ECAL Tutorial. E_T and E/p studies.

V. Riabov

Outlook

- ECAL tutorial
- E_T distributions, correlation with N_{TPC}, centrality
- E/p ratio for hadrons and electrons

ECAL Tutorial

Tutorial

- Text attached to the agenda. To be relocated in the right place by web designers

- Content:
  1. Geometry
  2. How to run
  3. Basic operation principles of emcKI code
  4. Data handling
  5. Cluster variables
  6. Photon identification
  7. Charged particle identification
  8. Code examples

1. Geometry

Current version of ECAL geometry is v.4. Please make sure that this version of ECAL geometry is called in mpdroot/macro/mpd/geometry_stage1.C. The following lines should be verified:

```
FairDetector *Emc = new MpdEmcKI("EMC", kTRUE);
Emc->SetGeometryFileName("emc_v4.root"); // geometry version
fRun->AddModule(Emc);
```

Details of the ECAL geometry can be found in the recent presentation by M. Martemyanov, https://indico.jinr.ru/event/2893/contributions/15531/attachments/11967/19918/Report-Martemianov_2893.pdf

Latest version of tower-by-tower calibration should be used:

mpdroot/input/MpdEmcCalib.root

2. How to run

mpdroot/detectors/emc/emcKI/ – digitizer/clusterizer that works with standalone clusters and high multiplicity collisions. It is recommended to be used for physics studies.

In order to run the code the following lines need to be added in reco.C:

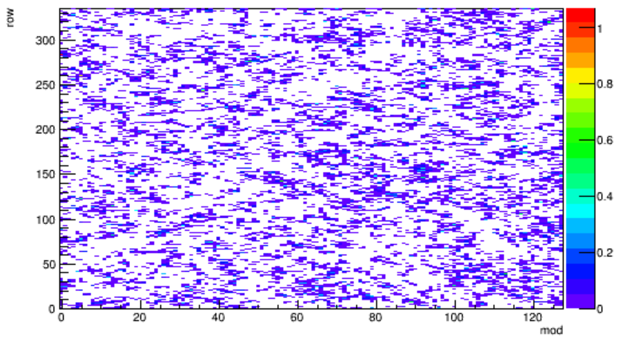
```
//emcKI digitizer
   FairTask *emcHP = new MpdEmcDigitizerKI();
   fRun->AddTask(emcHP);

// emcKI clusterizer
   MpdEmcClusterizerKI *EmcCluster = new MpdEmcClusterizerKI();
   fRun->AddTask(EmcCluster);

// emcKI matching
   MpdEmcMatchingKI * EmcMatcher = new MpdEmcMatchingKI() ;
   fRun->AddTask(EmcMatcher) ;
```

3. Operation principles

Details can be found in the Physics Forum presentation, https://indico.jinr.ru/event/1097/

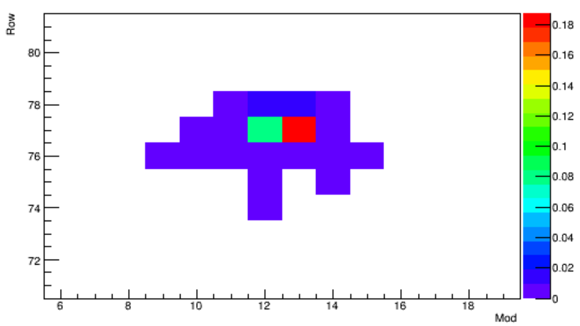
The operation principles are based on splitting (unfolding) of partly overlapping showers occurring in high multiplicity collisions. The splitting is based on the known shape of electromagnetic showers.

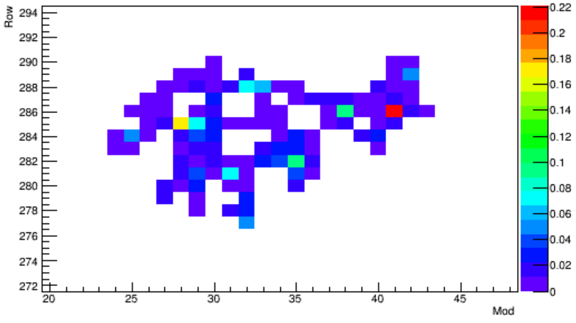
- UrQMD, AuAu@11, b ~ 1 fm → most central collisions
- $E_{threshold}$ = 5 MeV → occupancy is ~ 27%



occupancy ~ 0.27

- Clusters are reconstructed as groups around a local maximum:
  - ✓ a fraction of clusters is reconstructed as stand alone showers
  - ✓ others are reconstructed as groups of merged showers to be unfolded





4. Data handling

Produced DST files have an mpdEMCClusters array of MpdEmcClusterKI clusters.

Post-production macro:

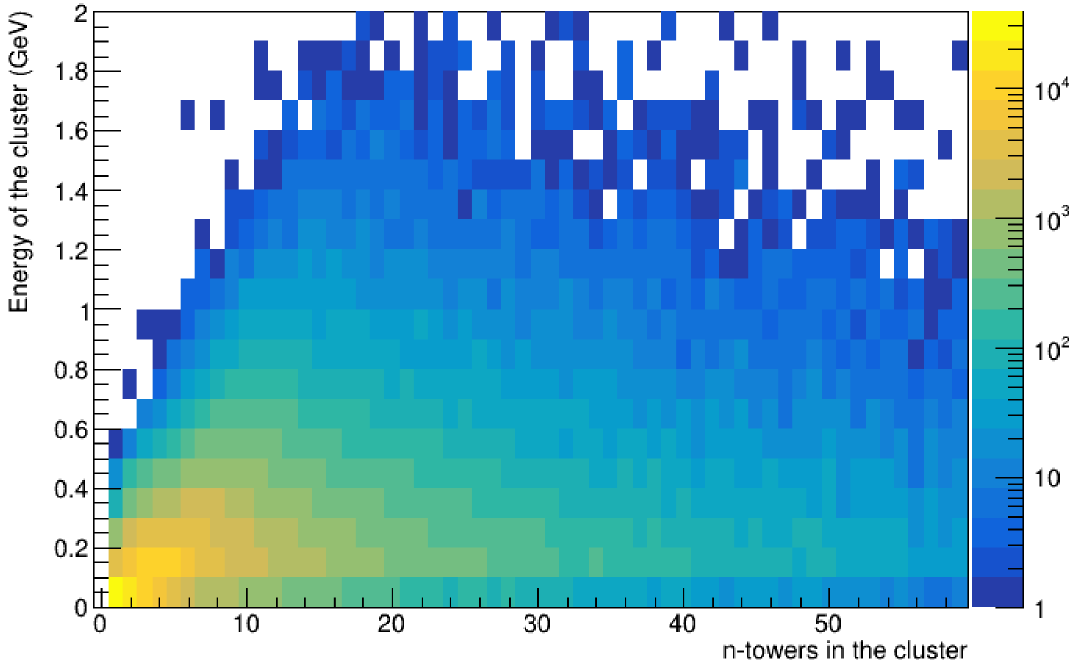
```
#include "MpdEmcClusterKI.h"

...................
 TTree *inTree;
 inTree = (TTree*) inFile->Get("mpdsim");

...................
 TObjArray * mpdEMCClusters  = new TObjArray() ;
 MpdEmcClusterKI* EMCCluster;

 inTree->SetBranchAddress("EmcCluster",&mpdEMCClusters);

...................
   for (long int j=0; j<mpdEMCClusters->GetEntries(); j++){
     EMCCluster = (MpdEmcClusterKI*)mpdEMCClusters->At(j);
     // do whatever you need with a given cluster EMCCluster here …
   }
```

5. Cluster variables - I

//================

EMCCluster->GetMultiplicity();// number of towers in the cluster.

Usually clusters consist of a few towers. Clusters that consist of one tower only are either very low energy clusters or just fragments of other (higher energy) showers. The cluster energy is counted as a sum of tower energies. The recommendation is to select clusters with number of towers > 1.



5. Cluster variables - II

//================

EMCCluster->GetE(); // simulated energy of the cluster

In order to get the true particle energy deposition the simulated energy should be corrected.

For v.4 geometry of the ECAL we recommend the following corrections:

```
float     conv = 1.0/0.3065; // only ~ 30% energy is collected
...................
float E_true = EnCorr( EMCCluster->GetE() * conv );
```

Function EnCorr() compensates for non-linearity (3% effect)

```
...................
Float EnCorr(float e)
{
  float x = e;
  if (x > 2.5) x = 2.5; // parameterization is available up to 2.5 GeV only

  float corr[6] = { -1.321760e-002, 8.475623e-002, -9.007282e-002, 4.742396e-002, -1.279561e-002, 1.397394e-003 };
  float ecorr = corr[0] + corr[1]*x + corr[2]*x*x + corr[3]*x*x*x + corr[4]*x*x*x*x + corr[5]*x*x*x*x*x;

  return e/(1.0 + ecorr);
}
```

Recommendation is to select clusters with E_true > 0.05 (50 MeV) in the physical analyses.

5. Cluster variables - III

```
EMCCluster->GetX(); // x-coordinate of the cluster
EMCCluster->GetY(); // y-coordinate of the cluster
EMCCluster->GetZ(); // z-coordinate of the cluster
EMCCluster->GetRho(); // radius of the cluster, sqrt(x**2+y**2)
```

Center of gravity X,Y,Z coordinates of the cluster in the global coordinate system

Recommendation is to use these coordinates to calculate px, py, pz momentum components:

```
x = EMCCluster->GetX() - primVert.X();
y = EMCCluster->GetY() - primVert.Y();
z = EMCCluster->GetZ() - primVert.Z();
r = TMath::Sqrt(x*x + y*y + z*z);

Px = E_true * x / r;
Py = E_true * y / r;
Pz = E_true * z / r;
```

5. Cluster variables - IV

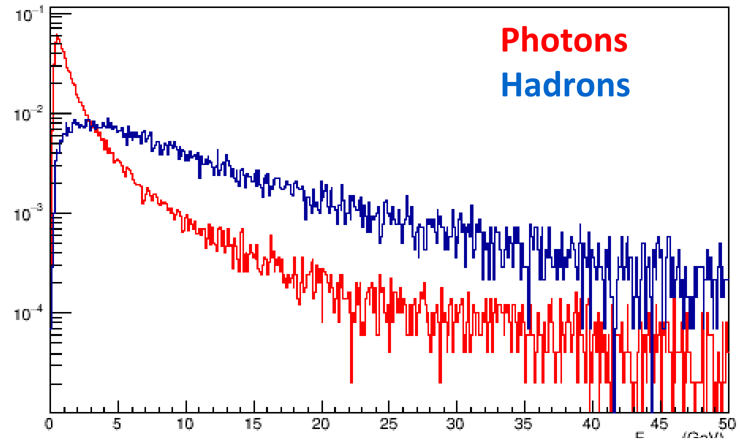
EMCCluster->GetChi2(); // chi2/NDF of the cluster
defines how close the reconstructed cluster shape to the shape expected for electromagnetic shower. The clusters from photons and high-pT electrons are expected to have electromagnetic shape in the ECAL. The shape analysis is possible only is there are more than one tower in the cluster.

Please note that clusters from low-momentum electrons are highly asymmetric due to a large incident angle of the electron tracks bent in the magnetic field. As a result, one cannot expect electromagnetic shape for such clusters.

$$\text{Chi2} = \sum_i \frac{\left(E_i^{measured} - E_i^{expected}\right)^2}{\sigma_i^2},$$

$$\sigma_i^2 = A \cdot E_i^{expected} \cdot \left(1 - \frac{E_i^{expected}}{E}\right),$$

σ_i^2 is tuned from simulations,
NDF – number of towers in the cluster



Chi2/NDF

5. Cluster variables - V

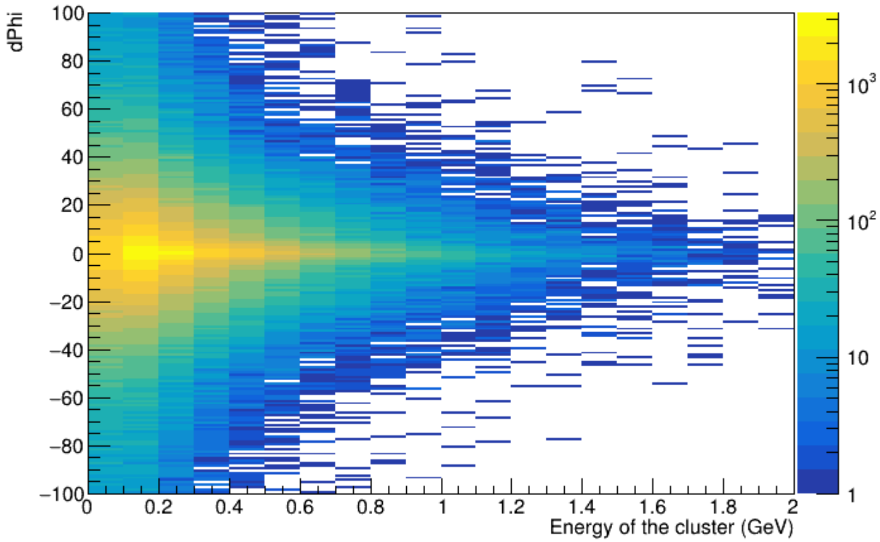
EMCCluster->GetTime(); // simulated time of the shower

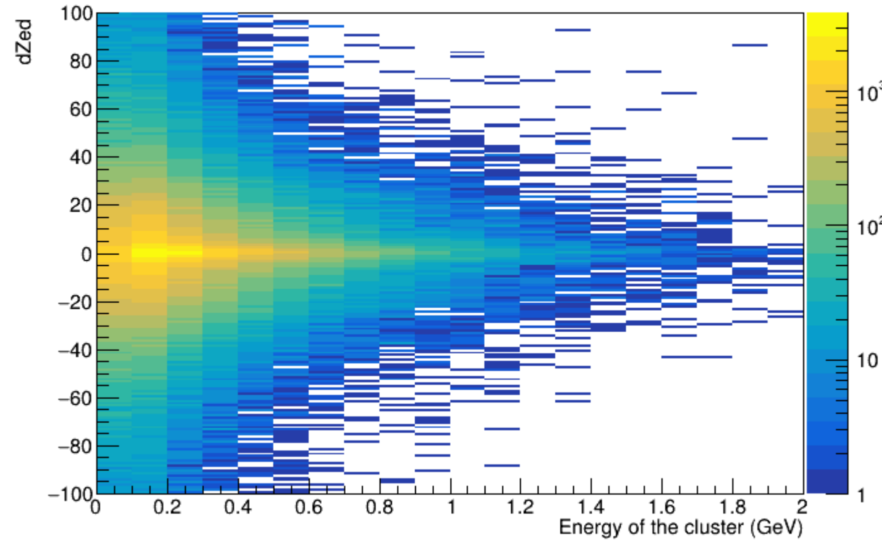
This variable is a simulated time of the shower. The meaning of this variable is close to intrinsic time resolution of the ECAL. The real ECAL time resolution is defined by parameters of the electronics, not by intrinsic time resolution of the towers. For realistic timing please smear the simulated time with a resolution of 0.5 ns:

```
TRandom RND;
Tcl = RND.Gaus(EMCCluster->GetTime(), 0.5);
```

5. Cluster variables - VI

EMCCluster->GetDPhi(); // distance from the cluster to the closets TPC track in dPhi
EMCCluster->GetDZ(); // distance from the cluster to the closets TPC track in dZed
These variables help to assert whether the ECAL cluster is initiated by a charged particle or not. The decision is to be made based on n-sigma(p_T) deviation of the cluster from the closest track.





5. Cluster variables - VII

EMCCluster -> GetNumberOfTracks(); // number of MC tracks that contribute to the cluster

MCCluster -> GetMCTrack(track, MC track index, energy contribution); // returns energy contribution and MC track index for track-th contributor

These variables are used to understand what MC tracks contribute to the measured cluster energy. In low multiplicity collisions only one MC track contributes to the cluster and EMCCluster -> GetNumberOfTracks() is equal to one. In high multiplicity events where most of the showers overlap it is impossible to set unambiguous association between the clusters and MC tracks. Instead, up to five MC tracks that contribute the most to the cluster energy are stored. The contributions are sorted by energy so that the first contributor is the most important. See an example below:

```
long int ind_mc_cont;
float e_mc_cont;

for (int kk = 0; kk < EMCCluster -> GetNumberOfTracks(); kk++){
  EMCCluster -> GetMCTrack(kk,ind_mc_cont,e_mc_cont);
  ind_mc_cont = MC index of the kk-th contributor to the cluster
  e_mc_cont - energy deposited by kk-th continutor
}
```

6. Photon ID

1) Shower shape: EMCCluster->GetChi2() < 4.
The tighter the cut the higher the probability that the cluster is produced either by a single photon or by high-energy electron but the lower the reconstruction efficiency. The higher the photon energy the higher the chance of correct identification.

2) Time of flight. Photons trajectories do not bend in the magnetic field and they travel with a speed of light. Hence one can expect smaller time of flight for photon signals. Recommendation is to select clusters with Tcl < 2 ns. The cut is quite efficient at low cluster energies where signal-to-background ratio is the smallest.

3) Charged particle veto. dPhi and dZed from a cluster to the closest track are parameterized as a function of cluster energy and then n-sigma(E_true) selection can be used to reject clusters reconstructed too close to the charged tracks. For rough estimations one can use:
!(fabs(EMCCluster->GetDPhi()) < 15 && fabs(EMCCluster->GetDZ()) < 15).

Charged particle veto cut is very efficient in low multiplicity events. It remains to be efficient in high multiplicity events, however, due to higher probability to find a charged track next to the true photon signal the reconstruction efficiency decreases.

The cuts are correlated to some extent. It means that efficiency of multiple selections won't be equal to the product of efficiencies for each of them.

7. Track ID

ECAL has nothing to offer in addition to hadron identification in the TOF subsystem. However, ECAL can be used for efficient identification of electrons. Electrons are identified in the ECAL by E/p ratio and time of flight. The identification procedure consists of a few steps:

1) select a high quality track. The definitions depend on a particular physical analysis.
2) identify track as electron in the TPC (by dE/dx) and TOF (by time of flight).
3) extrapolate track to the ECAL surface and find the closest ECAL cluster.
4) verify that the distance from the extrapolated track to the closest ECAL cluster is within 2-3 sigma in dPhi and dZed or within a constant limit optimized for a given physical analysis.
5) determine energy and time of flight of the ECAL cluster associated with the track in p.4.
6) calculate E/p ratio and compare it to the value expected for true electron with the same energy or momentum.
7) calculate mass^2 of the particle and compare it to the value expected for true electron with the same energy or momentum.

8. Code examples

Two code examples are given to illustrate usage of variables:

1) /eos/nica/mpd/users/riabovvg/ECAL_Tutorial_Examples/Pi0_analysis
This is a simplified example of pi0 reconstruction. This code works with the simulated DST files. In order to run the code please type: root -b -q Pi0Analysis.C'("DST.root")‘.

The code will process all events in the input file and produce a number of histograms in the output Pi0Analysis.root file:

2) /eos/nica/mpd/users/riabovvg/ECAL_Tutorial_Examples/Photon_analysis
This is a simplified analysis of single photons and tracks matched to the ECAL. The code works with the simulated DST files. In order to run the code please type: root -b –q macro_run.C’(1)’

The code will process all events in the input file and produce a number of histograms and a root Tree in the output TreeHisto_1.root file:

Conclusions

- V.1 of ECAL Tutorial has been released
- Please have a look and report any problems

E_T distribution

Selection cuts

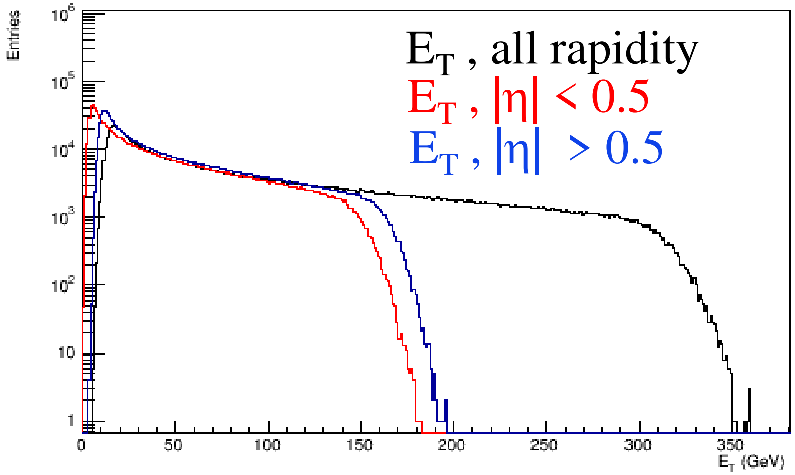
- Event selections:
  - ✓ BiBi@9.2, UrQMD v.3.4
  - ✓ z-vertex smeared with σ = 22 cm, |z-vertex| < 50 cm
  - ✓ no centrality/multiplicity selections

- Track selections:
  - ✓ n-hits > 10
  - ✓ $|\eta| < 1$
  - ✓ $|DCA_{x,y,z}| < 3\sigma$

- ECAL cluster selections:
  - ✓ E_{γ} > 50 MeV
  - ✓ n-towers > 1

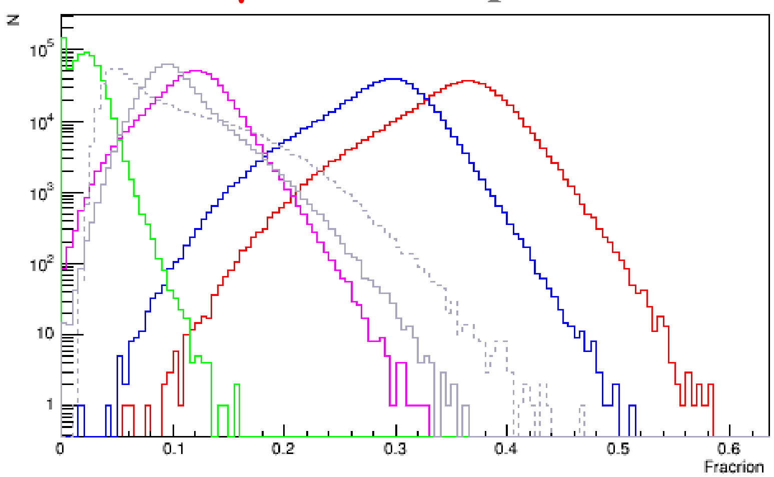
E_T distributions

- Transverse energy E_T



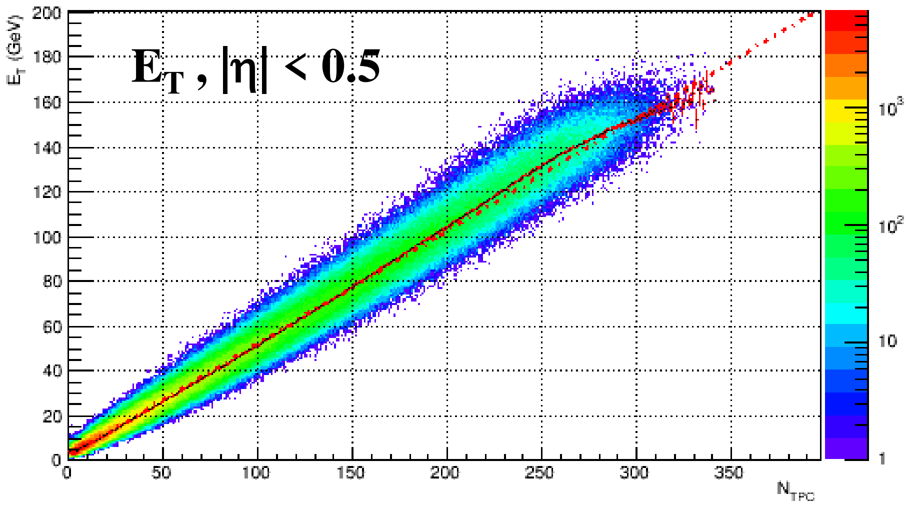
- Contributors:

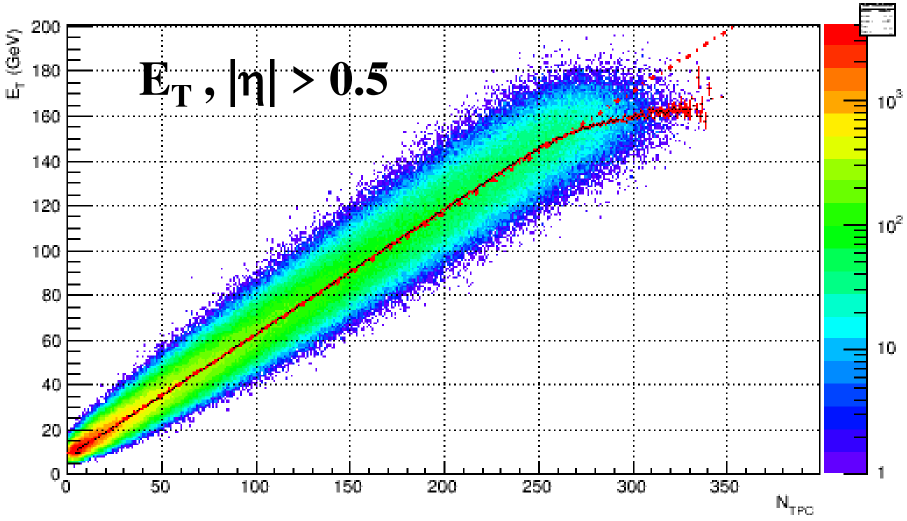
γ, $\pi^{\pm}$, $e^{\pm}$, $K^{\pm}$, $p^{\pm}$, n



- Main contributors:
  - ✓ pions (**photons**, $\pi^{\pm}$, $e^{\pm}$)

E_T vs. N_{tracks}

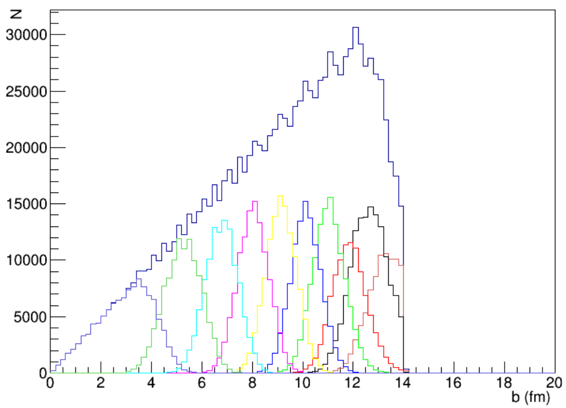


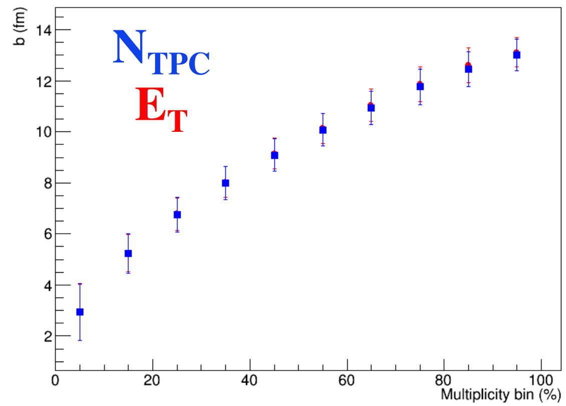


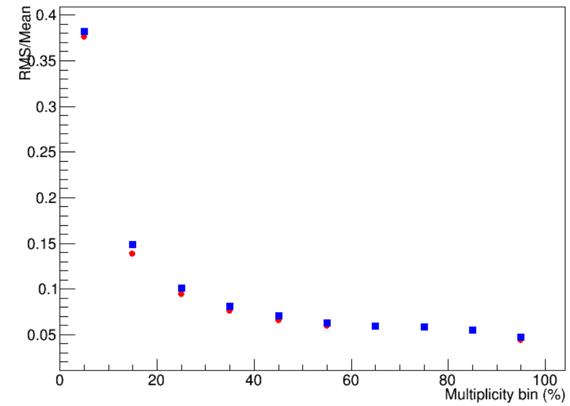
- Width of E_T vs. N_{TPC} correlations depends on rapidity selections

Multiplicity bins by E_T

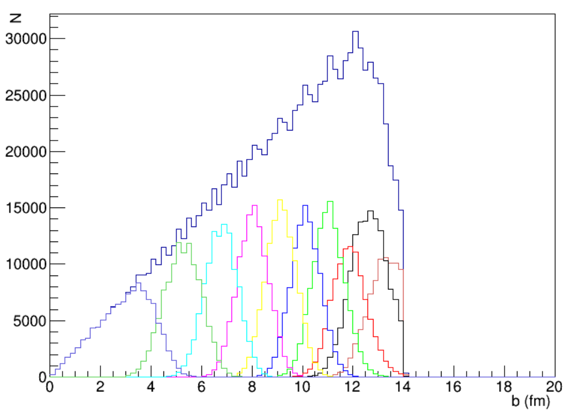
- E_T , $|\eta| < 0.5$

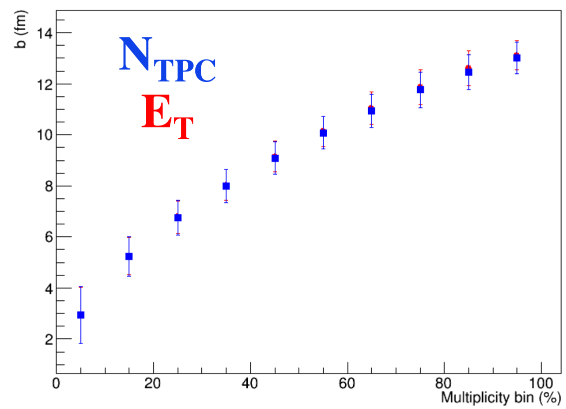


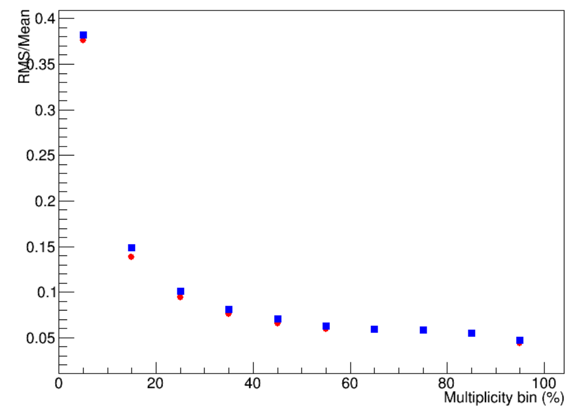




- E_T , $|\eta| > 0.5$







- Very similar events are selected with N_{TPC} ($|\eta|<0.5$) and different E_T multiplicity selections
- No obvious dependence on rapidity for E_T

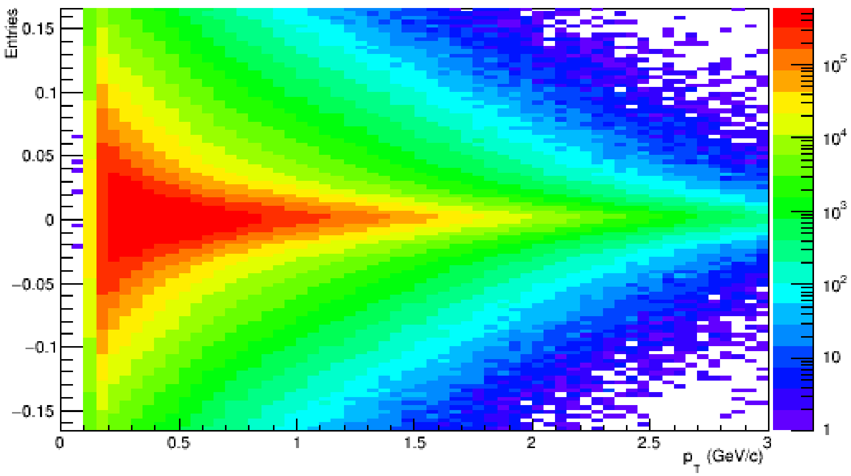
Conclusions

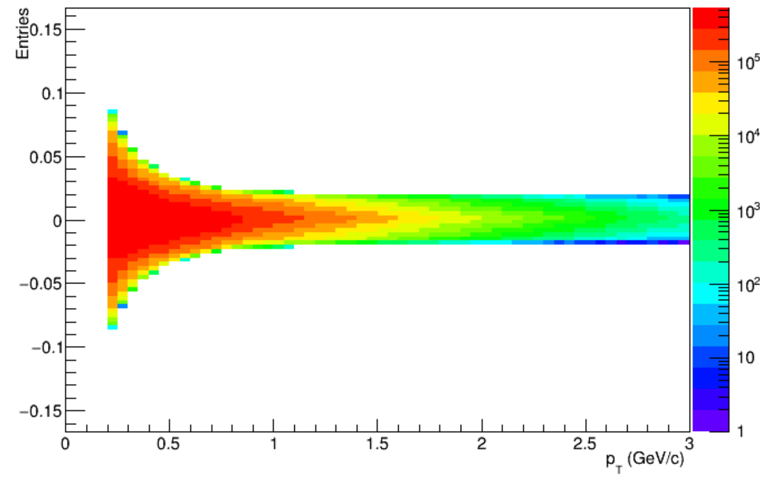
- Observe close correlation between E_T and N_{TPC}
- Correlation width depends on rapidity selections

E/p distribution

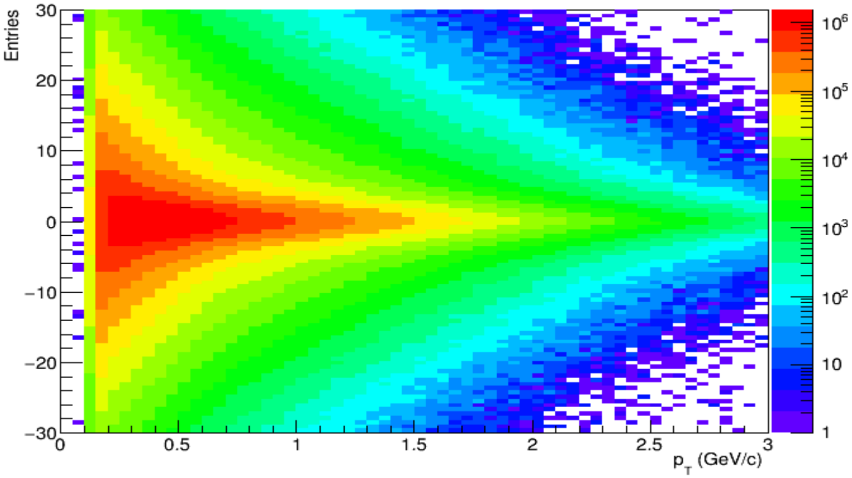
Track-to-cluster matching: 2σ selections

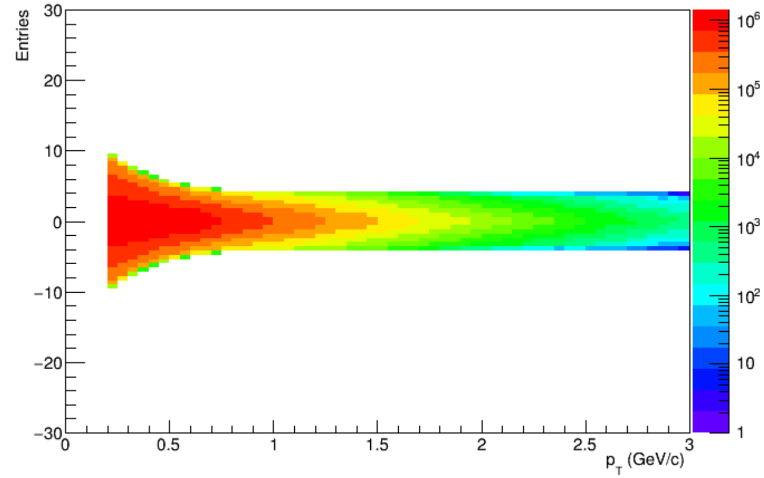
- BiBi@9.2, UrQMD
- Track selections: n-hits > 39, $|\eta| < 1$, $|DCA_{x,y,z}| < 3\sigma$
- dPhi





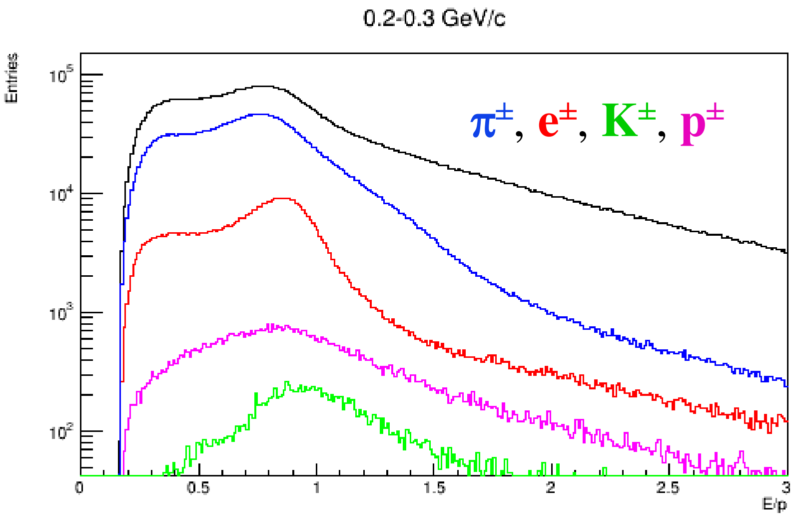
- dZed

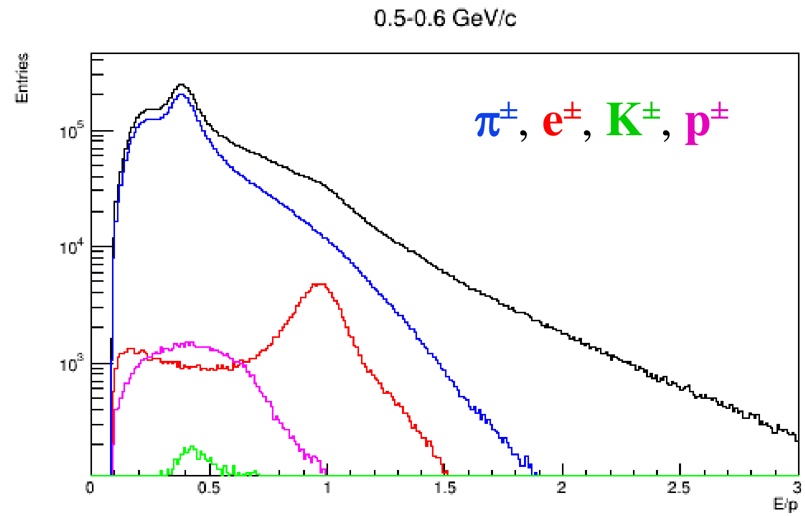




E/p in different p_T bins

- BiBi@9.2, UrQMD
- Track selections: n-hits > 39, $|\eta| < 1$, $|DCA_{x,y,z}| < 3\sigma$
- 2σ matching in dPhi and dZed





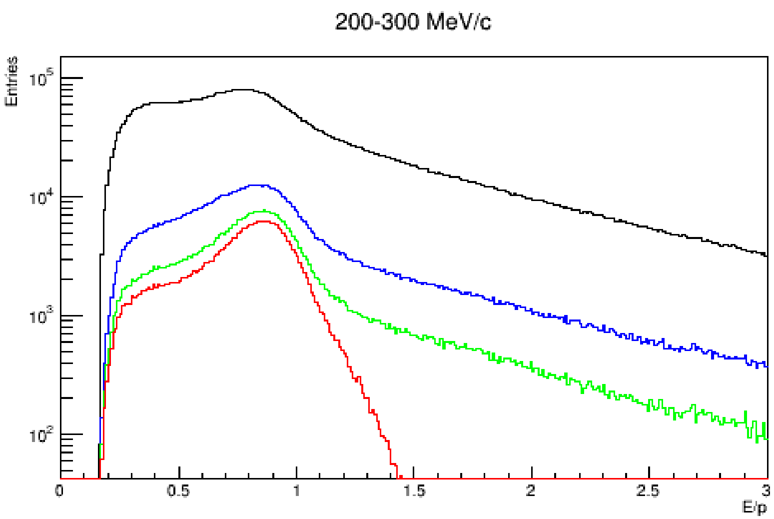
- At low momentum, E/p ~ 1 for most of charged hadrons → low efficiency of e/h
- At high momentum, E/p ~ 1 is for electrons only, high hadron rejection power

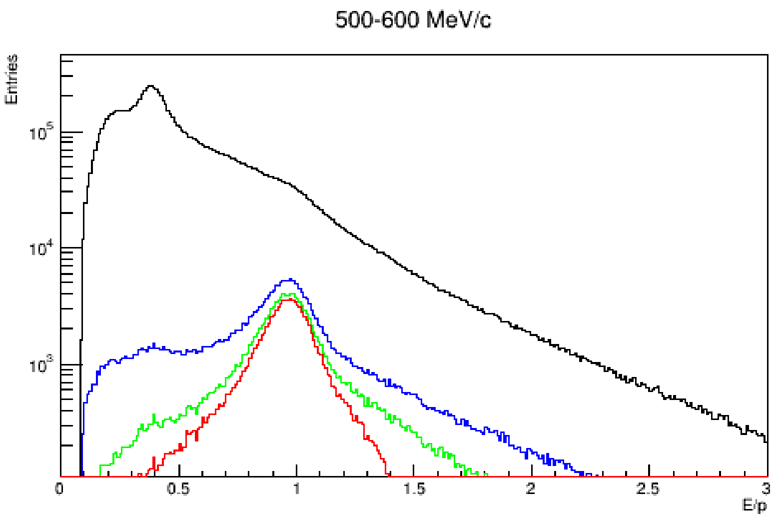
E/p rejection power

- BiBi@9.2, UrQMD

- Track selections: n-hits > 39, $|\eta| < 1$, $|DCA_{x,y,z}| < 3\sigma$
- 2σ matching in dPhi and dZed
- eID in the TPC && TOF

eID in TPC, **eID in TPC && TOF**, **true $e^{\pm}$**



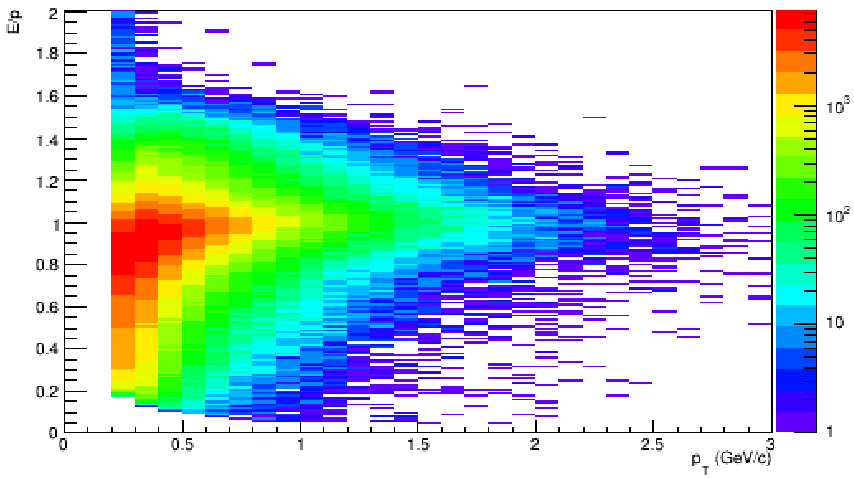


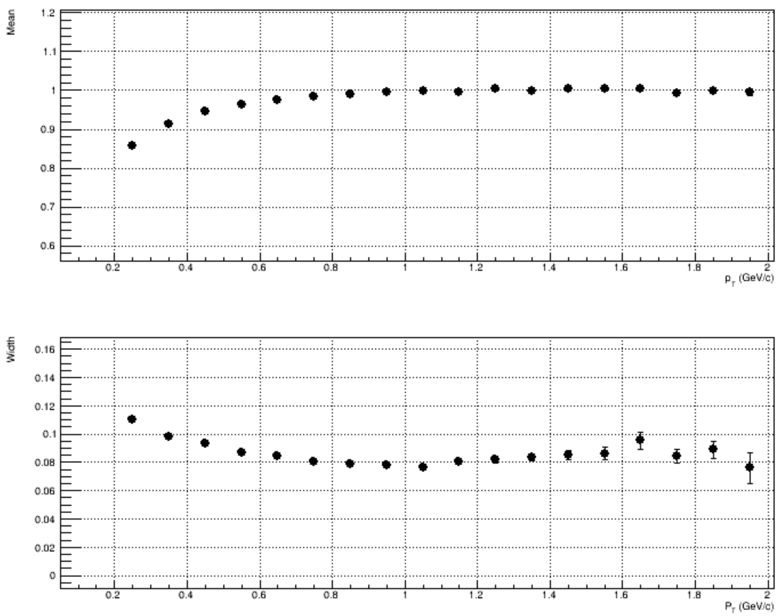
- E/p peaks are observed at low and high momentum
- Electron peak is shifted from E/p ~ 1 and is not Gaussian at low momentum
- Width of the peaks is momentum dependent

E/p rejection power

- BiBi@9.2, UrQMD

- Track selections: n-hits > 39, $|\eta| < 1$, $|DCA_{x,y,z}| < 3\sigma$
- 2σ matching in dPhi and dZed
- eID in the TPC && TOF





- E/p eID selections are effective at p_T > 300-500 MeV/c
- E/p peak position and width will be used for global calibration of the ECAL

Conclusions

- E/p can be used for the absolute calibration of the ECAL after track identification in the TPC/TOF
- E/p electron ID selections are effective at p_T > 300-400 MeV/c