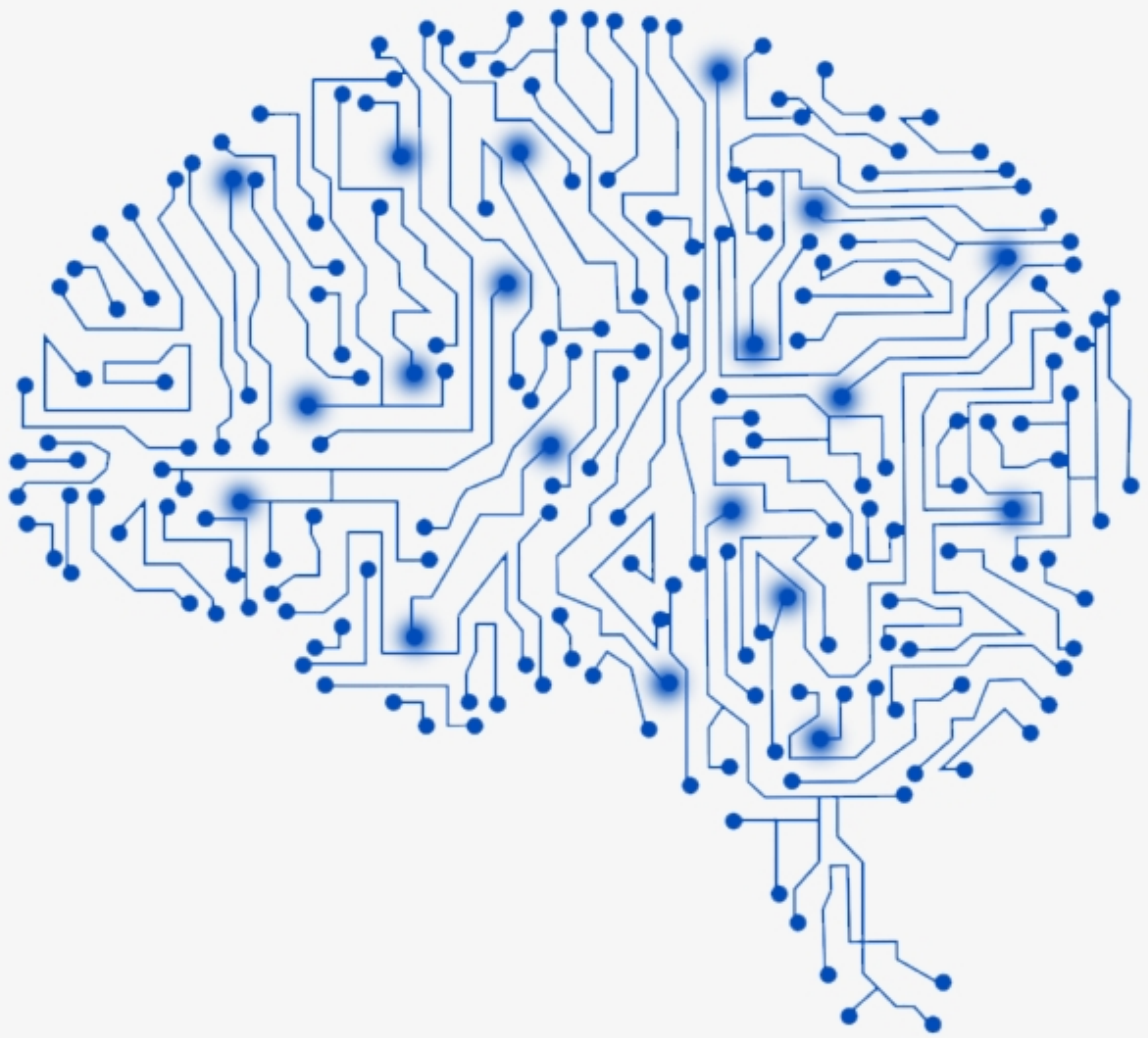
Artificial Intelligence

Artificial Intelligence/Machine Learning for Physics Applications

G.Gavalian (Jefferson Lab)

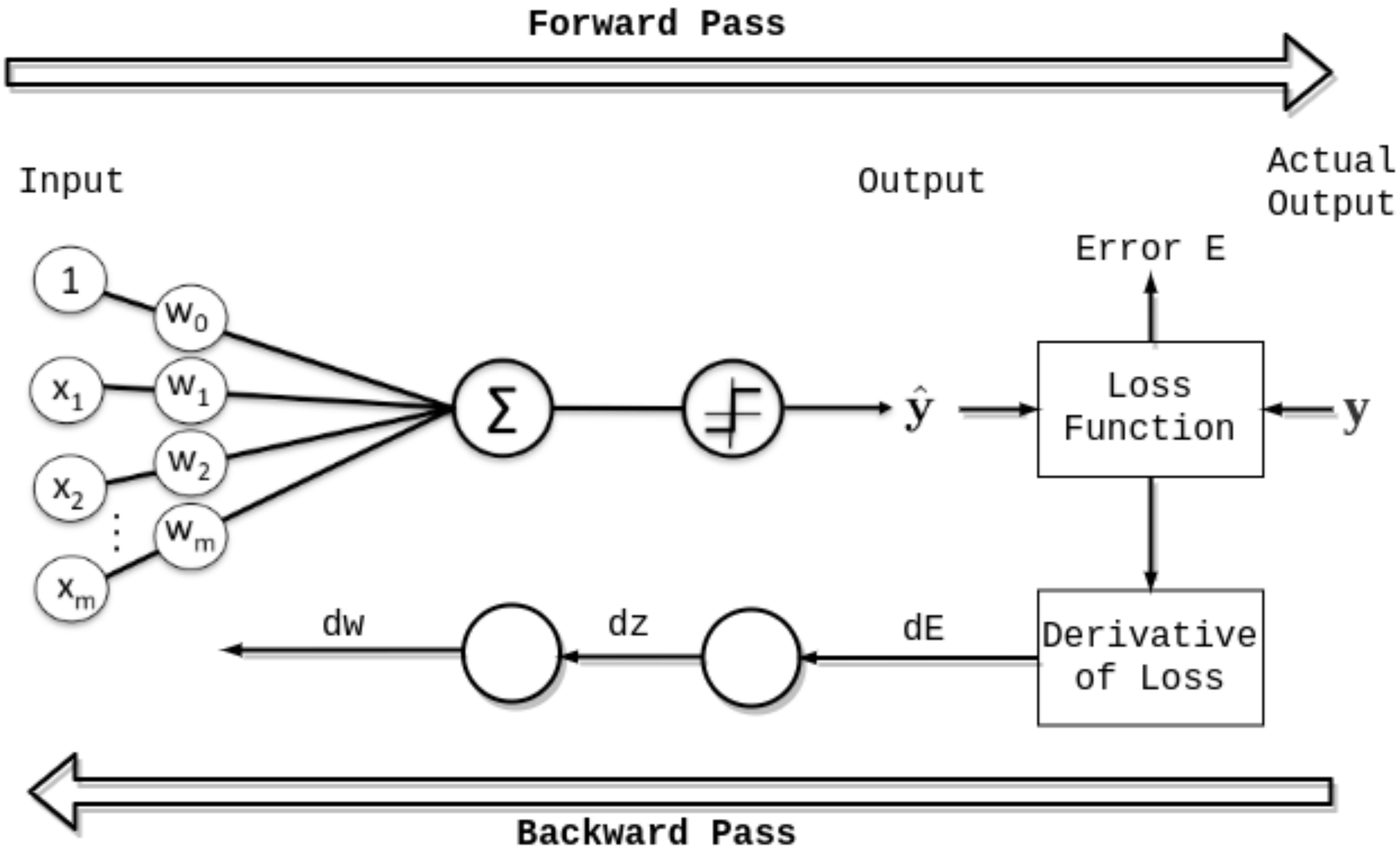


HUGS (June 7-9,2022)

Projects

- Introduction to Machine Learning
  - Multi-Layer Perceptron
  - Convolutional Neural Networks
  - Extremely Randomized Trees, Gradient Boosted Trees
  - Long-Short Term Memory networks (RNN)
- Applications in physics
  - Particle Tracking assistance
  - Experiment Triggering
  - Physics Reaction identification
- Discussions on Network Types and their applications
  - Classifier networks
  - Linear Regression Networks
  - Logistic Regression
  - Series Prediction
  - Auto-Encoders, and their uses

Machine learning (ML) 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. It is seen as a part of artificial intelligence. 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.

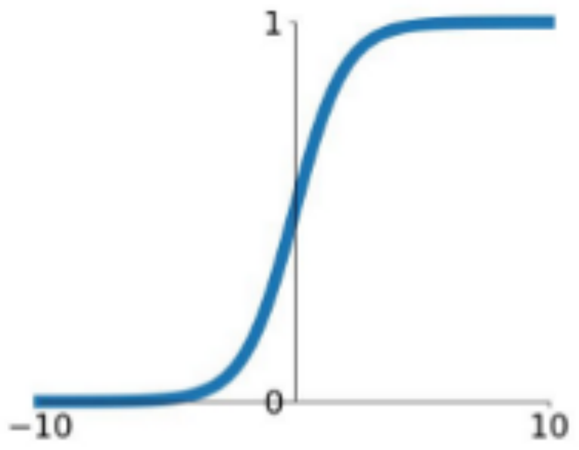

Forward Pass
Input
Output
Actual Output
Error E
1
w_0
x_1
w_1
x_2
w_2
w_m
x_m
Σ
ŷ
Loss Function
y
dw
dz
dE
Derivative of Loss
Backward Pass

Connecting the Neural Network Nodes

Activation Functions

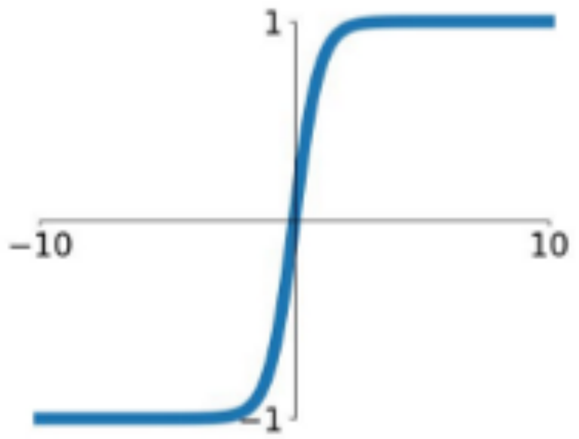
Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



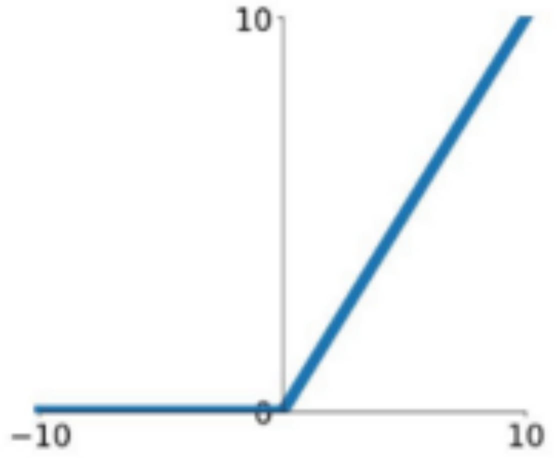
tanh

$\tanh(x)$



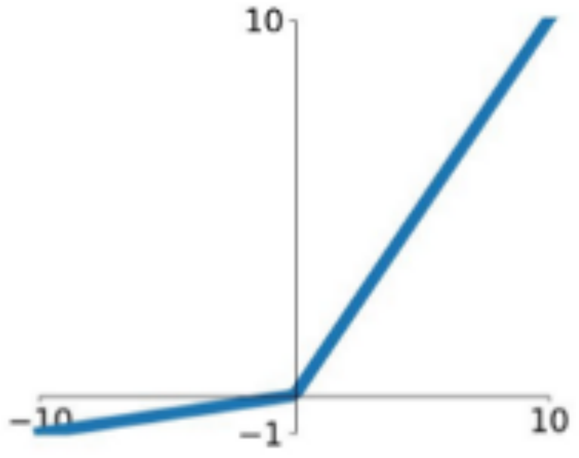
ReLU

$\max(0, x)$



Leaky ReLU

$\max(0.1x, x)$

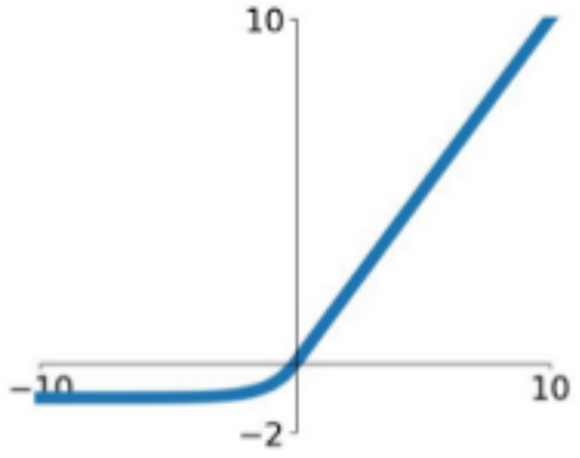


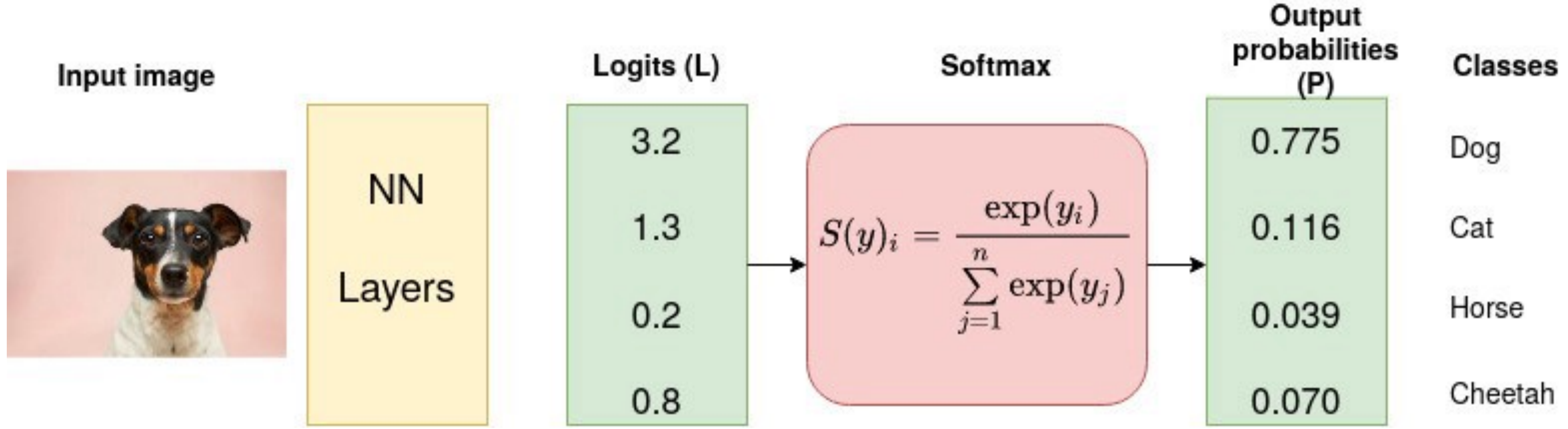
Maxout

$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



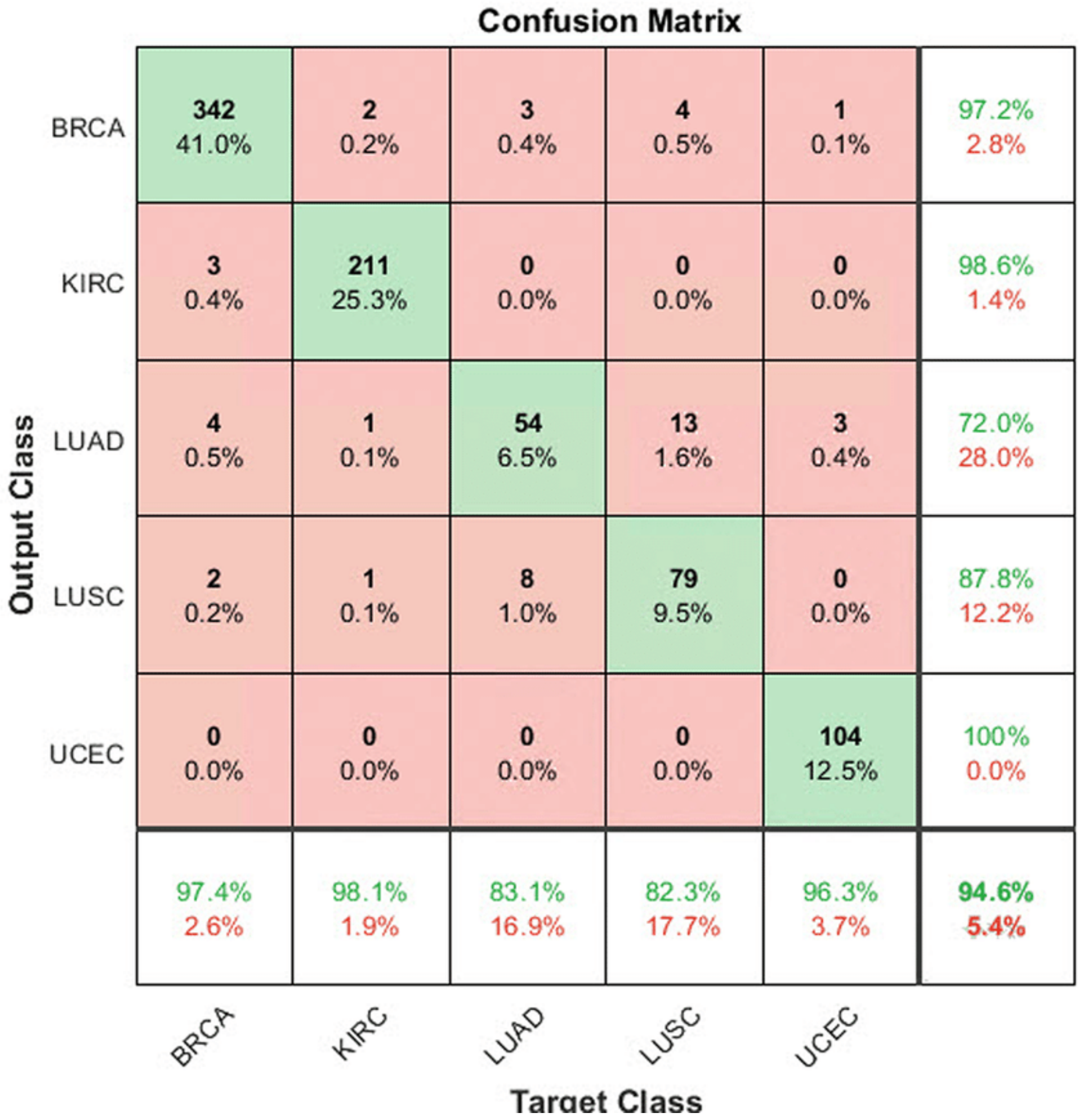


The **softmax function,** also known as **softargmax** or **normalized exponential function**, is a generalization of the logistic function to multiple dimensions. It is used in multinomial logistic regression and is often used as the last activation function of a neural network to normalize the output of a network to a probability distribution over predicted output classes

Confusion Matrix

		Predicted Class		
		Positive	**Negative**	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Multi-Class Confusion Matrix



Confusion Matrix

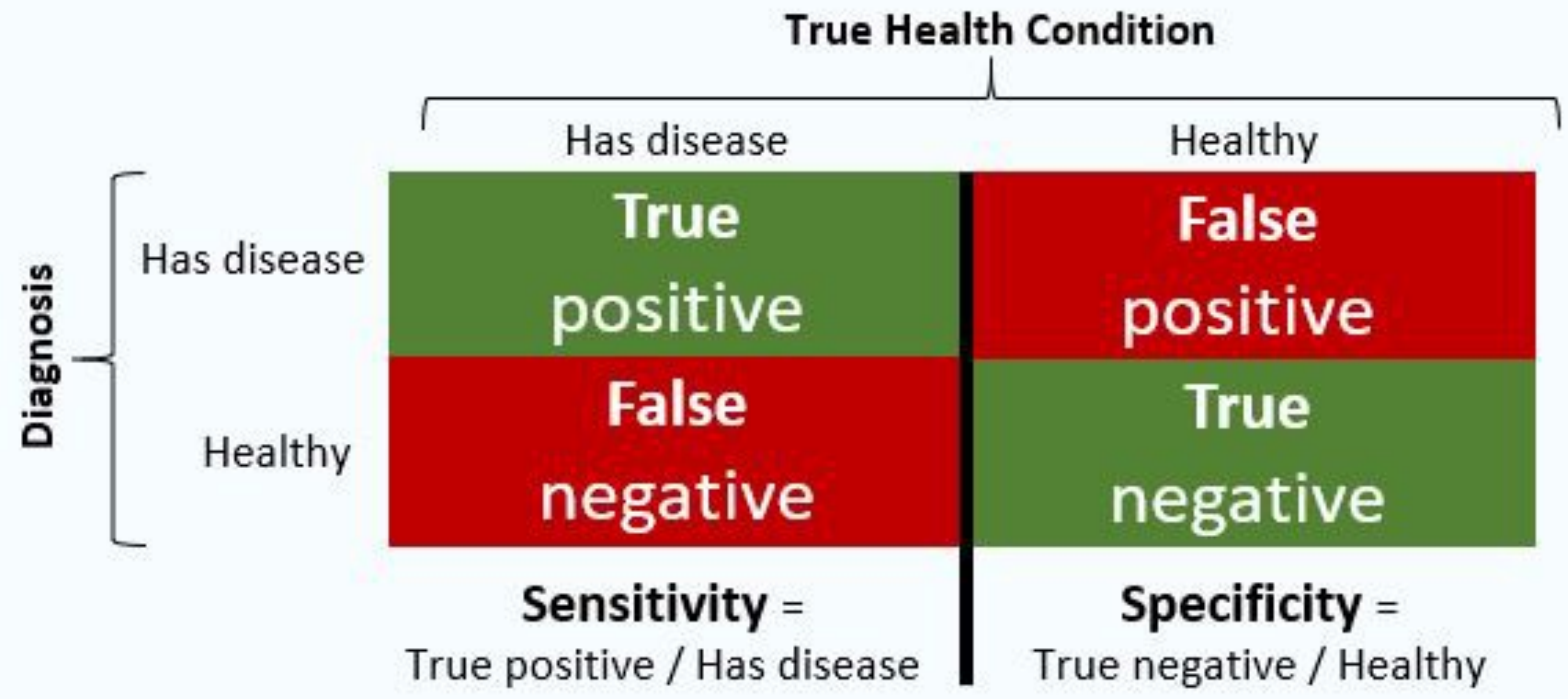
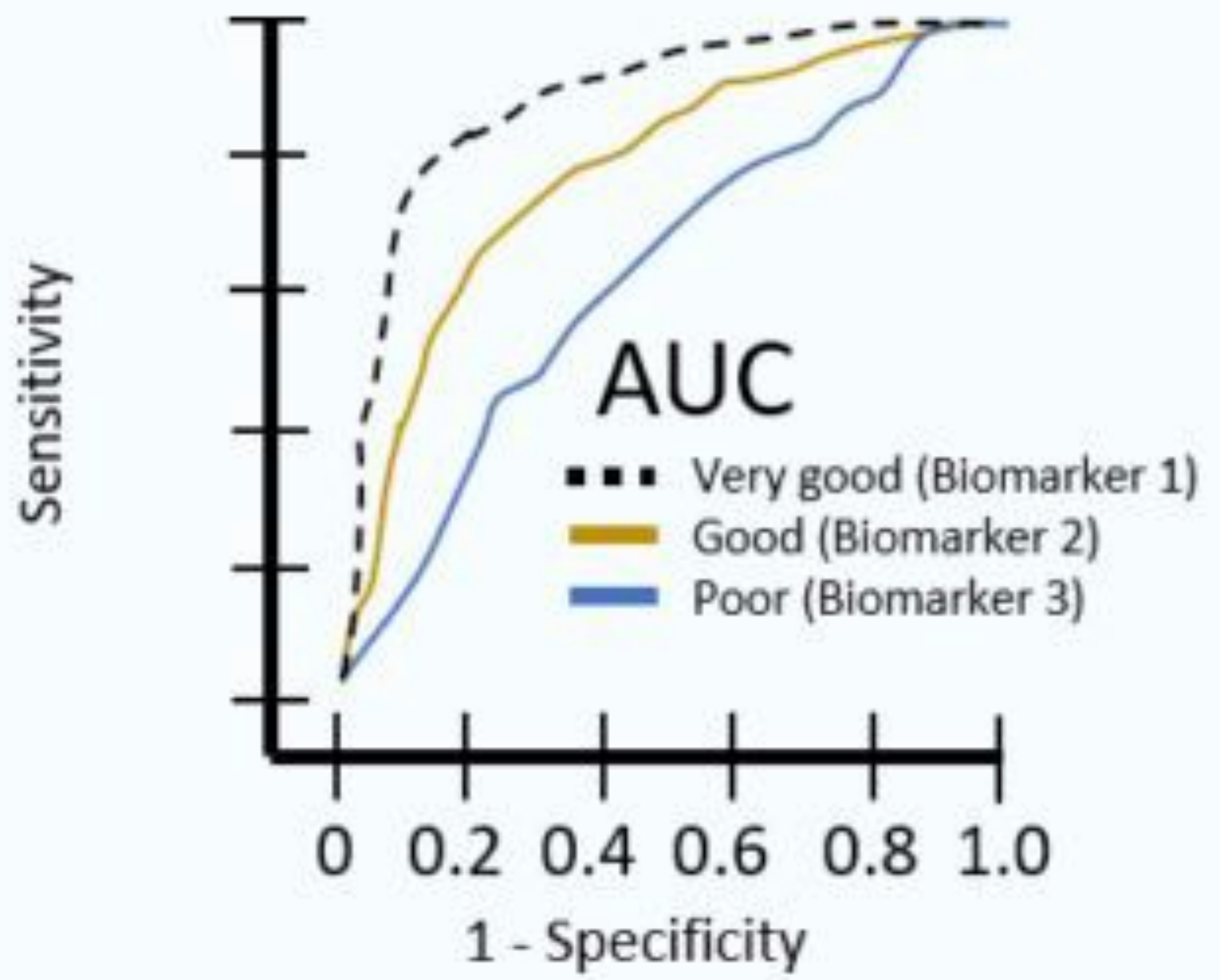


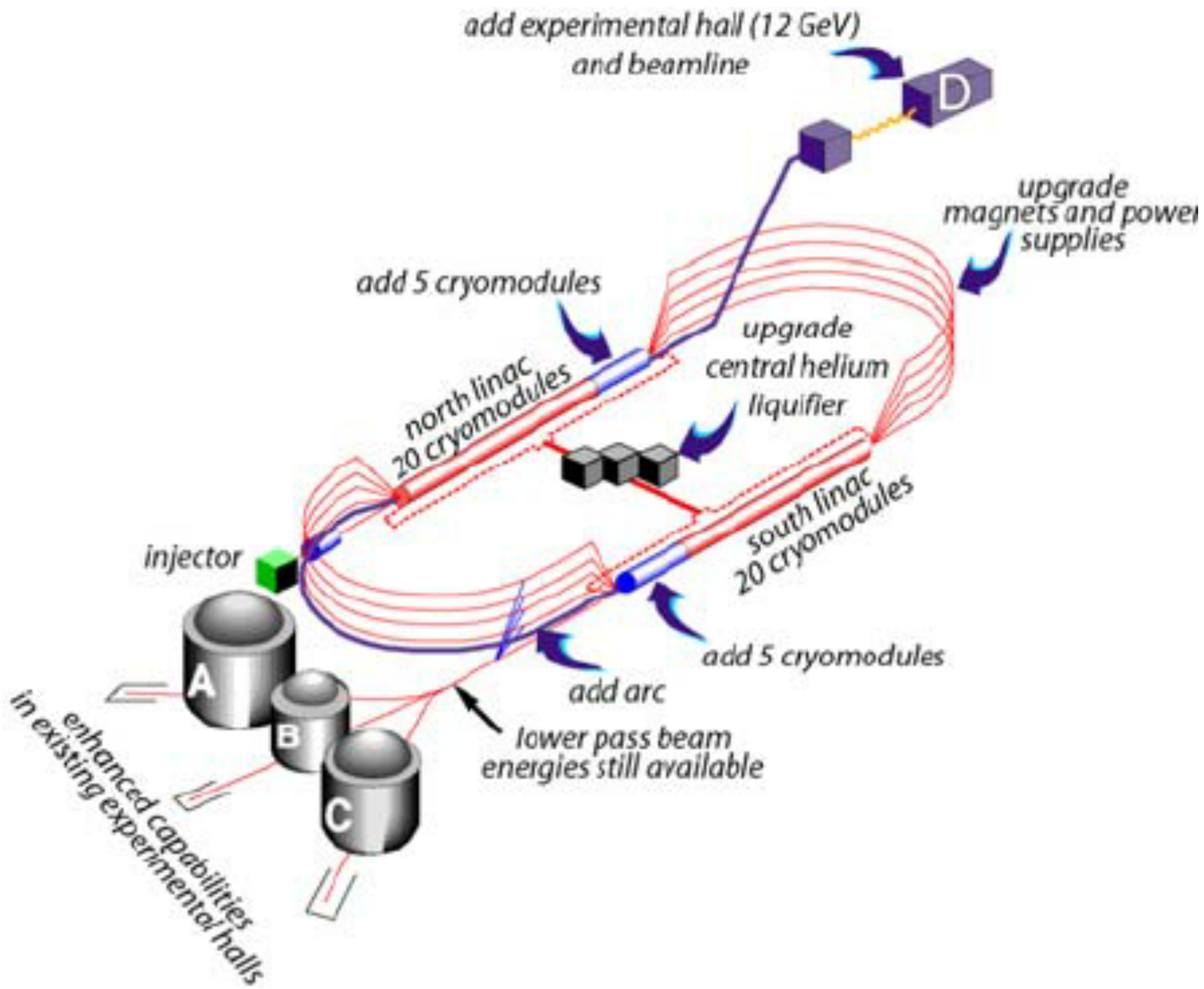
Figure 2. Calculation of senstivity and specificty.



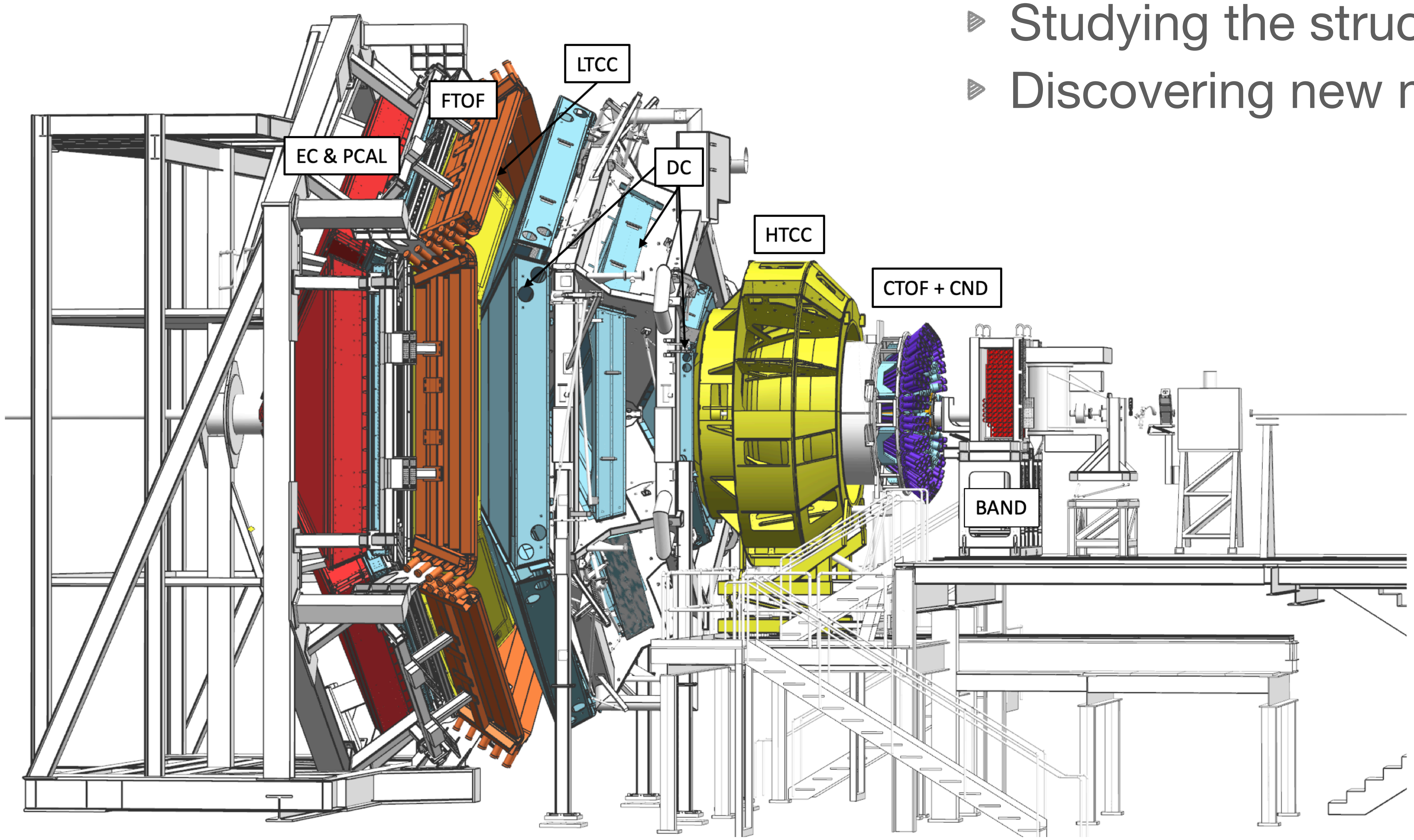
The **Receiver Operator Characteristic (ROC)** curve is an evaluation metric for binary classification problems. It is a probability curve that plots the **TPR** against **FPR** at various threshold values and essentially **separates the 'signal' from the 'noise'**. The **Area Under the Curve (AUC)** is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

AUC-ROC curve is a performance measurement for the ***classification problems*** at ***various threshold*** settings.

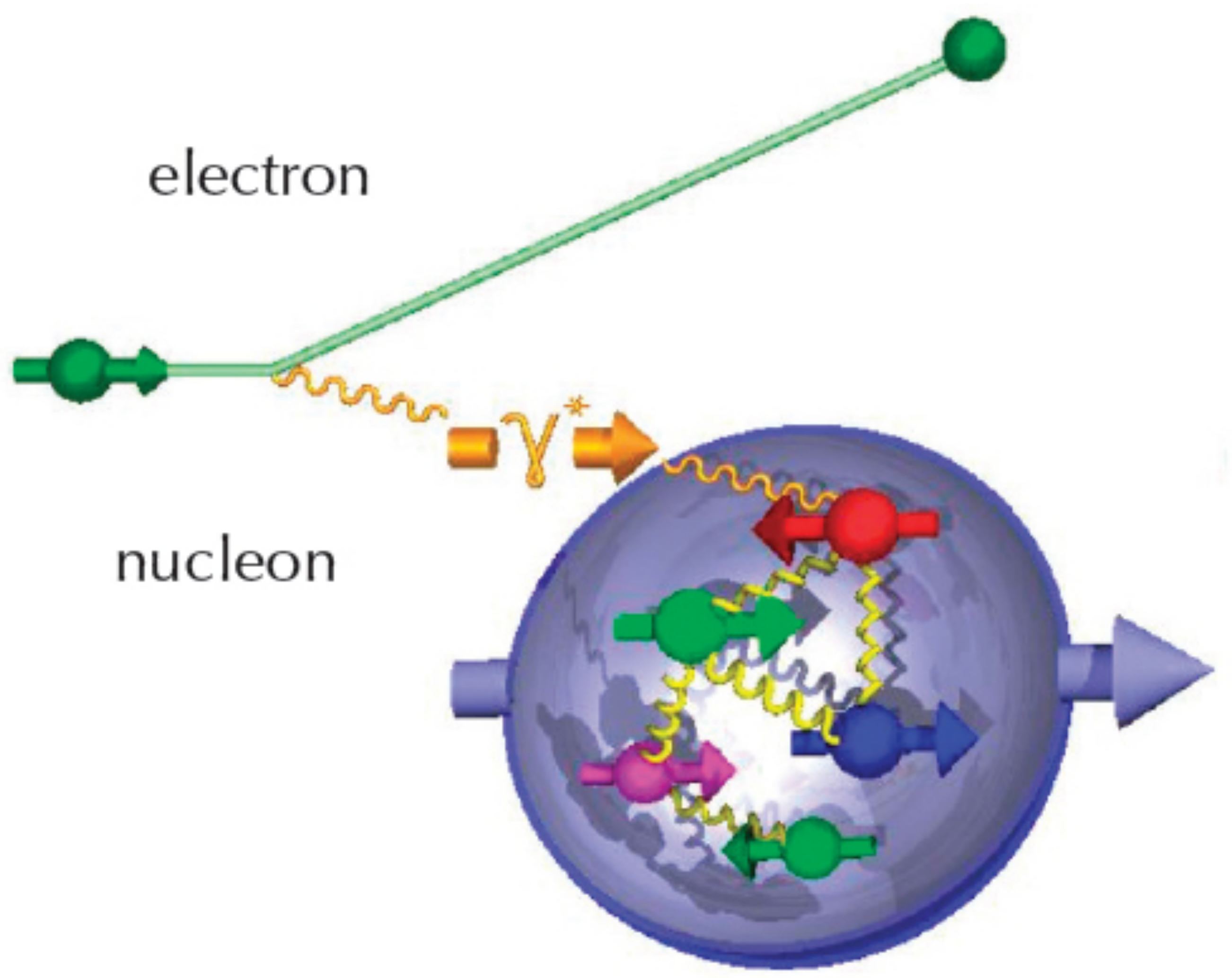
Jefferson Lab



- CEBAF
  - 12 GeV electron beam distributed to 4 experimental hall
  - Each experimental hall contains a detector system for specific experiments

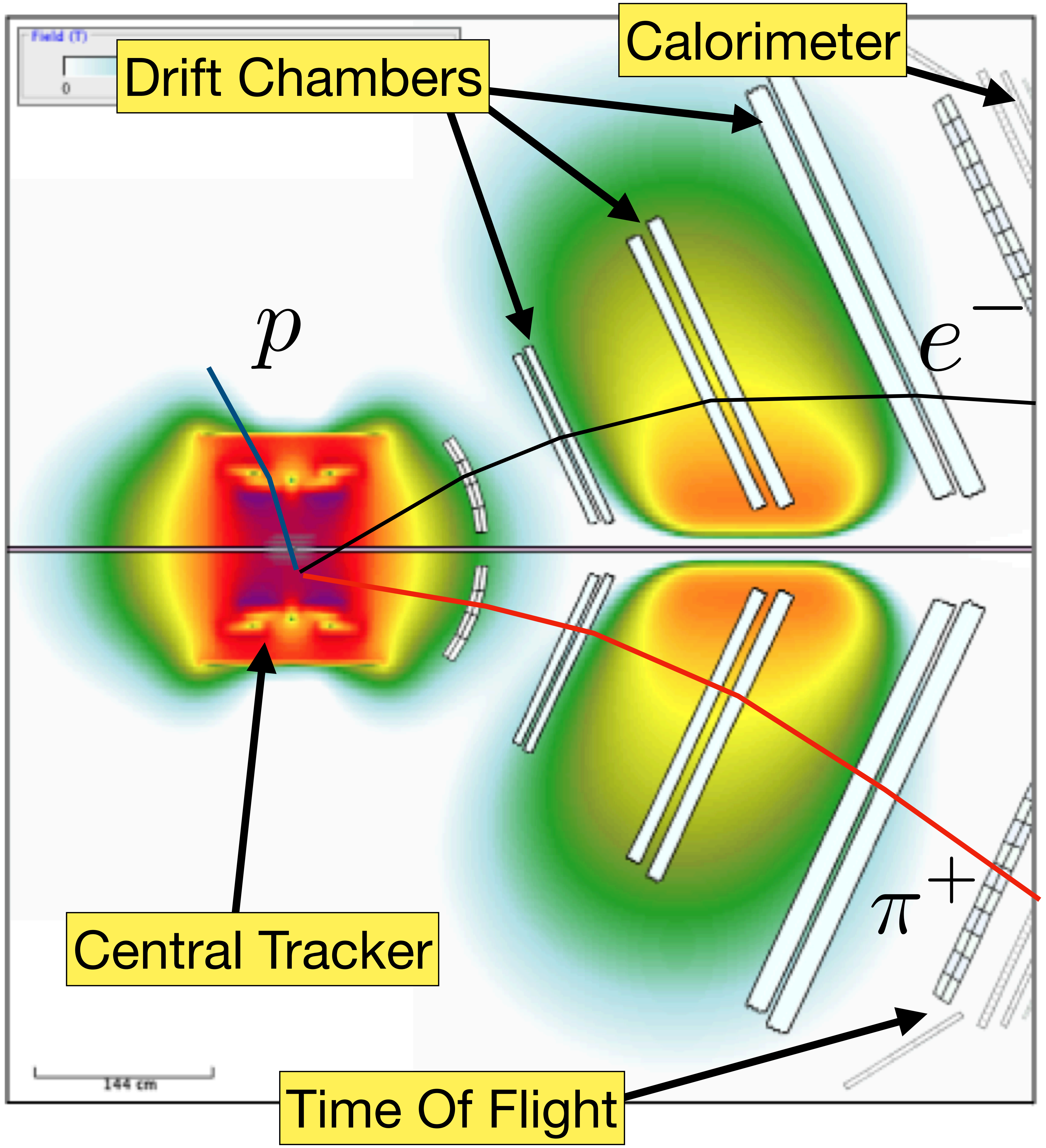


- Nuclear Experiments
  - Electron beam scattering off of proton
  - Charged and neutral particles detected by CLAS12 detector
  - Studying the structure of the nucleon
  - Discovering new matter states



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.



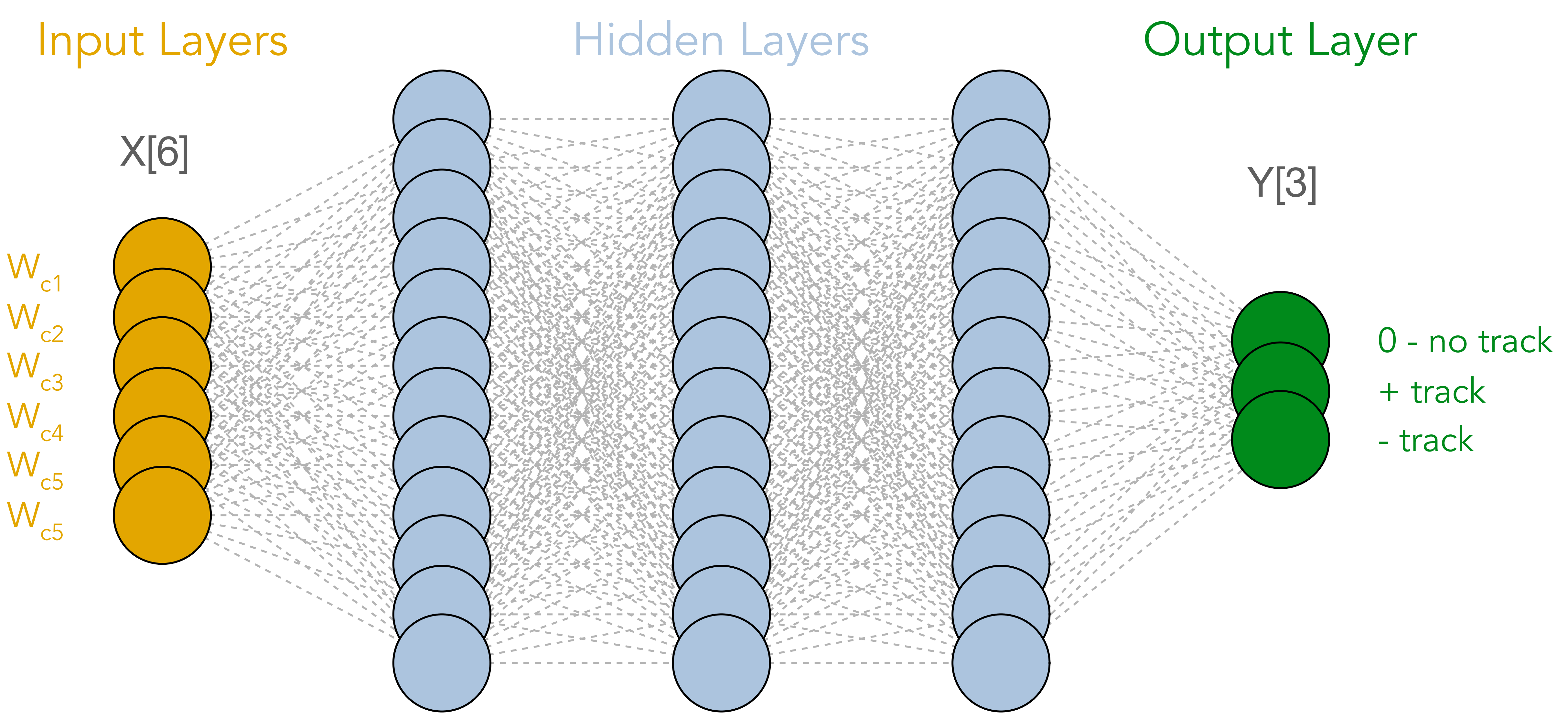
Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron is a network with an input vector X(x1,x2,…..,xn) and output vector Y(y1,y2,…ym)

It can have many hidden layers with any number of nodes

Connected with any activation functions
- RELU
- SIGMOID
- TANH
- LINEAR
- SOFTMAX

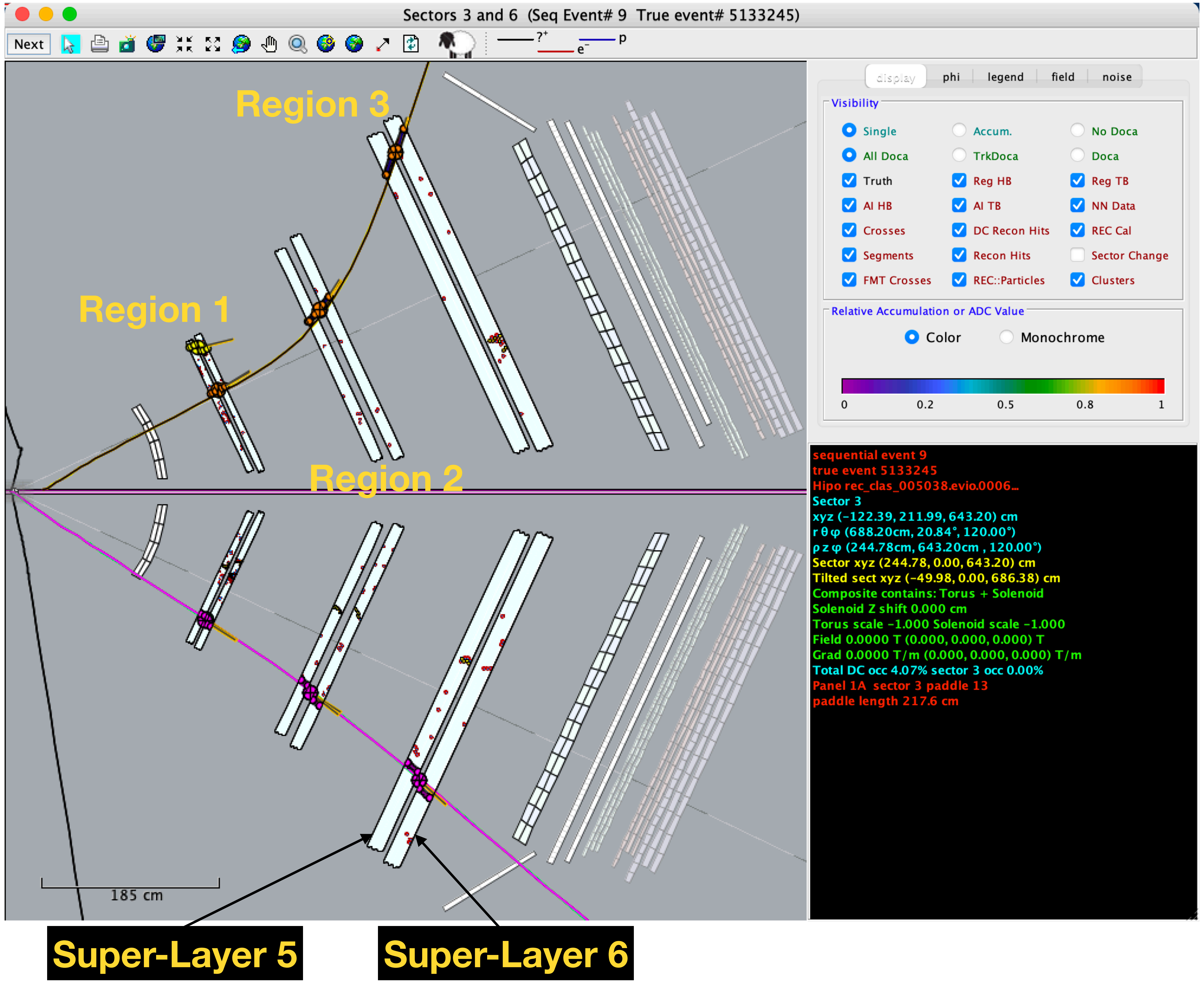
Defining any loss function:
- Mean Square Error (MSE)
- CrossEntropy
- ……



- Use cases
  - Classification
  - Linear Regression
  - Logistic Regression
  - Fault Correction Auto-encoders (in later slides)

Classifier
(MLP)

CLAS12 tracking

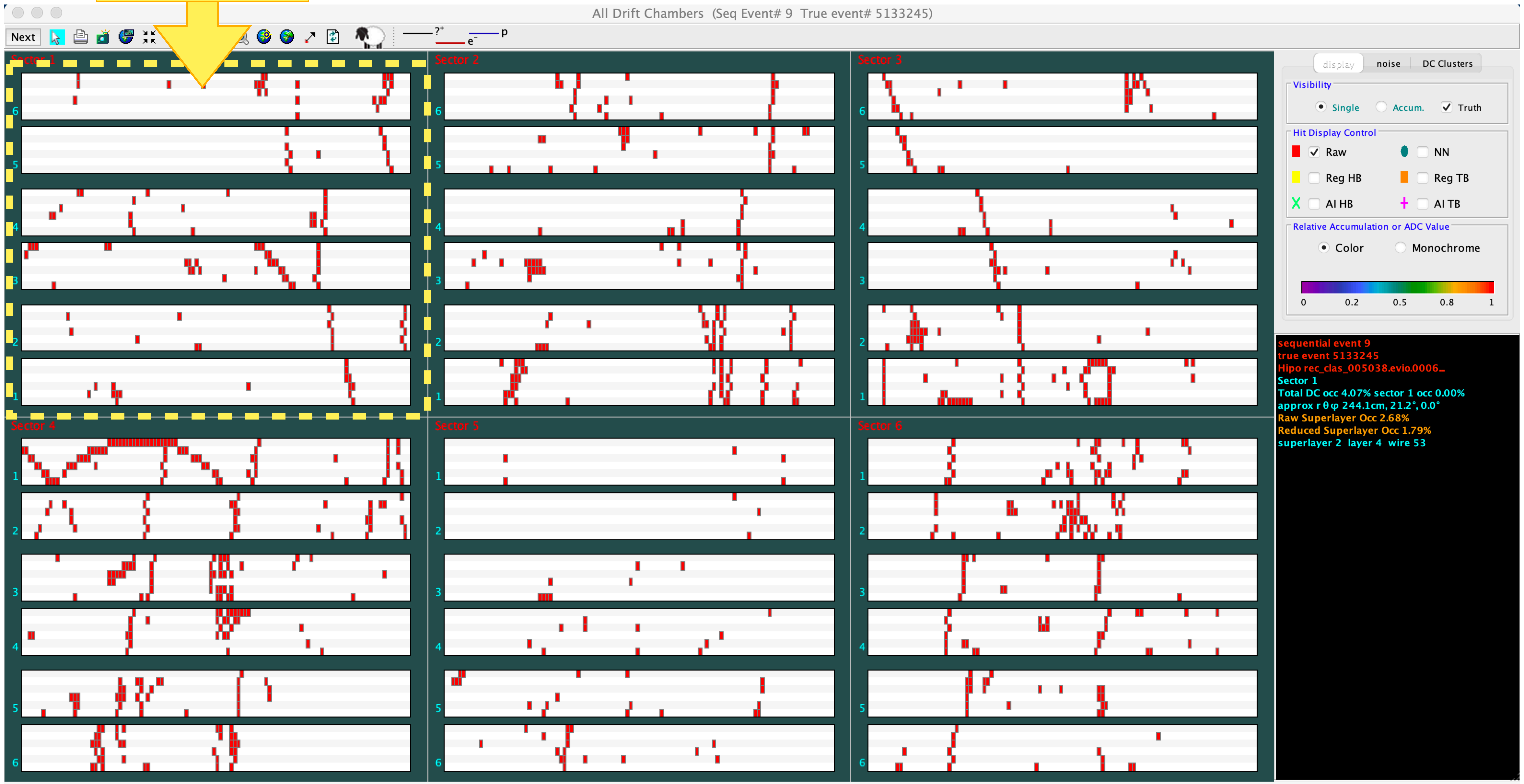


Charged Particle Tracking

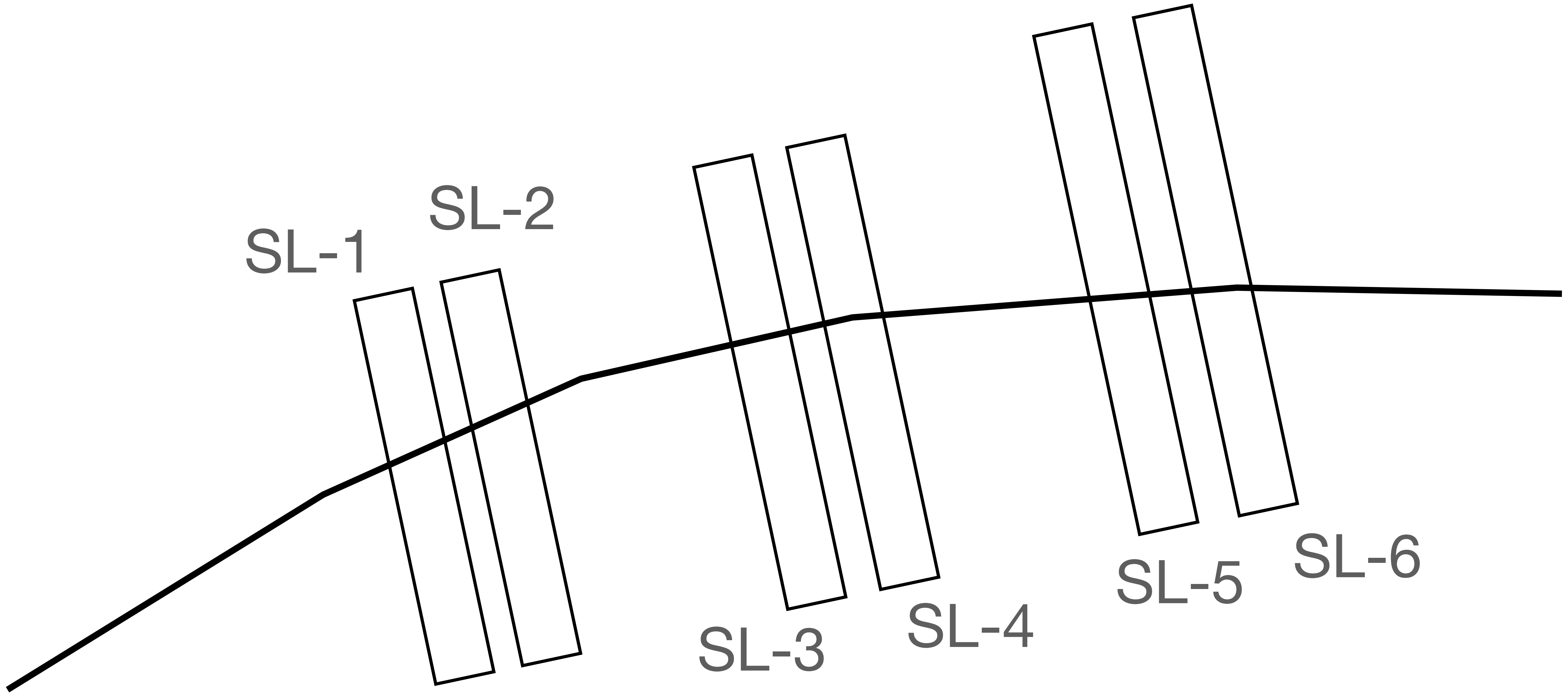
- Charged particles are tracked using Drift Chambers inside toroidal magnetic field.
  - Each sector consists of 3 regions
  - Each region consists of two cambers (Super-Layer)
  - Super-Layer has 6 layers
  - Each Layer has 112 wires
- Each sector is matrix of 36x112 wires that charged particles passes
- Each super layer hits are clustered together
- Track candidate is format from 6 clusters (one from each super layer)

Sector 1

Six sectors shown



CLAS12 tracking



$$X_6 = \sum_{i=31}^{36} \frac{w_i}{6}$$

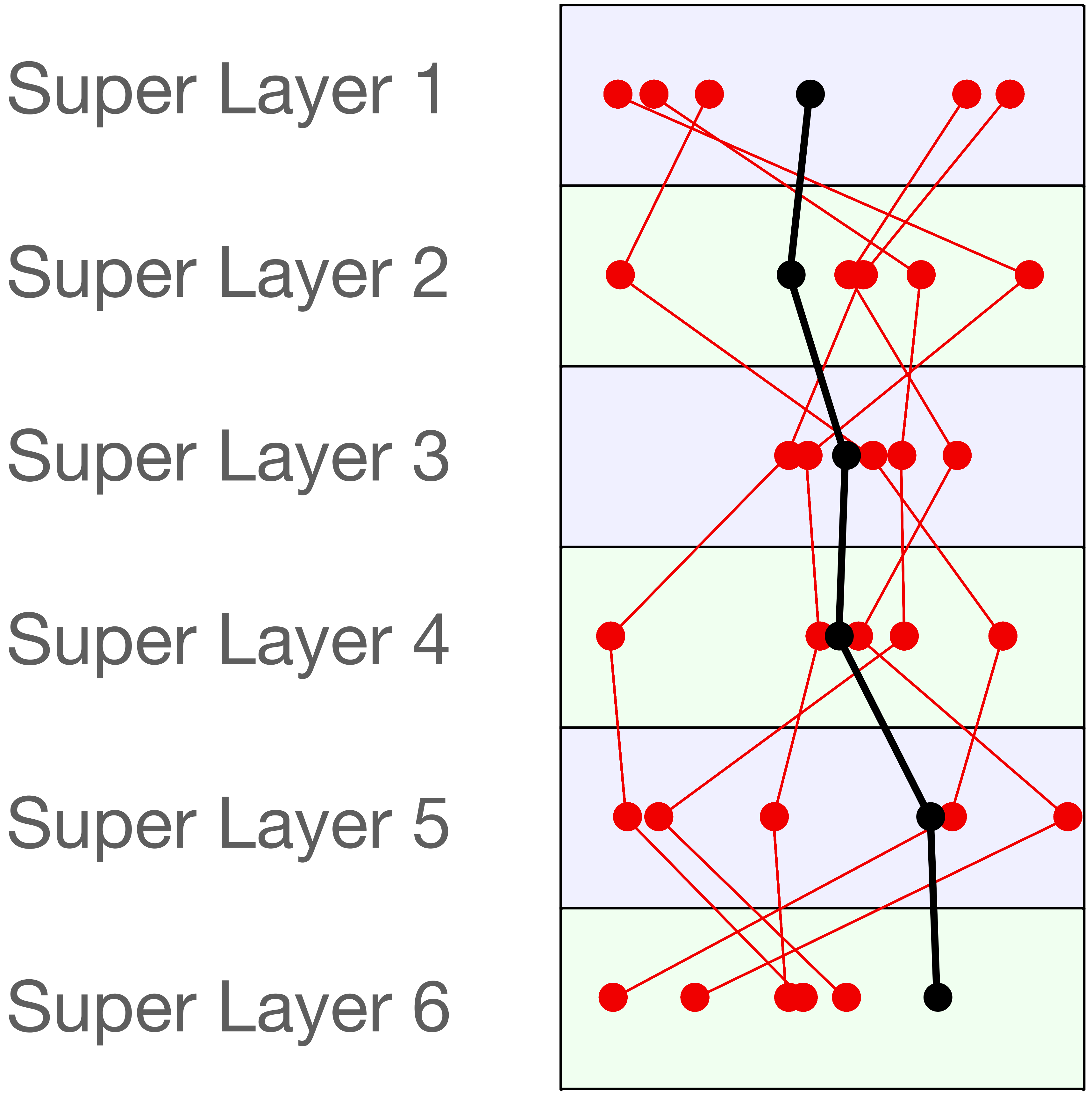
$$X_5 = \sum_{i=25}^{30} \frac{w_i}{6}$$

$$X_4 = \sum_{i=19}^{24} \frac{w_i}{6}$$

$$X_3 = \sum_{i=13}^{18} \frac{w_i}{6}$$

$$X_2 = \sum_{i=7}^{12} \frac{w_i}{6}$$

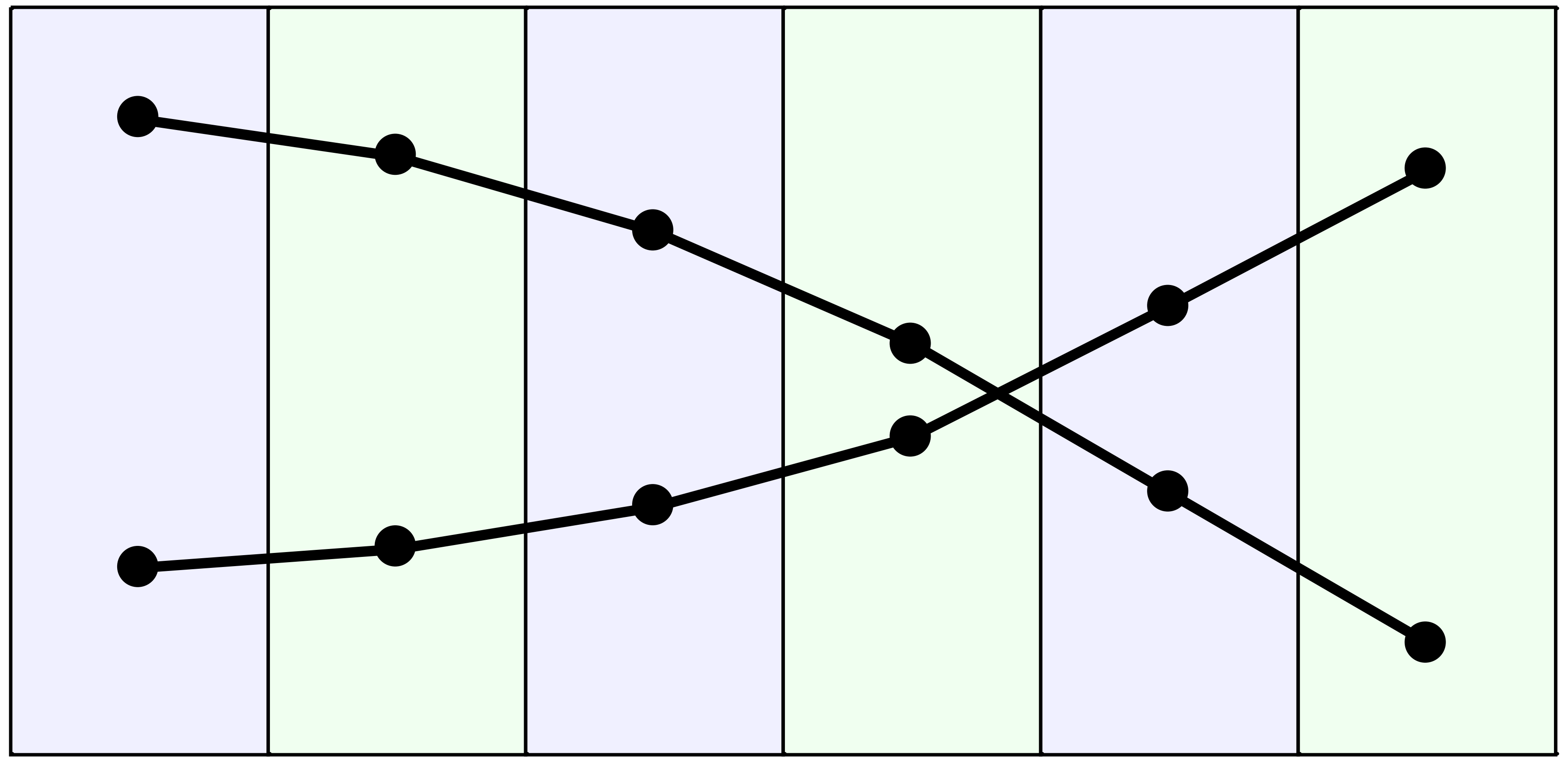
$$X_1 = \sum_{i=1}^{6} \frac{w_i}{6}$$

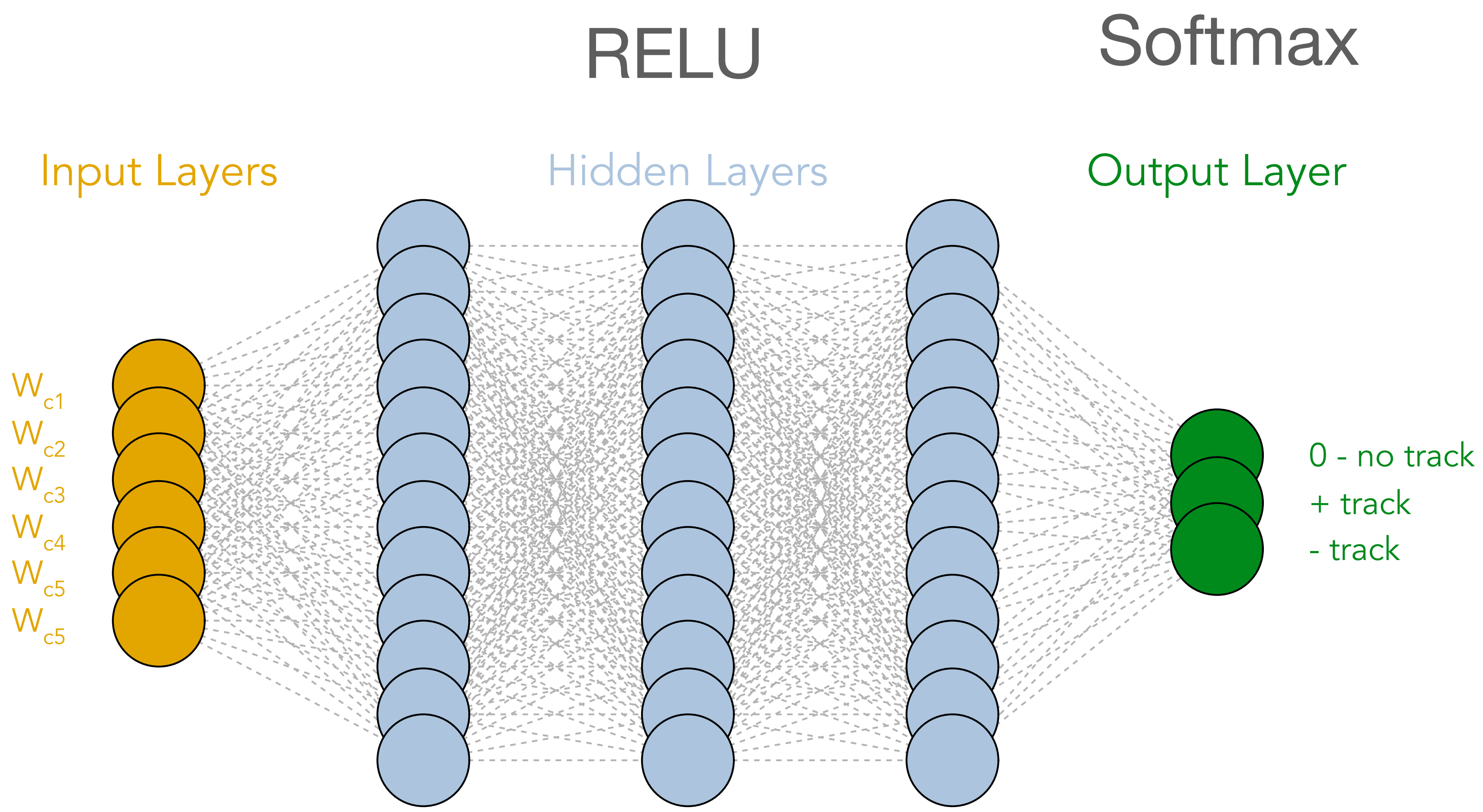


Classification

- Each event contains many combinations of clusters that can form a track.
- Teaching AI which combinations are good and which are bad will help the network to discern from given combinatorics which candidate has a higher probability to be a good track.
- Possibly will speed up tracking code (80%-90% of total data processing time) by considering only AI suggested track candidates.

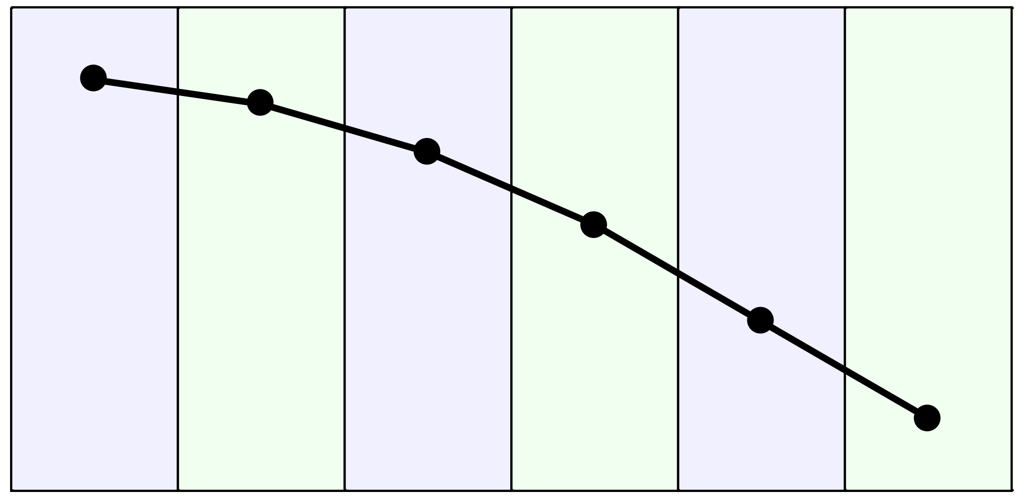
Track Data



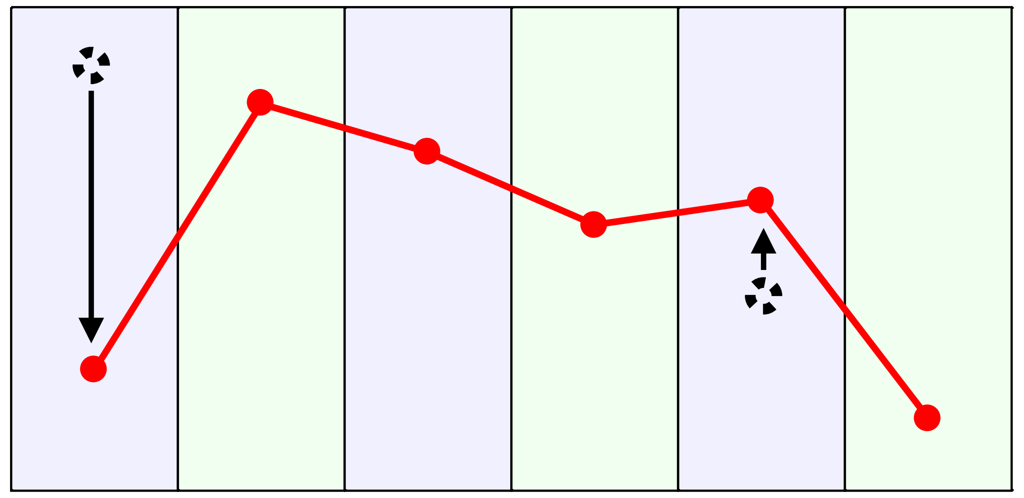


- Events with 2 tracks in one sector are chosen for training sample generation.
- 4 training track candidates are constructed:
  - 2 "**TRUE**" tracks that were reconstructed by tracking algorithm
  - 2 "**FALSE**" tracks by swapping 1 or 2 (decided by random number generator) clusters from adjacent track.

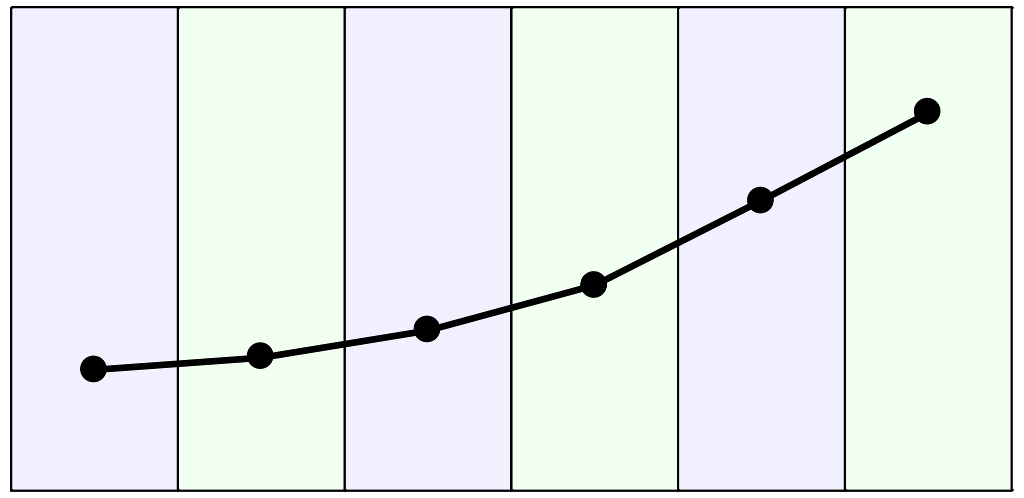
TRUE TRACK



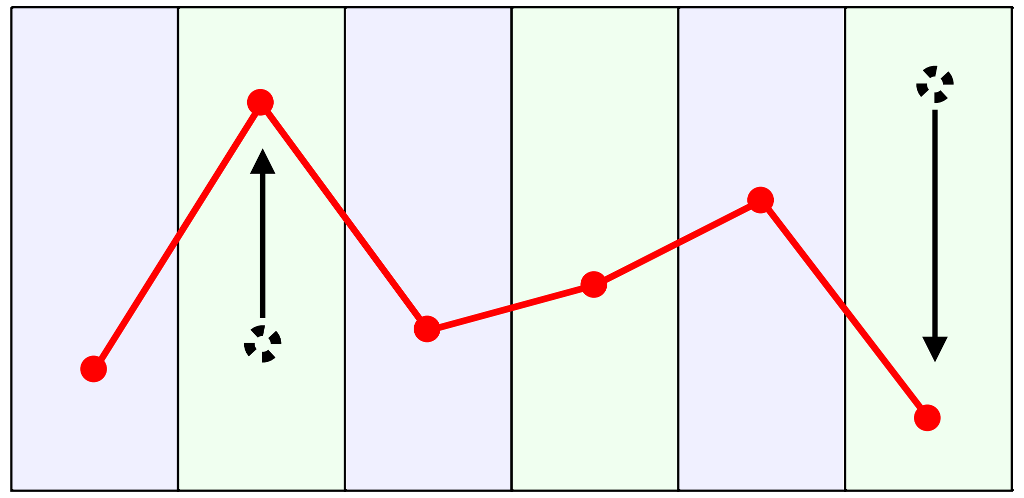
FALSE TRACK



TRUE TRACK

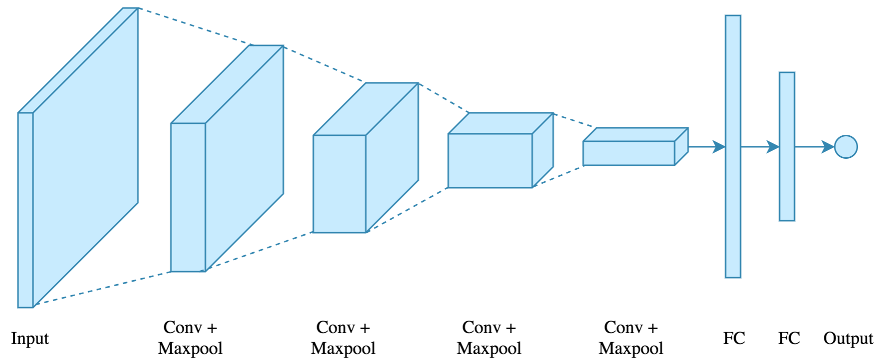


FALSE TRACK

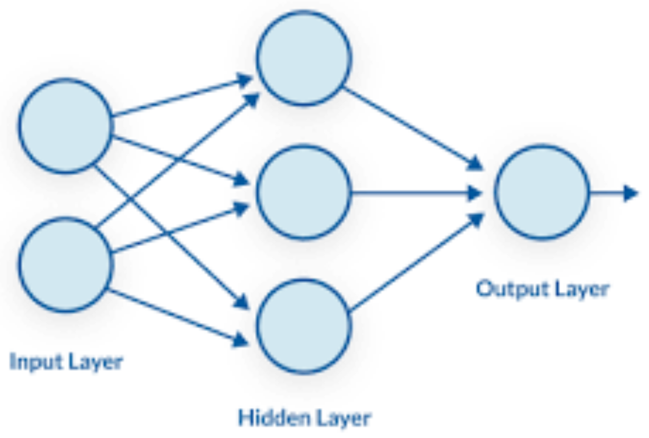


Neural Networks

Convolutional Neural Network (**CNN**)

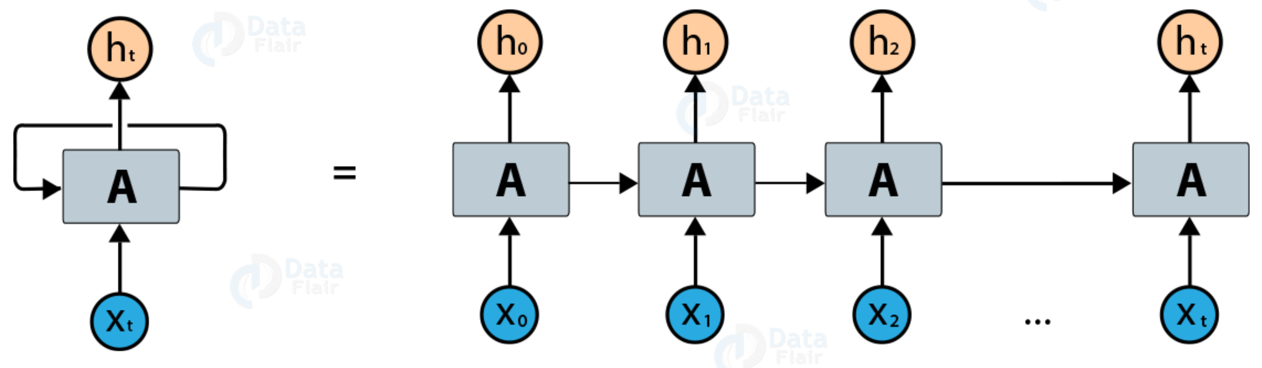


Multi-Layer Perceptron (**MLP**)



Extremely Randomized Trees (**ERT**)

Recurrent Neural Network (**RNN**)



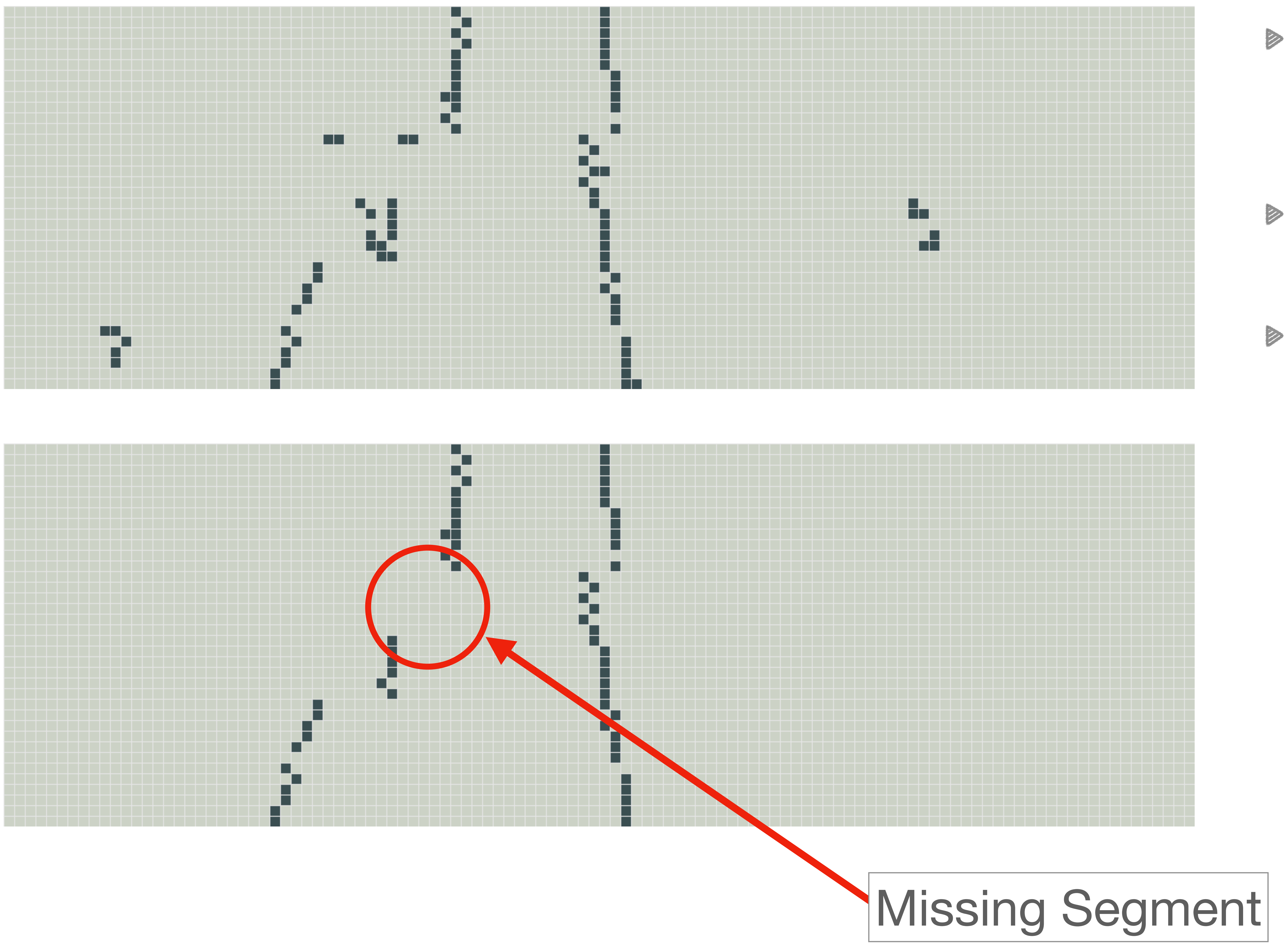
- Different Network types were evaluated for accuracy and speed.
- MLP is chosen to be the best fit, due to implementation simplicity, accuracy and inference speed.

	Features	TP	FP	PA	TA	Time (ms)
ERT	6	100%	6.14%	100%	100%	0.36
MLP	**6**	**99.96%**	**10.77%**	**98.88%**	**99.65%**	**0.12**
CNN	36x112	96.11%	28.11%	94.26%	94.26%	1.2
RNN	36	88.40%	11.60%	-	-	-

TP - True Positive
FP - False Positive
TA - Training Accuracy
PA - Positive Accuracy : percentage of tracks where False Positive in an event has lower probability than True Positive

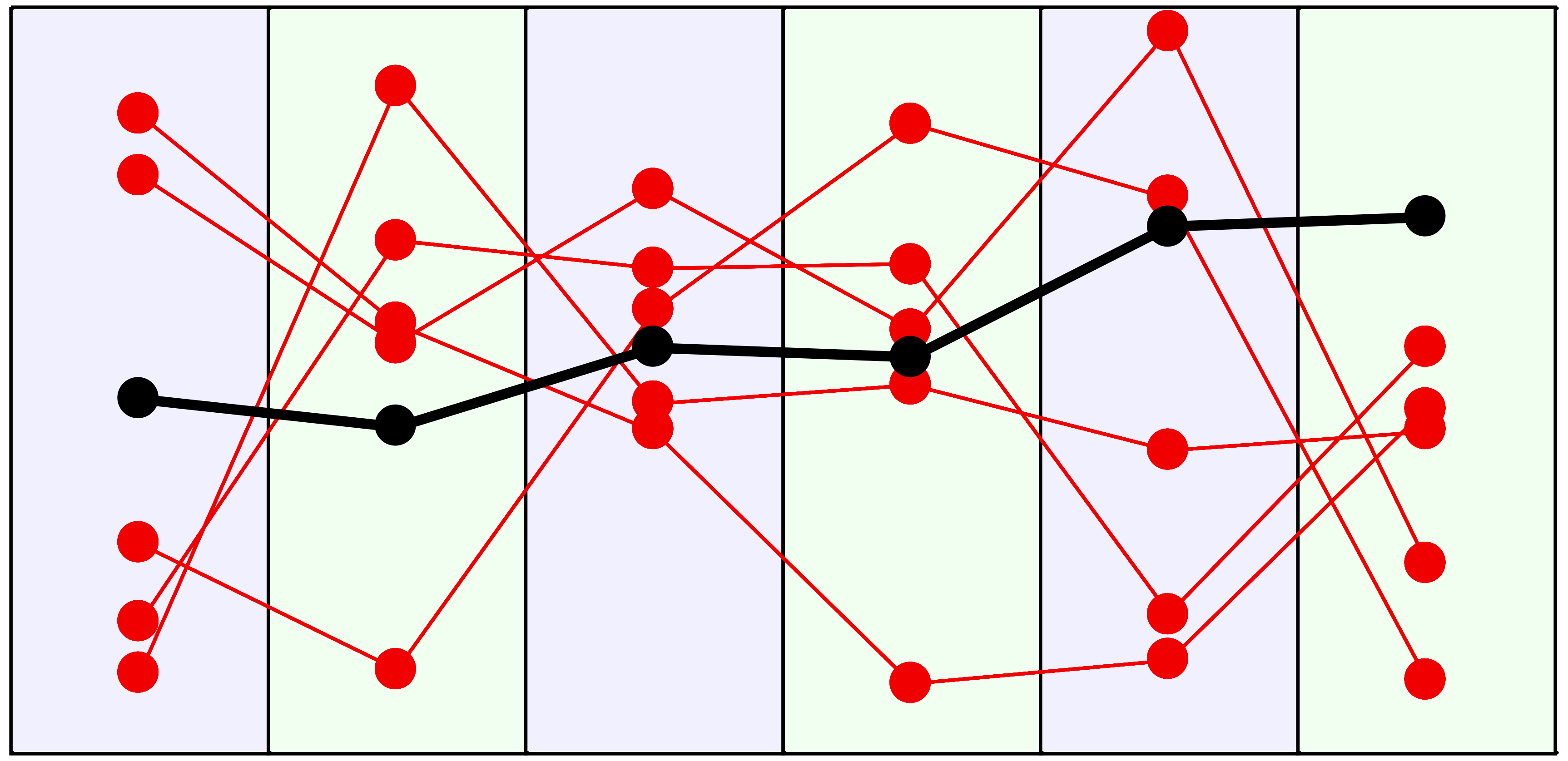
Corruption Auto-Encoder (MLP)

Problems we are trying to solve



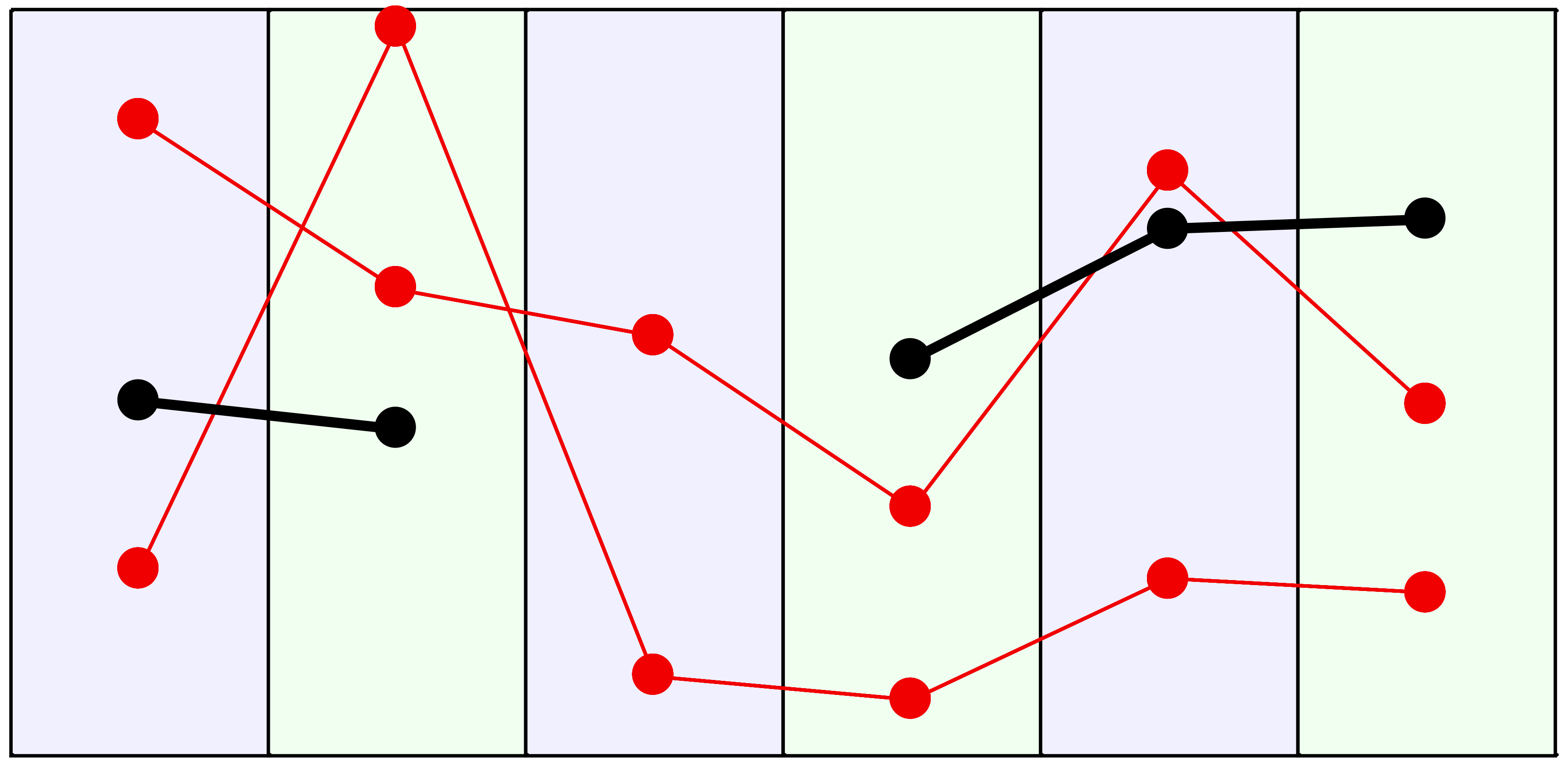
- Inefficiency in the Drift Chambers can result in missing segment along the track trajectory
- It is possible to fit the track using only 5 segments
- Need to identify the tracks with 5 segments (using AI)

Problems we are trying to solve



Classification

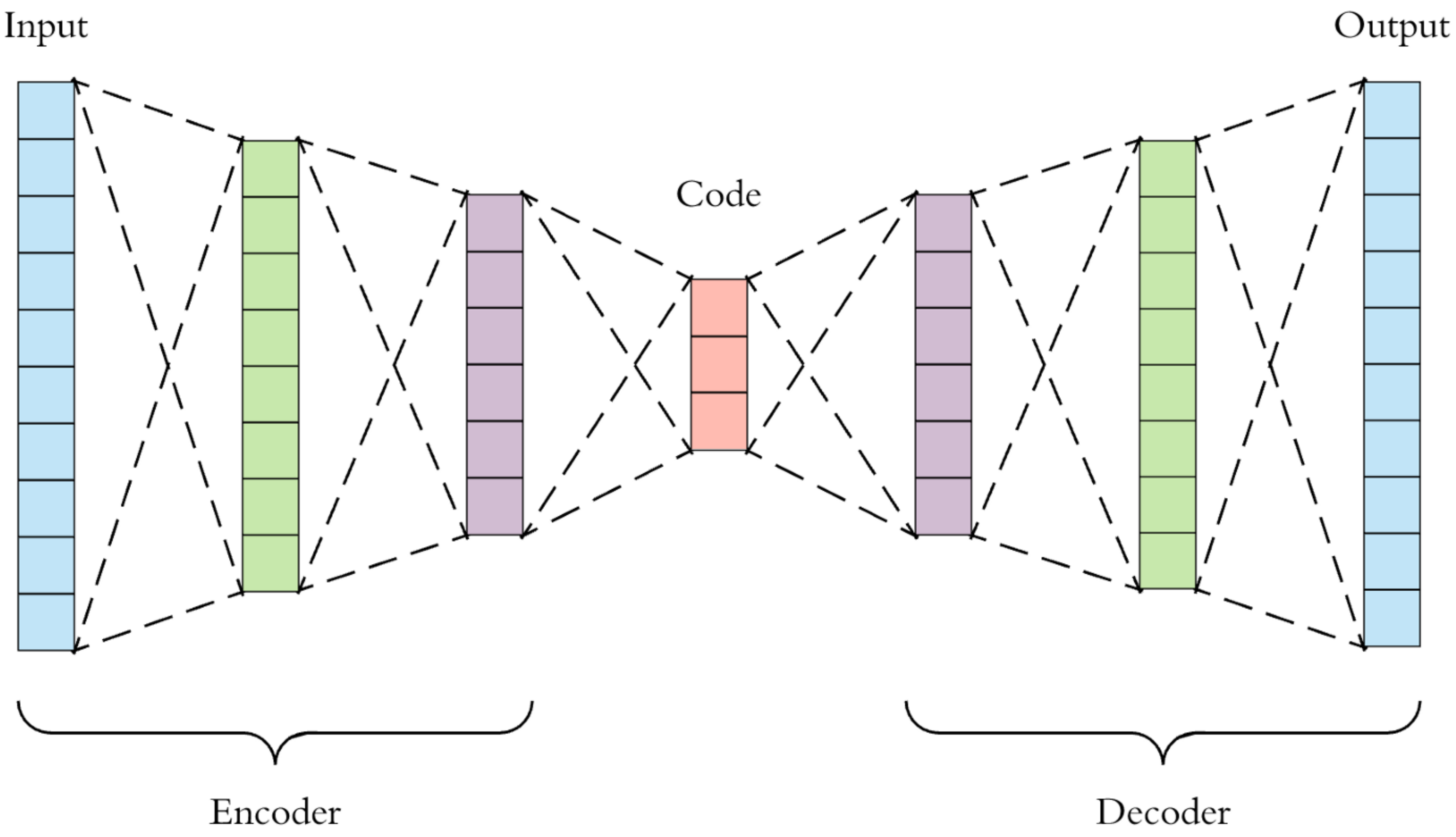
- Each event contains many combinations of clusters that can form a track.
- Teaching AI which combinations are good and which are bad will help the network to discern from given combinatorics which candidate has higher probability to be a good track.
- Possibly will speed up tracking code (80%-90% of total data processing time) by considering only AI suggested track candidates.

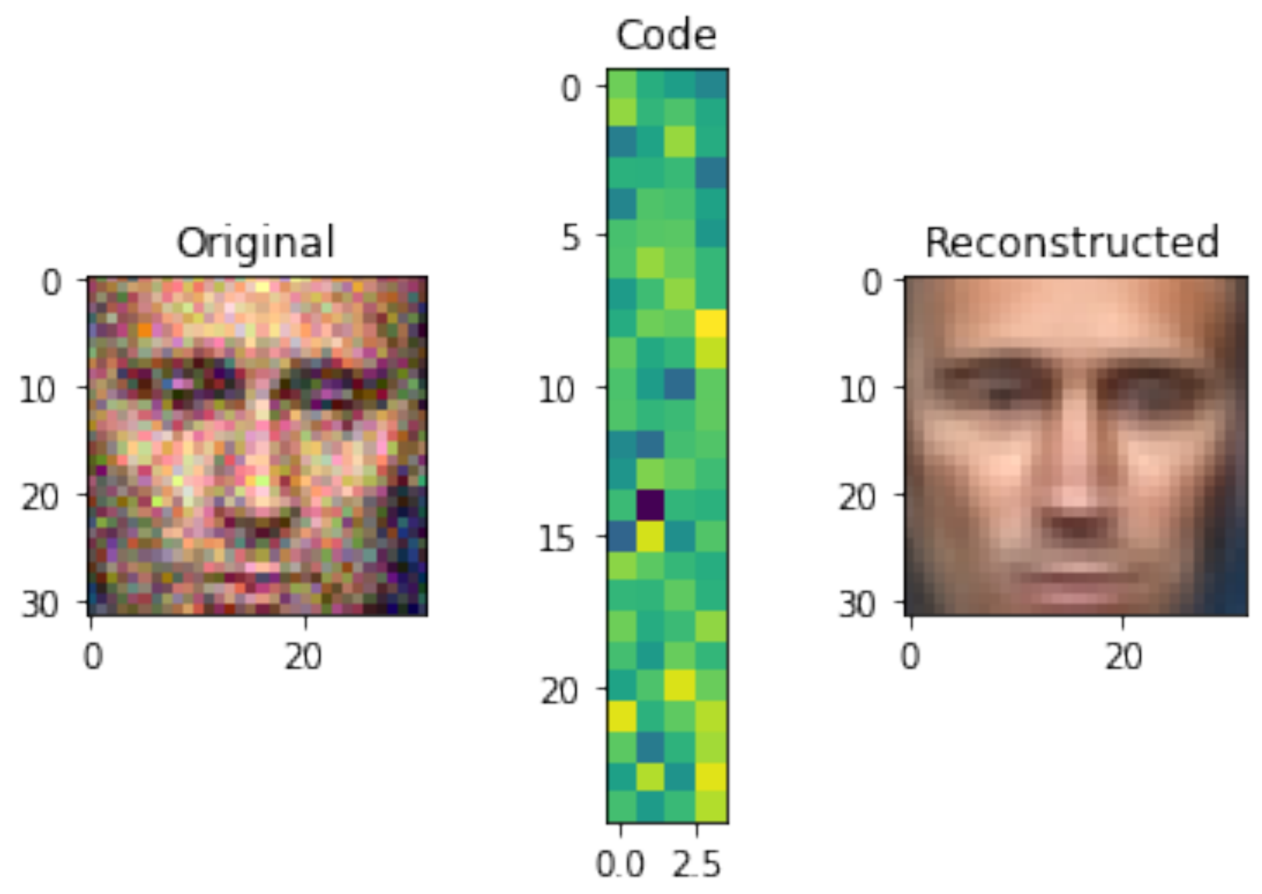


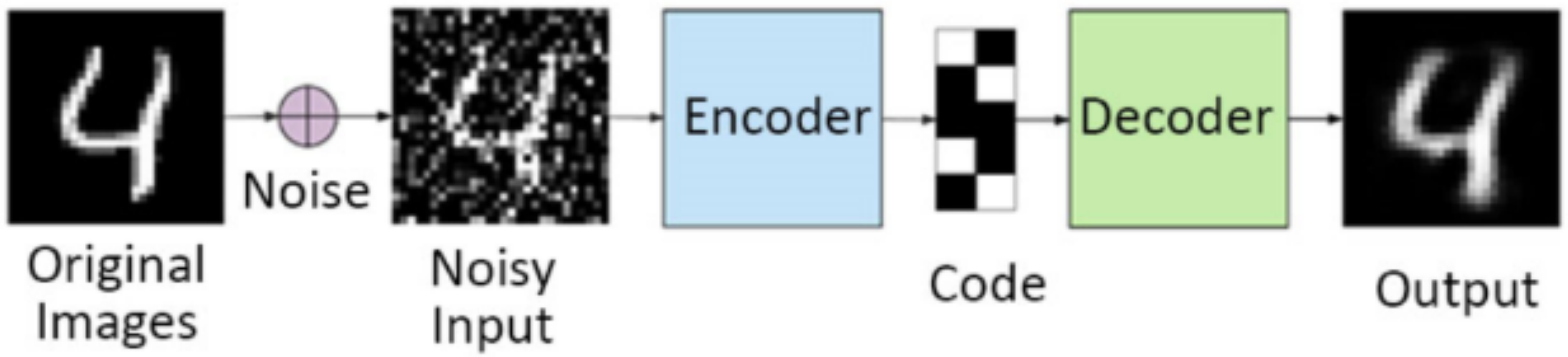
Fixing Inefficiencies

- Some regions of inefficiency in drift chambers can result in missing clusters in one of the super layers.
- Track classifier can recognize good tracks composed of 6 clusters.
- We need some methods to predict where missing cluster position will be.
- Then classifier can identify good track candidate.

- The Input has the same dimensions as the output
- Parts of the output are different from the input
- The most common uses are de-noising images







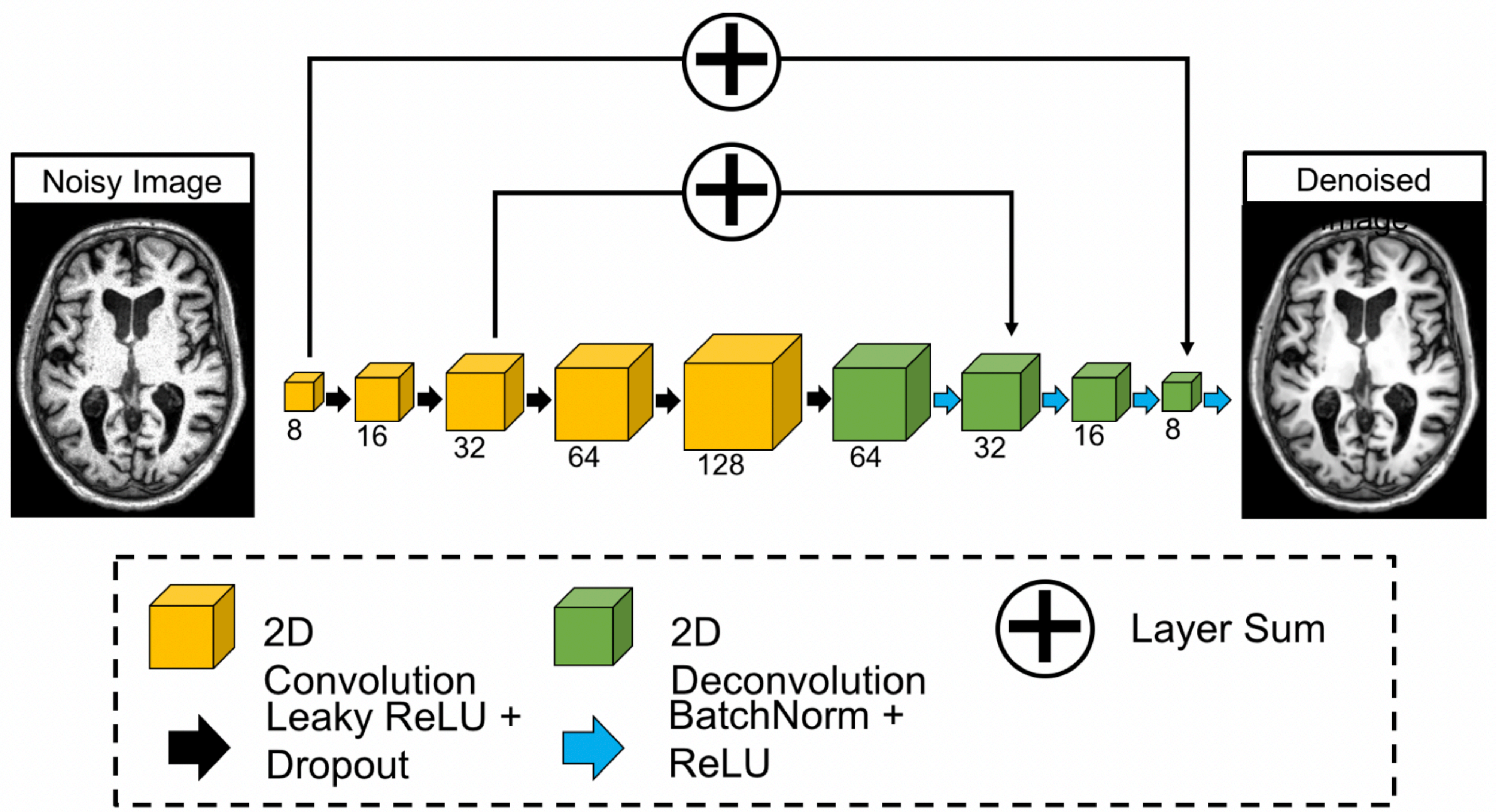
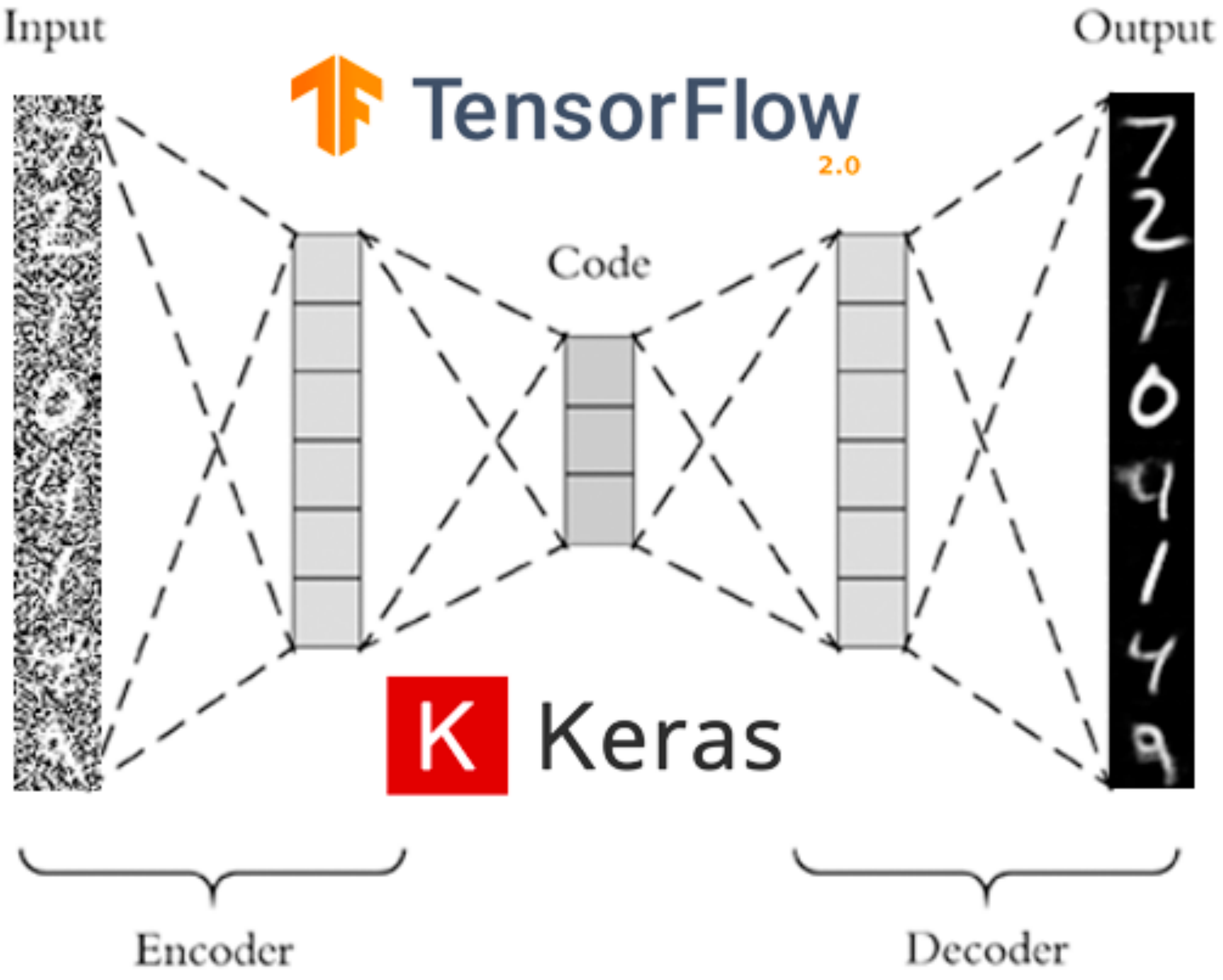


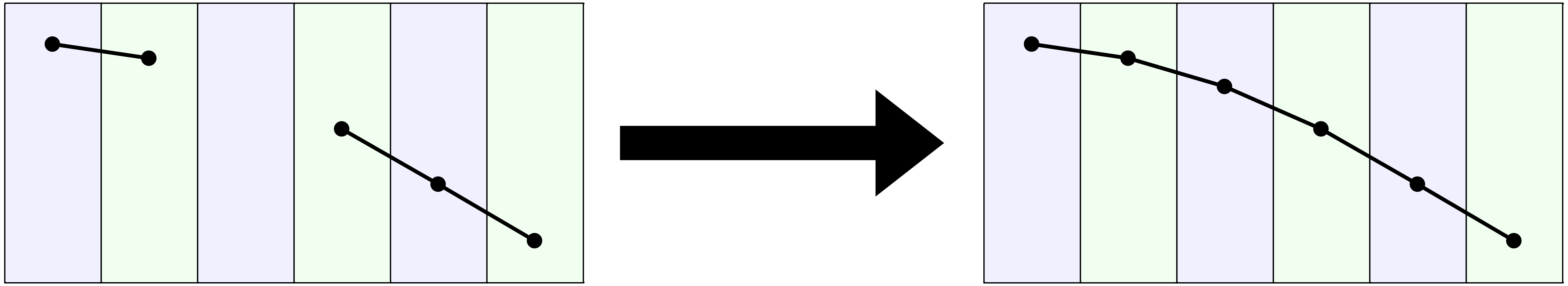
Figure 2. Denoising Autoencoder Network. This network consists of a convolutional neural network of increasing filter size, followed by a deconvolutional neural network of decreasing filter size. It takes a noisy image as the input and returns the denoised image.

Can This work backward?



Fixing inefficiencies

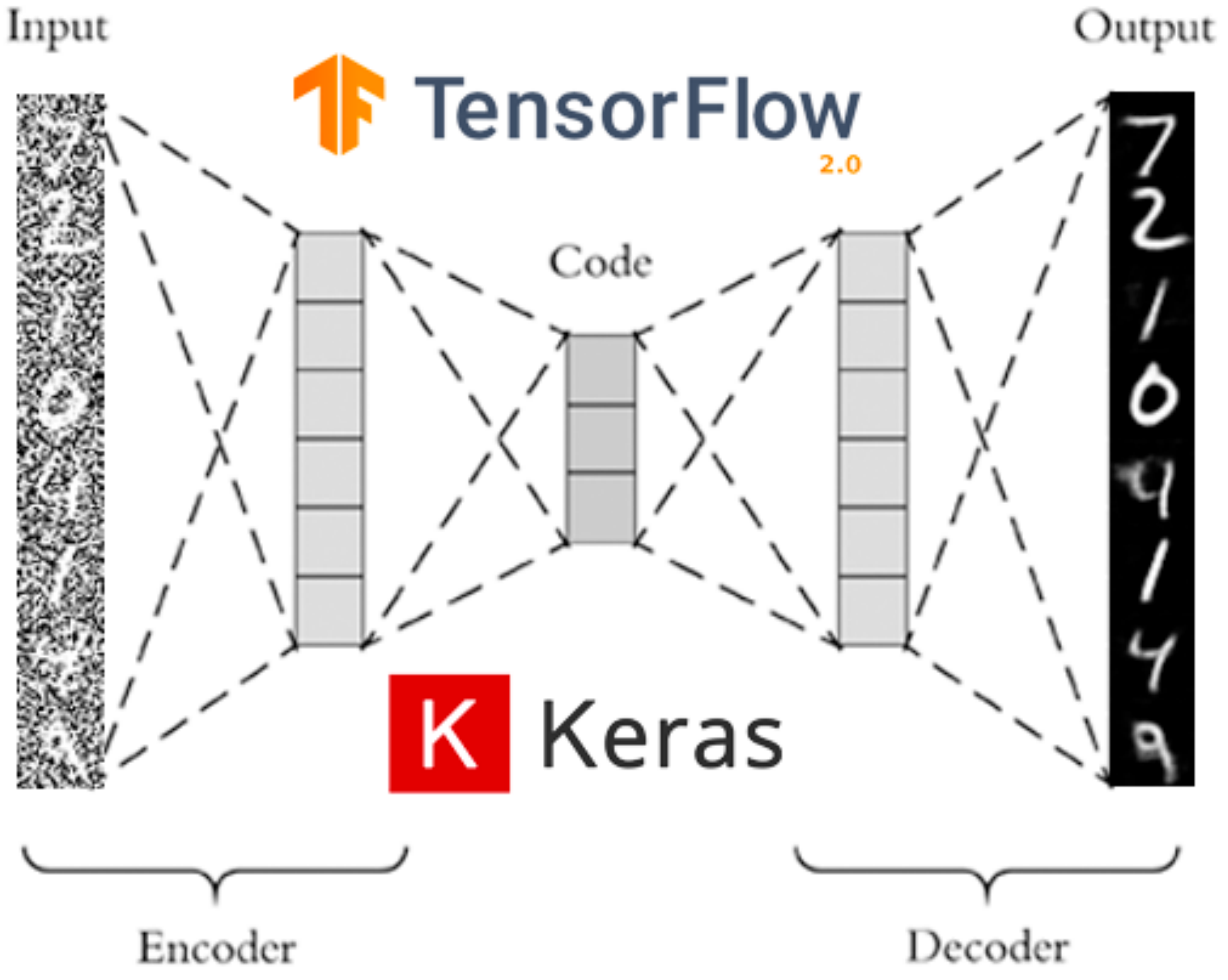
Noise remover removes part of the information
From original image

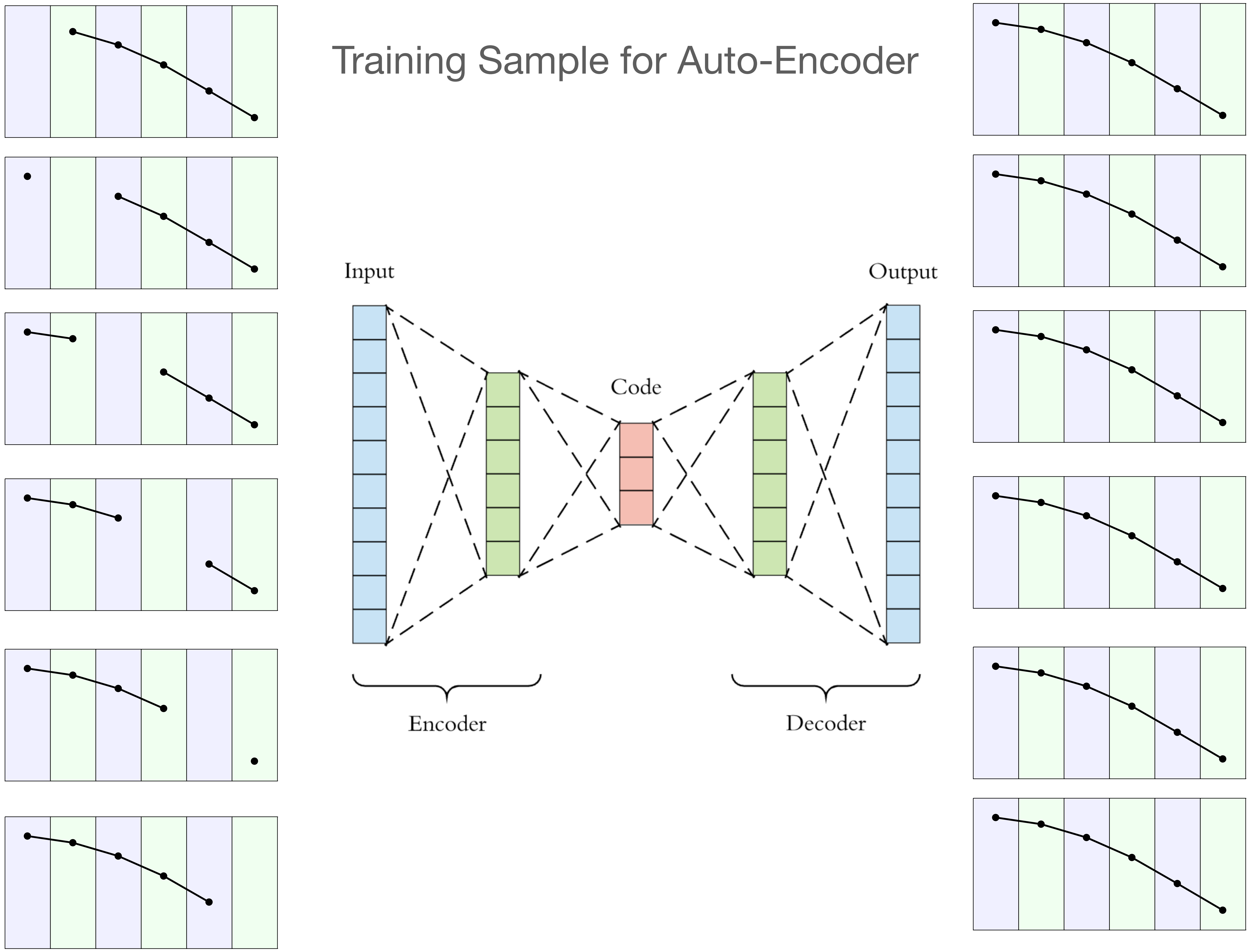


Can it add information in the right place?

Fixing inefficiencies

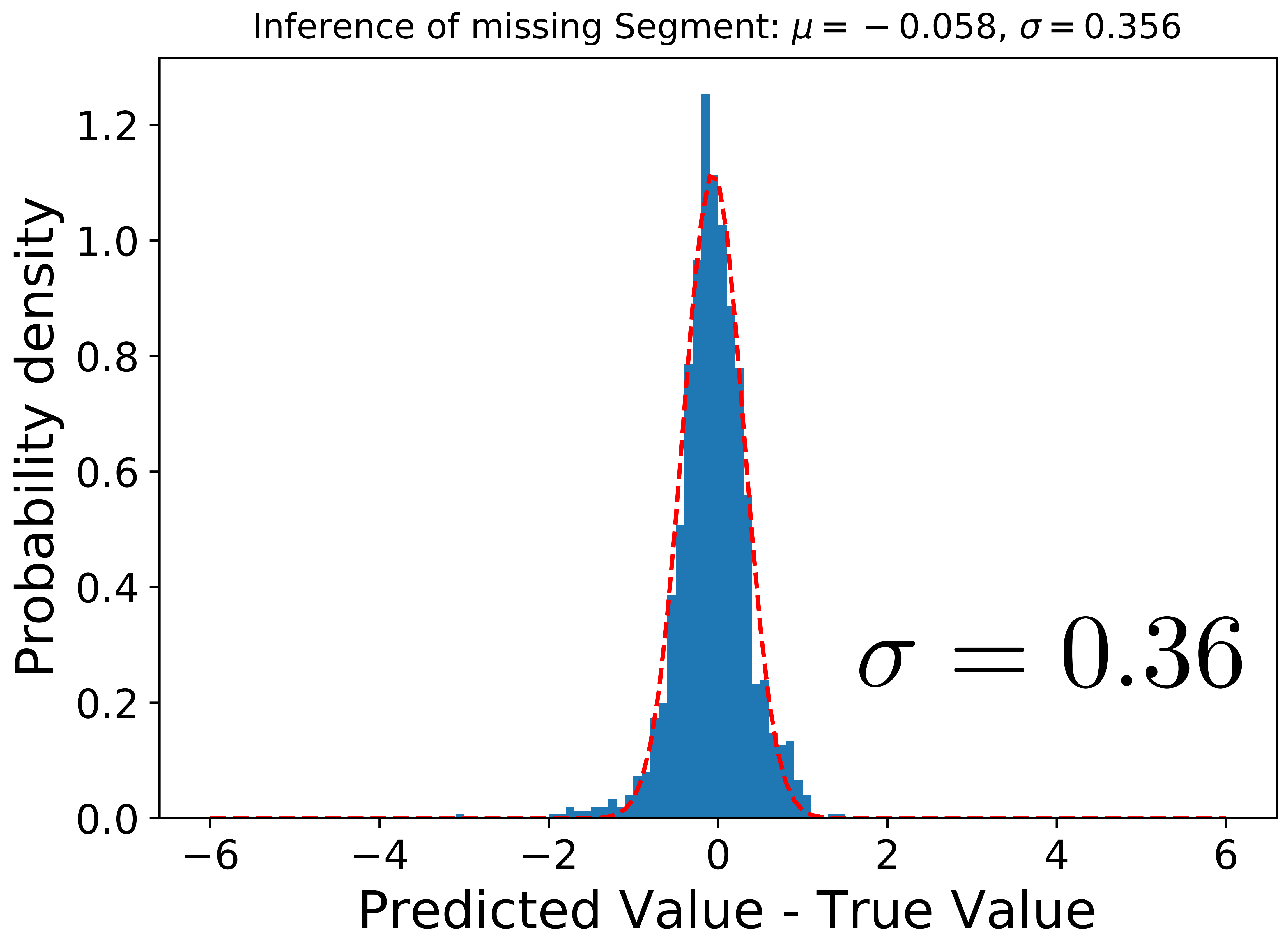
- Auto-encoder is a type of neural network that can be used to learn a compressed representation of raw data.
- An auto-encoder is composed of an encoder and a decoder sub-models. The encoder compresses the input and the decoder attempts to recreate the input from the compressed version provided by the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

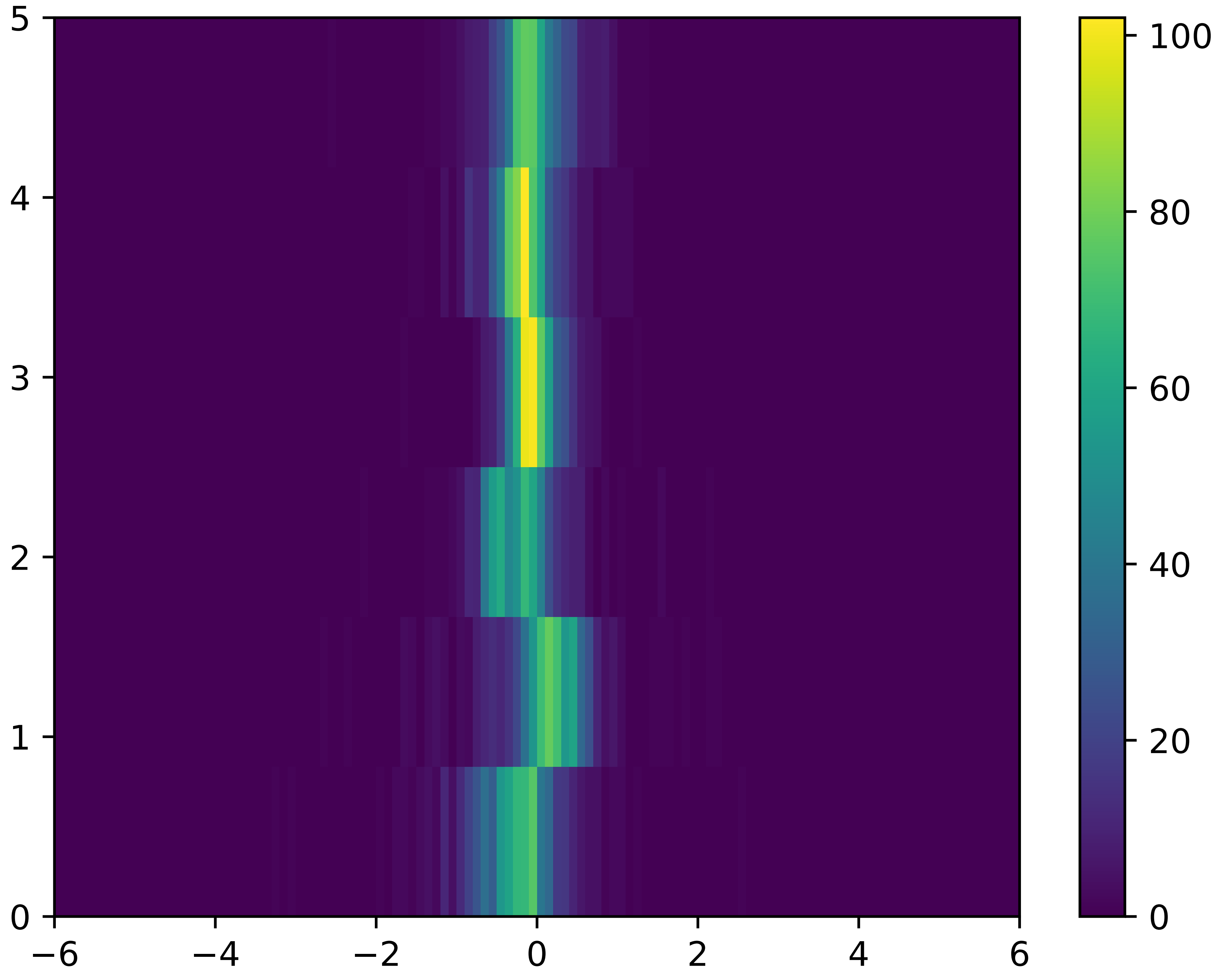




- Use Auto-Encoders to fix the missing cluster (provide a position)
- Good reconstructed tracks are used to generate training samples by removing one cluster from each super layer

Fixing inefficiencies (Auto-Encoder Performance)

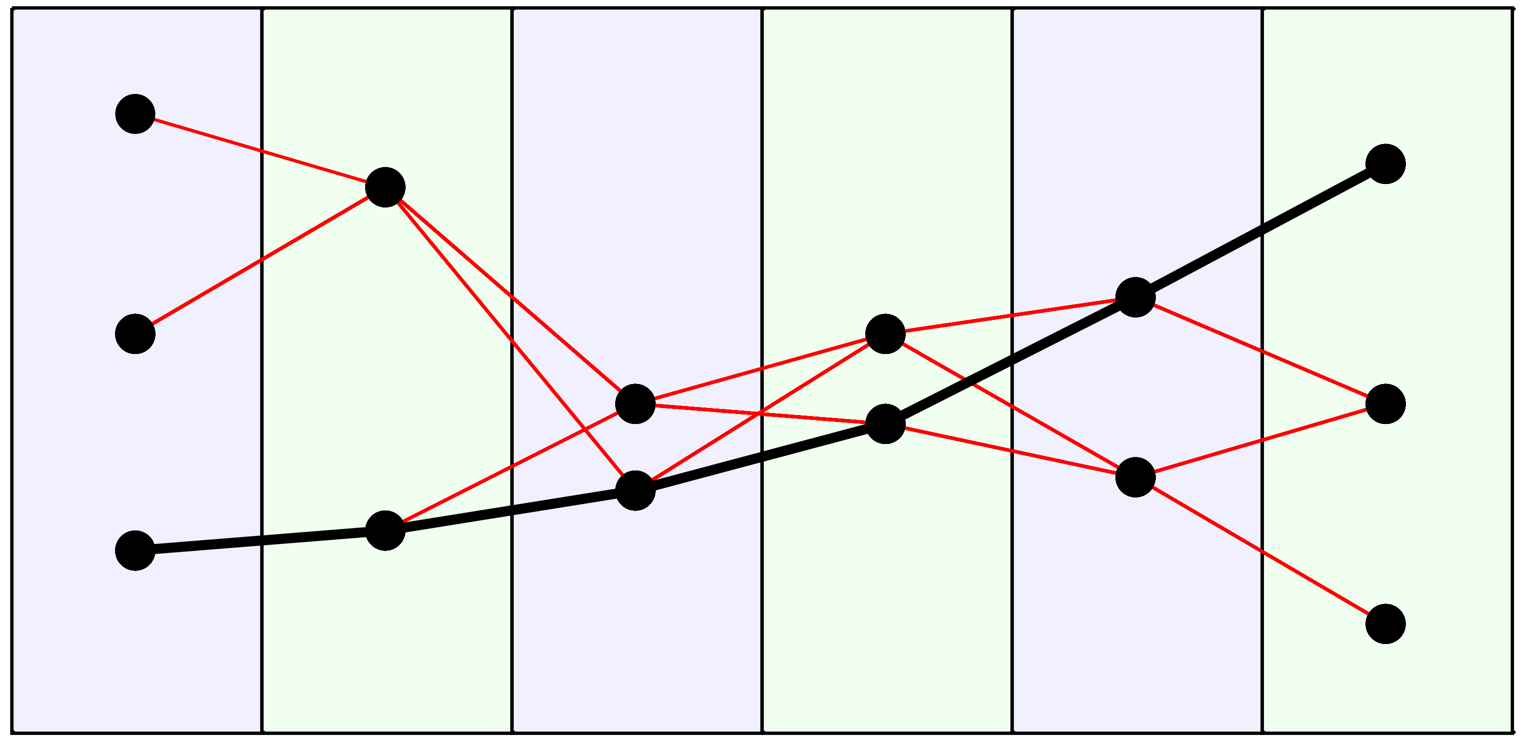




- Uncertainty in prediction for cluster position for good tracks is 0.36 wire out of 112
- Uncertainty in prediction for cluster position vs Super-layer with missing cluster

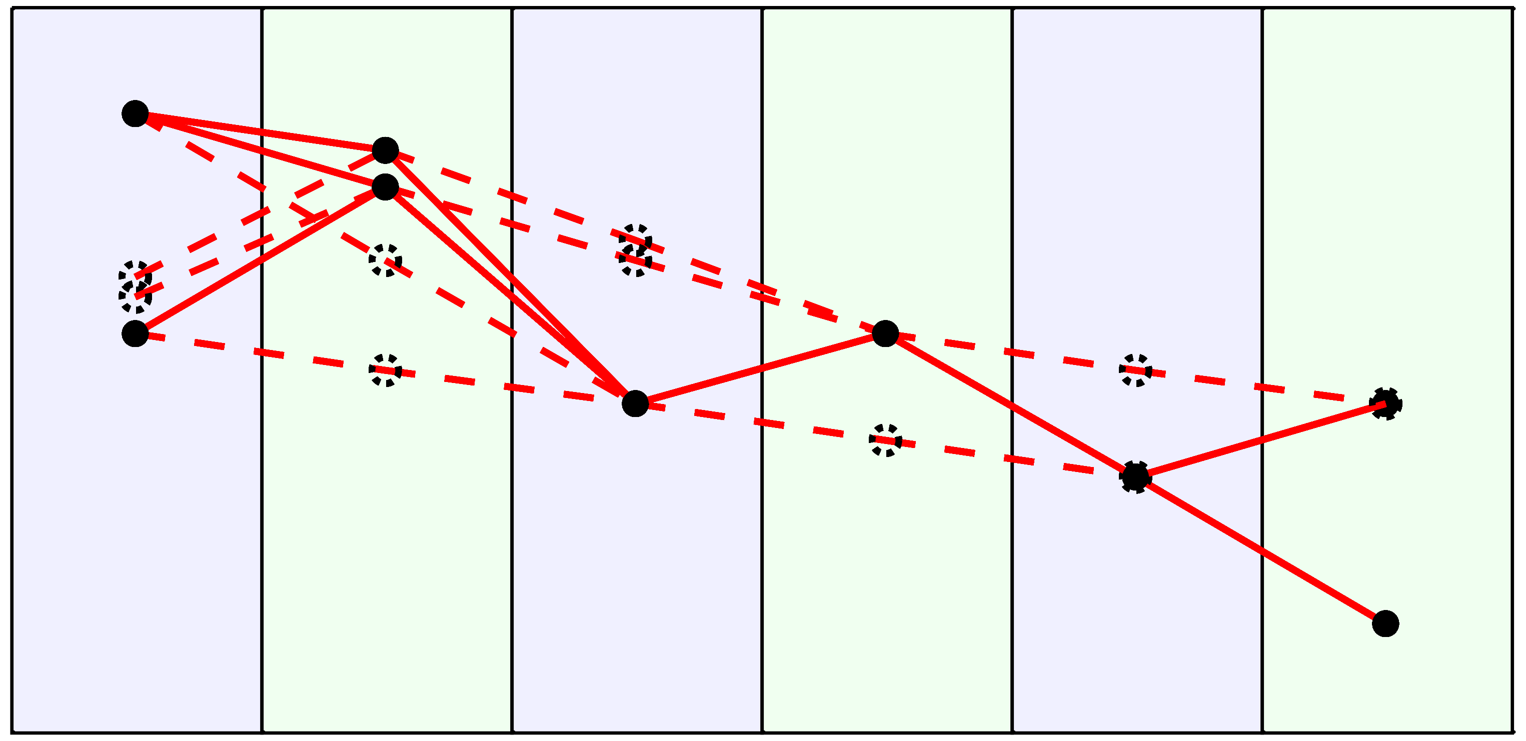
Identifying tracks using both networks

25 candidates

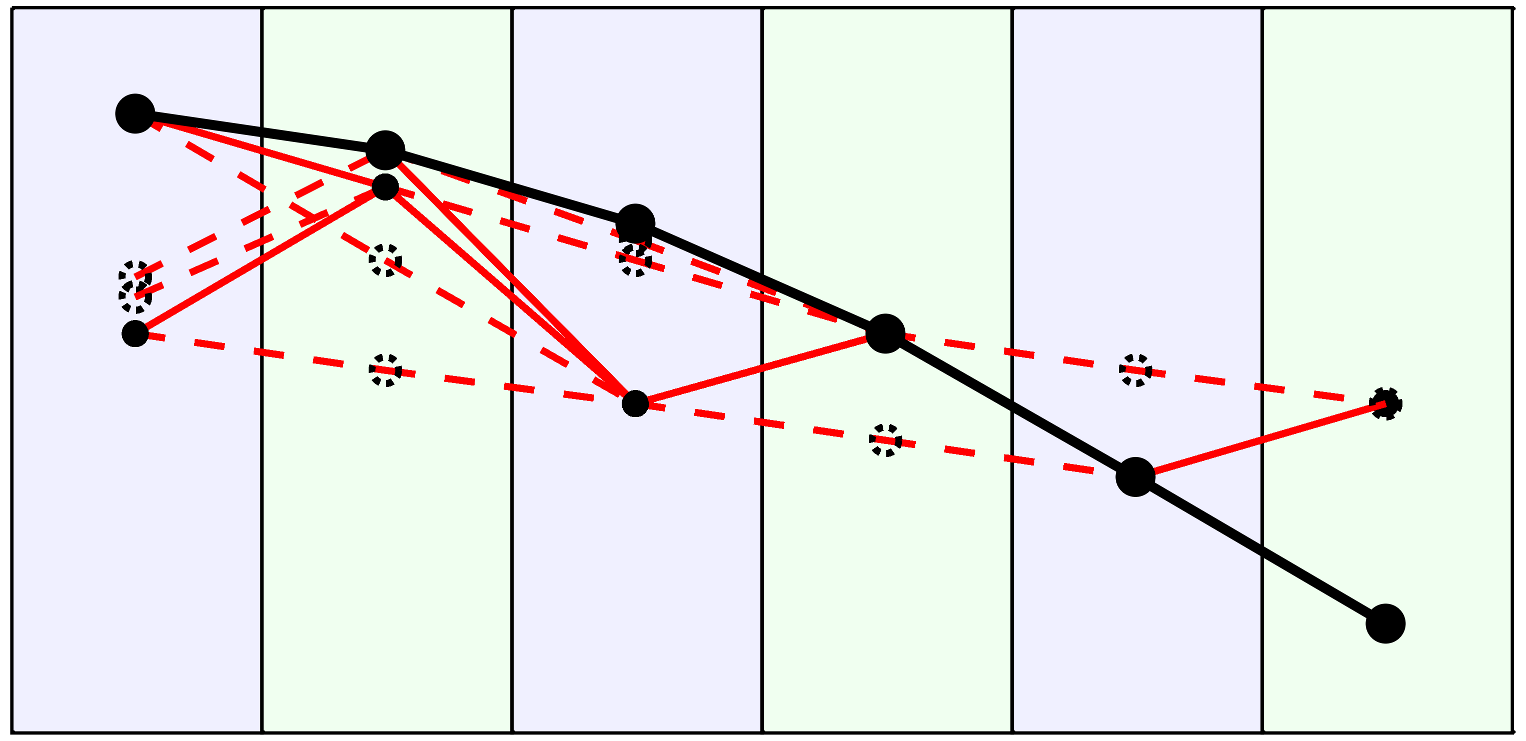


- Construct all combinations of 6 cluster tracks from hits (25 candidates in the example)
- Evaluate track candidate likelihood using classifier neural network
- Remove hits belonging to the track from list of hits.
- Add track candidate to the list of possible tracks (with it’s probability provided by classifier)

29 candidates



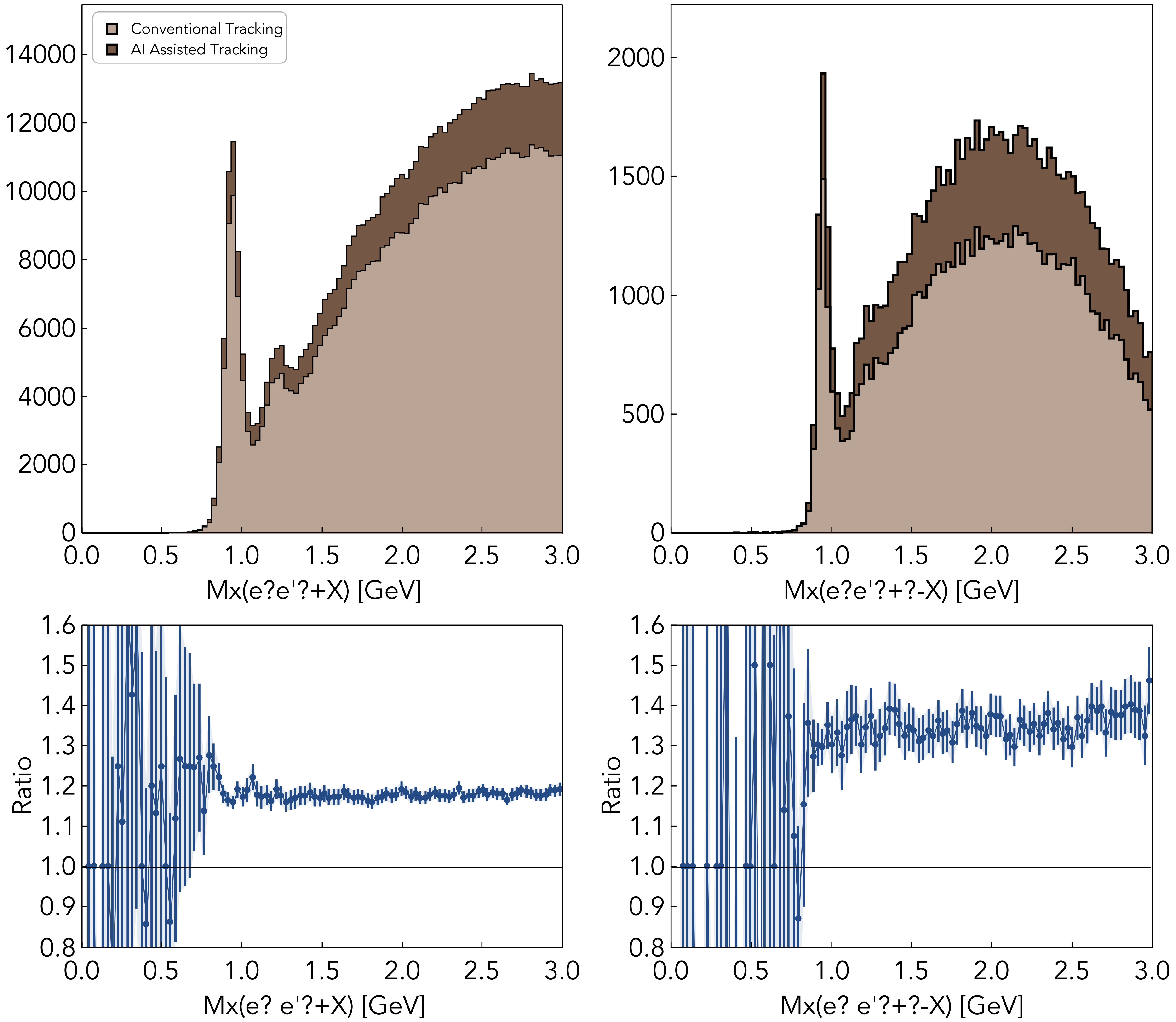
- Construct combinations of 5 cluster track candidates (29 combinations in the example)
- Generate pseudo-hits in missing super-layers using Auto-Encoder neural network
- Turn them into 6 super-layer track candidates



- Evaluate 6 clusters track candidates (with pseudo-hit) using classifier neural network
- Add track candidate to the list of possible tracks with appropriate probability

Physics Impact

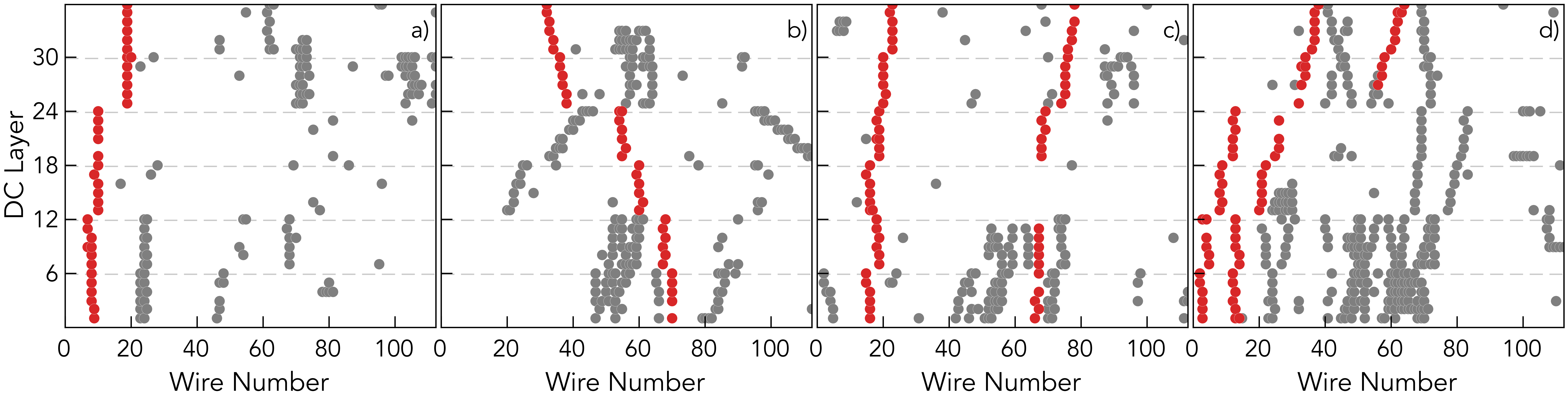
$ep \to e'\pi^+(X)$ $\quad$ $ep \to e'\pi^+\pi^-(X)$

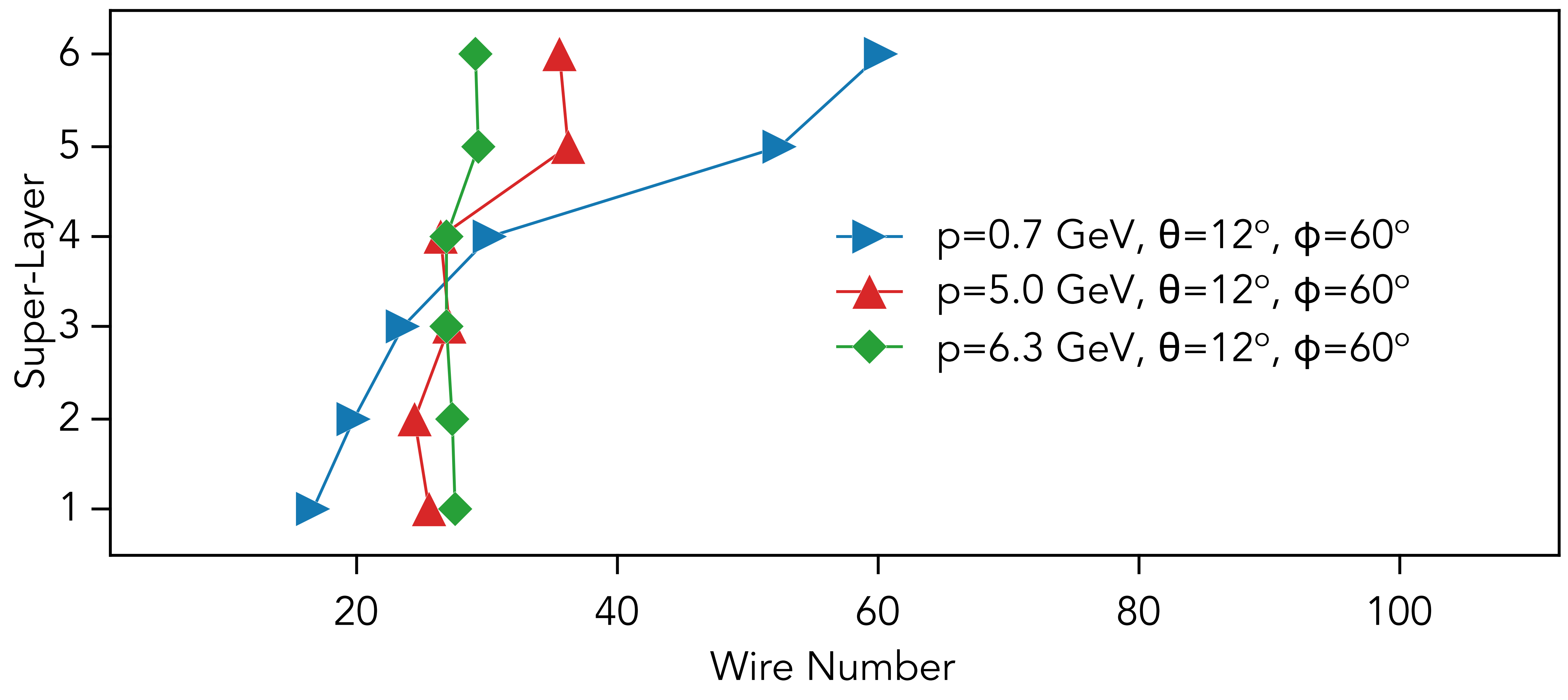


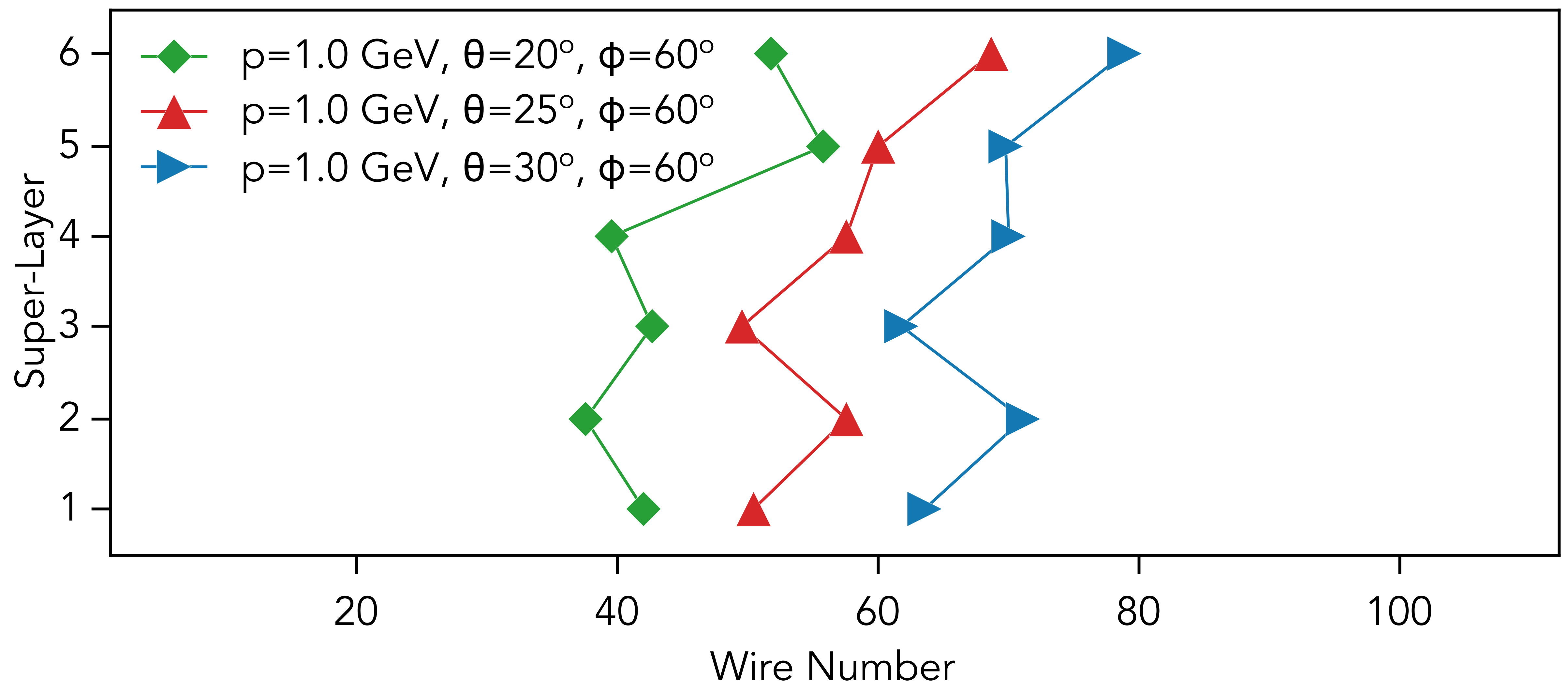
- CLAS12 tracking code reconstruction efficiency improved with the introduction of AI into track candidate finding.
- The tracking code speed improved by ~35%
- What is the physics impact?
- Tow particle final state (ep->e'pi+X) missing mass shows ~20% more event under proton peak. The gain is constant over the whole range of missing mass.
- Three particle final state (ep->e'pi+pi-X) missing mass shows ~35% increase in statistics of the missing proton.

Linear Regression (MLP)

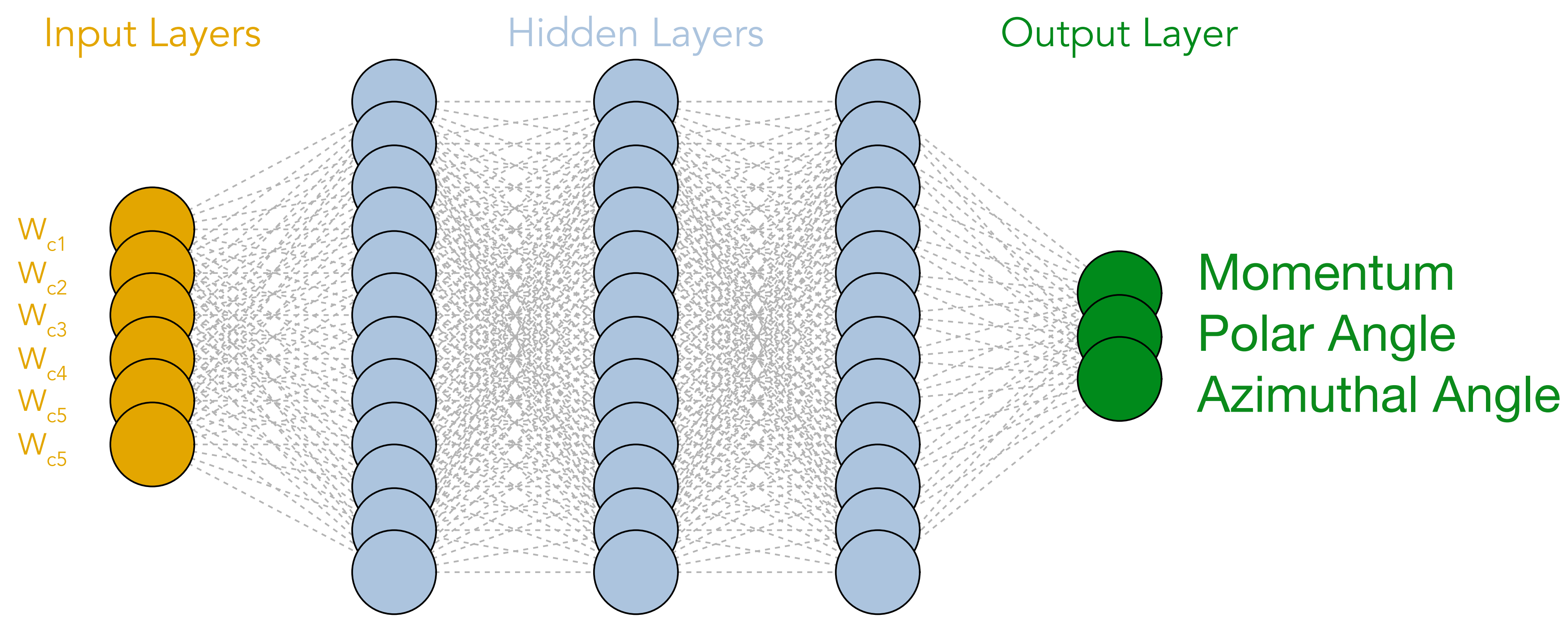
- Drift Chamber Charged Track reconstruction
  - The classifier and auto-encoder network together can identify the clusters that make a track
  - Can we predict the track parameters using Neural Networks?

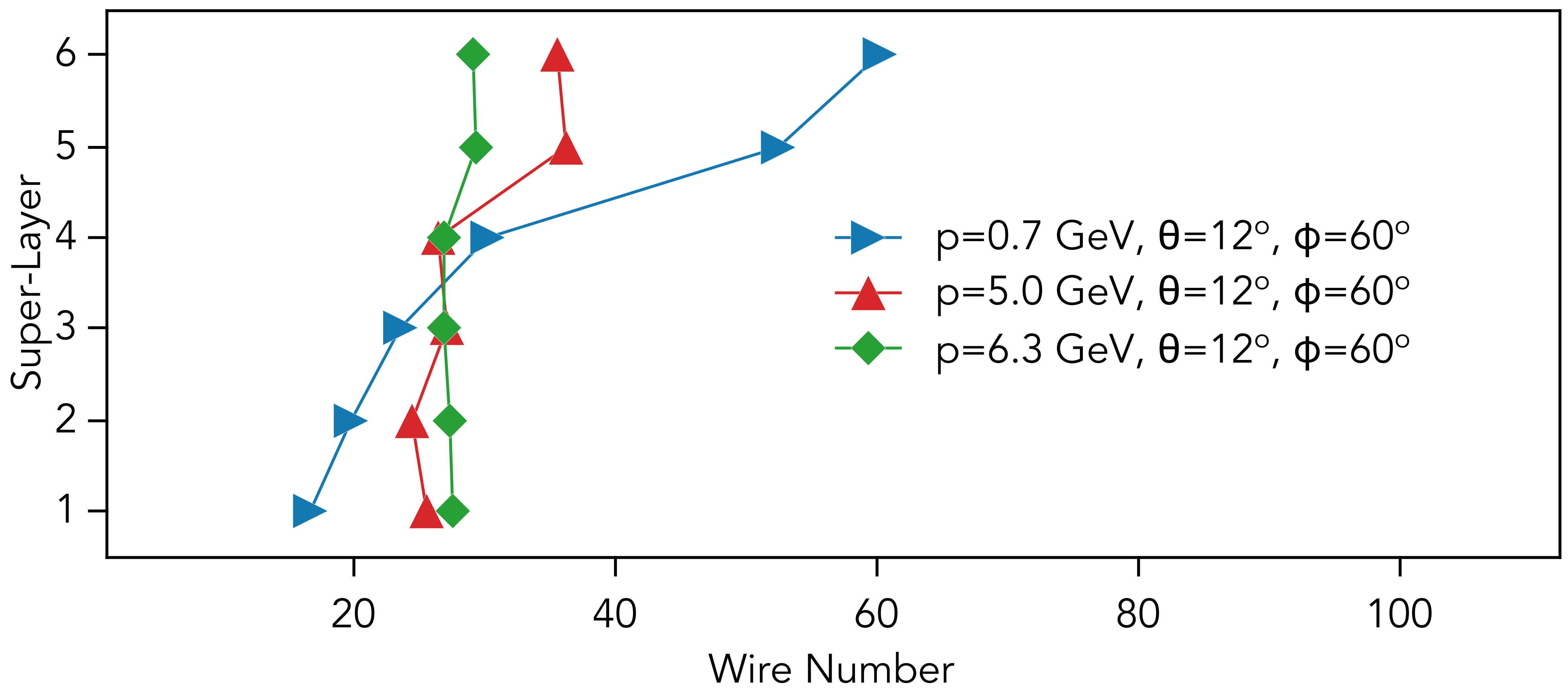






- Charge Track Parameter Inference
  - Reconstruct momentum and angles of particles based on the cluster positions of the tracks
  - Particles have distinct trajectories through drift chambers depending on their momentum, polar and azimuthal angle.
  - Design an MLP network and investigate different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





Input normalization
(1-112) -> (0-1)

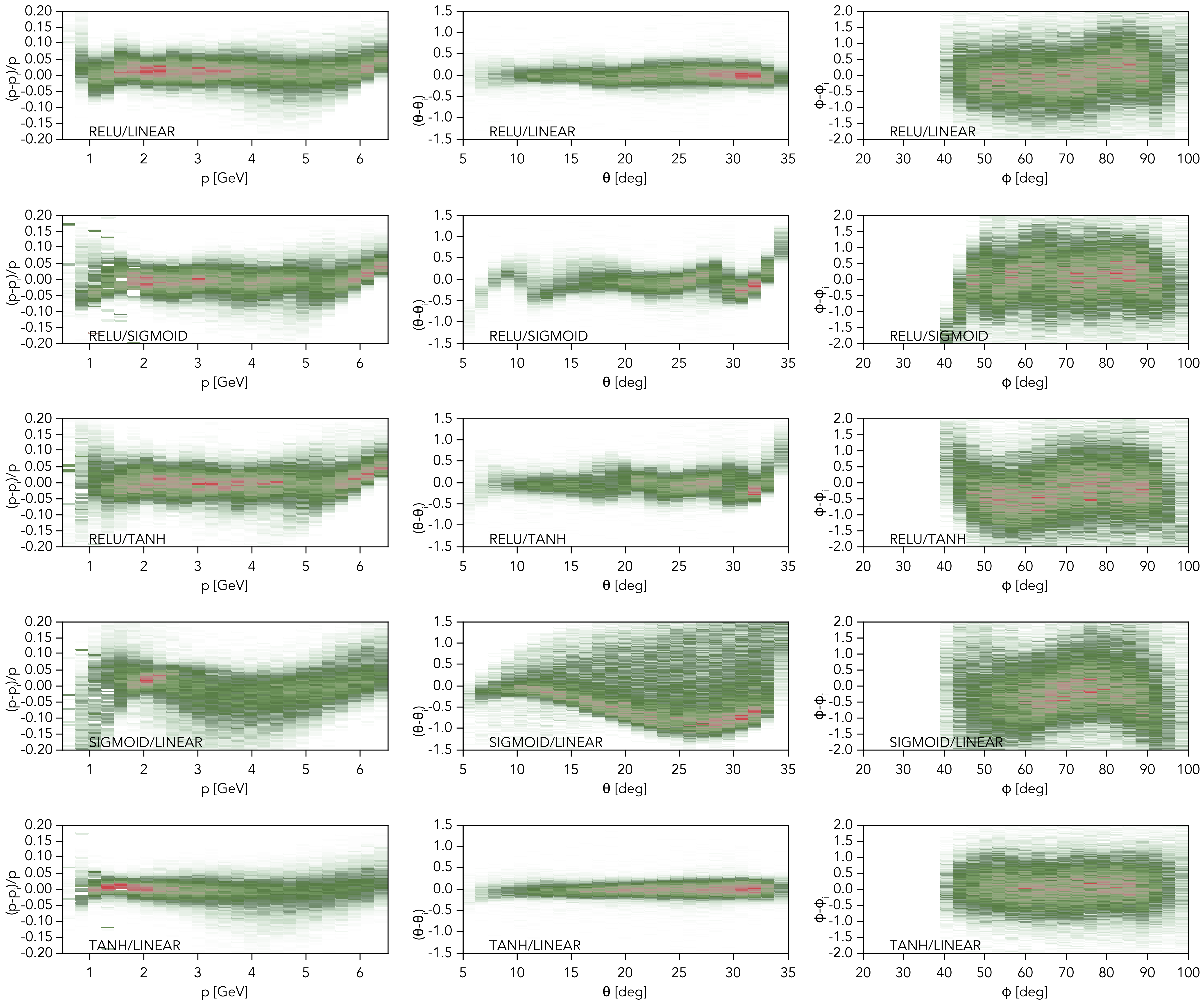
O1: normalization
(0.0-6.5) -> (0-1)

O2: normalization
(5.0-35.0) -> (0-1)

O3: normalization
(40.0-120.0) -> (0-1)

Input	Output	Input	Output	Input	Output
28 (0.25)		24		18	
27 (0.24)	0.7 (0.107)	23	5.0 (0.769)	20	6.3 (0.970)
26 (0.23)	12 (0.825)	27	12 (0.825)	22	12 (0.825)
25 (0.22)	60 (0.250)	26	60 (0.250)	30	60 (0.250)
30 (0.27)		38		52	
29 (0.26)		36		61	

Inputs and outputs are normalized to (0-1) range



RELU/LINEAR

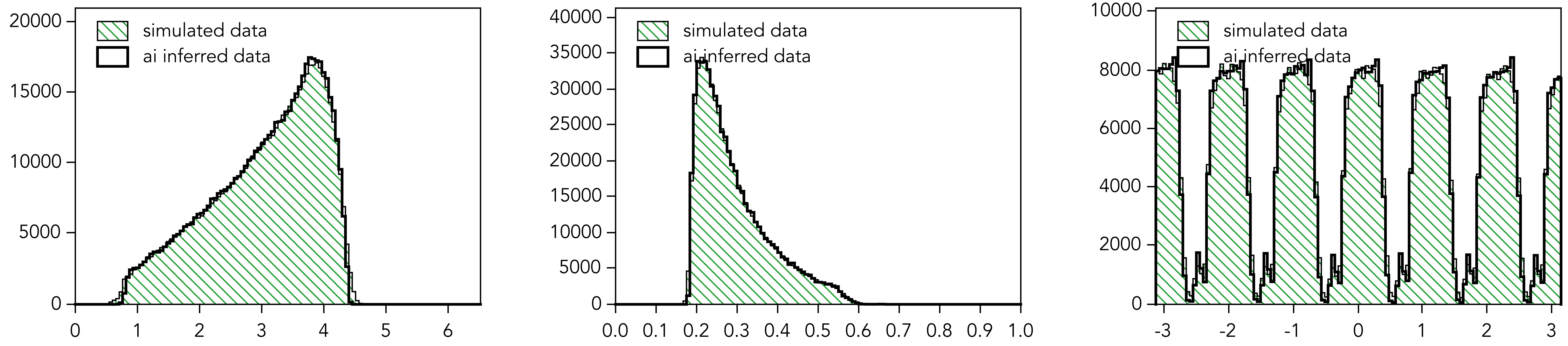
RELU/SIGMOID

RELU/TANH

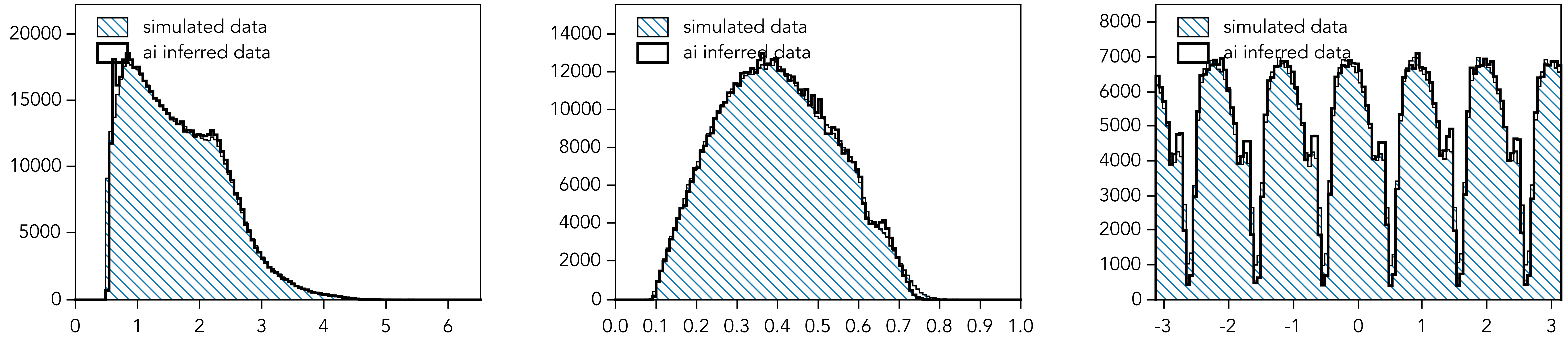
SIGMOID/LINEAR

TANH/LINEAR

Negative tracks

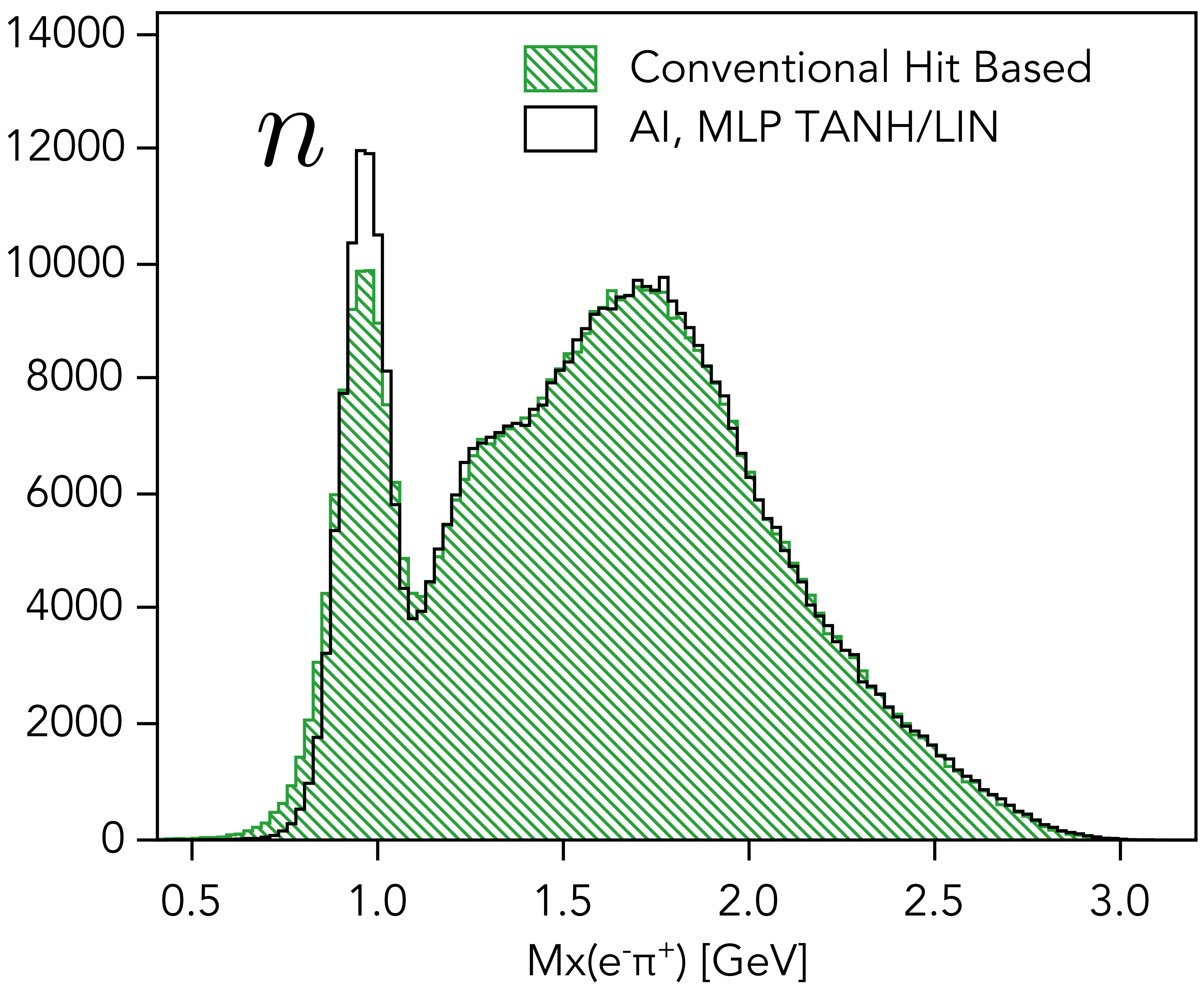


Positive tracks



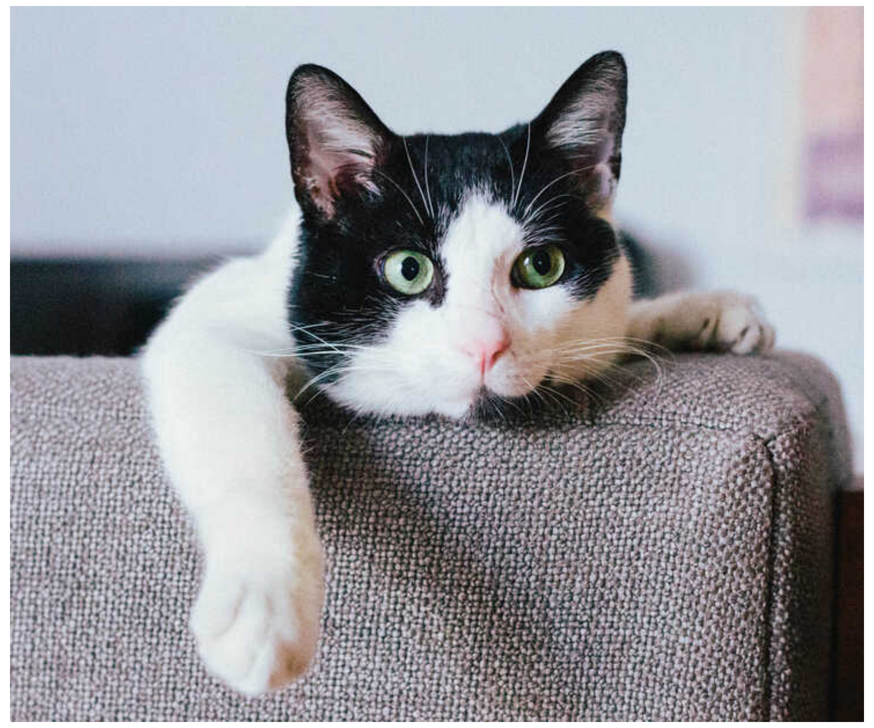
- Track Reconstruction With Artificial Intelligence
  - Reconstructed momentum and angle resolutions are better than what conventional algorithms can do at HIT BASED track reconstruction
- Data Reconstruction Speed
  - Track parameters are calculated at the rate of 34 kHz (using 32 Threads)
  - Data Collection rate in CLAS12 12kHz
  - Physics Reaction identification
- Applications
  - Tag physics reaction at Data Acquisition time
  - Reduce data volume based on physics requirements
  - Perform Online data calibration
  - Monitor detector performance

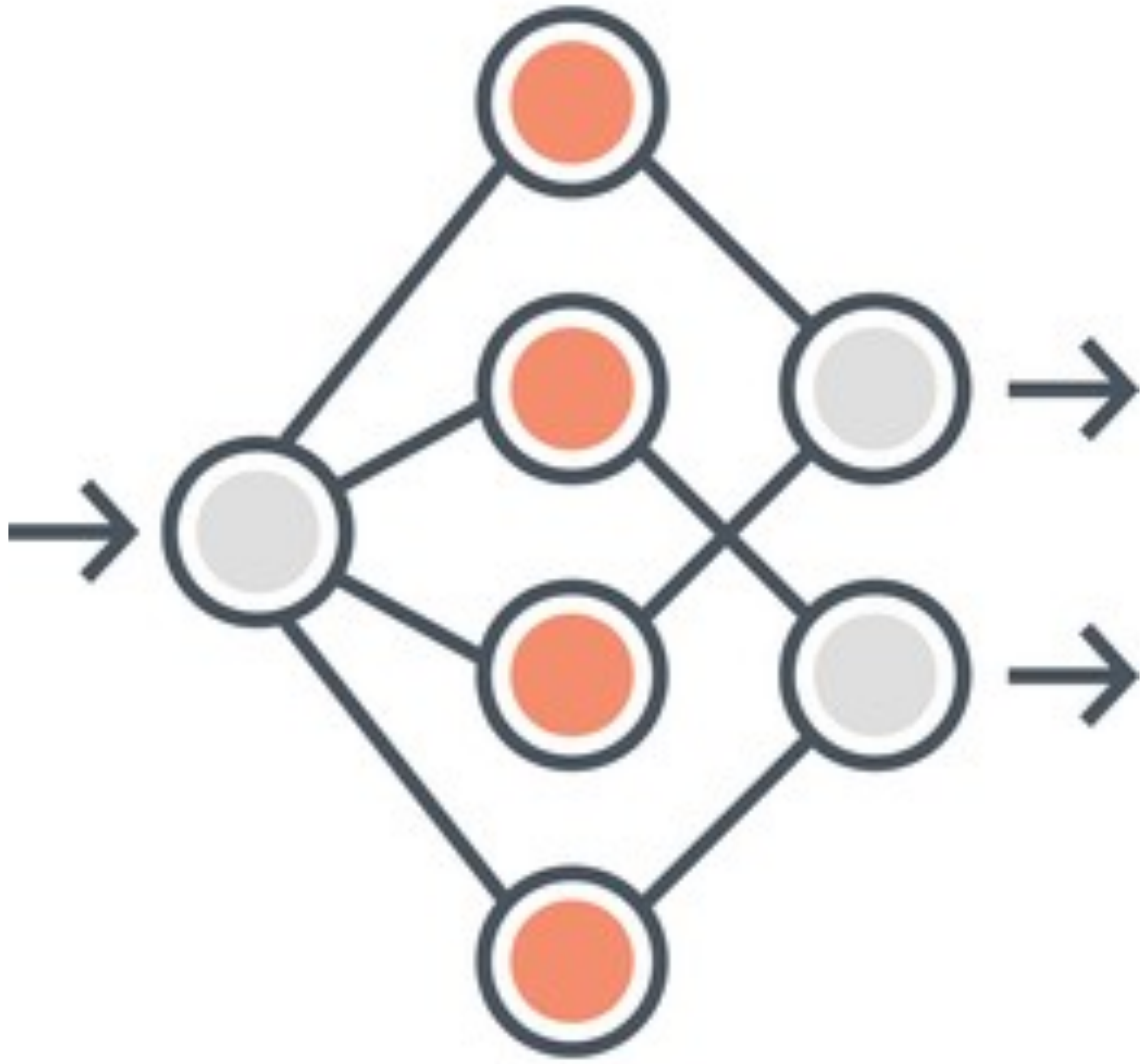
$$ep \to e^- n\pi^+$$



Chihuahua or Muffin?

Should I care about the training sample?





Cat

Not A Cat

Neural Networks (Classification)

?

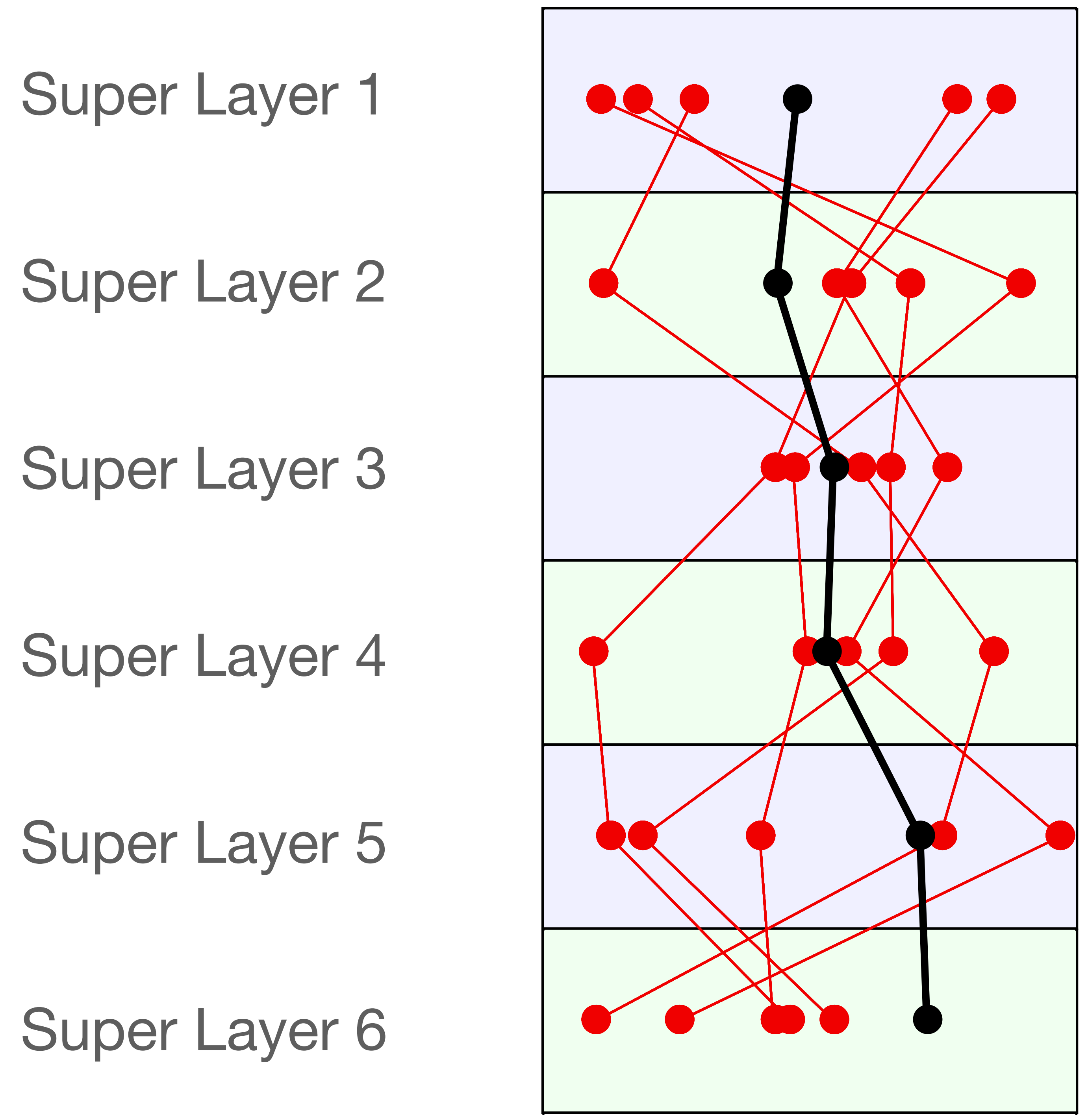
Watson

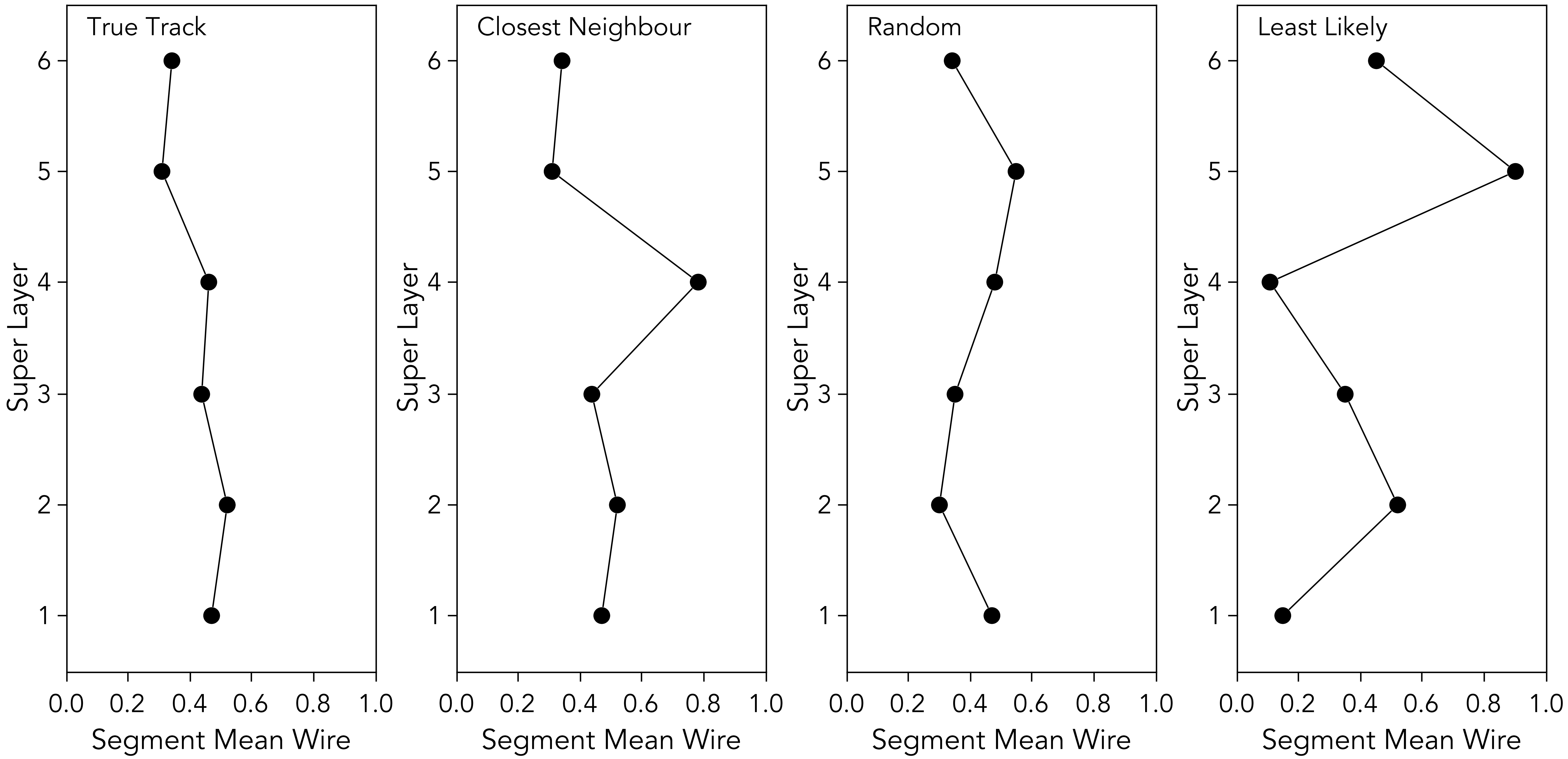
Gets really upset if you classify him as a “CAT”

When creating a training sample:
Special care should be given to negative samples.

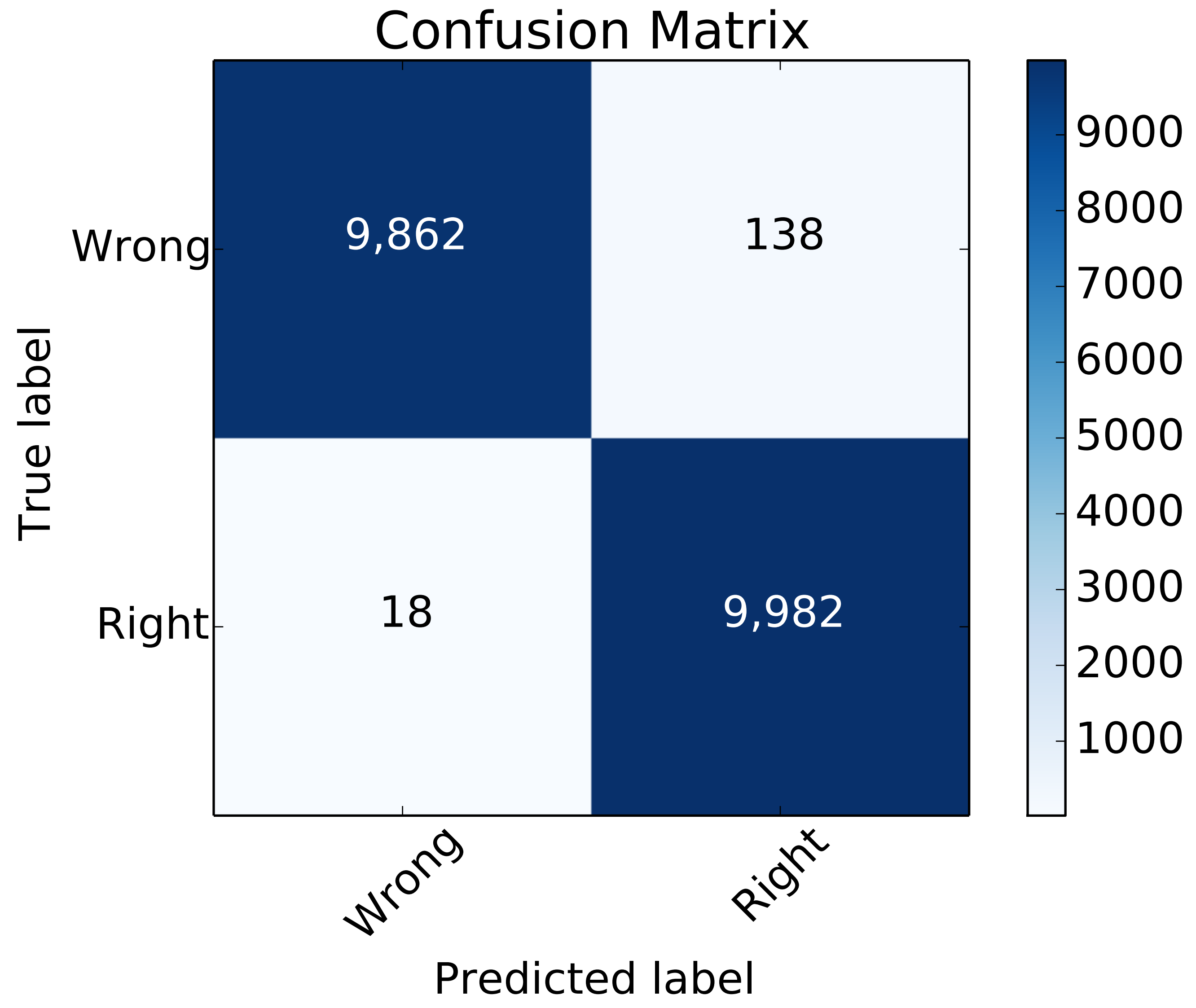
Neural Networks (Classification)

Different Training Samples



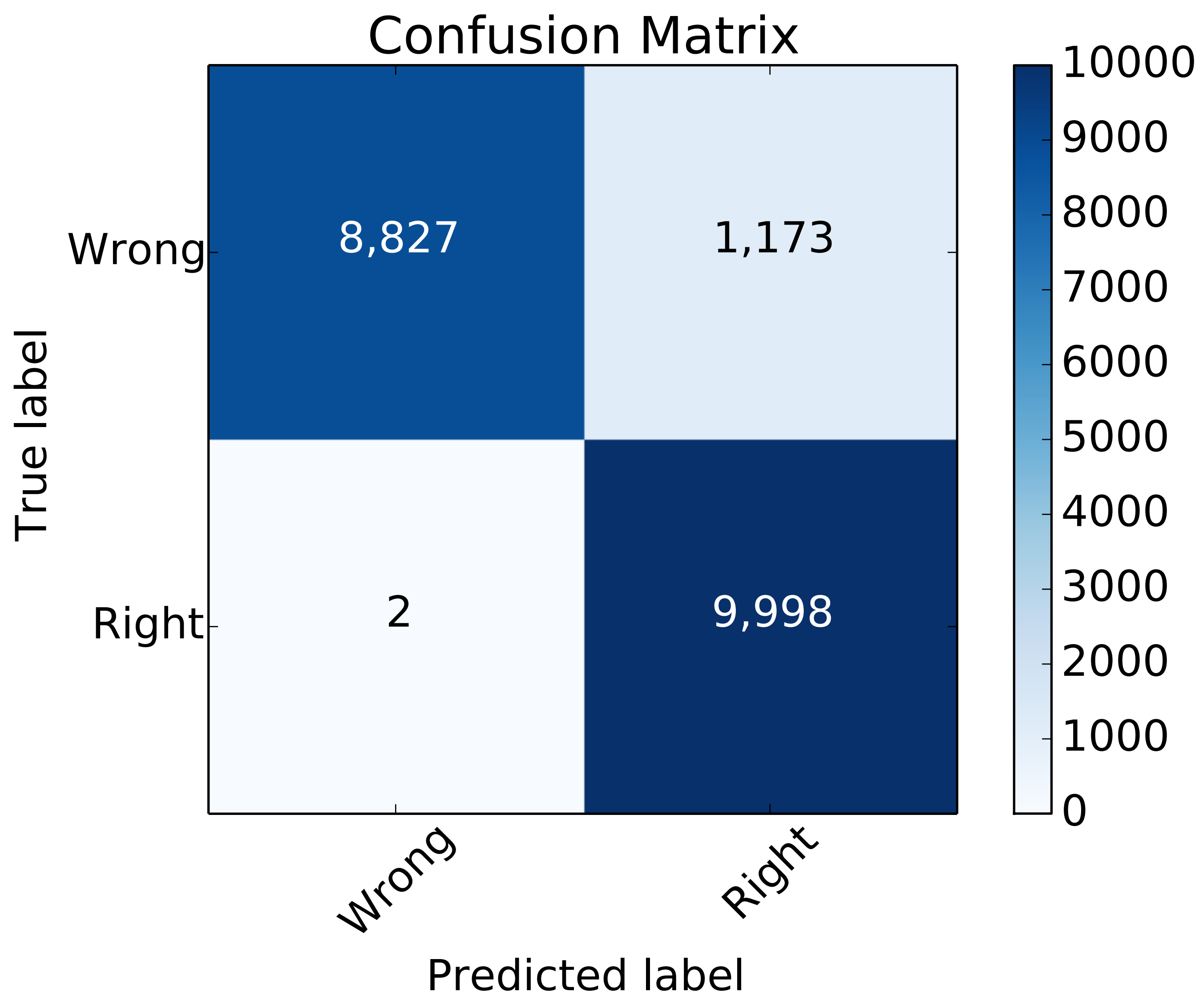


Closest Neighbor



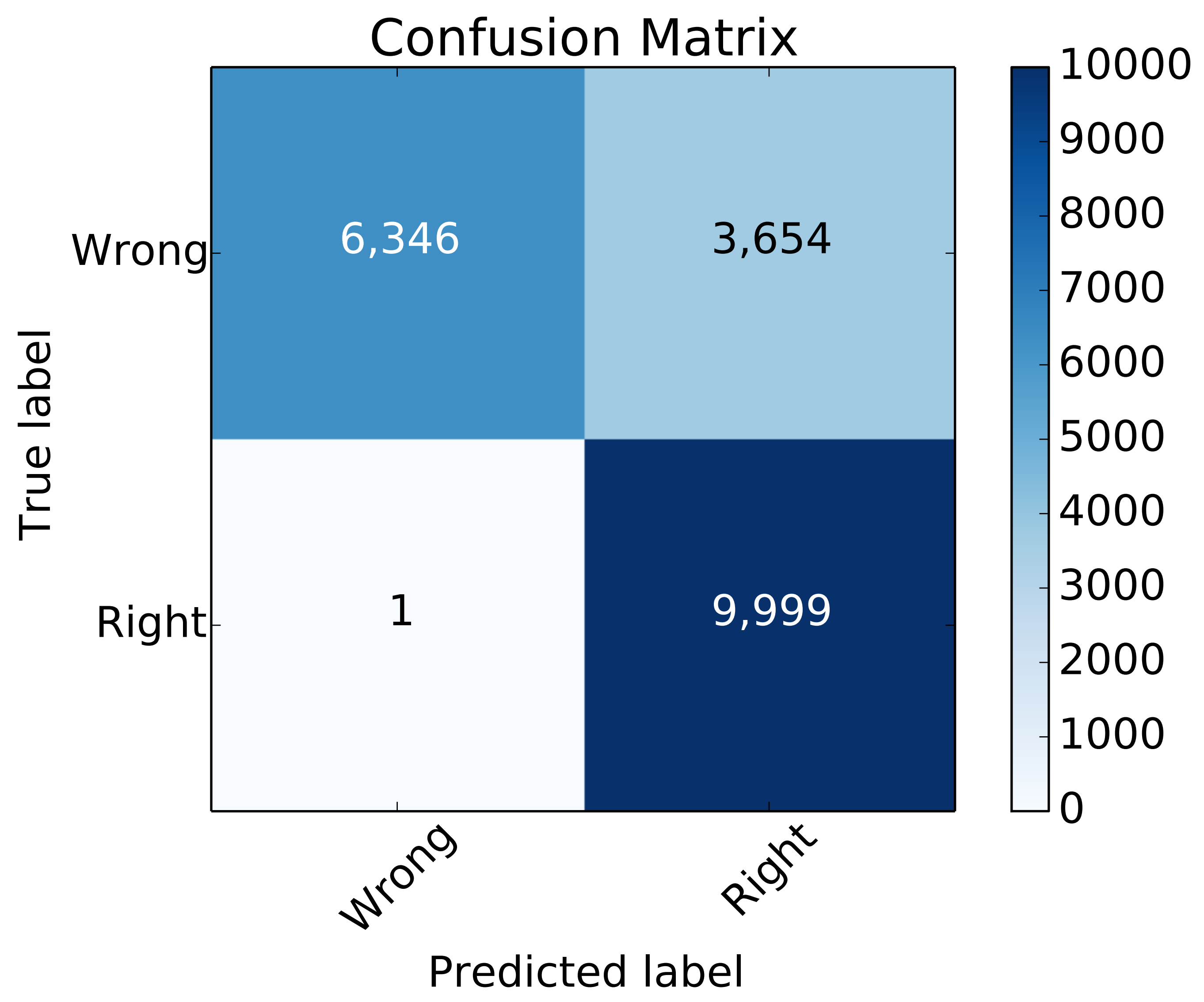
False Positive 1.38%

Random



False Positive 11.73%

Least Likely



False Positive 36.54%

Classification
- MLP network to classify track candidates based on segment positions in each super layer of drift chambers
- Increased the efficiency of track reconstruction by ~5%

Auto-Encoders
- Auto Encoders are used to identify the missing segment in the track trajectory and create complete tracks from 5 segment track candidates
- Combined with the classifier increased track reconstruction efficiency by ~12%-15%
- Track identification resulted in an experimental statistic increase of 15%-35% (reaction dependent)

Linear Regression
- MLP Linear regression is used to calculate (infer) track parameters (i.e. momentum, polar and azimuthal angles) from identified track candidates
- Reconstructed tracks with MLP have a better resolution that Hit based tracking and provide rates >30kHz
- Makes it possible to identify physics reactions at run-time, and can be used for triggering specific reactions.