ML for Tracking at JLab

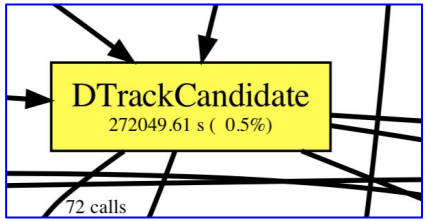
Feb. 12, 2019

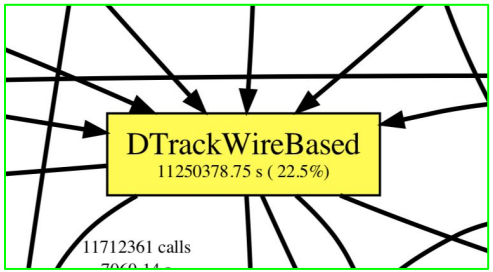
David Lawrence (presenter),

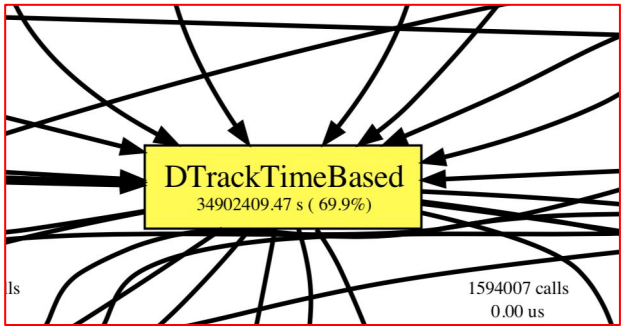
M. Diefenthaler, G. Gavalian, D. Romanov.

Motivation:

The largest CPU resource driver for event reconstruction is charged particle tracking.







GlueX TRACKING CPU Usage:

DTrackCandidate: ~2.0%
DTrackWireBased: 22.5%
DTrackTimeBased: 69.9%

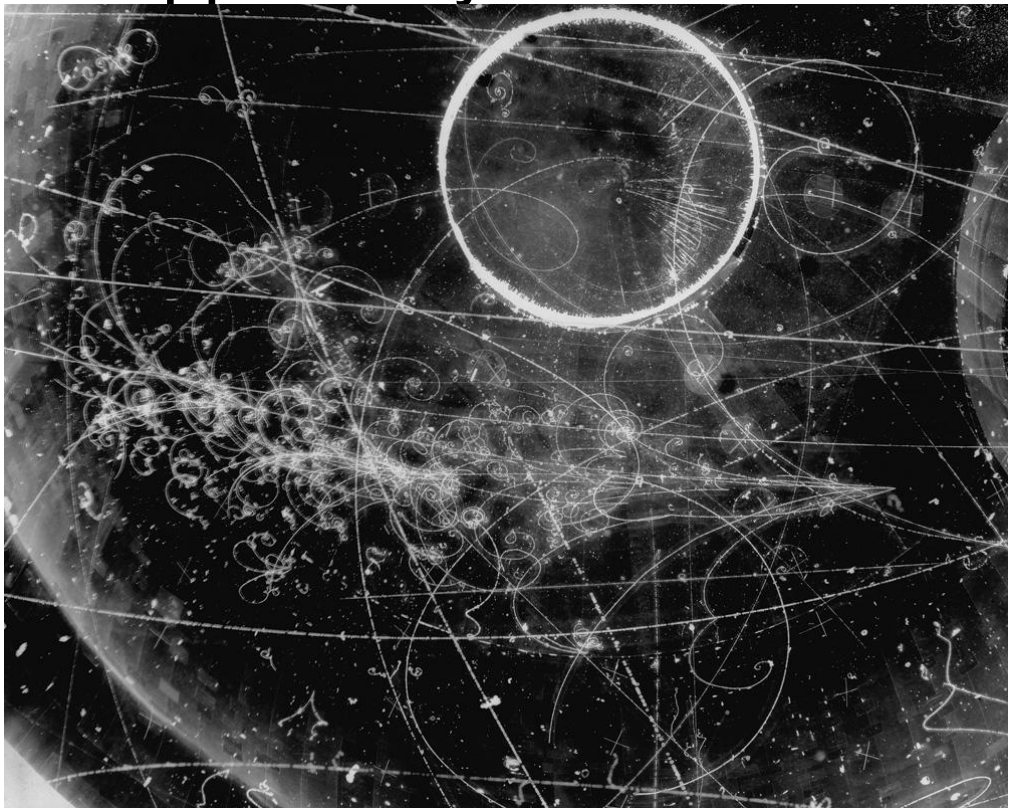
Tracking Total: 94.4%

- Hall-B requested a 50M unit from NERSC for 2019.
- Roughly **20M core-hours** (80 units/hour on 32 core node)
- At *$0.06793/core-hour * this amounts to ~$1.3M*
- Additional CPU will be used on JLab farm and on OSG for reconstructing simulated data.
- Hall-D has similar numbers.
- Total cost per year for reconstruction computing is O($10M)

** https://acg.umaine.edu/pricing/fee-structure/*

ML and Tracking: A new opportunity

- ML is ideal for automated recognition of patterns and regularities in data.
- Tracking in HEP and NP is classical pattern recognition problem that until recently has been solved in separate steps not involving ML methods.
- ML has the potential to revolutionize tracking.
- Example from NOvA: "*It improved the headline analysis performance by 30%, equivalent to an equipment savings of approximately $72 million.*" (Dr. Aristeidis Tsaris (FNAL), Computing Round Table 11/18)



Bubble chamber film, analyzed by manual pattern.

Expected Benefits of ML

- Data Reduction:
  - If the events with no tracks are identified during writing of the data, the data volume will be reduced significantly.

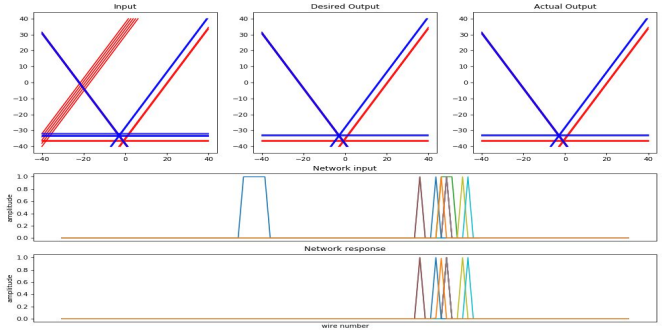
- Tracking Speed:
  - If we can match crosses to the right tracks, it will eliminate need for combinatorics.
  - Especially for high luminosity runs this will reduce tracking time for up to 40%.

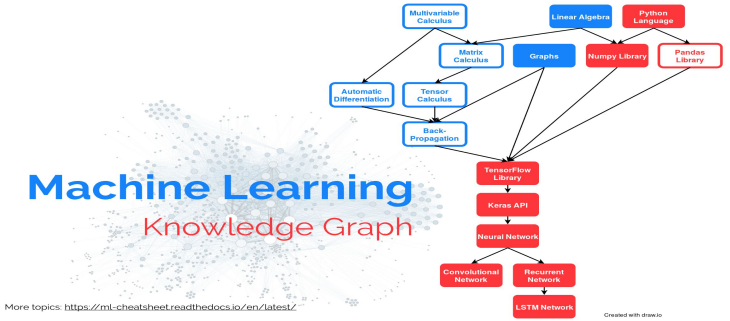
- Tracking Speed (more):
  - If we can calculate state vectors from the pattern in the drift chamber, this will reduce number of iterations needed for Kalman-Filter.
  - We might even be able to recognize tracks from hit patterns and replace track finding and fitting algorithms with a ML algorithm.
  - Potentially very big gain in speed (don’t have estimates yet)

Wouter Deconinck & Markus Diefenthaler ML Initiative

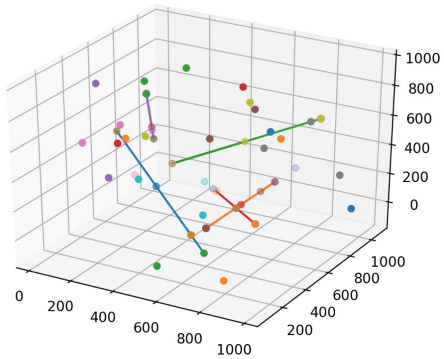
using JupyterHub at https://jupyter.jlab.org/

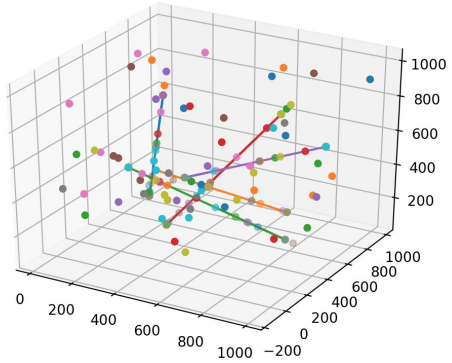
Track finding in JLAB 12 data





Track finding in toy data

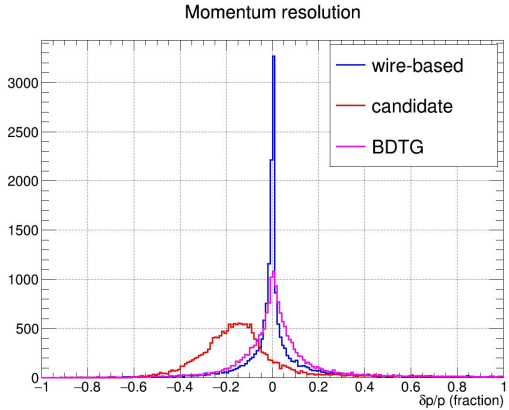


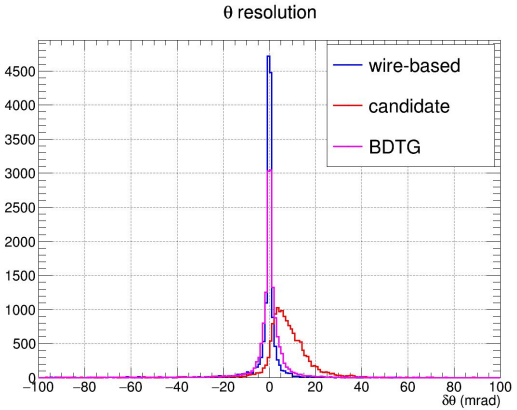


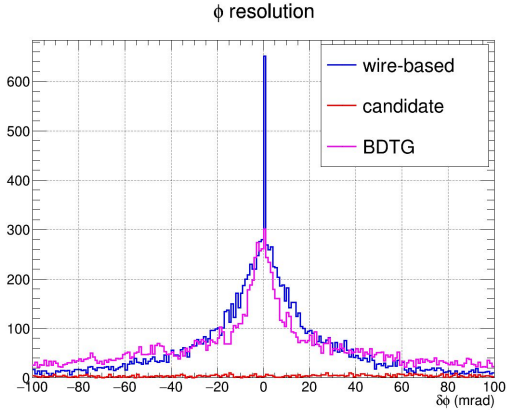
TensorFlow (TF)

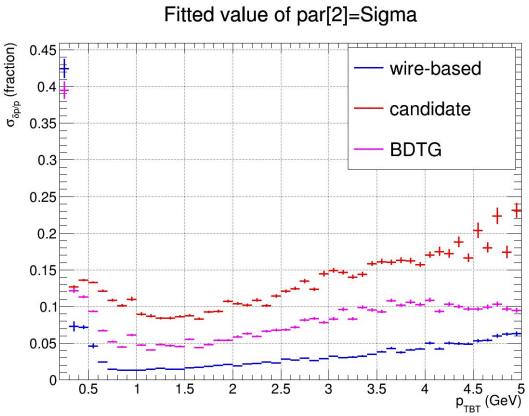
- Repository of Jupyter notebook TF tutorials for beginners: https://github.com/Hvass-Labs/TensorFlow-Tutorials
- Repository of Jupyter notebook TF tutorials for beginners: https://github.com/nlintz/TensorFlow-Tutorials
- Written tutorial for beginners on the basics of TF: https://www.datacamp.com/community/tutorials/tensorflow-tutorial
- Written tutorial for beginners on the basics of TF: https://hackernoon.com/machine-learning-with-tensorflow-8873fdee2b68
- Paper describing the internal workings of TF: https://arxiv.org/pdf/1610.01178.pdf

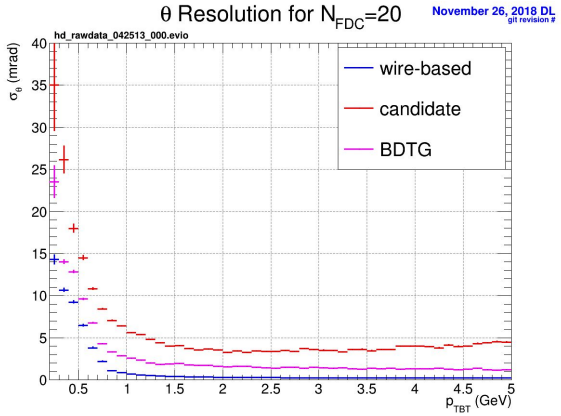
N_{FDC}:20 -- PID: π- -- Input Parms: Candidate p3; Pseudo X,Y,Z

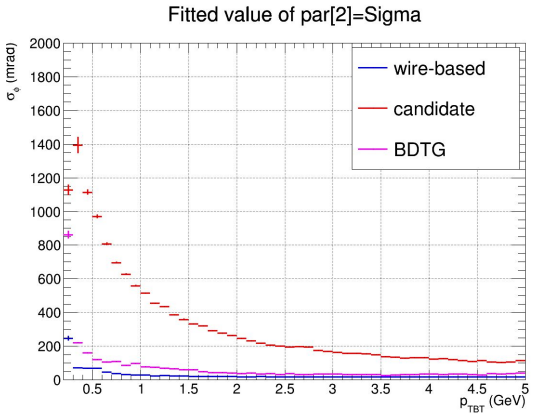








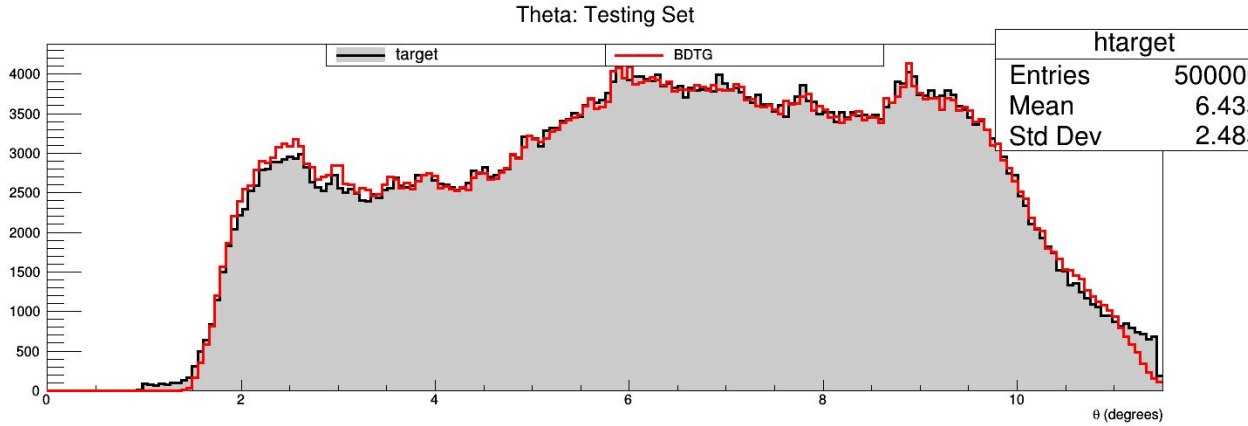


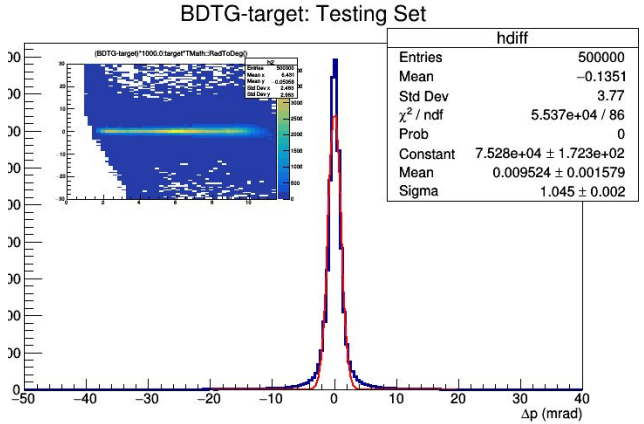


BDTG θ

Train BDTG on zenith angle

- all TBT with N_{FDC}==24
- Inputs:
  - DFDCPseudo X,Y
- 0.200 <= p <= 5 GeV/c
- 500k events
- run 42513





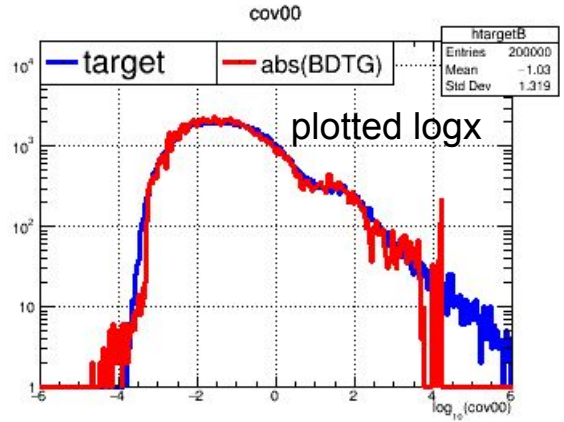
Covariance Matrix Elements: Trained on State vector

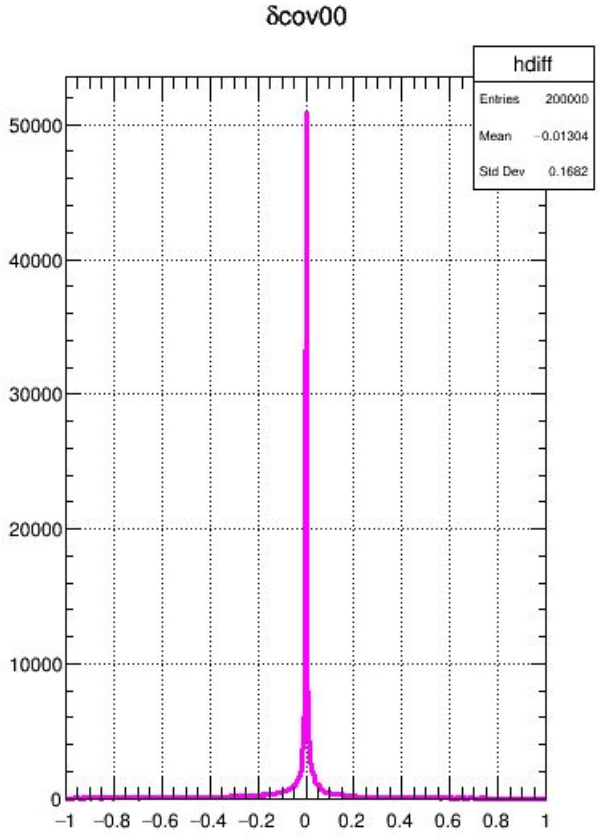
GlueX:

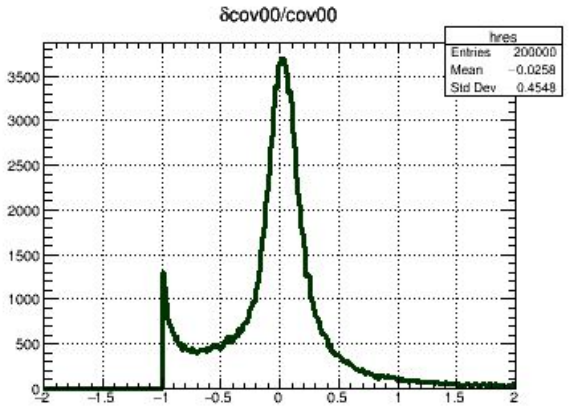
Single event per core: ~225ms

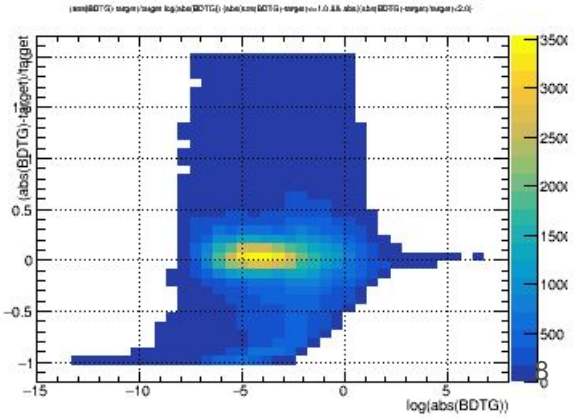
Covariance matrix: 15 elements

Single track full covariance matrix: ~300-350us or ~6ms/event





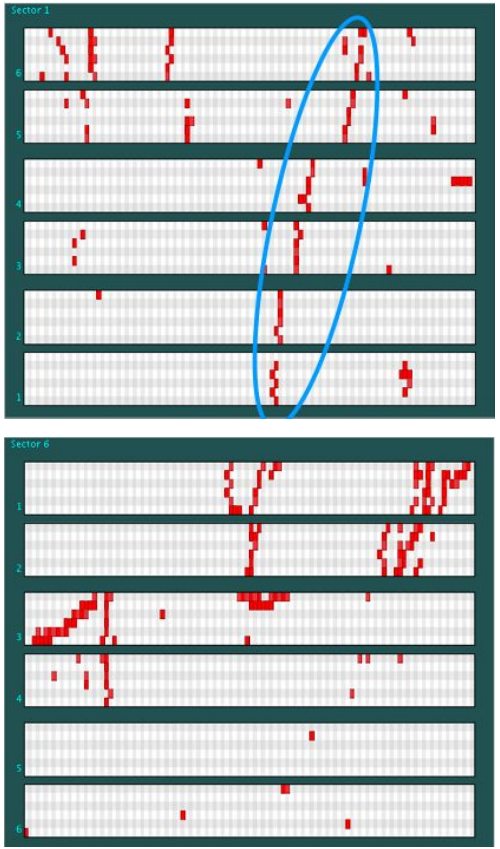




Track Identification (data cleanup)

Identifying existence of a track in a sector can be used as Level 3 trigger, about 30% of events in current data have no tracks in either sector.

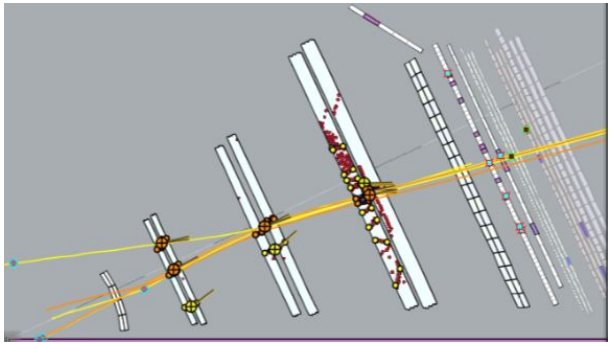
Even in events where some sectors have tracks, some of them contain only noise hits (proportional to luminosity), dropping data from sectors with no potential tracks can reduce data even further. Drift chamber is 32% of recorded data.

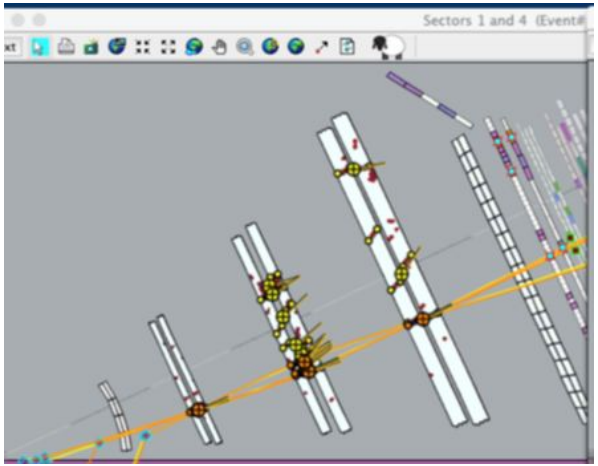


Track Combinatorics (processing speed)

The reconstructed segments in one sector are combinatorially iterated to find those that form a track, with ML noise segments can be discarded.

In case of two tracks in one sector identifying segments that belong to each track will help reducing combinatorics and significantly reduce Kalman filter iterations over track candidates.





How to proceed

- Repository created on github for Multi-hall use to develop code and samples

  https://github.com/JeffersonLab/trackingML

- Draft plan for file formats being developed collaboratively
- See if we can learn from HEP, e.g., TrackML Particle Tracking Challenge
- Time scale unknown, best effort approach needs help from ML trained scientists