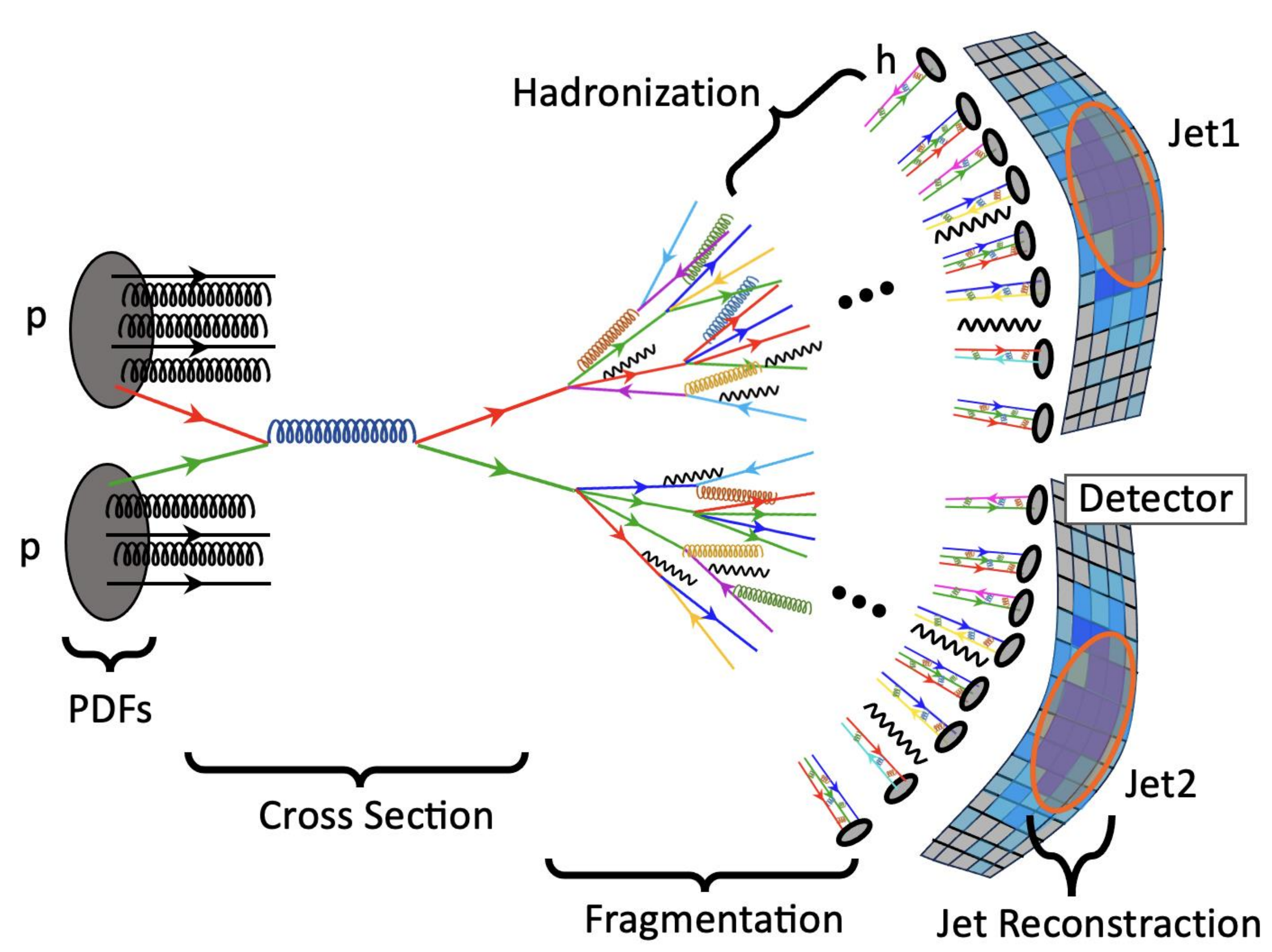
Experimental Jet Reconstruction Way

RIKEN Takuya Kumaoka

Main perpose:

Reconstruct original high-momentum parton.
→ Such parton fragments into many hadrons shower.

By comparing with simulations, we extract jet's and parton properties.



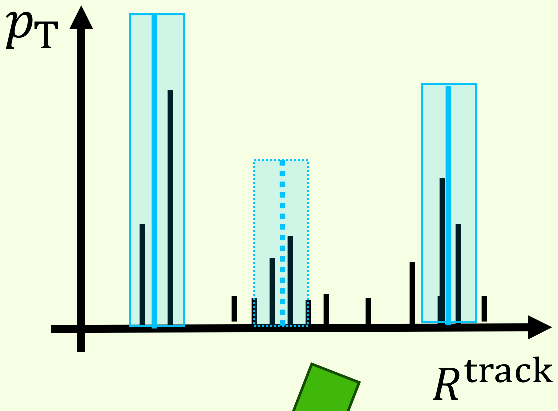
Brief Jet Analysis Flow

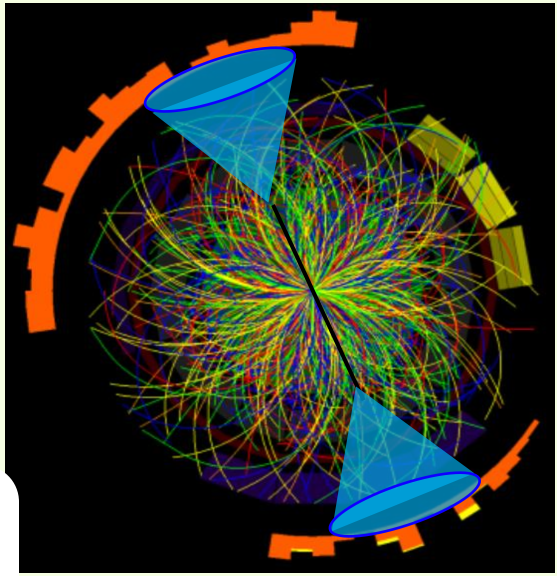
1. Track clustering algorithm (ex: anti-k_T)

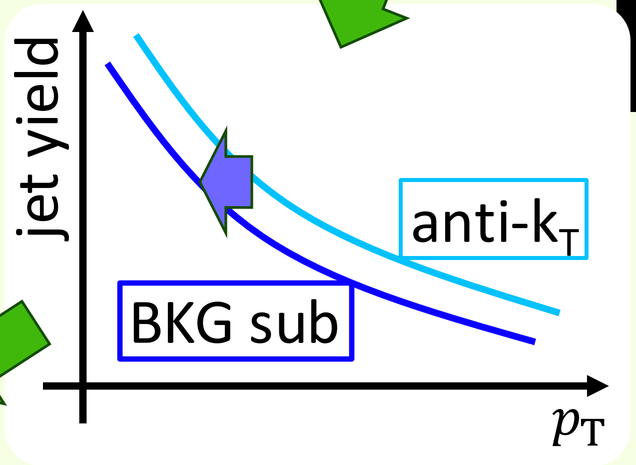
2. Background subtraction

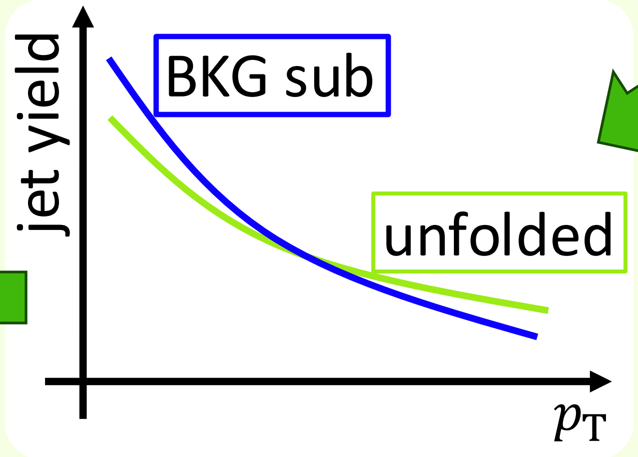
3. Unfolding

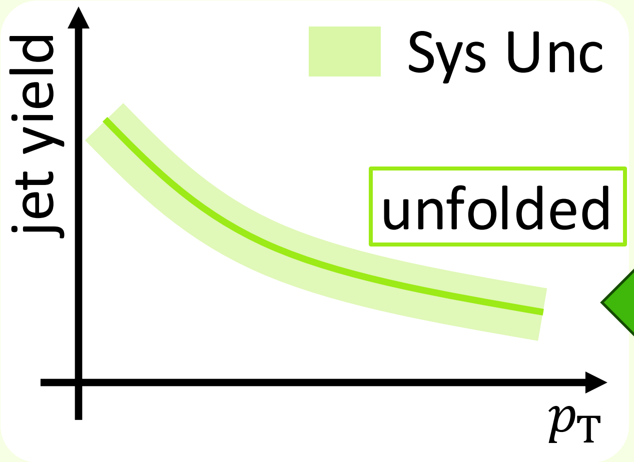
4. Systematic uncertainties estimation











1. Jet Reconstruction way

Major Jet Reconstraction Ways

There are three major jet reconstraction ways: anti- k_T, Cam/Aachen, k_T

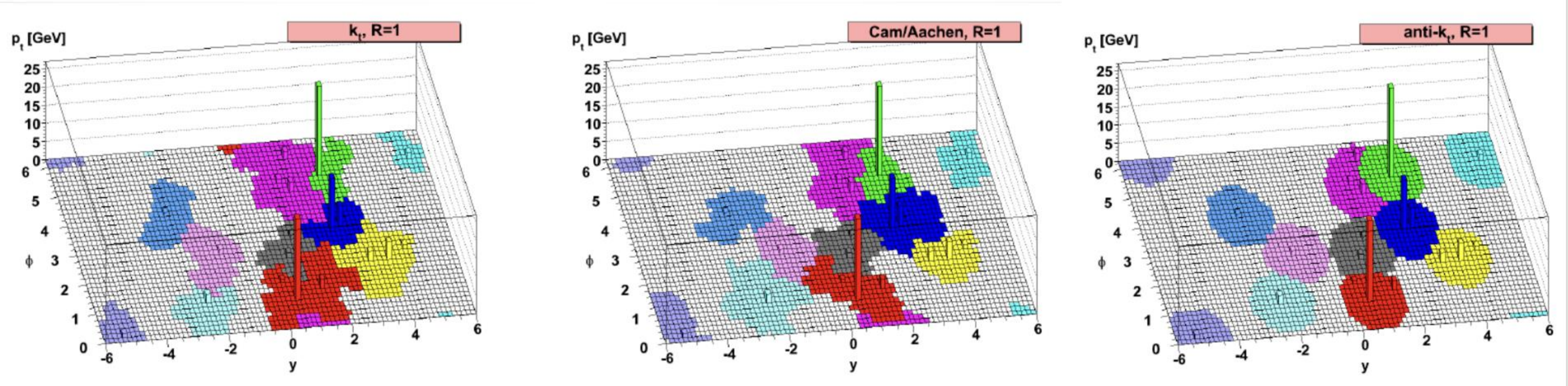
All algoriths make clusters to minimize d_{ij}

$$d_{ij} = \min(k_{ti}^{n}, k_{tj}^{n})\Delta R_{ij}^{2}/R^{2}$$, (anti-k_T: n = -1, Cam/Aachen: n = 0, k_T: n = 1)

$$R = \sqrt{\eta^2 + \phi^2}$$

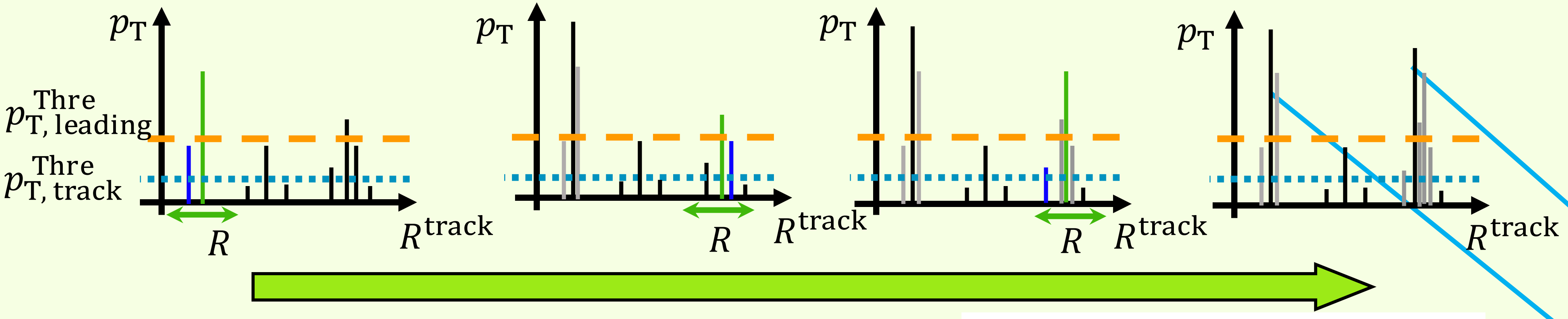
Summarize slide:
https://indico.cern.ch/event/367368/contributions/1783356/attachments/730376/1002150/HEPP2015_Presentation.pdf

Reconstracted clusters by each algorithm



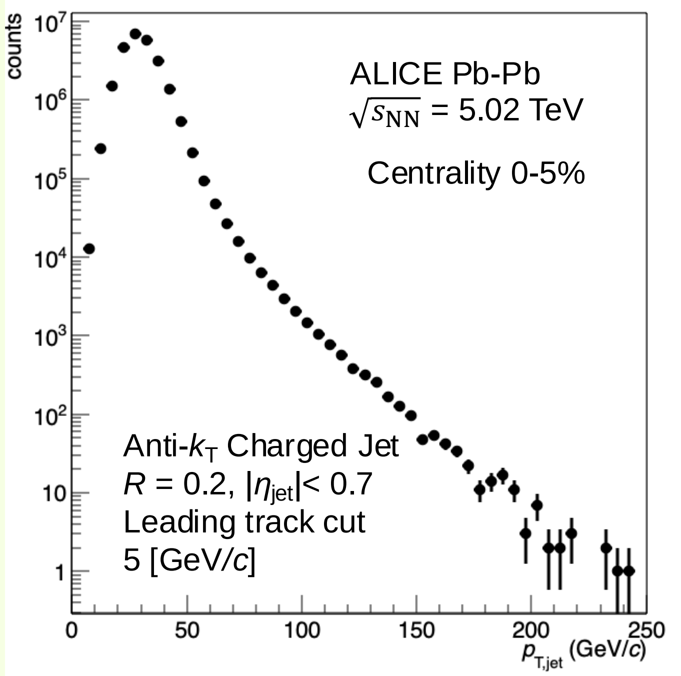
Anti-k_T signal jet reconstraction

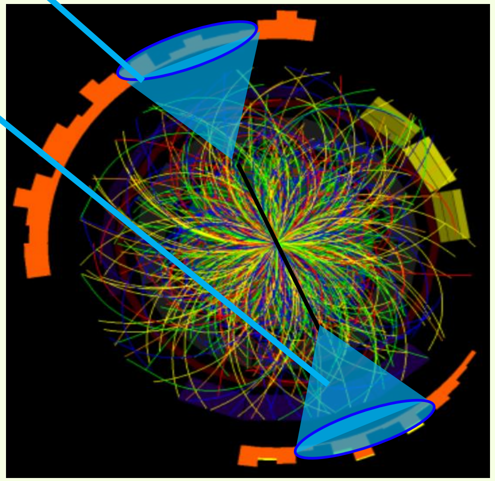
$$d_{ij} = \min(k_{ti}^{-2}, k_{tj}^{-2})\Delta R_{ij}^2/R^2 \text{ (anti- } k_T) \qquad R = \sqrt{\eta^2 + \phi^2}$$



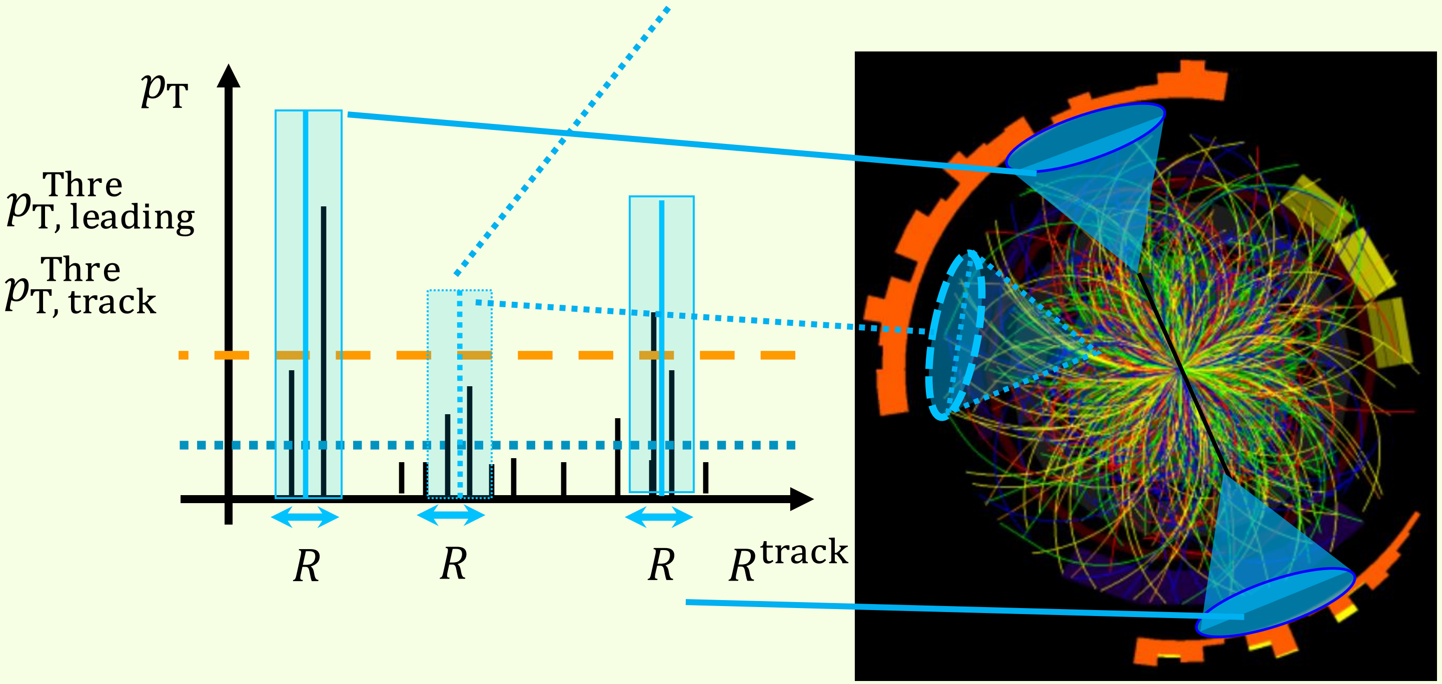
In the anti-kT algorithm, it cluster the tracks from higher p_T particles.

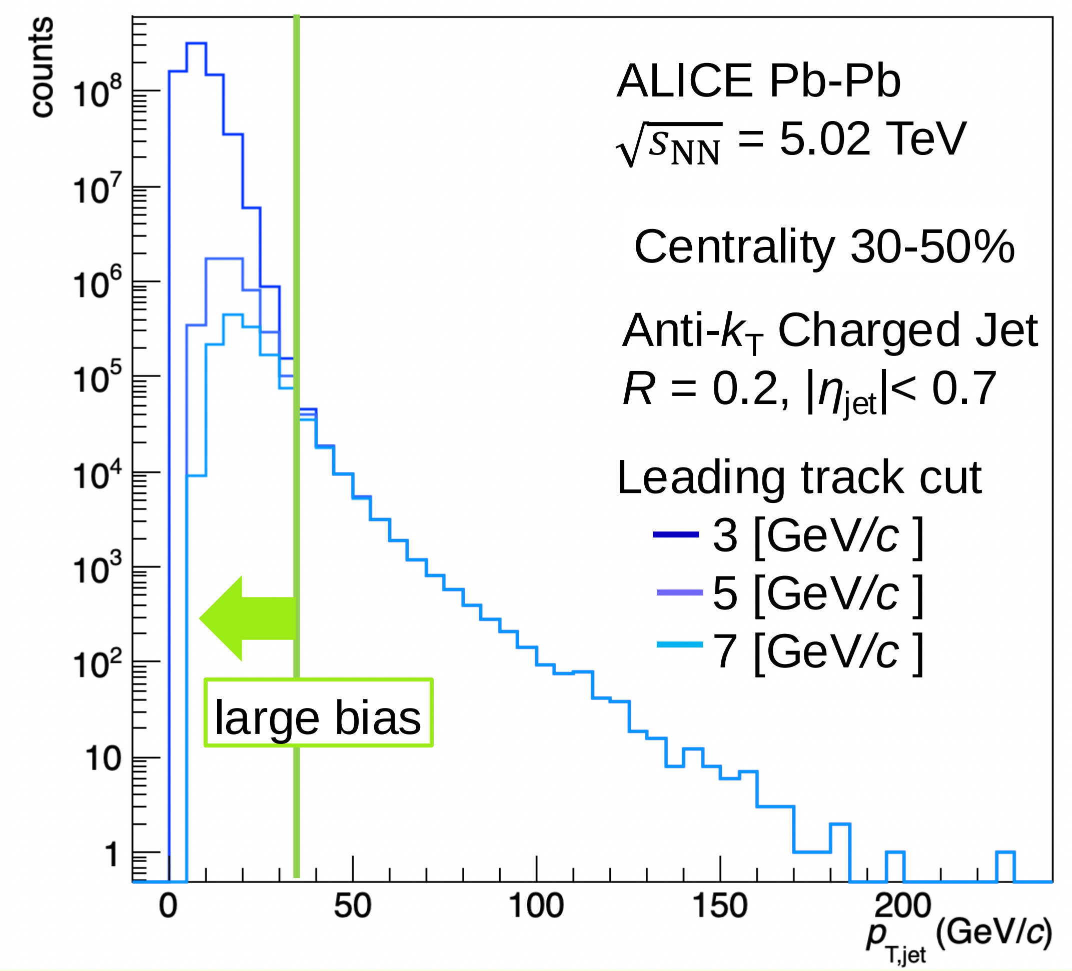
→ Anti-kT algorithm is sensitive to signal jets.





Merging a log of small p_T tracks, which are not came from a signal parton, can make a jet.
→ It is called *combinatorial jet*.

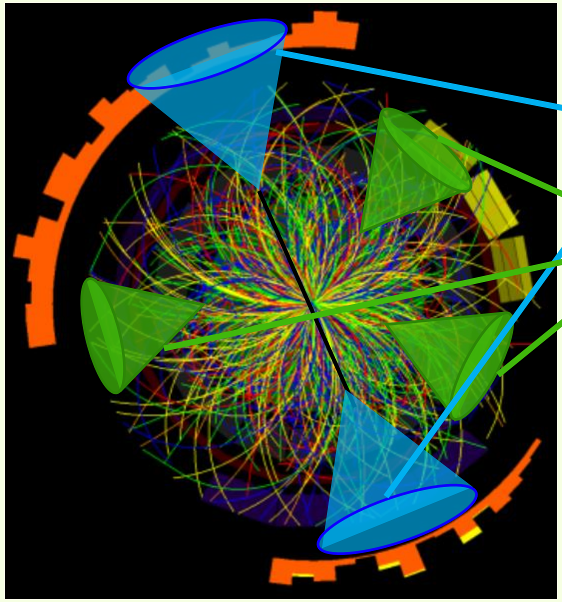




By requiring the leading track cut, highest p_T track in a jet , we can remove them.
However, it has a bias, so we need to take care. (3-7 GeV/c)

2. Background subtraction

Background p_{T} distribution (ordinary inclusive jet way)

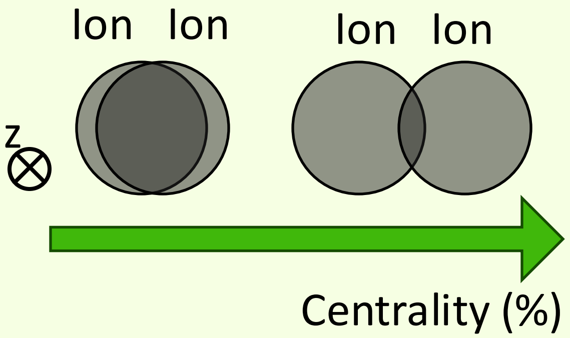


In HIC, a huge number of particles are produced.

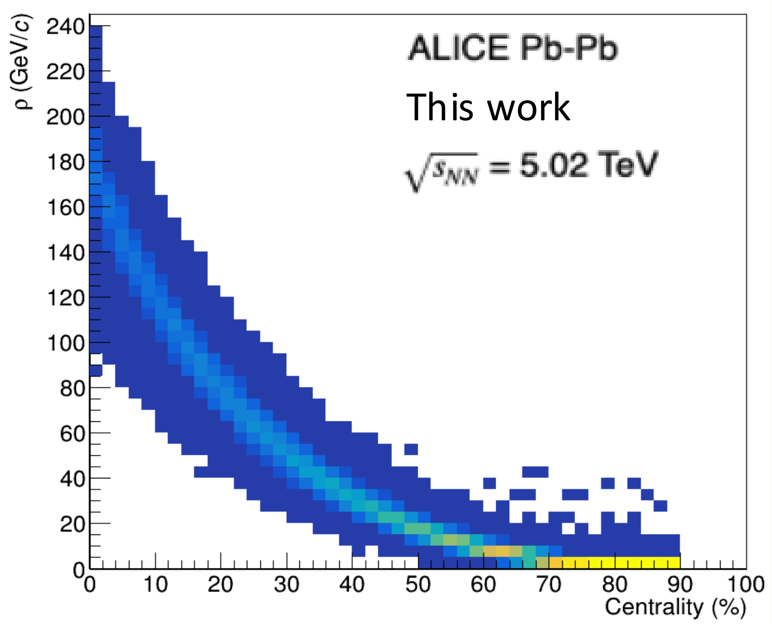
→ Signal jets are reconstructed with the background particles.

→Estimate background p_{T} density (ρ) except for signal jet area

$$\rho = \mathrm{median}(p_{\mathrm{T},i}/A_i)$$ A : cluster area, i: cluster id



background p_{T} for centrality



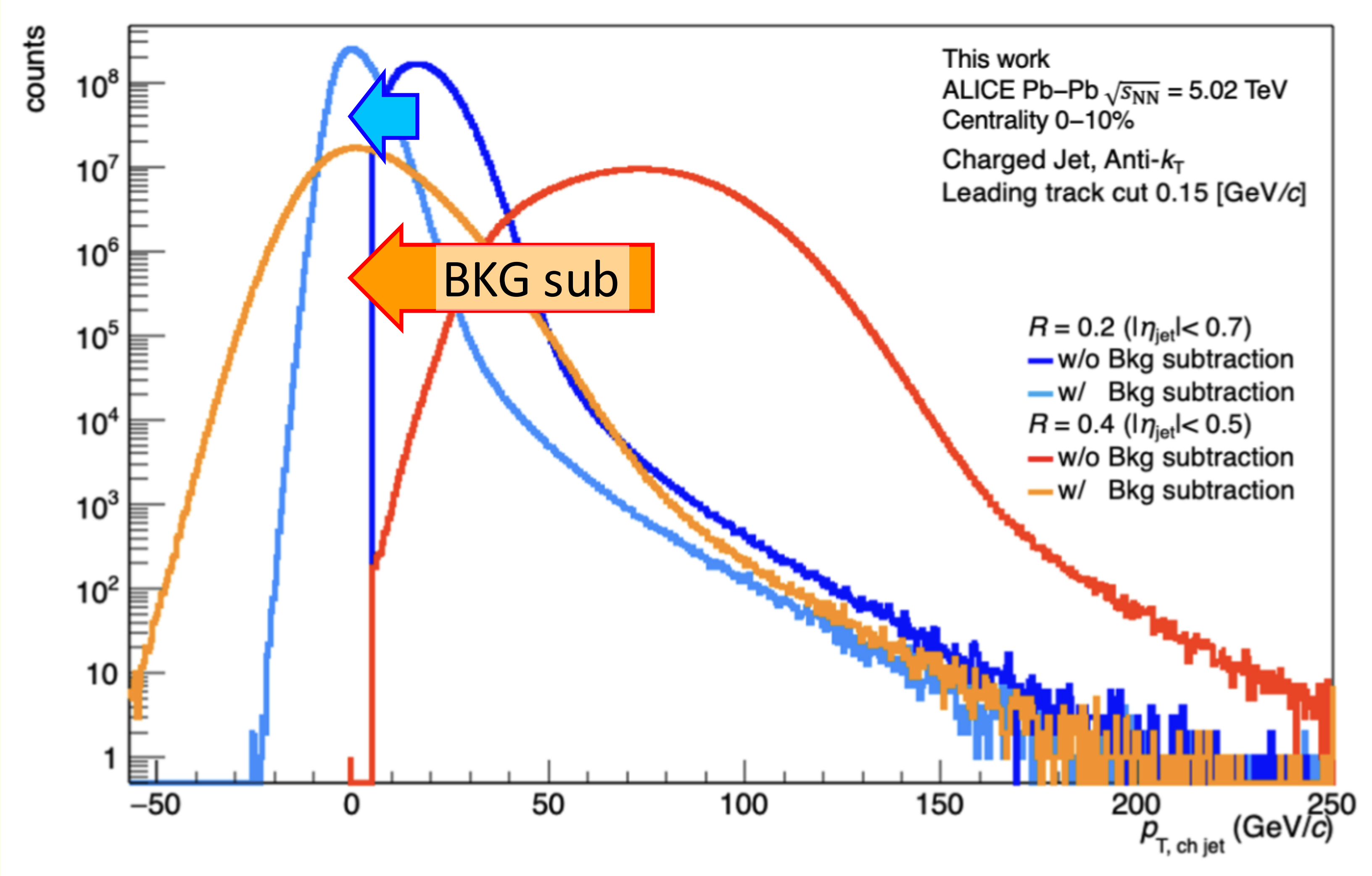
ρ is considered uniform for azimuthal angle and determined event by event

→ subtract the background from each signal jet

$$p_{\mathrm{T,corr}}^{\mathrm{jet}} = p_{\mathrm{T}}^{\mathrm{jet}} - \rho A$$

A : jet area

Jet p_T distribution before/after background subtraction



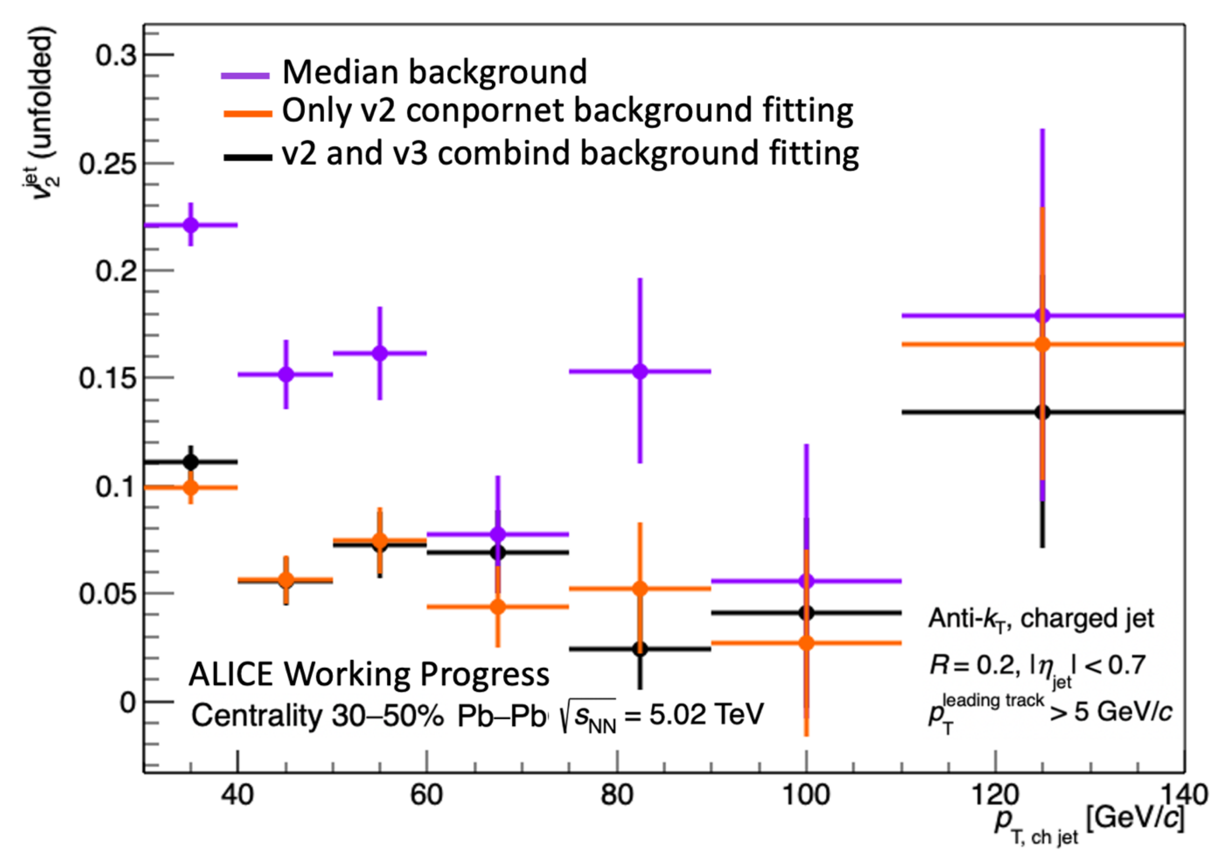
Subtracting bkg moves the jet distribution to low p_T direction.

Background subtraction considering the azimuthal anisotoropy

The actual background is not uniform for the azimuthal angle (ϕ).

For the inclusive jet analysis, the difference does not affect to the results.

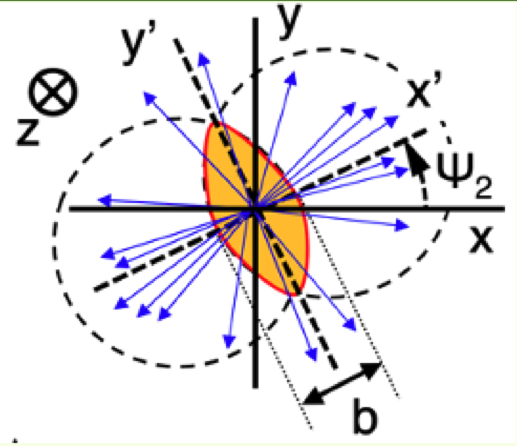
However, for the measurements depending azimuthal angle like the jet v_2, the effect is not negligible.



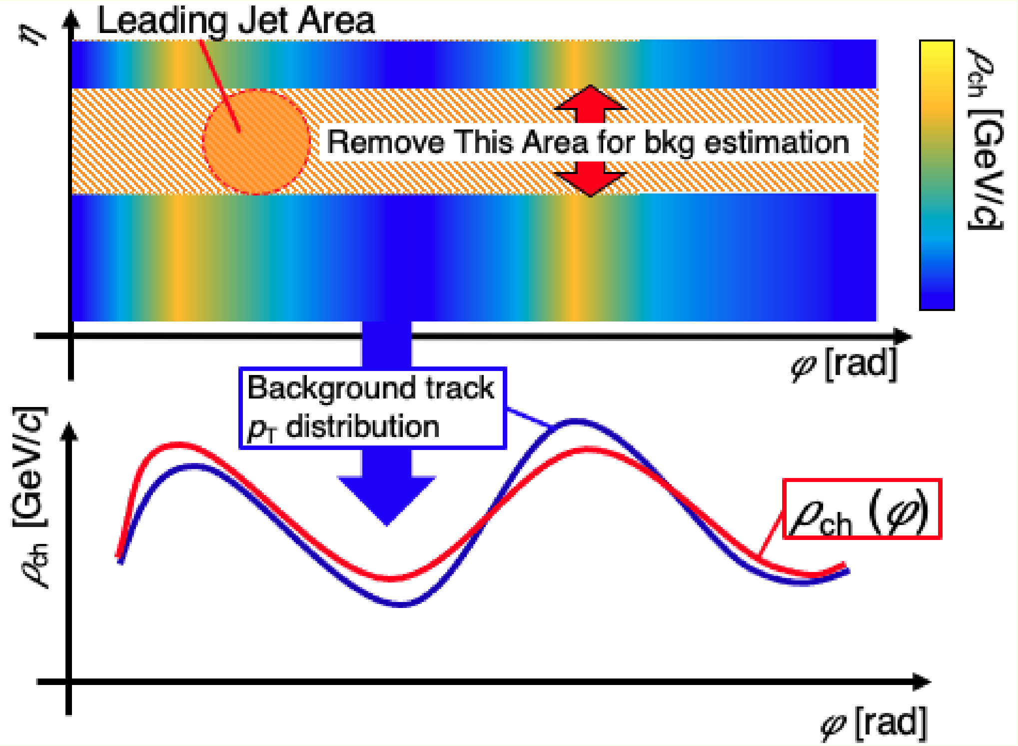
→Following slides show the stimation way and quality check ways.

Local background p_T estimation

The soft particle background is **not uniform** for azimuthal angle (φ).
→The background calculation should take the φ dependency into account.



The local rho is estimated using tracks except the leading jet η region. (Because of the statistic problem, it includes the sub-leading jet region.)



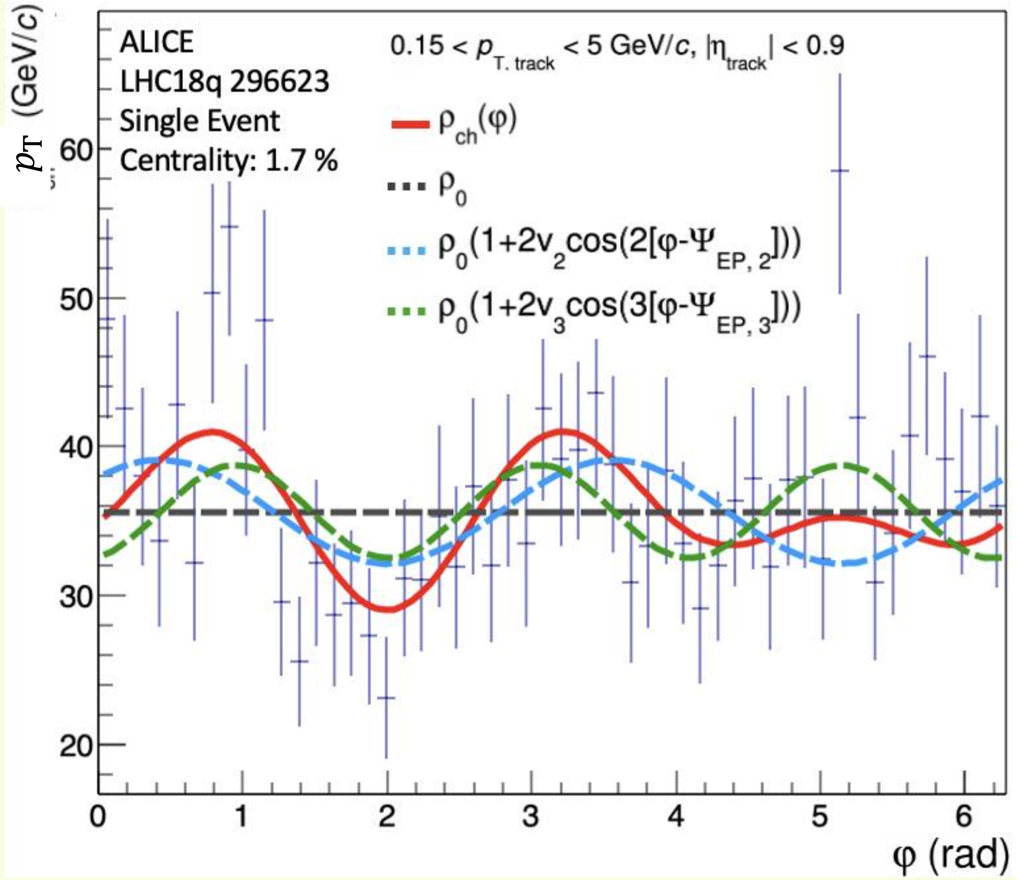
In this analysis, a following equation is used.

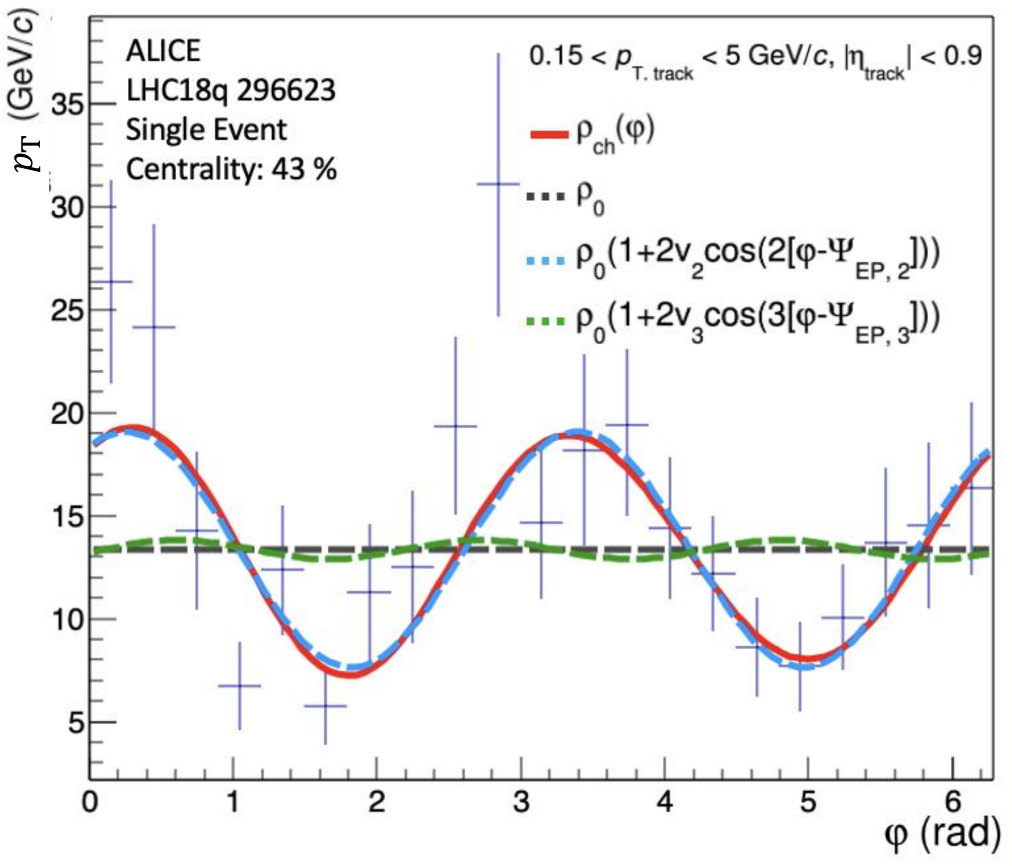
$$\rho_{ch}(\varphi) = \rho_0 \times \left(1 + 2\left\{v_2^{obs}\cos(2[\varphi - \Psi_{EP,2}]) + v_3^{obs}\cos(3[\varphi - \Psi_{EP,3}])\right\}\right)$$

$\Psi_{EP,2}$ and $\Psi_{EP,3}$ are calculated by the Qn vectors.
And ρ_0, v_2^{obs}, and v_3^{obs} are fitting value.

Local background p_T results

$$\rho_{ch}(\varphi) = \rho_0 \times \left(1 + 2\left\{v_2^{obs}\cos(2[\varphi - \Psi_{EP,2}]) + v_3^{obs}\cos(3[\varphi - \Psi_{EP,3}])\right\}\right)$$

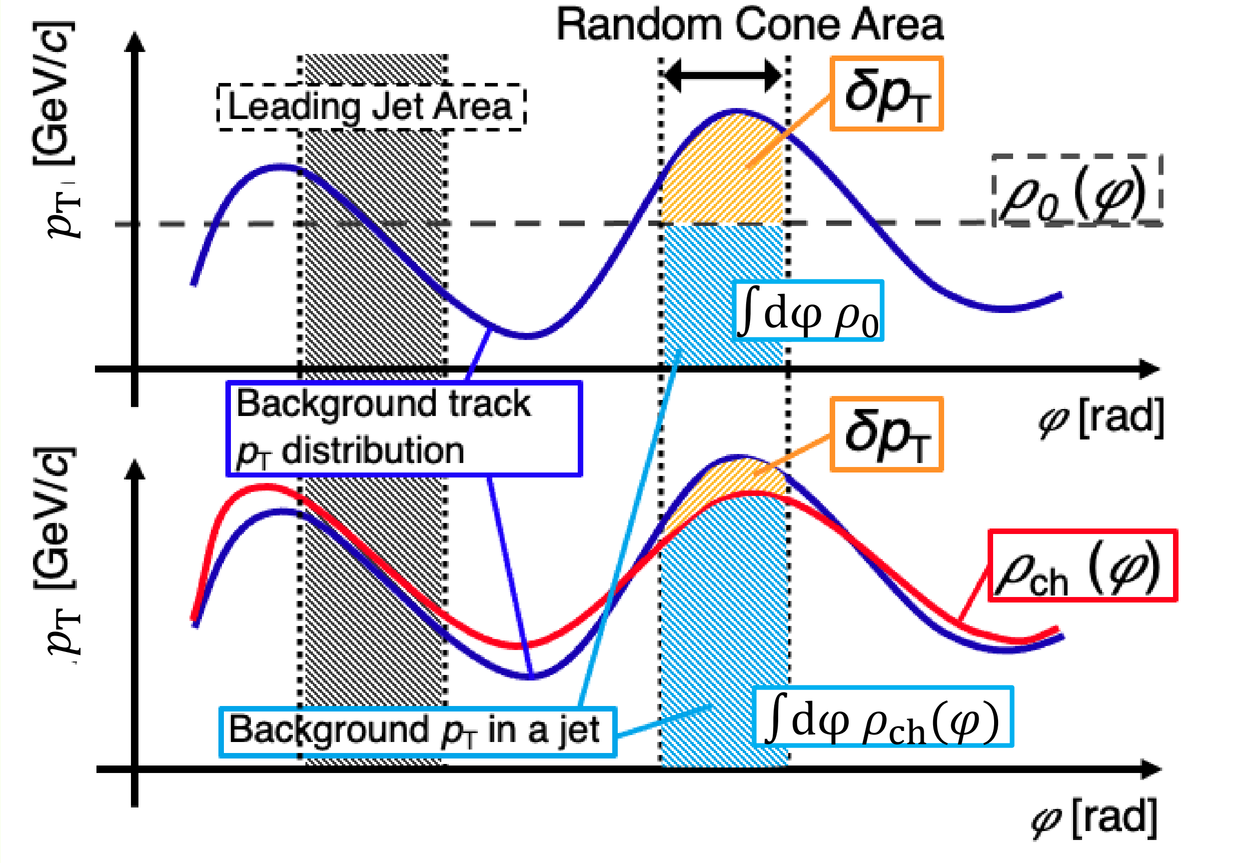




single event

# of bins = $\sqrt{N_{track}}$

Evaluation of background fit (δp_T)



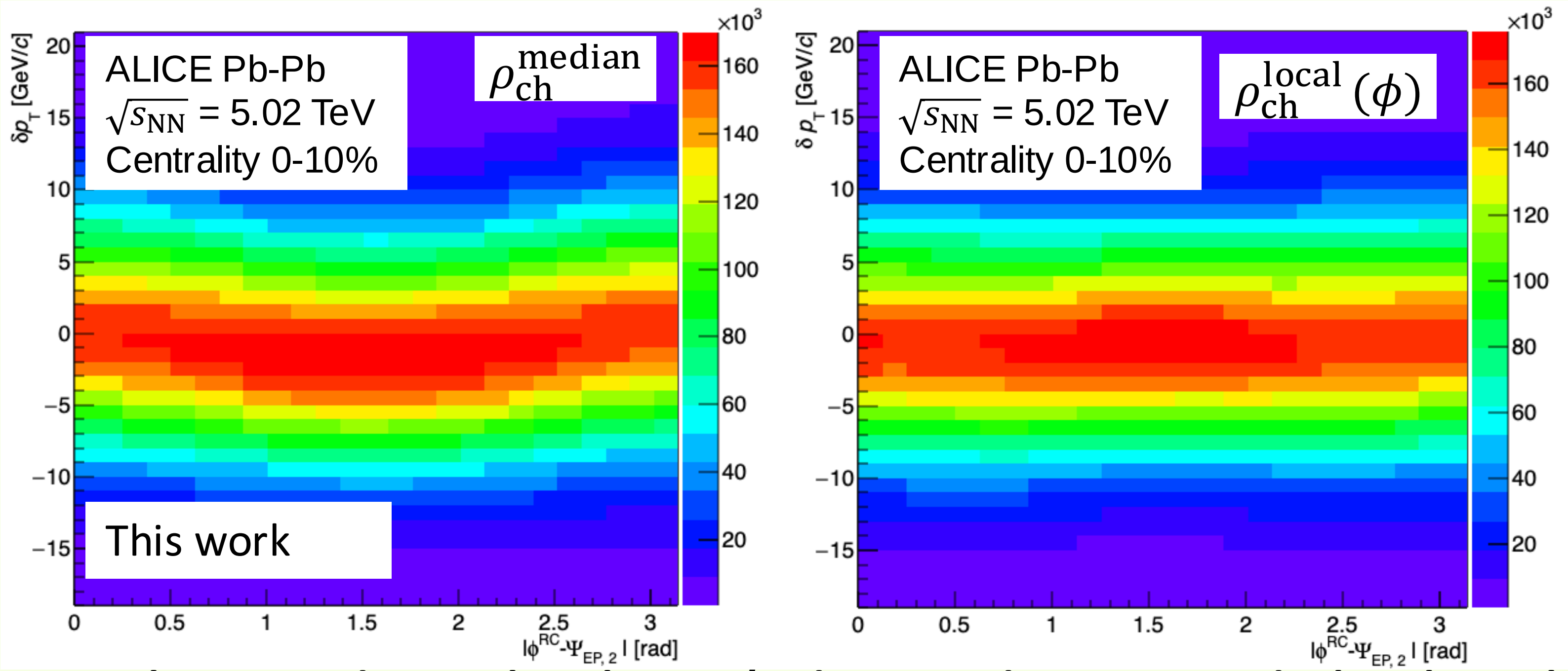
δp_T is a gap between integration of background tracks p_T and integration of background function in a random cone area.

We expect the local rho's δp_T should be smaller than the median one.
And in the local rho case, δp_T phi dependency is expected to make small.

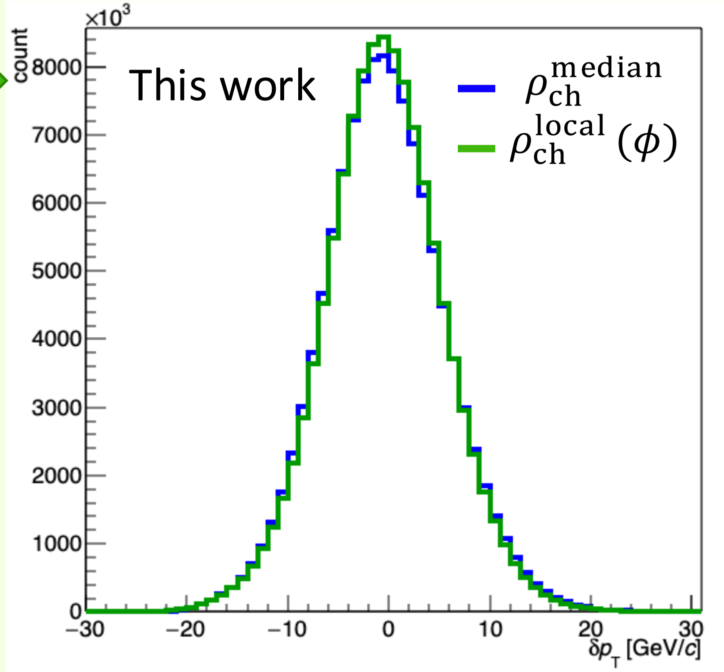
The Random cone is created once per event except the leading jet region.

The background δp_{T} distribution

$$\delta p_{\mathrm{T}} = \sum_i p_{\mathrm{T},i}^{\mathrm{track}} - \int_{-\mathrm{RC}}^{\mathrm{RC}} \rho(\phi)\, d\phi$$

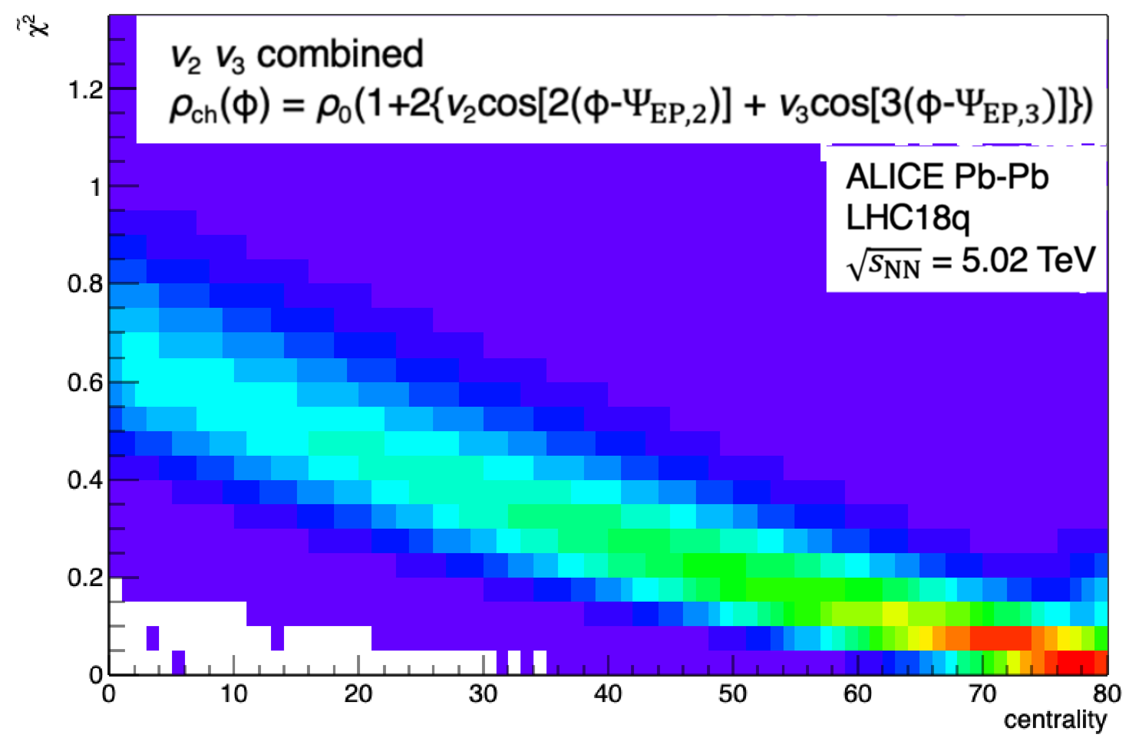


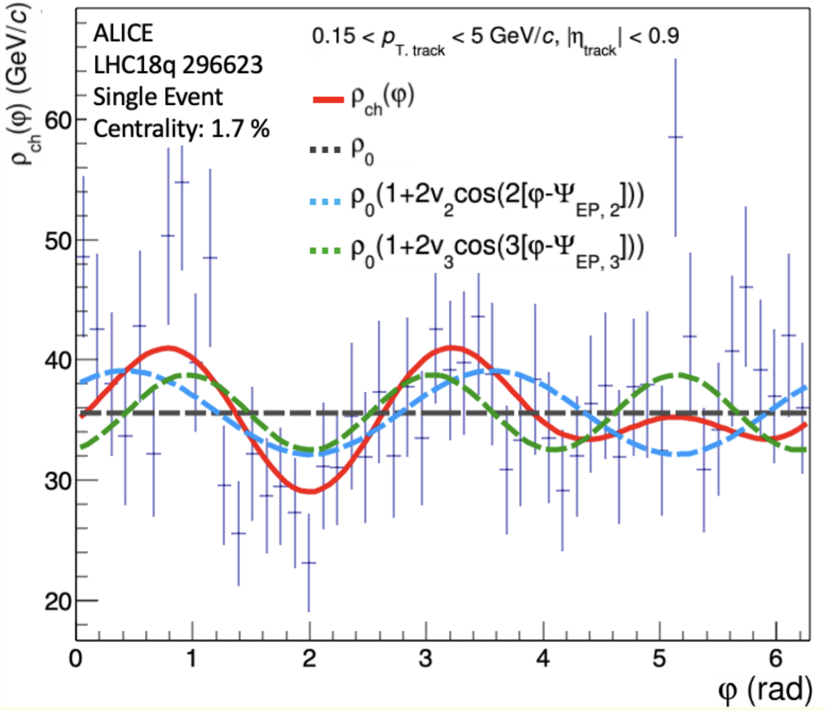
Projection on y axis



- The median rho has ϕ dependency and the local rho makes smaller the ϕ dependency.
- The dispersion of local rho background is more narrow than median rho.
  And these same tendency is seen in the all centrality regions.

Background pT function fit quality





$$\tilde{\chi}^2 = \left(\sum_{n=0}^{i} \frac{\left(p_T^{track} - p_T^{function}\right)^2}{p_T^{track}} \right) \Big/ (\#\text{ of bins } - 3)$$

\# of free parameters (ρ_0, v_2, v_3)

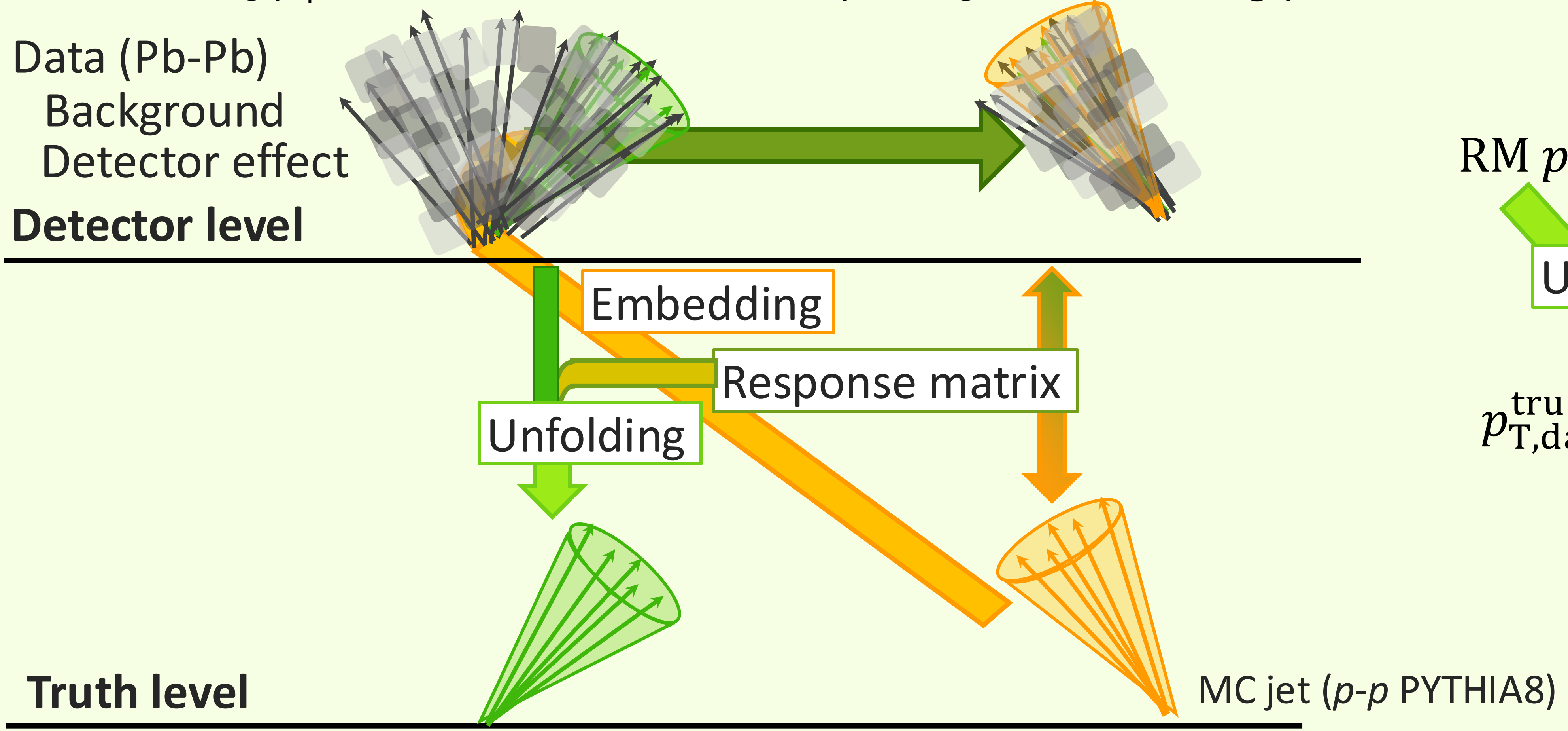
$\tilde{\chi}^2$ is smaller than 1. → Fitting quality is good.

3. Unfolding

Unfolding Process

The measured jet p_T distribution is affected by the background fluctuations and the finite resolution / efficiency of the detector

→ Correcting p_T distribution distortions by using the **unfolding** procedure.



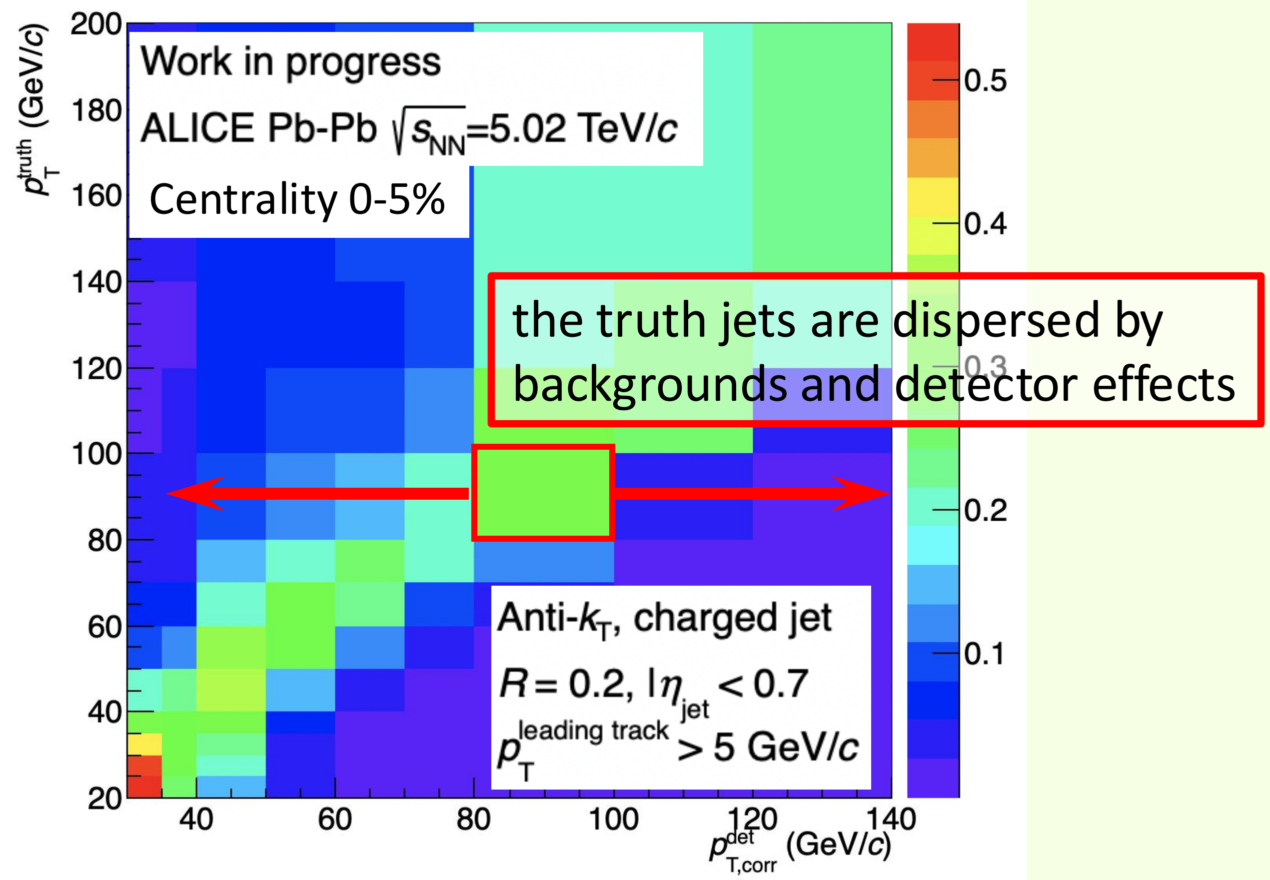
$$\mathrm{RM}\, p_{\mathrm{T,MC}}^{\mathrm{tru}} = p_{\mathrm{T,MC}}^{\mathrm{hyb}}$$

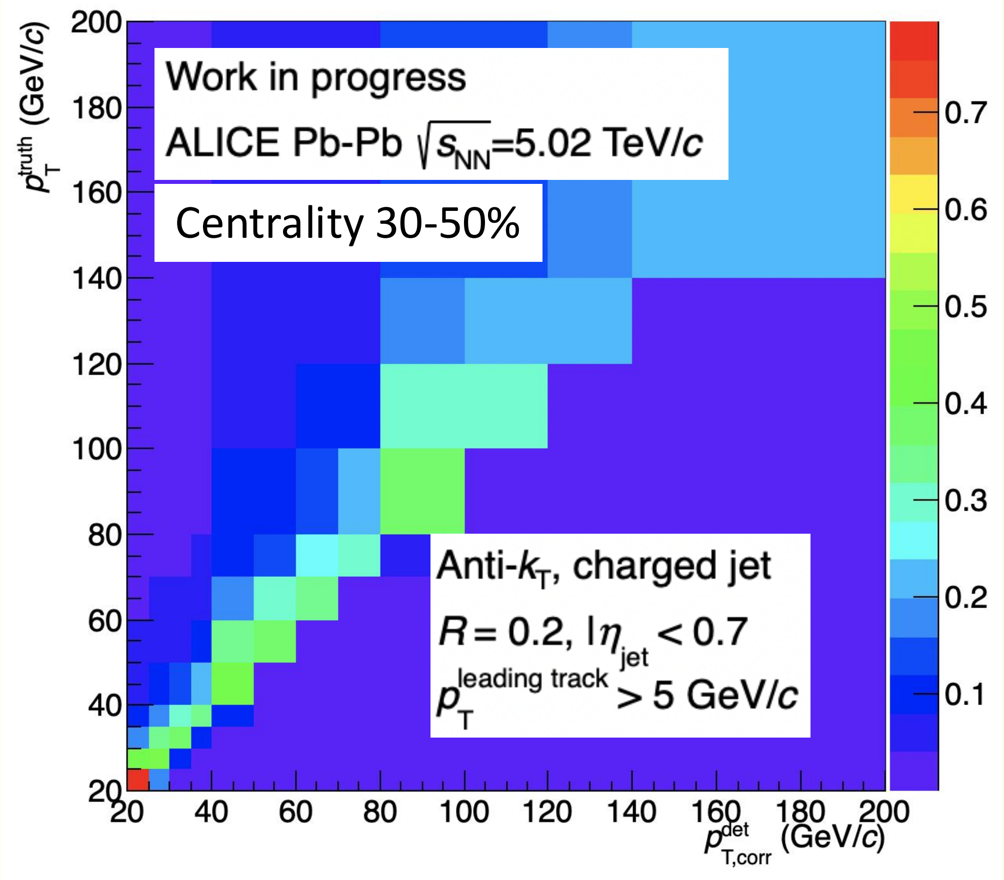
Unfolding

$$p_{\mathrm{T,data}}^{\mathrm{tru}} = \mathrm{RM}^{-1}\, p_{\mathrm{T,data}}^{\mathrm{meas}}$$

Response Matrix

Response matrix (RM) prepresents the jet p_T relation between truth and data.



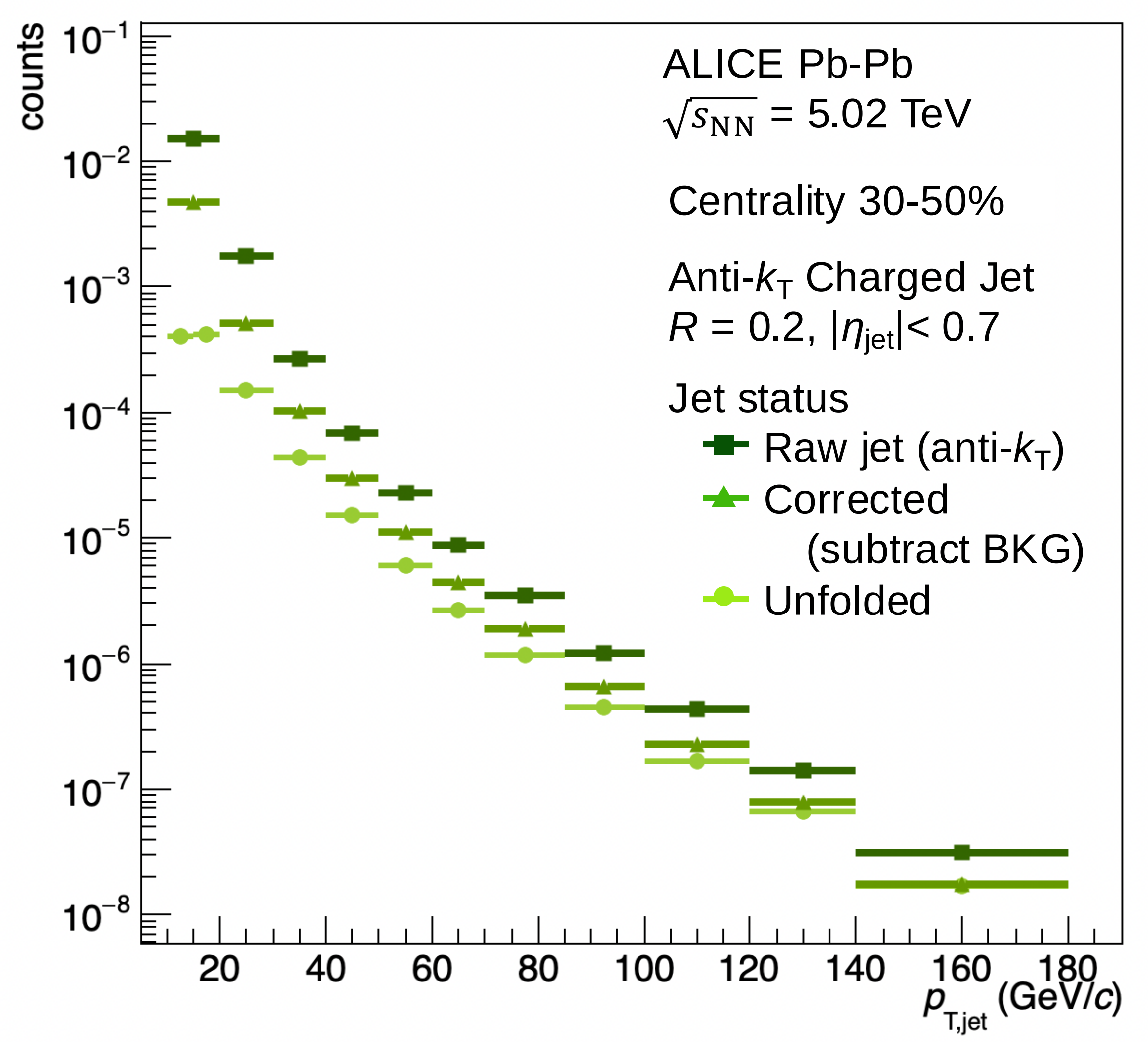


The RM inversion is not simple. RM → RM^{-1} .

→ Use unfolding package: RooUnfolding[link] (Baysian, SVD, and etc..)

Unfolded jet p_T distributions

Unfolding modify the jet distribution shape by considering the jet's p_T shift.



3. Unfolding

Kinds of Systematic Uncertainties

By chenging the parameters, we estimate systematic uncertainties.

Depends on the measurements and the value also depends on experiments.

Example

- Tracking efficiency (98%, 94%)
- Detector level p_{T} range in the response matrix ($\pm 5\ \mathrm{GeV}/c$)
- Unfolding iterations (± 1)
- Unfolding different prior (Modify input MC simulation)

\+ Event Plane Analysis

- Different background fitting function (Two type functions)
- Different event plane angle determination detector (V0M, V0A, V0C)

Backup Slides

Simple Jet Reconstraction Code

```
int main () {
  vector<PseudoJet> particles;
  // an event with three particles:   px     py  pz       E
  particles.push_back( PseudoJet(   99.0,  0.1,  0, 100.0) );
  particles.push_back( PseudoJet(    4.0, -0.1,  0,   5.0) );
  particles.push_back( PseudoJet(  -99.0,    0,  0,  99.0) );

  // choose a jet definition
  double R = 0.7;
  JetDefinition jet_def(antikt_algorithm, R);

  // run the clustering, extract the jets
  ClusterSequence cs(particles, jet_def);
  vector<PseudoJet> jets = sorted_by_pt(cs.inclusive_jets());

  // print out some info
  cout << "Clustered with " << jet_def.description() << endl;

  // print the jets
  cout <<   "        pt y phi" << endl;
  for (unsigned i = 0; i < jets.size(); i++) {
    cout << "jet " << i << ": "<< jets[i].pt() << " "
                   << jets[i].rap() << " " << jets[i].phi() << endl;
    vector<PseudoJet> constituents = jets[i].constituents();
    for (unsigned j = 0; j < constituents.size(); j++) {
      cout << "    constituent " << j << "'s pt: "<< constituents[j].pt() << endl;
    }
  }
}
```