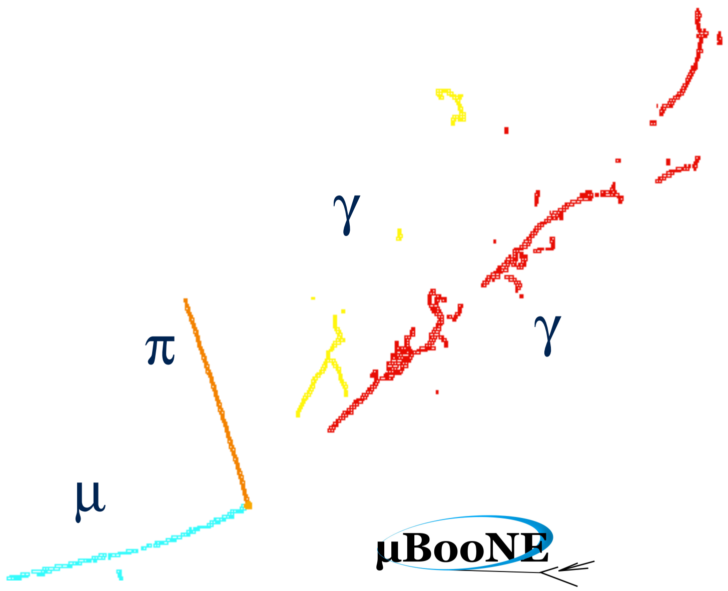


Pandora LAr TPC Reconstruction

J. de Vries for The Pandora Team

Proto-DUNE Workshop

28 June 2016

Pandora Multi-Algorithm Approach

- Pandora provides a multi-algorithm approach to LAr TPC pattern recognition:
  - Uses a large numbers of algorithms (80+) to examine hits and identify particles.
  - Each algorithm carefully developed to address specific topology
  - Some algs very sophisticated, others rather simple: gradually build-up picture of events.

- Multi-algorithm approach made possible using functionality provided by Pandora SDK:
  - Algs provide all logic, but must use APIs for access to, or to modify, hits/clusters/particles.
  - Advanced functionality enables complex algorithms using recursion or reclustering.

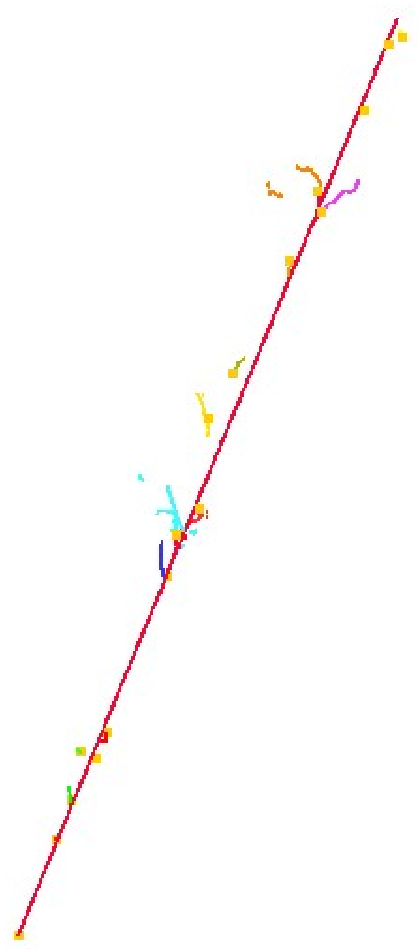
- Intense development during the past year, prioritising MicroBooNE:
  - Continually improving communication with analysis/reco groups, plus LArSoft.
  - All developments are reusable for DUNE and proto-DUNE
  - Highly adaptable framework

Further algorithm details here
Today: an overview...

Pandora Cosmic vs Nu

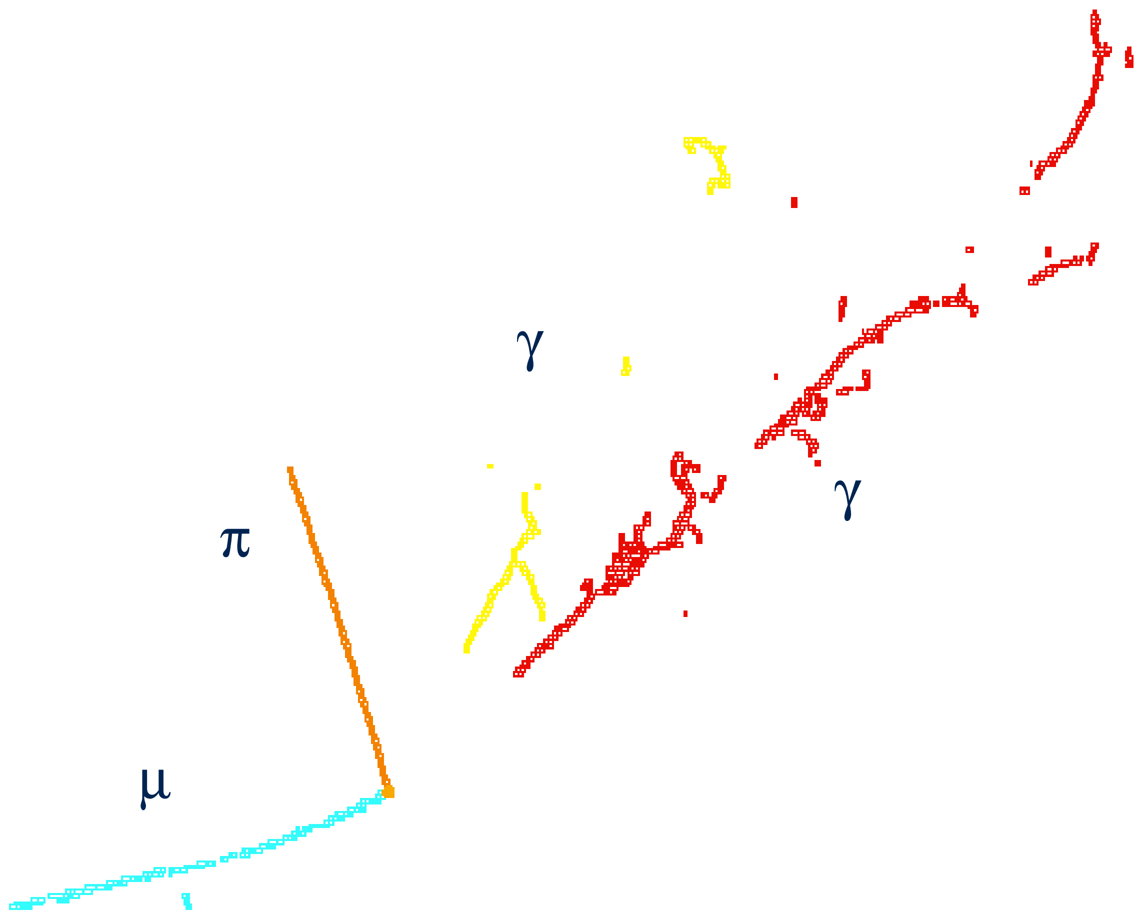
Illustration of the potential of the multi-algorithm approach. Two very different "passes" can be done based on Pandora algorithms (selected through settings .xml files)

Cosmic pass



Optimised for cosmic rays reconstruction

Neutrino pass



Optimised for neutrino reconstruction

Pandora Cosmic vs Nu

Illustration of the potential of the multi-algorithm approach. Two very different "passes" can be done based on Pandora algorithms (selected through settings .xml files)

Cosmic pass

- Strongly track-oriented
- Top-level particles representing cosmic ray muons
- Showers are assumed to be delta rays - daughters of cosmic ray muons
- Cosmic removal highly relevant for protoDUNE

Optimised for cosmic rays reconstruction

Neutrino pass

- Identify neutrino interaction vertex
- Particles emerging from the vertex are reconstructed as individual primary particles
- Daughters of the neutrino can have own daughters
- Careful treatment shower/track reconstruction

Optimised for neutrino reconstruction

Pandora Algorithms Overview

More than 80 algorithms, years of development in Pandora

1) **Track clustering in 2D**

2) **Vertex reconstruction in 3D**

3) **Track reconstruction in 3D**

4) **Shower reconstruction in 3D**

5) **Mop-up in 2D and 3D**

6) **Event building in 3D**

Each step encloses different algos

2D Reco Snippet from PandoraSettings XML file:

```
<algorithm type = "LArClusteringParent">
    <algorithm type = "LArTrackClusterCreation" description = "ClusterFormation"/>
    <InputCaloHitListName>CaloHitListW</InputCaloHitListName>
    <ClusterListName>ClustersW</ClusterListName>
    <ReplaceCurrentCaloHitList>false</ReplaceCurrentCaloHitList>
    <ReplaceCurrentClusterList>true</ReplaceCurrentClusterList>
</algorithm>

<algorithm type = "LArLayerSplitting"/>
<algorithm type = "LArLongitudinalAssociation"/>
<algorithm type = "LArTransverseAssociation"/>
<algorithm type = "LArLongitudinalExtension"/>
<algorithm type = "LArTransverseExtension"/>
<algorithm type = "LArOvershootSplitting"/>
<algorithm type = "LArBranchSplitting"/>
<algorithm type = "LArKinkSplitting"/>
```

2D Cluster Creation

2D Cluster Merging/Splitting

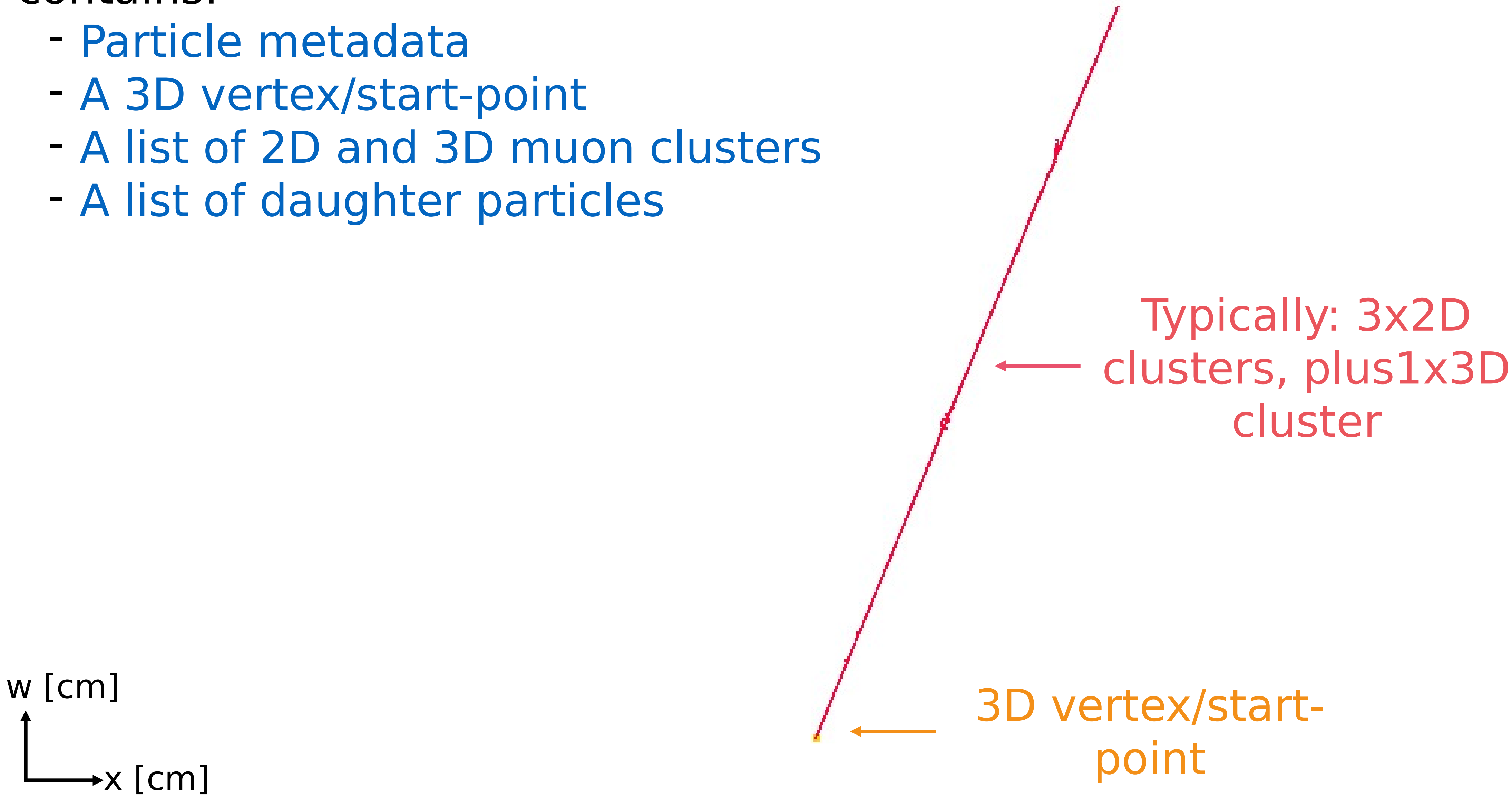
All of the 2D reconstruction framework can be applied to the dual phase detector at ProtoDUNE.

Not enough time in this talk, but please find more details here: http://goo.gl/NIxBj7
MicroBooNE technical note available soon

Cosmic Ray Muon: Display 1/2

Cosmic pass

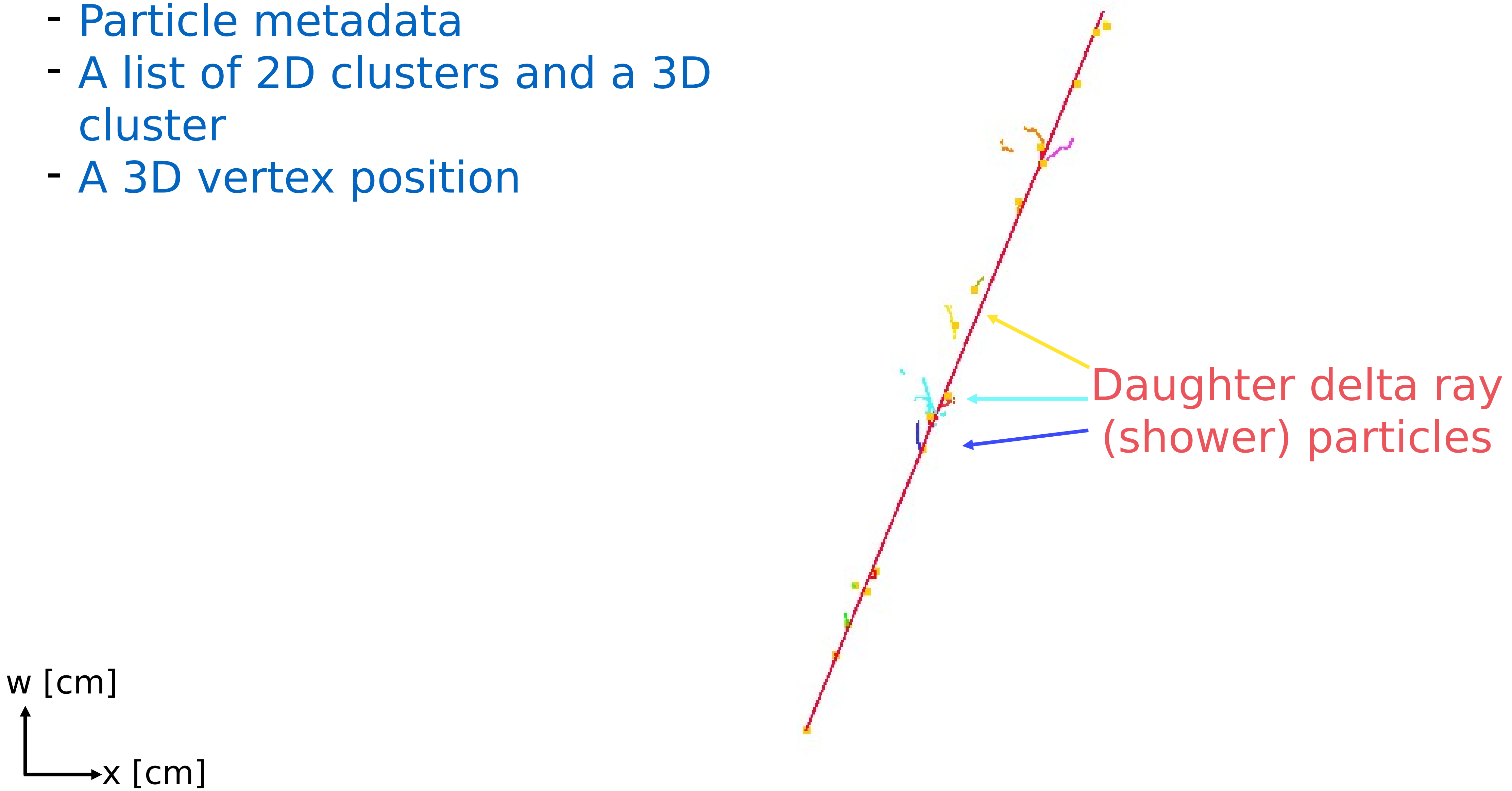
The reconstructed cosmic ray contains:
- Particle metadata
- A 3D vertex/start-point
- A list of 2D and 3D muon clusters
- A list of daughter particles



Cosmic Ray Muon: Display 2/2

Cosmic pass

- Daughter delta-rays, each of which has:
  - Particle metadata
  - A list of 2D clusters and a 3D cluster
  - A 3D vertex position



Pandora Example Events

Pandora algorithms and tools create 3D SpacePoints for each reconstructed particle. The full particle hierarchy is also reconstructed, so a typical event output is as shown below:

5 GeV νe CC: Display 1/4

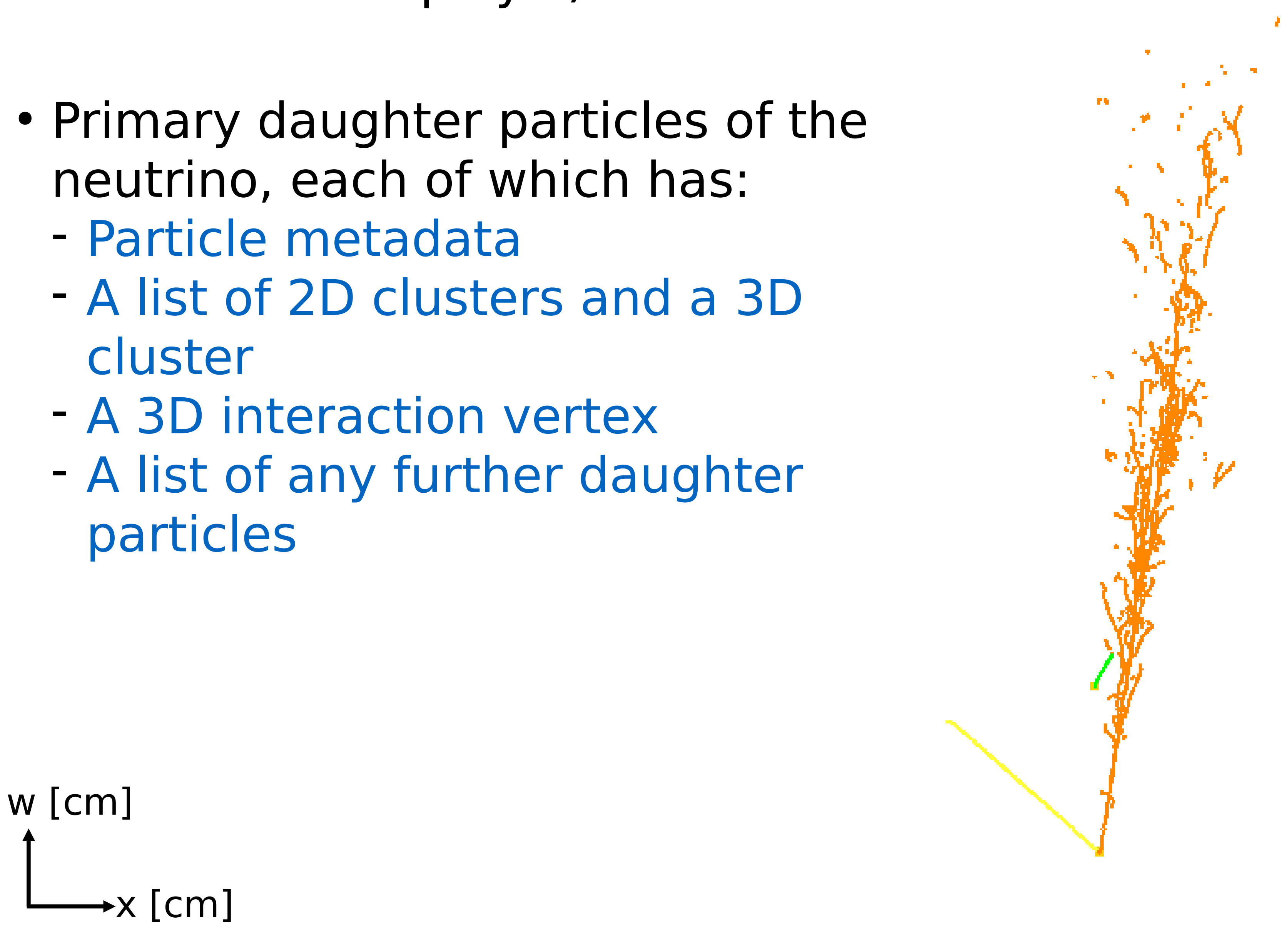
Neutrino pass

- The reconstructed neutrino particle contains:
  - Metadata: PDG code, 4-momentum, etc
  - A 3D interaction vertex
  - A list of daughter particles

Pandora algorithms and tools create 3D SpacePoints for each reconstructed particle. The full particle hierarchy is also reconstructed, so a typical event output is as shown below:

5 GeV νe CC: Display 2/4

- Primary daughter particles of the neutrino, each of which has:
  - Particle metadata
  - A list of 2D clusters and a 3D cluster
  - A 3D interaction vertex
  - A list of any further daughter particles

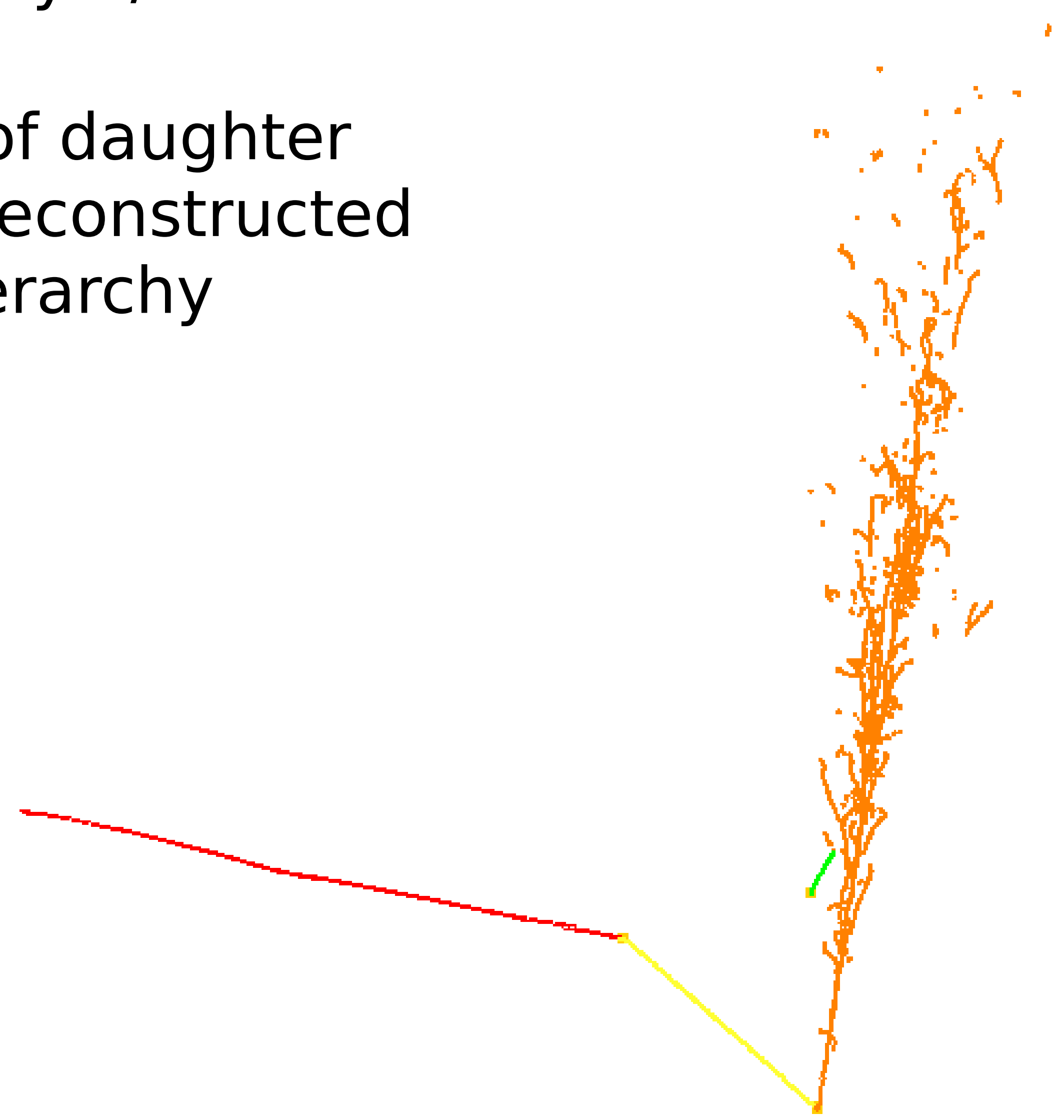


Pandora algorithms and tools create 3D SpacePoints for each reconstructed particle. The full particle hierarchy is also reconstructed, so a typical event output is as shown below:

5 GeV νe CC: Display 3/4

- Complete list of daughter particles in the reconstructed particle hierarchy

Neutrino pass



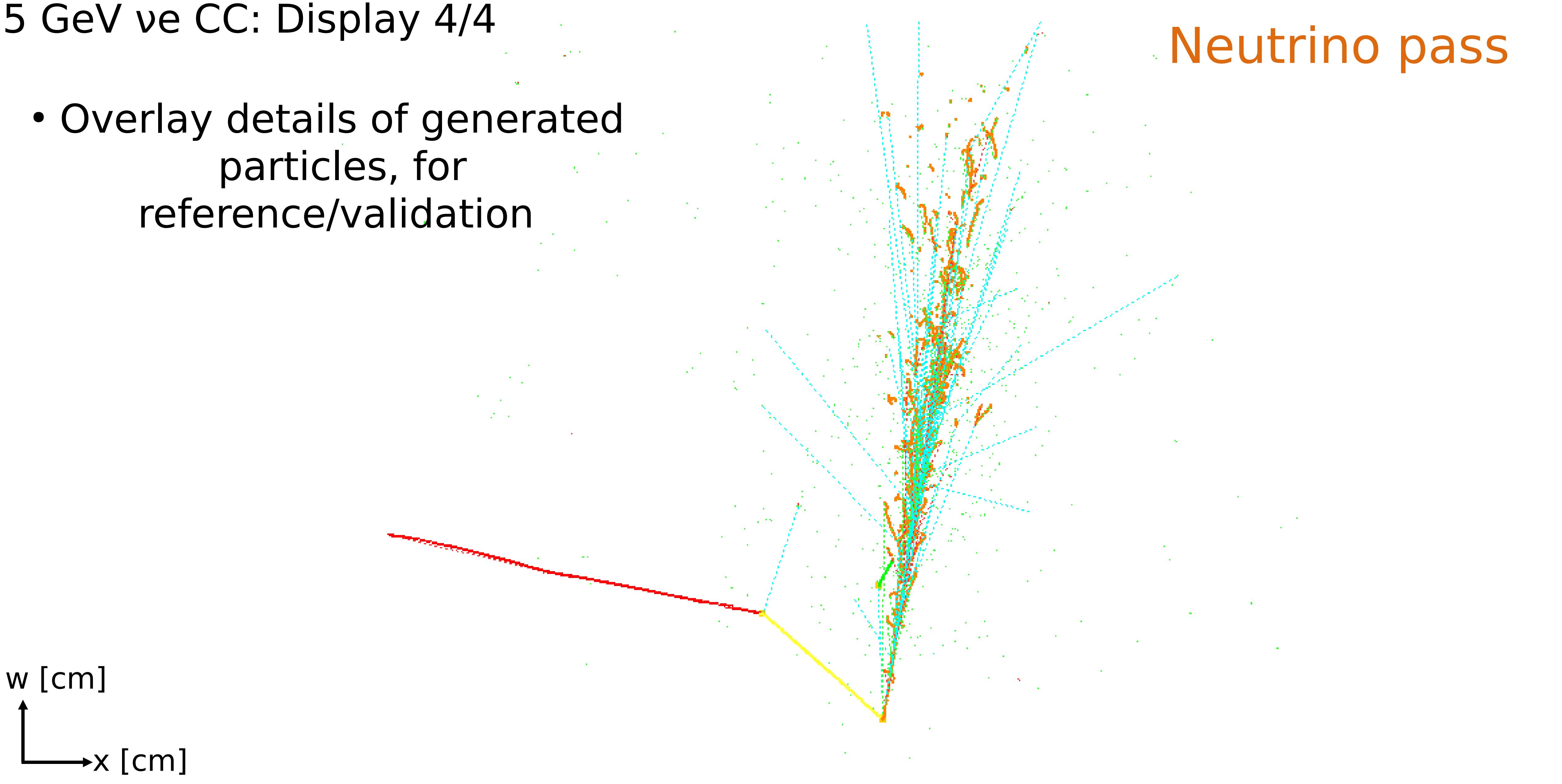
w [cm]

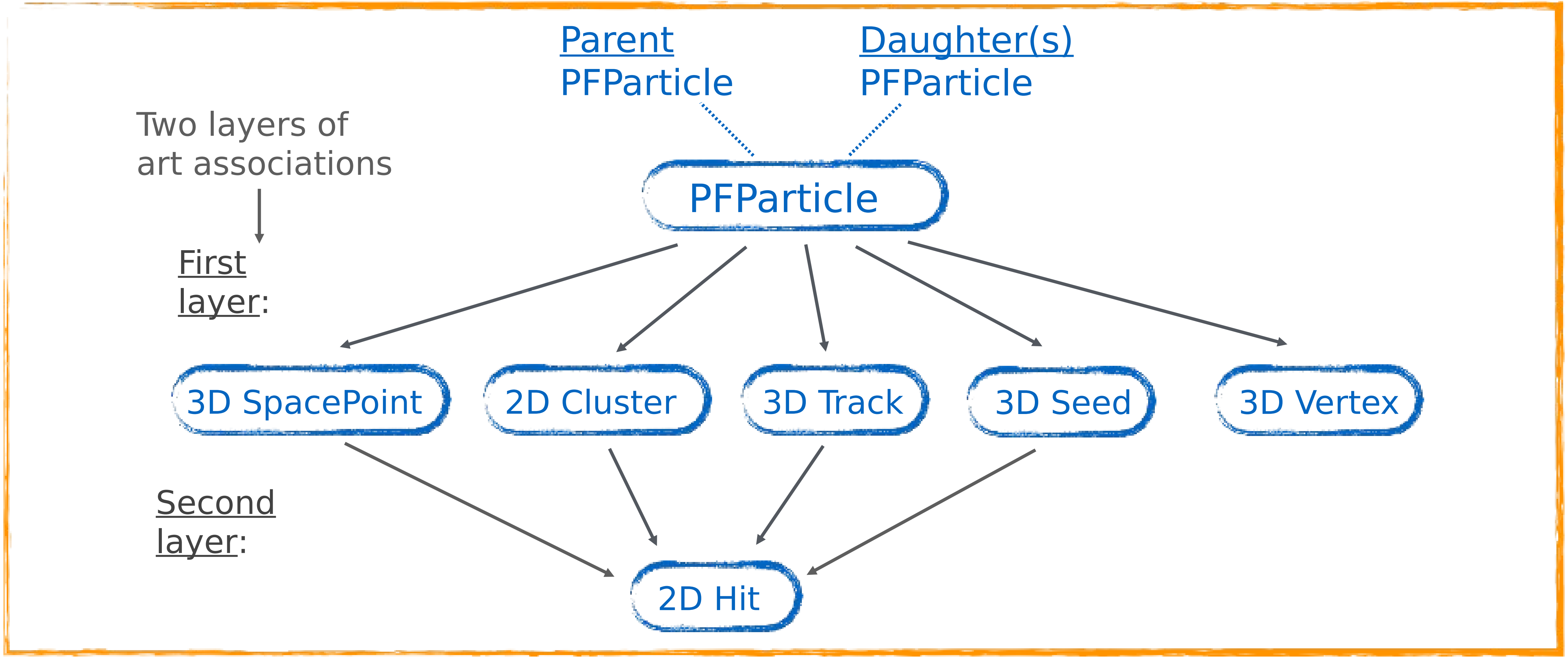
x [cm]

Pandora algorithms and tools create 3D SpacePoints for each reconstructed particle. The full particle hierarchy is also reconstructed, so a typical event output is as shown below:

5 GeV νe CC: Display 4/4

- Overlay details of generated particles, for reference/validation





- Note distinction between Found Tracks and Found Showers provided by Pandora and any downstream Fitted Tracks or "Value-added" Showers (with calorimetry information).
- LArSoft output must be handled carefully: use PFParticle functionality to navigate along particle hierarchies, then must always use art associations to navigate to related objects.

- Example usage: **`larpandora/LArPandoraInterface/LArPandoraHelper`**

Pandora Performance Metrics

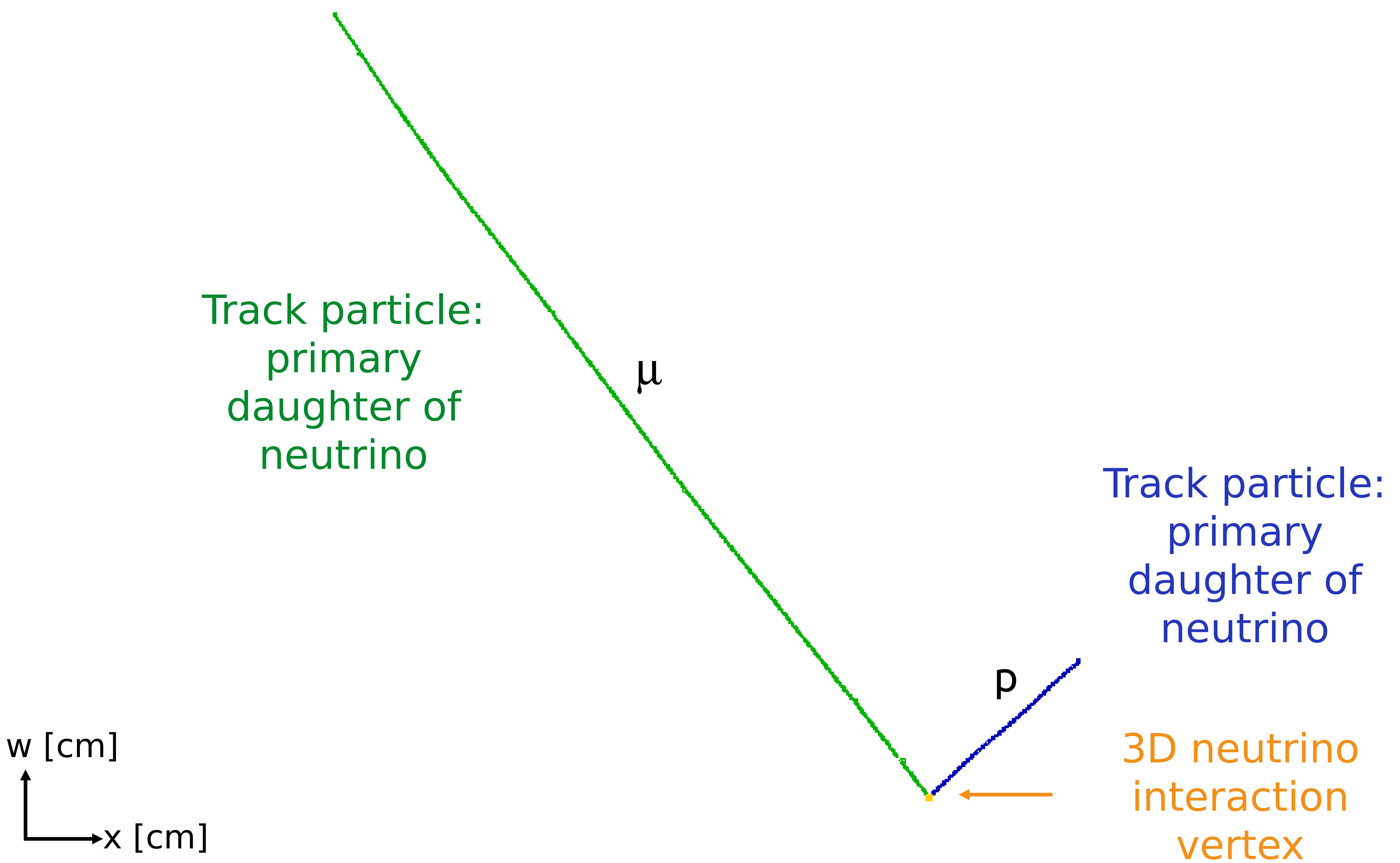
- Use **performance metrics** to assess reconstruction output and drive development:
  - Look at specific types of neutrino interactions in simulation.
  - Carefully match reconstructed particles to each true (primary) particle.
  - Count reconstructed particles for each true particle and assess quality of matches.

- A well-defined approach (see backup for full details), but not very forgiving: events with minor errors, readily dismissed by eye, often classed as failures.

- Striving for perfection - look to match precisely one reco particle to each true particle.

Next pages illustrate performance for MicroBooNE with the most current LArSoft release (LArSoft v05_04_00 - v05_08_00)

MicroBooNE simulation: BNB νμ CC QEL



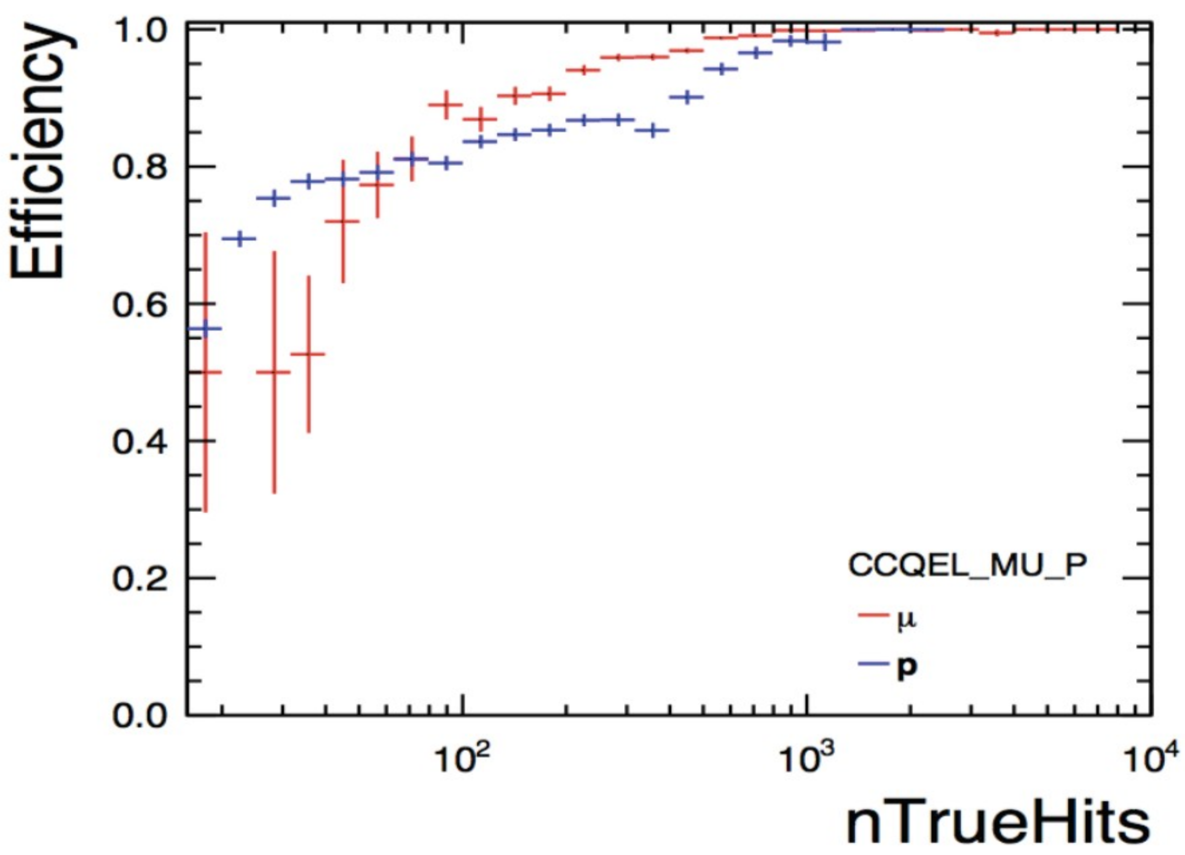
MicroBooNE simulation: BNB νμ CC QEL μ, p

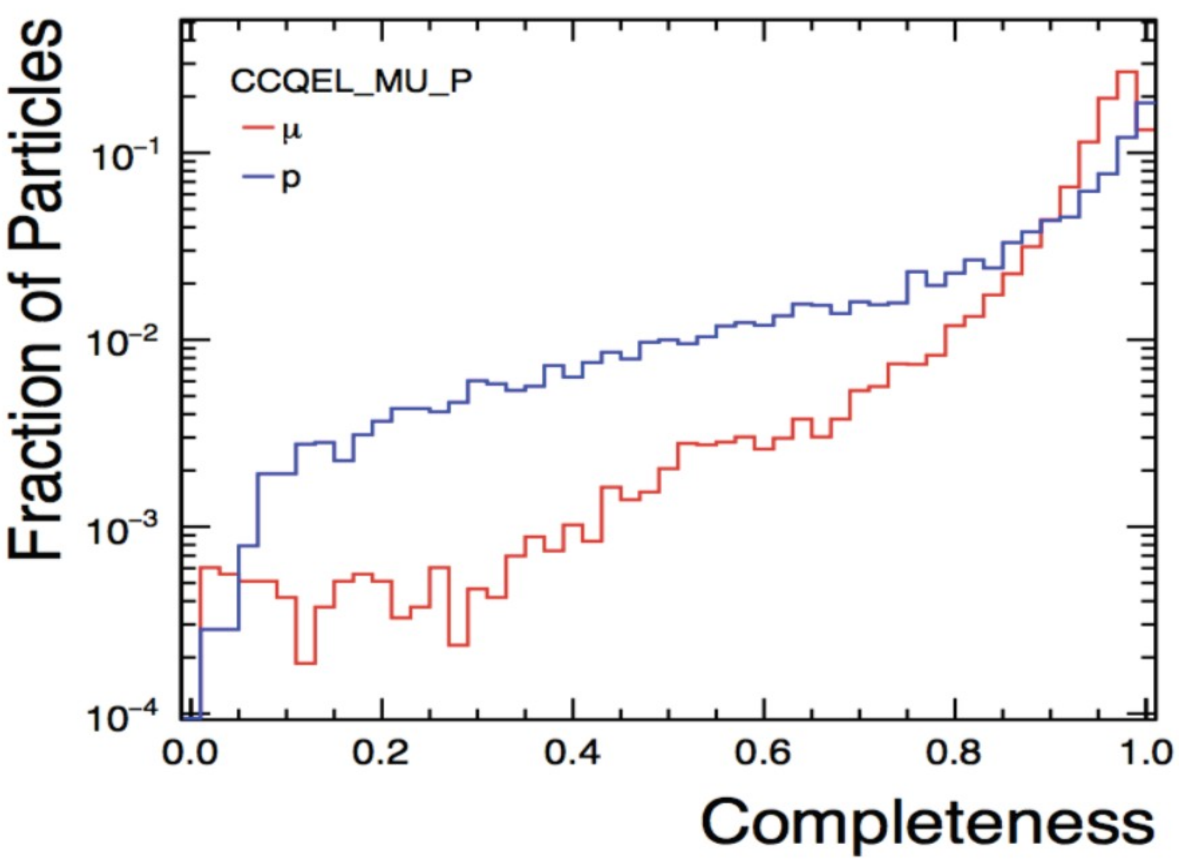
Input: LArSoft v05_08_00

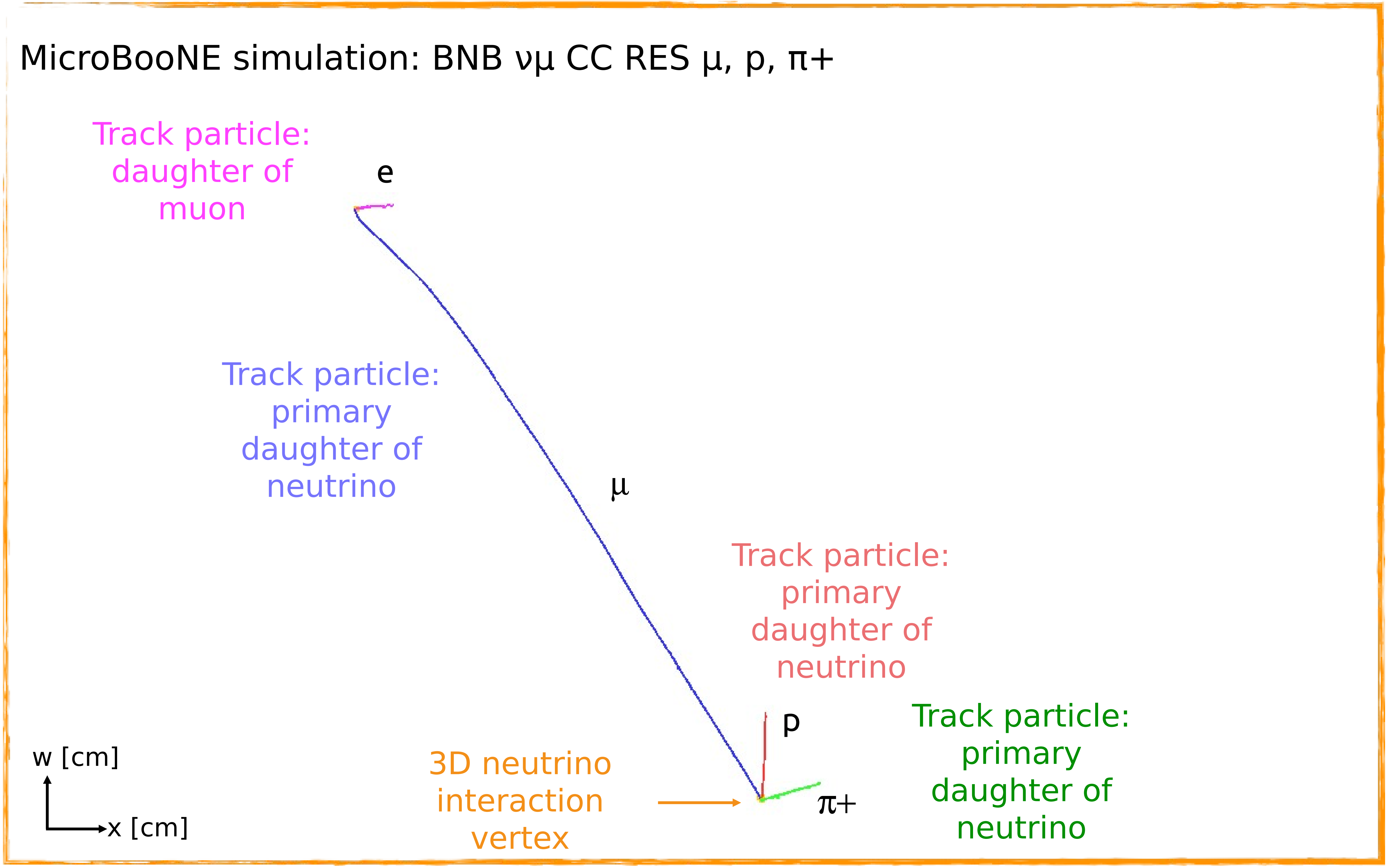
#MatchedPFOs	*0*	*1*	*2*	*3+*
μ	*2,7 %*	*90,7 %*	*6,1 %*	*0,5 %*
p	*19,8 %*	*76,4 %*	*3,5 %*	*0,3 %*

#Events: 22,102
#Perfect: 71.1%

Results for Neutrino 2016






MicroBooNE simulation: BNB νμ CC RES μ, p, π+
Track particle: daughter of muon
e
Track particle: primary daughter of neutrino
μ
Track particle: primary daughter of neutrino
p
Track particle: primary daughter of neutrino
π+
3D neutrino interaction vertex
w [cm]
x [cm]

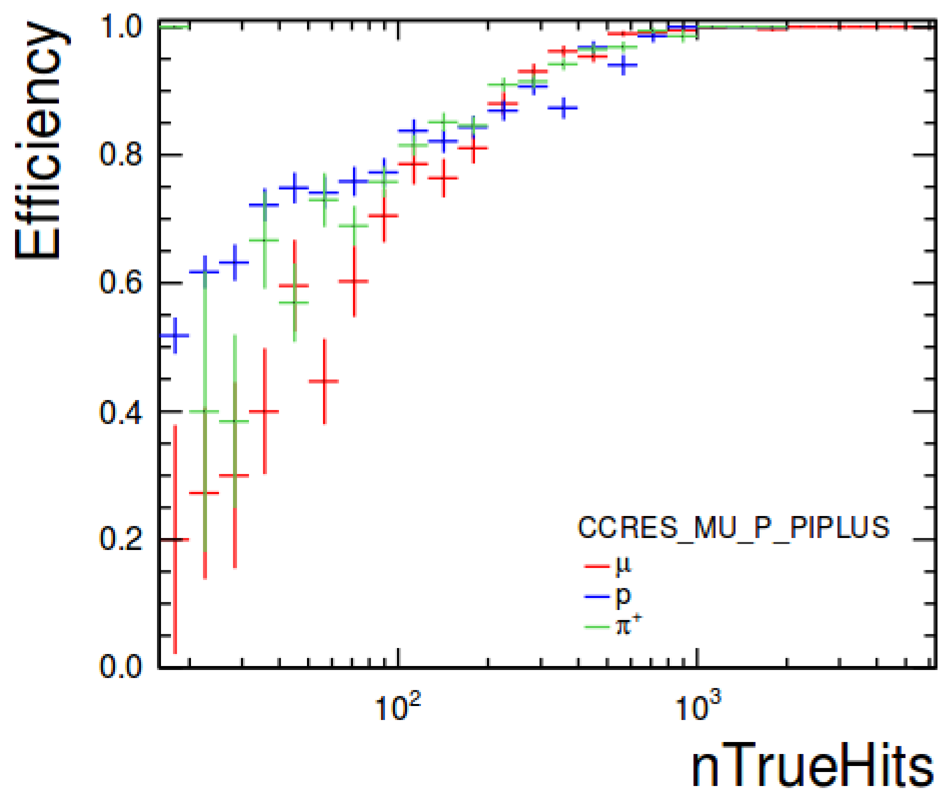
MicroBooNE simulation: BNB νμ CC RES μ, p, π+

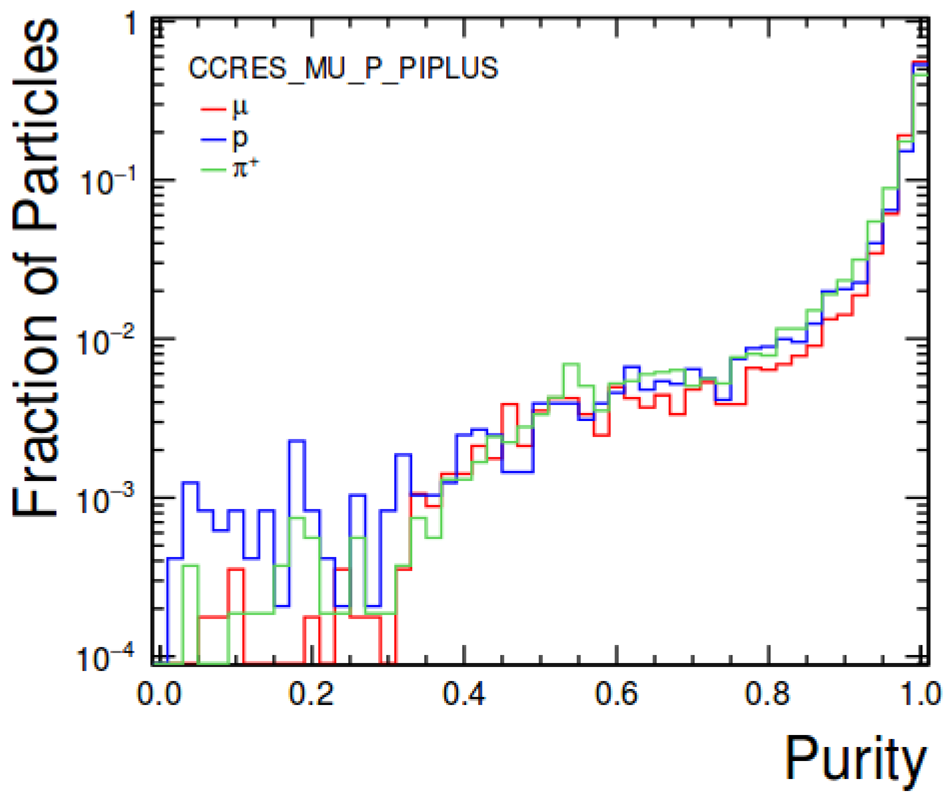
Input: LArSoft v05_08_00

#MatchedPFOs	0	1	2	3+
μ	6.8 %	87.7 %	5.1 %	0.4 %
p	20.5 %	75.0 %	4.0 %	0.5 %
π+	11.6 %	71.3 %	13.0 %	4.1 %

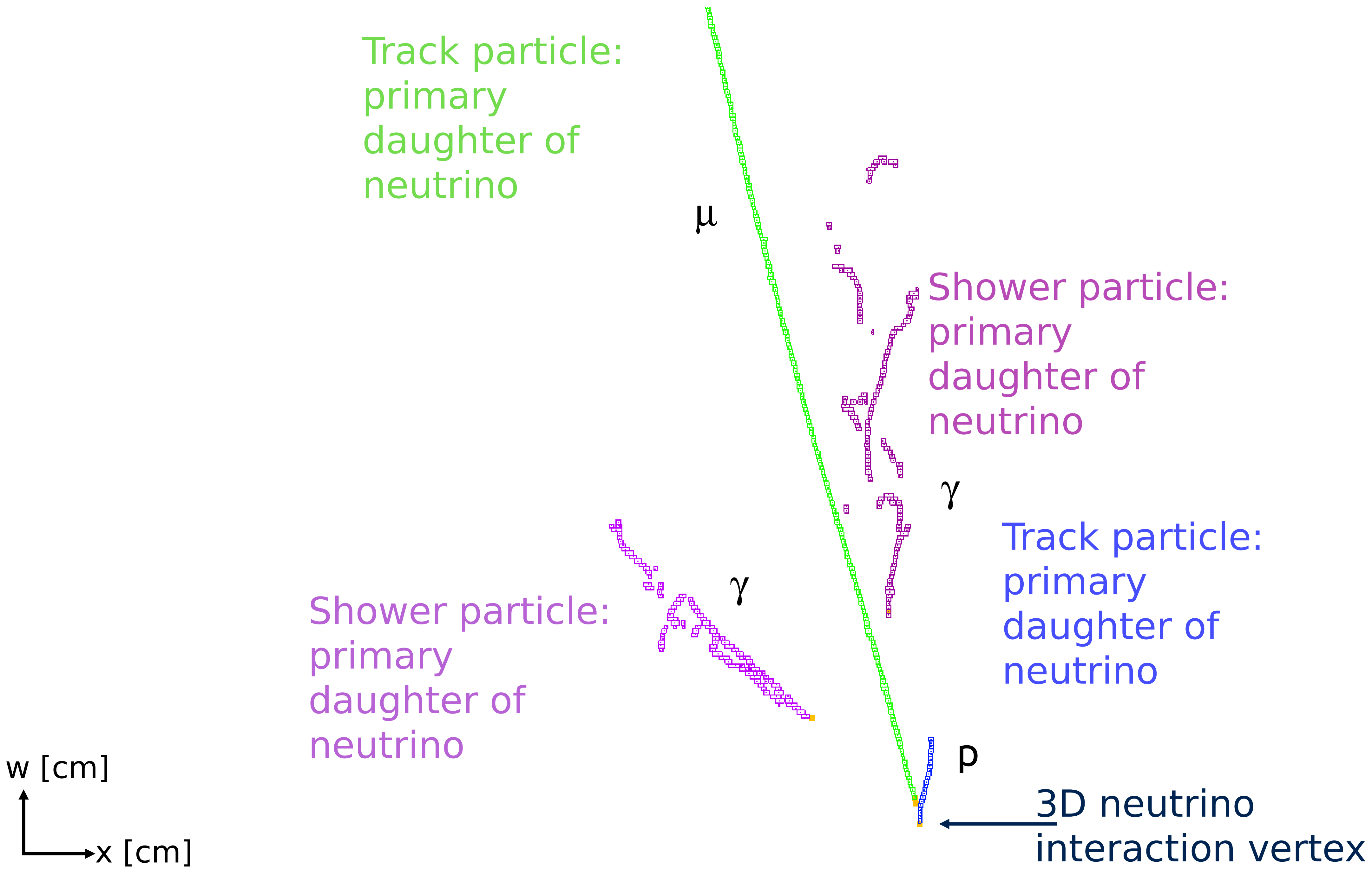
#Events: 6,070
#Perfect: 50.8%

Results for Neutrino 2016





MicroBooNE simulation: BNB νμ CC RES μ, p, π0



Pandora Performance: CC RES μ, p, π0

MicroBooNE simulation: BNB νμ CC RES μ, p, π0

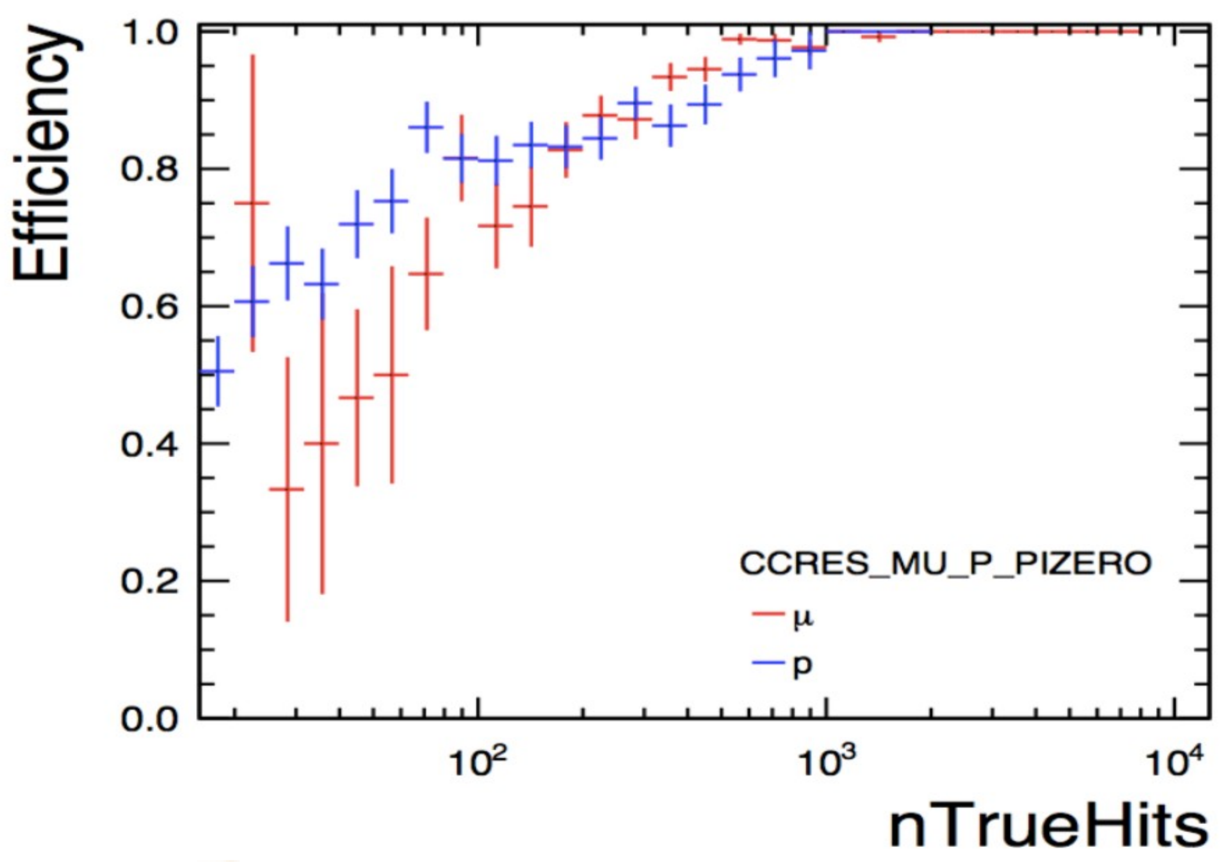
Input: LArSoft v05_08_00

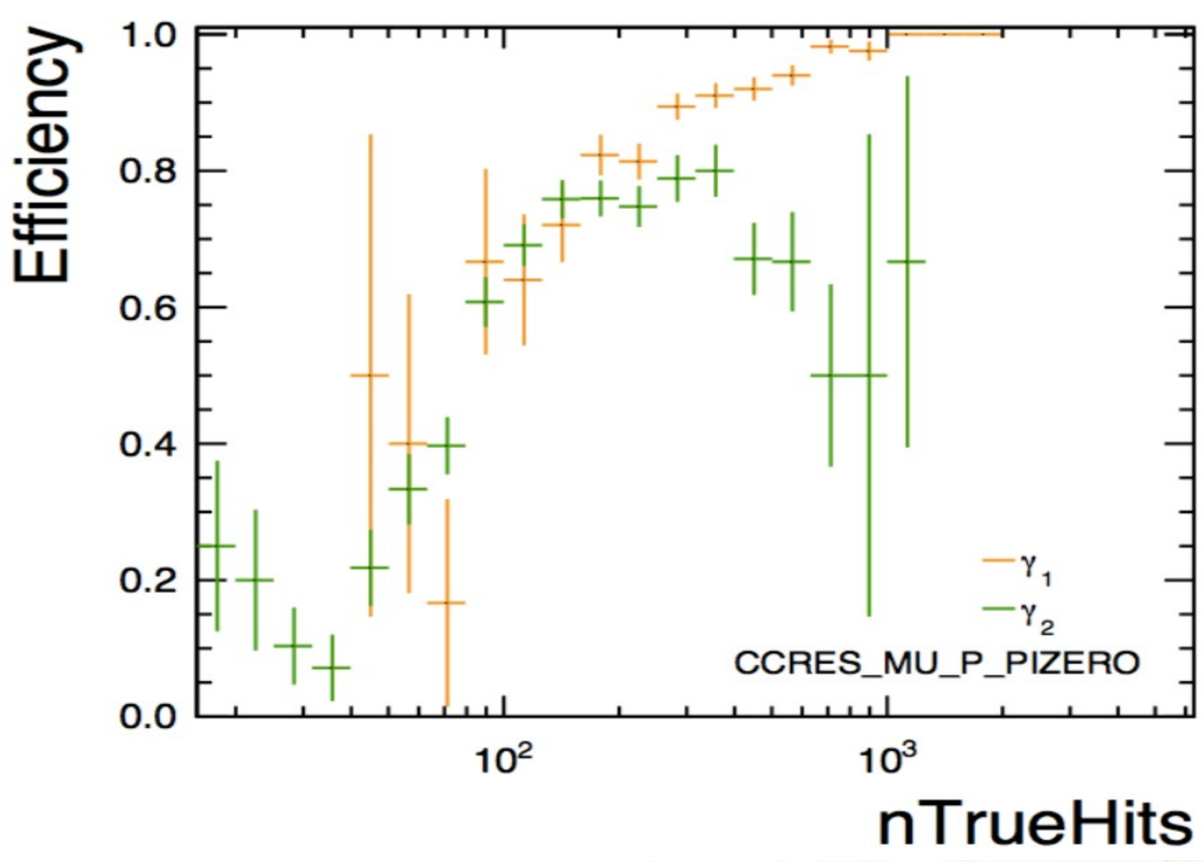
#MatchedPFOs		1	2	3+
μ	*7,9 %*	*87,4 %*	*4,4 %*	*0,3 %*
p	*19,8 %*	*74,0 %*	*5,5 %*	*0,7 %*
γ1	*10,8 %*	*60,0 %*	*18,2 %*	*11,0 %*
γ2	*35,9 %*	*51,0 %*	*10,1 %*	*3,0 %*

#hits(γ1) > #hits(γ2)

#Events: 1,874
#Perfect: 22.3%

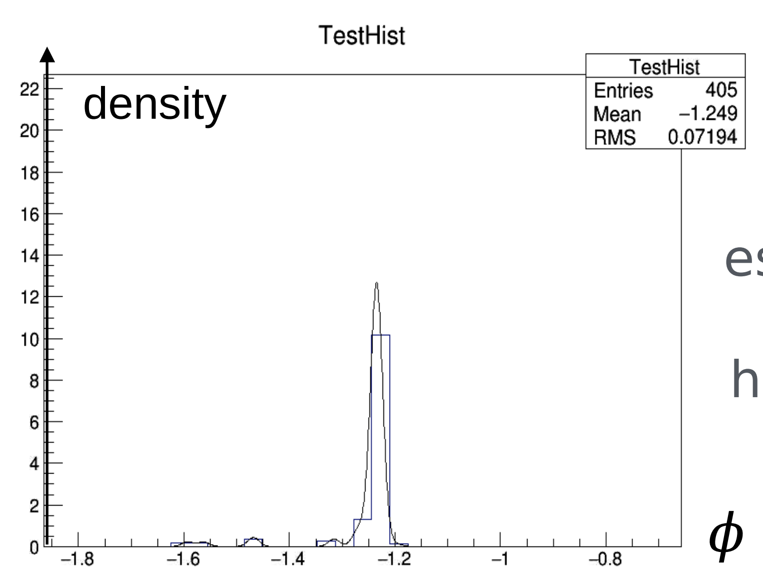
Results for Neutrino 2016



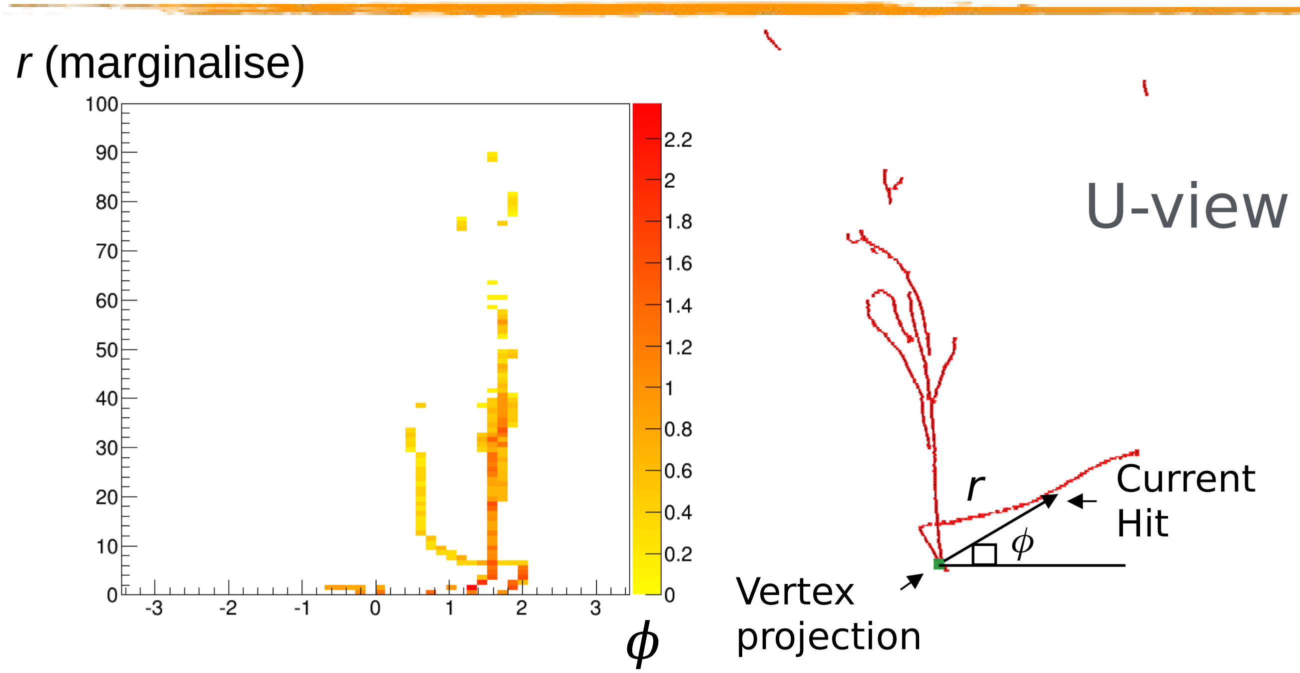


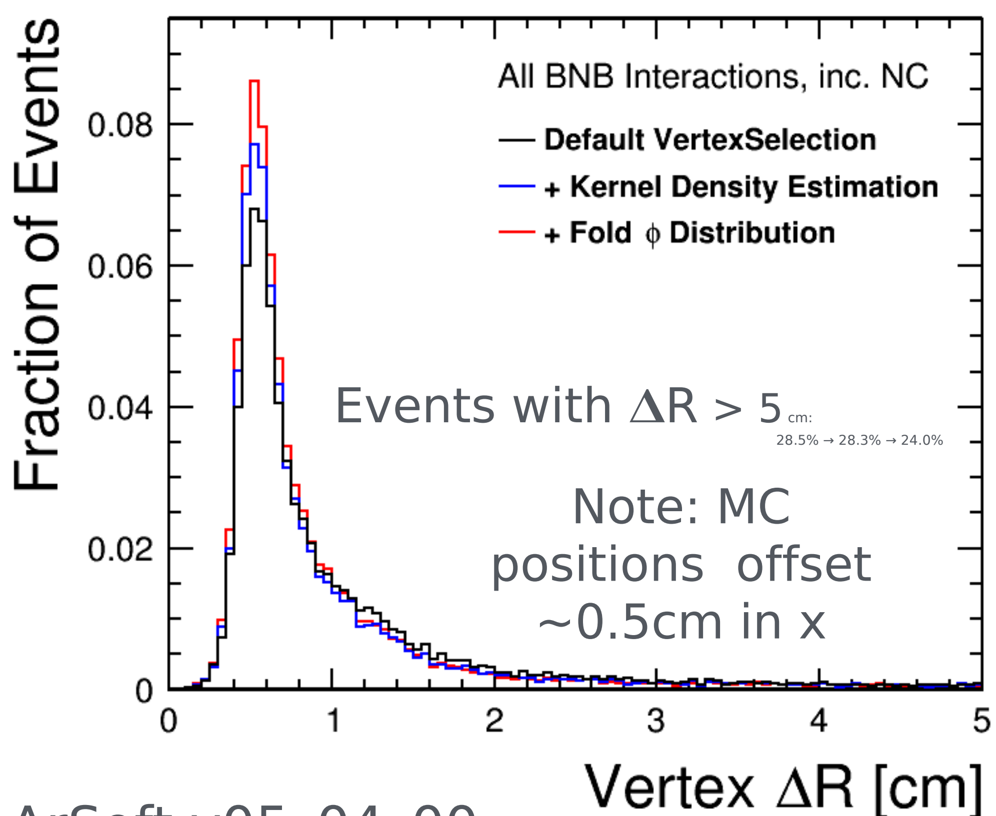
Vertex Improvements

- Use pairs of 2D clusters to produce list of 3D candidate vertex positions. Assess each by examining surrounding hits.
- Use kernel density estimation to provide non-parametric estimations of hit ϕ distributions (each r-deweighted).
- Sample distributions to obtain score that promotes grouping hits in tight ϕ ranges, each for distinct particle leaving vertex.
- Fold ϕ distribution into range 0 to π, with cancellation between π-separated hits to disfavour candidates placed on tracks.
- The existing vertex reconstruction can easily be adapted to incorporate the features of the protoDUNE detector(s)



Kernel density estimation vs. histogram

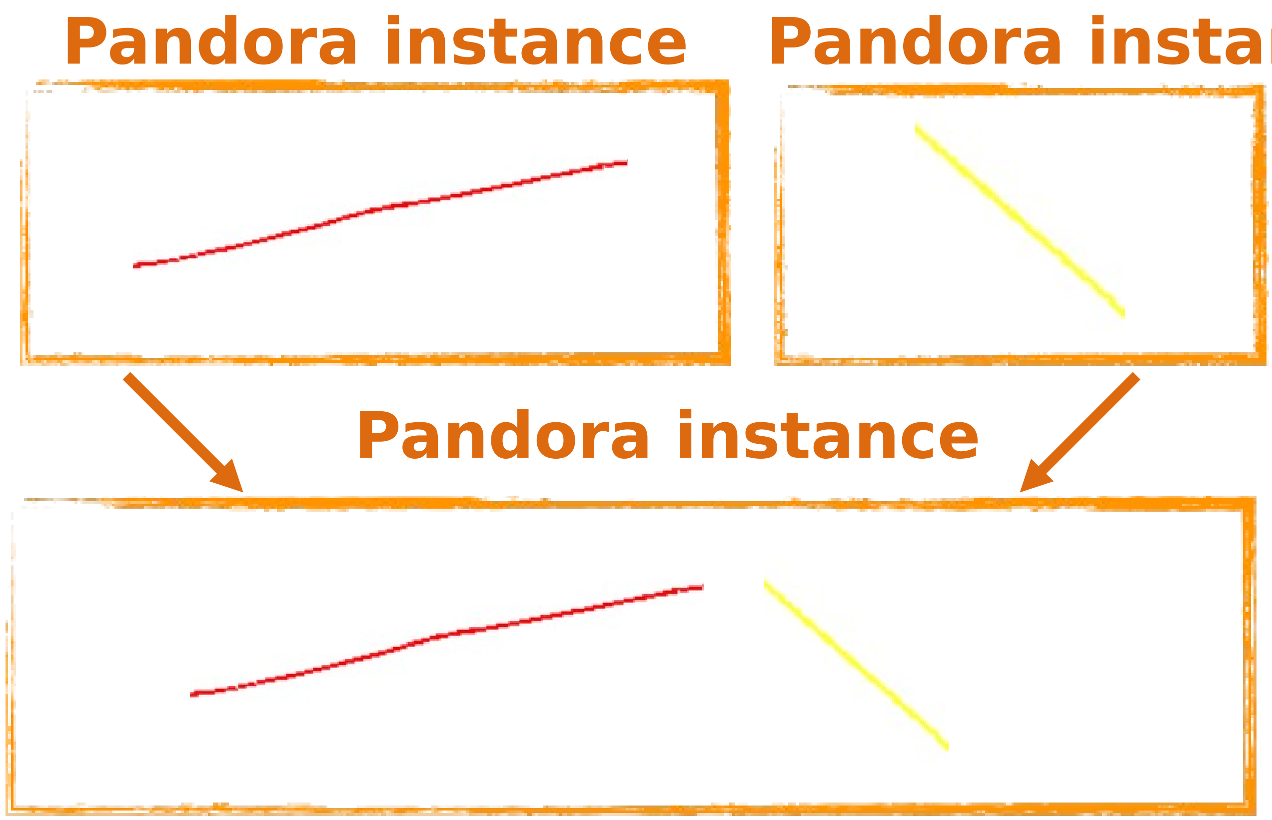




Input: LArSoft v05_04_00

Pandora for DUNE

- Handle events crossing between multiple TPC 'drift volumes':
  - Technical details addressed: Pandora instances apply MicroBooNE reco to each volume.
  - Logic required to decide i) which particles to stitch together and ii) which is the parent.



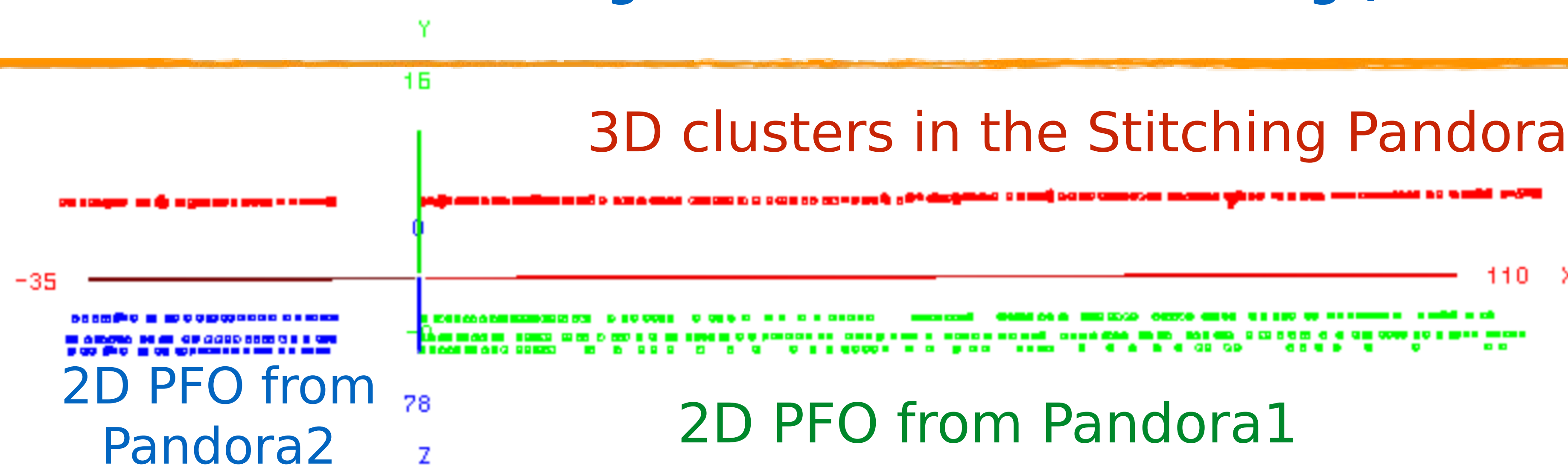
N independent Pandora instances to reconstruct particles in N different drift volumes

N+1 instance grabbing the reconstructed particles and comparing them as 3D particles in a world volume

Logic to match them: Stitching (todo list)

Example display

"sideways" cosmic in DUNE 35t



Pandora for DUNE

- Path for providing Pandora multi-algorithm reconstruction for DUNE seems clear:
  - Pandora interfaces ensure that all MicroBooNE developments reusable for DUNE.
  - Some additions required, but more a case of re-optimisation than all-new development.

- Optimise for event topologies associated with different beam spectrum:
  - Reduce ‘tensions’ between algorithms for tracks, dense showers and
  - sparse showers.
  - Re-consider CPU and memory efficiency for events with larger numbers of hits.

- Achieve high quality communication with reco and analysis working groups.

- Pandora is an open project and new contributors would be extremely welcome.

Thanks for your attention!

Pandora's home on the web: https://github.com/PandoraPFA

Contact details overleaf…

Pandora is an open project and new contributors would be extremely welcome. We'd love to hear from you and we will always try to answer your questions!

Contact details:

Framework development	John Marshall (marshall@hep.phy.cam.ac.uk) Mark Thomson (thomson@hep.phy.cam.ac.uk)
LAr TPC algorithm development	John Marshall Andy Blake (a.blake@lancaster.ac.uk)
Performance metrics and validation	John Marshall Andy Blake Lorena Escudero (escudero@hep.phy.cam.ac.uk) Joris Jan de Vries (jjd49@hep.phy.cam.ac.uk) Jack Weston (weston@hep.phy.cam.ac.uk)

Please visit https://github.com/PandoraPFA

UNIVERSITY OF CAMBRIDGE

Lancaster University

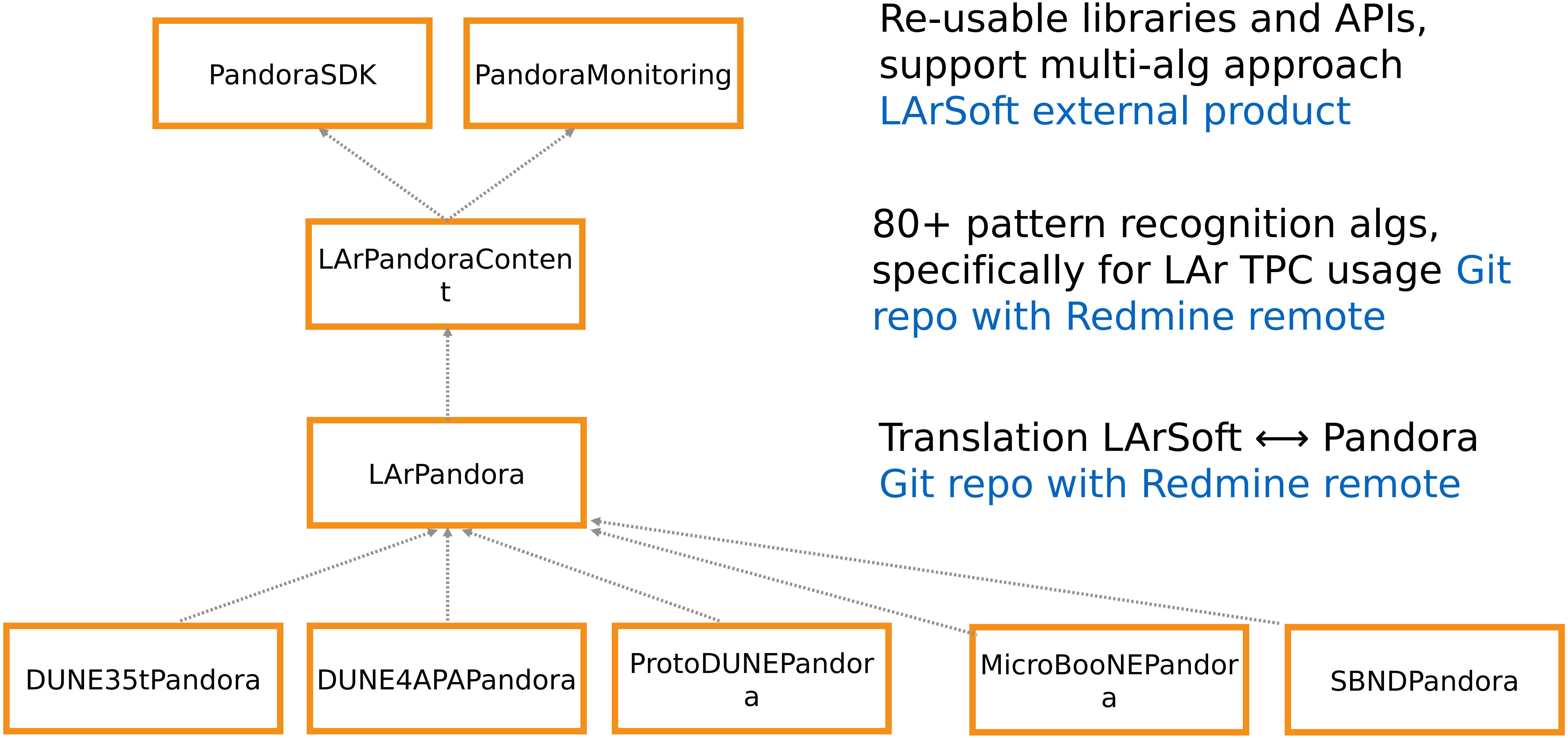
Pandora LArSoft Integration

Place some of the more recent Pandora developments in a LArSoft context:

	LArSoft	Pandora SDK	Pandora Monitoring	Pandora Algorithms	Notes
	v04_25_00	v02_01_00	v02_01_00	v02_03_00	
	…	↓	↓	↓	Vertex improvements
Production release	v04_36_00	Sync. with ILC versions			
	v04_36_01			v02_04_00	
	…	↓	↓	↓	Shower improvements
	v05_04_00	v02_02_00	v02_02_00	v02_05_00	
	…	↓ Add Gap Support	↓	↓	Vertex gap treatment
	v05_07_00	v02_03_00	v02_03_00	v02_06_00	Improved gap treatment
Production release	v05_08_00			v02_07_00	
	…			↓	Algorithm reproducibility
	v05_09_01			v02_07_01	
	…			↓	Improved LArSoft integration
	v05_11_01	↓	↓	v02_07_03	

Pandora LArSoft Integration

mple cartoon showing current packages and an indicative hierarchy:



Re-usable libraries and APIs, support multi-alg approach
LArSoft external product

80+ pattern recognition algs, specifically for LAr TPC usage Git repo with Redmine remote

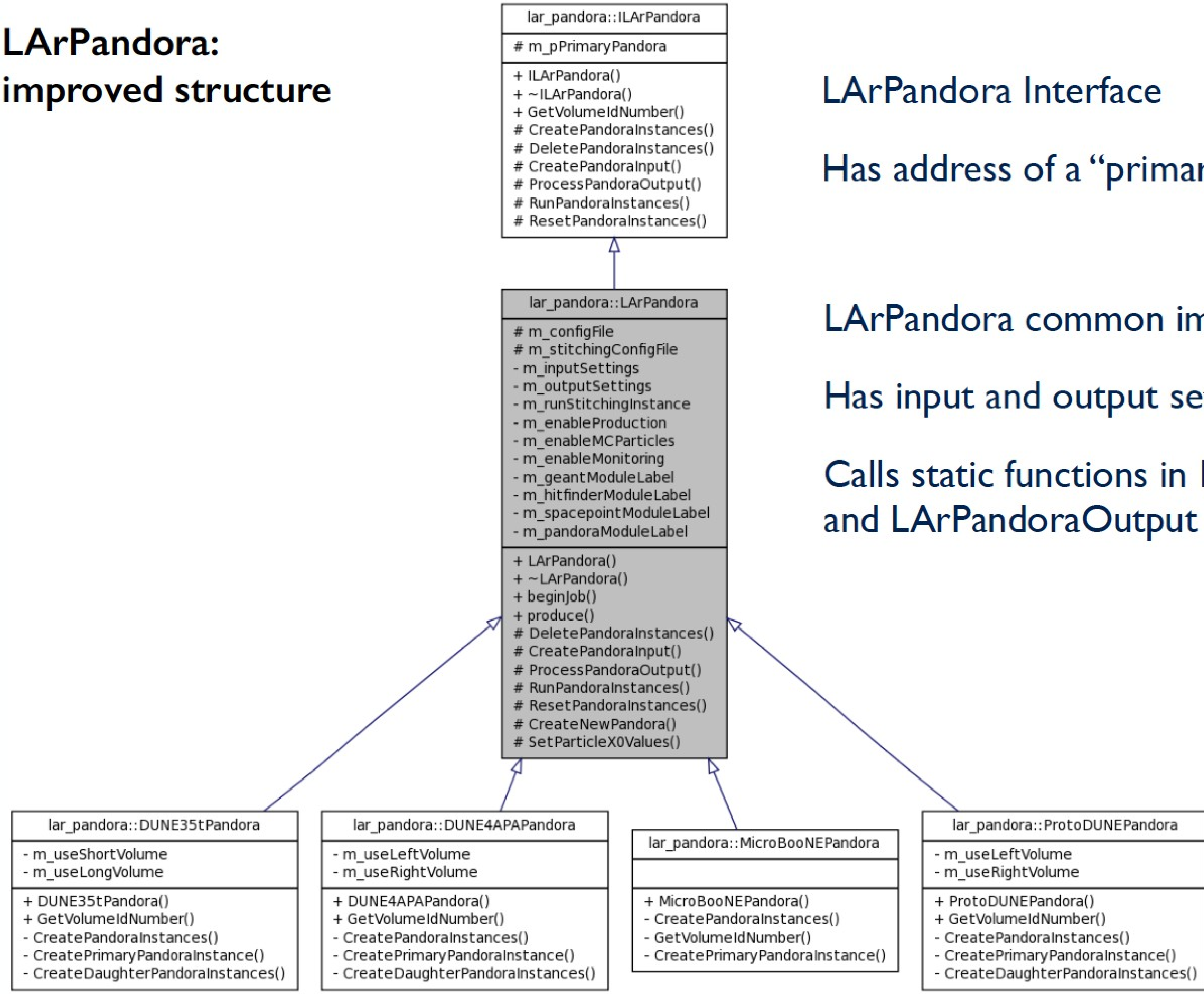
Translation LArSoft ⟷ Pandora
Git repo with Redmine remote

Any detector-specific aspects. Create Pandora Instances, define drift volumes
Git repos with Redmine remote

Ongoing: Replace detector-specific producer modules with a single, abstracted module.

LArPandora Structure

LArPandora: improved structure



LArPandora Interface

Has address of a "primary" Pandora instance

LArPandora common implementation

Has input and output settings instances

Calls static functions in LArPandoraInput and LArPandoraOutput

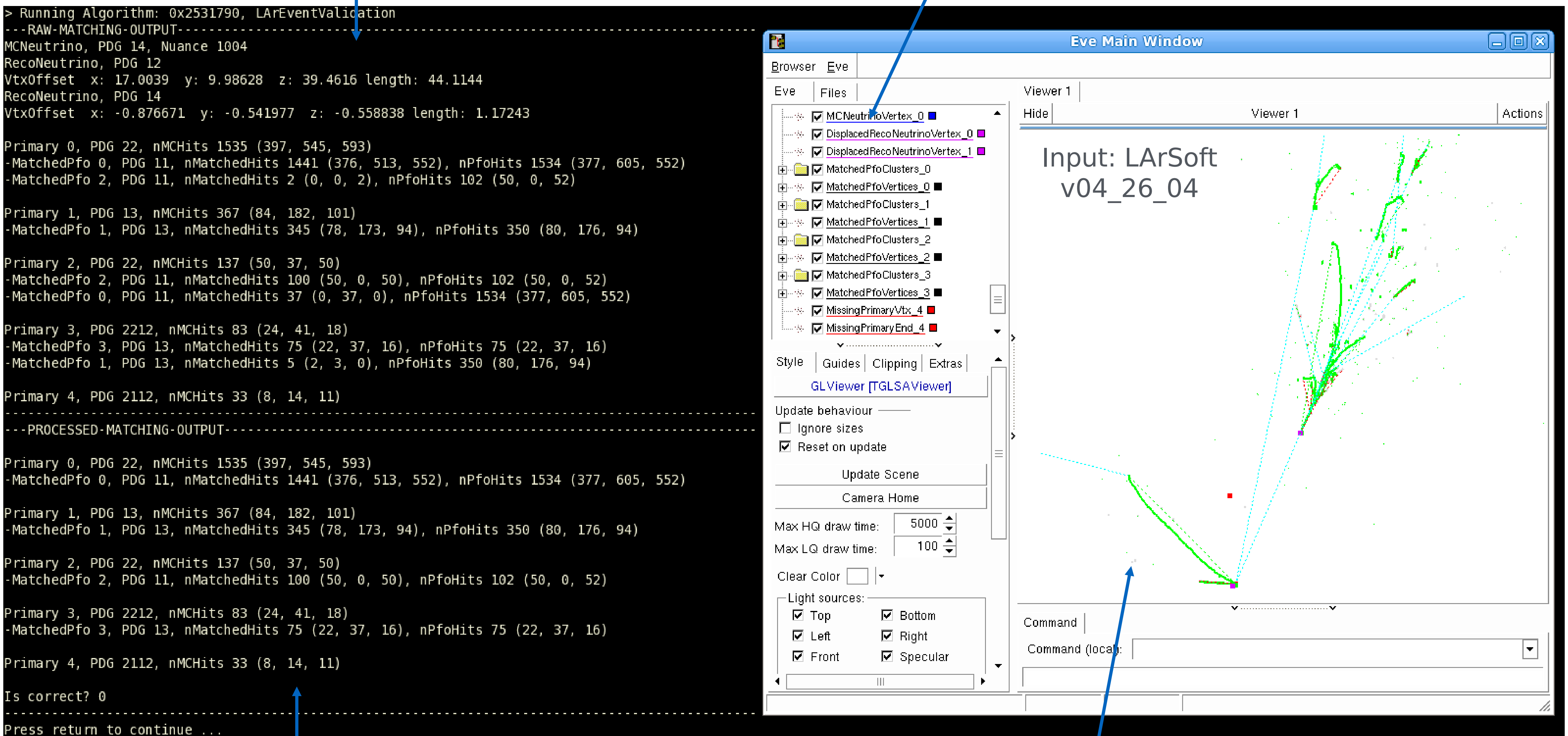
Create Pandora instances

Define drift volumes

Pandora Validation Tools

Comprehensive list of reco particle to MC primary matches

Clickable list of objects for each MC primary
Indication of reco quality: angry colours for poor matches



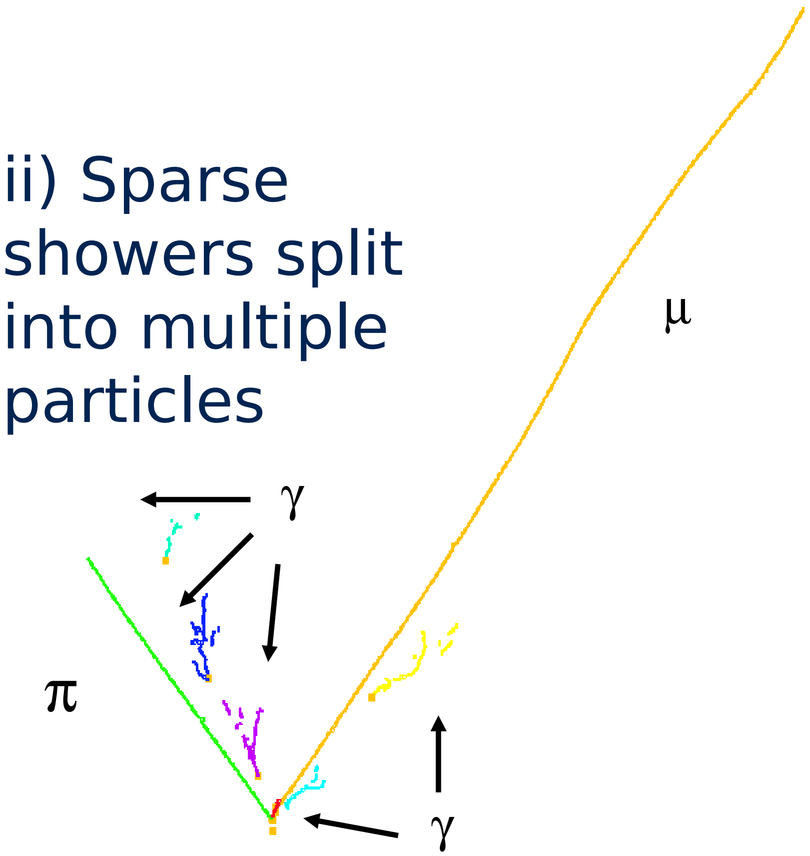
Output of our interpretative matching scheme

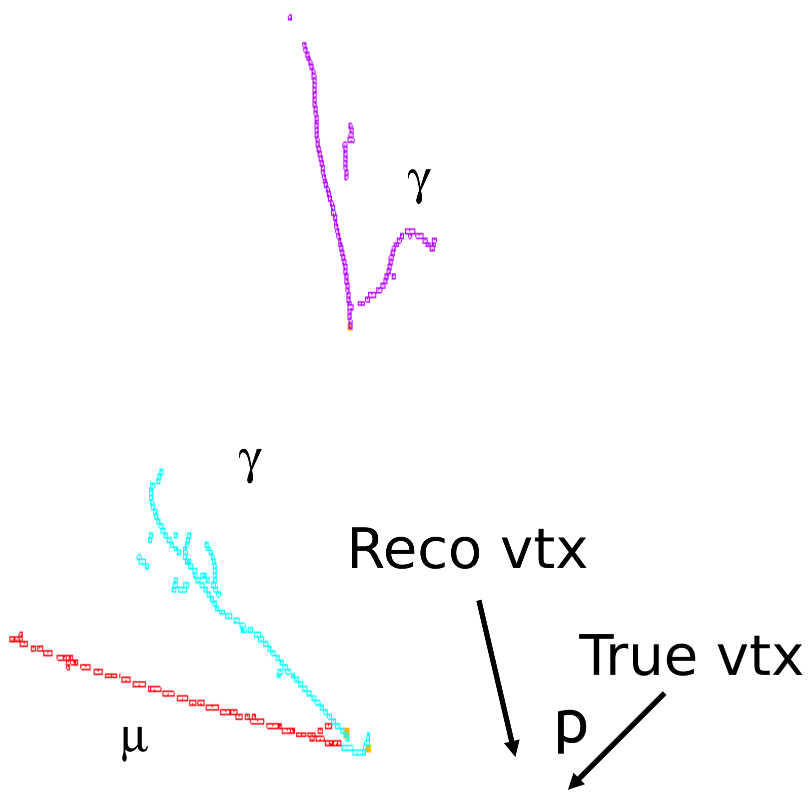
Matched particles appear in green
Split particles appear in red
Markers placed at vtx/end of missing particles

What kind of things go wrong?

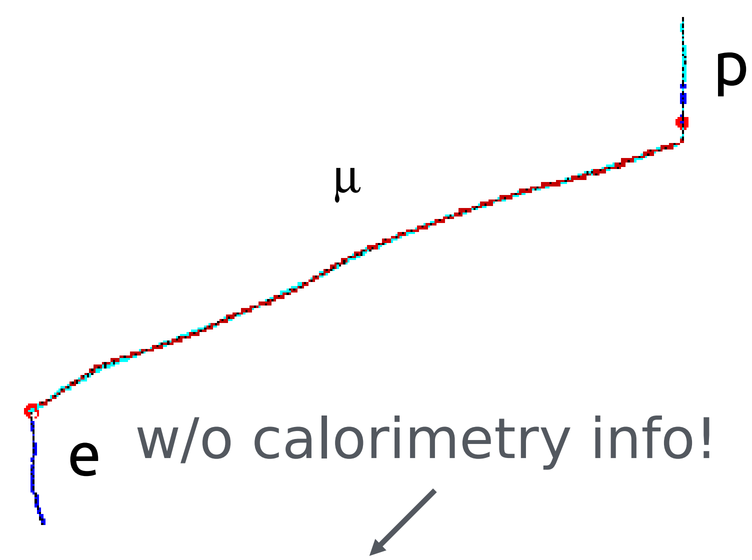
Input: LArSoft v04_16_00

ii) Sparse showers split into multiple particles





i) Reconstructed vertex slightly displaced, so proton is lost



iii) Distinguish between proton and electron? True vtx: high Z.

Pandora Performance Metrics

Metrics are calculated based on reconstructed PFParticles (and their associated collections of 2D hits)

True particles must have ≥15 true hits
Reco/true particles must share ≥5 hits to match

1. Determine the primary true particle in each 2D hit.
   - Use true particle hierarchy to determine primary "reco targets".
   - Associate hits to primary particles making largest E contribution.

2. Match reconstructed particles to true particles:
   - For each reco/true combination, find number of 'matched' 2D hits (common to both reco and true particles). Fold all daughter reco and true particles back into parent primaries.
   - Matching algorithm, find all "strong" matches, then pick-up remaining "weak" matches:
   - Find strongest (most shared hits) match between any reco and true particle
   - Repeat step i, using reco and true particles at most once, until no further matches possible
   - Assign any remaining reco particles to true particle with which they share most hits

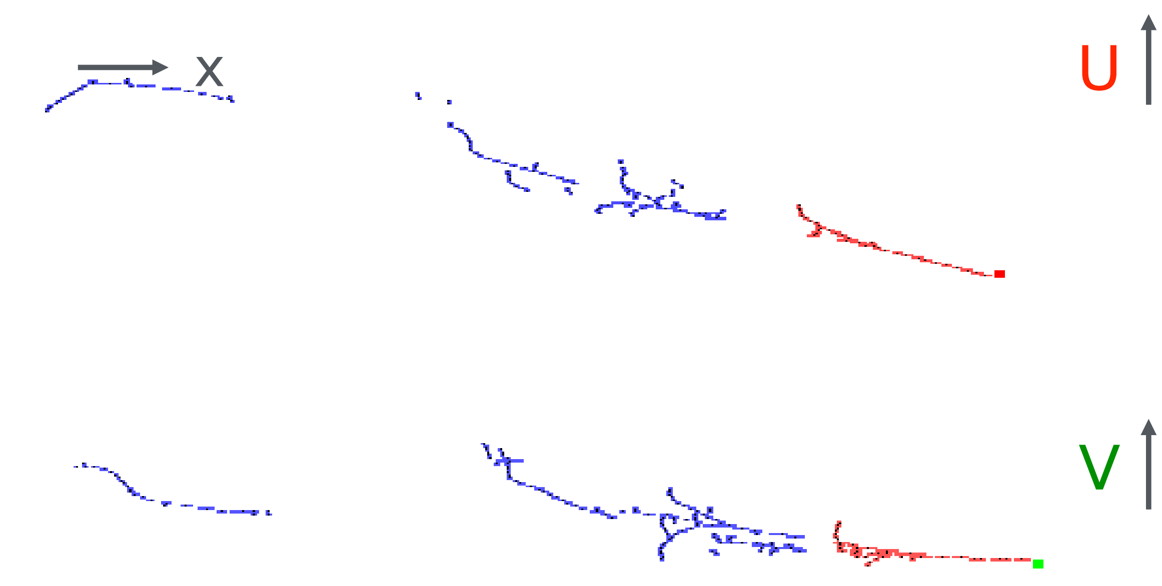
Calculate Performance Metrics:
- 'Efficiency' = fraction of true particles with at least one matched reco particle
- 'Completeness' = fraction of 2D hits in true particle shared with the reco particle
- 'Purity' = fraction of 2D hits in reco particle shared with the true particle

Striving for perfection - look to accurately match one reco particle to each true particle.

Shower Improvements

- Added SplitShowerMerging algorithm to improve completeness of sparse showers:
  - Find vertex-associated particles.
  - Fit cones to constituent 2D clusters.
  - Iteratively pick-up daughter particles.
  - Expect sparse elements labelled as tracks.

- "Looser" version of VertexBasedPfoMerging algorithm e.g. only need clusters in two views.



#MatchedPFOs	0	1	2	3+
μ	4.9% → 5.7%	90.0% → 89.8%	4.6% → 4.1%	0.5% → 0.4%
p	16.7% → 17.7%	76.1% → 75.6%	6.3% → 5.8%	0.9% → 0.9%
γ1	4.5% → 5.8%	55.3% → 63.6%	25.0% → 19.0%	15.2% → 11.6%
γ2	23.4% → 26.0%	58.9% → 58.7%	13.6% → 12.0%	4.1% → 3.3%

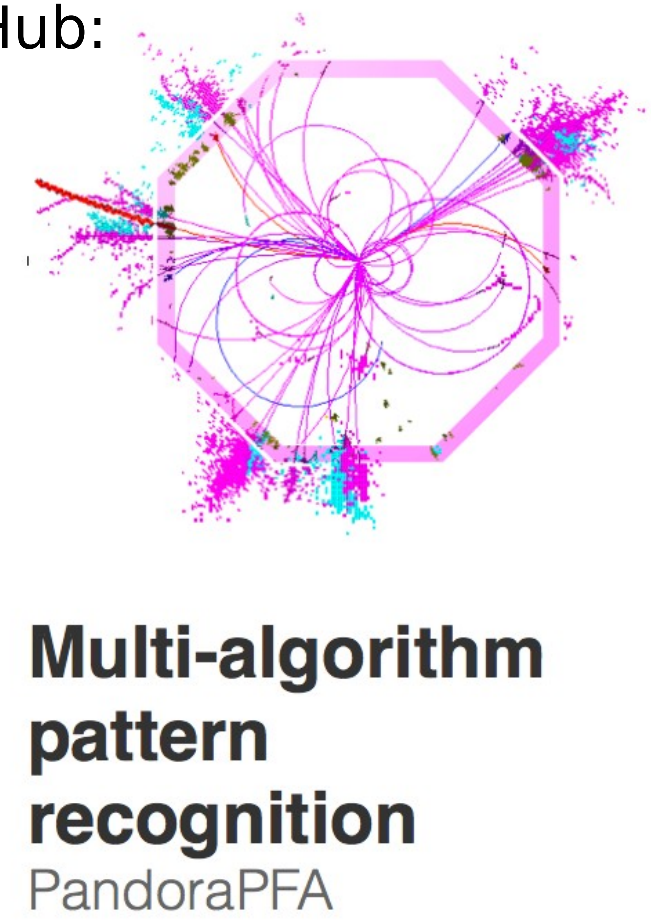
BNB νμ CC RES μ, p, π0 Input: LArSoft v04_16_00

#hits(γ1) > #hits(γ2)

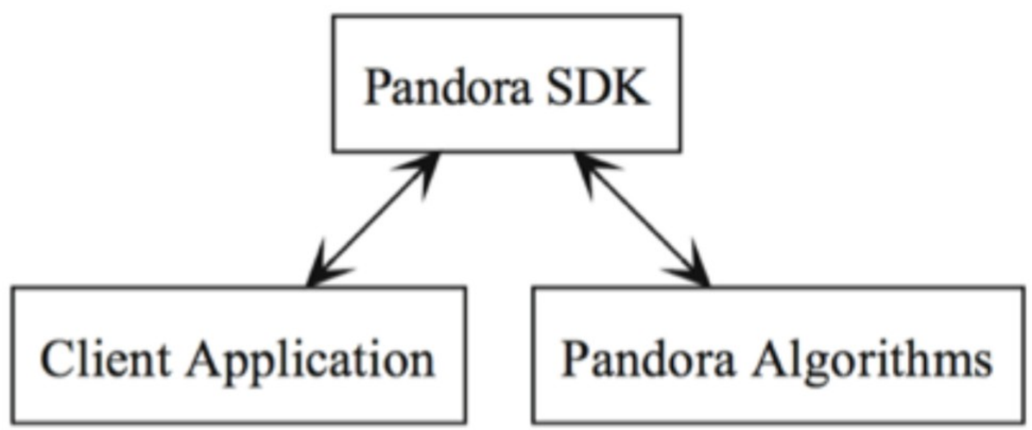
Still work to do: Pandora approach means each alg only makes "safe" cluster merges, so do expect left-over elements.

Pandora: Open Source

Pandora is an open source project, code is available from GitHub:
https://github.com/PandoraPFA (EPJC.75.439)



Code is distributed with LArSoft



LArPandoraInterface

User provides input information receives output

LArPandoraContent

User can add own algorithms, specify algorithm configuration via PandoraSettings XML file

See documentation here:
https://github.com/PandoraPFA/Documentation