Systematic study of the spectral shape dependence on neutral meson reconstruction in the ALICE EMCal

Alena Lösle

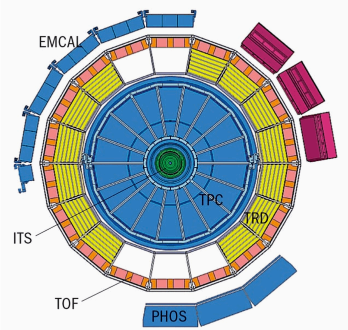
Supervisors: Dr. Jason Kamin
Dr. Constantin Loizides

Student Session

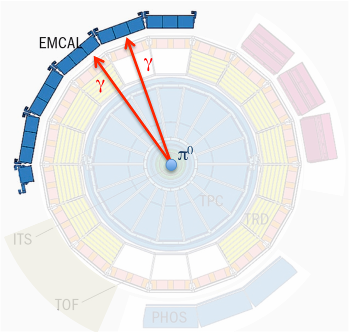
Outline

- π^0 reconstruction in ALICE EMCal
- our ToyMC analysis
- spectral shape dependence of reconstructed π^0
- summary

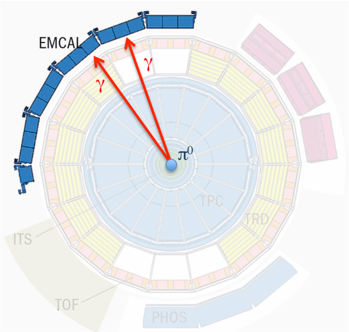
π^0 reconstruction in ALICE EMcal



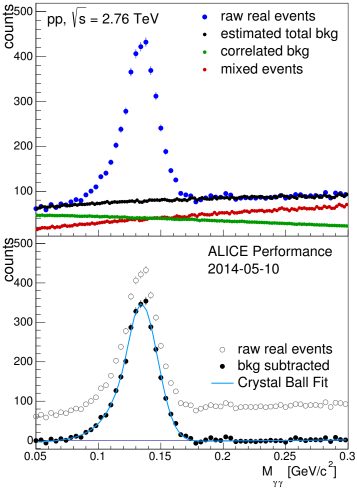
π^0 reconstruction in ALICE EMcal



π^0 reconstruction in ALICE EMcal



p_T range: 1.8 - 2.0 GeV



π^0 reconstruction:

- combine all $\gamma\gamma$ pairs in EMCal
- estimate and subtract background
- fit mass peak

Our ToyMC analysis

$\pi^0 \to \gamma\gamma$ using TGenPhaseSpace

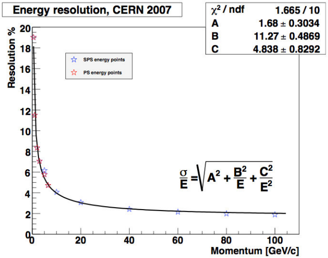
- **energy smearing:** **position smearing:**

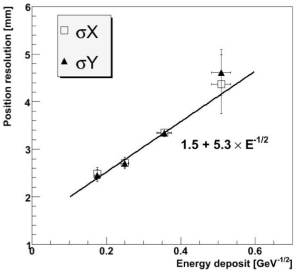
$$\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E} \qquad \sigma_P = a + \frac{b}{\sqrt{E}}$$

A constant term: detector geometry

B sampling term: counting statistics $\propto$ signal

C noise term: pedestal due to electronics





http://arxiv.org/abs/1008.0413

Our ToyMC analysis

$\pi^0 \rightarrow \gamma\gamma$ using TGenPhaseSpace

- **energy smearing:**

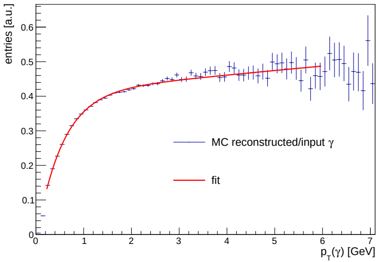
$$\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$$

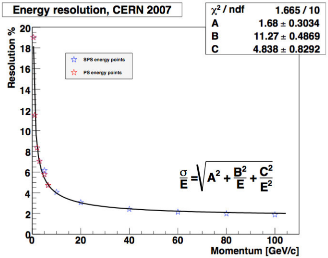
position smearing:

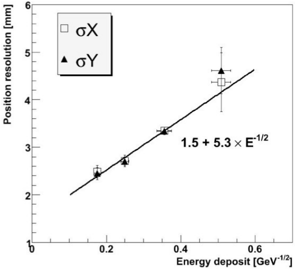
$$\sigma_P = a + \frac{b}{\sqrt{E}}$$

- kinematic cut $p_T^{\gamma} > 0.2$ GeV

- apply single photon efficiency







http://arxiv.org/abs/1008.0413

Our ToyMC analysis

$\pi^0 \to \gamma\gamma$ using TGenPhaseSpace

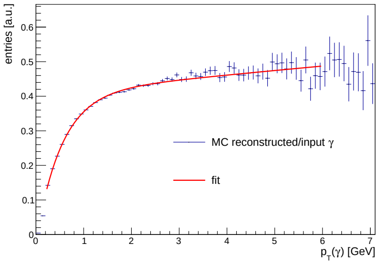
- **energy smearing:**

$$\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$$

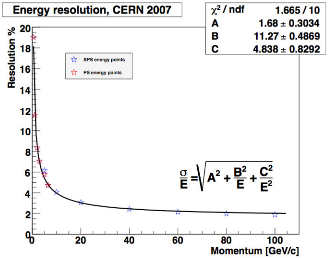
position smearing:

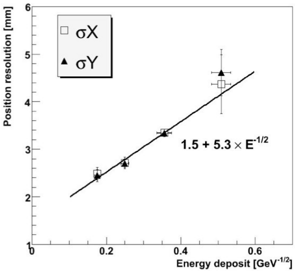
$$\sigma_P = a + \frac{b}{\sqrt{E}}$$

- kinematic cut $p_T^\gamma > 0.2$ GeV
- apply single photon efficiency



- reconstruct π^0 by adding $\gamma\gamma$ pairs





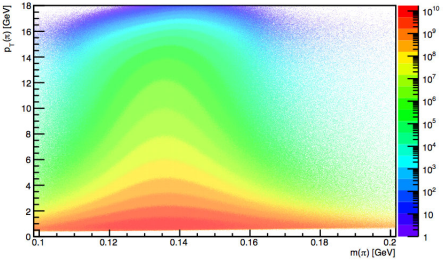
http://arxiv.org/abs/1008.0413

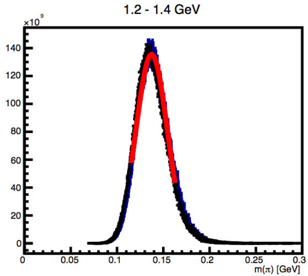
Mass distribution of reconstructed π^0

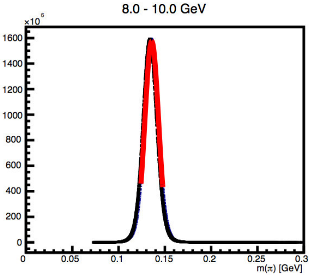
- mass distribution as function of pion p_T
$\rightarrow$ project different p_T slices

- fitting of mass peaks for different p_T slices with gaussian

$\rightarrow$ get μ and σ as function of pion p_T

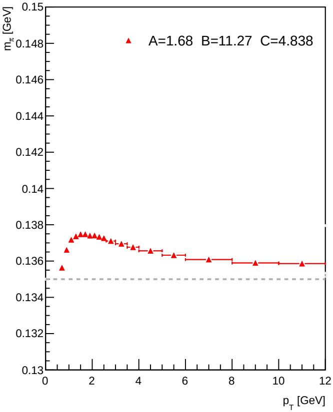


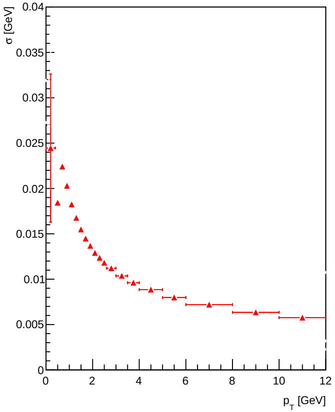




Effect of energy smearing on reconstructed π^0

using EMCal parametrization for mass distribution: $\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$

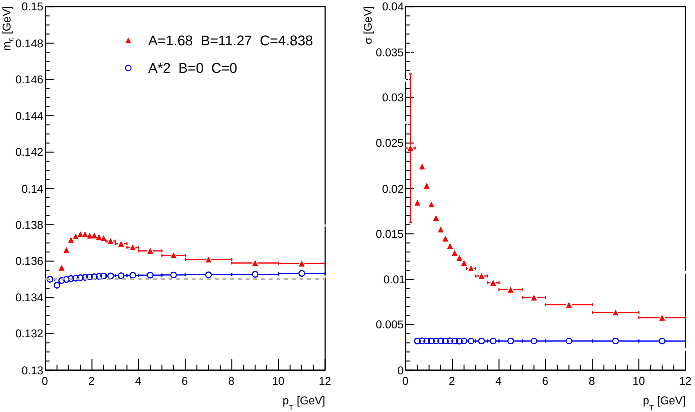




Effect of energy smearing on reconstructed π^0

using EMCal parametrization for mass distribution: $\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$

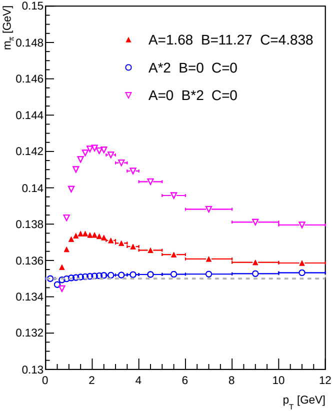
constant term

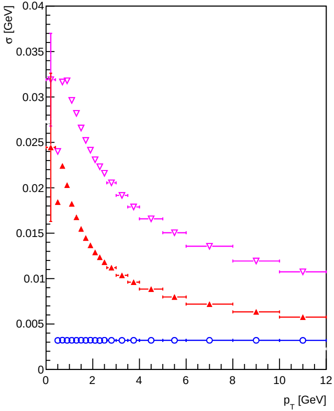


Effect of energy smearing on reconstructed π^0

using EMCal parametrization for mass distribution: $\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$

constant term   sampling term

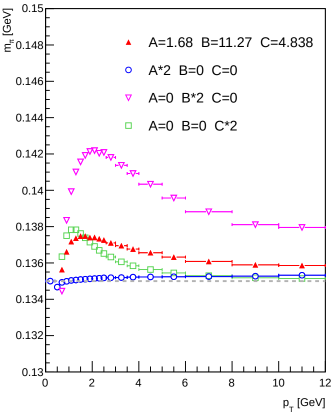




Effect of energy smearing on reconstructed π^0

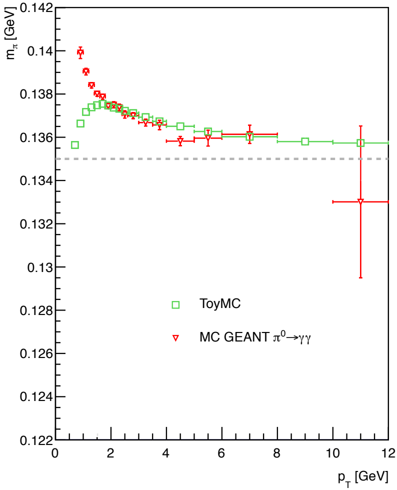
using EMCal parametrization for mass distribution: $\frac{\sigma_E}{E} = A \oplus \frac{B}{\sqrt{E}} \oplus \frac{C}{E}$

constant term sampling term noise term



Compare fitted π^0 mass distribution

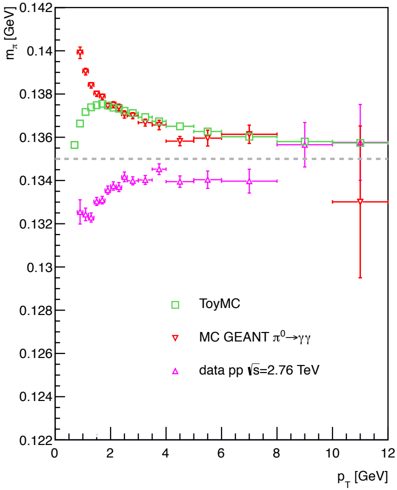
compare to GEANT simulation:



→ ToyMC and MC GEANT fit reasonably well

Compare fitted π^0 mass distribution

compare to GEANT simulation and data:

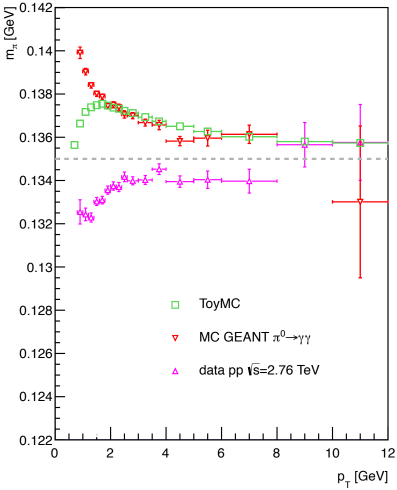


→ ToyMC and MC GEANT fit reasonably well

→ different shapes for ToyMC and data

Compare fitted π^0 mass distribution

compare to GEANT simulation and data:



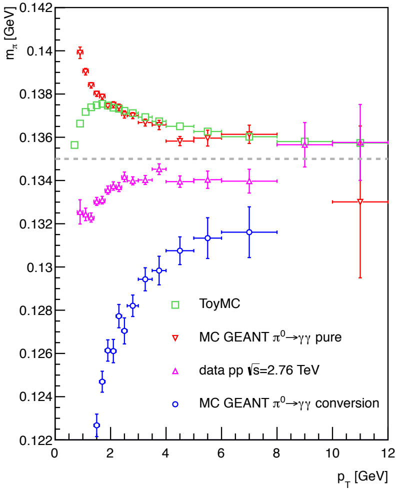
→ ToyMC and MC GEANT fit reasonably well

→ different shapes for ToyMC and data

→ merged conversion photons!

Compare fitted π^0 mass distribution

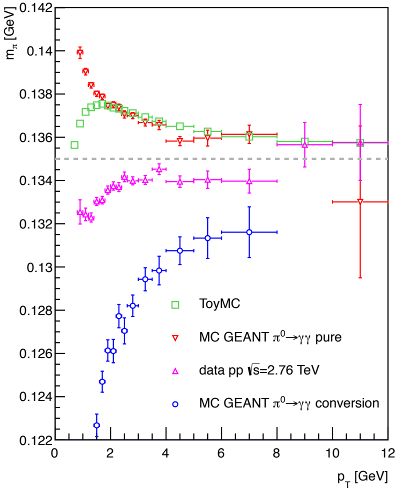
compare to GEANT simulation and data:



- $\rightarrow$ ToyMC and MC GEANT fit reasonably well
- $\rightarrow$ different shapes for ToyMC and data
- $\rightarrow$ merged conversion photons!

Compare fitted π^0 mass distribution

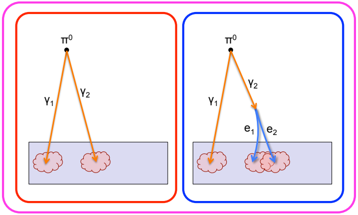
compare to GEANT simulation and data:



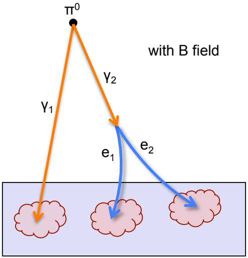
→ ToyMC and MC GEANT fit reasonably well

→ different shapes for ToyMC and data

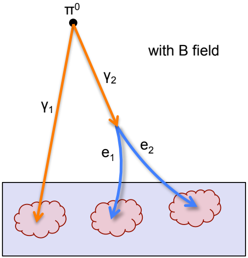
→ merged conversion photons!

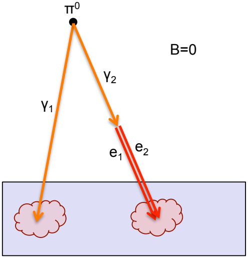


How to deal with non merging e^+e^-



How to deal with non merging e^+e^-

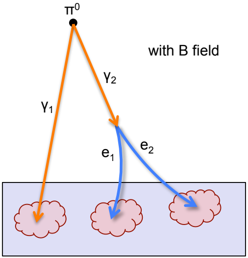


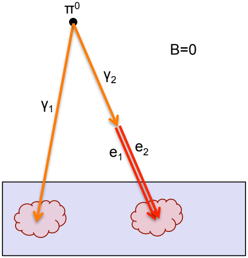


- conversion decreases reconstructed π^0 yield
- without magnetic field: conversion e^+e^- merge in one EMCal cluster
- compare yield with and without magnetic field

$\rightarrow$ estimate material budget

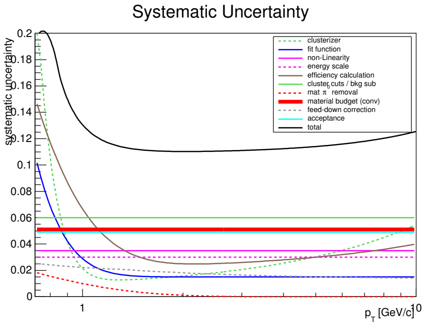
How to deal with non merging e^+e^-





- conversion decreases reconstructed π^0 yield
- without magnetic field: conversion e^+e^- merge in one EMCal cluster
- compare yield with and without magnetic field

$\rightarrow$ estimate material budget



Summary

- ToyMC looking at $\pi^0 \rightarrow \gamma\gamma$ (no photon conversion)

- comparison of π^0 mass position from ToyMC to data and GEANT

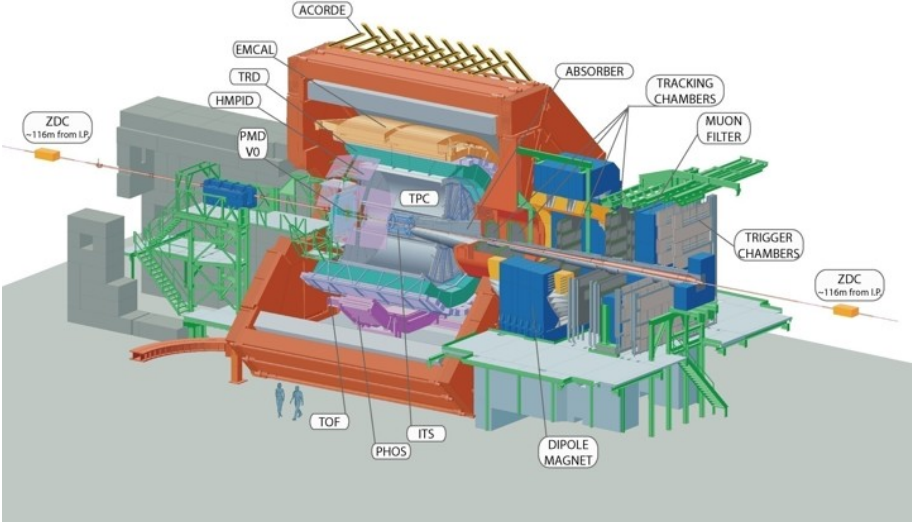
$\rightarrow$ take converison photons into account

- to deal with conversion e^+e^- that don't overlap in EMCal:

$\rightarrow$ compare π^0 yield with and without magnetic field

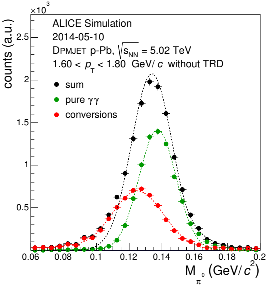
$\rightarrow$ material budget estimation in front of EMCal

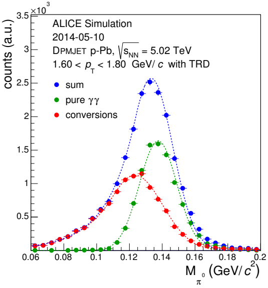
Back-up

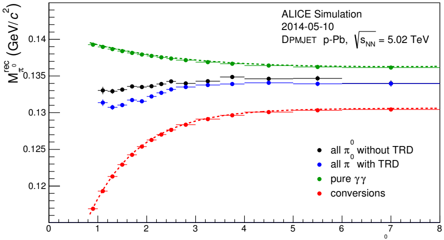

ACORDE
EMCAL
TRD
HMPID
ZDC
~116m from I.P.
PMD
V0
TPC
ABSORBER
TRACKING
CHAMBERS
MUON
FILTER
TRIGGER
CHAMBERS
ZDC
~116m from I.P.
TOF
ITS
PHOS
DIPOLE
MAGNET

Back-up

Example: Impact of TRD on π^0 reconstruction



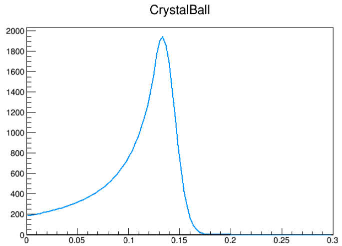




Crystal Ball function:

Crystal Ball

```
Double_t CrystalBall(Double_t *x, Double_t *par) {

  // The Crystal Ball shape is a Gaussian that is 'connected' to an exponential tail at
  // 'alpha' sigma of the Gaussian. The sign determines if it happens on the left or
  // right side. The 'n' parameter controls the slope of the exponential part.
  // typical par limits:   1.0 < alpha < 5.0   and   0.5 < n < 100.0

  Double_t alpha = par[0];
  Double_t     n = par[1];
  Double_t meanx = par[2];
  Double_t sigma = par[3];
  Double_t    nn = par[4];
  Double_t a = TMath::Power((n/TMath::Abs(alpha)), n) * TMath::Exp(-0.5*alpha*alpha);
  Double_t b = n/TMath::Abs(alpha) - TMath::Abs(alpha);
  Double_t arg = (x[0] - meanx)/sigma;
  Double_t fitval = 0;
  if (arg > -1.0*alpha) {
    fitval = nn * TMath::Exp(-0.5*arg*arg);
  } else {
    fitval = nn * a * TMath::Power((b-arg), (-1*n));
  }
  return fitval;
}



// here's just the lefthand part:

TF1 *f_cr = new TF1("f_cr",
  "[4]*TMath::Power(([1]/TMath::Abs([0])), [1])*
   TMath::Exp(-0.5*[0]*[0])*TMath::Power((([1]/TMath::Abs([0]) -
   TMath::Abs([0]))-(x - [2])/[3]),(-1*[1]))", 0.01,0.13)
```



remaining issue: understanding the width distributionn