ILD Tracking Fix Update

Uli Einhaus
with contributions from Victor Schwan and Thomas Madlener

ILD Detector Optimisation Meeting
25.11.2026



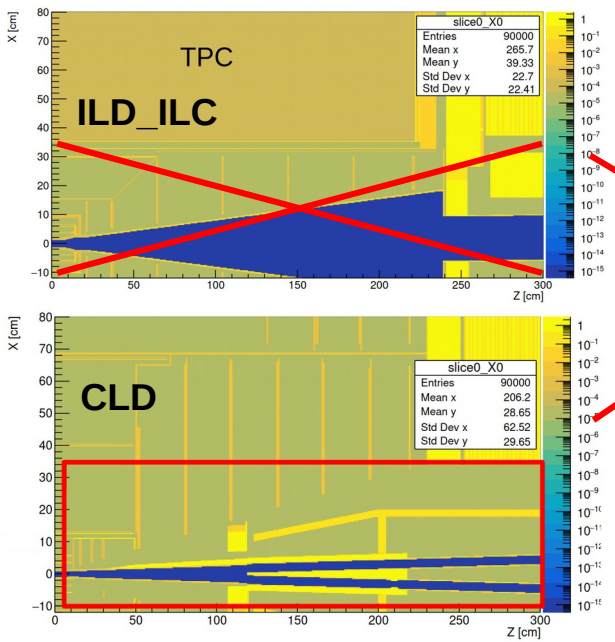


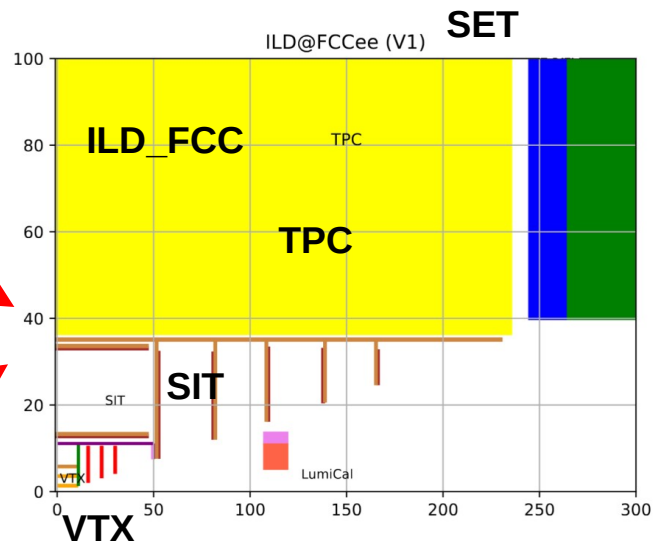
Overview

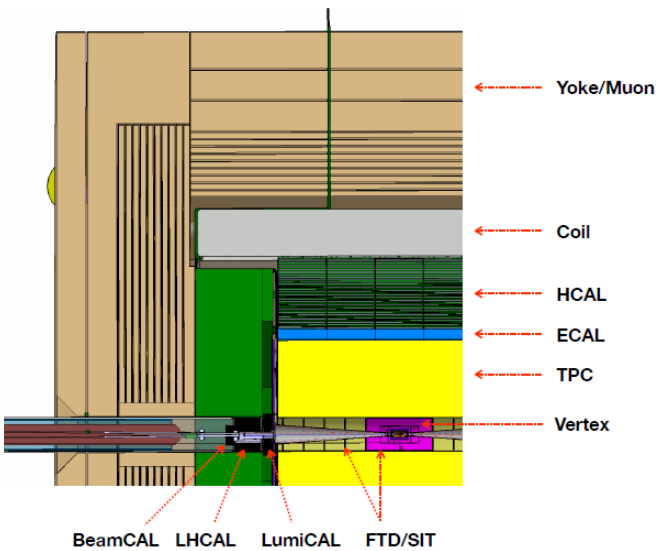
- Detector structure
- Tracking approach
- Issue and fix
- Performance

Detector Structure

- ILD_FCC is a hybrid using ILD_ILC with the CLD MDI and inner tracker



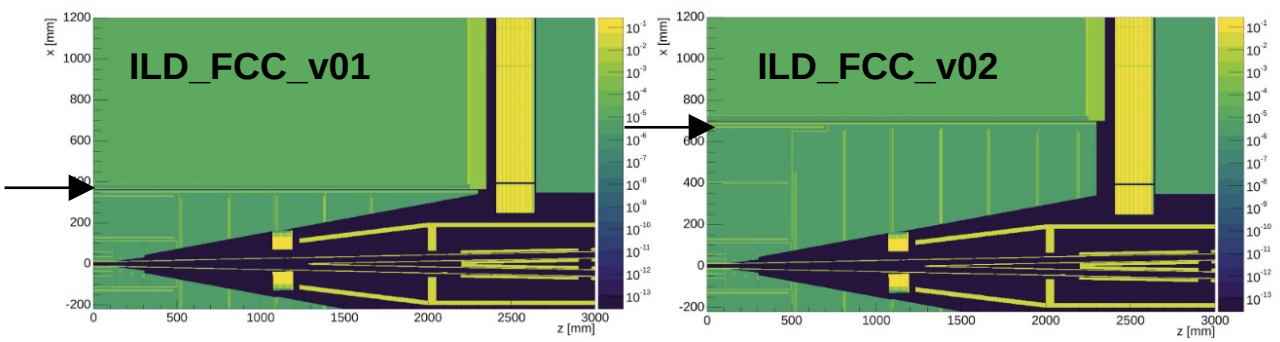




[talk by V. Schwan]

Detector Structure

- Two variants of ILD_FCC: v01 and v02
- CLD inner tracking outer radius is larger than TPC inner radius, so
  - v01: (nearly) full TPC, reduced SIT
  - v02: full SIT, reduced TPC

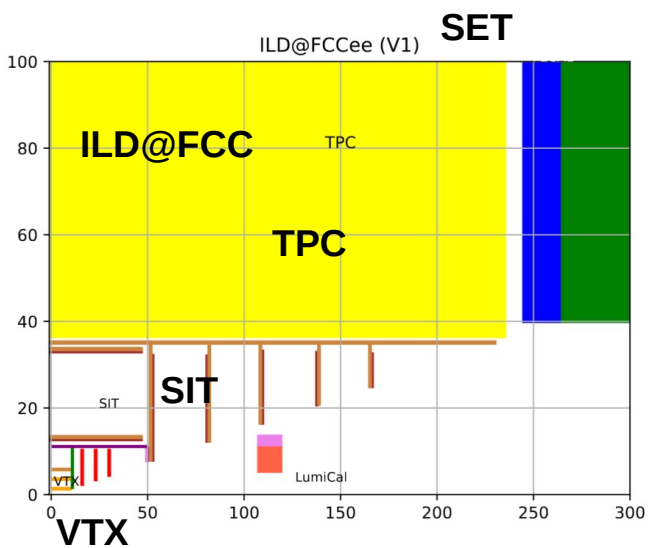


- Comparisons done between
  - ILD at FCC_v01, FCC_02
  - ILD at ILC_3.5T, ILC_2T (each with current nightly software version)
  - official performances from ILD IDR and CLD 2019 report

Tracking Approaches Overview

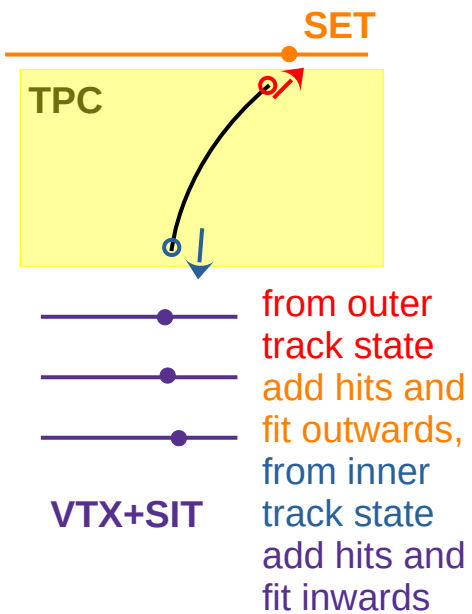
- Old approach, i.e. including with ILD@ILC models in **ILDReconstruction.py**:
  - **Clupatra** for TPC hits
  - **SiliconTracking_MarlinTrk** and **ForwardTracking** for Si hits
  - **FullLDCTracking_MarlinTrk** for combination → highly tailored to ILD@ILC, not suitable for ILD@FCC adaptation

- New approach:
  - **Clupatra** for TPC hits + pick-up of SET hits
  - **ConformalTracking** for inner Si hits (including forward)
  - Dedicated **TrackMerger** to merge



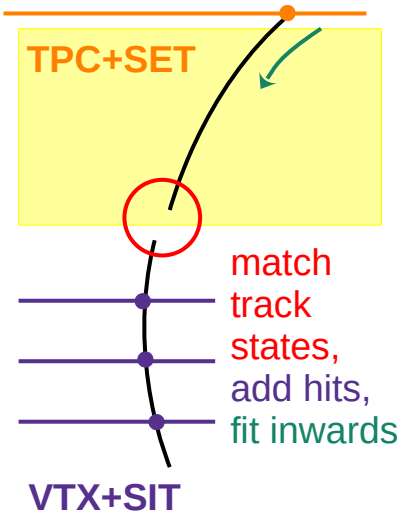
Clupatra

- TPC track finding and fitting
- New: pick up of Si hits, both inwards (VTX, SIT; not used) and outwards (SET)
  - extrapolate tracks from closest track state
  - check if hit is within fixed area (allow additional z uncertainty for strip hits)
  - add hit if additional chi² is within limit (default: 100)
  - re-fit and store new track states



TrackMerger

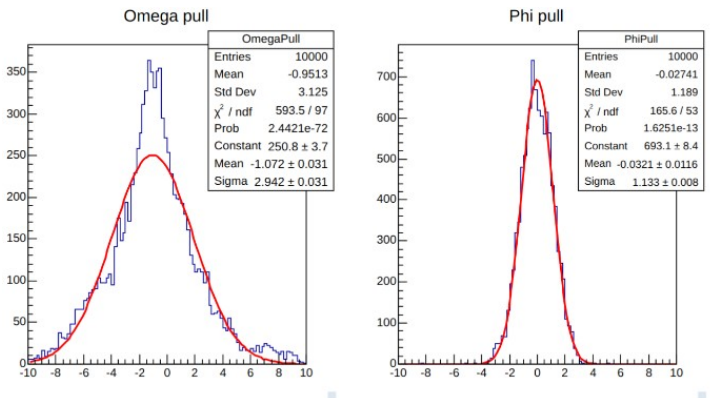
- Takes Clupatra tracks (TPC+SET) and Conformal tracks (VTX+SIT) and combines them
  - check if track states (inner track `AtLastHit`, outer track `AtFirstHit`) match within limits (algorithm parameters)
    → can choose ‘absolute’ or ‘significance’; now rather generous
  - for all matches: add inner hits to outer track, fit outside-in, save `Prob(chi2,ndf)`
  - starting from `Prob(chi2,ndf)` add track combinations to output, remove used tracks
  - input tracks with no match are added to output without modification
    → e.g. forward tracks only in SIT, backscatter only in TPC
  - `warning()` if input tracks remain that match but fit rejects all inner hits (~15 in 10k muons)
- Afterwards use RefitProcessor to refit all tracks



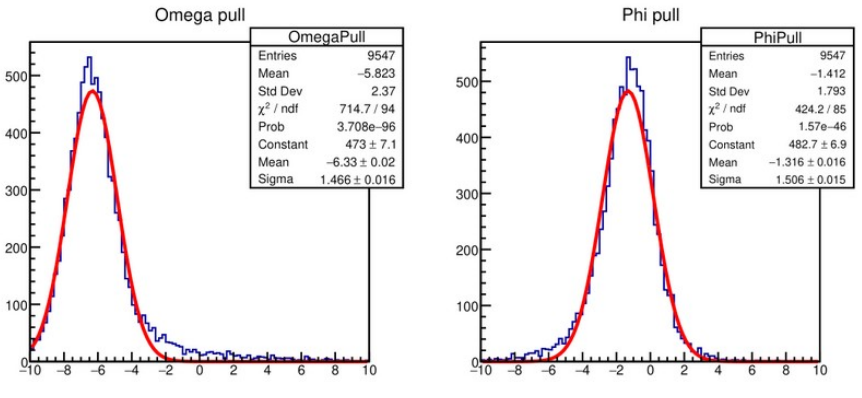
The Issue

- Issue: poor tracking performance, measured as pull biases, primarily Omega

100 GeV muons

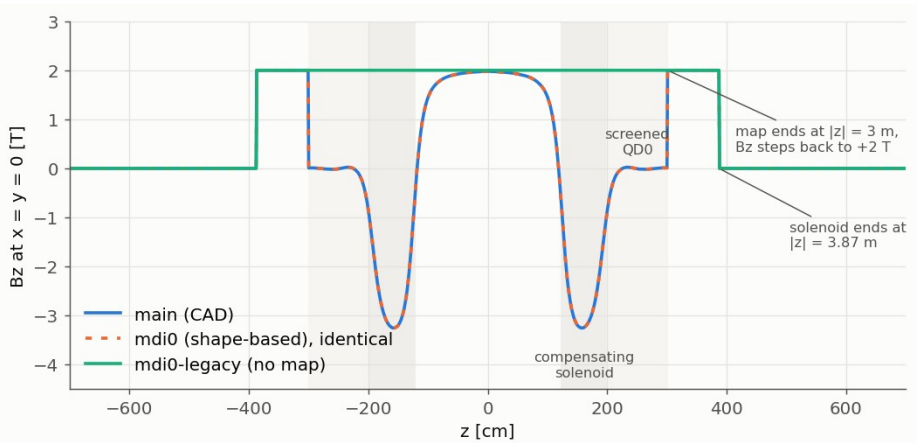


5 GeV muons



Antisolenoid Field

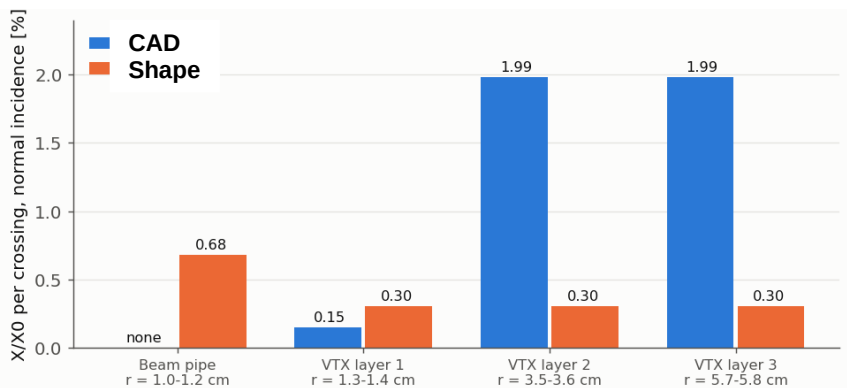
- Culprit found: inhomogeneous antisolenoid field in (full-detail) FCC MDI
  - affecting mostly beam pipe region, but - crucially - changing B at IP from 2 T to 1.979 T
- MarlinTrk not set up to deal with it:
  - assumes homogeneous B field, gets `BFieldAtOrigin`, uses this one field value for propagation and fit in KalTest
  - Tracks in TPC see B_z = 2T but are reconstructed with B_z = 1.979T
  - 2013 addition of `TBField::fUseUniformField`; setting this flag did not resolve issue
- No other detector concept uses it
  - CLD: simplified antisolenoid not affecting B at origin
  - IDEA, ALLEGRO: no antisolenoid at all
  - physics runs done without detailed field map, backgrounds with it
- For ILD: replace detailed field map with simple antisolenoid (á la CLD)



plot: V. Schwan

CAD and Shape-based Beampipe

- New versions added for MDI collections with simple antisolenoid:
  - MDI_o1_CADBased_v02/MDI_o1_CADBased_v02_collection.xml
  - MDI_o1_ShapeBased_v02/MDI_o1_ShapeBased_v02_collection.xml
- CAD-based more detailed than Shape-based but also a lot more compute-intensive
- Victor recently found a substantial material budget inconsistency in CAD-based model:
  - Layer — X/X0 in CAD — X/X0 in Shape
  - Beam pipe — none — 0.68%
  - VTX layer 1 — 0.15% (AlBeMet) — 0.30%
  - VTX layers 2 and 3 — 1.99% (Cu) — 0.30%
  - IT ring at r = 9.5 cm — 36.9% (Cu) — 1.0%
  - Mechanism: TGeoTessellated::Contains returns true inside the hollow pipes, so TGeo treats those points as solid pipe.
  - Confirmed cause: removing only the STL beam-pipe include from main makes all 14,698 sensor surfaces match mdi0.



plot: V. Schwan

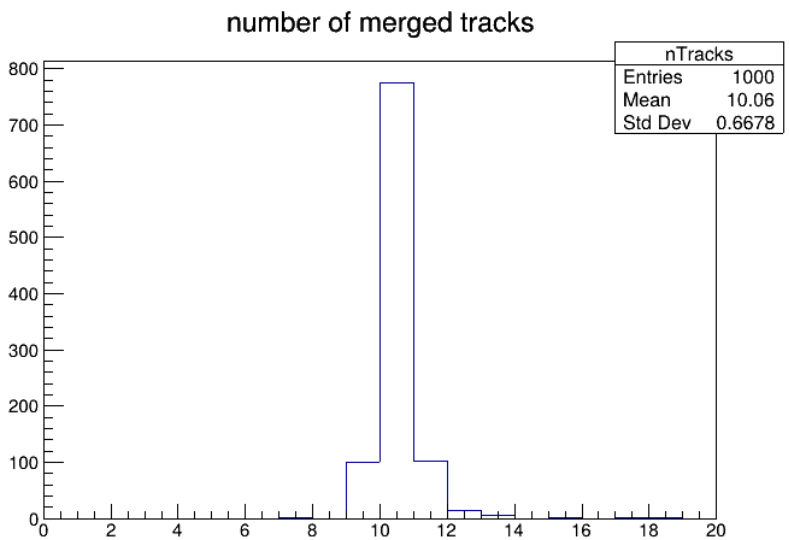
- Created Issue, likely fixed with new root version
- Moved ILD_FCC from CADBased to ShapeBased

Pull Requests

- Clupatra::Clupatra: merged (pick-up SET hits)
- k4RecTracker::TrackMerger: merged (include track fit in merger, forward tracks in output)
- MarlinReco::KinkFinder: merged (minor adjustment to ILD_FCC collections naming)
- k4geo:: CAD-based and Shape-based collections: merged
- ILDConfig: open
  - TrackingReco_FCCeeMDI.py: re-combine Reco, Merging and PostReco; remove multiple track merging options, i.e. only keep the one described above
  - HighLevelReco_FCCeeMDI.py: remove deleted merged track collections
  - ILDReconstruction.py: remove deleted tracking includes

Merging Efficacy

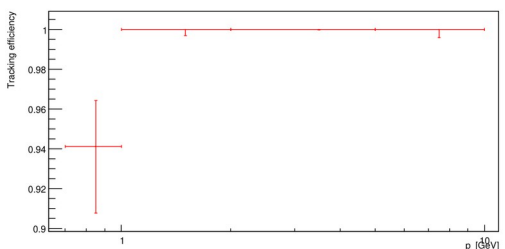
- 1k x 10 muons with 5 GeV each
- unmatched tracks double-count, matched-but-unmerged tracks are missing



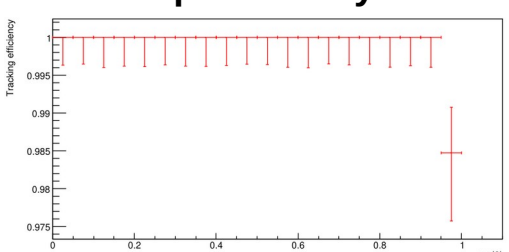
Tracking Efficiency

- 10k 5 GeV muons, isotropic

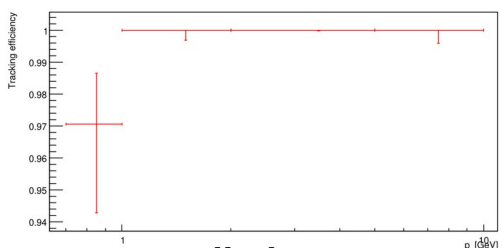
ILD_ILC_3.5T



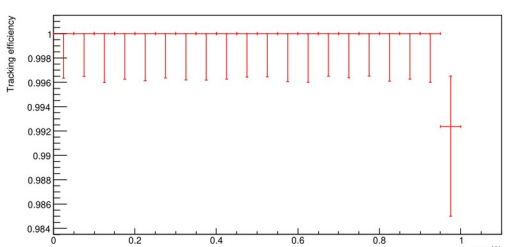
preliminary



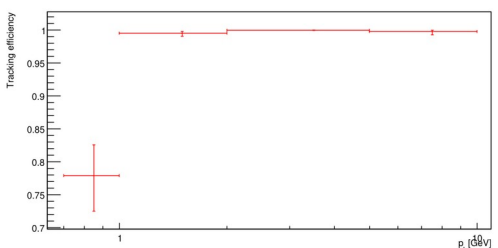
ILD_ILC_2T



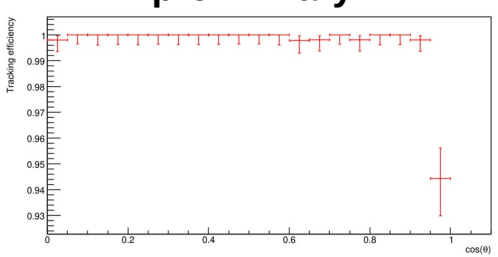
preliminary



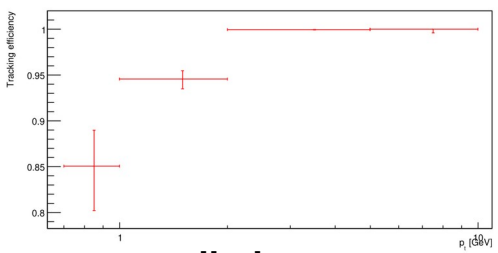
ILD_FCC_v01



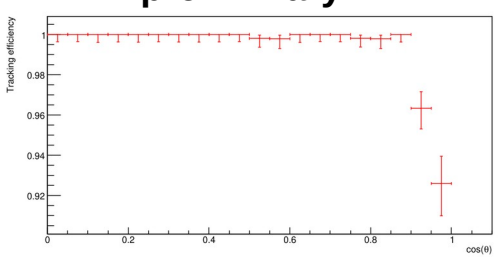
preliminary



ILD_FCC_v02



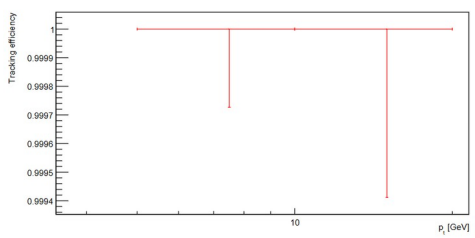
preliminary



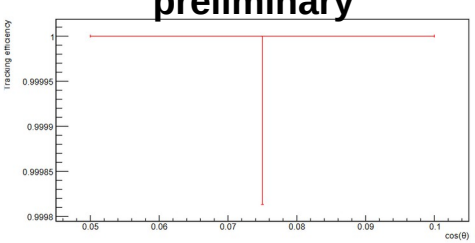
Efficiency Example

- 10k 10 GeV muons, 85° (central barrel)
- ILD_FCC at > 0.997

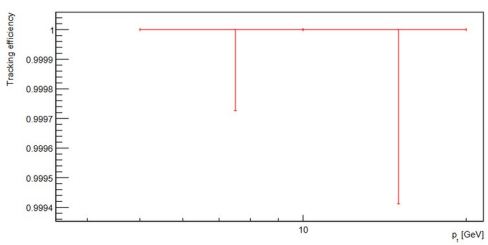
ILD_ILC_3.5T



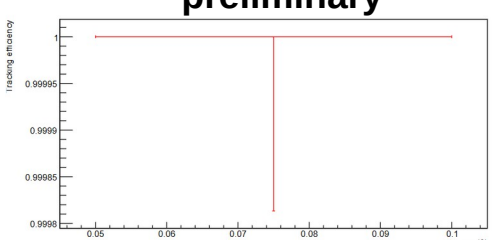
preliminary



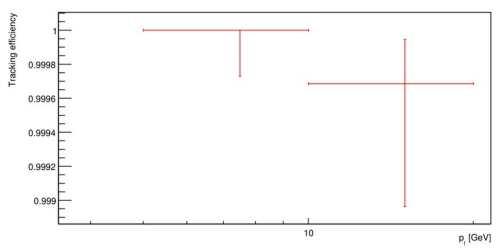
ILD_ILC_2T



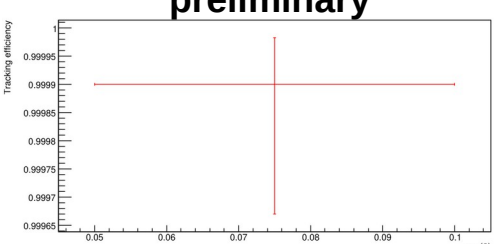
preliminary



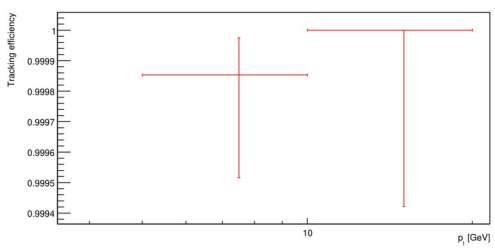
ILD_FCC_v01



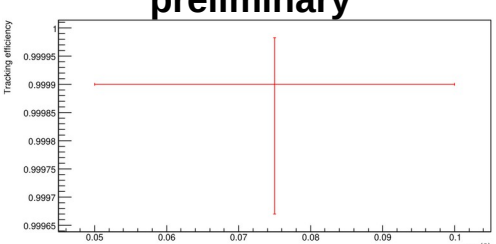
preliminary



ILD_FCC_v02



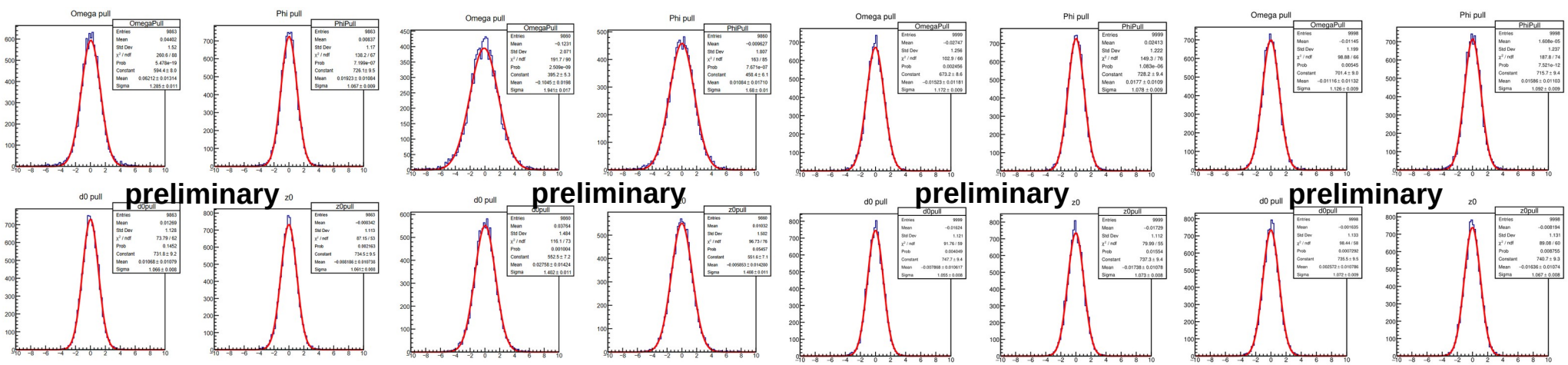
preliminary



Pulls Example

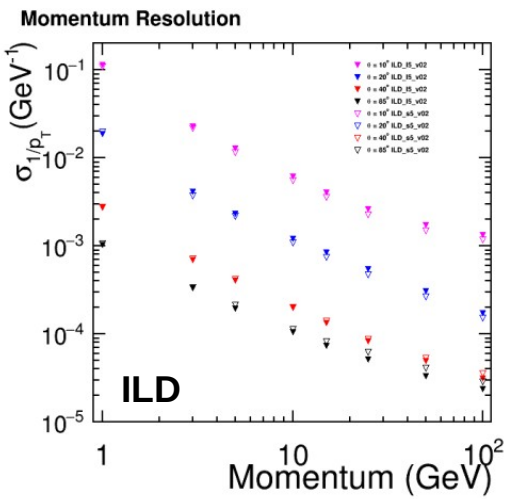
- 10k 10 GeV muons, 85° (central barrel)
- ILD_FCC pulls ~ 1.1; comparable with ILC_3.5T, a bit better for Omega
- ILC_2T: width between 1.4 and 2

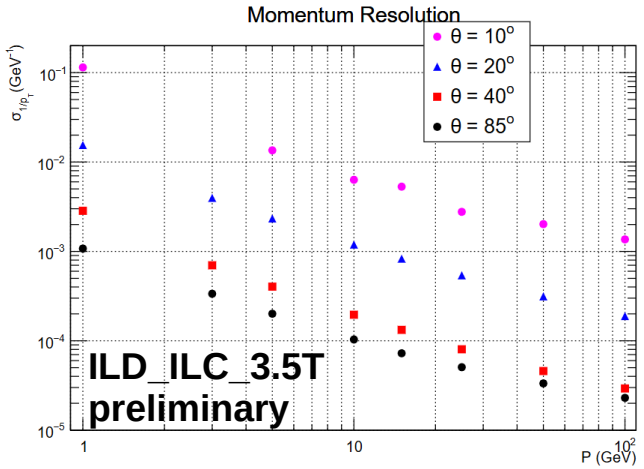
ILD_ILC_3.5T **ILD_ILC_2T** **ILD_FCC_v01** **ILD_FCC_v02**

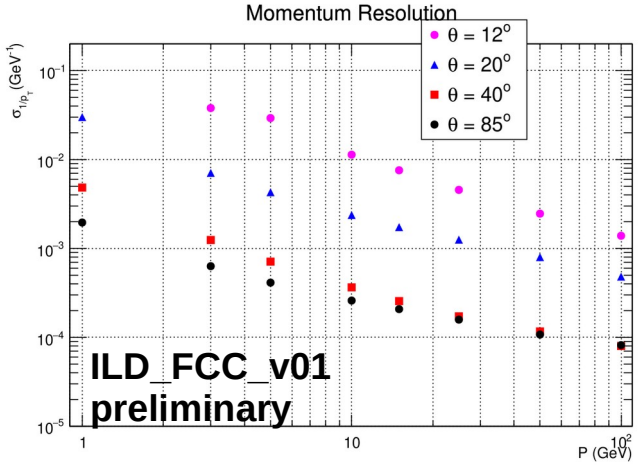


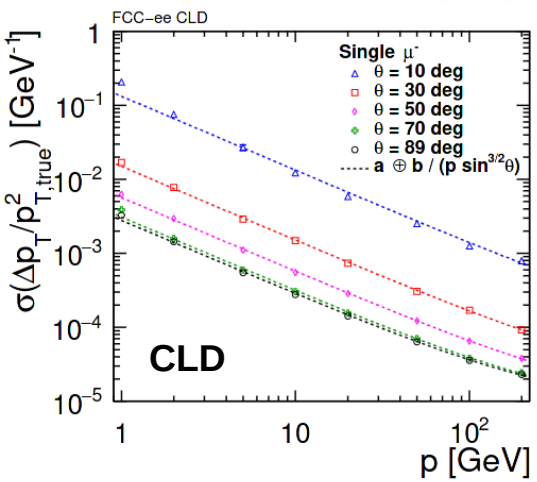
Momentum Resolution - single muons, specific p and θ

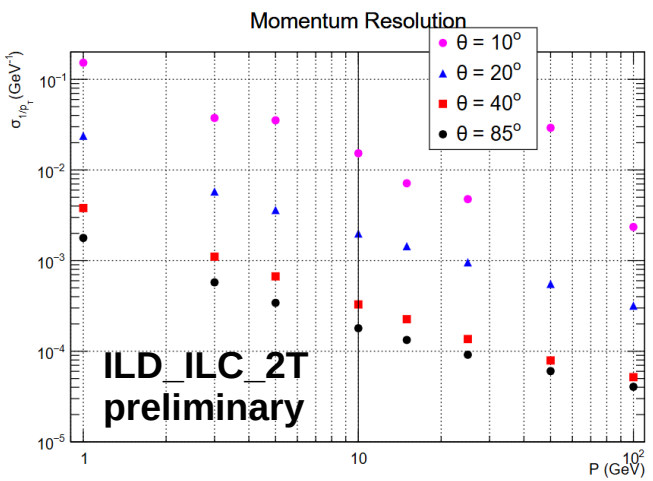
- Considerably degraded resolution - likely due to 2T, more material budget and unoptimised reco

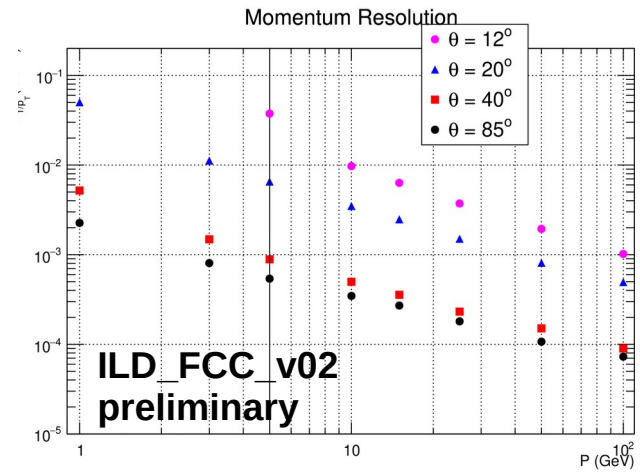






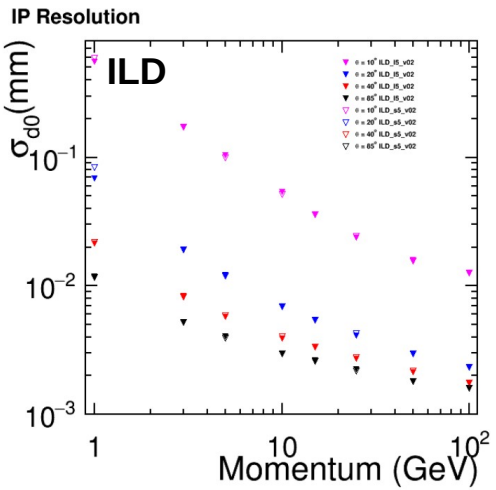


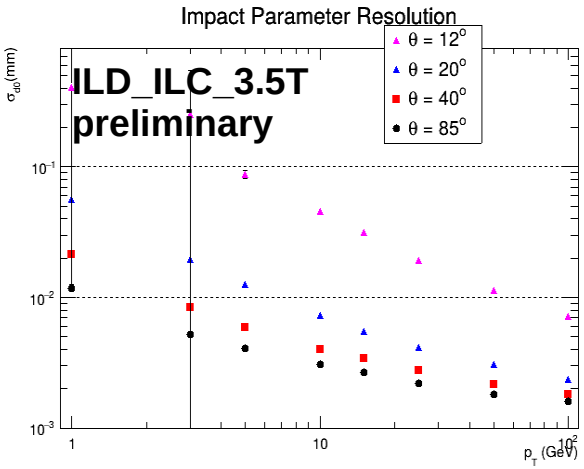


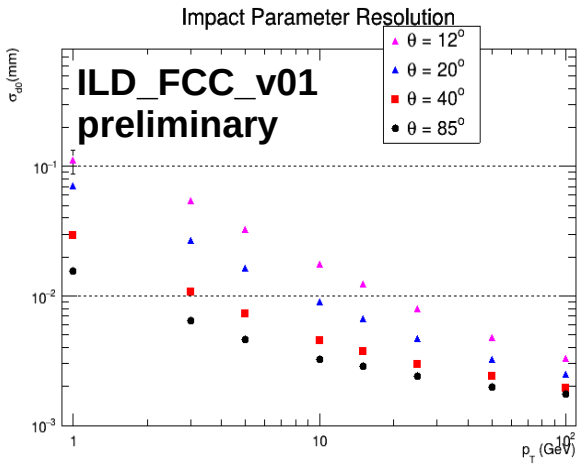


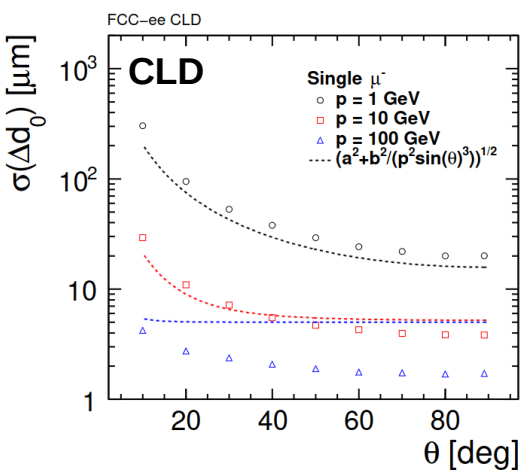
d_0 Resolution - single muons, specific p and θ

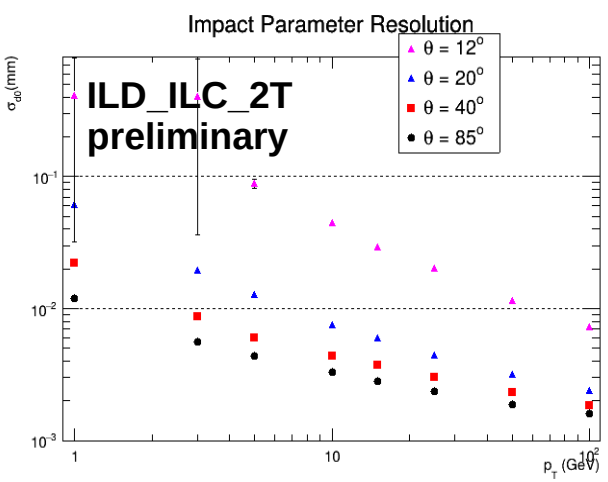
- Comparable resolution

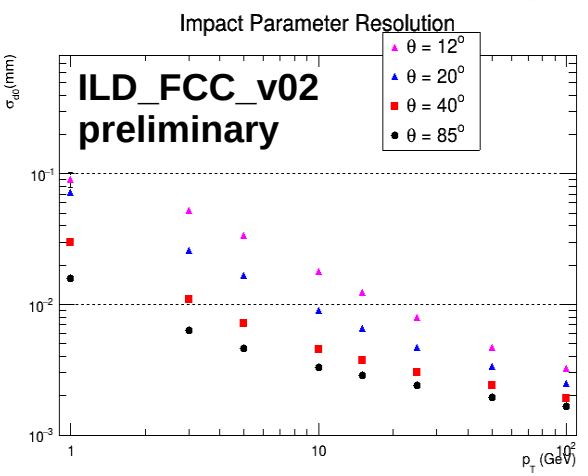






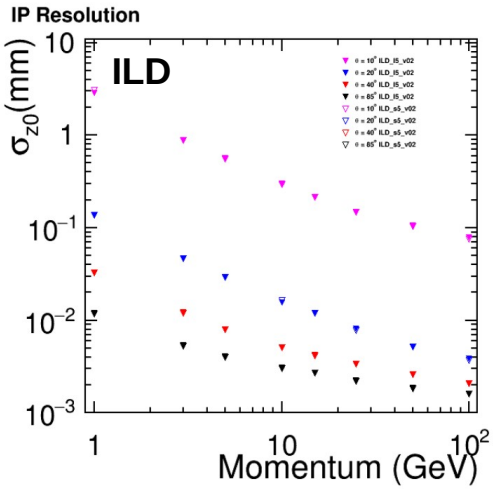


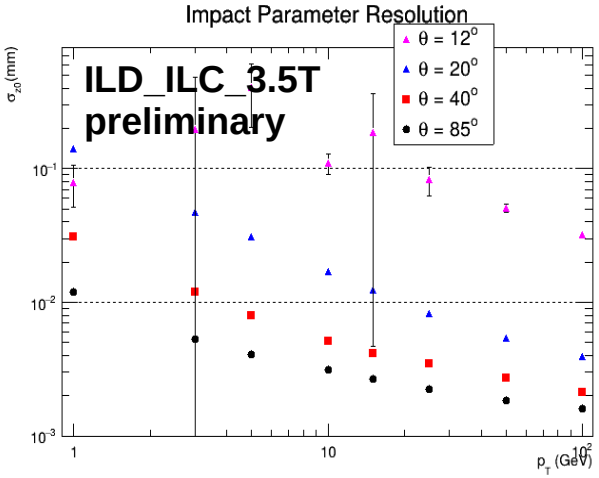


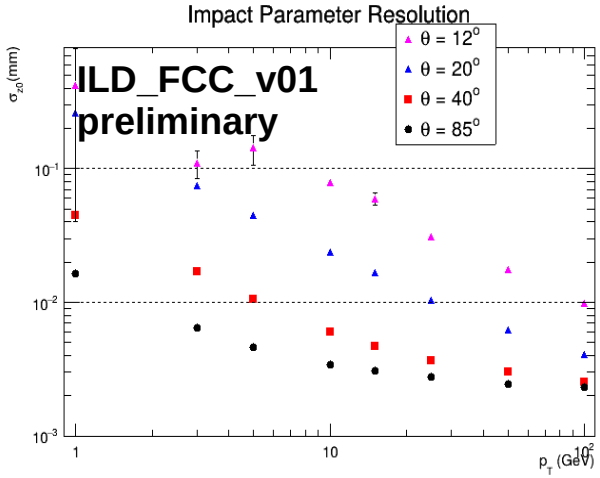


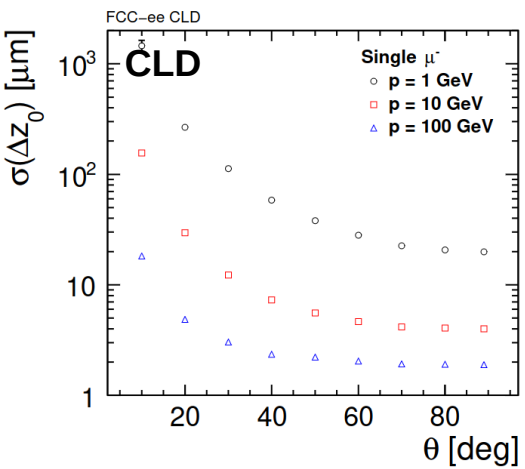
z_0 Resolution - single muons, specific p and θ

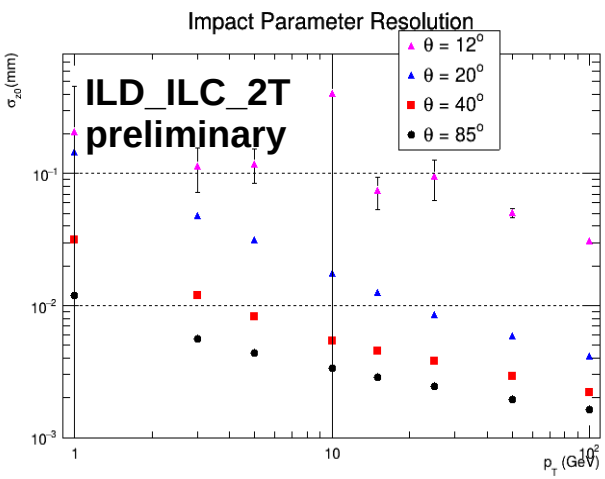
- Comparable resolution

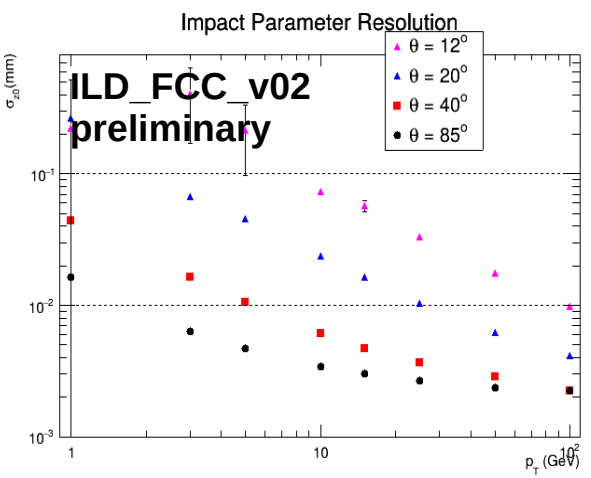






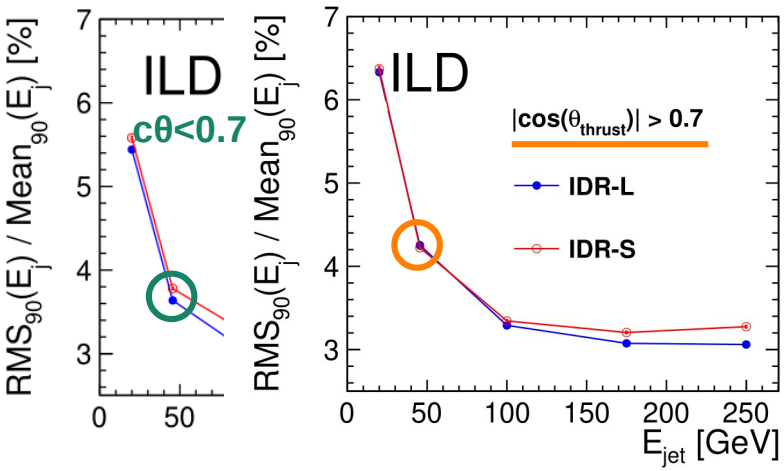


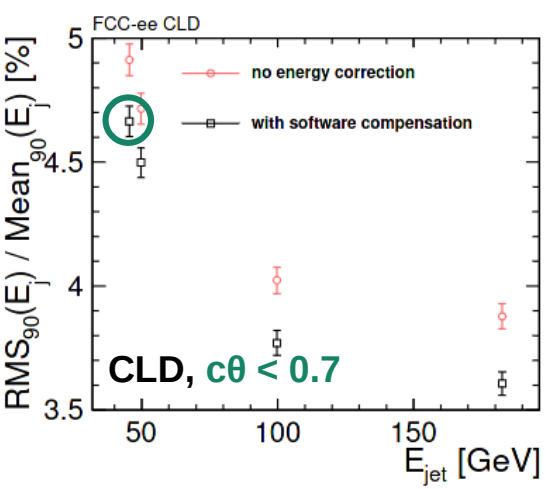


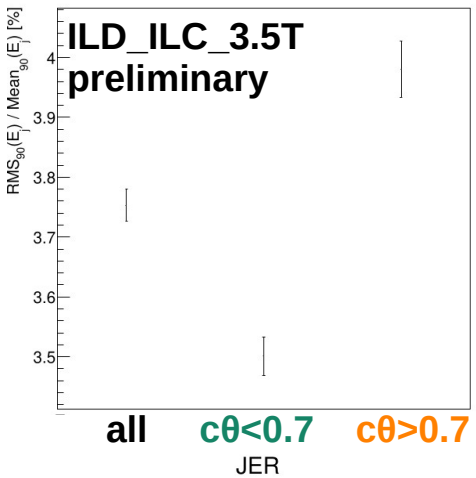


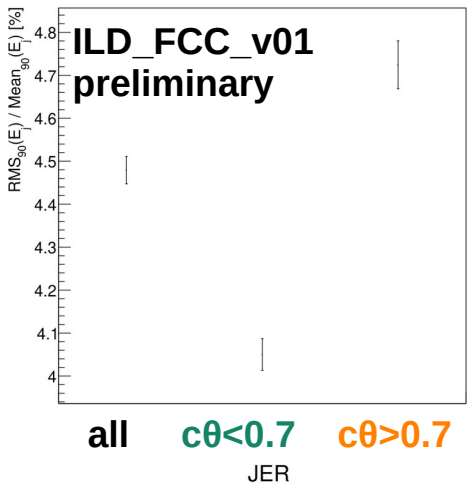
Jet Energy Resolution

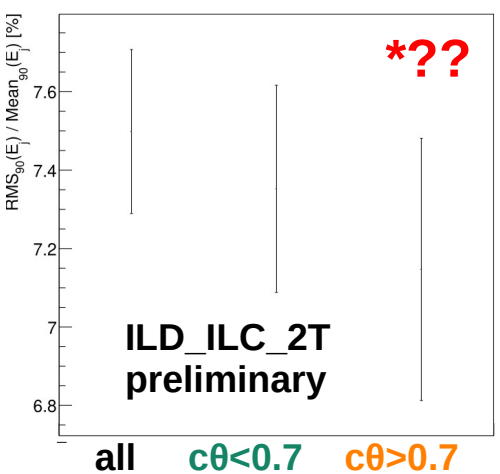
- 91 GeV $q\bar{q}$ with q ∈ {u,d,s}, performance between ILD and CLD

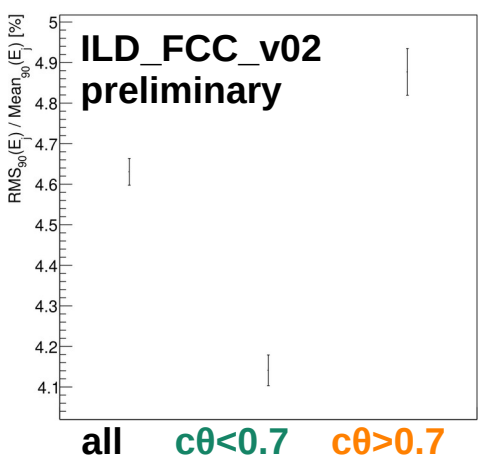












The Timeline

- Sorry for reporting this in the FCC Reco Meeting and merging the code in the framework before reporting here - there was some urgency…

- Oct 28th at FCC Reco Workshop: antisolenoid is discovered as culprit
- Nov 4th (Wed of LCWS rehearsals): confirmation that it's fine to throw the antisolenoid out for now - showstopper gone
- Nov 8th : (soft) deadline for software freeze for central FCC MC physics production, got initial PRs out by then
- following week: presentation at ILD Planning WG Meeting and FCC Reco Meeting to have feedback for the PRs
  since then PR reviews, CI tests and bug fixes resulting in the merges

Central FCC MC Production

- As presented in FCC Physics, Software and Computing Coordination Meeting
- Use iLCDirac and distributed computing to generate physics samples, available on CERN **/eos/experiment/fcc/rucio**
- Z pole ($q\bar{q}$, $\tau\tau$), Higgs recoil, $t\bar{t}$, some more; send requests here
- Originally planned to use CLD in full sim and Delphes, IDEA in Delphes,
  now ILD in full sim is also an option

Tags for the release

Package	Latest tag	Waiting for
Delphes	3.5.2pre02	#217, #218, #219; tag requested in #216
k4SimDelphes	v00-09	#179
KKMCee	2026-11-18	key4hep/KKMCee#2, then a new snapshot
k4Gen	v0.1pre17	#70, #72; #74 (Pythia 8.318) already merged
k4PFHitML	v00-02	#11
CLDConfig	r2026-09-09	#101 (HitPF integration)
k4RecTracker	v0.8.0	#124 (ILD)

- **Ready to tag**: EDM4hep, DD4hep, k4FWCore, k4geo, k4RecCalorimeter, k4ActsTracking, k4Reco, k4GaudiPandora, Clupatra, MarlinReco, LCFIPlus, MarlinTrk, FCC-config
- **Already tagged**: podio v01-10, MarlinTrkProcessors v02-13
- **To check**: k4GeneratorsConfig, ILDConfig, k4SimGeant4

slide by J. Smiesko

Outlook until Software Freeze

- Software still not exactly frozen for the key4hep release
- Decided to ‘package’ ILDConfig now for the release, continue to work on additional to-dos until freeze actually happens:
  - Look into optimisation of TrackMerger parameters for matching and combined fit
  - Add option to enable PfoAnalysis to run on current physics events
  - Run dE/dx calibration and update parameters of Compute_dEdxProcessor (rather than using angular correction in MiniDST.xml
  - Move LeptonID and CPID from MiniDST to StdReco
  - Move additional small algorithms form MiniDST to StdReco?
    - Thrust, Sphere, Fox; WWCategorisation
    - Isolated Electron/Muon/Tau/Photon Tagging
  - Run CPID calibration with updated dE/dx values
- These points are rather ‘nice to have’ than crucial for a production at this point

Summary & Outlook

- Tracking works with a basic performance
- Many possible corners to look for optimisation potential (FullLDCTracking was huge)
  - matching/merging parameters
  - curlers, unusual tracks
- Mid-term: switch entirely to ACTS

- Big question:
  Is the current status and performance reasonable to go into a central MC production?

- (I think yes)