中国科学技术大学
University of Science and Technology of China

Energy calibration for EMC

Dong Liu

University of Science and Technology of China

Dec. 29, 2020

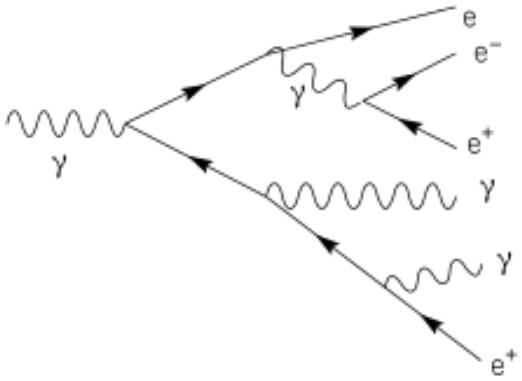
PANDA China Collaboration Meeting

Outline

- EMC
- Calibration algorithm
  - Sample preparation
  - Input and output
  - Fitting
  - Iteration
- Calibration Result
- Calibration with single cluster
- Multi-thread implementation
- Summary and plan

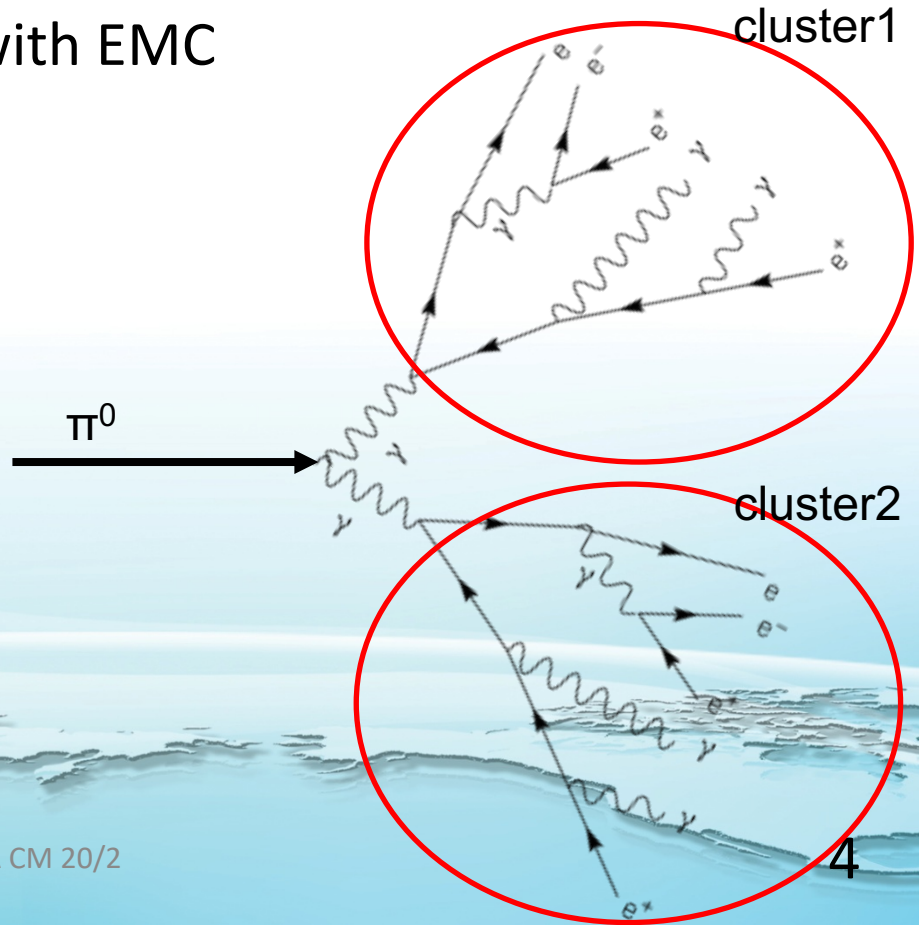
EMC

- Functionality
  - Energy measurement of γ, e and part of hadrons
  - Position measurement
  - Separation of γ/e and hadrons
  - Shower shape measurement
- Calibration
  - Detection unit uniformity
  - Leakage
  - Preshower
  - Light yield non-uniformity
  - Energy response non-linearity
  - Electronics



Calibration

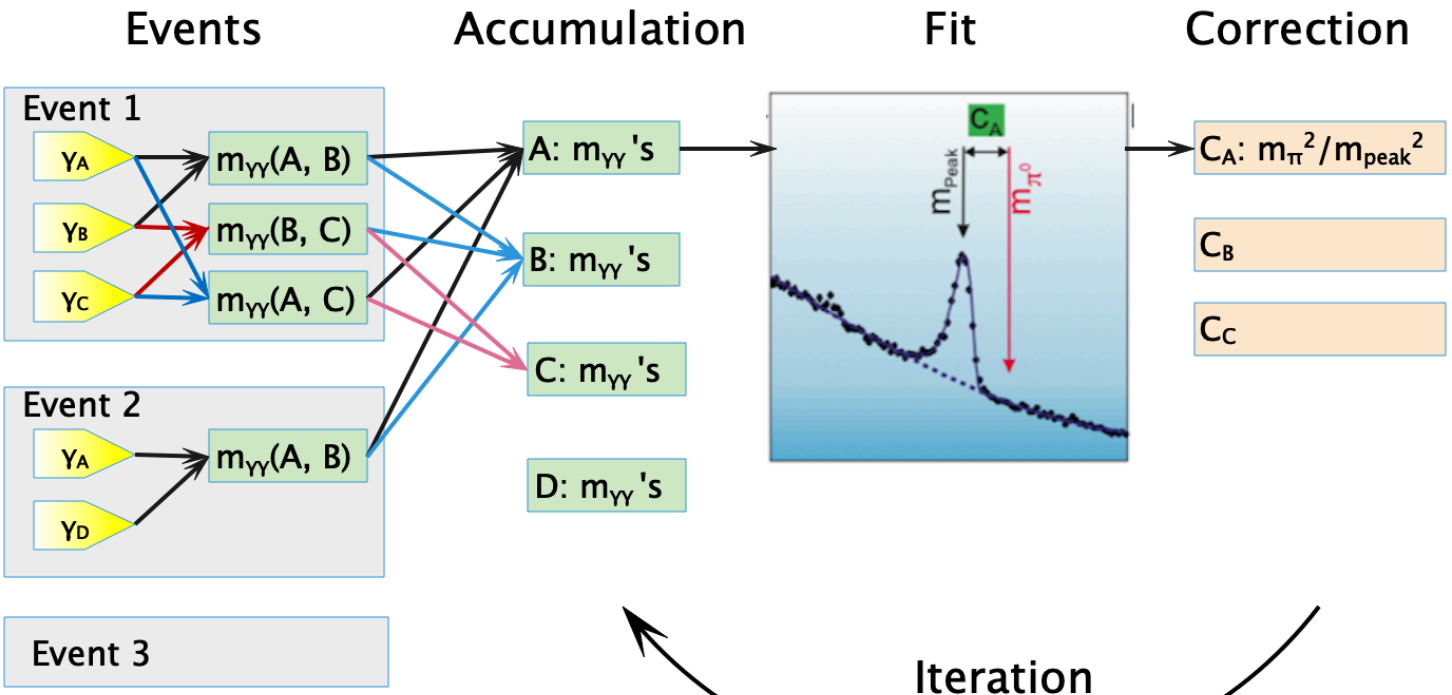
- Calibration sample
  - Abundance
  - Accurate quantity as standard value
  - Well understand interaction with EMC
- π^0 / η → γγ
  - Abundantly produced
  - Known mass and small width
  - Energy coverage



Calibration algorithm

- Based on π^0 mass

From Bernhard Roth & Marc Pelizaeus, Ruhr University



$$E_\gamma = \sum_i E_i$$

$$m_{\gamma\gamma} = \sqrt{2E_{\gamma 1}E_{\gamma 2}(1-\cos\theta)}$$

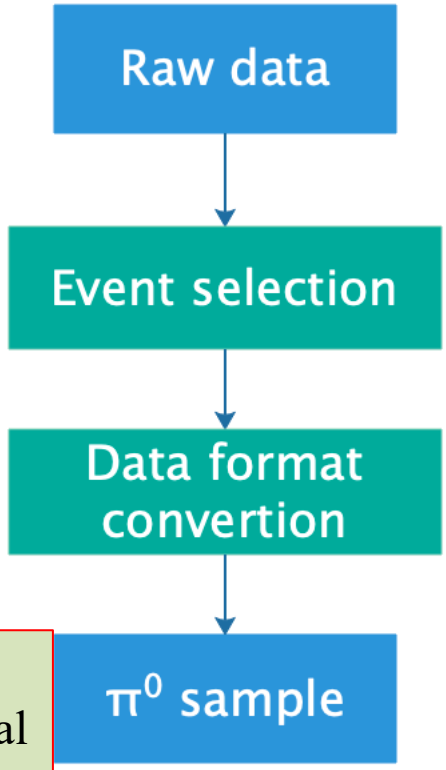
$$E'_i = C_i E_i$$

$$E'_\gamma = \sum_I E'_I$$

$$m'_{\gamma\gamma} = \sqrt{2E'_{\gamma 1}E'_{\gamma 2}(1-\cos\theta')}$$

Calibration algorithm

- Implementation
  - Sample preparation
    - Raw data from beam test, physical run or MC simulation
    - π^0 candidates are saved in root file in a specific format (simplified data)



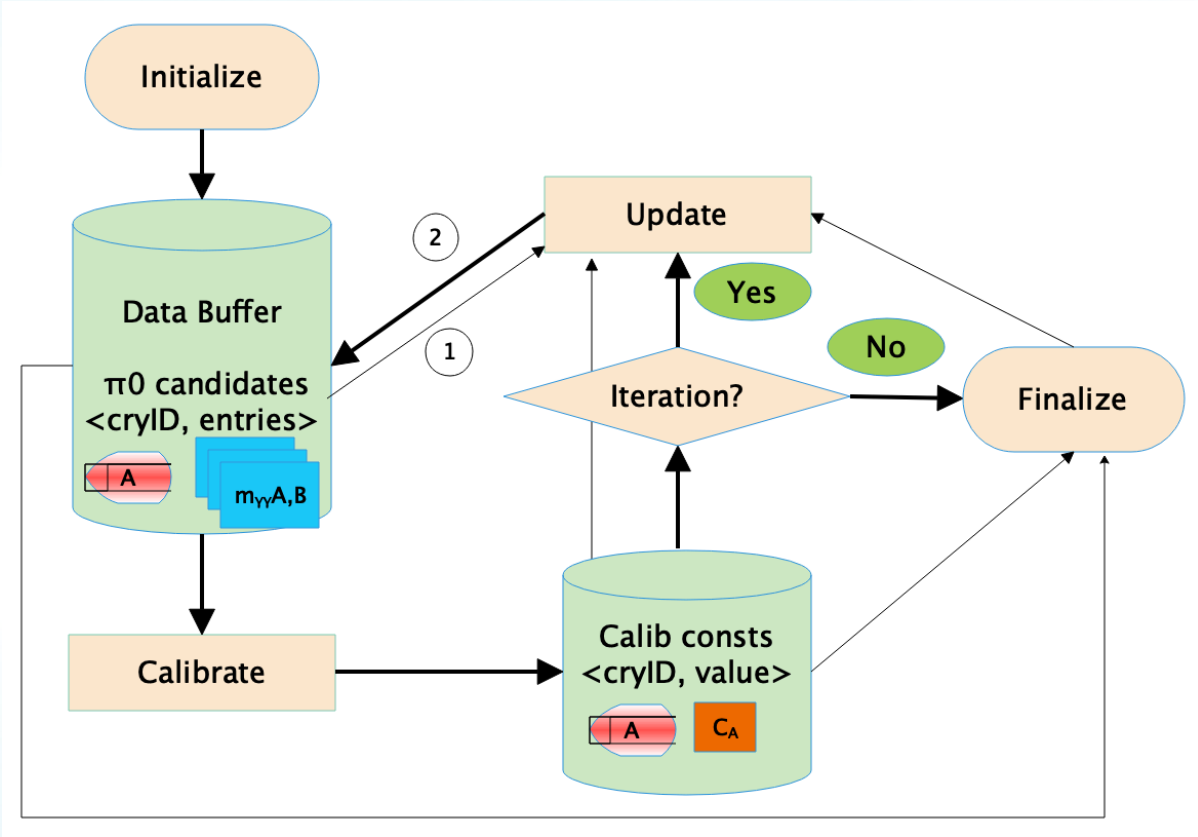

```
class entry{ // pi0 candidate
  float m_gg;  // invariant mass of two γ
  int cpnr2;
  float fraction;
  float fraction2;
  float angle; // opening angle

  std::vector<hit> this_bump; // gamma
  std::vector<hit> associated_bump;
}
```

```
class hit{ // single crystal
  int cpnr; // index of a crystal
  float E_dep;
  float X;
  float Y;
  float Z;
}
```

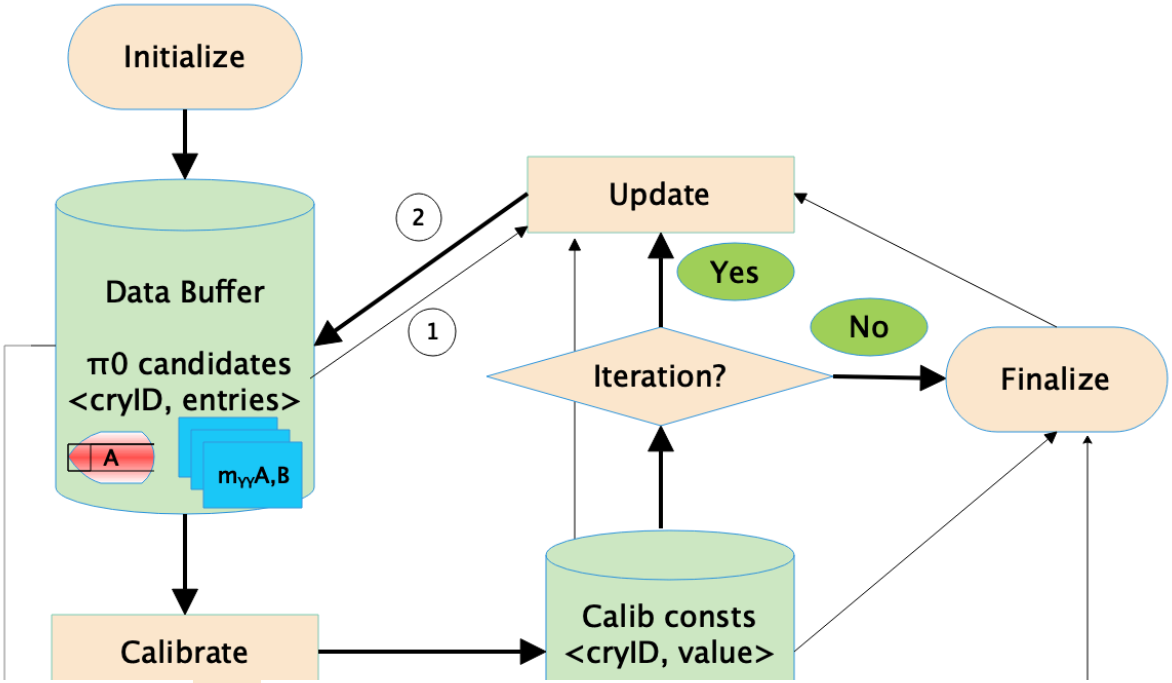
Calibration algorithm

- Implementation
  - Calibration
    - Initialize
    - Calibrate
    - Update
    - Iteration
    - Finalize



Calibration algorithm

- Implementation
  - Calibration
    - Initialize
      - Set parameters
      - Load/Initialize calibration map
      - Load crystal map
      - Load π^0 samples

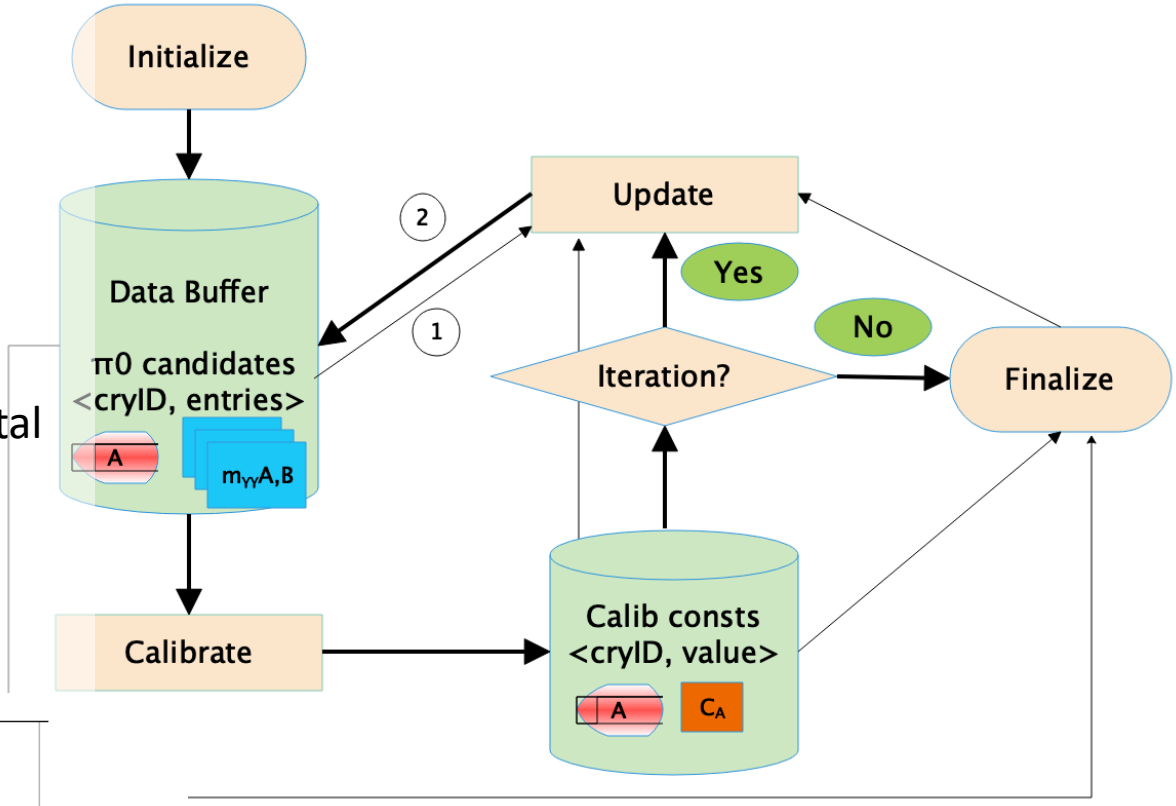


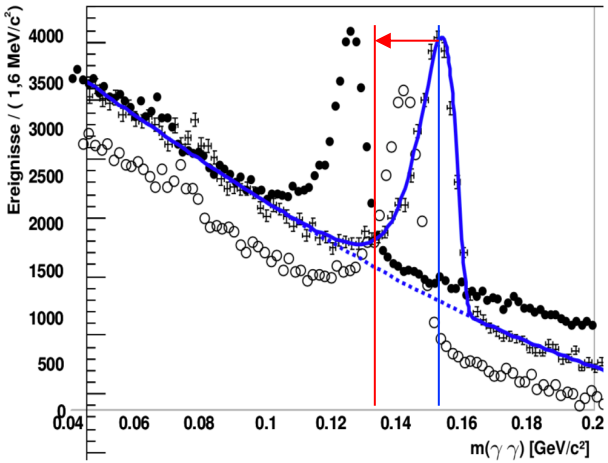
Load π^0 samples
- From root file
  - Readable format with ROOT
  - Ability to deal with large amounts of data
- Cache sample in memory
  - High performance for later usage
  - 5M π^0 need ~1G memory

Load π^0 samples (old strategy)
- From database
  - Multi-threaded design, locally or network connected threads
- Cache in database

Calibration algorithm

- Implementation
  - Calibration
    - Initialize
    - Calibrate
      - Load sample from cache for each crystal
      - Fit to $m_i(\gamma\gamma)$
      - Determine C_i
      - Save in calib_map



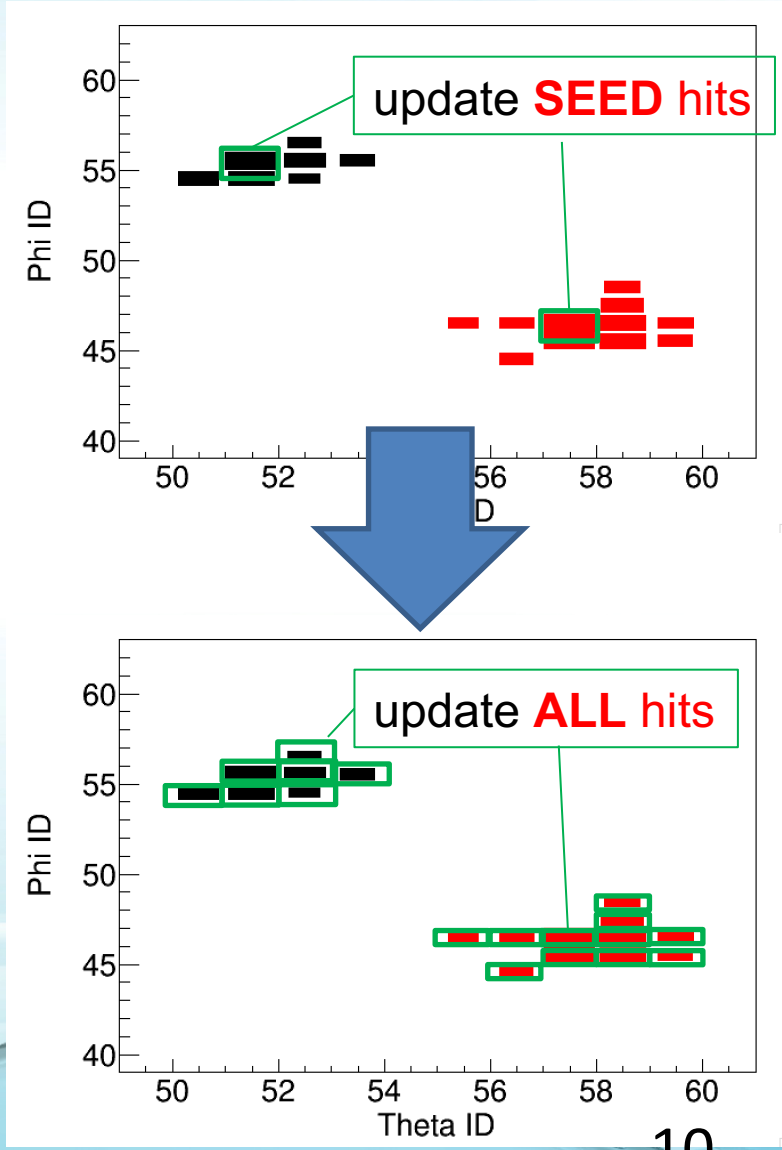


$$c_i = \frac{m^2_{\pi,PDG}}{\left\langle m^2_{\pi,i} \right\rangle},$$

Calibration algorithm

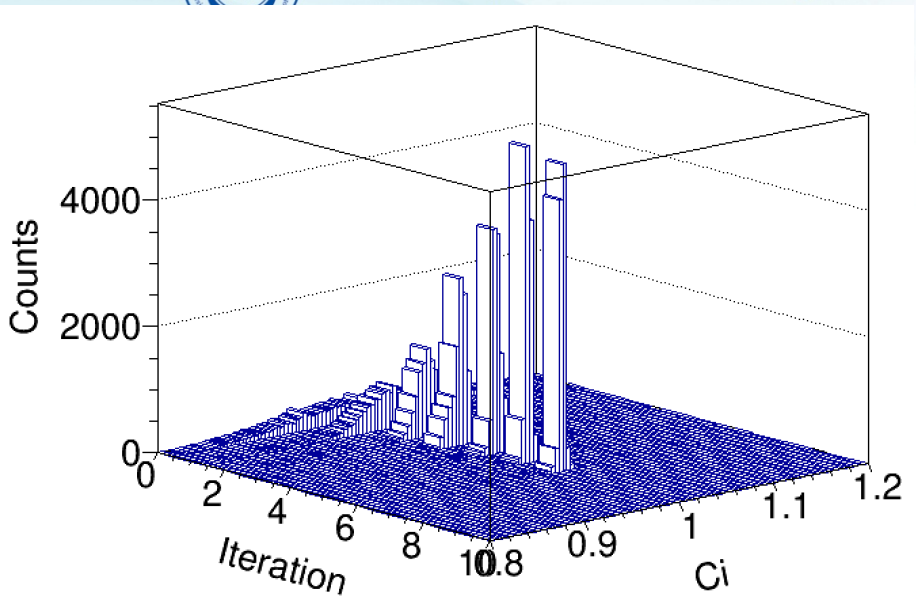
- Implementation
  - Calibration
    - Initialize
    - Calibrate
    - Update
      - Load sample from cache for each crystal
      - Update hit energy
      - Update E_γ, m(γγ) for each candidate

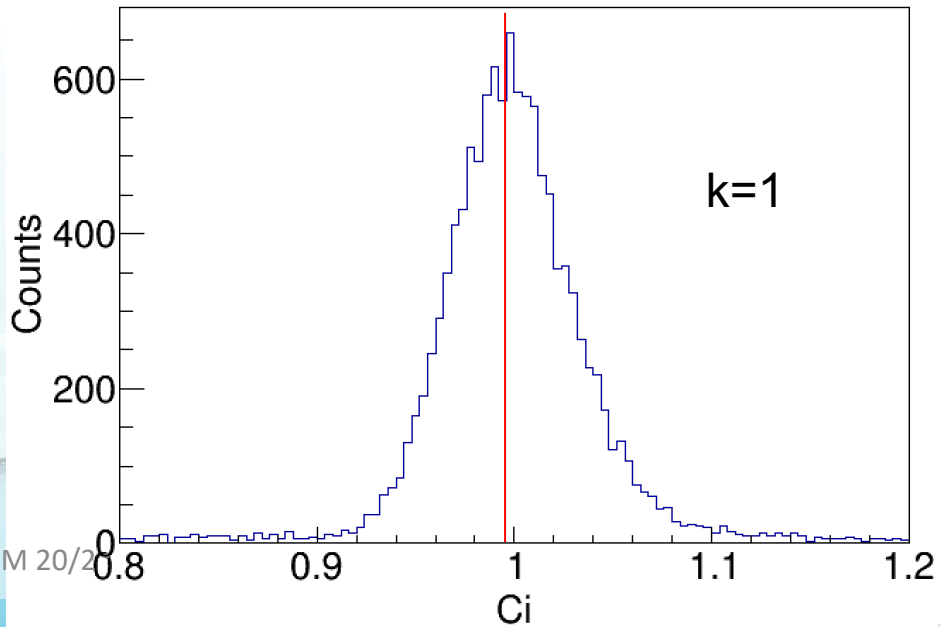
$$E_{Cluster} = \sum_{i=1}^{n} c_i A_i$$



Calibration algorithm

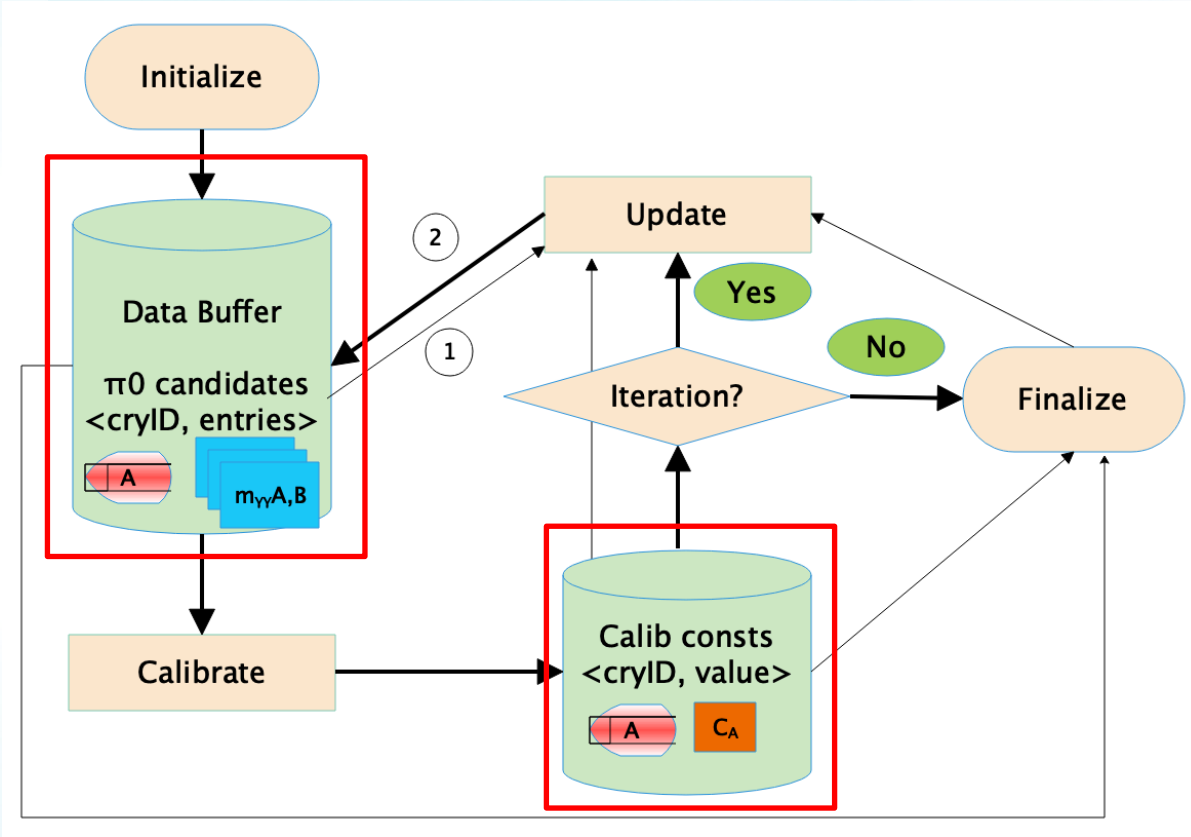
- Implementation
  - Calibration
    - Initialize
    - Calibrate
    - Update
    - Iteration
      - After iterations, $\lim_{k \to \infty} C_{i,k} = 1$
      - Iteration k < maxIteration
      - C_i close enough to 1
      - C_i closer to 1 than last iteration
      - Width of C_i narrower than last iteration





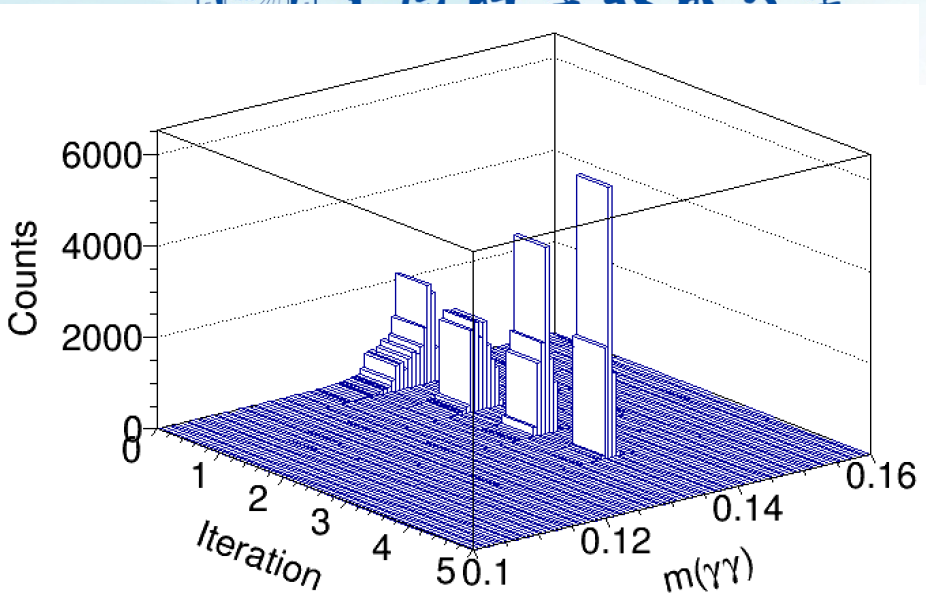
Calibration algorithm

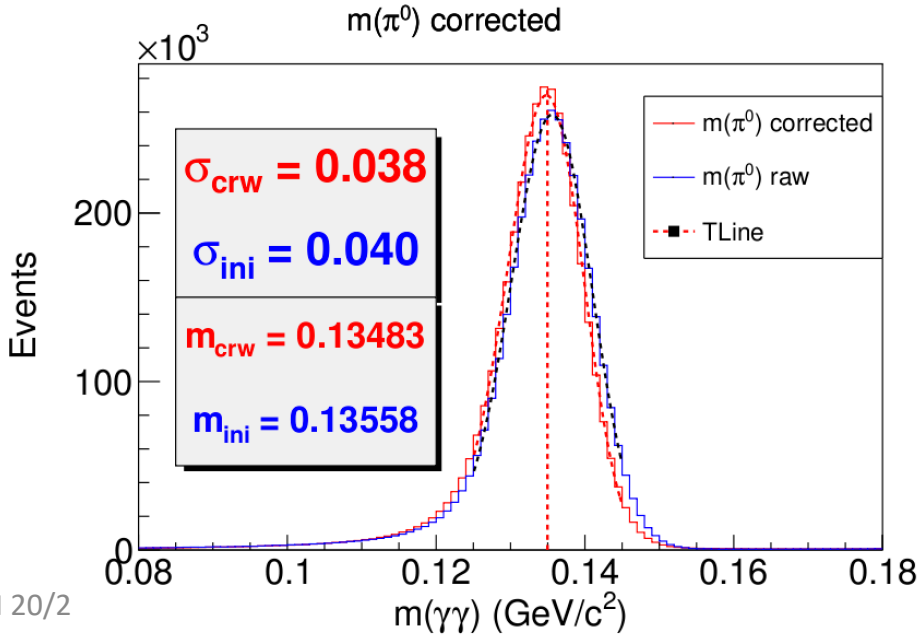
- Implementation
  - Calibration
    - Initialize
    - Calibrate
    - Update
    - Iteration
    - Finalize
      - Save calibration constants
      - Save calibrated candidates



Test

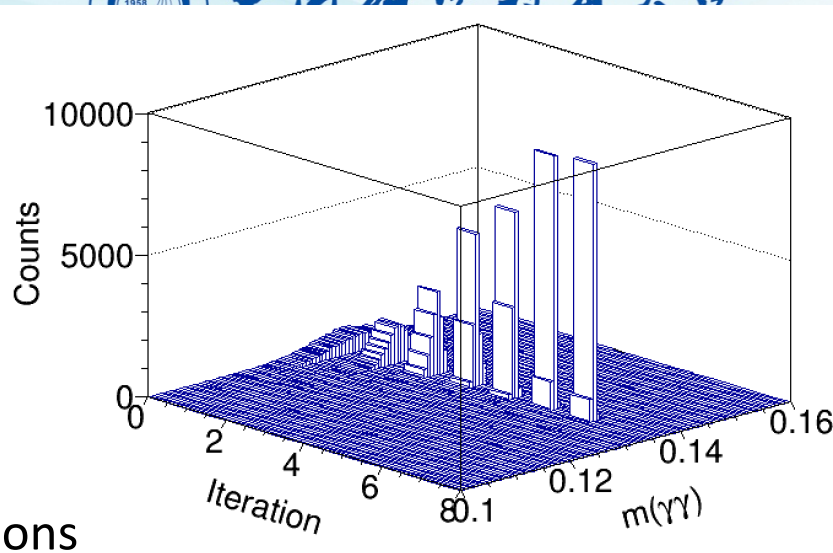
- MC sample as input
  - 1 GeV π^0, 5 M, ~400/crystal
  - Cover the barrel region
  - Uniform distribution
- Result check
  - Function
    - Novosibirsk function for π^0
    - 3rd Chebyshev for bkg
  - Performance
    - Memory usage, ~ 1G
    - Time usage, ~30min/iteration

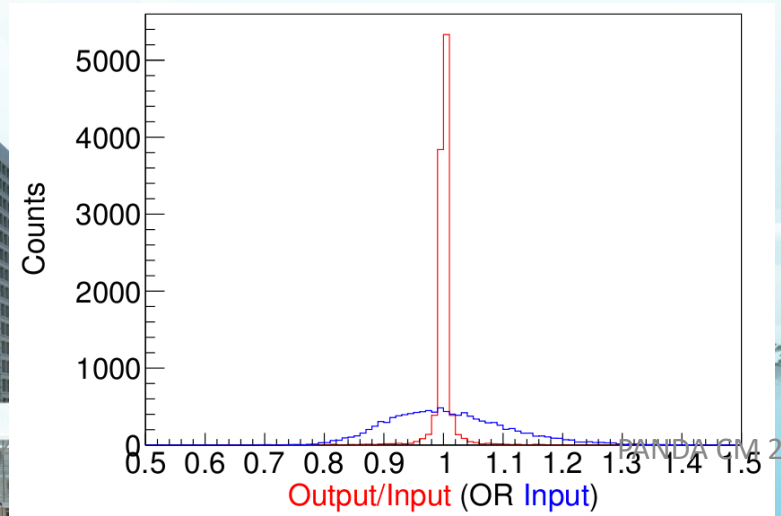


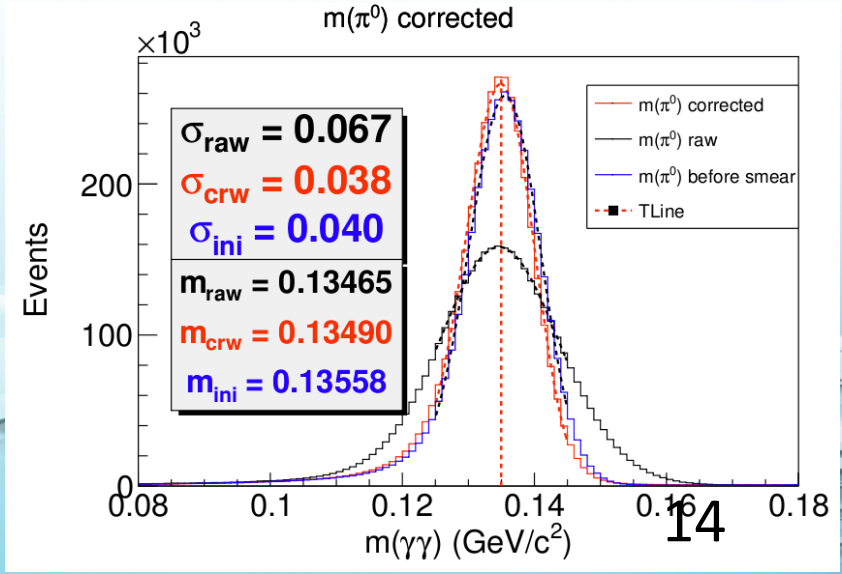


Validation

- Input/output check
  - Gain fluctuations are assigned to crystals, C_i=1 → C_i = Gaus(1, 0.1)
  - Hit energies are scaled with C_i
  - Use scaled sample to do calibration
  - Output consistent with input gain fluctuations
  - The calibration algorithm can recover gain fluctuations

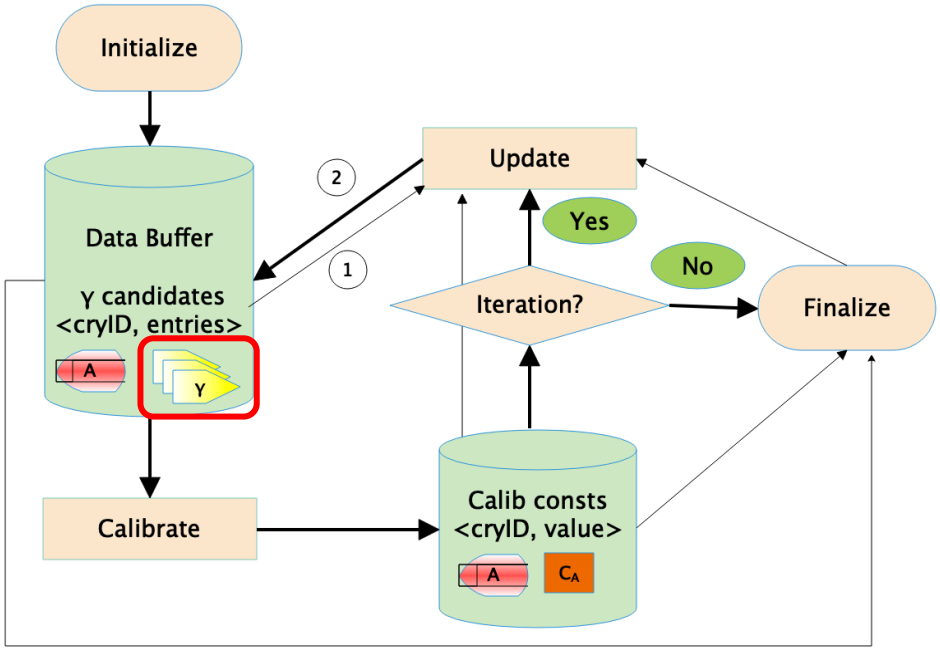






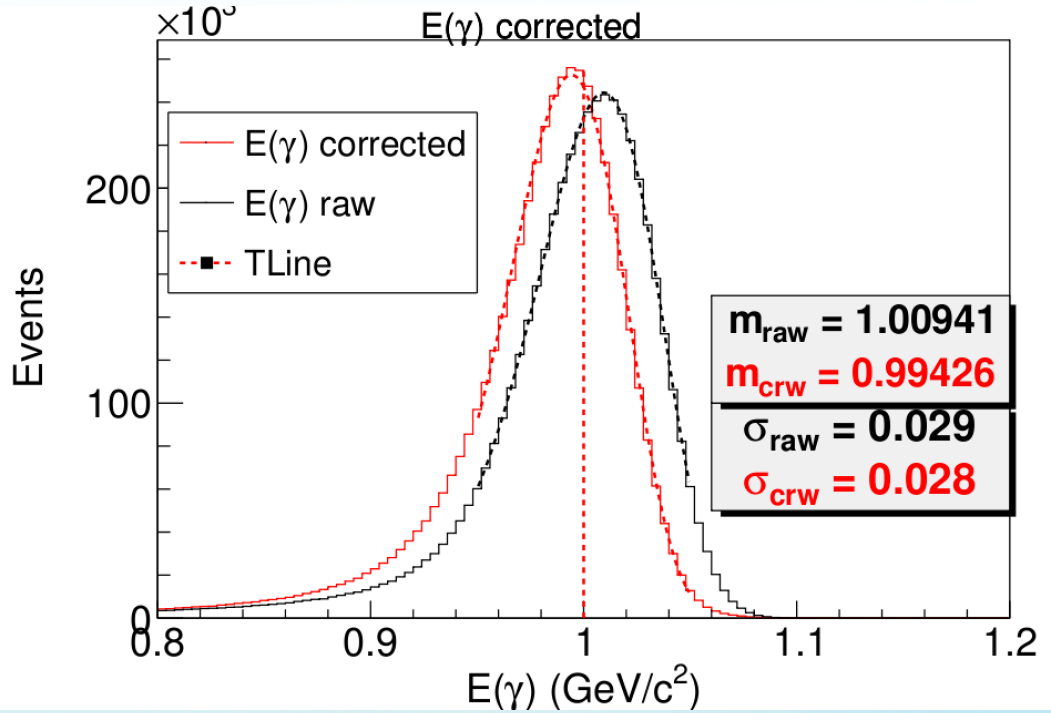
Single cluster calibration

- Single cluster mode
  - γ or e with specific energy
  - One cluster per candidate
- Implementation
  - Same data format,
    but only use one bump
  - Add a flag to mark sample type
  - Set the calibration goal from pi0 mass to the specific energy
  - Recalculate the energy of the cluster with constants



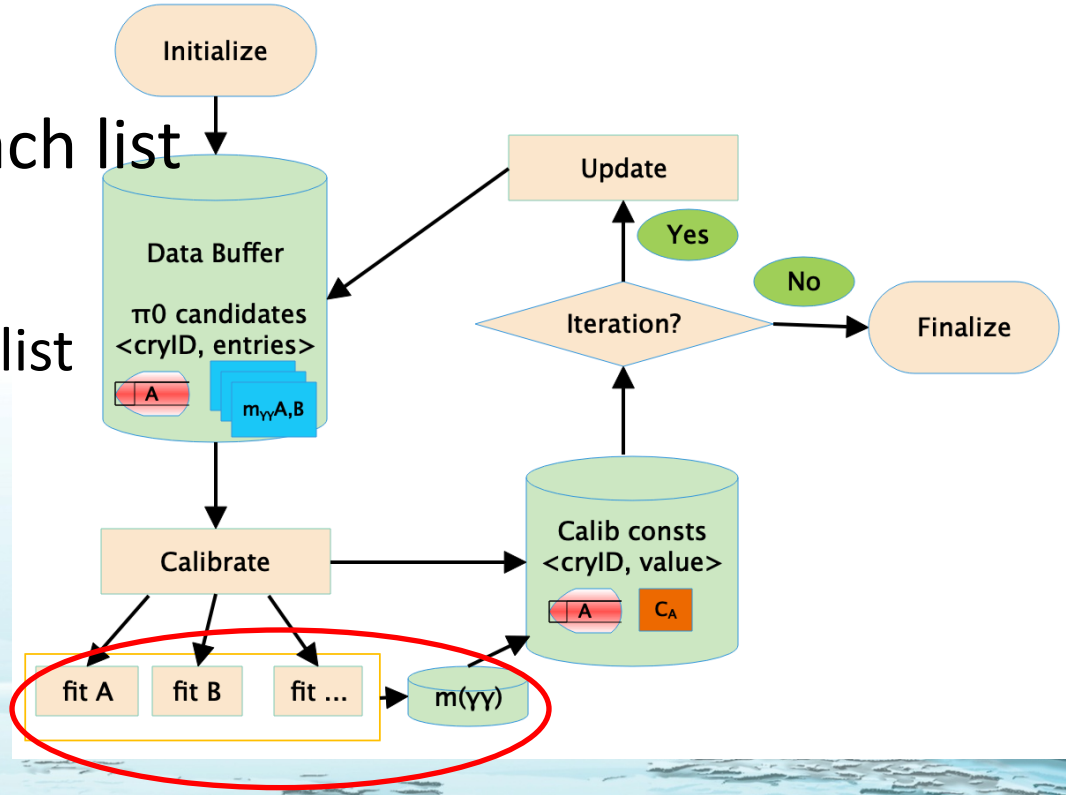
Test

- Gamma sample
  - 5M, 1GeV γ
  - Barrel region
- Result
  - Closer to 1 GeV
  - Slightly narrower
  - Works fine



Multi-thread

- In calibrate
  - std::thread
  - Split xtals to n lists
  - Create a thread for each list
  - In each thread
    - Fit m(γγ) for xtals in a list
    - Cache fit results
  - Check fit results
  - Calculate calib consts



Multi-thread

- In calibrat
  - Split xtal
  - Create a
  - In each t
    - Fit m(γ
    - Cache
  - Check fit
  - Calculate

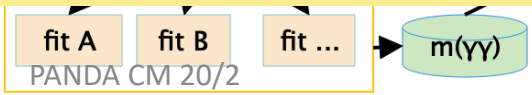
```
// assign xtal to threads
mapThreadXtals.clear();
int ixtal = 0;
int nxtal = xtal_index.size();
while (ixtal<nxtal) {
        for (int i=0; i<nthreads; i++) {
--- 11 行 : if (mapThreadXtals.find(i)==mapThreadXtals.end()) {----------
                int cpnr = xtal_index.at(ixtal);
                mapThreadXtals[i].push_back(cpnr);
                ixtal++;
                if (ixtal>=nxtal) break;
        }
}
// create multithreads
thread th[100]; // no more than 100 thread
for (int i=0; i<nthreads; i++) {
        th[i] = thread(getCinThread, i);
}
// wait for all threads
for (int i=0; i<nthreads; i++) {
        th[i].join();
}

// check
// deal with the result
for (int i=0; i<nthreads; i++) {
      cout<<"Info: size of calibMap "<<mapThreadErgebnis[i].size()<<endl;
-- 81 行 : for (int icry=0; icry<mapThreadXtals[i].size(); icry++) {----------
}
```

Create n lists

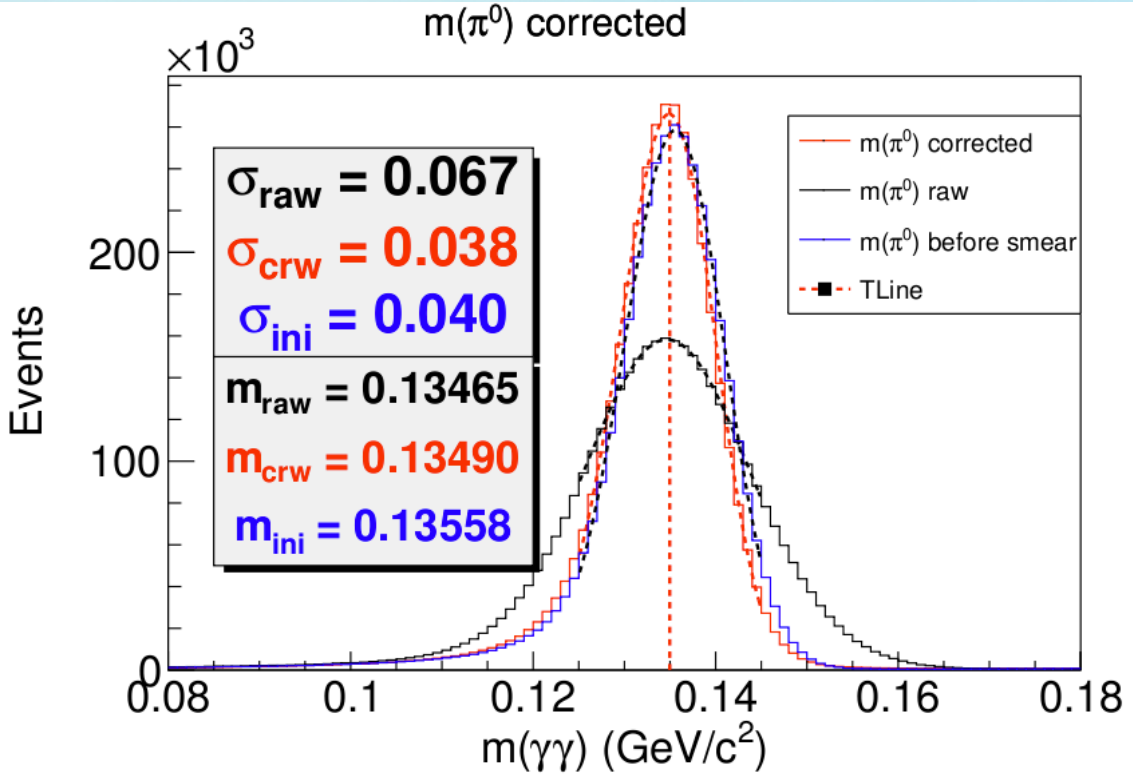
work in n threads

single thread to analysis the result



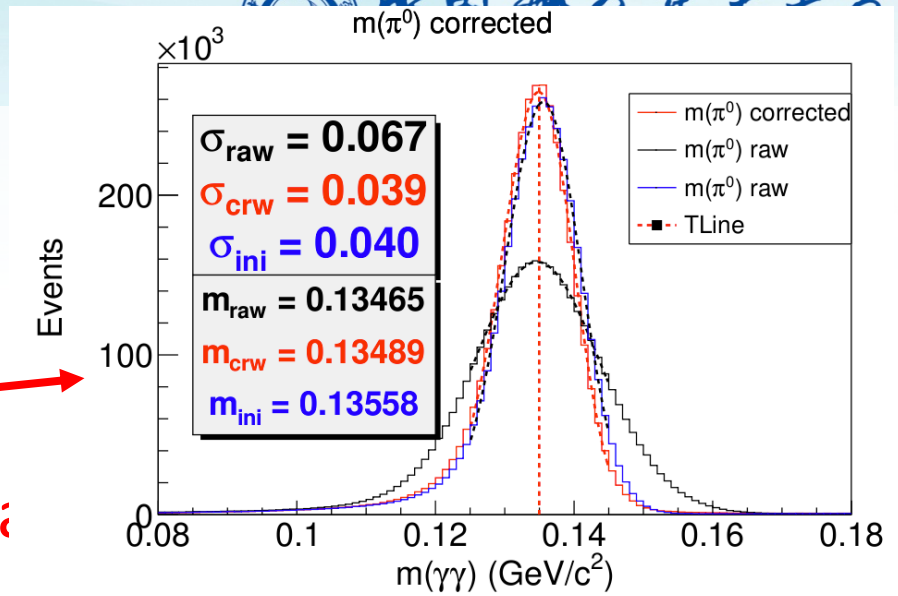
Multi-thread

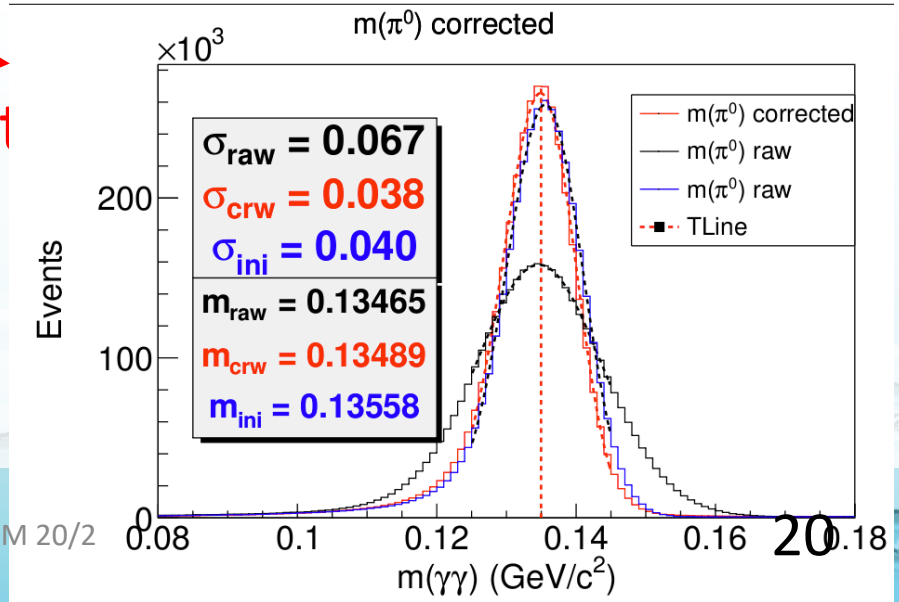
- Test
  - Single thread
    - Iterations: 7
    - Time: ~3h30min
  - 8 threads
    - Iterations: 7
    - Time: ~41min (5.9min/iteration)
  - Same result as single thread case
  - Improvement
    - Reduce 80% consuming time (1/5)



Multi-thread

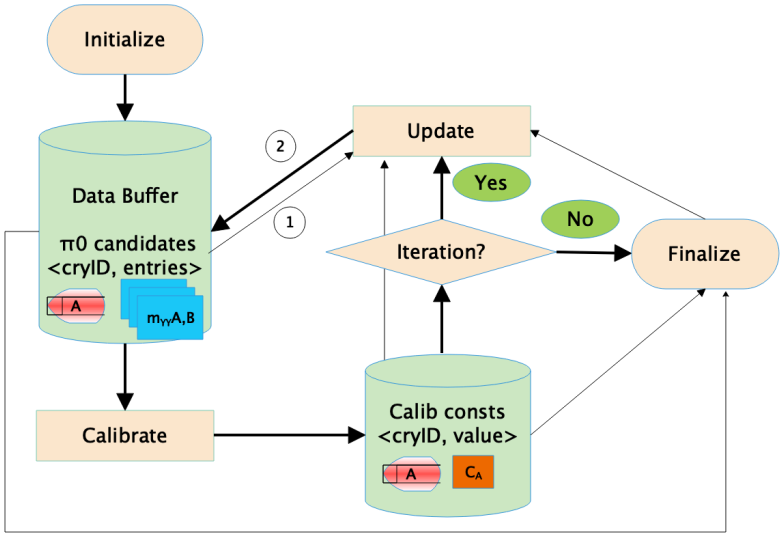
- More test
- 8 threads
  - Iterations: 11
  - Time: 47min (4.3min/itera
- 10 threads
  - Iterations: 8
  - Time: 29min (3.6min/iterat





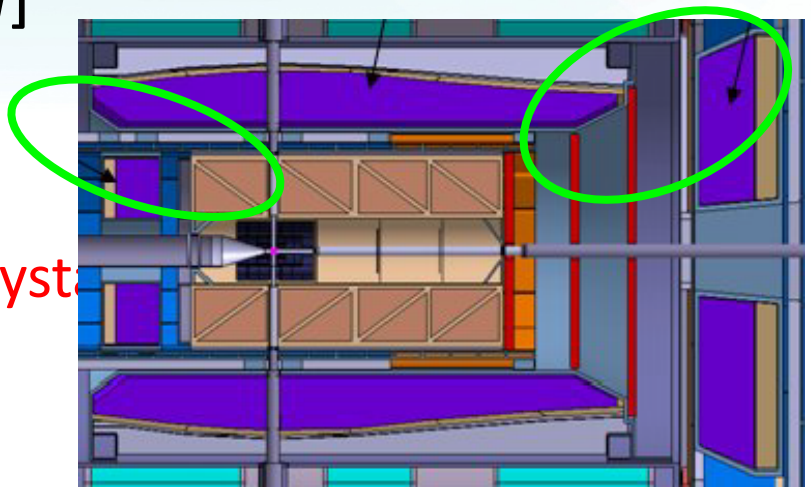
Summary

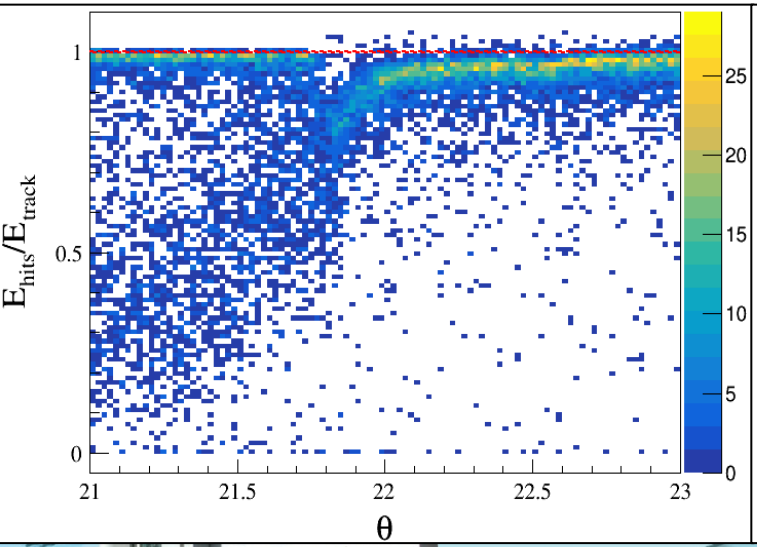
- Calibration algorithm
  - Calibration samples preparation
  - ROOT file as input, cached in memory
  - Calibrate
  - Update all hits
  - Validation
  - Multi-threads implementation and test
- Extend to single cluster case
  - Same data format
  - Change calibration goal
- Work to do
  - Optimize the algorithm
  - Test with MC closer to physics events

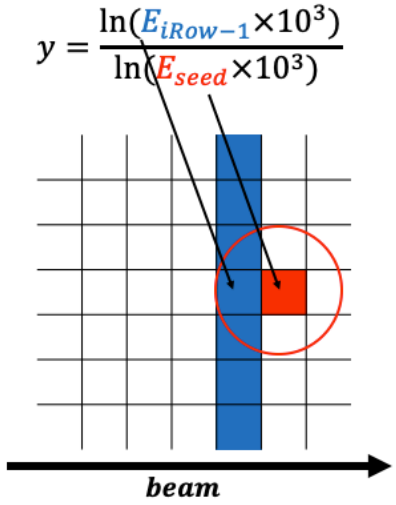


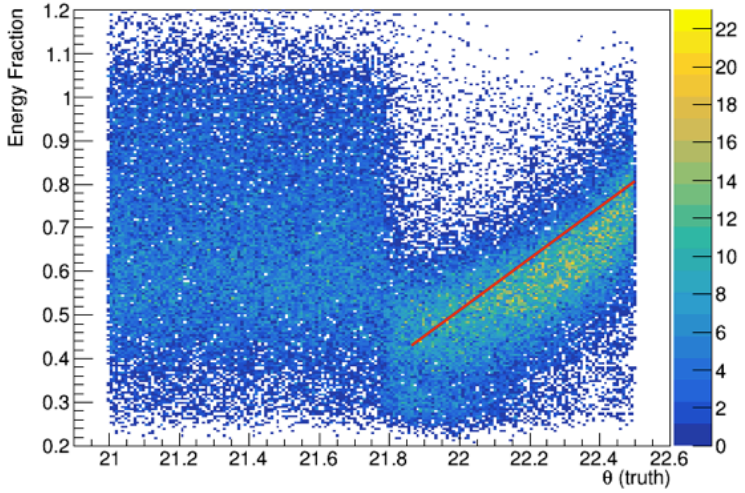
Further work

- Edge crystals calibration [*Hang Qi*]
  - Obvious leakage at edge
  - Leakage related to angle
  - Correction with info from nearby crysta
    - Ratio E_{i-1}/E_i?









Further work

- Edge crystals calibration [*Hang Qi*]
- Energy correction
  - Lateral correction, θ and ϕ related correction [*2007 Jinst 2 P04004*]
    - Variable definition, E1, E2
      - Compare two sides of seed crystal
    - Relation between Ecls Vs ln(E2/E1)
      - Both θ and ϕ direction

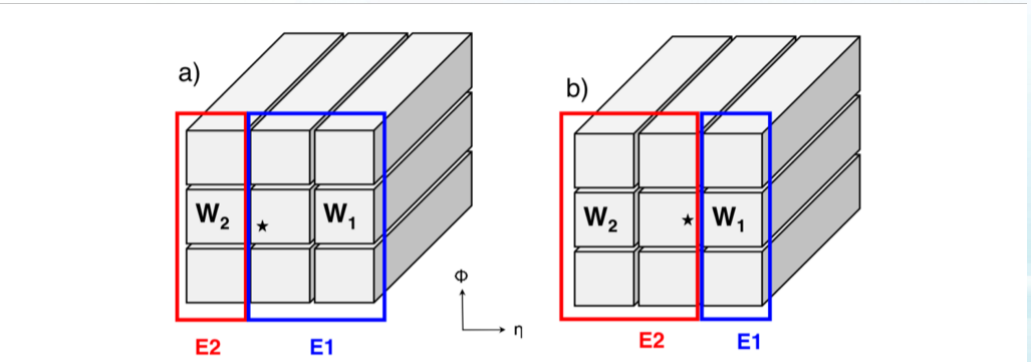


Figure 8. Definition of E1 and E2 when a) W1 < W2 and b) W1 > W2 (see text). The star represents the electron's incident position.

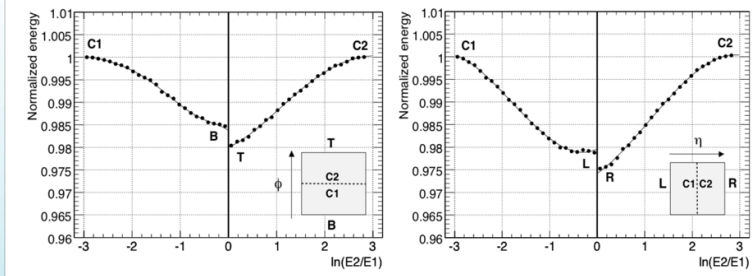


Figure 10. Normalized mean energy measured in the 3×3 array around crystal 204 versus ln(E2/E1) in the Φ (left) and η (right) directions. The curves are 3rd order polynomial functions fitted to the measured distributions, independently for positive and negative values of ln(E2/E1). The square panels represent the central crystal with various regions indicated: T (top), B (bottom), L (left), R (right) and C1 and C2 (just off centre on each side). The labels on the distributions indicate in which region the electrons were incident.

Further work

- Edge crystals calibration [*Hang Qi*]
- Energy correction
- Database
  - Sample and parameters accessed via database
- PandaRoot
  - A sub package in PANDAROOT