ML status @MINERvA

Anushree Ghosh, UTFSM, Chile

On behalf of the MINERvA collaboration

Reconstruction and machine learning in
Neutrino experiment, Hamburg, Germany
2019-09-17

Plan for the talk

- Introduction to MINERvA
- Application of Convolutional Neural Network for finding the interaction vertex
    - Deal with the possible bias from training sample: **DANN**
    - Evolutionary algorithm: **MENNDL**
- Classification of hadron multiplicity using ML
    - **Transfer learning**
- Application of **semantic segmentation** approach for neural pion reconstruction

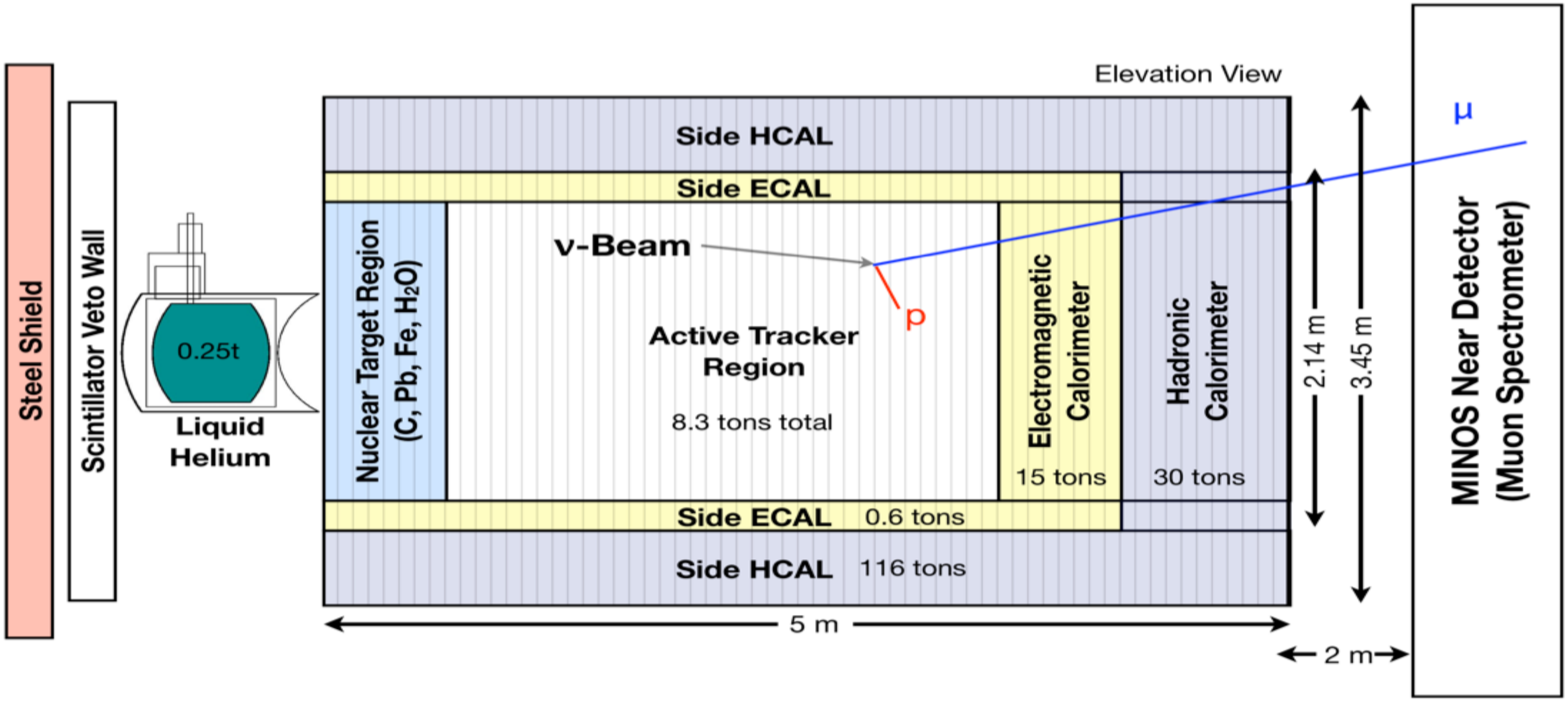
MINERvA Experiment

- **MINERvA** is a dedicated neutrino-nucleus experiment situated at Fermilab's **NuMI Beam** along with other two experiments MINOS and NOvA

- A precise **understanding of the A-dependence** of the neutrino-nucleus cross section is important to reduce systematic uncertainties in the measurements of oscillation experiments.

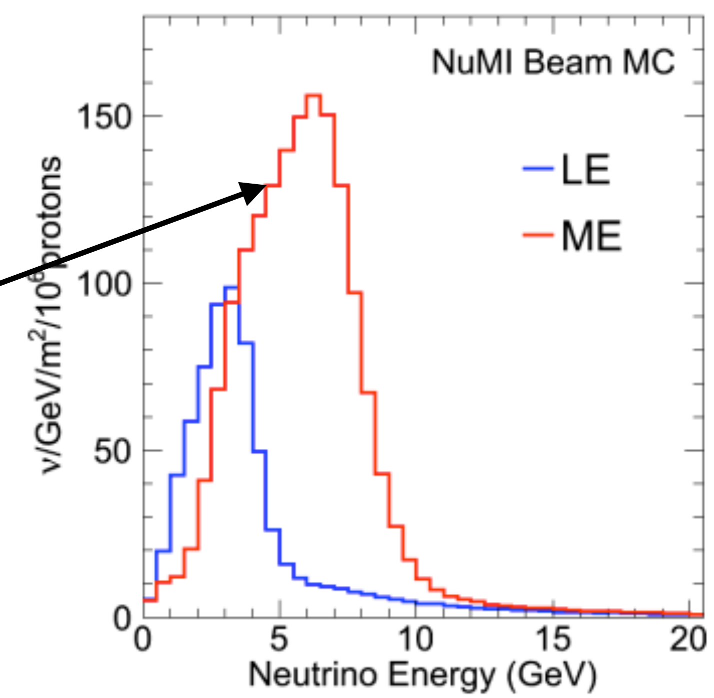
- MINERvA having **different nuclear targets** (iron, carbon, lead, water, helium, scintillator) and excellent tracking ability, is able to provide high precision measurement of neutrino interactions on various nuclei in the 1-20 GeV energy range.

MINERvA Detector



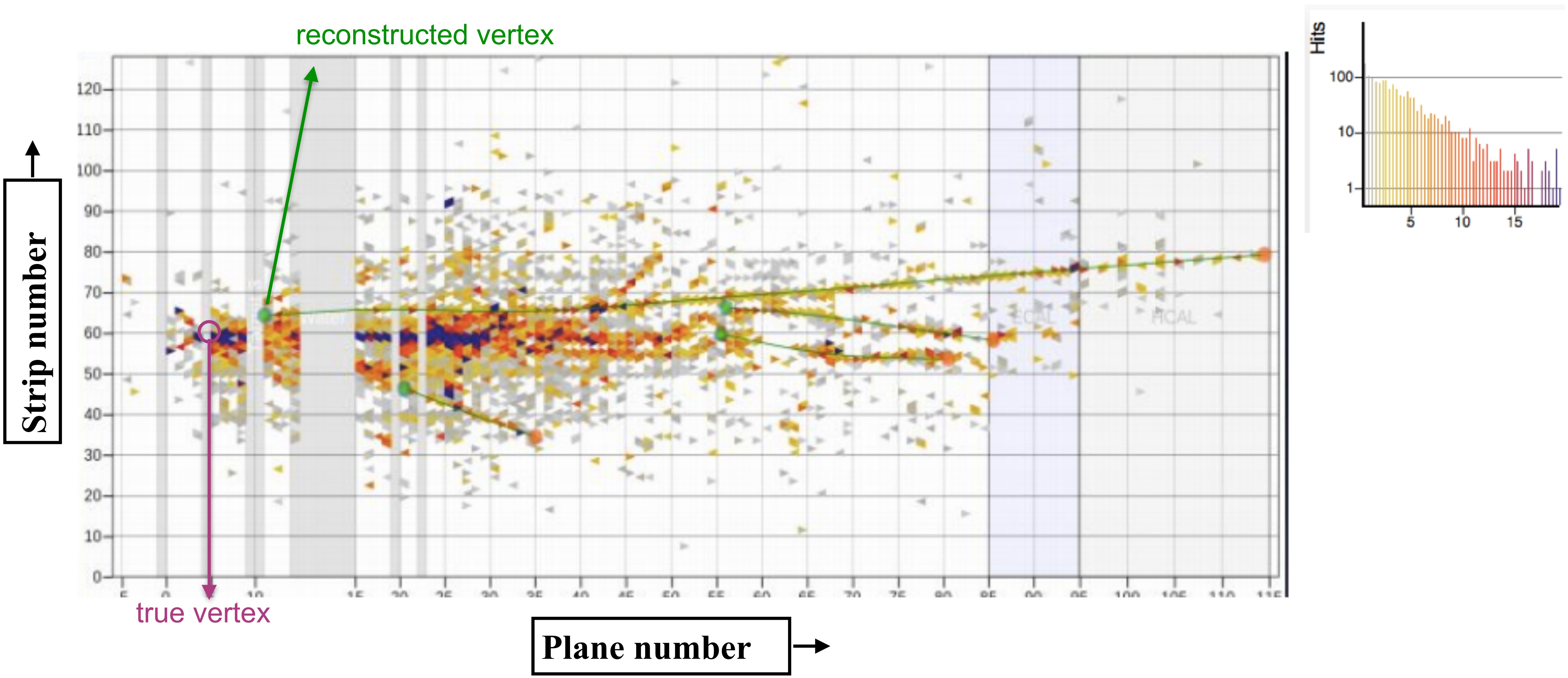
- Consists of a core of scintillator strips surrounded by ECAL and HCAL
- MINOS Near Detector for muon charge and momentum

Medium energy flux era



Vertex reconstruction: Why ML?

- With the increase of our beam energy, there is an increase in the hadronic showers near the event of interactions.
- Cause more difficulty in vertexing with increase rates of failure in getting the correct vertex position

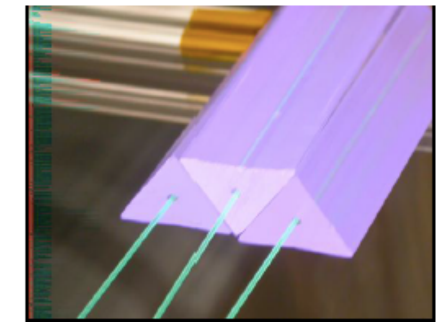


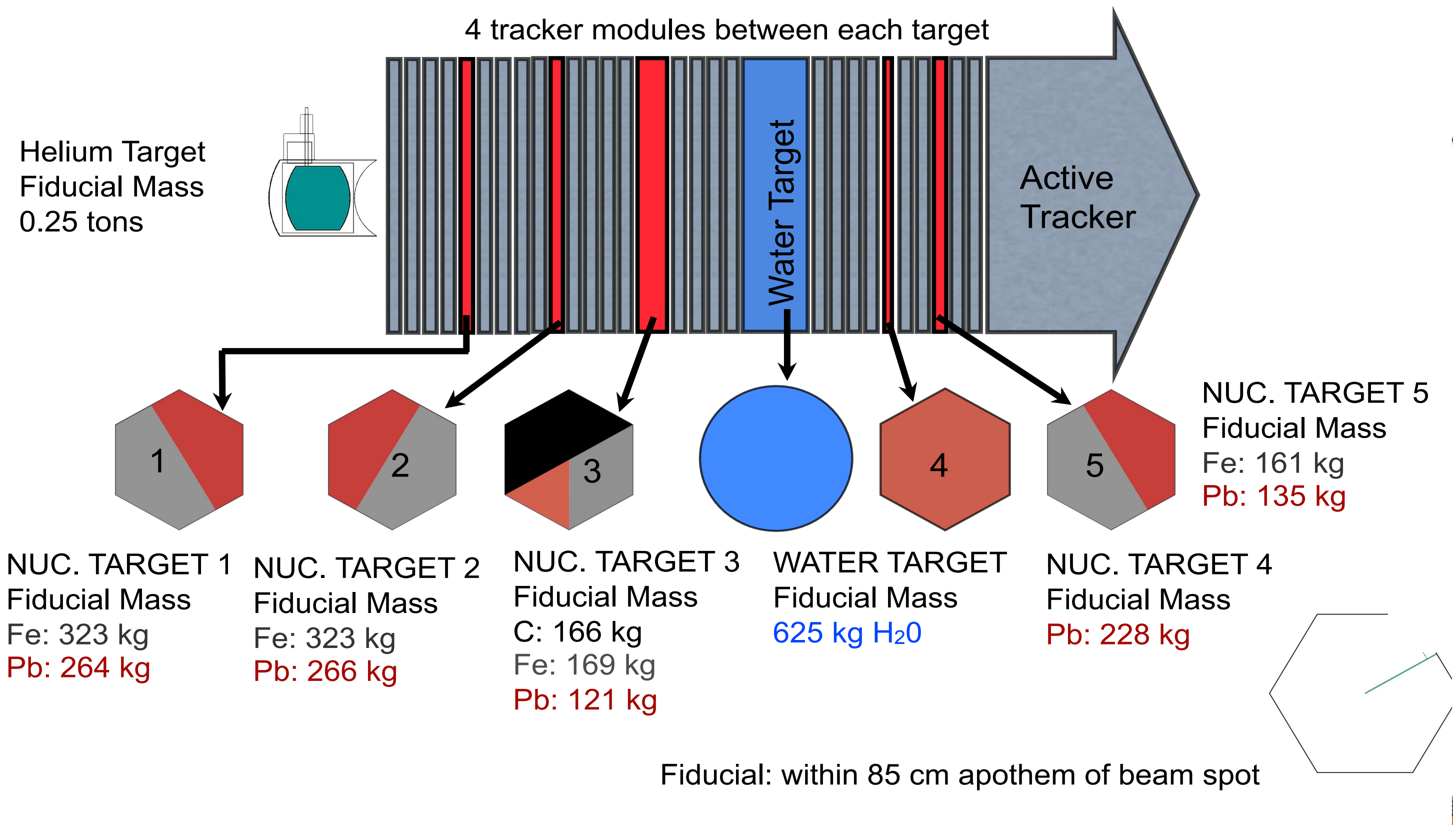
ML Approach To Determine Event Vertex

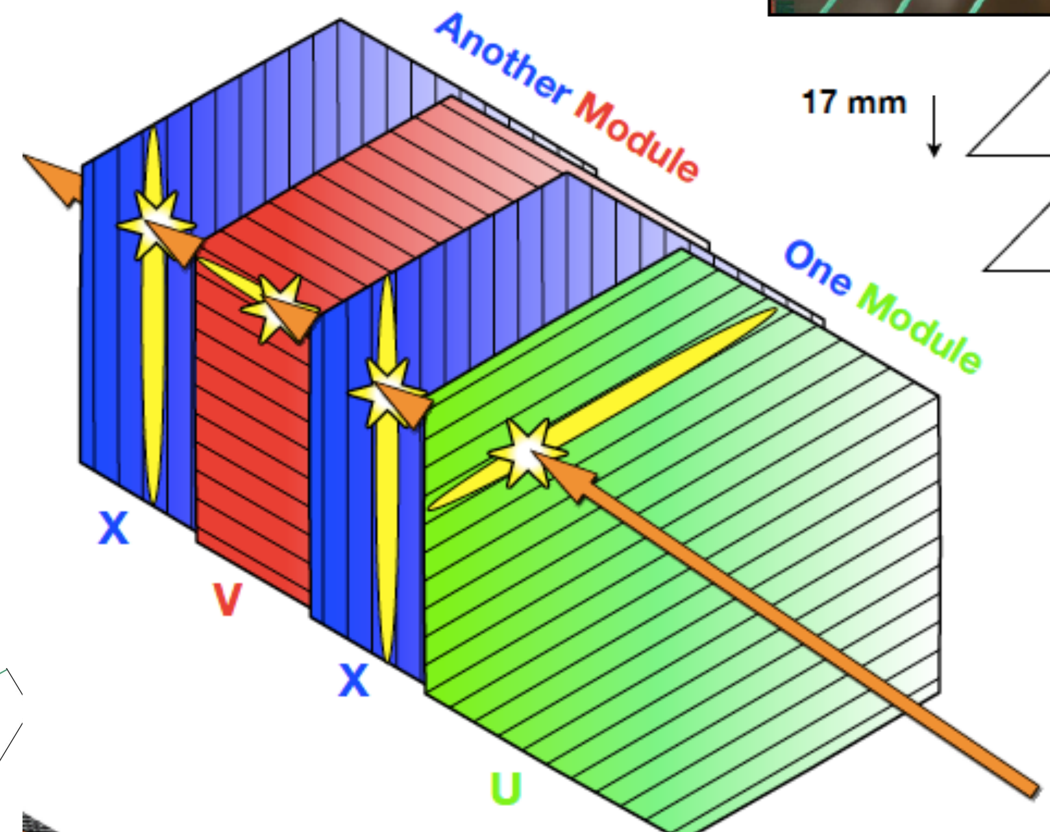
arXiv:1808.08332

Find the location of the event vertex

Treat the localization as a classification problem

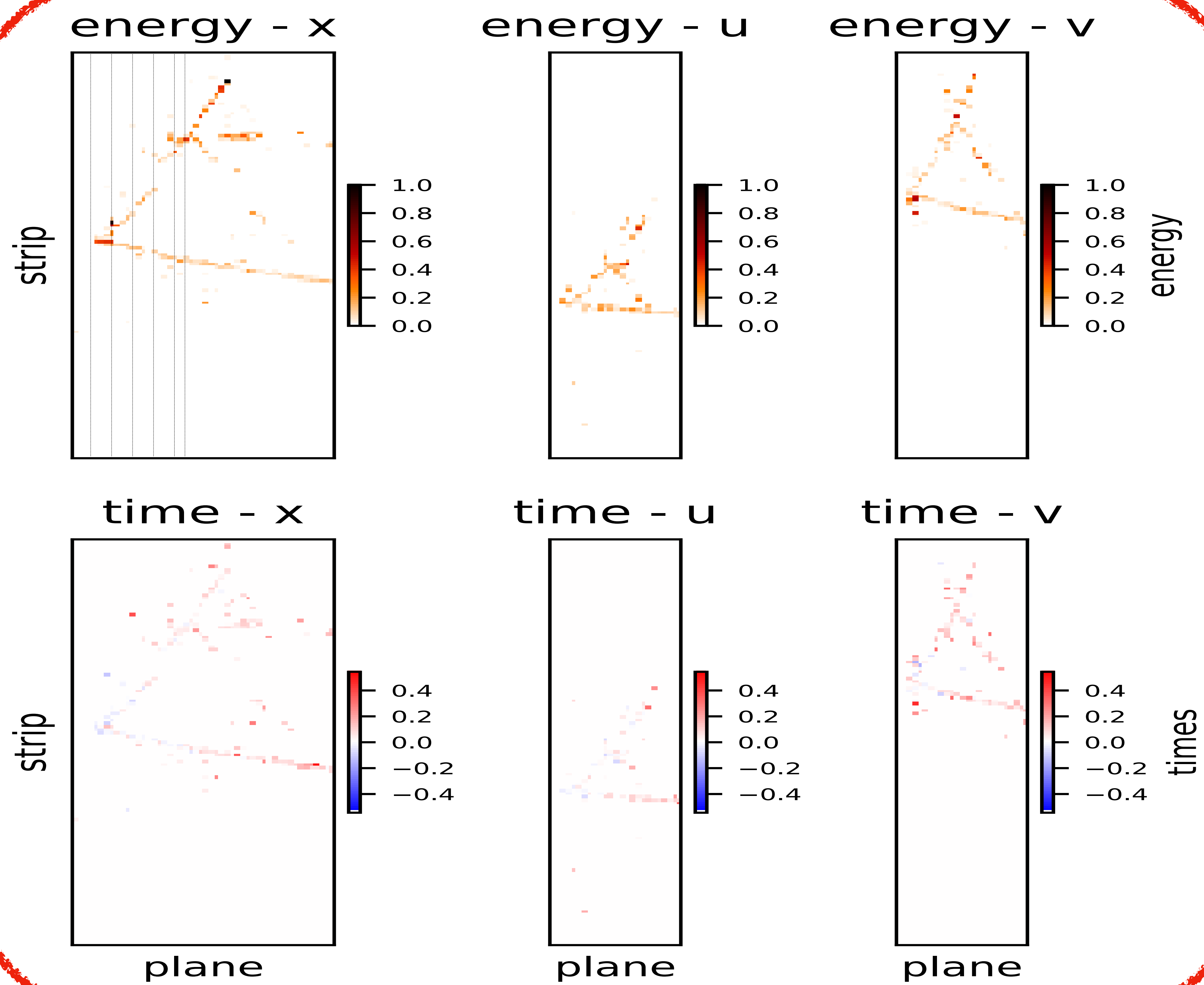






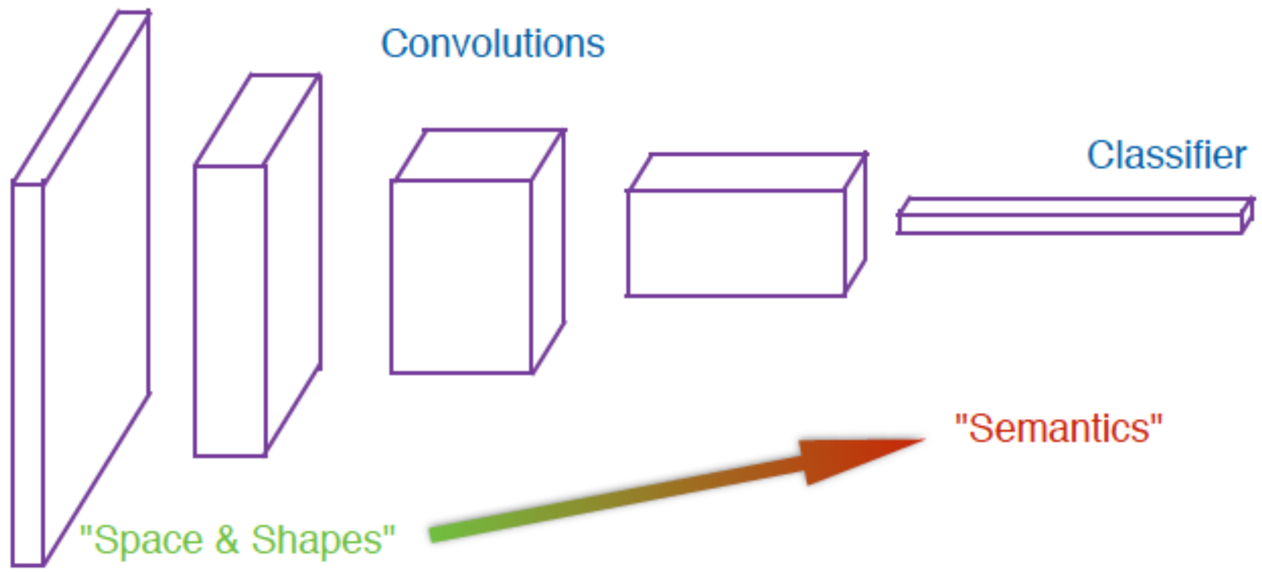
Make images for three different views → DNN(Convolutional neural network) → Classify the location of the interaction

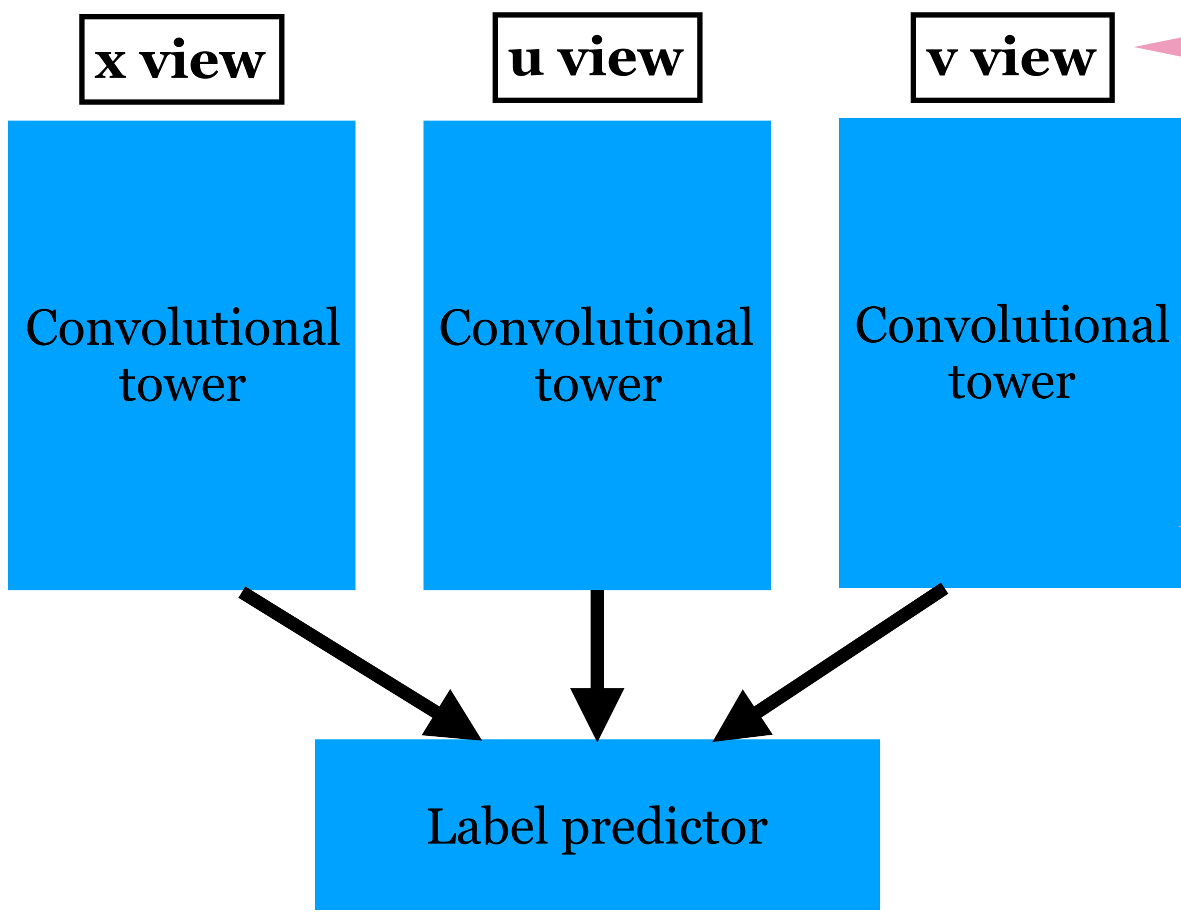
Input images



Convolutional neural network (CNN)

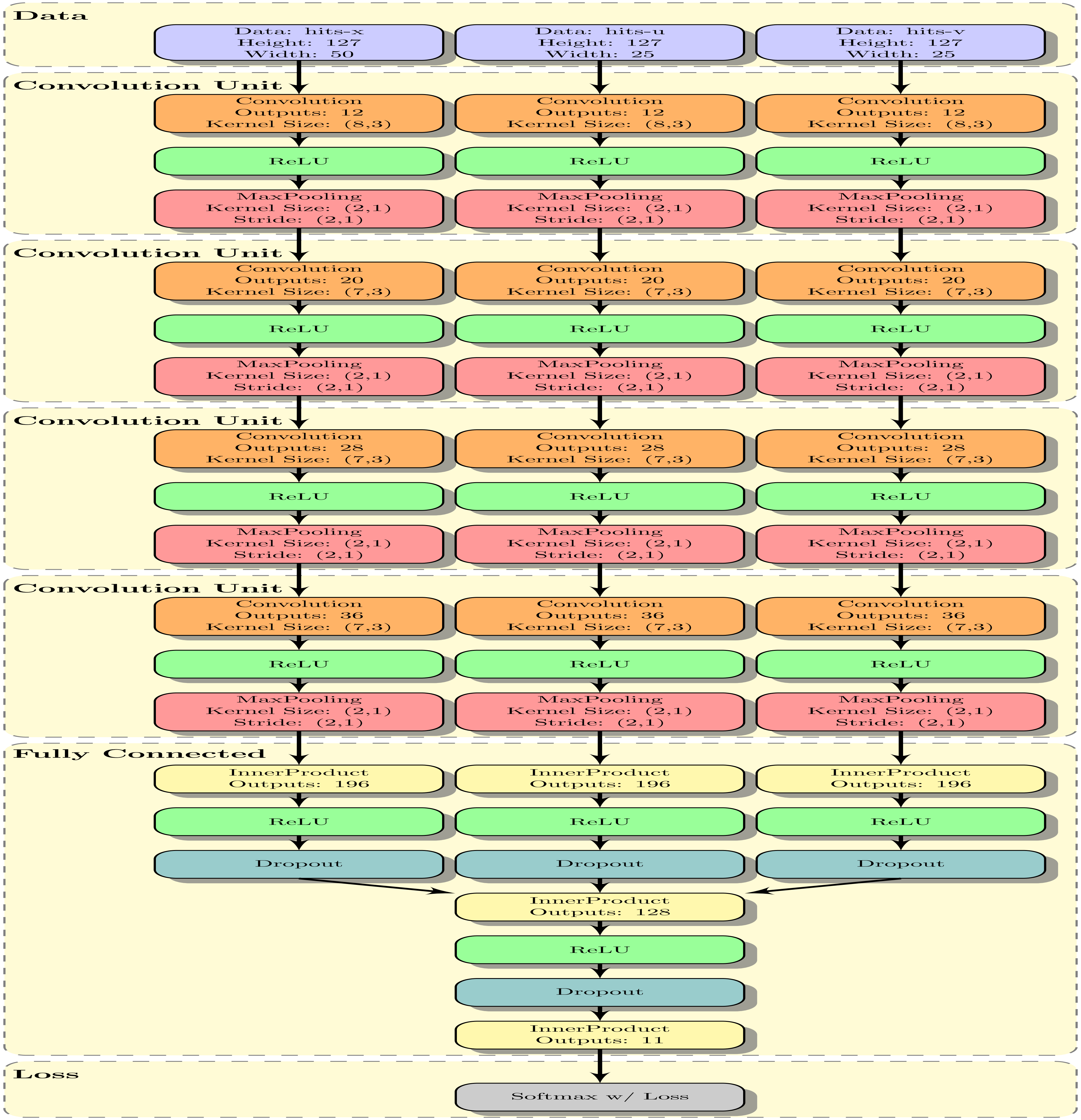
Stacking layers of convolutions leads from geometric / spatial representation to semantic representation





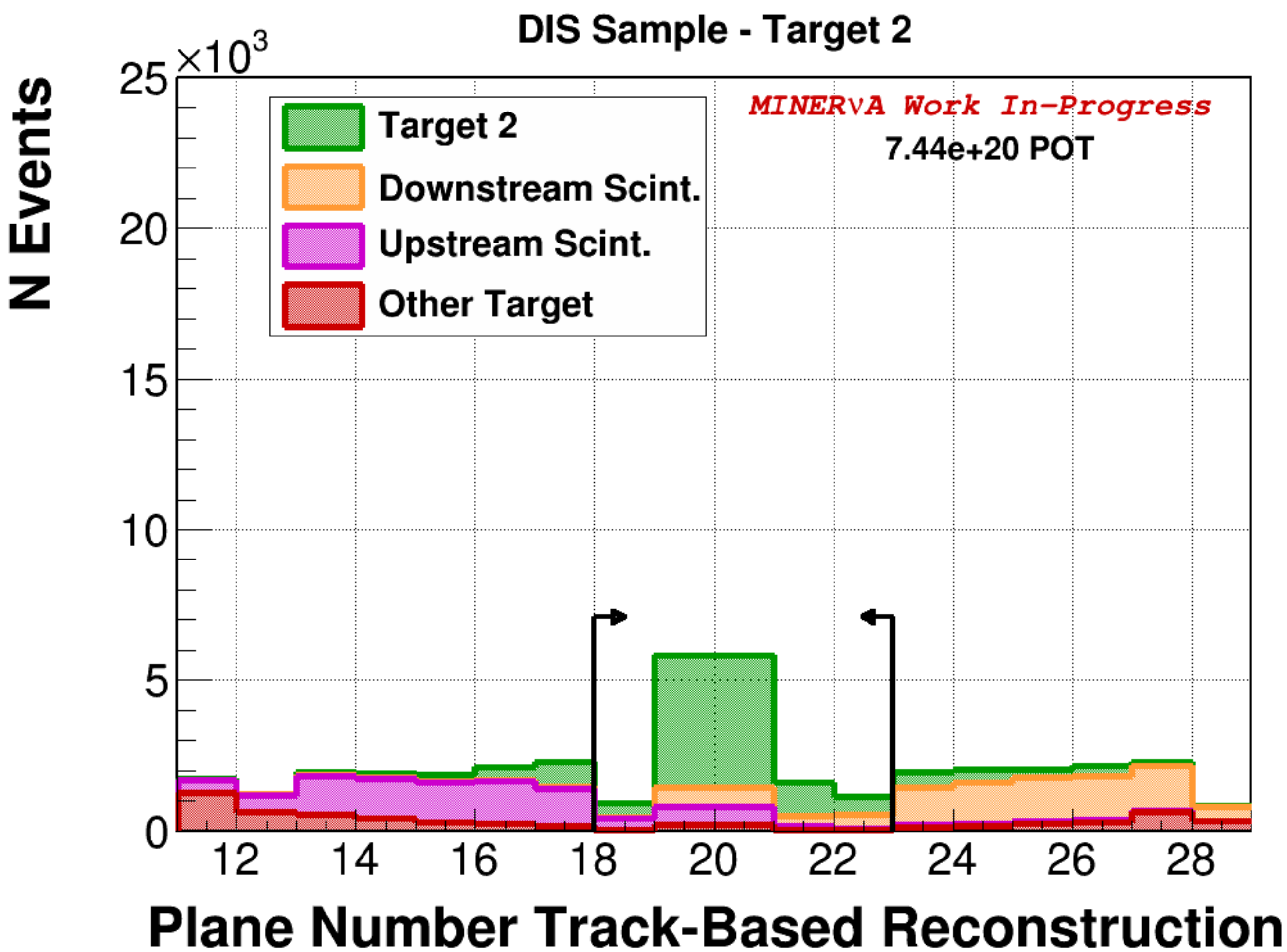
Each view represents a different angle of the interaction, hence, a pixel location in one view does not correspond to that same pixel location in another view

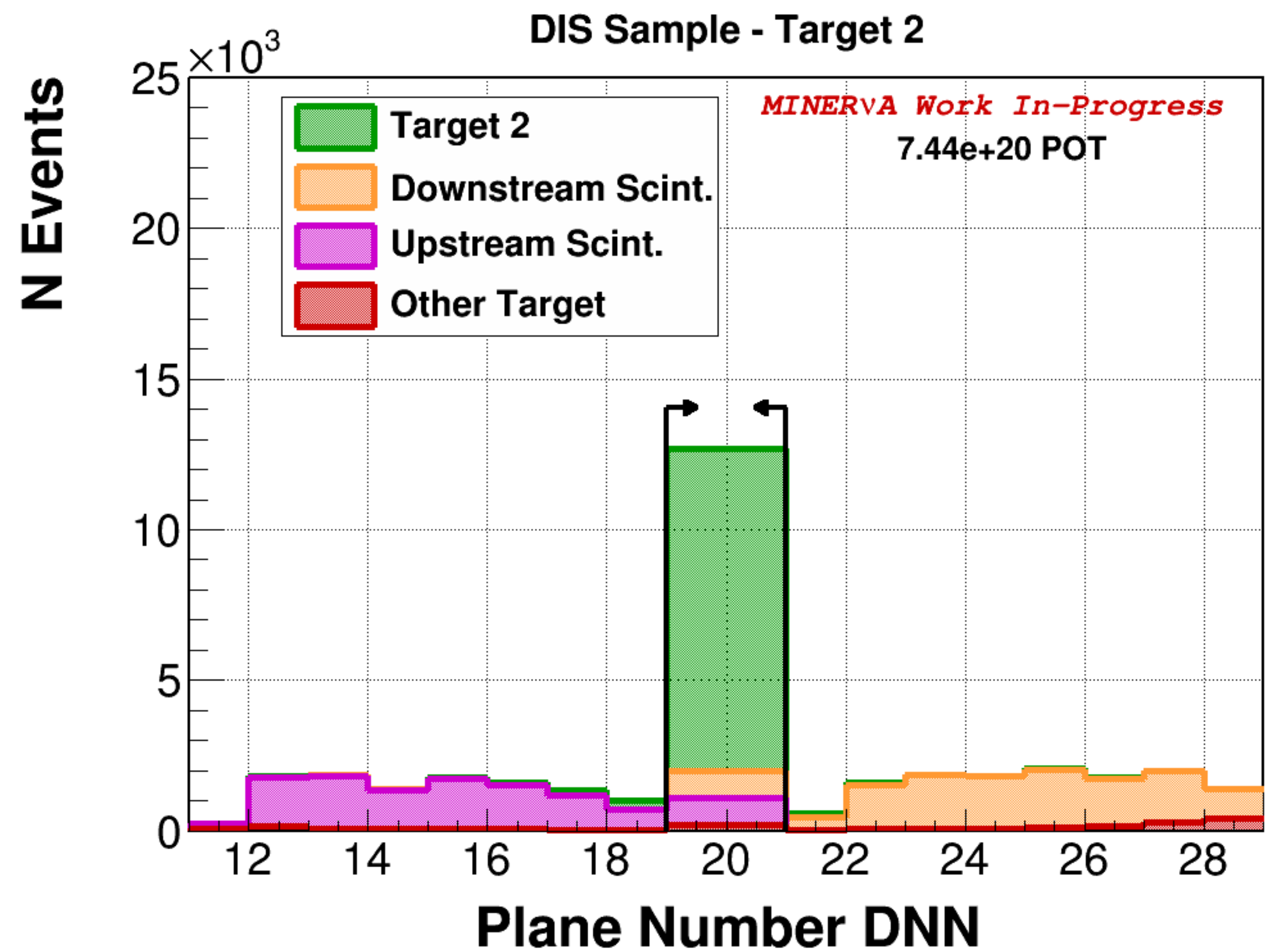
We have three separate convolutional towers that look at each of the X, U, and V images.



Data
Data: hits-x
Height: 127
Width: 50
Data: hits-u
Height: 127
Width: 25
Data: hits-v
Height: 127
Width: 25
Convolution Unit
Convolution
Outputs: 12
Kernel Size: (8,3)
ReLU
MaxPooling
Kernel Size: (2,1)
Stride: (2,1)
Convolution Unit
Convolution
Outputs: 20
Kernel Size: (7,3)
ReLU
MaxPooling
Kernel Size: (2,1)
Stride: (2,1)
Convolution Unit
Convolution
Outputs: 28
Kernel Size: (7,3)
ReLU
MaxPooling
Kernel Size: (2,1)
Stride: (2,1)
Convolution Unit
Convolution
Outputs: 36
Kernel Size: (7,3)
ReLU
MaxPooling
Kernel Size: (2,1)
Stride: (2,1)
Fully Connected
InnerProduct
Outputs: 196
ReLU
Dropout
InnerProduct
Outputs: 128
ReLU
Dropout
InnerProduct
Outputs: 11
Loss
Softmax w/ Loss

Track-based approach vs ML approach





Signal purity has been improved **by the factor of 2-3** using ML technique compared to track based approach

Be aware of Model bias!

- **Train and prediction in different domains**
  - Train with labeled data: in our case it is **Monte Carlo** (**source domain**)
  - Test with unlabeled data: in our case it is **real data** (**target domain**)

If Monte Carlo is in good agreement with data, life is good…. but

Our models are not perfect!

↓

Domain discrepancy arises

Find ways to reduce any biases in the algorithm that may come from training our models in one domain and applying them in another

Domain adversarial neural network may help us here!

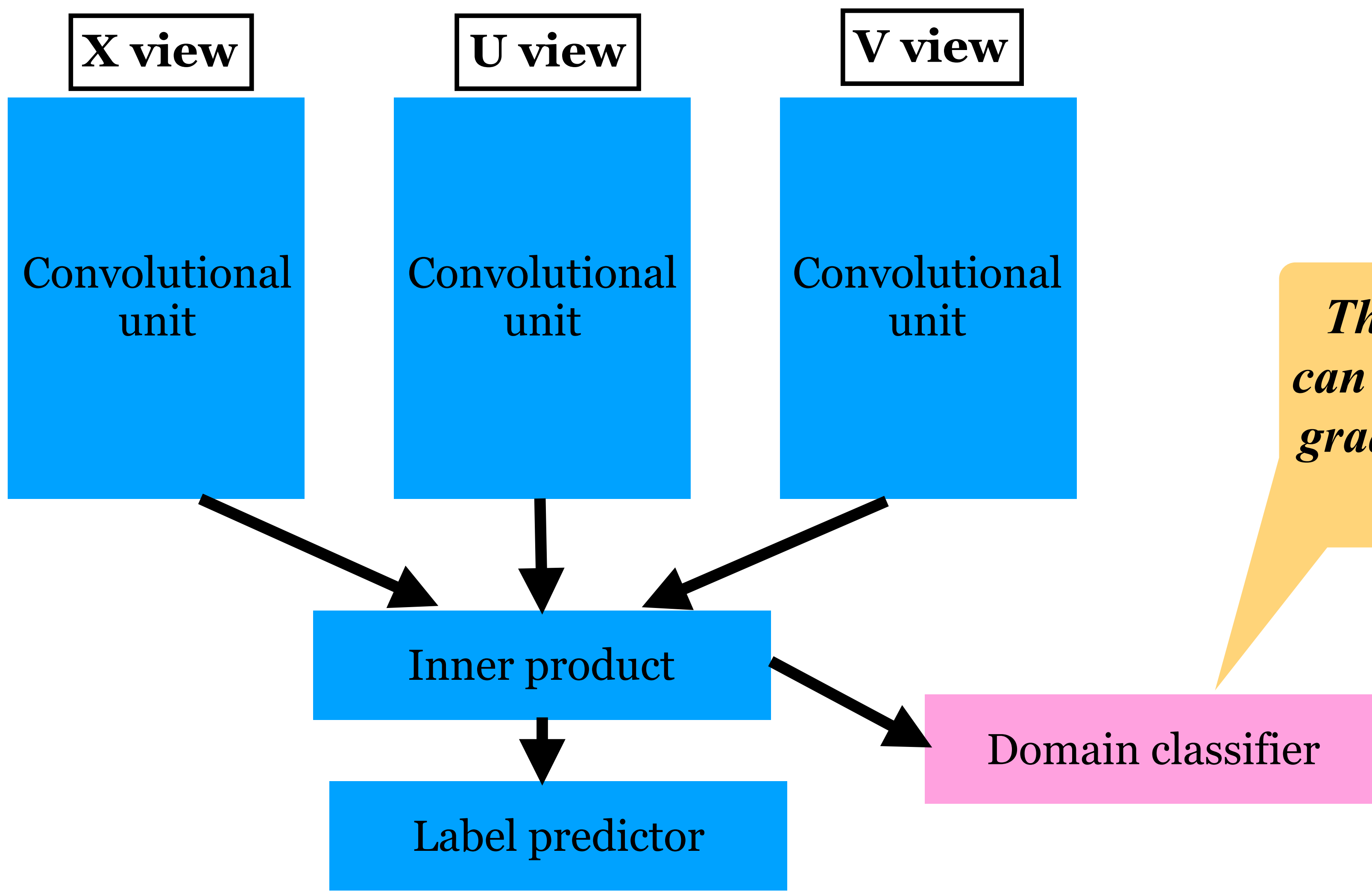
Domain Adversarial Neural Network (DANN)

http://adsabs.harvard.edu/cgi-bin/bib_query?arXiv:1505.07818

Train from the ***labeled source domain (MC)*** and ***unlabeled target domain (real data)***

Goal to achieve the features:

1) ***discriminative*** for the main learning task on the source domain
2) ***indiscriminate*** with respect to the ***shift between domains***

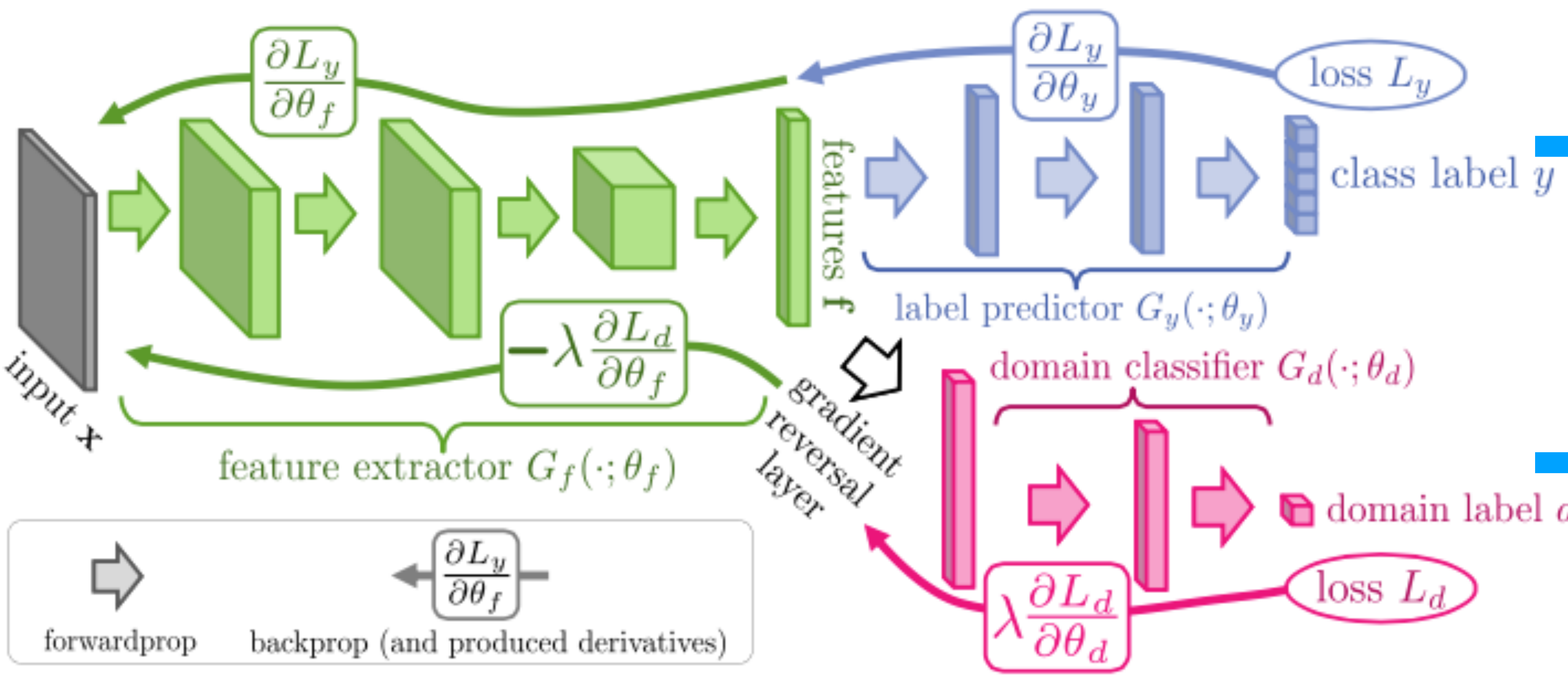


This adaptation behavior can be achieved by adding a gradient reversal layer with few standard layers

How does DANN work?

Minimize the loss of the label classifier so that network can predicts the input level

With DANN:
Two classifiers into the network



Label predictor: output

Domain classifier: works internally

Maximize the loss of the domain classifier so that network can not distinguish between source and target domain

The network develops an insensitivity to features that are present in one domain but not the other, and train only on features that are common to both domains

How to test DANN?

arXiv:1808.08332

- Find source and target with **distinct features**

our source and target domains may be **too similar for the domain classifier** to be able to distinguish between them.

- We train with Monte Carlo (MC) events and use different MC as target

We tried by few ways to get the target sample having different features than source: changing the flux, physics model, kinematic division etc

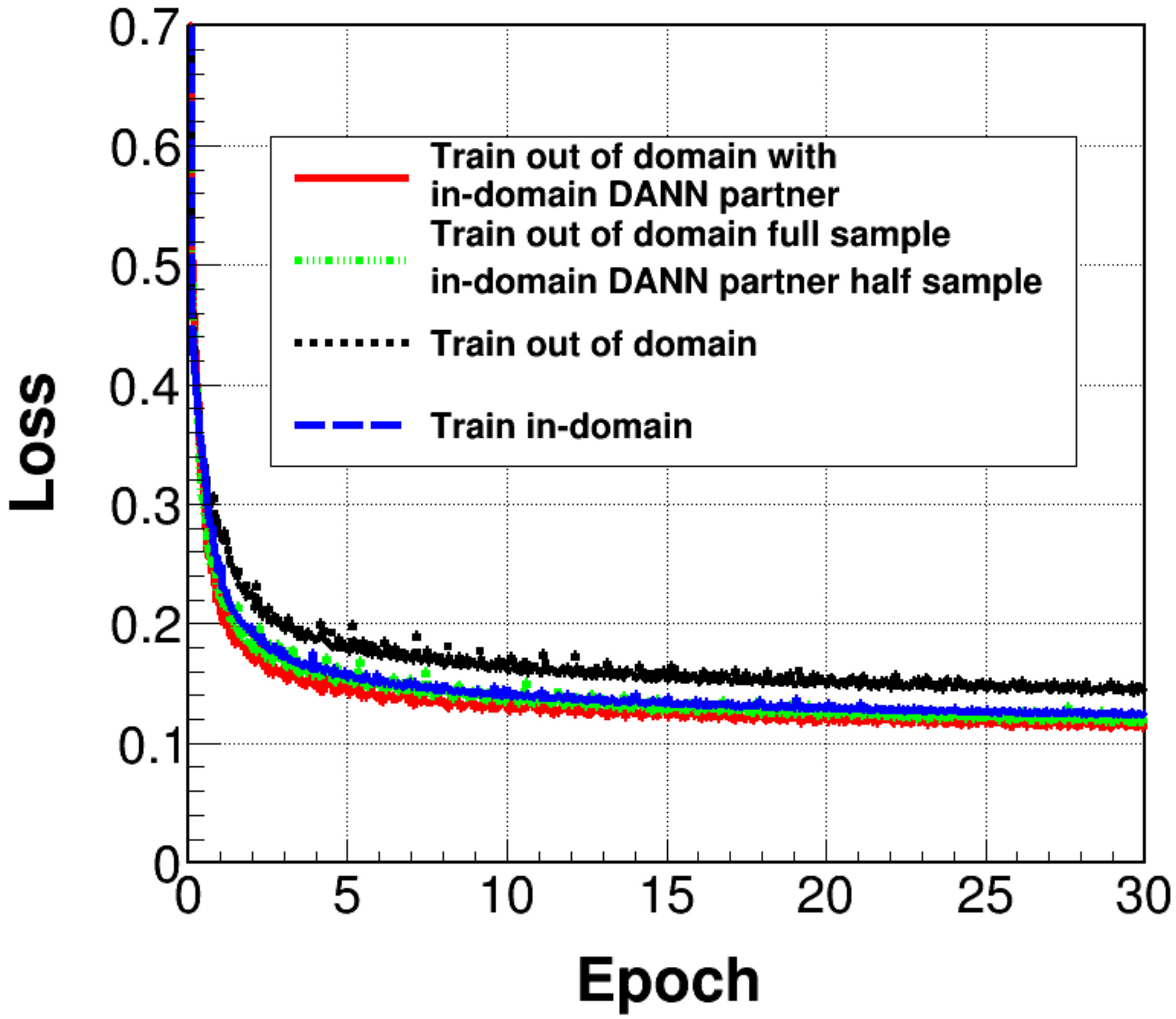
FIS On/Off

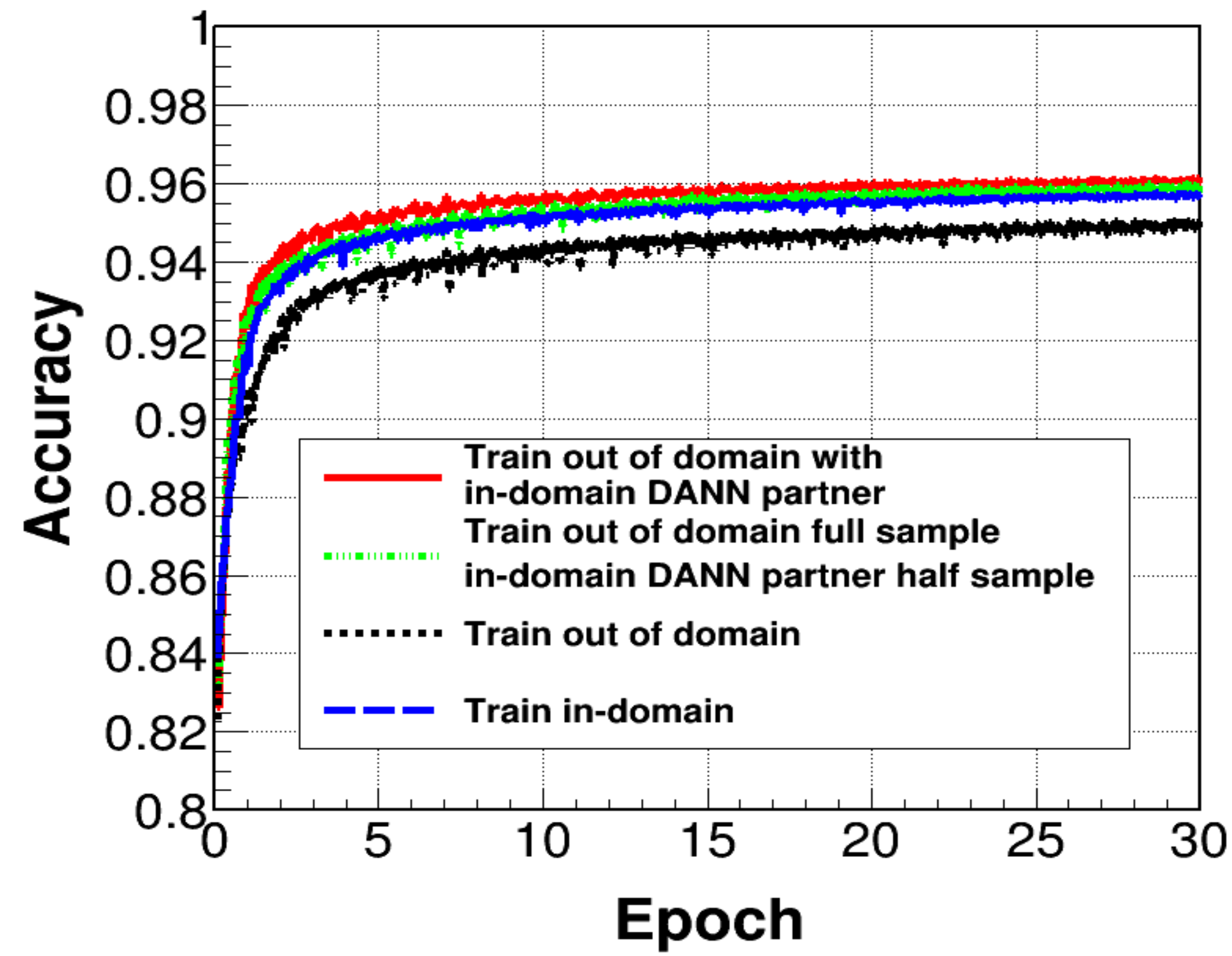
Training sample (Source domain)	DANN partner	Model
FSI off (1.2M)	N/A	Out of domain
FSI on(1.2M)	N/A	In domain
FSI off(1.2M)	FSI on(1.2M)	Out of domain with in domain DANN partner
FSI off(1.2M)	FSI on(0.6M)	Out of domain with in domain DANN partner(half sample)

The expectation: *CNN in domain* will perform *better* than ***CNN out of domain***.

Check the effect of size of training sample on the performance of the model

The expectation: though model is trained "out of domain", it would show the ***similar performance*** as ***"CNN in domain"*** since we consider ***"in domain" DANN***





Red curve: Adding a DANN partner to the model trained in the out-of domain we are able to ***recover the performance*** of the model natively trained in the correct domain

Green curve:
Perform worse than red curve as the sample size is reduced by half
Perform better than than black curve as it has information ***from the correct domain***

DANN helps to recover the domain information

Vertex reconstruction :MENNDL

- MINERvA and NOvA have jointly worked on **an automated NN-architecture optimization problem** with computer scientists at at Oak Ridge National Laboratory (ALCC project)

- We use an evolutionary algorithm - MENNDL(**Multi-node Evolutionary Neural Network for Deep Learning**) - to evolve neural network topologies using the Titan supercomputer.
  - Evolutionary algorithm as a solution for searching hyper-parameter space for deep learning
  - Leverage more GPUs; ORNL's Titan has 18k GPUs

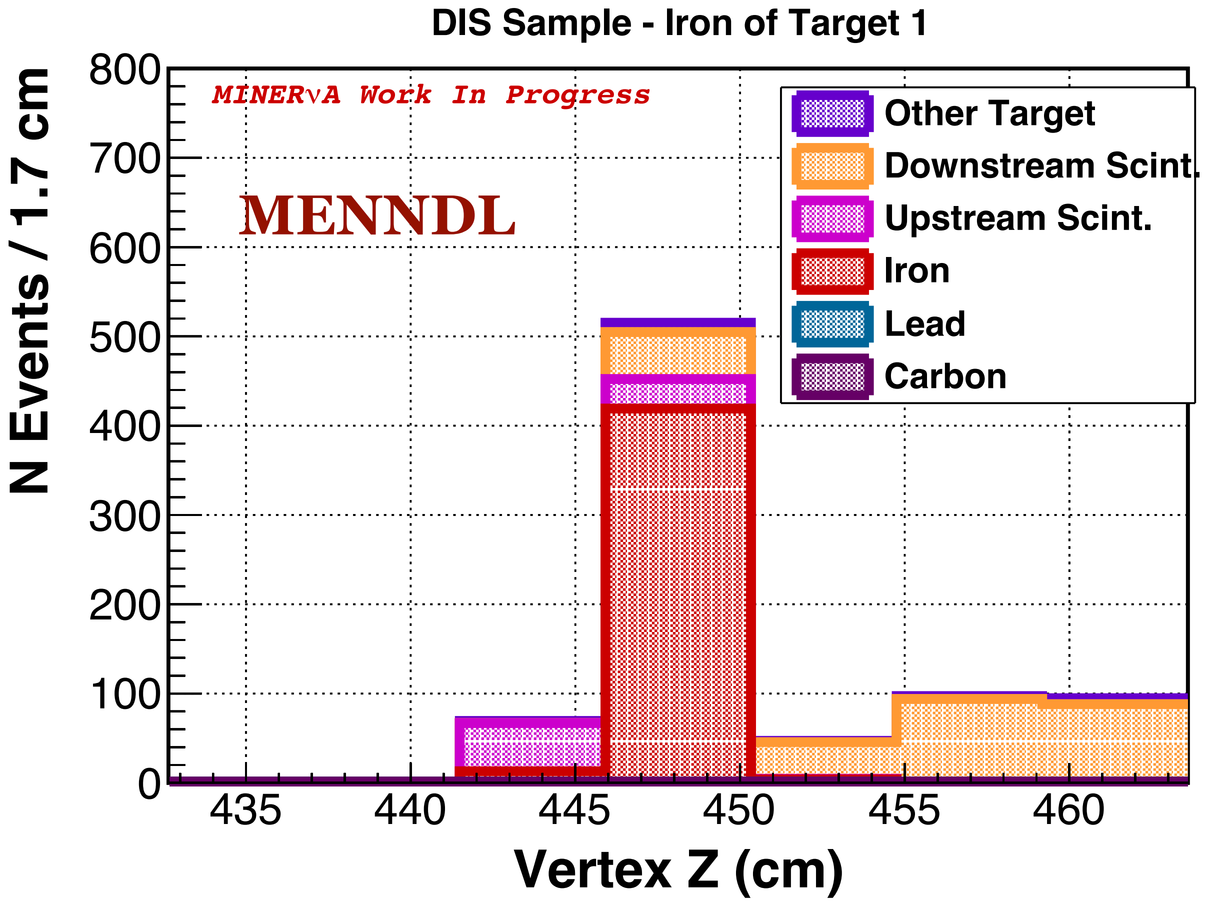
    -Next generation, Summit, will have increased GPU capability

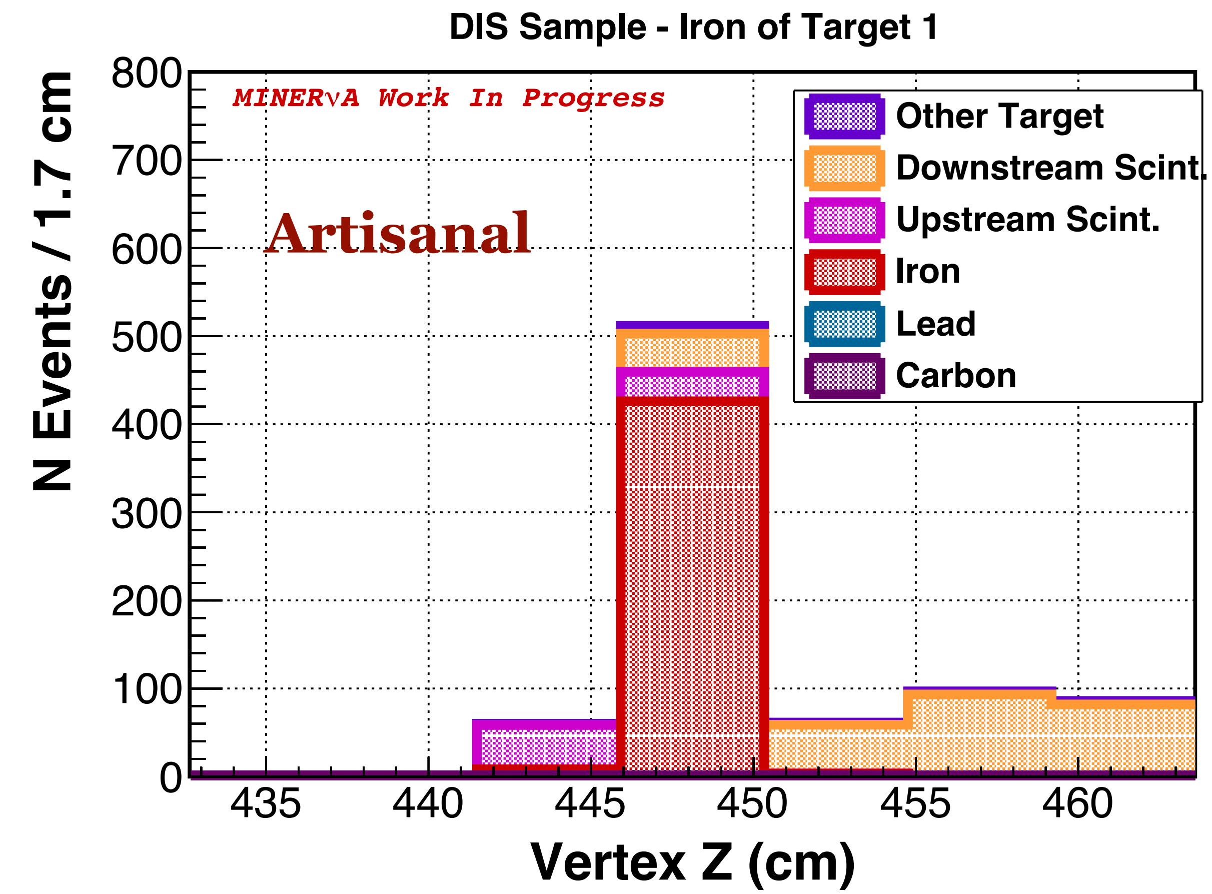
☑ We obtained model from MENNDL for vertex finding model.

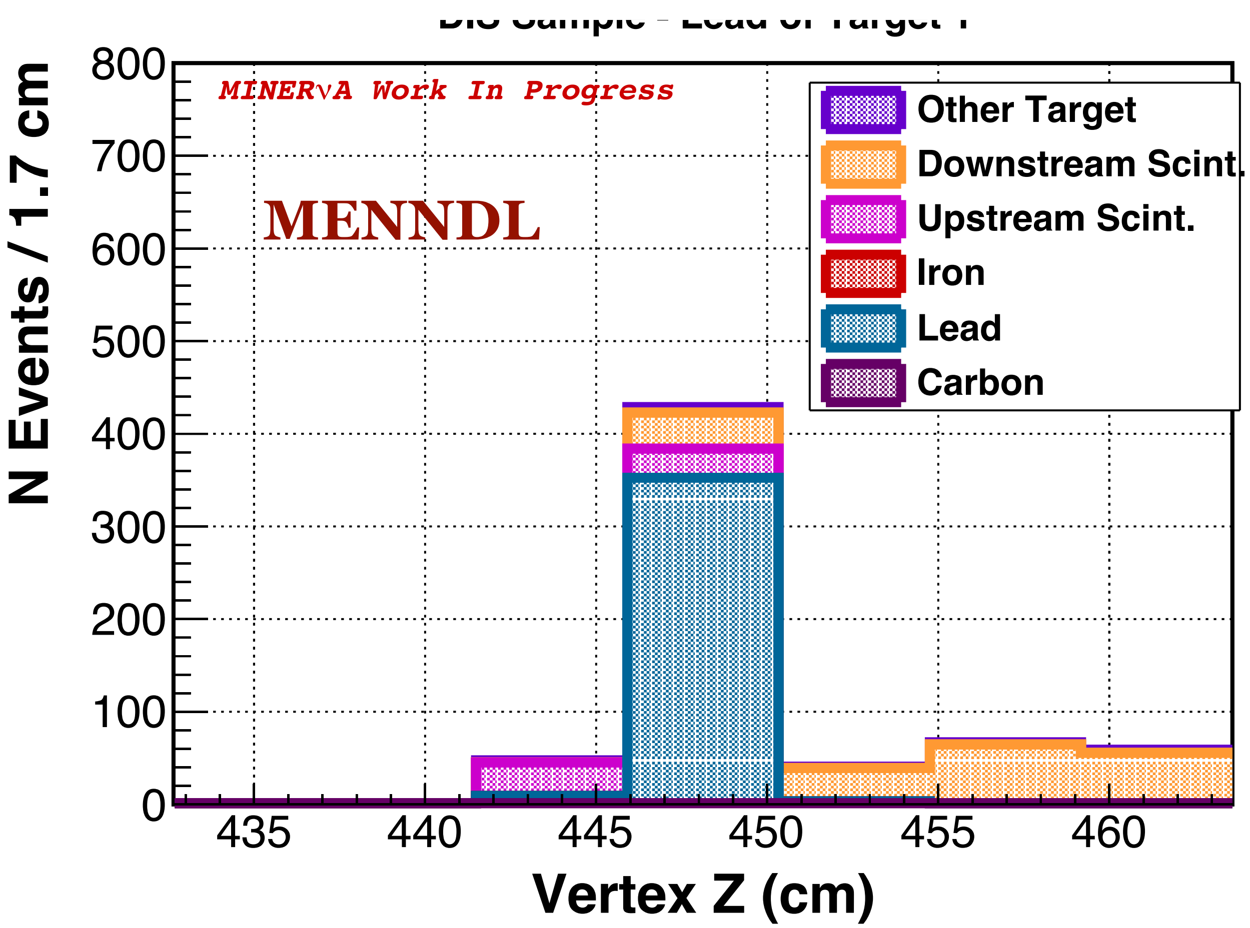
☑ We compare the performance of MENNDL model with the "hand craft" or artisanal model developed inside MINERvA

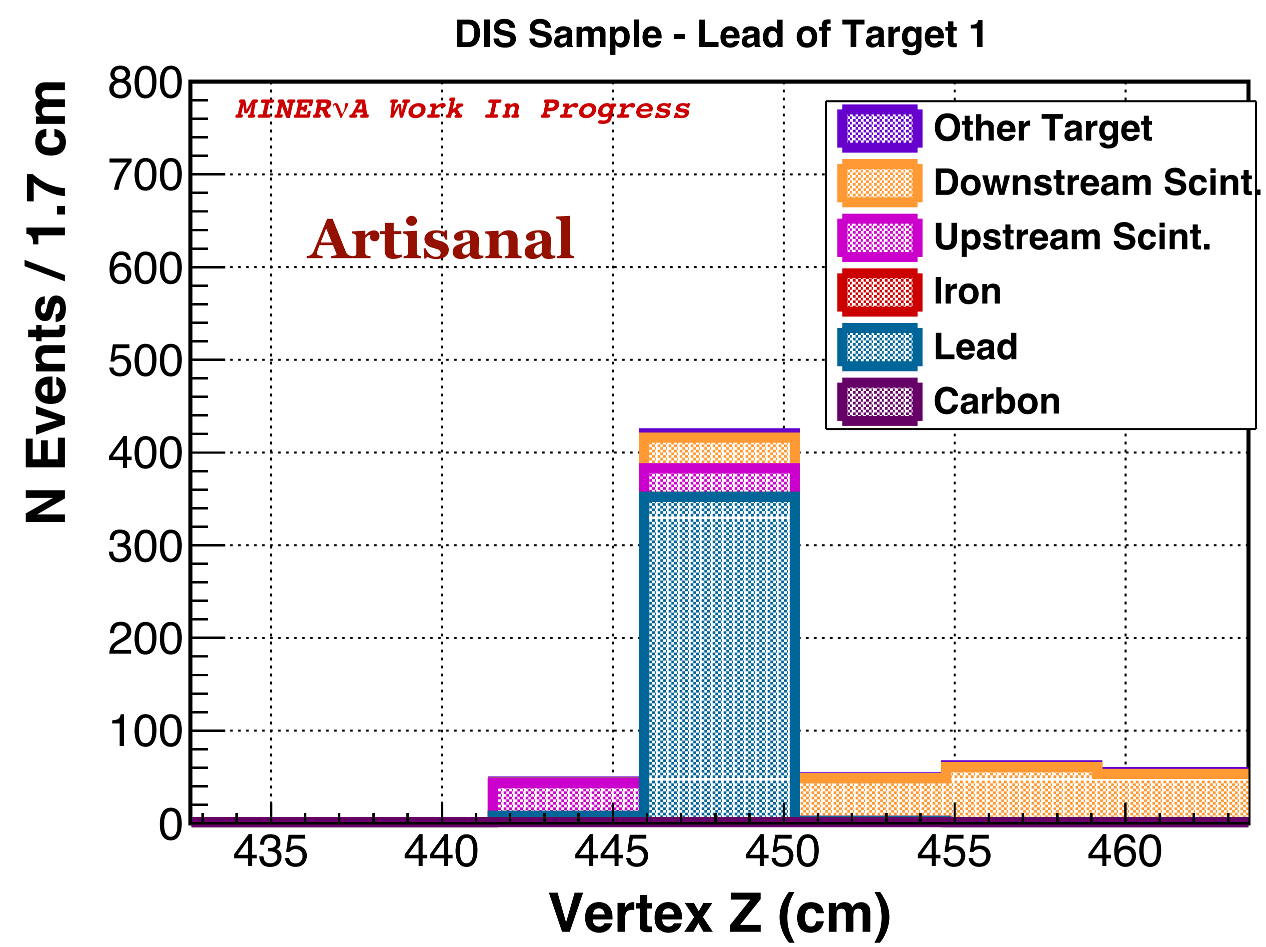
the MENNDL models are optimized over short run (5000 iteration)

Reconstructed vertex distribution from Anti-neutrino events









ML: Hadron multiplicity at final state

Goal: Get the prediction for number charged hadrons at the final state using Machine learning algorithm as it is difficult to do in track-based method.

Train the network

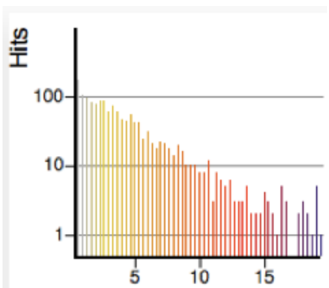
- Training is done with Charge current inclusive sample

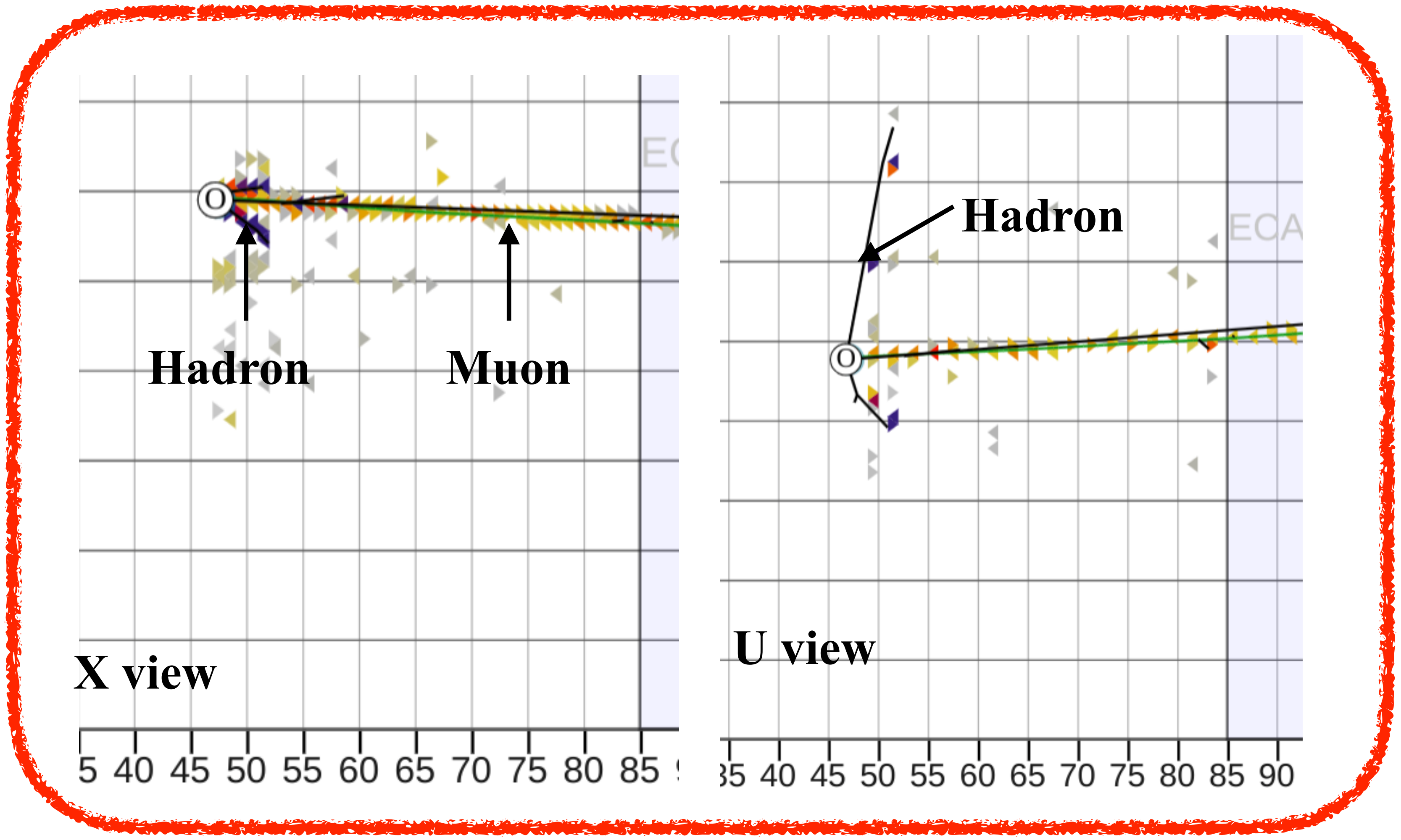
- We count proton, pi+, pi-, K+, K- as number of charged hadrons with kinetic energy **above 50 MeV**. Neutron, photon, neutral pions are not included.

Application of the trained model

- We applied ML prediction for hadron multiplicity in **low recoil analysis.** Events CC inclusive events where reconstructed three momentum transfer is between **0.4 to 0.8 GeV**.

Hadron Multiplicity: Challenge to ML





Final State:

- μ^- ▸ KE=7011.4 MeV
- p ▸ KE=49.7 MeV
- p ▸ KE=312.4 MeV
- p ▸ KE=231.2 MeV

Module number

For traditional tracking algorithms, threshold is **100 MeV** when the hadron is going forward

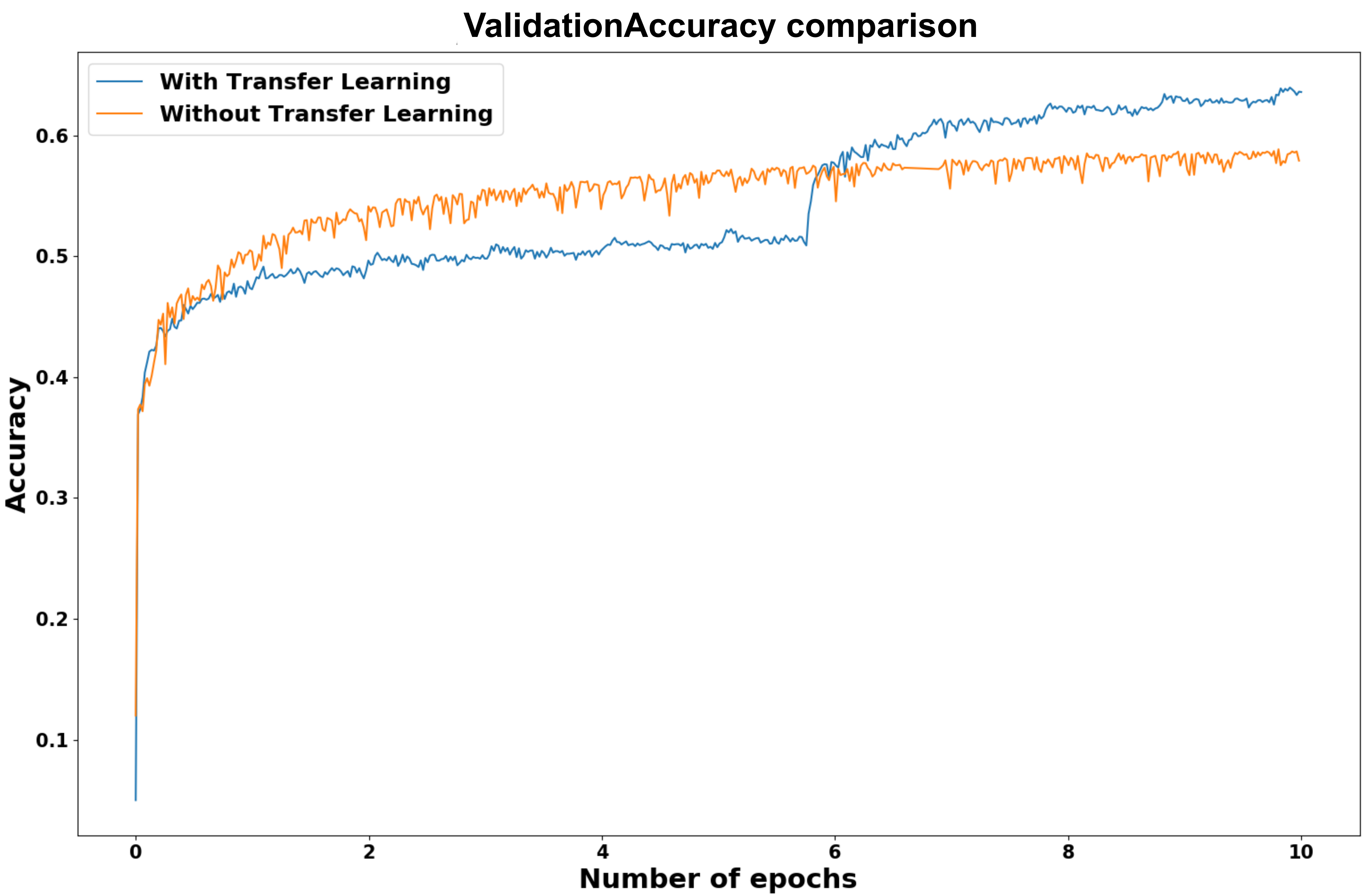
One proton <50 MeV, another proton is at high angle, tracker can not recognize it, can ML?

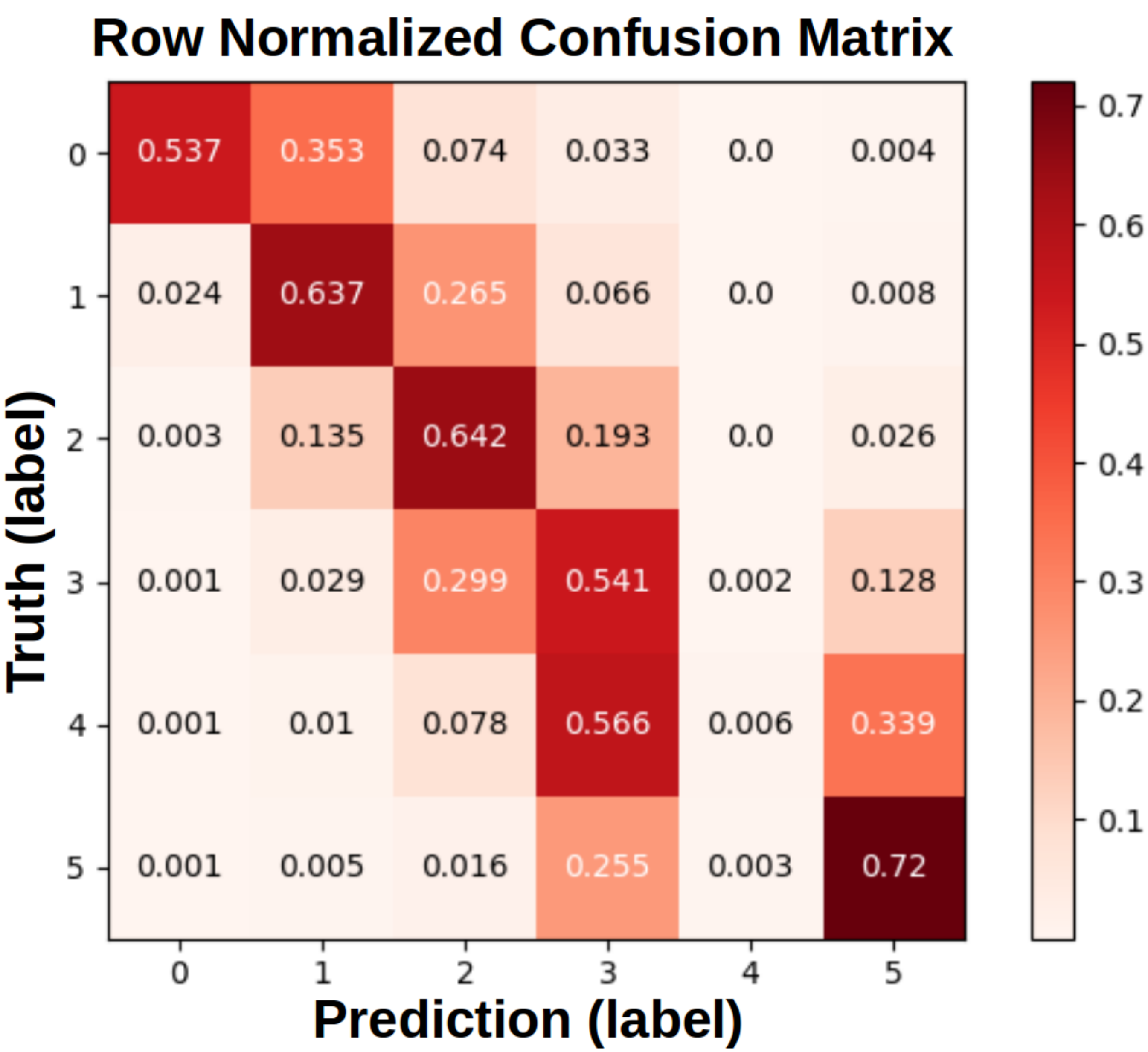
Implementing the transfer learning

Transfer learning is a machine learning method where a model developed for a task (e.g., vertex finding) is **reused as the starting point** for a model on a second task (e.g., hadron multiplicity)

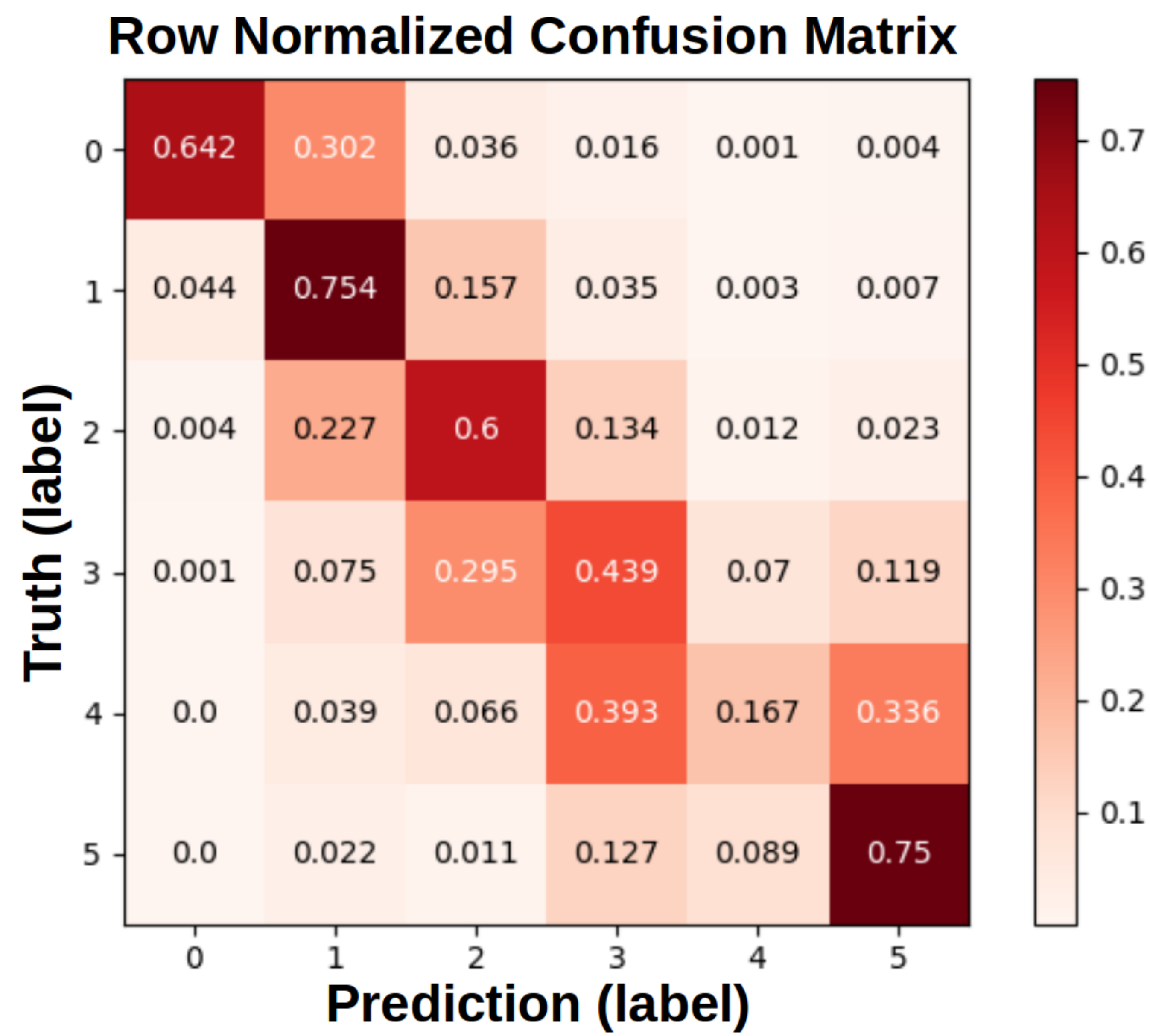
- MENNDL based vertex finding model is trained over large Monte Carlo Sample. This model is reused as starting point for a model of hadron multiplicity
- **Freeze the weights** of the convolutional layers and run upto few epochs for hadron multiplicity
- Next **unfreeze the weights** of the convolutional layers and run up to another few epochs: **fine tuning**

Validation accuracy

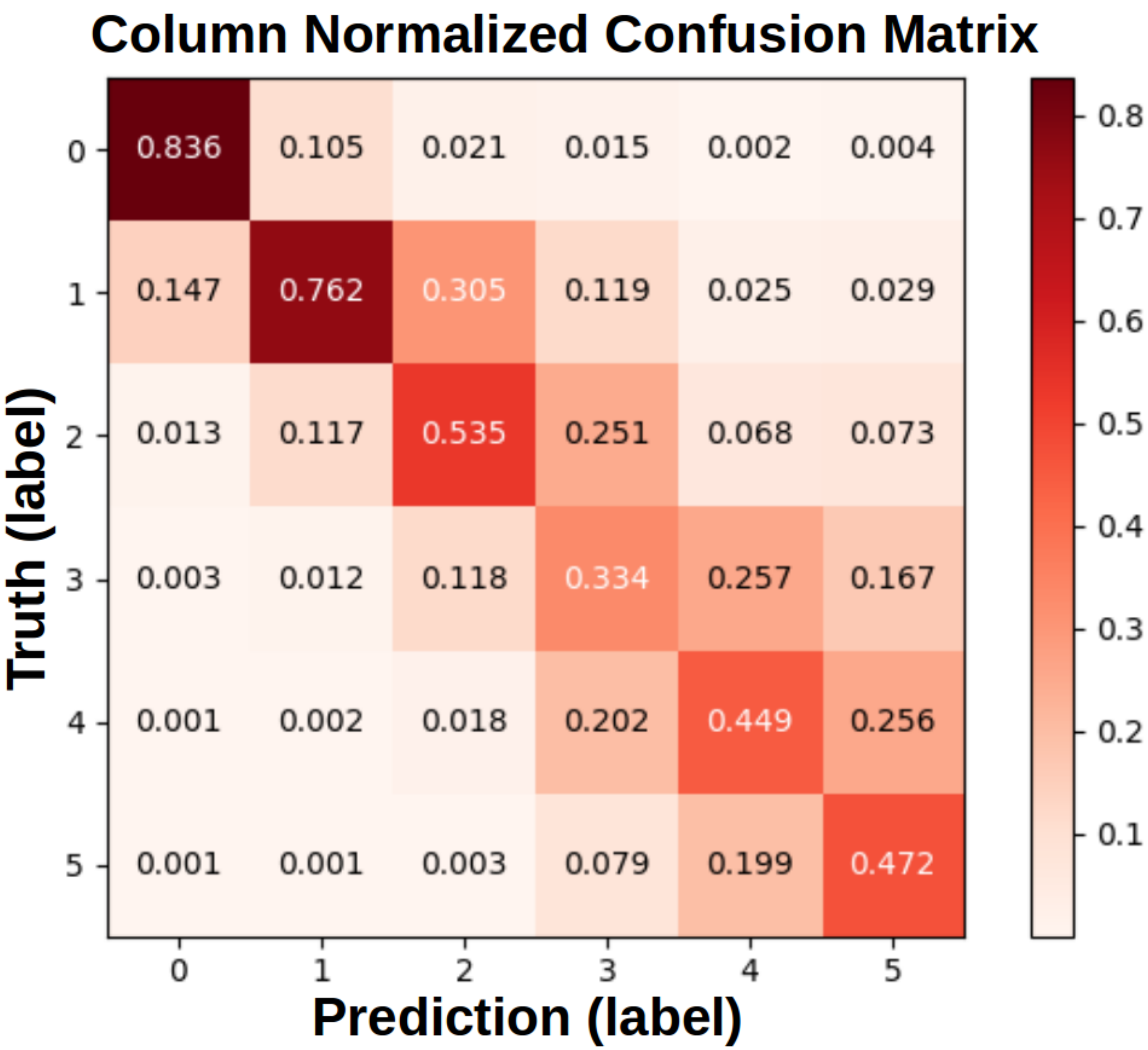




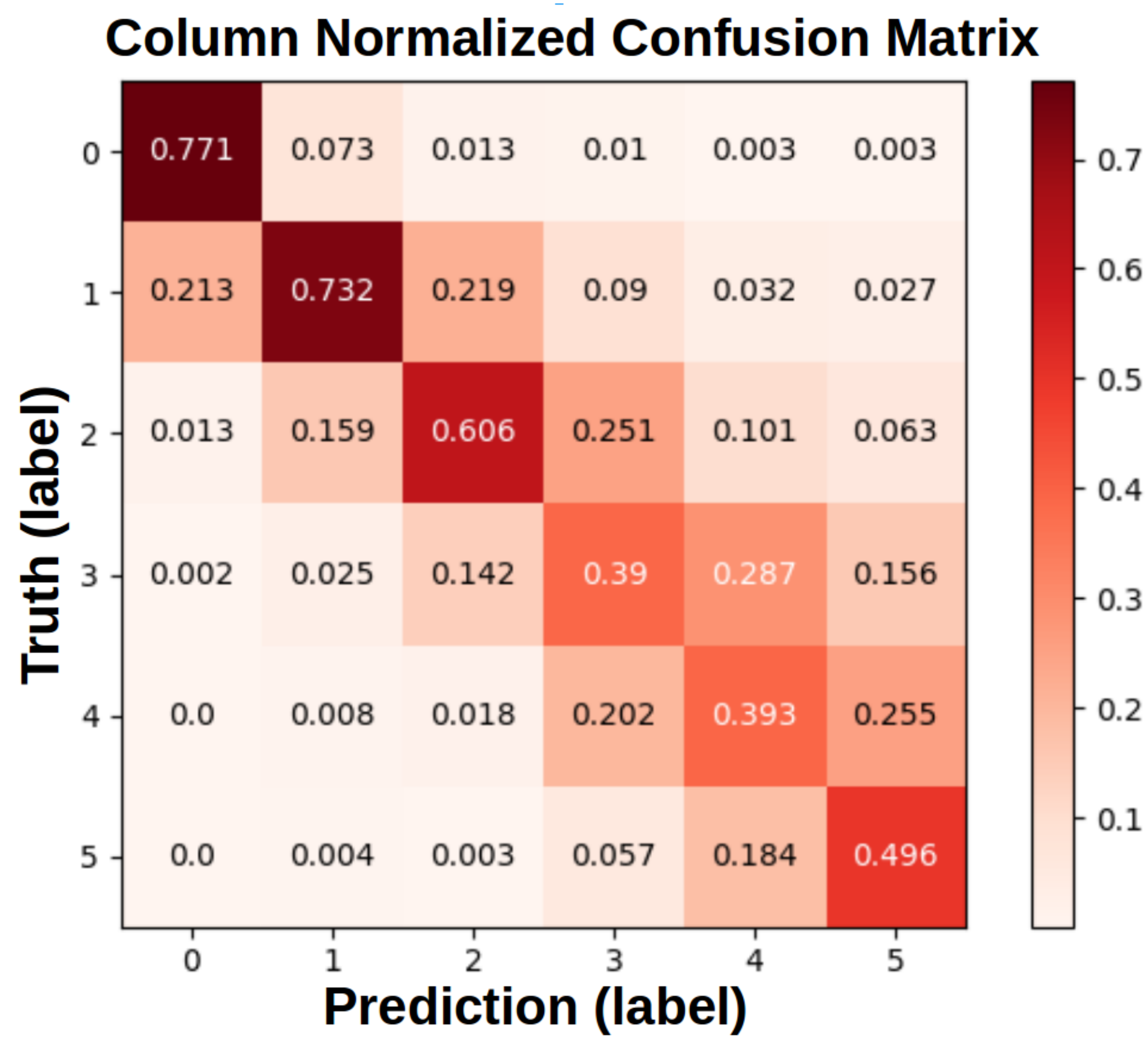
Without Transfer learning



With Transfer learning

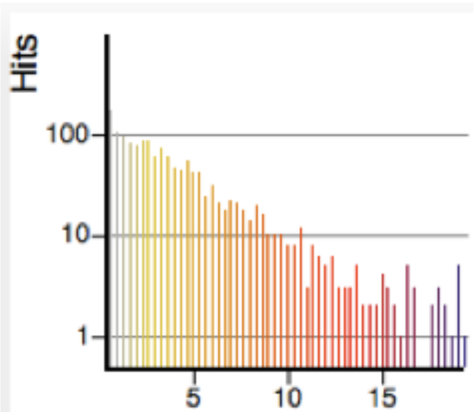


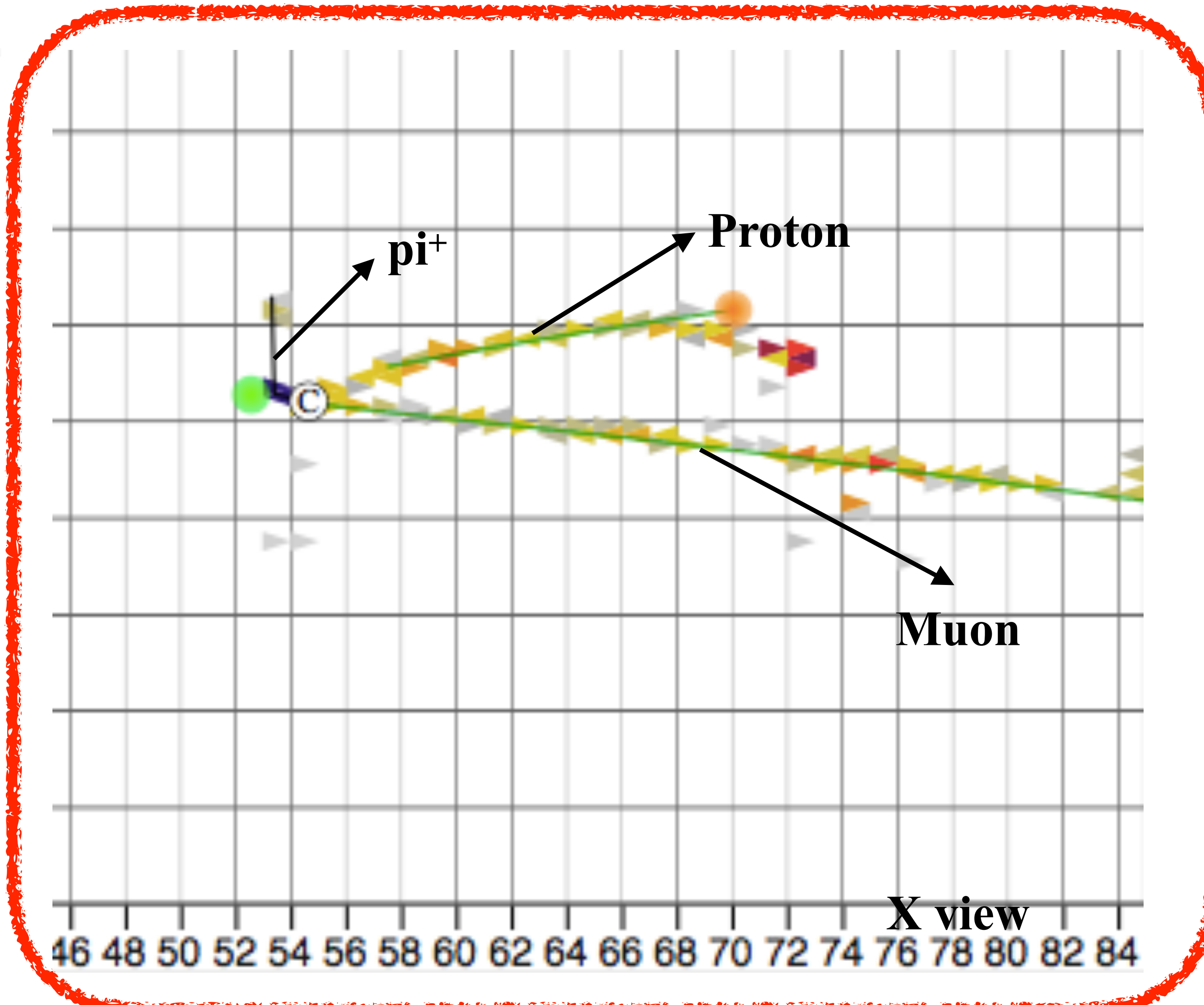
Without Transfer learning



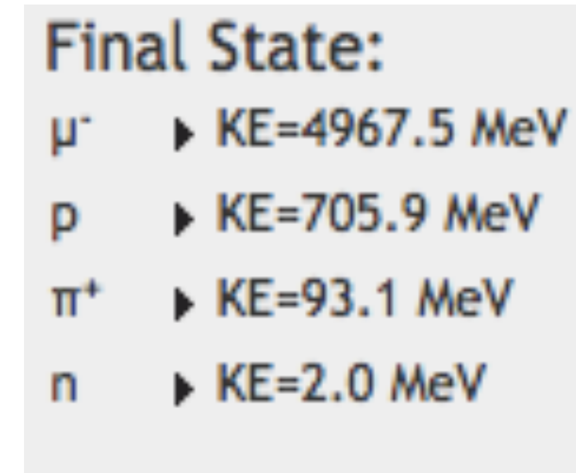
With Transfer learning

Event classified by track-based and ML-based method





Module number



Track-based method **recognized** the hadron with **705 MeV** but **can not find** the hadron with kinetic energy **93 MeV**

ML-based method has been able to find both the hadrons

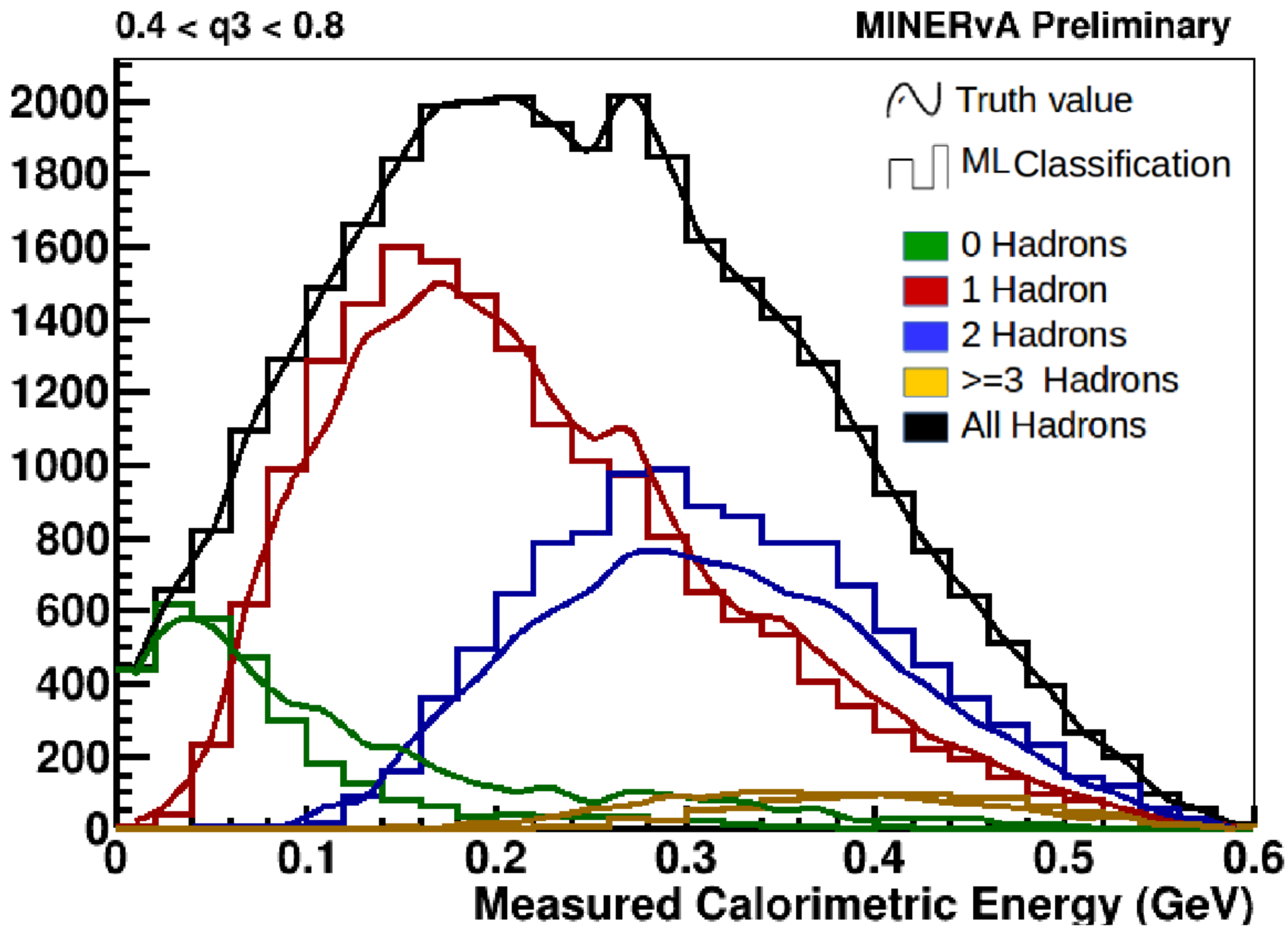
Comparisons among Truth, Track-based and ML-based

Threshold 100 MeV

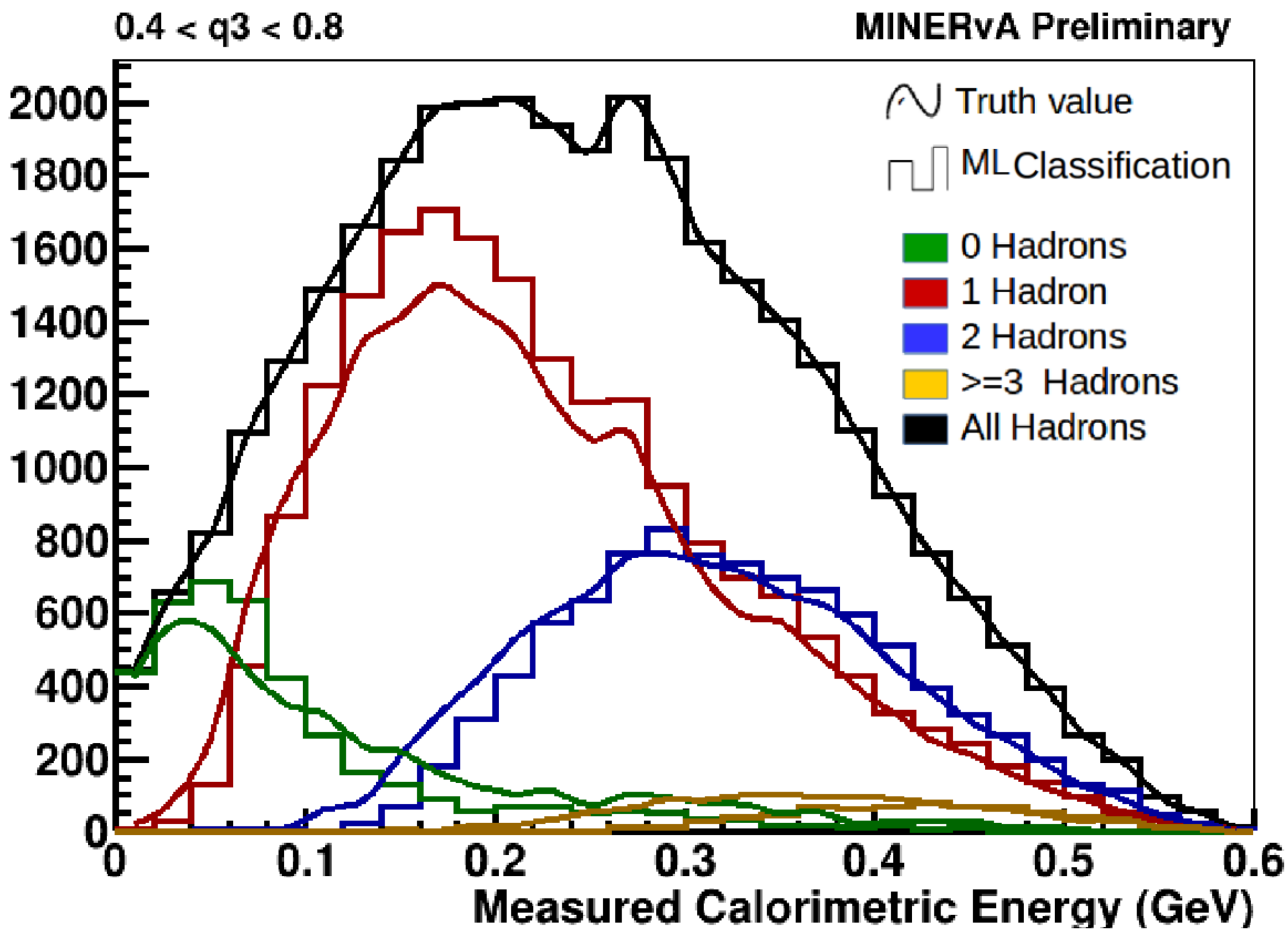
Threshold 50 MeV

Number of hadrons	Truth Value	Track-based	ML without transfer learning	ML with transfer learning
0	4773	23673	4415	4473
1	20302	12612	19928	21809
2	10619	899	12804	10283
>=3	1364	88	1080	794

Event distribution for different hadron multiplicity



Event distribution for different hadron multiplicity



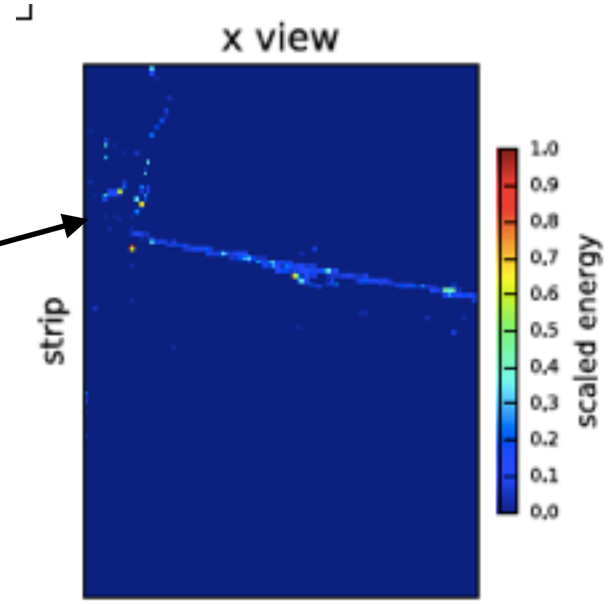
Neutral pion reconstruction using Semantic Segmentation via U-Resnet

Semantic segmentation

we deal our vertex fining problem by performing image classification.

classifying what appears in an image into one out of a set of predefined classes

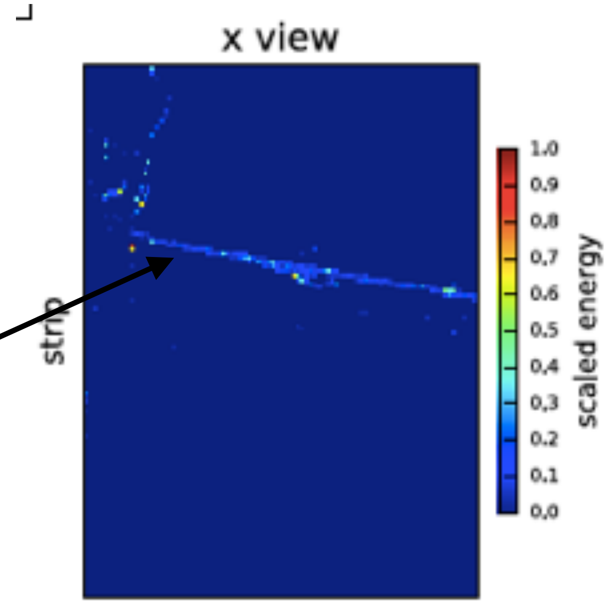
classifying the region of interaction vertex



Semantic segmentation is understanding an image at pixel level i.e, we want to assign each pixel in the image an object class

image semantic segmentation is the task of classifying each pixel in an image into one out of a set of predefined classes

classify each pixel - is this pixel, belongs to muon/proton/pion….?



Data format and algorithm

Image processing is done by **LarCV** package(https://github.com/DeepLearnPhysics/larcv2.git), developed by Kazuhiro Terao et al.

- An analysis framework developed for processing LArTPC image data this software is generic use by supporting C++ data structures, IO interface, and data processing machinery.
- It is used to directly manipulate the image data with or without OpenCV.
- It plays a key role to interface with open source deep learning softwares including

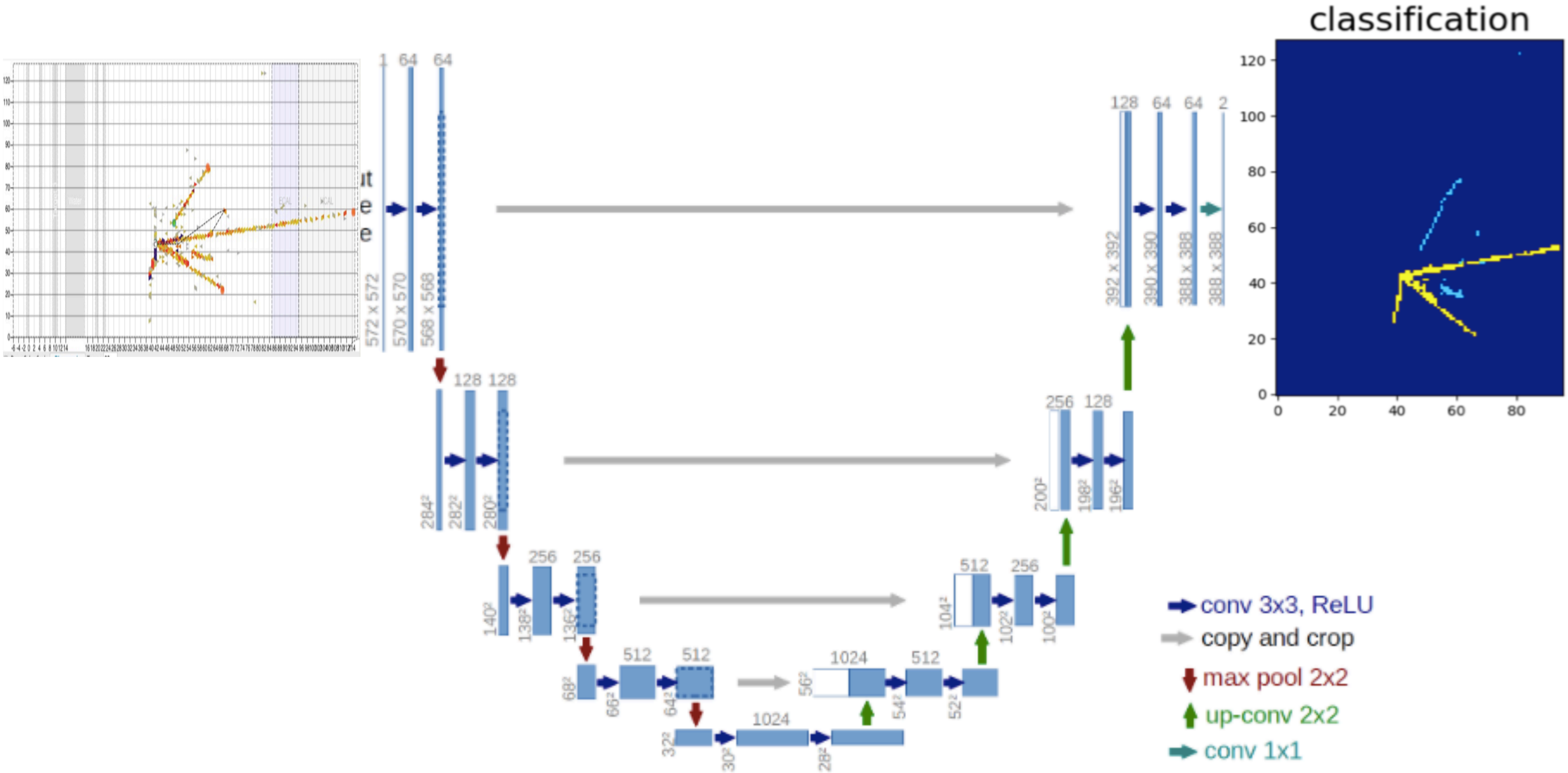
caffe by Berkeley Lab. and TensorFlow by Google.

- ROOT format- easy to handle, can do other things like crop images , resize images etc

U-resnet(https://github.com/DeepLearnPhysics/u-resnet.git) is used to implement the semantic segmentation algorithm

- Hybrid of the U-Net(arxiv: 1505.04597) and residual network (arxiv: 1512.03385, arxiv:1603.05027)

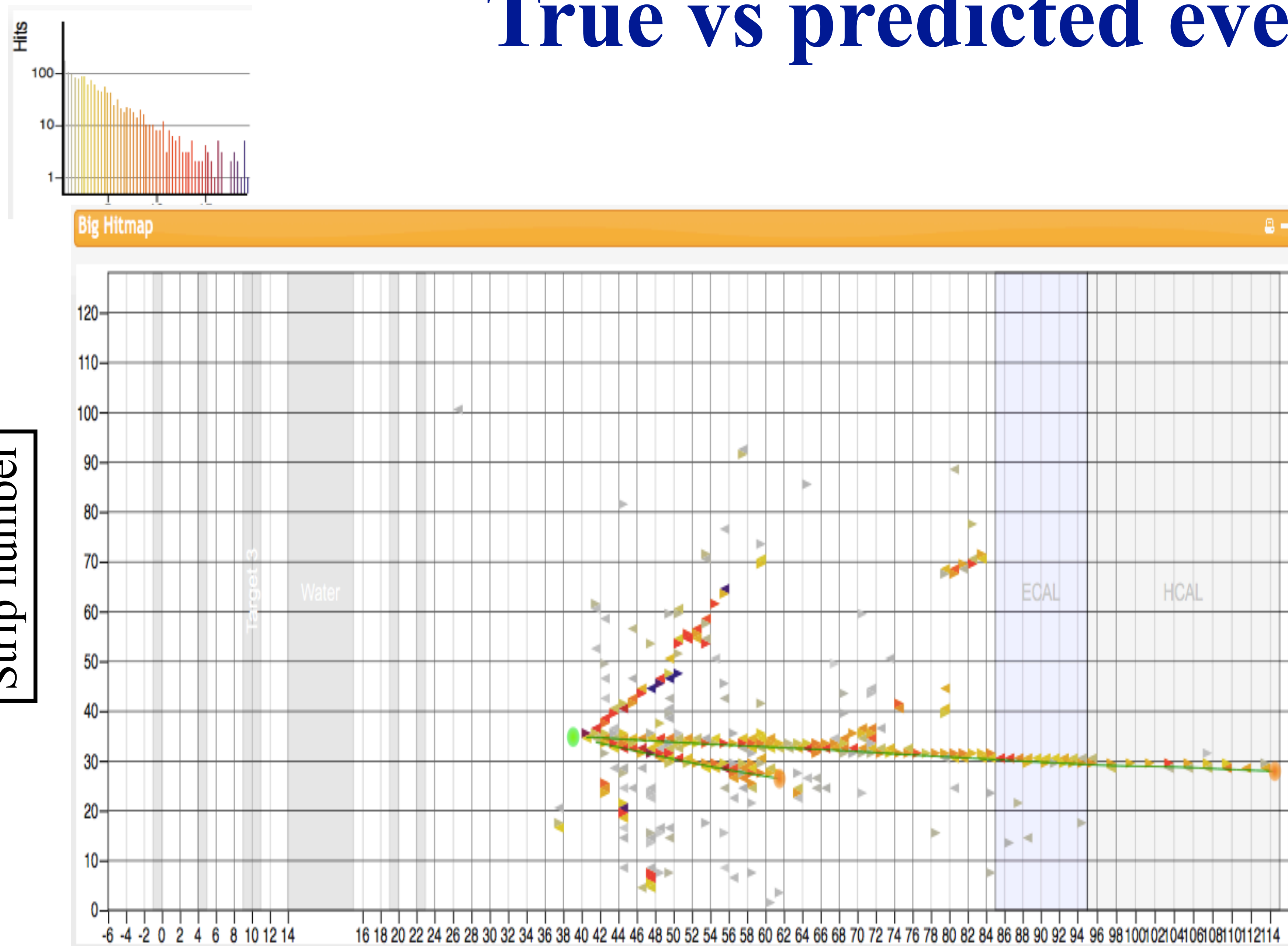
U-Resnet



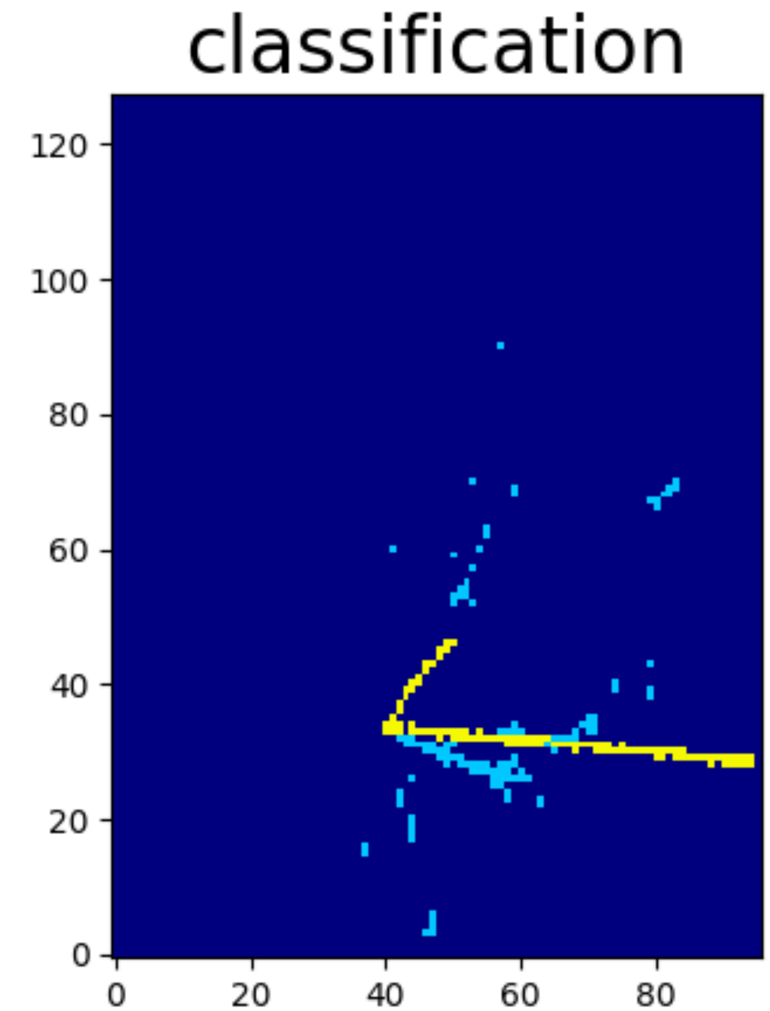
Goal: Tag the electromagnetic-like(em-like), non-emlike pixel.

True vs predicted event

Strip number

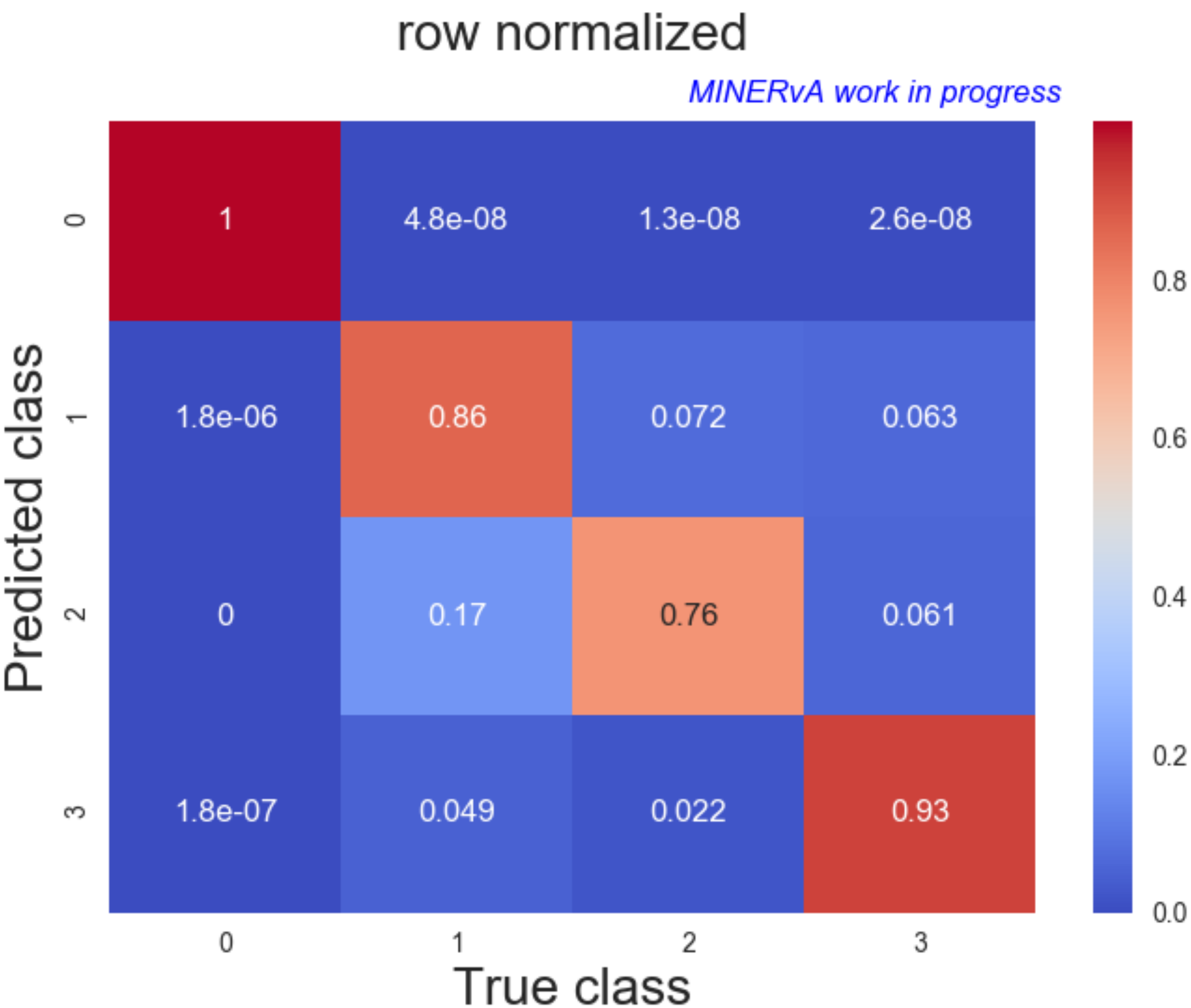


Module number



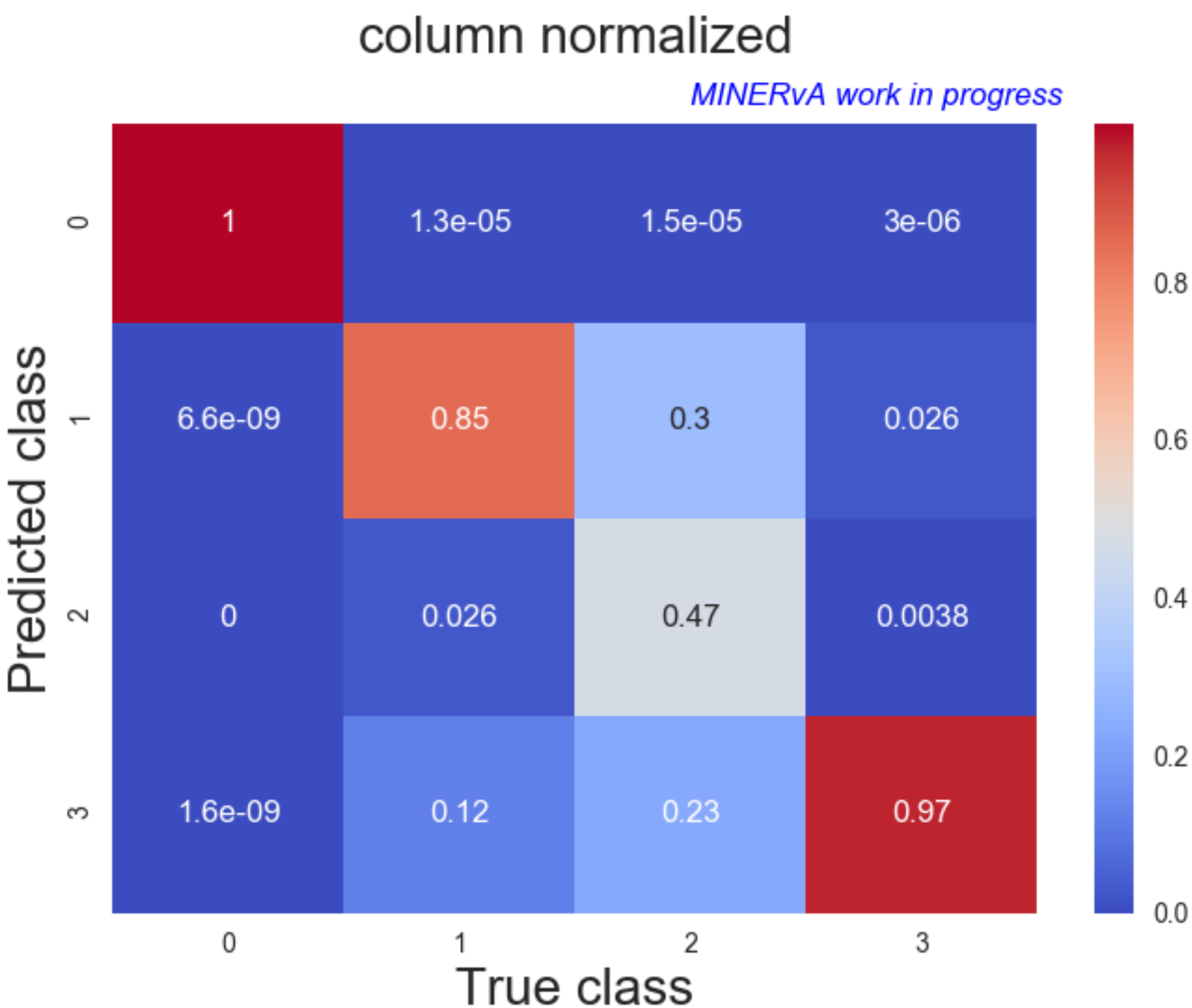
EM like: light blue
NON-EM like: yellow

Confusion matrix



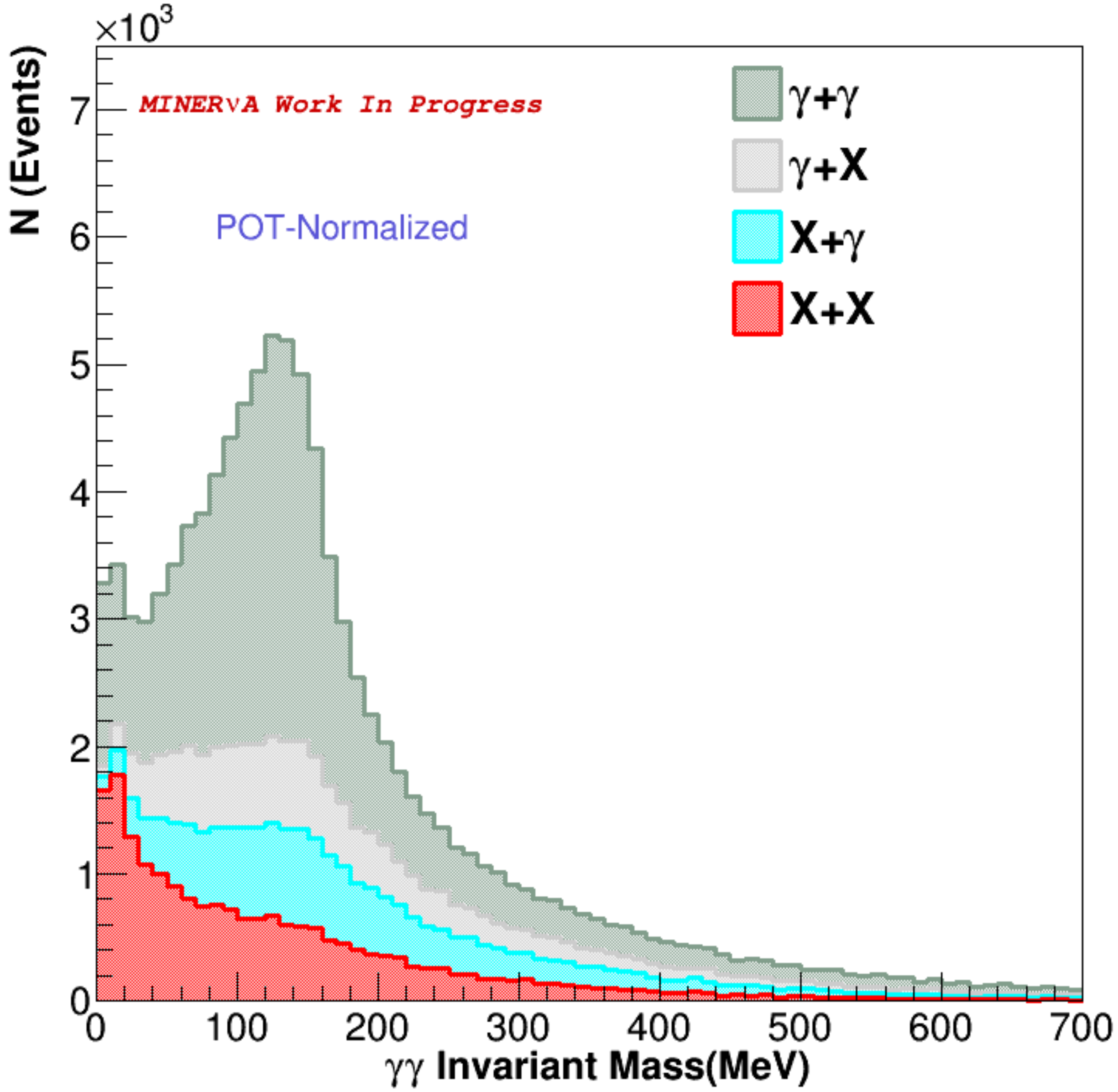
Class	Diagonal value from row normalized matrix
0	1
1 (EM-like)	0.86
2(Neutron)	0.76
3(Non-EM like)	0.93

Confusion matrix

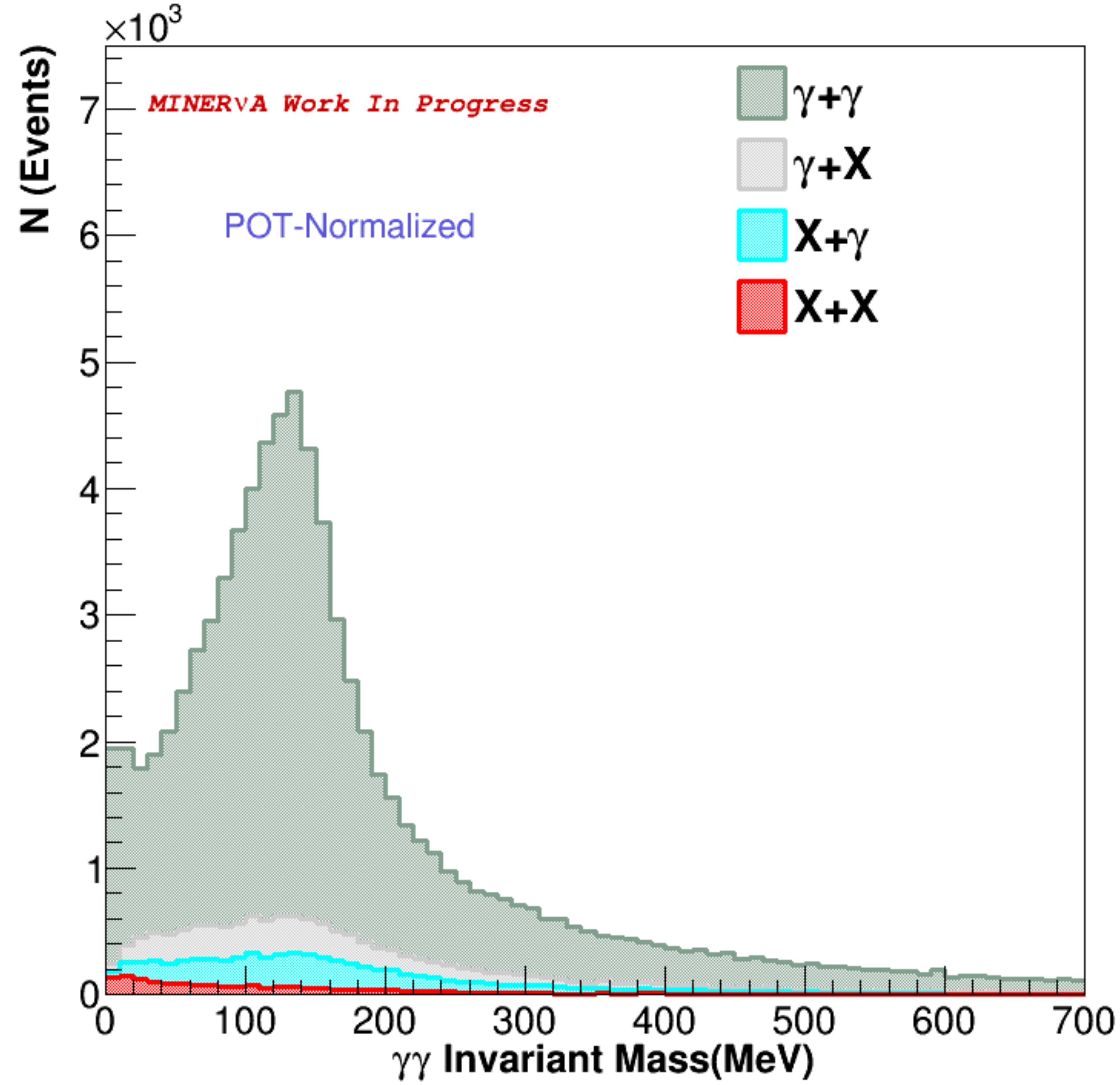


Class	Diagonal value from column normalized matrix
0	1
1 (EM-like)	0.85
2(Neutron)	0.47
3(Non-EM like)	0.97

Invariant mass distribution of neutral pion



Traditional reconstruction



With ML prediction (Only x view)

Summary

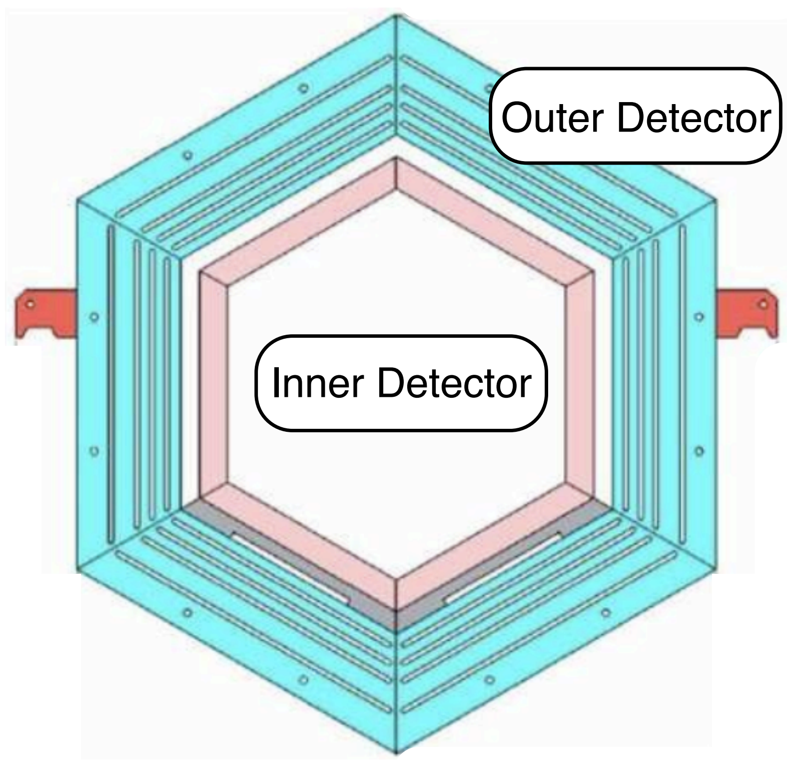
- ☑ We see significant improvement using ML based reconstruction over track-based method for vertex reconstruction at nuclear target region.
- ☑ We simulated with different FSI behavior and we saw the cross domain performance validation. However, by using DANN, to restrict the feature extraction only to features in both domains, we can train a domain-invariant classifier
- ☑ We compared MENNDL-based models with artisanal. It shows that human time can be saved by utilizing the computation time
- ☑ We have been applying ML-based classification for counting the number of hadrons and we tested the transfer learning technique. We are testing different loss functions (such as bilinear loss) and Resnet in order to improve the performance.
- ☑ Semantic segmentation approach has been successfully applied to reconstruct the neutral pion at the final state using prediction for one view. Next is to apply the prediction from all three views.
- ☑ We are planning to apply DANN in hadron multiplicity and neutral pion reconstruction problem

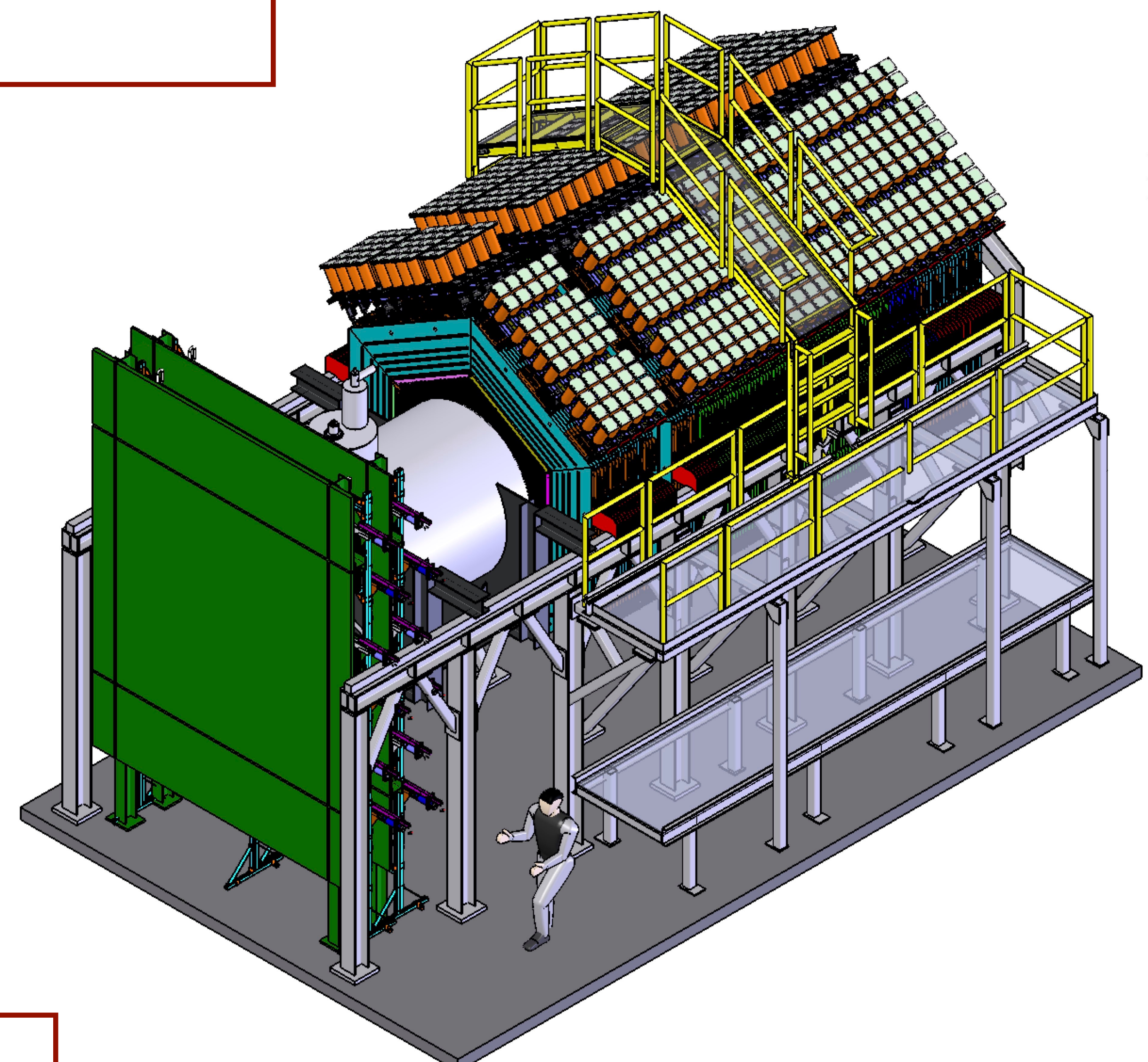
From MINERvA Collaboration:

Backup slides

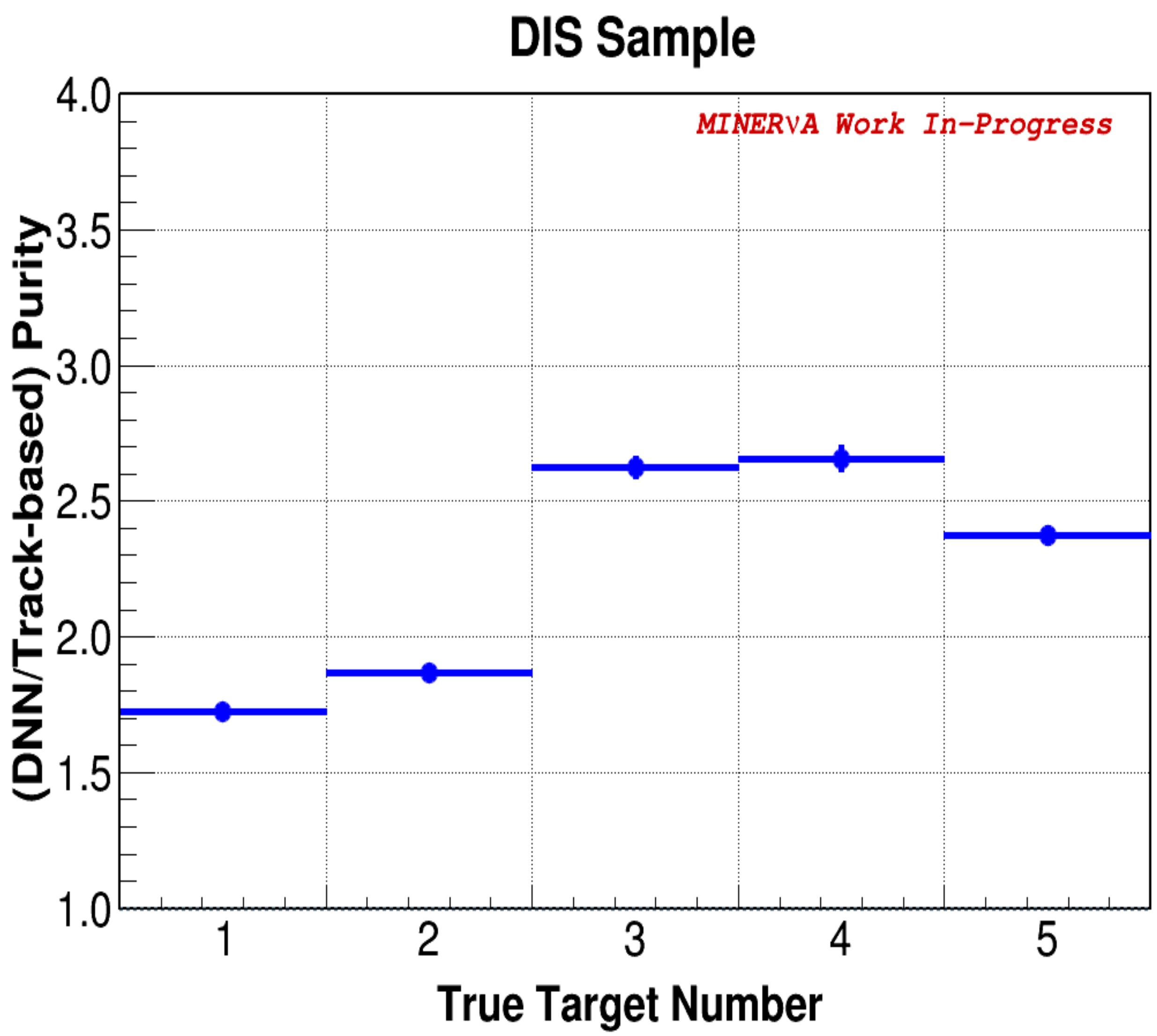
Introduction

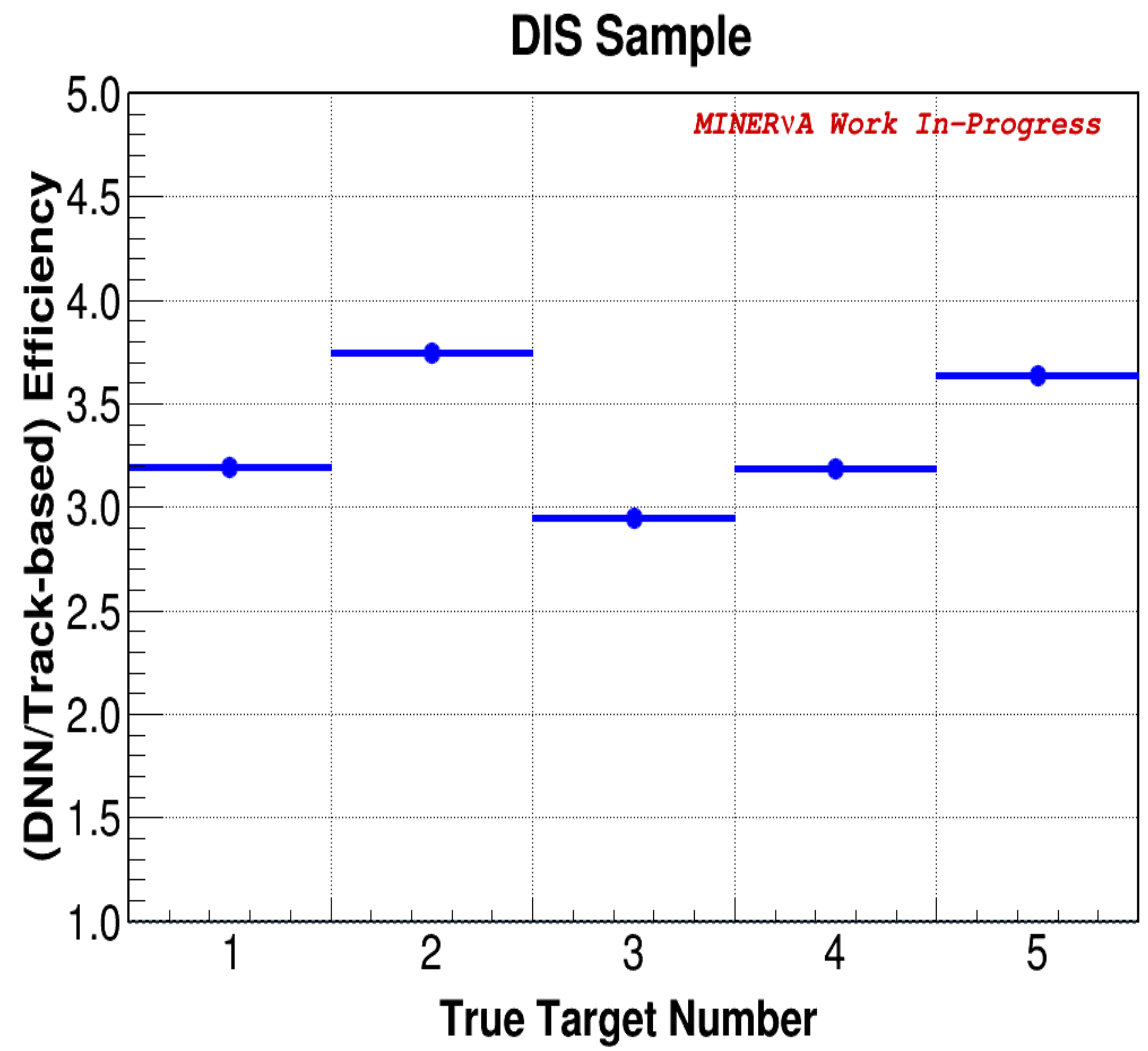
Outer detector (OD):
composed of a heavy steel frame, interspersed with scintillator bars, which serves both for calorimetry and as a support structure for the detector





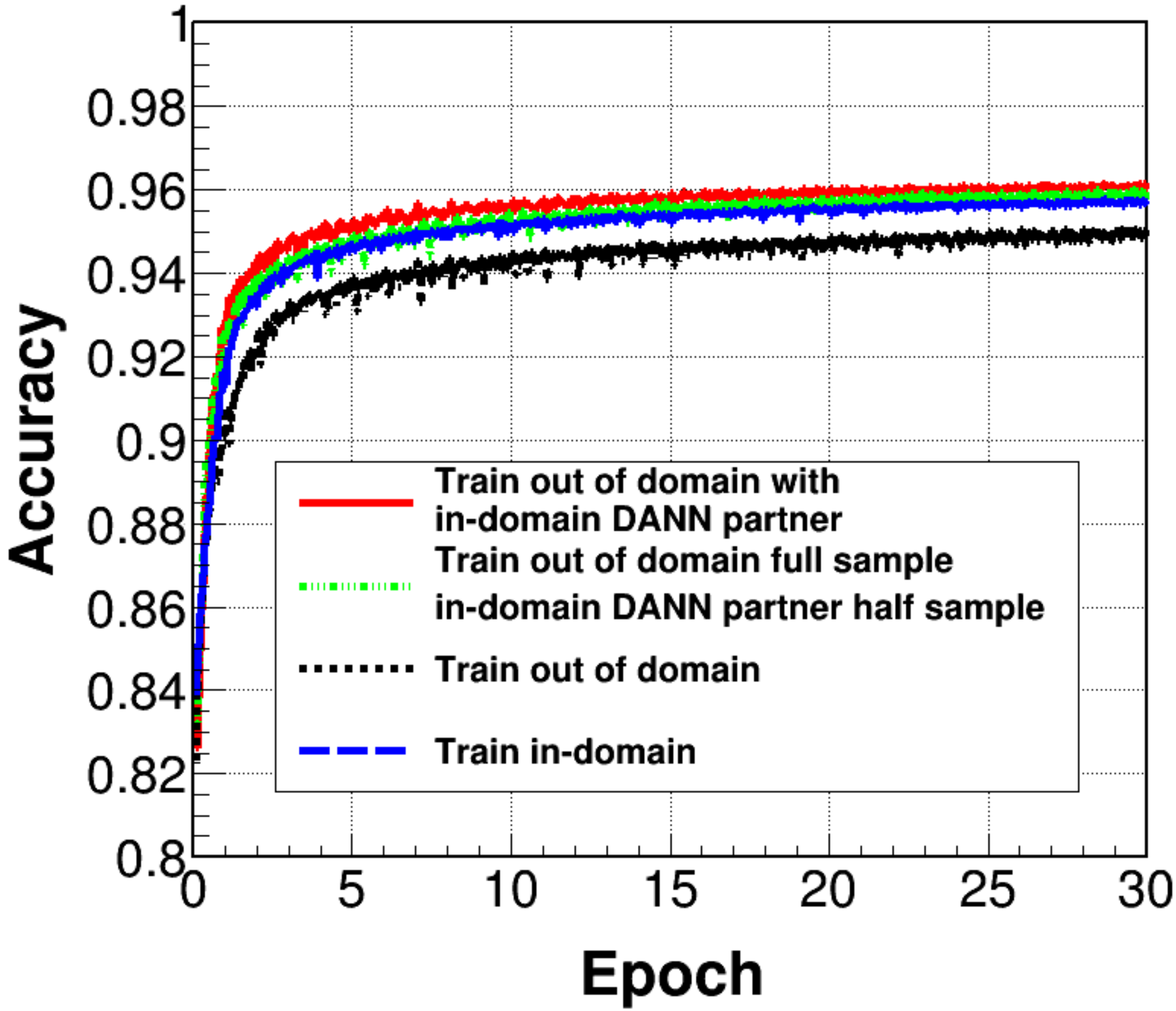
Inner Detector (ID):
Plastic scintillator strips create light from moving charged particles



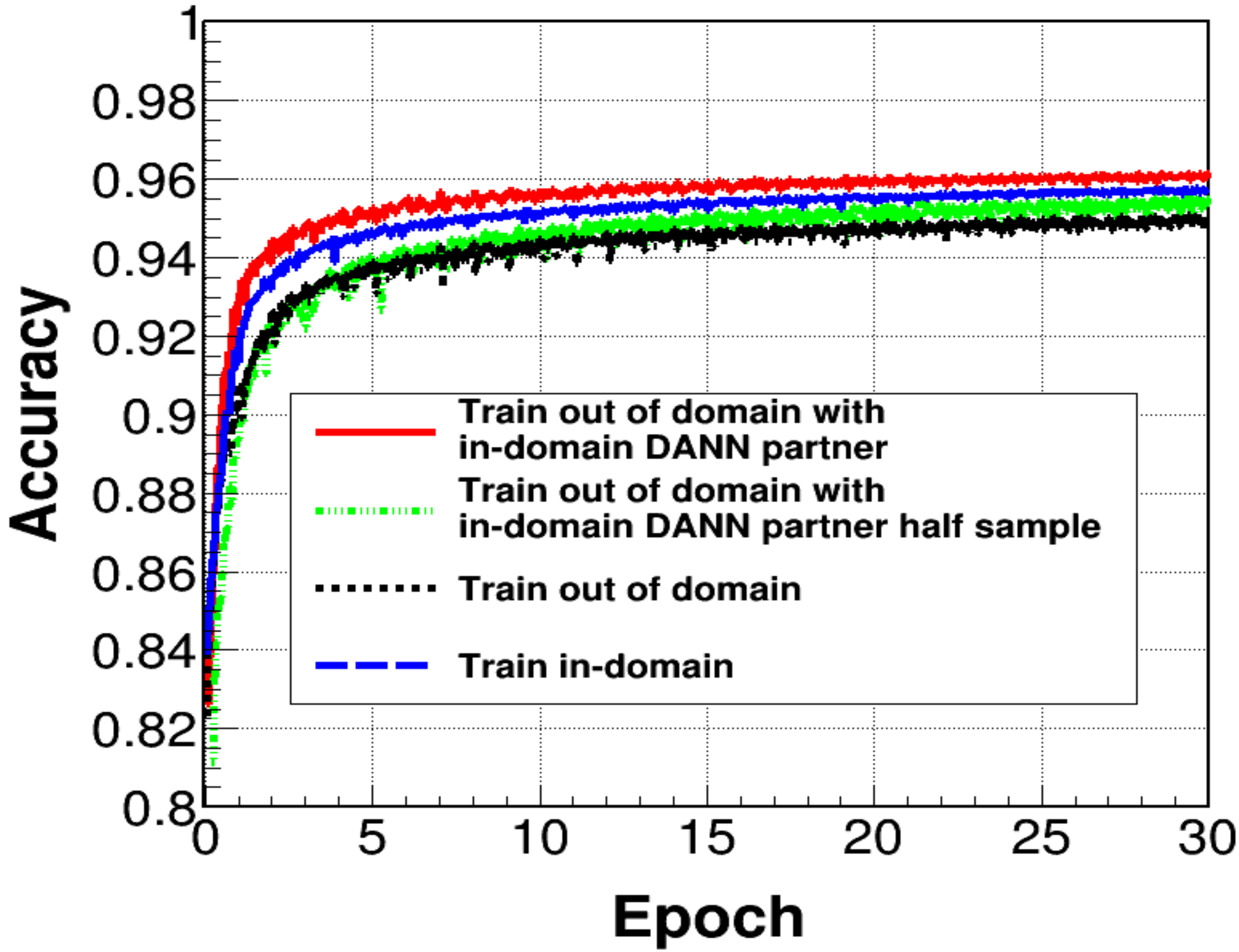


Signal purity has been improved by the ***factor of 2-3*** using ML technique compared to track based approach

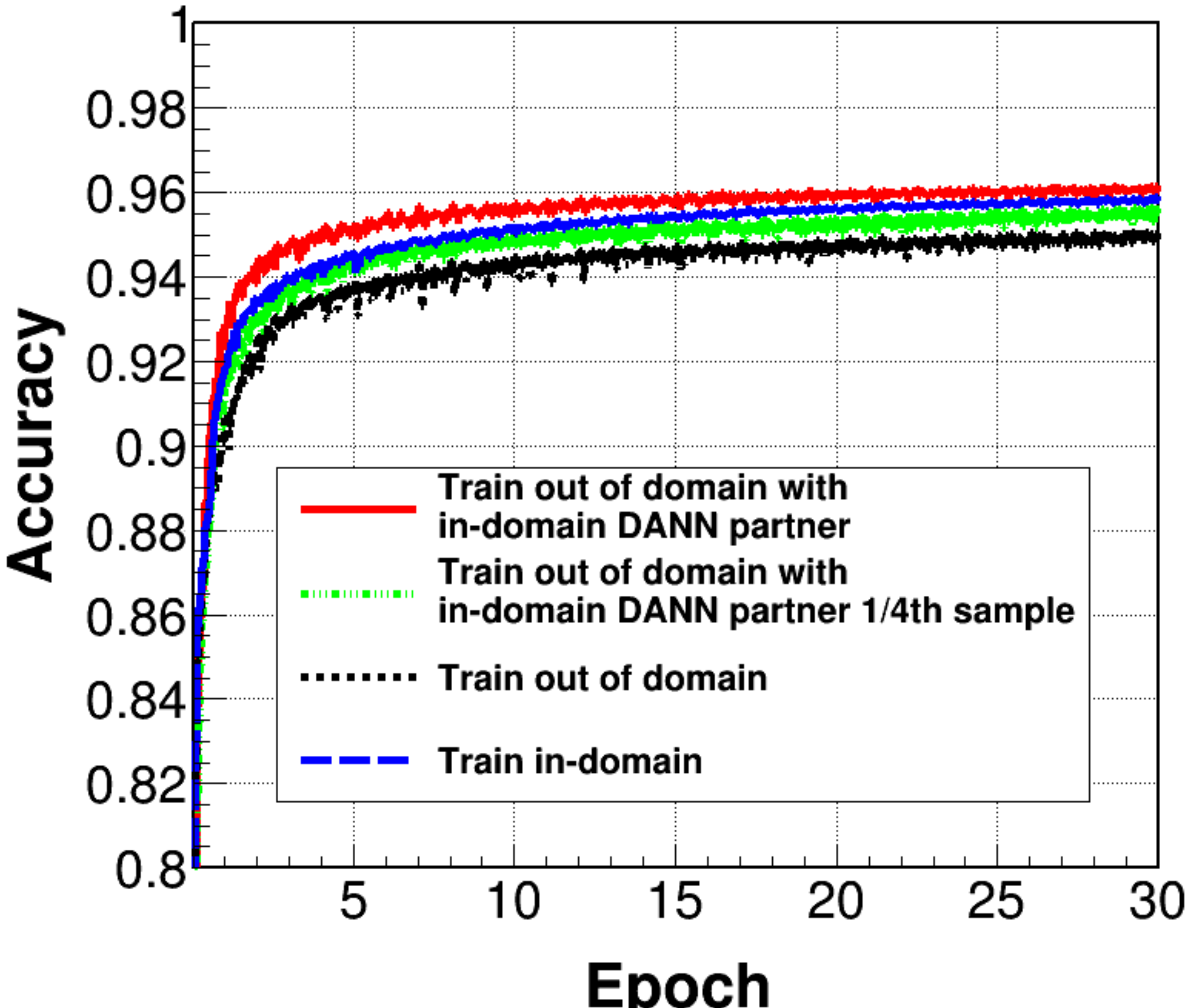
Effect of Sample size on the performance of DANN



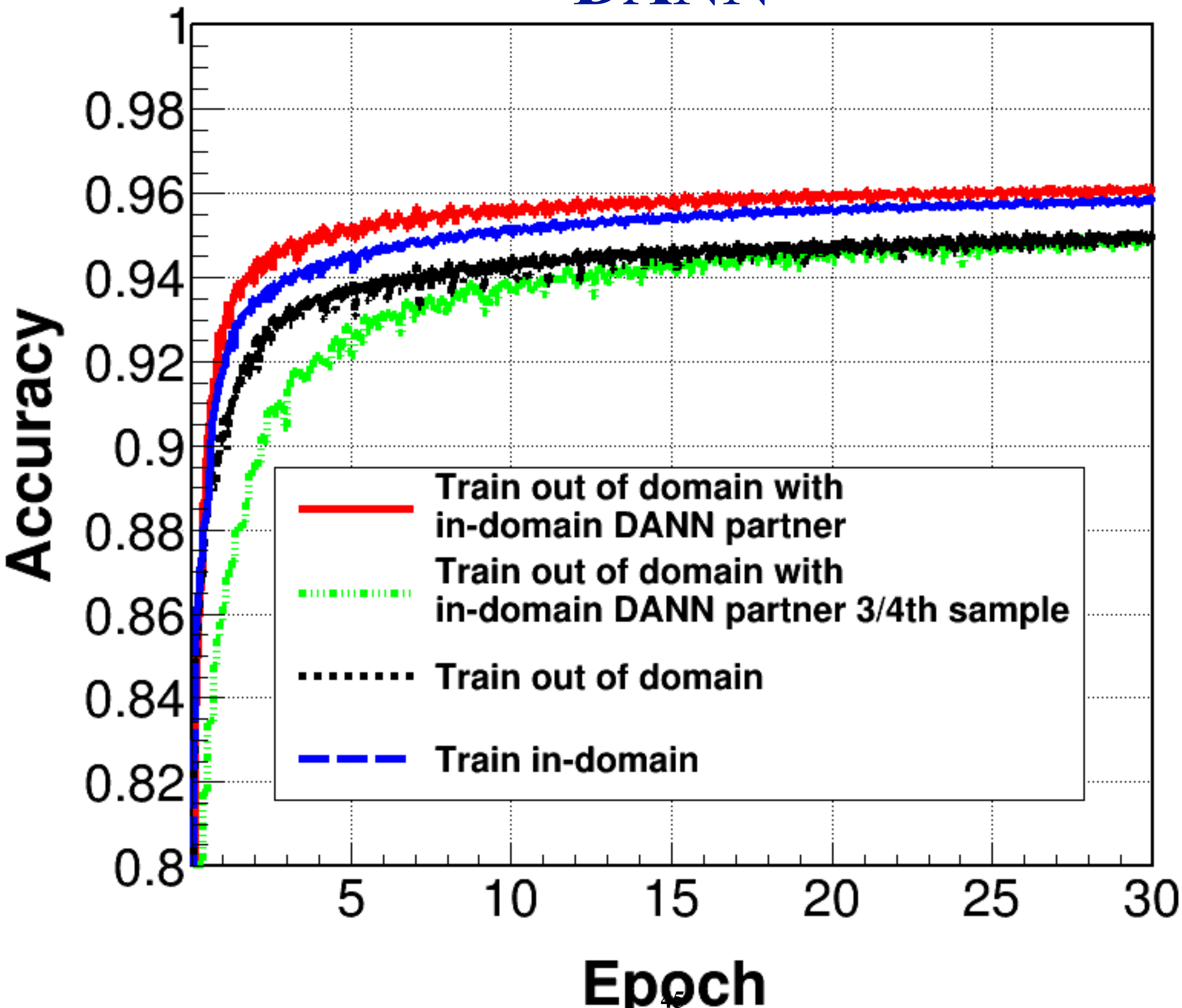
Effect of Sample size on the performance of DANN



Effect of Sample size on the performance of DANN



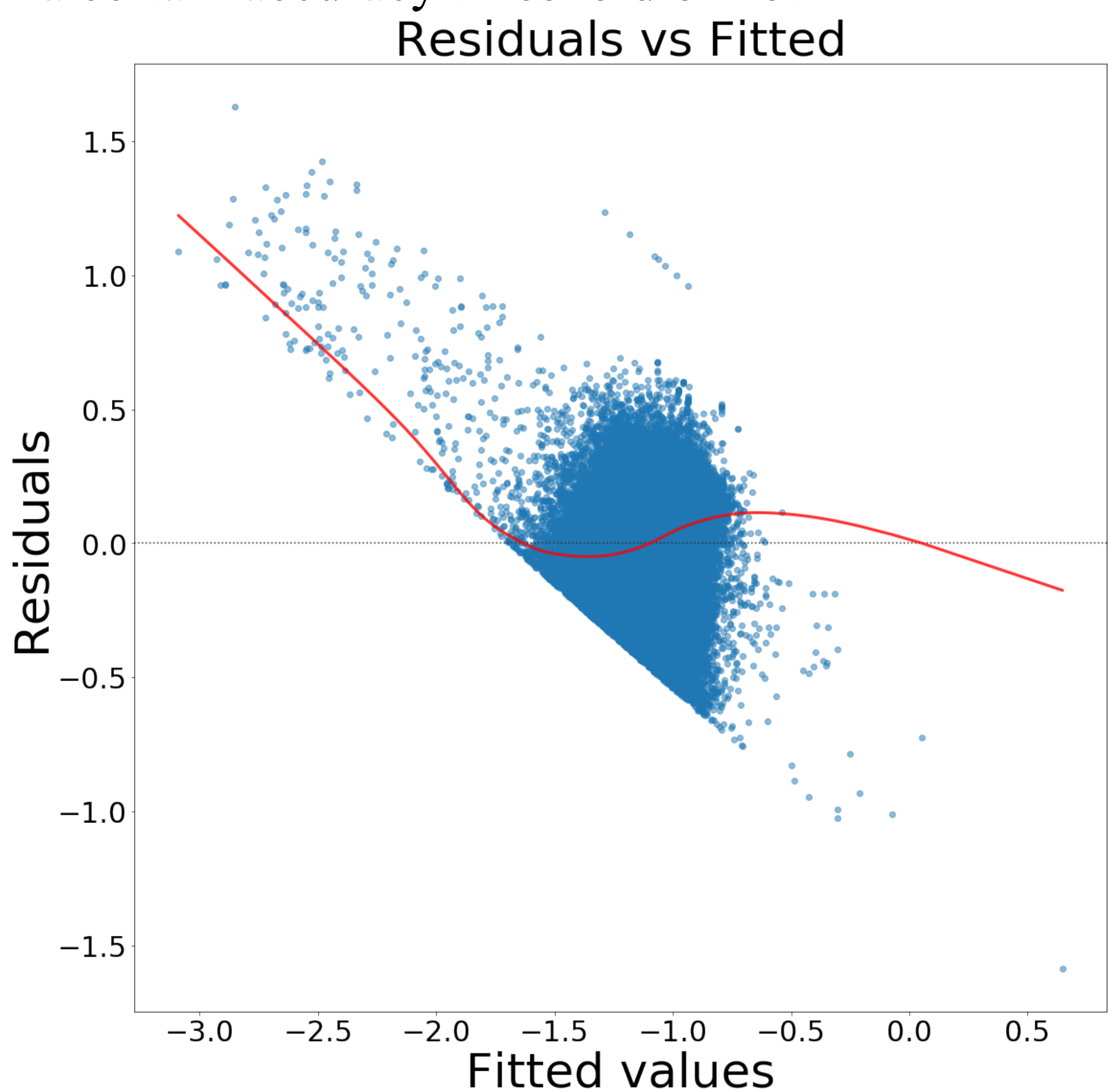
Effect of Sample size on the performance of DANN

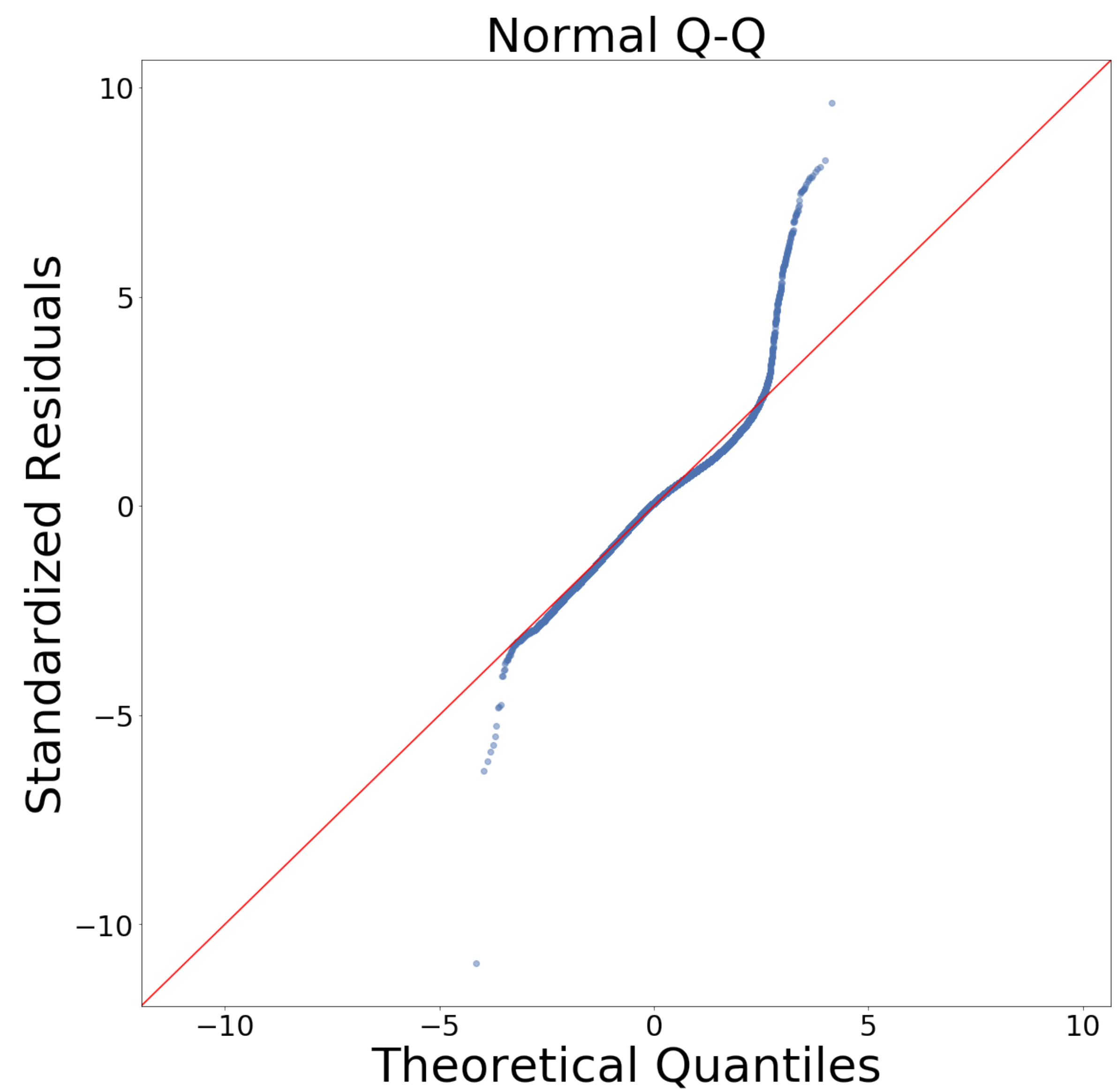


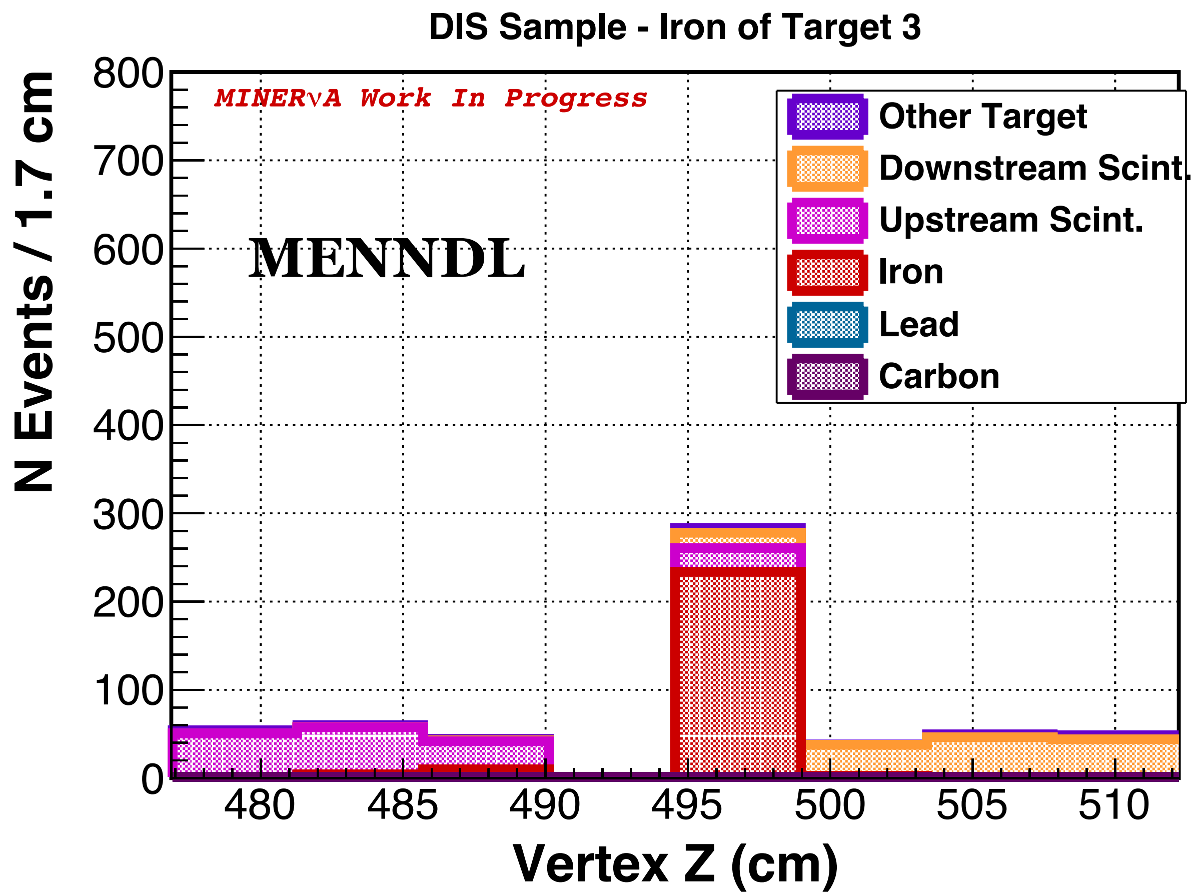
Infer the accuracy of CNN from its structure

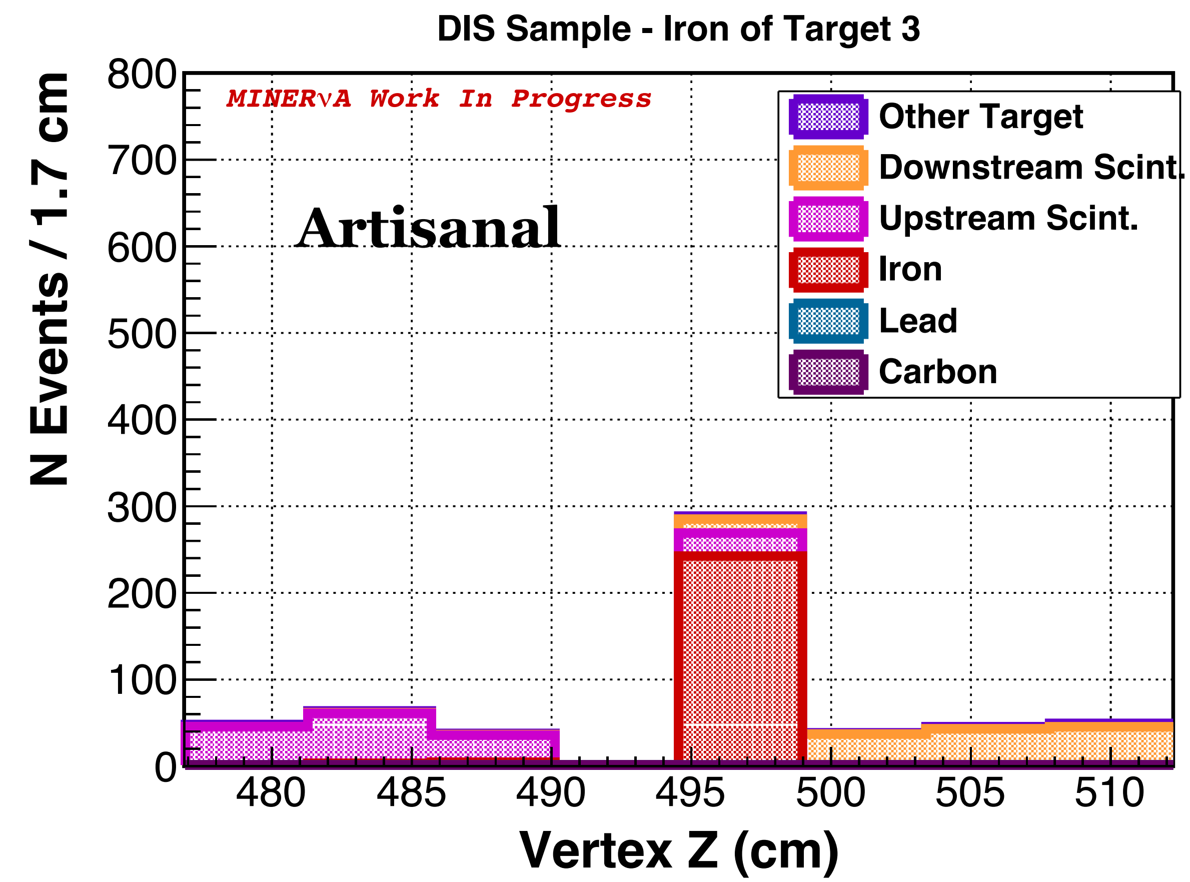
proposed a systematic methods to uniformly extract architectural parameters from different architecture of CNNs

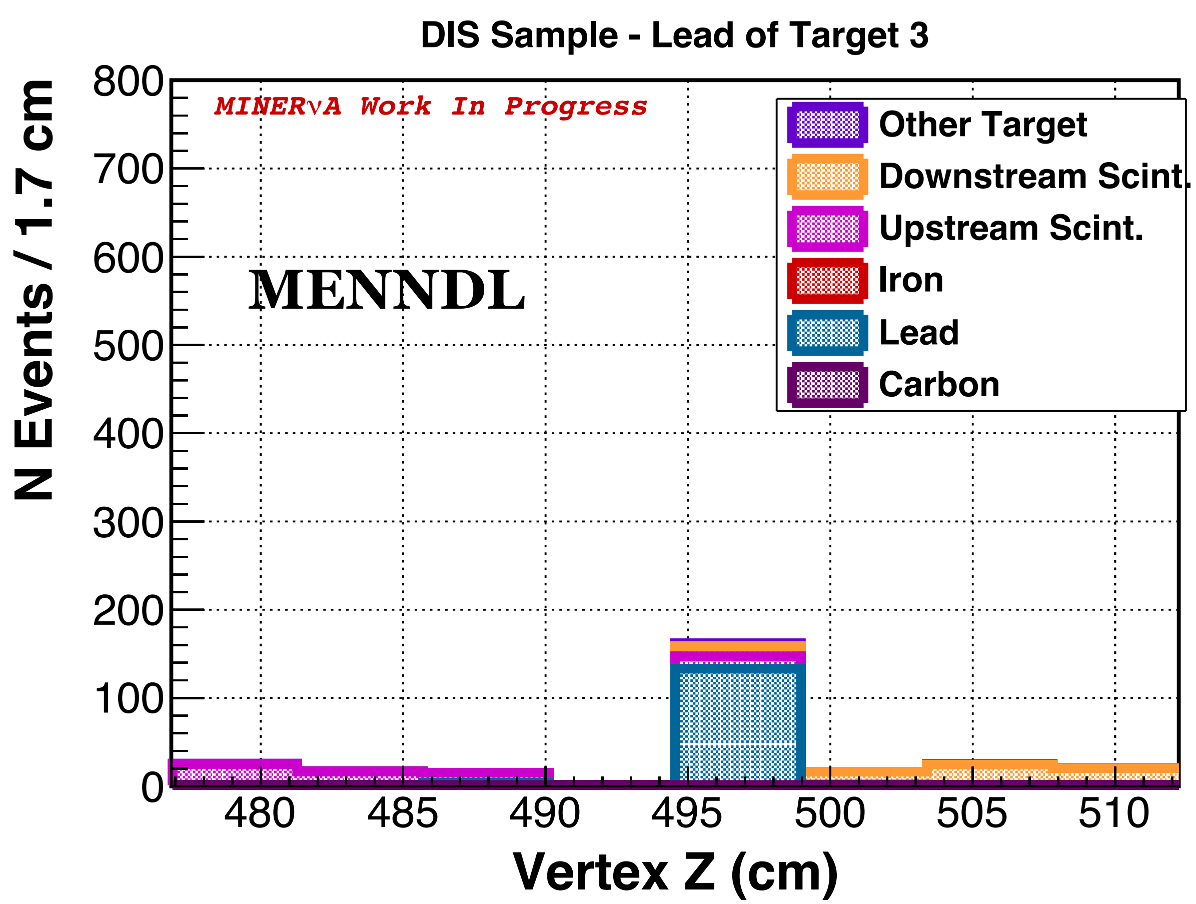
demonstrated the predictive nature of those architectural attributes of CNNs in two specific problems—MINERvA Vertex Finding and MINERvA Hadron Multiplicity Counting—through building classification models, which can predict whether a CNN's architecture is likely to perform better than a certain accuracy threshold or not

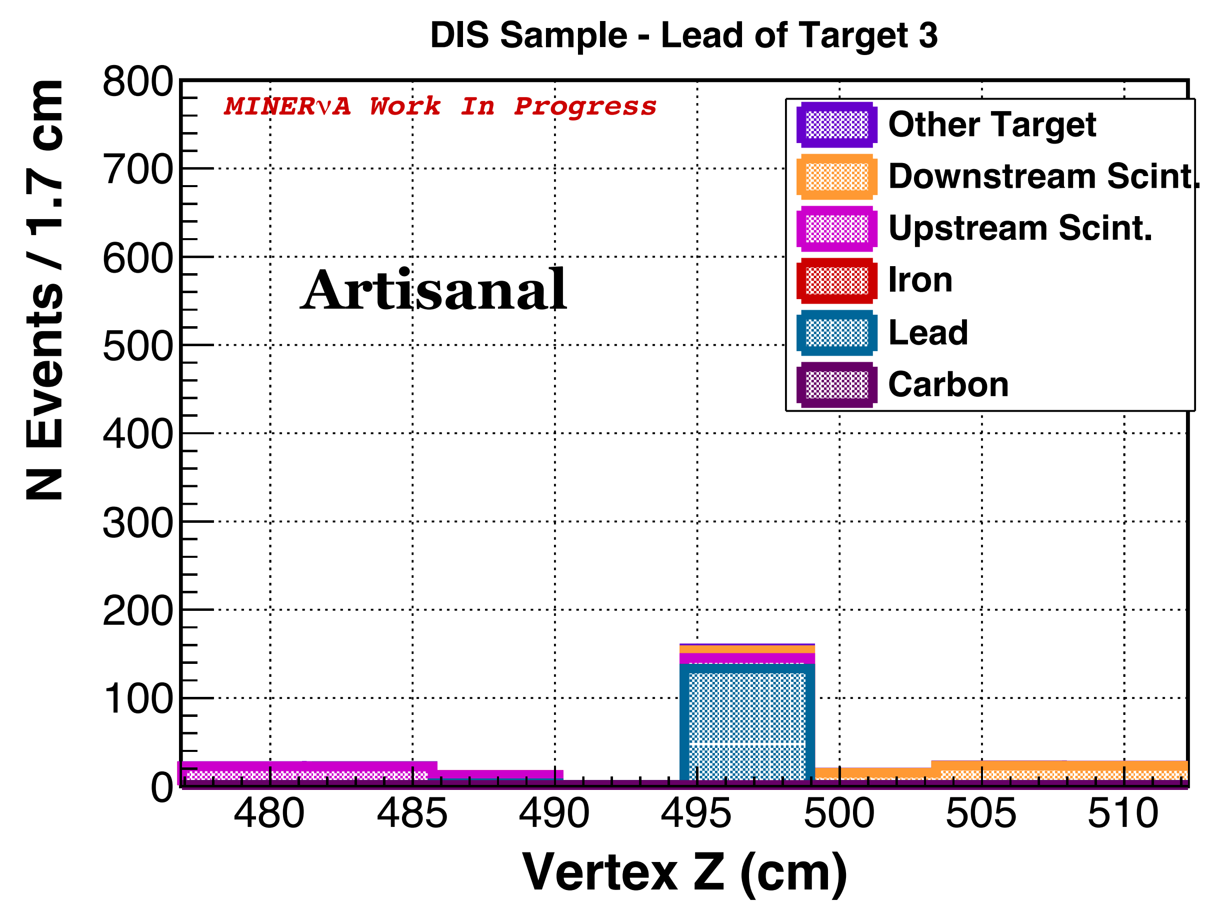


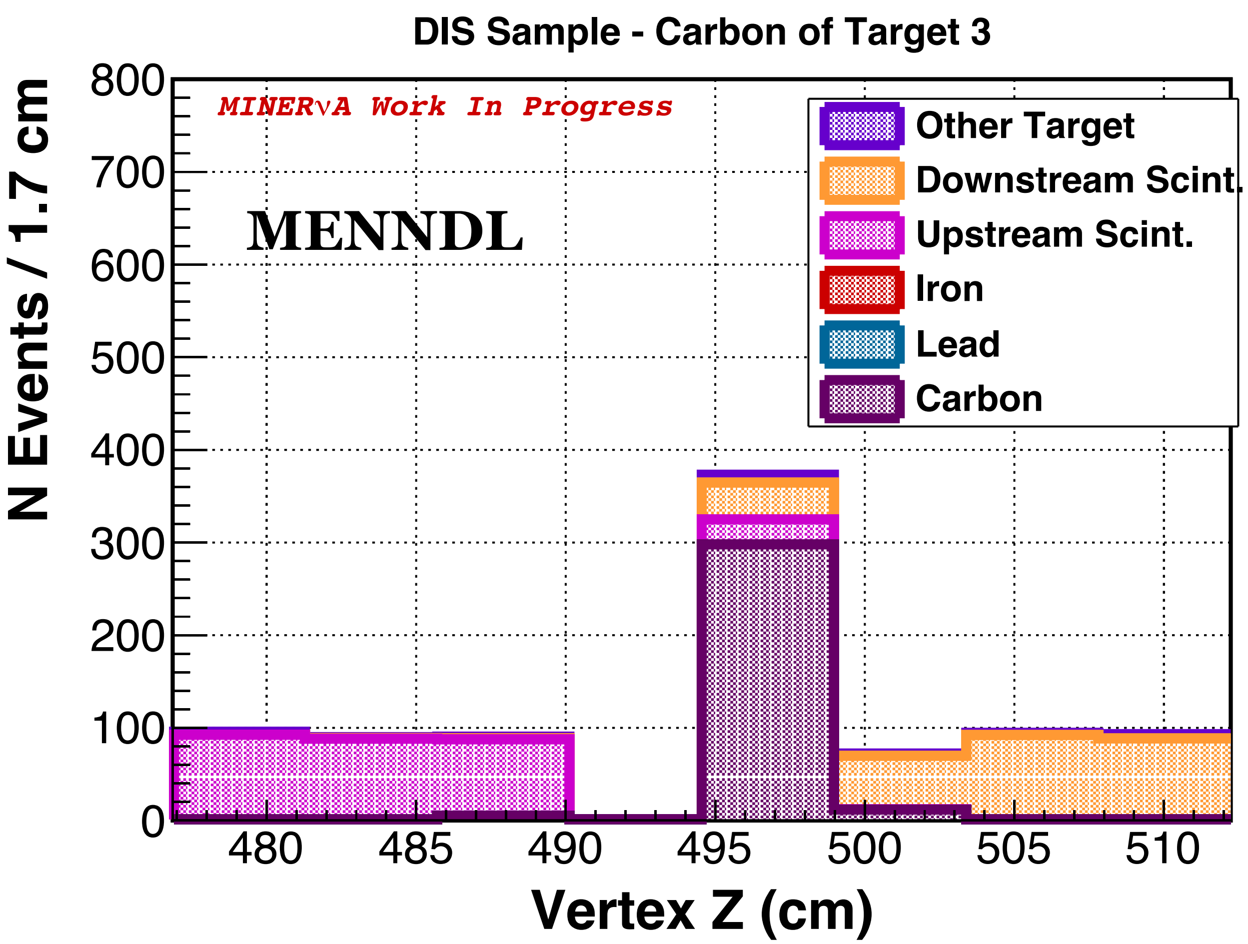


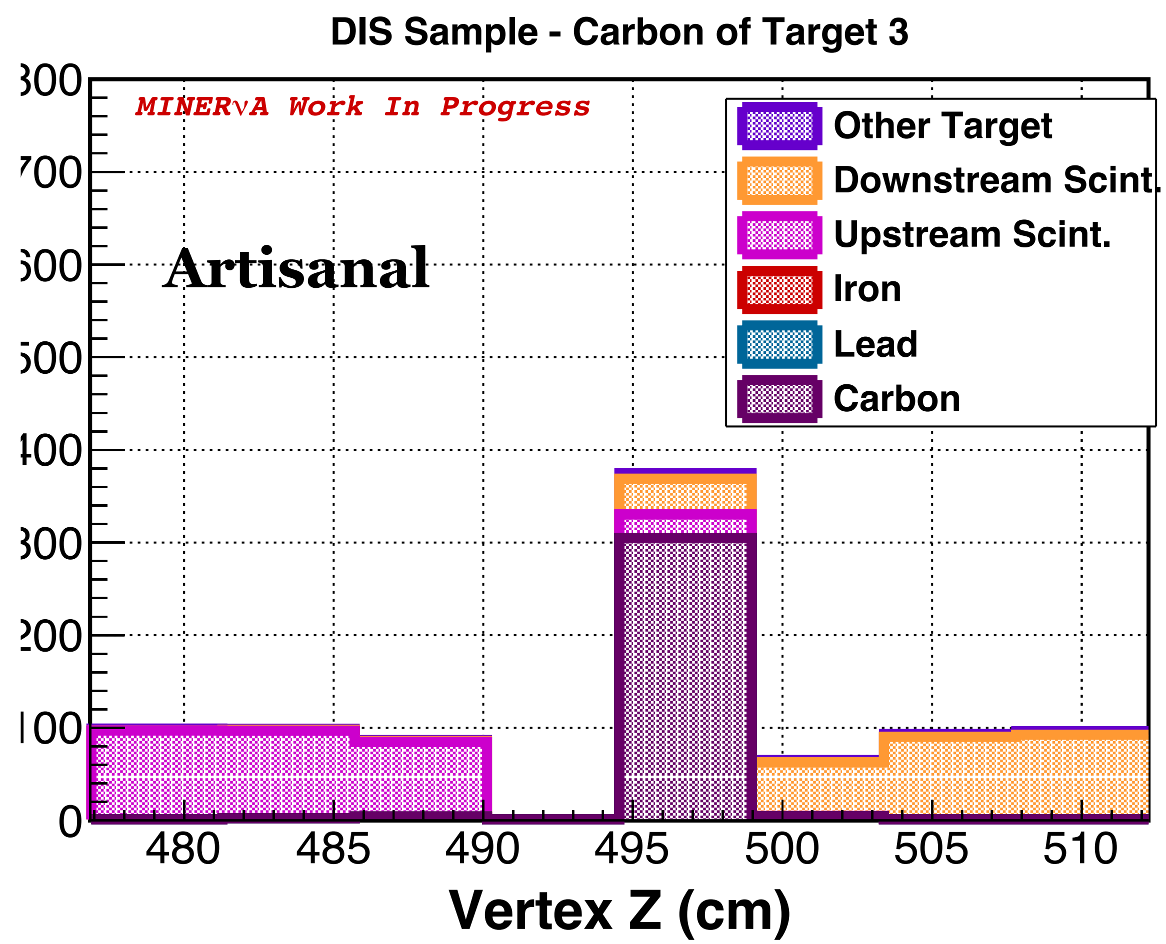

DIS Sample - Iron of Target 3
MINERνA Work In Progress
MENNDL
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
N Events / 1.7 cm
0
100
200
300
400
500
600
700
800
480
485
490
495
500
505
510
Vertex Z (cm)

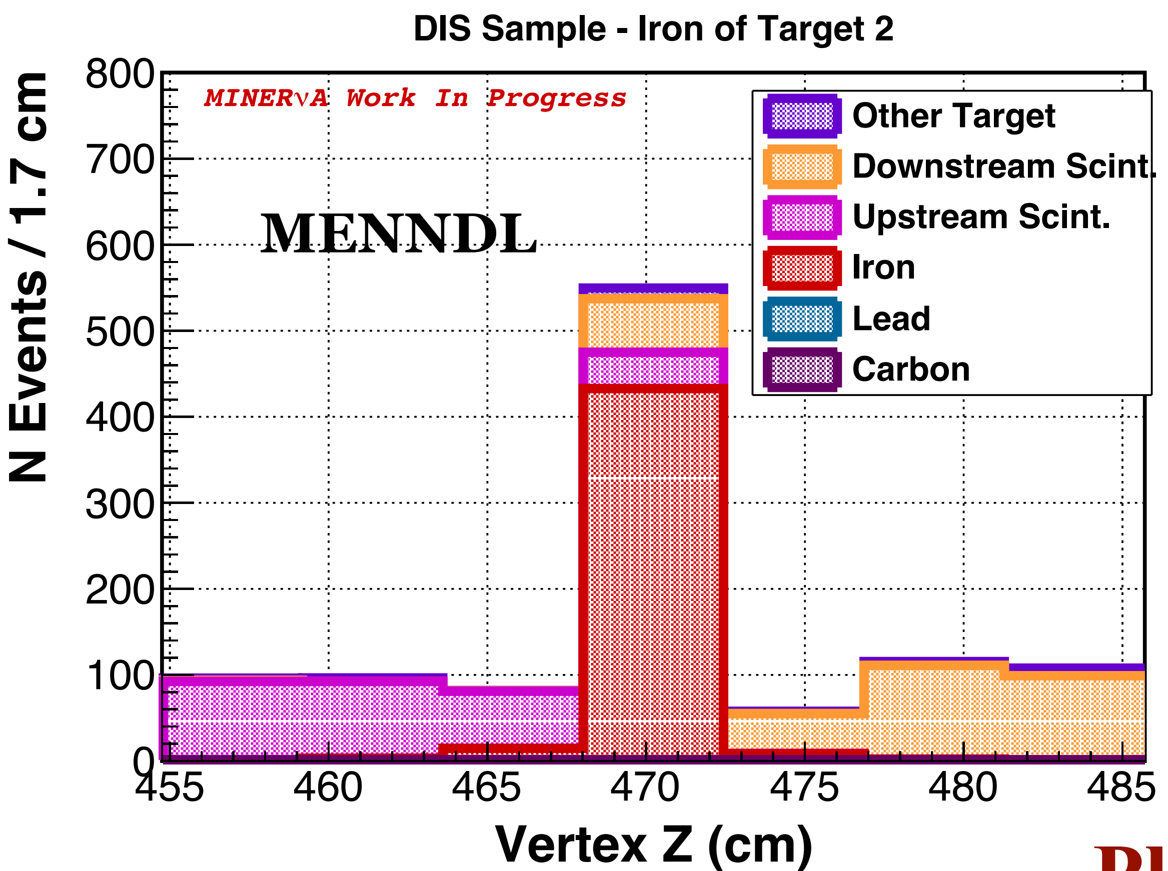

DIS Sample - Iron of Target 3
MINERνA Work In Progress
Artisanal
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
N Events / 1.7 cm
0
100
200
300
400
500
600
700
800
480
485
490
495
500
505
510
Vertex Z (cm)

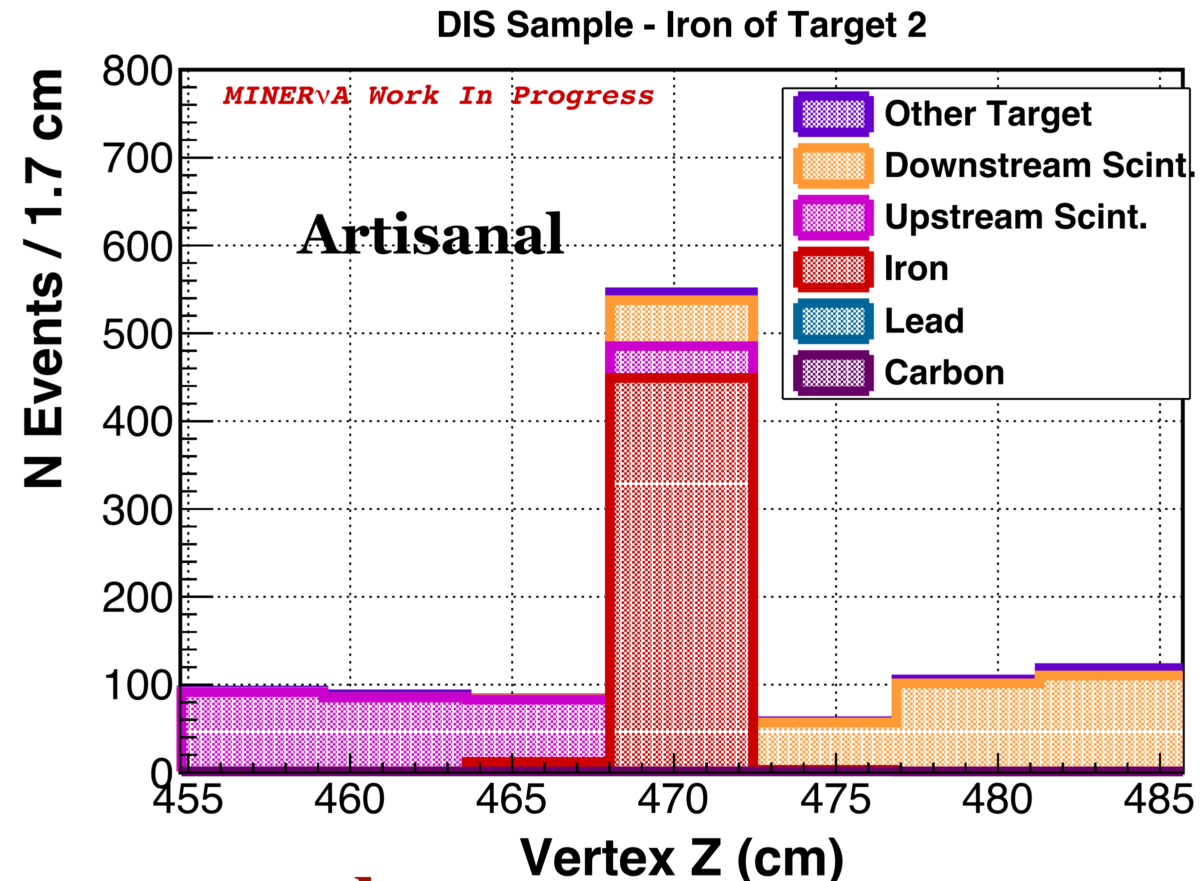

DIS Sample - Lead of Target 3
MINERνA Work In Progress
MENNDL
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
N Events / 1.7 cm
0
100
200
300
400
500
600
700
800
480
485
490
495
500
505
510
Vertex Z (cm)


DIS Sample - Lead of Target 3
MINERνA Work In Progress
Artisanal
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
N Events / 1.7 cm
0
100
200
300
400
500
600
700
800
480
485
490
495
500
505
510
Vertex Z (cm)

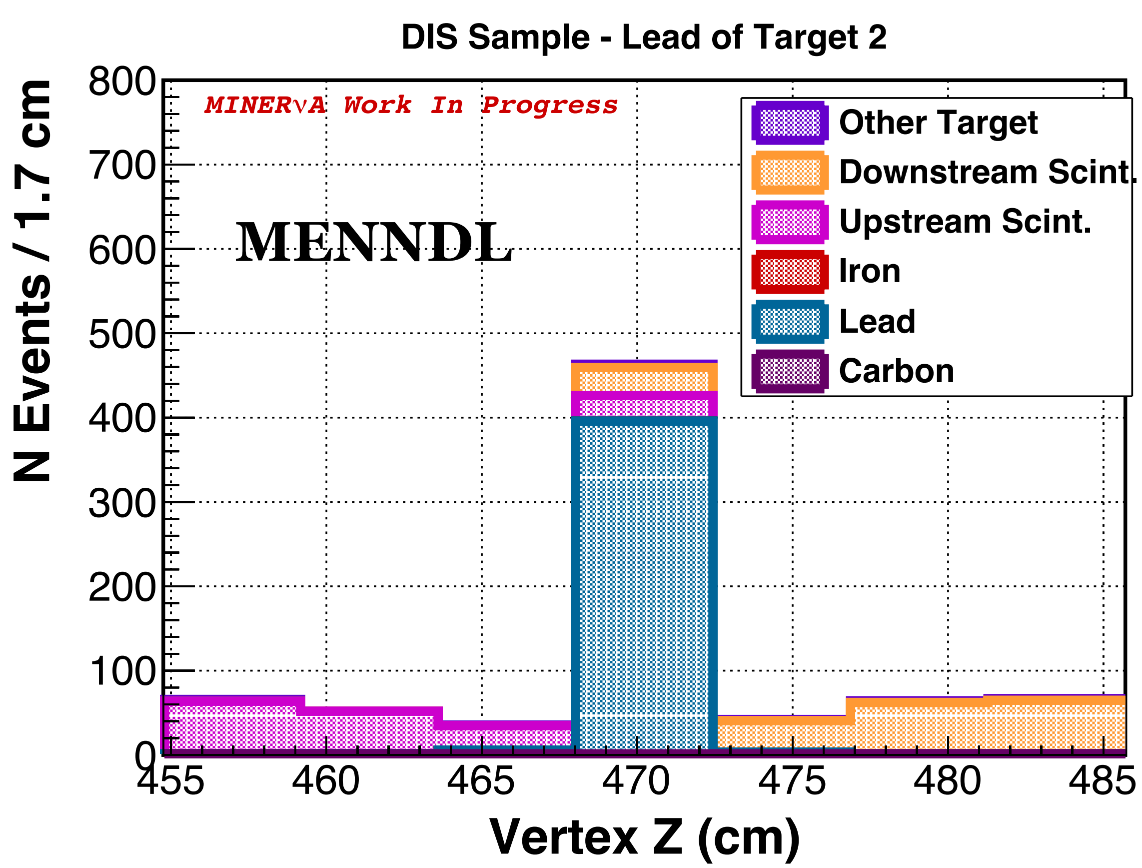

DIS Sample - Carbon of Target 3
MINERνA Work In Progress
MENNDL
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
N Events / 1.7 cm
800
700
600
500
400
300
200
100
0
480
485
490
495
500
505
510
Vertex Z (cm)

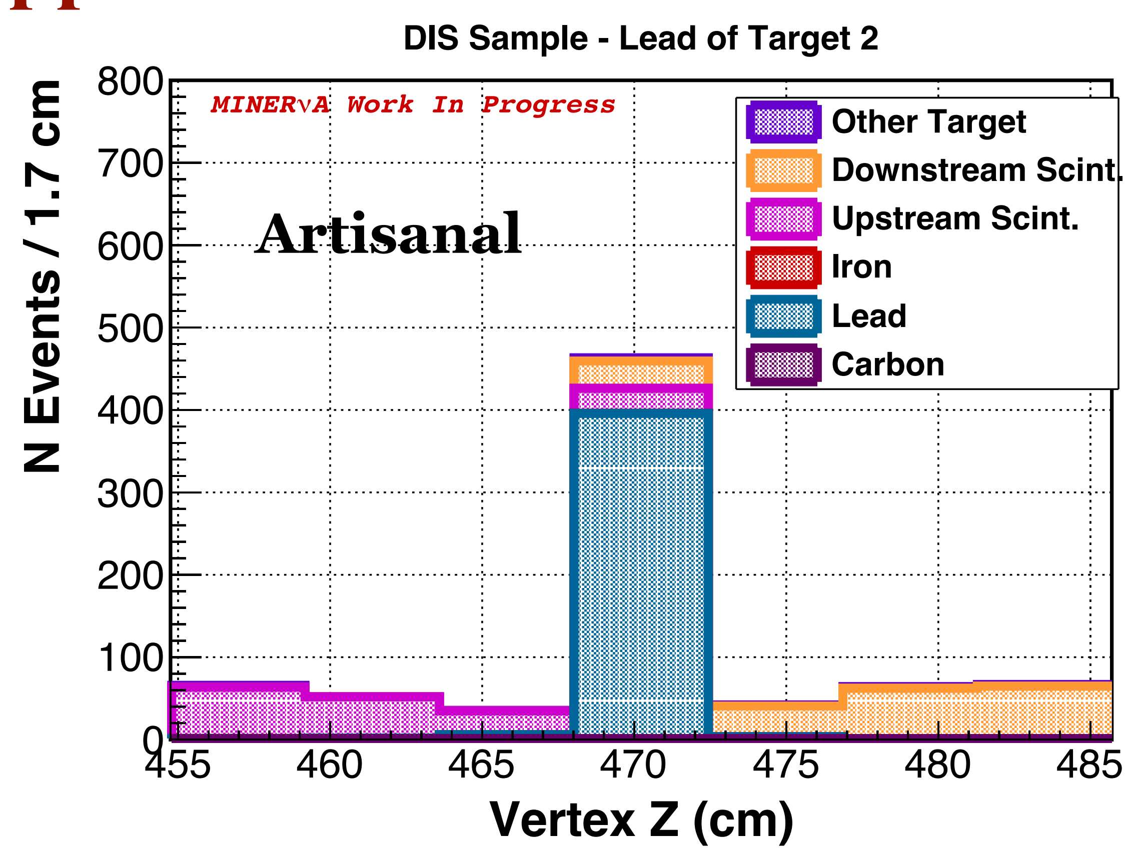

DIS Sample - Carbon of Target 3
MINERνA Work In Progress
Artisanal
Other Target
Downstream Scint.
Upstream Scint.
Iron
Lead
Carbon
0
480
485
490
495
500
505
510
Vertex Z (cm)



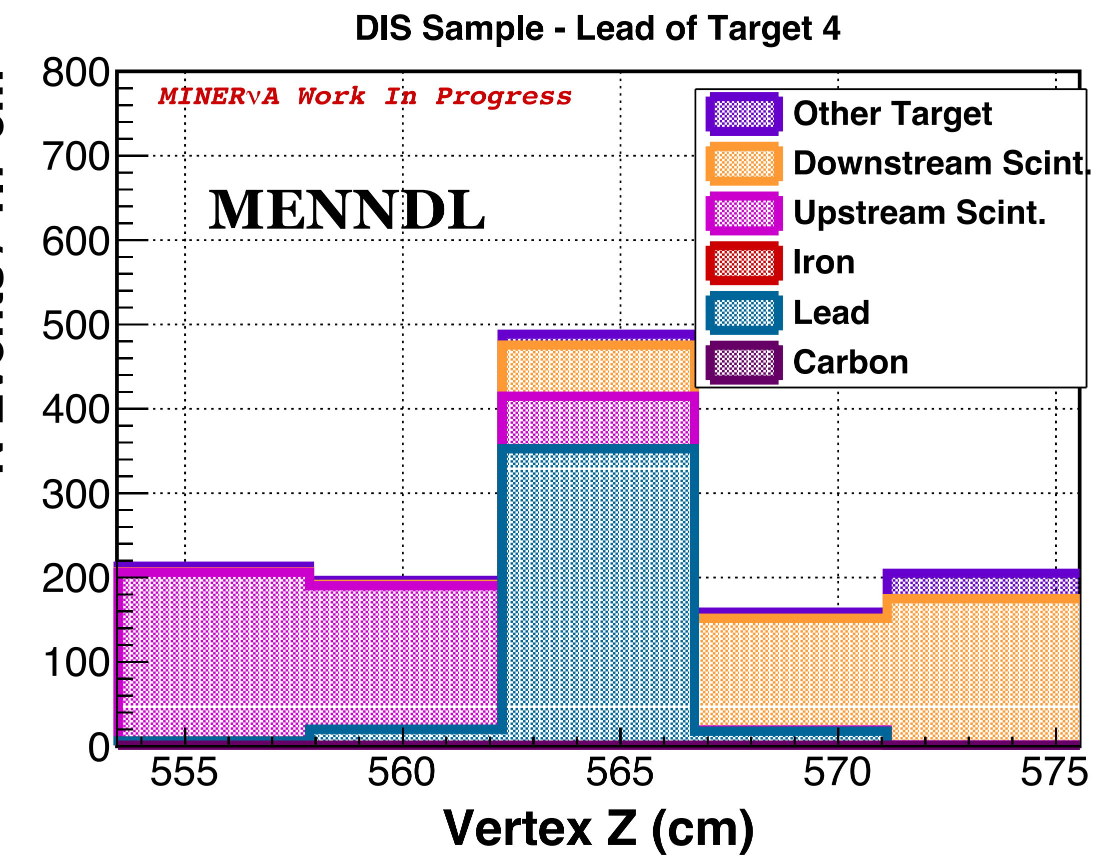


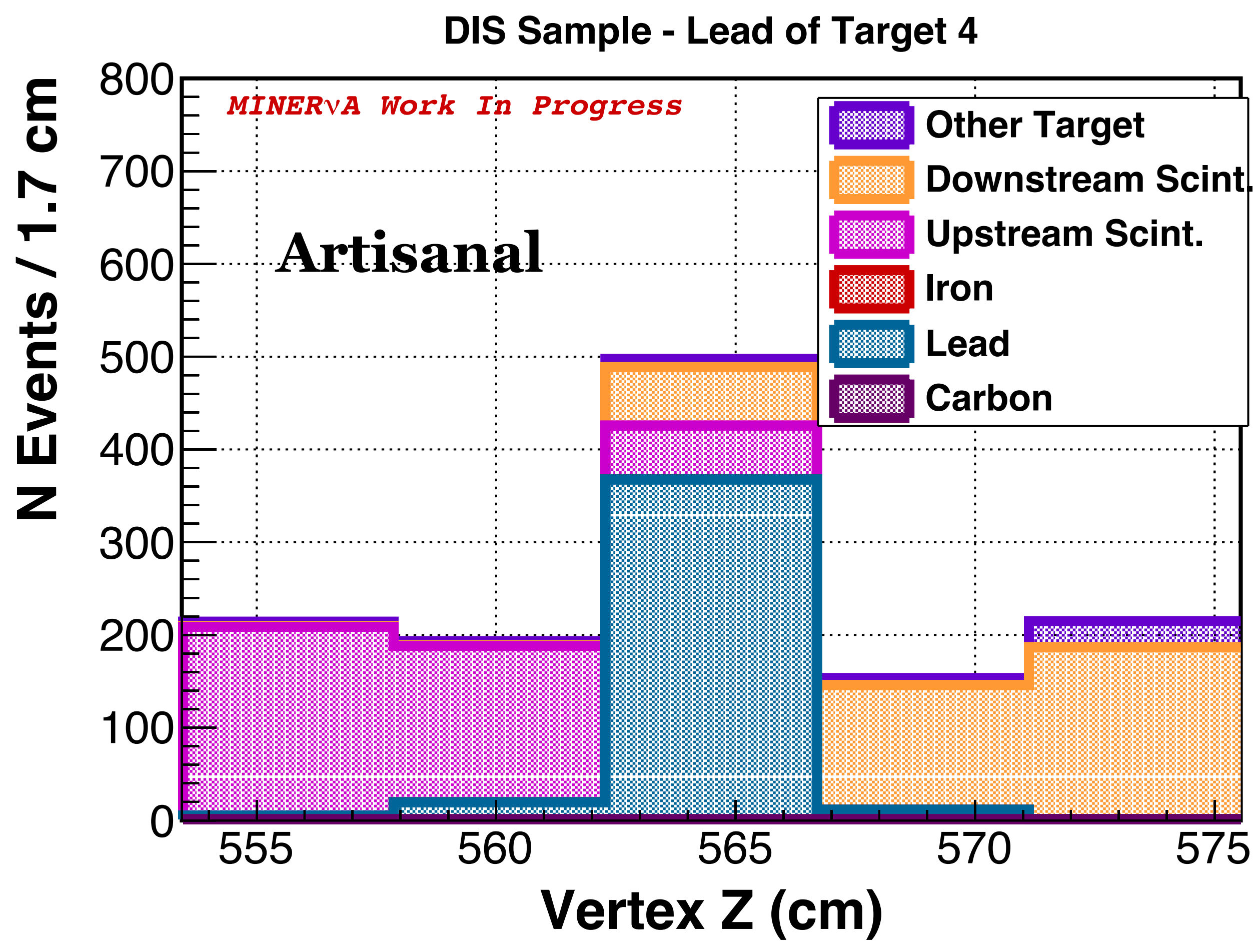
Plot for approval

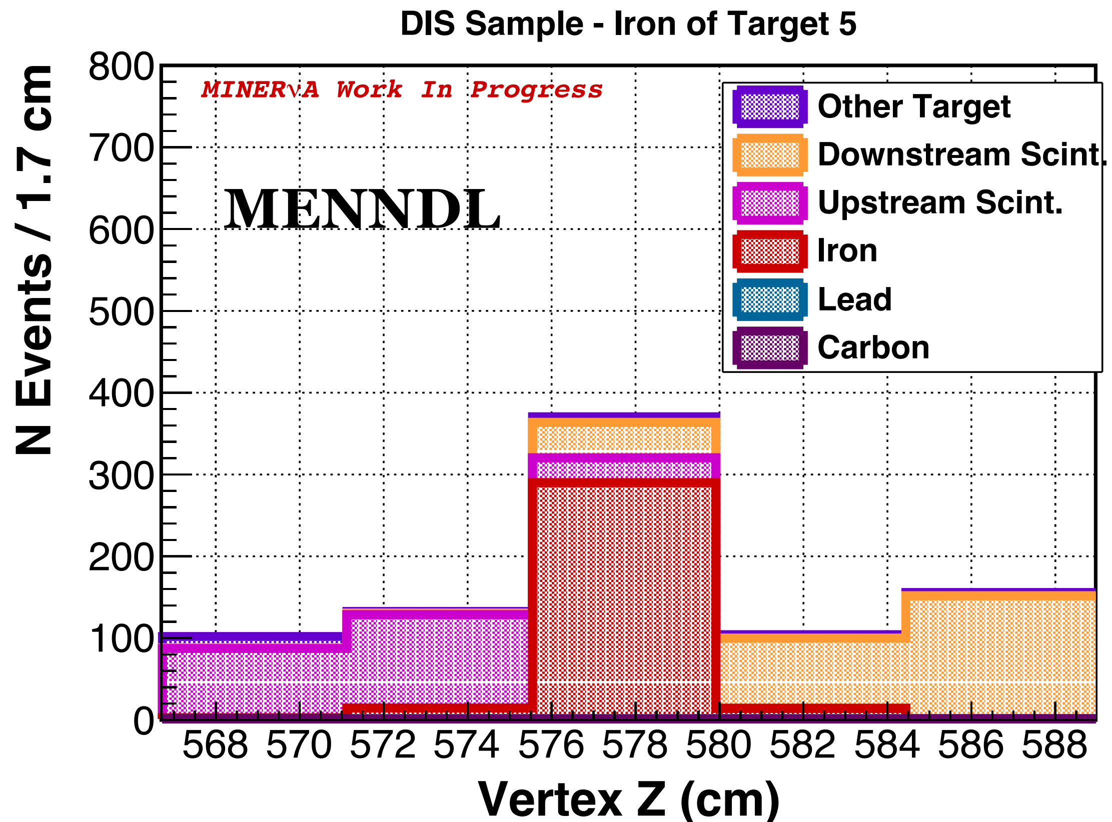


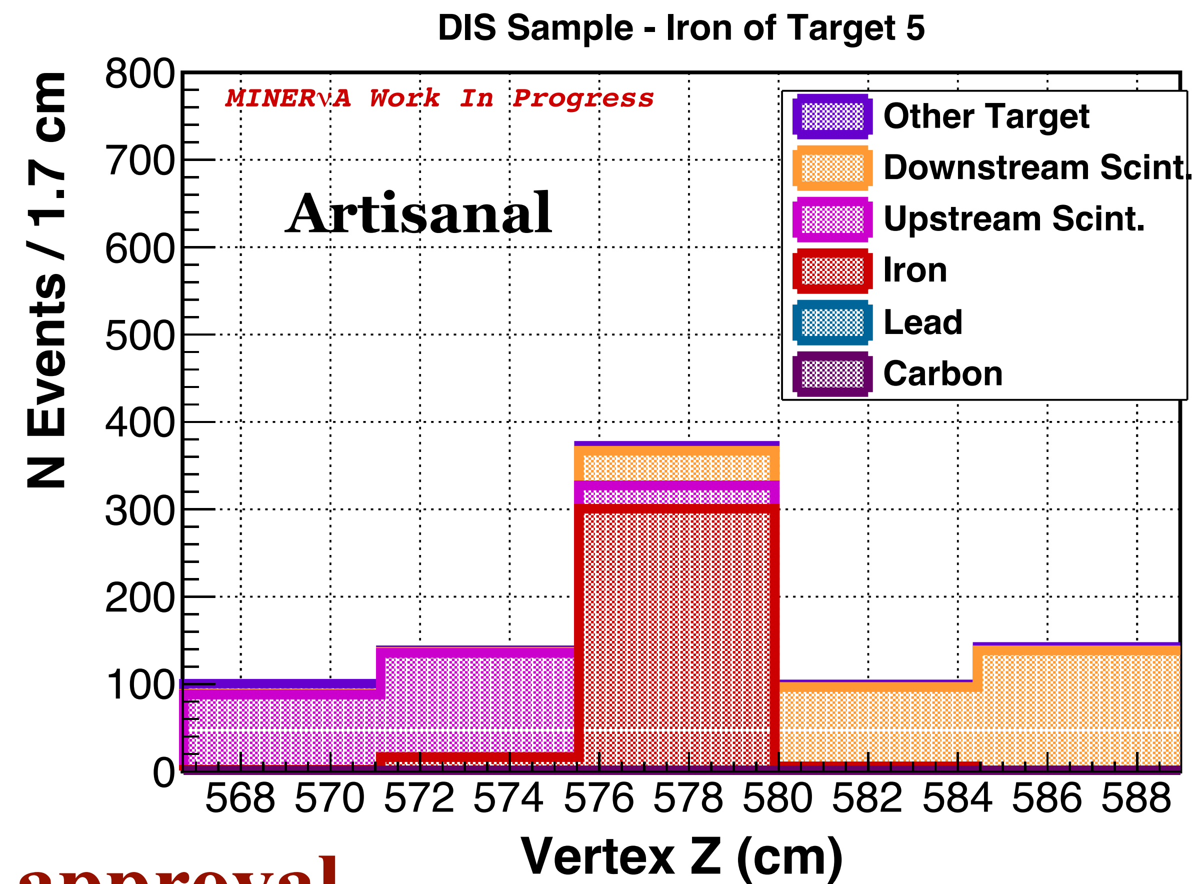


Plot for approval

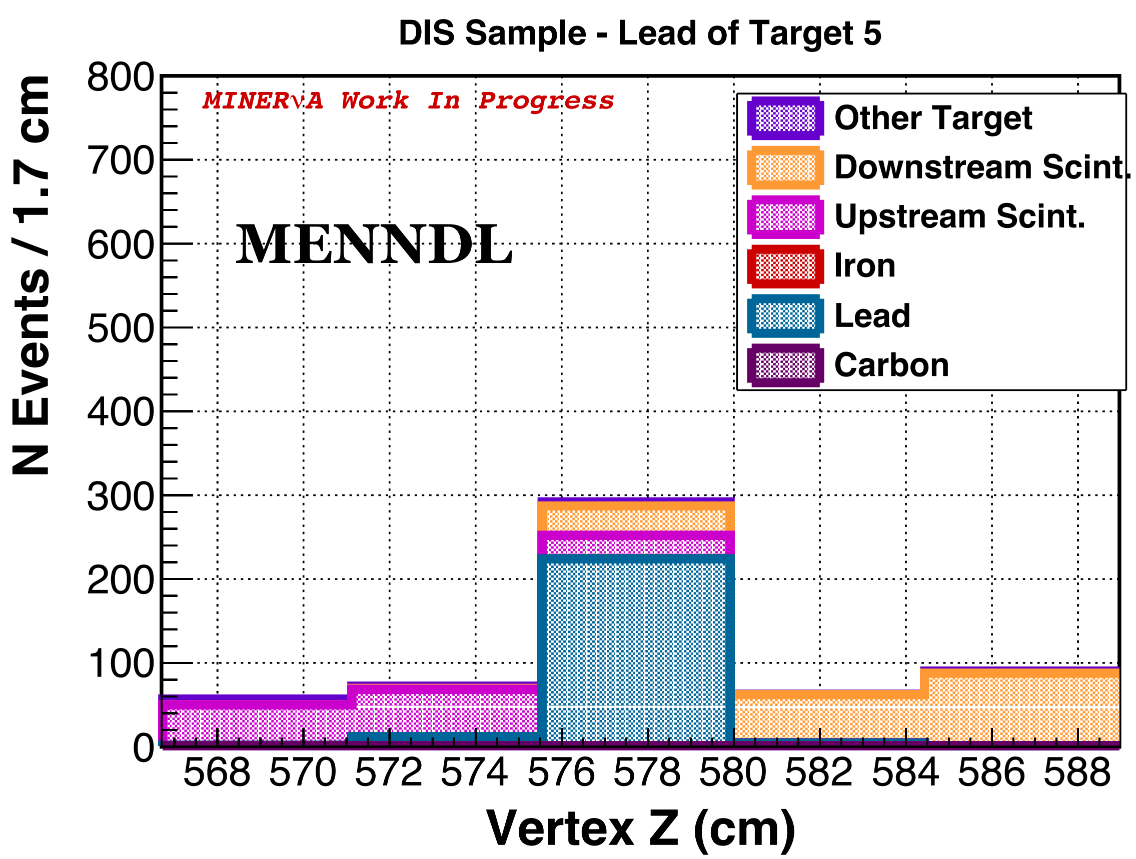


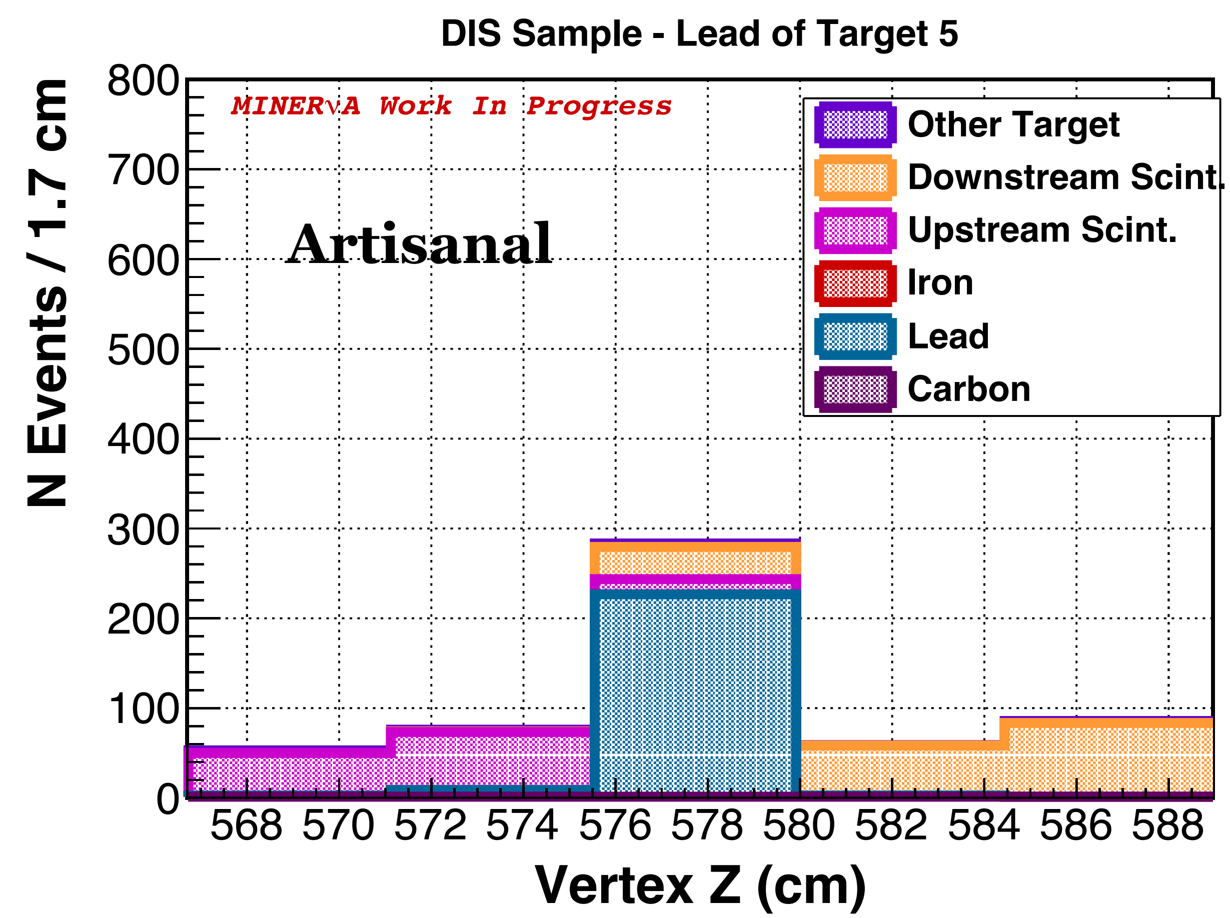






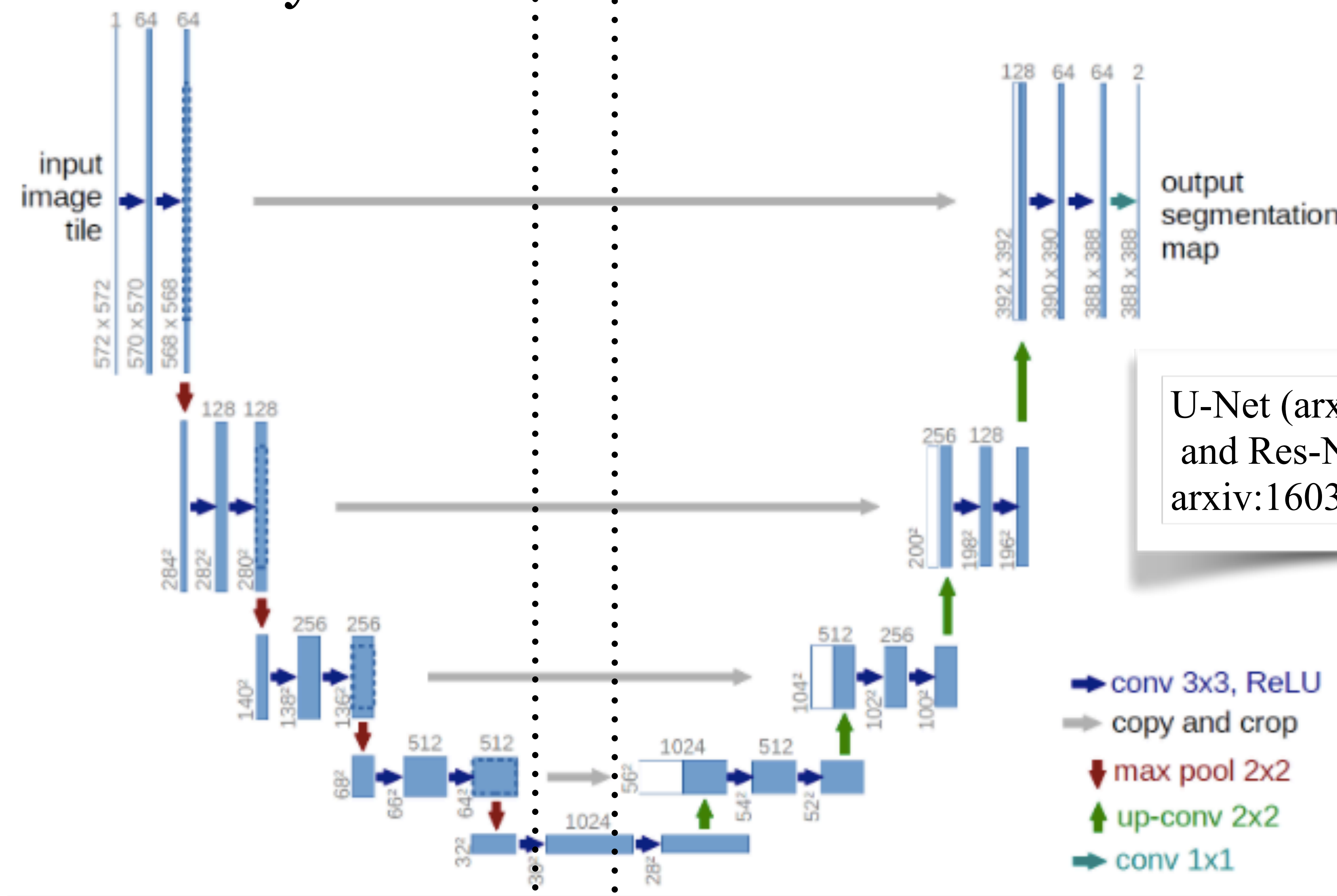
Plot for approval



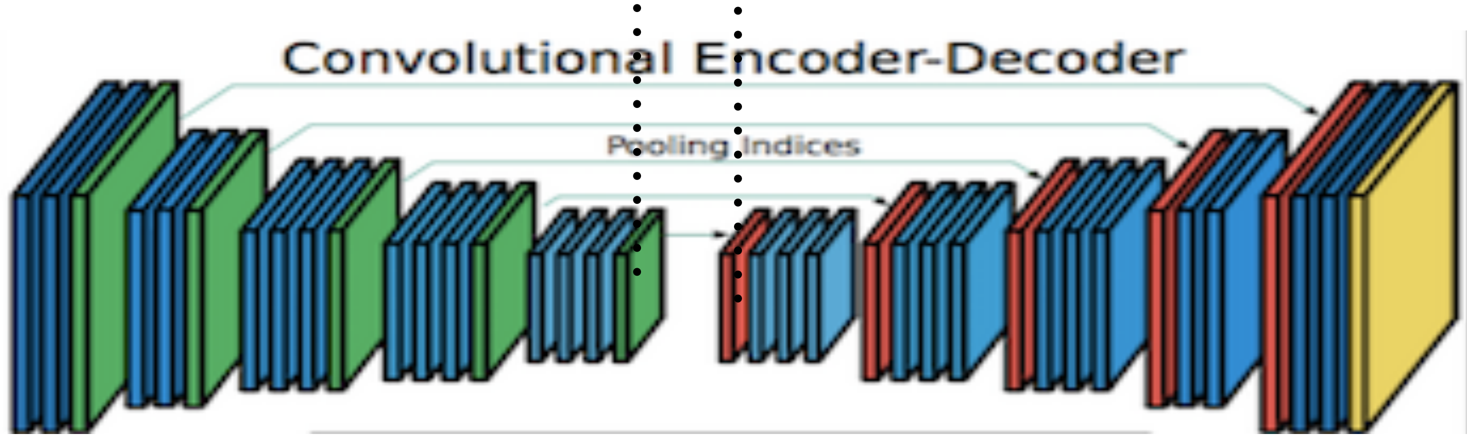


U-resNet(https://github.com/DeepLearnPhysics/u-resnet.git)

Hybrid of the U-Net and residual network



U-Net (arxiv: 1505.04597)
and Res-Net (arxiv: 1512.03385, arxiv:1603.05027)



Application: Neutral pion reconstruction

To test application, we can consider the signal: μ + 1 π0 + X (includes mesons).

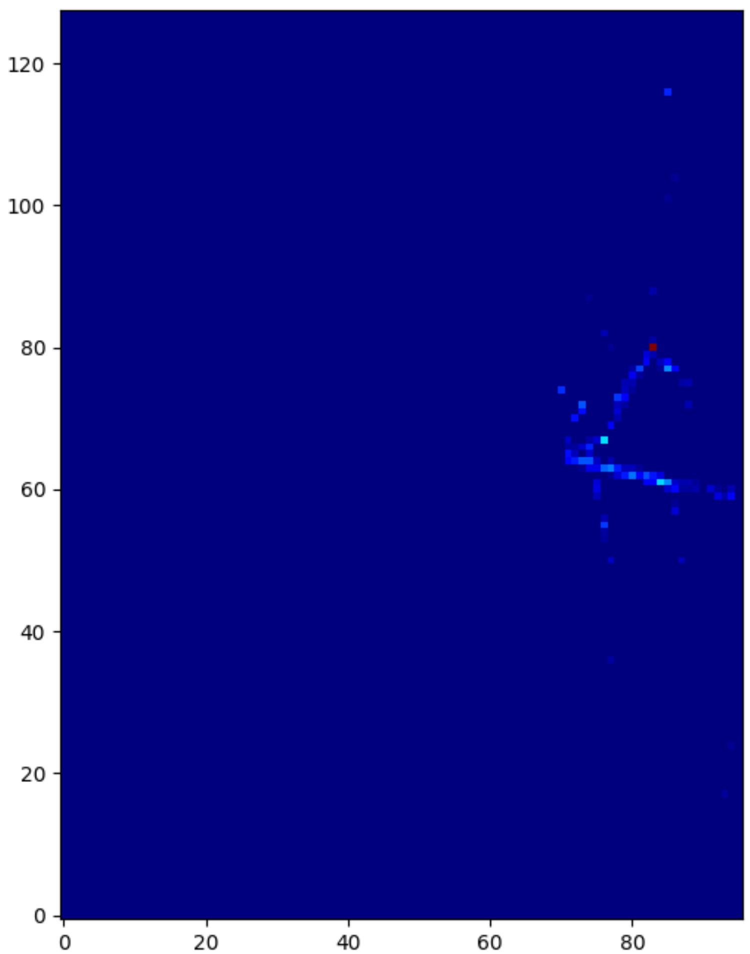
In the x view, Remove clusters from consideration where the the energy weighted pixel probability of being em-like is under some threshold.

Clusters are then used in blob formation via angle scan routine.

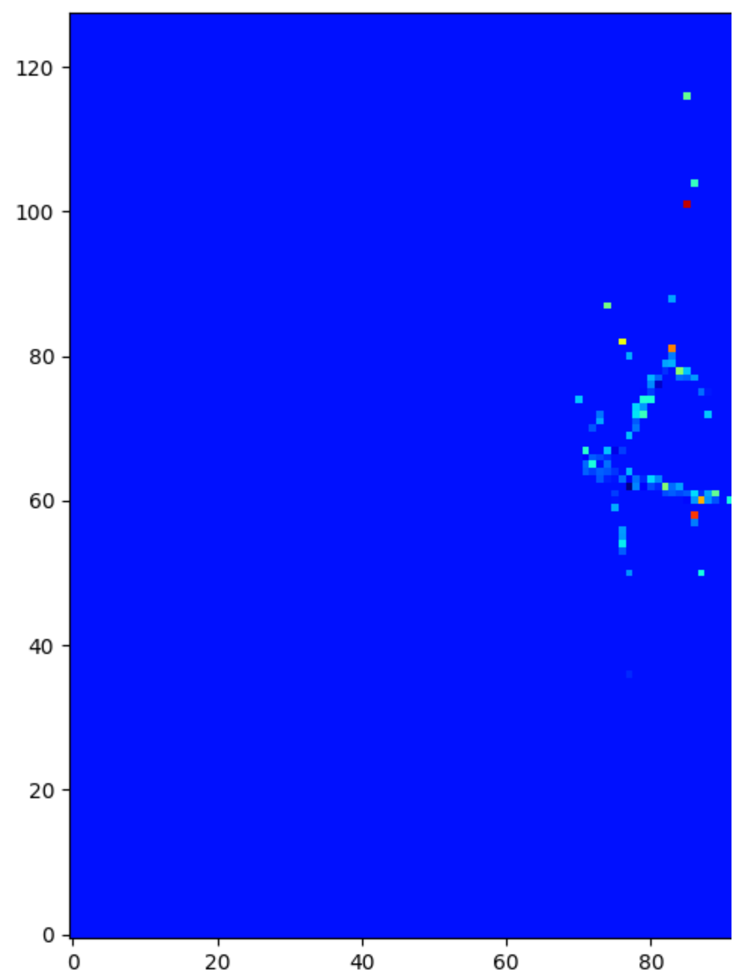
Step 1: Make images

We already had HDF5 image format. I converted them into LarCV

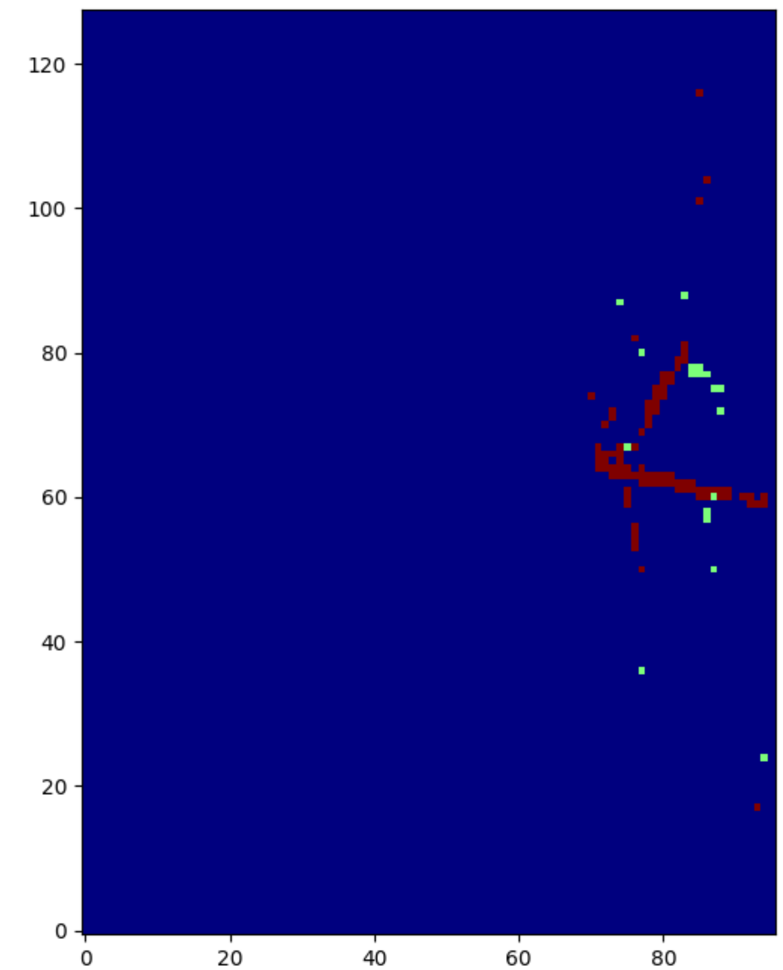
Start with three class image : EM , non-EM , and background(zero label)



Data image:hit (X-view)



Data image :time (X-view)

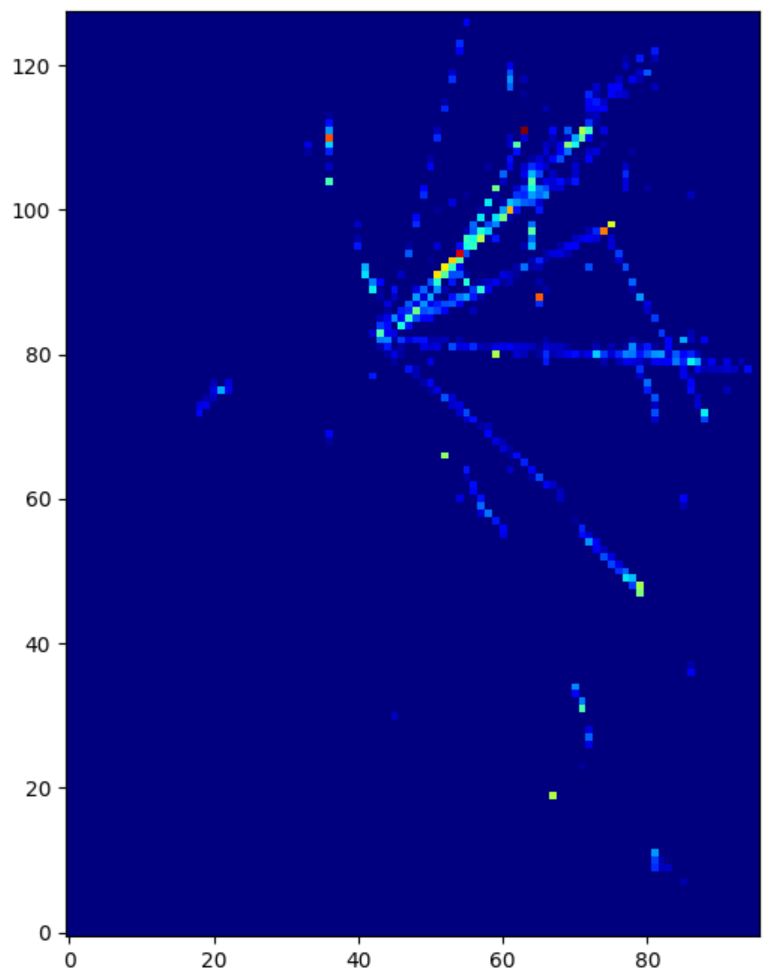


PID image(X-view)

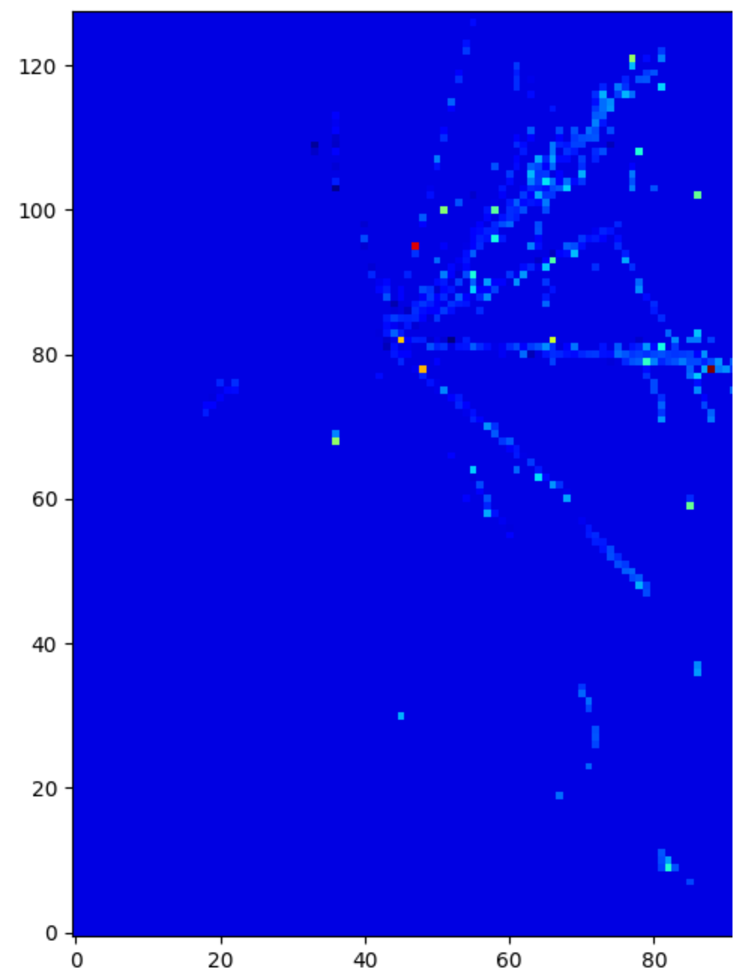
Step 1: Make images

We already had HDF5 image format. I converted them into LarCV

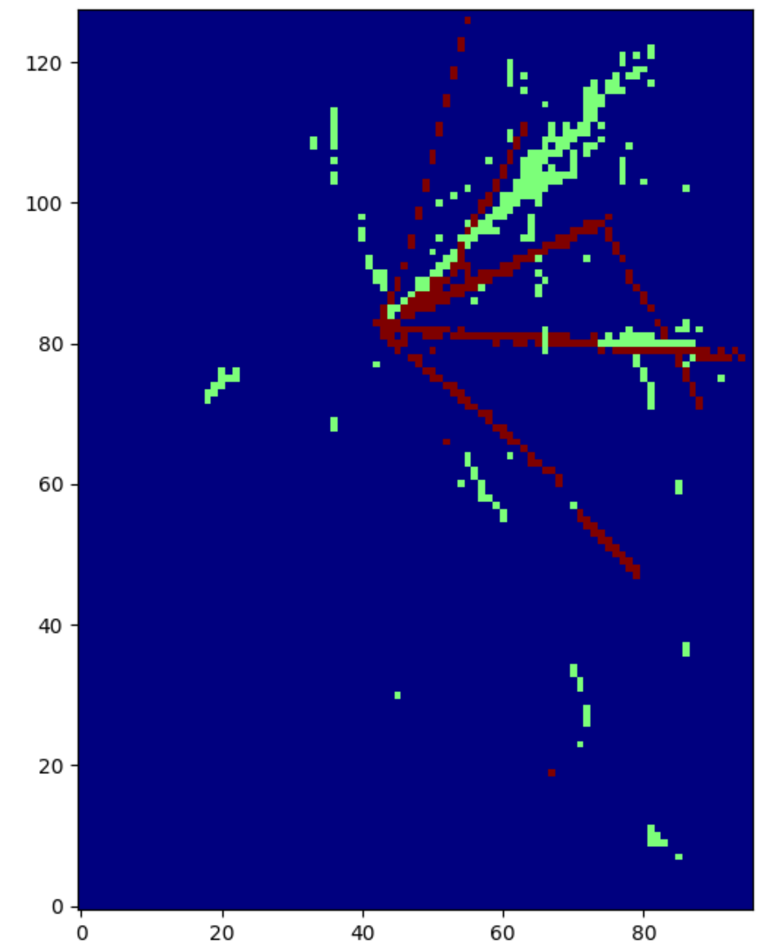
Start with three class image : EM, non-EM , and background(zero label)



Data image:hit (X-view)



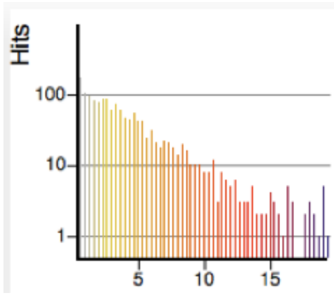
Data image :time (X-view)

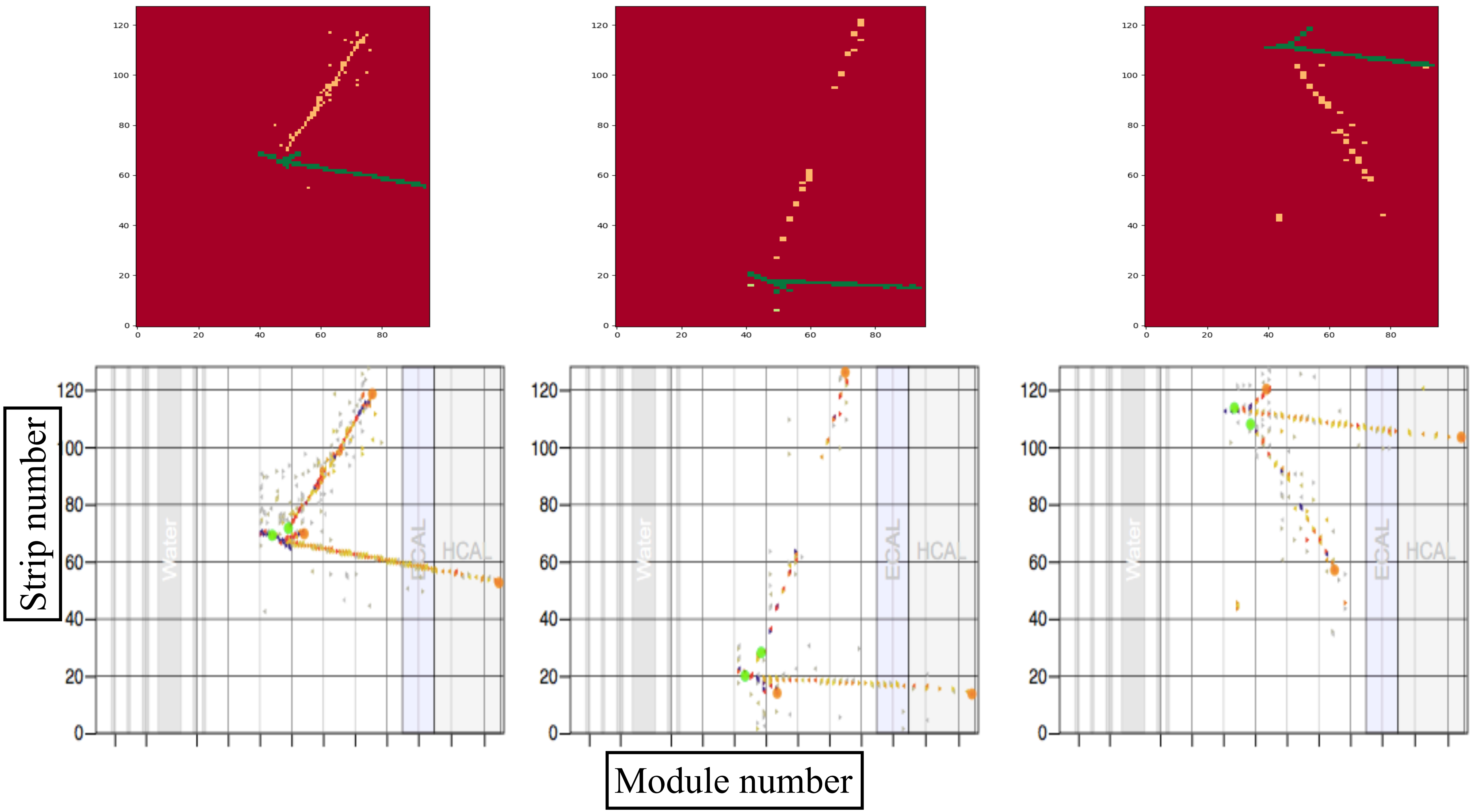


PID image(X-view)

True vs predicted event

Model :layered model, 4class; Class 0-zero label,
class 1- EM like, class 2- Neutron, class 3- Non-EM-like





Anushree Ghosh, UTFSM, Chile

Train and validate the network

Trained the network for x view separately.

Training sample: Monte Carlo events

- -Full detector configuration, only CC events
- -1M images for training
- -train over only X view
- -Also, we layered three views and train over them all together
- - Used both Hit and time informations (two channels)

Loss is calculated using softmax_with_logits to calculate the loss per pixel. Finally the loss is summed over all the pixel

Ran upto 15 epoch

4 classes (non zero training accuracy 93%)

Tested on me1F MC sample. For creating the confusion matrices, we use 50K events.