ALERT AI Track Finding

Mathieu Ouillon (Mississippi State University)

CLAS Collaboration Meeting 03/05/2025

The ALERT detector

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

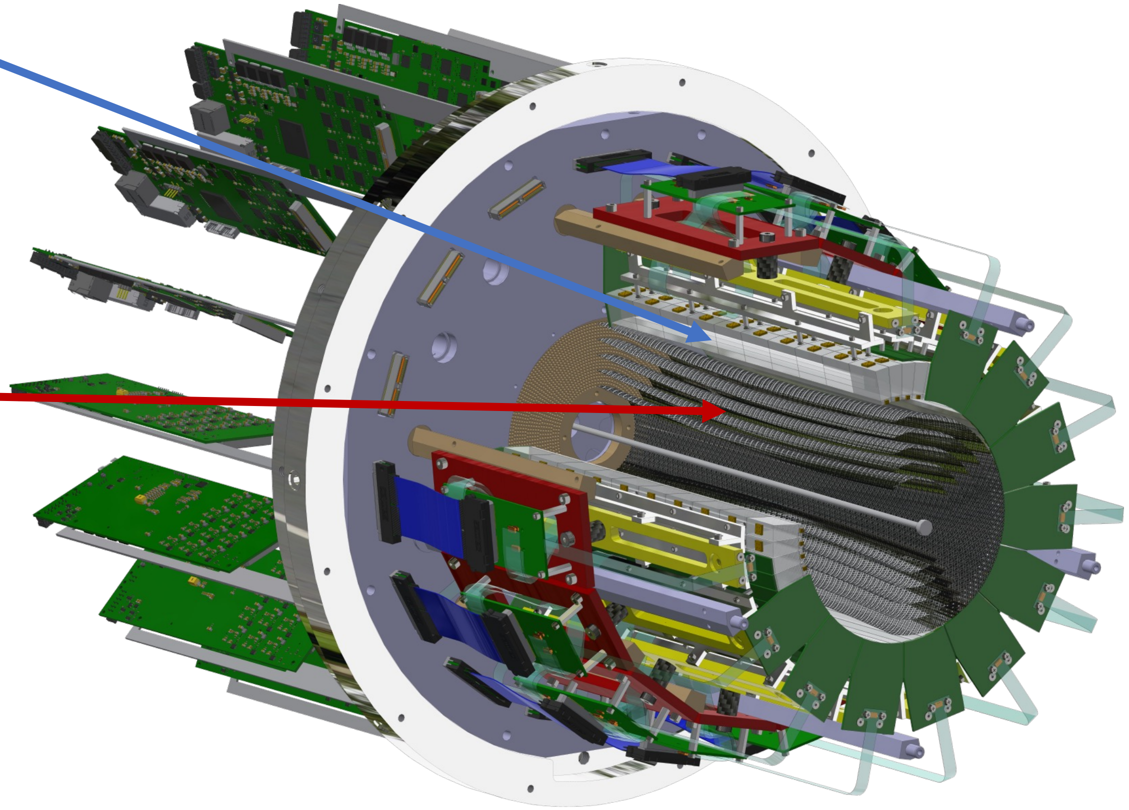
- Identify light ions: p, ^{2}H, ^{3}H, ^{3}He, ^{4}He.
- Detect the lowest momentum possible.

ALERT ToF

- Time-of-Flight: use for PID
- 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

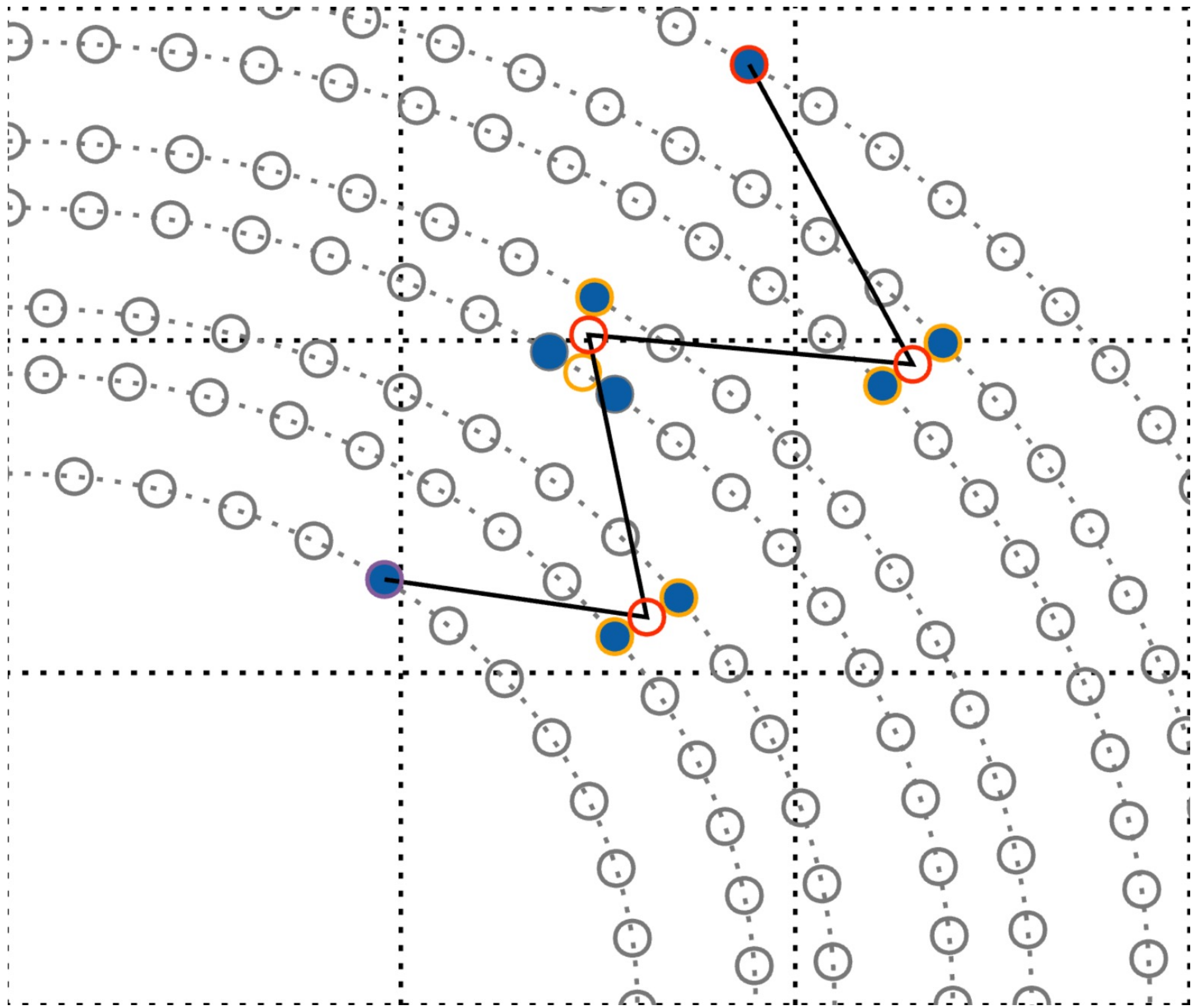
ALERT HDC

- Aluminum wire: 2mm spacing
- 20-degree stereo angle (hyperbolic shape)
- 5 superlayers, each composed of 2 layers
- 576 signal wires:
  - 47, 56, 72, 87, 99 for each superlayer.



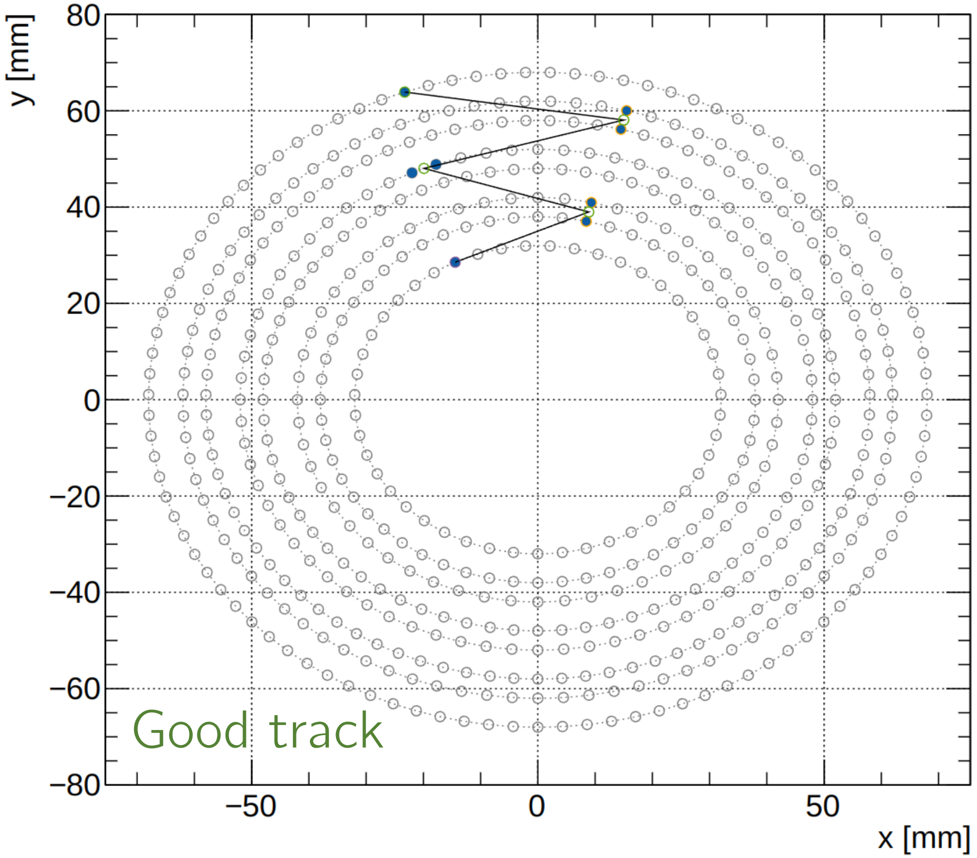
Clustering and track candidates

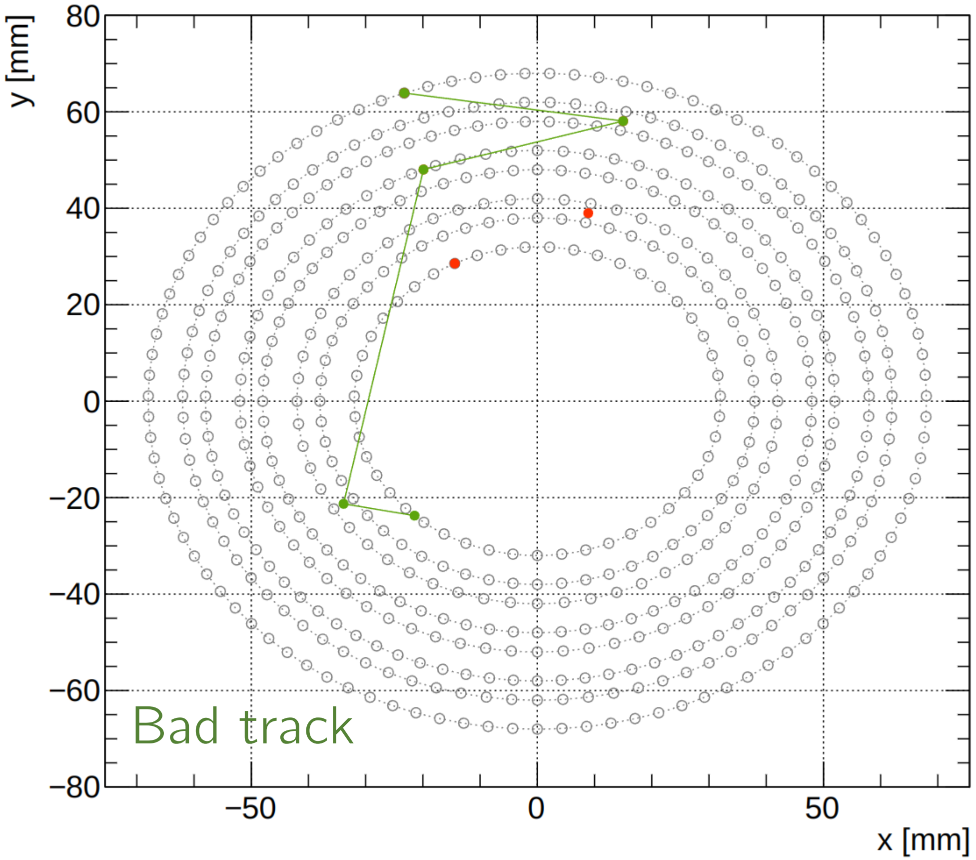
- The first step is to find all track candidates:
    - First step is clustering, which means merging hits close in the x-y plane.
    - Goal is to reduce the combinatoric (number of track candidates).
    - Merge hits on the same layer that are one wire apart into precluster.
    - Merge precluster in the same superlayer that are less than 8mm apart into superprecluster.
    - Generate all track candidates with 5 superpreclusters (one on each superlayer)



- End up with around 20-100 track candidates per event (depending on the background).

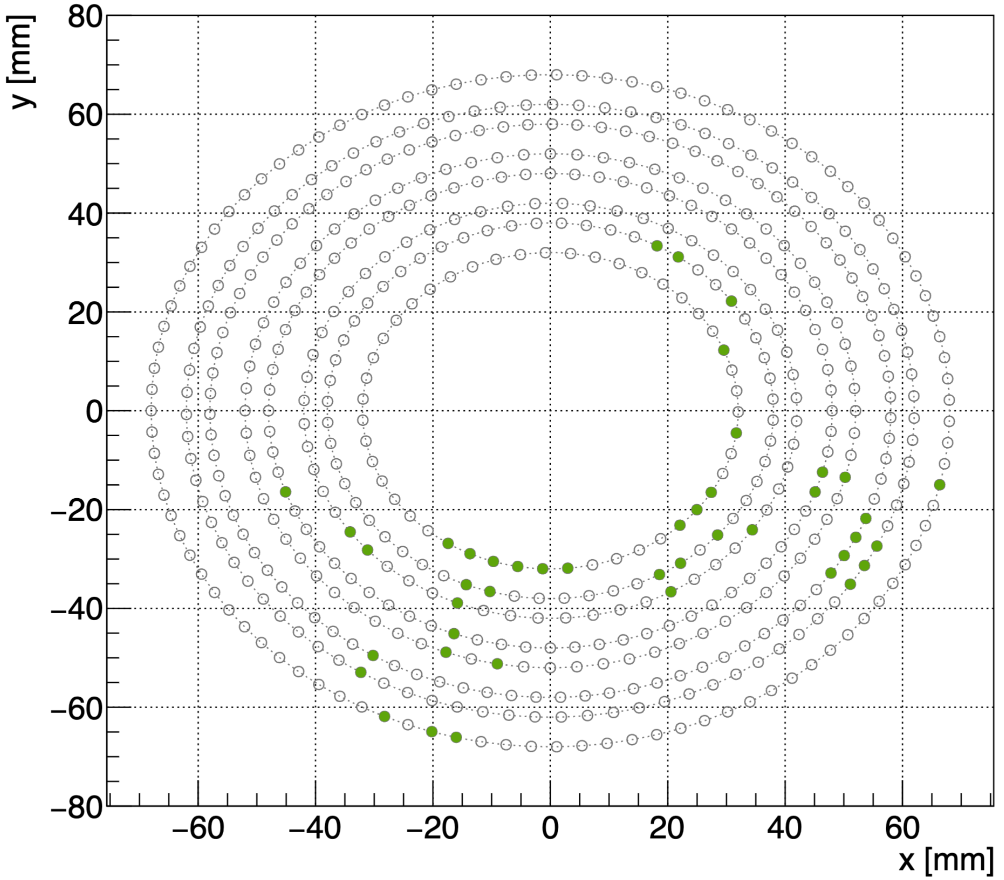
- Model: MLP 10 inputs, 3/5 hidden layer (20/100 neurons), 1 output.
- The input for the model is five combinations for x and y, representing the superpreclusters.
- GEMC generates good tracks: proton with p ∈ [70, 250] MeV/c, φ ∈ [0, 360]°, θ ∈ [60, 120]° and Vz ∈ [-15, 15] cm.
- False tracks: interchanging randomly up to two superpreclusters with another event.
- Output: Number between 0 and 1, with 1 mean good track and 0 mean bad track

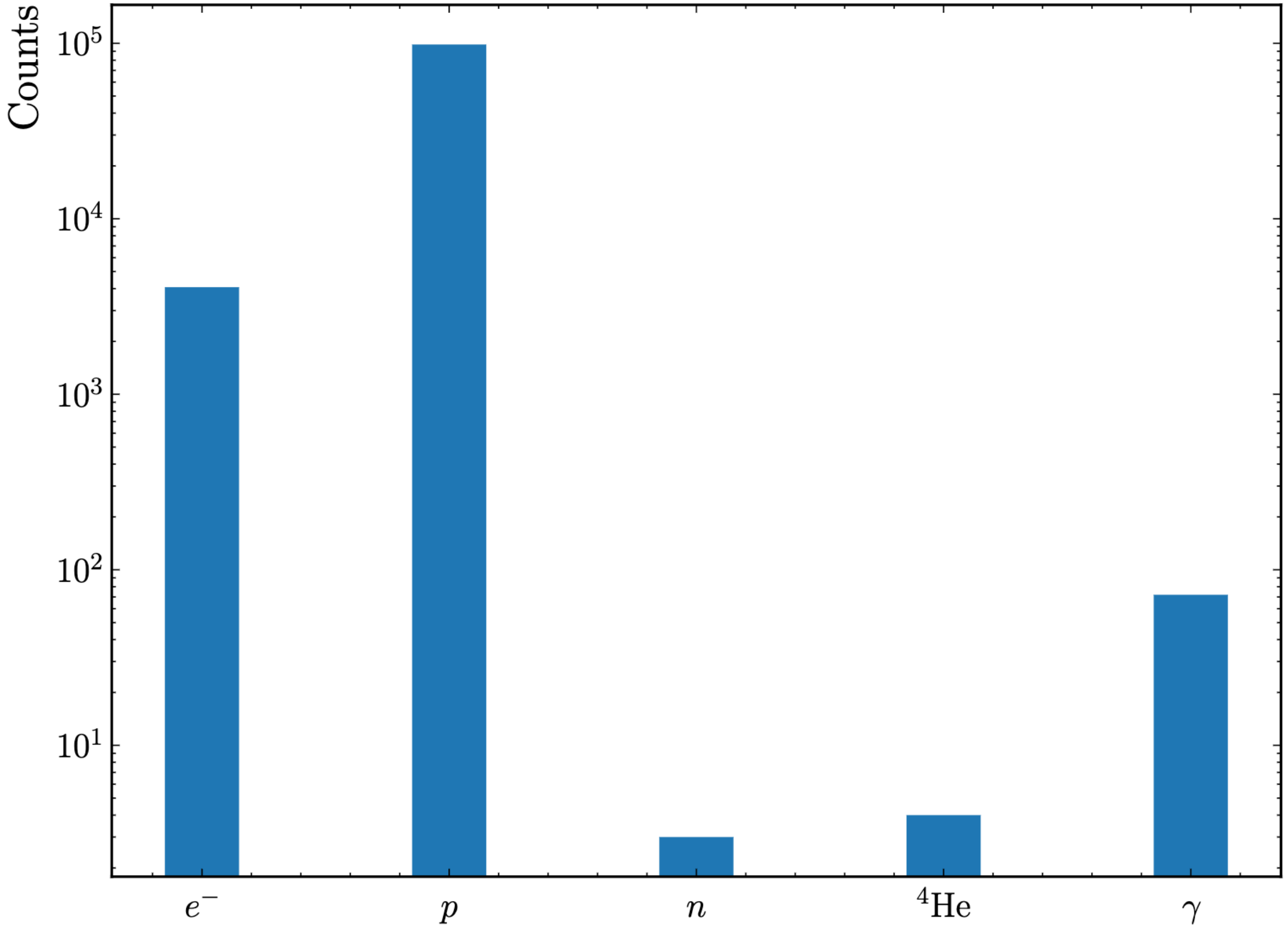




Background generation of evaluation

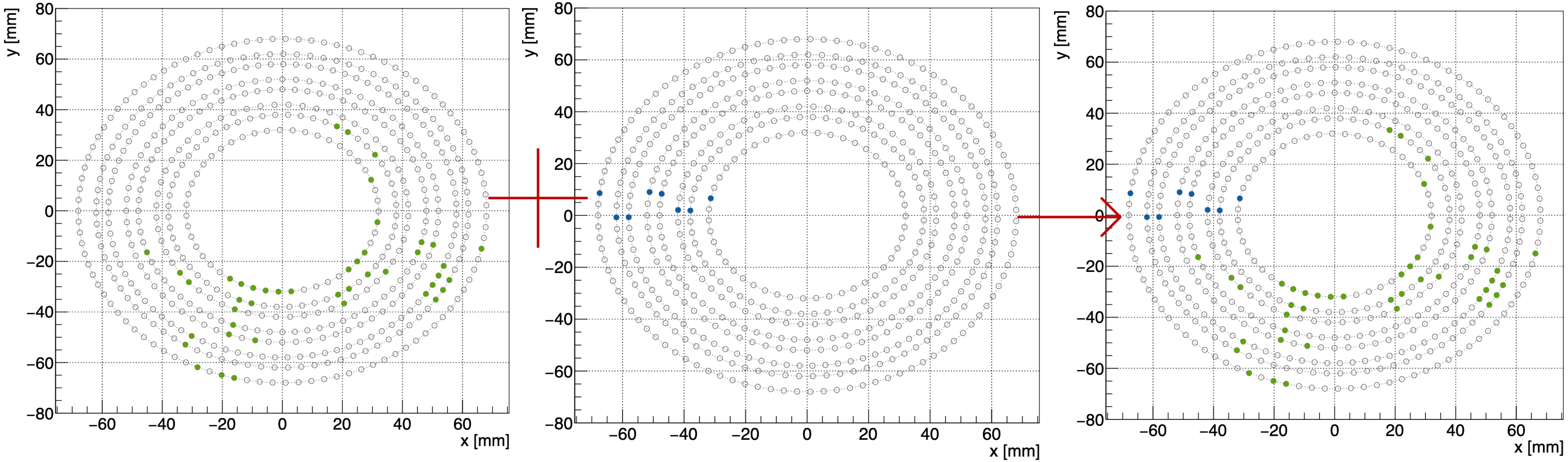
- To evaluate the model, we need to generate some background events:
  - Use luminosity simulation: 80k to 800k electrons going through the target were simulated for each event within a 248.5 ns time window.
  - Take the hits in the AHDC as background.





Evaluation sample

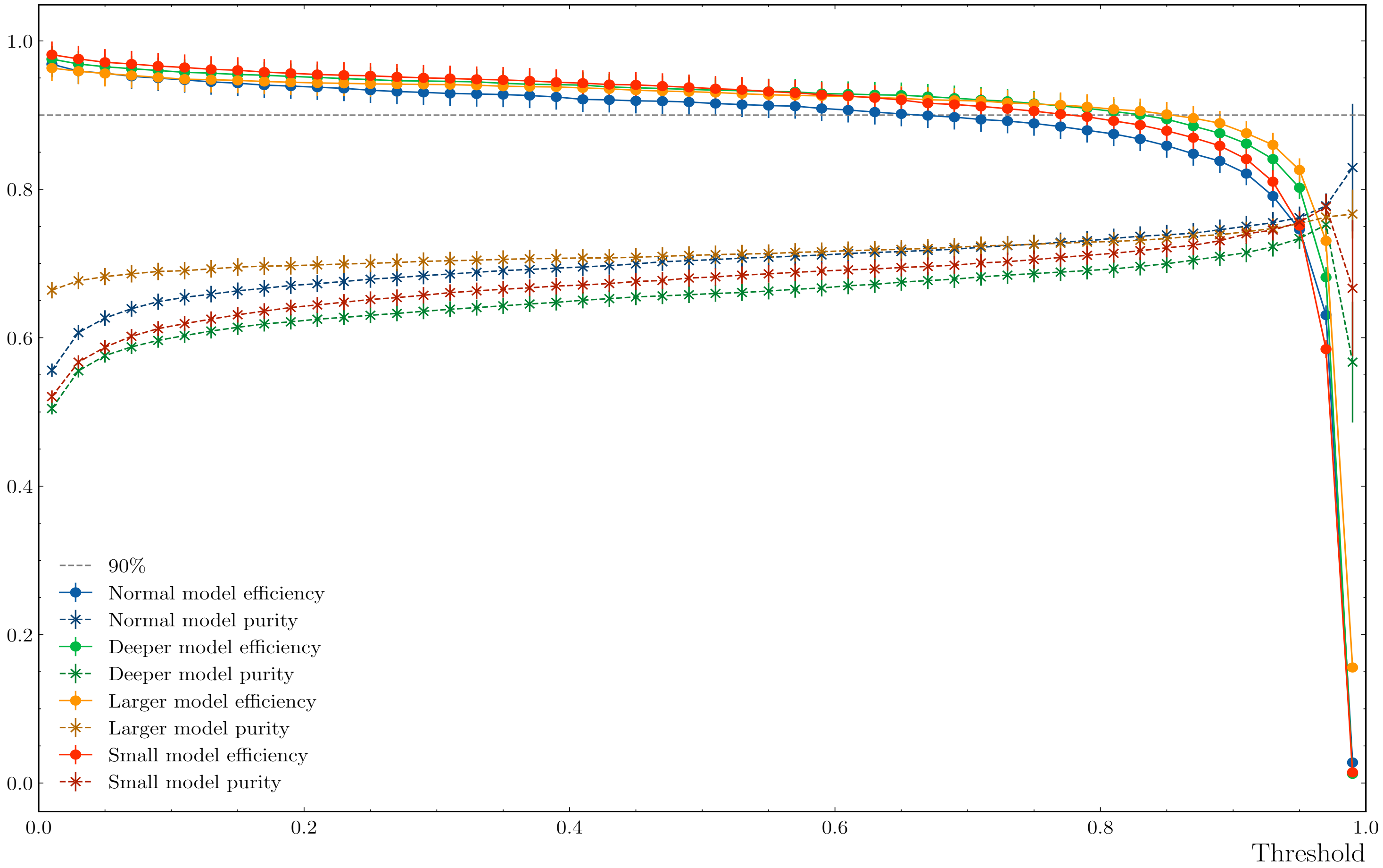
- Now that we have the background, we need to add the proton track.
- Generate a proton track using GEMC particle gun with the same initial parameters as the one used for the training sample.



Efficiency and purity vs. threshold

- Have everything to evaluate the model. First thing is to set the threshold: value above what we consider the model output as good track. To evaluate the model, we use:
- Efficiency: Number of good tracks classified as good / number of events.
- Purity: Number of good tracks classified as good / number of tracks (good or bad) classified as good.
- The event needs to have at least a track candidate.
- To have a higher efficiency without sacrificing to much purity, we set the threshold to 0.2

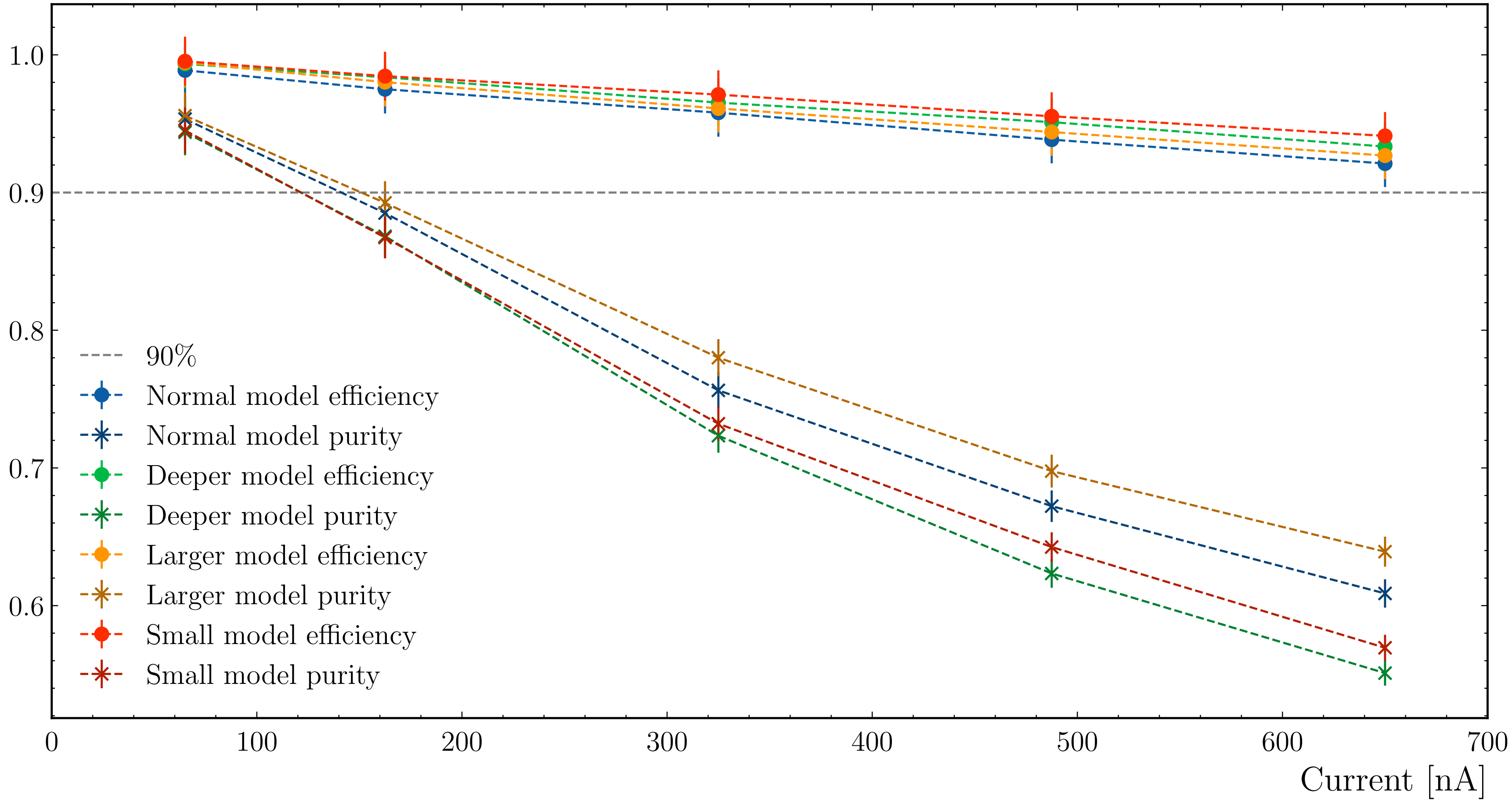
- Blue: model with 20 neurons in 3 hidden layers
- Orange: model with 100 neurons in 3 hidden layers
- Green: model with 20 neurons in 5 hidden layers
- Red: model with 20 neurons in 1 hidden layer



Efficiency and purity vs. current

- Evaluate the model for different currents: from 65 to 650 nA
- Efficiency is always higher than 90%, and the purity is between 95% and 55%.
- Prefer higher efficiency and lower purity than the opposite.

- Blue: model with 20 neurons in 3 hidden layers
- Orange: model with 100 neurons in 3 hidden layers
- Green: model with 20 neurons in 5 hidden layers
- Red: model with 20 neurons in 1 hidden layer

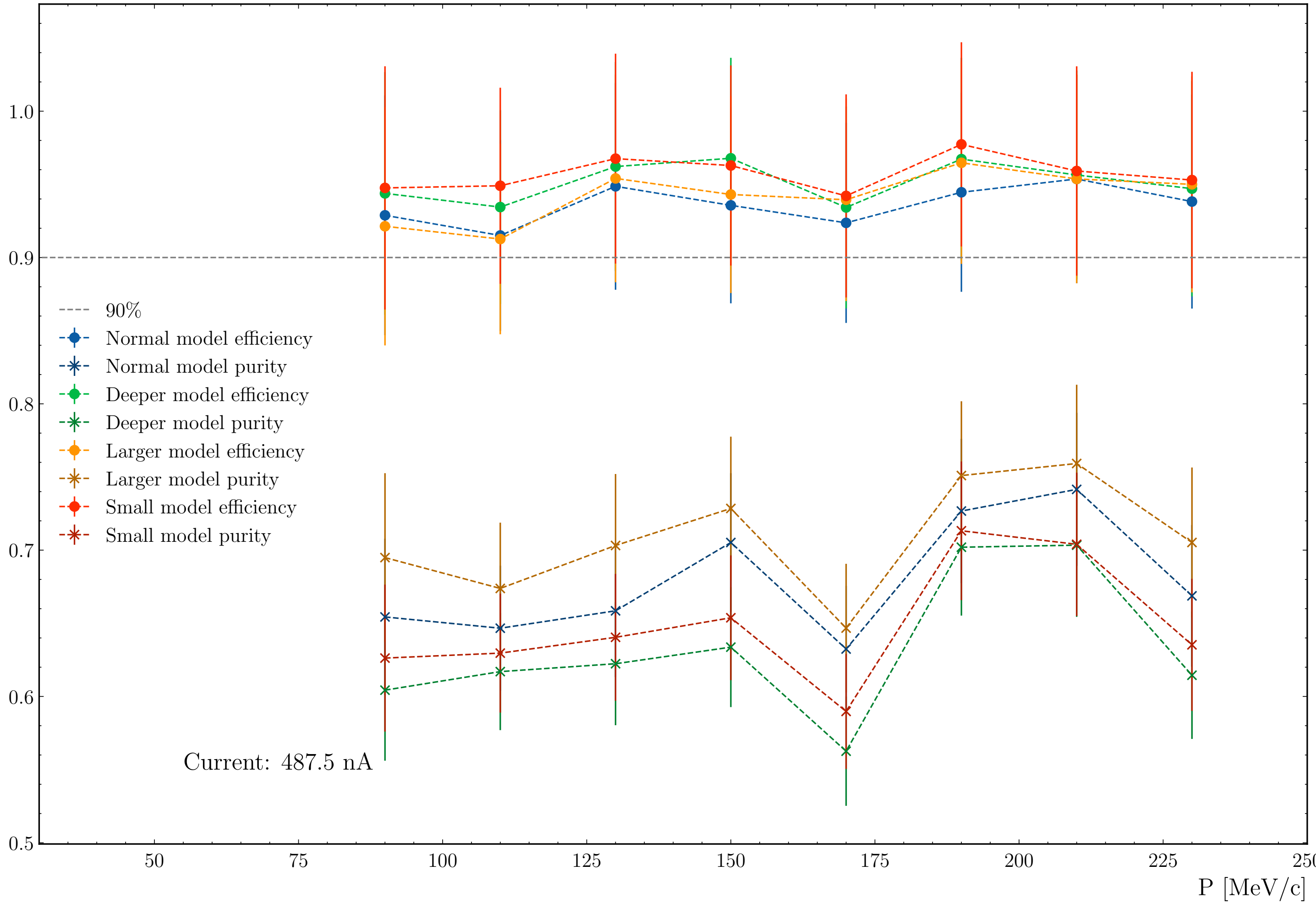


Efficiency and purity vs. momentum

- Can also compute the efficiency and purity as function the the proton momentum.
- Use the background generated with current I = 487.5 nA

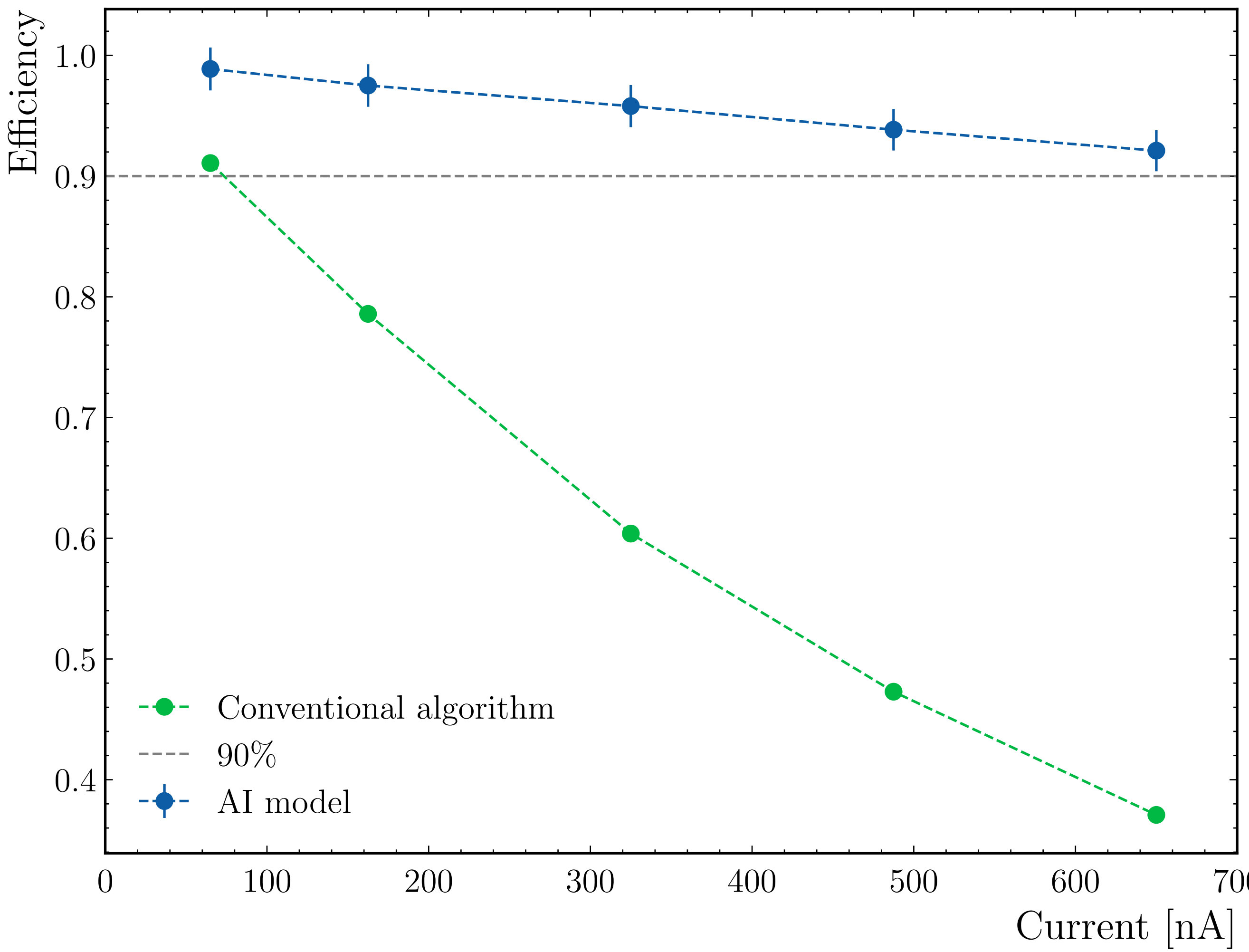
- Efficiency and purity are constant across the momentum range.

- Blue: model with 20 neurons in 3 hidden layers
- Orange: model with 100 neurons in 3 hidden layers
- Green: model with 20 neurons in 5 hidden layers
- Red: model with 20 neurons in 1 hidden layer



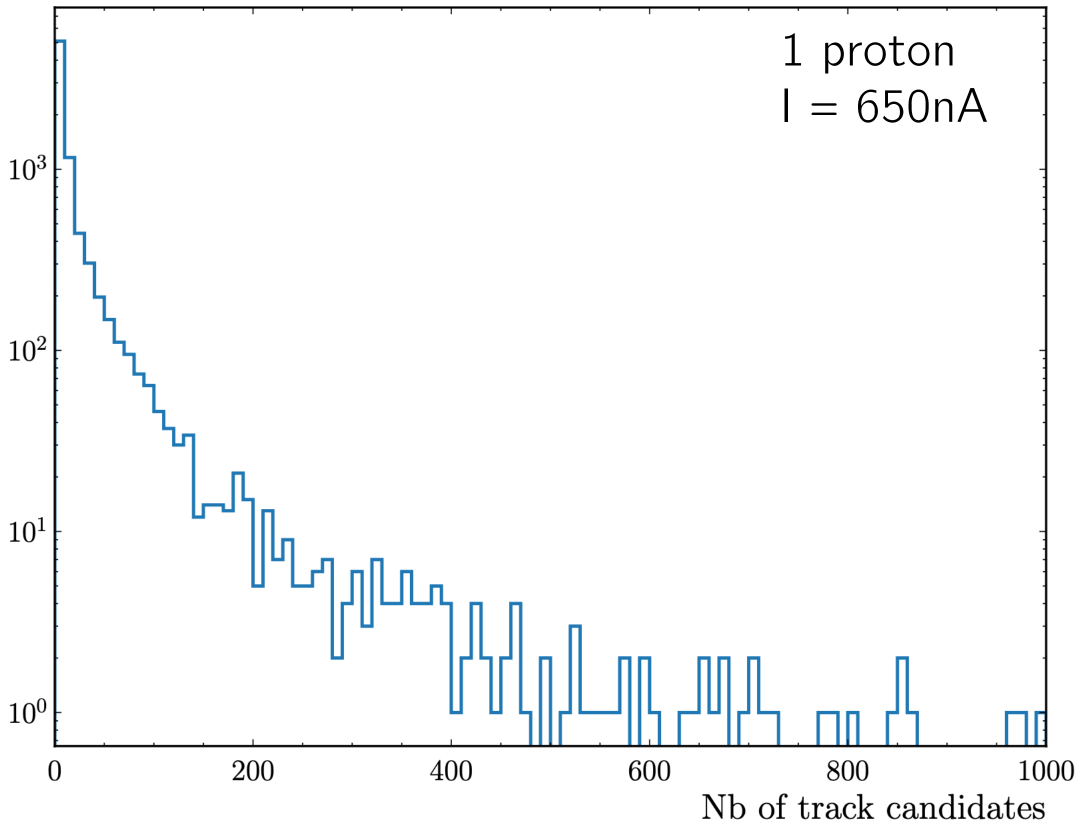
Conventional vs AI track finding

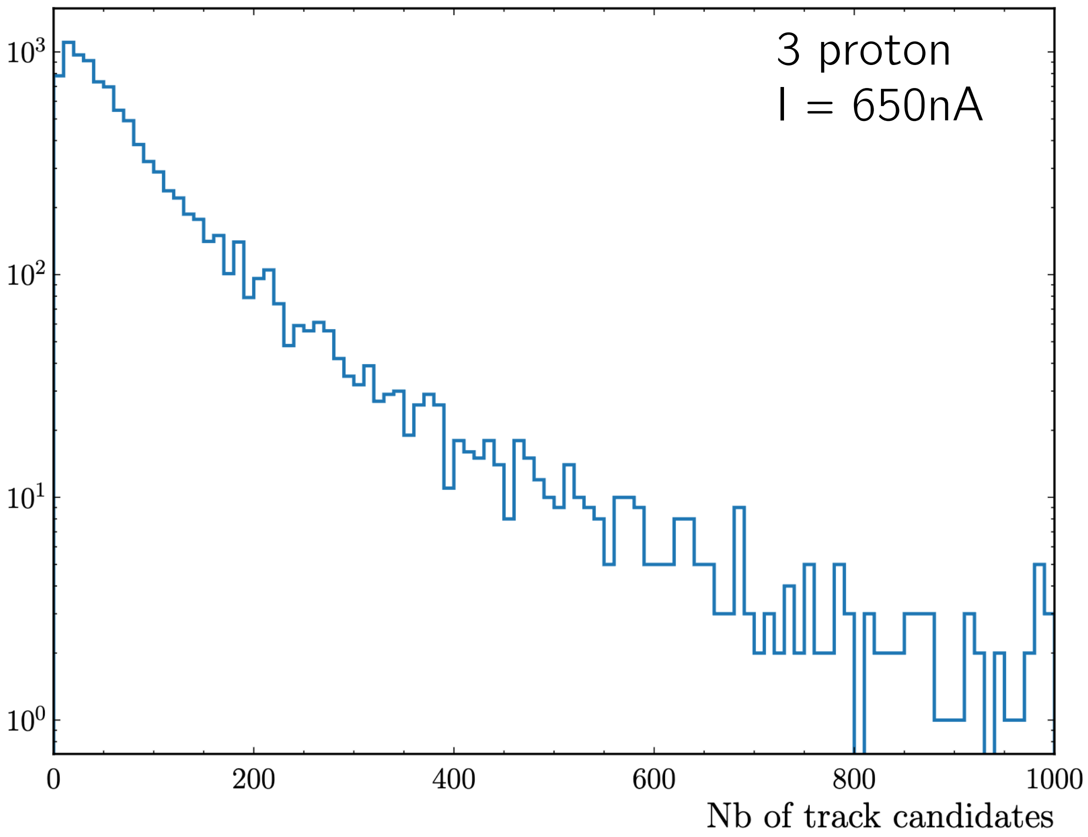
- Compare the conventional track finding to the AI model.
- Take all the track candidates given after merging the background and the proton track to both algorithms.
- For the higher current, the conventional track finding has an efficiency of 40% compared to the AI.
- For the lowest current, the difference is much smaller: 92% for conventional and almost 100% for the AI.



Inference time

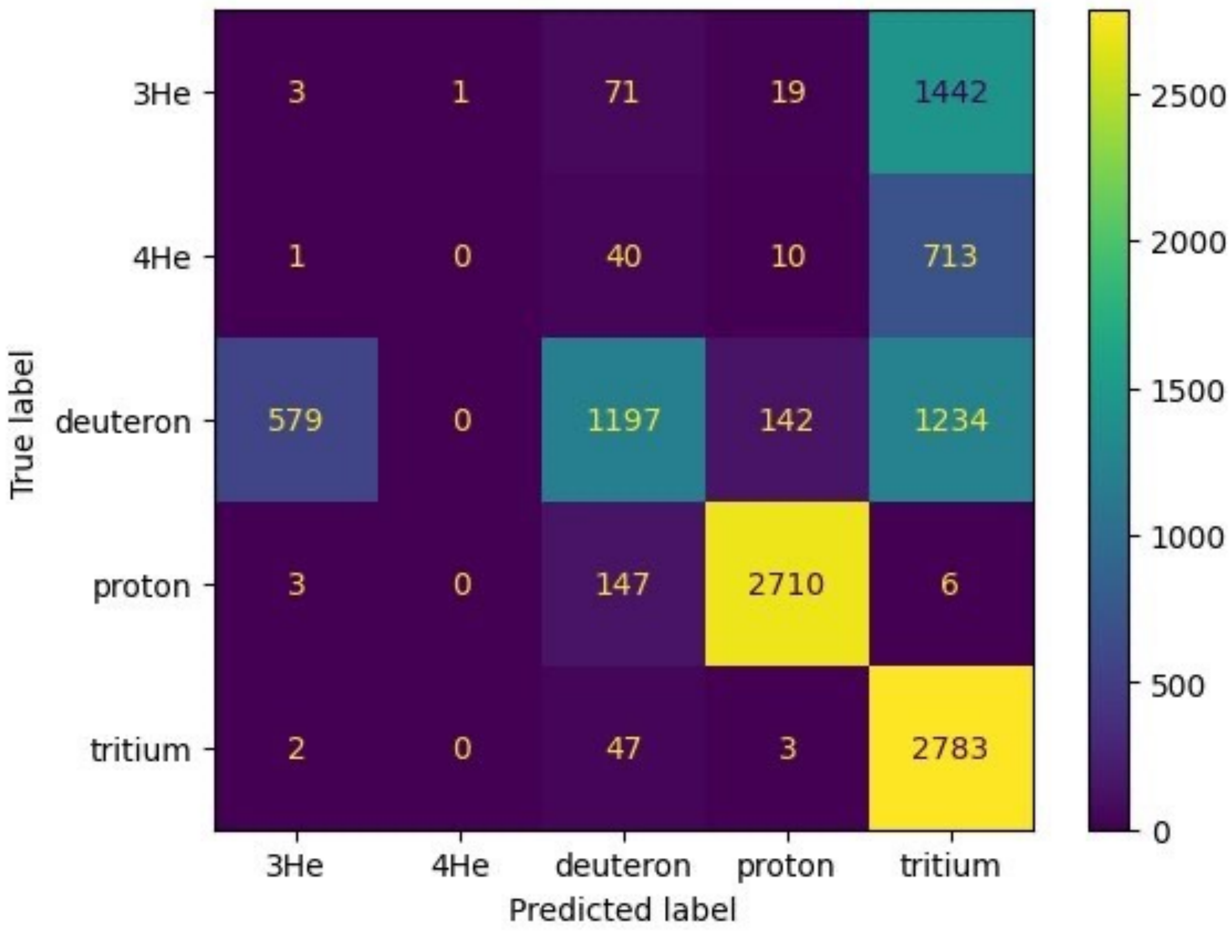
- For the normal model, the inference time is around 15µs ↦ 60KHz.
- For a typical event, with 1 proton track with a current of 487.5:
  - Mean: 24 track candidates but can have up to 500 possible tracks for some events.
- For a busy event with 3 proton tracks and a current of 650:
  - Mean: 100 track candidates but can have up to 1000 possible tracks for some events.





AI for ALERT Particle IDentification

- Develop an AI to identify (classify) particles:
  - Want to identify p, ^{2}H, ^{3}H, ^{3}He, ^{4}He
  - Use an MLP with 3 hidden layers with 20 neurons each
  - The inputs are the 5 superpreclusters and the cluster in the ATOF
  - The confusion matrix shows some results for the proton and the tritium



- Work done by U. Weerasinghe. Still in progress.

Summary and Outlook

- We have developed an MLP for track finding for ALERT:
  - Used luminosity simulation to simulate a more realistic background than randomly picking up some wires.
  - Evaluated the model's efficiency and purity as a function of momentum / threshold / current.
  - Compared different hyperparameters for the model.
  - Compared conventional and AI algorithms for track finding.
  - Worked on a classifier for the PID
- Remaining possibilities:
  - Improve the model by adding the ADC for each hit (sum the ADC for the superpreclusters)
  - Use the angle between two consecutive superpreclusters
  - Work on the classifier to improve its performance