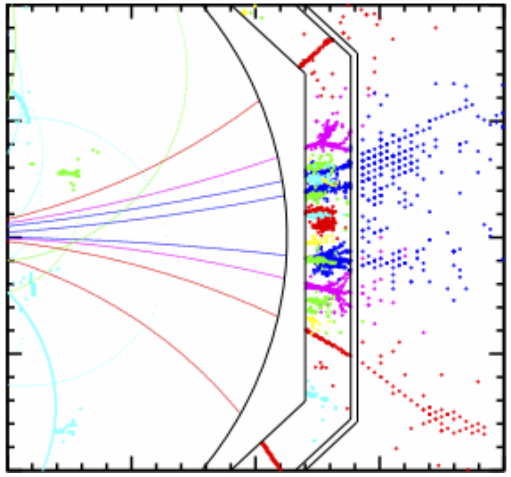
Software Compensation in Particle Flow reconstruction



Huong Lan Tran

Linear Collider Forum - 22 November 2016

10th Workshop on Physics at the Terascale

HELMHOLTZ ASSOCIATION

CALICE

DESY

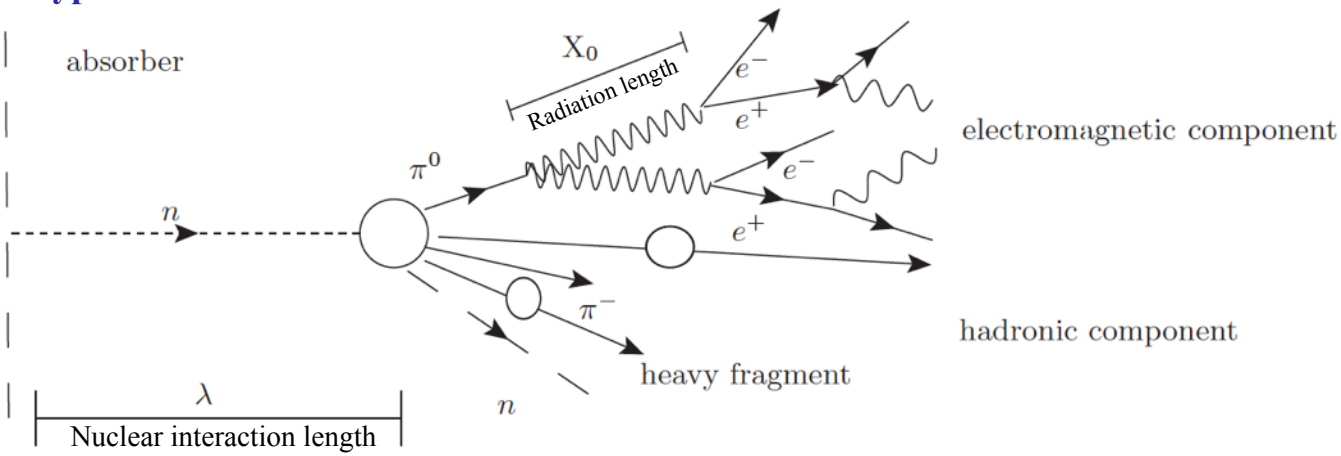
Outlines

- Motivation for software compensation
- Particle flow reconstruction & Software compensation (SC)
- Implementation of SC into Particle flow reconstruction
- Application of SC for ILD detector performance study
  - AHCAL transverse granularity optimisation

Motivation for Software compensation

- ILD calorimeters are ***non-compensating***

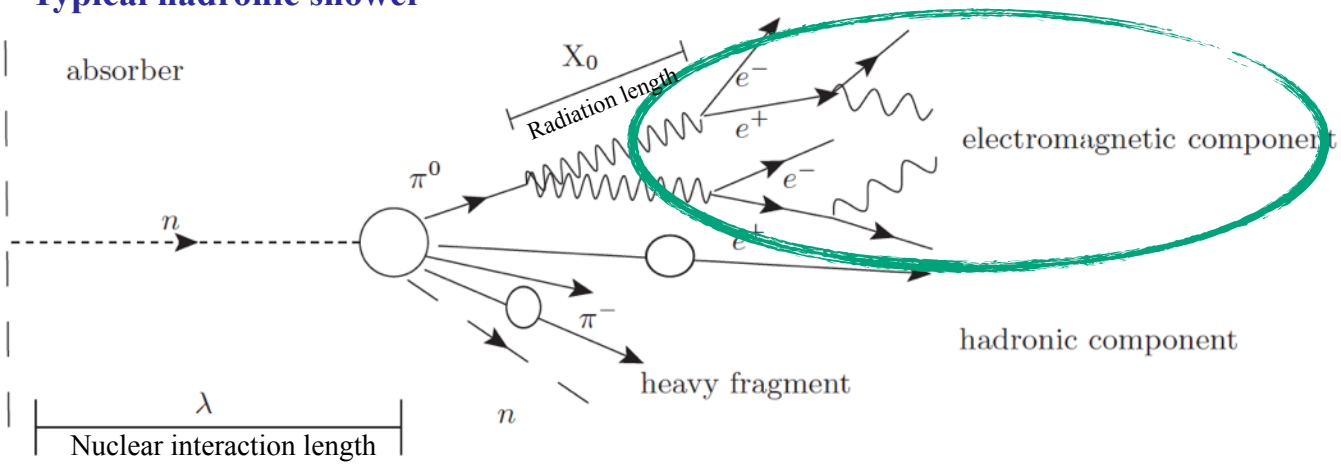
Typical hadronic shower



Motivation for Software compensation

- ILD calorimeters are ***non-compensating***

Typical hadronic shower

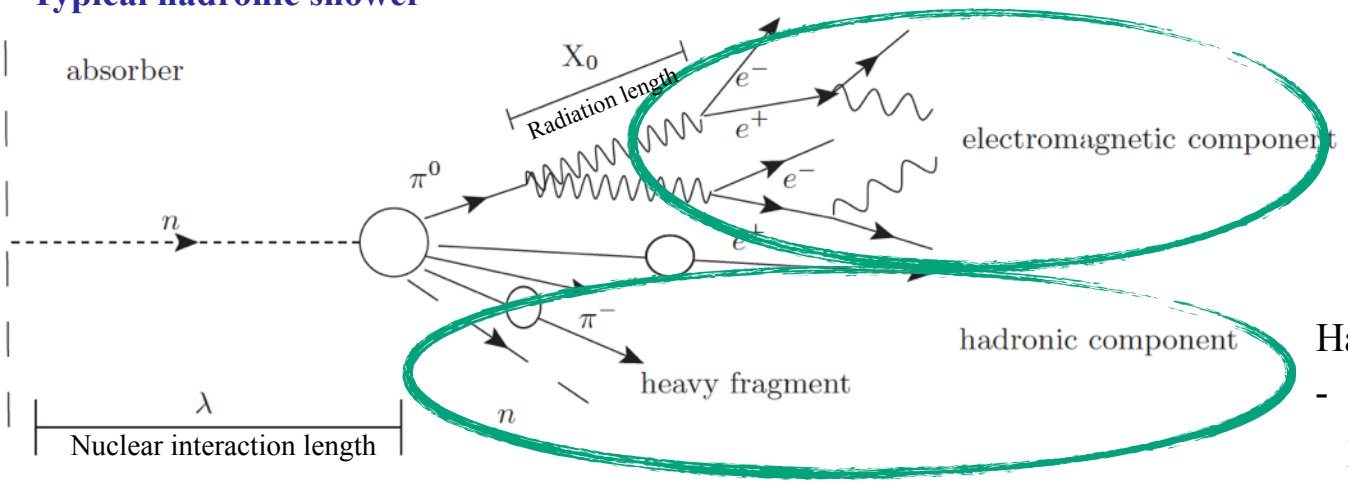


Detected via energy loss of electrons and photons in active medium

Motivation for Software compensation

- ILD calorimeters are ***non-compensating***

Typical hadronic shower



Detected via energy loss of electrons and photons in active medium

Hadronic components:

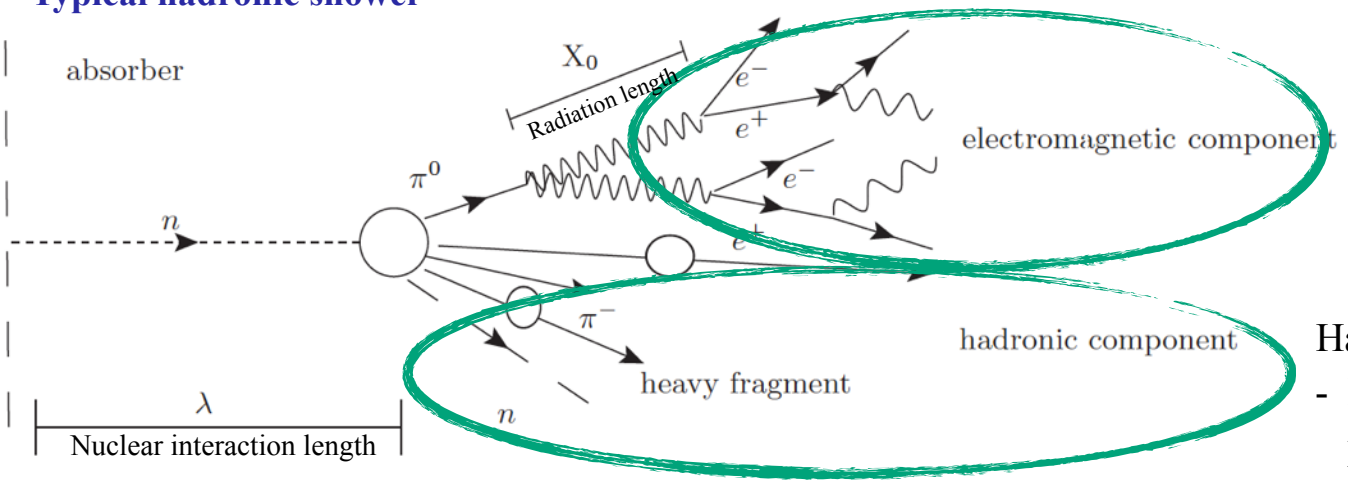
- Detectable ionisation processes of charged hadrons
- *Invisible energy*: nuclear binding energy or target recoil

⇨ Smaller calorimeter response for this part

Motivation for Software compensation

- ILD calorimeters are ***non-compensating***

Typical hadronic shower



Detected via energy loss of electrons and photons in active medium

Hadronic components:
- Detectable ionisation processes of charged hadrons
- *Invisible energy*: nuclear binding energy or target recoil

➪ Smaller calorimeter response for this part

➢ **Consequences:**

- Higher detector response for electromagnetic compared to hadronic showers $\frac{e}{h} > 1$
- Non-linearity for hadronic calorimeter response
- Degradation of energy resolution

Methods to achieve Compensation

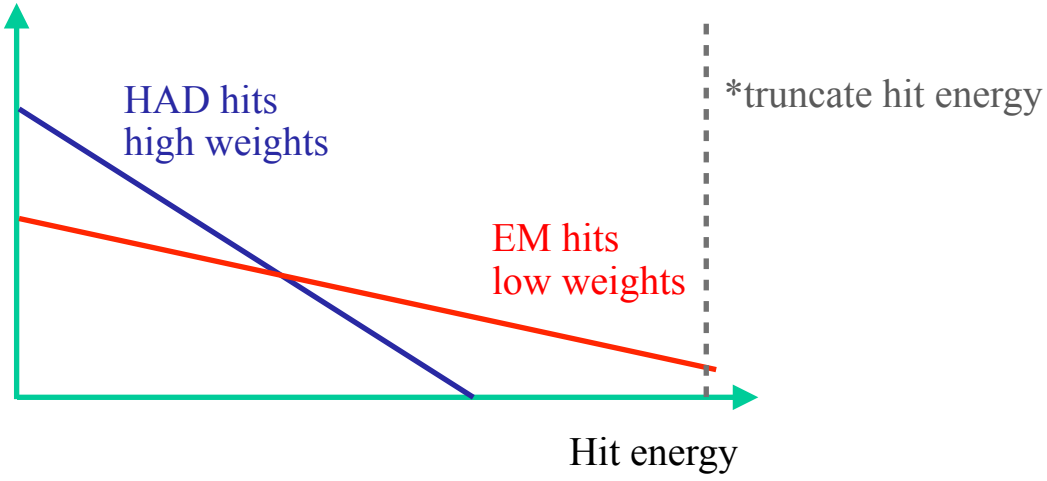
- Reducing electromagnetic response
- Increasing hadronic response

Achievable with detector design

- "Offline" compensation: **Software Compensation**
  - Electromagnetic showers denser than hadronic showers ➢ energy of hits inside electromagnetic sub-showers are typically higher compared to hits inside hadronic sub-showers.

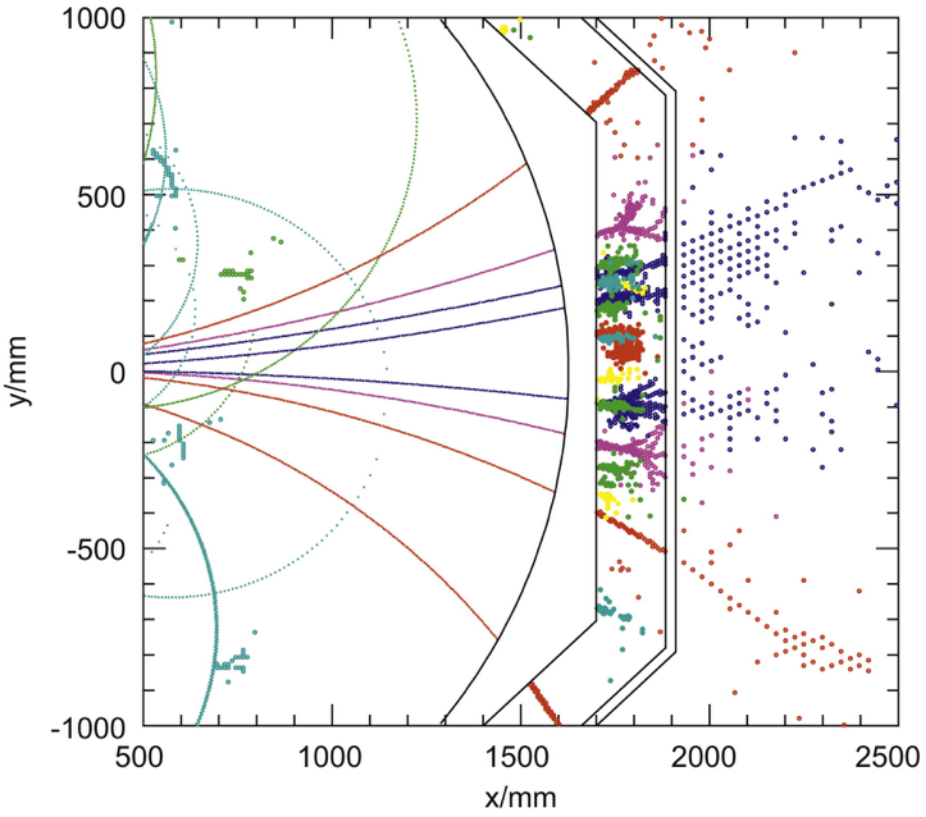
    ➢ Cut out high energy hits to reduce EM response *

    ➢ Applying different weights for hits of different energy densities



Particle Flow reconstruction & Software compensation

- Particle Flow reconstruction: trace individual particles
  - Need **precise measurement of particle's energy** with calorimeters
- ILD calorimeters are non-compensating: degrade energy resolution
  - Compensation with electromagnetic response truncation (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

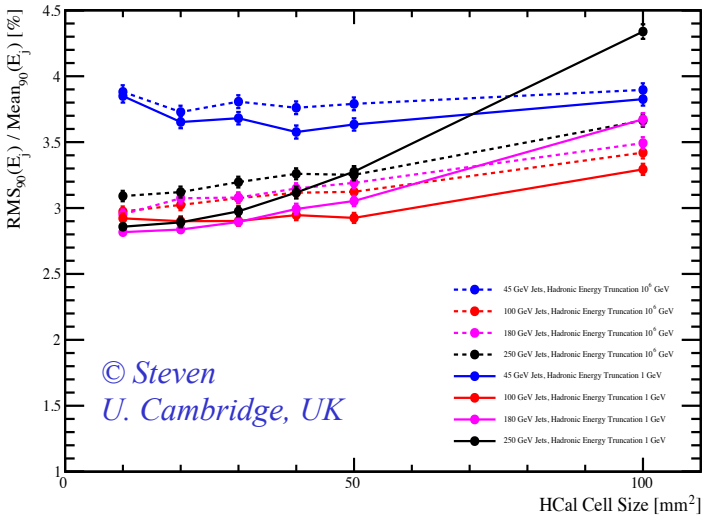


Particle Flow reconstruction & Software compensation

- Dependence of jet energy resolution on HCAL cell size apparently reduced compared to results from LoI (both study use PandoraPFA):
  - *HCAL cell energy truncation* degrades resolution at high energy for higher cell size
  - But: improve energy resolution at smaller cell sizes

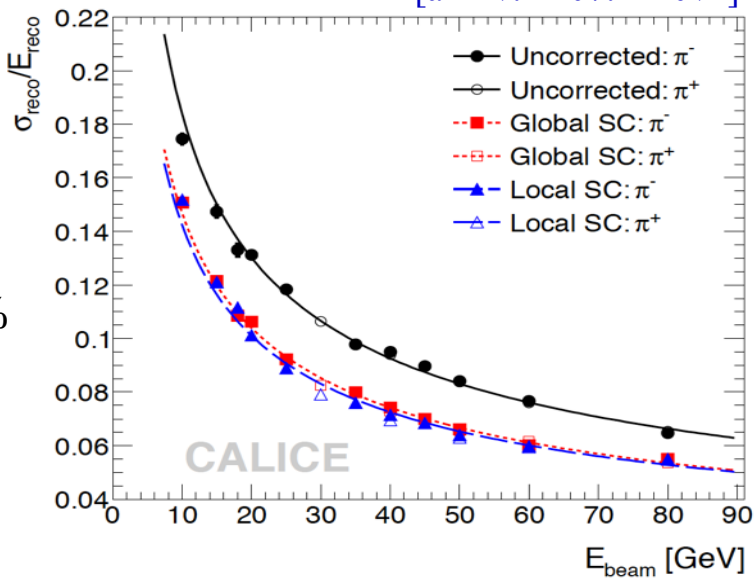
➢ Cell energy truncation mimics software compensation

➢ Software compensation can do better and must be applied properly in Particle Flow reconstruction



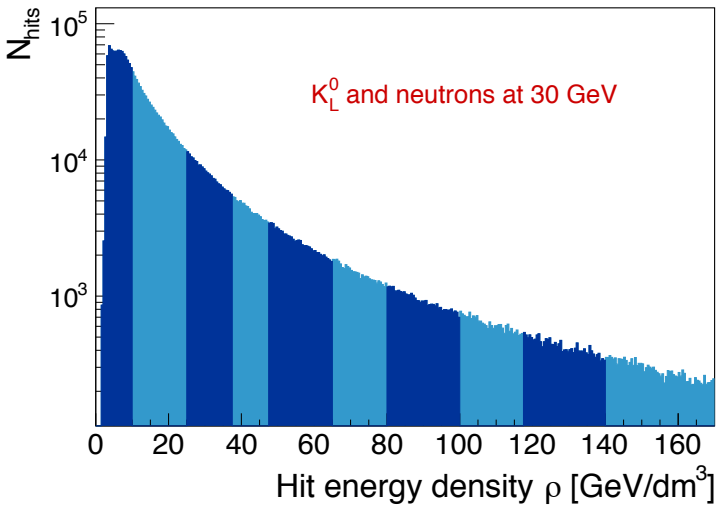
- Software compensation applied to test beam data from CALICE-AHCAL physics prototype:
  - Improvement of hadronic energy resolution by 20% for single hadrons from 10 to 80 GeV

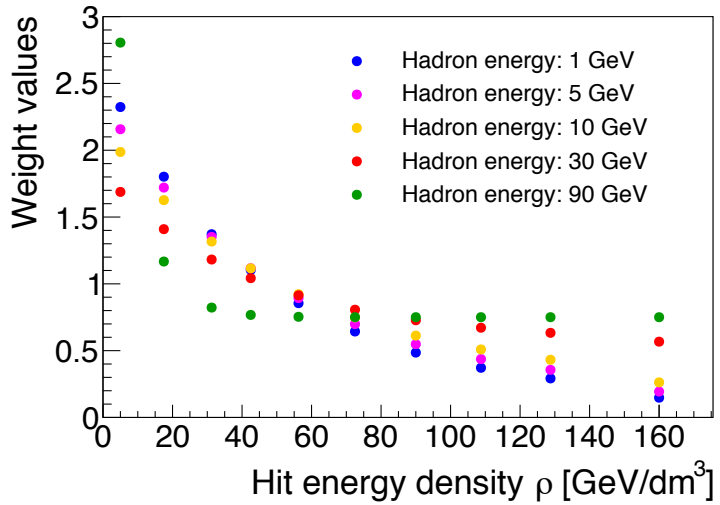
[arXiv: 1207.4210v2]



Software compensation idea

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



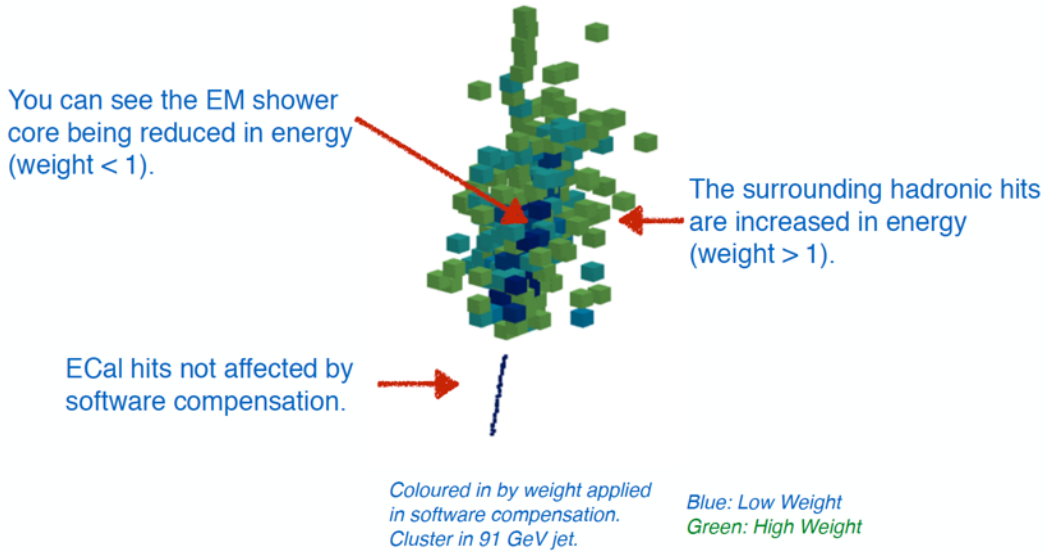


$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$

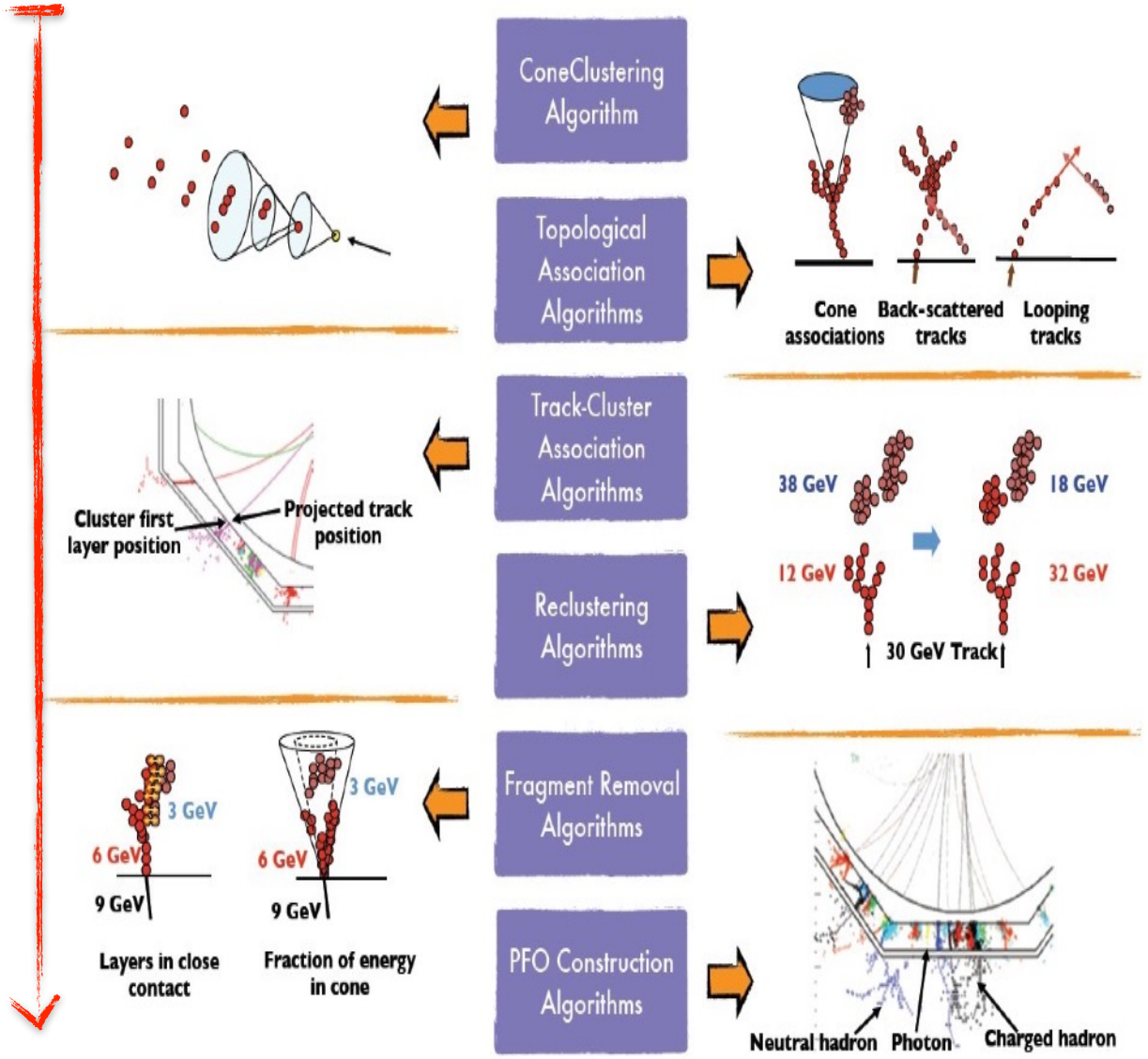
$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{bin}(E^{bin}_{HCAL} \times \omega_{bin}(\rho))$$

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

Example on software compensation's operation



Software compensation implementation



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

Application of SC for ILD detector performance study

HELMHOLTZ ASSOCIATION

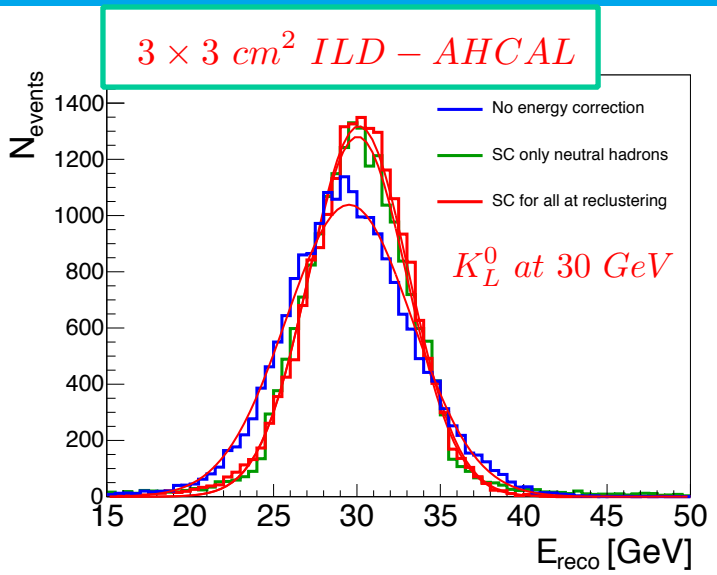
DESY

Software version and configuration

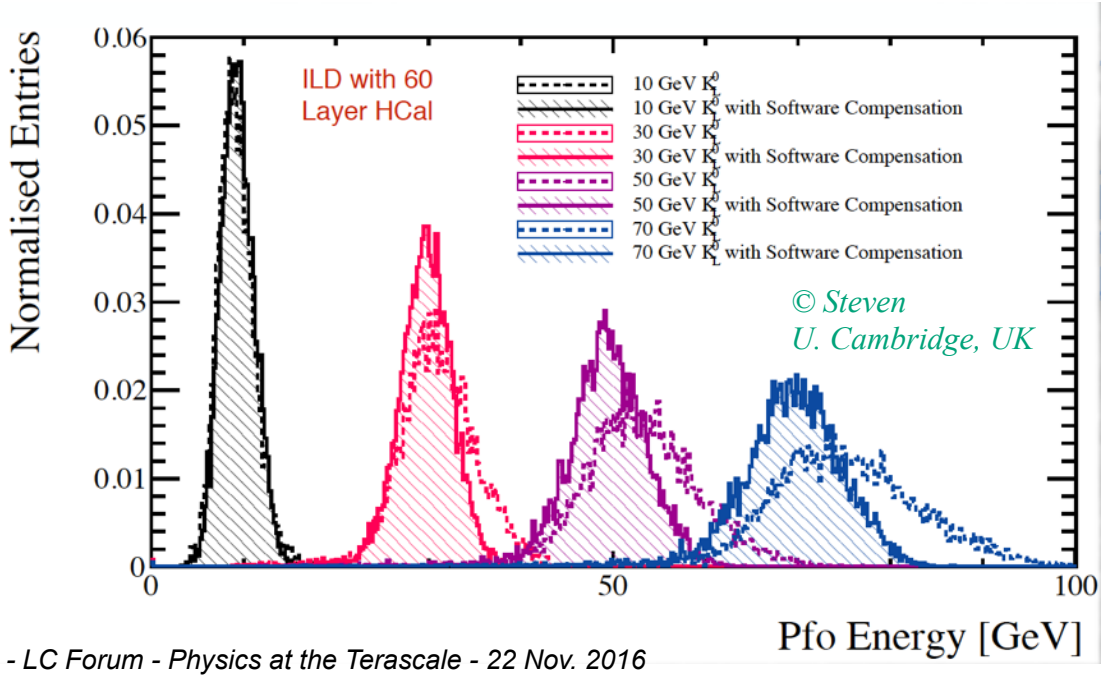
- **Detector model**: ILD_o1_v06
- **Reconstruction software**: ilcsoft_v01-17-07 combined with PandoraPFA version v02-09-00:
  - PandoraSDK v02-03-01
  - LCContent v02-04-00 including software compensation in LCPlugins and hits information registration for software compensation weight training in LCUtility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

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

Energy resolution with software compensation

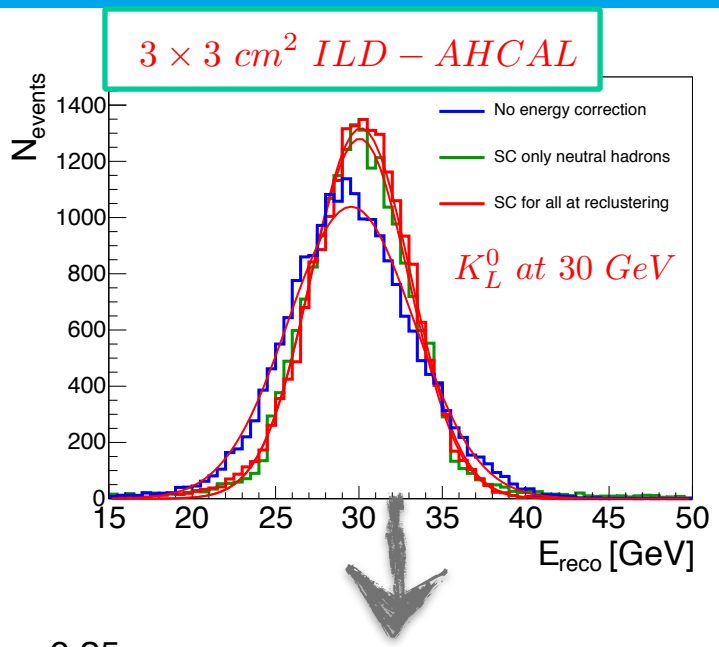


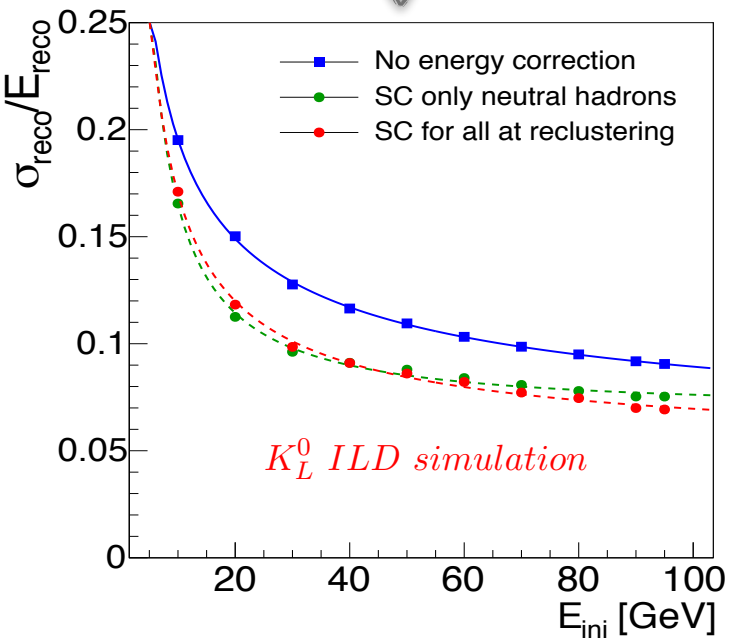
- Software compensation benefits in two-fold way:
  - Improve energy reconstruction of neutral objects
  - Improve cluster energy estimator for better track-cluster association ➢ confusion mitigation



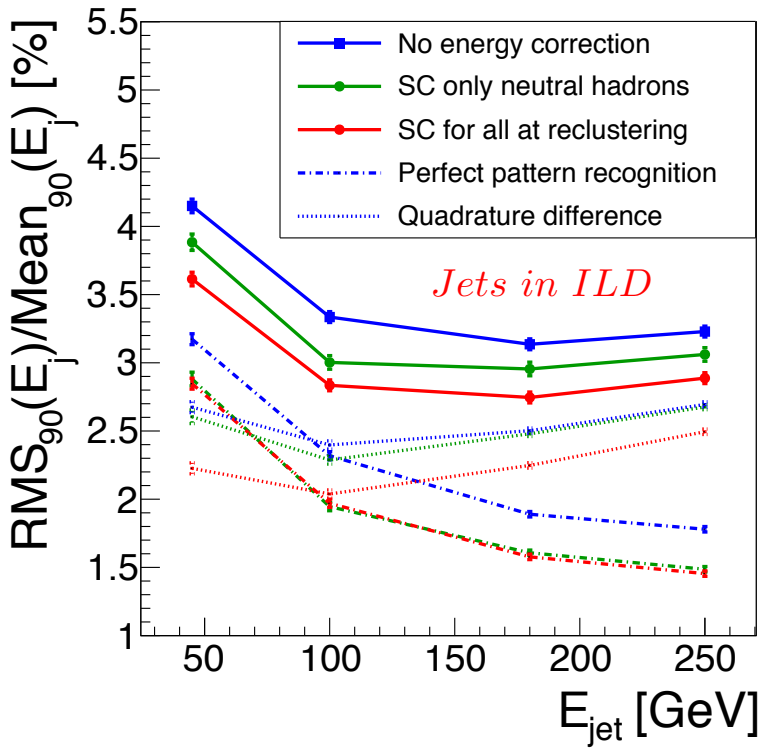
Study with 60 layer HCAL and higher jet energies (relevant for CLIC studies)

Energy resolution with software compensation



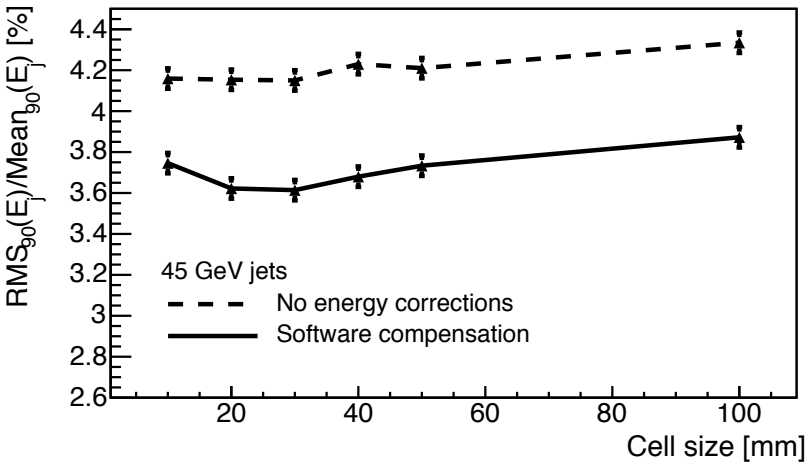


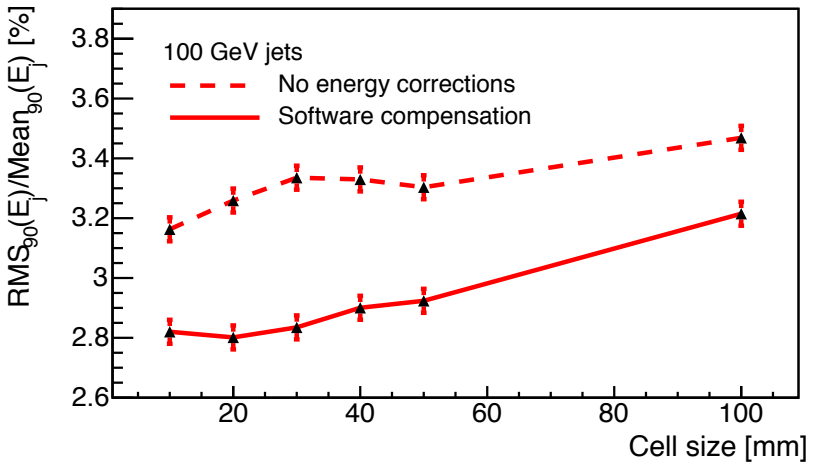
- Software compensation benefits in two-fold way:
  - Improve energy reconstruction of neutral objects
  - Improve cluster energy estimator for better track-cluster association ➢ confusion mitigation
- Significant improvement at both single particle and jet level
- Software compensation applied at re-clustering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

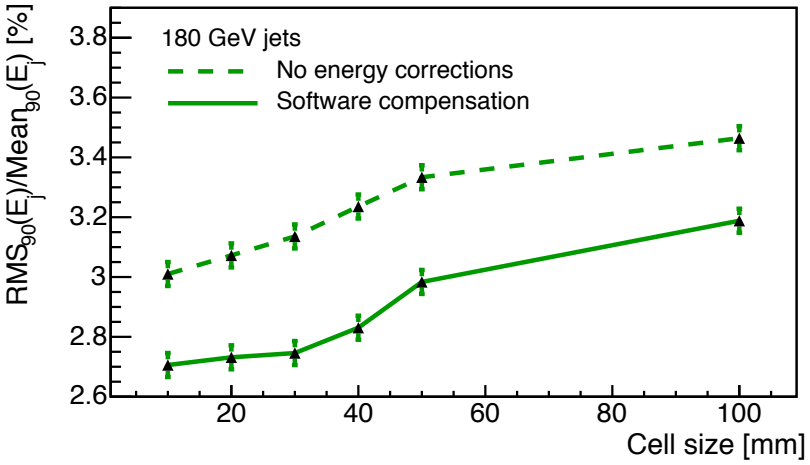


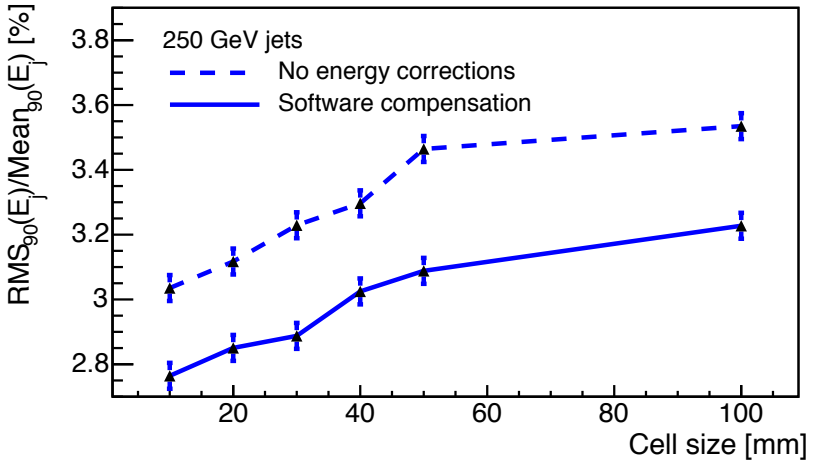
JER vs transverse granularity (cell size)

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



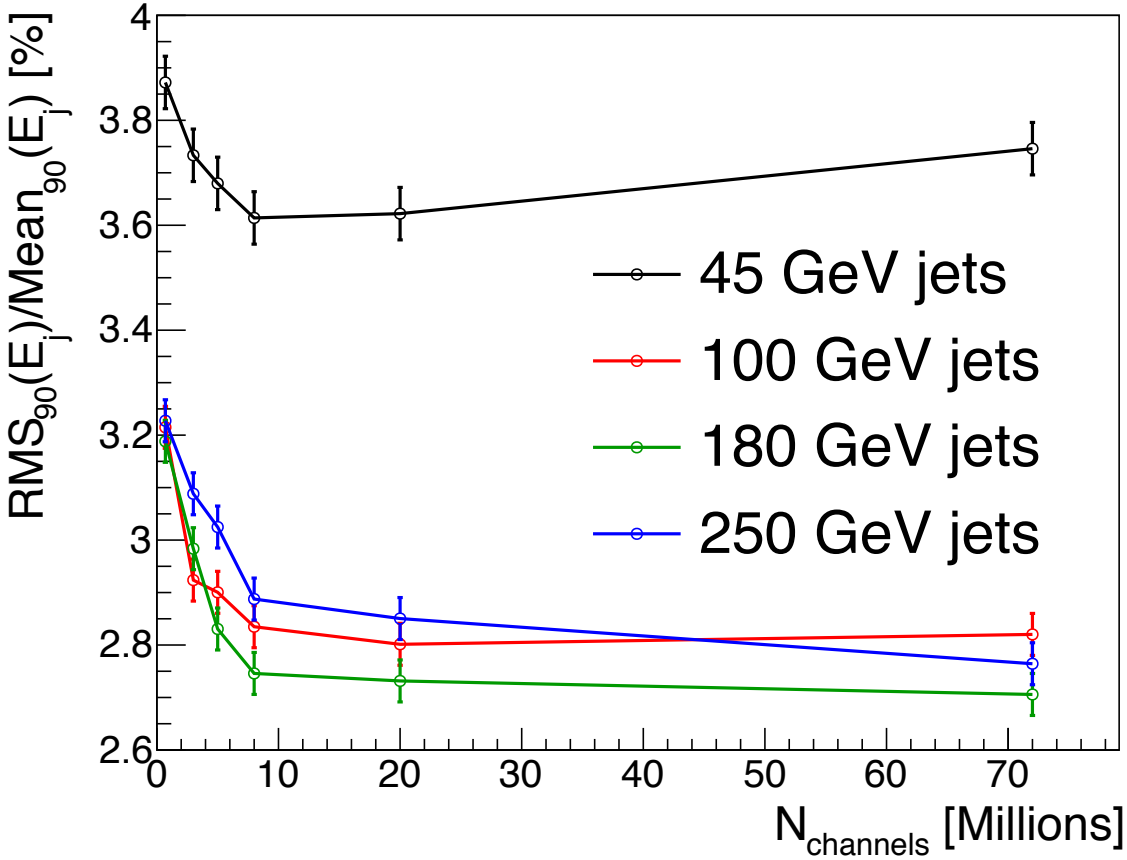






JER vs number of cells

- 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



Summary

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

- Software compensation code and utilities in latest version of PandoraPFA

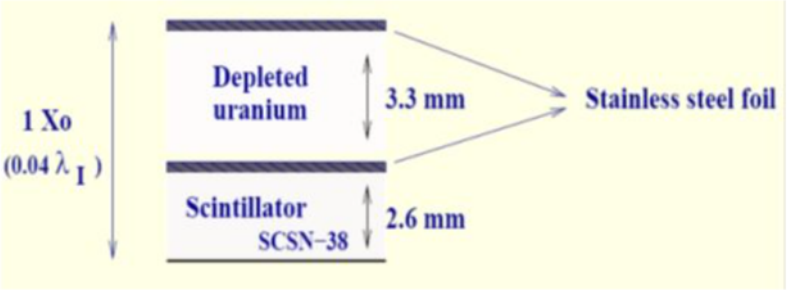
- Installed in new ILCsoft v01-17-10
  - Being used as standard cluster energy correction, shows improvement everywhere
    (even with DD4hep, without re-optimisation of SC weight yet)

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

Back-up slides

Methods to achieve Compensation

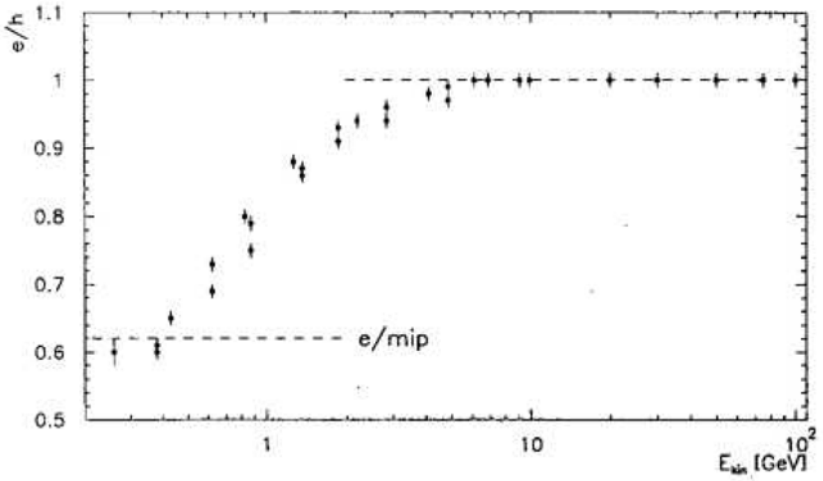
- Reducing electromagnetic response
- Increasing hadronic response



ZEUS Uranium-Scintillator calorimeter

Achievable with detector design

- Increase nuclear fission with absorber material
  - Example: ZEUS detector using ^{238}U
- Manipulating response to (slow) neutrons
- Sampling fractions
- …



ZEUS e/h response ratio
=1 within 1% for E > 3GeV

Software Compensation

- ***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:

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

- In following slides: Results on standard ILD detector (with 3x3 *cm2* AHCAL)

Semi-digital Reconstruction

- Semi-digital reconstruction:
  - Counting hits at 3 thresholds N1, N2, N3
  - Ntot = N1 + N2 + N3
  - EnergySD = alpha*N1 + beta*N2 + gamma*N3

where:

alpha = alpha1 + alpha2*N + alpha3*N*N

beta = beta1 + beta2*N + beta3*N*N

gamma = gamma1 + gamma2*N + gamma3*N*N

Software compensation mimics Semi-Digital:

- Define bin
- Energy total = Sum_bin (weight_bin * SumEnergy_bin)
- weight_bin = a + b*E + c*E*E