Hadron energy resolution in the ILD with global compensation based on neural network approach

S. Korpachev (MIPT/LPI)

CALICE Collaboration meeting, 30 September 2020

Outline

1. Goals and motivation

2. Details of the work

3.Correlation between variables

4. Deep neural network

5. Preliminary results and ideas

6. Plans

Goals and Motivation

Highly granular calorimeters provide some additional information about the structure of hadronic showers.

The goal is to improve the energy resolution using the information about shower substructure in highly granular calorimeter.

The task is to apply the extended version of global compensation approach (see CAN-028 and CAN-035).

The current presentation focuses on implementation of global compensation method based on neural network technology. The global compensation means that variables used characterise a shower as a whole.

Some details of the work

- iLCSoft: x86_64_gcc49_sl6 / v02-00-02
- Version of the ILD for dd4hep: ILD_l5_v02.xml
- Version of the ILD for marlin reconstruction: ILD_o1_v02.xml

- Geant4 hadronic model: FTFP_BERT
- SIM.gun.isotrop is True
- Particles are single K0L with energies: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 80, 90, 120 GeV

- There are 5000 and 10000 events for each energy point
- Only hits from CalorimeterHit collections are used
- Energy is a sum of all hits (in GeV) in both calorimeters (ECAL + HCAL)
- The calibration is standard from iLCSoft
- No clustering is applied

Cuts:

- Absolute value of pseudorapidity is up to the end of the calorimeter system ($\eta < 3.0$)
- Events rejected when both (ecal + hcal) CalorimeterHit collections are empty
- If primary and doesn’t decay in the calorimeter system => skip an event (about 5-10% particles from full set are interacting or decaying before calorimeter)

Number of events after cuts

Full set of 10000 events

Energy	1	3	5	10	30	40	50	70	90	120
N	7727	8492	8632	8814	8912	8858	8897	8895	8950	8964
%	77.27	84.92	86.32	88.14	89.12	88.58	88.97	88.95	89.50	89.64

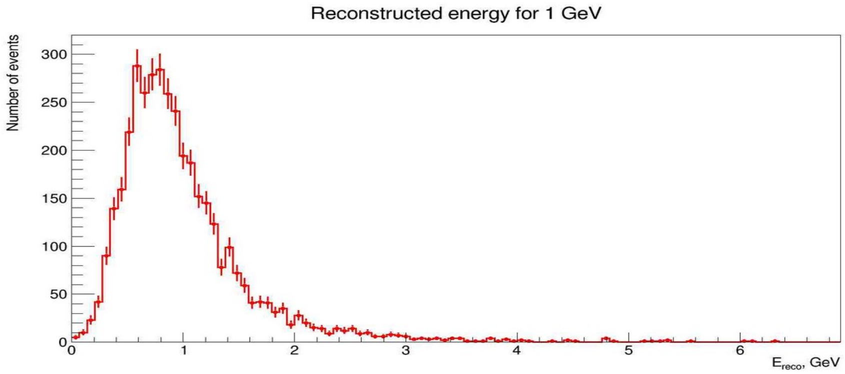
Similar efficiencies for all energies, slightly lower for 1 GeV

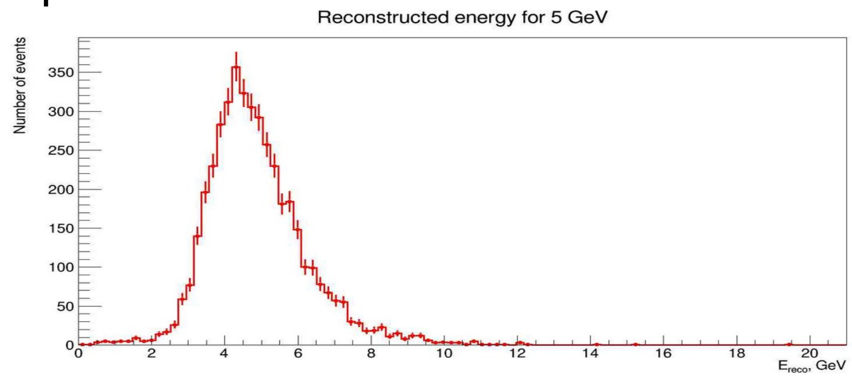
Full set of 5000 events

Energy	1	3	5	10	30	40	50	70	90	120
N	3850	4242	4344	4342	4446	4463	4439	4456	4431	4455
%	77.00	84.84	86.88	86.84	88.92	89.26	88.78	89.12	88.62	89.10

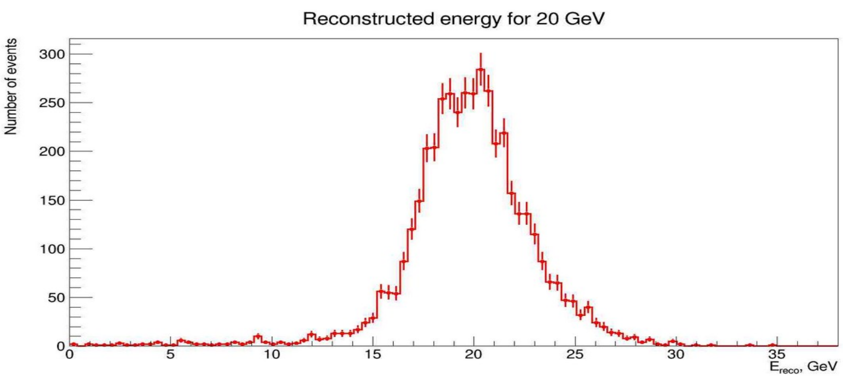
Energy distribution for single hadron: 1, 5, 20 and 40 GeV

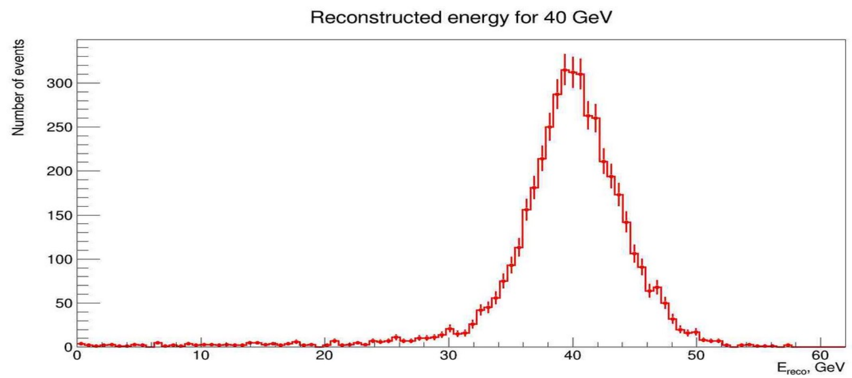
For 5000 sample





Asymmetric distributions with right tail



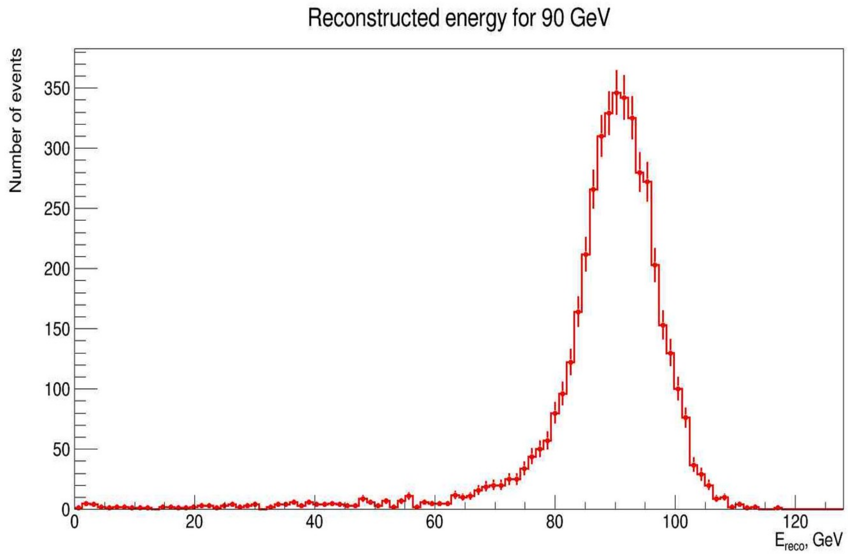


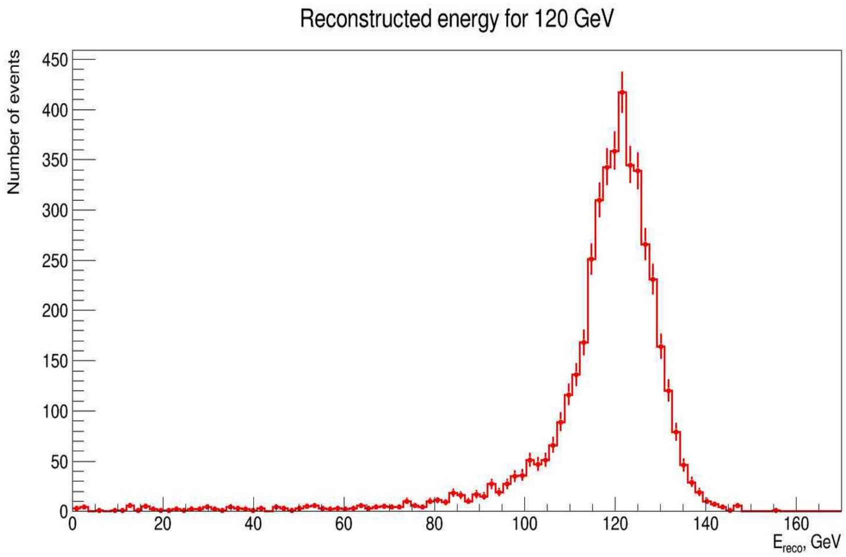
Gaussian shape

$$Energy=\sum_{i}^{N_{hits}} Ehit_i$$

Energy distribution for single hadron: 90 and 120 GeV

For 5000 sample

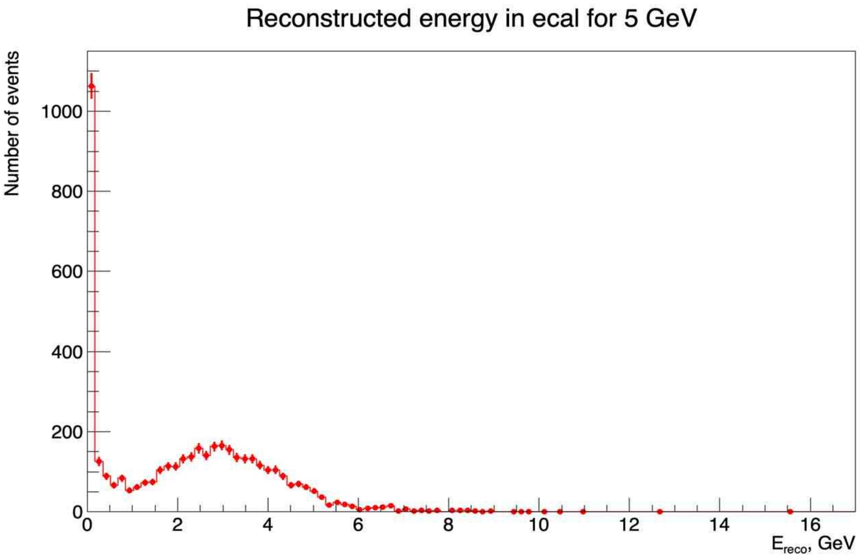


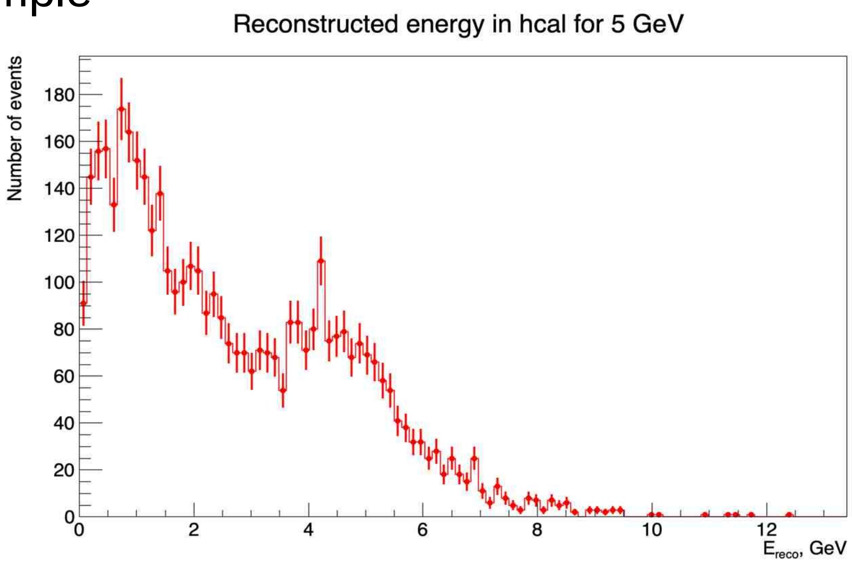


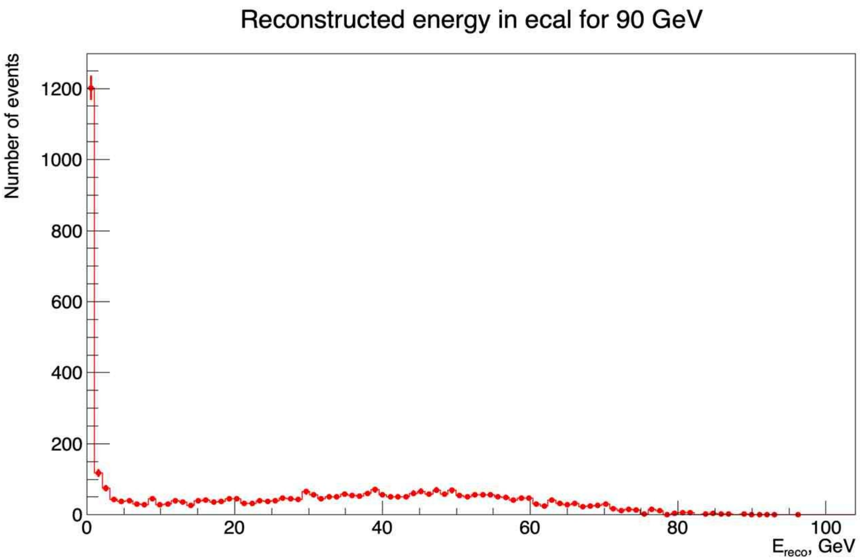
Asymmetric distributions with left tail (leakage)

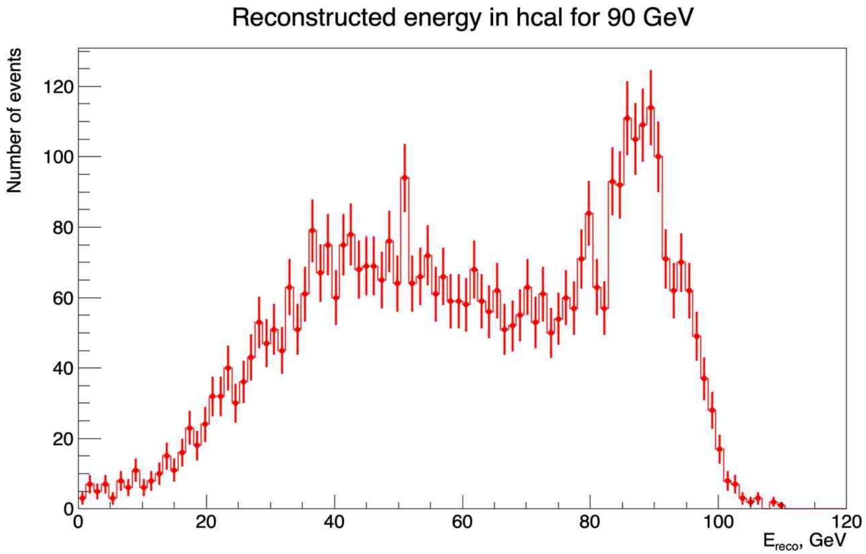
Energy distributions in ecal and hcal: 5 and 90 GeV

For 5000 sample







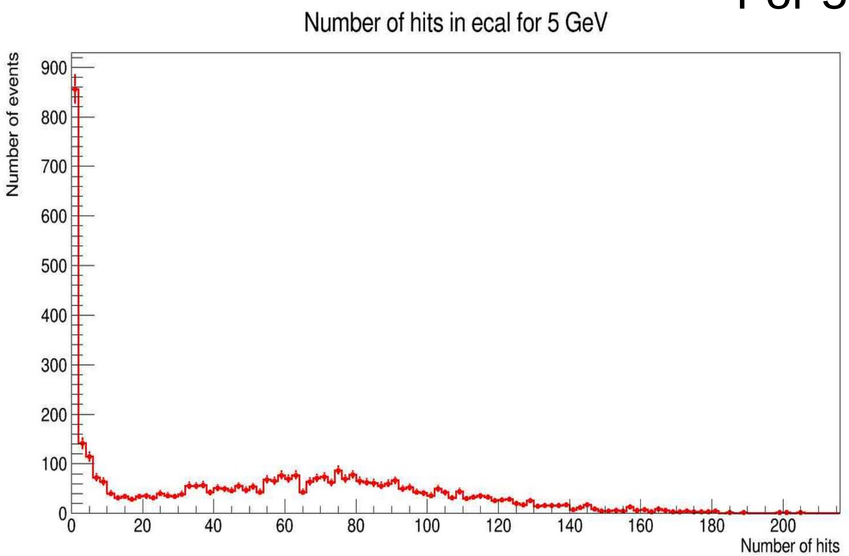


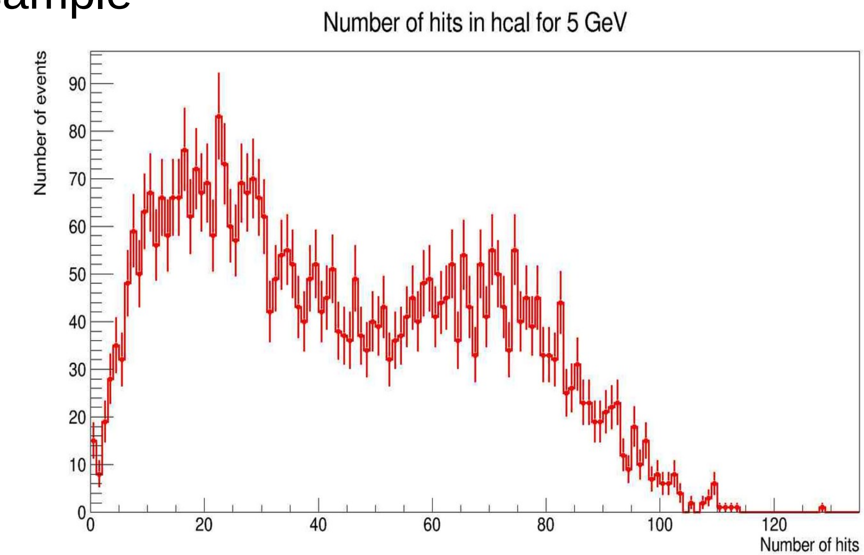
Reconstructed energy for each calorimeter in GeV

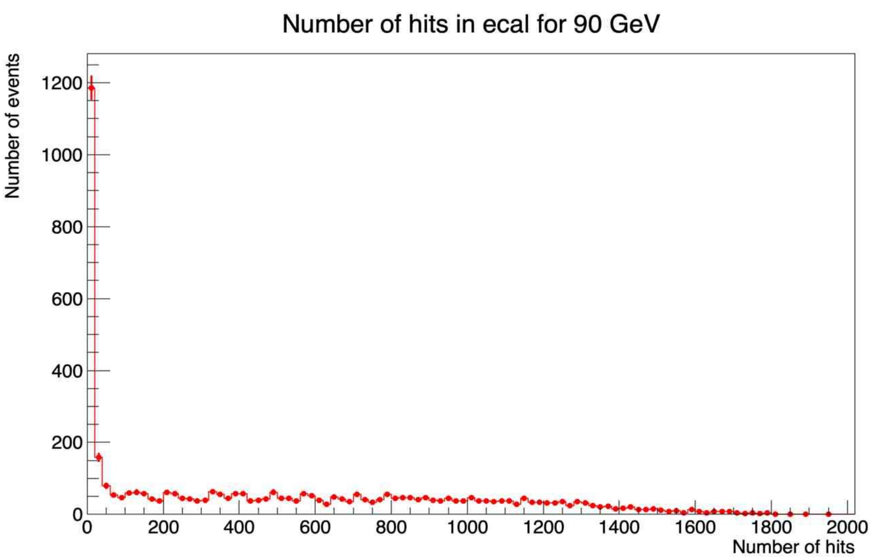
Peak at zero in ecal means hadrons start in hcal

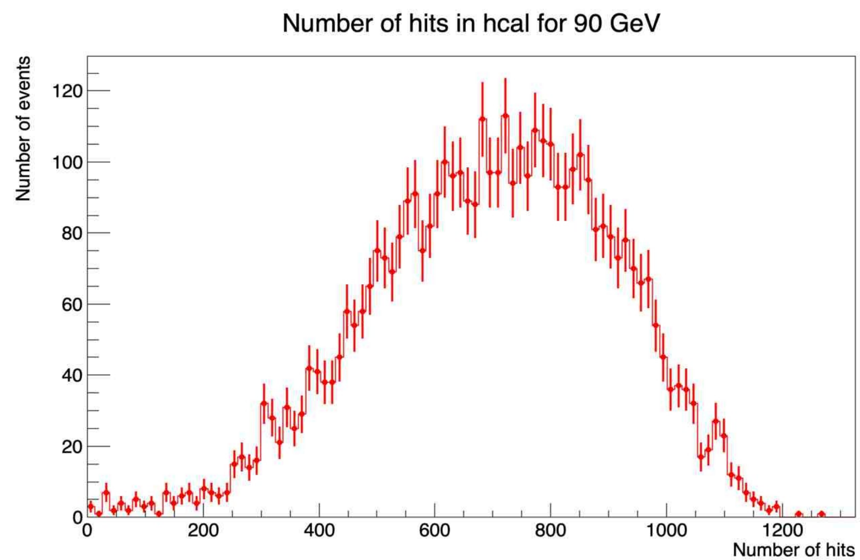
Distribution of number of hits: 5 and 90 GeV

For 5000 sample





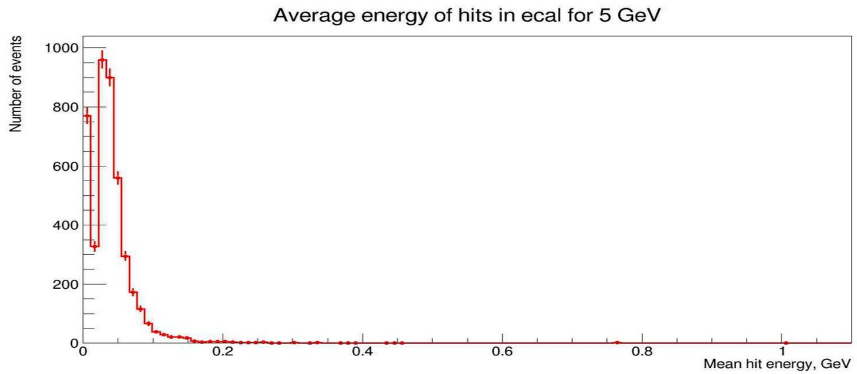


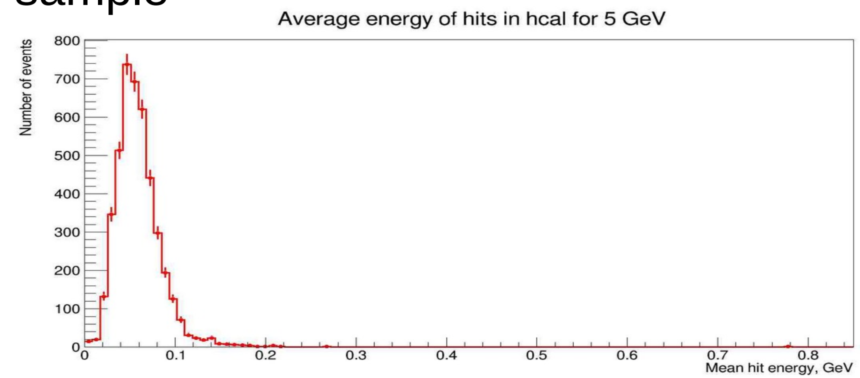


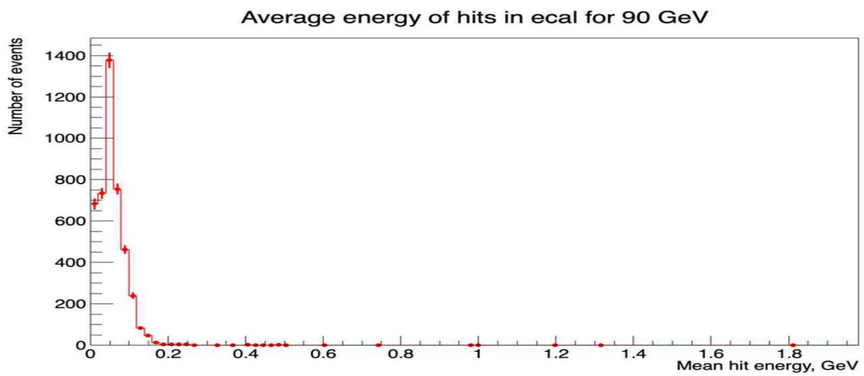
Number of hits in ecal and hcal
Peak at zero in ecal means hadrons start in hcal

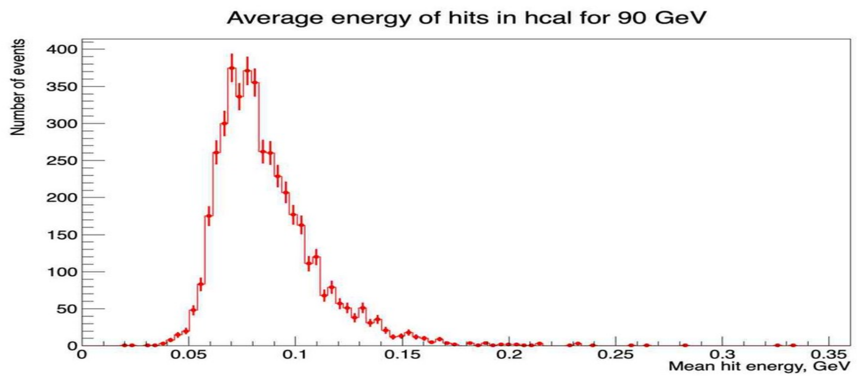
Mean hit energy in ecal and hcal: 5 and 90 GeV

For 5000 sample









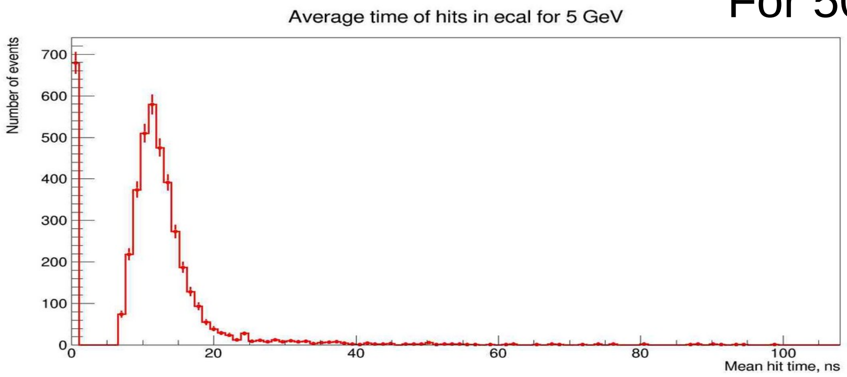
Average energy of hits in ecal and hcal in GeV

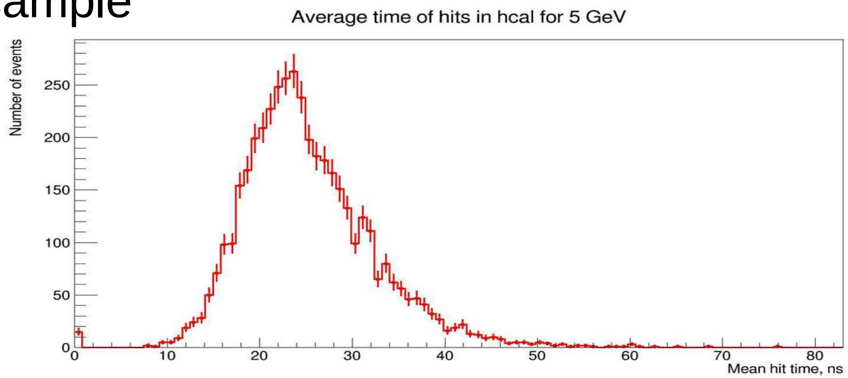
Peak at zero in ecal means hadrons start in hcal

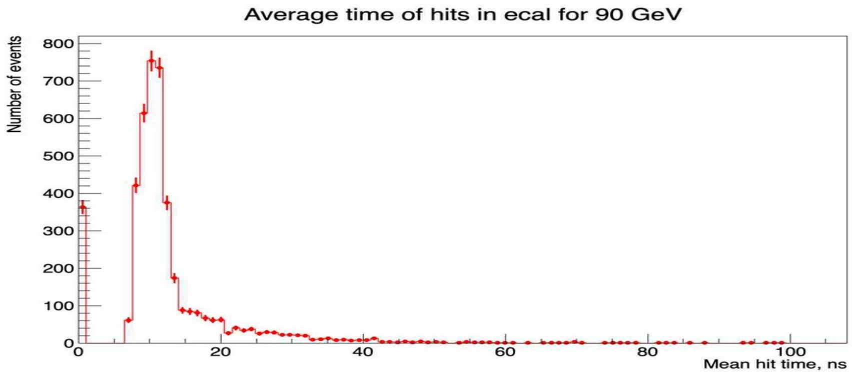
$$Mean\,hit\,energy = \frac{\sum_{i}^{N_{hits}} Ehit_i}{N_{hits}}$$

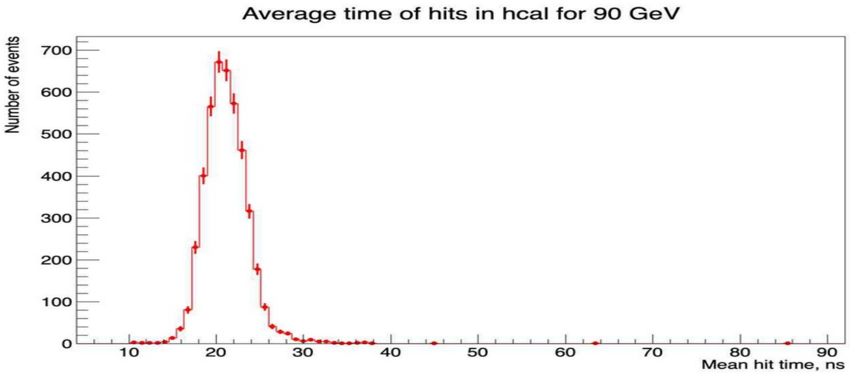
Mean hit time: 5 and 90 GeV

For 5000 sample









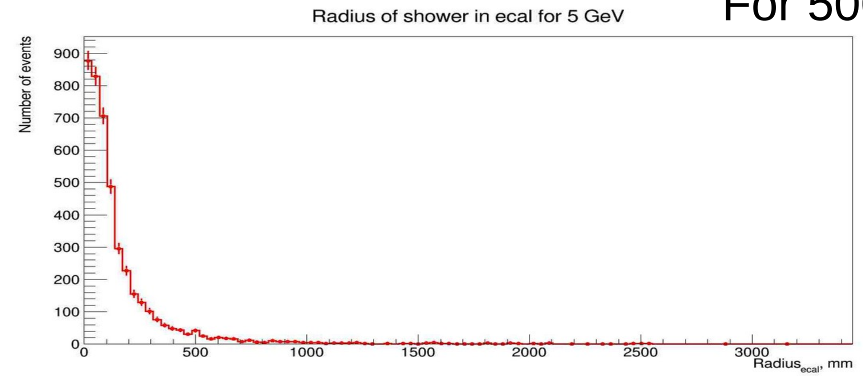
Mean time of hits in ecal and hcal in ns

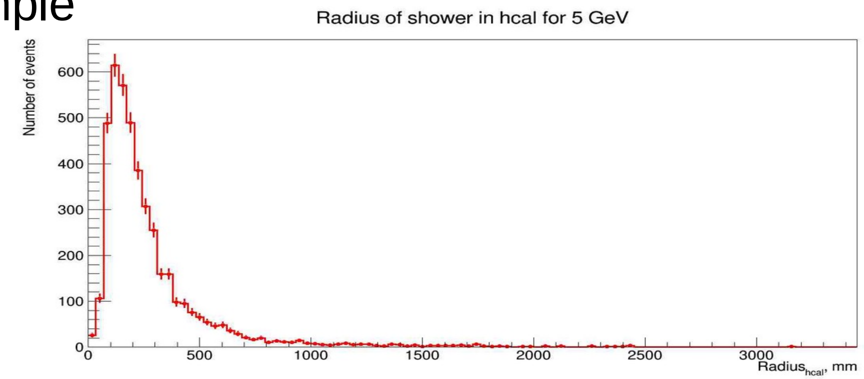
Peak at zero in ecal means hadrons start in hcal

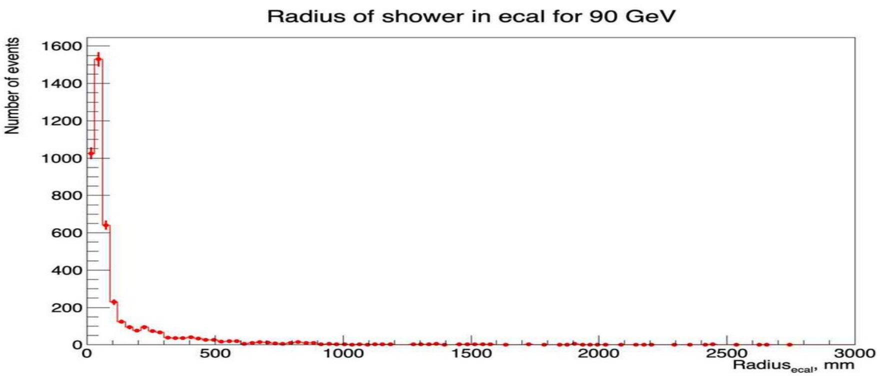
$$Mean\, hit\, time = \frac{\sum_{i}^{N_{hits}} Thit_i}{N_{hits}}$$

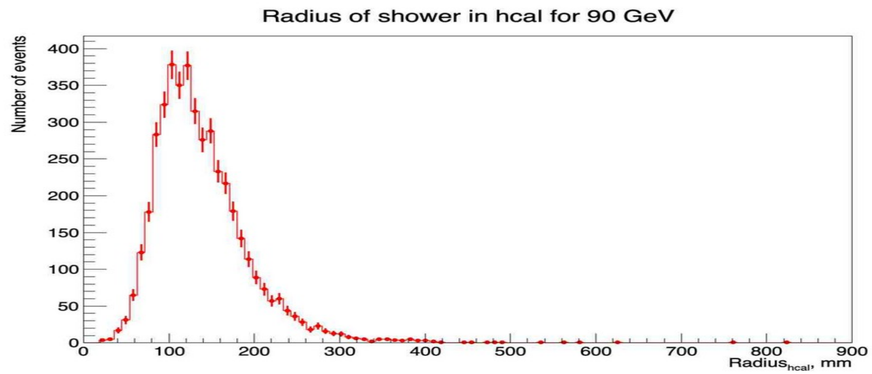
Energy weighted radius of shower: 5 and 90 GeV

For 5000 sample









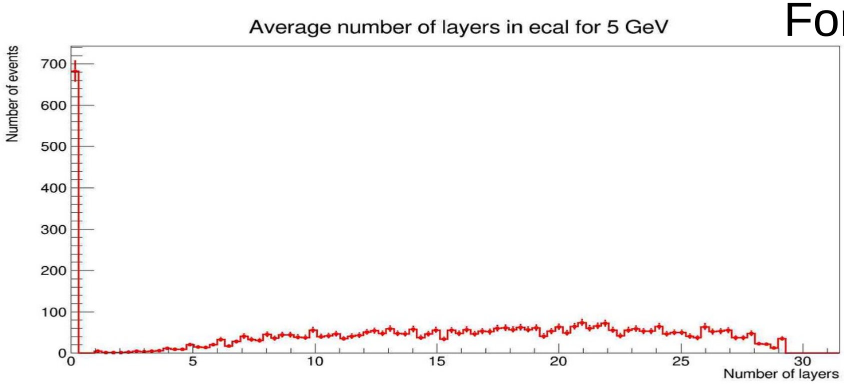
Radius of shower for ecal and hcal in mm

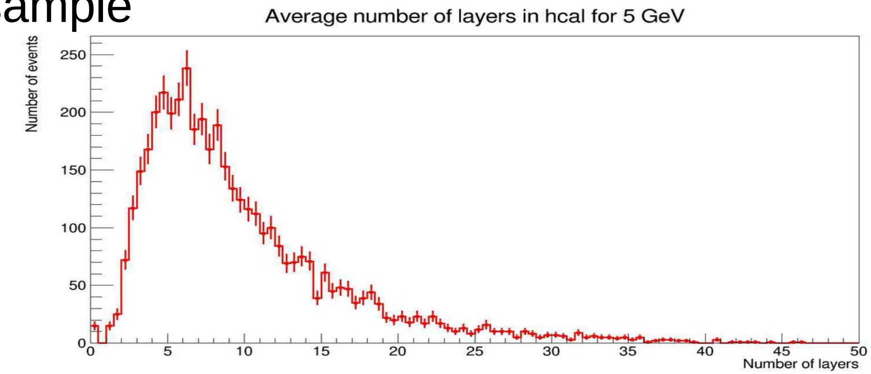
Distance between each hit position and straight line of IP (0,0,0) and cog (x,y,z)

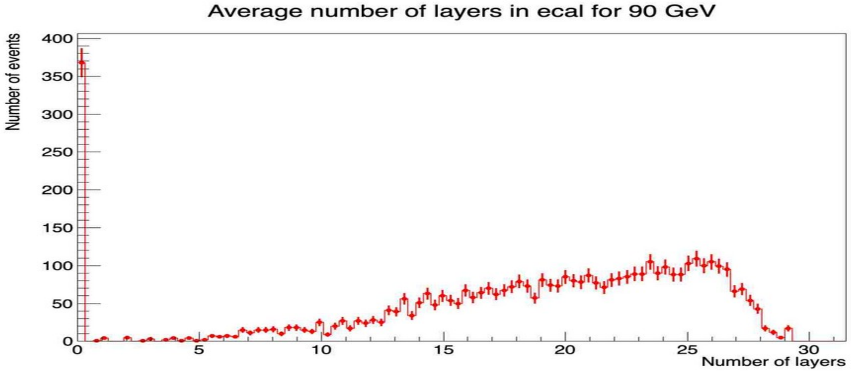
$$Energy\ weighted\ radius\ of\ shower = \frac{\sum_{i}^{N_{hits}} Distance\left(\left(IP, CoG\right), hit\ position_i\right) * Ehit_i}{\sum_{i}^{N_{hits}} Ehit_i}$$

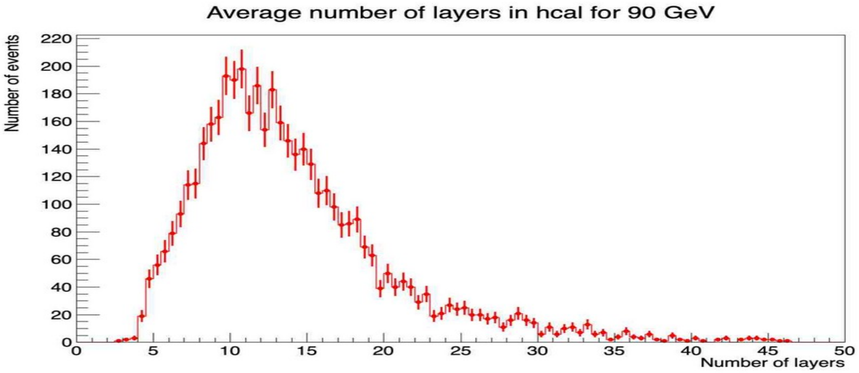
Weighted number of layers: 5 and 90 GeV

For 5000 sample









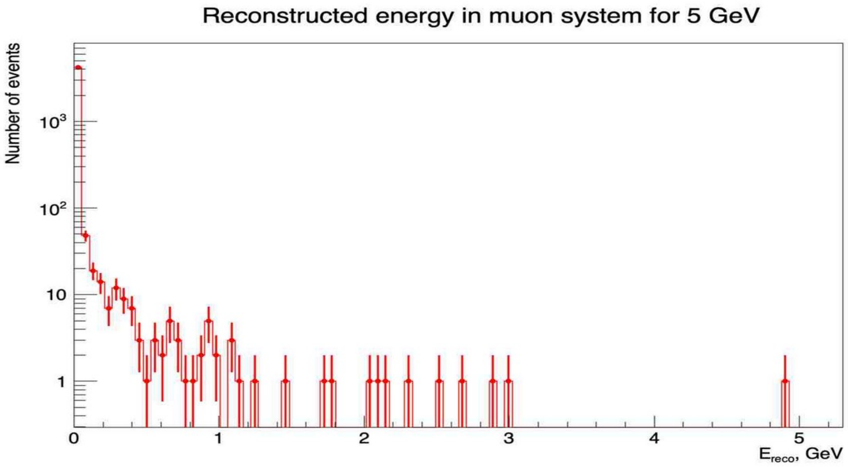
Average number of layer for each events in ecal and hcal

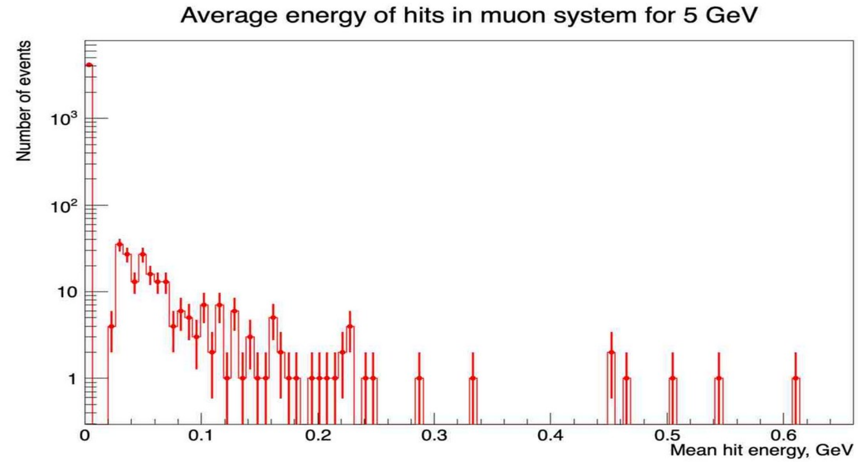
Peak at zero in ecal means hadrons start in hcal

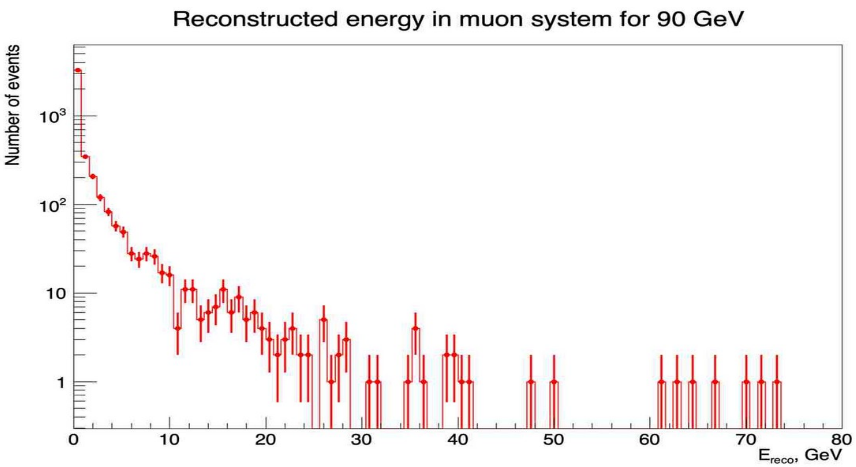
$$Weighted\,number\,of\,layer = \frac{\sum_{i}^{N_{hits}} Number\,hit\,layer_i * Ehit_i}{\sum_{i}^{N_{hits}} Ehit_i}$$

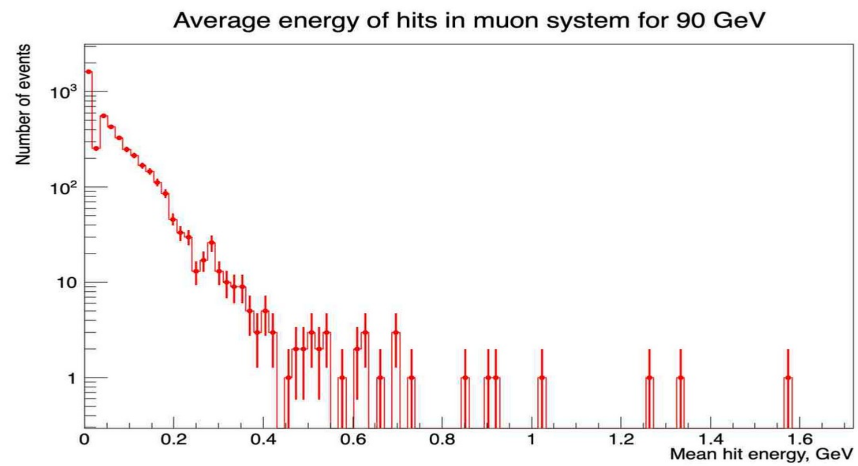
Energy and hit energy in muon system: 5 and 90 GeV

For 5000 sample









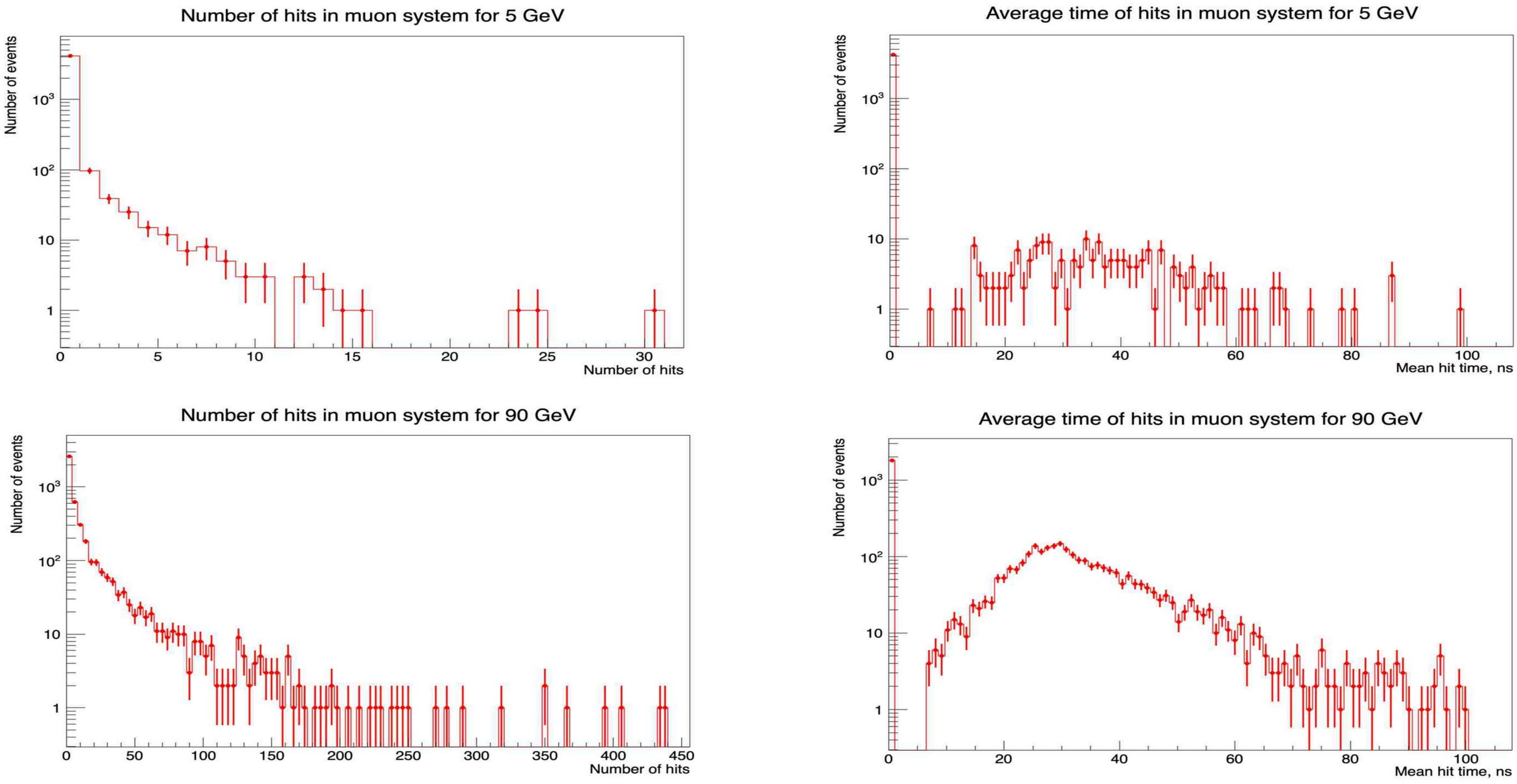
Reconstructed energy in GeV

Average energy of hits in GeV

About 5% (63%) of hadrons reach muon system for 5 (90) GeV

Number of hits and hit time in muon system: 5 and 90 GeV

For 5000 sample

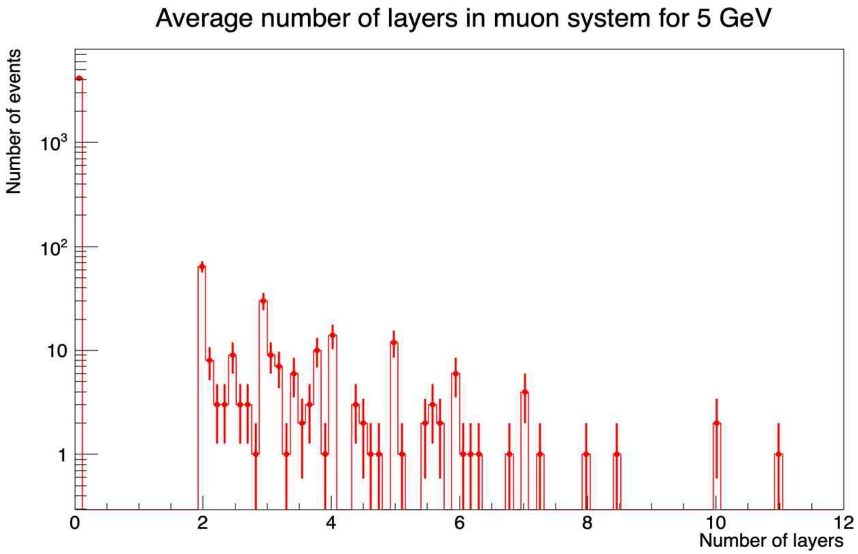


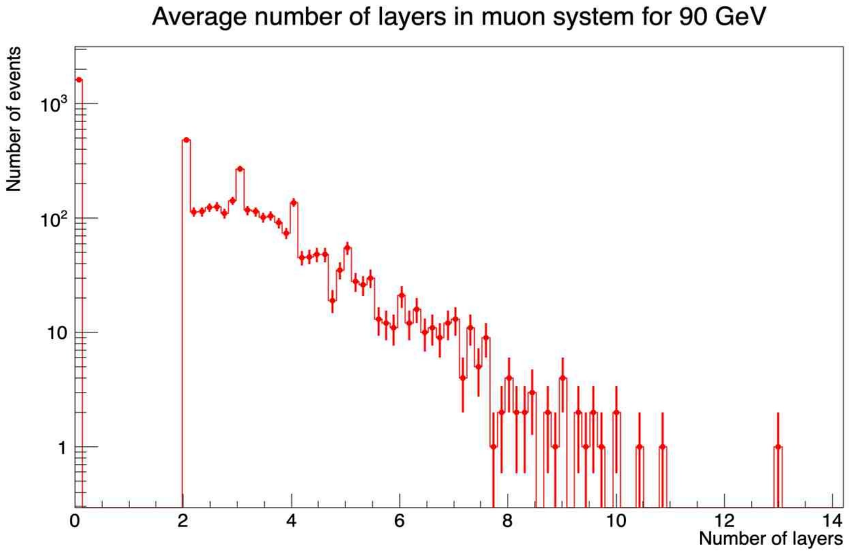
Number of hits

Mean time of hits in ns

Weighted number of layers in muon system: 5 and 90 GeV

For 5000 sample





Average number of layer

Target: true energy from mc collection

The largest positive correlation with eecal, ehcal and mean hit time in muon system, the largest negative correlation with radius in hcal and mean time of hit in hcal

Preprocessing

- 17 input variables and 1 target
- no data normalization

- 26 energies * 10000 = 260k events
- after cuts we have about 228k events (full set)

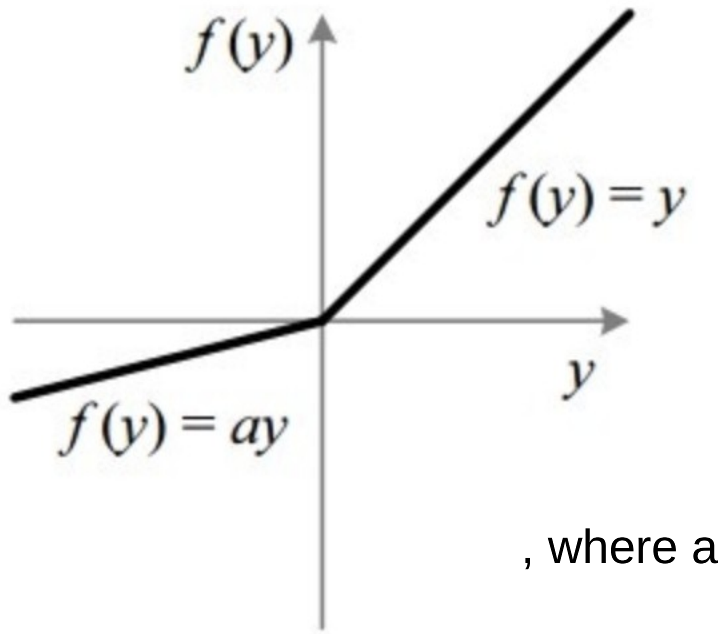
- train is about 46k events (20% of full set)
- validation is about 182k events (80% of full set)
- events are selected randomly without intersections

- test is 29 energy samples with 5000 events on each energy point
- further results of DNN performance are for test set

The full sample contains single hadron events in all energy range
studied (except for 7, 45 and 120 GeV). The last three energies only for test of DNN.

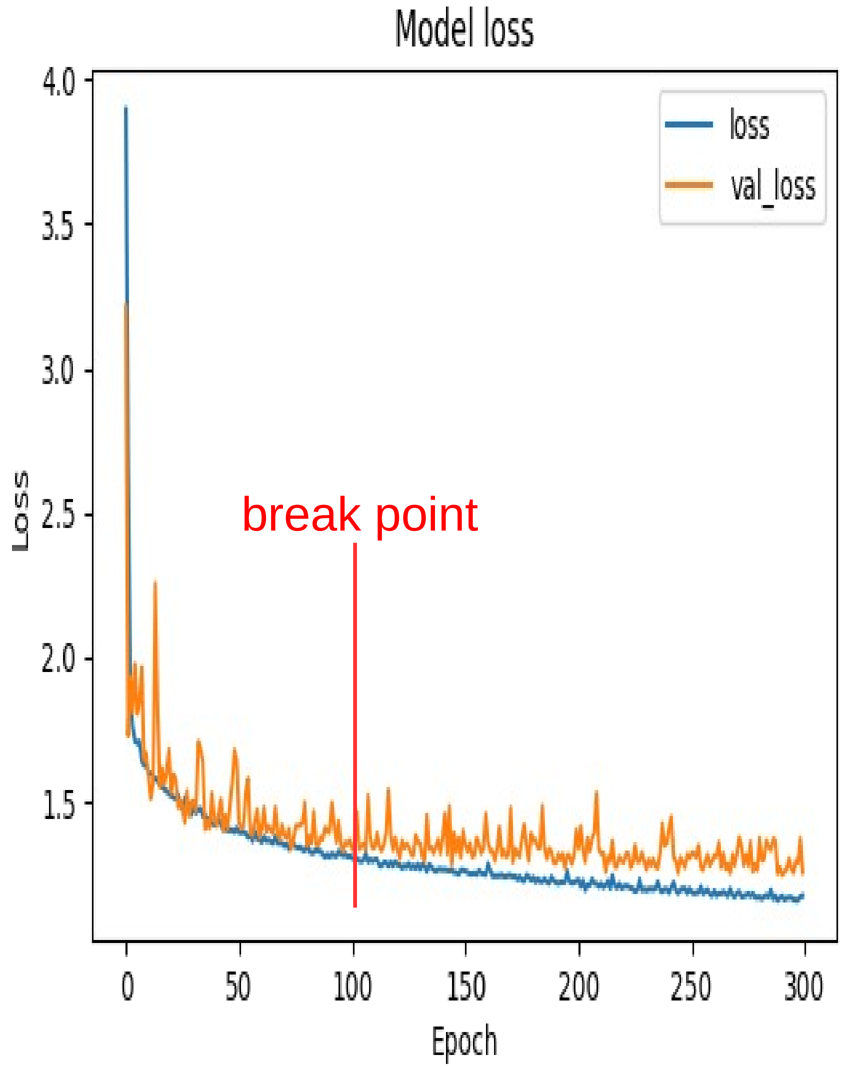
Neural network structure

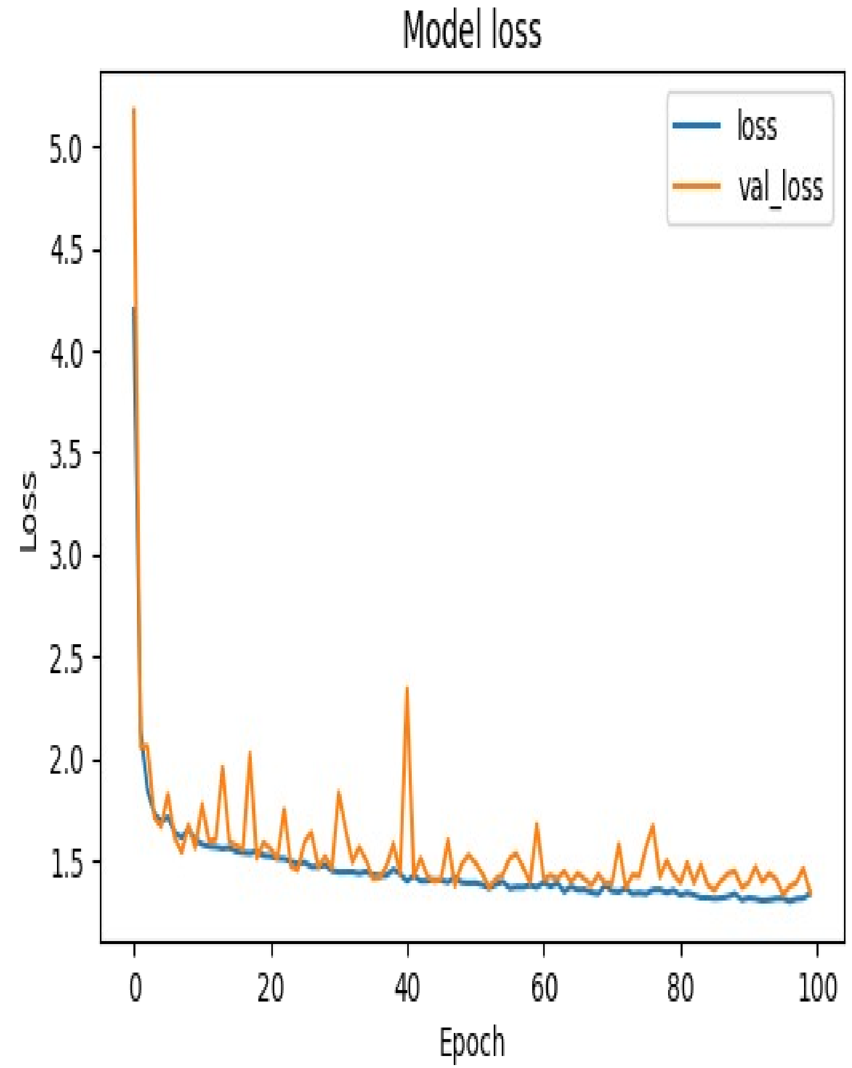
- Library Keras
- Layers: 1 input layer, 5 hidden layer, 1 output layer
- Number of neurons: 17 / 128 / 64 / 32 / 16 / 4 / 1
- Neuron activation function: leaky_relu for hidden layers and linear (f(y)= y) for output layer
- Number of epochs: 100



, where a is 0.3

Loss function





Loss function is $\frac{1}{N}*\sum_{i}^{N}\frac{(Epred_i - Etrue_i)^2}{0.25*Etrue_i}$, where Epred is predicted energy and Etrue is true energy from mc collection.

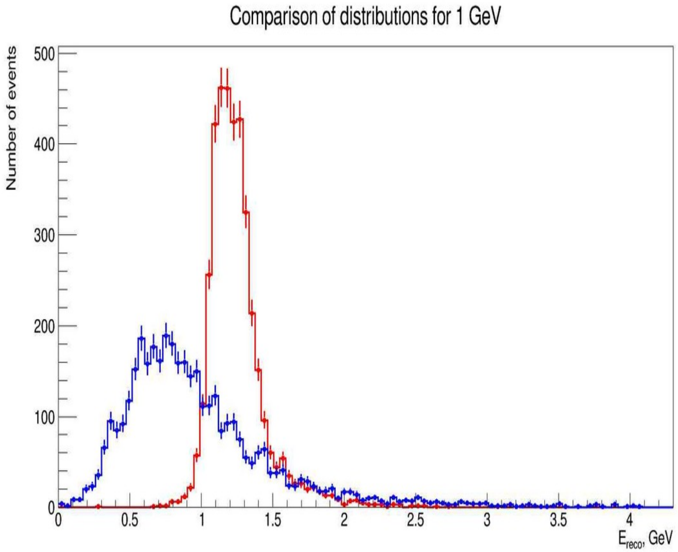
Hist90 procedure

Hist90:

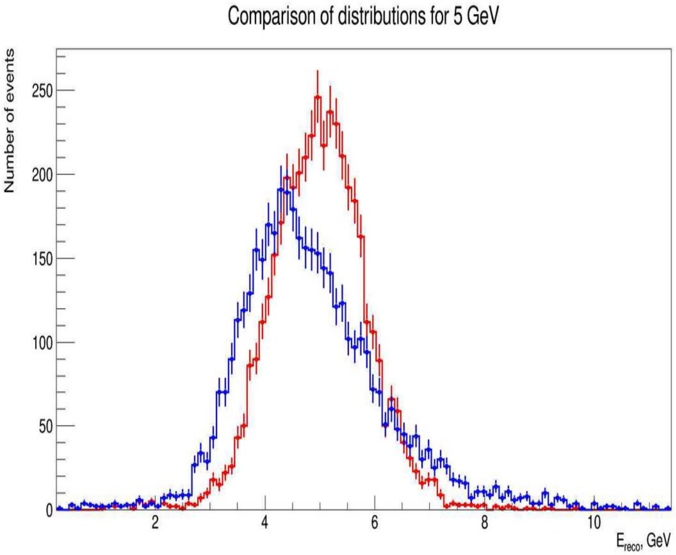
- Find a bin of a mean of the histogram
- Define 90% of the histogram as N_{90} = 0.9 * (hist->GetEntries)
- RMS formula = $$\sqrt{\frac{\sum w\, x^2}{\sum w} - \left(\frac{\sum w\, x}{\sum w}\right)^2},$$

  Sums are calculated by moving symmetrically to the left and to the right bin-by-bin from the mean. The calculation stops when number of events reaches N_{90}
- where x is GetBinCenter (bin of mean plus/minus step of iteration) and w is GetBinContent (bin of mean plus/minus step of iteration)

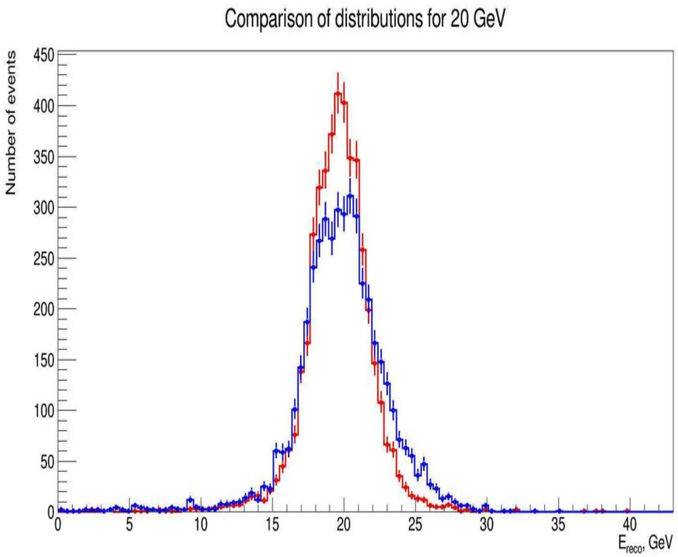
DNN performance for 1, 5, 20 and 40 GeV

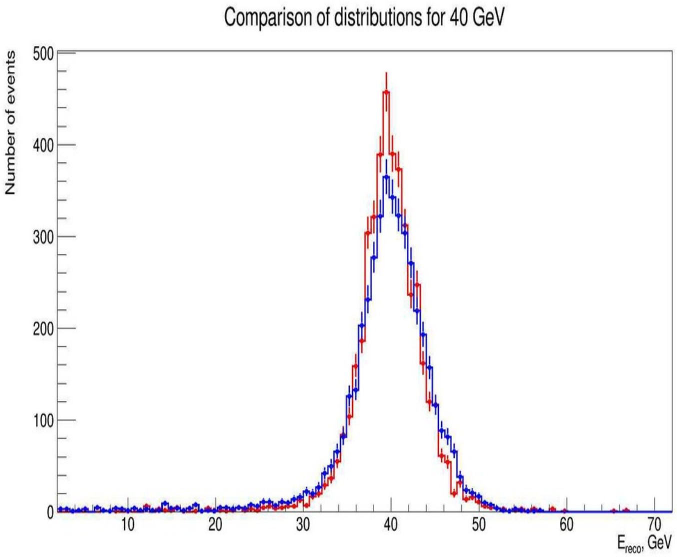


For 5000 sample



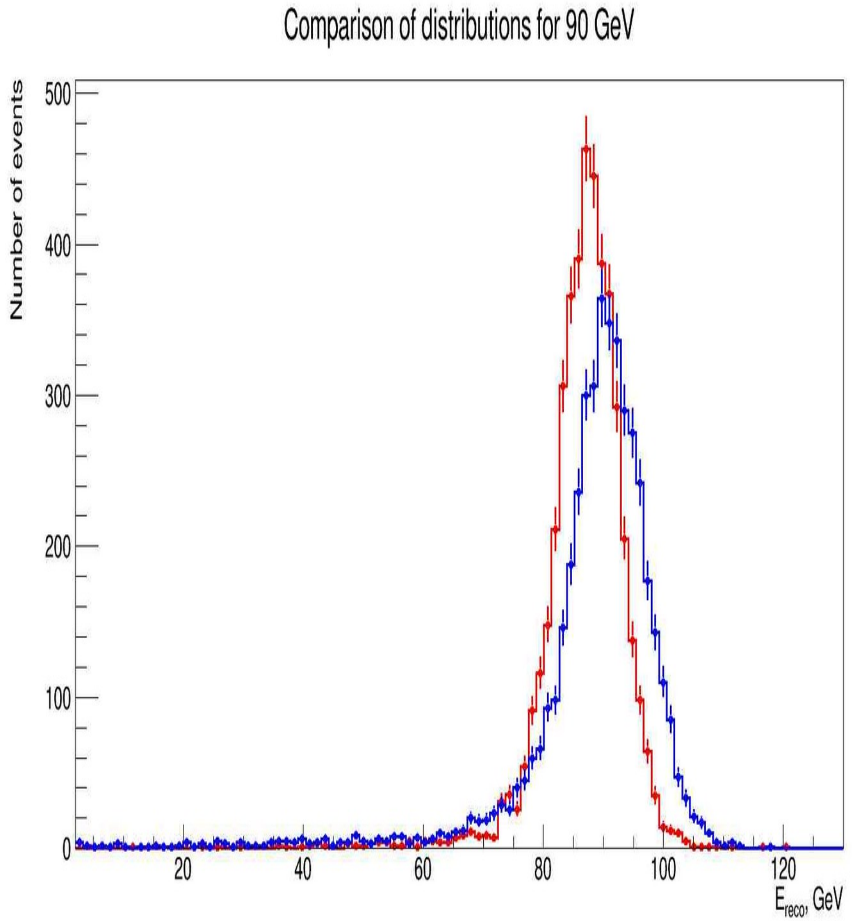
Blue line is without DNN, red line is with DNN. The distribution width is improved. The mean has shifted to the right.

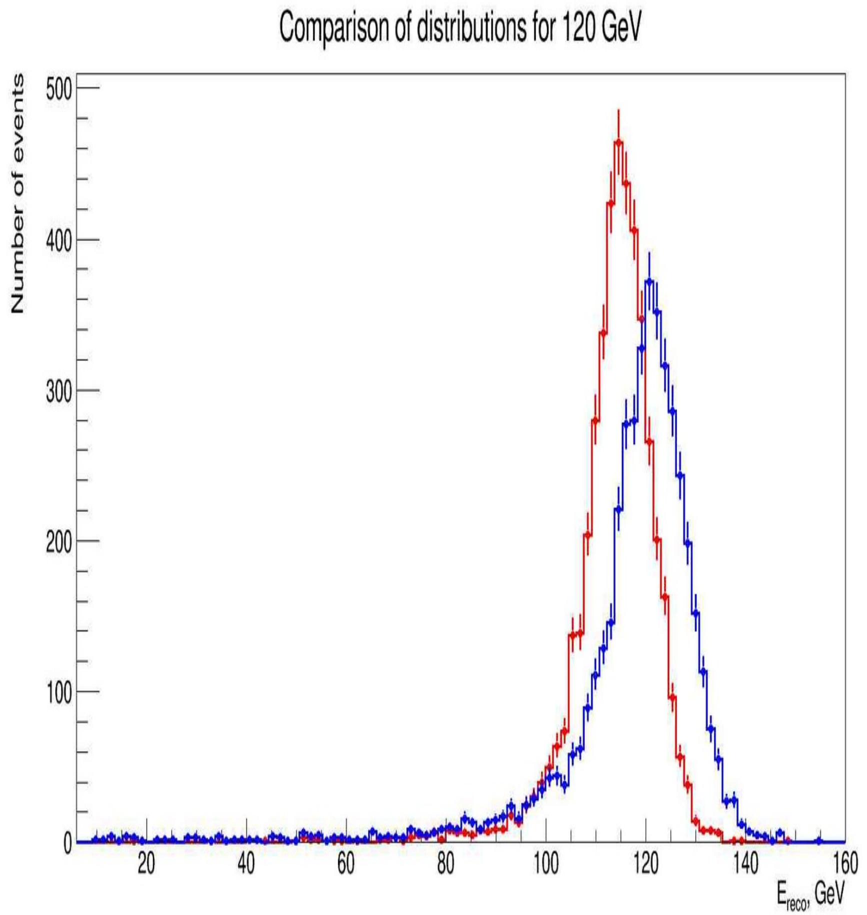




The distribution width is improved. The mean has not shifted.

DNN performance for 90 and 120 GeV

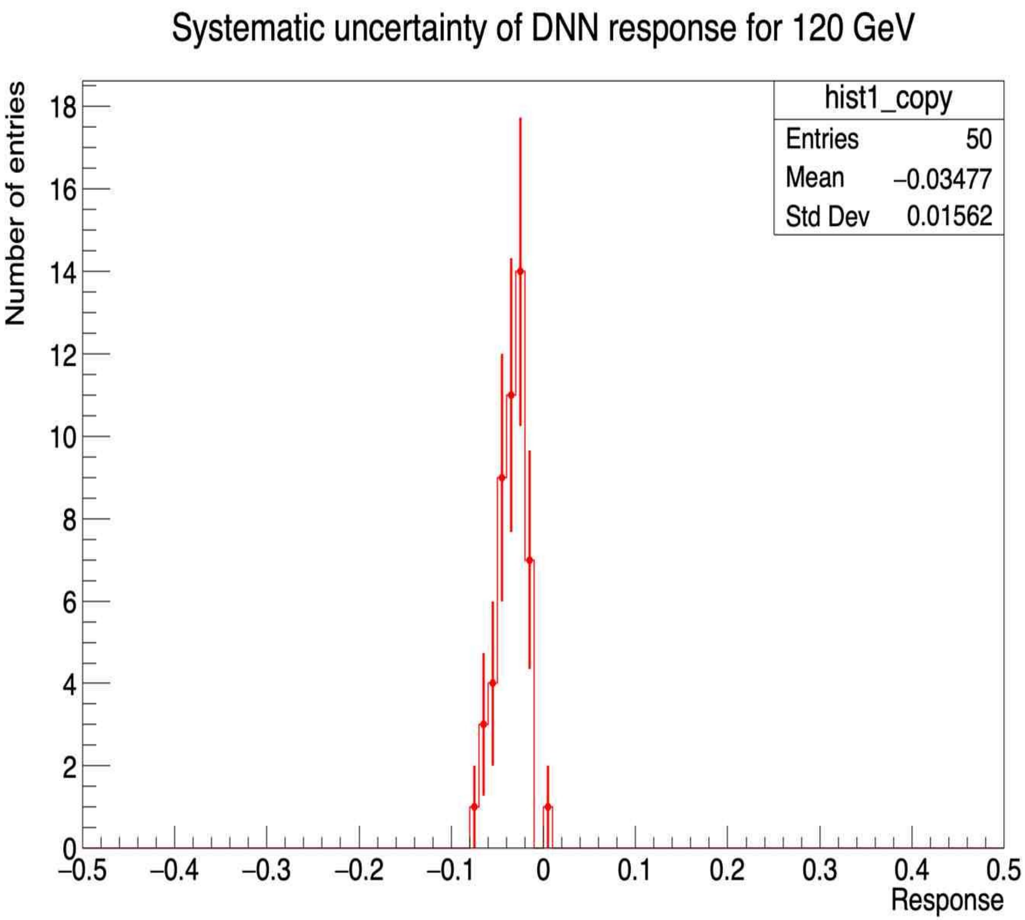




For 5000 sample

Blue line is without DNN, red line is with DNN.
The distribution width is improved. The mean has shifted to the left.

Example of systematic uncertainty



Example histogram for response at 120 GeV

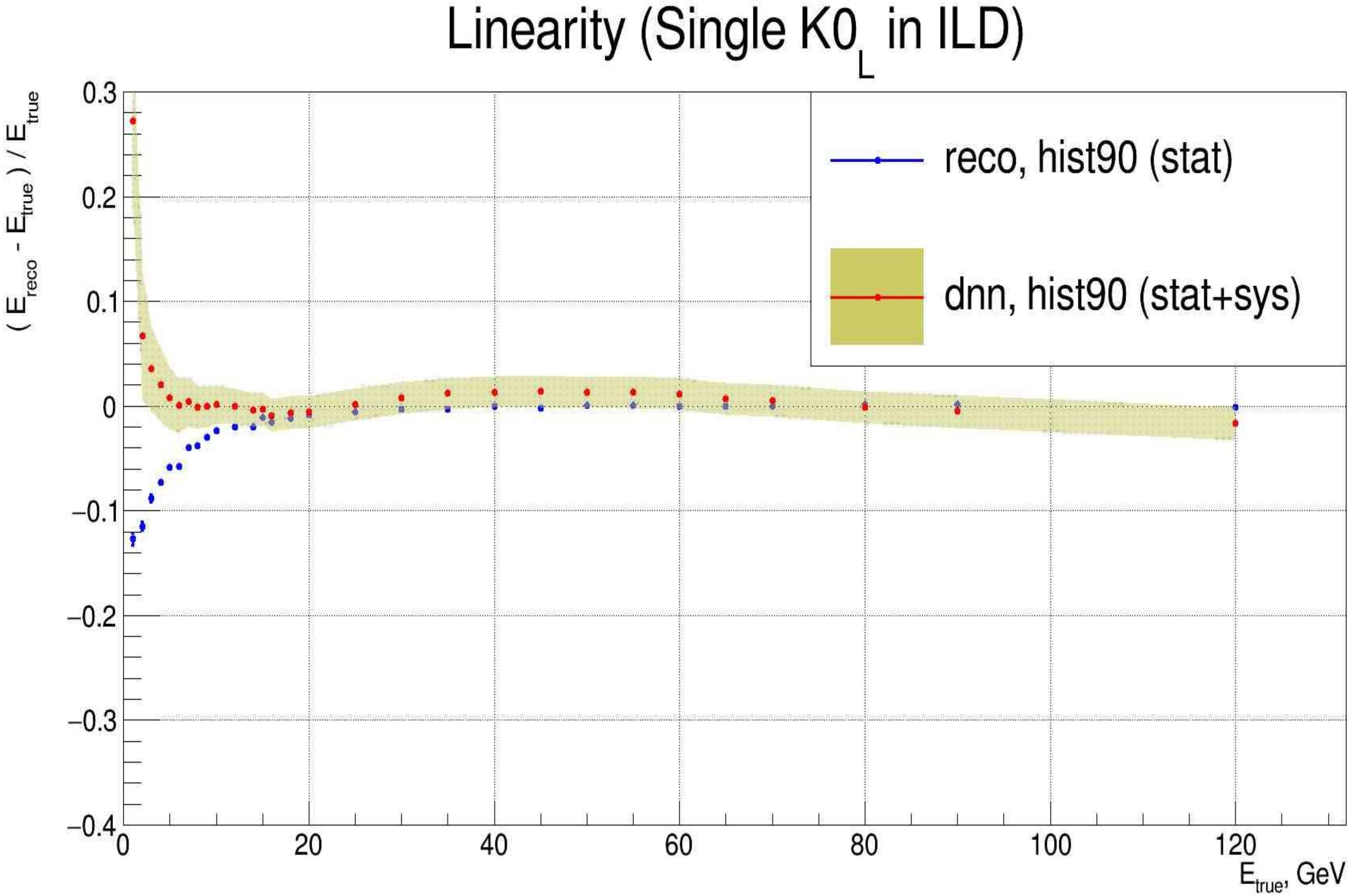
Response is $\frac{(Ereco - Etrue)}{Etrue}$

- 50 runs with the same DNN

- Fluctuations due to random selection of train/validation samples and random initialization of weights of DNN

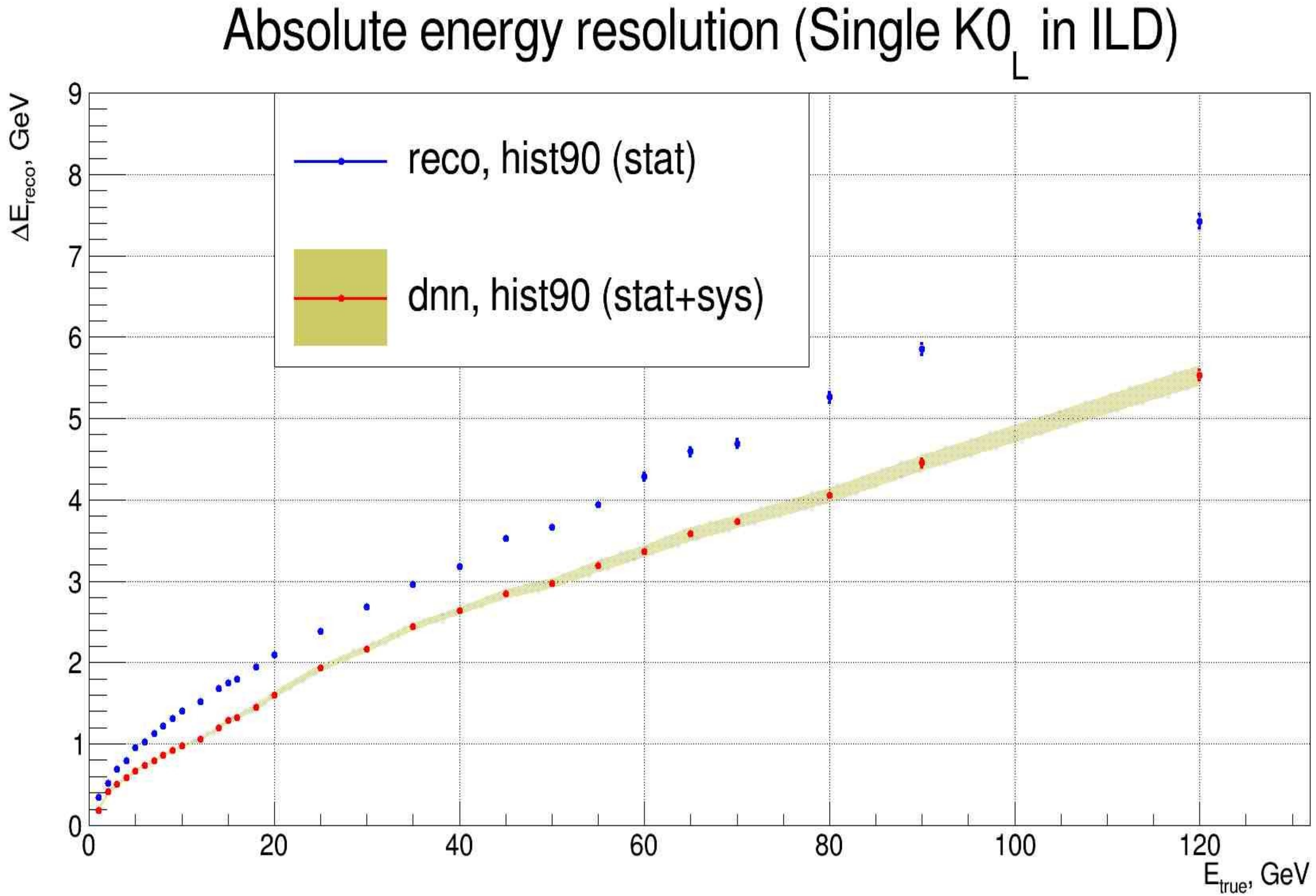
- Response (1)
- Absolute energy resolution (2)
- Relative energy resolution (3)

- 29 energies * 3 = 87 histograms
- Separate uncertainty for each energy point



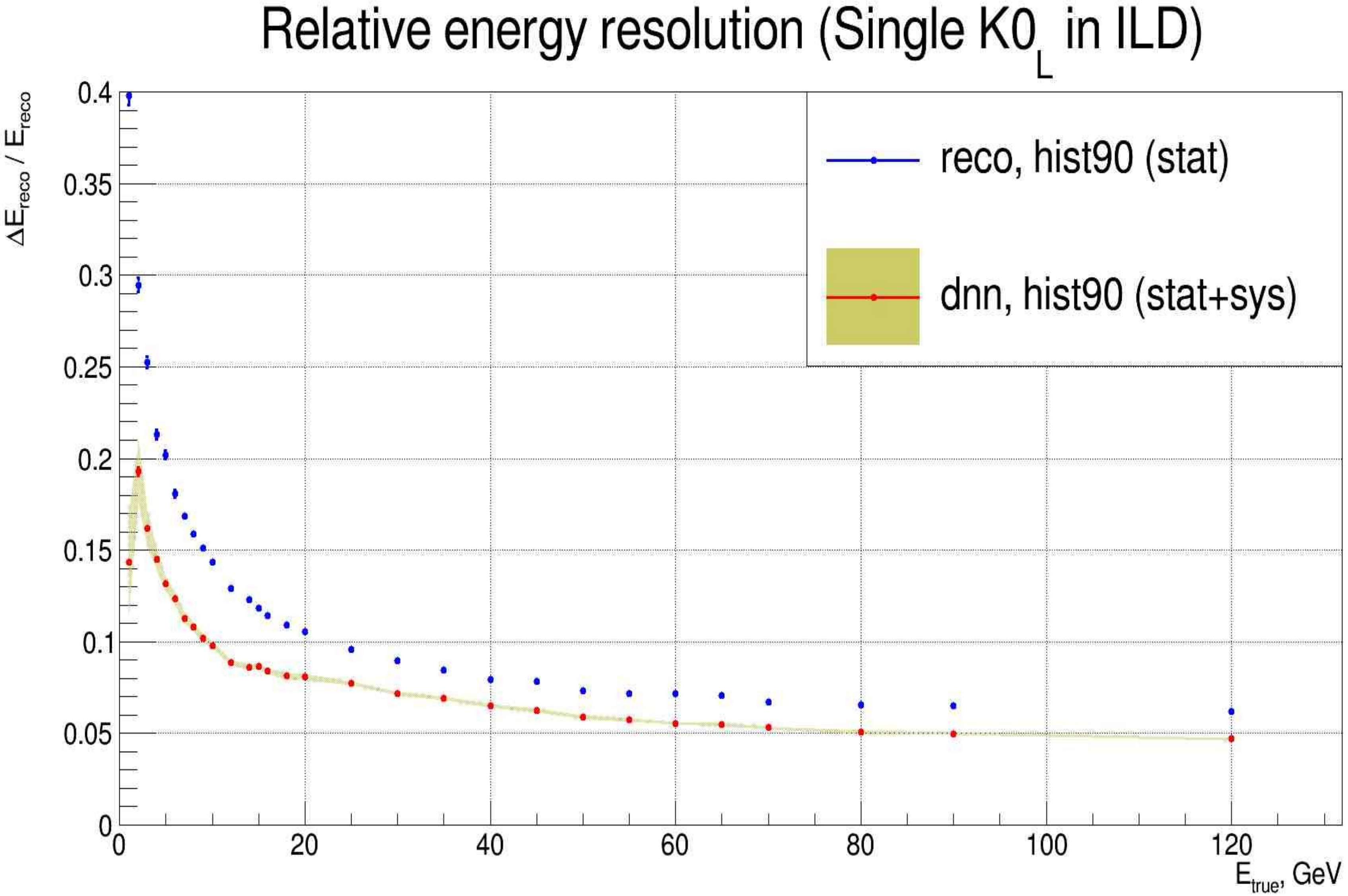
Mean and sigma from hist90 of the energy distributions

The training set didn’t have energies: 7, 45 and 120 GeV. But they are in good agreement with other energy points.



Mean and sigma from hist90 of the energy distributions

The training set didn’t have energies: 7, 45 and 120 GeV. But they are in good agreement with other energy points.



Mean and sigma from hist90 of the energy distributions

The training set didn't have energies: 7, 45 and 120 GeV. But they are in good agreement with other energy points.

Summary and to do

- Hadronic showers from K0L in the range 1-120 GeV are simulated in ILD
- The neural network from Keras package was trained and tested
- The neural network shows noticeable improvement in energy resolution

TO DO

- Implement feature importance (significance)
- Test different parameters and techniques in the DNN
- Try PyTorch software
- Try the global compensation variables from CAN
- Apply the DNN to CALICE AHCAL data

Backup slides

List of collections

EcalBarrelCollectionGapHits
EcalBarrelCollectionRec
EcalEndcapsCollectionGapHits
EcalEndcapRingCollectionRec
EcalEndcapsCollectionRec

HcalBarrelCollectionRec
HcalEndcapsCollectionRec
HcalEndcapRingCollectionRec

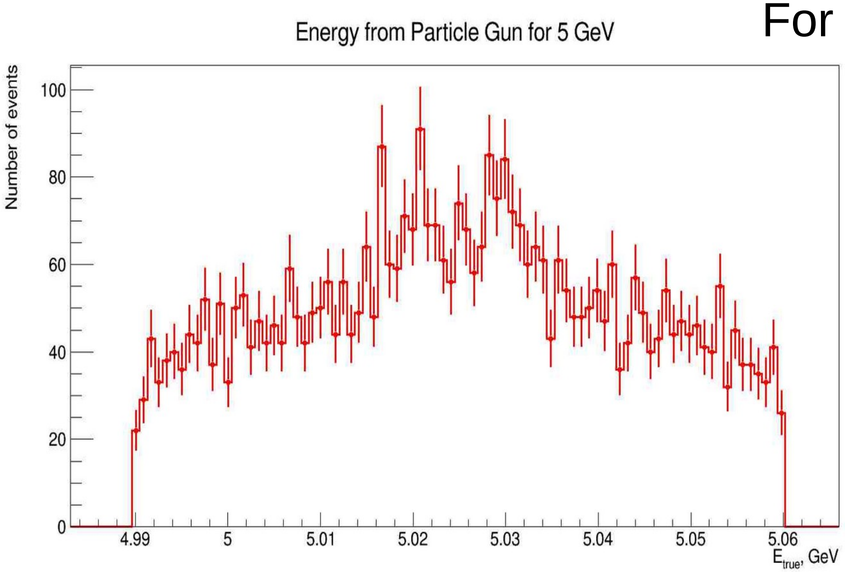
PandoraPFOs

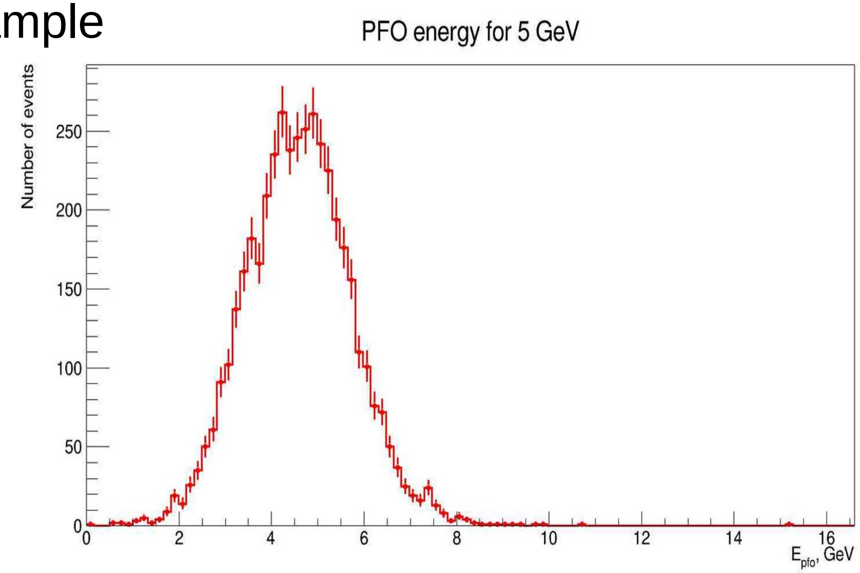
MCParticles

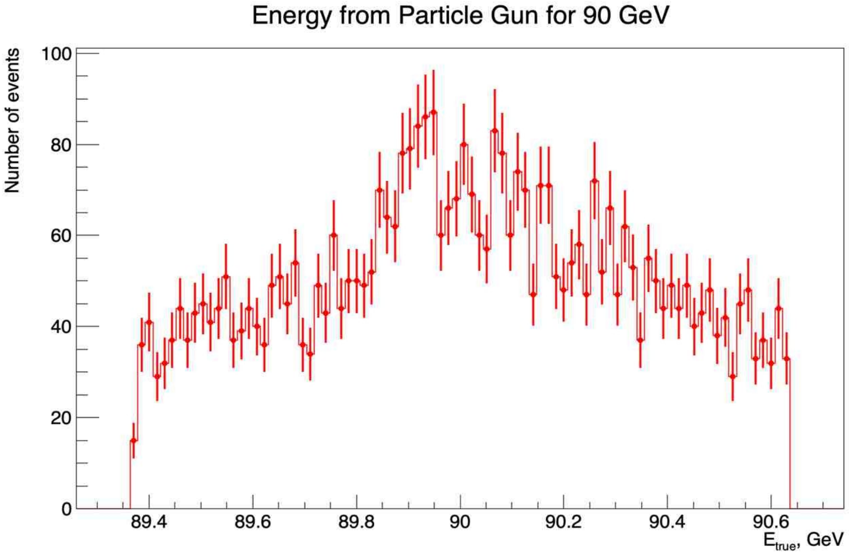
MUON

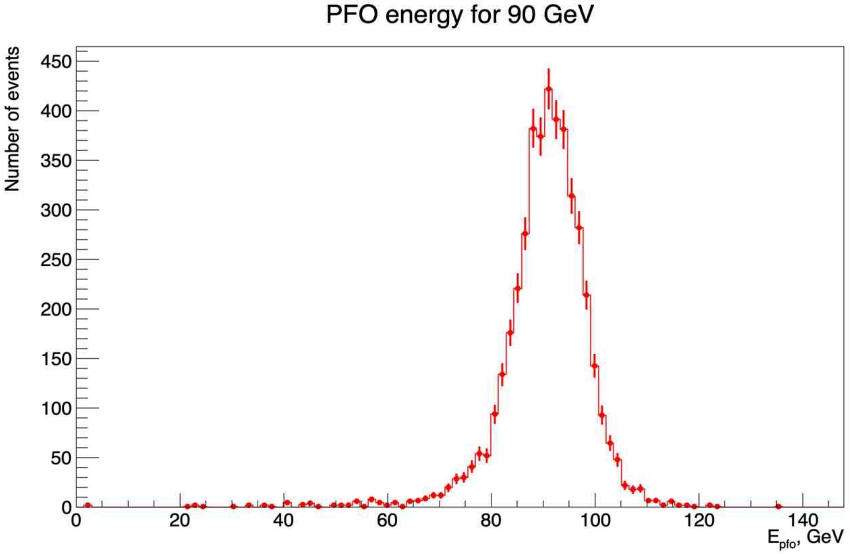
MC and PFO for analysis

For 5000 sample









No clustering, PFO’s only for cross check.

Correlation between target and some variables

	eecal	ehcal	muon_energy	etime	htime	eradius	hradius	elayer	hlayer	muon_time	muon_layer	mc_energy
eecal	1.000000	0.131429	-0.055831	-0.115008	-0.014099	-0.202531	0.027832	0.172889	-0.152756	0.039403	0.028724	0.585834
ehcal	0.131429	1.000000	0.155594	0.121166	-0.257103	0.085121	-0.299339	0.101742	0.327006	0.405211	0.418303	0.852807
muon_energy	-0.055831	0.155594	1.000000	-0.049307	0.014769	-0.007633	-0.068726	-0.120089	0.306266	0.224953	0.334404	0.212033
etime	-0.115008	0.121166	-0.049307	1.000000	0.023430	0.561189	0.016539	0.364091	-0.137274	0.007689	0.005676	0.035826
htime	-0.014099	-0.257103	0.014769	0.023430	1.000000	0.026714	0.553583	-0.085131	-0.065480	-0.067692	-0.072625	-0.194386
eradius	-0.202531	0.085121	-0.007633	0.561189	0.026714	1.000000	0.163117	0.041665	0.008397	0.018068	0.019396	-0.030379
hradius	0.027832	-0.299339	-0.068726	0.016539	0.553583	0.163117	1.000000	-0.052394	-0.166443	-0.141630	-0.150124	-0.222829
elayer	0.172889	0.101742	-0.120089	0.364091	-0.085131	0.041665	-0.052394	1.000000	-0.323224	-0.054740	-0.062230	0.144776
hlayer	-0.152756	0.327006	0.306266	-0.137274	-0.065480	0.008397	-0.166443	-0.323224	1.000000	0.336597	0.390446	0.229130
muon_time	0.039403	0.405211	0.224953	0.007689	-0.067692	0.018068	-0.141630	-0.054740	0.336597	1.000000	0.769852	0.375865
muon_layer	0.028724	0.418303	0.334404	0.005676	-0.072625	0.019396	-0.150124	-0.062230	0.390446	0.769852	1.000000	0.400483
mc_energy	0.585834	0.852807	0.212033	0.035826	-0.194386	-0.030379	-0.222829	0.144776	0.229130	0.375865	0.400483	1.000000

Techniques of resolution estimate

- **Hist**: mean and rms of the full distribution
- **Hist90**: mean and rms of the 90% of the full distribution
- **Hist95**: mean and rms of the 95% of the full distribution
- **Fit**: mean and sigma of Gaussian fit of the full distribution

- Legend: E_{reco} is mean from fit or histogram, ΔE_{reco} is sigma from fit or rms from histogram and E_{true} is energy from generator

  For hist method standard ROOT procedure are used

Hist90 code example

It depends how you define the central 90% (model dependent).
Below a brute force example.

Rene

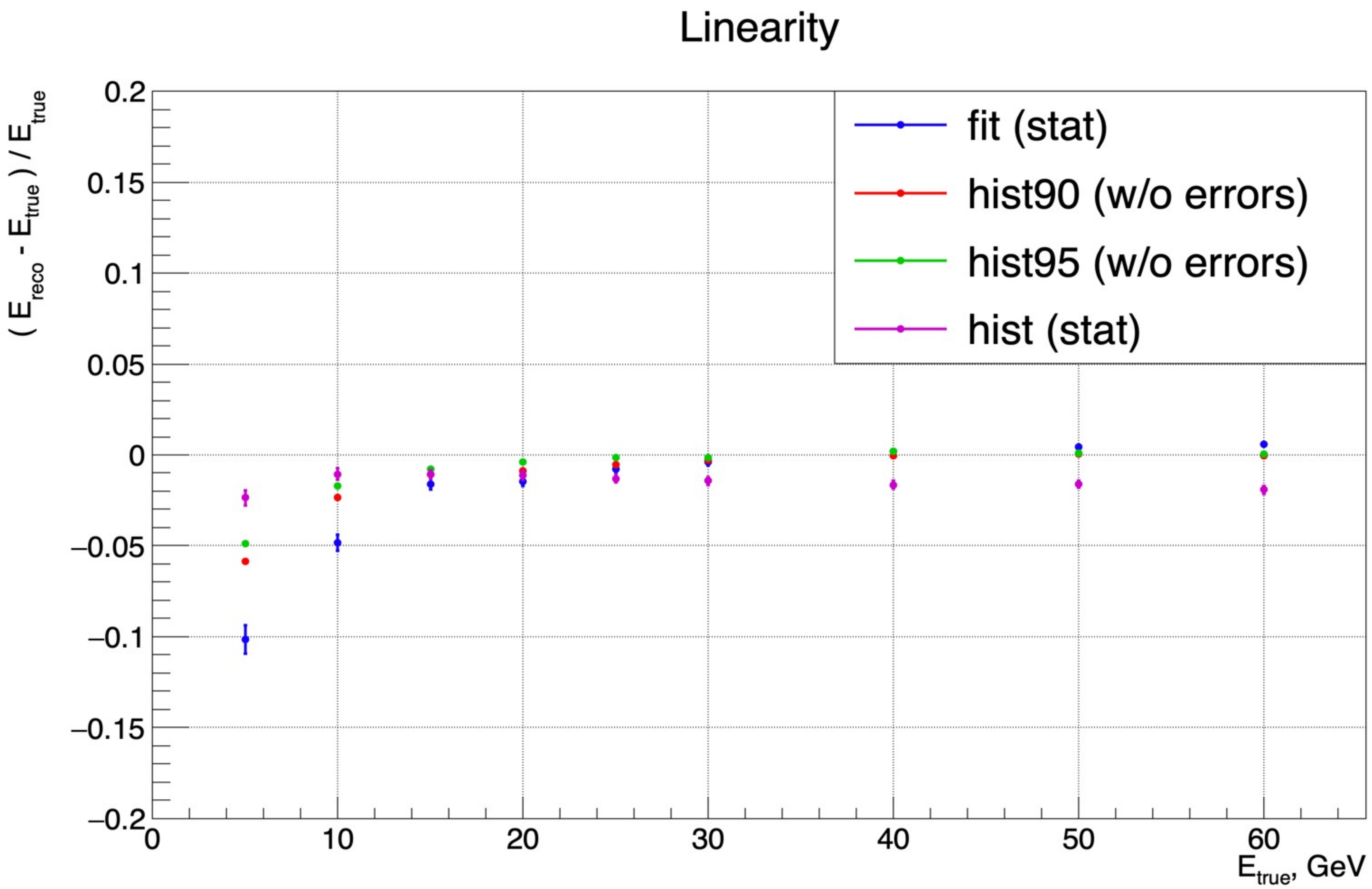
```
[code]void rms90(TH1 h) {
TAxis axis = h->GetXaxis();
Int_t nbins = axis->GetNbins();
Int_t imean = axis->FindBin(h->GetMean());
Double_t entries =0.9h->GetEntries();
Double_t w = h->GetBinContent(imean);
Double_t x = h->GetBinCenter(imean);
Double_t sumw = w;
Double_t sumwx = wx;
Double_t sumwx2 = wxx;
for (Int_t i=1;i<nbins;i++) {
if (i> 0) {
w = h->GetBinContent(imean-i);
x = h->GetBinCenter(imean-i);
sumw += w;
sumwx += wx;
sumwx2 += wxx;
}
if (i<= nbins) {
w = h->GetBinContent(imean+i);
x = h->GetBinCenter(imean+i);
sumw += w;
sumwx += wx;
sumwx2 += wxx;
}
if (sumw > entries) break;
}
x = sumwx/sumw;
Double_t rms2 = TMath::Abs(sumwx2/sumw -x*x);
Double_t result = TMath::Sqrt(rms2);
printf("RMS of central 90% = %g, RMS total = %g\n",result,h->GetRMS());
}

void central90() {
TH1F *h = new TH1F("h","test",100,-4,2);
h->FillRandom("gaus",10000);
rms90(h);
}[/code]
```

The code is from:
https://root-forum.cern.ch/t/rms90

Fit procedure

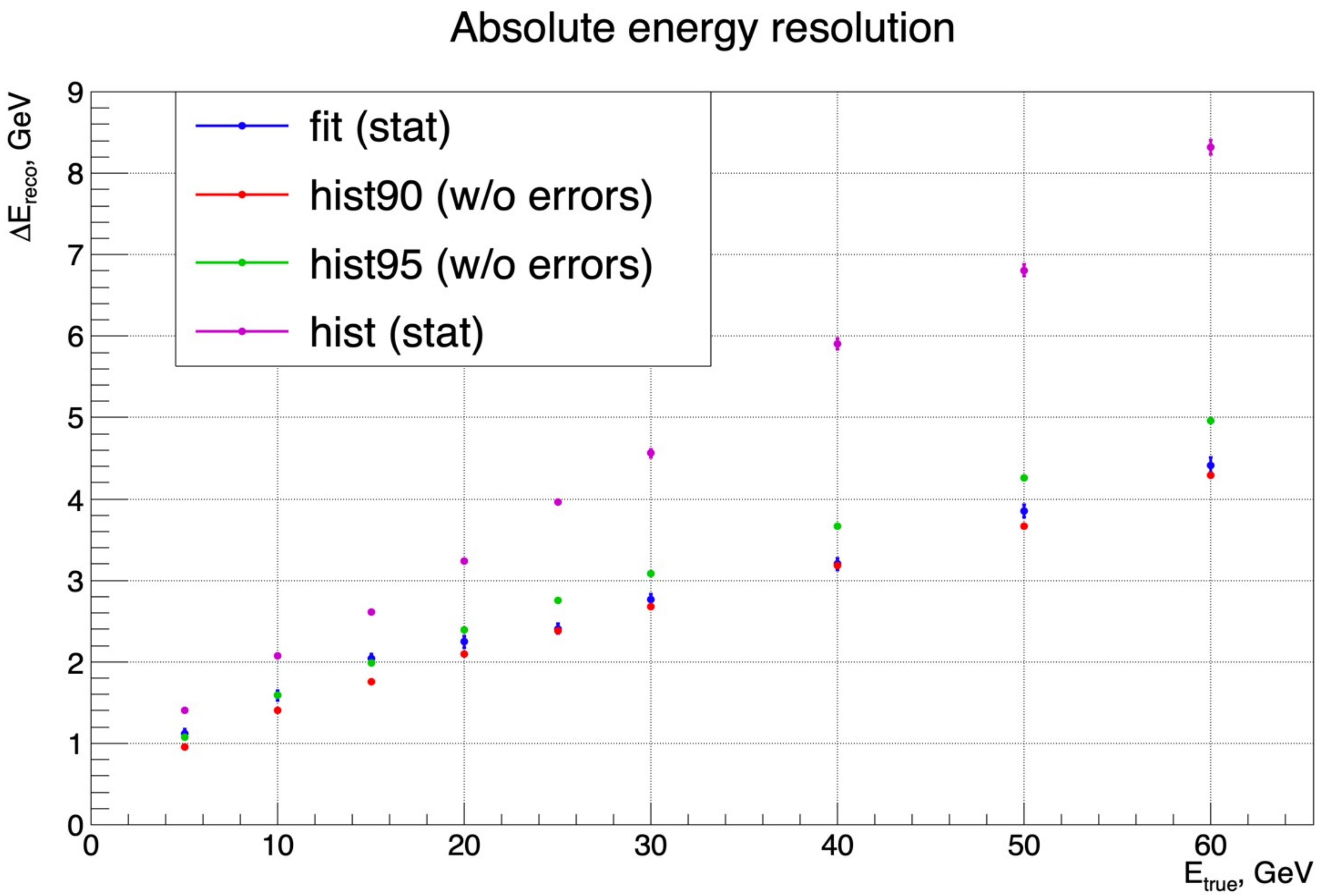
- Take mean and rms from full histogram
- Perform Gaussian fit in [mean ± (range * rms)], where range is values from 1.0 to 2.5 => array of means and sigmas from Gauss fits
- Fit is accepted if (chi-square / NDF) < 1.5
- Final fit result is that with minimum (chi-square / NDF)

Linearity



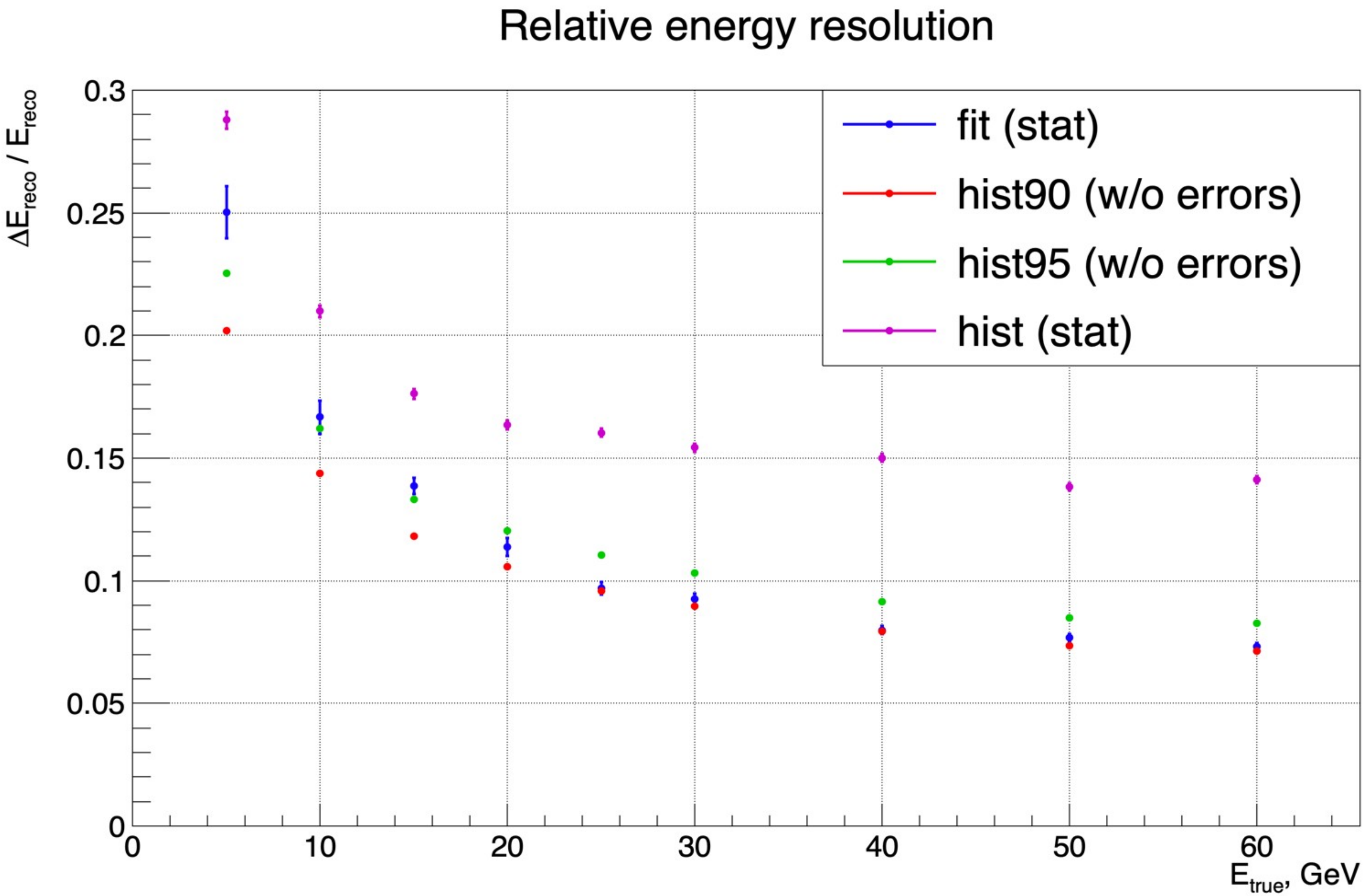
Good coincidence of fit, hist90 and hist95 above 15 GeV.
The worst linearity for fit (in agreement with physics).

Absolute energy resolution



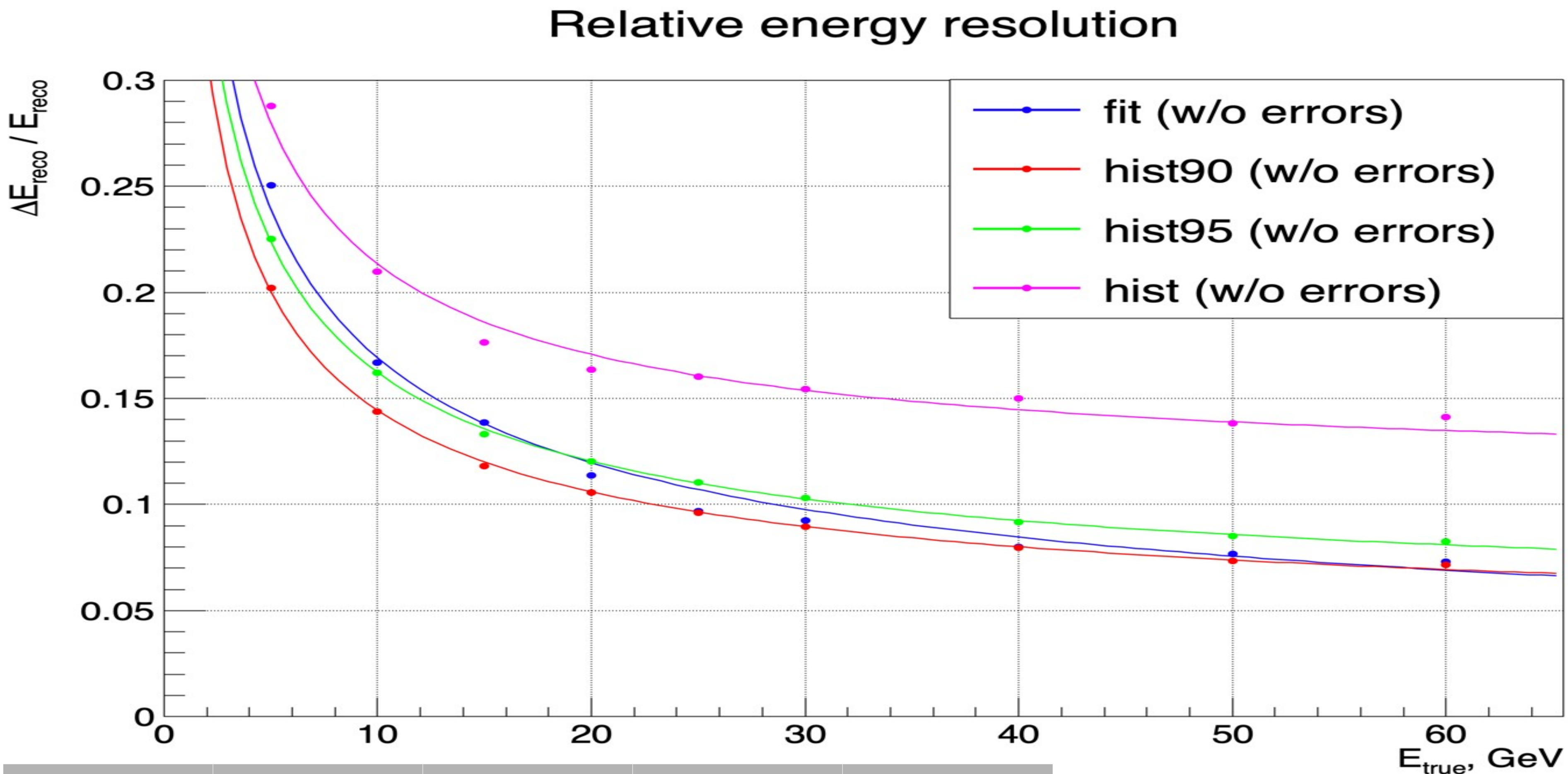
Fit and hist90 look similar.
Hist95 in agreement with fit and hist90 before 20 GeV.

Relative energy resolution



The better resolution is for hist90.

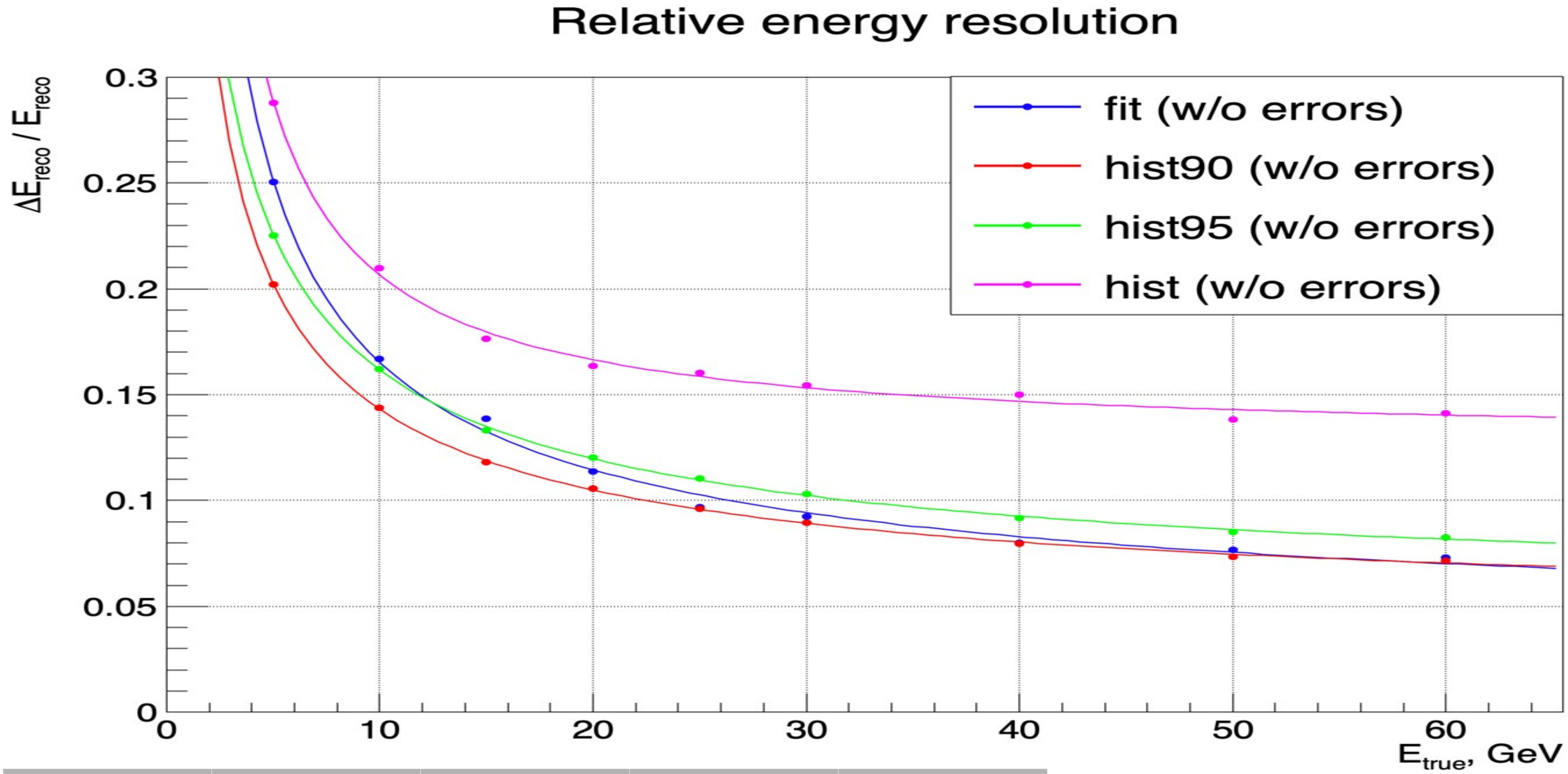
Fit for relative energy resolution



	fit	hist90	hist95	hist
A, sqrt(GeV)	0.53	0.44	0.49	0.57
B, %	0.0	4.0	5.1	11.3
C, GeV	0.0	0.0	0.0	0.0

$$\sqrt{\left(\frac{A}{\sqrt{E}}\right)^2 + B^2}$$

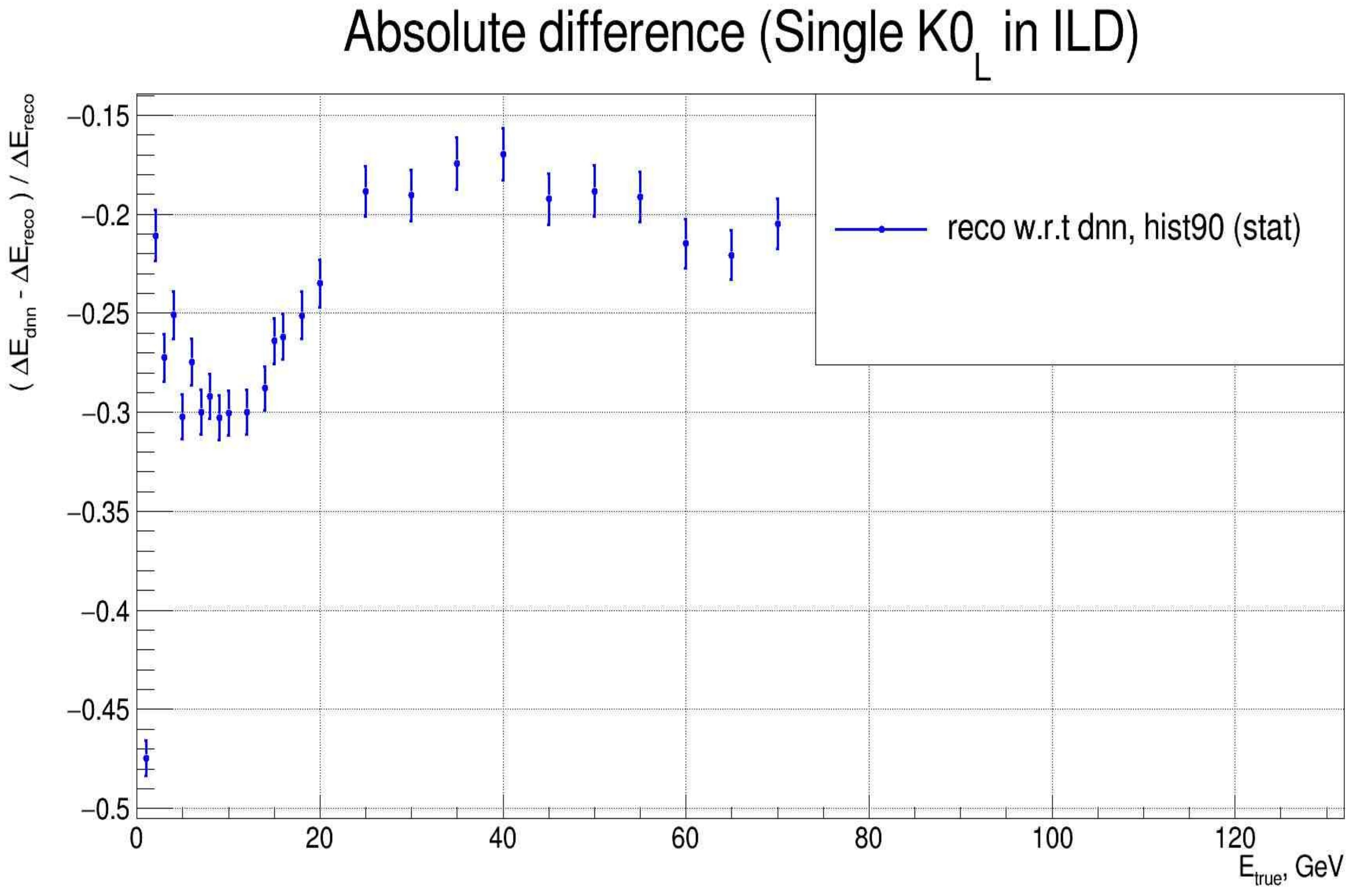
Fit for relative energy resolution



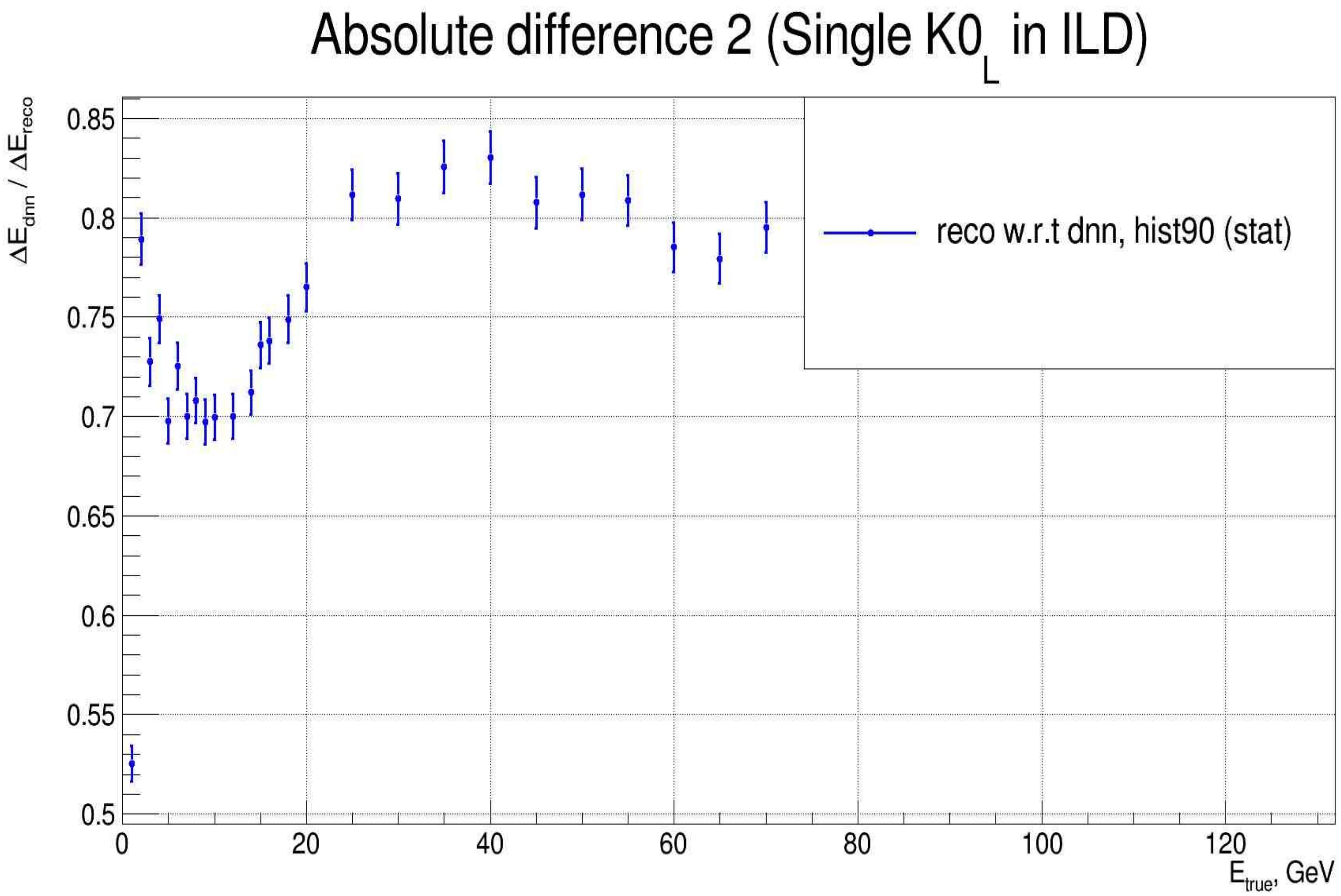
	fit	hist90	hist95	hist
A, sqrt(GeV)	0.46	0.42	0.47	0.44
B, %	3.5	4.5	5.4	12.8
C, GeV	0.7	0.3	0.3	0.8

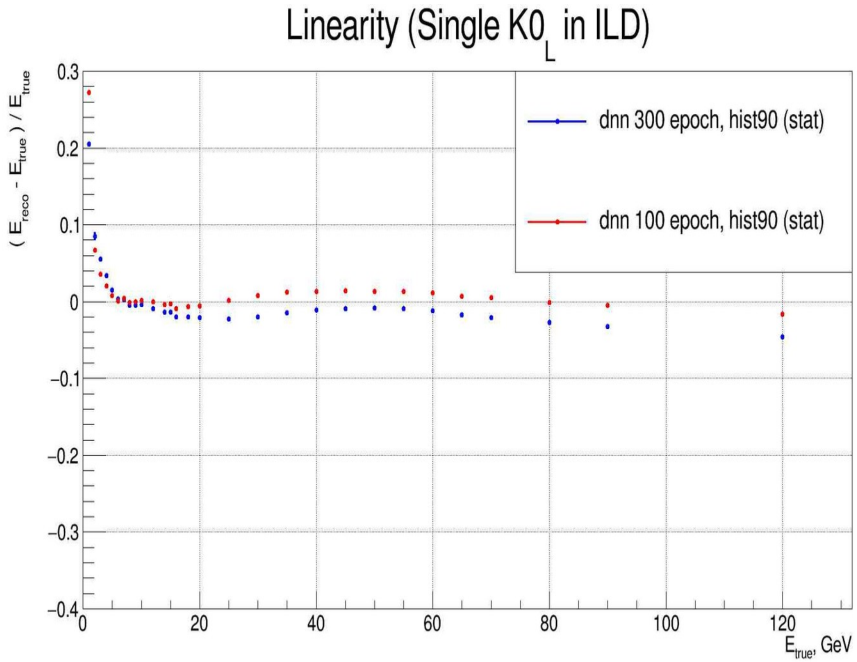
$$\sqrt{\left(\frac{A}{\sqrt{E}}\right)^2 + B^2 + \left(\frac{C}{E}\right)^2}$$

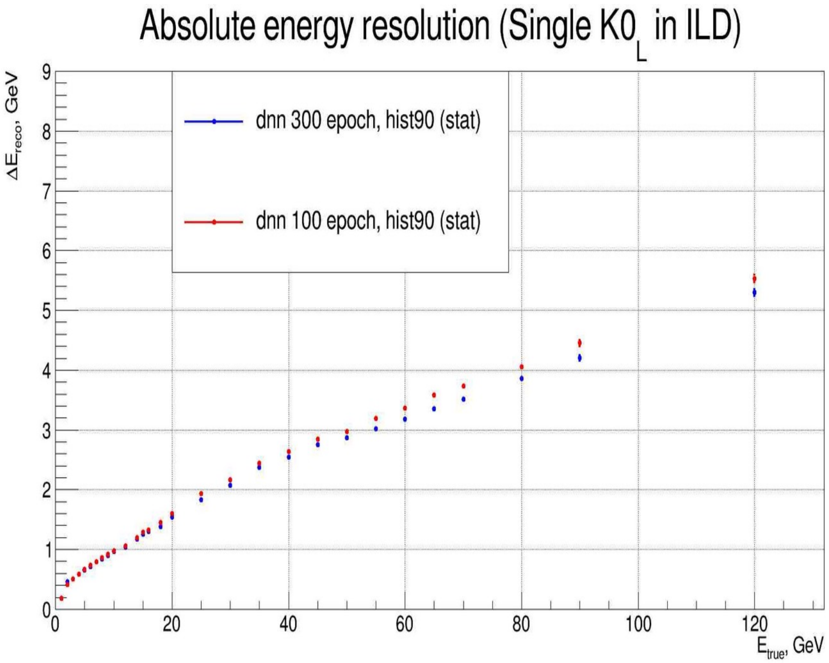
Absolute difference between RECO and DNN

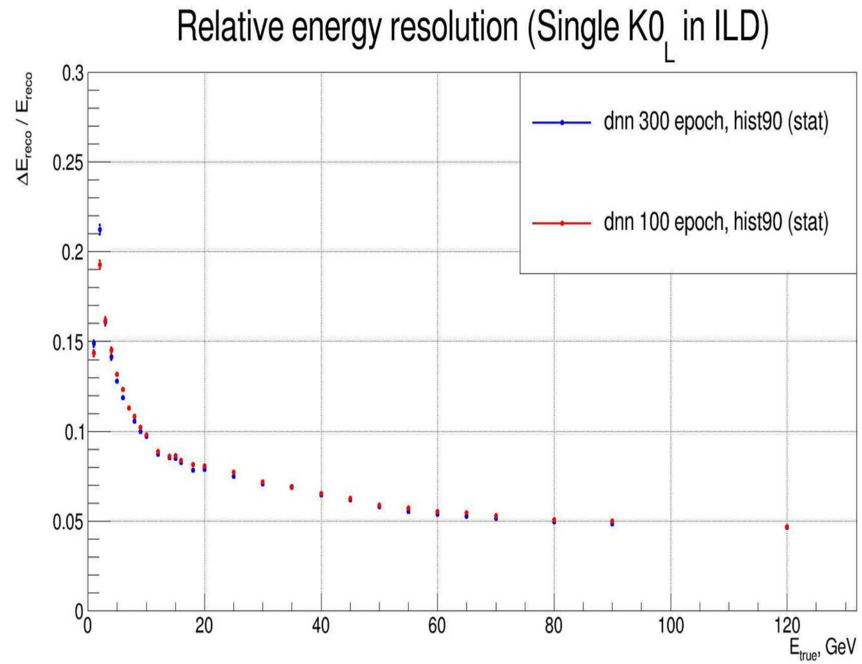


Absolute difference between RECO and DNN




Linearity (Single K0_L in ILD)
(E_reco - E_true) / E_true
dnn 300 epoch, hist90 (stat)
dnn 100 epoch, hist90 (stat)
E_true, GeV


Absolute energy resolution (Single K0_L in ILD)
ΔE_reco, GeV
dnn 300 epoch, hist90 (stat)
dnn 100 epoch, hist90 (stat)
E_true, GeV


Relative energy resolution (Single K0_L in ILD)
ΔE_reco / E_reco
dnn 300 epoch, hist90 (stat)
dnn 100 epoch, hist90 (stat)
E_true, GeV