# Tutorials on Tracking, PID and Vertex

C.Zhang/04Mar2025

## Preparation

- There is public git repository for this tutorial
  - Copy the code from my working area
    - /cefs/higgs/zhangcg/cepc/04Mar/CEPCSW/Analysis/
    - RecVertexing, TrackInspectE, ksvtx.py, tracking.py
    - See also: https://code.ihep.ac.cn/zhangcg/tutorial/-/tree/master?ref\_type=heads
  - Add "Minuit2" in the L53 of cmake/CEPCSWDependencies before compiling

### Tracking

- Track parameters
  - edm4hep::Track is not link to track parameters directly
  - Track parameters are defined with different reference points, you may be interested in the one at IP
- Match track-MCParticle using hits information
  - Tracking algo. can not guarantee one-to-one map between track-MCParticle.
  - Track hit efficiency / purity
  - Useful for performance diagnostics
- Tracking breakdown
  - The default track is "CompleteTracks". You may be interested in some others, for example tracking performance with gas chamber

### • TrackInspectE.cpp

- L600-634, for track parameters
- L374-423; MCParticle-Hit association
- L783-841; Link Track-MCParticle, and evaluate track hit efficiency/purity
- tracking.py
  - L258-280, How to make a full-silicon track
  - Compare it with L308-331, you will find how to set a tracking algorithm



### Vertexing

- Following LCFIPlus, under the Gaudi framework and edm4hep, a class HelixPlus was developed to merge tracks and return a possible vertex along with its probability
- Based on the probability and some other kinematic variables, you can design a specific vertexing task.
  - For K-short, we require the number of tracks = 2, invariant mass between 4.8 and 5.1, and the tracks used for this vertex to have PID as 211,211
  - Similar procedure for Do->Kpi with corresponding PID and mass window

- KoSIdentifier.cpp
  - L292-306, collect secondary tracks and make them as HelixPlus
  - L310-324, form all possible track pairs as vertex candidates
  - L380-392, fit the vertex seed
  - ksvtx.py, will produce some KS, you run it and check its output





## PID (TPC + TOF)

- SW provides us with the likelihood of a stable charged particle being an e/mu/pi/K/proton.
- No place to store this information in the final PFO, you may want to know how get them.
  - TPC reconstruction, DndxTracks
  - TOF reconstruction, RecTofCollection
  - Above two link to CompleteTracks
- $\bullet$ <u>ref\_type=heads</u>
  - L131-165 show how to read the TPC and TOF information
- reconstructed PID information

### https://code.ihep.ac.cn/cepc/CEPCSW/-/blob/master/Reconstruction/ParticleID/src/FinalPIDAlg.cpp?

• You just generated some muons with all the above information. Feed them to the PID algo. and print the

### Summary

- Many opportunities to make contributions
  - For vertexing
    - standalone cross-check
    - We need to mute the errors from Minuit2
    - Develop a beam spot constrain to reduce the competition between PV and SV reconstruction
    - Some c++ structs are better re-written as c++ classes

### • You are welcome to use other packages like ACTS, LCFIPLus to perform a