AHDC AI Tracking

Mathieu Ouillon (Mississippi State University)

07/08/25

CLAS Collaboration Meeting (8-11 July, 2025)

ALERT Experimental Setup

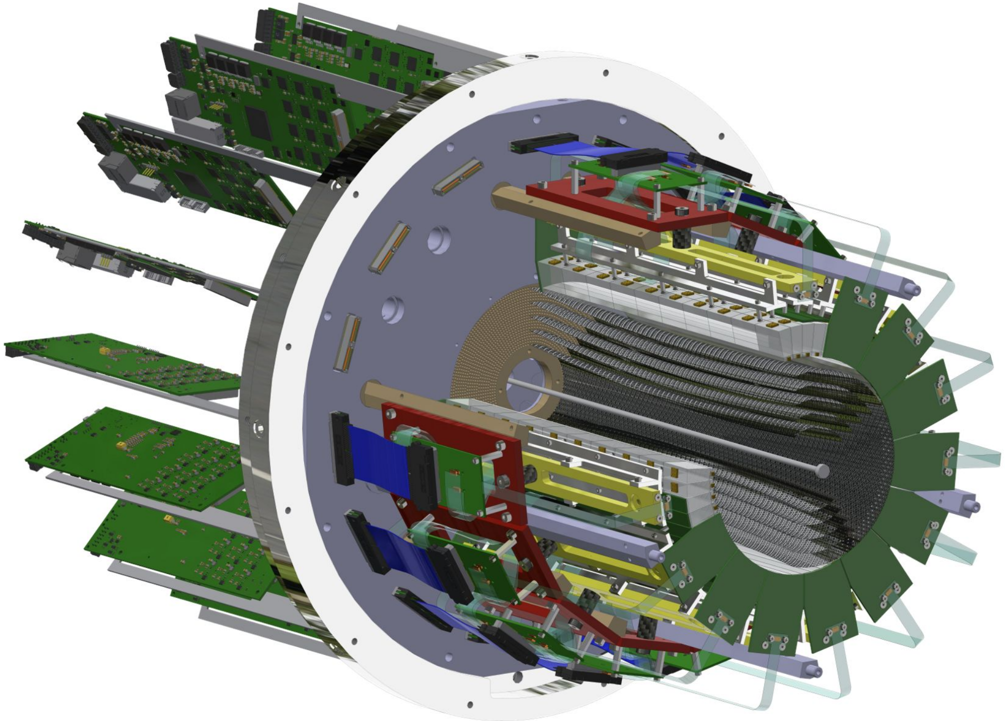
- ALERT have two sub-detectors: A Hyperbolic Drift Chamber (AHDC) and A Time of Flight (ATOF)

ATOF

- Time of flight: used for Particle IDentification
- Small barrel of segmented scintillators
- The TOF measurement is degenerate for ^{2}H and ^{4}He, but dE/dx can distinguish the two nuclei bands

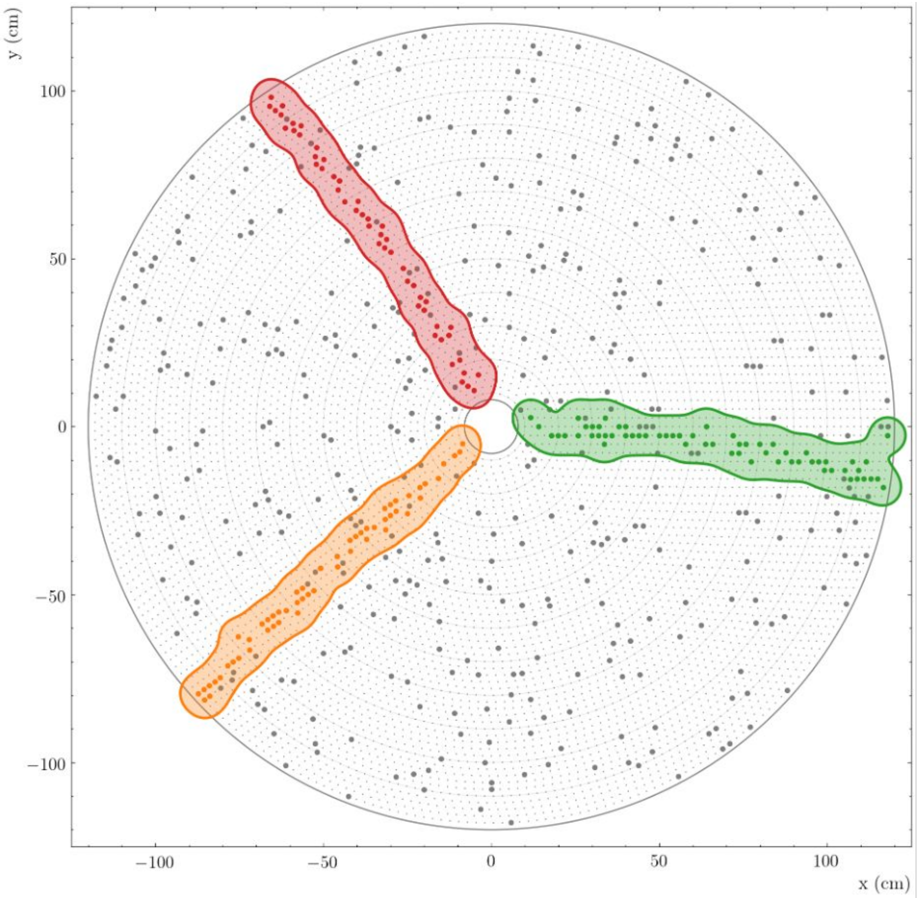
AHDC

- Aluminum wires: 2 mm apart
- 20-degree stereo angle (hyperbolic shape)
- 5 superlayers, each composed of 2 layers
- 576 signal wires (6 ground wires of each signal)



Track Finding

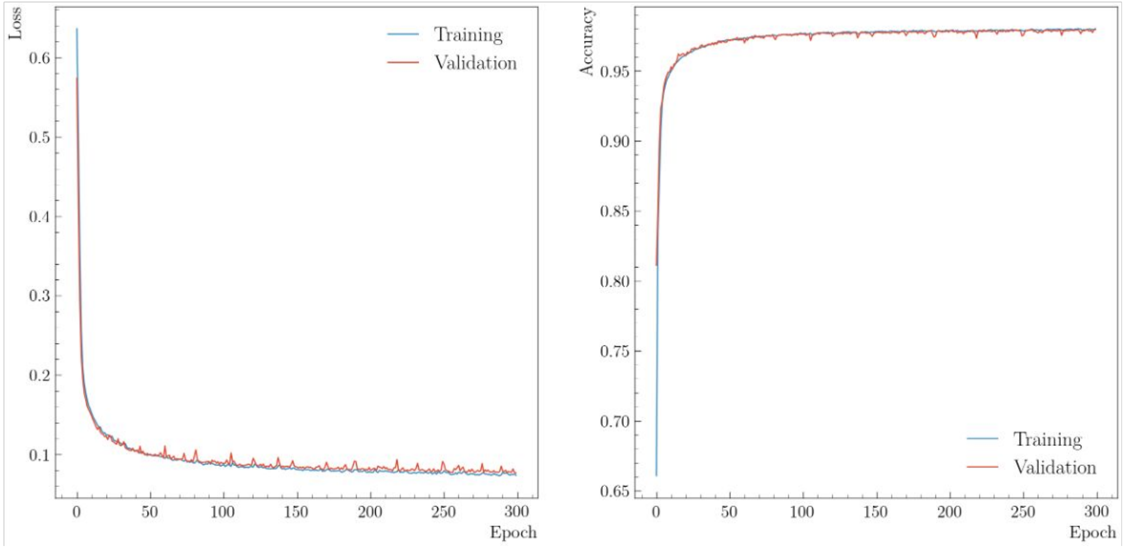
- Track finding is a clustering problem:
  - Set of points (hits) ⇒ cluster in sets (tracks) originating from the same particle
  - Hits: particles deposit energy when interacting with the detector material
  - Tracks: reconstructed sequences of hits representing charged particle trajectories

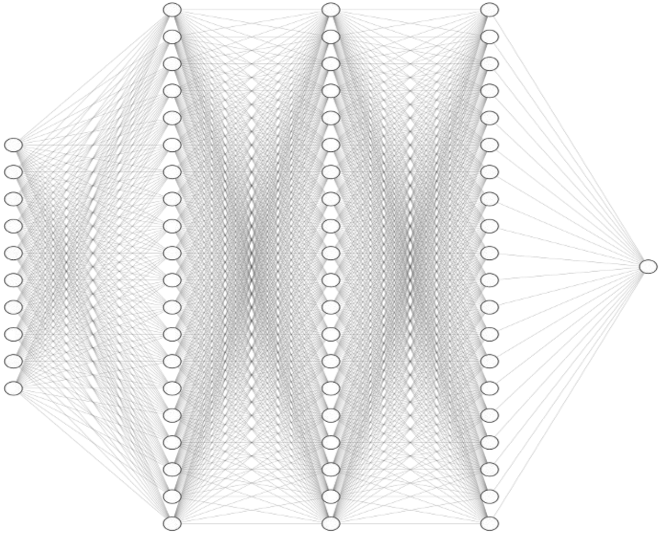


- Different algorithms:
  - Distance between hits + fit
  - Hough transform
  - Combinatorial Kalman Filter
  - Artificial Intelligence models (MLP, GNN…)

AI-assisted model: Description and Training

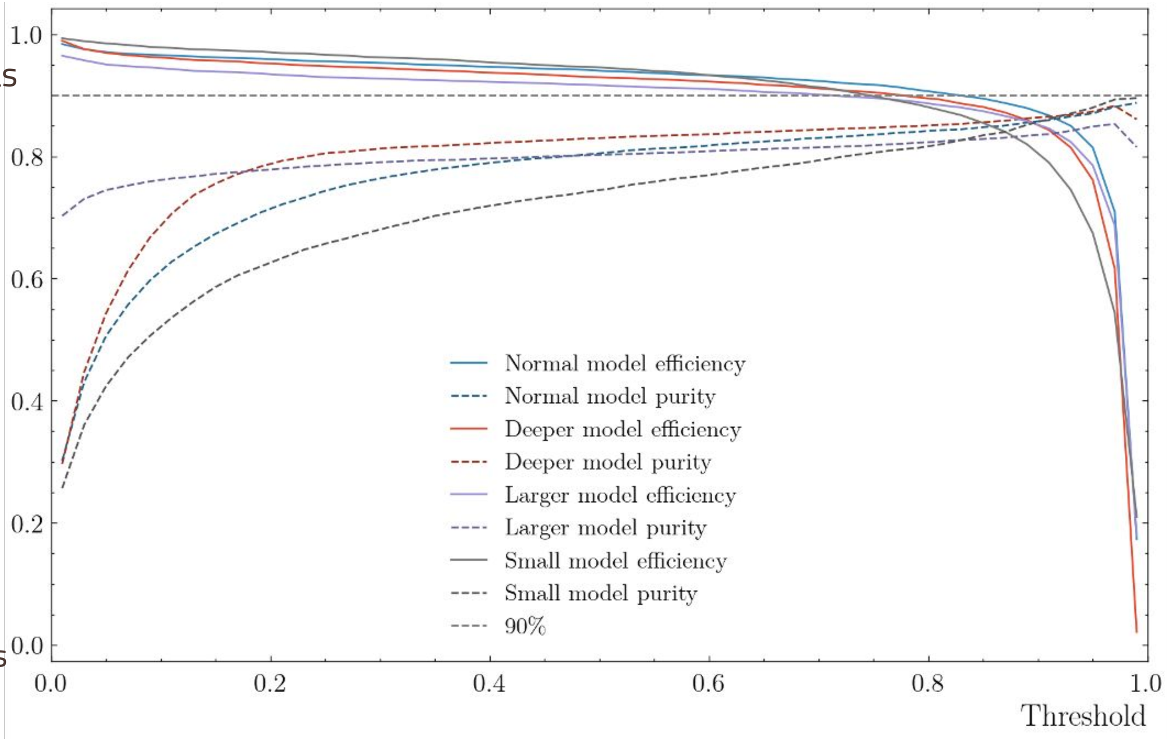
- Model: MultiLayer Perceptron, 10 inputs, 1/3/5 hidden layer (15/20/100 neurons), 1 output
- Inputs: x and y values of the five inter-clusters
- For the training ⇒ Need good and bad tracks:
  - Good tracks: GEANT4 simulation (particle with $p \in [0.07, 1.5]$ GeV/c, $\varphi \in [0, 360]°$, $\theta \in [30, 150]°$, and $Vz \in [-15, 15]$ cm)
  - False tracks: Interchanging randomly up to two inter-clusters with another event
  - Generate 5M events composed of all light nuclei (flat distribution for all variables)
  - Output: Number between 0 and 1, with 0/1 means bad/good track





Simulation Validation: Efficiency and Purity vs. Threshold

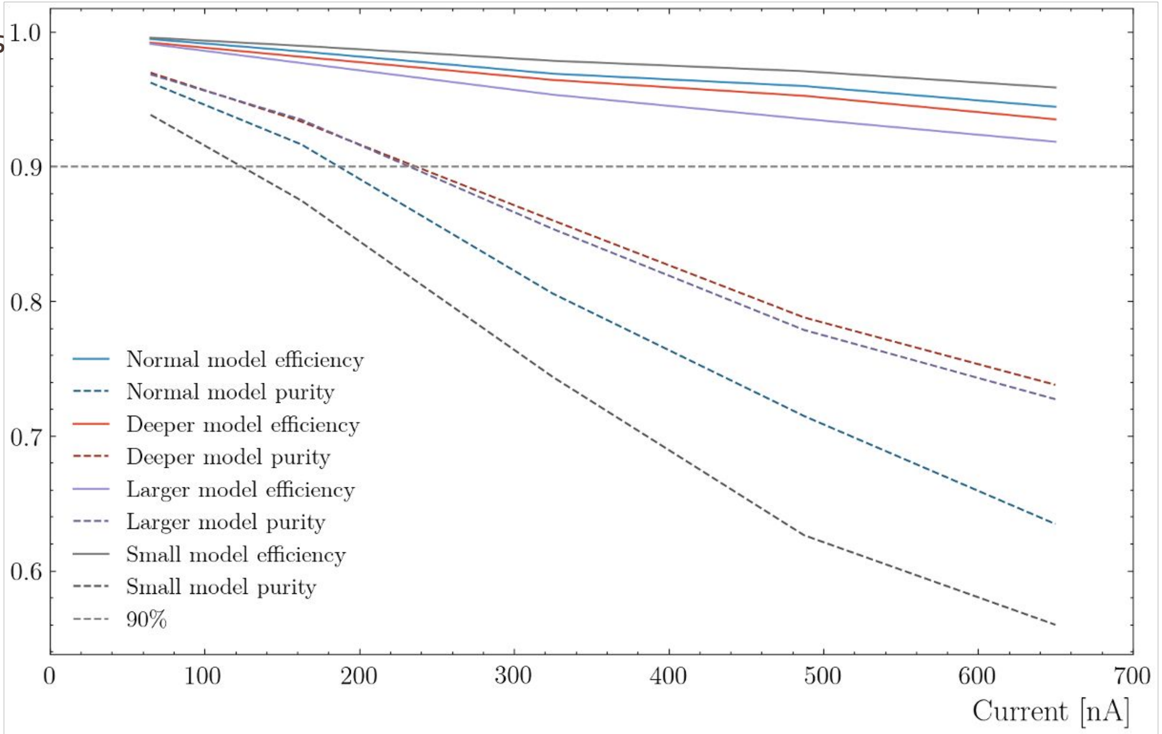
- Threshold: if output above/lower than the threshold ⇒ good/bad tracks
- To evaluate the model:
  - Efficiency: Number of good tracks classified as good normalized by the number of events.
  - Purity: Number of good tracks classified as good normalized by the number of tracks (good or bad) classified as good.
  - Events need to have at least one track candidate.
  - Set the threshold to 0.2 to have a higher efficiency



  - Blue: model with 20 neurons in 3 hidden layers
  - Violet: model with 100 neurons in 3 hidden layers
  - Red: model with 20 neurons in 5 hidden layers
  - Gray: model with 15 neurons in 1 hidden layer

Simulation Validation: Efficiency and Purity vs. Current

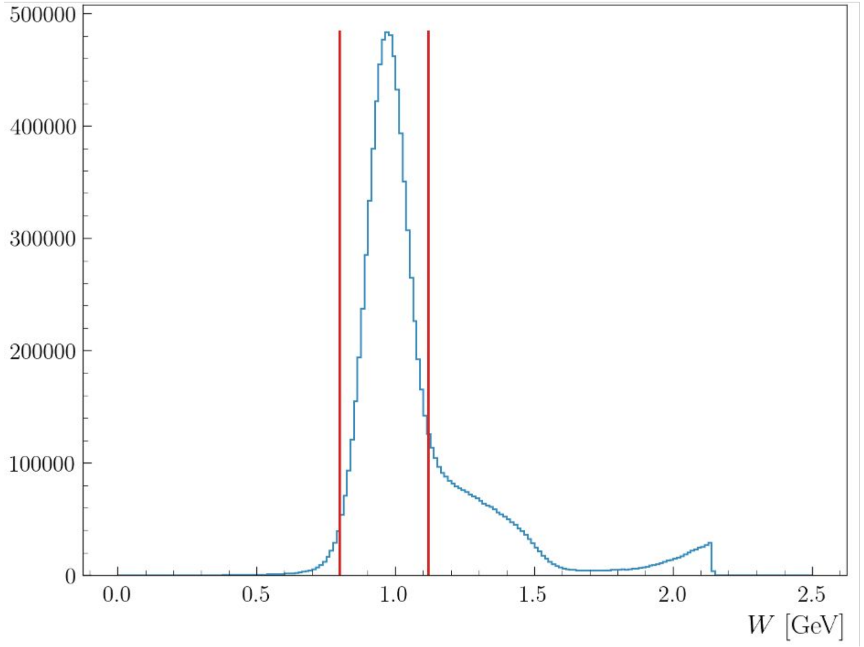
- Efficiency is always higher than 90% and the purity is between 55% and 95%
- More current means more background
  - Efficiency: Number of good tracks classified as good normalized by the number of events.
  - Purity: Number of good tracks classified as good normalized by the number of tracks (good or bad) classified as good.
  - Events need to have at least one track candidate.
  - Threshold set to 0.2 to have a higher efficiency

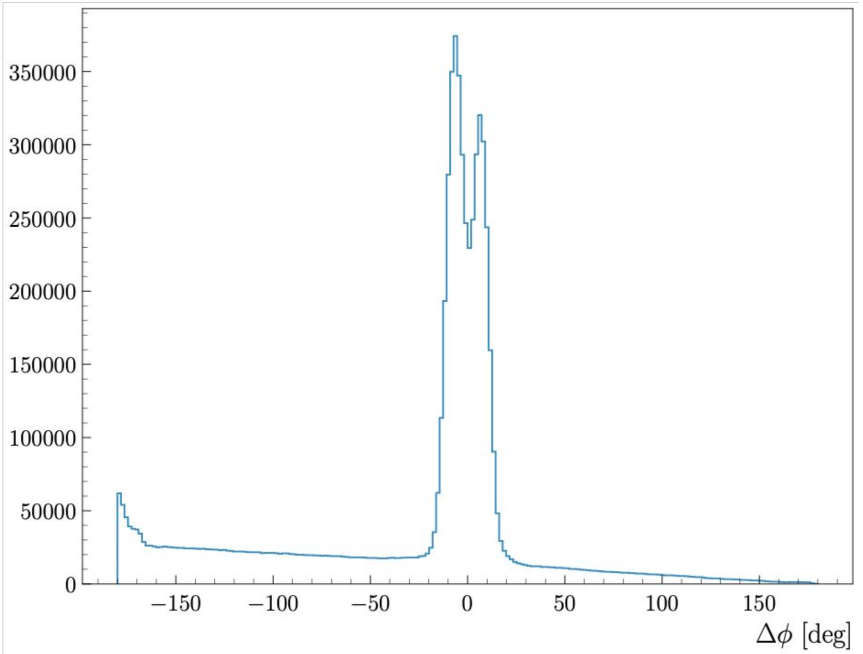


  - Blue: model with 20 neurons in 3 hidden layers
  - Violet: model with 100 neurons in 3 hidden layers
  - Red: model with 20 neurons in 5 hidden layers
  - Gray: model with 15 neurons in 1 hidden layer

Validation with Proton data: Elastic Scattering

- To evaluate the performance of the AI, use elastic scattering on proton
- For the AHDC, want the low momentum proton ⇒ use electron at low θ
- Compute Δϕ using electron and AHDC hits ⇒ shift in Δϕ approx. 20°



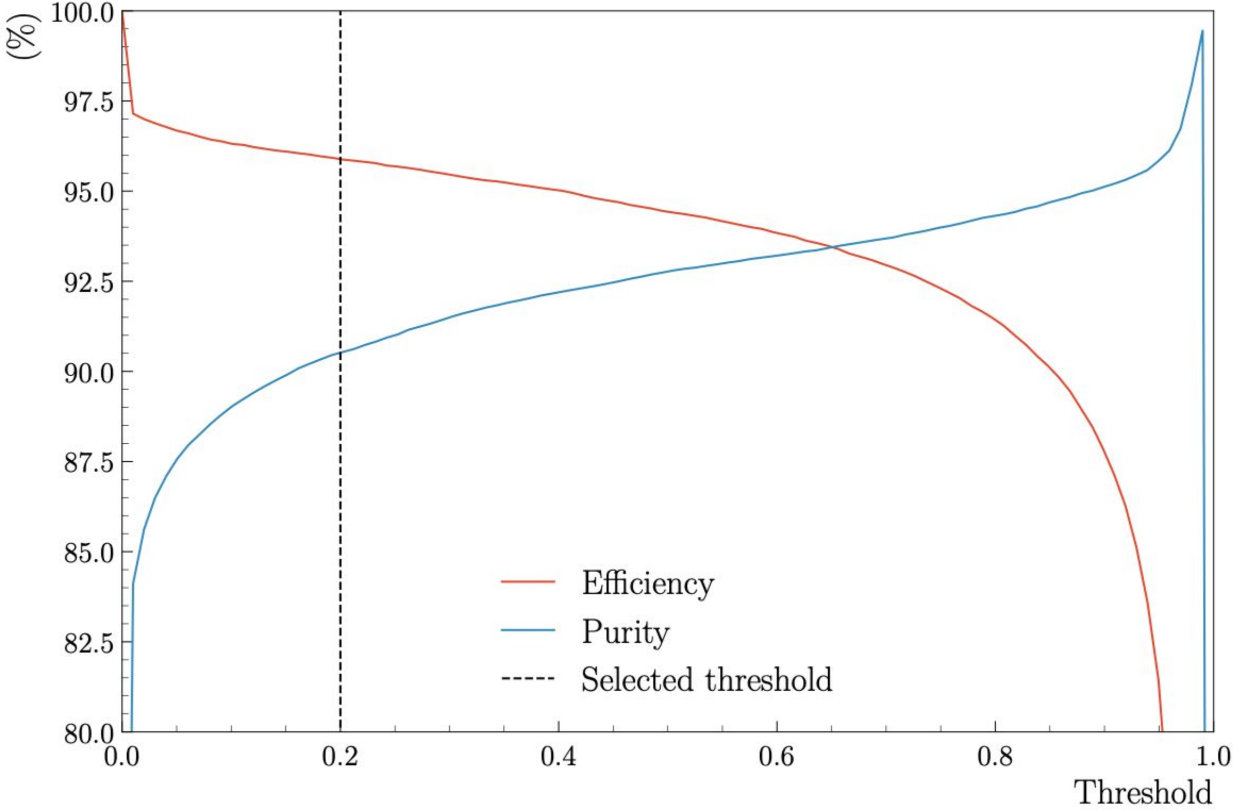


Validation on Proton Data: AI Efficiency and Purity

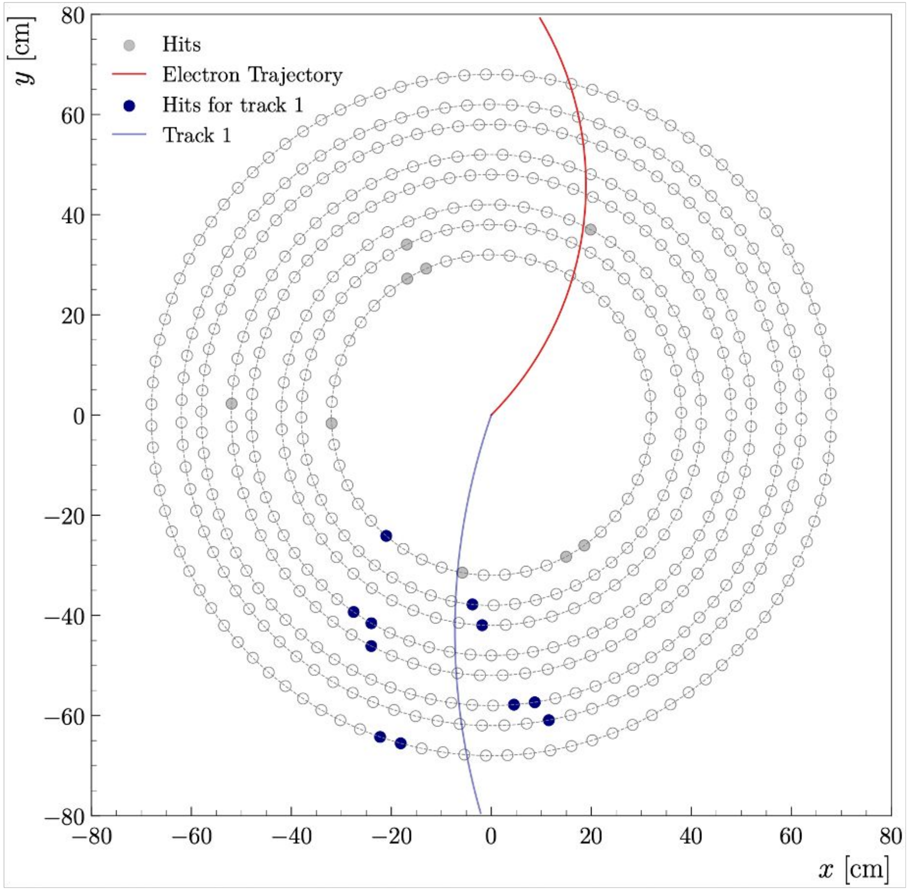
- Efficiency and purity as a function of the threshold:

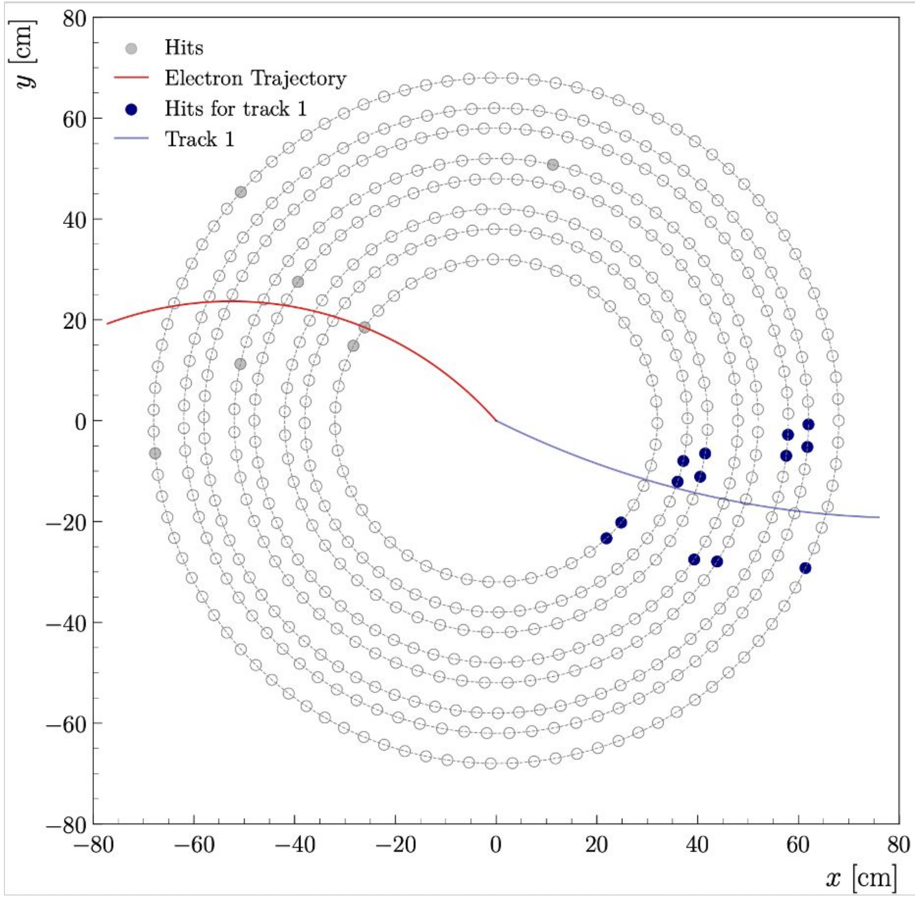
- Threshold: if output above/lower than the threshold ⇒ good/bad tracks
- Efficiency: Number of good tracks classified as good normalized by the number of events.
- Purity: Number of good tracks classified as good normalized by the number of tracks (good or bad) classified as good.

- We have an efficiency of 96%, and a purity of 90% at 0.2



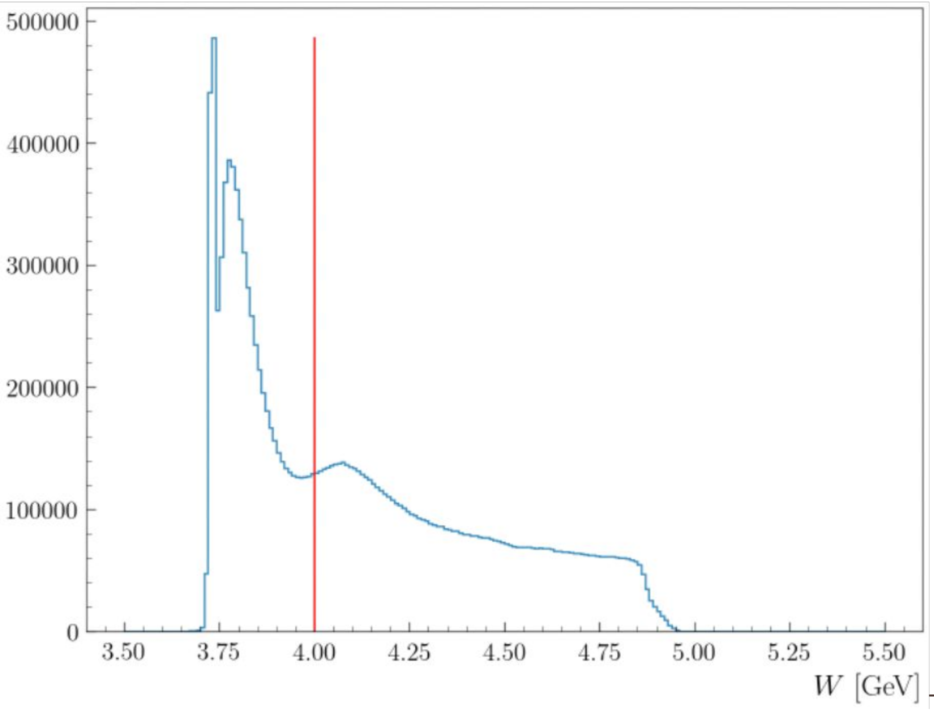
Validation with Proton data: Event Display

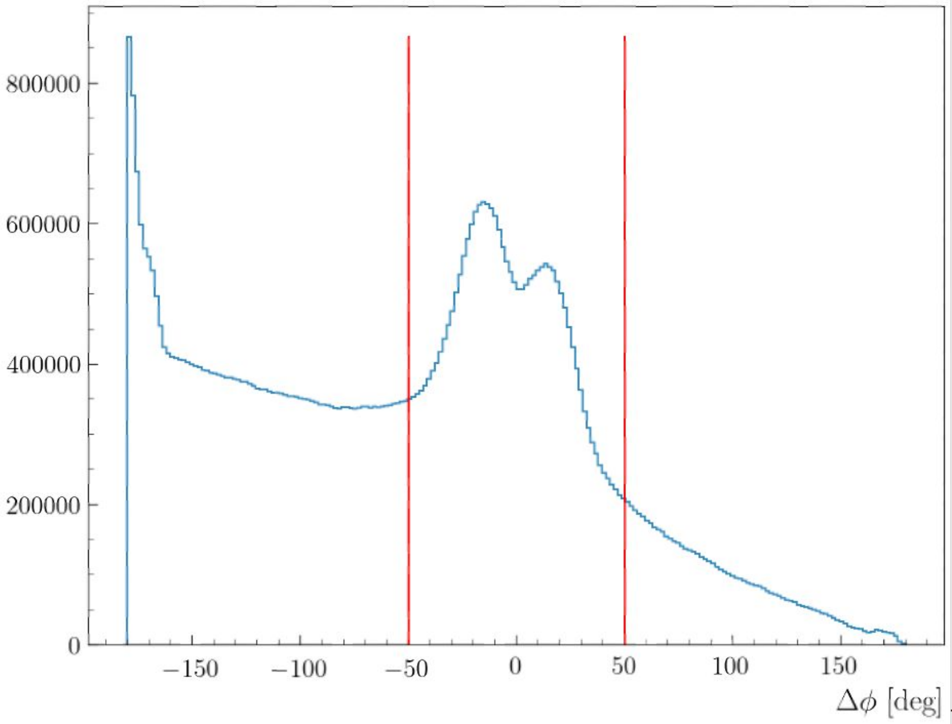




Validation with ^{4}He Data: Elastic Scattering

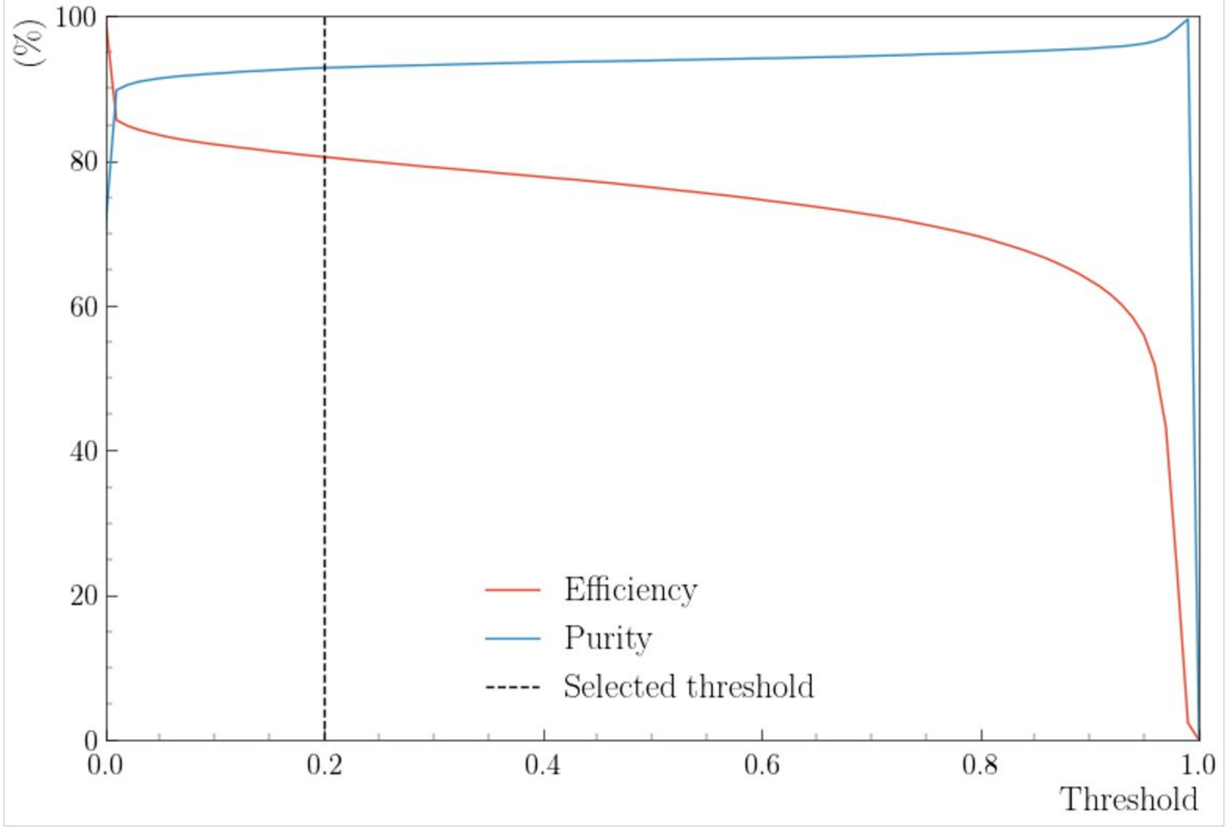
- To evaluate the performance of the AI, use elastic scattering on 4He
- For the AHDC, want the low momentum ^{4}He ⇒ use electron in FD
- Compute Δϕ using electron and AHDC hits





Validation with ^{4}He data: AI Efficiency and Purity

- Efficiency and purity as a function of the threshold:

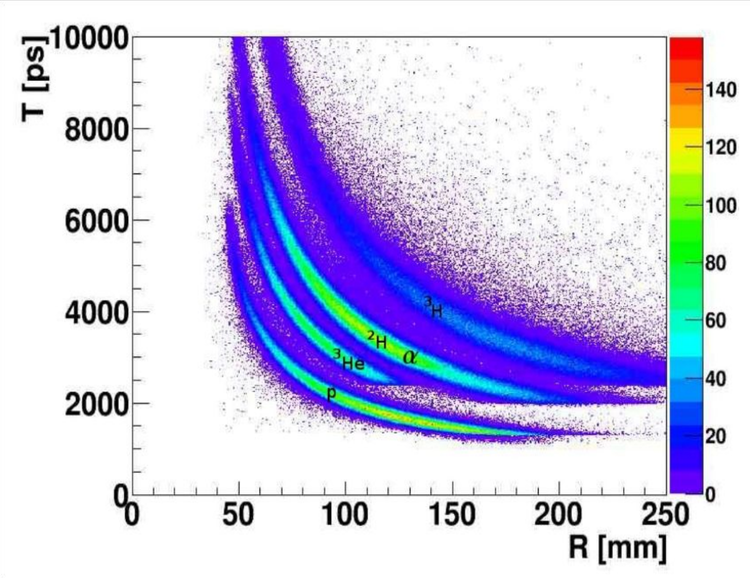


- Threshold: if output above/lower than the threshold ⇒ good/bad tracks
- Efficiency: Number of good tracks classified as good normalized by the number of events.
- Purity: Number of good tracks classified as good normalized by the number of tracks (good or bad) classified as good.

- We have an efficiency of 81%, and a purity of 93% at 0.2

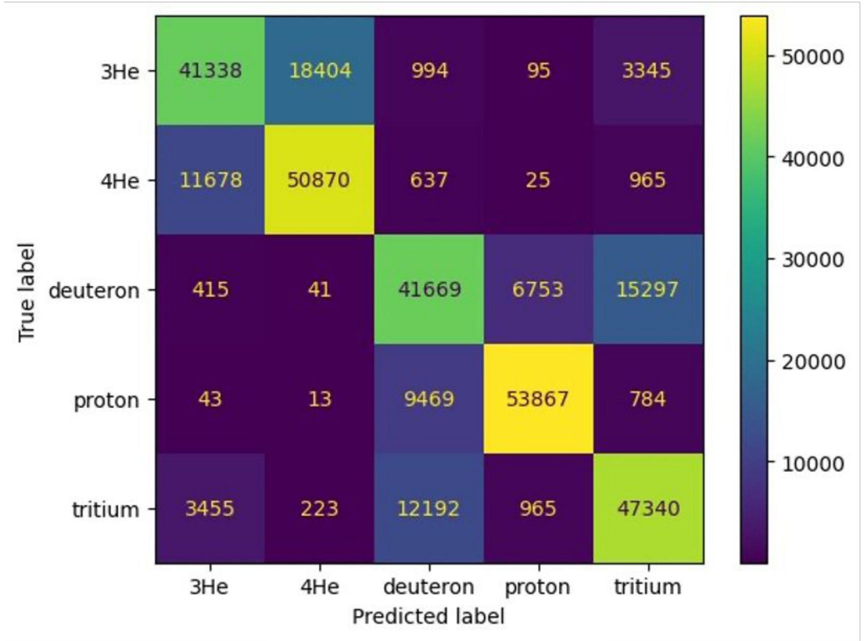
ALERT AI-Assisted PID

- Ongoing efforts to deploy AI techniques to improve particle identification by a teammate Uditha Weerasinghe:
  - perfect task for machine learning
  - can learn non-trivial relations between different track parameters and PID
- MultiLayer Perceptron model has been used to classify recoil nuclear-target fragments that are detected by ALERT
- A set of 27 features to include:
  - momentum, energy deposited, inter-cluster position, AHDC residual, ATOF cluster position, time, and path length.
- Main limitation: Quality of the classifier will depend on the MC sample



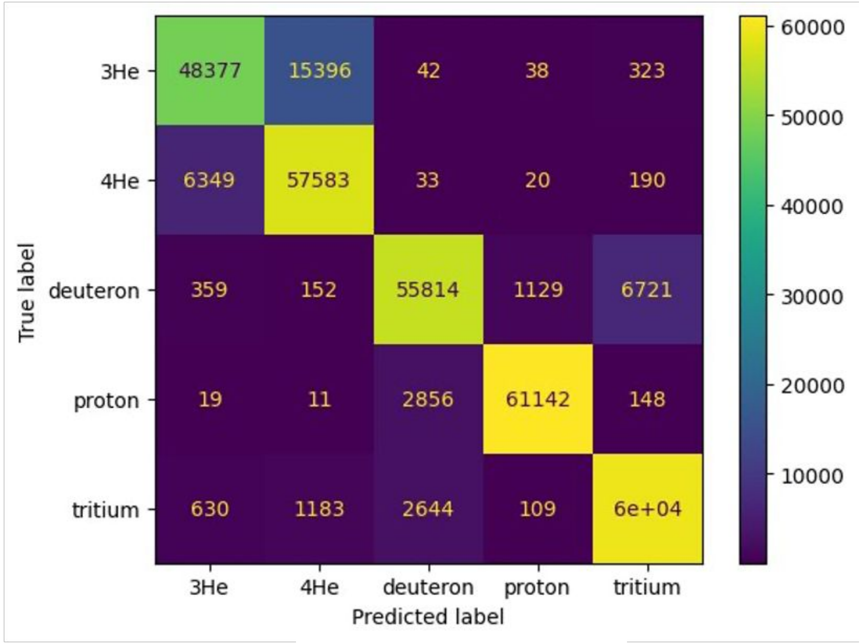
ALERT AI-Assisted PID

For reconstructed momentum



Protons: 83.9 %
Deuterons: 64.9 %
Tritium: 73.7 %
Helium-3: 64.4%
Helium-4: 79.3%

For generated momentum



Protons: 95.3 %
Deuterons: 87.0 %
Tritium: 92.9 %
Helium-3: 75.4%
Helium-4: 89.7%

Summary and Outlook

- An MLP have been developed to improve track finding for ALERT:
  - Evaluated efficiency and purity as a function of momentum, threshold, and current for simulation
  - Evaluated efficiency and purity as a function of momentum and threshold for elastic data
  - Efficiency is always higher than 90% on proton and 80% on ^{4}He
  - Worked on a classifier for the PID

- Remaining works:
  - Matching hit in the ATOF with track in the AHDC using AI
  - Improve the performance of the PID classifier

THANKS