Detection efficiency of the hit cluster in the sPHENIX-INTT detector

Progress Report

Ryota Shishikura

Contents

1. Single mu MC

2. pp collision MC (varied vertex)

    1. Using truth vertex

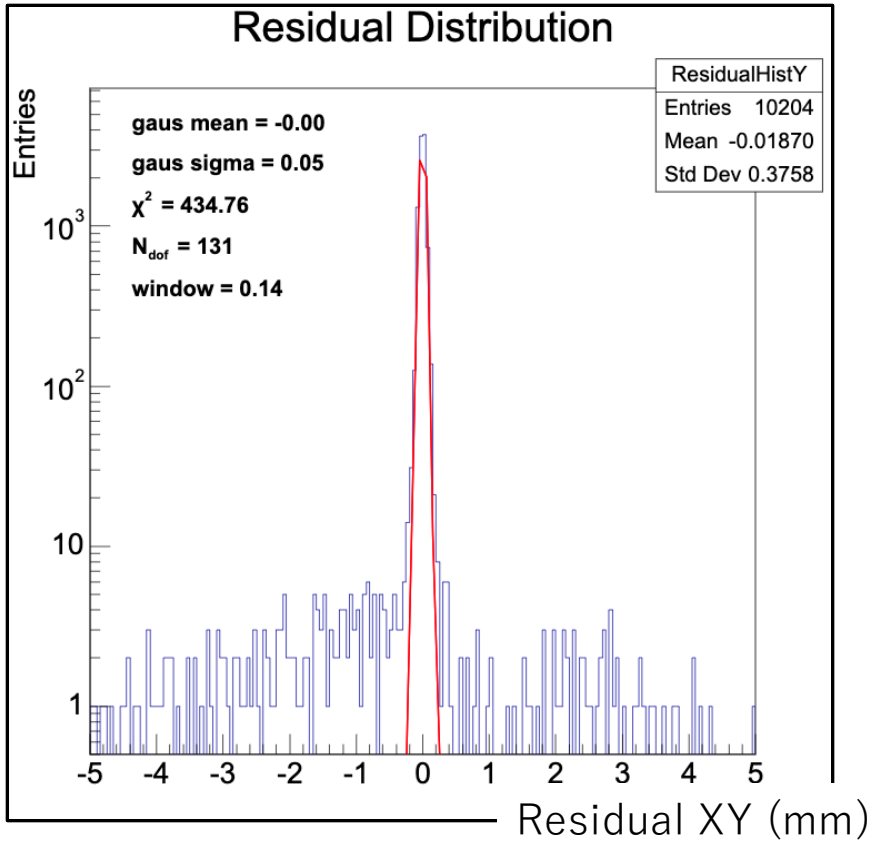
    2. Using Ikemoto reconstruction vertex

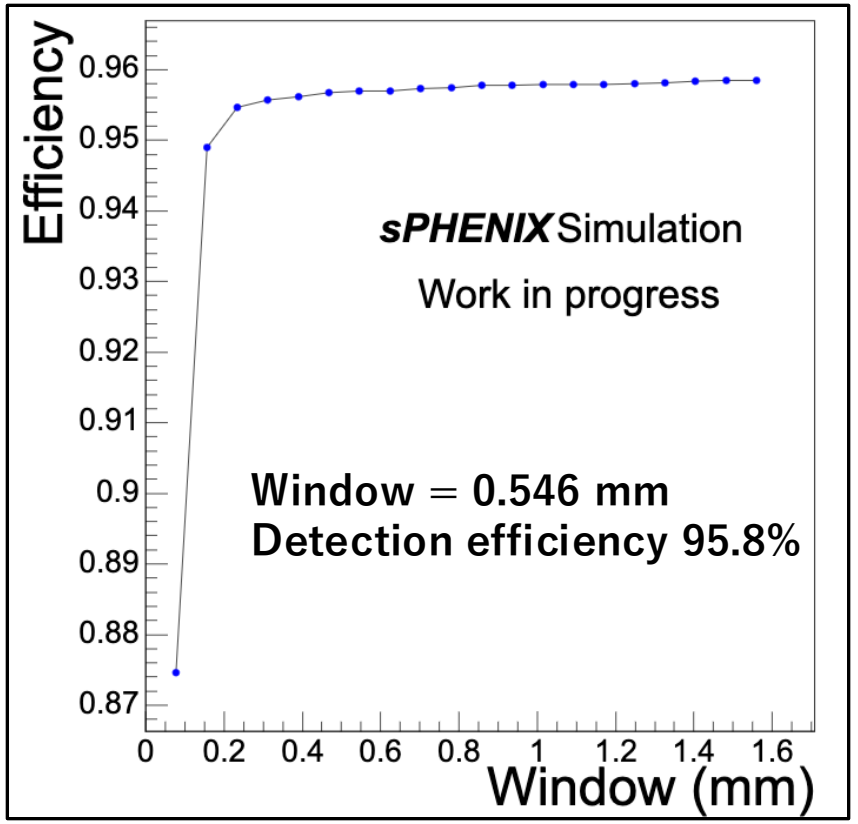
Single mu MC

- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel

Single mu MC

- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel





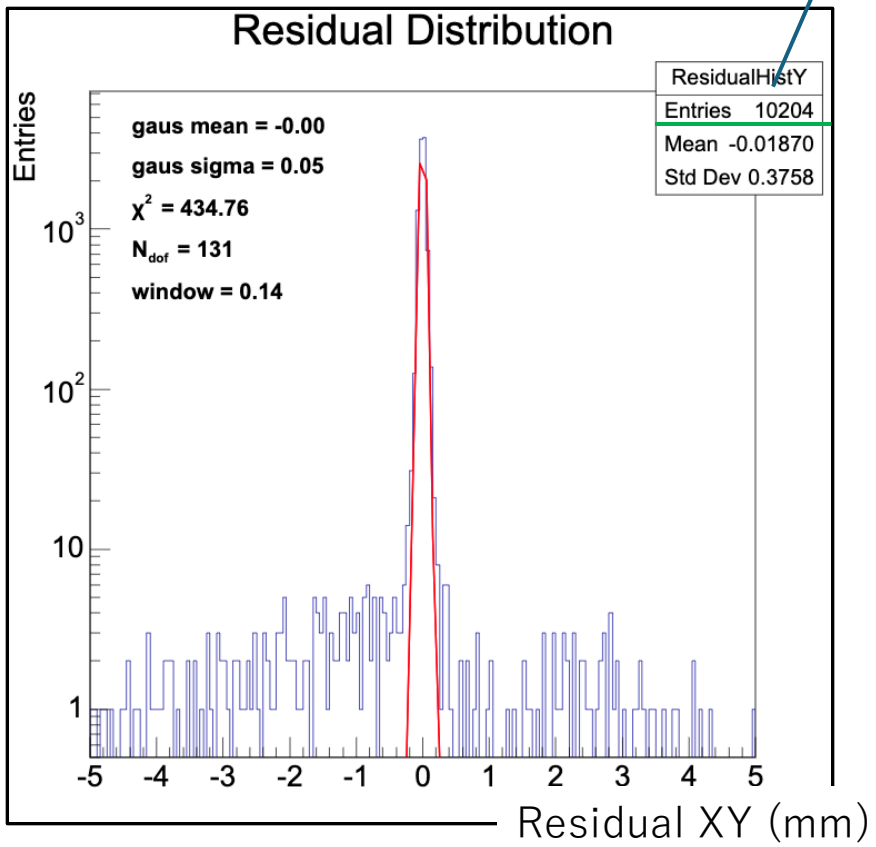
Y axis: Entries
X axis: Residual in X-Y plane

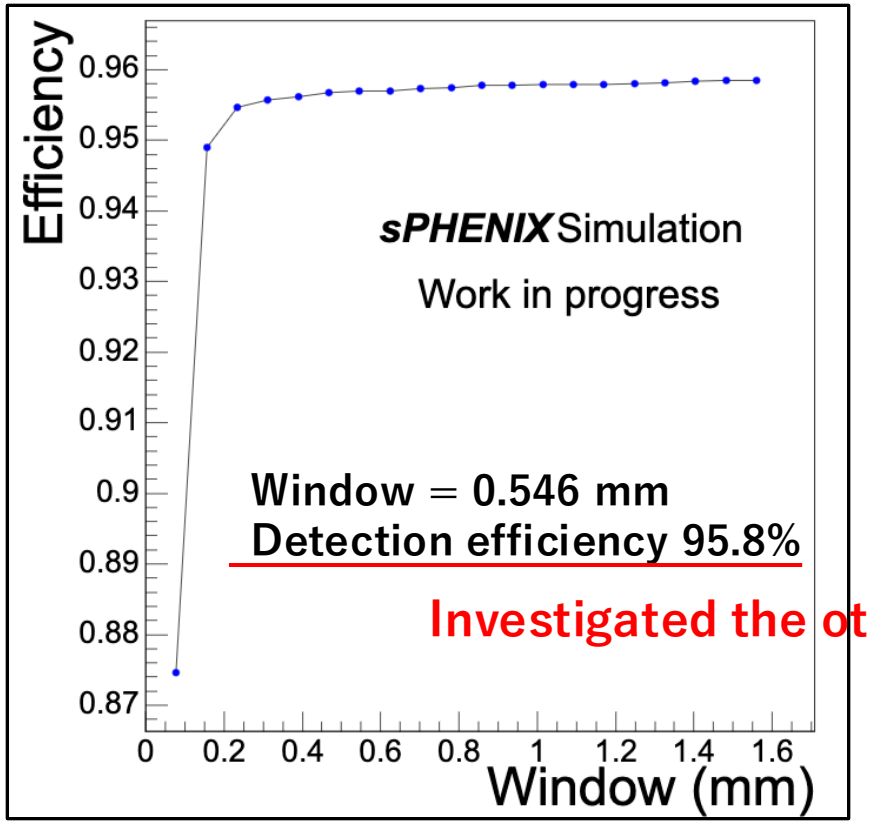
Y axis: Efficiency
X axis: Window = $78\mu m \cdot i$ $(i = 1, 2, 3, \ldots)$

Single mu MC

The generated the other particles are the cause?

- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel





Investigated the other 4%

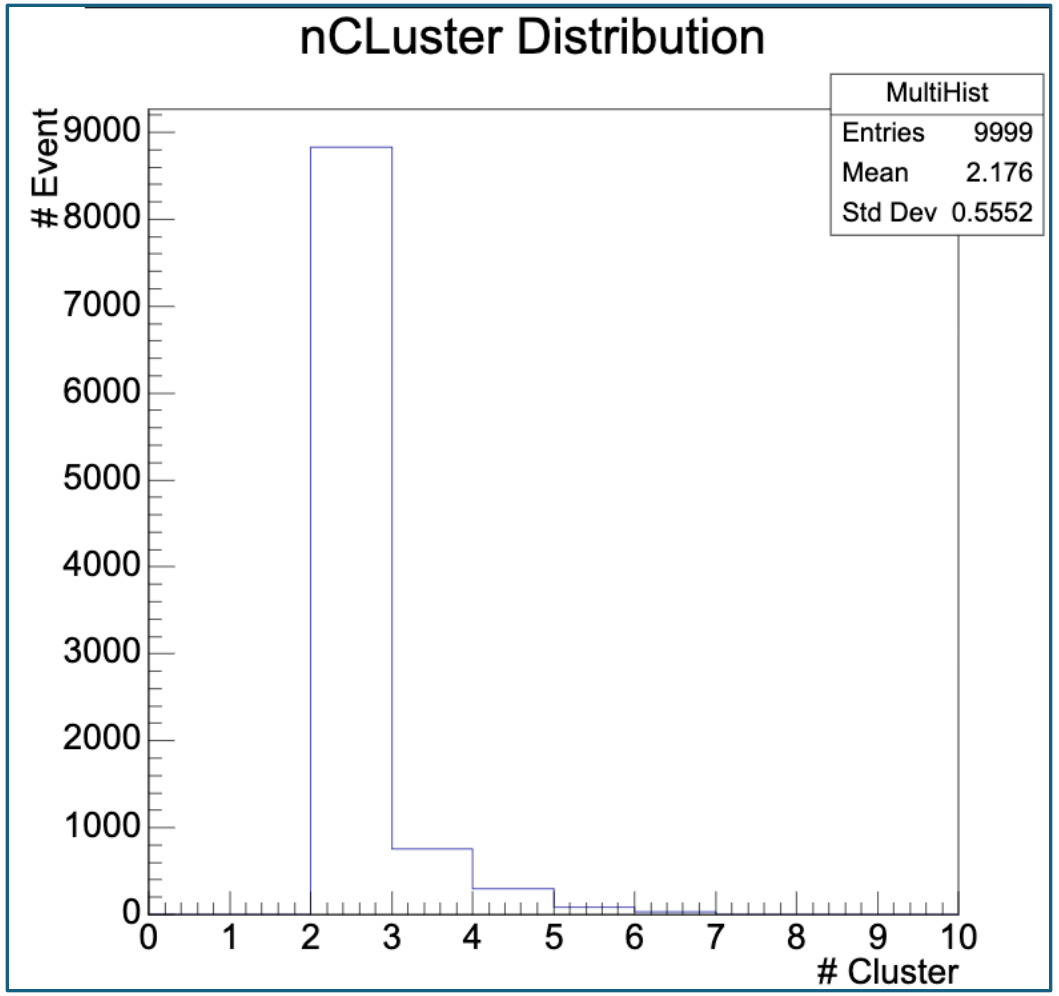
Y axis: Entries
X axis: Residual in X-Y plane

Y axis: Efficiency
X axis: Window = $78\mu m \cdot i$ $(i = 1, 2, 3, \ldots)$

Single mu MC

- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel

The cause of the inefficiency is the other particle generated ?
→ Let's see the Multiplicity distribution.



- This simulation is a particle (μ^-)/event. So, the base num of clusters is 2 .
- But, the num of the event which has more than 3 clusters is more than 1000.
  → **There are the events generate the other particle.**
- The possibility of influence of the inefficiency by them is high.

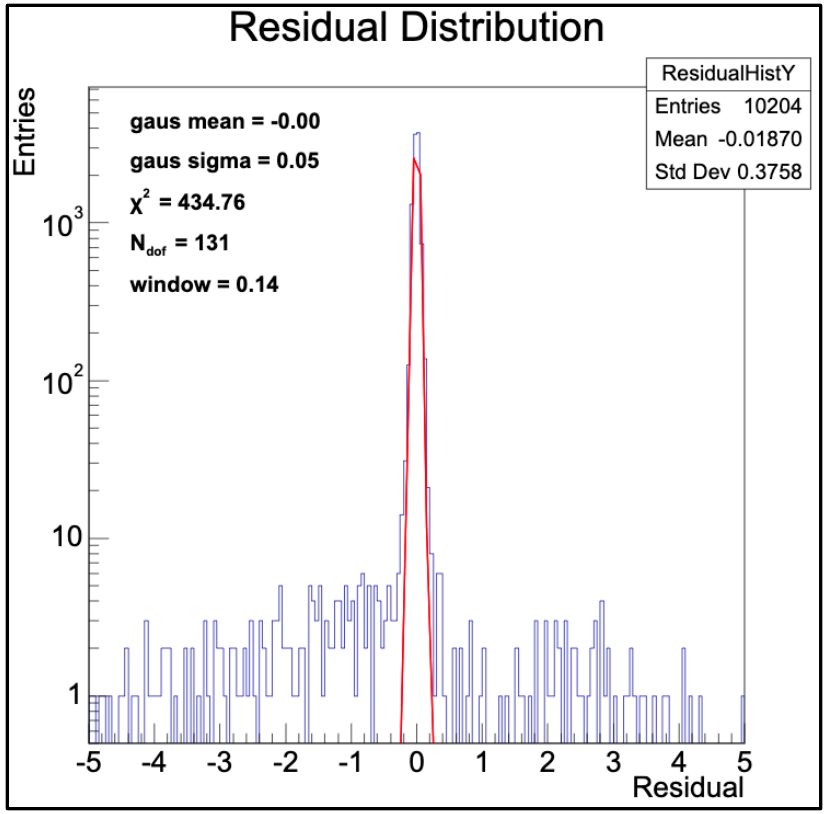
→I did the event cut (**# of outer clusters = 1**)

Single mu MC

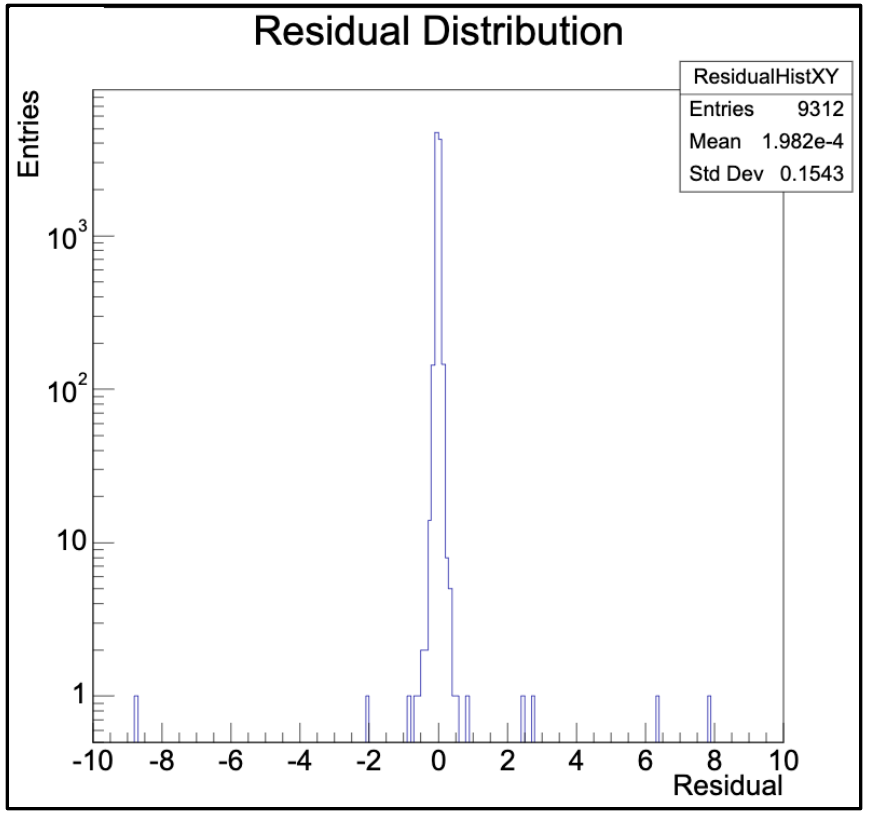
- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel

Add the event cut "# of outer clusters = 1"

Before



After

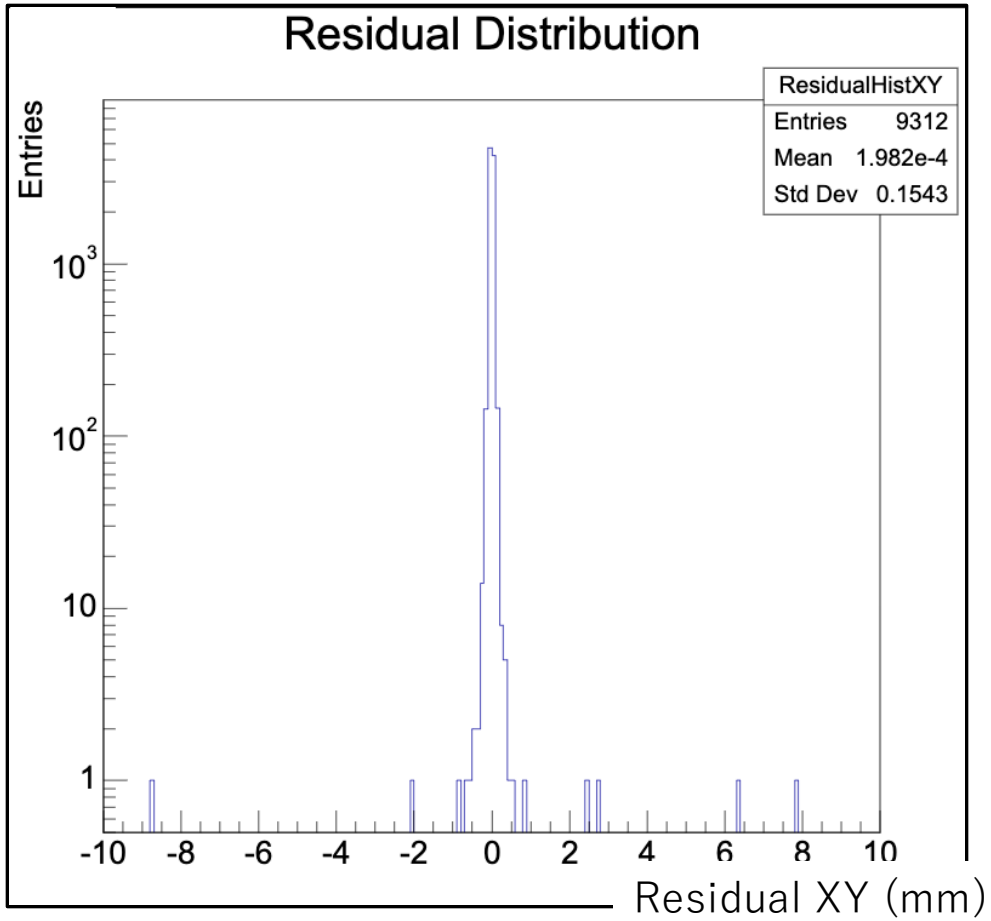


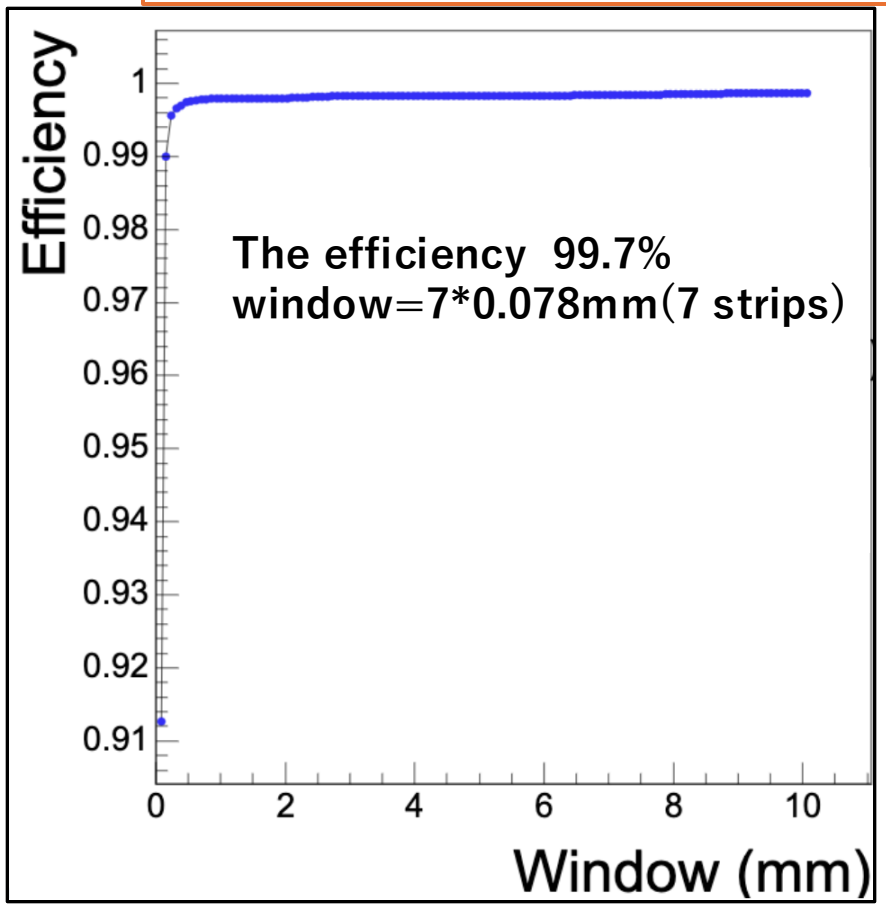
The cut works for the part of the large residual

Single mu MC

- A particle (μ^-)/event
- P_T = 1 GeV
- # of events : 10K
- Magnetic field : zero field
- Incident point: Fixed (x, y, z) = (0, 0, 1)cm
- Incident direction : ϕ =0 rad, η =0
- No dead channel

Add the event cut "# of outer clusters = 1"





The remaining 4% can mostly be explained by the particle generation.

p+p MC(Varied vertex)

- I analyzed the pp MC in which the vertex is varied.

  - PYTHIA8
  - # of events : 10K
  - Magnetic field : zero field
  - Vertex: mean (x, y, z) = (0, 0, 0)cm
    width $(\sigma_x, \sigma_y, \sigma_z)$=(0, 0, 20)cm
  - No dead channel

  - Using truth vertex
  - Using Ikemoto reconstruction vertex

Cut

- I did the following cut.

  - **Cutting high-density areas (only outer layer cluster)**
    → high-density cause mismatching the cluster pair
    → demand isolation

  - **ADC cut (30<ADC<255)**

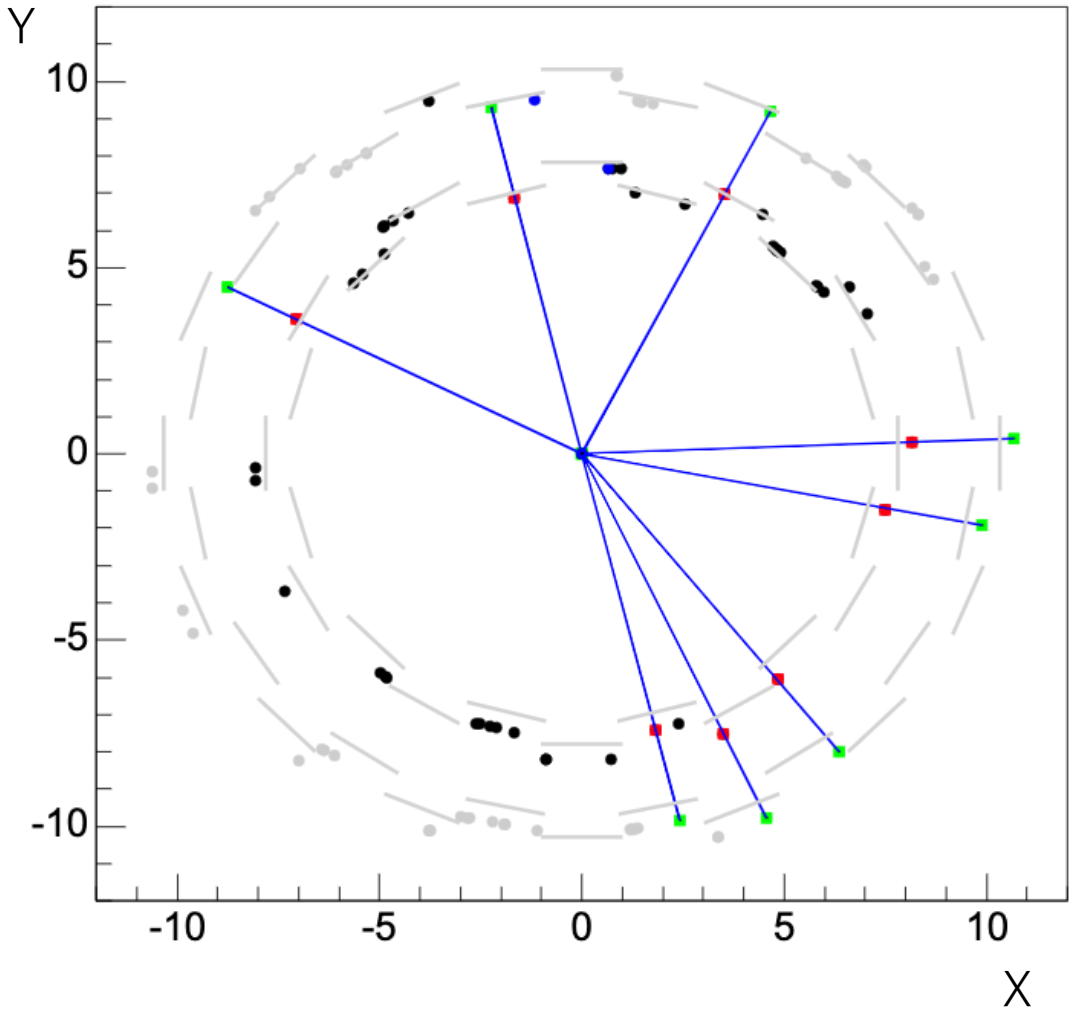
The pair matching

- Additionally, the following requirements are imposed when selecting cluster pairs to prevent bad cluster matching pair:

  - **Residual XY < 2 cm (within approximately 1 ladder)**

  - **Residual Z < 4 cm (within 2 chips)**

Event display X-Y (using truth vertex)



Window XY = 1 mm

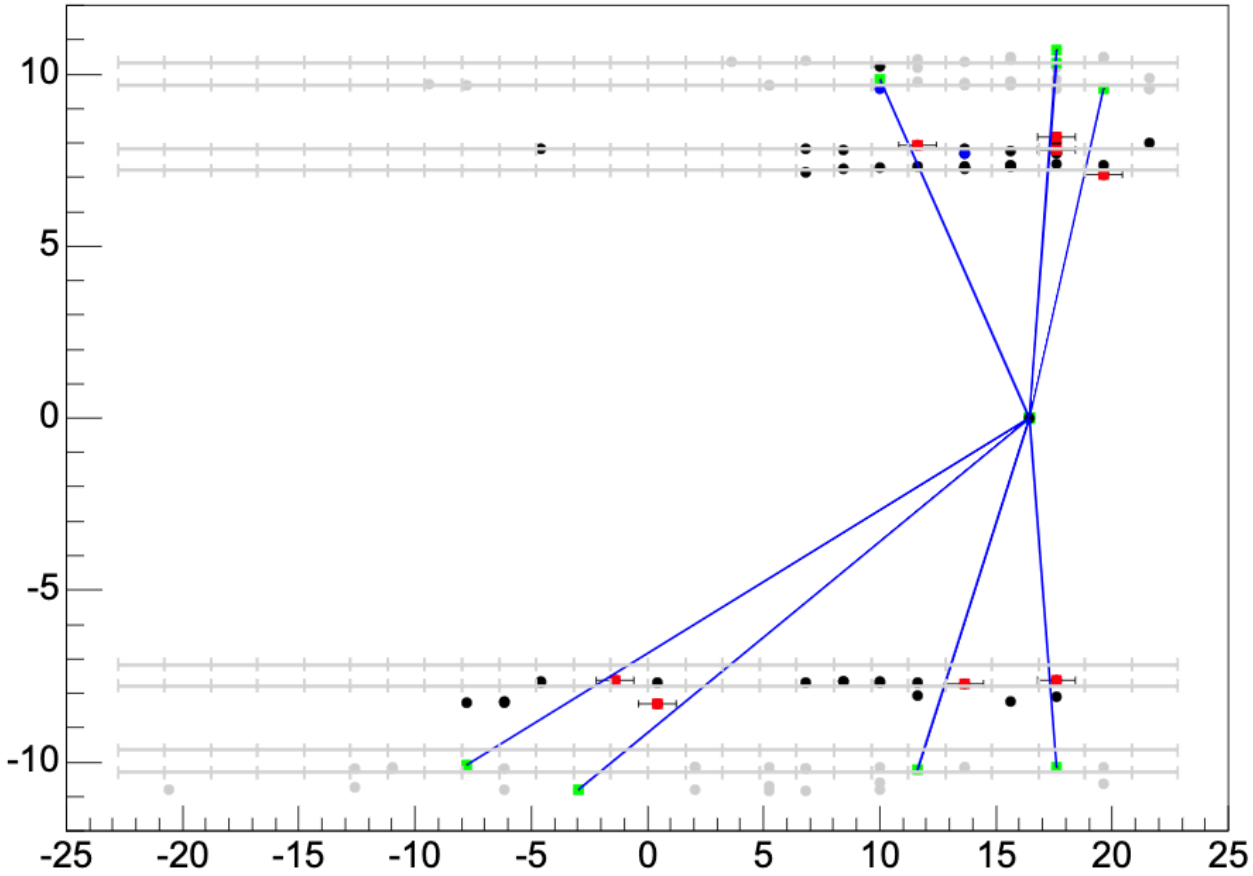
For Inner Clusters:

- **Black**: Not selected as a pair
- **Red**: Selected as a pair and within the window
- **Blue**: Selected as a pair but outside the window

For Outer Clusters:

- **Grey**: the cluster removed by high-dencity cut
- **Blue**: Pair found but not within the window
- **Black**: No pair found
- **Green**: Pair found and within the window

Event display R-Z (using truth vertex)



Window XY = 1 mm

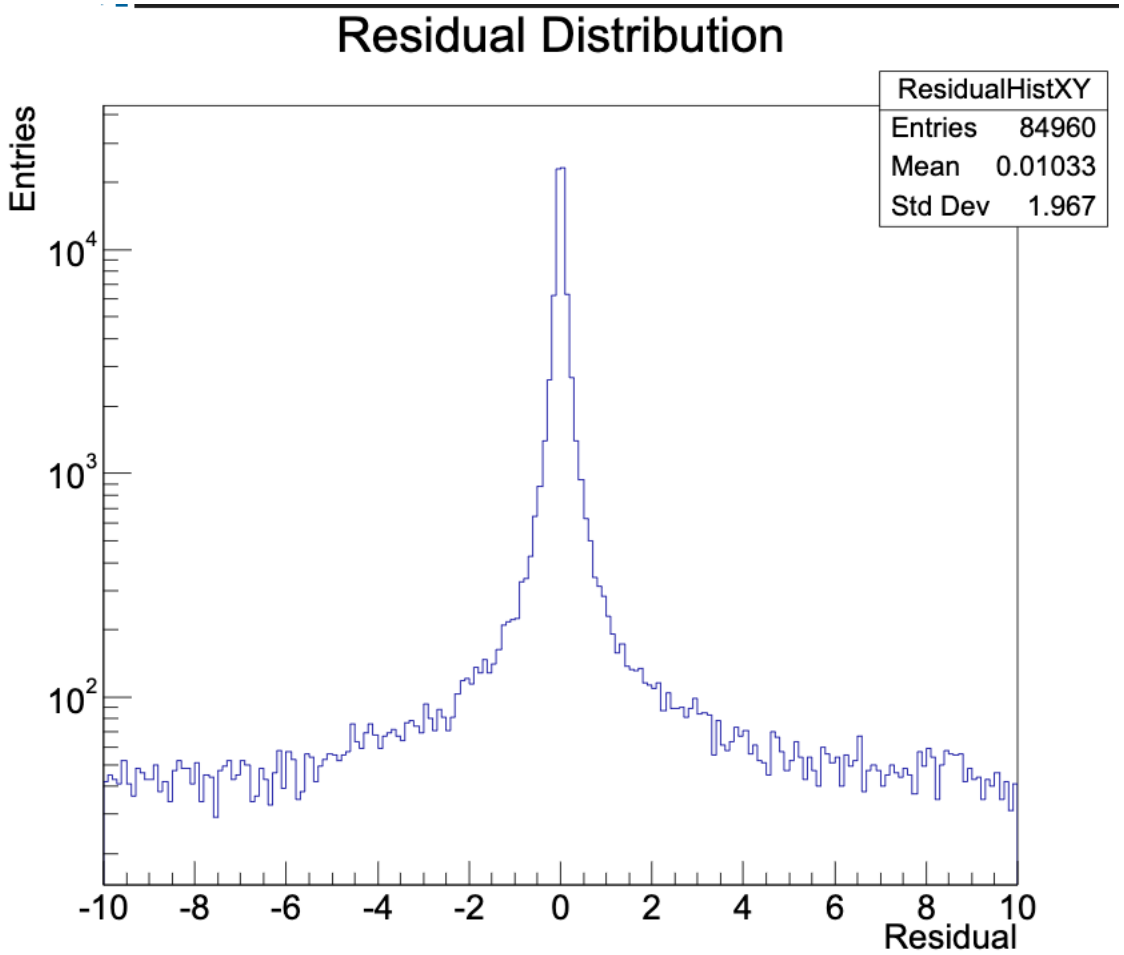
For Inner Clusters:

- **Black**: Not selected as a pair
- **Red**: Selected as a pair and within the window
- **Blue**: Selected as a pair but outside the window

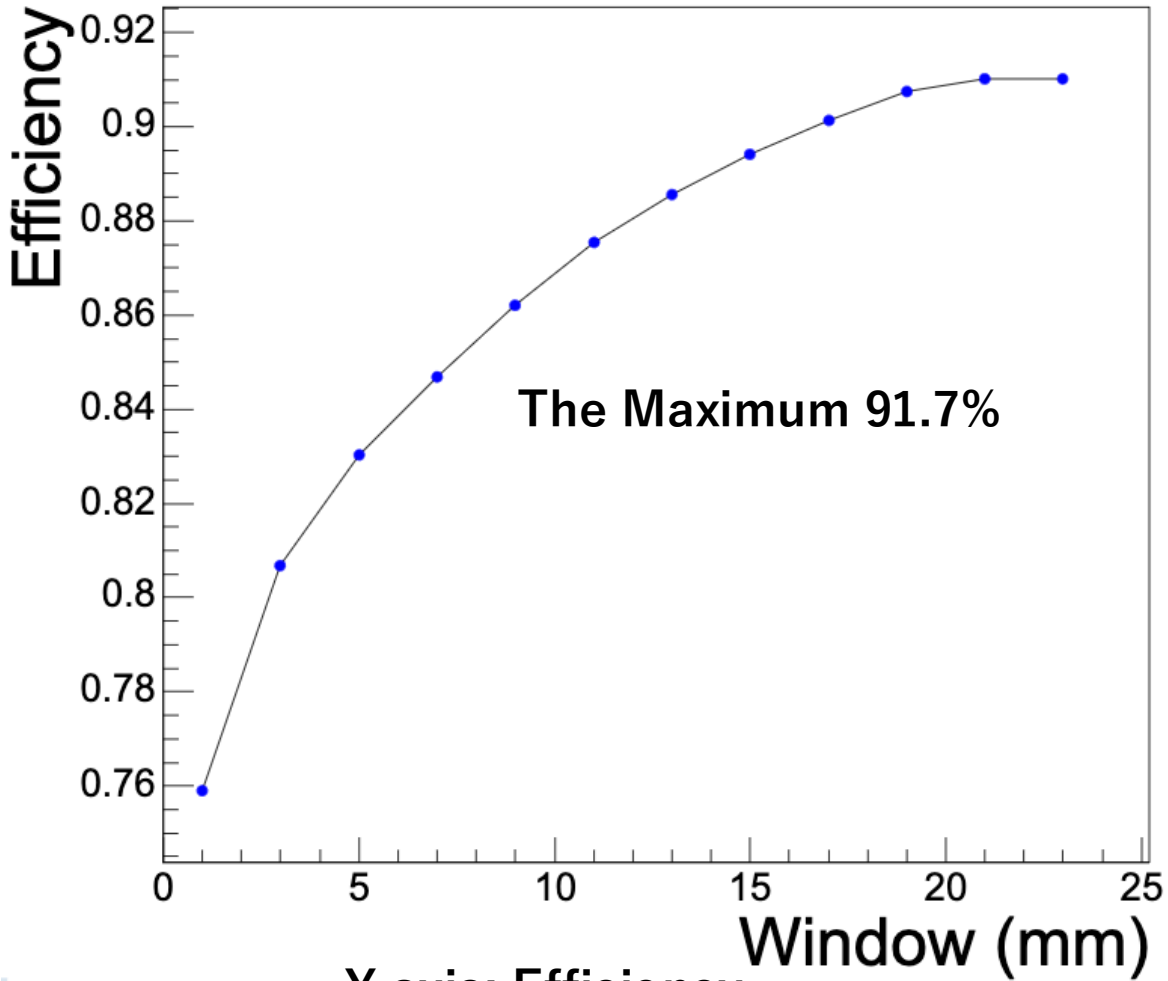
For Outer Clusters:

- **Grey**: the cluster removed by high-dencity cut
- **Blue**: Pair found but not within the window
- **Black**: No pair found
- **Green**: Pair found and within the window

Efficiency (using truth vertex)

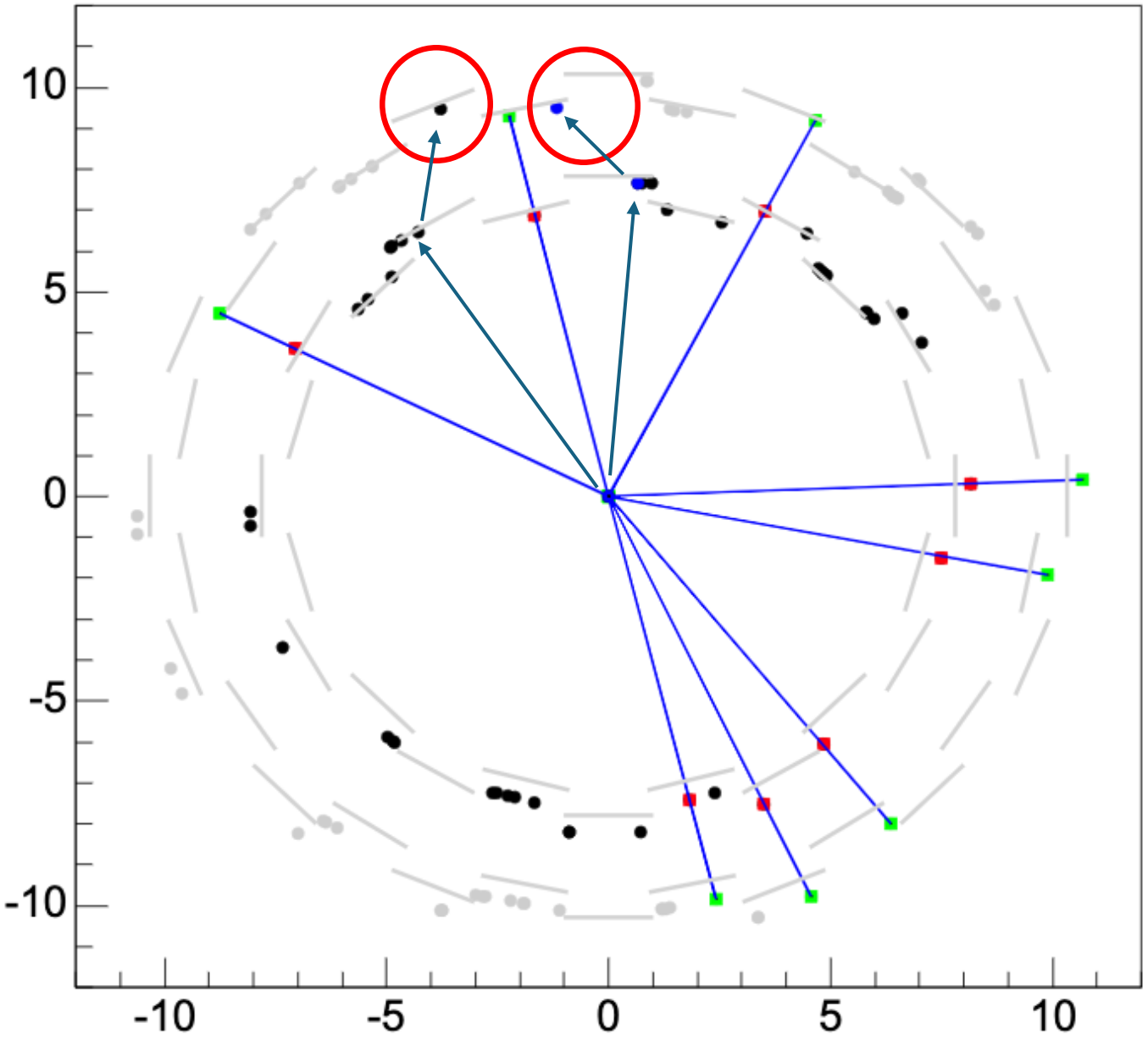


Y axis: Entries
X axis: Residual in X-Y plane



Y axis: Efficiency
X axis: Window (mm)

The cause of inefficiency



The generated soft particle couldn't find pair.

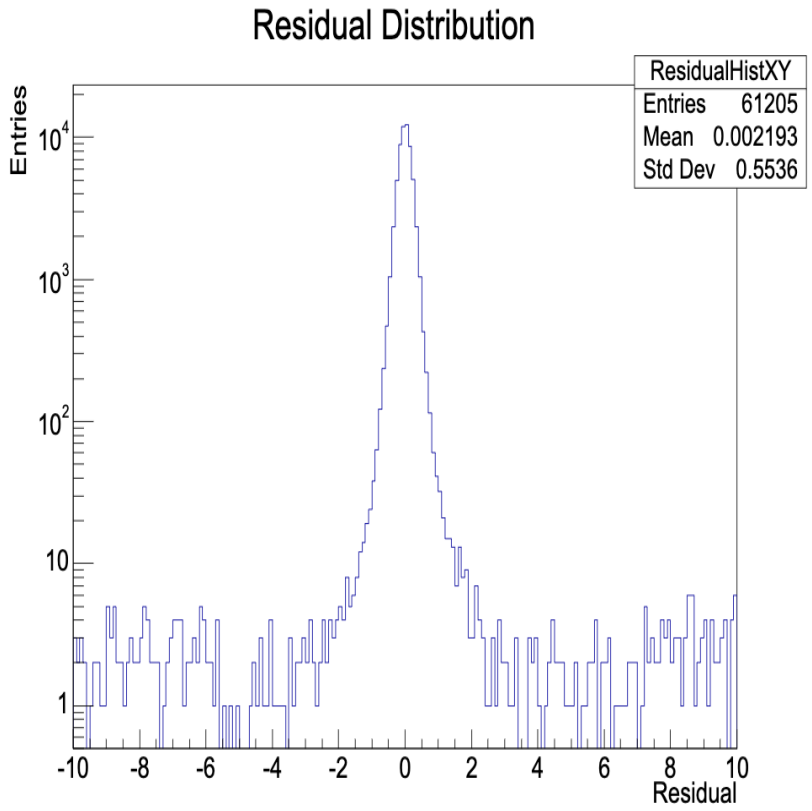
→ I analyzed mu MC in several P_T to investigate P_T dependence

mu MC

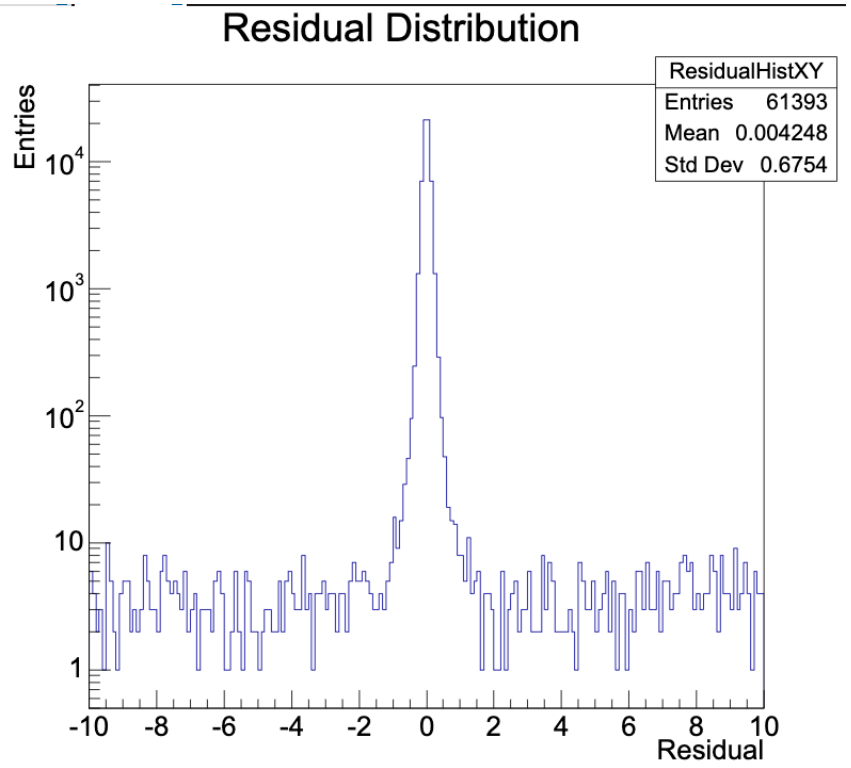
I analyzed in same way as pp

- A particle (μ^-)/event
- # of events : 10K
- Magnetic field : zero field
- Incident point: witdh $(\sigma_x, \sigma_y, \sigma_z)$=(0, 0, 20)cm
- Incident direction : - $\pi < \phi < \pi$, -1< η <1
- No dead channel

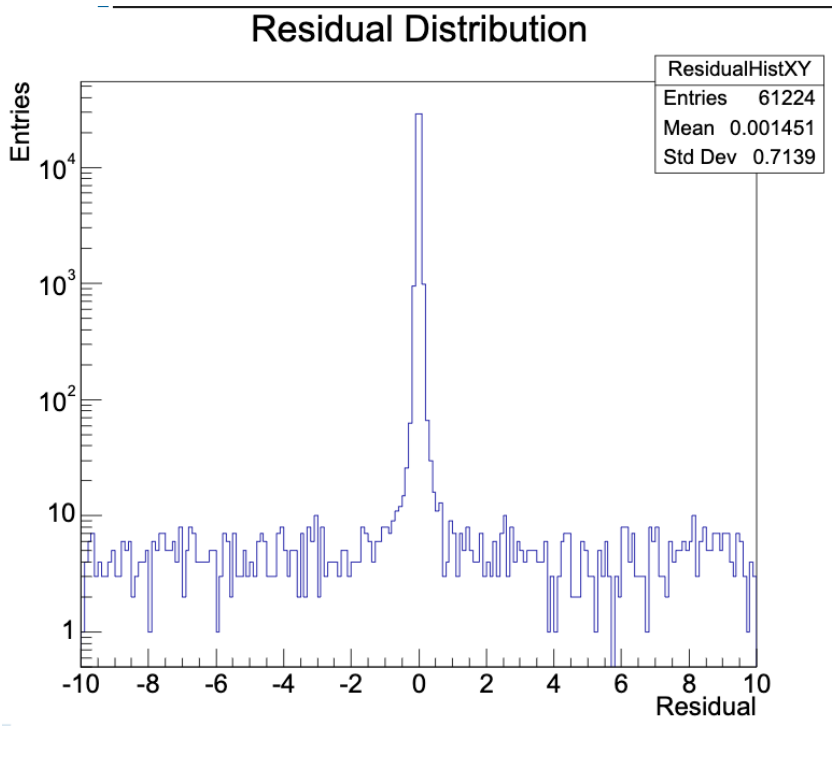
$P_T = 200\text{MeV}$



$P_T = 400\text{MeV}$



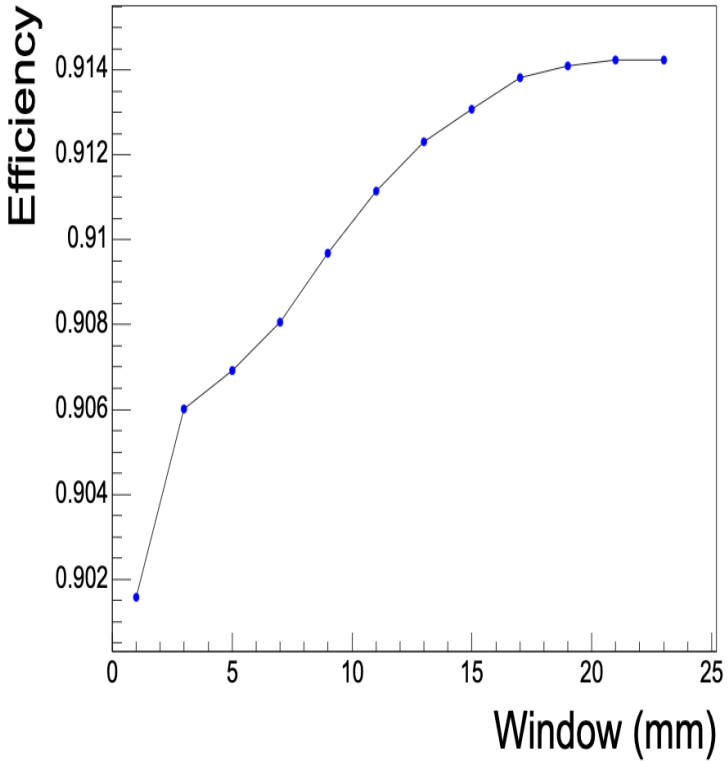
$P_T = 1\text{GeV}$



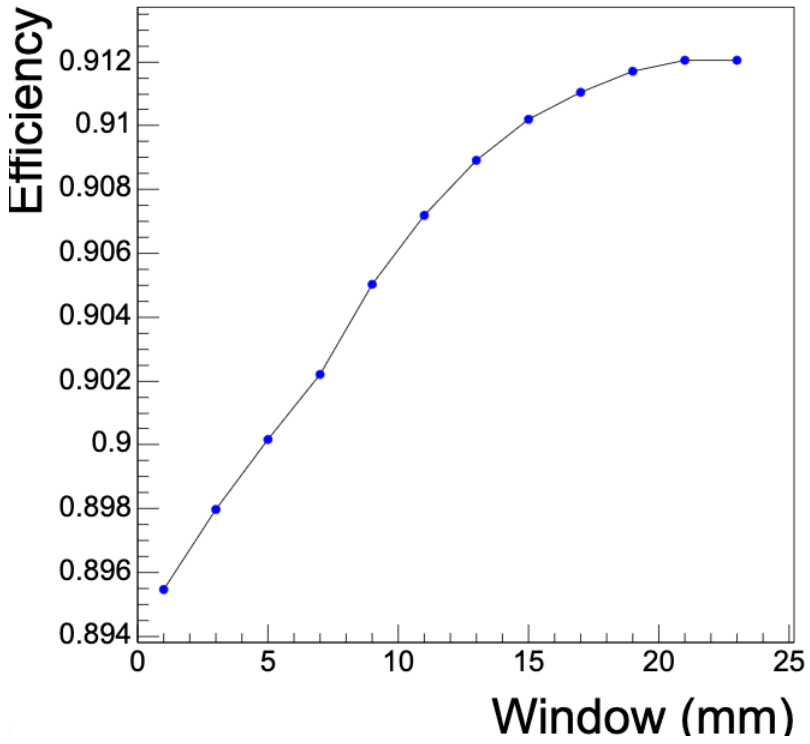
mu MC

- A particle (μ^-)/event
- # of events : 10K
- Magnetic field : zero field
- Incident point: witdh $(\sigma_x, \sigma_y, \sigma_z)$=(0, 0, 20)cm
- Incident direction : $-\pi < \phi < \pi$, $-1 < \eta < 1$
- No dead channel

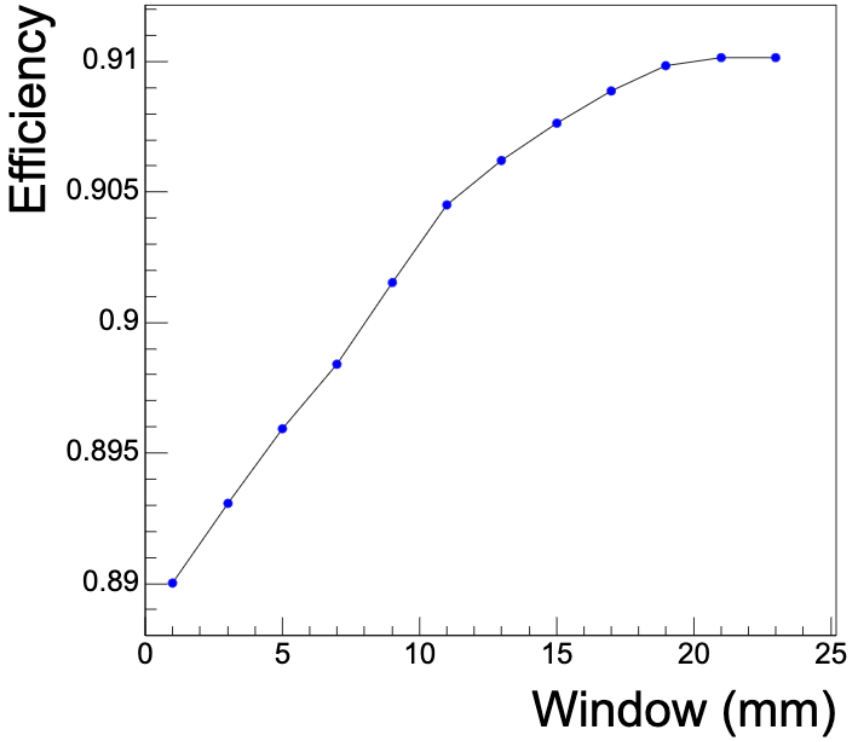
$P_T = 200\text{MeV}$



$P_T = 400\text{MeV}$

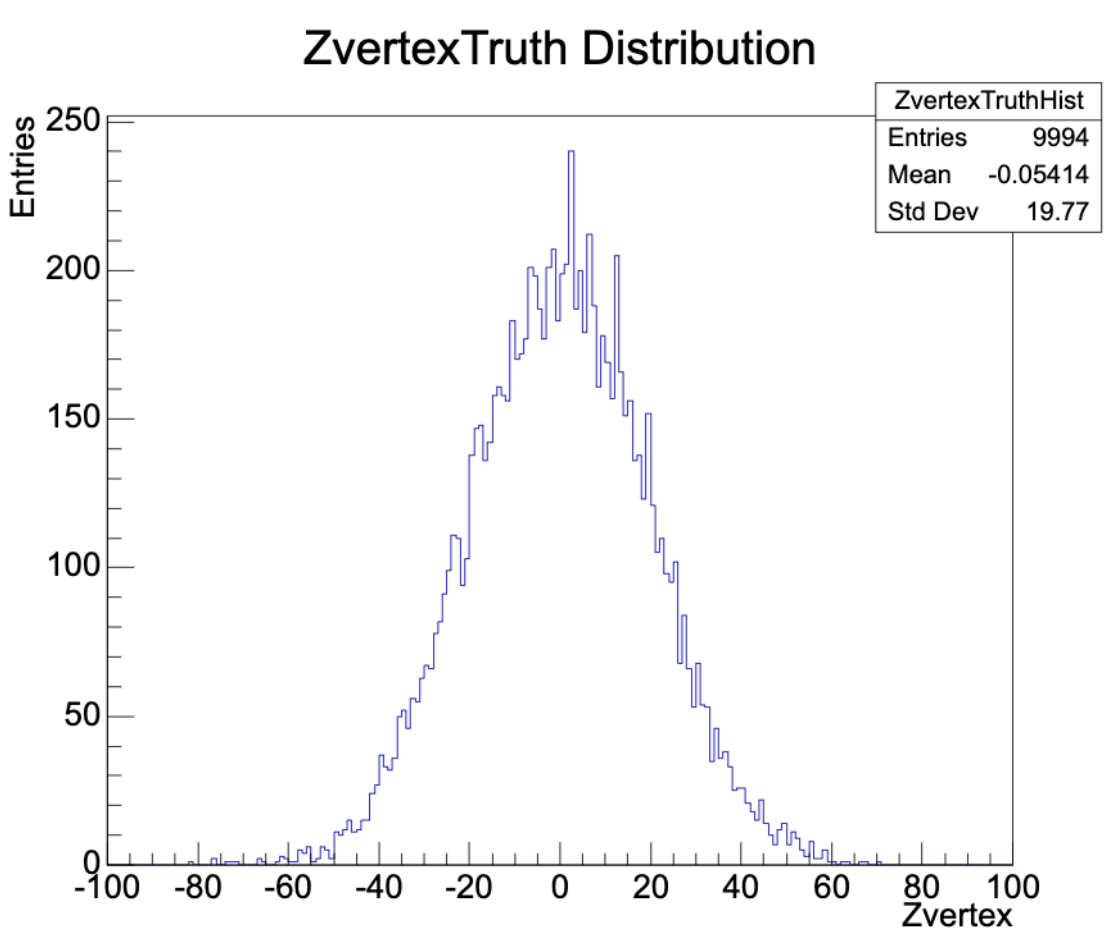


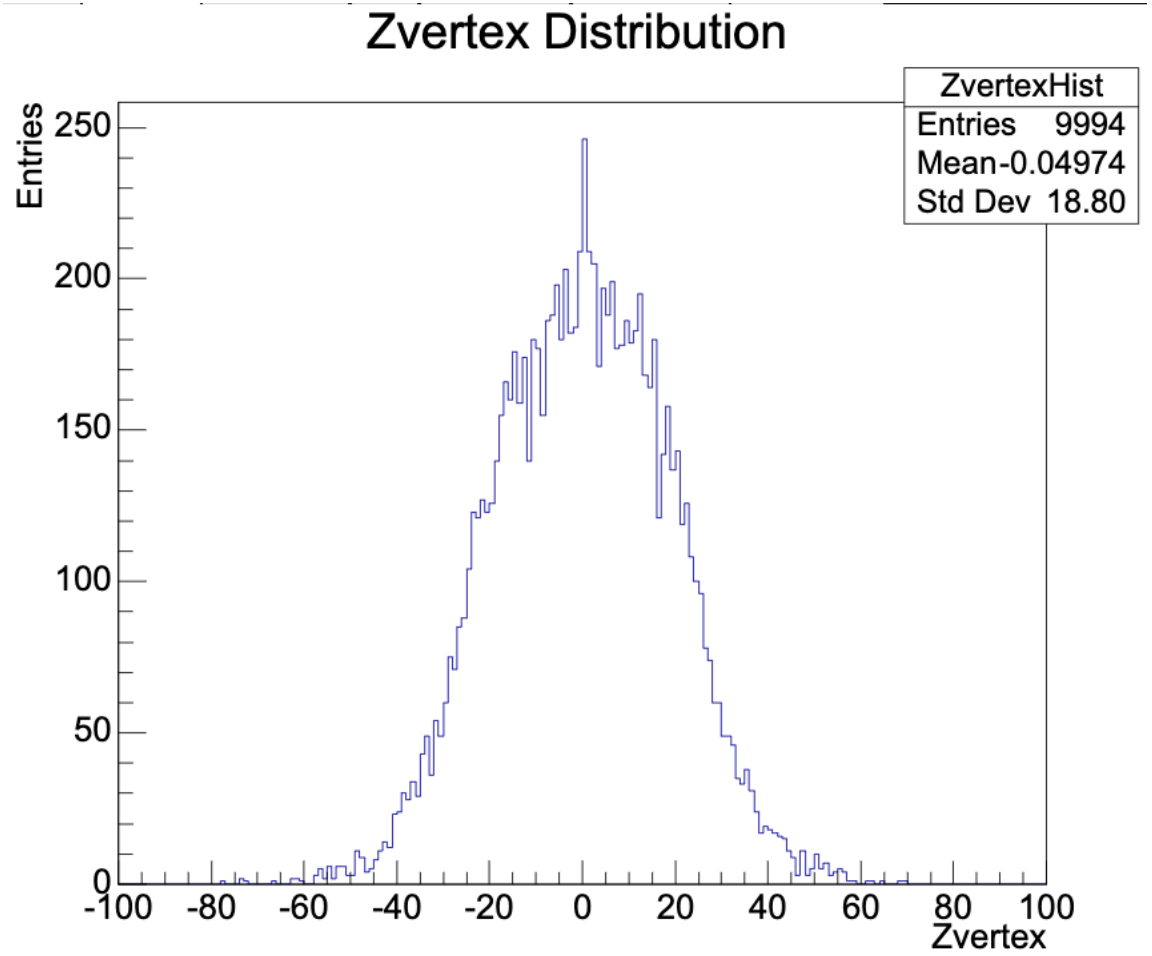
$P_T = 1\text{GeV}$



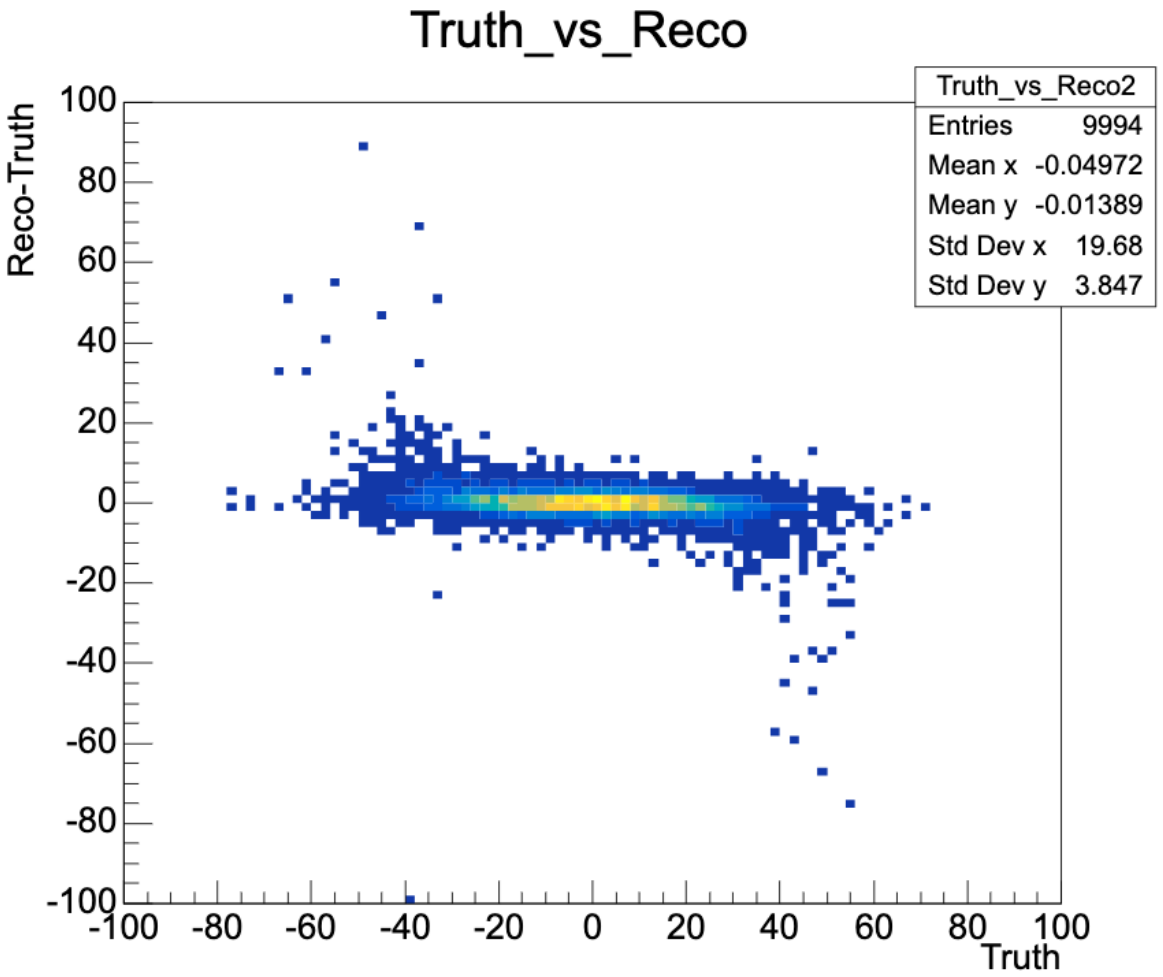
No momentum dependence was observed in the results in my method

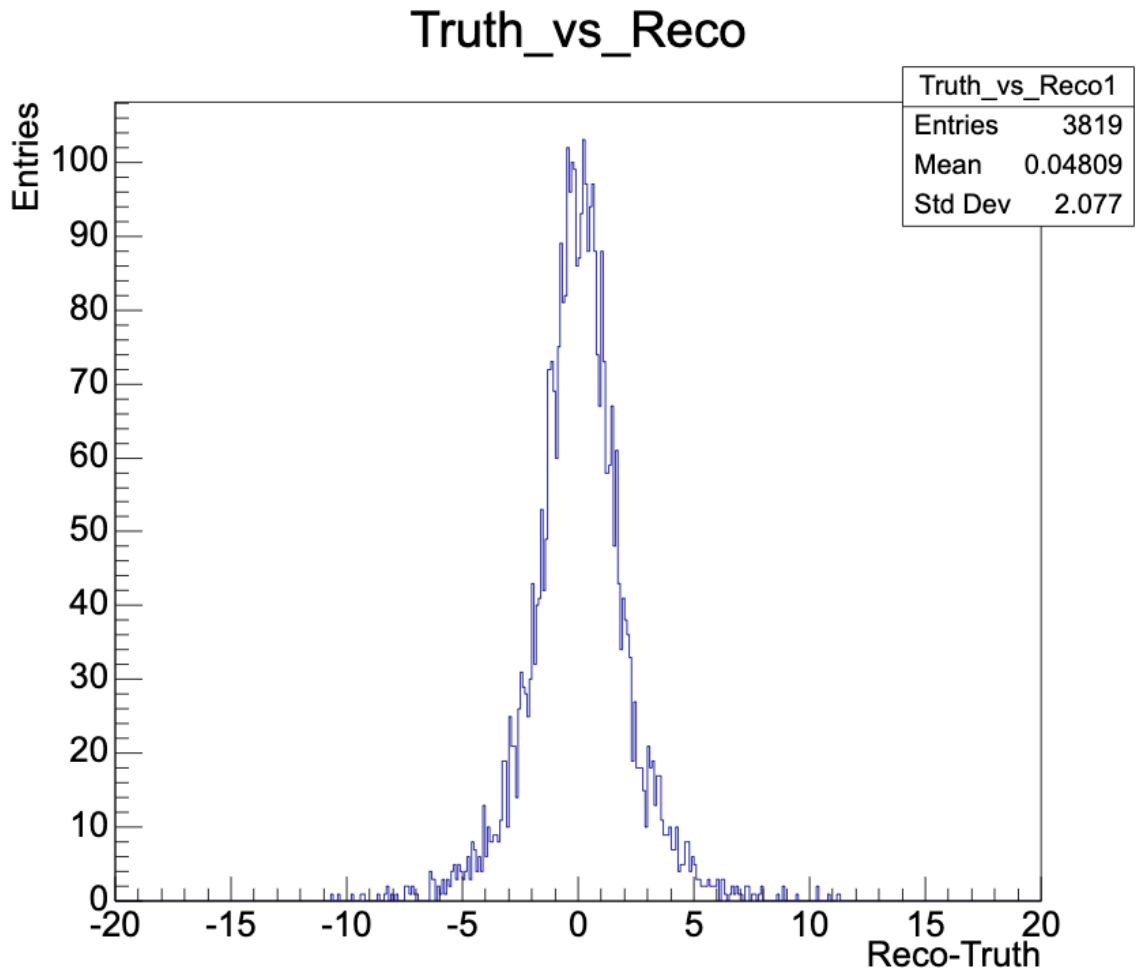
Truth vs reco



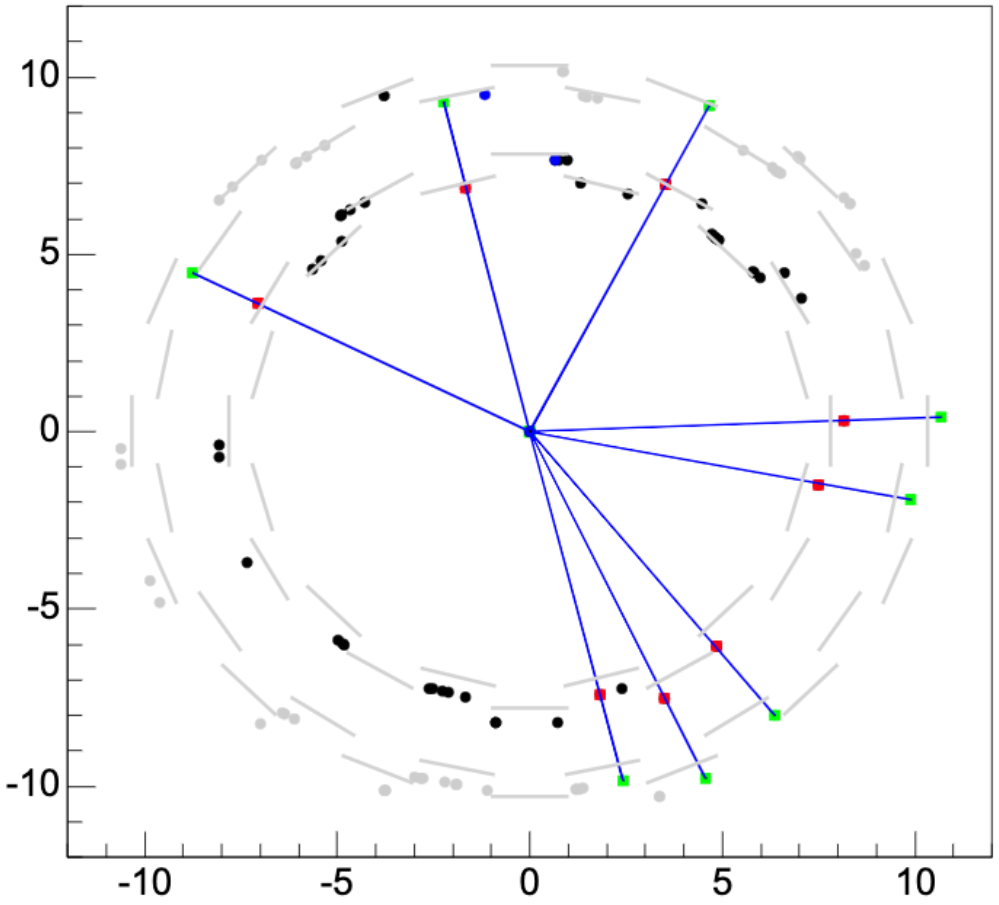


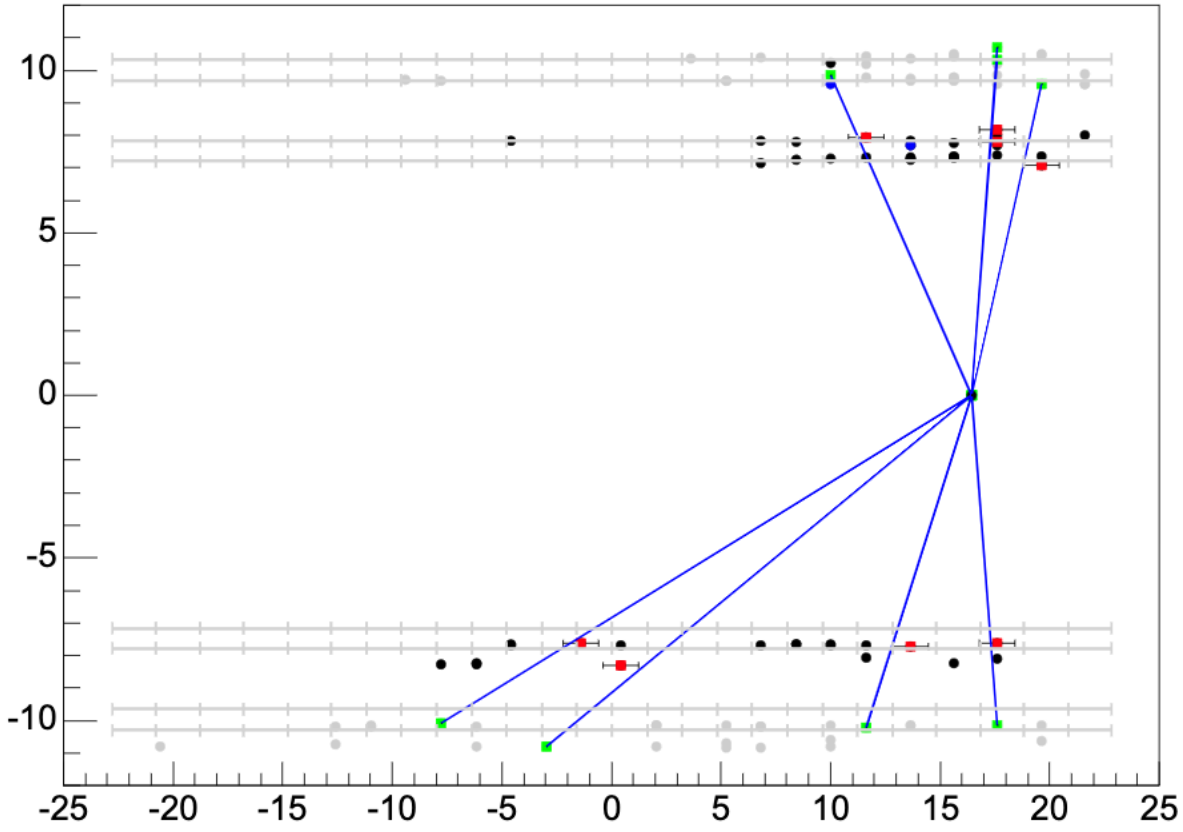
Truth vs reco



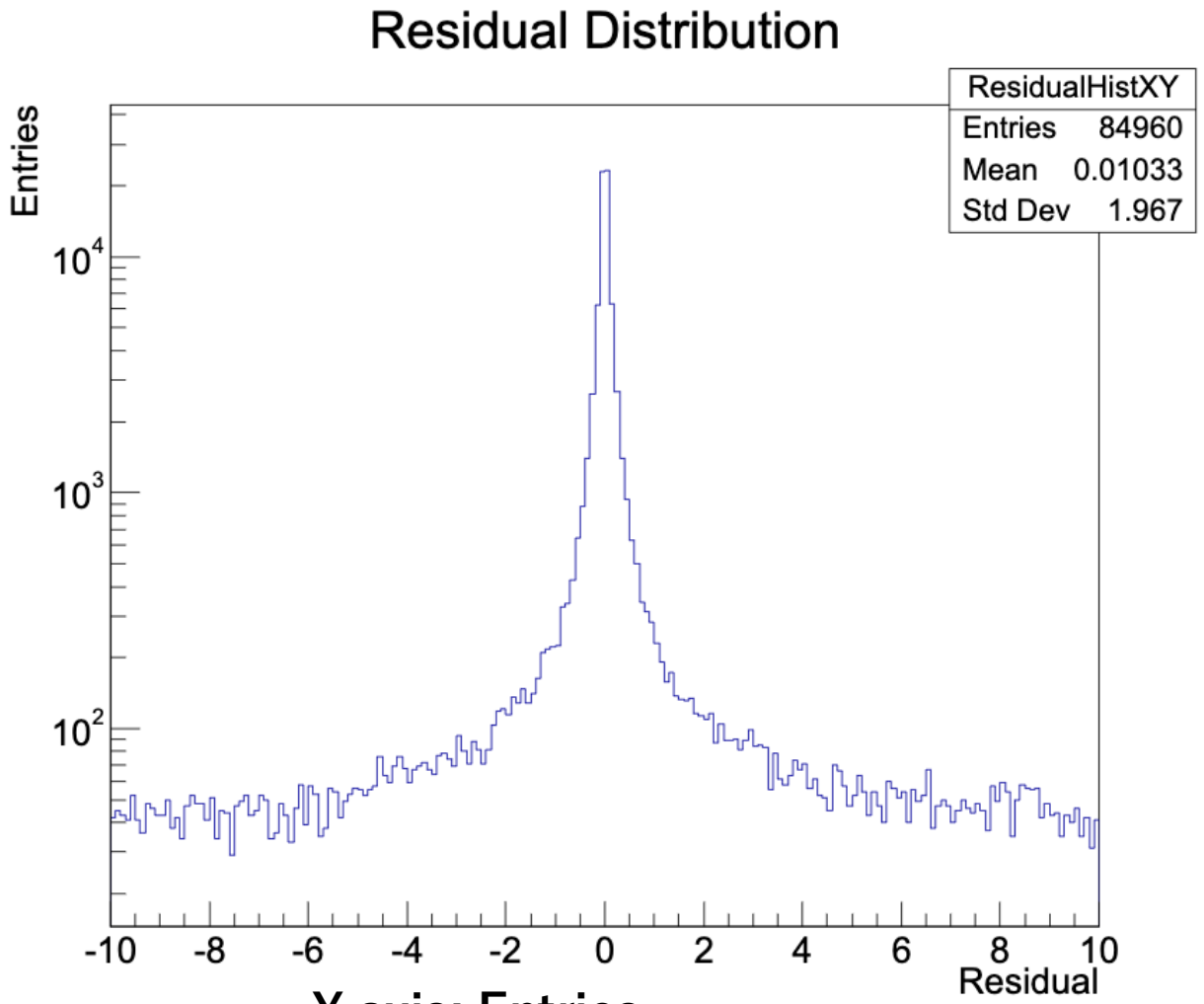


Event display

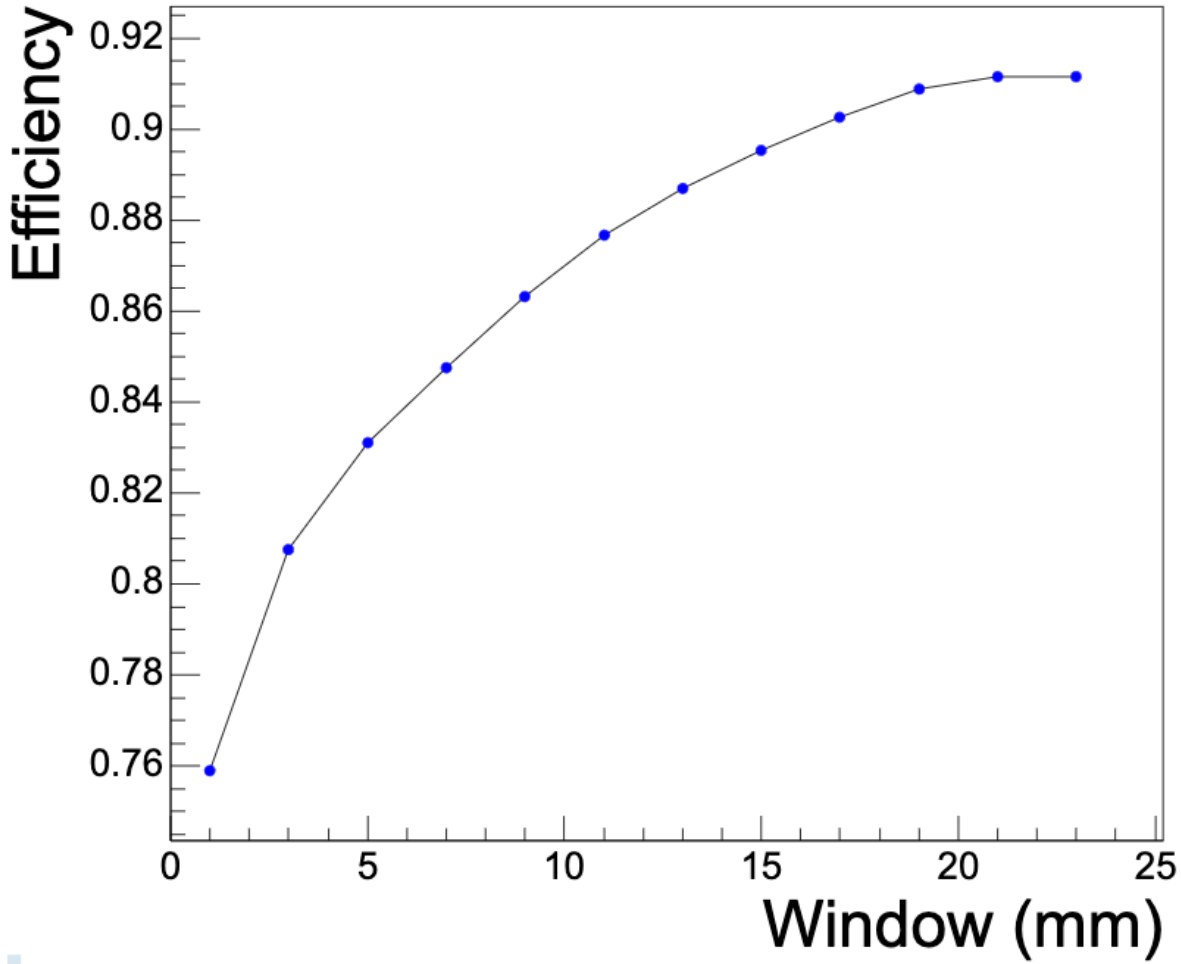




Efficiency (using Reco vertex)



Y axis: Entries
X axis: Residual in X-Y plane



Y axis: Efficiency
X axis: Window (mm)

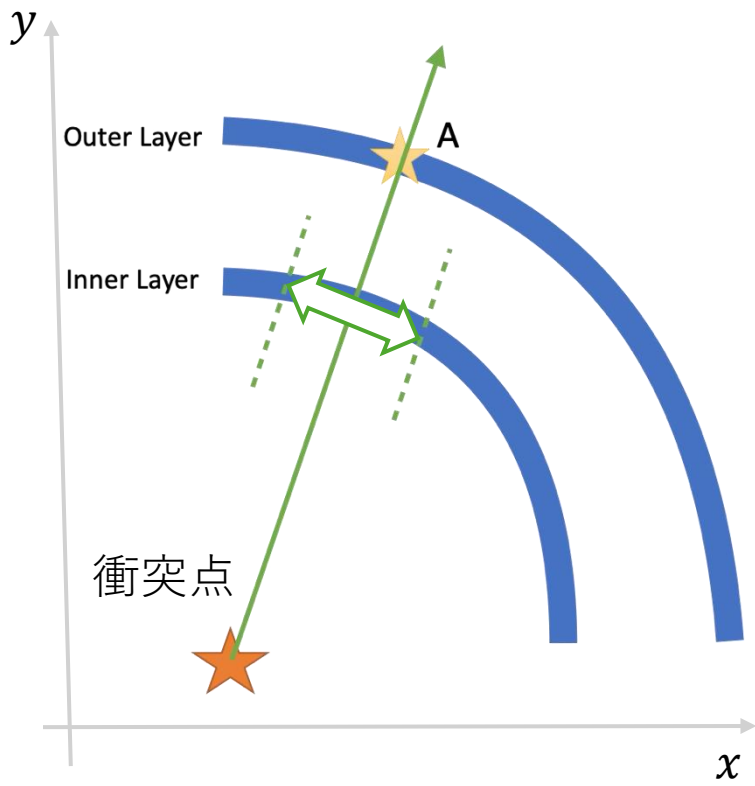
Back up

Method

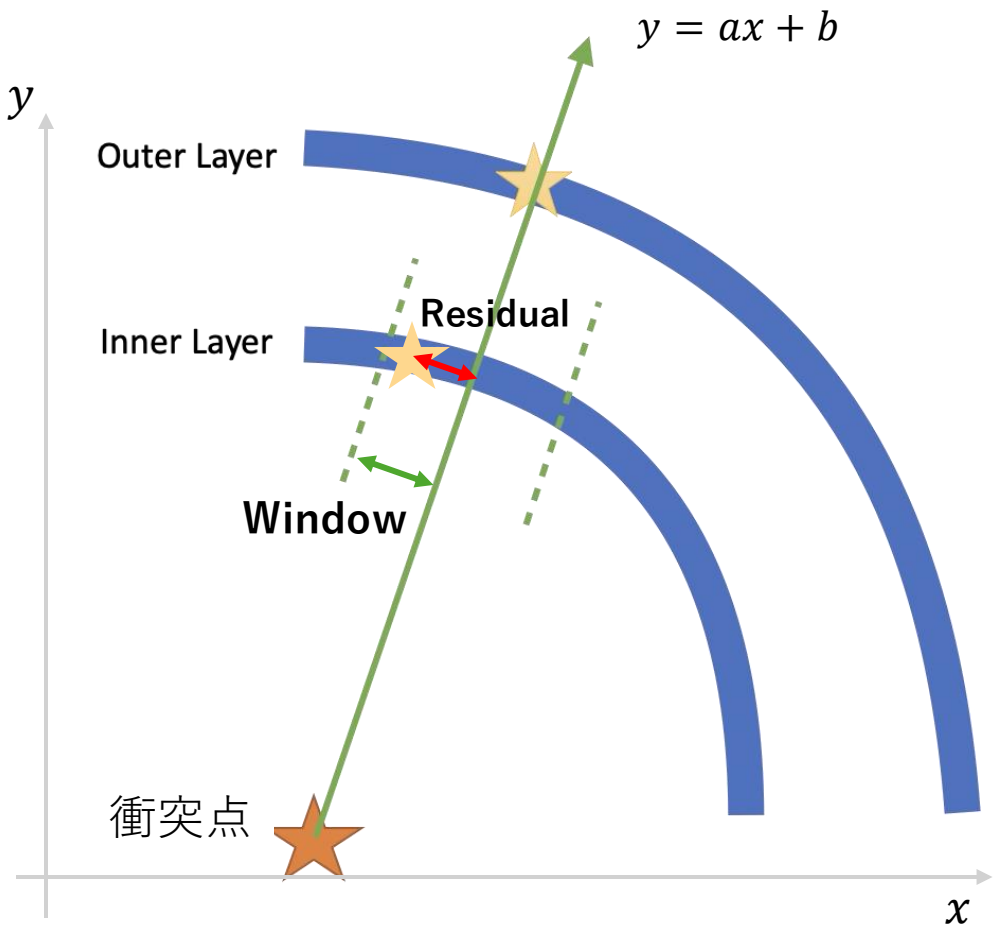
1. Get the coordinates of the cluster in the outer layer in the single event.

2. Determine the expected range in the inner layer using the collision point and cluster A.

3. Check for the presence of clusters within the expected range and count the number of clusters in each case to calculate the detection efficiency.



2. Determine the expected range in the inner layer using the collision point and cluster A.



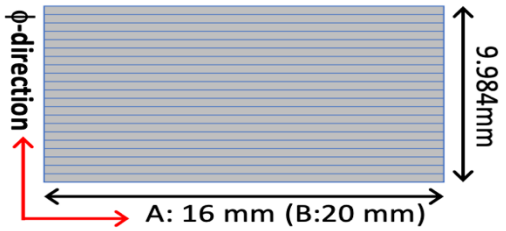
1. Fitting using the outer cluster and the vertex.

2. Calculate "*Residual*" between the fitting line and the inner cluster using the following equation:

$$Residual = \left| \frac{a \cdot x_{in} - y_{in} \cdot b}{a^2 - 1} \right|$$

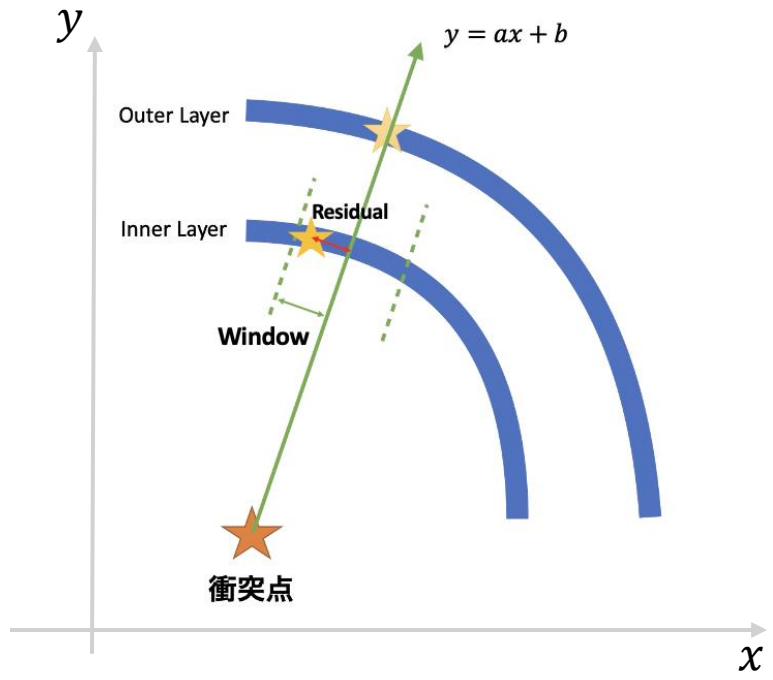
1. Set the expected range ($window$).
   In this case, the window is defined based on the silicon strip width.

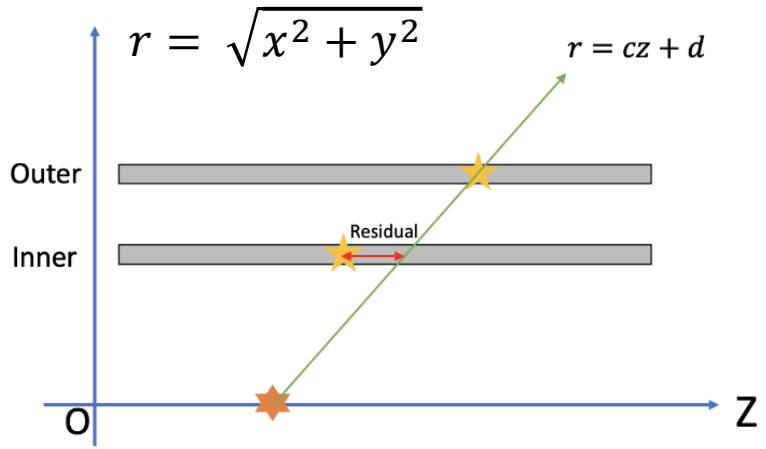
$$window = 78\mu m \cdot i \ (i = 1, 2, 3, \dots)$$



Silicon strip sensor
Thickness 320 μm
Width of strip 78 μm
128 strips

3. Check for the presence of clusters within the expected range and count the number of clusters in each case.





1. Select the Inner Cluster that minimizes d^2 , calculated using the residuals $d_{xy}$and d_z in the XY plane and RZ plane, respectively, based on the following equation:

$$d^2 = \left(\frac{d_{xy}}{\sigma_{xy}}\right)^2 + \left(\frac{d_z}{\sigma_z}\right)^2$$

σ_{xy}: The resolution of the INTT sensor in the x-y plane. 78 μm

σ_z : The resolution of the INTT sensor in the z-axis. 20 mm

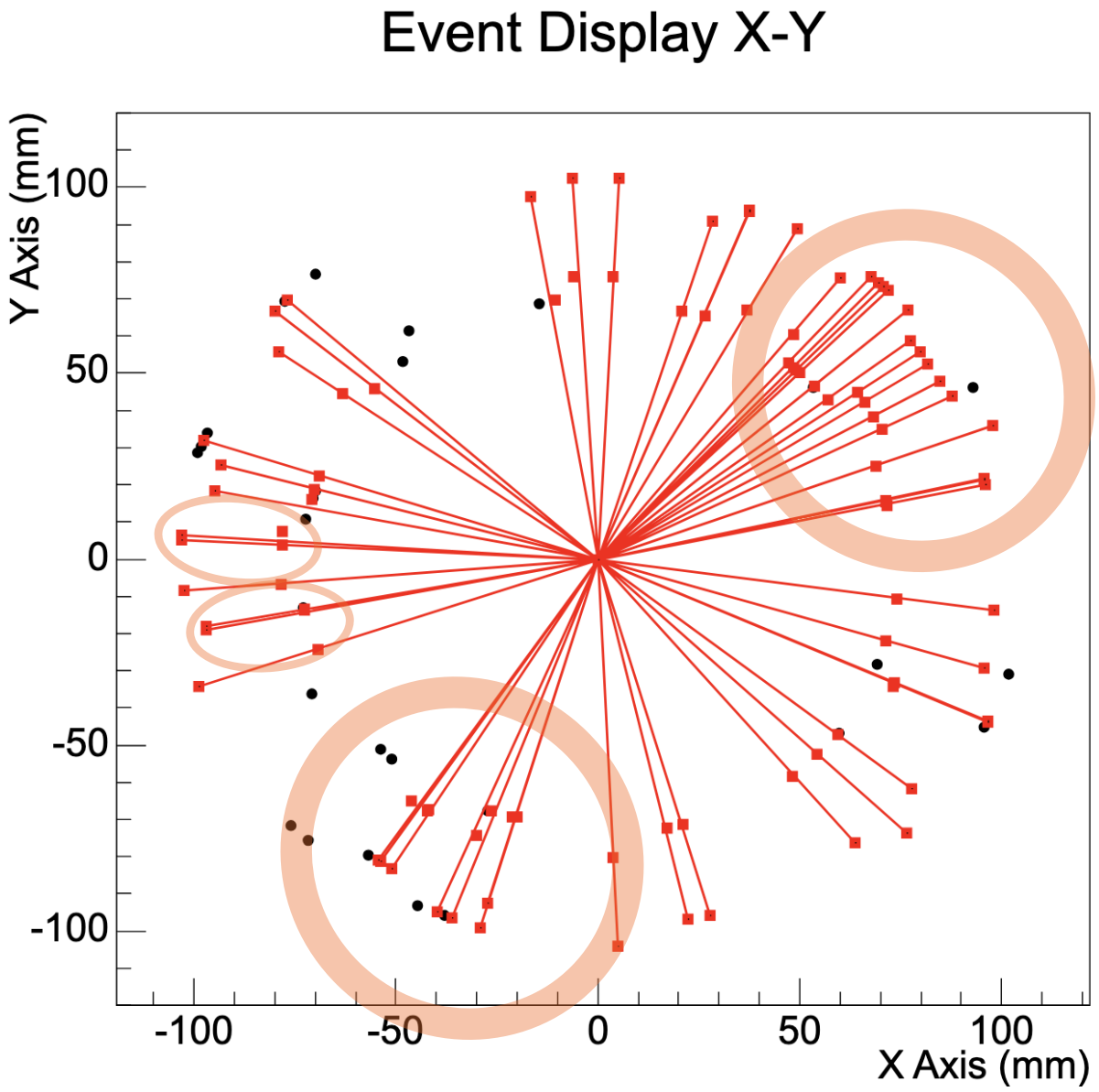
Tag the selected Inner Cluster to prevent it from being counted again.

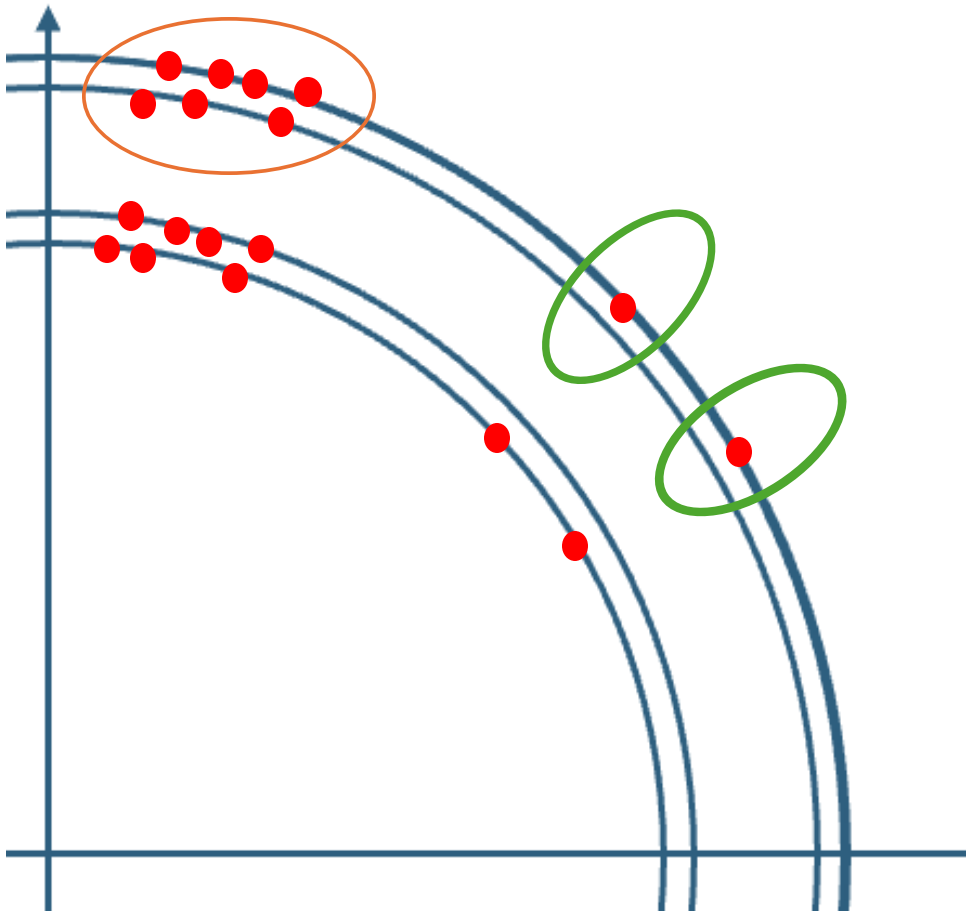
2. Count the Outer Cluster as N_{hit} when an Inner Cluster exists within the expected range in the XY plane $(Residual < window)$.

3. Count the Outer Cluster as $N_{no\ hit}$when no Inner Cluster exists within the expected range.

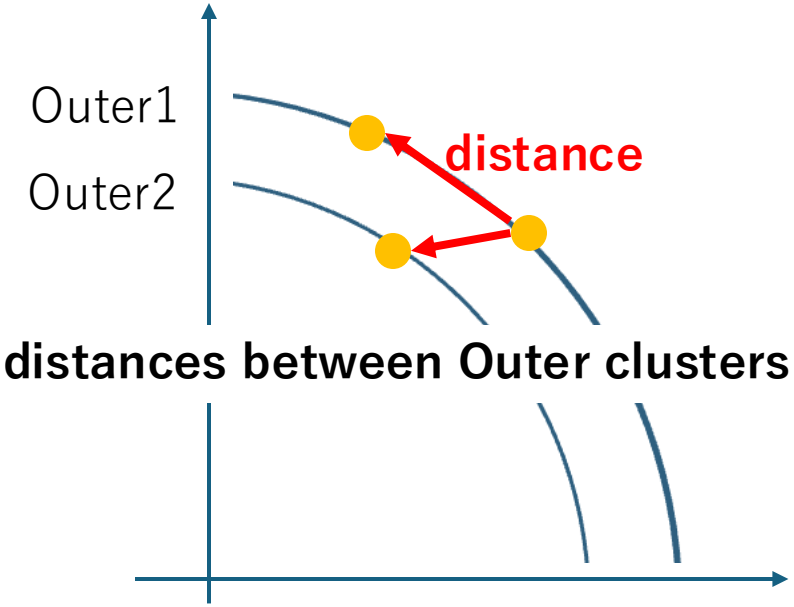
検出効率 $\varepsilon = \dfrac{N_{hit}}{N_{hit} + N_{no\ hit}}$

Cutting high-density areas





Remove high-density areas in the Outer.



The distances between Outer clusters = Density

```
for (size_t i = 0; i < Clusters.size(); ++i) {
    if (lay->at(i) > 1) { // 外層のみループ
        for (size_t j = 0; j < Clusters.size(); ++j) {
            if (i == j || lay->at(j) < 2) continue;

            double dx = Clusters[i].X() - Clusters[j].X();
            double dy = Clusters[i].Y() - Clusters[j].Y();
            double d = std::sqrt(dx * dx + dy * dy);
            d_hist->Fill(d);

            if (d < 10.0) {
                labeledClusters.insert(i);
                labeledClusters.insert(j);
            }
        }
    }
}
```