Progress of Tracking Software in CEPCSW

Chengdong FU for CEPC Software Group
IHEP, CAS

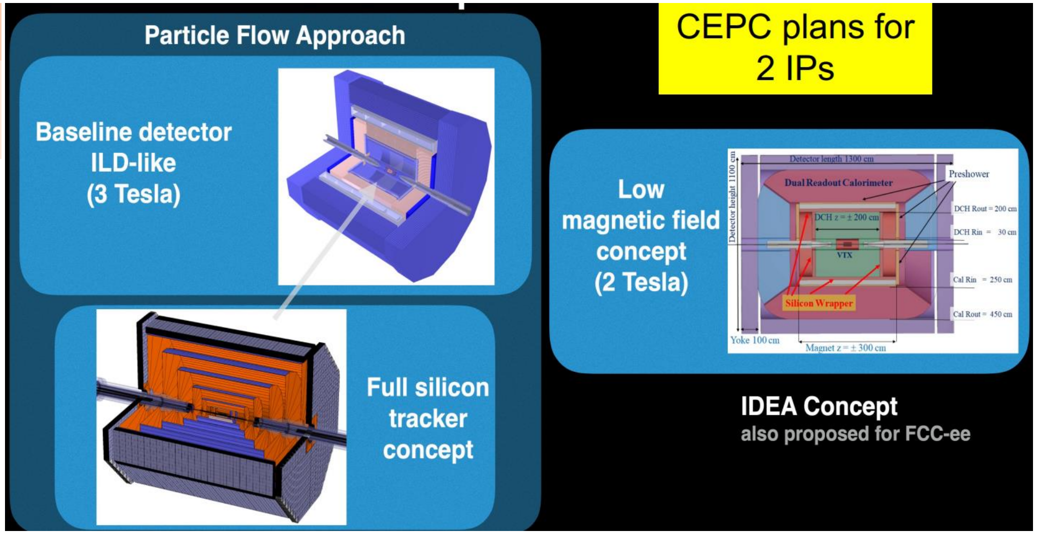
CEPC Day
June 29, 2022

Outline

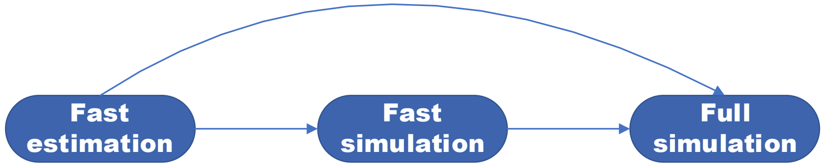
- Introduction
- Tracking software
- Tracking option
- Performance test
- Plan & Summary

Introduction

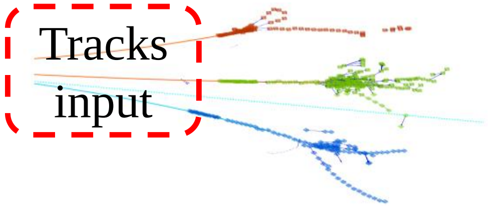
Physics process	Measurands	Requirement on tracker
ZH, $Z\to e^+e^-(\mu^+\mu^-)$, $H\to\mu^+\mu^-$	m_H, $\sigma(ZH)$, $BR(H\to\mu^+\mu^-)$	$\Delta(1/p_T) = 2\times10^{-5} \oplus \frac{0.001}{p(GeV)\sin^{3/2}\theta}$

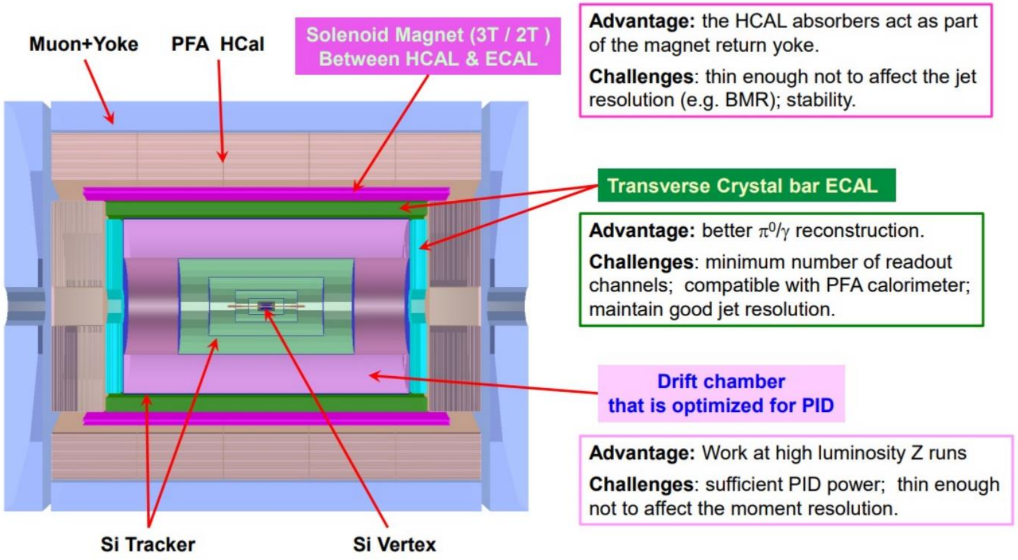


- CEPC being designed as Higgs&Z factory, has basic physics requirements, such as tracking resolution.
- Three detector concepts were designed at CDR stage, and the 4th conceptual detector design has been proposed since 2021.
- Track reconstruction for estimation on detector performance at post age of CDR, exactly as at CDR stage



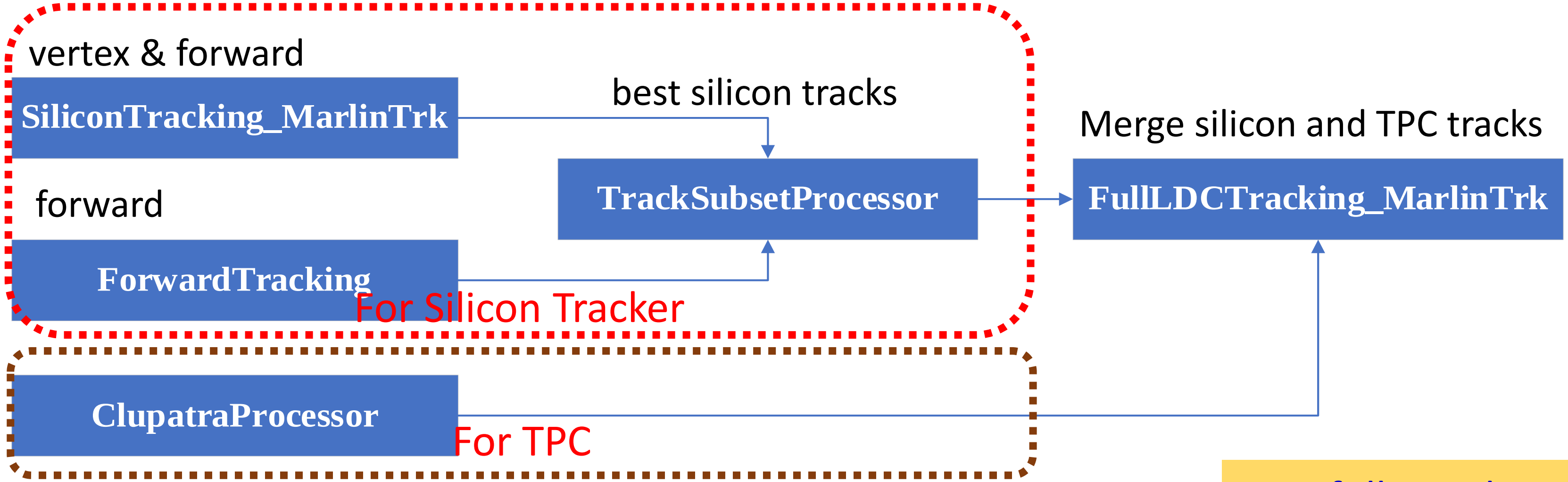
- Tracker design: track performance estimation
- As PFA input





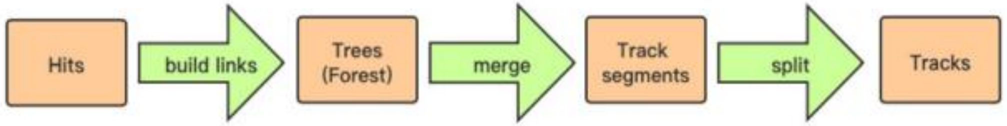
History: Tracking for CDR in Marlin

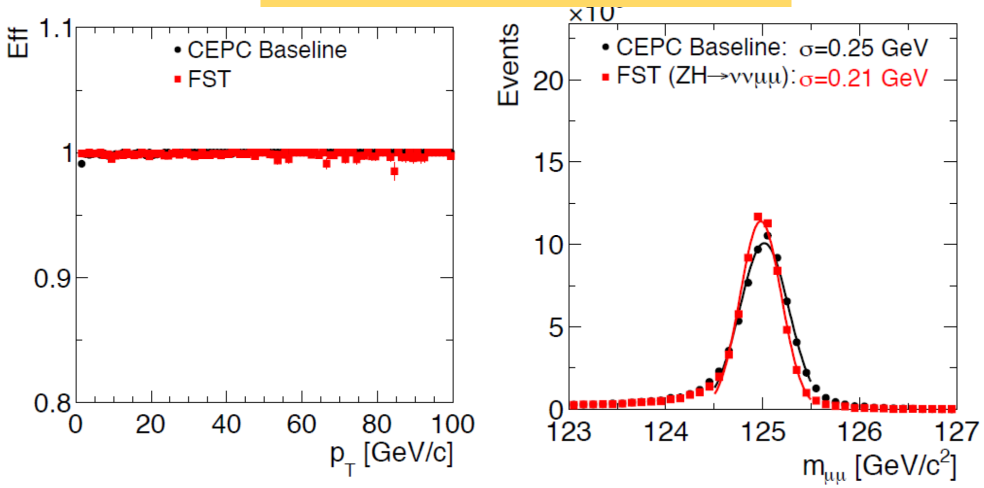
- From ILCSoft
  - Use different tracking for different trackers, and then merge



CDR full simulation study

- Developed inspired by the idea of Arbor
  - ArborTracking algorithm

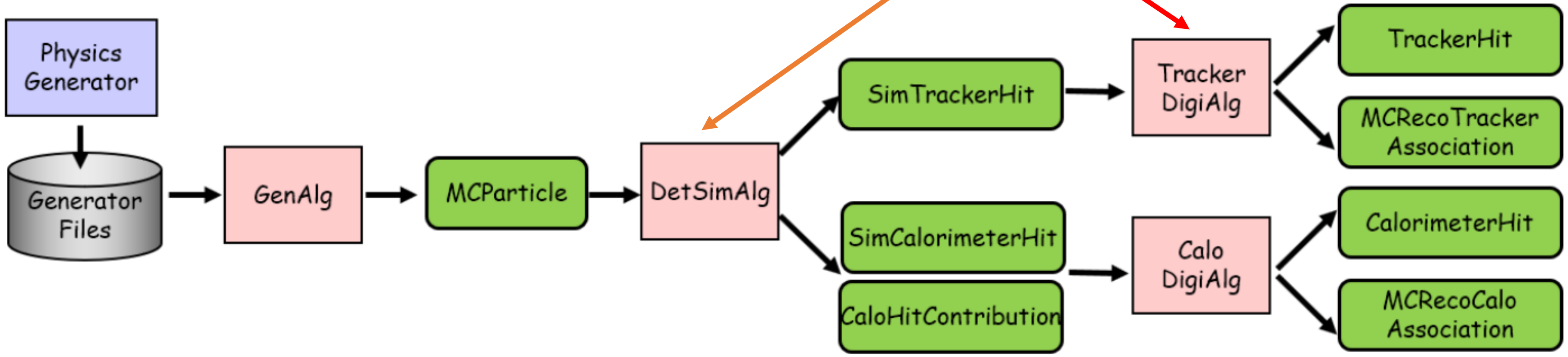




- Implement ConformalTracking
  - Test for the full silicon tracker (FST) concept

CEPCSW & Input

- CEPCSW is a Gaudi-based framework
  - Core software, application, external libraries
- EDM4hep for event data model
- DD4hep for detector description
  - Originally developed for ILC and CLIC but with all of HEP in mind
  - Covering the full life cycle of an experiment
    - ✓ Detector concepts, optimization, construction and operation
  - DDG4 provides API from xml compact files and DD4hep constructor to Geant4 geometry, DDCore for interface to DD4hep geometry (DetElement, Surface, etc) & Gear geometry
    - ✓ a single source of information for Geometry, materials, visualization, readout, alignment, calibration, reconstruction etc

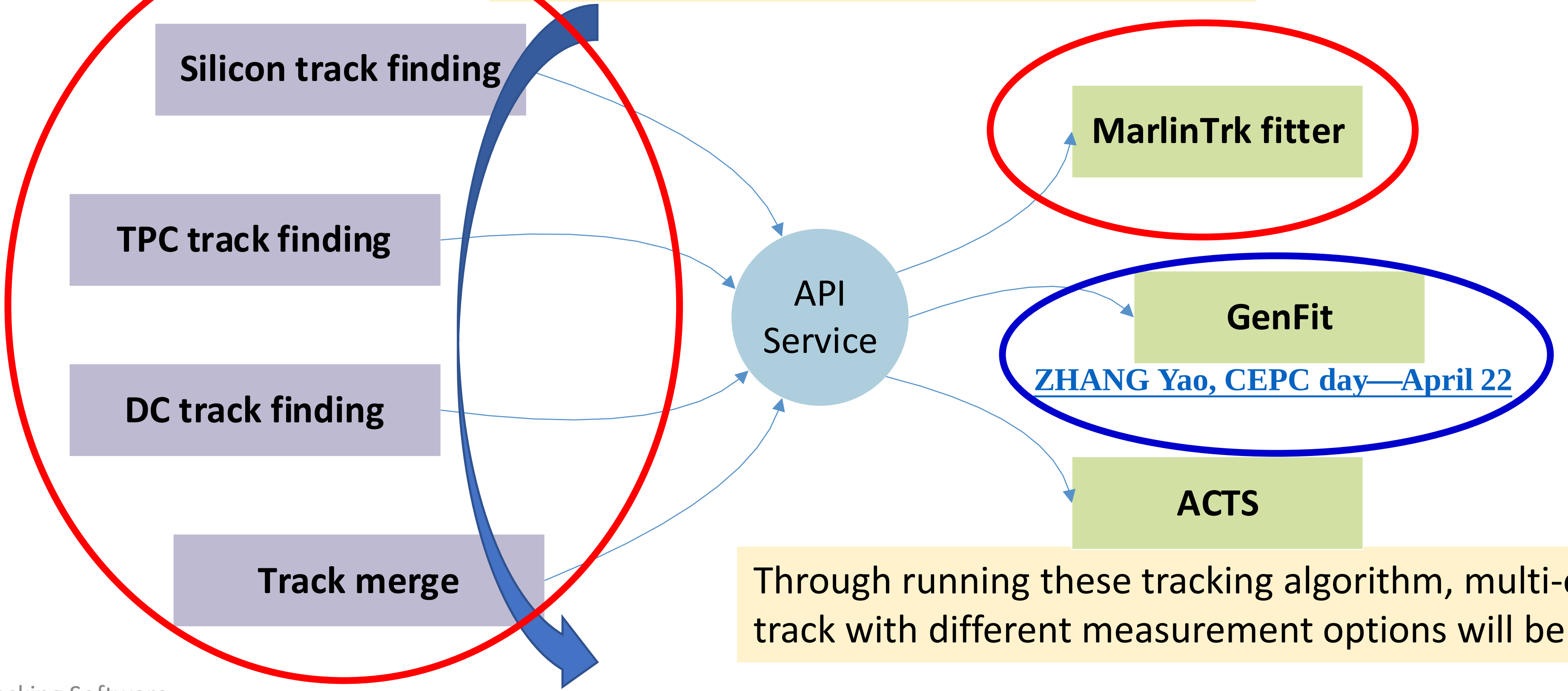


Propose of Tracking Chain

- Service to call the API of fitter according to option
- To choice best combination of track finding and fitting
  - For middle tracking, low CPU time
  - For final tracking, high performance
- Comparison on same detector design will be performed

BEST: appropriate performance and CPU time

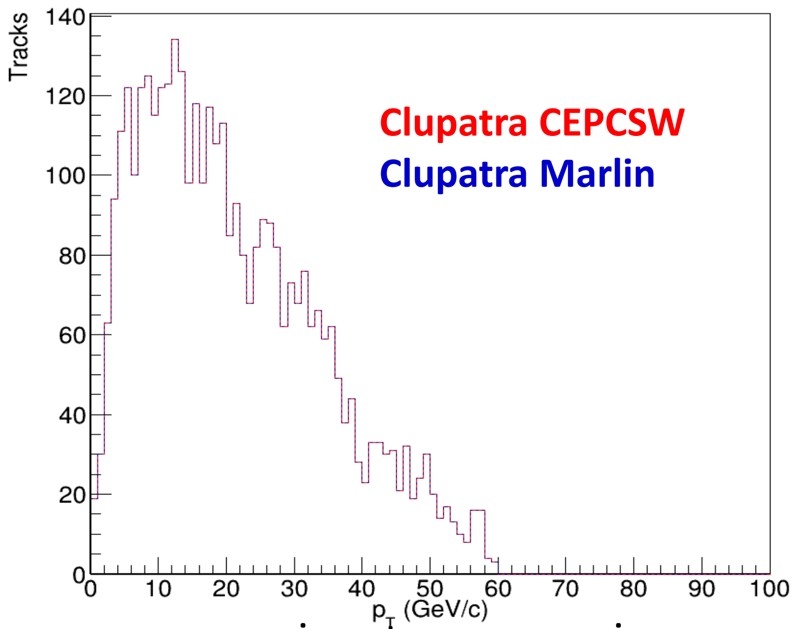
Matrix manager for finding and fitting



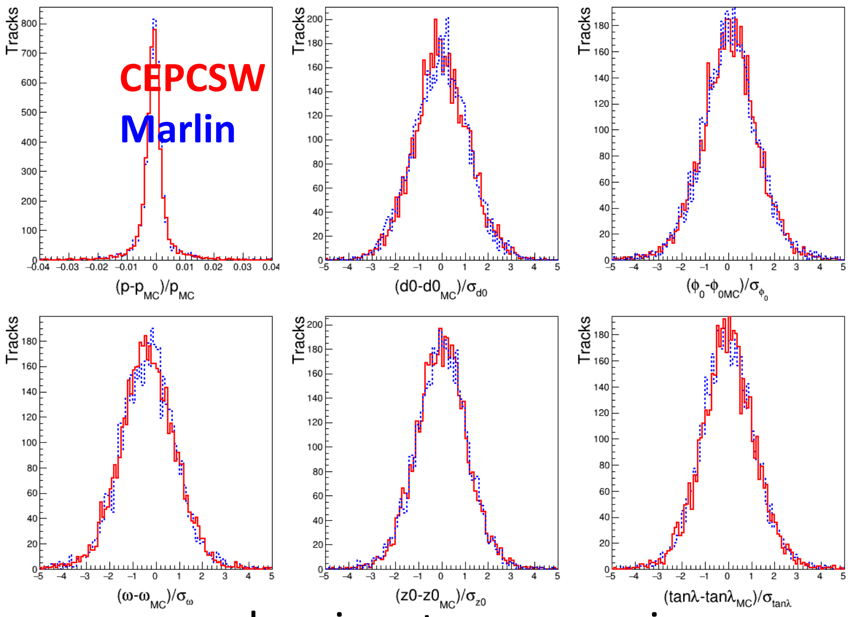
Through running these tracking algorithm, multi-candidates track with different measurement options will be obtained

Migrated MarlinTrk

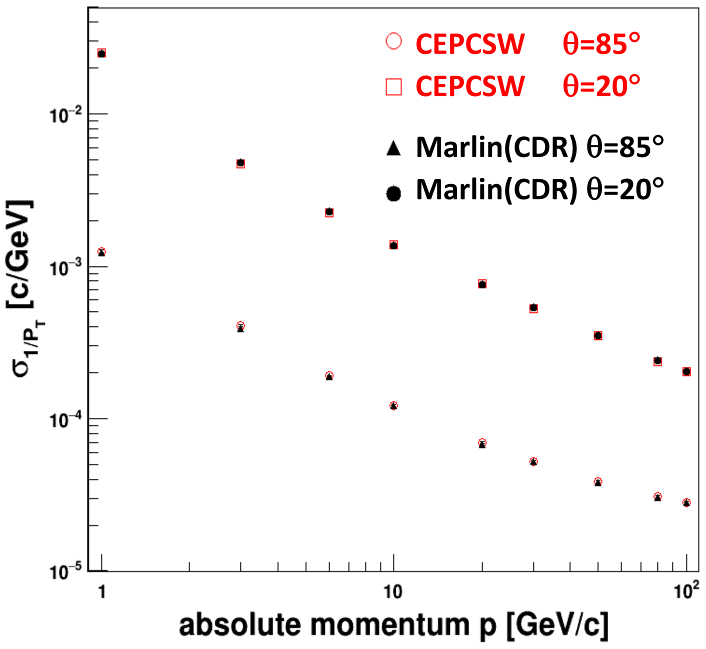
- Completed
  - Geometry convertor
  - Service to call fitter
  - Switch data model to EDM4hep
  - CRD tracking chain
- Upgrade to DD4hep surface (DDKalTest) is in plan



same input as comparison



random input as comparison



Output

edm4hep::Track

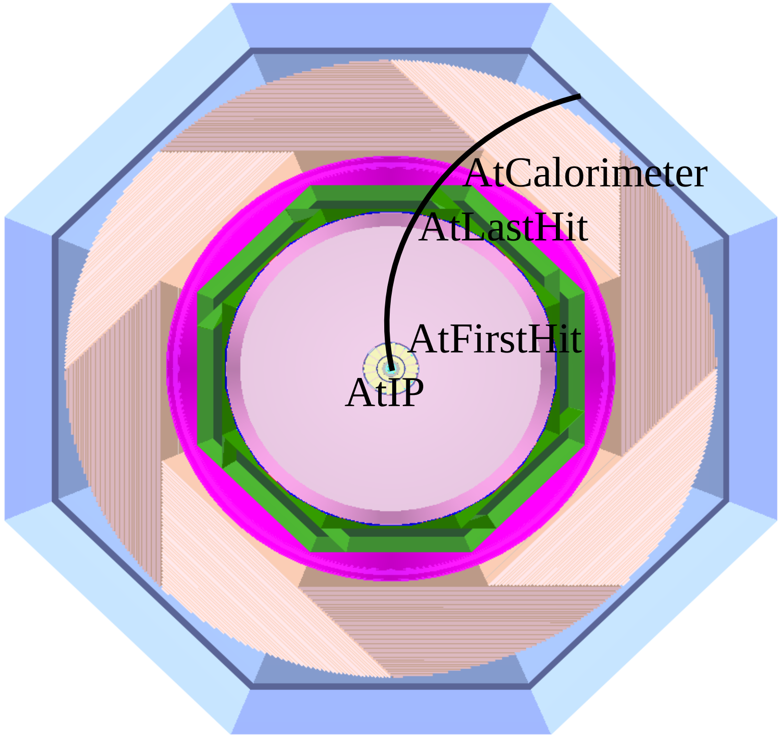
……

edm4hep::TrackerHit s

……

edm4hep::TrackState s

std::int32_t location
float D0
float phi
float omega
float Z0
float tanLambda
float time
edm4hep::Vector3f referencePoint
std::array<float, 21> covMatrix

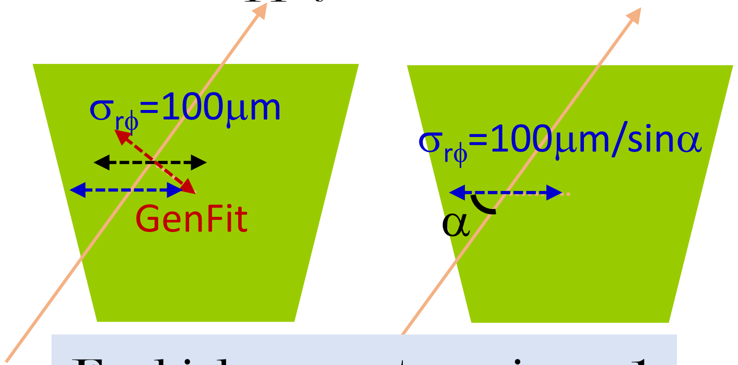


Currently, four TrackStates are in storage:
if(location==edm4hep::TrackState::AtIP)
if(location==edm4hep::TrackState::AtFirstHit)
if(location==edm4hep::TrackState::AtLastHit)
if(location==edm4hep::TrackState::AtCalorimeter)

CylinderDigiAlg

- Ability to perform on bent CMOS sensor
- Test to apply on drift chamber for MarlinTrk fit

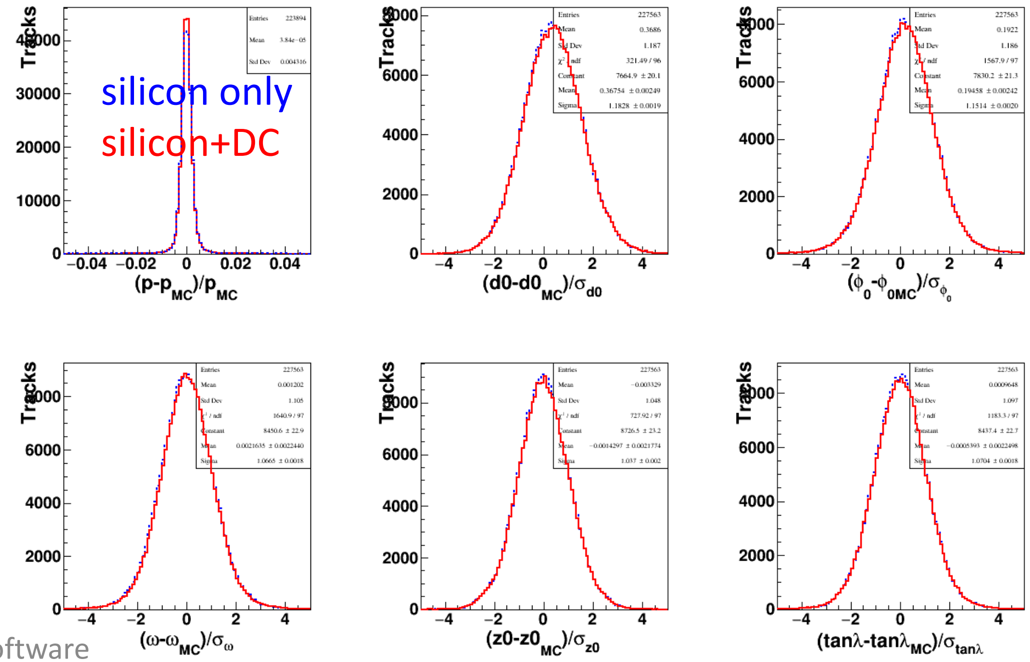
from Magnus Mager's talk

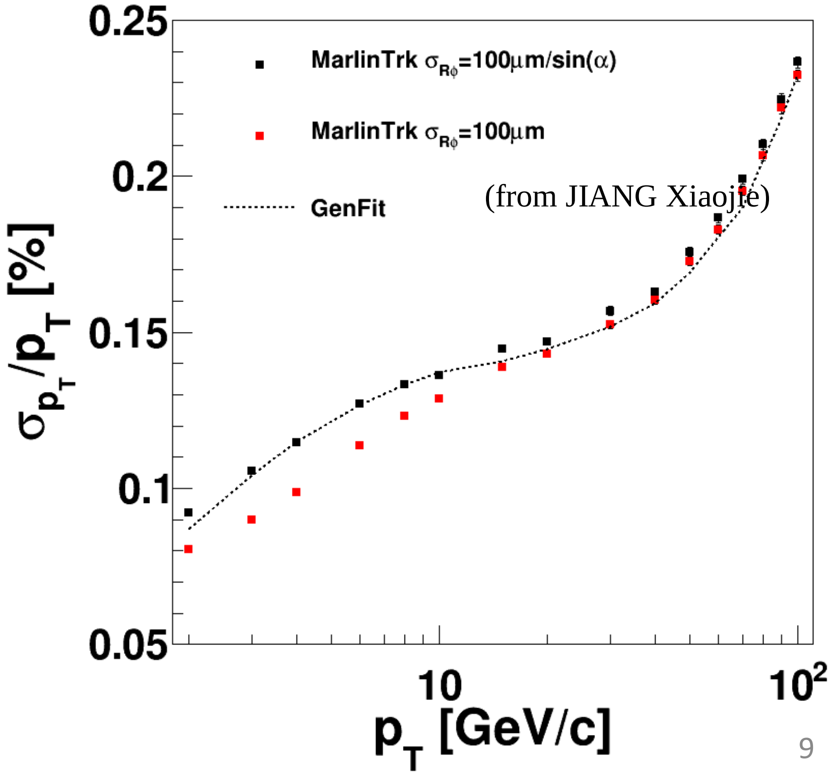


With ratio: close contribution in fit

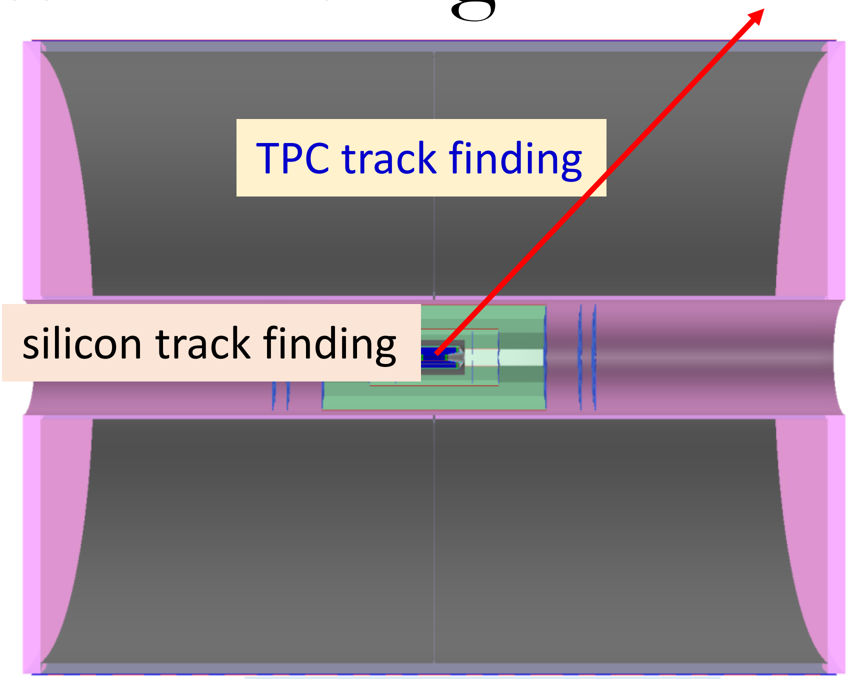
$$\frac{(r\phi - r\phi_{extra})}{\sigma_{r\phi}} \approx \frac{(L_{drift} - L_{drift,extra})}{\sigma_L}$$

For high momentum, sinα→1

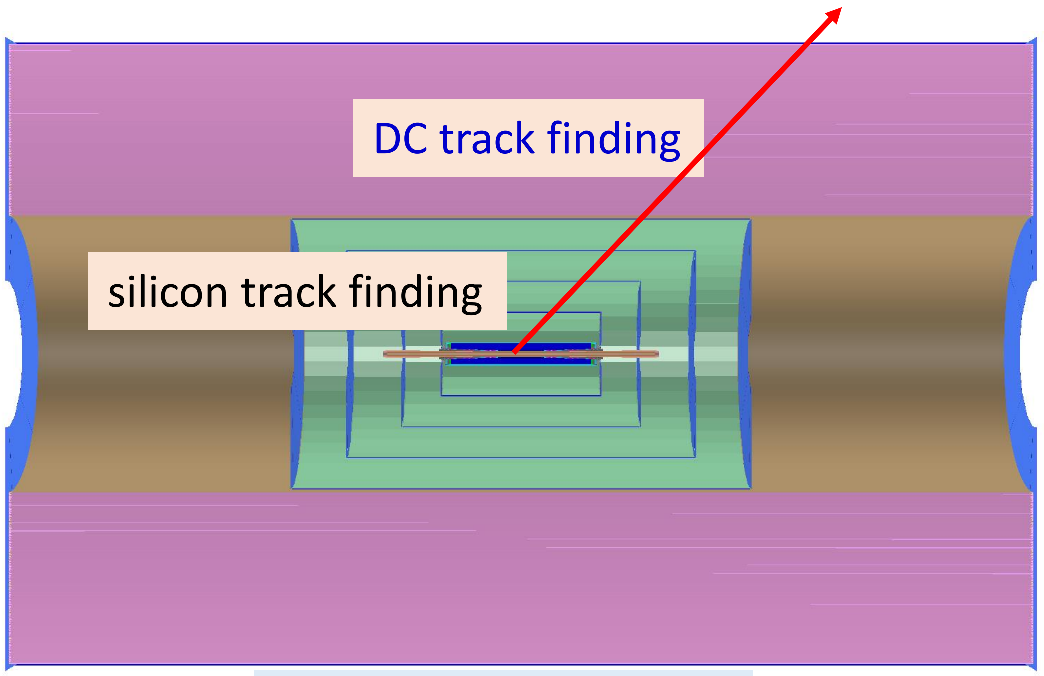




Track Finding



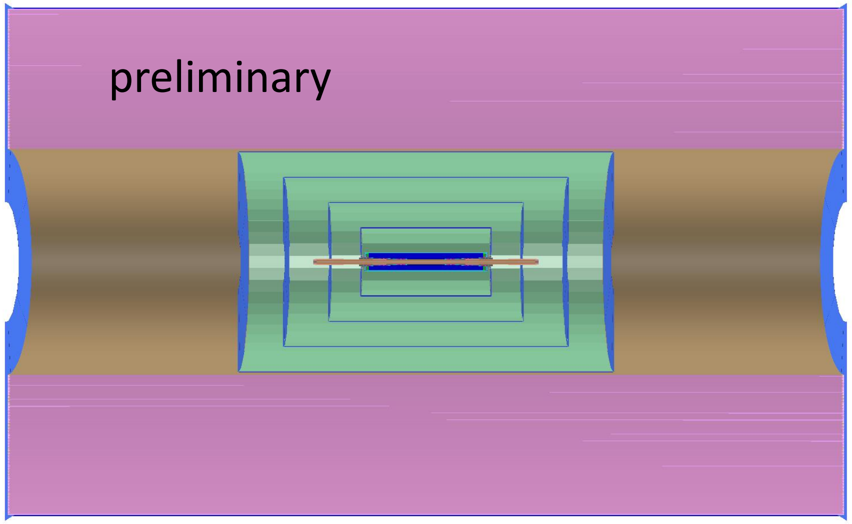
CDR Baseline detector

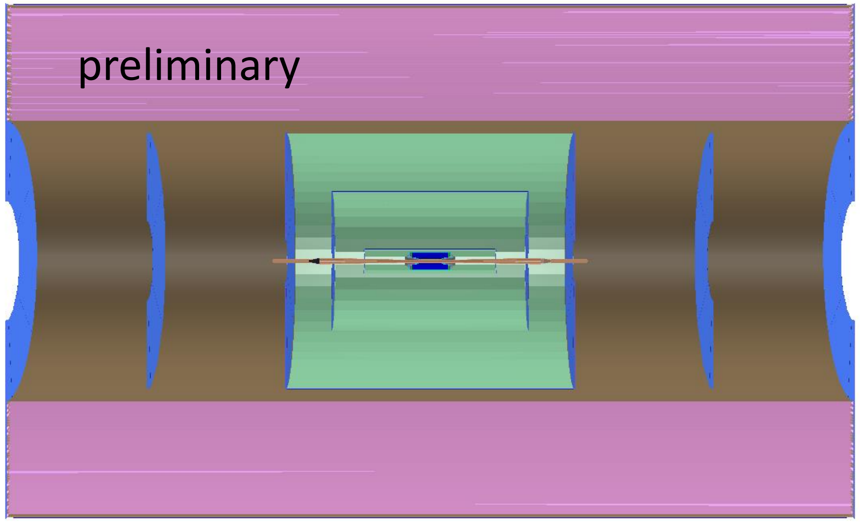


4th conceptual detector

- Different pattern recognition for gas tracker and silicon tracker
  - CDR baseline detector
    - ✓ better resolution from TPC track → TPC track as base
  - 4th conceptual detector
    - ✓ worse resolution from DC track → silicon track as base
- Same pattern recognition in once time is in consider

Tracking System

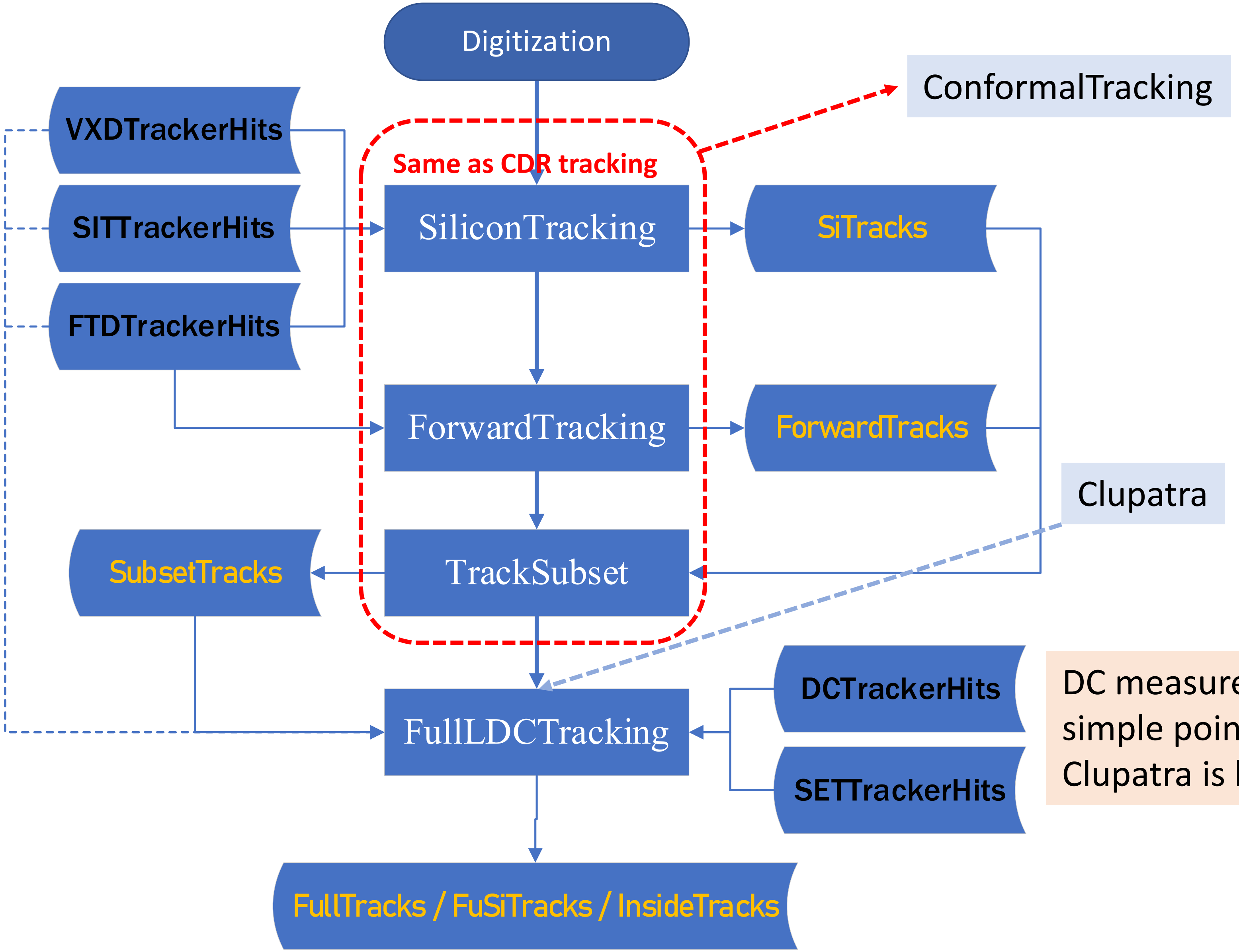




- Vertex detector (VXD): 6 pixel layers
  - $\sigma_{rphi,z}$=2.8μm, 6μm, 4μm, 4μm, 4μm, 4μm
- Silicon inside/internal DC tracker (SIT): 4 or 3 pixel layers
  - σ_{rphi} =7.2μm, σ_z=86μm
- Silicon outside/external DC tracker (SOT/SET): 1 pixel layer
  - σ_{rphi} =7.2μm, σ_z=86μm
- Endcap tracker (EIT&EOT/FTD): 2 + 3 pixel layers
  - $\sigma_{x,y}$=3μm, 3μm, 7.2μm, 7.2μm, 7.2μm
- Drift chamber (DC): 18mm or 10 mm cell size
  - σ_{rphi} =100μm, σ_z=2.828mm

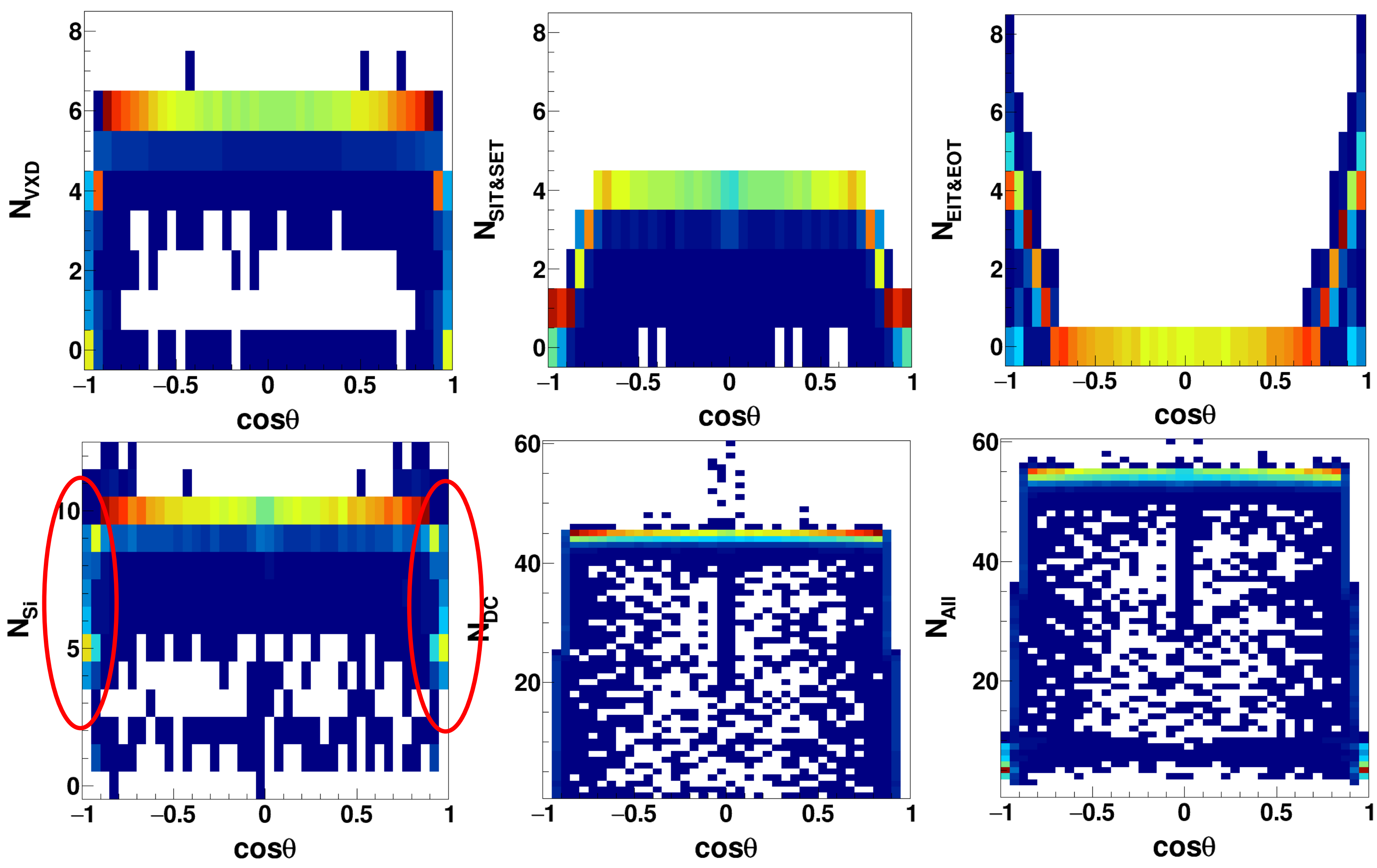
optional layout
optional resolution

Tracking Option for the 4th Conceptual Detector



DC measurements regarded as simple point to add, merge from Clupatra is being optimized

Hits Number in Fit

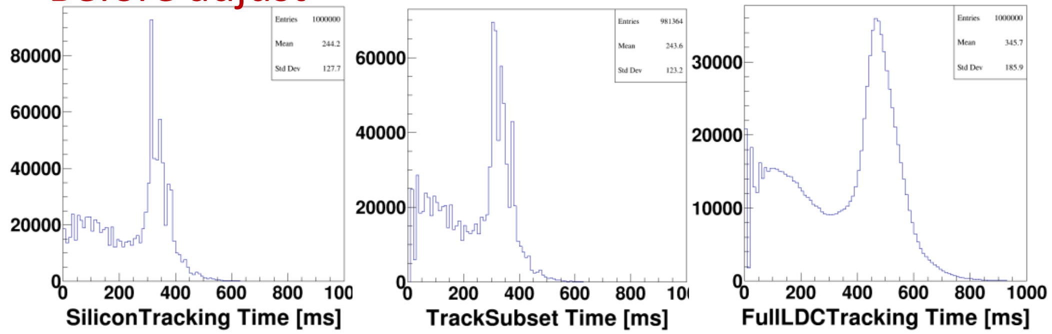


- **Barel**: ~10 silicon hits used in fit; **Endcap**: 10 or less, determined by vertex cover range

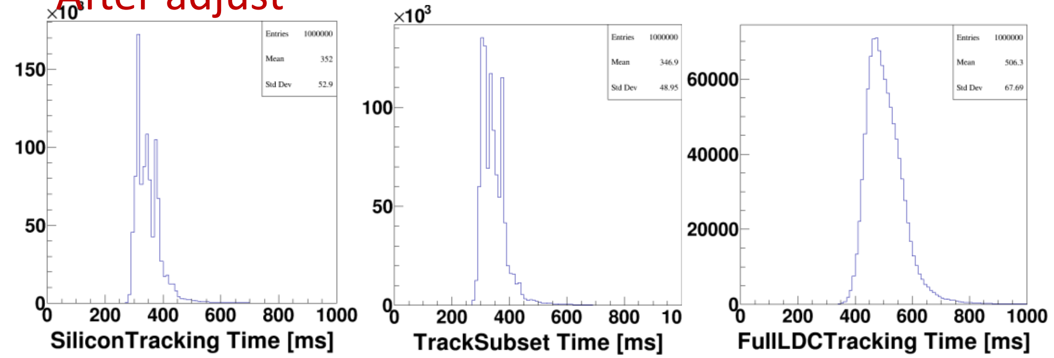
Efficiency VS Cost Time()

- 20 particles in each event
  - $p \in [2, 50]$ GeV/c
  - $\theta \in [40°, 140°]$
  - Independent: $(e^+, e^-, \mu^+, \mu^-, \pi^+, \pi^-, K^+, K^-, p^+, p^-) \times 2$
  - Vertex: (beam parameter @Higgs)
    - ✓ $\sigma x = 15\mu m$
    - ✓ $\sigma y = 36nm$
    - ✓ $\sigma z = 3.9mm/\sqrt{2}$
- Tracking efficiency
  - Match: >50% of hits shares between MC truth and tracks
  - Improve through adjust pattern recognition

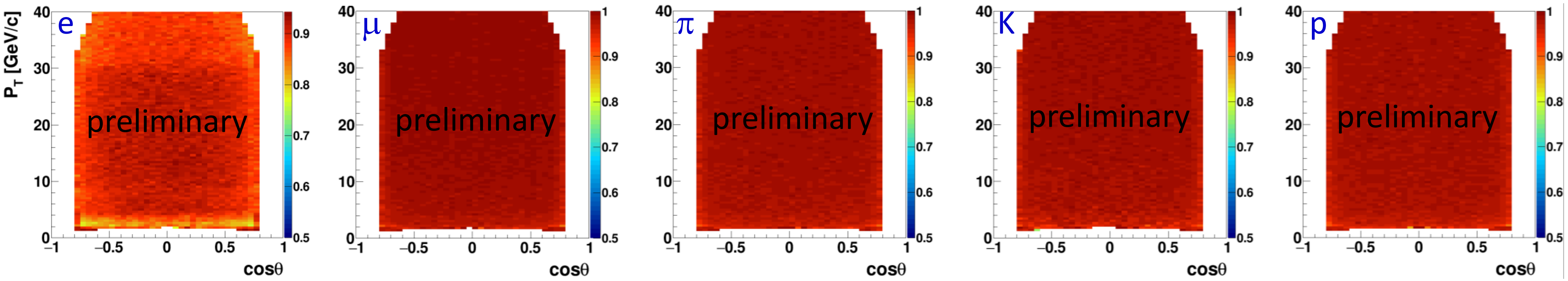
Before adjust



After adjust

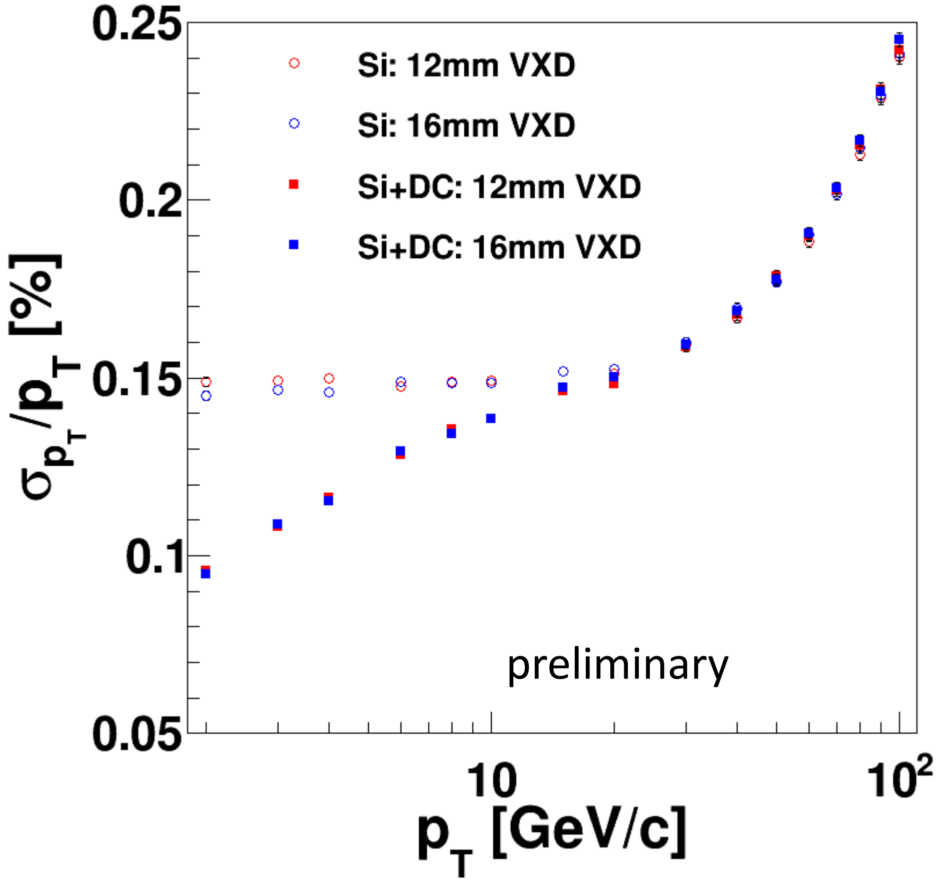


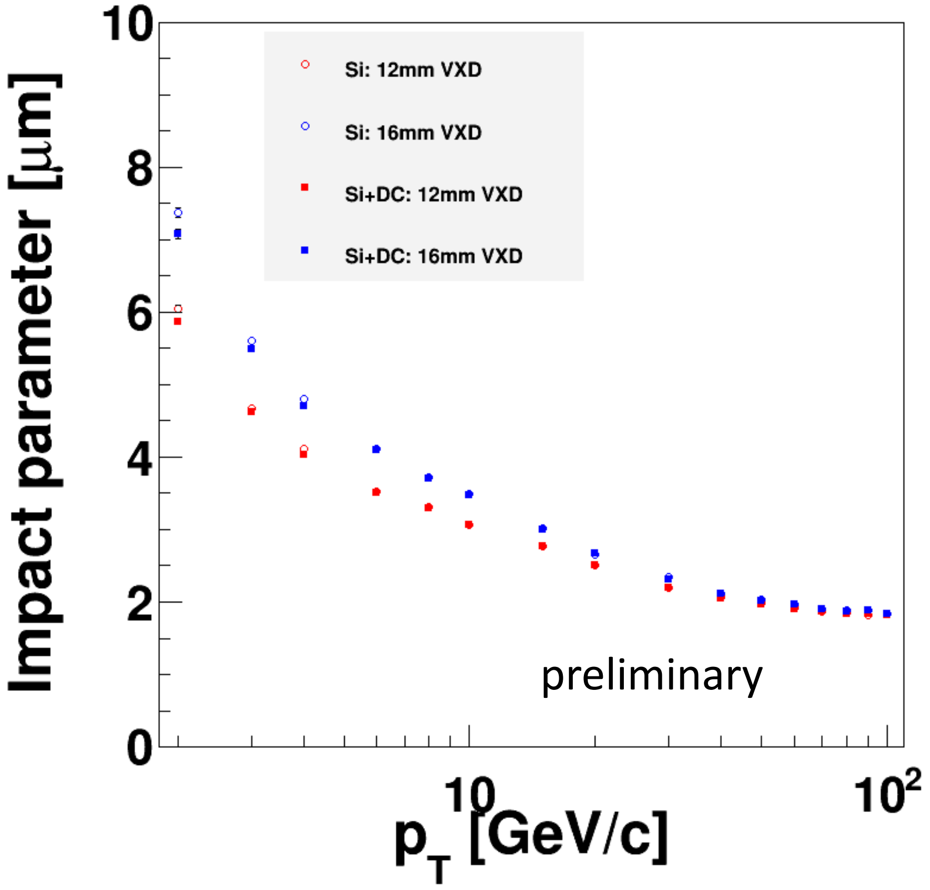
totally about 100 ms per track



Effect of Inner Radius of Vertex

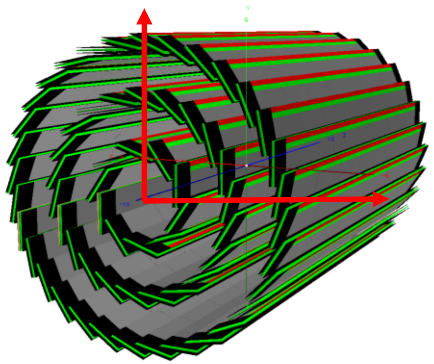
- CRD beam pipe
  - Inner radius: 14mm → Vertex: 16mm
- Newest beam pipe:
  - Inner radius: 10mm → Vertex: 12mm (keep layer5/6, and move layer3/4 2mm)

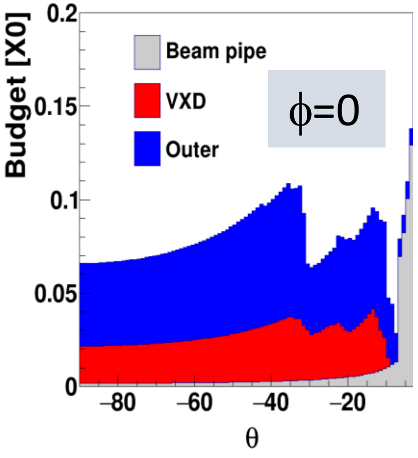


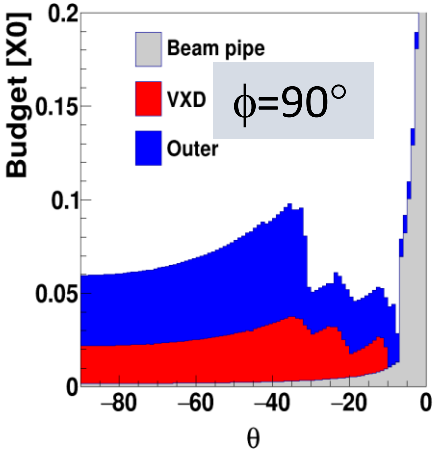


Vertex Detector from MOST2

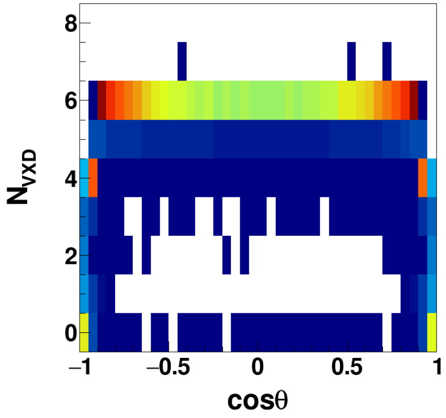
- Geometry has been implemented in CEPCSW with first version (ZENG Hao)
- Tracking software works smoothly, and shows close performance with ILD-like vertex detector.



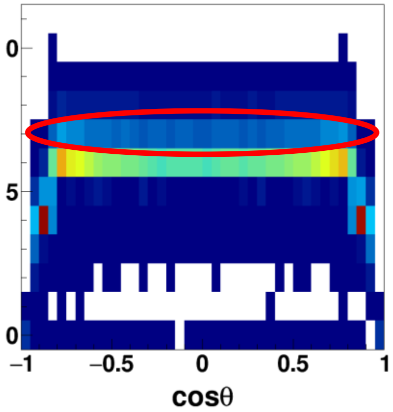




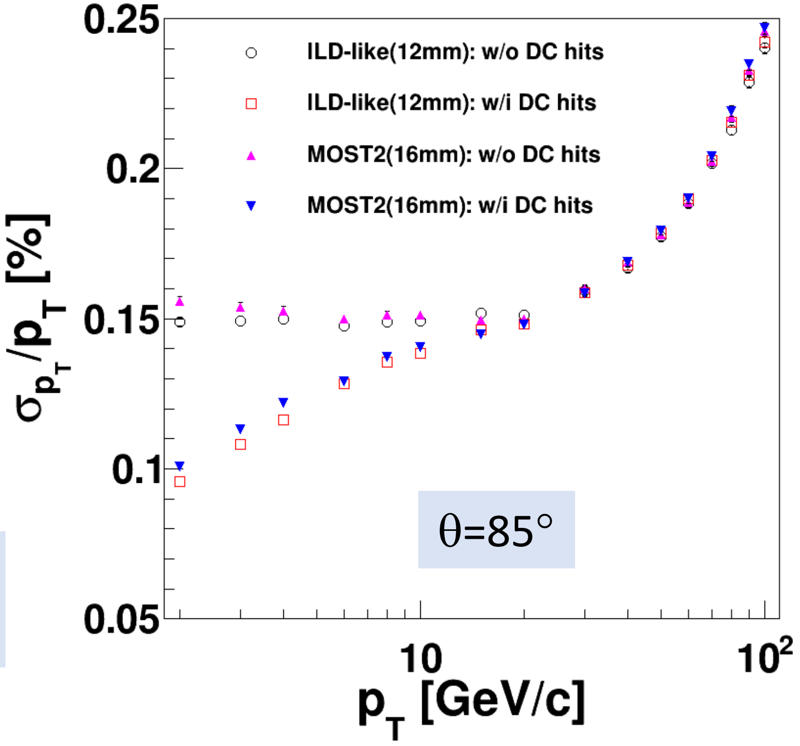
ILD-like

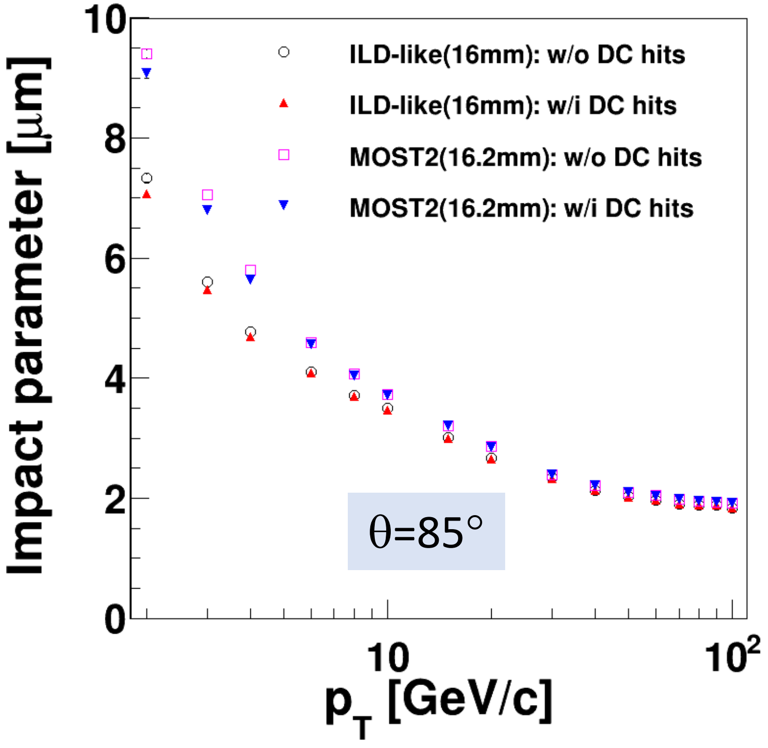


MOST2



ILD-like: mostly 6 hits
MOST2: more possible > 6 hits

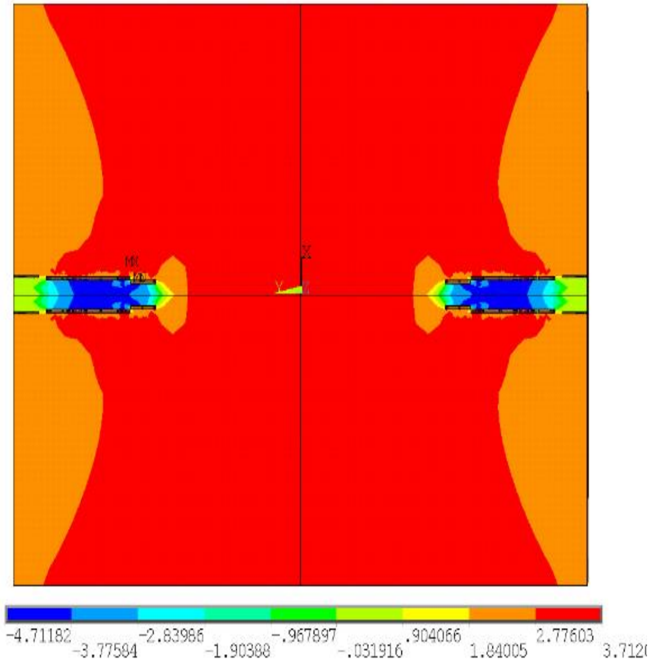




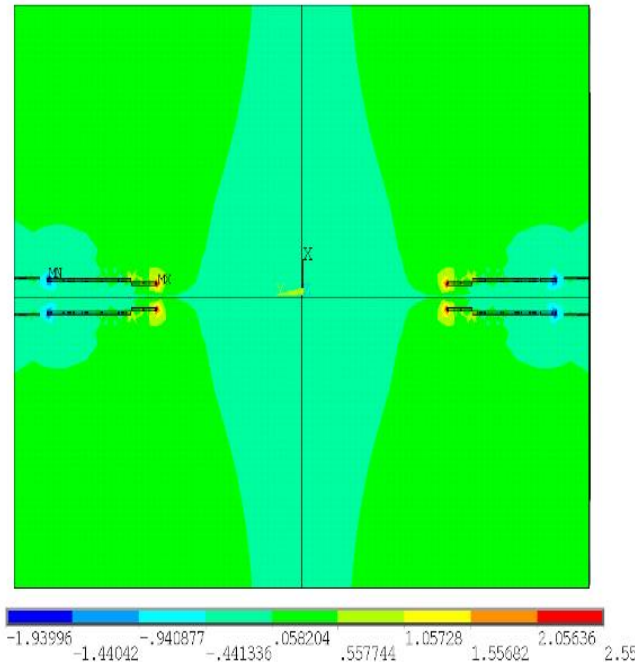
Non-uniform Field Test on MarlinTrk

- Include non-uniform field by map files through GenericBFieldMapBrBz in simulation
- Keep to use field value at (0,0,0) in reconstruction
  - Resolution changes very small: $(\sigma_{Pt}-\sigma_{Pt,non})/\sigma_{Pt}$ ~4%@100GeV
  - momentum departure from MC truth, to correct through average past field

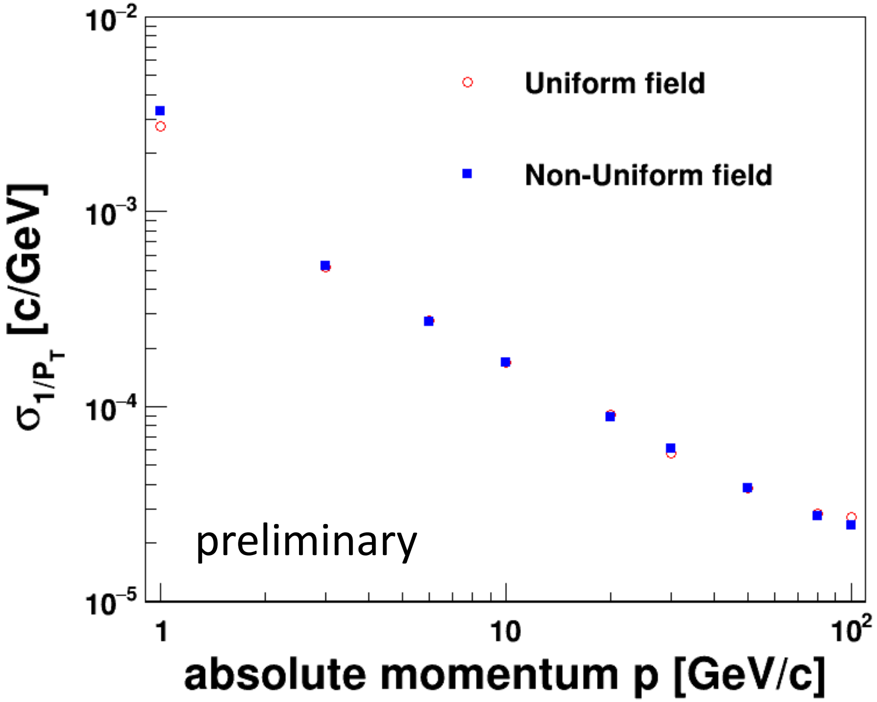
~5% non-uniform level



Bz



Br



Background

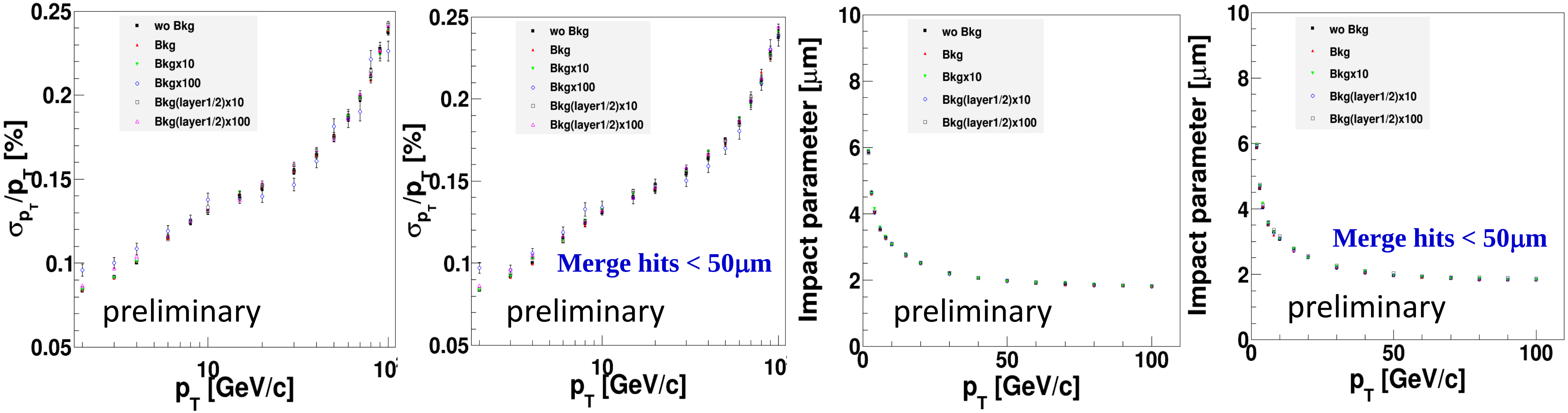
- As known, more close to beam line, more background from beam
- According to Haoyu's talk on 2021 Workshop

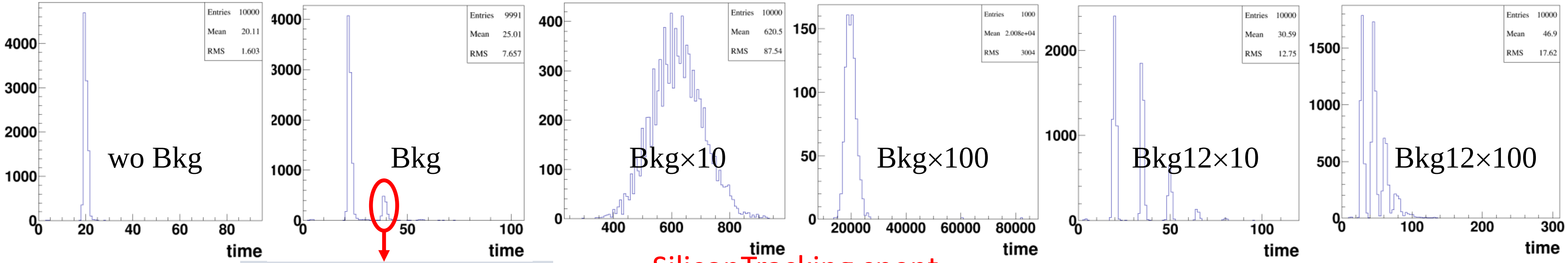
Name	Position	Hit/cm^2/BX	Hit/cm^2/s
VTX	15 mm	~2.3	~3.33e7
SIT	15 cm	~0.01	~14507
TPC	50 cm	~0.005	~7253
Ecal	200 cm	~1e-4	~145
Hcal	220 cm	~2e-6	~2.9

  - Assume $\sim e^{-r/c}$
  - For VTX ($r_{1st\ layer}$=12mm), Hit/cm^2/BX
    - ✓ 2.6, 2.4, 1.0, 0.96, 0.41, 0.38 (6 VTX layers) as baseline
    - ✓ × 10 or 100 for all layers
    - ✓ × 10 or 100 for 1st/2nd layers
- Background mixing (simply) with single muon (θ=85°)
  - Random hits at silicon sensors
  - Optional merge for distance less than 50μm
- More reliable estimation using believable beam background from beam simulation in plan

Effect of Background/Noise

- Small affect on resolution of real tracks, but cause fake tracks and increase spent CPU time
- If serveral Hits/cm²/BX, background's effect will be small





events have fake tracks

SiliconTracking spent

Release

- Official release v0.2.5: single particle already
  - CRD_o1_v01-SimRec.py for pixel SET
  - CRD_o1_v02-SimRec.py for strip SET
- Github share before official release
  - CEPCSW or personal fork
  - At any time
  - Need close contact
- Plan
  - Release of new software updates
  - Option for new detector design
  - Option for physical channels
  - According to development

```
#!/usr/bin/env python
from Gaudi.Configuration import *

from Configurables import k4DataSvc
dsvc = k4DataSvc("EventDataSvc")
……

full.SiTracks  = "SubsetTracks"
full.OutputTracks  = "MarlinTrkTracks"
full.SITHitToTrackDistance = 3.
full.SETHitToTrackDistance = 5.
#full.OutputLevel = DEBUG

#TODO: more reconstruction, PFA etc.

# output
from Configurables import PodioOutput
out = PodioOutput("outputalg")
out.filename = "CRD-o1-v01-SimRec00.root"
out.outputCommands = ["keep *"]
```

Challenge & Plan

- Man power
  - Dominantly dependent on migration/implementation
  - Algorithm development focus on optimization and machine learning
  - Key4hep is hopeful to reduce requirement on manpower: such as ACTS, once integrated into Key4hep, easy to load
- Multiple detector designs
  - **Confirmed** for silicon tracker: SiliconTracking, ForwardTracking, TrackSubset, ConformalTracking
  - **Confirmed** for TPC: Clupatra
  - **Testing** for drift chamber: Clupatra
  - To develop new algorithm on machine learning, and to study same track finding for both silicon and DC
- Long life cycle of experiment from pre-CDR to running, different requirements
  - Resolution study: test done
  - Non-uniform magnetic field: test done
  - Background & noise: test done
  - CPU time & memory: test done
  - Real digitization: in plan
  - Alignment: in plan
- Large data sample
  - Multi-thread: GaudiHive
  - Platform independent: OneAPI→GPU、Super-computer

Summary & Conclusion

- Tracking algorithm and Kalman filter tool have been migrated successfully from Marlin into CEPCSW as first step, work well for the CDR baseline detector.
- Update for the 4^{th} conceptual design detector has been done, complete tracking chain from track finding to track fitting works.
  - Cross check between MarlinTrk and GenFit shows consistent result.
- Tracking for new MDI and new vertex detector from MOST2 works respectively, combined with other sub-detector.
- Performance test shows reasonable result: tracking efficiency, resolution, CPU time.
  - But tracking control parameter set still has improving capacity. Optimization will be continued.
- Non-uniform field and background are considered in preliminary, showing small effect on resolution.
  - Further detail study is still needed.

Thanks very much for your attention!