



BeamCal reconstruction

High Level Reconstruction Week, DESY
8 July 2015

André Sailer, Andrey Sapronov

Overview

- BeamCalReco processor for Marlin
- Reconstruction algorithms
- Background simulation
- How-to's
- Validation and comparison
- Summary

BeamCalReco Marlin processor

- Main purpose is tagging high energy electrons in BeamCal
- Implements two conceptually different algorithms: Clustering and Shower fitting. Users can implement their own clustering algorithms
- and three methods of beam-induced background simulation: pregenerated, parametrised and averaged.
- The steerable parameters are accessible in xml config
- Geometry information can be read from Gear or soon DD4hep
- ‘Event display’ for easy inspection and debugging
- Efficiency and resolution histograms on the output. Can potentially be used for fast simulation/smearing
- Part of ILCSSoft since v01-17-07

BeamCal Electron Clustering Algorithm



General Idea of Clustering

- 1 Energy in the BeamCal Pads is

$$\vec{E}_{\text{Total}}^{\text{Event}} = \vec{E}_{\text{Signal}}^{\text{Event}} + \vec{E}_{\text{BKG}}^{\text{Event}}$$

- 2 Subtract the average background plus one standard deviation for each cell

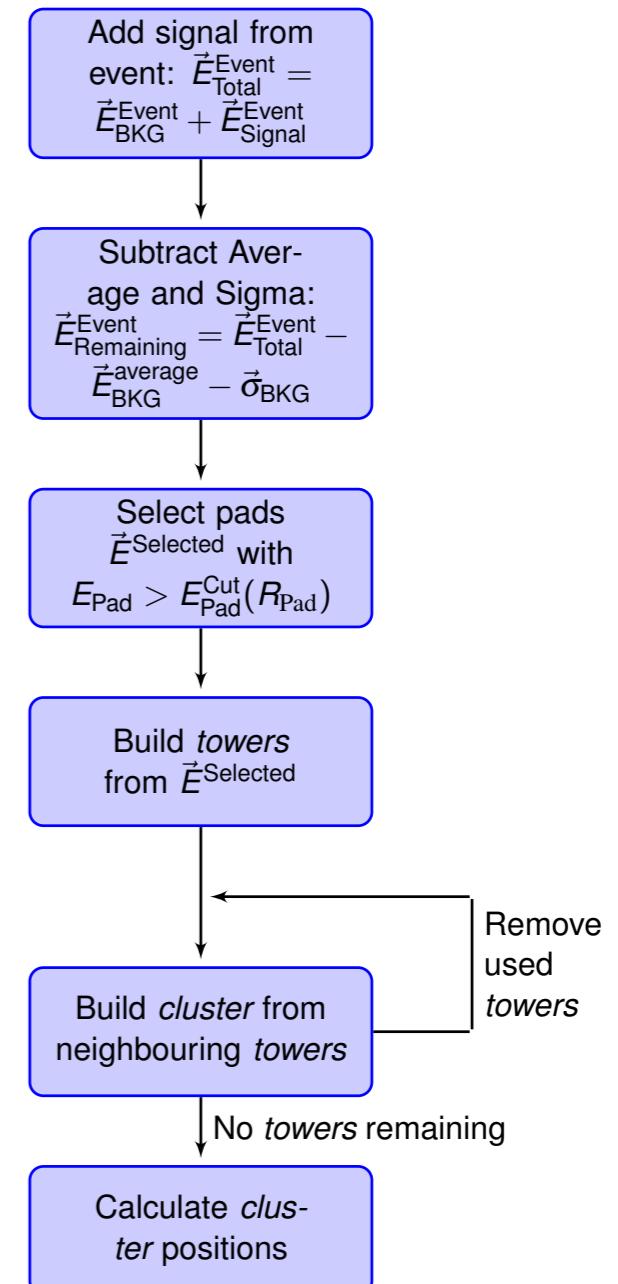
$$\vec{E}_{\text{Remaining}}^{\text{Event}} = \vec{E}_{\text{Total}}^{\text{Event}} - (\vec{E}_{\text{BKG}}^{\text{average}} + \vec{\sigma}_{\text{BKG}})$$

- 3 Select the pads with sufficiently large remaining energy, e.g., $E_{\text{Remaining}}^i > \sigma_{\text{BKG}}^i$

- 4 Find *towers* in the selected pads: Pads with the same position in different layers

- 5 Merge neighbouring *towers* into *cluster*

- 6 Calculate *cluster* position



Shower fitting method

- As usually, the signal energy distribution is overlaid over prepared background sample
- A simple χ_p^2 is calculated along the tower:

$$\chi_p^2 = \sum_{\text{pads}} (E_{\text{sig}} - E_{\text{bg}})^2 / \sigma_{\text{bg}}$$

- The energies are projected along calorimeter axis to a single layer, so that each pad contains energies of all the pads of the tower.
- A **central shower pad** is selected:
 - by maximum of this χ_p^2 (gives highest deviation)
 - χ_p^2/ndf is higher than config parameter `TowerChi2ndfLimit`
 - energy is over $0.7 * \text{ETCluster}$ (shower energy threshold)

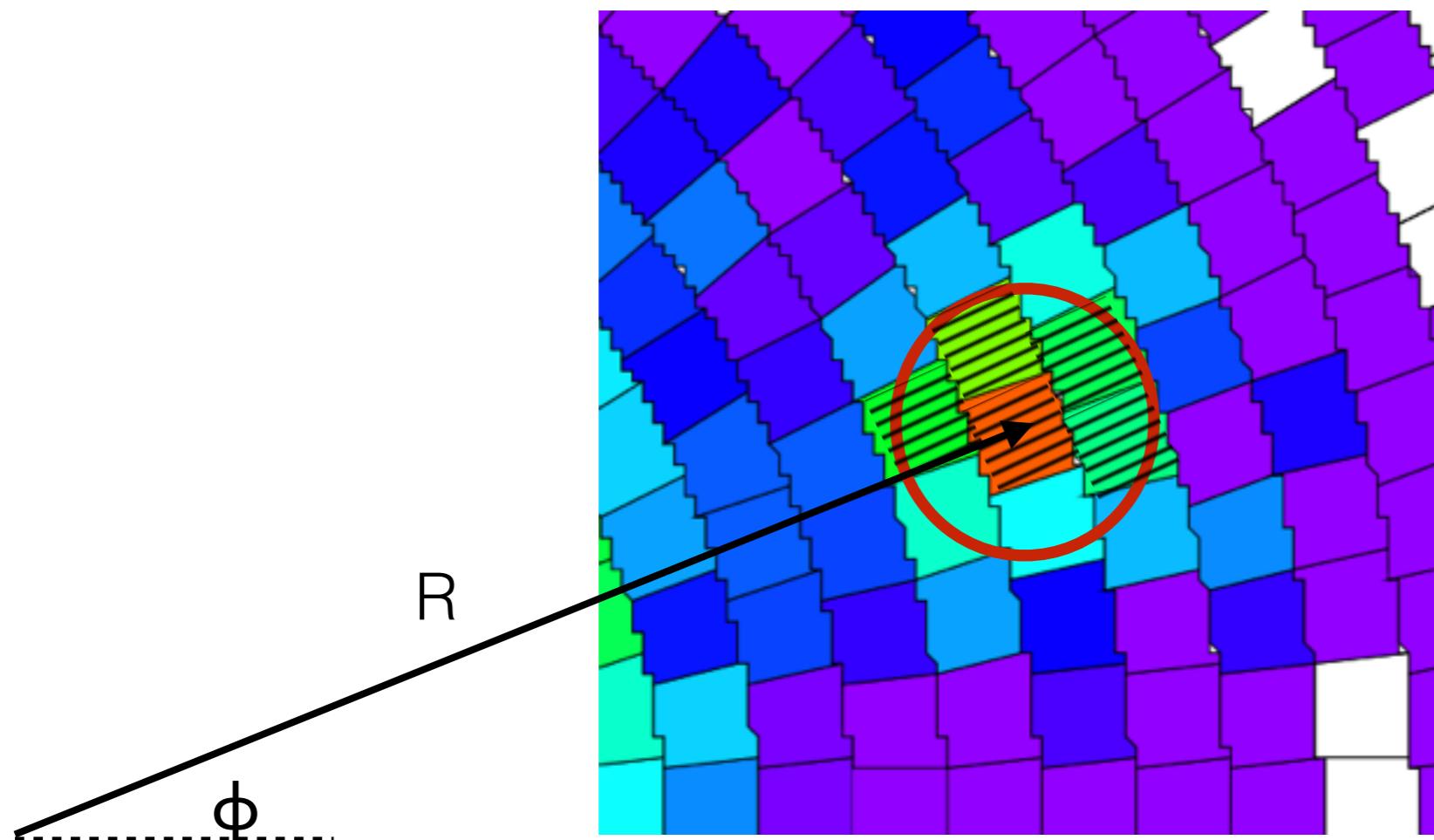
Shower fitting method (contd.)

- Around the central pad, select others to form a ‘spot’:
 - distance $< 2R_{\text{mol}}$
 - energy above $0.1^*(\text{lower shower energy threshold})$
 - energy above σ_{tow} (st.dev. of the projected background energy)
- Assume that the shower has gaussian shape and its centre is in the central pad and estimate the energies in the spot pads by Simpson integration -> $E_{\text{int.}}$.
- Calculate χ_s^2 of the spot pads

$$\chi_s^2 = \sum_{\text{spot}} (E_{\text{int}} - E_{\text{dep}})^2 / \sigma_{\text{tow}}^2$$

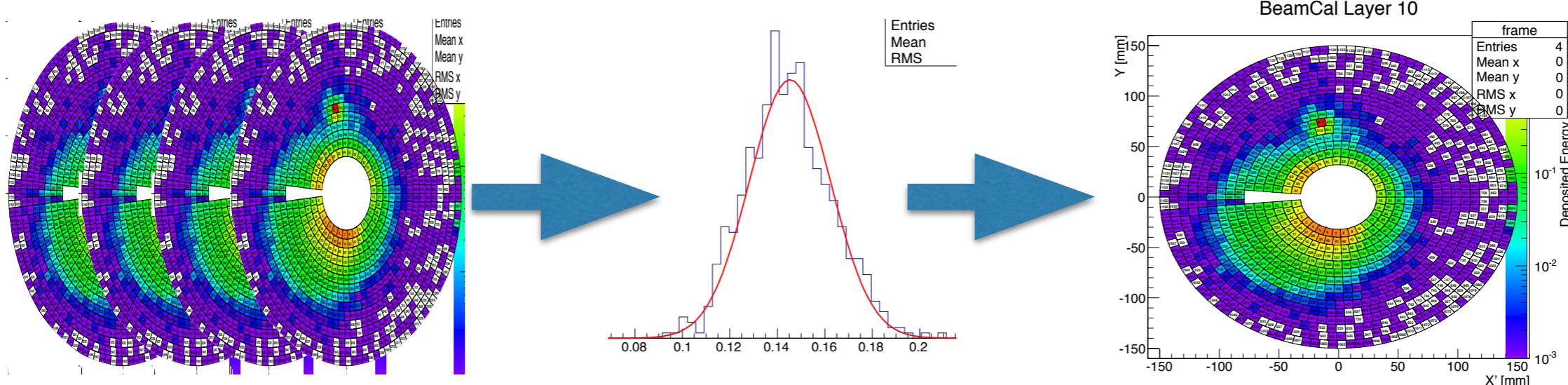
Shower fitting method (contd.)

- Perform minimisation of the χ^2_s with MINUIT over shower centre R and ϕ , and two gaussian parameters.
- If the fit is good, shower energy is high enough, select it as an electron candidate.
- Otherwise search another spot and iterate.



Background simulation methods

- ‘Monte Carlo’ method with *pregenerated* background samples. For each event BG is arbitrary composed from pool. There is a dedicated tool to create the samples
- ‘Average’ method, same as in earlier reconstruction. Uses `bg_aver...` files
- *Parametrised* method (almost same as average). The BG is created by random generation according to distribution in each pad. Also dedicated tool available to compute distributions.



How-to run 1: Input/Output

```
<processor name="MyBeamCalClusterReco" type="BeamCalClusterReco">

    <!--Name of BeamCal Collection-->
    <parameter name="BeamCalCollectionName" type="string"
        lcioInType="SimCalorimeterHit"> BCAL </parameter>

    <!--Name of the MCParticle Collection, only needed and used to estimate
        efficiencies-->
    <parameter name="MCParticleCollectionName" type="string"
        lcioInType="MCParticle">MCParticle </parameter>

    <!--Name of the Reconstructed Cluster collection-->
    <parameter name="RecoClusterCollectionname" type="string"
        lcioOutType="Cluster">BeamCalClusters </parameter>

    <!--Name of the Reconstructed Particle collection-->
    <parameter name="RecoParticleCollectionname" type="string"
        lcioOutType="ReconstructedParticle">
        BeamCalRecoParticle </parameter>
```

How-to run 2: Background method

```
<!--How to estimate background [Parametrised, Pregenerated, Averaged]-->
<parameter name="BackgroundMethod" type="string"> Averaged </parameter>

<!--Root Inputfile(s)-->
<parameter name="InputFileBackgrounds" type="StringVec">
    bg_aver.sv01-14-01-p00_fieldX02.mILD_o1_v05.E250-TDR_ws.PBeamstr-
    pairs.I270000.root
</parameter>

<!--Number of Bunch Crossings of Background per event-->
<parameter name="NumberOfBX" type="int"> 1 </parameter>
```

How-to run 3: pad selection/shower fitting

```
<!--Use the cuts for the pads specified in ETPad, if false, the variance in  
each pad is used times SigmaPad Factor. If false the first entry in ETPad  
is used as a minimum energy to consider a pad at all--><parameter name="UseConstPadCuts" type="bool"> False </parameter>  
  
<!--Use Chi2 selection criteria to detect high energy electron in the  
signal.--><parameter name="UseChi2Selection" type="bool">false </parameter>  
  
<!--Limit on square norm of tower energy chi2/ndf, where  
chi2 = (E_dep - E_bg)^2/sig^2. Reasonable value for pregenerated bkg is 5.,  
for parametrised is 2. --><parameter name="TowerChi2ndfLimit" type="double"> 2.0 </parameter>  
  
<!--If not using ConstPadCuts, each pad SigmaCut*variance is considered for  
clusters--><parameter name="SigmaCut" type="double"> 1.0 </parameter>  
  
<!--Layer (inclusive) from which on we start looking for signal clusters--><parameter name="StartLookingInLayer" type="int"> 1 </parameter>  
  
<!--Rings from which onwards the outside Thresholds are used--><parameter name="StartingRing" type="FloatVec"> 0 1 </parameter>  
  
<!--Energy in a Pad, after subtraction of background required to consider it  
for signal--><parameter name="ETPad" type="FloatVec"> 0.005 0.3 </parameter>
```

How-to run 4: energy reconstruction

```
<!--Energy in a Cluster to consider it an electron-->
<parameter name="ETCluster" type="FloatVec"> 0.5 0.5 </parameter>

<!--Minimum number of (non-consecutive) pads in a single tower to be considered
for signal-->
<parameter name="MinimumTowerSize" type="int"> 6 </parameter>

<!-- Calibration factor to account for sampling fraction -->
<parameter name="LinearCalibrationFactor" type="double"> 72.76 </parameter>
```

How-to run 5: inspection and debugging

```
<!--Flag to create the TEfficiency for fast tagging library-->
<parameter name="CreateEfficiencyFile" type="bool"> true </parameter>

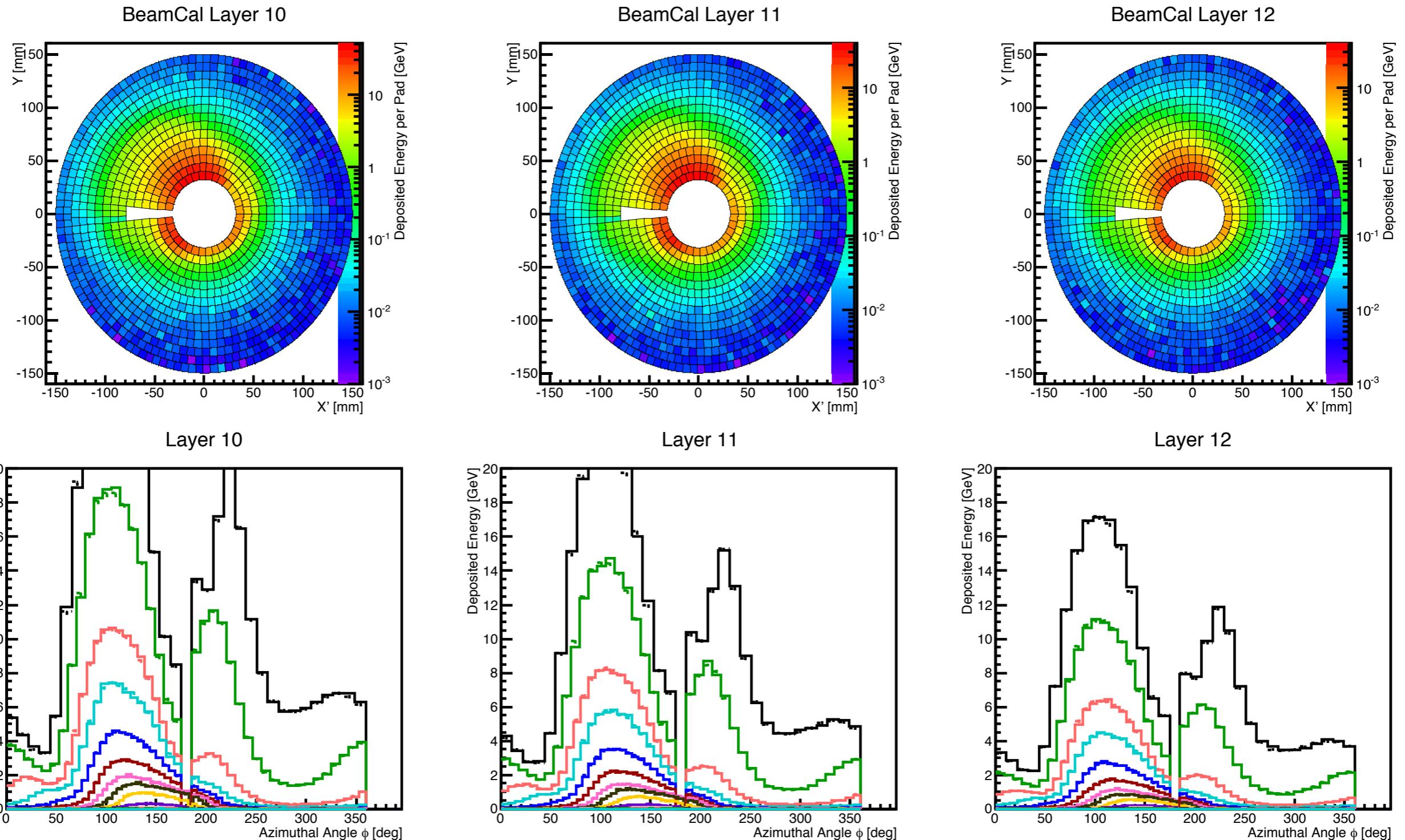
<!--The name of the rootFile which will contain the TEfficiency objects-->
<parameter name="EfficiencyFilename" type="string">TaggingEfficiency.root
</parameter>

<!--Number of Event that should be printed to PDF File-->
<parameter name="PrintThisEvent" type="int"> -1 </parameter>

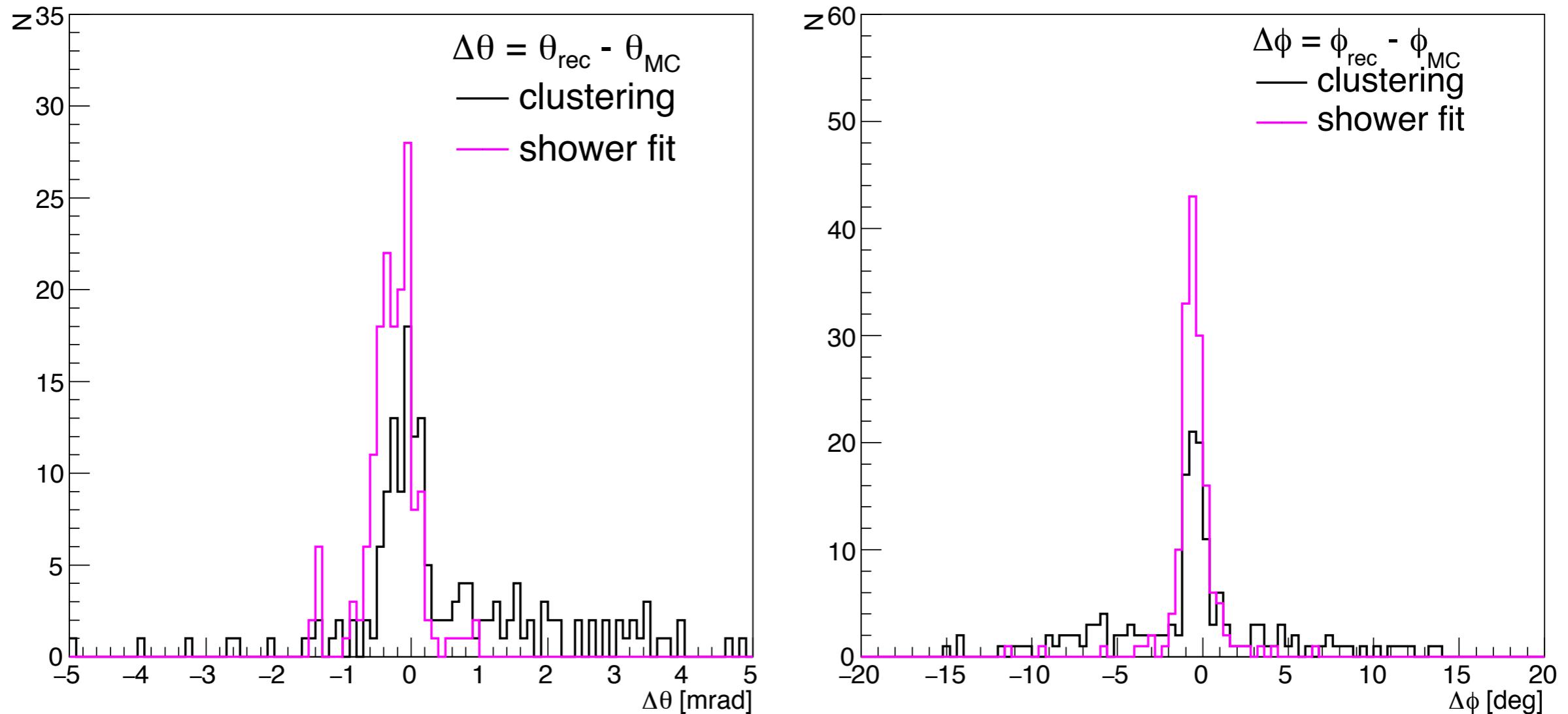
<!--verbosity level of this processor
     ("DEBUG0-4,MESSAGE0-4,WARNING0-4,ERROR0-4,SILENT")-->
<parameter name="Verbosity" type="string"> DEBUG2 </parameter>

</processor>
```

Event display



Resolution comparison: clustering vs fitting



- A simple test of reconstruction performed for 250GeV electrons with 500GeV CMS energy ILC background.
- Angular resolution for two reconstruction methods is similar
- The efficiency of shower fitting is slightly better: 0.95 vs 0.92
- None of methods were tuned for these energies and background

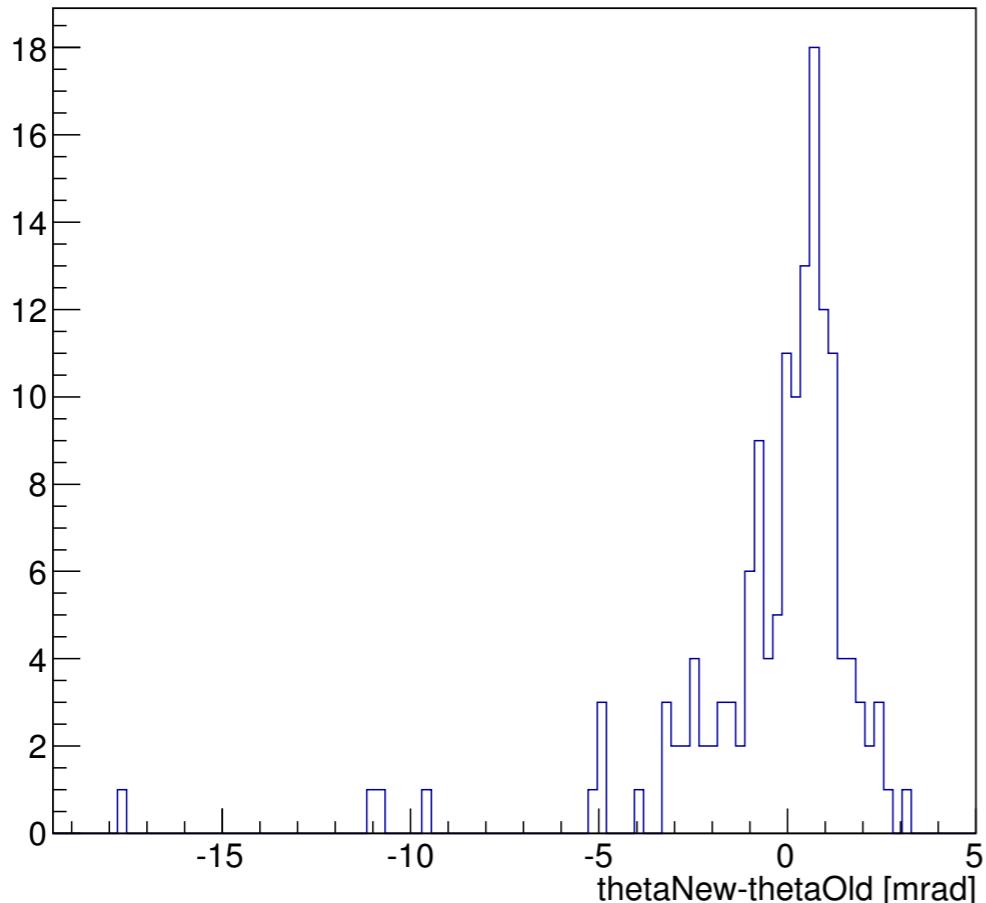
Summary

- A shower fitting algorithm is implemented in BeamCal reconstruction along with clustering
- It gives sensible improvement in reconstruction precision of phi over clustering, but is worse at theta.
- The fit procedure still has to be optimised and tuned for different energies and backgrounds
- The approach is sensitive to background statistics, therefore performs better with parametrised/averaged background rather than pregenerated
- A CLICdp note is in work with a detailed algorithm description.

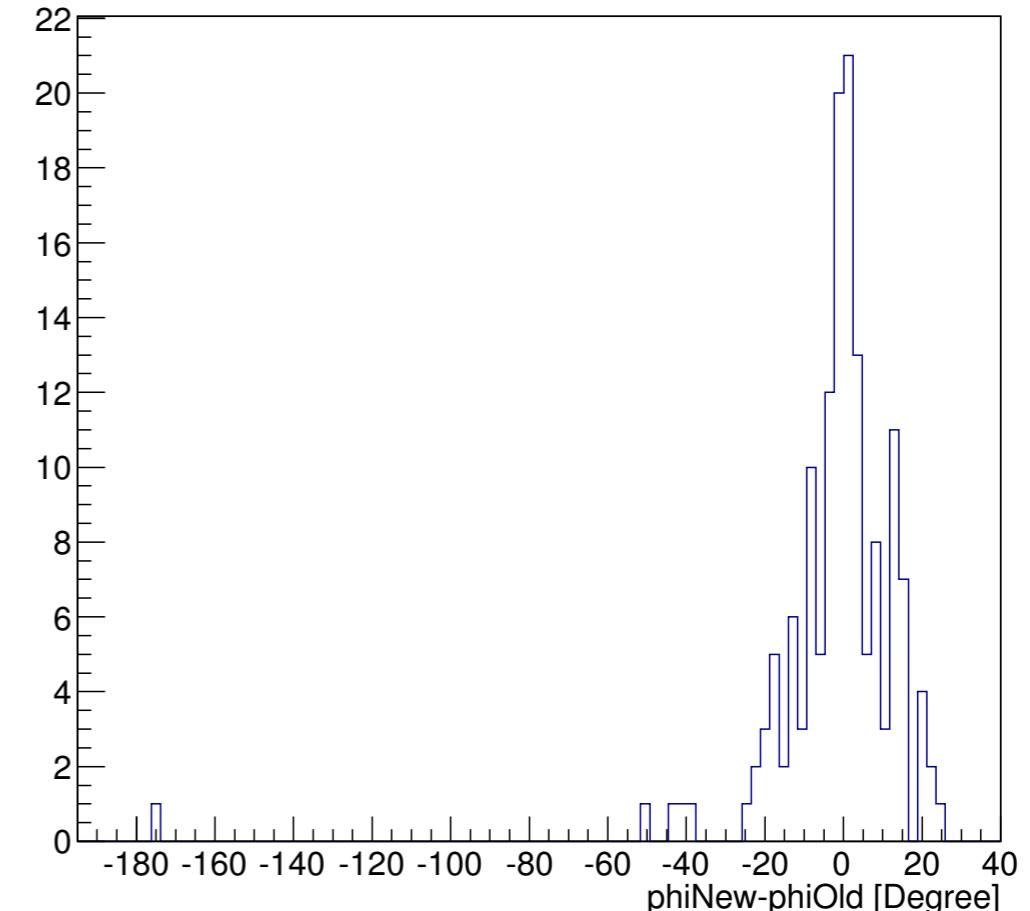
Backup slides

DBD comparison with Clustering

Difference in Theta



Difference in Phi



How to: install

- Source code in FCAL svn repository:
<https://svnsrv.desy.de/svn/FCAL/Software/FCalClusterer/>
- Standard installation (Needs Marlin, LCIO, Gear and ROOT)

```
svn co https://svnsrv.desy.de/public/FCAL/Software/FCalClusterer/
trunk FCalCluster cd FCalCluster
mkdir build
cd build

cmake -C $ILCSOFT/ILCSoft.cmake .. make install
export MARLIN_DLL=...
```

- Configure run in the xml format as follows

Example from the Code: doClustering



```
BeamCalClusterList doClustering( BCPadEnergies totalEnergy ,
                                BCPadEnergies background ,
                                BCPadEnergies backgroundSigma ,
                                BCPadCuts cuts ) {

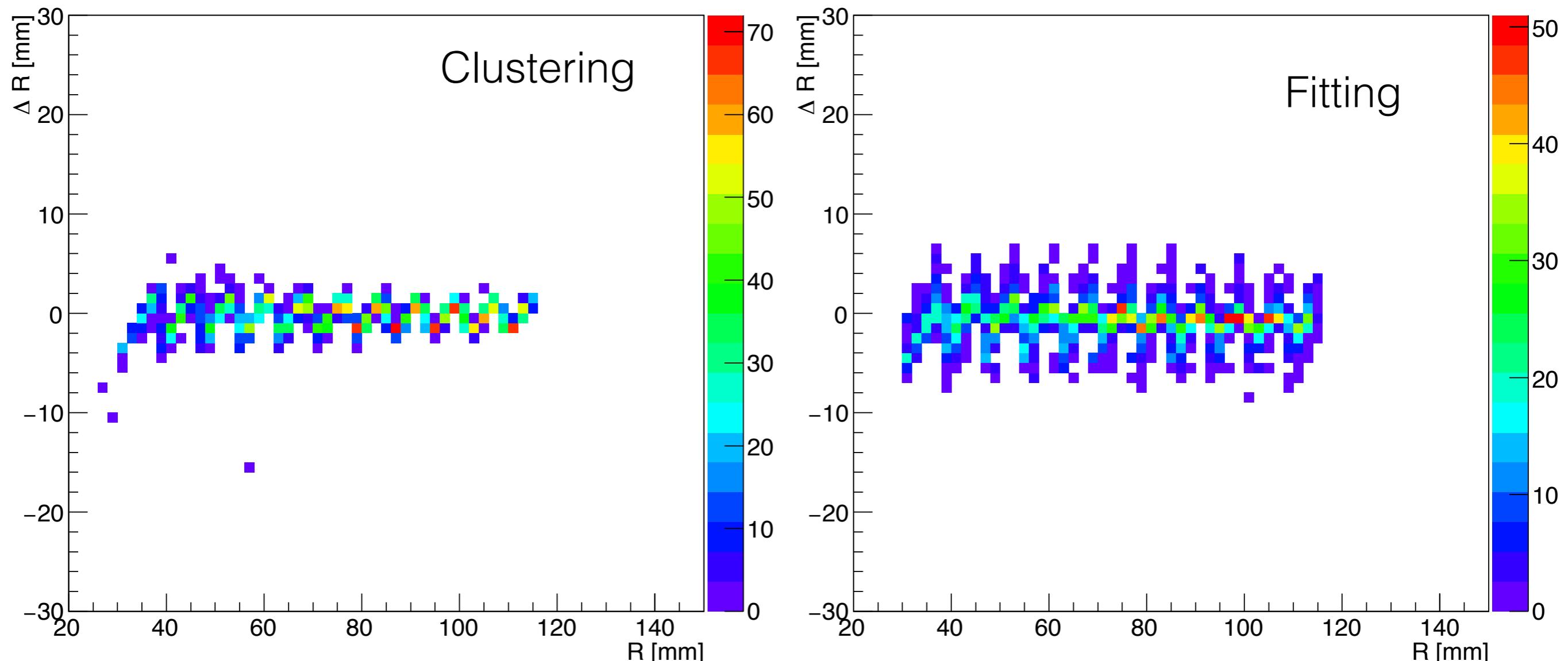
    BeamCalClusterList beamCalClusters ;
    totalEnergy .subtractEnergies( background );

    PadIndexList selectedPads = getPadsAboveSigma( totalEnergy , backgroundSigma , cuts );

    while( not selectedPads .empty() ) {
        PadIndexList padsForNextRound ;
        TowerIndexList myTowerIndices = getTowersFromPads( selectedPads );
        TowerIndex largestTower = max_element( myTowerIndices );
        forall( TowerIndex testTower in myTowerIndices ) {
            if ( not areNeighbours( largestTower , testTower ) ) {
                selectedPads .removeTower( testTower );
                padsForNextRound .addTower( testTower );
            }
        }
        beamCalClusters .addCluster( getClusterFromAcceptedPads( totalEnergy , selectedPads ) );
        //try again for the remaining pads without the cluster that was already found
        selectedPads = padsForNextRound ;
    } //while there are towers

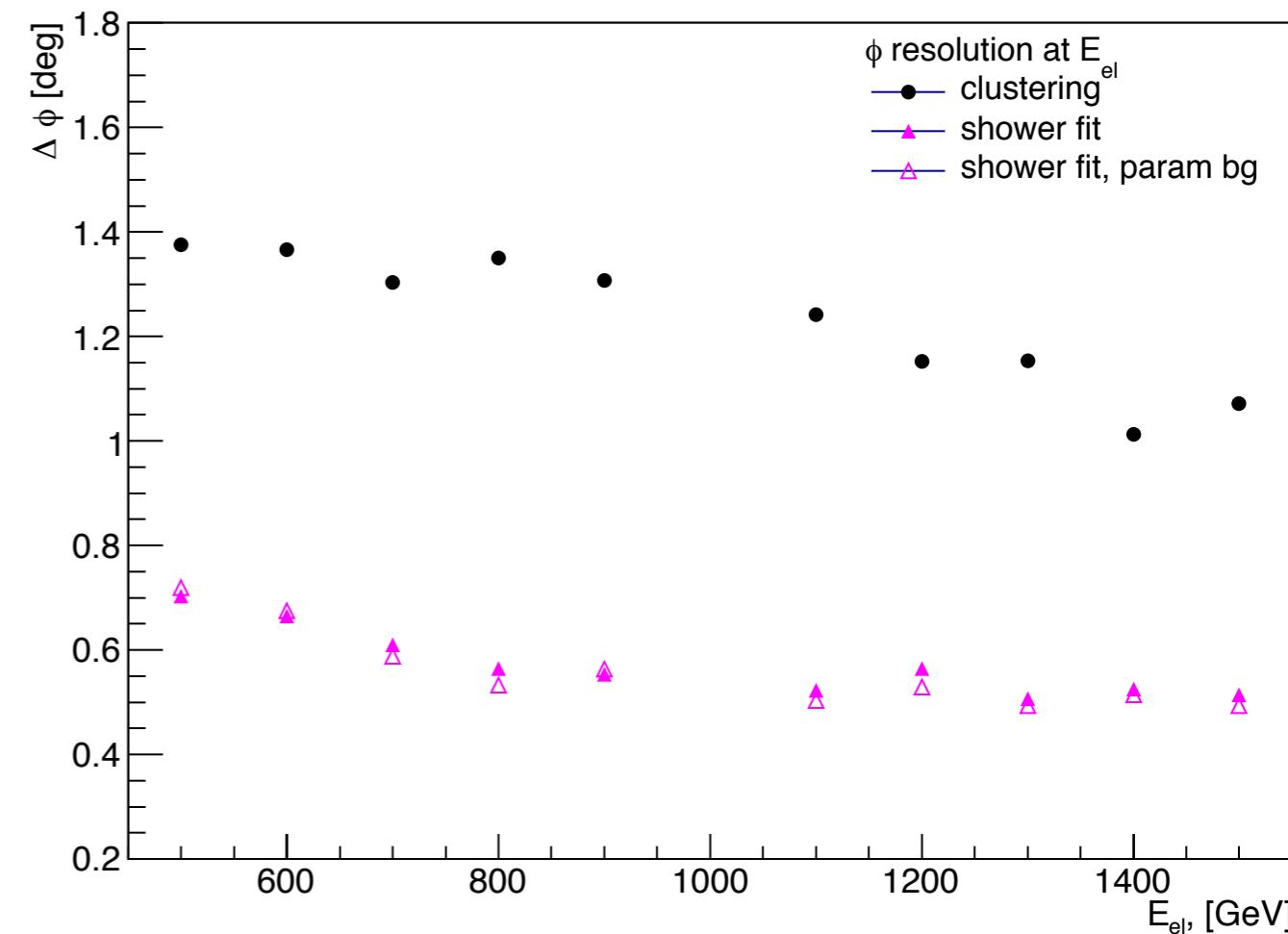
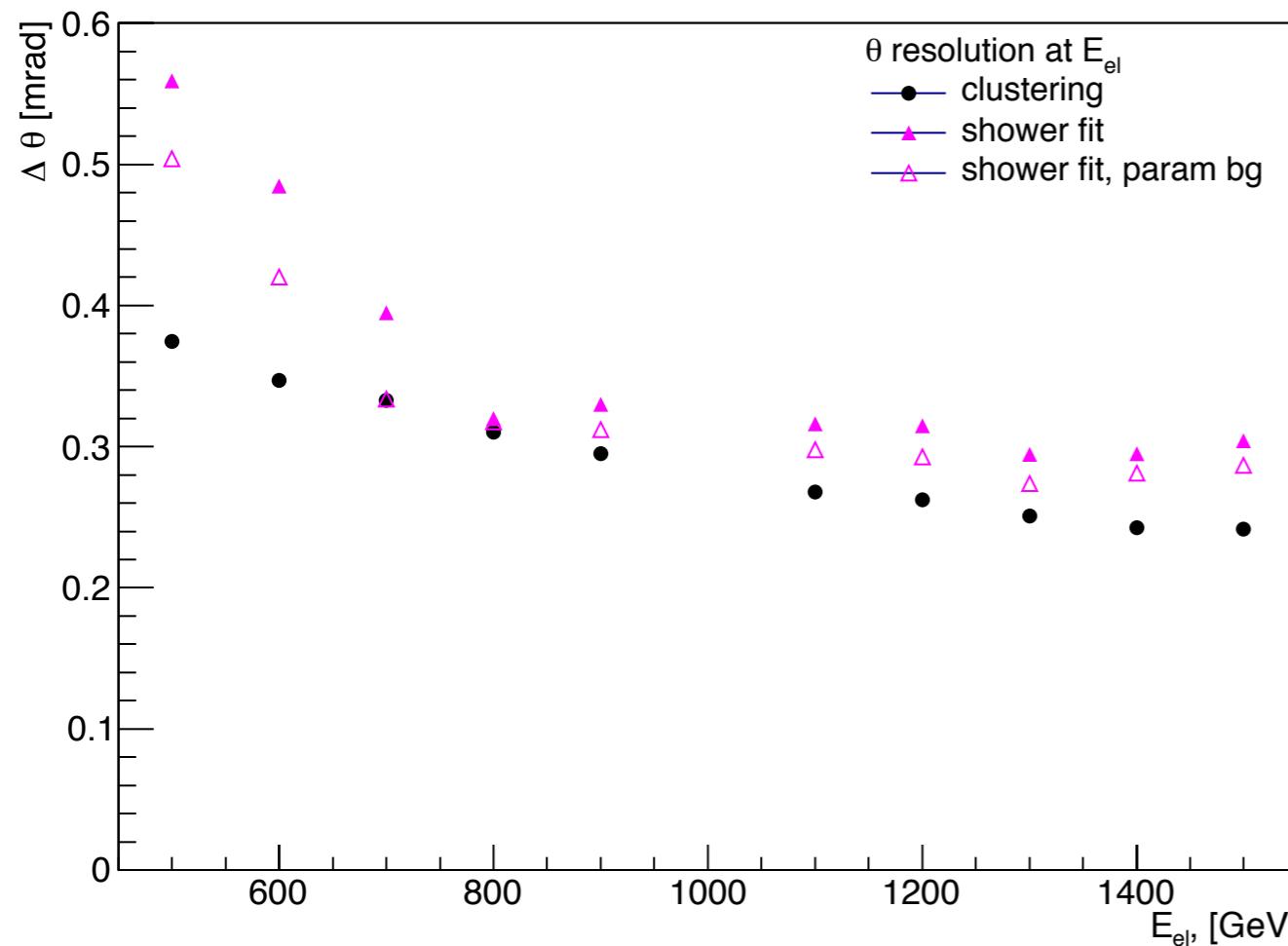
    return beamCalClusters ;
}
```

Resolution comparison

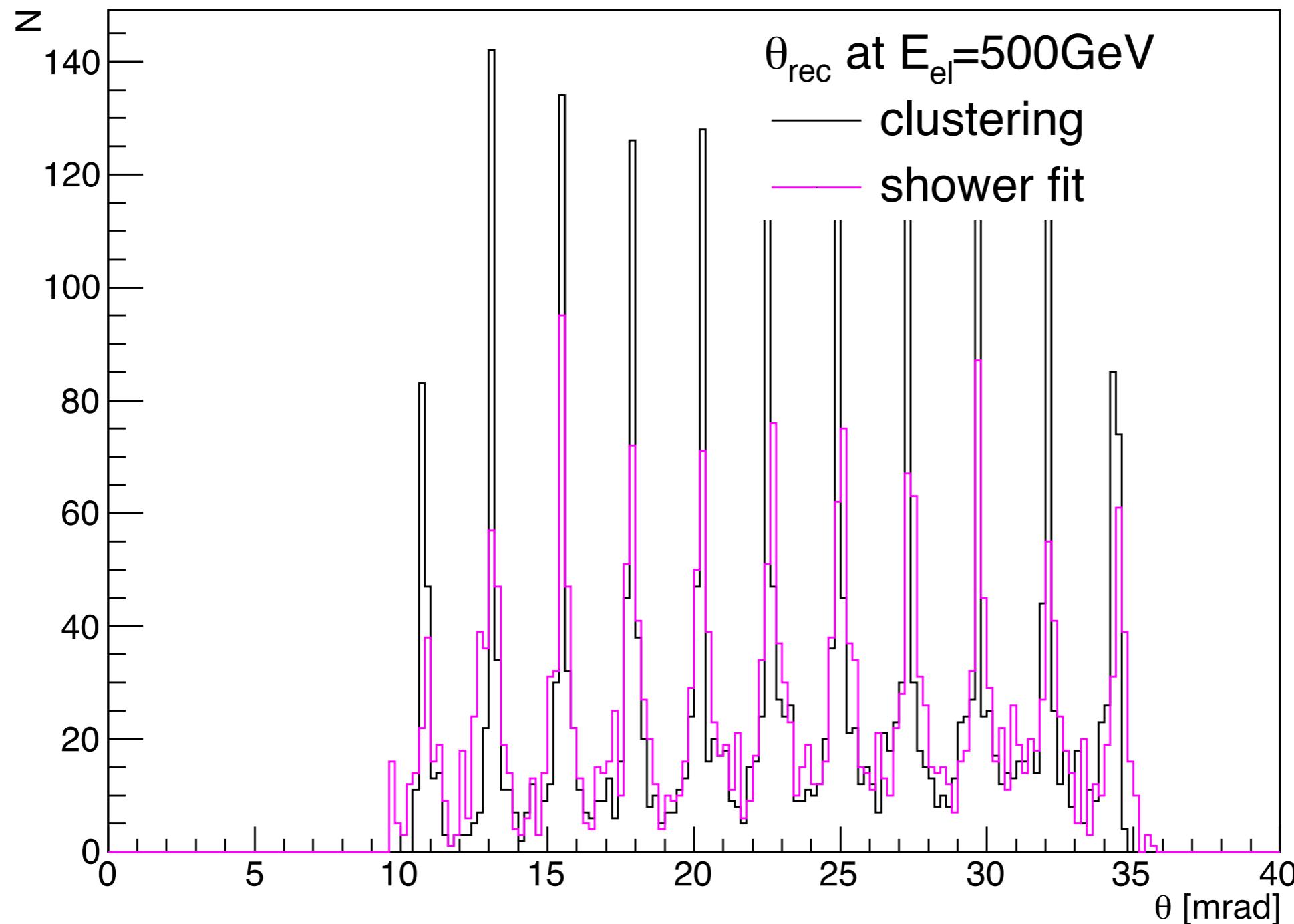


- Granularity induced systematic offsets
- The effect is absent in ϕ direction due to relatively shifted pads (?)

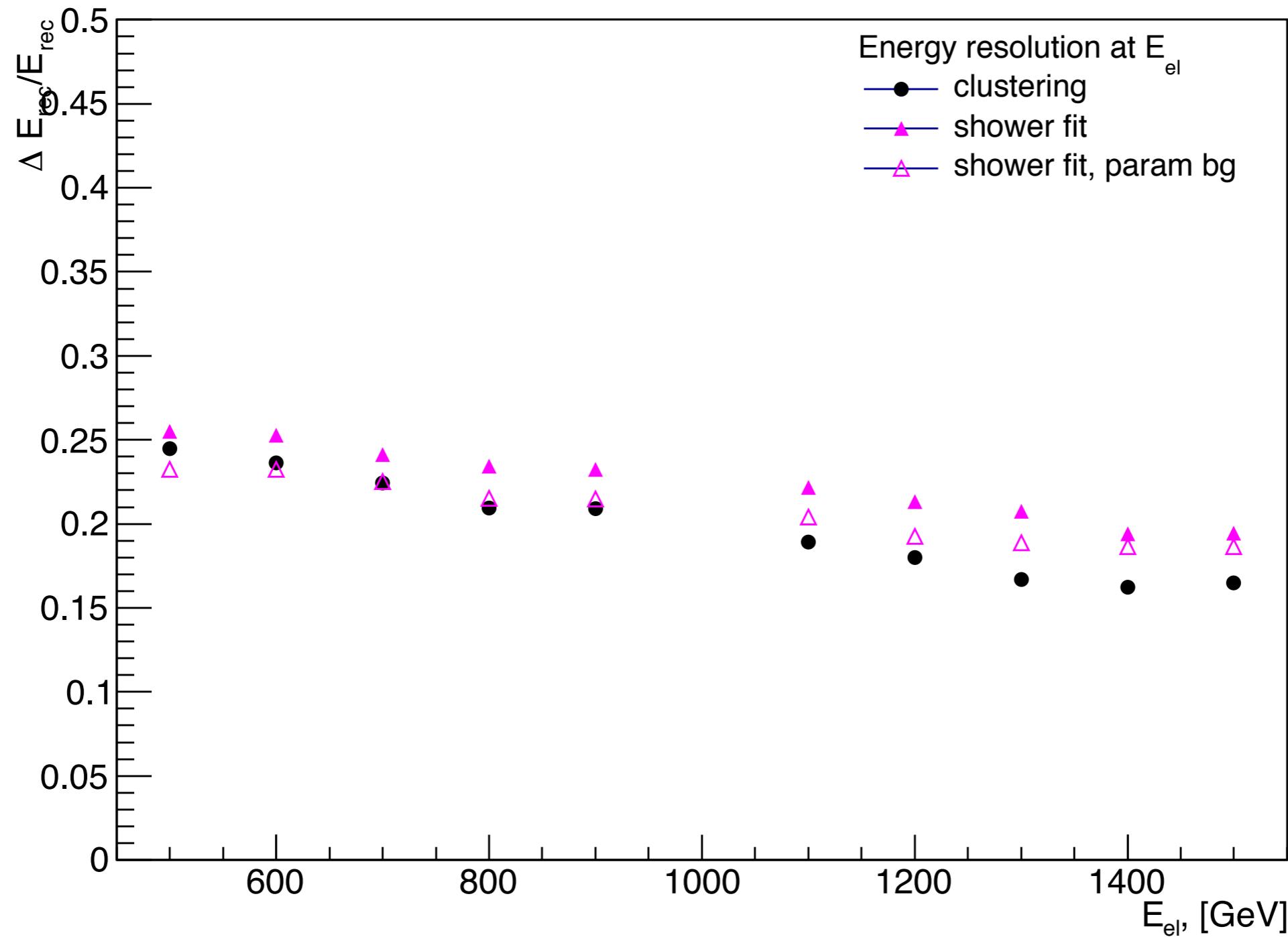
Angular resolution



θ reconstructed values

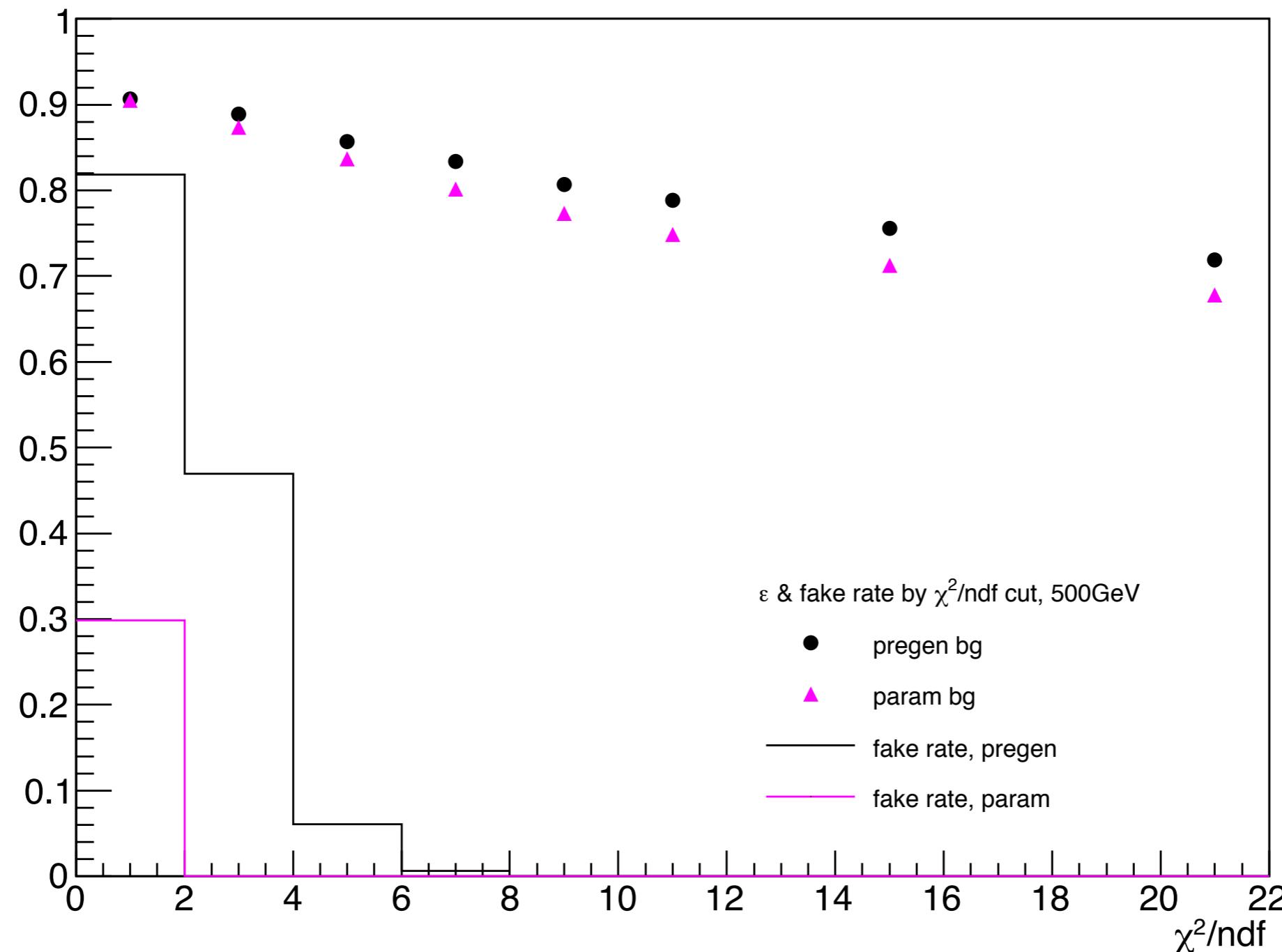


Energy resolution



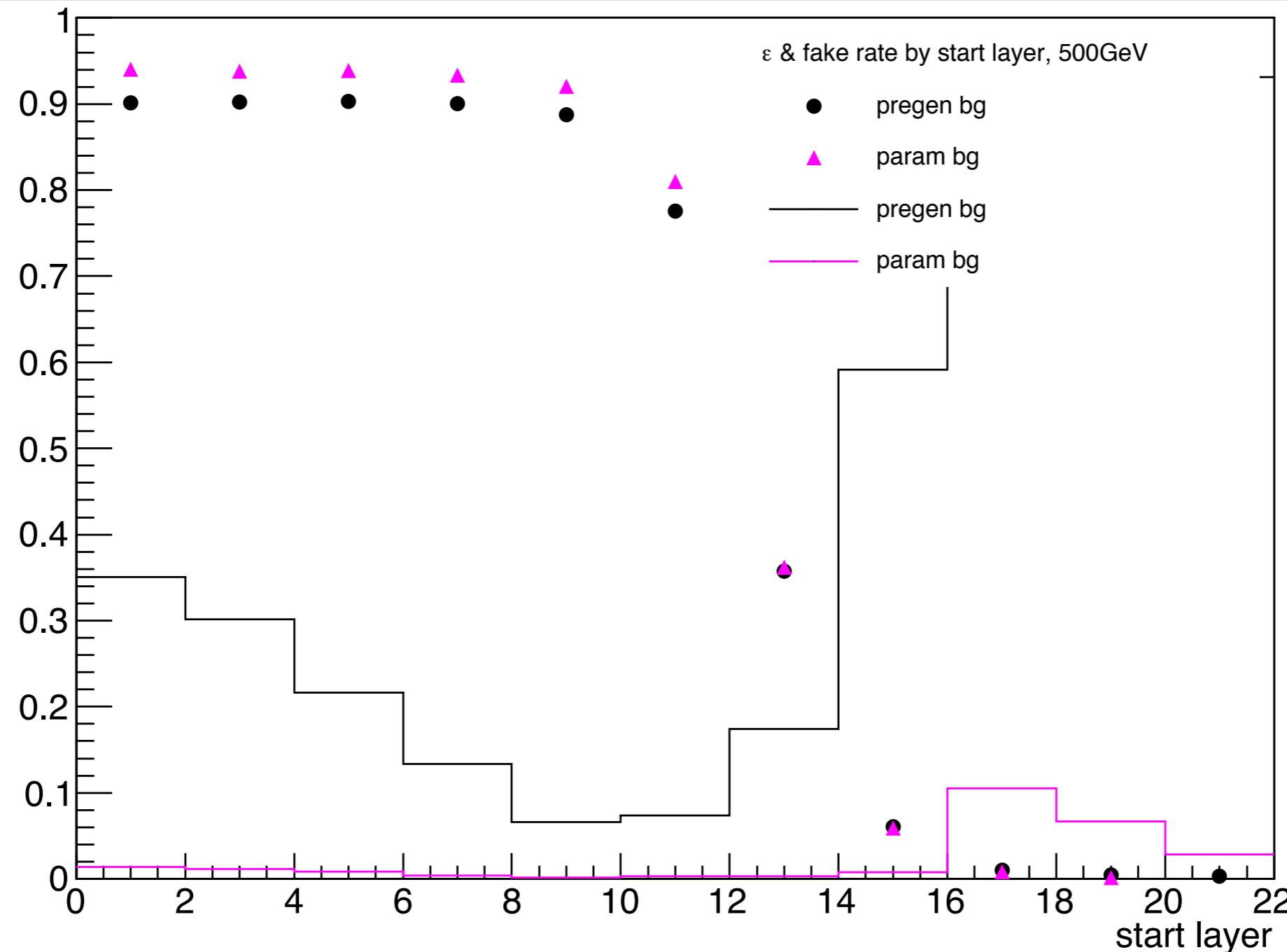
- Energy resolution is approximately same as for clustering

Scan on χ^2/ndf cut



Optimal χ^2/ndf cut value is 2 for parametrised background, and 5 for regenerated

Starting layer for cluster search



- The parametrised background induces much less fakes in fitting
- This is due to high sensitivity to background statistics in fitting procedure

Pregenerated background BeamCal reco.

- Currently: a Monte-Carlo style for the beamstrahlung induced background.
 - a large set of pregenerated BG distributions (=1 BX)
 - upon reconstruction a desired number of BX's is randomly selected, added and overlaid with signal.
 - cluster search and reconstruction

BeamCal_100_313.root
BeamCal_100_314.root
BeamCal_100_315.root
BeamCal_100_316.root
BeamCal_100_317.root
BeamCal_100_318.root
BeamCal_100_319.root
BeamCal_100_320.root
BeamCal_100_321.root

...

BeamCal_100_932.root
BeamCal_100_933.root
BeamCal_100_934.root
BeamCal_100_935.root

600 files
 $\Sigma \sim 0.5\text{Gb}$

Parametrised background

- To avoid management of all the files, initial idea was to
 - parametrise the distributions of BX-to-BX energy deposition in pads, assuming gaussian shape
 - for reconstruction: random generation in each pad according to their distributions.
 - overlay the bg, etc.

