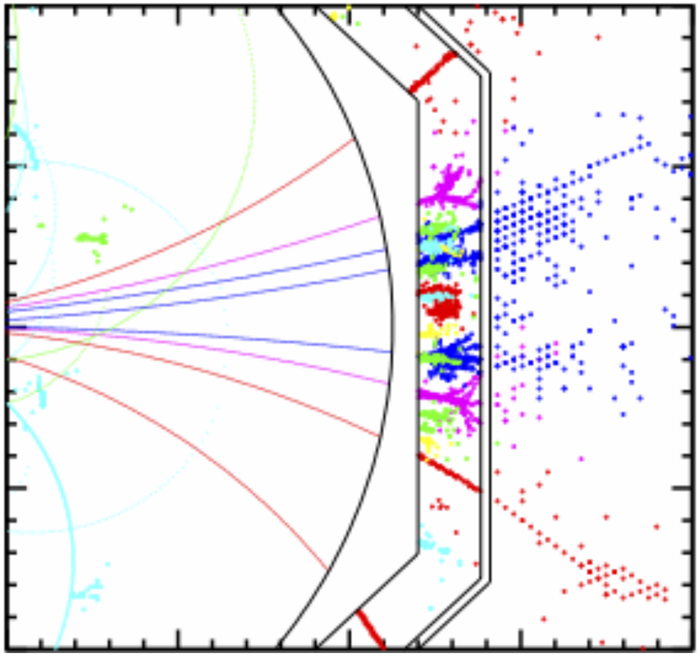
AHCAL Optimisation in ILD



Huong Lan Tran

CALICE AHCAL main meeting

10-11 December 2015

Outlines

- Efforts for AHCAL optimisation are converging
  - Fruitful collaboration with Munich and Cambridge group concerning software compensation and its implementation
  - Good results achieved

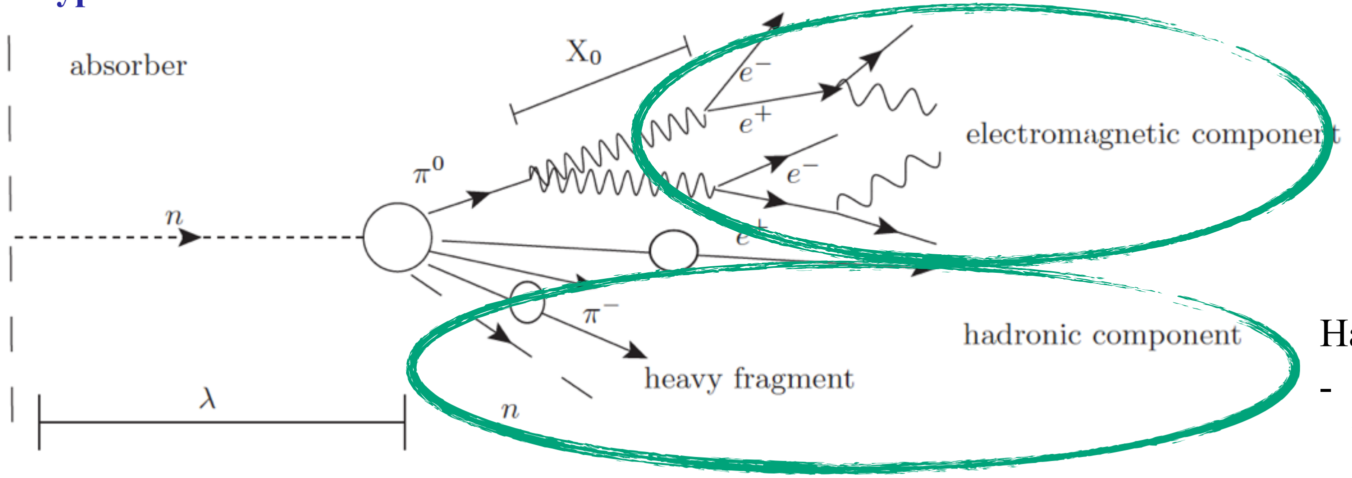
In this talk: Software compensation for AHCAL optimisation

- Motivation
- Updates on fitting procedure
- Towards a common SC technique for different types of HCAL
- Bonus: Implementation in Particle Flow Algorithm

Why Compensation?

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

Typical hadronic shower



Detected via energy loss of electrons and photons in active medium

Hadronic components:
- Energy loss of charged hadrons, photons, neutrons…
- *Invisible energy*: nuclear binding energy or target recoil

➪ Smaller calorimeter response for this part

➢ **<u>Consequences:</u>**

- 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

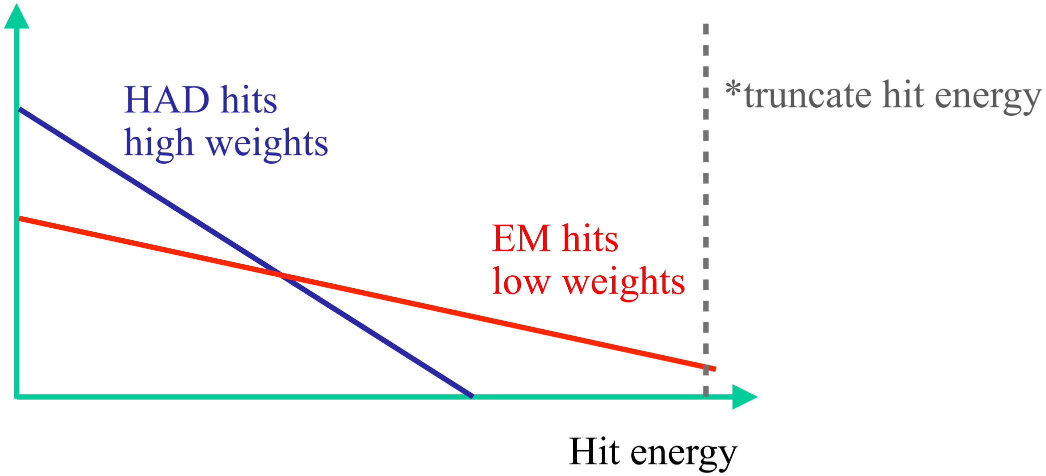
- Reducing electromagnetic response
- Increasing hadronic response

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.

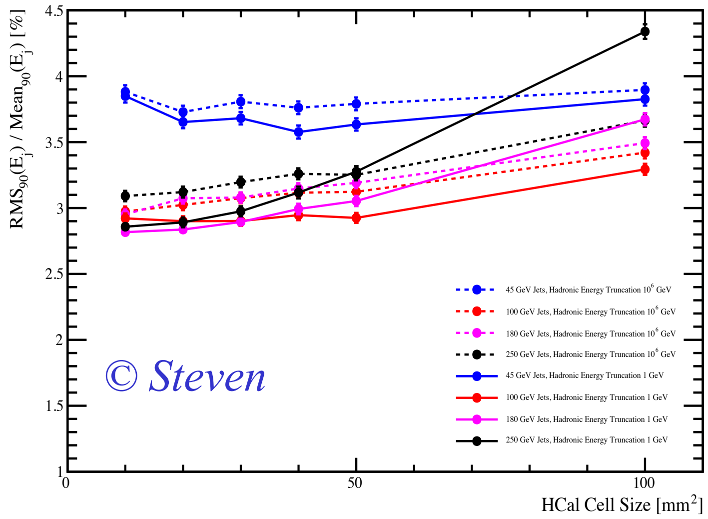
➢ Cut out high energy hits to reduce EM response *

➢ Applying different weights for hits of different energy densities



Software Compensation in AHCAL optimisation

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

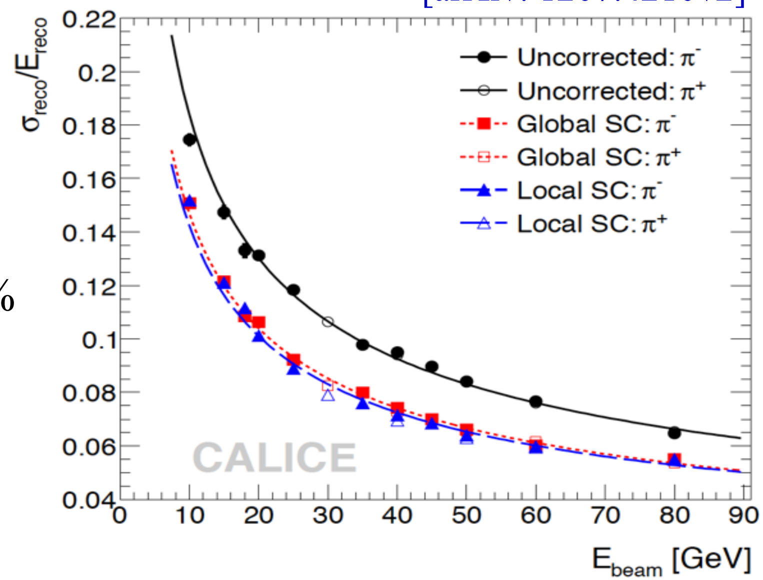


➢ Cell energy truncation mimics software compensation

➢ Software compensation can do better and must be applied properly

[arXiv: 1207.4210v2]

- 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



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*

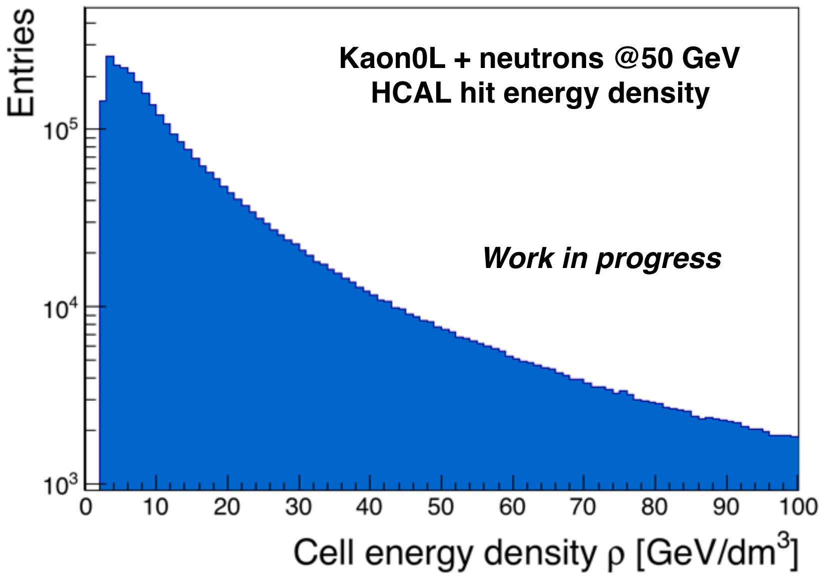
- Energy of cluster then computed in software compensation method as:

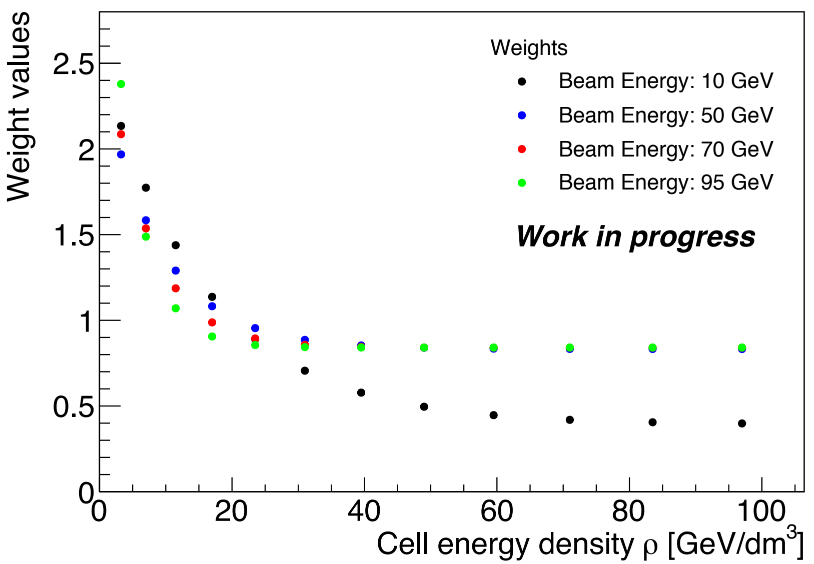
$$E_{SC} = \sum_{hits} E_{ECAL} + \sum_{hits} (E_{HCAL}.\omega(\rho))$$

- 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

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

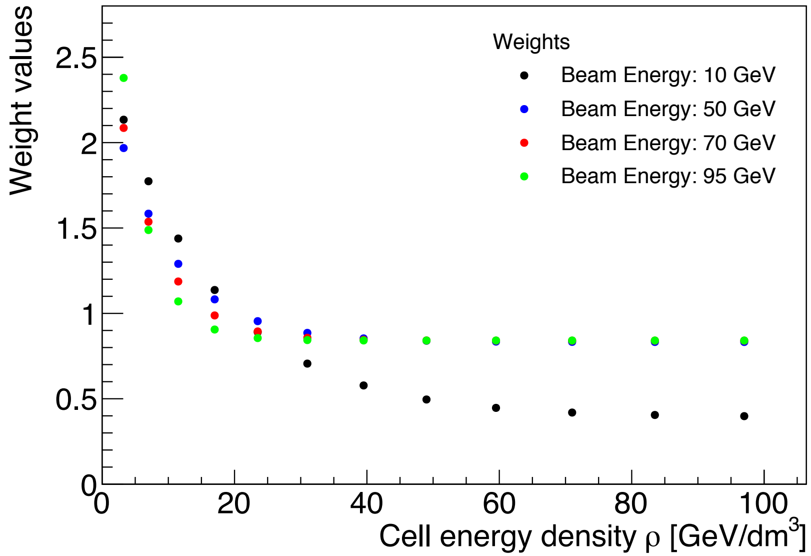
where p_1, p_2, p_3 are energy dependent parameter (defined directly in χ^2)

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

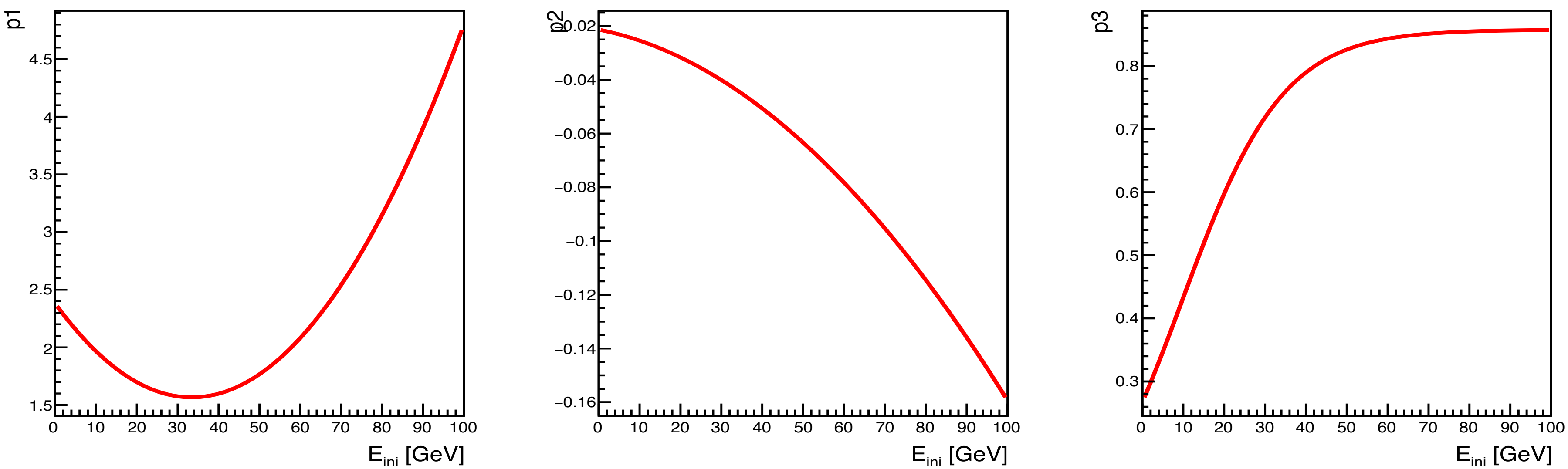
$$p_2 = p_{20} + p_{21} \times E_{ini} + p_{22} \times E_{ini}^2$$

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

Weight parameters

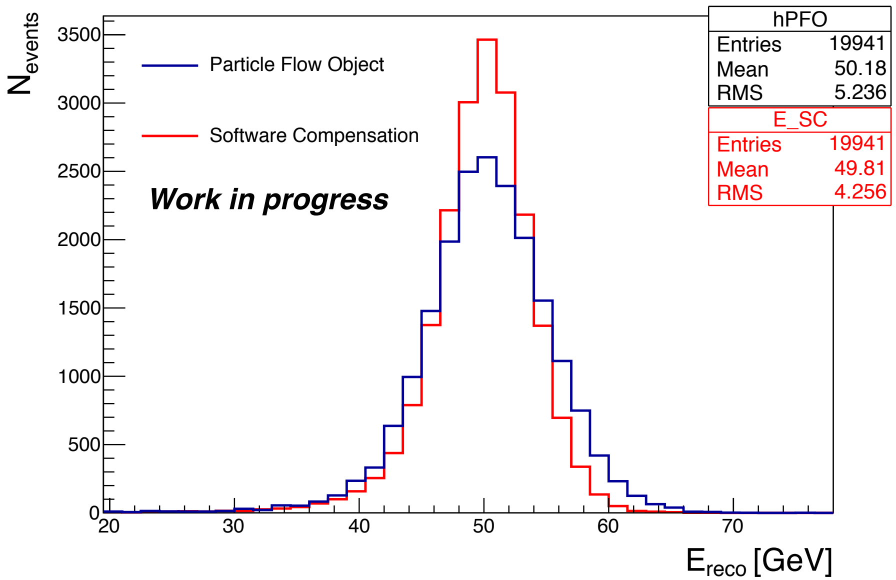


$$\omega(\rho) = p_1.exp(p_2.\rho) + p_3$$



Single Particle Energy Reconstruction

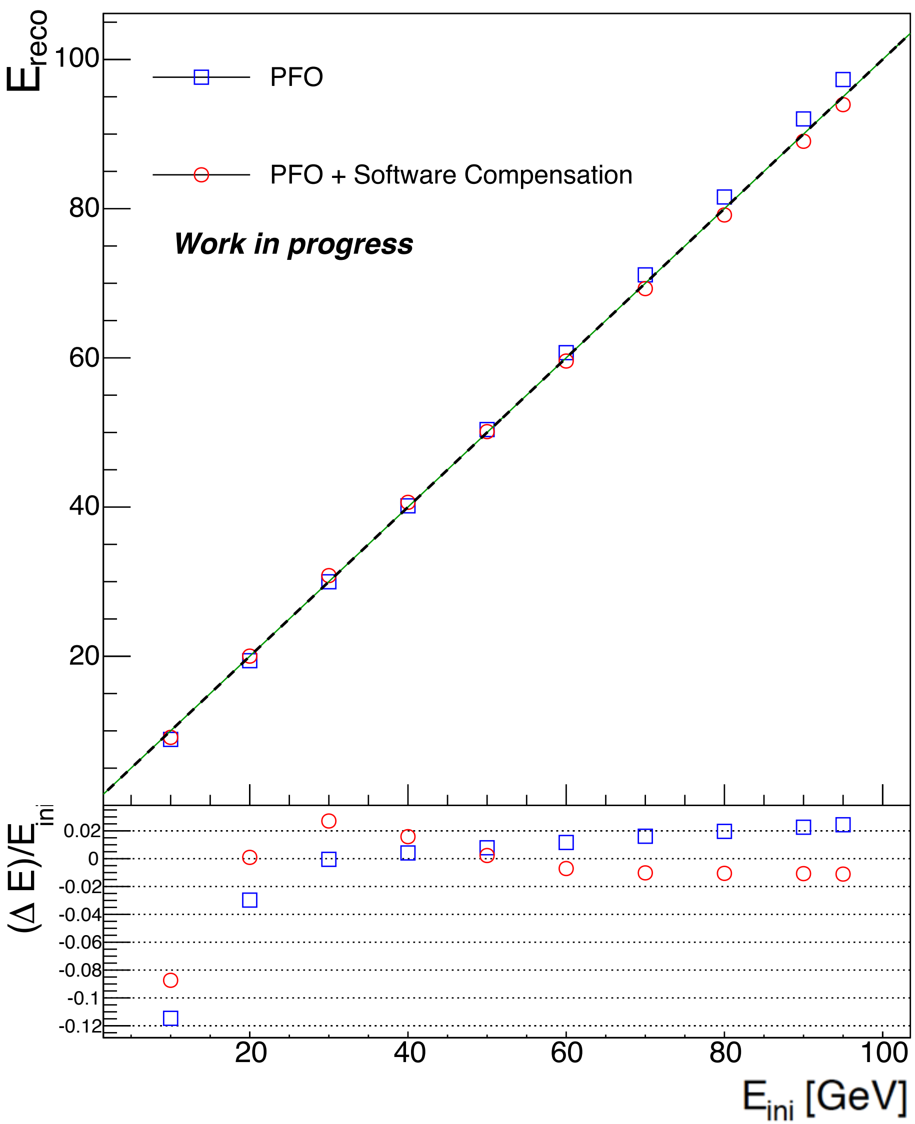
- Correction of neutral hadron PFOs energy
- Initial estimation of cluster's energy used for determination of weights
- Apply to set of Kaon0L and neutron samples from 10 to 95 GeV



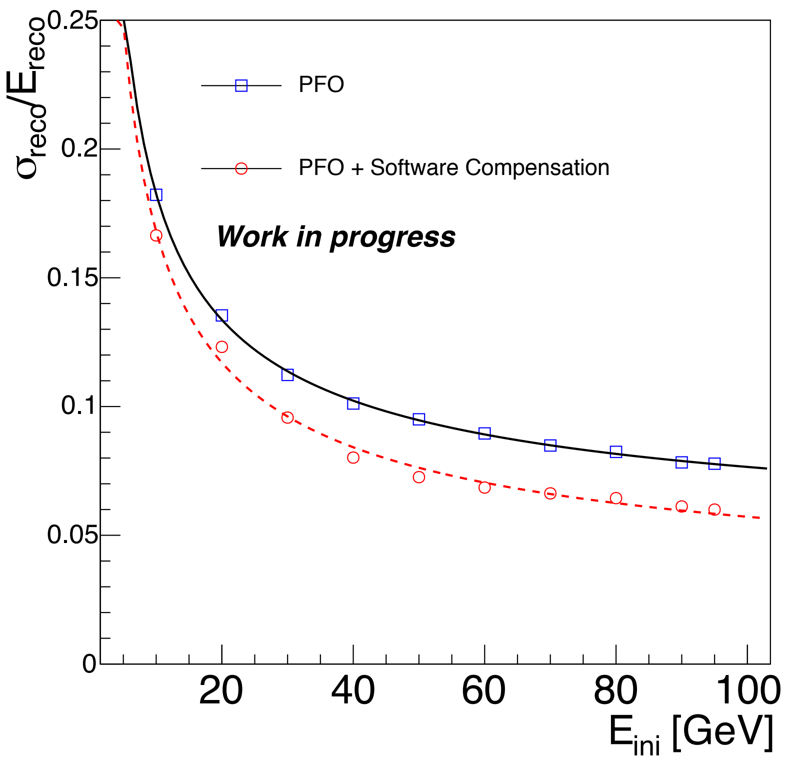
50 GeV Kaon0L

- Improvement of mean reconstructed energy
- RMS significantly reduced

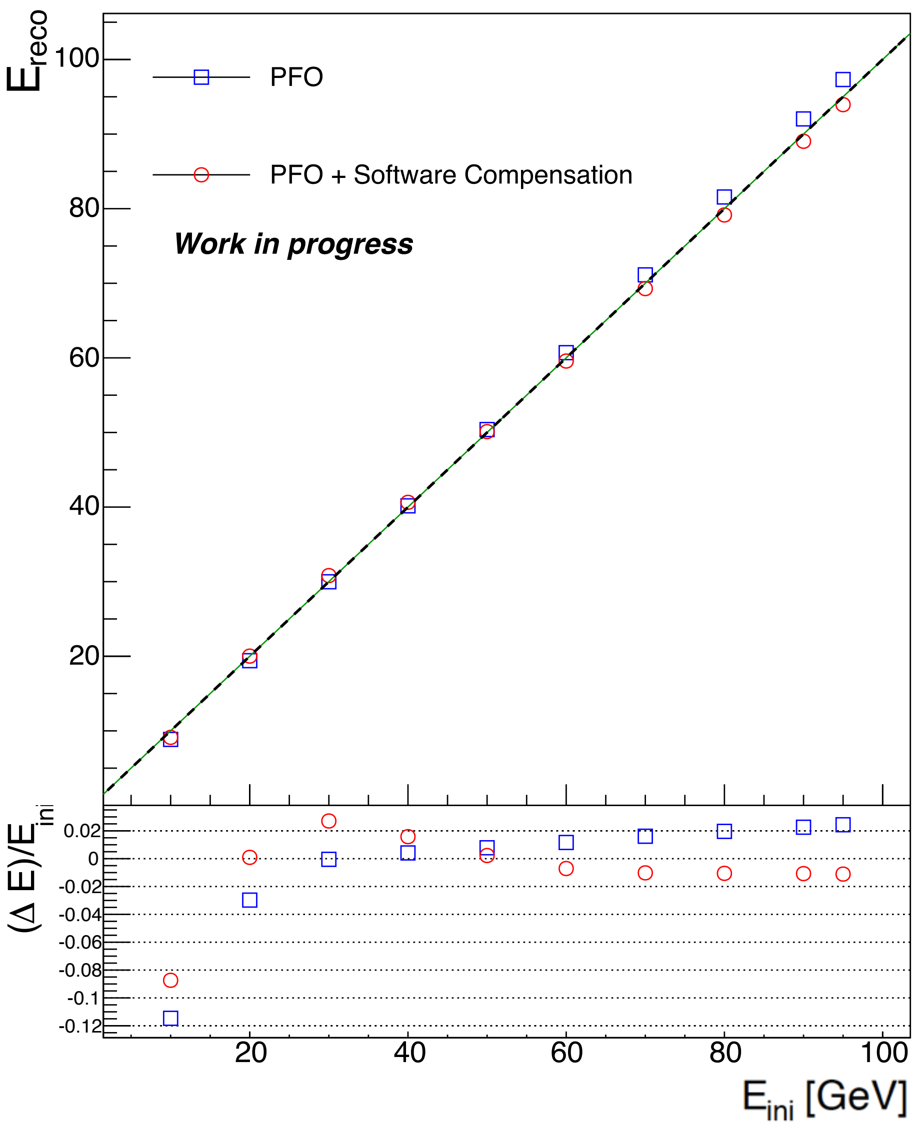
Single Particle Energy Reconstruction



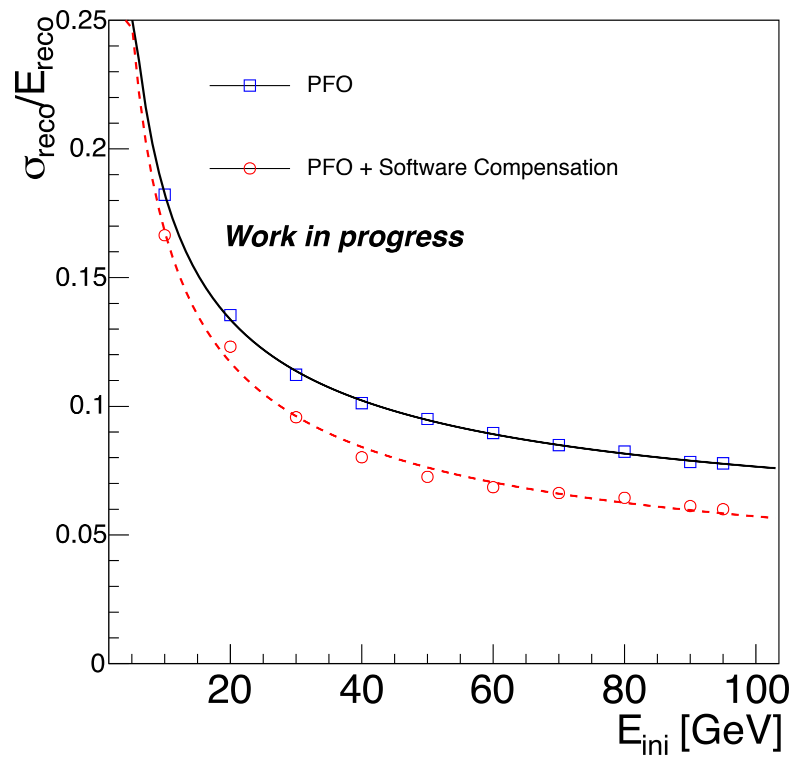
- Improves linearity in whole range
- Improves resolution by ~20% (similar to results obtained for physics prototype)



Single Particle Energy Reconstruction



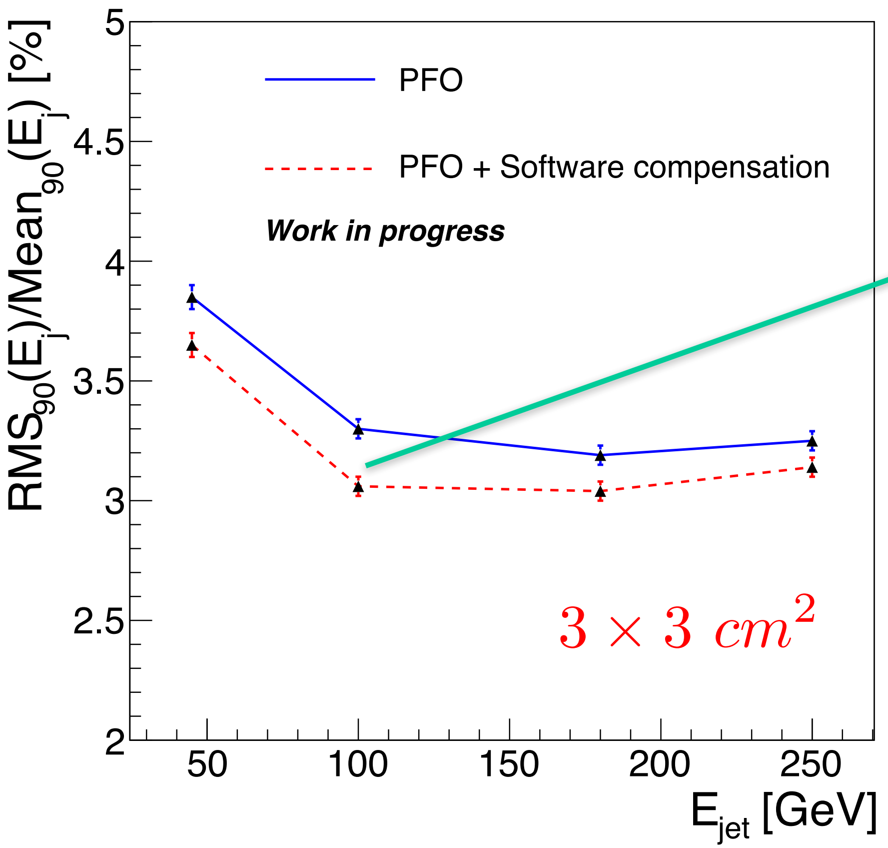
- Improves linearity in whole range
- Improves resolution by ~20% (similar to results obtained for physics prototype)

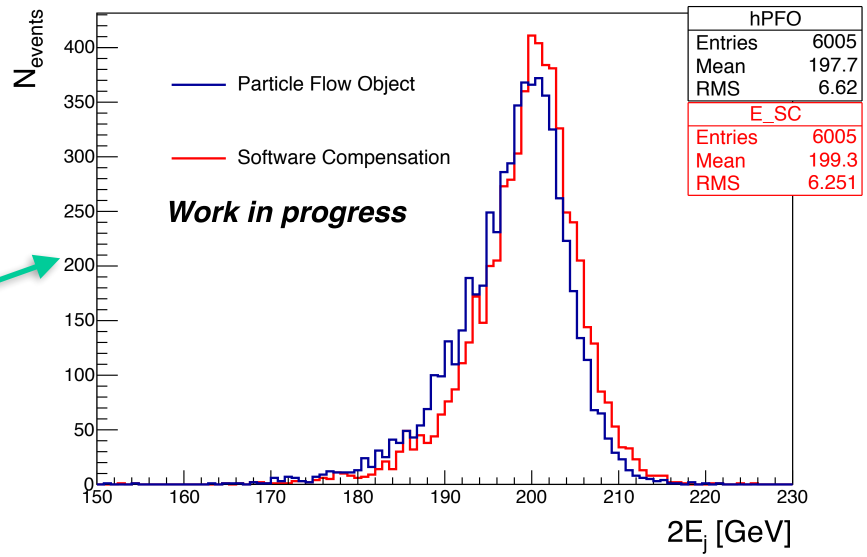


- Testbeam results reproduced
- Overall slightly worse because of missing tail catcher

Jet Energy Resolution

- Software compensation applied for jets
  - Only for neutral hadrons, after clustering and re-clustering step
  - Only hits in HCAL are weighted as explained previously



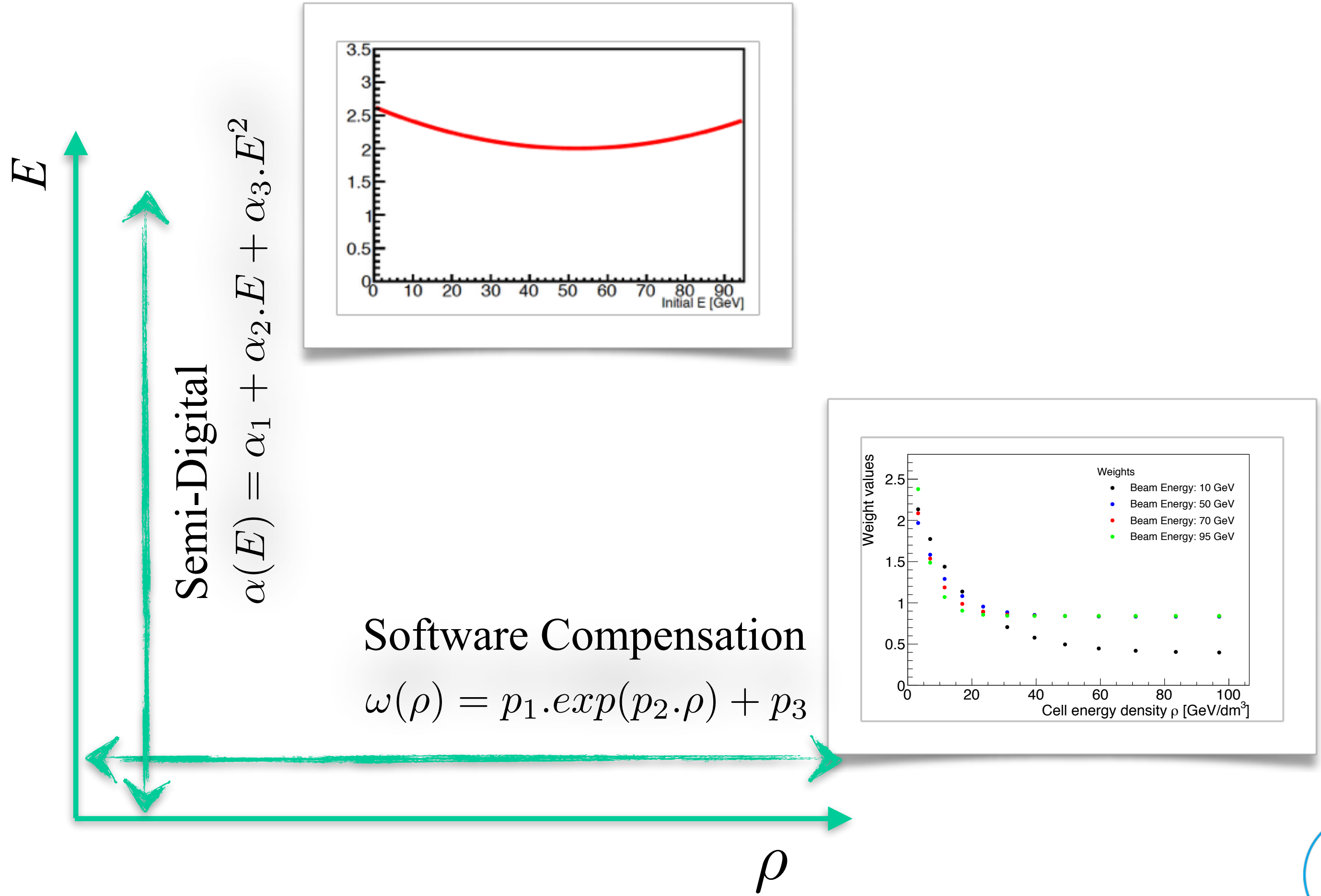


- Reconstructed energy distribution closer to simulated energy and width of distribution smaller
- Improves jet energy resolution in whole range

Towards a common SC technique for different types of HCAL

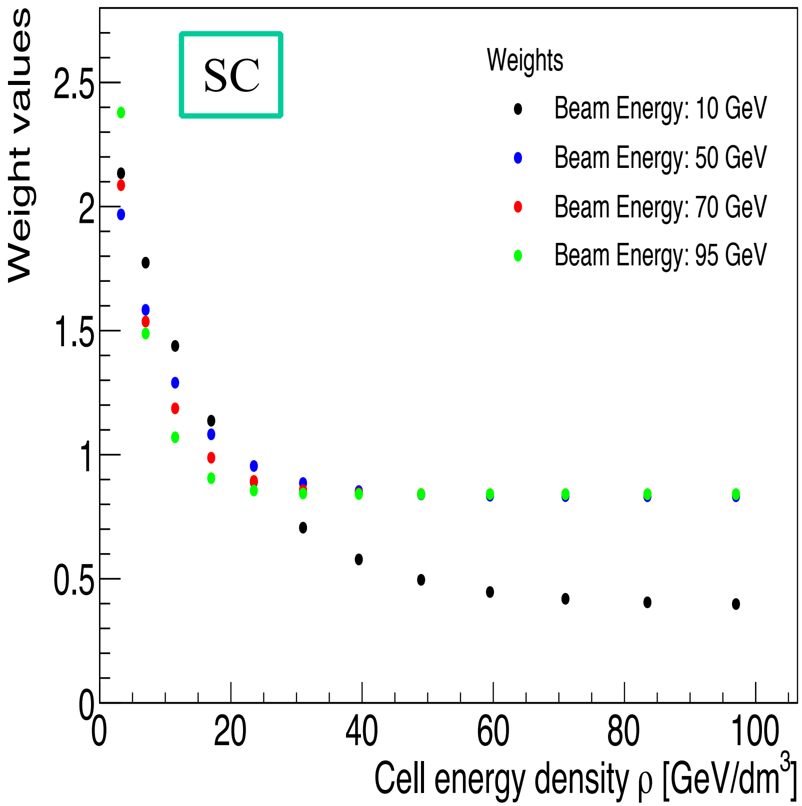
Semi-Digital and Software Compensation

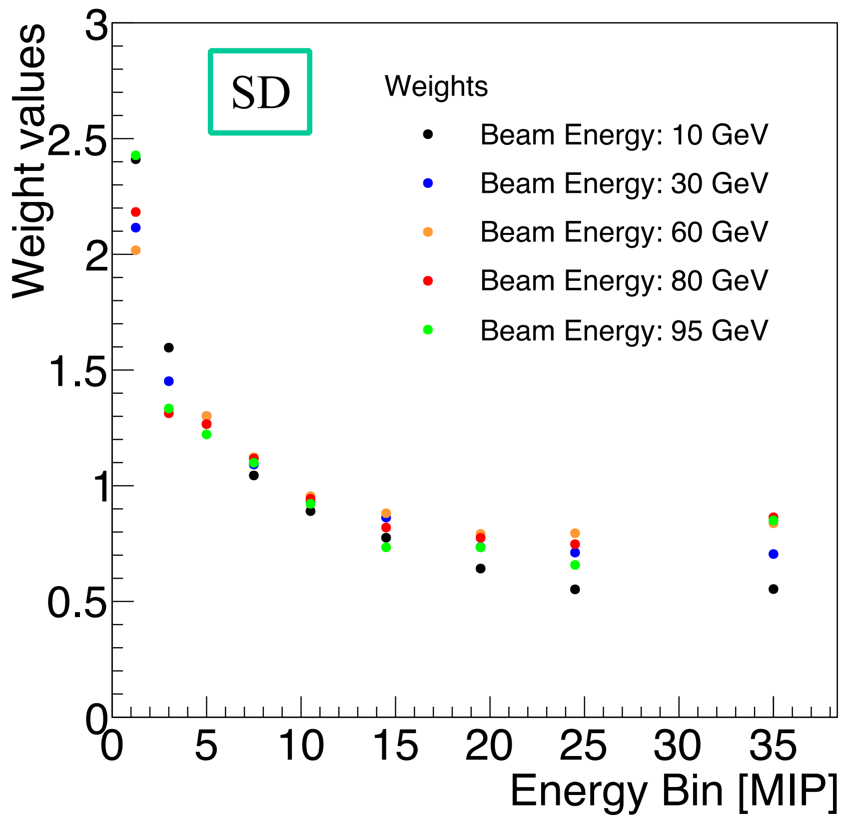
- Software Compensation: weight optimised as a function of ***hit energy density*** ρ
- Semi-Digital: weight optimised as a function of ***particle energy*** E



Software Compensation in S-D style

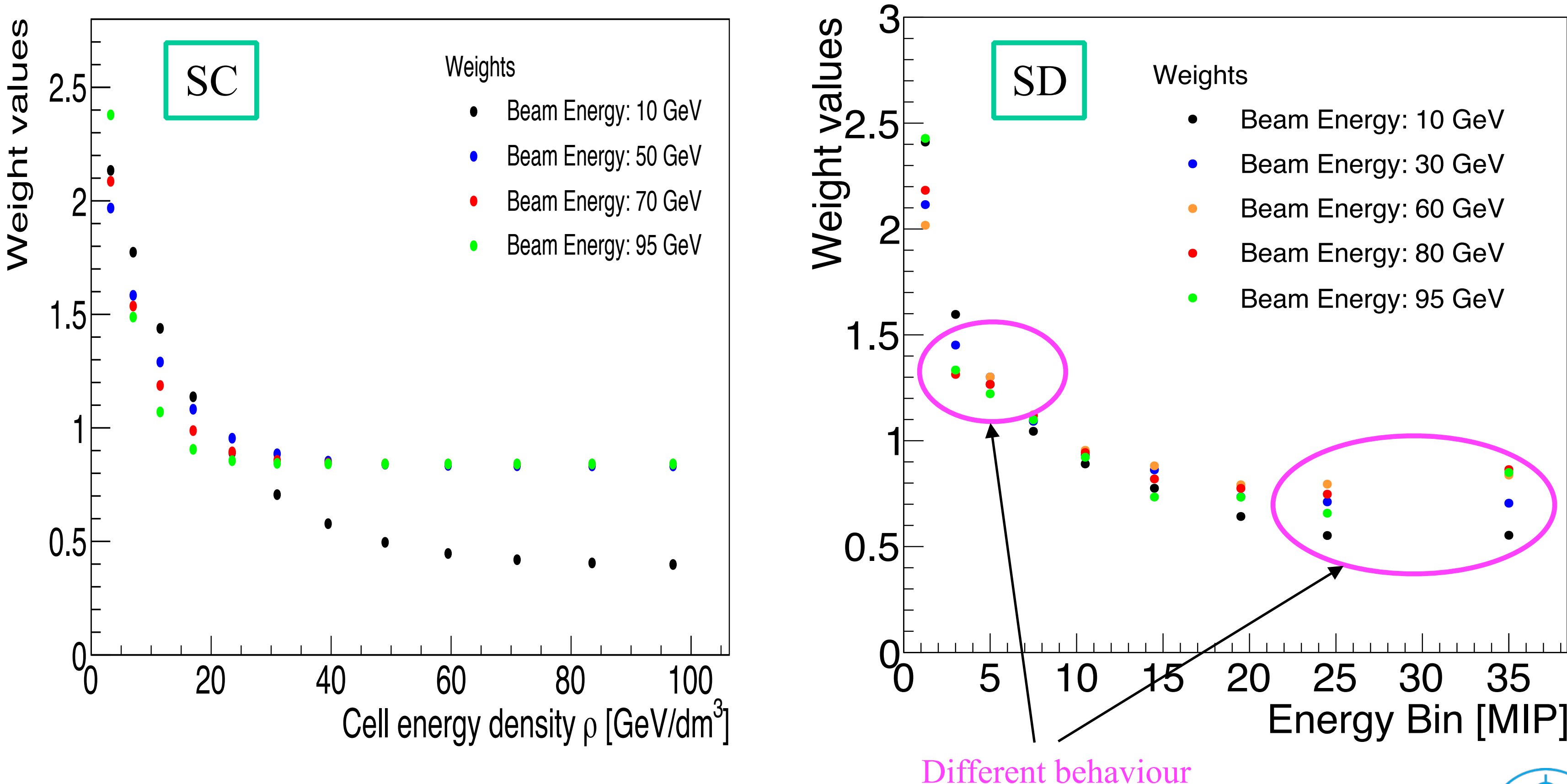
- New procedure defined:
  - No longer enforce weight to follow exponential behaviour ~~$\omega(\rho) = p_1\ exp(p_2.\rho) + p_3$~~
  - Weights determined for each bin of hit energy as a function of beam energy (all-at-one fit)





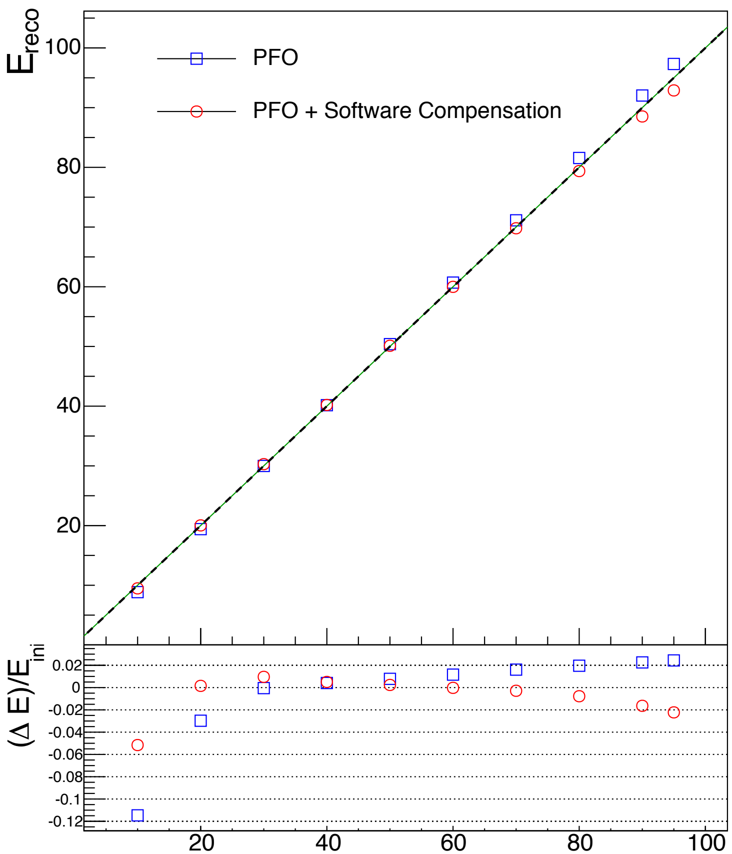
Software Compensation in S-D style

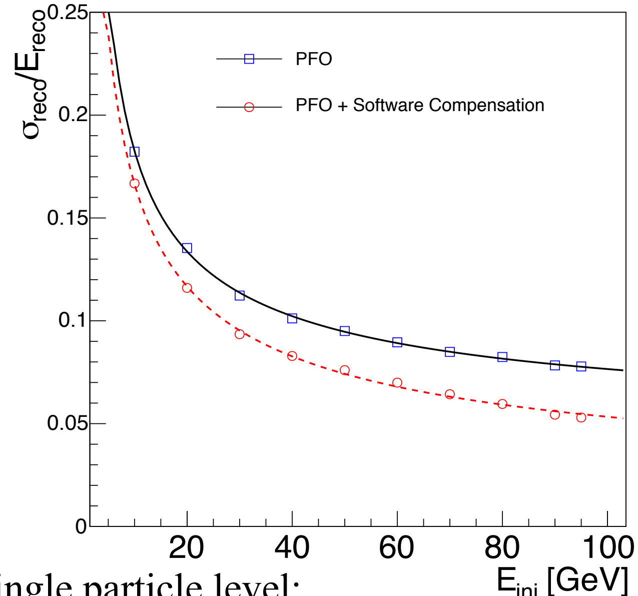
- New procedure defined:
  - No longer enforce weight to follow exponential behaviour $\omega(\rho) = p_1 \, exp(p_2.\rho) + p_3$
  - Weights determined for each bin of hit energy as a function of beam energy (all-at-one fit)



Software Compensation in S-D style

- New procedure defined:
  - No longer enforce weight to follow exponential behaviour $\omega(\rho) = p_1\, exp(p_2.\rho) + p_3$
  - Weights determined for each bin of hit energy as a function of beam energy (all-at-one fit)
  - Correction for neutral hadrons energy, after clustering and re-clustering step





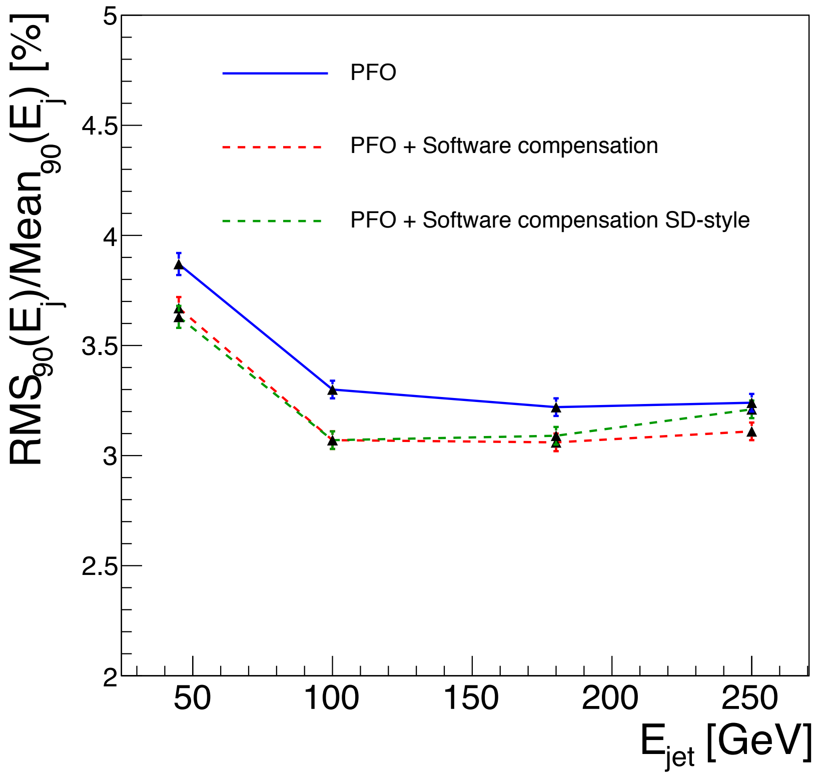
Single particle level:

- Better compared to previous results
- Improves linearity in whole range
- Improves resolution ~ 20%

  For higher energies ~ 30%

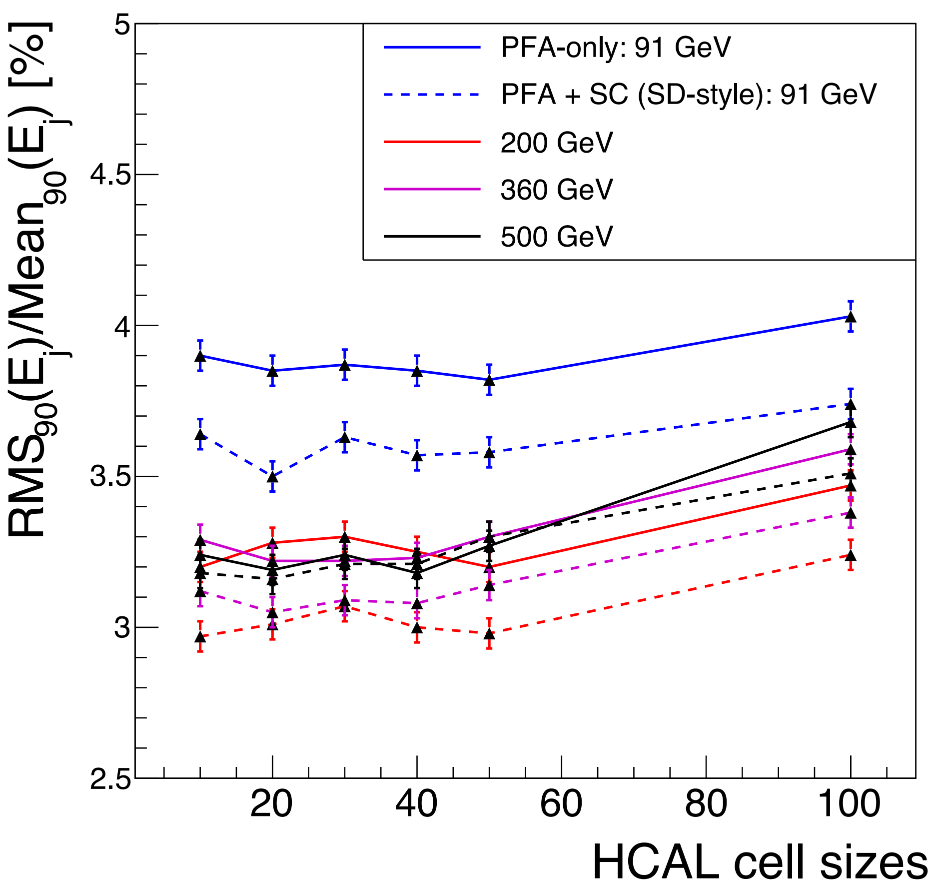
Software Compensation in S-D style

- New procedure defined:
  - No longer enforce weight to follow exponential behaviour $\omega(\rho) = p_1\ exp(p_2.\rho) + p_3$
  - Weights determined for each bin of hit energy as a function of beam energy (all-at-one fit)
  - Correction for neutral hadrons energy, after clustering and re-clustering step



- At jet level gives more or less the same result as previously

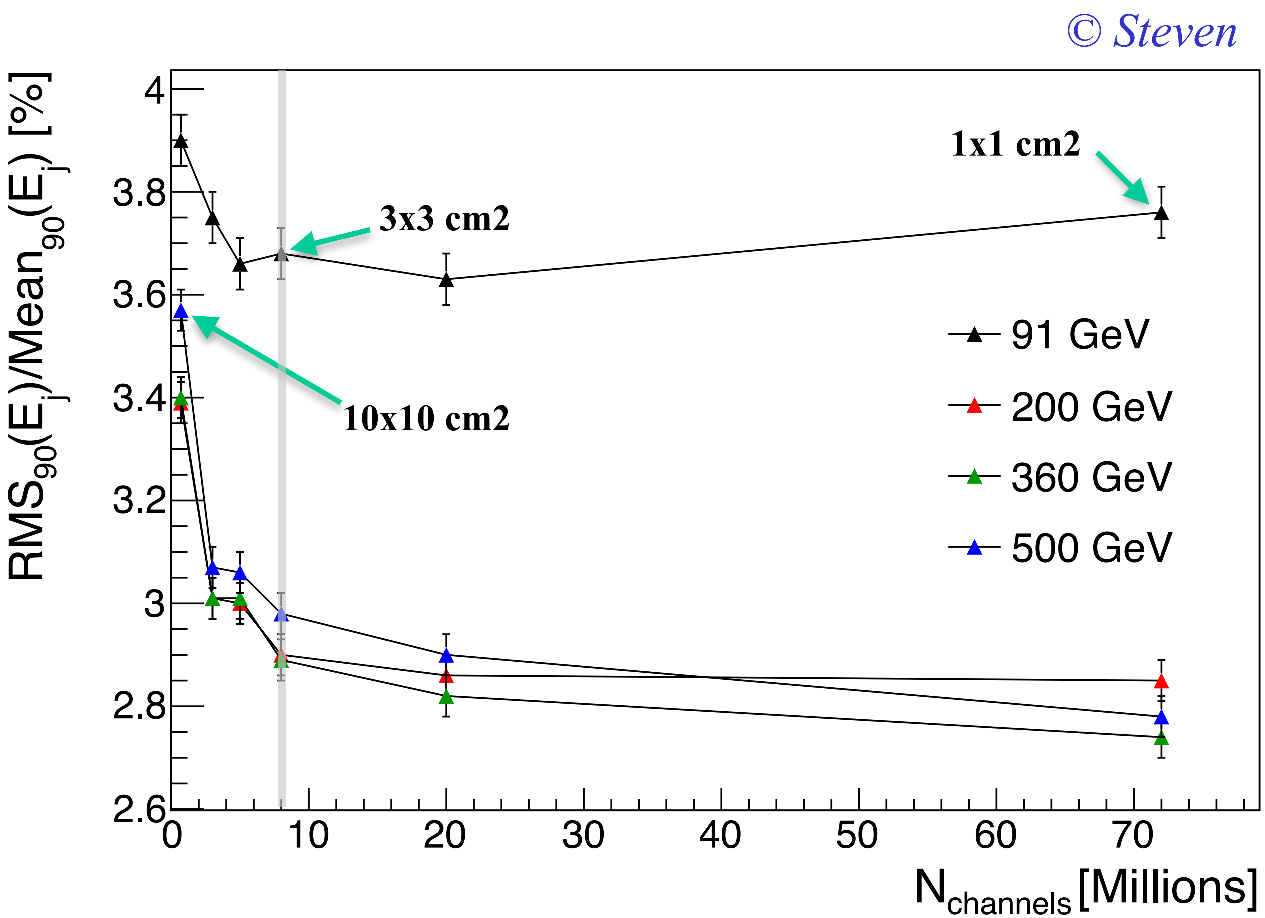
JER vs cell size



- Over all improvement
- At high energy slightly worse compared to Steve's numbers using truncation cut
    - HCAL truncation cut takes effect already at clustering step: for higher energy jets (large confusion) this is expected

Towards cost optimisation

- Look at jet energy resolution as a function of number of channels
- Plot shows that 3x3 *cm2* cell size is still a very reasonable choice with latest Pandora
- Software compensation to be applied



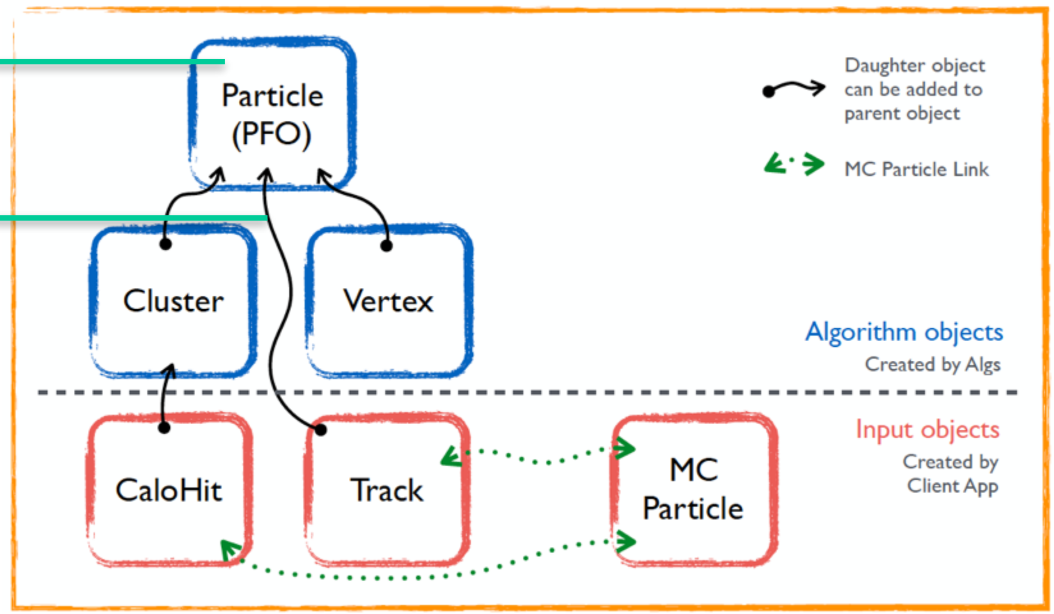
Latest results from Steven
To be updated with
software compensation

Implementation into Pandora

- Software compensation can help at different stages of Particle Flow Algorithm:
    - Re-clustering: Cluster-Track compatibility
    - Partile Flow Object creation: **Correction of neutral hadrons energy**

Neutral hadrons PFO registered with software compensated energy

Cluster-Track compatibility: if failed, re-clustering



- Flag in MarlinPandora steering to apply software compensation:

NEW

```
<parameter name="ApplySoftwareCompensation" type="bool"> false </parameter>
<parameter name="SoftwareCompensationParameters" type="FloatVec"> 2.54231 -0.0470912 ...
</processor>
```

Summary & Outlook

- *Software compensation and cell size optimisation:*
  - Software compensation *implemented in Pandora*
    - To be made official
  - *Improves* single particle and jet energy resolution
  - Re-clustering step to be done
- Common SC technique for different types of HCAL developed

- Final goal: HCAL cell size and sampling optimisation (3D granularity) as a function of depth and for different detector radii

- One week in Cambridge (whenever my visa makes it way back)

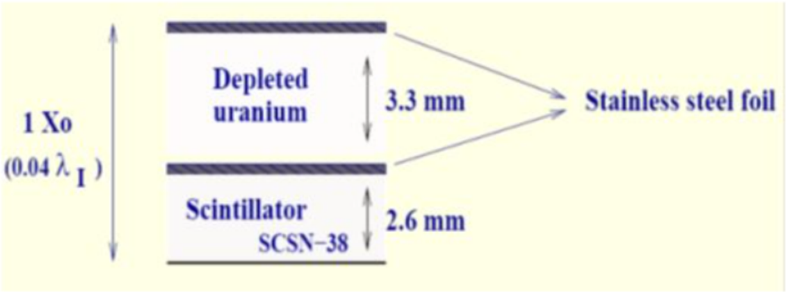
Back-up slides

Methods to achieve Compensation

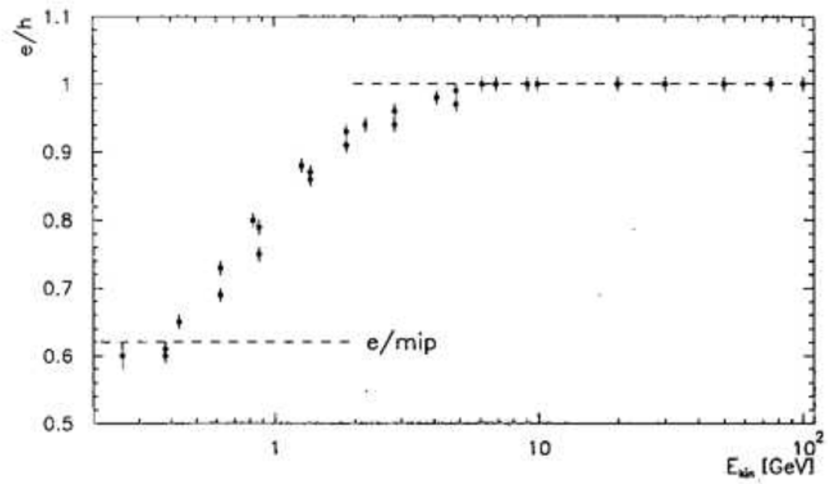
- Reducing electromagnetic response
- Increasing hadronic response

Achievable with detector design

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

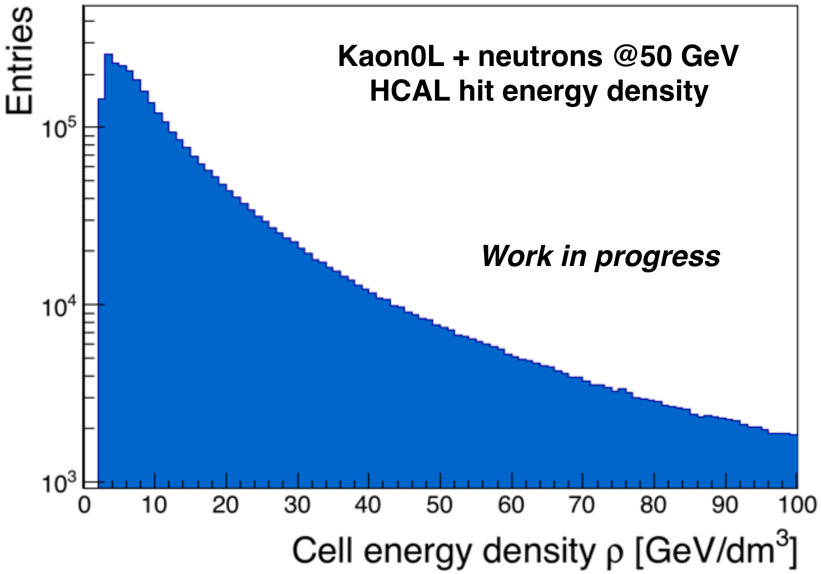


ZEUS Uranium-Scintillator calorimeter



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

Reminder: Hit Energy Density and Weights

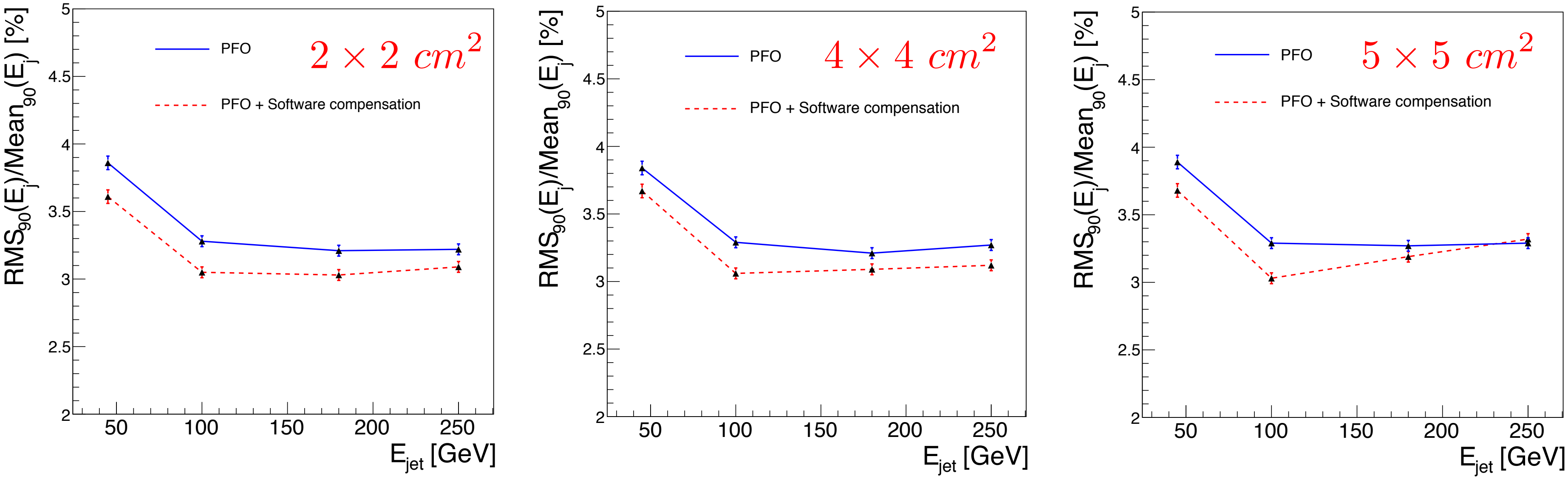


Samples:

- Kaon0L and neutrons from 10 to 95 GeV generated from IP, targeted only to barrel part
- Select only events with 1 cluster
  - Events where hadronic showers started already in EM calorimeter: only HCAL hits are weighted
  - Cluster with no hit in muon chamber

Jet Energy Resolution for Different Cell Sizes

- For similar cell sizes still expect improvement using weights defined with $3 \times 3\ cm^2$

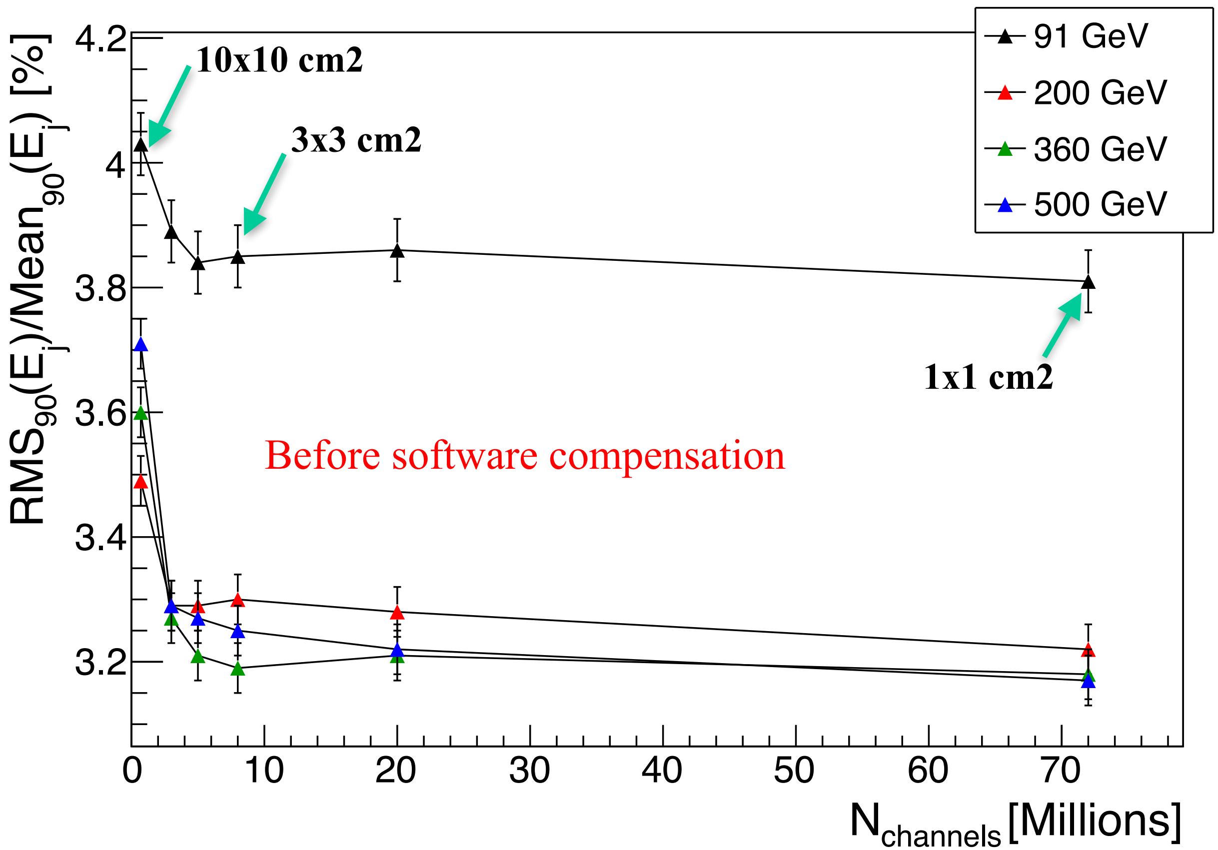


- Proper weights to be done, especially for very small or very large granularities
- SC could also help at re-clustering stage of Pandora
  - At the moment degrades JER, under investigation

Outlook - Using my numbers

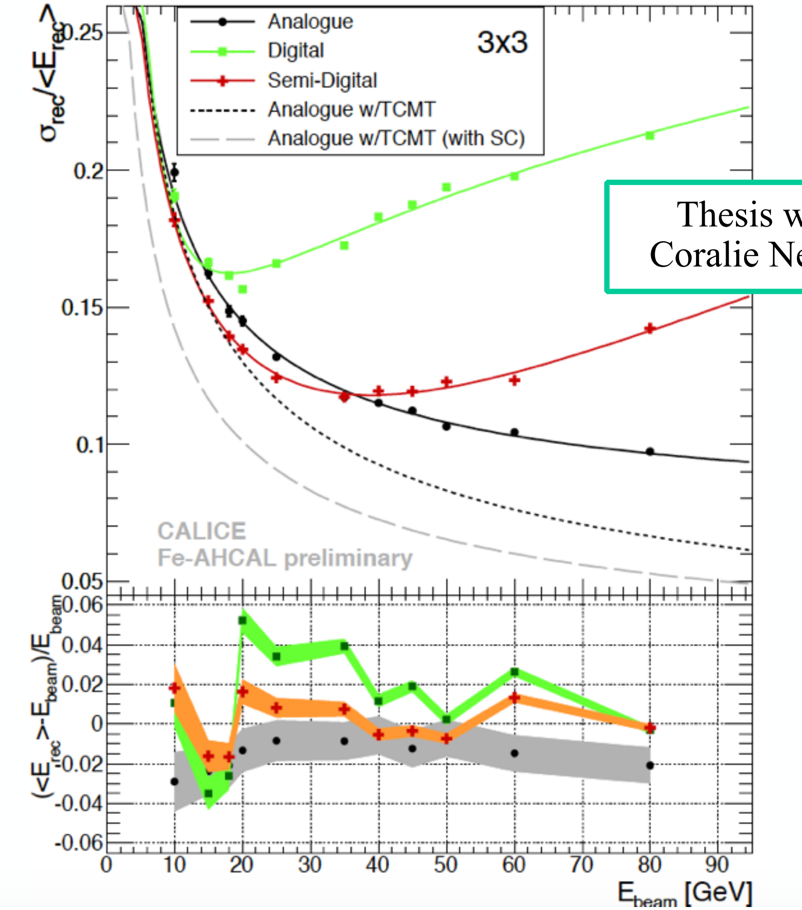
Towards cost optimisation

- Look at jet energy resolution as a function of number of channels
- Plot shows clear preference for 3x3 *cm2* cell size
- Software compensation to be applied



Semi-digital Reconstruction

- Semi-digital reconstruction is particularly successful at low energies
  - Counting hits at 3 thresholds N_1, N_2, N_3



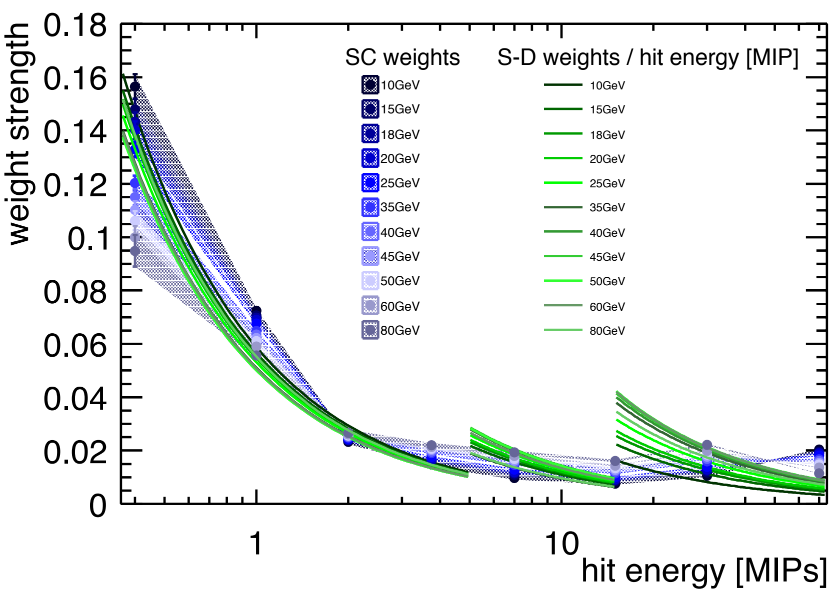
- Reconstructed energy: $E_{SD} = \sum_{bins} \alpha_i . N_i$

or

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

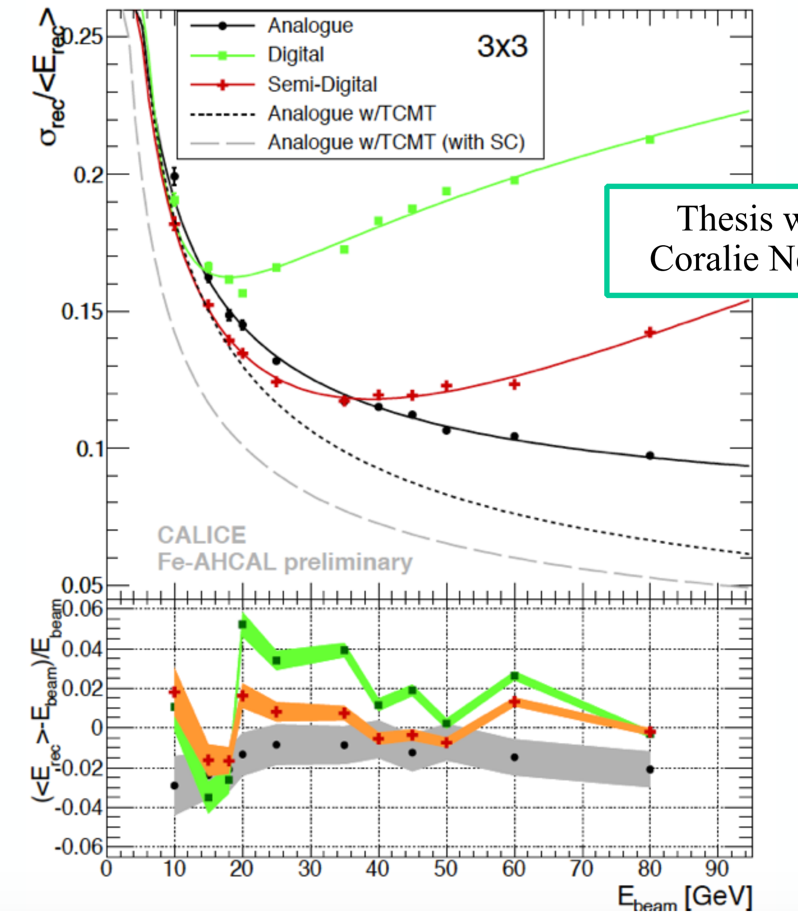
- Both reconstruction methods in same formalism
- Understand differences and learn from each other

Thesis work in progress
Coralie Neubueser (DESY)



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: $E_{SD} = \sum_{bins} \alpha_i . N_i$

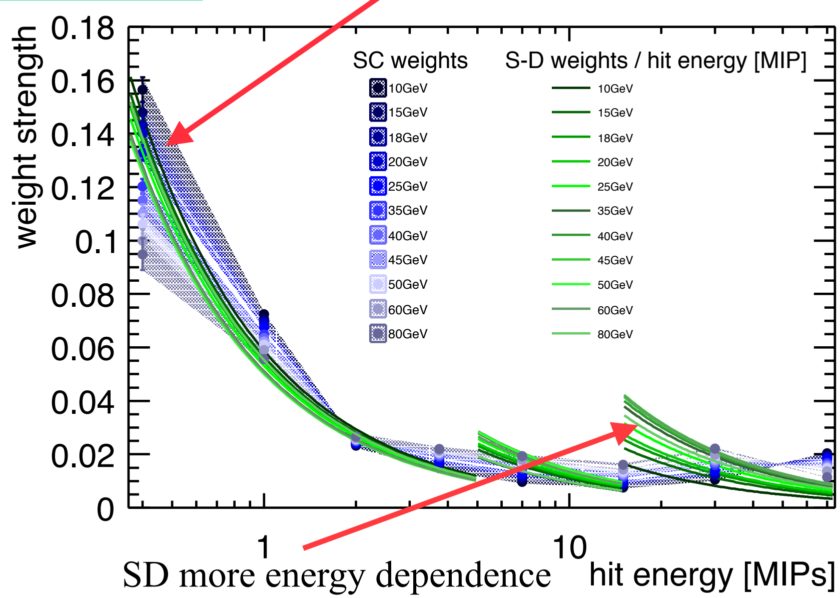
or

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

- Both reconstruction methods in same formalism
- Understand differences and learn from each other

Thesis work in progress
Coralie Neubueser (DESY)

SD less particle energy dependence at small hit energy



SD more energy dependence at high hit energy

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

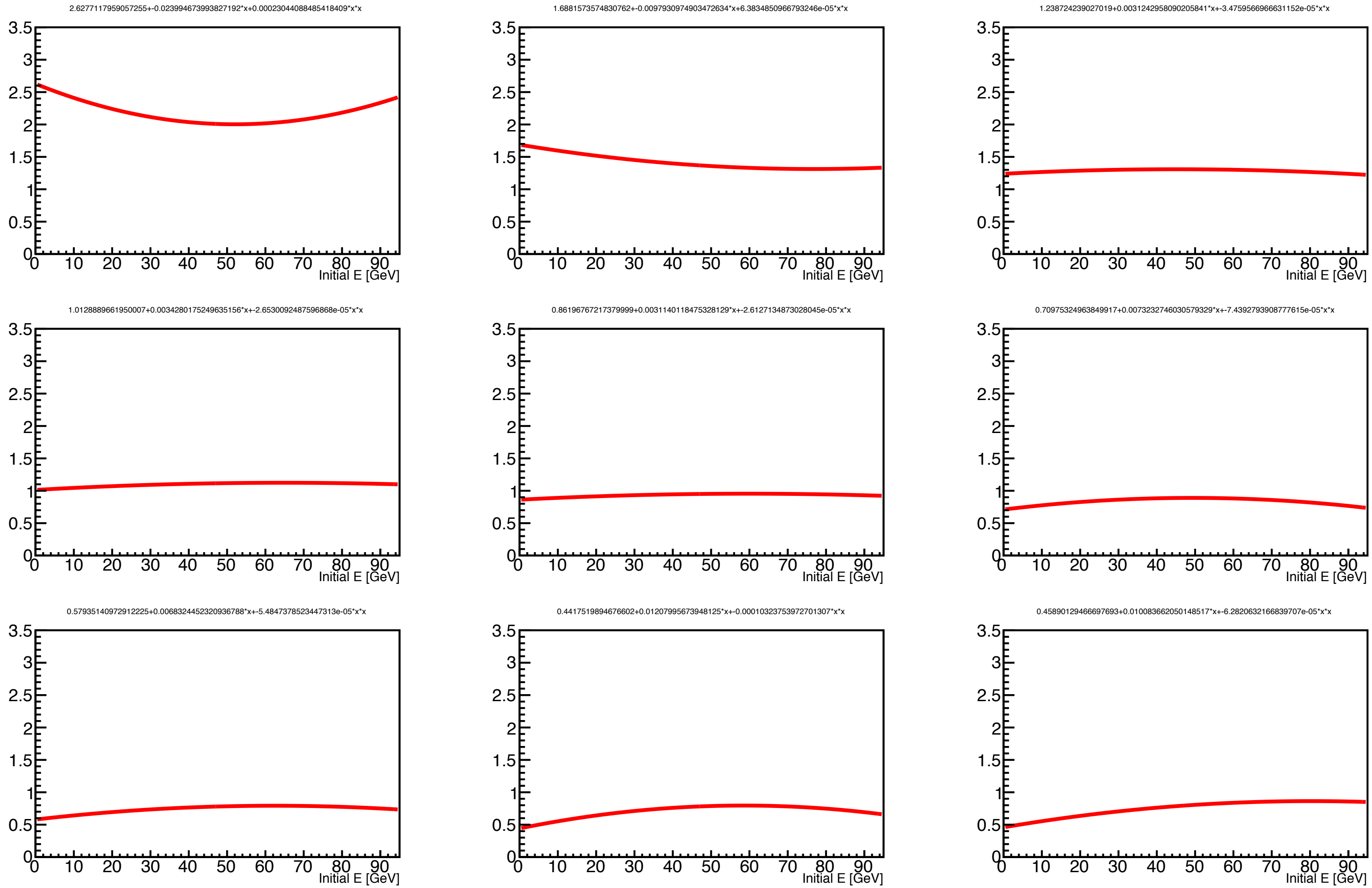
beta = beta1 + beta2*N + beta3*N*N

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

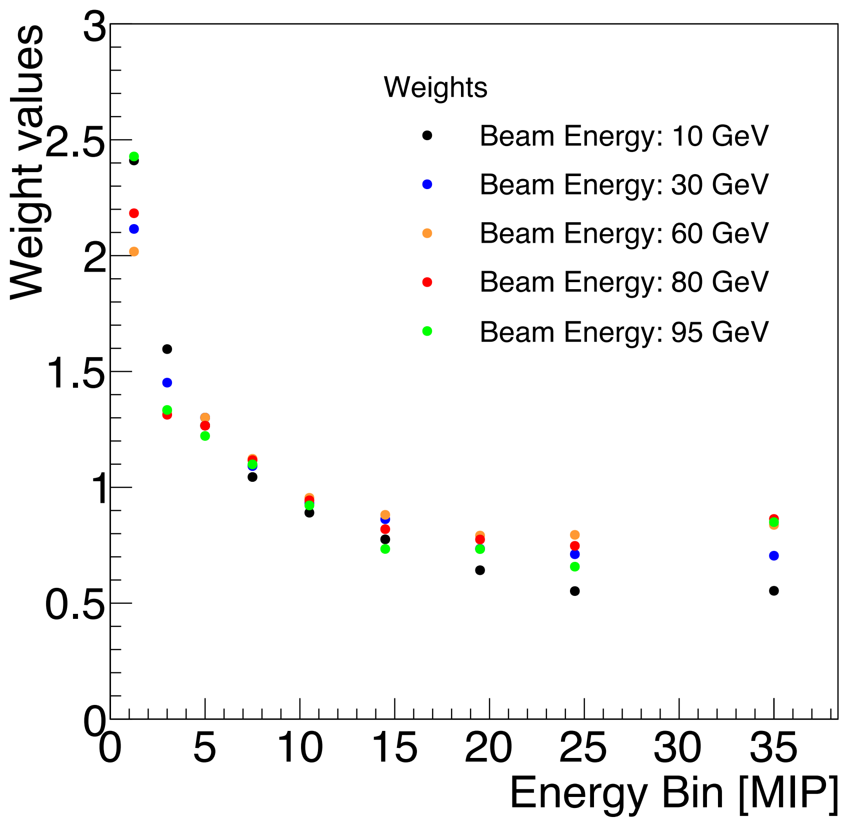
Software compensation mimics Semi-Digital:

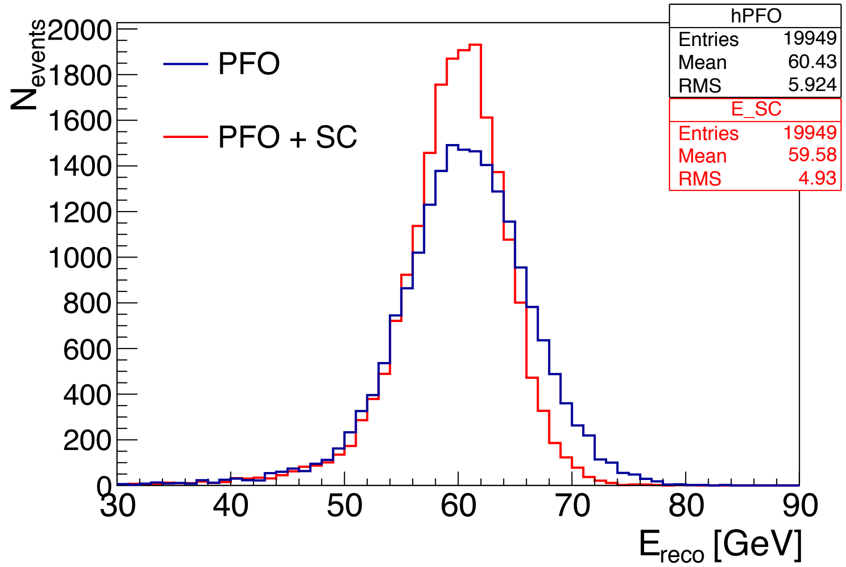
- Define bin
- Energy total = Sum_bin (weight_bin * SumEnergy_bin)
- weight_bin = a + b*E + c*E*E

Semi-digital Reconstruction



Semi-digital Reconstruction





JER vs cell size