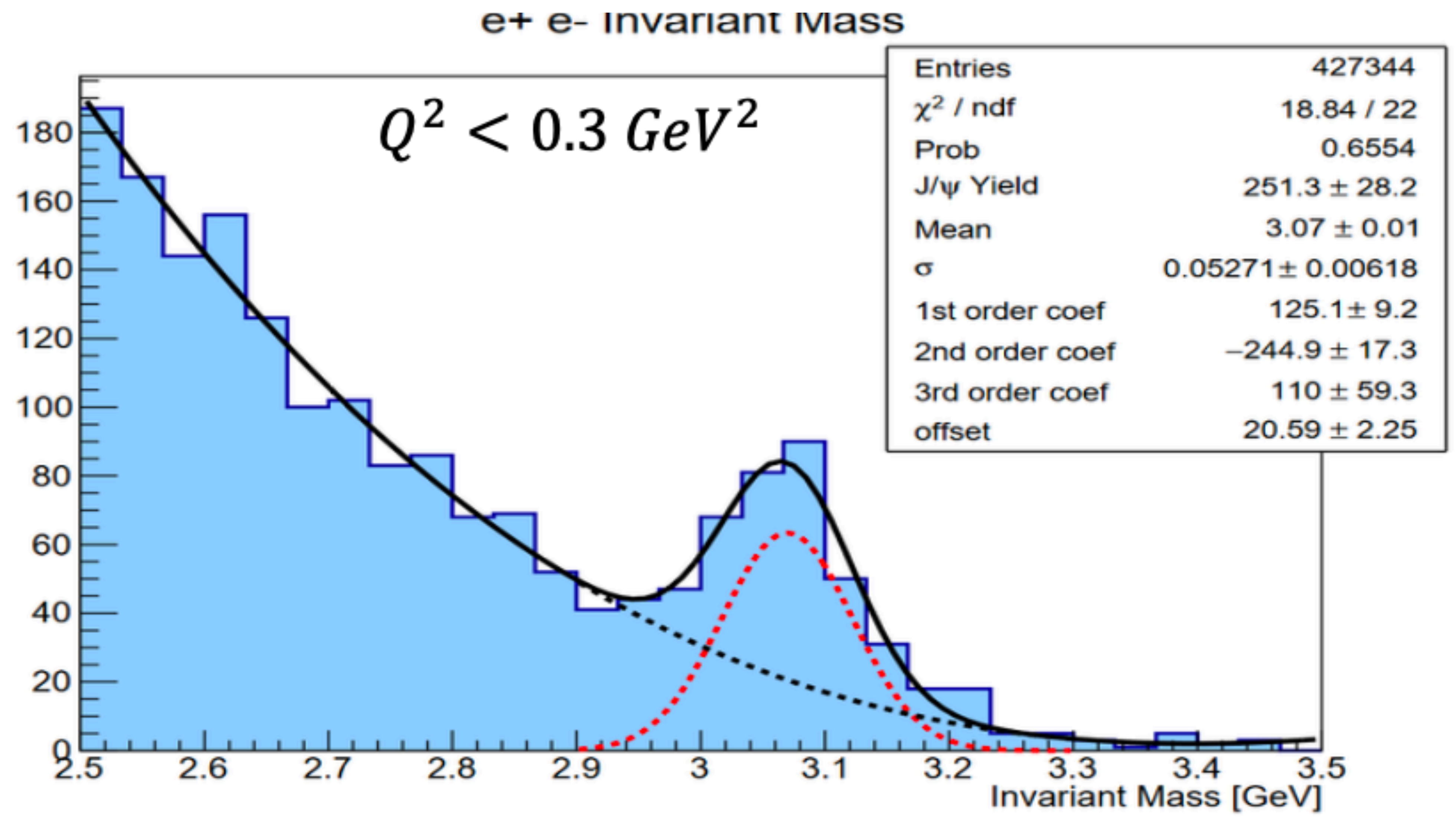


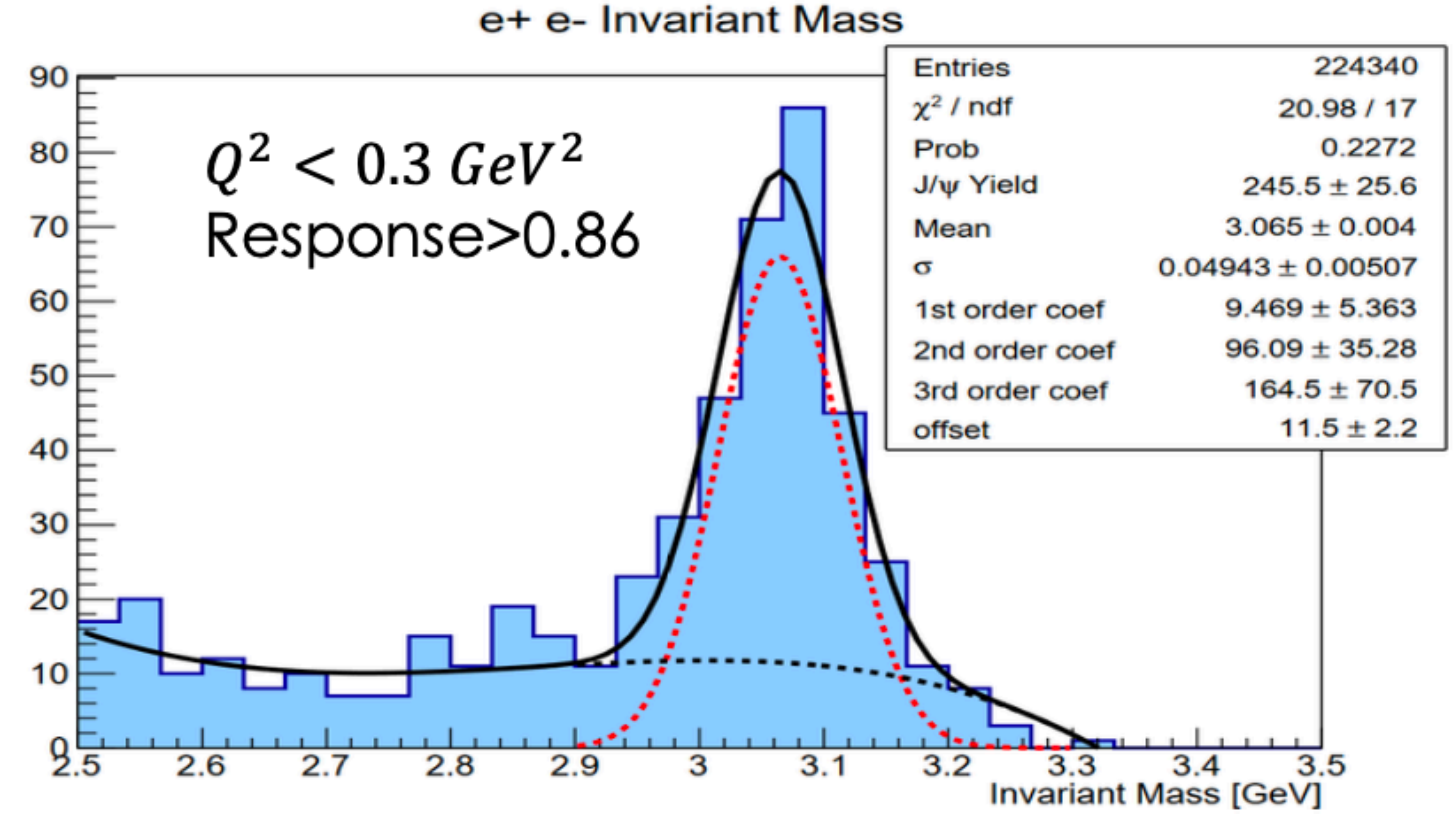


Reaction Identification

Richard Tyson (Glasgow)

- Change training samples when using a deuterium target, or different beam energies (10.2 GeV and 10.6 GeV in spring 2019 RG-B).
- Simulated deuterium target with elSpectro event generator.
- We apply cuts on $Q2$ to produce a clean sample. Most J/ψ past these cuts are retained by the classifier.
- Currently working on the publication which will include many more reaction examples.





Observable Extraction

$$\frac{d\sigma}{dxdyd\phi dzd\phi_h dP_\perp^2} = \frac{\alpha^2}{xyQ^2}\frac{y^2}{2(1-\epsilon)}(1+\frac{\gamma^2}{2x}) \times K$$

$$f_1(x) = (1-x)^{C_1} x^{C_2}$$

$$D_1(z) = C_3(1-z)^{C_4}$$

$$P_T = z^2 < k_T > + < P_T >$$

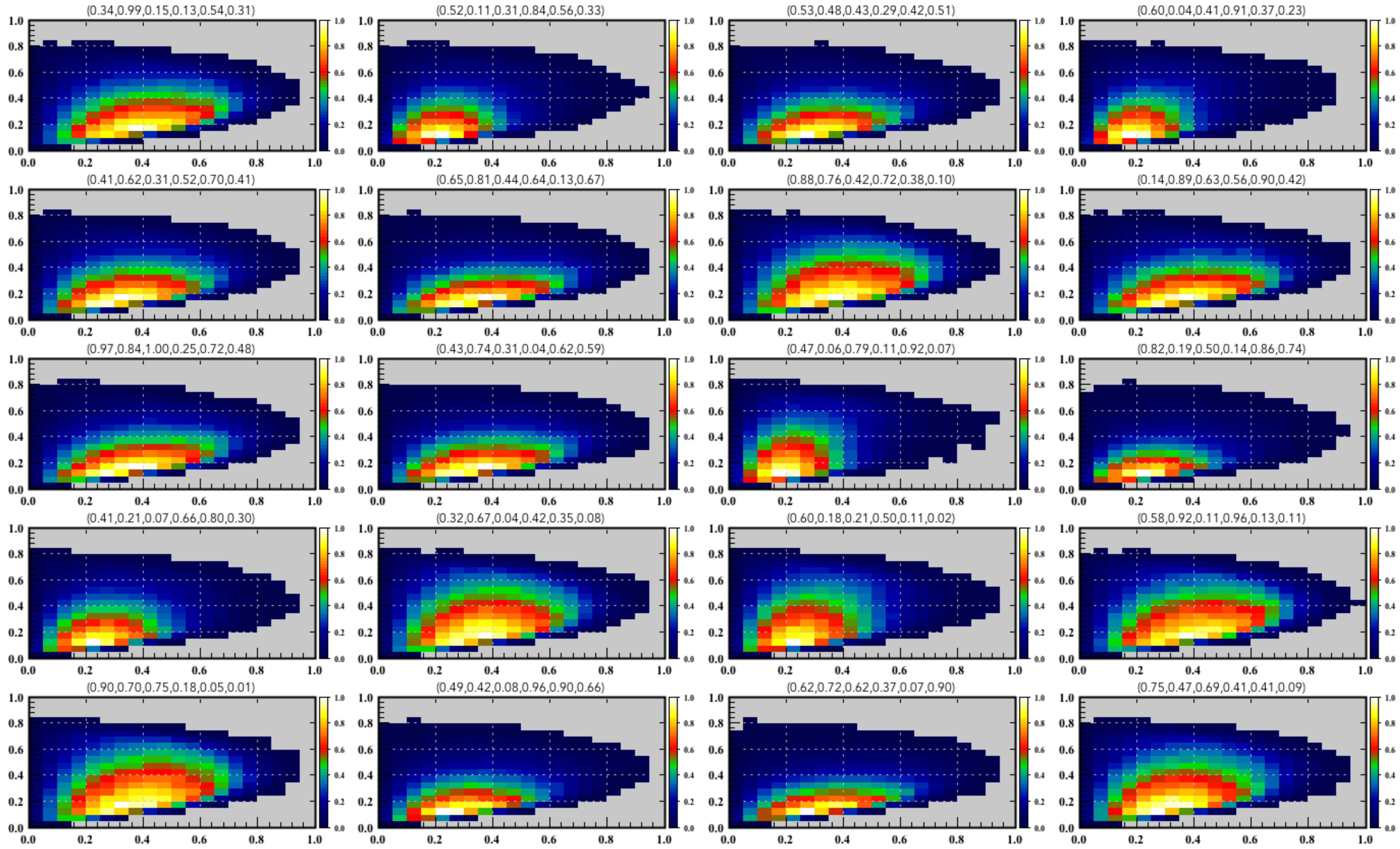
$$K = (F_{UU,T} + \epsilon F_{UU,L} + \sqrt{2\epsilon}(1+\epsilon)\cos\phi_h F_{UU}^{\cos\phi_h} + \epsilon\cos(2\phi_h) F_{UU}^{\cos 2\phi_h})$$

$$F_{UU,T} = x f_1(x) D_1(z) \frac{1}{\pi P_T} e^{-\frac{p_T^2}{P_T}}$$

- **Modeling functions with parameters**
- **Each parameter has a defined range**

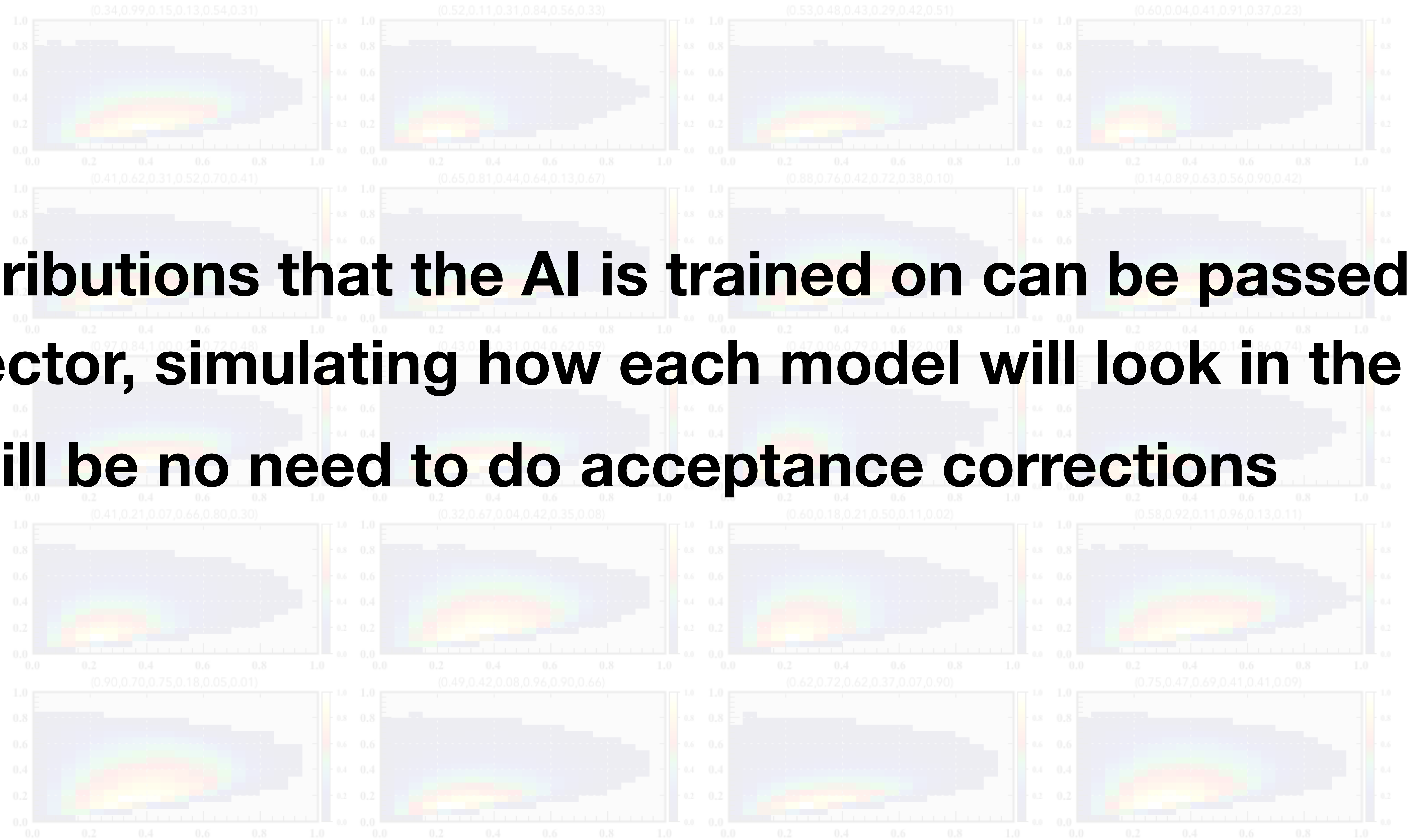
P	Value	Min	Max
<KT>	0.33	0.05	0.4
<PT>	0.16	0.05	0.4
C1	3.00	2.00	5.00
C2	-1.313	-2.00	-1.00
C3	0.80	0.50	1.00
C4	2.00	1.00	4.00

- Z vs PT for different combinations of parameters C1,C2 …



- Inclusive pion production

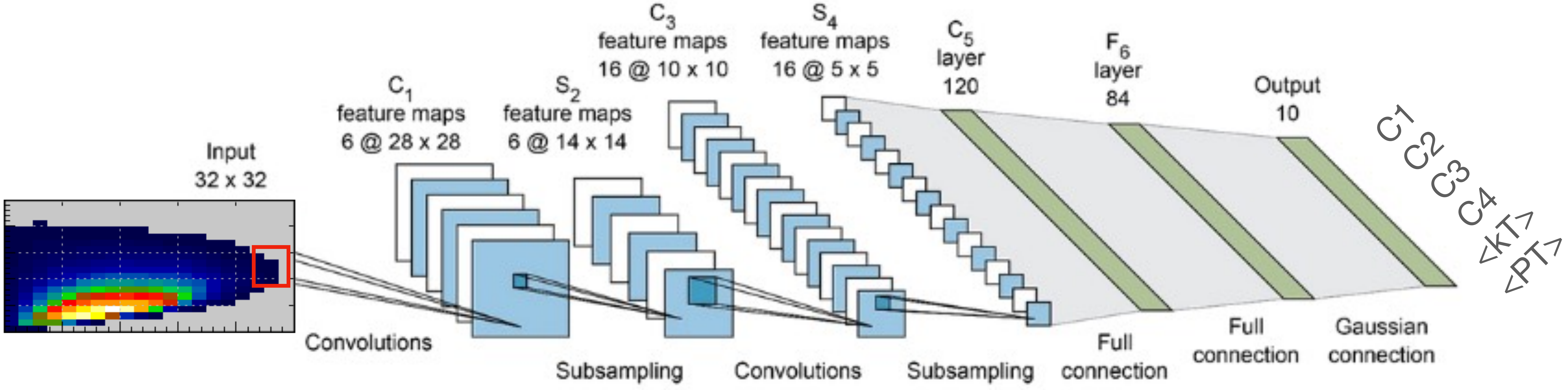
- Z vs PT for different combinations of parameters C1,C2 …

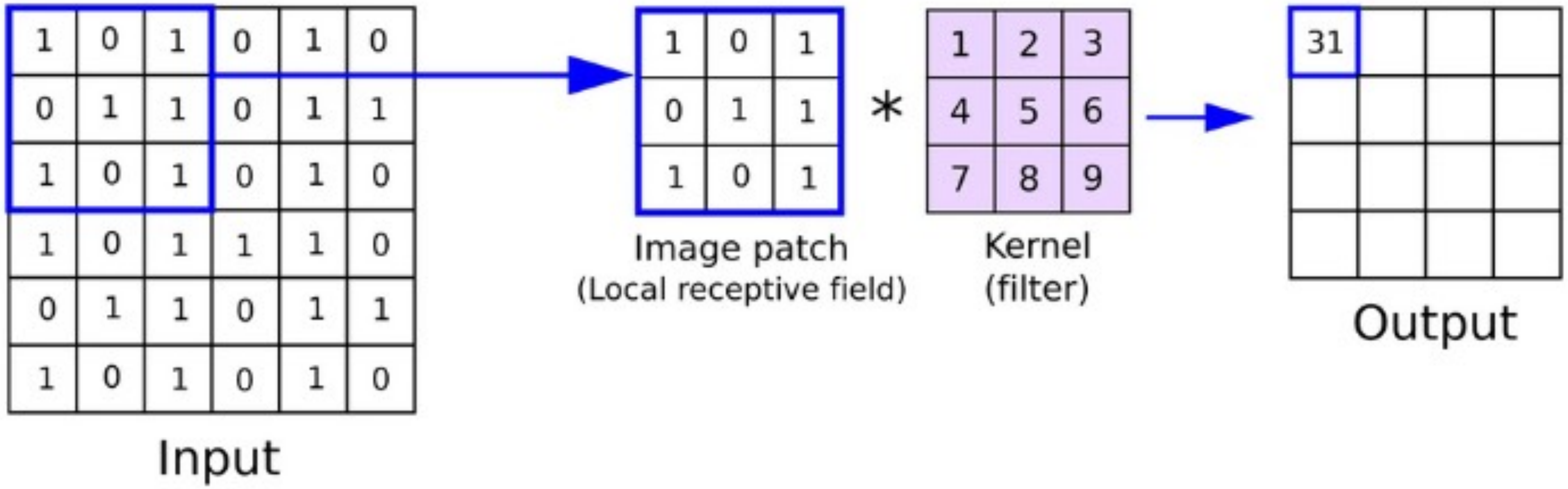


- The distributions that the AI is trained on can be passed through the detector, simulating how each model will look in the detector.
- There will be no need to do acceptance corrections

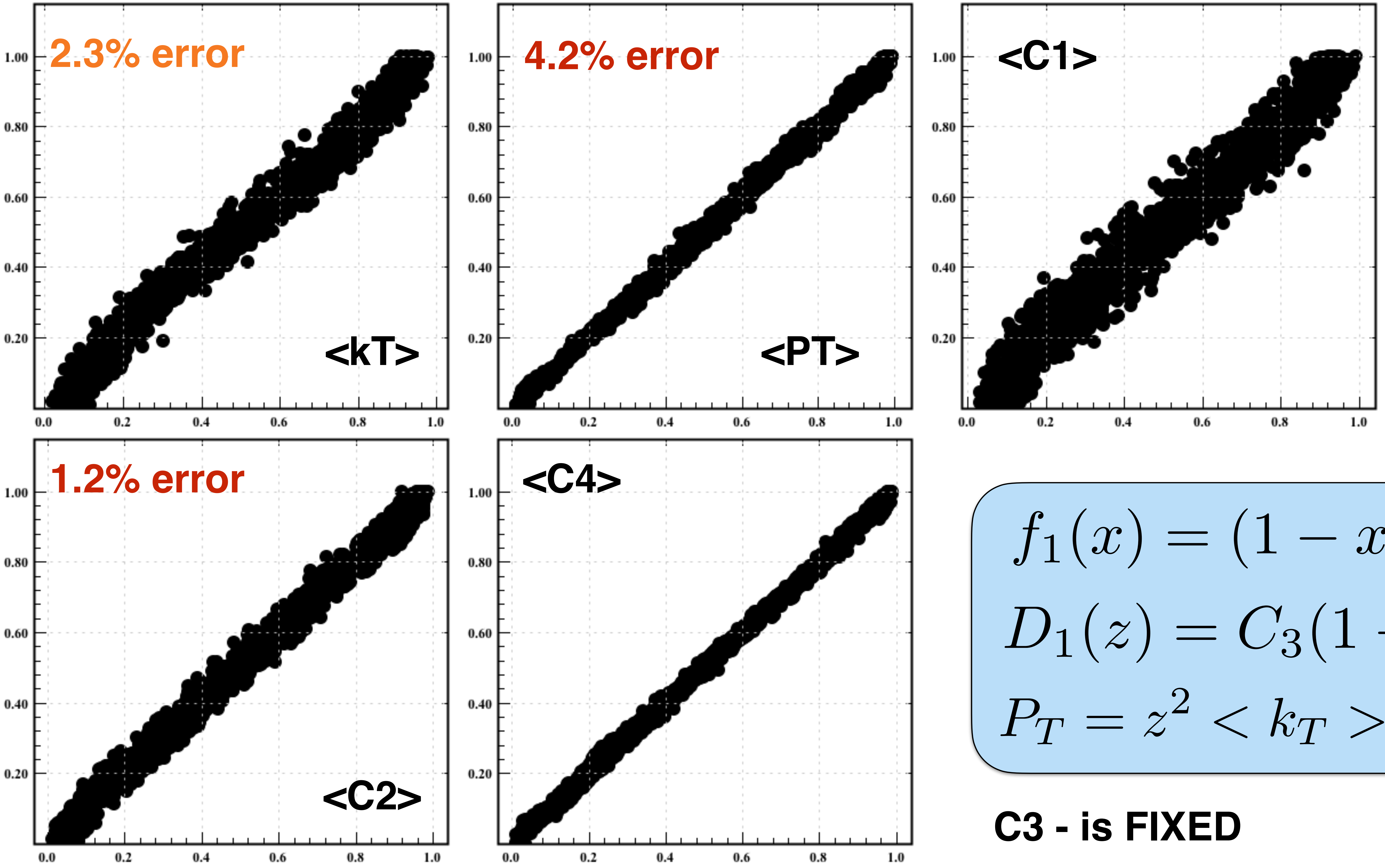
- Inclusive pion production

Observable Extraction





Observable Extraction

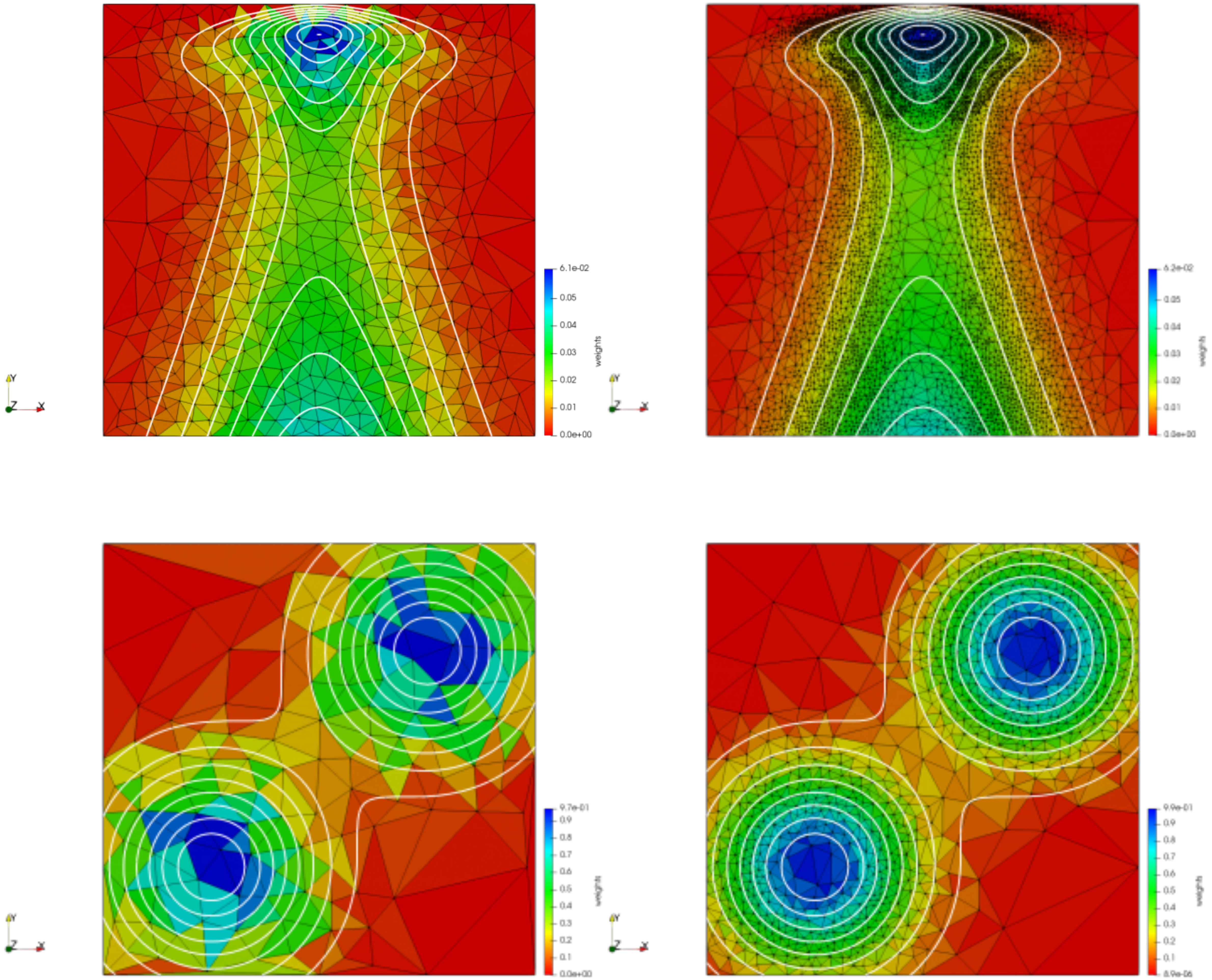


$$f_1(x) = (1-x)^{C_1} x^{C_2}$$

$$D_1(z) = C_3(1-z)^{C_4}$$

$$P_T = z^2 < k_T > + < P_T >$$

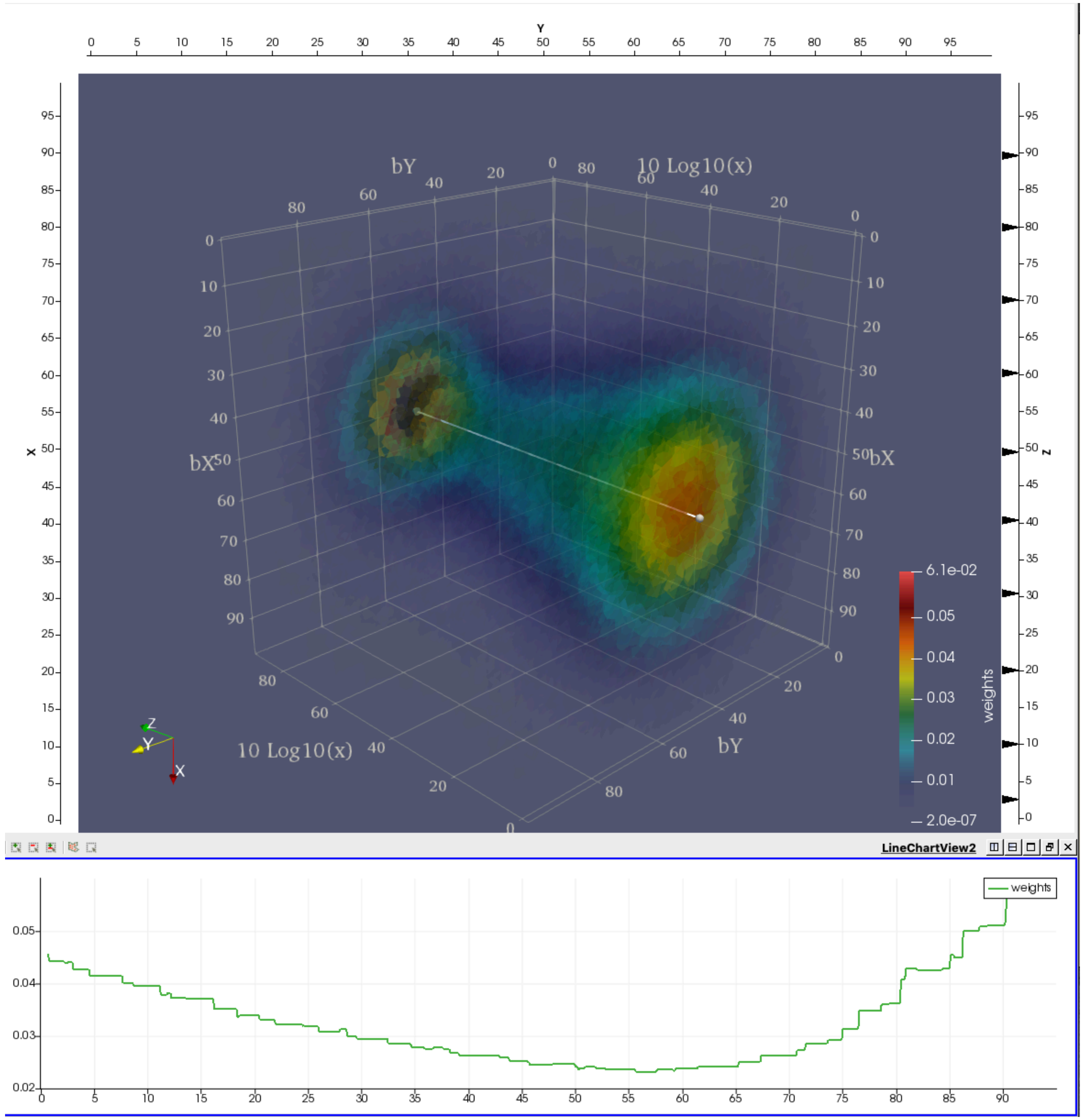
C3 - is FIXED



- **Adaptive Mesh generation**
  - Pre-calculated grids are passed through tessellation software.
  - Adaptive element size is used based on the gradient change from cell to cell.
  - The resulting mesh can be used to generate random numbers much faster than can be done with the original code.
- **Advantages**
  - Less memory is required to store the tessellated mesh.
  - Moving to higher dimensions memory requirement is increasing linearly compared to O(n).
  - Mesh can be used for visualization and analysis using industry-standard software (like. Paraview)

Simulations

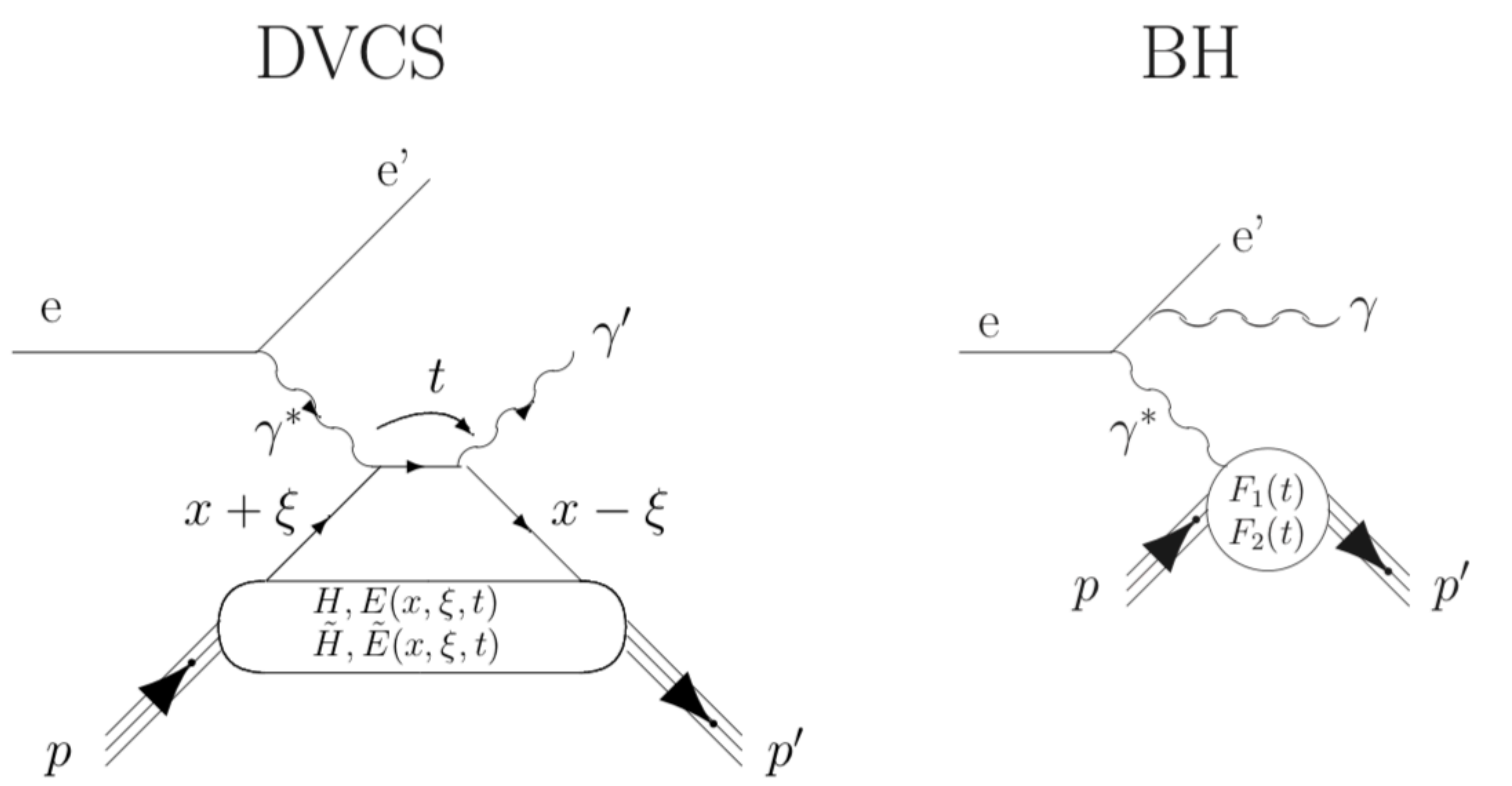
- PARTONS software was used to generate u-quark density distribution as a function of impact parameters.
- Generated GRID was processed with tessellation software to produce 3-D mesh for ParaView.
- **Tessellated Mesh in ParaView:**
  - sliced view of distribution with intensity color map
  - plot distribution along any line in space.
  - plot integrals for any slice and projection of density distribution.
- **Using meshes for simulation:**
  - tessellated objects can be used to generate random numbers following density distribution.
  - the process is very fast compared to calculating convoluted integrals numerically (in the case of DVCS cross-sections)
  - tessellated physics models can be shared with experimental physicists for easy particle simulations.

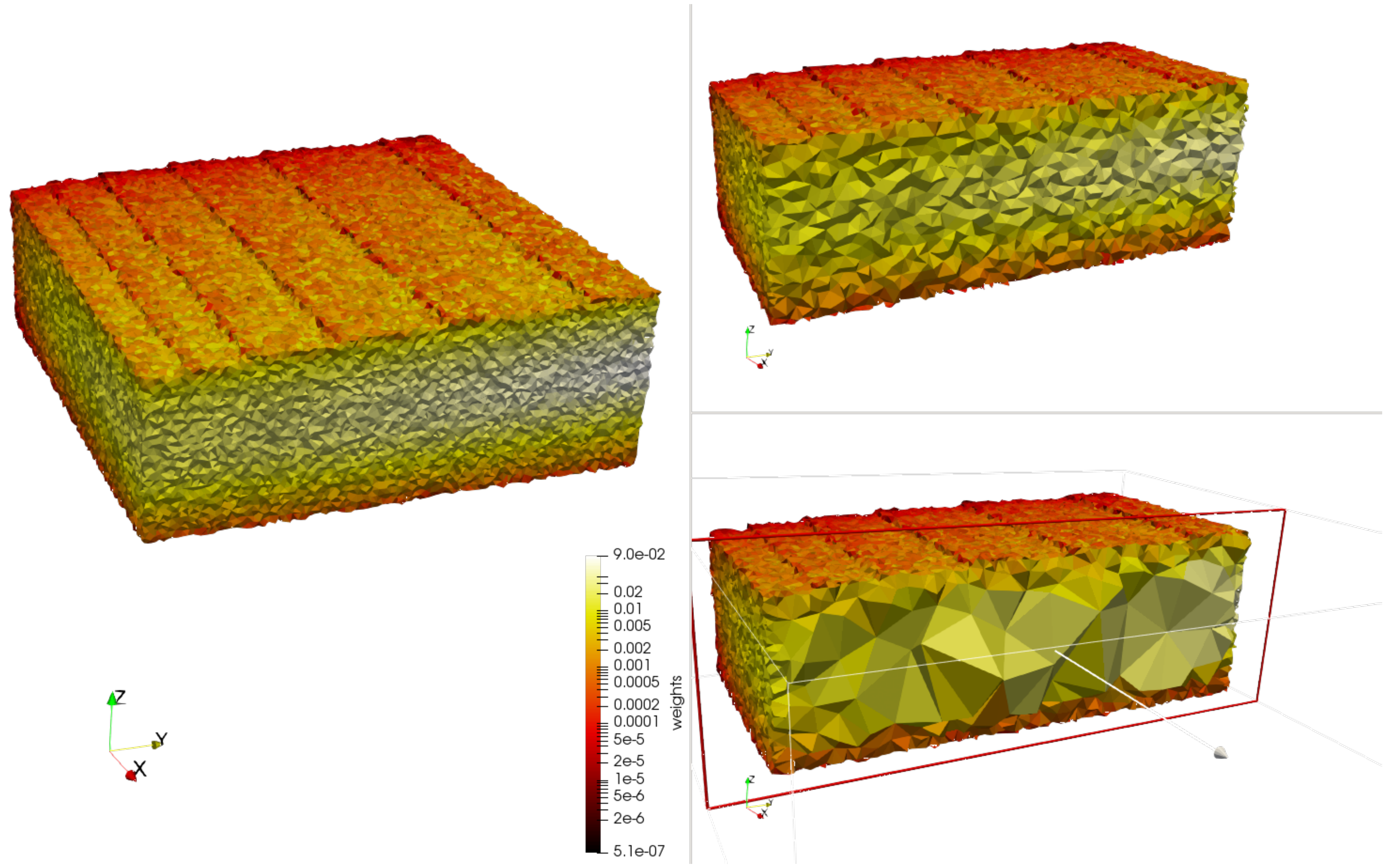


- PARTONS software was used to generate Deeply Virtual Compton Scattering cross sections as a function of Q2, x, and phi.
- This process probes the internal structure of the proton, to learn about quark orbital momentum inside the protons.
- Generated GRID was processed with tessellation software to produce 3-D mesh for ParaView.

- **Tessellated Mesh in ParaView:**
  - sliced view of distribution with intensity color map
  - plot distribution along any line in space.
  - plot integrals for any slice and projection of density distribution.

- **Using meshes for simulation:**
  - tessellated objects can be used to generate random numbers following density distribution.
  - the process is very fast compared to calculating convoluted integrals numerically (in the case of DVCS cross-sections)
  - tessellated physics models can be shared with experimental physicists for easy particle simulations.





Simulating DVCS events is 1000 times faster with Meshes

Summary

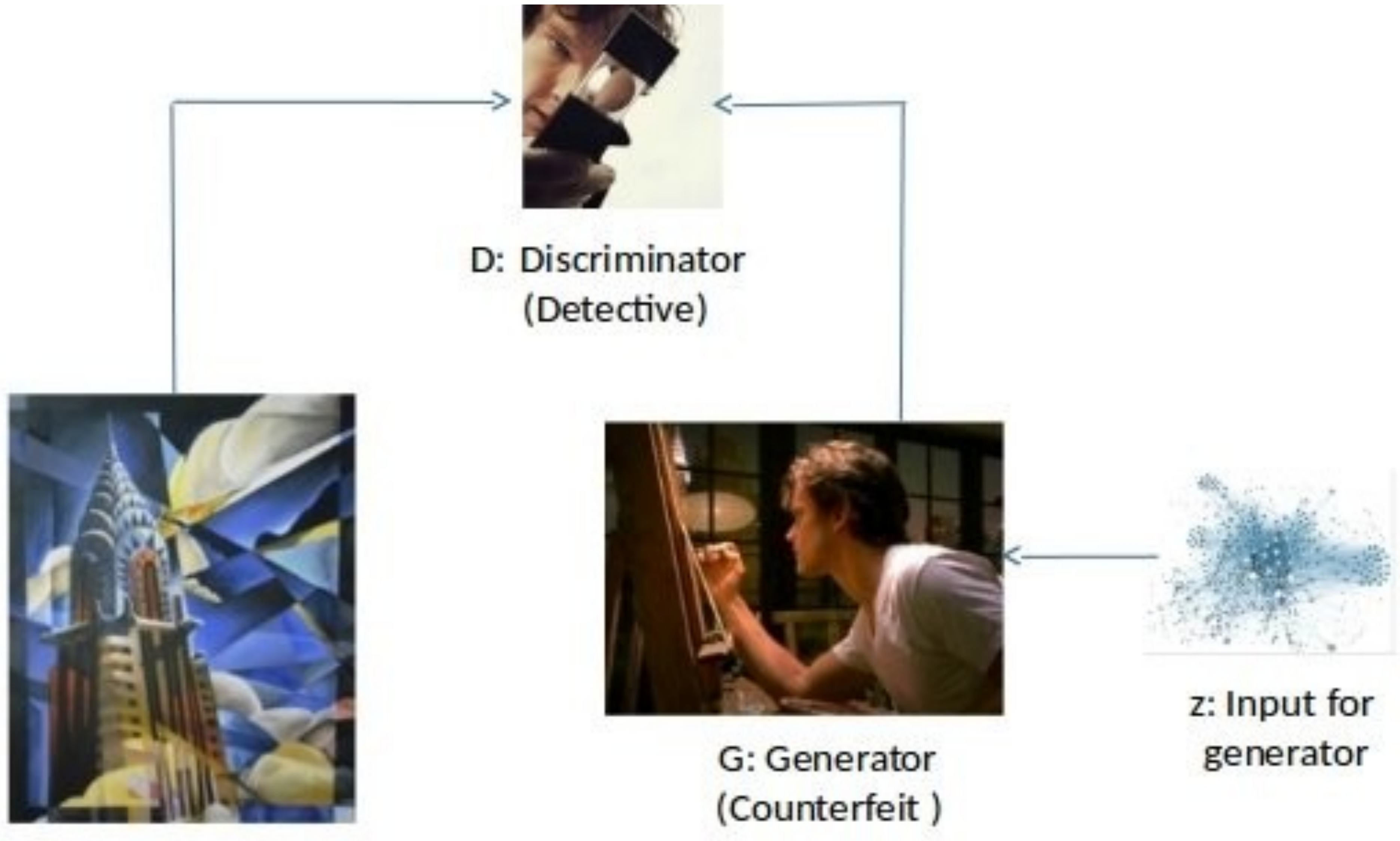
- AI/ML tracking provides significant improvements in physics yield for completed experiments in CLAS12.
- Based on previous networks a real-time workflow is developed allowing to identify final states during data acquisition (at the DAQ rate). Many uses for this kind of network:
  - Triggering specific reactions
  - Skimming data based on physics reactions
- Physics analysis can benefit from using AI/ML methods in areas of particle identification, reaction isolation, observable inference, and simulations.
- Unifying efforts and sharing tools will definitely benefit the collaboration.

CLAS12 is on the frontiers of using AI/ML in Nuclear Physics Experiments

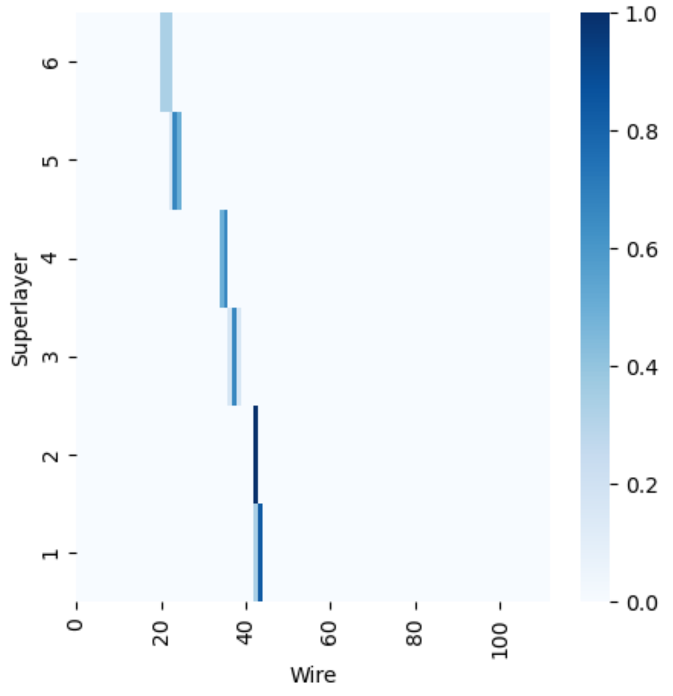
Backup Slides

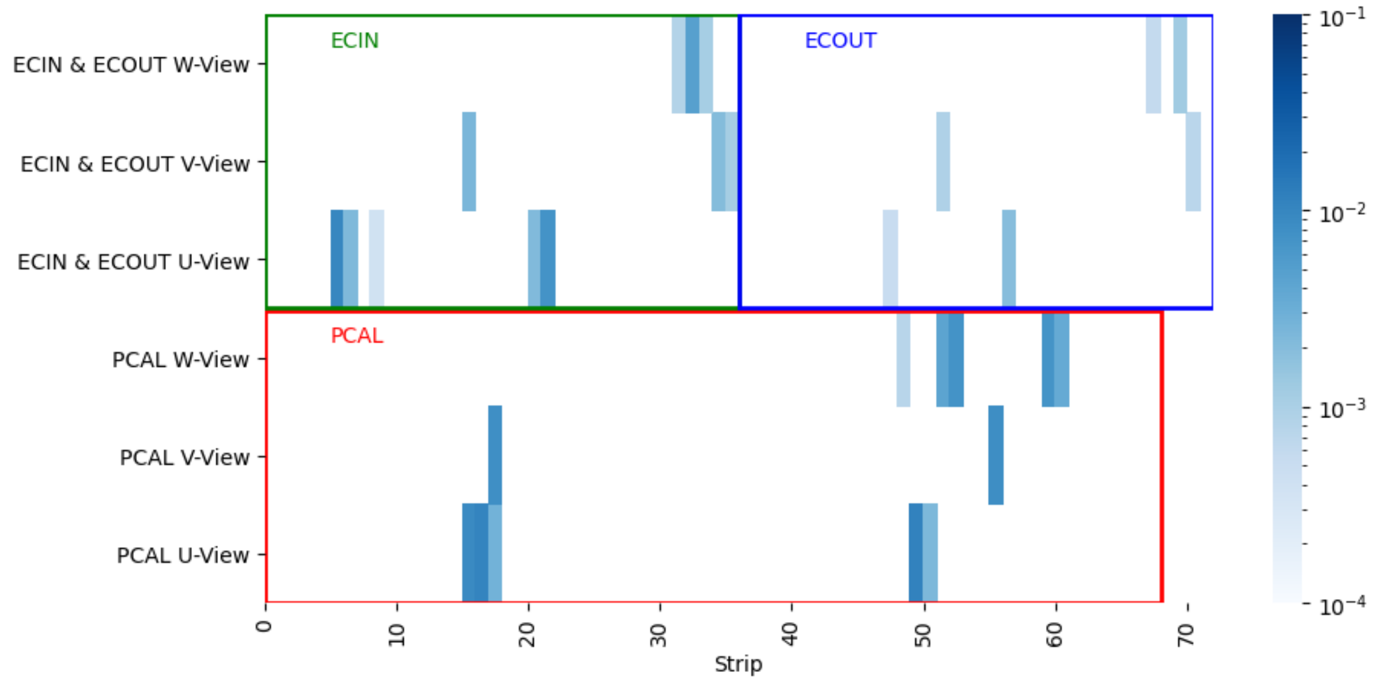
DALL.E
Open AI

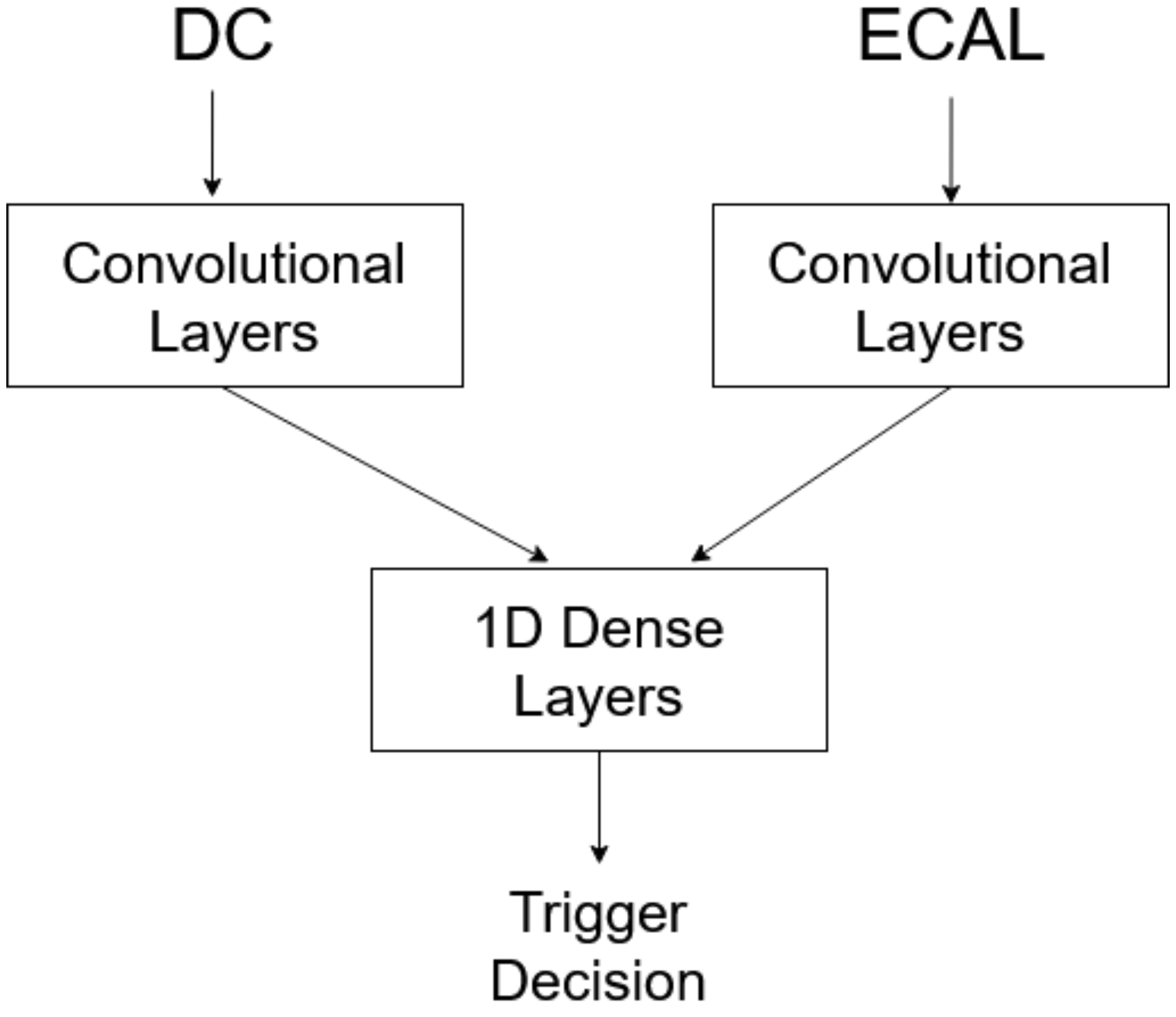
- Image Generation:
  - AI tools to generate images based on the description
  - Ability to generate images with the style of a certain painter



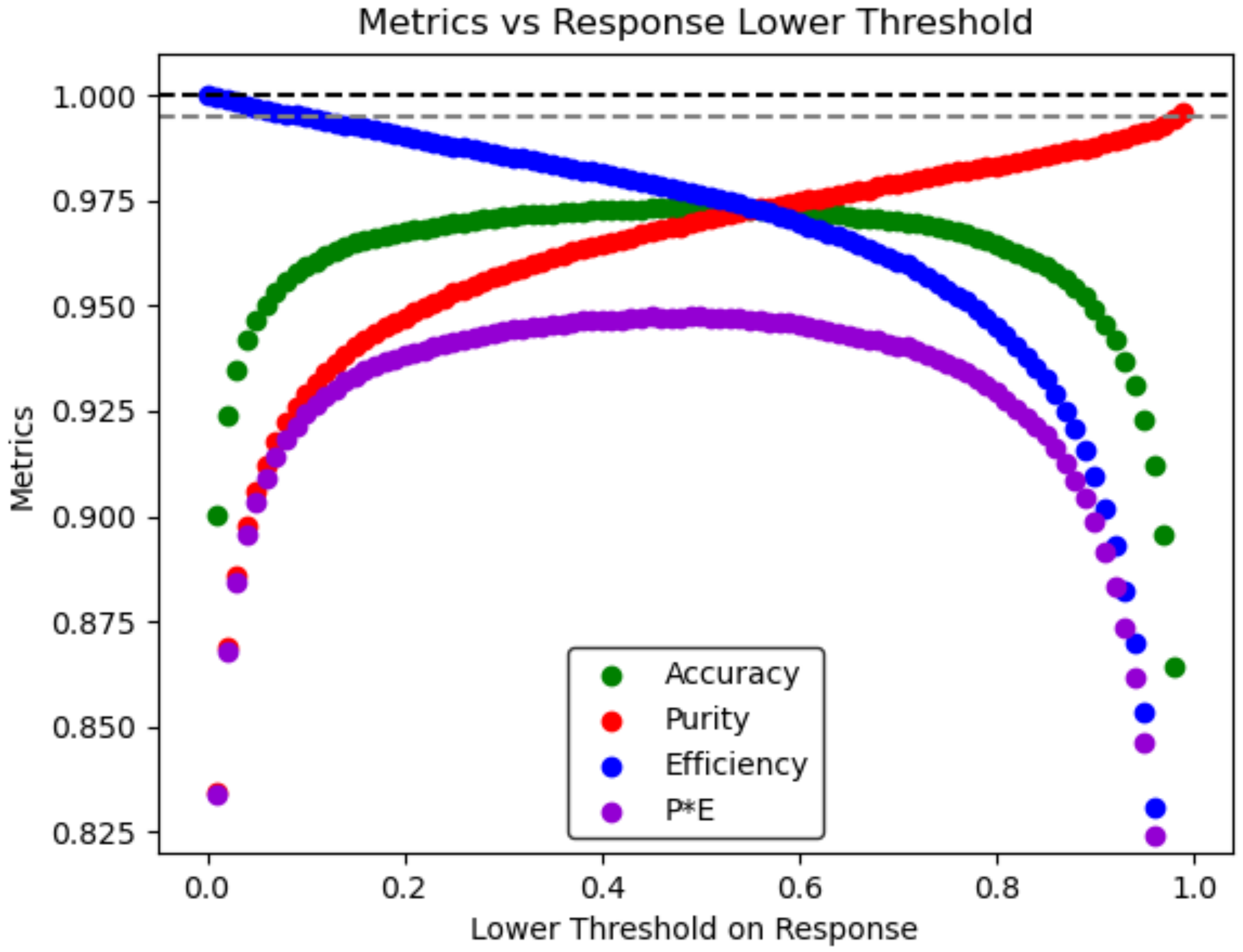
Level-3 Trigger





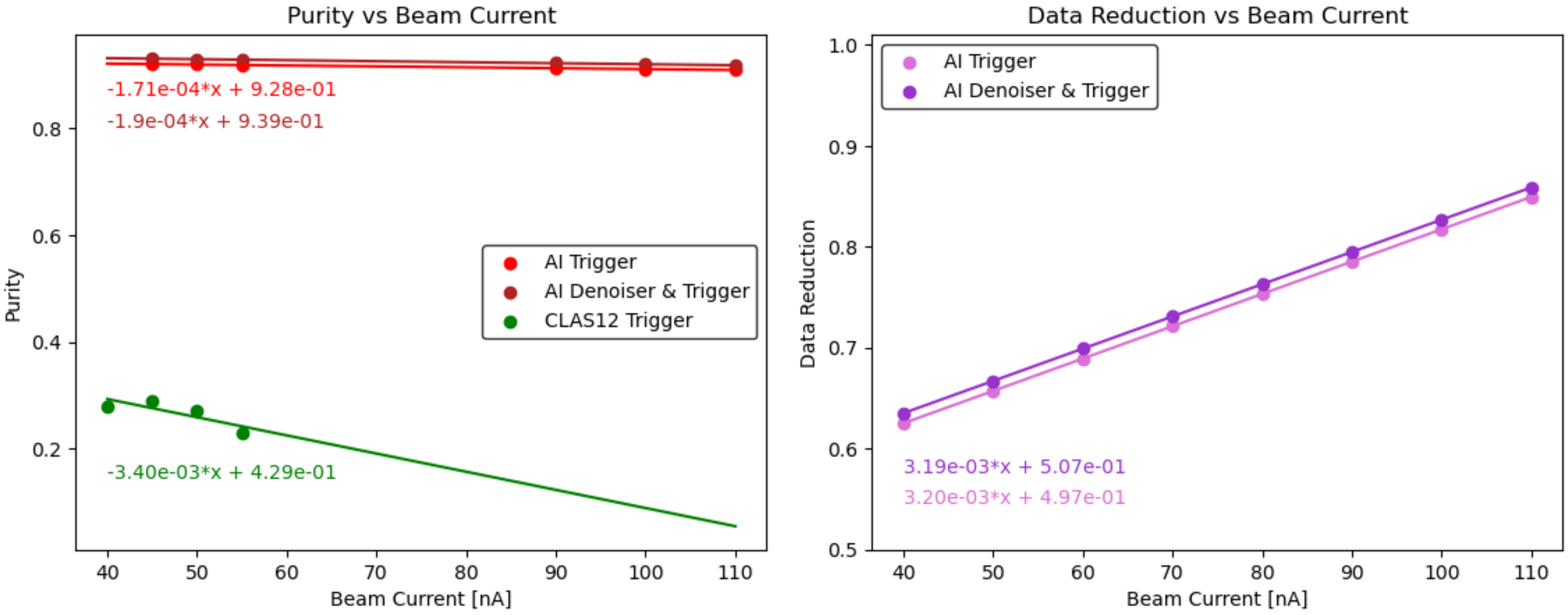


Threshold	Purity	Efficiency	Accuracy
0.0012	0.841	0.9999	0.906
0.03	0.930	0.999	0.962
0.47	0.977	0.99	0.983



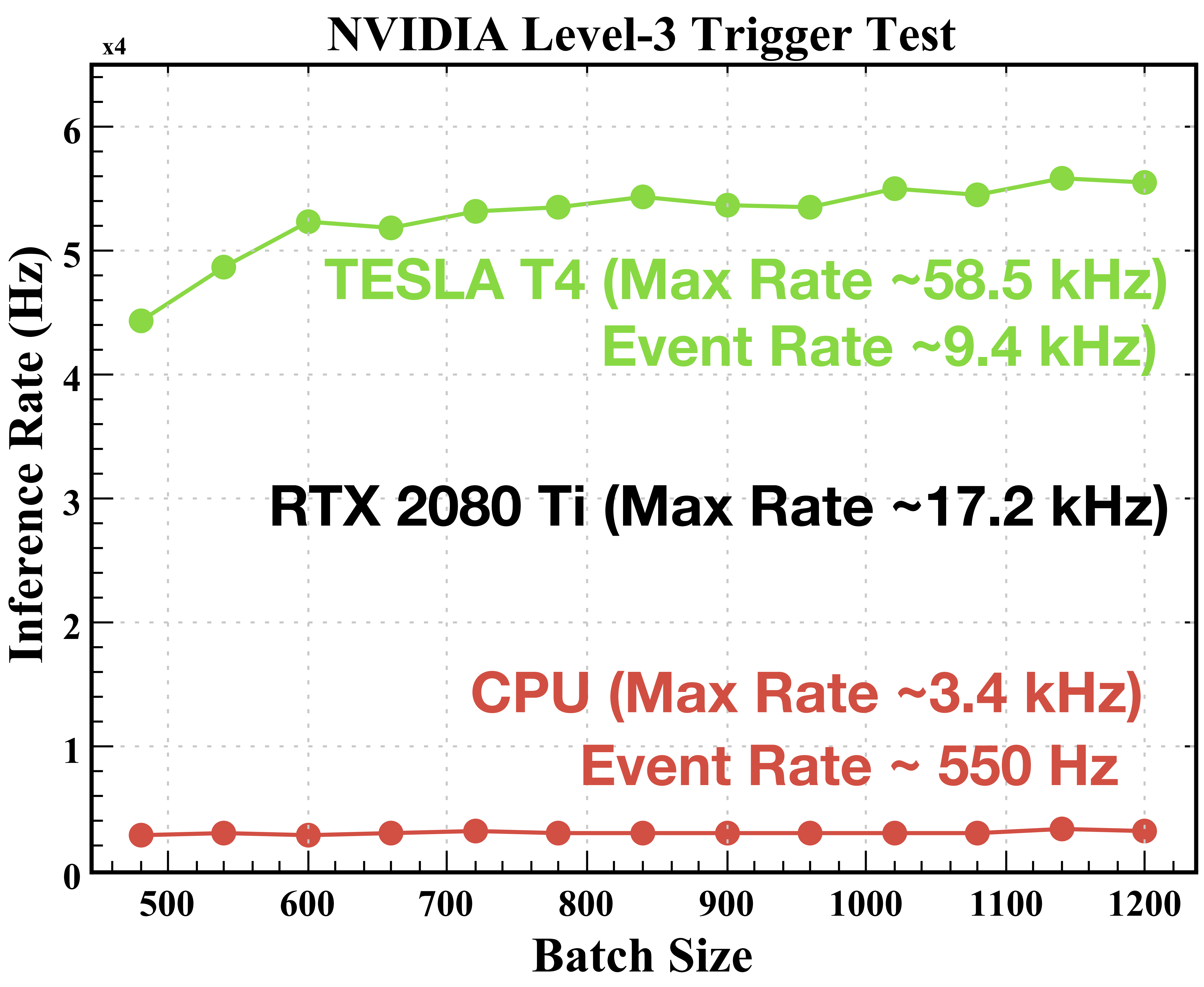
- Level-3 Trigger
  - A Convolutional Neural Network with a Computation graph is used to identify electrons.
  - The DC image is analyzed separately from the EC image, then combined to make a decision.
  - The ECIN, ECOUT, and PCAL are combined into one image 6x72
  - The current implementation does not use information from High-Threshold Cherenkov Detector

Level-3 Trigger Performance compared to conventional Trigger



Level-3 Trigger

- Neural Network was developed for Level-3 trigger studies. (Richard Tyson, University of Glasgow)
- The Software was tested on clonfarm11 node with two NVIDIA Tesla T4 GPUs (2 available, tested only on 1), over 3 times faster than RTX 2080 Ti
- Results are reported as inference per second (inference is per one sector)
- The real data rate is inference divided by 6
- Results are reported for 1 CPU core and 1 GPU unit

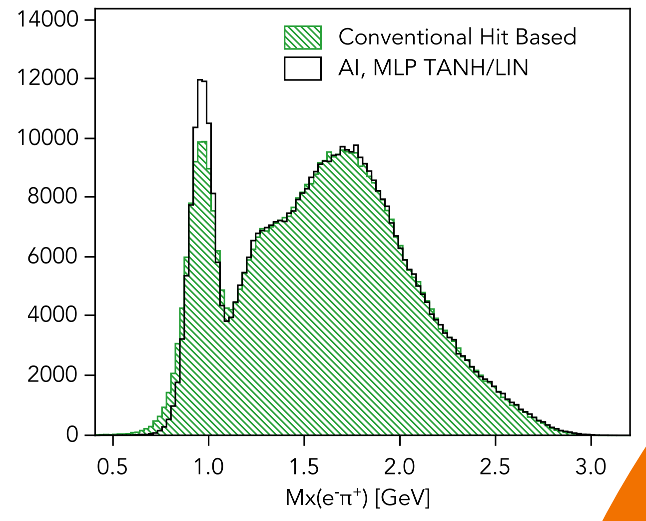


- Online multi-threaded data decoder into HIPO is implemented (C++)
- Currently contains only DC and ECAL decoding
- The ET-RING is set up to convert EVIO events into HIPO data frames (100 events per frame) and store HiPO frames in secondary ET-RING
- The Level-3 trigger will be tested during the next run
- With HiPO ET-RING we can now implement online data reconstruction (AI track reconstruction will be easy to add)
- Online data calibration is also possible

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

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

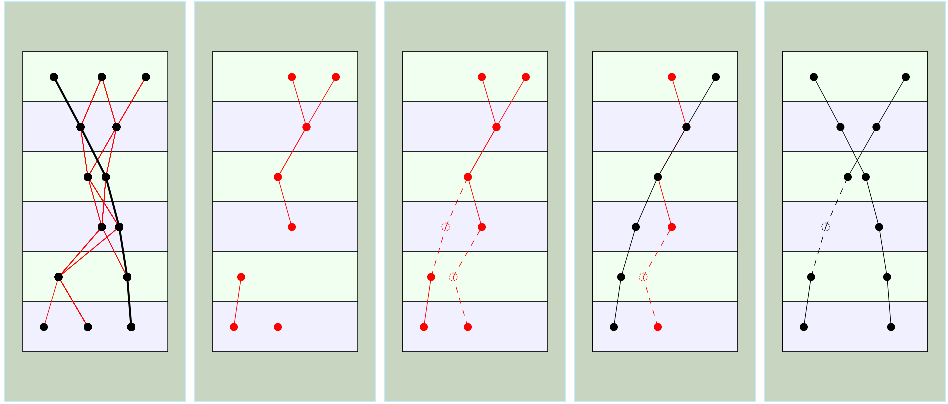
Physics Reconstruction (AI)



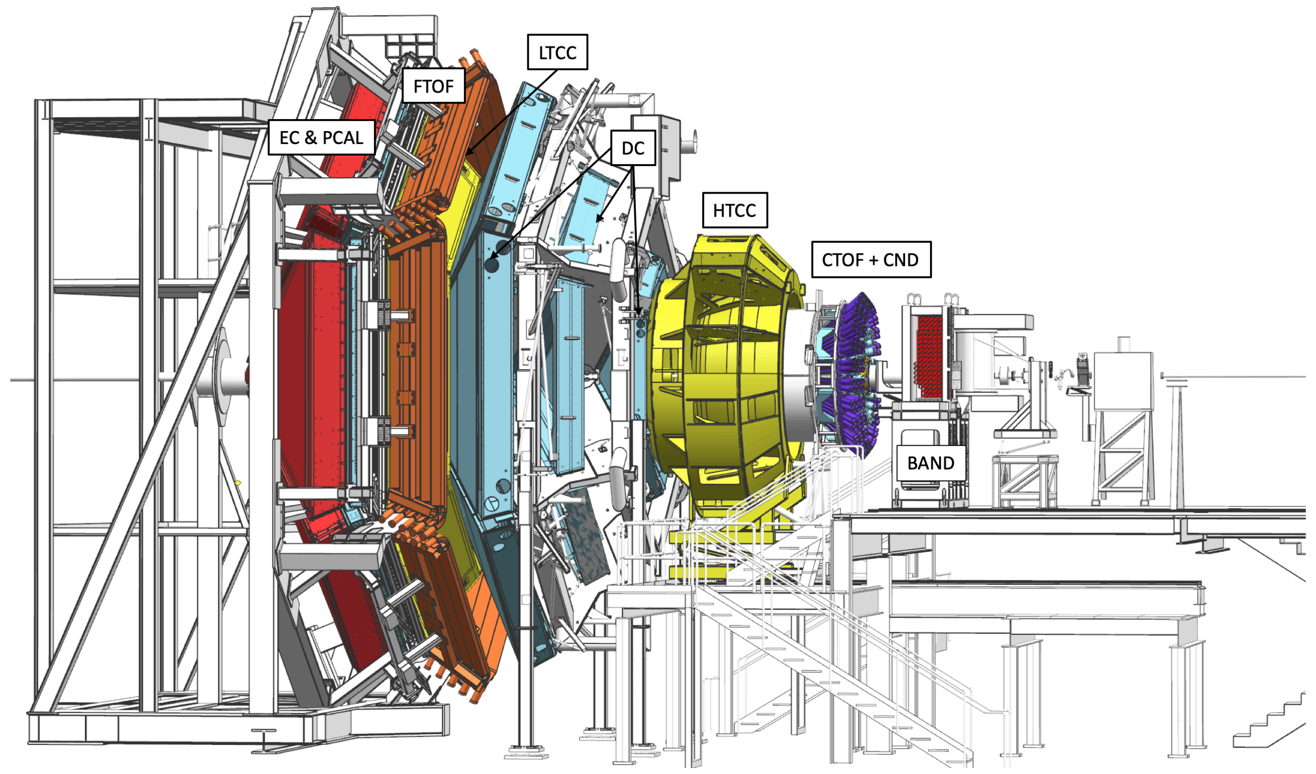
Data Persistence

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

Track Classification (AI)



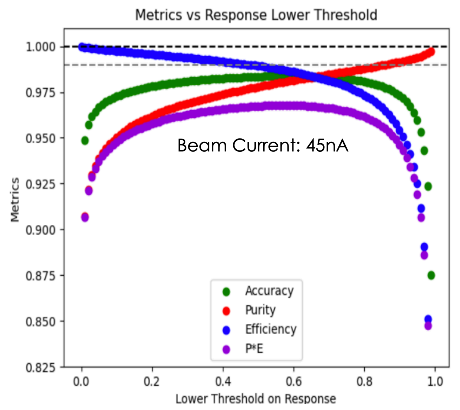
Classifying track candidates from
Reconstructed clusters
In real-time

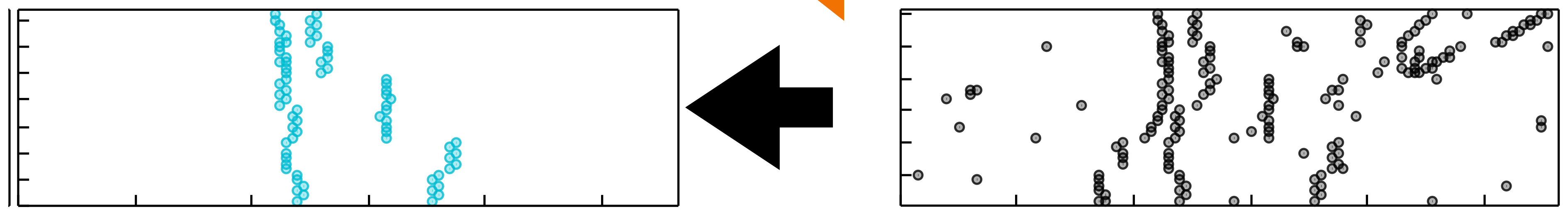


Data Acquisition

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





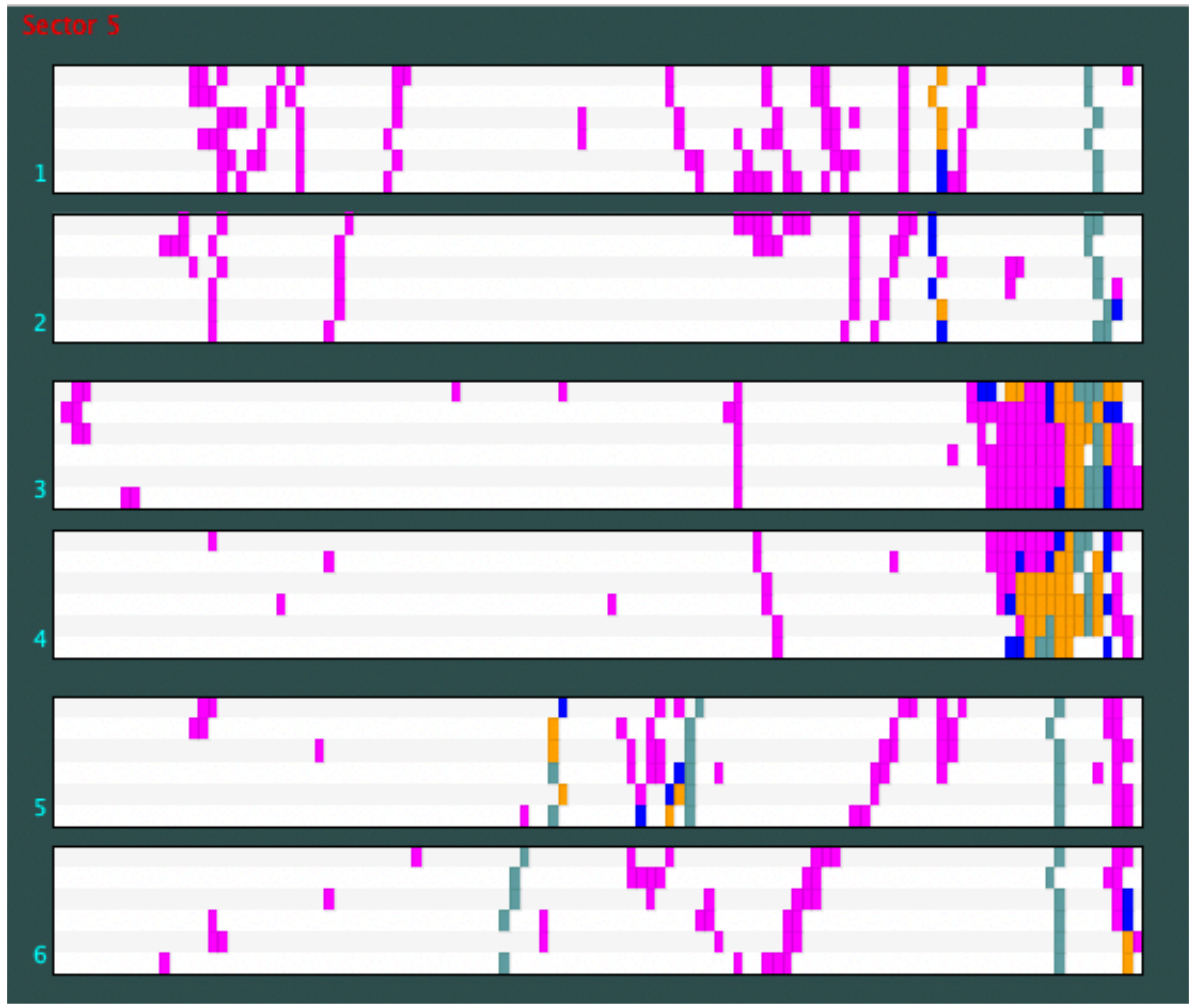
Removing Noise signals
From tracking detectors

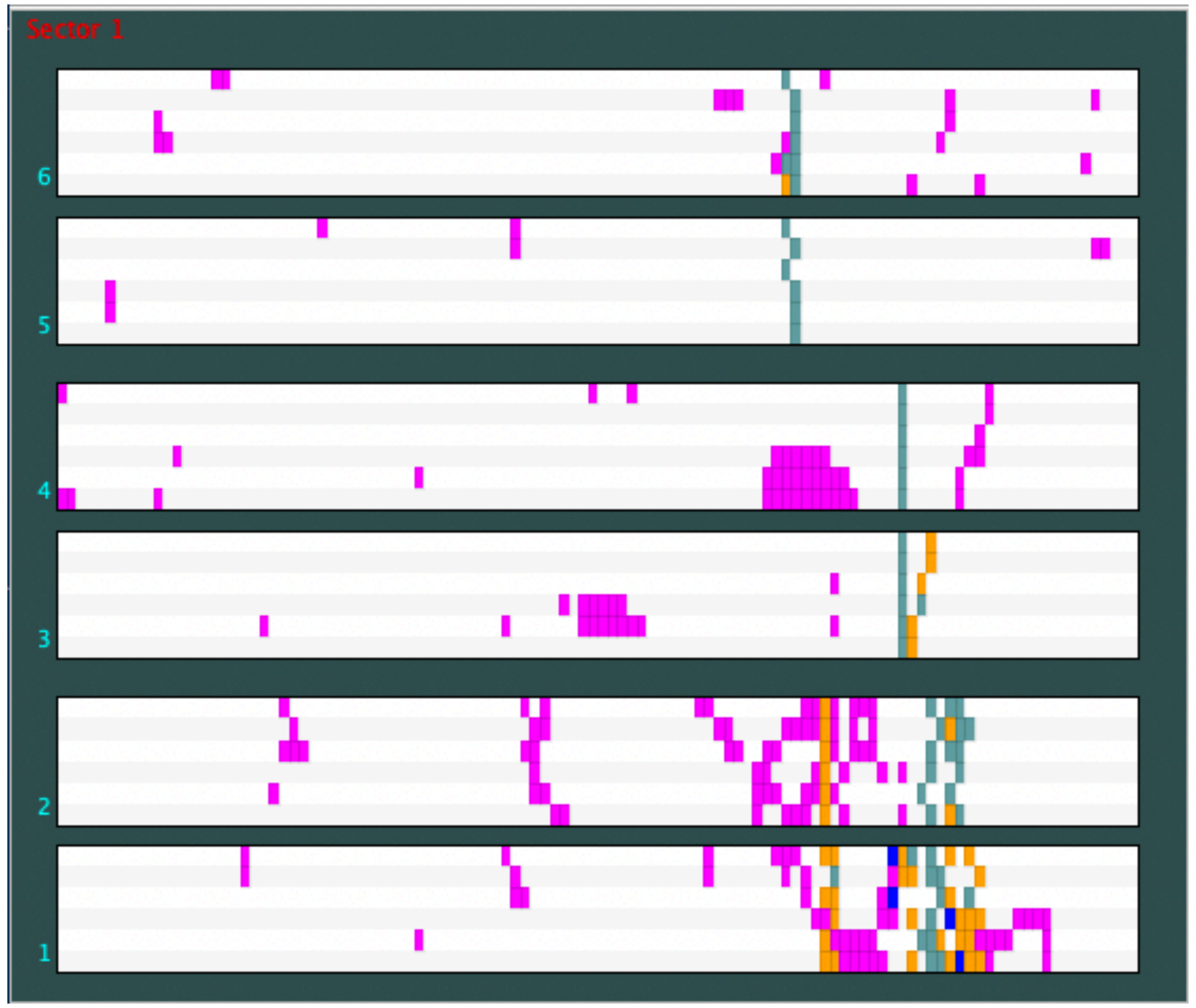
Data De-Noising (AI)

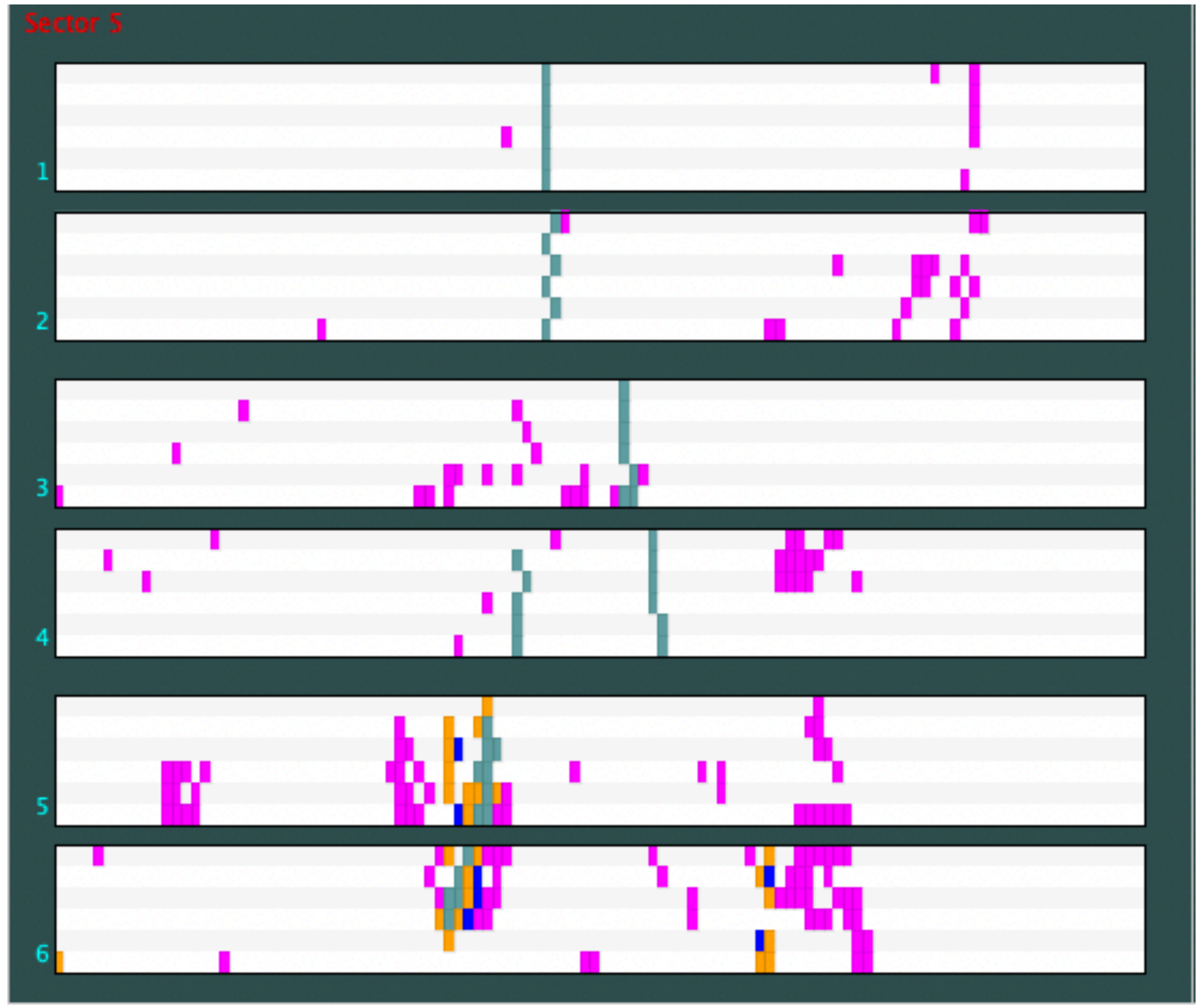
De-Noising

- 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)

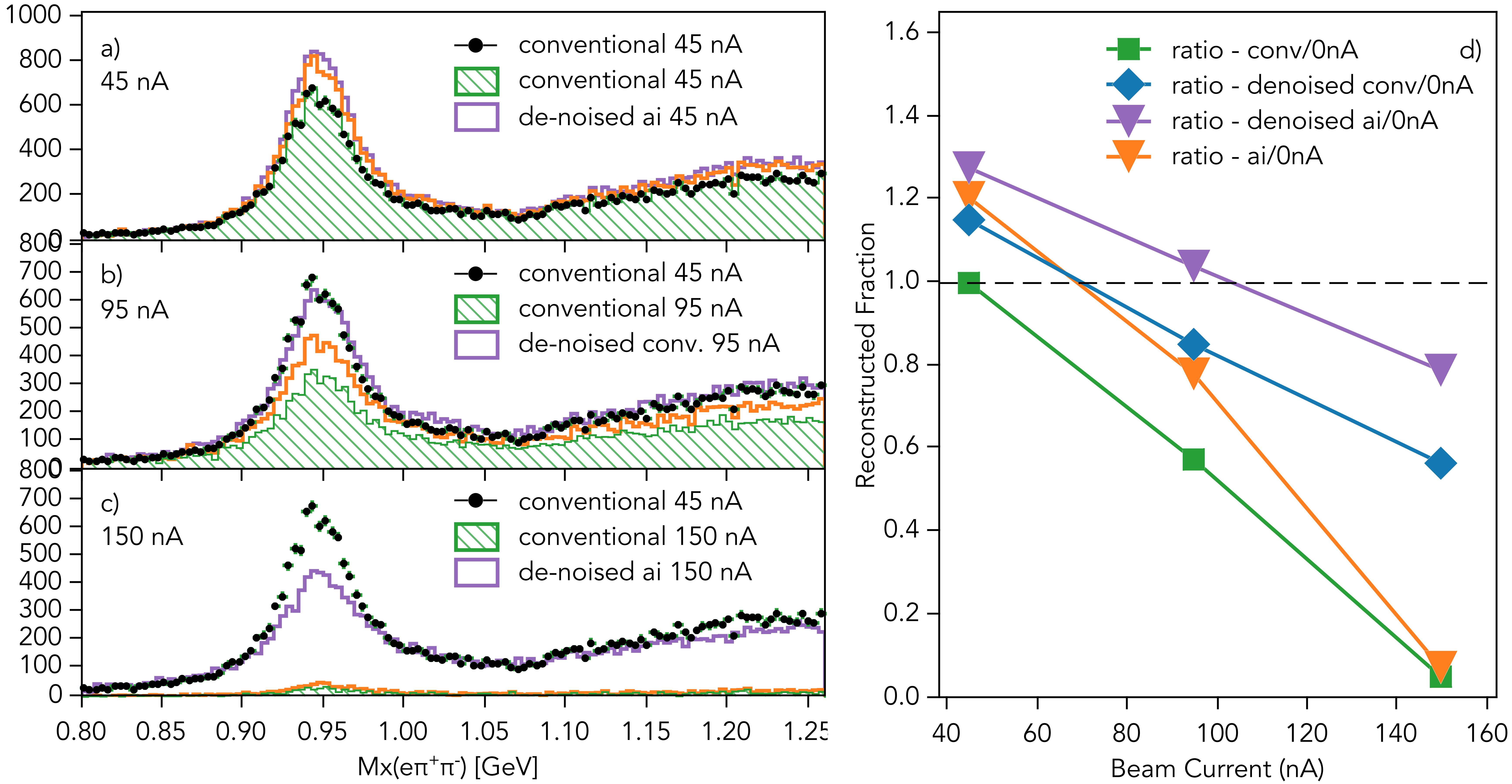






De-Noising Results (simulation)

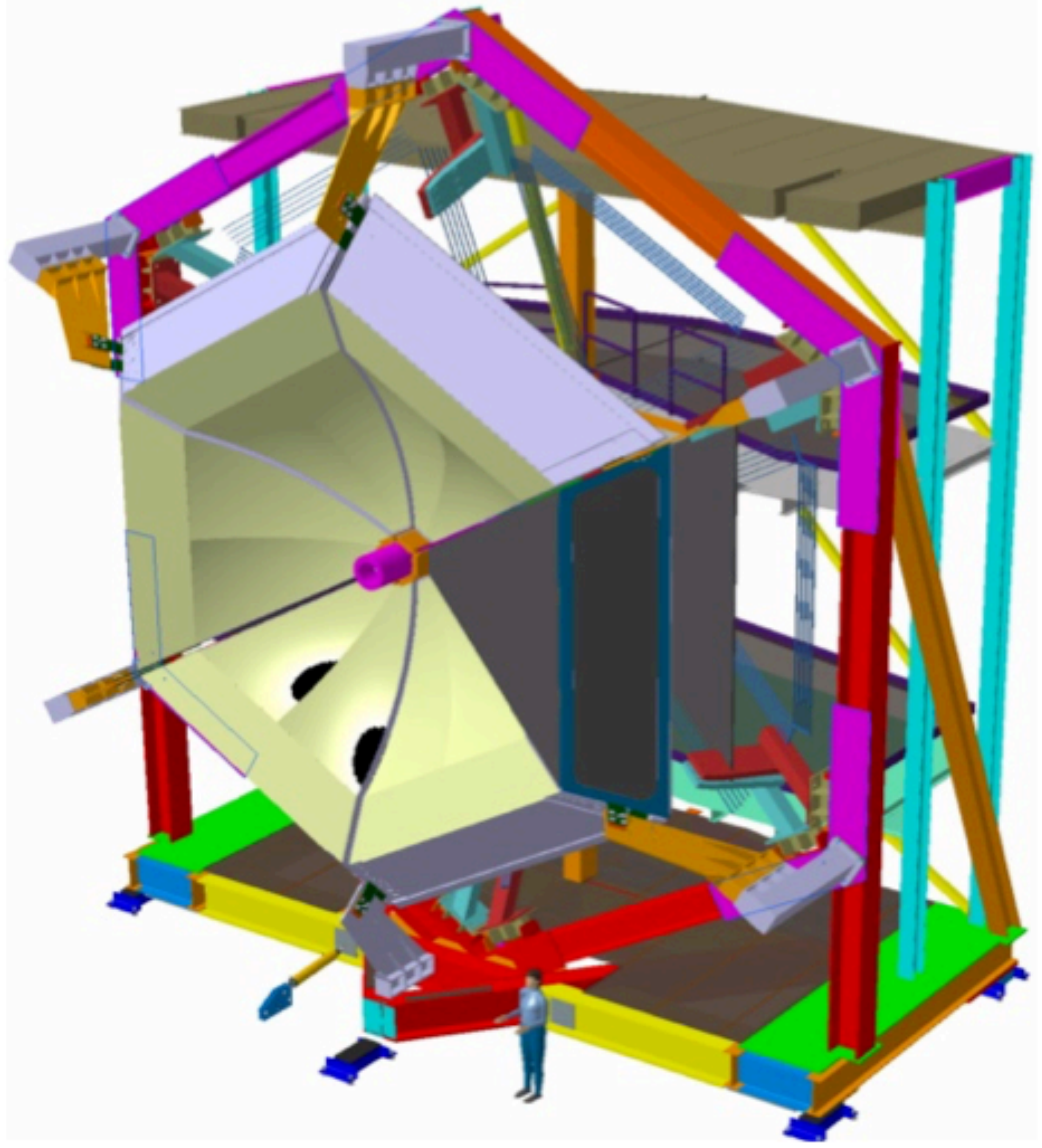
- The reconstruction is run on simulated data with a merging background for different incident beam currents (luminosity)
- The simulated three-particle final state is analyzed to measure yield for de-noised data and for conventional



- 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.

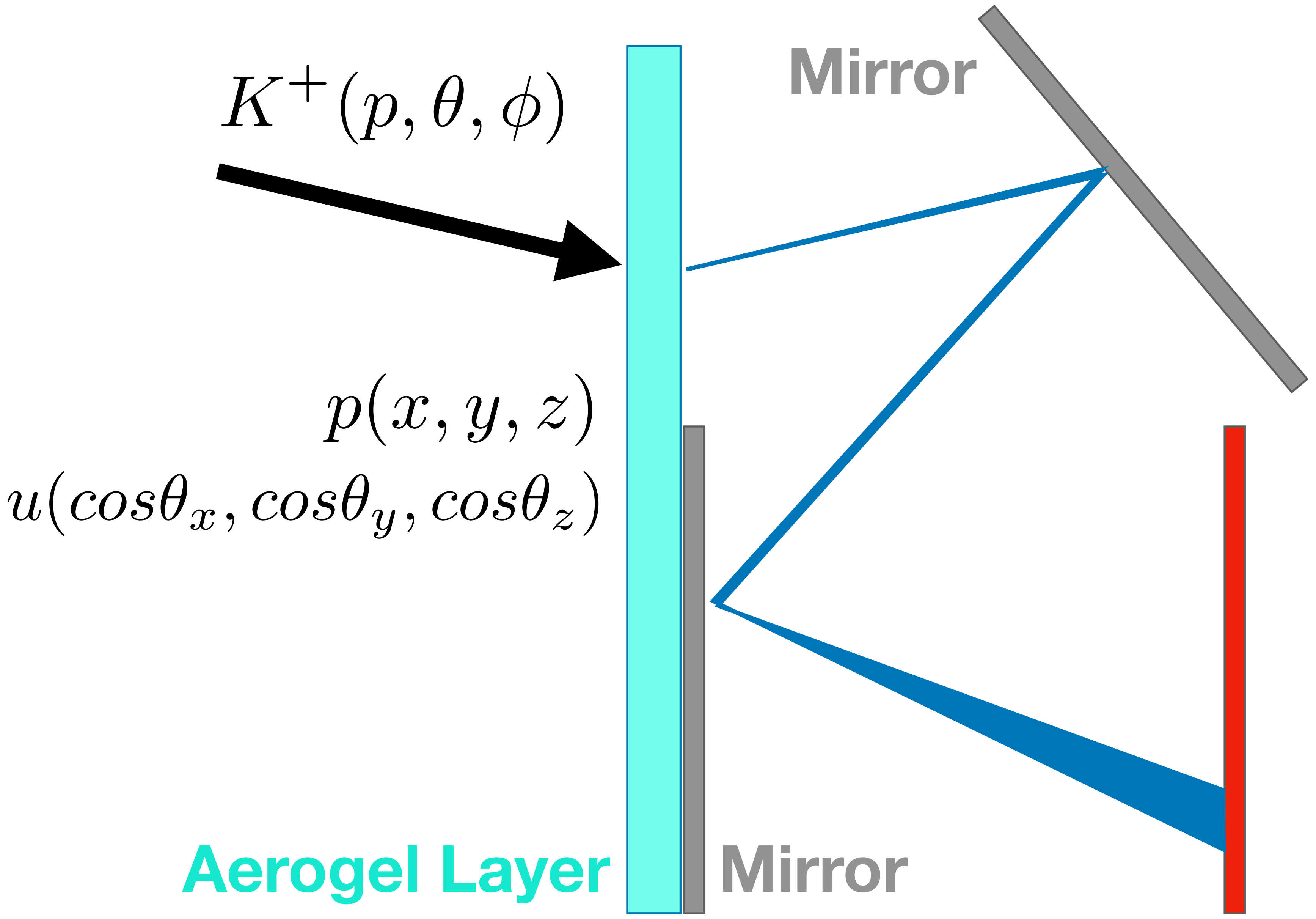
CLAS12 – RICH

- 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.

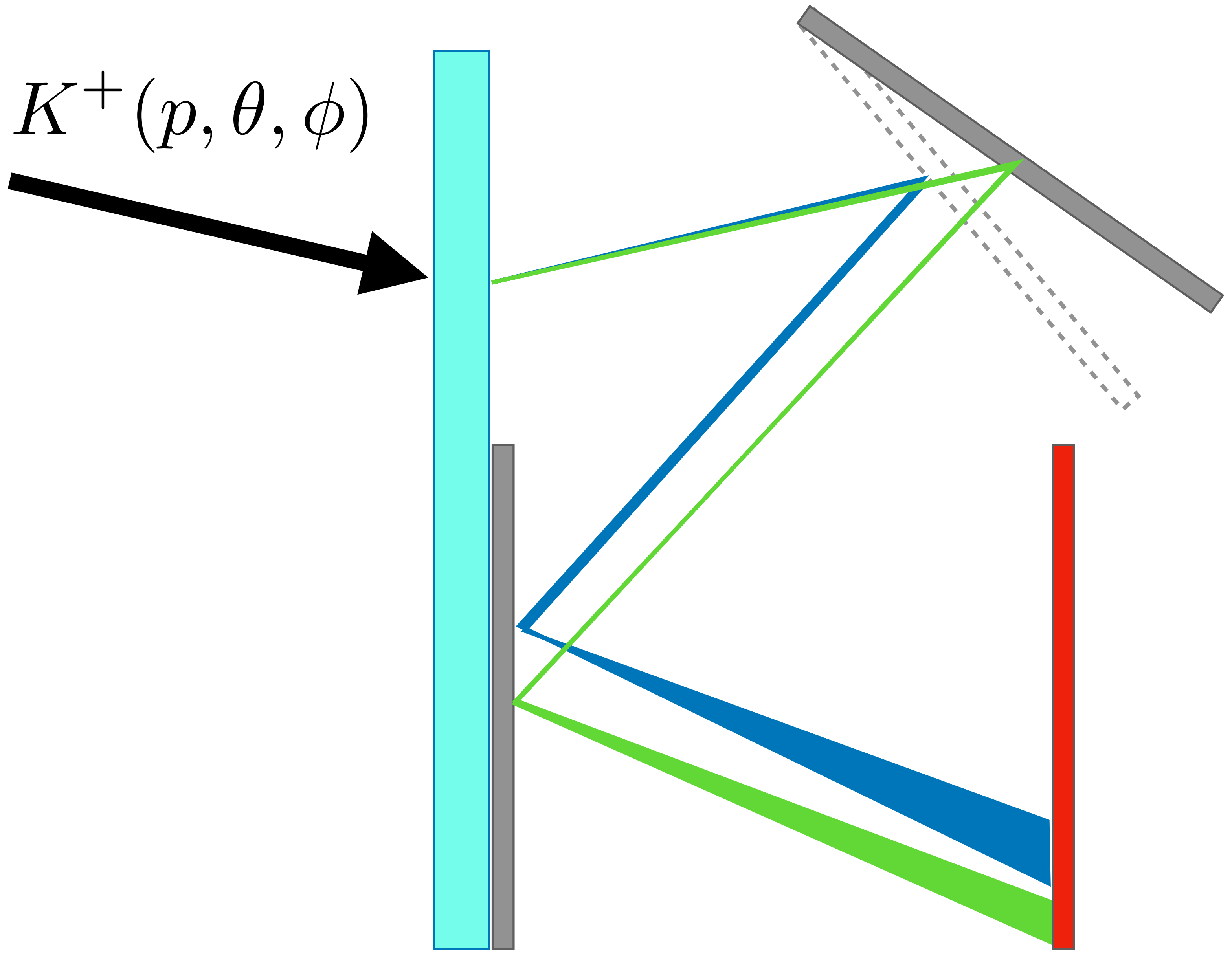


Rich Detector (preliminary)

- 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.

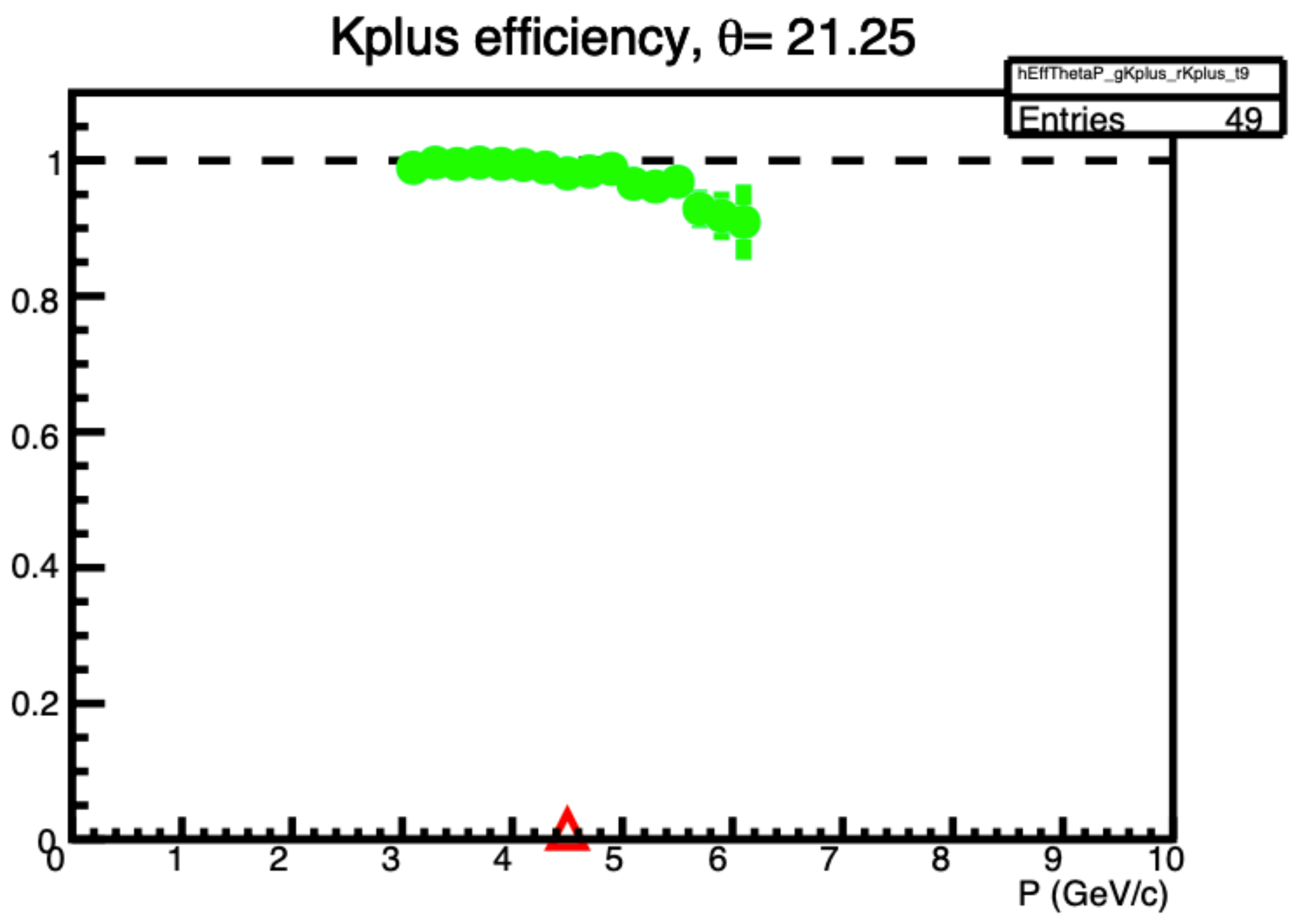


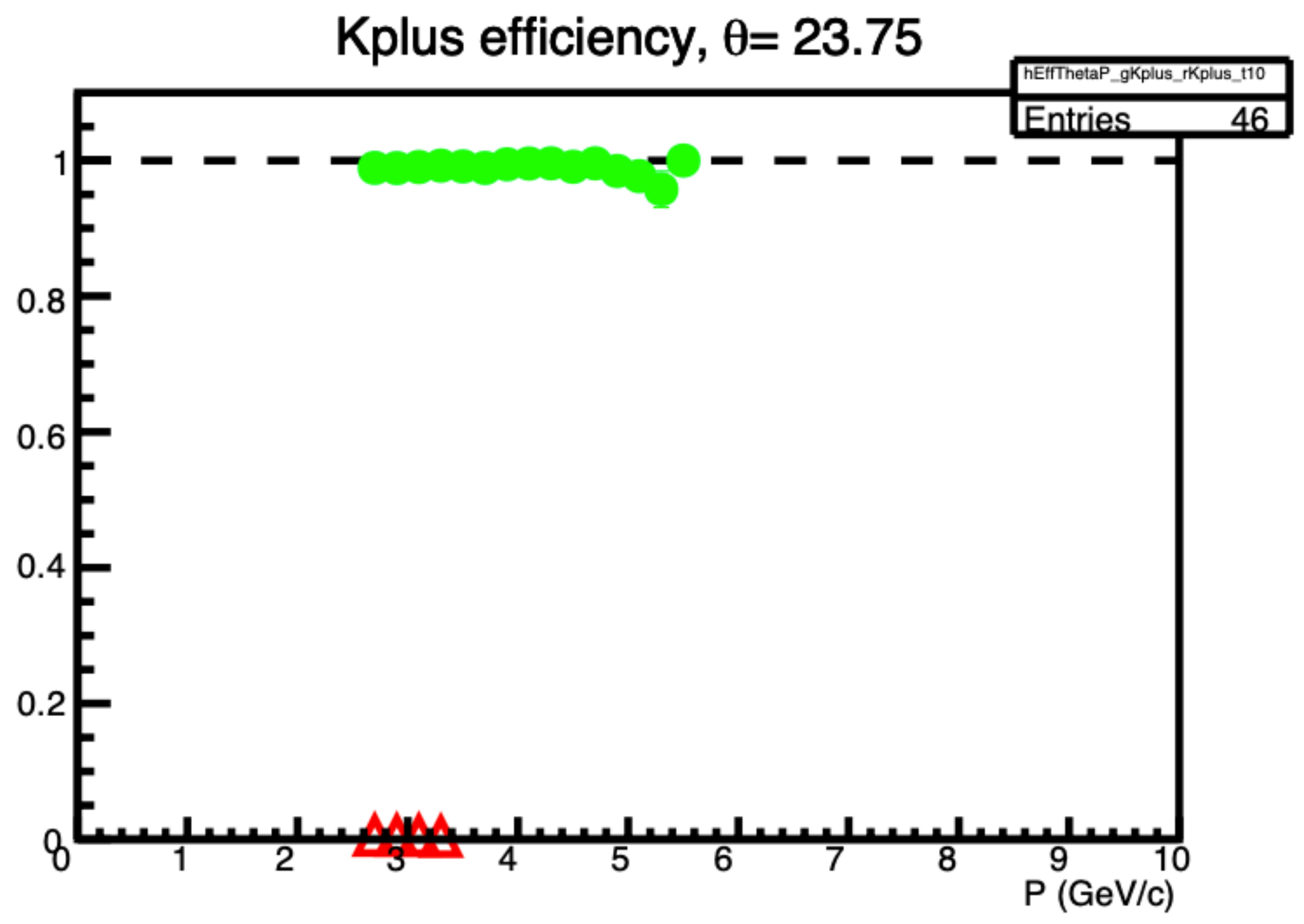
- RICH Real World Geometry
  - Ray tracing will predict an inaccurate position for the hit on the detector plane
  - This affects the efficiency of particle identification



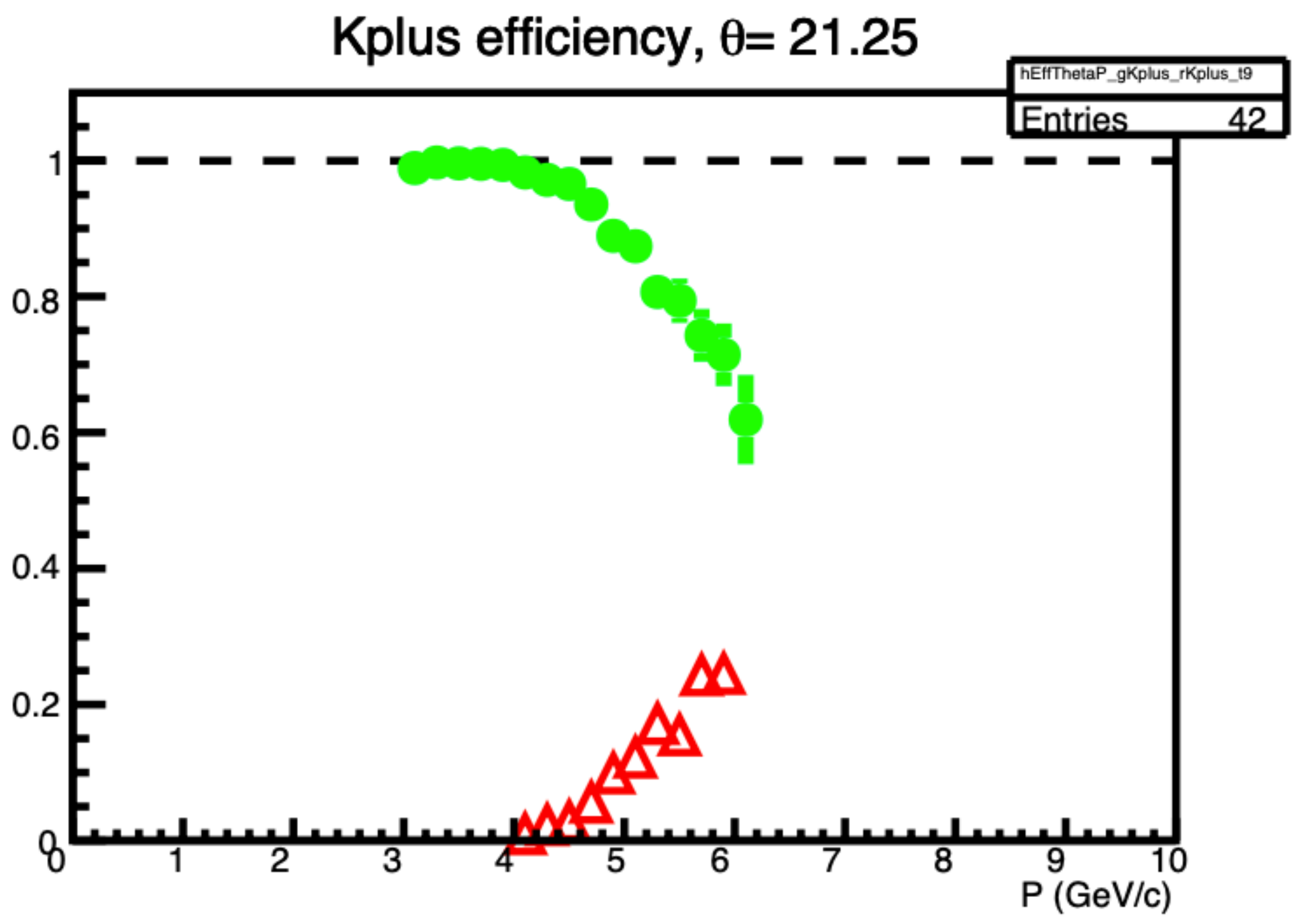
- Neural Networks can be trained on Real-World data which includes miss-alignments
- It can learn the Cherenkov ring patterns for incident particles, given interaction point and direction at crossing the aerogel layer

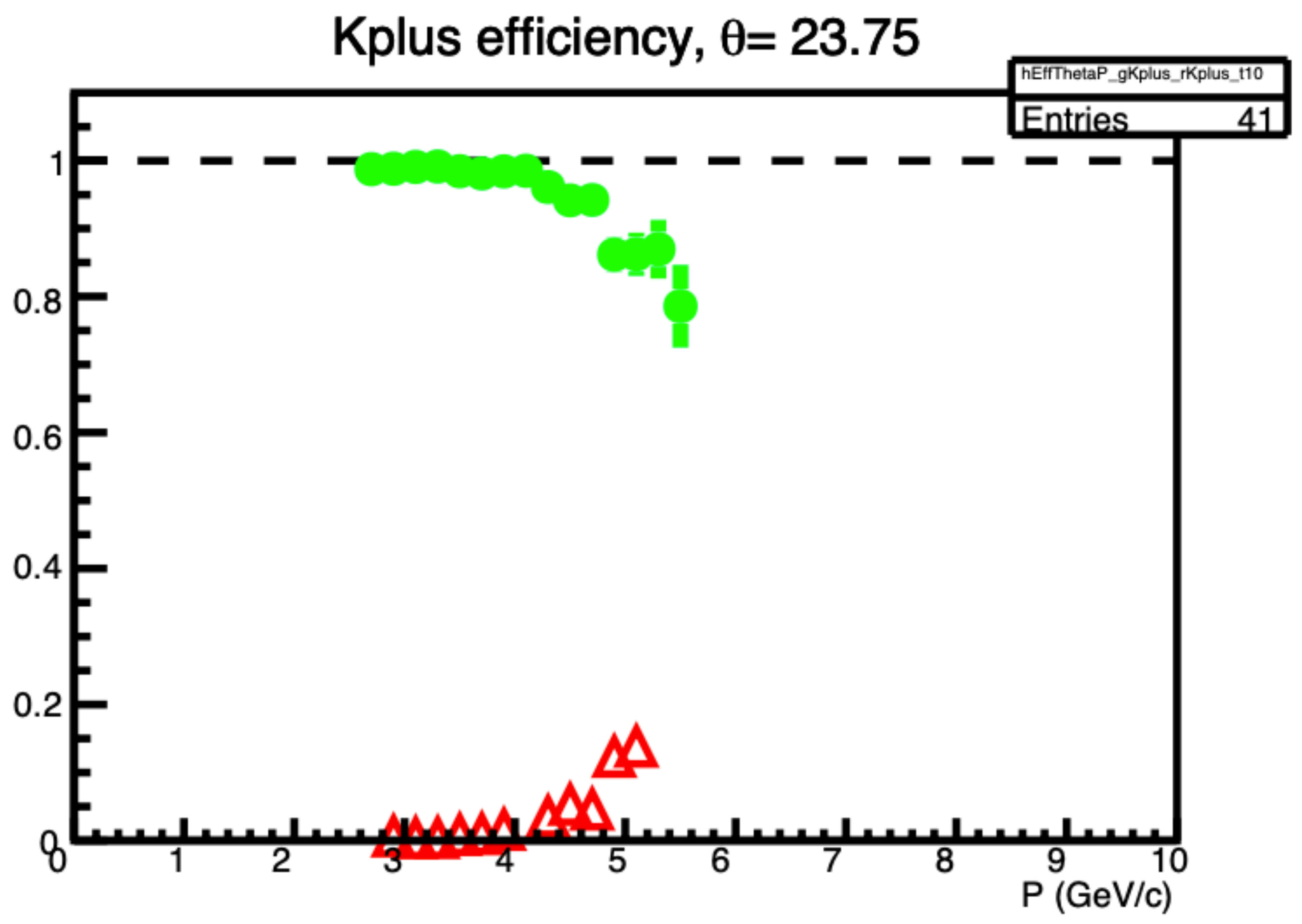
Kaon Identification Efficiency (IDEAL GEOMETRY)

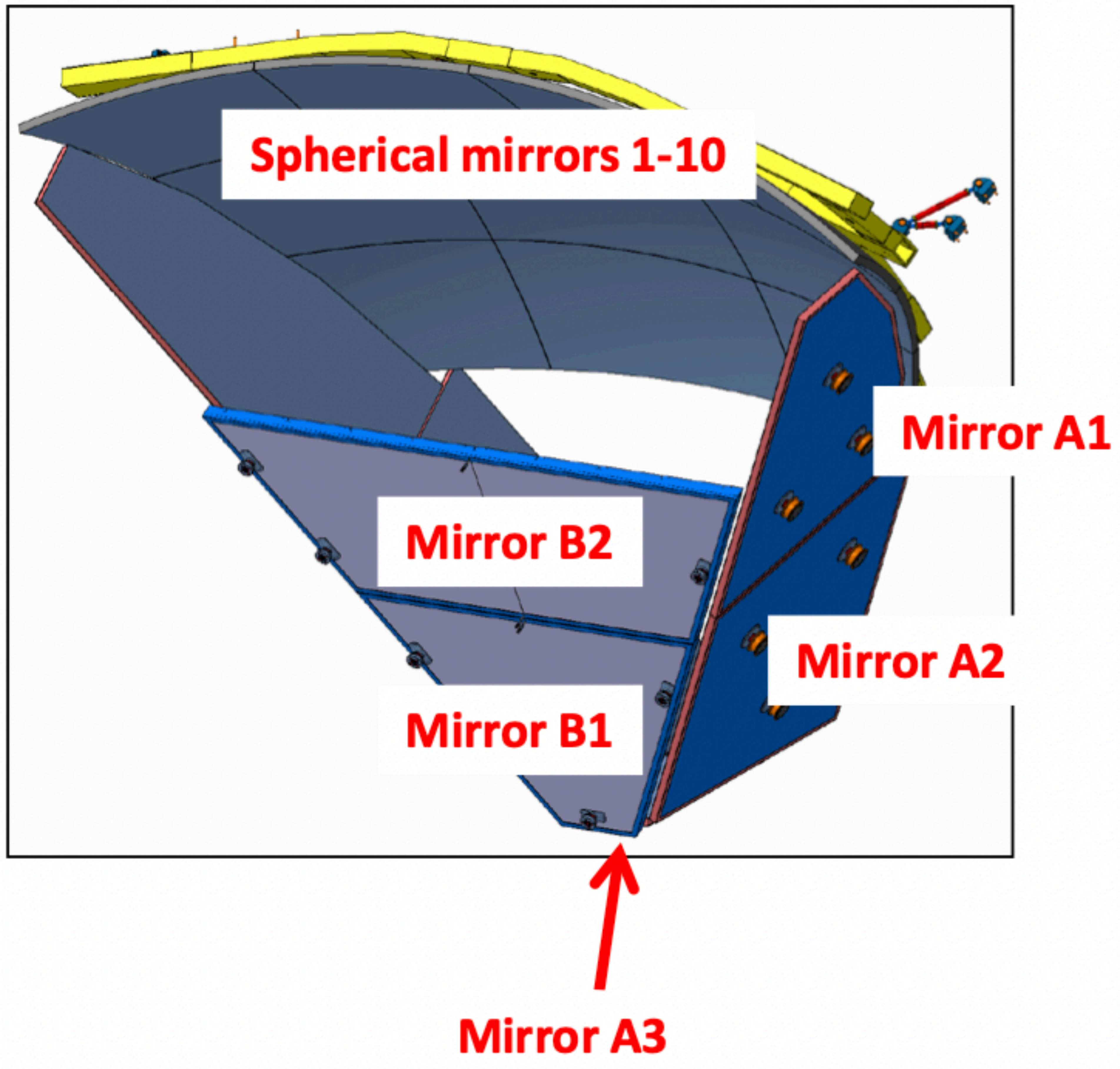




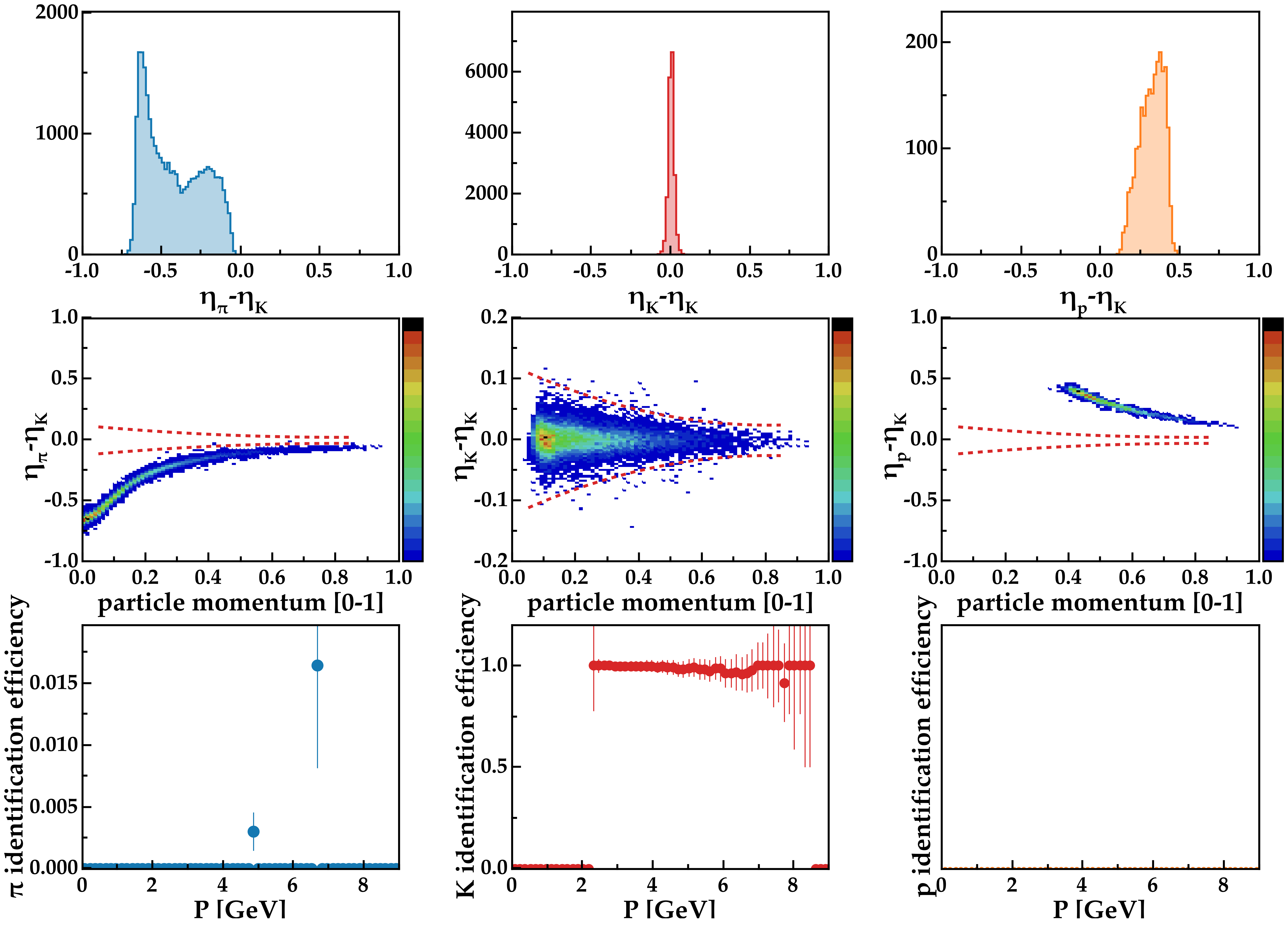
Kaon Identification Efficiency (MIS-ALLIGNED GEOMETRY)







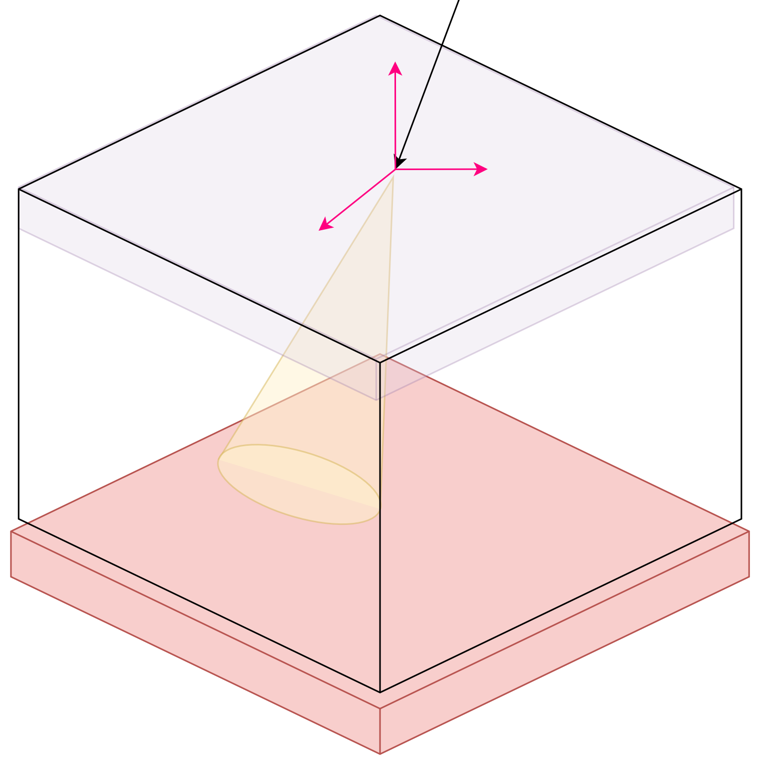
Rich Detector (particle identification)



- 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:

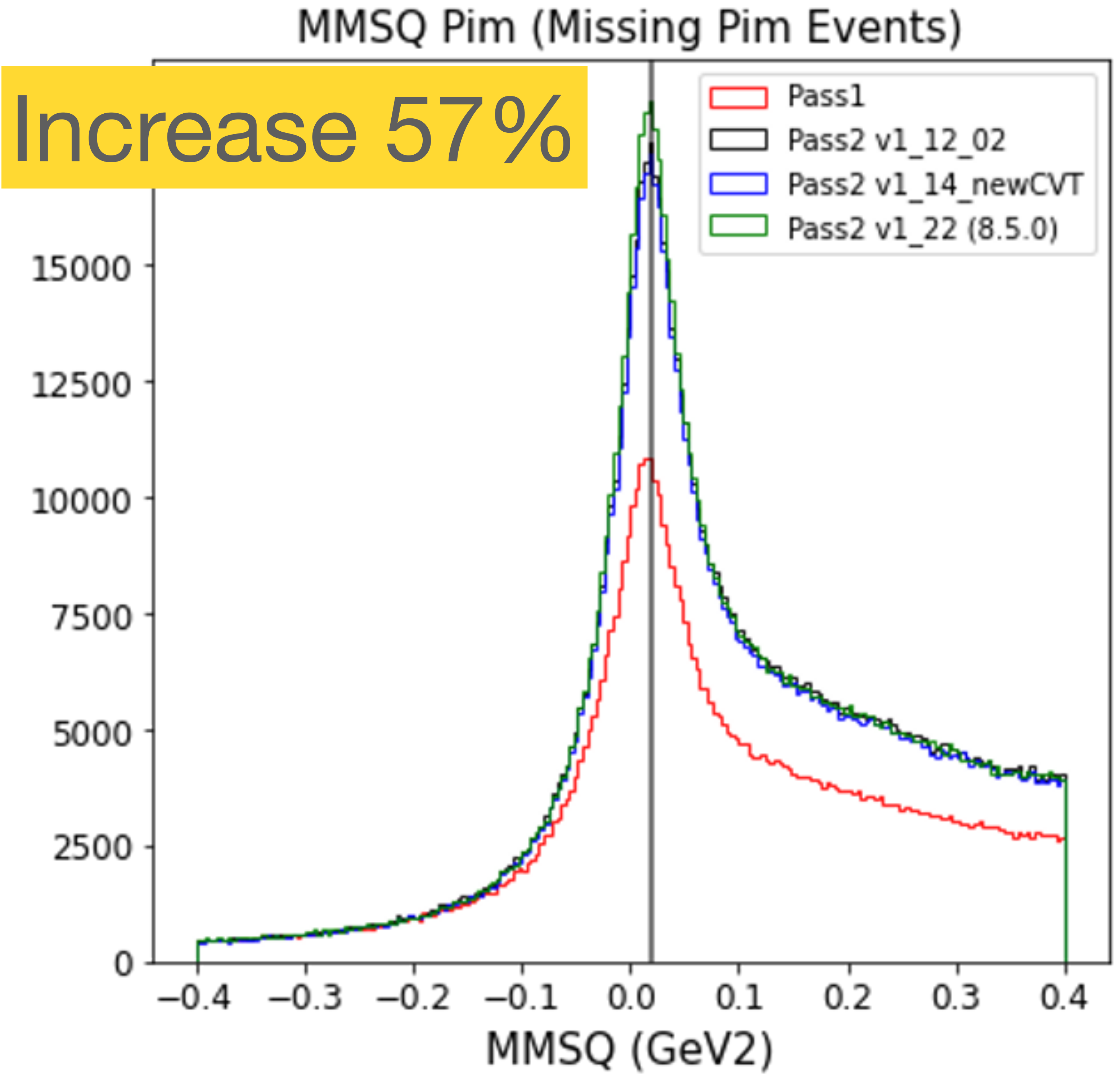
$$X[P, X, Y, cos\theta_x, cos\theta_y, cos\theta_z, X_h, Yh]$$



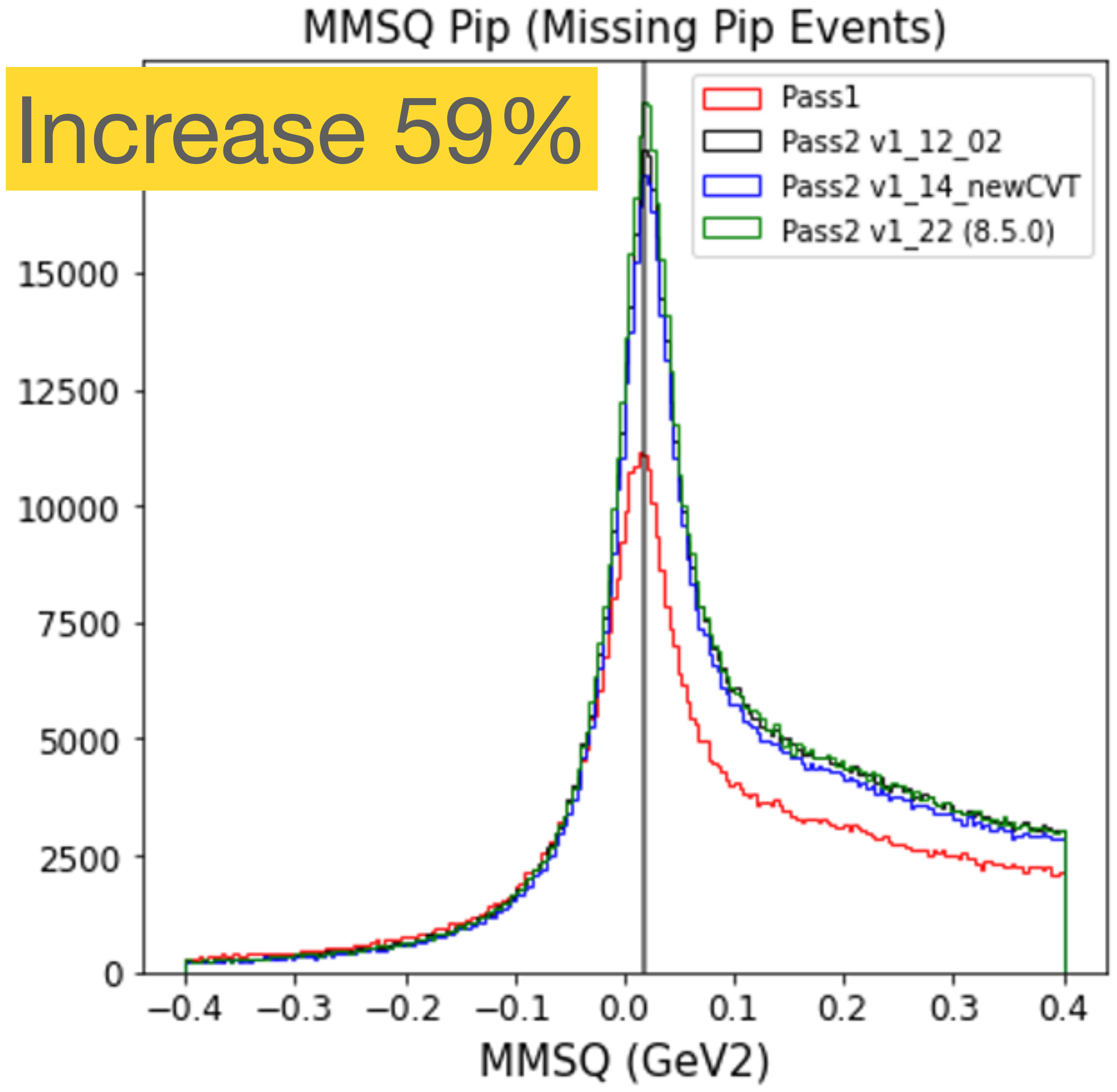
Output: η

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

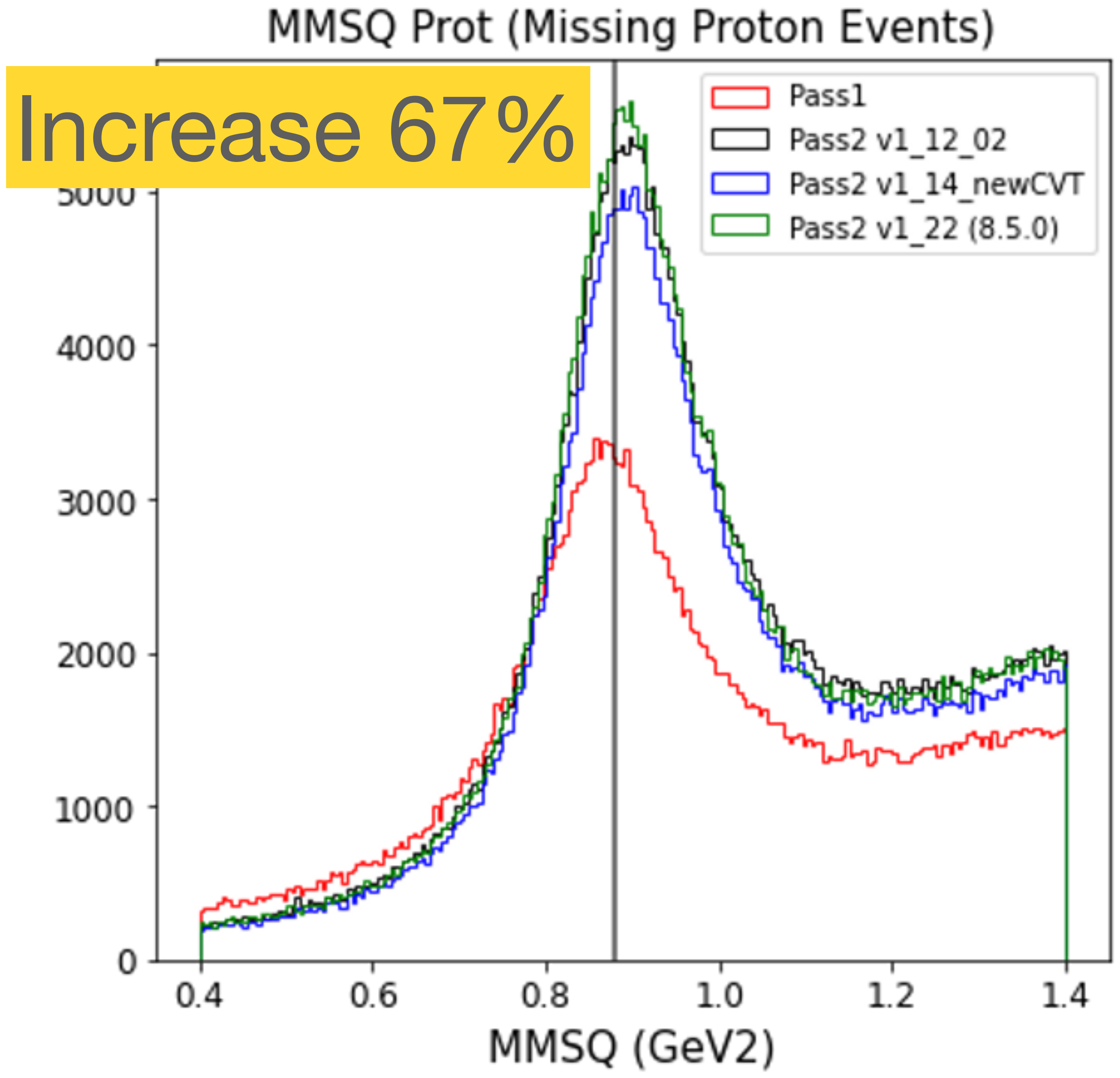
$$ep \rightarrow e'p\pi^-(X)$$



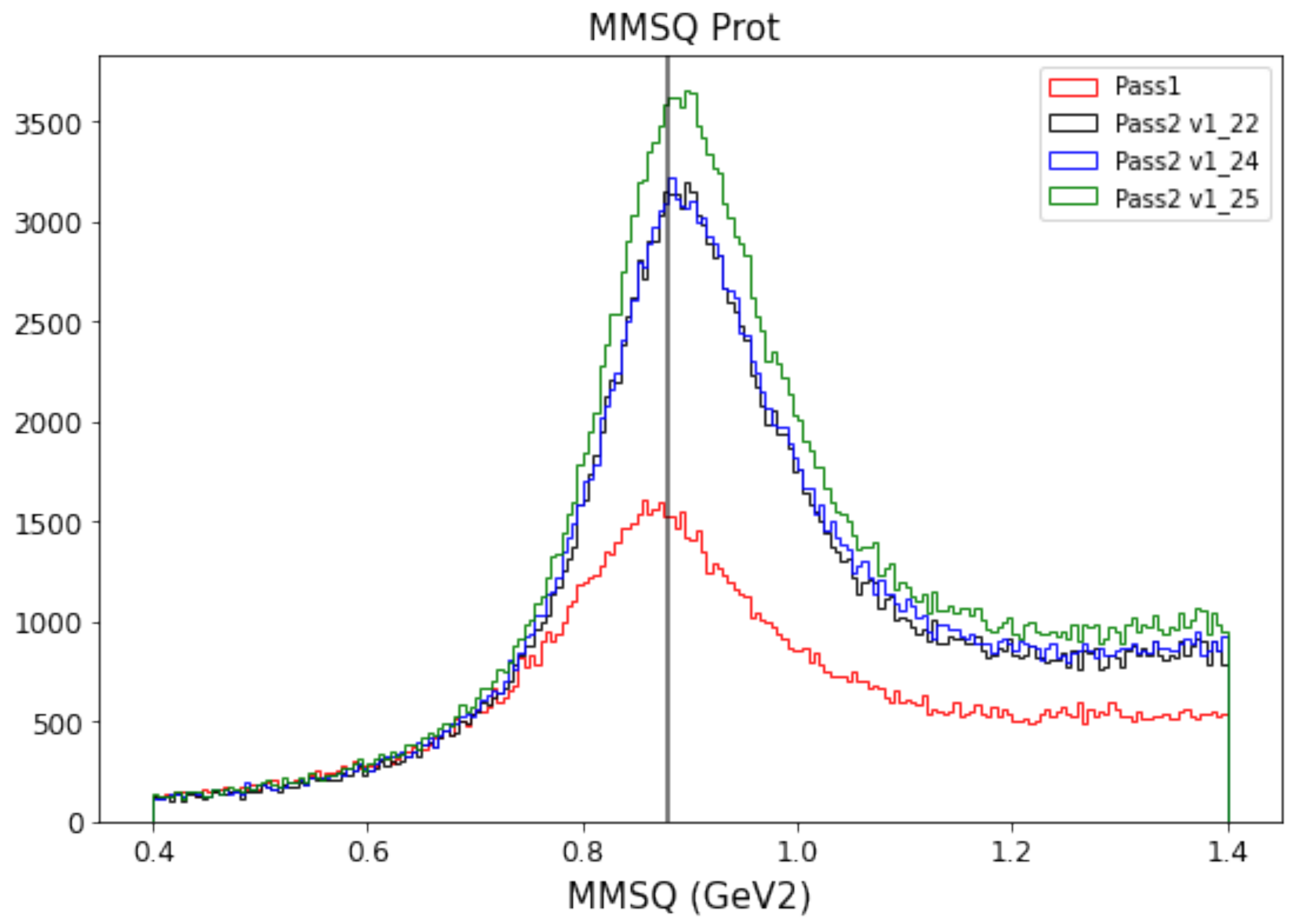
$$ep \rightarrow e'p\pi^+(X)$$

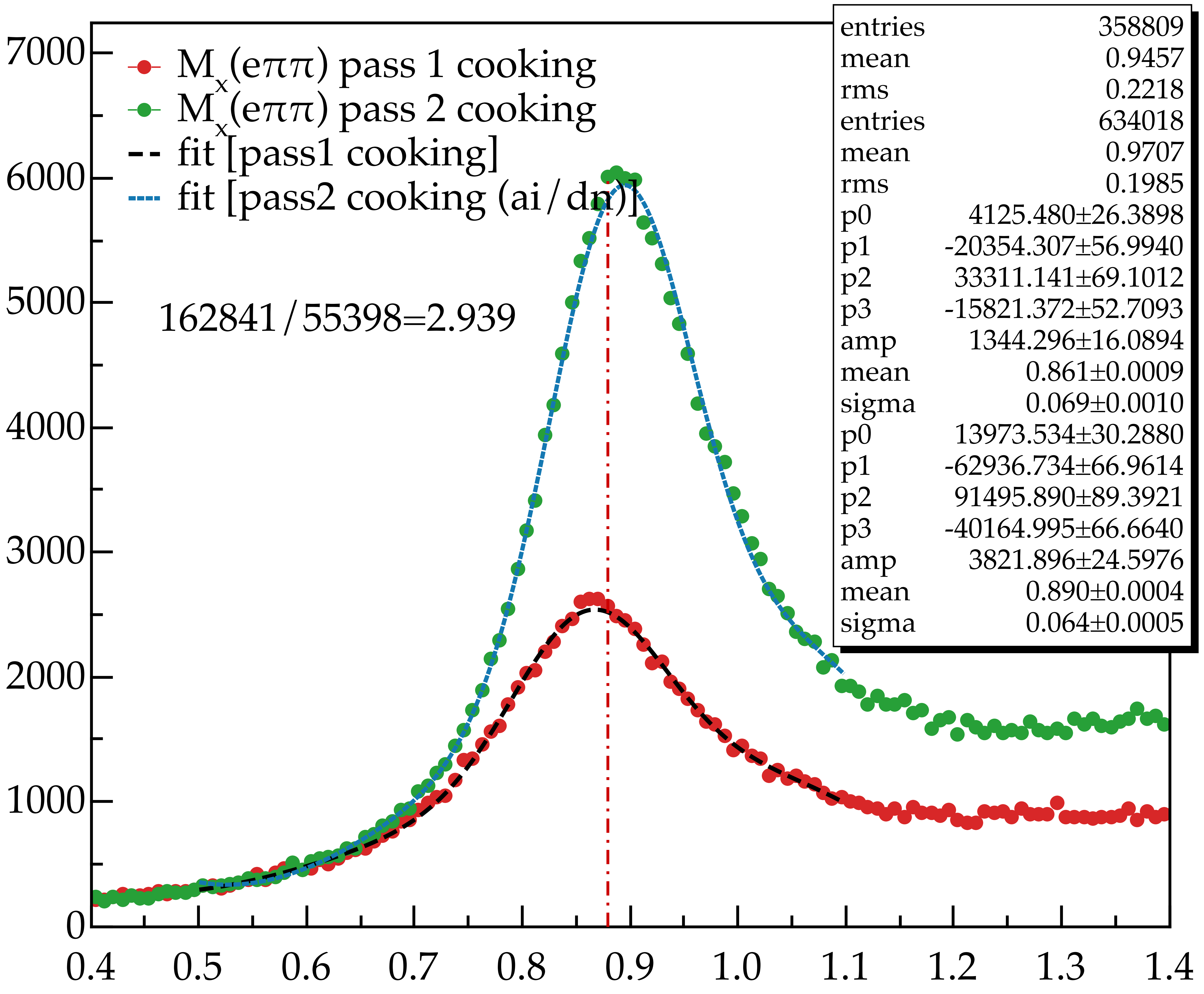


$$ep \rightarrow e'\pi^+\pi^-(X)$$

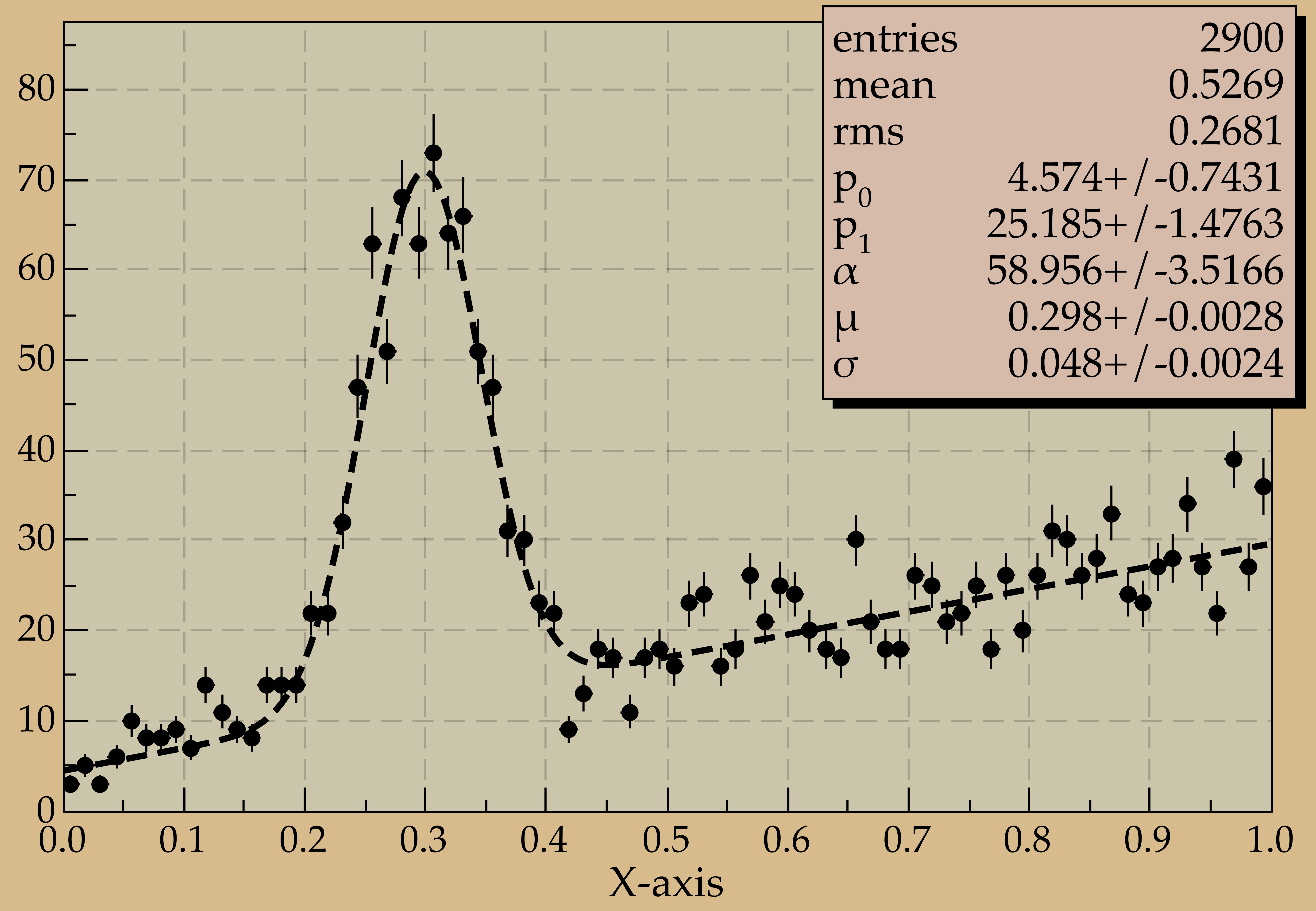


De-nosing Performance With Central Detector





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



entries 2900
mean 0.5269
rms 0.2681
p0 4.574+/-0.7431
p1 25.185+/-1.4763
α 58.956+/-3.5166
μ 0.298+/-0.0028
σ 0.048+/-0.0024
0
10
20
30
40
50
60
70
80
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
X-axis