Updates on discrete single-track finding models

Steve Farrell

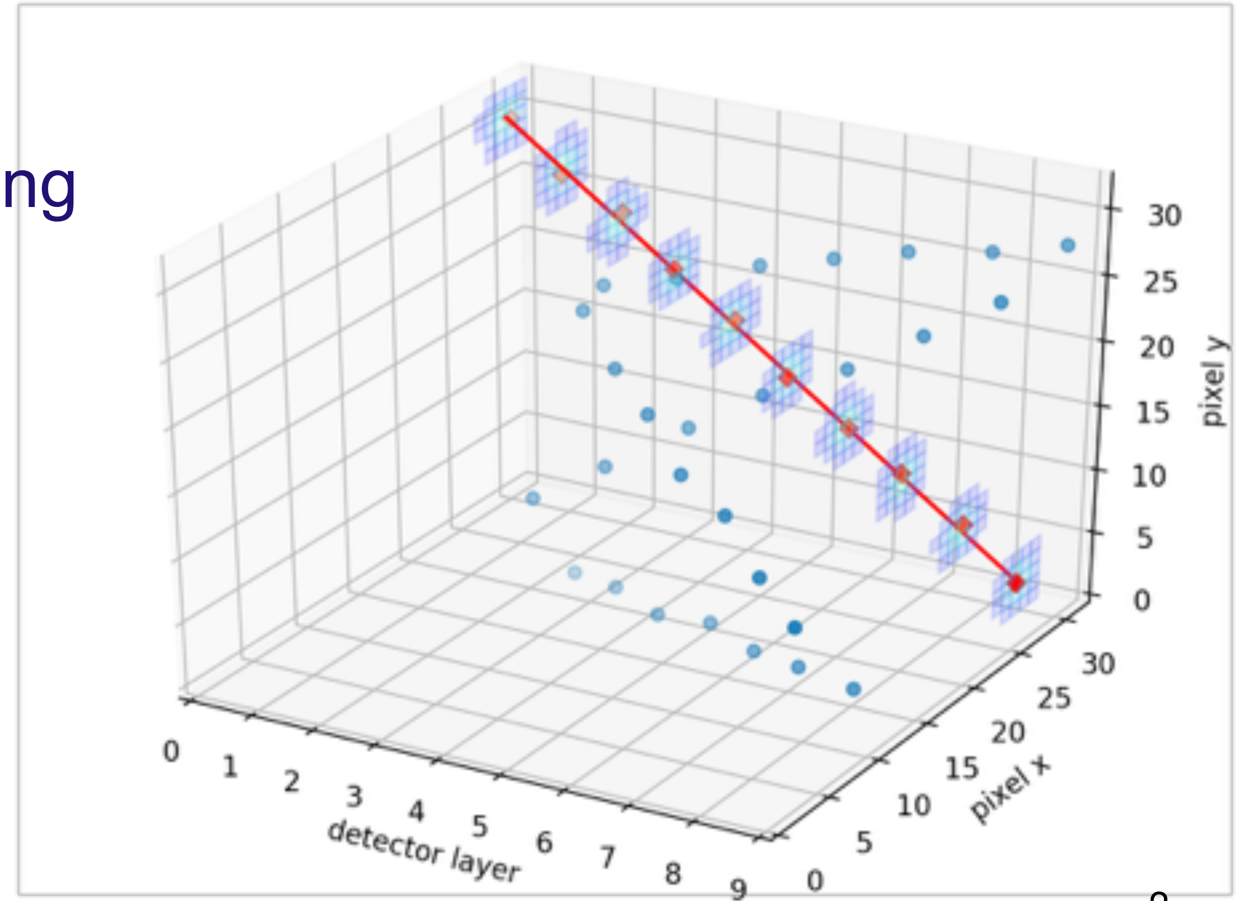
HEP.TrkX

Ernest Orlando Lawrence Berkeley National Laboratory

Feb 13, 2017

Technical updates and issues

- Can't currently run models on our GPU box. Seems like I need to rebuild nvidia drivers
- I've been running experiments on Cori Phase 1
  - generally pretty convenient when you want to run many different experiments at once
  - but there's something wrong with the 3D convolutions
    - runs much slower than my macbook…
    - under investigation
- I have cool new 3D plots, useful for showing at upcoming meetings and workshops



Not a real model output

Experiments

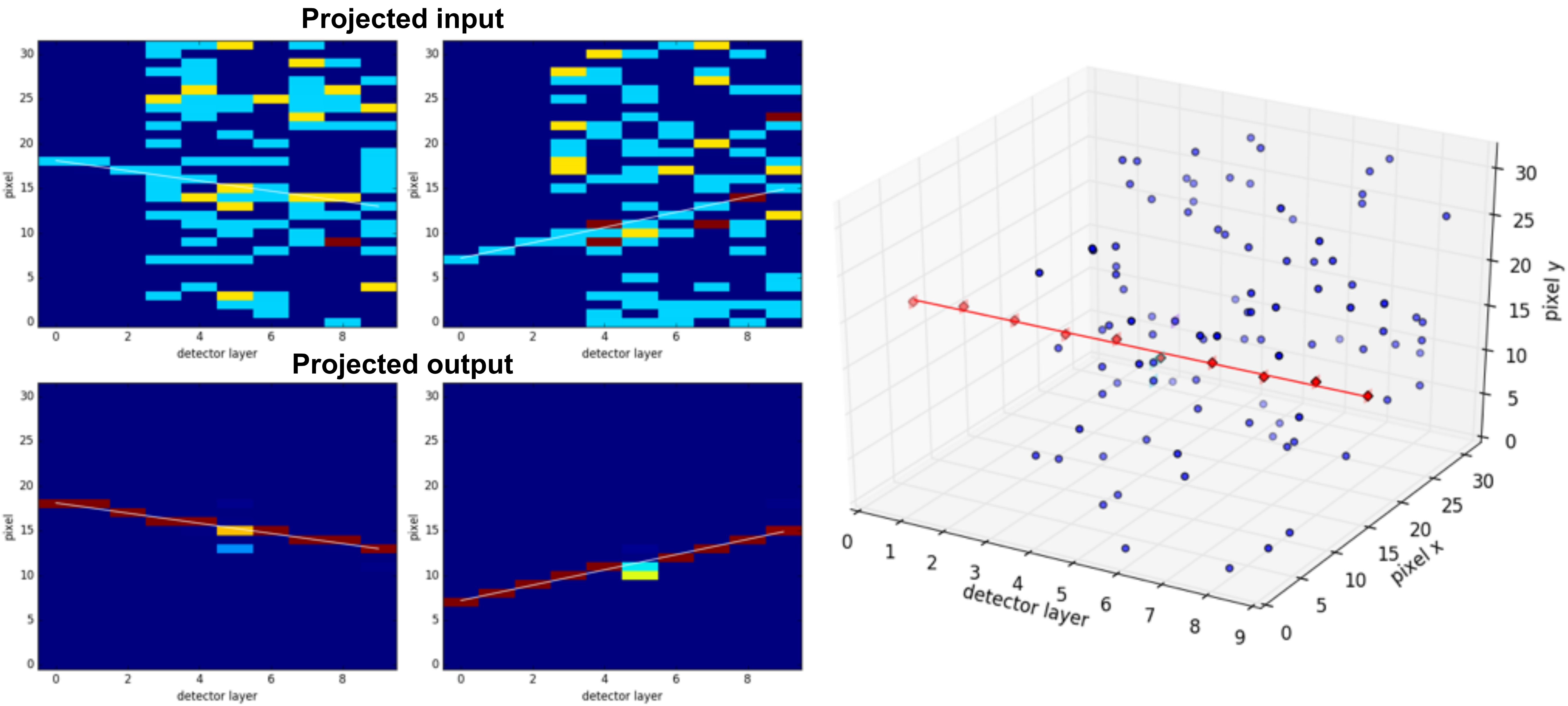
- Getting a little more serious about comparisons between models
  - Training on a large sample (~5M) of events, fixed 100k test set
    - 10 detector planes with 32x32 pixels
    - Number of bkg tracks ~ Poisson(mu=3)
    - Uniform noise hit probability = 0.01
  - Varying size of hidden dimensions from 128 to 1024
- Comparing all of the models I’ve tried so far:
  - LSTM shallow and deep architectures for *next-layer* predictions
  - LSTM shallow and deep architectures for *current-layer* predictions
  - Bidirectional-LSTM model for current-layer predictions
  - Conv3D model for current-layer predictions

LSTM predictions

architecture	hidden dim	Pixel accuracy	Hit accuracy
shallow	128	0.861	0.913
shallow	256	0.898	0.932
shallow	512	0.919	0.946
shallow	1024	0.940	0.960
deep	128	0.711	0.913
deep	256	0.811	0.916
deep	512	0.810	0.919
deep	1024	0.855	0.934
bidirectional	128	0.886	0.932
bidirectional	256	0.921	0.952
bidirectional	512	0.937	0.961
bidirectional	1024	0.948	0.968
next-layer	256	0.370	0.977
next-layer	512	0.43	0.982
next-layer	1024	0.471	0.985

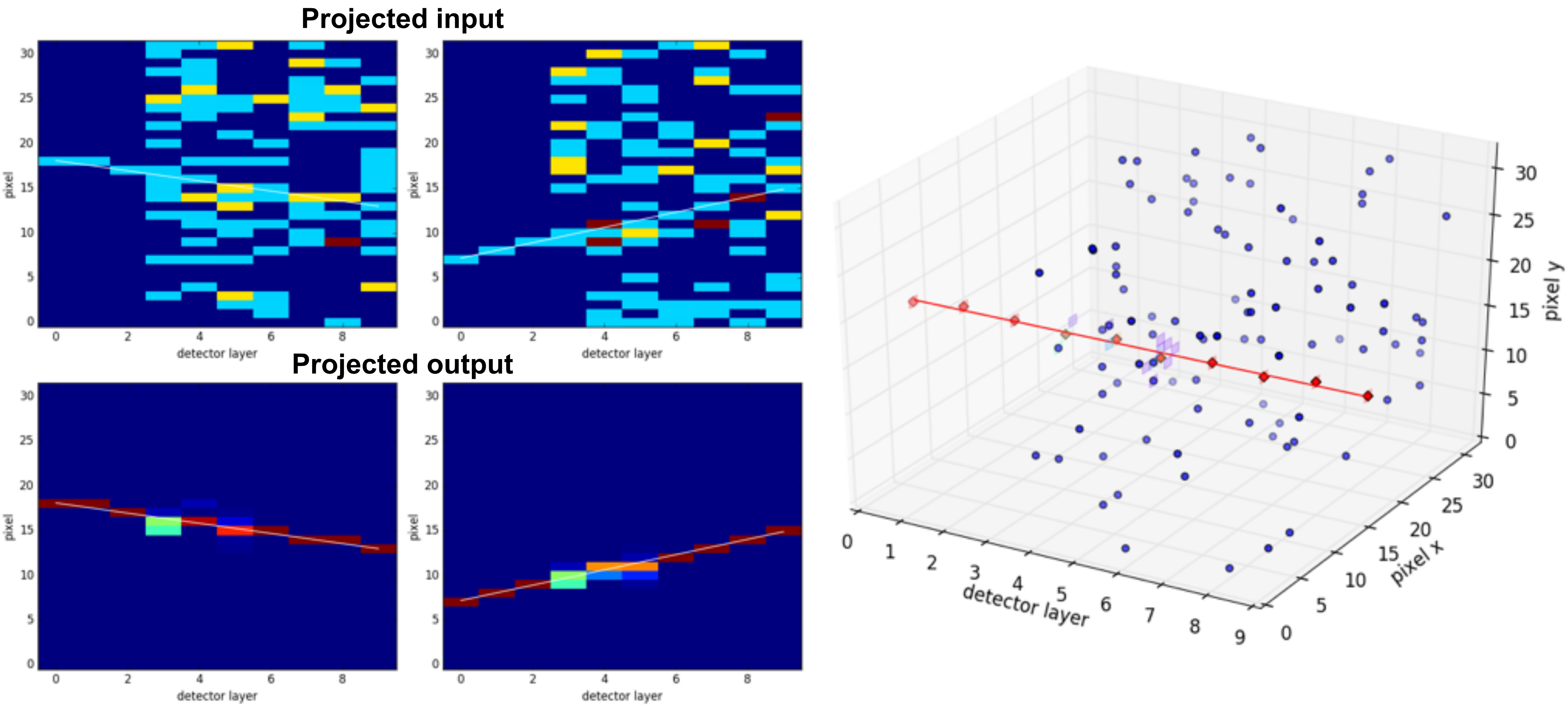
- Shallow models out-perform deep models on fixed size training set
  - they have more parameters in the LSTM, though
  - deep model is maybe harder to train
- Bi-directional model maybe better than single LSTM
  - but they have more parameters
- Larger hidden dim is better
  - but the models are already quite large
- Next-layer predictions give the best hit classifications
  - more stable?

Example current-layer LSTM prediction



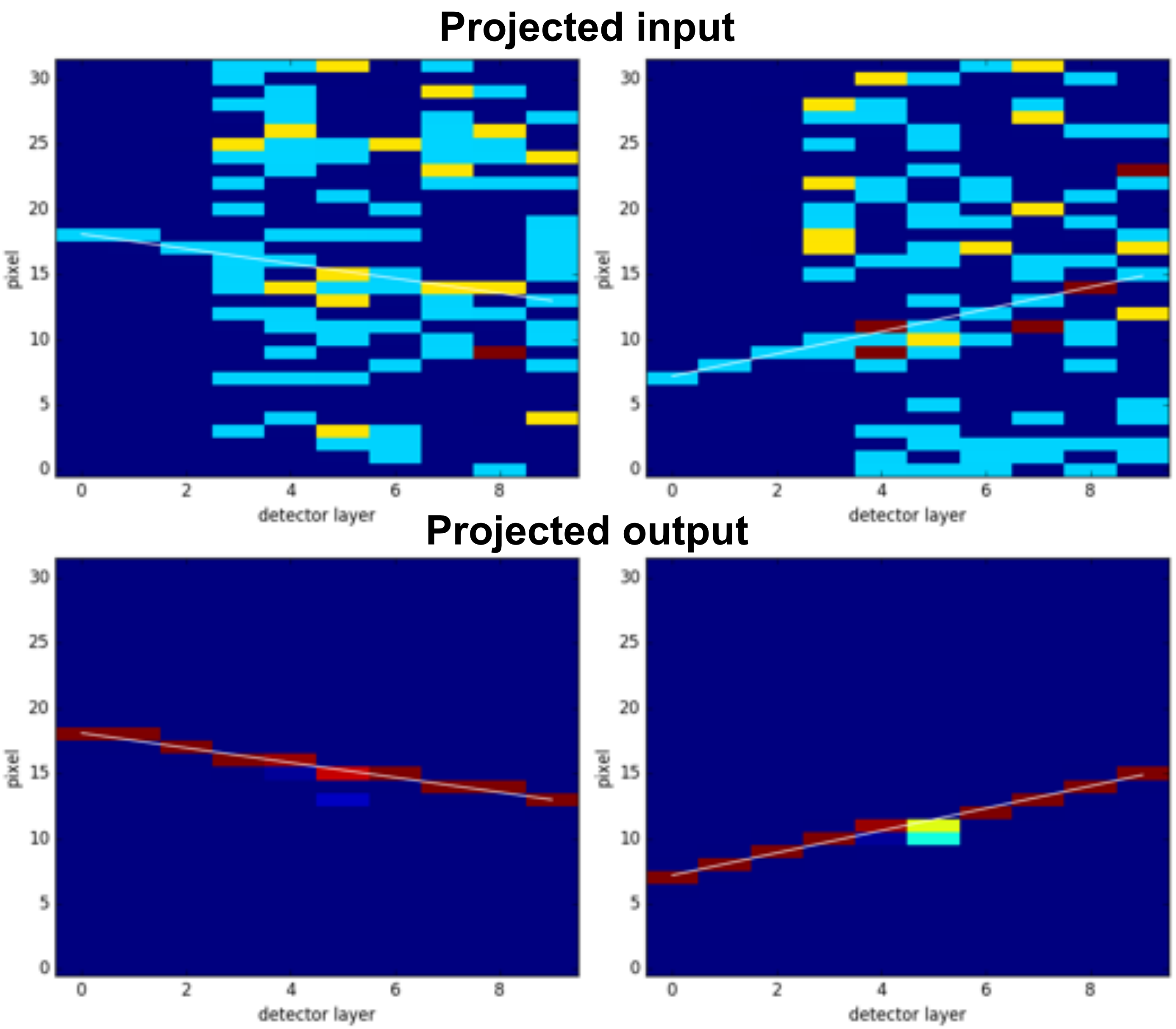
- Fairly precise predictions, though there are some discontinuous artifacts that can result in classification errors on a detector plane
  - Seems to be a general feature of current-layer prediction models without some other form of training stabilization

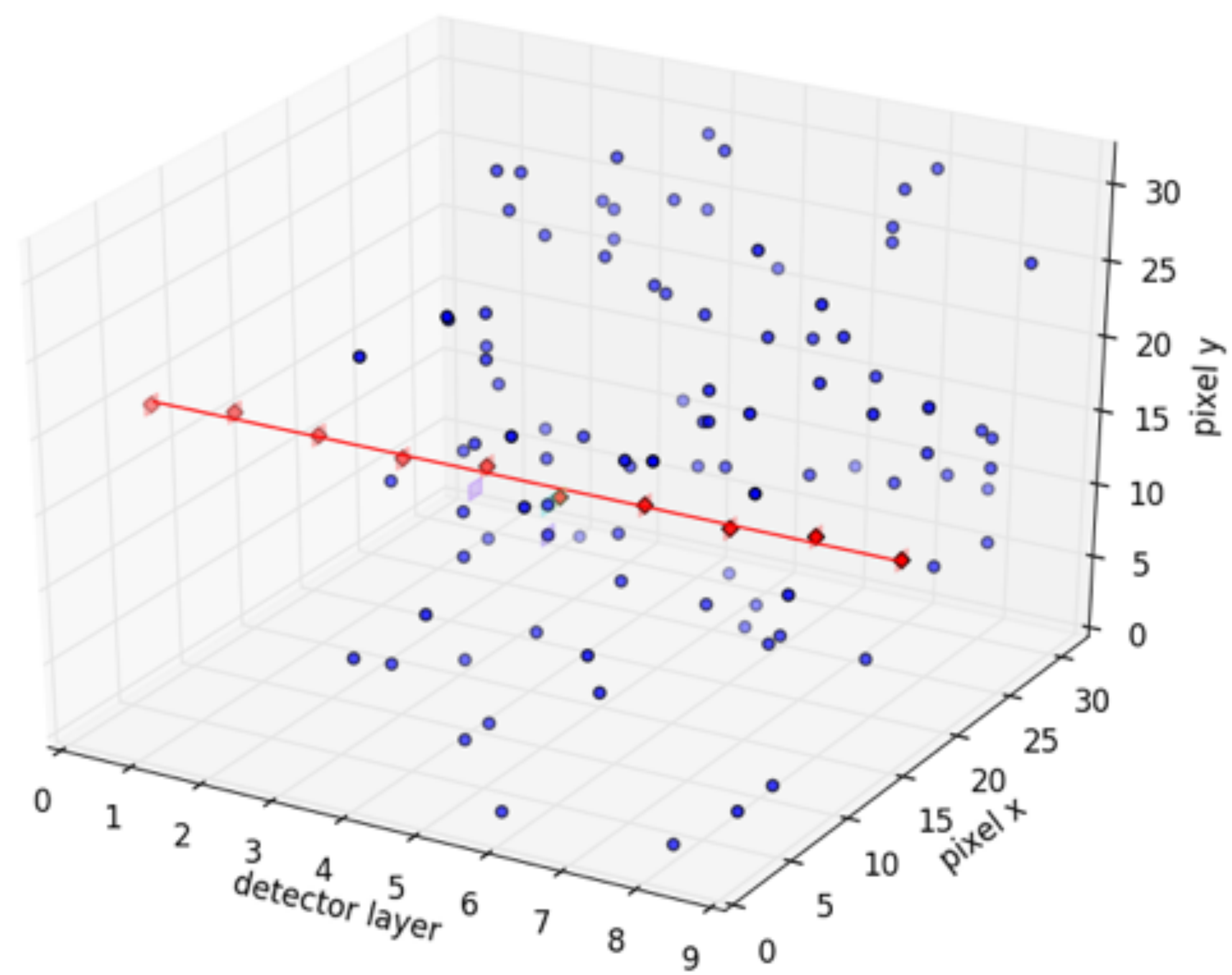
Example current-layer deep LSTM prediction



- Seems to be more susceptible to these artifacts, more easily confused by nearby hits on the detector planes
- This architecture might do better in other cases
  - non-straight tracks and/or higher occupancy
  - multi-target predictions (e.g. track parameters) to stabilize training

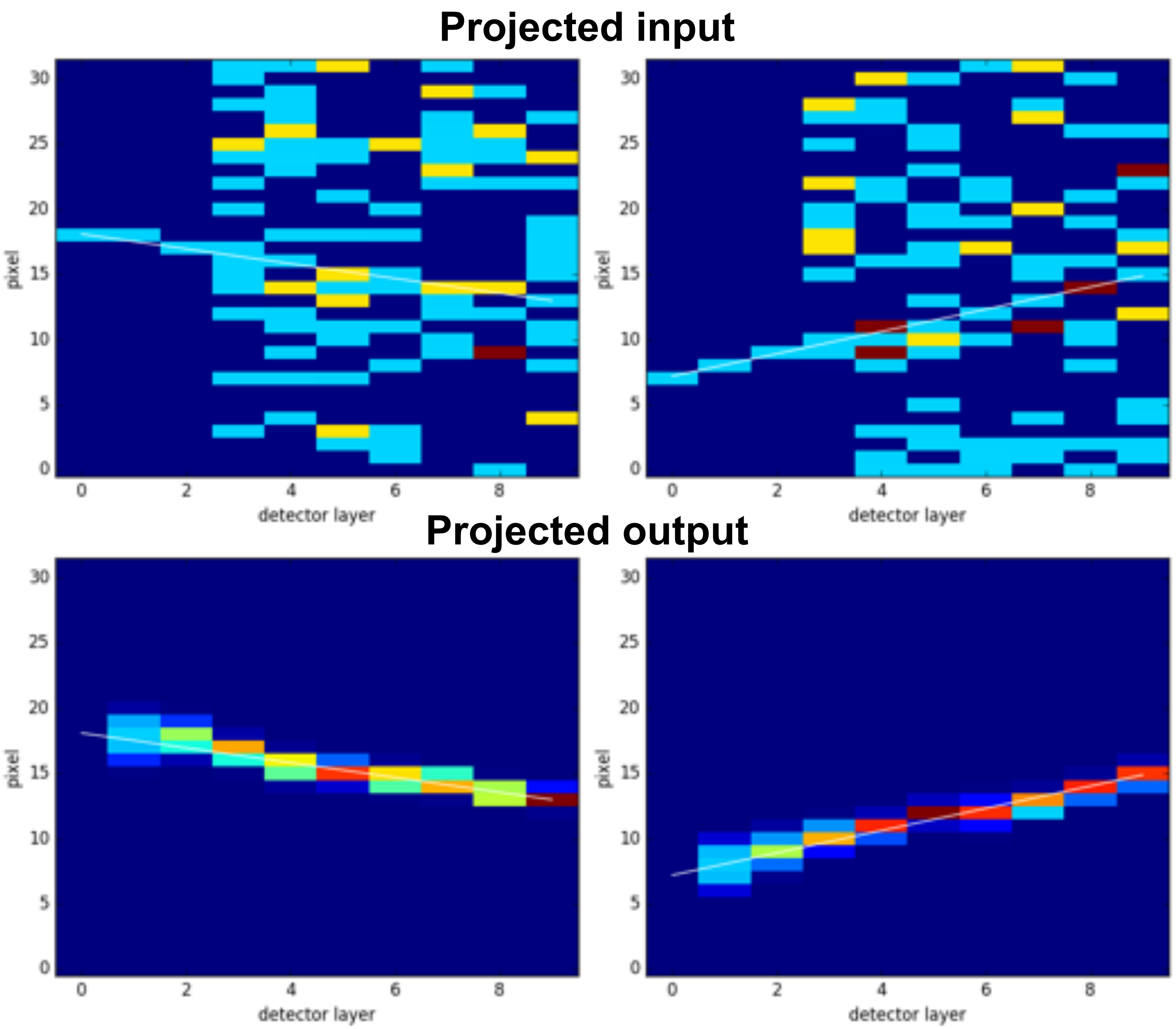
Example current-layer bi-LSTM prediction

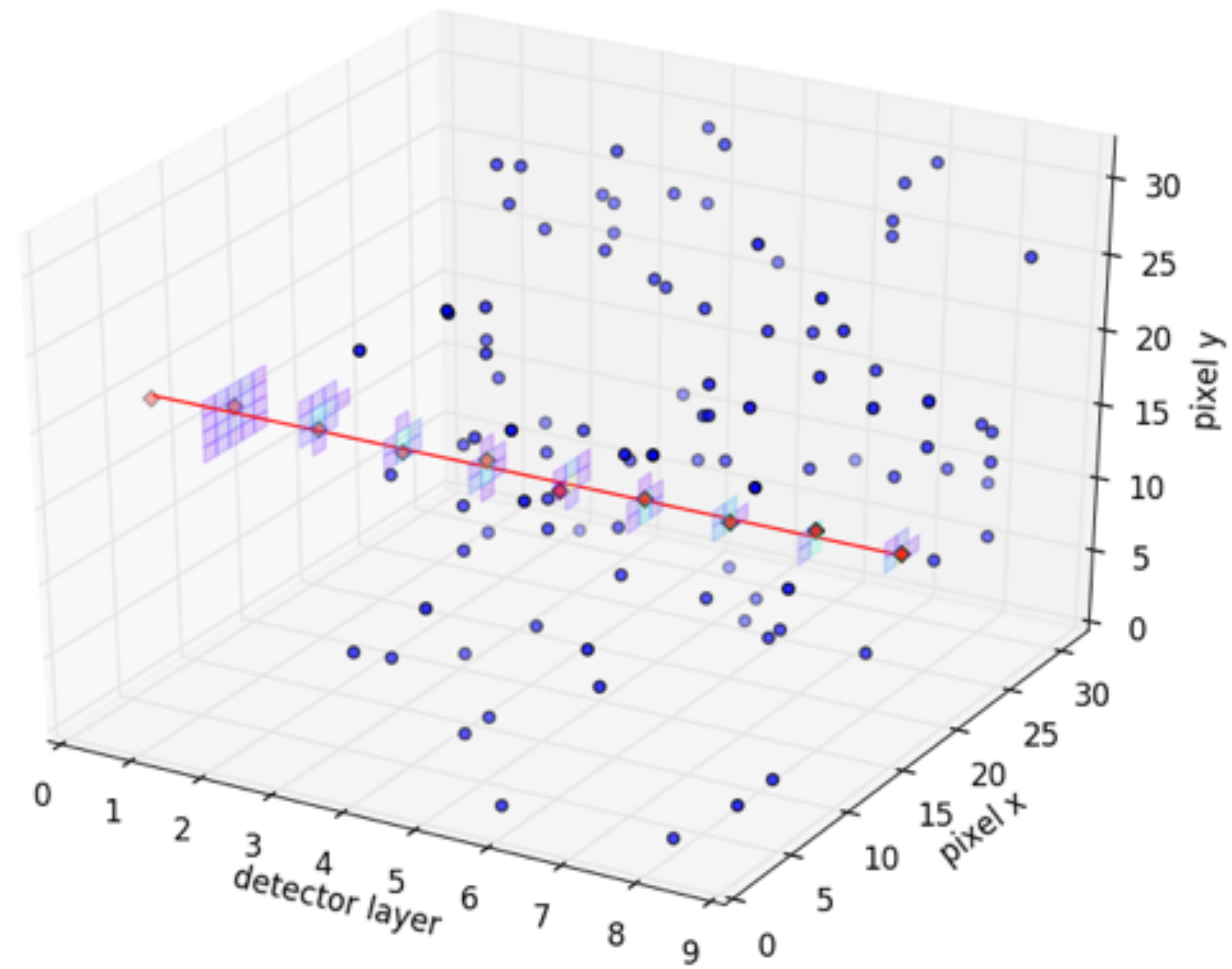




- Very precise predictions
  - can see into the future, which presumably helps
- still has rare artifacts

Example next-layer LSTM prediction





- Next-layer model gives predictions that are less precise but smoother and more accurate
    - Mostly unaffected by nearby stray hits
- With this detector occupancy, they are the best at classifying hits
    - but this may change with higher occupancy

Next steps

- Fold in the track parameter estimations (à la Dustin) to quantify stabilization effect
- Compare the Conv3D models in the same way
- Adjust the parameters of the data generation and see how each model performs
  - Drop the noise, increase number of background tracks
- Figure out a nice way to visualize what the Conv3D models are learning
  - Drawing the filters, saliency maps, etc.
  - Might be too messy in 3D, but we could use 2D detectors just for this kind of study.
- Take ACTS data, discretize the hits (or use the “digitized” hits), and try the models there

- Prepare slides for Connecting the Dots