PandoraPFA on ALLEGRO

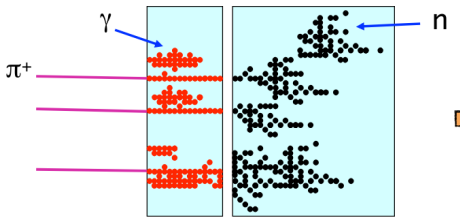
A. Durglishvili (HEPI TSU), M. Mlynarikova (CERN), H. Wilkens (CERN)

FCC Full Sim Working Meeting
27 November 2024

Particle Flow Calorimetry

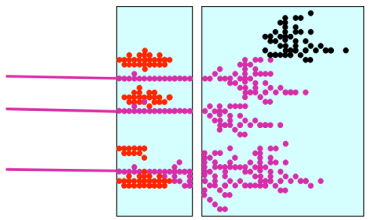
- **In a typical jet:**
  - 60 % of jet energy in charged hadrons
  - 30 % in photons (mainly from $\pi^0 \rightarrow \gamma\gamma$)
  - 10 % in neutral hadrons (mainly *n* and K_L)
- **Conventional calorimetric approach:**
  - Measure all components of jet energy in ECAL/HCAL
  - 50-70% of energy measured in HCAL: $\sigma_E/E \approx 60\% /\sqrt{E}$
- **Particle Flow Calorimetry: reconstruct individual particles**
  - Charged particle momentum measured in tracker (essentially perfectly)
  - Photon energies measured in ECAL
  - Neutral hadron energies measured in HCAL

Conventional approach



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

Particle Flow approach



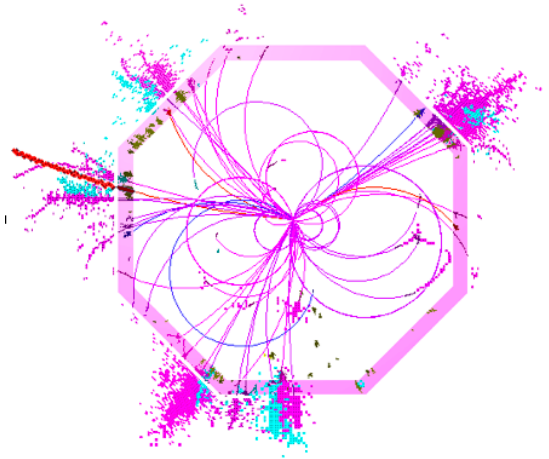
$E_{JET} = E_{TRACK} + E_{\gamma} + E_n$

Picture taken from the link

Pandora Particle Flow Algorithm

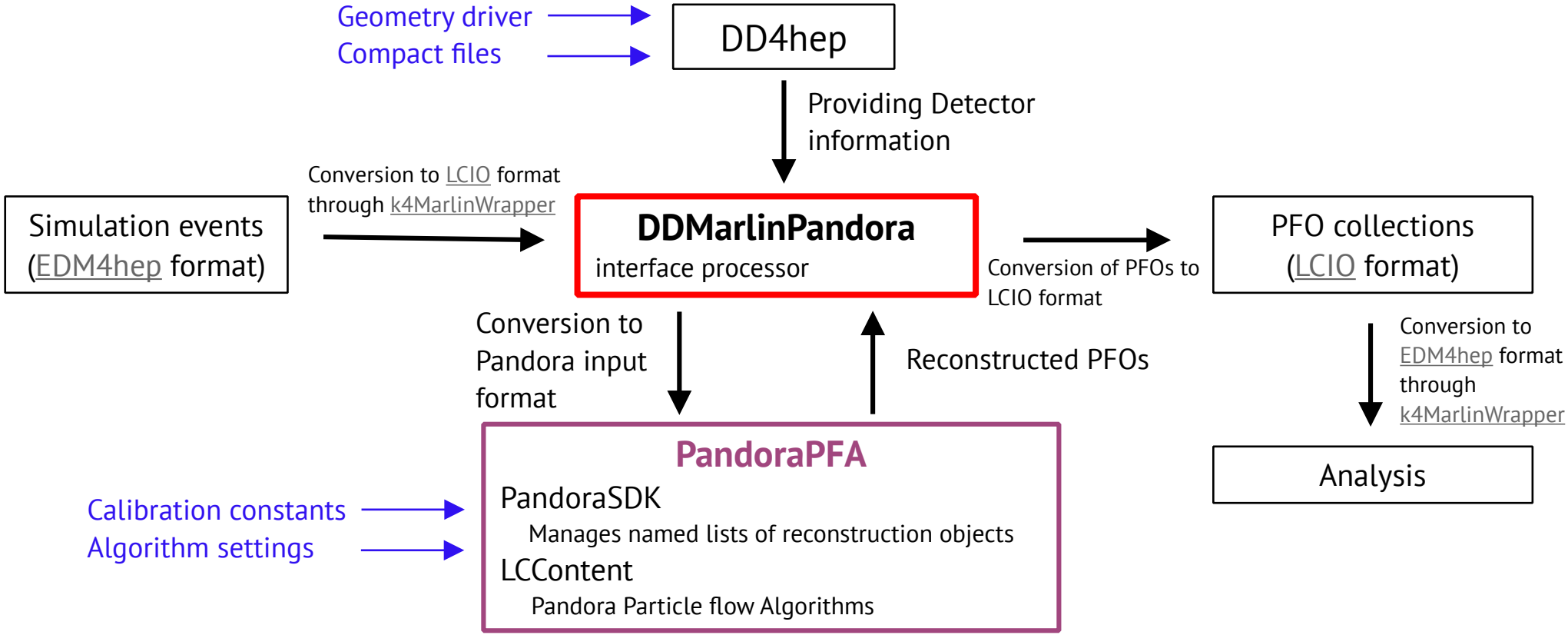
- **PandoraPFA** originally developed for the application in a future Lepton Collider experiments
  - Implemented in iLCSoft
- Widely considered as a “state of the art” in particle flow reconstruction
- Used in FCC-ee CLD detector simulation
- We aim to implement the PandoraPFA in the ALLEGRO detector simulation

Multi-algorithm pattern recognition
PandoraPFA



https://github.com/PandoraPFA
https://arxiv.org/abs/0907.3577
https://arxiv.org/abs/1506.05348
PandoraPFA algorithm overview

PandoraPFA implementation in key4hep



Eventually, *DDMarlinPandora* and *k4MarlinWrapper* combination should be replaced with k4GaudiPandora

Inputs for PandoraPFA

Minimal inputs that have to be provided to *DDMarlinPandora* to be able to run the *PandoraPFA* without crashing:

- Detector information (layers, material, etc...) ← Fully available only for ECAL barrel in k4geo
- Reconstructed Calorimeter (ECAL/HCAL/MUON) hits ← No reconstructed MUON hits in ALLEGRO
- Tracks (at IP/FirstHit/LastHit/Calorimeter states) ← Tracks reconstructed from truth particles are available but only at IP state: k4RecTracker
- KinkVertexCollections
- ProngVertexCollections
- SplitVertexCollections
- V0VertexCollections

These are not available

→ A lot of work has to be done to be able to run the PandoraPFA on ALLEGRO

PandoraPFA on ALLEGRO: Work plan

- Prepare steering scripts to run the PandoraPFA reconstruction
- Make PandoraPFA see the ALLEGRO ECAL and HCAL
- Reconstruct **photons** -- allows to check if ECAL implementation works
- Reconstruct K^0_L -- allows to check if HCAL implementation works
- Produce tracks from MCParticles with all information needed for PandoraPFA
- Make PandoraPFA see the tracks
- Reconstruct **electrons** -- allows to check if PandoraPFA can see both tracks and ECAL
- Reconstruct **charged pions** -- allows to check if PandoraPFA can see tracks, ECAL and HCAL
- Reconstruct **jets** -- allows to check if PandoraPFA can separate the charged and neutral particle showers

Eventually, the tracker and muon system should be implemented and muons reconstructed using PandoraPFA

Steering scripts

- An example steering script has been prepared: ALLEGROReconstruction.py

```
391  ##############################################
392  ##  Pandora
393  ##############################################
394  from Configurables import MarlinProcessorWrapper
395  pandora = MarlinProcessorWrapper('DDMarlinPandora')
396  pandora.OutputLevel = DEBUG
397  pandora.ProcessorType = 'DDPandoraPFANewProcessor'
398  pandora.Parameters = {
399                                  "PandoraSettingsXmlFile": ["PandoraSettingsDefault.xml"],
400                                  "ECalMipThreshold": ["0."],
401                                  "HCalMipThreshold": ["0."],
402                                  "ECalToHadGeVCalibrationBarrel": ["1."], # this must be calculated for ALLEGRO
403                                  "ECalToHadGeVCalibrationEndCap": ["1."], # this must be calculated for ALLEGRO
404                                  "HCalToHadGeVCalibration": ["1."], # this must be calculated for ALLEGRO
405                                  "ECalToMipCalibration": ["175.439"], # value is from CLD -> this must be calculated for ALLEGRO
406                                  "HCalToMipCalibration": ["49.7512"], # value is from CLD -> this must be calculated for ALLEGRO
407                                  "DigitalMuonHits": ["0"],
408                                  "MaxHCalHitHadronicEnergy": ["10000000."],
409                                  "MuonToMipCalibration": ["20703.9"], # value is from CLD -> this must be calculated for ALLEGRO
410                                  "ECalToEMGeVCalibration": ["1.0"], # this seems to be an EM scale factor for ECAL: set to 1 since input cell energy is already calibrated at EM scale
411                                  "HCalToEMGeVCalibration": ["1.0"], # this seems to be an EM scale factor for HCAL: set to 1 since input cell energy is already calibrated at EM scale
412                                  "DetectorName" : ["ALLEGRO"],
413                                  "UseDD4hepField" : ["1"],
414                                  "MCParticleCollections" : ["MCParticle"],
415                                  "ECalCaloHitCollections" : ["ECalBarrelModuleThetaMergedPositioned"],
416                                  #"HCalCaloHitCollections" : ["HCalBarrelReadoutPositioned","HCalEndcapReadoutPositioned"],
417                                  "HCalCaloHitCollections" : ["HCalBarrelReadoutPositioned"],
418                                  "TrackCollections" : ["TracksFromGenParticles"],
419
420                      }
```

Default PandoraPFA algorithm settings file (found in CLDConfig repository)
Only the muon reconstruction algorithm is removed

"DetectorName" parameter has been added in DDPandoraPFANewProcessor to setup the ALLEGRO detector geometry/CaloHits/Tracks

Currently working with ECAL and HCAL barrel cells only

ALLEGRO ECAL and HCAL in PandoraPFA

- *DDMarlinPandora* retrieves the detector information from *LayeredCalorimeterData* extension attached to the calo *DetElement*
  - If the extension is not available for any component of ECAL/HCAL/MUON system then the code crashes
  - Extension must include:
    - Type of the detector
    - Information about layers (X0, lambda, size, …)
  - In official k4geo repo, this information is fully provided only for ECAL Barrel
  - In my k4geo repo, I have added minimal information for ECAL Endcap, HCAL Endcap and Muon system while complete (in a first approximation) information for HCAL Barrel

- *DDMarlinPandora* package is forked and modified for ALLEGRO detector
- New classes:
  - DDGeometryCreatorALLEGRO
  - DDCaloHitCreatorALLEGRO
  - DDTrackCreatorALLEGRO
- At this stage, MUON hits creation is disabled as well as tracking, focusing on reconstruction of particle flow objects in ECAL and HCAL barrels only

→ these are not enough to avoid crashing… →

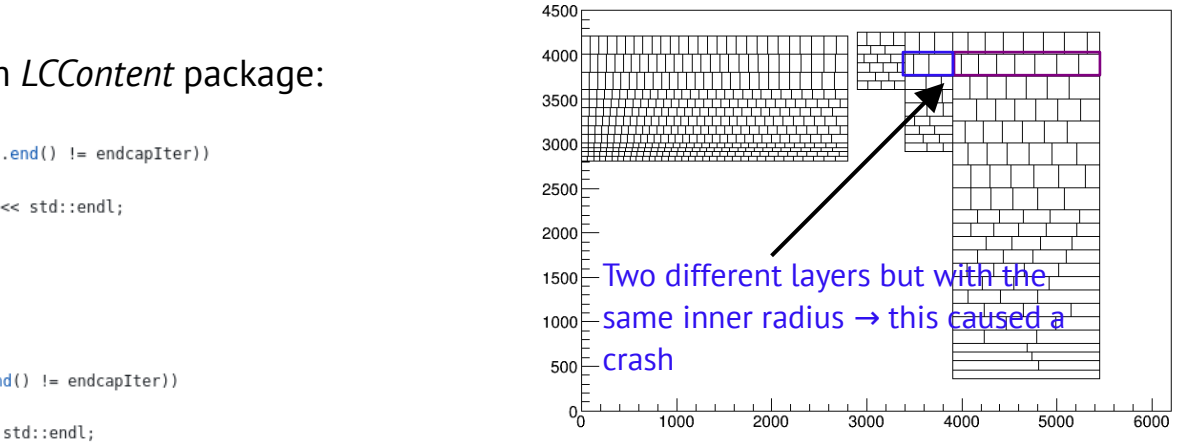
```
LCPseudoLayerPlugin: Duplicate layer position detected.
LCPseudoLayerPlugin: Incomplete geometry - consider using a different PseudoLayerCalculator.
m_pPseudoLayerPlugin->Initialize() return STATUS_CODE_FAILURE
    in function: InitializePlugins
    in file:     /tmp/root/spack-stage/spack-stage-pandorasdk-3.4.2-zulcppdz6l6e47lrfsuo4qqajrjlv2
we/spack-src/src/Managers/PluginManager.cc line#: 230
m_pPandoraImpl->InitializePlugins(&xmlHandle) throw STATUS_CODE_FAILURE
```

ALLEGRO ECAL and HCAL in PandoraPFA (cont'd)

The crash is caused by the *LCPseudoLayerPlugin* from *LCContent* package:

```
175         if ((m_barrelLayerPositions.end() != barrelIter) || (m_endCapLayerPositions.end() != endcapIter))
176         {
177             std::cout << "LCPseudoLayerPlugin: Duplicate layer position detected." << std::endl;
178             throw StatusCodeException(STATUS_CODE_FAILURE);
179         }
```

```
175         if ((m_barrelLayerPositions.end() != barrelIter) || (m_endCapLayerPositions.end() != endcapIter))
176         {
177             std::cout << "LCPseudoLayerPlugin: Duplicate layer position detected." << std::endl;
178             // FIXME:  AD: for the ALLEGRO HCAL ENDCAP we have different layers with the same inner radius -> for the moment throwing the exception is disabled...
179             // throw StatusCodeException(STATUS_CODE_FAILURE);
180         }
```



→ this change enabled running the PandoraPFA with ALLEGRO detector!

Temporarily, throwing the exception is disabled to avoid a crash.
Later, HCAL Endcap segmentation should be optimized for PandoraPFA:
e.g. group cells in the longitudinal readout layers (this is what PandroaPFA expects as input).

We should keep in mind (eventually fix it) that the Pandora will assume some of the cells from HCAL endcap as HCAL barrel cells:

```
92  StatusCode LCPseudoLayerPlugin::GetPseudoLayer(const float rCoordinate, const float zCoordinate, const float rCorrection,
93      const float zCorrection, const float barrelInnerR, const float endCapInnerZ, unsigned int &pseudoLayer) const
94  {
95      if (zCoordinate < endCapInnerZ)
96      {
97          return this->FindMatchingLayer(rCoordinate, m_barrelLayerPositions, pseudoLayer);
98      }
```

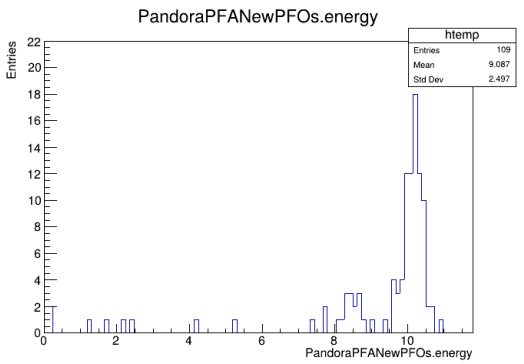
endCapInnerZ is the inner z-coordinate of ECAL endcap while some of the cells in HCAL Endcap have lower inner z-coordinate

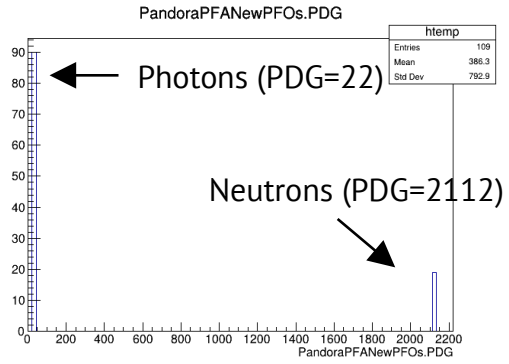
Photon reconstruction using PandoraPFA in ALLEGRO

- 100 events of 10 GeV *photons* are simulated at 60° to check if PandoraPFA can use ECAL hits for the reconstruction
- PhotonReconstruction algorithm calibration file (PandoraLikelihoodData9EBin.xml) is taken from CLDConfig
- For comparison, CLD simulation + reconstruction is also performed following this instructions

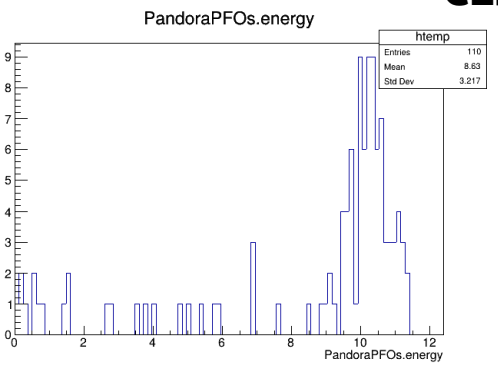
PandoraPFA can identify photons in the ALLEGRO detector!

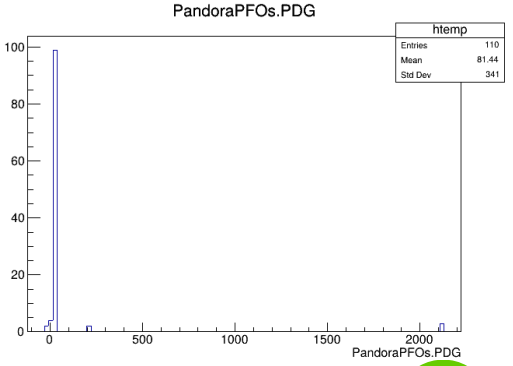
ALLEGRO





CLD

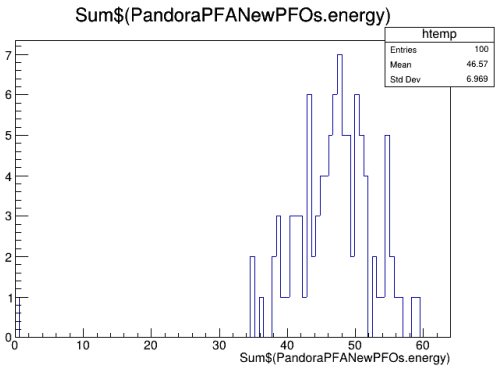


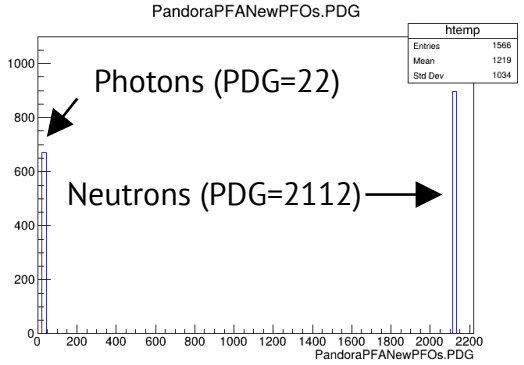


K^0_L reconstruction using PandoraPFA in ALLEGRO

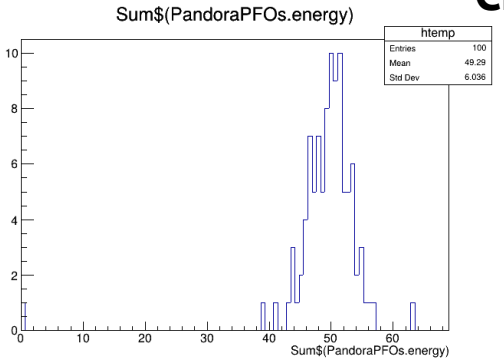
- 100 events of 50 GeV K^0_L are simulated at 60° to check if PandoraPFA can use both ECAL and HCAL hits for the reconstruction
- For comparison, CLD simulation + reconstruction is also performed following this instructions
- PandoraPFA can reconstruct hadronic showers in the ALLEGRO detector
- However, too many clusters are produced → needs further investigation

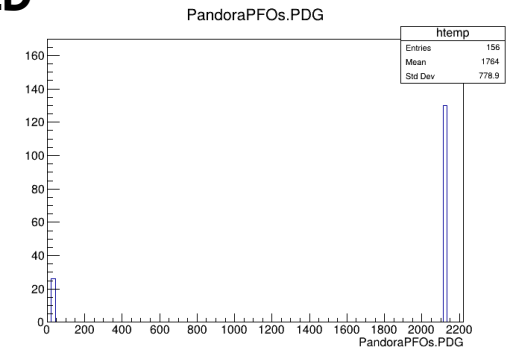
ALLEGRO





CLD





Tracks for PandoraPFA

- Currently, the tracking system taken from IDEA "as is"
  - No track reconstruction performed
- Since the reconstructed track resolution is expected to be almost perfect, we can work with "truth tracks"
- TracksFromGenParticles algorithm available in k4RecTracker is modified to form the track objects with all necessary information for PandoraPFA
  - Tracks are created only from charged MCParticles
  - Track state at IP is defined at the position of MCParticle production vertex
  - Track state at FirstHit is defined using the position of the MCParticle associated lowest radius Geant4 hit in the Drift Chamber, and the extrapolated momentum from the track state at IP
  - Track state at LastHit is defined using the position of the MCParticle associated highest radius Geant4 hit in the Drift Chamber, and the extrapolated momentum from the track state at FirstHit
  - Track state at Calorimeter is defined using the position of the extrapolated track intersection point to the cylinder of the ECal barrel at inner radius (2172.8 mm), and the extrapolated momentum from the track state at LastHit
  - The algorithm can be further improved to produce kink/prong/split/V0 vertex collection

NOTE: Since I do not have any experience with tracking, what I did here may not be correct/optimal choice...

Tracks from MCParticles in PandoraPFA

- Newly created class of DDTrackCreatorALLEGRO in DDMarlinPandora allows to feed the PandoraPFA with the tracks produced using TracksFromGenParticles
  - The default cuts are applied on d0 and z0 (50 mm)
  - Checked if a track reaches the ECAL barrel (based on the track state at Calo)
  - By default, the track is assumed to be a charged pion
- CreateTrackAssociations is disabled due to missing relevant collections
  - Attempts to identify $\gamma \rightarrow e^+e^-$, $K_S^0 \rightarrow \pi^+\pi^-$, $\Lambda \rightarrow p\pi^-$ and $\overline{\Lambda} \rightarrow \overline{p}\pi^+$ decays
  - Attempts to identify muons from $\pi^{\pm}/K^{\pm}$ decays, and charged pions from charged Hyperon decays
  - Relates daughter tracks to the parent and produces sibling relationship →

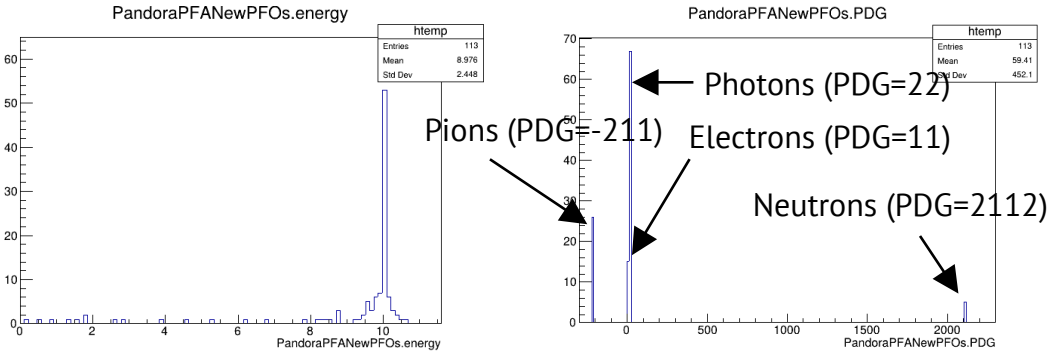
    at the later stage, the PFO is reconstructed using the parent track, and the particle Id is assigned retrieved from the corresponding vertex

→ since this is disabled, all charged hadron tracks will be assigned the default particle Id of charged pion

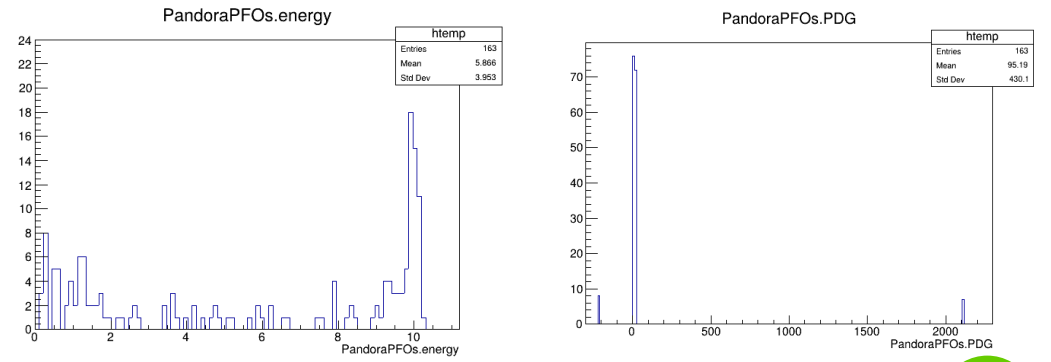
Electron reconstruction using PandoraPFA in ALLEGRO

- 100 events of 10 GeV *electrons* are simulated at 60° to check if PandoraPFA can use tracks and ECAL hits for the reconstruction
- For comparison, CLD simulation + reconstruction is also performed following this instructions
- PandoraPFA can reconstruct electrons using tracks and showers in the ECAL of the ALLEGRO detector
- It can well associate the shower in ECAL to the track and use the track energy instead of Calorimeter energy
- Failed to identify the track as an electron
  - A track is identified as electron if the associated shower cluster is classified as EM shower
  - By changing the cluster RMS cut from 40mm to 70mm manged to identify ~15 tracks as electrons
  - Clearly, tuning of EM shower related parameters is necessary

ALLEGRO



CLD

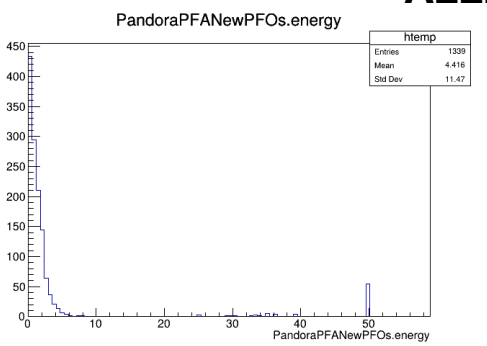


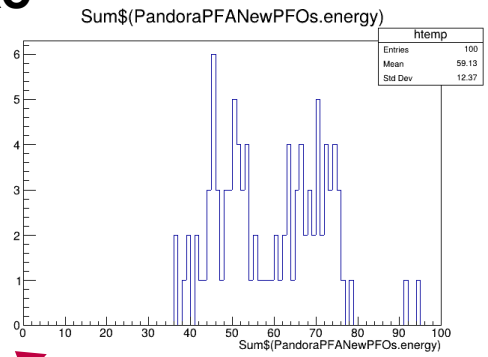
Charged pion reconstruction using PandoraPFA in ALLEGRO

- 100 events of 50 GeV π- are simulated at 60° to check if PandoraPFA can use tracks and ECAL+HCAL hits for the reconstruction
- For comparison, CLD simulation + reconstruction is also performed following this instructions
- PandoraPFA can reconstruct hadronic showers in the ALLEGRO detector and use charged hadron tracks
- Failed to associate the showers to the track → energy is double counted (confusion)!
- Too many clusters are produced

  → needs further investigation

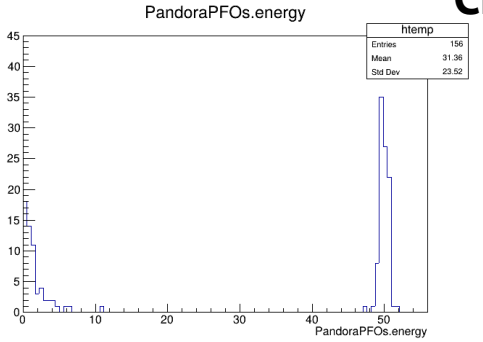
ALLEGRO

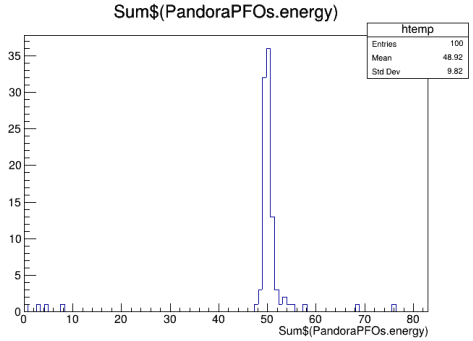




Energy is double counted!

CLD





Summary

Implementation of PandoraPFA in the ALLEGRO detector simulation has started and progressing well

- Currently focusing on the ECAL and HCAL barrels
- PandoraPFA can identify photons in the ALLEGRO detector very well
- Managed to reconstruct hadronic showers, however, too many clusters are created → needs further investigation
- Tracks for MCParticles are produced to use in the PandoraPFA algorithm
- Managed to reconstruct electrons from tracks and showers in the ECAL
- Shower in ECAL is well associated to the electron track, however, failed to classify electron shower as EM shower → tuning of parameters is necessary
- Failed to associate charged pion shower to the track → energy is double counted → needs further investigation

Instructions how to run the PandoraPFA (in the current state) in the ALLEGRO reconstruction:
https://github.com/Archil-AD/ALLEGRO_PandoraPFA

Thank you for your attention

Backup