Tracking Reconstruction in IDEA
+
ML Particle Flow

Andrea De Vita, Dolores Garcia, Brieuc Francois,
Michele Selvaggi, Andreas Loeschcke Centeno, Lena Herrmann

Table Of Contents

Drift Chamber Digitizer

Toward A More Realistic Digitizer

Many thanks to Andreas! Take a look at the PR!

The **DCHdigi_v02** digitiser for the IDEA drift chamber is an intermediate step between **DCHdigi_v01** and a future full waveform digitiser.

Compared to v01:

- It produces **one DigiHit per cell**, including both **drift time** and **signal propagation time** along the wire in the total DigiHit time.
- It performs a **per-cell dN/dx calculation** using the **Delphes parametrisation** (see here).
- It **combines multiple SimHits within the same cell** into a single DigiHit (or multiple DigiHits if they are significantly separated in time).
- The **time and position measurements** of each DigiHit are determined by the **first hit arriving at the readout**.

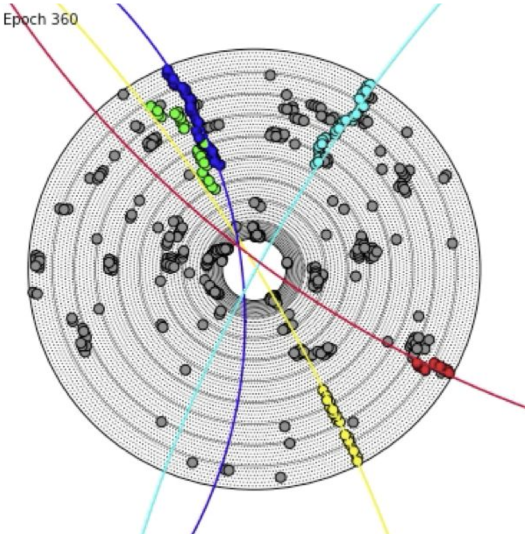
Geometric Graph Neural Network Based Track Finder

From Current Challenges to a Novel Approach

Traditional track finding methods are often complex and detector-specific, limiting their adaptability. To address this, we propose the **Geometric Graph Track Finding (GGTF)** method, an end-to-end detector-agnostic approach.

Key features include:

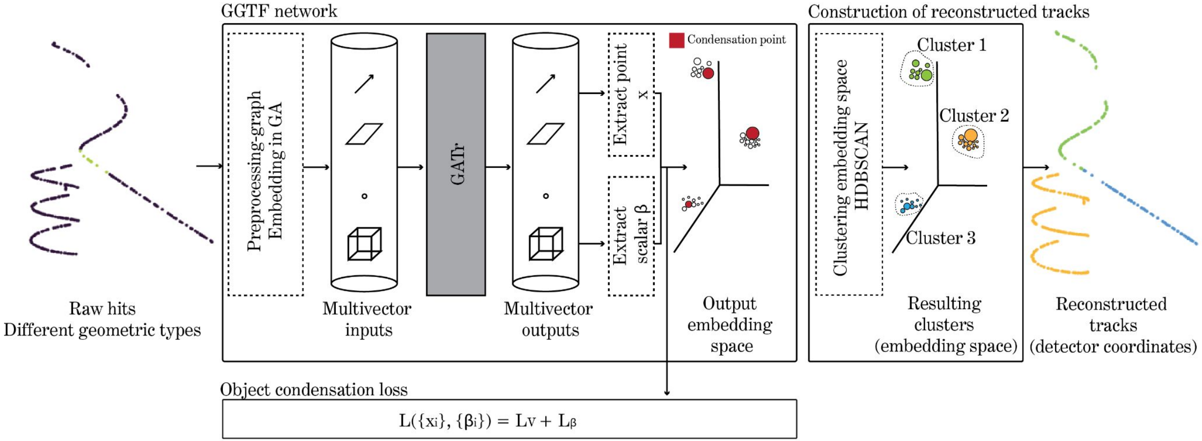
- **Compatibility** with multiple sub-detectors and tracking technologies.
- **Independence from detector geometry** and material specifications.
- **No reliance on analytical trajectory parameterization**.



Insights into the Neural Network Design

Inputs : For each event, a set of hits is collected from various tracking components. The input to the pipeline is the aggregated collection of all hits, denoted as X={X_v,X_i,X_o, …) (vertex, inner tracker, outer tracker, …)

Outputs : For each event, a set of track is returned. Each track is a collection of hits from different tracking components.

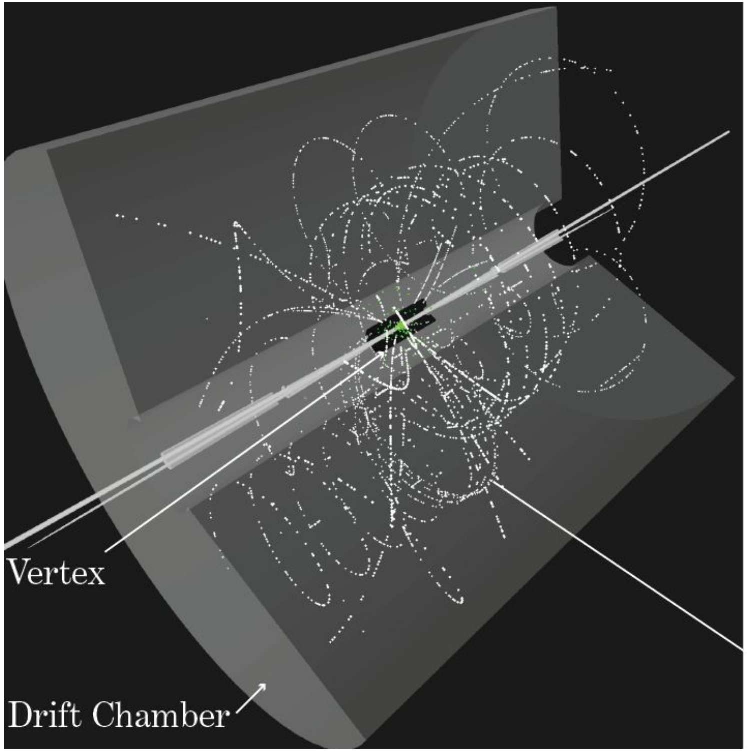


Garcia, D., Francois, B., & Selvaggi, M. (2024). Geometric Graph Neural Network based track finding. CERN.

Tracking Efficiency

Each reconstructed track is matched to a ground truth particle to which it shares the largest number of hits.

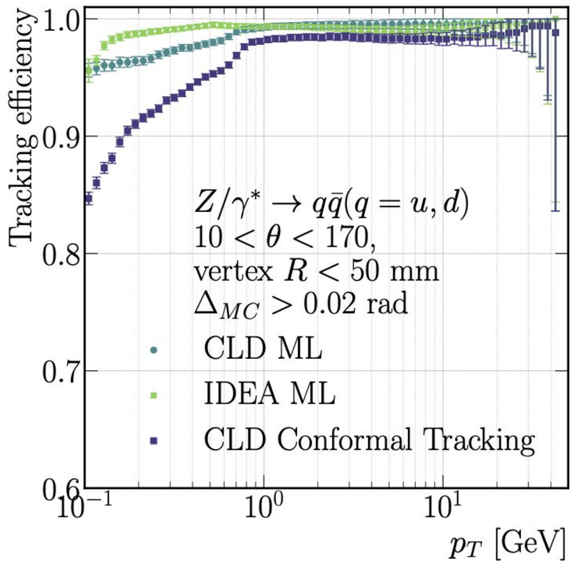
Tracking Efficiency is the probability to reconstruct a track and it can be defined as the percentage of reconstructable charged particles matched to a reconstructed track with at least 4 hits.

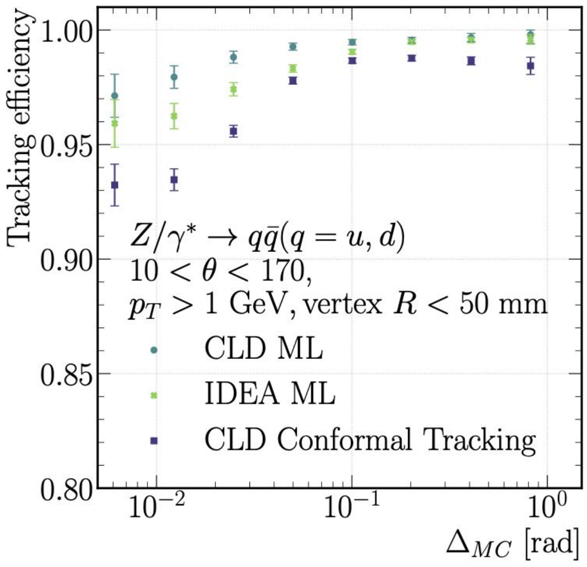


Comparison between CLD and IDEA

Without background, a comparison of CLD and IDEA shows that IDEA achieves better tracking efficiency at low p_T. This is due to the drift chamber design, which allows particles to **traverse a larger portion of the tracker**.

CLD, on the other hand, ensures better identification in the case of **closely spaced tracks**.





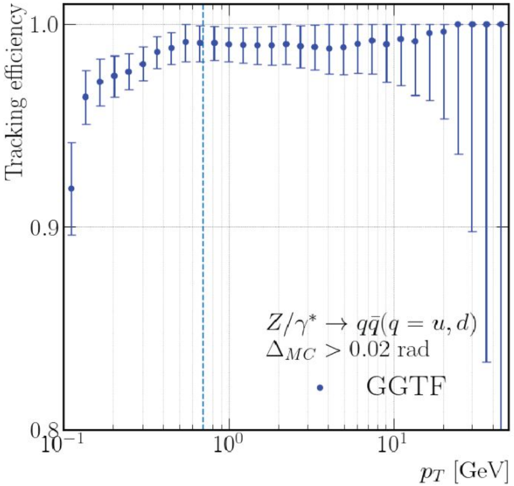
Performance Analysis on IDEA

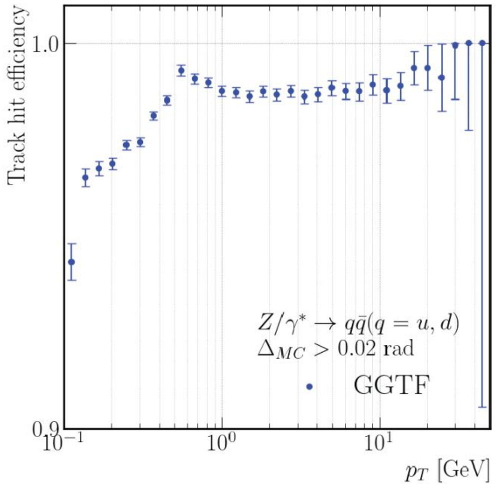
Another Definition of the Tracking Efficiency

Track Hit Purity is the ratio of the number of hits in the track that belong to the MC particle and the total number of hits of the reconstructed track.

Track Hit Efficiency is the ratio of the number of hits in the track that belong to the MC particle and the total number of hits this particle left in the detector.

Tracking Efficiency is the percentage of reconstructable charged particles with both track hit purity and efficiency above 50%.

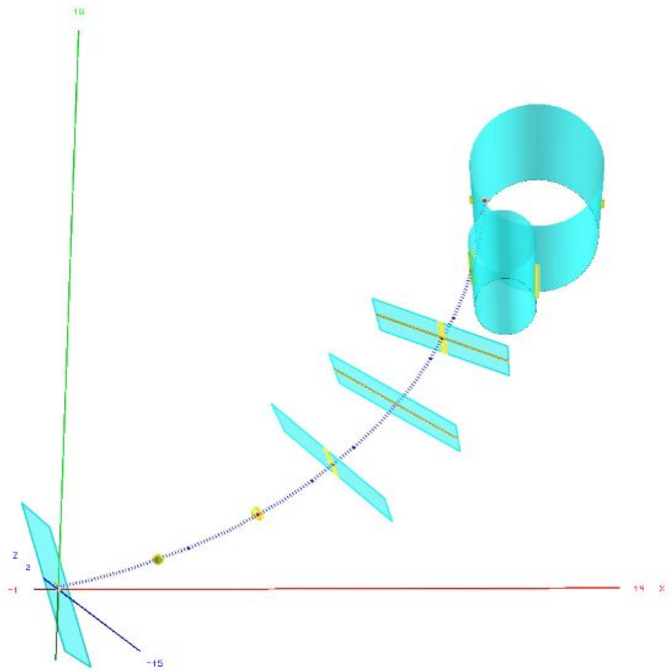




Track fitting using Genfit2

Introduction to Genfit2

The Genfit2 toolkit provides track representation, **track-fitting algorithms** and graphic visualization of tracks and detectors, and it can be used for any experiment that determines parameters of charged particle trajectories from **spatial coordinate measurements**.



Bilka, T., et al. (2019). Implementation of GENFIT2 as an experiment independent track-fitting framework.

Preliminary Steps for Track Fitting

Before performing the fit on a hit collection (`genfit::Track`), several components must be initialized:

- **Define a `TGeoManager`**: It handles the geometry of the detector.
- **Define `genfit::MaterialEffects`**: It manages the effects of energy loss, multiple scattering, and Bremsstrahlung.
- **Define the `genfit::FieldManager`**: It can use a uniform, constant, or map-based field.
- **Define the fitter**: It is responsible for performing the track fit. Different fitters available (e.g., Kalman Filter, DAF, GBL).
- **Define the track representation**: It describes the track parameterization and extrapolation method. Options include:
    - `RKTrackRep`: It uses Runge-Kutta-Nyström method.
    - `HelixTrackRep`: It assumes a uniform magnetic field.

Measurements within the Genfit2 framework

In Genfit2, a track is a collection of hits, where each hit is represented as a measurement. The abstract base class `genfit::AbsMeasurement` defines the interface for all measurement types.

- **Key4hep Hit Types in IDEA**:
  - `extension::SenseWireHit` (Drift Chamber Hits): It represents a circle around the sense wire.
  - `edm4hep::TrackerHitPlane` (Silicon Hits): A planar measurement on a silicon detector.
- **Genfit2 Measurement Classes**:
  - Drift Chamber Hits: `genfit::WirePointMeasurement`
  - Silicon Hits: `genfit::PlanarMeasurement`

These measurement types allow for proper hit encoding within the Genfit2 tracking framework.

IDEA - Vertex and Silicon Wrapper

The IDEA vertex and wrapper are based on silicon detectors. The corresponding `genfit::PlanarMeasurement` is a 2D measurement in local coordinates. To define it, it is necessary to provide the local coordinates of the hit and the position of the detector surface. Moreover, each measurement must store also the covariance matrix of the hit position.

Here are the ingredients to create a `genfit::PlanarMeasurement`:

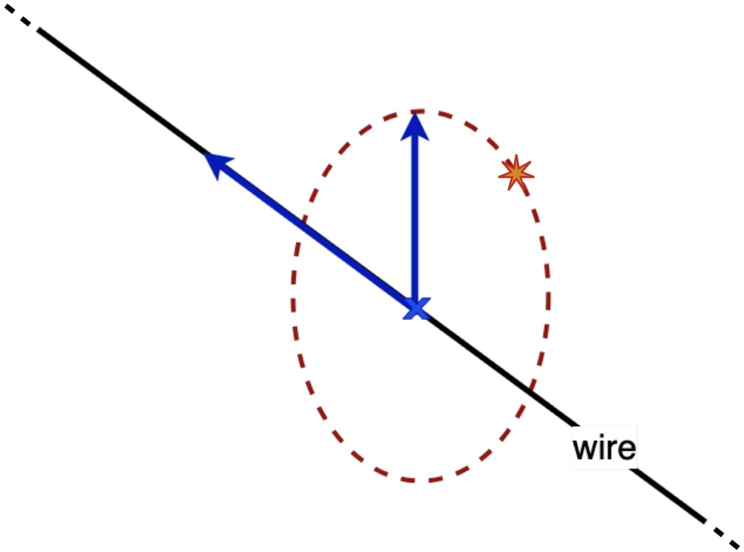
- **Hit index, Plane index, Detector index**
- **Measurement Data (Local Coordinates)**:
  - Hit local coordinates: `TVectorD(2)`
  - Hit local covariance matrix: `TMatrixDSym(2x2)`
- **Plane Definition (Global Coordinates)**:
  - Plane origin: `TVector3`
  - U and V directions: `TVector3`

IDEA - Drift Chamber

IDEA drift chamber provides `extension::SenseWireHit` data. The corresponding `genfit::WirePointMeasurement` stores wire information, drift distance and the position along the wire.

Information stored in `genfit::WirePointMeasurement`:

- **Hit index, Detector index**
- **Wire Definition (Global Coordinates)**:
  - Extremity positions (6 parameters)
- **Measurement Data (Local Coordinates)**:
  - Closest point on sense wire (1 parameter)
  - Drift distance (1 parameter)
- **Uncertainty Representation**:
  - 8×8 covariance matrix

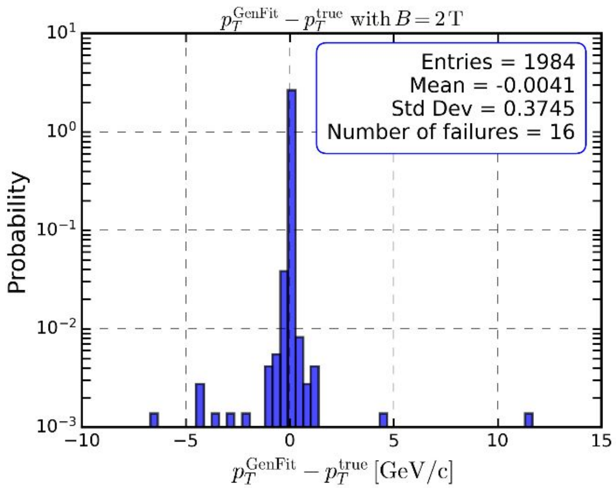


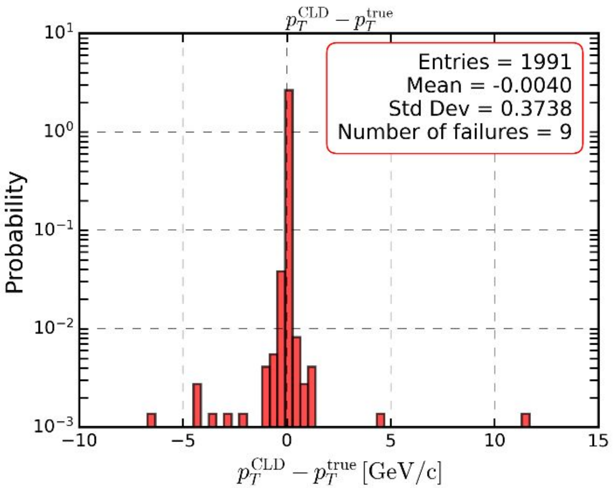
Performance on Silicon Hits

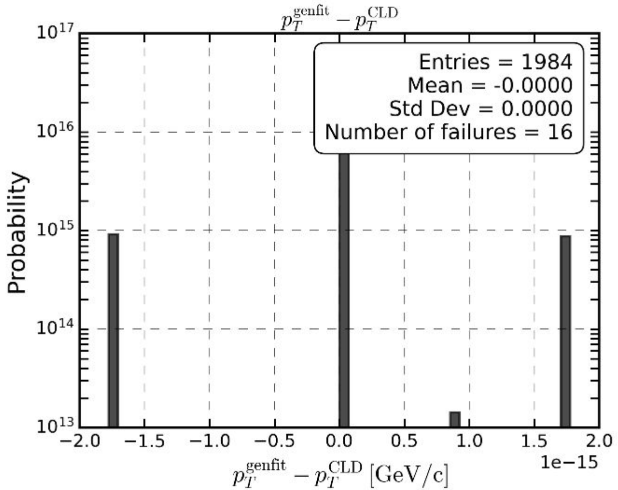
Validation with CLD

Both ILC-inspired results (using `MarlinWrappers`) and Genfit2 results were obtained from the same reconstructed tracks (from Conformal Tracking).

Genfit results are compatible with those of the ILC-like algorithm.





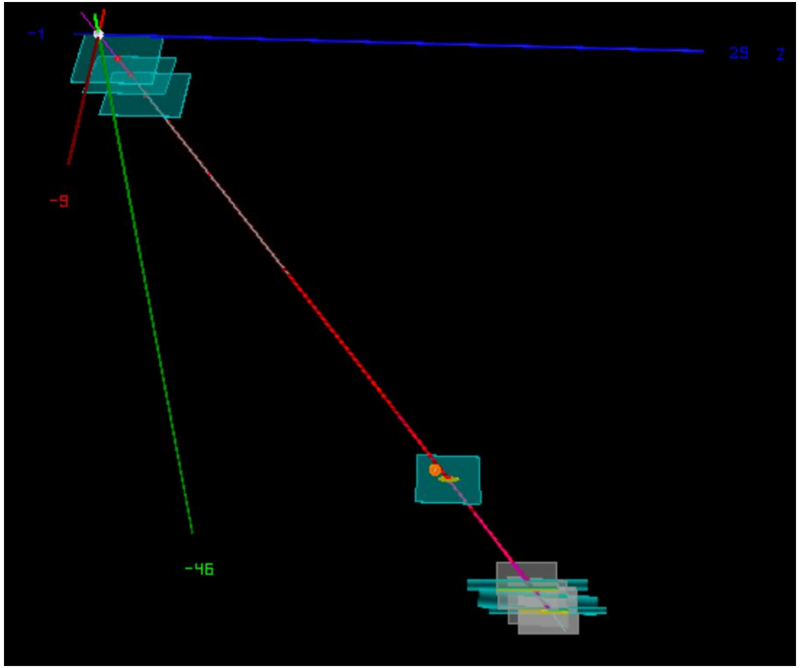


Deterministic Annealing Filter

DAF (Deterministic Annealing Filter): **Iterative Kalman Filter with reweighted observations**

Designed for track fitting in presence of outlier and background hits

- Capable of outlier rejection, **L/R ambiguity resolution**
- Outliers: wrongly assigned hits (background hit found during track finding step)
- Hits are weighted according to their residual to the smoothed track

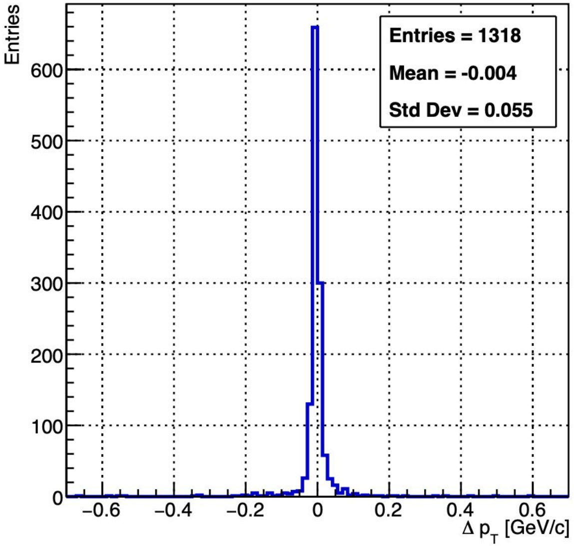


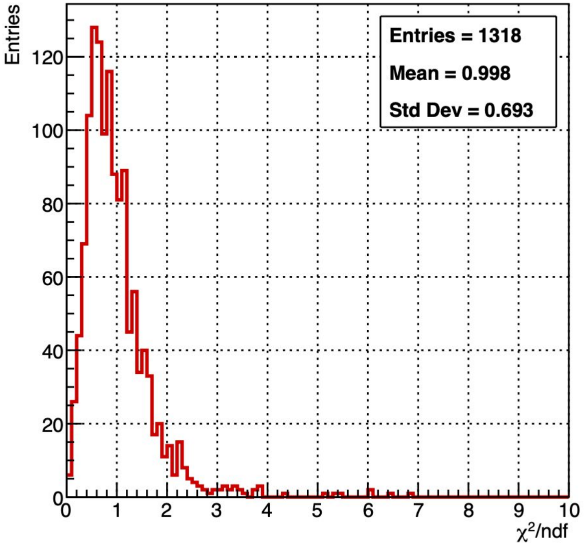
Performance on Pixel + Drift Chamber

pT and χ^2/ndf analysis on MC tracks

Results with fixed theta (60°) and momentum in the range between 0.5 GeV/c and 10 GeV/c.

The simulation involved only pi+ generated in the IP.



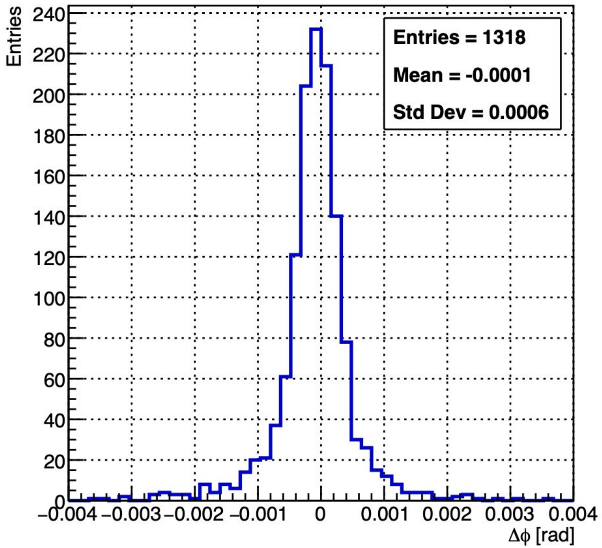


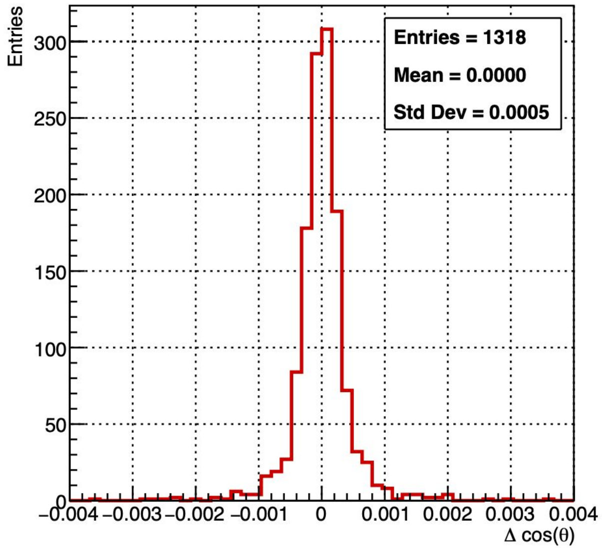
Performance on Pixel + Drift Chamber

cos(θ) and ϕ analysis with fixed θ (60°) on MC tracks

Results with fixed θ (60°) and momentum in the range between 0.5 GeV/c and 10 GeV/c.

The simulation involved only pi+ generated in the IP.



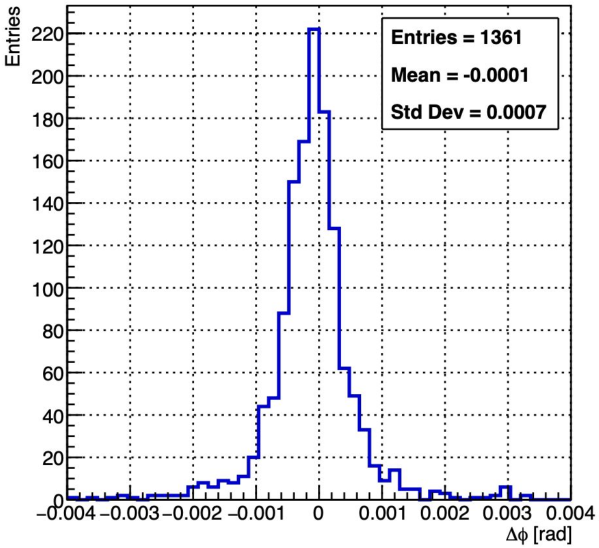


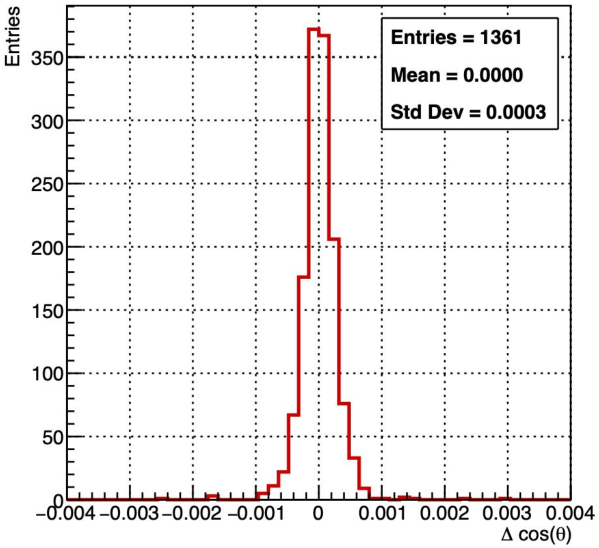
Performance on Pixel + Drift Chamber

cos(θ) and ϕ analysis with fixed energy (5 GeV) on MC tracks

Results with E = 5 GeV and θ in the range between 15° and 80°.

The simulation involved only pi+ generated in the IP.





ML Particle Flow

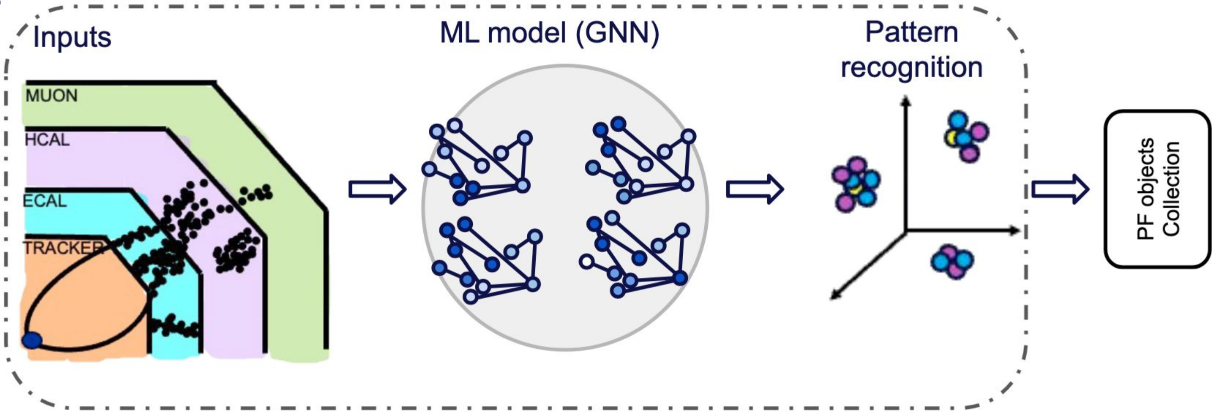
Introduction to the ML approach

See Dolores' presentation for more details!

Detectors geometries are close to fully implemented in full simulation → **focus shifting to reconstruction**

- Adaptable reconstruction is crucial for a systematic design optimization:
  - MLPF is a **flexible framework** that can be adapted to changing detectors
- The **truth definition** is important and needs to be designed carefully: What do we want to reconstruct?
  - For example, are we interested in the bremsstrahlung-recovered electron or in the electron–photon pair?

Hit-Based MLPF



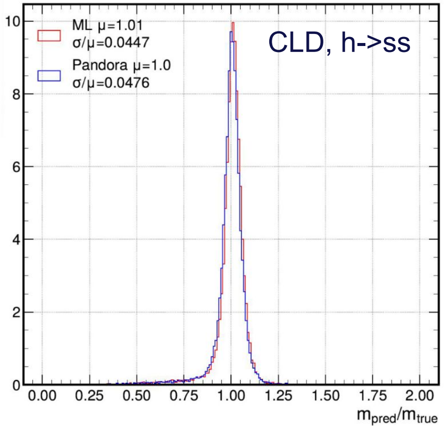
ML Particle Flow Status @ FCC

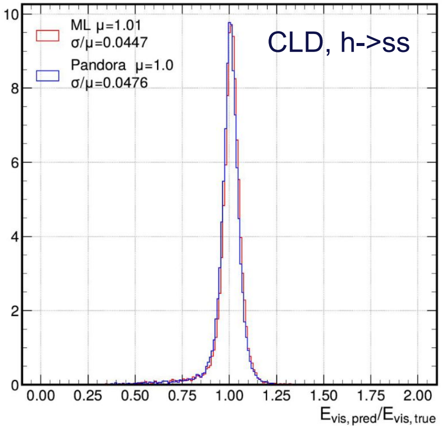
CLD:

- The results are consistent with those obtained using **Pandora**, and **outperform Pandora in jet-like event scenarios**.
- **MLPF (hit-based)** applied to CLD across different case studies (*Dolores Garcia, Lena Herrmann*):
    - *Particle guns*
    - *Jet-like events:* simplified dataset used to study performance and ensure balanced training
    - *Physics events*
- **MLPF (cluster-based)** applied to CLD on physics events (*Farouk Mokhtar*)

ALLEGRO:

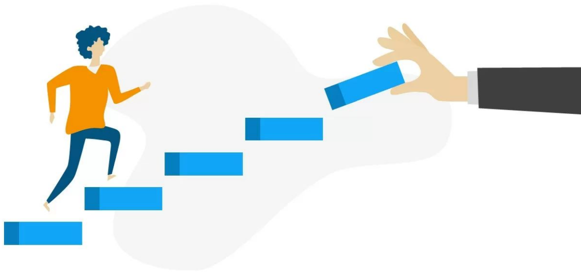
- Pipeline training in progress (Giovanni Marchiori, Rami Chaafa)
- Dataset with particle gun samples is ready
- First training runs for clustering and regression have started





Summary

Summary



Geometric Graph Neural Network Based Track Finder:

- **GGTF Method**: An end-to-end, detector-agnostic approach that shows encouraging results for both CLD and IDEA detector concepts.

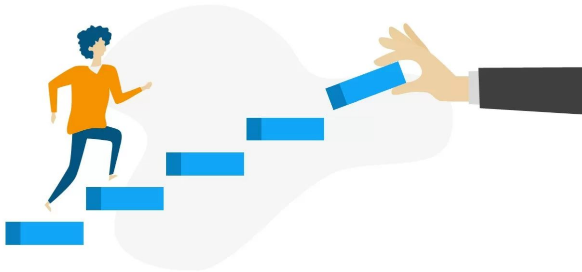
Track Fitting using Genfit2:

- **Genfit2 framework**: The Genfit2 toolkit provides track representation, track-fitting algorithms and graphic visualization of tracks and detectors.

ML Particle Flow

- **ML Particle Flow:** End-to-end approach applied to full sim for jet-like particle gun and physics events in CLD. MLPF outperforms pandora in the jet-like particle gun approach and shows similar performance for physics events.

Opportunities



Geometric Graph Neural Network Based Track Finder:

- Background studies on IDEA to evaluate the effect of drift chamber noise on GGTF performance.
- Studies on the best training dataset definition: how do we labels for the training?
- Studies on an heuristic-based approach to validate the machine learning algorithm

Track Fitting using Genfit2:

- Studies on physics events, e.g. Z->qq

ML Particle Flow

- Studies on the definition of ground truth: what do we want to reconstruct?
- Studies for optimizing tau reconstruction (ongoing studies by Olmo Arquero and Maria Cepeda)
- Studies on track-informed MLPF

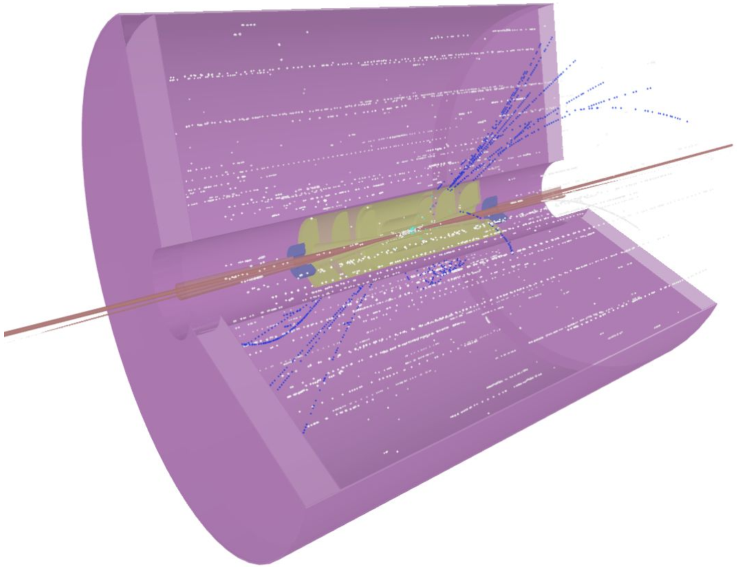
Thank you
for your attention.

Backup

Background Studies on IDEA

Future studies will evaluate tracking performance with background on IDEA, focusing on challenges posed by **low-energy noise particles** in the drift chamber.

These particles create horizontal lines of **loopers** with a very small radius. To prevent the identification of false tracks, they must be detected and removed before pattern recognition.



Tracks within the Genfit2 framework

The object processed by the fitter is a `genfit::Track`. Tracks are parameterized by 5 parameters:

- q/p: **Charge over momentum**
- u, v: **Rectangular coordinates of the plane intersecting the track**
- u', v': **Projections of momentum direction on the coordinate axes**

All per-track data is stored within the `Track` object. It contains a set of `TrackPoint` objects, which may include `Measurements` and `FitterInfo` objects. Each `TrackPoint` holds the track status at the volume where the hit was registered. This status corresponds to the parametrized track parameters after the fit.