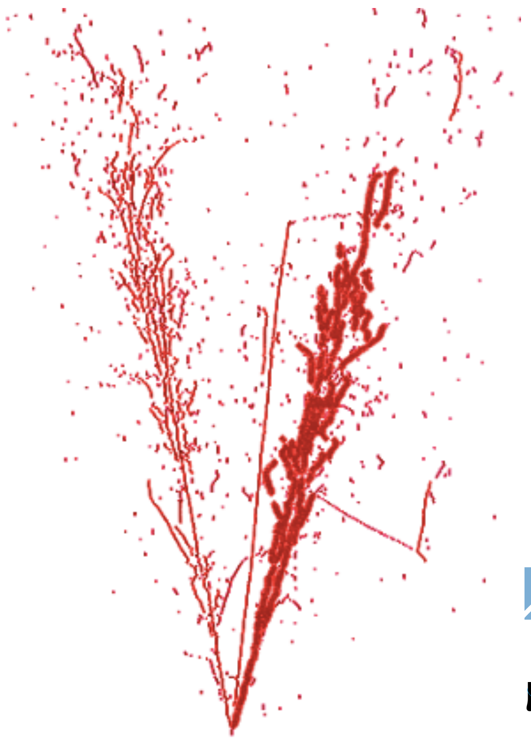
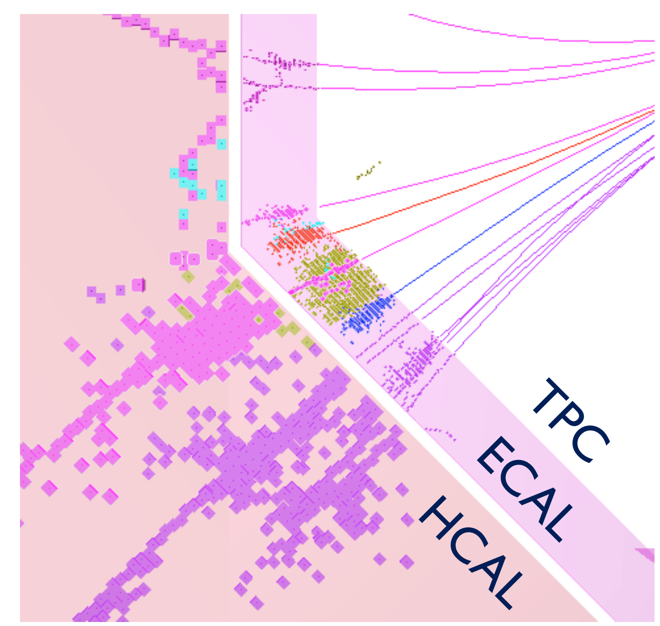


Pandora: Recent Development Highlights



J. S. Marshall, M. A. Thomson

27 May 2015

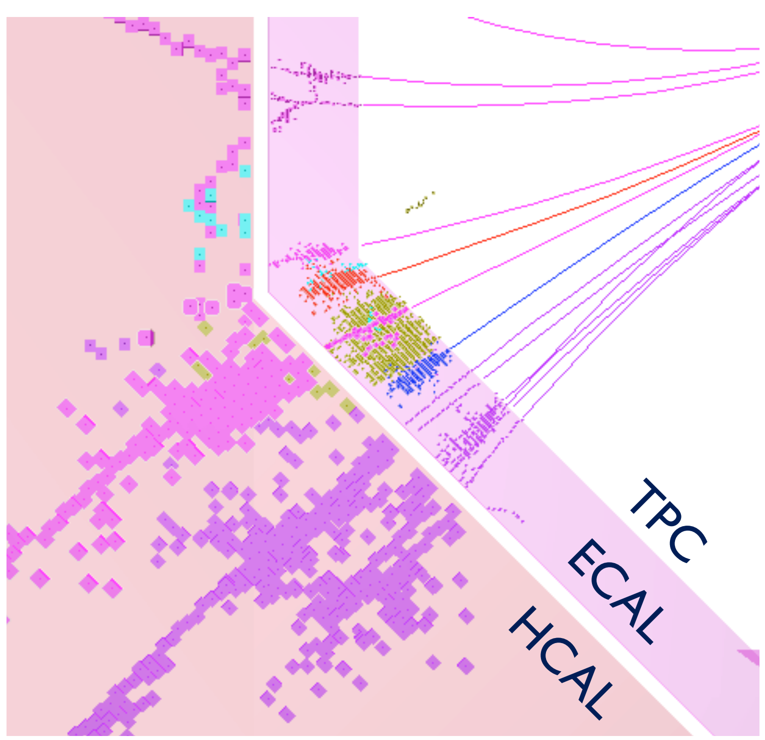
Pandora Approach

Pandora brings two key elements to the reconstruction of events in fine granularity detectors:

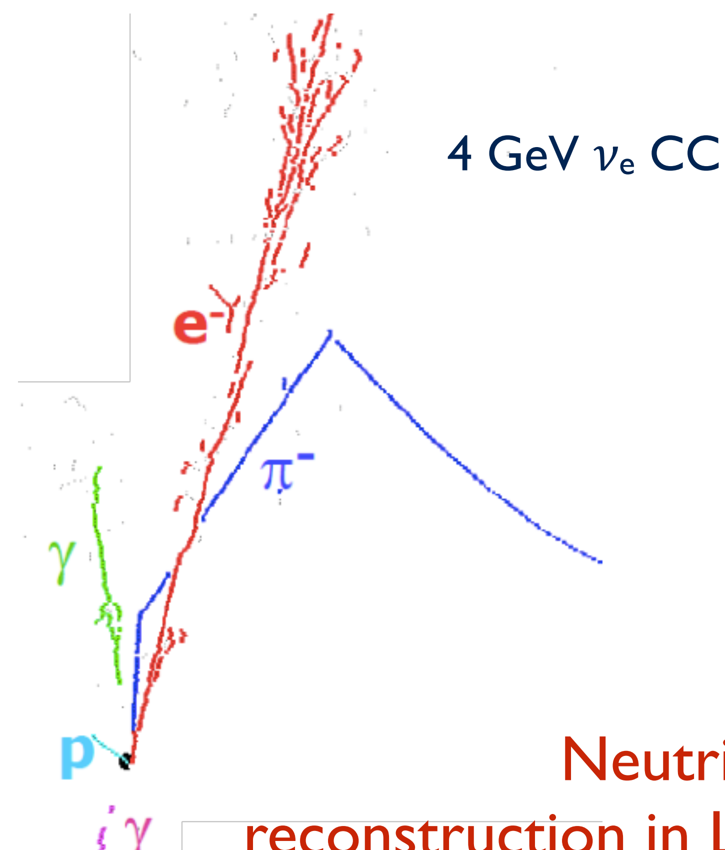
1. A reconstruction philosophy: "it's easier to put clusters together, than to split them up again". Large numbers of independent algorithms address specific event topologies, without mistakes.

2. A sophisticated software framework for developing pattern-recognition algorithms. All core memory management is performed by the framework, keeping algorithms simple and efficient.

Use same core software, with different algorithm logic, for two use-cases in HEP reconstruction:



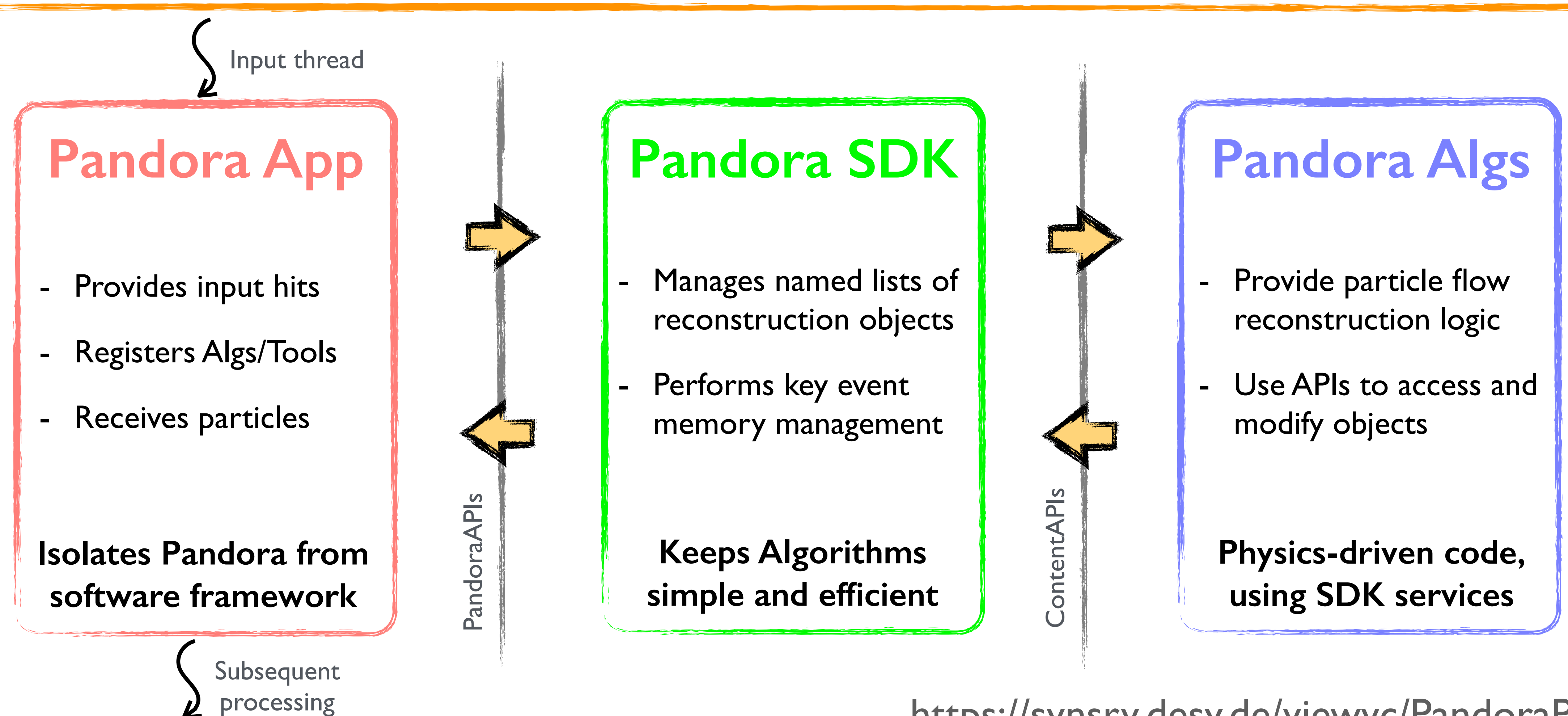
Particle flow calorimetry at a e^+e^- collider



Neutrino event reconstruction in LAr TPCs

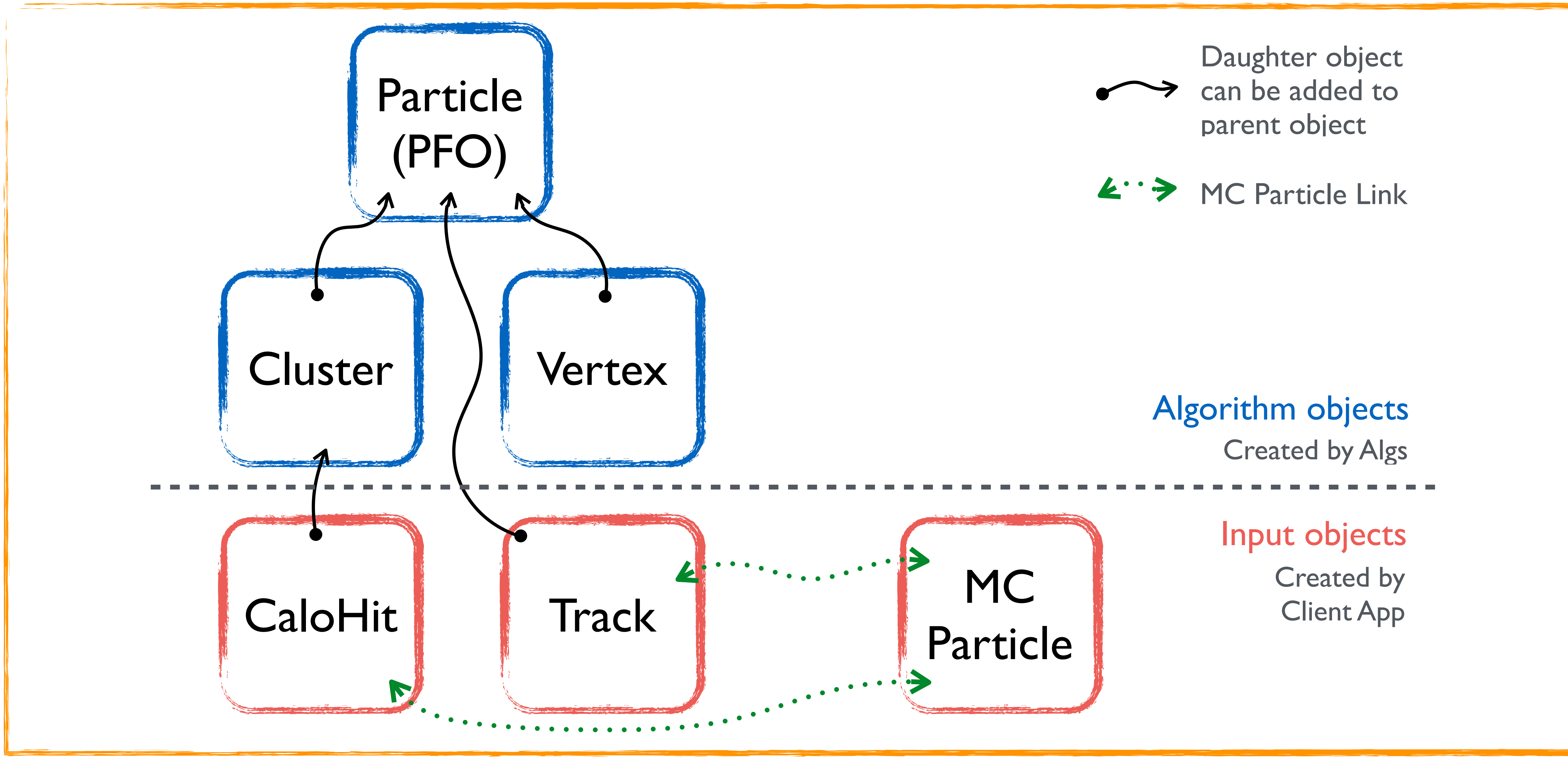
Pandora SDK

- A Pandora Client App passes details of each event (building-blocks for pattern-recognition) to Pandora, which creates and manages its own self-describing reconstruction objects.
- Reconstruction logic is provided by large numbers of Pandora Algorithms. Each Algorithm tries to address a particular event topology, making pattern-recognition changes without mistakes.
- Algorithms can only perform non-const operations (e.g. create/split/merge Clusters) by asking Pandora to provide the service. Pandora then performs memory-management and book-keeping.



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

Pandora Event Data Model



- EDM is flexible and re-usable, but sufficiently fully featured to make Pandora an excellent development environment. Pandora Monitoring allows easy pop-up of 3D event displays in Algs.
- Pandora persistency (can write Pandora self-describing objects to binary or xml files) allows for rapid development in a standalone Pandora environment; only need to run Client App once!

Pandora SDK Paper

To be submitted in near future

The Pandora Software Development Kit for Pattern Recognition

J. S. Marshall*, M. A. Thomson

Cavendish Laboratory, University of Cambridge, Cambridge, United Kingdom

Abstract

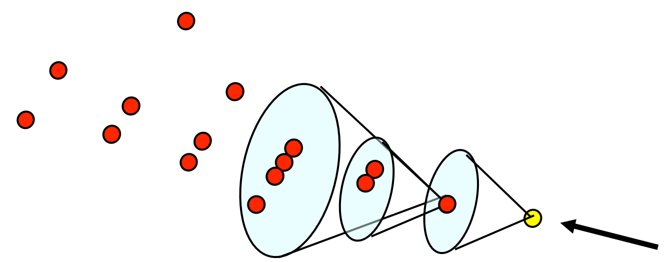
The development of automated solutions to pattern recognition problems is important in many areas of scientific research and human endeavour. This paper describes the implementation of the Pandora Software Development Kit, which aids the process of designing, implementing and running pattern recognition algorithms. The Pandora Application Programming Interfaces ensure simple specification of the building-blocks defining a pattern recognition problem. The logic required to solve the problem is implemented in algorithms, with all operations to create or modify event data structures requested by algorithms and performed by the Pandora framework. This design promotes an approach using many decoupled algorithms, each addressing specific topologies. Details of algorithms addressing two pattern recognition problems in High Energy Physics are presented: reconstruction of events at a high-energy e^+e^- linear collider and reconstruction of cosmic ray or neutrino events in a liquid argon time projection chamber.

Keywords: Software Development Kit, Pattern recognition, High Energy Physics

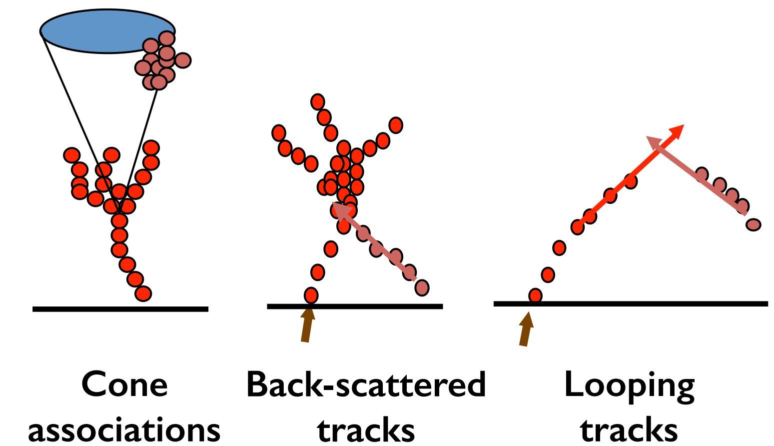
Pandora LC Algorithms

"Basic" LC reconstruction uses 56 algorithms:

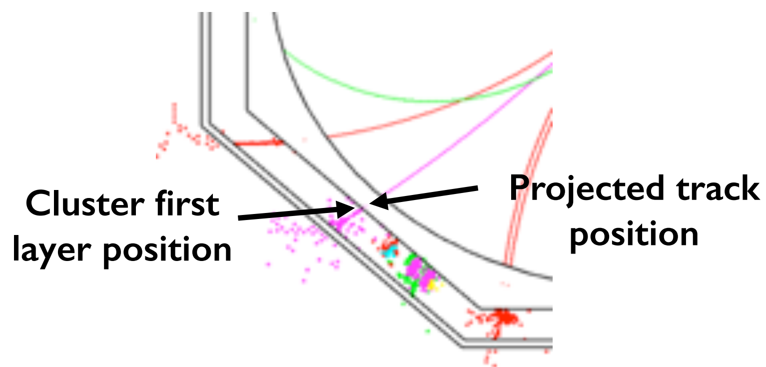
ConeClustering Algorithm



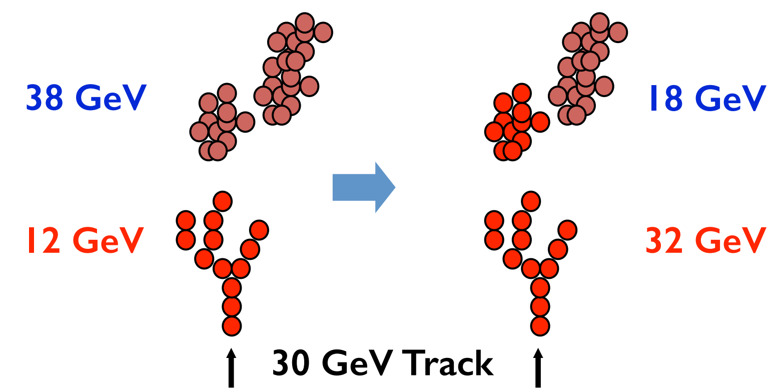
Topological Association Algorithms



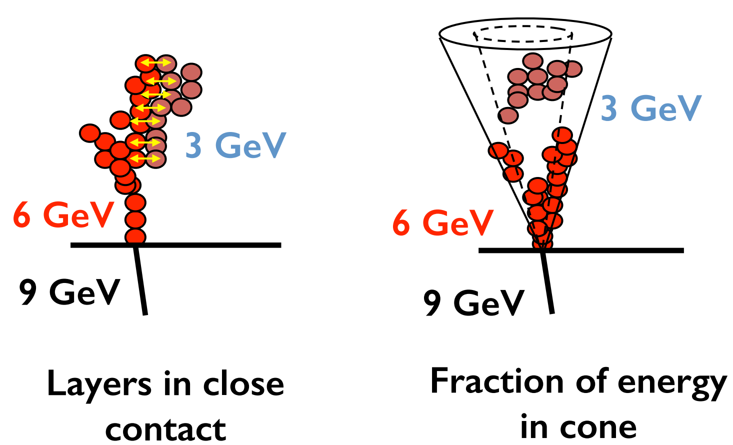
Track-Cluster Association Algorithms



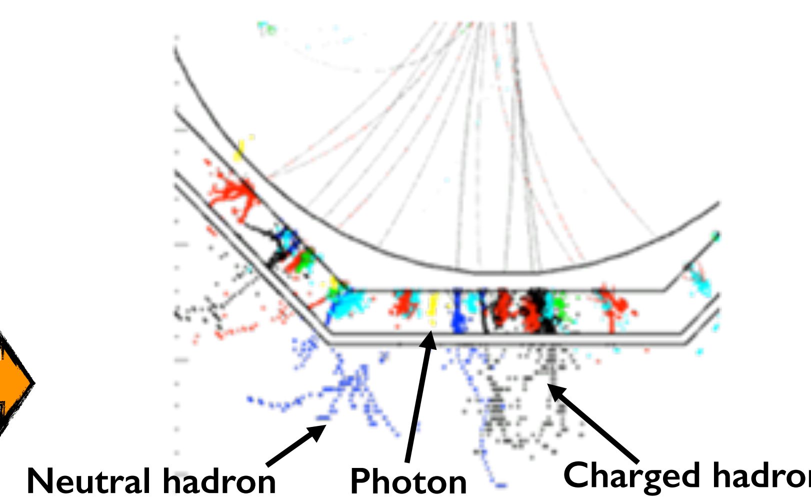
Reclustering Algorithms



Fragment Removal Algorithms

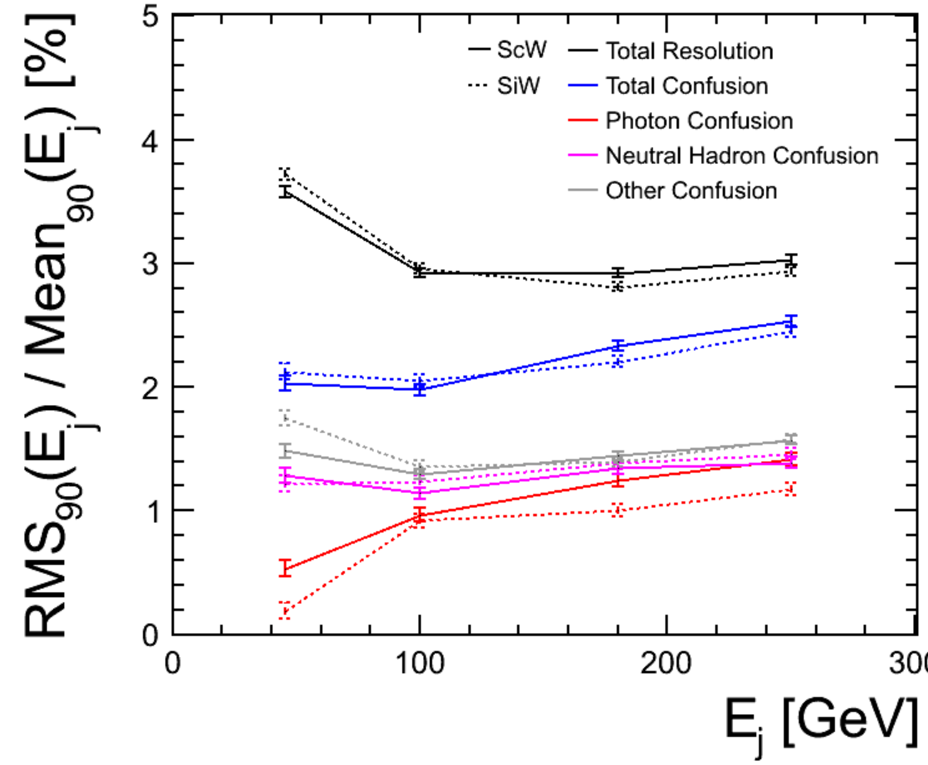


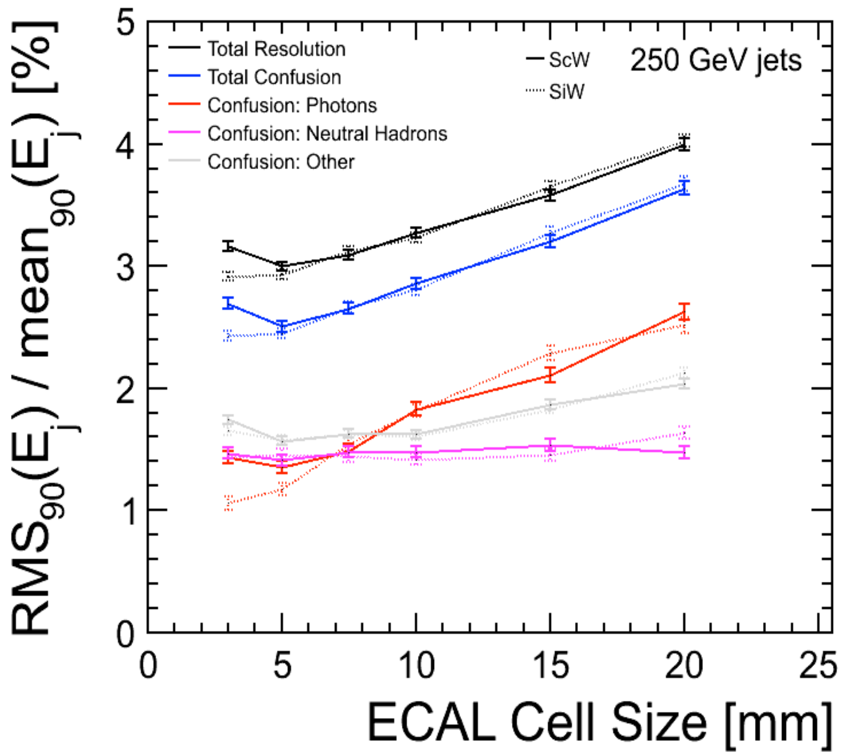
PFO Construction Algorithms

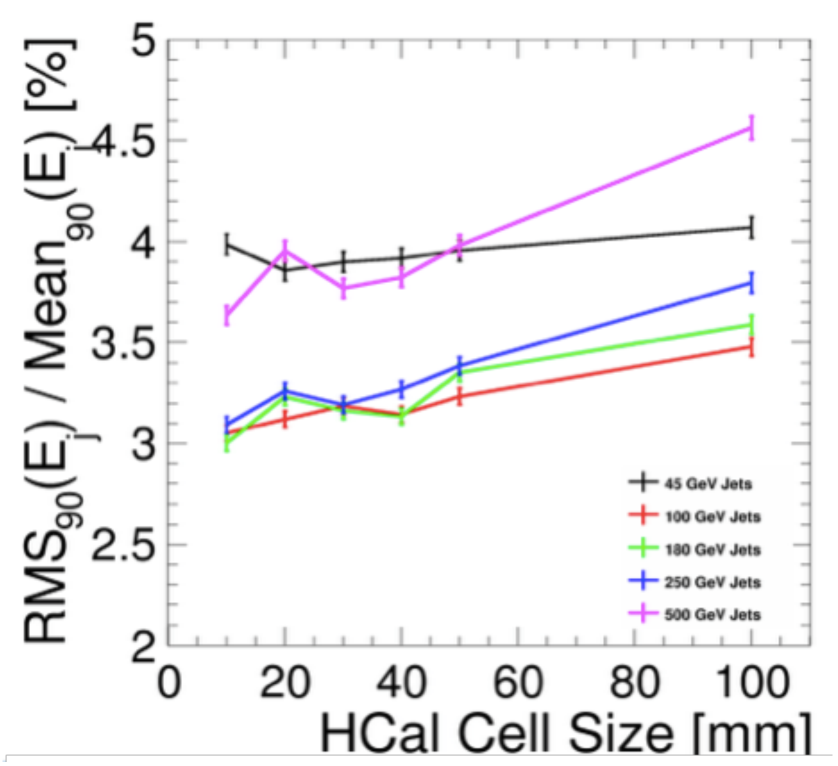


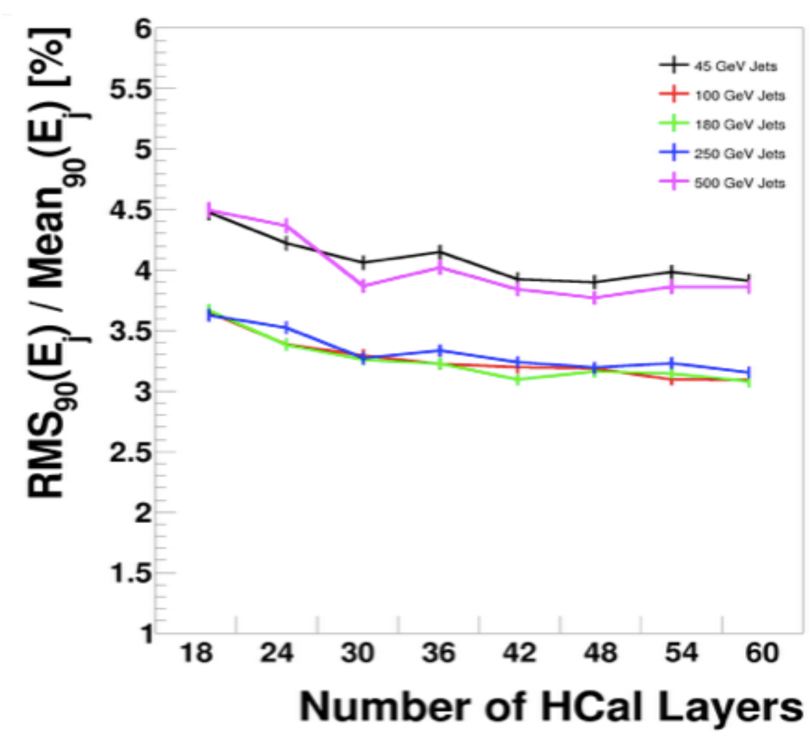
Pandora Detector Optimisation

- Physics motivation for particle flow calorimetry:
  Jet energy resolution: $\sigma_E/E < 3.5\%$
- Benchmark performance using jet energy resolution in Z decays to light quarks.
- Use jet energy resolution as figure of merit for extensive detector optimisation studies.
- **Publication currently under construction.**









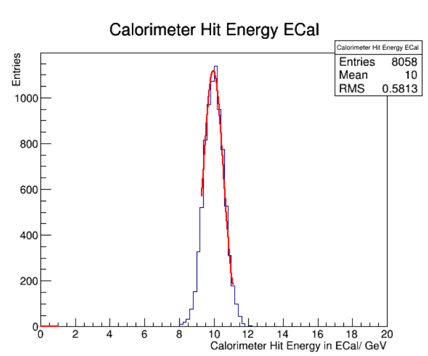
S. Green.

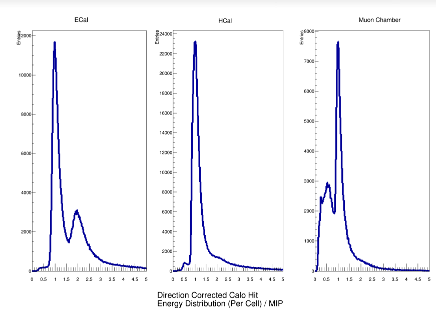
Simulation Output
Energy Deposited in Active Material of Calorimeters

Calorimeter Energy
Energy Deposited in the Total Calorimeter Volume

Reconstructed Particle Energy
Energy of Particle(s) Depositing Energy in Calorimeters

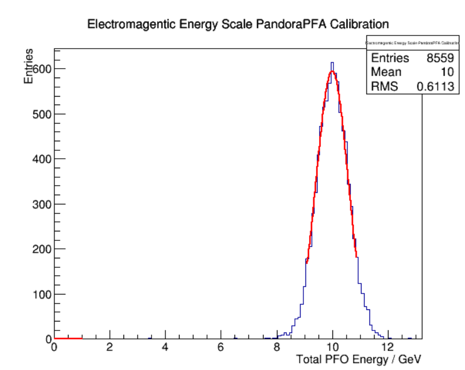
Digitisation
Set by looking at contained kaonL and γ events throughout the calorimeters.

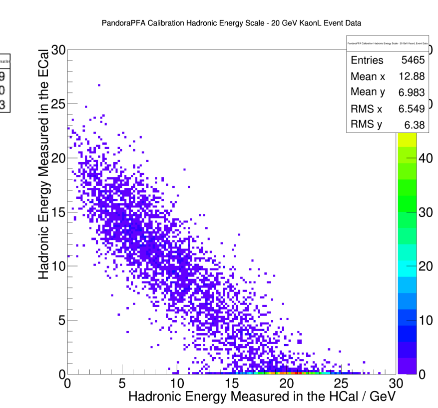




Minimum Ionising Particle (MIP) Detector Response
Determine the response of each part of the detector to a MIP, μ^- events. MIP scale used in PandoraPFA.

Electromagnetic and Hadronic Scale Setting
Electromagnetic/Hadronic scale set using PFO energy of contained γ and kaonL events.

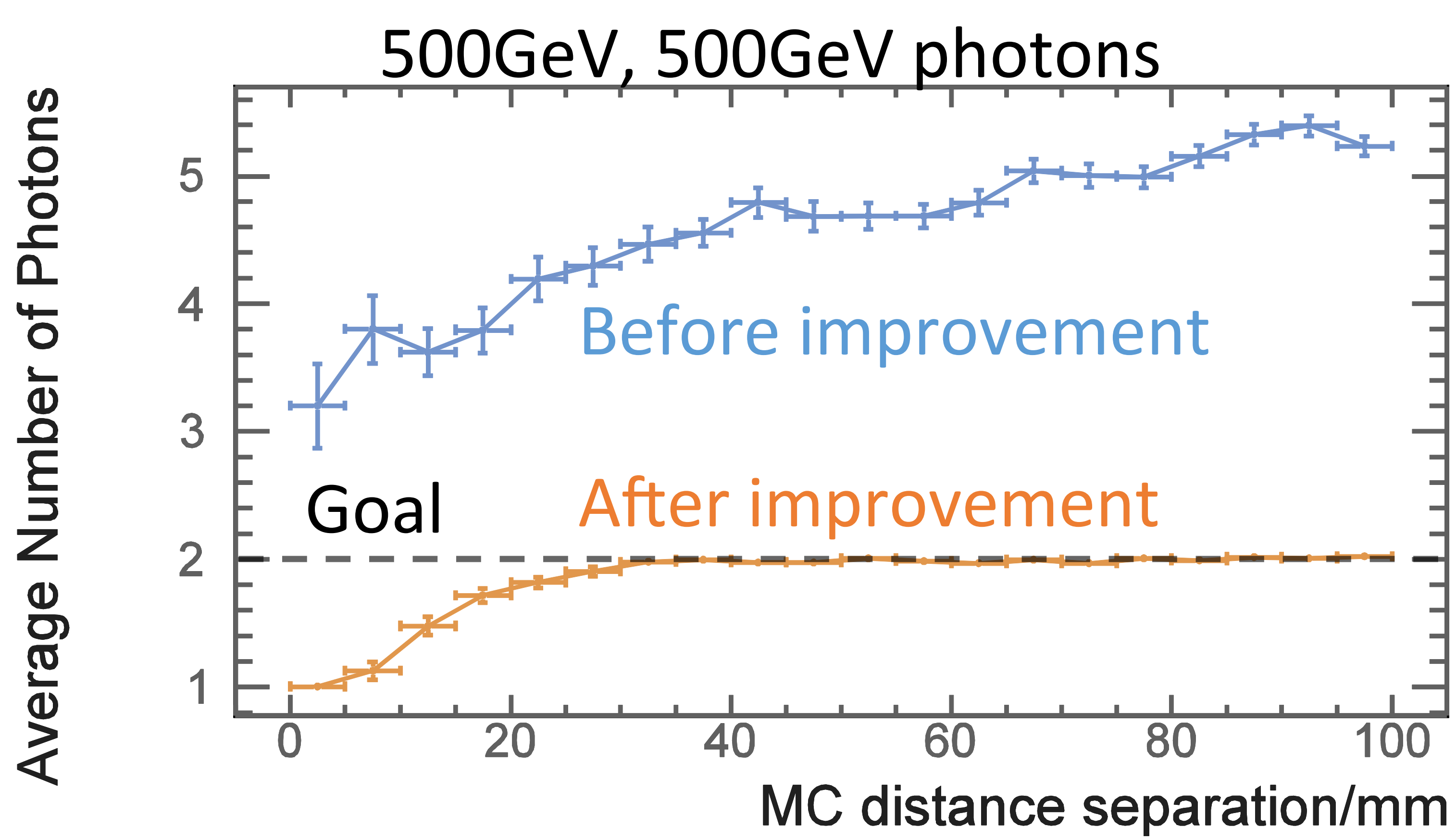


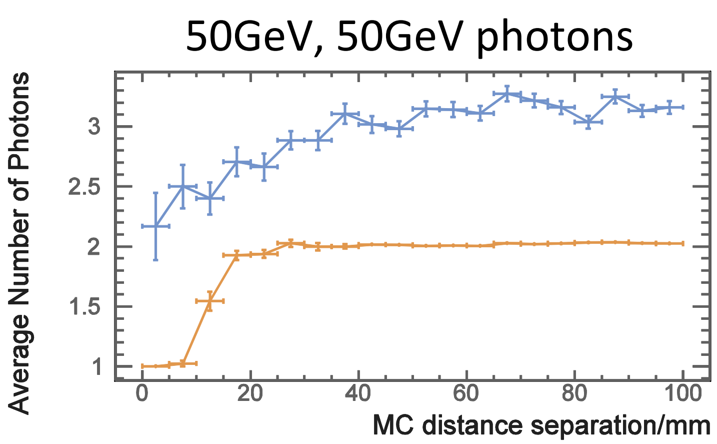


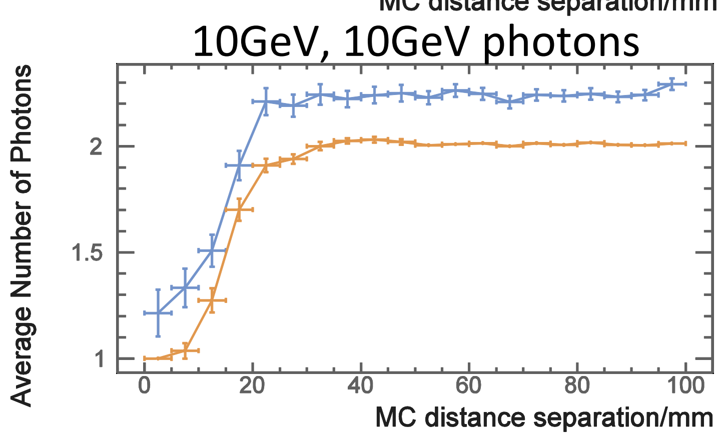
The **PandoraAnalysis** toolkit has scripts designed for setting the digitisation and calibration constants. The user has to provide samples of kaonL, γ and μ^-.
These scripts make automated calibration possible. Procedure is fully documented.

B. Xu.

- Improve completeness of reconstructed photons, particularly at high energies, where small fragments of EM showers could often be reconstructed as separate particles.
- Two new Pandora algorithms carefully compare candidate photon clusters, collecting evidence of association, based on cluster separation and energy profiles.
- Performance plots below show average number of reconstructed photons (as a function of true separation) for samples consisting of two photons, generated with random directions.

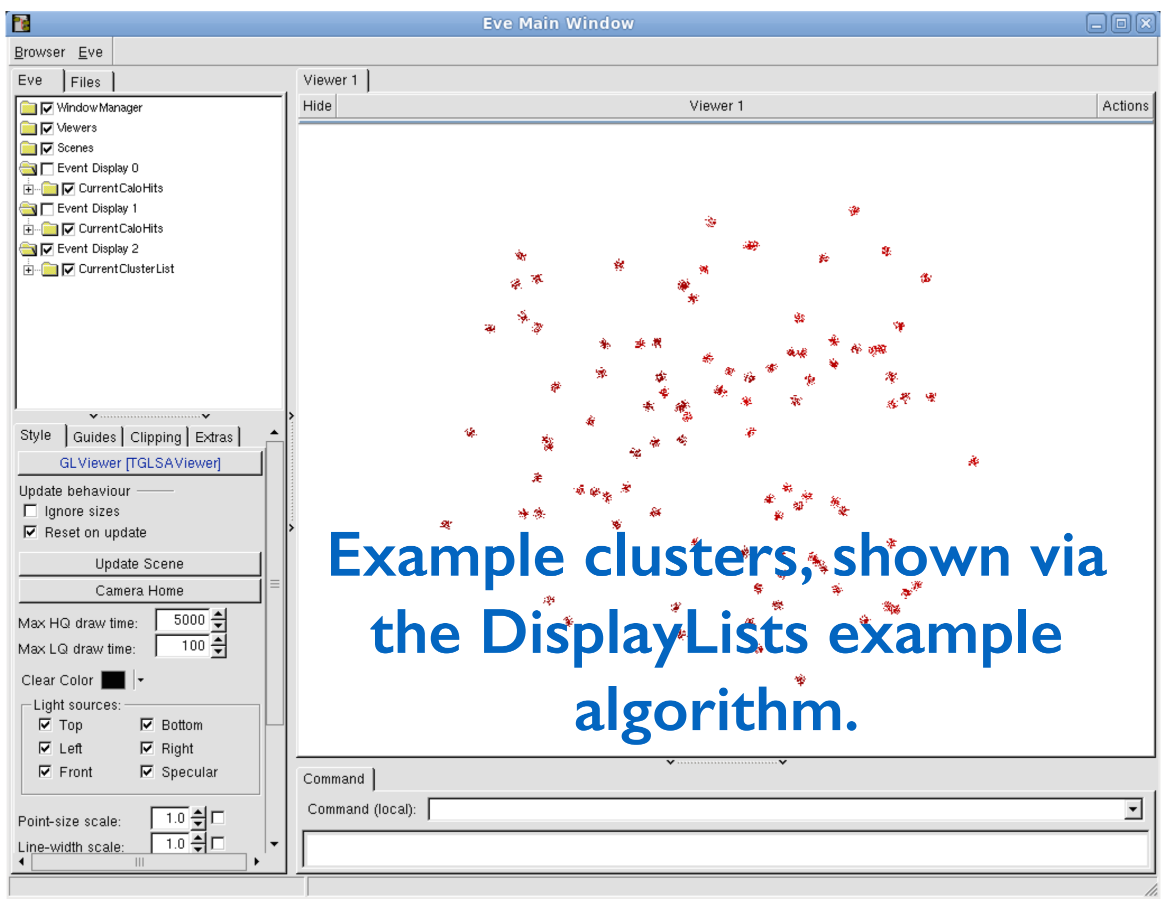






Pandora Learning Library

- Pandora algorithms create and/or modify clusters, vertices and PFOs. Their decisions (the algorithm logic) whether to proceed with operations can be complex and use-case specific.
- The aim of the Pandora ExampleContent library and test application is to demonstrate the key Pandora functionality in a very simple testing and learning environment.
- The ExampleContent library is structured in exactly the same manner as the LCContent and LArContent libraries, currently in use for Linear Collider and LAr TPC reconstruction.



Example clusters, shown via the DisplayLists example algorithm.

- The library consists of example Algorithms, AlgorithmTools, Plugins and Helper functions:
  - Example list access and display
  - Example Cluster, Vertex and PFO creation
  - Cluster manipulation, including merging, deletion, fragmentation and reclustering
  - Creating and saving new lists of objects
  - Using Algorithm Tools and Plugins
  - Writing a tree using PandoraMonitoring.

http://www.hep.phy.cam.ac.uk/~marshall/PandoraExample.pdf

Pandora EDM Object Extensions

Object creation API

Note optional PandoraObjectFactory argument

```
/**
 *  @brief  Object creation helper class
 */
template <typename PARAMETERS, typename OBJECT>
class ObjectCreationHelper
{
public:
    typedef PARAMETERS Parameters;
    typedef OBJECT Object;

    /**
     *  @brief  Create a new object from a user factory
     *
     *  @param  pandora the pandora instance to create the new object
     *  @param  parameters the object parameters
     *  @param  factory the factory that performs the object allocation
     */
    static pandora::StatusCode Create(const pandora::Pandora &pandora, const Parameters &parameters,
        const pandora::ObjectFactory<Parameters, Object> &factory = pandora::PandoraObjectFactory<Parameters, Object>());
};
```

Enhanced object

```
/**
 *  @brief  Example calo hit class - add an additional property to pandora calo hit
 */
class ExampleCaloHit : public pandora::CaloHit
{
public:
    /**
     *  @brief  Constructor
     *
     *  @param  parameters the example calo hit parameters
     */
    ExampleCaloHit(const ExampleCaloHitParameters &parameters);

    /**
     *  @brief  Get the additional property string
     *
     *  @return the additional property string
     */
    const std::string &GetAdditionalProperty() const;

private:
    std::string     m_additionalProperty;       ///< The additional property string
};
```

- Advanced users can add additional functionality to Pandora objects.
- Provide implementation of a custom PandoraObjectFactory.
- Specify non-default ObjectFactory at point of object creation.
- PandoraSDK will work with pointers to base objects.
- Algorithms can access derived object content via dynamic_cast

R. Ete.

Current status of Arbor(-like)PFA in PandoraSDK (R. Ete, IPNL)

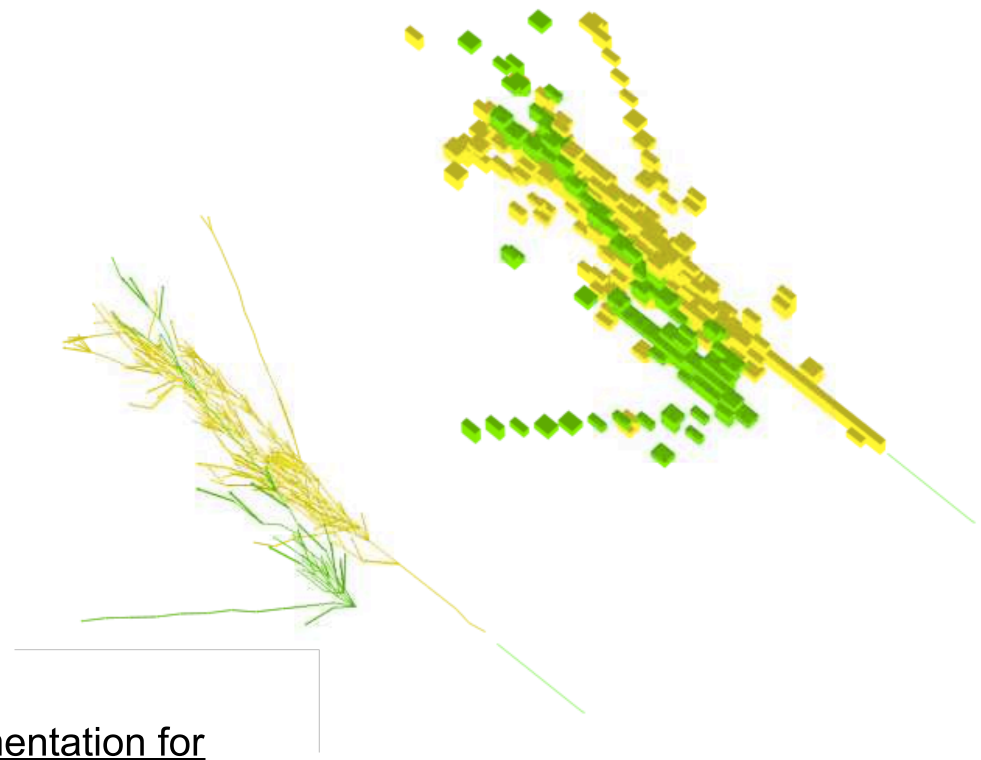
Implementation for the SDHCAL prototype

* Package ArborPFA
→ https://github.com/SDHCAL/ArborPFA

* Developed for SDHCAL test beam data

* Single particle study

* Separation of close-by hadronic showers



Implementation for full detector (i.e ILD)

* Package ArborContent (algorithms) + MarlinArbor (marlin interface)
→ https://github.com/rete/ArborContent
→ https://github.com/rete/MarlinArbor

* Uses the new PandoraSDK (v01.00.01) object extension functionality → extension of CaloHit (arbor_content ::CaloHit)

* Algorithms currently being reviewed and new ones added for full detector implementation
→ Ecal specific impl, energy estimators, Ecal-Hcal link, gap handling, reclustering, etc...

* Refactorization of algorithms with AlgorithmTools to test new ideas on key points (specific seeding, cleaning, branching, etc ...)

Pandora Multiple Instances

PANDORA

AlgorithmManager	Algorithms and AlgorithmTools
CaloHitManager, MCManager, TrackManager	Input object Managers
ClusterManager, PfoManager, VertexManager	Algorithm object Managers
GeometryManager	SubDetectors and DetectorGaps
PluginManager	Pseudo-Layer, BField, Particle Id
PandoraSettings	Instance config.
PandoraApiImpl, PandoraContentApiImpl, PandoraImpl	Hidden implementation

Marlin steering

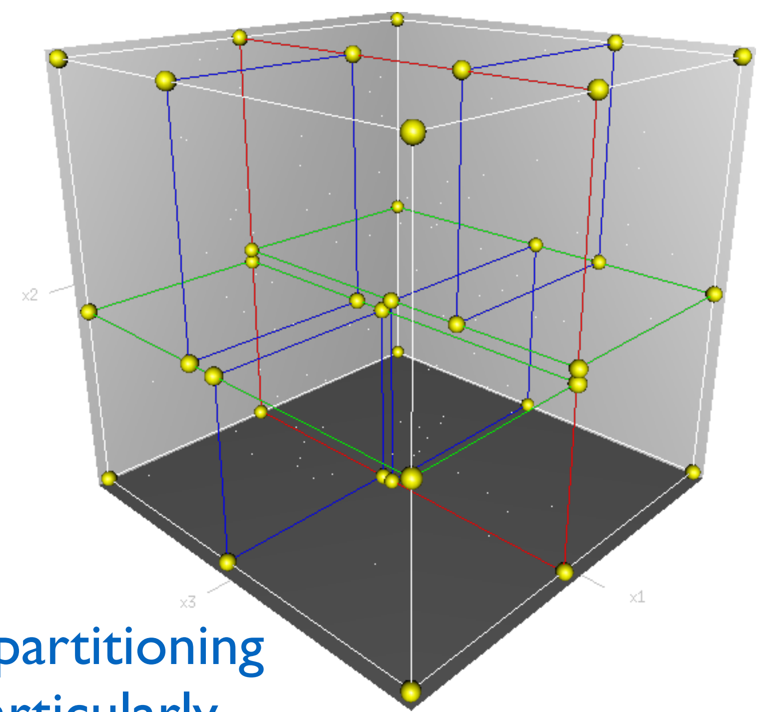
```
<execute>
  <processor name="MyMarlinPandoraMuon"/>
  <processor name="MyPfoAnalysisMuon"/>
  <processor name="MyMarlinPandoraDefault"/>
  <processor name="MyPfoAnalysisDefault"/>
  <processor name="MyMarlinPandoraPerfectPhoton"/>
  <processor name="MyPfoAnalysisPerfectPhoton"/>
  <processor name="MyMarlinPandoraPerfectPhotonNK0L"/>
  <processor name="MyPfoAnalysisPerfectPhotonNK0L"/>
  <processor name="MyMarlinPandoraPerfectPFA"/>
  <processor name="MyPfoAnalysisPerfectPFA"/>
</execute>
```

- Older Pandora releases: configurable static members, so enforce can only be one Pandora instance per system process.
- From Pandora v00-17-00, each Pandora instance stands alone and contains all it needs for a complete reconstruction.
- Pandora instance is a container of Manager instances, API Impl instances and a Settings instance.
- Algorithms and Algorithm Tools created (as requested by XML config) and owned by Algorithm Manager.
- All Pandora Process instances (includes Algorithms and AlgorithmTools) know with which Pandora they are associated.
- Can e.g. run multiple Pandora passes (with different algs) in one Marlin job.

Pandora LCContent FAST

- Recent **successful** application of Pandora LC algorithms to reconstruction of events in CMS HGCal upgrade geometry.
- Challenging environment with significant pile-up; algorithms need to deal with large numbers of hits and tracks.
- Tested and improved LC algorithm performance using events at CLIC with sizeable $\gamma\gamma\rightarrow$hadrons background.

Example 3-dimensional k-d tree:



k-d trees are space-partitioning data structures, particularly suited to range searches using multidimensional keys.

- **Significant** gains made using **k-d trees** in cone clustering, fragment removal, track-cluster and topological association algs.
- **Typical CLIC pile-up events: from 60.3 s / event to 17.2 s / event**

- k-d tree implementation uses C++11, so LCContent contains FAST versions of algorithms, only built if C++11 supported.
- To use, append "Fast" keyword to alg types in PandoraSettings XML file. Will become default once ilcsoft uses C++11 as standard.

Pandora Tags

- **Recommended tags for ilcsoft:**
  - PandoraPFA v01-02-00: ⟵ Metadata/build package, collects consistent tags
    - Pandora SDK v01-00-01 ⟵ Pandora framework
    - Pandora Monitoring v01-00-01 ⟵ ROOT EVE visualisation, plus tree writing
    - Pandora LCContent v01-01-00 ⟵ Algorithms for use at ILC or CLIC
  - MarlinPandora v01-00-01 ⟵ Client App in Marlin framework
  - PandoraAnalysis v01-00-00 ⟵ Marlin processors for analysis and calibration
- These tags include all developments from previous slides, except ability to extend Pandora object; functionality currently only available in trunk (will tag new version ASAP).

- **Pandora:** a reusable reconstruction philosophy and a reusable and reliable software framework.
- Using different algorithm logic, address pattern-recognition problems in multiple areas of HEP.

Backup

Pandora LC Event Reconstruction

Traditional calorimetric approach:

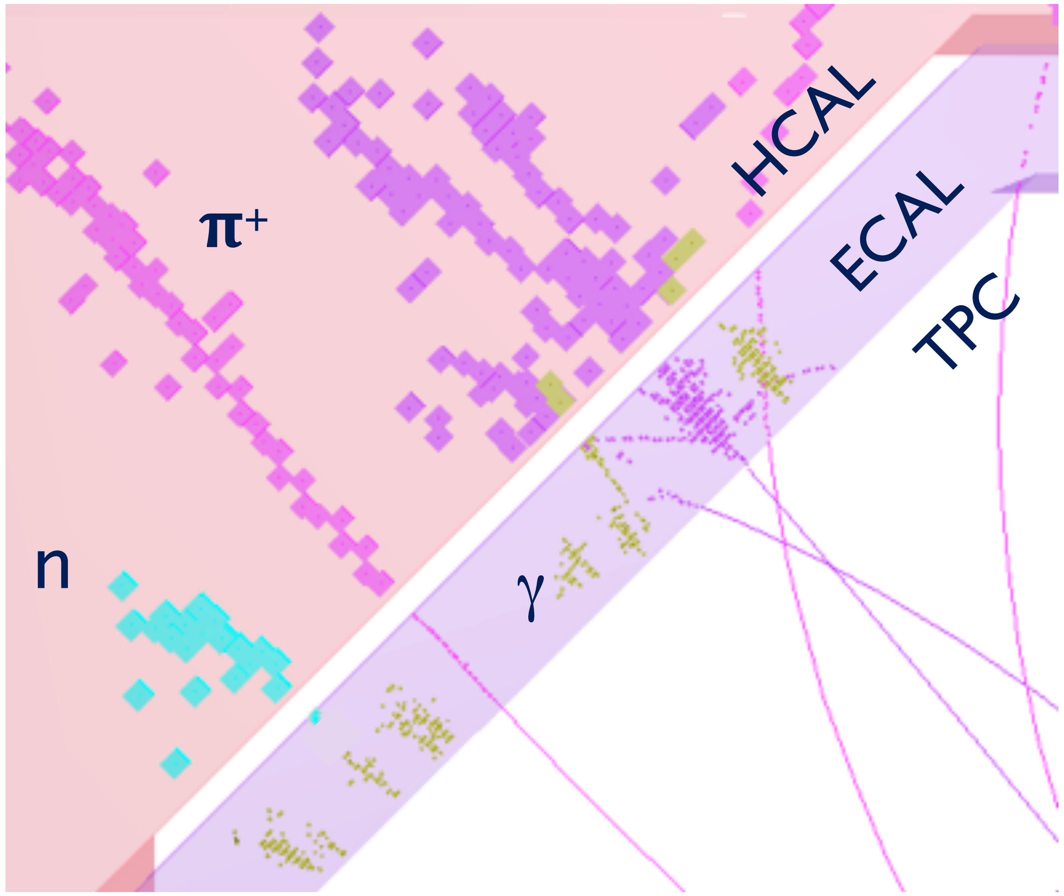
- Measure all components of jet energy in ECAL/HCAL
- Approximately 70% of energy measured in HCAL: $\sigma_E/E \approx 60\% / \sqrt{E(\mathrm{GeV})}$

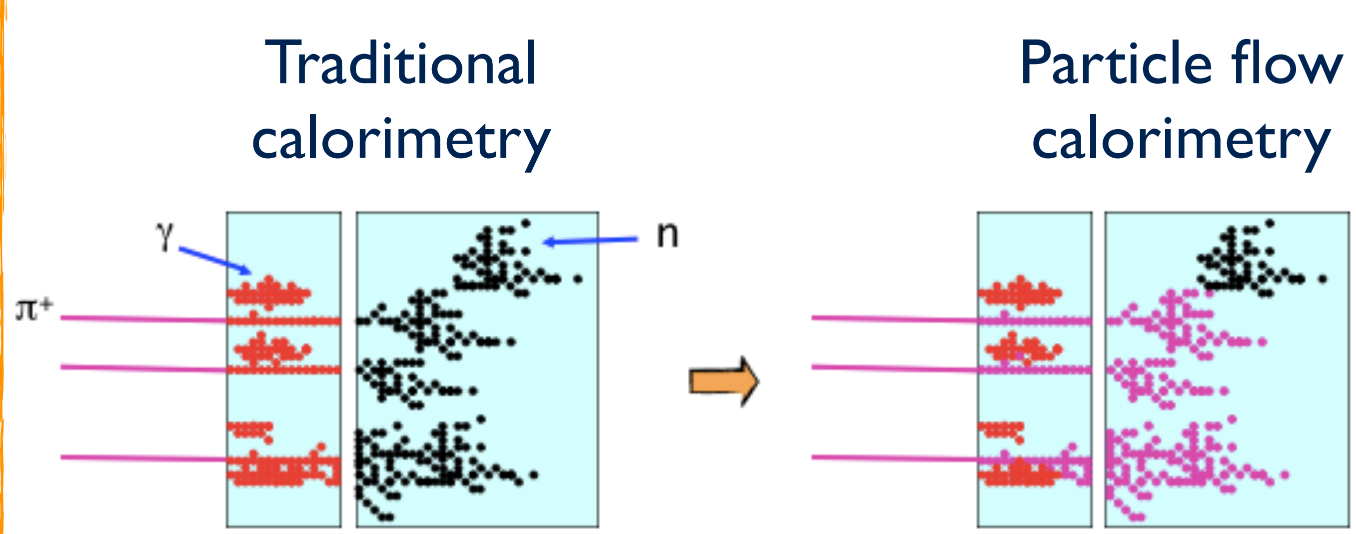
Particle Flow Calorimetry:

- Trace paths of individual particles through the detector.
- Charged particle momentum measured in tracker (essentially perfectly)
- Photon energies measured in ECAL: $\sigma_E/E < 20\% / \sqrt{E(\mathrm{GeV})}$
- Only neutral hadron energies (10% of jet energy) measured in HCAL.

Particle Flow Calorimetry requires:

- Fine-granularity calorimeters
- Sophisticated software algorithms





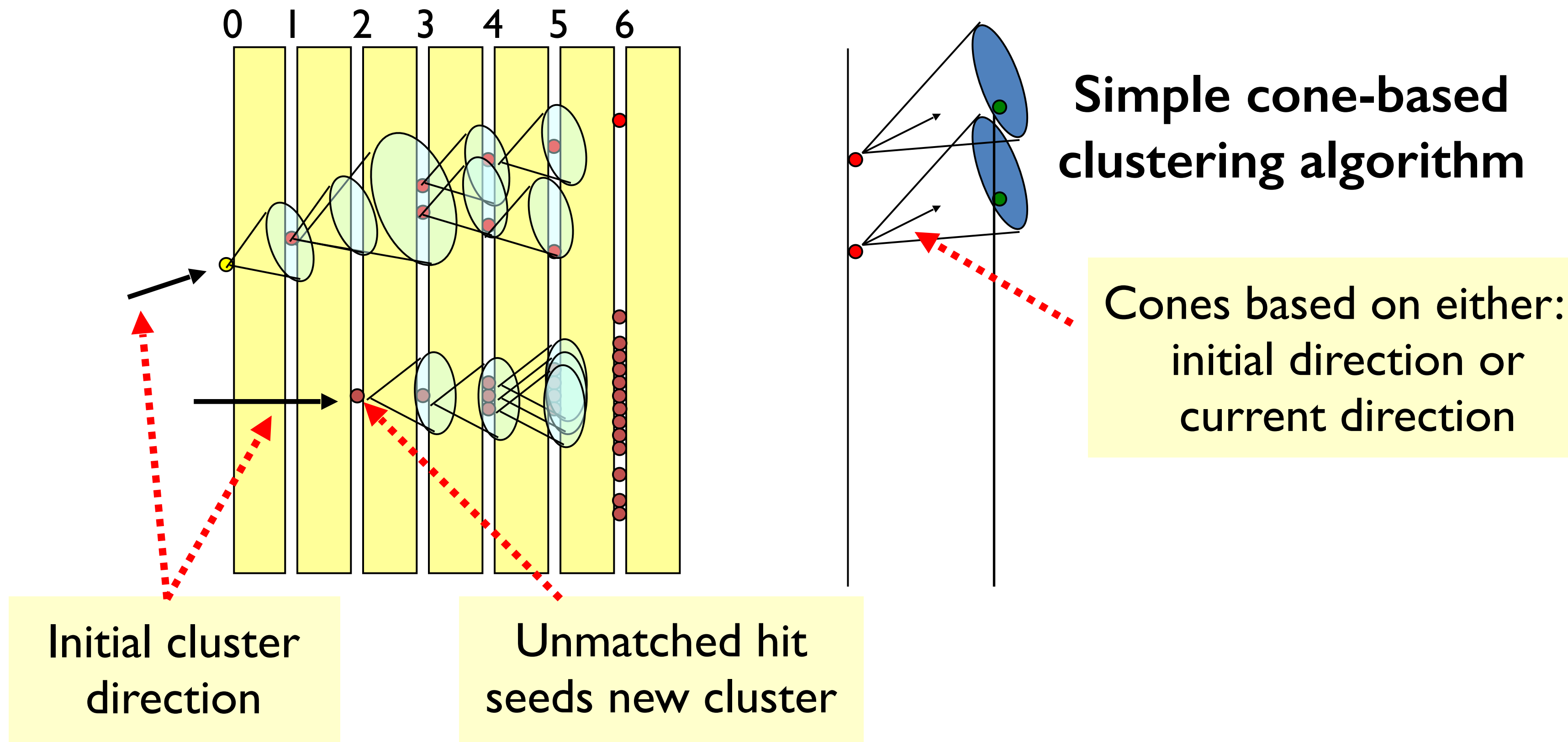
$E_{JET} = E_{ECAL} + E_{HCAL}$

$E_{JET} = E_{TRACK} + E_{\gamma} + E_{n}$

Pandora LC Clustering

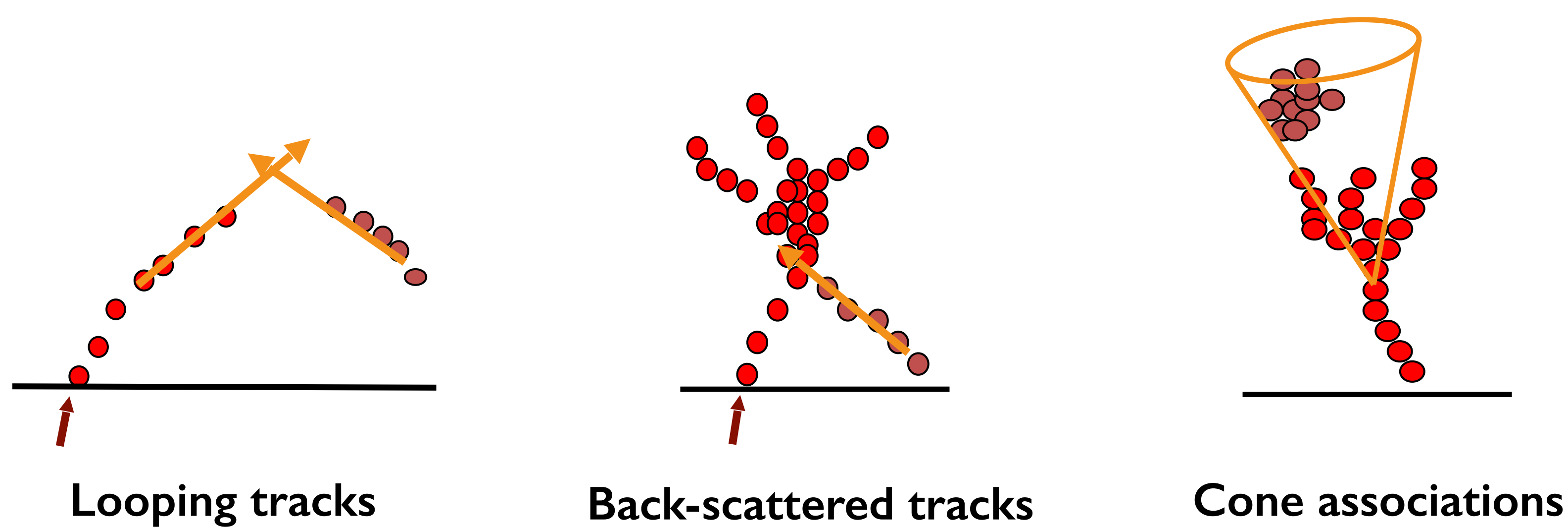
- **Philosophy:** "It's easier to put clusters together, than to split them up again."

- Clustering algorithm very careful to avoid accidentally merging energy deposits from separate particles.



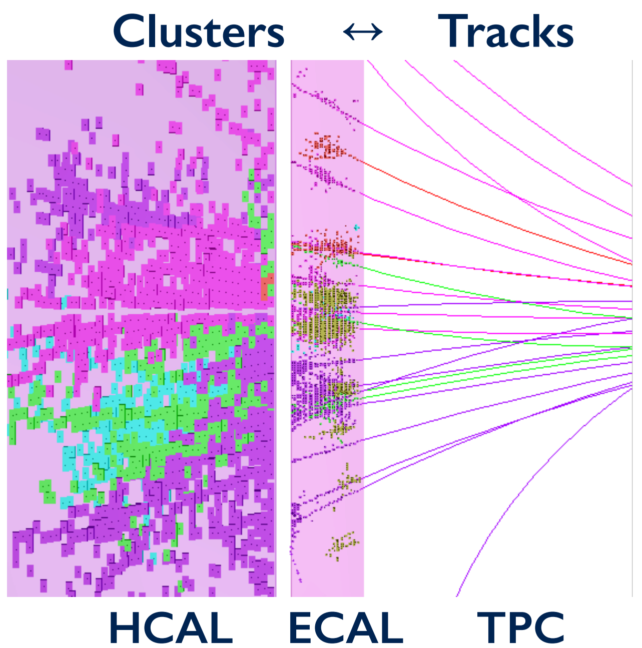
- Fine granularity of the calorimeters exploited to merge cluster fragments that are clearly associated.

- Very few mistakes made.

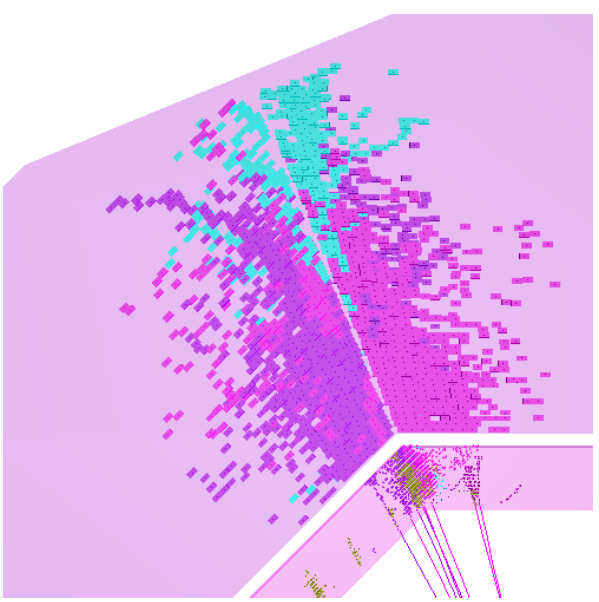


Pandora LC Reclustering

- Key aspect of particle flow calorimetry is association of calorimeter clusters to inner detector tracks.
- Look for consistency between cluster properties and helix-projected track state at front face of calorimeter:
  - Close proximity between cluster and track positions.
  - Consistent track and initial cluster directions.

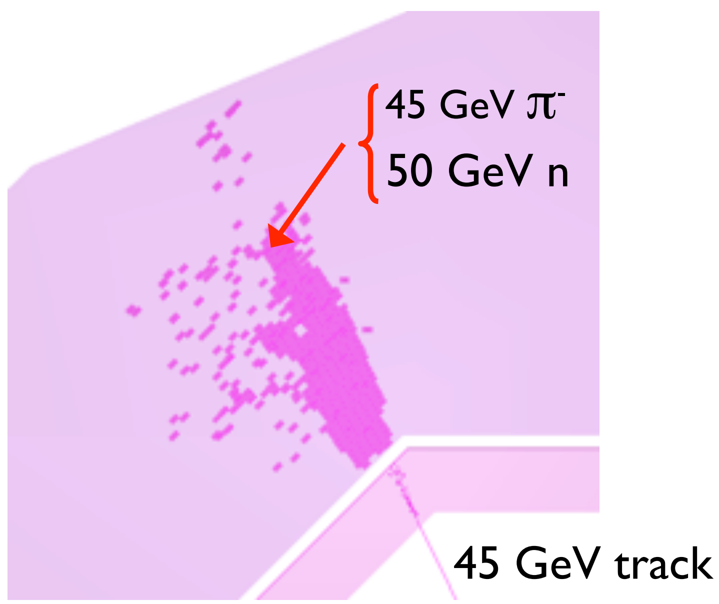


- At some point, in high energy jets, cannot cleanly resolve neutral hadrons in hadronic showers.
- Use information from track-cluster associations to identify pattern-recognition problems:



After topological association

Compare E/p values to find problems



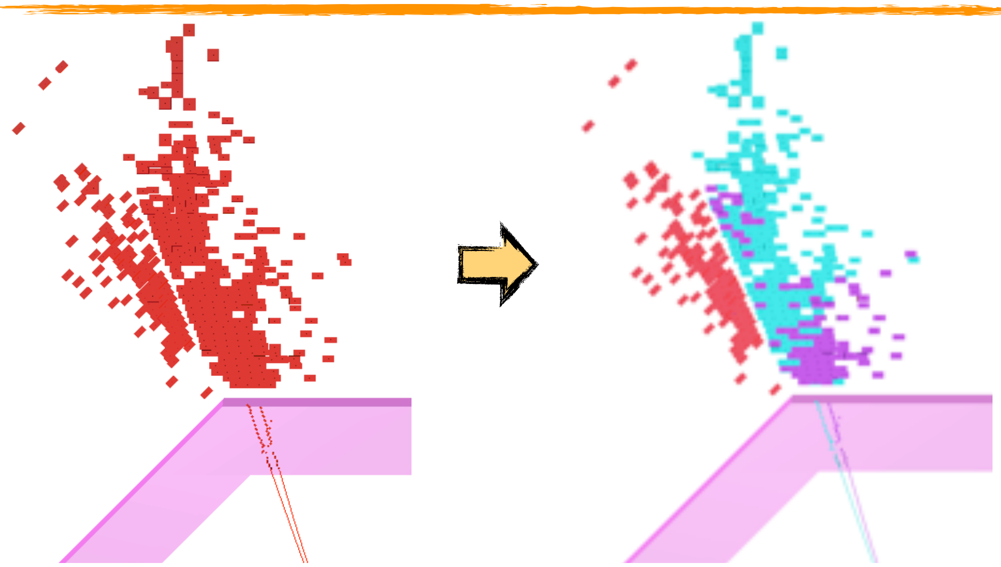
Find n absorbed into π^- cluster

e.g. 45GeV track associated to 95GeV cluster:

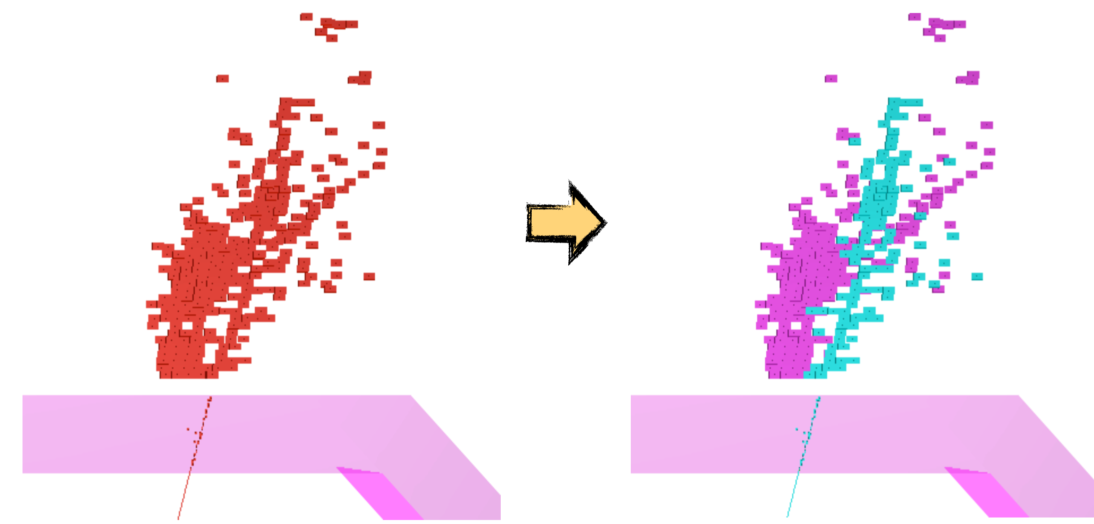
identify and address clustering problem

Pandora LC Reclustering

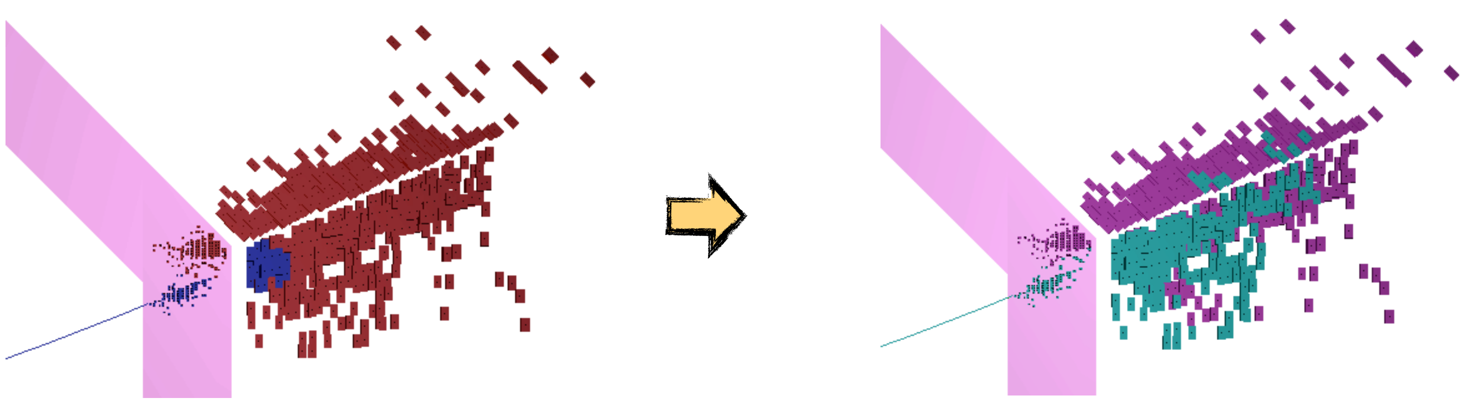
If identify significant discrepancy between cluster energy and associated track momentum, choose to recluster. Alter clustering parameters until cluster splits to obtain track-cluster consistency.



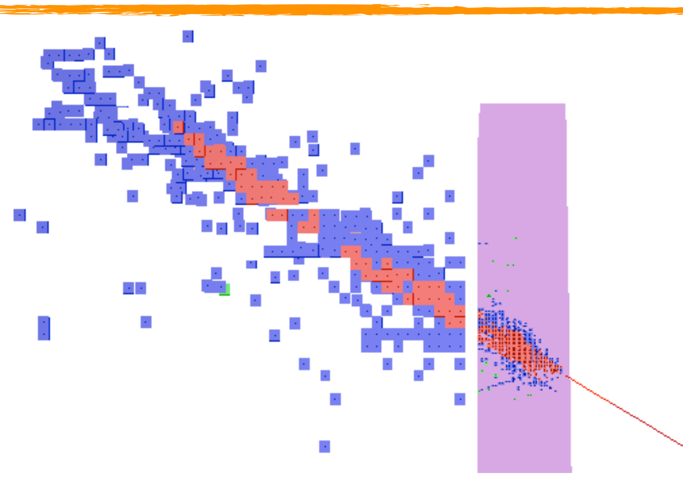
1. Multiple tracks associated to single cluster – split cluster.



2. Cluster energy much greater than track momentum – split cluster.

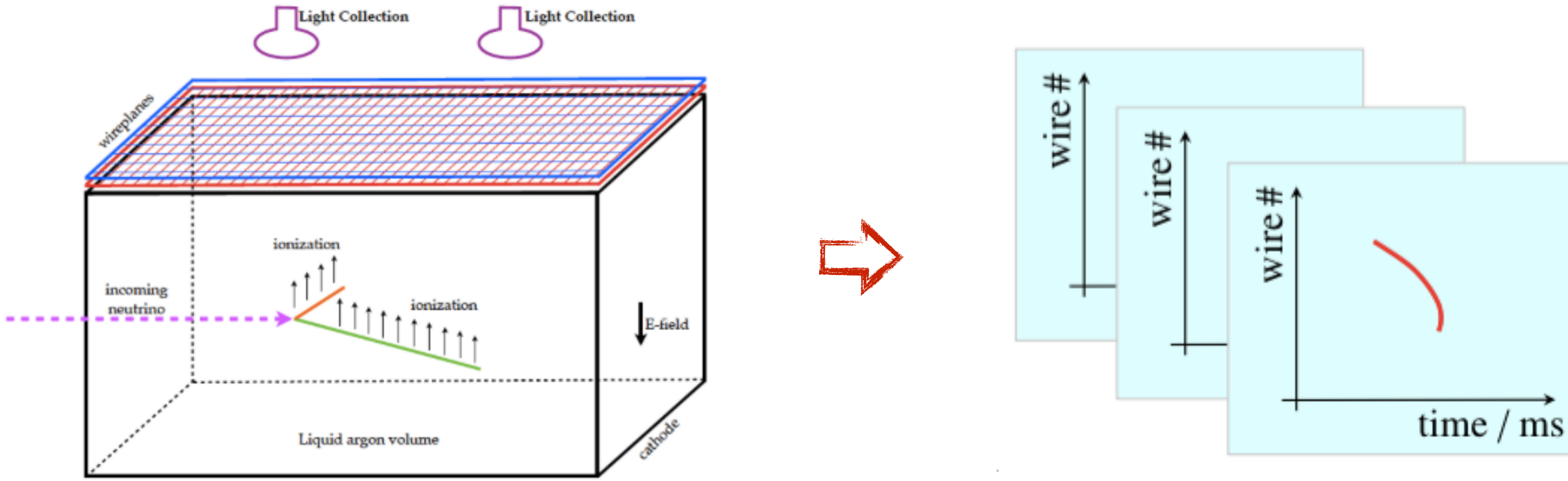


3. Track momentum much greater than cluster energy – bring in nearby clusters and reconfigure.



4. If, and only if, no E/p match emerges, can force track-cluster consistency ⇒ energy flow.

Pandora LAr TPC Reconstruction



Picture from M. Soderberg

- Ionisation electrons detected by series of wire planes, enabling particle tracking and calorimetry.
- Reveal neutrino interactions in unprecedented detail. Obtain 3 "images": wire no. vs drift time.
- **Software challenge:**
  - 3x2D reconstruction, combine results to obtain 3D image of neutrino interaction.
  - Many 'hits', diverse event topologies, 2D views with features often obscured in 1+ view.