Inside PandoraNu (per slice): multi-view 2D → 3D pattern recognition

TPC hits
(cosmic-removed)

↓

PandoraNu slicing
(partition hits into candidate interactions)

↓

Slice CaloHits
(all planes: U, V, Y)

U view (induction)

U CaloHits

↓

2D clustering U
(seed/grow/split/merge)

↓

2D clusters U

V view (induction)

V CaloHits

↓

2D clustering V
(seed/grow/split/merge)

↓

2D clusters V

Y view (collection)

Y CaloHits

↓

2D clustering Y
(seed/grow/split/merge)

↓

2D clusters Y

↓

Cross-view association
(match 2D clusters across planes)
using geometry + drift-time consistency

Principle: each view gives a 2D projection.
3D comes from combining:
• shared drift coordinate (x from time)
• wire geometry constraints
• topological consistency

↓

3D clusters / prongs
(spacepoints, skeletons)

↓

Topology refinement (iterative)
• resolve ambiguities
• missing-view recovery
• fragment removal
• prong splitting/merging

↓

Track vs shower building
• track-like: linear, MIP-like
• shower-like: branching, increasing width

Discriminants (typical):
• track: straightness, ~constant dQ/dx
• shower: start dE/dx, spread, branching

↓

Vertex hypothesis
(connect prongs to common origin)

↓

PFO hierarchy assembly
(parent-daughter, interaction tree)

↓

Candidate flashes
(time, PE pattern)

Slice-level PFOs
+ associations & scores

↓

Flash matching / t0 assignment
(match slice ↔ in-beam flash)

Purpose: validate the charge-only slice hypothesis.
Provides an absolute **t0** and a powerful cosmic-rejection handle.

↓

Selected neutrino slice
+ t0 + cosmic-rejection score