LAr Event Reconstruction with the PANDORA Software Development Kit

Andy Blake, John Marshall, Mark Thomson
(Cambridge University)

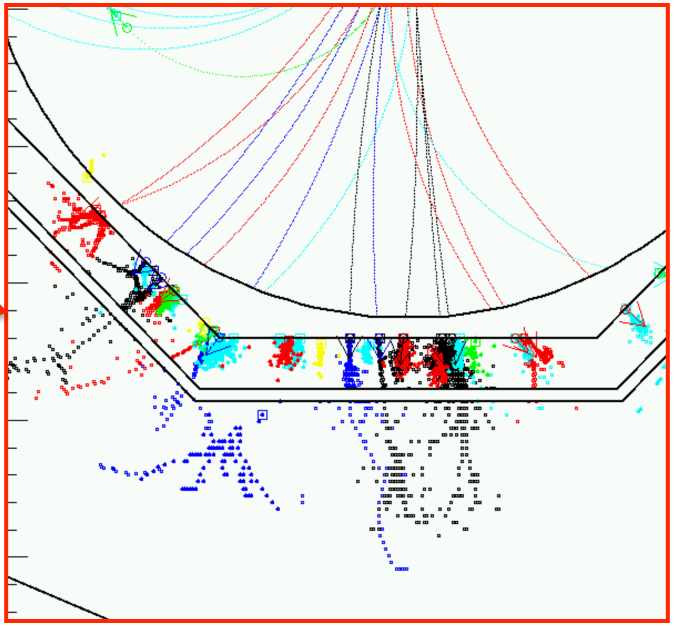
UK Liquid Argon Meeting,
Manchester, November 28th 2012.

From ILC/CLIC to LBNE

- **Cambridge is the world-leader in high granularity particle flow calorimetry for a future collider experiment**
- **PandoraPFA particle flow reconstruction software developed to provide a proof-of-principle of the concept for the ILC**

  M. A. Thomson, *NIMA 611 (2009) 25-40.*
- **Now used in all ILC and CLIC full simulation physics studies**

Aim to reconstruct individual particles in an event



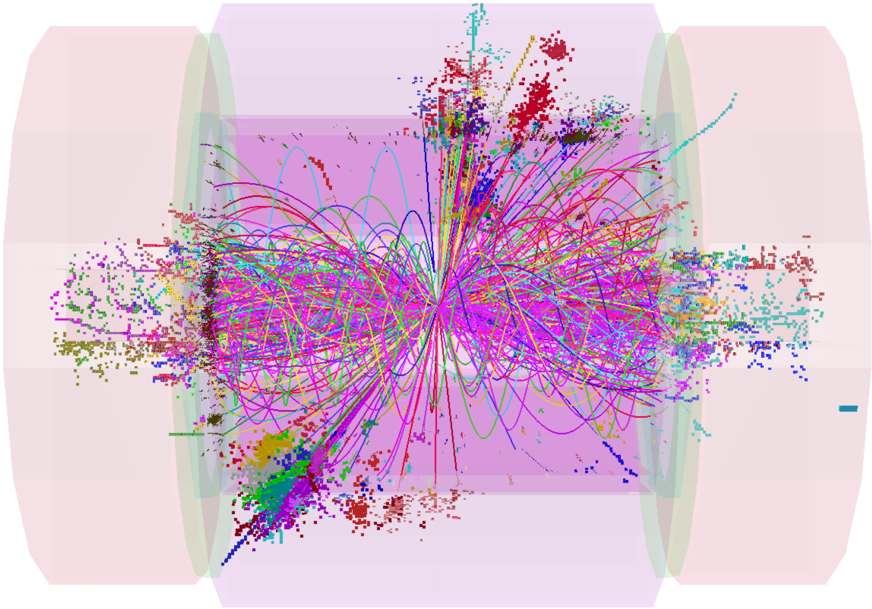
From ILC/CLIC to LBNE

- **Original code was rewritten from scratch and now exists in the form of the PANDORA Software Development Kit for event reconstruction (Mark Thomson, John Marshall).**
  - Analysis tools and template algorithms for pattern recognition.
  - Developed for multi-purpose and multi-experiment use.
  - Slots into any experiment's software: 'framework agnostic'.
- **The PANDORA readily applicable to LAr reconstruction.**
  - Serious effort in last few months

Pandora

- **Pandora is a software toolkit for developing and running pattern recognition algorithms. It provides the following:**
  - Tools for analysing the topology of particle interactions.
  - Template algorithms for reconstructing tracks and showers, using topological information.
  - An environment for building reconstruction algorithms.
  - Visualisation tools
  - A set of robust APIs for running reconstruction tasks.
  - A set of reconstruction objects, managed using STL containers.
- **A single-library C++ framework.**
  - No dependencies (other than ROOT-based event display).

H^+H^- production in CLIC detector, reconstructed by PandoraPFA.



$e^+e^- \rightarrow H^+H^- \rightarrow t\bar{b}b\bar{t} \rightarrow 8$ jets

https://svnsrv.desy.de/viewvc/PandoraPFANew/

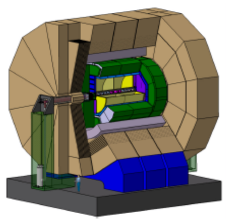
Pandora

Non-reusable e.g. detector specific

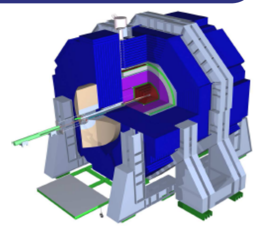
External Content Libraries

Isolates specific detector and software details, creating self-describing hits, tracks, etc.

Pandora Client Application



ILD/Marlin



SiD/org.lcsim

Reusable, applicable to multiple detectors

Pandora Content Libraries

FineGranularity Content

CoarseGranularity Content

Register, via APIs

Runs registered content and performs book-keeping

Pandora Framework

Algorithm Manager

CaloHit Manager

Cluster Manager

Plugin Manager, etc.

Pandora content:

algorithms, particle id functions, energy correction functions, shower profile calculators, etc...

Software Engineering

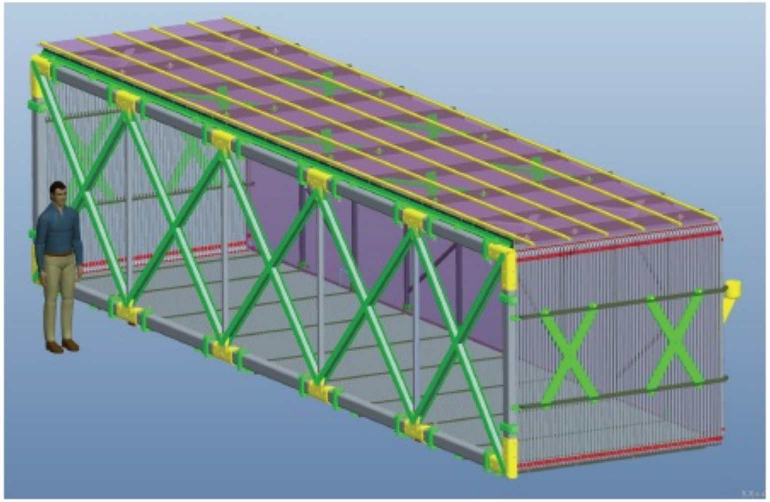
- **In the PandoraPFA rewrite - designed from scratch**
- **Flexibility built in from the start**
- **Efficient memory management and data containers**
  - for efficient CPU/memory performance with high hit densities
- **Algorithms can only access data through APIs**
  - Memory management performed by framework
  - Data encapsulation reduces possibility of coding errors
- **Powerful framework**
  - e.g. iterative reconstruction where part of an event is reconstructed multiple times
- **Flexibility**
  - moving from ILC to LAr reconstruction, involved almost no code changes (just new algortithms)

LArSoft

- **Currently, reconstruction efforts are centred on the Fermilab LAr programme.**
  - Simulated data from LBNE and MicroBooNE.
- **The underlying software framework is LArSoft.**
  - A common framework for all the Fermilab LAr experiments (ArgoNeuT, MicroBooNE, LBNE etc…).
  - Provides simulation, calibration, reconstruction, geometry.

LBNE and MicroBooNE

- **Both LBNE and MicroBooNE have wire readout (3x 2D views).**
  - Two induction planes (‘U’ and ‘V’), with wires aligned at 60 degrees to the vertical; and a collection plane (‘Y’), with vertical wires.
  - Therefore, need to develop reconstruction algorithms that handle 2D views and perform 2D→3D combination.
- **For this work, use beam neutrino interactions simulated with LBNE energy spectrum and MicroBooNE detector geometry:**



Schematic of MicroBooNE TPC

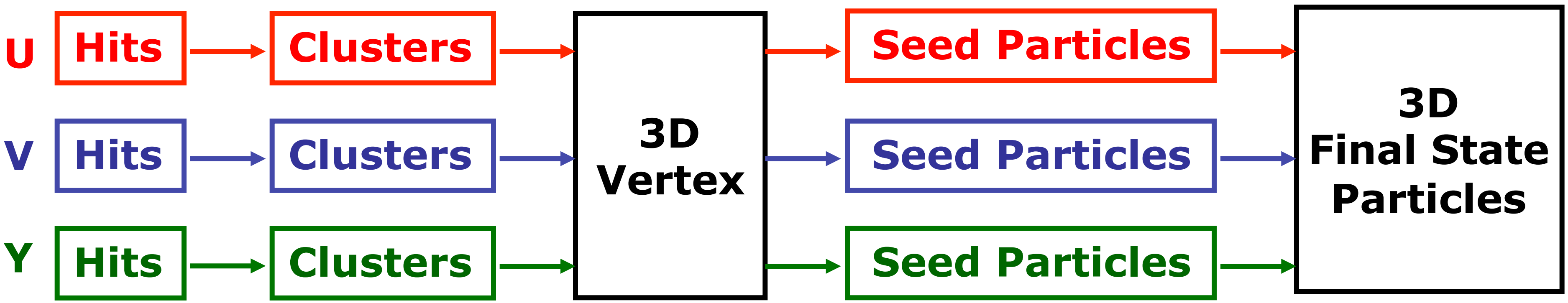
- **Run LArSofy hit-finder**
- **Pass the reconstructed hits to Pandora for pattern recognition.**
  - At this point decoupled from LArSoft

Reconstruction Strategy

2D Reconstruction (Prototype reconstruction, using one view):

Strategy: Reconstruct 2D tracks and showers simultaneously.

3D Reconstruction (Full reconstruction, using all three views):



Strategy: Use 3D information as much and as early as possible.

Full 3D Reconstruction (simple? adaption of 2D algorithms):

2D Reconstruction

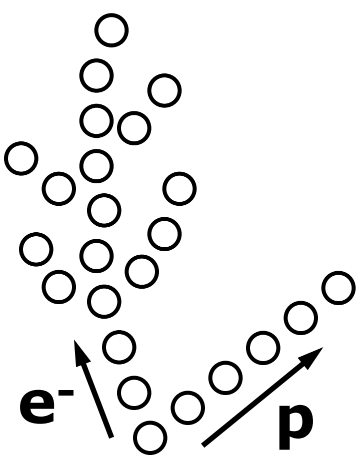
- **<u>Current status</u>: so far, have tools in Pandora to develop a 2D (i.e. single-view) reconstruction chain.**
  - Have developed a full chain of reconstruction algorithms.
  - Each algorithm is a prototype for further development.
  - Currently, we reconstruct each view (U,V,Y) separately, without using 3D information.
- **<u>Basic reconstruction strategy</u>: use topological associations to merge hits into clusters and then final-state particles.**
  - Pandora provides tools for forming topological associations between hits and clusters, and has template algorithms for merging together hits based on these associations.
- **<u>Pandora approach</u>: break down the problem into many separate algorithms, each with a particular purpose.**
  - Each algorithm grows the event in a particular way.
  - Try to be conservative: each algorithm should only make definite merges. It's hard to undo bad merges!

2D Reconstruction

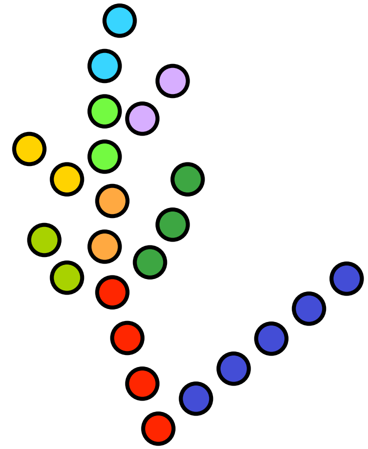
1. **Start by merging together hits to form clusters.**
   – Use nearest-neighbour clustering algorithm to join together contiguous hits.
2. **Use topological associations (proximity and pointing) to grow clusters.**
   – Making end-to-end merges, creating extended clusters. Identify 'spine' of event.
3. **Use resulting extended clusters to reconstruct vertex.**
   – Find the location that best 'connects' the event.
4. **Identify final-state particles emitted from vertex.**
   – Apply 'length-growing' algorithms to grow tracks.
   – Apply 'branch-growing' algorithms to grow showers.
5. **Use topological associations to add remaining clusters.**
   – 'Box' algorithm for enclosed clusters.
   – 'Parallel' algorithm for neighbouring clusters.
   – 'Cone' algorithm for downstream clusters.

2D Reconstruction

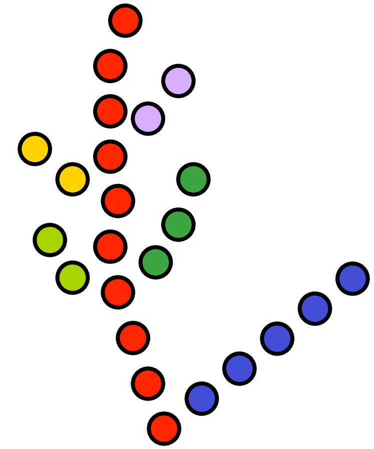
Example: ν_e CC QE



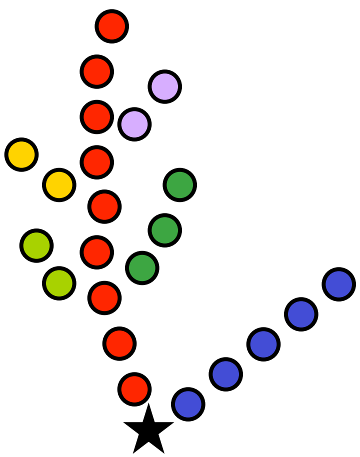
1. Cluster Creation



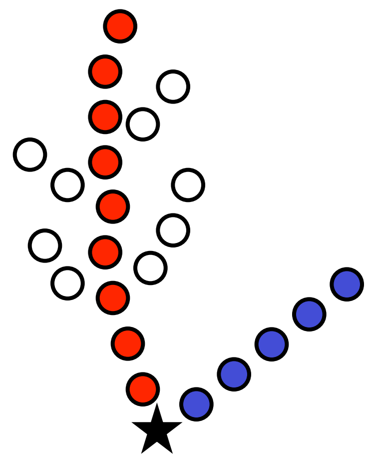
2. Cluster Extension



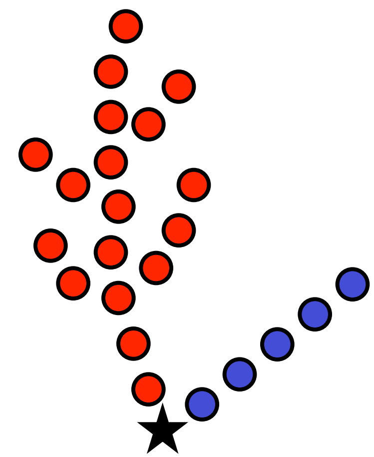
3. Vertex Finding



4. Seed Particles



5. Consolidation



2D Reconstruction

- **So far, we've developed 18 algorithms that implement these steps.**
  - Each algorithm performs a particular role in the reconstruction.
    - For potted summary, see appendix at end of this talk.
  - Continuing to tune and improve + adding more
  - Currently pure 2D reconstruction. No attempt at using 3D information yet.

Algorithms	Step
ClusterCreation	**1. Create clusters**
ClusterAssociation **ClusterExtension** **KinkSplitting**	**2. Extend clusters**
VertexFinding	**3. Identify vertex**
VertexParticleFinding **ParticleLengthGrowing** **ParticleFinding** **ParticleBranchGrowing** **ParticleConsolidation** **ParticleRelegation**	**4. Find final-state particles (tracks and showers)**
ParallelMerging **BoundedBoxMerging** **ConeBasedMerging** **RemnantClustering** **RemnantConsolidation** **IsolatedHitMerging**	**5. Add remaining hits and clusters to final-state particles**
2DParticleCreation	

Philosophy

- **Each algorithm is relatively simple**
  - performs a specific job

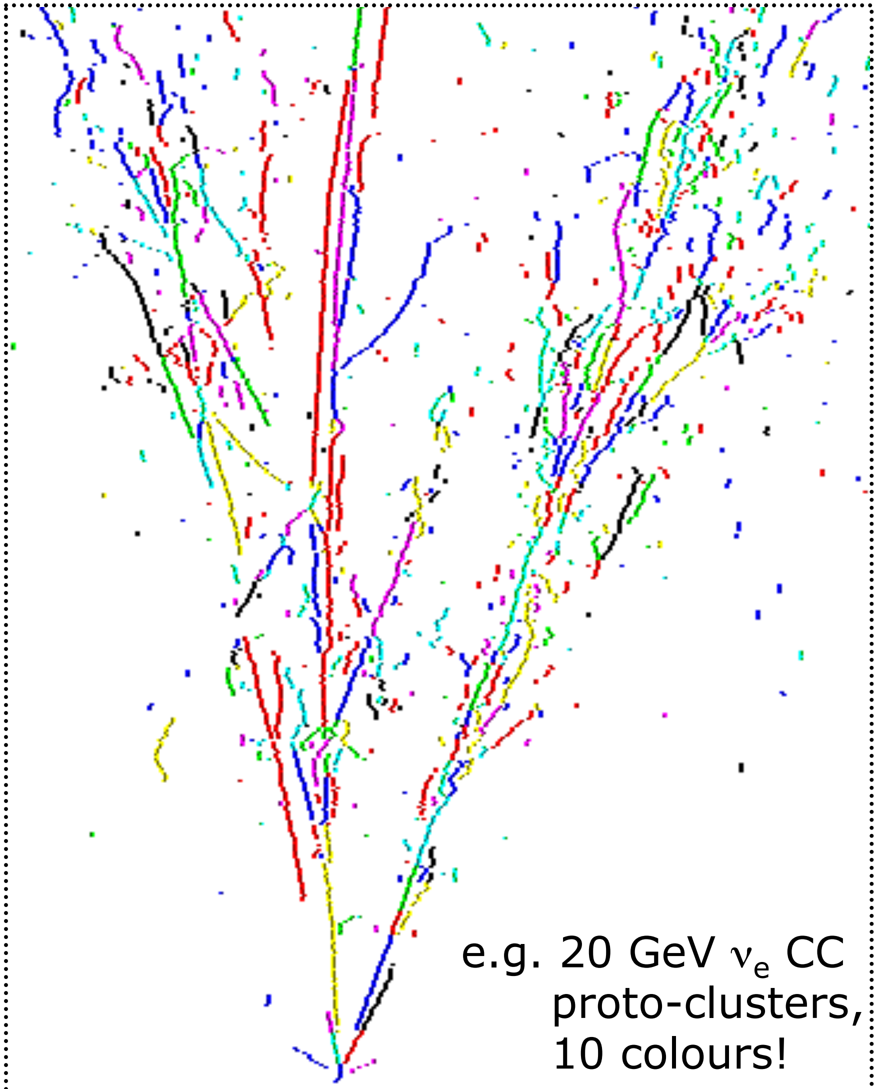
- **These are not generic reconstruction algorithms**
  - they know about physics

- **Connection to underlying physics is central to PandoraPFA approach**

- **So how does it work in practice…**

1. Cluster Creation

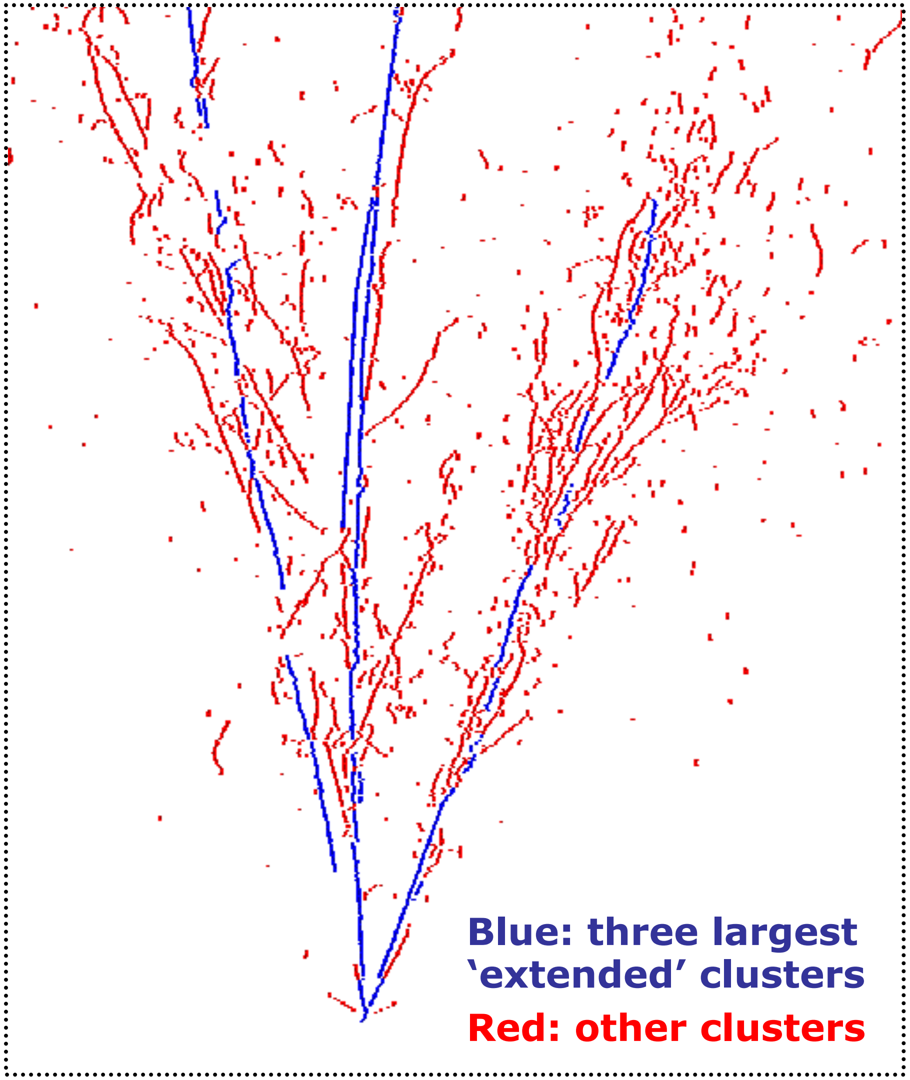
- **Start by merging together groups of hits to form an initial set of proto-clusters.**
- **Method**:
  - Join together contiguous hits using a custom-built nearest neighbour algorithm.
- **Results**:
  - Obtain a patchwork of clusters that can be joined together to form candidate particles.
    - ◇ We just need to make the correct joins!
  - Note that most proto-clusters have good pointing information.
    - ◇ Valuable in determining which joins are correct.



e.g. 20 GeV ν_e CC proto-clusters, 10 colours!

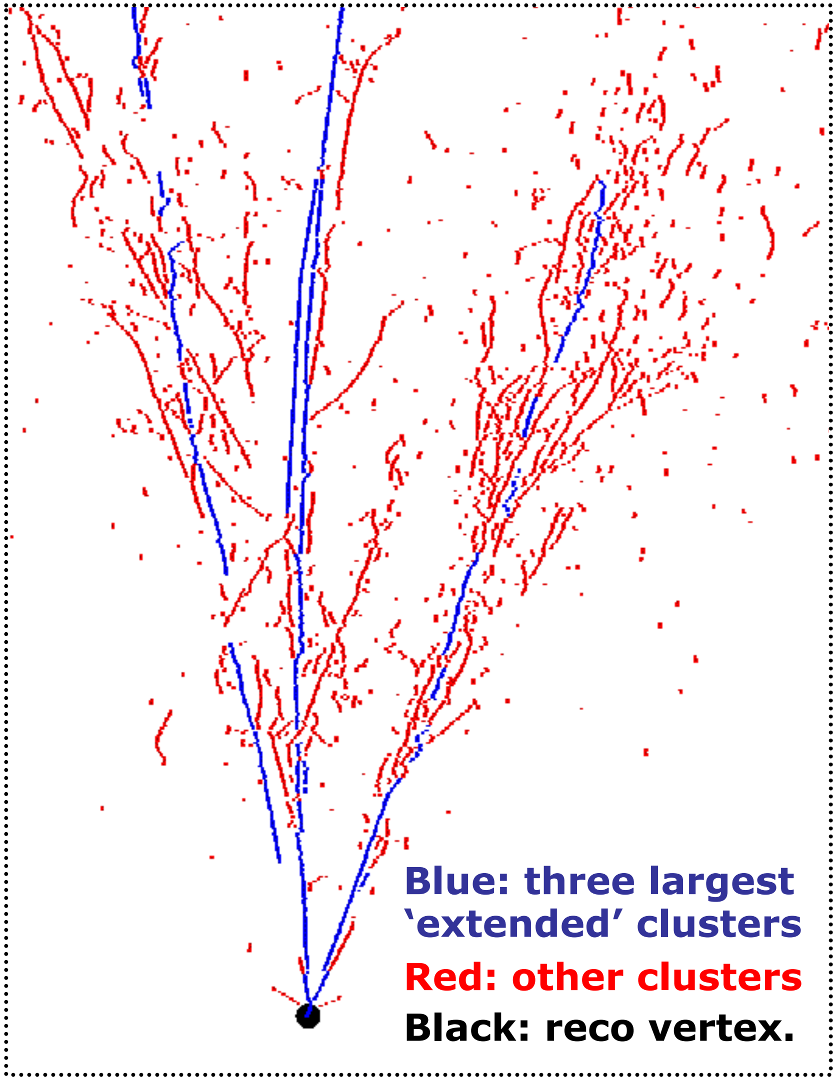
2. Cluster Extension

- **Next, reconstruct the 'spine' of each event by making end-to-end joins between proto-clusters.**
- **<u>Method</u>:**
  - Compare all possible pairs of clusters and generate a matrix of possible end-to-end joins.
    - ◇ Small angular deviation.
    - ◇ Small impact parameter.
  - Then make 'good' joins that maximise 'length'.
    - ◇ Treat ambiguities with care.
- **<u>Results</u>:**
  - Does the job, but needs work. Crucial to do this step well !!



3. Vertex Reconstruction

- **Extended clusters provide enough information to identify vertex.**
- **<u>Method</u>:**
  - For a given vertex hypothesis, apply a 'fast reconstruction', which joins up extended clusters according a simple physics model.
  - Calculate a likelihood function which evaluates the degree of 'connectedness'.
    - ◇ Favours: nodes, emissions, prongs, branches etc…
    - ◇ Disfavours: bad pointing, large angles, isolation etc…
  - Able to find displaced vertices (i.e. $\pi \rightarrow \gamma\gamma$) as well as secondary decays/interactions.



Blue: three largest 'extended' clusters

Red: other clusters

Black: reco vertex.

3. Vertex Reconstruction

- **Vertex Finding Algorithm (contd.)**

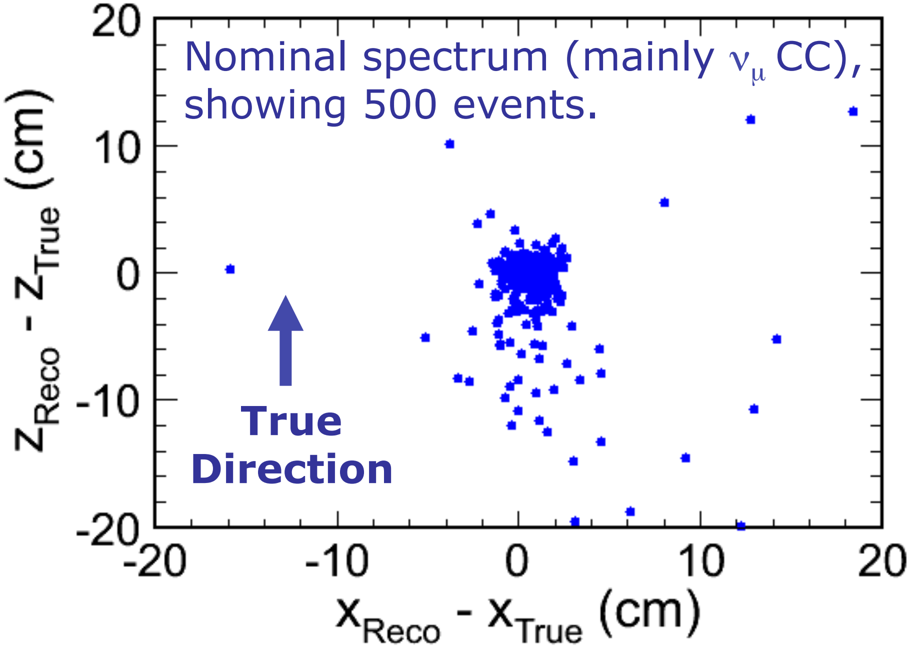
  Results:

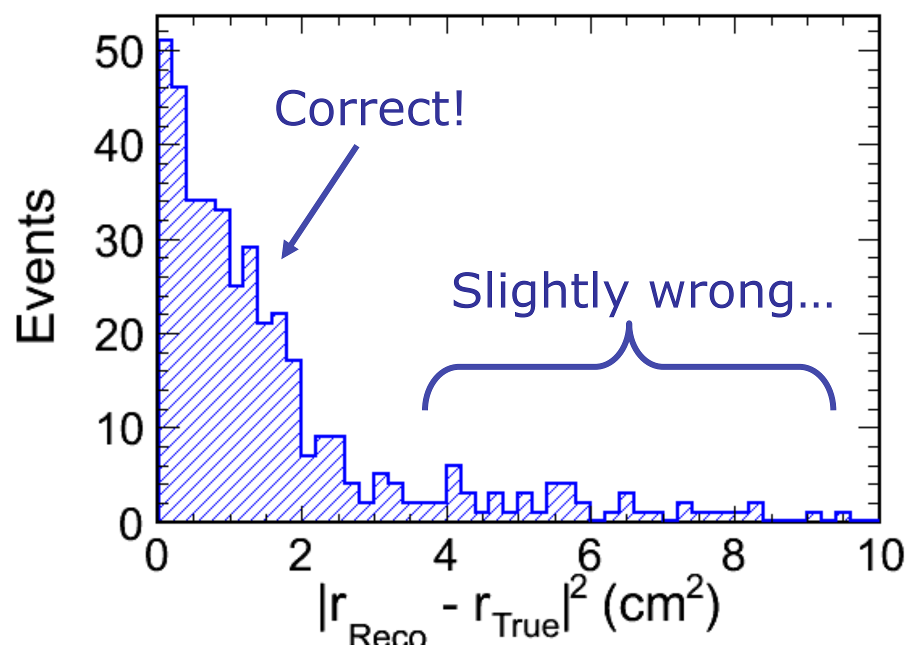
  ◇ Most events well-reconstructed (68% of events within 1.5 cm).

  ◇ However, a substantial tail (10%) have overshot the true vertex.

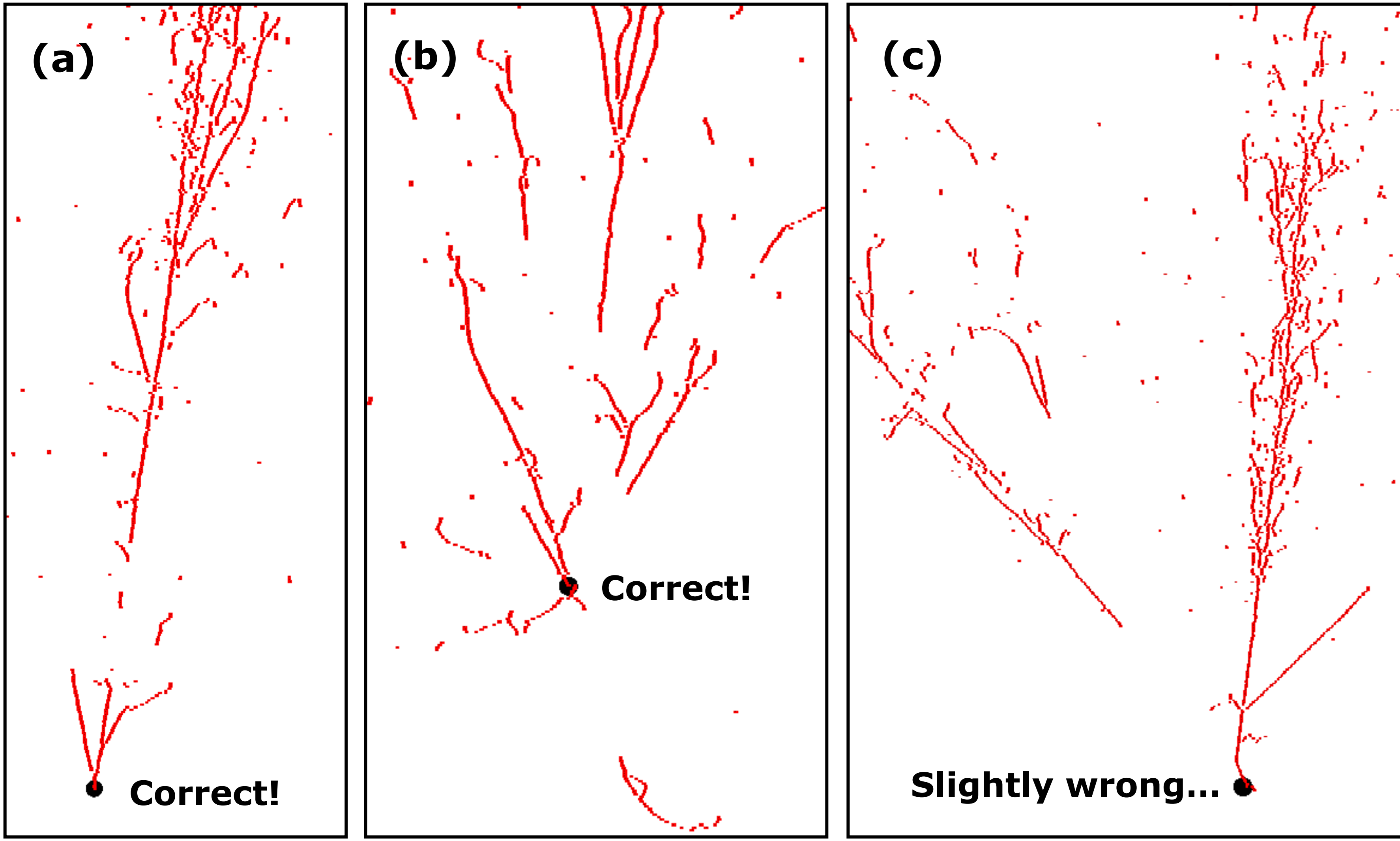
Summary of vertex resolution

Resolution	Nominal	$\nu_\mu \leftrightarrow \nu_e$
68%	1.5 cm	1.6 cm
90%	7 cm	6 cm



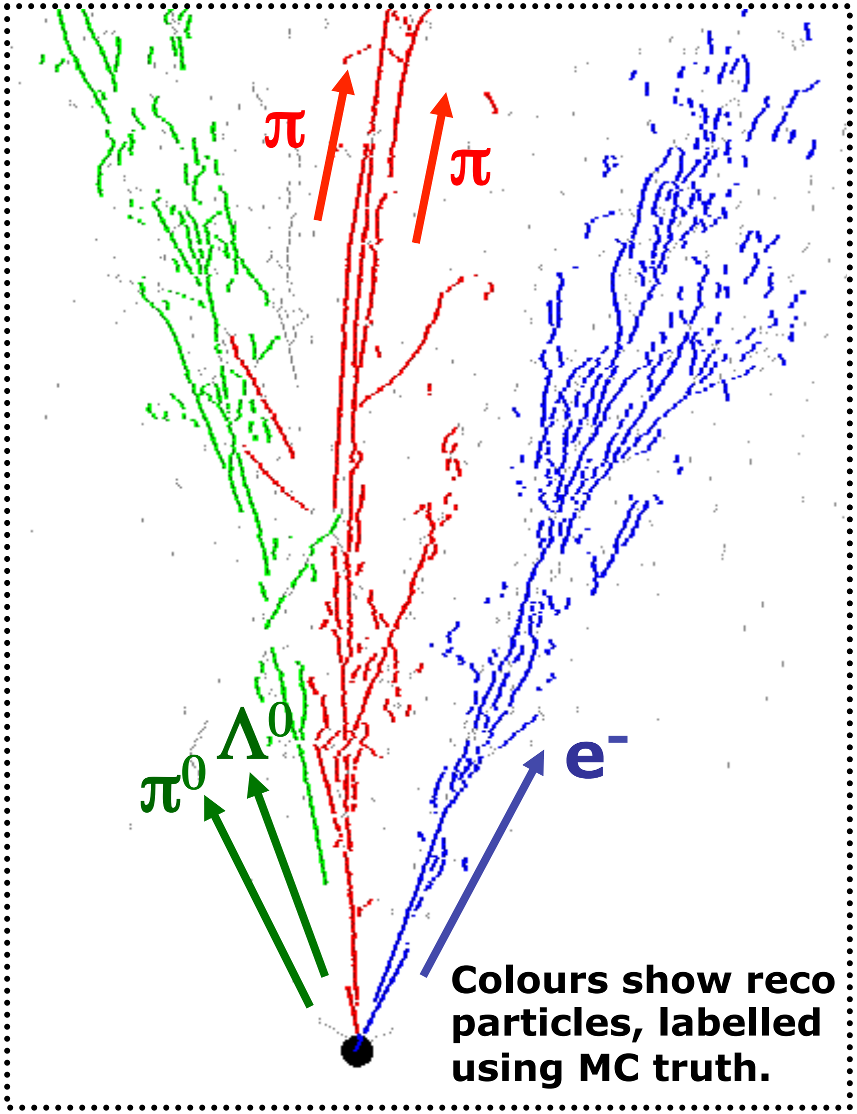


3. Vertex Reconstruction

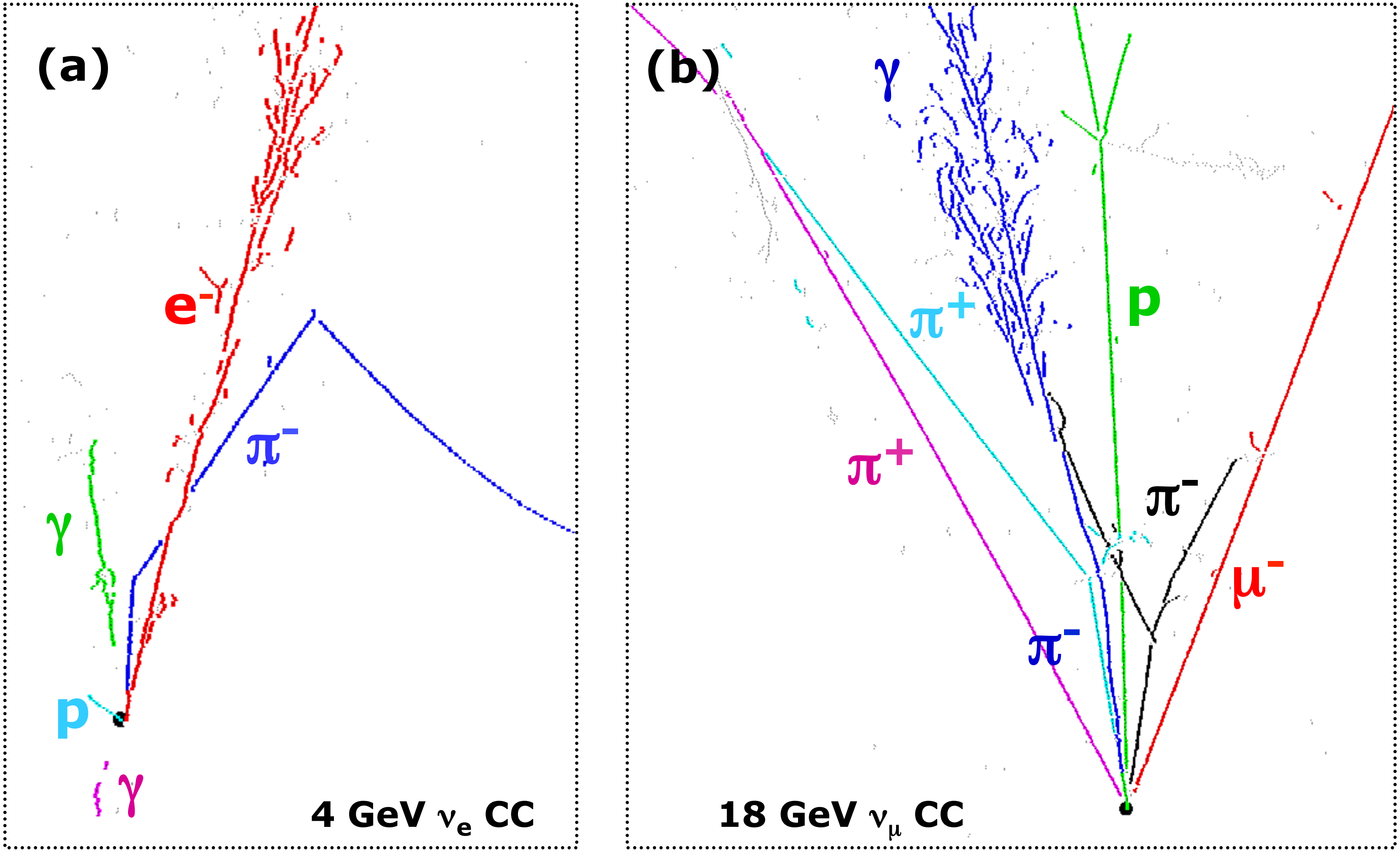


4. Final-State Particles

- **Use extended clusters, along with vertex, to reconstruct final-state particles.**
- **Method:**
  - Select 'particle seeds' based on length and proximity to vertex.
  - Use 'length-growing' algorithms to grow clusters longitudinally.
    - ◇ Gives track-like clusters.
  - Use 'branch-growing' algorithms to grow showers transversely.
    - ◇ Gives shower-like clusters.
- **Results:**
  - Next slides show some illustrative examples.

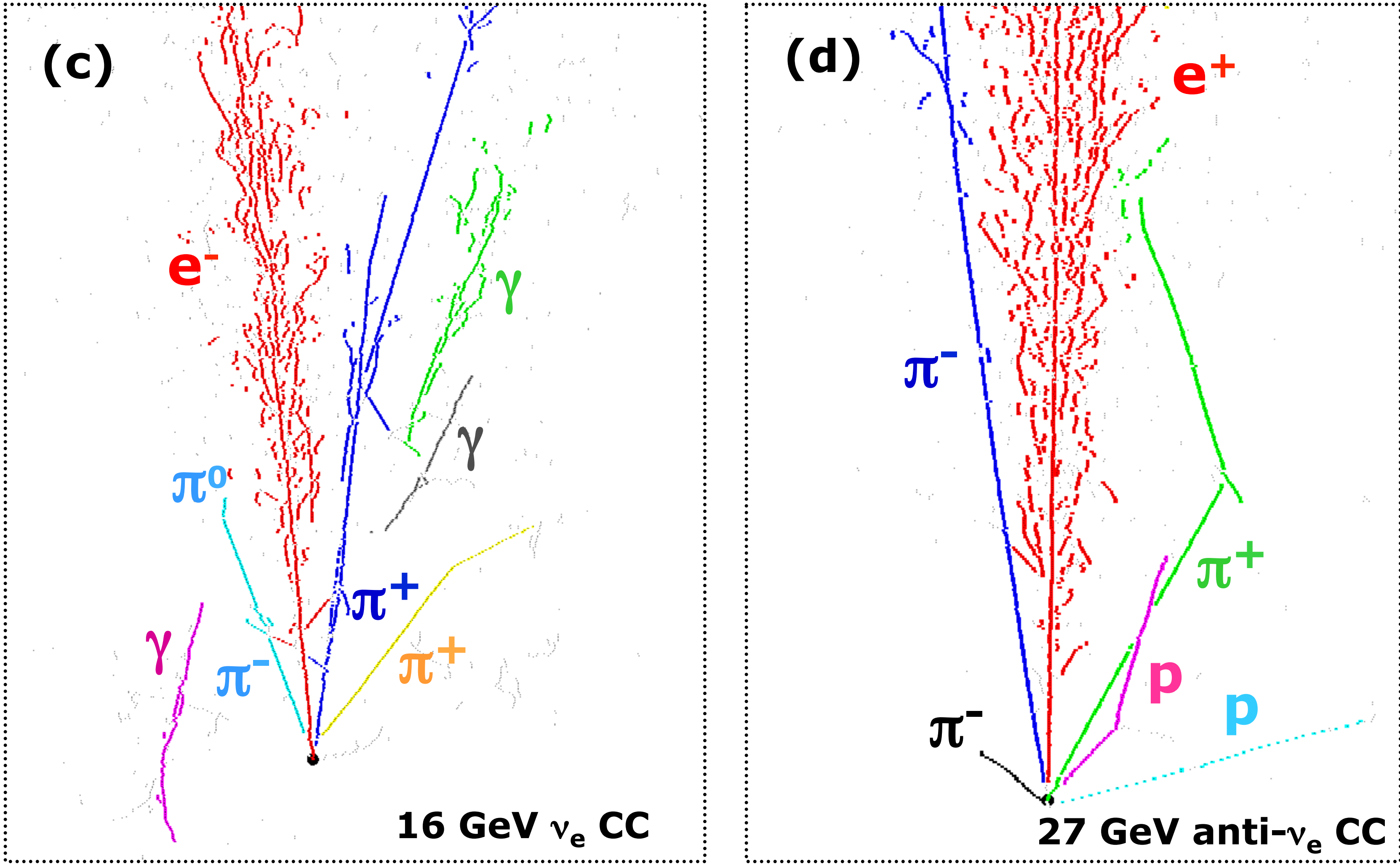


4. Final-State Particles



Coloured lines show reconstructed particles (labelled using MC truth)

4. Final-State Particles



Coloured lines show reconstructed particles (labelled using MC truth)

Thoughts and Future Work

- **Initial efforts at LAr reconstruction have produced promising results.**
  - Have integrated Pandora into LArSoft software framework.
  - A complete chain of 2D pattern recognition algorithms exists, yielding an interaction vertex and 2D tracks and showers.
  - Able to handle quite complex multi-particle final states
- **We're now starting to make the transition from 2D to 3D.**
  - Crucial to compare views as early and as much as possible
  - For some algorithms, can merge all three views straight away.
    e.g. Convert vertex reconstruction into a 3D algorithm.
  - For higher level algorithms, run an initial 2D reconstruction, then use 3D information to improve each view.
    e.g. Identify common features between views.
    Match up tracks and showers.
    Seed new tracks and showers.

Thoughts and Future Work

- **Have achieved a lot over the past few months, but don't yet have a finished product**
  - Needs further development: both the existing 2D algorithms, and the development of a full 3D combination.

- **Believe we will soon be at the point where we have a 'proof of principle' that Pandora can provide automated pattern recognition of LAr events.**

BACKUP

Reconstruction Algorithms

1. **ClusterCreation**
2. **ClusterAssociation**
3. **ClusterExtension**
4. **KinkSplitting**
5. **VertexFinding**
6. **VertexParticleFinding**
7. **ParticleLengthGrowing**
8. **ParticleFinding**
9. **ParticleBranchGrowing**
10. **ParticleConsolidation**
11. **ParticleRelegation**
12. **ParallelMerging**
13. **BoundedBoxMerging**
14. **ConeBasedMerging**
15. **RemnantClustering**
16. **RemnantConsolidation**
17. **IsolatedHitMerging**
18. **2DParticleCreation**

1. Cluster Creation
Generate initial set of proto-clusters by joining groups of contiguous hits

2. Cluster Association
Extend clusters longitudinally (tight cuts)

3. Cluster Extension
Extend clusters longitudinal (loose cuts)

4. Kink Splitting
Break any significant angular deviations in cluster trajectories.

5. Vertex Finding
Reconstruct vertex position.

6. Vertex Particle Finding
Use vertex to seed candidate particles

7. Particle Length Growing
Grow particles outwards from vertex.

Reconstruction Algorithms

1. ClusterCreation
2. ClusterAssociation
3. ClusterExtension
4. KinkSplitting
5. VertexFinding
6. VertexParticleFinding
7. ParticleLengthGrowing
8. ParticleFinding
9. ParticleBranchGrowing
10. ParticleConsolidation
11. ParticleRelegation
12. ParallelMerging
13. BoundedBoxMerging
14. ConeBasedMerging
15. RemnantClustering
16. RemnantConsolidation
17. IsolatedHitMerging
18. 2DParticleCreation

8. Particle Finding

Use any remaining long clusters as seeds for further candidate particles.

9. Particle Branch Growing

Grow particles, building tracks and showers.

10. Particle Consolidation

Merge together showers that are actually part of the same final-state particle

11. Particle Relegation

Apply quality cuts to candidate particles.

12. Parallel Merging

Add in connected or neighbouring clusters to candidate particles.

13. Bounded Box Merging

Build a box around each particle and add in any enclosed clusters.

Reconstruction Algorithms

1. ClusterCreation
2. ClusterAssociation
3. ClusterExtension
4. KinkSplitting
5. VertexFinding
6. VertexParticleFinding
7. ParticleLengthGrowing
8. ParticleFinding
9. ParticleBranchGrowing
10. ParticleConsolidation
11. ParticleRelegation
12. ParallelMerging
13. BoundedBoxMerging
14. ConeBasedMerging
15. RemnantClustering
16. RemnantConsolidation
17. IsolatedHitMerging
18. 2DParticleCreation

14. <u>Cone Based Merging</u>

Build a cone around each particle and add in clusters enclosed in the cone, or emitted downstream of the cone.

15. <u>Remnant Clustering</u>

Re-cluster those clusters that have not been merged into a candidate particle.

16. <u>Remnant Consolidation</u>

Either use the remnant clusters to build new particles, or merge them into the existing particles.

17. <u>Isolated Hit Merging</u>

Add in remaining isolated hits or small clusters to the nearest particle.

18. <u>2D Particle Creation</u>

Register the 2D particles and reconstruct their properties.

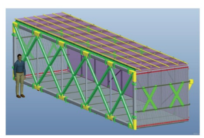
Pandora

Non-reusable e.g. detector specific

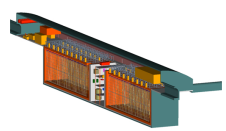
LArSoft

Producer

Isolates specific detector and software details, creating self-describing hits, tracks, etc.



MicroBooNE



LBNE

Reusable, applicable to multiple detectors

Pandora Content Libraries

LAr Content

Register, via APIs

Runs registered content and performs book-keeping

Algorithm Manager

CaloHit Manager

Cluster Manager

Plugin Manager, etc.

Pandora content:
algorithms, particle id functions, energy correction functions, shower profile calculators, etc...