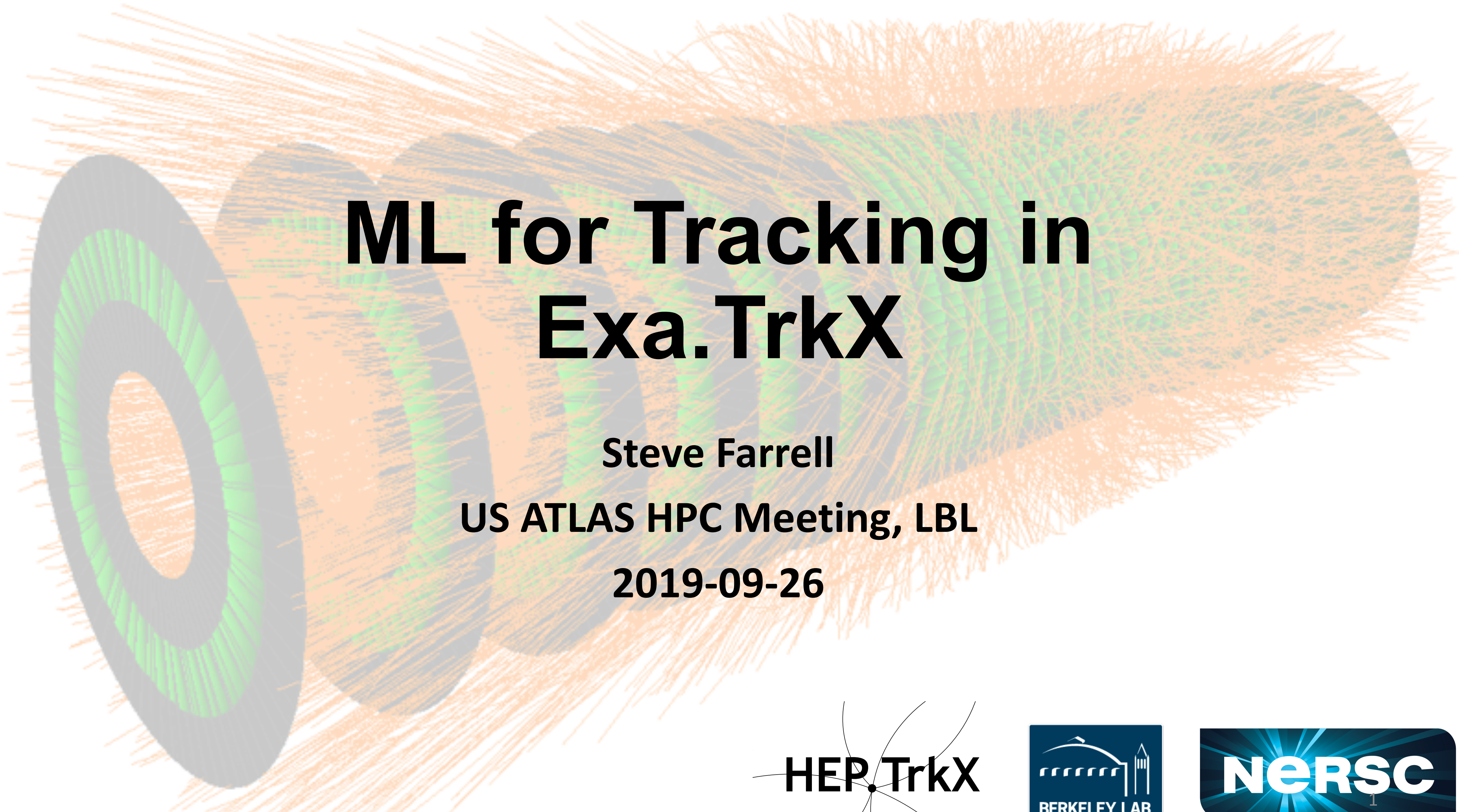


ML for Tracking in Exa.TrkX

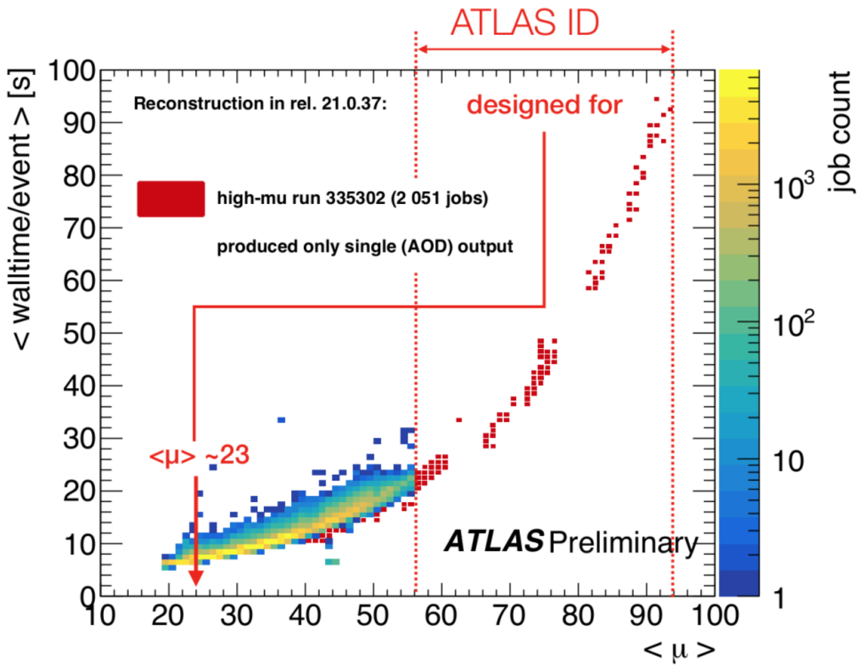
Steve Farrell

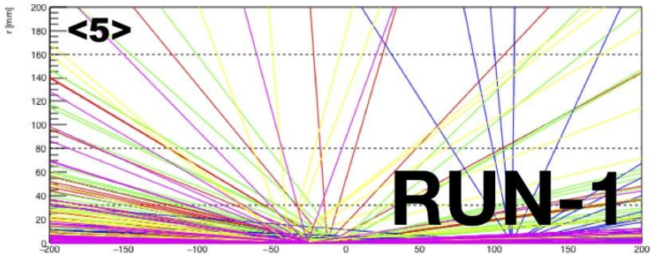
US ATLAS HPC Meeting, LBL

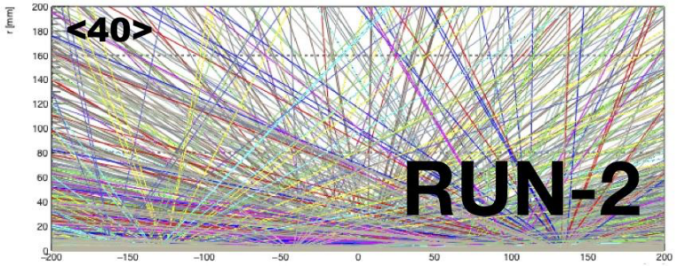
2019-09-26

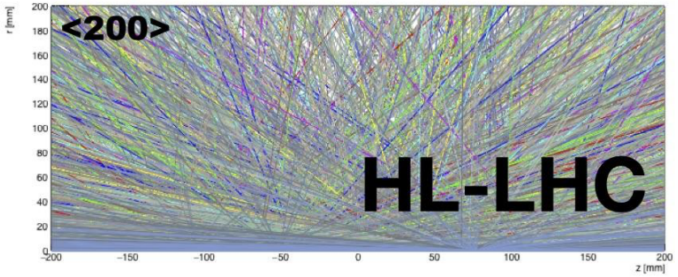
Tracking challenges

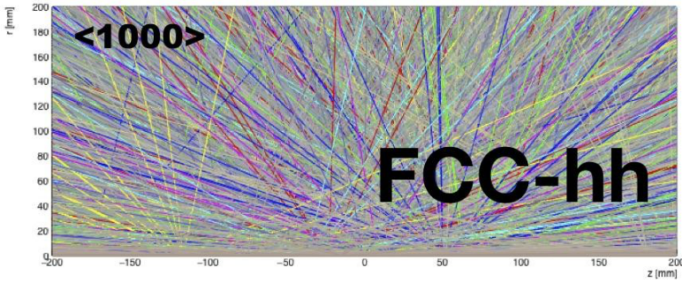
- **Combinatorial explosion with increasing occupancy**
- **Track reconstruction will dominate CPU consumption**
- **Algorithms are**
    - hard to parallelize
    - hard to run on SIMD architectures











Thinking outside the box

- **The HEP.TrkX Project**
    - DOE HEP-CCE pilot project to develop Deep Learning solutions to particle track reconstruction
    - Collaboration between LBNL, Caltech, and FNAL
- **The TrackML Challenge**
    - Engaging with the broader DS/ML community to develop solutions
    - Challenges hosted on Kaggle and Codalab
- **The HEP.QPR Project**
    - Developing quantum computing solutions

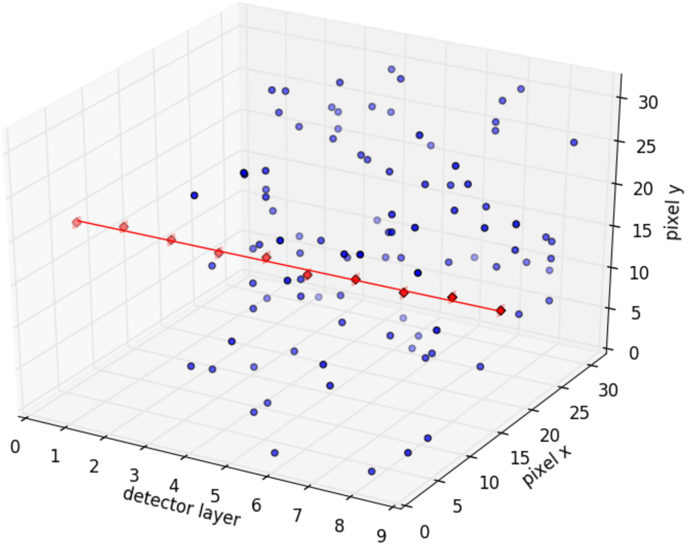
ML + HPC for HEP

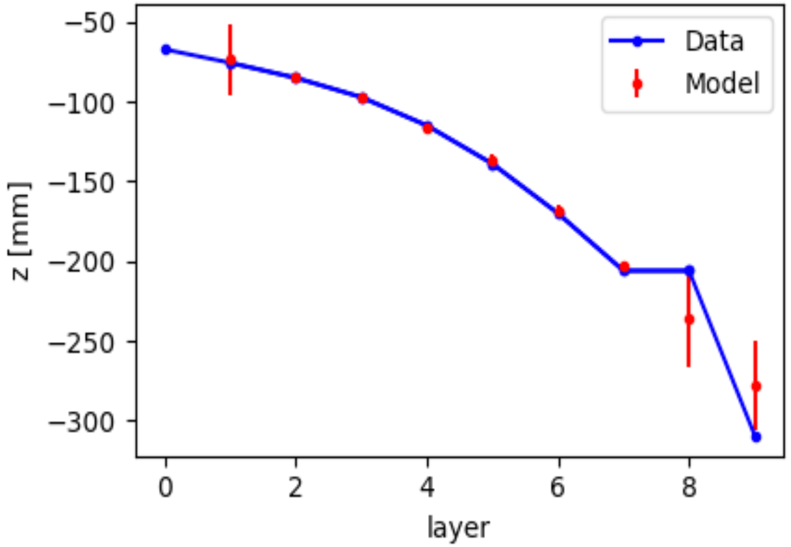
- **Why ML?**
  - Expressive models learned from data
  - Regular computation which maps well onto modern hardware
- **Why HPC?**
  - Large scale systems with high performance hardware
  - Potentially fast model training times
  - Fast model inference for deployed reconstruction workloads

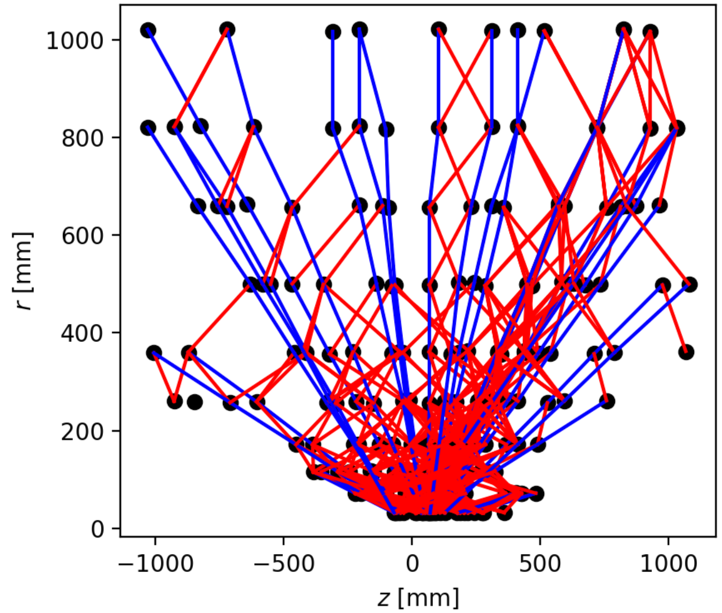
The HEP.TrkX Project

https://heptrkx.github.io

- **Pilot project to investigate ML solutions to tracking at the LHC**
- **We tried various methods and representations:**
    - Detector “images” with segmentation and “captioning” models
    - Track sequences with Recurrent Neural Networks
    - Hit graphs with Graph Neural Networks (GNNs)







The Exa.TrkX Project

https://exatrkx.github.io

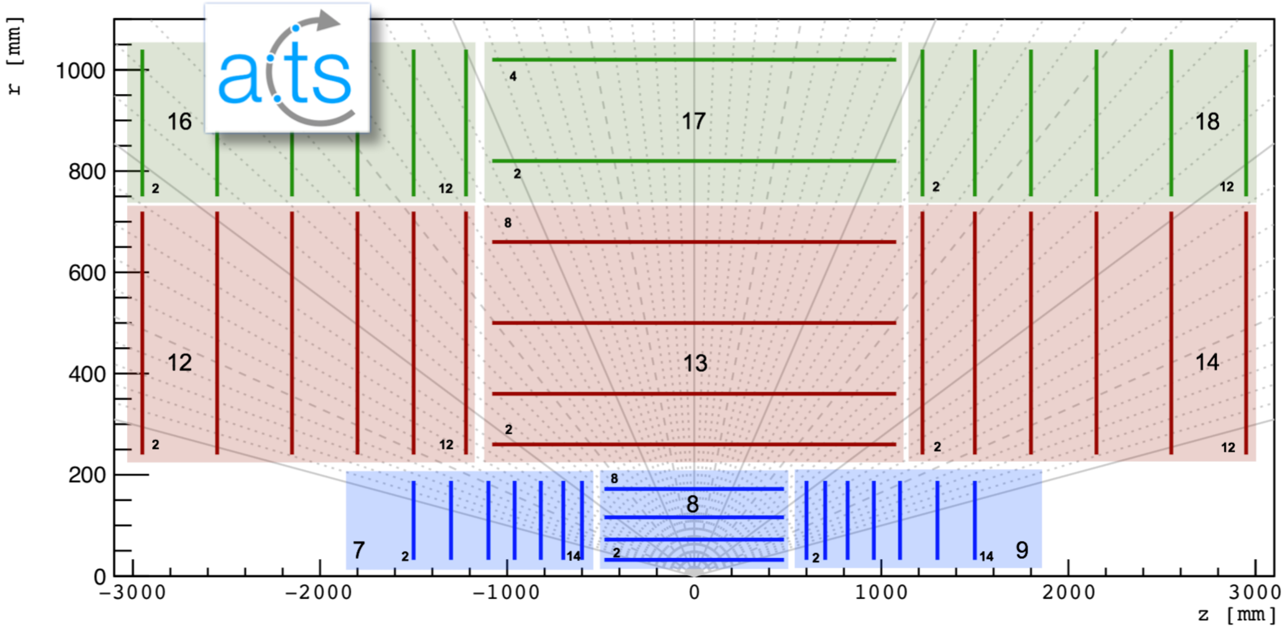
*A DOE CompHEP project that will deliver production-quality **ML tracking models** that run efficiently on next-gen computing architectures, from triggering systems to **DOE exascale-class HPC systems**.*

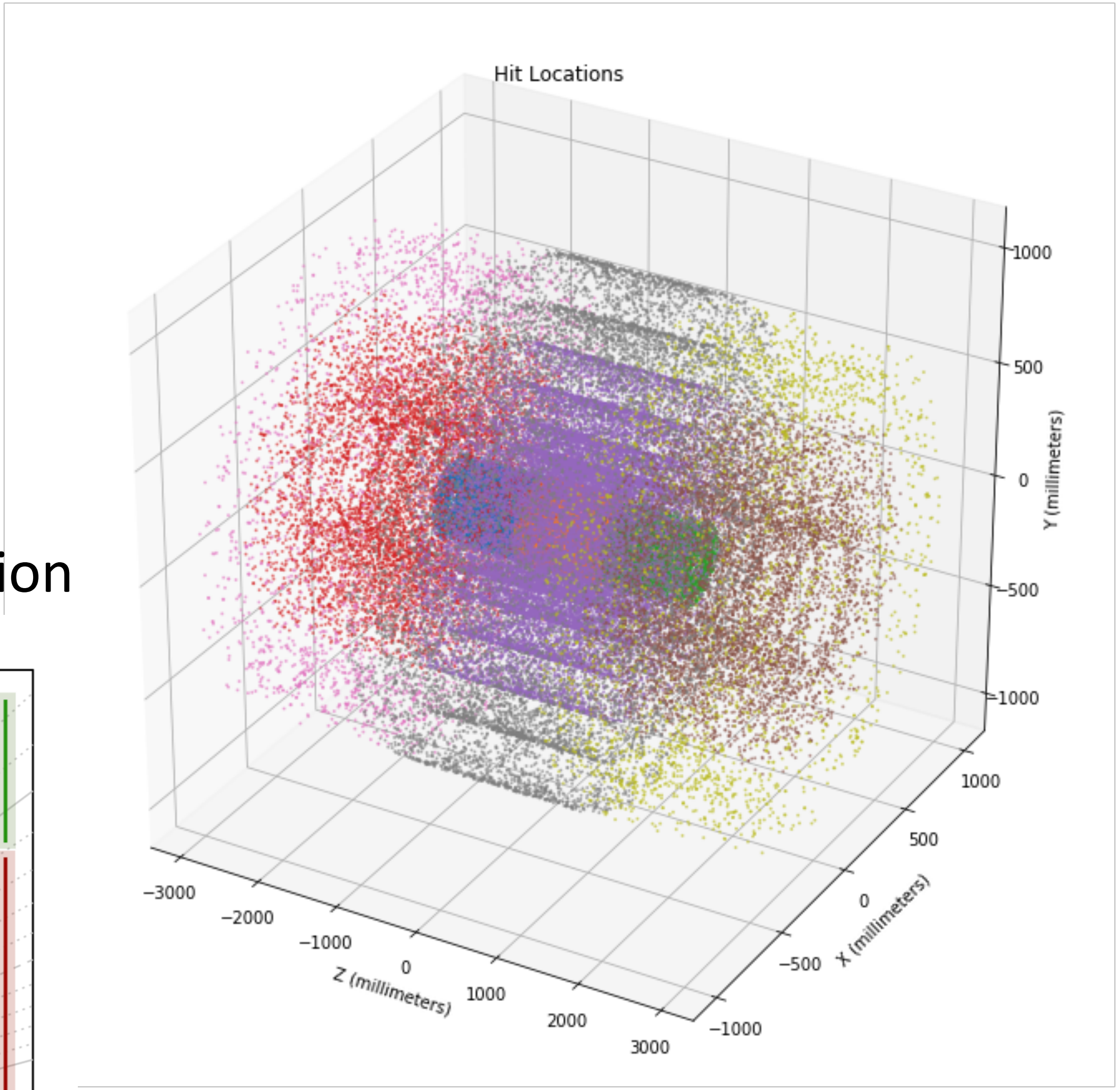
People

- **Caltech**: Joosep Pata, Maria Spiropulu, Jean-Roch Vlimant
- **Cincinatti**: Adam Aurisano, Jeremy Hewes
- **FNAL**: Giuseppe Cerati, Lindsey Gray, Thomas Klijnsma, Jim Kowalkowski, Gabriel Perdue, Panagiotis Spentzouris
- **LBNL**: Paolo Calafiura, Steve Farrell, Xiangyang Ju, Daniel Murnane, Prabhat
- **ORNL**: Aristeidis Tsaris
- **SLAC**: Kasuhiro Terao, Tracy Usher

Data used for studies

- 2D and 3D toy data (planes)
- Simulated data with ACTS toolkit
  - Uses *generic HL-LHC* detector description





https://gitlab.cern.ch/acts/acts-core
https://www.kaggle.com/c/trackml-particle-identification/data

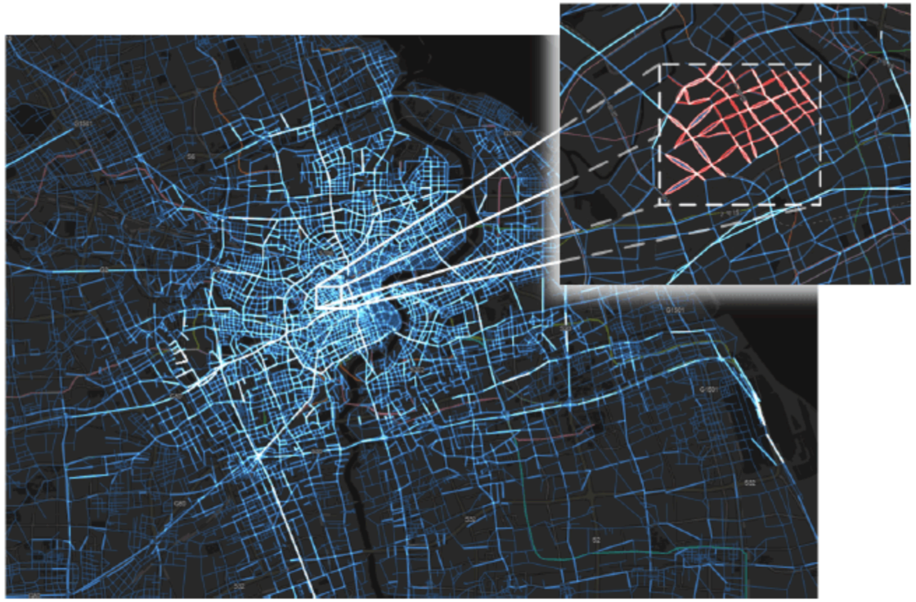
Geometric Deep Learning

http://geometricdeeplearning.com

Shape analysis

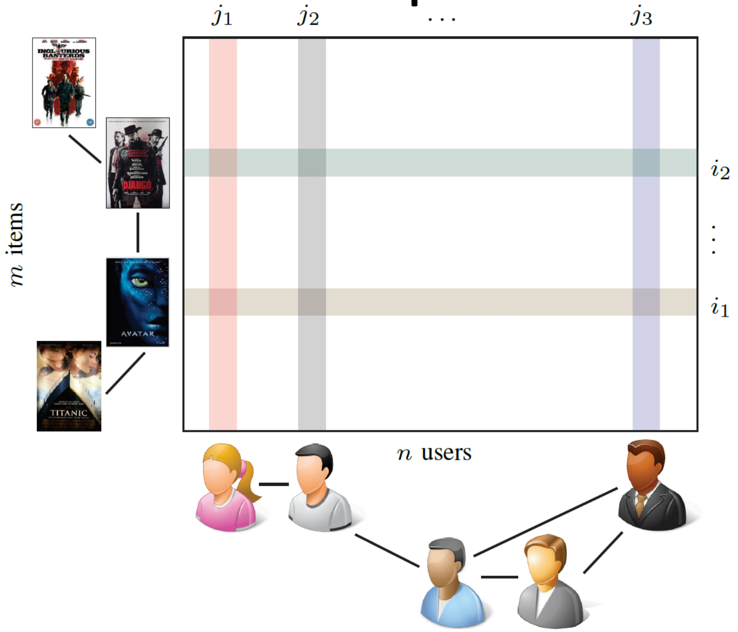
https://arxiv.org/pdf/1611.08097.pdf

Modeling traffic



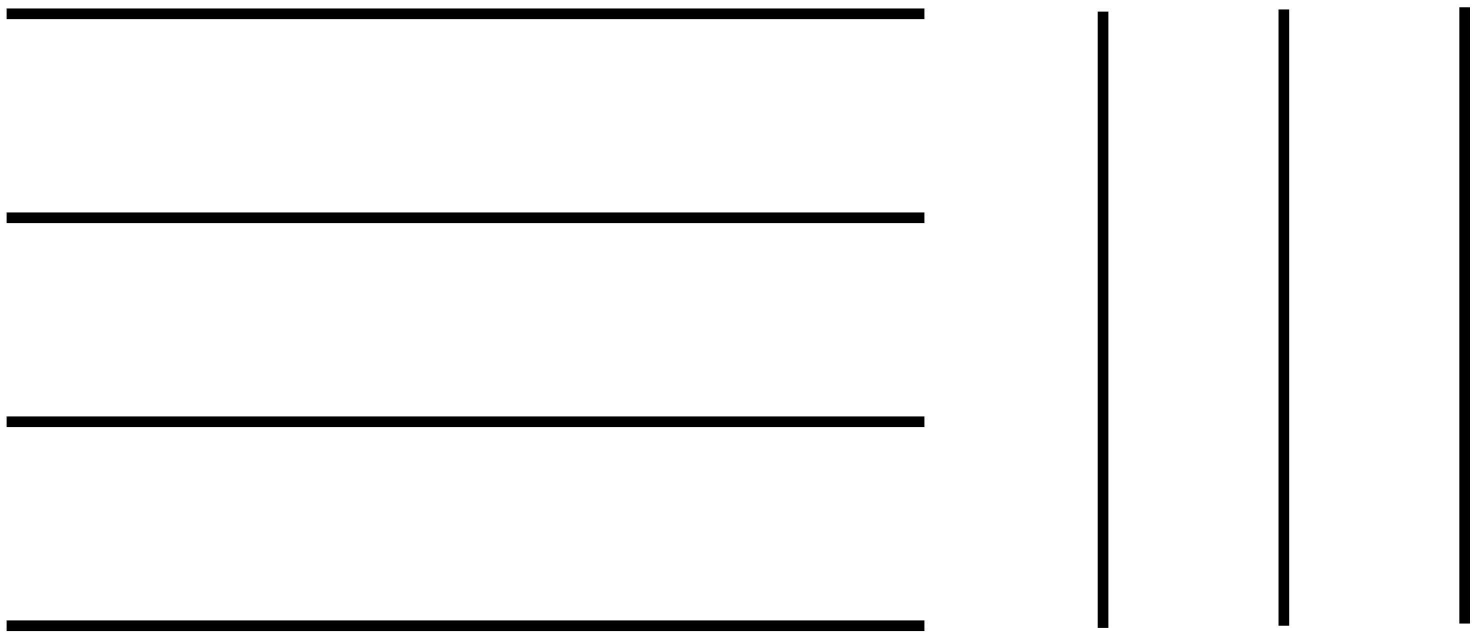
https://medium.com/syncedreview/shanghai-tests-graph-recurrent-neural-networks-for-traffic-prediction-fdd4c2182b53

Recommender systems / matrix completion



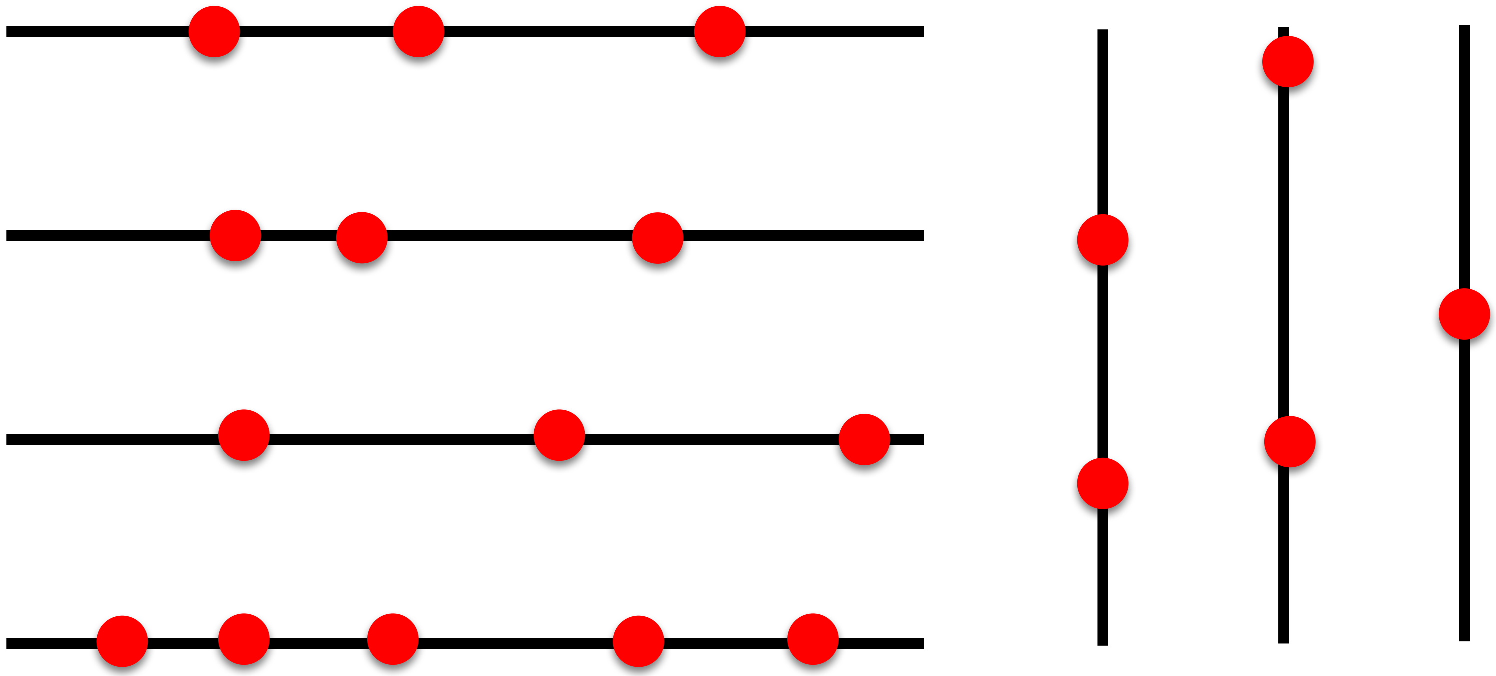
https://arxiv.org/abs/1704.06803

Graph representation



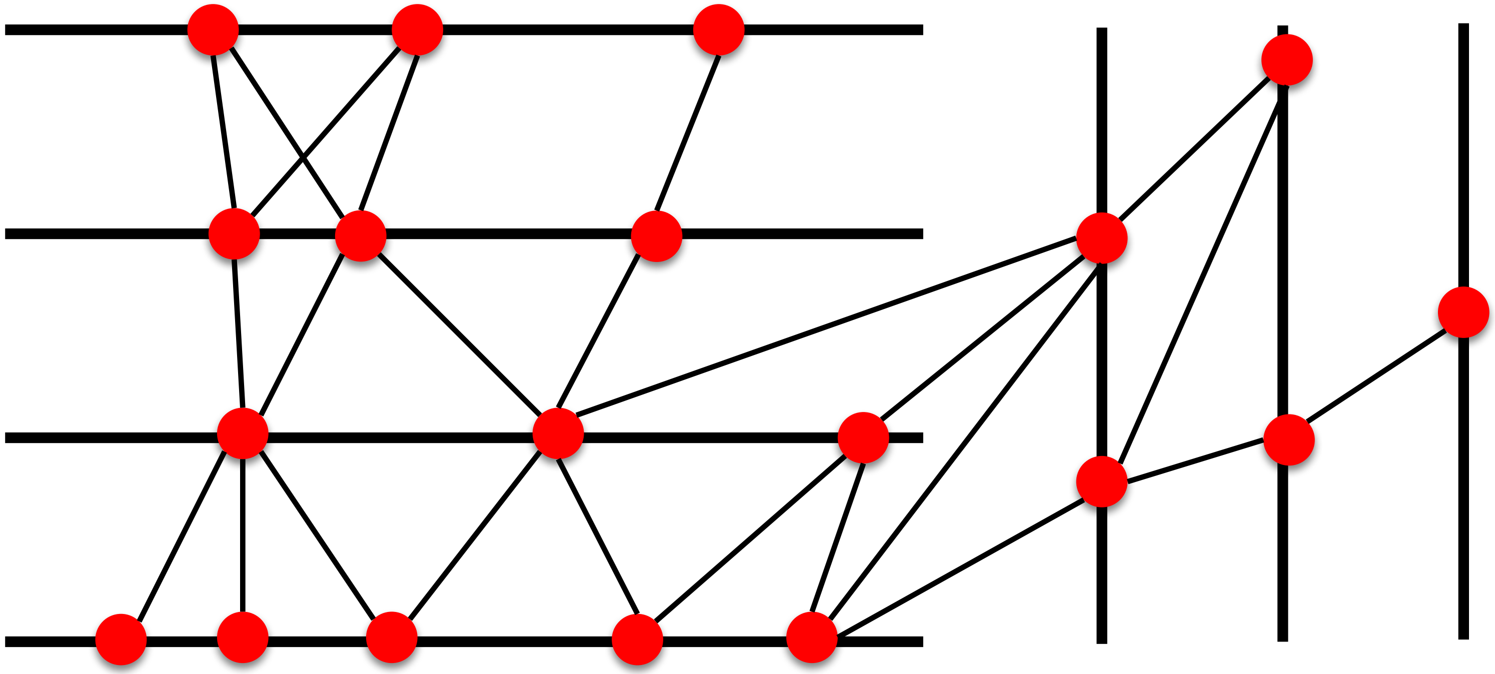
- Detector geometry

Graph representation



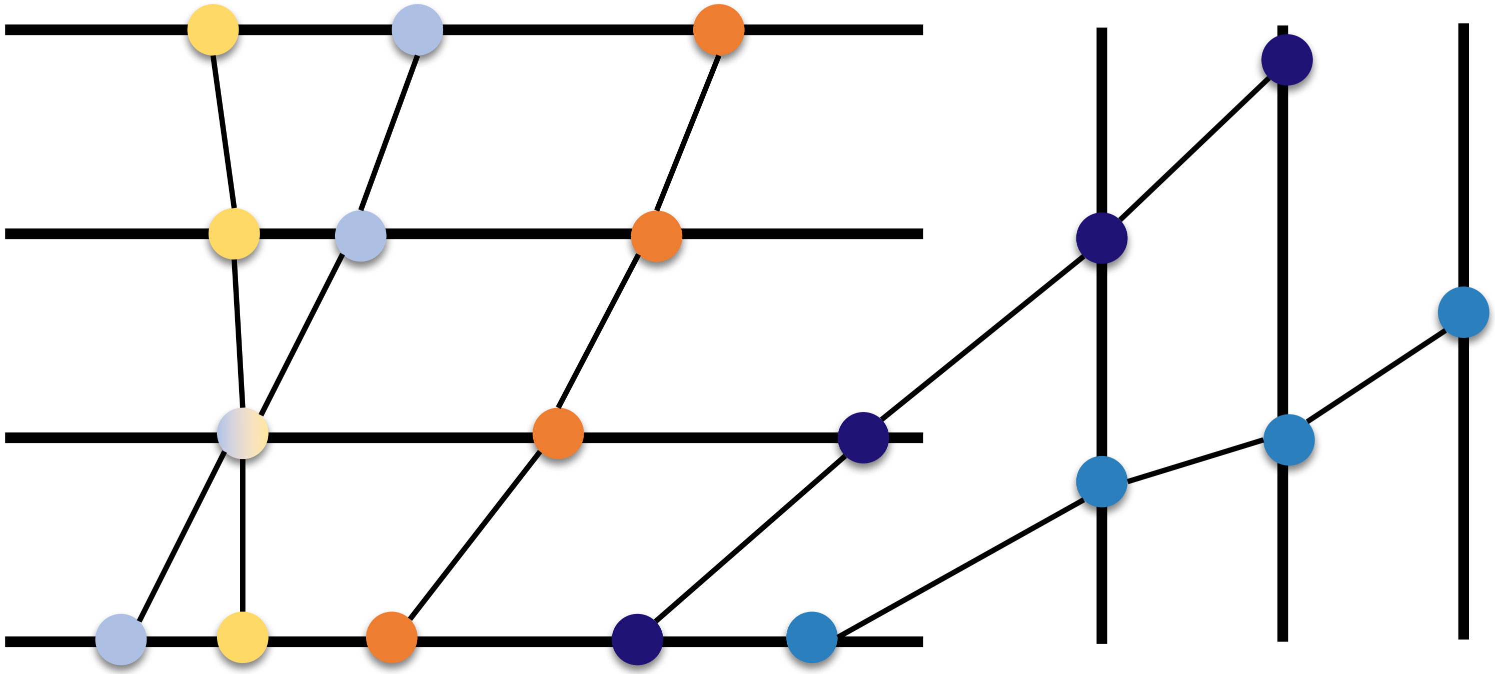
- Particle hit data

Graph representation



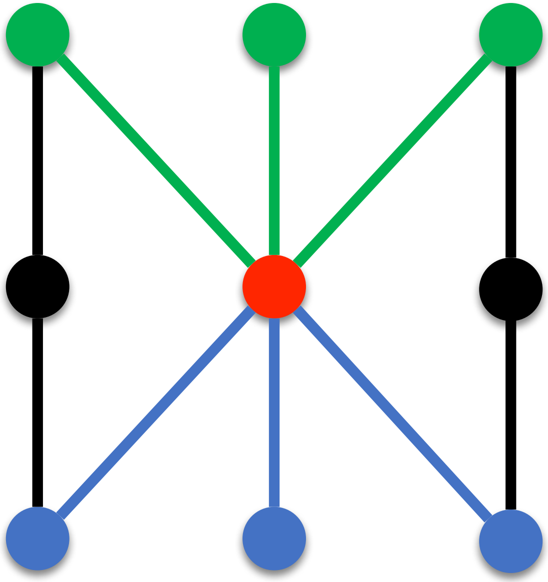
- Connect compatible hits together to construct a graph

Graph representation

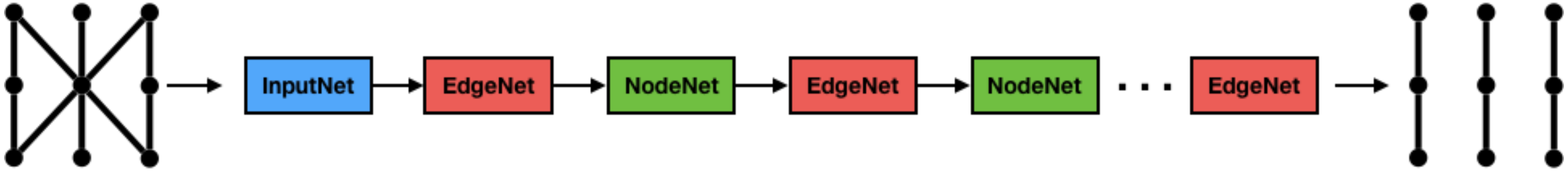


- Try to resolve the tracks with *Graph Neural Networks*

GNNs for tracking



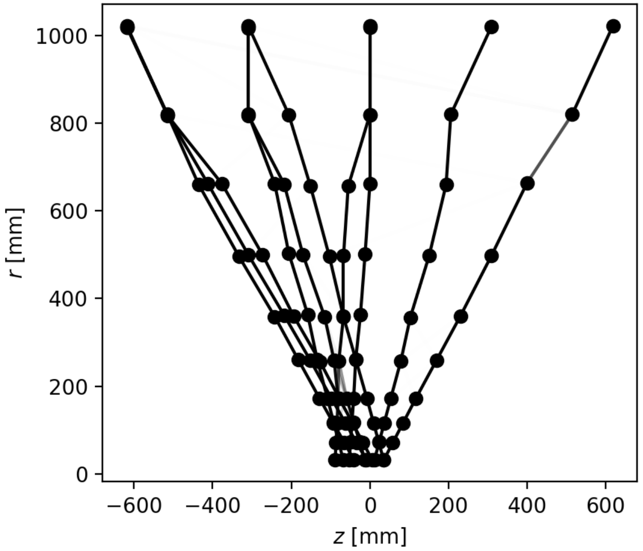
- **Message-passing architecture**
  - Computes messages to send to neighbors
  - Aggregates messages at nodes and computes new node features
- **Binary edge classification**
  - Identifies true track segments

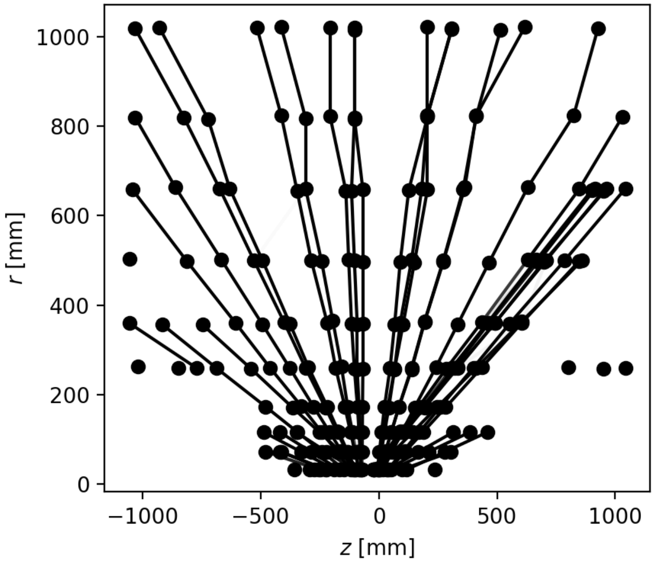


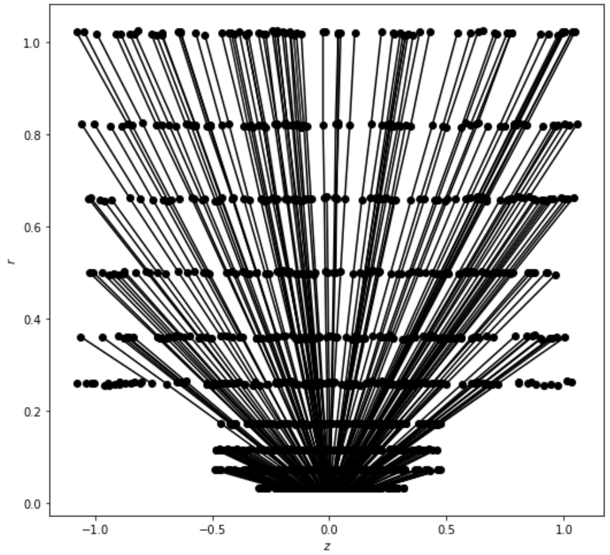
Code: https://github.com/HEPTrkX/heptrkx-gnn-tracking

Progress of results

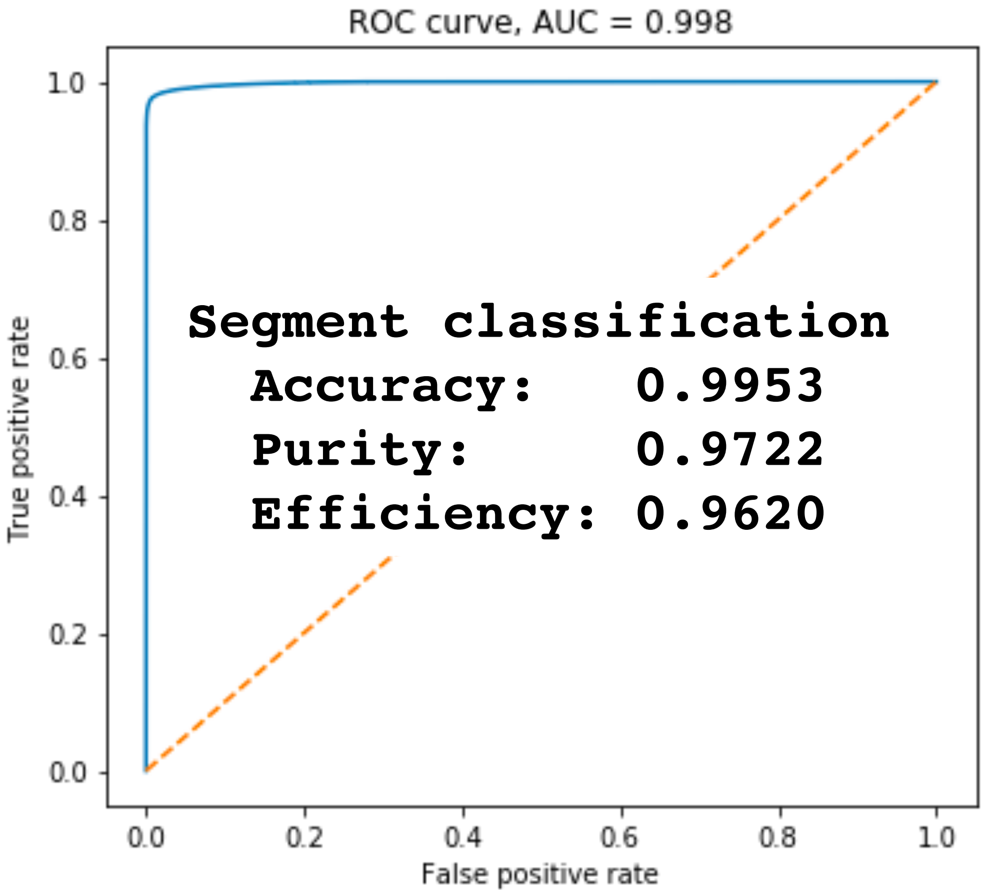
- **The basic approach has held up as we increase data complexity**
- **Ongoing work to add detector endcaps and polish the graph post-processing**
  - With robust handling of shared, missing, and double hits







Results

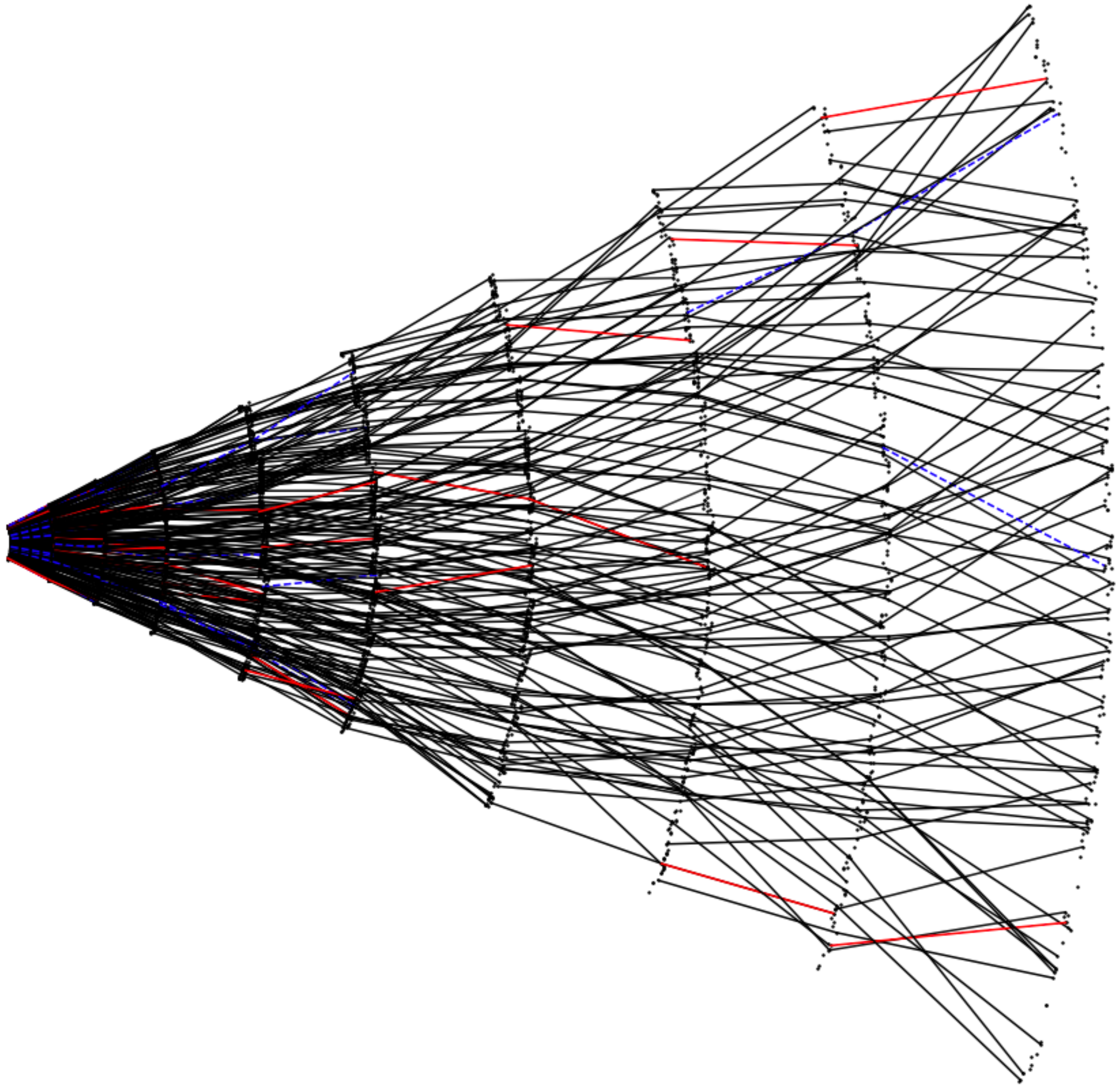


Segment classification

Accuracy: 0.9953

Purity: 0.9722

Efficiency: 0.9620

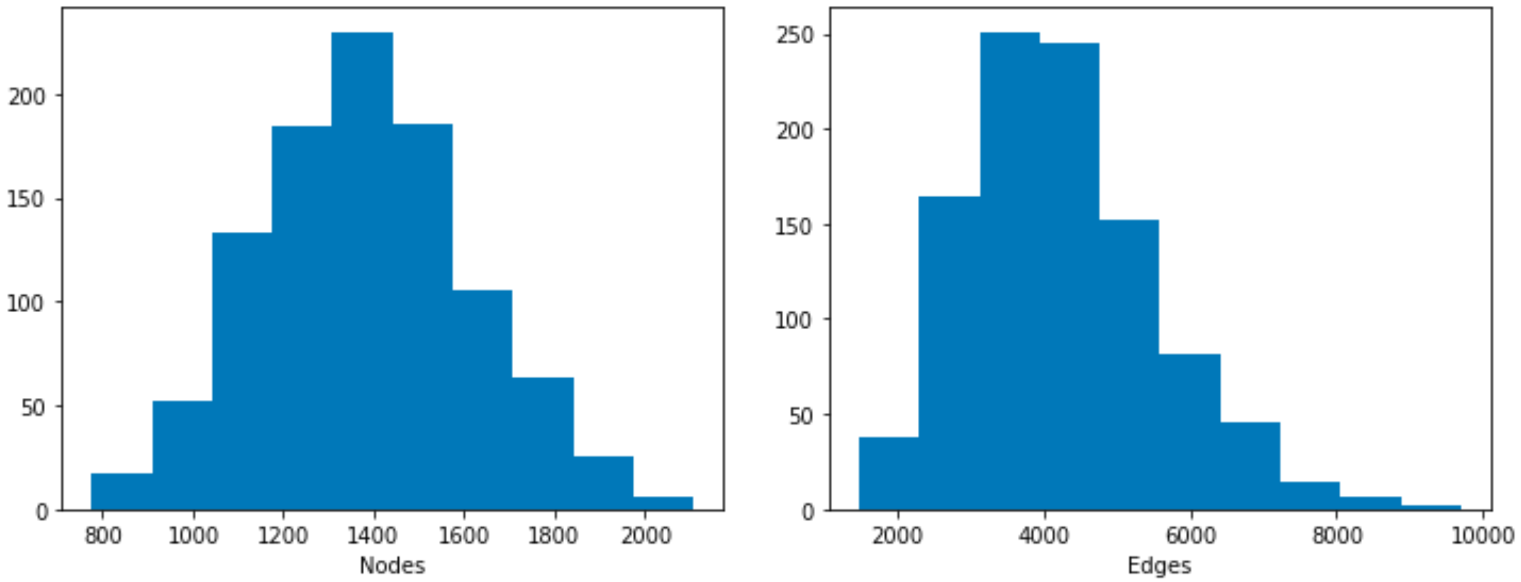


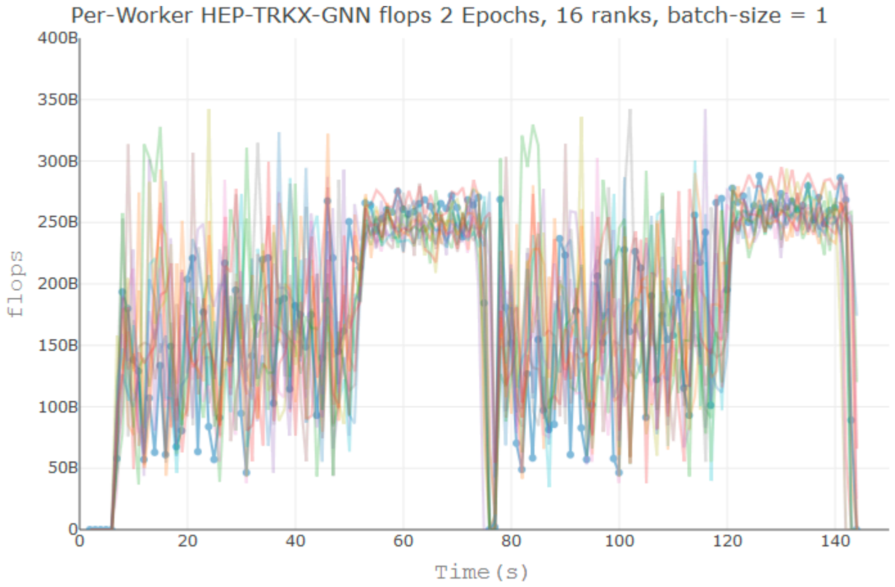
Large scale training and inference

- **Can we utilize large-scale HPC resources with these models?**
  - Faster training on large datasets
  - Efficient accelerator utilization for reconstruction
- **What are some of the challenges?**
  - Sparse graph connectivity, load imbalance, GNN convergence at scale
- **We’re partnering with the Cray Big Data Center and LBL CRD to investigate and develop optimized solutions**

Load balancing

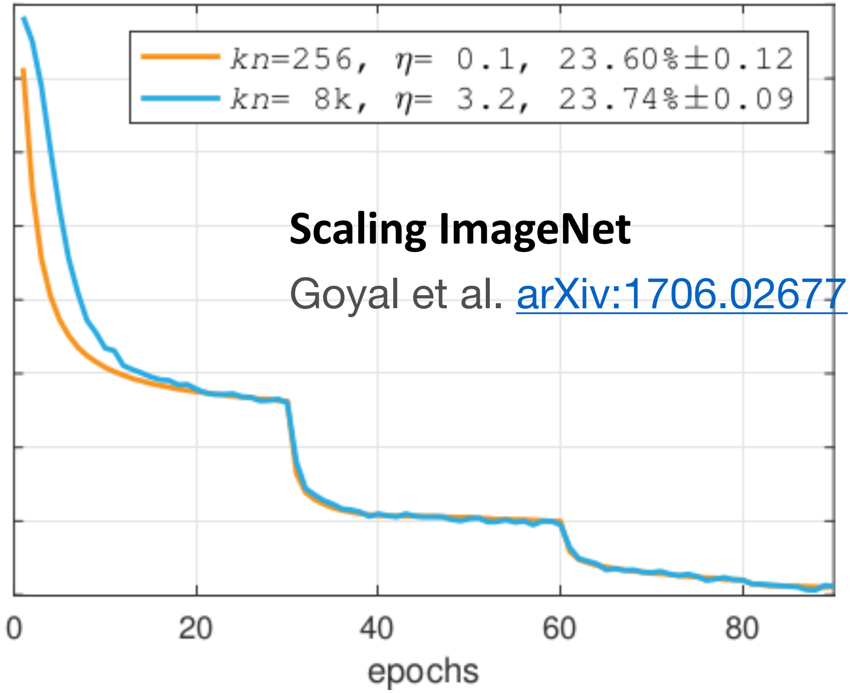
- **Training samples have variable-sized graphs**
  - Big load imbalance in synchronized training
- **How to address it?**
  - Dynamic graph partitioning
  - Batch like-sized samples across workers
  - Need to be careful not to introduce too much bias




Per-Worker HEP-TRKX-GNN flops 2 Epochs, 16 ranks, batch-size = 1

GNN convergence at scale

- **Significant research effort has gone into scaling computer vision applications (e.g. ResNet ImageNet)**
  - Many different techniques to address large-batch convergence issues
- **However, large scale training of GNNs still relatively unexplored**
  - Do the same optimizers work?
  - Do the same learning rate scheduling tricks work?



Scaling ImageNet

Goyal et al. arXiv:1706.02677

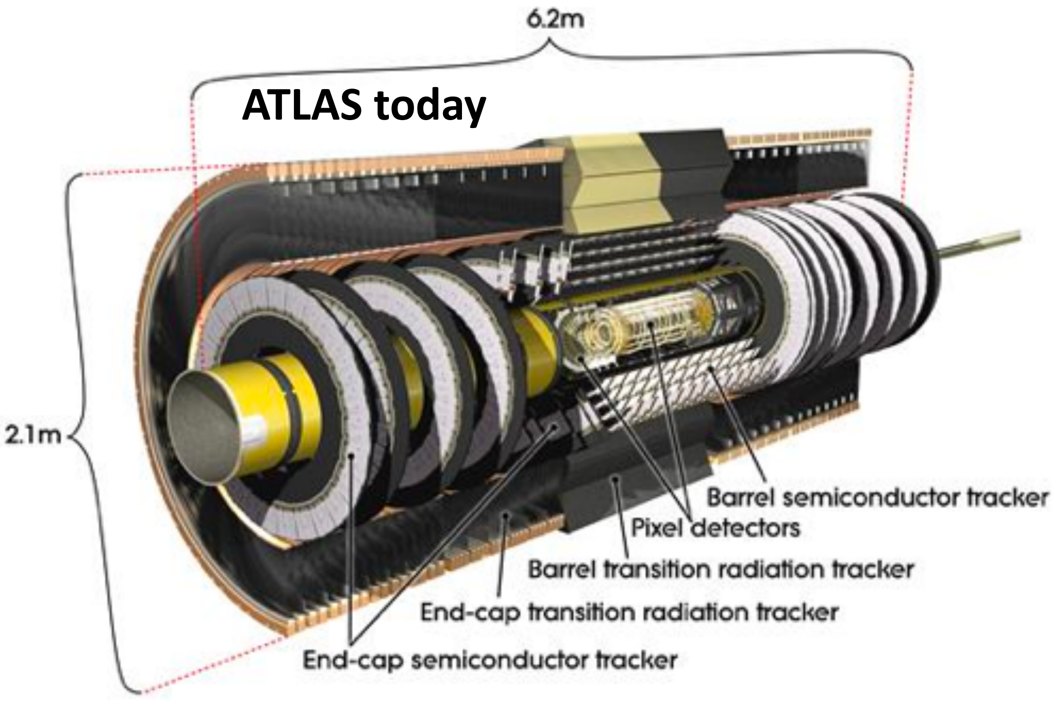
Summary

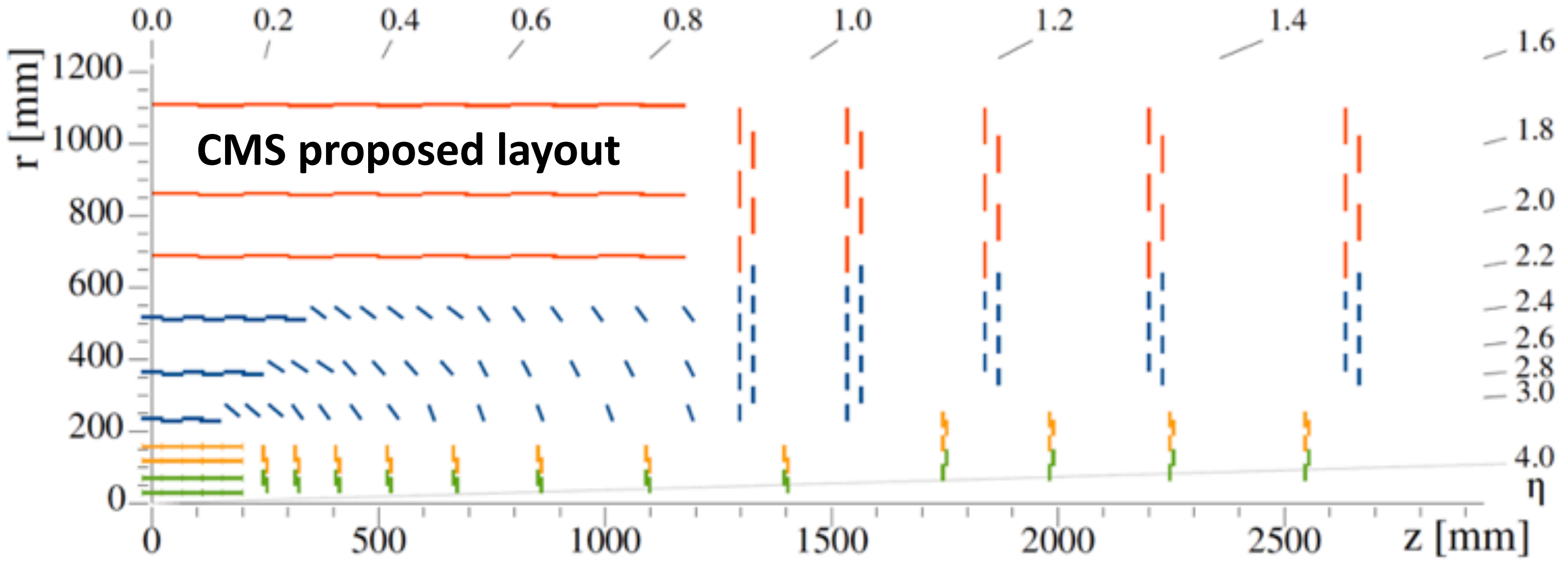
- **The Exa.TrkX Project is picking up after HEP.TrkX**
    - Finishing, productionizing methods
    - Expanding to new applications (e.g. LArTPC)
    - Scaling
- **Scaling GNN training on HPC is particularly interesting/challenging**
    - Progress has been made
    - Work ongoing

Backup

HL-LHC tracking detectors

- Cylindrical barrel and disk-shaped endcap detector layers
- Silicon pixel and strip detector technologies

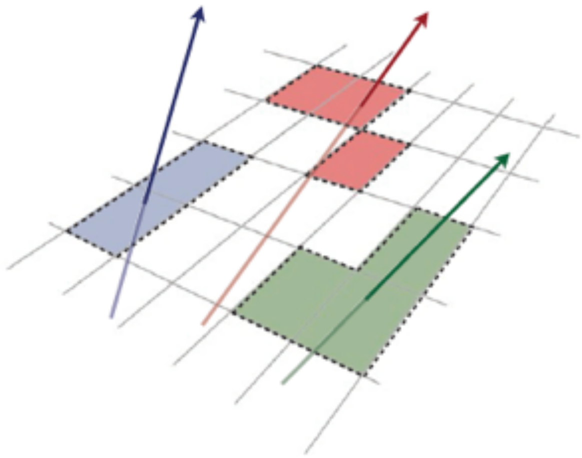

6.2m
ATLAS today
2.1m
Barrel semiconductor tracker
Pixel detectors
Barrel transition radiation tracker
End-cap transition radiation tracker
End-cap semiconductor tracker

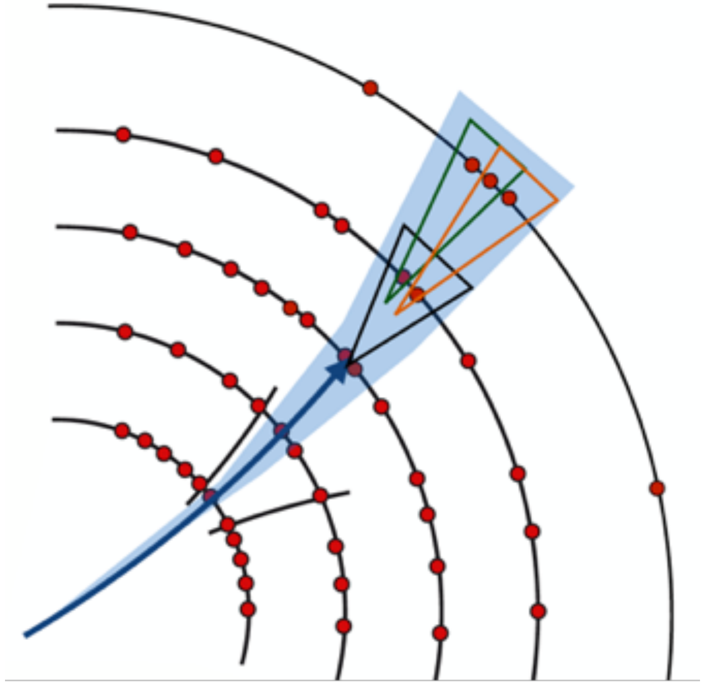


- 100 million readout channels
- Complex layouts

Today’s tracking algorithms

- **Hit clustering**: cells → spacepoints (“hits”)
- **Seed finding**: construct hit triplets
- **Track building**: extend seeds and search with combinatorial Kalman Filter
- **Track fitting/selection**: Resolve ambiguities, fit track parameters





Credit: Andy Salzburger

Deep Learning inspirations

Image segmentation

https://arxiv.org/abs/1604.02135

Our goal (more or less…):

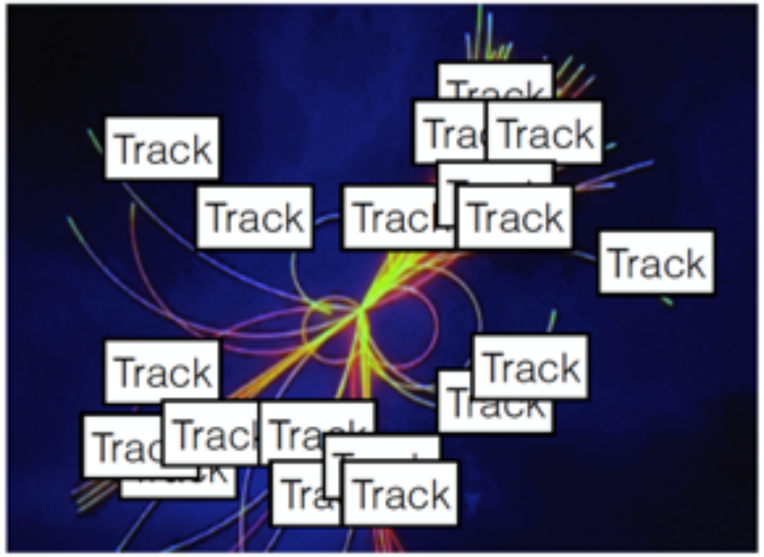
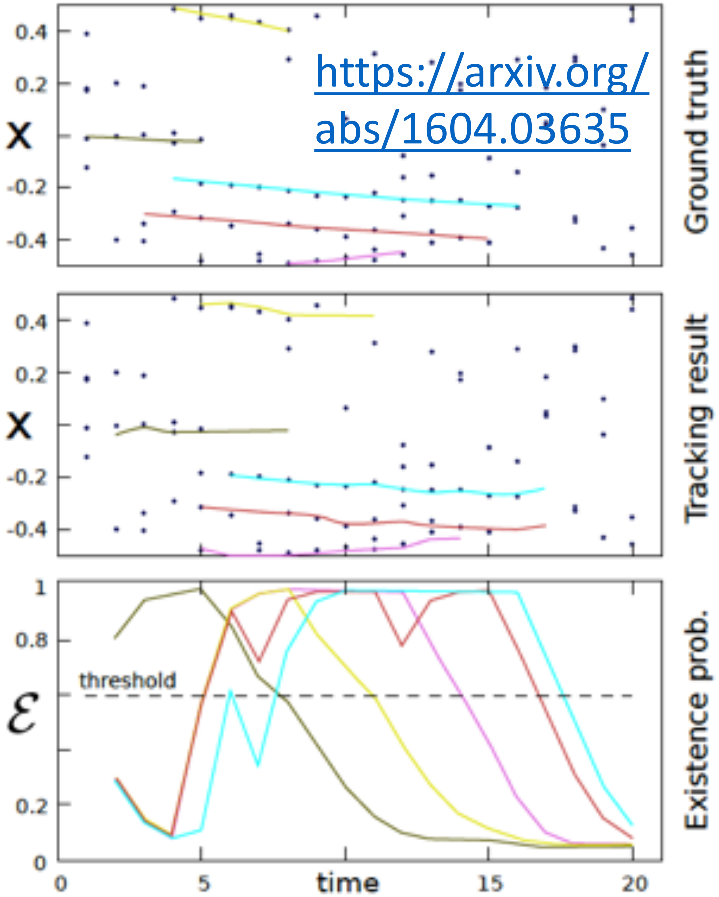


Photo by Pier Marco Tacca/Getty Images

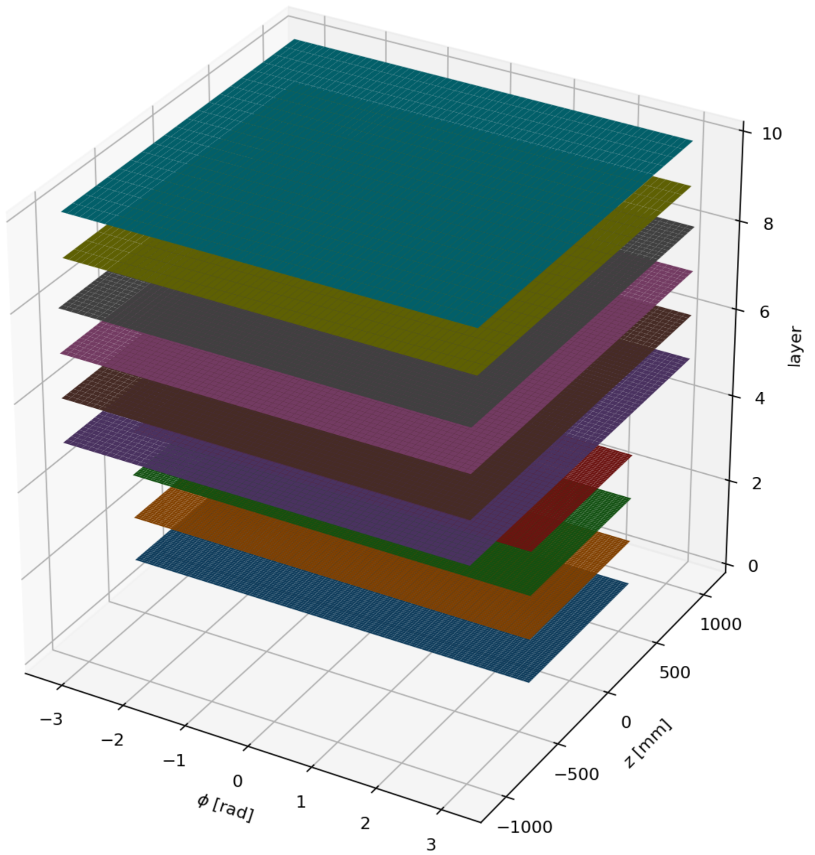
Video object tracking

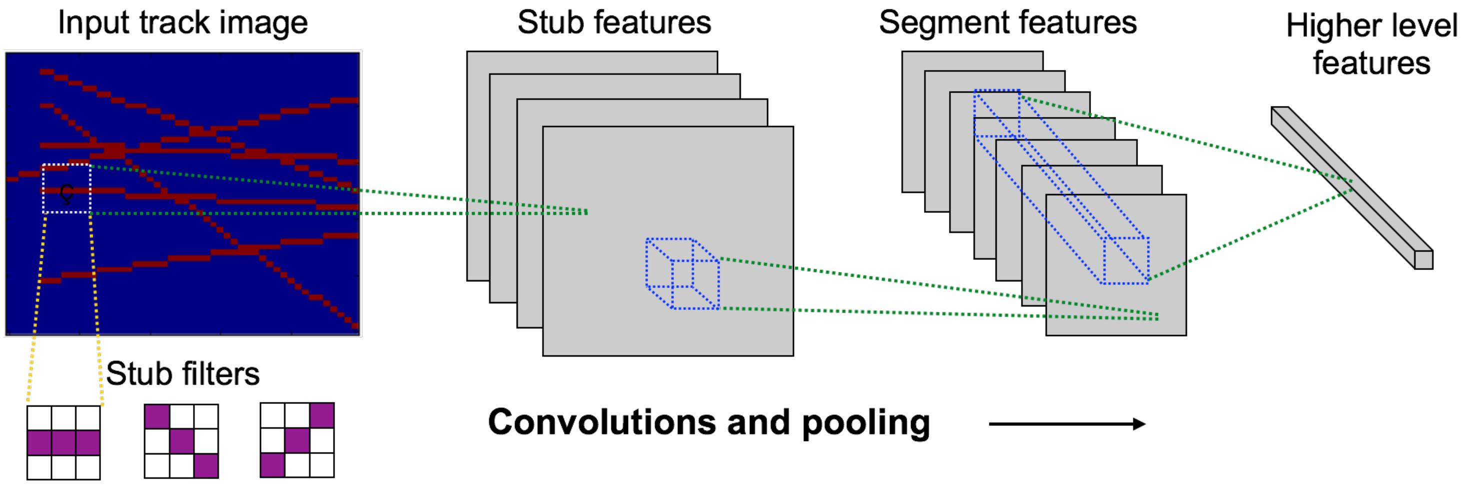


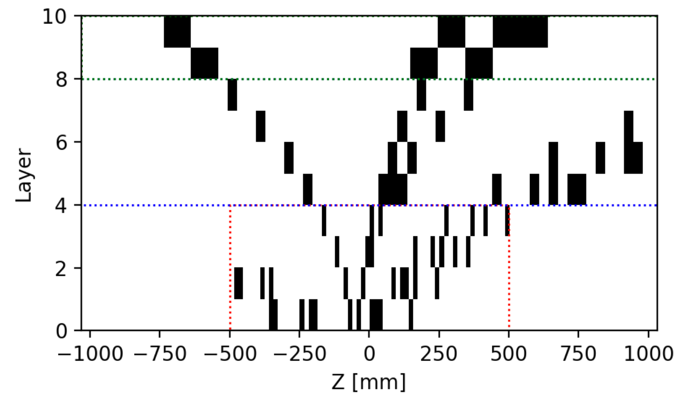
https://arxiv.org/abs/1604.03635

Image representations

- Unroll cylindrical detector layers
- Treat as multi-channel image
- Apply convolutional and recurrent neural networks

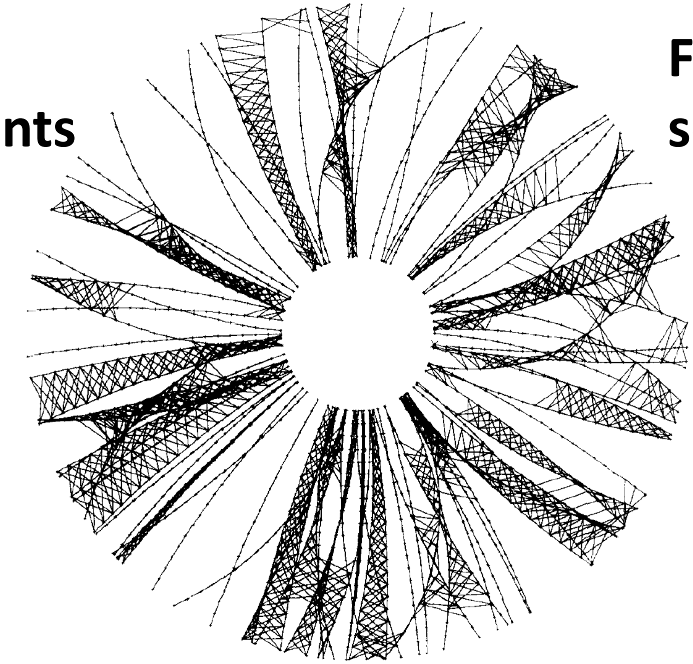




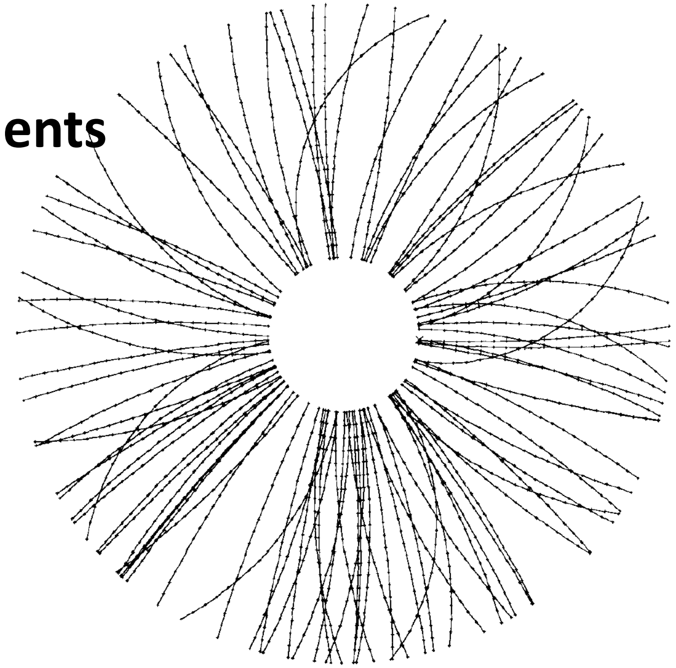


Hopfield networks for tracking (~1990)

Input segments



Final segments



$$E = -\frac{1}{2}\left[\sum_{kln} T_{kln} V_{kl} V_{ln} - \alpha\left(\sum_{kln(n \neq l)} V_{kl} V_{kn} + \sum_{klm(m \neq k)} V_{kl} V_{ml}\right) - \beta\left(\sum_{mn} V_{mn} - N_{\mathrm{a}}\right)^2\right]$$

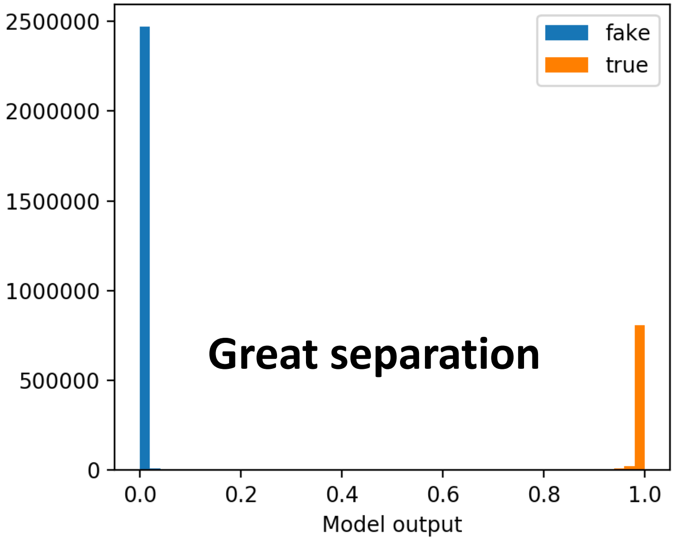
- Identify true segments in a graph of connected hits
- No learned parameters, but solved via annealing with an energy loss function

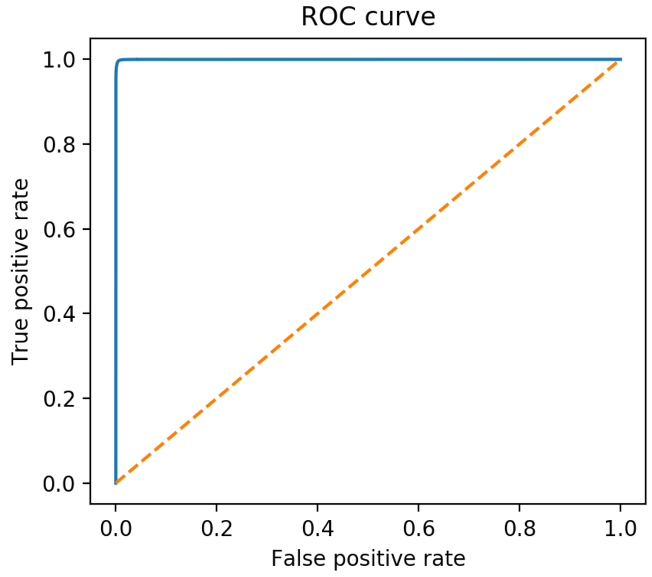
https://www.sciencedirect.com/science/article/pii/0010465588900045
https://www.sciencedirect.com/science/article/pii/0168900289913004
https://www.sciencedirect.com/science/article/pii/001046559190048P

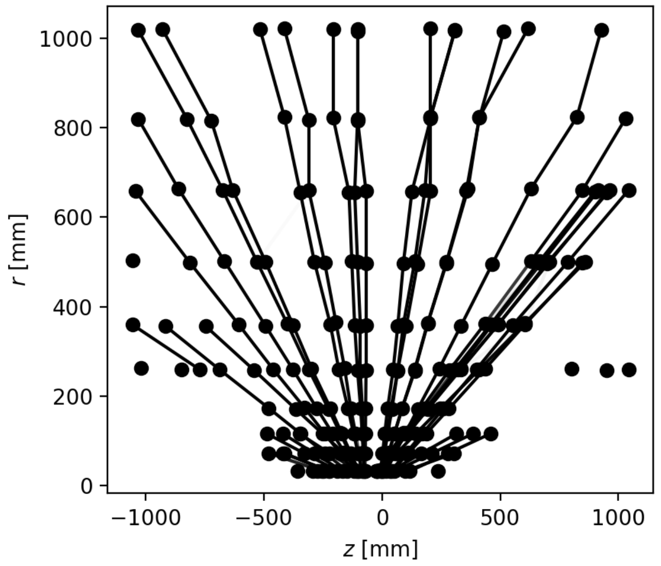
Segment classification

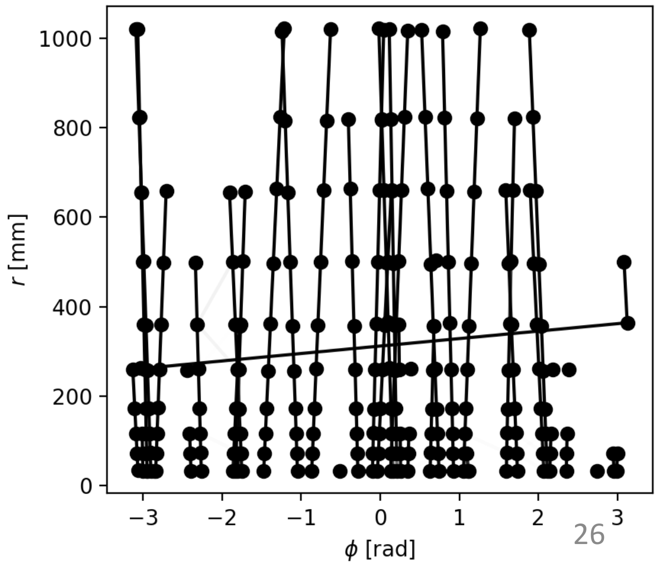
- 4-layer model with 7k parameters
- Performs well, with good purity and efficiency

```
Test set metrics
Accuracy:   0.9952
Purity:     0.9945
Efficiency: 0.9870
```



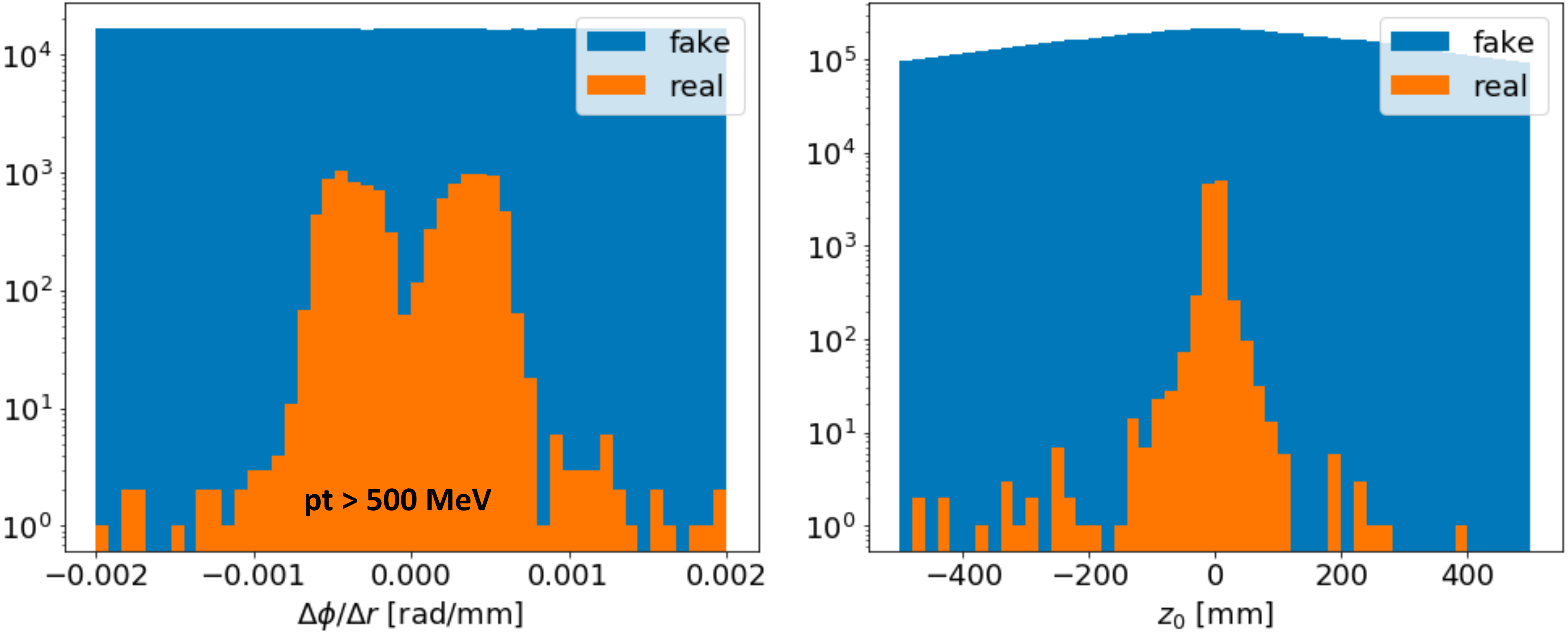






Building the graph

- **Select initial edges (doublets) by cutting on slope in phi-r and on z_0**



Low density

Truth cuts

- pt > 1 GeV

Doublet selection

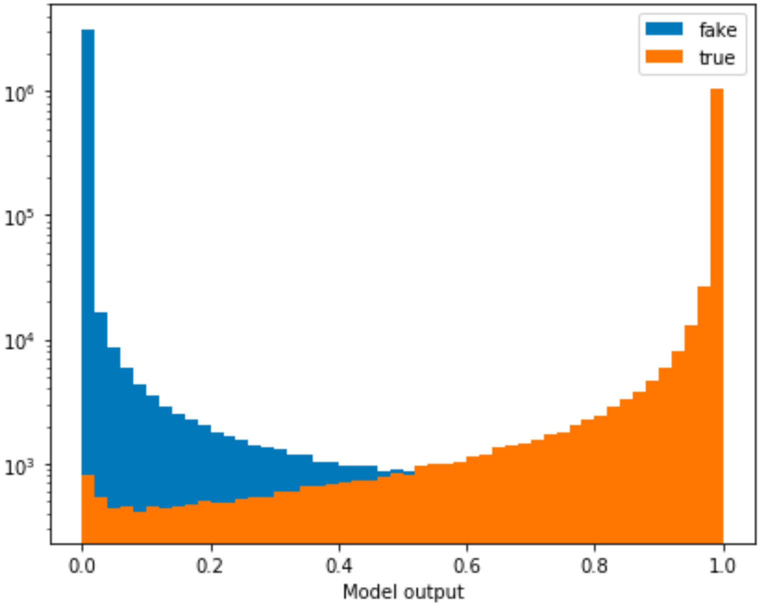
- phi slope < .001

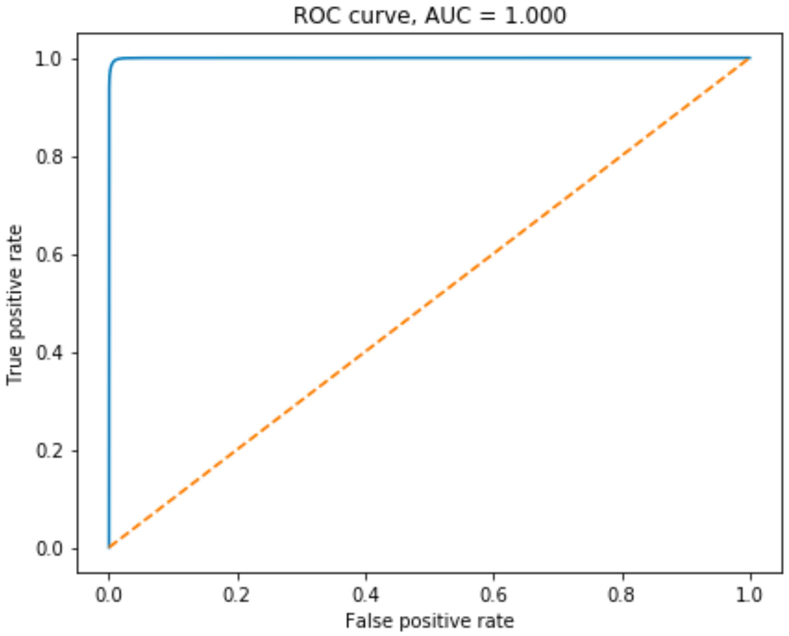
- z0 < 200mm

- 99% efficient, 33% pure

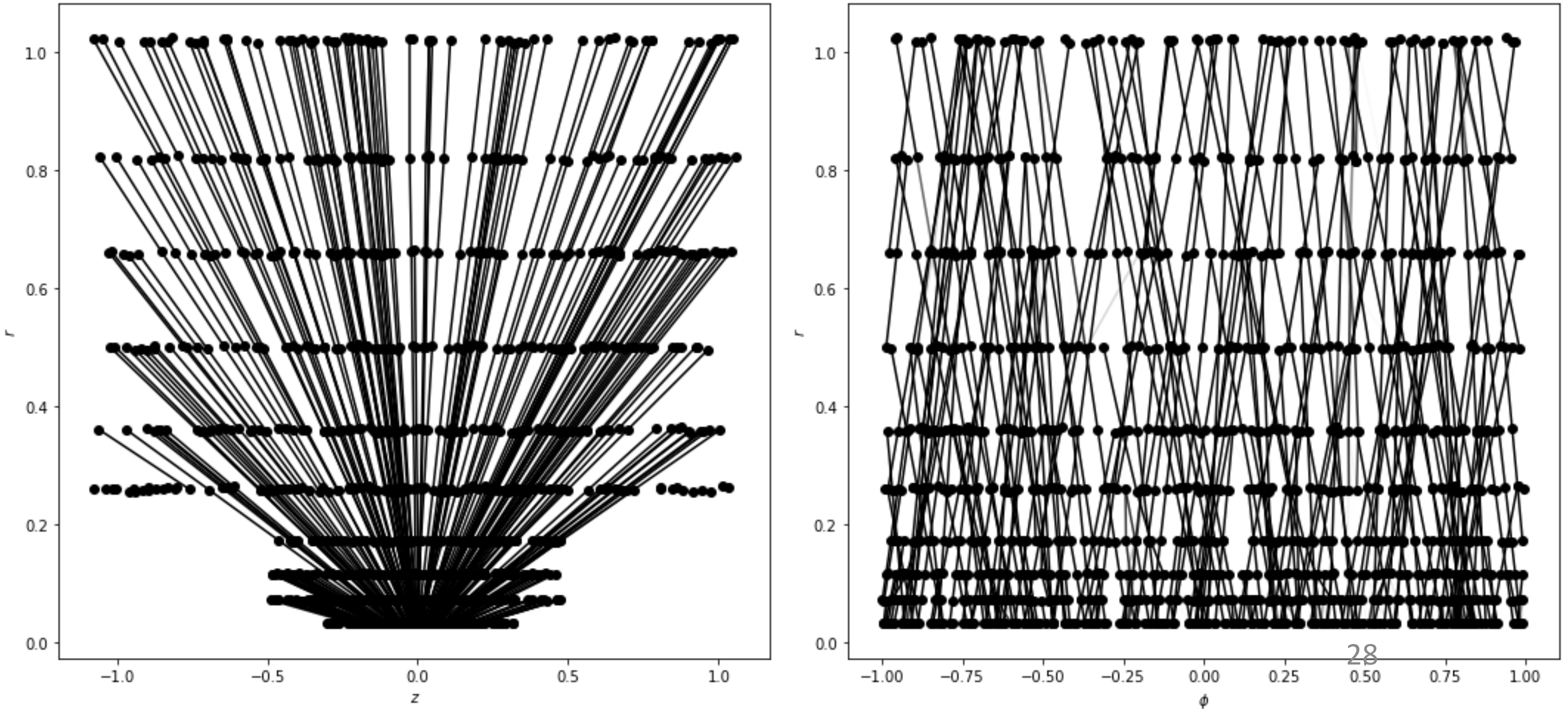
Segment classification

```
Accuracy:   0.9932
Precision:  0.9866
Recall:     0.9872
```





4 detector sections


r
z
−1.0
−0.5
0.0
0.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
r
ϕ
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00