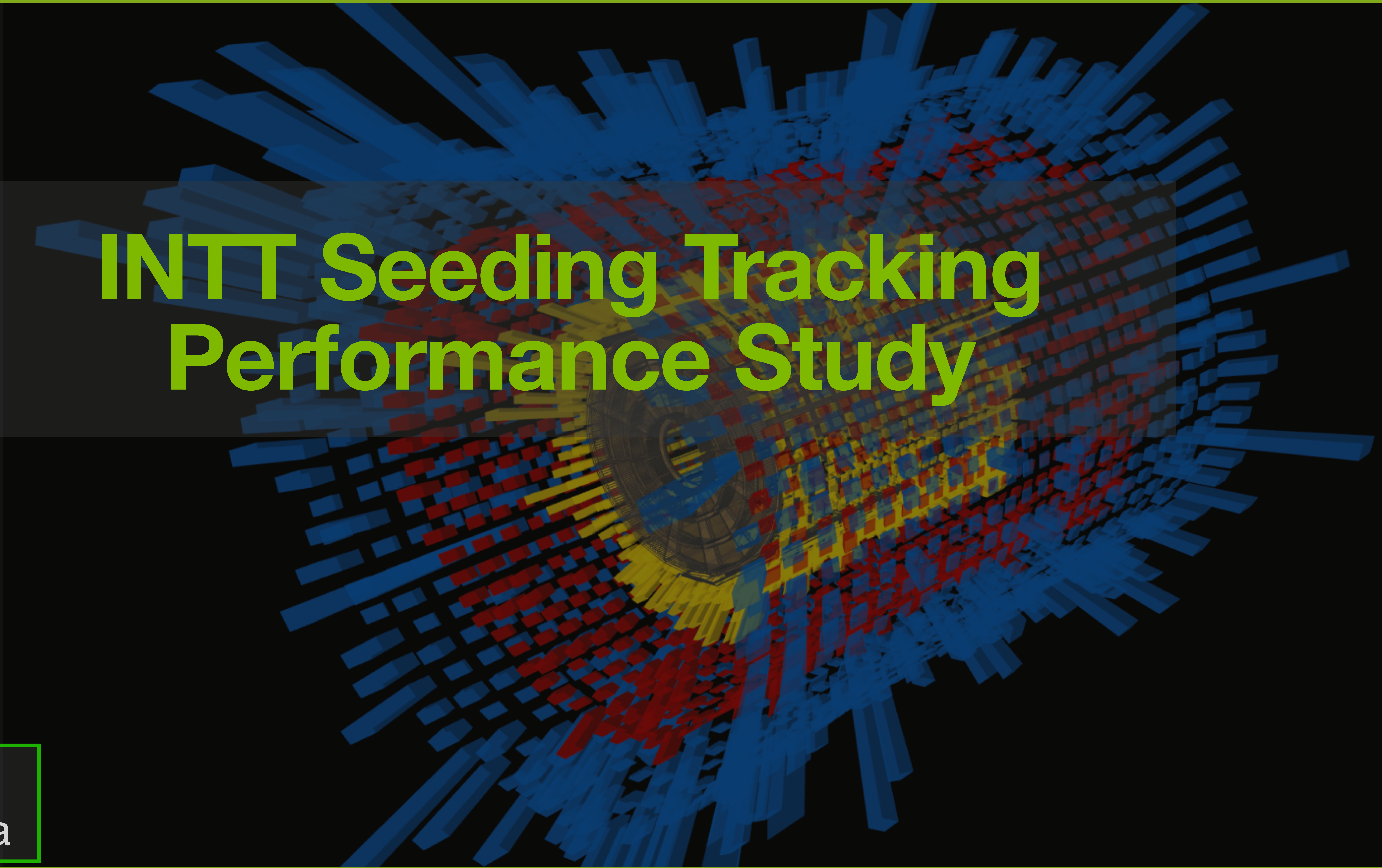


INTT Seeding Tracking Performance Study

RIKEN
Takuya Kumaoka

My Study Informations

Study Wiki Page:

https://wiki.sphenix.bnl.gov/index.php?title=INTT_AnalysisWorkshop2024_TakuyaKumaoka

Git link of this study:

- Particle Generation Simulation Codes
  https://github.com/sPHENIX-Collaboration/INTT/tree/main/general_codes/tkumaoka/InttSeedingTrackDev/ParticleGen

- INTT Seed Tracking Codes ← It will be explained in this slides of the algorithm part
  https://github.com/sPHENIX-Collaboration/INTT/blob/main/general_codes/tkumaoka/InttSeedingTrackDev/InttSeedTrackPerformance/src/InttSeedTracking.cxx

- INTT Seed Tracking Performance Estimation Codes ← It will be explained in this slides of the result part
  https://github.com/sPHENIX-Collaboration/INTT/blob/main/general_codes/tkumaoka/InttSeedingTrackDev/InttSeedTrackPerformance/src/InttSeedTrackPerformance.cxx

How to run this study codes

https://indico.bnl.gov/event/24622/contributions/99967/attachments/58840/101806/2024Dec16_Kumaoka_HowToRunMyCode.pdf

Aim of my study

Improve electron tracking using INTT + calorimeters

→ By adding calorimeter hit point, the tracking quality is expected to improve.

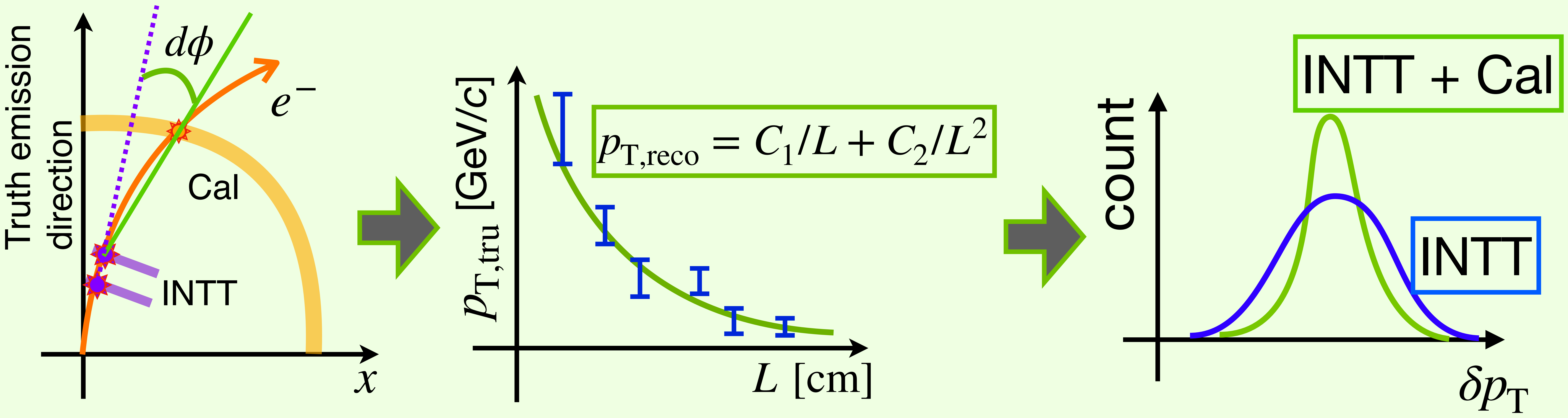
<**My study goal**> Evaluate how much the ***p_T* resolution** improve **by including the calorimeter** hits and show the potential of this tracking without TPC.

We expect the p_{T} can be described by a magnetic shift angle ($d\phi$) equation.

The coefficients (C_1 and C_2) is estimated using single electron simulation.

→ The function performance is evaluated by:

$$\delta p_{\mathrm{T}} = \frac{p_{\mathrm{T,reco}} - p_{\mathrm{T,truth}}}{p_{\mathrm{T,reco}}}$$

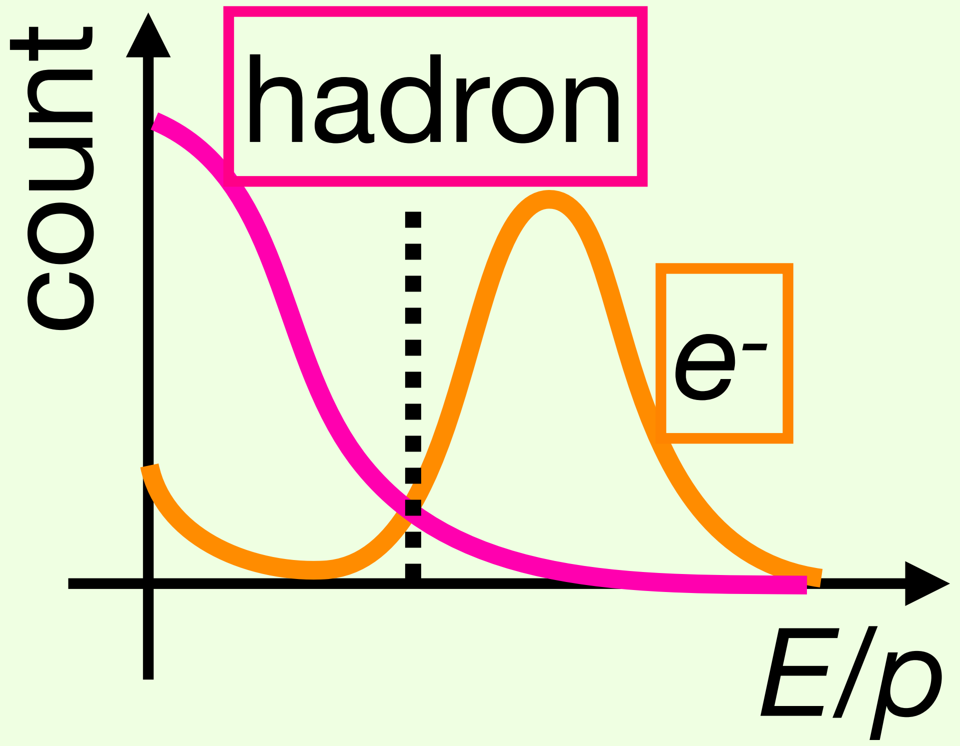


Background And Final Goal

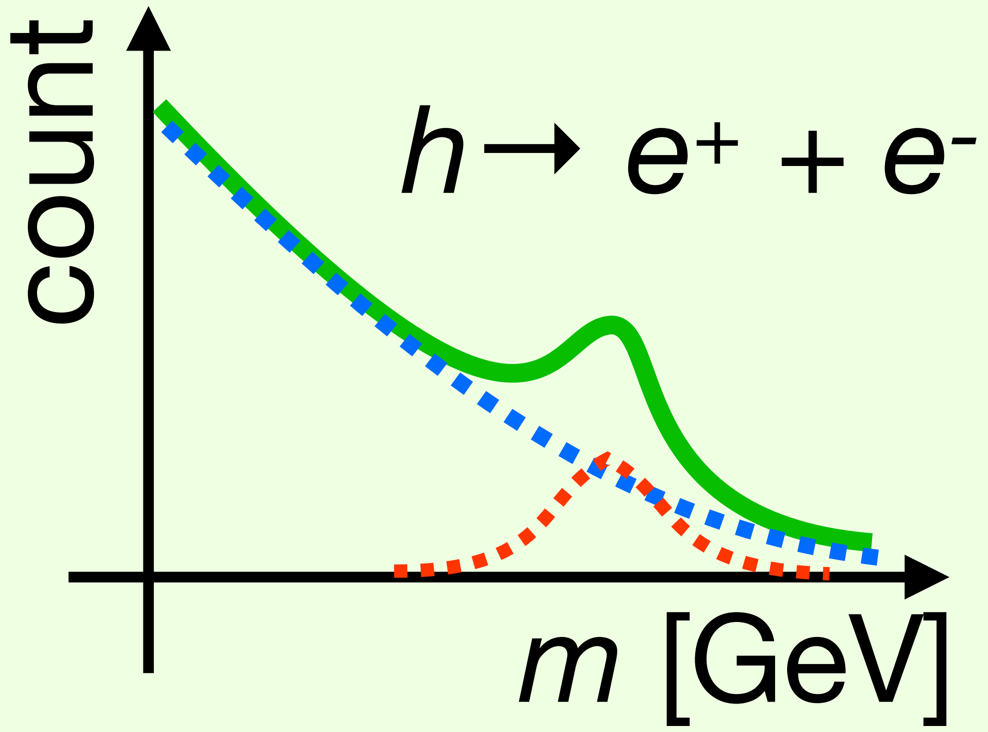
The RHIC-sPHENIX has a TPC detector to identify the charged particles.
→ However, TPC is very difficult detector to operate.

Current situation of tracking in sPHENIX
- TPC calibration will need much time to show physics.
- There are data without TPC.
- Streaming read-out data includes events that happen outside the TPC acceptance range.
- On the other hand, the tracking with the only MVTX+INTT seems difficult to identify electrons to get hadrons decaying into electrons with sufficient statistics.

➡ We expect that adding **calorimeter** information into the MVTX+INTT tracking will show enough PID performance.



➡



Short Results

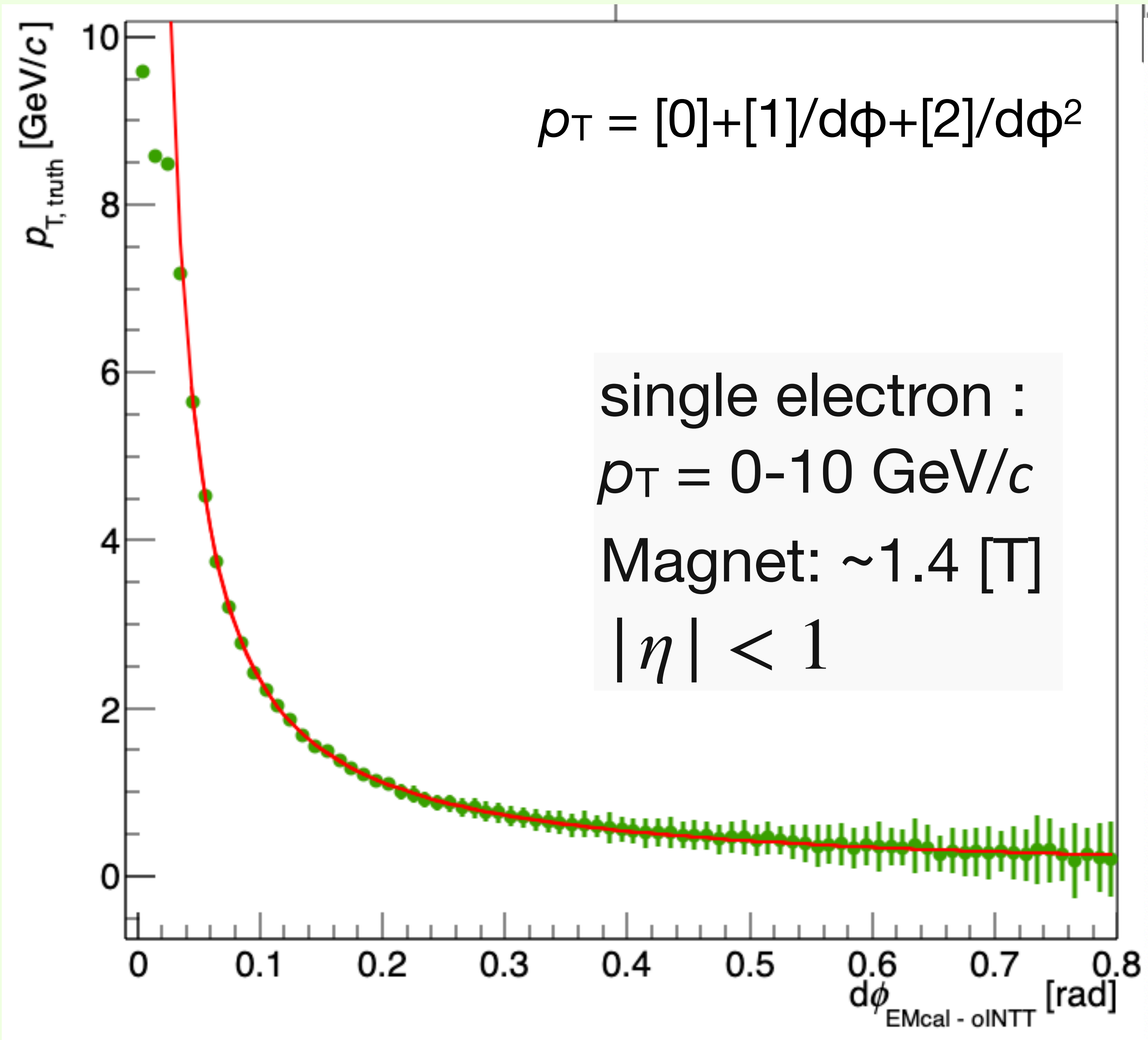
(1) The result could show the track w/ EMCal has better p_T resolution than only INTT track.

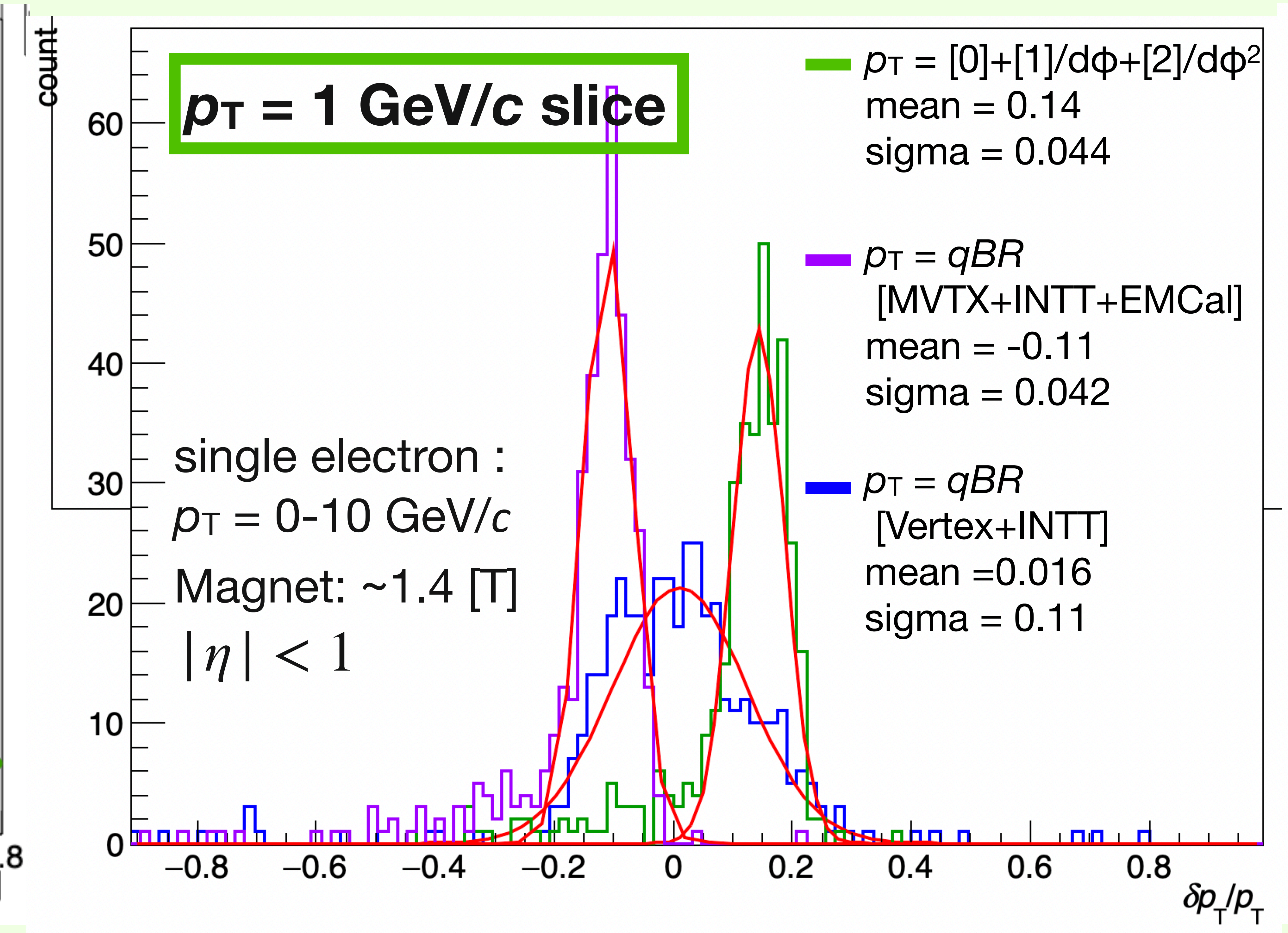
However, this track w/ EMCal could not achieve the 1-2 percents resolution expected.

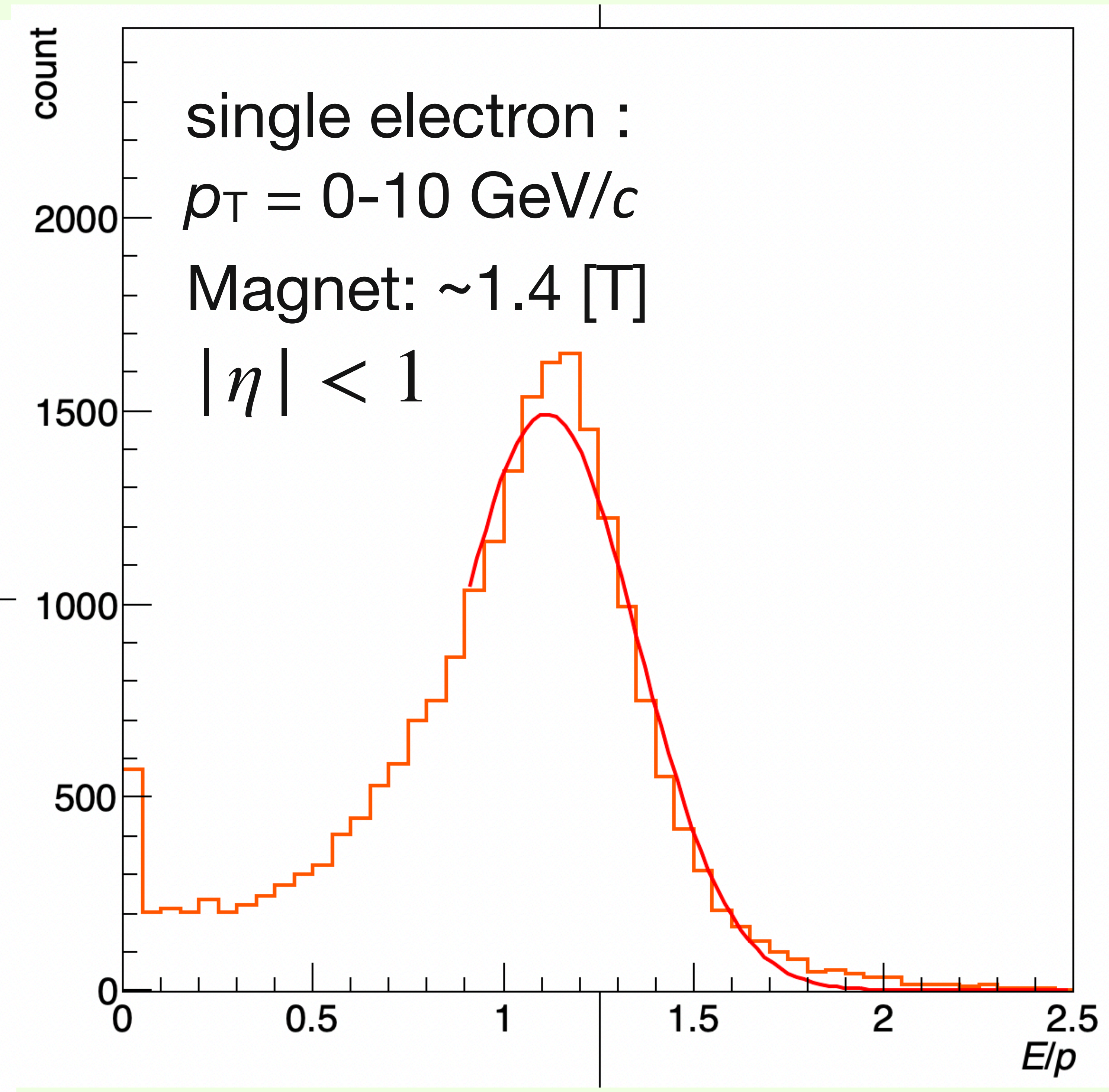
On the other hand, this result still does not include calibrated EMCal position.

Therefore, it indicates the possibility to achieve the 1-2 percents p_T resolution!

(2) The E/P distribution shows a peak around 1.







INTT Seeding Algorithm

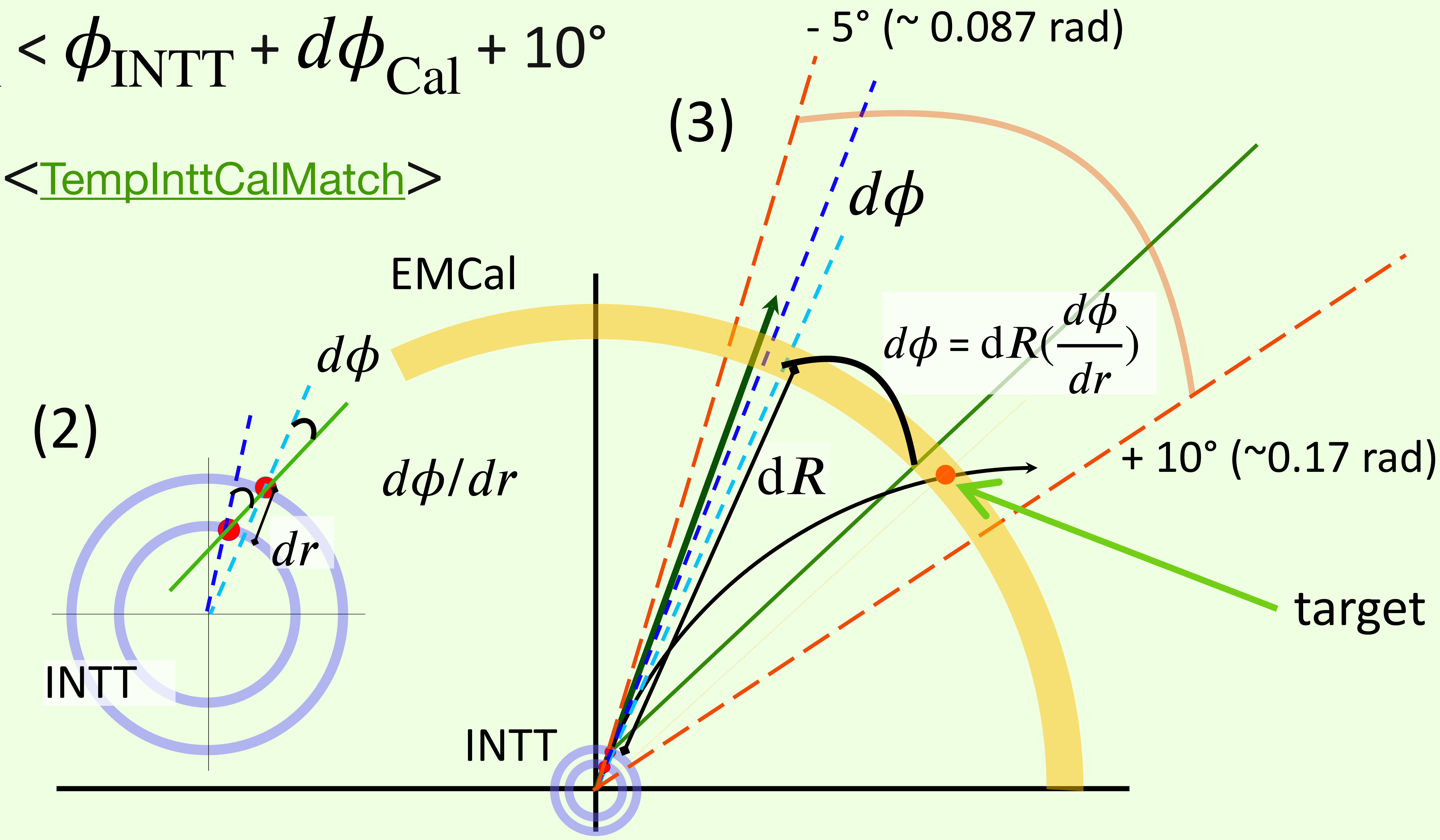
INTT + EMcal Hit Matching Algorithm

(1) Find a inner INTT cluster having the closest $\phi_{\text{outer INTT}}$ <TempINTTIOMatching>

(2) Caclulate $d\phi/dr$ (outer INTT - inner INTT) <TempCalcdPhidR>

(3) Searching for an EMCal cluster (> 0.1 MeV) having the highest energy in the ϕ_{Cal} range $\phi_{\text{INTT}} - 5° < \phi_{\text{Cal}} < \phi_{\text{INTT}} + d\phi_{\text{Cal}} + 10°$

$d\phi_{\text{Cal}} = d\phi/dr * (R_{\text{EMCal}} - R_{\text{INTT}})$ <TempInttCalMatch>

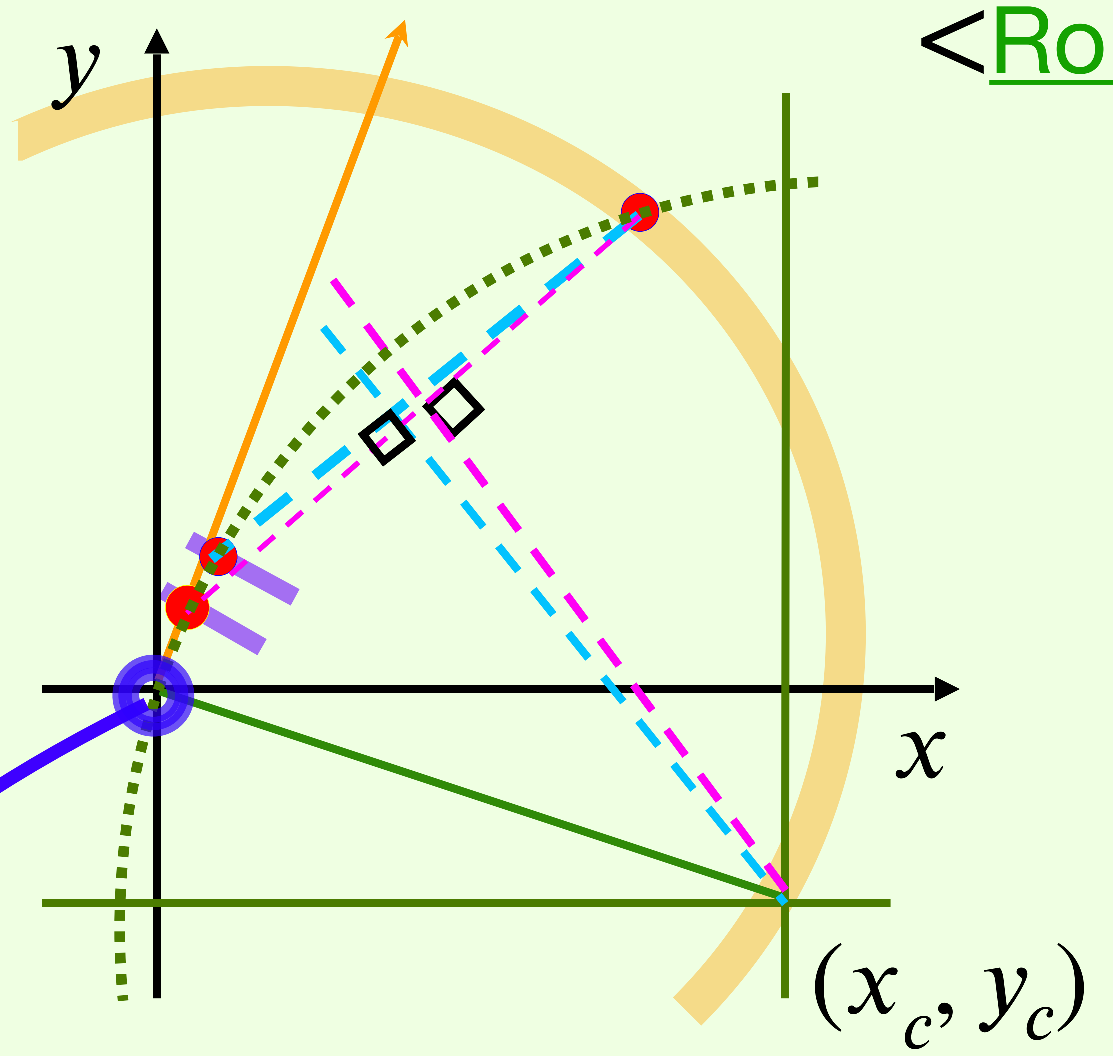


MVTX Hit Matching Algorithm

(1) Draw a circle using three hit points (iINTT + oINTT + EMcal)

(a) No fit approach (The last page result)

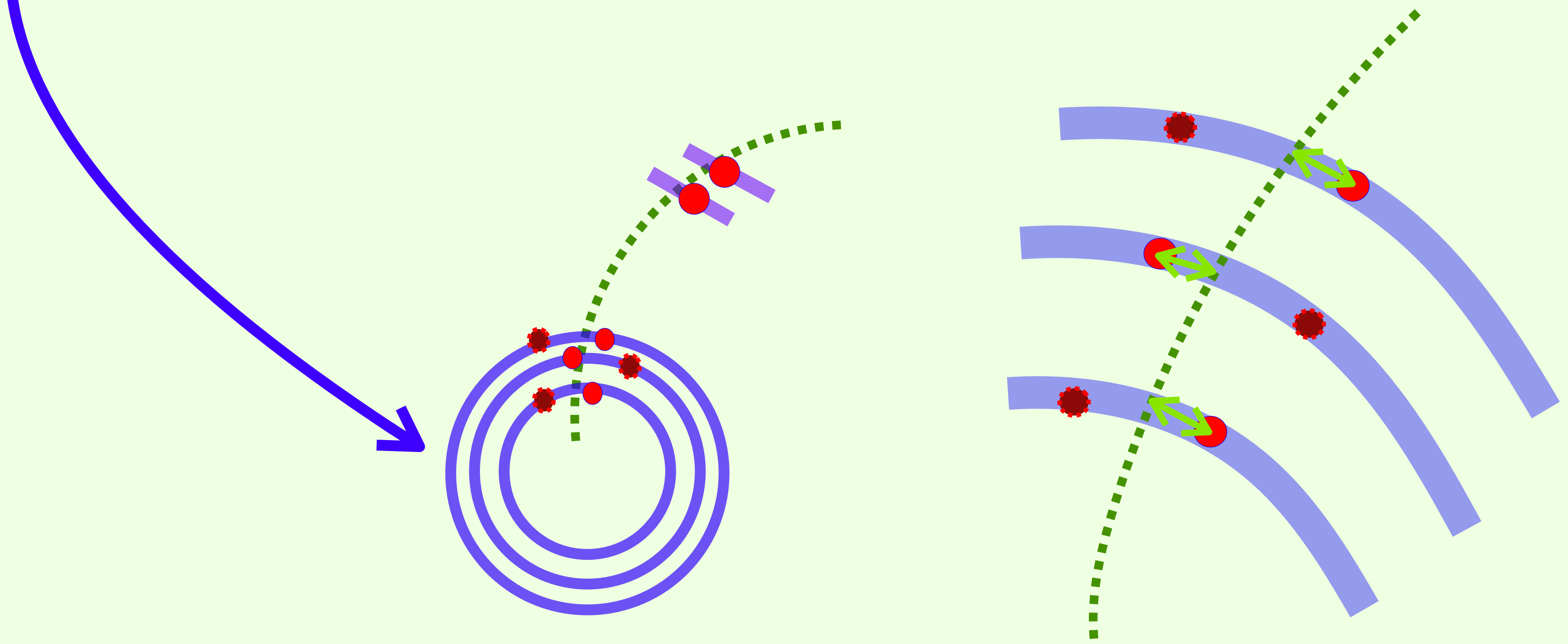
<RoughEstiSagittaCenter3Point>



(b) Use ROOT fit <SagittaRByCircleFit>

$$(y = \sqrt{R^2 - (x - x_c)^2} - y_c)$$

(2) Select Closest Points of MVTX <AddMvtxHits>

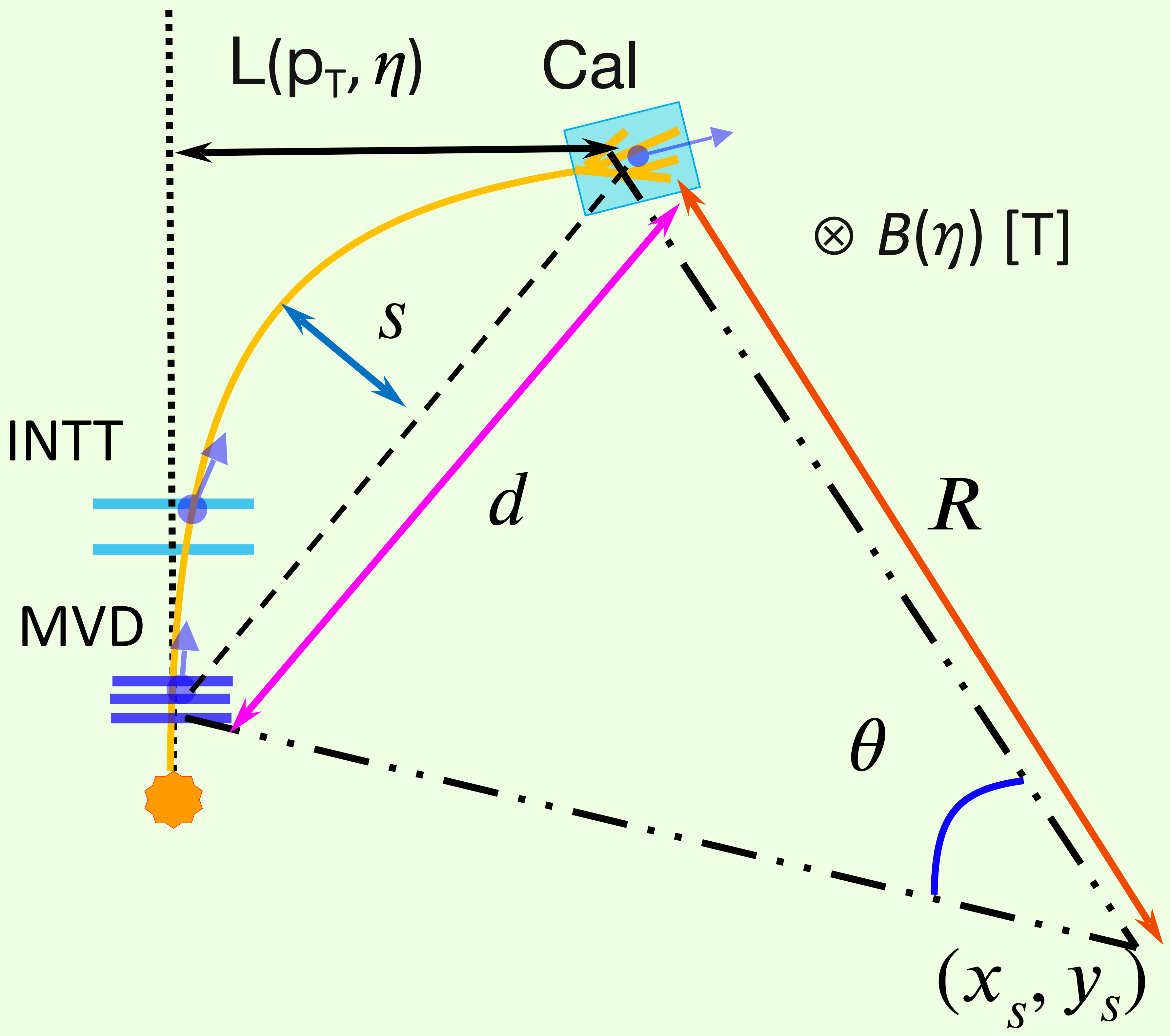


Sagitta p_T equation

$$p_T[\mathrm{GeV}] = qBR$$
$$= 0.3B[T]R[m]$$

Fitting the circle equation ($y = \sqrt{R^2 - (x - x_s)^2} + y_s{}^2$) for the three points (inner INTT, outer INTT, and EMCal) and estimate the R.

Using this R, the p_T can be calculated.

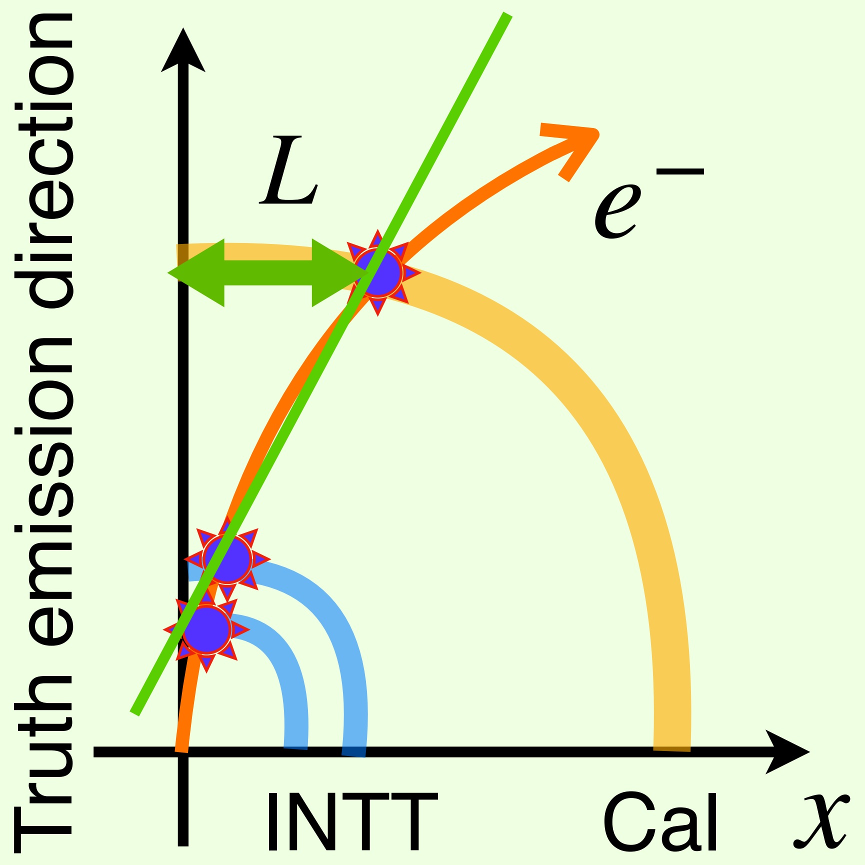


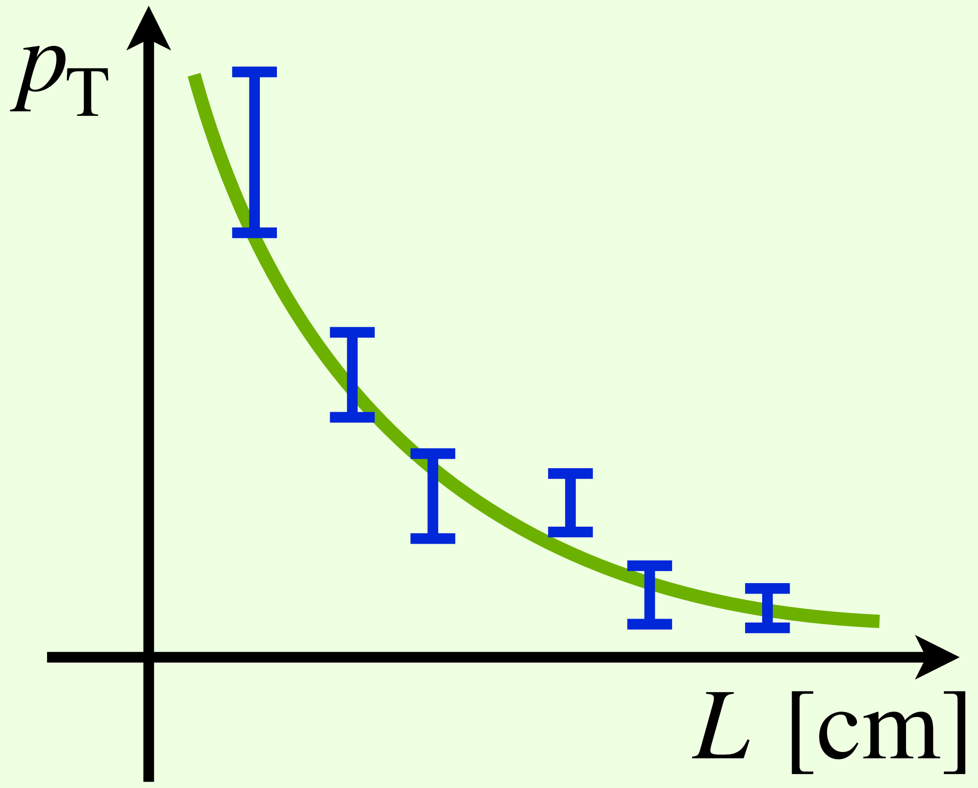
Acculate p_{T} Estimation Idea

We expect the p_{T} can be described by a magnetic shift length (L) equation.

Now, I am estimating the coefficients (C_1 and C_2) using single electron simulation.
→ I need to estimate the function performance.

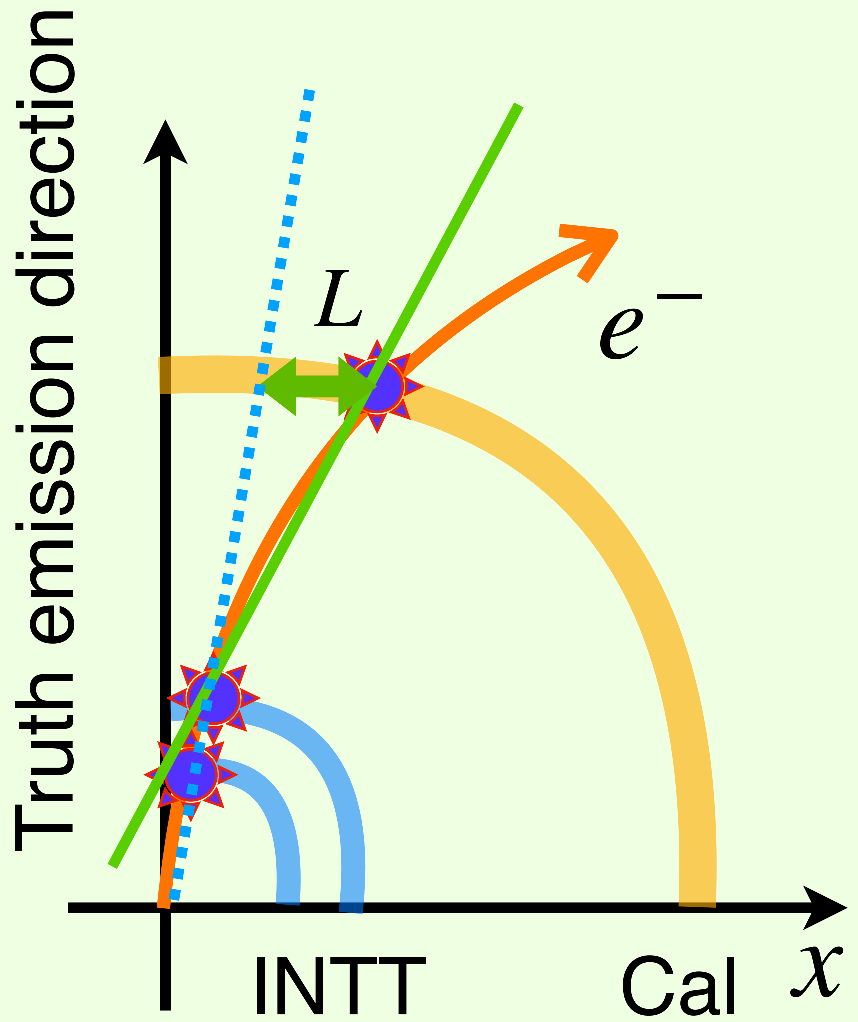
$$\sigma p_T = \frac{p_{\mathrm{T,reco}} - p_{\mathrm{T,truth}}}{p_{\mathrm{T,reco}}}$$



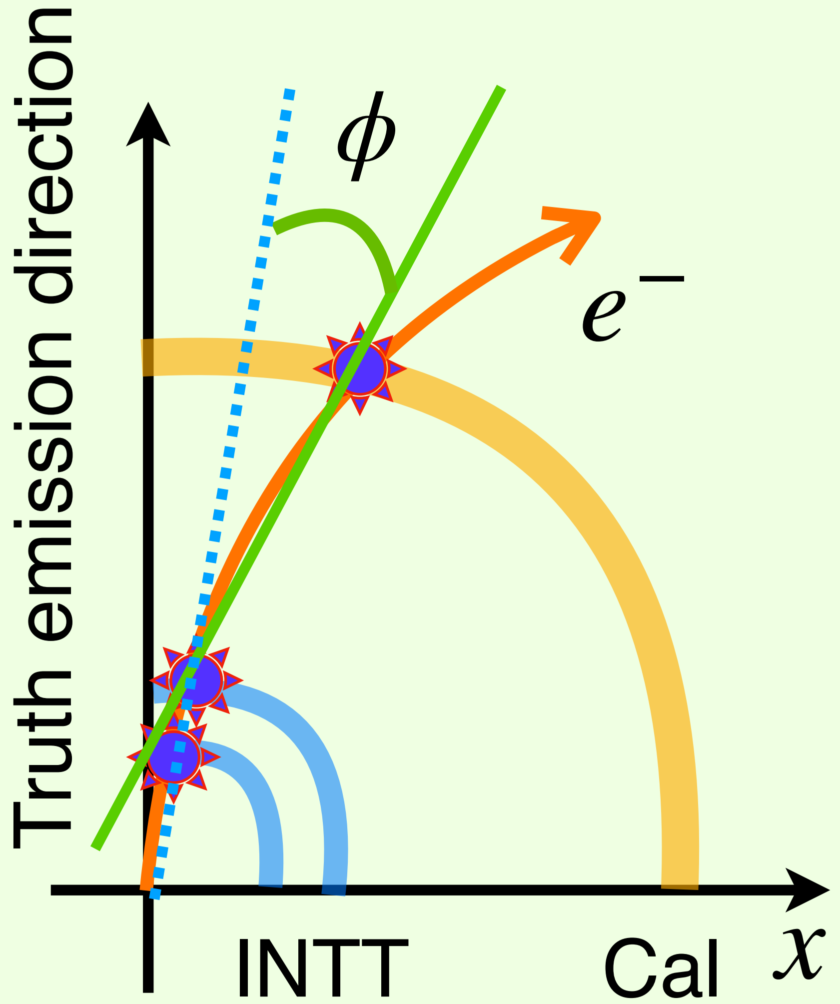


$p_{\mathrm{T}} = C_1/L + C_2/L^2$

We do not know the truth emission direction.

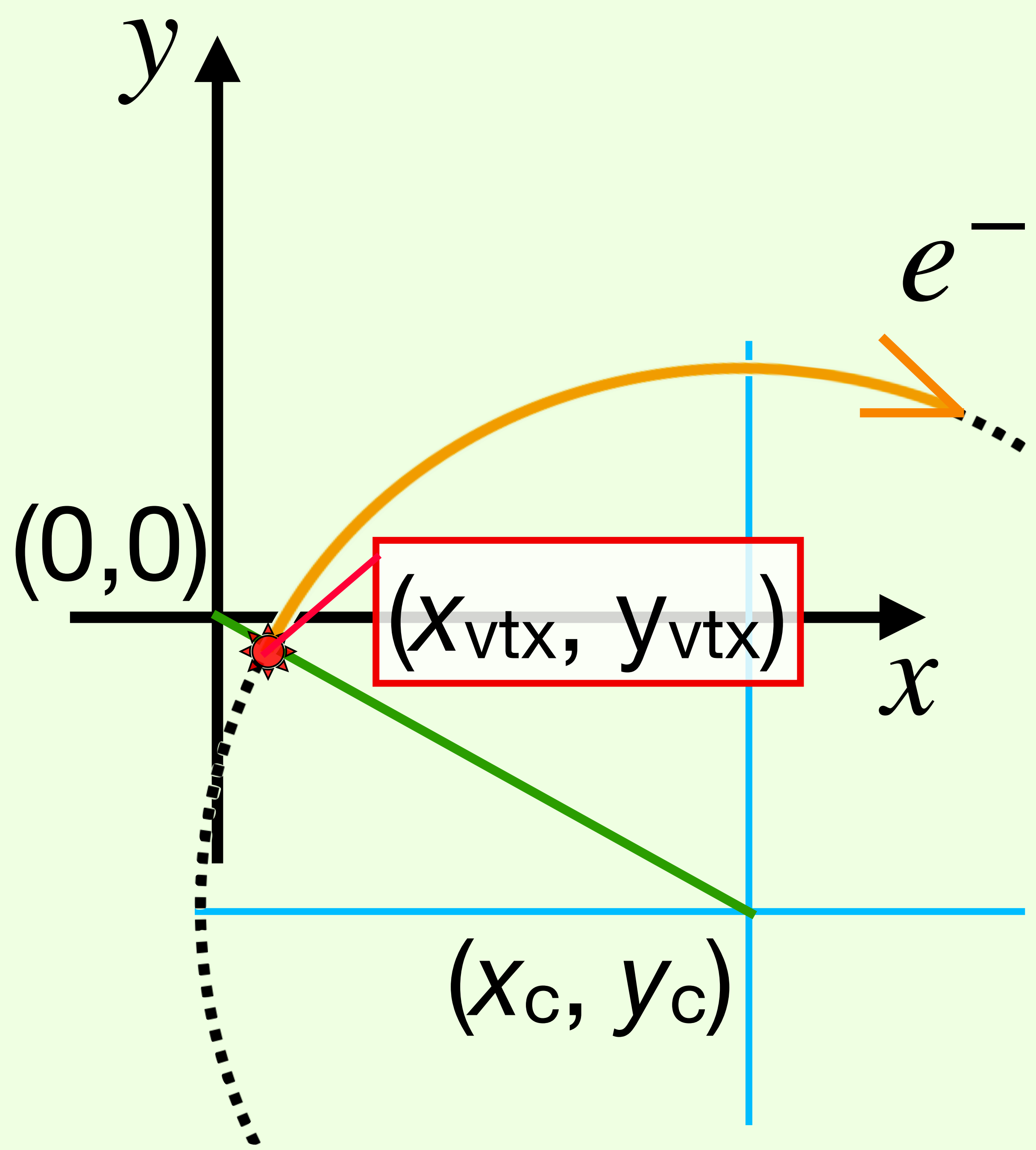


or

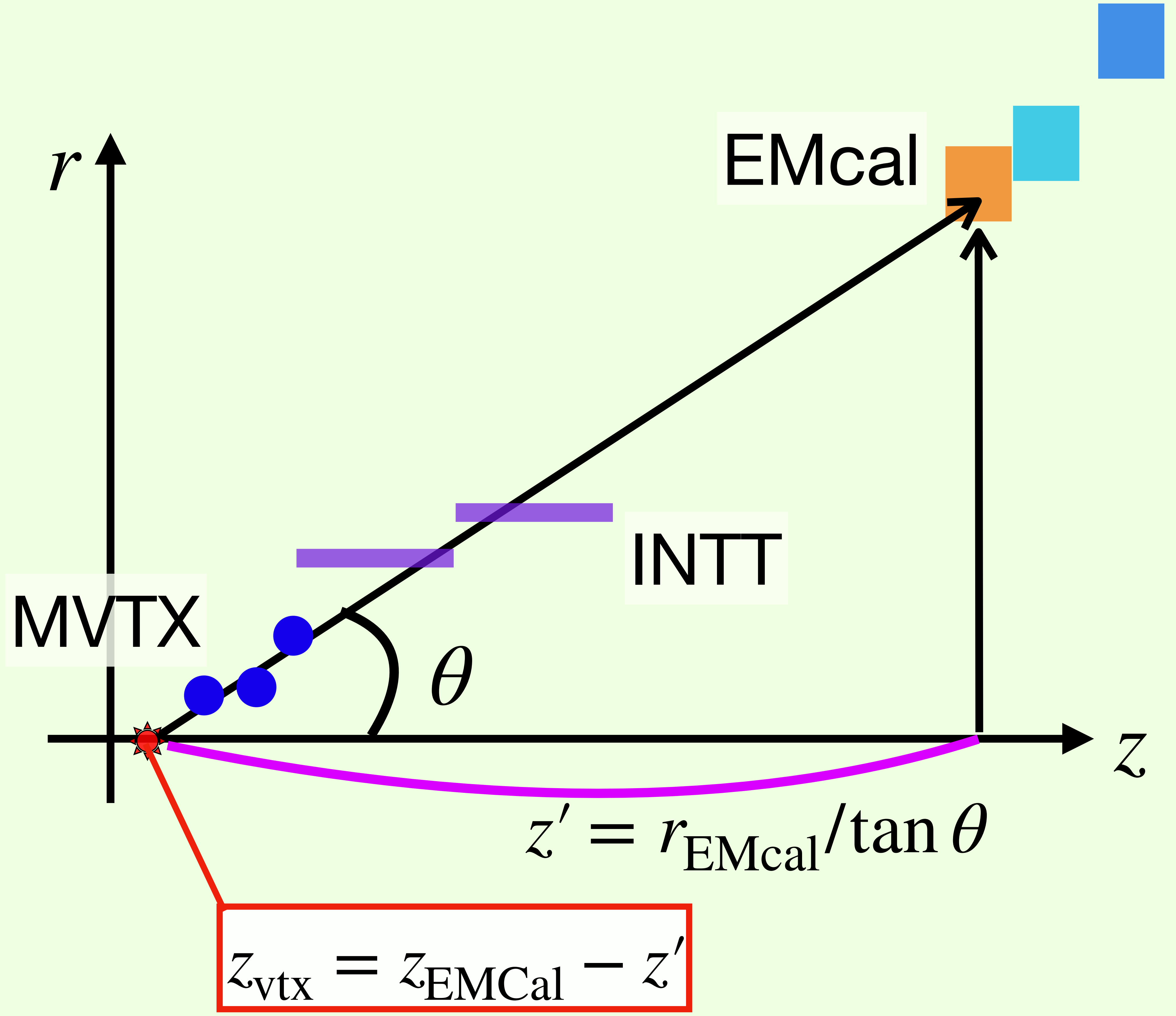


Vertex Determination

x, y position: <CrossLineCircle>
The cross point of the circle drown by the hits connection and the line between the (0, 0) and the center of circle.



z position:
The cross point of the line drown by the hits and the horizontal line.



Tracking Performance Results

Input Event File

Simulation: Single particle gun + GEANT4

→ output: DST file format

Output Contents <some container info>

- Truth Info <>
- Tracking detectors cluster <TrkrClusterContainerv4, container, lines in Fun4All>
- Calorimeter cluster <RawCluster, container, lines in Fun4All>

Inject electron p_T : 0-10 GeV/c

Inject range: ϕ: $-\pi$ to π, η: -1 to 1

GEANT4 Setting: Magnet 1.4 T

Detector: MVTX, INTT, TPC, EMCal, iHCal, oHCal

sPHENIX Magnetic Field

Document Location

https://indico.bnl.gov/event/7081/attachments/25527/38284/sphenix_tdr_20190513.pdf

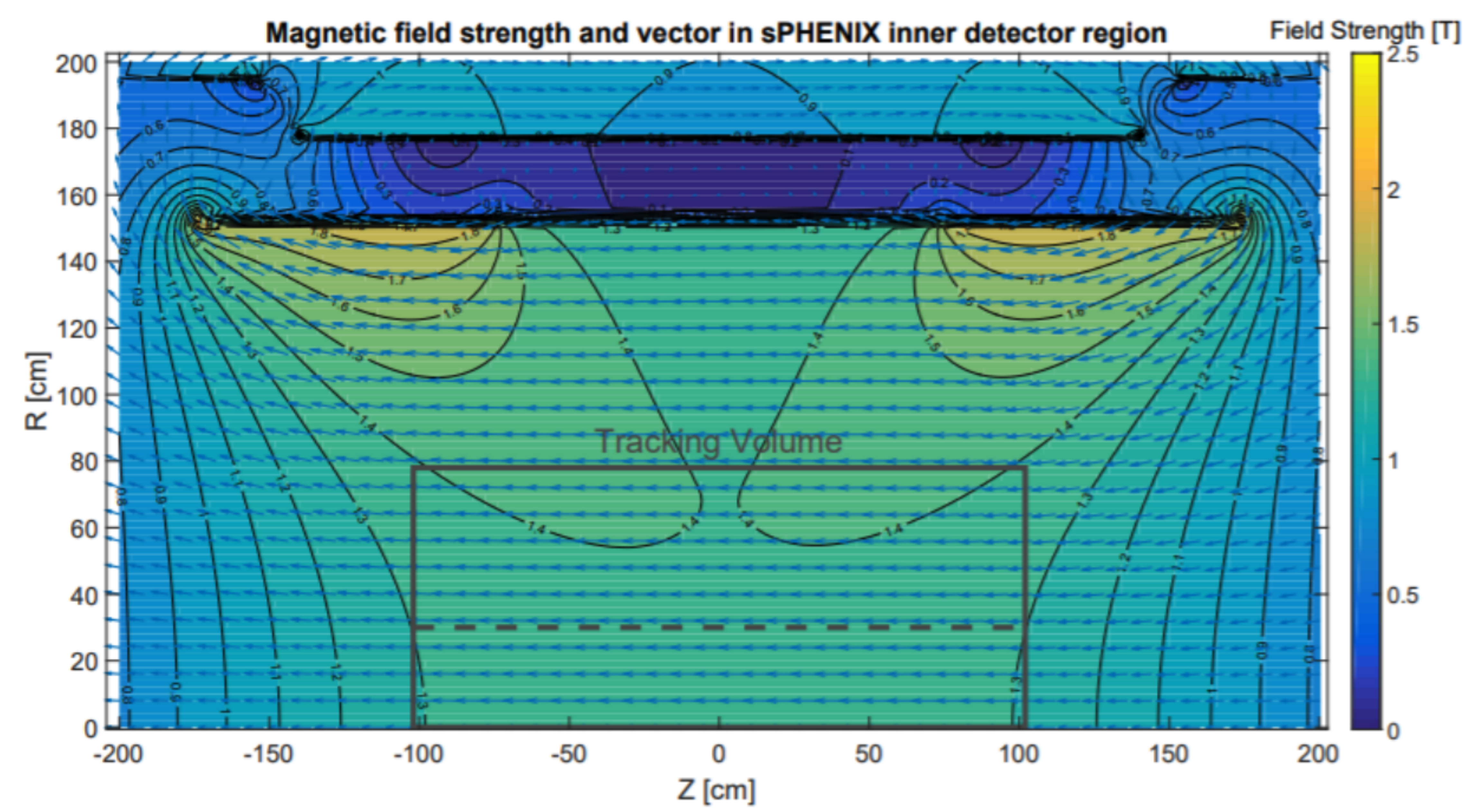
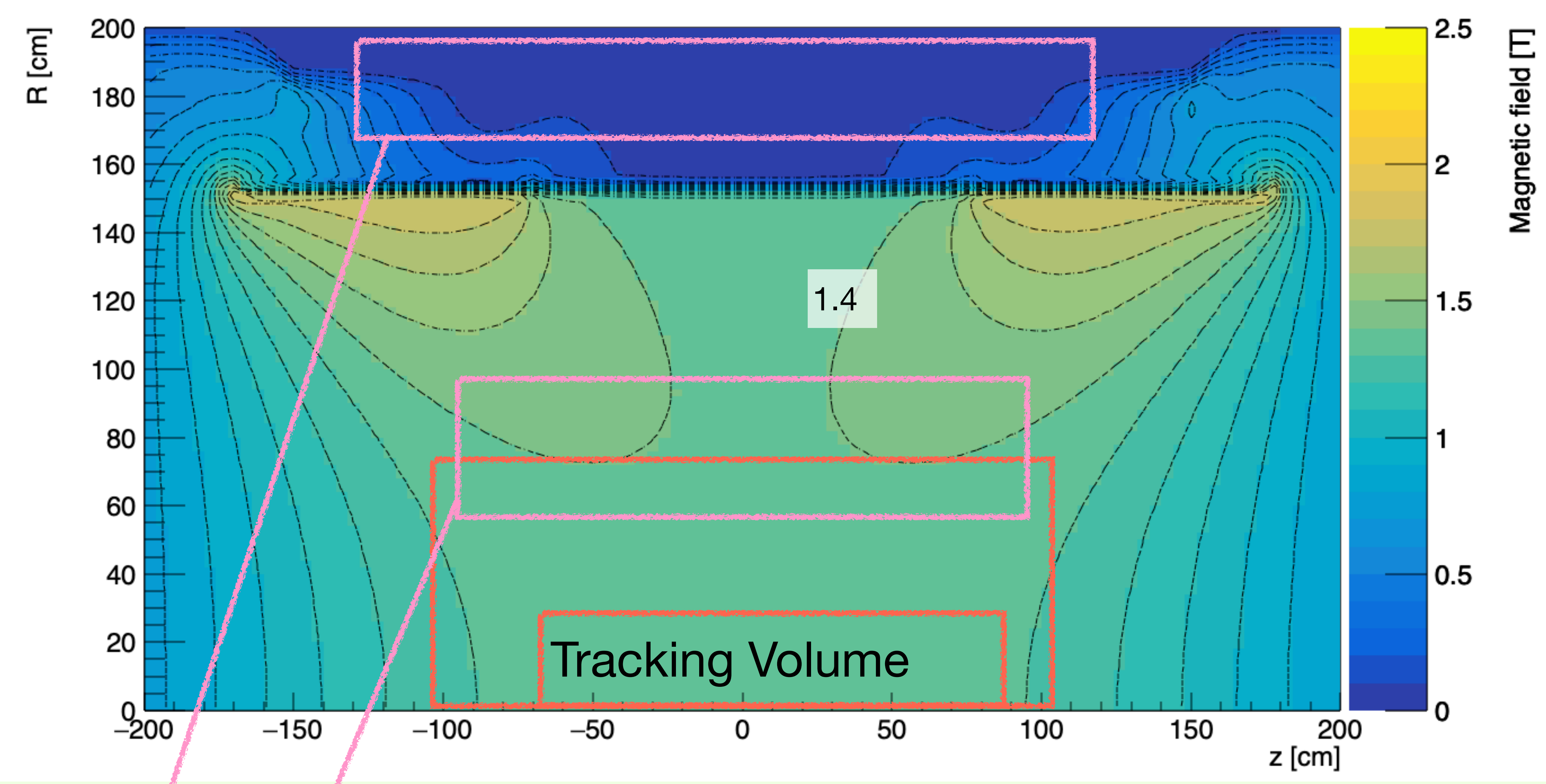


Figure 12. Field Map of the sPHENIX Solenoid

ROOT file Location

/cvmfs/sphenix.sdcc.bnl.gov/calibrations/sphnxpro/cdb/FIELDMAP_GAP/65/a9/65a930ed6de9c0e049cd0f3ef226e6b4_sphenix3dbigmapxyz_gap_rebuild_v2.root



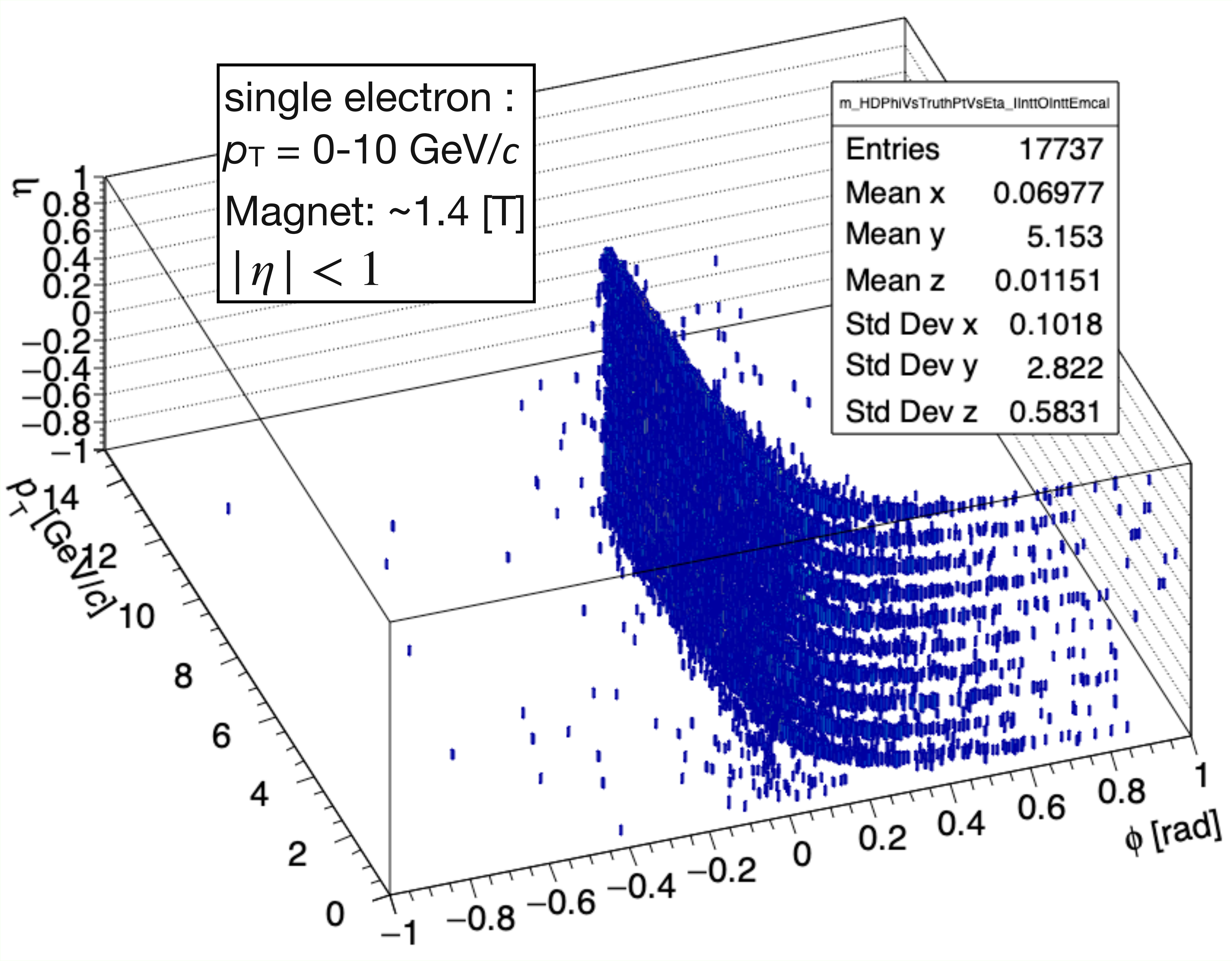
Different behavior:
However, I do not know what is this magnetic field.

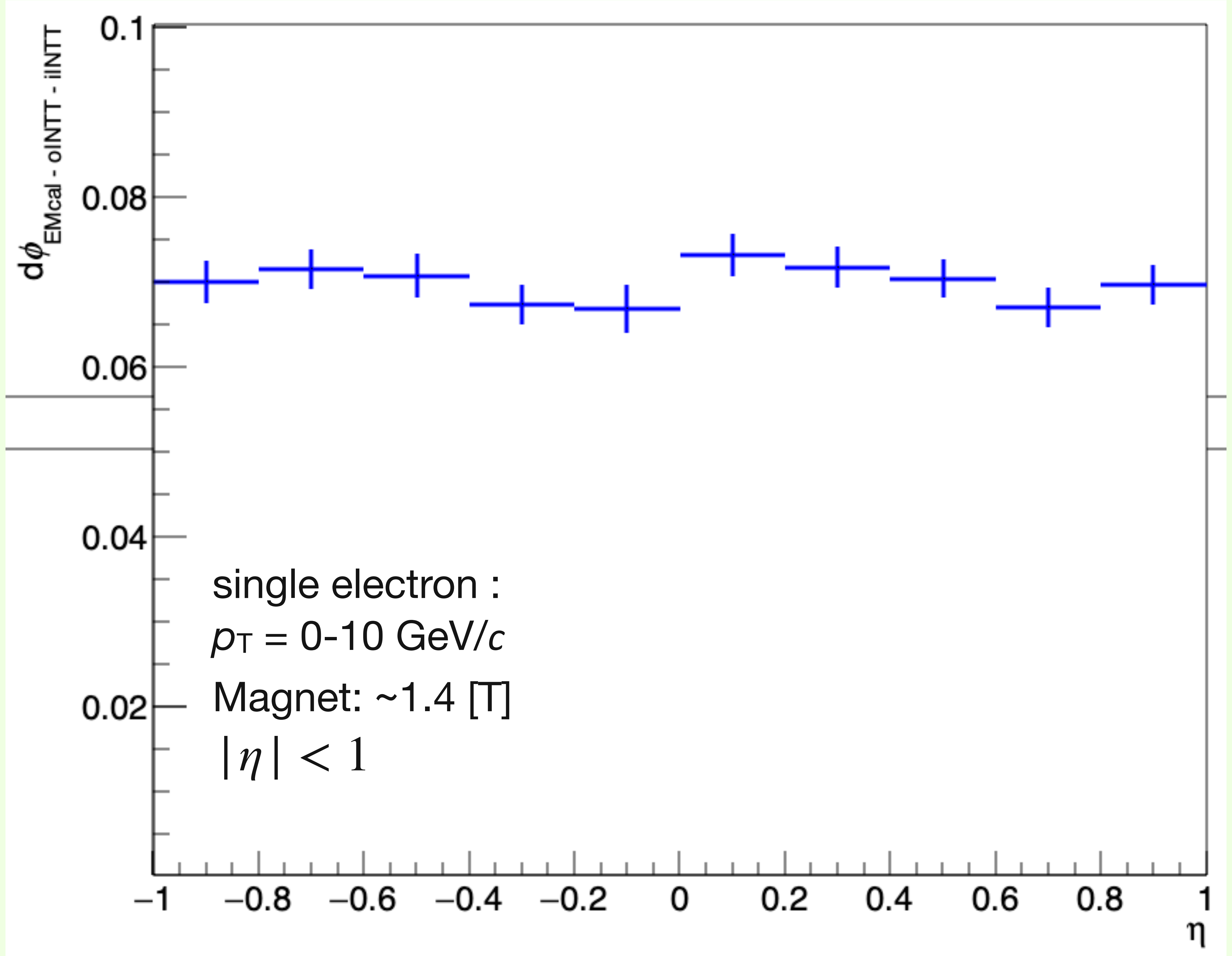
η Dependency of the Emission Angle Shift dϕ by Magnetic Field

There is a possibility that the bending of a track by magnetic field is depends on η.

a. The magnetic field is not completely uniform for η.

b. Flight length in the higher η region is longer than the smaller η one.



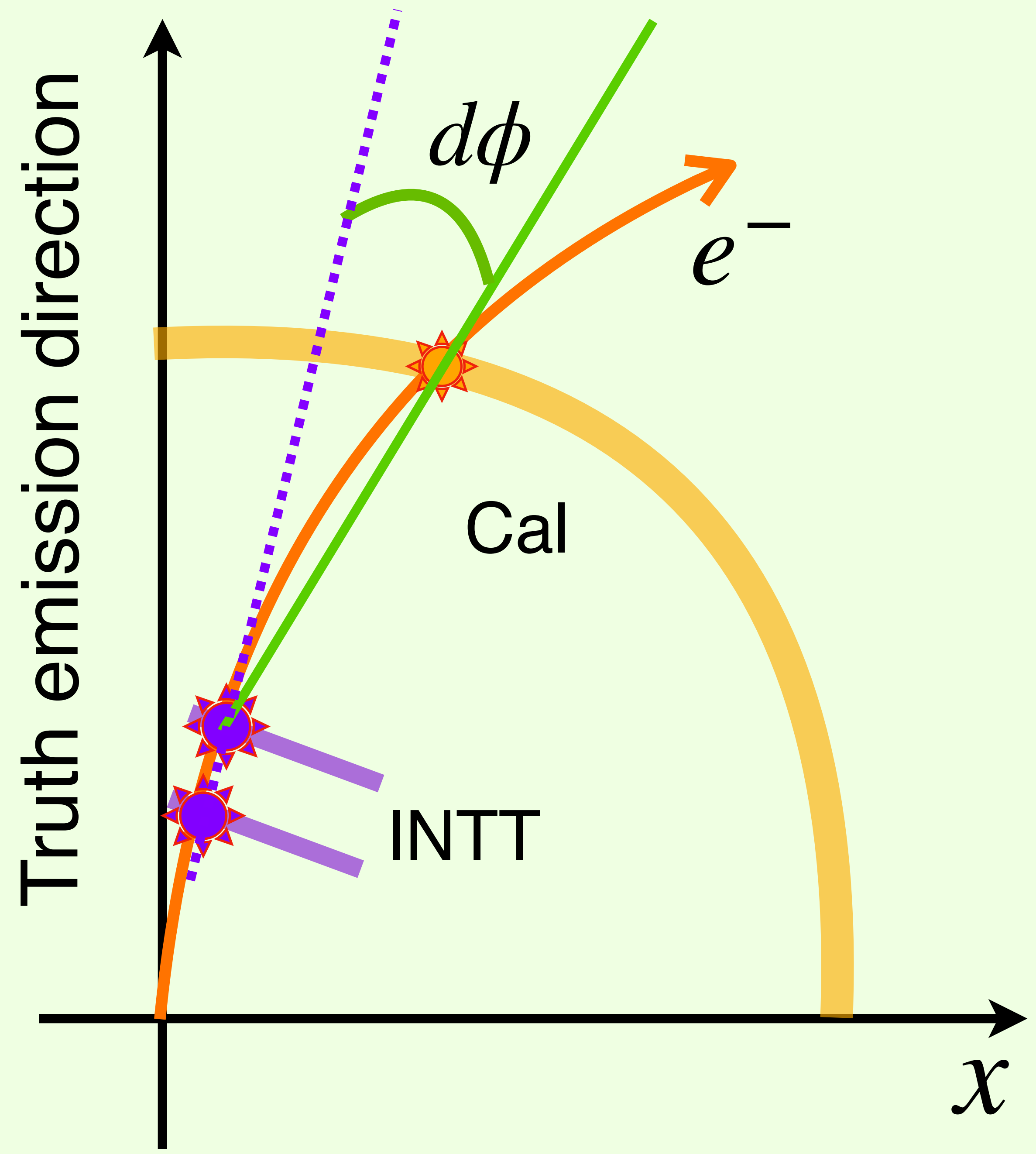


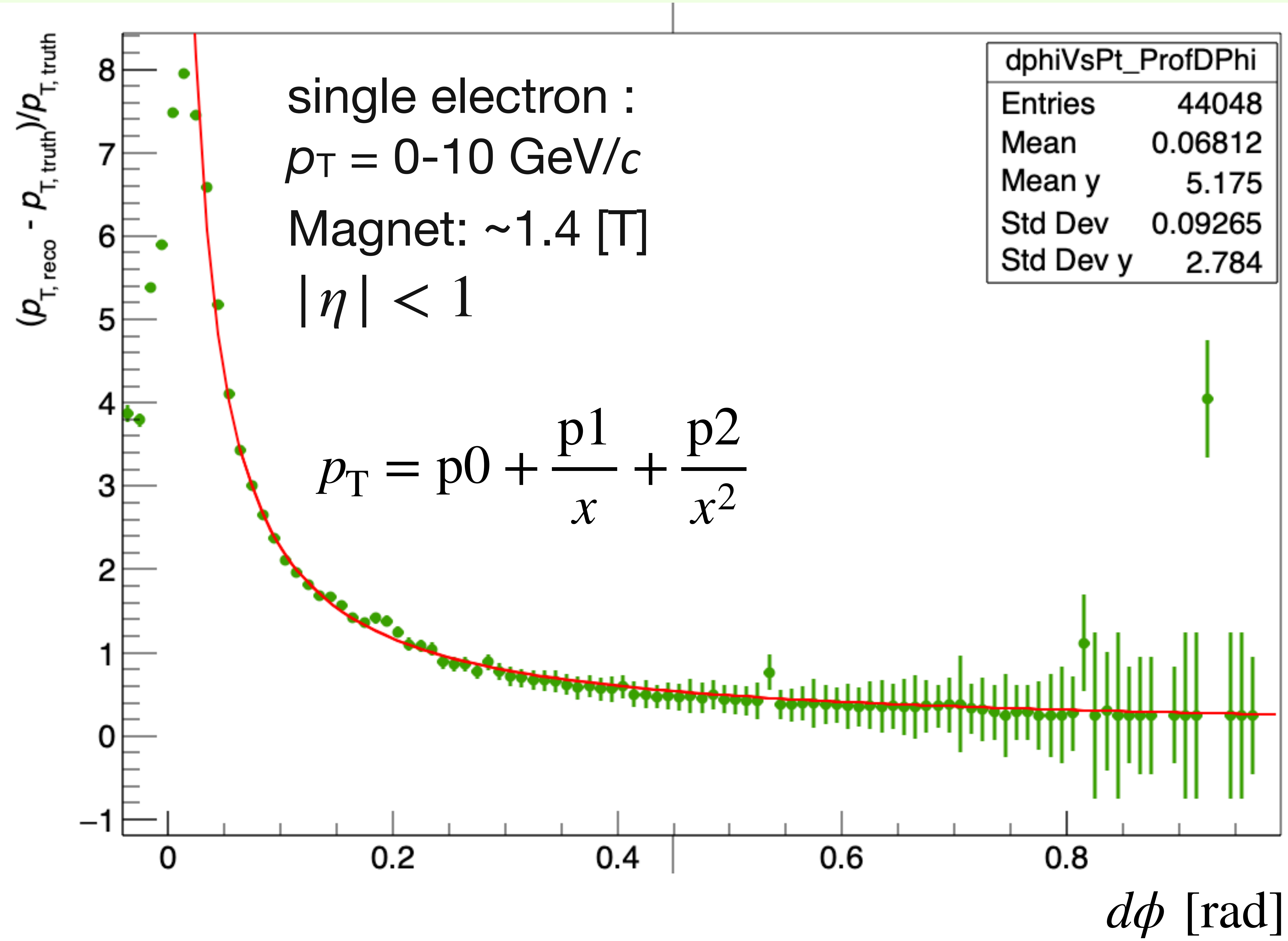
The η dependency seems negligible.

d$\phi_{\mathrm{Emcal-oIntt-iIntt}} - p_{\mathrm{T,truth}}$ fitting

Use fitting function

$$p_{\mathrm{T}} = \mathrm{p0} + \frac{\mathrm{p1}}{d\phi} + \frac{\mathrm{p2}}{d\phi^2}$$





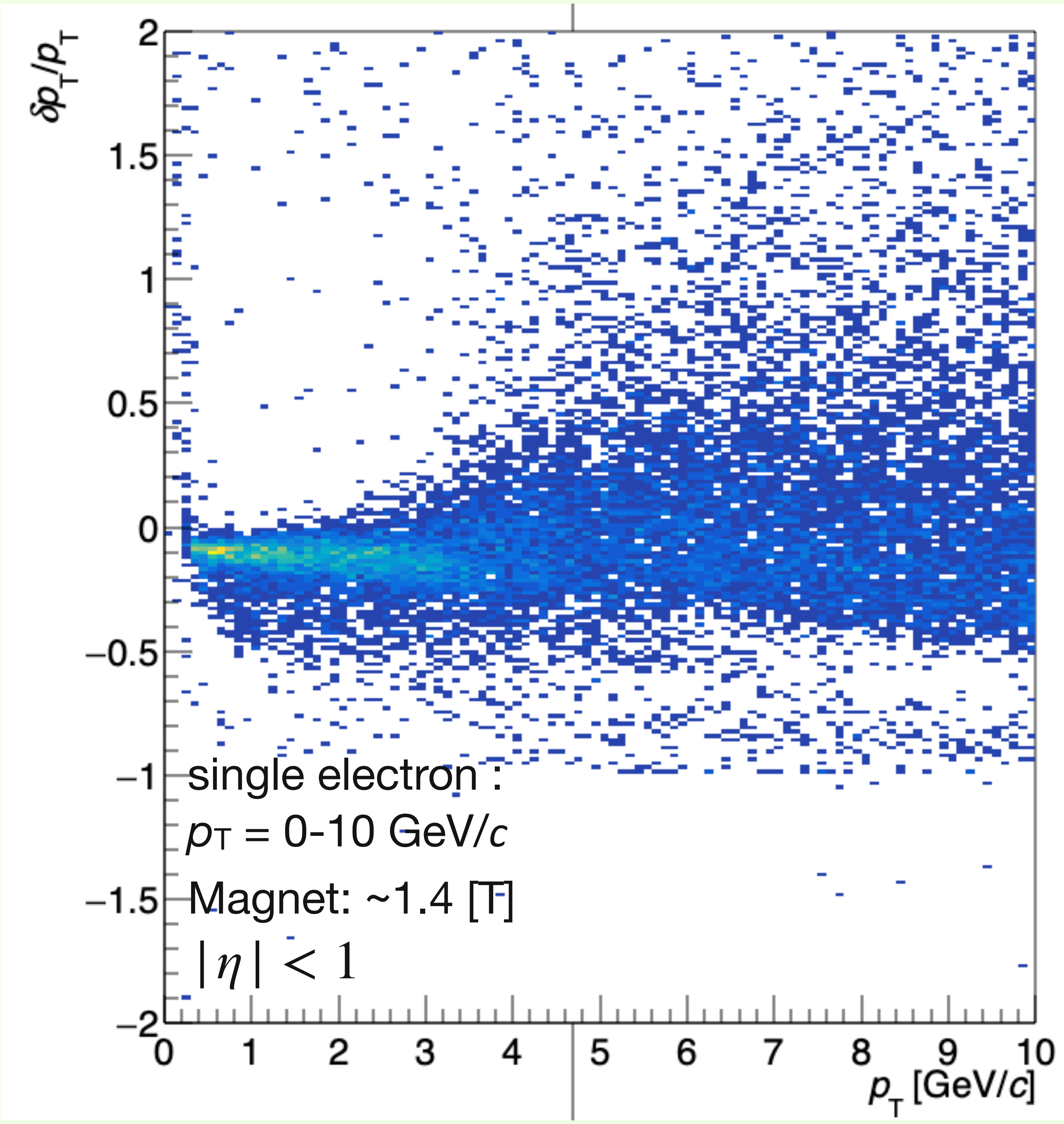
→ Apply the coefficients results into the tracking code. <FitFunctionPt(Double_t dPhi)>

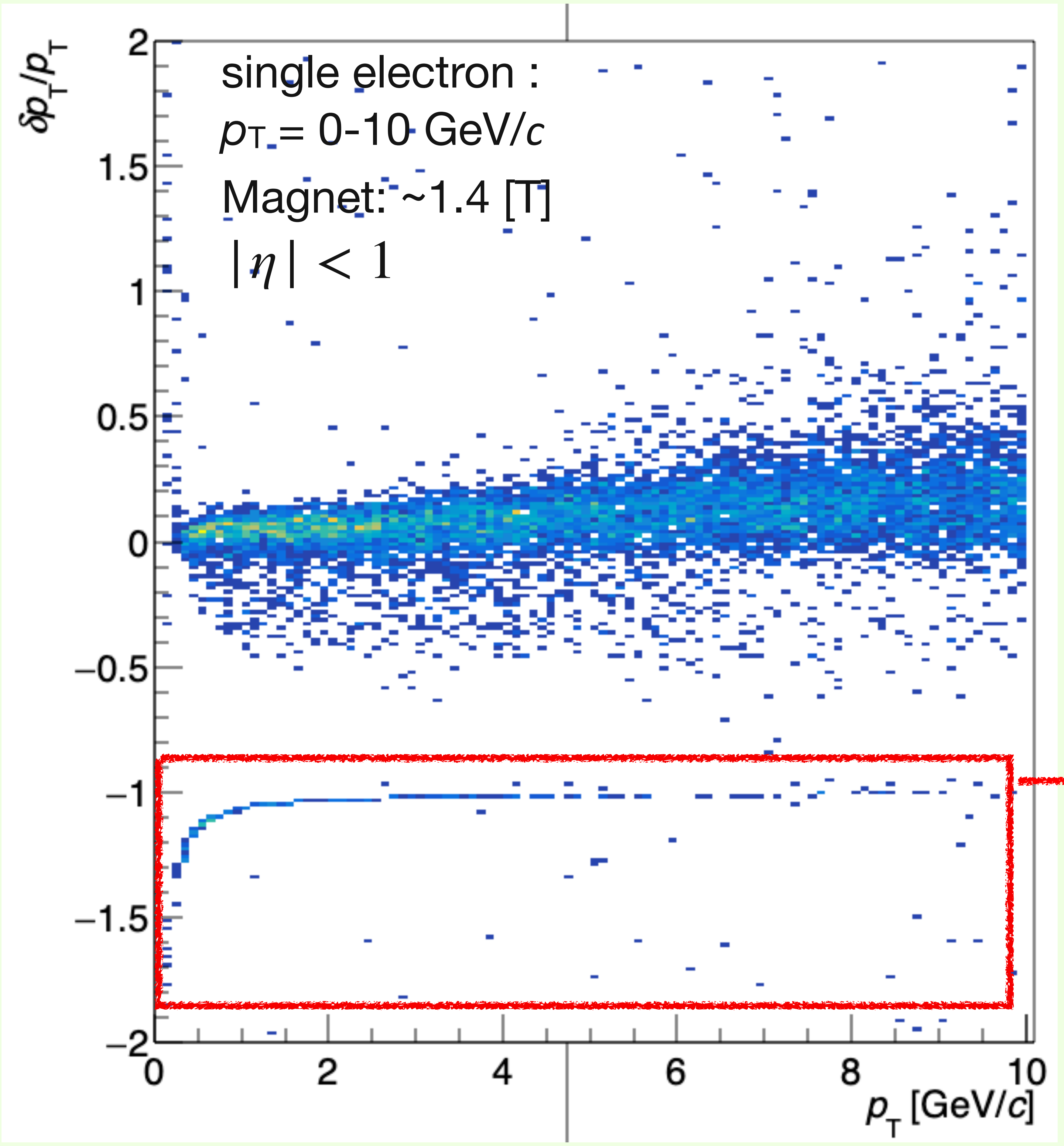
p_T resolution vs p_T

MVTX+iINTT+oINTT+EMCal

ordinal way: $p_T[\mathrm{GeV}] = qBR$

Fitting Function way: $p_T = \mathrm{p0} + \frac{\mathrm{p1}}{d\phi} + \frac{\mathrm{p2}}{d\phi^2}$

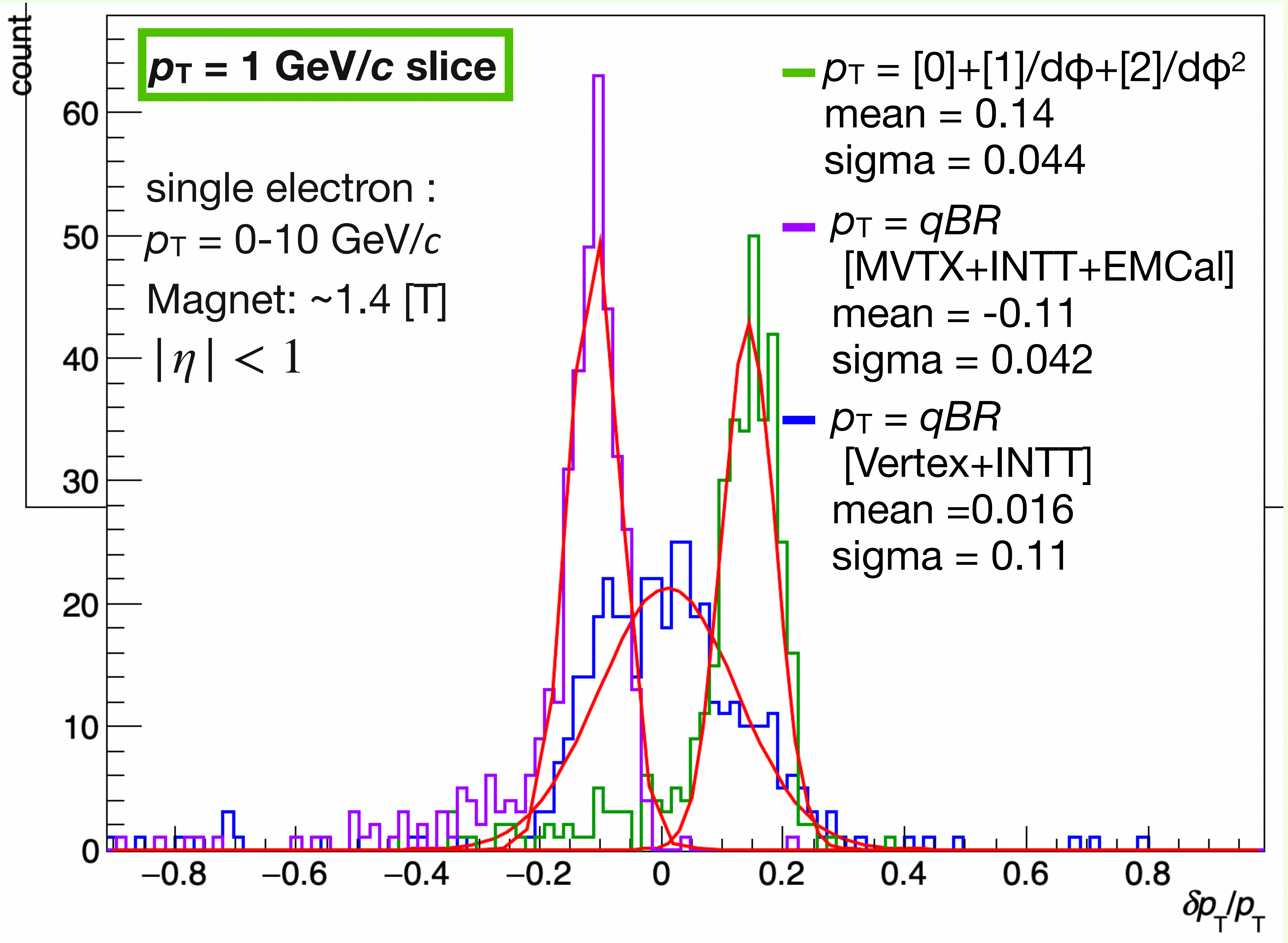




?
Probably it made by mis-tracking
(reco-truth)/truth

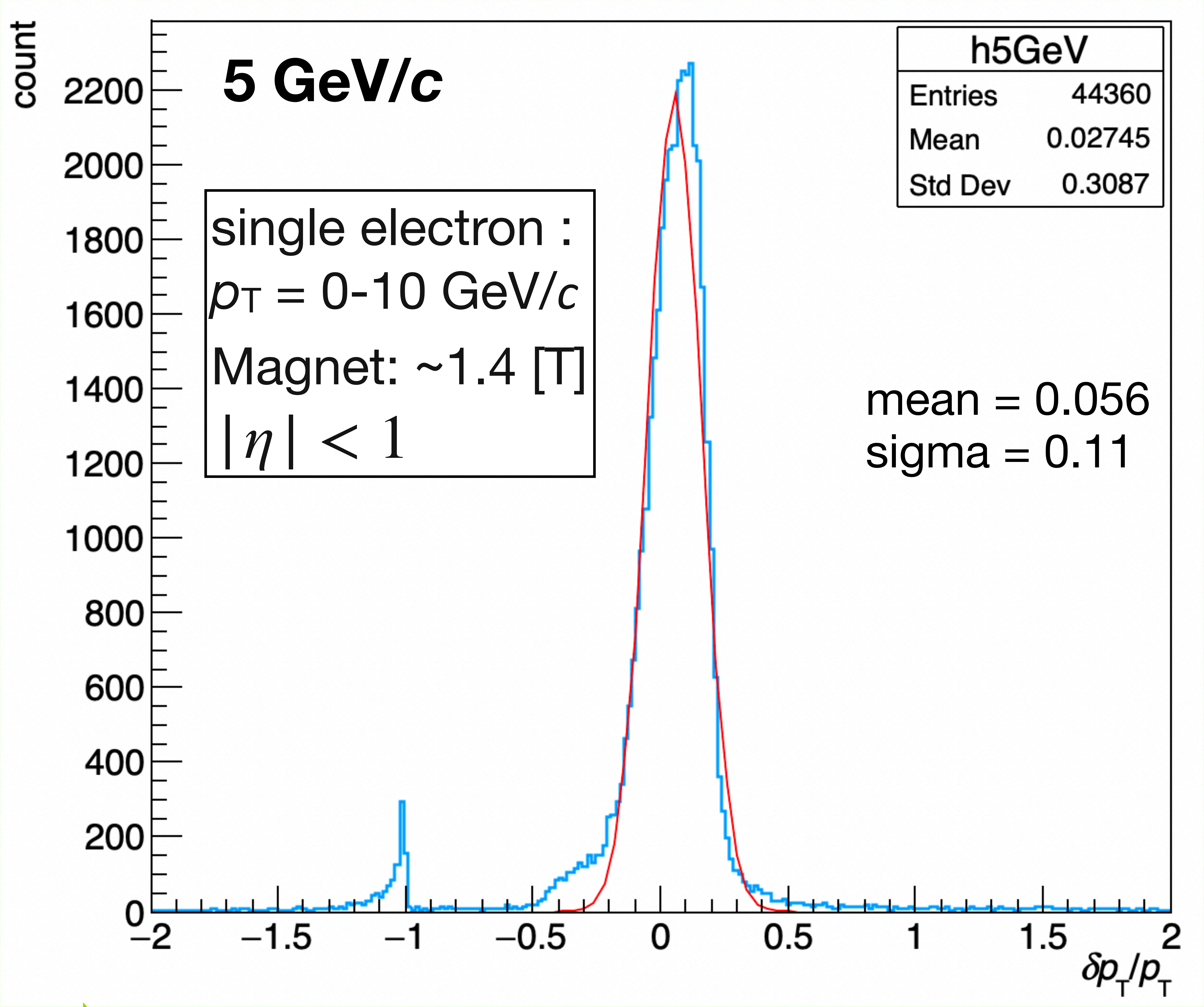
For the pT fluctuation, the fitting function way is clearly better than the ordinal way.

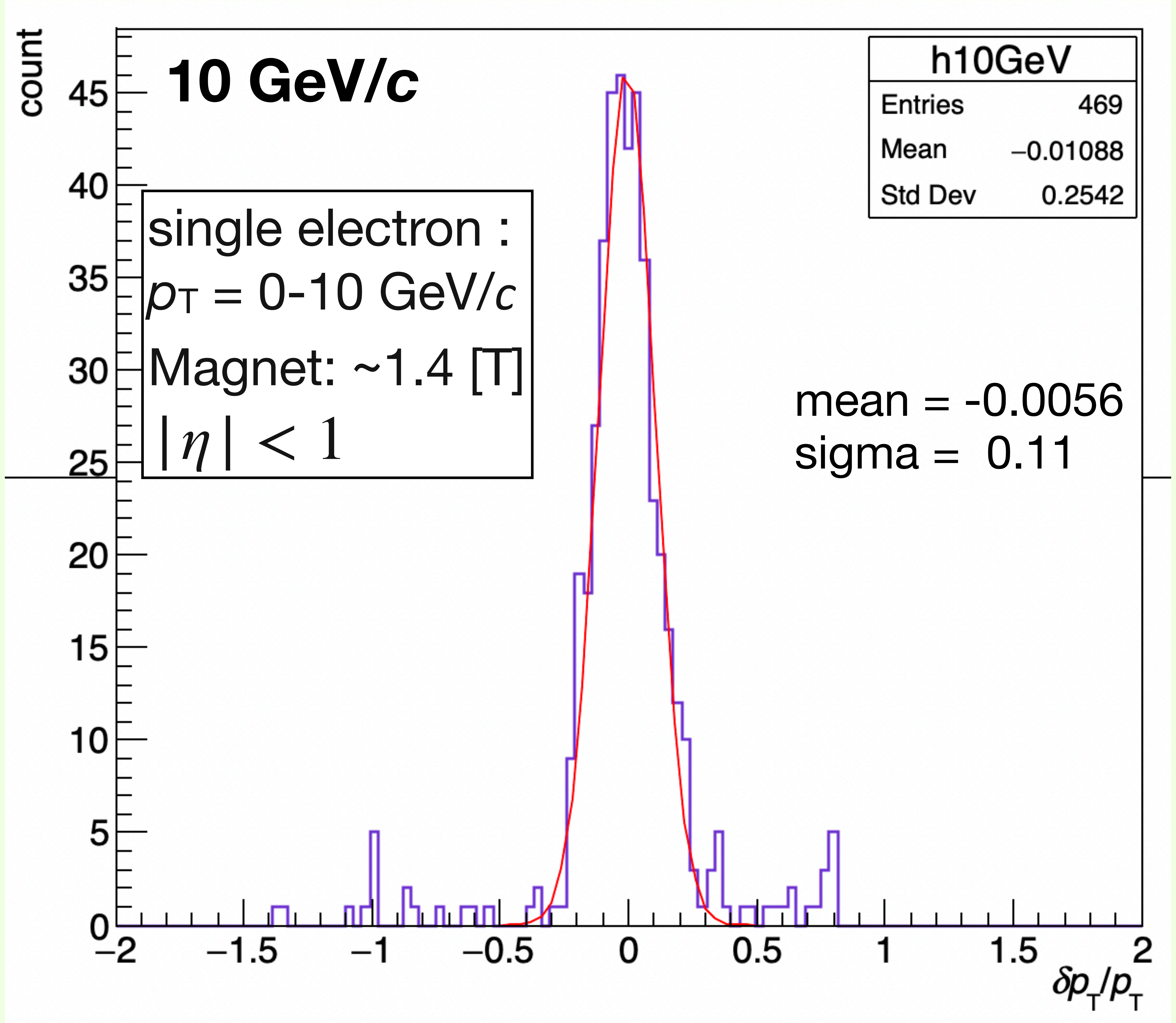
p_T resolution with fitting function



The results w/ EMCal has sufficient better resolution than only INTT.
On the other hand, there are much room to improve.

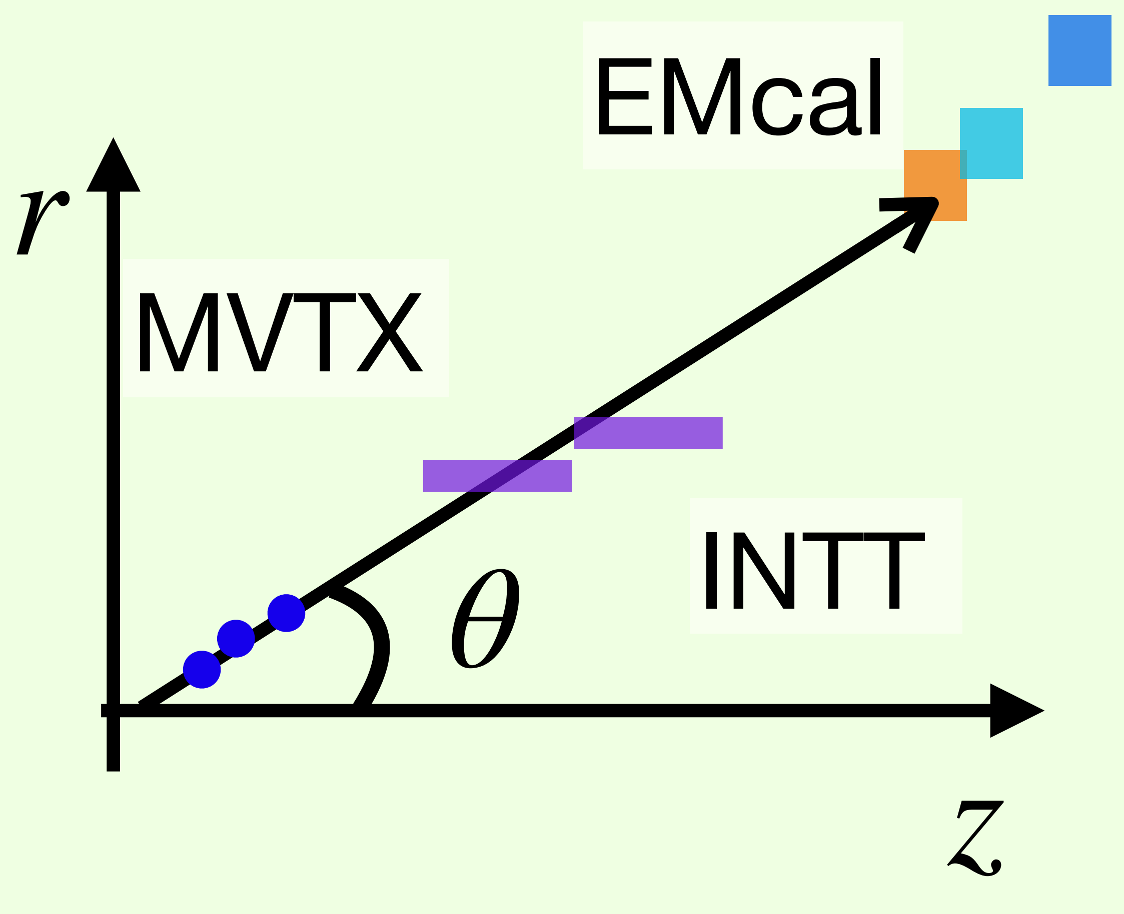
($p_{T,\,reco}$ − $p_{T,\,truth}$)/$p_{T,\,truth}$ other pT slices (fitting function way)



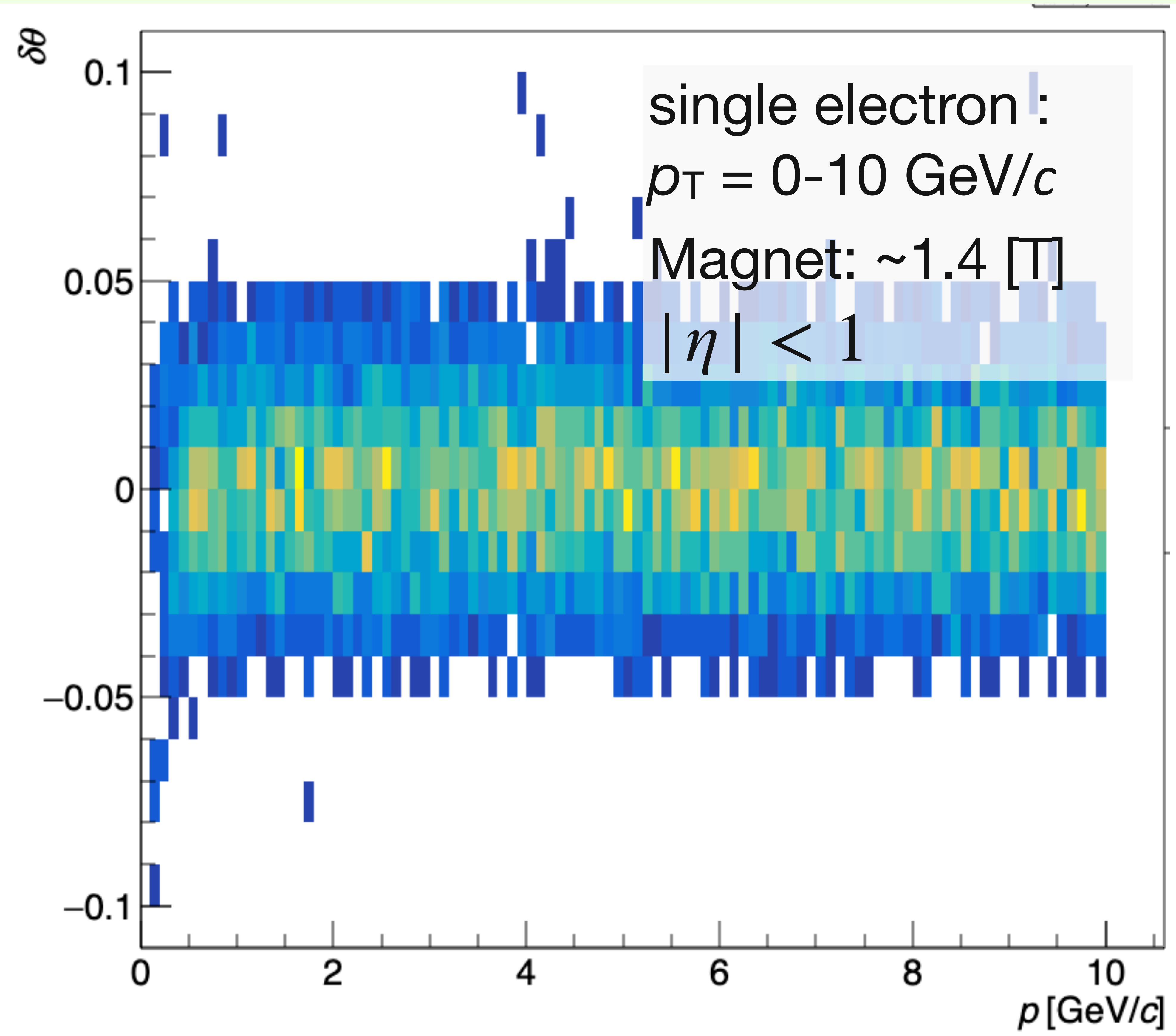


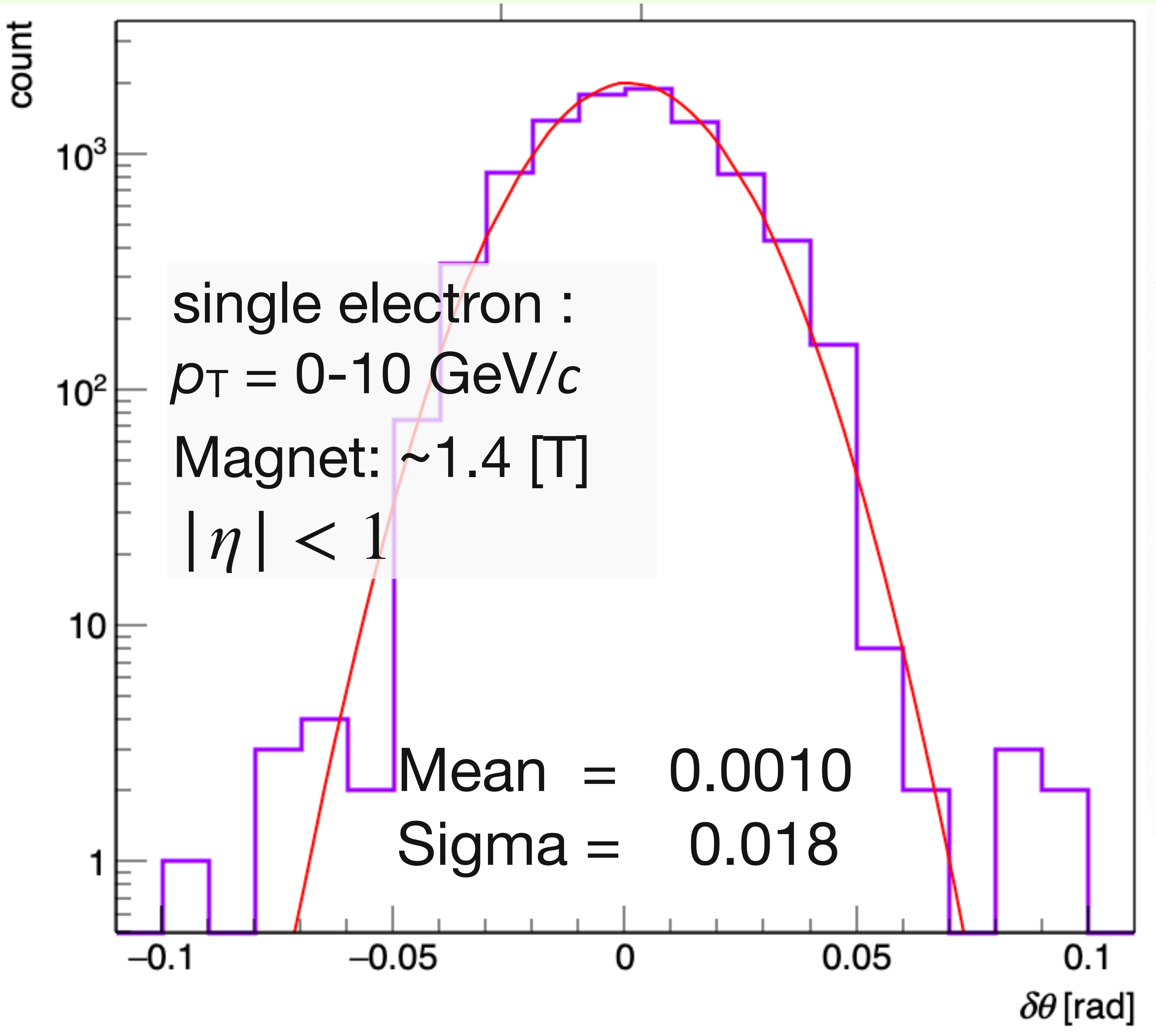
➡ The resolution becomes worse.
However, for the over pT = 5 GeV/c, the resolution keeps.

dθ (reco - truth)



MVTX+iINTT+oINTT+EMCal



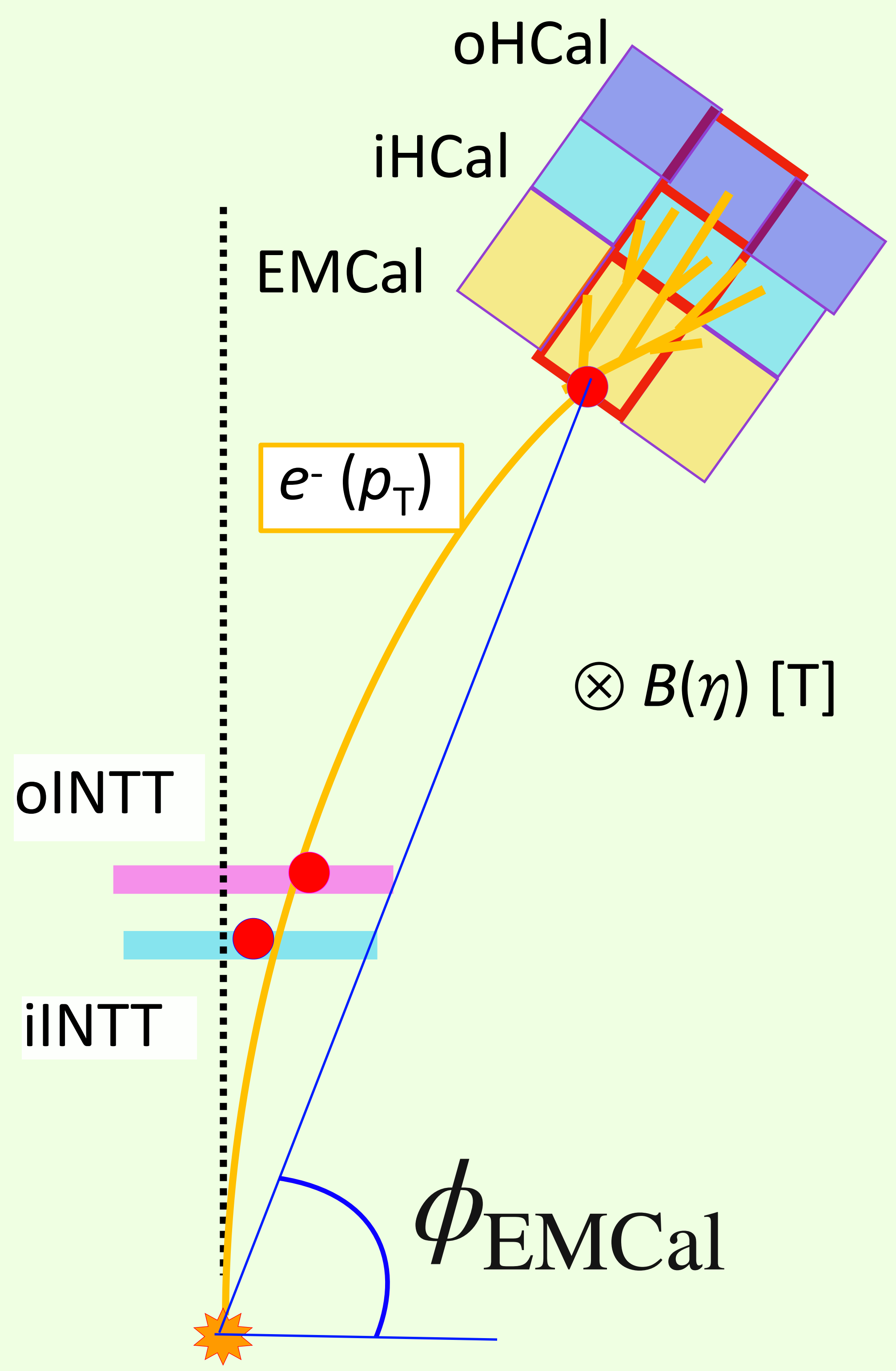


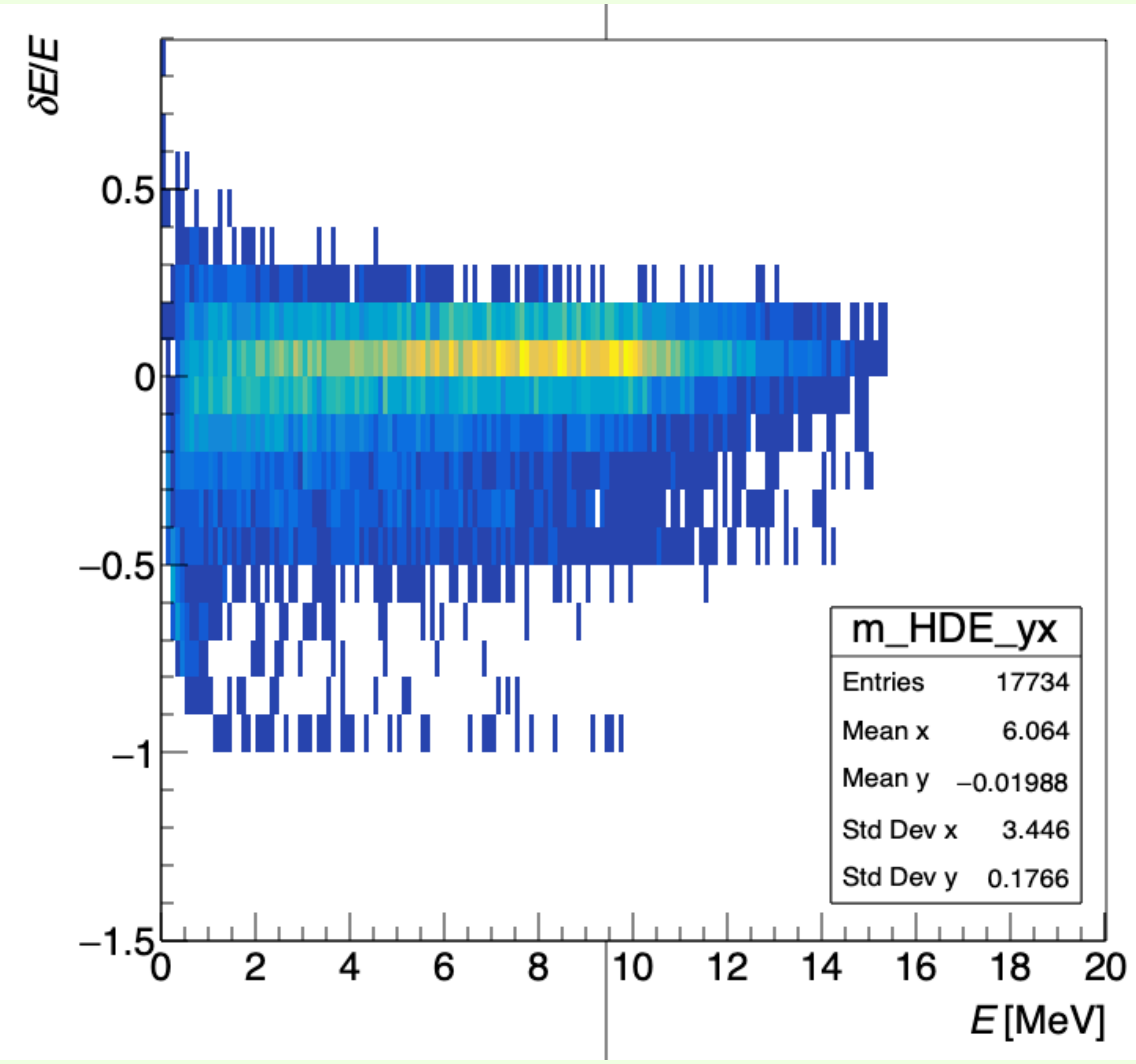
θ affect to the momentum calculation ($p = p_{\mathrm{T}}/\sin\theta$).
It is necessary to improve the quality using more sophisticated algorithm.

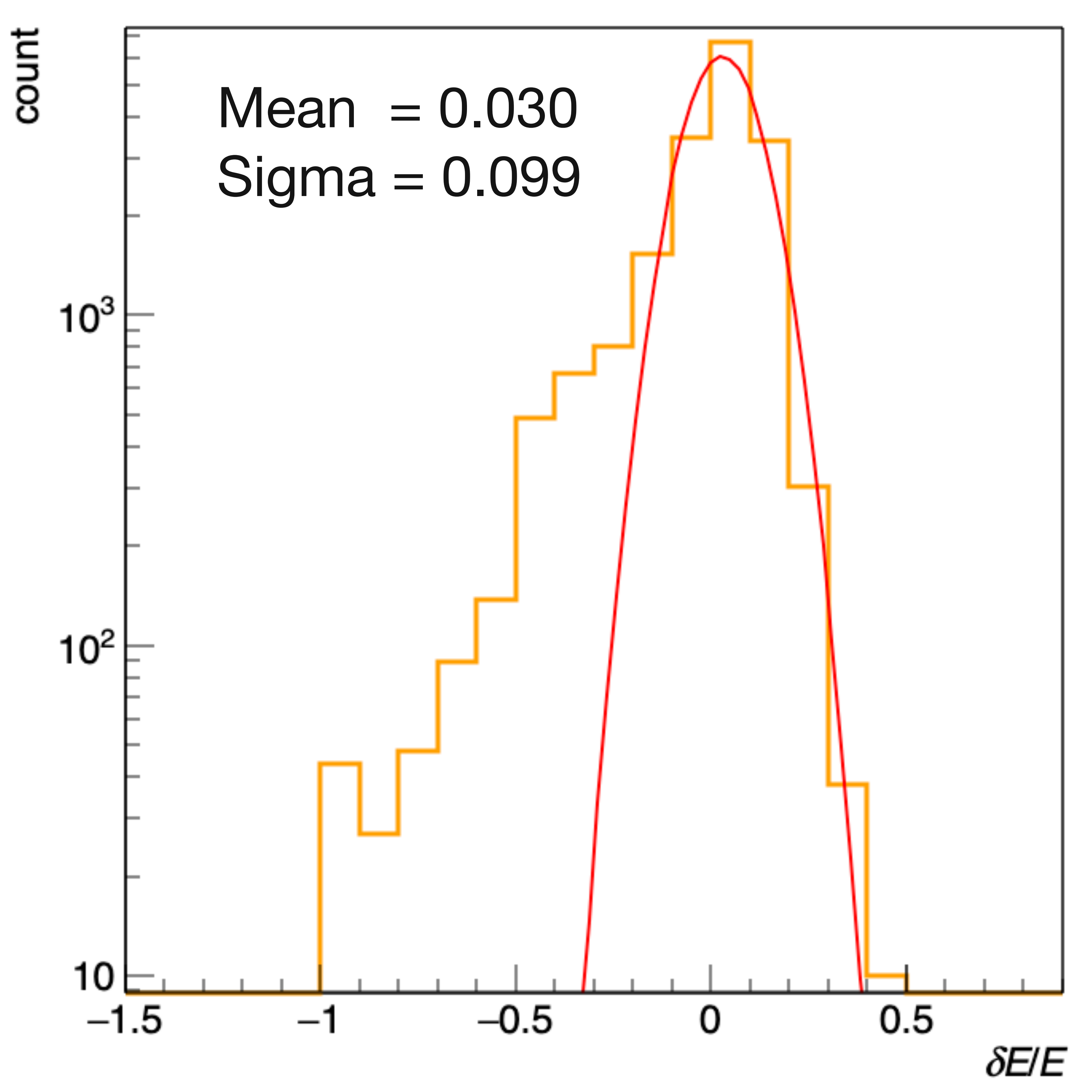
dE

The track energy use the EMCal + iHCal + oHCal.

Only the HCal cluster which locate on the closest ϕ_{EMCal} is selected.





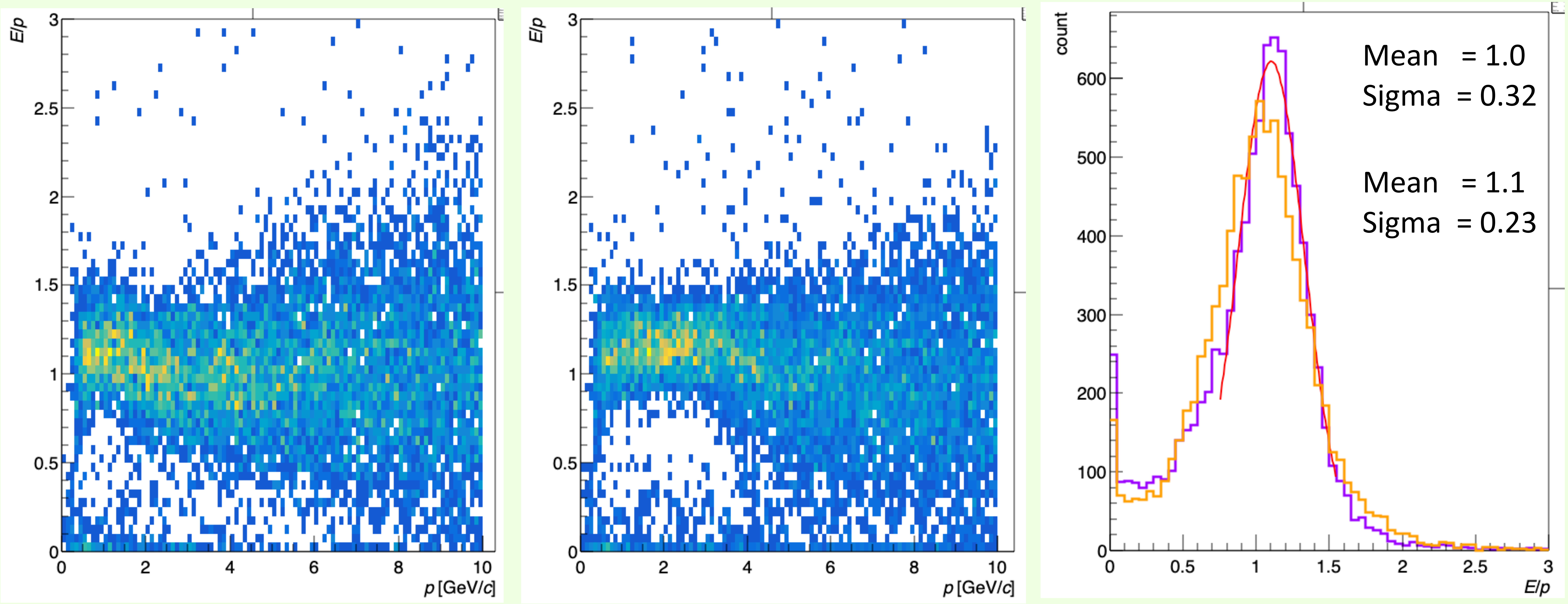


➡ The $\delta E/E$ distribution has a long negative tail.

It is necessary to cluster or merge around calorimeter clusters or towers.

E/p

iINTT+oINTT+EMCal MVTX+iINTT+oINTT+EMCal



The peak is expected around 1 in the electron case.

$(E = \sqrt{m^2 + p^2},\ \ m \ll p\ \rightarrow E = p)$

Summary

In this study, to estimate the performances of the track w/ EMCal, I made softwares to rough tracking and estimate its performances.

I could get following results:

(1) The result could show the track w/ EMCal has better p_T resolution than only INTT track.

However, this track w/ EMCal could not achieve the 1-2 percents resolution expected.
On the other hand, this result still does not include calibrated EMCal position.
Therefore, it indicates the possibility to achieve the 1-2 percents p_T resolution!

(2) The E/P distribution shows a peak around 1.

To achieve the final aim, reconstructing hadrons decaying into electrons, we need more studies.
I will show the remaining tasks in the following slides.

Remain Tasks

Remanings

1. Study the calorimeter clustering algorithm.

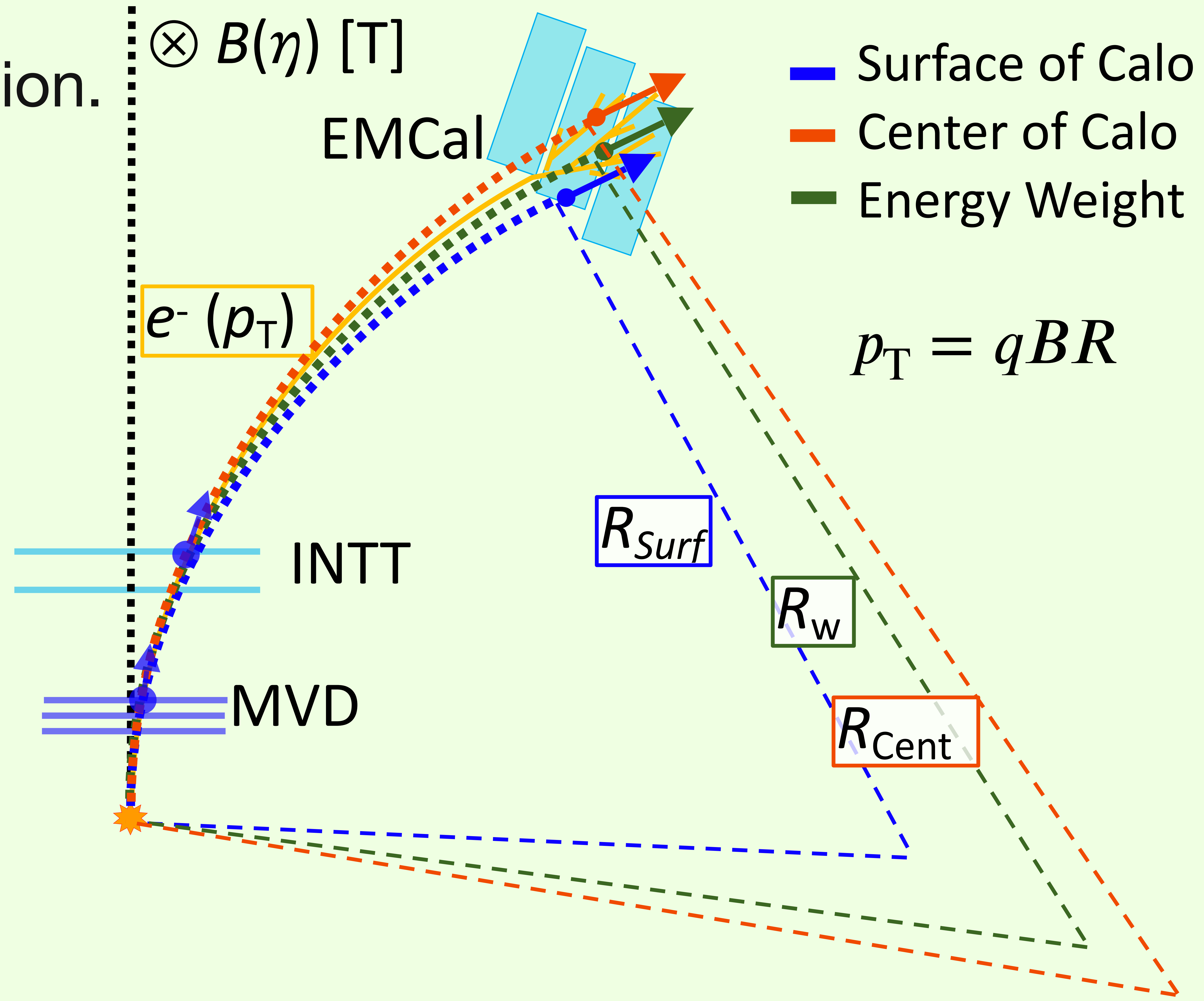
2. Implement a more sophisticated hit matching algorithm.

3. Estimate tracking efficiency.

4. Run other particles simulation. (Hadrons)

5. Multi-particles simulation (ex: PYTHIA)

Clastering way and the postion is very sensitive for the p_T estimation.
→ Need to modify clustering or shift position.



Calorimeter Clustering code

RawClusterContainer (← Now I am using):
https://sphenix-collaboration.github.io/doxygen/d6/d12/classRawClusterContainer.html

RawCluster:
https://sphenix-collaboration.github.io/doxygen/d2/d4e/classRawCluster.html

I think the algorithm making calorimeter cluster is written in the "ClusterBuilder" source codes .
However, there are three codes having the name "ClusterBuilder"…

The RawClusterPositionCorrection seems important for tracking…
However, I have not yet read it.

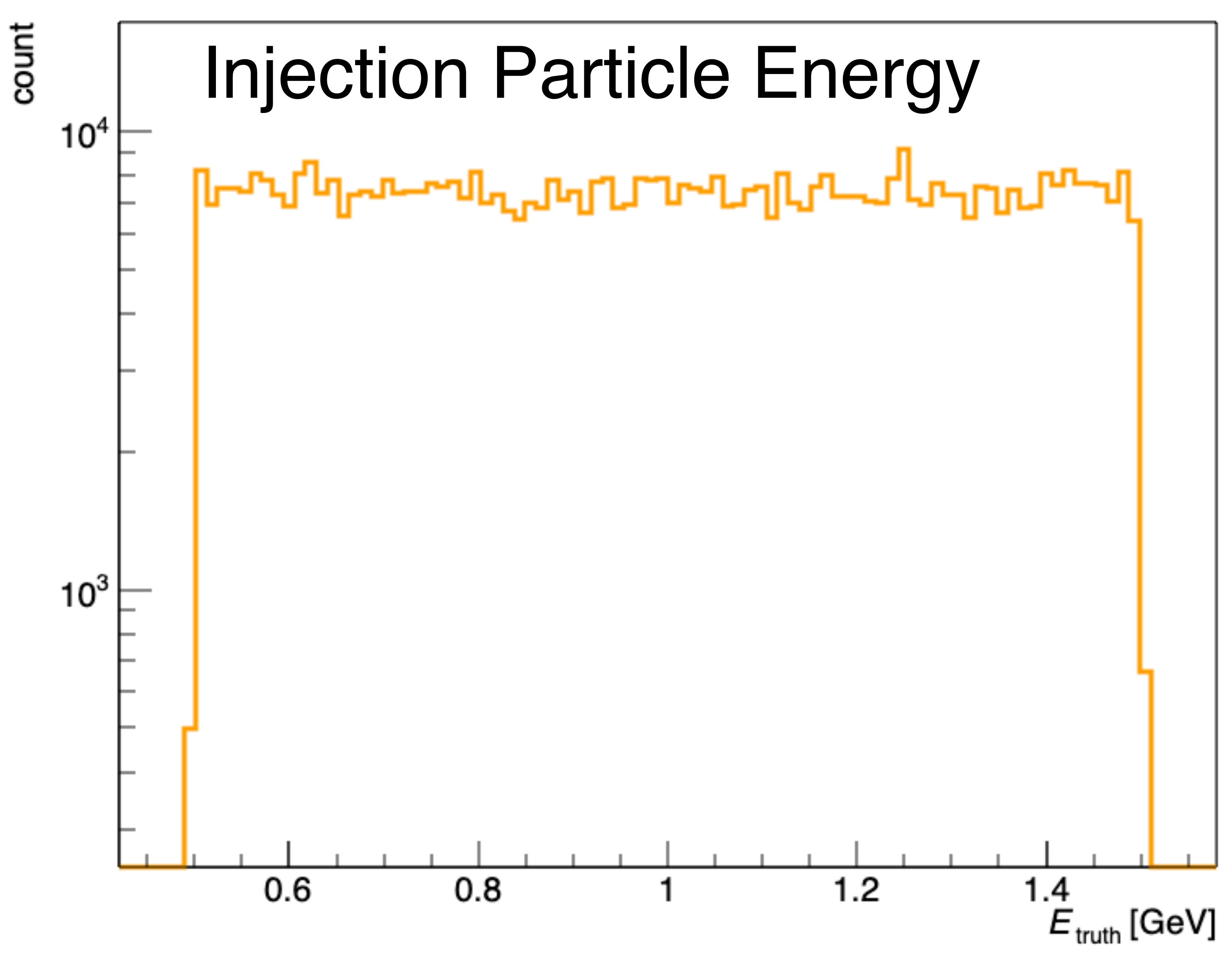
- RawClusterBuilderGraph.cc
- RawClusterBuilderGraph.h
- RawClusterBuilderTemplate.cc
- RawClusterBuilderTemplate.h
- RawClusterBuilderTopo.cc
- RawClusterBuilderTopo.h
- RawClusterCNNClassifier.cc
- RawClusterCNNClassifier.h
- RawClusterDeadHotMask.cc
- RawClusterDeadHotMask.h
- RawClusterPositionCorrection.cc
- RawClusterPositionCorrection.h
- RawClusterZVertexRecorrect.cc
- RawClusterZVertexRecorrect.h

https://github.com/sPHENIX-Collaboration/coresoftware/tree/02804b5a691b92395e4aae83726ae2c04979c0e2/offline/packages/CaloReco

Single electron generator simulation

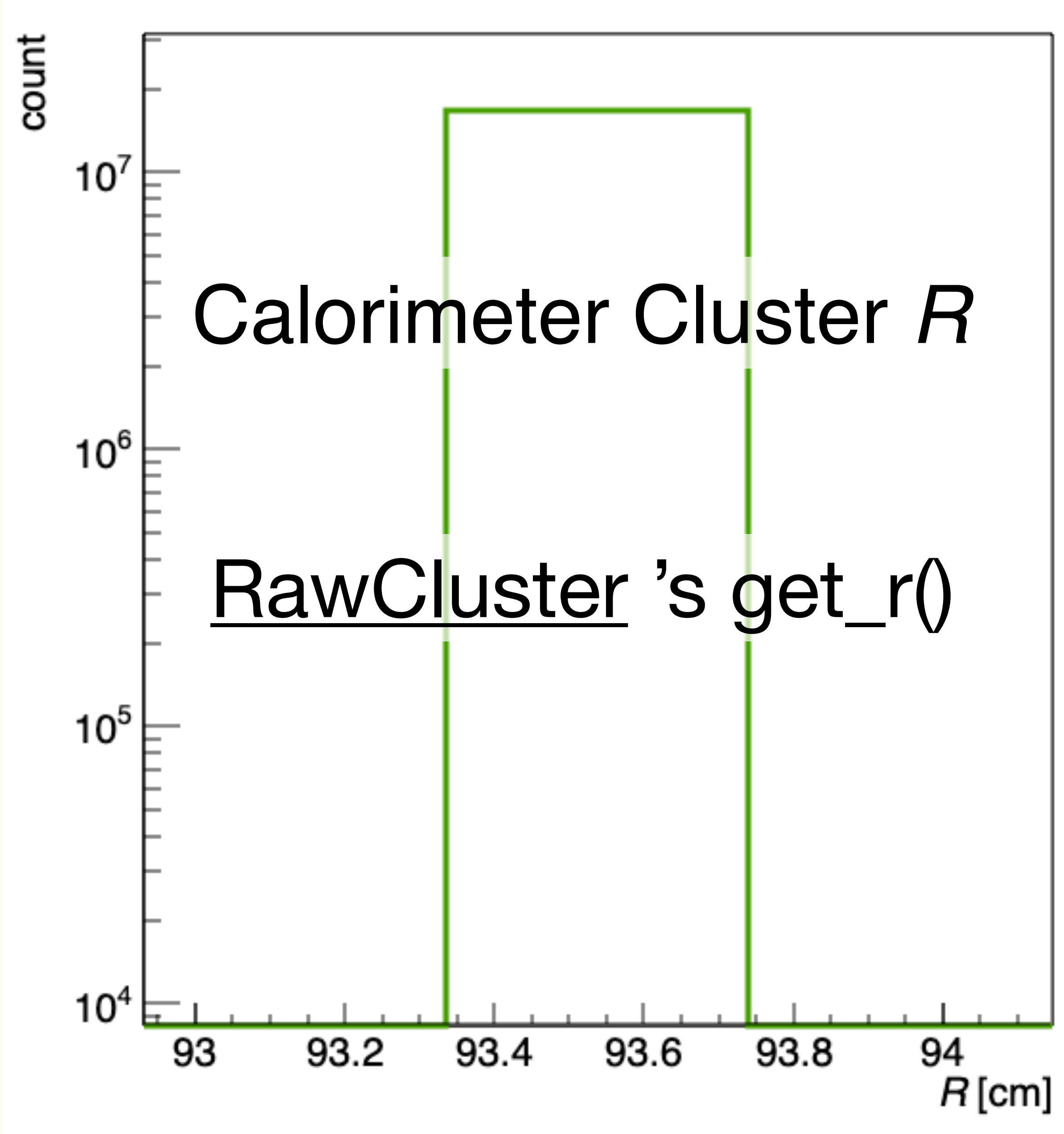
injection p_T: 0.5-1.5 [GeV/c]

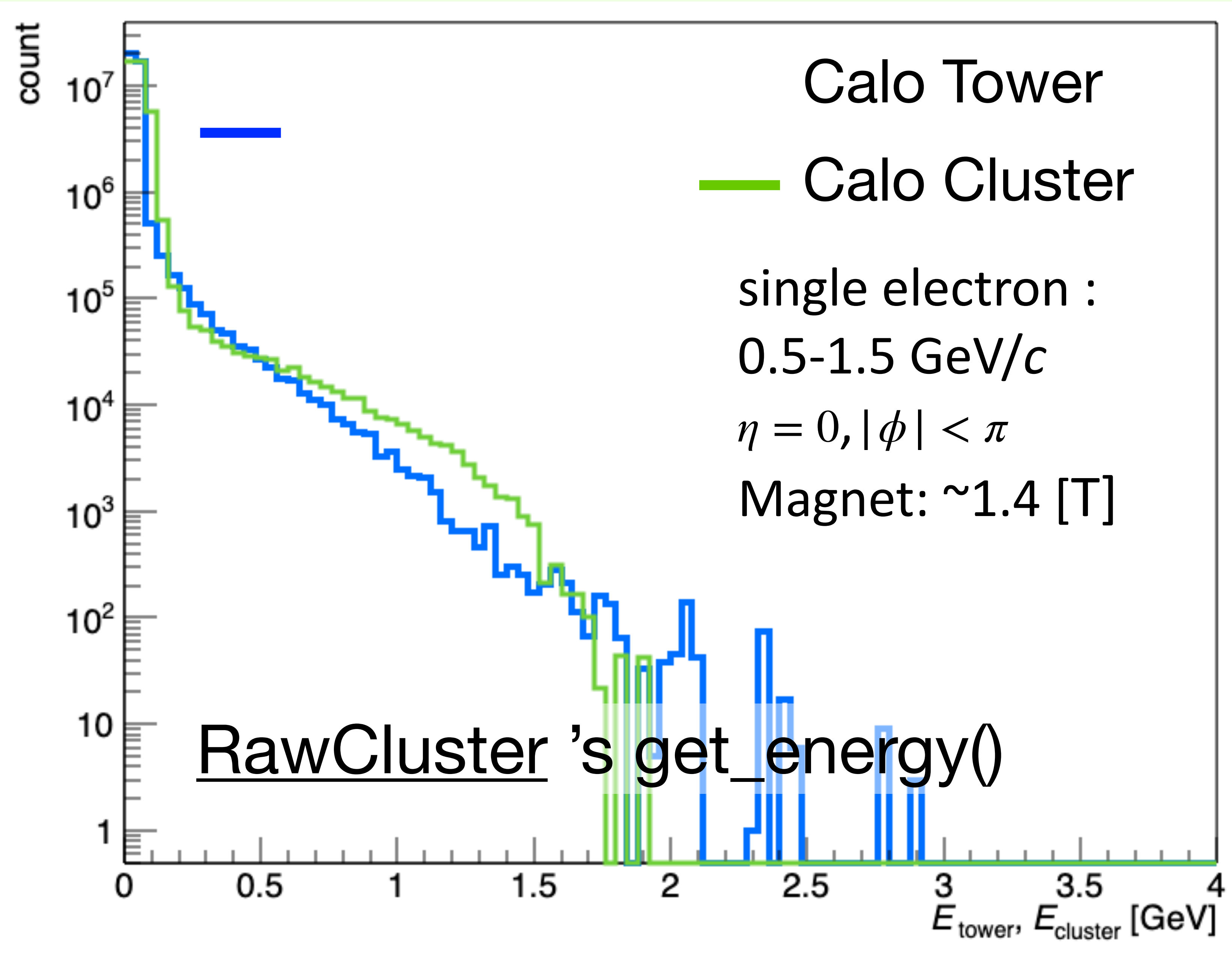
$\eta = 0, |\phi| < \pi$



Calorimeter Tower
("TOWERINFO_CALIB_CEMC")
<TowerInfo>

Calorimeter Cluster
(“CLUSTER_CEMC")
<RawCluster>





Tracking Fail Events Ratio

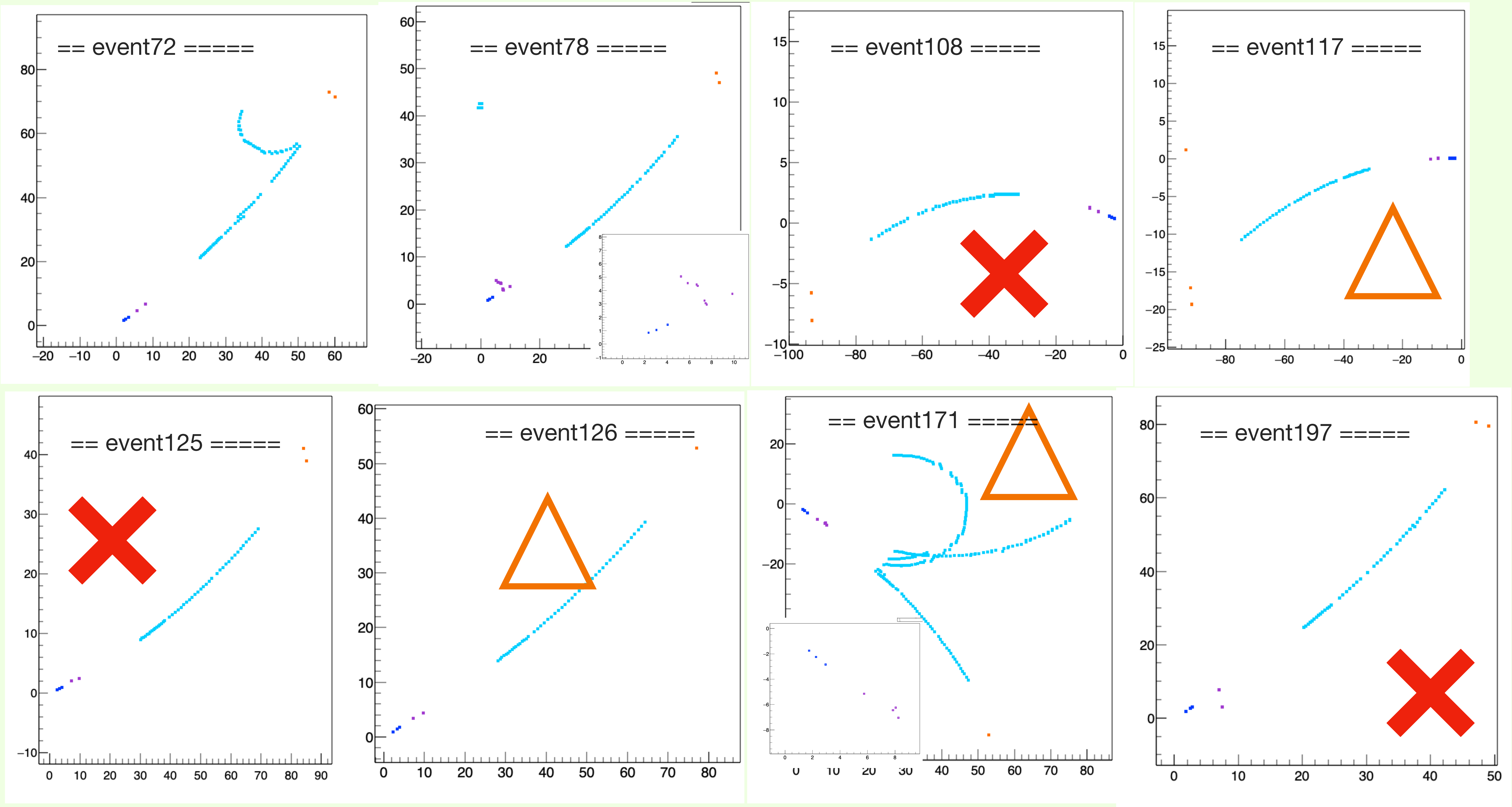
Event Ratio

(1) Only single INTT Cluster (only iINTT or oINTT): ~7%

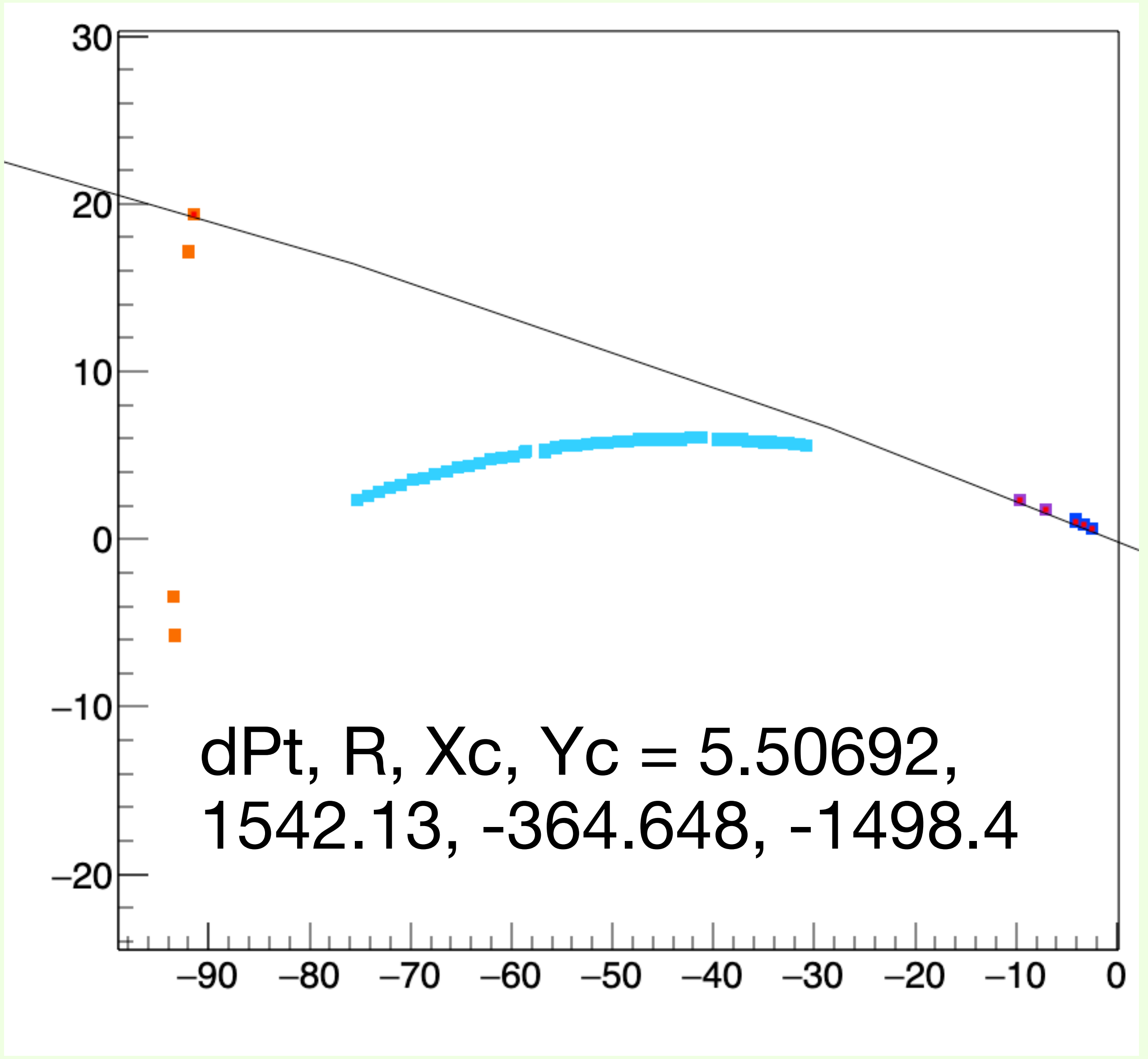
(2) Matching fail by the algorithm reason: ~2%

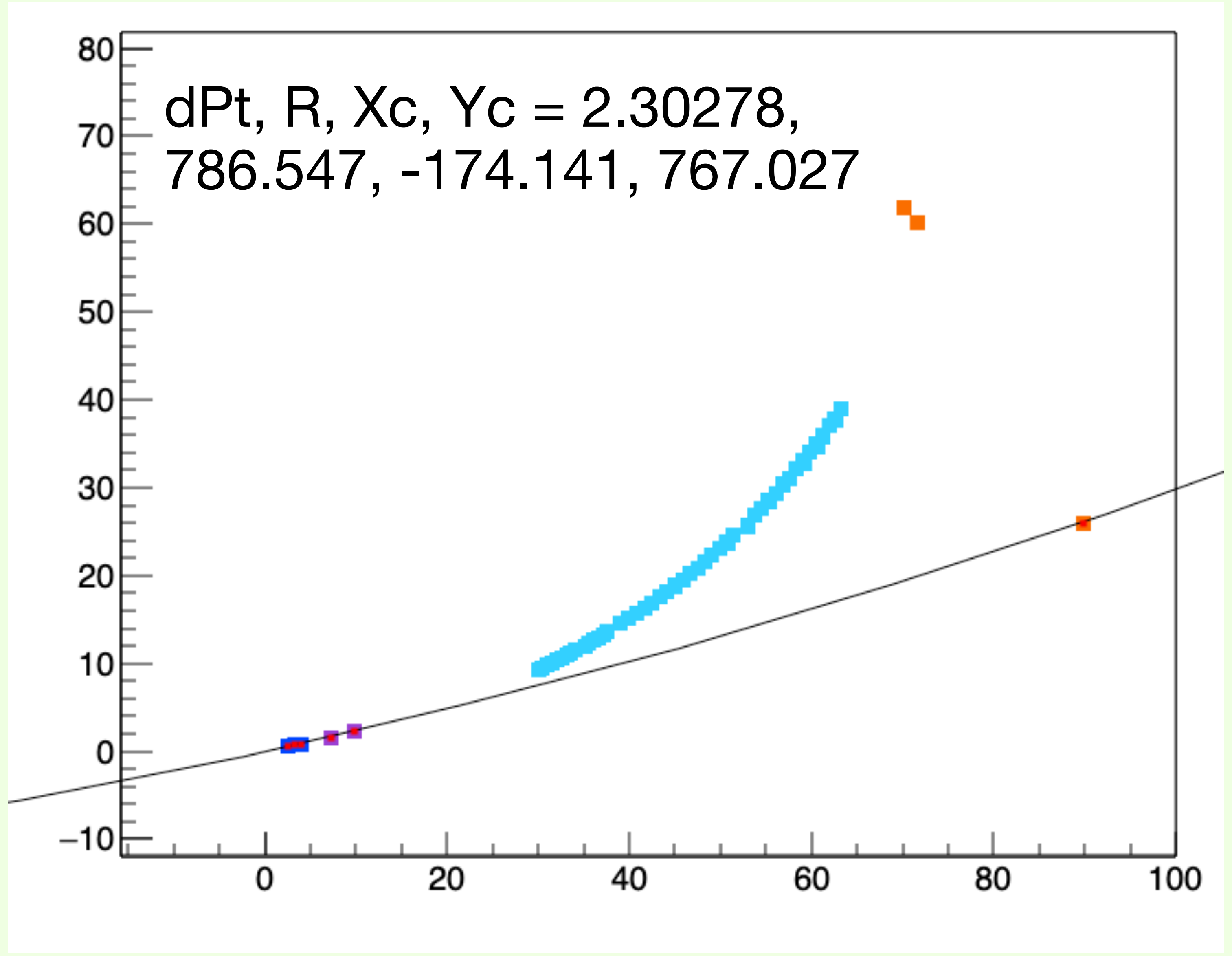
(3) Large p_T tracking: ~8% (by mostly decayed events)

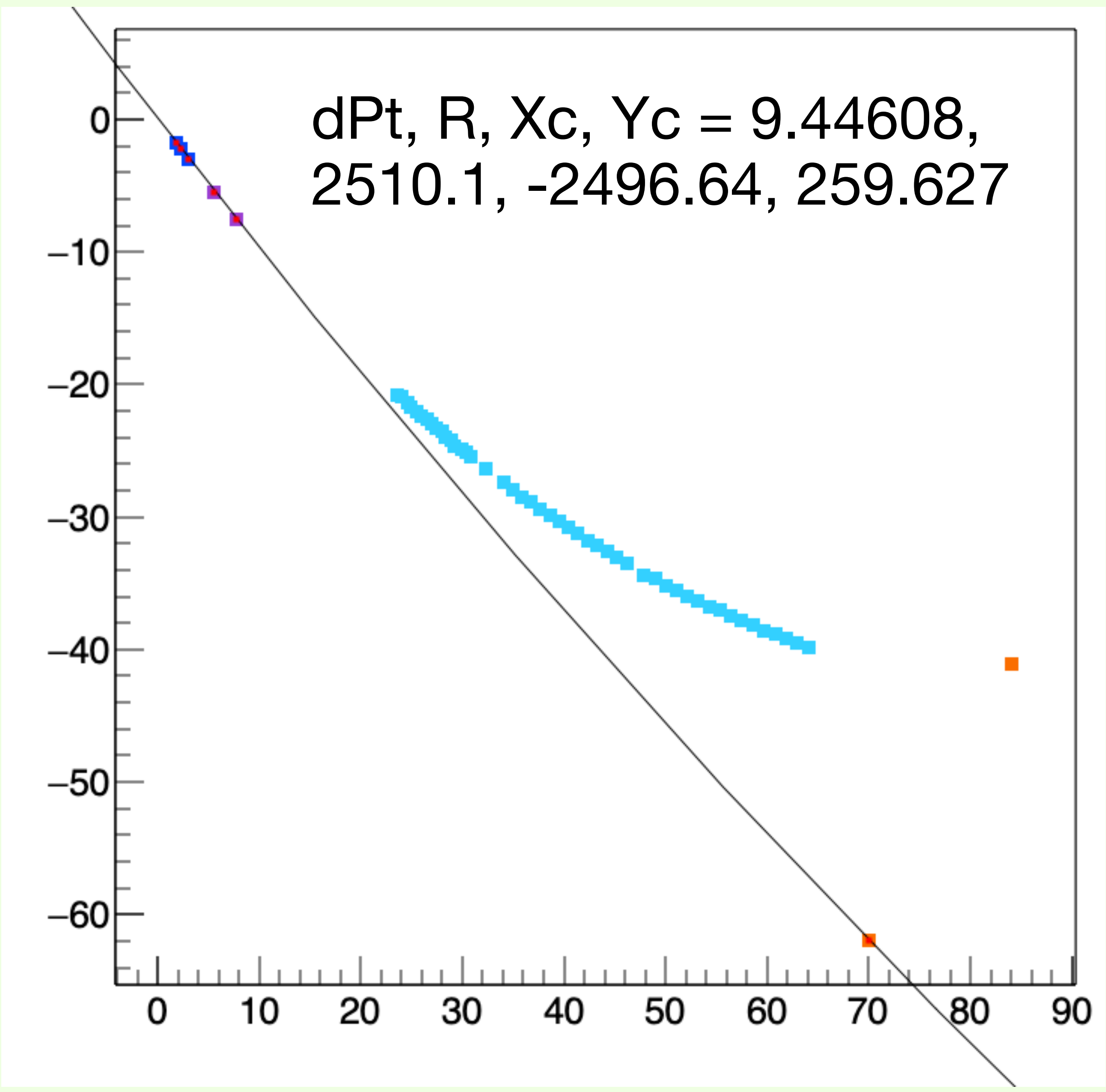
(2) Fail Track Event Display Examples (~20%)

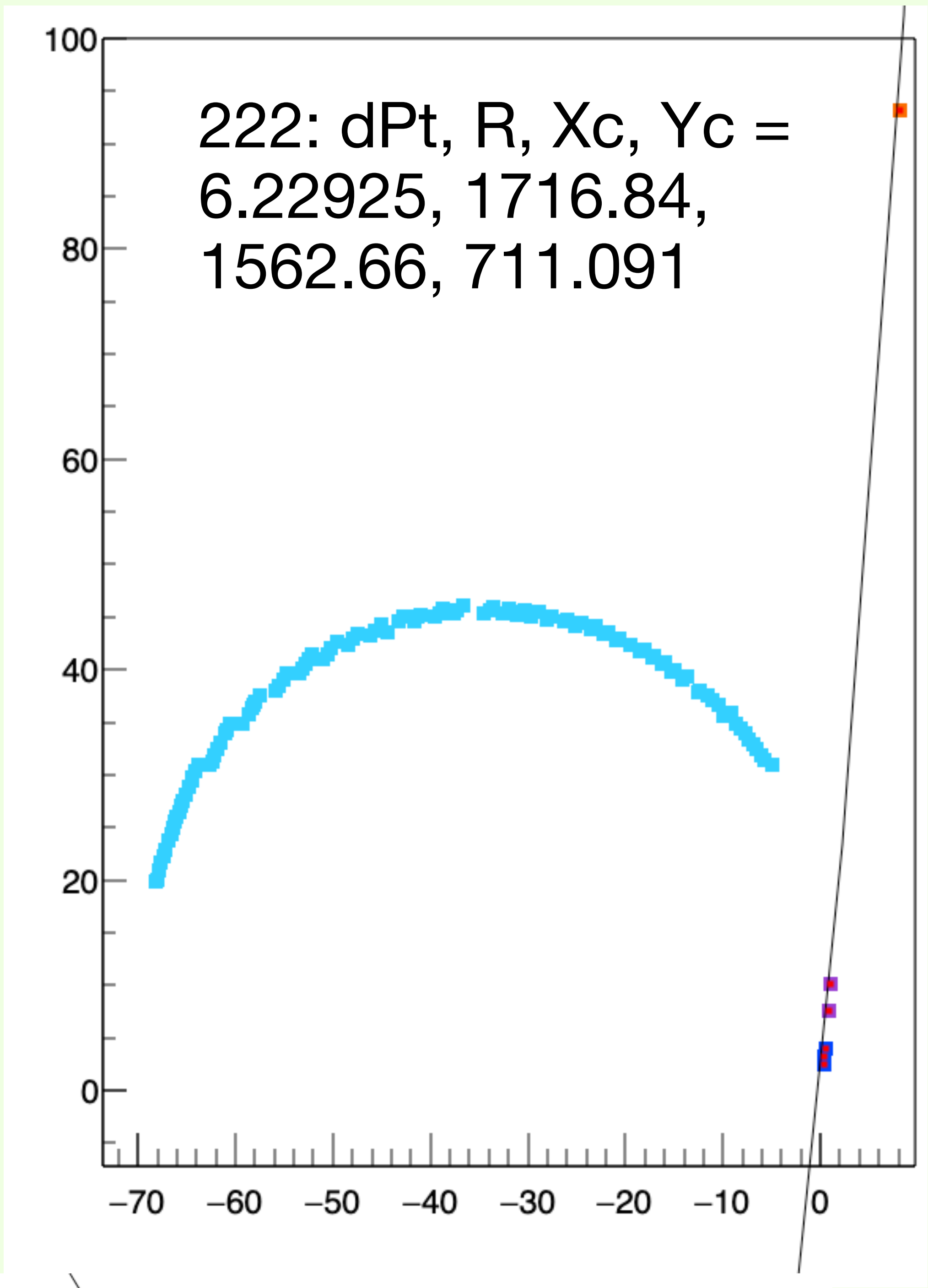


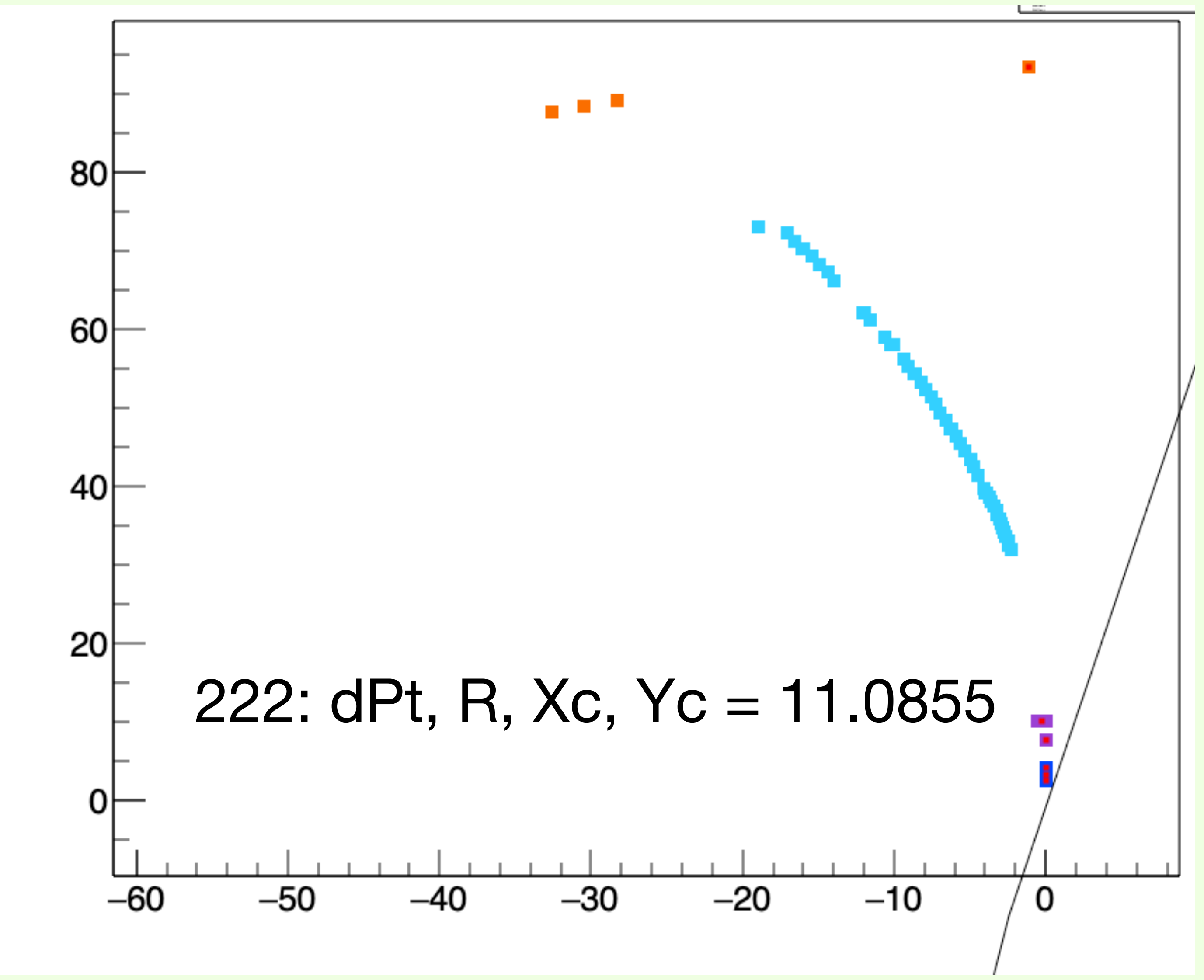
(3) Large dpT Events Examples (~8%)

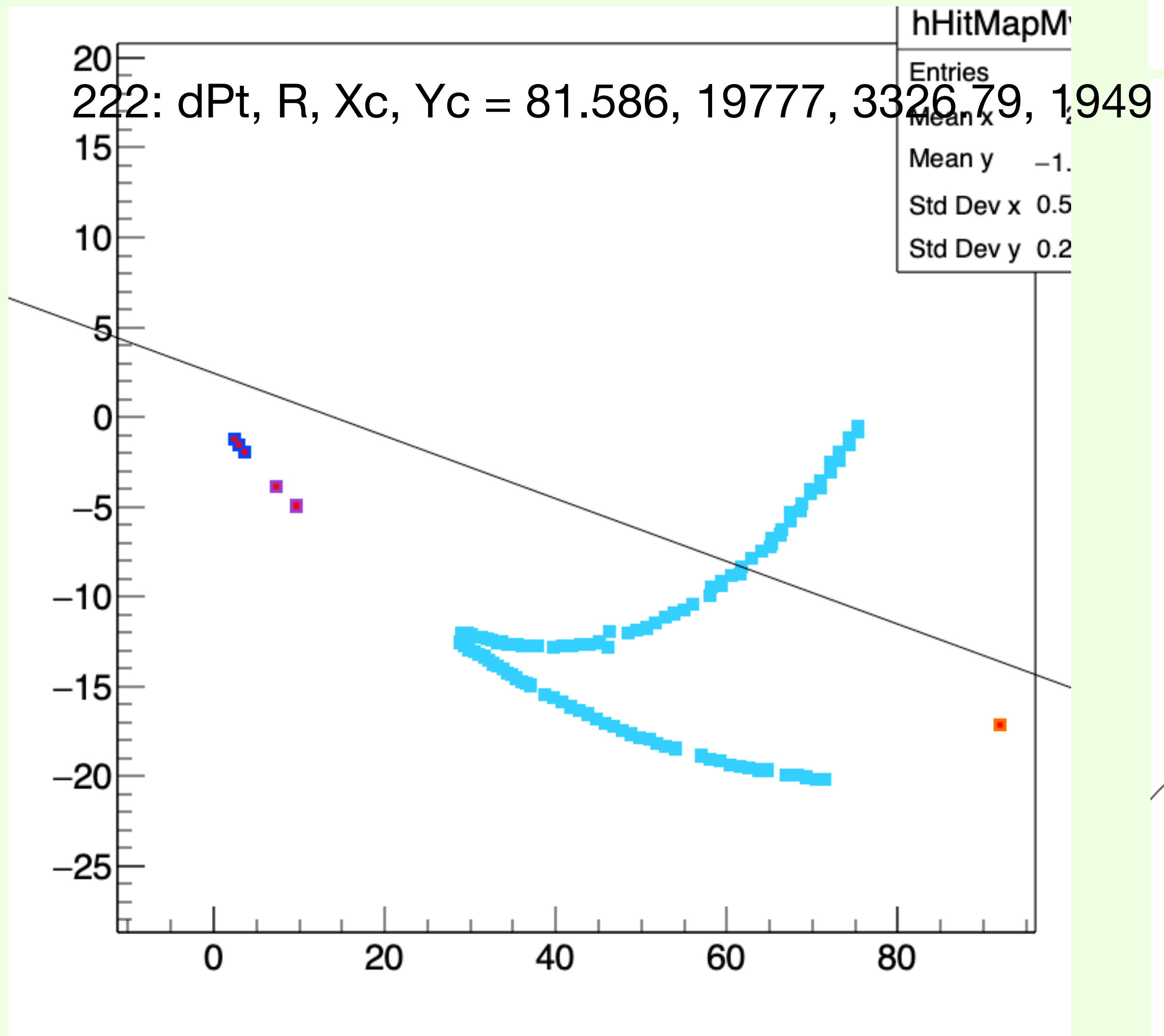


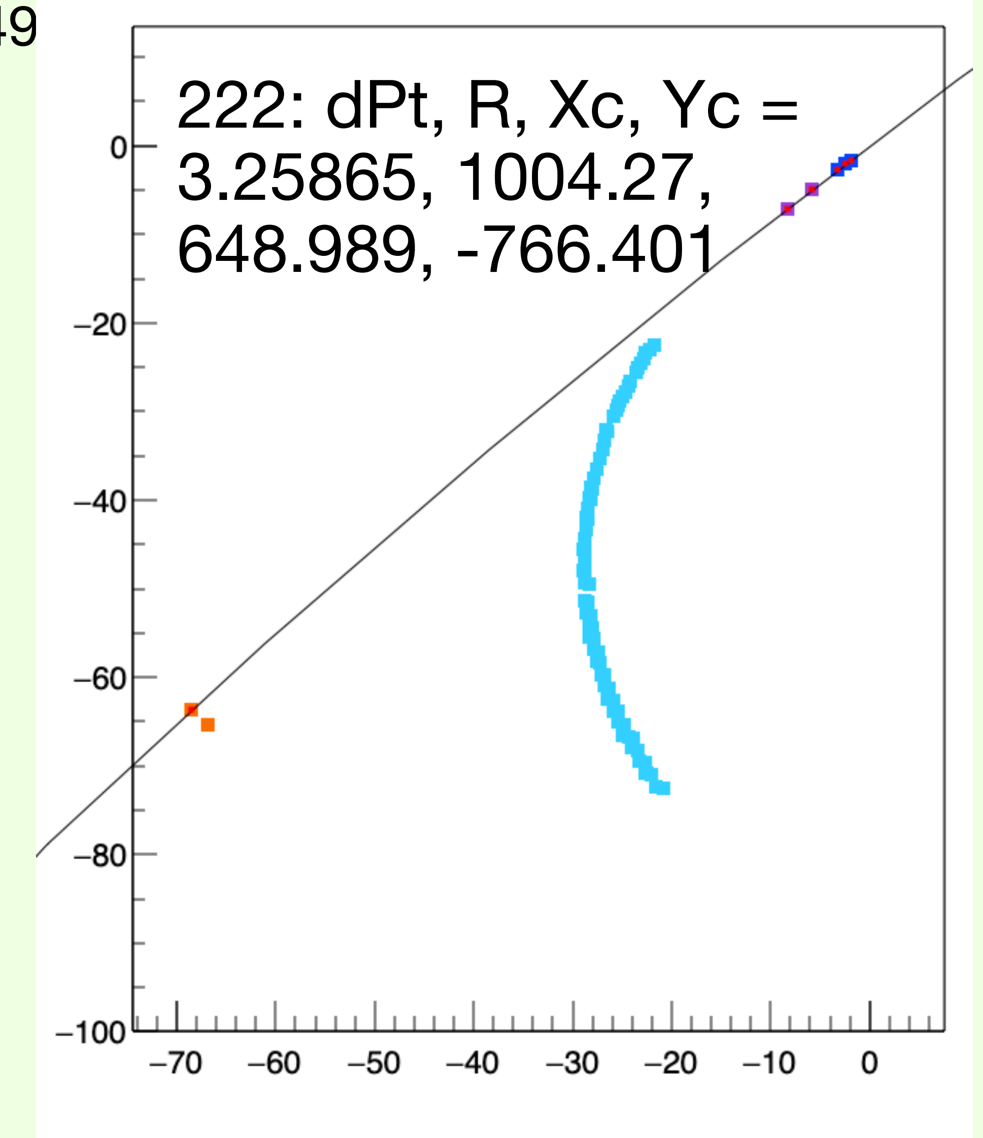


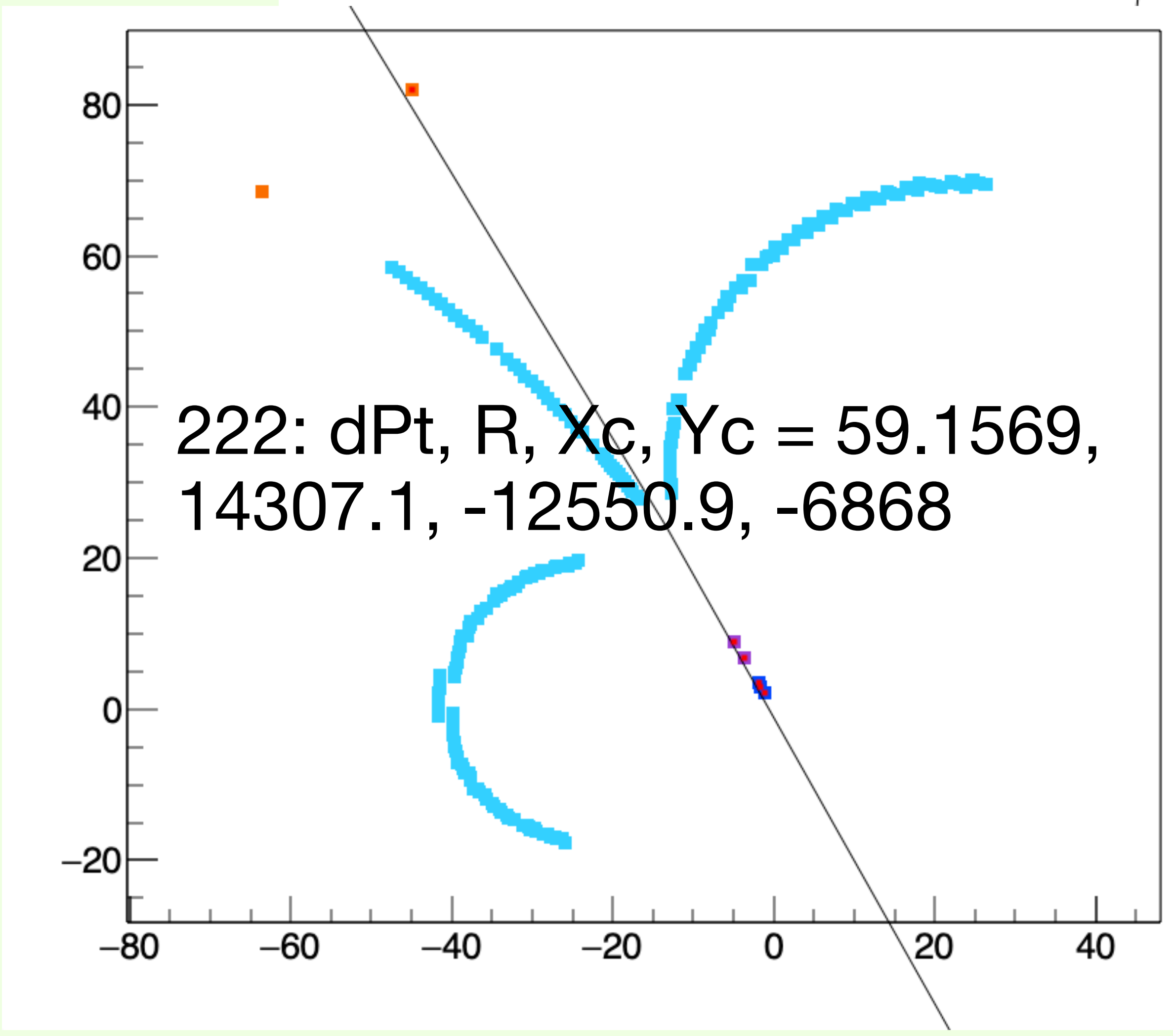




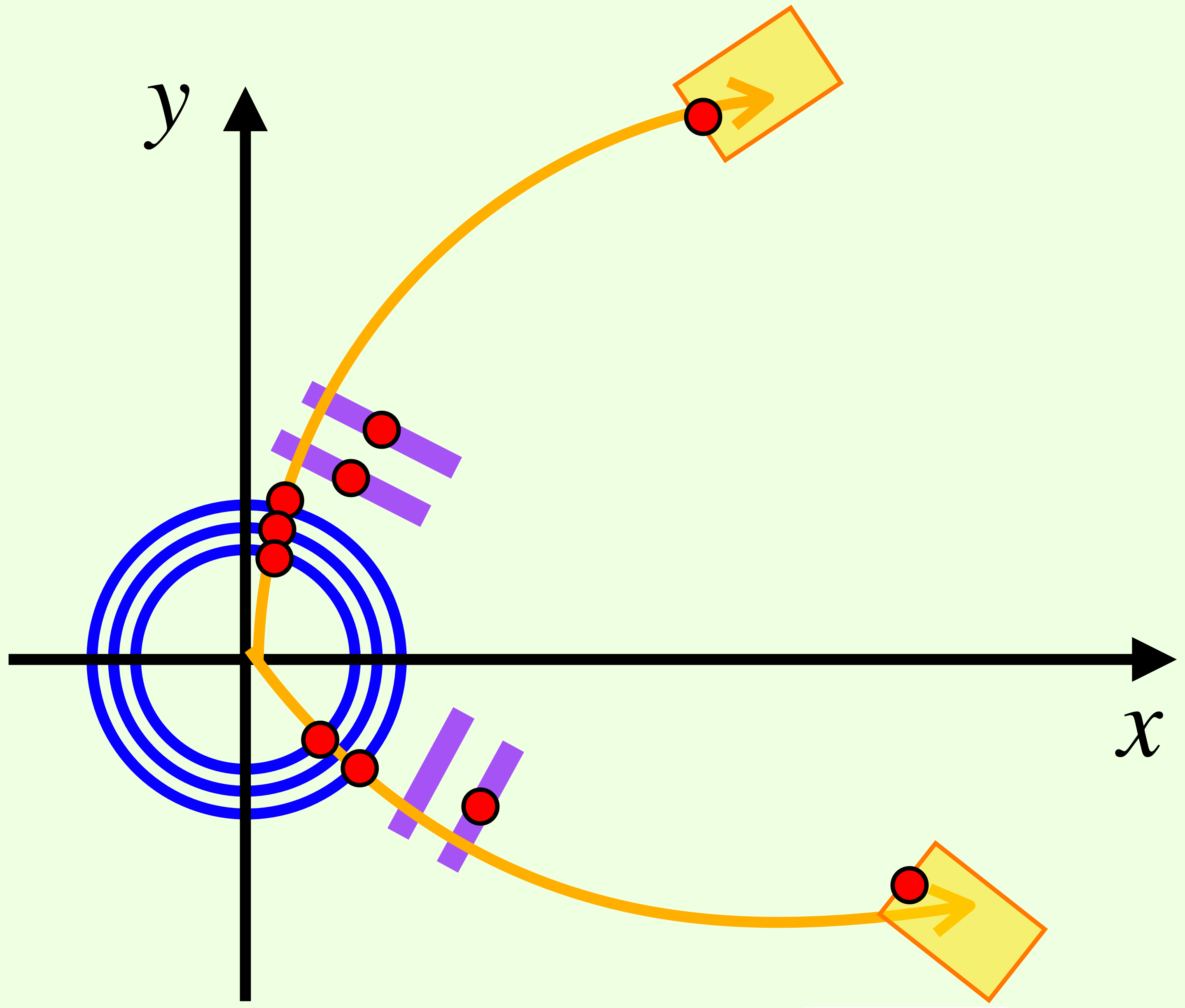








New Tracking Algorithm



The reason to make new hit matching algorithm to pick up single INTT events.
Such events are ~7%.

InttSeedTracking.cxx

Previous

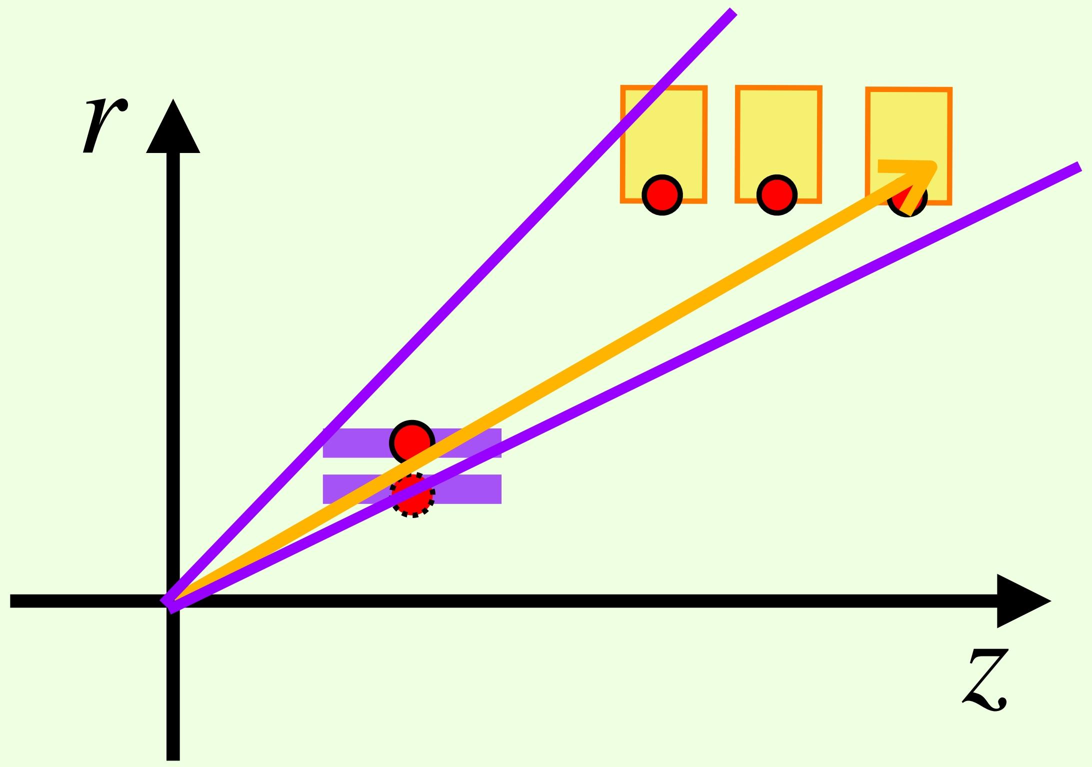
```
    // reco way1
    // HitMatching(tracks, vFMvtxHits, vSMvtxHits, vTMvtxHits, vIInttHits, vOInttHits,\
    //    vEmcalHits, vIHCalHits, vOHCalHits);
    // for(Int_t iTrk = 0; iTrk < tracks.size(); iTrk++){
    //    TrackPropertiesEstimation(tracks.at(iTrk), vFMvtxHits, vSMvtxHits, vTMvtxHits);
    // }

    // reco way2
    RecoTracksInttSeed2(tracks, vFMvtxHits, vSMvtxHits, vTMvtxHits,\
      vIInttHits, vOInttHits, vEmcalHits, vIHCalHits, vOHCalHits);
    for(Int_t iTrk = 0; iTrk < tracks.size(); iTrk++){
      TrackPropertiesEstimation2(tracks.at(iTrk));
    }
```

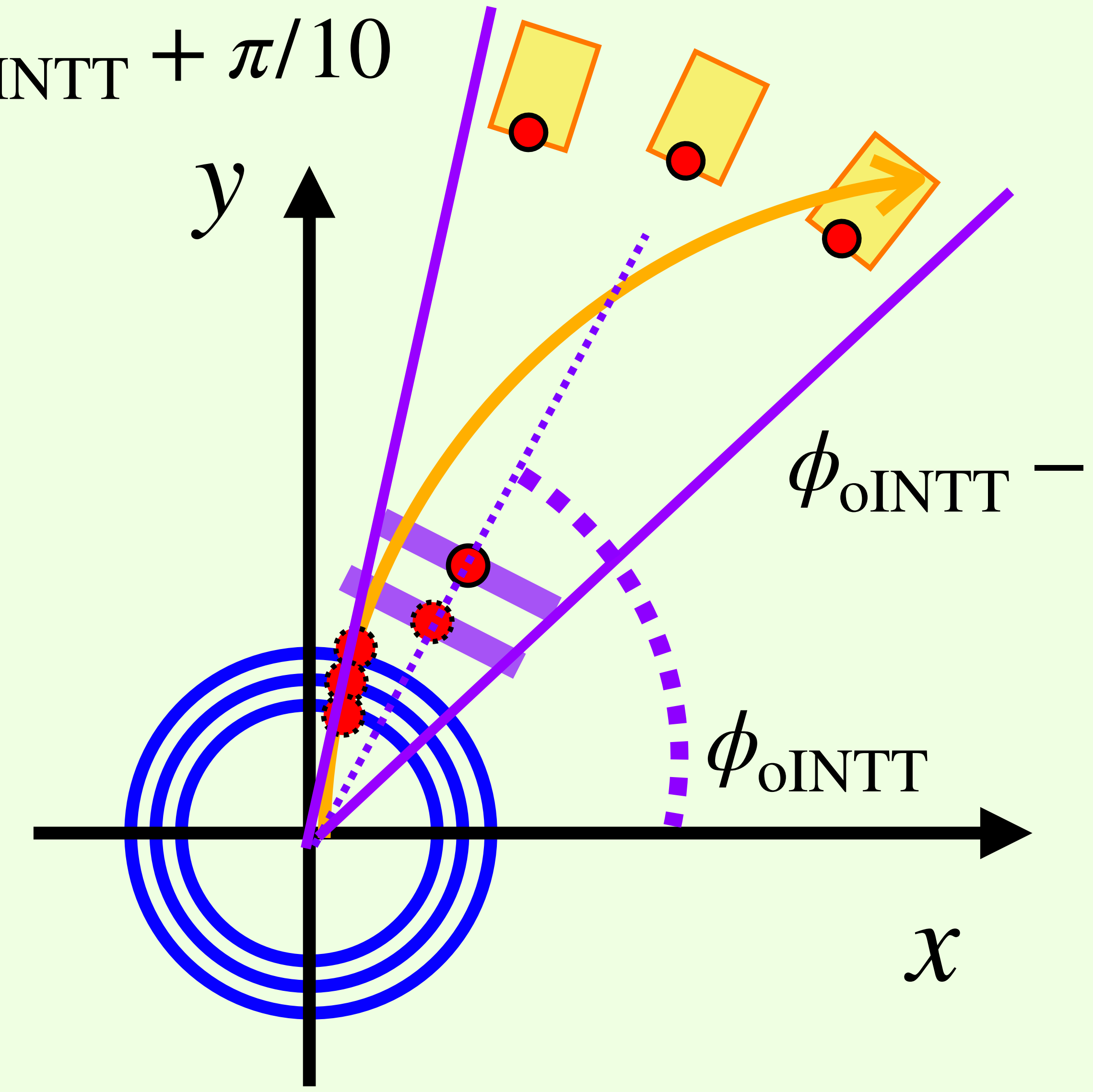
New

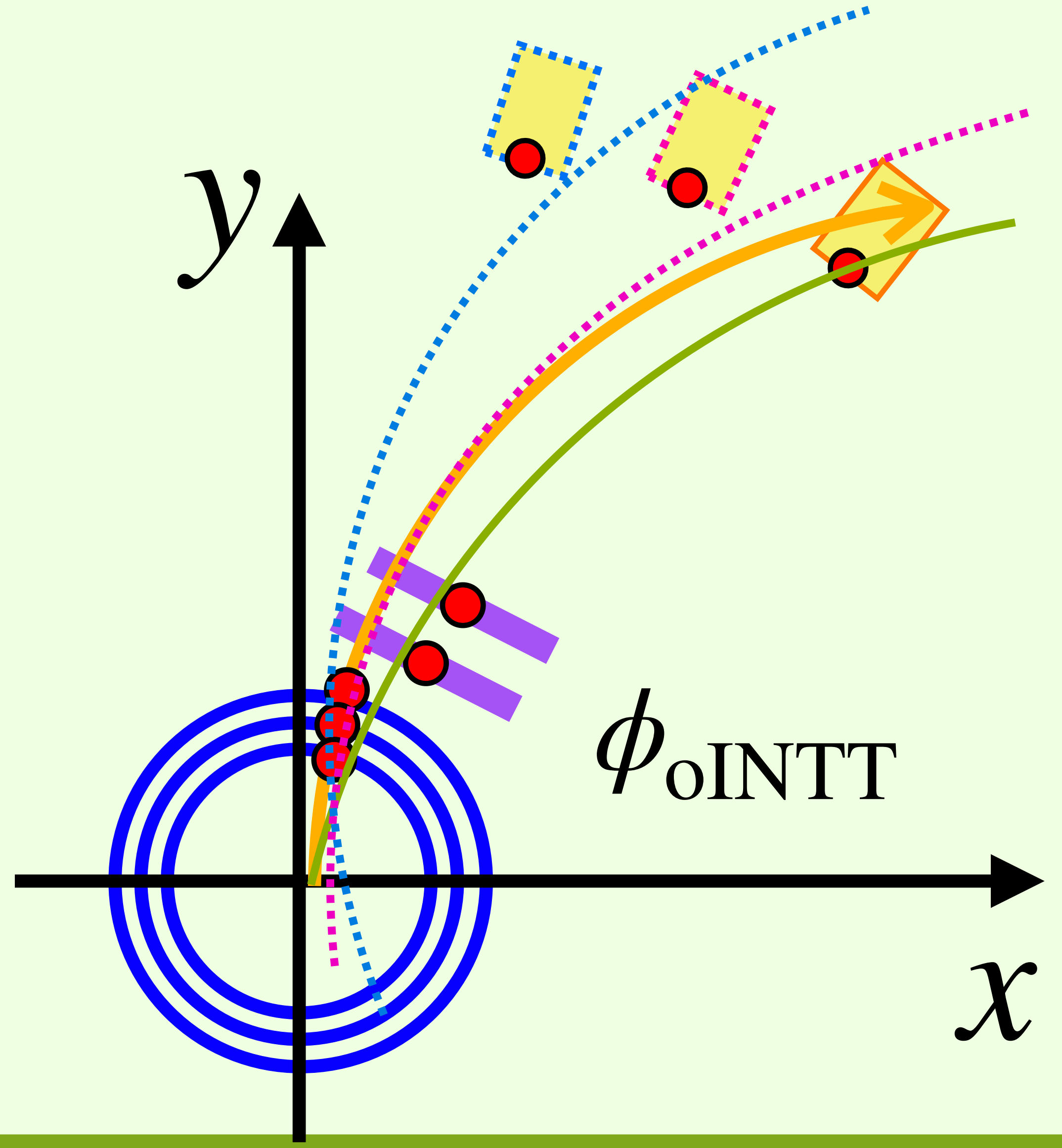
1.1 Searching for EMCal hits in the range.

1. $\eta_{\mathrm{oINTT}} - \pi/10 < \eta_{\mathrm{EMCal}} < \eta_{\mathrm{oINTT}} + \pi/10$
2. $\phi_{\mathrm{oINTT}} - \pi/10 < \phi_{\mathrm{EMCal}} < \phi_{\mathrm{oINTT}} + \pi/10$
3. $E_{\mathrm{EMCal}} > 0.1\ \mathrm{GeV}$

I still not optimize the each value







2.1 Draw a rough circle using 3 points vertex (0, 0).

2.2 Select closest hits for each detector.

2.3 Re-fit by a circle using all detector hits.

2.4 Estimate χ^2 for the circle.

2.5 Select the best track with minimum χ^2.

Track requirement:

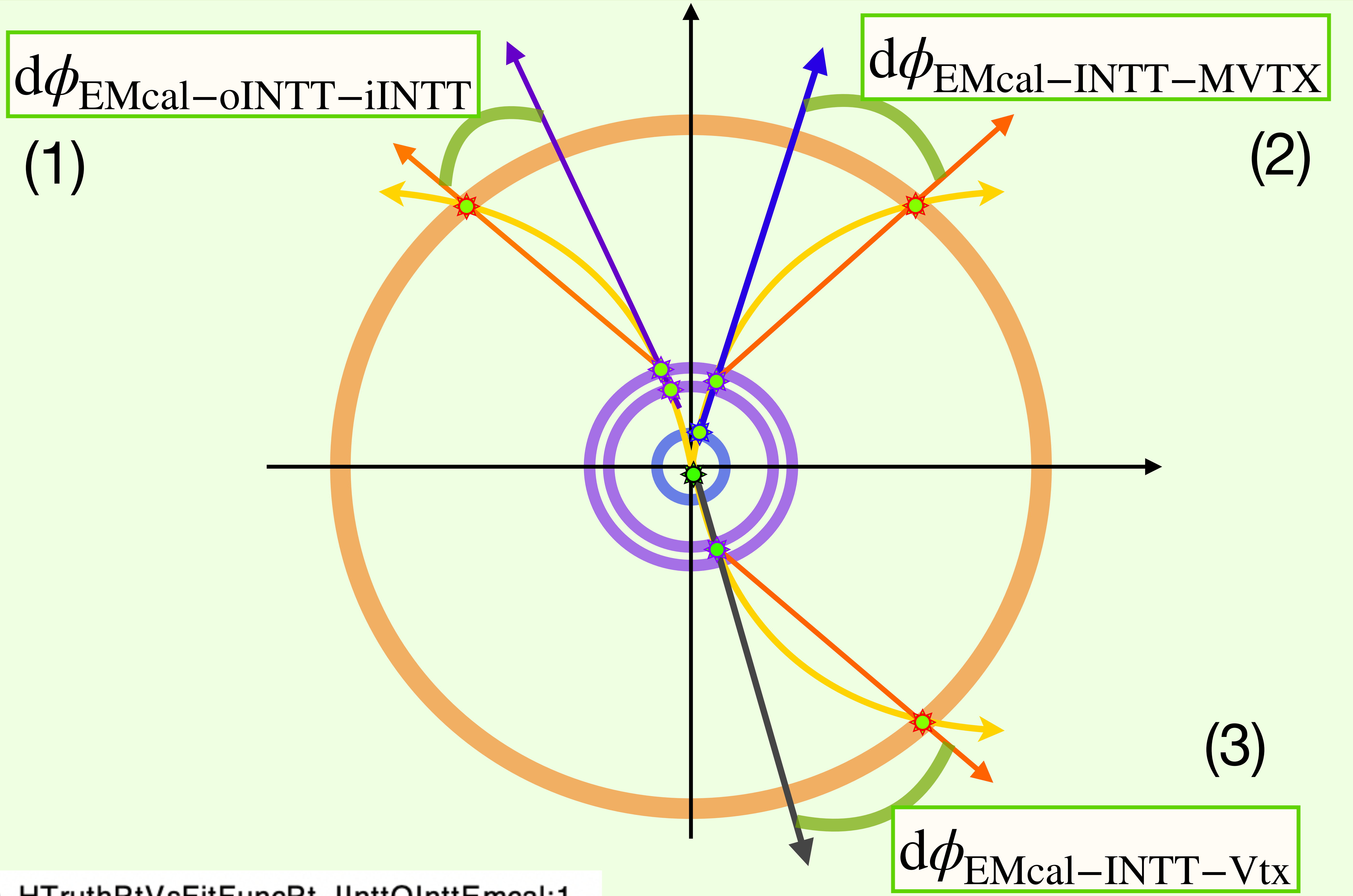
(1) both INTT (iINTT+oINTT)

(2) Single INTT + 2 MVTX

Kinds of dPhi vs dPt functions

The new algorithm can track single INTT event.
On the other hand, the previous $\mathrm{d}\phi - \mathrm{d}p_{\mathrm{T}}$ function way (1) requires two INTT hits.

Then, I prepared additional two 3 points ways [(2)-(3)].



(1) m_HDPhiVsTruthPtVsEta_IInttOInttEmcal;1

(3) m_HDPhiVsTruthPtVsEta_VtxIInttEmcal;1
m_HDPhiVsTruthPtVsEta_VtxOInttEmcal;1
m_HDPhiVsTruthPtVsEta_VtxInttEmcal;1

(2) m_HDPhiVsTruthPtVsEta_MvtxIInttEmcal;1
m_HDPhiVsTruthPtVsEta_MvtxOInttEmcal;1
m_HDPhiVsTruthPtVsEta_MvtxInttEmcal;1

→

m_HTruthPtVsFitFuncPt_IInttOInttEmcal;1
m_HTruthPtVsFitFuncPt_VtxIInttEmcal;1
m_HTruthPtVsFitFuncPt_VtxOInttEmcal;1
m_HTruthPtVsFitFuncPt_VtxInttEmcal;1
m_HTruthPtVsFitFuncPt_MVtxIInttEmcal;1
m_HTruthPtVsFitFuncPt_MVtxOInttEmcal;1
m_HTruthPtVsFitFuncPt_MVtxInttEmcal;1

Estimate Performance in the Mixed Events for MVTX

The time resolution of MVTX is not enough to distinguish each event. (left figure)
Therefore, within the sPHENIX concept, the INTT is used to select MVTX hits from certain events.

We need to check that the tracking performances are kept in the such mixed up situations.

If you set stockEvents to a high number (e.g., 5), you can test such situations.
The three event displays show that situation, but you need to test it or PYTHIA.