Measuring the neutron-argon inelastic cross-section at the 2x2

2x2 First Analysis Meeting (03/27/2024)

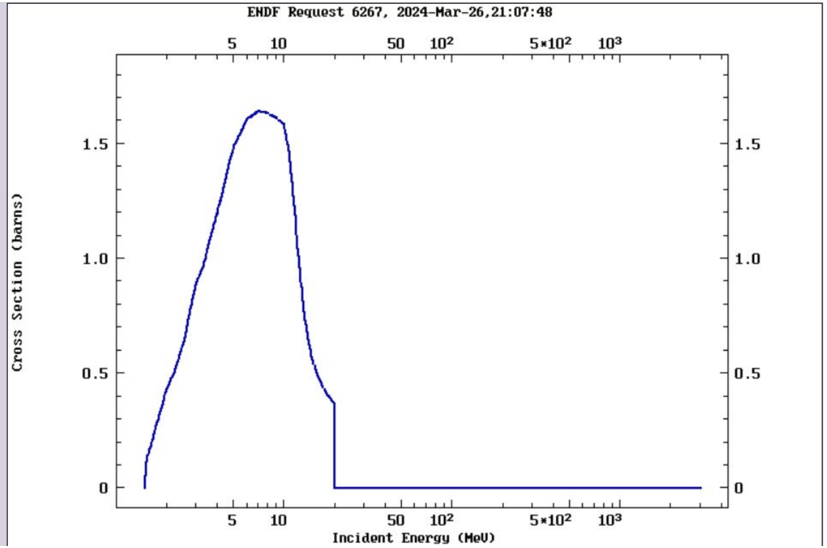
Presented by *Nicholas Carrara* on behalf of the ***nAr-inelastic xsec group***

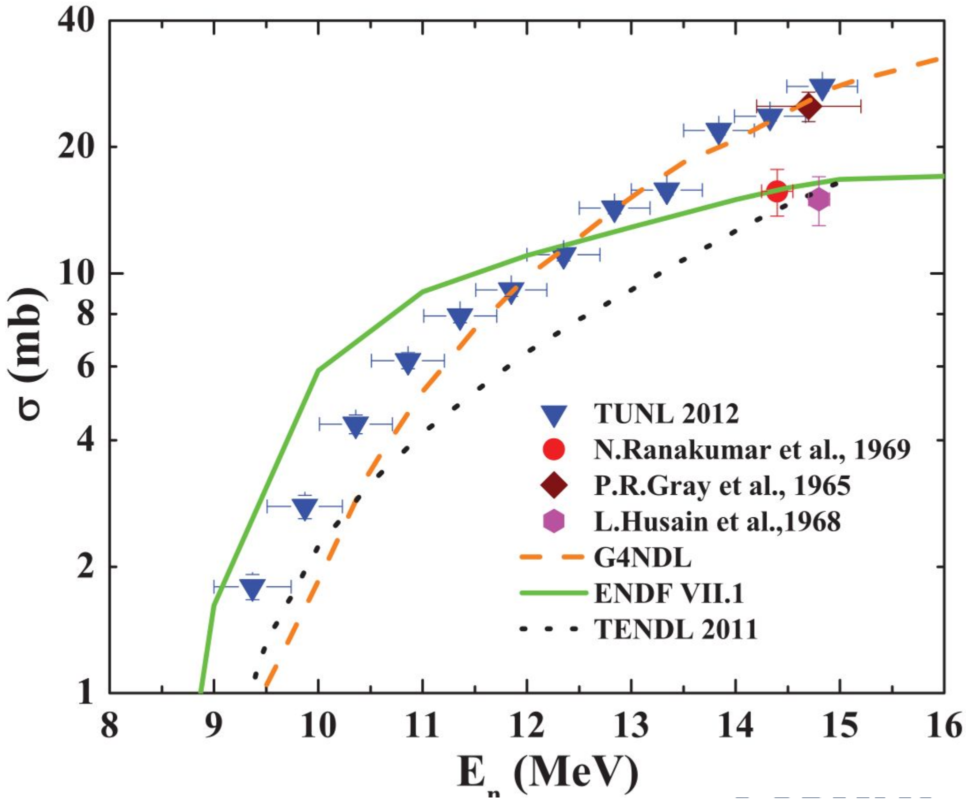
DEEP UNDERGROUND NEUTRINO EXPERIMENT

UC DAVIS
UNIVERSITY OF CALIFORNIA

n-Ar Inelastic Cross-section

- ENDF only goes up to 20 MeV.
- Only data on argon is from 2012: (https://journals.aps.org/prc/abstract/10.1103/PhysRevC.86.041602)

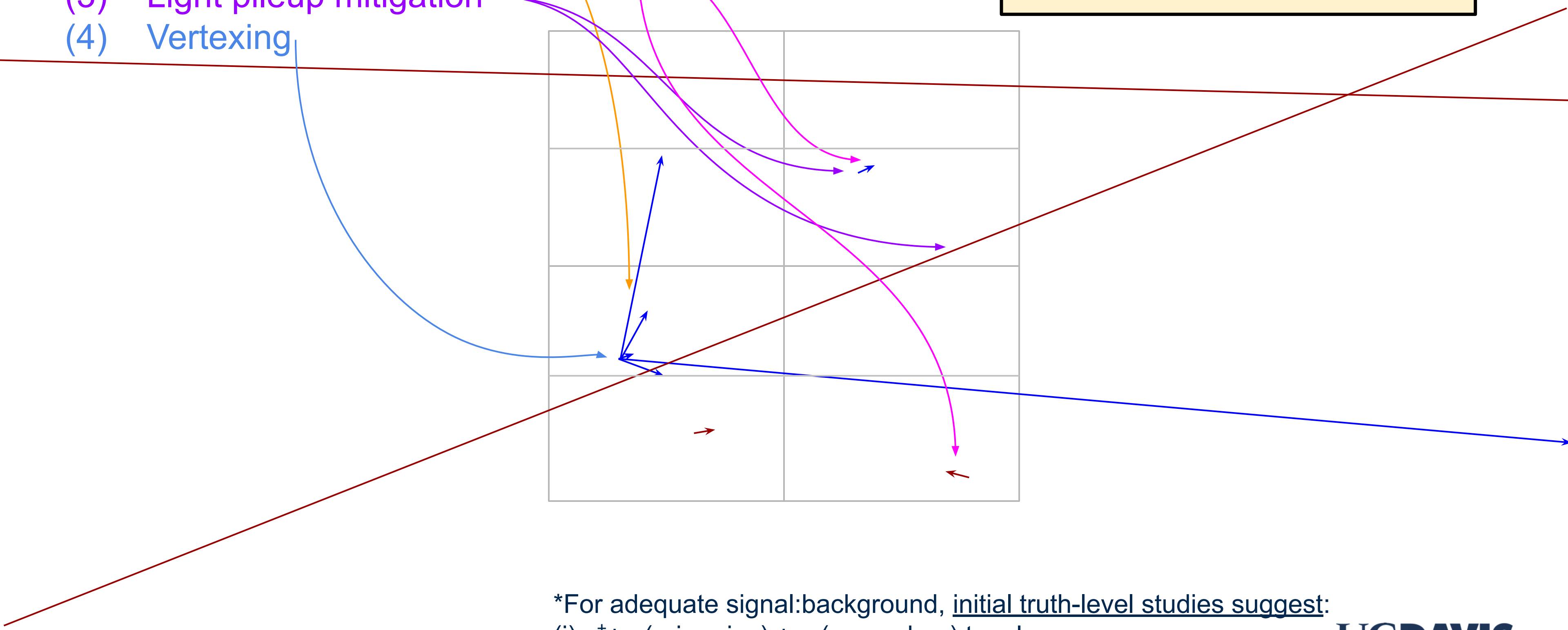




How do we measure n_TOF?

(1) Neutrino identification
(2) Proton identification/endedness
(3) Light pileup mitigation
(4) Vertexing

Analysis targets:
(1) Event-by-event Primary Neutron Kinetic Energy Spectrum
(2) n-Ar Differential Cross Section



Beam neutrinos

*For adequate signal:background, <u>initial truth-level studies suggest</u>:
(i) μ^{+}+n (primaries) + p (secondary) topology
(ii) neutrino vertex and p reside in <u>different</u> TPCs/modules

UCDAVIS

Analysis Plan

1. Use a convolutional neural network to classify events by interaction type.
2. Use a convolutional neural network (*BlipSegmentation*) to semantically label points according to Arrakis.
3. Collect track points and feed them into the hierarchical clustering algorithm (*BlipNet*).
   a. For every node in the resulting dendrogram, feed the results into a point-cloud neural network which assigns a score for how "proton-like" the cluster is, as well as how pure.

(1) Neutrino identification
(2) Proton identification/endedness
(3) Light pileup mitigation
(4) Vertexing

UCDAVIS

Arrakis

Arrakis (https://github.com/Neutron-Calibration-in-DUNE/ArrakisND/tree/develop) is a framework for generating:

- High-level variables associated to charge and light data which can be used to train ML algorithms and benchmark performance.
- CAF “truth” objects which can be used to benchmark reconstruction algorithm performance.
- Running Arrakis generates a new h5 file with “FLOW” replaced with “ARRAKIS”.

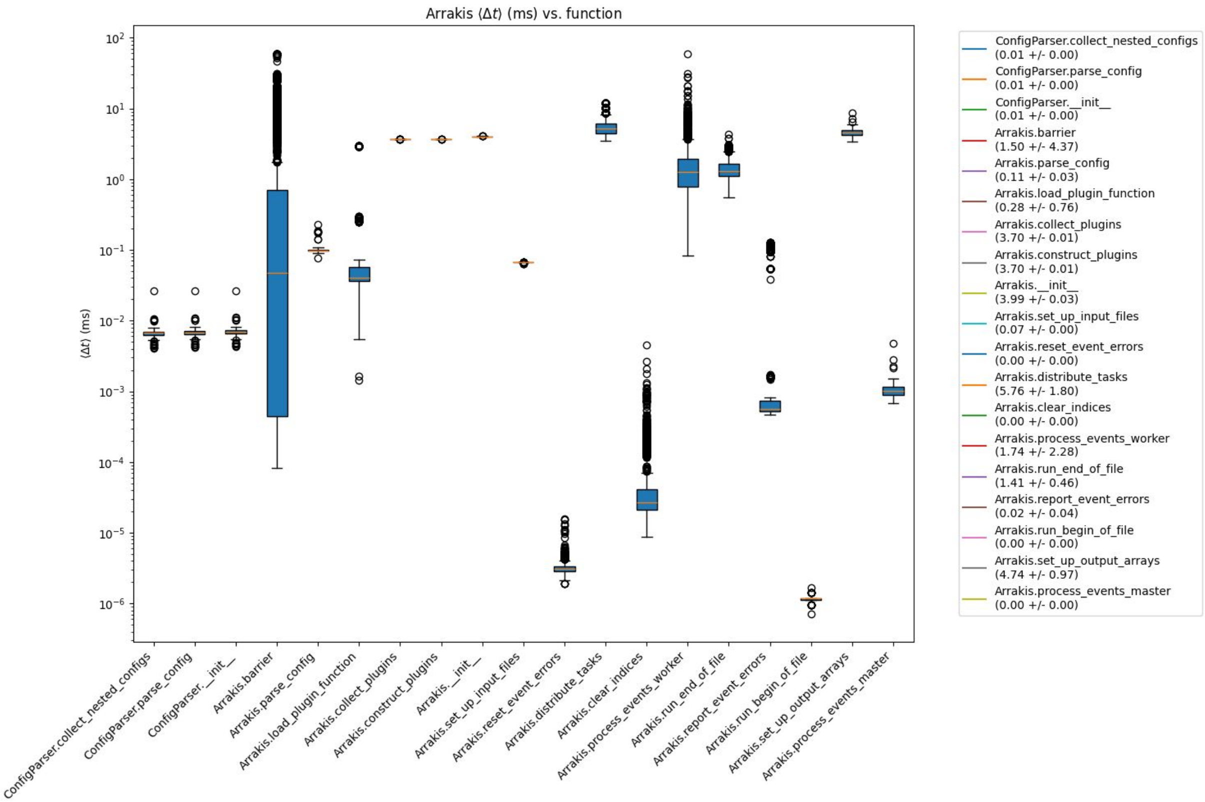
```
new_charge_data_type = np.dtype([
    ('event_id', 'i4'),
    ('topology', 'i4'),
    ('particle', 'i4'),
    ('physics_micro', 'i4'),
    ('physics_macro', 'i4'),
    ('unique_topology', 'i4'),
    ('vertex', 'i4'),
    ('tracklette_begin', 'i4'),
    ('tracklette_end', 'i4'),
    ('fragment_begin', 'i4'),
    ('fragment_end', 'i4'),
    ('shower_begin', 'i4')
])
```

```
track_data_type = np.dtype([
    ('event_id', 'i4'),
    ('track_id', 'i4'),
    ('tracklette_ids', 'i4', (1, 20)),
    ('start', 'f4', (1, 3)),
    ('end', 'f4', (1, 3)),
    ('start_hit', 'f4', (1, 3)),
    ('end_hit', 'f4', (1, 3)),
    ('dir', 'f4', (1, 3)),
    ('enddir', 'f4', (1, 3)),
    ('dir_hit', 'f4', (1, 3)),
    ('enddir_hit', 'f4', (1, 3)),
    ('Evis', 'f4'),
    ('qual', 'f4'),
    ('len_gcm2', 'f4'),
    ('len_cm', 'f4'),
    ('E', 'f4'),
    ('truth', 'i4', (1, 20)),
    ('truthOverlap', 'f4', (1, 20)),
])
```

UCDAVIS

Arrakis

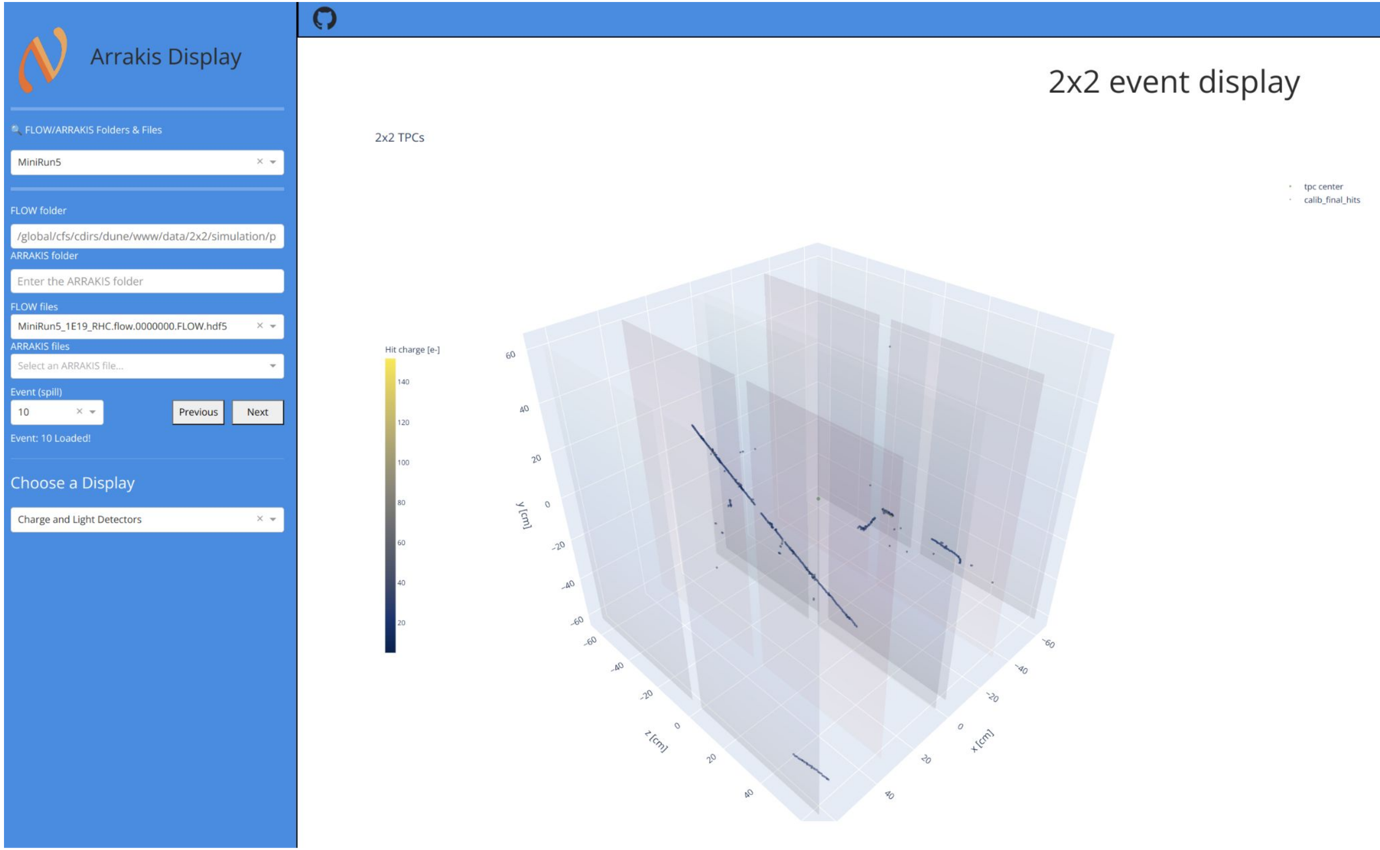
- Arrakis splits up ***files*** among jobs and ***events*** among worker nodes.
- Dedicated container on NERSC.
- Runs over all of MiniRun5 in about ***15 minutes*** (~1000 files) with 25 separate jobs (should scale linearly with number of jobs).
- Compare that to larnd-sim and flow which take approximately ***10 minutes*** per file.



UCDAVIS

Arrakis

We are also working on an event display, which can be launched following the instructions here (https://github.com/Neutron-Calibration-in-DUNE/ArrakisND/blob/develop/arrakis_nd/utils/display/README.md)

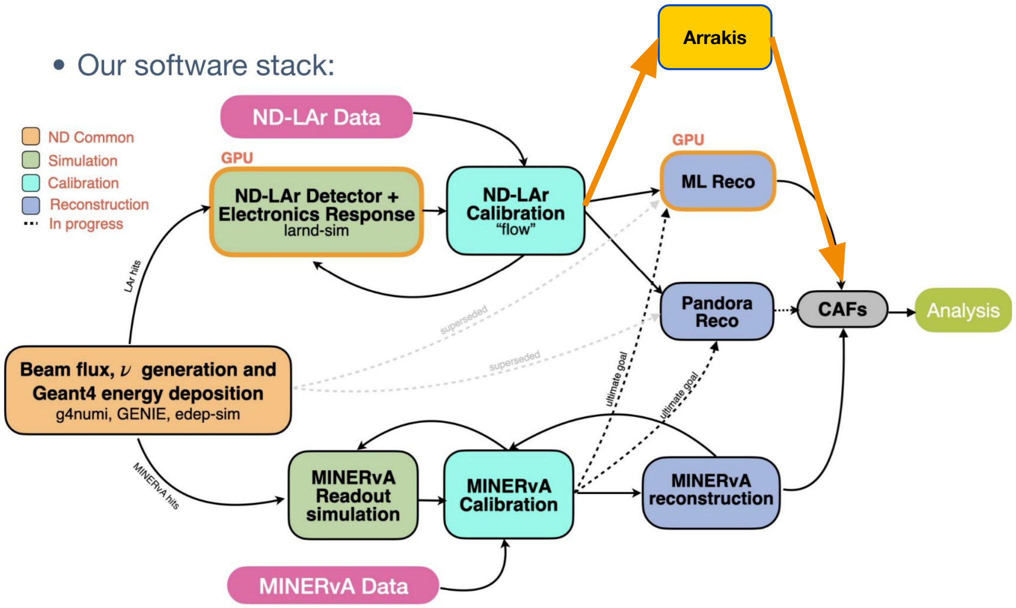


UCDAVIS

Arrakis

Next steps for Arrakis:

- Complete basic labeling logic plugins.
- Coordinate with 2x2 analysis group to add Arrakis to the **flow chain**.
- Set up some benchmark datasets to quantify performance of labeling logic.
- Make the event display available to login nodes on perlmutter.
- We have lots of ideas for what to add to the event display, but are welcome to whatever people would like to see (please give us suggestions)!



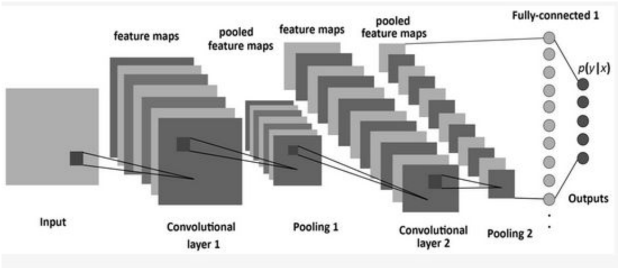
Neutrino Classification Network

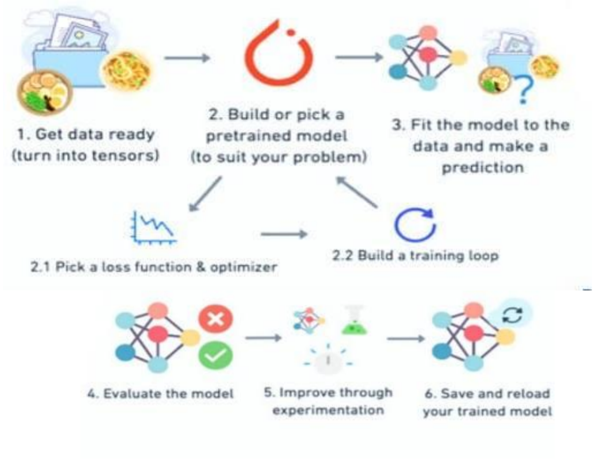
Multi-class classification Problem

Classification of interaction events

initial labels **CCe, CCmu, NC**

1. Uses a Convolutional Neural Network
2. Independent of Arrakis.
3. Create dataset for each flow file
4. Build and Evaluate the model (CNN)





Blip Segmentation

1. Updates required with new format
2. Evaluate the network with analysis tasks

Hierarchical Clustering

We want to find **protons** from a **neutron parent**.

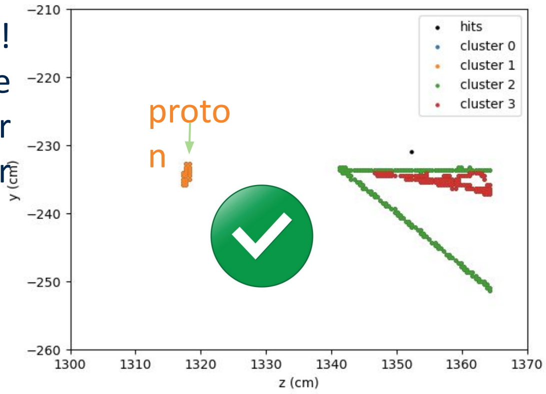
The neutron is **invisible**, so it is just a short disjoint proton track we are looking for. Current reconstruction is not good at **short tracks**.

Can we use a simple clustering algorithm to cluster these short proton tracks very well? *Very well* means that the clusters contain **all** of the proton track and **nothing but** the proton track (efficient and pure).

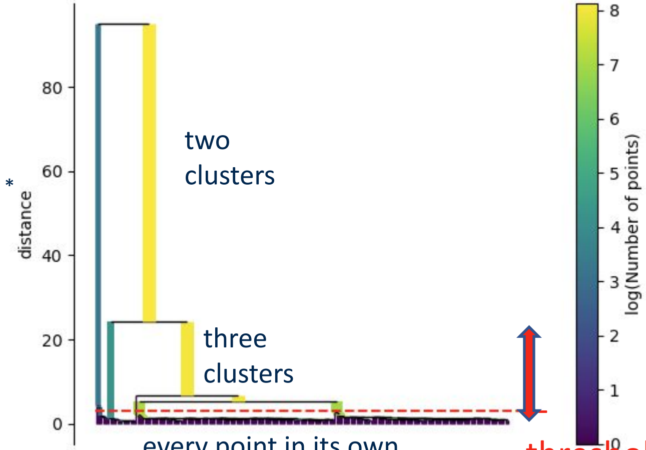
Idea: hierarchical clustering, where we walk over the **single linkage tree** and try to find a sweet spot

good!
all proton hits in one cluster
no other hits in the cluster

NB. after doing all this, we still need to figure out which clusters *are* protons



all points in one cluster



every point in its own cluster

threshold=3

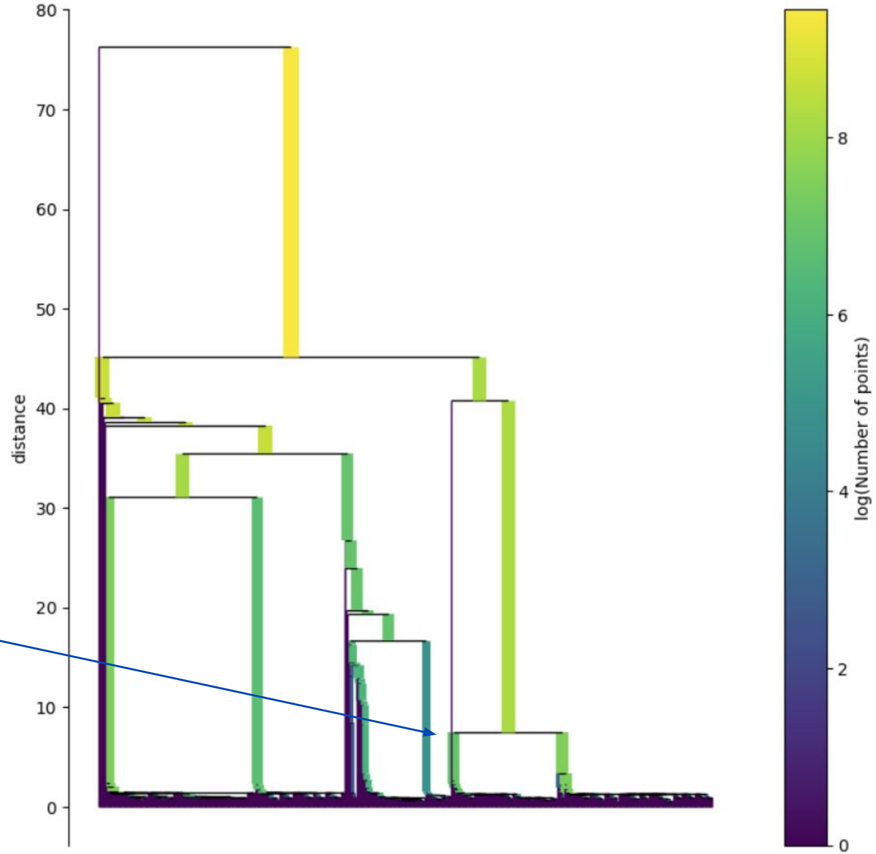
UCDAVIS

*distance = mutual reachability distance of the minimum spanning tree

How well can we do in principle?

Assuming that tracks are correctly labeled.

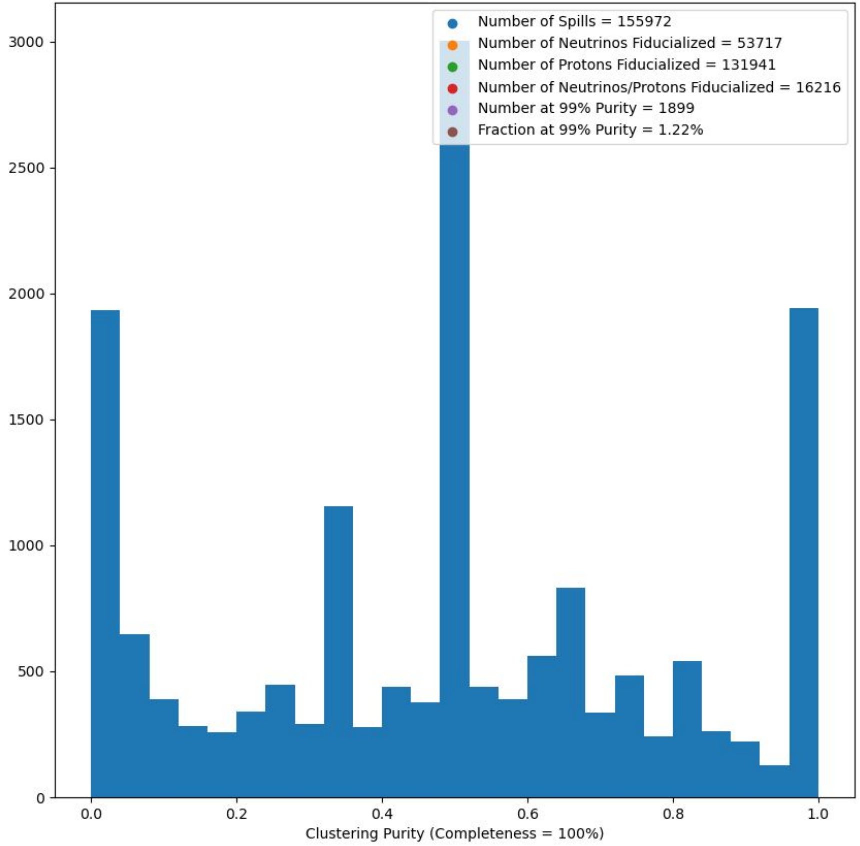
1. Create a dendrogram from a distance metric.
2. For each nar-inelastic proton:
   a. Traverse from the bottom of the dendrogram until we reach the node with completeness = 1.0.
   b. Record the “best_purity” at this node.



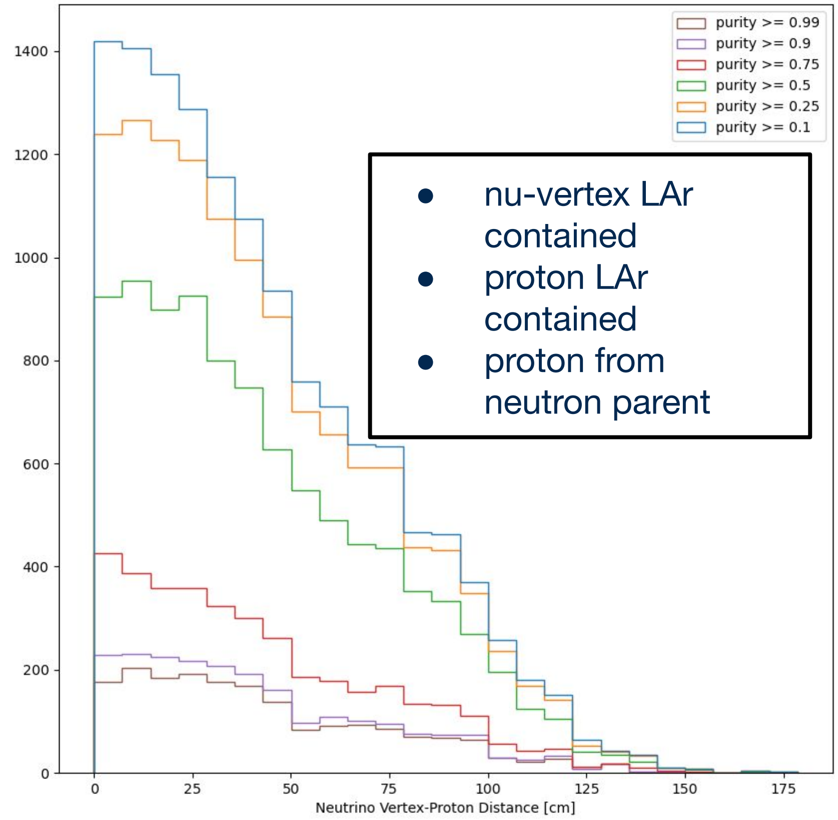
How well can we do in principle?

If our target are clusters with high purity and completeness (**purity ~ 99%**), then MiniRun5 predicts:

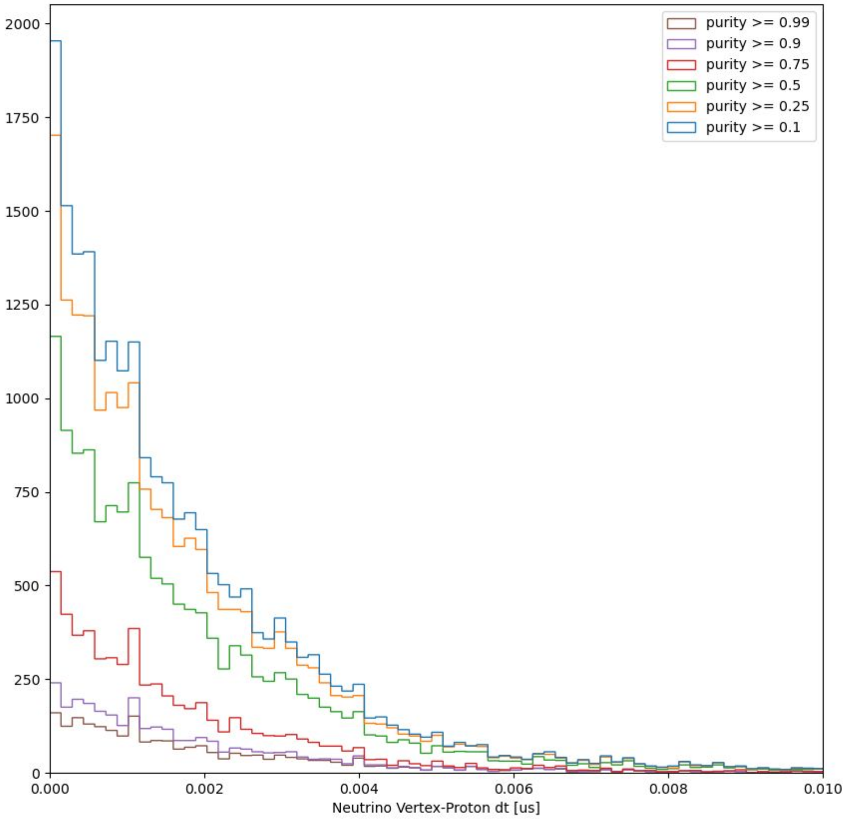
- An event rate of **1.22%**
- For three days of running (1E19 protons), we expect roughly **2K** events with isolated proton tracks from neutron-inelastic interactions.
- Need to see how this diminishes with an optical detector constraint.



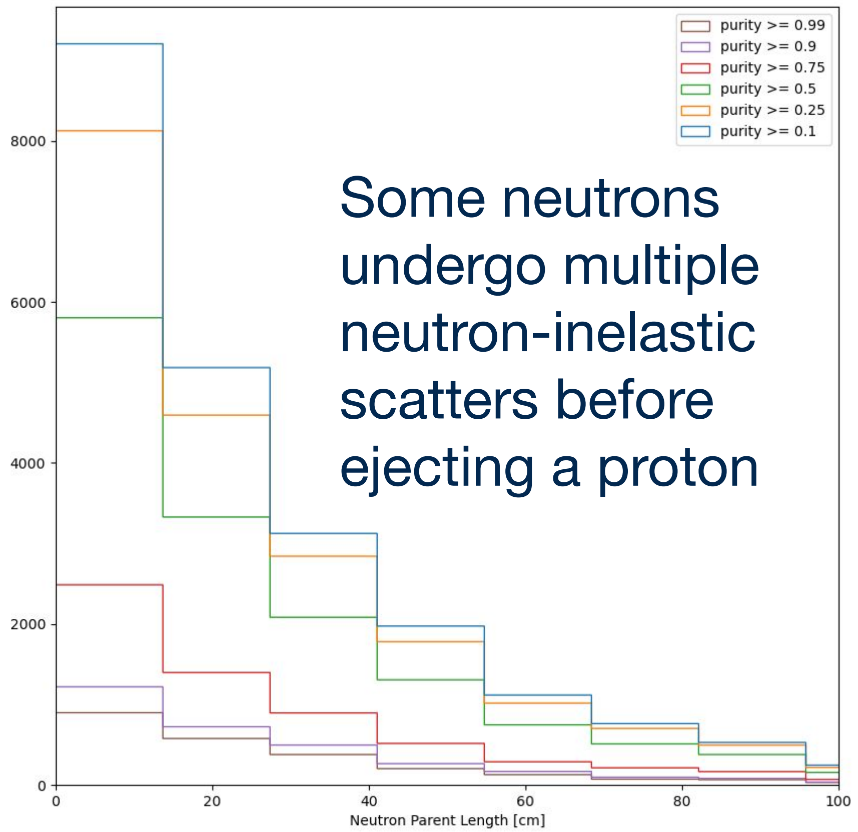
How well can we do in principle?

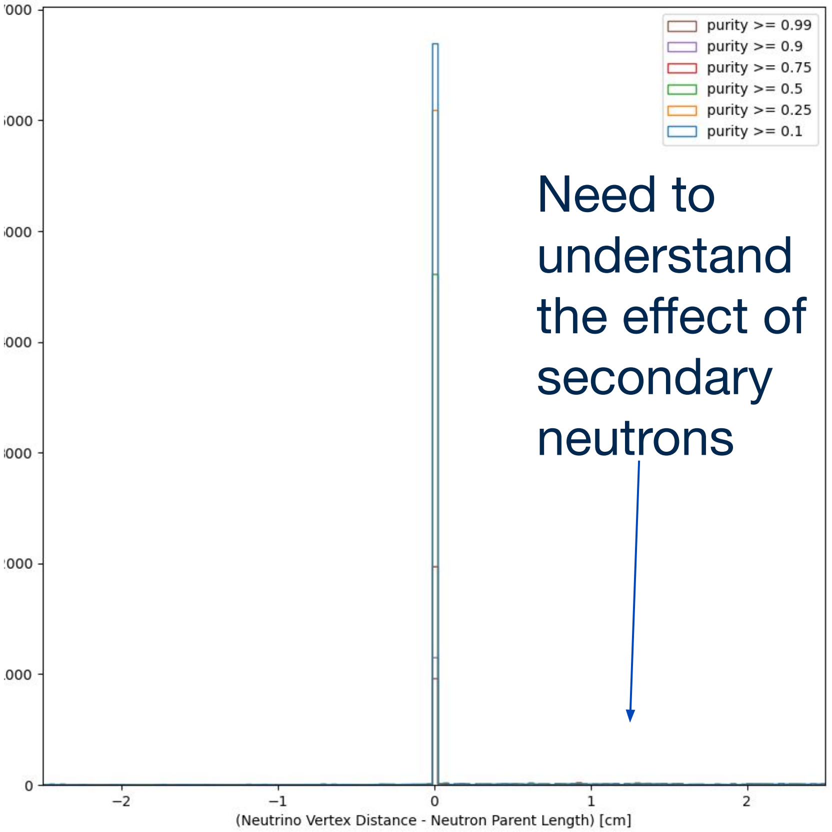


- nu-vertex LAr contained
- proton LAr contained
- proton from neutron parent

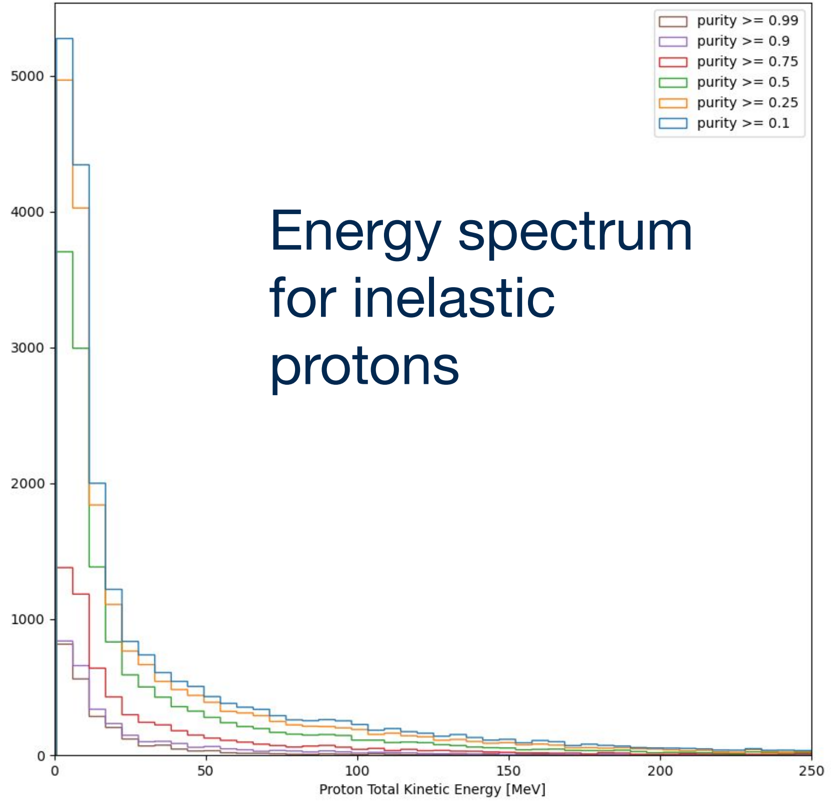


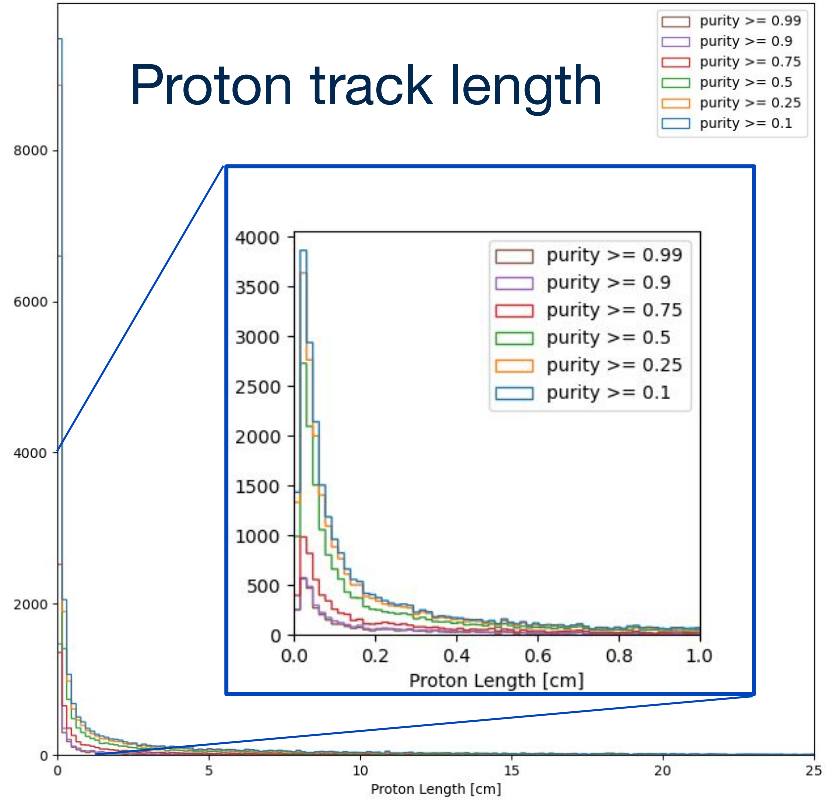
How well can we do in principle?





How well can we do in principle?

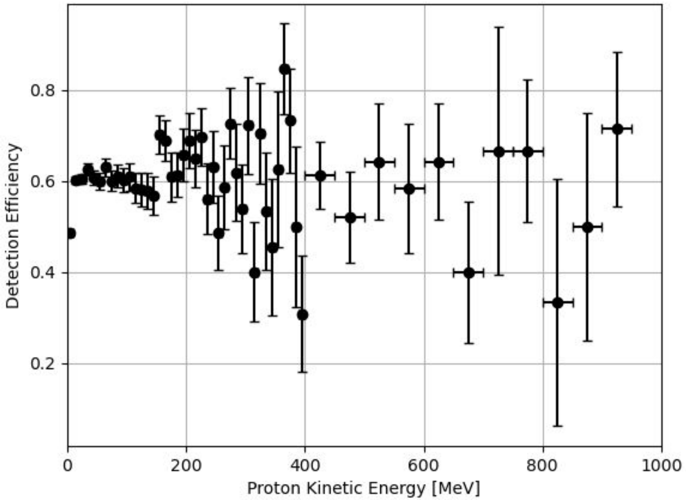




CRO Proton Detection

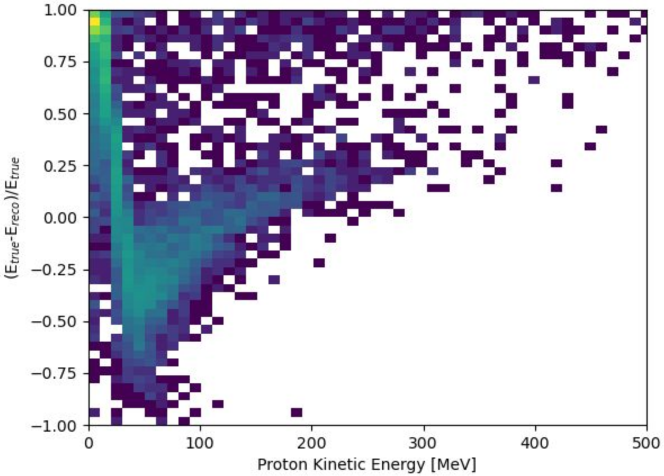
Truth selection: require LAr totally contained* proton hailing from neutron parent; no neutrino vertex criteria

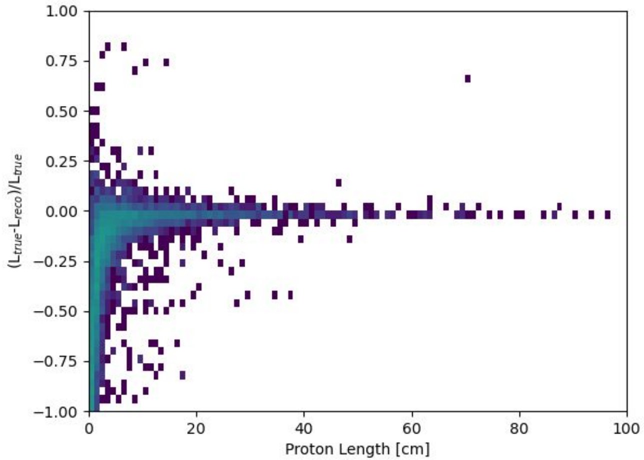
Reconstructed topology and detection efficiency strongly tied to LArPix field, electronics response



12,922 detected protons in 1E19 POT

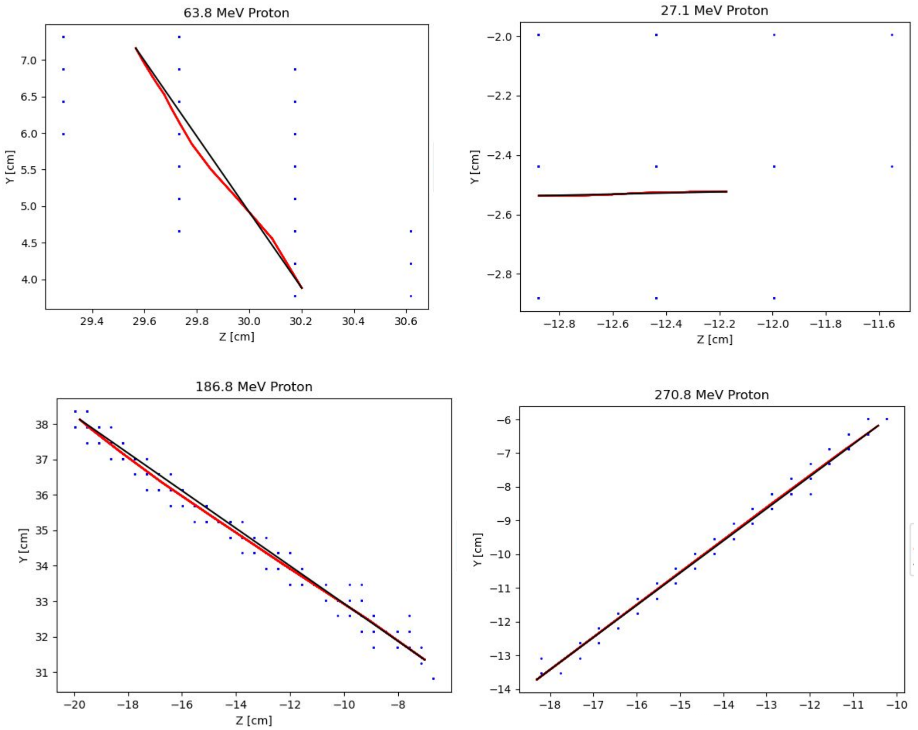
55% detection efficiency, where proton is detected if at least one hit is reconstructed





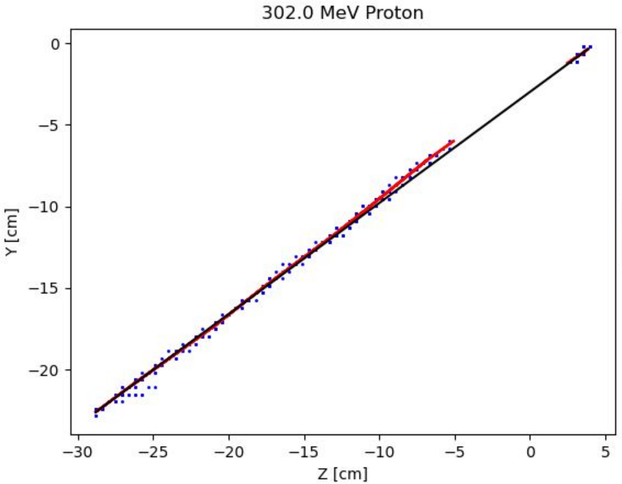
*Geant4 proton trajectory endpoints must reside in the LAr active volume

CRO Detection Features



'calib prompt hits'
backtracked segments
truth trajectory

Opportunity for improved timing resolution with proton tracks spanning multiple active volumes



Unsurprisingly, improved track angle fidelity with increased track length

UCDAVIS

Next Steps

Viability Studies:

- Need to understand effects of secondary neutrons on energy resolution.
- Evaluate the incidence and impact of light pileup and implications on signal yield and timing resolution

Analysis/Algorithms:

- Develop BlipSegmentation to reconstruct:
  - tracks
  - vertices
  - neutrinos
- Develop BlipNet for classifying pure proton tracks.
- Develop end-to-end analysis software for deployment on data.

UCDAVIS