Application of Machine learning in vertex finding@MINERvA

Anushree Ghosh, UTFSM, Chile

On behalf of the MINERvA collaboration

State of Nu-tion premeeting

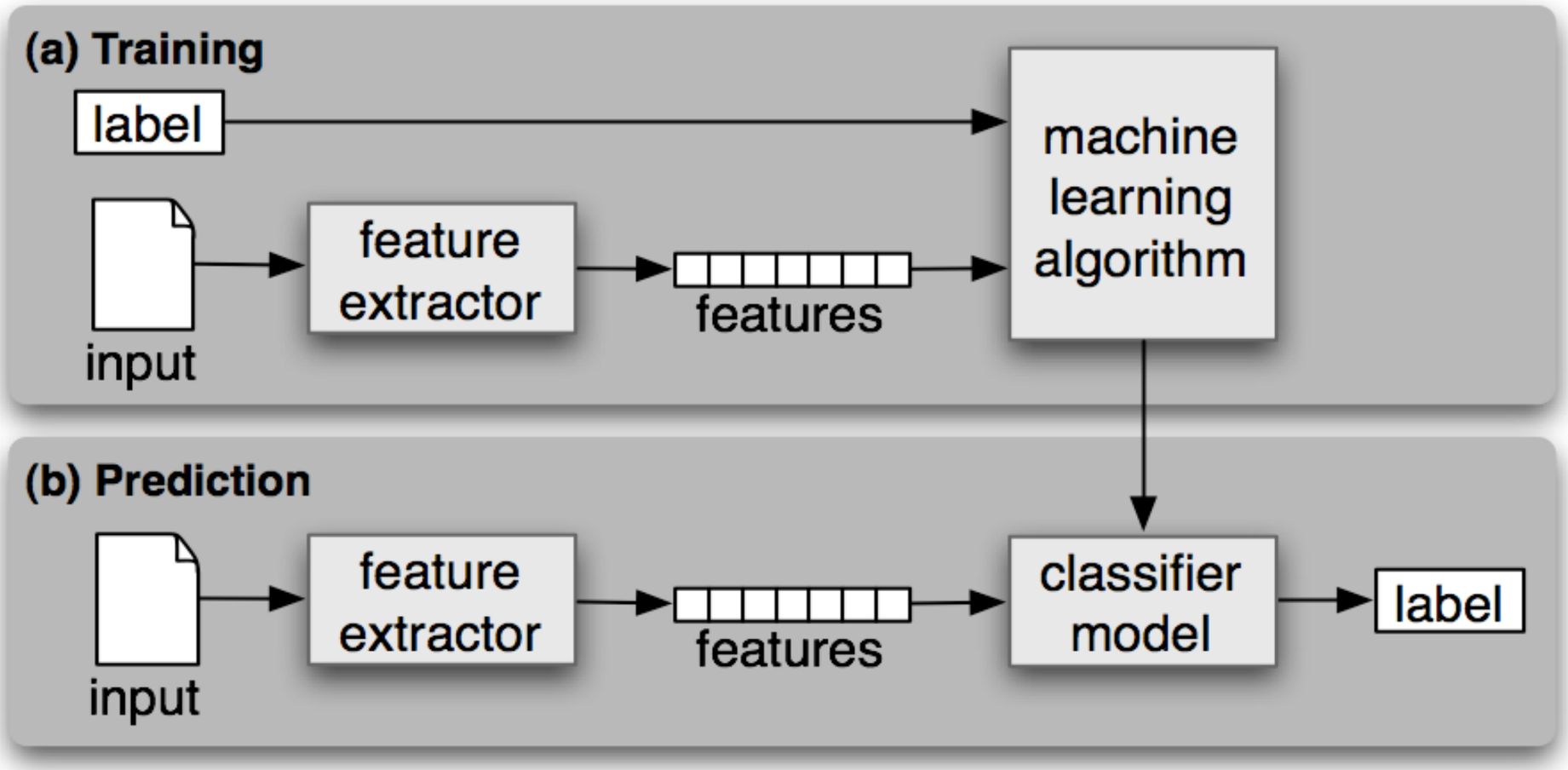
Date: 06/23/2017

Outline

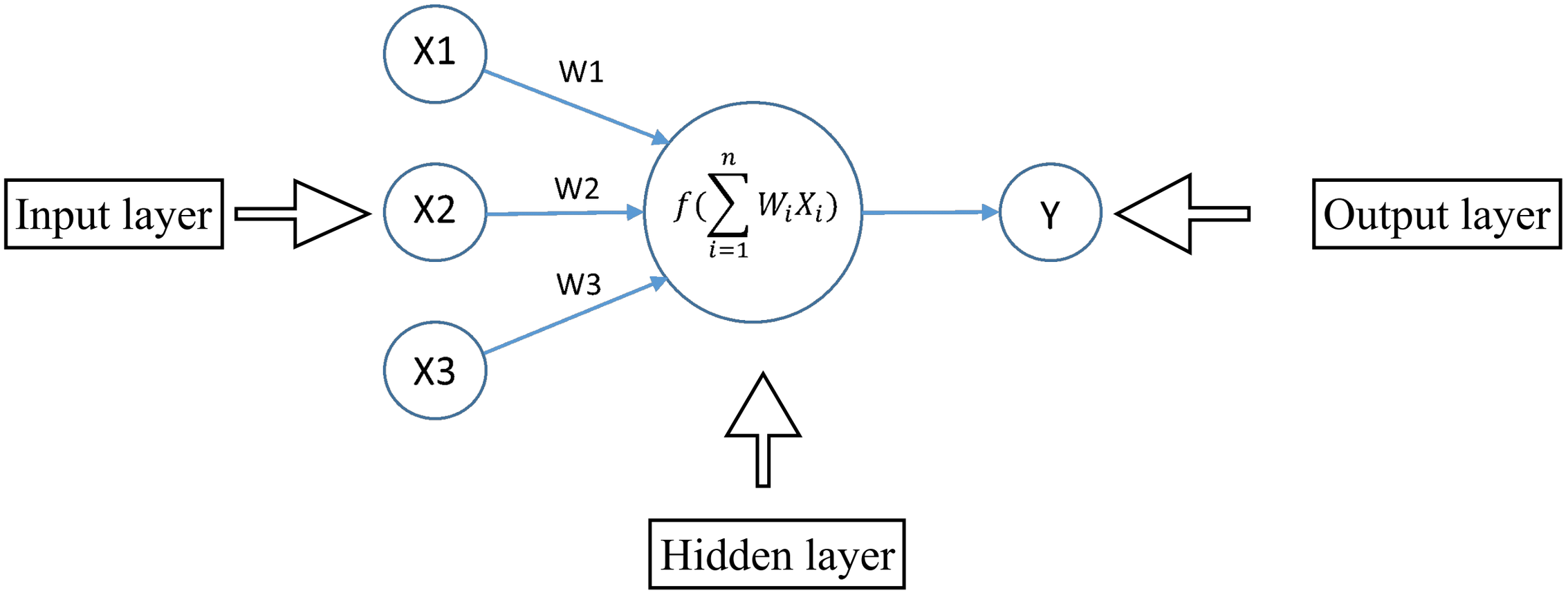
- Brief discussion on Deep learning neural network
- Application of Machine learning in vertex finding
- Preliminary results: comparison of performance between Machine learning and traditional reconstruction method
- Introduction to Domain adversarial neural network
- Plots showing comparison between Convolutional neural network and Domain adversarial neural network

Machine Learning

- We take some data (our case Monte Carlo events)
- train a model on that data
- and use the trained model to make predictions on new data. (our case, it is real data or different MC)



Basic neural network

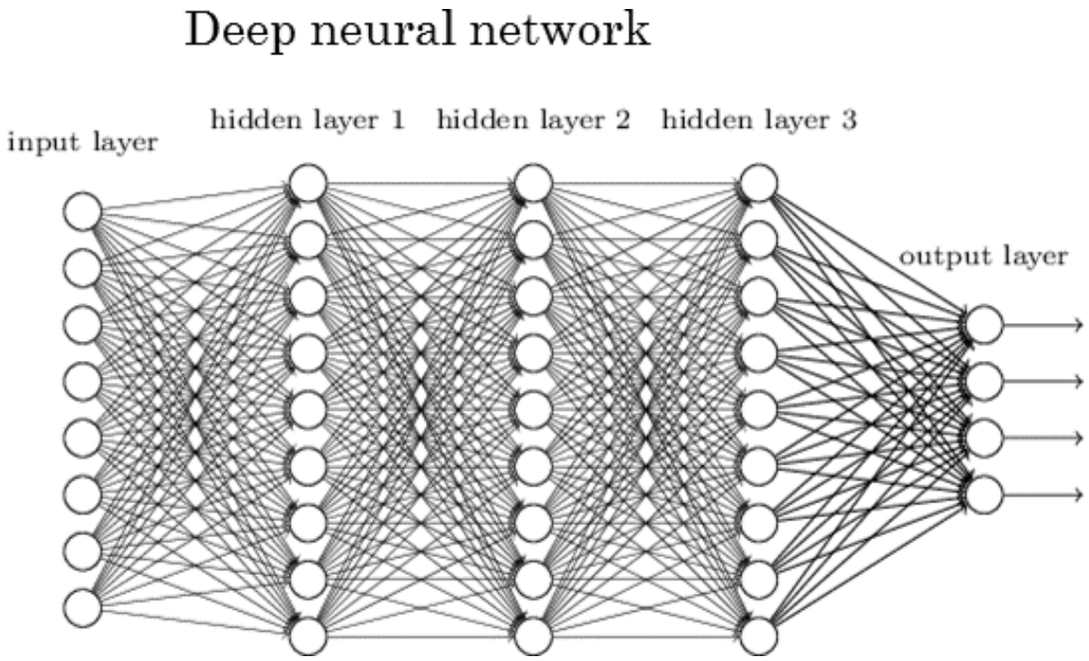


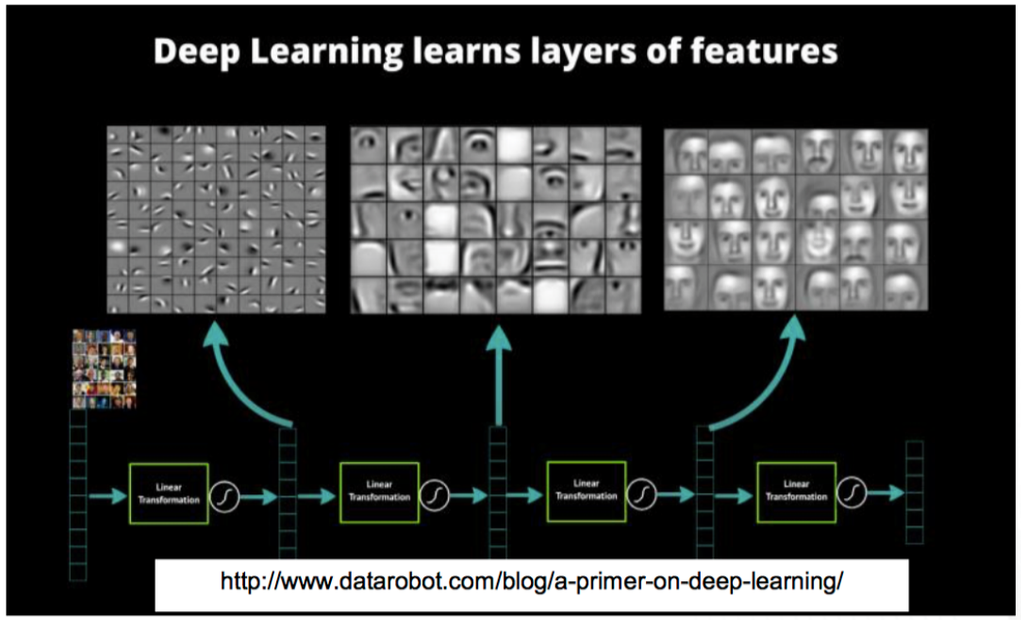
- The single artificial neuron does a dot product between x (input data) and w (weight), then add a bias.
- the result is passed to an activation function that adds some non-linearity.
- The non-linearity allows different variations of an object of the same class to be learned separately.

Deep learning neural network

neural network with many hidden layers

- Each layer learn a concept with the output of the previous layer.
- The layer does not need to learn the whole concept at once, but actually build a chain of features that build that knowledge.

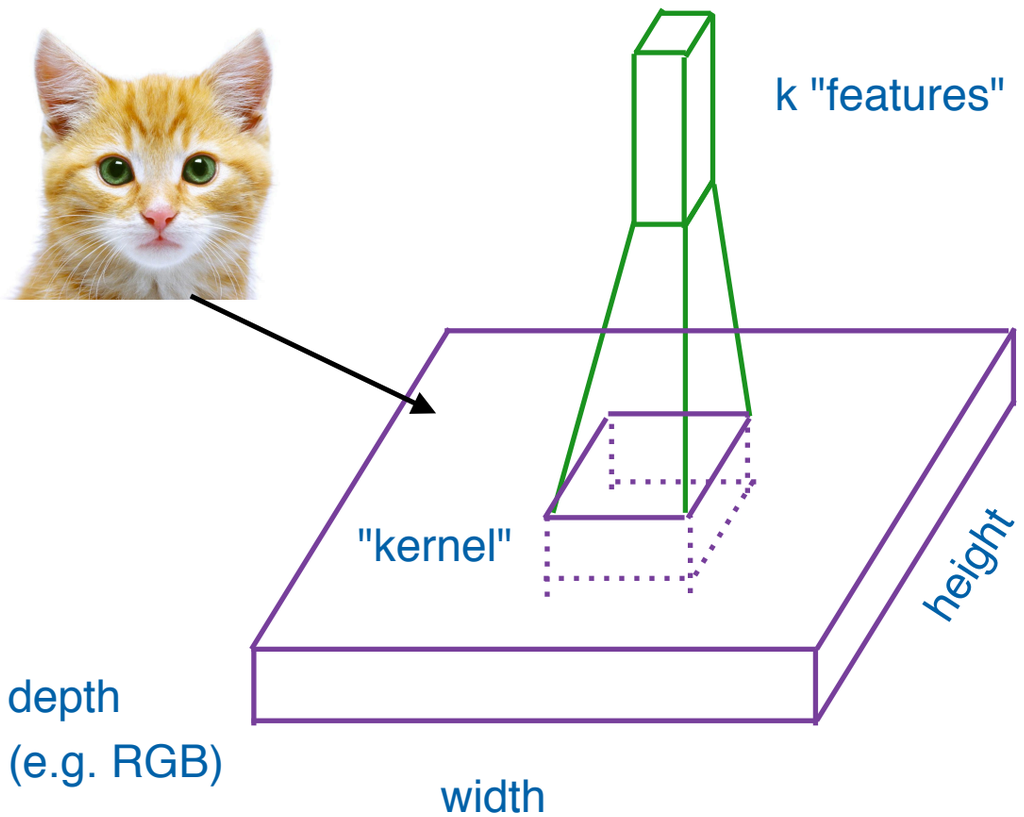


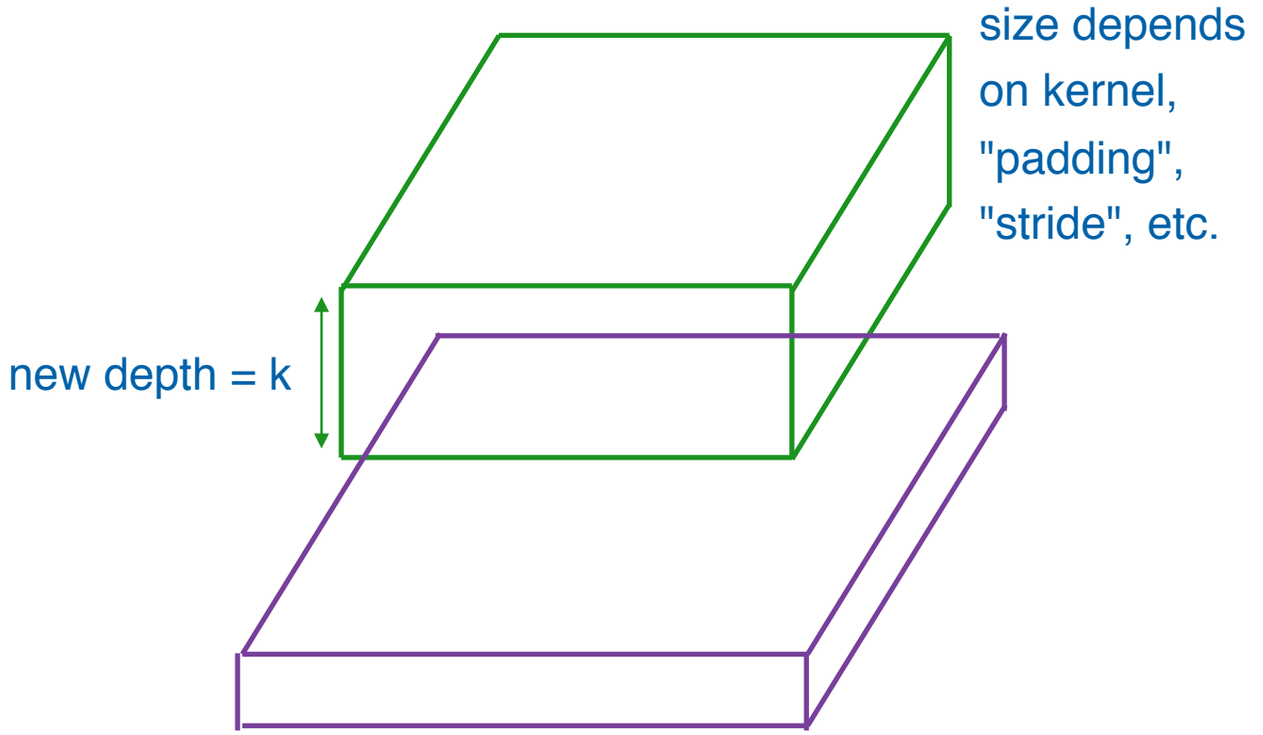


"hierarchal model" - low levels in the network build low-level representations that get combined in higher levels

Convolutional neural network (CNN)

- well-suited to image recognition and computer vision problems.
- Normal neural network with fully connected neuron, needs huge number of parameters to fit images
- Convolutional network share the parameters across the space-> fewer weights.

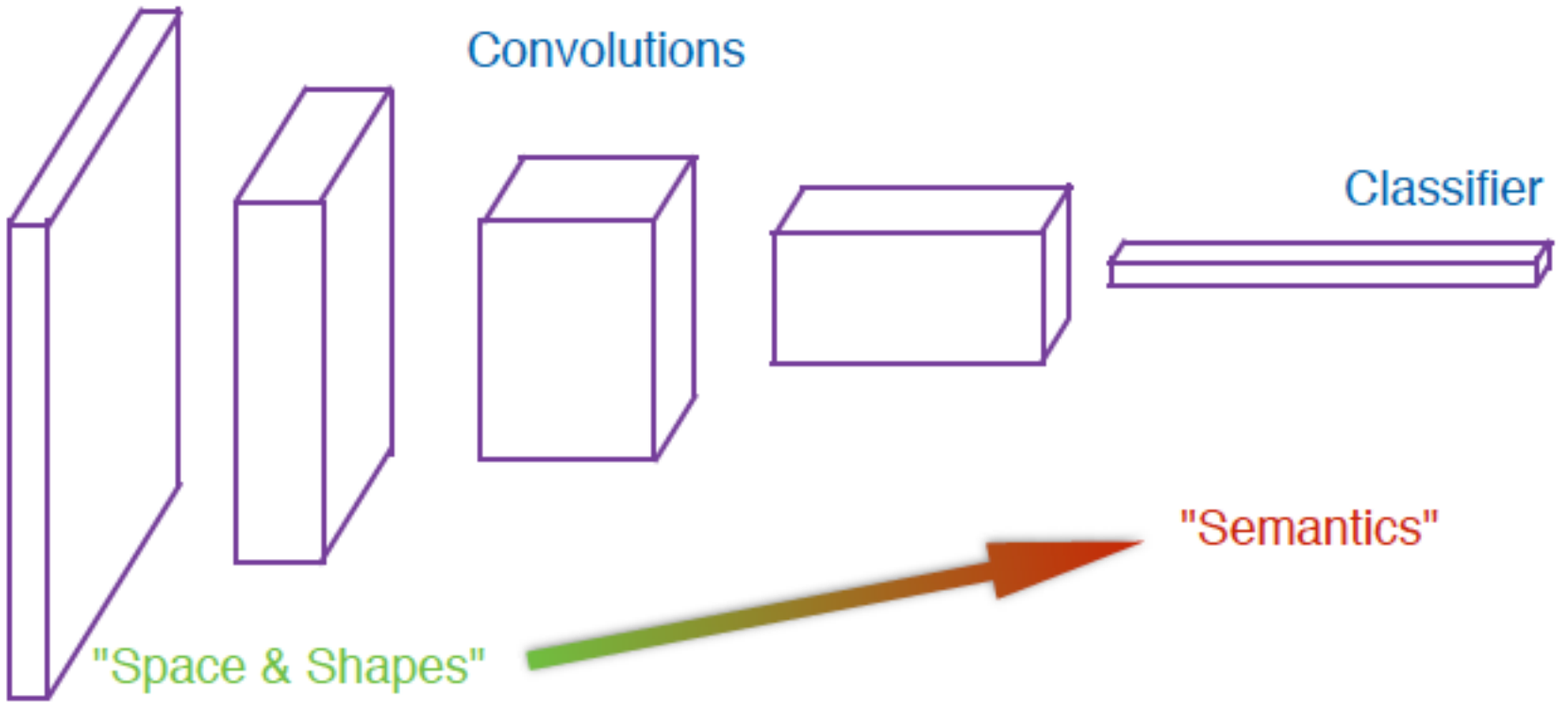




Scan the ***same*** kernel
over the full input space.

Convolutional neural network (CNN)

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

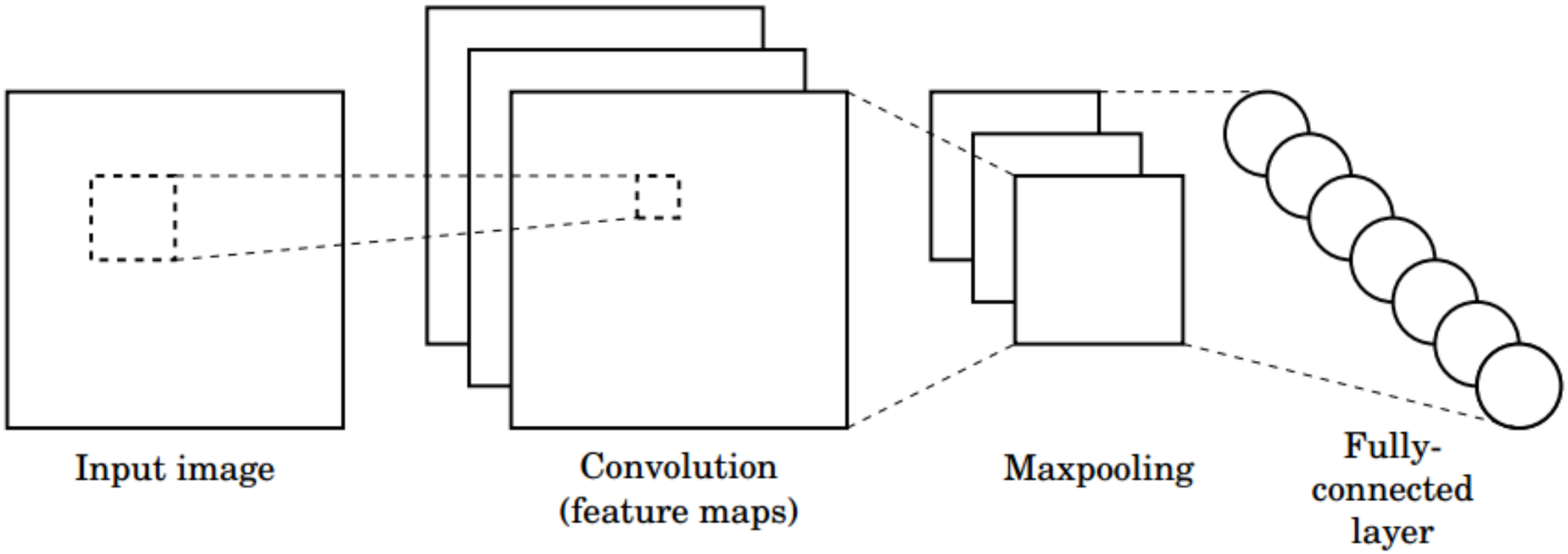


Deep learning neural network: layers commonly used

- **Convolutional layers:** Normal neural layer - Made up of neurons with learnable weights. Convolution layers share weights across neurons

- **Pooling:** as the number of feature maps grow, the complexity of the network explodes. Pooling reduces the "spatial size" or amount of parameters and computation in the network.

- **Fully connected layer:** Neurons in a fully connected layer have full connections to all activations in the previous layer



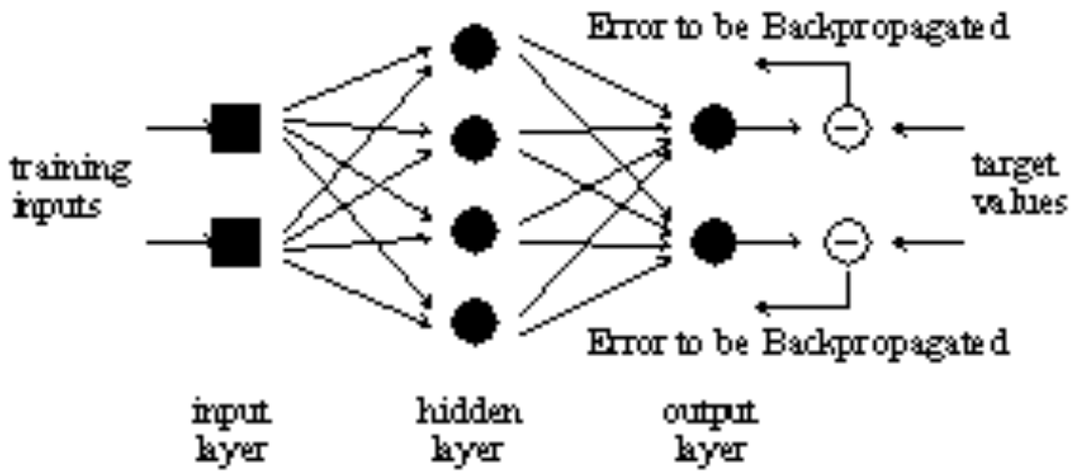
Deep learning neural network: layers commonly used

- **Dropout layer:**Randomly drop connections between layers on each pass during training to eliminate co-adaptations in the network

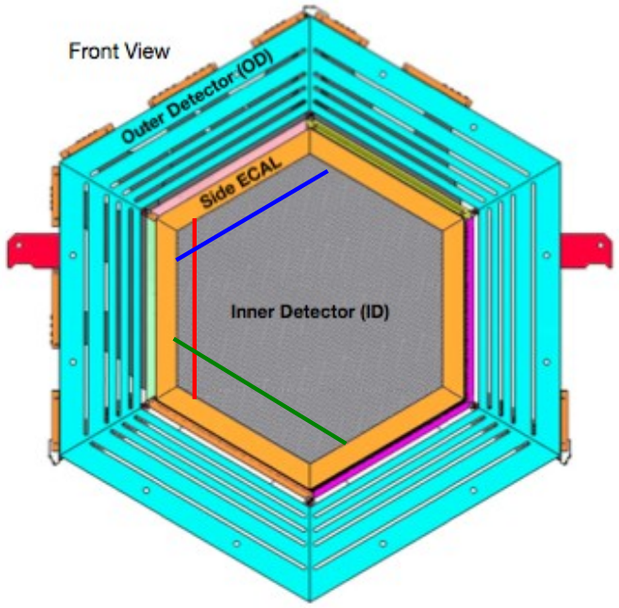
- **Loss function :**Loss function indicate the penalty for an incorrect prediction

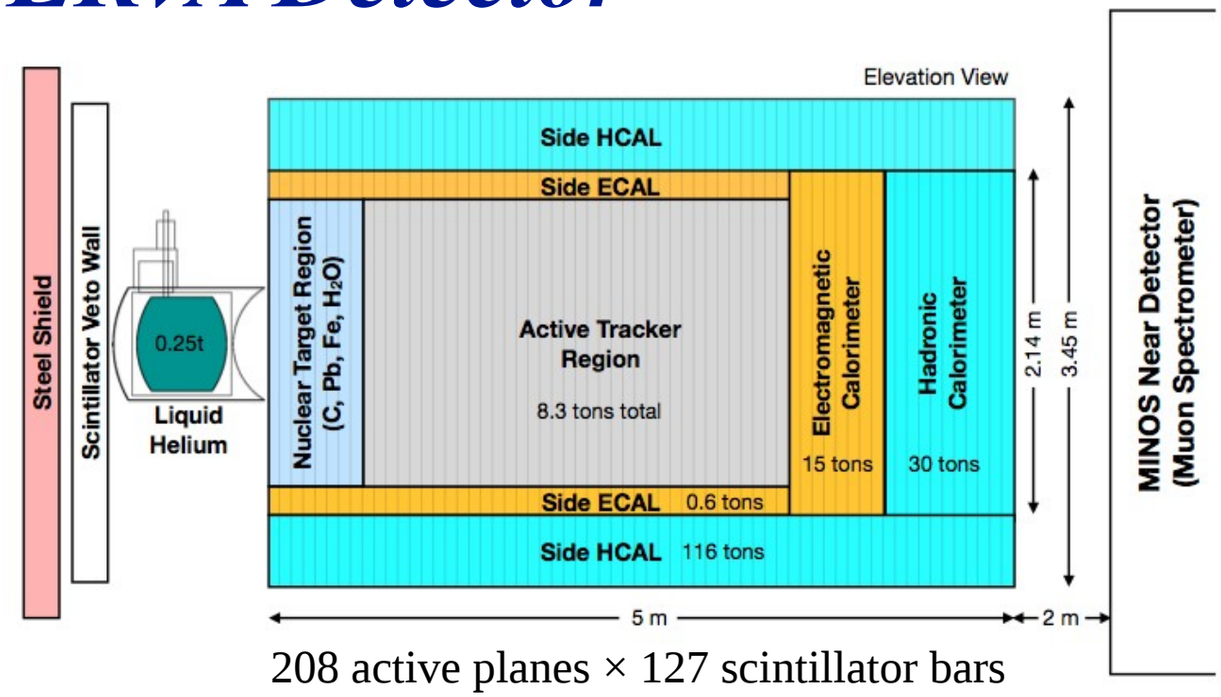
L(y`, y) : loss for prediction y` instead of y

- Loss has to be minimized to make better network. The loss is calculated on training and validation and it indicates how well the model is doing for these sets.

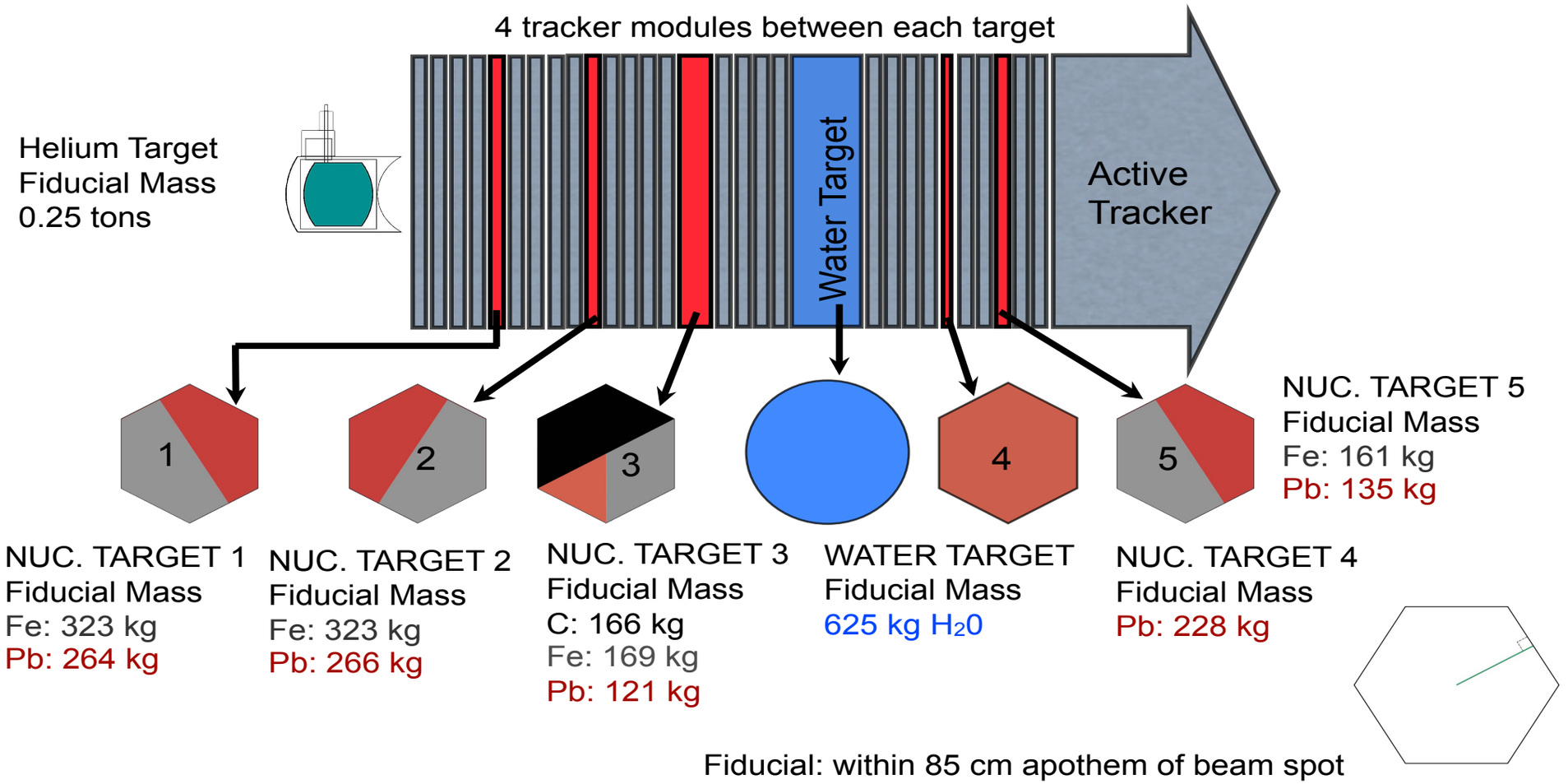


MINERvA Detector



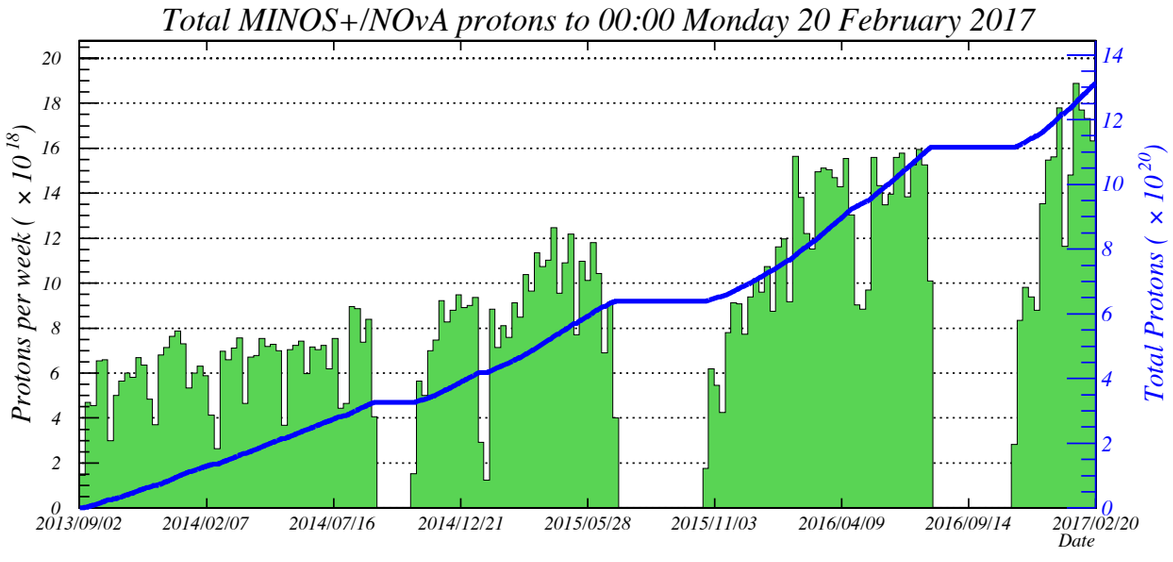


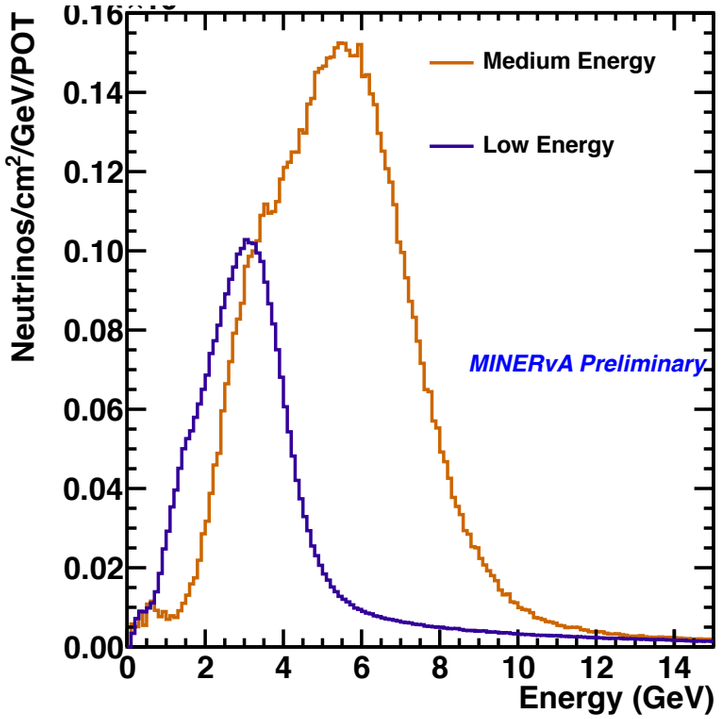
208 active planes × 127 scintillator bars



Fiducial: within 85 cm apothem of beam spot

NuMI beam: medium energy regime

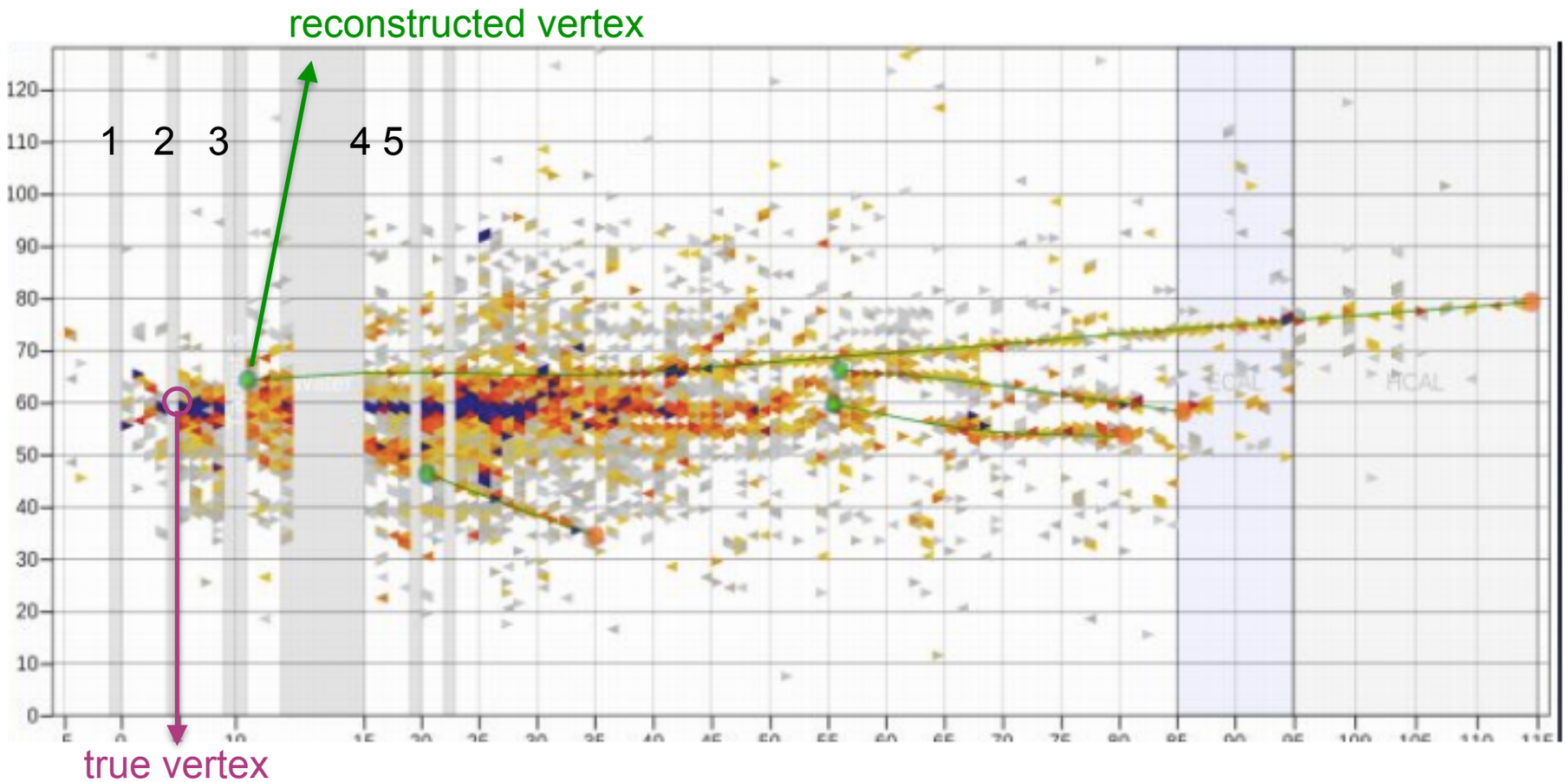




- NuMI beamline currently running with increased beam energy mode which peaks at ~6 GeV (ME mode).
- We have taken ~12E20 POT in neutrino mode and currently taking data in anti-neutrino mode.
- About factor of 3 increase from LE data at 3.9E20 POT!
- Higher statistics yields improve comparisons across nuclei
- The peak of energy now moves to the DIS-rich kinematic region. *Access to expanded kinematics and nuclear structure functions.*

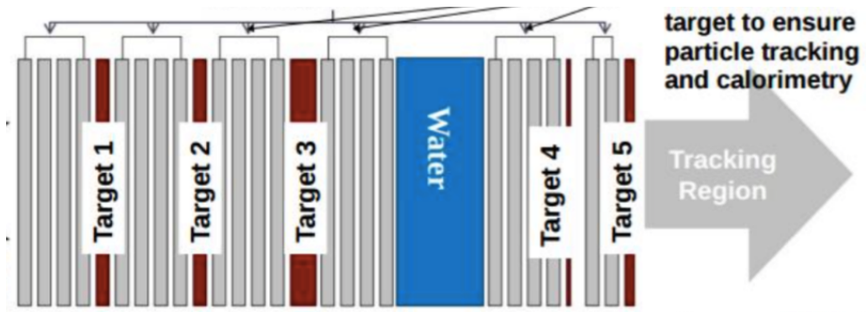
Problem with vertex finding

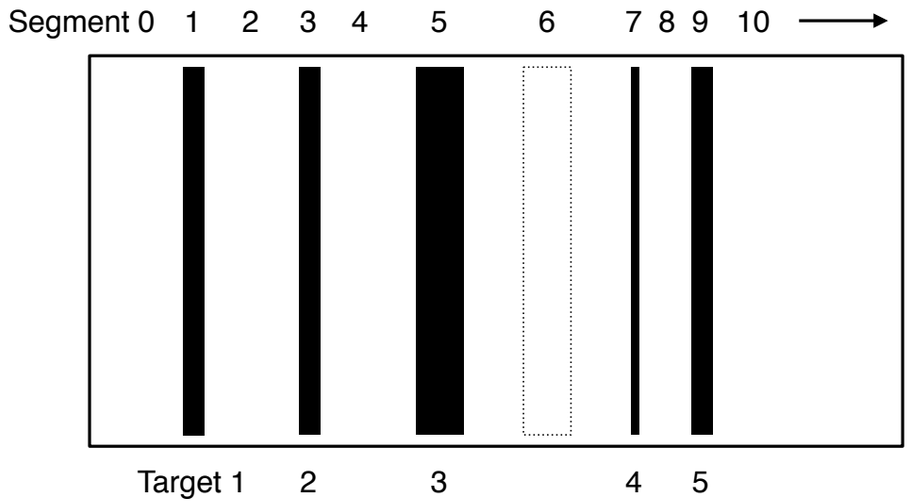
- With the increase of our beam energy, there is an increase in the hadronic showers near the event of interactions.
- Cause more difficulty in vertexing with increase rates of failure in getting the correct vertex position:
  - Events with high invariant hadronic mass tend to have tracks that are created by secondary interactions or decays.
  - Shower activity occludes the vertex region.



Machine Learning Approach To Determine Event Vertex

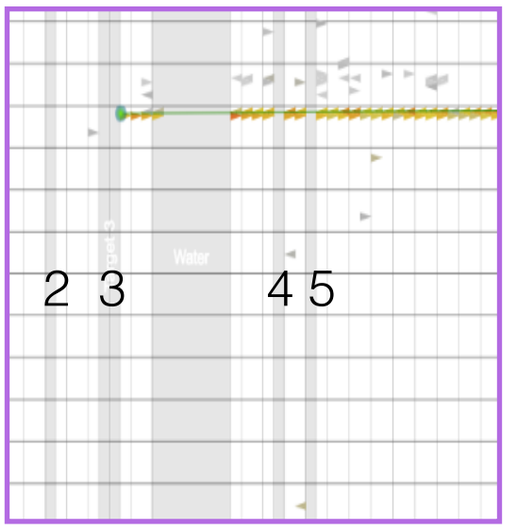
- Goal: Find the location of the event vertex
- Treat localization as a classification problem: DNN gives prediction which segment out of the 11 segments(or which plane out of 67 plane number) an interaction is from.



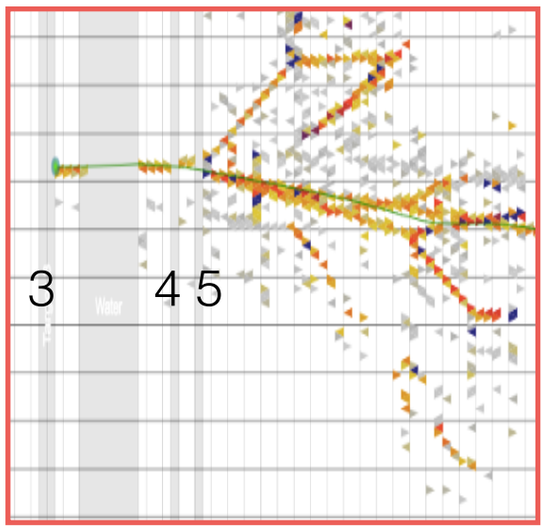


Events in MINERvA are easily represented as images.

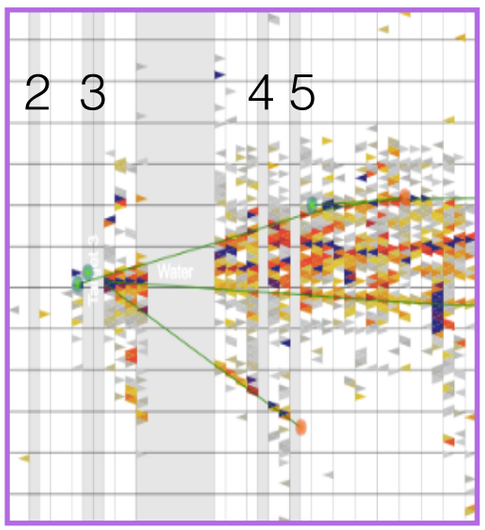
Challenges: Different type of interaction ➔ different characteristics



Single Track

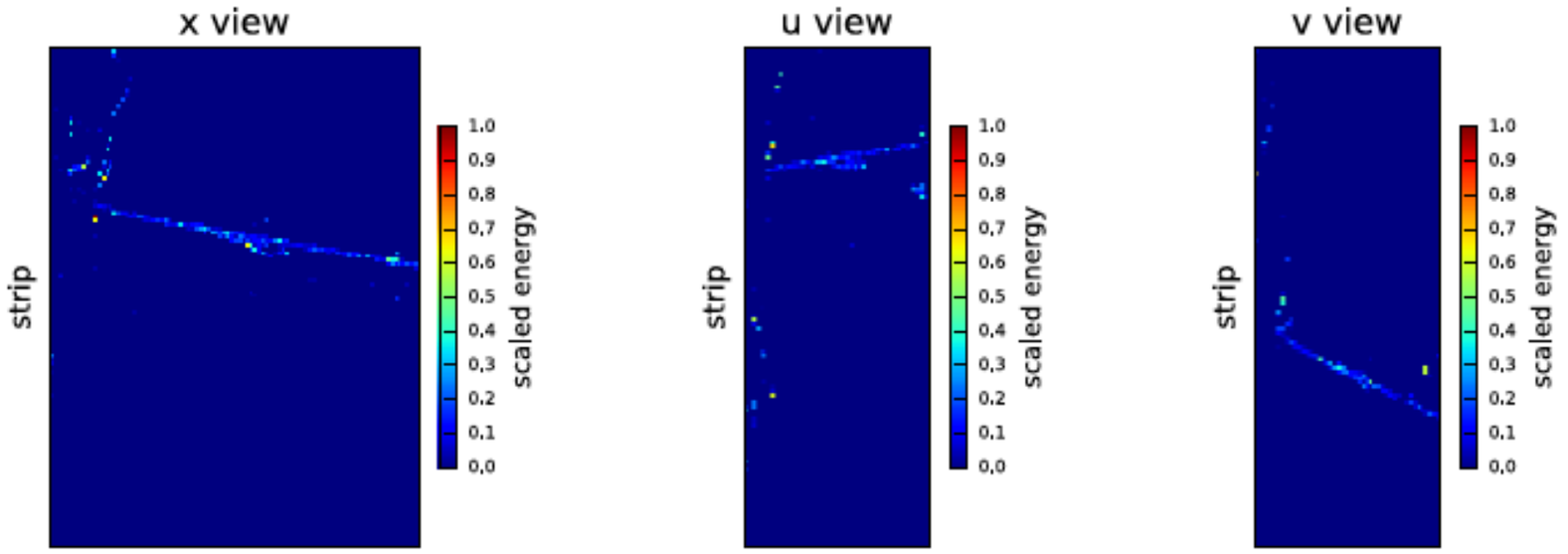


Backward Track



Large Shower

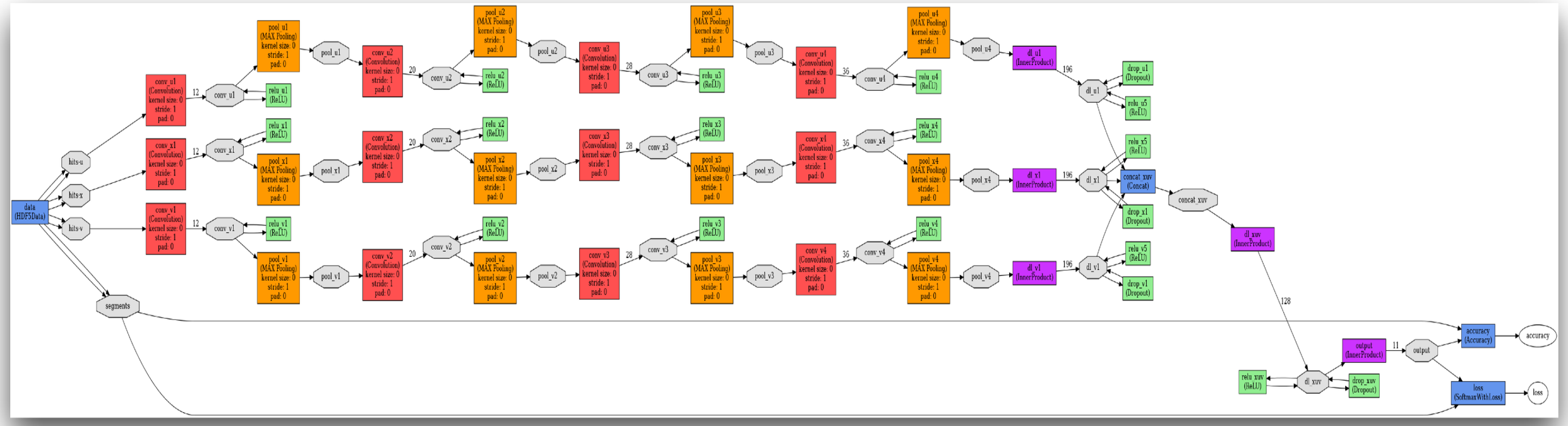
- The ntuple is skimmed to make image files which are input to the neural network
- Input images to the network: three views: X view, U view, V view



Event images from HDF5 (a high performance data storage format)

Network structure

- We have three separate convolutional towers that look at each of the X, U, and V images.
- These towers feature image maps of different sizes at different layers of depth to reflect the different information density in the different views.
- The output of each convolutional tower is fed to a fully connected layer.
- The output of those fully connected layers is concatenated and fed to another fully connected layer, and then that is fed to a softmax layer.

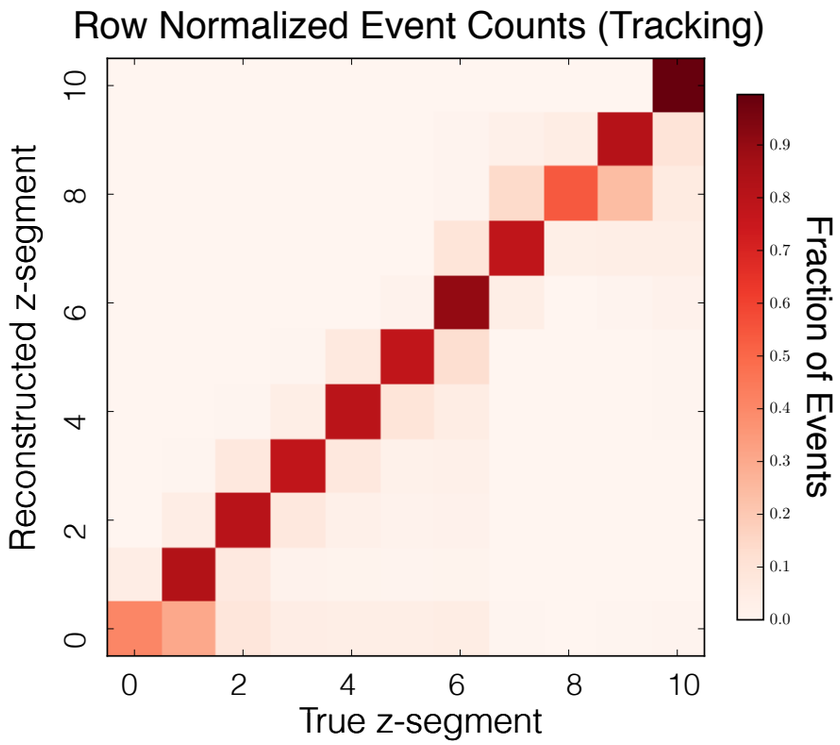


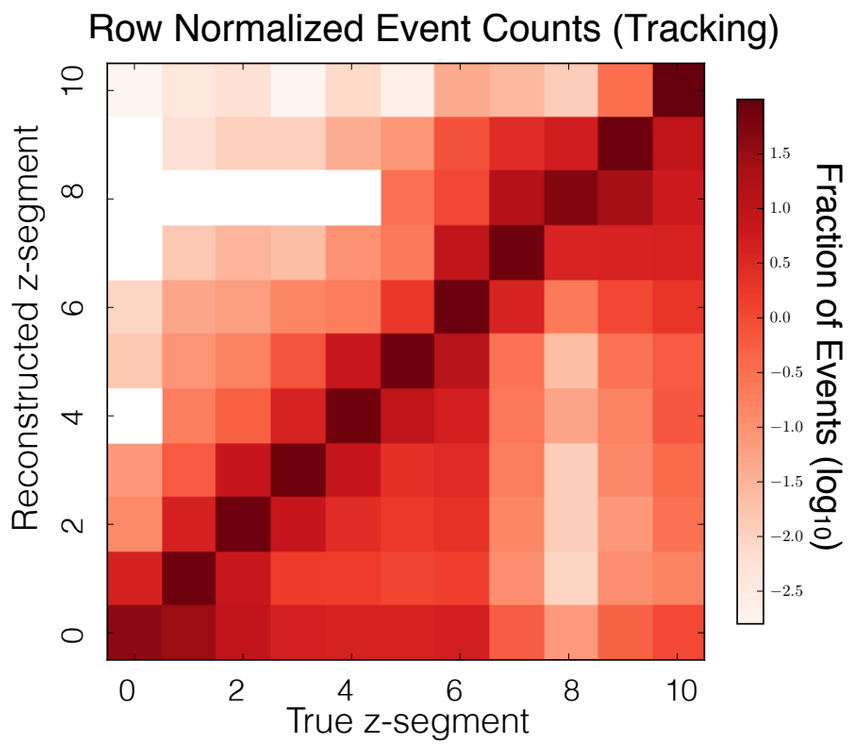
We train using both Theano and Caffe.

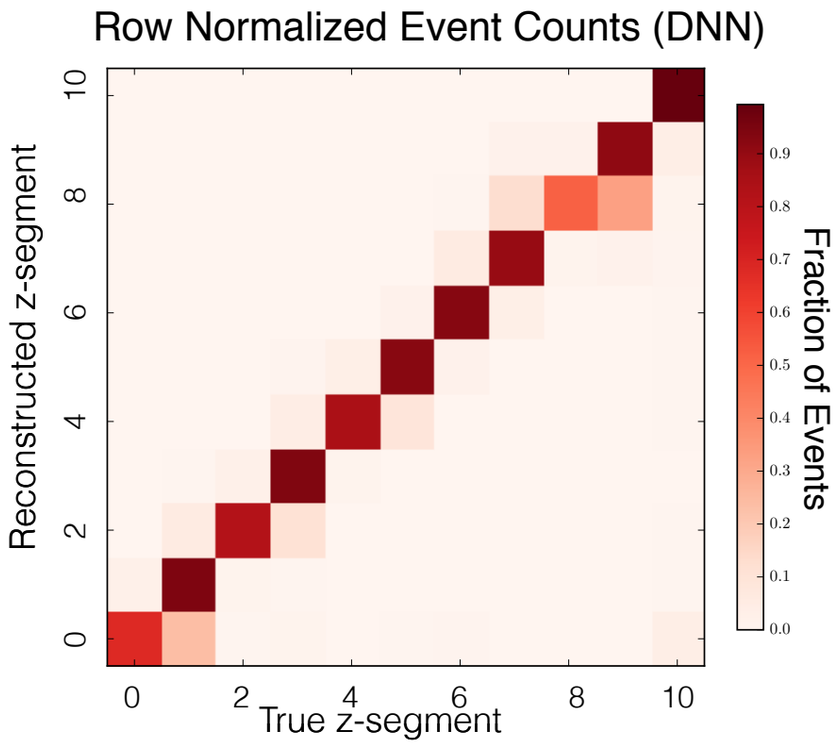
Theano: pure python API

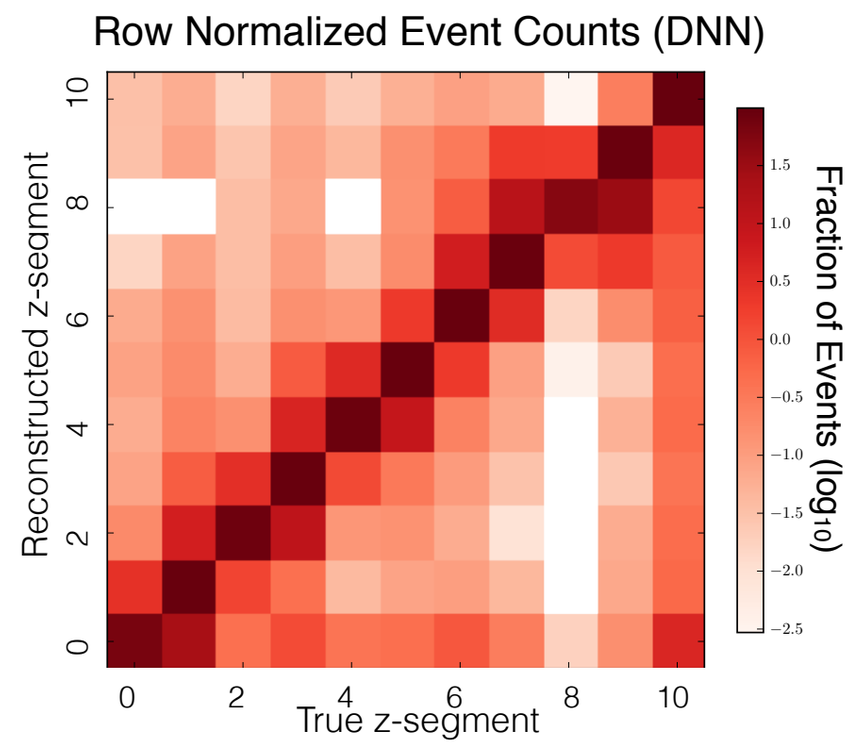
Caffe: C++ and python API

classifying events in 11 segments







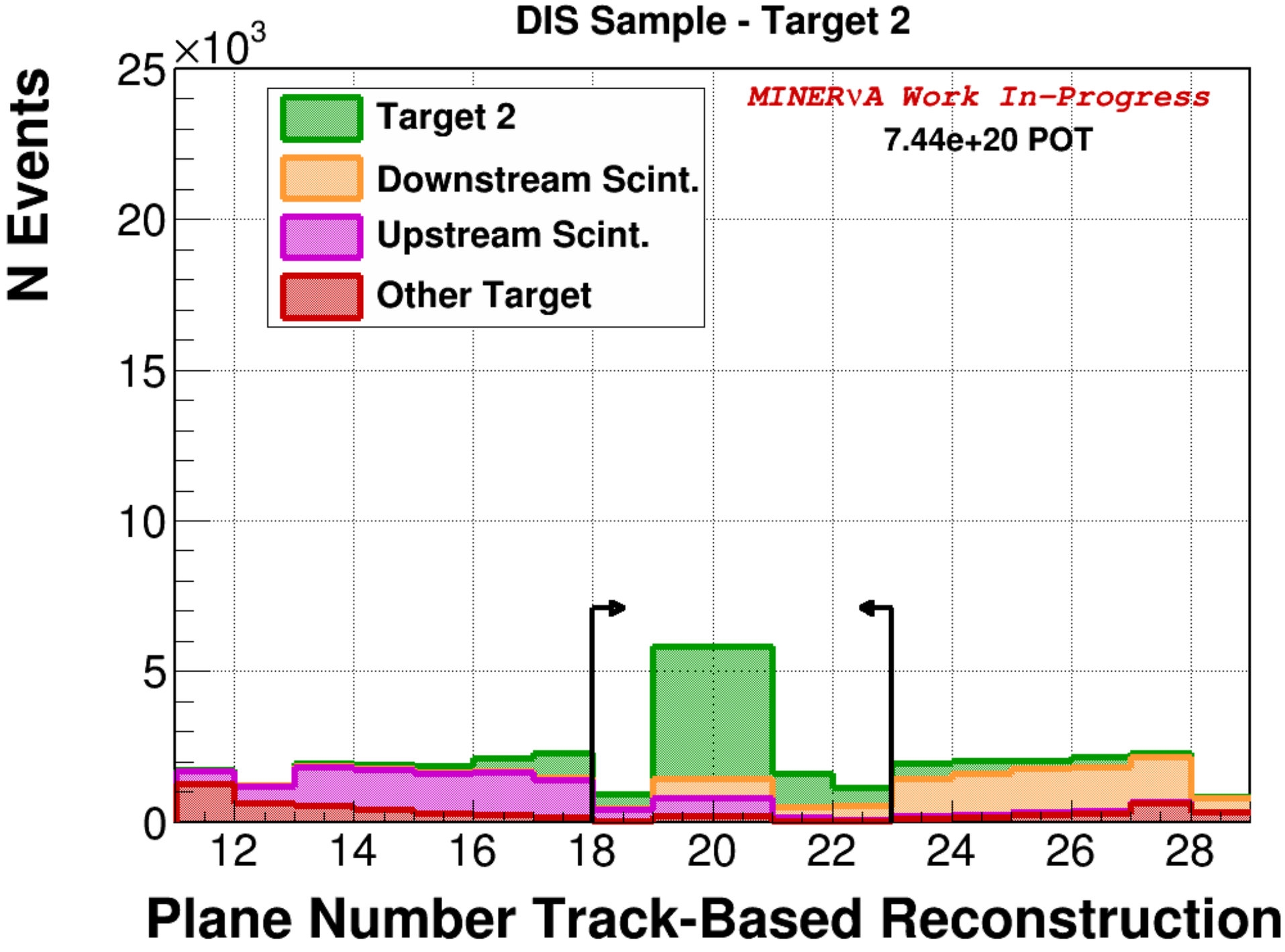


- **Row-normalized**: the difference in the probability of an event truly originating in a given segment for given reconstructed segment For a given reconstructed target

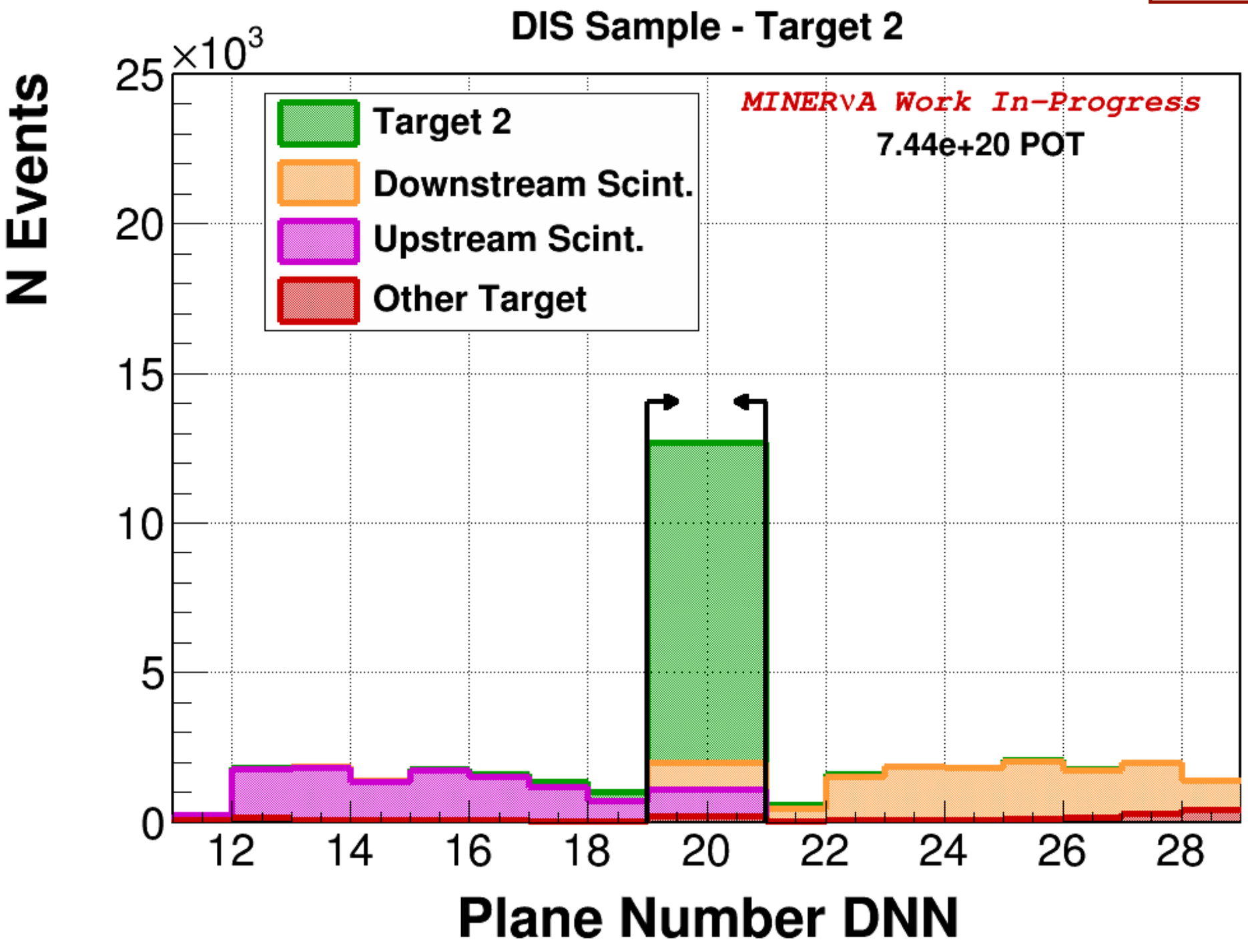
Odd segment: passive target
Even segment: active scintillators

classifying events in 11 segments

Target	Track-Based Row Normalized Event Counts +stat error (%)	DNN Row Normalized Event Counts+stat error (%)	Improvement+ stat error (%)
Upstream of Target 1	41.11±0.95	68.1±0.6	27±1.14
1	82.6±0.26	94.4±0.13	11.7±0.3
Between target 1 and 2	80.8±0.46	82.1±0.37	1.3±0.6
2	77.9±0.27	94.0±0.13	16.1±0.3
Between target 2 and 3	80.1±0.46	84.8±0.34	4.7±0.6
3	78±0.3	92.4±0.16	14.4±0.34
Between target 3 and 4	90.5±0.2	93.0±0.14	2.5±0.25
4	78.3±0.35	89.6±0.22	11.3+0.42
Between target 4 and 5	54.3±1.12	51.6±0.95	-2.7±0.15
5	81.6±0.3	91.2±0.18	9.5±0.34
Downstream of target 5	99.6±0.01	99.3±0.13	-0.3±0.02

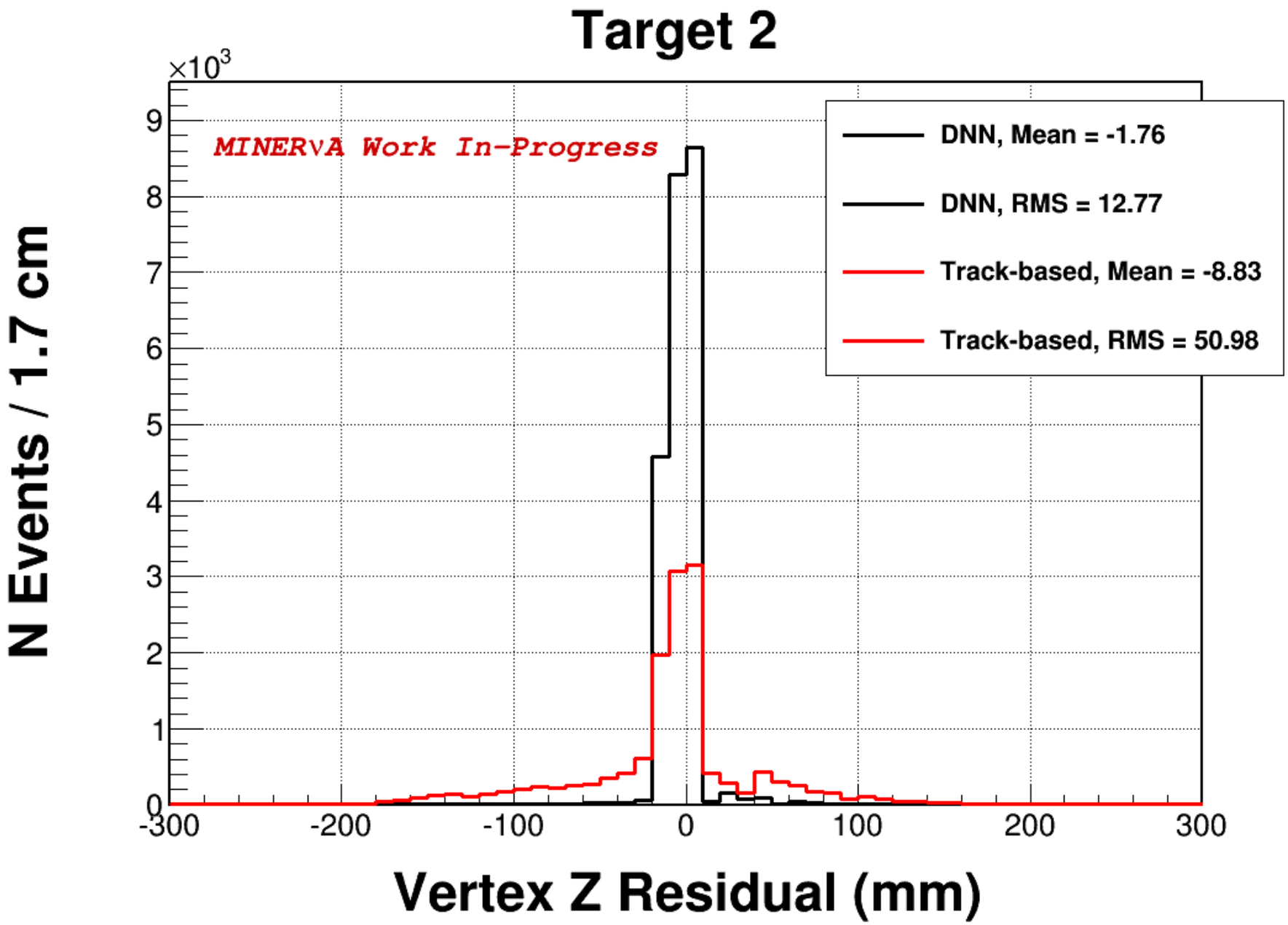

DIS Sample - Target 2
MINERvA Work In-Progress
7.44e+20 POT
Target 2
Downstream Scint.
Upstream Scint.
Other Target
N Events
×10^3
25
20
15
10
5
0
12
14
16
18
20
22
24
26
28
Plane Number Track-Based Reconstruction

Classifying events in 67 plane number



DNN vertex Z residual: True vertex Z - Z center of predicted plane

Track based vertex Z residual: True vertex Z - reconstructed vertex Z



Classifying events in plane number

Dealing with Systematic uncertainties

- With machine learning (ML) method, we are going to deal with systematics in traditional way, i.e., varying the Monte Carlo(MC) for different cases (different GENIE model, flux etc) and calculate the systematic error.

- We should calculate the systematic error coming from ML also. we need to train ML with few MCs, obtain the smearing over prediction, that will be the systematic error from ML.

Domain Adversarial Neural Network (DANN)

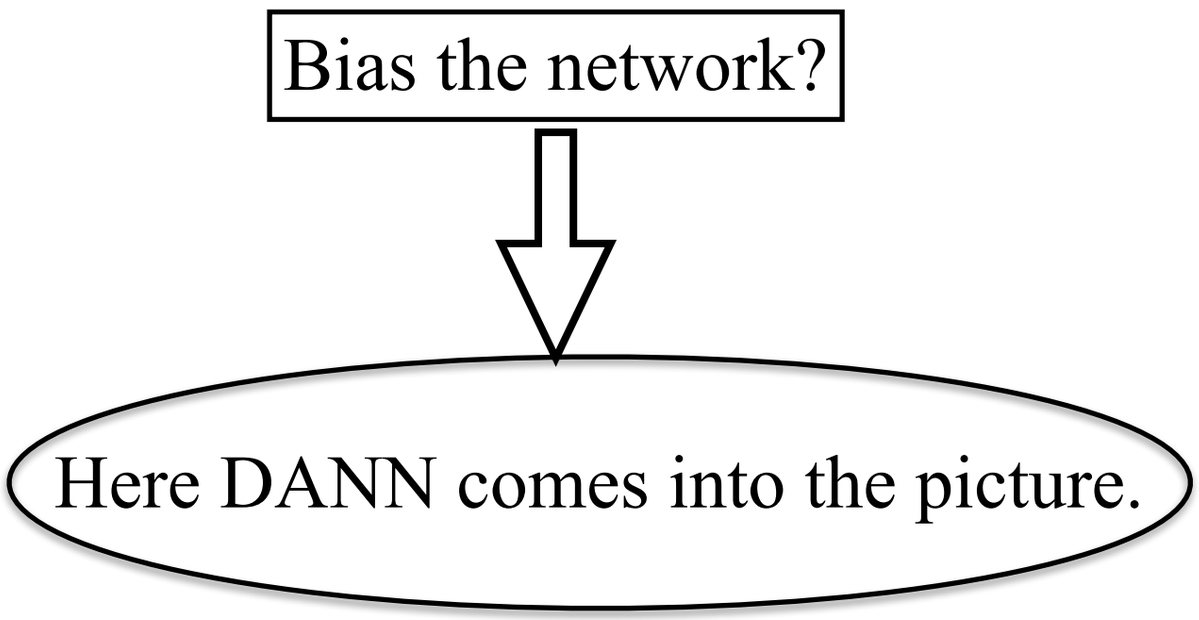
http://adsabs.harvard.edu/cgi-bin/bib_query?arXiv:1505.07818

Convolutional Neural Network(CNN):

- Train with labeled data: in our case it is Monte Carlo
- Test with unlabeled data: in our case it is real data

Limitation:

Labeled simulated data for training >> unlabeled real data for testing

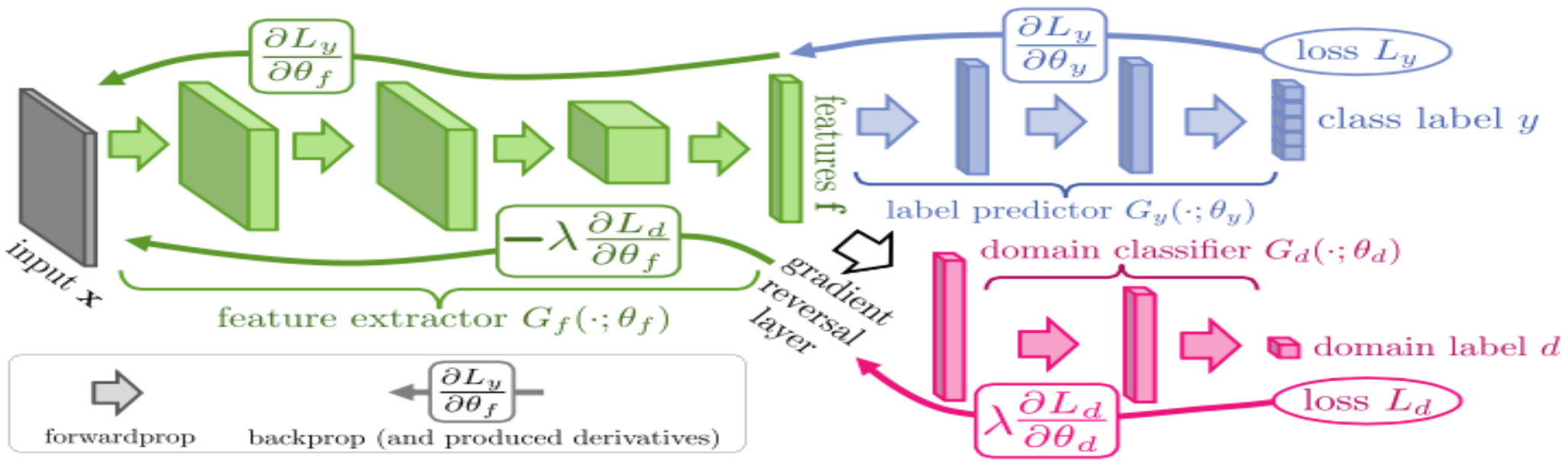


DANN

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



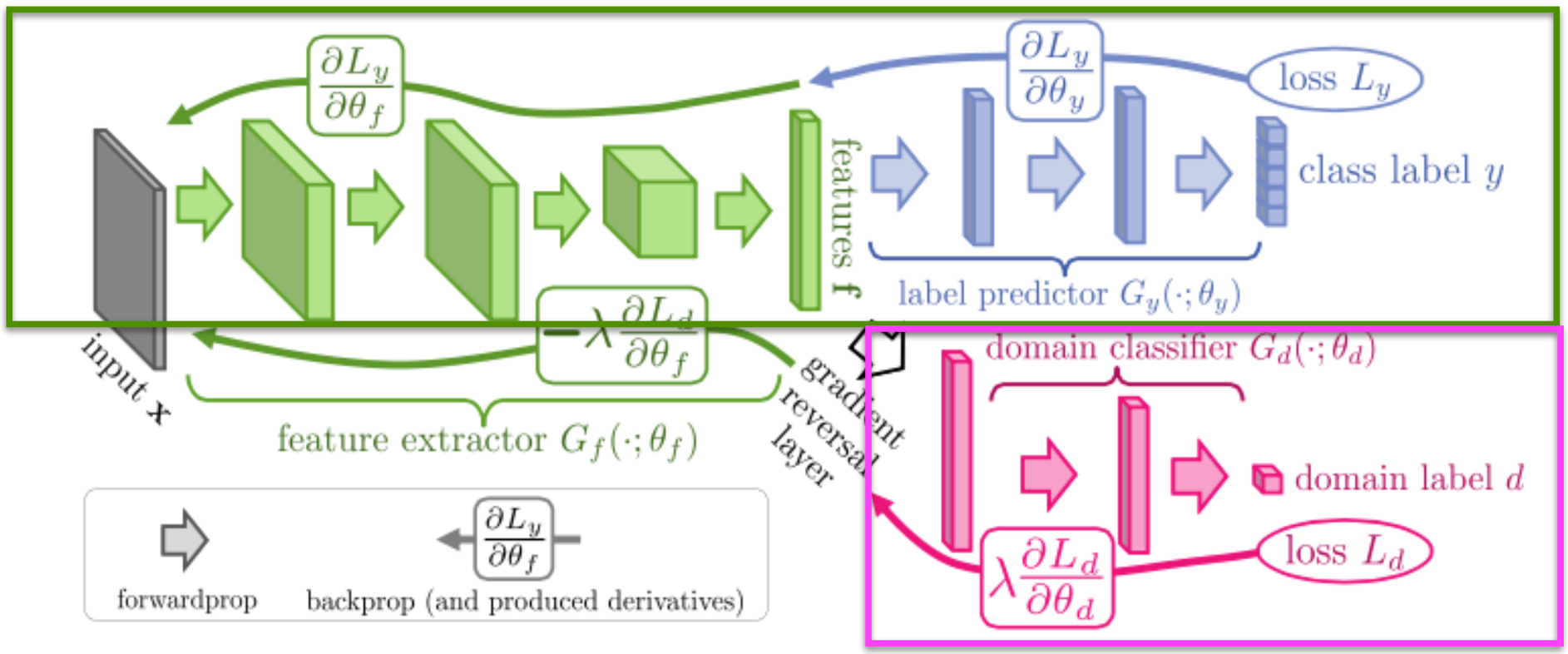
DANN

- Two classifiers into the network:

  Label predictor: output

  Domain classifier: works internally

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

- 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 are trying in a few ways to get the target sample having different features than source: changing the flux, physics model, kinematic division etc.

Changing the physics model (With and W/O FSI)

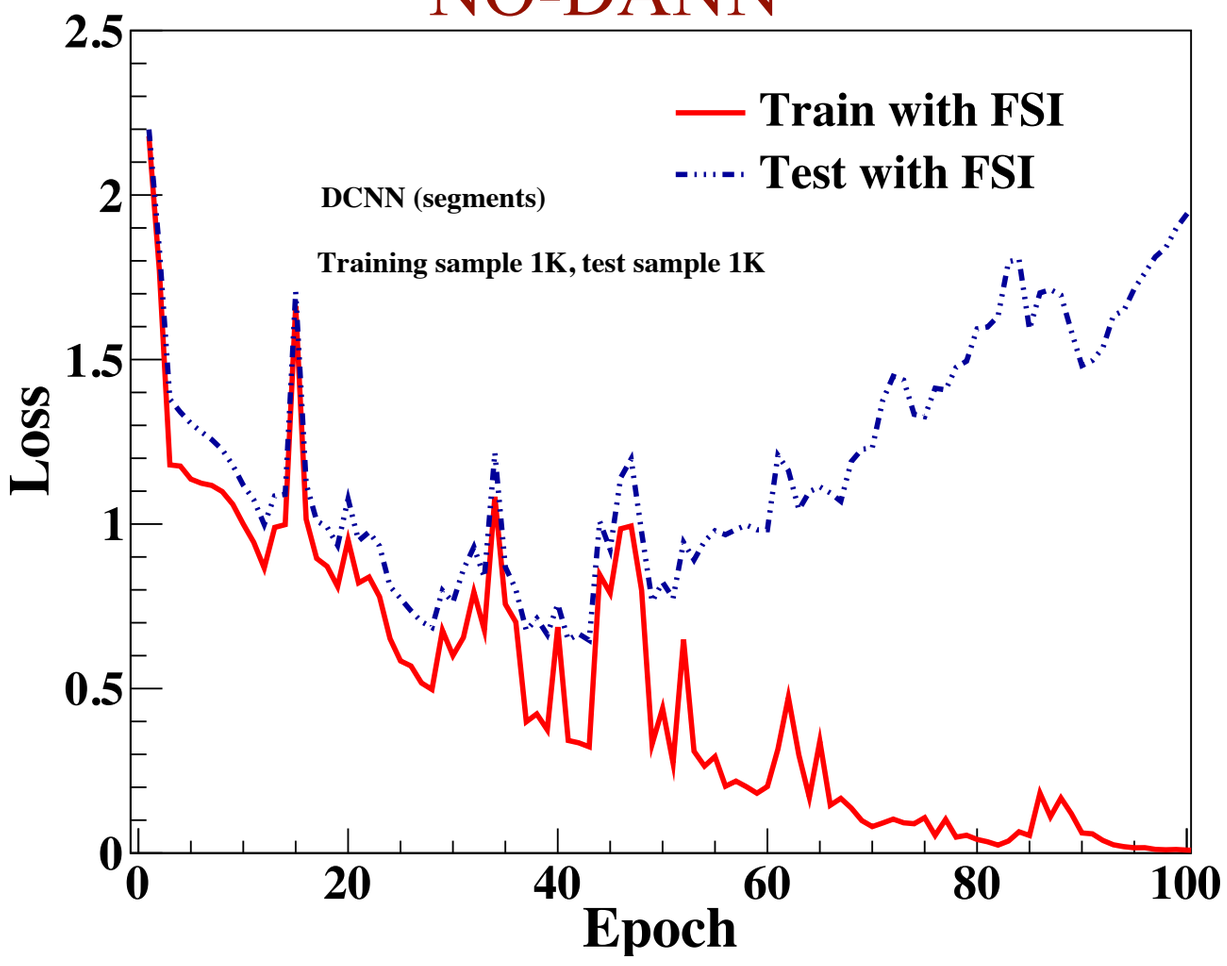
Test for overfitting

Remove the dropout layer, train with small sample

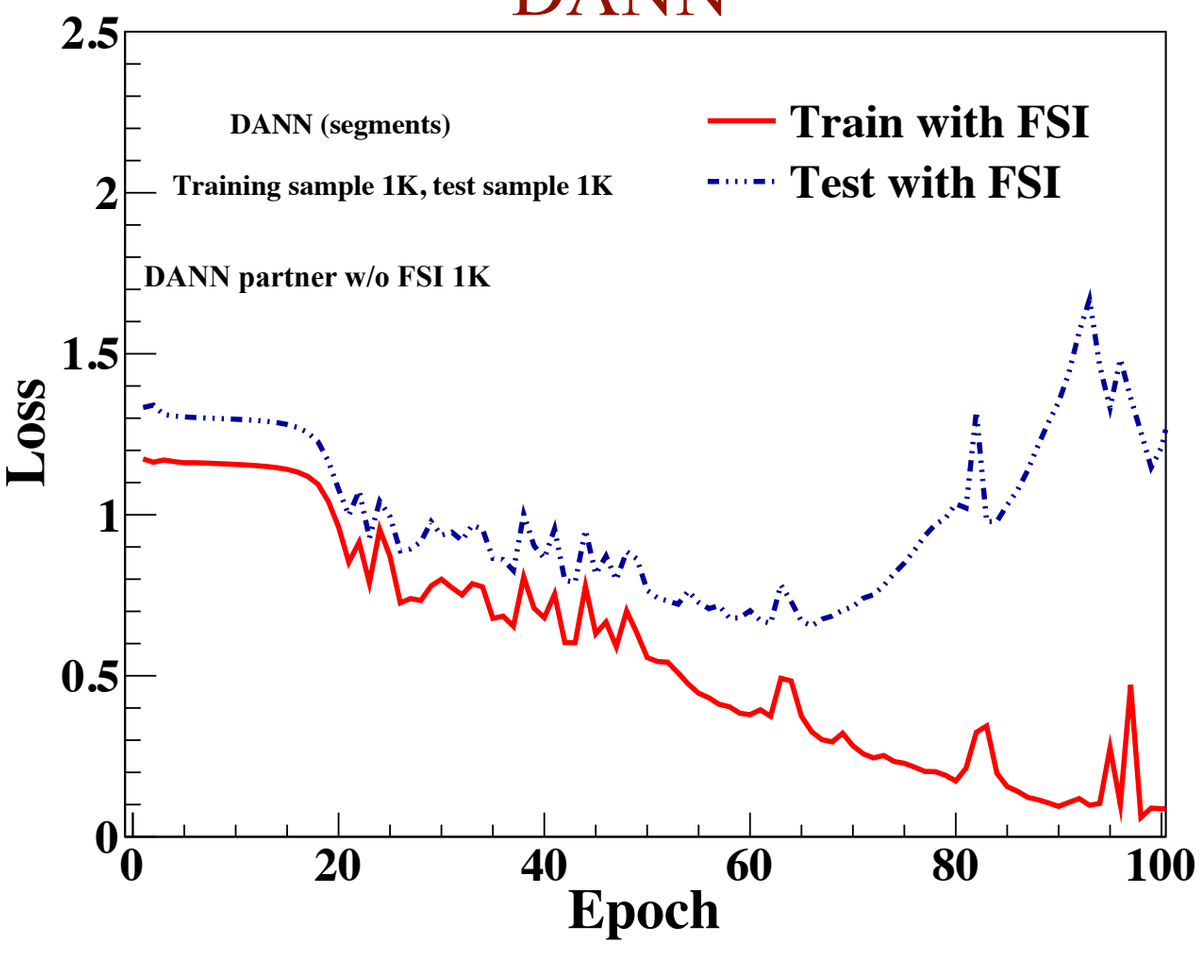
classifying events in segments

Training sample : 1k validation sample : 1k

NO-DANN



DANN



- **Epoch**: one forward pass and one backward pass of all the training examples

- For NO-DANN overfitting starts ~45 epoch. For DANN, overfitting starts ~65 epoch
  So, here DANN starts overfitting later than NO-DANN

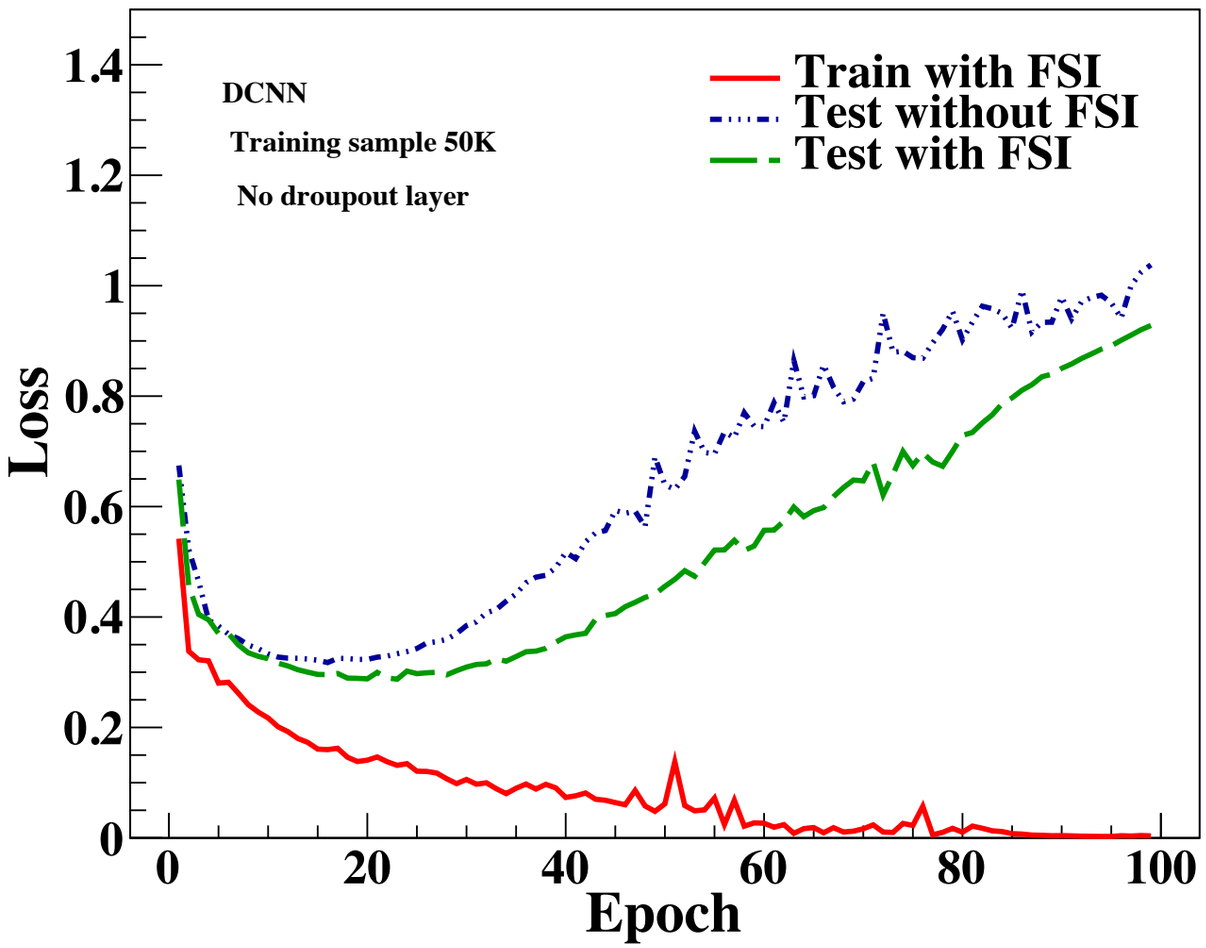
Changing the physics model (With and W/O FSI)

Test for overfitting

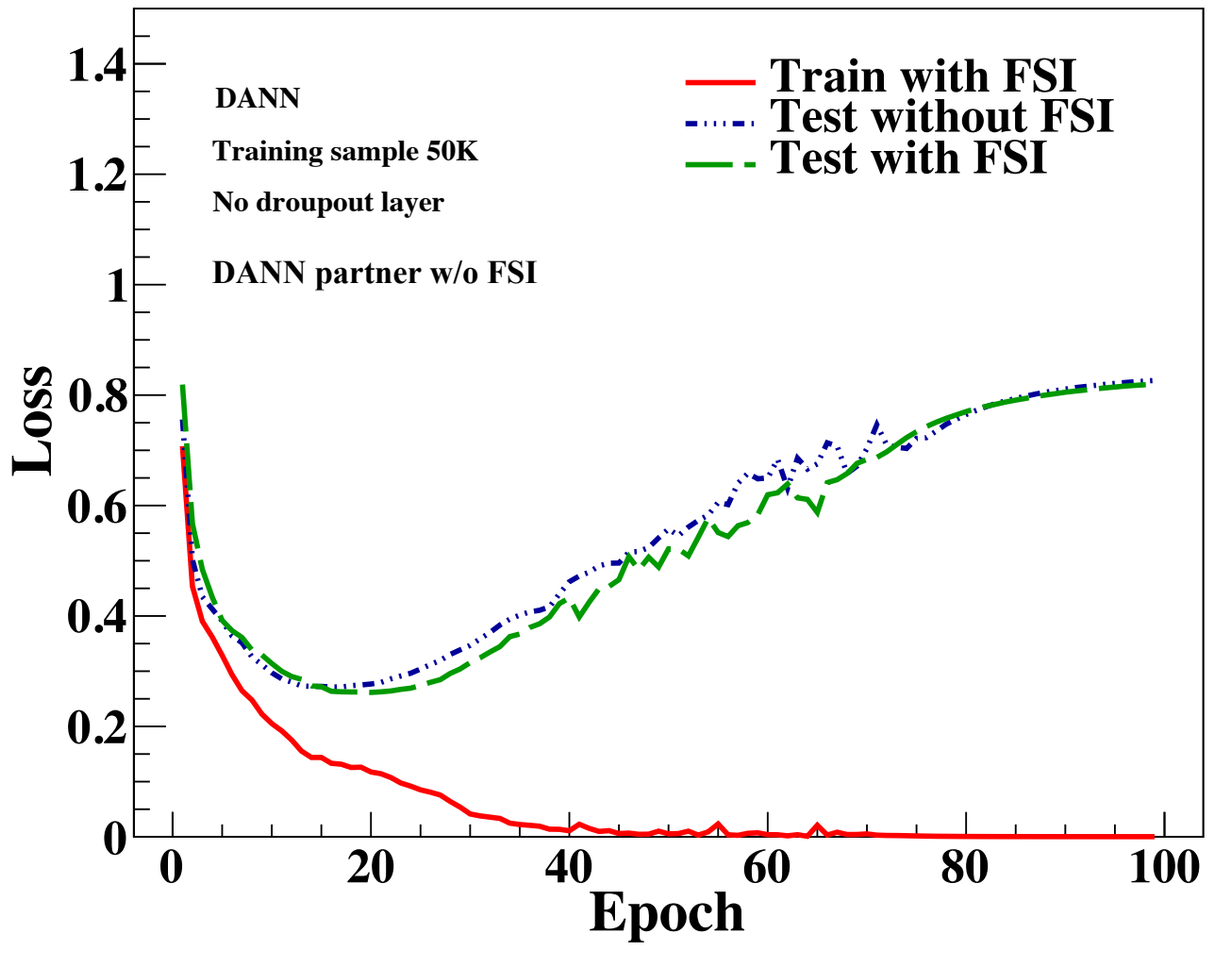
Remove the dropout layer, train with small sample

classifying events in segments

NO-DANN



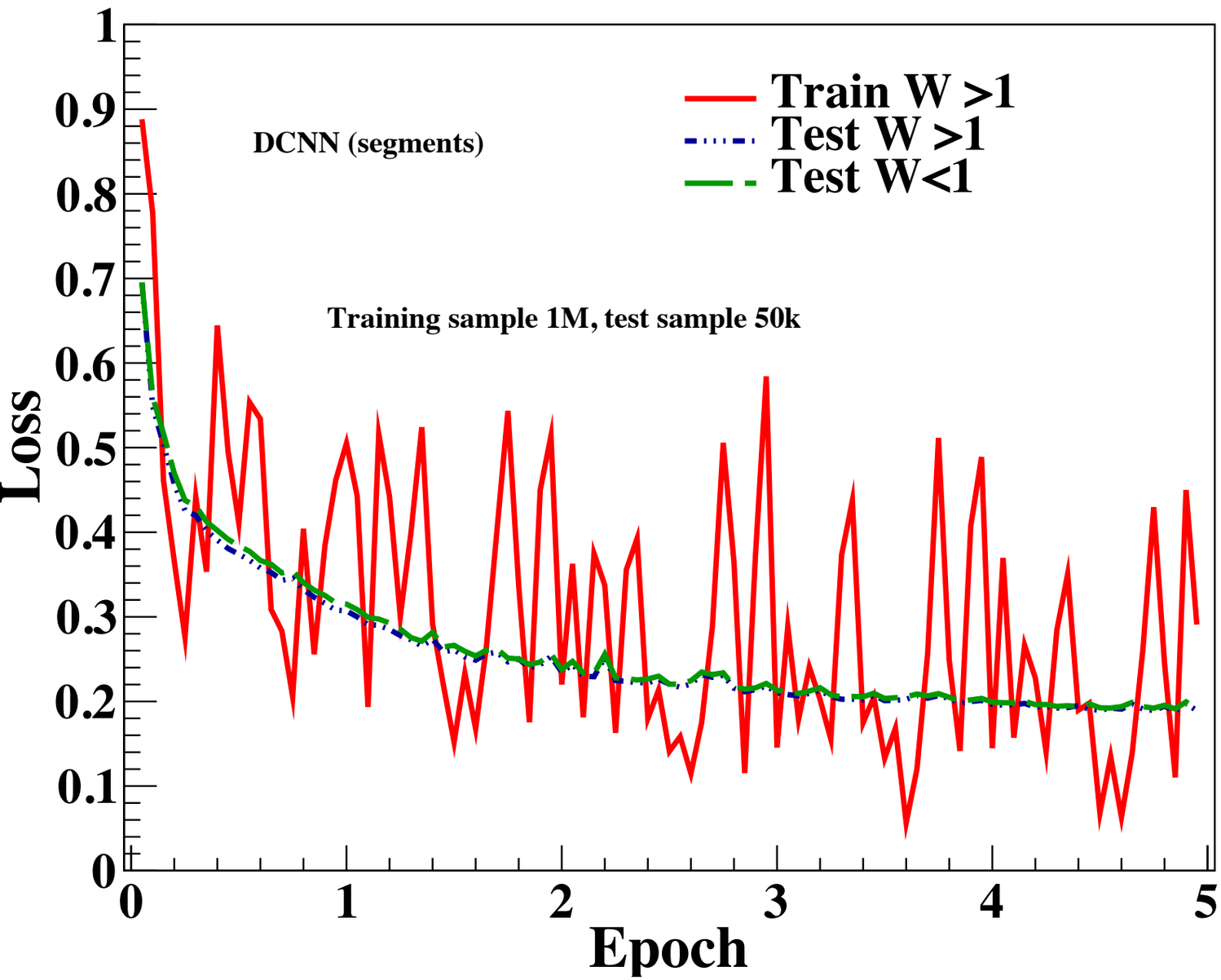
DANN



- DANN and NO-DANN both start overfitting almost at the same place , after~ 20 epoch
- For DANN, the validation-loss plot increases in slower rate than NO-DANN
- For NO-DANN, in case of validation, the loss value for NO-FSI is higher than with FSI . For DANN, both validation plots have similar behaviour(NO-FSI validation plot is slightly higher than with FSI validation plot).

Changing the parameters in hadronization model (W-split)

Train with large sample, with dropout layer



DCNN

Train with 1M sample W >1

Validation done with W >1 and W <1 sample

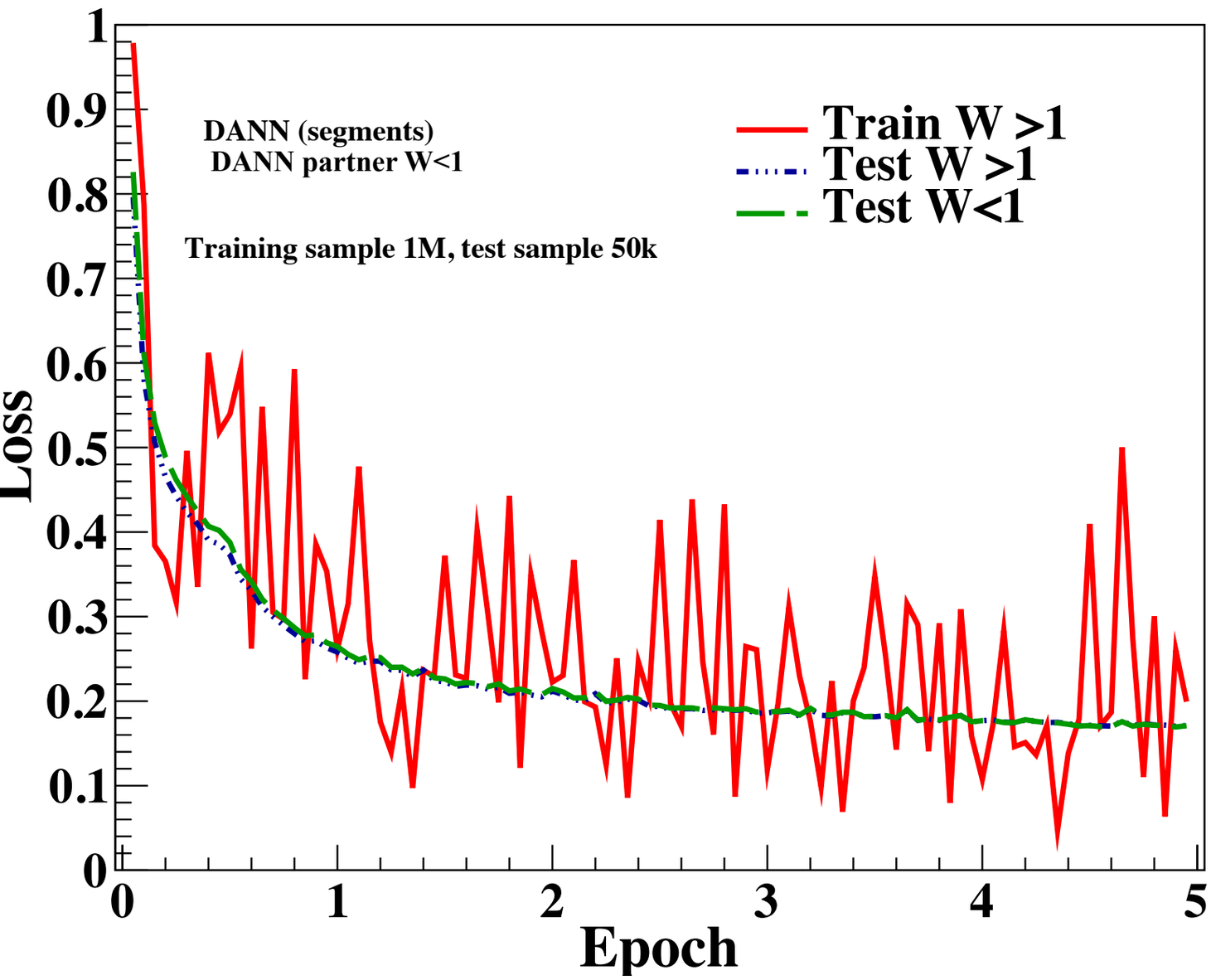
Size of validation sample 50 K

classifying events in segments

- Network is not showing any bias to the training sample once we train with large sample.

Changing the parameters in hadronization model (W-split)

Train with large sample, with dropout layer



DANN

Train with 1M sample W >1

DANN partner W <1

Validation done with W >1 and W <1 sample

Size of validation sample 50 K

classifying events in segments

- Network is not showing any bias to the training sample once we train with large sample.
  - Loss is lower for DANN compared to DCNN

Summary

- We see improvement with DNN based reconstruction over track-based reconstruction

- We are working with domain adversarial neural network to understand the bias present in the training sample (if any).

- In vertex finding, we still did not able to "break" our network. (Plausibly "vertex finding" relatively insensitive to the changing physics models)

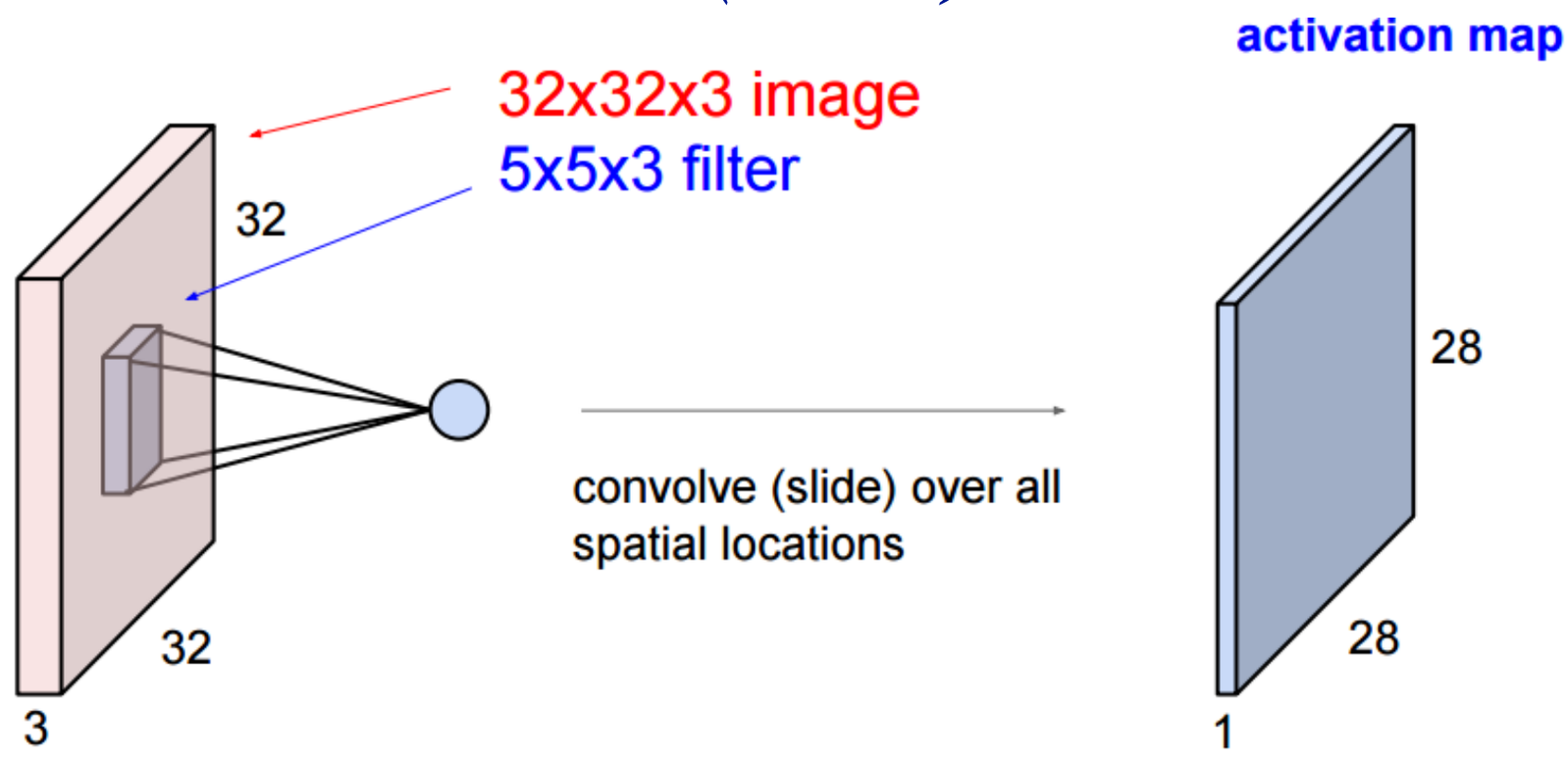
- We will test DANN with other measurements (like hadron multiplicity) which would be sensitive to changing physics model.

Thank you!

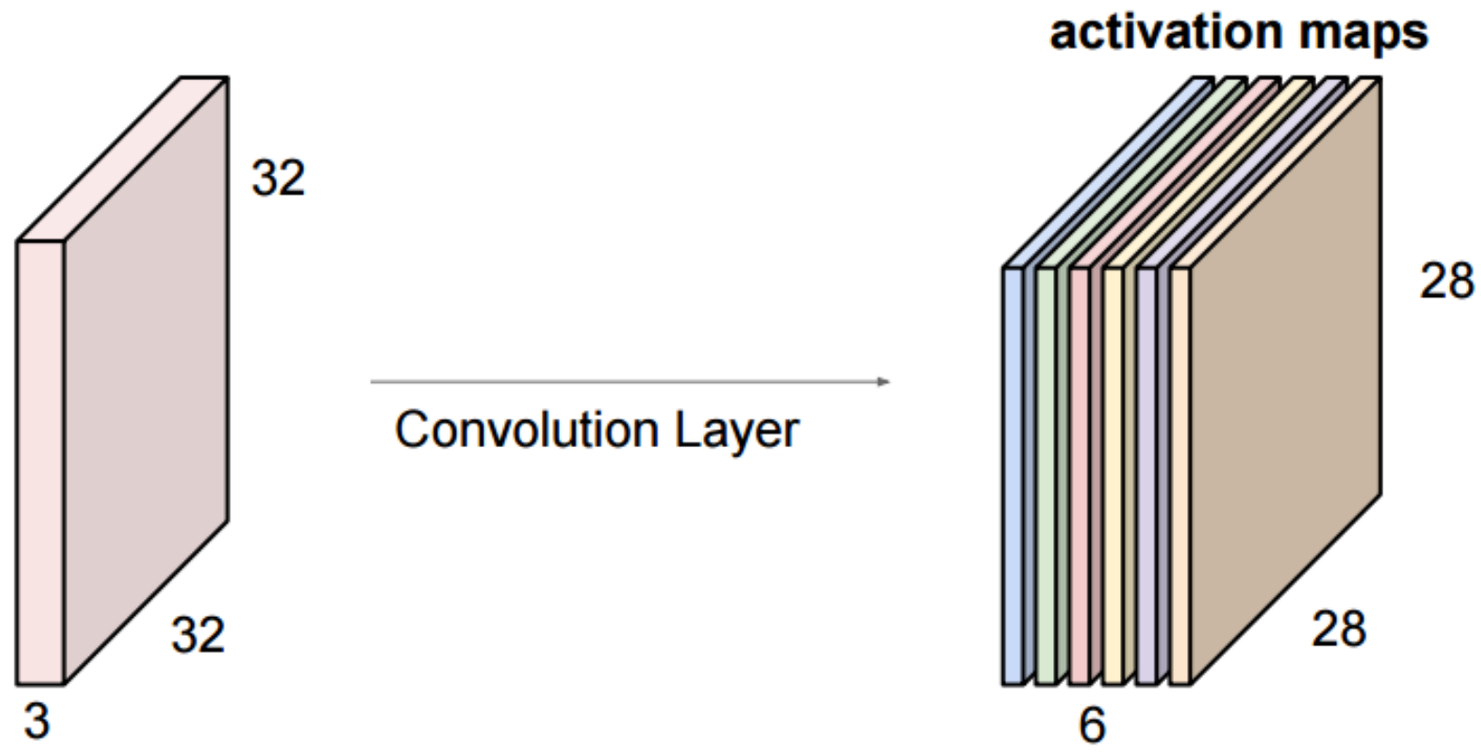
Back-up

Convolutional neural network (CNN)

The filter will look for a particular thing on all the image, this means that it will look for a pattern in the whole image with just one filter.



Convolutional layer look for six different things.
So, convolutional layer will have 6 5X5X3 filter



Isolating DIS sample

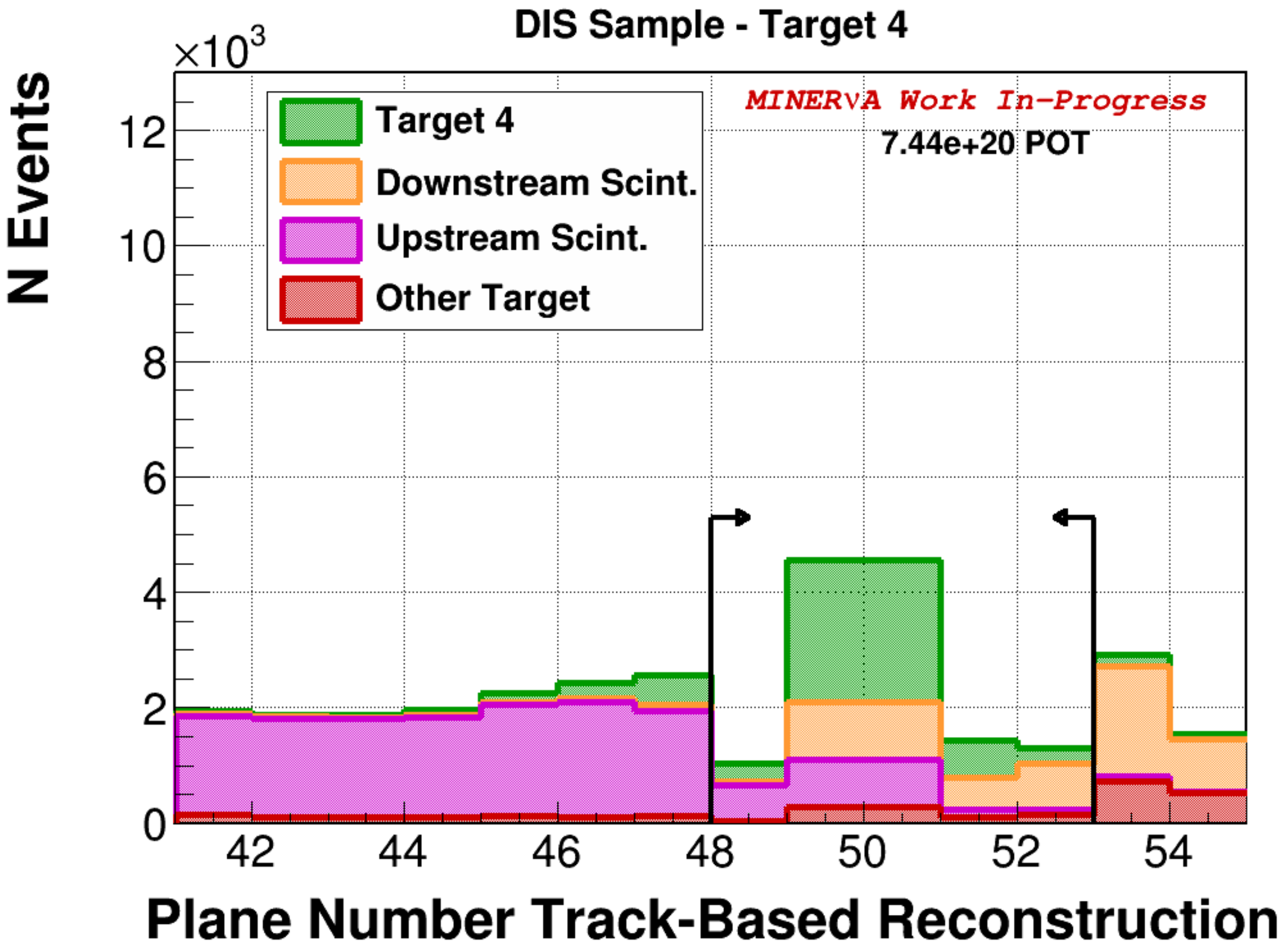
Q^2: square of the momentum transfer

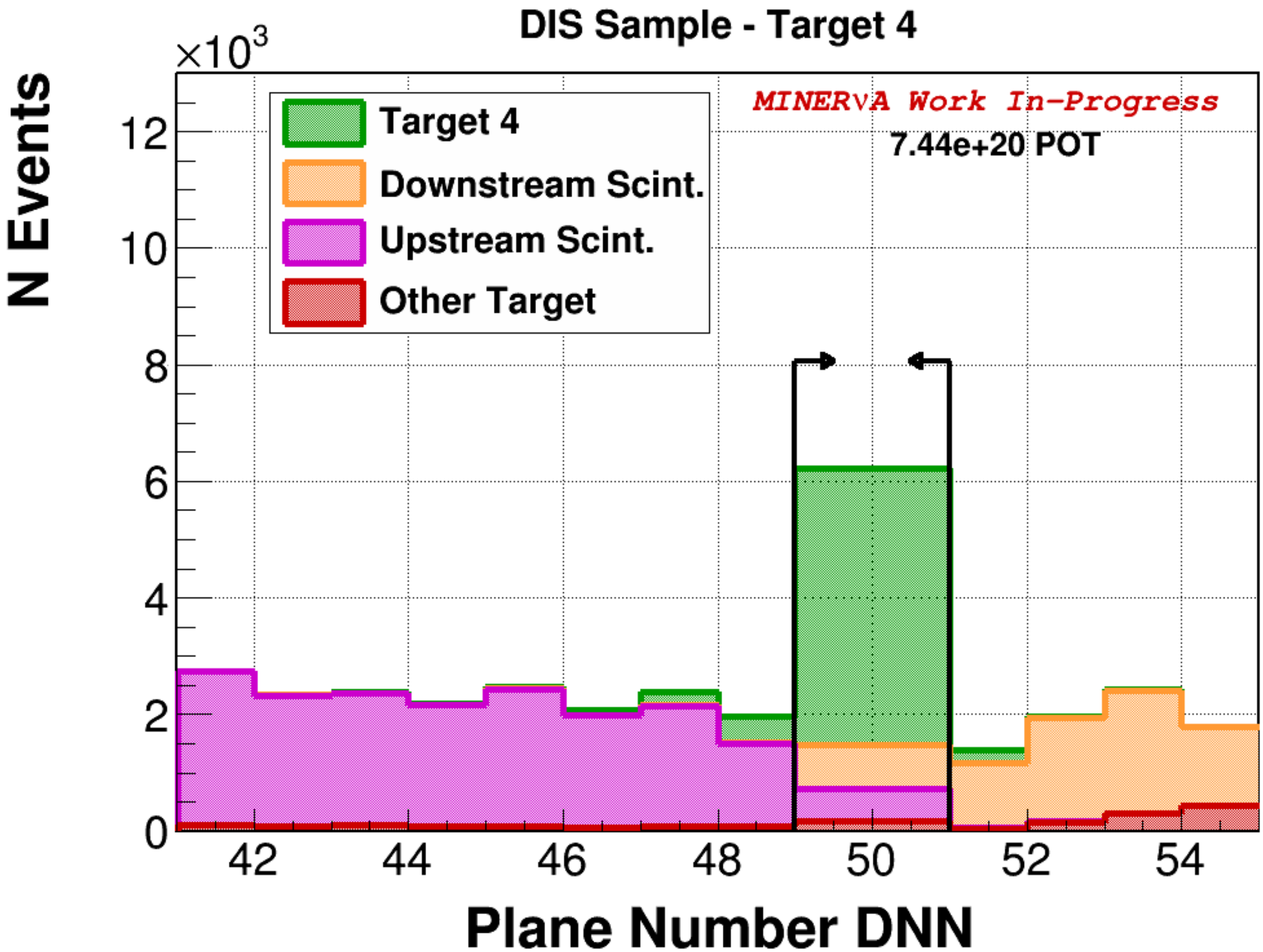
$$Q^2 = 4E_\nu E_\mu \sin^2\left(\frac{\theta_\mu}{2}\right)$$

W: invariant mass of final state hadronic system

$$W = \sqrt{m_n^2 + (2m_n^2(E_\nu - E_\mu)) - Q^2}$$

- We consider $Q^2 > 1.0$ (GeV/c)^2 to be enough momentum transfer to resolve the quark structure of the nucleons.
- W > 2.0 (GeV/c) safely *avoids* the majority of resonances, and gives us confidence the hadronic shower is from deep inelastic scattering off of a parton.

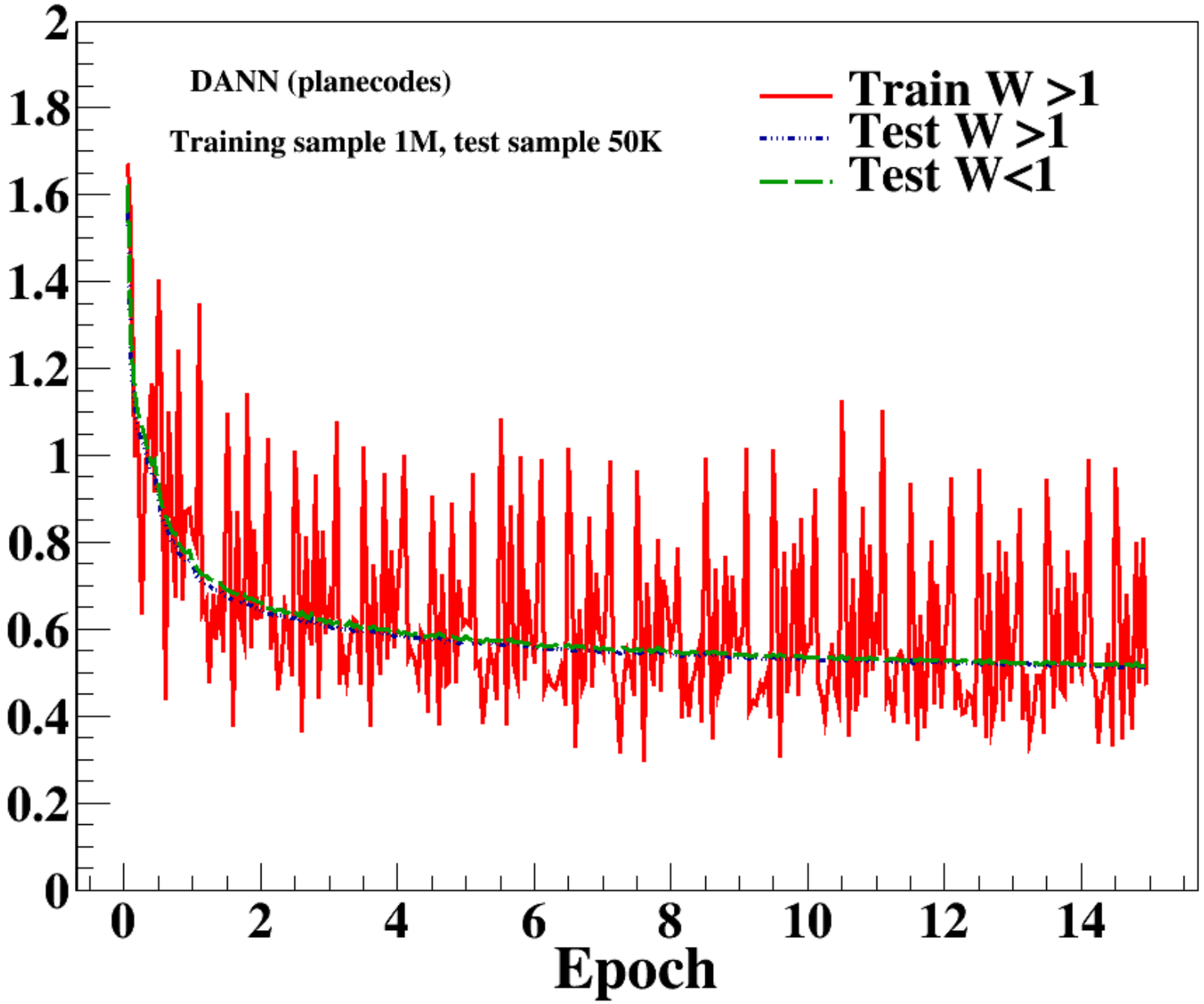

DIS Sample - Target 4
MINERvA Work In-Progress
7.44e+20 POT
Target 4
Downstream Scint.
Upstream Scint.
Other Target
N Events
×10^3
0
2
4
6
8
10
12
42
44
46
48
50
52
54
Plane Number Track-Based Reconstruction


DIS Sample - Target 4
MINERvA Work In-Progress
7.44e+20 POT
Target 4
Downstream Scint.
Upstream Scint.
Other Target
N Events
×10^3
0
2
4
6
8
10
12
42
44
46
48
50
52
54
Plane Number DNN

Changing the parameters in hadronization model (W-split)

Train with large sample, with dropout layer

classifying events in plane number



DANN

Train with W >1 sample
training sample size : 1M

Train with W >1 sample
training sample size : 1M

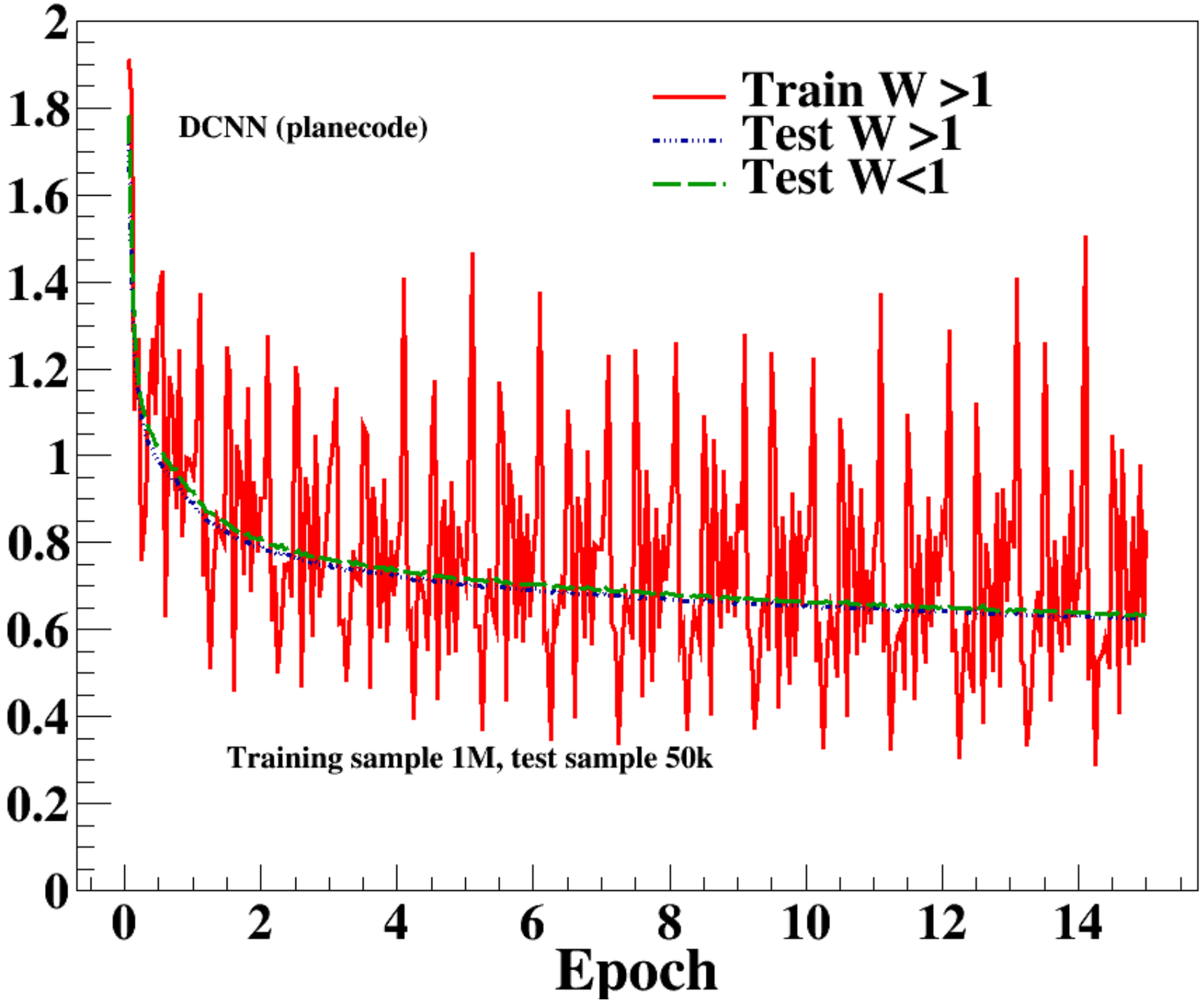
Train with W >1 sample
training sample size : 1M

The difference between green and blue line is little bit less than compare to NO-DANN

Changing the parameters in hadronization model (W-split)

Train with large sample, with dropout layer

classifying events in plane number



DCNN

Train with W >1 sample
training sample size : 1M

Train with W >1 sample
training sample size : 1M

green line:Test with W <1
test sample size 50 K