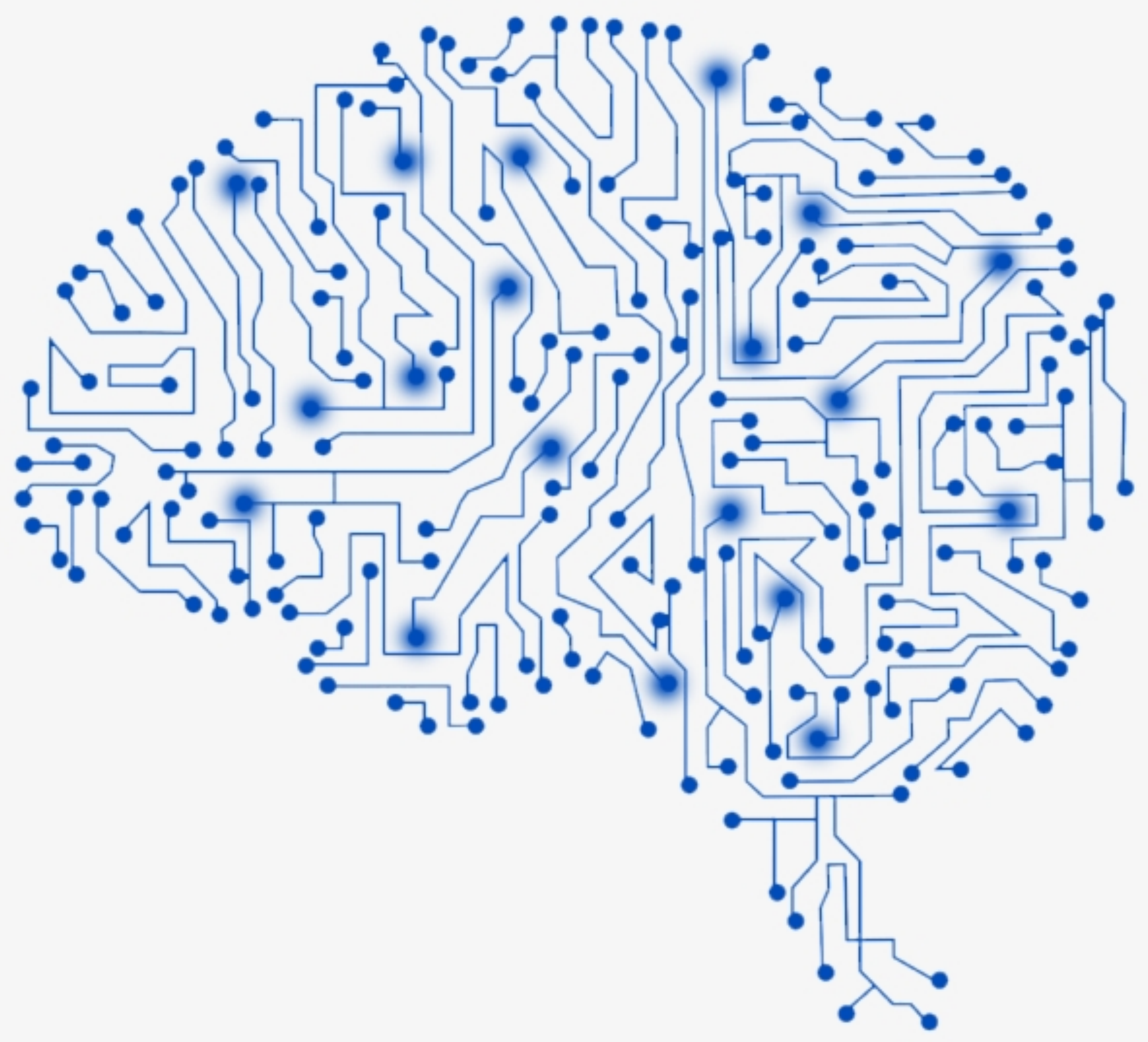
Artificial Intelligence

Artificial Intelligence/Machine Learning for Physics Applications

G.Gavalian (Jefferson Lab)

Richard Tyson (Glasgow)

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

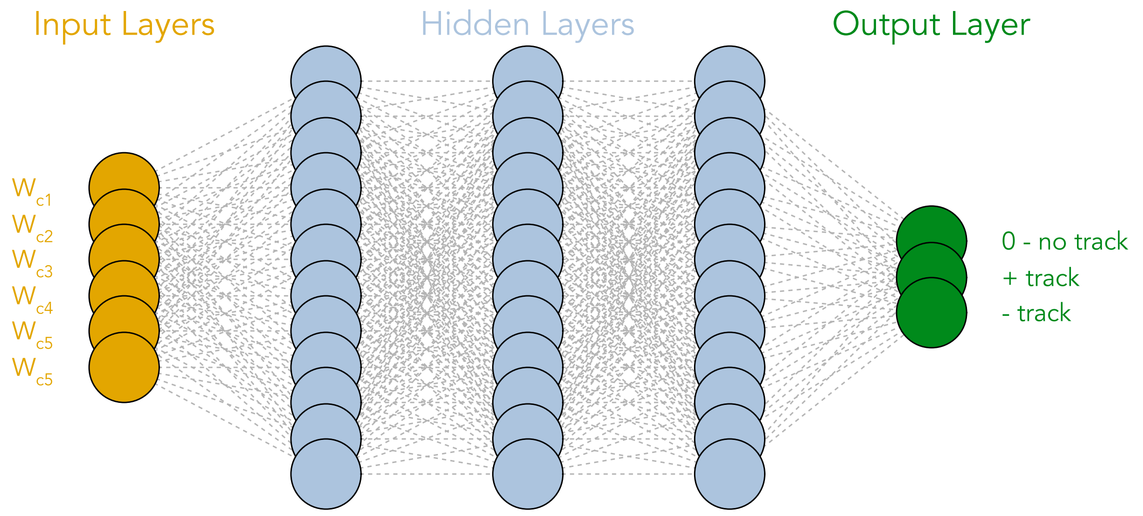


HUGS (June 7-9,2022)

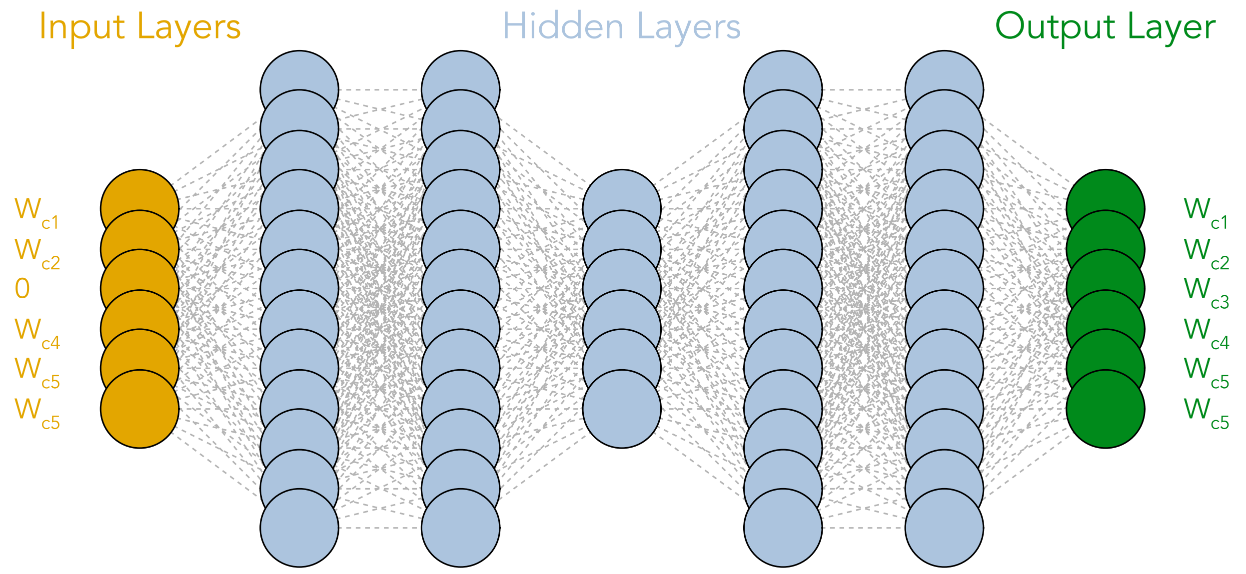
Machine Learning (MLP summary)

	Activation Hidden/Output	Loss Function
Classifier	RELU/SOFTMAX	Mean Squared Error (MSE)
Corruption Auto-Encoder	RELU/SEGIMOID	Mean Squared Error (MSE)
Linear Regression	TANH/LIN	Mean Squared Error (MSE)

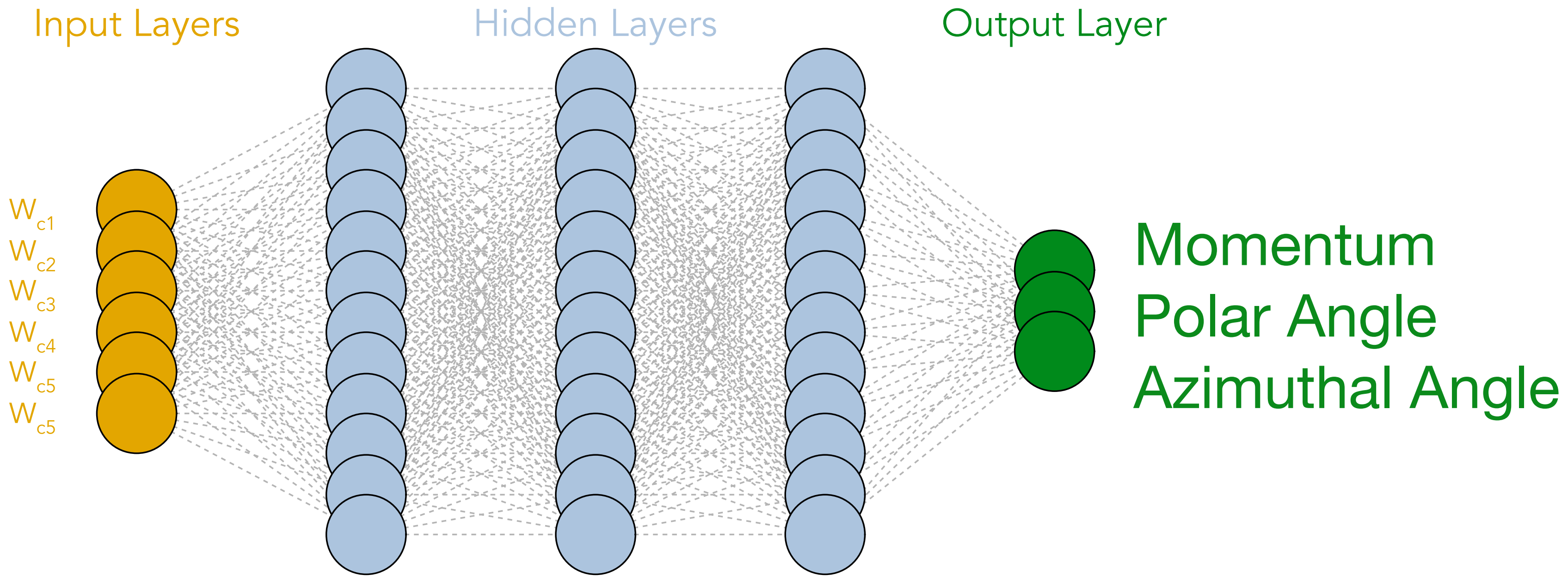
Classifier



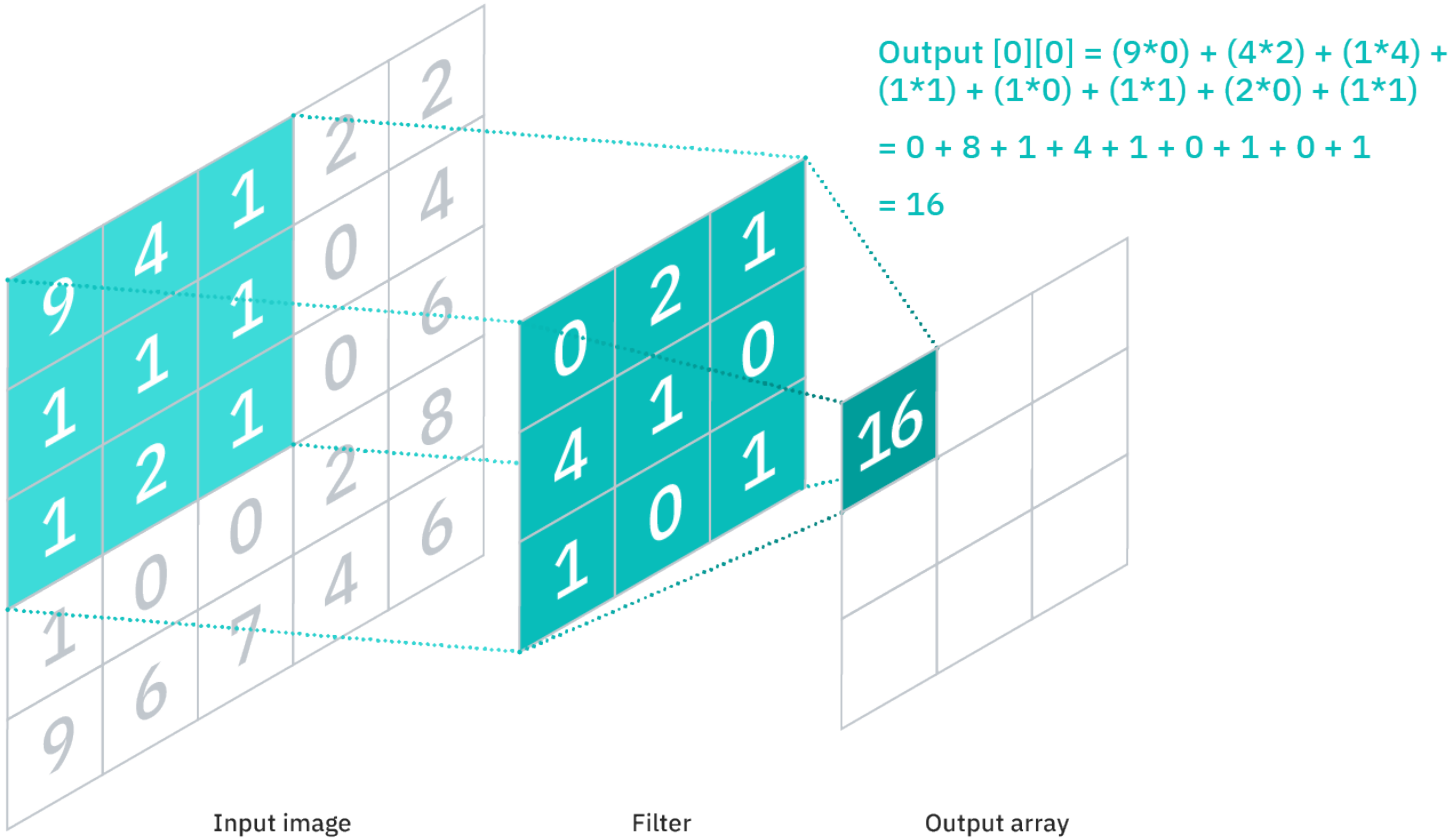
Corruption Auto-Encoder



Linear Regression

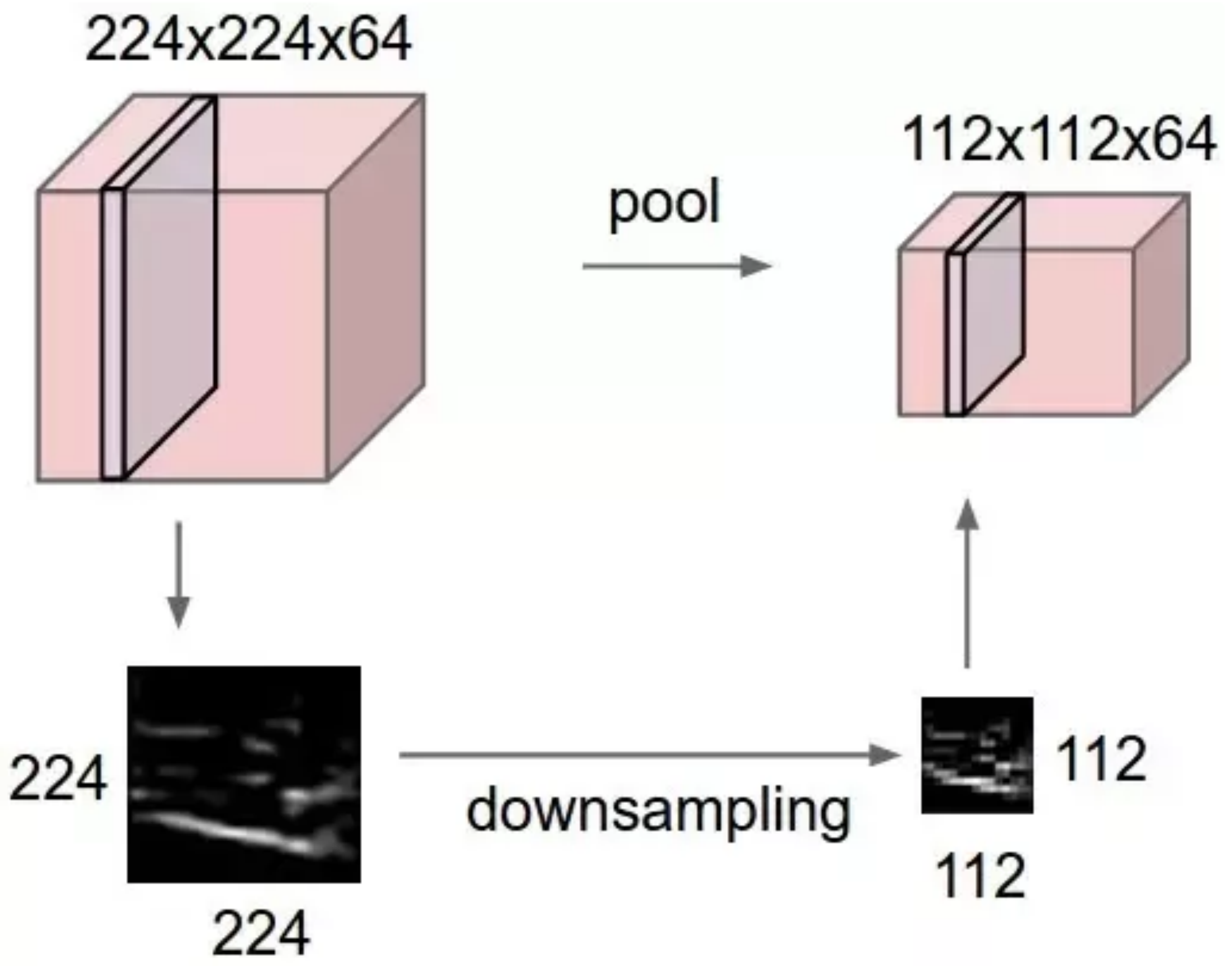


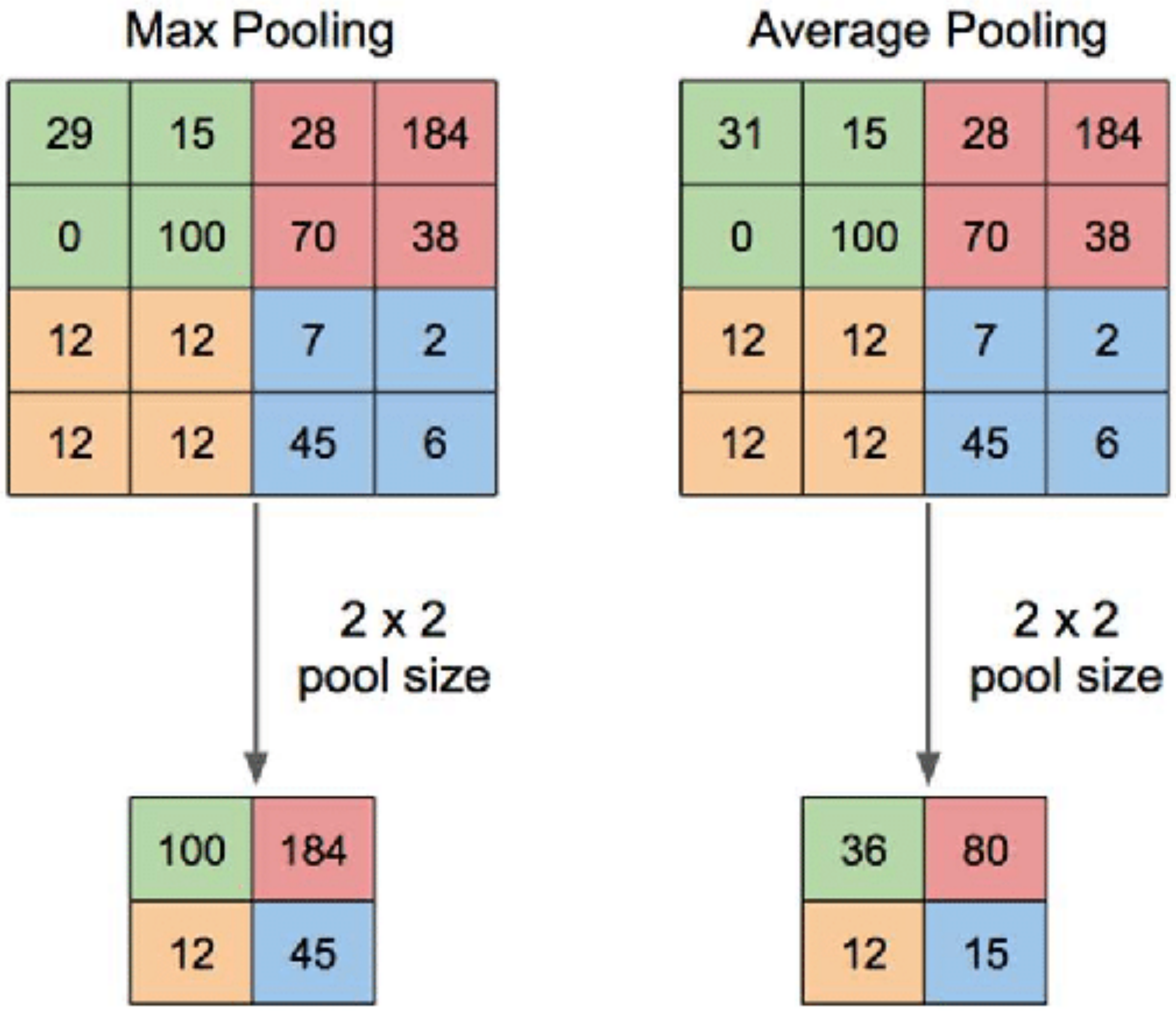
Convolutional Neural Networks (CNN)

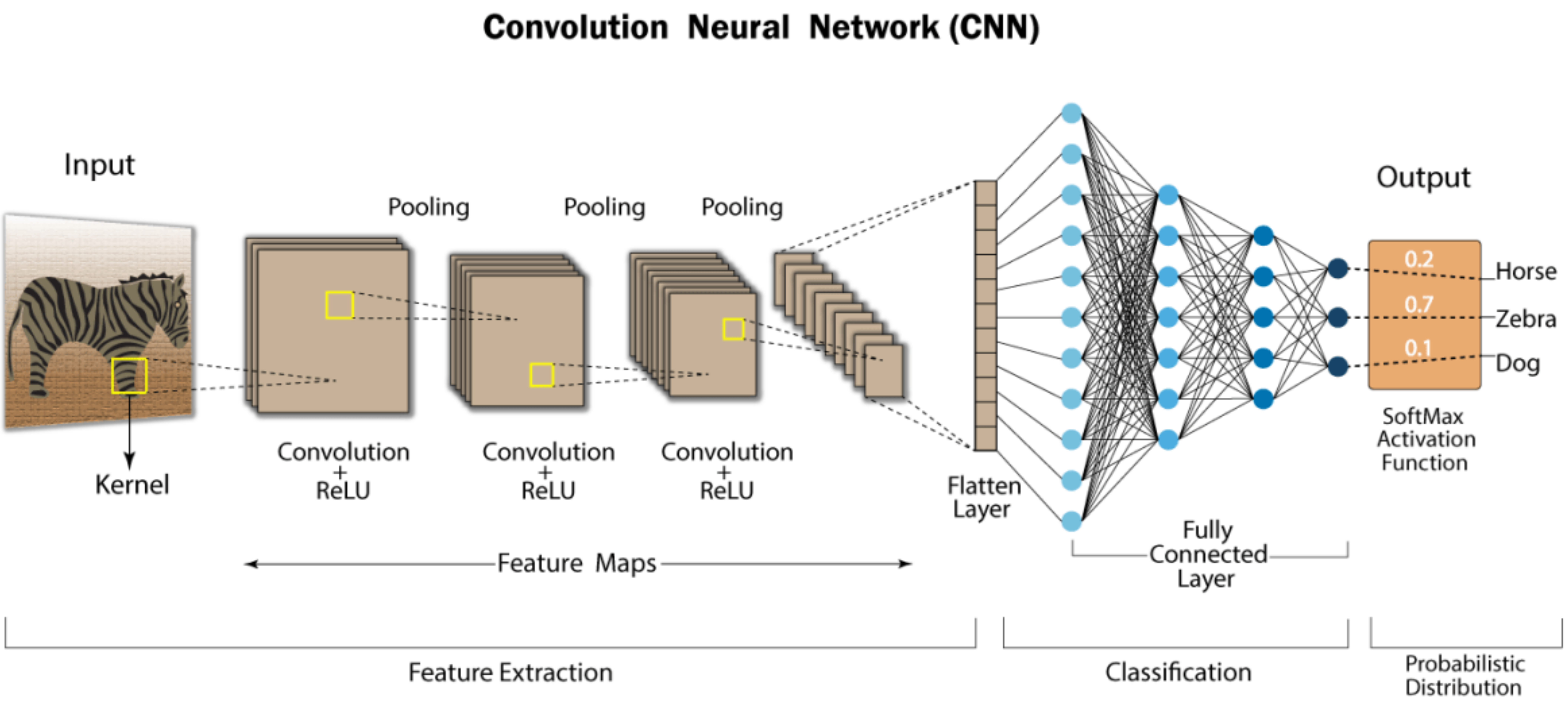


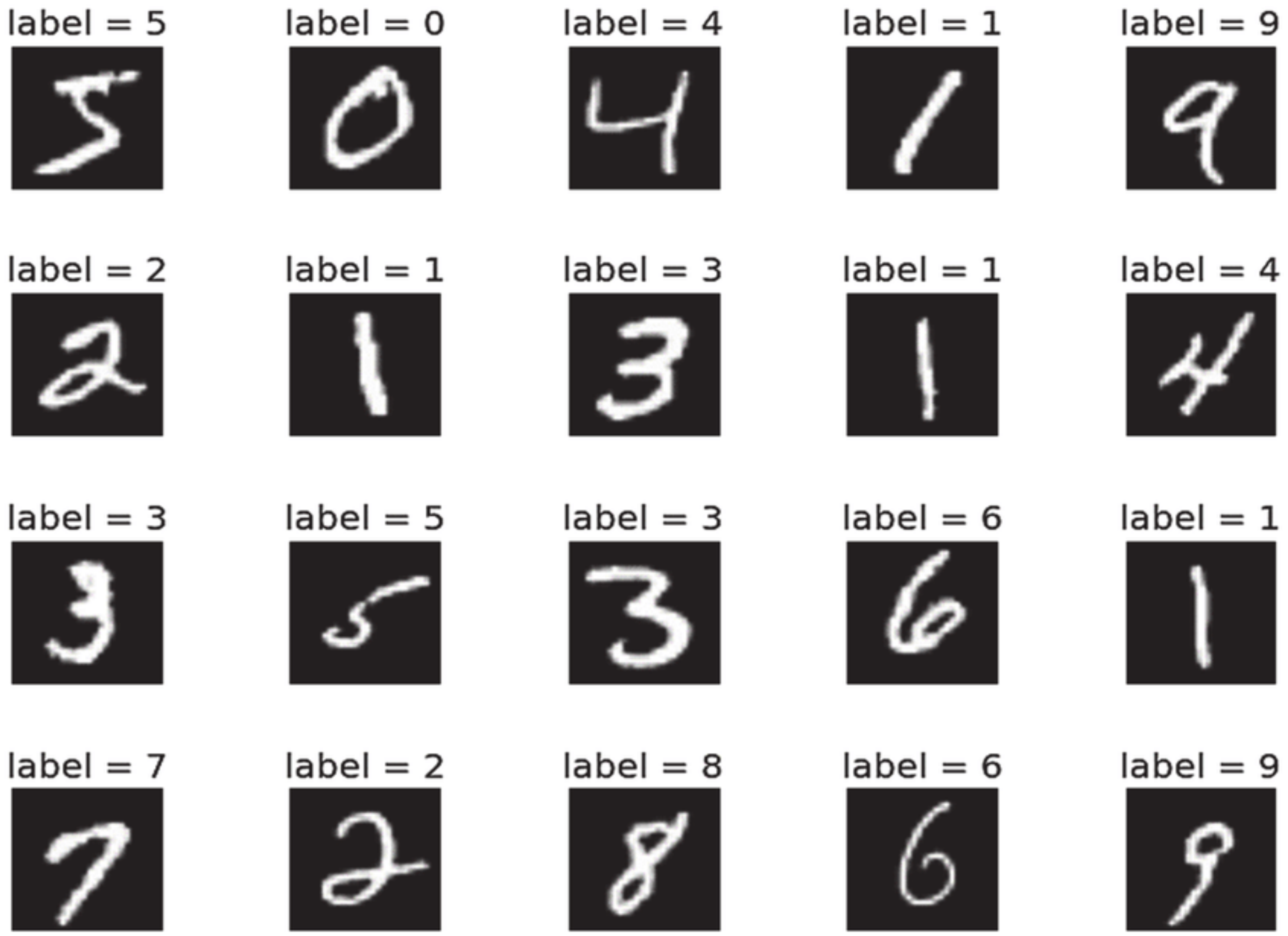
Convolutional neural networks are a specialized type of artificial neural networks that use a mathematical operation called convolution in place of general matrix multiplication in at least one of their layers. They are specifically designed to process pixel data and are used in image recognition and processing.

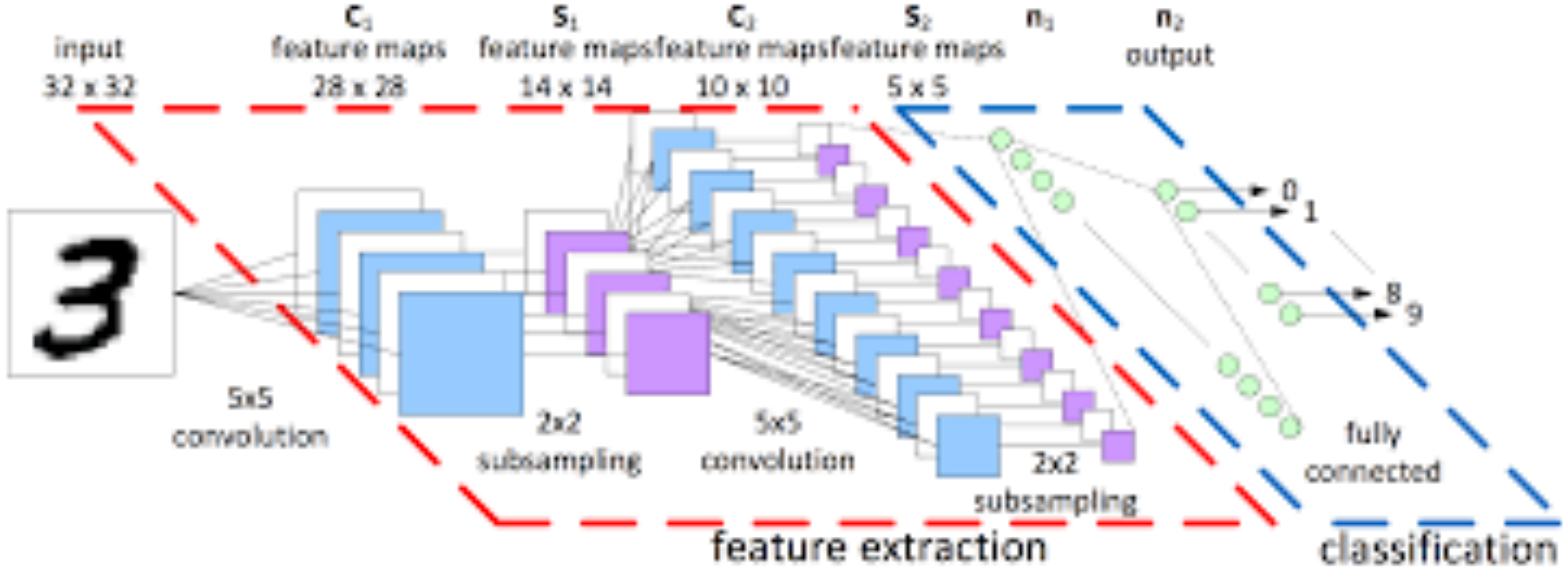
Max Pooling is **a pooling operation that calculates the maximum value for patches of a feature map, and uses it to create a downsampled (pooled) feature map**. It is usually used after a convolutional layer.






Convolution Neural Network (CNN)
Input
Pooling
Pooling
Pooling
Kernel
Convolution + ReLU
Convolution + ReLU
Convolution + ReLU
Feature Maps
Flatten Layer
Fully Connected Layer
Output
0.2
0.7
0.1
Horse
Zebra
Dog
SoftMax Activation Function
Feature Extraction
Classification
Probabilistic Distribution

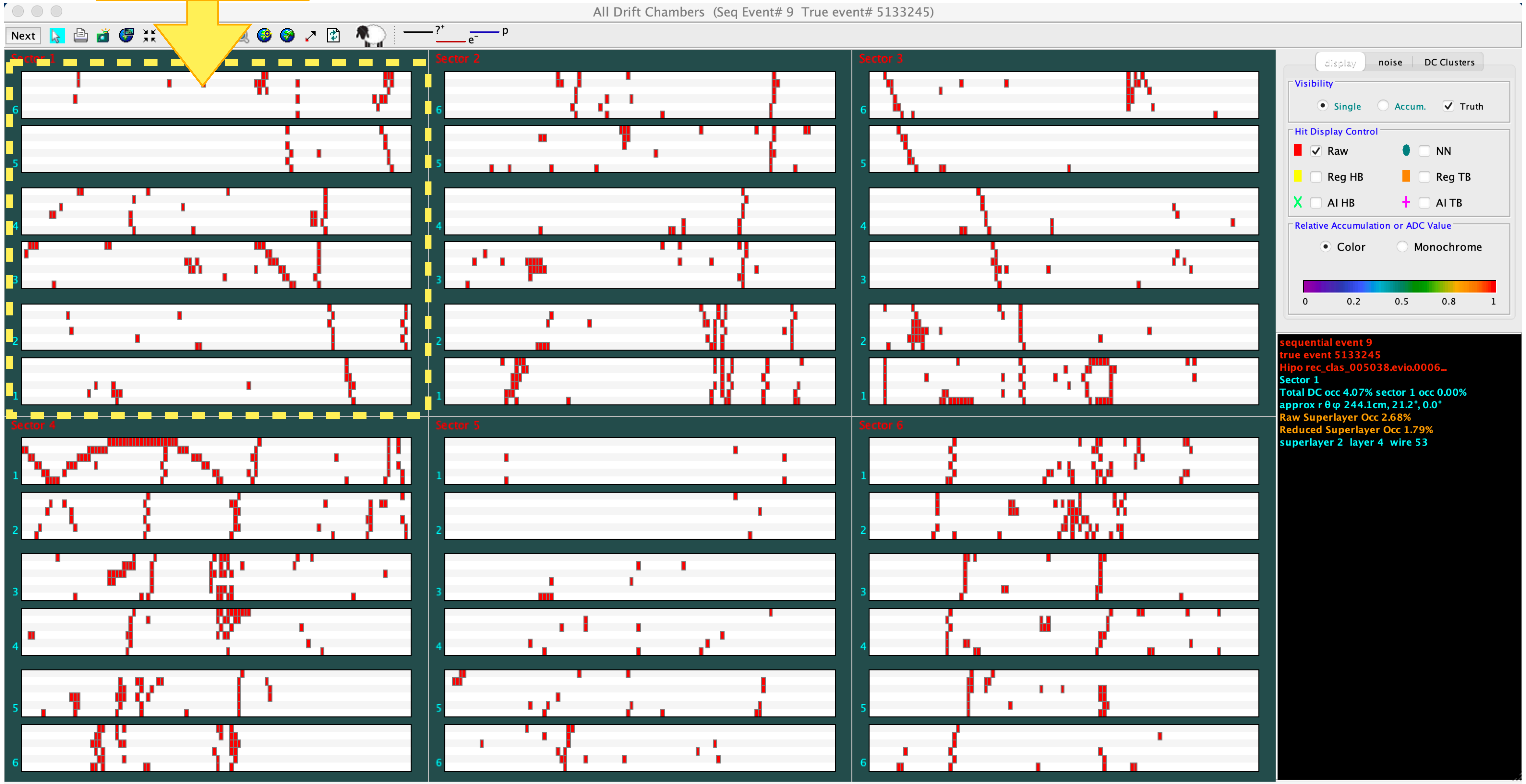

label = 5
label = 0
label = 4
label = 1
label = 9
label = 2
label = 1
label = 3
label = 1
label = 4
label = 3
label = 5
label = 3
label = 6
label = 1
label = 7
label = 2
label = 8
label = 6
label = 9


input
32 x 32
C1
feature maps
28 x 28
S1
feature maps
14 x 14
C2
feature maps
10 x 10
S2
feature maps
5 x 5
n1
n2
output
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
feature extraction
classification

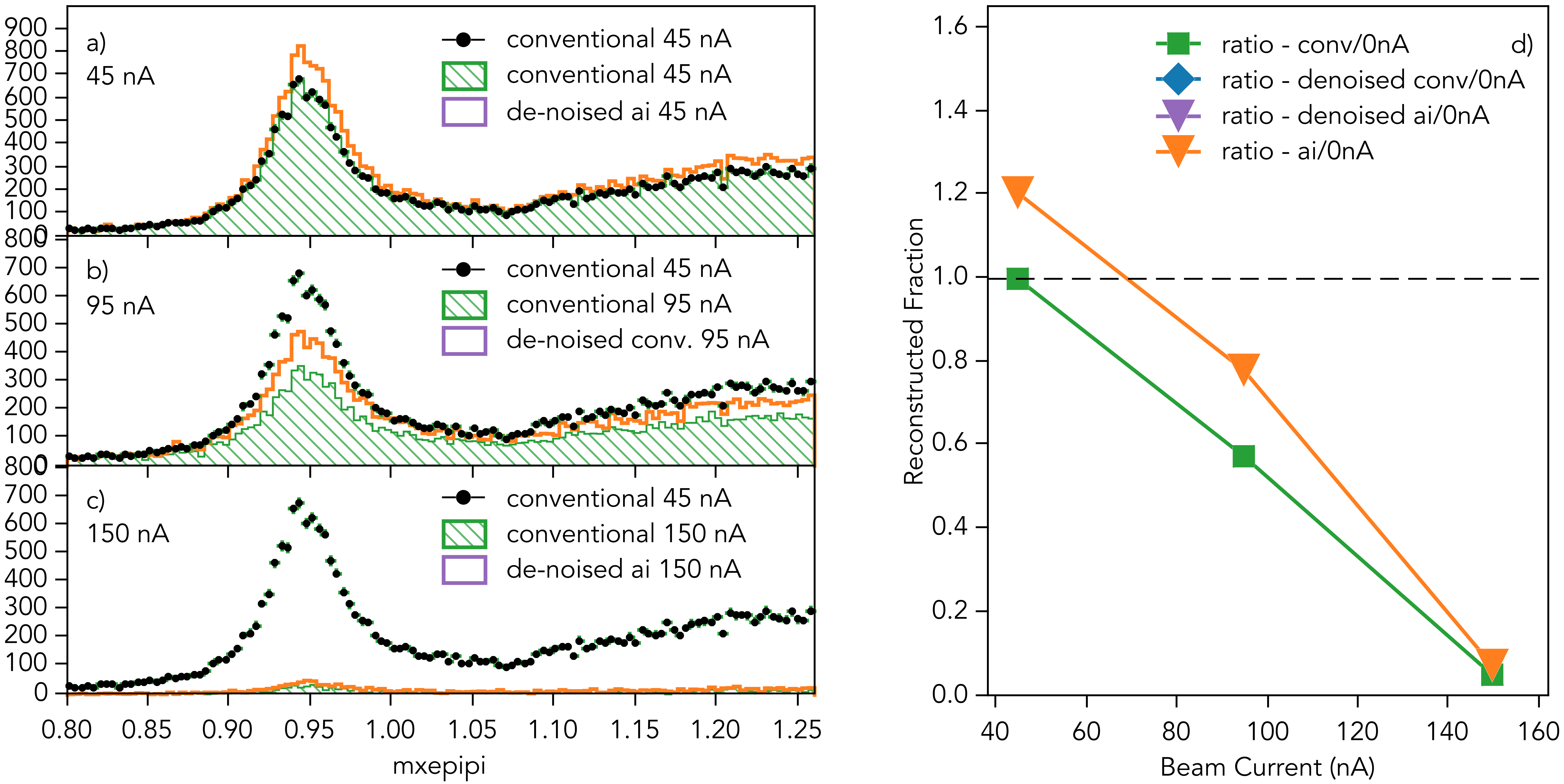
CLAS12 tracking

Sector 1

Six sectors shown

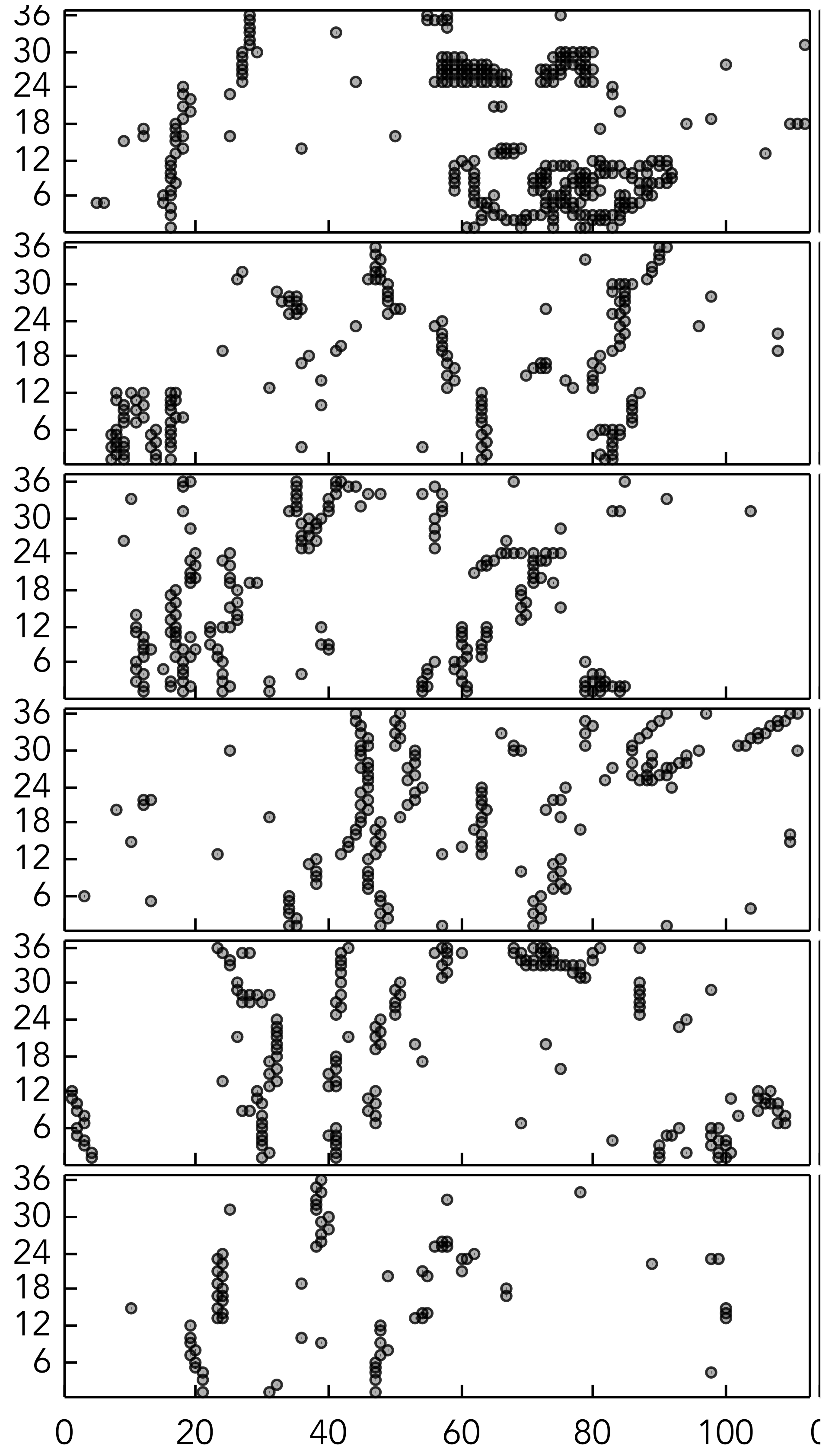


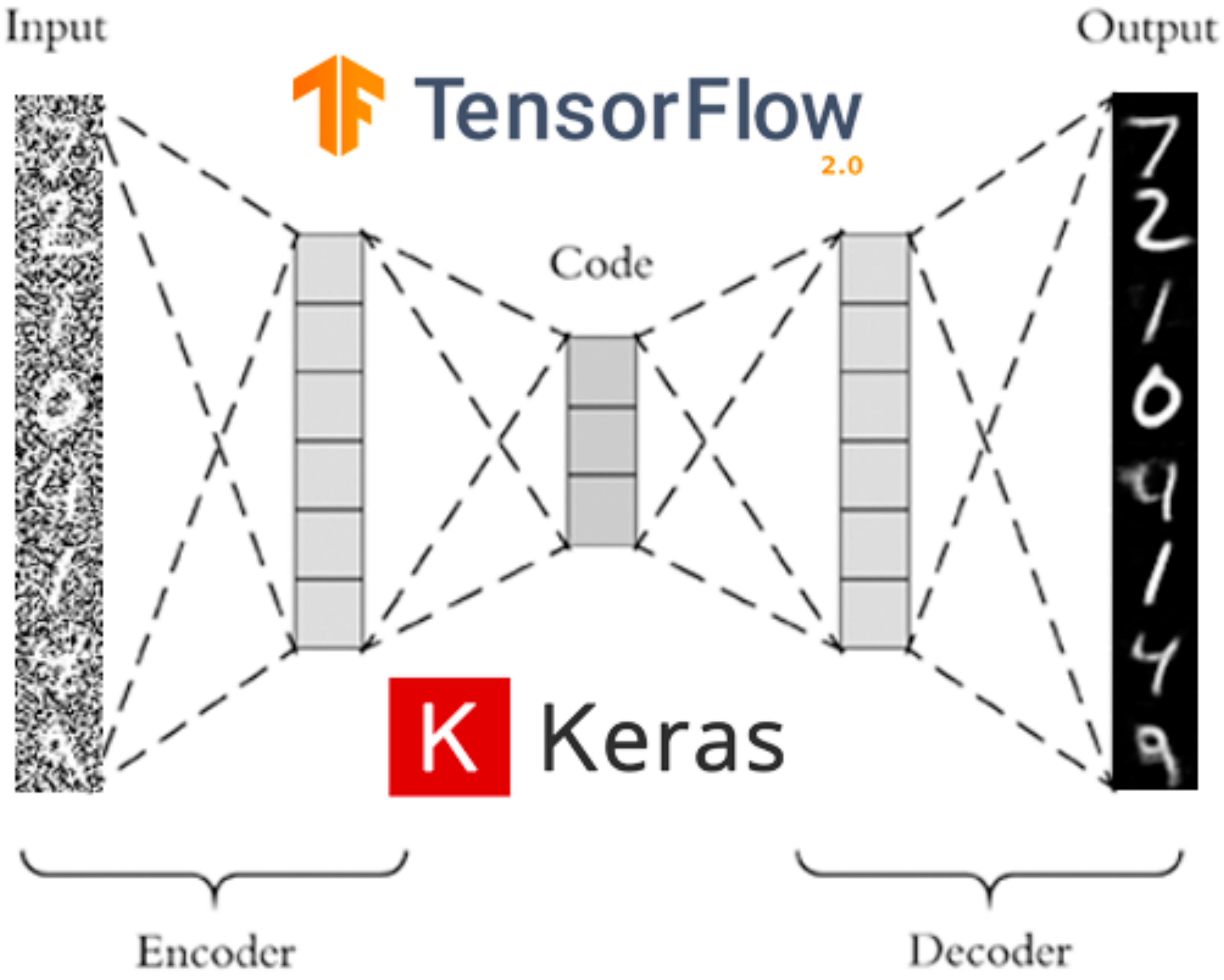
Increasing beam current (luminosity) increases the number of background segments and combinatorics. (45 nA is the nominal running current)

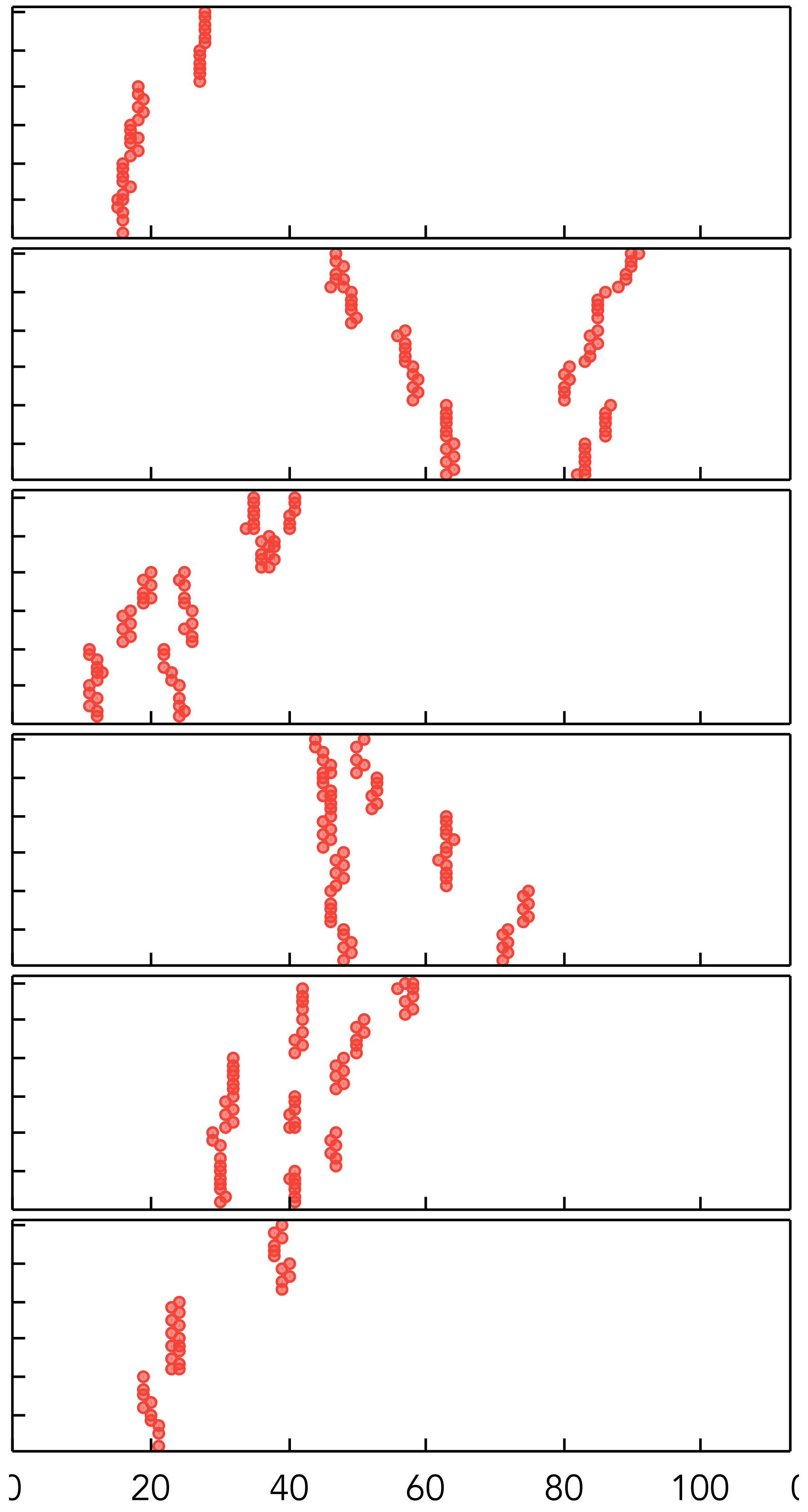


At high beam currents, AI-assisted tracking drops to the level of conventional.

- Convolutional Auto-Encoder is used to de-noise raw data from drift chambers.
- Network is trained of reconstructed data with track hits isolated from raw DC hits.







True probability distribution (one-hot)

$$H(p,q) = -\sum_{x \in \text{classes}} p(x) \log q(x)$$

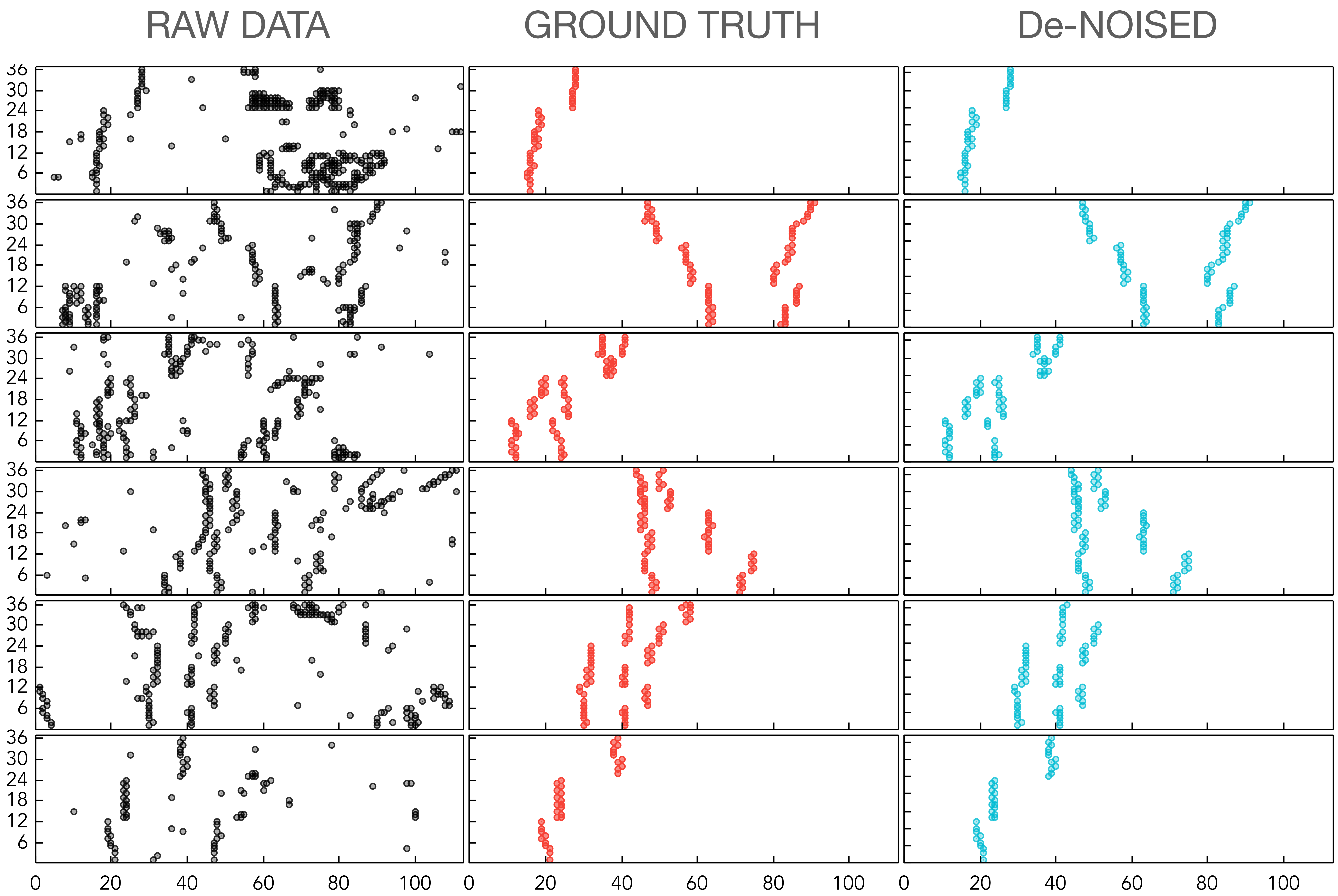
Your model's predicted probability distribution

For logistics problems Cross-Entropy Loss function is better
The penalty of not getting binary outcome is logarithmic

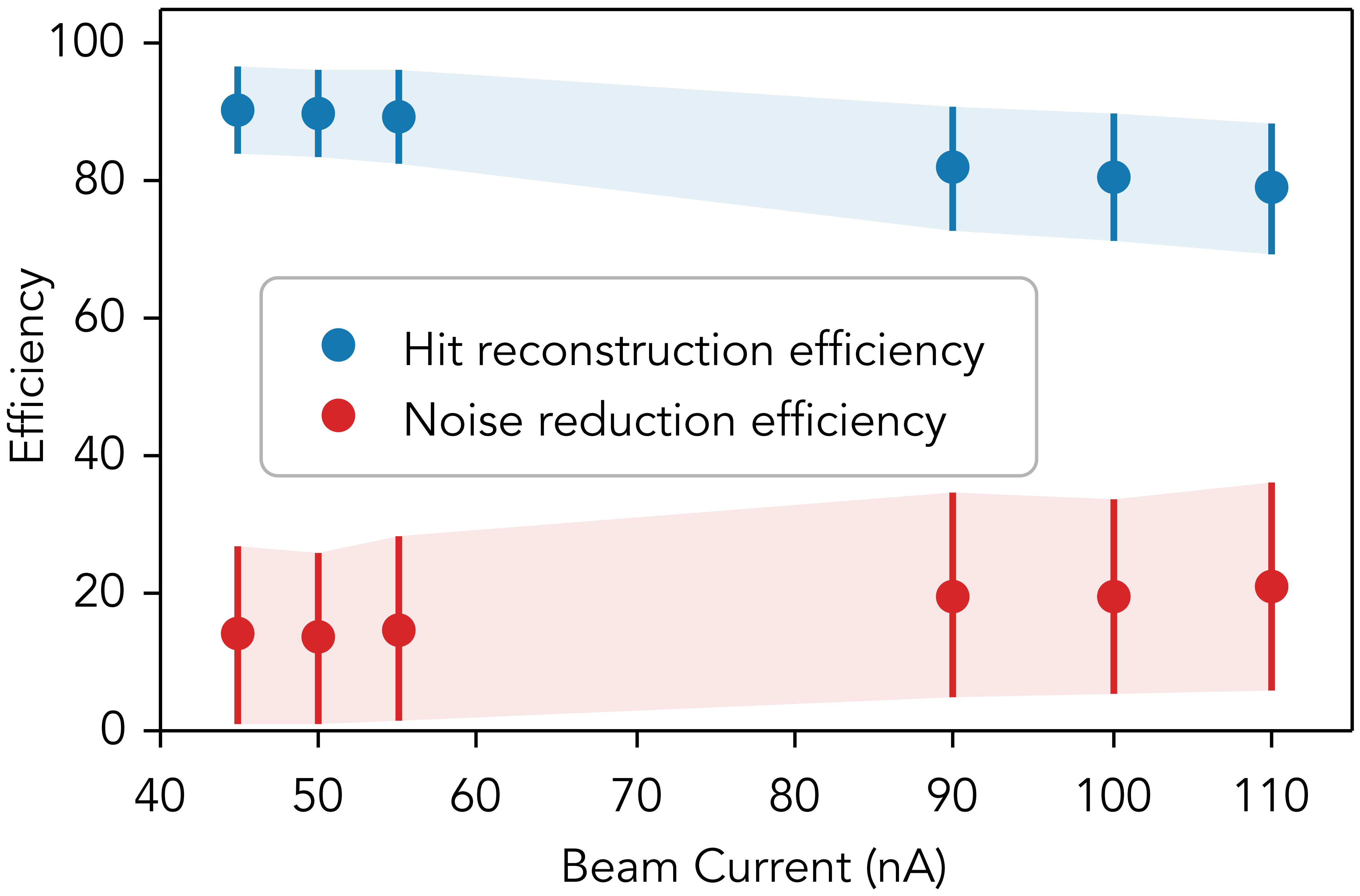
$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

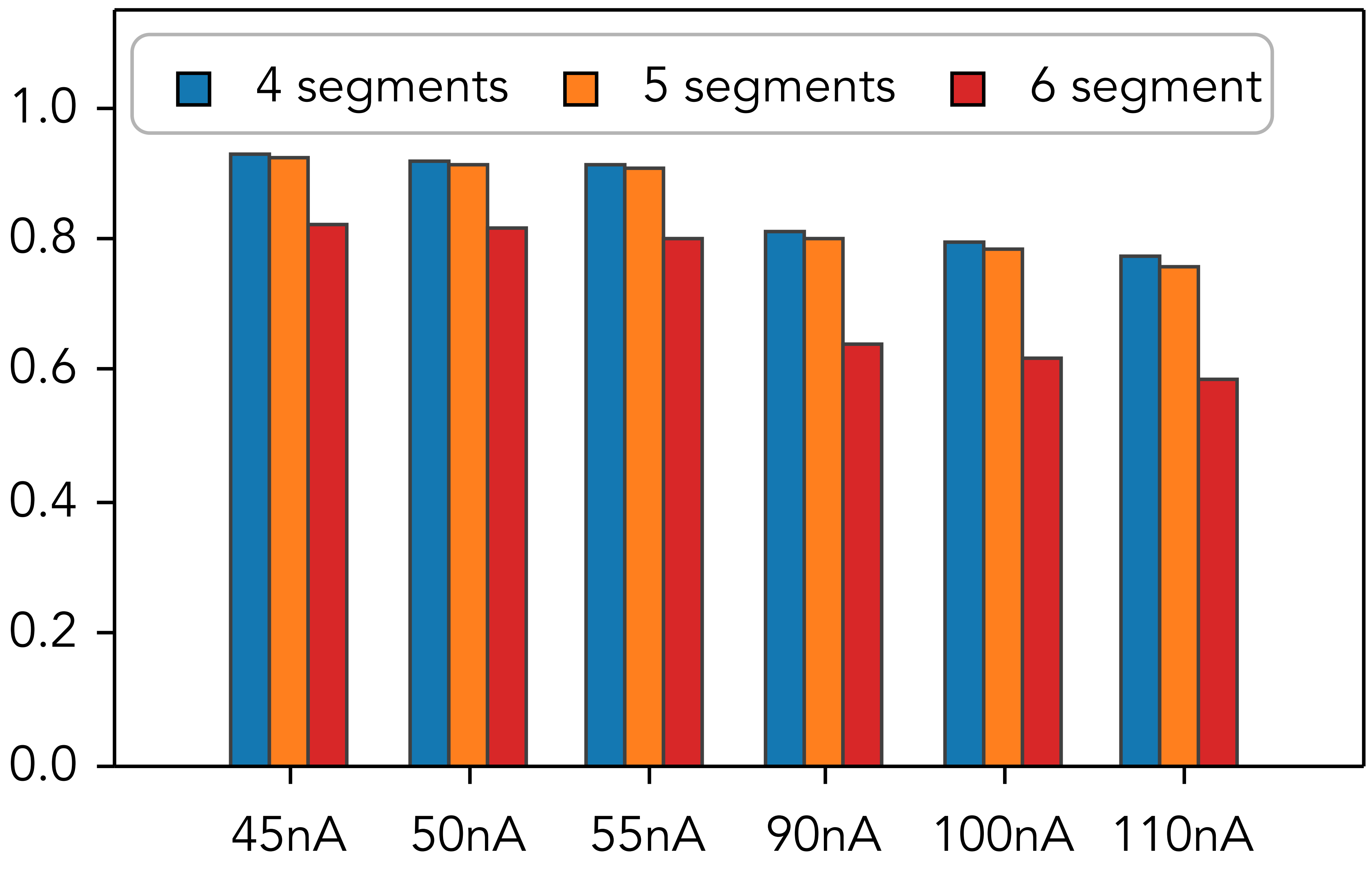
MSE is good for linear regression problems

CLAS12 tracking

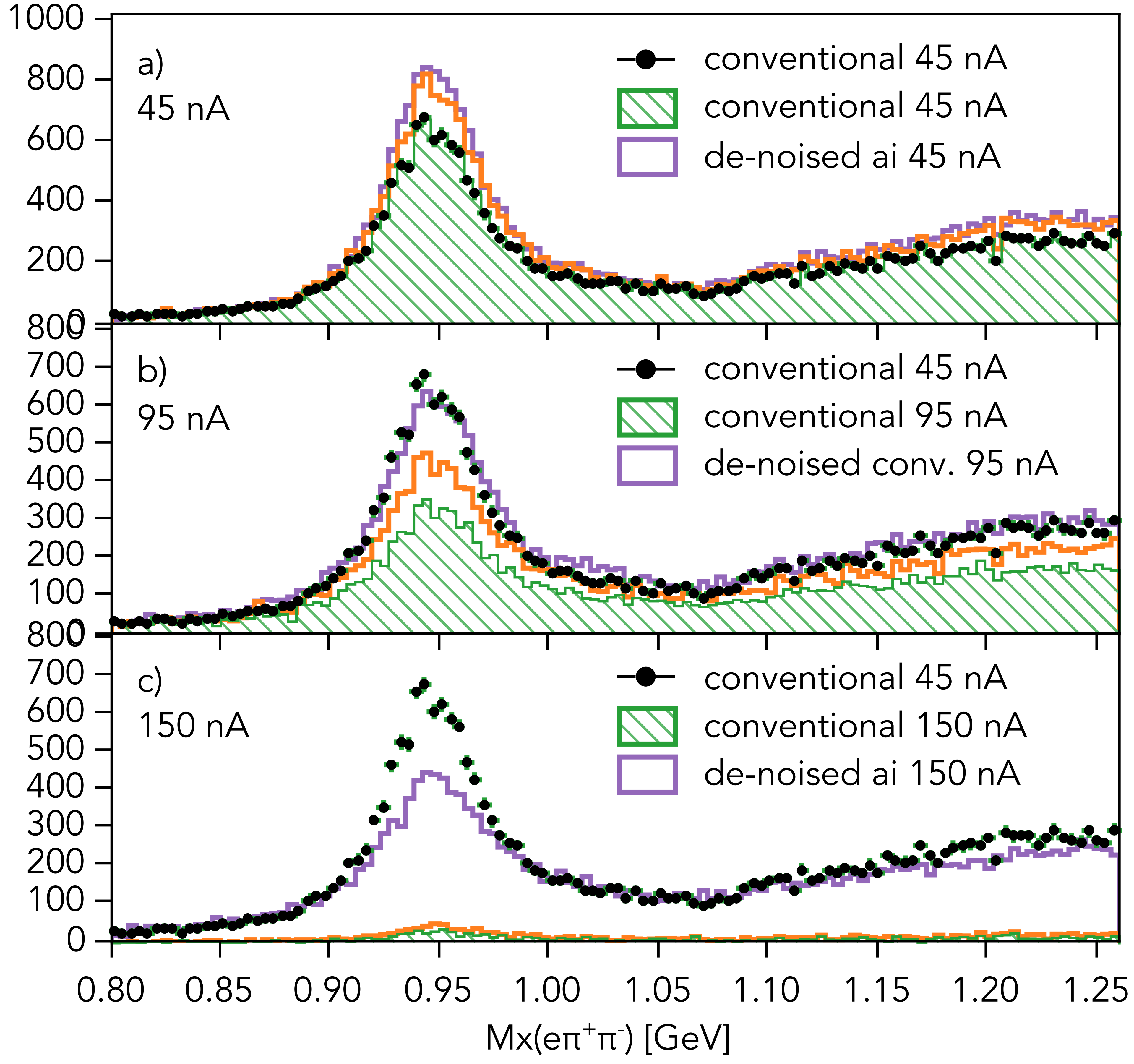


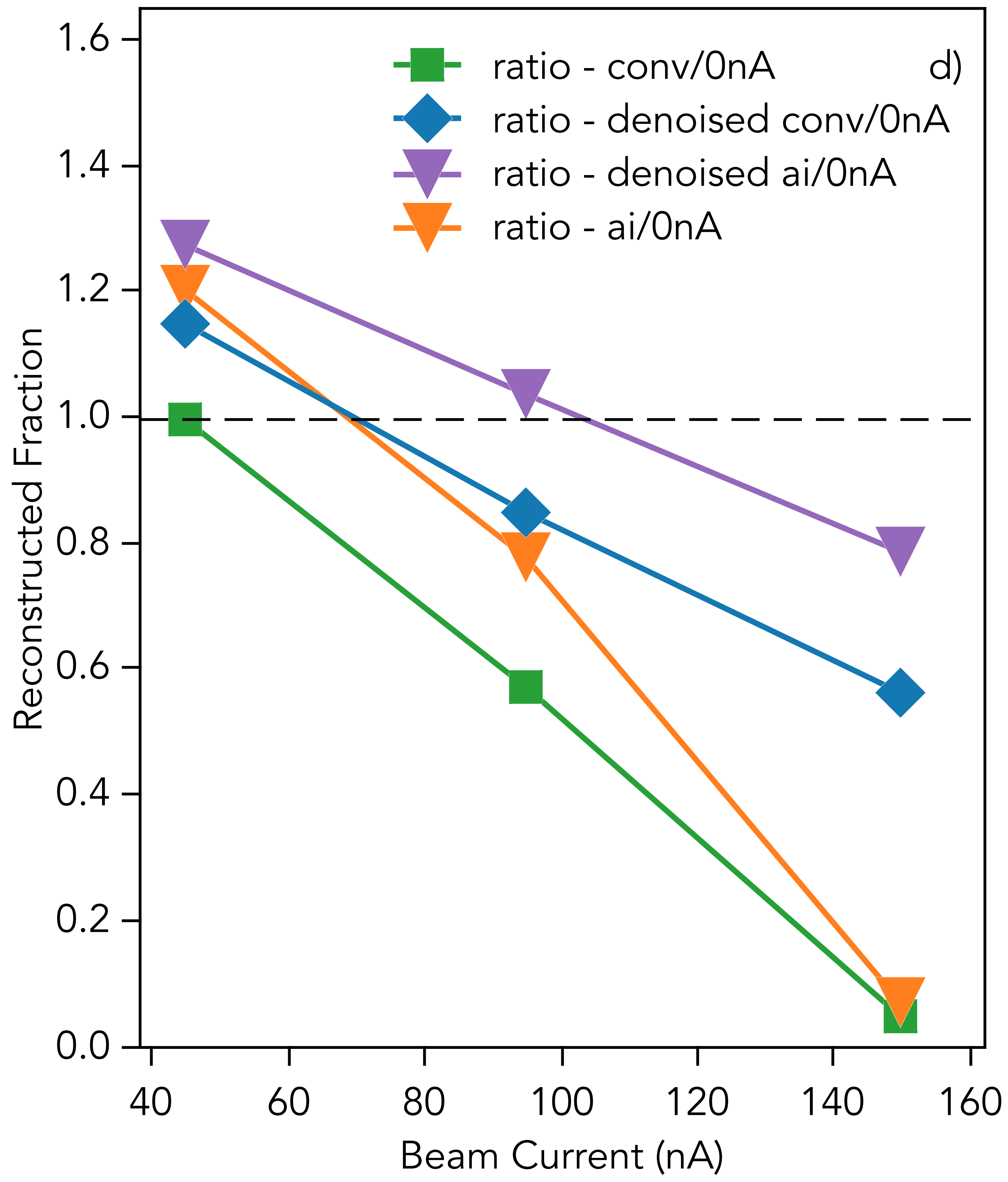
De-noising (CLAS12 Drift Chambers)



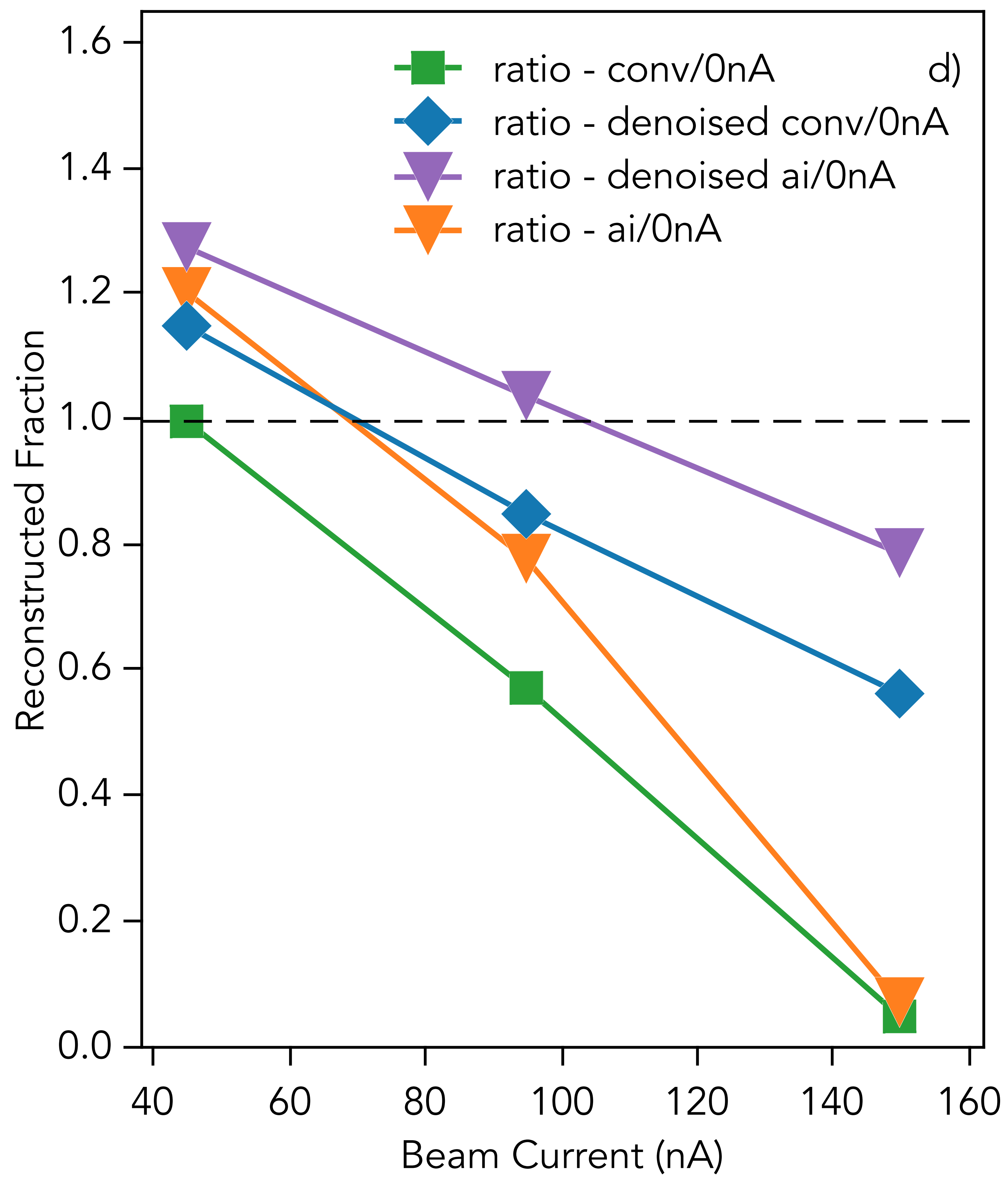


We used Convolutional De-noiser and MLP classifier together and separately





We used Convolutional De-noiser and MLP classifier together and separately



- The efficiency at 100 nA is higher than conventional at 45 nA
- If running an experiment at 100 nA, the same data can be collected less than half the time.
- With the same running time the statistics more than doubles.

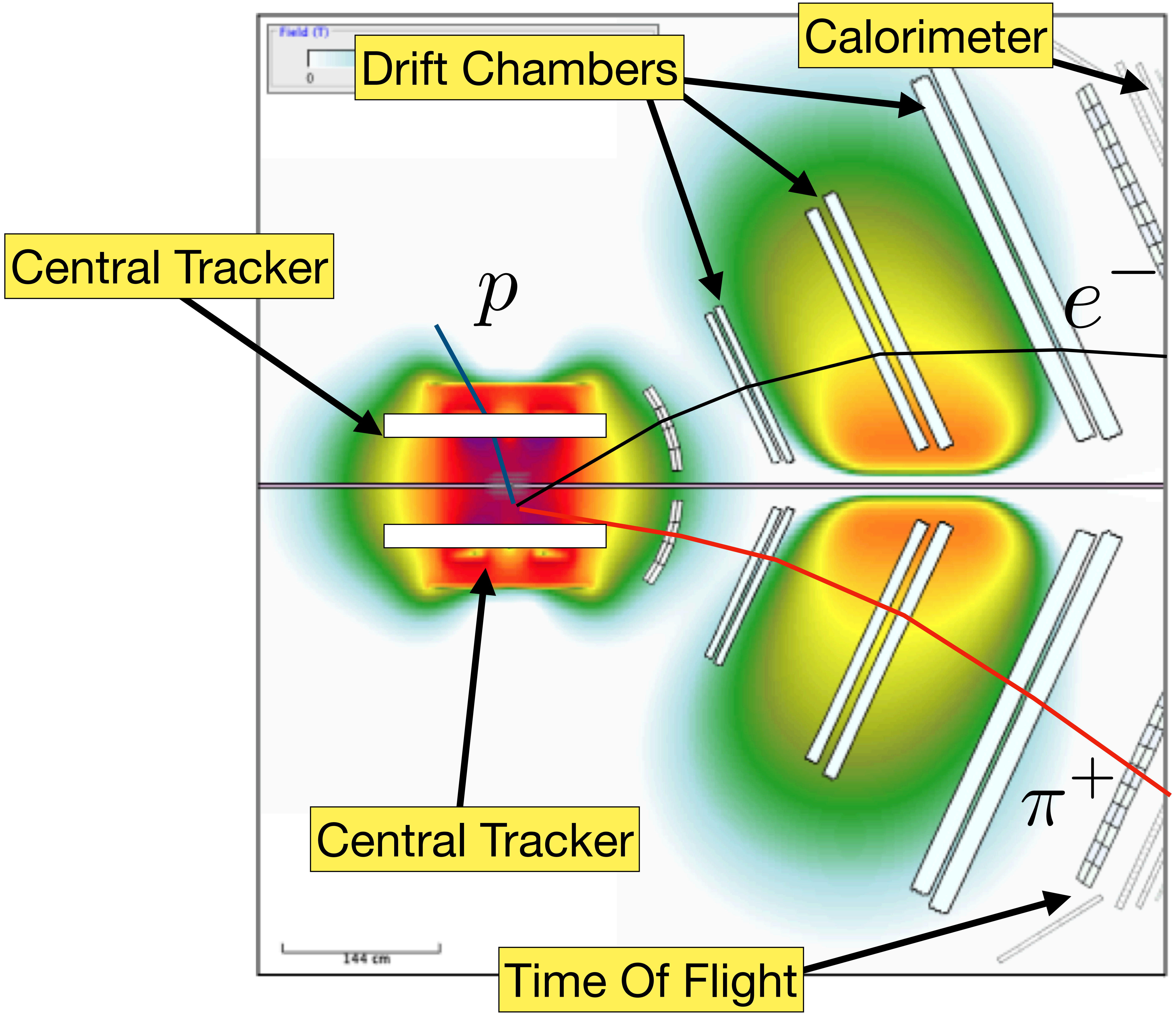
It was 35% increase in statistics with AI track classifier

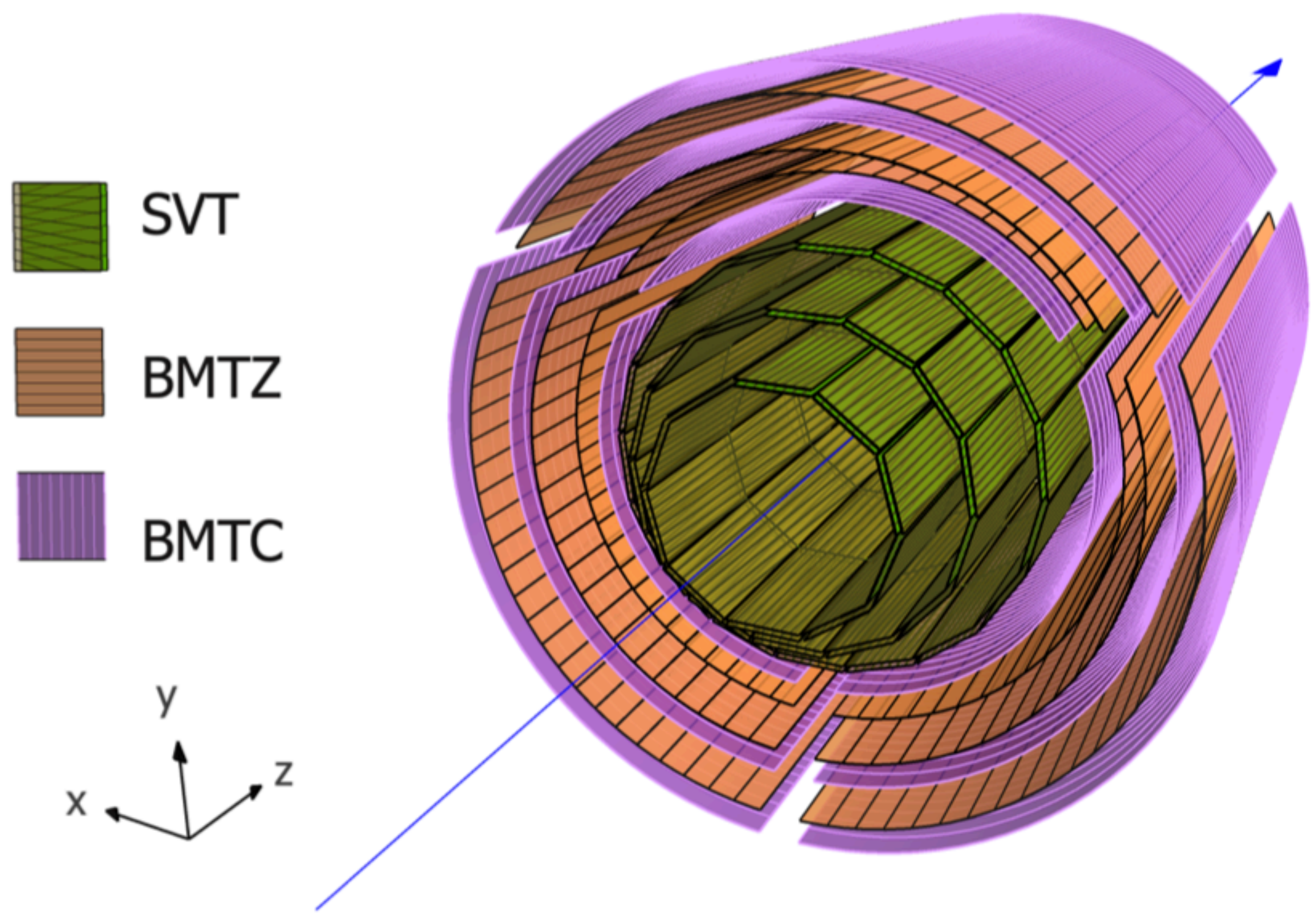
Now we get 2.2x statistics (120% increase)

Convolutional Neural Networks (CNN)
Logistic Regression

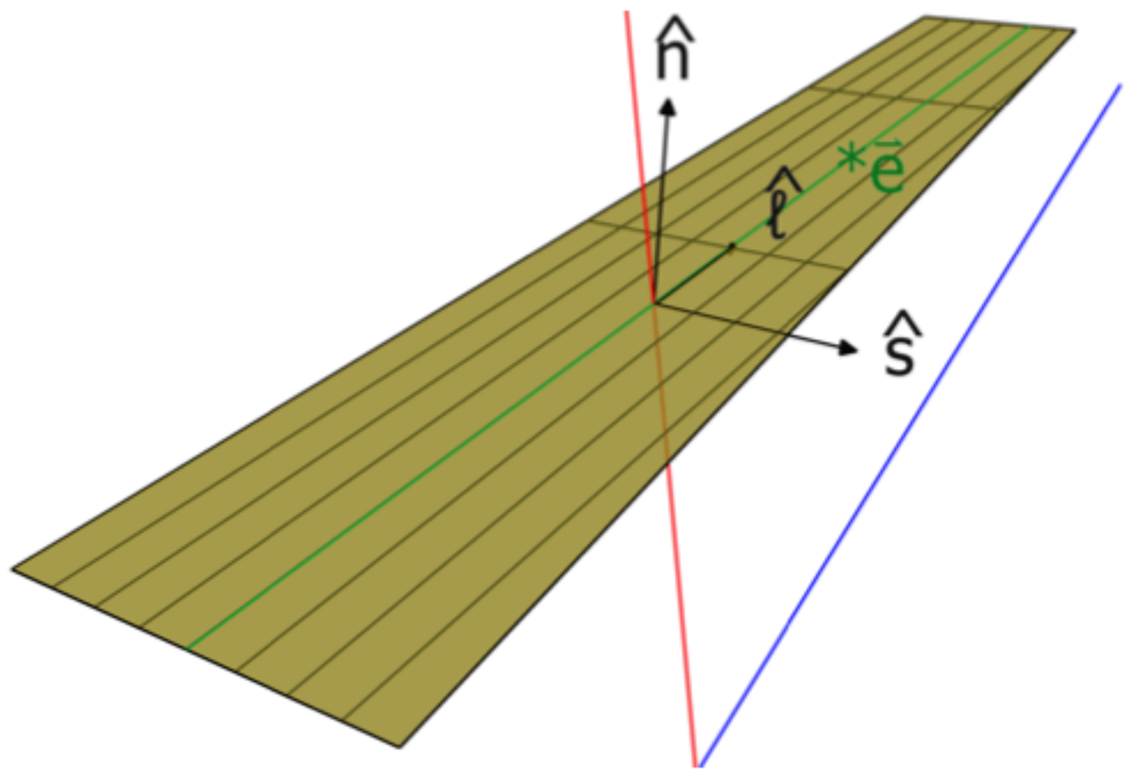
Machine Learning

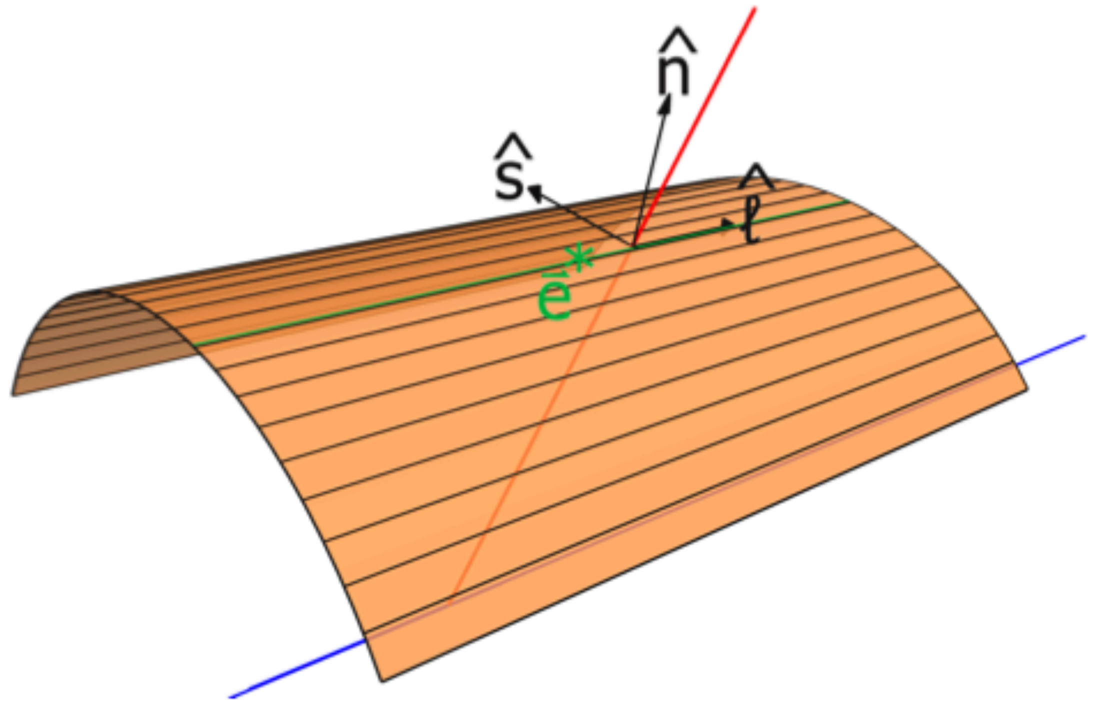
- CLAS12 detector
  - Forward Drift Chambers:
    - In toroid magnetic field (6 sectors)
    - 6 super layers
    - 6 wire planes in each super-layer
  - Central Tracker:
    - Barrel Micromega Trackers
    - 3 CVT barrels
    - 3 Z-plane detector layers
    - 3 Phi plane detector layers
- Data Reconstruction
  - Reconstructing tracks from the detector responses takes 750 ms in a single thread.
  - Data is collected at the rate of 12kHz
  - Takes about 4-6 months to process data collected in 1 month.
  - Track reconstruction is 90% of the computational time.

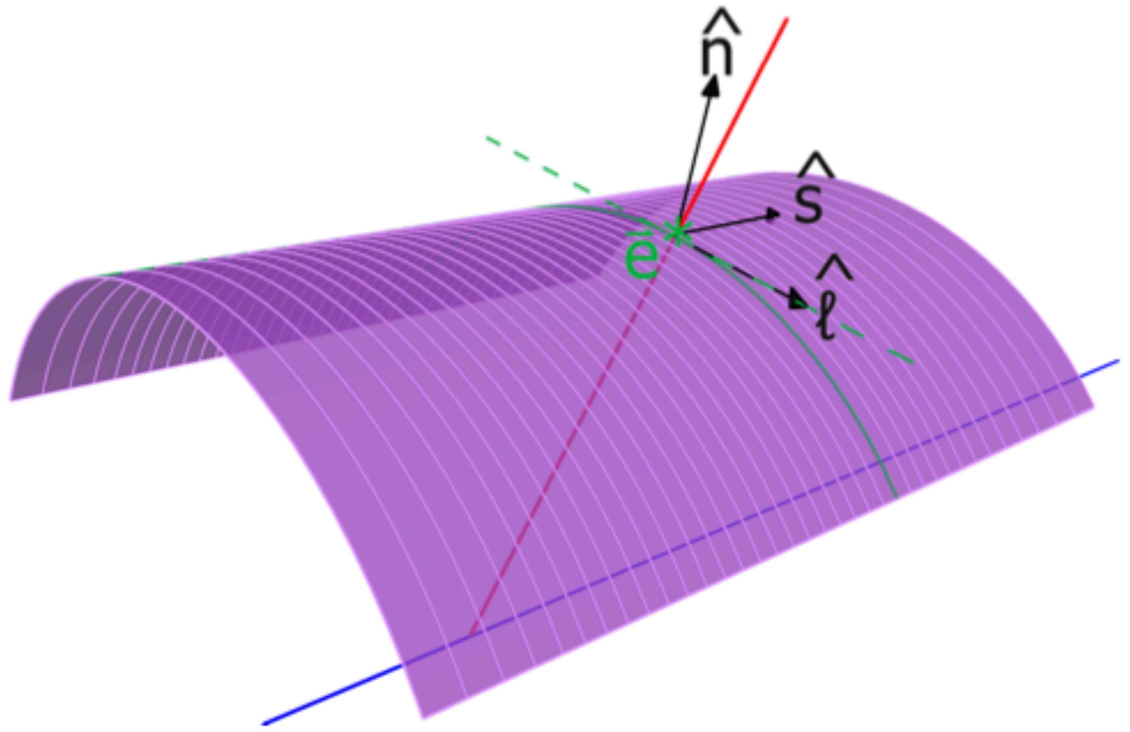


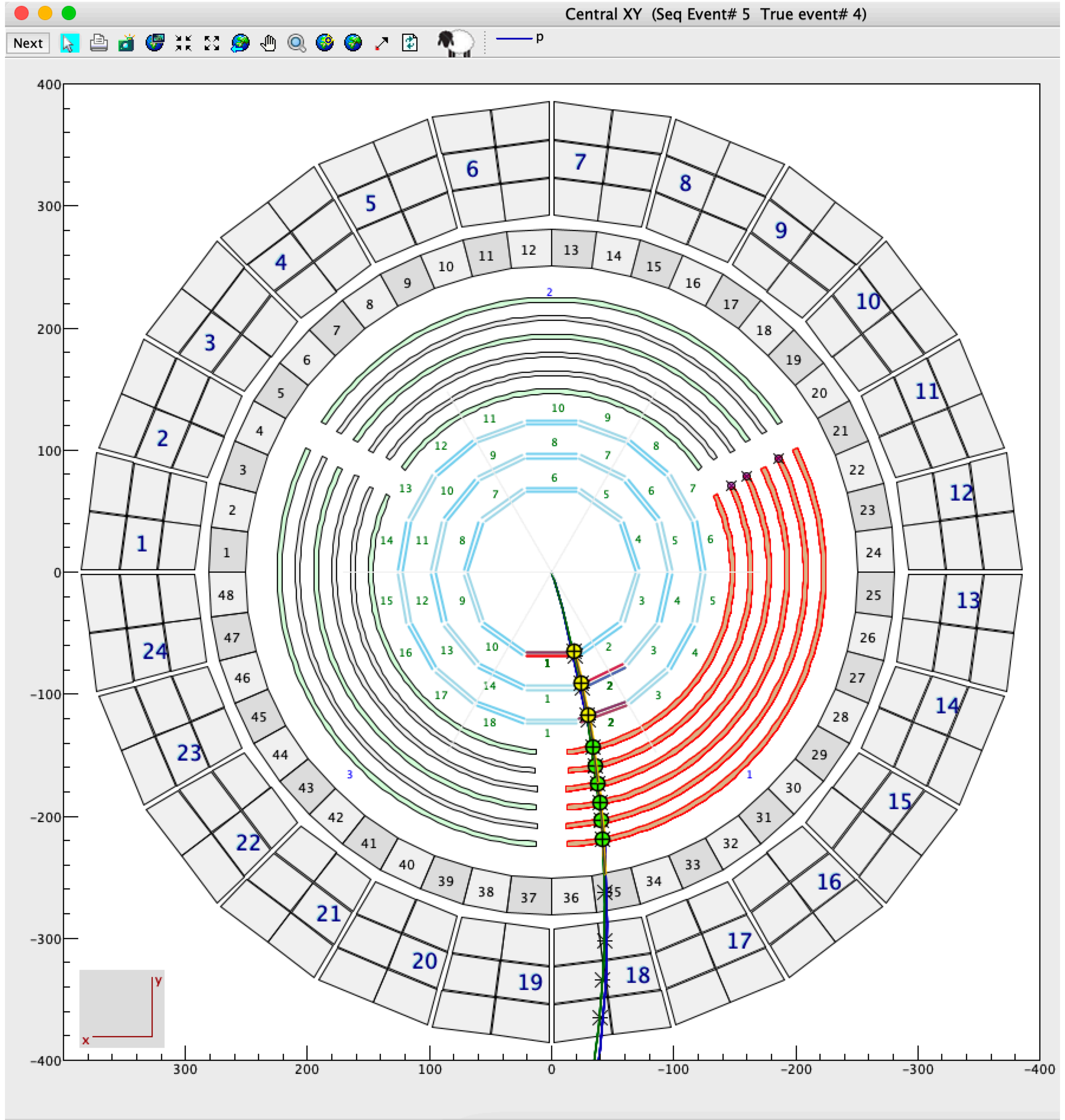


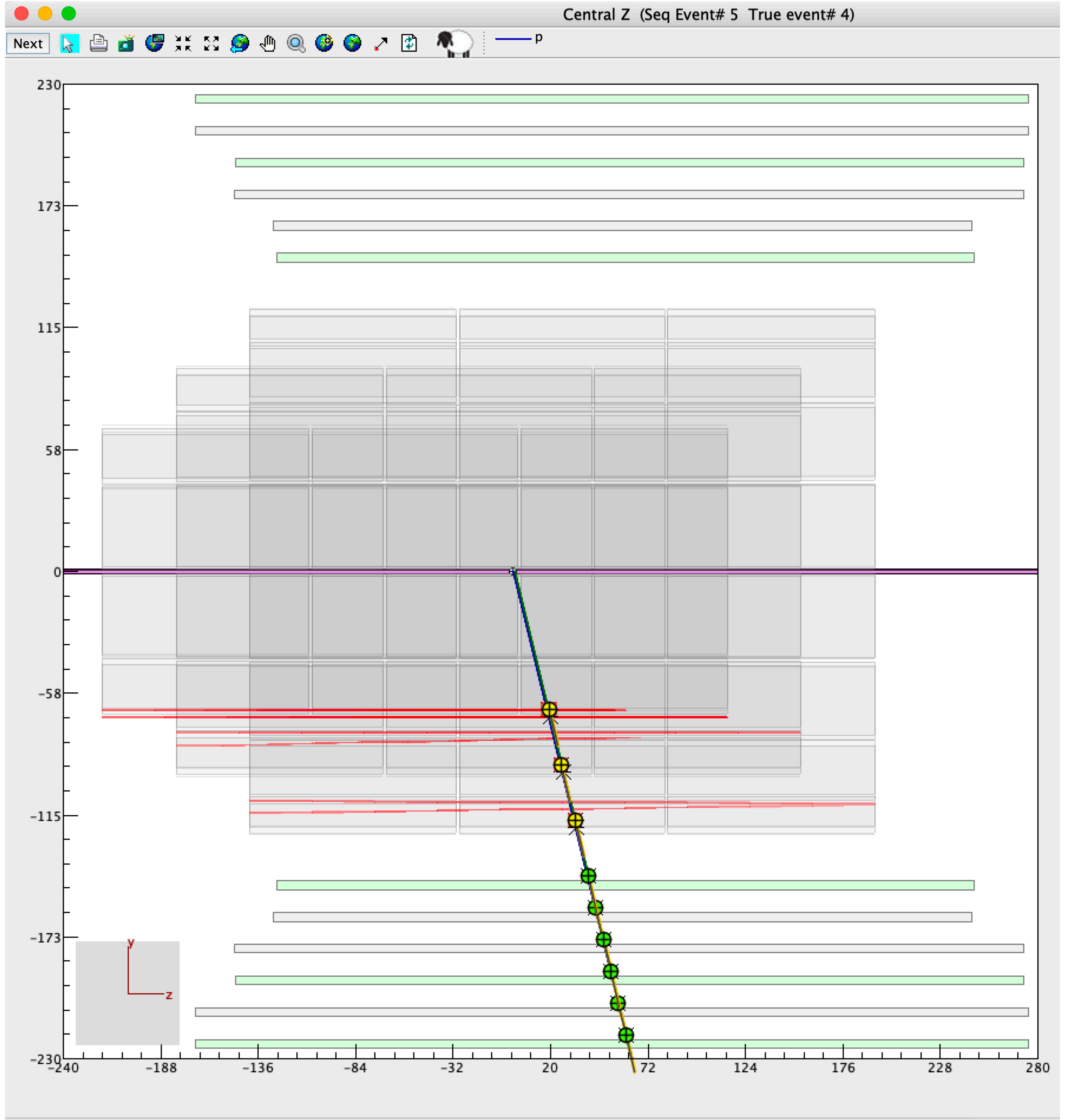
- Central Barrel Detector
  - 3 rings of CVT
  - 10, 14, and 18 double paddles in each ring
  - 3 Layers of micro-megas with scripts in the Z direction
  - 3 Layers of micro mega with strips in the shape of an arc (C-Layers)
- Track reconstruction
  - Relies on signals from both detector types
  - At least 3 points along the trajectory (CVT and Z)
  - At least on C point

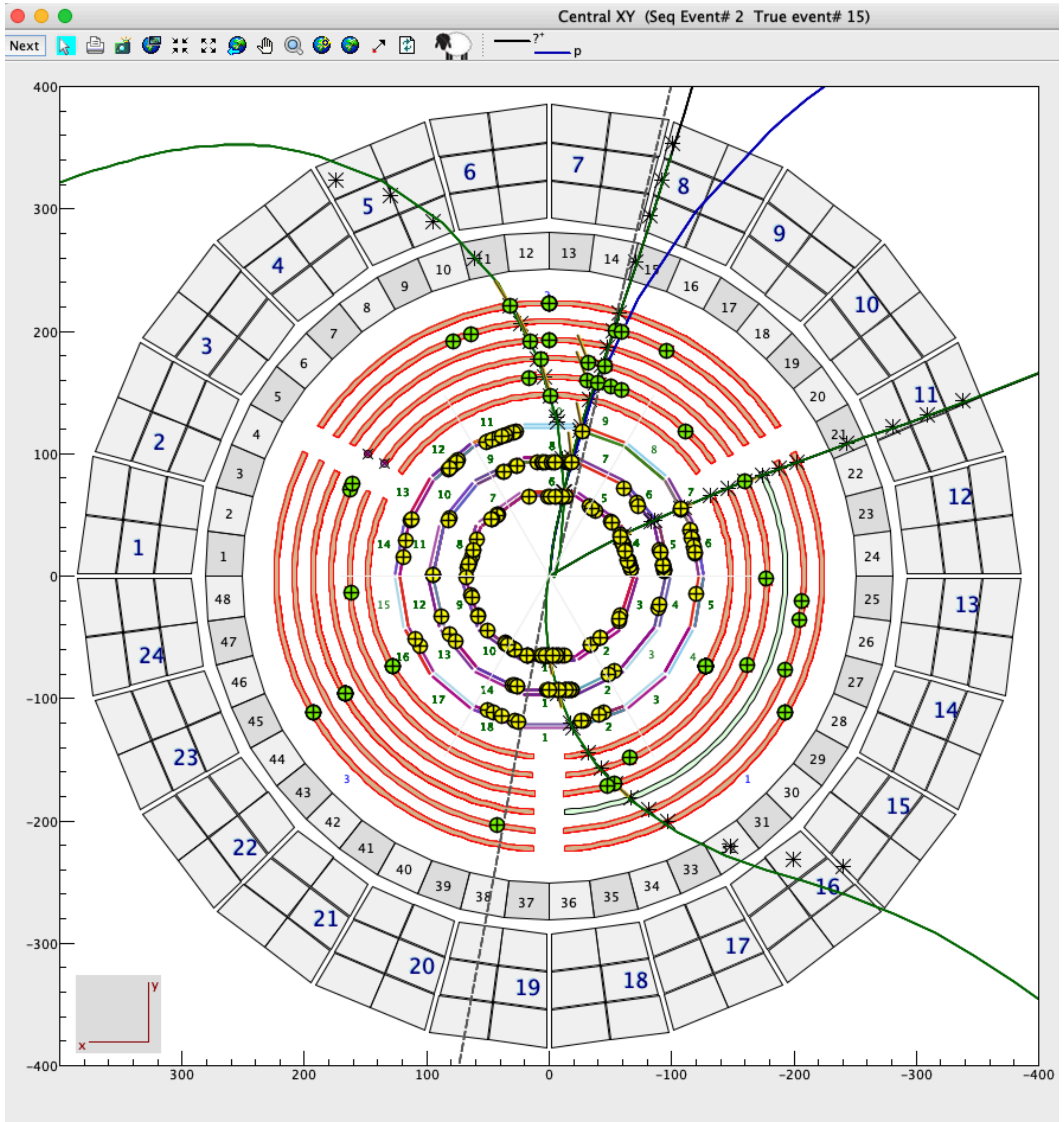




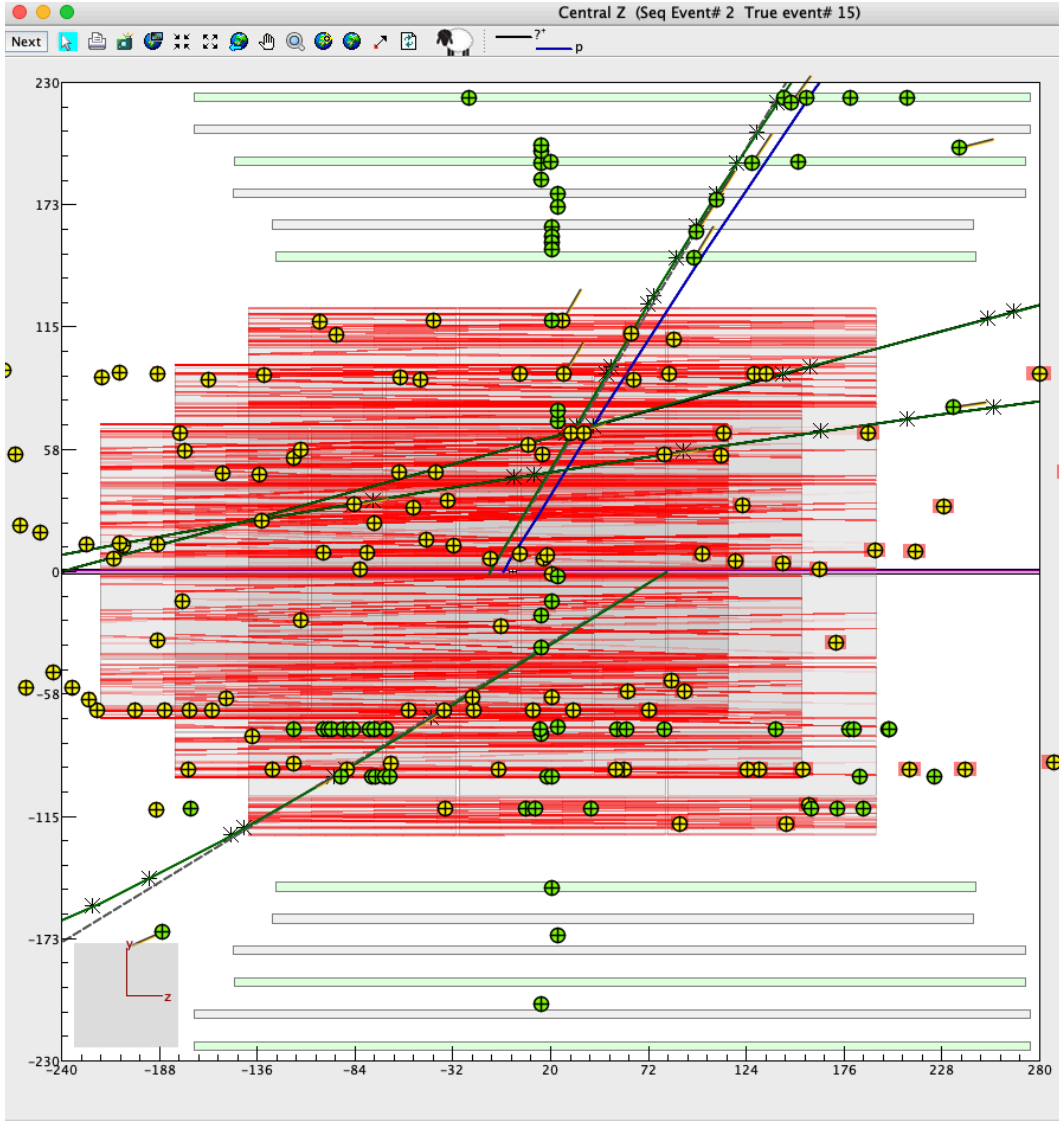



Central XY (Seq Event# 5 True event# 4)

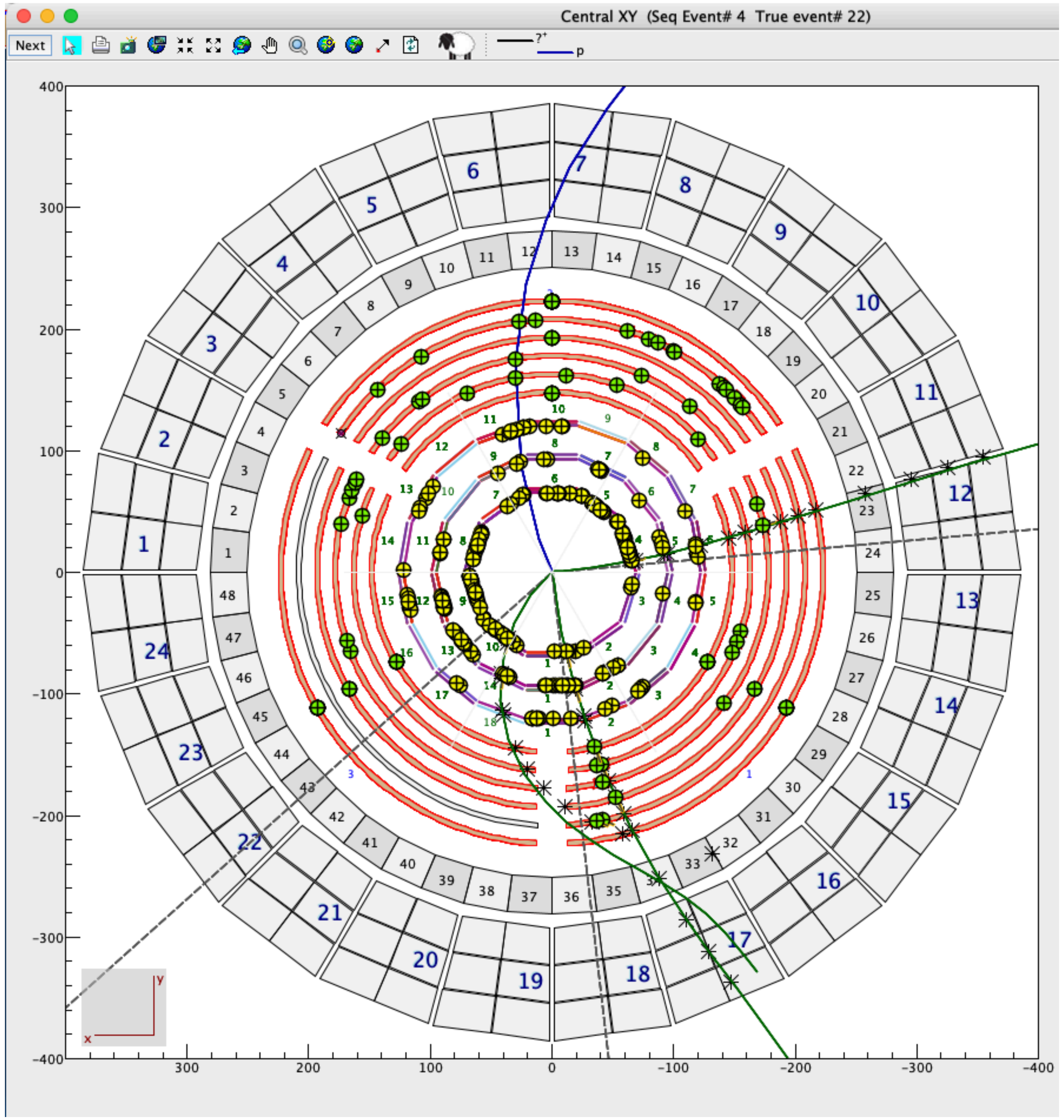

Central Z (Seq Event# 5 True event# 4)



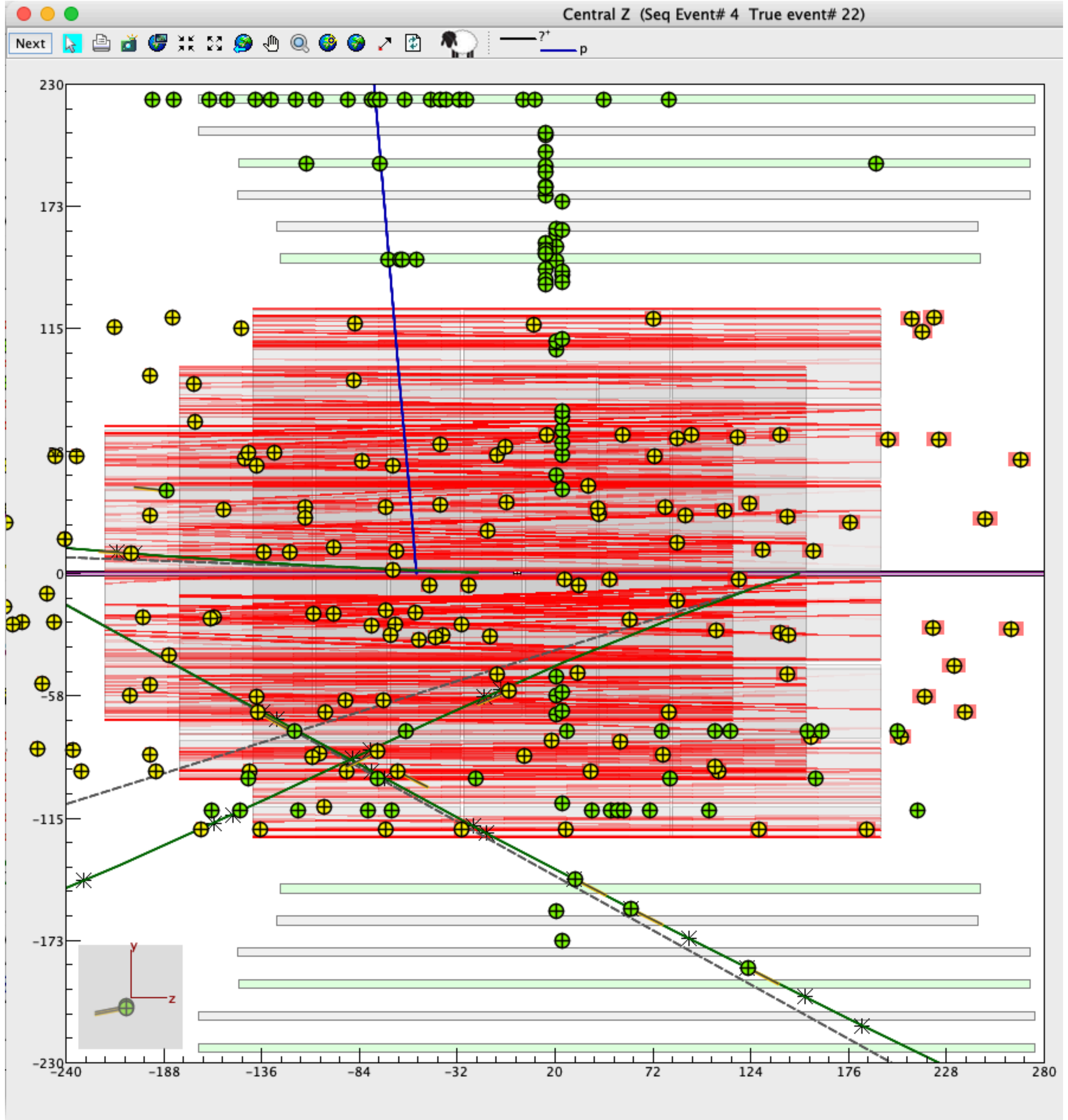
Central XY (Seq Event# 2 True event# 15)
Next



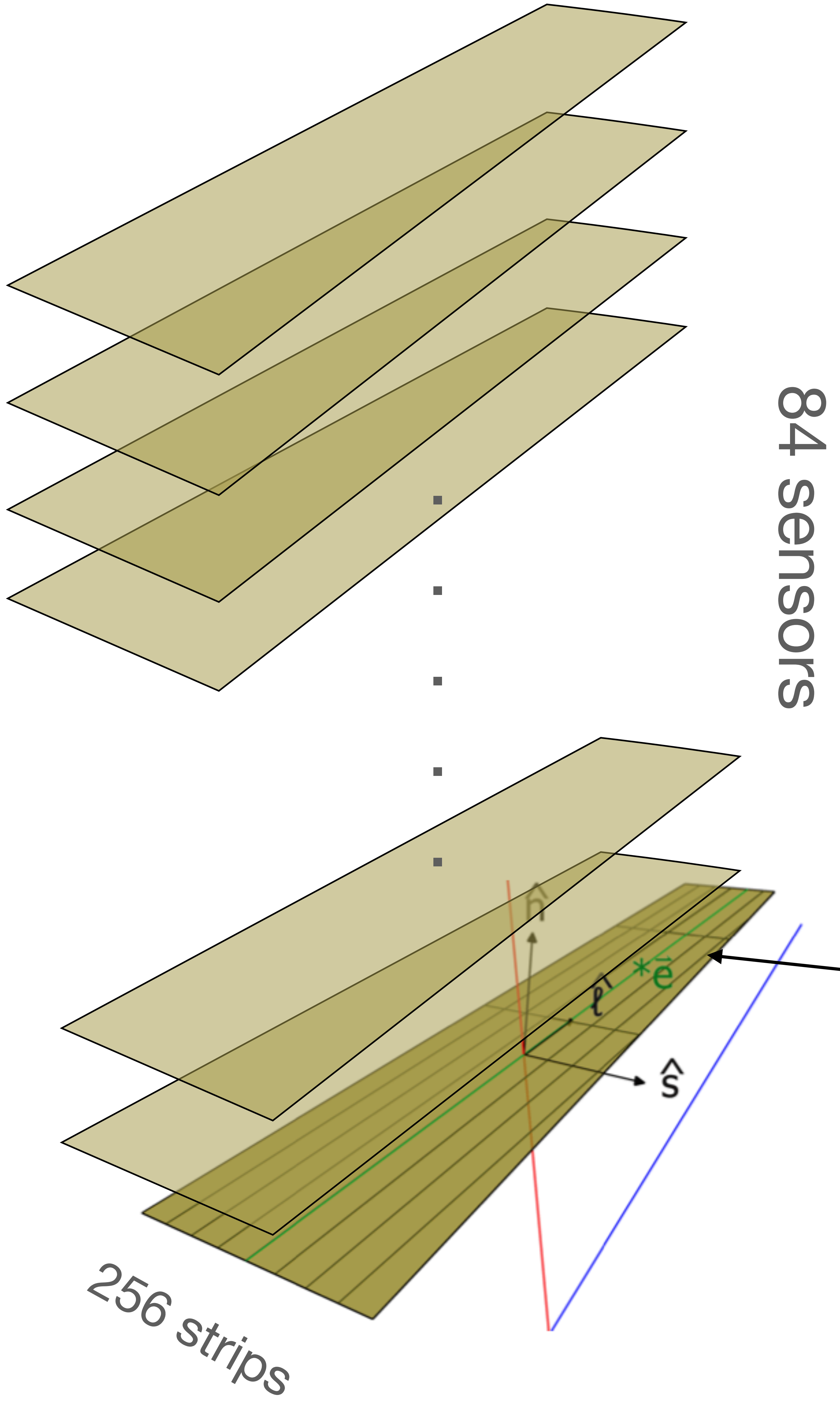
Central Z (Seq Event# 2 True event# 15)
Next

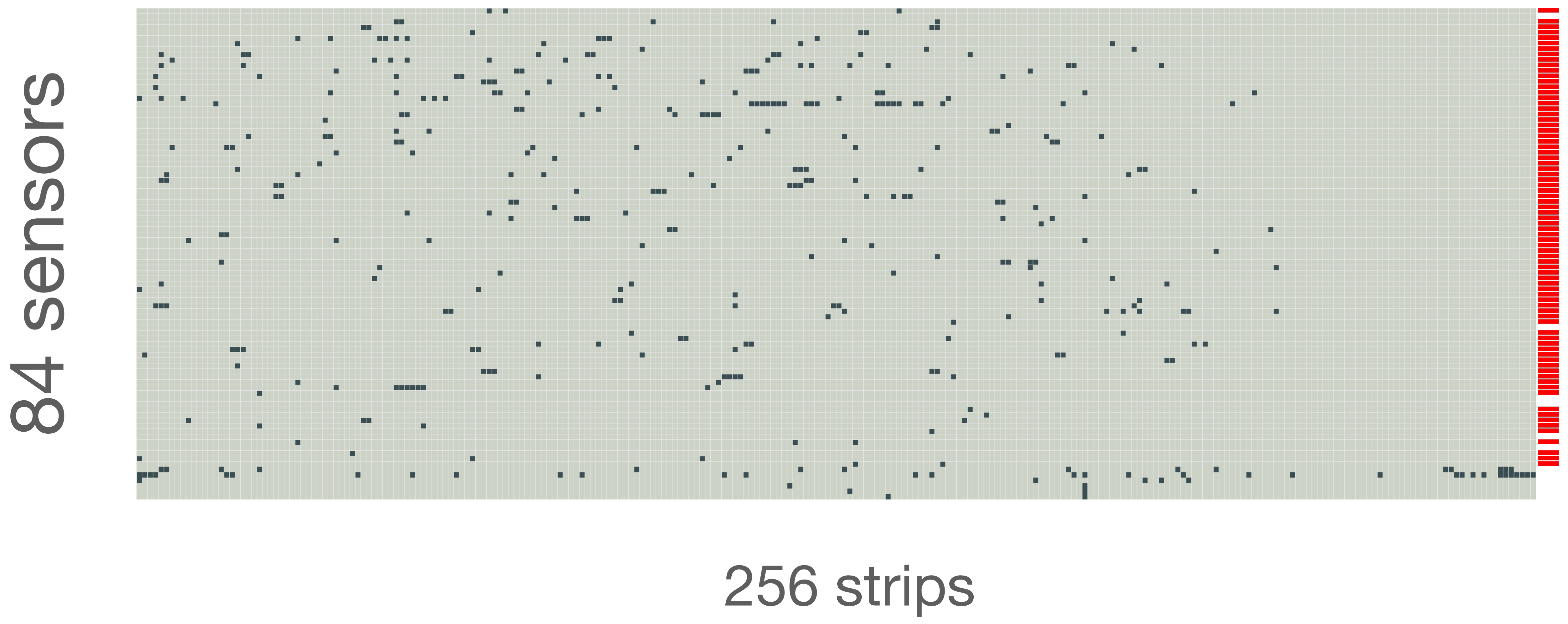


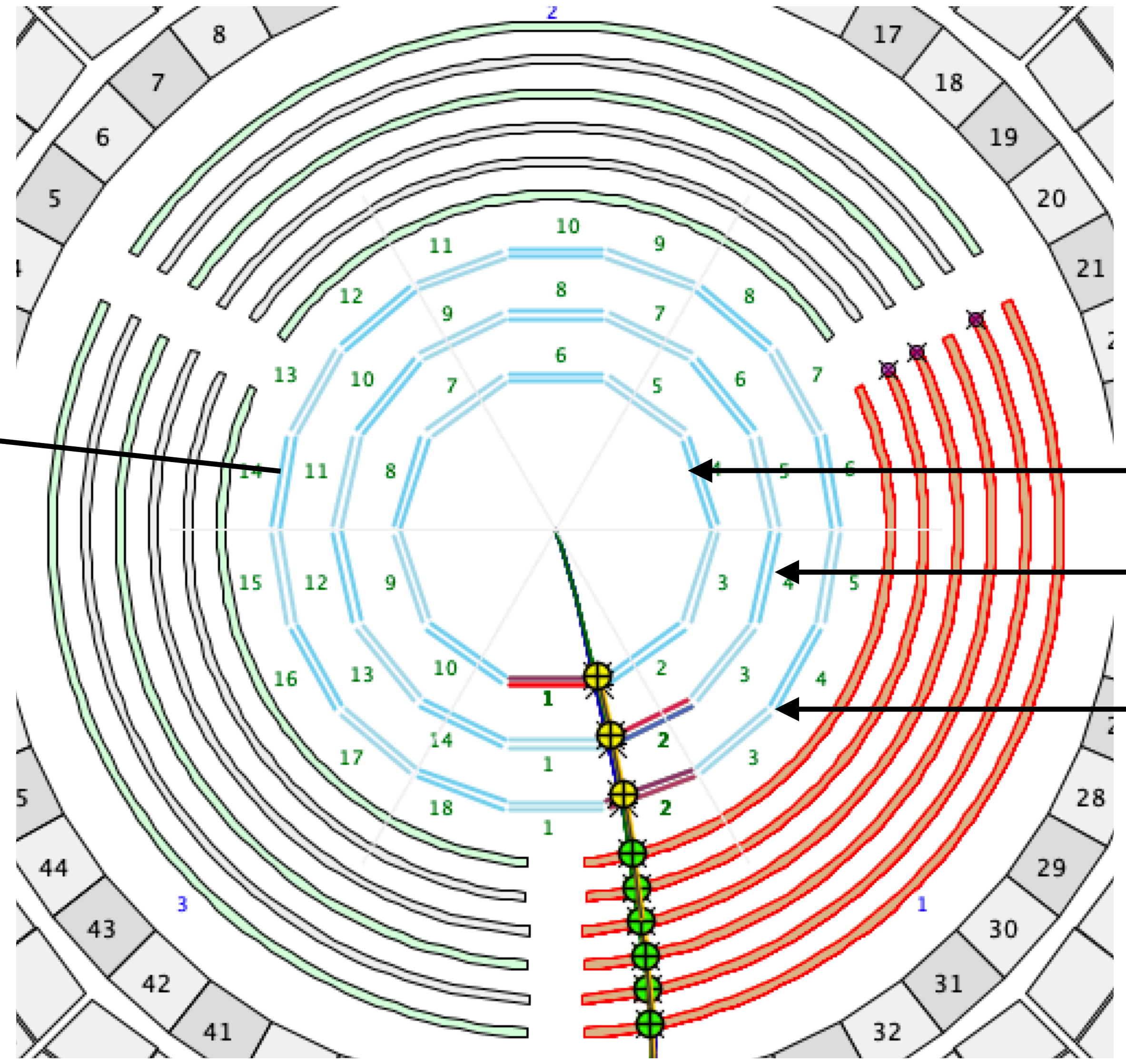
Central XY (Seq Event# 4 True event# 22)
Next



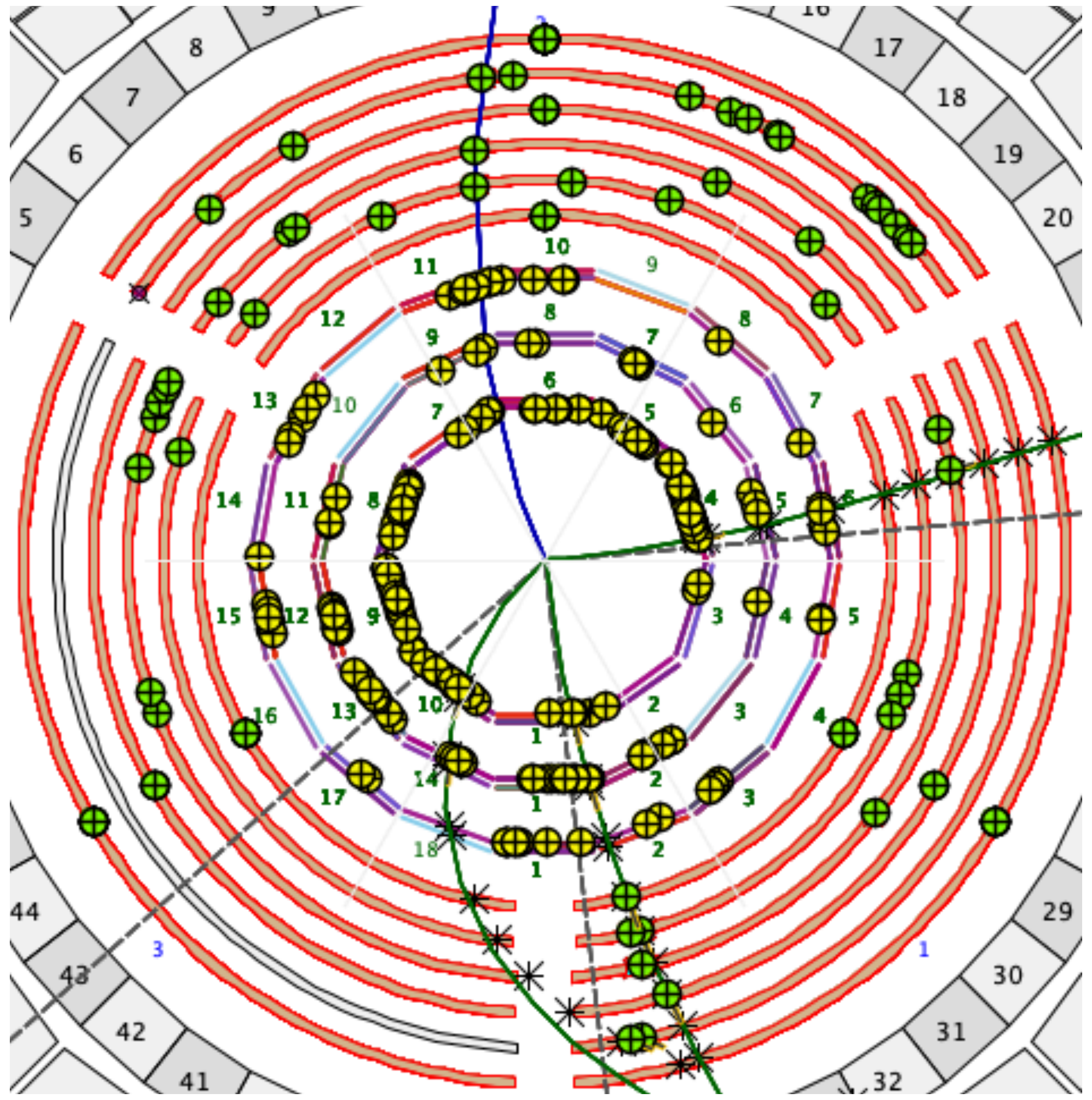
Central Z (Seq Event# 4 True event# 22)
Next

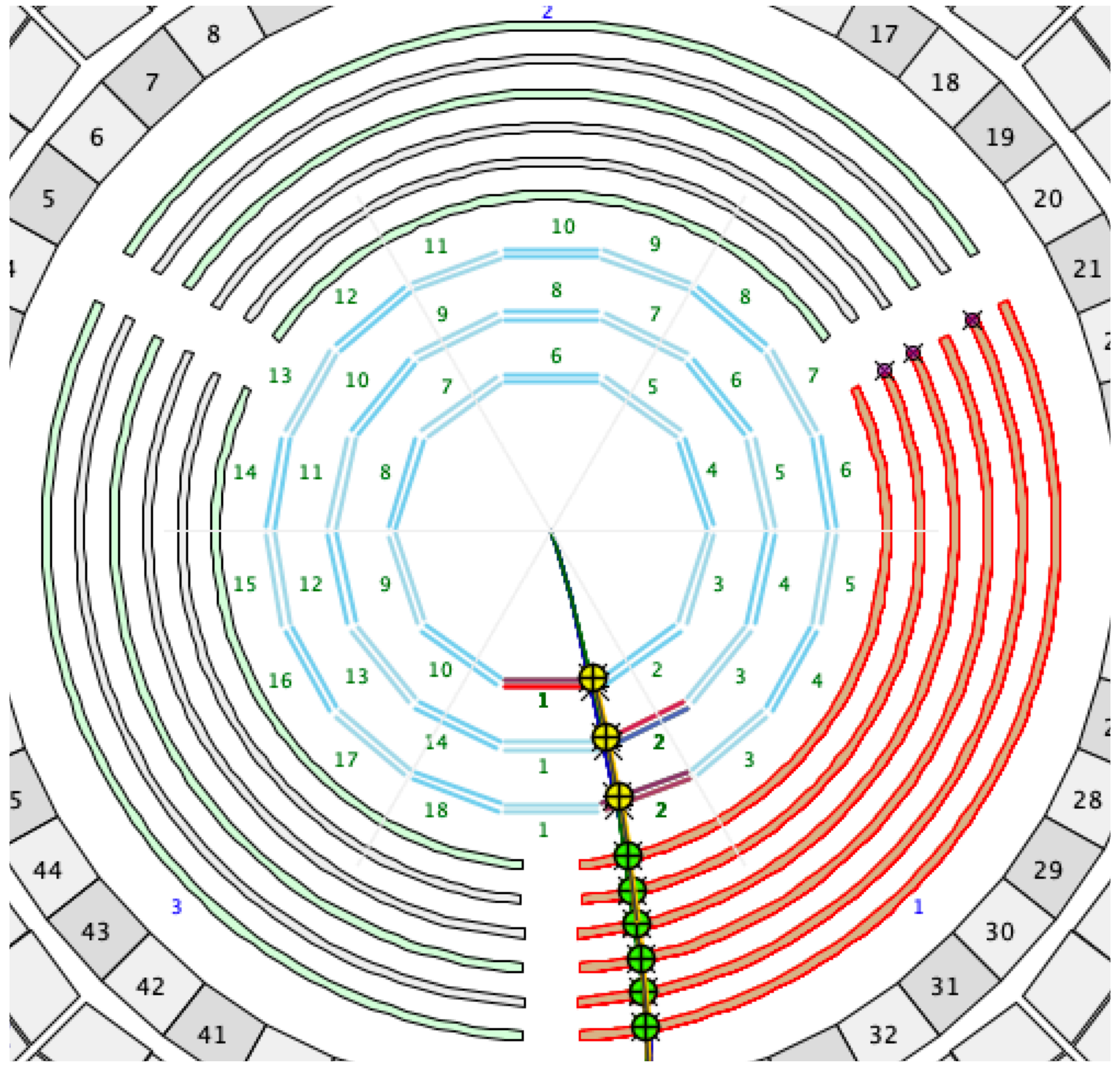


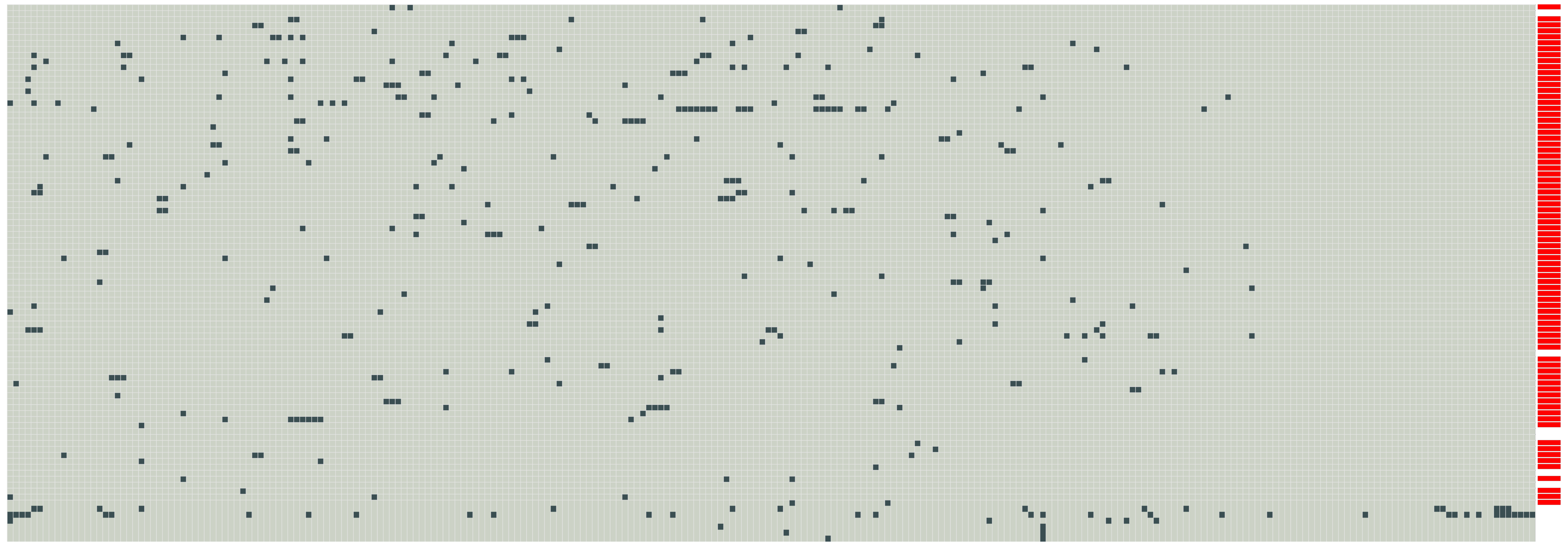


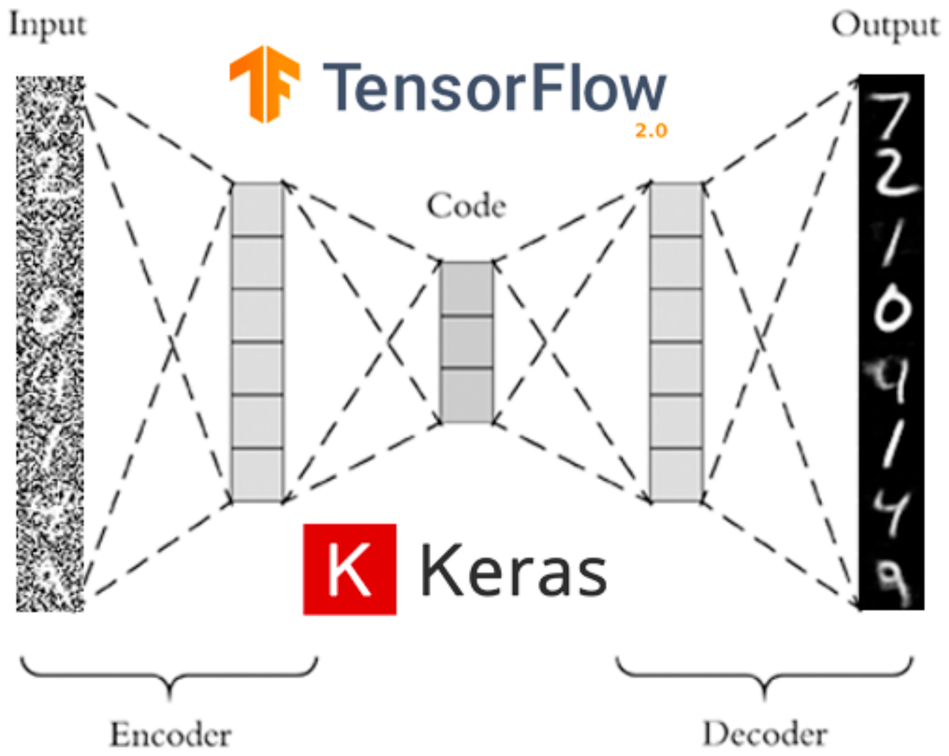


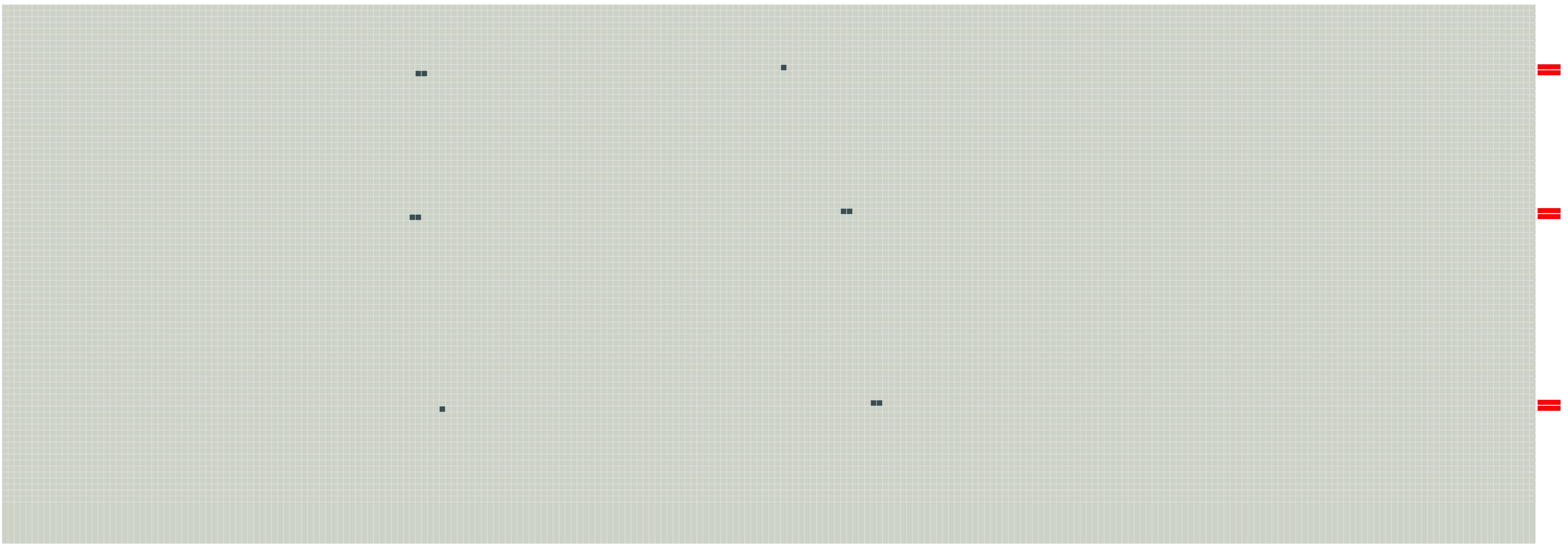
3 rings
1 - 10 double sensors (20)
2 - 14 double sensors (28)
3 - 18 double sensors (36)











This did not work. Why?

Drift Chambers image 36x112 - 4032 pixels

Track trajectory 36 pixels

~1% for single track
~2% for two tracks

Central Detector image 256x90 - 23040 pixels

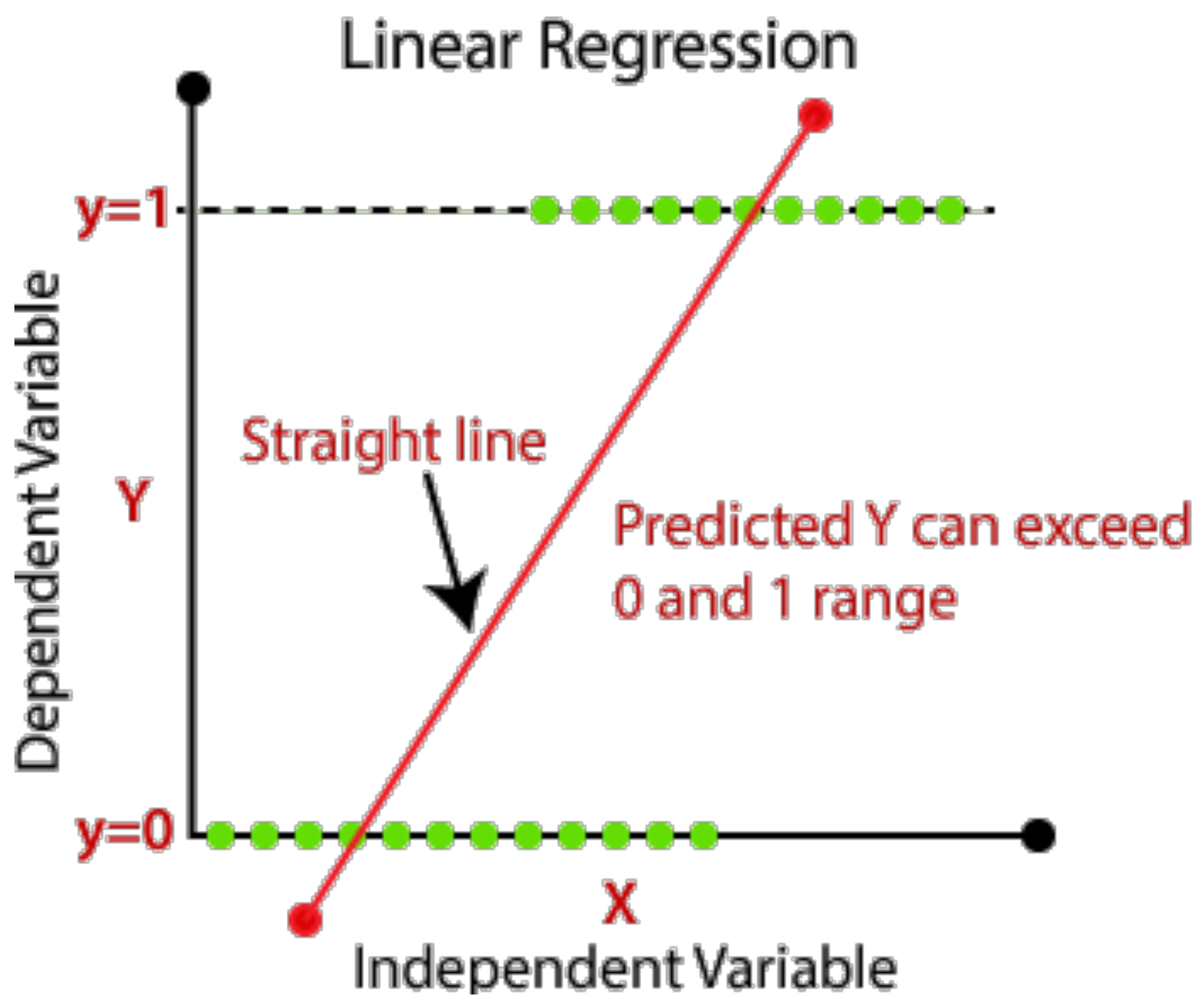
Track trajectory 6 pixels

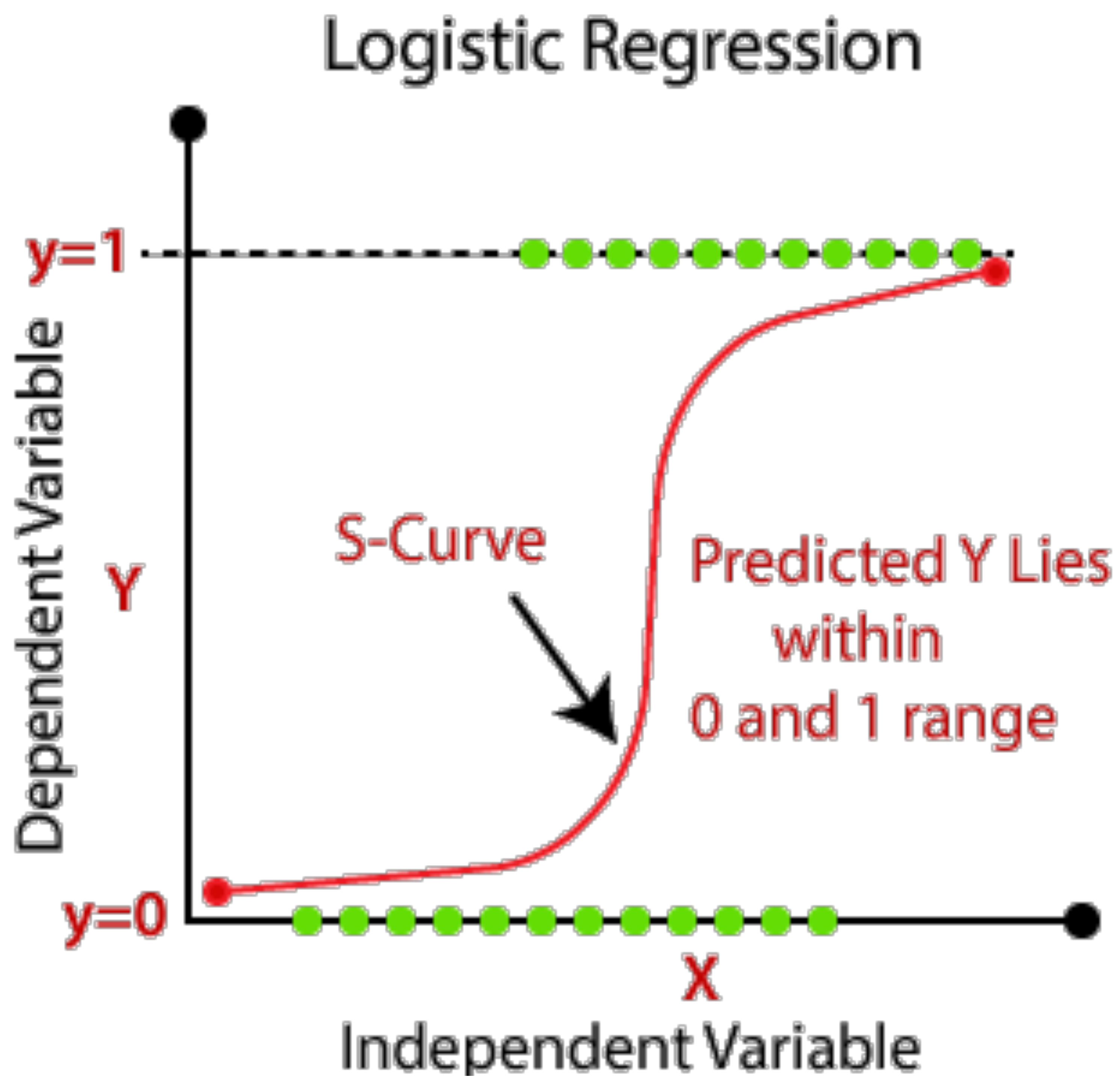
~0.02% for single track
~0.04% for two tracks

Neural networks can not efficiently converge and result in output where all pixels are set to 0

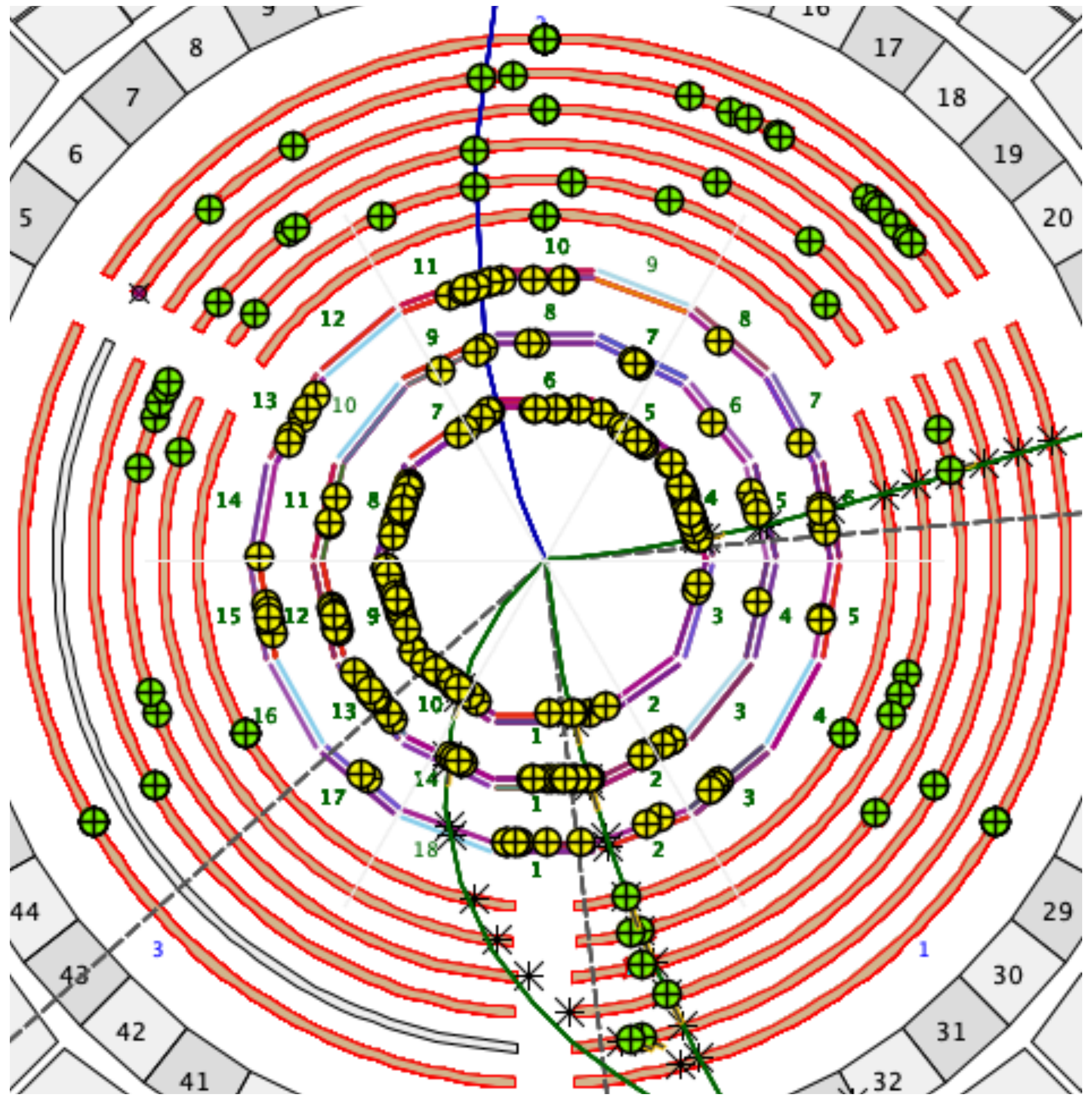
Any other ideas?

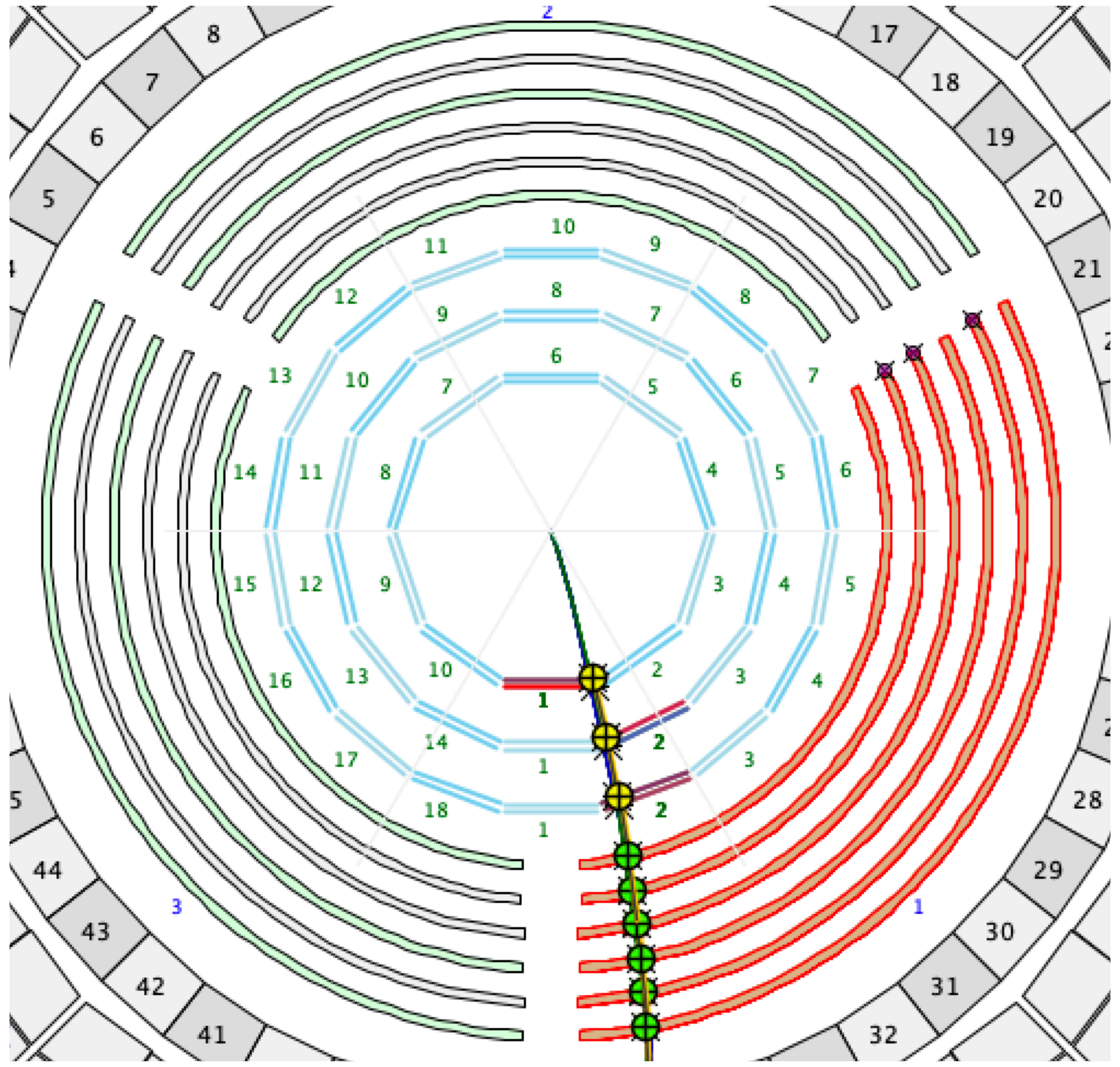
Logistic regression is **a process of modeling the probability of a discrete outcome given an input variable**. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.



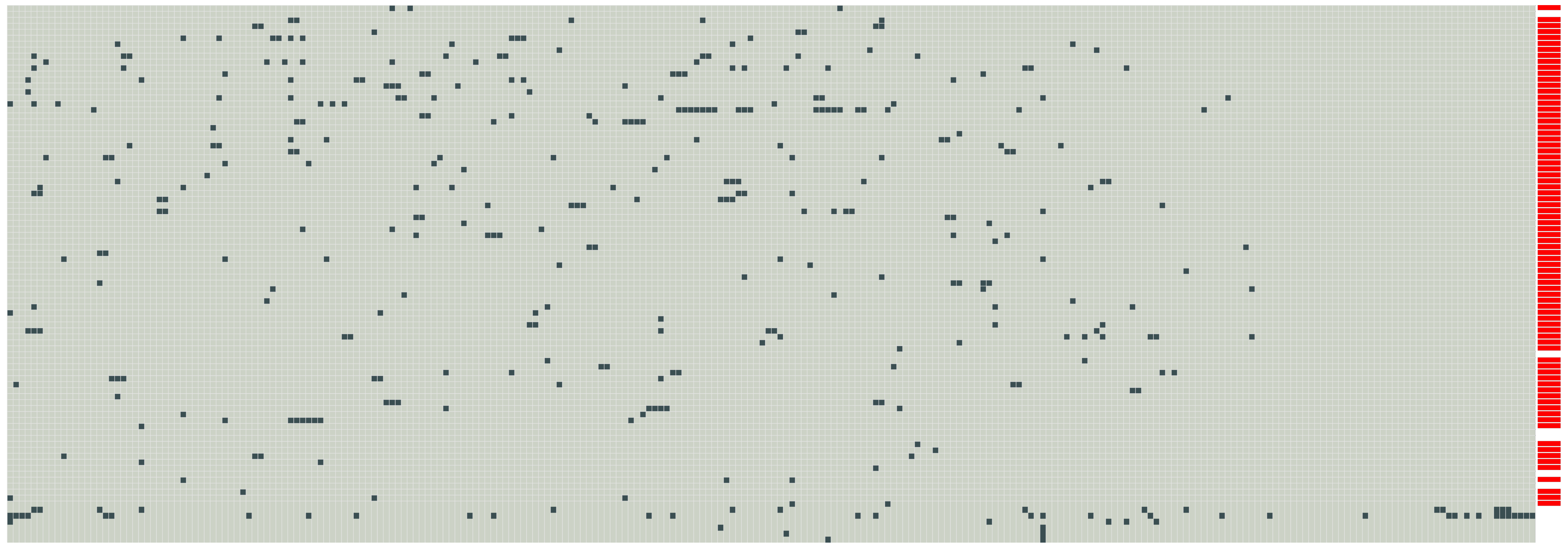


Convolutional Neural Networks

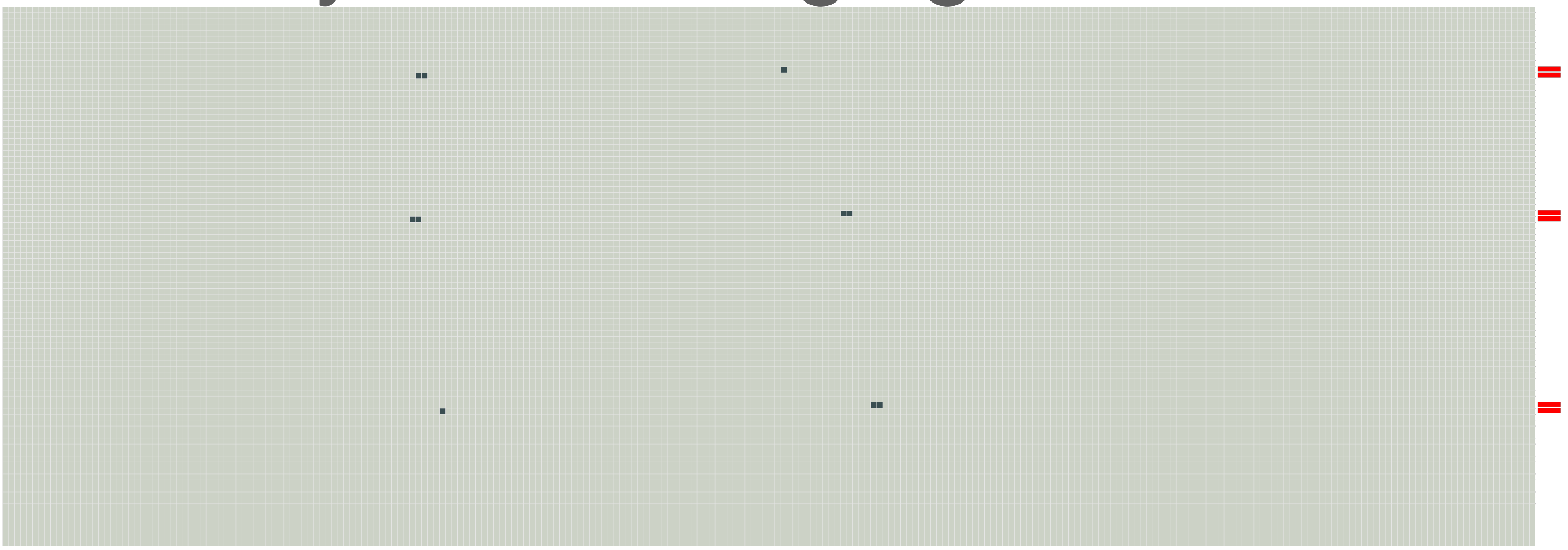




All Hits

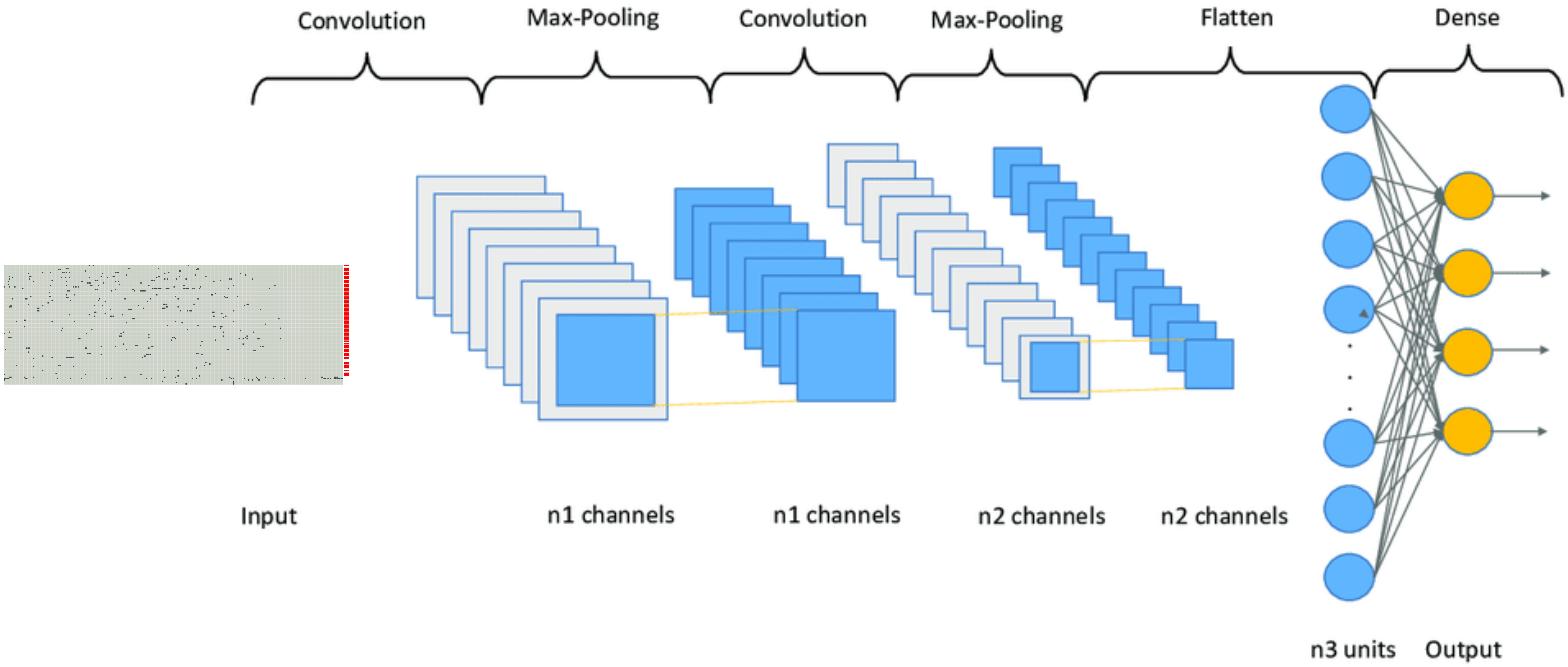


Only hits belonging to a track



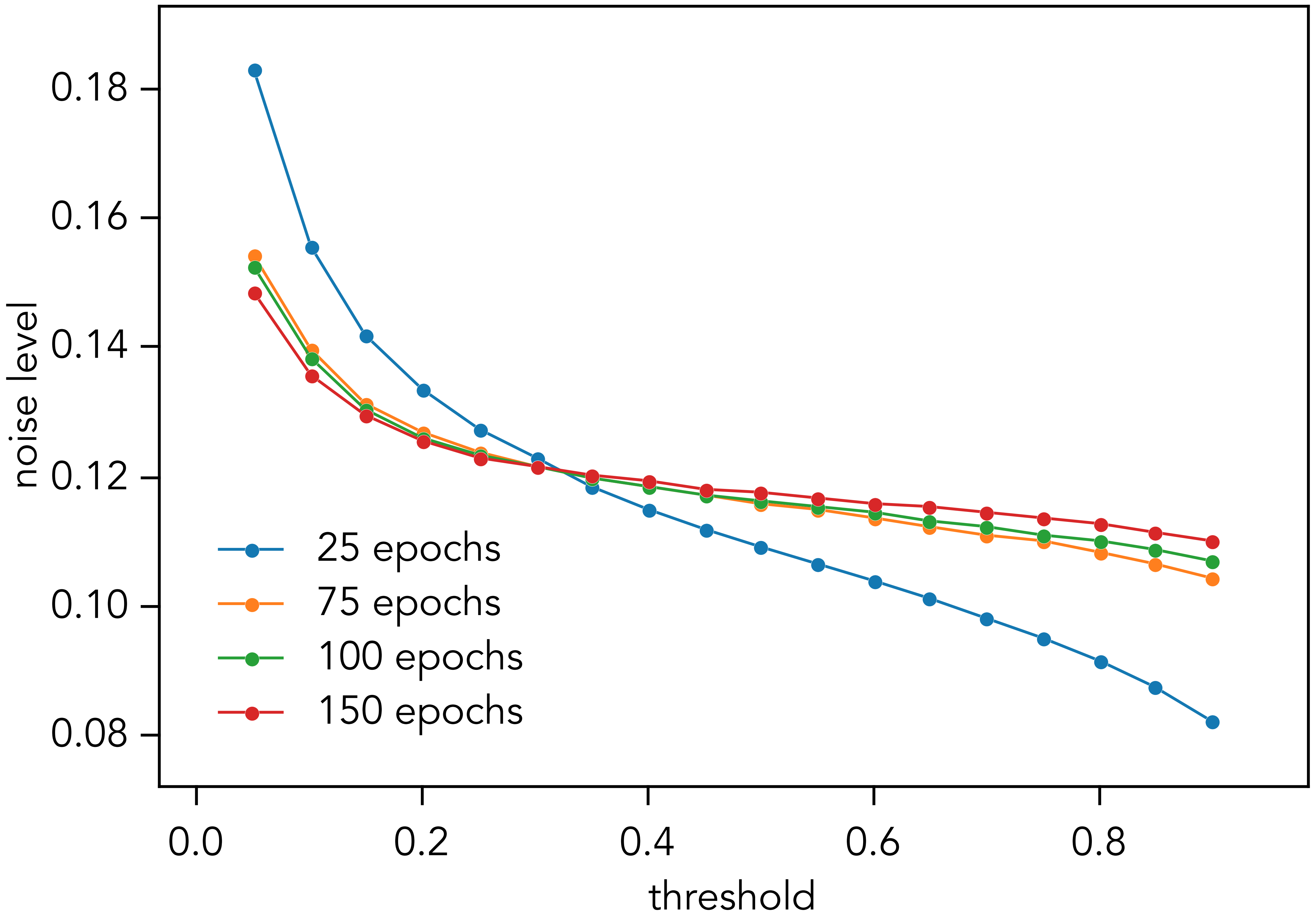
Output is 84 numbers , 0 or 1

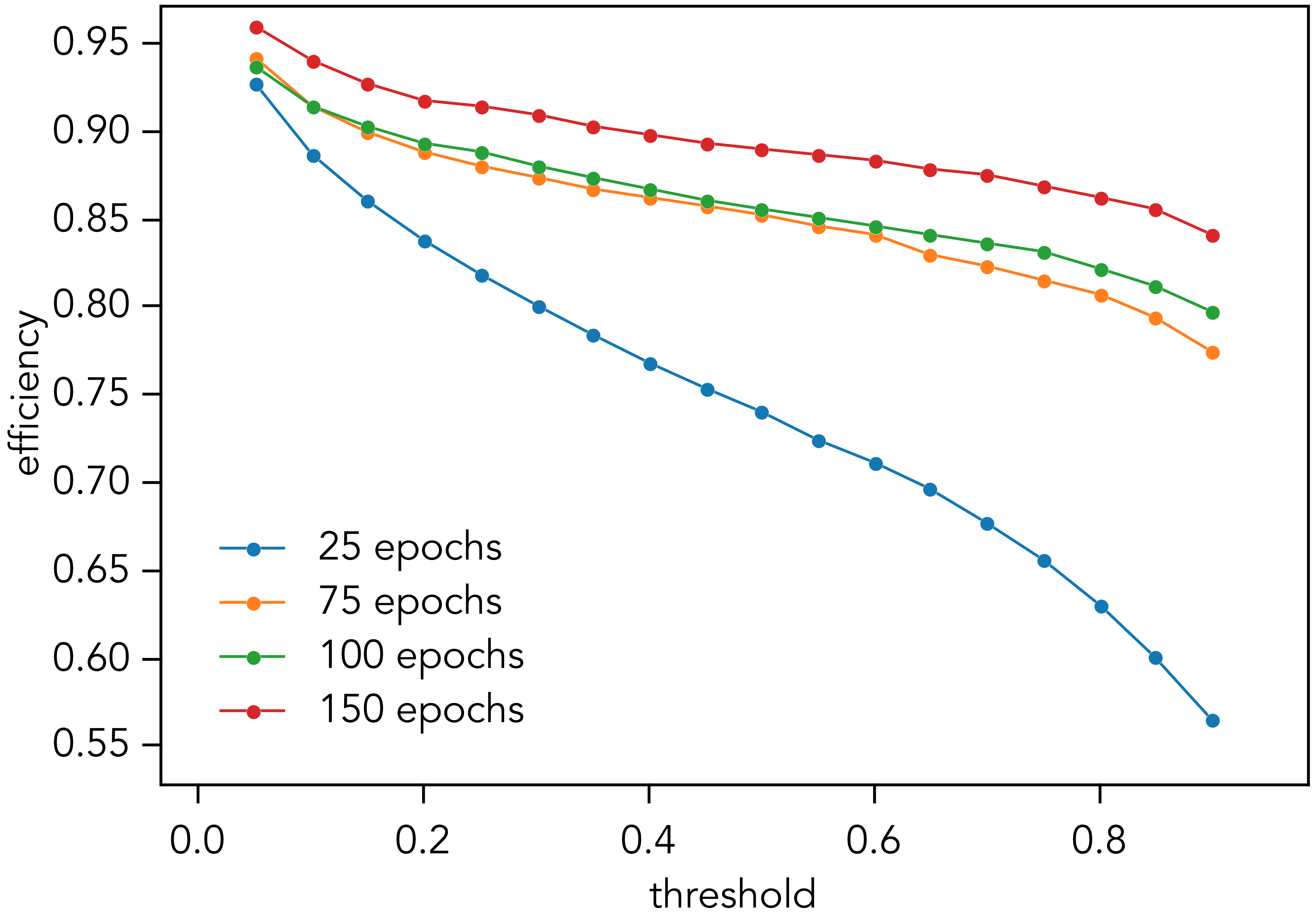
84 Output Nodes (0 or 1)



84 Output Nodes

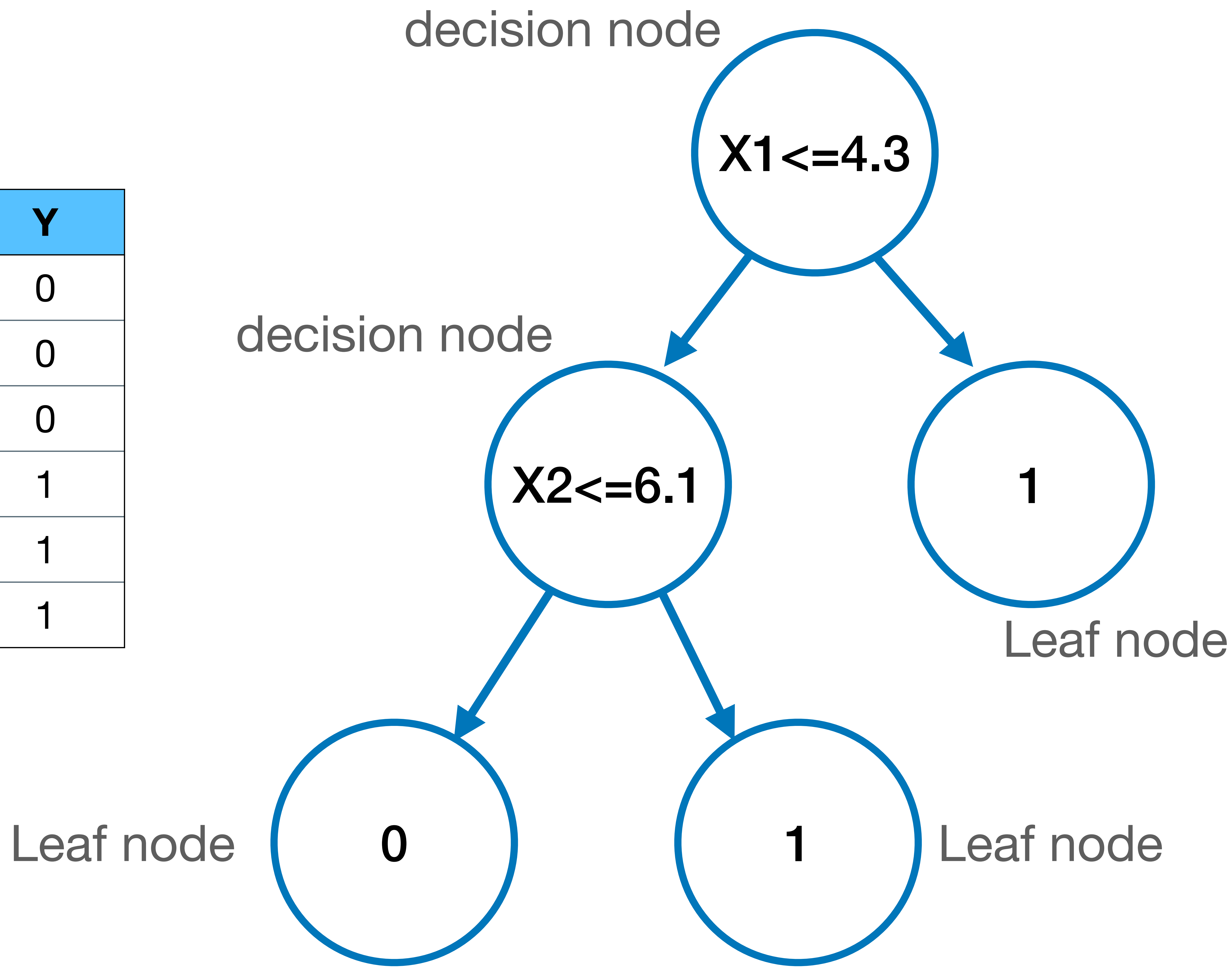
- Noise Level
  - Gradually increases with the lower threshold
  - Below 20% with efficiency above >90%
- Sensor Identification Efficiency
  - 94% at the lowest threshold



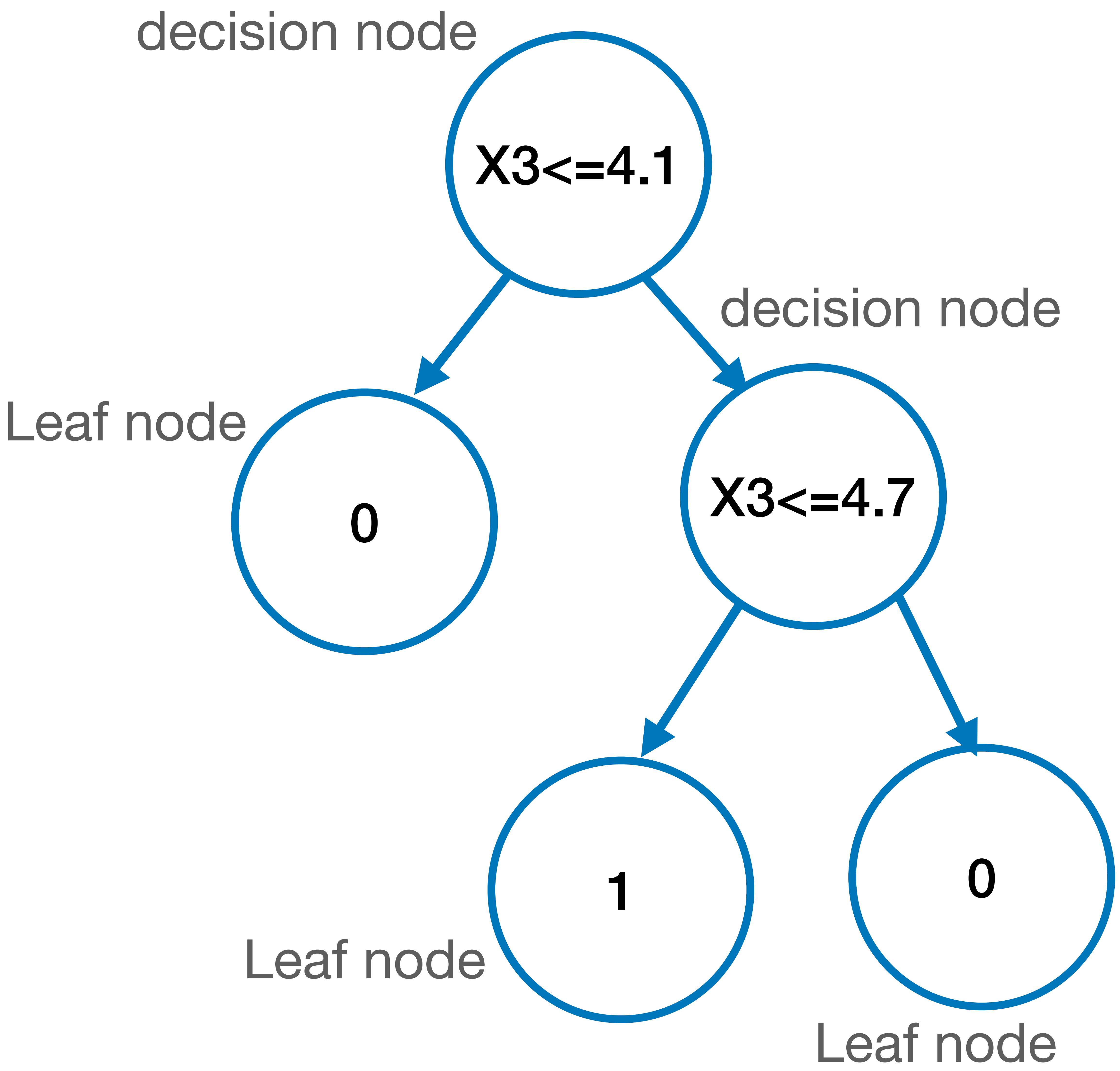


Physics Analysis (Trees)

Id	X1	X2	X3	X4	Y
0	4.3	4.9	4.1	4.7	0
1	3.9	6.1	5.9	5.5	0
2	2.7	4.8	4.1	5.0	0
3	6.6	4.4	4.5	3.9	1
4	6.5	2.9	4.7	4.6	1
5	2.7	6.7	4.2	5.3	1

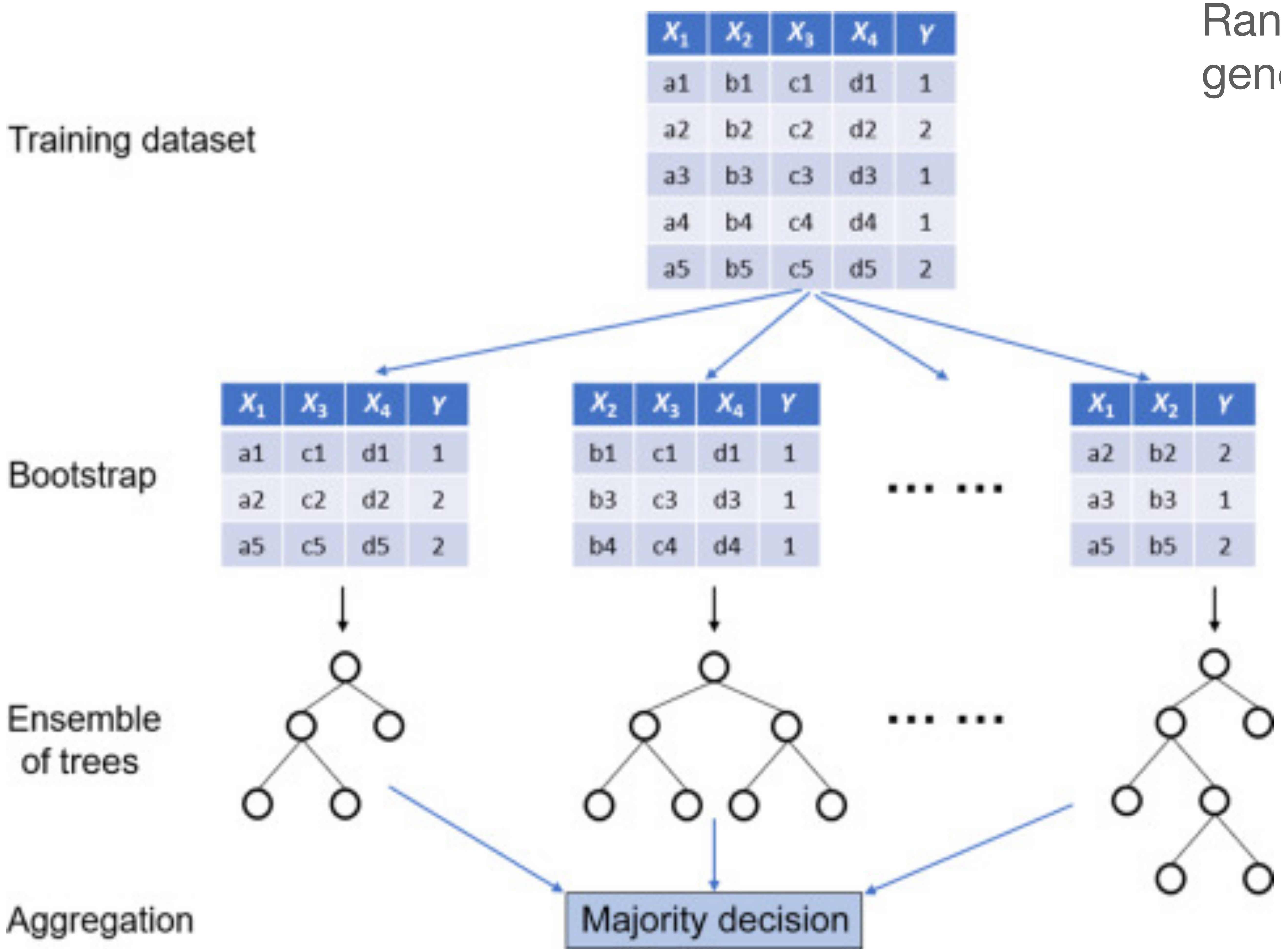


Id	X1	X2	X3	X4	Y
0	4.3	4.9	4.1	4.7	0
1	6.5	4.1	5.9	5.5	0
2	2.7	4.8	4.1	5.0	0
3	6.6	4.4	4.5	3.9	1
4	6.5	2.9	4.7	4.6	1
5	2.7	6.7	4.2	5.3	1



Decision trees are highly sensitive to training data, resulting in high variance
The model may fail to generalize

Random Forest

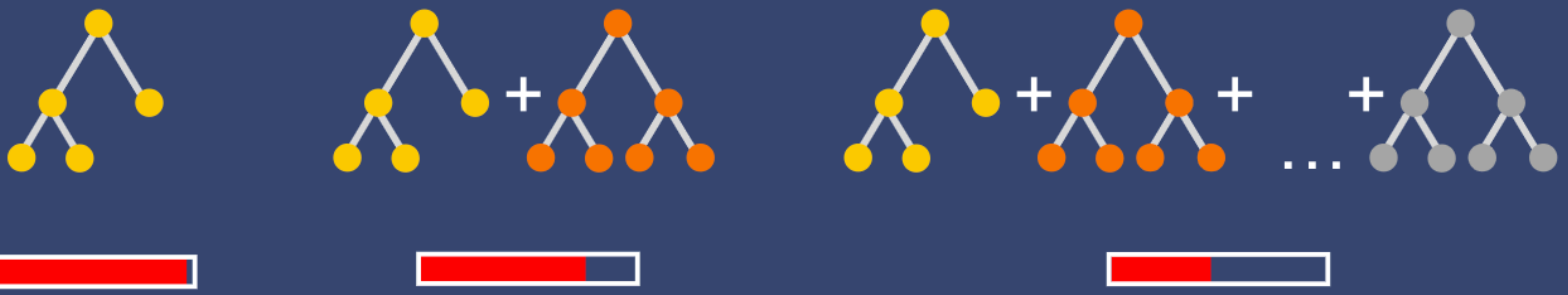


Random Forest is less sensitive to training data and generalizes the model much better

- Bootstrapping
  - Select random data set (duplicates allowed)
  - Select randomly features to use for each tree
  - Train each tree separately
- Aggregation
  - Evaluate input data using all trees
  - A majority voting decides the outcome

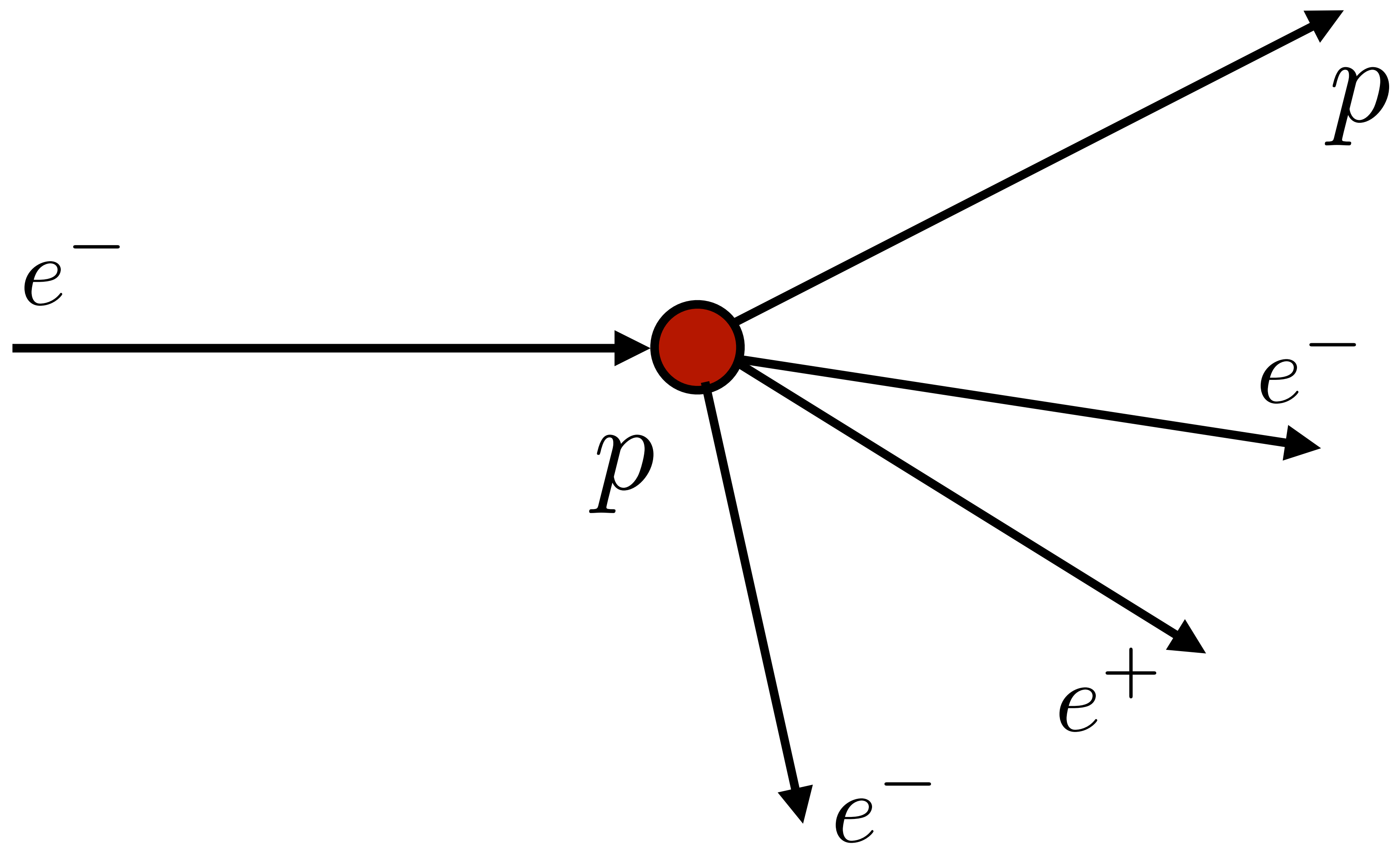
Gradient Boosted Decision Trees

$\hat{y}_i^1 = f1(x_i)$ $\quad$ $\hat{y}_i^2 = \hat{y}_i^1 + f_2(x_i)$ $\quad$ $\hat{y}_i^M = \hat{y}_i^{M-1} + f_M(x_i)$

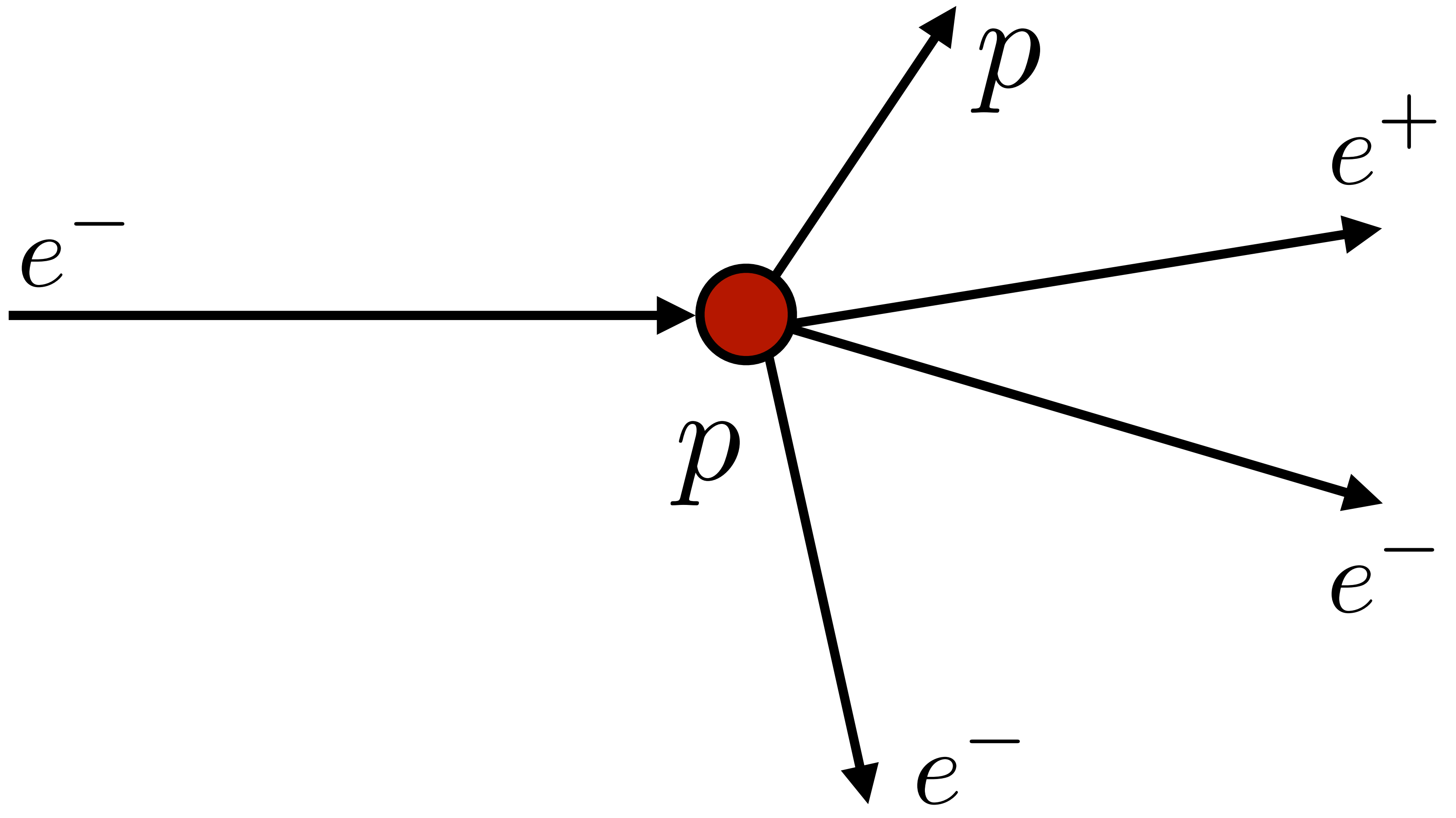


$f1(x_i) \rightarrow y_i$ $\quad$ $f_2(x_i) \rightarrow y_i - \hat{y}_i^1$ $\quad$ $f_M(x_i) \rightarrow y_i - \hat{y}_i^{M-1}$

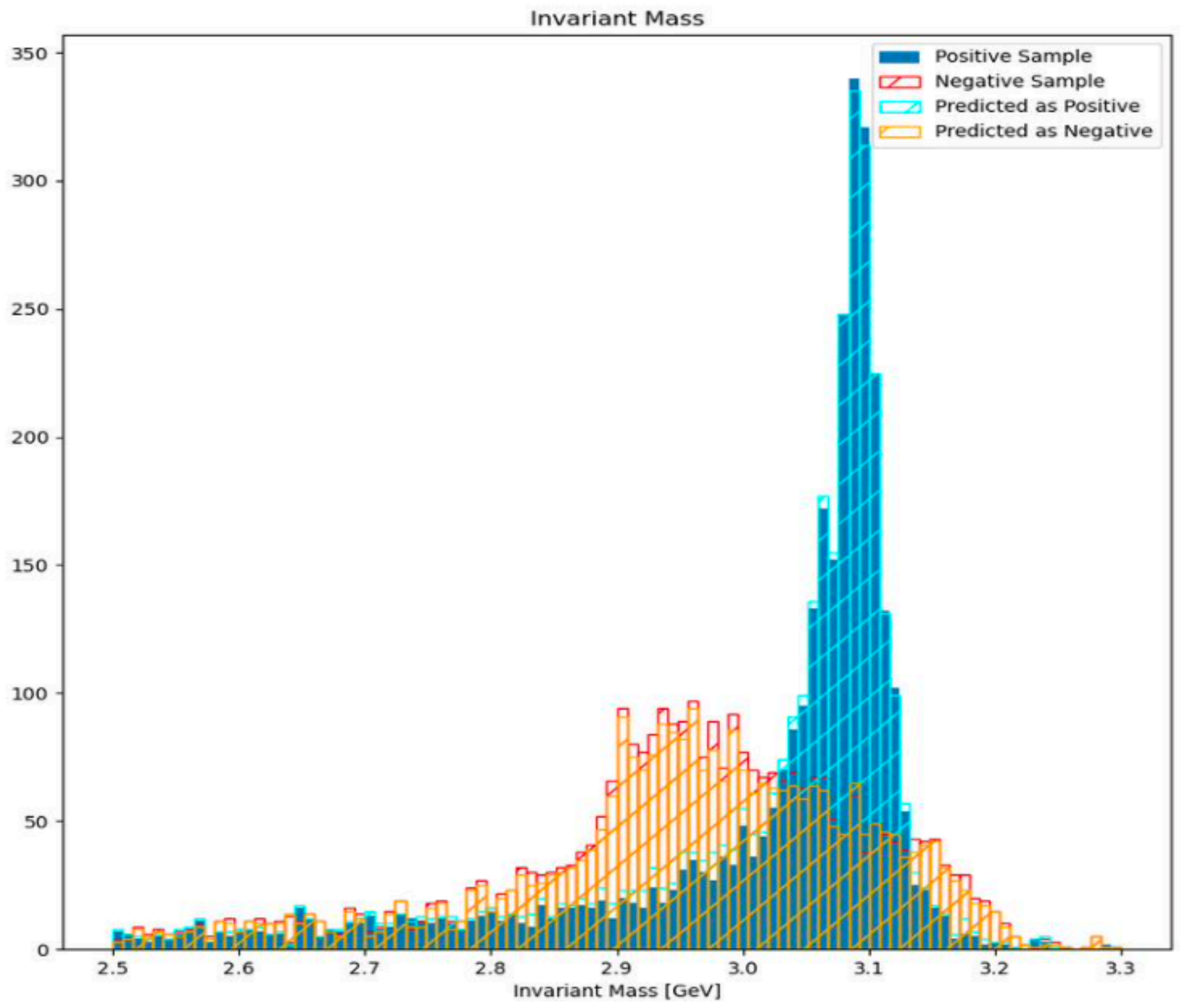
Physics Reaction Identification

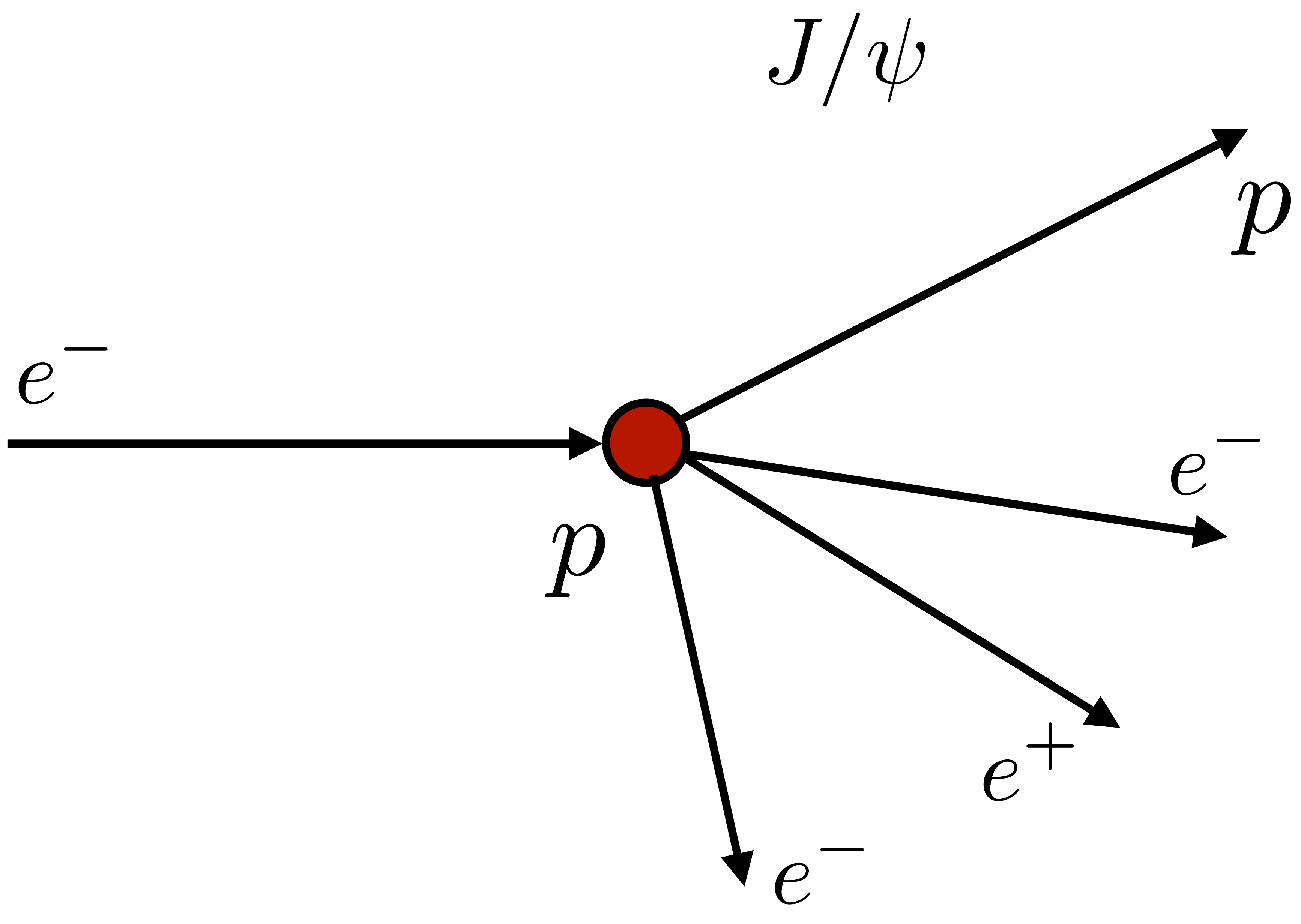


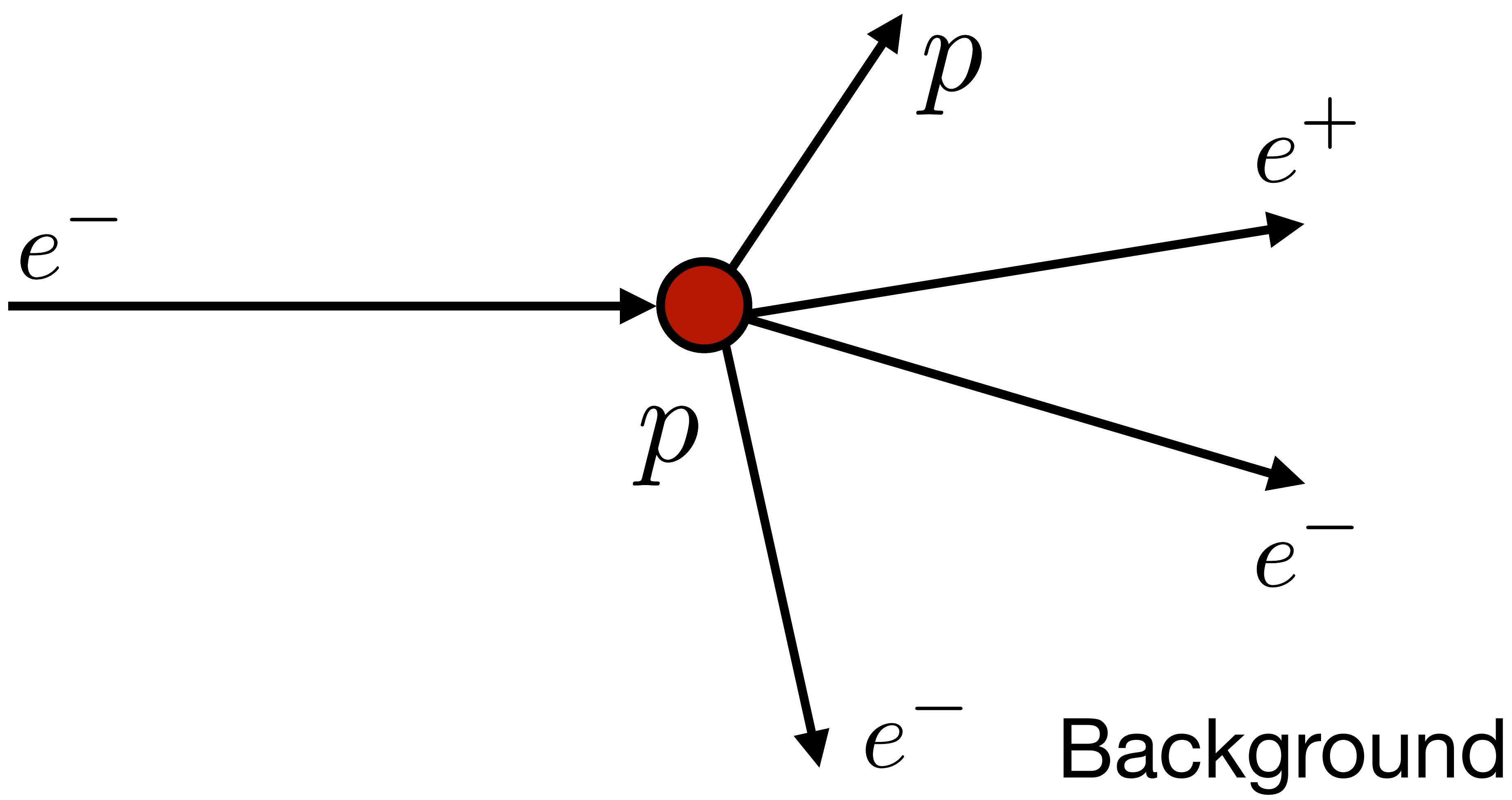
$J/\psi(e^+e^-)$



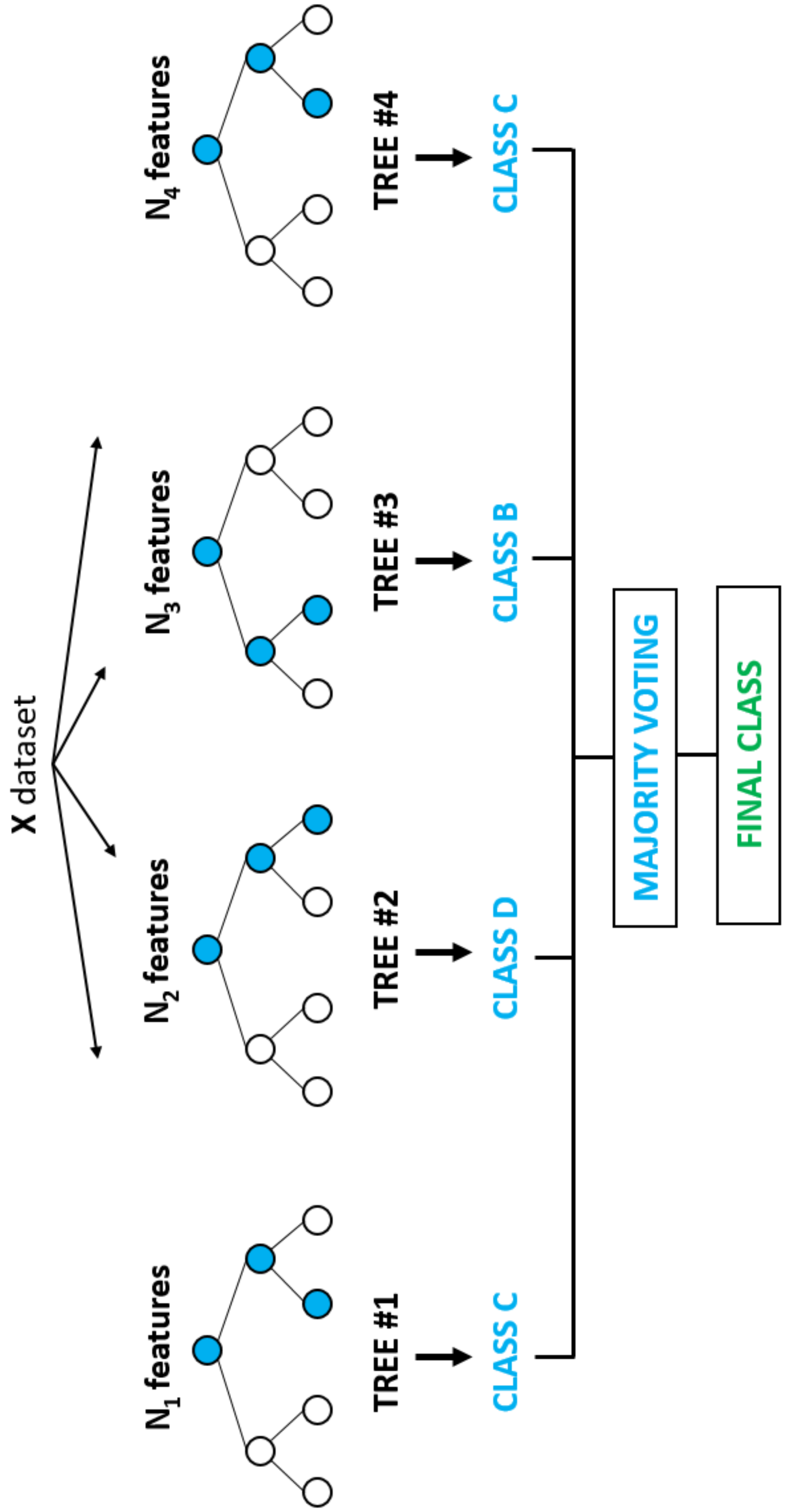
Background







p_{e^-}
θ_{e^-}
ϕ_{e^-}
p_p
θ_p
ϕ_p
p_{e^-}
θ_{e^-}
ϕ_{e^-}
p_{e^+}
θ_{e^+}
ϕ_{e^+}



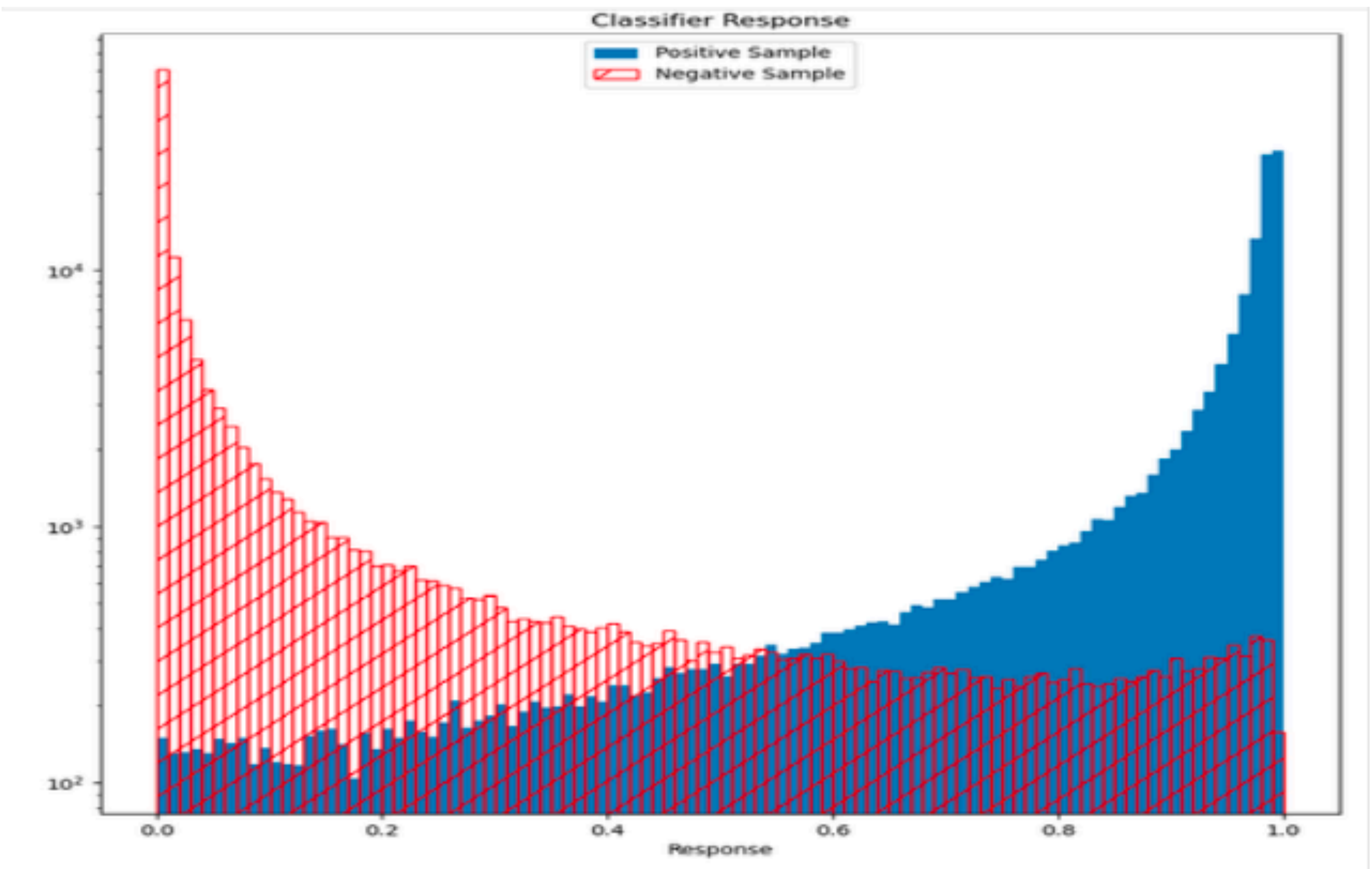
Signal

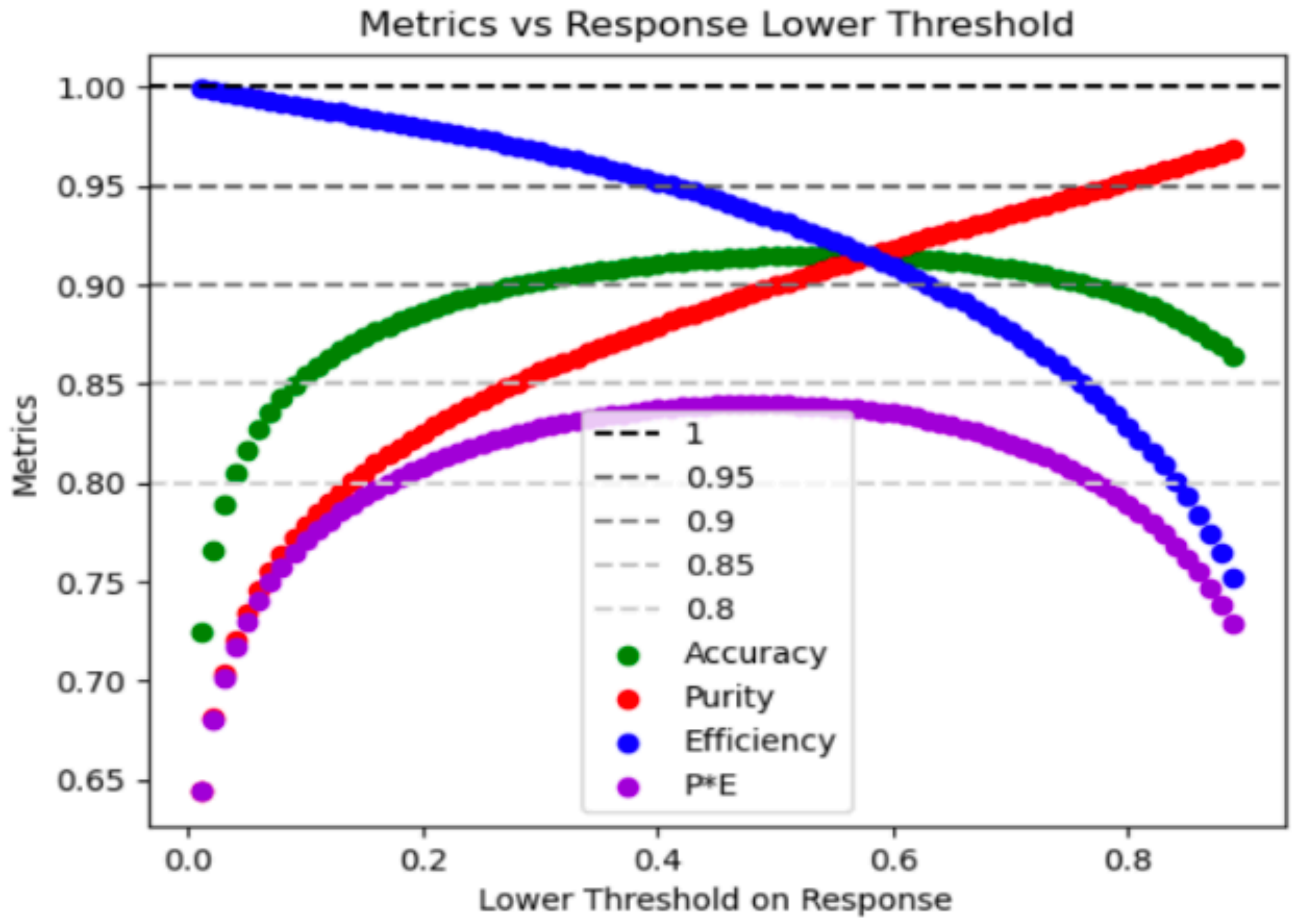
Background

Physics Reaction Identification

- The response is the classifier's output.
- A tight cut on the response leads to a cleaner event
- Metrics are calculated here for the testing sample with equal amounts of positive and negative sample events.
- The prediction rate for all is of order 10^5 s^-1

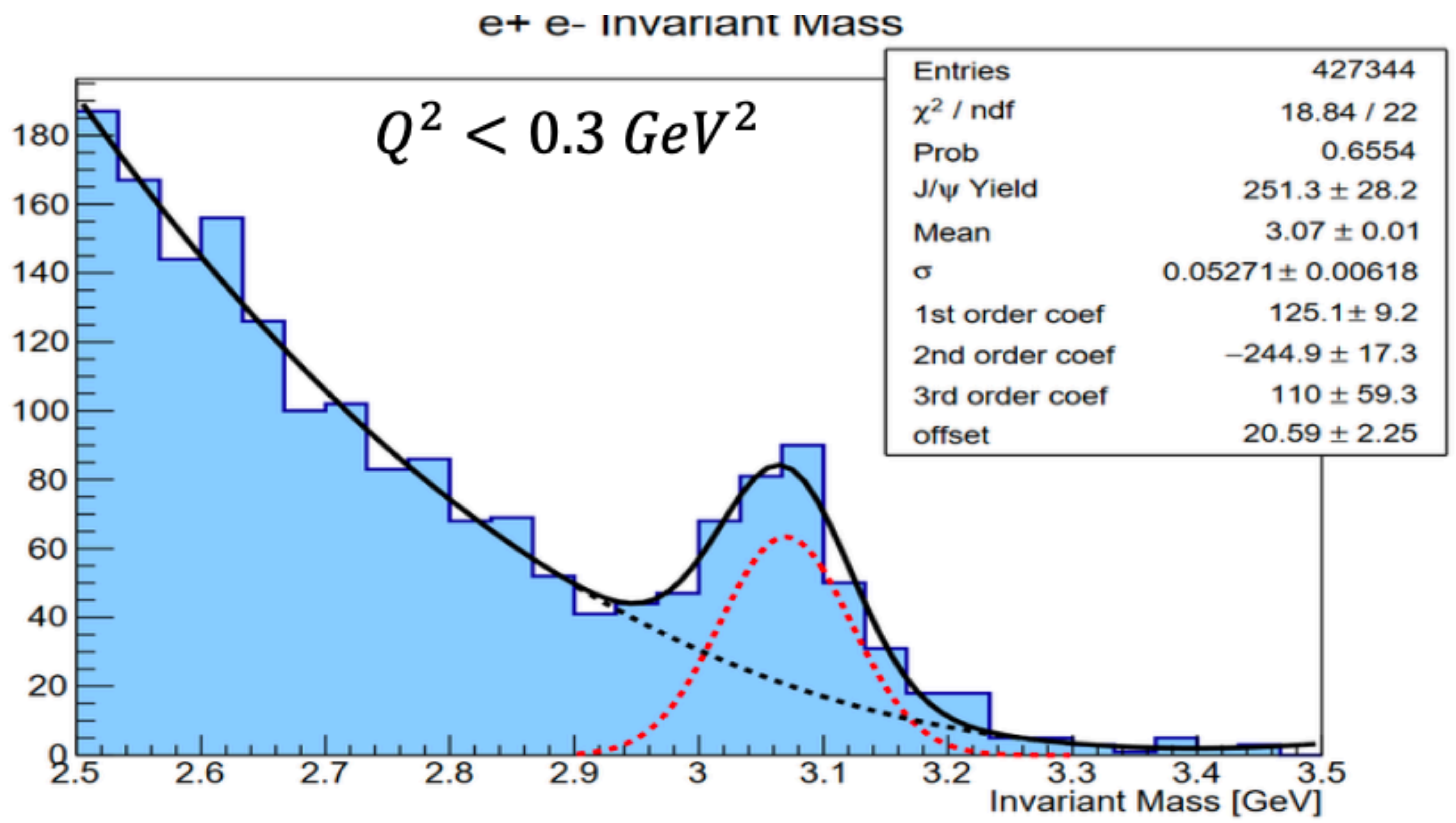
Training Sample	Accuracy	Purity at Efficiency of 0.95
e' e+ e- p	0.95	0.94
e' p	0.95	0.94
e' e+ e-	0.75	0.6
e+ e- p	0.9	0.85

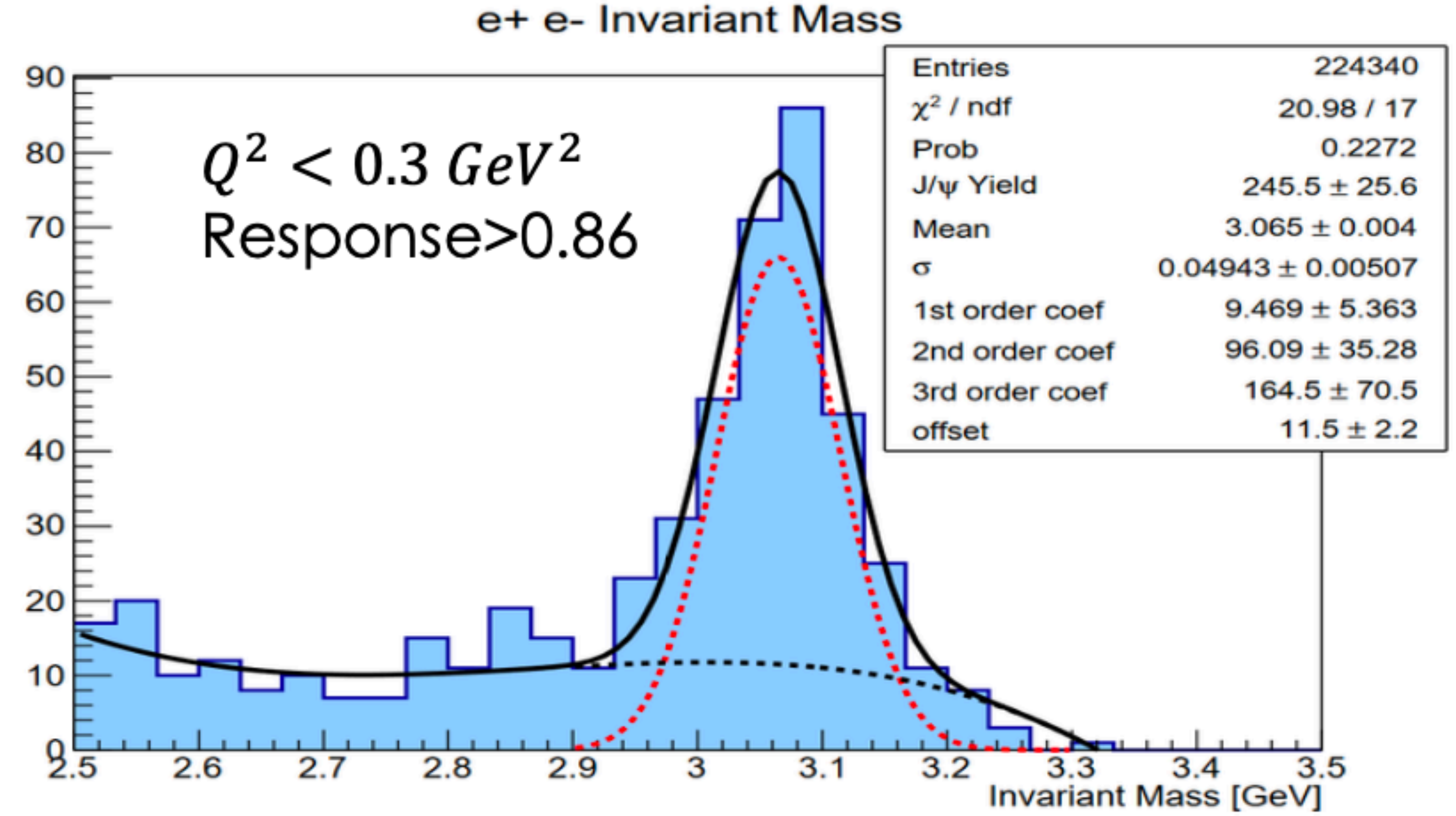




Physics Reaction Identification

- 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.

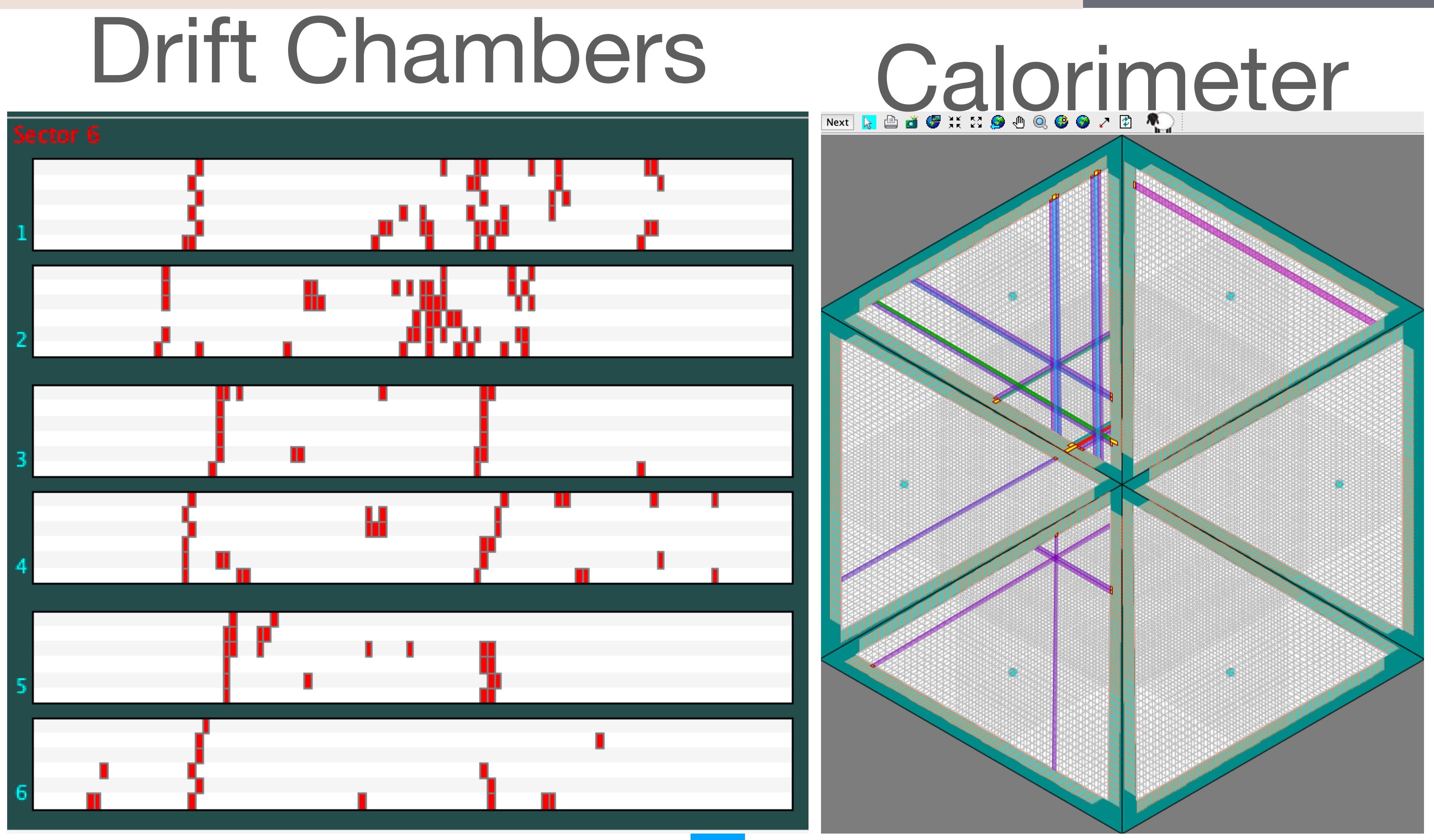


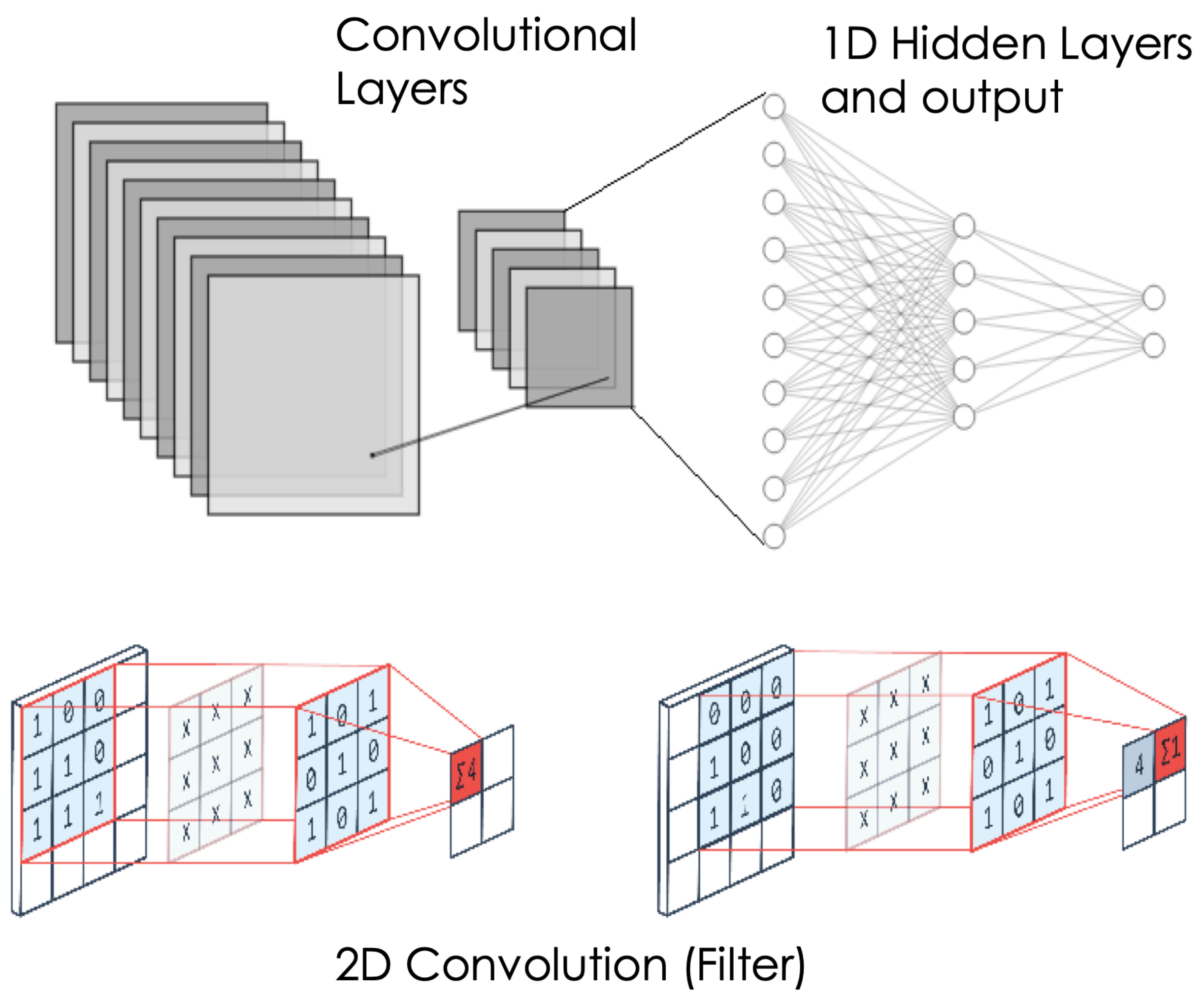


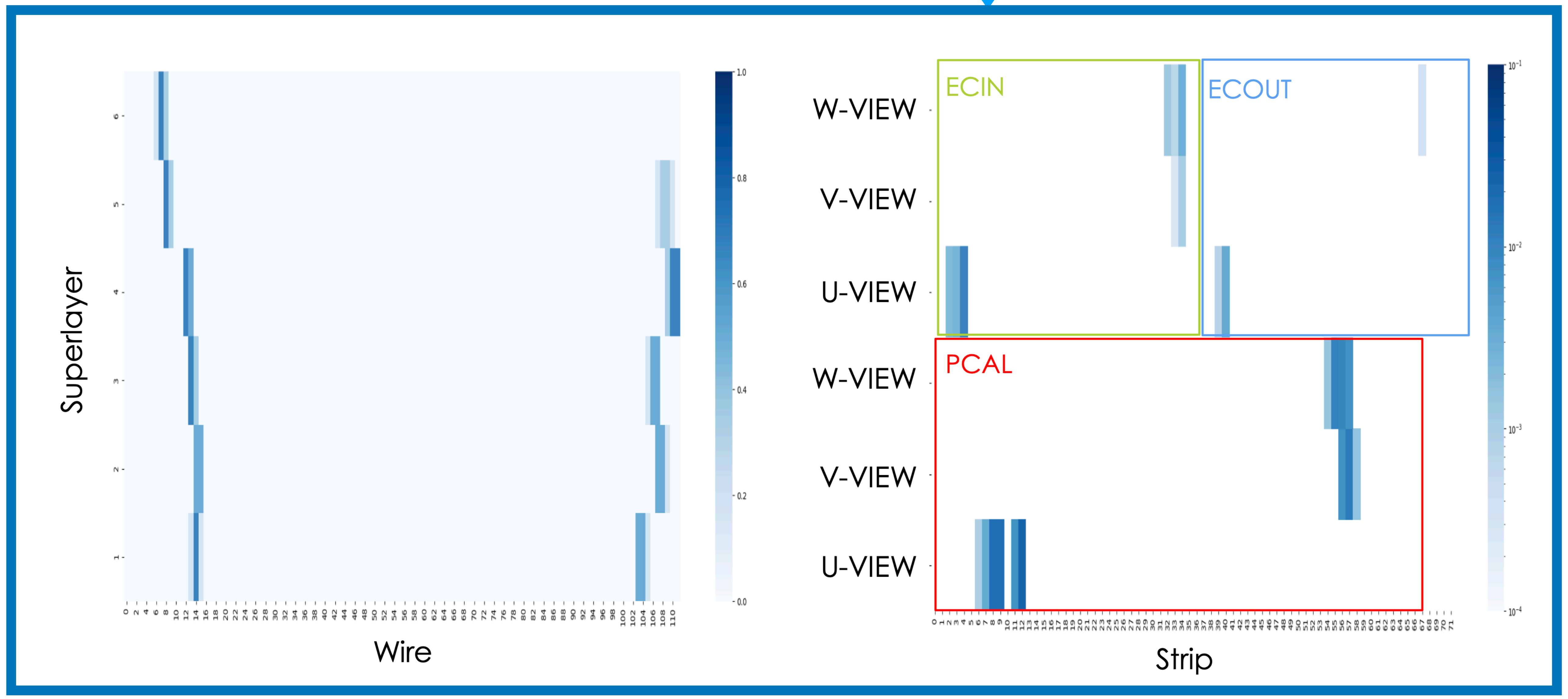
Experiment Control

Combining Two Detectors into one image

- Identify electrons based on raw information from Drift Chambers and calorimeter
- Network trained on reconstructed data:
  - Matrix of size 6x112 for drift chambers
  - Matrix of 6x72 for calorimeter
  - Combined matrix of 6x184
- Convolutional Neural network is used to train on data with reconstruct electron in the sector and random tracks with calorimeter hits in the same sector.

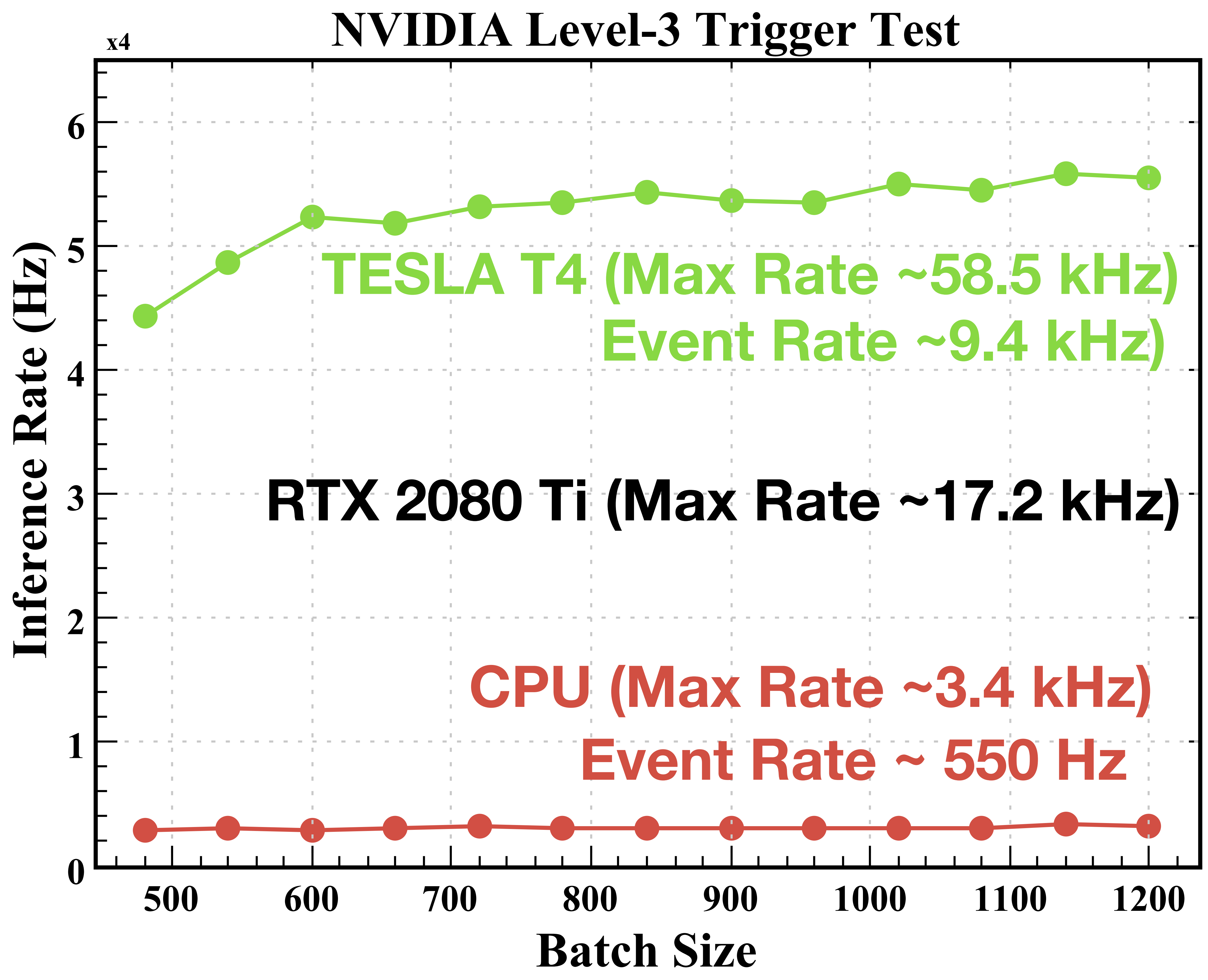


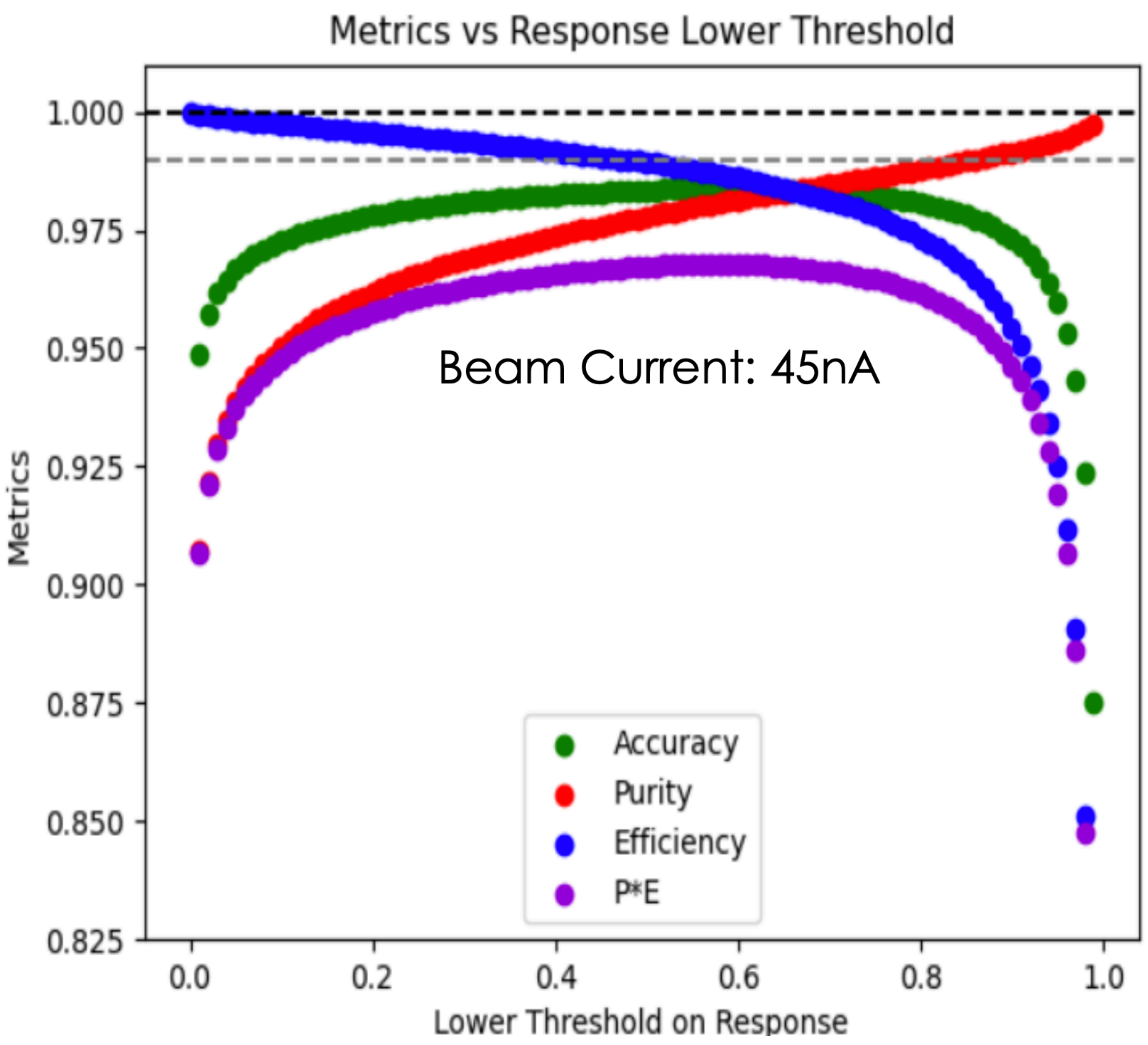




Level-3 trigger (with Artificial Intelligence)

- 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 reported as inference per second (inference is per one sector)
- Real data rate is inference divided by 6
- Results are reported for 1 CPU core and 1 GPU unit

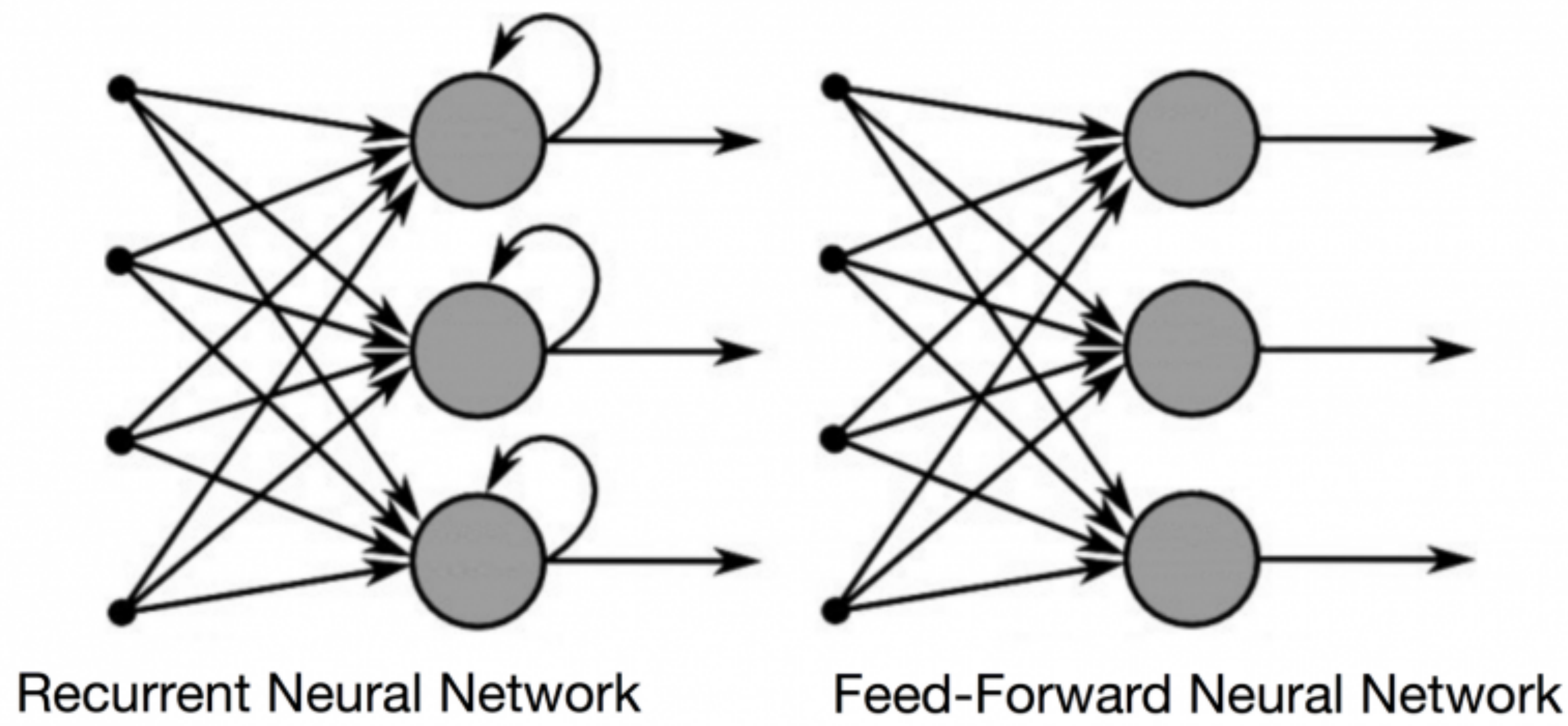


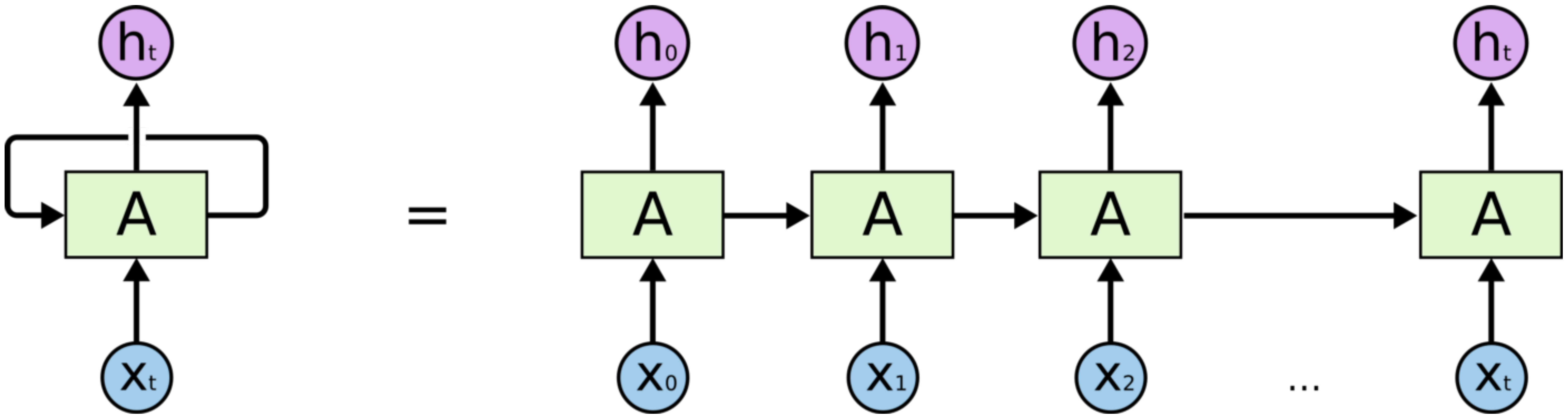


Threshold	Purity	Efficiency	Accuracy
0.0012	0.841	0.9999	0.906
0.03	0.930	0.999	0.962
0.47	0.977	0.99	0.983

Recurrent Neural Networks (RNN)

Recurrent neural networks (RNN) are **the state of the art algorithm for sequential data and are used by Apple's Siri and and Google's voice search**. It is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data.





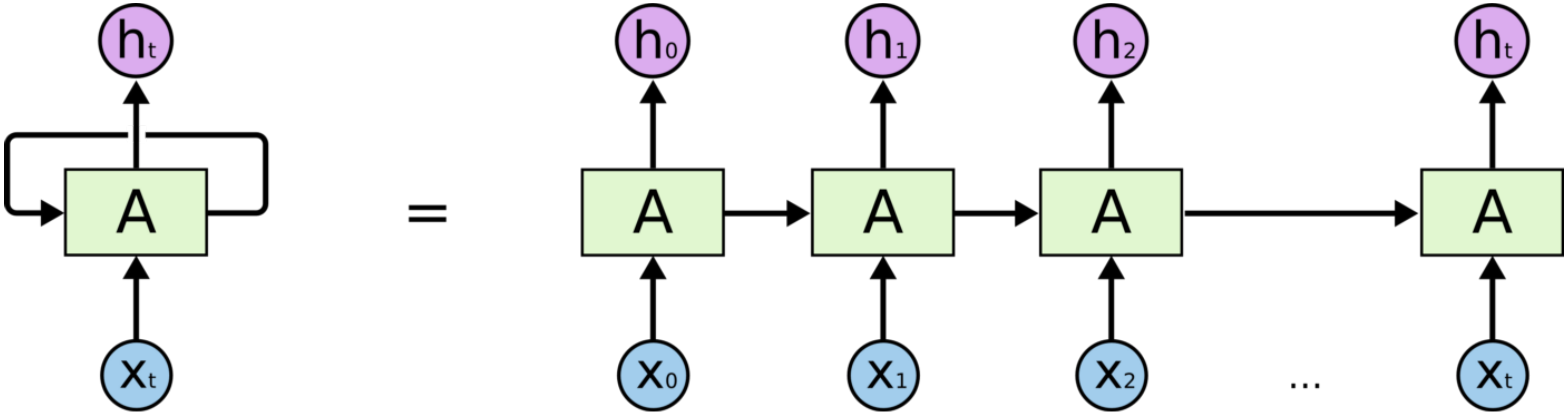
A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs

I just woke up and I’m going to have ______________

I’m tired, I’m going to go _____________

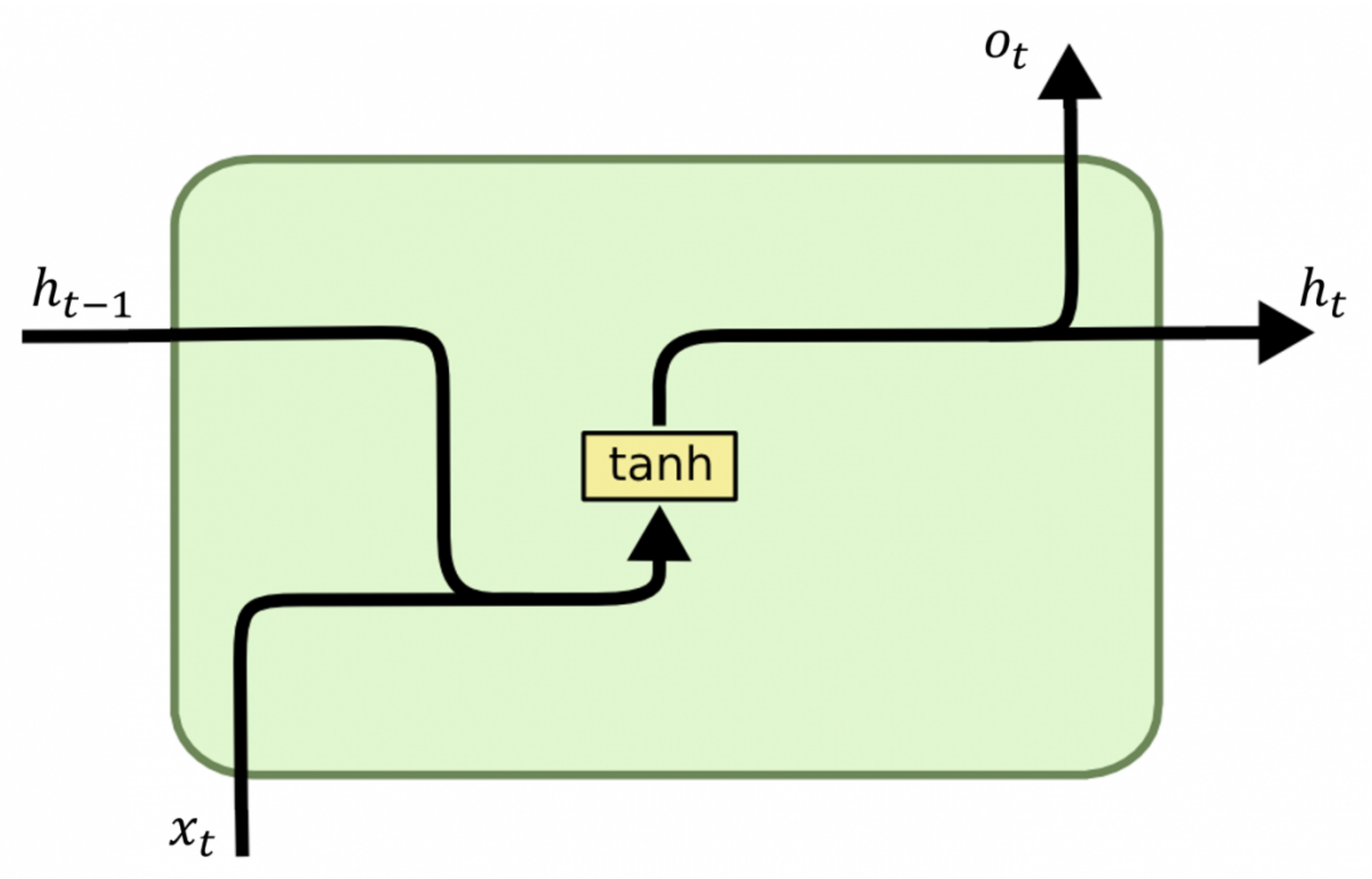
The grass is _____________ (green)

It was raining at night. The grass is ______________

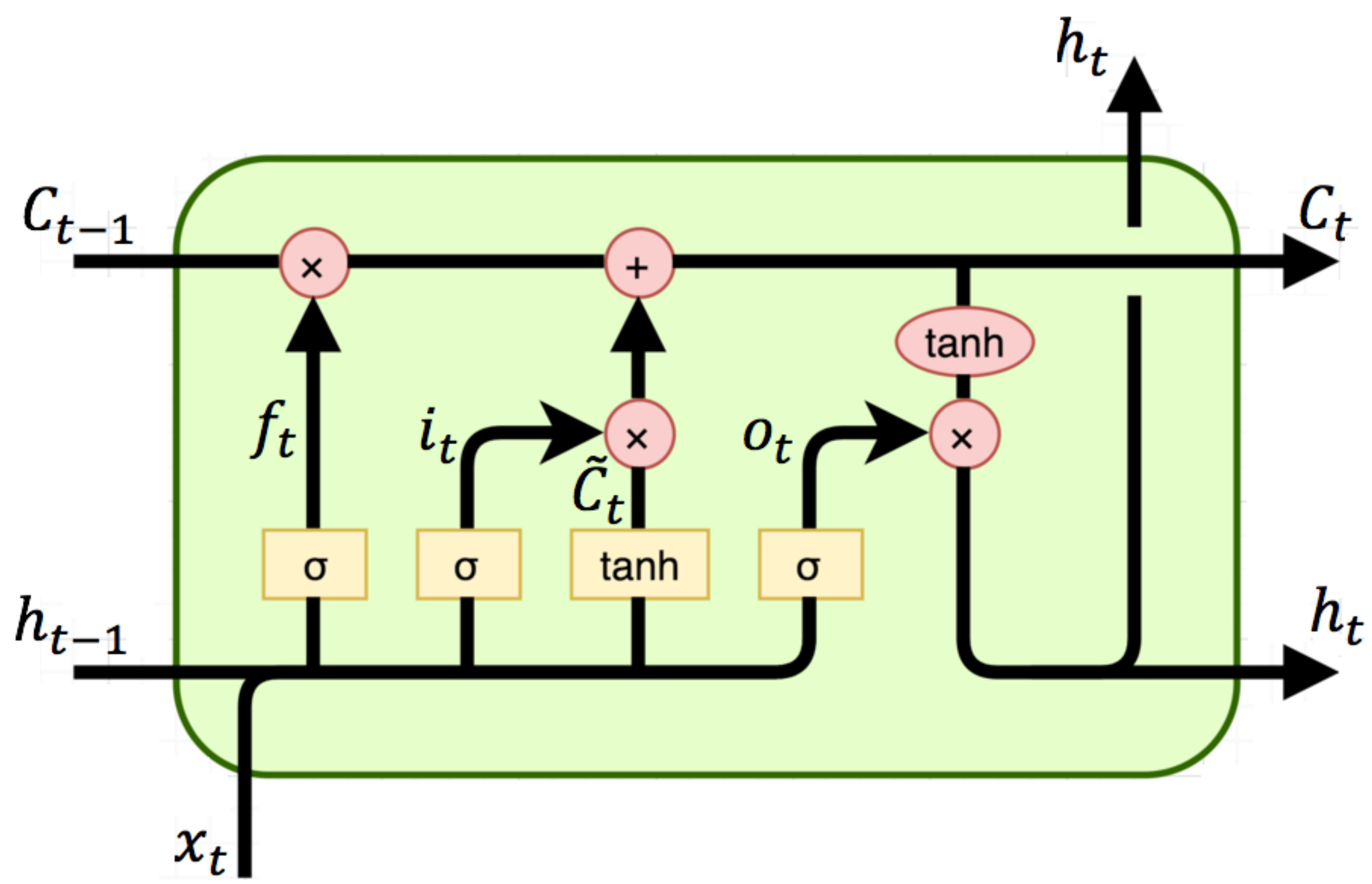


The more steps in series you go, the memory of the earlier stages fades

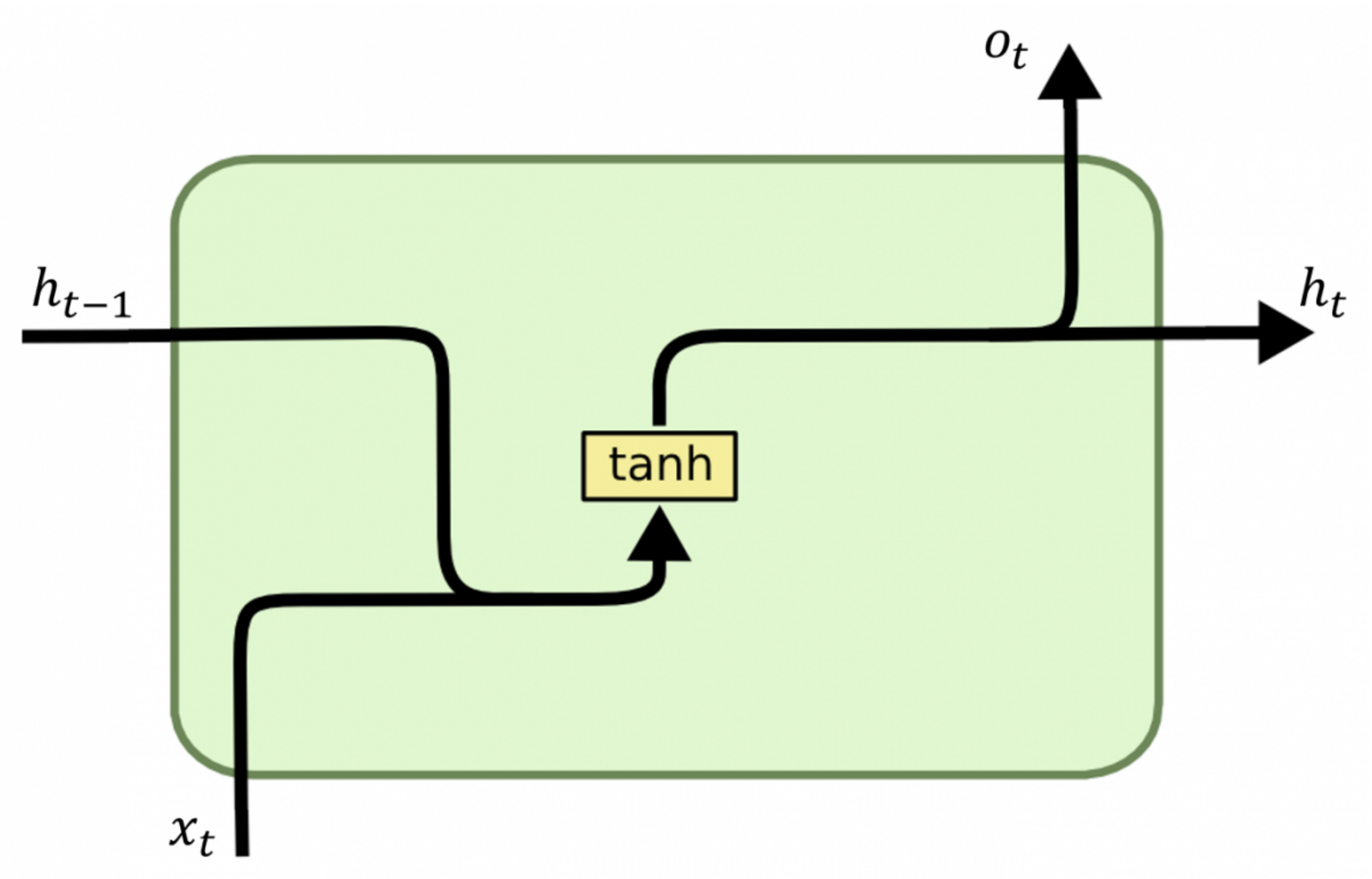
RNN



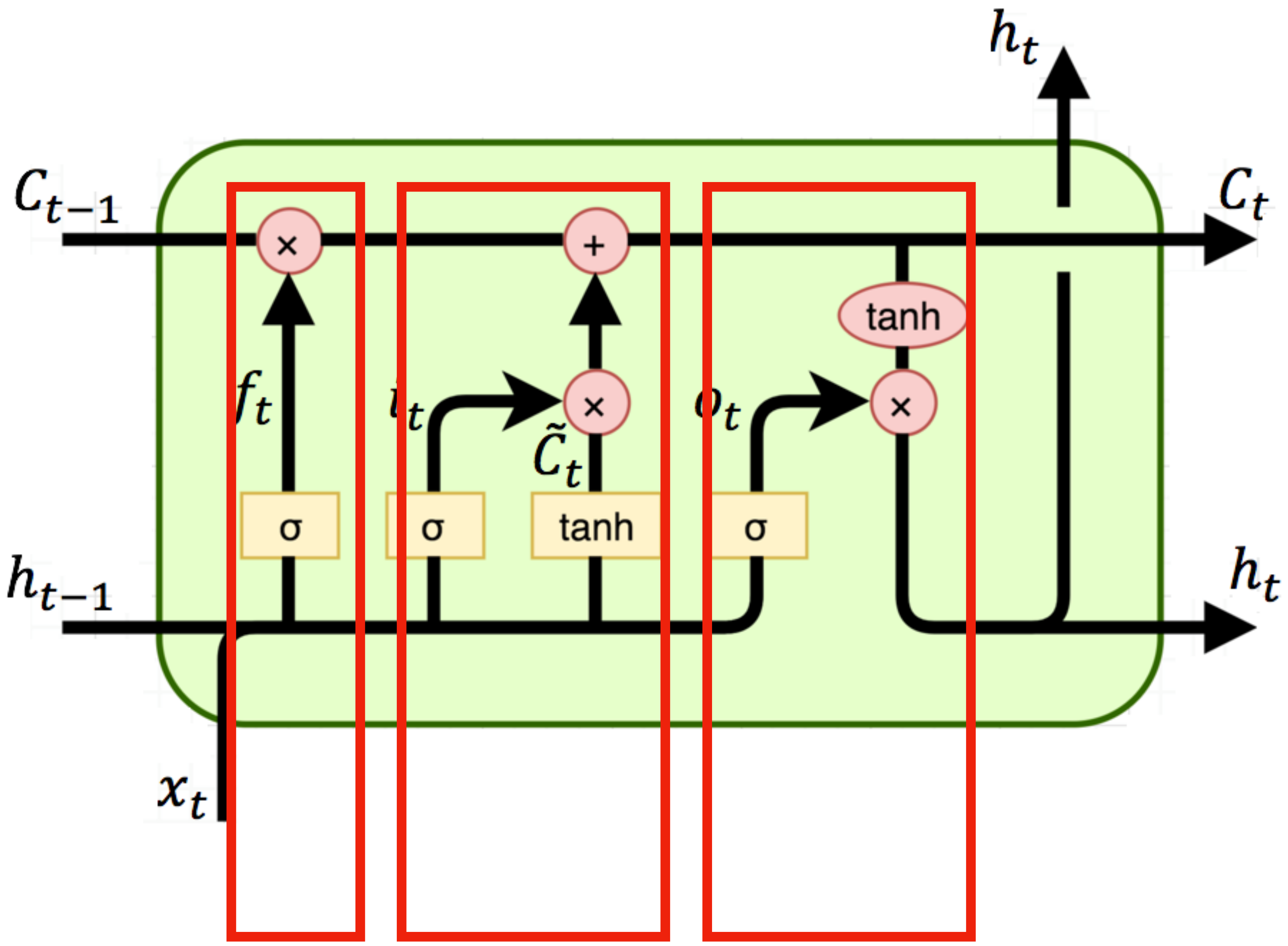
Long-Short Term Memory (LSTM)



RNN



Long-Short Term Memory (LSTM)

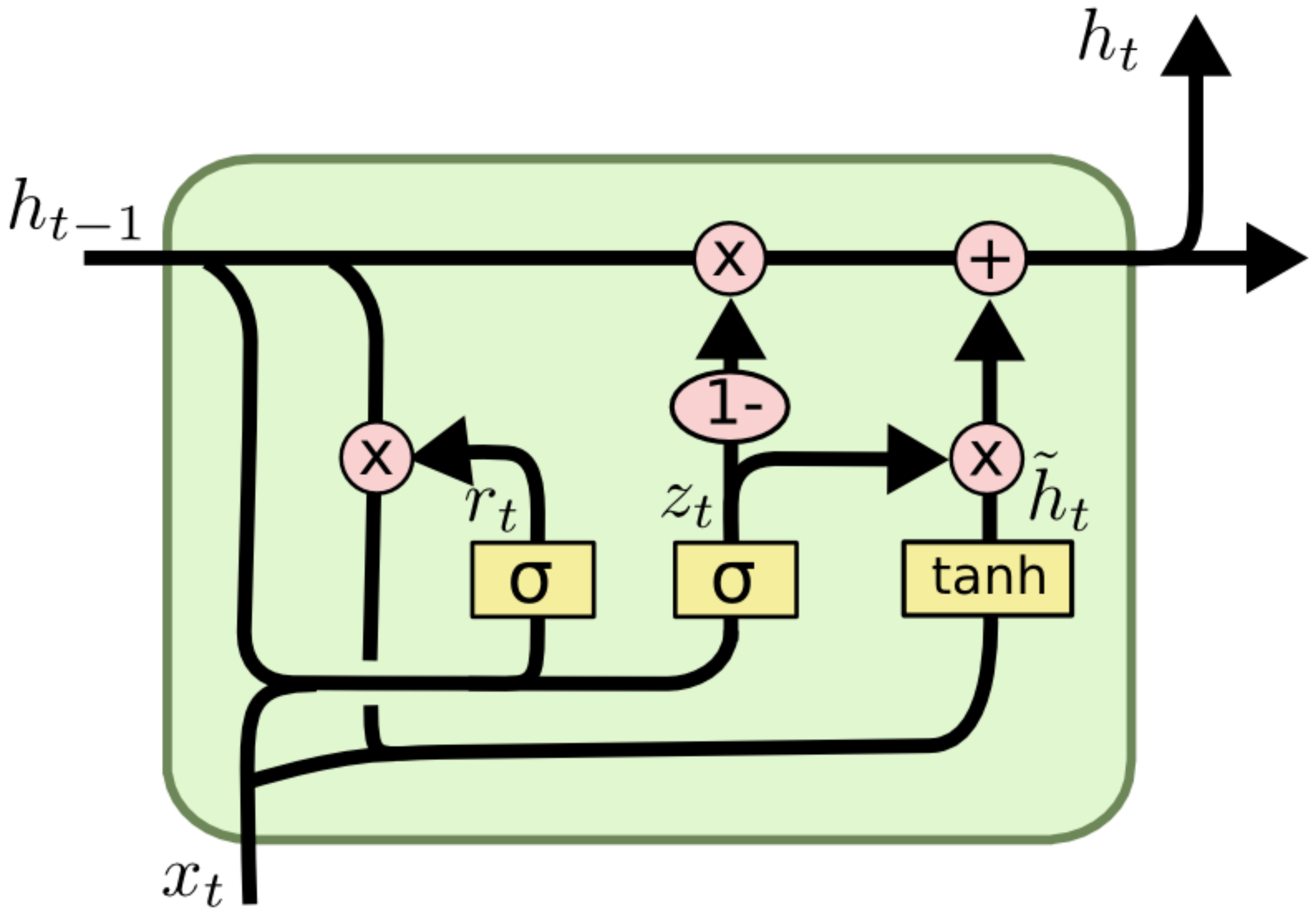


Forget Gate

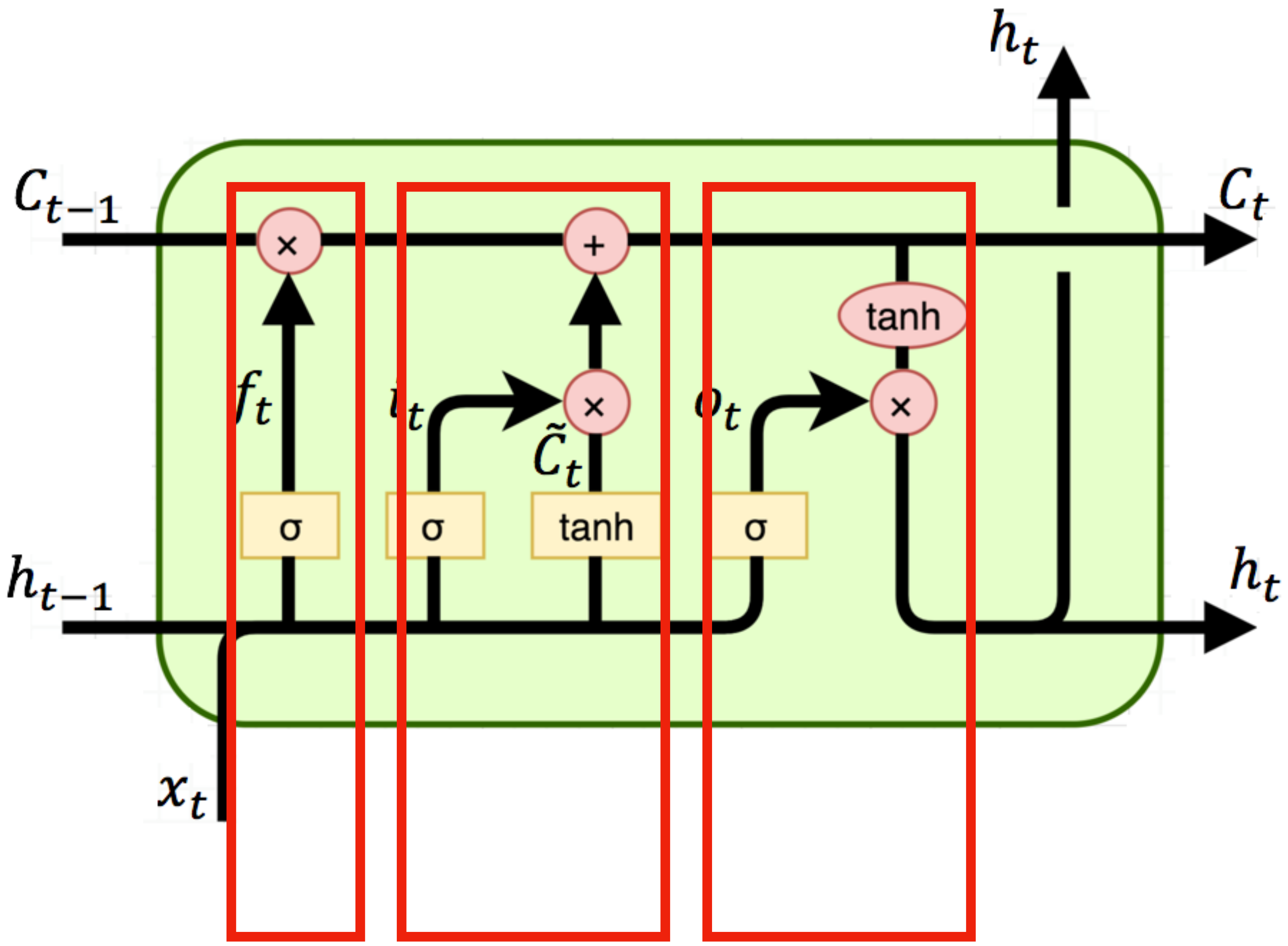
Output Gate

Input Gate

Gated Recurrent Unit
(GRU)



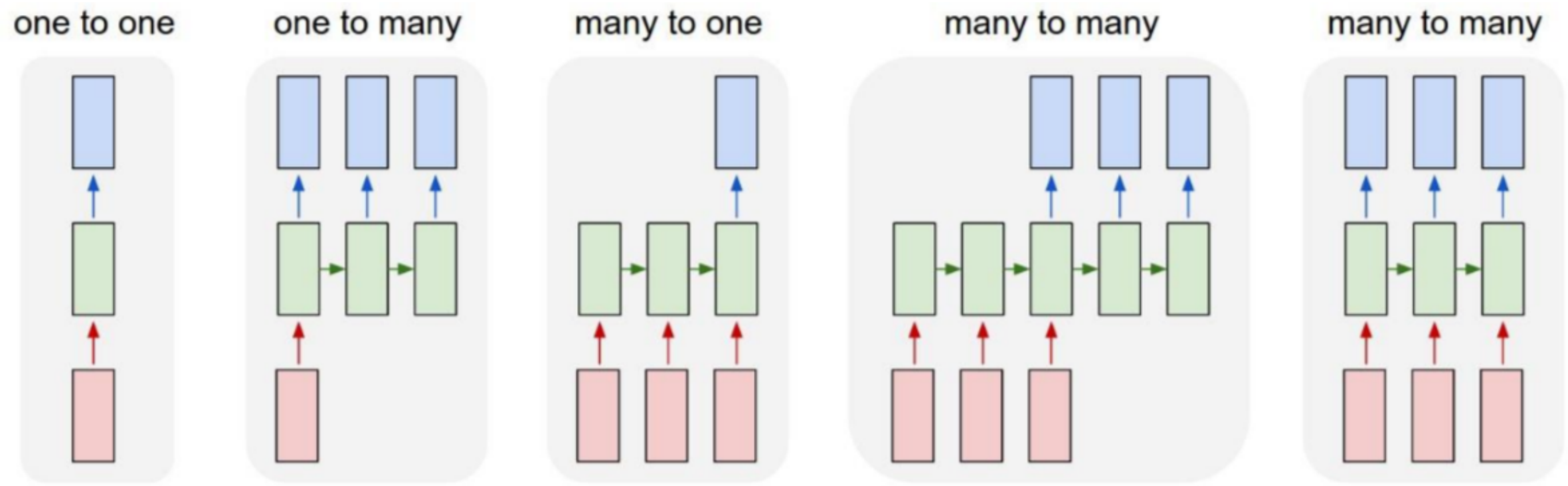
Long-Short Term Memory
(LSTM)



Forget Gate

Output Gate

Input Gate



Fixed-sized input to fixed-sized output (e.g. image classification)

Sequence output (e.g. image captioning takes an image and outputs a sentence of words).

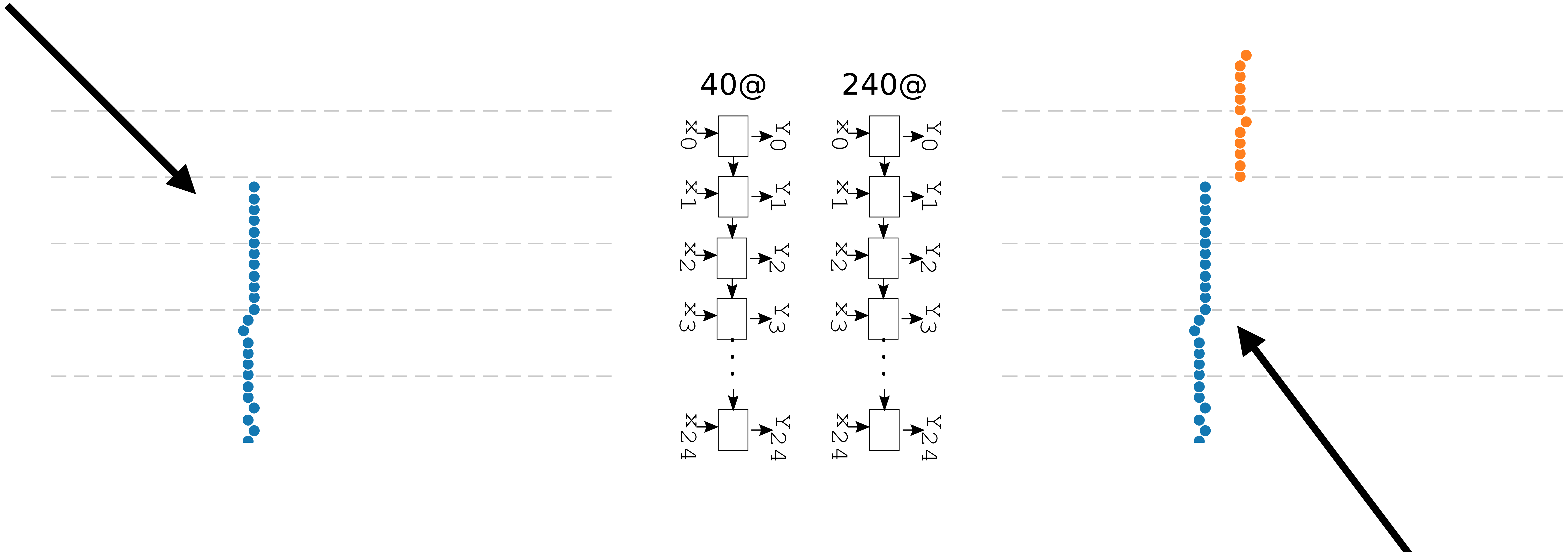
Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).

Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French)

Synced sequence input and output (e.g. video classification where we wish to label each frame of the video)

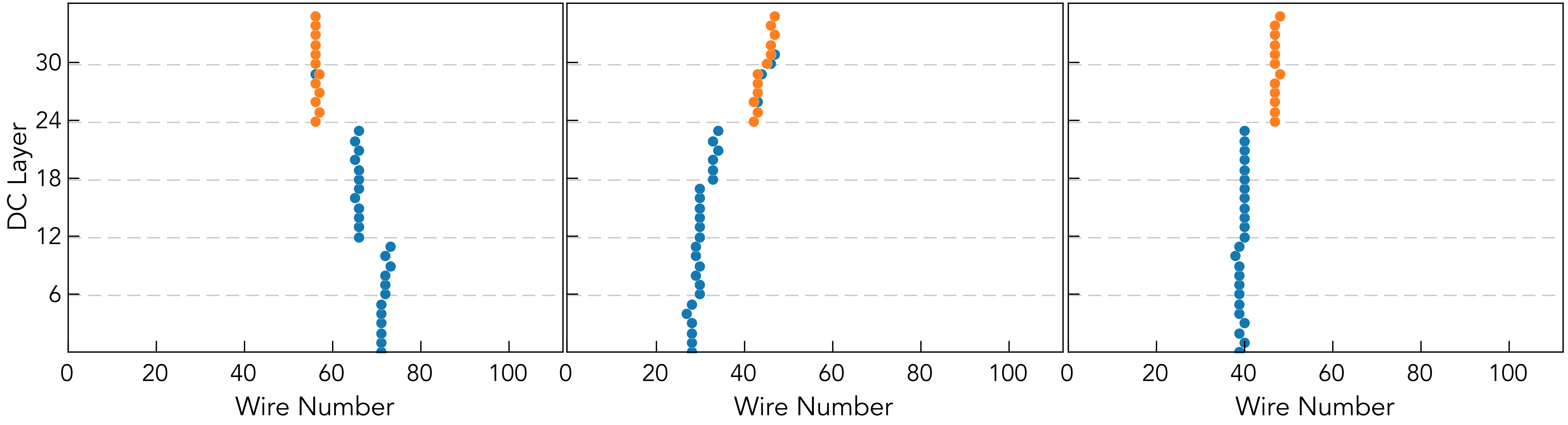
- The developed network will predict the hits in Super-Layer 1&2 (*where the noise is the highest due to proximity to the beamline*) based on hits in super layers 3,4,5 & 6.
- This network will be used in high-luminosity run analysis to aid the clustering algorithm in high occupancy regions of drift chambers.

Training INPUT contains 24 numbers (6 wires for each 4 super layers)

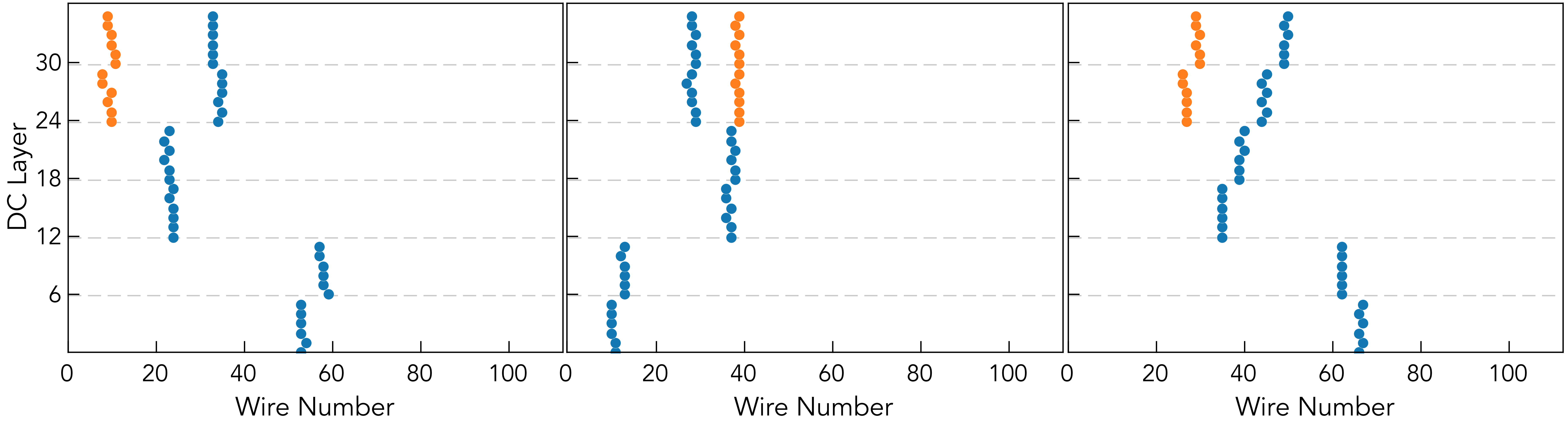


Training OUTPUT contains 36 numbers (6 wires for each 6 super layers)

True-Track Candidates



False-Track Candidates



Performance matrix comparing reconstructed true tracks depending
On the closeness cut of the distance of wires to the real track.

	Max Dist: 2	Max Dist: 3	Max Dist: 4
True Positive (%)	91.98	97.58	99.0
False Negative (%)	8.02	2.41	1.0
True Negative (%)	92.45	88.35	84.0
False Positive (%)	7.55	11.65	16.0

This network is an excellent aid for tracking detectors in high-luminosity experiments where parts of the detector (closer to the interaction point) have higher noise levels and need clustering or identified area for searching for track continuation

Software

- How to get started:
  - Python: https://www.python.org
  - TensorFlow: https://www.tensorflow.org
  - Scikit-learn: https://scikit-learn.org/
- Other Options:
  - DeepLearning4J (Java): https://deeplearning4j.konduit.ai
  - DeepNeets (Java) : https://www.deepnetts.com
  - EJML (Java): http://ejml.org
- Described Projects
  - All projects described in this talk are done in JAVA
  - The implementation of networks in CLAS12 reconstruction software is in JAVA
  - Java provides completely portable software deployable anywhere
  - Even the JDK can be shipped inside the software package

Backup Slides

Why Did we need the second Network (Auto-Encoder)

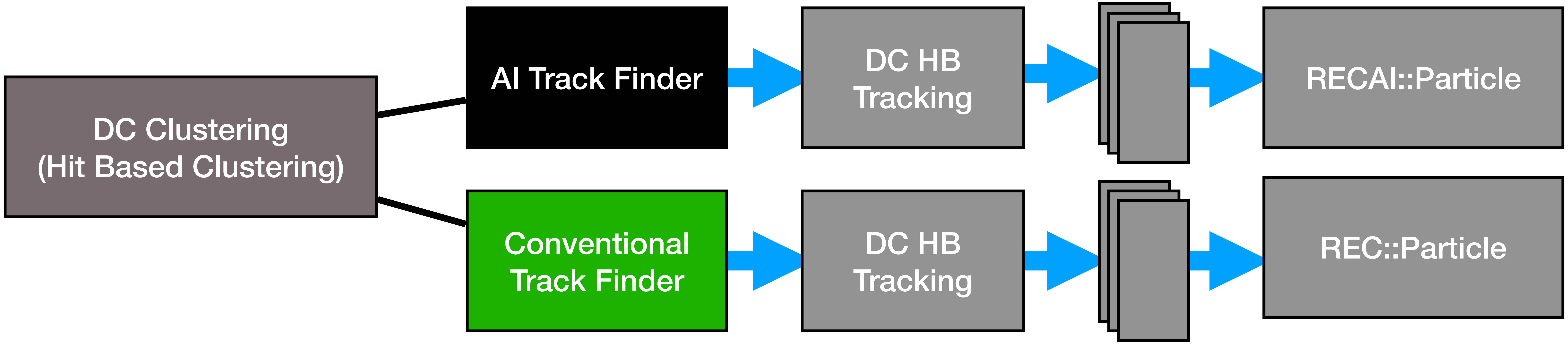
- Relative fraction of 5 super-layer tracks is about ~10% of total positively charge particles.
- The gain in number of 5 super-layer tracks is about x2.2 (120% increase)
- The gain in 6 super-layer track reconstruction with AI suggested track candidates is ~6%.
- Due to high gain in 5 super-layer track (where combinatorics is much larger for given number of segments) the total increase in tracks reconstructed is ~15.6%

Positive Charge	Conventional	Artificial Inteligence	Gain
6 CLUSTER	242,145	256,175	1.0579
5 CLUSTER	24,155	52,839	**2.1875**
TOTAL	267,339	309,058	**1.1561**

- Questions:
  - Are these real tracks ?
  - How does this translate into physics ?
  - Is this gain real ?

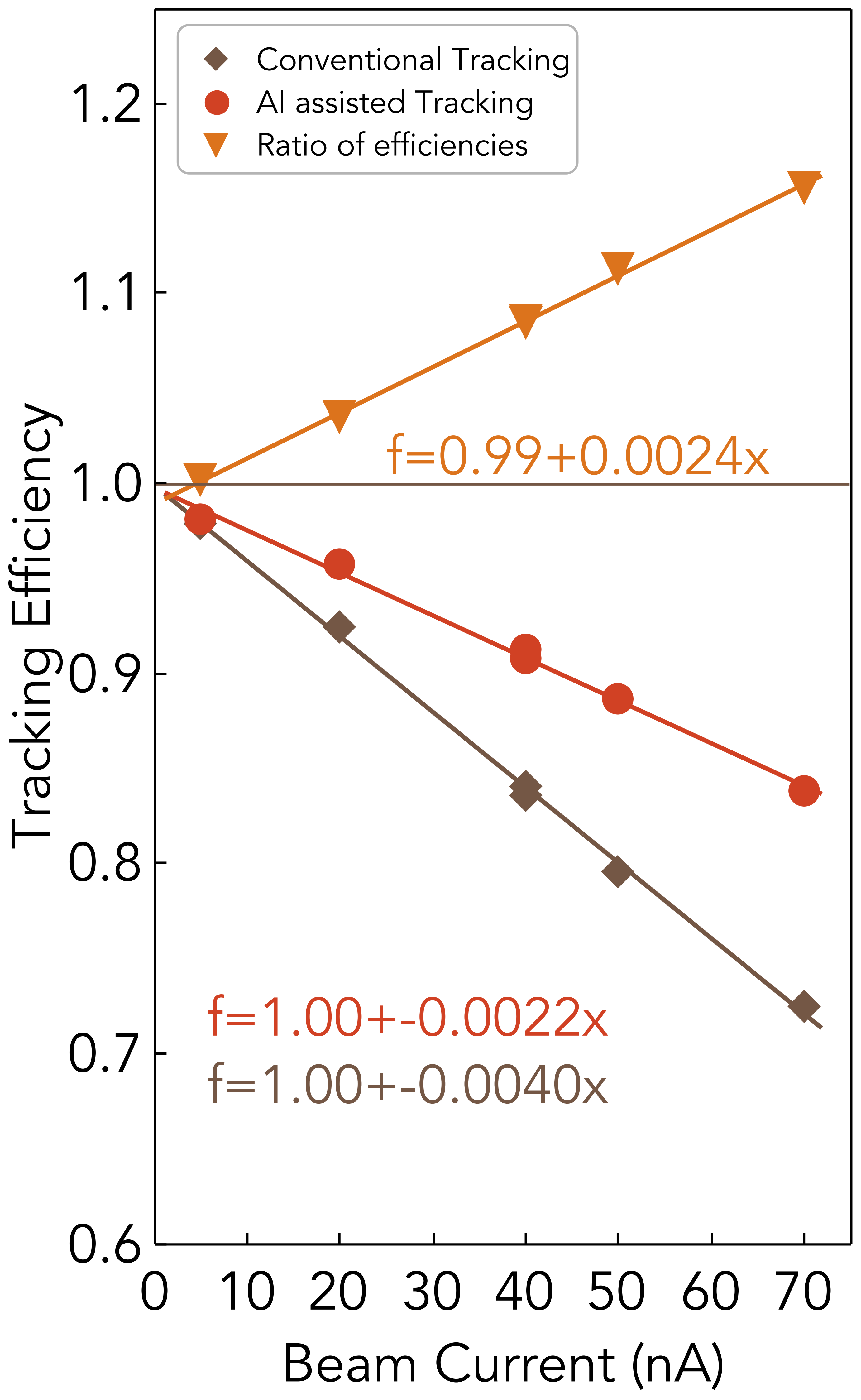
AI Tracking Reconstruction Tools

- AI track classification and segment recovery network was implemented as a CLARA service.
- Tracking code was modified to separate clustering from track finding



- Data analyzed in two parallel service compositions with separate output for Time Based Tracking
- The parallel branches produce separate particle banks
- Tracking code in the AI branch is 35% faster compared to conventional branch
- The full chain will be available soon for users to analyze and compare results from AI assisted tracking with conventional tracking.

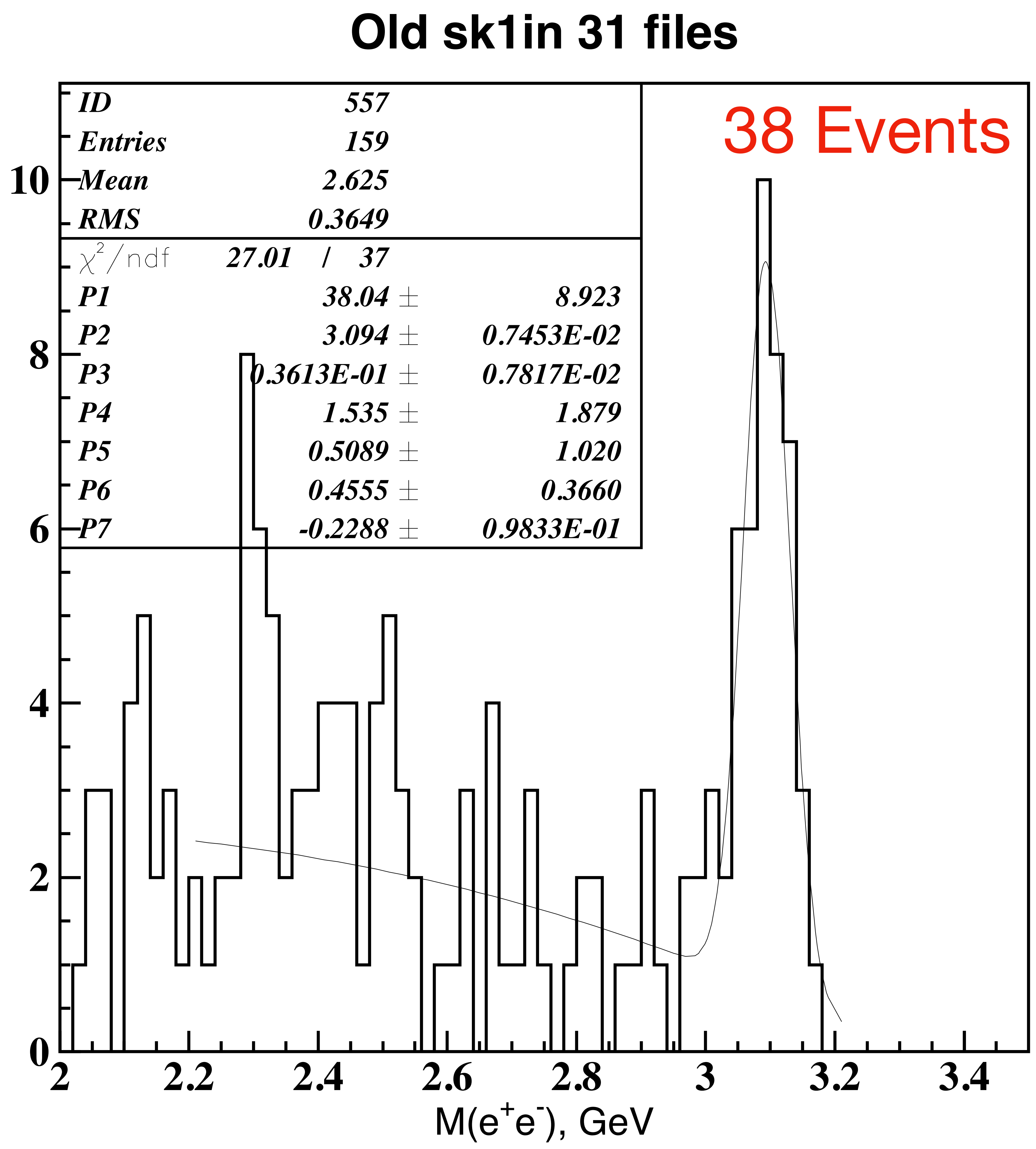
Physics Impact

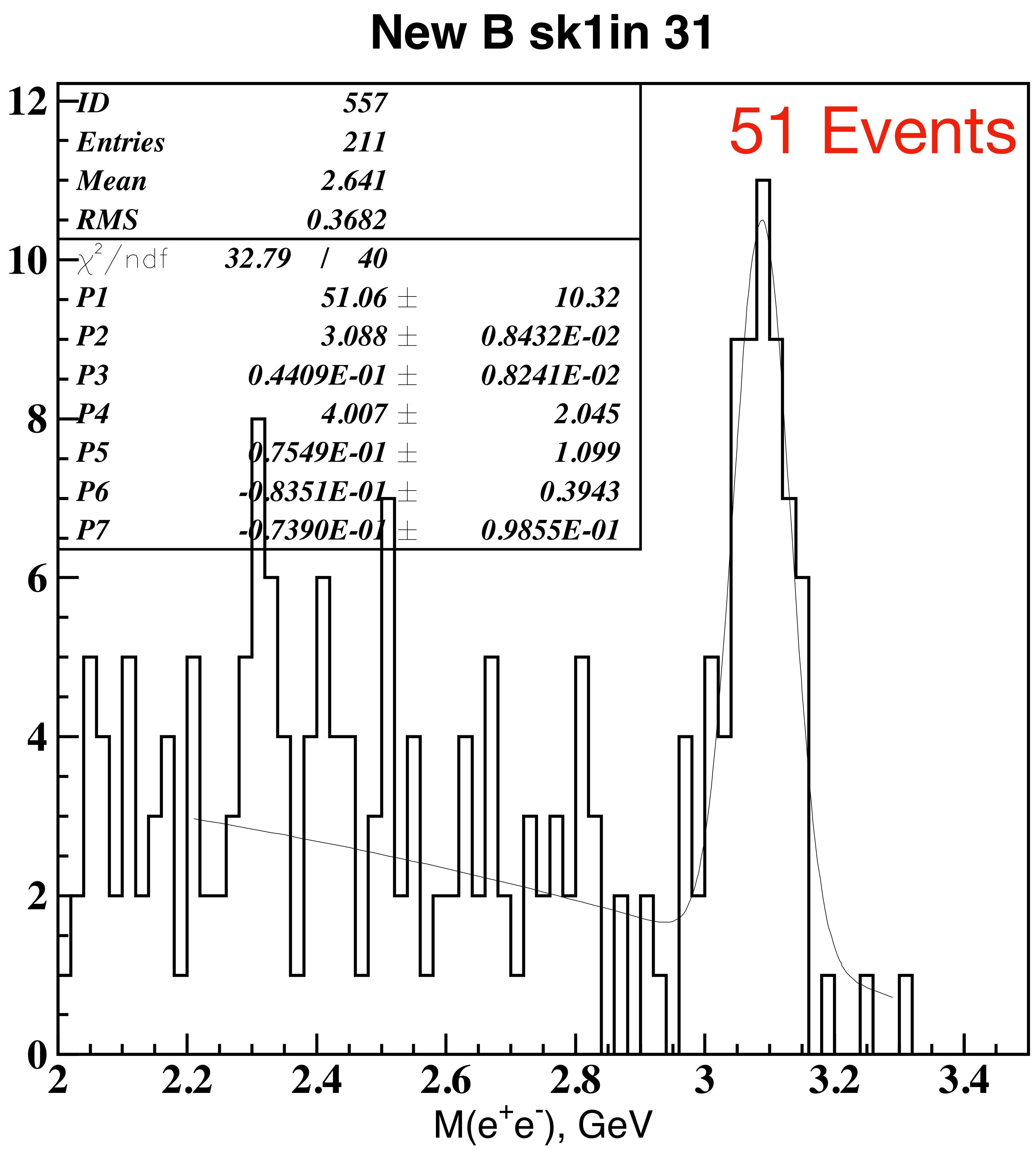


- Implementation of AI assistance in CLAS12 tracking lead to tracking speed improvement of ~35%.
- Particle reconstruction efficiency increased when using only AI suggested tracks.
- Study was performed to measure tracking efficiency as a function of experiment luminosity (beam current)
- Conventional tracking efficiency decreases by 0.40% per nA of beam current.
- AI assisted tracking efficiency drops by 0.22% per nA.
- Efficiency drop improved by factor of ~2x.

S.Stepnyan's Analysis

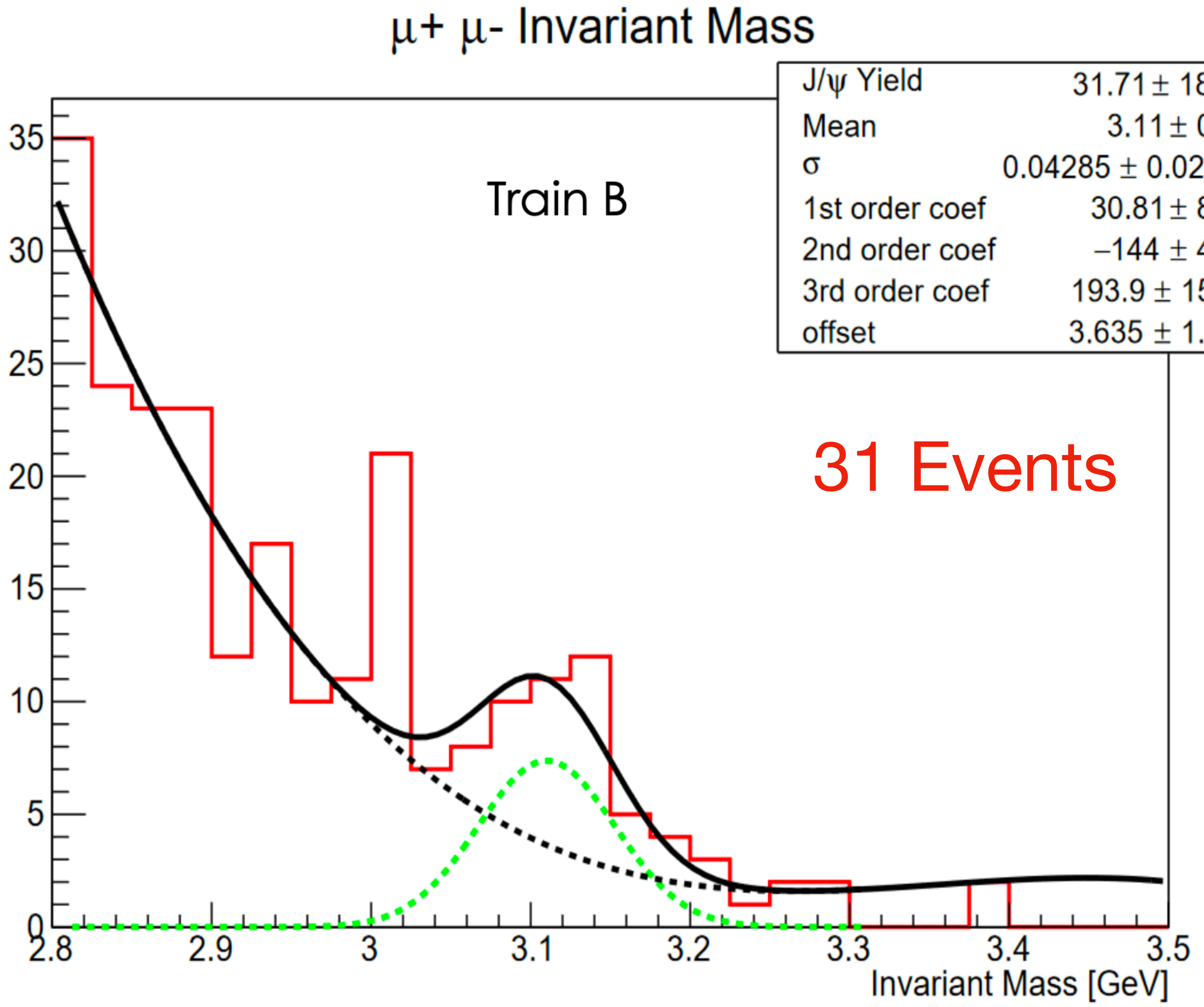
J/ψ 25% more than in pass1

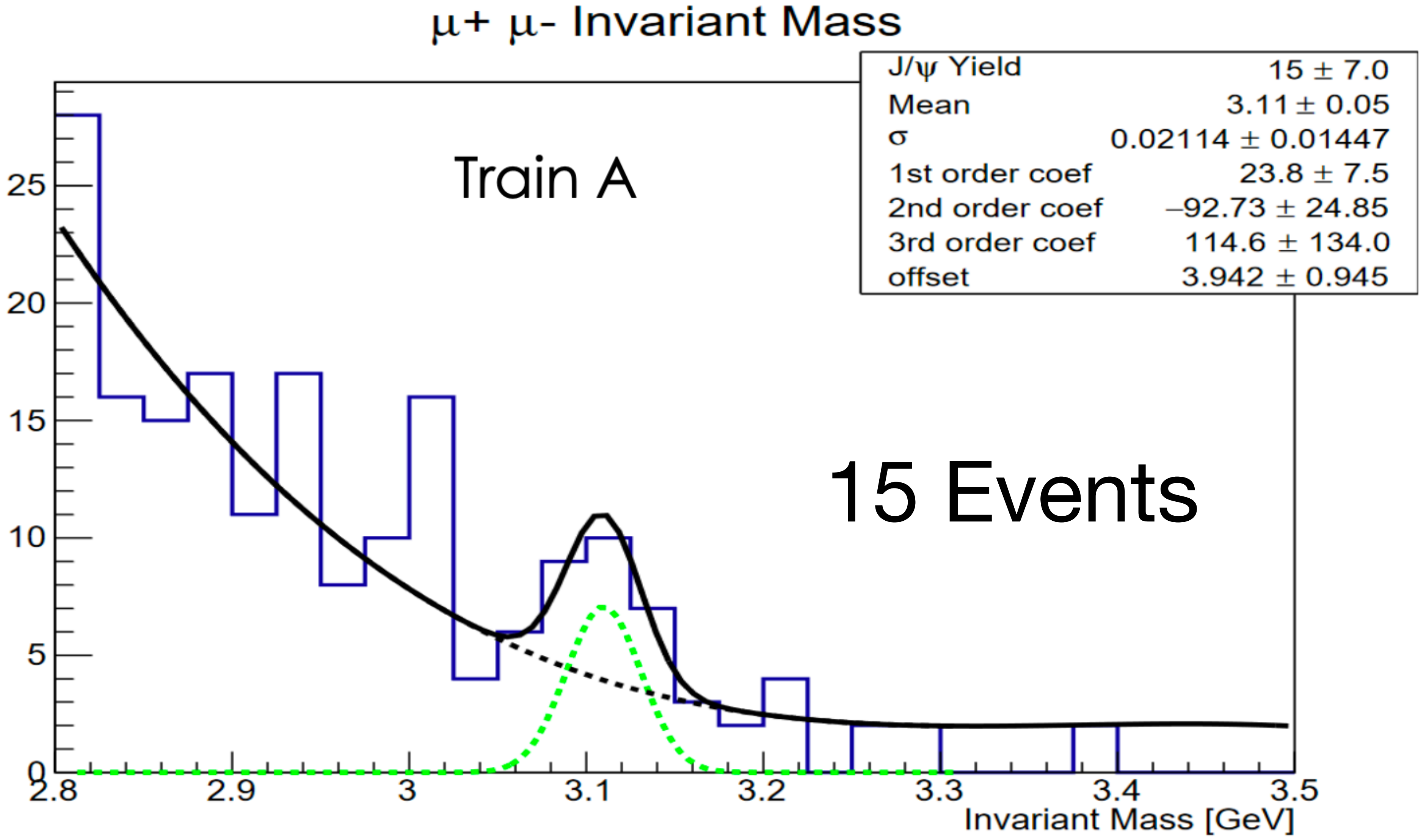


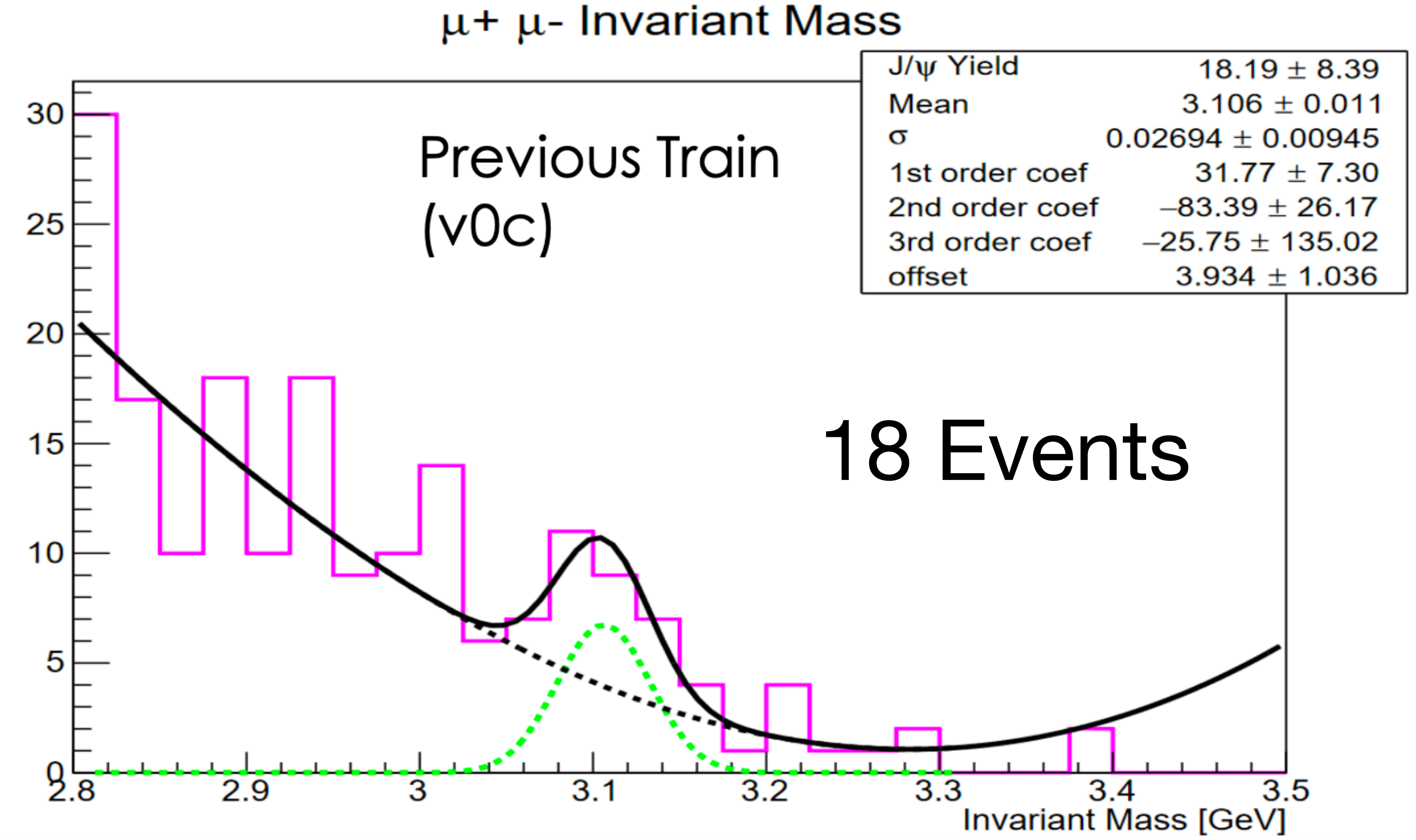


R.Tyson's Analysis

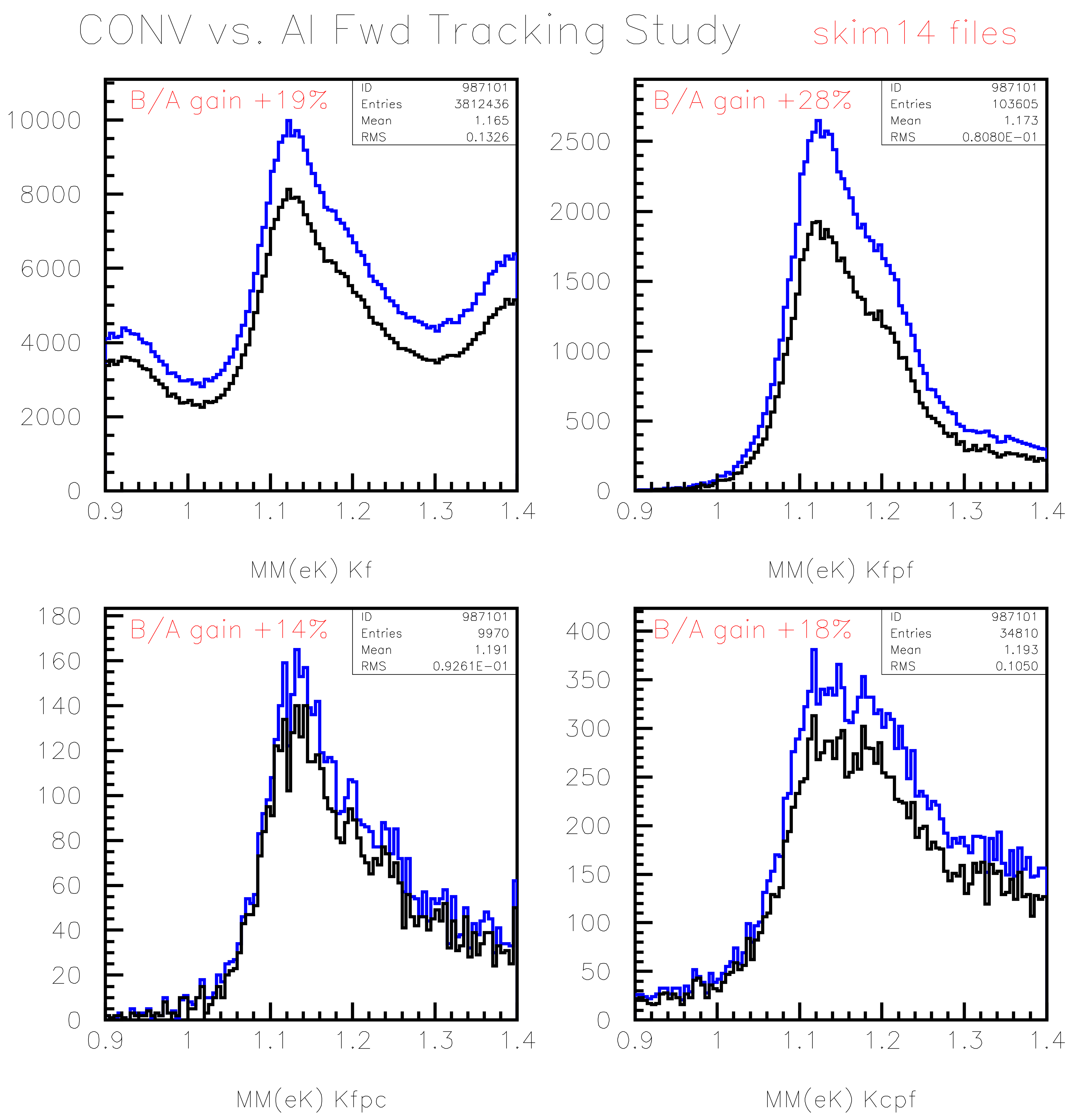
$\mu^+\mu^-$ Invariant Mass







D. Carman's Analysis



UL: e'K+(FD) - B has 19% more events than A
UR: e'K+(FD)p(FD) - B has 28% more events than A
LL: e'K+(FD)p(CD) - B has 14% more events than A
LR: e'K+(CD)p(FD) - B has 18% more events than A

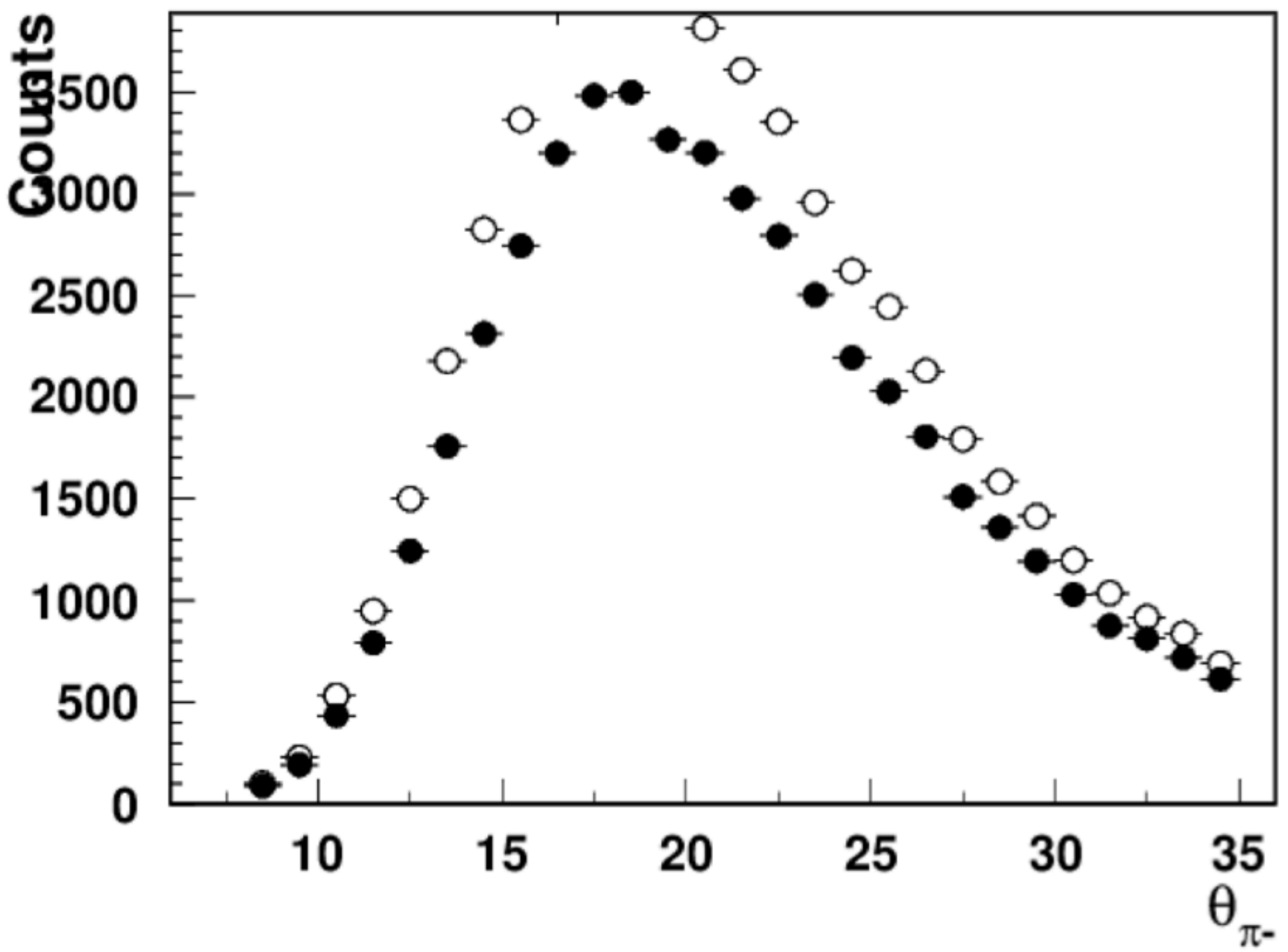
Monte-Carlo Validation

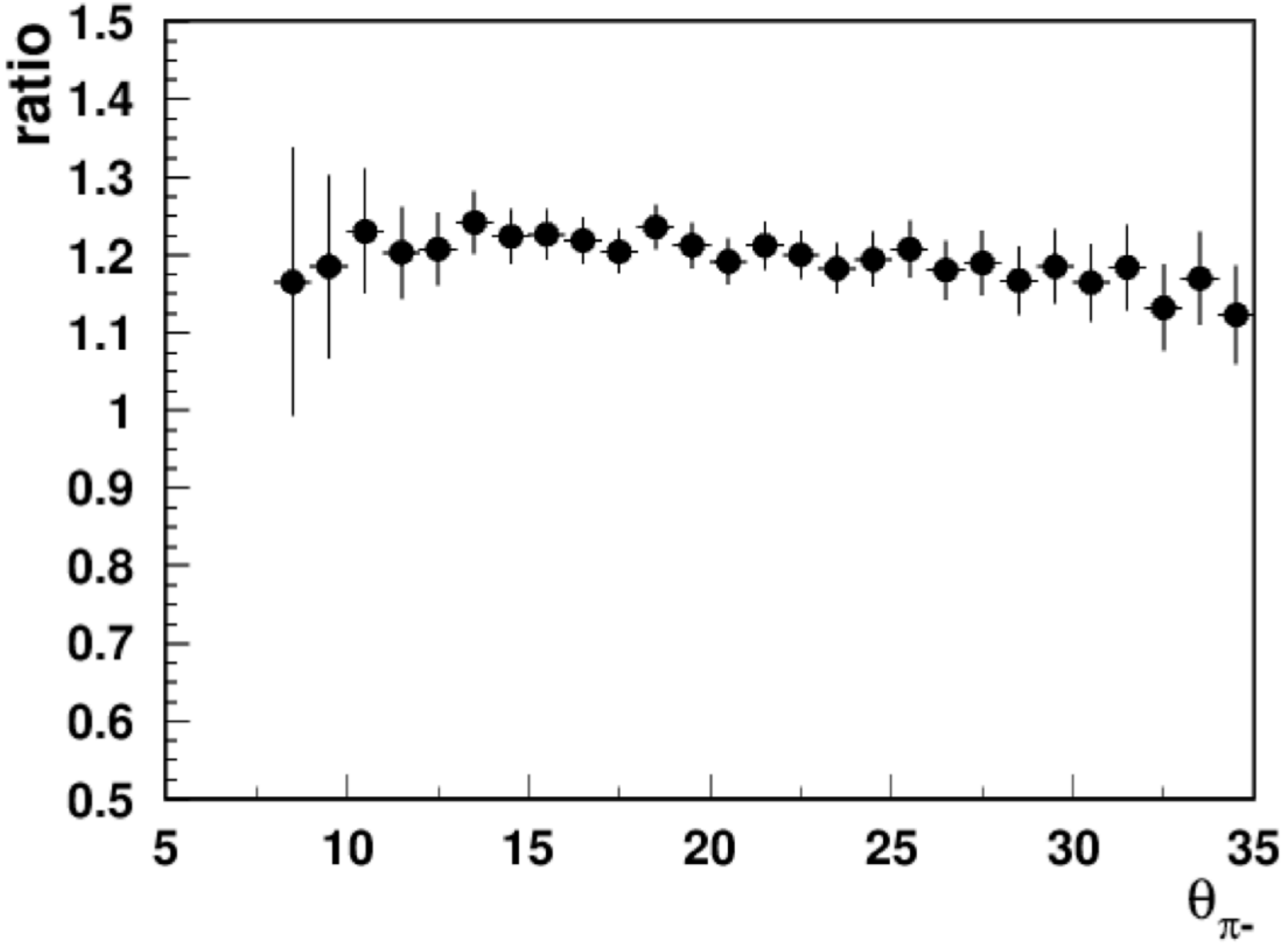
H. Avakian's Analysis

Comparing B and A production in MC: with background

All ratios B/A

All reconstructed particles are checked to be valid particles using generated MC, and identified properly by the EB

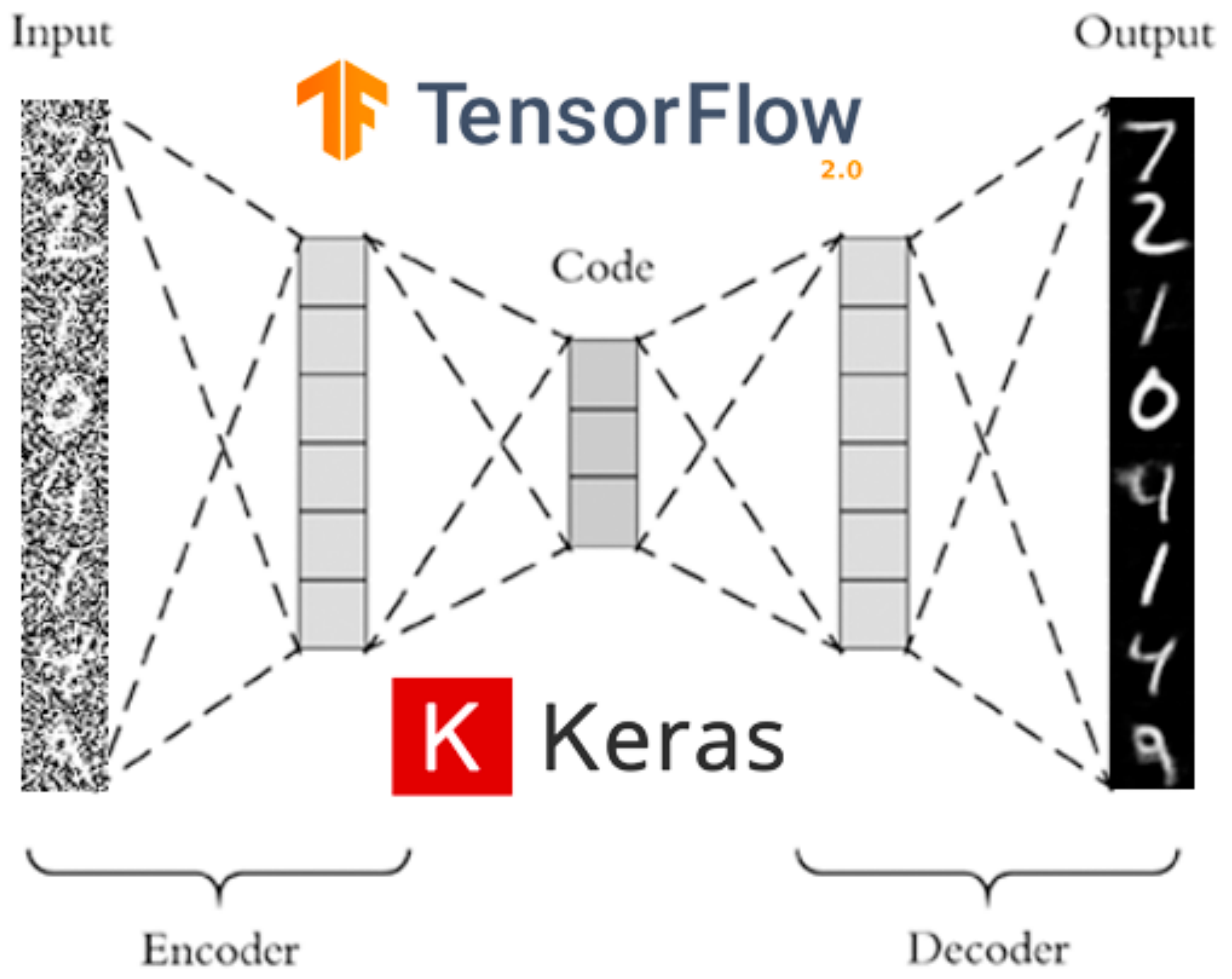


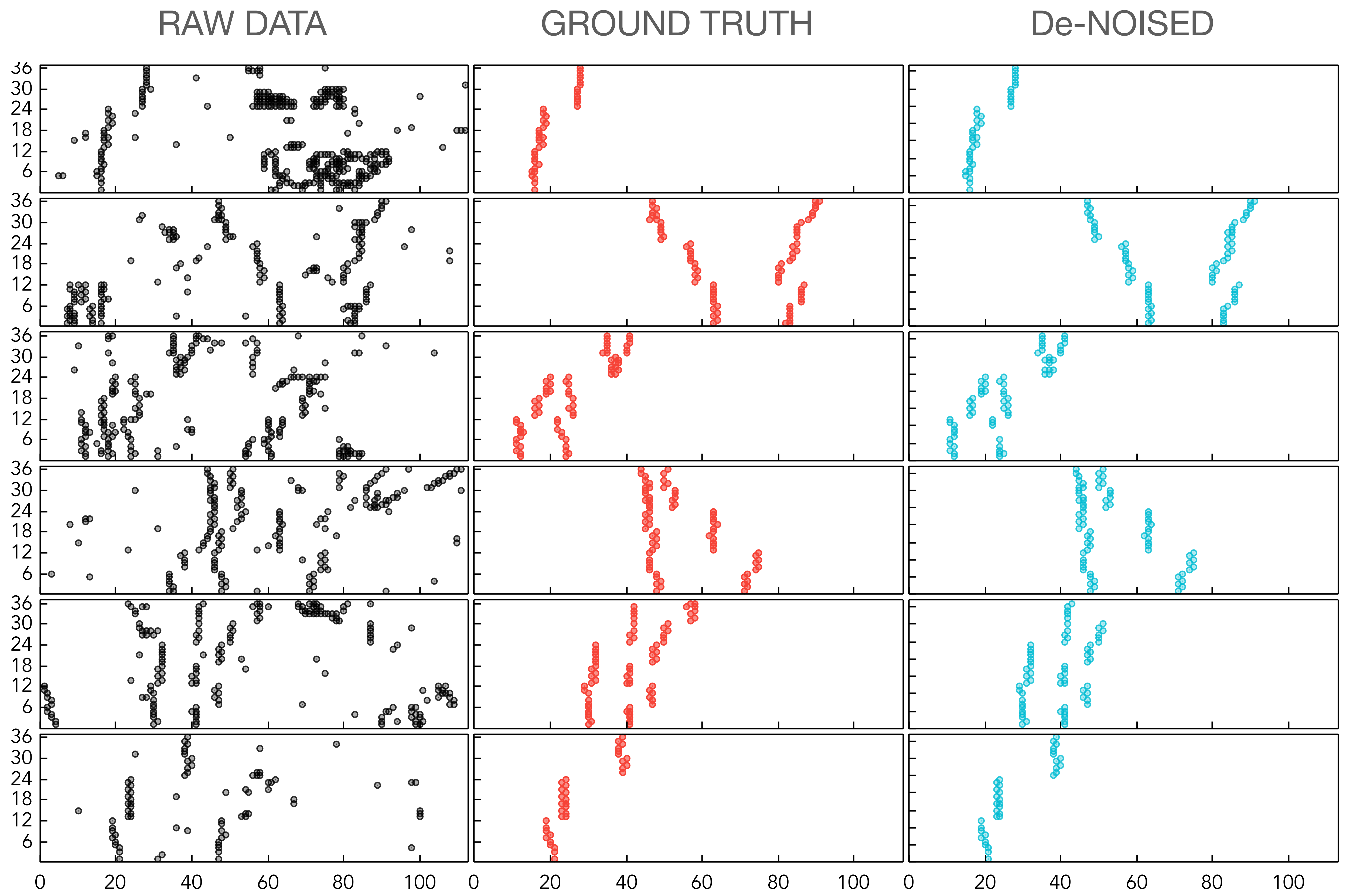


There is increase for B-version for ~20% e-pi-

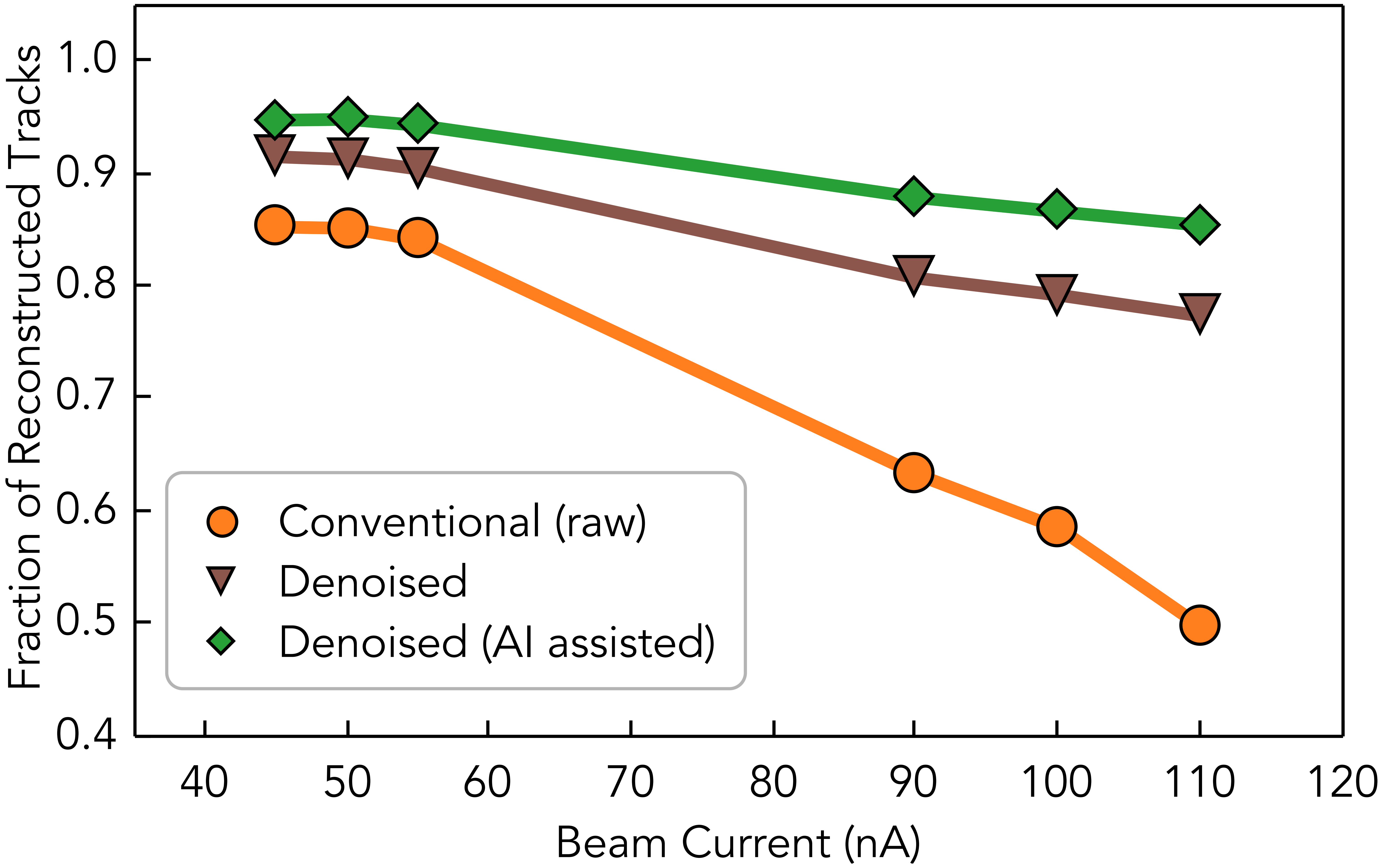
De-noising (CLAS12 Drift Chambers)

- Using Convolutional Auto-Encoders we can clean raw data sample to leave only hits that belong to a track.
- Network is trained on "good" reconstructed tracks from experimental data.



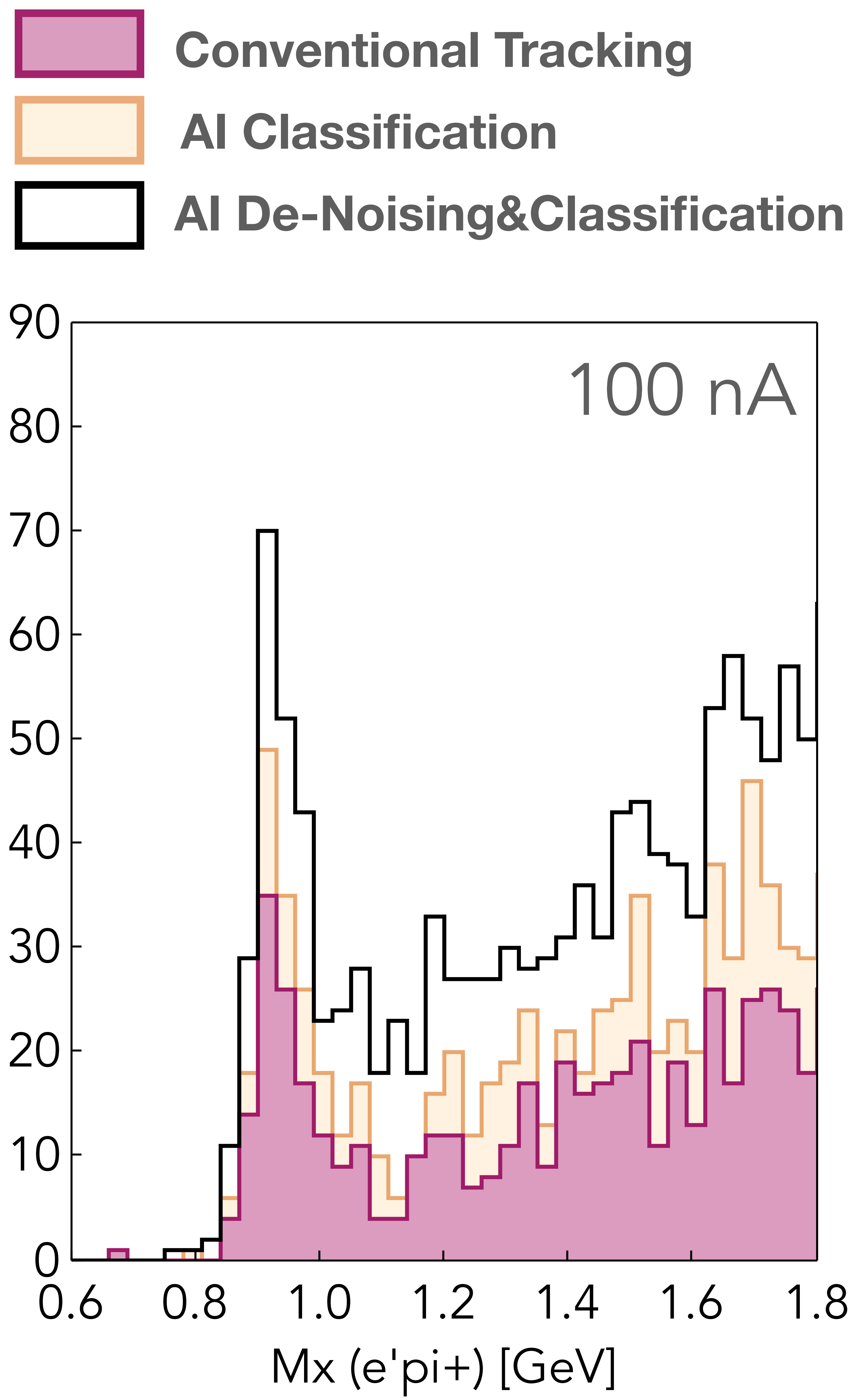


Published: Comput.Phys.Commun. 271 (2022) 108201



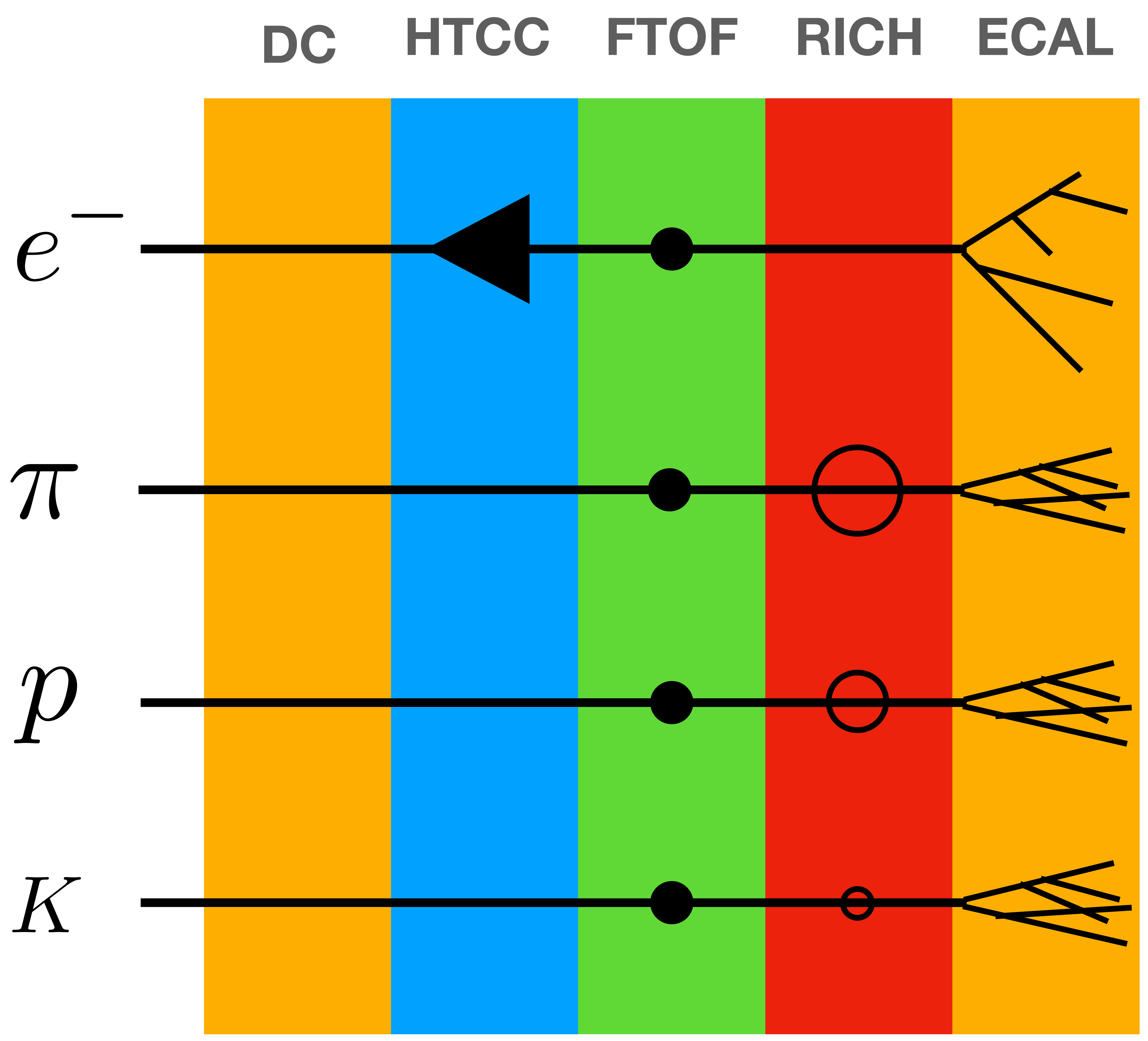
What We Learned so Far ?

- AI assisted tracking:
  - Two types of Neural Networks are developed to assist tracking code:
    - Track candidate classifiers
    - Inefficiency recovery network based on Auto-Encoders
  - The implementation in standard reconstruction code lead to improvements:
    - Tracking code speedup of ~35%.
    - Particle track reconstruction efficiency improvement of ~15% for standard running conditions (40-50nA).
- Physics Impact
  - Improved efficiency for physics outcome for multi particle final sates
    - Improvement in statistics 20%-35% (for standard running conditions)
- High Luminosity Running
  - De-noising DC and running standard reconstruction shows significant improvements in particle reconstruction.
  - With both de-noising and AI assistance makes it possible to run at higher luminosities.
  - De-Nosing network is ready and can be implemented as part of the workflow.
  - Heavy restructuring of DC code is needed to implement de-noising.



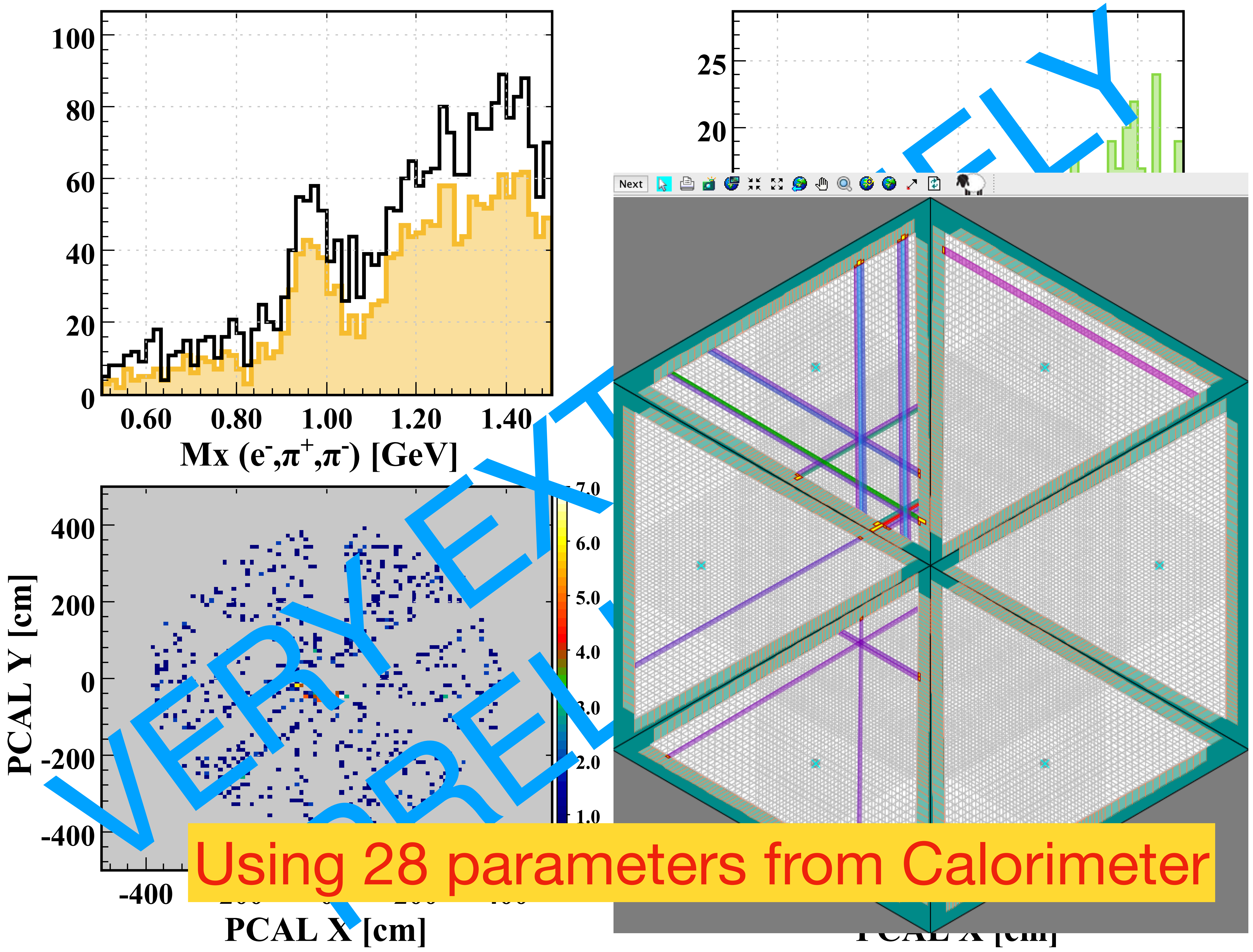
Particle Identification

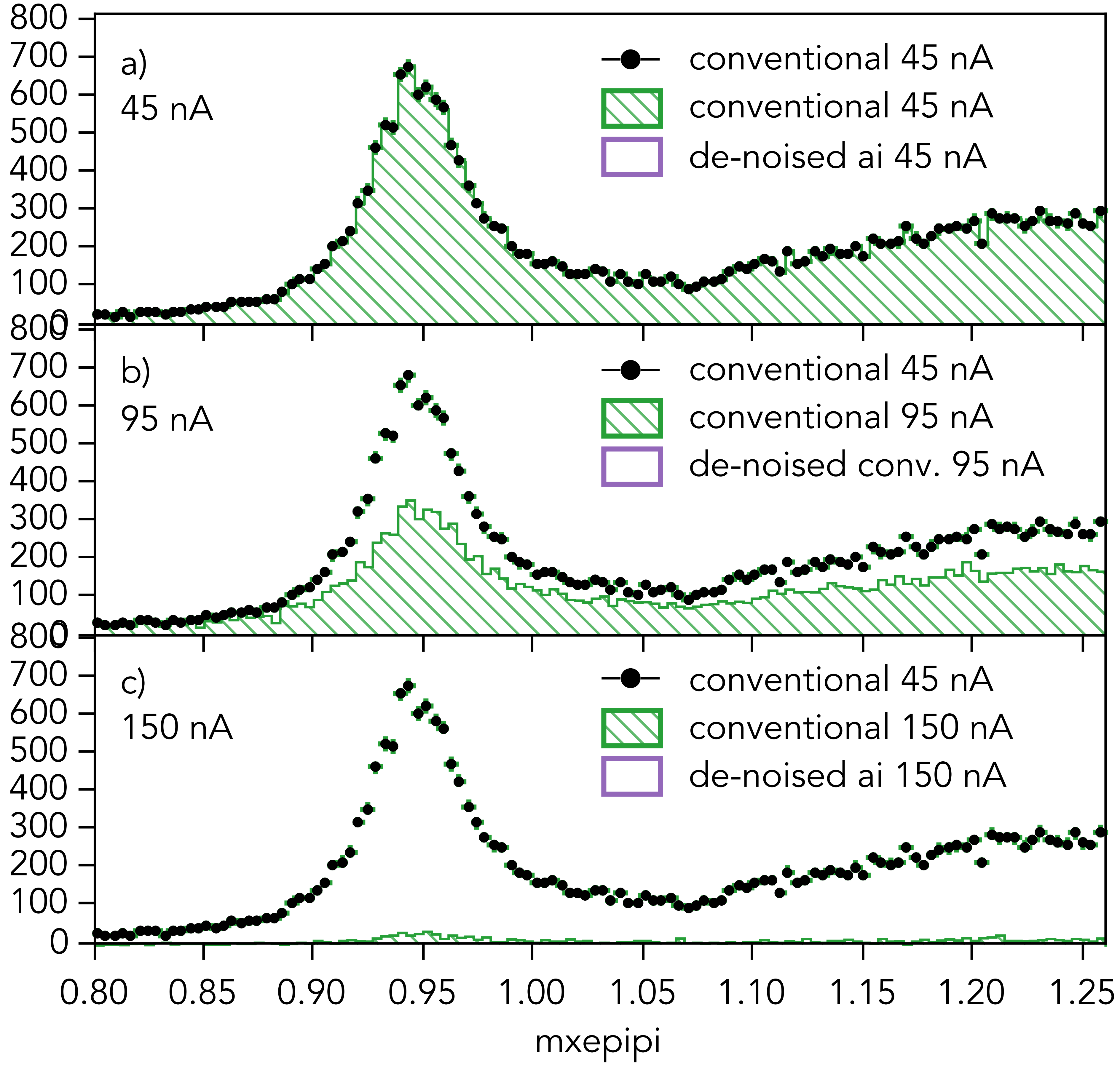
- Identify particles based on detector responses
- Each particle is assigned features depending on traveled path:
  - Drift Chamber track trajectory
  - High Threshold Cherenkov response
  - Time of Flight
  - RICH detector response
  - Electromagnetic Calorimeter shower (energy, shape and moments of the shower)
- Neural Network will be trained on Monte-Carlo data.
- Reaction specific framework will be developed to refine particle identification for specific final states.



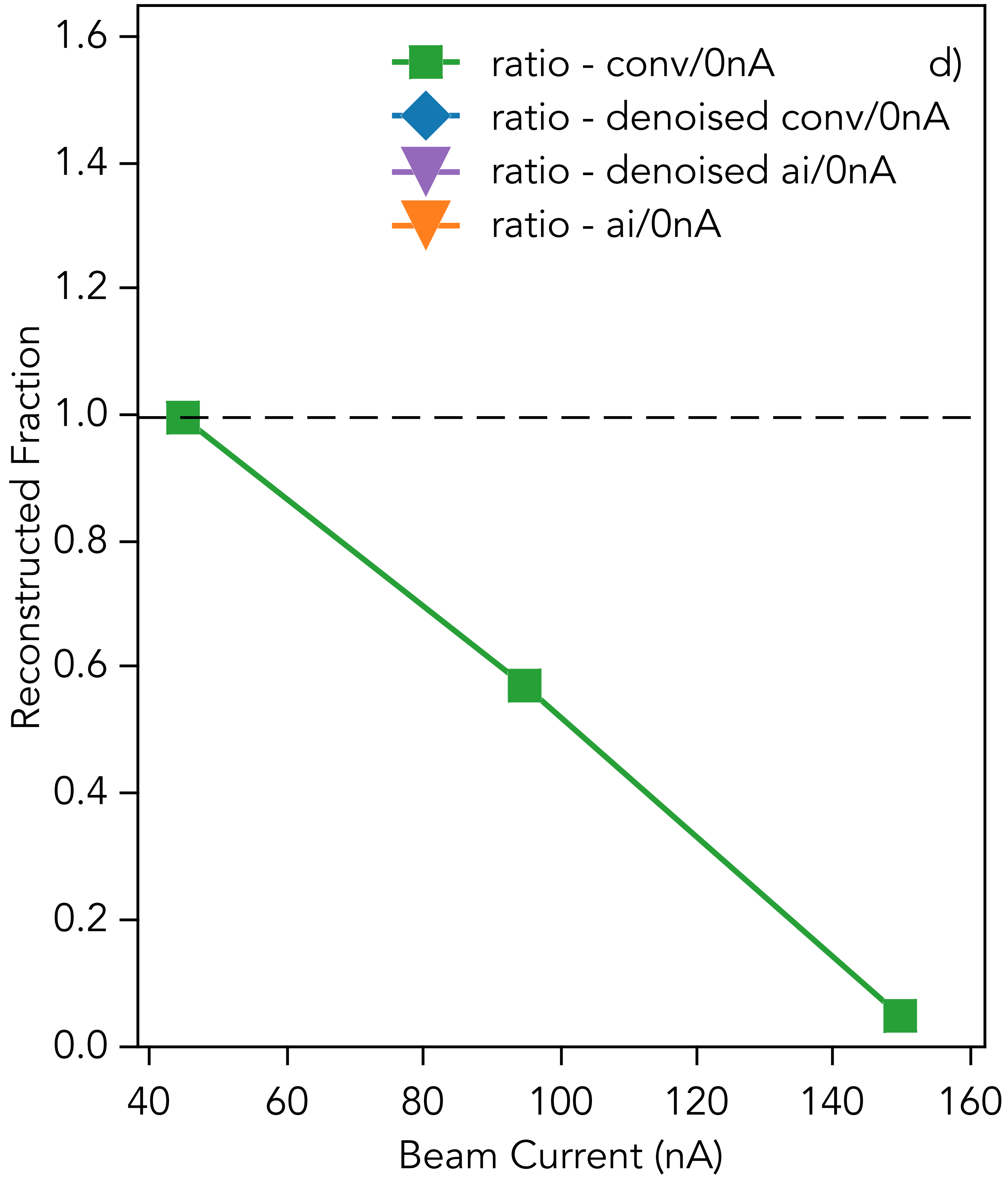
Particle Identification AI

- Network was trained on uniformly generated electrons, pions and muons in GEMC
- Testing was done on Inclusive e,pi+,pi- event simulated using Pythia
- What is the physics impact ?
  - Electrons identified by Neural Network lead to increased statistics in the exclusive e,pi+,pi-,p event
  - The recovered electrons seem to come from edges of calorimeter where there are some energy losses, and AI can identify them better than a a simple Sampling fraction CUT.





a)
45 nA
conventional 45 nA
conventional 45 nA
de-noised ai 45 nA
b)
95 nA
conventional 45 nA
conventional 95 nA
de-noised conv. 95 nA
c)
150 nA
conventional 45 nA
conventional 150 nA
de-noised ai 150 nA
mxepipi



ratio - conv/0nA
ratio - denoised conv/0nA
ratio - denoised ai/0nA
ratio - ai/0nA
d)
Reconstructed Fraction
Beam Current (nA)