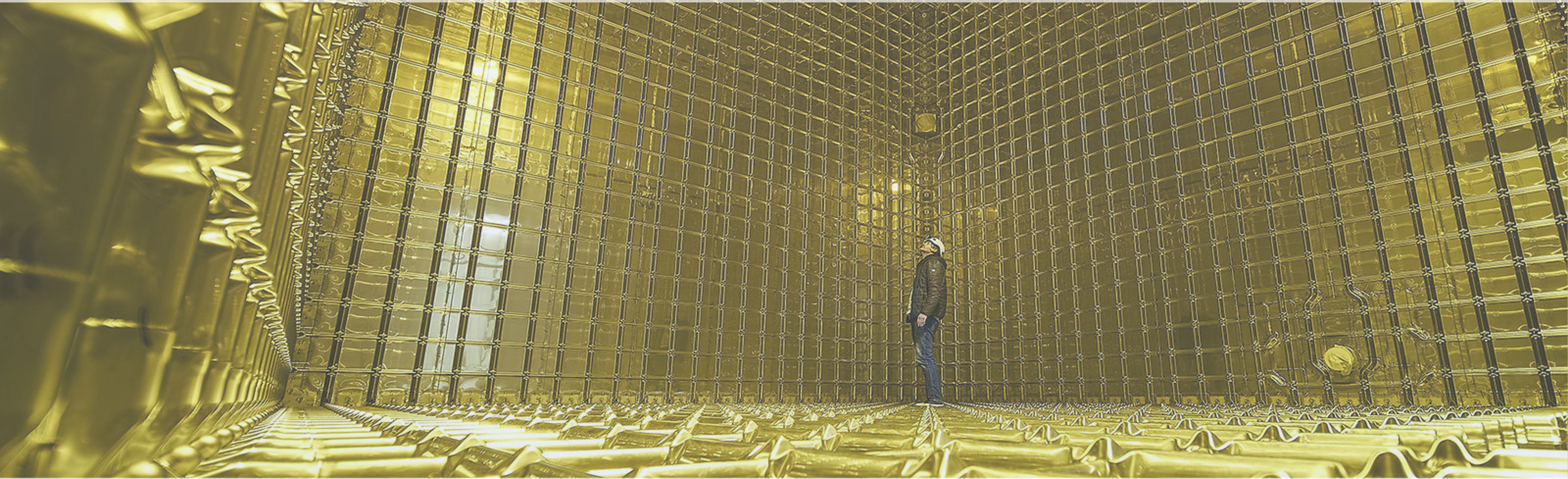
Pandora Pattern Recognition Overview



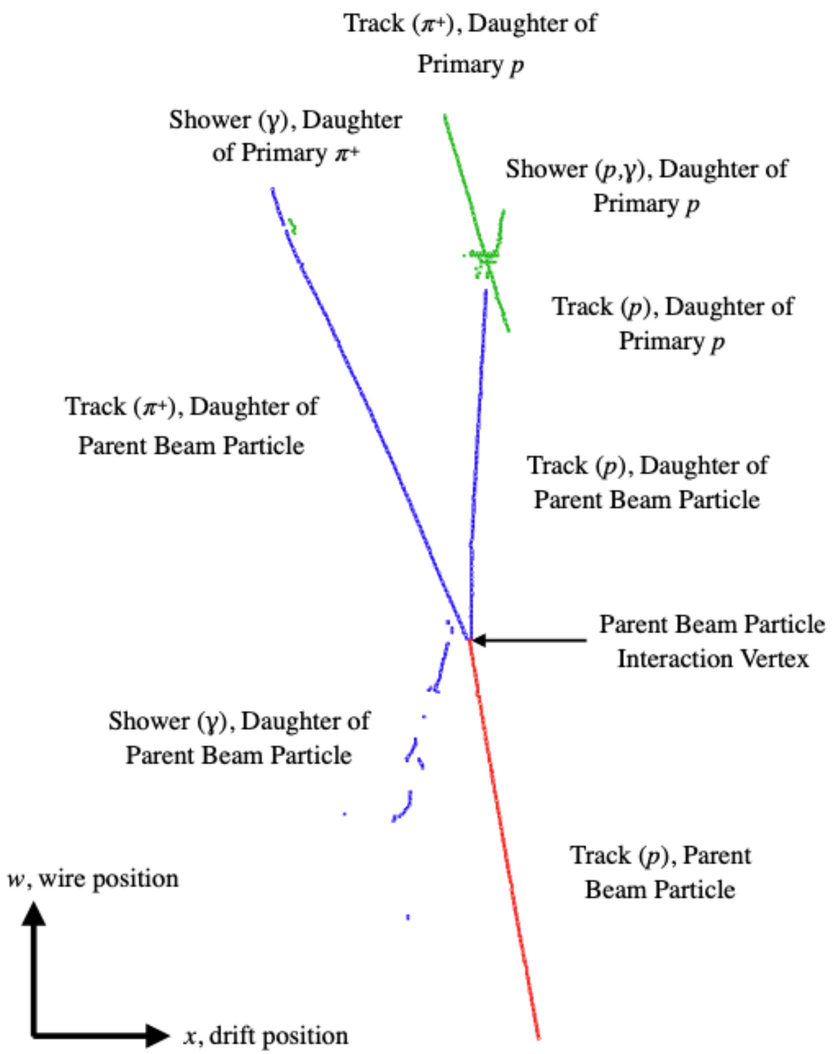
Isobel Mawby (they/them)

Lancaster University

Pandora – the hope left in the jar

'A multi-algorithm approach to pattern recognition'

- Each algorithm has a **small, specific** job and together build our output particle hierarchies
- Mainly traditional pattern recognition algorithms, but growing use of **machine learning** (see Andy C's talk)
- Algorithms are designed to be detector-agnostic ⇒ **Pandora is utilised in all LArTPC experiments**

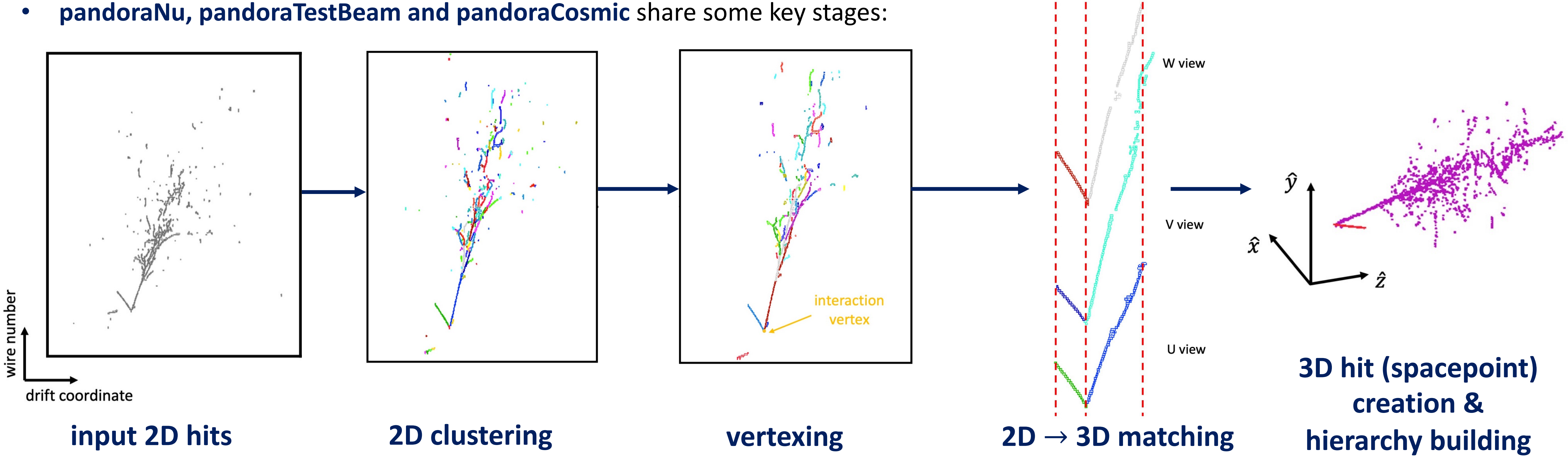


Lancaster University

Reconstruction Chains

- We utilise the following **reconstruction chains:**
  - **pandoraNu:** a neutrino-optimised reconstruction chain
  - **pandoraTestBeam:** a test beam-optimised reconstruction chain
  - **pandoraCosmic:** a cosmic ray-optimised reconstruction chain

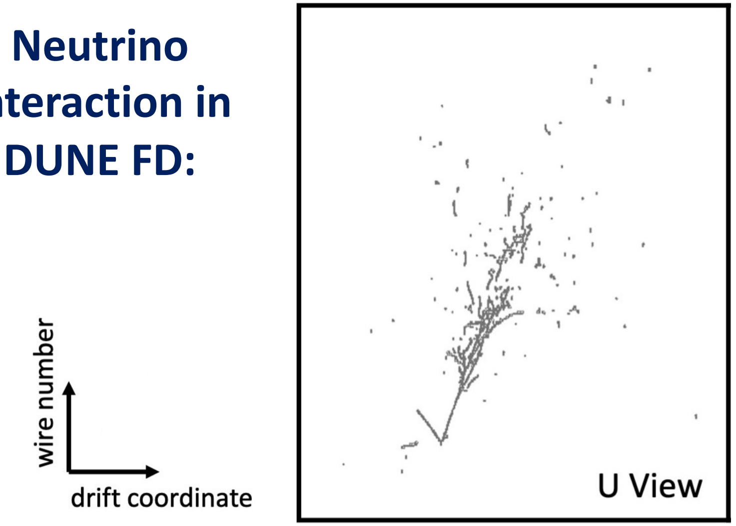
- **Analysis specific reconstruction** chains (i.e. LBL, supernova) are in development for DUNE
- **pandoraNu, pandoraTestBeam and pandoraCosmic** share some key stages:



Consolidated Reconstruction Chain

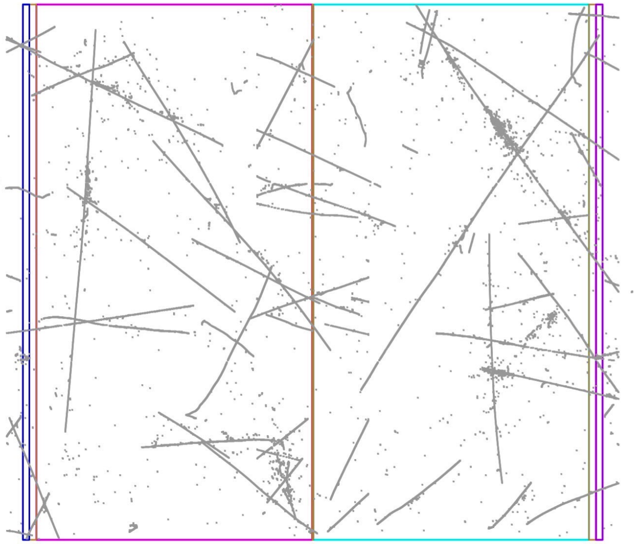
- Detectors present different challenges: high cosmic-ray flux? multiple TPC chambers?

Neutrino interaction in DUNE FD:



vs.

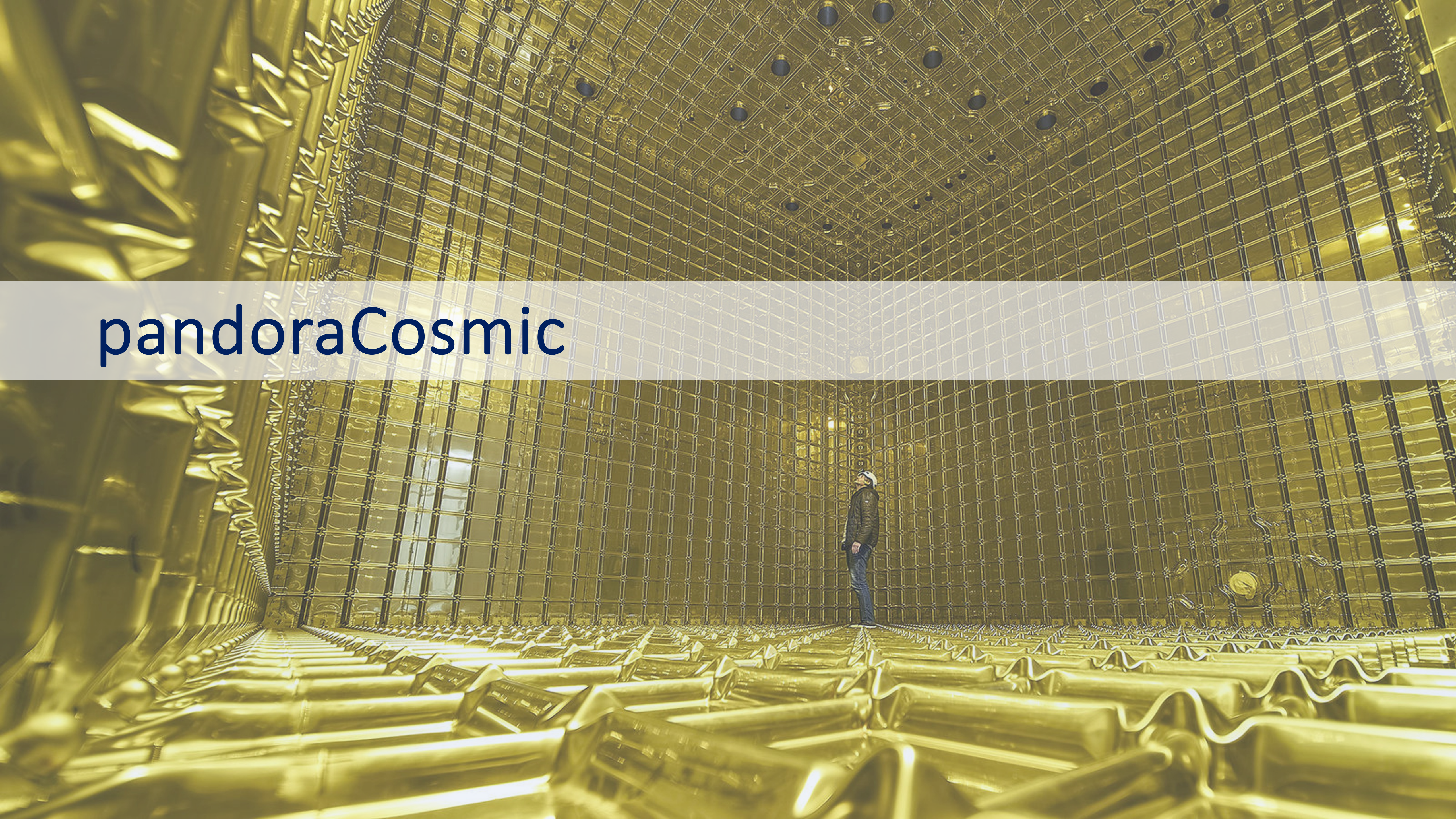
Test-beam interaction in ProtoDUNE:



- We construct different reconstruction chains for each detector **e.g. for MicroBooNE:**

2D Hits → *PandoraCosmic* → Clear cosmic ray identification and removal → Group hits of each assumed particle hierarchy i.e. *'slicing'* → *'slices' of hits* → *PandoraCosmic* / *PandoraNu* → BDT choses best output for each 'slice'

pandoraCosmic



pandoraCosmic

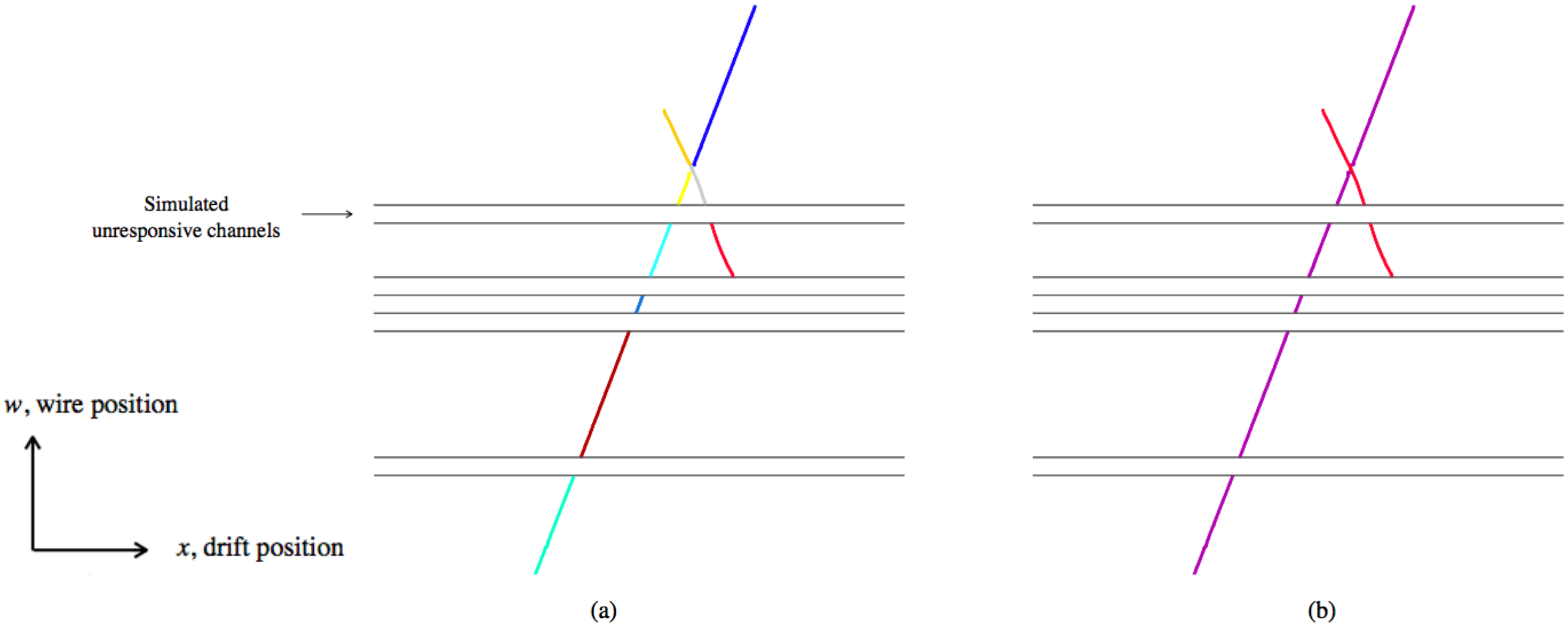
- Strongly track-orientated:
  - Cosmic-ray tracks first sought, and ‘protected’
  - Remaining clusters assumed to belong to michel electrons and delta-rays; added as children of primary muons
  - Cosmic-ray vertices placed at highest vertical-coordinate

- Let’s go through these stages in more detail…

ATTN: In the following we’re going to assume we’re working with a detector with three readout planes, which is true for MicroBooNE, ProtoDUNE-SP, DUNE-HD, SBND, ICARUS and DUNE-VD

2D reconstruction (i.e. clustering)

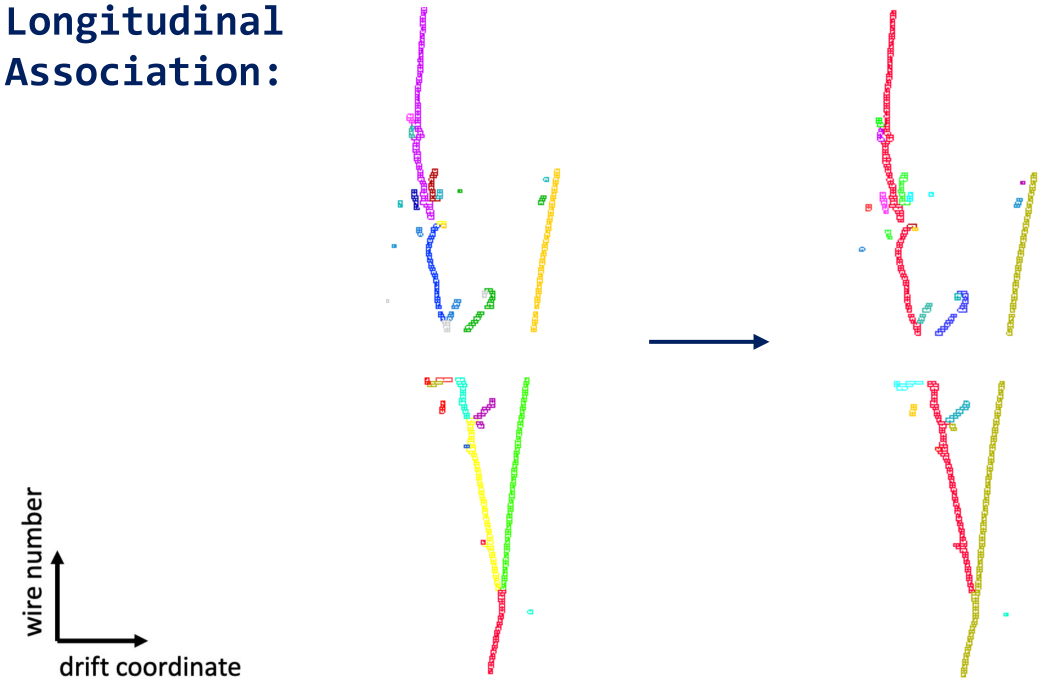
- In each view, a **cautious clustering algorithm** uses proximity-based logic to group hits into seed clusters
- 15 **cluster-merging** and **cluster-splitting** algorithms grow and refine the seed cluster based on topological information



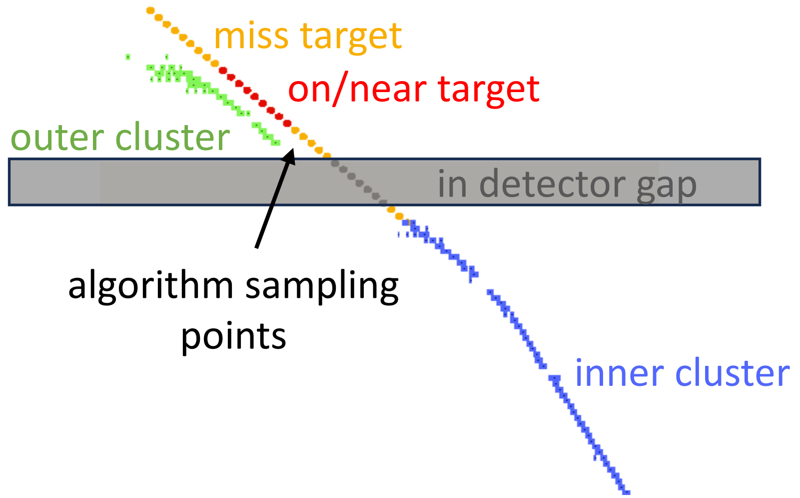
e.g. cluster merging algorithms

- Cluster-merging algorithms identify associations between multiple 2D clusters, growing them to **improve completeness, without compromising purity**
  - Cluster-merging decisions must consider cluster pairs in the **context of the event,** rather than in isolation
  - Each algorithm provides its own definition of **'is associated'** depending on its sought topology
  - With this, we navigate forwards and backwards to identify **chains of associated clusters** that can be safely merged

Longitudinal Association:



CrossGapsAssociation:



Lancaster University

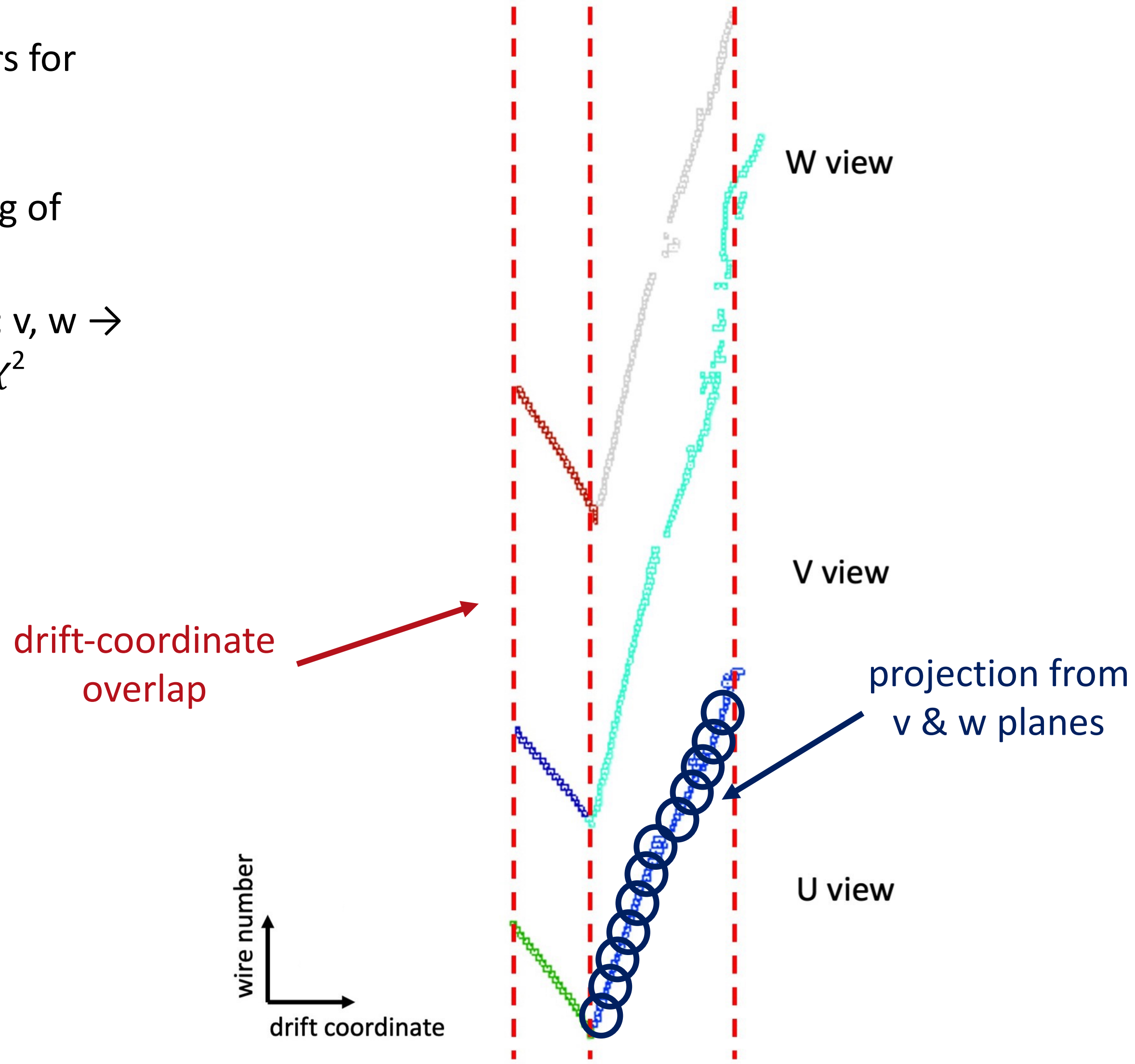
3D track reconstruction

- Assuming three input views (u, v, w), we can expect three 2D clusters for each particle

- We exploit the common drift-time coordinate and our understanding of wire plane geometry to identify them
  - At given drift-coordinate, compare predictions {u, v → w; v, w → u; w, u → v} with cluster positions, calculating a pseudo-χ^2

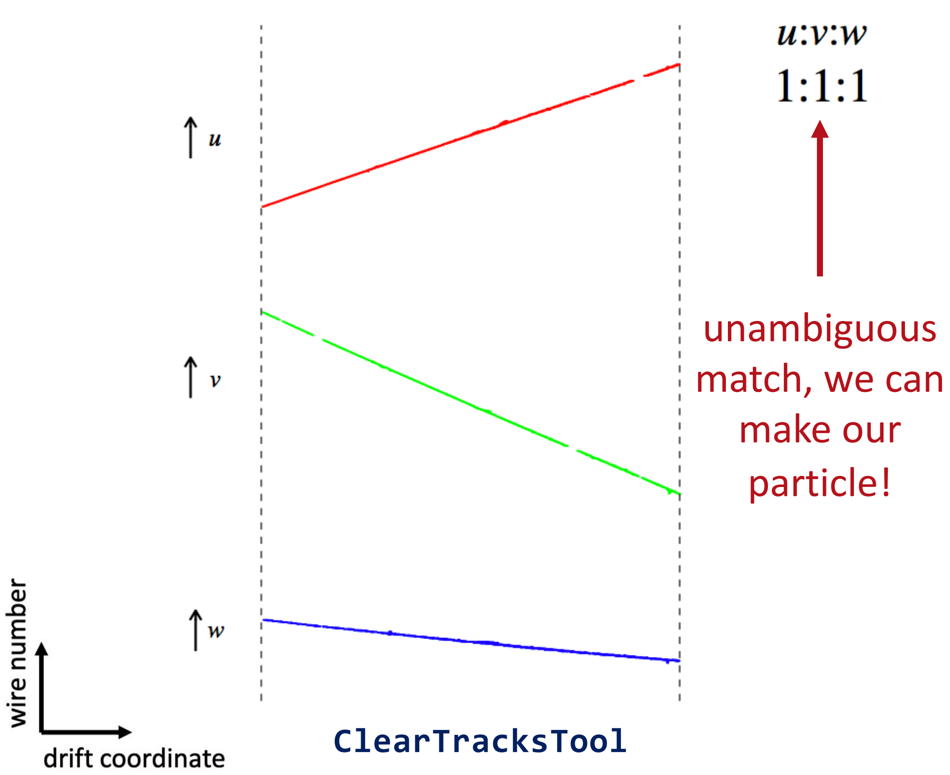
- All matches are stored in a **3x3 matrix** which records:
  - Drift overlap
  - no. sampling points
  - no. ‘matched’ sampling points
  - pseudo-χ^2

- **This matrix allows us to solve cluster-matching ambiguities**

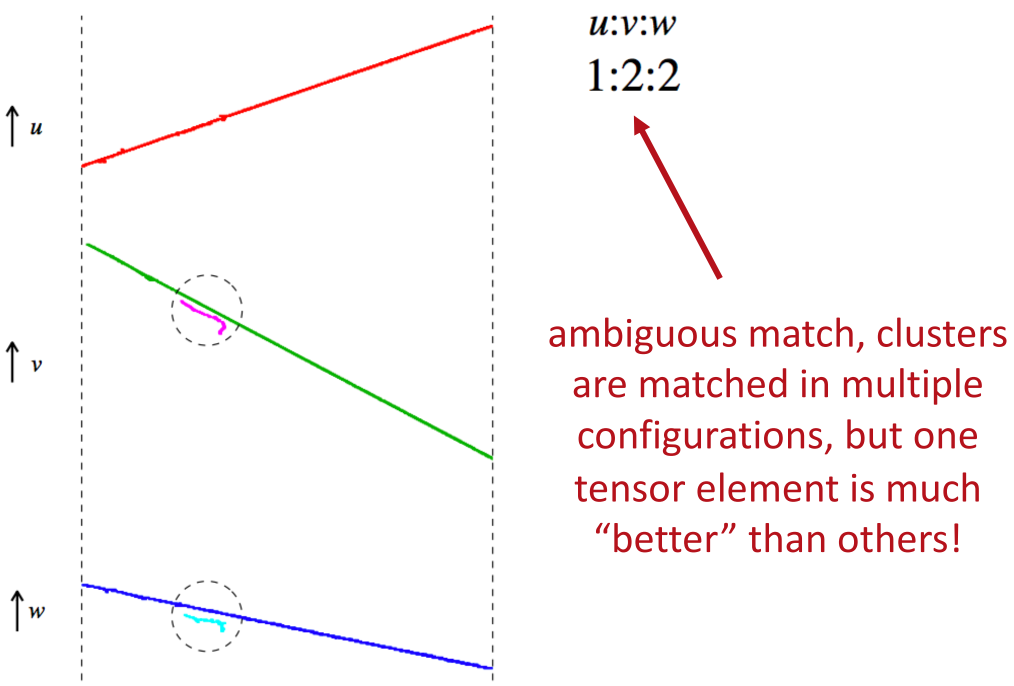


3D matching matrix

- Our **matrix** of matches allows us to construct our particle, and solve any cluster-matching ambiguities e.g. if we have one U cluster, one V cluster and two W clusters that 'match'
- Tools probe our **matrix** and, if appropriate, modify our clusters to **resolve any ambiguities**
- This approach maximises our achieved completeness and purity

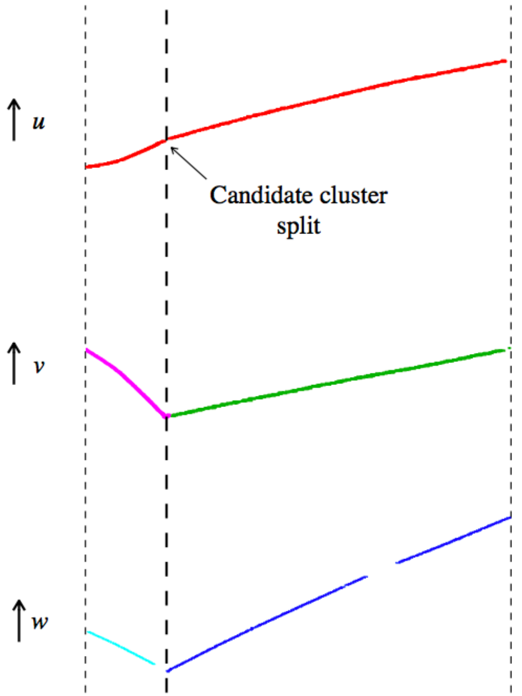


`ClearTracksTool`



`LongTracksTool`

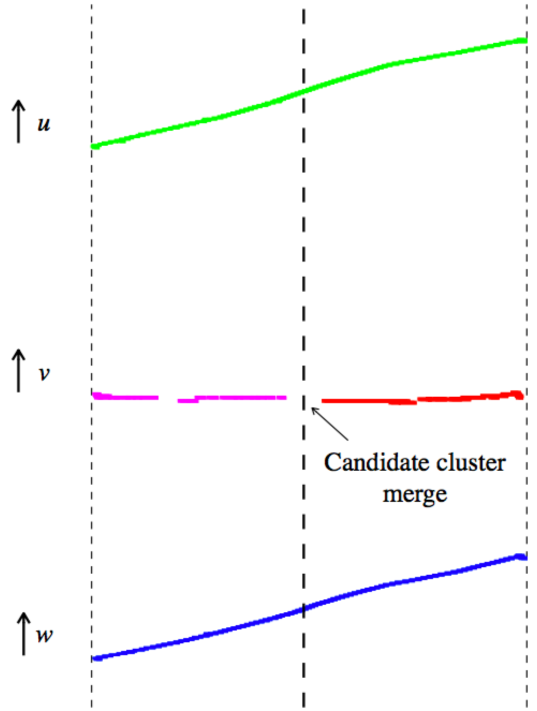
3D matching matrix



u:*v*:*w*
1:2:2

two clusters in w and v, matched to common u cluster
⇒ if true 3D kink topology, split ***u*** cluster

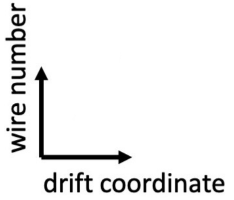
`OvershootTracksTool`



u:*v*:*w*
1:2:1

two clusters in v view, matched to common clusters in u and w views
⇒ merge ***v*** clusters

`UndershootTracksTool`

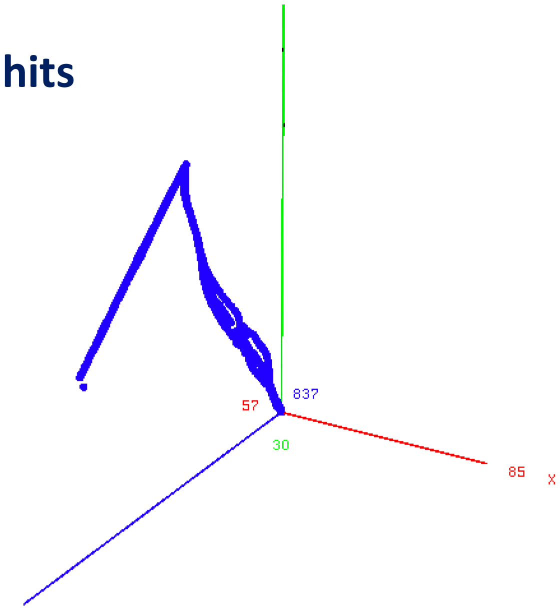


Following these changes, the **`ClearTracksTool`** tool identifies these unambiguous matches and form particles

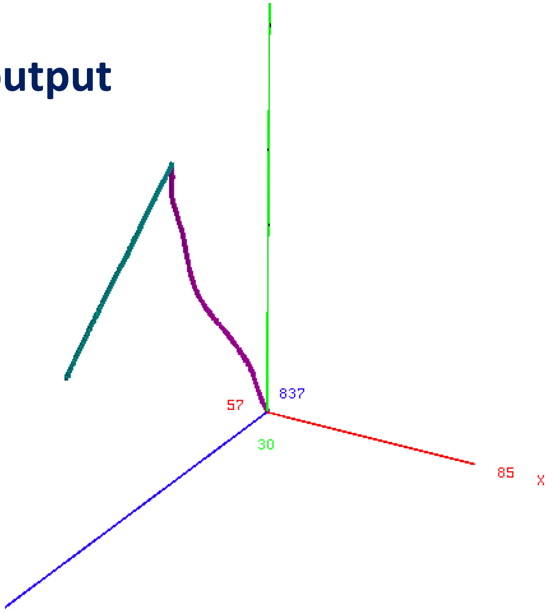
3D hit/cluster creation

- For each 2D hit, we sample the clusters in the other views at same drift-coordinate, obtaining u_{in}, v_{in} and w_{in}
- Each (u_{in}, v_{in}, w_{in}) triplet doesn't necessarily correspond to a specific point in 3D space
- An analytic expression is used to find the most consistent 3D space point
  - $\chi^2 = (u_{out} - u_{in})^2 / \sigma_u^2 + (v_{out} - v_{in})^2 / \sigma_v^2 + (w_{out} - w_{in})^2 / \sigma_w^2$
  - to solve: write in terms of, and differentiate wrt, the unknown y and z values
  - iterate, using a fit to current 3D hits (extra terms in χ^2), to produce a smooth trajectory

first pass 3D hits



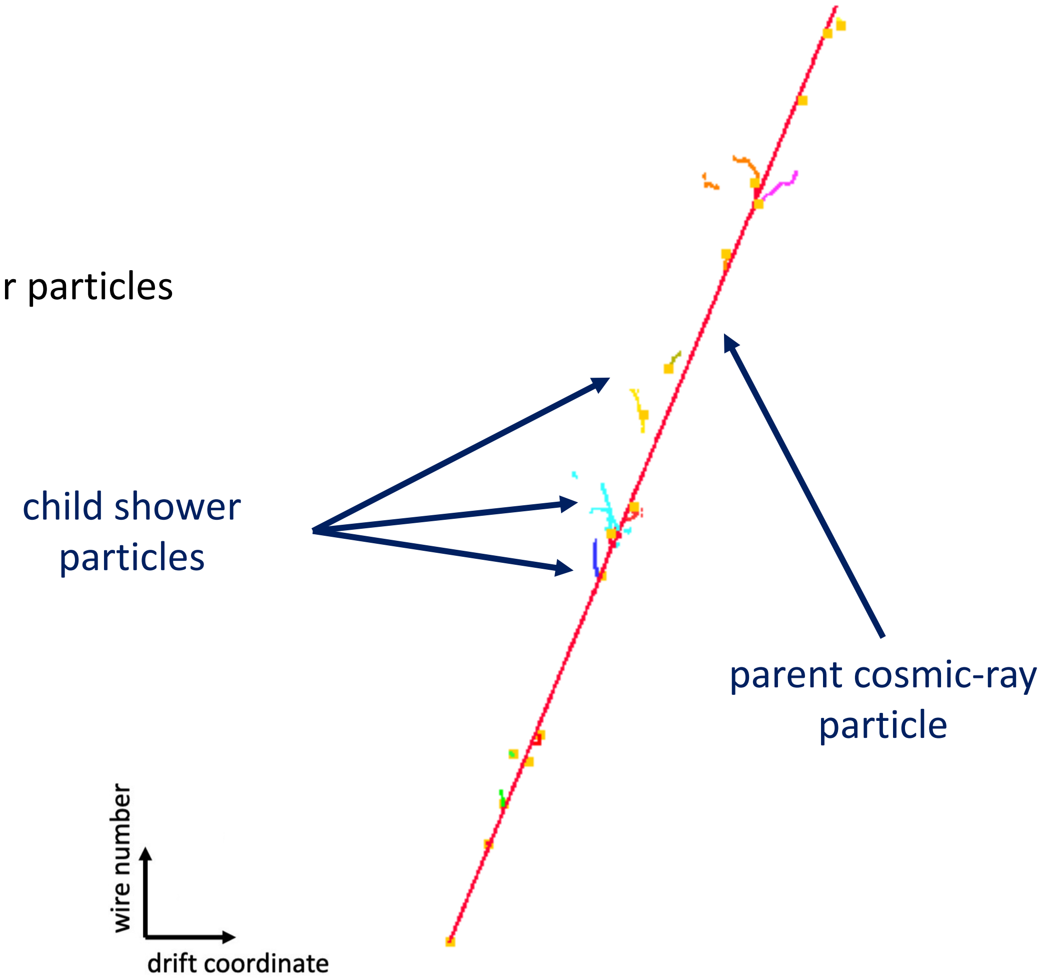
final 3D output



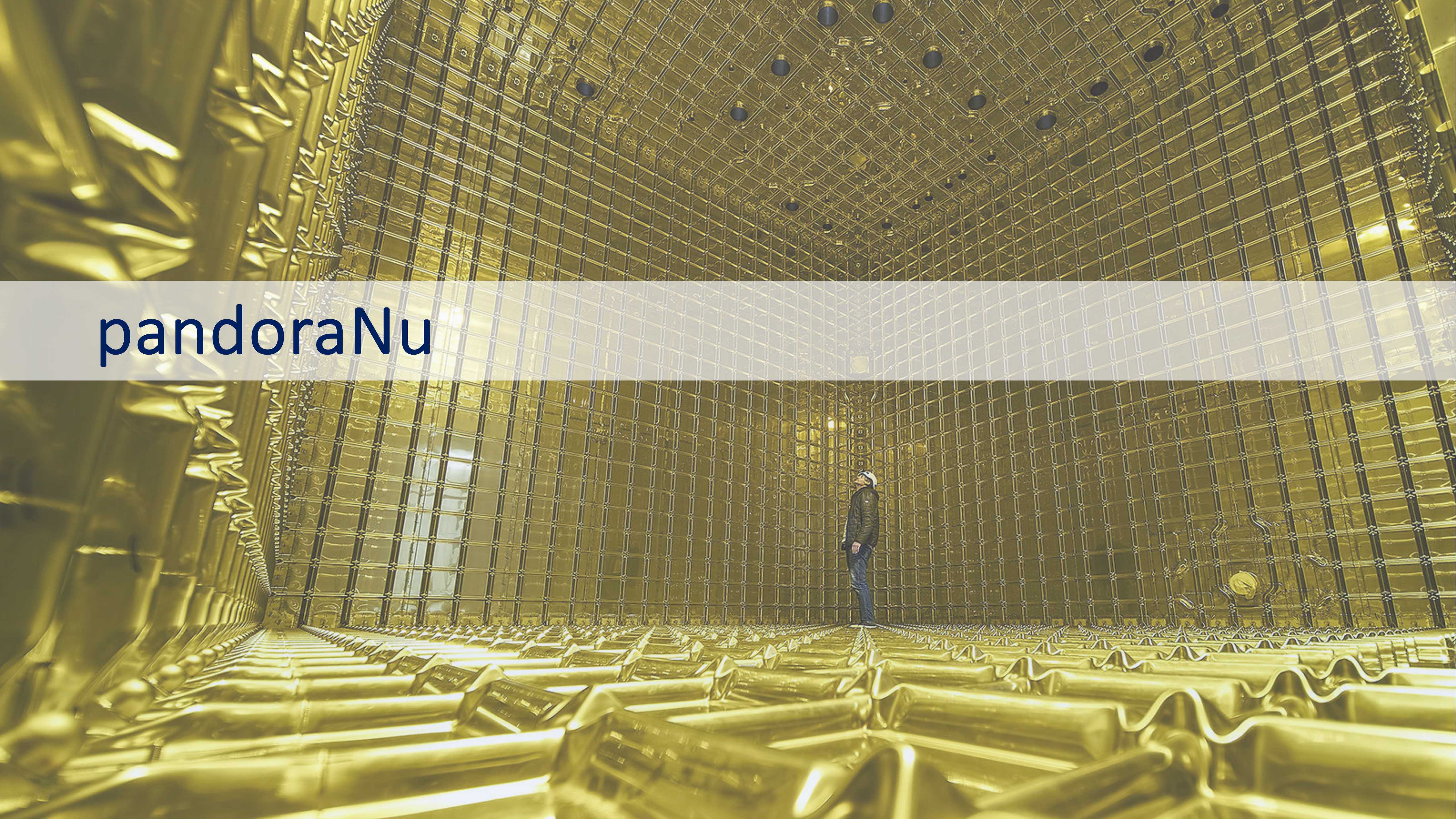
Delta-ray reconstruction

- At this point, we assume that we've found our cosmic-rays and that any unmatched belong to child delta-ray or michel electron showers

- We turn our focus to their reconstruction:
  - Simple proximity-based re-clustering of hits
  - Topological association cluster growing
  - 2D $\rightarrow$ 3D cluster matching, forming our child shower particles
  - 3D hit creation
  - Parent cosmic-ray $\rightarrow$ child shower links identified

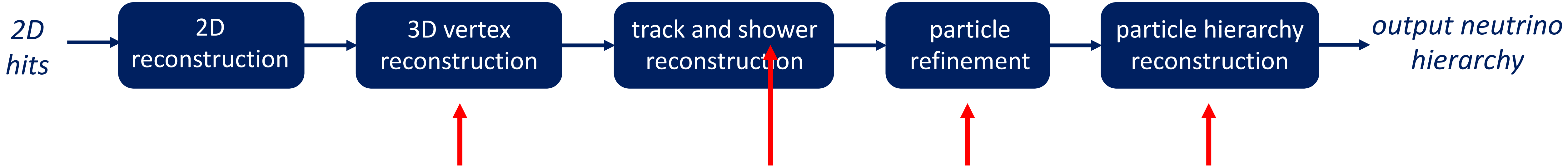


pandoraNu



pandoraNu

- **pandoraNu** uses a lot of the initial track-oriented clustering and topological association algorithms seen in **pandoraCosmic**
- There are some significant differences:
  - Topological association algorithms face topologies of **higher complexities**
  - Specific effort to reconstruct the neutrino interaction vertex
  - More sophisticated efforts to reconstruct showers



- Let's go through the new stages in more detail…

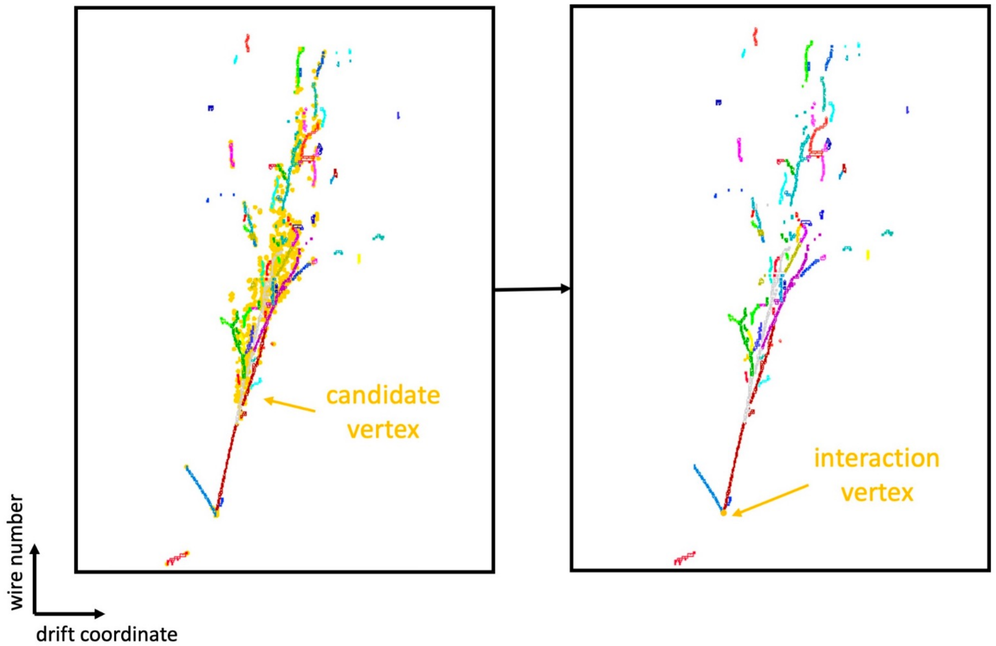
ATTN: In the following we're going to assume we're working with a detector with three readout planes, which is true for MicroBooNE, ProtoDUNE-SP, DUNE-HD, SBND, ICARUS and DUNE-VD

3D vertex reconstruction

- The identification of the interaction vertex guides the neutrino reconstruction chain
- Mis-placements can be detrimental, splitting particles and resulting in an incorrect neutrino hierarchy
- Across the LArTPC experiments, two approaches are used:

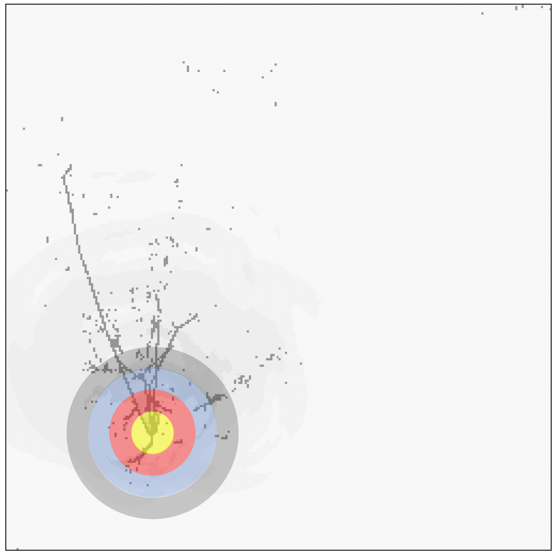
An older, approach:

- Candidate vertices are created at cluster endpoints
- A BDT/SVM is used to select the best candidate



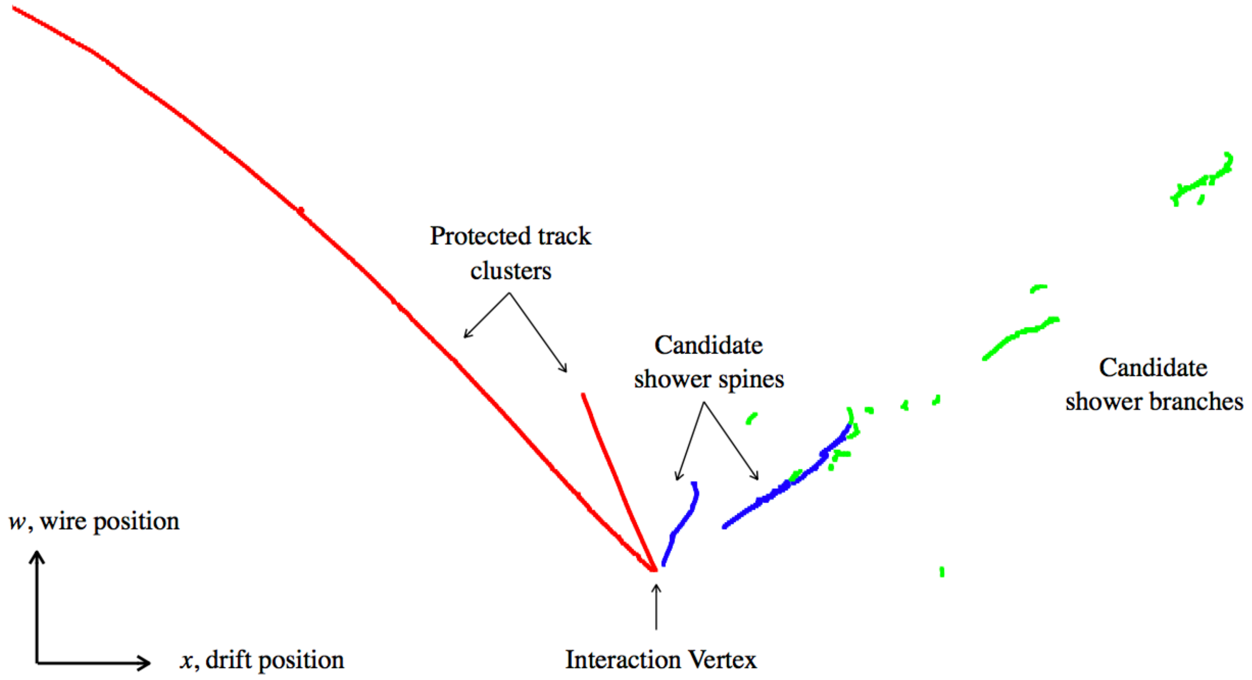
A new, approach:

- A 'deep network' predicts each hit's distance from the neutrino vertex
- A heatmap is used to identify the vertex position



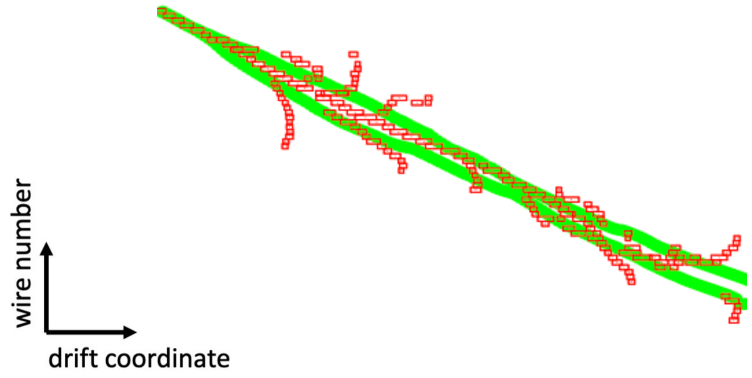
2D shower reconstruction

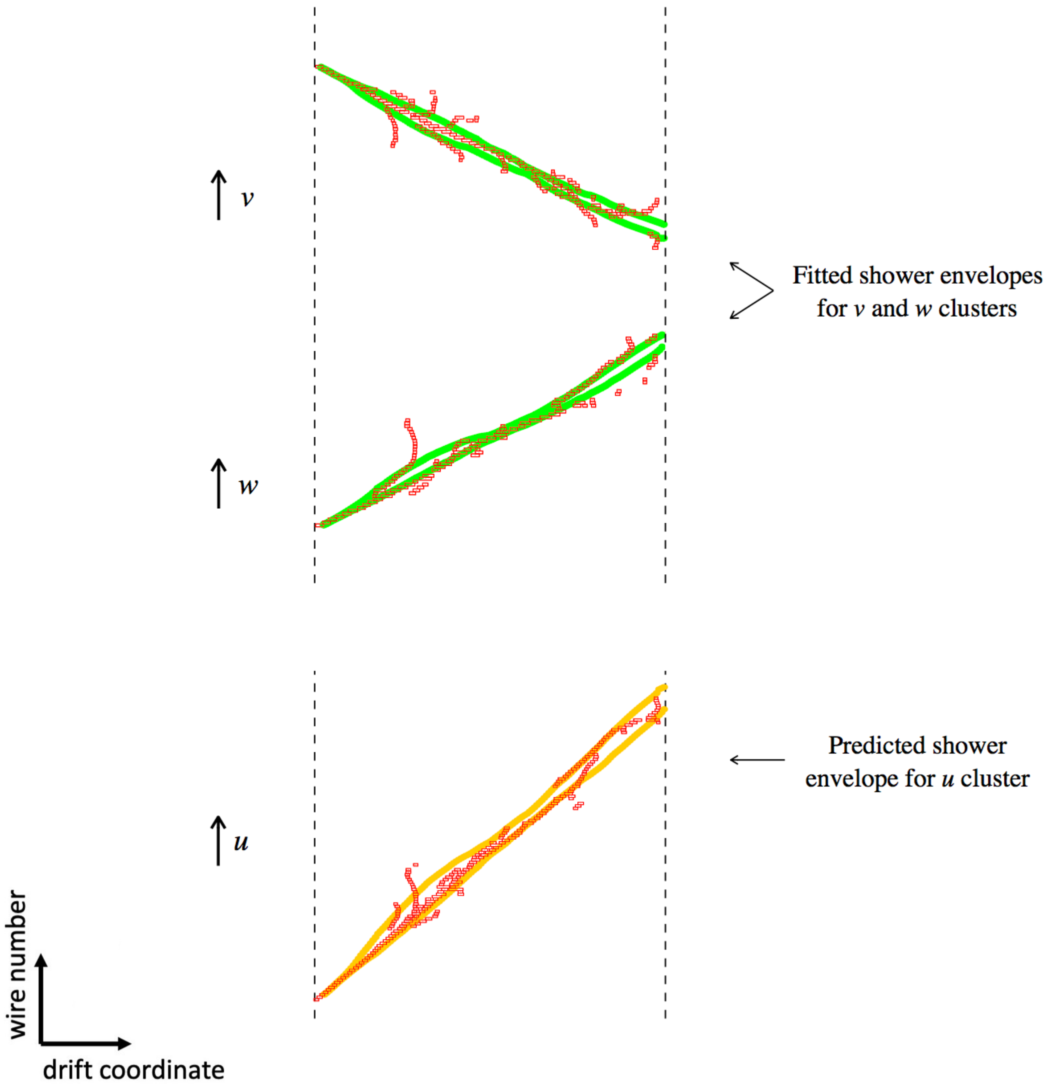
- As previously mentioned, the track reconstruction proceeds as in **pandoraCosmic**
- For the remaining showers, we start by forming 2D clusters:
  - Use topological properties to identify clusters which represent 'shower spines'
  - Grow showers by iteratively merging shower branches to identified shower spines



3D shower reconstruction

- To create our shower particles, we use a simpler version of our 2D → 3D cluster matching procedure:
  - Build a **matrix** to store cluster overlap and relationship information
  - Showers are parameterised by fitted 2D envelopes
  - The envelope edges of two views are used to predict those in a third view, forming the basis of a matching metric
  - Particles creation is guided by this metric

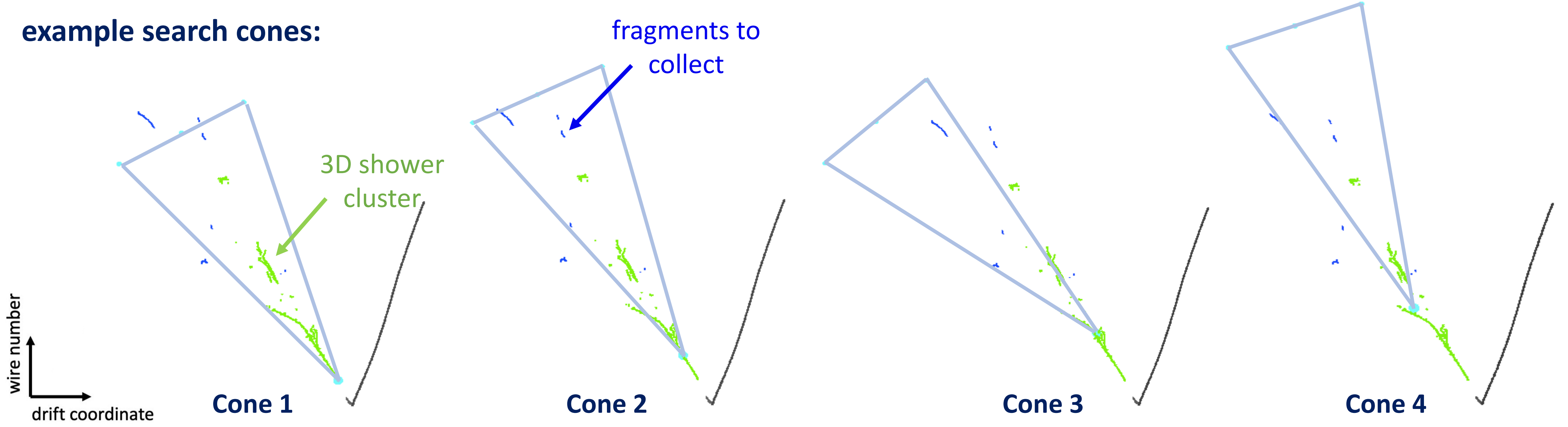




Particle Refinement – 2D, 3D

- Series of ‘refinement’ algorithms focus on the ‘leftovers’, maximising our particle completeness (esp. sparse showers):
  - Pick up small, unassociated clusters bounded by the 2D envelopes of 2D particles
  - Use 3D fits of our showers to define cones in which to search for remnants
  - After this, dissolve any remaining clusters and assign hits to nearest showers within a given range

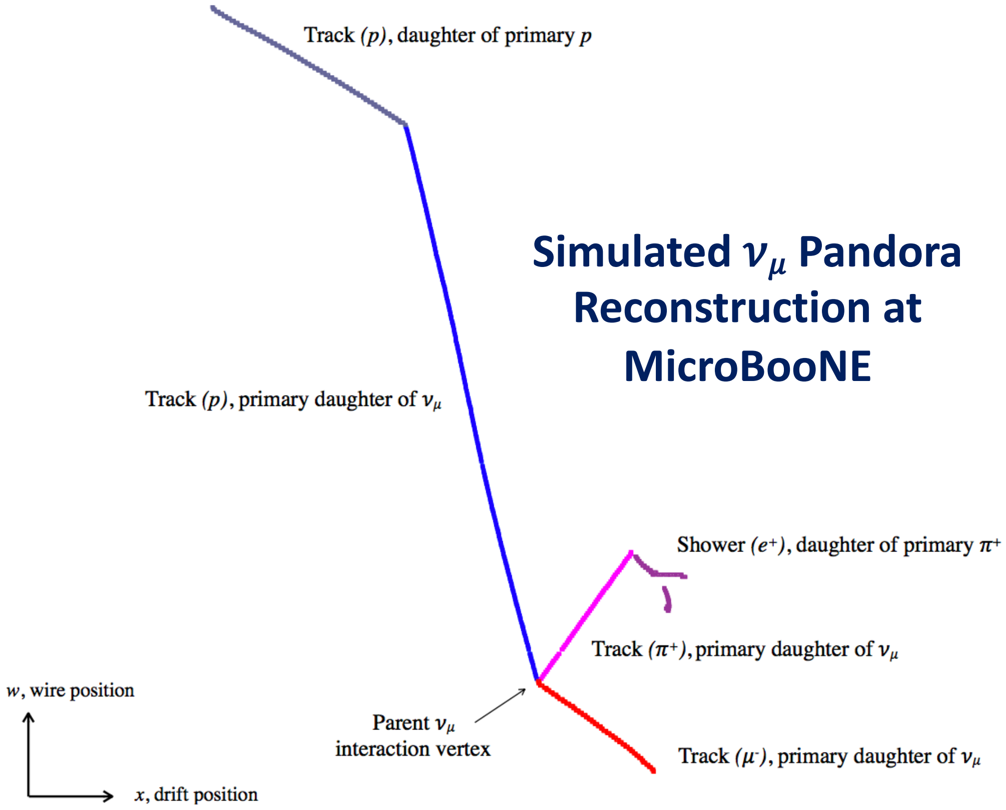
example search cones:



Hierarchy building

- Working out from the neutrino vertex, topological information is used to construct our neutrino hierarchy
- **This is the output which is written into .root files, using the LArSoft format, and given to our analysers!**

- **Each reconstructed particle has:**
  - A vertex
  - An identified parent and children (may be none)
  - A track/shower score
  - Associated 2D clusters and 2D hits
  - Associated 3D cluster and 3D SpacePoints



Simulated ν_μ Pandora Reconstruction at MicroBooNE

Conclusions

- **Pandora** is an event reconstruction software, which utilises a **multi-algorithm approach to pattern recognition**
- Pandora is employed by **all modern LArTPC experiments**
- **Isolated algorithms** allows for the development of reconstruction chains to meet the challenges of different detectors, and analyses
- Our main reconstruction chains focus on the reconstruction of cosmic-rays, test-beam interactions and neutrinos, and have been detailed in this talk
- Despite the 'traditional' foundations of Pandora, the use of **deep-machine learning algorithms** has shown to be affective and the number of ML algorithms is growing (see Andy C's talk)