Generalised Geometric Track Finding Algorithm for IDEA

Andrea De Vita, Dolores Garcia, Brieuc Francois

FUTURE CIRCULAR COLLIDER

CERN

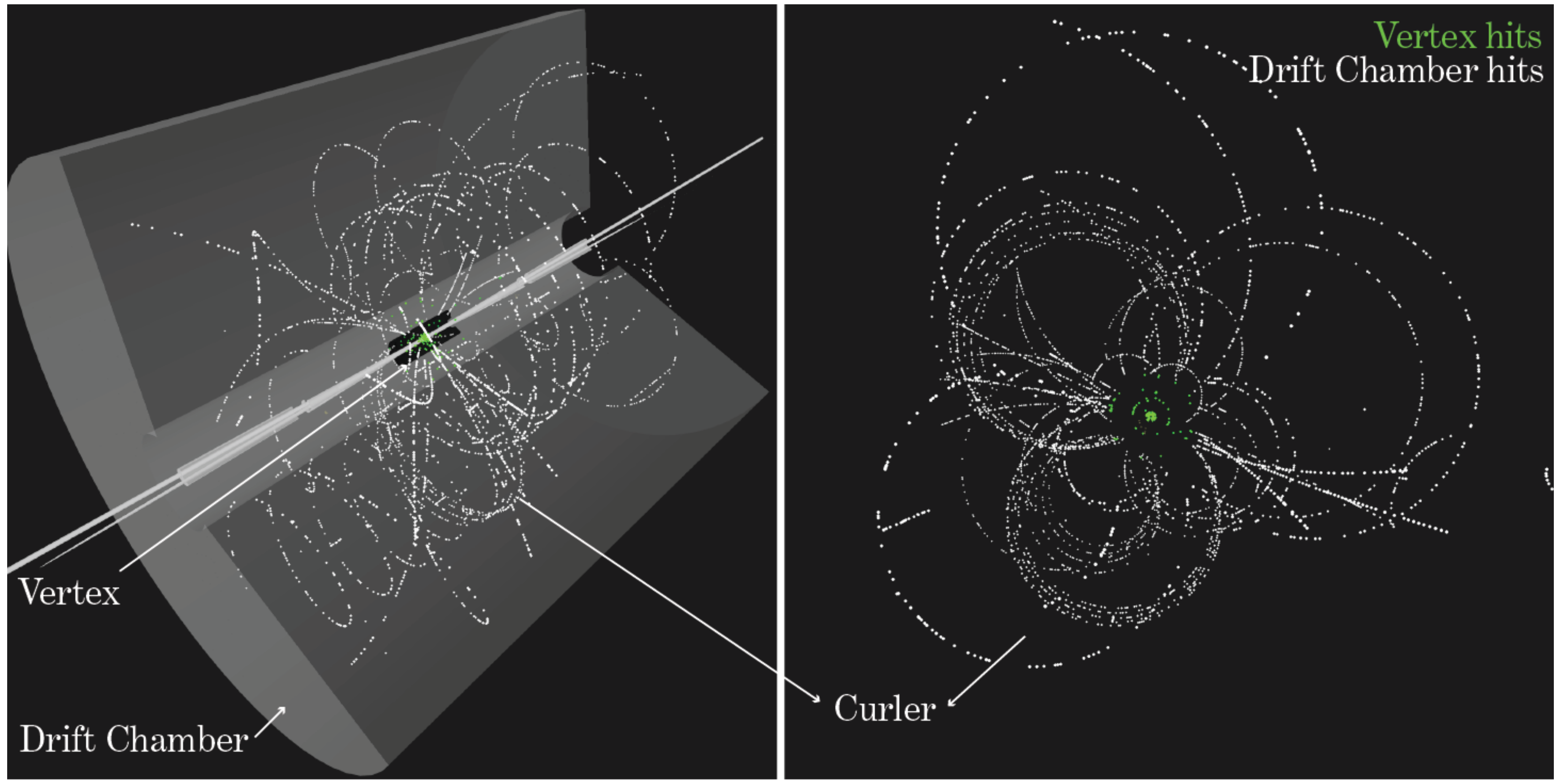
2nd FCC Italy&France Workshop
4-6 November 2024

Introduction

The tracking algorithm is typically tailored to the specific sub-detectors in use. This study, however, introduces a **generalized geometric method for track finding**.

An end-to-end pipeline can be employed to process hits from all tracking systems, producing a complete set of tracks and **evaluating tracking performance**.

The analyses focus on the IDEA detector, which utilises a **vertex detector and a drift chamber**.



Complete Pipeline

Idea detector simulation
(IDEA_o1_v02.xml)

+

Digitizer v01
(moving to Digitizer v02)

↓

Generalised geometric
track finding algorithm

+

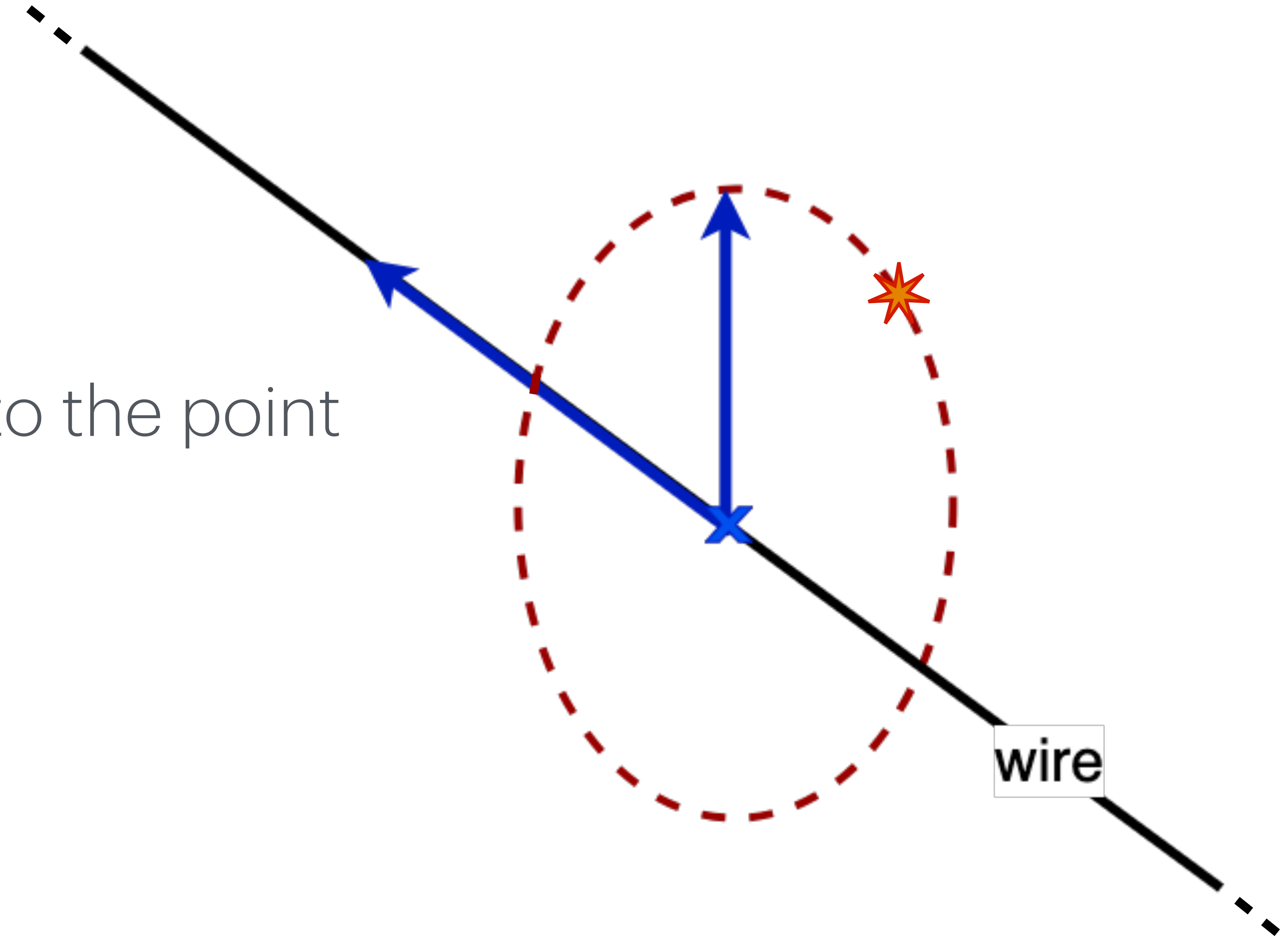
Evaluation Step
(tracking efficiency)

From Simulation to Digitizer

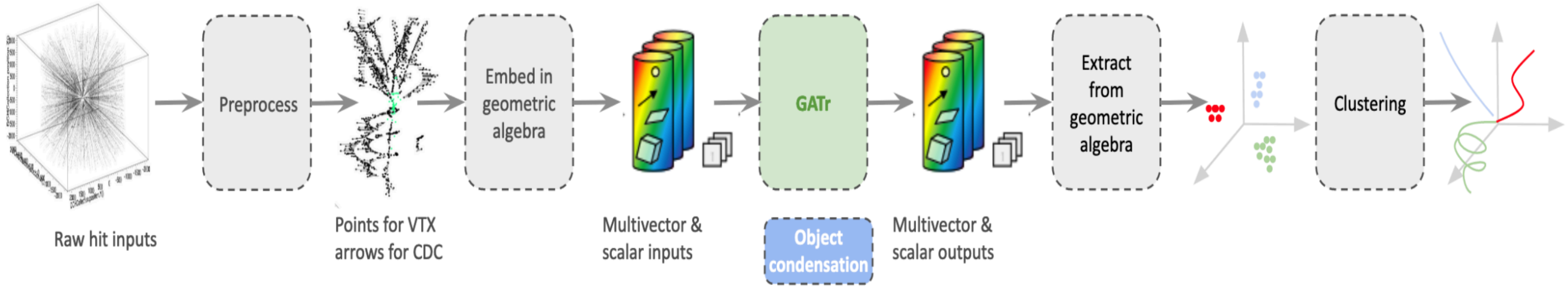
Digitizer v01 vs Digitizer v02

Drift chamber hits exhibit a **positional ambiguity**, as it is necessary to consider all possible positions along a circle with its center coinciding with the wire.

- Digitizer v01 describes the drift chamber hits through two positions, **left** and **right** of the wire.
- Digitizer v02 uses three quantities:
    1. the **position of the point on the wire** closest to the point where the particle passed through;
    2. **radius** of the circle;
    3. the **direction of the wire**;



Generalised Geometric Track Finding algorithm



ML step: The Track-finding approach is based on a graph structure of the inputs, where geometric algebra transformations are applied. The result is a set of pairs (β, coordinates) in the embedding space.

Clustering step: Tracks can be identified in the embedding space by applying a clustering algorithm, establishing a one-to-one correspondence between clusters and tracks.

More at D. Garcia CHEP 2024 talk

Track Finder - ML step

1 **Preprocessing** → Each set of hits $X = \{X_{vxt}, X_{cdh}\}$ has to be converted into a **graph of multivectors**, where each node corresponds to an hit.

2 **Geometric Algebra Transformer (GATr)** → Each multivector graph is transformed with **GATr layers**.

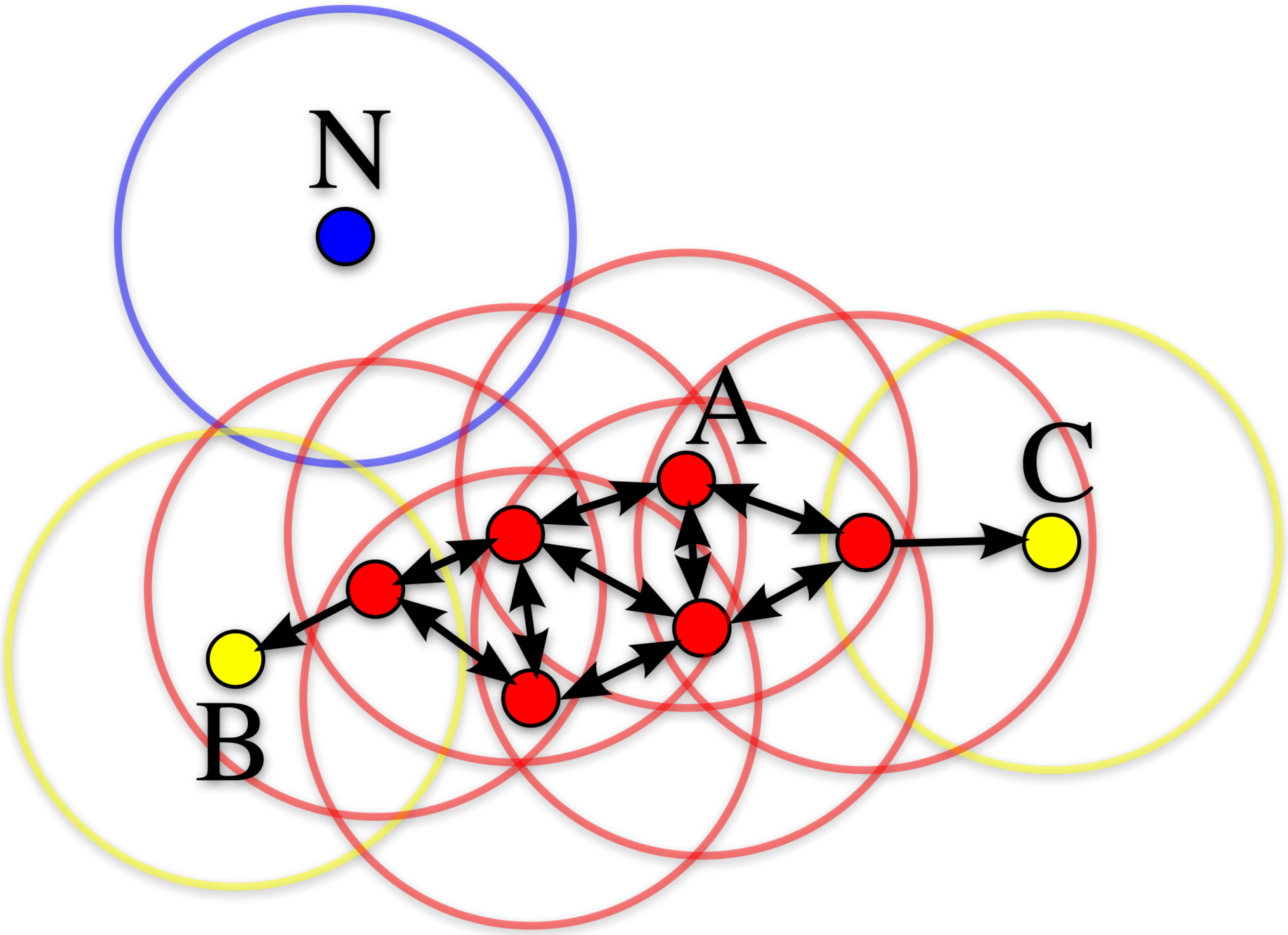
3 **Object Condensation** → Each node of the transformed graph is mapped into an **embedding space coordinate with a corresponding scalar (β)**.

Track Finder - Clustering

The clustering algorithm is implemented with **DBSCAN**.

DBSCAN uses a definition of clusters based on the notion of **density**:

- if a point has a minimum number of points within a certain epsilon distance (ϵ), it is classified as a **core point**;
- If a point is not a core point and it is not close to a core point, then it is classified as **noise**;



Starting with the core points, the clusters are expanded until all points are classified as noise or belonging to a cluster.

Tracking Efficiency

Introduction

It is essential to define **tracking efficiency** in a way that aligns with the detector's design.

IDEA relies on a tracker with drift chambers and vertex detectors, so the hit collection is predominantly influenced by **contributions from the drift chambers**.

By analogy of design, the definition proposed by BELLE II is chosen, for which a particle is assigned to a track depending on **purity** and **efficiency** values.

Tracking Efficiency - Step 1

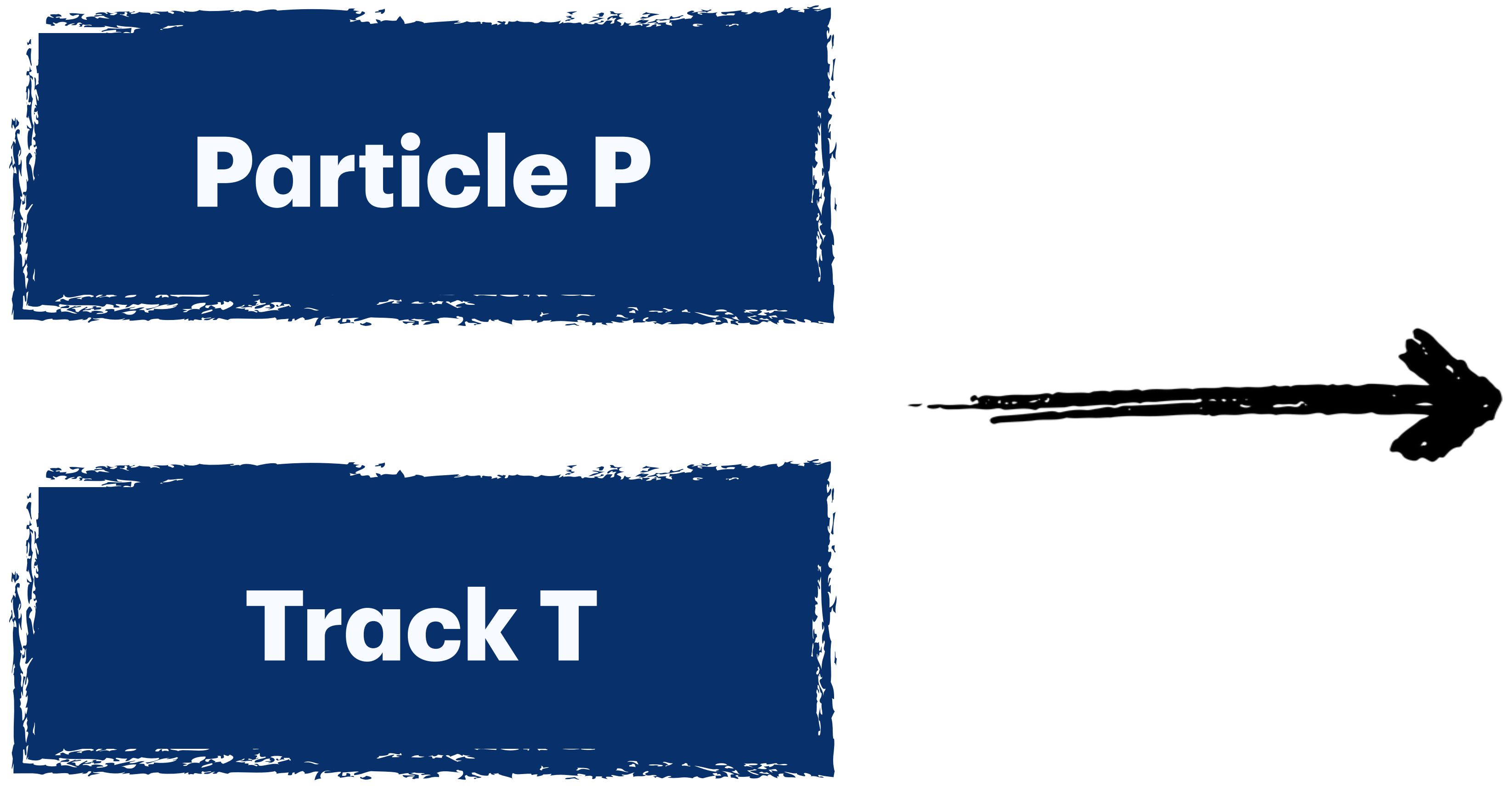
Reconstructable particles

A particle is defined as **reconstructable** if it satisfies the following conditions:

1. $p_T > 100$ MeV
2. $\cos(\theta) < 0.99$
3. Number of unique hits (Drift Chamber + Vertex) > 15
4. Number of Drift Chamber hits > 4
5. Generator Status == 1
6. Vertex < 50 mm

Tracking Efficiency - Step 2

Purity and efficiency - Assigned particles



$$Pur_{MC_p}^{TRACK_t} = \frac{\text{num of hits from } MC_p \text{ in } TRACK_t}{\text{num of hits of } TRACK_t}$$

$$Eff_{TRACK_t}^{MC_p} = \frac{\text{num of hits from } TRACK_t \text{ in } MC_p}{\text{num of hits of } MC_p}$$

A **particle P is assigned to the track T** if $Pur_{MC_p}^{TRACK_t} > 0.5$ and $Eff_{TRACK_t}^{MC_p} > 0.5$

Tracking Efficiency - Step 3

Tracking efficiency and Fake Tracks

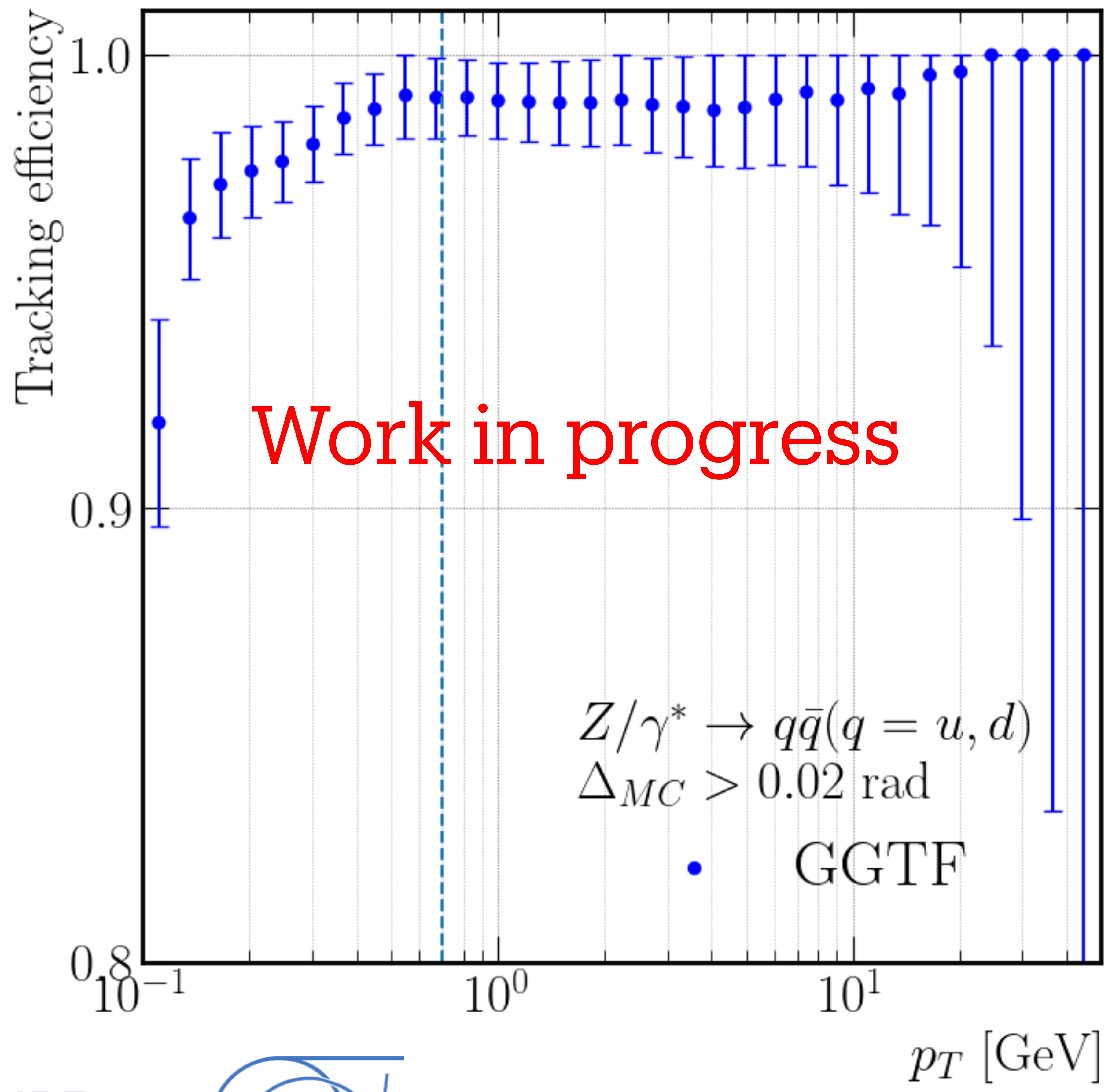
$$\text{Tracking efficiency} = \frac{\text{number of reconstructable and assigned particles}}{\text{number of reconstructable particles}}$$

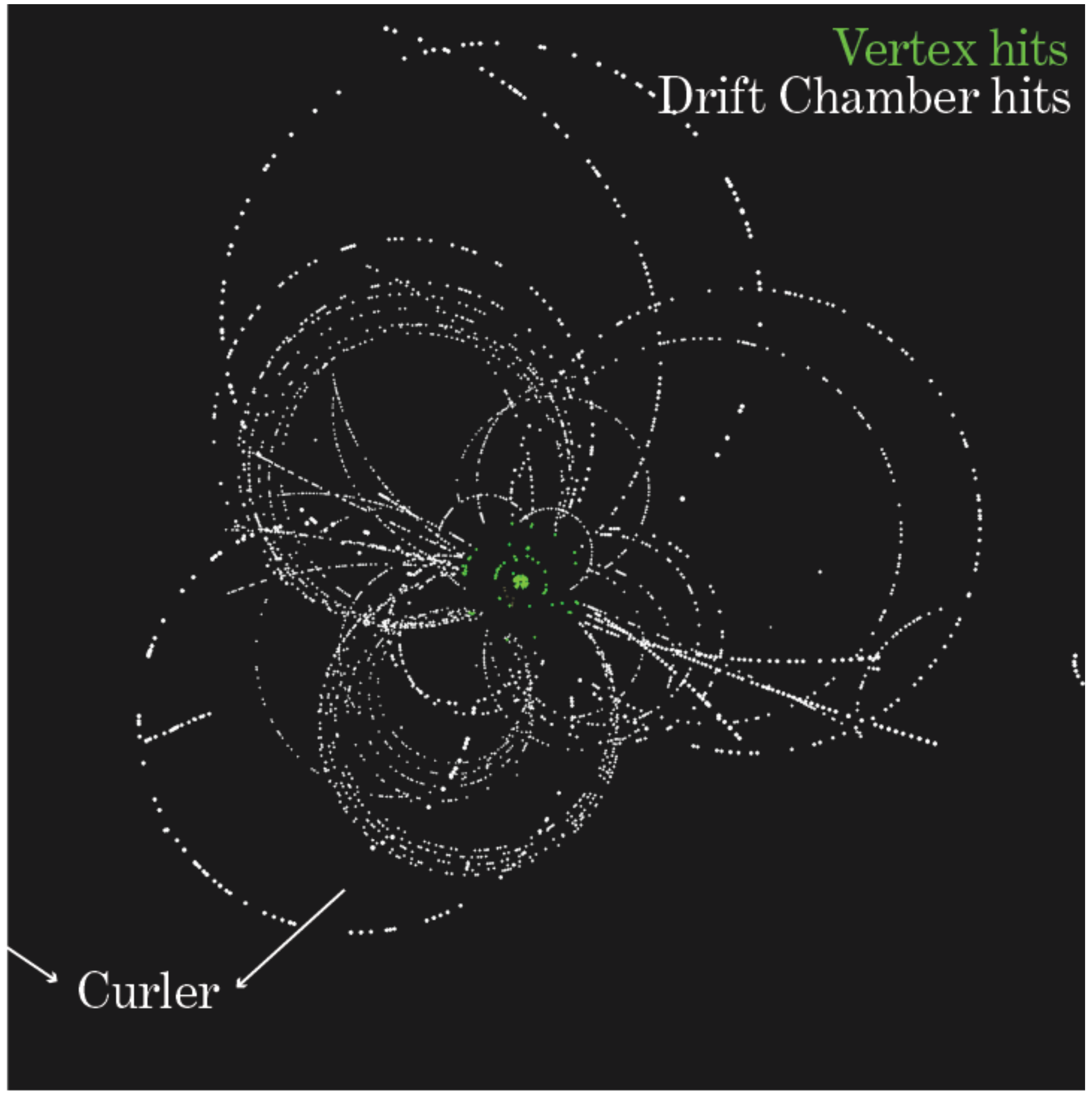
Tracks with no assigned particles are considered **fake tracks**.

Particles that are assigned but are not reconstructable may indicate that it is possible to relax the definition of “reconstructable”.

Results

Performance for complex events - $Z/\gamma^* \rightarrow q\bar{q}\ (q = u, d)$





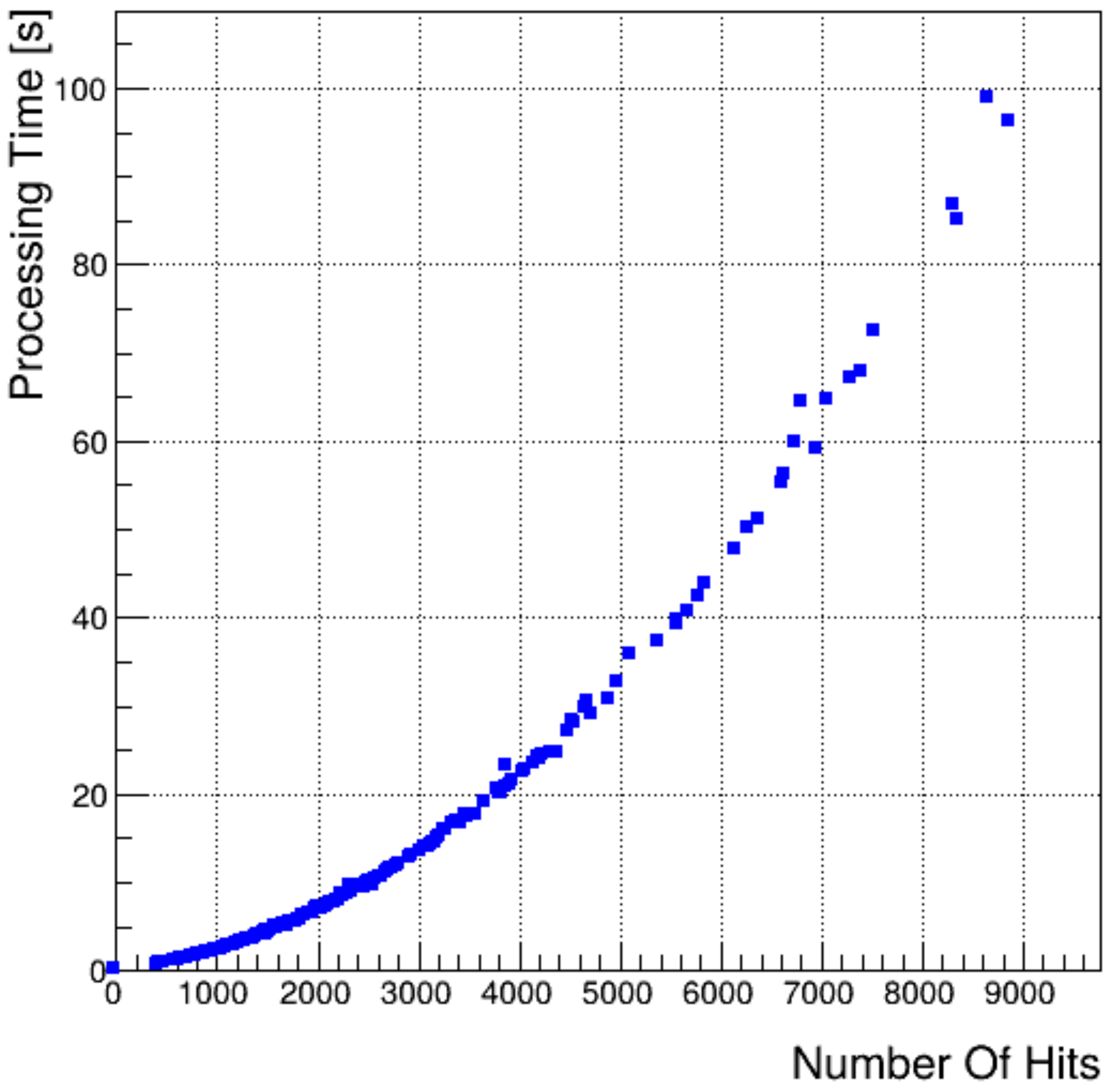
What's next

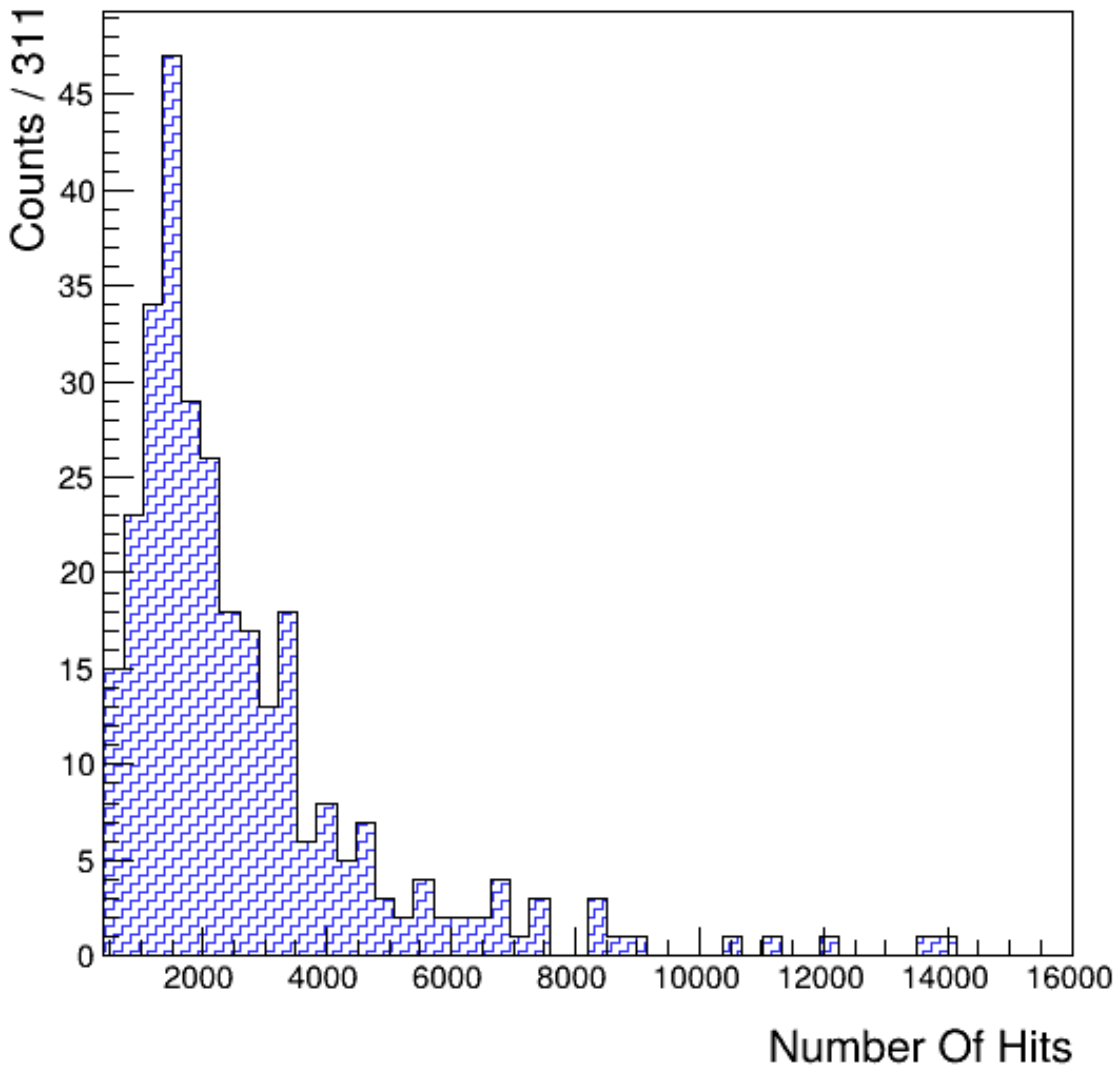
- **Optimization of the GGTF architecture** aims to enhance performance while reducing computational load.
- A pull request (PR) will be submitted to the central Key4Hep repository, implementing a **complete end-to-end pipeline for the GGTF**.
- Additionally, developing a **physics-based track-fitting** algorithm will be valuable as both an alternative and complementary tool to the ML-based track-finding algorithm.

Backup

Results

Track finder - processing time





Performance for complex events CLD: tracking efficiency

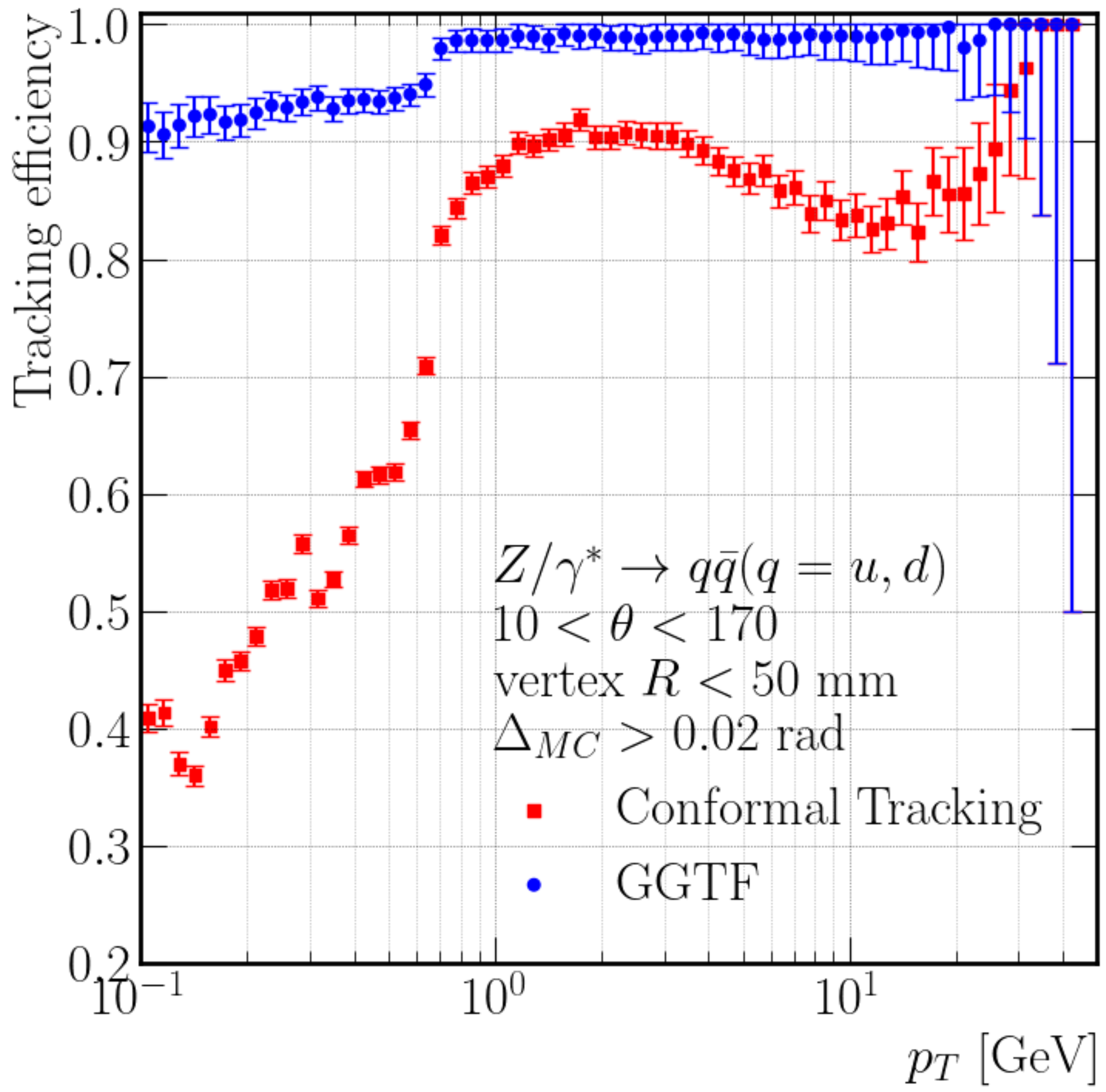


Table of Features

Structure

```
Event: 49
     PDG        pt  costheta       phi  ...  assigned_track  isReconstructable  isAssigned  isRecoAndAssigned
0     11  0.000000  1.000000  0.000000  ...               0                  0           0                  0
1     11  0.000000  1.000000  0.000000  ...               0                  0           0                  0
2     22  0.000000  1.000000  0.000000  ...               0                  0           0                  0
3    -11  0.000000  1.000000  0.000000  ...               0                  0           0                  0
4    -11  0.000000  1.000000  0.000000  ...               0                  0           0                  0
..   ...       ...       ...       ...  ...             ...                ...         ...                ...
```

- **Assigned_track**: index of the track which the particle has been assigned to (0 if not assigned)
- **isReconstructable**: boolean value which is 1 if the particle is reconstructable and 0 otherwise
- **isAssigned**: boolean value which is 1 if the particle is assigned and 0 otherwise
- **Purity**: purity of the track which the particle has been assigned to (-1 if not assigned)
- **Efficiency**: efficiency of the particle with respect to the track which the particle has been assigned to (-1 if not assigned)