Charged Track Reconstruction with Silicon Detector and Calorimeter in sPHENIX

Jingyu Zhang

07.08

Work Introduction

Motivation

During the early stages of the sPHENIX experiment, the TPC encountered malfunctions and failed to operate normally, resulting in a large number of pp collision events without any TPC data. In these "without-TPC" events, charged tracks could not be reconstructed via the TPC. We therefore set out to reconstruct charged tracks using the INTT/Silicon detectors together with the calorimeter, aiming to achieve a p_T resolution suitable for physics analysis.

Principle of Operation

Both the INTT/Silicon detectors and the calorimeter are immersed in a magnetic field, the distribution of which is shown in Figure 1. Charged particles originate at the interaction vertex, traveling through the INTT/Silicon detectors and reaching the calorimeter. As they move through the magnetic field, the Lorentz force causes them to bend; their transverse momentum direction is deflected by the Lorentz force in a magnetic field aligned with the z (beam) axis. The curvature is directly related to their transverse momentum p_T, and can therefore be used to reconstruct p_T.

Existing methods include using the position points from the INTT/Silicon detectors together with the calorimeter hit to fit the sagitta R of the curved trajectory and compute p_T via the relation $p_T = qBR$, or describing the deflection by an angle $\Delta\Phi$ from INTT/Silicon and Calo position, then estimating p_T with the function $p_T = f(\Delta\Phi)$.

Our current work is dedicated to electron reconstruction, so only the EMCal component of the calorimeter is employed.

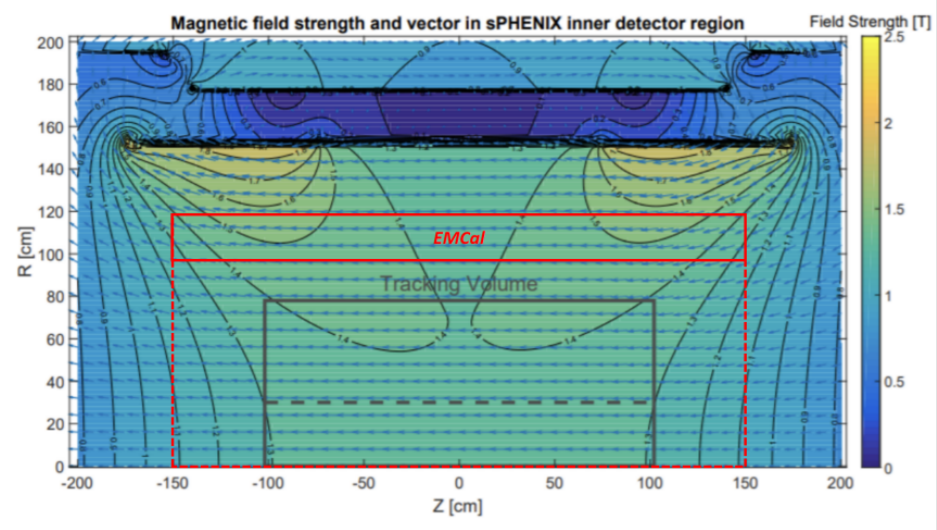


Figure 1: sPHENIX magnetic field

Data Preparation

For our charged particle track reconstruction, we require information from both the INTT/Silicon detectors and the calorimeter. This includes position data as particles traverse the INTT/Silicon, as well as position and energy data when particles pass through or hit the calorimeter.

INTT/Silicon Position

EMCal Overview

The electromagnetic calorimeter (EMCal) is a key sub-detector in the sPHENIX experiment used to measure the energy of electromagnetic particles such as electrons and photons. It is a sampling calorimeter constructed from scintillating fibers embedded in tungsten powder. The sPHENIX EMCal is located at a radial distance of approximately 100 cm, symmetrically arranged around the beam axis, covering $|\eta| < 1.1$ and the full 2π azimuthal range.

The EMCal comprises 6,144 blocks, each containing four independent readout units called towers—the smallest readout elements—each read out by four SiPMs. The current design covers a pseudorapidity range of $|\eta| \leq 1.1$ and full 2π in azimuth, with an inner radius of approximately 97.5 cm. It employs a segmentation of 0.025×0.025 in $\Delta\eta \times \Delta\phi$, resulting in $96 \times 256 = 24{,}576$ towers. Each tower corresponds to an approximately 0.025×0.025 region in η-ϕ space, capable of recording the electromagnetic energy deposited within that region. To balance transverse spatial resolution and energy measurement precision, the EMCal design incorporates tilted towers and a zig-zag arrangement to reduce shower leakage and improve uniformity of response with respect to the incident direction. These blocks and towers are assembled into $2 \times 32 = 64$ sectors (see Figure 2, which are then combined to form the complete EMCal.

The EMCal measures energy by detecting the electromagnetic shower process initiated by electrons and photons within its volume. When a charged particle (e.g., an electron) or a neutral particle (e.g., a photon) enters the calorimeter, it interacts with the absorber

(a) EMCal XY (b) EMCal RZ

Figure 2: EMCal arrangement

material via bremsstrahlung or pair production, generating a cascade of secondary e^-/e^+ and γ photons—known as an electromagnetic shower. This shower deposits energy across multiple towers, producing a characteristic two-dimensional energy distribution. To make use of this information, the energy readout from each tower is used as input for cluster reconstruction. This process identifies and groups spatially adjacent, energy-correlated towers to form clusters that represent individual showers. A cluster is the fundamental physics object in EMCal reconstruction, providing the particle's energy deposition, spatial centroid, and shower width—information essential for subsequent analyses such as track matching, electron identification (eID), and photon reconstruction.

Cluster Position Reconstruction in EMCal

In our study, we want the EMCal to provide the electron's position at approximately $R = 100$ cm. When an electron produces an electromagnetic shower in the EMCal, it deposits energy across several towers. We record the identifiers of these towers and their deposited energies, then use this information to reconstruct a cluster object. A cluster is the fundamental physics object in EMCal reconstruction: it supplies the particle's energy in the calorimeter, its spatial position (centroid), and the shower's lateral spread. Clusters form the basis for subsequent analyses such as track matching, electron identification (eID), and photon reconstruction.

The reconstruction algorithm is implemented in the `RawClusterBuilderTemplate` class. First, it applies an energy threshold to remove background noise; towers exceeding this threshold are treated as signal. Next, it groups spatially adjacent towers into clusters, summing their energies to obtain the cluster energy. The cluster position is computed as the energy-weighted average of its constituent towers, and additional features (e.g., χ^2) are derived from the internal energy distribution.

Mathematically, for a cluster containing towers indexed by i:

$$E_{\text{cluster}} = \sum_i E_i, \quad \mathbf{x}_{\text{cluster}} = \frac{1}{E_{\text{cluster}}} \sum_i \mathbf{x}_i \, E_i,$$

where $\mathbf{x}_i$ denotes the position of tower i. In the EMCal, the reconstructed particle position—the cluster position, that is obtained as the energy-weighted average of its constituent tower positions. By specifying each tower's position, one reconstructs the positions of clusters

composed of those towers. In our implementation, each tower's position $\mathbf{x}_i$ can be defined in two ways:

(1) Using the default EMCal software setting:

```
findNode::getClass<RawTowerGeomContainer>(topNode, m_TowerGeomNodeName);
```

```
m_TowerGeomNodeName = "TOWERGEOM_CEMC";
```

We can get the tower default geometry on this node, Towers are projected onto a cylinder at $R = 93.5$ cm. Each tower's position is taken as the midpoint of its z-face projection (see Figure 3).

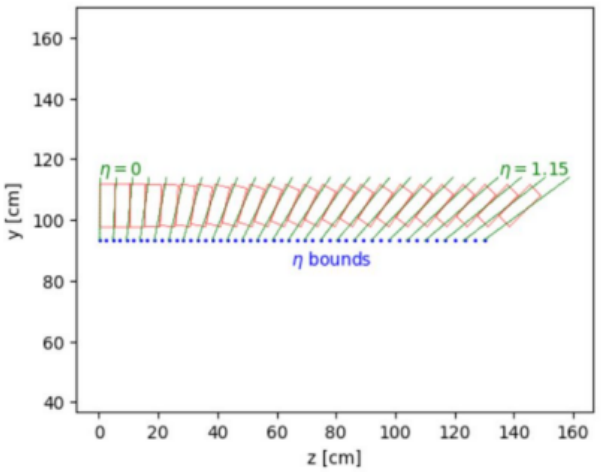


Figure 3: EMCal arrangement

(2) Using either the inner-face center or the geometric center of each tower. In addition to projecting each tower to a radius of 93.5 cm, we can describe its full geometry via Virgile's `CaloGeomMappingv2` class, which writes detailed geometry into `RawTowerGeomv5` by calling `setTowerGeomNodeName("exact_geom_nodename")`. This geometry defines the eight vertices of each tower. From these we compute:

- the inner face center of each tower (the centroid of its four vertices),
- and the geometric center of each tower (the centroid of all eight vertices).

In our study, we use the inner-face center as the tower position x to represent the cluster impact point on the EMCal surface, and the geometric center as the tower position x when reconstructing the cluster position inside the EMCal.

When reconstructing tower positions, simply read the `RawTowerGeomContainer` node named `m_TowerGeomNodeName = "TOWERGEOM_CEMC"`. In `RawClusterBuilderTemplate`, call `SetTowerGeometry()` to pass each tower's position x, then compute the cluster position as the energy-weighted average:

$$x_{\text{cluster}} = \frac{\sum_i x_i E_i}{\sum_i E_i}.$$

In my modified `RawClusterBuilderTemplate` class, I added two interfaces:

```
void setUseRawTowerGeomv5(bool flag = true){ m_use_RawTowerGeomv5 = flag; }

void setProjectToInnerSurface(bool flag = true)
        { m_project_tower_innersurface = flag; }
```

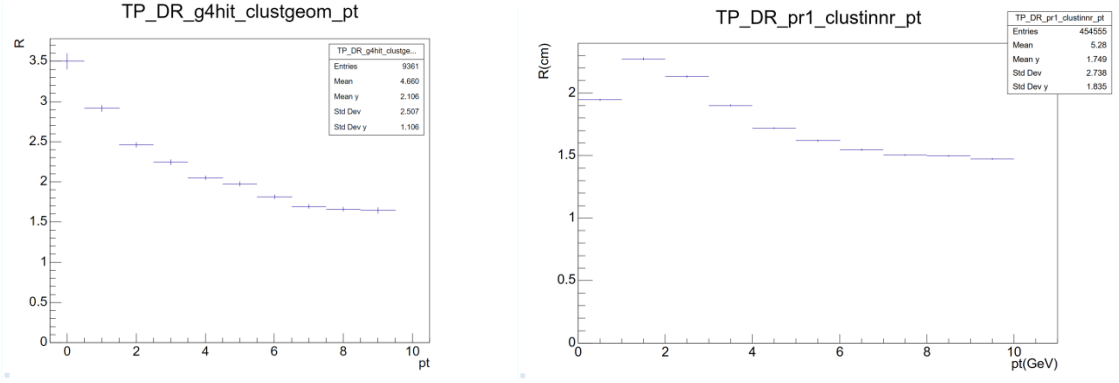
to enable cluster reconstruction using `RawTowerGeomv5`. Setting `m_use_RawTowerGeomv5 = true` uses the precise tower geometry rather than a single projected point, and setting `m_project_tower_innersurface = true` uses the tower's inner-face center (producing `cluster_innr` on the EMCal inner surface), and when `m_project_tower_innersurface = false` uses the tower's geometric center (producing `cluster_geom` inside the EMCal).

In EMCal electron reconstruction, the cluster energy is the sum of its constituent towers' energies, and the cluster position is the energy-weighted average of their positions. We can reconstruct:

- `cluster_proj`: projection at 93.5 cm,
- `cluster_innr`: using the tower inner-face center,
- `cluster_geom`: using the tower geometric center.

In the single-electron simulation, the electron's position in the EMCal is available at both the truth level (from G4Hit) and the reconstruction level. We compare these positions to study the position resolution and to develop corrections for improved accuracy.

Specifically, we examine Distance-R between the reconstructed cluster geometric center (`cluster_geom`) and the shower centroid, as well as between the inner-face cluster center (`cluster_innr`) and the primary electron's first hit on the CEMC, as illustrated in Figure 4, where Distance-R is defined as Distance-R $= \sqrt{\Delta x^2 + \Delta y^2}$.



(a) cluster geom performance (b) cluster innr performance

Figure 4: Distance between electron reconstruction position and first G4Hit on CEMC

Compared to the shower centroid, the primary electron's first hit on the CEMC more faithfully represents the true energy-deposition point in the simulation. Moreover, the distance between `cluster_innr` and the primary electron first hit is observed to be smaller, indicating that using the inner-face centroid yields better position resolution.

In studying the separation between the reconstructed inner-face cluster center (`cluster_innr`) and the truth-level primary electron's first hit position on the CEMC (see Figure 5), we decompose this distance into tangential and radial components. Under the small-angle approximation, the tangential displacement is given by $d\phi \cdot R = (\phi_{\text{truth}} - \phi_{\text{reco}})\, R_{\text{radius}}$ and the radial displacement by $dR = R_{\text{reco}} - R_{\text{truth}}$.

In studying the separation between the reconstructed cluster with tower inner face center (`cluster_innr`) and the truth-level primary electron's first hit position on the CEMC (see Figure **??**), we decompose this distance into tangential and radial components: under the small-angle approximation, the tangential displacement is $d\phi\, R = (\phi_{\text{truth}} - \phi_{\text{reco}})\, R_{\text{radius}}$ and the radial displacement is $dR = R_{\text{reco}} - R_{\text{truth}}$.

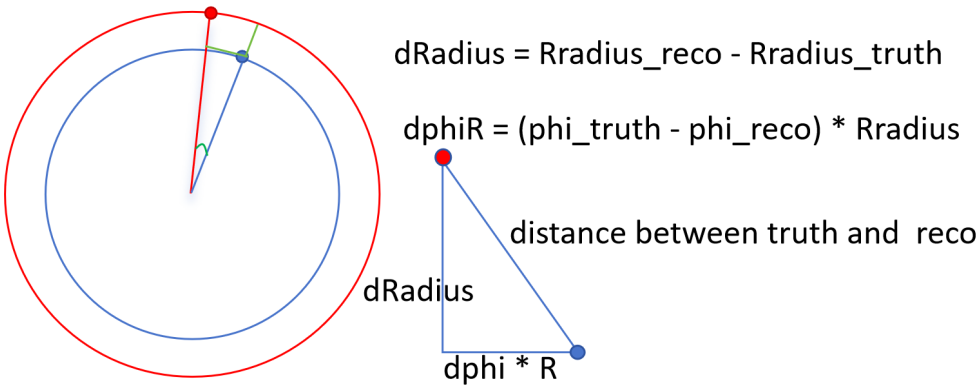
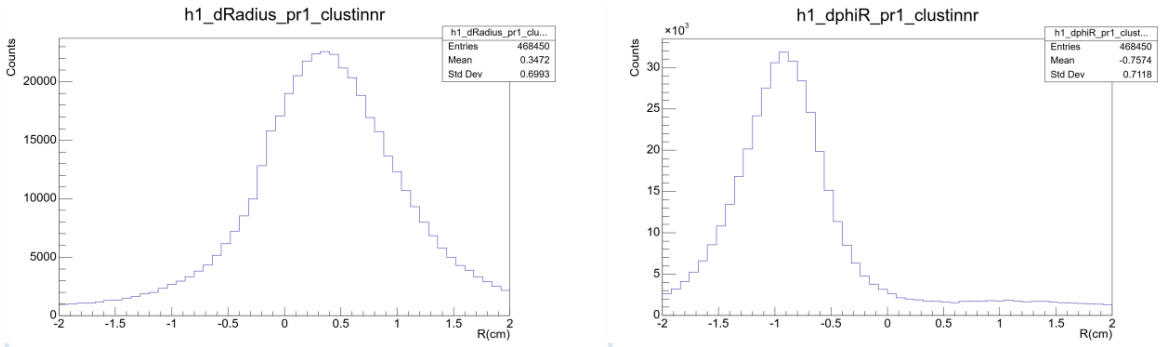


Figure 5: Dirrection seperate

After decomposing this distance into radial (ΔR) and tangential ($\Delta\phi \times R$) components and plotted their distributions (Figure 6). A bias was observed, with a systematic shift in $\Delta\phi$ and ΔR, indicating that the reconstruction introduces an angular bias.



(a) dR on electron

(b) dphiR on electron

Figure 6: Distance in two direction: radial (ΔR) and tangential ($\Delta\phi \times R$)

In Figure 7, we show the ϕ difference between the reconstructed inner-face cluster center (`cluster_innr`) and the truth-level primary electron's first hit on the CEMC, ($\mathrm{d}\phi = \phi_{\mathrm{truth}} - \phi_{\mathrm{reco}}$. The reconstructed position exhibits an angular bias relative to the true position: the $d\phi$ distribution peaks well below zero, indicating a systematic angular offset in the cluster_innr reconstruction.

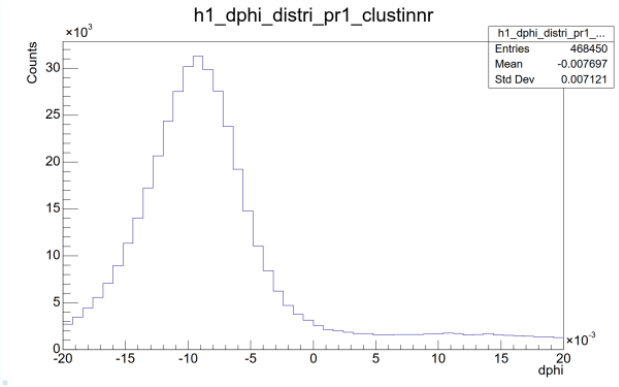
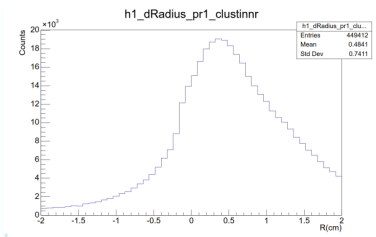
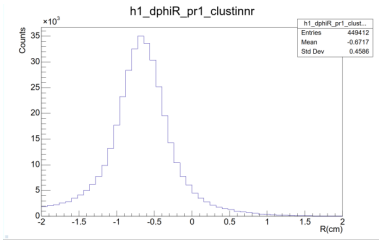


Figure 7: dphi on electron

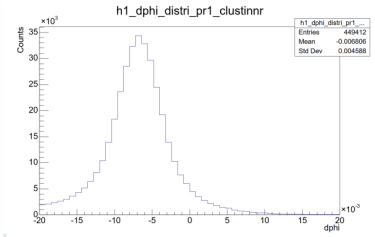
In positron reconstruction, we studied the tangential and radial components of `cluster_innr`, as shown in Figure **??**. Similar to the electron case, the reconstructed `cluster_innr` ϕ angle exhibits a bias. Due to the effect of the positive charge, this bias in positron reconstruction is slightly reduced but maintains the same sign, with the peak of the $\Delta\phi$ distribution occurring below zero.



(a) dR on positron



(b) dphiR on positron



(c) dphi on positron

Figure 8: Positron position reconstruction performance.

In p_T reconstruction, this angular bias causes underestimation of electron pT and overestimation for positrons. This bias produces opposite distortions for electrons and positrons, As shown in Figure 9: for electrons, the angular shift is in the same direction as their magnetic deflection, causing the reconstructed p_T to be underestimated (and the underestimation grows with increasing true p_T); for positrons, the shift is opposite to the deflection, leading to an overestimation of reconstructed p_T that likewise increases with true p_T.

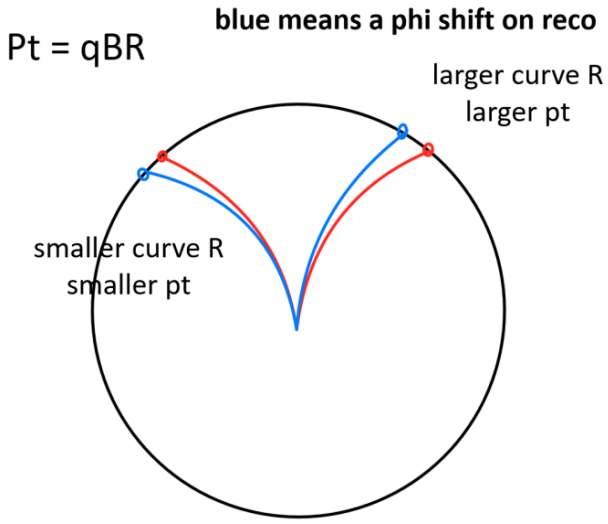


Figure 9: phi angle reconstruct bias effect

Based on the observed offset between the reconstructed cluster with tower inner face center (`cluster_innr`) and the truth primary electron's first hit on the CEMC, we apply a simple bias correction to all `cluster_innr` positions by adding the peak values in both the tangential and radial directions. The corrected positions are illustrated in Figure 10, where each original reconstructed point (blue) is rotated by a fixed angle $\Delta\phi_{\text{peak}}$ and shifted in radius by ΔR_{peak}, yielding the green points that lie closer to the red true hit locations.

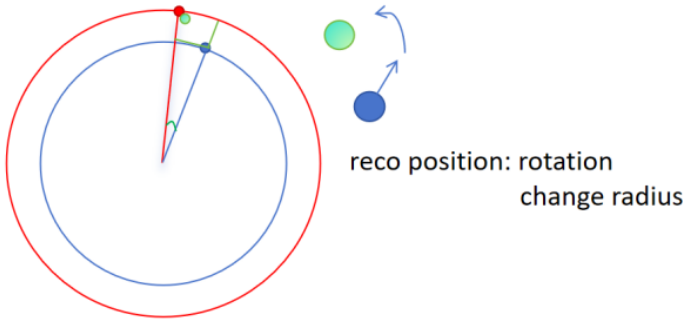


Figure 10: cluster innr reconstruct correction

After applying the bias corrections, the reconstructed positions align more closely with the true hit locations, eliminating the ϕ-angle bias and improving the p_T reconstruction for both electrons and positrons. As shown in Figure 11, using uncorrected `cluster_innr` positions yields an underestimation of electron p_T that increases with true p_T, and an overestimation of positron p_T that likewise grows with true p_T. After the correction is applied, this bias is effectively removed.

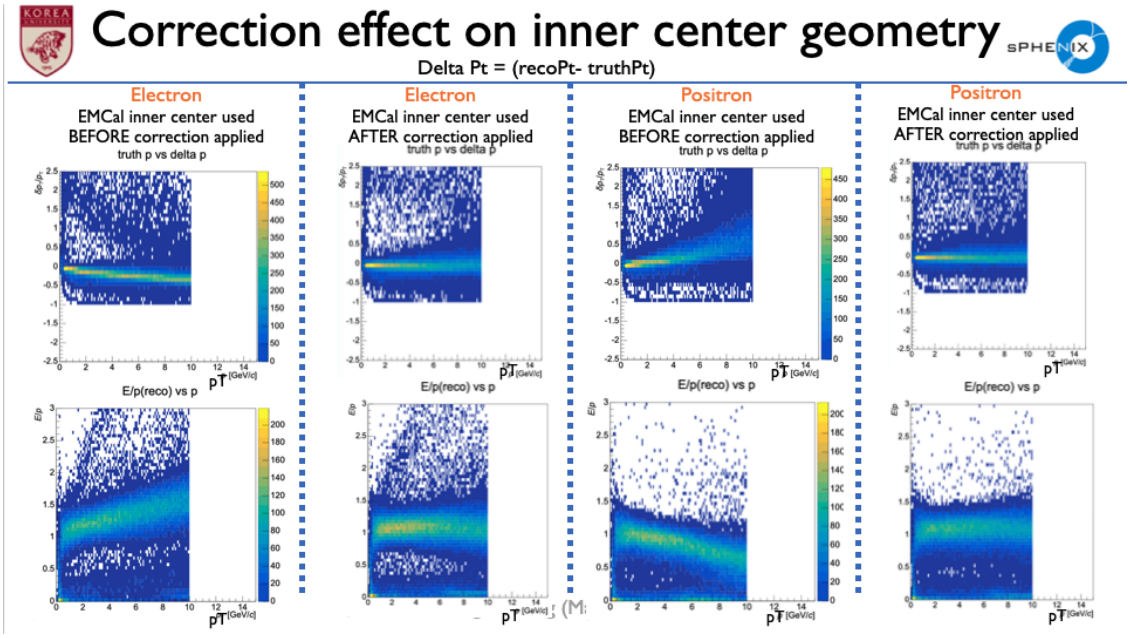


Figure 11: correction effect

Alternative p_T Reconstruction Methods

Position Method

Electrons originate at the interaction vertex and travel through the INTT/Silicon detectors into the calorimeter, bending under the Lorentz force in the magnetic field. By fitting the sagitta R from the INTT/Silicon and calorimeter hit positions, we calculate

$$p_T = q\,B\,R.$$

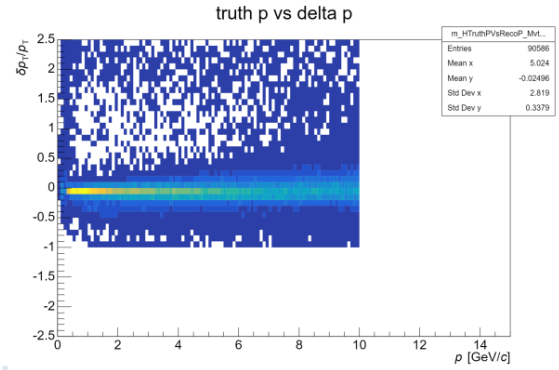
Figure 12 shows the Position Method performance for 0–10 GeV electron reconstruction. It excels at low momentum, but at high p_T small errors in position translate into large p_T deviations, degrading its performance in that region.

Energy Method

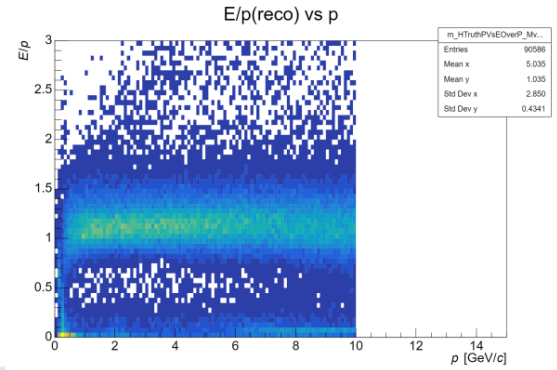
After the charged particle hits the calorimeter and deposits energy, we record both its hit position and deposited energy. From the particle's energy E and pseudorapidity η, we project the energy onto the xy-plane (perpendicular to the beam) to obtain

$$p_T = E\,\sin\theta, \quad \theta = 2\arctan\left(e^{-\eta}\right).$$

Figure 13 shows the Energy Method performance in the 0–10 GeV range. Because the EMCal energy resolution is $16\%/\sqrt{E}$, relative resolution improves with increasing E, and p_T correlates positively with E. The Energy Method thus performs better at high momentum and does not significantly overestimate the reconstructed p_T.

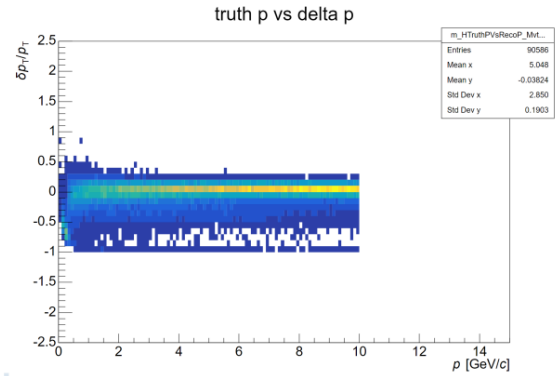


(a) /qBR: pt resolution

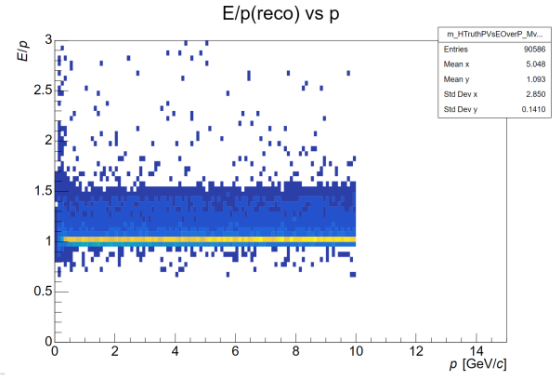


(b) qBR: E/P

Figure 12: Position method (pt = qBR) performance



(a) /Esintheta: pt resolution



(b) Esintheta: E/P

Figure 13: Energy method performance

Combined Approach

From the results in Figure 12 and Figure 13, the two methods exhibit complementary characteristics. Combining them not only provides cross-validation but also leverages each method's optimal momentum range.

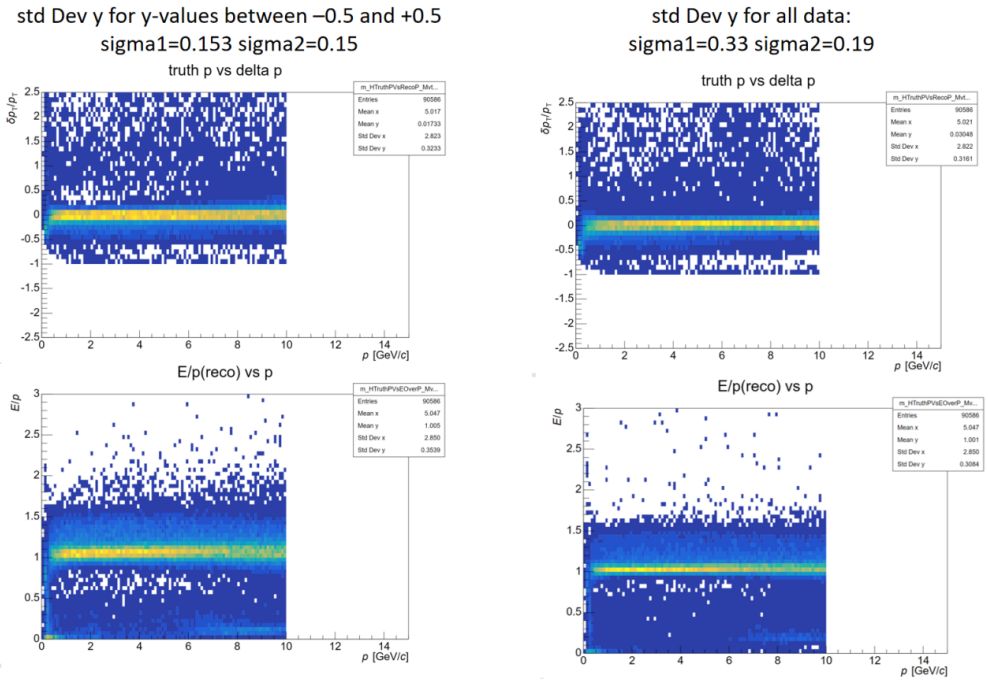


Figure 14: Combined Performance within diff. weight

Assuming the Position and Energy Methods are statistically independent, we form a weighted average of their p_T estimates:

$$\hat{p}_T = \frac{w_1\, p_{T,1} + w_2\, p_{T,2}}{w_1 + w_2}, \quad w_i = \frac{1}{\sigma_i^2},$$

where subscript 1 refers to the Position Method and 2 to the Energy Method, and σ_i denotes the resolution of each method. Larger σ_i implies smaller weight w_i, giving more emphasis to the more precise method.

Determining σ_i is nontrivial. Initially, we used the global standard deviations $\sigma_1 = 0.33$, $\sigma_2 = 0.19$, and then restricted to events with $|p_T^{\text{reco}} - p_T^{\text{true}}| < 0.5$, obtaining $\sigma_1 = 0.153$, $\sigma_2 = 0.15$. Figure 14 shows the p_T performance for both weight choices. As the ratio σ_1/σ_2 increases, the Energy Method's relative weight grows, enhancing its high-p_T advantage and illustrating the impact of resolution choice on the combined result.

To optimize the fusion, we studied $\sigma_i(p_T, \eta)$ and generated two-dimensional performance maps over $(p_T^{\text{true}}, |\eta|)$, exploiting η-symmetry (Figure 15). Within each (p_T, η) bin, we fitted the reconstructed-p_T distribution to extract the peak shift (bias) and peak width (resolution). These populate four maps: position-method bias, position-method resolution, energy-method bias, and energy-method resolution (Figure 16).

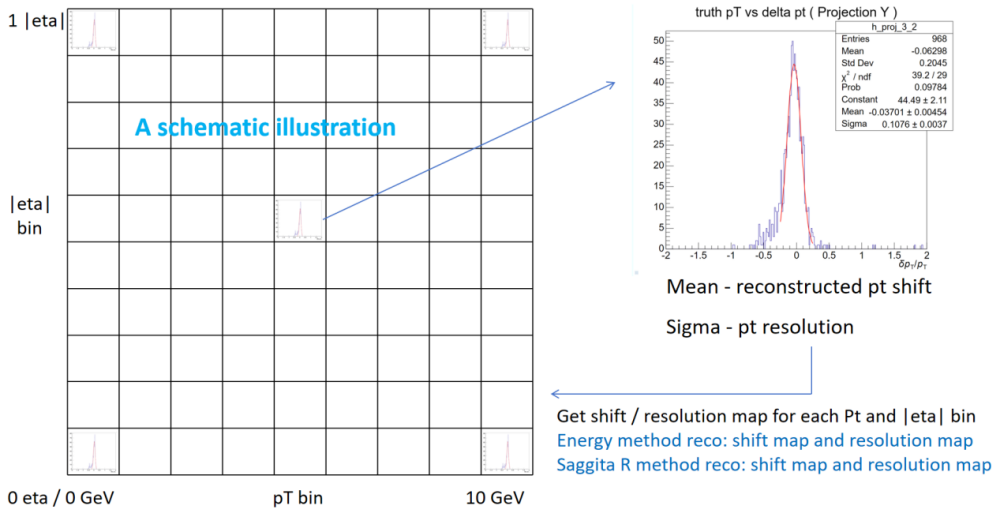


Figure 15: Map principle

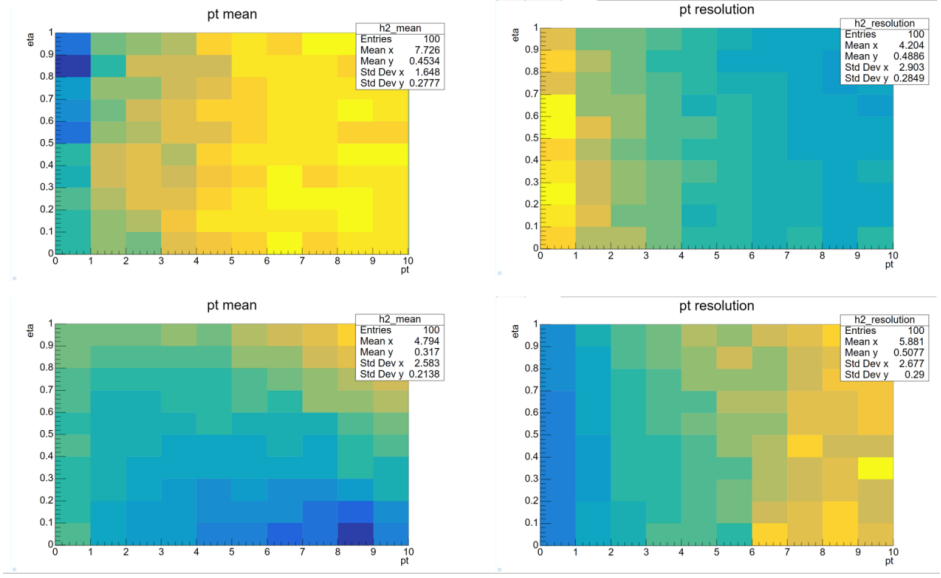


Figure 16: 4 map

For each electron event:

1. Determine $p_{T,1}$, $p_{T,2}$, and η.
2. Correct each p_T using the corresponding shift map.
3. Retrieve the local σ_i from the resolution map.
4. Combine the corrected p_T values with weights $1/\sigma_i^2$.

Then, using the reconstructed p_T and the true p_T, I calculate the relative resolution to evaluate the performance.

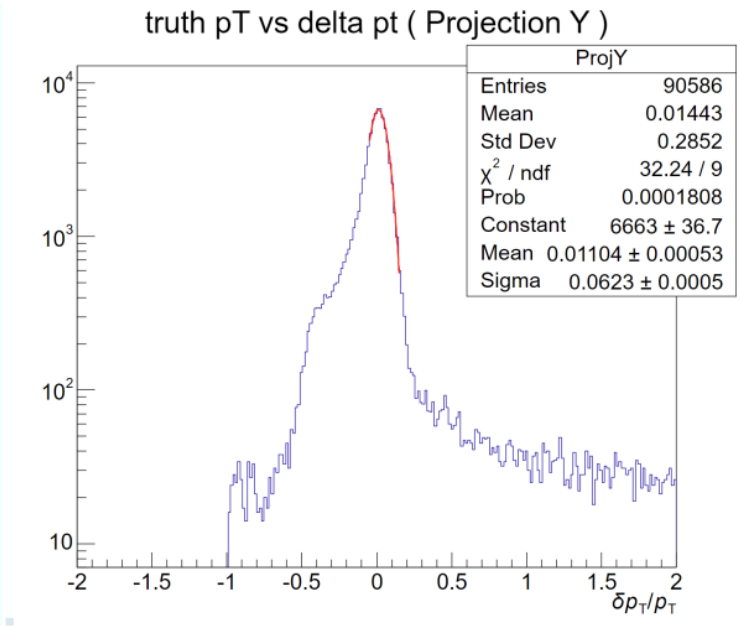
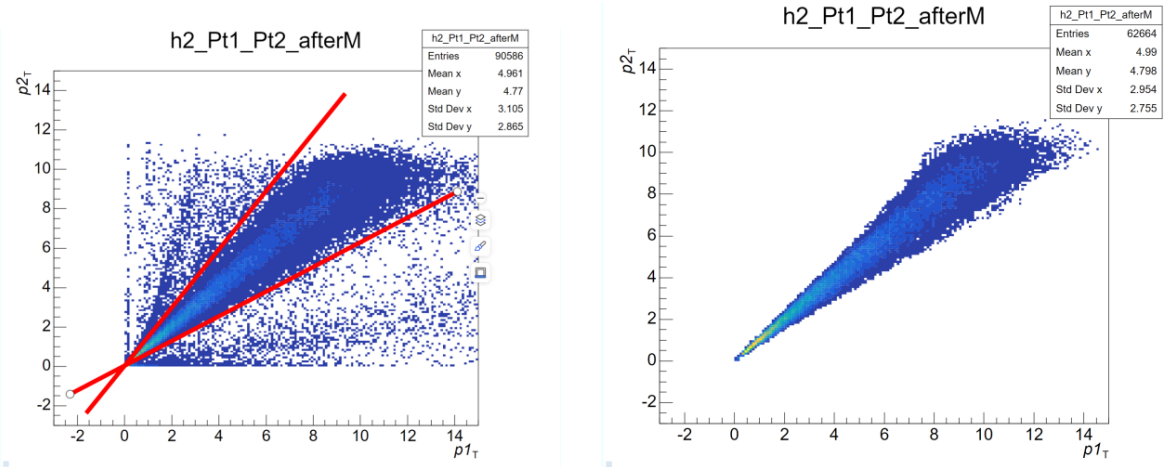


Figure 17: Enter Caption

As shown in Figure 17, the combined p_T reconstruction from both methods yields better performance—narrower width and smaller peak shift. Moreover, the peak of the p_T resolution shows no significant dependence on p_T, and neither the peak position nor the resolution exhibits a clear dependence on η, demonstrating that we have successfully merged the outputs of the two reconstruction methods.

Select on reconstruction



(a) Cut on $E\sin\theta - qBR$

(b) $E\sin\theta - qBR$ after select

Figure 18: select be using two method reconstruct results

We further apply a rejection criterion—"Reject the event if $\frac{p_{T1}}{p_{T2}} > 1.2$ or $\frac{p_{T1}}{p_{T2}} < 0.8$."—to

discard events where one method fails catastrophically, as shown in Figure 18a, After this selection, Figure 18b shows a much cleaner performance, at the cost of approximately one-third efficiency.

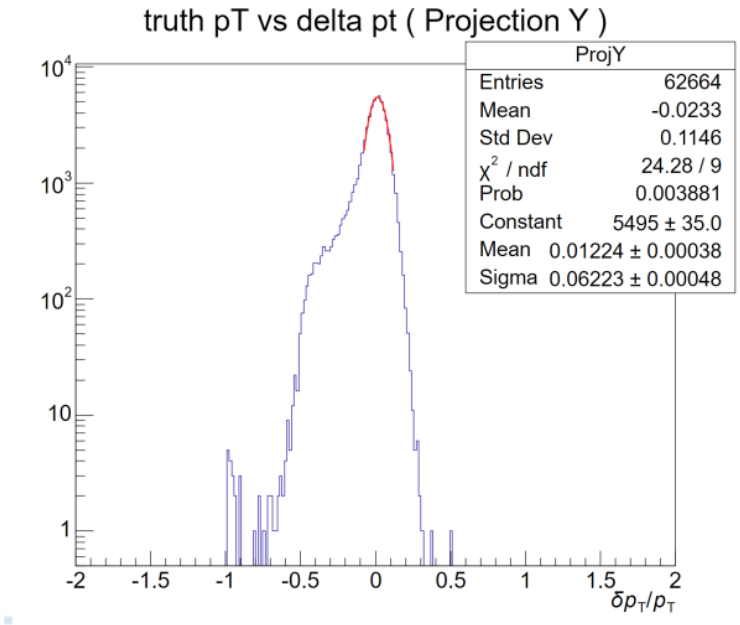
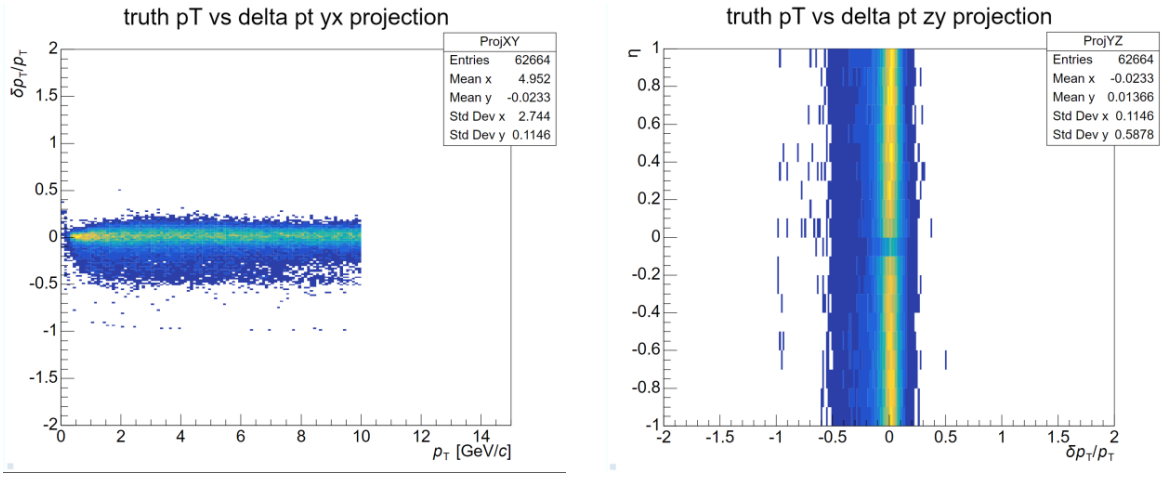


Figure 19: pt resolution with select

Compared to the original sagitta-only reconstruction, the combined method achieves smaller width, smaller peak shift, and reduced standard deviation, though with some efficiency loss and mild asymmetry in the p_T resolution distribution. Additionally, the Energy Method provides an approach to Si–Calorimeter matching for event reconstruction.



(a) pt resolution - truth pt

(b) resolution - eta

Figure 20: combined two method with select performance

Machine Learning for Reconstruction

In our study of charged-track reconstruction using INTT/Silicon and the calorimeter, the detectors provide multidimensional information, including three-dimensional hit positions and calorimeter energy. When I have collected a sufficient dataset, I explored using machine learning to predict p_T.

Data Generation and Split

Within the sPHENIX Fun4All framework, single-electron events 1 M were simulated with vertex $(0, 0, 0)$, $p_T \in [0, 10]$, GeV, full azimuth (2π), and $|\eta| < 1.1$.

- 0.5 M events for training and validation
- 0.5 M independent events for testing performance evaluation

Feature Sets

As shown in Figure 21, the reconstruction uses the three-dimensional coordinates at the 2/5 positions provided by the INTT/Silicon detectors, the three-dimensional coordinates of the reconstructed hit from the EMCal, and the particle's energy, which can be combined as input features. Ground Truth Y is the true transverse momentum (p_T). A data-scaling process is then applied to handle the different types of feature information, such as spatial coordinates and energy.

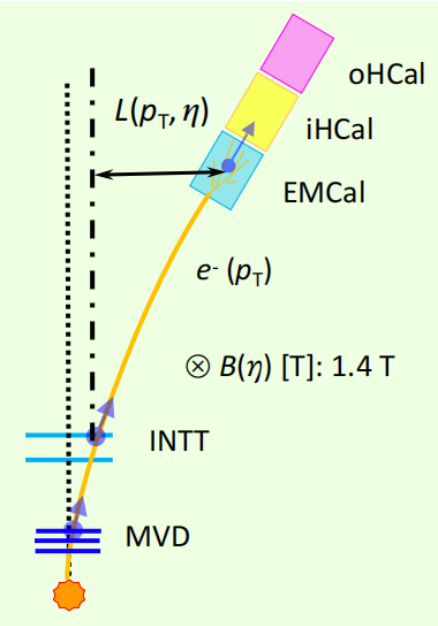


Figure 21: subdetector informantion

Different detector subsystems impose distinct constraints on the p_T reconstruction process, so during model training we introduce additional penalty terms to achieve accurate

results; I therefore first split the detector information into two categories and train them separately before combining them with an MLP. One set of features uses the radius and z components of the position information plus the EMCal energy, and the other describes the degree of deflection in the magnetic field - $[1/\Delta\Phi]$, where $\Delta\Phi$ is the angle between the EMCal–outer INTT line and the outer INTT–inner INTT line, as illustrated in Figure 22.

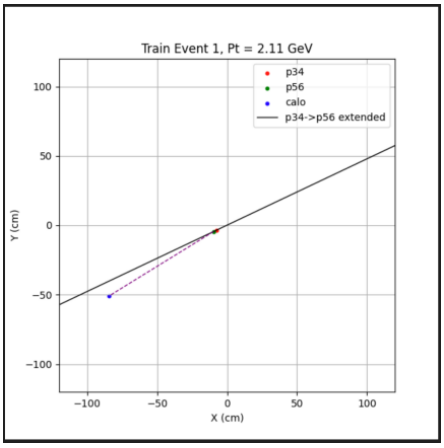


Figure 22: Schematic illustration of the definition of $\Delta\phi$.

Set 1: $[R_{\text{INTT}},\, Z_{\text{INTT}},\, R_{\text{Calo}},\, Z_{\text{Calo}},\, E_{\text{Calo}}]$

Set 2: $[1/\Delta\Phi]$, where $\Delta\Phi$ is the angle between the EMCal–outerINTT and outerINTT–innerINTT lines in the xy plane.

Model A: $[R_{\text{INTT}},\, Z_{\text{INTT}},\, R_{\text{Calo}},\, Z_{\text{Calo}},\, E_{\text{Calo}}] \to p_T$

- **Input:** $\mathbf{X} = [R_{\text{iINTT}}, Z_{\text{iINTT}}, R_{\text{oINTT}}, Z_{\text{oINTT}}, R_{\text{EMCal}}, Z_{\text{EMCal}}, E_{\text{EMCal}}]$
- **Label:** $\mathbf{Y} = [p_T^{\text{true}}]$
- **Network:** MLP with 3 hidden layers of 256 neurons each
- **Hyperparameters:** batch_size=1024, epochs=200, lr=5e-5, val_ratio=0.3
- **Loss:**

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} w_i\, \text{pt_reso}_i^2, \quad \text{pt_reso}_i = \frac{p_{T,i}^{\text{truth}} - \hat{p}_{T,i}}{p_{T,i}^{\text{truth}} + 10^{-6}}, \quad w_i = \begin{cases} 3, & \text{if } |\text{pt_reso}_i| < 0.2, \\ 1, & \text{otherwise.} \end{cases}$$

Input features $X = [R_{\text{iINTT}},\, Z_{\text{iINTT}},\, R_{\text{oINTT}},\, Z_{\text{oINTT}},\, R_{\text{EMCal}},\, Z_{\text{EMCal}},\, E_{\text{EMCal}}]$ and ground truth $Y = [p_T^{\text{truth}}]$. The model is a three-layer fully connected network with 256 neurons per layer. The training hyperparameters are set to batch_size $= 1024$, epochs $= 200$, lr $= 5 \times 10^{-5}$, and val_ratio $= 0.3$. During each epoch, the dataset is shuffled and split

into a 70% training set and a 30% validation set. The training set is fed into the model in batches of size 1024 to produce predictions $\hat{Y}$. We define the relative resolution as

$$\text{pt_reso} = \frac{Y - \hat{Y}}{Y + 10^{-6}},$$

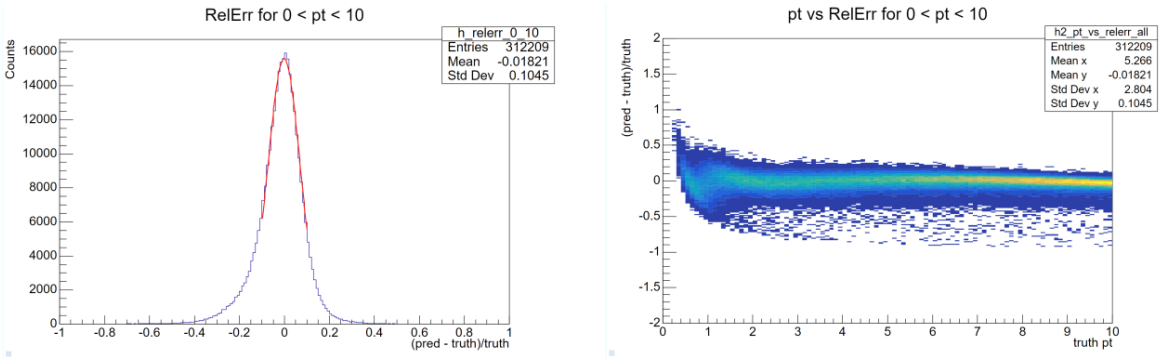
where 10^{-6} prevents division by zero. Samples with $|\text{pt_reso}| < 0.2$ are given additional weight via

$$w = 2 \cdot \mathbf{1}(|\text{pt_reso}| < 0.2) + 1.$$

The loss function is

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} w_i \, \text{pt_reso}_i^2,$$

which uses the squared weighted resolution to focus on minimizing errors near the peak. After each epoch, the model is evaluated on the validation set and the best-performing weights are saved. Finally, the trained model is applied to the test set to obtain reconstructed p_T predictions and compute the resolution.



(a) model A: pt resolution

(b) model A: pt resolution vs pt

Figure 23: Model A performance on electron reco

Model B: $[1/\Delta\Phi] \to p_T$

- **Input:** $\mathbf{X} = [1/\Delta\Phi]$
- **Label:** $\mathbf{Y} = [p_T^{\text{true}}]$
- **Network:** MLP with 3 hidden layers of 256 neurons each
- **Hyperparameters:** batch_size=1024, epochs=500, lr=5e-5, val_ratio=0.3
- **Loss:**

$$\mathcal{L} = \mathcal{L}_{\text{main}} + 0.3\,\mathcal{L}_{\text{mono}} + \min(0.005\,\text{epoch}, 0.2)\,\mathcal{L}_{\text{boundary}}.$$

A line connecting the EMCal position and the outer INTT position in the xy plane and a line connecting the outer INTT position and the inner INTT position in the xy plane define an angle $\Delta\Phi$, which characterizes the degree of deflection of the electron in the magnetic field. The training input is

$$X = \big[1/\Delta\Phi\big],$$

and the ground truth is

$$Y = \big[p_T^{\text{truth}}\big].$$

The hyperparameters are set to batch_size = 1024, epochs = 500, lr = 5×10^{-5}, and val_ratio = 0.3. The model is trained for 500 epochs; in each epoch, the dataset is shuffled and split into 70% training and 30% validation sets, and the training set is fed into the model in batches of size 1024 to produce predictions $\hat{Y}$.

The loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \lambda_{\text{mono}}\,\mathcal{L}_{\text{mono}} + \lambda_{\text{boundary}}\,\mathcal{L}_{\text{boundary}},$$

where

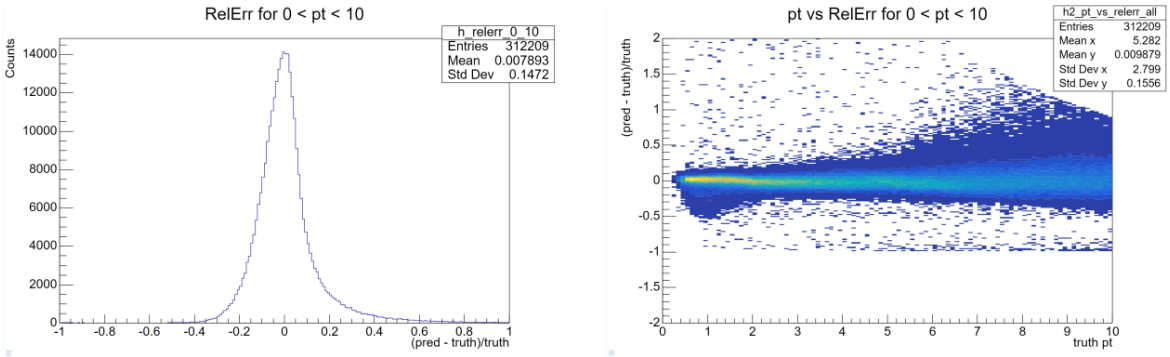
$$\mathcal{L}_{\text{main}} = \frac{1}{N}\sum_{i=1}^{N} w_i\,\text{pt_reso}_i^2, \quad \text{pt_reso}_i = \frac{p_{T,i}^{\text{truth}} - \hat{p}_{T,i}}{p_{T,i}^{\text{truth}} + 10^{-6}}, \quad w_i = \begin{cases} 3, & \text{if } |\text{pt_reso}_i| < 0.2, \\ 1, & \text{otherwise.} \end{cases}$$

$\lambda_{\text{mono}} = 0.3$, and $\mathcal{L}_{\text{mono}}$ enforces that p_T decreases as $\Delta\Phi$ increases, while $\mathcal{L}_{\text{boundary}}$ anchors the model with boundary conditions outside the data range using points

$$\{(0.5, 0.0961), (1, 0.1922), (2, 0.3844), (10, 1.922), (15, 2.883), (25, 4.805), (50, 9.61), (100, 19.22), (200, 38.44)\}$$

and a dynamic weight

$$\lambda_{\text{boundary}} = \min\big(0.005 \times \text{epoch},\ 0.2\big).$$



(a) model B: pt resolution (b) model B: pt resolution vs pt

Figure 24: Model B performance on electron reco

Fusion MLP $\rightarrow p_T$

Comparison of the two models shows that the p_T reconstruction preserves the characteristics of the detector information. Model A, based on $[R_{\mathrm{INTT}}, Z_{\mathrm{INTT}}, R_{\mathrm{Calo}}, Z_{\mathrm{Calo}}, E_{\mathrm{Calo}}] \rightarrow p_T$, retains the EMCal's energy-resolution characteristic—showing better relative resolution at high p_T. Model B, using $[1/\Delta\Phi] \rightarrow p_T$, excels at low p_T, where larger bending angles reduce the impact of position resolution. As p_T increases and $\Delta\Phi$ shrinks, the influence of position resolution grows, which is reflected in the steeper slope of the mapping function at high p_T. When these two models are fused with an MLP, the combined predictor preserves both high-p_T energy-resolution performance and low-p_T deflection sensitivity, yielding a robust reconstruction across the full momentum range.

Once both submodels are trained, we merge them using a final MLP. The input feature vector is

$$[\Delta\Phi_i,\ p_{T,1,i},\ E_{\mathrm{calo},i},\ p_{T,2,i}] + \mathrm{pt_onehot},$$

where $\Delta\Phi_i$ is the angle between the EMCal–outer INTT and outer–inner INTT lines for event i; $p_{T,1,i}$ is the prediction from the $[1/\Delta\Phi]$ model; $E_{\mathrm{calo},i}$ is the calorimeter energy; and $p_{T,2,i}$ is the prediction from the $[\mathrm{INTT}_R, Z + \mathrm{Calo}_R, Z, E]$ model. The term pt_onehot denotes a 10-dimensional one-hot encoding of the estimated transverse momentum: we compute $p_T^{\mathrm{est}} = 0.5\,(p_{T,1,i} + p_{T,2,i})$ and then set the element corresponding to the integer bin $\lfloor p_T^{\mathrm{est}} \rfloor$ (with edges at 0, 1, …, 10) to 1, all others to 0.

Model Fusion:

- **Input:** $\mathbf{X} = [\,\Delta\Phi,\ p_{T,1},\ E_{\mathrm{calo}},\ p_{T,2}\,] + \mathrm{pt_onehot}$
- **Label:** $\mathbf{Y} = [\,p_T^{\mathrm{true}}\,]$
- **Network:** MLP with 3 hidden layers of 128 neurons each
- **Hyperparameters:** $\mathrm{batch_size} = 1024,\ \mathrm{epochs} = 300,\ \mathrm{lr} = 5 \times 10^{-5},\ \mathrm{val_ratio} = 0.3$
- **Loss:**

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} w_i\,\mathrm{pt_reso}_i^2, \quad \mathrm{pt_reso}_i = \frac{p_{T,i}^{\mathrm{truth}} - \hat{p}_{T,i}}{p_{T,i}^{\mathrm{truth}} + 10^{-6}}, \quad w_i = \begin{cases} 3, & \text{if } |\mathrm{pt_reso}_i| < 0.2, \\ 1, & \text{otherwise.} \end{cases}$$

.

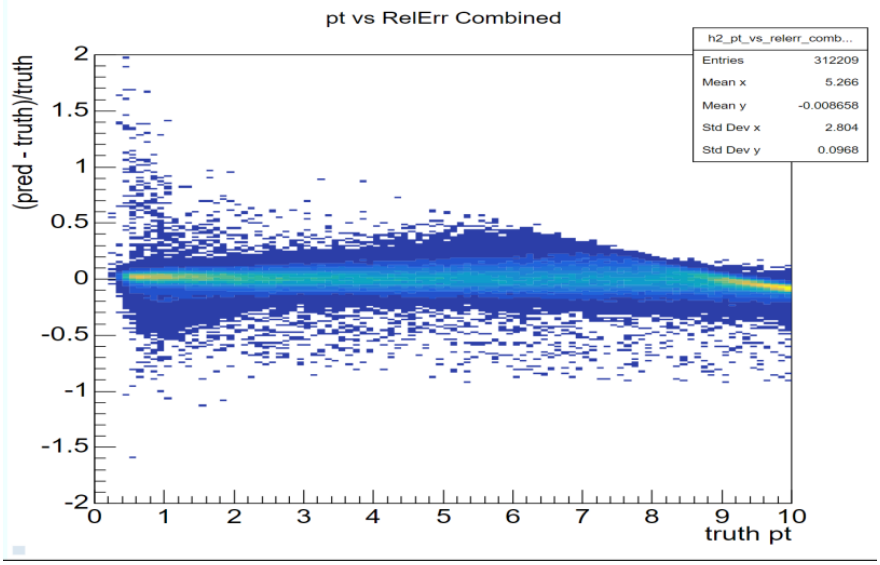


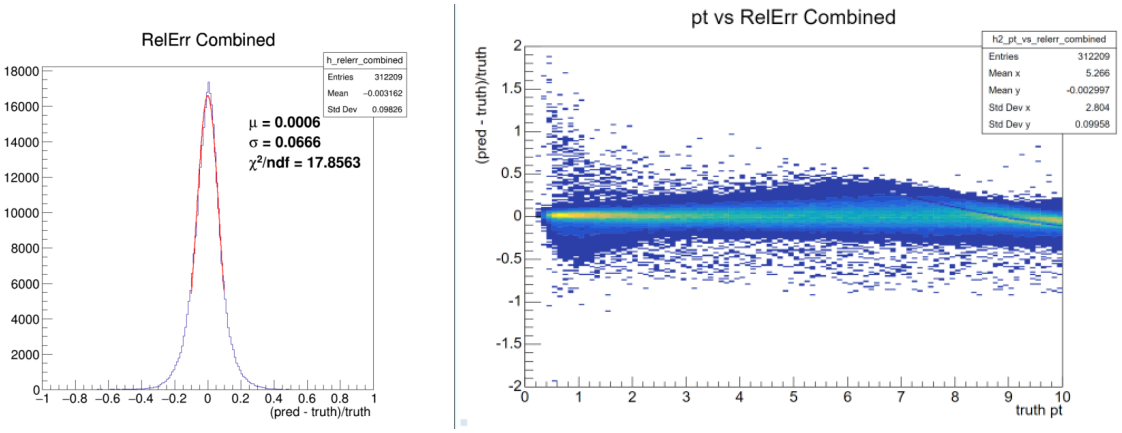
Figure 25: Fusion performance before post-correction

Post-Correction and Performance

Figure 25 shows the electron reconstruction performance of the fused model. A boundary effect appears near the high-p_T edge due to the lack of higher-p_T truth data in the simulation, leading to underestimation. This can be mitigated by expanding the dataset's truth-p_T range or by applying a post-correction.

We introduce a simple post-correction for reconstructed $p_T > 8.8$ GeV:

$$\text{corr_factor} = 0.02 + 0.08\,(p_{T,\text{reco}} - 8.8), \qquad p_{T,\text{pred}} = p_{T,\text{reco}}\,(1 + \text{corr_factor}).$$



(a) model Fusion: pt resolution (b) model Fusion: pt resolution vs pt

Figure 26: Model Fusion performance on electron reco

The corrected performance is shown in Figure 26. Compared to formula-based reconstruction, the machine-learning approach with post-correction achieves:

- **Better efficiency:** only events with both inner–outer INTT and calorimeter hits (cluster energy > 0.5 GeV) are reconstructed, improving hit selection yield.
- **Reduced bias:** the mean value and Gaussian peak of $(p_{T,\mathrm{pred}} - p_T^{\mathrm{true}})/p_T^{\mathrm{true}}$ are closer to zero.
- **Improved resolution:** the residual distribution is more symmetric, with the narrowest width and smallest standard deviation.