The Iterative Clustering framework for the CMS HGCAL Reconstruction

Felice Pantaleo
CERN, Experimental Physics department

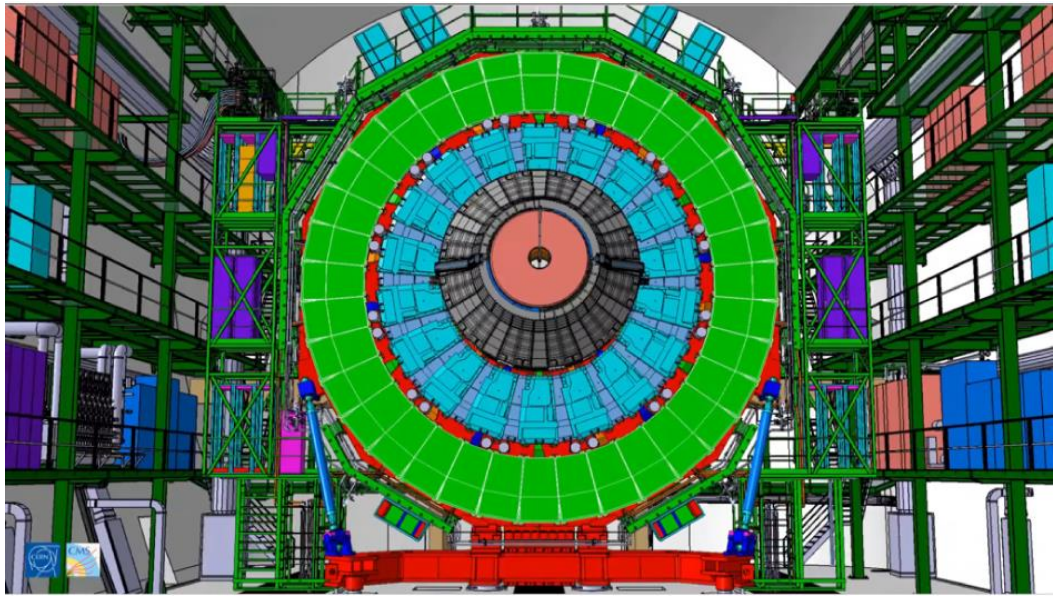
On behalf of the CMS Collaboration

felice@cern.ch

CMS Phase-2 Calorimeter Endcap

Major CMS Phase2 upgrade:

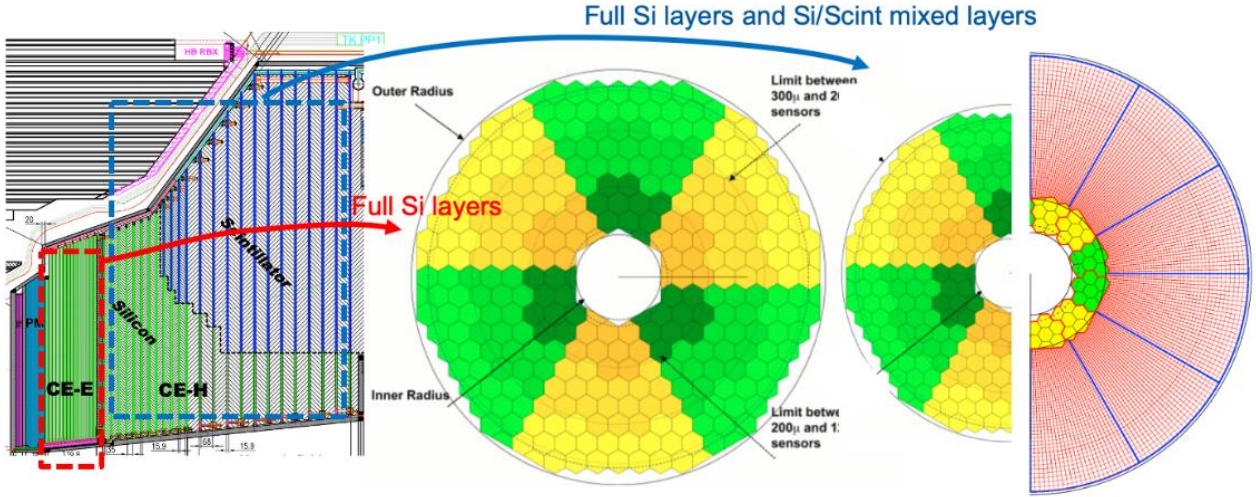
- Silicon sensors (EM + HAD)
  - 28 (EM) + 22 (HAD) layers
  - about ~6M channels, cell sizes (about 0.5 cm^2 and 1.2 cm^2)
- Plastic Scintillator + SiPM (HAD)
  - 14 layers
  - ~4K tiles (~240K channels)



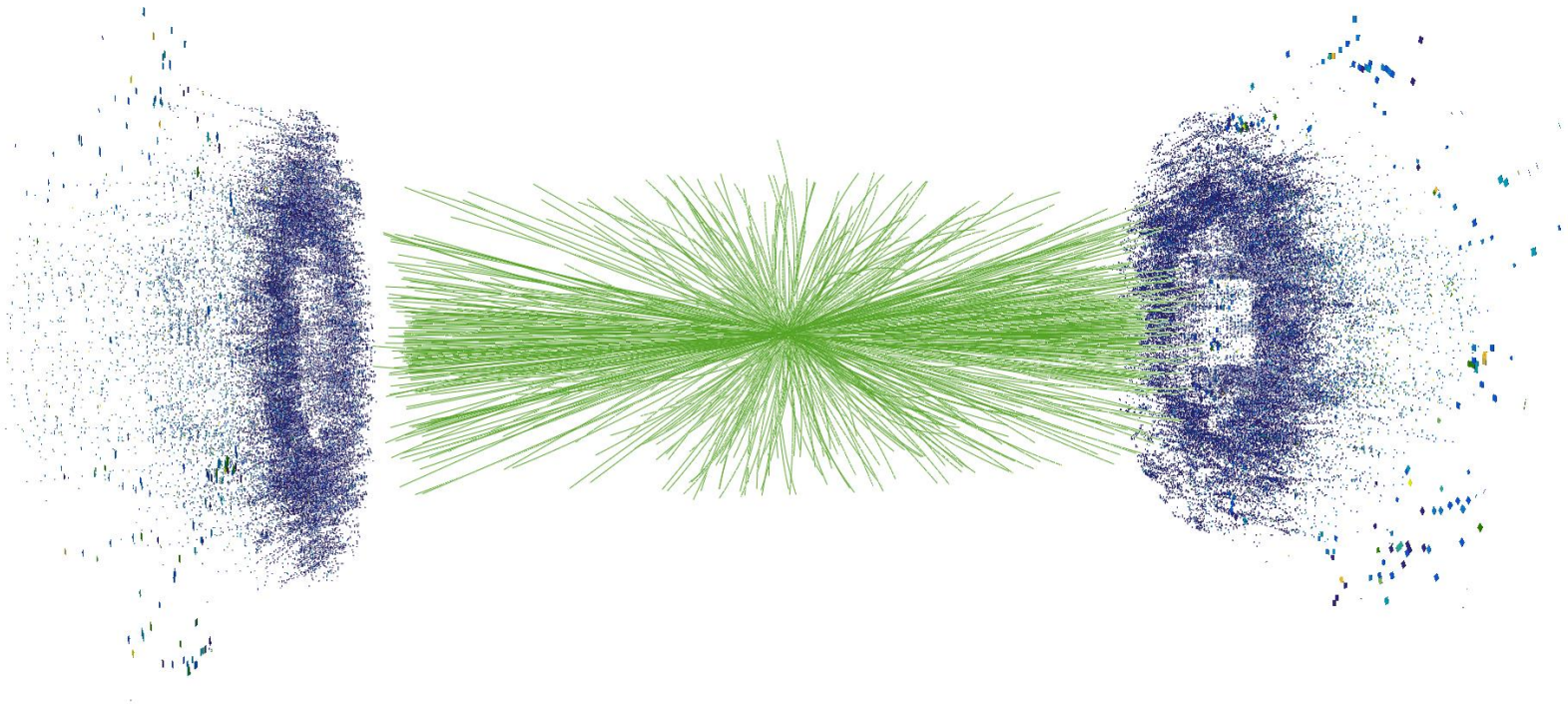
CMS Phase-2 Calorimeter Endcap

Major CMS Phase2 upgrade:

- Silicon sensors (EM + HAD)
  - 28 (EM) + 22 (HAD) layers
  - about ~6M channels, cell sizes (about 0.5 cm^2 and 1.2 cm^2)
- Plastic Scintillator + SiPM (HAD)
  - 14 layers
  - ~4K tiles (~240K channels)



The pile-up challenge



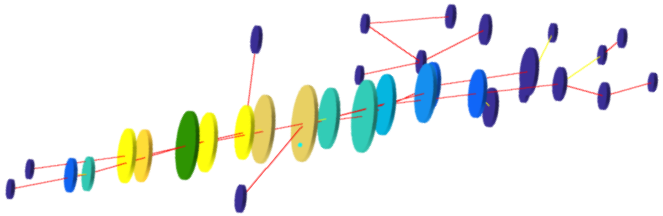
Single electron in PU200

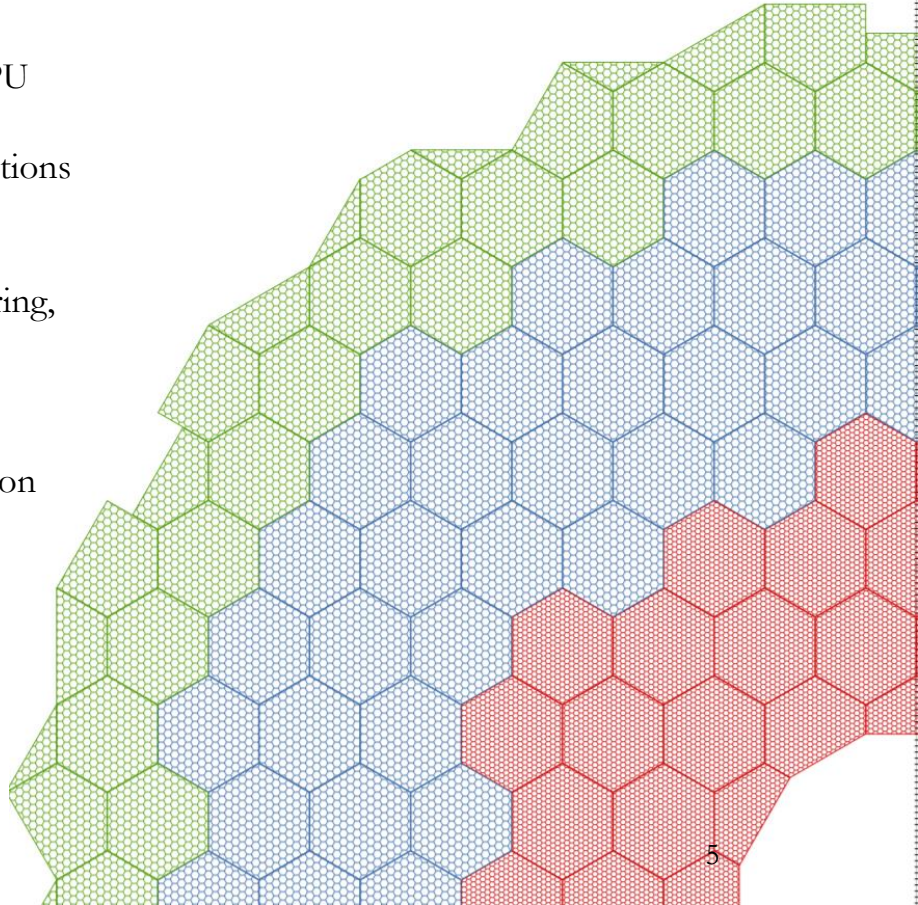
Reconstruction in HGCAL

Reconstruction in HGCAL is a very interesting task in high PU environments as overlapping showers are the norm

- naive reconstruction algorithms exploring many combinations among all possible paths are expected to fail due to memory/timing explosion
- Fertile ground for new techniques and algorithms: clustering, machine learning, graph theory, and modern computer architectures

- Must be planned and designed, considering the information from the surrounding tracking and timing detectors

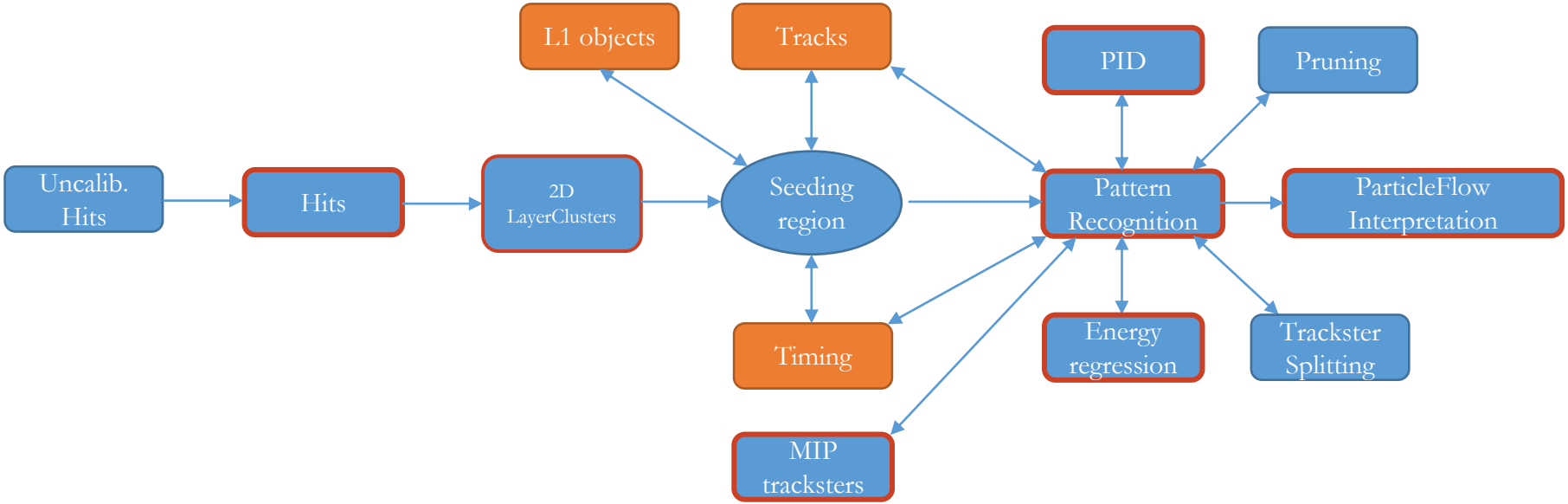




TICL – "The Iterative CLustering" Framework

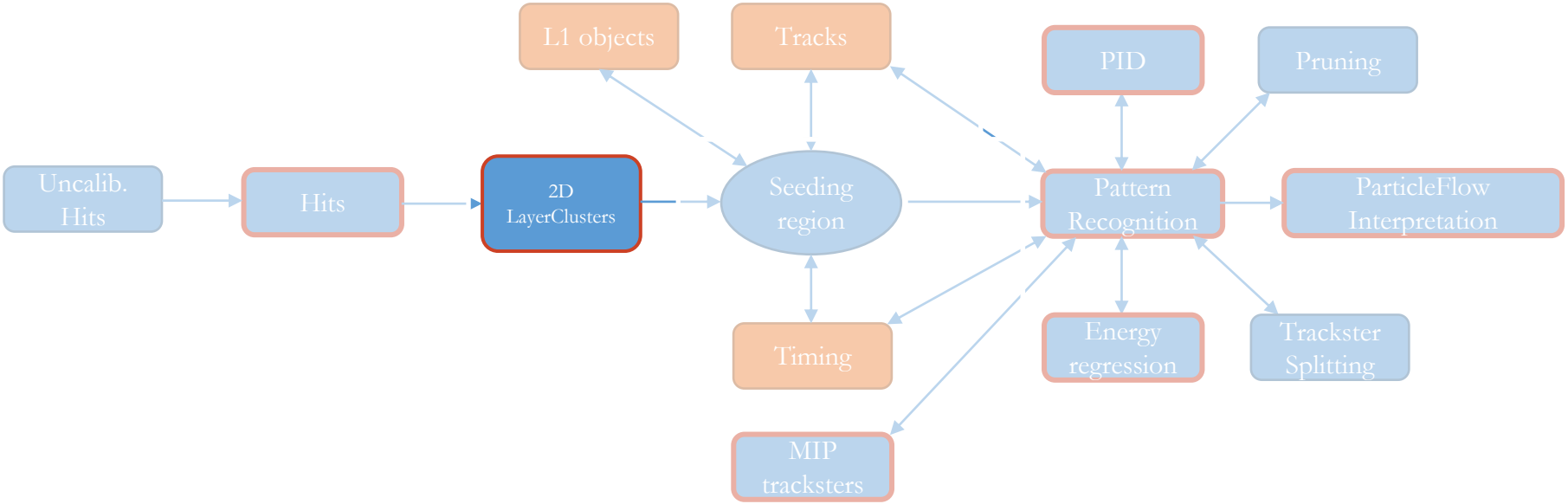
- TICL is a **modular framework** integrated and under development in CMSSW
  - Profit from regular release validation, high-quality review, computing performance monitored
- Its final purpose is to process HGCAL Hits (x, y, z, t, E) and return particle properties and identification probabilities
- The philosophy:
  - Staged and modular: decouple independent parts of the reconstruction for:
    - **independent development** by multiple collaborators
    - promptly localize and identify actions to be taken in case of unexpected failures
    - **algorithms can be easily swapped for comparison** (aka don't get too attached to your favourite clustering algo)
  - **Defined interface**: new developers don't have to know anything about the CMSSW core framework in order to contribute
  - **Validation-driven**: The performance of each stage of the reconstruction can be monitored
  - **Parallel Architecture-Friendly** (SoA data structures and algorithms being ported to CUDA/Alpaka)

HGCAL Reconstruction



- Many of these chains can co-exist in "iterations" targeting different particles
- **Performance monitored by CMSSW Validation and full Visualization support by Fireworks**

HGCAL Reconstruction

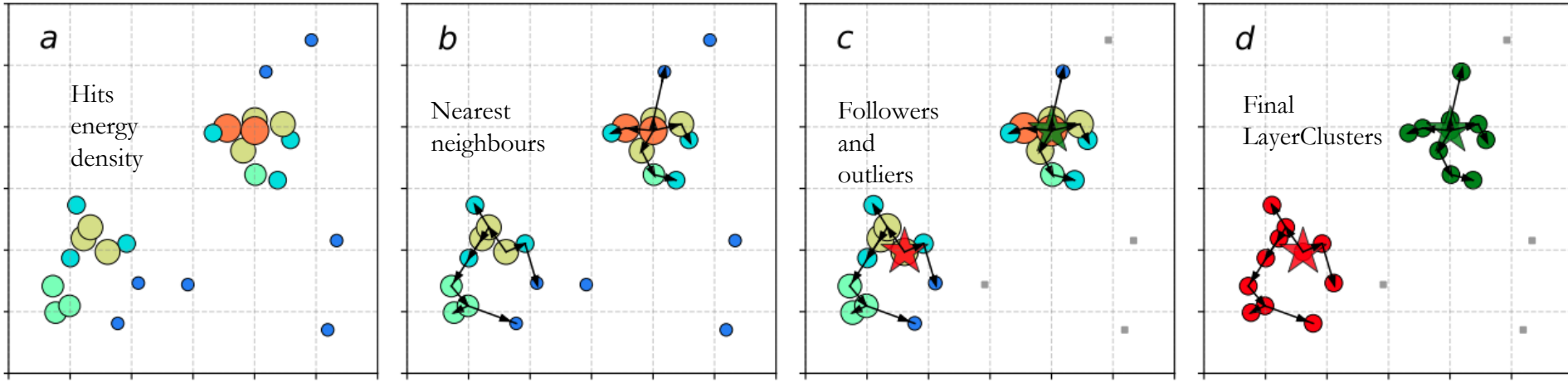


- Many of these chains can co-exists in "iterations" targeting different particles
- **Performance monitored by CMSSW Validation and full Visualization support by Fireworks**

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based

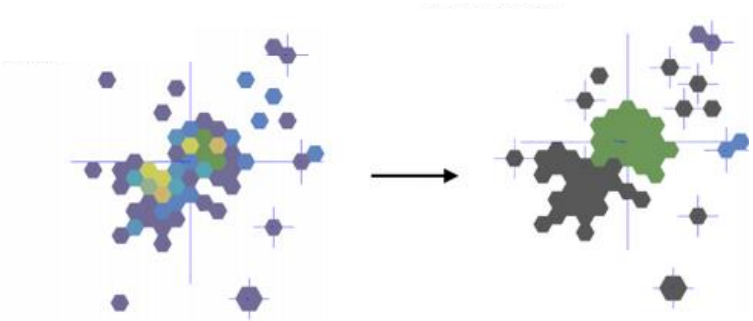


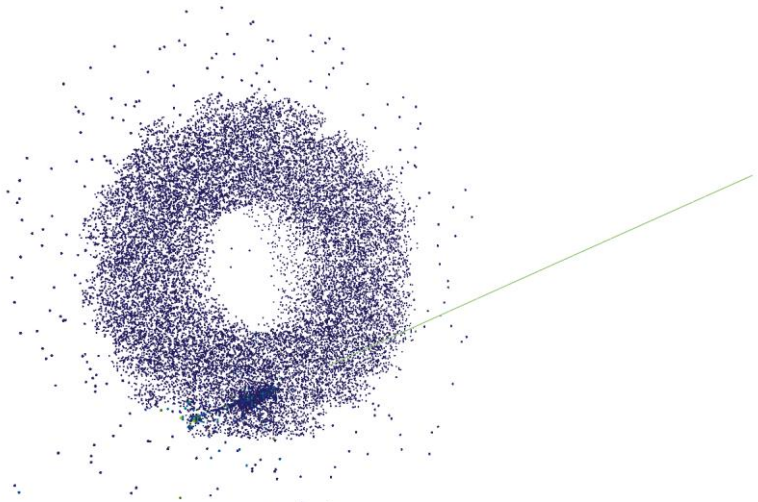
[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude





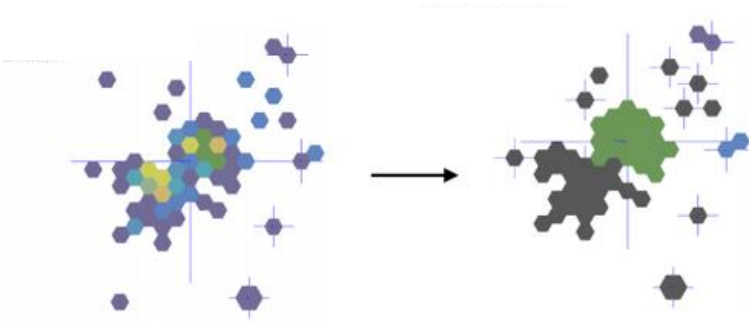
Single Pion, 0PU, **Hits**

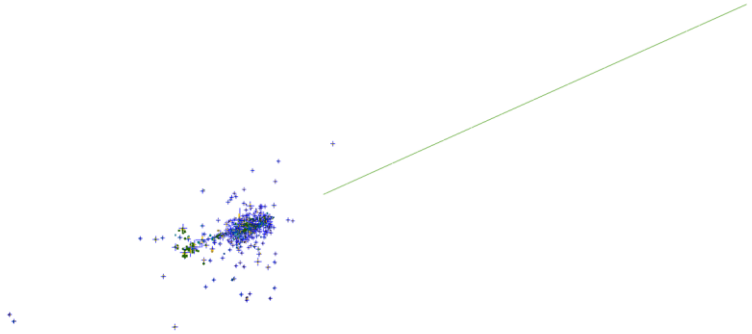
[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude





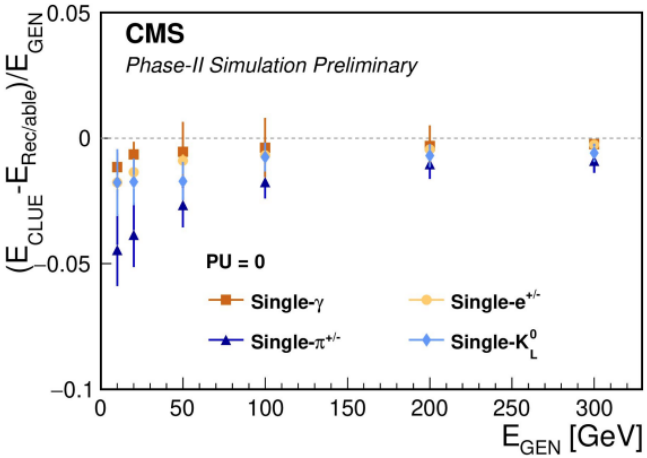
Single Pion, 0PU, **LayerClusters**

[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude
  - Minimal loss of information



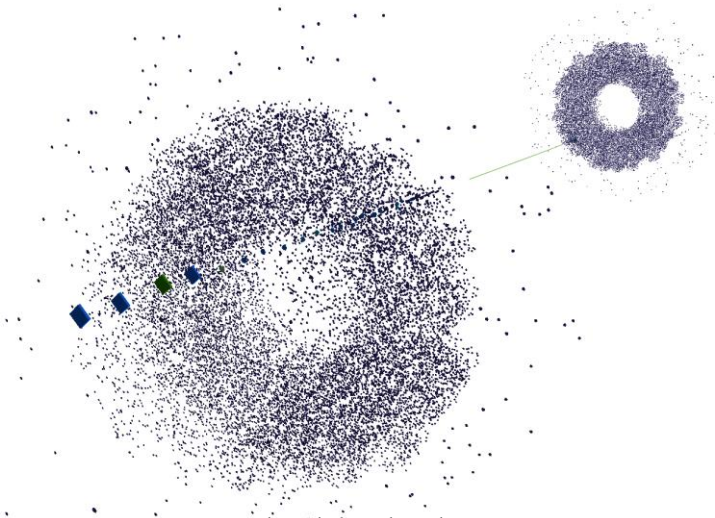
See Poster by E. Brondolin

[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude
  - Minimal loss of information



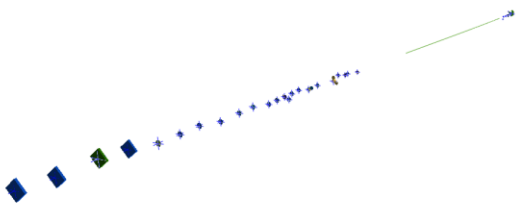
Two 10GeV Muons back-to-back, 0PU, **Hits**

[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude
  - Minimal loss of information



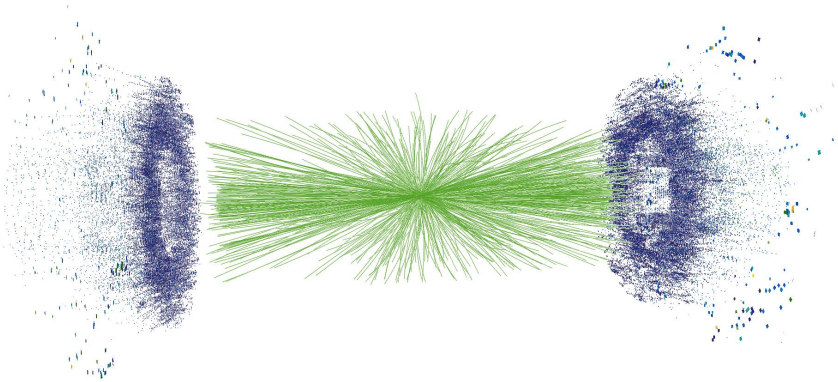
Two 10GeV Muons back-to-back, 0PU, **LayerClusters**

[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

CLUE: Layer Clustering

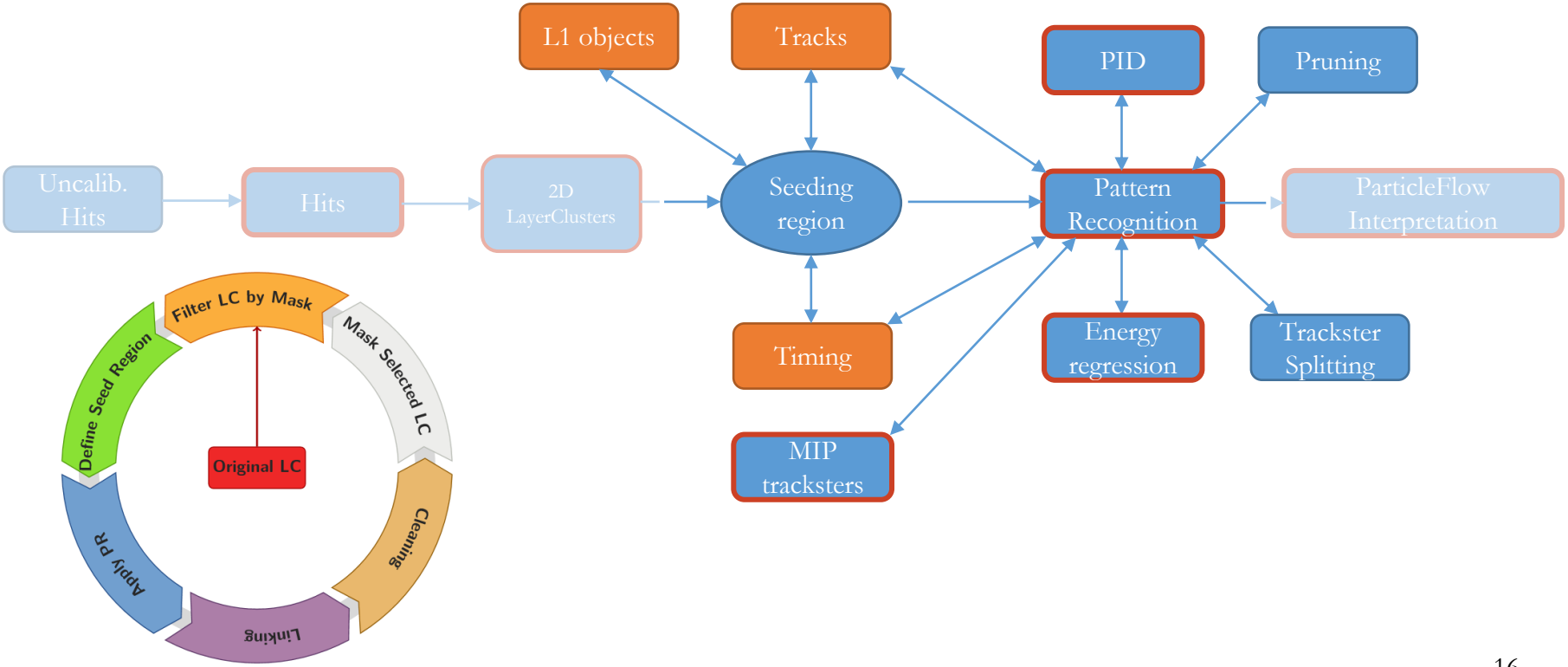
TICL's modularity starts with the Layer Clustering:

- Input Hits, output layer clusters
- Swappable plugins for 2D clustering algo
- CLUE [1] default clustering algo:
  - Energy Density Based
  - Removes Noise
  - Reduces dimensionality of the problem by an order of magnitude
  - Minimal loss of information
  - Blazing fast:
    - 0.8% of total offline reconstruction on single CPU core @PU200
    - 2 orders of magnitude faster on small GPU NVIDIA T4
    - See presentation by B. Alves



[1] https://arxiv.org/abs/2001.09761, M. Rovere, Z. Chen, A. Di Pilato, F. Pantaleo, C. Seez
CLUE: A Fast Parallel Clustering Algorithm for High Granularity Calorimeters in High Energy Physics

HGCAL Reconstruction



Seeding regions

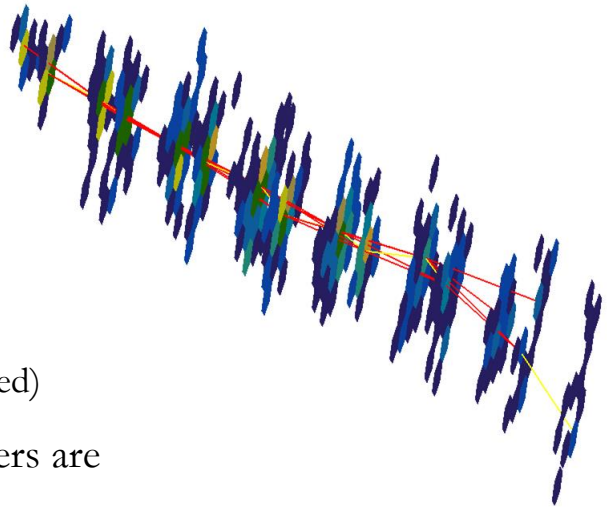
- Plugin to filter and mask LayerClusters in input
  - Layerclusters (x, y, z, E, time, hits) can be filtered
    - using HGCAL internal information
      - Min/max number of Hits in a layercluster
      - By layer id
      - By previous iteration
    - using external information:
      - Tracks
      - L1 objects
      - Timing
- Mask is a floating-point number ranging between 0 and 1

Pattern recognition: Tracksters

Tracksters are small Direct Acyclic Graphs whose nodes are layerClusters

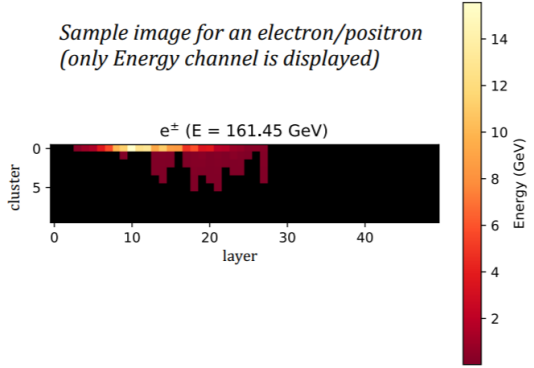
Pattern recognition can be any algorithm that connects layer clusters together to produce a **Trackster** aiming **at reconstructing single clusters** (not necessarily full particles!)

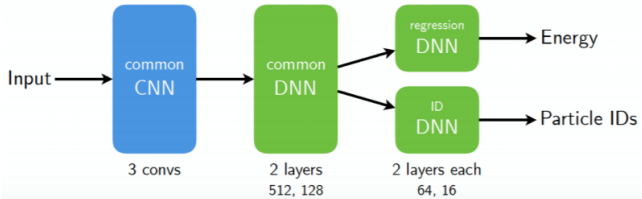
- Currently implemented plugins in CMSSW
  - Cellular Automaton
  - CLUE3D
  - FastJet
  - SimTracksters
  - MIP
- Tracksters can **be linked to a seeding object** if needed, and filled with:
  - LayerClusters & Fraction
  - Energy-weighted **PCA** decomposition
  - Regressed energy
  - Probabilities for **all particle-id classes** (allowing for fallback if needed)
  - **Timing**
- If a trackster passes some customizable filter, then all its layerclusters are **masked out.**
  - **The mask is saved in the event** and can be used by a later iteration or by validation to know which LCs are left
  - Nothing prevents one to develop a single global iteration

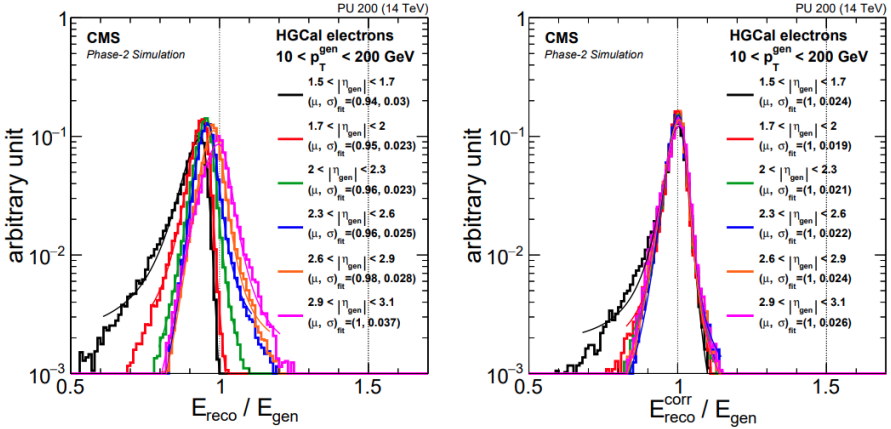


PID and Energy regression

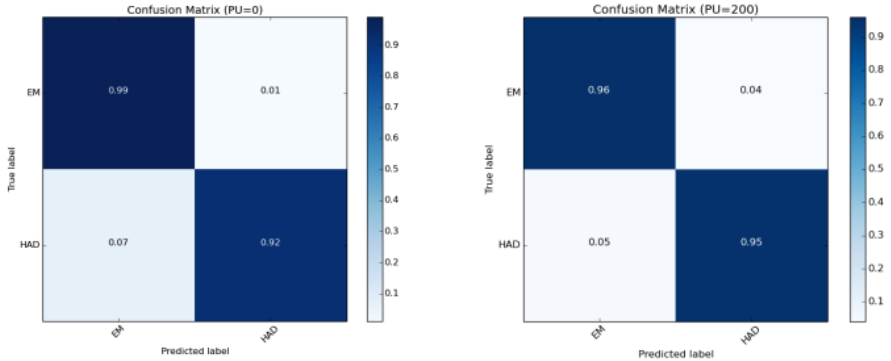
- Particle ID is applied using a single Convolutional Neural Network
- TensorFlow C++ interface
- PID probabilities and energy value as output







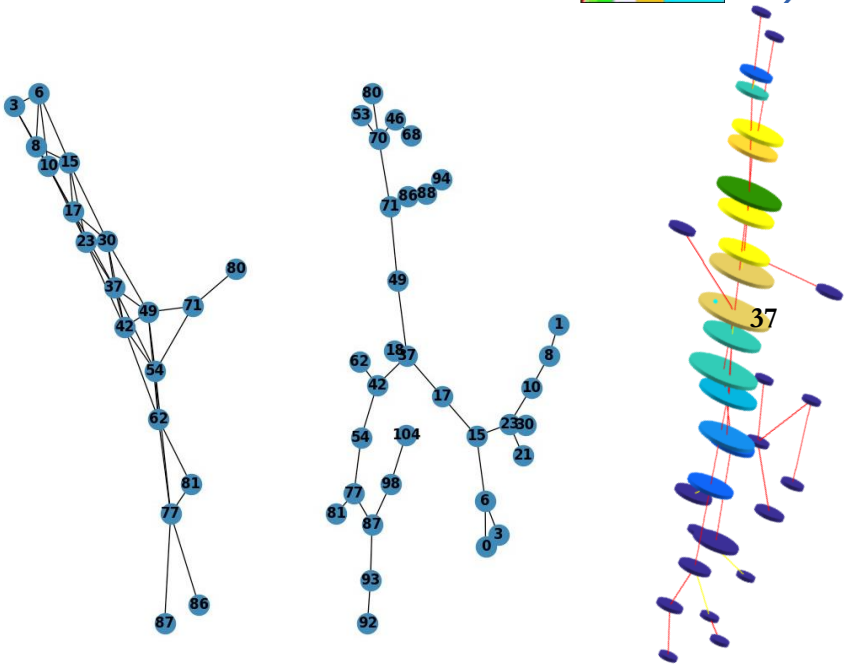
Ratio of the reconstructed to the true electron energy before (left) and after (right) the energy correction, shown for different η regions in PU200



Confusion matrix for electromagnetic showers and hadron showers in events with PU=0 (left) and in events with <PU> = 200 (right)

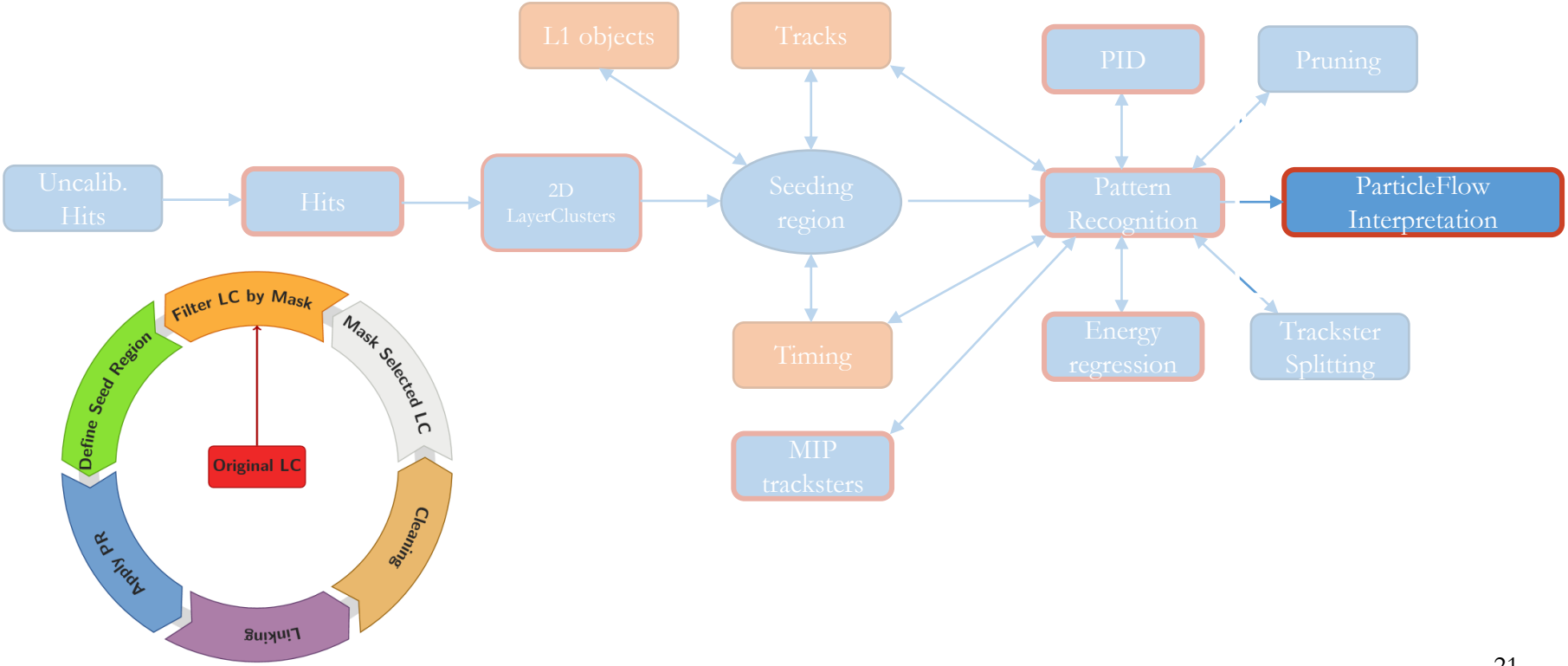
Graph studies

- Tracksters are graphs
  - LayerClusters as nodes
  - Edges depend on algorithm used“
- **Enabler for different studies** for:
  - PID/Energy regression
  - Pruning
  - Weakly connected subcomponent analysis and splitting
  - Topology, shower shape, missing information
  - Energy flow
  - Linking, ParticleFlow Interpretation



Shower reconstructed by CA (left) or CLUE3D (center and right)

HGCAL Reconstruction



Particle Flow interpretation

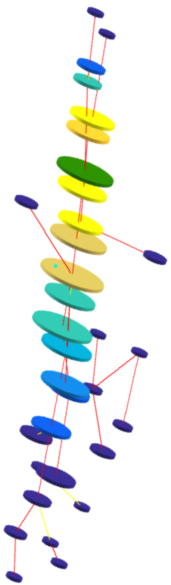
Linking

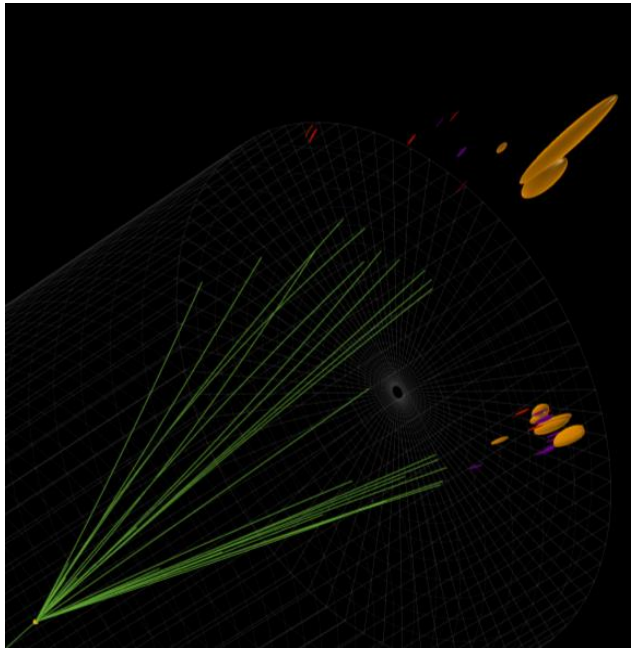
- PCA and pT/Energy/timing/PID enables compatibility between Tracksters and Tracks

MIP reconstruction

- link energy blobs
- muons
- calibration
- muon-seeded tracking

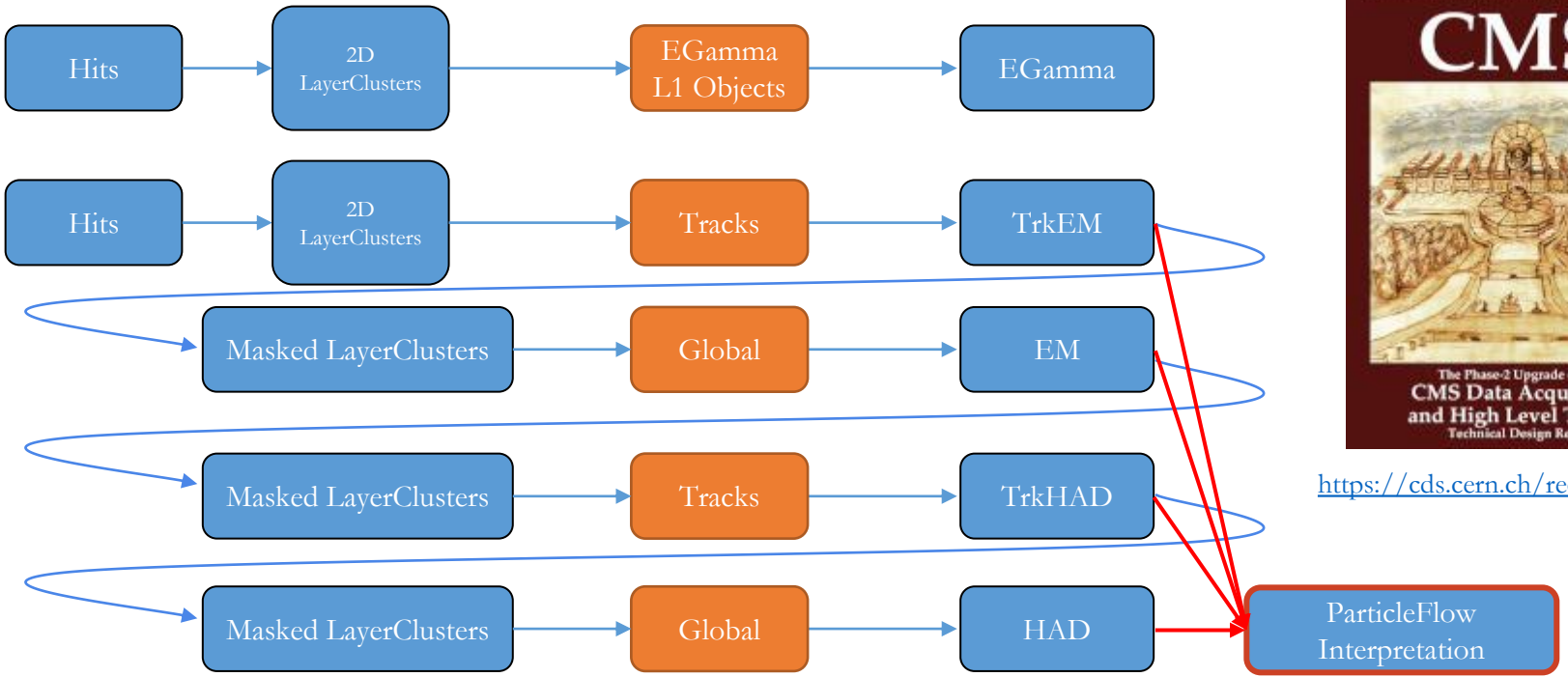
TICL Candidate is then fed to Particle Flow

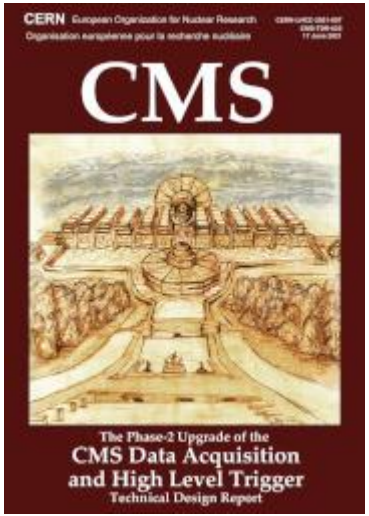




TICL iterations for HLT TDR studies

Multiple iterations and configurations implemented for the CMS Phase 2 HLT TDR studies





https://cds.cern.ch/record/2759072

Conclusion

- TICL is a flexible framework in release under active development
  - New pattern recognitions can be plugged in and validated automatically
  - Relieves the developer from bringing the idea from proof of concept to production by defining interfaces
- Full HGCAL Reconstruction using TICL takes about 5% of total Phase-2 offline reconstruction on single CPU core @ PU200
  - Unique playground for new ideas and technologies in CMSSW
- Porting of existing algos to GPUs and Portability in CMSSW advanced
- Robust PID/ER/PCA are key to achieve best ParticleFlow Interpretation