Finding secondary vertices using deep learning in Pandora

Andy Chappell

04/07/2023

LBNF/DUNE UK July 2023

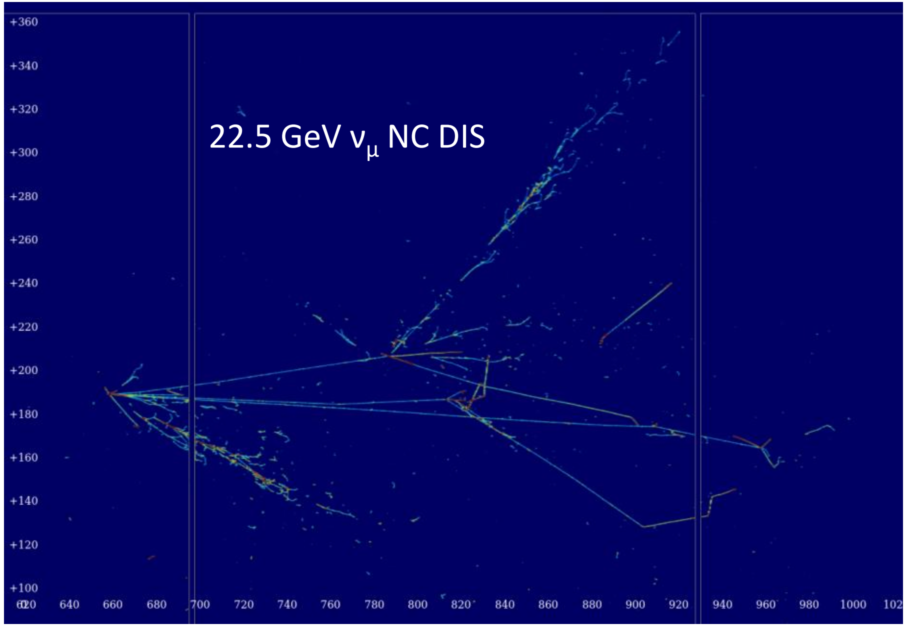
DUNE DEEP UNDERGROUND NEUTRINO EXPERIMENT

Pandora

WARWICK

Overview

- The neutrino interaction vertex is a critical feature for reliable reconstruction of far detector events

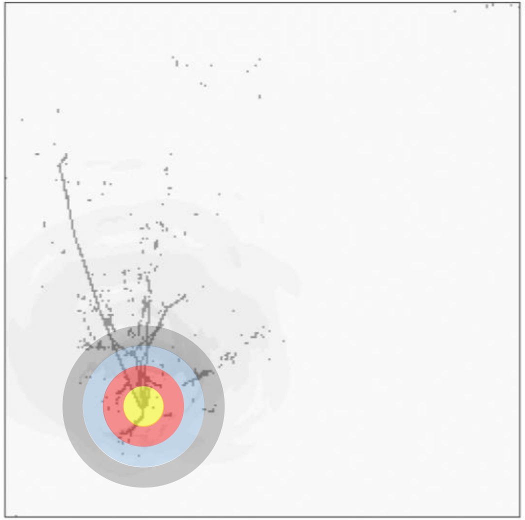


- It determines how hits are clustered into particles and sets a starting point for the hierarchical relationships between them

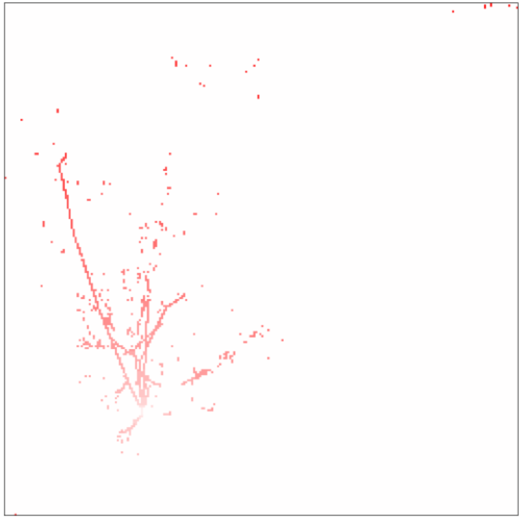
- It would be useful if we could identify secondary vertices and use these to guide reconstruction decisions for downstream interactions

Primary Vertexing Concept

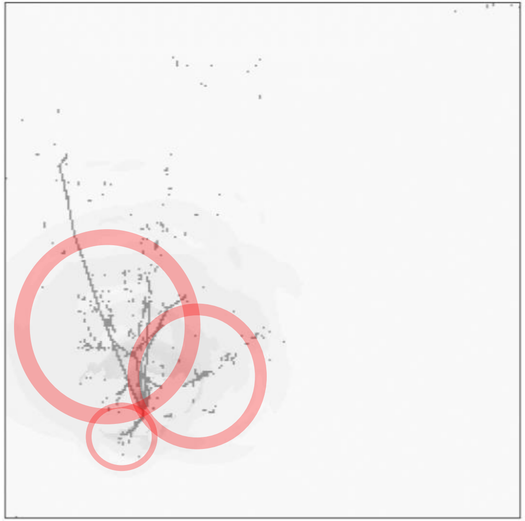
Hits are assigned a class according to distance from true vertex



Network trained to learn those distances from input images

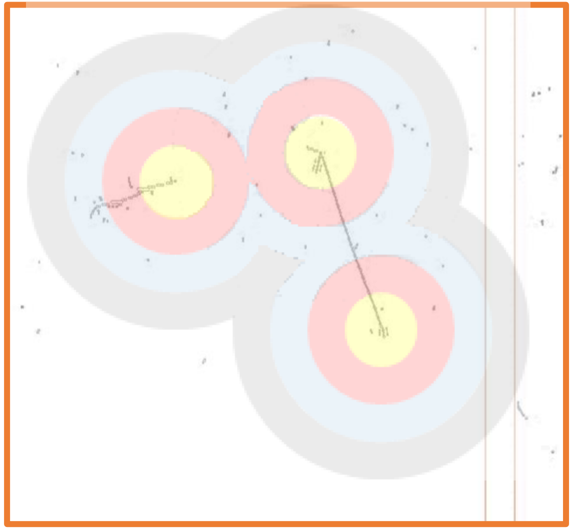


Network infers hit distances and resultant heat map isolates candidate vertex

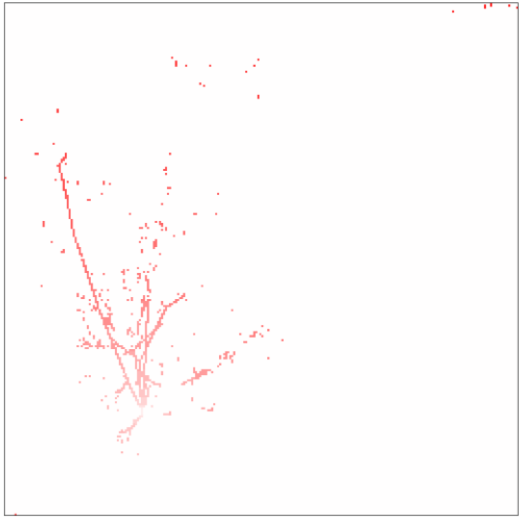


Moving to secondary vertexing

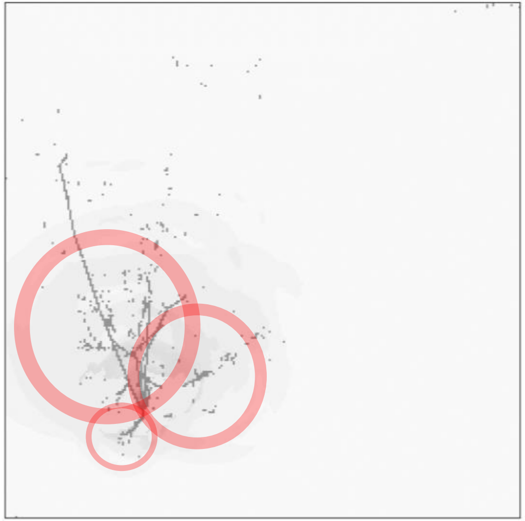
Hits are assigned a class according to distance from true vertex



Network trained to learn those distances from input images



Network infers hit distances and resultant heat map isolates candidate vertex



Extracting a 3D vertex from 2D views

- Finding multiple vertices is, of course, a harder problem
- For primary vertexing:
  - One network per view - independent candidate identification
  - Use knowledge of wire plane geometry to correlate identified candidates
  - Extracting a coherent 3D vertex from view correlations not too tricky
- For secondary vertexing:
  - Currently leveraging the same approach – one network per view
  - Inference step more challenging due to increased number of features to identify
  - Combinatorics and reduced precision makes extracting reliable, coherent 3D vertices harder
- Can we process the three heat maps simultaneously to get vertex candidates that are already consistent across views?

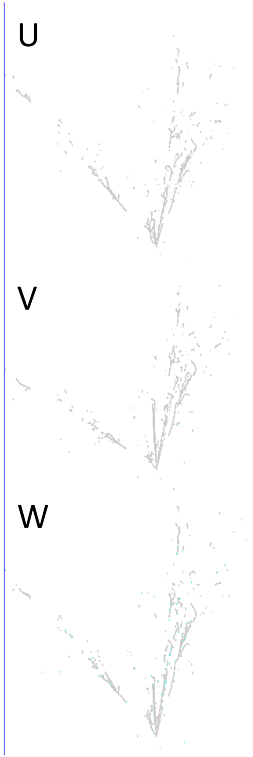
Extracting a 3D vertex from 2D views

- Can we process the three heat maps simultaneously to get vertex candidates that are already consistent across views?
  - With images of U, V and W views this is difficult
  - Pixels aren't hits and there isn't a clear one-to-one-to-one correspondence between pixels in different views
- Can we work with orthogonal views?
  - Extract clear 3D hits* from 2D views at start of reconstruction
  - Project into XZ, XY and YZ views for vertexing network
  - Simultaneous processing of heat maps becomes trivial as pixel pairs in two views have clear corresponding pixel in third view

* Also opens up possibility for graph networks, for example

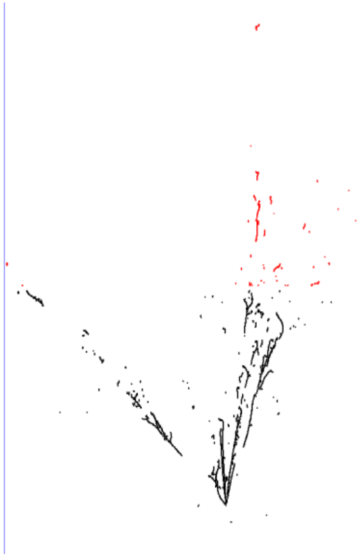
Creating 3D hits

- Pandora’s reconstruction creates 3D hits at the end of the reconstruction chain
- Uses common x-coordinate to correlate clusters in the 2D views
- We don’t have clusters when the vertex network runs, so we need to extract 3D hits directly from 2D hits
- Hypothesis is that the network need only see topological structure, so we’ll only pass it 3D hits inferred with high confidence



Identifying hit relationships

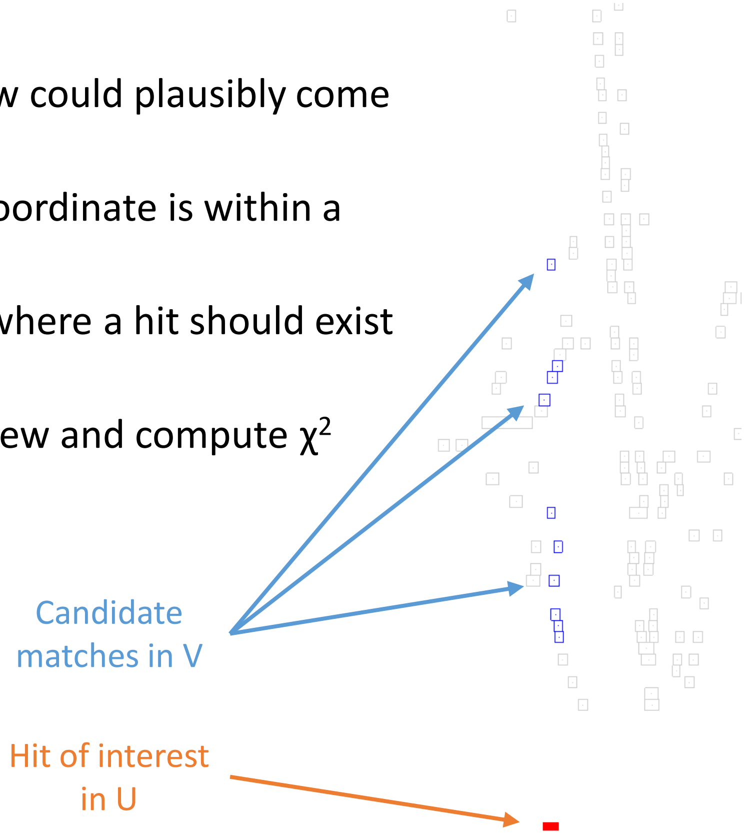
- Split the event into TPC child volumes
  - Reduces combinatorics for high hit multiplicity events (especially for isochronous trajectories)



Different colours represent different child volumes within a TPC

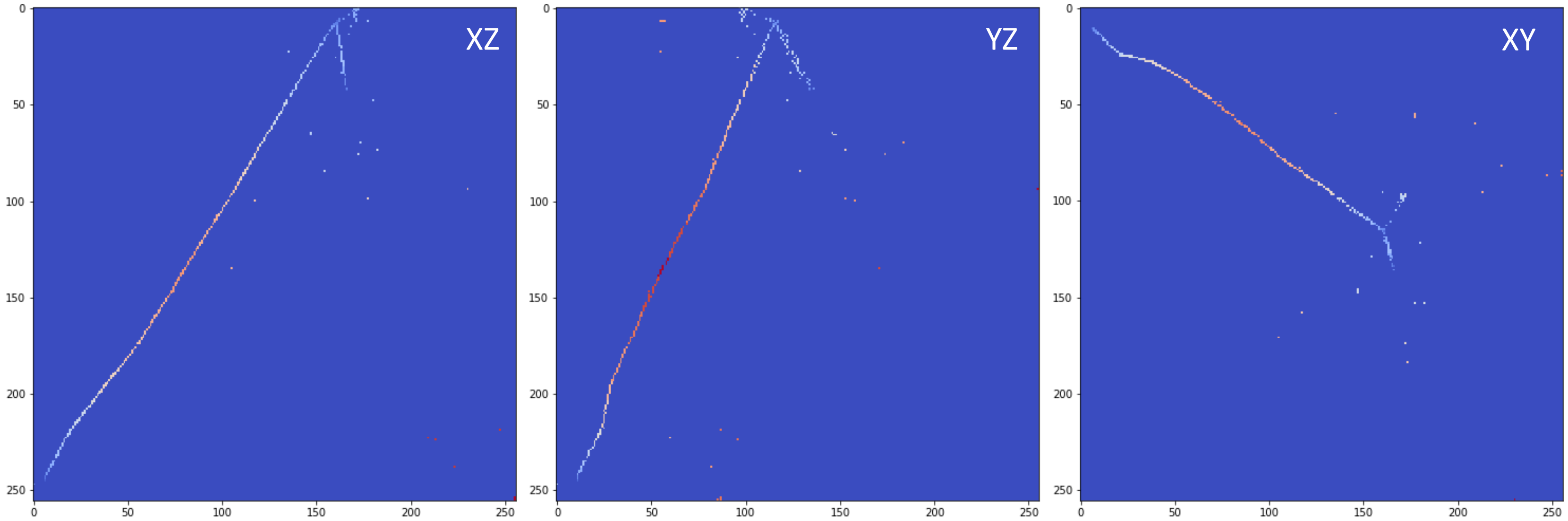
Identifying hit relationships

- In each volume determine which hits in each view could plausibly come from the same 3D hit based on drift coordinate
- A hit is potentially related to another if its drift coordinate is within a region proportional to the partner hit's hit width
- Use coordinate pairs in two views to determine where a hit should exist in the third view
- Find the hit closest to that location in the third view and compute χ^2 to see if the hit should be accepted

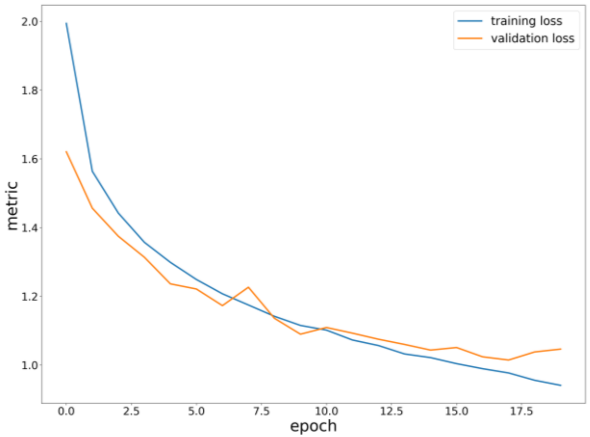


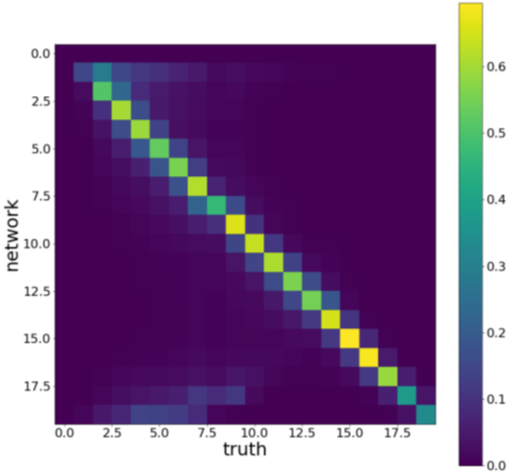
Orthogonal projections

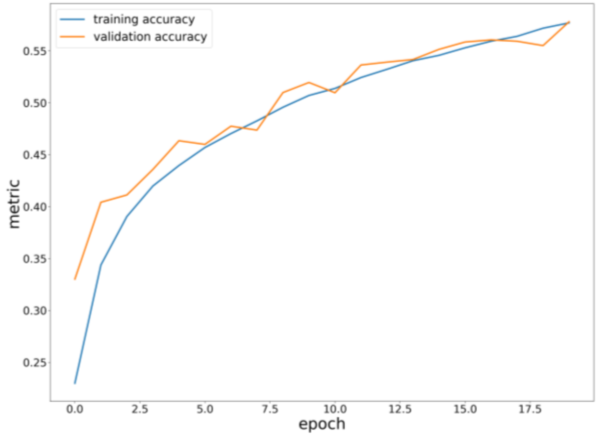
- Example set of 3D hits projected into orthogonal views
- Technique appears viable
- Blue hits closer to a vertex, red hits farthest away



Pass 1 training results



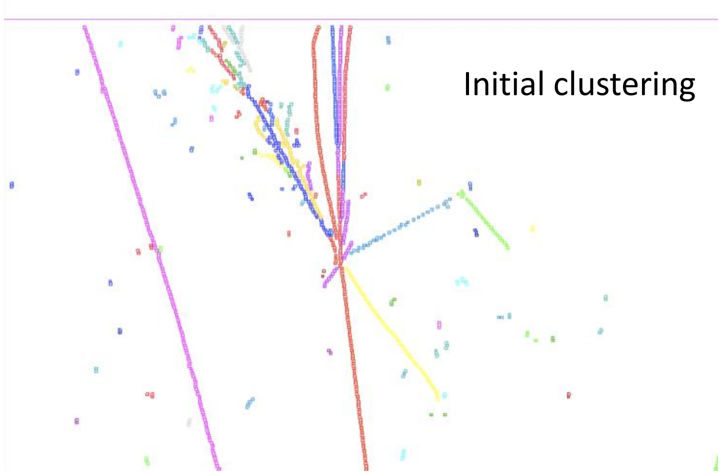




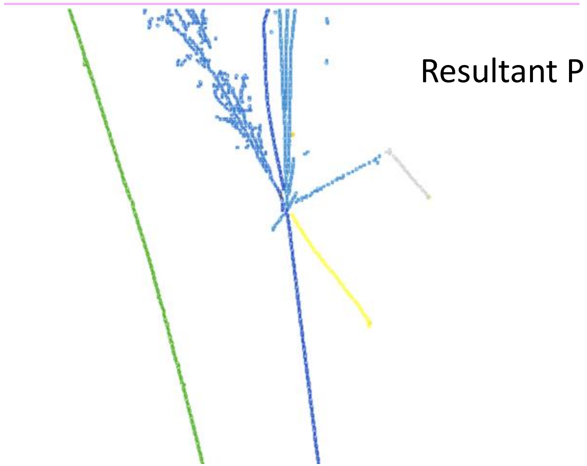
- XZ view training result shows reasonable class accuracy
  - Inference voting system quite robust to errors
- Errors largely true class adjacent
- Appears to be scope for improvement with more training
- Currently working on modifications to inference code to support simultaneous view processing

Using secondary vertices

- Pandora’s current reconstruction can sometimes struggle with complex secondary vertex regions
- Early clustering errors can result in incorrect merging



Initial clustering

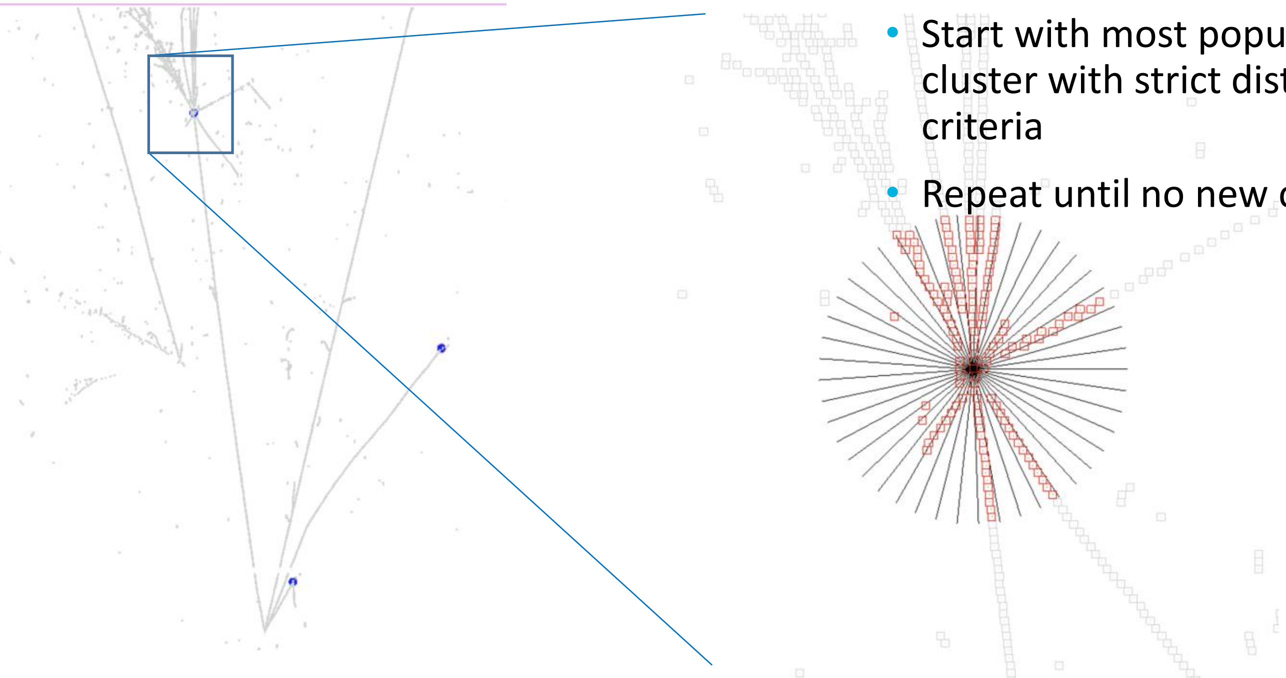


Resultant PFPs

Provisional algorithm

- Cheated secondary vertex identification
- New vertex-anchored clustering algorithm

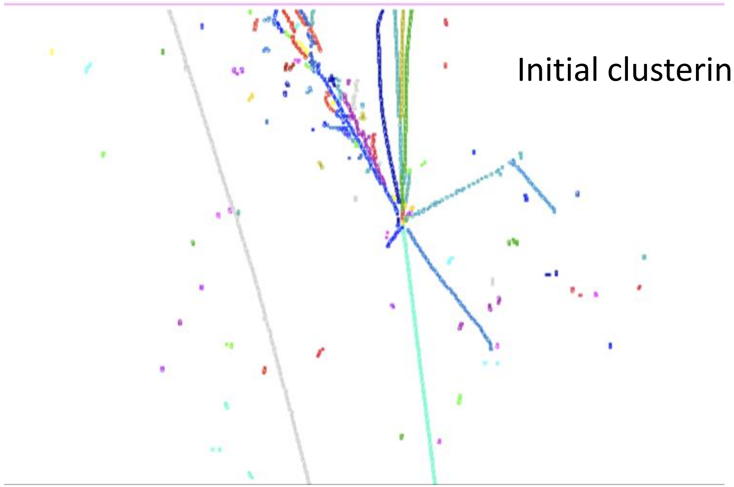
- Isolate 10 cm radially about a vertex
  - Bin by angle
  - Order hits by radial distance to vertex
- Start with most populated bin and build cluster with strict distance and direction criteria
- Repeat until no new clusters formed



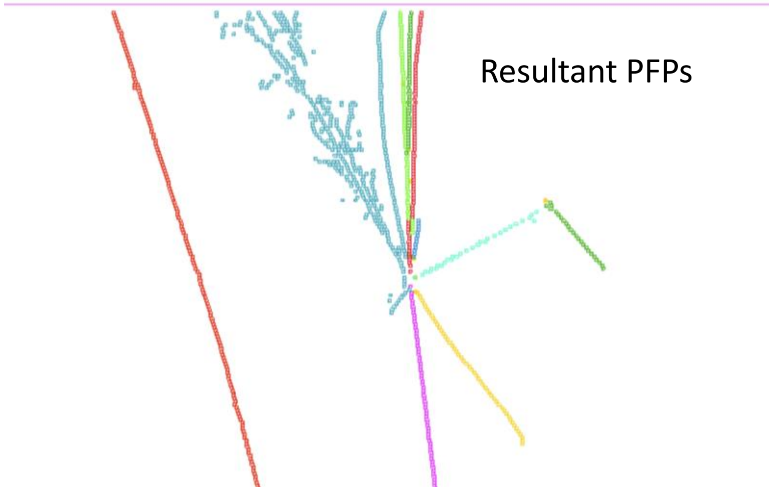
Aim is to build short stubs with good separation and let subsequent association algorithms grow them

Vertex-anchored reconstruction

- Vertex anchoring yields much better 2D clustering, no errant merging
- Resultant PFPs are improved (not perfect, but encouraging start)



Initial clustering



Resultant PFPs

Final thoughts

- Very much work in progress
- Provisional networks trained
  - Will need re-training once new production is ready
- Updating inference step to simultaneously process heat maps
- Provisional cheated algorithm shows promising use case if we can identify the vertices reliably