A(I)LERT Track Finding

Mathieu Ouillon (Mississippi State University)

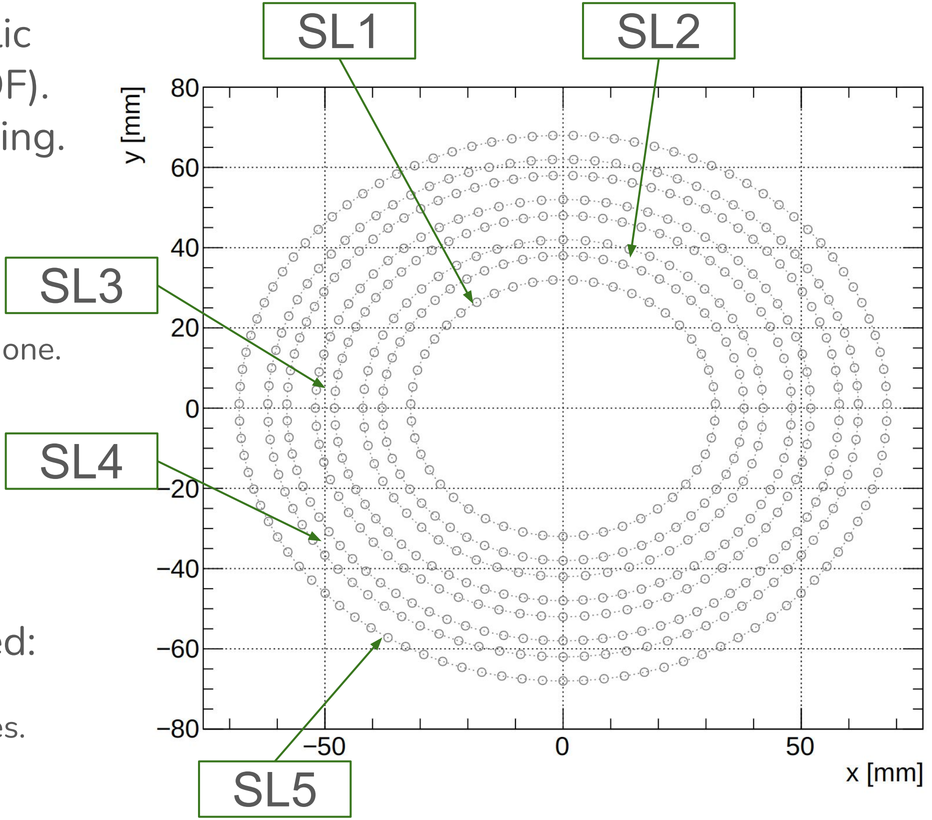
CLAS Collaboration (November 12th, 2024)

The ALERT Detector

- ALERT comprises two sub-detectors: A Hyperbolic Drift Chamber (AHDC) and A Time of Flight (ATOF).
- Focus on the AHDC used for track finding and fitting.

- The AHDC is composed of:
  - 5 superlayers with 2 layers each except for the first and last one.
  - Each layer has a given number of cells.
  - A cell is 6 field wires and 1 signal wire.

- A stereo angle of 20° between each superlayer.
- AHDC is composed of 576 signal wires distributed:
  - 47, 56, 57, 72, 87, 99 for each superlayer.
  - Layer in each superlayer has the same amount of signal wires.

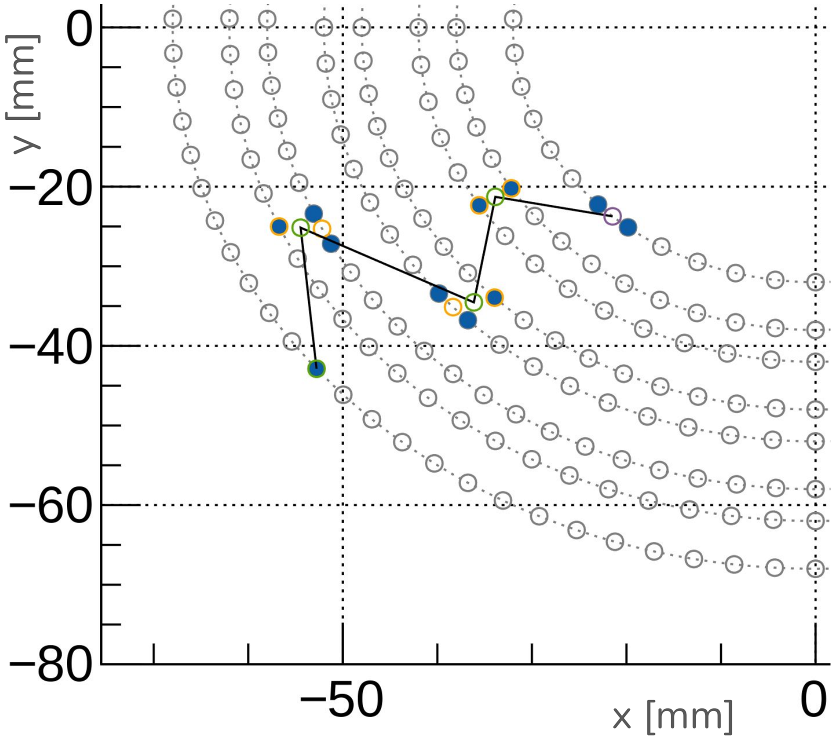


Clustering

- The first step before the model is clustering. Goal: reduce the combinatorics to find track.

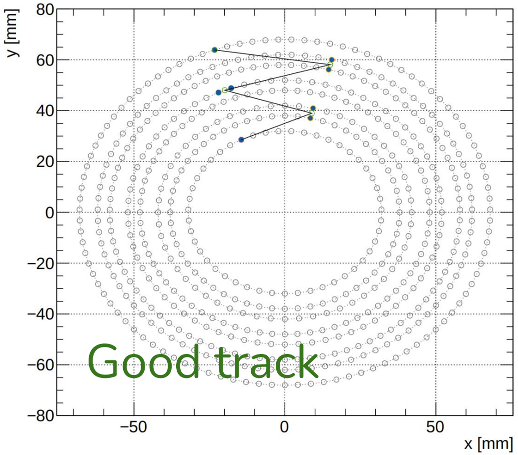
- Test differents clustering:
  - Raw hits in blue.
  - Merge hits that on the same layer and 1 wire away together into precluster (orange circle).
  - Merge precluster that are on the same superlayer and 8mm away into super-precluster (green circle).
  - End up with 5 super-preclusters.

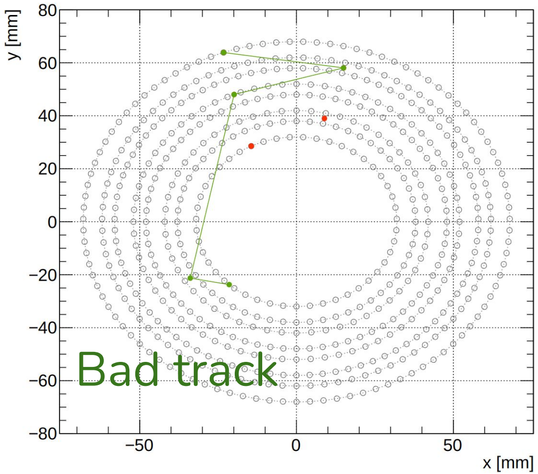
- Look for a track with 5 super-clusters.
- Test the AI model with preclusters and super-preclusters.

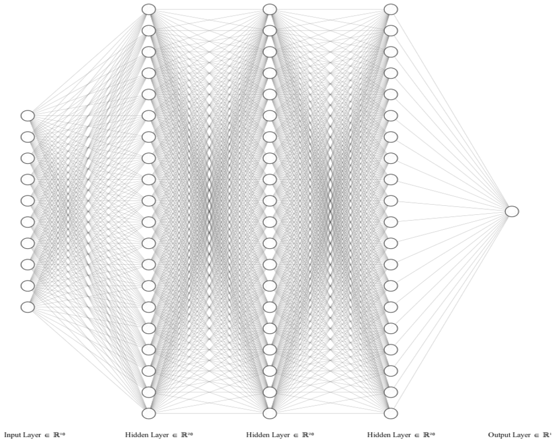


Model: MLP

- The input for the model is 5 combinations for x and y which represent the super-preclusters.
- Good tracks are generated by GEMC:
  - proton with p ∈ [60, 250] MeV/c, φ ∈ [0, 360]°, θ ∈ [60, 120]° and vz ∈ [-15, 15] cm.
- False tracks: interchanging randomly up to two super-precluster, with a adjacent event.
- Model: MLP 10 inputs, 3/5 hidden layer (20/100), 1 output.
- Output of the model: Probability 1-good track or 0-bad track





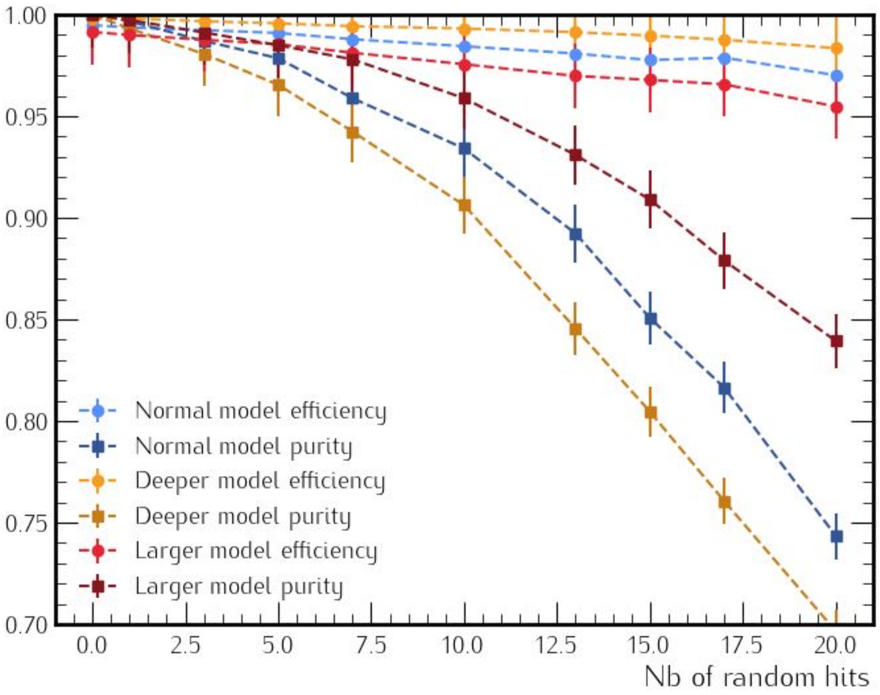


Efficiency and purity as function of occupancy

- Efficiency: Number of good tracks classified as good / number of event.
- Purity: Number of good tracks classified as good / number of tracks (good or bad) classified as good.

- To compute the efficiency/purity: Add some random hits in the drift chamber and generate all possible tracks per event.
- Expect around 15 random hits per events (5% occupancy).

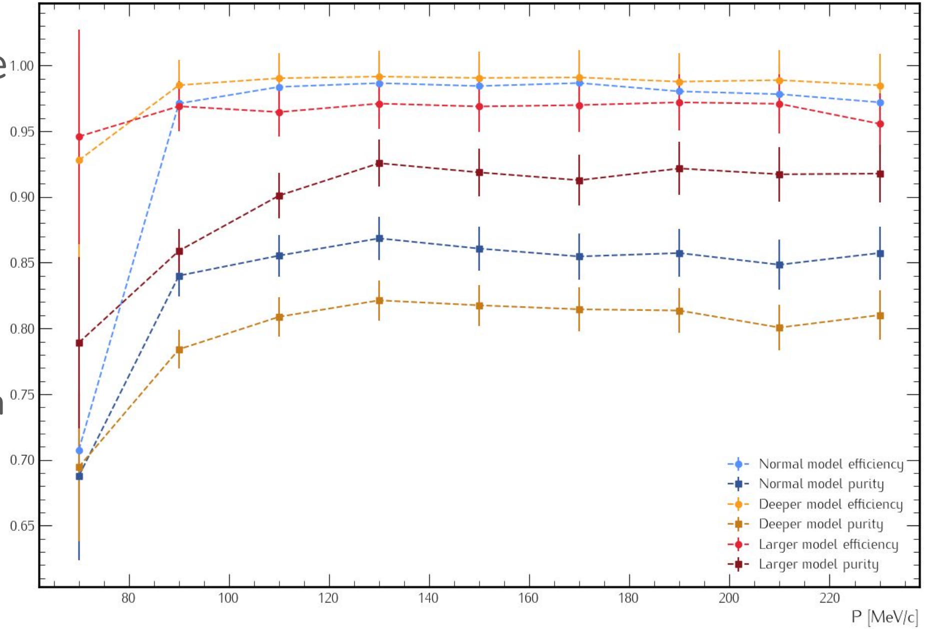
- Blue line/dot: model with 20 neurons in the 3 hidden layers
- Red line/dot: model with 100 neurons in the 3 hidden layers
- Orange line/dot: model with 20 neurons in the 5 hidden layers



Efficiency and purity as function of momentum

- Can also compute the efficiency as function of the momentum of the proton:
  - Using event with 1 track + 15 random hits
  - proton with p ∈ [60, 250] MeV/c, φ ∈ [0, 360]°, θ ∈ [60, 120]° and vz ∈ [-15, 15] cm

- Efficiency and purity almost constant for the range of momentum.

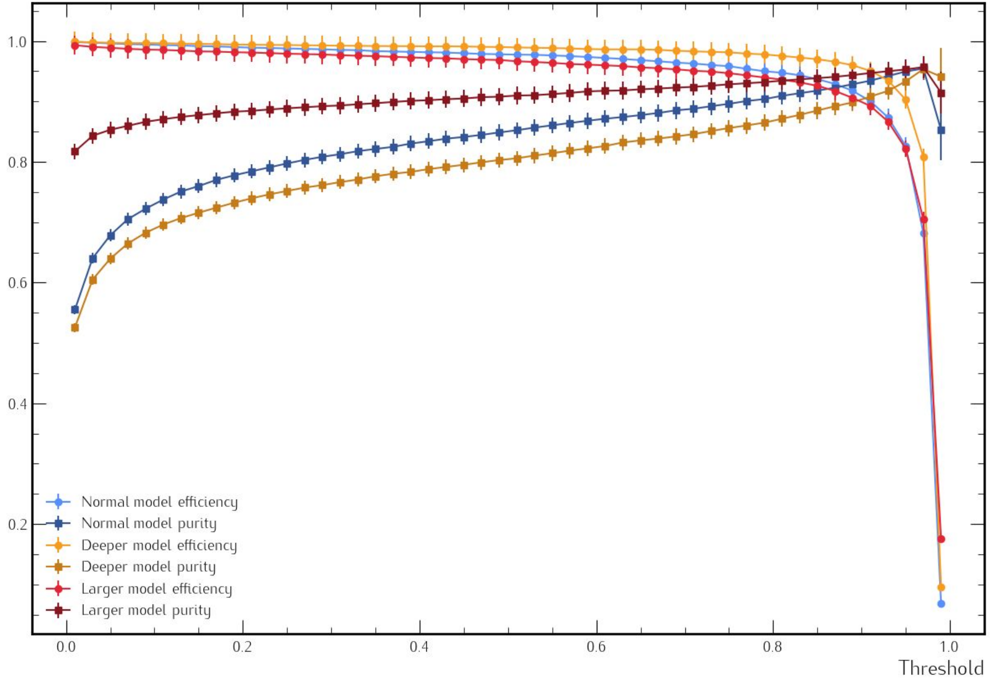


- Blue line/dot: model with 20 neurons in the 3 hidden layers
- Red line/dot: model with 100 neurons in the 3 hidden layers
- Orange line/dot: model with 20 neurons in the 5 hidden layers

Efficiency and purity as function of threshold

- Can look at the efficiency as function of the threshold.
- Threshold: value above what the we consider the output of the model 1 and below 0.

- Want high efficiency and purity, but not possible.
- Prefer high efficiency to high purity.

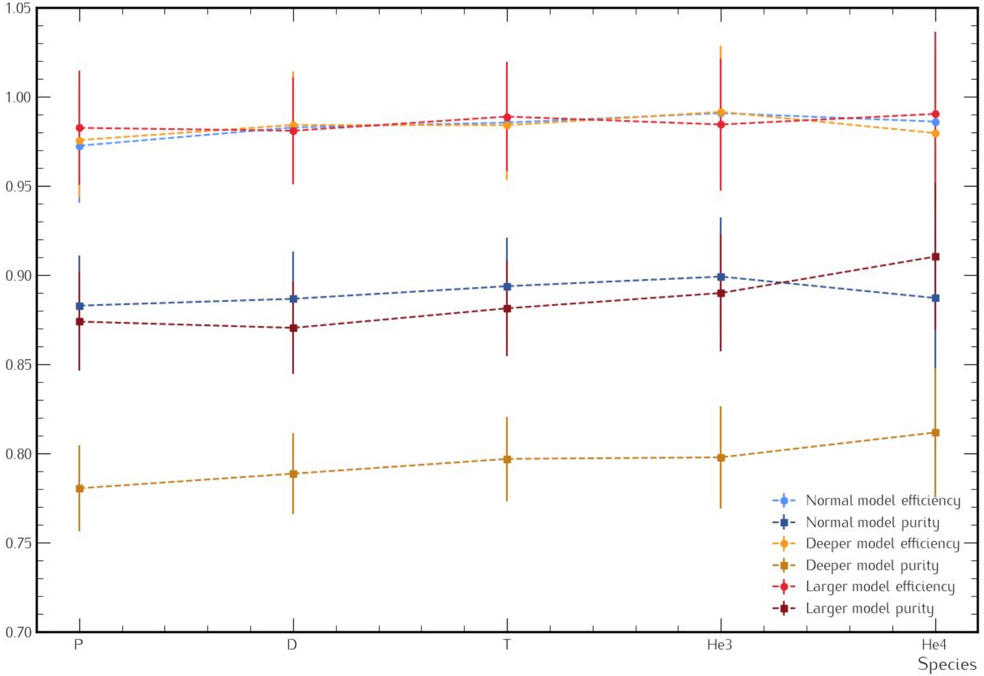


- Blue line/dot: model with 20 neurons in the 3 hidden layers
- Red line/dot: model with 100 neurons in the 3 hidden layers
- Orange line/dot: model with 20 neurons in the 5 hidden layers

Efficiency and purity as function of species

- Can look at the efficiency as function of the particle.

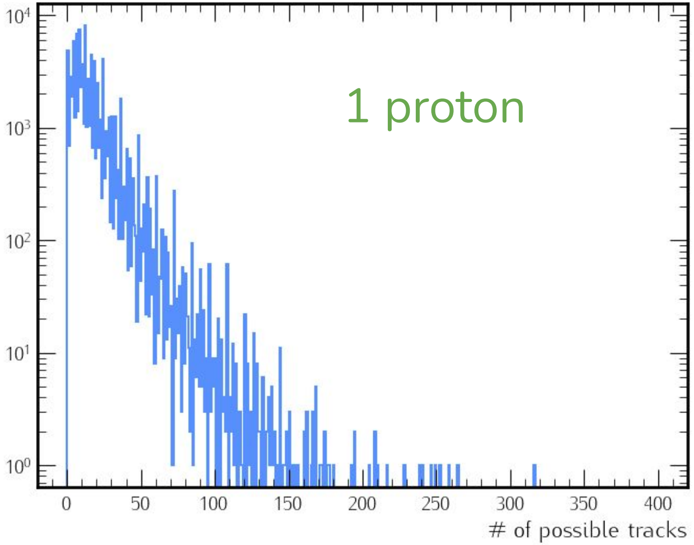
- Train the model with tracks from proton, deuterium, tritium, helium 3 and helium 4.
- Constant for all species. Purity of the deeper model is lower.

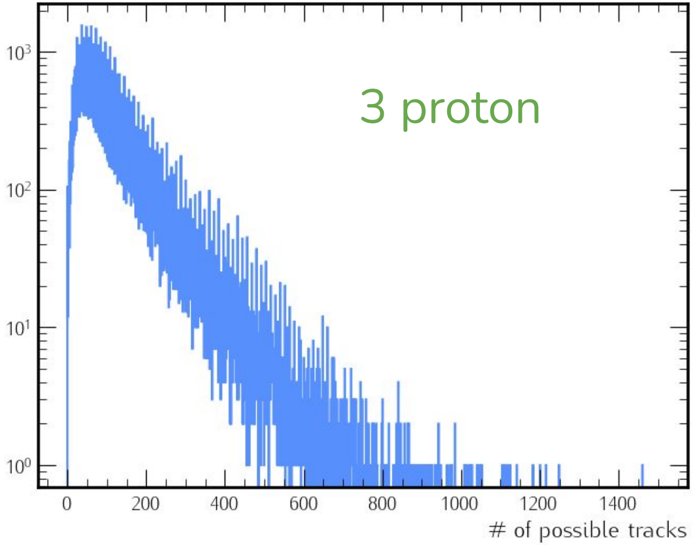


- Blue line/dot: model with 20 neurons in the 3 hidden layers
- Red line/dot: model with 100 neurons in the 3 hidden layers
- Orange line/dot: model with 20 neurons in the 5 hidden layers

Inference time and number of possible tracks

- For the normal model, the inference time is around 15μs ↦ 60KHz.
- For a typical event, with 1 proton track and 15 random hits:
  - Mean of 16 possible tracks but can have up to 300 possible tracks for some events
- For a busy event with 3 proton tracks and 15 random hits:
  - Mean of 100 possible tracks but can have up to 1500 possible tracks for some events (need a threshold)





Summary and Outlook

- Develop an MLP for track finding for ALERT
  - New clustering to reduce the number of possible tracks
  - Evaluated the model's efficiency and purity as a function of a number of random hits added.
  - Evaluated the model's efficiency and purity as a function of momentum / threshold
  - Evaluated the model's efficiency and purity for differents particles
- Remaining tasks:
  - Compare AI to conventional track finding
  - Use luminosity simulation to have a more realistic occupancy.
  - Implementation in COATJAVA using Deep Java Library
    - Already started, can load the pytorch model in Java and predict output
    - Same prediction between pytorch and DJL
  - Develop an AI for PID using MLP and a classifier with 5 classes.

Work done by Uditha Weerasinghe and me.