Fermilab U.S. DEPARTMENT OF ENERGY Office of Science

The HEP.TrkX Project: Deep Learning for Particle Tracking

Aristeidis Tsaris, for the HEP.TrkX Collaboration

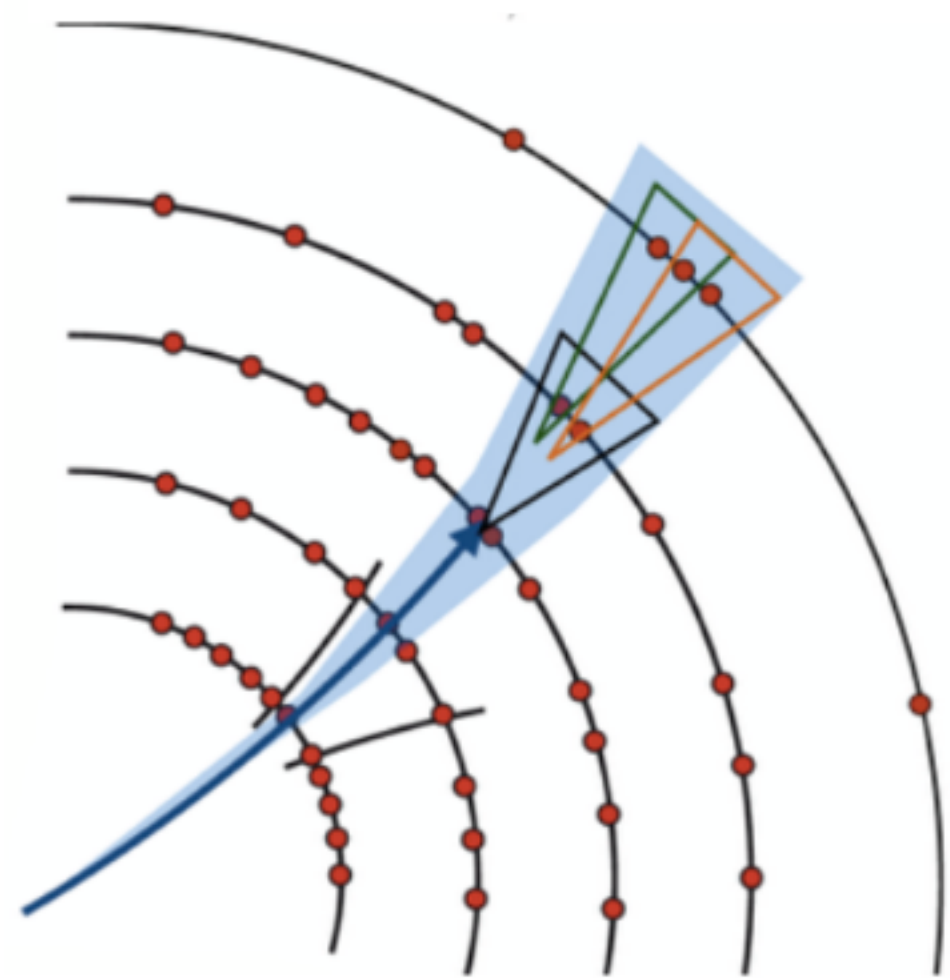
ACAT 2017

21-25 August 2017

University of Washington, Seattle

Tracking at the LHC

- LHC particle tracking algorithms have seen great success in Runs I and II. In a nutshell:
  - Track seeding: using combinatorial search
  - Track building: using combinatorial Kalman Filter (is the most time consuming part)
  - Track fitting: final parameter estimation



Credit: Andy Salzburger

Tracking at the HL-LHC

- The High-Luminosity LHC will increase the number of charged particles and the detector occupancy
  - Will increase the collisions per crossing up to 200
  - Traditional tracking algorithms scale at least quadratically with increasing detector occupancy
  - Tracking algorithms will need to run faster and in parallel

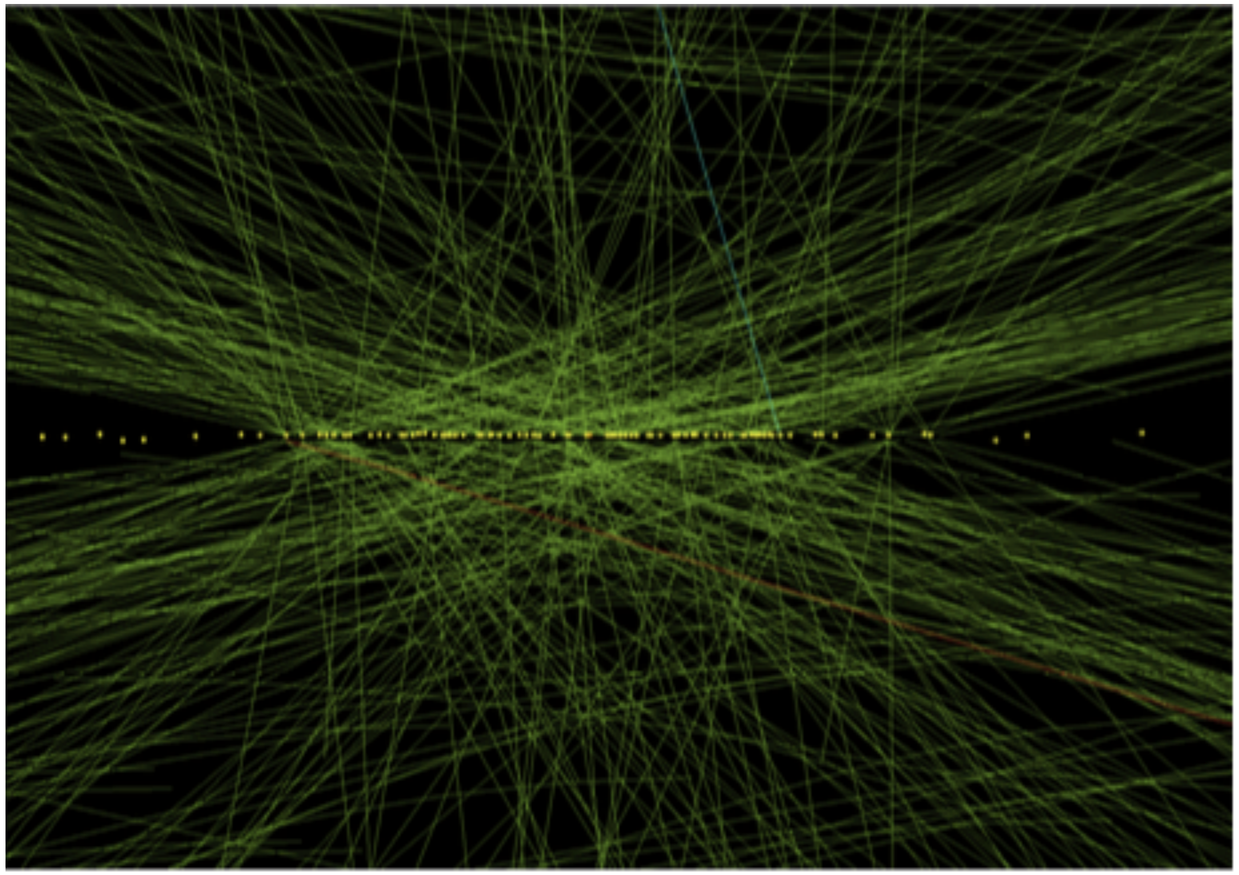
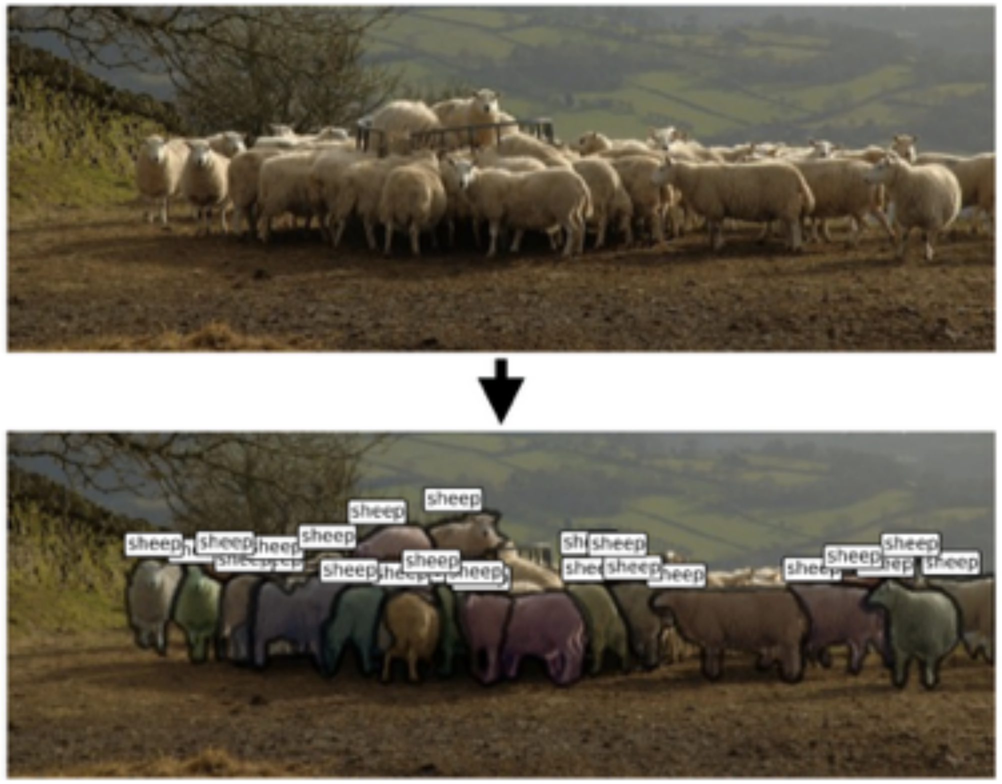


Image: CMS

Fermilab

Some Deep Learning Inspirations

Image Segmentation using R-CNN



Zagoruyko et al, https://arxiv.org/pdf/1604.02135.pdf

Our goal (more or less…):

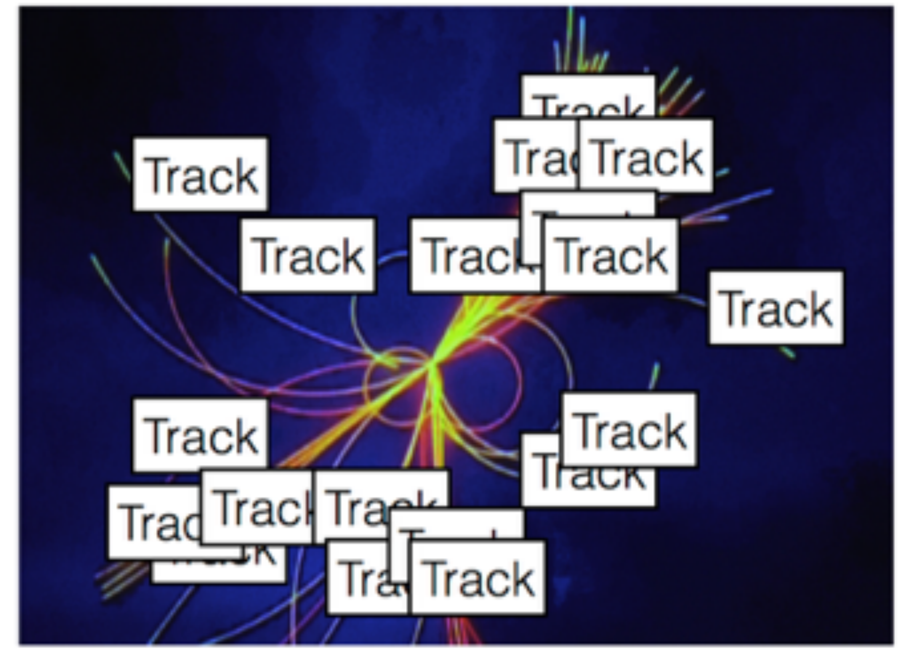
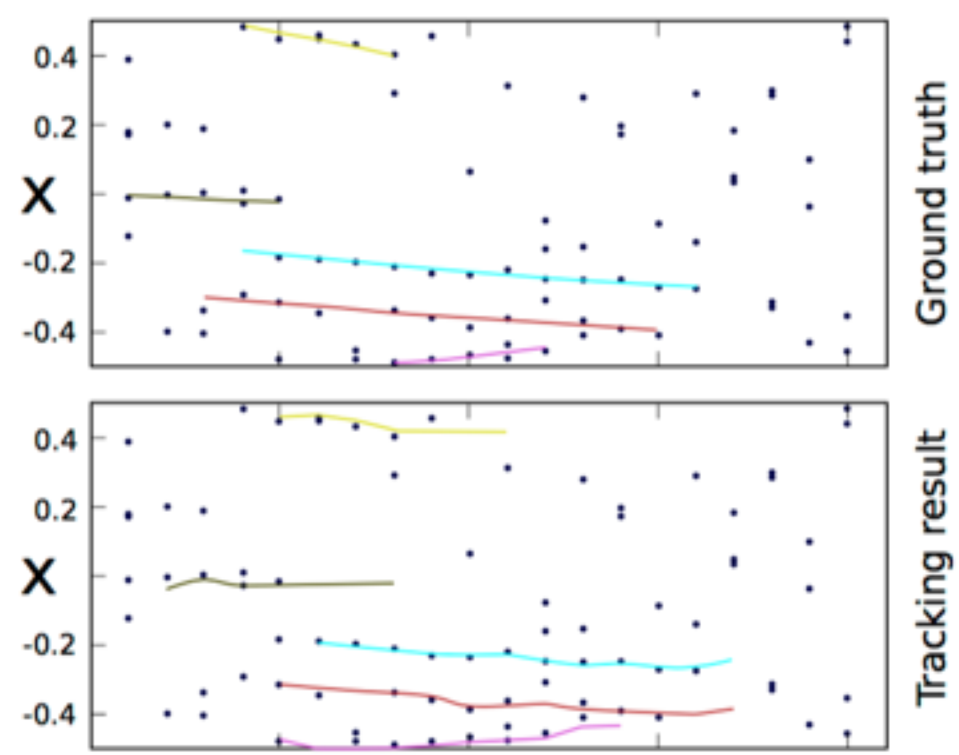


Photo by Pier Marco Tacca/Getty Images

Online object tracking using RNN

Anton Milan et al, https://arxiv.org/pdf/1604.03635.pdf



The HEP.TrkX Project

- Pilot project funded by DOE ASCR and COMP HEP
- Part of HEP CCE
- Our collaboration:
  - **LBL:** Steve Farrell, Mayur Mudigonda, Prabhat, Paolo Calafiura, Julien Esseiva
  - **Caltech:** Dustin Anderson, Jean-Roch Vlimant, Josh Bendavid, Maria Spiropoulou, Stephan Zheng
  - **FNAL:** Me, Giuseppe Cerati, Jim Kowalkowski, Lindsey Gray, Panagiotis Spentzouris, Daniel Zurawski, Keshav Kapoor
- Goals:
  - Explore and develop new tracking algorithms based on modern ML techniques
  - Demonstrate a scalable algorithm with the potential to reconstruct tracks in the HL-LHC conditions

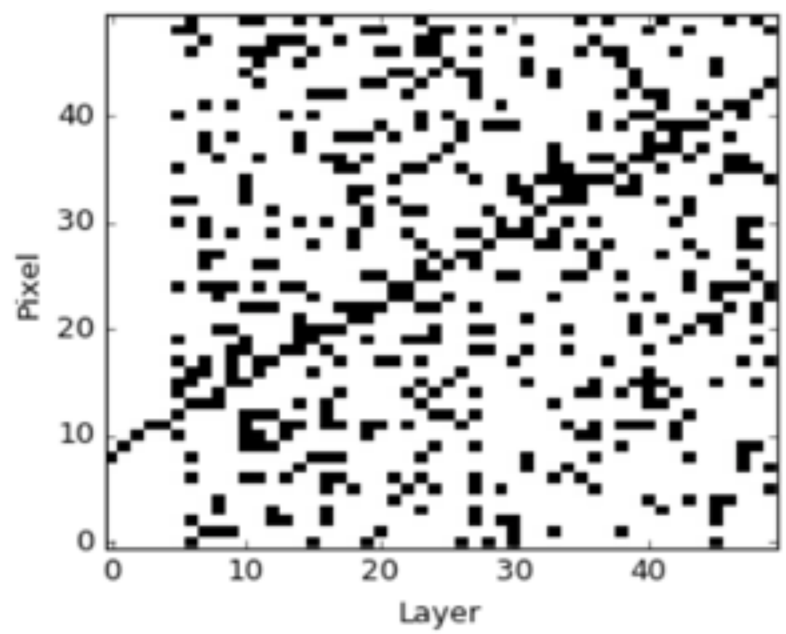
Possible Applications in Tracking

- **Seed finding:** improve the scaling of current algorithms
- **Hit Clustering and building tracks:** replace Kalman Filter with a better/faster iterative algorithm
- **Track fitting:** use RNN for track parameter estimation
- **End to end method:** cluster hits directly into tracks or produce values for track parameters

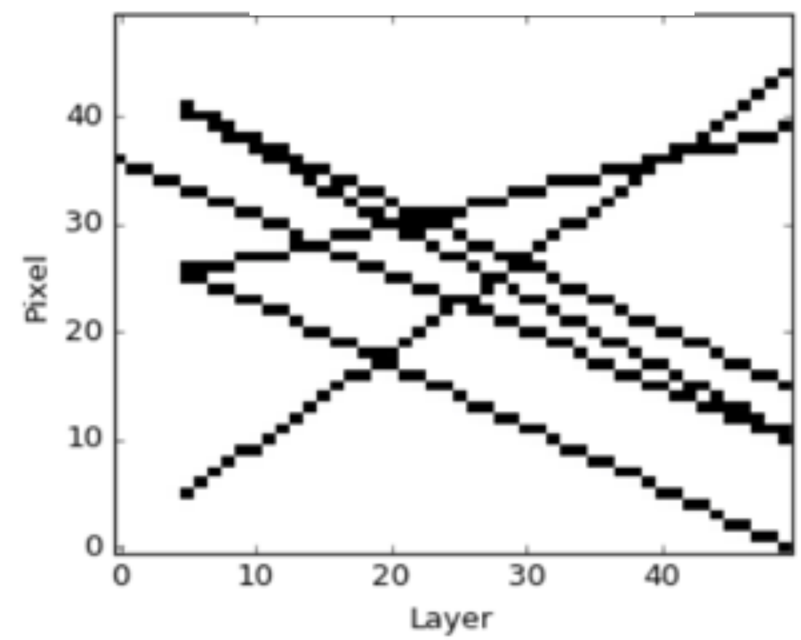
Toy Datasets

- Most of the development so far has been done in toy datasets
- Straight line tracks
- No missing hits
- Random background tracks and or uniform noise

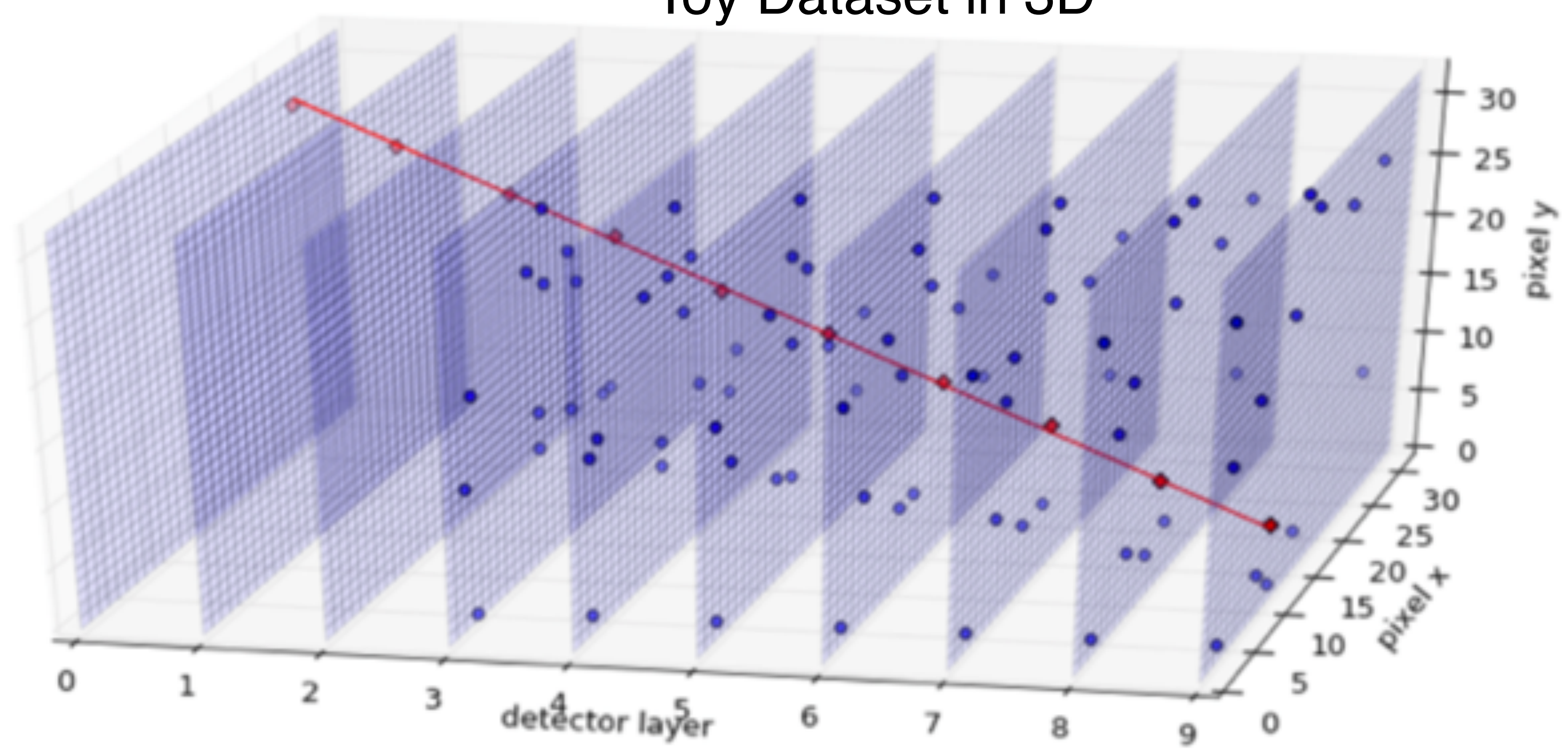
Single track with noise 2D



Multi-track in 2D

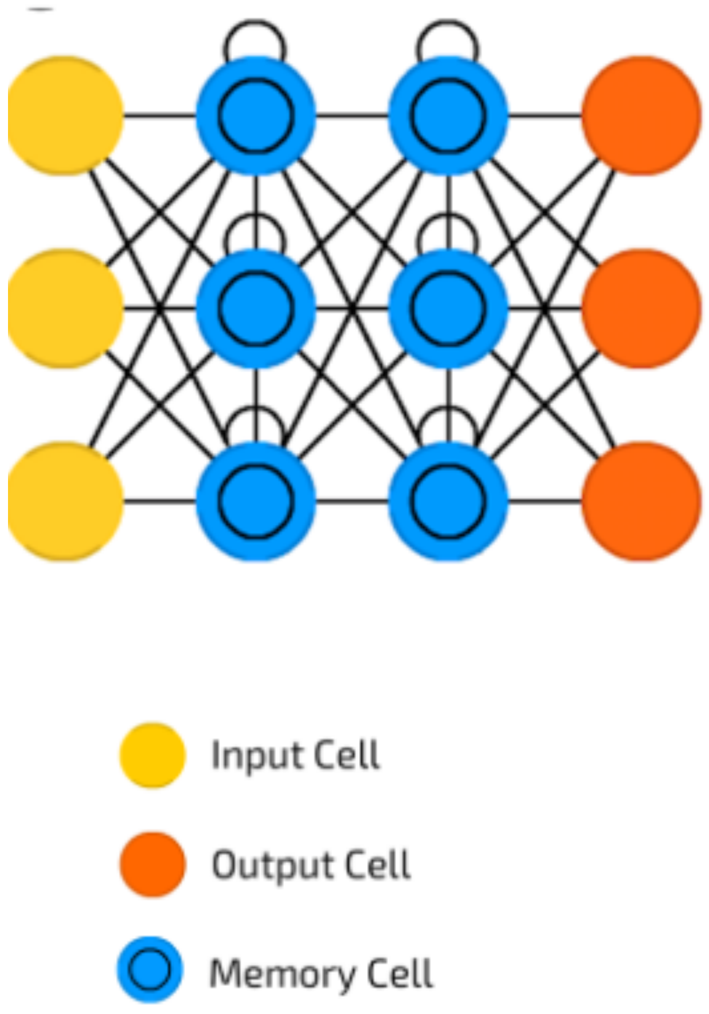


Toy Dataset in 3D



Long-Short-Term Memory (LSTM)

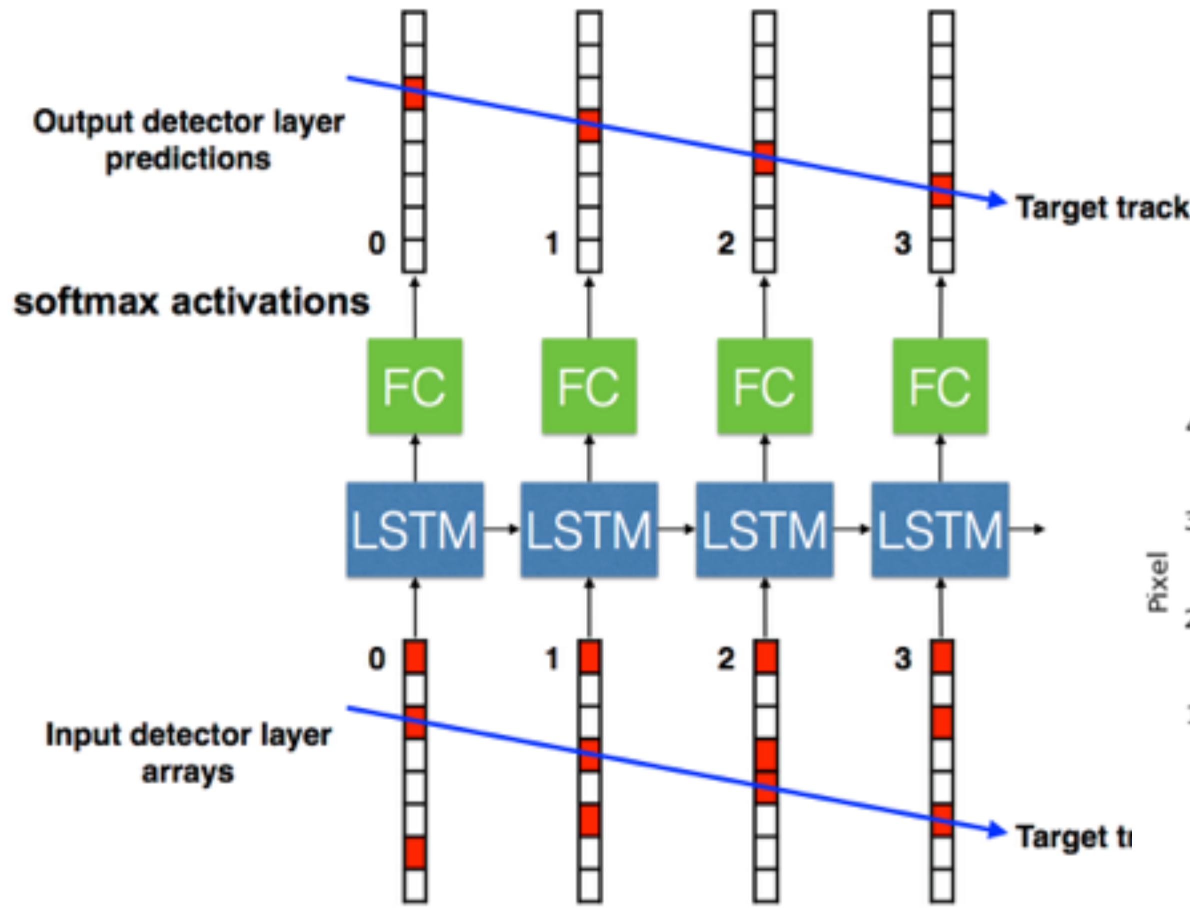
- LSTM are recurrent neural networks that model long term dependencies in sequence data by carrying memory
  - Produce a sequence of outputs
  - Could be a better alternative to the combinatorial scaling problem in KF algorithms
  - Find multi tracks at once



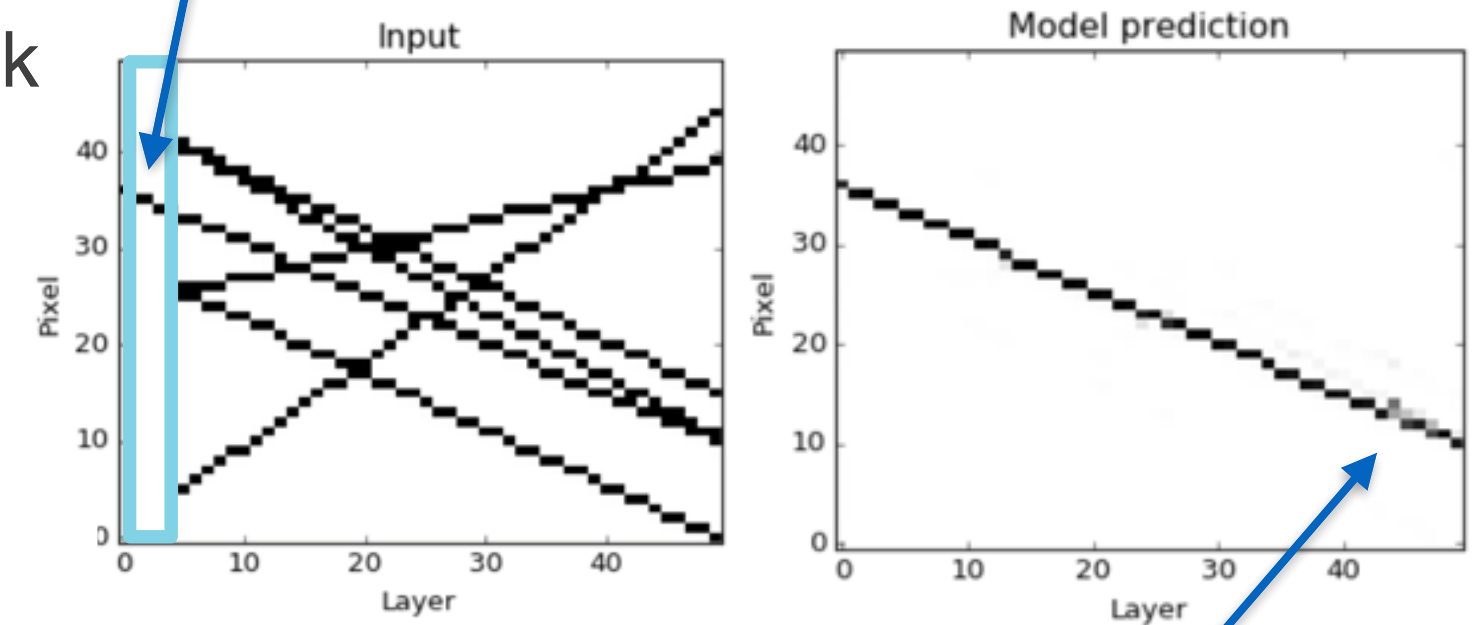
http://www.asimovinstitute.org/neural-network-zoo/

Hit Classification with LSTM in 2D

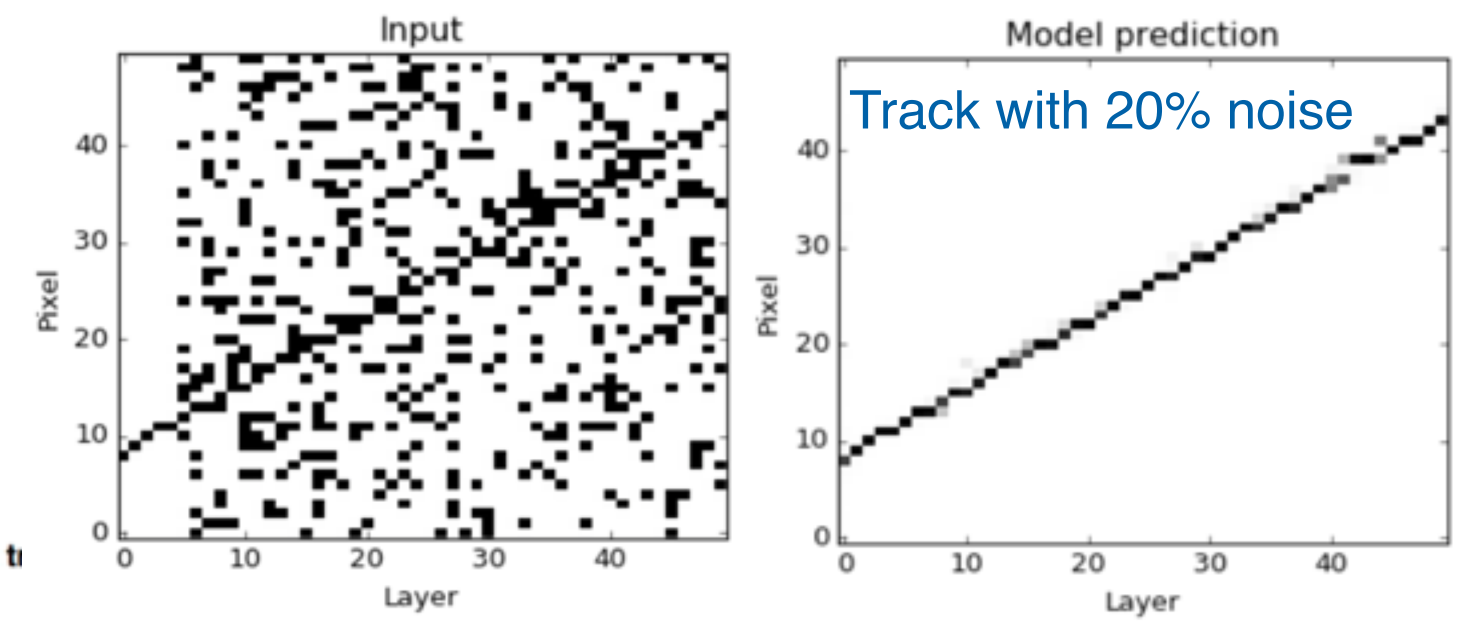
- At each step it considers a slice of the detector and outputs a probabilistic estimate of the track hit location in the current slice
- The LSTM memory state propagates relevant information from layer to layer



Try to reconstruct this seeded track



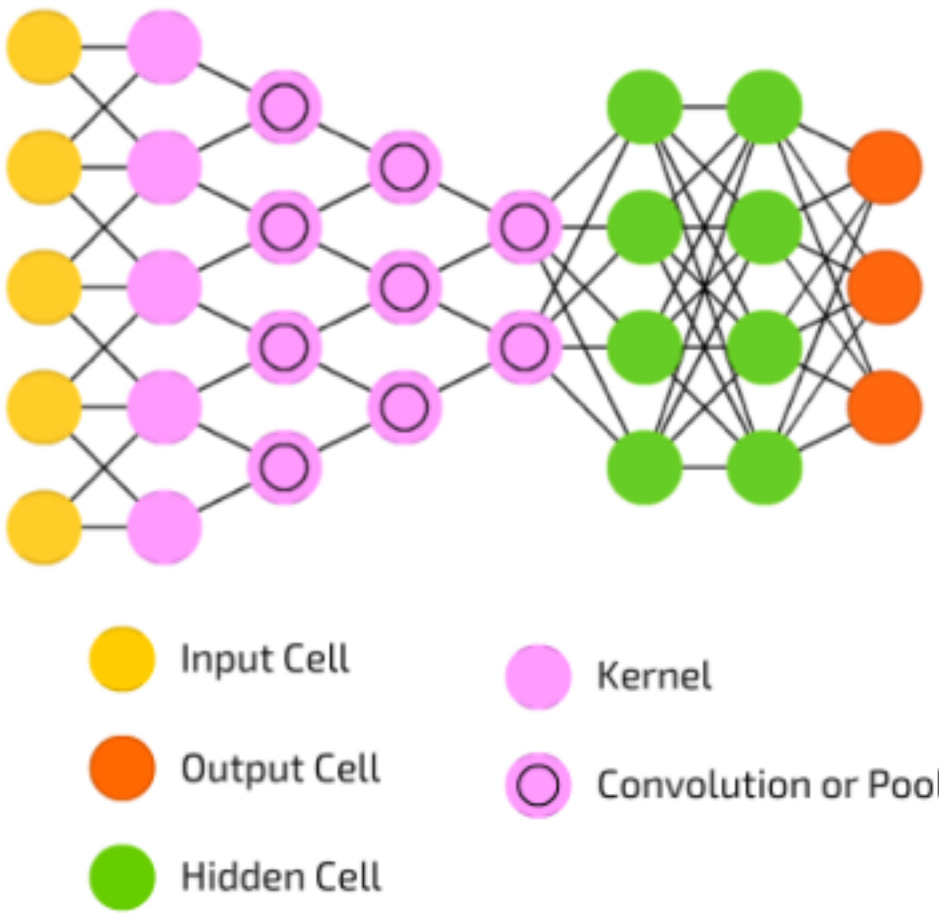
Uncertainty is larger near track intersection points



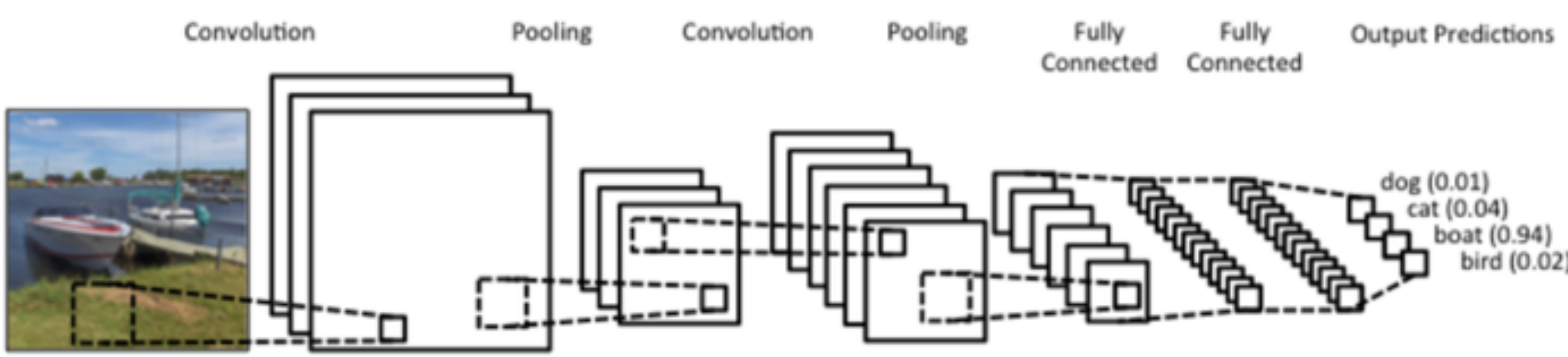
Fermilab

Convolutional Neural Networks (CNN)

- CNNs have great success in image classification, can be used as track finders:
  - Treat track finding as an image recognition problem
  - Early layers look for track stubs
  - Later layers connect stubs together to build tracks
  - Learn abstract features of the data that can be used to extract track parameters or classify hits



http://www.asimovinstitute.org/neural-network-zoo/

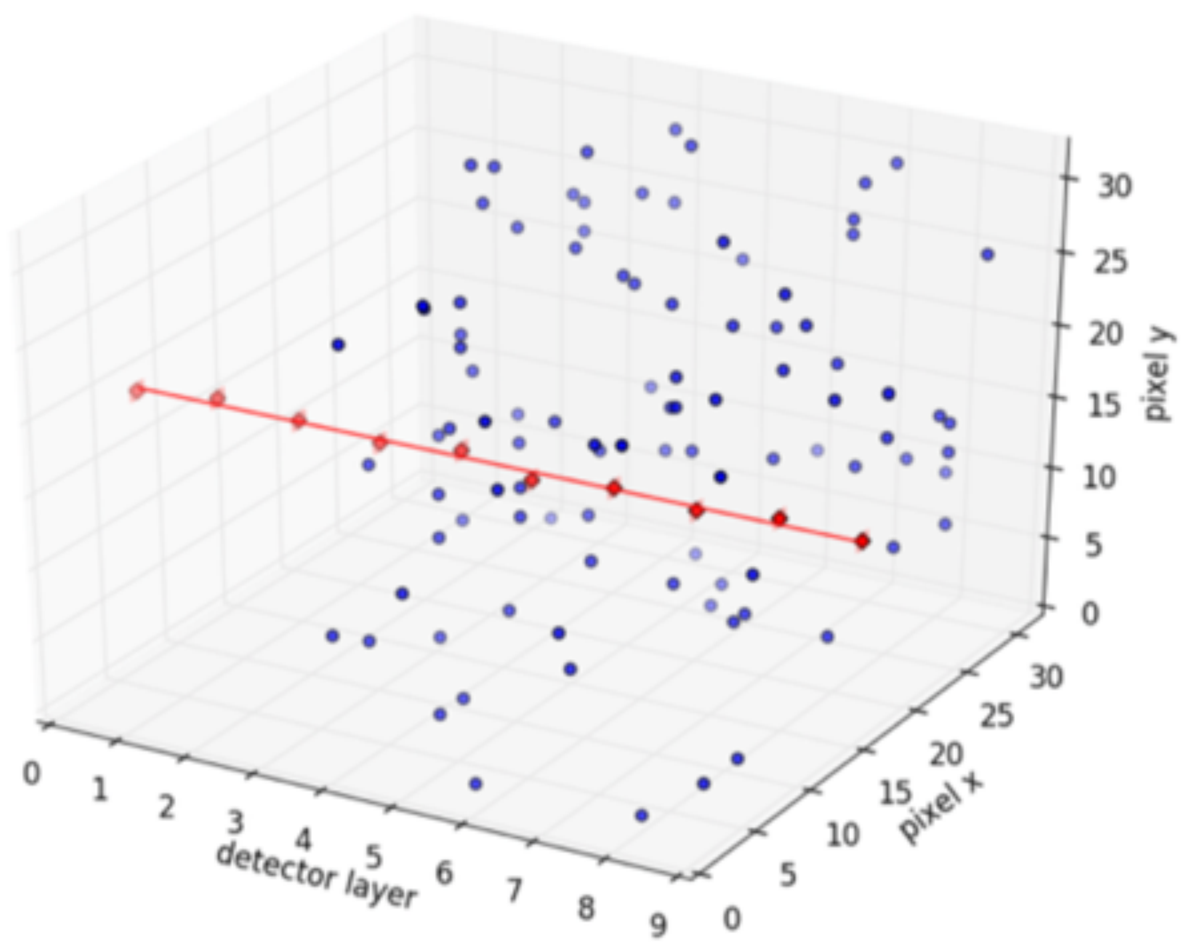


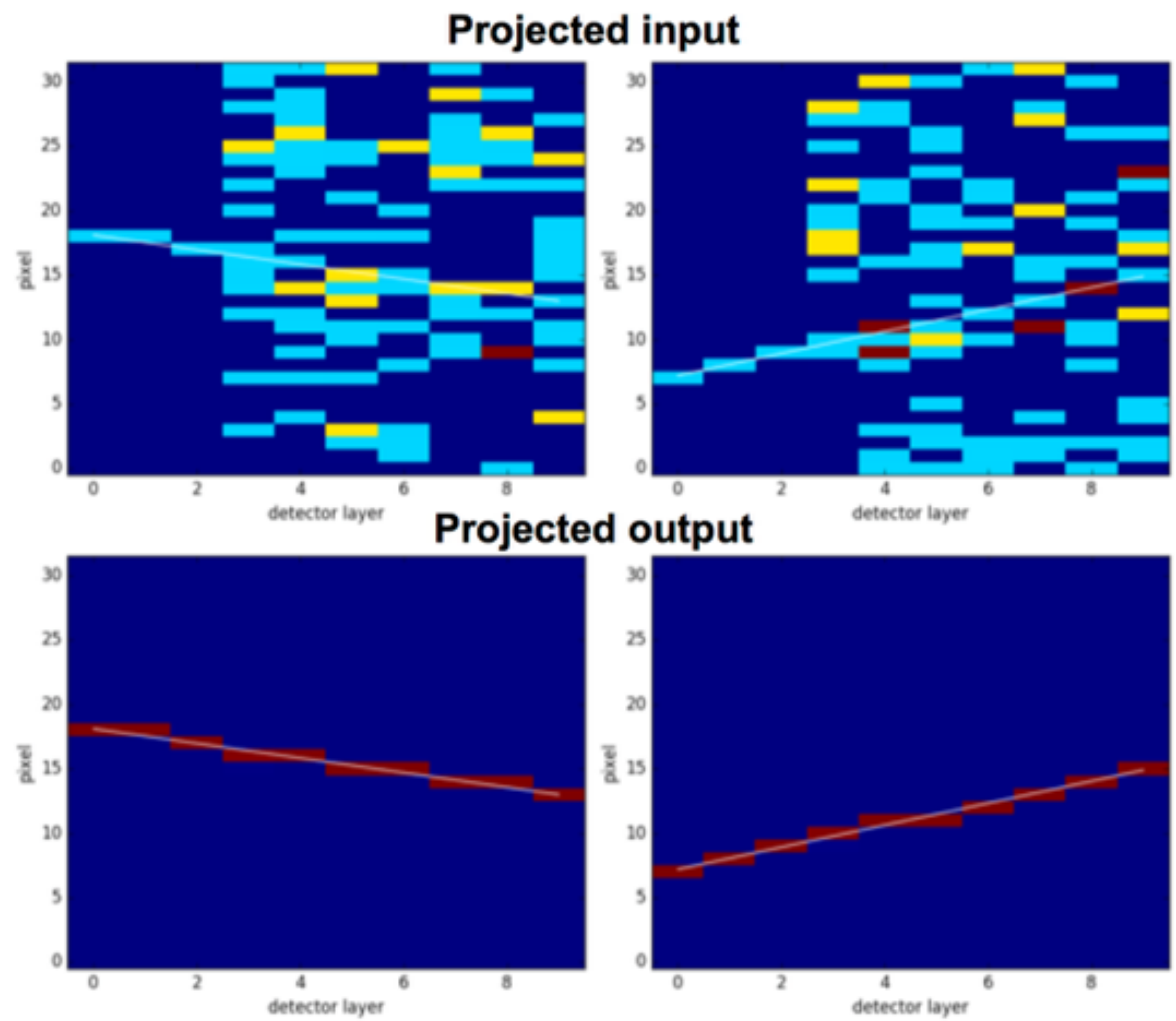
http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

Hit Classification with CNN in 3D

- Basic CNN model with 10 layers and 3x3x3 filters

3 avg bkg tracks, 1% noise

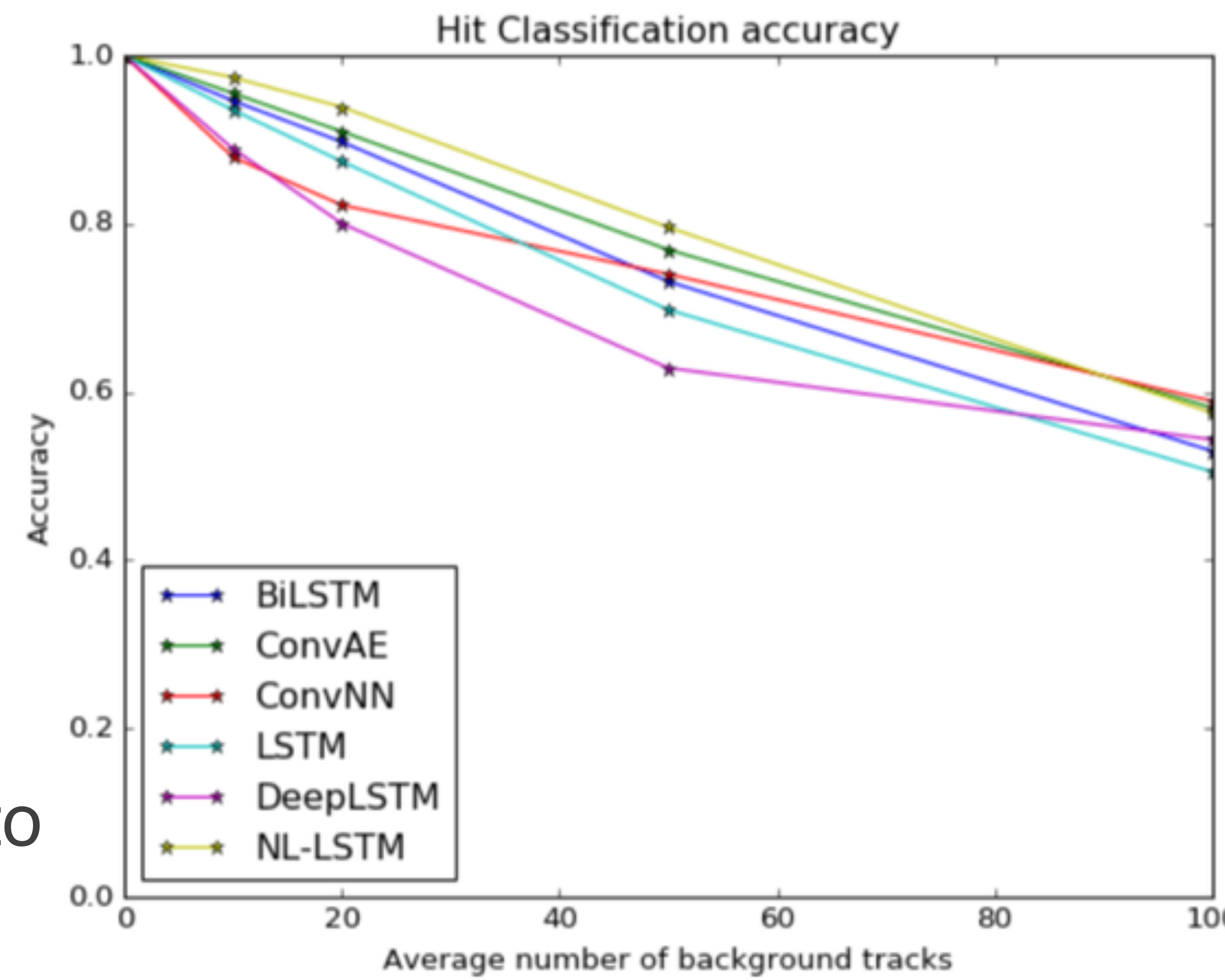




Fermilab

Architectures Comparison for Hit Classification in 3D

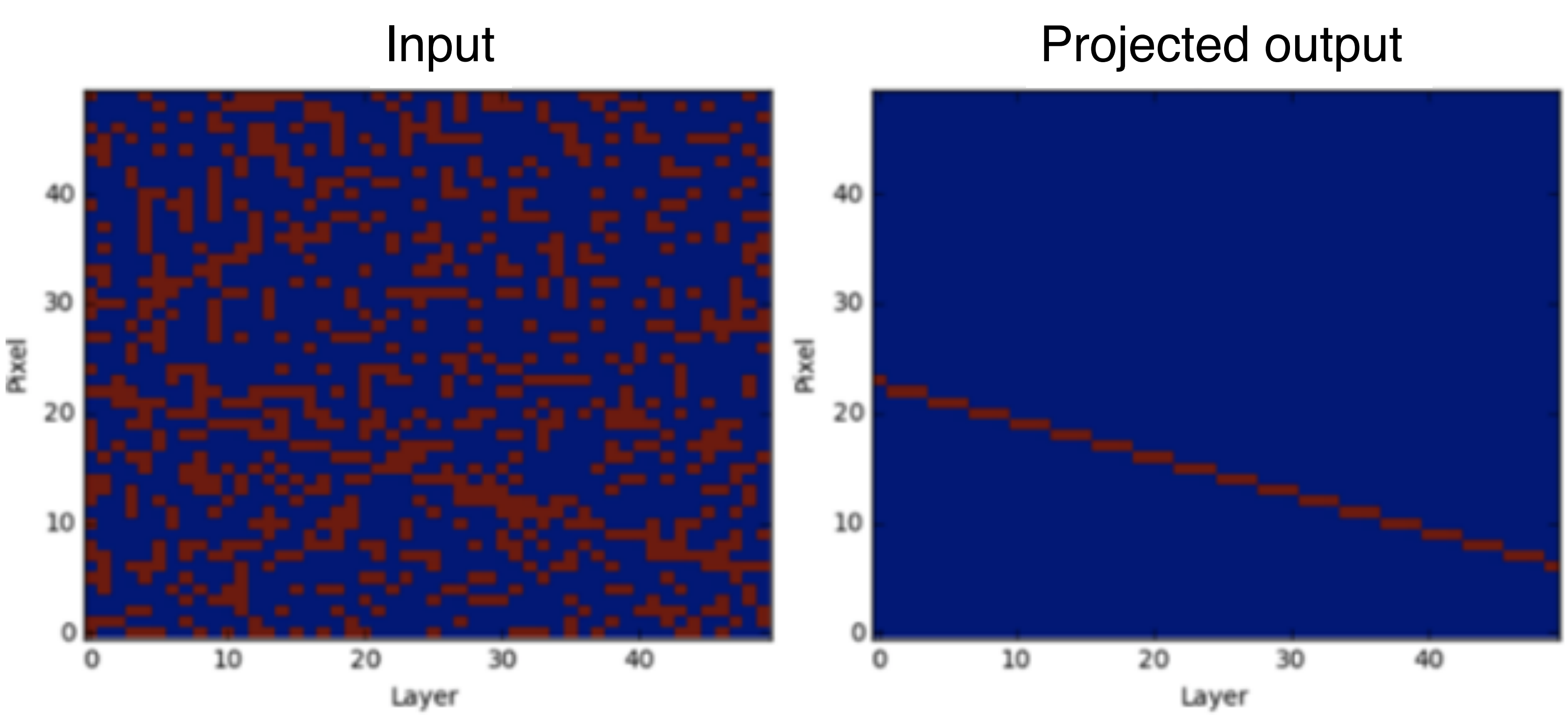
- **Deep LSTM:** more fully connected layers
- **Bi-directional LSTM:** run forward and backward simultaneously
- **Next-layer LSTM:** prediction of the hit in the next detector layer
- **Convolutional autoencoder:** alternative to LSTM for layer by layer prediction

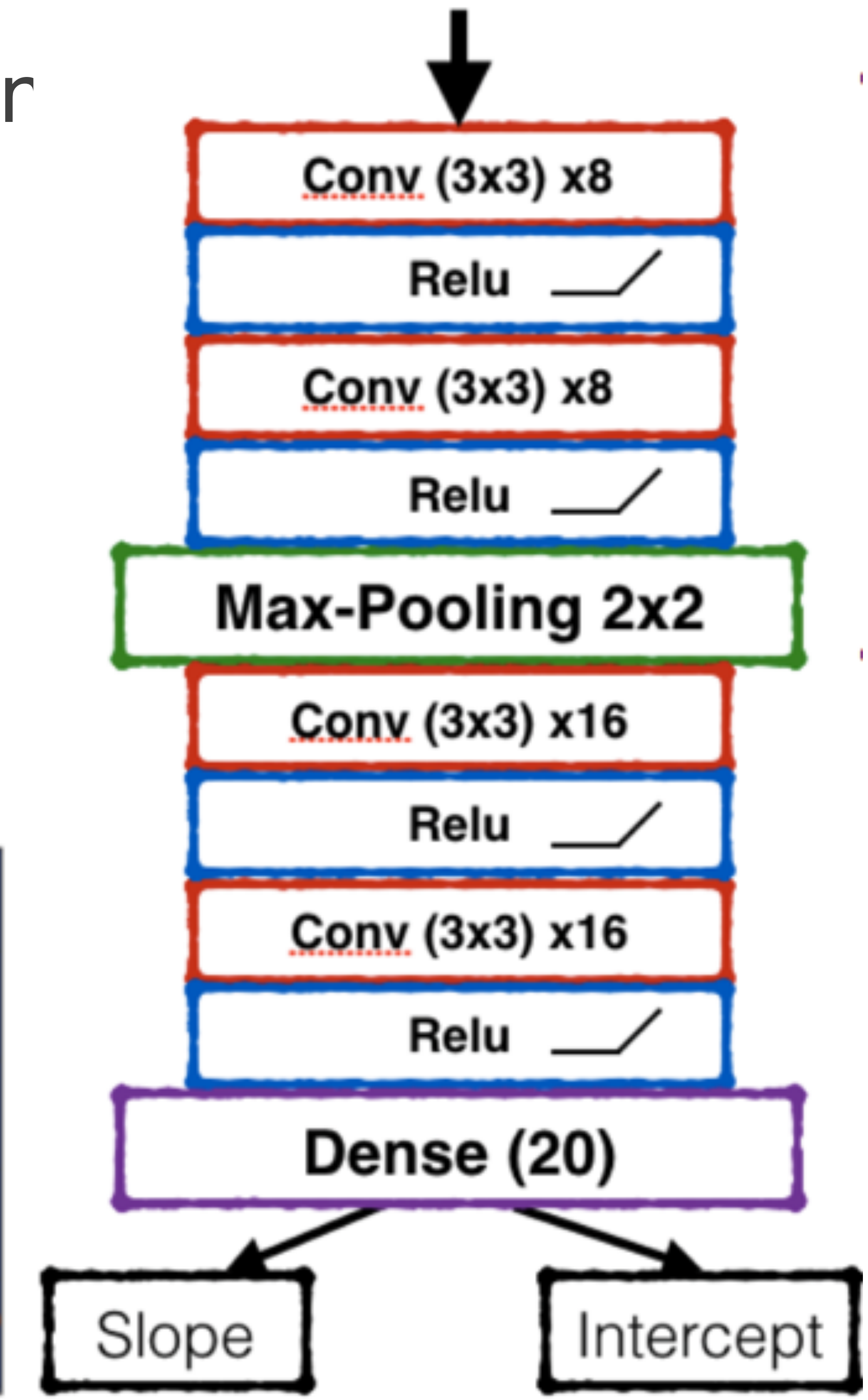


End to End Approach for Predicting Track Parameters

- Target directly on parameter prediction (slope & intercept in this case)

- Example single track with large noise

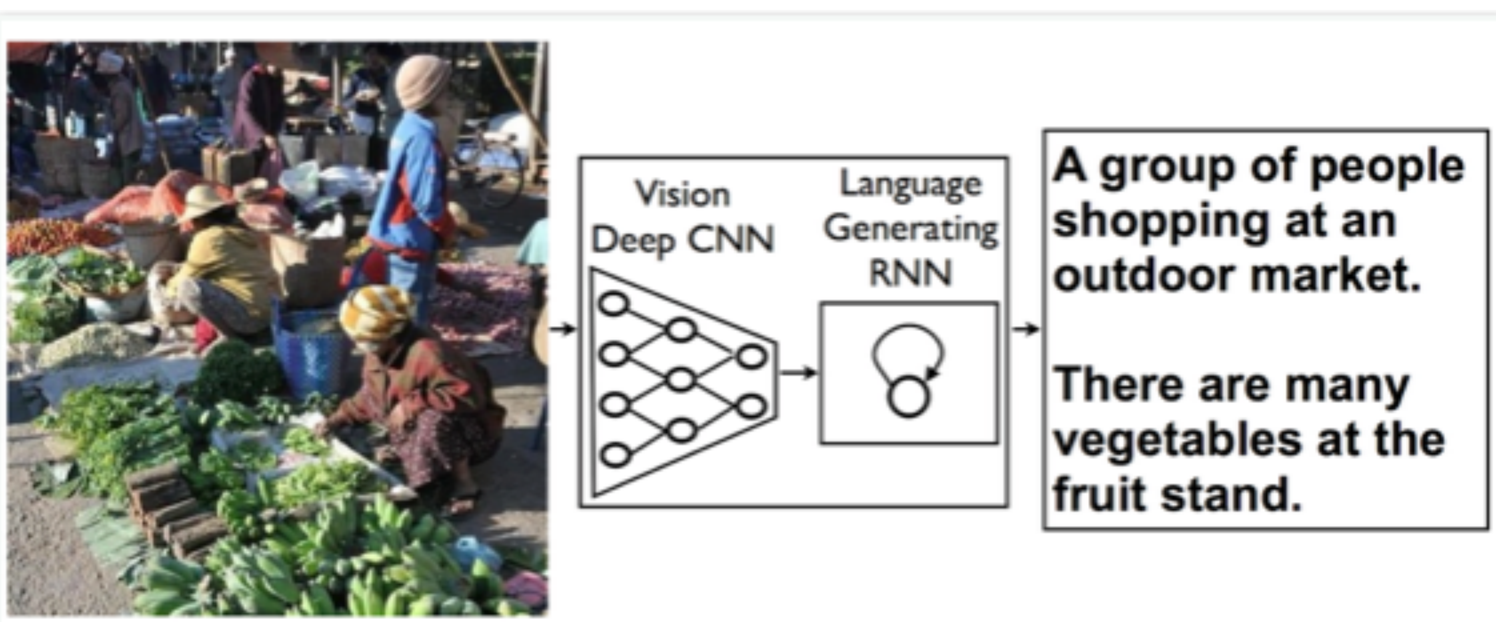


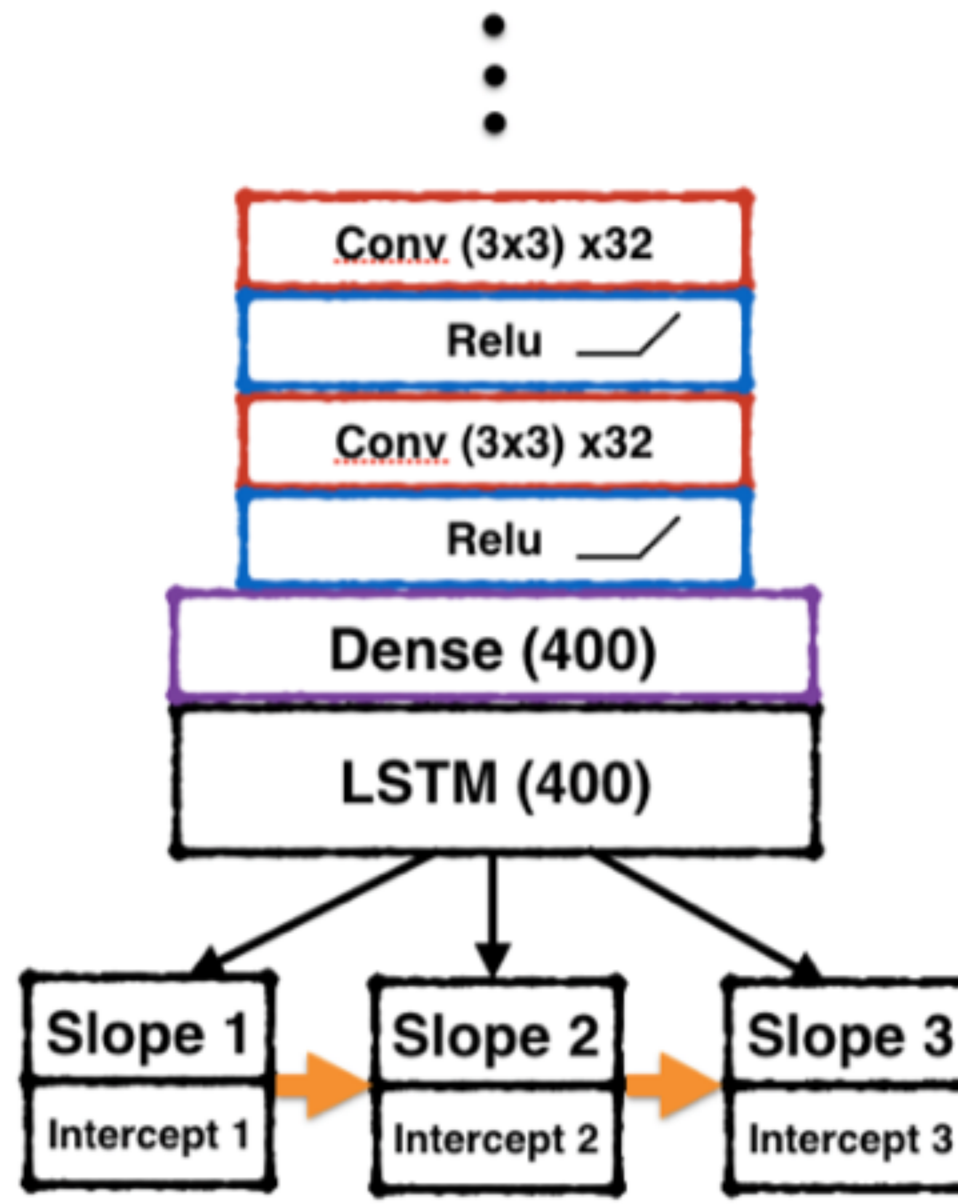


End to End Approach for Predicting Track Parameters

- For many tracks per event add LSTM network at the end

- The memory cell updates to focus on a new track in the image

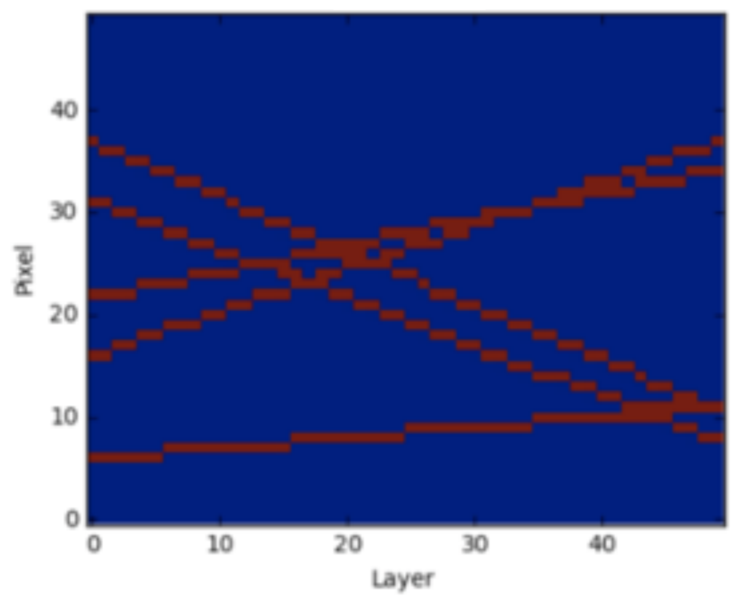




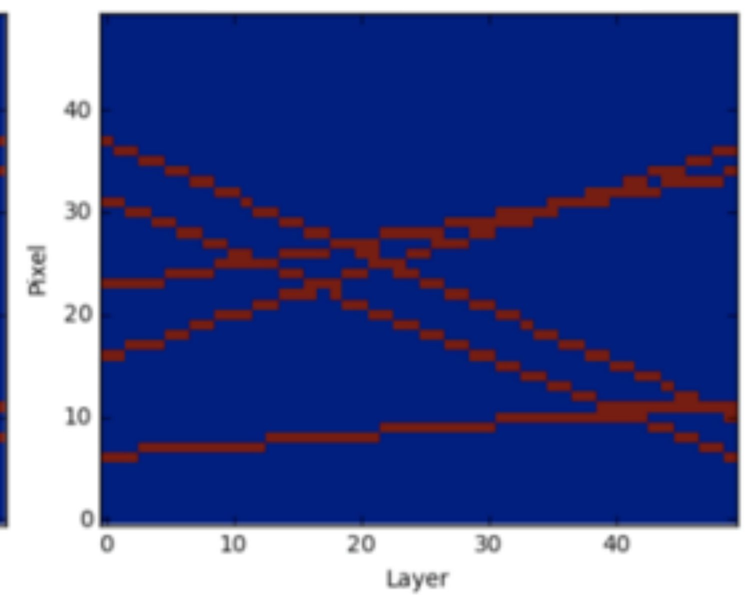
End to End Approach for Predicting Track Parameters

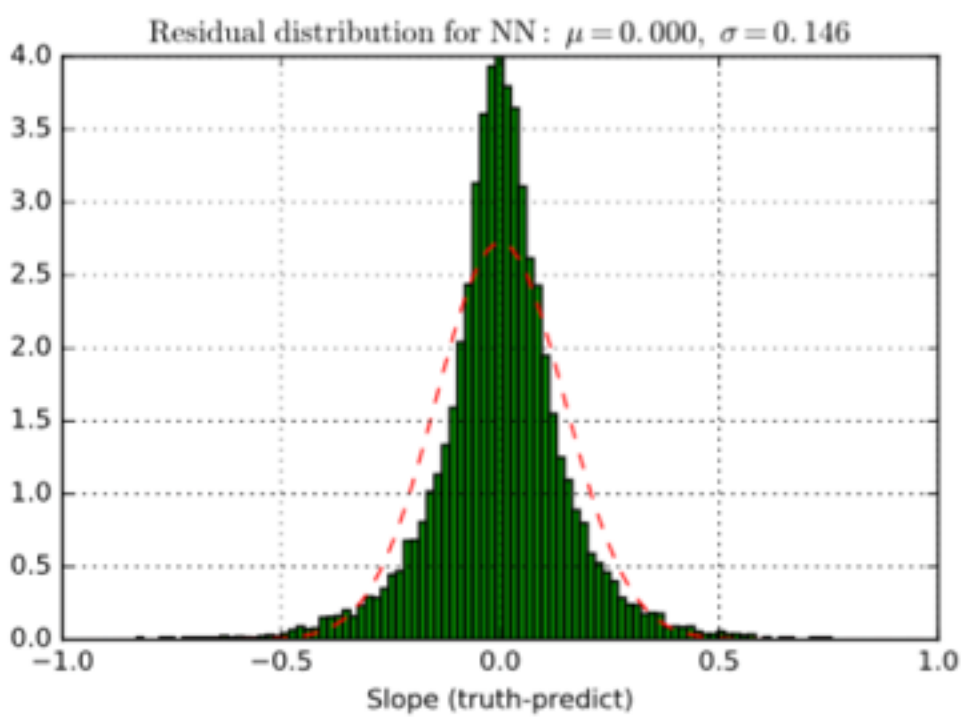
- Model process the image and identifies all track in one pass

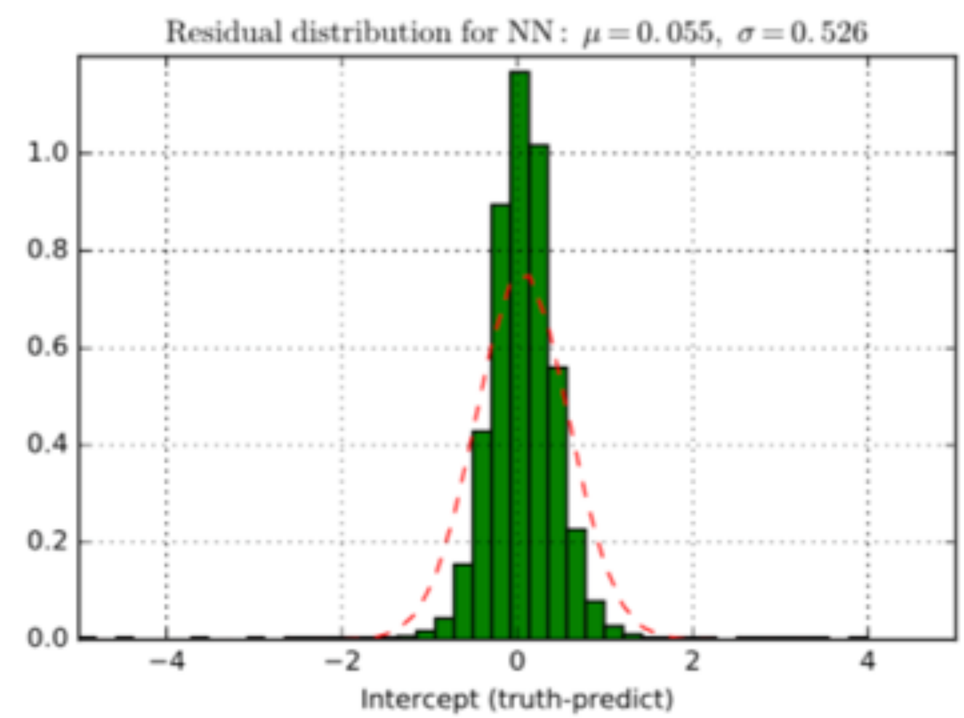
Input



Projected output



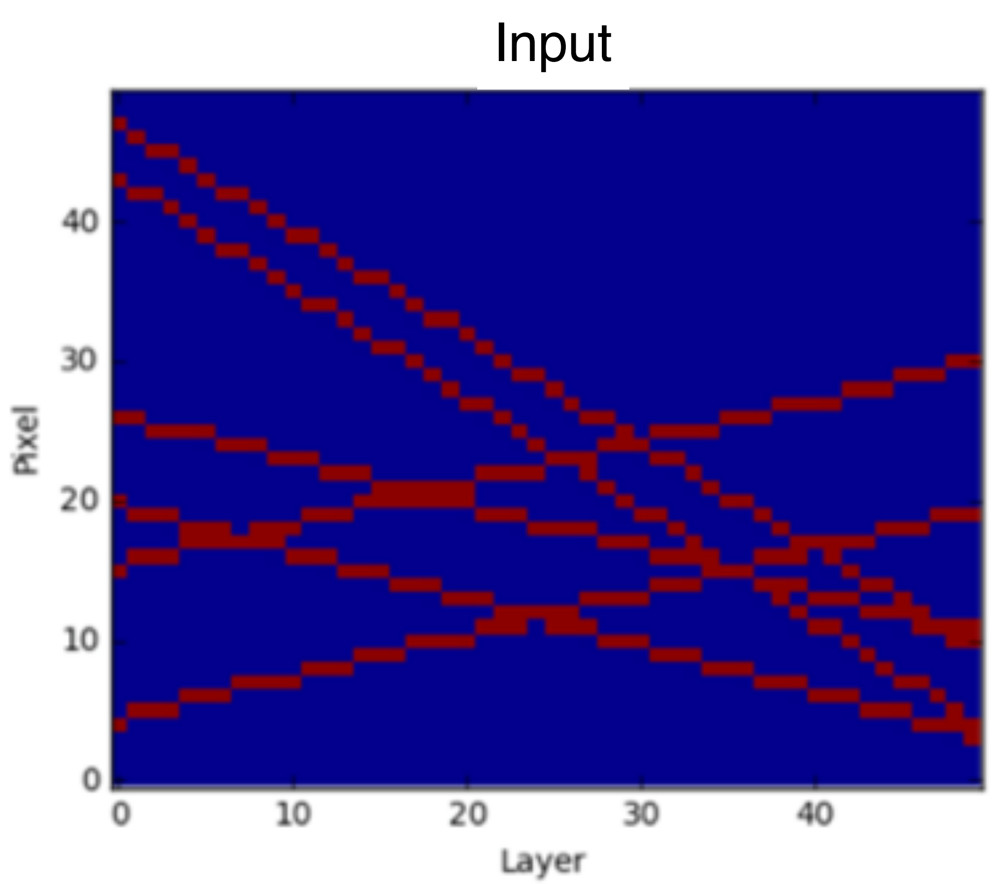


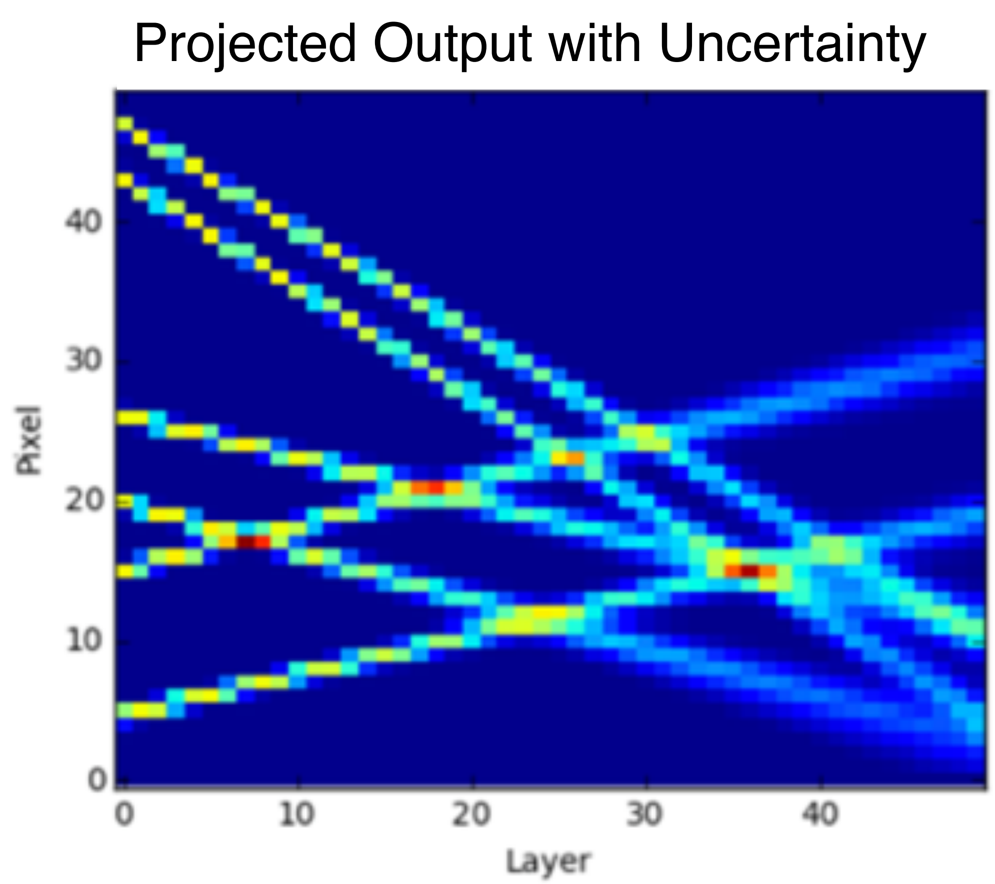


Estimate Uncertainties on Parameters

- Add additional targets to estimate the uncertainties, by minimize negative gaussian log likelihood:

$$L(\boldsymbol{x}, \boldsymbol{y}) = \log|\boldsymbol{\Sigma}| + (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x}))$$

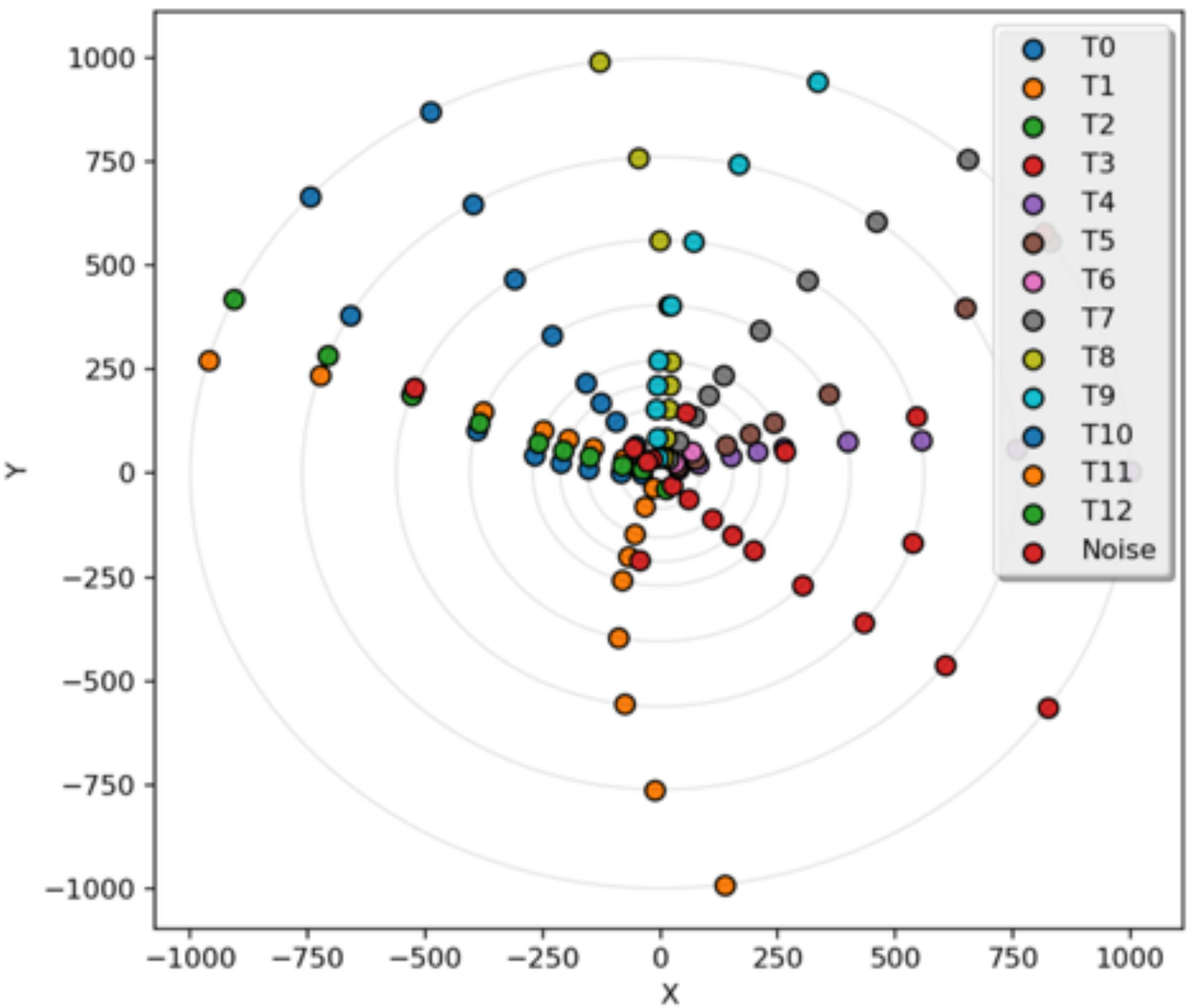




Hit Assignment to Tracks: Detector Geometry

- Using a 3D approach with curve tracks and missing hits we can try to find tracks given a set of hits
  - Base of generate the dataset was the “TrackMLRamp hackathon : a 2D tracking challenge” where an extra dimension was add
  - Events where generated with a constant magnetic field
  - Random noise was also added

XY projection



Hit Assignment to Tracks: Input Format

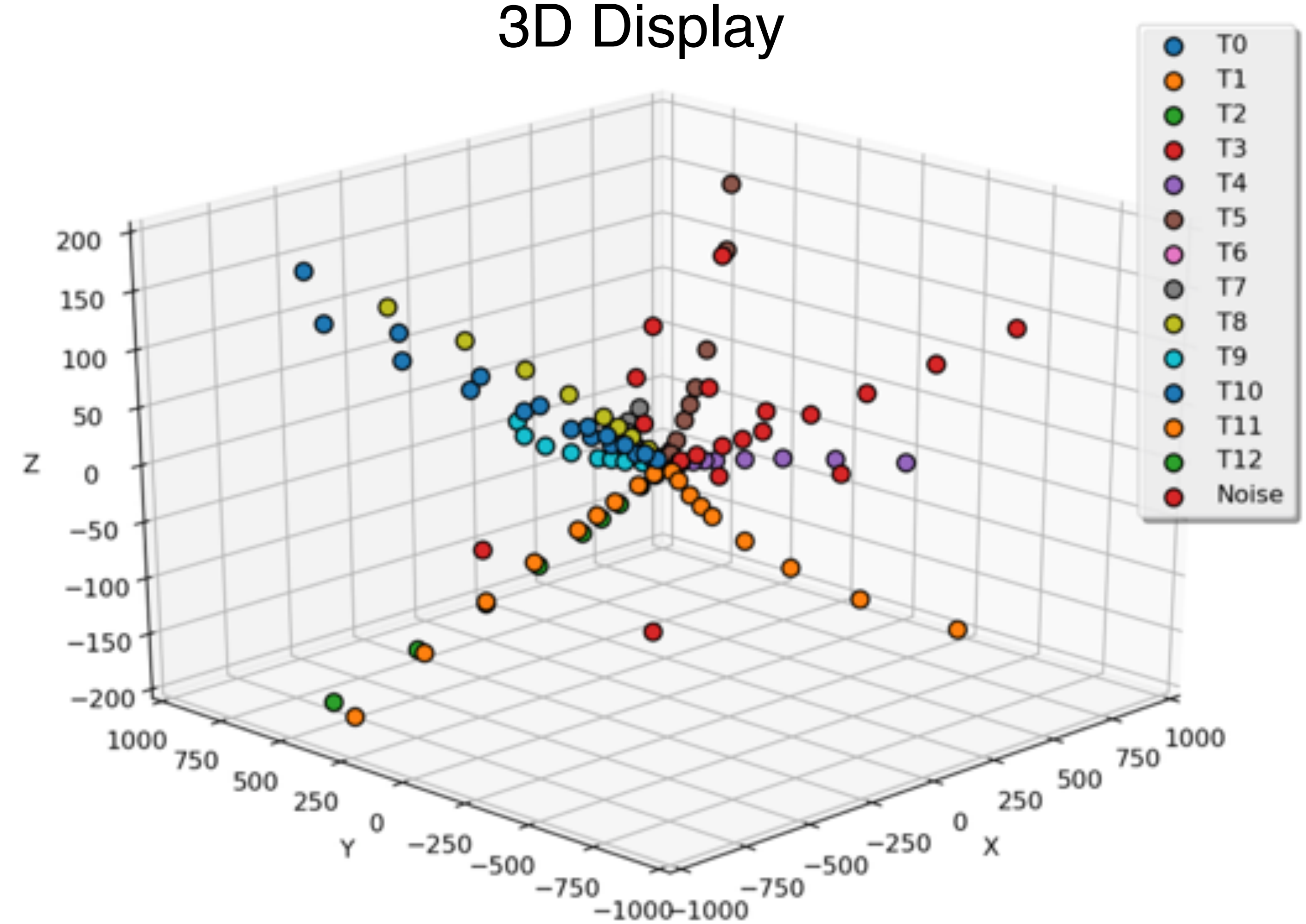
- Input is a set of hit positions in the detector (r, Φ, z in this case)
- All input hits are fed in across all layers in this way

Input Data

	Φ	z	r
0	-2.983331	-165.984650	1000.0
1	-2.795776	-126.480309	762.0
2	-2.650811	-93.283379	562.0
3	-2.578860	7.786216	39.0

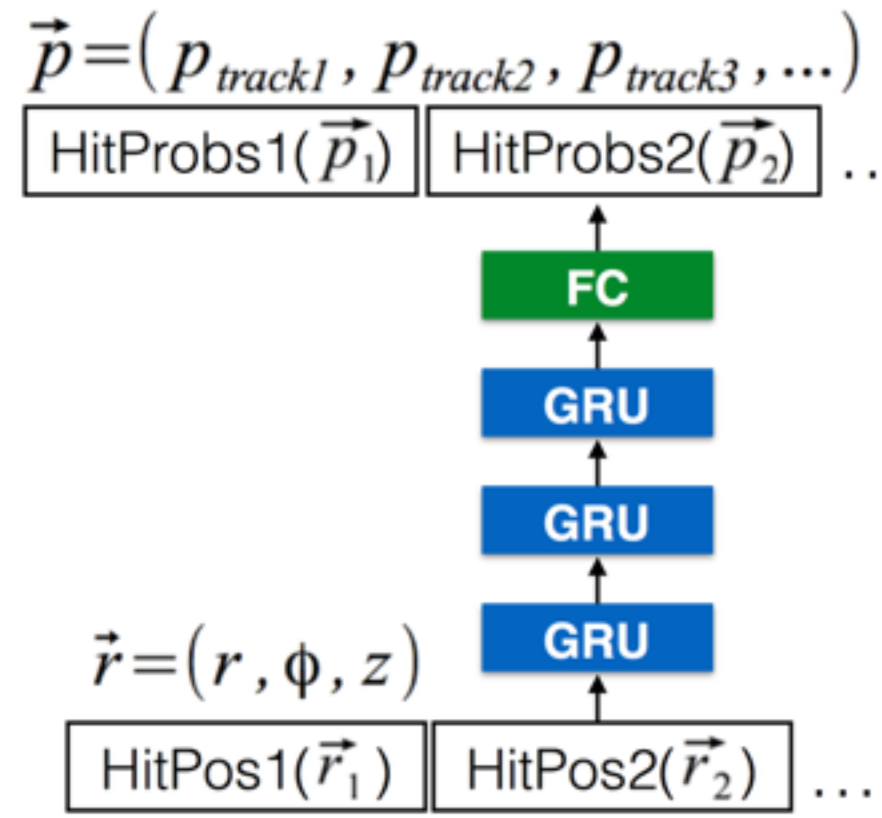
Ground Truth	Track 0	Track 1
Hit 0	1	0
Hit 1	1	0
Hit 2	0	1
Hit 3	0	1

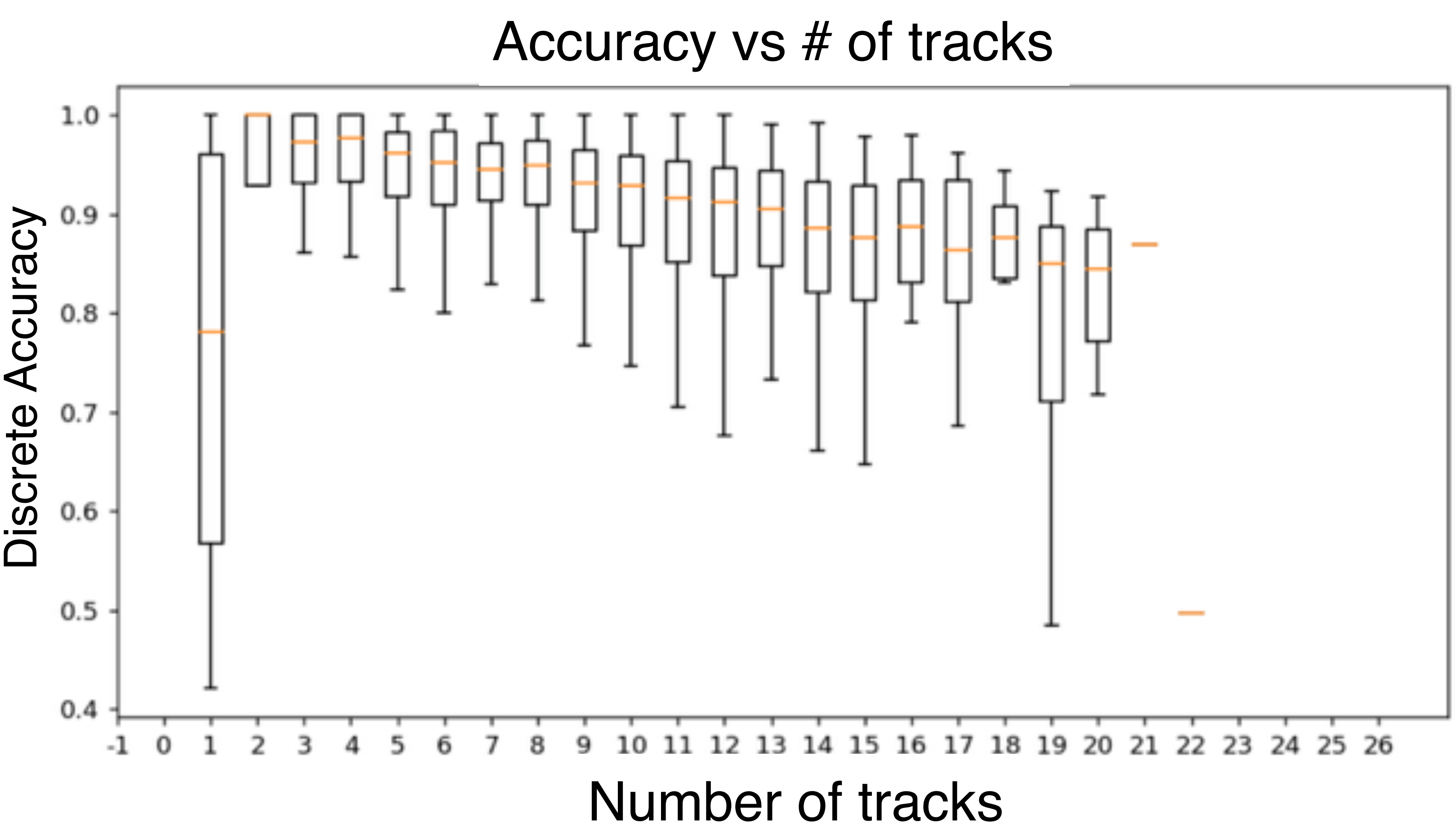
3D Display



Hit Assignment to Tracks: Model and Performance

- Output is a matrix of probabilities of hits in tracks
- Three stack Bi-directional GRU layers were used instead of LSTM (faster training for shallow networks) with two dropout layers in between



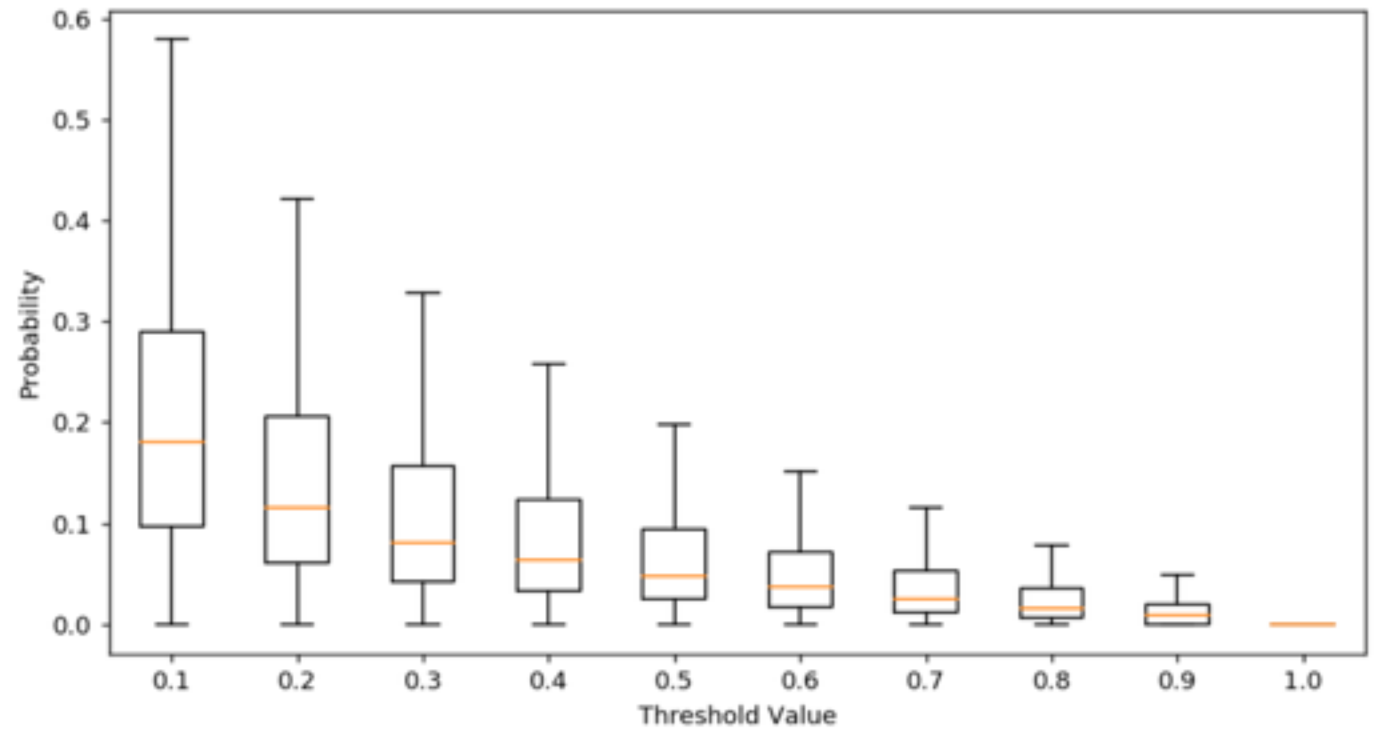


- Accuracy is been calculated selected the hit with the highest probability
- Seem we can benefit on training with higher statistics (train was done with 25K events) on a deeper network

Hit Assignment to Tracks: Performance and Thresholds

- The track selection efficiency can be increased by putting a threshold in the score of hits belong to a track (given small ambiguities)
- Further study is under going

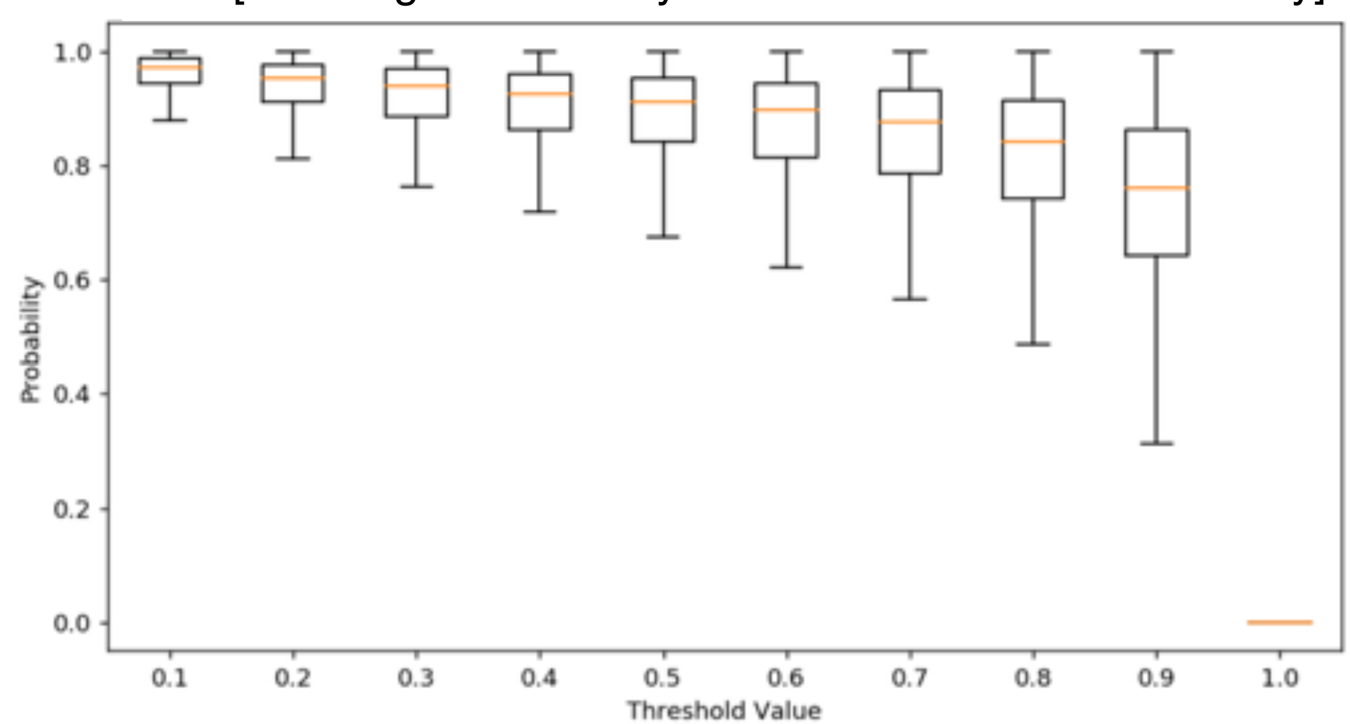
Prob[Hit assigned to multiple tracks with at least threshold certainty]



Output Matrix

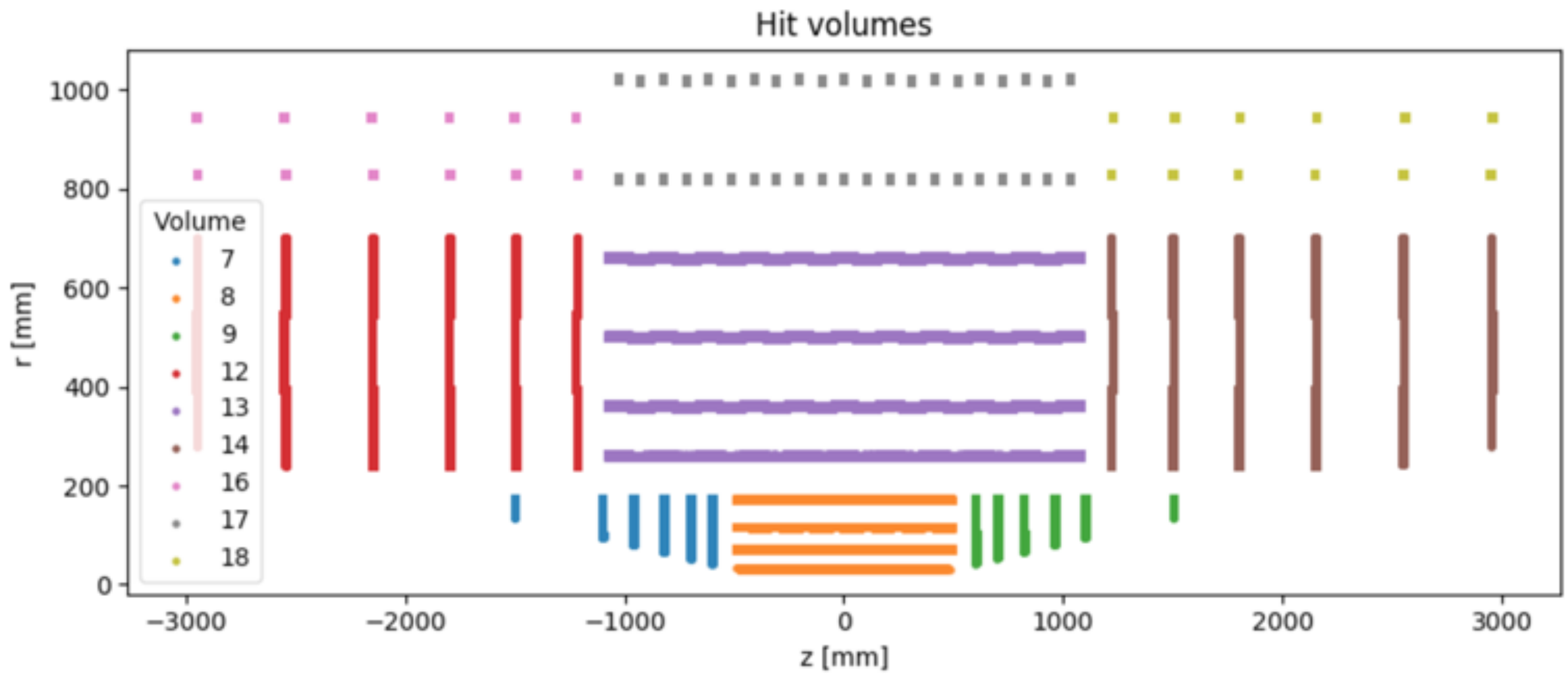
Prediction	Track 0	Track 1
Hit 0	0.8	0.2
Hit 1	0.9	0.1
Hit 2	0.6	0.4
Hit 3	0.1	0.9

Prob[Hit assigned correctly with at least threshold certainty]



The ACTS (A Common Tracking Software) Dataset

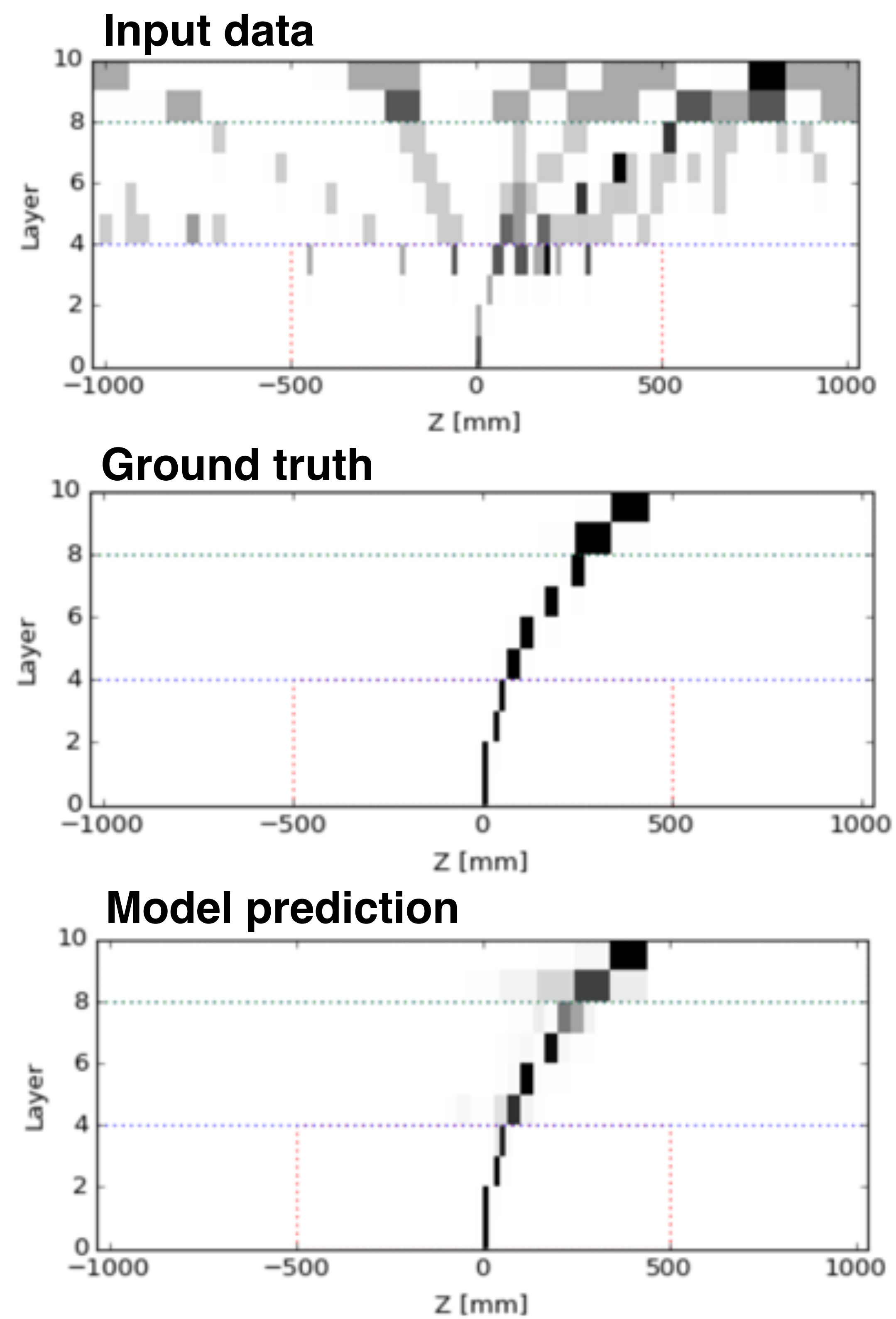
- Increase complexity and realism by using datasets from a LHC-like detector with different layer boundaries
  - Details about the ACTS framework and simulation can be found in [A. Salzburger CTD 2017 talk]



- The barrel volumes (8, 13, 17) have been used for now, but the end-cap hits are also available

Track Finding with LSTM (on ACTS Dataset)

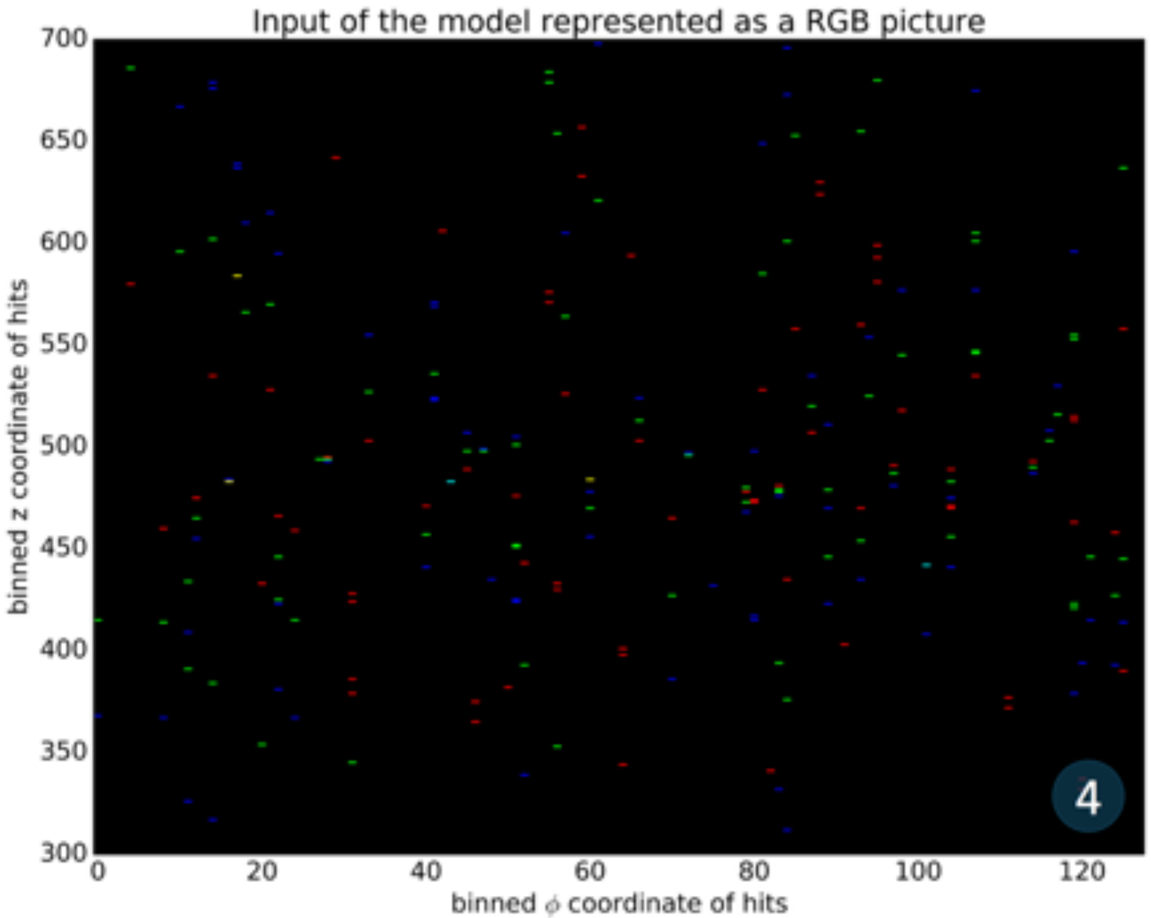
- Apply the 3D hit classification LSTM to a realistic dataset (one projection of the model predictions is shown here)
- Binning was done per barrel detector volume (layer transformation is shared within a volume)
- Promising proof of concept. Detailed studies underway

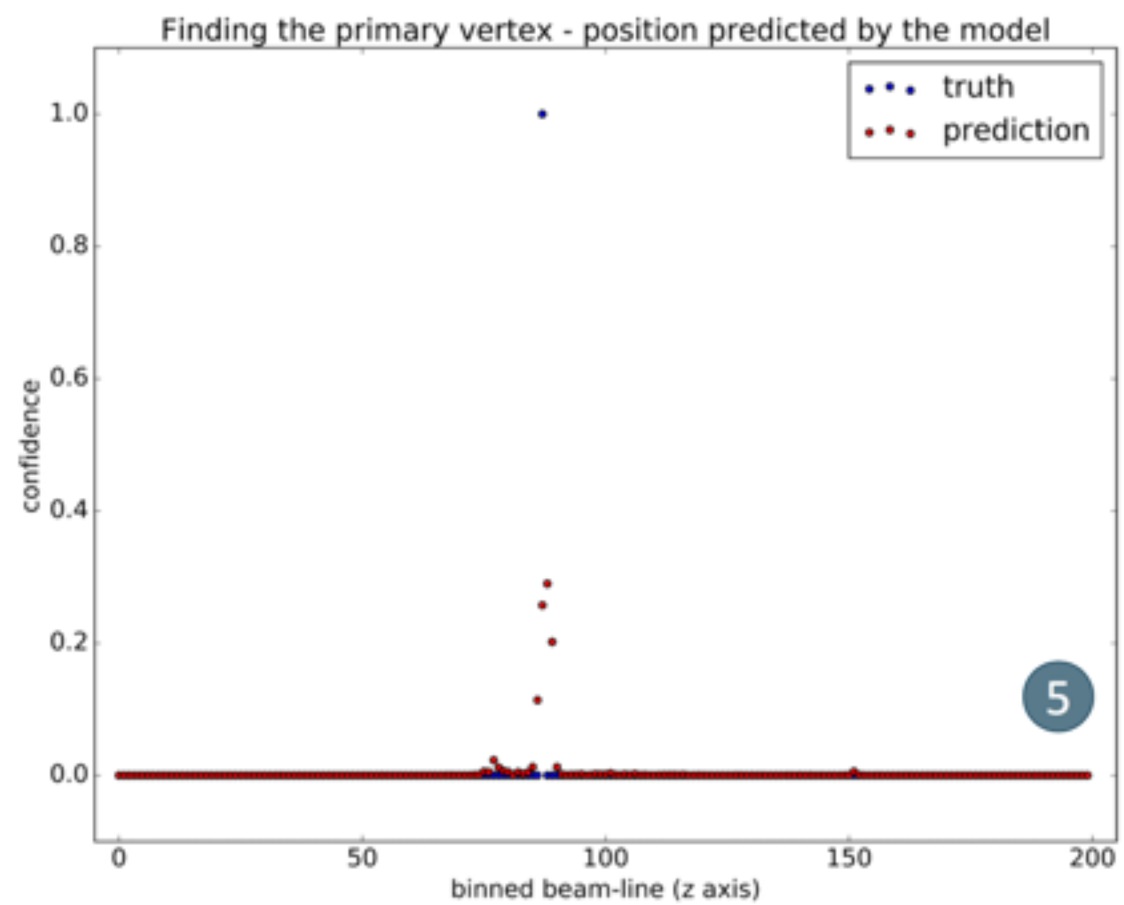


Fermilab

Vertex Finding with CNN (on ACTS Dataset)

- Find the primary vertex position to constrain the seeding algorithm
- An architecture similar to LeNet was used
- Input is the 3 innermost barrel pixel layer hits binned in Z and ϕ (RGB image)
- Output is the binned primary vertex Z position





Vertex Finding with CNN (on ACTS Dataset)

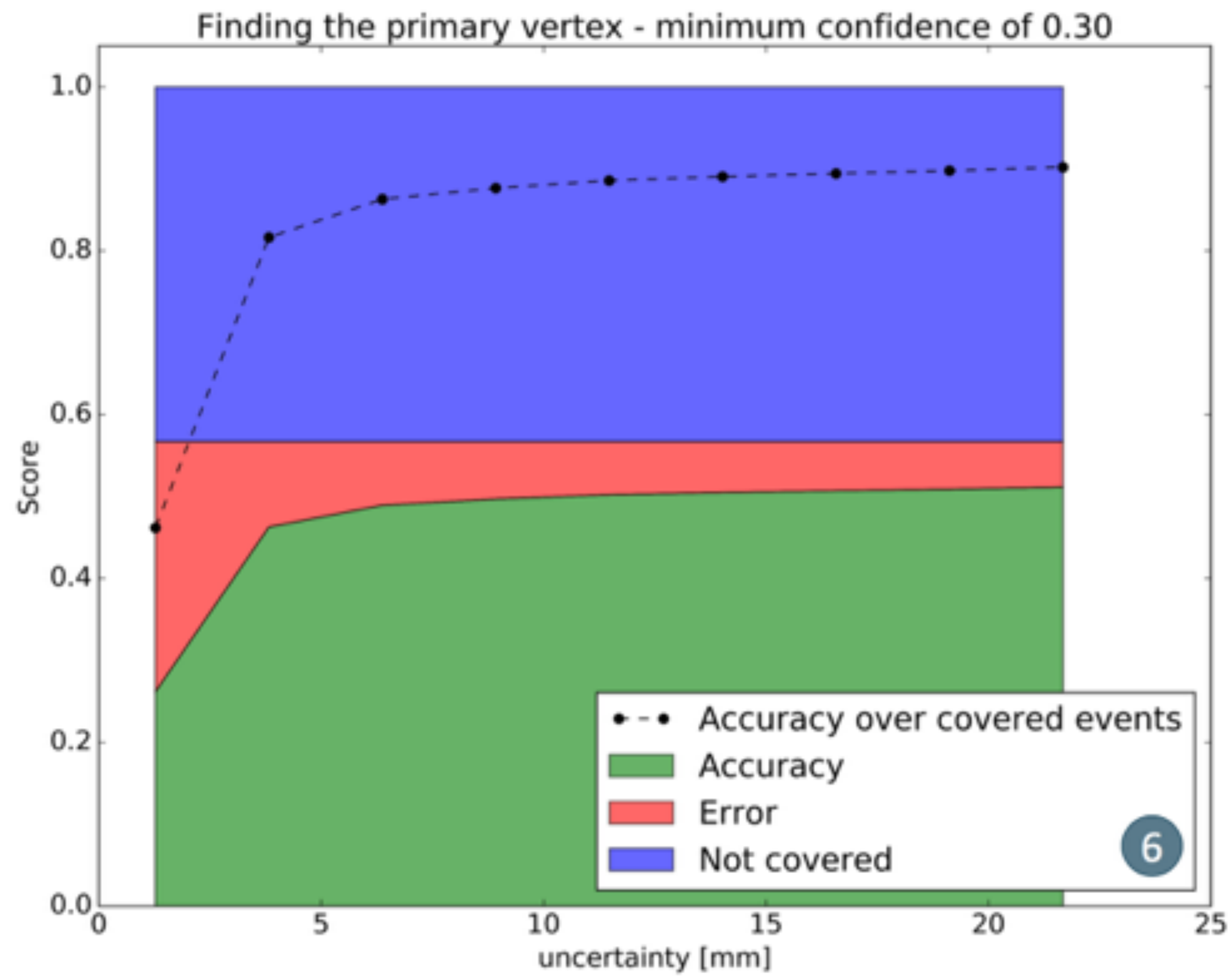
- Both events with single and multiple vertex were explored

Average vertices per event	Accuracy
5	0.46
25	0.15

Performances for the primary vertex finding on two different datasets with a resolution of 7.65mm

- More studies are on going, there is still potential to improve the efficiency of the algorithm.

- Increase the accuracy of the vertex finding by cutting on confidence of the output:
- 81% accuracy for a resolution of 7.65 mm and a coverage of 57%



Conclusion

- Developing a new, scalable tracking algorithm of HL-LHC era is critical for detector performance
- The HEP.TrkX project is formed to explore ideas for applying ML algorithms for this problem
- RNN and CNN showed promising results in toy model testings
- Going forward:
  - Increase the complexity and realism of the problem. This has already started with a fix framework (i.e. ACTS generic tracker)
  - Converge to the most promising ideas and study them in depth
  - Compare the performance with novel algorithms (i.e. Parallel Kalman Filter)
- ***Stay tune for further results !!!***

Backup

Previous HEP.TrkX Presentations

- Convolutional NNs for Tracking DS@HEP Workshop at FNAL, **Steve Farrell**
- HEP.TrkX IML Machine Learning Workshop, **Dustin Anderson**
- The HEP.TrkX project: Deep Neural Networks for HEP Tracking @ CTD/WIT 2017, **Steve Farrell**
- ML LHC Tracking Challenge@ CHEP 2016, **Paolo Calafiura**