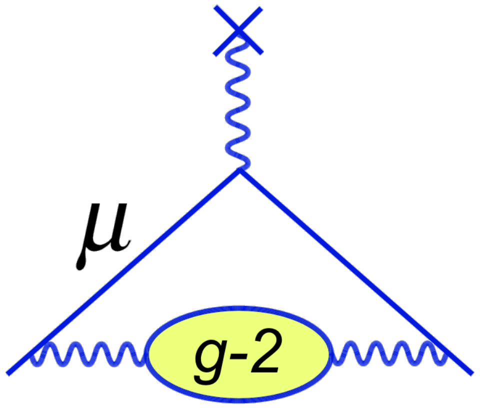
Machine Learning group for g-2 Tracking

Manolis Kargiantoulakis
03/02/2022
Muon g-2/SCD Computing Workshop



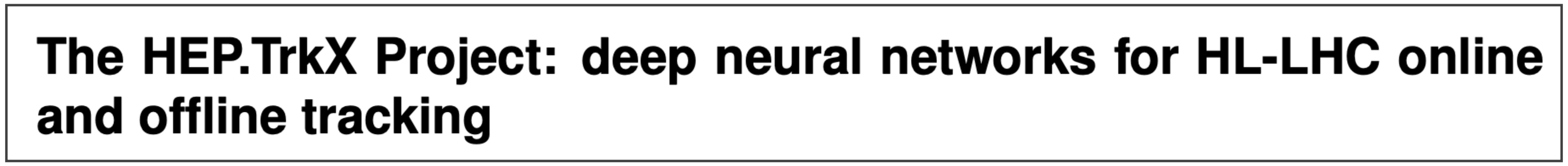
The g-2 Machine Learning group

- The g-2 Machine Learning (ML) group is a recent effort to:
  - Find applications of ML tools to empower g-2 analyses
  - Build ML expertise and help more groups use these powerful techniques

- We have started by focusing on tracking applications
  - Could be good fit for ML techniques, with a lot of available literature
  - There could be significant room for improvement in tracking algorithms
    - And such improvement would be significantly beneficial, especially for the EDM measurement

Track finding

- Among other projects, I will focus on one that we have invested most effort
- Track finding: selecting hits that belong to same track
  - A challenging pattern recognition problem, for which NNs could be a good fit
- We started based on work from the HEP.TrkX project, which includes Fermilab SCD scientists
  - They have been very welcoming, even invited us to Exa.TrkX meetings
  - Though our targets diverge, since we don’t need to push for very high performance that would be suitable for HL-LHC



The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking

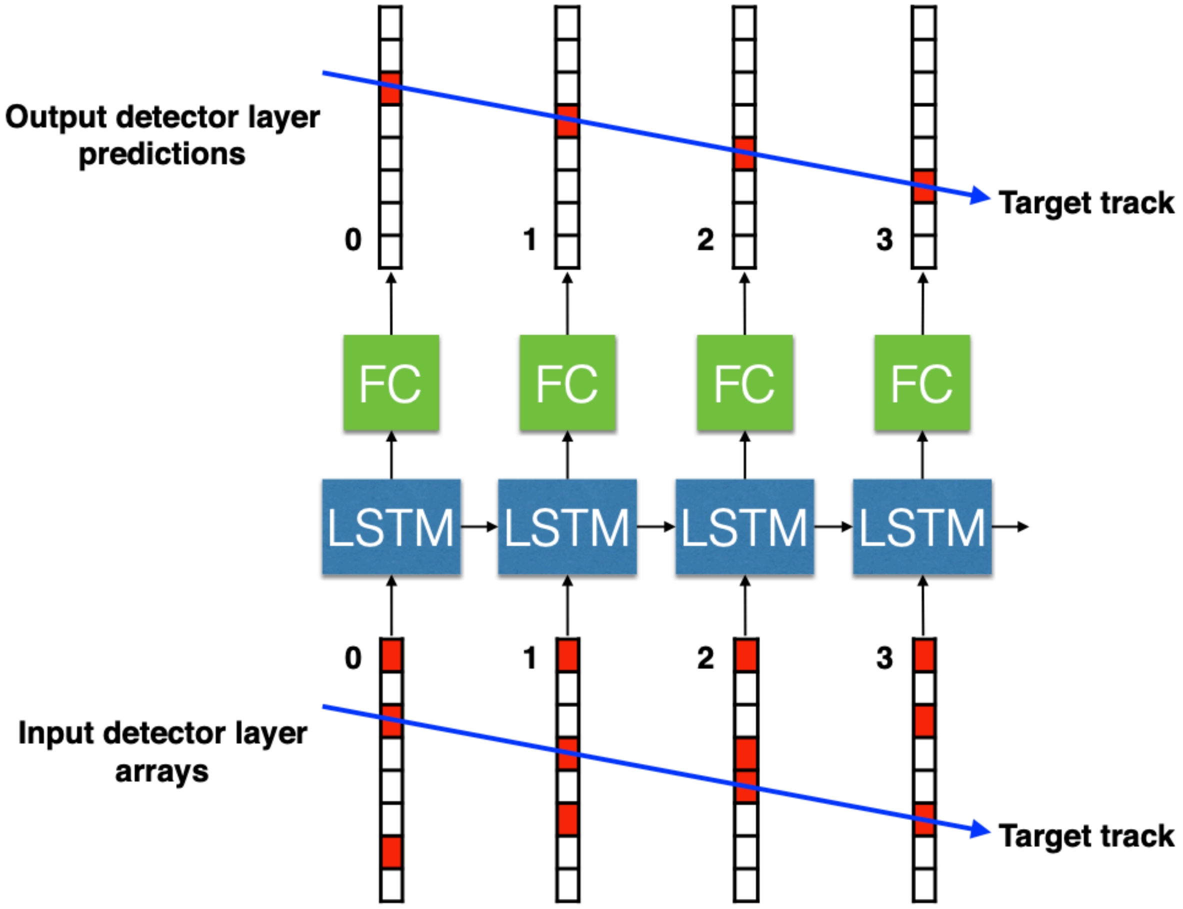
https://doi.org/10.1051/epjconf/201715000003

Track finding model

- Our track finding model is an evolution of one of the original HEP.TrkX studies

- Model overview:
  - RNN with LSTM layers, treating detector layers as successive time steps
  - Followed by time-distributed fully-connected layers that predict hits associated with a track

- We have evolved this design for our application


Output detector layer predictions
Target track
0
1
2
3
FC
FC
FC
FC
LSTM
LSTM
LSTM
LSTM
0
1
2
3
Input detector layer arrays
Target track

The HEP.TrkX Project: deep neural networks for HL-LHC online and offline tracking

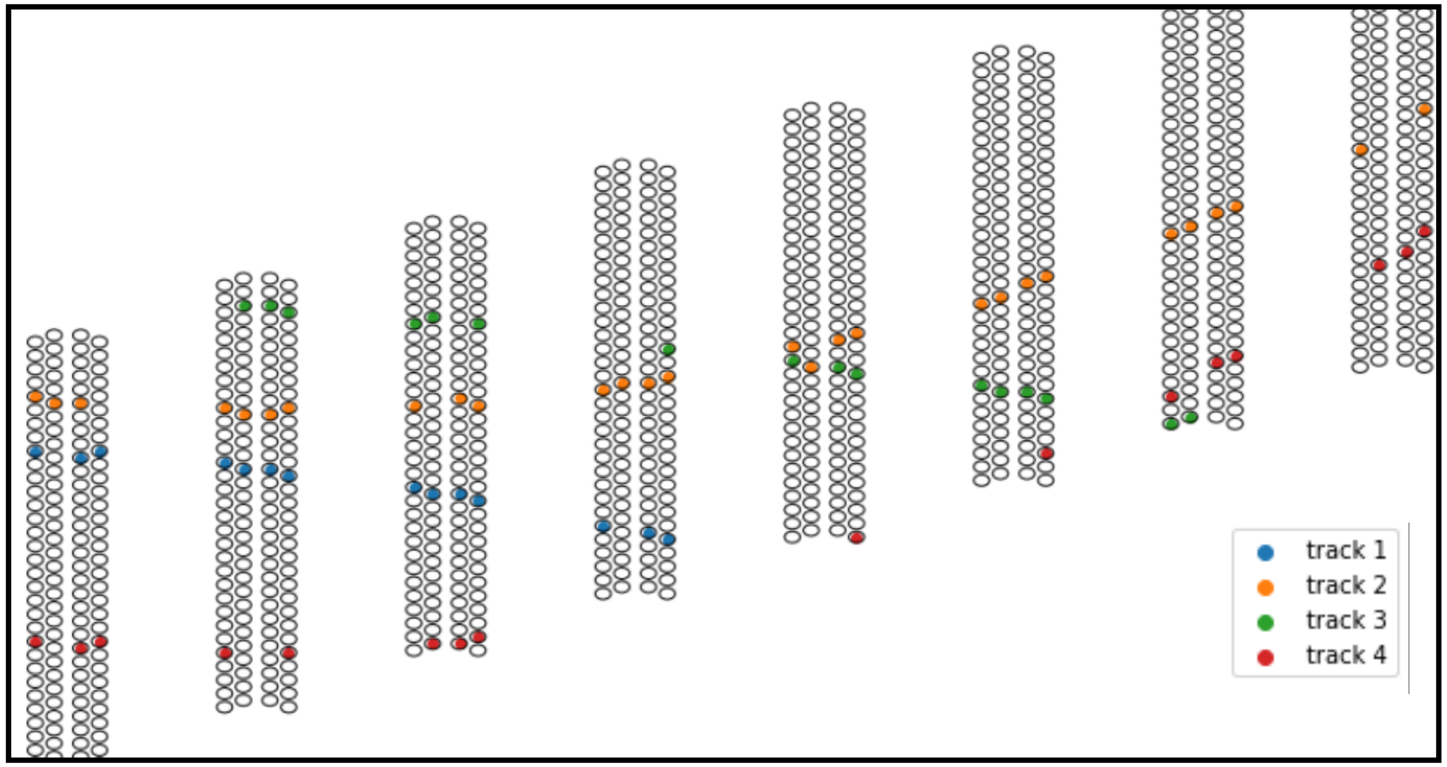
https://doi.org/10.1051/epjconf/201715000003

Track finding model

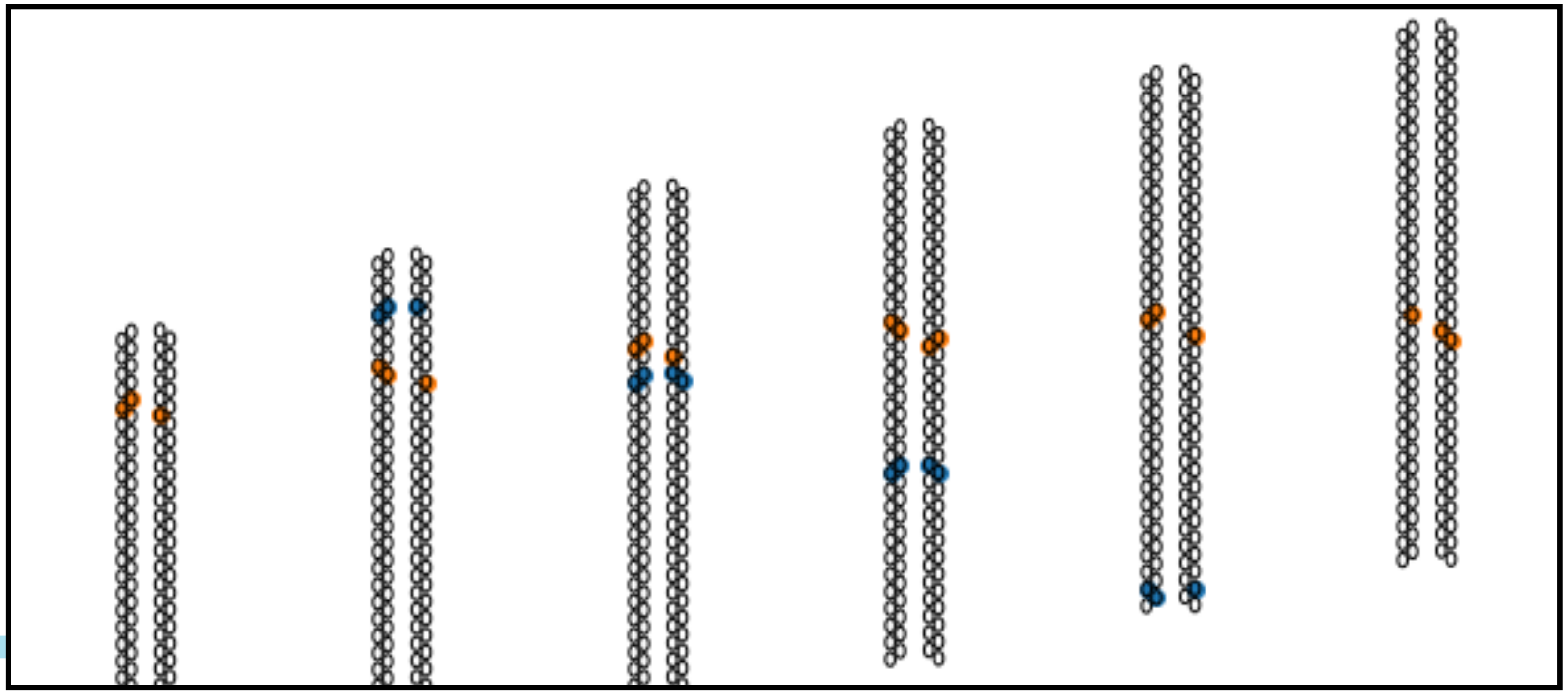
- Model input: All tracker hits within a 100 ns “time island”
- Output: Hits clustered into track candidates

Event prediction examples:

High-occupancy event in 2D simulation



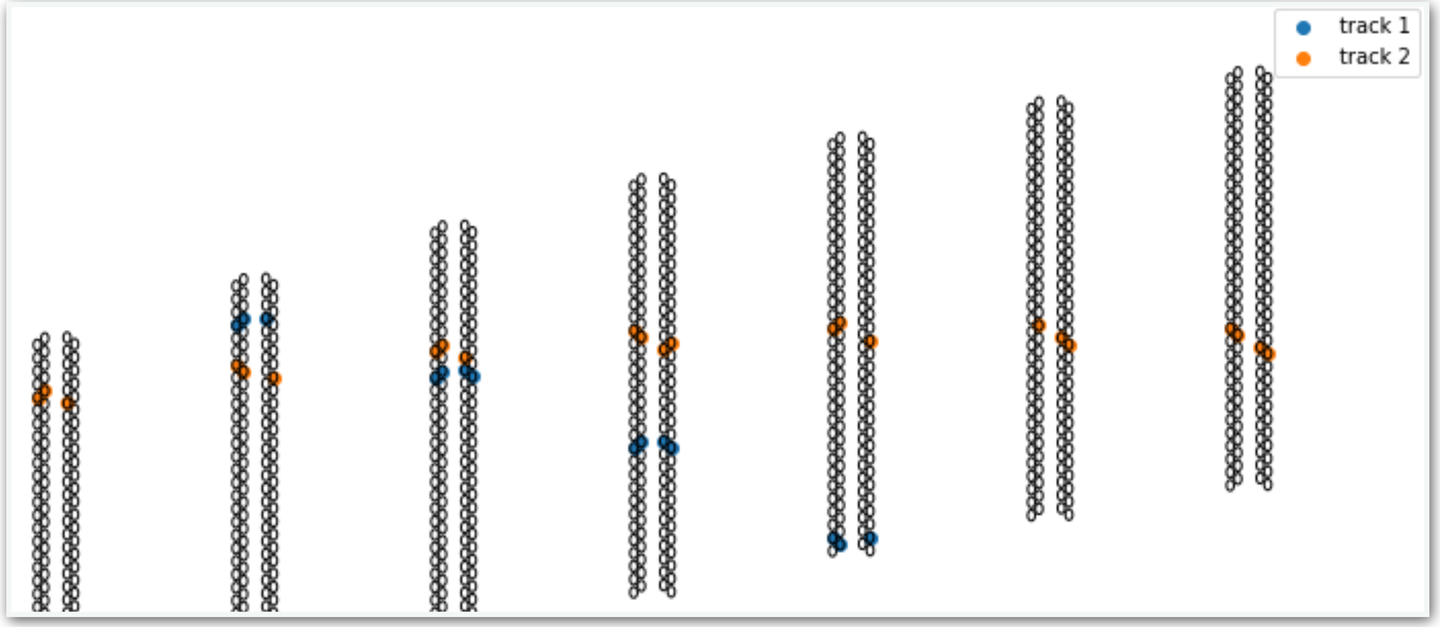
Synthetic event from real isolated tracks



Data augmentation

- Our collaborators from Liverpool have been instrumental in developing synthetic datasets

- Art module to:
  - Find isolated (late-in-fill), well-reconstructed track candidates from tracked data
  - Overlay them into a synthetic event
  - Simulate noise/crosstalk digits



- This synthetic dataset has been invaluable for realistic supervised training, and extraction of performance metrics

Interfacing with the gm2 framework

- Our ML code is written in python3 and Tensorflow
- Still focusing on developing models and characterizing performance
- Interfacing with the gm2 framework has been ad-hoc so far
  - Eg with the dedicated module to produce synthetic events, which are then imported to the python code

- If we can demonstrate significant performance improvement, then we will work on integrating the python ML code into tracking production
- Some potential issues we expect around that work:
  - Integrating Tensorflow and python3
    - TF currently installed in user directory on g-2 machine; and gm2 framework built against python2
  - Compiling python, too slow to run interpreted

Summary, outlook

- We are building Machine Learning expertise in Muon g-2
  - For now focusing on tracking applications

- Recent effort but mature model and workflow development
  - Hoping to demonstrate improved performance in dedicated pieces of the tracking chain

- When that happens, our next big target will be integration with gm2 framework