Hadronic Energy Reconstruction: Software Compensaton in the AHCAL

Frank Simon, **Katja Seidel**
MPI for Physics & Excellence Cluster 'Universe'
Munich, Germany

CALICE Collaboration Meeting, Arlington, TX, USA, March 2010

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

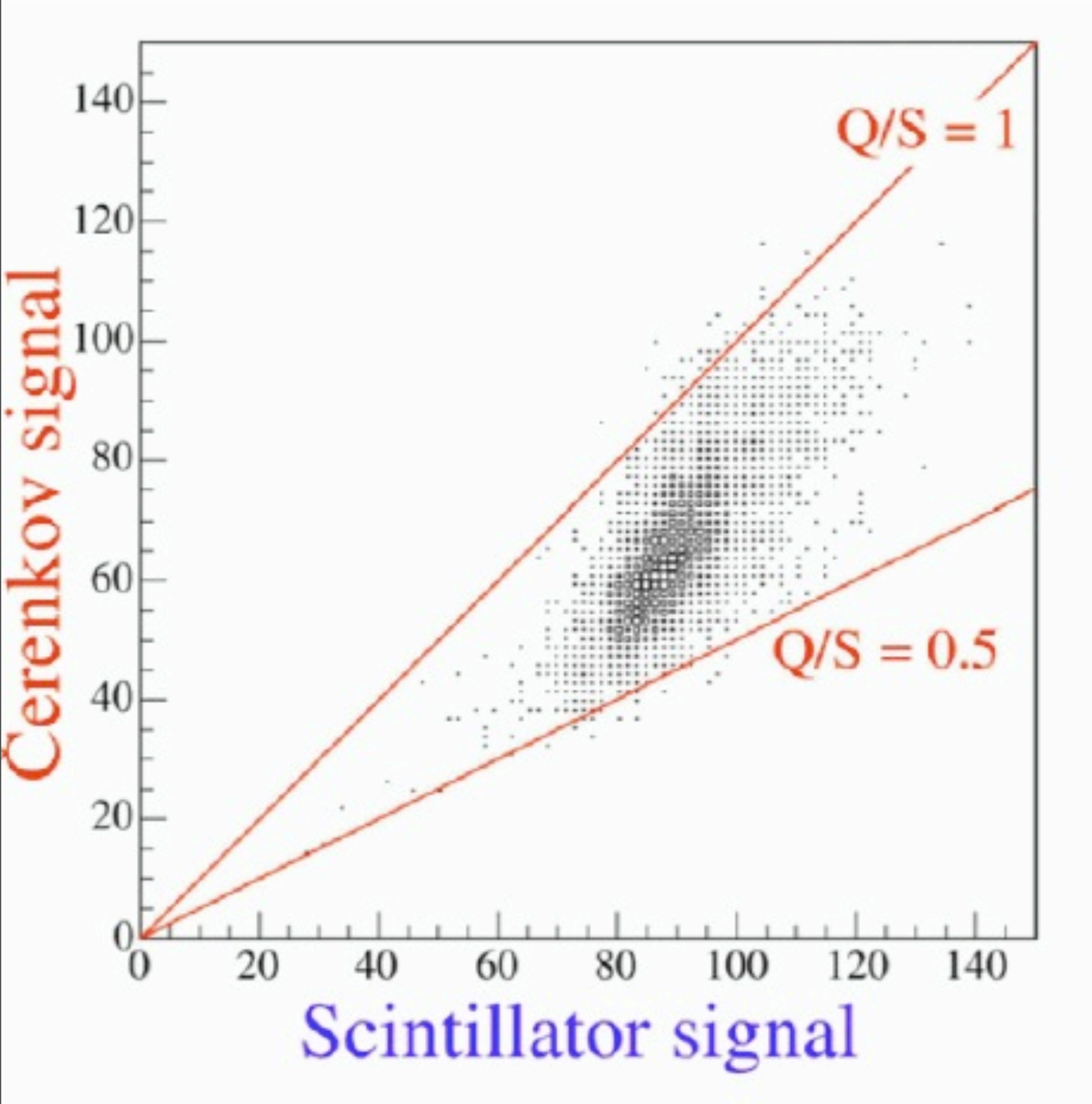
Calorimeter for ILC

Excellence Cluster
Universe

Overview

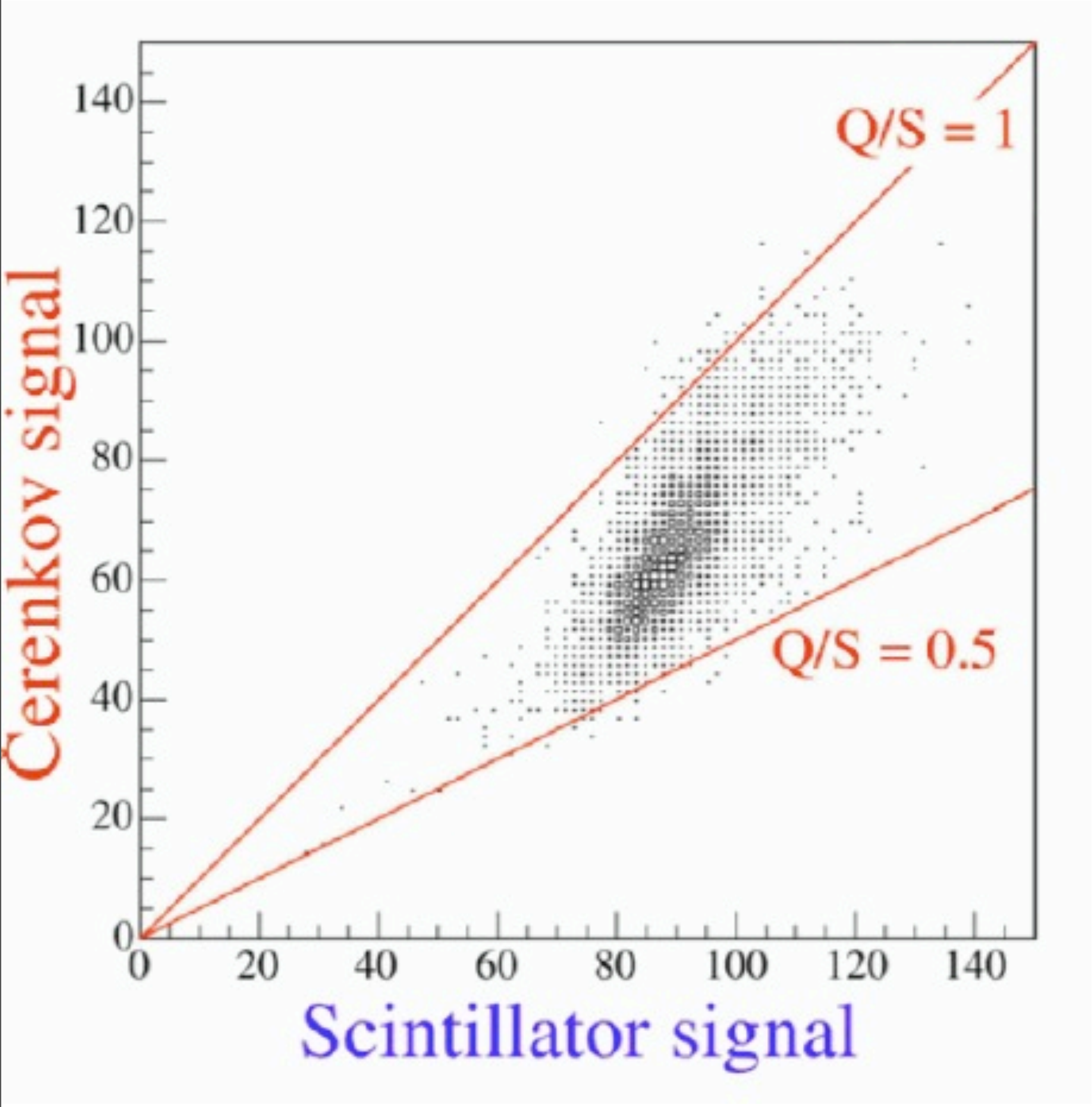
- Software Compensation: Why it works
- New approach: Cluster-based compensation
  - Simple weighting: Single weight per shower
  - Neural Network
- Summary / Outlook

DREAMing of Compensation

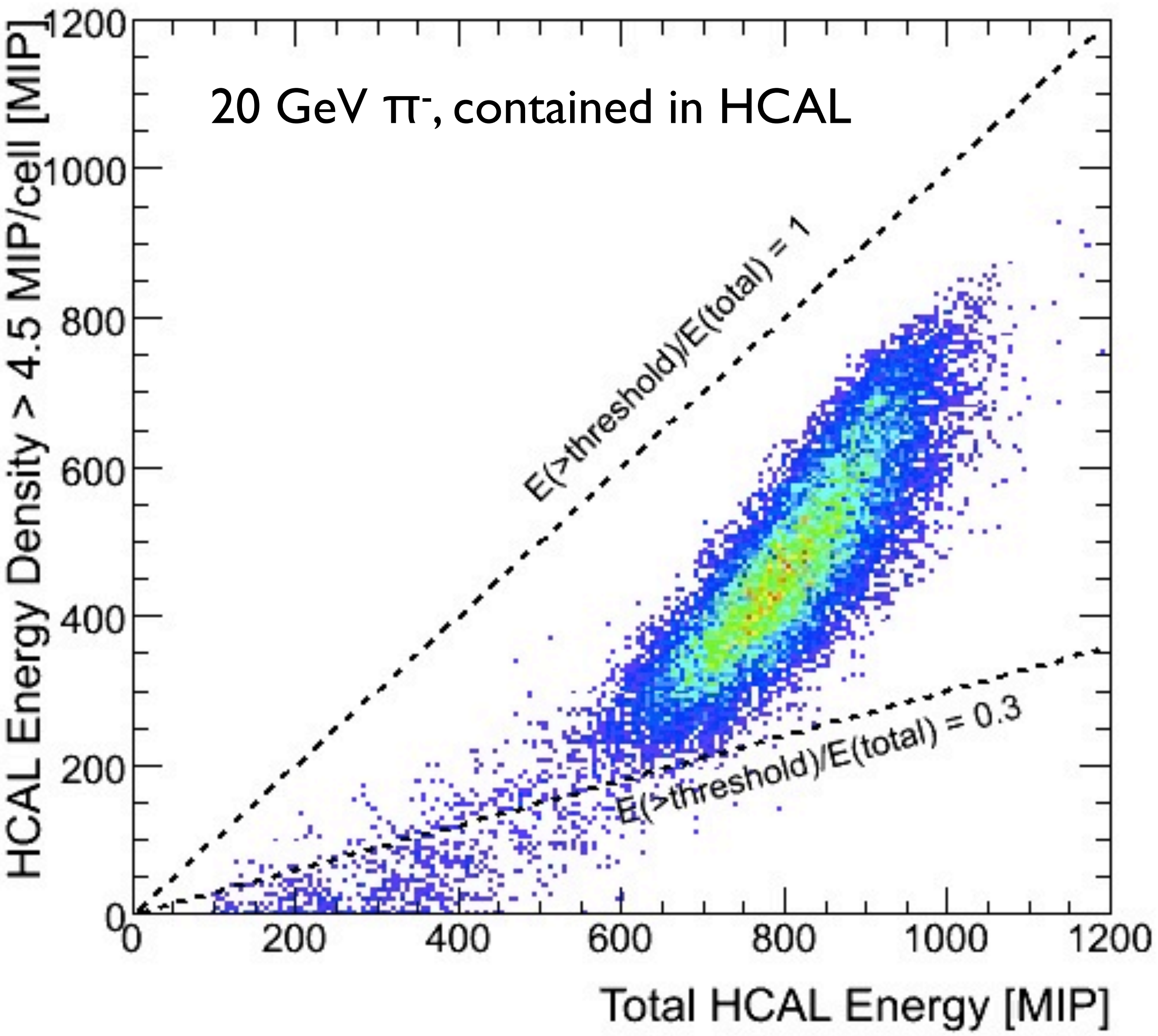


The DREAM "money plot": the reconstructed energy given by the scintillator signal can be improved with the Cherenkov signal (e.m. component) since the slope of the distribution is ≠ 1

DREAMing of Compensation

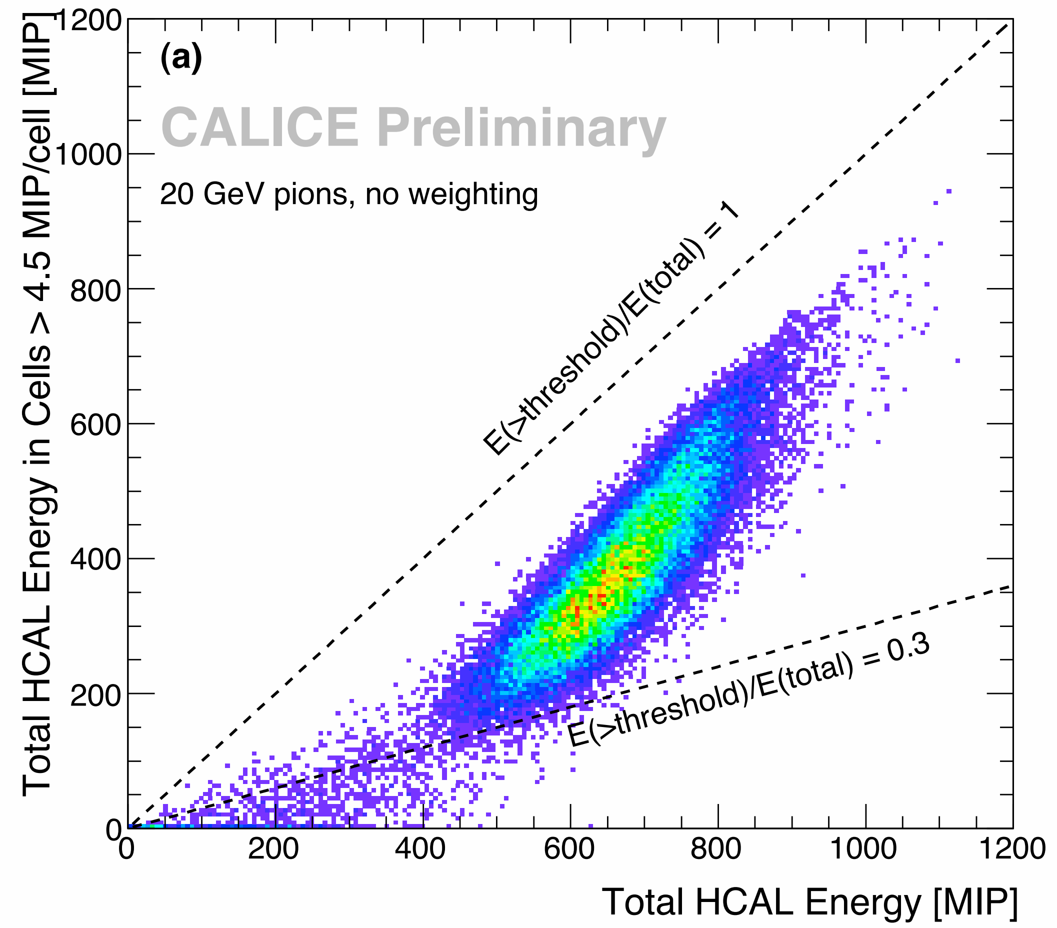


The DREAM "money plot": the reconstructed energy given by the scintillator signal can be improved with the Cherenkov signal (e.m. component) since the slope of the distribution is ≠ 1

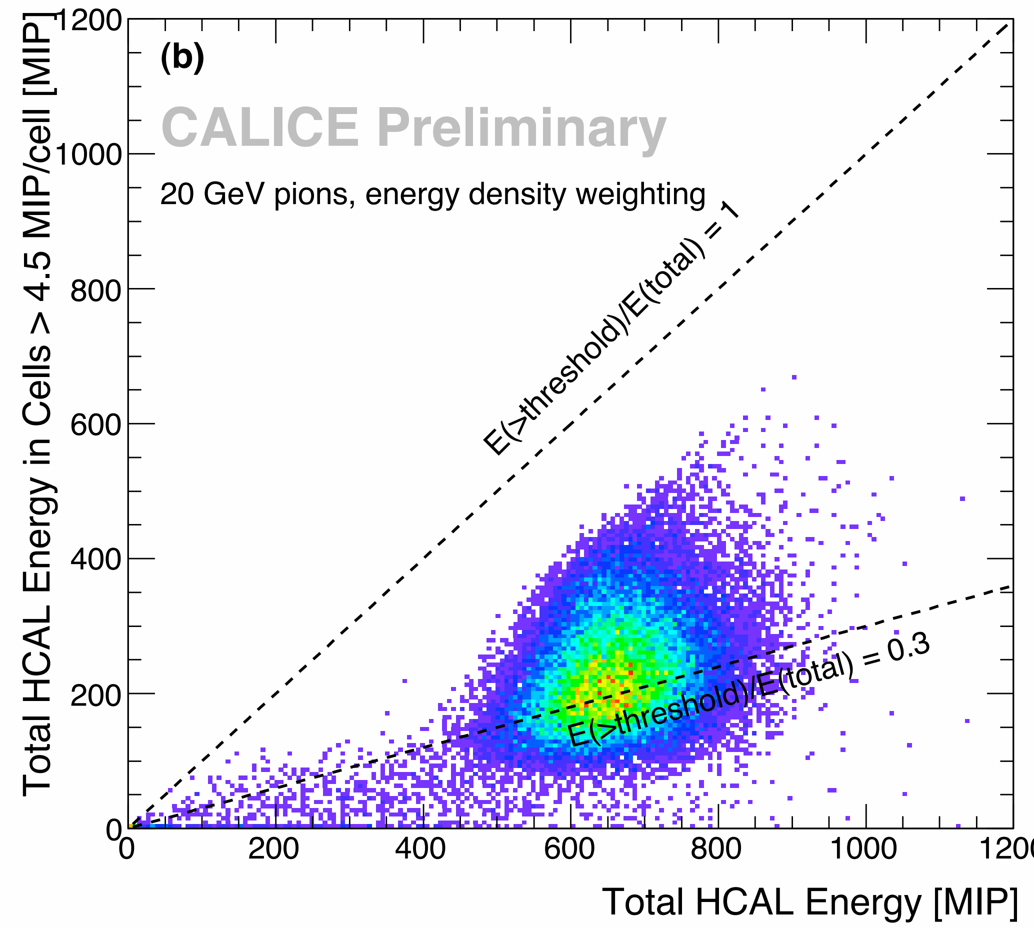


Local energy density works pretty much the same: events with a low total energy have a lower fraction of high density cells, this information can be used to improve the resolution: We can "DREAM", too...

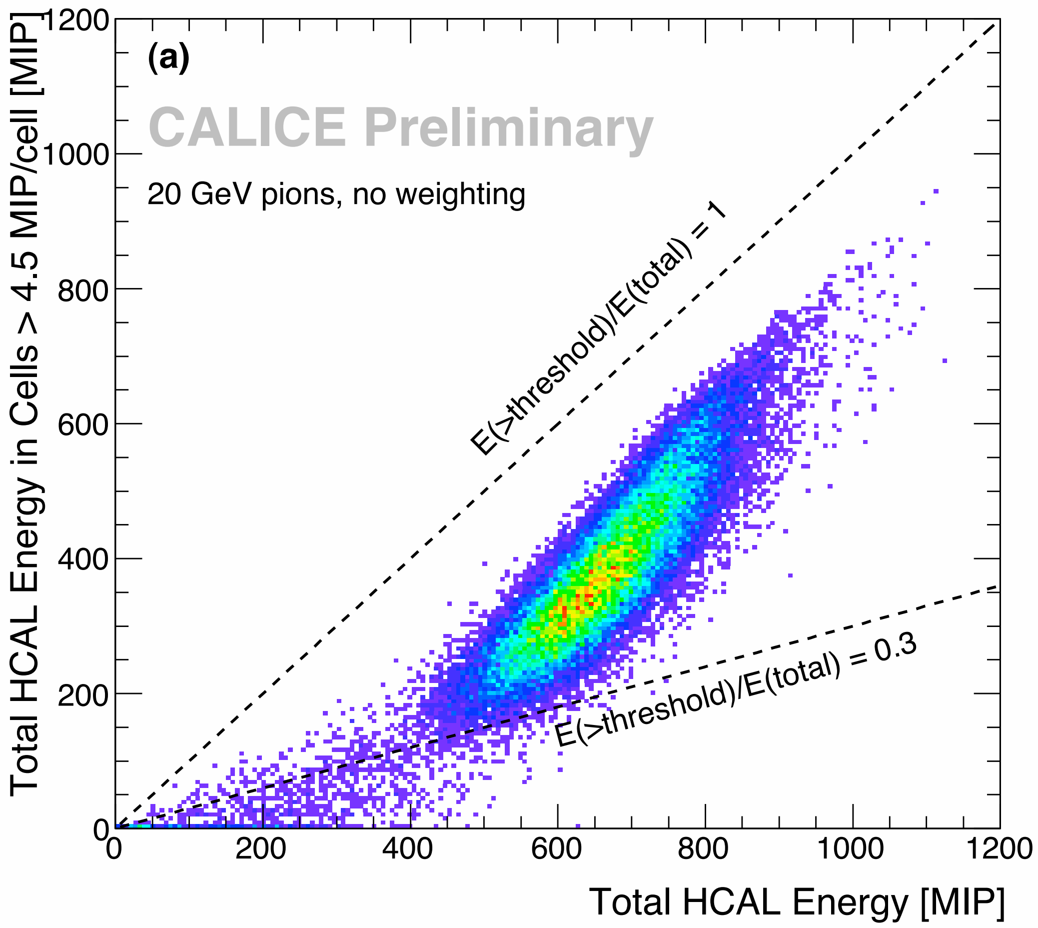
Software Compensation: How it works



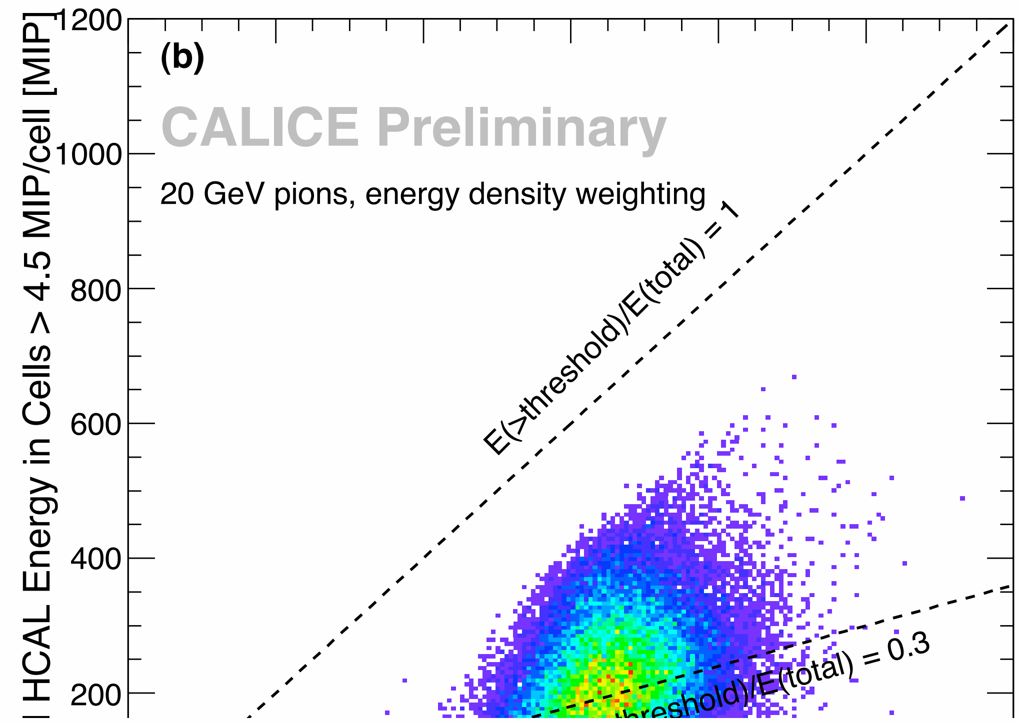
cell by cell weights



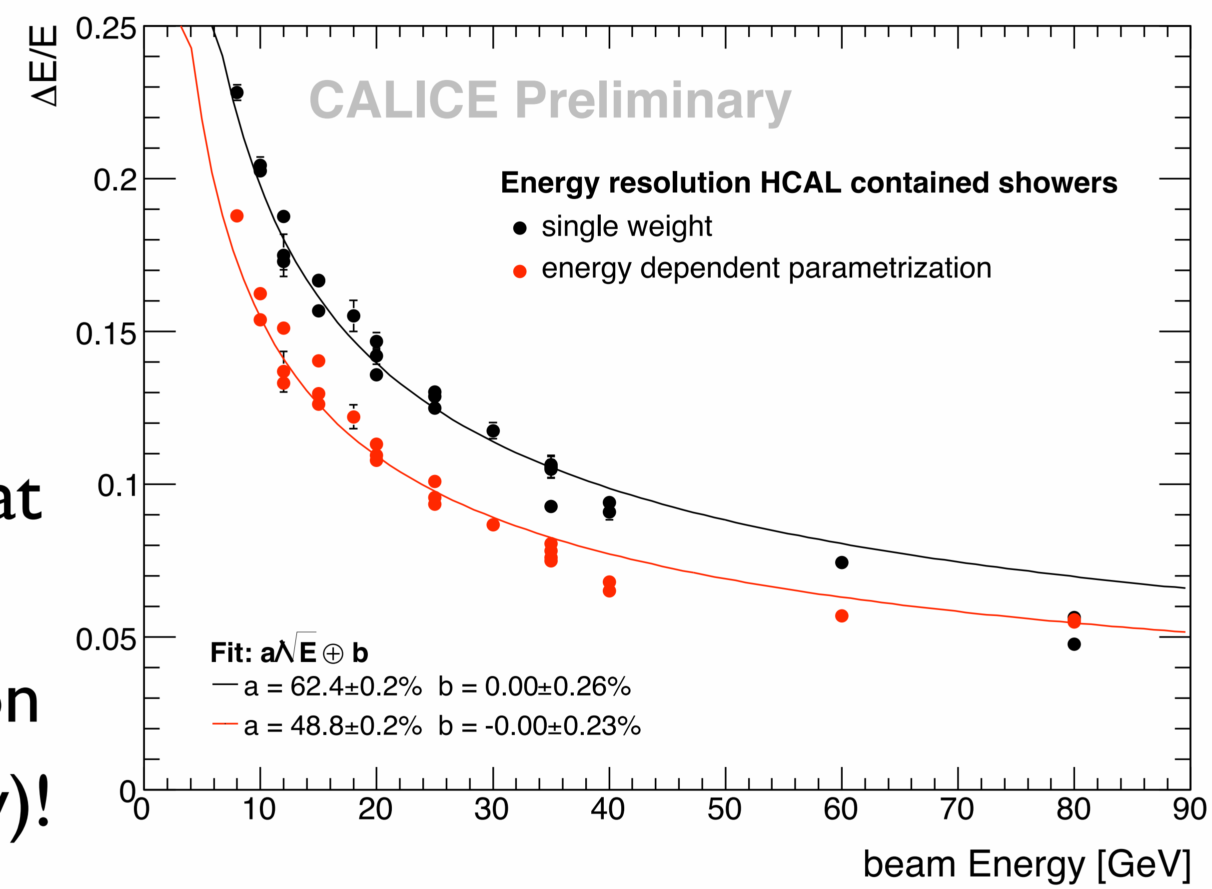
Software Compensation: How it works



cell by cell weights

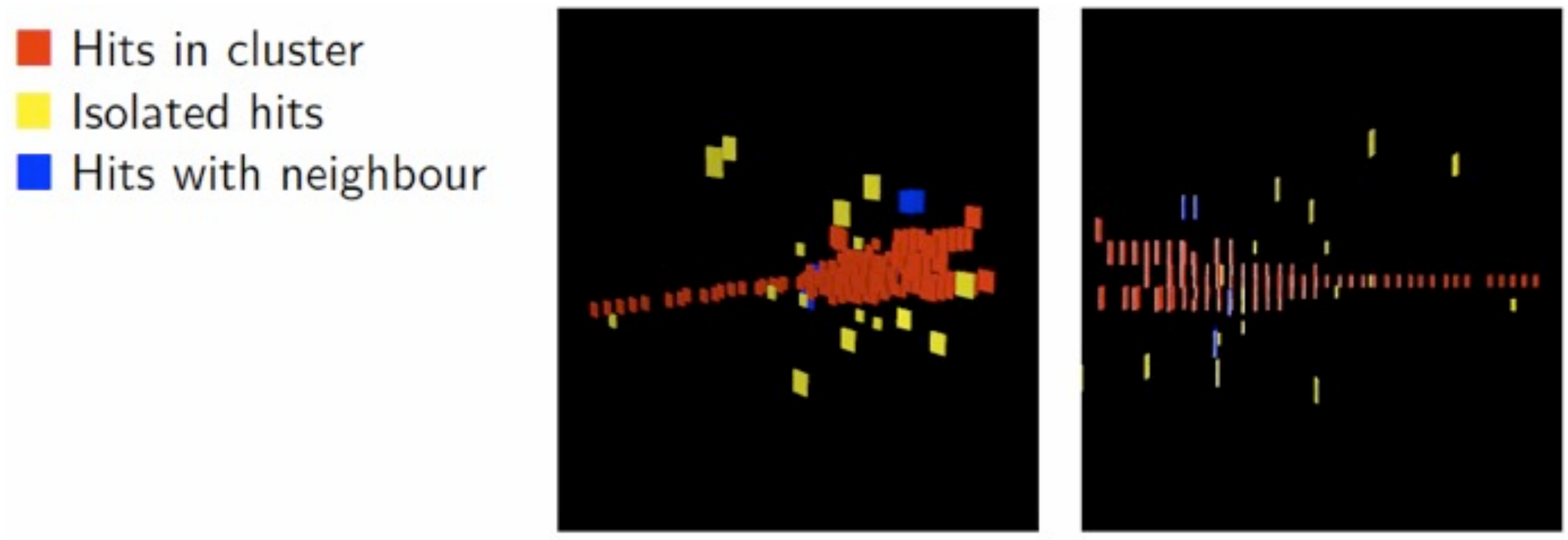


- Yields about 20% improvement in energy resolution
- Weights determined on data: Did not work at first try when using MC weights
  ➡ Large differences between data and MC on the cell by cell level (might be improved now)!



Alternative Approach: Cluster-based Weighting

- Identify all hits belonging to a shower (first simple approach)
  - project shower on the front face of the HCAL, find maximum as shower axis
  - in each layer expand from the axis until energy does not grow significantly

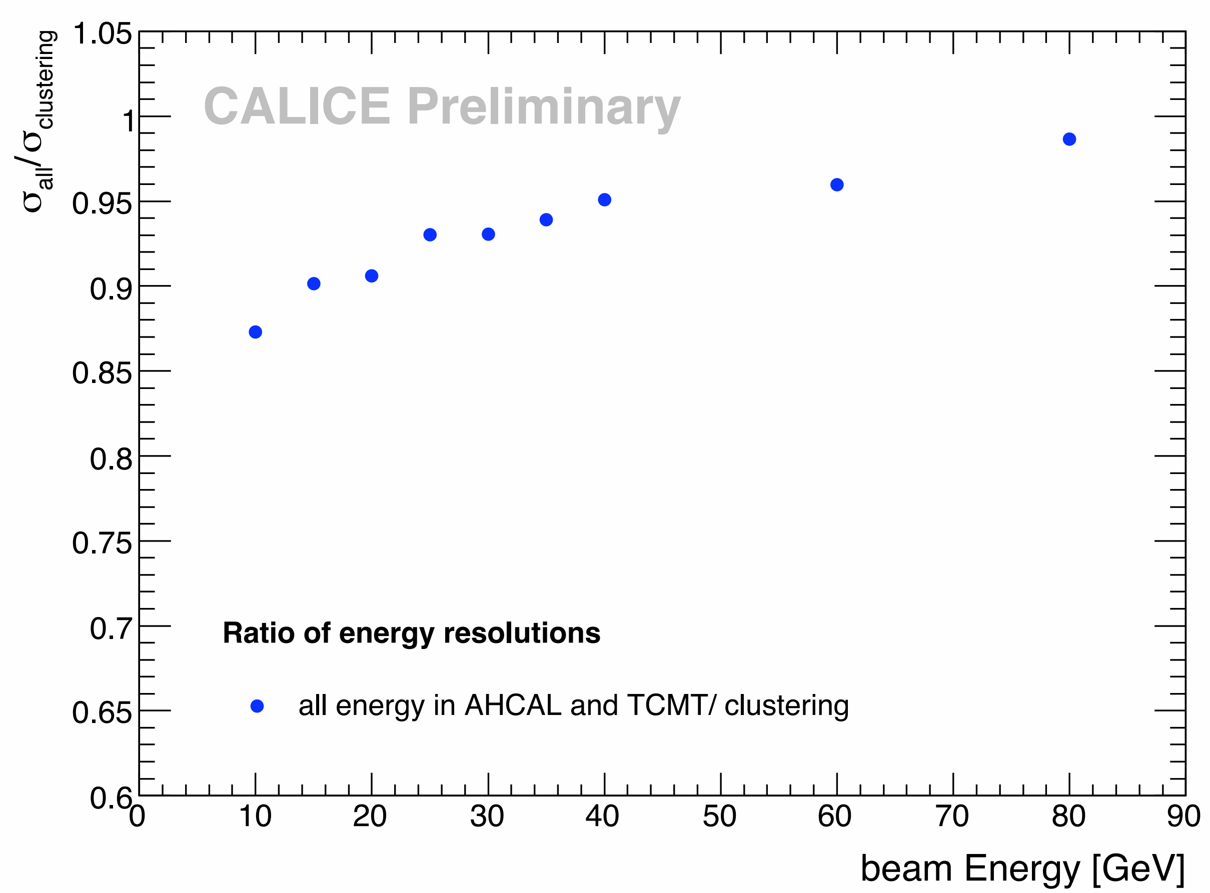


- Clustering in HCAL and TCMT, track required in ECAL

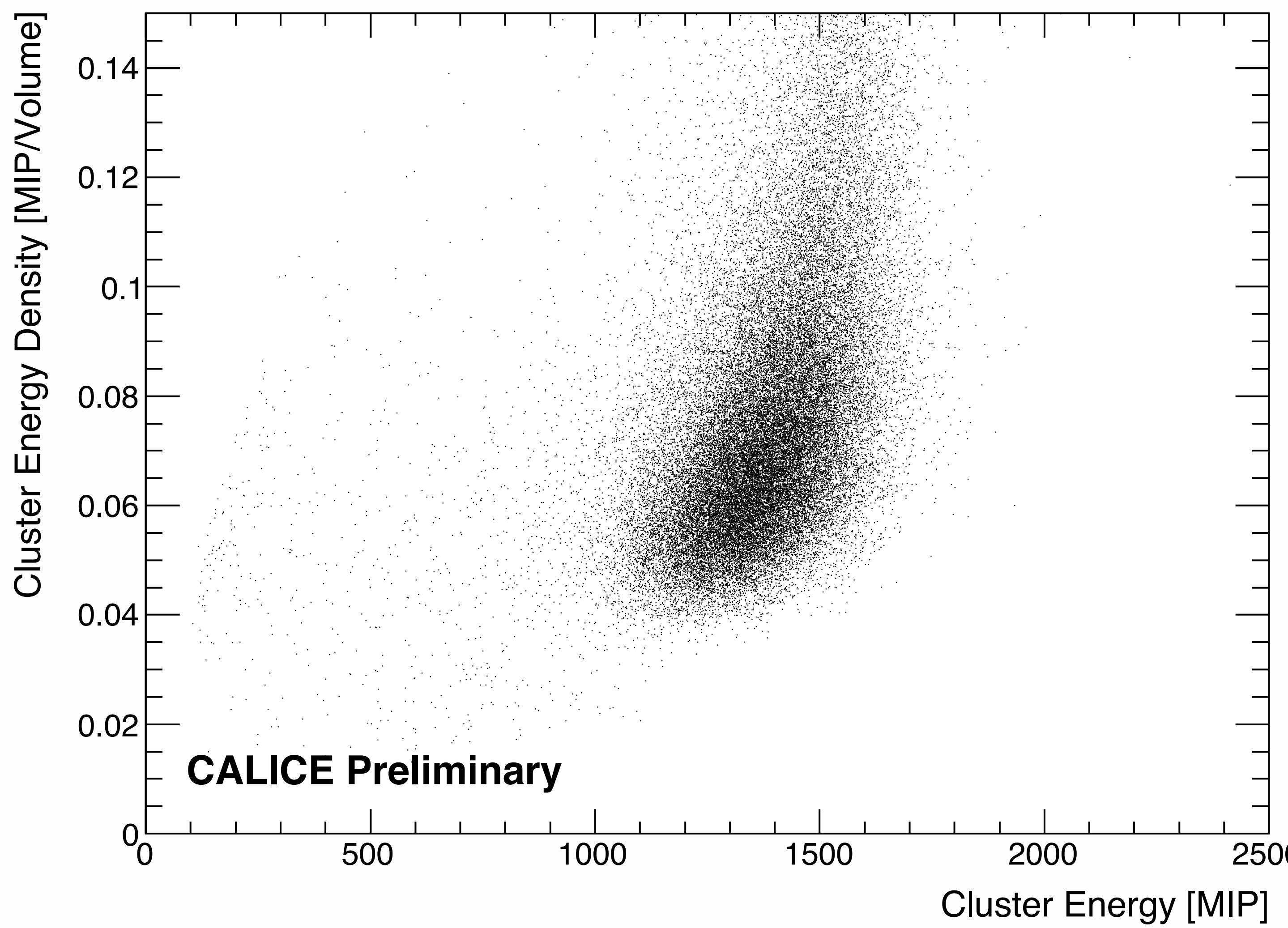
Motivation:
➩ Look at bulk properties of the shower: MC can be used to tune weights!
➩ Easily transferrable to PandoraPFA
But: Give up some of the information available in cell-by-cell weighting...

Clustering: Resolution & Sensitivity

- Decrease of energy resolution: In particular at low energy: some loss of information in the clustering



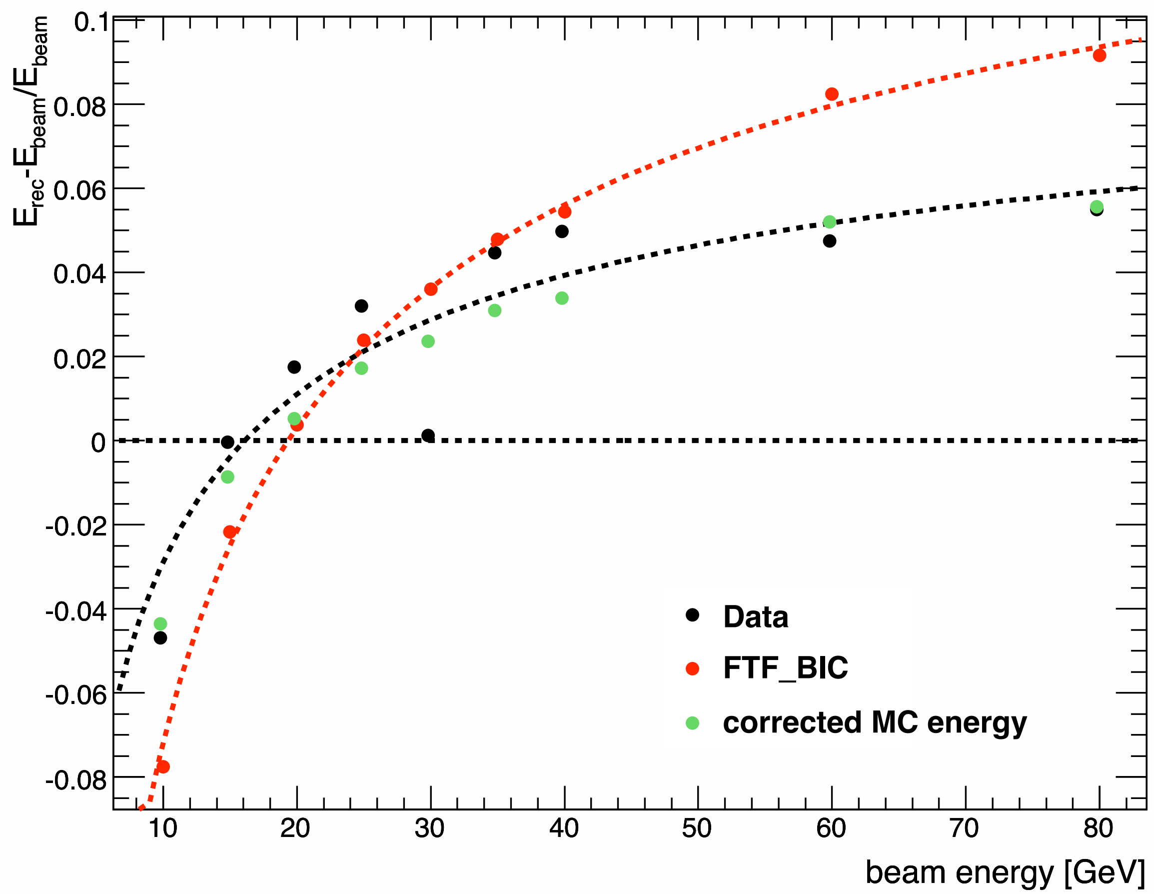
40 GeV pions:

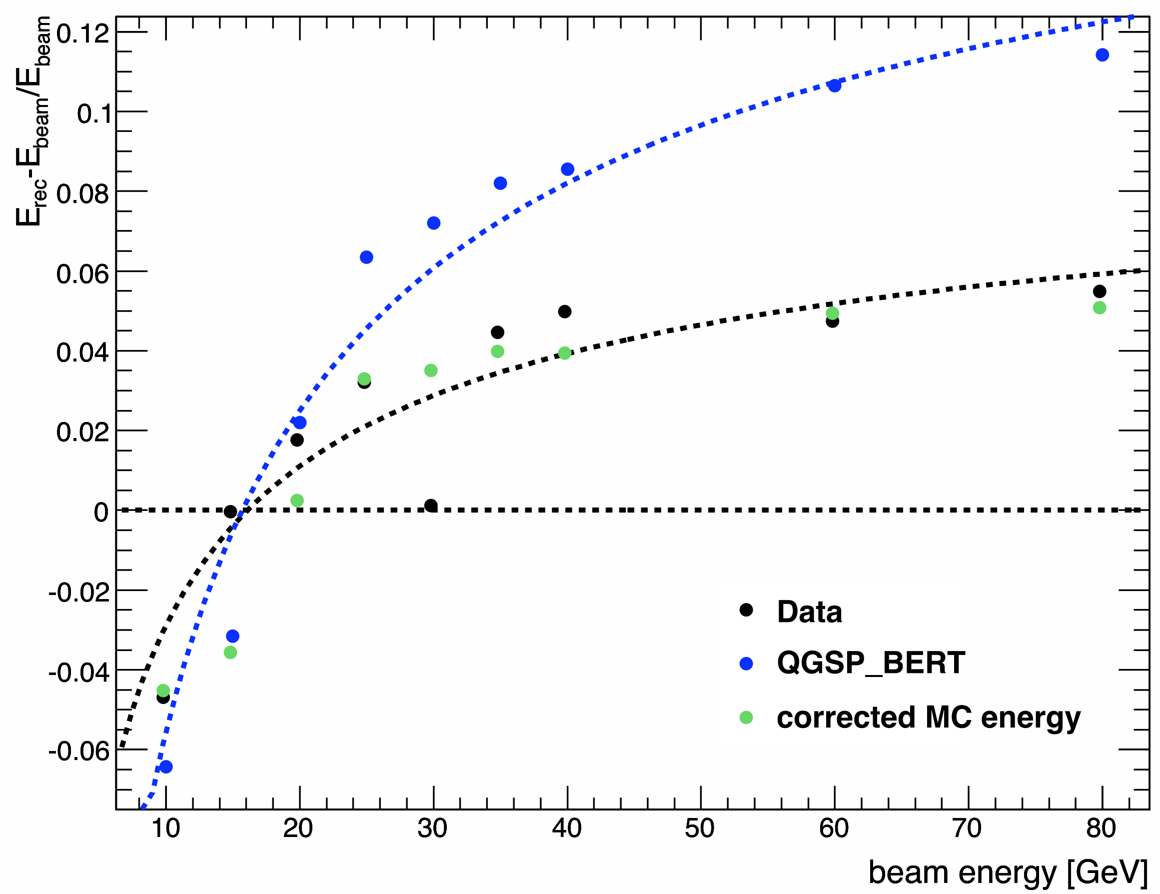


- Correlation of cluster energy density and reconstructed energy: The basis of software compensation

Data - MC Missmatch: Recalibration

- Observed discrepancy between reconstructed energy in data and MC
  - Leads to problems for the linearity of the response!



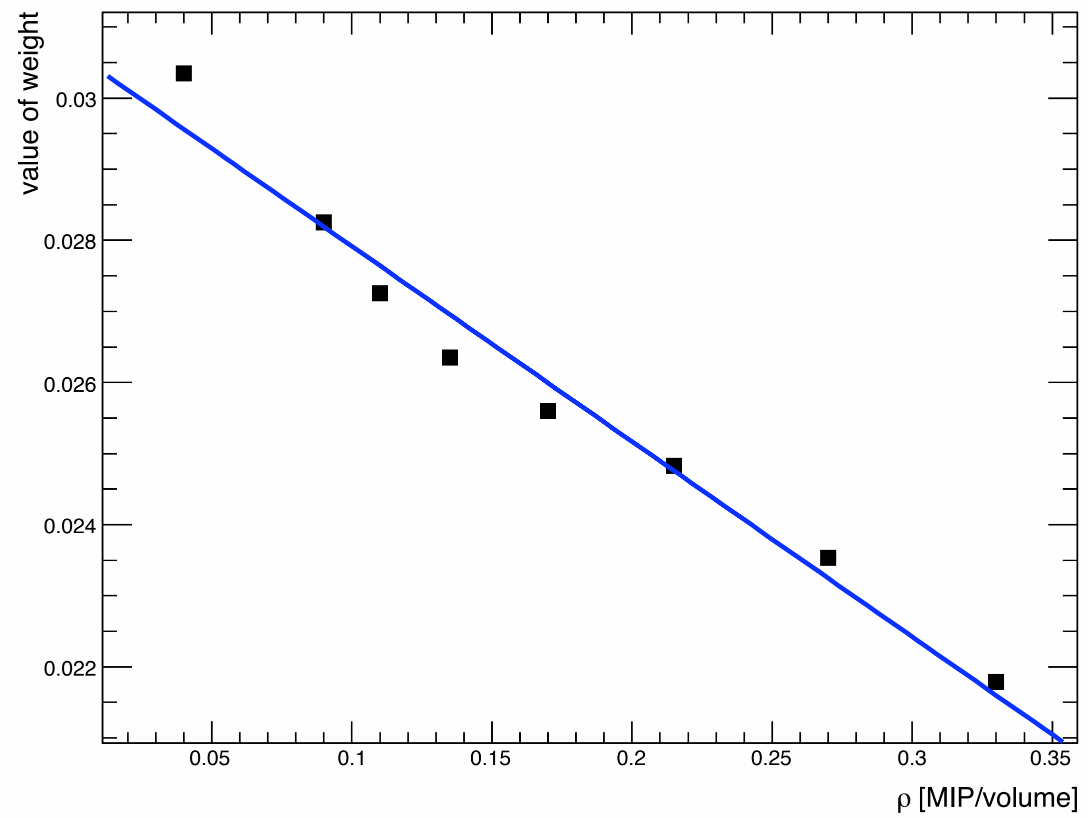


- Determine a correction factor for the MC energy from a fit to the observed difference, correct MC energy (no corrections to density etc.)

Simple Weighting: Weights based on Density

- Weights determined from simulated data using a minimization procedure (one weight per shower!)

$$E_{rec,weighted}[GeV] = \sum_{hit} E_{hit}[MIP] \cdot \omega(\rho, E) = E_{rec}[MIP] \cdot \omega(\rho, E)$$

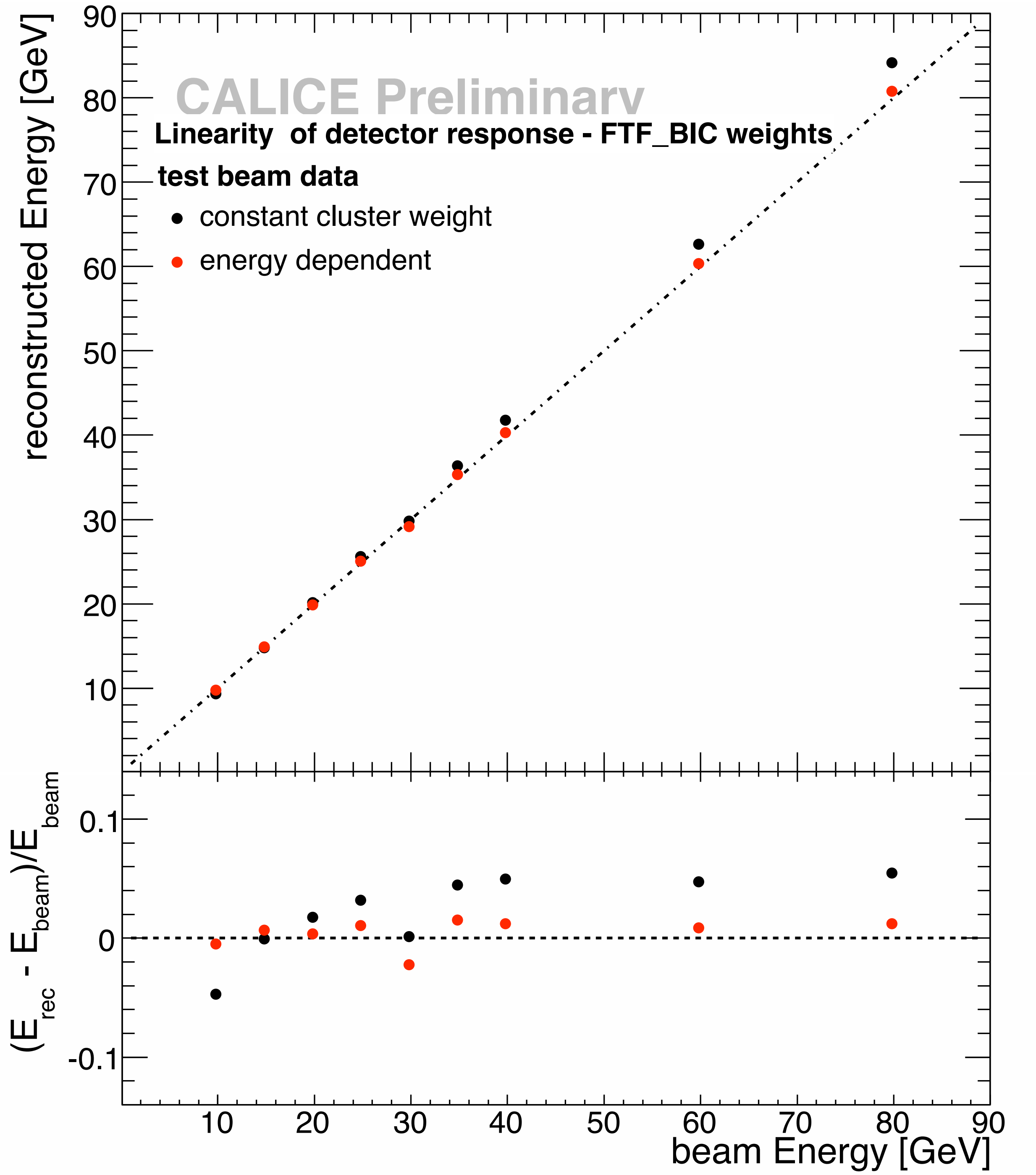


Weight as a function of shower density: 40 GeV run, determined from QGSP_BERT

Now apply the usual technique: Parametrize energy dependence, choose weights according to unweighted energy

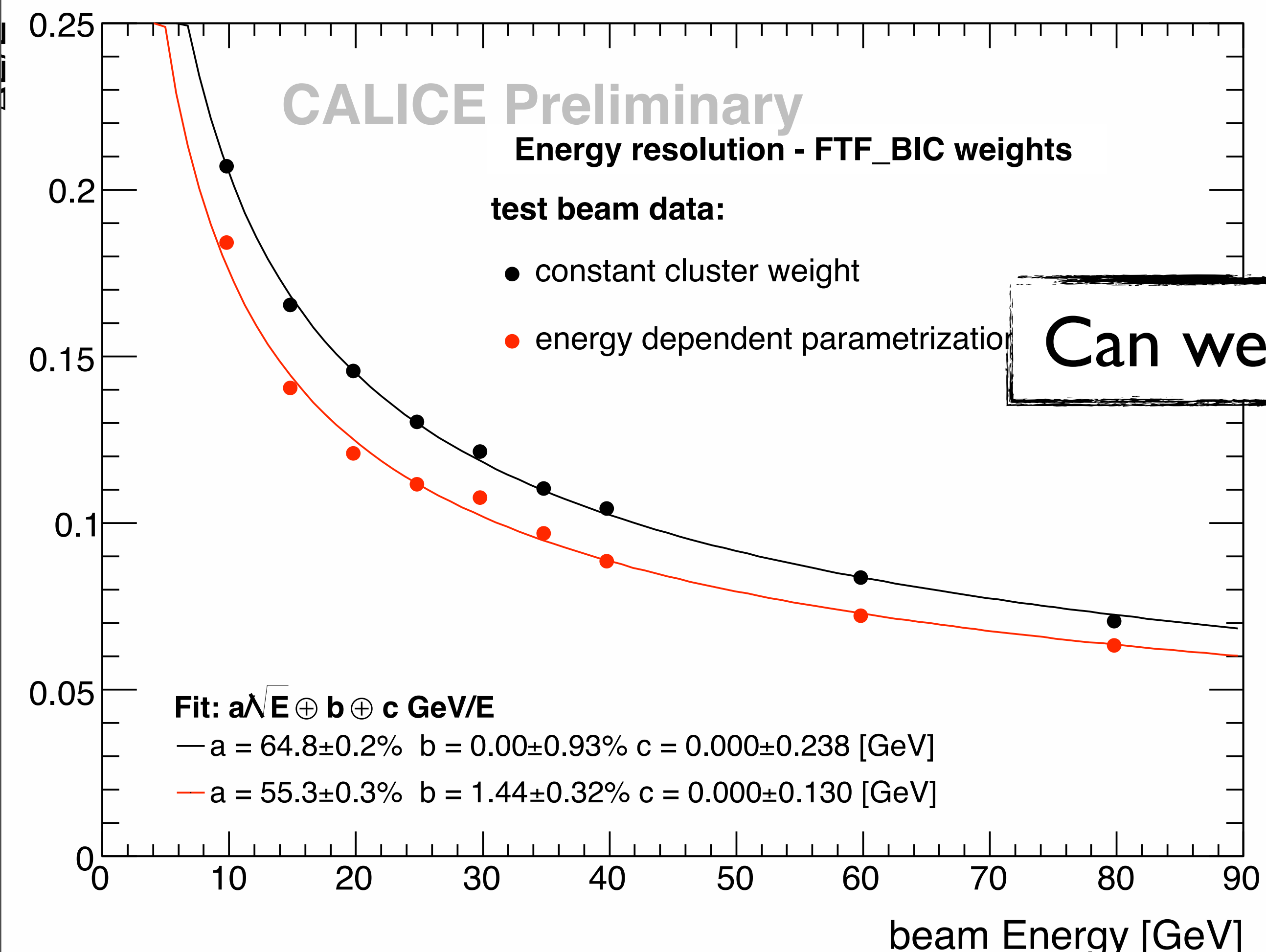
Simple Weighting: Performance - Linearity

- Significant improvement of linearity

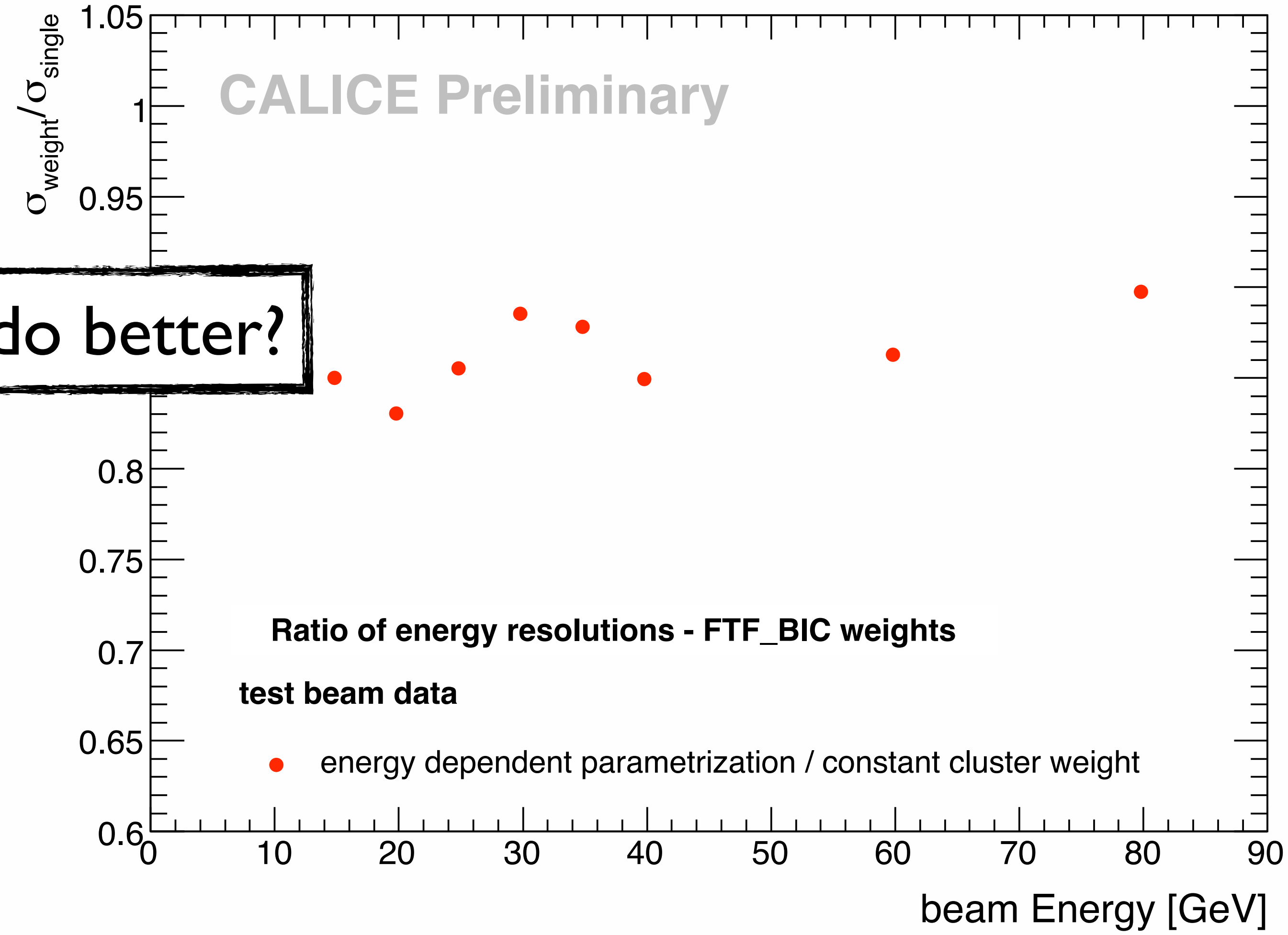


Simple Weighting: Performance - Resolution

- Resolution: Weights determined with FTF_BIC (similar results for QGSP_BERT)



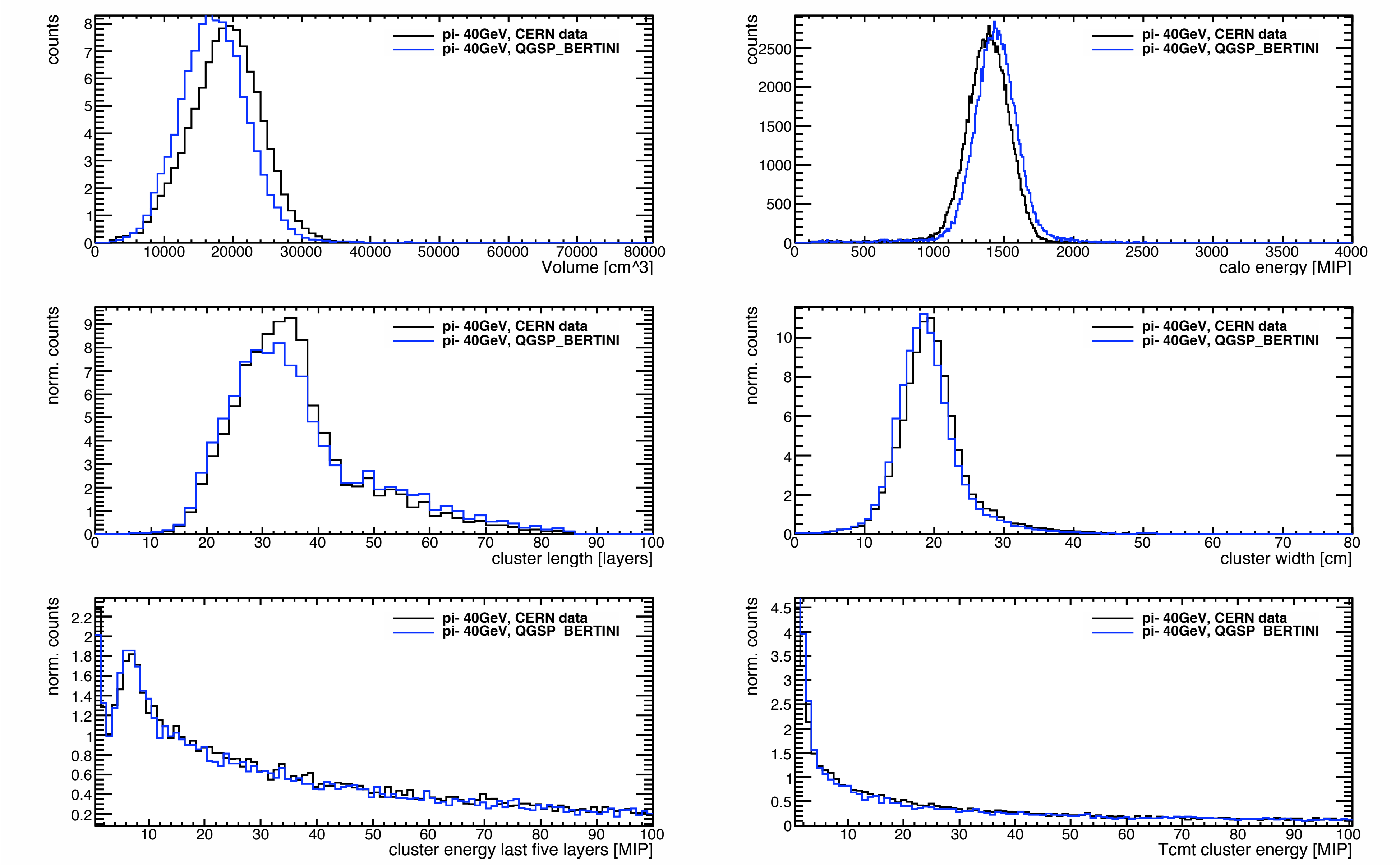
Can we do better?



10% to 15% improvement in resolution, best performance at intermediate energies: Leads to the constant term in the fit

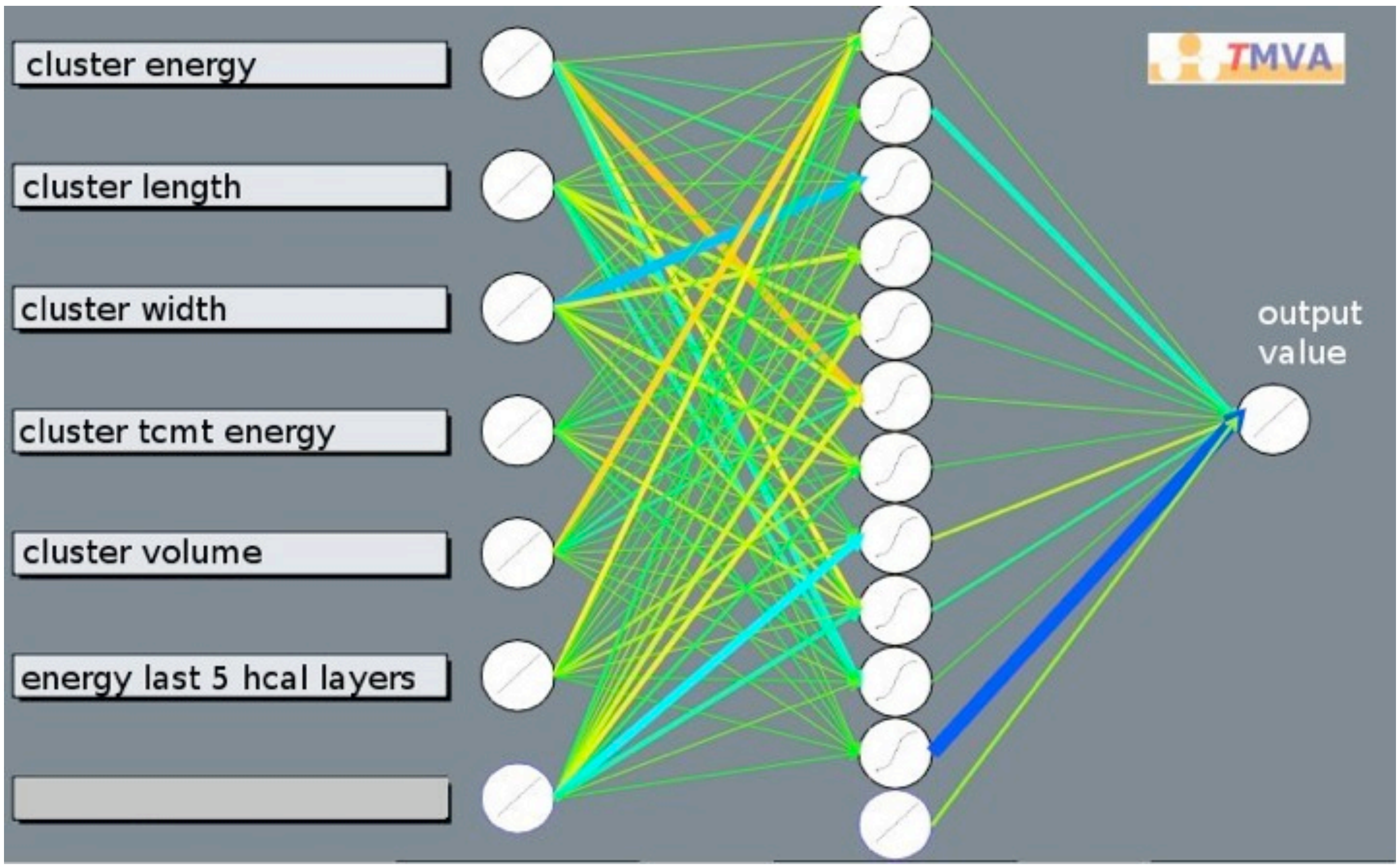
Advanced Weighting: Using a Neural Network

- Select 6 shower properties that are sensitive to reconstructed energy and energy density of shower, use as NN inputs



Neural Network: Training

- Neural network trained on simulated data: Quasi-continuous energy distribution to avoid bias due to specific beam energies
  - from 5 to 105 GeV in 0.1 GeV steps



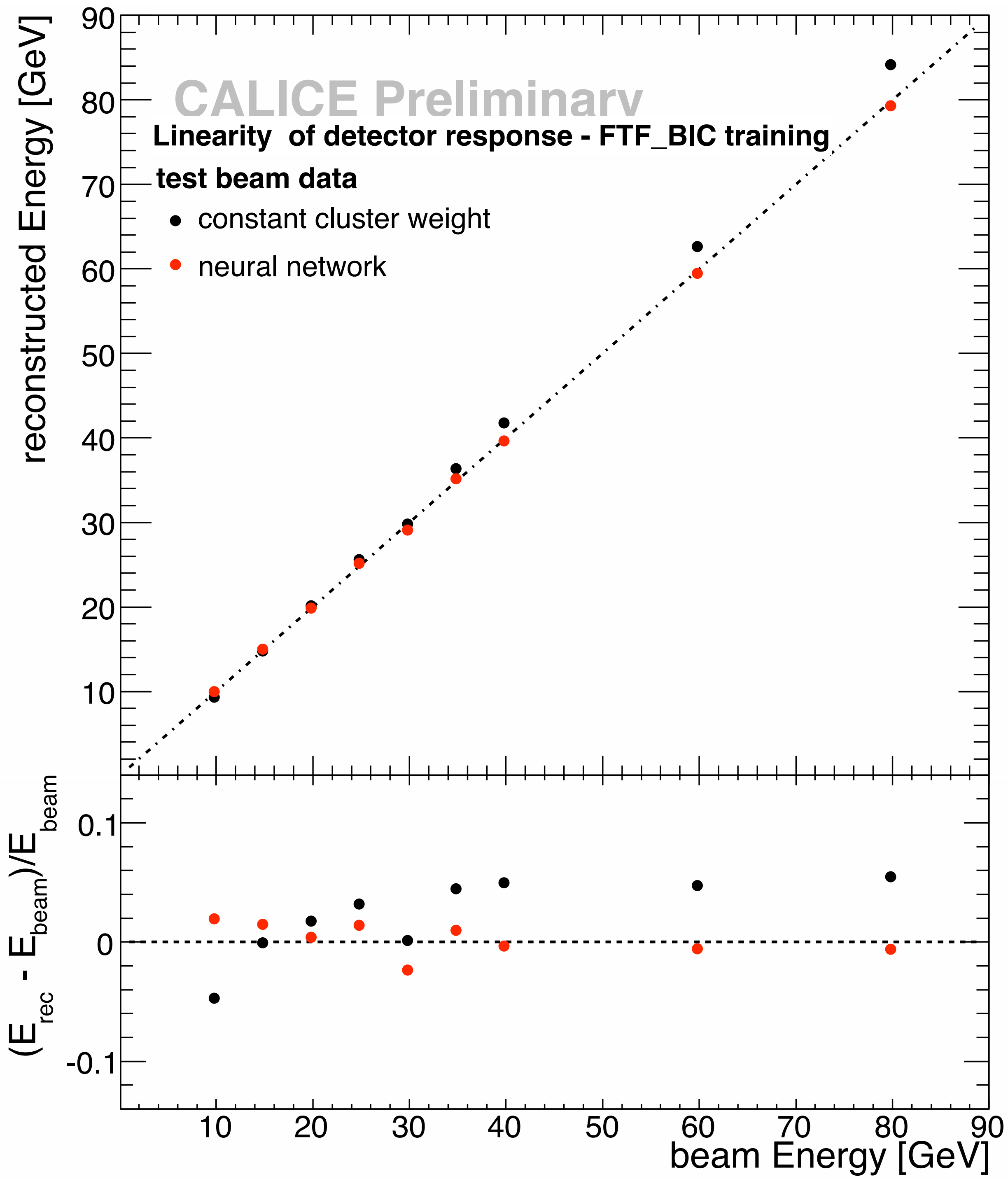
6 input variables

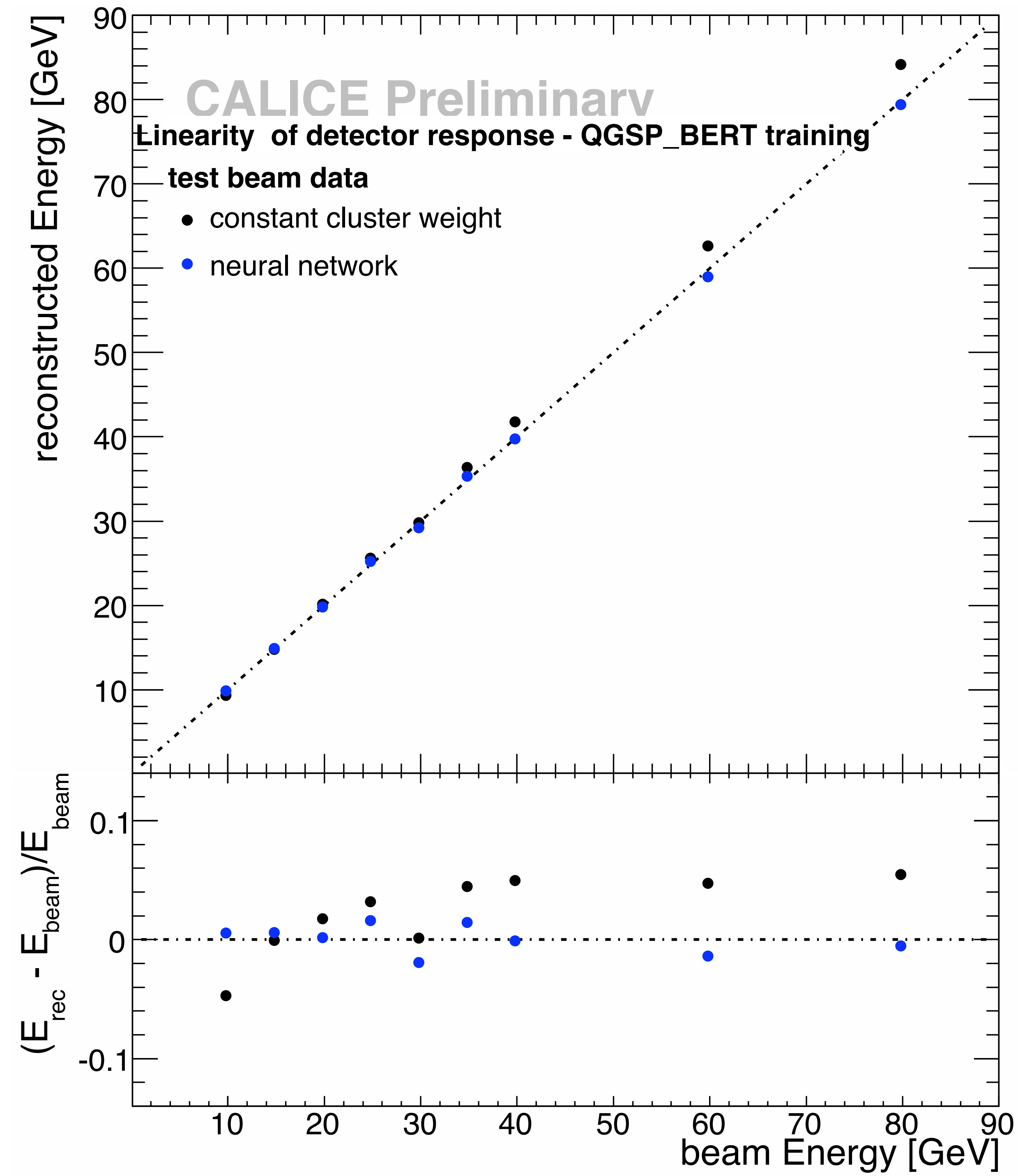
1 hidden layer
11 nodes

reconstructed energy as target value

Neural Network: Performance - Linearity

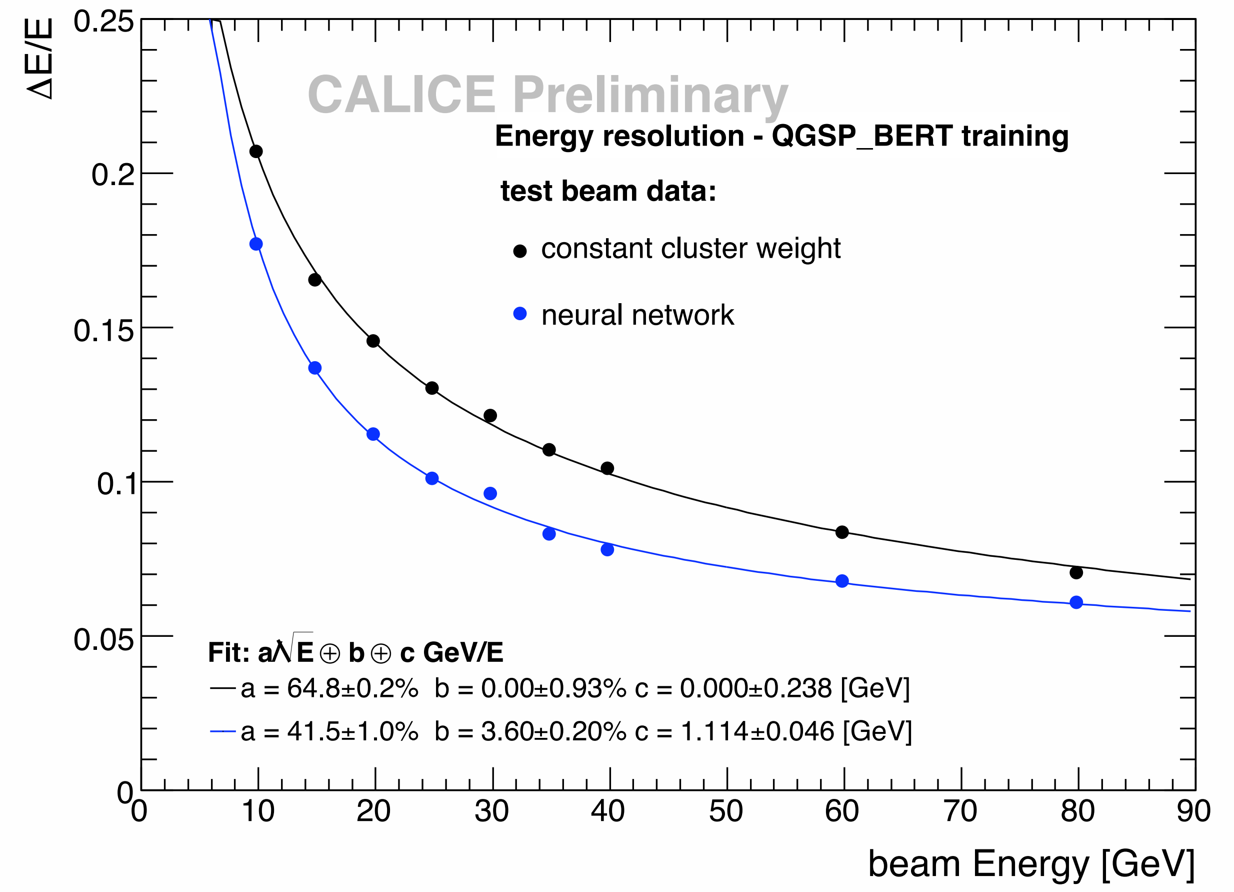
- Excellent linearity for both training with both physics lists

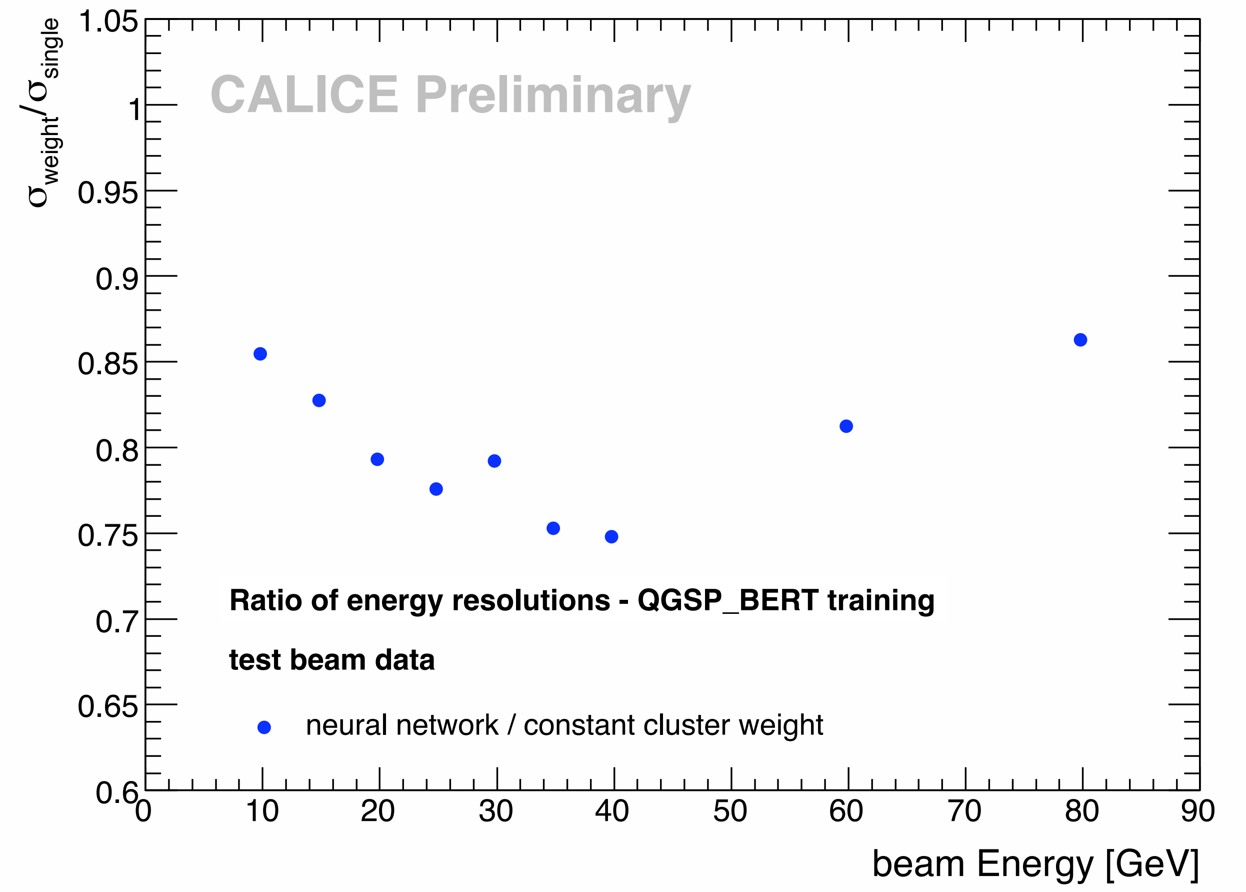




Neural Network: Performance - Resolution

- Training with QGSP_BERT

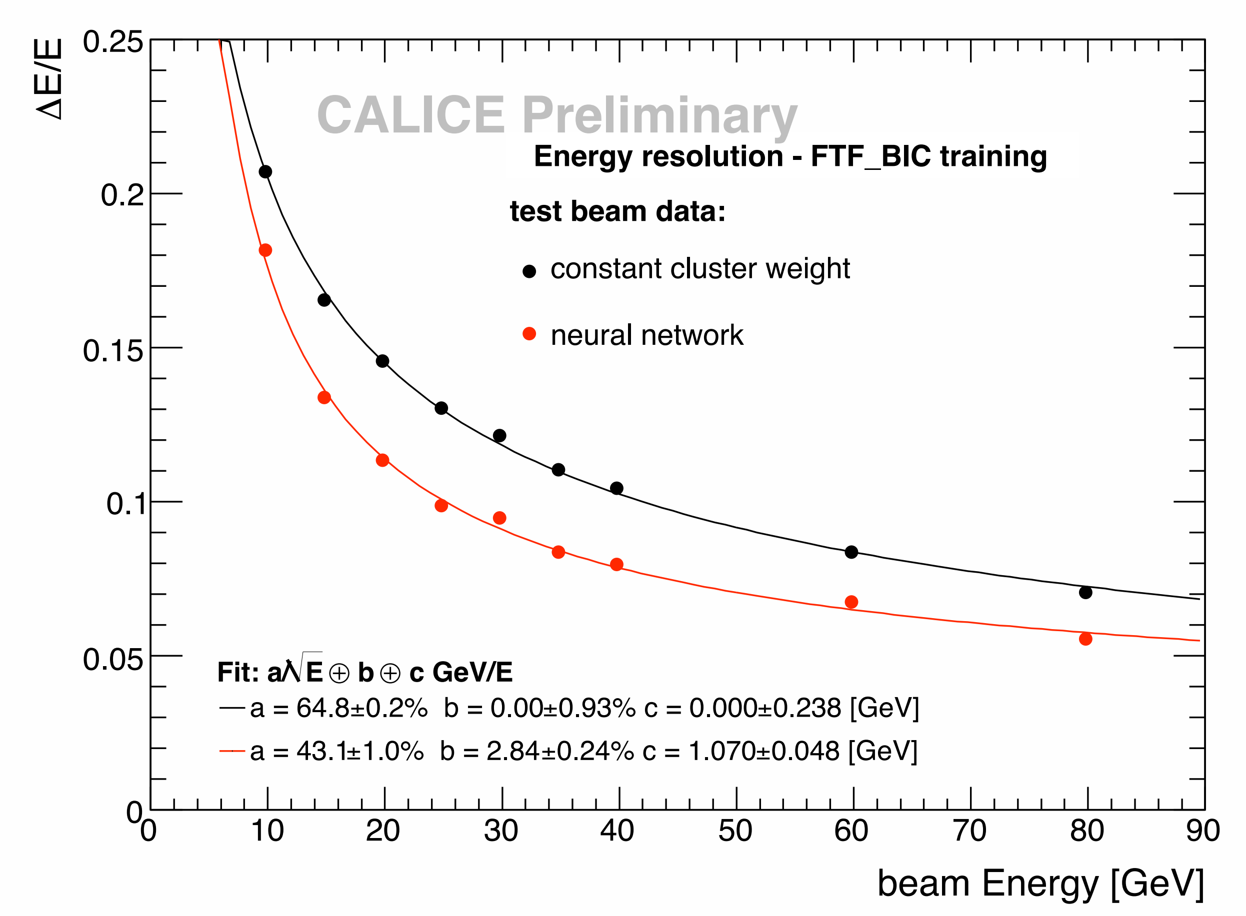


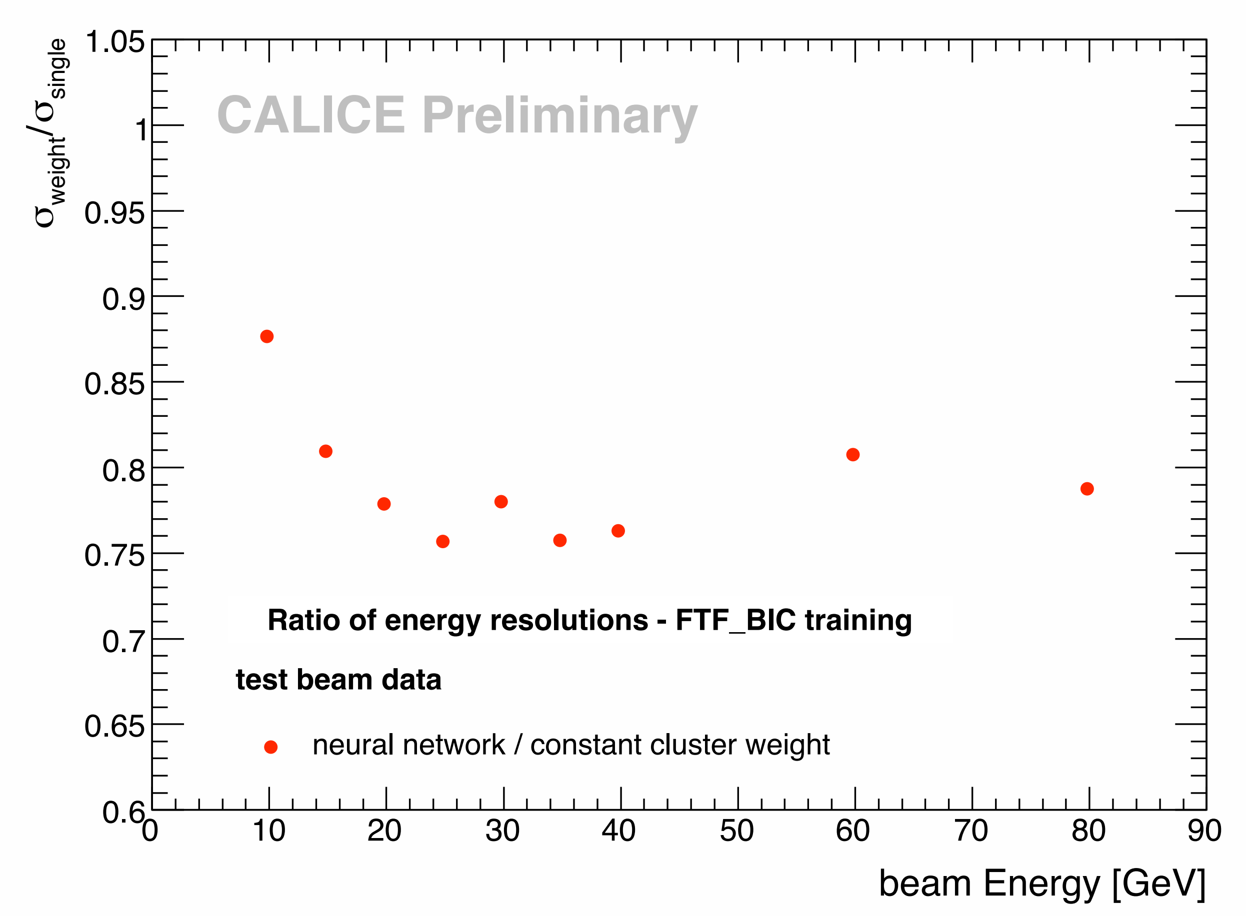


- Improvement by up to 25%, poorer performance at low and high energies
  - Introduces constant term in the fit

Neural Network: Performance - Resolution

- Training with FTF_BIC





- Improvement by up to 25%, poorer performance at low energies, constant for high energies
  - Introduces (a smaller) constant term in the fit

Summary and Next Steps

- Software compensation in imaging calorimeters now well established
  Two approaches investigated so far:
  - Cell-by-cell weighting
  - Cluster-based weighting

- The new results: Cluster-based weighting with simple weight and neural network
  - Neural network yields very good results, slightly better than the cell-by-cell approach

- Next step: Integrate cluster-based weighting into PandoraPFA
- Analysis note CAN-021 for presentation at LCWS with editorial board

Summary and Next Steps

- Software compensation in imaging calorimeters now well established
  Two approaches investigated so far:
  - Cell-by-cell weighting
  - Cluster-based weighting

- The new results: Cluster-based weighting with simple weight and neural network
  - Neural network yields very good results, slightly better than the cell-by-cell approach

- Next step: Integrate cluster-based weighting into PandoraPFA
- Analysis note CAN-021 for presentation at LCWS with editorial board

Very detailed and rather coarse energy density measurements yield comparable results

➩ The optimum might be somewhere in the middle: Look at sub-clusters