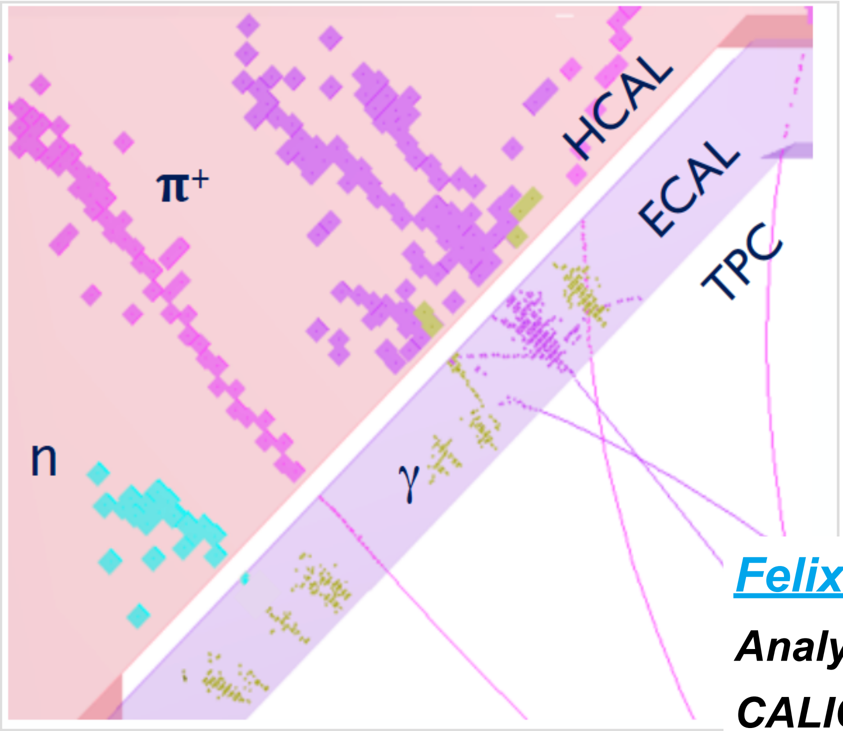
Software compensation in Pandora



Felix Sefkow, Huong Lan Tran

Analysis session - 16/09/2016

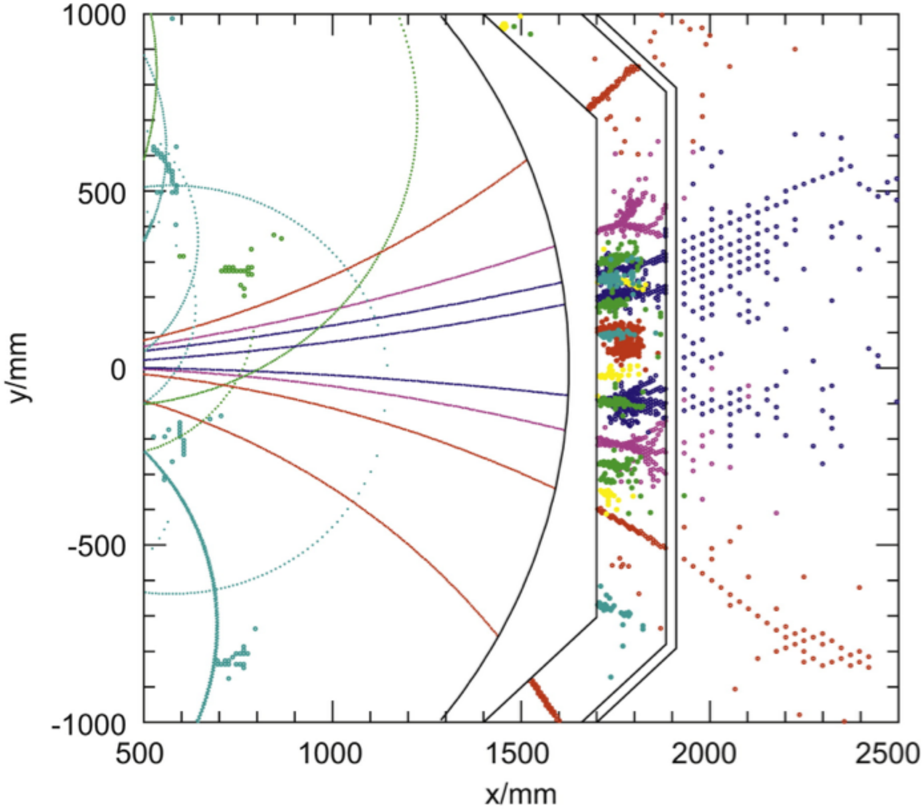
CALICE Collaboration Meeting, UT Arlington

Outlines

- Particle flow reconstruction and software compensation
- Software compensation technique
- Implementation of software compensation into Pandora
- Some results with ILD detector model

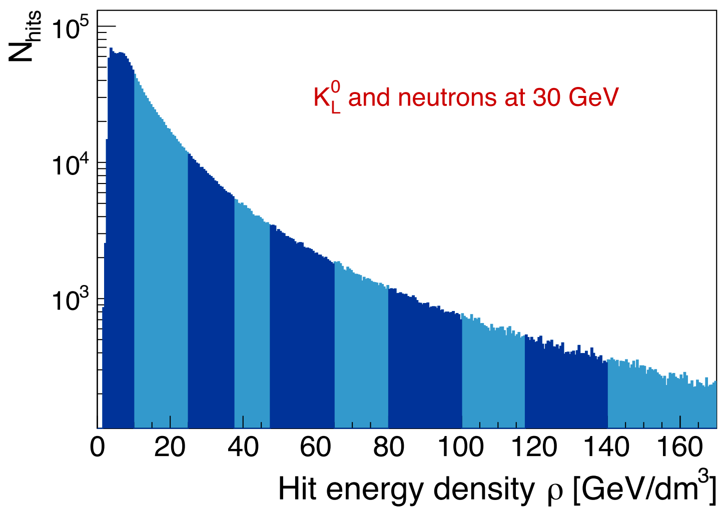
Particle Flow reconstruction & Software compensation

- Particle Flow reconstruction: reconstruct individual particles
  - Need precise measurement of particle's energy with calorimeters
- ILD calorimeters are non-compensating: degrades energy resolution
  - Up to now, simple compensation using cell energy truncation)
- But ILD calorimeters are highly granular:
  - Allow assessment at sub-shower level for electromagnetic and hadronic **sub-shower distinction** for software compensation



Software compensation

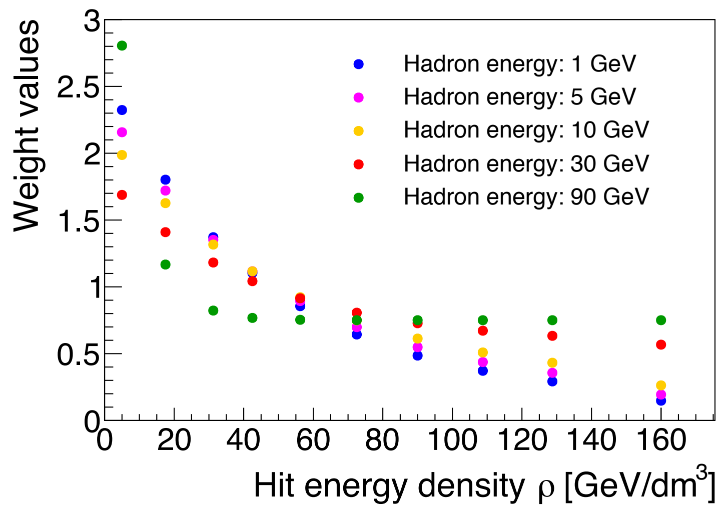
- **Software compensation** technique by CALICE: weighting hit energy according to its energy density



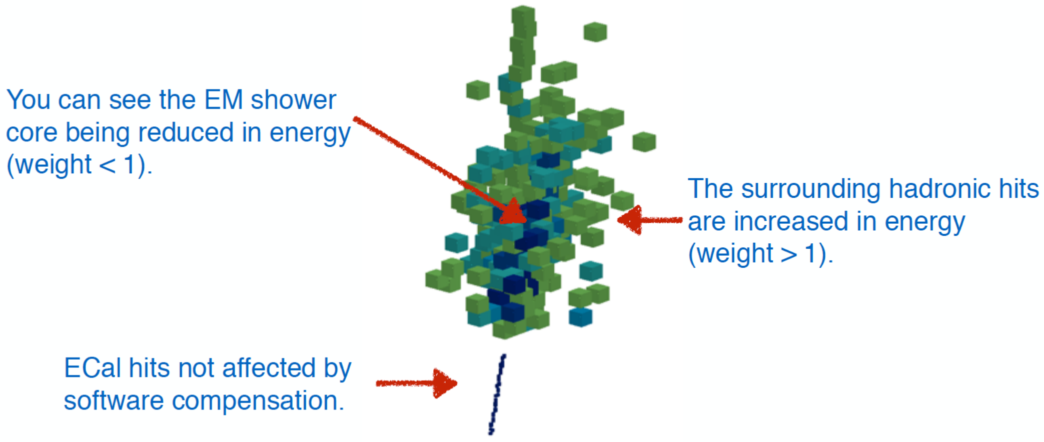
$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$

$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{bin} (E^{bin}_{HCAL} \times \omega_{bin}(\rho))$$

$$with\ E^{bin}_{HCAL} = \sum_{hits \in bin} E_{hit}$$



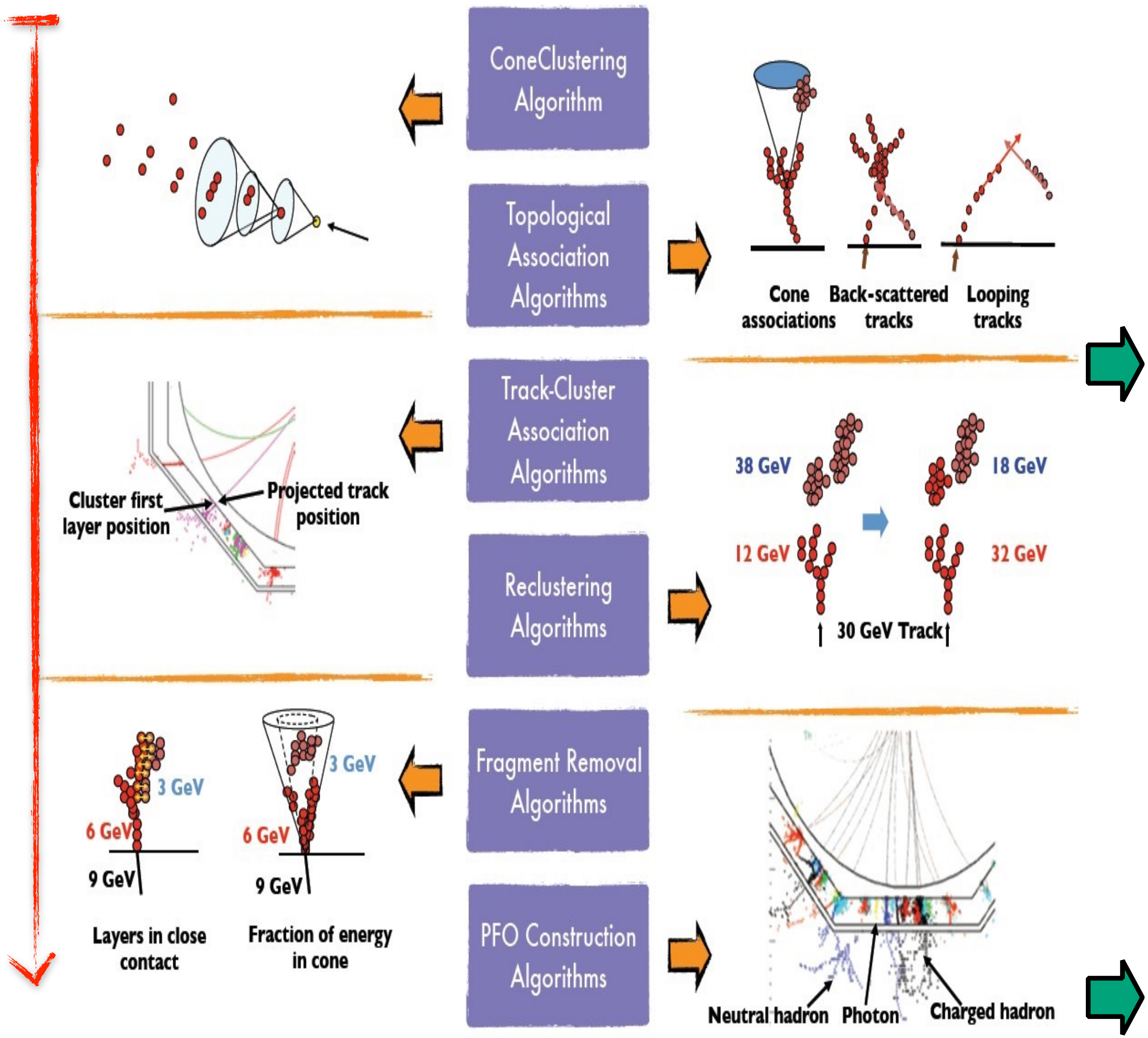
Example on software compensation's operation



Coloured in by weight applied in software compensation. Cluster in 91 GeV jet.

Blue: Low Weight
Green: High Weight

Software compensation application



First set of clusters obtained

- Clusters without track: neutral particles, fragment, …
- Clusters with associated track: cluster-track energy comparison. **Crucial** as it decides how good the energy reconstruction will be

➪ Software compensation for all clusters

OR

Software compensation for neutral hadrons

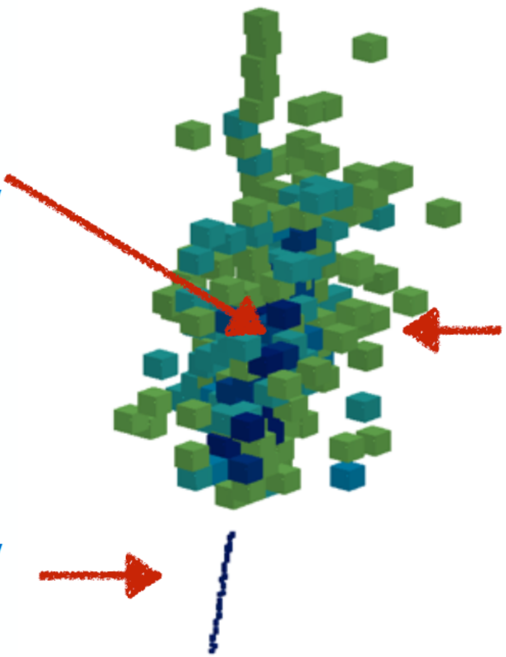
Software version and configuration

- **Detector model**: ILD_o1_v06
- **Reconstruction software**: ilcsoft_v01-17-07 combined with PandoraPFA version v02-09-00:
  - PandoraSDK v02-03-01
  - LCContent v02-04-00 including software compensation in LCPlugins and hits information registration for software compensation weight training in LCUtility
  - PandoraMonitoring v02-03-00

- **Digitiser**: ILDCaloDigi with realistic options for ECAL and HCAL
- **Calibration constants** optimised using PandoraAnalysis toolkit
- **Timing cut**: 100 *ns*

Final software compensation implementation

- **<u>Visual Pandora:</u> PandoraMonitoring v02-03-00**
    - New functionality in cell algorithm (visualise hit and cluster energy) allow colouring hits in a cluster according to weight applied in software compensation
    - Thanks to Steven Green



Coloured in by weight applied in software compensation. Cluster in 91 GeV jet.

Blue: Low Weight
Green: High Weight

Final software compensation implementation

- **<u>Setting in Pandora:</u> SC enabled by default in PandoraSettingsDefault.xml**
  - Software compensation weights for standard ILD detector are used by default
  - All variables are steerable

```xml
<!-- PLUGIN SETTINGS -->
<HadronicEnergyCorrectionPlugins>SoftwareCompensation</HadronicEnergyCorrectionPlugins>
<EmShowerPlugin>LCEmShowerId</EmShowerPlugin>
<PhotonPlugin>LCPhotonId</PhotonPlugin>
<ElectronPlugin>LCElectronId</ElectronPlugin>
<MuonPlugin>LCMuonId</MuonPlugin>
```

- Paper written on this: first draft done

Software compensation in Particle Flow reconstruction

Huong Lan Tran, Katja Krüger, Felix Sefkow
Deutsches Elektronen-Synchrotron DESY, Hamburg, Germany

Steven Green, John Marshall, Mark A. Thomson
Cavendish Laboratory, Cambridge, United Kingdom

Frank Simon
Max-Planck-Institut für Physik, Munich, Germany

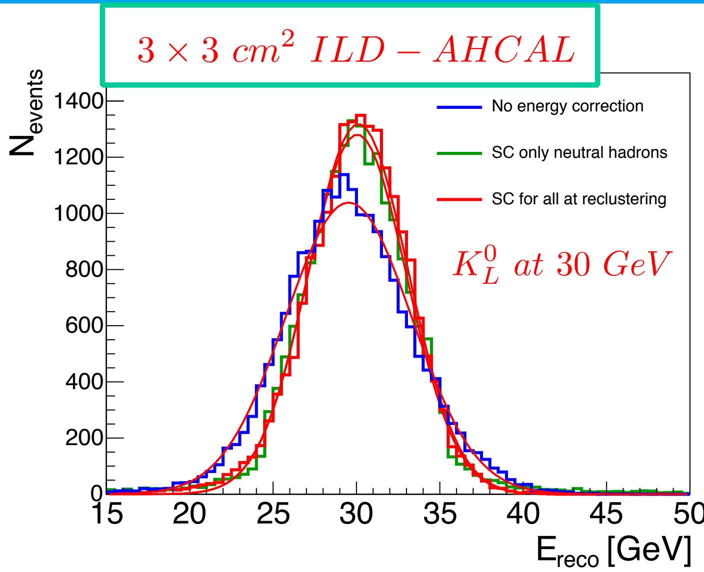
High calorimeter granularity is the prerequisite requirement for particle separation in Particle Flow reconstruction. It can be further utilised in the so-called *software compensation technique*, in which it provides a discrimination of the electromagnetic sub-showers in hadron showers and therefore improves the energy resolution for single particles. This improvement in the single particle energy resolution can then lead to a better jet energy resolution. This paper describes the software compensation technique and its implementation in Particle Flow reconstruction. The impact of the software compensation on the cell size optimisation for the hadronic calorimeter of the International Large Detector (ILD) is also discussed.

CONTENTS

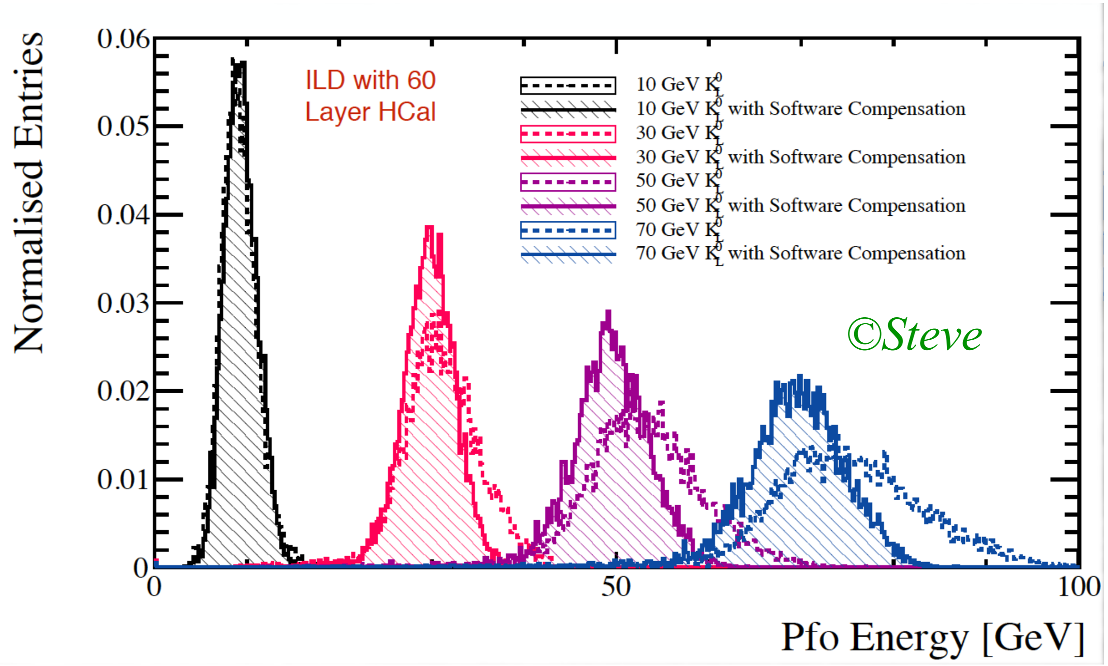
I. INTRODUCTION

A. The particle flow approach to calorimetry and the PandoraPFA framework

Energy resolution with software compensation

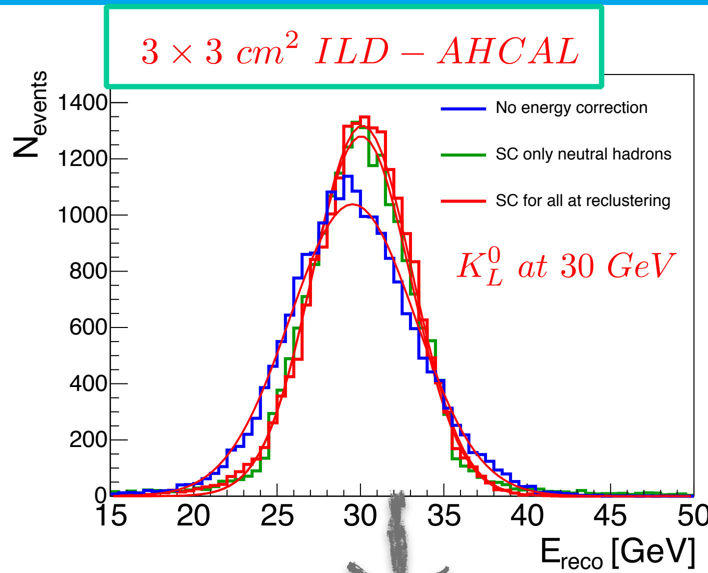


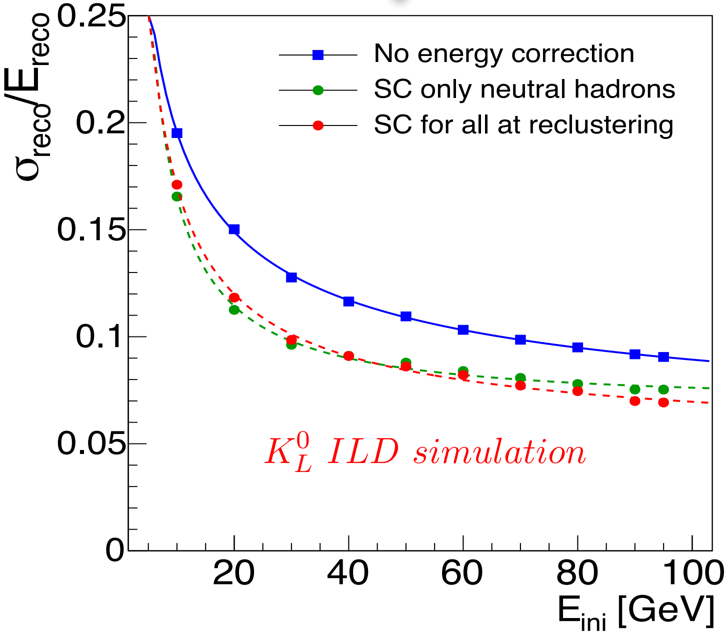
- Software compensation benefits in two-fold way:
  - Improve energy reconstruction of neutral objects
  - Improve cluster energy estimator for better track-cluster association ➢ confusion mitigation



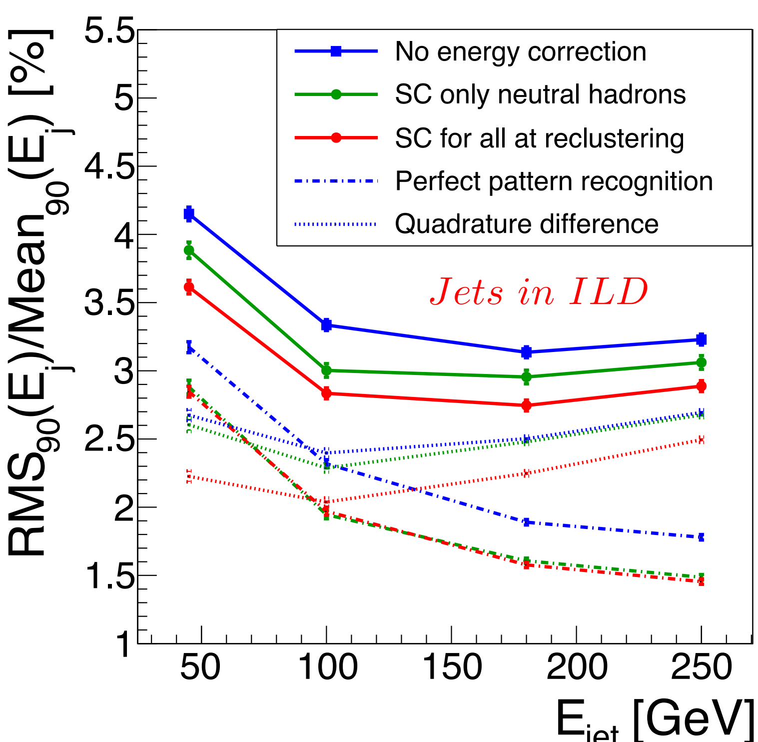
Study with 60 layer HCAL and higher jet energies (relevant for CLIC studies)

Energy resolution with software compensation



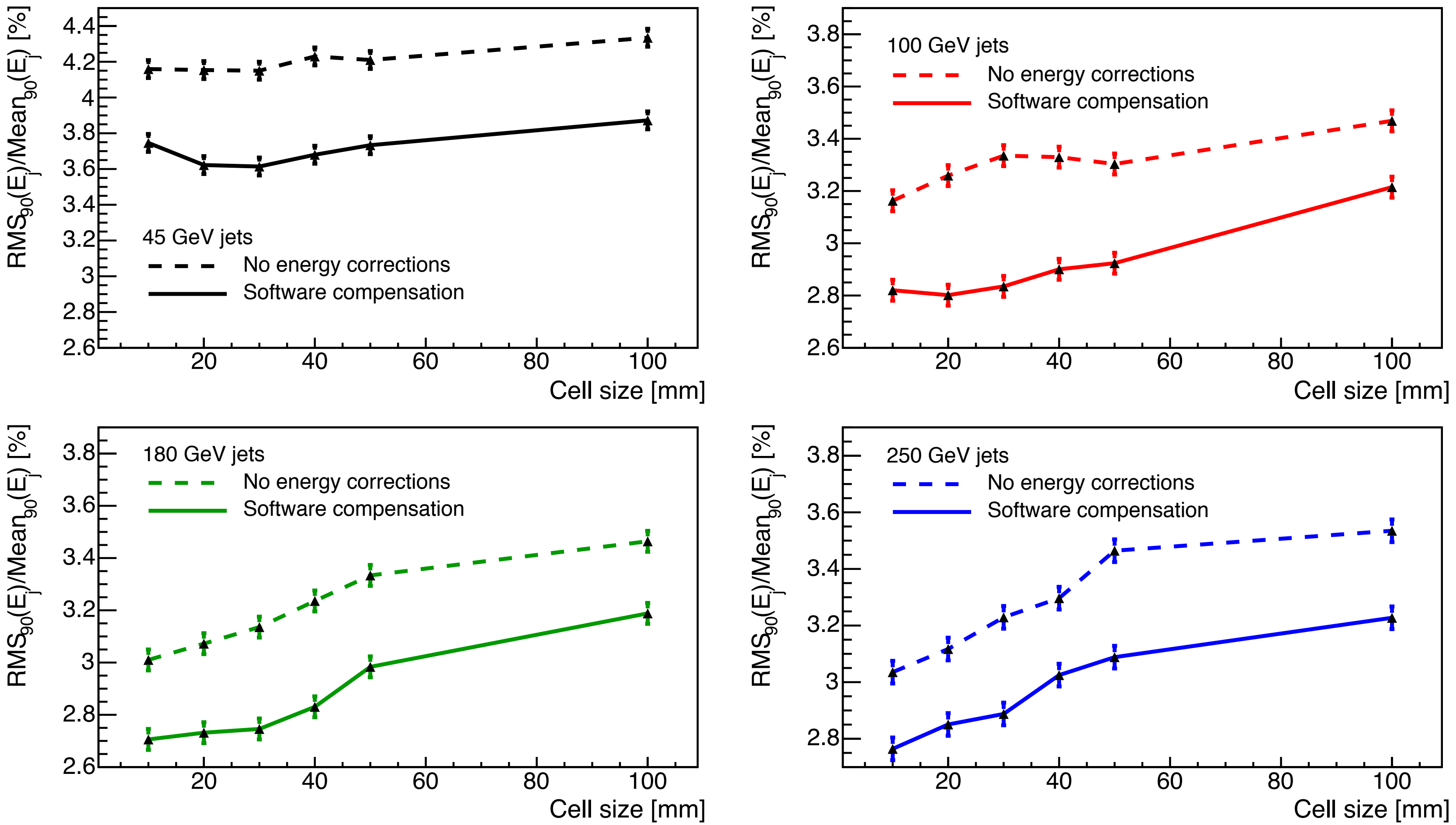


- Software compensation benefits in two-fold way:
  - Improve energy reconstruction of neutral objects
  - Improve cluster energy estimator for better track-cluster association ➢ confusion mitigation
- Significant improvement at both single particle and jet level
- Software compensation applied at re-clustering stage more beneficial for jet energy resolution



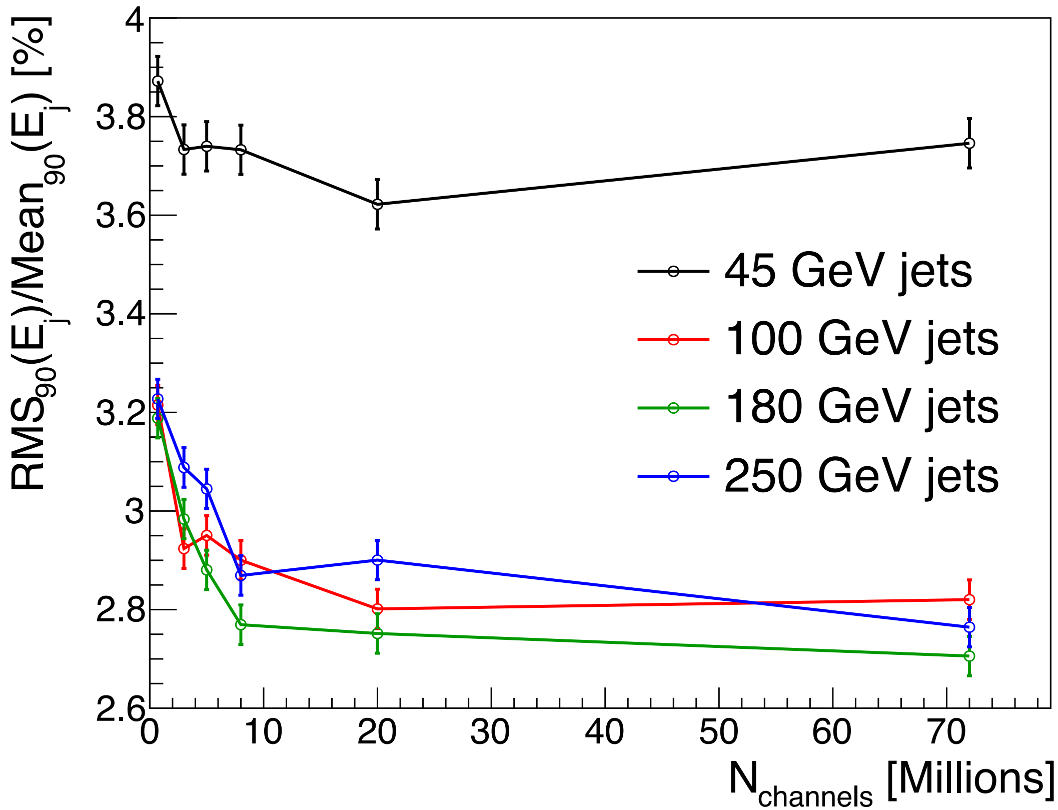
JER vs cell size

- Effectiveness of software compensation depends on granularity
  - Software compensation included in cell size optimisation
  - Weights optimised for each cell size



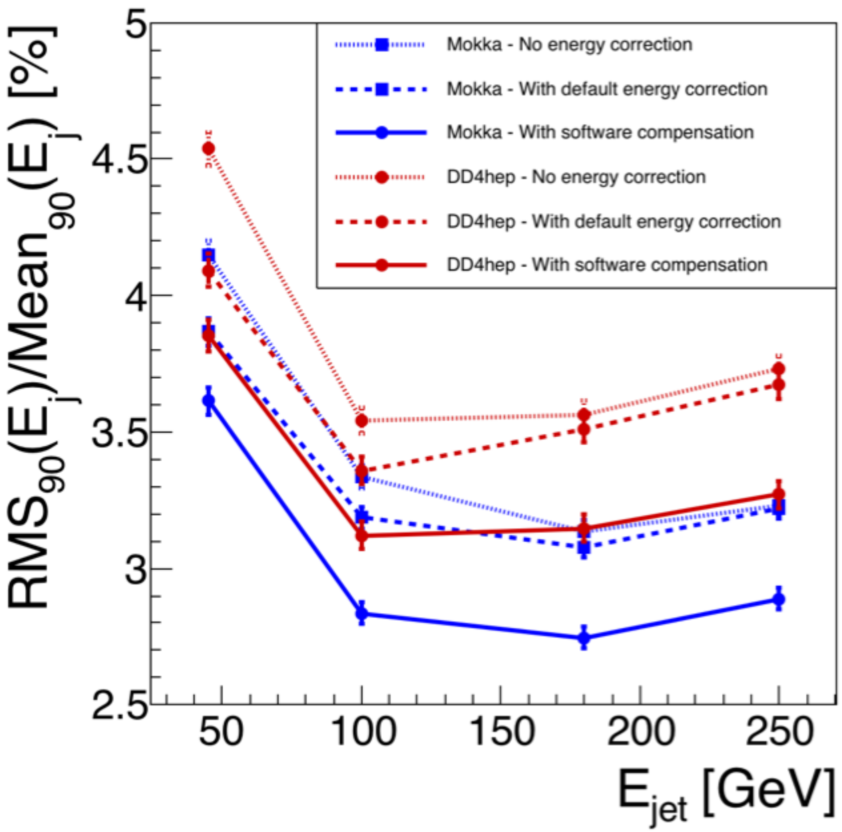
JER vs number of cells

- Jet energy resolution plotted as a function of number of HCAL cells
  - Towards cost optimisation
  - 3 x 3 cm^2 cell size is still a very reasonable choice



Towards DD4hep simulation

- Software compensation is now applied for DD4hep simulation
  - Over all JER with DD4hep at the right scale, still worse compared to Mokka
  - Applying software compensation or default energy corrections does improve JER, the size of improvement is compatible to what obtained in Mokka



Software compensation and semi-digital reconstruction

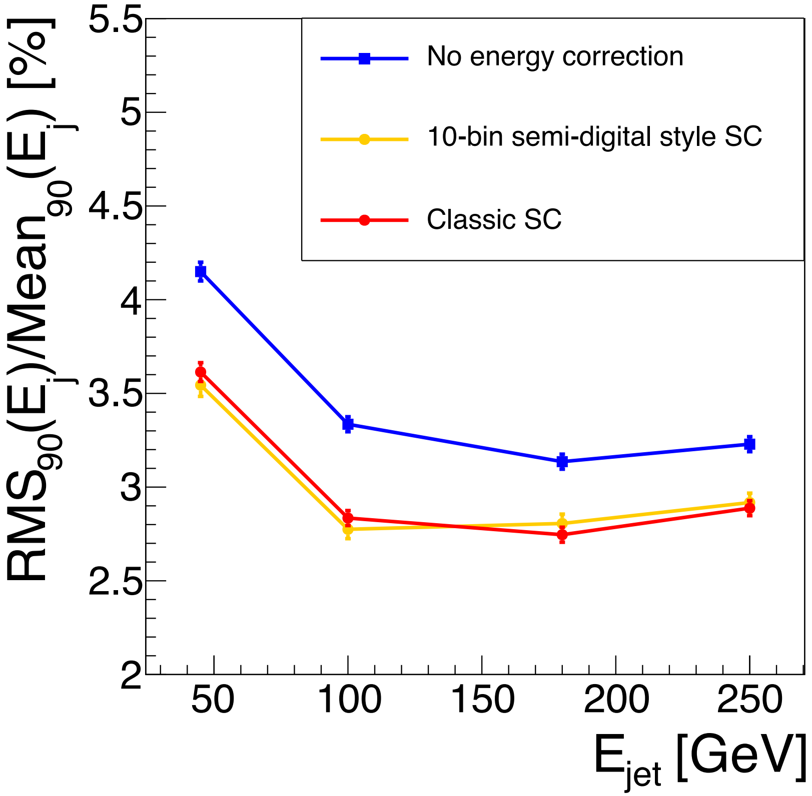
- Semi-digital reconstruction is particularly successful at low energies
  - Counting hits at 3 thresholds N_1, N_2, N_3
  - Suppress Landau fluctuations
- Reconstructed energy: $E_{SD} = \sum_{bins} \alpha_i . N_i$

or

$$E_{SD} = \sum_{hits} \alpha_i . \frac{E_j}{E_j} = \sum_{hits} \omega_j . E_j \text{ with } \omega_j = \frac{\alpha_i}{E_j}$$

- Can apply software compensation using the **same formalism**

$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{bin} (\alpha + \beta E_{sum} + \gamma E_{sum}^2) \times E_{HCAL}^{bin}$$



- Gives results consistent with classic software compensation
  - Number of bin and binning definition steerable
- Allows for semi-digital reconstruction in the same framework for direct comparison by introducing $1/E_j$ factor (counting hits) - not shown here, $3x3cm^2$ case

Summary

- Jet energy resolution with software compensation in Pandora:
  - Significant gain in performance over a wide jet energy range, best performance so far achieved for ILD detector
  - Inclusion of SC does not significantly alter view on cell size optimisation

- Software compensation code and utilities in latest version of PandoraPFA
- Includes semi-digital reconstruction scheme

- Installed in new ILCsoft v01-17-10
  - Software compensation applied to DD4hep simulation shows the same improvement on jet performance

- Study summarised in a paper, soon to be on review

Back-up slides

Software Compensation in AHCAL optimisation

- ***Idea:*** Applying different weights for hits of different energy densities
- ***Weight*** defined as:

$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$

where ρ is hit energy density, p_1, p_2, p_3 are *beam energy dependent parameters*

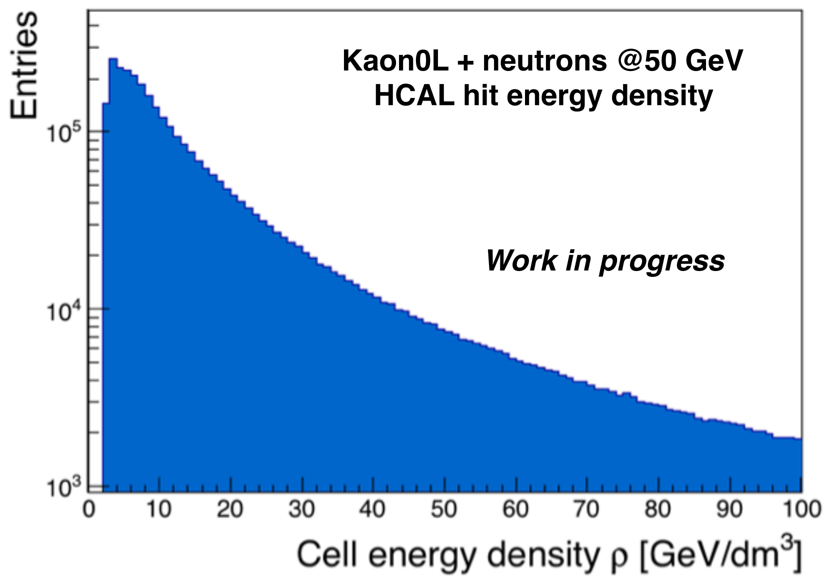
- Energy of cluster then computed in software compensation method as:

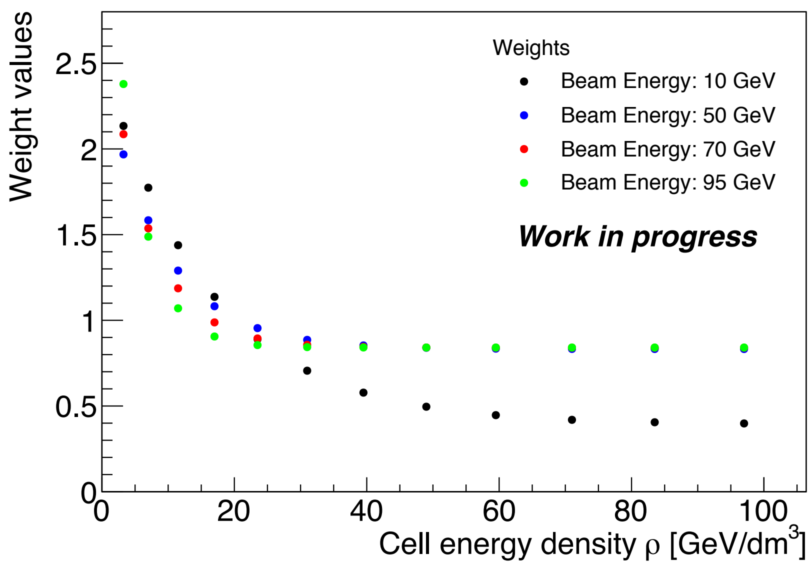
$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{hits} (E_{HCAL}.\omega(\rho))$$

- Weights determined through minimising a χ^2 function:

$$\chi^2 = \sum_{events} (E_{SC} - E_{beam})^2$$

Hit Energy Density and Weights





Weight determination:

- Through χ^2 minimisation
- For each beam energy weights are defined with three parameters p_1, p_2, p_3

$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$

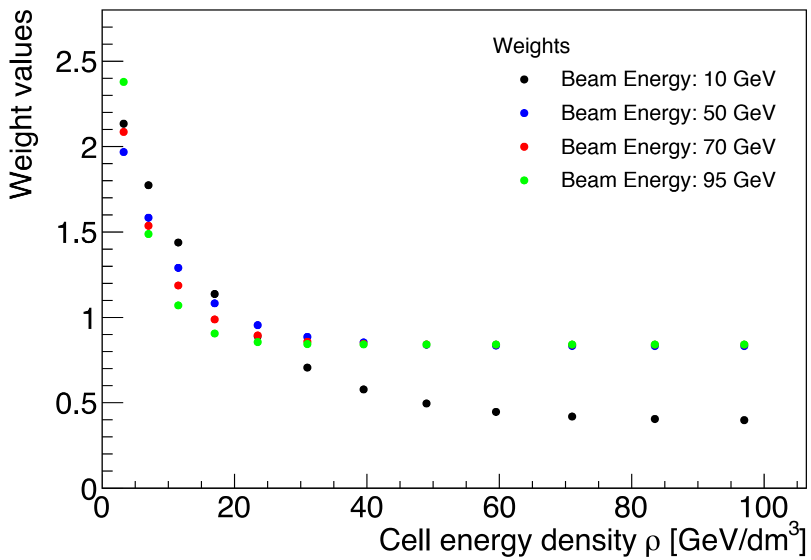
where p_1, p_2, p_3 are energy dependent parameter (defined directly in χ^2)

$$p_1 = p_{10} + p_{11} \times E_{ini} + p_{12} \times E_{ini}^2$$

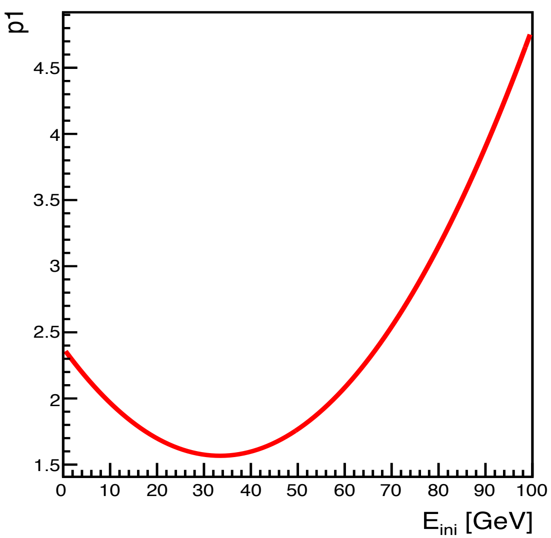
$$p_2 = p_{20} + p_{21} \times E_{ini} + p_{22} \times E_{ini}^2$$

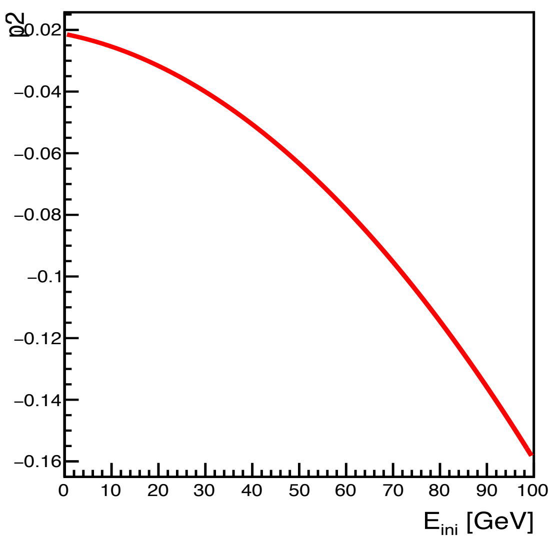
$$p_3 = \frac{p_{30}}{p_{31} + e^{p_{32} \times E_{ini}}}$$

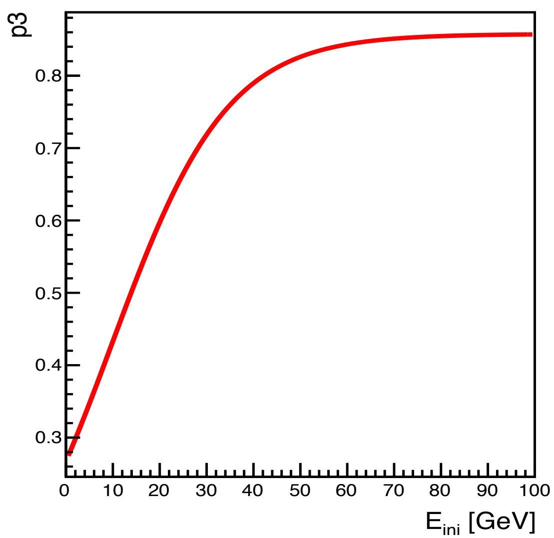
Weight parameters



$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$

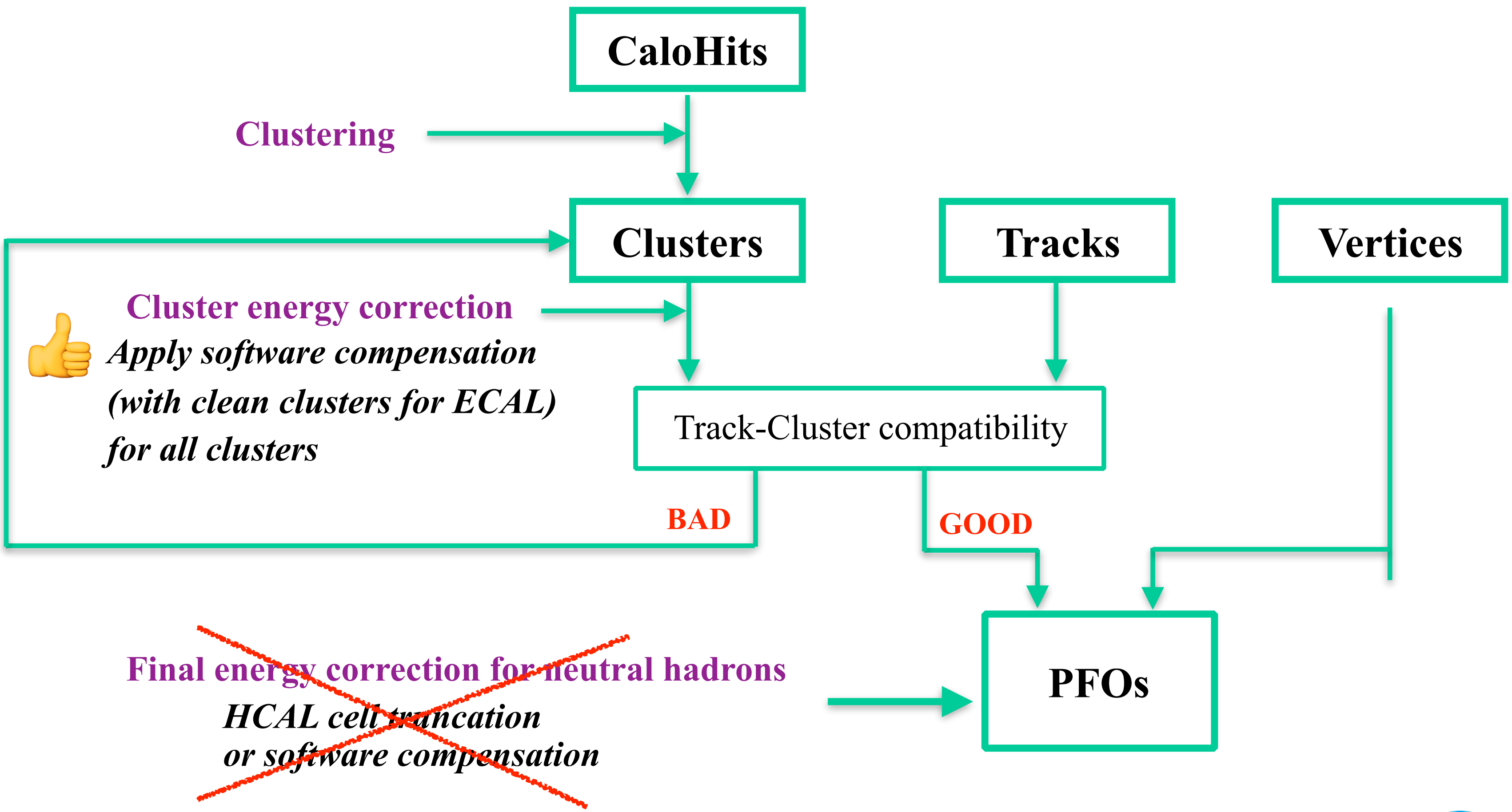






Final software compensation implementation

- Software compensation now applied **all-at-once** in re-clustering step



Software compensation in PFlow reconstruction

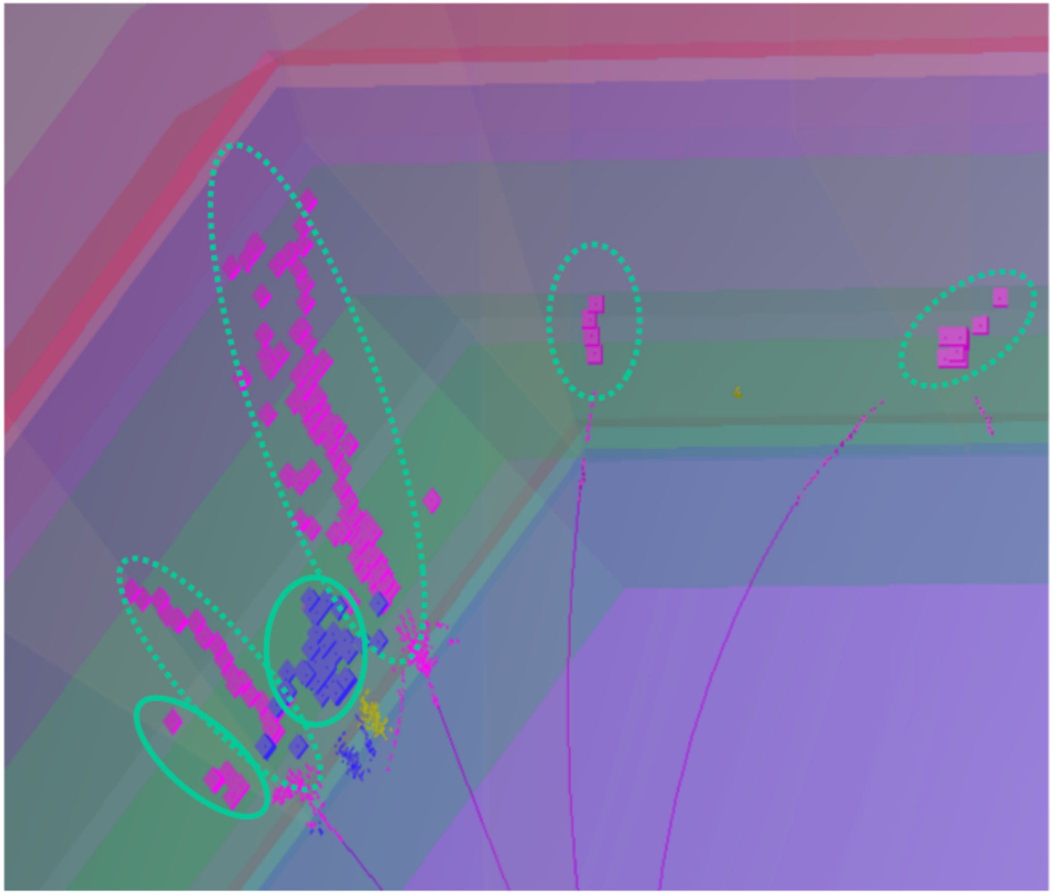
<u>First set of clusters</u>

No associated tracks
- Neutral particles
- Fragments
- To be merged in other clusters

Has associated tracks
- Cluster energy will be compared to track energy re-clustering if not compatible

➢ Software compensation applied for ***only clusters with associated tracks*** for energy comparison

In principle ***can be applied for both type*** of clusters



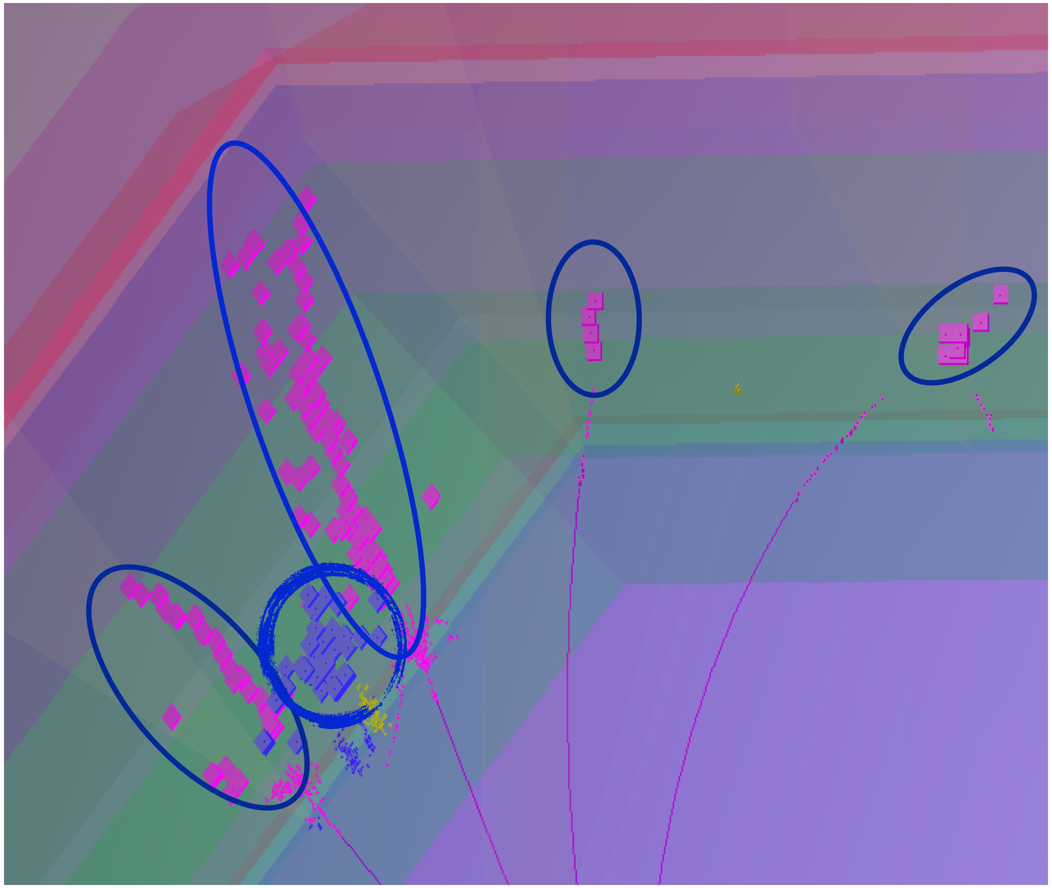
Software compensation in PFlow reconstruction

<u>First set of clusters</u>

No associated tracks
- Neutral particles
- Fragments
- To be merged in other clusters

Has associated tracks
- Cluster energy will be compared to track energy re-clustering if not compatible

➢ Software compensation applied for ***only clusters with associated tracks*** for energy comparison
In principle ***can be applied for both type*** of clusters

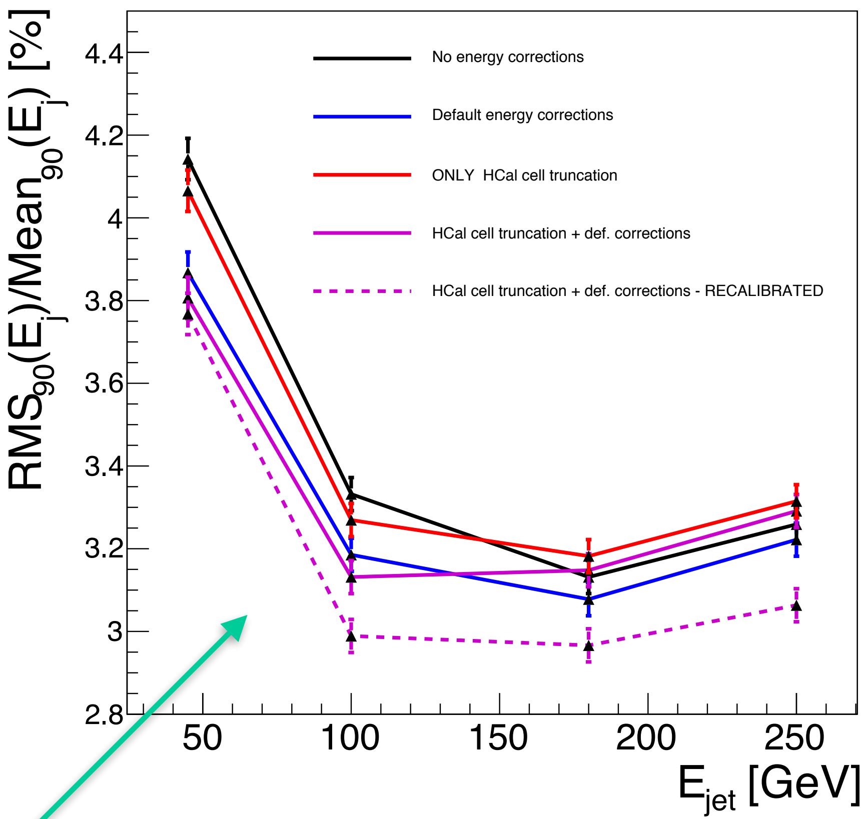


<u>Particle Flow Objects</u>

➢ Software compensation applied for ***neutral PFOs***
- Stand-alone: sometimes better, sometimes worse compared to Steve's results with truncation
- When combine with application at re-clustering for clusters with associated tracks: improvement at high energies but degradation at small energies

HCAL cell truncation performance

- HCAL cell truncation is not applied ALONE: Two default energy correction plugins are turned on
  - ***CleanClusters***: clean hot hits in ECAL track, which is quite effective in *low energy range* for clusters which are largely contained in ECAL
  - ***ScaleHotHadrons***: some sort of simple software compensation for clusters that have up to 100 hits (affects low energy range)



- Study to separate the effect of CleanClusters +ScaleHotHadrons and HCAL cell truncation

CleanClusters energy correction

- CleanClusters mainly affects ECAL clusters by changing energy significantly (remove hot cell)

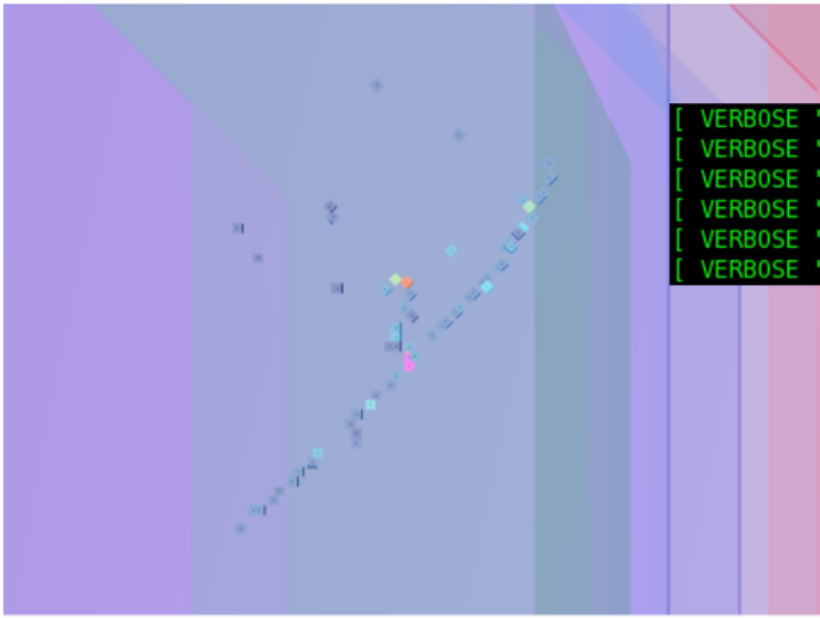
```
[ VERBOSE "MyMarlinPandoraDefault"] Software compensation changing energy : 1
[ VERBOSE "MyMarlinPandoraDefault"] Number of hits in cluster    : 40
[ VERBOSE "MyMarlinPandoraDefault"] Hadronic Energy of Cluster   : 3.79469
[ VERBOSE "MyMarlinPandoraDefault"] Corrected Energy of Cluster : 1.74793
[ VERBOSE "MyMarlinPandoraDefault"] NECalHits                   : 40
[ VERBOSE "MyMarlinPandoraDefault"] NHCalHits                   : 6
```

Clean Cluster Changing Energy Significantly

```
[ VERBOSE "MyMarlinPandoraDefault"] 0x1c6a140
[ VERBOSE "MyMarlinPandoraDefault"] energyInPreviousLayer          : 0.0160862
[ VERBOSE "MyMarlinPandoraDefault"] energyInNextLayer              : 0.0199764
[ VERBOSE "MyMarlinPandoraDefault"] energyInCurrentLayer           : 2.45802
[ VERBOSE "MyMarlinPandoraDefault"] energyInAdjacentLayers         : 0.0180313
[ VERBOSE "MyMarlinPandoraDefault"] hitHadronicEnergy              : 1.06995
[ VERBOSE "MyMarlinPandoraDefault"] Min Clean Hit Energy           : 1
[ VERBOSE "MyMarlinPandoraDefault"] Fraction of cluster energy in hit : 0.281959
[ VERBOSE "MyMarlinPandoraDefault"] m_minCleanHitEnergyFraction    : 0.2
[ VERBOSE "MyMarlinPandoraDefault"] newHitHadronicEnergy(energyInAdjacentLayers - energyInCurrentLayer + hitHadronicEnergy) : 0.2
[ VERBOSE "MyMarlinPandoraDefault"] energy change (newHitHadronicEnergy - hitHadronicEnergy)                                : -0.869948
[ VERBOSE "MyMarlinPandoraDefault"] Changing the energy...
[ VERBOSE "MyMarlinPandoraDefault"] energyInPreviousLayer          : 0.0160862
[ VERBOSE "MyMarlinPandoraDefault"] energyInNextLayer              : 0.0199764
[ VERBOSE "MyMarlinPandoraDefault"] energyInCurrentLayer           : 2.45802
[ VERBOSE "MyMarlinPandoraDefault"] energyInAdjacentLayers         : 0.0180313
[ VERBOSE "MyMarlinPandoraDefault"] hitHadronicEnergy              : 1.37682
[ VERBOSE "MyMarlinPandoraDefault"] Min Clean Hit Energy           : 1
[ VERBOSE "MyMarlinPandoraDefault"] Fraction of cluster energy in hit : 0.362827
[ VERBOSE "MyMarlinPandoraDefault"] m_minCleanHitEnergyFraction    : 0.2
[ VERBOSE "MyMarlinPandoraDefault"] newHitHadronicEnergy(energyInAdjacentLayers - energyInCurrentLayer + hitHadronicEnergy) : 0.2
[ VERBOSE "MyMarlinPandoraDefault"] energy change (newHitHadronicEnergy - hitHadronicEnergy)                                : -1.17682
[ VERBOSE "MyMarlinPandoraDefault"] Changing the energy...
[ VERBOSE "MyMarlinPandoraDefault"] This cluster requires an energy correction according to the CLEAN CLUSTERS logic.
[ VERBOSE "MyMarlinPandoraDefault"] Number of hadronic plugins registered: 2
[ VERBOSE "MyMarlinPandoraDefault"] 0x1c6a0a0
```

CleanClusters energy correction

- CleanClusters mainly affects ECAL clusters by changing energy significantly (remove hot cell)



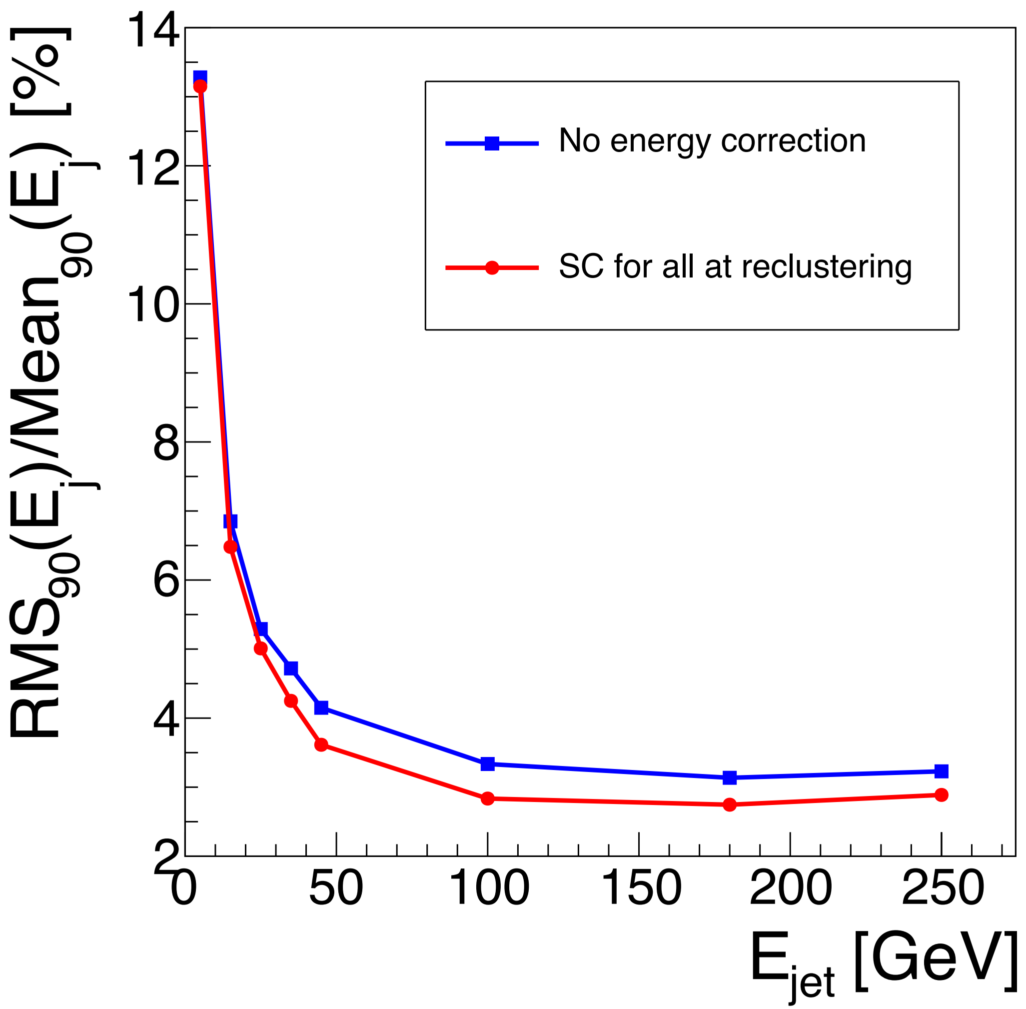
```
[ VERBOSE "MyMarlinPandoraDefault"] Software compensation changing energy : 1
[ VERBOSE "MyMarlinPandoraDefault"] Number of hits in cluster   : 59
[ VERBOSE "MyMarlinPandoraDefault"] Hadronic Energy of Cluster  : 3.67985
[ VERBOSE "MyMarlinPandoraDefault"] Corrected Energy of Cluster : 1.96347
[ VERBOSE "MyMarlinPandoraDefault"] NECalHits                   : 70
[ VERBOSE "MyMarlinPandoraDefault"] NHCalHits                   : 0
```

Clean Cluster Changing Energy Significantly

➢ **Use CleanClusters for ECAL clusters**

```
VERBOSE "MyMarlinPandoraDefault"] energyInPreviousLayer              : 0.151975
VERBOSE "MyMarlinPandoraDefault"] energyInNextLayer                  : 0.170483
VERBOSE "MyMarlinPandoraDefault"] energyInCurrentLayer               : 2.29877
VERBOSE "MyMarlinPandoraDefault"] energyInAdjacentLayers             : 0.161229
VERBOSE "MyMarlinPandoraDefault"] hitHadronicEnergy                  : 1.91638
VERBOSE "MyMarlinPandoraDefault"] Min Clean Hit Energy               : 1
VERBOSE "MyMarlinPandoraDefault"] Fraction of cluster energy in hit : 0.520776
VERBOSE "MyMarlinPandoraDefault"] m_minCleanHitEnergyFraction        : 0.2
VERBOSE "MyMarlinPandoraDefault"] newHitHadronicEnergy(energyInAdjacentLayers - energyInCurrentLayer + hitHadronicEnergy) : 0.2
VERBOSE "MyMarlinPandoraDefault"] energy change (newHitHadronicEnergy - hitHadronicEnergy)                                 : -1.71638
```

JER with small energies (3x3cm2 HCAL)



Software compensation and semi-digital reconstruction

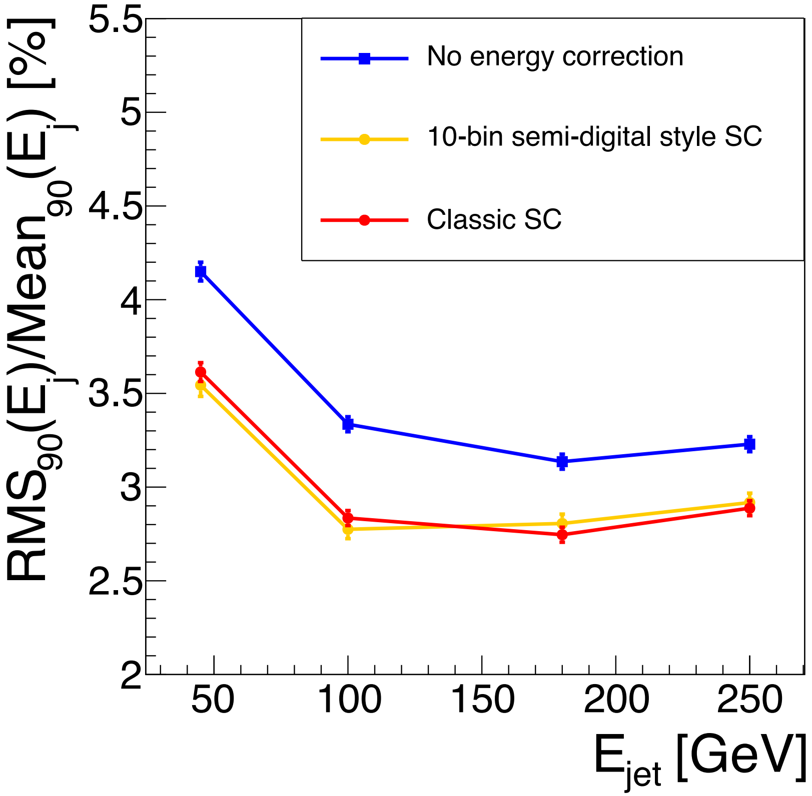
- Semi-digital reconstruction is particularly successful at low energies
  - Counting hits at 3 thresholds N_1, N_2, N_3
- Reconstructed energy: $E_{SD} = \sum_{bins} \alpha_i . N_i$

or

$$E_{SD} = \sum_{hits} \alpha_i . \frac{E_j}{E_j} = \sum_{hits} \omega_j . E_j \text{ with } \omega_j = \frac{\alpha_i}{E_j}$$

- Software compensation can also apply the **same formalism** keeping **10 bin definition** of classic SC:

$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{bin} (\alpha + \beta E_{sum} + \gamma E_{sum}^2) \times E_{HCAL}^{bin}$$



- Give compatible results to classic software compensation
  - Number of bin and binning definition steerable
  - Allow semi-digital reconstruction in the same framework for direct comparison