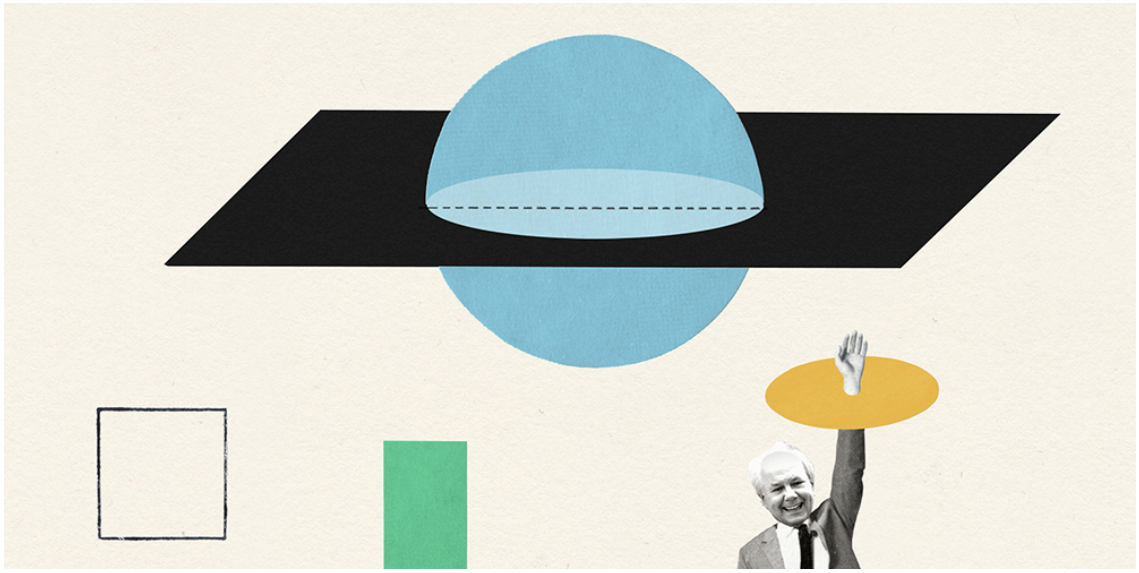


Beyond 2D representations: track/shower separation in 3D

Ji Won Park

Kazu Terao

11/14/17

SLAC National Accelerator Laboratory

INTRODUCTION

Motivations and goals

Long-term mission: build a full 3D reconstruction chain for LArTPC data using deep learning.

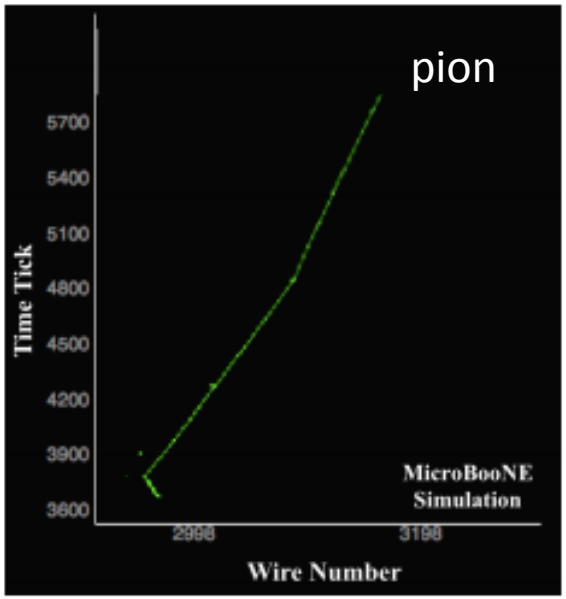
Why in 3D?

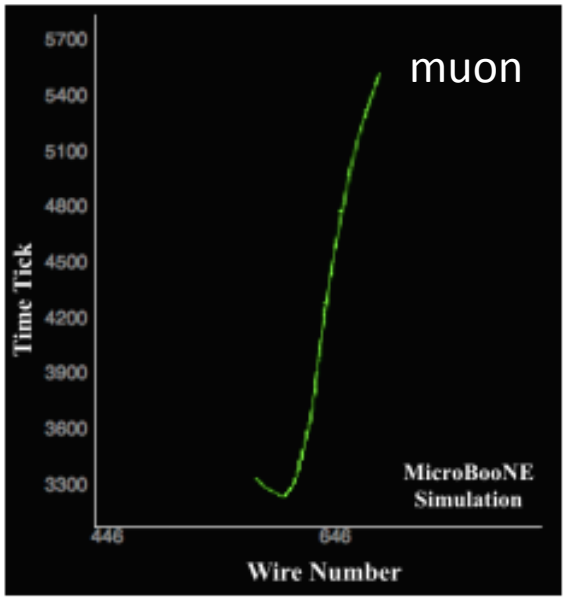
- Less optical illusion in interpreting 3D data.
- PID in 2D and track/shower separation in 2D have been done for MicroBooNE data. Pattern recognition in 3D is a natural extension from 2D.

Goal for the next ~18 minutes: report the results of training a semantic segmentation network to perform track/shower separation on 3D simulation data, as a working test case for pattern recognition in 3D.

Image classification task in 2D

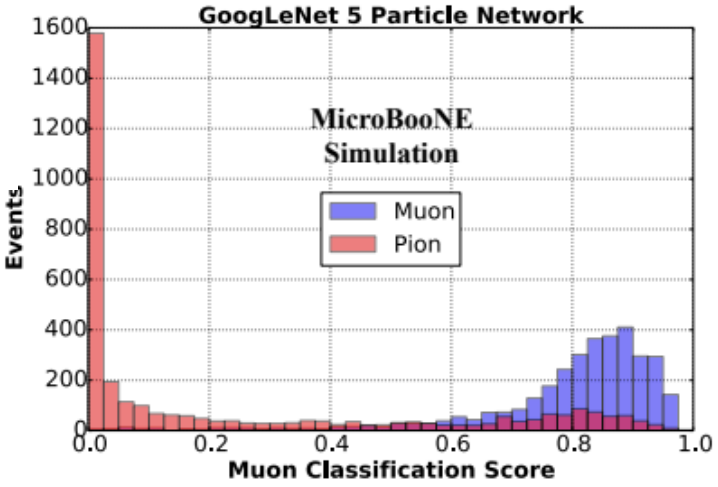
Five-particle PID has been done: given a 2D image of a single particle, label it as a gamma ray, electron, muon, pion, or proton.





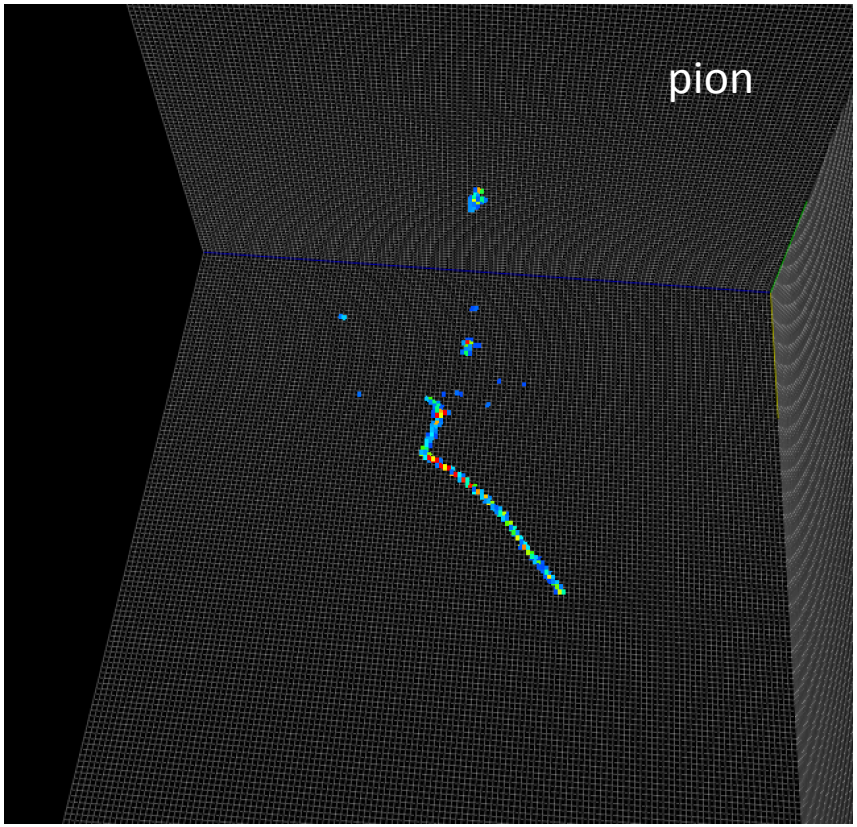
gave a happy score distribution.

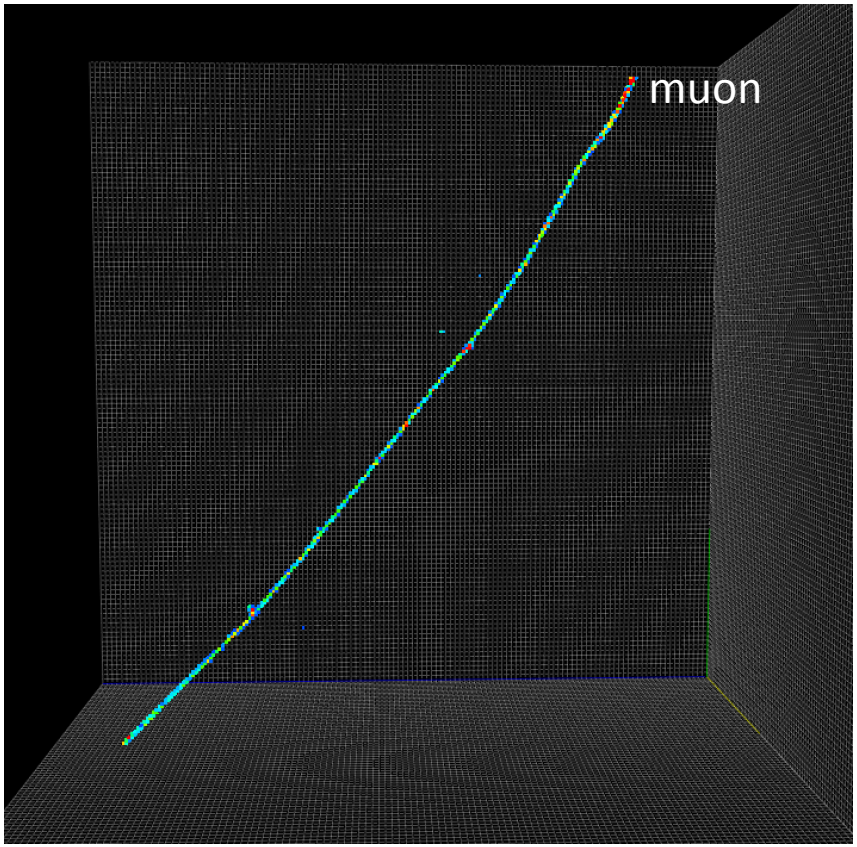
*Muon classification score ~ high likely the algorithm thinks an image is a muon. Assigned high scores for muons vs. low scores for pions → confidence in prediction ☺



From the MicroBooNE CNN paper (2016)

5-particle PID in 3D is a natural extension
(achieved similar results as 2D)





Voxel = the 3D equivalent of *pixel*

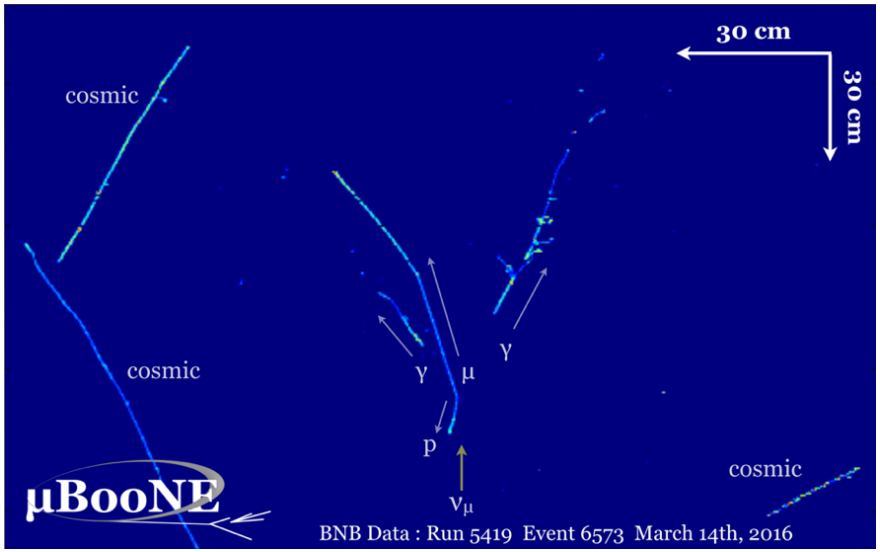
METHODS

Our study: **shower/track separation**.

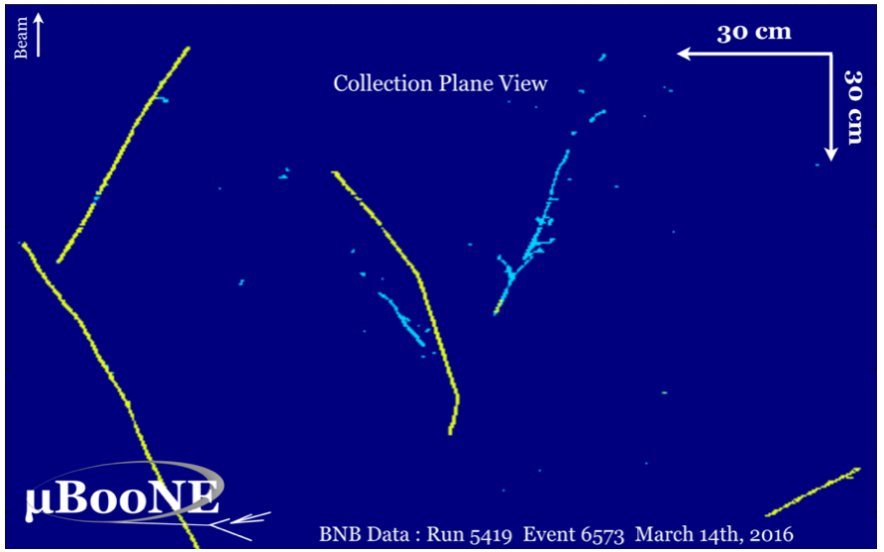
(Now 3 classes for track, shower, background instead of 5 particle classes)

It has been done pixel-level in 2D using semantic segmentation.

Truth label



Prediction



Yellow: track,
Cyan: shower

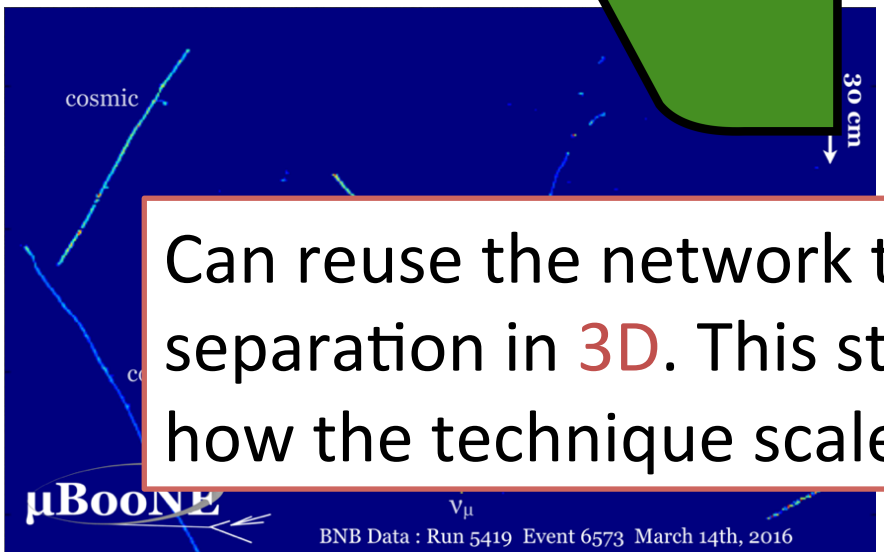
From MicroBooNE DNN paper under review

Our study: **shower/track separation.**

(Now 3 classes for track, sh[illegible]d instead of 5 particle classes)

It has been done pixel-[illegible]n 2D[illegible] semantic segmentation.

Tr[illegible]bel [illegible]tion

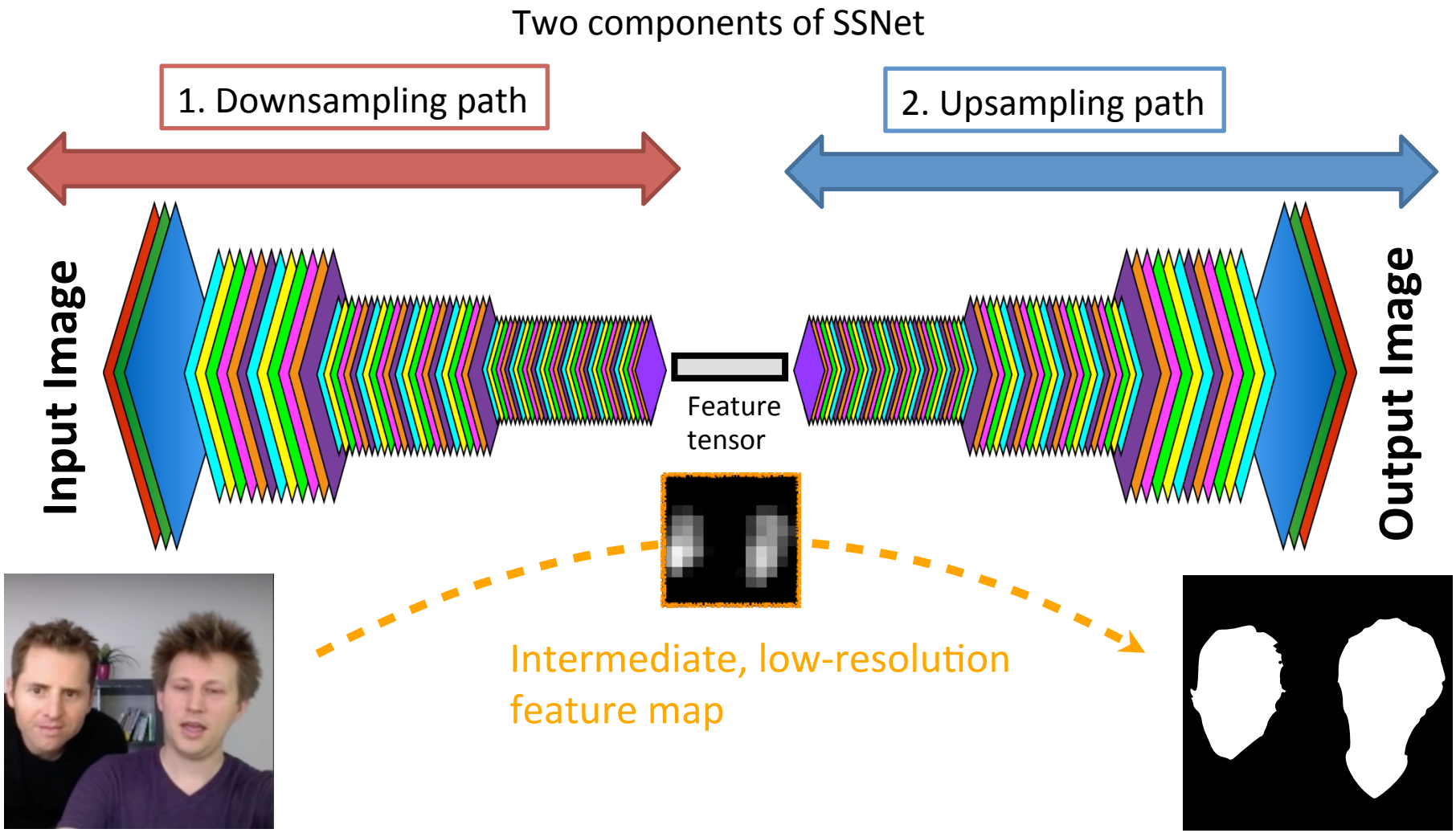


Can reuse the network to do shower/track separation in 3D. This study allows us to explore how the technique scales to 3D.

Yellow: track,
Cyan: shower

From MicroBooNE DNN paper under review

The semantic segmentation network (SSNet)

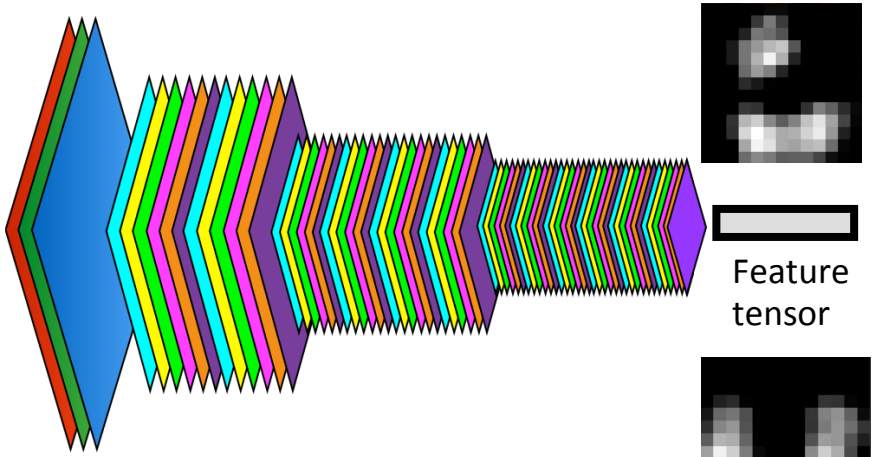


Thank you Kazu for the diagram ☺

“Written texts” input image

Feature extraction

1. Downsampling path



Feature tensor

“Written texts” feature map

“Human face” feature map

“Human face” input image

Feature extraction

Role: **classification**
A series of convolutions and downsampling which reduce the input image down to the lowest-resolution feature map.

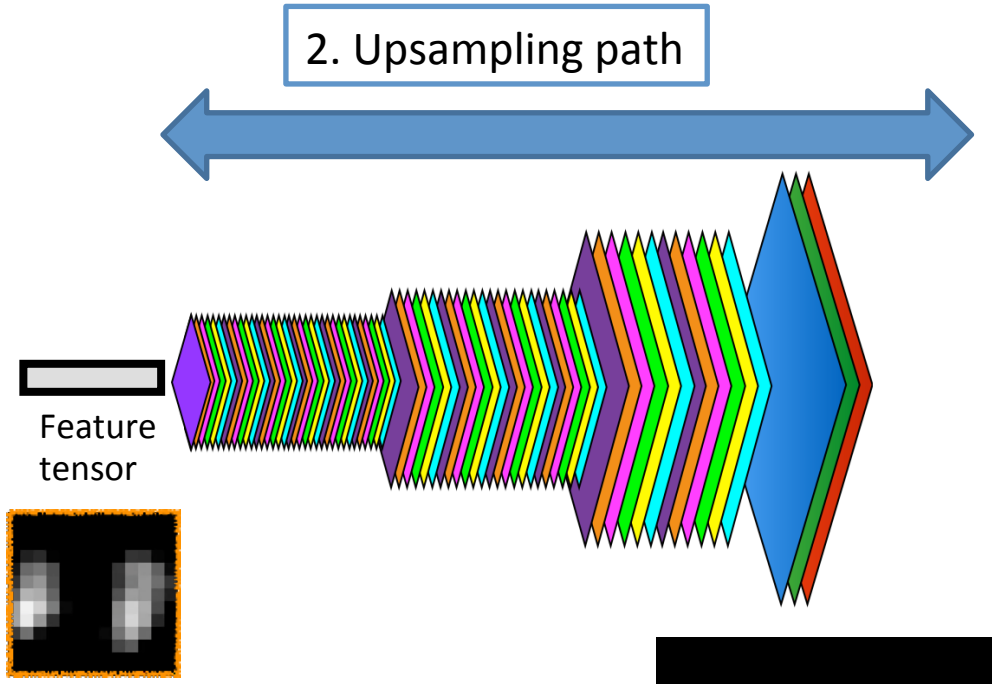
Each downsampling step increases *field of view* of the feature map and allows it to understand the relationship between neighboring pixels.

Thank you again Kazu for the diagram ☺

Role: **pixel-wise labeling**

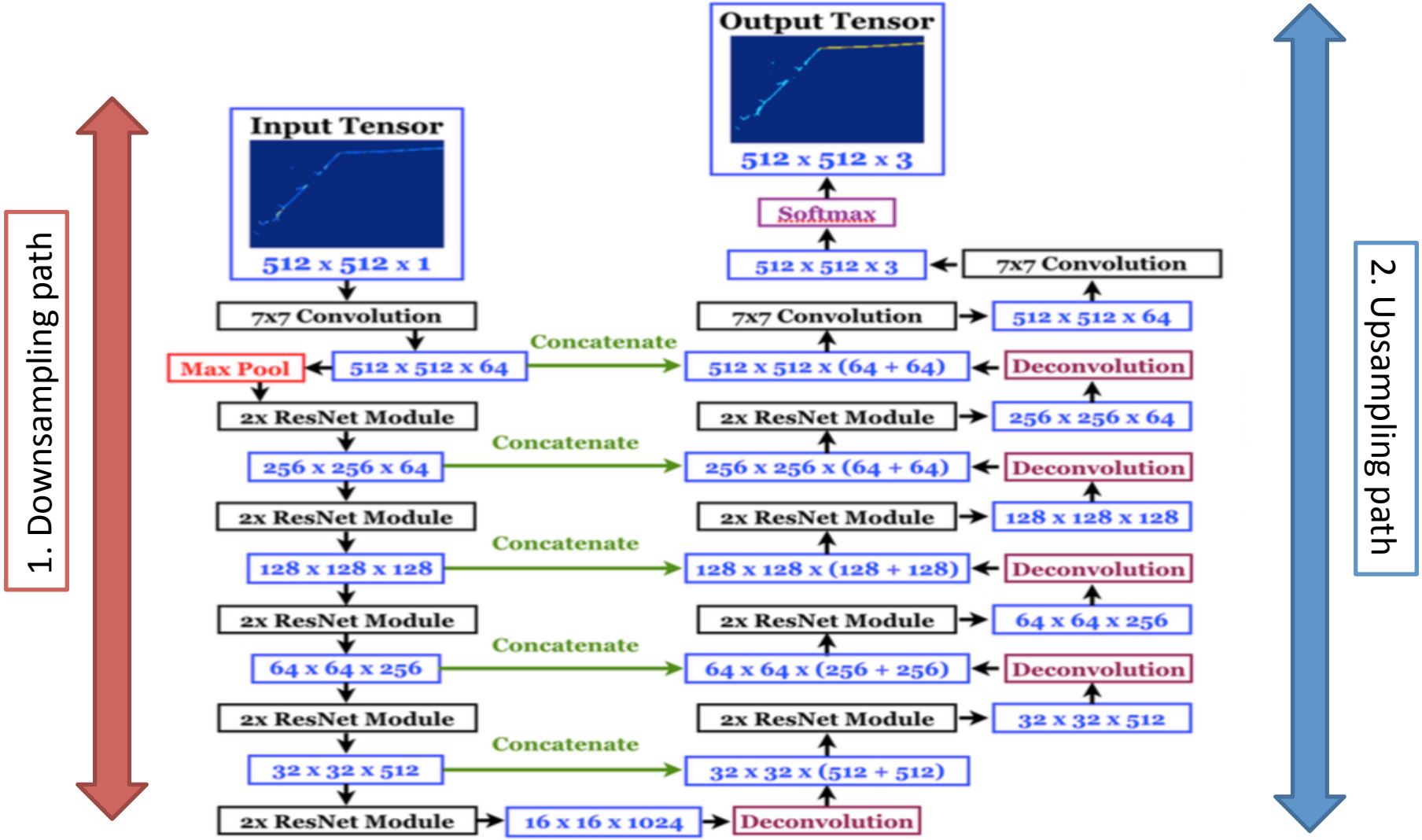
~ reverse version of downsampling path.

A series of convolutions-transpose, convolutions, and upsampling which retrieve the original resolution of the image, with each pixel labeled as one of the classes.



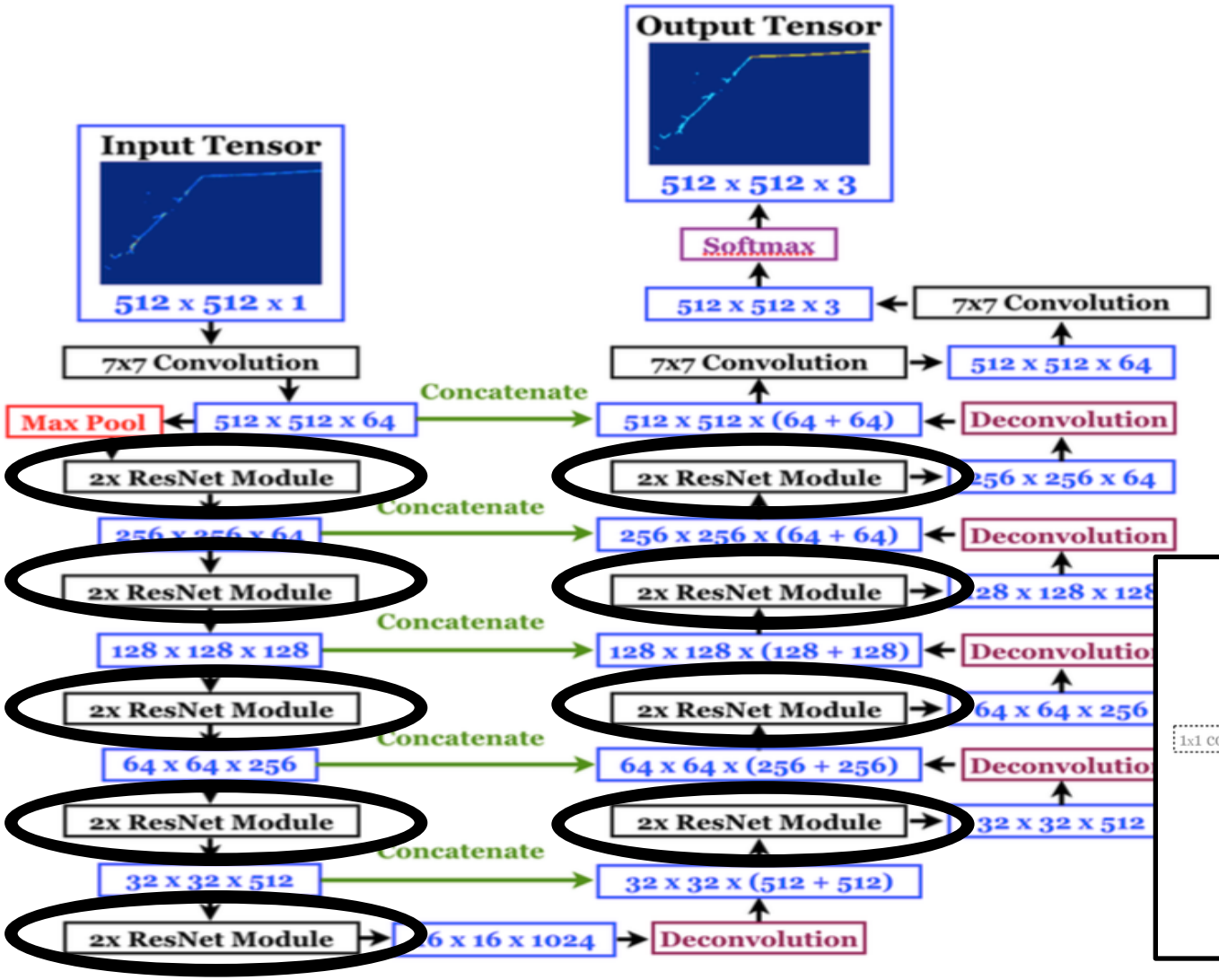
Segmented output image
Each pixel is either “human” or “background”

The type of SSNet we used: **U-ResNet**



From MicroBooNE DNN paper under review

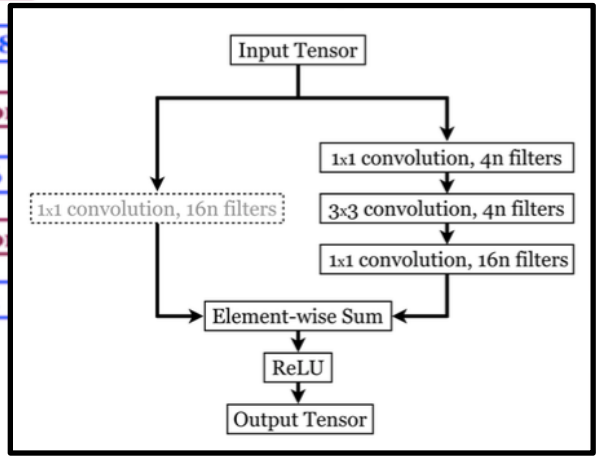
The type of SSNet we used: U-ResNet = U-Net + **ResNet**



Within the U-Net architecture, use ResNet modules.

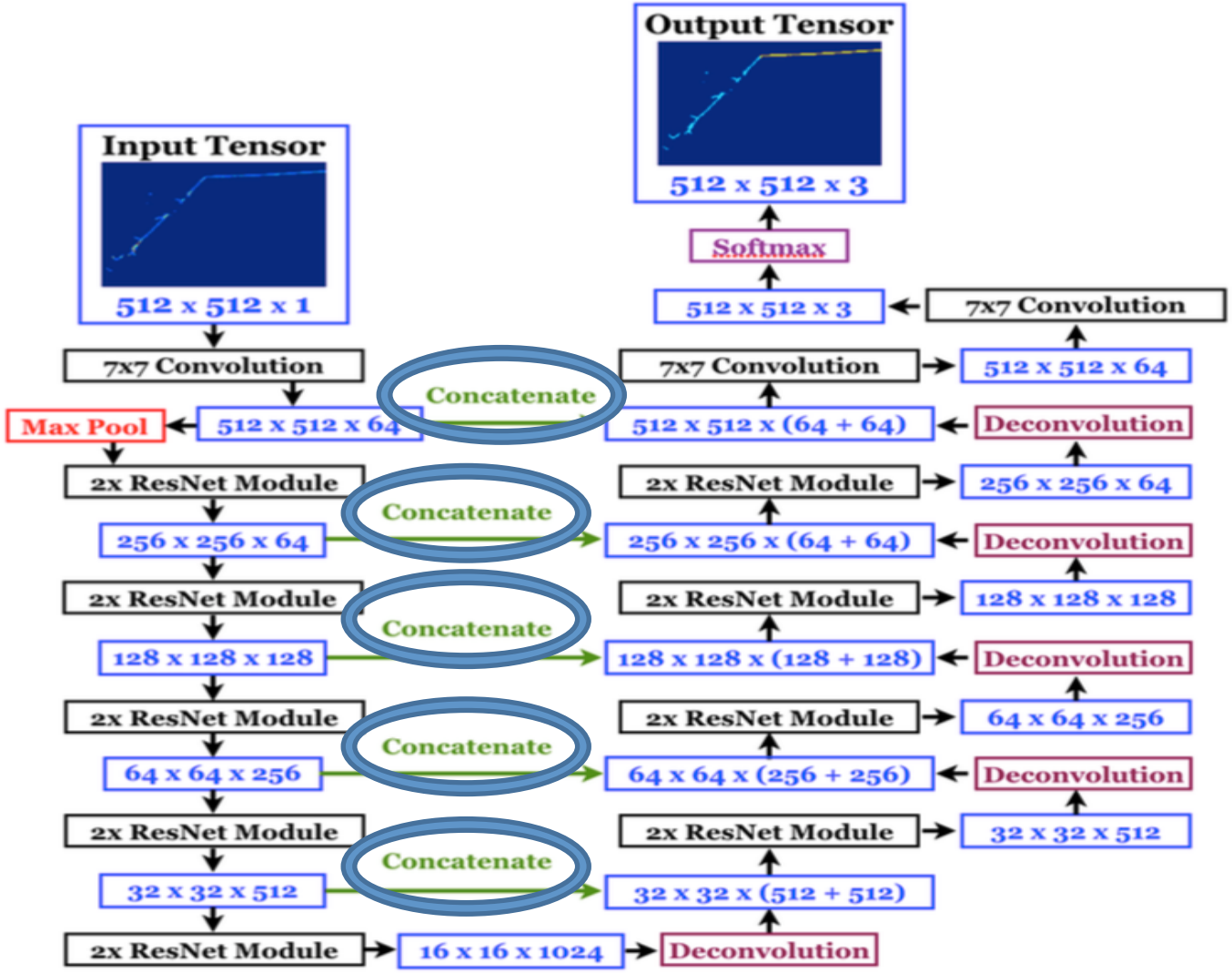
In U-ResNet, the convolutions are embedded within ResNet modules.

One ResNet module:



From MicroBooNE DNN paper under review

The type of SSNet we used: U-ResNet = **U-Net** + ResNet



Concatenations: a feature of U-Net.

We stack the feature maps at each downsampling stage with same-size feature maps at the upsampling stage.

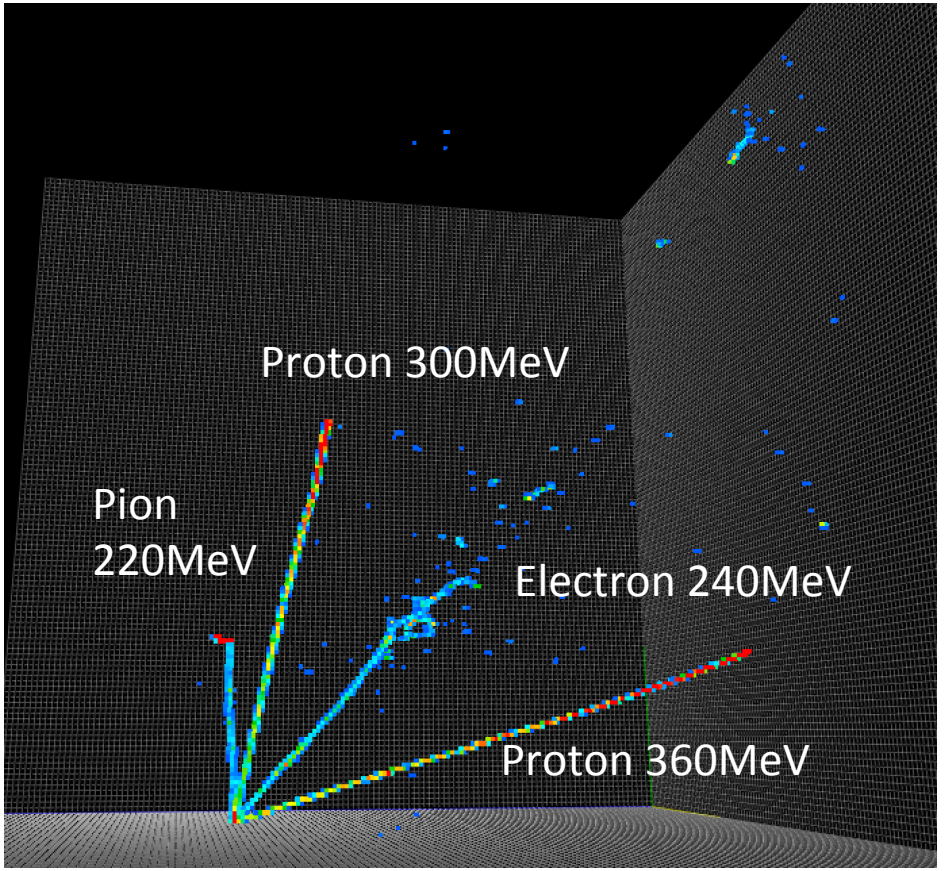
~ "shortcut" operations to strengthen correlation between the low-level details and high-level contextual information.

From MicroBooNE DNN paper under review

Generating images for our training set

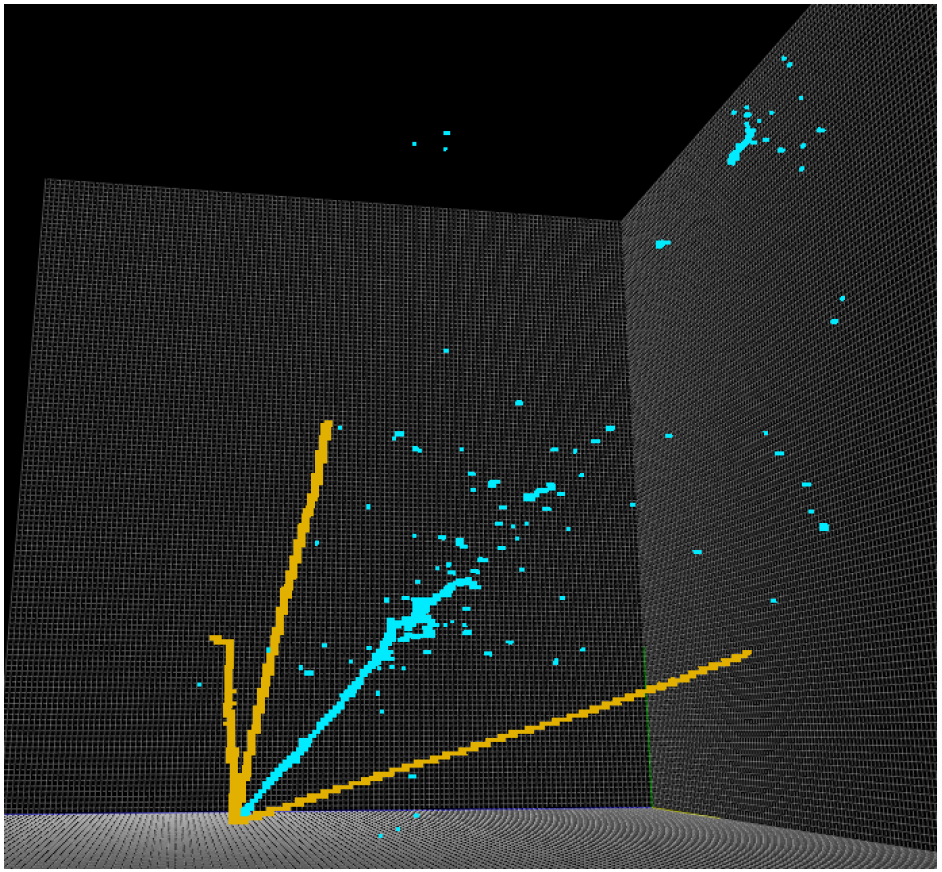
- 3D (voxelized)
- Each event (image) generated from truth energy deposition from LArSoft. With:
  - Randomized particle multiplicity 1~4 from a unique vertex per event, where the 1~4 particles are chosen randomly from 5 particle classes.
  - Momentum varying from 100MeV to 1GeV in isotropic direction.
  - 128 x 128 x 128 voxels → 1cm^3 per voxel (for quick first trial)

Input image: each voxel contains charge info.



Supervised learning: each training example is an ordered pair of *input image* and *true output image* (label).

Label image: each voxel is 0 (background), 1 (shower), or 2 (track).



Yellow: track,
Cyan: shower

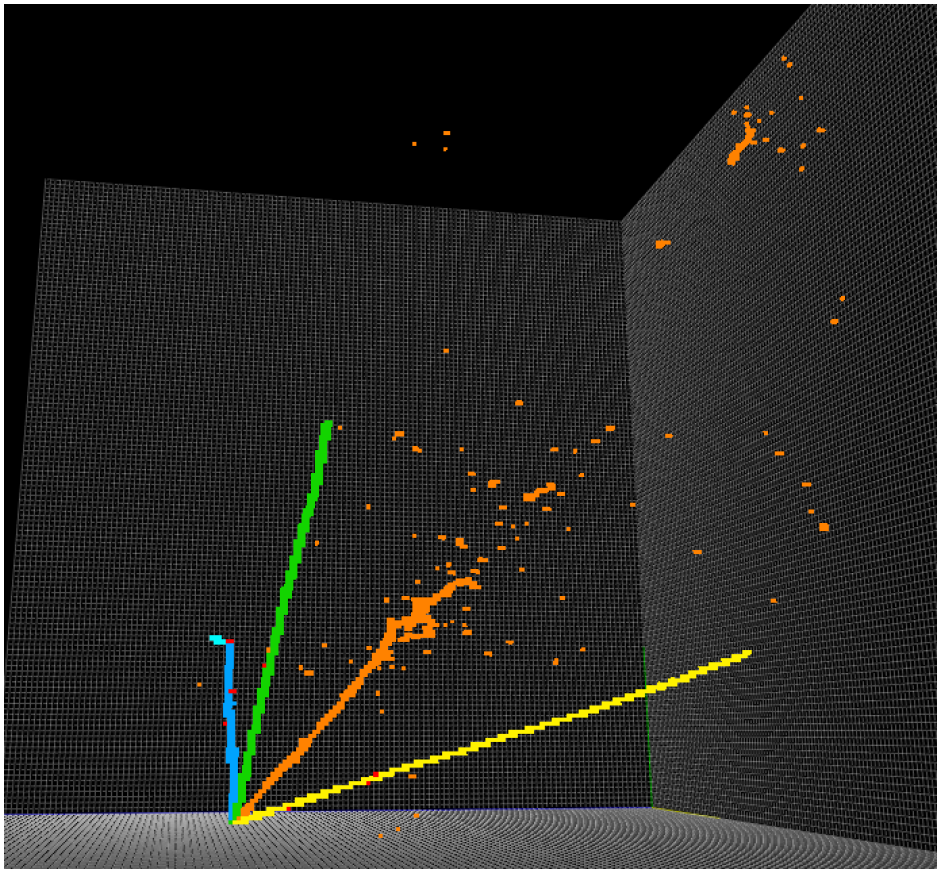
Defining the optimization objective (loss function)

Must weight the softmax cross entropy.

Typically, an image has 99.99% background (zero-value) voxels. Even among non-zero voxels, can have uneven number of track voxels vs. shower voxels.

So upweight the "rarer" classes in the image,

e.g. if the truth label has ratio of BG: track: shower = 99: 0.7: 0.3, incentivize the algorithm to do focus on shower most and BG least by using inverses as weights, 1/99: 1/0.7: 1/0.3.



Similarly, monitor algorithm's performance by evaluating accuracy only for non-zero pixels

Training

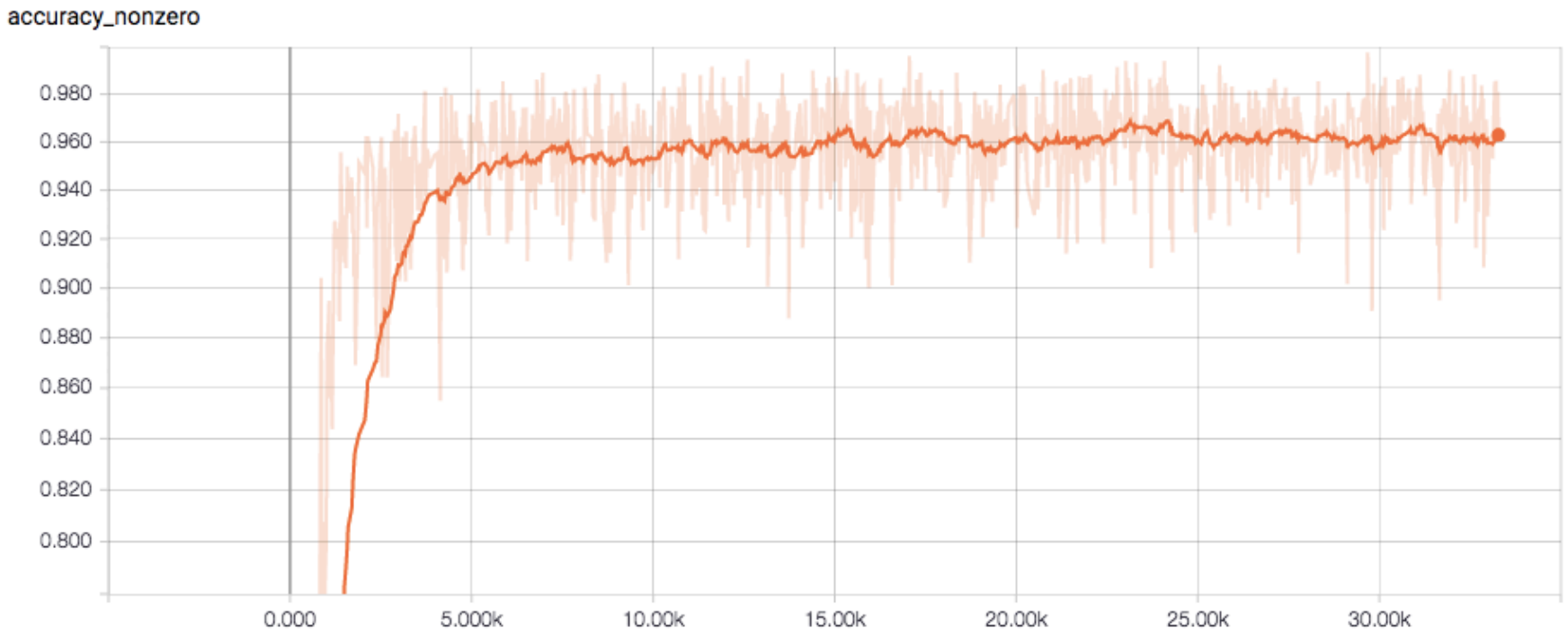
Optimizer: Adam

Choose batch size to be 8 images

- batch size ~ size of ensemble, so bigger the better BUT limited by GPU memory
- one iteration consumed 8 images

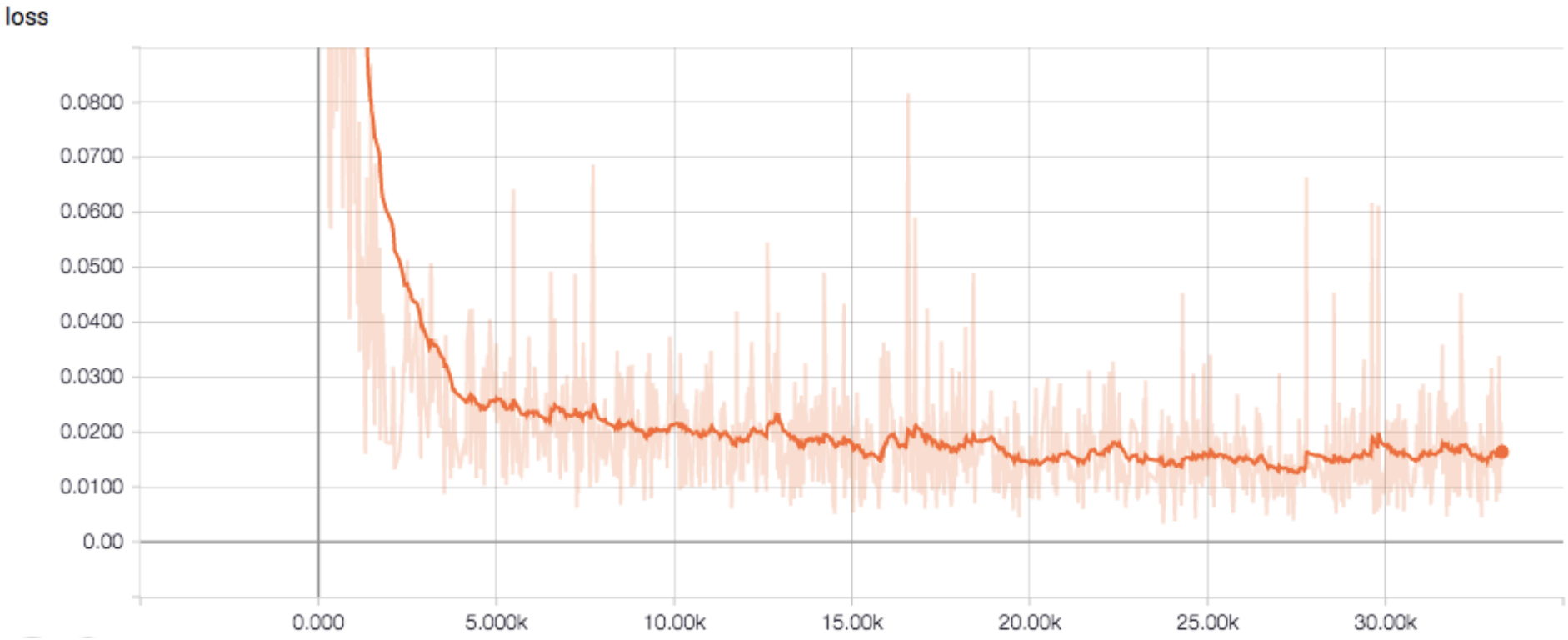
RESULTS

Non-zero pixel accuracy curve



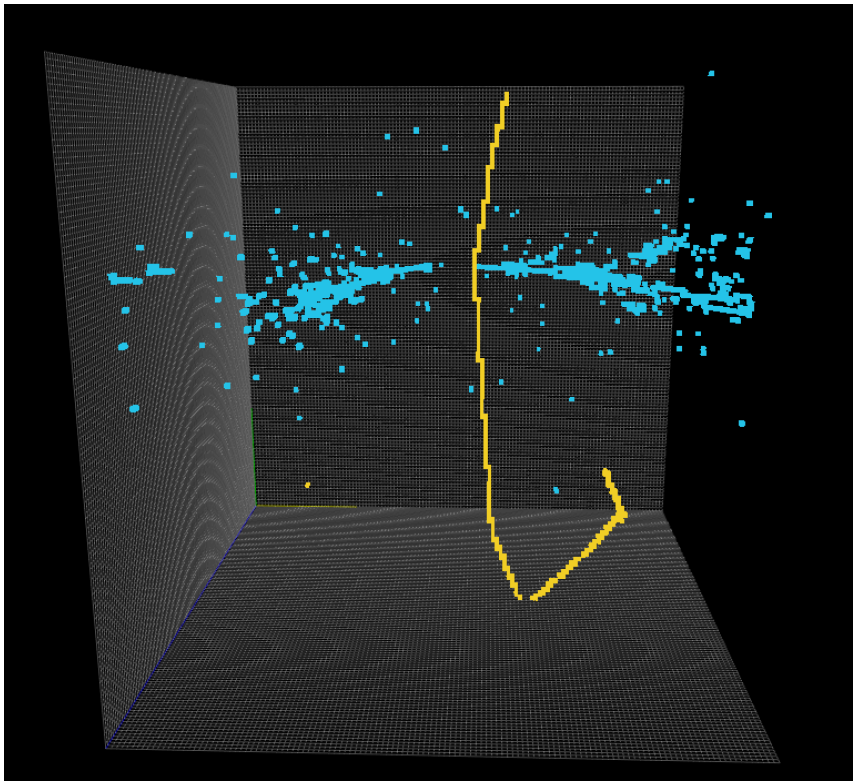
Non-zero pixel accuracy = correctly predicted nonzero pixels / total nonzero pixels
Each iteration consumed 8 images.
Light orange: raw plot
Dark orange: smoothed plot

Loss curve

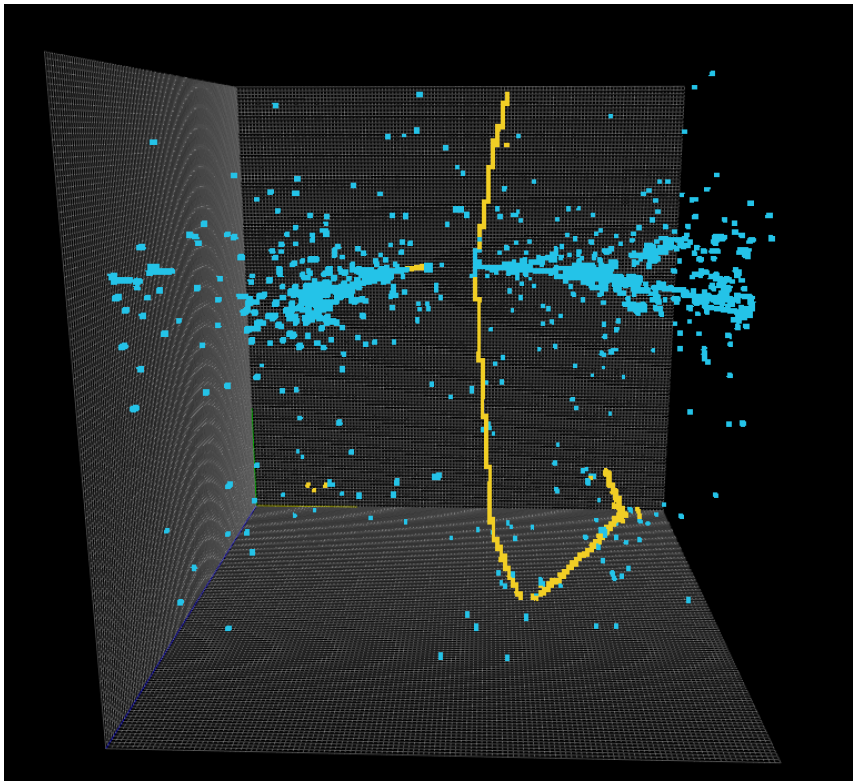


Each iteration consumed 8 images.
Light orange: raw plot
Dark orange: smoothed plot

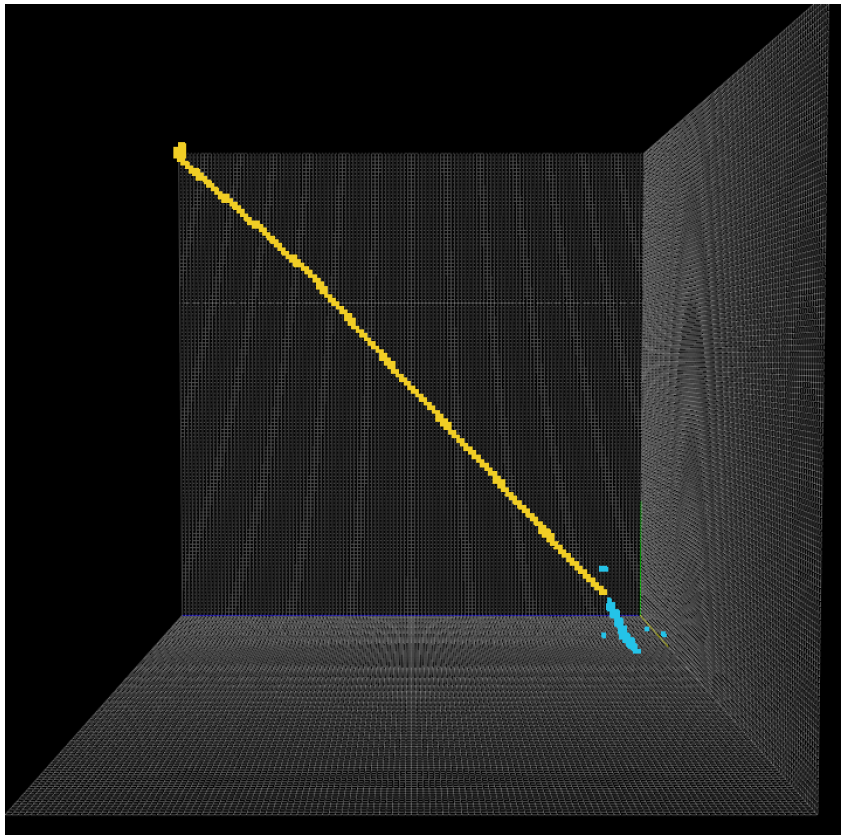
Truth label



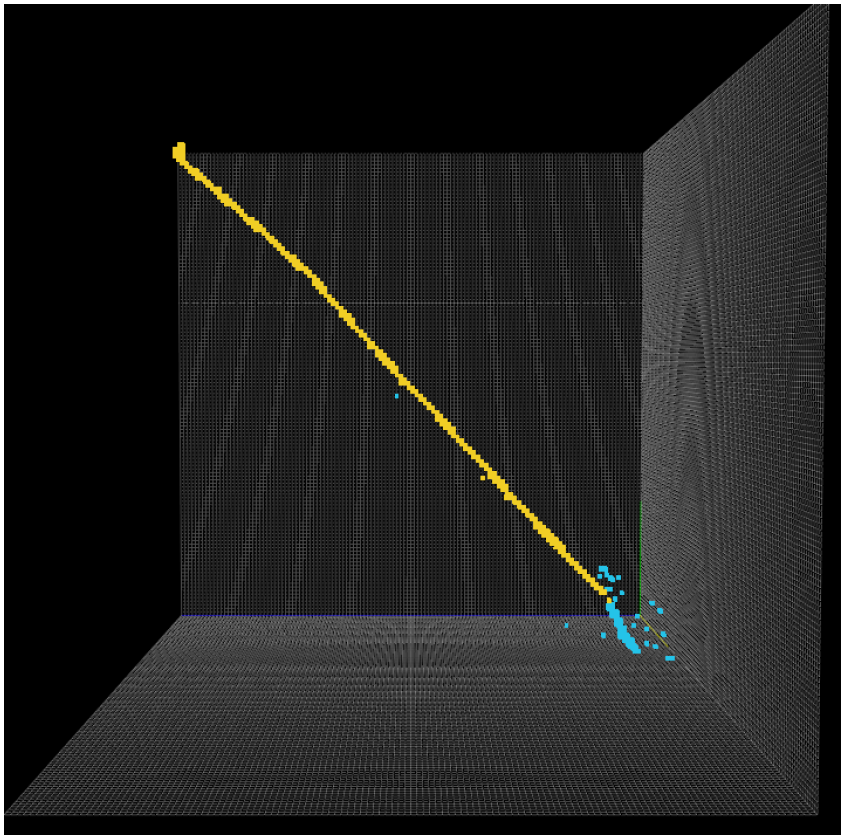
Prediction



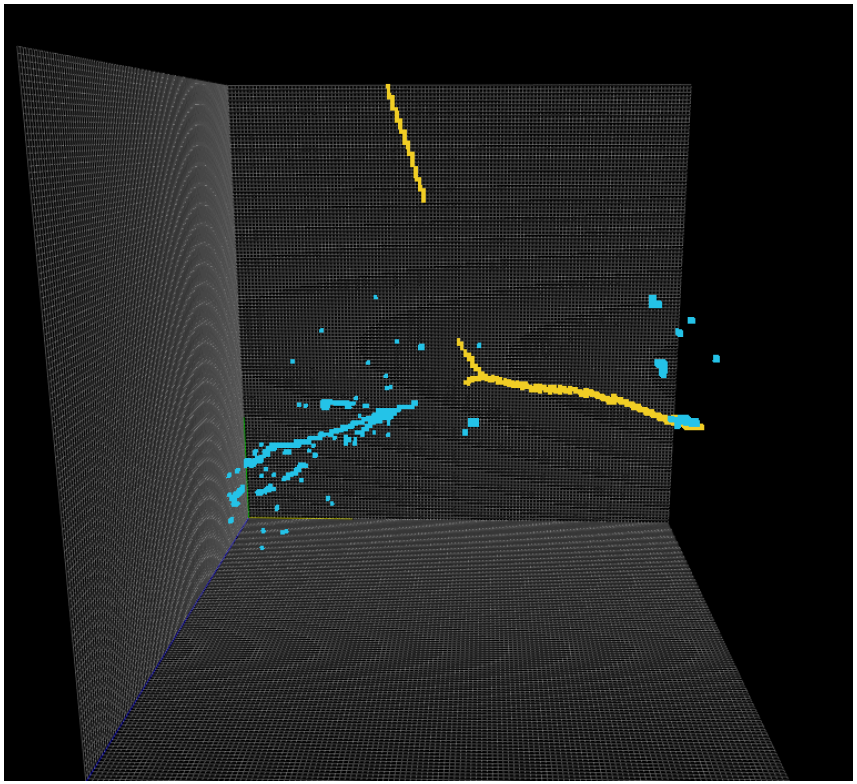
Truth label



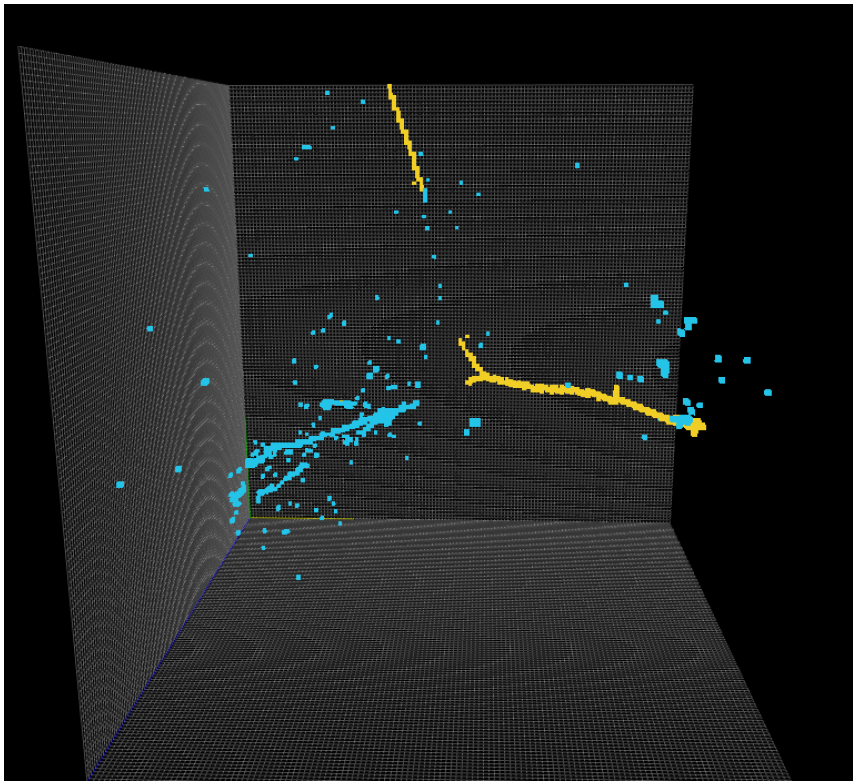
Prediction



Truth label



Prediction



Summary and future work

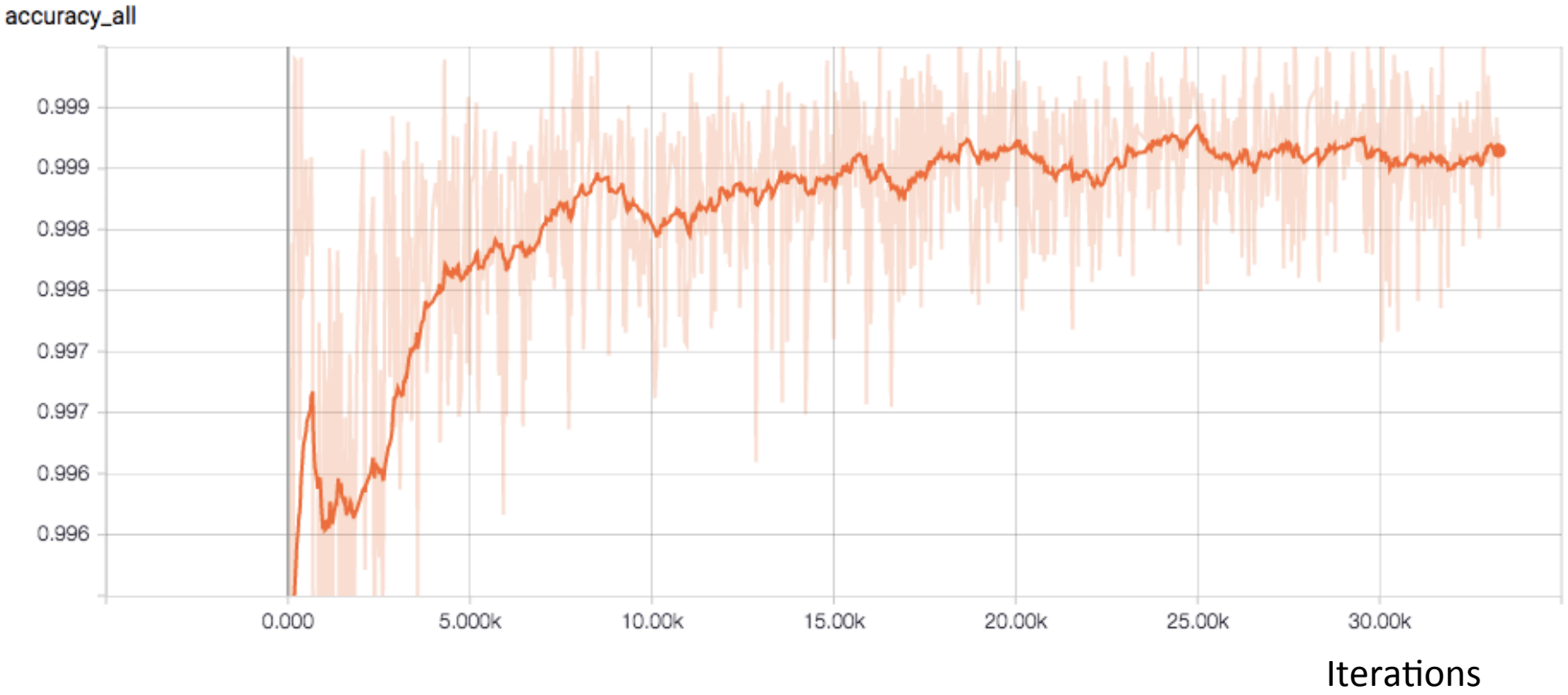
We have trained U-Resnet to perform shower/track separation on 3D simulation data and report a training accuracy of ~96%.

To do:

- Explore smaller voxel sizes for higher precision
- Vertex finding (adding 1 more class to the classification task)
- Particle clustering (instead of pixel-level, instance-aware classification)

BACKUP SLIDES

Overall accuracy curve



Why ResNet?

This paper demonstrates why ResNet is superior to vgg, etc. in semantic segmentation.