Fermilab Updates

Aristeidis Tsaris

March 20, 2017

Overview

- 1st Part:
  - Look the residuals and color map them according to the number of tracks of the event that they came from.
  - Compare between Conv and Conv+LSTM.
    - Is there different mapping when you look the pattern recognition style, or when you go deeper in the track dynamics?

Overview

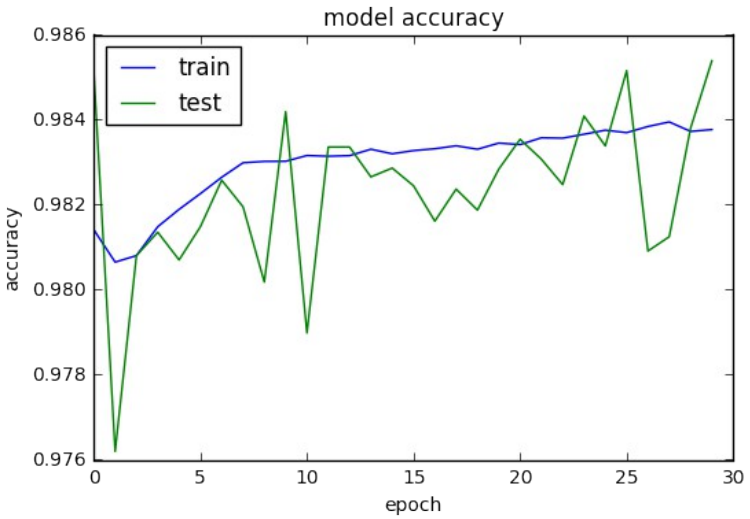
- 2nd Part:
  - The uniform noise is not realistic, add a smearing to the track and that way we can compare it to a linear fit for example.
  - Look events that fail the residuals (unreasonable large values).
  - Compare Conv with Conv + LSTM for 1 track events and “maximum 6 track events”.

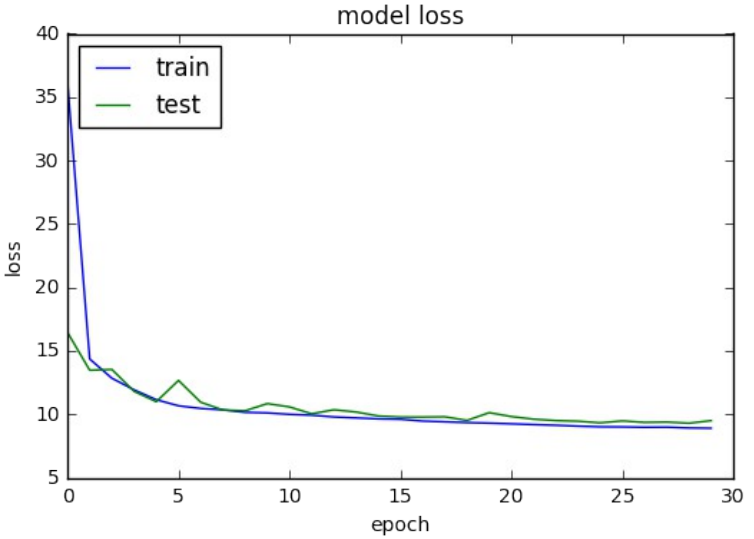
Same Parameters for both Conv and Conv+LSTM

```
# Training parameters
batch_size = 64
epoch_size = 64000
num_epochs = 30

# Data parameters
det_width = 5
det_depth = 50
det_shape = (det_width, det_depth)

# Number of tracks in each event follows Poisson distribution
mean_tracks = 3
max_tracks = 6
```

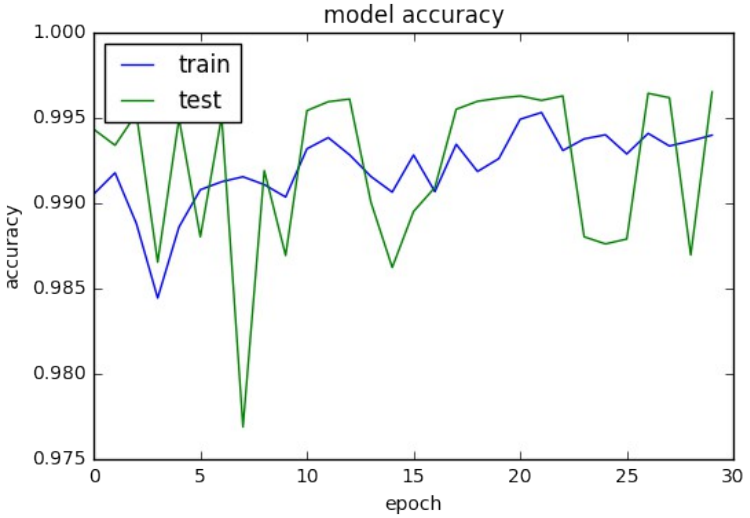
Training Performance

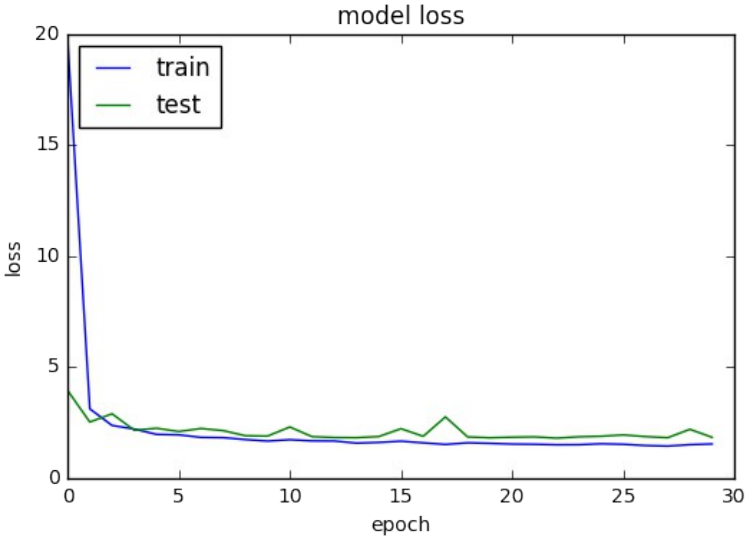
Convolutional Layers :



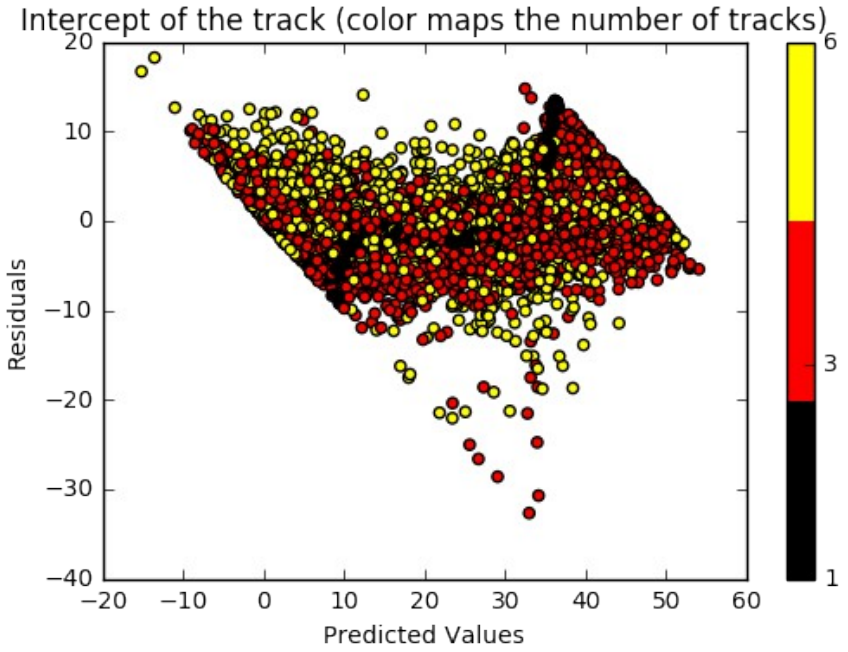


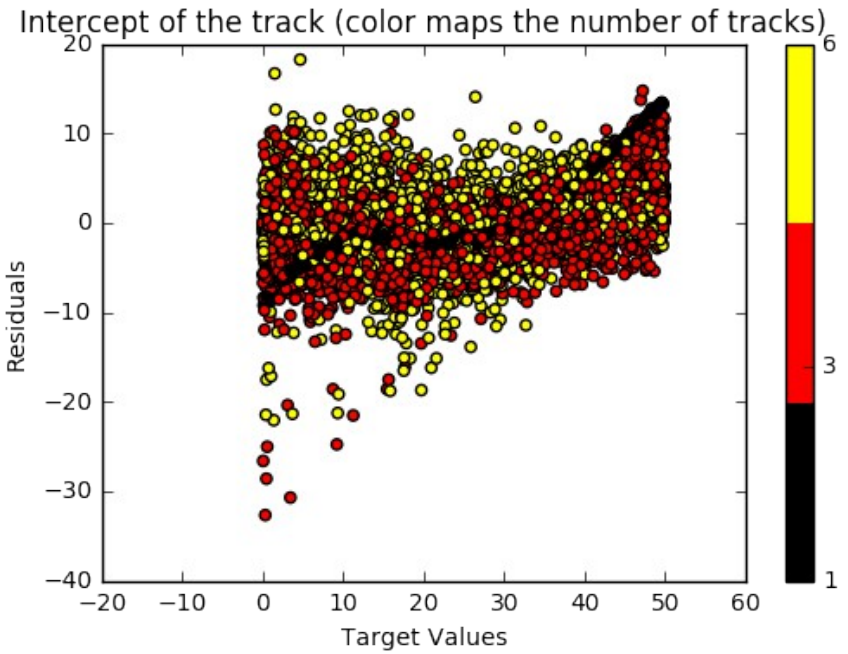
Convolutional + LSTM Layers :

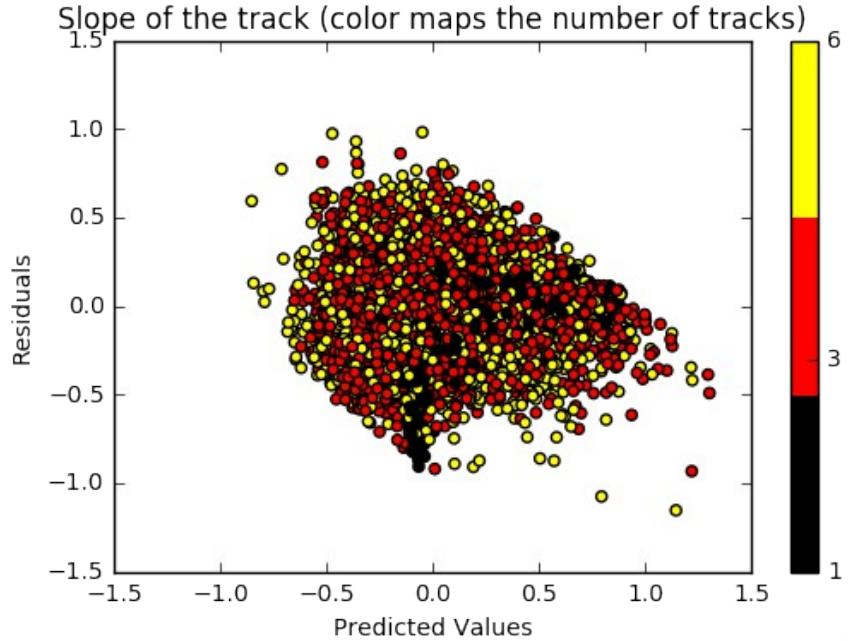


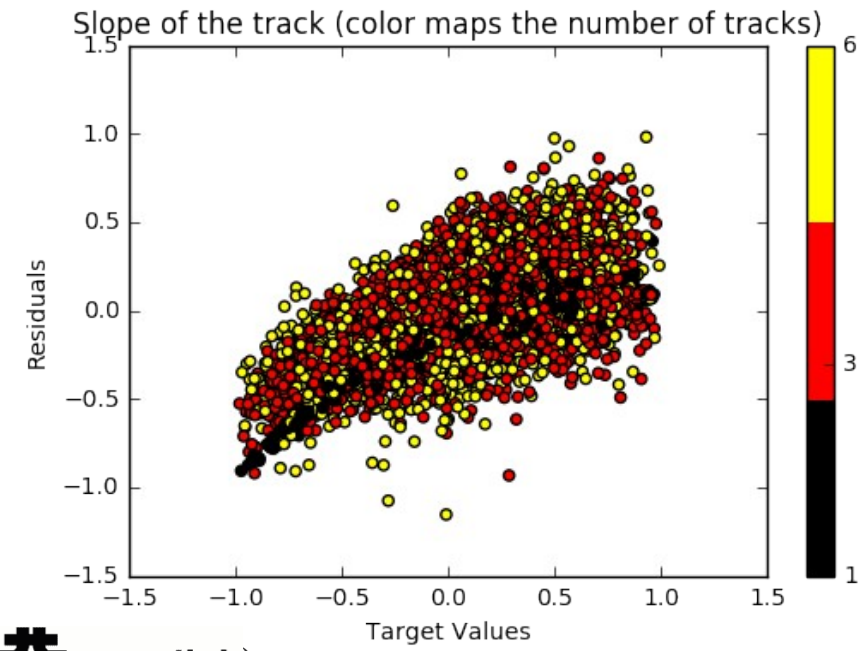


NN with Conv layers

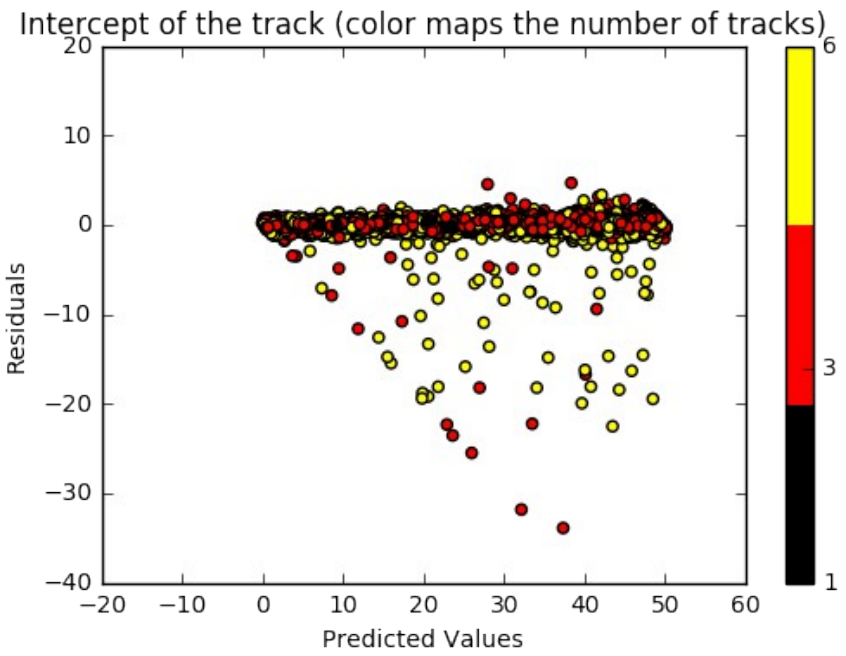


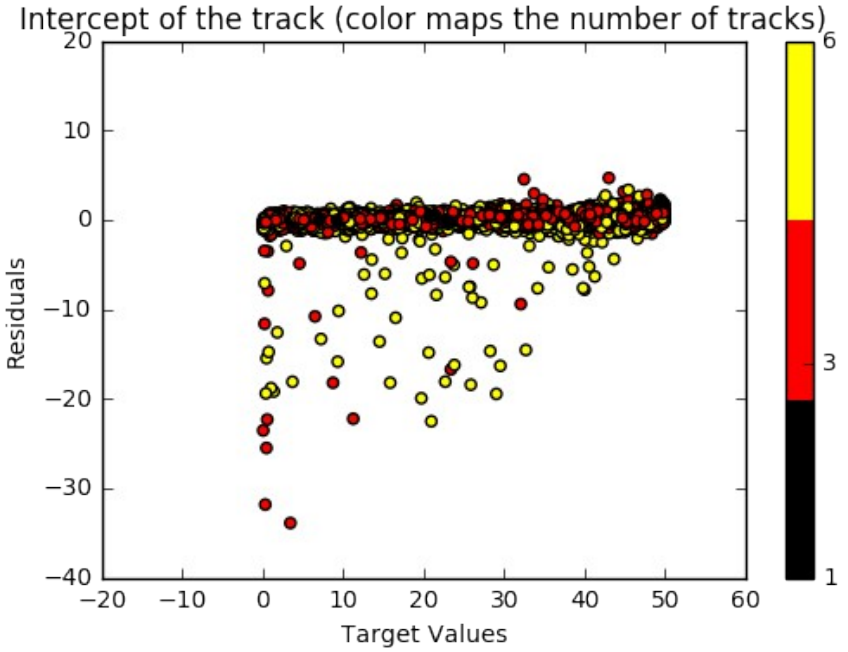


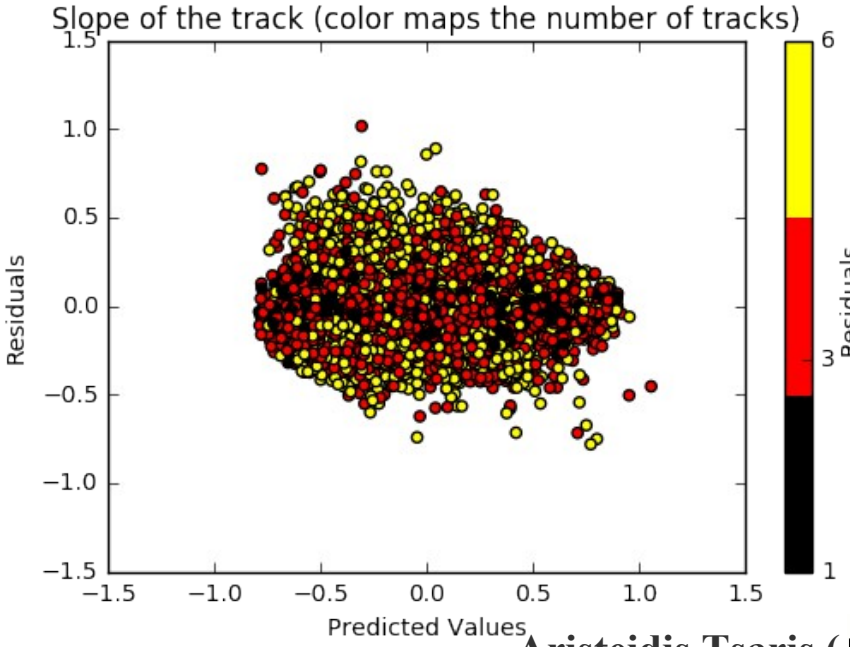


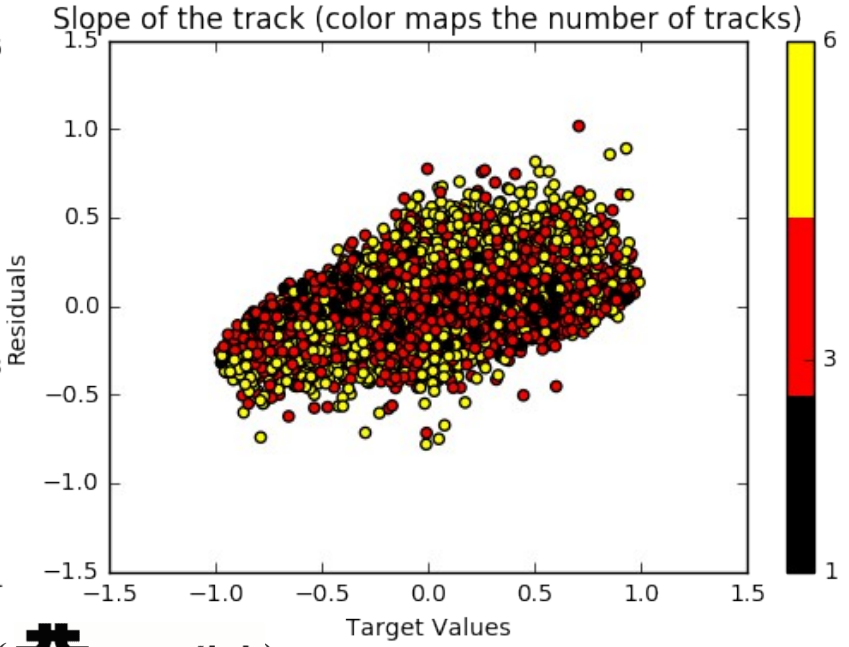


NN with Conv + LSTM layers

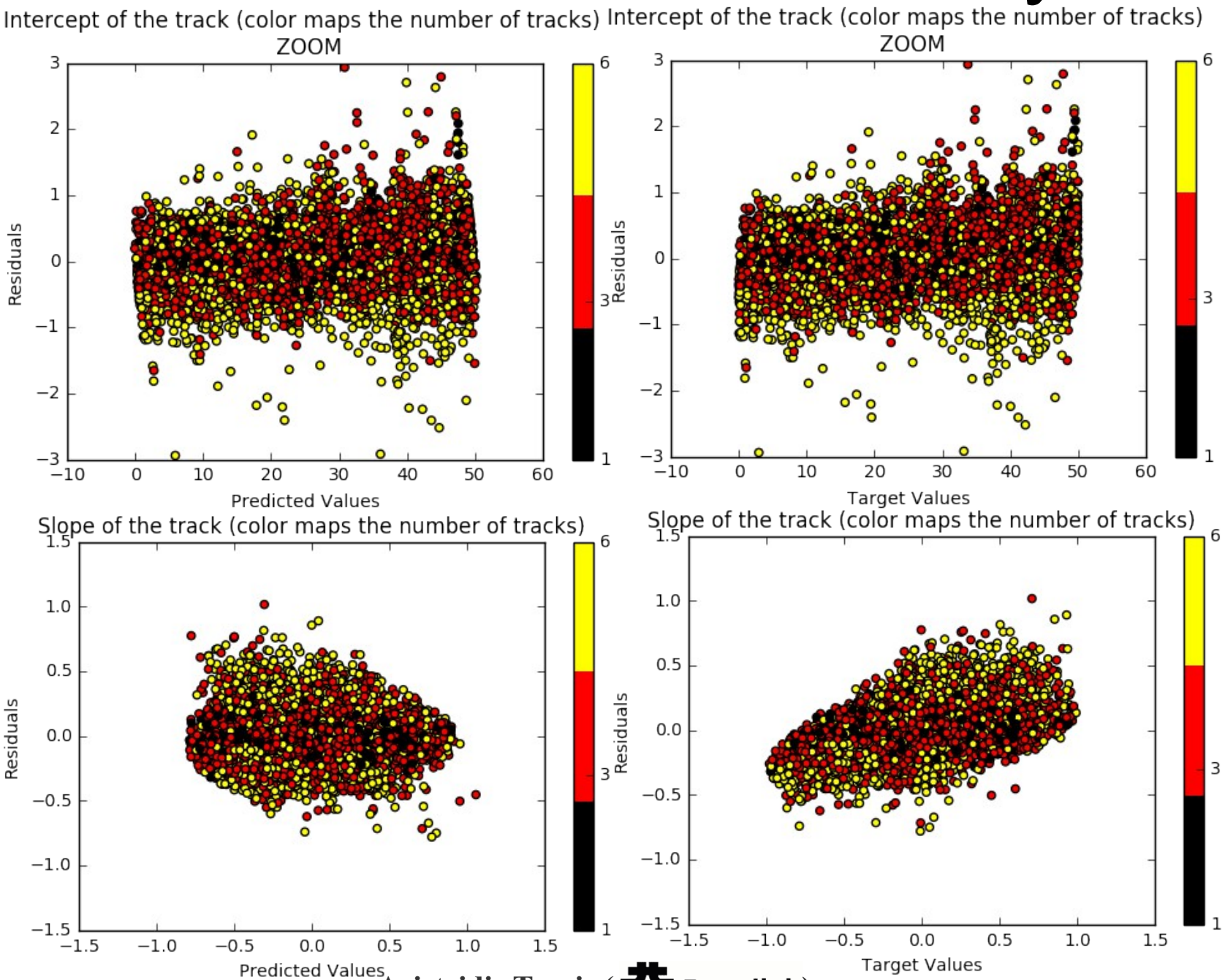






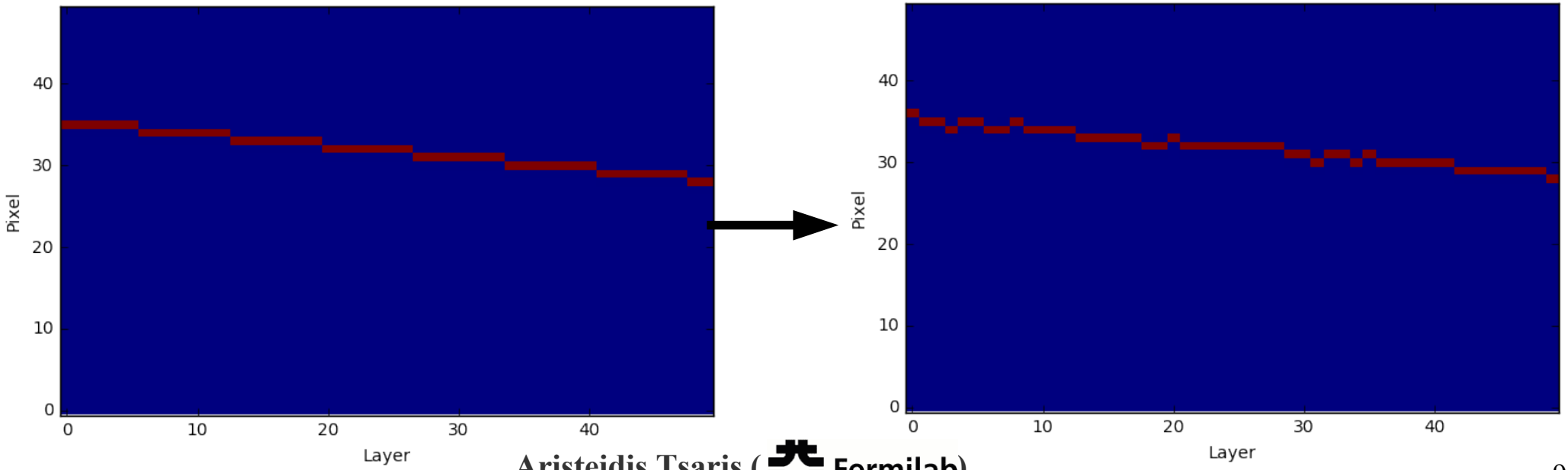


NN with Conv + LSTM layers



Smear the Track && LinFit

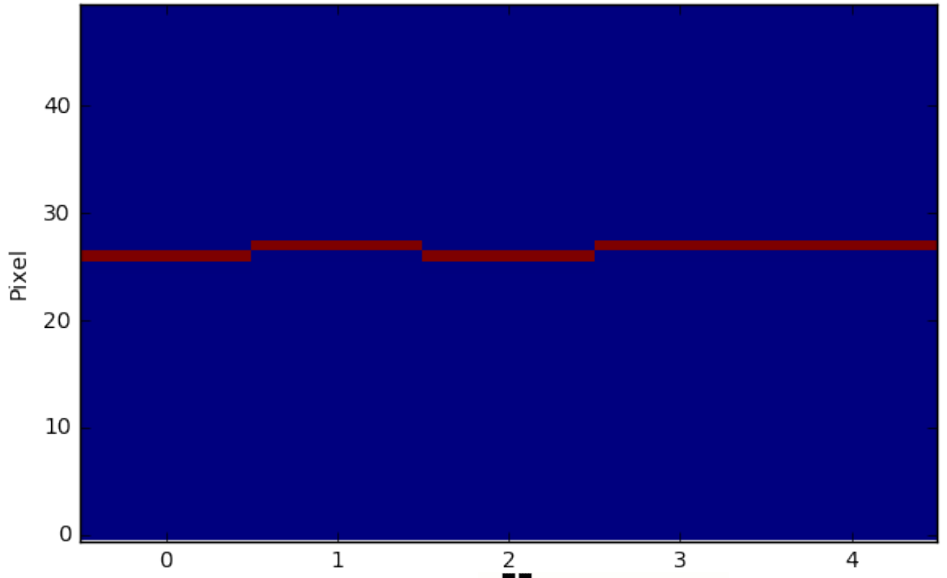
- When making the hits draw a random number from a normal distribution with width 1/Sqrt(12). This allows to go to the next pixel randomly and account the detector resolution.
- Then do a linear fit for each track separately. In that case we compare the fit of the track but not track finding.
- That way it is a fair comparison for single track events.



Smear the Track && LinFit

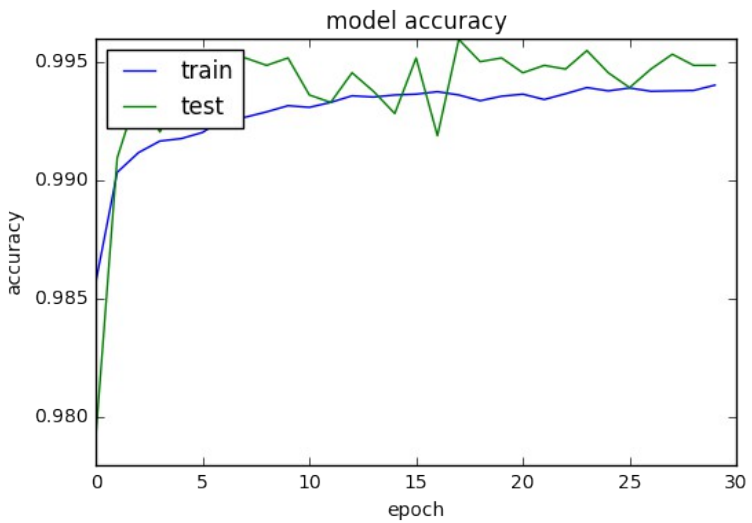
- When making the hits draw a random number from a normal distribution with width 1/Sqrt(12). This allows to go to the next pixel randomly and account the detector resolution.
- Then do a linear fit for each track separately. In that case we compare the fit of the track but not track finding.
- That way it is a fair comparison for single track events.

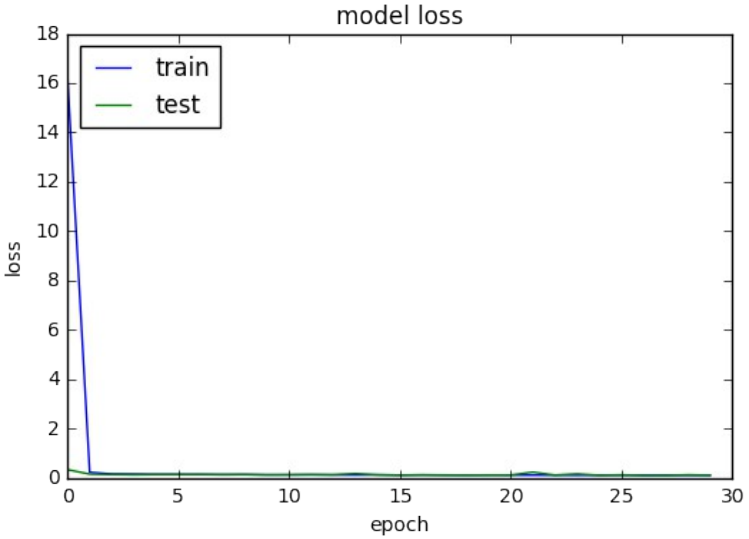
I am looking to a 5x50 image →



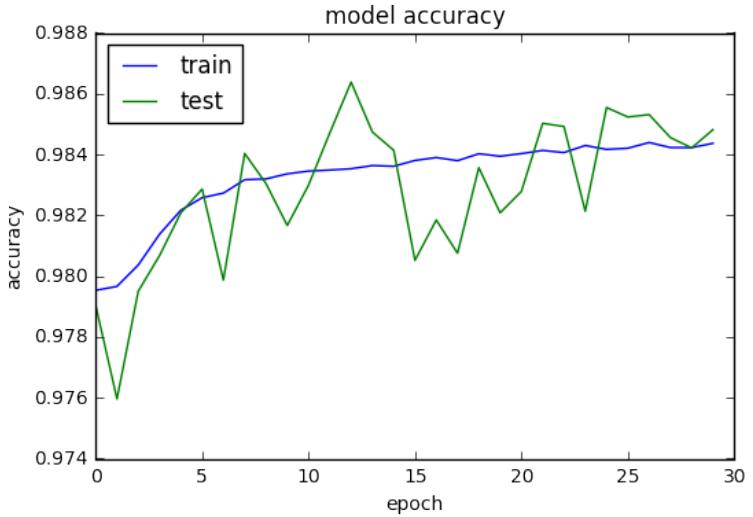
Training Performance

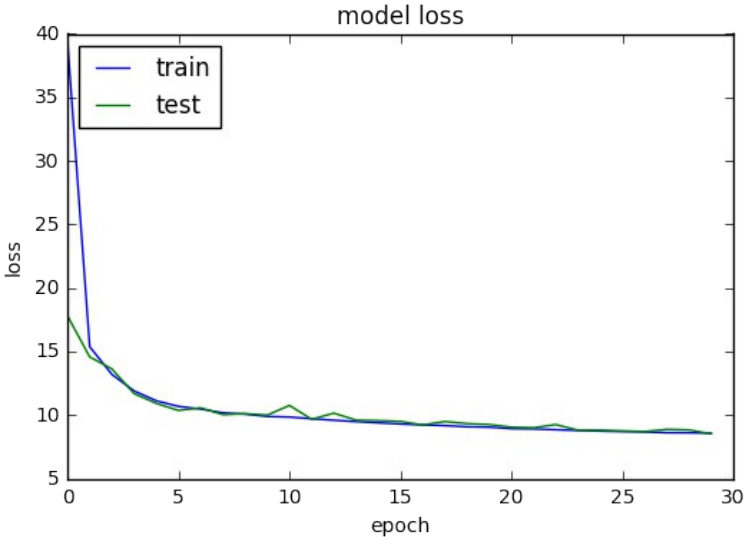
Convolutional Layers Single Track Events:



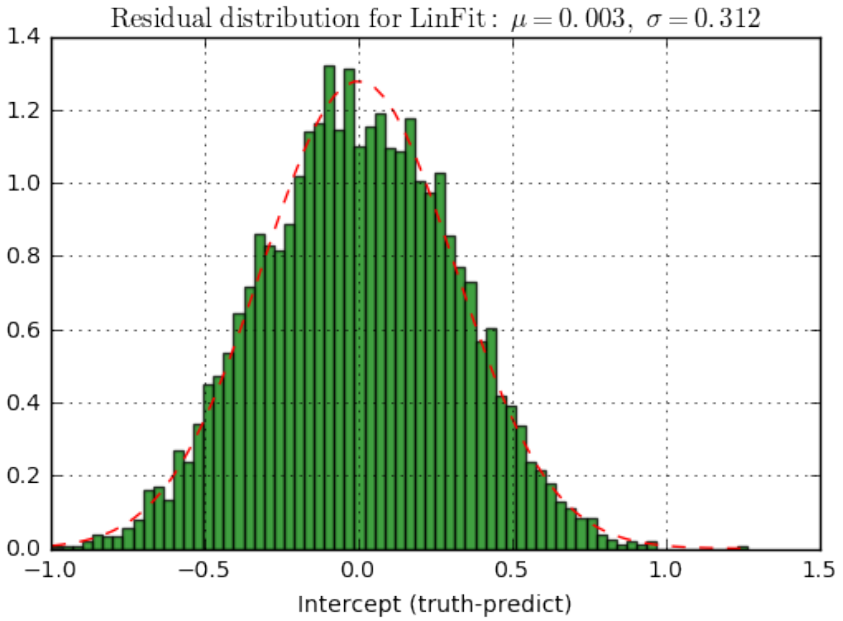


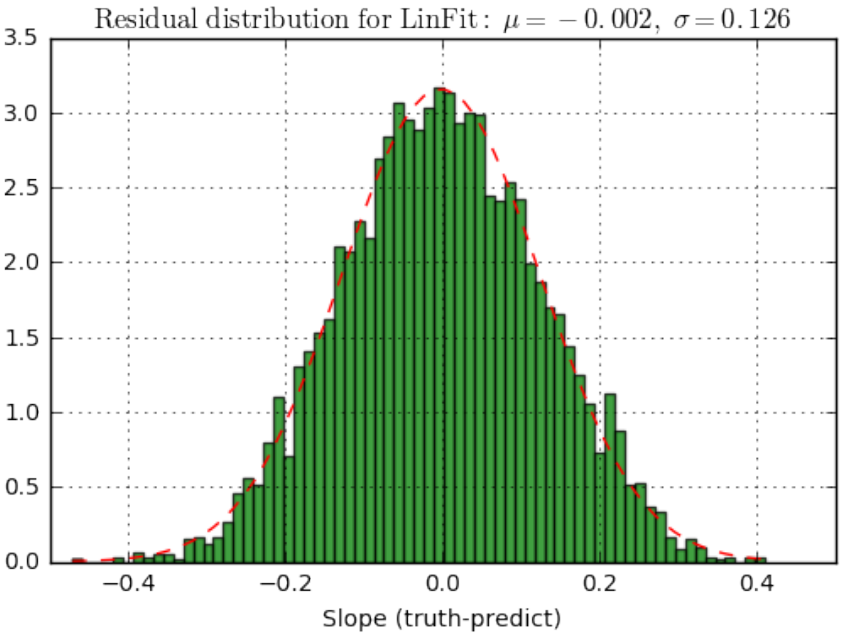
Convolutional Layers Multi Track Events:

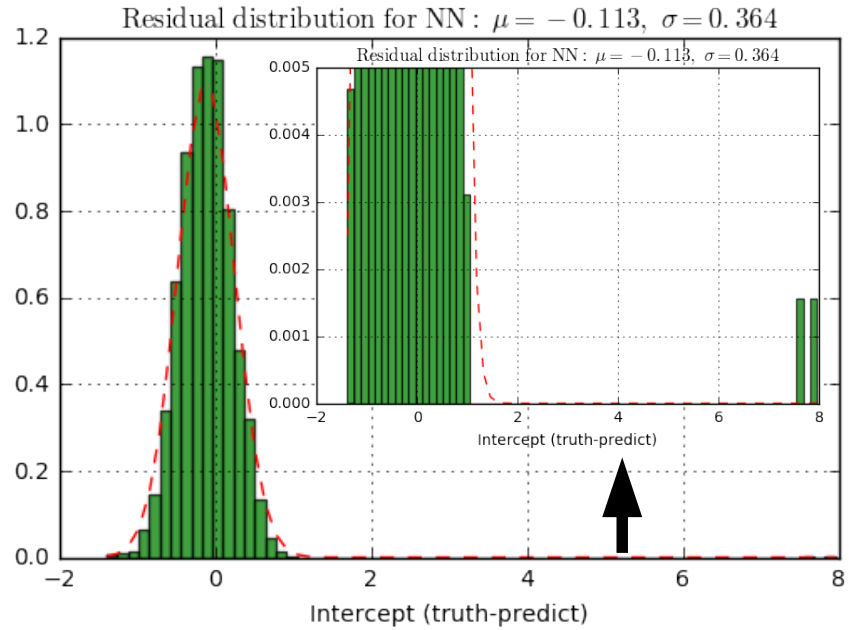


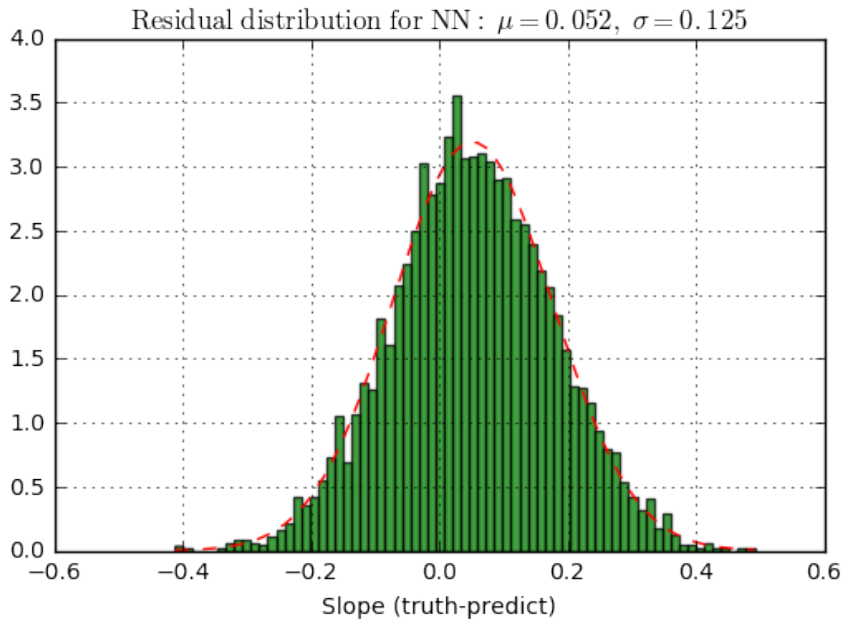


NN with Conv Layers (Single Track Evt)

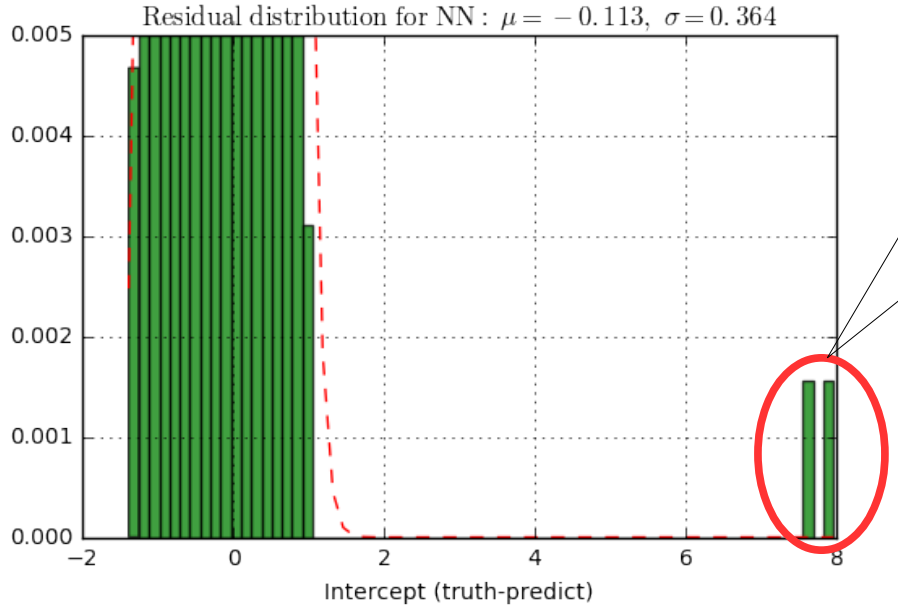


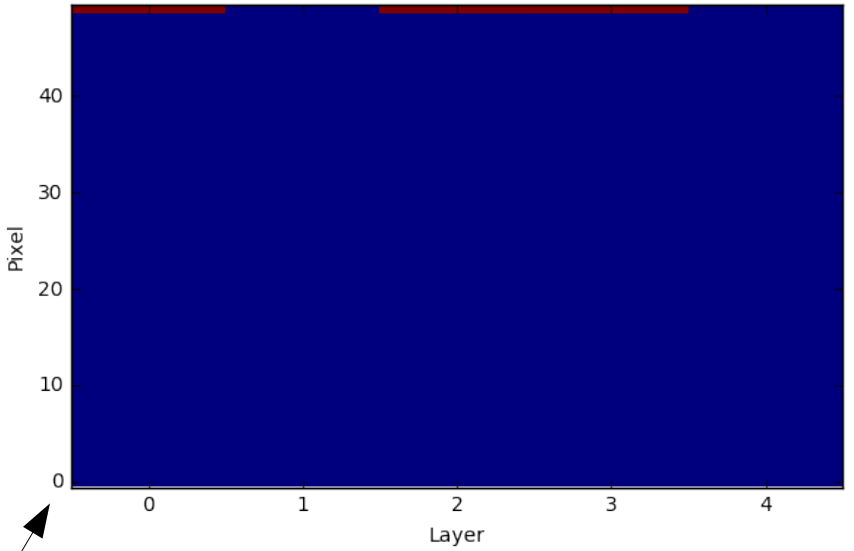


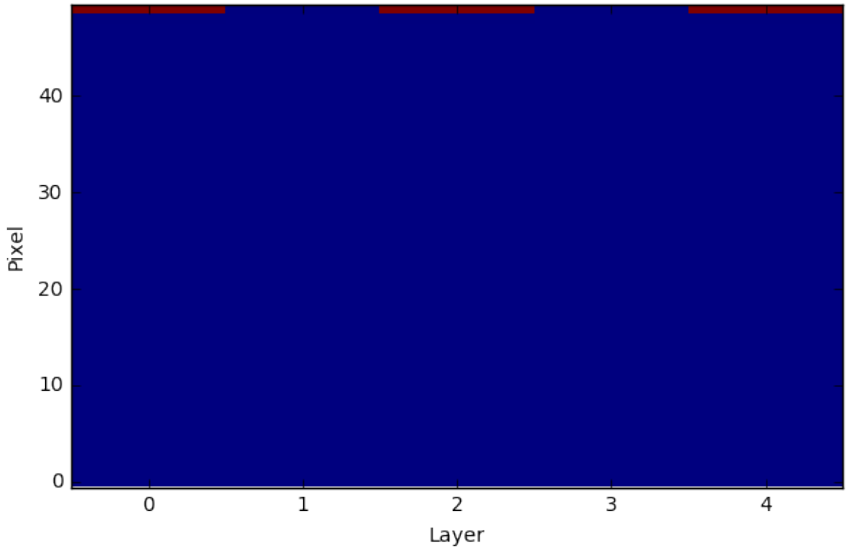




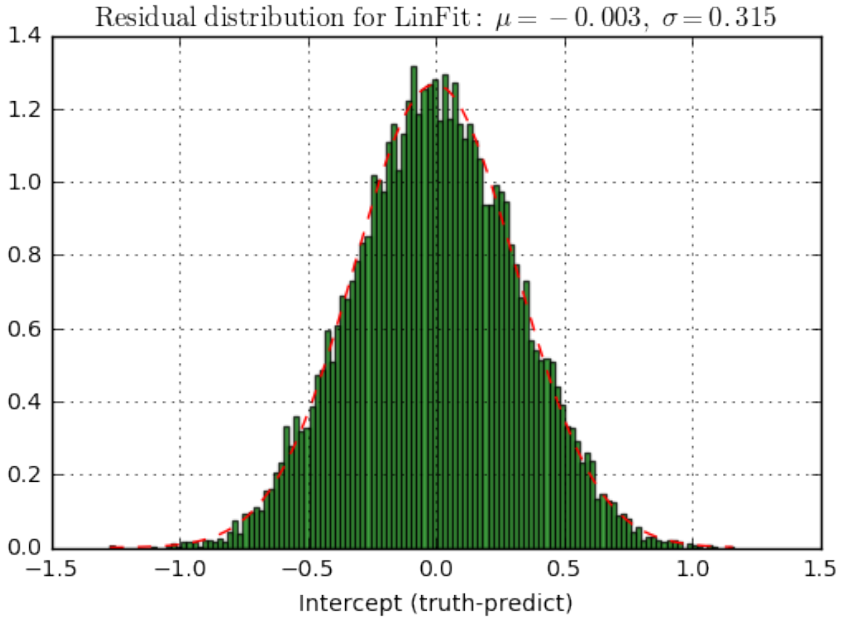
NN with Conv Layers (Single Track Evt)

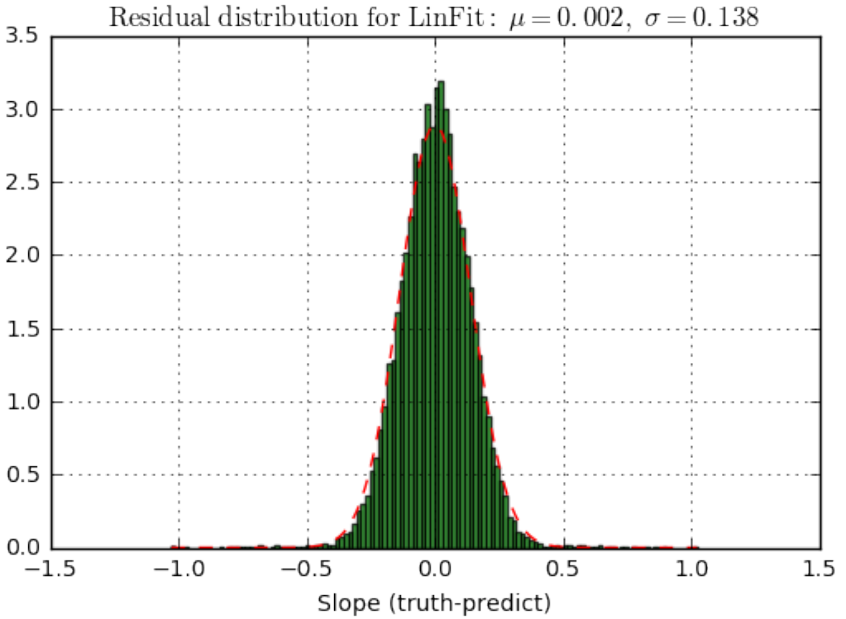


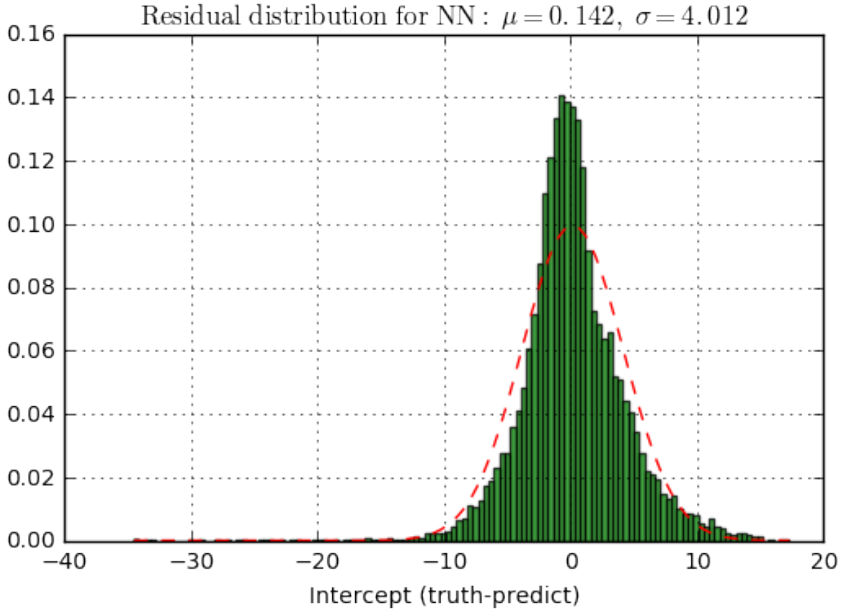


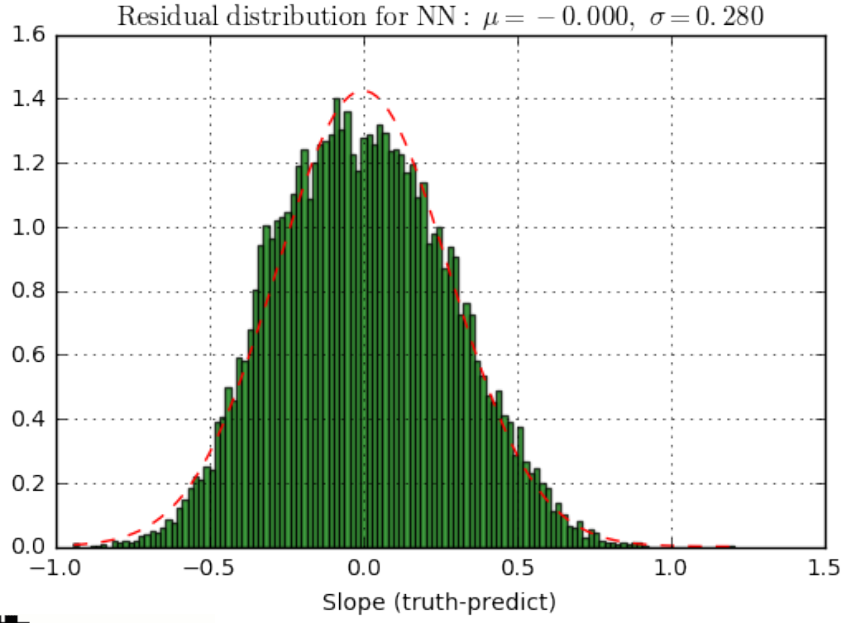


NN with Conv Layers (Multi Track Evt)



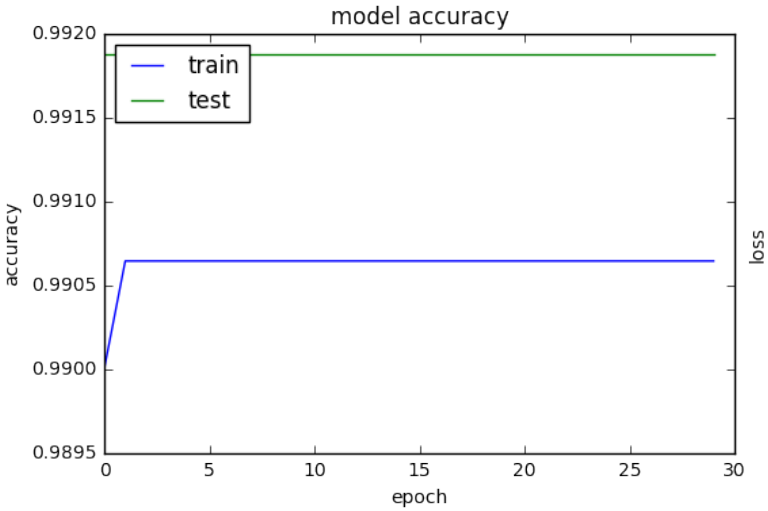


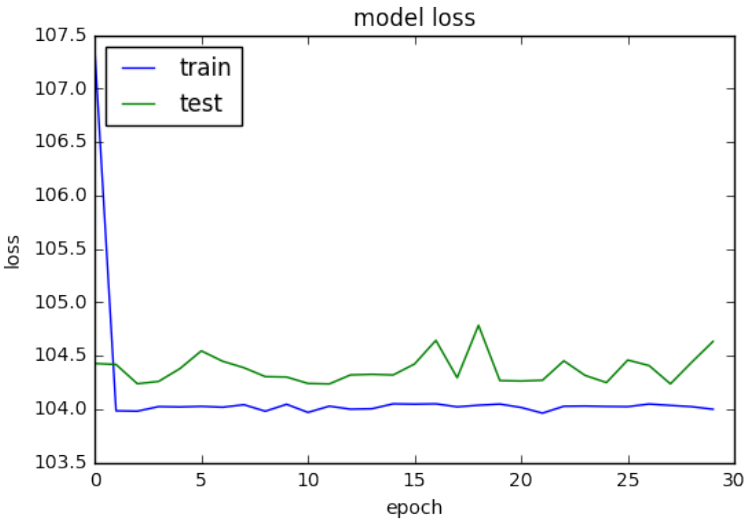




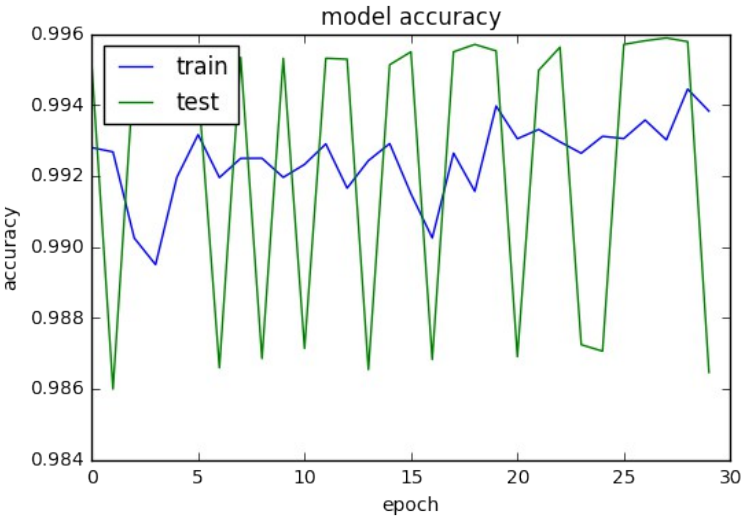
Training Performance

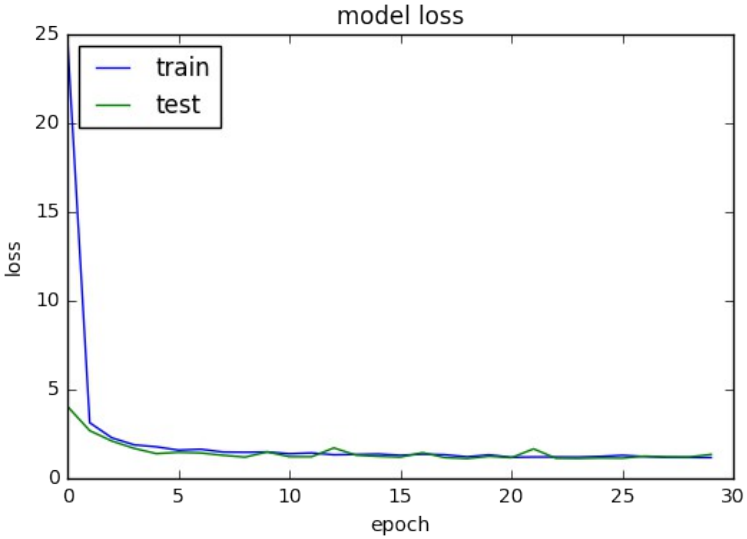
Convolutional Layers + LSTM Single Track Events:



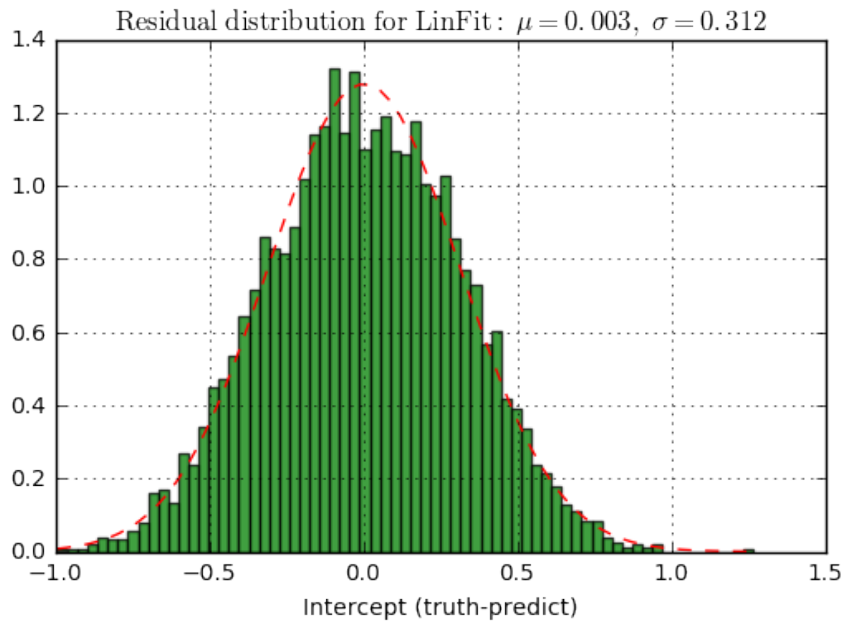


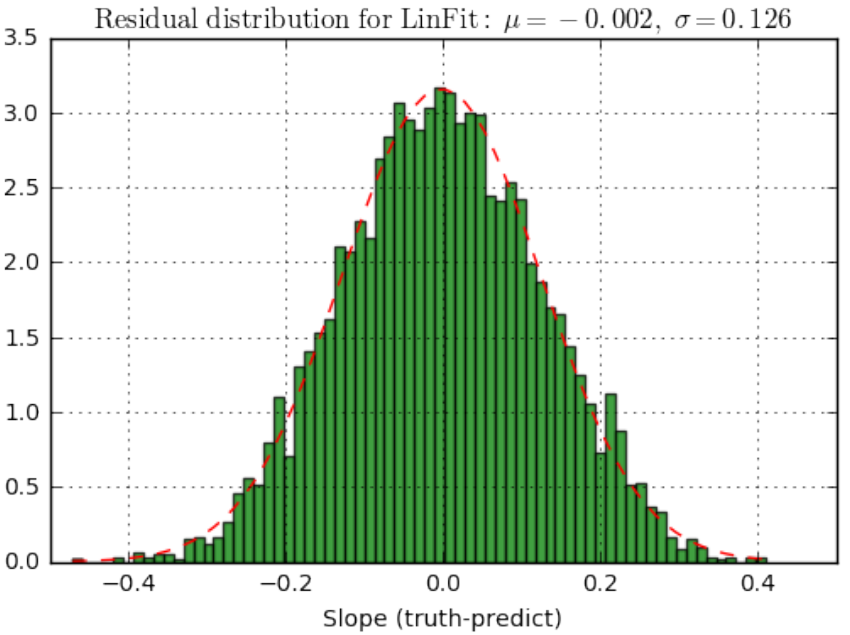
Convolutional Layers + LSTM Multi Track Events:

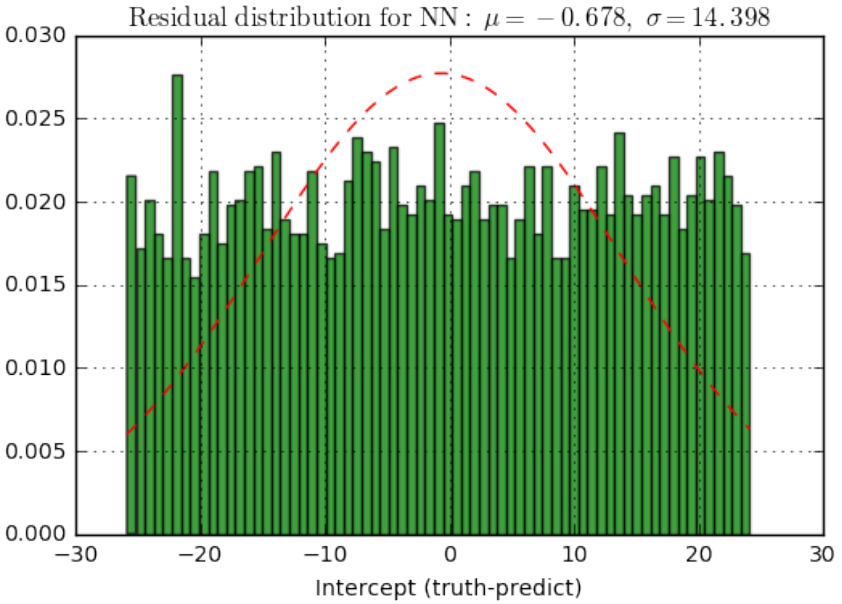


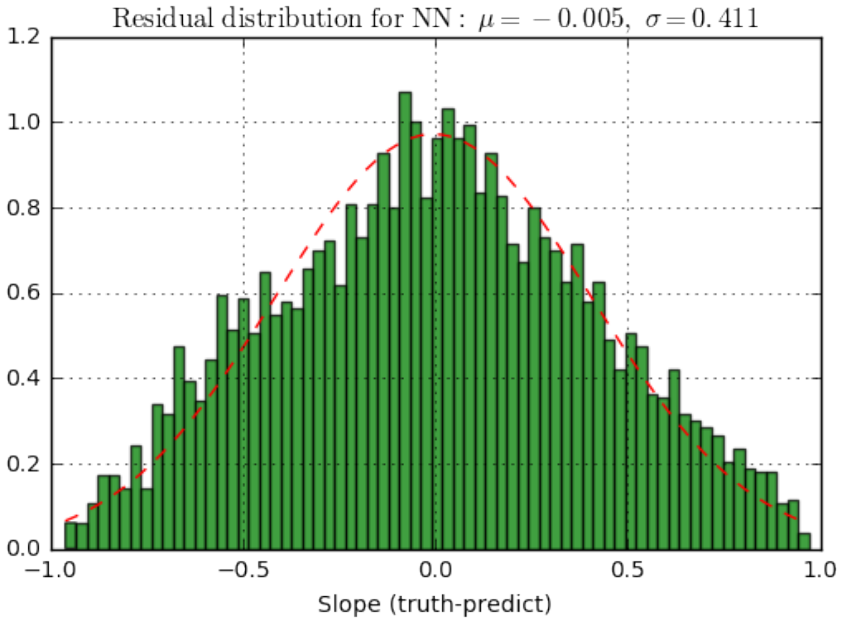


NN with Conv + LSTM Layers (Multi Track Evt)

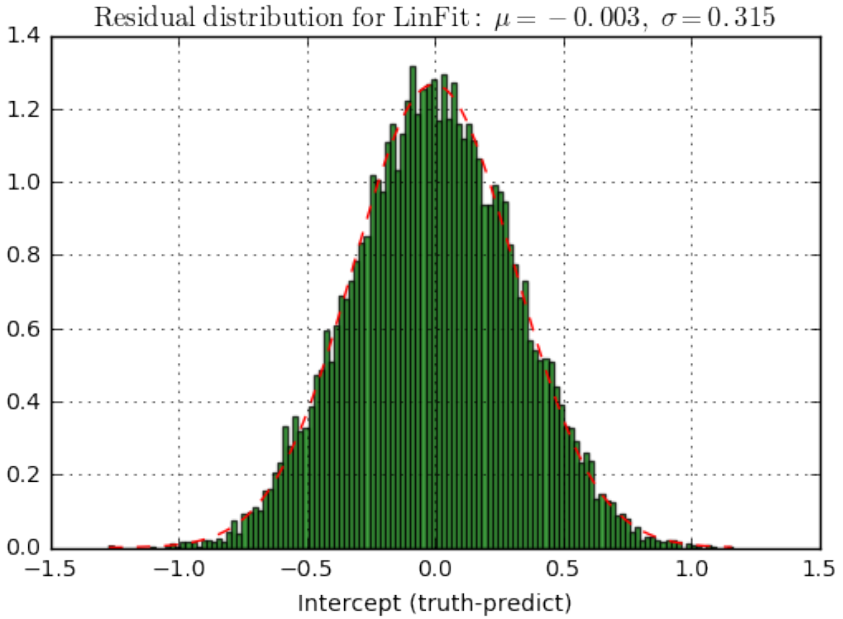


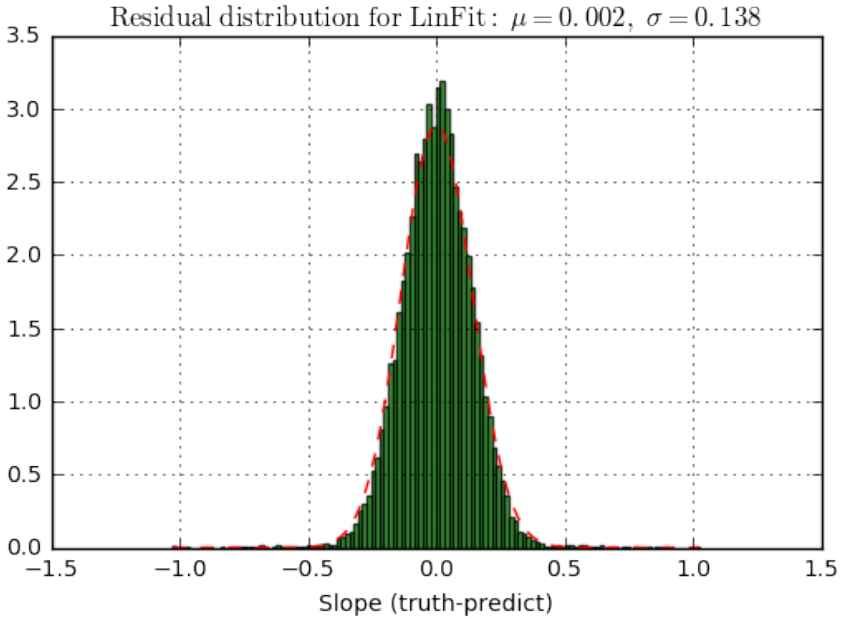


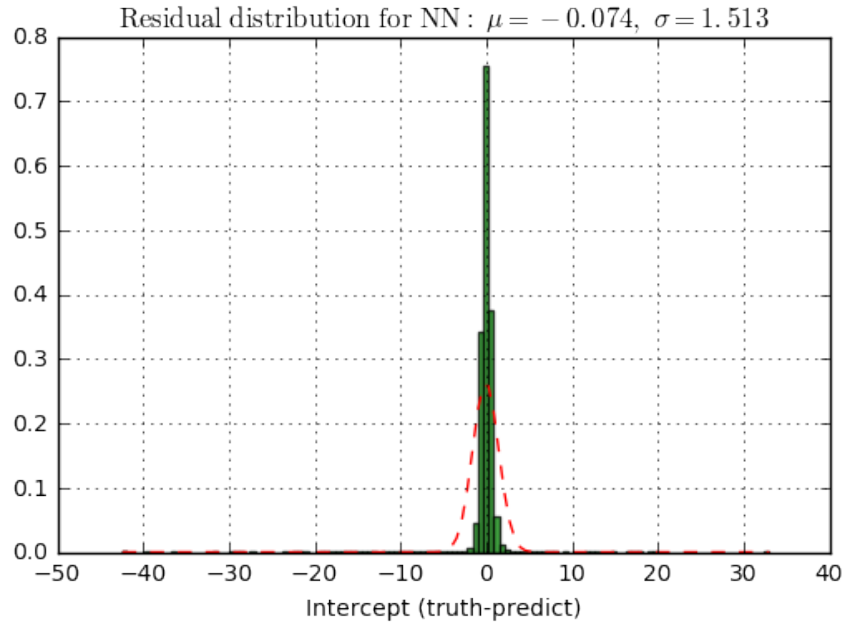


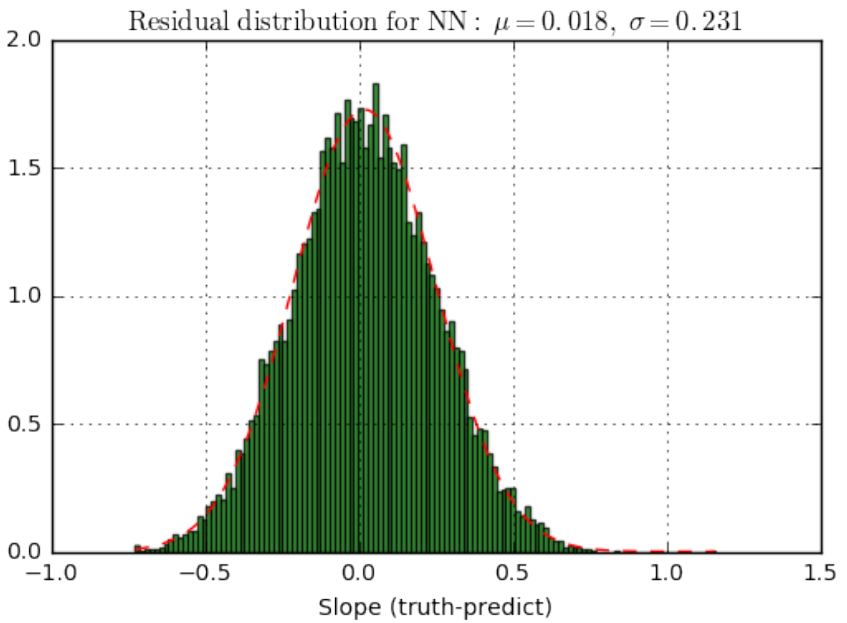


NN with Conv + LSTM Layers (Multi Track Evt)

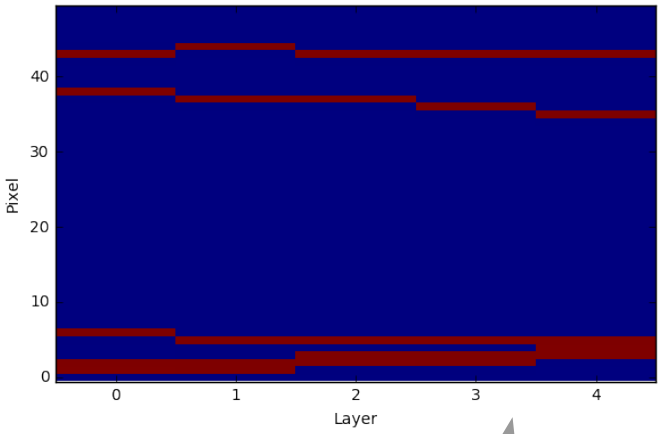


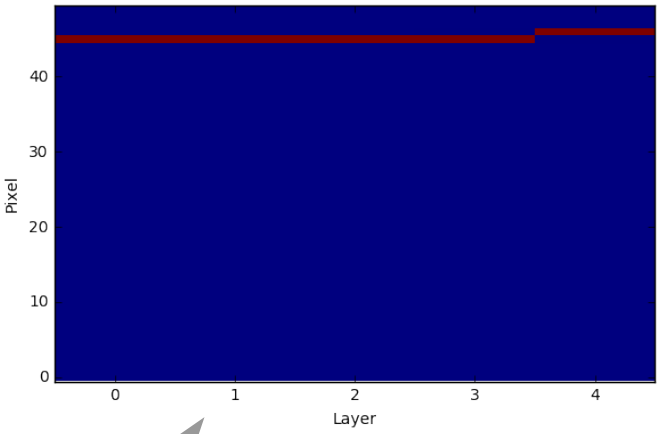


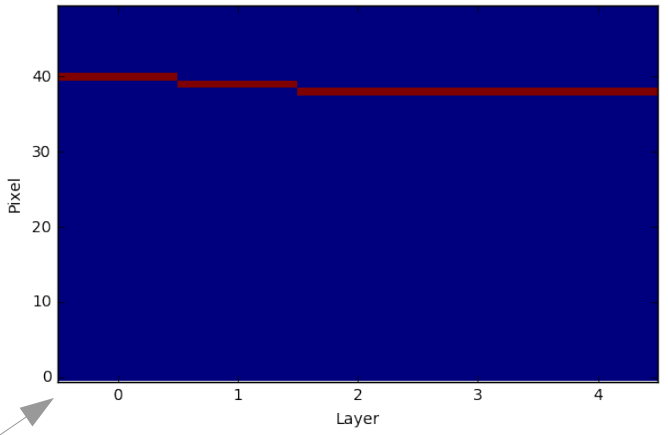


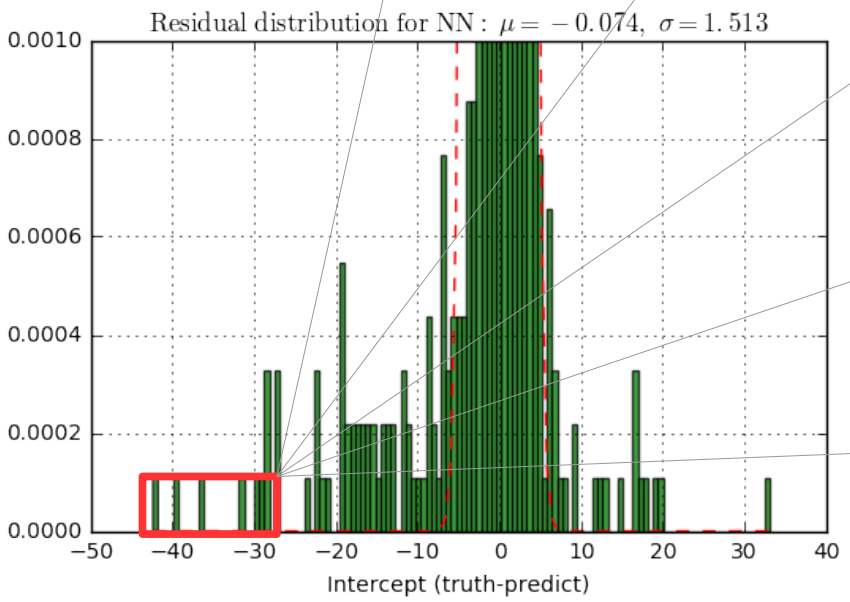


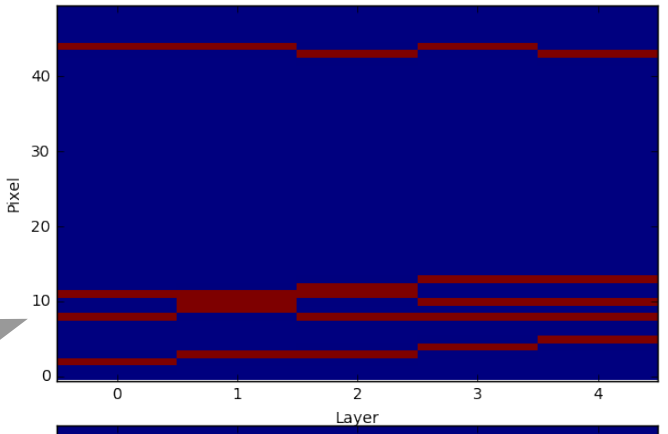
NN with Conv + LSTM Layers (Multi Track Evt)

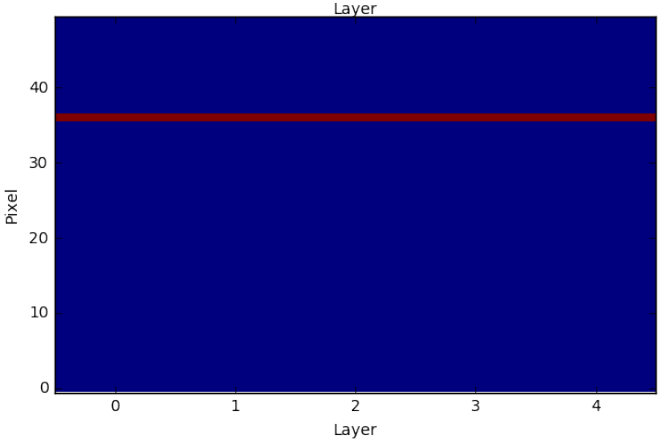




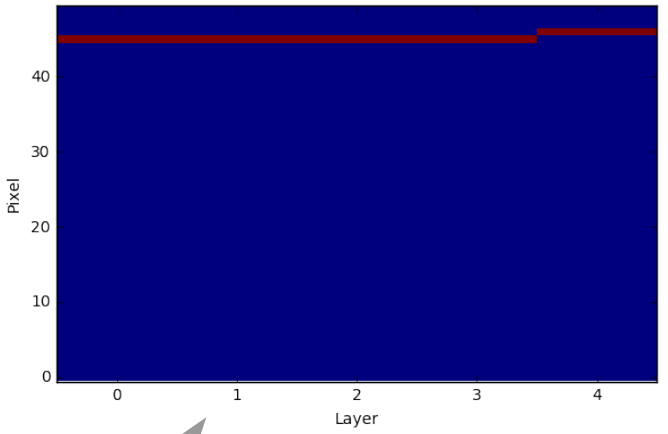


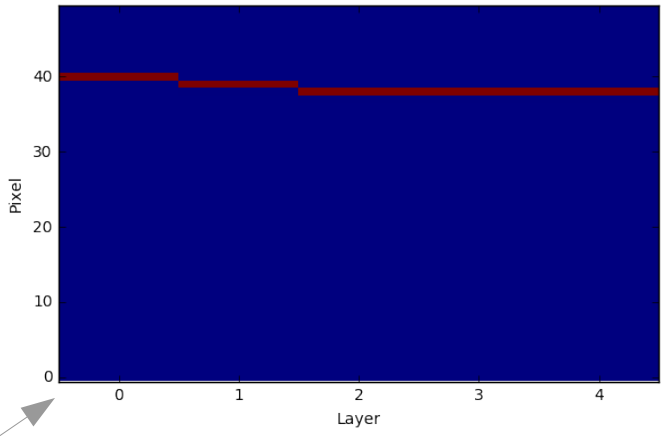




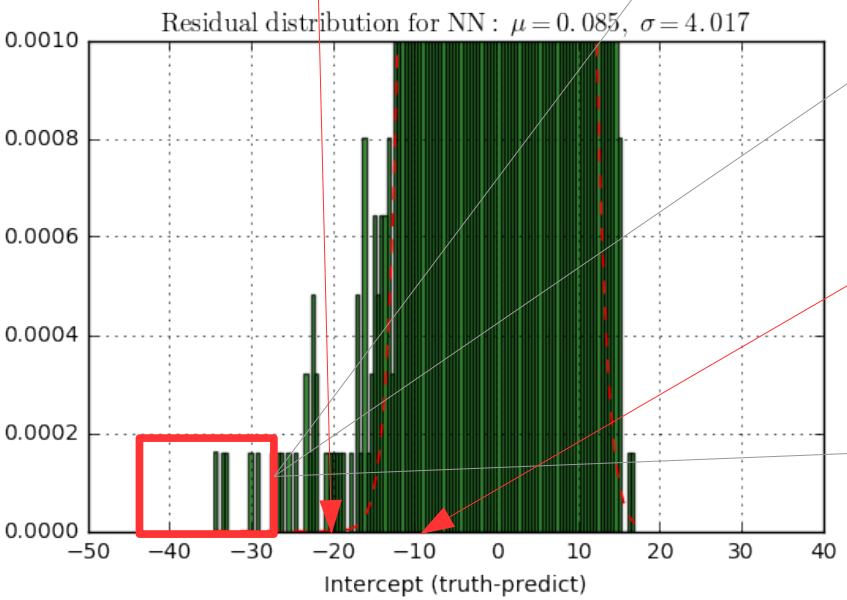


NN with Conv ~~+ LSTM Layers~~ (Multi Track Evt)

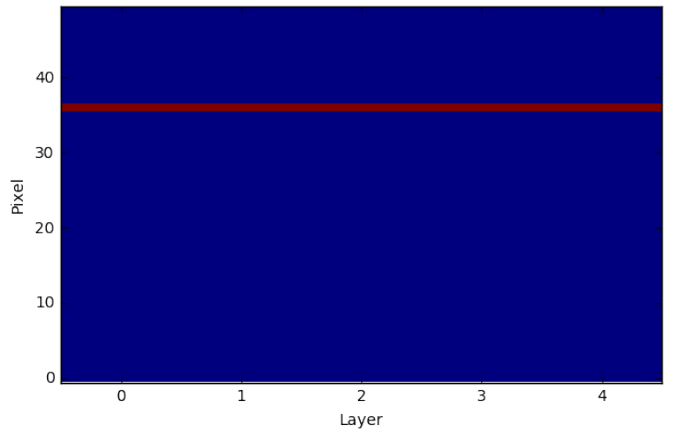




Residual ~ 20



Residual ~ 10



Summary && Future Work

1st Part

- Apart from the one track events, I do not see any obvious pattern for the number of tracks from the residuals.
- Clearly the Conv + LSTM does a much better job.

2nd Part

- Not sure why the Conv + LSTM for one track gives a flat residual for the intercept. There might be an ambiguity.
- As expected almost the same events fail for convolutional layers and convolution + LSTM. I do not see any common topology of those events (maybe 0 slope?).

Summary && Future Work

- Not a fair comparison of LinFit Vs NN for multi track events.
  - We need a tool to map the output of NN to the true hits and then do the comparison with some criteria.
- Also, with the LinFit we test the Fit of track from NN but not efficiency of finding tracks.
  - Most likely not a simple way to compare it with a classic approach for the toy model. Suggestions?
- We can feed the result to a classic algorithm to continue the tracking and realistic test the performance.
- Move to a more realistic 3D detector to add more than one modules or to the ACTS data.