Preprint typeset in JINST style - HYPER VERSION

CALICE Analysis Note CAN-021
April 12, 2010

Software Compensation for Hadronic Showers in the CALICE AHCAL and Tail Catcher with Cluster-based Methods

The CALICE Collaboration*

This note contains preliminary CALICE results, and is for the use of members of the CALICE Collaboration and others to whom permission has been given.

ABSTRACT: The hadronic energy resolution of the CALICE setup, consisting of a silicon tungsten electromagnetic calorimeter, an analog hadron calorimeter and an analog tail catcher has been studied using data taken in 2007 at CERN and simulations. To improve the energy resolution of hadronic showers in the hadron calorimeter and the tail catcher, a weighting procedure based on the energy density of the hadron shower is studied. The shower itself and its energy density was reconstructed using a simple clustering algorithm. Furthermore, the use of a neural network has been studied for the same purpose. Both methods use simulated data to determine weights which are then applied to test beam data. These first preliminary studies yield an relative improvement of the energy resolution by roughly 15 % for the shower weighting technique and 23 % for the neural network approach compared to the energy resolution of hadronic showers without software compensation applied.

*Corresponding authors: Katja Seidel (kseidel@mpp.mpg.de), Frank Simon (frank.simon@universe-cluster.de)

Contents

1. Introduction

The CALICE collaboration has constructed highly granular calorimeter prototypes for future collider experiments. The CALICE calorimeters were tested in various different configurations in particle beams at DESY, CERN and FNAL. For the data set studied in this note, taken in 2007 at CERN, a silicon-tungsten electromagnetic calorimeter (ECAL) [1], an analog scintillator-steel hadron calorimeter (AHCAL) [2] and a scintillator-steel tail catcher and muon tracker (TCMT) [3] were installed.

The ECAL has a total depth of $24\,X_0$ and consists of 30 active layers arranged in three longitudinal sections with different samplings. The first ten layers use 1.4 mm thick tungsten absorber plates ($0.4\,X_0$), followed by ten layers of 2.8 mm thick absorbers ($0.8\,X_0$) and 10 layers of 4.2 mm thickness ($1.2\,X_0$). The total thickness of the calorimeter is 20 cm. Each silicon layer has an active area of $18\times18\,\mathrm{cm}^2$, segmented into individual modules with 6×6 readout pads with a size of $1\times1\,\mathrm{cm}^2$. This results in a total of 9720 channels for the detector.

The AHCAL consists of small scintillator tiles with individual readout by silicon photomultipliers (SiPMs) [4] arranged in layers between 1.75 cm thick stainless steel absorber plates. The full layer thickness is approximatley 3 cm [5]. The size of the scintillator tiles ranges from $3\times3\,\mathrm{cm}^2$ in the center of the detector to $12\times12\,\mathrm{cm}^2$ on the outer edges of the calorimeter. In the last eight layers only tiles with $6\times6\,\mathrm{cm}^2$ and $12\times12\,\mathrm{cm}^2$ are used. In total, the hadron calorimeter has 38 sensitive layers, amounting to a depth of 4.5 interaction lengths λ_I. The total number of scintillator cells is 7608.

The TCMT consists of 16 readout layers each with twenty $100\times5\,\mathrm{cm}^2$ scintillator strips read out by SiPMs between steel absorber plates, resulting in 320 readout channels. The detector is subdivided into a fine and a coarse section, where the first 8 layers have 19 mm thick absorber plates, while the absorbers for the last 8 layers are 102 mm thick. The orientation of the scintillator strips alternates between horizontal and vertical in adjacent layers. In total, the TCMT thickness corresponds to a depth of $5.8\,\lambda_I$. This gives a total depth of approximately $11.3\,\lambda_I$ for the complete CALICE setup.

The present note describes a preliminary analysis of hadron data taken at CERN in 2007 and for the chosen test beam runs simulated data produced with the physics lists FTF_BIC and QGSP_BERT using GEANT4.9.3. The reconstructed energy of a hadronic shower in the AHCAL and the tail catcher has been studied in some detail. The potential to improve the energy resolution with two methods is described. The second method is a shower weighting procedure based on the energy density of the shower. The first method uses a neural network using cluster properties of the hadronic shower to reconstruct the energy of the shower.

2. Motivation and Overview

As described in greater detail in [7], a hadronic shower consists of a visible hadronic component, an electromagnetic component, and invisible energy deposited in the form of in the form of binding energy, nuclear recoil, neutrinos, and (mostly unseen) energy in the form of low energy neutrons. The electromagnetic component results from neutral pions created in the hadronic cascade, and is most prominent in the core of the shower. The observed signal for a particle showering in a non compensating calorimeter, like the CALICE calorimeter, is larger in the case of electromagnetic than of hadronic showers for a given energy, commonly expressed as the ratio e/h > 1. The average electromagnetic fraction of a hadron shower increases with the energy of the incident particle. Large fluctuations from event to event in the relative fractions of electromagnetic and hadronic subshowers together with a non-unity e/h ratio lead to a deterioration of the energy resolution for hadrons. Hadronic showers with a high energy density (energy/volume of the shower) tend to have a higher reconstructed energy than those with low densities, as shown in Figure 1. This is

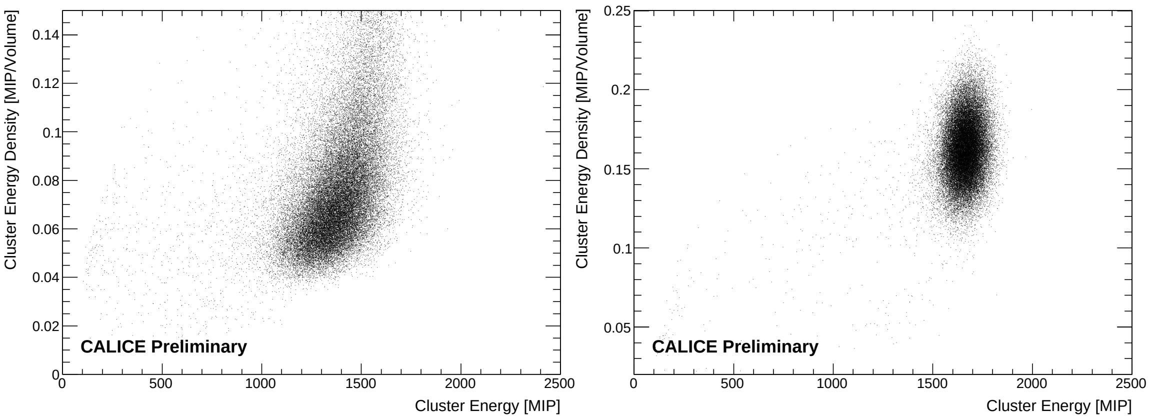

Figure 1. Relation between the density of a cluster and the reconstructed energy in MIP for a 40 GeV test beam pion data run on the left. For larger densities the reconstructed energy is generally larger. This is not the case for a 40 GeV positron run (right picture). Note the different scales of the y-axis. Pure electromagnetic showers have higher energy densities than hadronic ones.

exploited by software compensation techniques to improve the reconstructed energy and the energy resolution. In this note, we discuss two software compensation techniques developed on the basis of clustered hadronic showers. Chapter 5 describes a simple weighting technique based on the energy density of reconstructed clusters. This analysis is closely related to the analysis described in [7]. Not only do hadronic showers have a strong correlation between the reconstructed energy and the energy density of a showers due to the different electromagnetic and hadronic components, but also the interplay of other cluster variables such as cluster length, cluster width or the position of the cluster in the calorimeter may have a significant influence on the reconstructed energy. In Chapter 6 a neural network technique is used to exploit these relations and perform a software compensation on this basis.

A clustering algorithm, described in Chapter 3, is the first step of both analyses. This clustering algorithm finds clusters which start in the AHCAL and may leak into the tail catcher. If a cluster is found, basic cluster properties such as the cluster energy, the cluster length and width are determined. It should be noted that the focus of this note is on the algorithms to improve the energy resolution and not the clustering itself. Every (more developed) clustering algorithm, e.g. the clustering in PandoraPFA [8], could in principle be used with the software compensation techniques discussed here.

Both analyses types use Monte Carlo information to reconstruct the energy of pions showering in the calorimeter of the CALICE setup. In total three data sets are compared to each other. The

first data set is test beam data of π^- events between 10-80 GeV. The run numbers are listed in Appendix A. The other two data sets are Monte Carlo data, both simulated for the mentioned test beam runs. The FTF_BIC and QGSP_BERT physics lists were chosen because both show quite good agreement with test beam data for the energy resolution and the reconstructed energy (Section 3.1). The software versions used for the test beam run reconstruction and Monte Carlo simulations are listed in Appendix B.
In the Chapter 4 a method to correct the Monte Carlo energy is described. This is necessary because the energy is an important parameter to extract the weights from Monte Carlo data.

3. Clustering Algorithm

The clustering is only performed in the AHCAL. Only events with a valid beam trigger (for test beam data) and hits with an energy deposition above a threshold of 0.5 MIP are analyzed. For the test beam data the Cherenkov trigger was used as well reduce the electron and proton background. For the Monte Carlo data only π^- events were simulated.

Seed finding: The geometry of the tiles in the calorimeter are defined in the I, J, K system. A seed is found in the I, J plane by the projection of the hits on the AHCAL front face. This is done once weighted with the energy of the hit and once unweighted to get just the number of hits for the different I, J values. In both projections the local maxima are found. A local maximum is chosen as a seed if it has more than four hits in the hit projection and more than 5 MIP in the energy projection.

Finding the full shower: Starting with the seed and $K = 1$ (first layer of the AHCAL) the algorithm steps through the AHCAL layers and collects all hits with same I and J values as the seed hit as cluster hits, as well as all neighbors with a hit energy deposition above the threshold of 0.5 MIP. A gap is defined as a distance of more than three successive cells without an energy deposition after a cluster hit. If a gap is found, the algorithm stops to search for hits deeper (higher K values) in the AHCAL. Because the algorithm always starts searching for cluster hits in the first layer of the AHCAL, the track which leads to a cluster is found and will be counted as part of the cluster. If the last layer of the calorimeter is reached or a gap is found, the algorithm goes back to the front of the calorimeter and continues to collect all neighbors of cluster hits which are not already in the cluster. Every hit is collected if there is not a gap between the hit looked at and the closest cluster hit. This stage continues until no more cluster hits are found. A picture of an event with a cluster can be seen in Figure 2. The ECAL and the tail catcher are not shown in this figure.

3.1 Energy reconstruction of clusters in the AHCAL and TCMT

Event selection

For the following analyses only events are analyzed that have one cluster in the AHCAL, with has more than 70 hits. This cut on the number of hits in the AHCAL is used to reject muon events and events with a most likely not completely found shower. To select only showers which start in the AHCAL a second cut, requiring less than 50 hits and 70 MIP in the ECAL, is introduced.
The hits which are not found to be part of the cluster are most likely either noise hits or result of an energy deposition of a neutron which was created as part of the cluster. The mean not counted

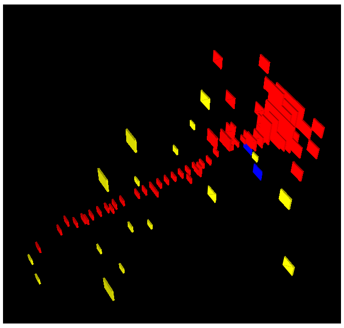

Figure 2. Example event of a 20 GeV pion run. The red squares are hits which are part of the cluster. The yellow squares are isolated hits which are not part of the cluster and the blue squares are non-isolated hits which are not part of the cluster.

energy is around 10 MIP for all beam energies.

Leakage to the tail catcher

If a cluster in the AHCAL has hits in the last layer of the AHCAL it is likely that the shower is not contained in the AHCAL but also deposited energy in the tail catcher. To define the cluster properties, the shower is extended to the tail catcher by looking for hits in successive layers starting at the first tail catcher layer behind the AHCAL. A hit (energy deposition larger than 0.5 MIP) belongs to the cluster if the full layer energy is above the threshold of 0.8 MIP. A hit with an energy deposition below this threshold will end the cluster in the tail catcher.

Energy resolution with clustering algorithm

The distributions of the cluster energy (AHCAL + TCMT) is calculated for the test beam data and the two Monte Carlo data sets. The reconstructed energy and resolution is extracted from a two-step gaussian fit of histograms of the event-by-event distribution of the reconstructed energy. First, a Gaussian was fitted over the full range of the histogram. Then, a second Gaussian was fitted only in the range of $\pm 1.5\,\sigma$ of the first fit. The mean and the σ of this second fit were used as the mean reconstructed energy and as the energy resolution, respectively. For the conversion from the MIP to the GeV scale a single energy independent factor of 0.03 GeV/MIP was used for every cluster. This factor was determined from a 15 GeV pion run and can be interpreted as an electromagnetic conversion factor, multiplied with the e/pi ratio at 15 GeV. The energy resolution, shown in Figure 3, for test beam data was found to be:

$$\textbf{Test Beam Data: } \frac{\sigma}{E} = \frac{64.3 \pm 0.4\,\%}{\sqrt{E[\mathrm{GeV}]}} \oplus 0.0 \pm 0.7\,\% \oplus \frac{0.2 \pm 0.4}{E[\mathrm{GeV}]}$$

The resolutions obtained from Monte Carlo were:

$$\textbf{FTF\_BIC: } \frac{\sigma}{E} = \frac{61.6 \pm 0.4\,\%}{\sqrt{E[\mathrm{GeV}]}} \oplus 2.7 \pm 0.3\,\% \oplus \frac{0.0 \pm 0.2}{E[\mathrm{GeV}]}$$

$$\textbf{QGSP\_BERT: } \frac{\sigma}{E} = \frac{56.7 \pm 1.2\,\%}{\sqrt{E[\mathrm{GeV}]}} \oplus 2.0 \pm 0.5\,\% \oplus \frac{0.9 \pm 0.1}{E[\mathrm{GeV}]}$$

for Monte Carlo data. The errors in the energy resolution and the linearity are statistical only and mostly smaller than the marker size.

The energy resolutions obtained from both sets of Monte Carlo data are both similar to that obtained of test beam data. The reconstructed energy for FTF_BIC data differs at most 3.5 % to the test beam data. The disagreement is larger for QGSP_BERT data with a maximum deviation of 5.5 %. The results of test beam and Monte Carlo data without the clustering are shown in the Appendix in Figure 21.

The fit of the energy resolution was done without start values or limits for the three parameters. A fit option to perform better errors estimation using Minos technique was applied. The stochastic, the constant and the noise term in the fit are strongly correlated, which does not allow comparison of single parameters from different fits. In the following the energy resolutions are fitted to guide the eye and the fit results are shown in the Figures. The argumentation of energy resolution improvement is done by showing the ratio of energy resolution with and without software compensation technique applied.

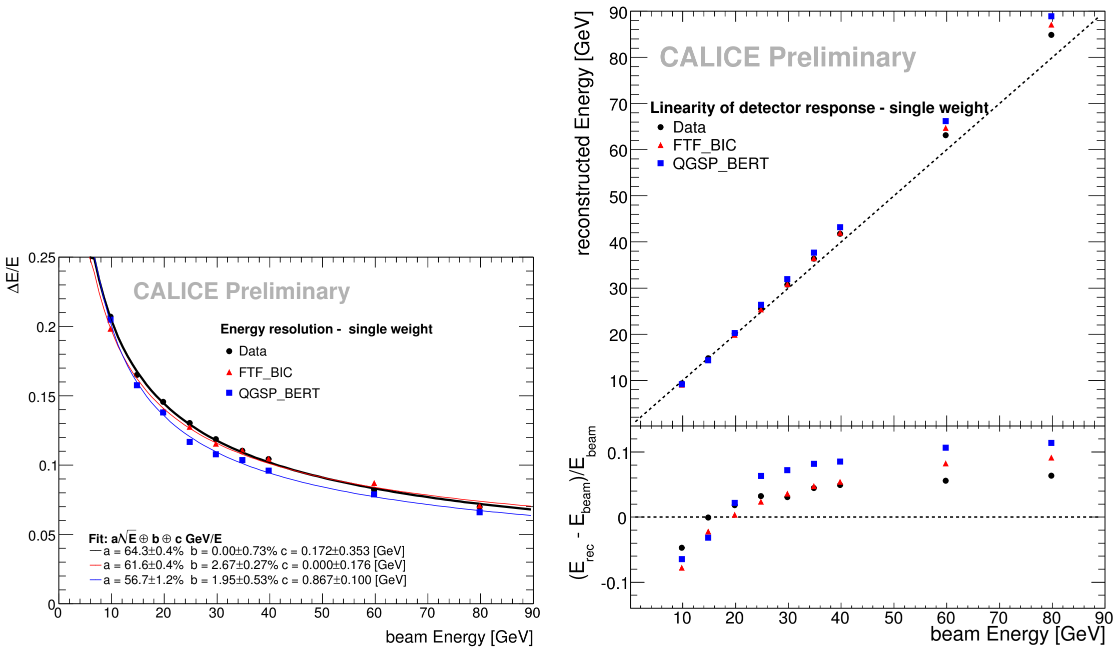

Figure 3. Energy resolution (left picture) and the linearity (right picture) for test beam (black points) and Monte Carlo (blue points) data.

The energy resolution of test beam data with and without the clustering can be compared via the ratio of the energy resolutions as a function of energy, shown in Figure 4. Since the clustering algorithm does not associate isolated hits or small isolated subclusters, stemming for example from neutrons, it is likely to miss a fraction of the total energy. This leads to a deterioration of the energy resolution compared to an analysis of the full visible energy in the calorimeter. This effect decreases with increasing shower energy.

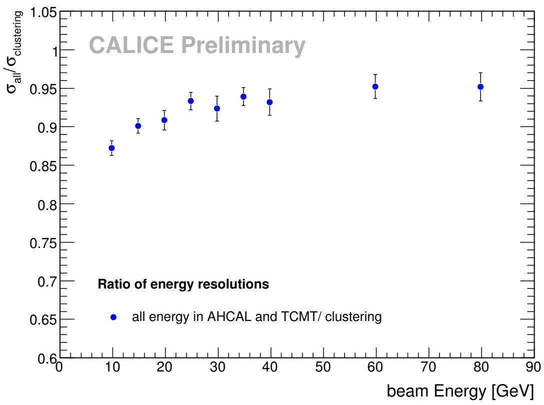

Figure 4. Ratio of the energy resolution with and without the clustering. σ_{all} is the width of the gaussian of the reconstructed energy taken all hist in the tail catcher and AHCAL into account. $\sigma_{clustering}$ is the width of the gaussian with the clustering algorithm.

3.2 Shower Properties

The analyses described in this note are based on the input of one or more cluster properties. In the following, the definitions of the chosen variables are given and comparisons of data and simulation are shown. The events with one cluster are analyzed to find shower properties which describe the hadronic showers at energies from 10 GeV to 80 GeV. For this purpose, the variables which were chosen display a strong beam energy dependence. For a 40 GeV run the described variables are shown for test beam and Monte Carlo data in Figure 5.

Shower energy: The energy sum of all hits in the AHCAL which belong to the shower and of the layers in the tail catcher which belong to the shower defines the total energy of shower.

Shower length: The total shower length is defined as the length in layers between the shower starting point and the hit with the highest K-value (in AHCAL or tail catcher) of the shower. The shower starting point is defined as the layer in which the number of hits in this layer and the two former layers is higher than 3 hits and the energy sum of these layer is higher than 8 MIP.

Shower width: It is assumed that the shower axis is always perpendicular to the front plane of the AHCAL and is defined by the shower seed. The distance of every hit to the shower axis is calculated. The mean value of these distances is defined as the cluster width. This value is only calculated for cluster hits in the AHCAL, not in the tail catcher.

Shower volume: If the shower is contained in the AHCAL the volume is defined by the sum of all single tile volumes (tile volume here: area of tile times thickness of layer). If the shower leaked into the tail catcher, the cluster volume in the tail catcher is calculated with the following method. For every layer a cylinder around the cluster axis is calculated with a length of 5 cm (layer thickness). For more than one hit in a layer the diameter of the cylinder is taken as the mean distance of the hits. If there is only one hit per layer the diameter is assumed to be the width of the scintillator bar 5 cm. The sum of the volumes of these cylinders summed up give the volume of the cluster in the tail catcher.

Tail catcher cluster energy: The energy of the cluster hits in the tail catcher form the tail catcher cluster energy. The value can be interpreted as longitudinal energy deposition information.

Energy in the last five AHCAL layers: The cluster energy which is deposited in the last five AHCAL layers.

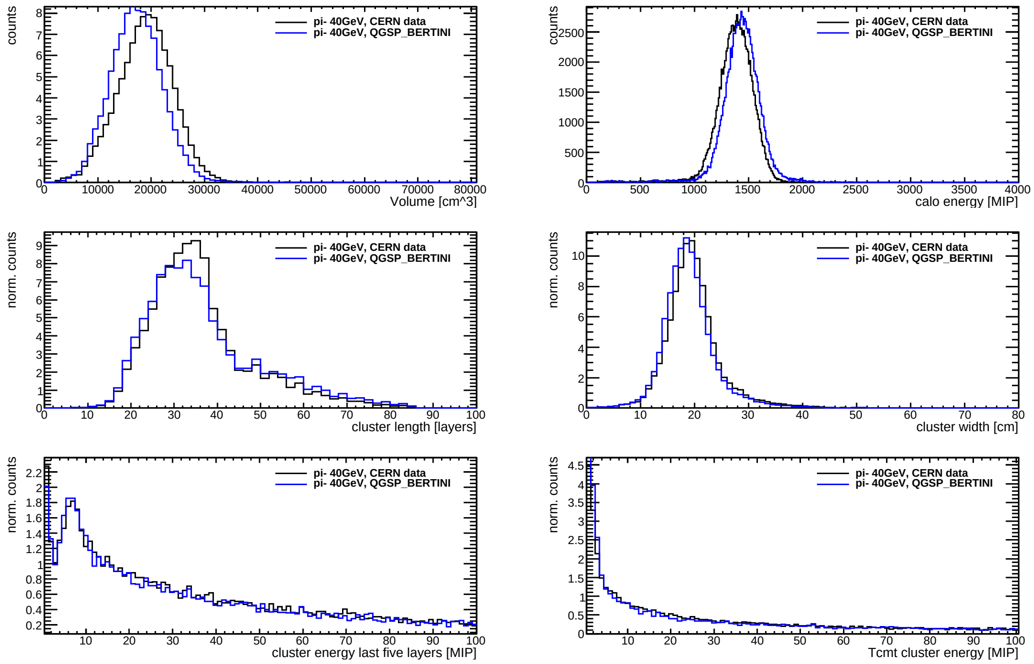

Figure 5. Shower property variables of test beam (black) and Monte Carlo (blue) data of a 40 GeV run.

4. Monte Carlo energy correction

The determination of the weights for both analyses depend strongly on the reconstructed shower energy E_{rec}. For the neural network technique, the reconstructed energy is used as one input variable. For the cluster energy density weighting technique, the parameterization of the weights depends on the reconstructed energy. We found that weights extracted from Monte Carlo data can only successfully be applied to test beam data if the reconstructed energy, with which the weights will be determined, does not differ too much from the reconstructed energy on which these weights are applied (see Figure 20). The difference between reconstructed energy and beam energy for test beam data and simulation is shown in Figure 6. This difference also reflects the differences between reconstructed energy in simulations and test beam data; which is large for low and high beam energies in both physics list compared to test beam data. Such a correction would not be necessary for a physics list which reproduces the behavior of the energy reconstruction in data to good precision. The reconstructed energy of the Monte Carlo data is corrected by a multiplicative

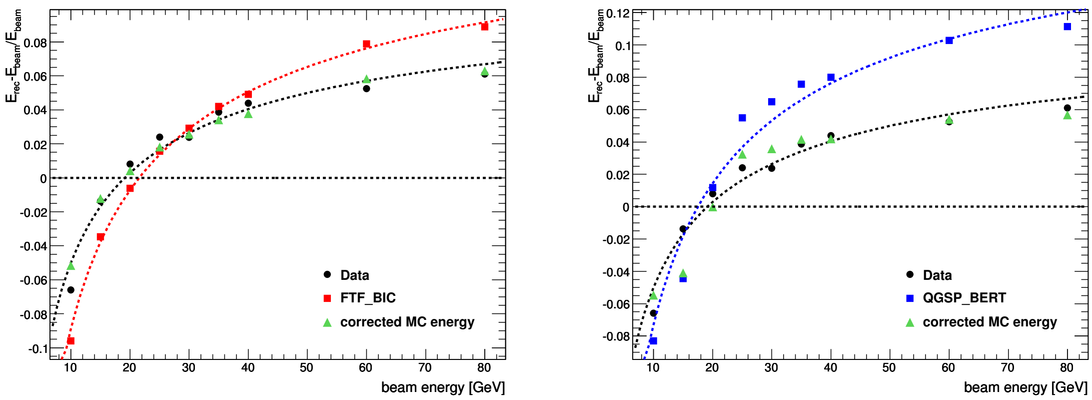

Figure 6.
Left: FTF_BIC: Fit of the energy dependence of the difference between reconstructed and beam energy for test beam data (black points) and Monte Carlo data (red squares) and to the test beam data corrected Monte Carlo data (green triangles).
Right: QGSP_BERT: Fit of the energy dependence of the difference between reconstructed and beam energy for test beam data (black points) and Monte Carlo data (blue squares) and to the test beam data corrected Monte Carlo data (green triangles).

factor $k_{\mathrm{E_{rec}}}$.

$$E_{\mathrm{rec,c}} = E_{\mathrm{rec}} \cdot k_{\mathrm{E_{rec}}} \tag{4.1}$$

The correction factor $k_{\mathrm{E_{rec}}}$ is extracted from the energy difference of the reconstructed energies to the beam energy shown in Figure 6. In these figure the energy dependence of the normalized difference is fitted with a function $f(E_{\mathrm{beam}}) = \frac{a}{\sqrt{E_{\mathrm{beam}}}} + b$. The black points and line belong to the test beam data, the red points and line belong to the FTF_BIC Monte Carlo data in the left plot of Figure 6 and the blue points and line in the right plot of Figure 6 belong to the QGSP_BERT Monte Carlo data. The correction factor is then extracted from the relations $E_{\mathrm{rec,data}} = k_{\mathrm{E_{rec}}} \cdot E_{\mathrm{rec,MC}}$ and

$f(E_{\text{beam}}) = \frac{E_{\text{rec}} - E_{\text{beam}}}{E_{\text{beam}}}$. This leads to a correction factor of

$$k_{\mathrm{E_{rec}}} = \frac{\frac{a_{\text{data}}}{\sqrt{E_{\text{beam}}}} + b_{\text{data}} + 1}{\frac{a_{\text{MC}}}{\sqrt{E_{\text{beam}}}} + b_{\text{MC}} + 1}. \tag{4.2}$$

Because the beam energy should not be used in the weight determination it is replaced by the reconstructed shower energy:

$$k_{\mathrm{E_{rec}}} = \frac{\frac{a_{\text{data}}}{\sqrt{E_{\text{rec,MC}}}} + b_{\text{data}} + 1}{\frac{a_{\text{MC}}}{\sqrt{E_{\text{rec,MC}}}} + b_{\text{MC}} + 1}. \tag{4.3}$$

With this correction factor, the energy of the clusters in Monte Carlo is increased for low energies and decreased for the high energies. The difference between corrected and beam energy for the corrected Monte Carlo data can be seen by the green triangles in figure 6 on the left for FTF_BIC and on the right by the green triangles for QGSP_BERT simulated data. These green triangles are much closer to the black line. In both following analyses, this corrected energy is used. All other cluster properties such as cluster energy density, cluster length are not changed.

5. Cluster Energy Density Weighting Technique

The cluster weighting technique is similar to the single cell weighting technique described in CAN-015, but this time only one weight per shower per event is used. The weight depends on the energy density of the shower which is defined as the cluster energy divided by cluster volume. The definitions of the cluster energy and volume are described in Chapter 3.2.

As shown in Figure 1, clusters with a high energy density tend to have a higher reconstructed cluster energy for the same particle energy. Since electromagnetic subshowers tend to be denser than purely hadronic ones, the higher the electromagnetic content in this shower the larger is the energy density and therefore the reconstructed energy. The cluster energy density is chosen as the property to determine the amount of the electromagnetic content. The strength of this correlation between the reconstructed energy and the cluster density depends on the beam energy. Therefore, a weighting technique based on the cluster energy density can be applied, if the weights are energy dependent.

5.1 Energy reconstruction and parameterization of cluster energy density weights

The simplest way to calculate the reconstructed cluster energy is to use one factor w to get from the MIP scale to the GeV scale.

$$E_{\text{rec}}[\text{GeV}] = \sum_{hit} E_{hit}[\text{MIP}] \cdot w = E_{\text{rec}}[\text{MIP}] \cdot w$$

This factor w is constant for every energy and event and was determined to be 0.03 GeV/MIP.

To improve the energy resolution, not one weight factor w for every event and energy is used, but weight factors $\omega(\rho, E) = (\omega_1(E), ..., \omega_8(E))$ which depends on the energy density ρ of the cluster. Therefore, the cluster energy density is divided into eight bins, which are shown in Figure 7.

The weighted cluster energy in GeV is calculated with

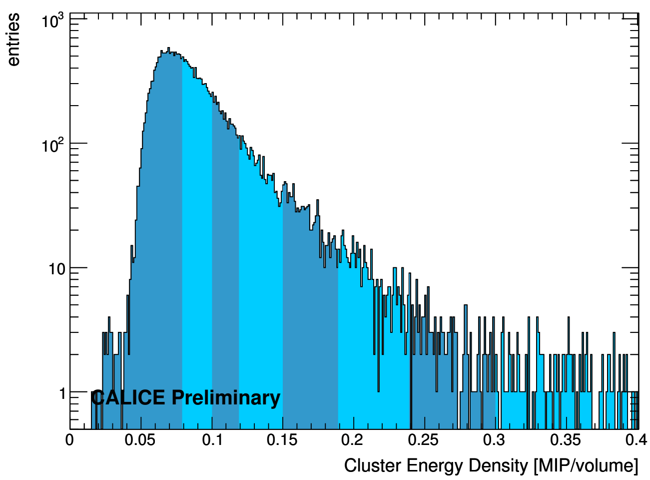

Figure 7. Cluster energy density for the run 331280 (80GeV). The subdivision of the energy density into eight different bins is illustrated by the color shading.

$$E_{\text{rec,weighted}}[\text{GeV}] = \sum_{hit} E_{hit}[\text{MIP}] \cdot \omega(\rho, E) = E_{\text{rec}}[\text{MIP}] \cdot \omega(\rho, E). \tag{5.1}$$

The weight for each cluster $\omega(\rho,E)$ depends on the cluster energy density. Suitable weights $\omega(\rho,E)$ are found by the minimization of the Function 5.2 for each run individually.

$$\chi^2 = E_{\mathrm{rec}} \cdot \omega(\rho,E) - E_{\mathrm{beam}} \tag{5.2}$$

In this determination, the energy loss of the incoming particle in the ECAL was taken into account by reducing the beam energy E_{beam} by 200 MeV. This corresponds to the mean energy loss of a minimum ionizing particle in this detector, calculated from the material properties.
At this stage the Function 5.2 was minimized for every run of the Monte Carlo data sets individually and weights were extracted. These individual weights for each run differ for the different cluster energy density bins. The weights are parameterized, see Figure 8, by a function with two parameters, given by

$$\omega(\rho,E) = (a(E) + b(E) \cdot \rho). \tag{5.3}$$

In this function, the parameter a,b are energy dependent functions itself and x is the center of the corresponding energy density bin.

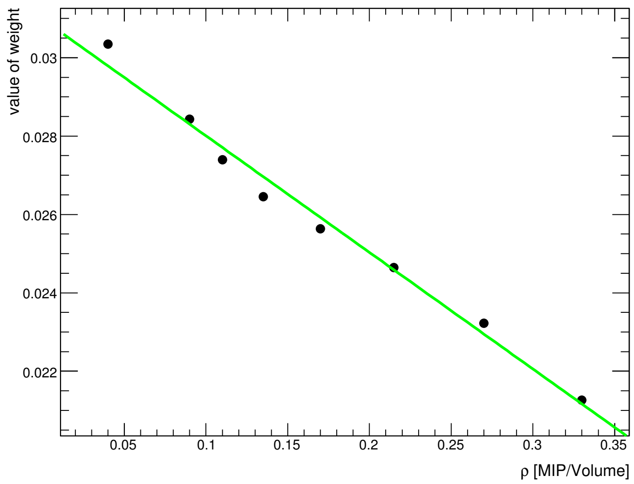

Figure 8. Individual weights for a 40 GeV Monte Carlo run. The fit describes the parametrization of these weights.

Using Function 5.3, each individual weight set of every beam energy was fitted and the energy dependence of the parameters a and b for all beam energies is shown Figure 9, where the parameters a and b are plotted versus the beam energy. The distributions were fitted with the following functions:

$$a(E) = p_1 \cdot (1 - \exp(p_2 \cdot E)) + p_3 \tag{5.4}$$

and

$$b(E) = q_1 + q_2 \cdot \exp(q_3 \cdot E). \tag{5.5}$$

For this analysis, the two weight parameters a and b are chosen according to these phenomenological descriptions, taking the reconstructed energy of the single weight method as input energy. Equation 5.1 giving the weighted reconstructed energy, can then be rewritten to

$$E_{\mathrm{rec,weighted}} = E_{\mathrm{rec}} \cdot ([p_1 \cdot (1 - \exp(p_2 \cdot E_{\mathrm{rec}})) + p_3] + [q_1 + q_2 \cdot \exp(q_3 \cdot E_{\mathrm{rec}})] \cdot x) \tag{5.6}$$

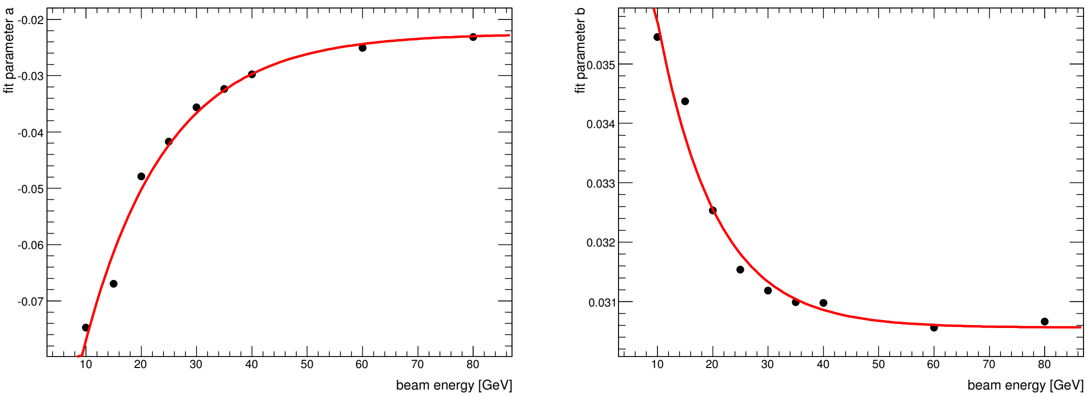

Figure 9. Energy dependence of the parameters in the weighting function $\omega(E) = (a(E) + b(E) \cdot x)$. left picture: Energy dependence of function parameter a. Right picture: Energy dependence of function parameter b.

where x is the center of the corresponding energy density bin. With this method, the reconstructed energy and the corresponding energy resolution is calculated for each run in the data set.

5.2 Results

This result section has two parts. The first Subsection 5.2.1 shows results on the energy resolution, the gain in energy resolution and the linearity of test beam data obtained with weights extracted from the physics list FTF_BIC. The second part 5.2.2 shows the results with weights obtained from the physics list QGSP_BERT.

5.2.1 Data with FTF_BIC weights

The energy dependent weight Function 5.3, was determined with the simulated data sample of FTF_BIC. The results for the energy resolution are shown in Figure 10. The black points show test beam data with one constant weight factor and the red circles the test beam data with the weights obtained from FTF_BIC data. The distributions of the reconstructed energies were fitted with the described two step Gaussian fit.

The gain in energy resolution is shown by the ratio of the energy resolution with and without weighting on the right plot on Figure 10. The energy resolution of the cluster weighting technique is labeled σ_{weight} and the energy resolution of the test beam data with one constant factor applied, is labeled σ_{single}. The ratio of these two values is between 0.83 and 0.9 over the full energy range. Therefore an improvement of 13 % in the energy resolution on test beam data with weights extracted from FTF_BIC data could be reached.

The reconstructed shower energy with the cluster weighting technique of the FTF_BIC physics list fulfills linearity better than 3 %, shown in Figure 11.

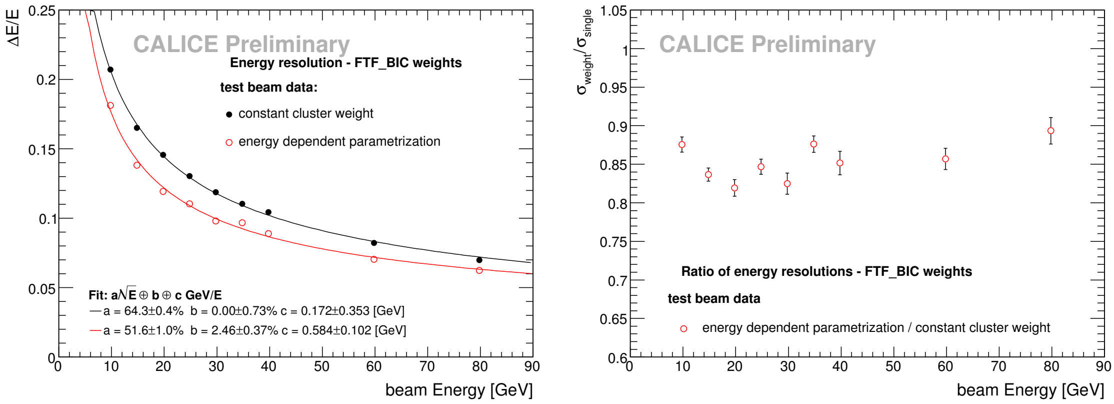

Figure 10. Left: Energy resolution of test beam data with a single weight factor (black points) and with the weights applied (red circles) which were extracted from the cluster energy density weighting approach which was determined with FTF_BIC Monte Carlo data.
Right: Ratio of energy resolution.

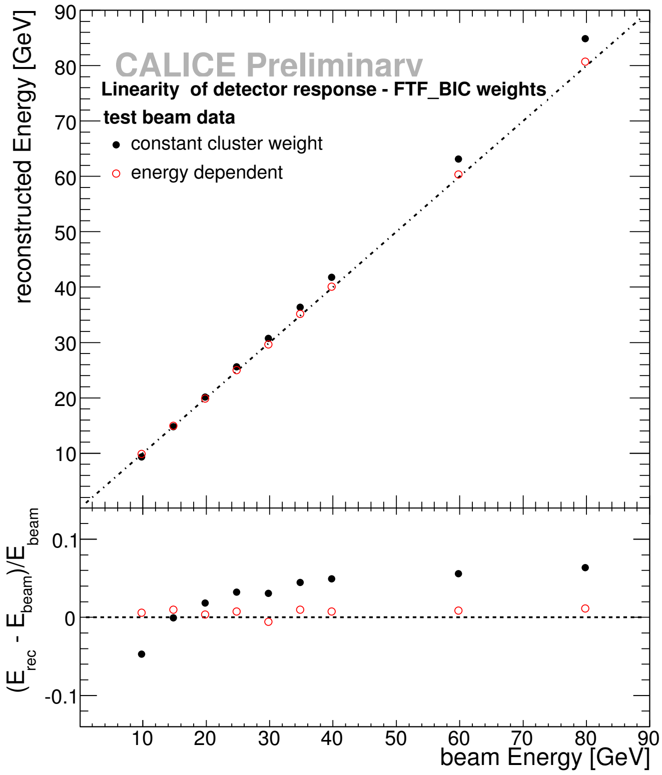

Figure 11. Linearity of test beam data with a single weight factor (black points) and with the weights applied (red circles) which were extracted from the cluster energy density weighting approach which was determined with FTF_BIC Monte Carlo data.

5.2.2 Data with QGSP_BERT weights

The same analysis described in Section 5.1 was performed with simulated data of the physics list

QGSP_BERT.

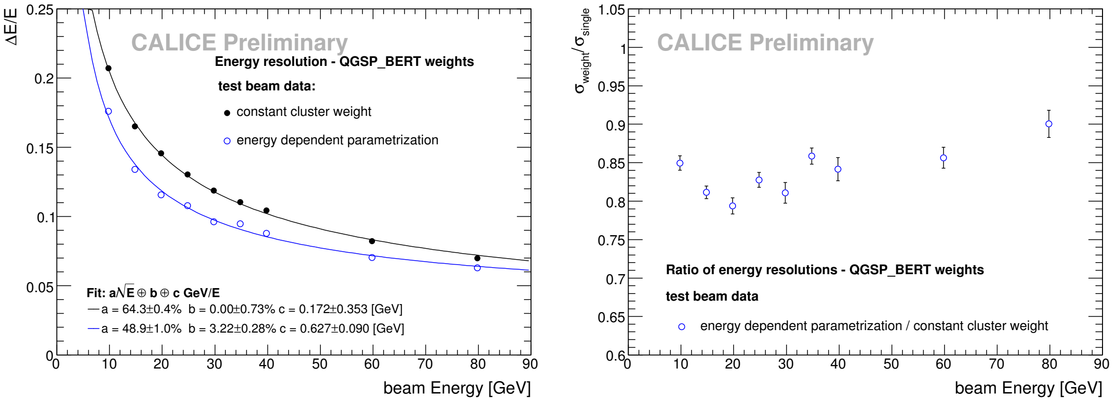

Figure 12. Left: Energy resolution of test beam data with a single weight factor (black points) and with the weights applied (blue circles) which were extracted from the cluster energy density weighting approach which was determined with QGSP_BERT Monte Carlo data.
Right: Ratio of energy resolution.

The improvement in energy resolution for each beam energy individually is shown on the right side of Figure 12. For low beam energies the gain in energy resolution is between 0.79 and 0.9 and therefore better than for the results obtained with the physics list FTF_BIC. For higher energies the ratio is similar to the ratio obtained with FTF_BIC weights. Overall, the energy resolution was improved by 15 %.

Figure 13 shows the linearity of the original reconstructed energy and the reconstructed energy with the weighting technique. The difference between reconstructed energy with the weighting technique and beam energy is better that 4 %, with a constant offset of approx. 2 %.

5.3 Conclusion

The cluster energy weighting technique gives similar results on test beam data for weights extracted from FTF_BIC and QGSP_BERT. The energy resolution improved by 13 % for FTF_BIC weights and 15 % for QGSP_BERT weights. The difference is mainly because the gain in energy resolution is slightly better for QGSP_BERT weights in the energy range from 10 to 40 GeV. For higher energy of both physics lists perform similar.

The linearity for the weights extracted from FTF_BIC data is better than 3 %, using the QGSP_BERT weights results in a similar linearity, but a constant offset in the reconstructed test beam energy of about 2 % in introduced.

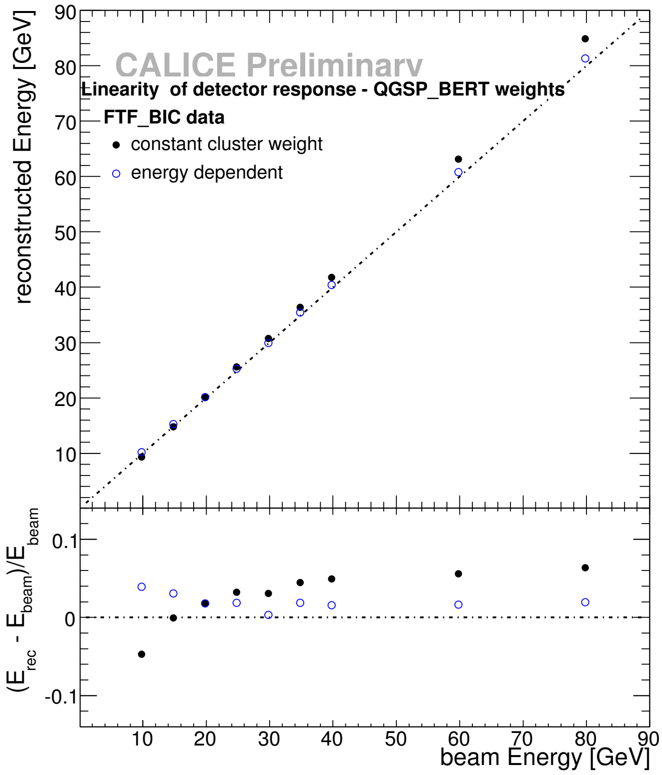

Figure 13. Linearity of test beam data with a single weight factor (black points) and with the weights applied (blue circles) which were extracted from the cluster energy density weighting approach which was determined with QGSP_BERT Monte Carlo data.

6. Neural Network Technique using TMVA

6.1 Overview

The use of a neural network for the reconstruction of the energy of hadronic showers from shower properties was studied. The program TMVA (Toolkit for Multivariate Data Analysis with ROOT) was used to perform a regression analysis. A neural network was built, trained and tested. In the training phase, the neural network was fed with cluster variables and one target value (the beam energy). The neural network was trained with a special set of simulated data, which had to be produced for this purpose. The trained neural network provides an energy estimation for each set of input variables. The goodness of the chosen neural network architecture can the tested by applying the trained neural network on other data sets. In the testing phase, the target variable was not used.

A regression method with neural network consists of a certain number of layers and each layer of a certain number of neurons or nodes. The first layer is always the input layer in which the input variables are defined as the input nodes. The last layer is the output layer with the output node of the neural network, in this case the reconstructed energy in GeV. The layers in between are called hidden layers. The number of hidden layers and the nodes in each hidden layer have to be defined by the user of the neural network. Working with the neural network means, in this case, to find suitable input variables and a reasonable number of nodes in the hidden layers and the target variable.

6.2 Training and Testing of the neural network

In the training phase an input data set was used, which covered the whole energy range between 5 GeV to 105 GeV of beam energies to be used as target values for the neural net.
The neural network could not be trained with test beam data, because the test beam data was only available in steps of 5 GeV. Taking test beam data in the training phase resulted in unrealistic good reconstructed energies, for the used beam energies, since the network picked up the quantization of the beam energy in the training phase. Consequently, events with different energies would not reconstructed correctly. Therefore, a training data sample was simulated with continuous small steps of beam energy. For training, 200000 π^- events were simulated in energy steps of 0.1 GeV with the physics list QGSP_BERT and FTF_BIC in the energy range between 5 and 105 GeV. Several neural network architectures were trained with the input variables shown in Figure 5.
A number of network architectures were tried to find the one with the best performance. The chosen architecture of the neural net consists of one hidden layer with N+5 neurons, where N is the number of input variables. For both physics list the same architecture was chosen to have comparable results.
In the training phase of the neural network, it was necessary to introduce a so called "weight expression". This "weight expression" is not a physics driven weighting but it gives some events more weight. The neural network is trained to minimize the absolute deviations of the reconstructed values from the target values. Large target values are thus overemphasized in the context of a hadronic calorimeter. With the used weight expression lower energies (target values) are taken into account more in the training phase. The function which was used for this weight expression is $f(\text{target}) = 500/(\text{target} - 5)$. The effect of this "weight expression" is shown in Figure 20 in the Appendix.

Testing the neural network means applying the neural network on a data set which has not been used for the training of the neural network. Two neural network were trained with the physics lists FTF_BIC and QGSP_BERT. The neural networks were applied on the test beam data sample and on the Monte Carlo data sample of the physics list which was not used in the training phase.

6.3 Results

The section is divided in two parts. In the subsection 6.3.1 the results obtained with the network trained on data simulated with FTF_BIC applied to test beam data are discussed. The linearity of the reconstructed energy, the energy resolution and the gain in energy resolution for the test beam data are studied. In the subsection 6.3.2 the results obtained with a neural network trained with QGSP_BERT simulated data and applied on test beam data are presented.

6.3.1 Energy reconstruction with a Neural Net trained with simulations with FTF_BIC

Figure 14 and 15 show the results of the energy reconstruction of test beam data with a neural network trained with FTF_BIC. The black points show the single weight cluster result, described in Section 3, and are shown as a reference. The red circles show the result of the energy reconstruction with the neural network. The distributions of the reconstructed energies were fitted with the described two step Gaussian fit. The difference between the fitted peak value and the beam energy is shown in the bottom of Figure 14.

The energy of the test beam data had to be adjusted after the energy reconstruction with the neural network. Without a scaling factor, all energies would have been reconstructed with an energy approximately 2.75 % higher than the beam energy. To get a linearity of better than 2 % all reconstructed energies had to be scaled down by 2.75 %.

The gain in energy resolution can mainly be seen on the right plot of Figure 15. The improvement

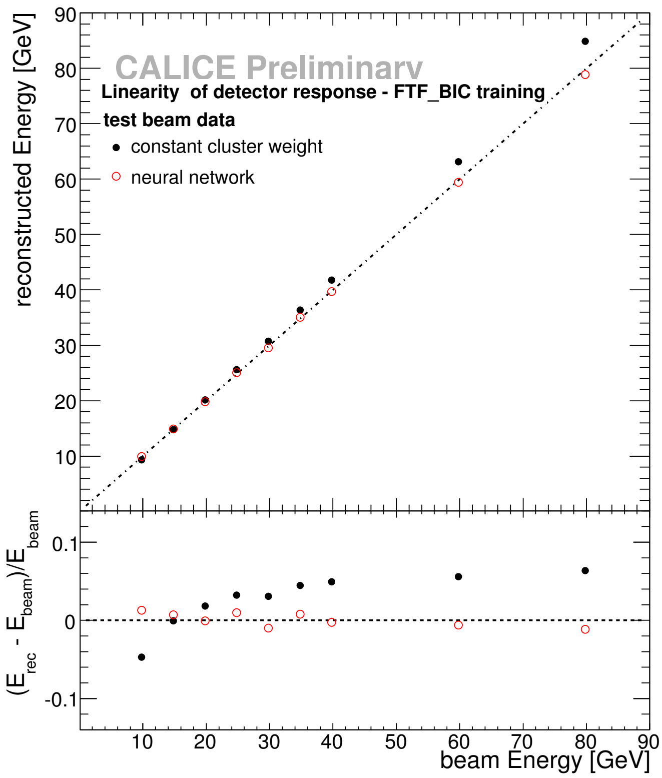

Figure 14. Linearity of test beam data with a single weight factor (black points) and with the weights applied (red circles) which were extracted from a neural network which was trained with FTF_BIC Monte Carlo data.

in energy resolution is, except for the 10 GeV point, between 19 % to 25 %. As shown on the right of Figure 15, the improvement of the energy resolution is largest in the middle energy range and reduces slightly for higher beam energies.

6.3.2 Energy reconstruction with a Neural Net trained with simulations with QGSP_BERT

The results of the energy reconstruction of test beam data with a neural network trained with QGSP_BERT simulated data are presented in Figure 16 and 17. As in the case for the FTF_BIC trained network a scaling factor of 2.75 % has been applied to the data. The largest improvement in energy resolution could be achieved at the medium energy range, shown on the right plot of Figure 17. An improvement of energy resolution, which is better than 19 %, can only be achieved in the energy range between 17 to 50 GeV. For higher energies the gain is less than for a neural network

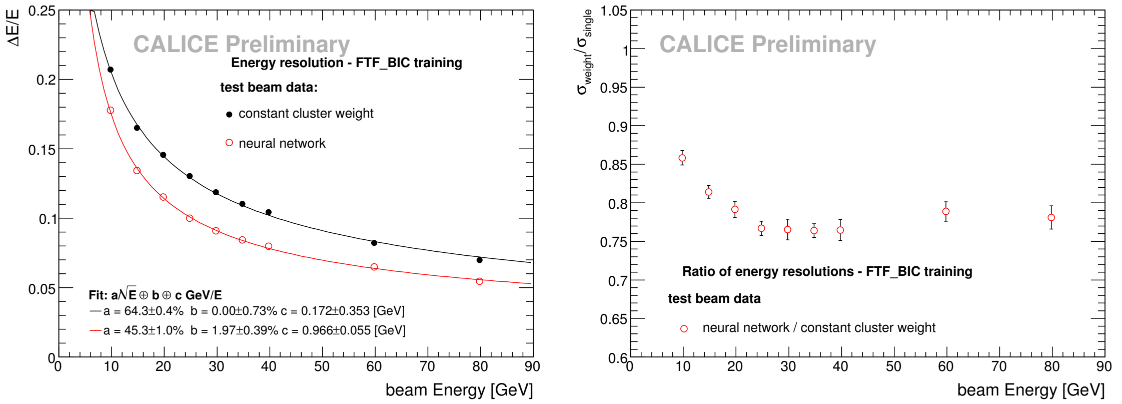

Figure 15. Left: Energy resolution of test beam data with a single weight factor (black points) and with the weights applied (red circles) which were extracted from a neural network which was trained with FTF_BIC Monte Carlo data.
Right: Ratio of energy resolution.

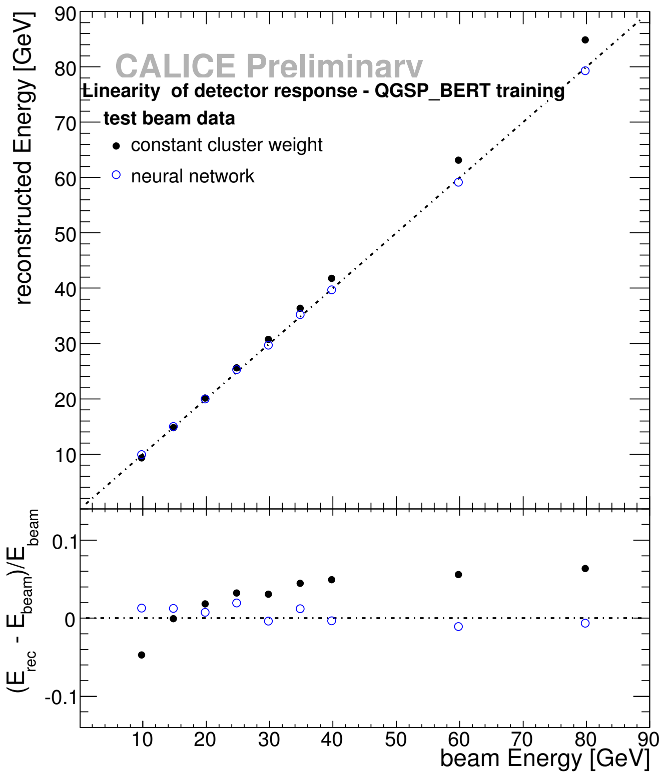

Figure 16. Linearity of test beam data with a single weight factor (black points) and with the weights applied (blue circles) which were extracted from a neural network which was trained with QGSP_BERT Monte Carlo data.

trained with FTF_BIC data. The neural network trained with QGSP_BERT data, gives a linearity of test beam data which is better than 2 %.

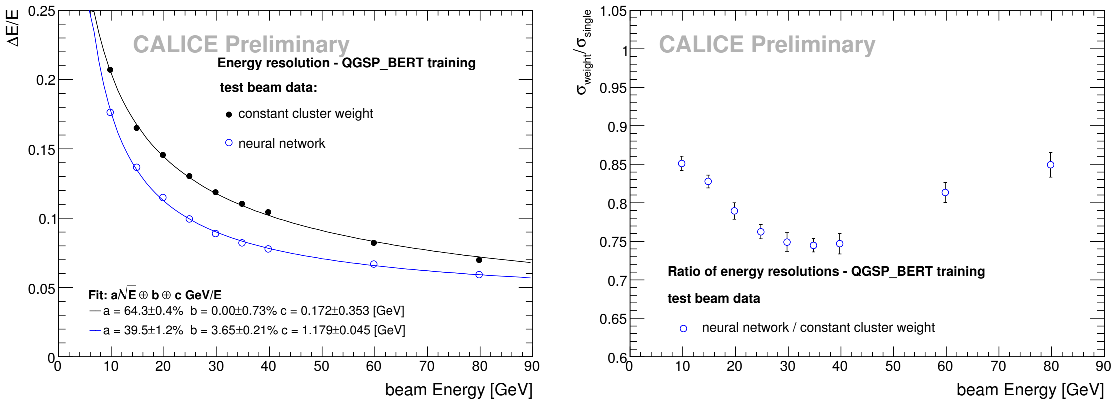

Figure 17. Left: Energy resolution of test beam data with a single weight factor (black points) and with the weights applied (blue circles) which were extracted from a neural network which was trained with QGSP_BERT Monte Carlo data.
Right: Ratio of energy resolution.

6.4 Conclusion

With the neural network technique an improvement in the energy resolution of around 23 % is reached with the two neural networks. Also, the reconstructed energy is closer to the beam energies over the full energy range than for the single cluster weight, leading to a significantly improved linearity.
The main difference between the application of the neural networks on test beam compared to Monte Carlo data is a residual discrepancy of the reconstructed energy. The test beam data had to be readjusted by a constant factor of 2.75 % and this is therefore a calibration effect. This is correlated with the fact, that the reconstructed cluster energy is the most important input variable in the neural network and need to be very similar for the training and testing data samples.
The network trained with FTF_BIC data gives the better energy resolution improvement over the full energy range, which is shown by the ratio of the energy resolutions.

7. Summary

Two analyses were presented which study software compensation with a weighting technique and a neural network on the hadronic cluster level. For the techniques only simulated data was used to extract weights and train neural networks respectively. It is the first time for CALICE analyses that Monte Carlo data is used to develop a software compensation techniques which then were successfully applied on test beam data.

The best results of the neural network technique, provided the neural network trained with FTF_BIC simulated data. An improvement of the energy resolution around 23 % could be achieved with a significantly improved linearity as well.

The simple technique using one weight per cluster, based on the energy density of the hadronic shower, improved the energy resolution by 15 %. These method has the advantage that it is straight forward to handle and understand. Here the physics list QGSP_BERT gave a slightly better performance, when applied to test beam data.

These software compensation techniques are also well suited for the integration into complex event reconstruction algorithms for a complete linear collider detector, such as the PandoraPFA particle flow algorithm [8].

References

[1] J. Repond *et al.* [CALICE Collaboration], *Design and Electronics Commissioning of the Physics Prototype of a Si-W Electromagnetic Calorimeter for the International Linear Collider*, JINST **3**, P08001 (2008).

[2] C. Adloff *et al.* [CALICE Collaboration], *Construction and Commissioning of the CALICE Analog Hadron Calorimeter Prototype*, submitted to JINST.

[3] A. Dyshkant, *Tail Catcher Muon Tracker for the CALICE test beam*, AIP Conf. Proc. 867 (2006) 592-599.

[4] G. Bondarenko *et al.*, *Limited Geiger-mode microcell silicon photodiode: New results*, Nucl. Instrum. Meth. A **442**, 187 (2000).

[5] A. Lucaci-Timoce, *Description of the Analog HCAL Prototype in Mokka*, http://www.desy.de/ lucaci/Others/hcalTBeam.pdf

[6] A. Höcker, P. Speckmayer, J. Stelzer, F. Tegenfeldt, H. Voss, K. Voss *et al.*, *TMVA - Toolkit for Multivariate Data Analysis with ROOT*, [arXiv:physics/0703039]

[7] F. Simon, K. Seidel [CALICE Collaboration], *Initial Study of Hadronic Energy Resolution in the Analog HCAL and the Complete CALICE Setup*, CALICE Analysis Note CAN-015 (2009).

[8] M. Thomson, *Particle flow calorimetry and the PandoraPFA algorithm*, Nuclear Instruments and Methods in Physics Research Section A, Volume 611, Issue 1, p. 25-40.

A. List of Runs

Run Number	Beam Energy in GeV	Partcile Type
330850	10	π^-
330647	15	π^-
330771	20	π^-
330650	25	π^-
331298	30	π^+
330551	35	π^-
330412	40	π^-
331282	60	π^+
331280	80	π^+

Table 1. The background is potentially quite different for the test beam runs due to the different particle type. π^+ runs have mainly proton background and π^- runs have mainly electron background. The Cherenkov trigger was used to eliminate this background sources.

B. Software Version

Test beam data reconstruction:

name	version
calice	calice-v02-00

Simulation:

name	version
geant4	4.9.3
mokka	7.02
detector model	TBCern0707_p0709
calice	calice-v02-00

C. Composition of physics lists

FTF_BIC: BIC up to 5 GeV; FTFB above 4 GeV
QGSP_BERT: BERT up to 9.9 GeV, LEP from 9.5 GeV up to 25 , QGSP above 12 GeV

D. Additional figures

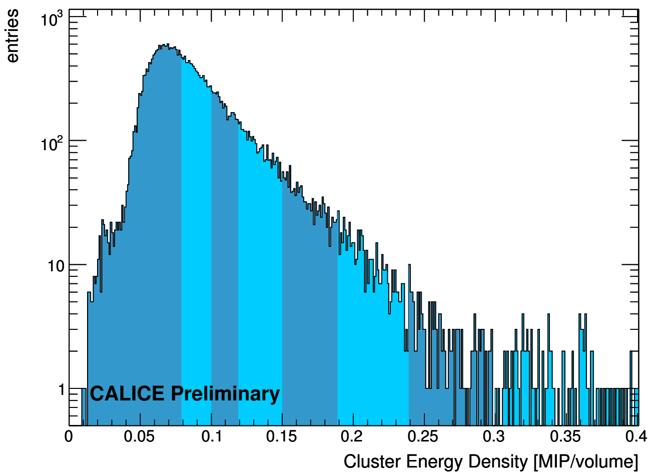

Figure 18. Cluster energy density for a 40 GeV run. The subdivision of the energy density into eight different bins is illustrated by the color shading.

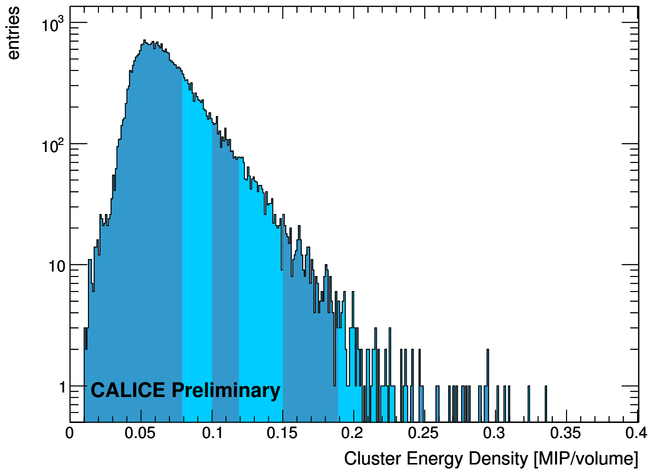

Figure 19. Cluster energy density of a 20 GeV. The subdivision of the energy density into eight different bins is illustrated by the color shading. The energy density is not as broad as for higher energies. Therefore less energy density bins are effectively used. There choice of the bin borders was strongly correlated to the energy density dependence of the individual weights.

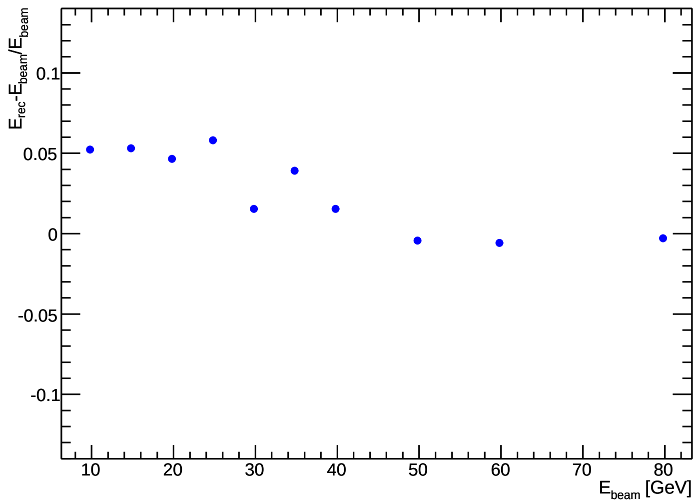

Figure 20. Difference of reconstructed and beam energy for test beam data and a neuronal net, trained with FTF_BIC simulated data. The reconstructed cluster energy in Monte Carlo of the training data set was not corrrected as described in Section 4. At low energies (< 40 GeV) the linearity is larger than 4 %. Without the “weight expression“ in the training phase of the neural net the difference between reconstructed and beam energy would be even higher at low energies.

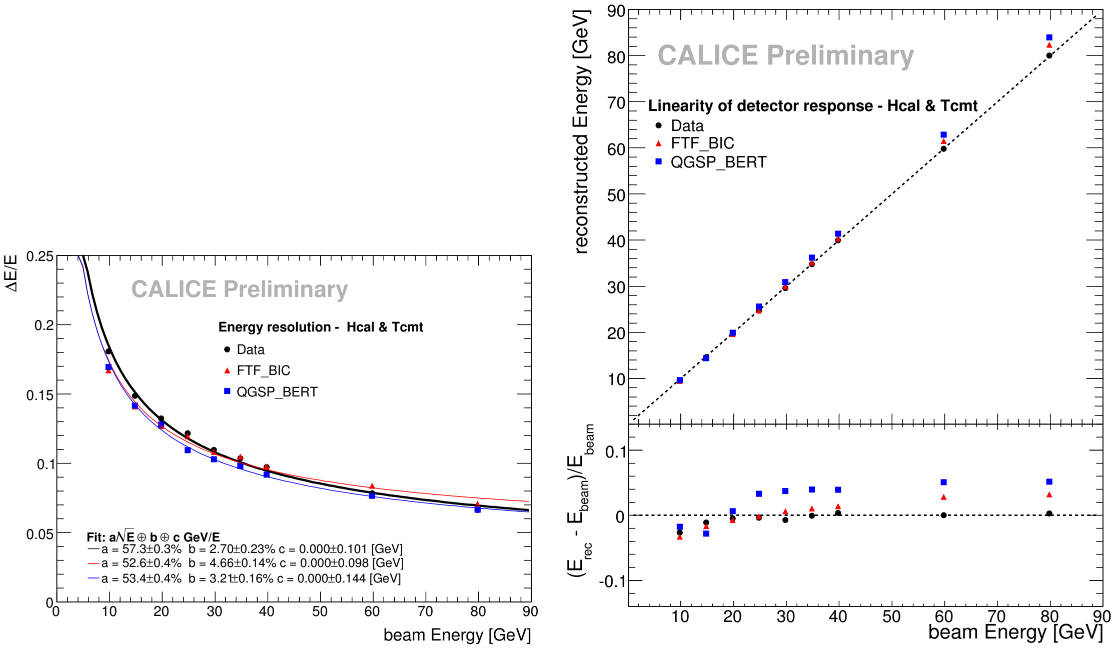

Figure 21. Energy resolution (left) and linearity (right) of test beam (black points), QGSP_BERT (blue squares) simulated and FTF_BIC (red triangles) simulated data without clustering (all energy in AHCAL and TCMT) and a single weight factor (MIP to GeV 0.028).

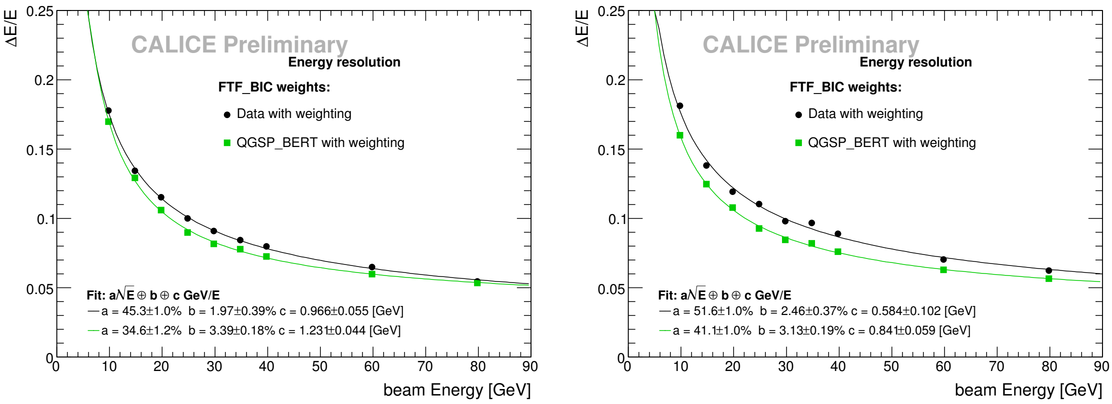

Figure 22. Energy resolution of test beam (black points), QGSP_BERT (green squares) simulated clustered data. **Left:** Neural Net trained with FTF_BIC simulated data applied on both data set. **Right:** Weights extracted with FTF_BIC simulated data applied on both data set.

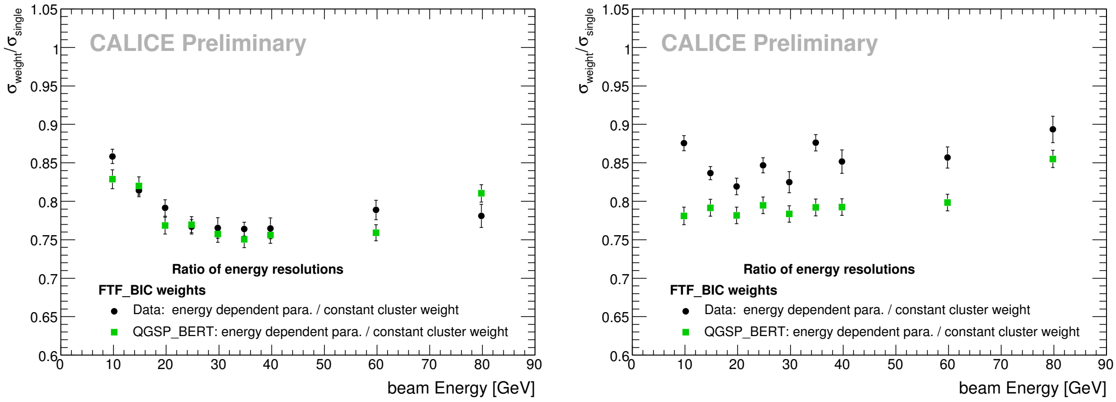

Figure 23. Ratio of energy resolution of test beam (black points), QGSP_BERT (green squares) simulated clustered data. **Left:** Neural Net trained with FTF_BIC simulated data applied on both data set. **Right:** Weights extracted with FTF_BIC simulated data applied on both data set.

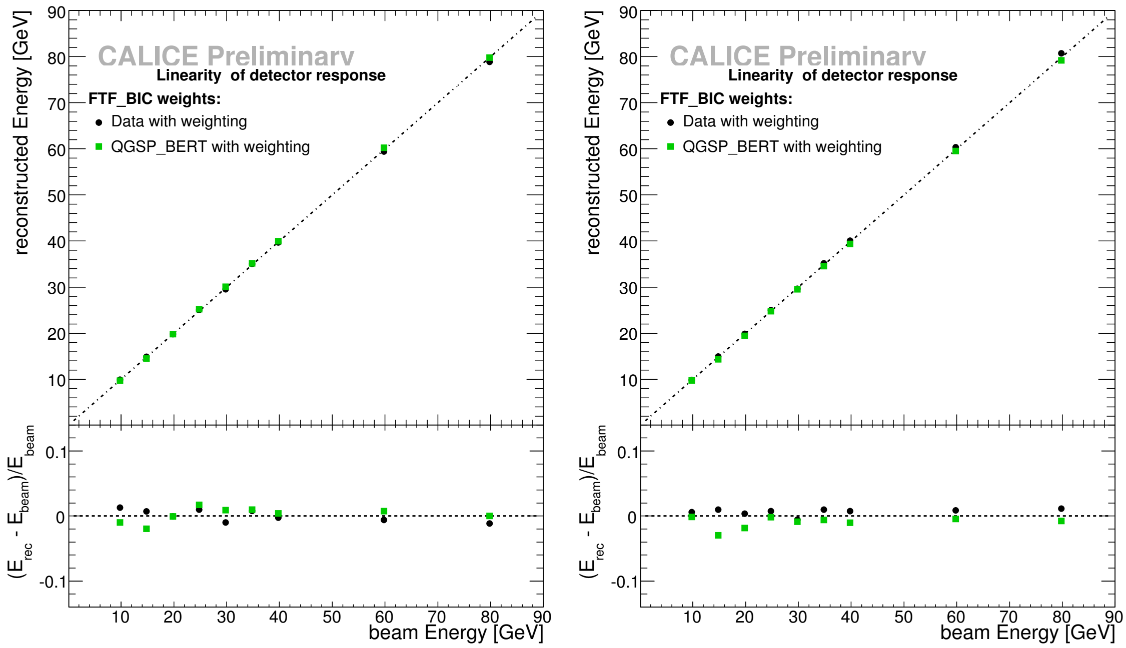

Figure 24. Linearity of test beam (black points), QGSP_BERT (green squares) simulated clustered data. **Left:** Neural Net trained with FTF_BIC simulated data applied on both data set. **Right:** Weights extracted with FTF_BIC simulated data applied on both data set.

E. Energie Density Weighting - Weights on Monte Carlo data

The results of the cluster energy density weighting technique on one Monte Carlo data set with weights extracted from the other Monte Carlo data set and vice versa, are presented in Figures 25, 26 and 27.

The linearity, see Figure 27, is better than 4 %, for both data sets.

The ratio of energy resolutions (right plots of Figure 25 and 26) are similar for energies between 10 to 40 GeV. At higher energies the weights obtained with data from the physcis list QGSP_BERT give better results. The gain in energy resolution is around 20 %.

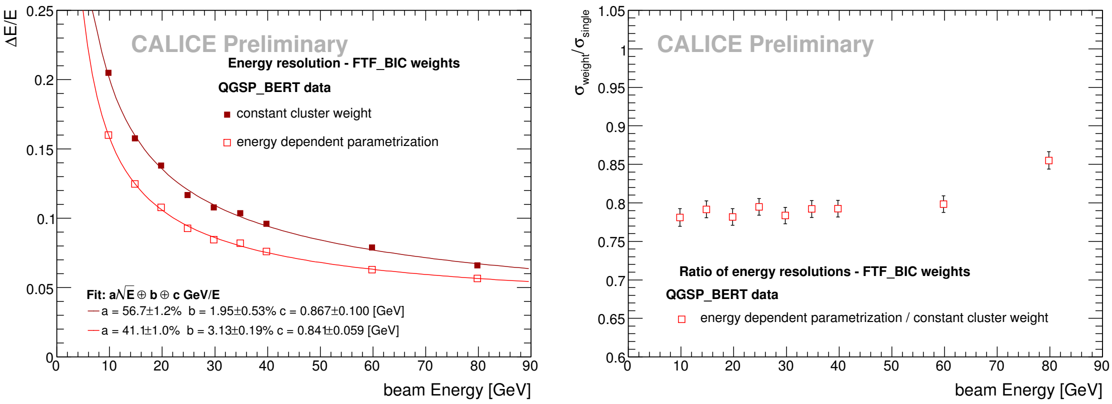

Figure 25. Left: Energy resolution of test QGSP_BERT data with a single weight factor (dark red squares) and with the weights applied (open red squares) which were extracted with the cluster energy density weighting approach of FTF_BIC Monte Carlo data.
Right: Ratio of energy resolution.

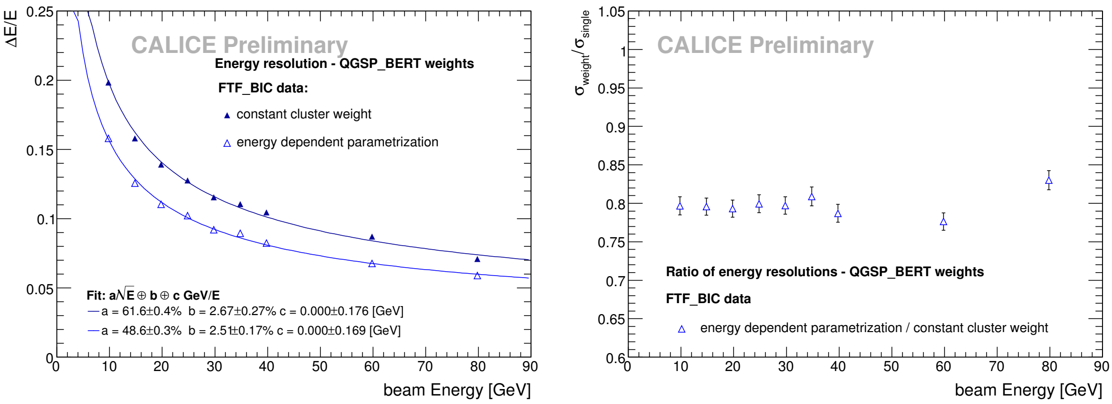

Figure 26. Left: Energy resolution of test FTF_BIC data with a single weight factor (dark blue triangles) and with the weights applied (open blue triangles) which were extracted with the cluster energy density weighting approach of QGSP_BERT Monte Carlo data.
Right: Ratio of energy resolution.

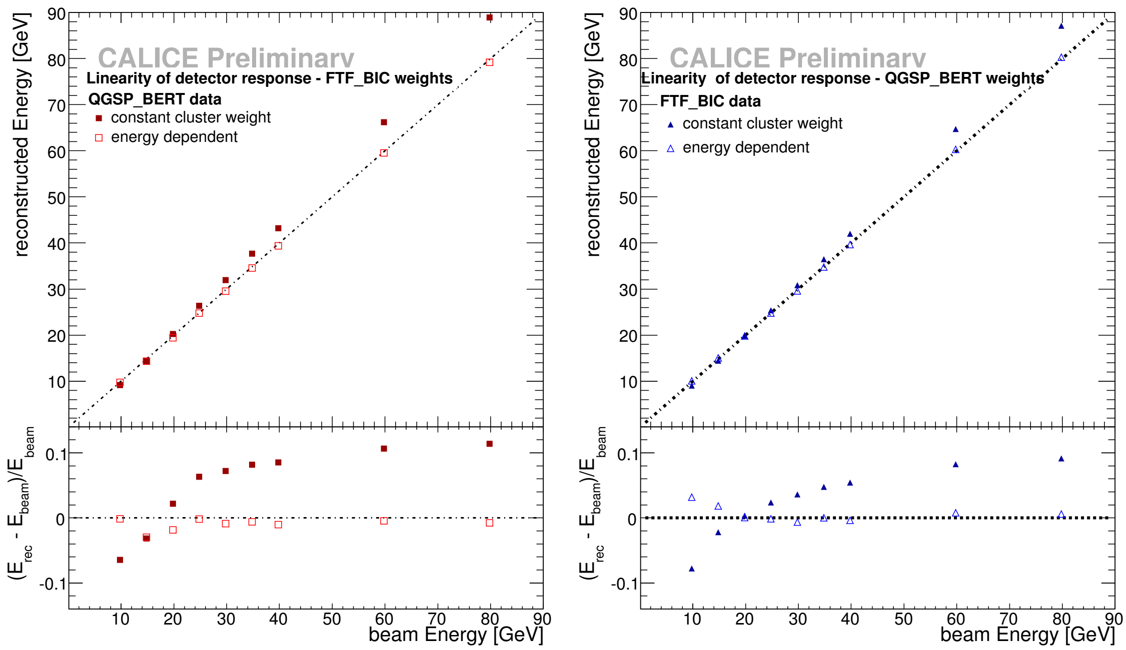

Figure 27. Left: Linearity of QGSP_BERT data with a single weight factor (dark red squares) and with the weights applied (open red squares) which were extracted with the cluster energy density weighting approach of FTF_BIC Monte Carlo data.
Right: Linearity of FTF_BIC data with a single weight factor (dark blue traingles) and with the weights applied (open blue triangles) which were extracted from a neural network which was trained with QGSP_BERT Monte Carlo data.

F. Neural Network on Monte Carlo data

To check the stability of this neural network technique with respect to the choice of the chosen physics list, the weights of the FTF_BIC physics list were also applied on the QGSP_BERT simulated data and vice versa.
Figures 28 and 29 show the energy resolutions and the ratios of the energy resolutions with and without the neural network applied.
The neural network trained with FTF_BIC data shows the higher gain in energy resolution of around 23 % (Figure 29). Over a larger energy range the improvement is bigger compared to the neural network trained with QGSP_BERT data (Figure 28). The linearity, see Figure 30, is better than 3 % for both neural networks over the full energy range.

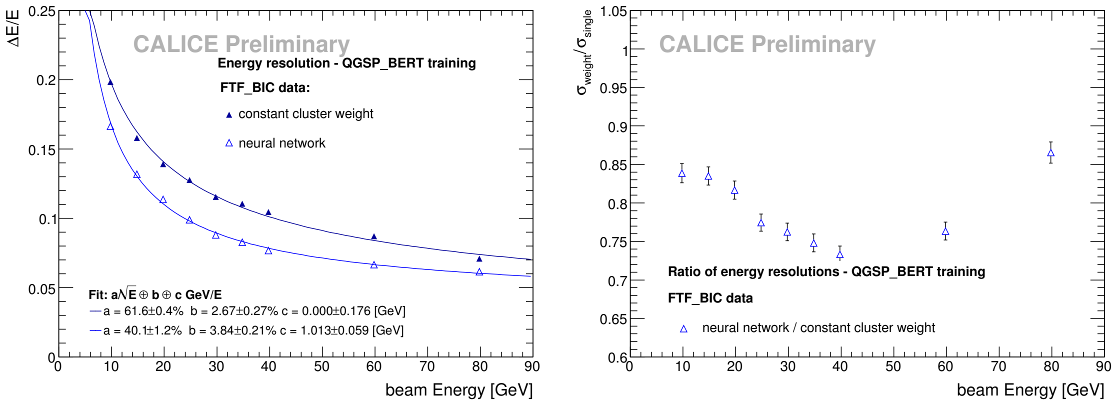

Figure 28.
Left: Energy resolution of FTF_BIC data with a single weight factor (dark blue points) and with the weights applied (blue points) which were extracted from a neural network which was trained with QGSP_BERT Monte Carlo data.
Right: Ratio of energy resolution.

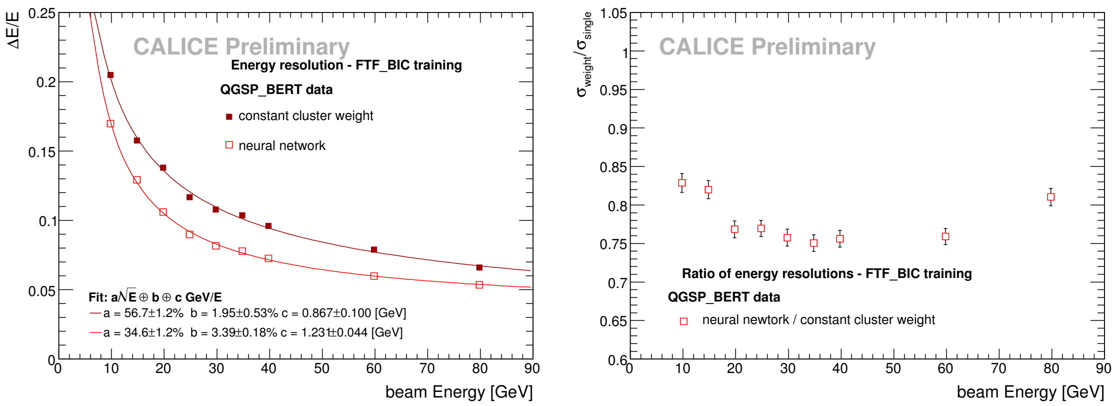

Figure 29.
Left: Energy resolution of test QGSP_BERT data with a single weight factor (dark red points) and with the weights applied (red points) which were extracted from a neural network which was trained with FTF_BIC Monte Carlo data.
Right: Ratio of energy resolution.

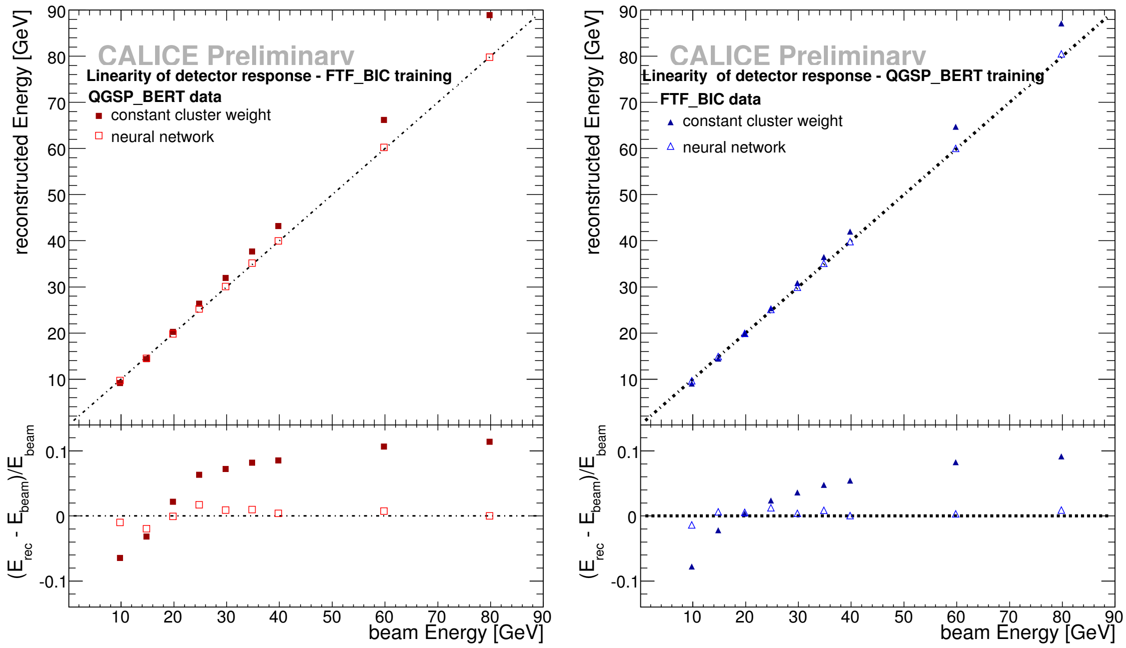

Figure 30.
Left: Linearity of QGSP_BERT data with a single weight factor (dark red points) and with the weights applied (red points) which were extracted from a neural network which was trained with FTF_BIC Monte Carlo data.
Right: Linearity of FTF_BIC data with a single weight factor (dark blue points) and with the weights applied (blue points) which were extracted from a neural network which was trained with QGSP_BERT Monte Carlo data.