Status of the Full Reconstruction for ILD@FCC

ILD - AnaSoft

12.08.2026

Victor Schwan

victor.schwan@cern.ch

HELMHOLTZ

UHH

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

ILD

FTX

DESY.

Content

- Short history of `ILD@FCC`
- First order solution
- An higher order solution
- Challenges & next steps

Disclaimer:

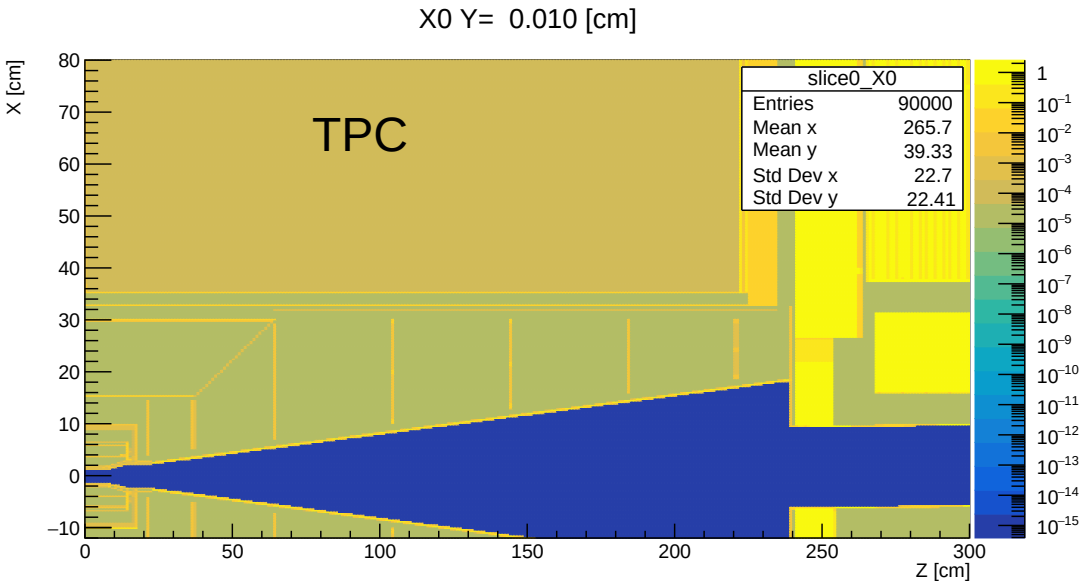
- Several requests for FCC samples → request for this talk
- Publicity within ILD
- Reco for FCC is work-in-progress → very technical talk

Short History of ILD@FCC

ILD Models for FCC: A Pragmatic Approach

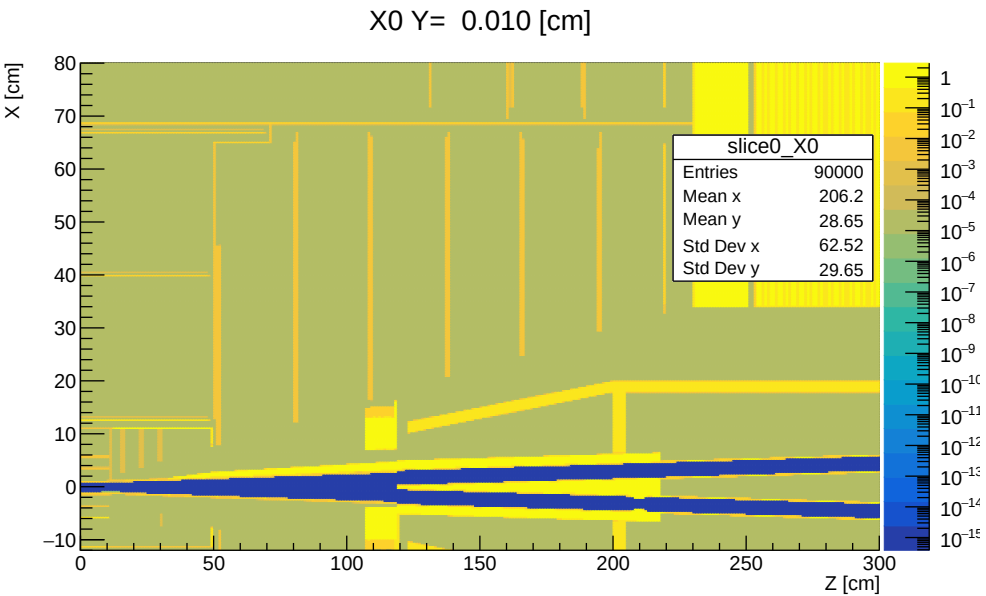
ILD@ILC

- ILC MDI
- Hybrid tracking layout



CLD (CLIC-like Detector)

- FCCee MDI
- All-silicon tracking



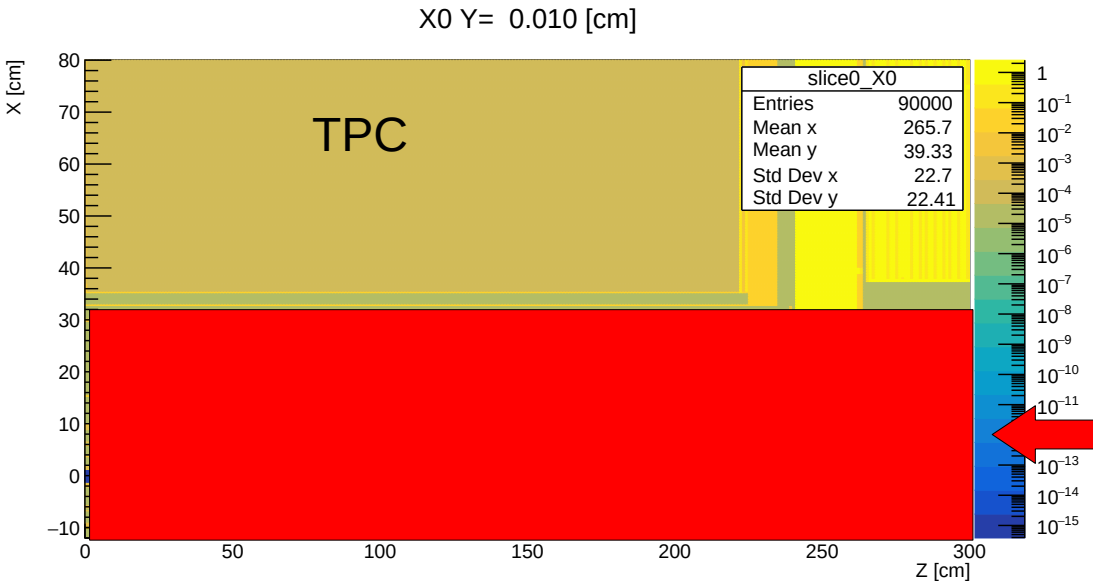
ILD Models for FCC: A Pragmatic Approach

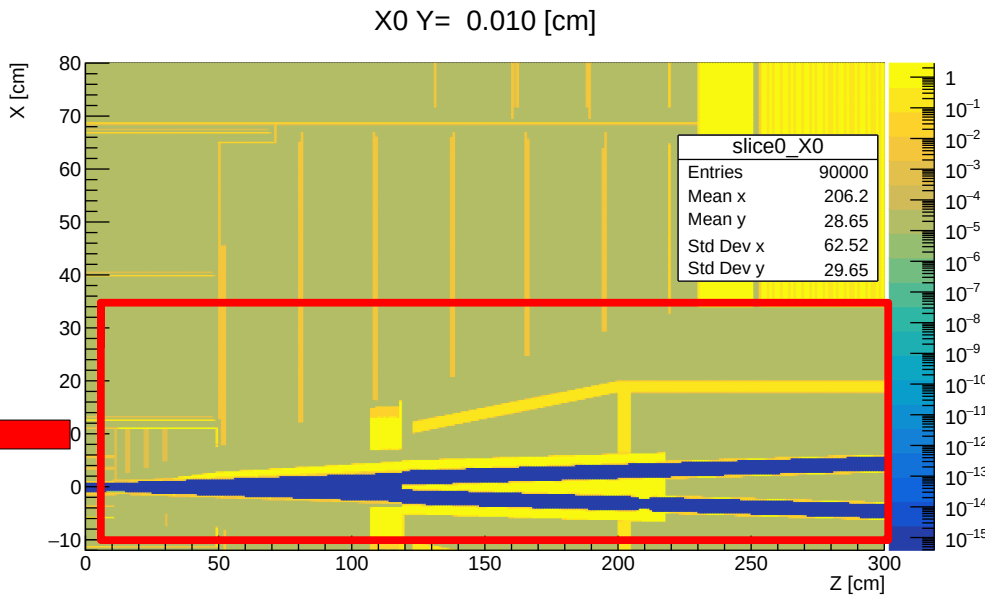
ILD@ILC

- ILC MDI
- Hybrid tracking layout

CLD (CLIC-like Detector)

- FCCee MDI
- All-silicon tracking





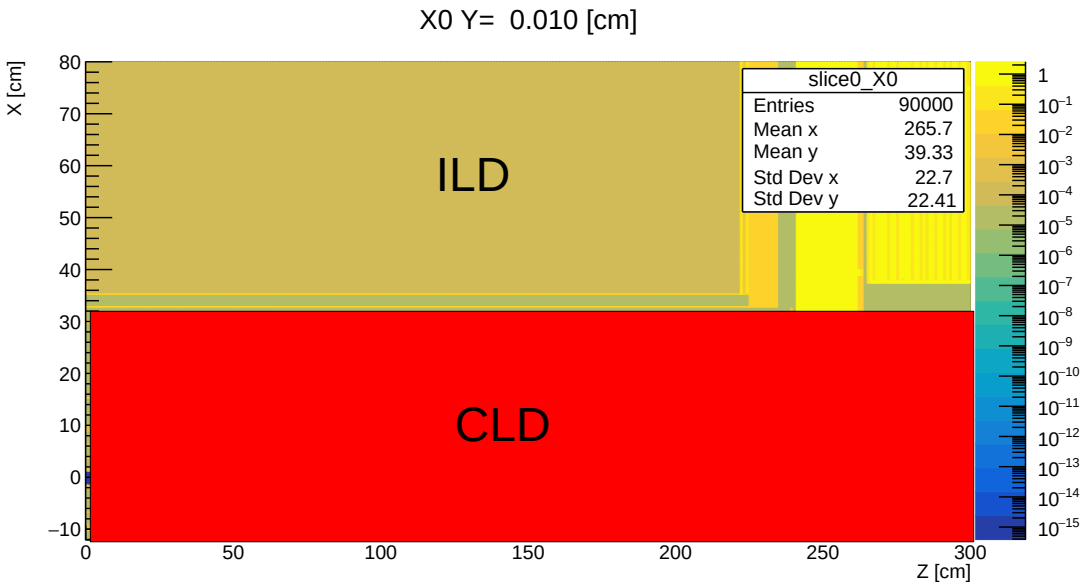
Ansatz:

1) Remove everything inside of the TPC from ILD
2) Replace with CLD subdetectors

ILD Models for FCC: A Pragmatic Approach

ILD@FCC

- FCC MDI
- Hybrid tracking layout
- ILD-CLD-Frankenstein model



Reconstruction for ILD

- Full sim reconstruction pipeline for **`ILD@ILC`** is well-established
  - Hit digitization in trackers & calos
  - Track finding and fitting
  - Particle Flow reconstruction
  - High level reconstruction: PID & flavor-tagging, `MCTruthLinking`, ...
- **`ILD@FCC`**
  - Vertex & Inner Tracker: `ConformalTracking` from CLD
    - Works out-of-the-box for `ILD@FCC`
  - TPC Reconstruction: `Clupatra` (dedicated TPC algorithm)
  - Abandoned ILD's custom `FullLDCTracking` algorithm
    - Combines track data from different subdetectors
    - highly tailored, not adaptable to FCC geometry

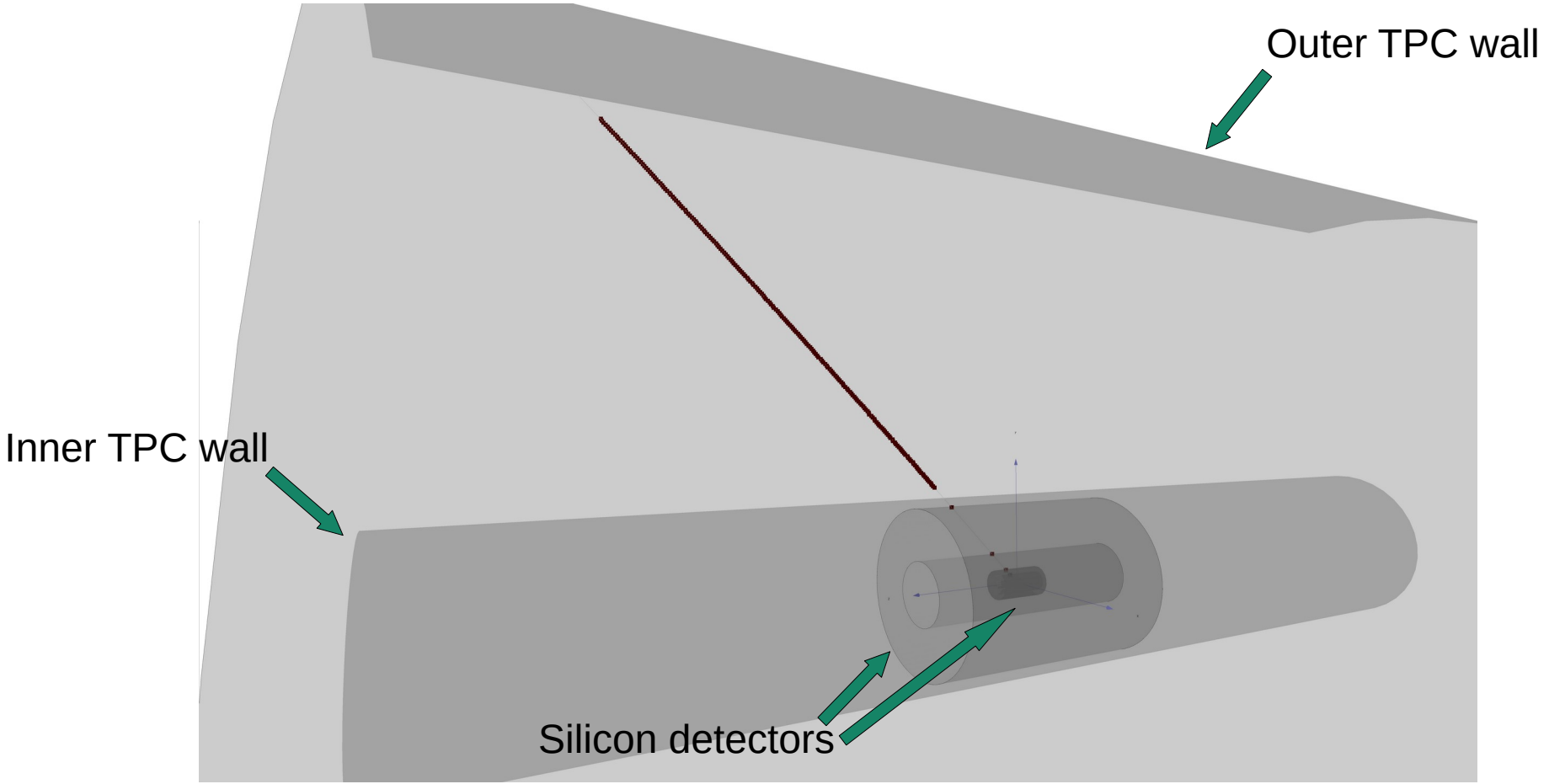
Reconstruction for ILD

- Full sim reconstruction pipeline for **`ILD@ILC`** is well-established
  - Hit digitization in trackers & calos
  - Track finding and fitting
  - Particle Flow reconstruction
  - High level reconstruction: PID & flavor-tagging, `MCTruthLinking`, ...
- **`ILD@FCC`**
  - Vertex & Inner Tracker: `ConformalTracking` from CLD
    - Works out-of-the-box for `ILD@FCC`
  - TPC Reconstruction: `Clupatra` (dedicated TPC algorithm)
  - Abandoned ILD's custom `FullLDCTracking` algorithm
    - Combines track data from different subdetectors
    - highly tailored, not adaptable to FCC geometry

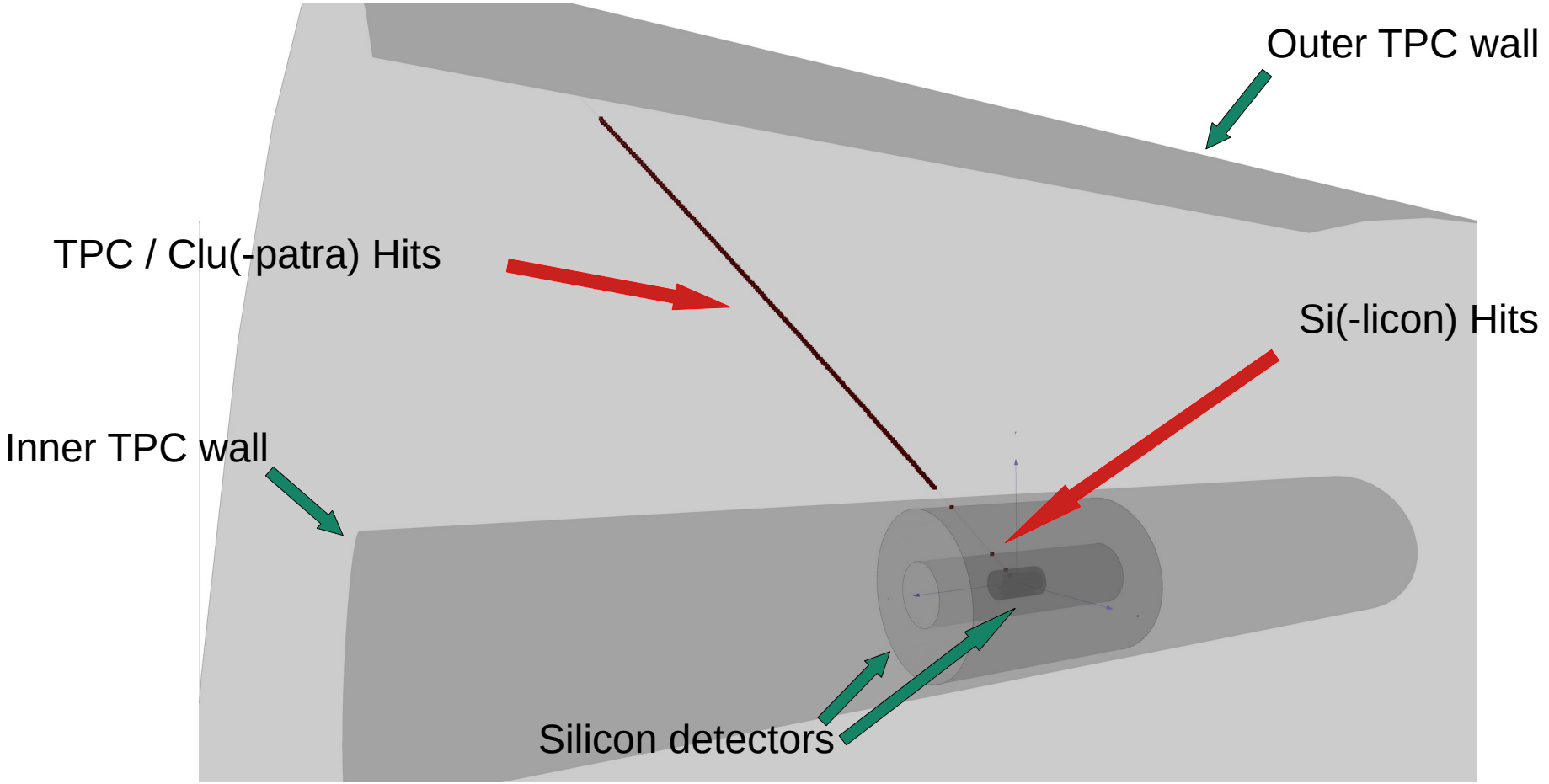
How to merge `Clupatra`(TPC) and `ConformalTracking` (inner trackers)?

First Order Solution: Extrapolate TPC Tracks

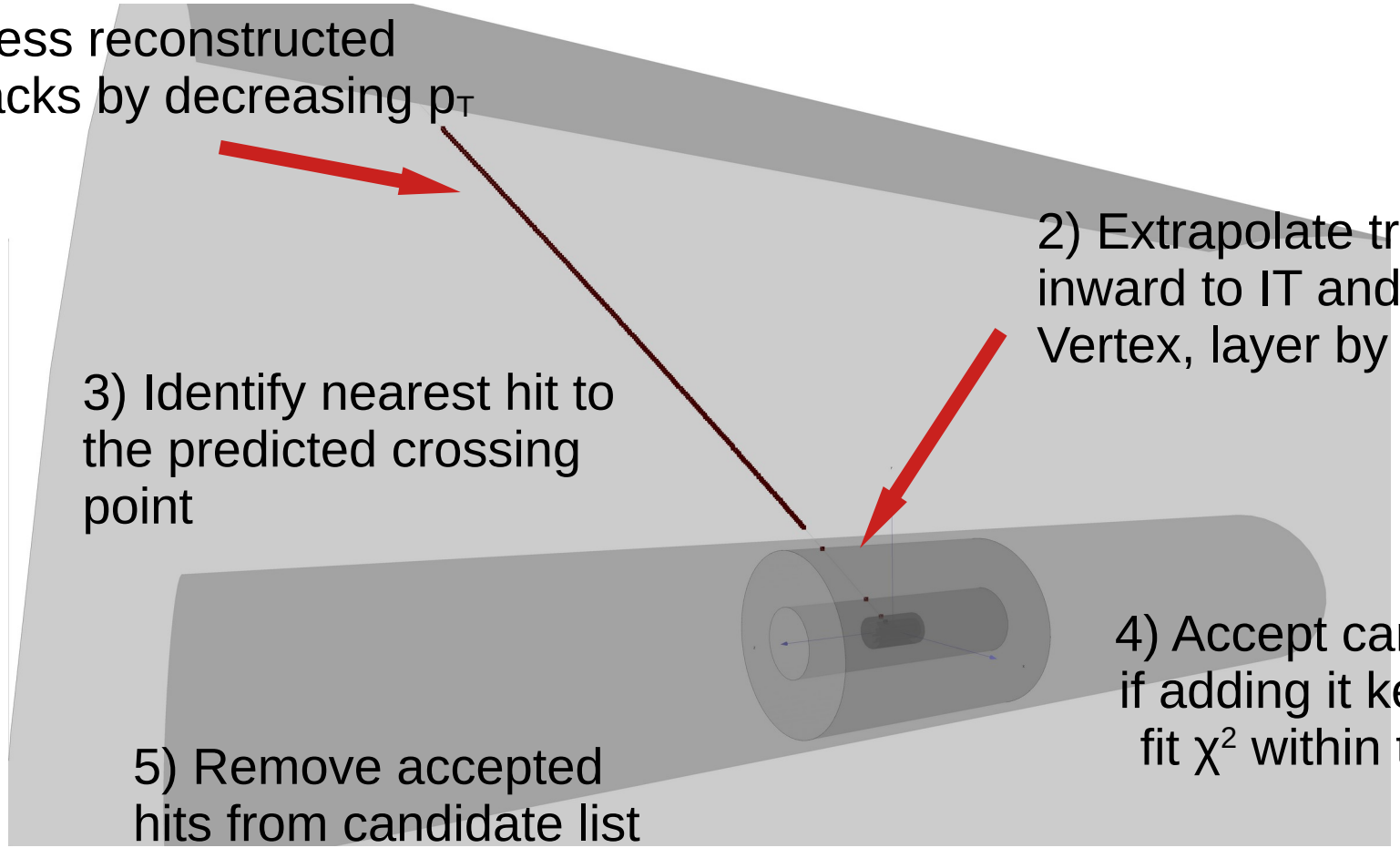
Tracking Detectors



ILD’s Hybrid Tracking System

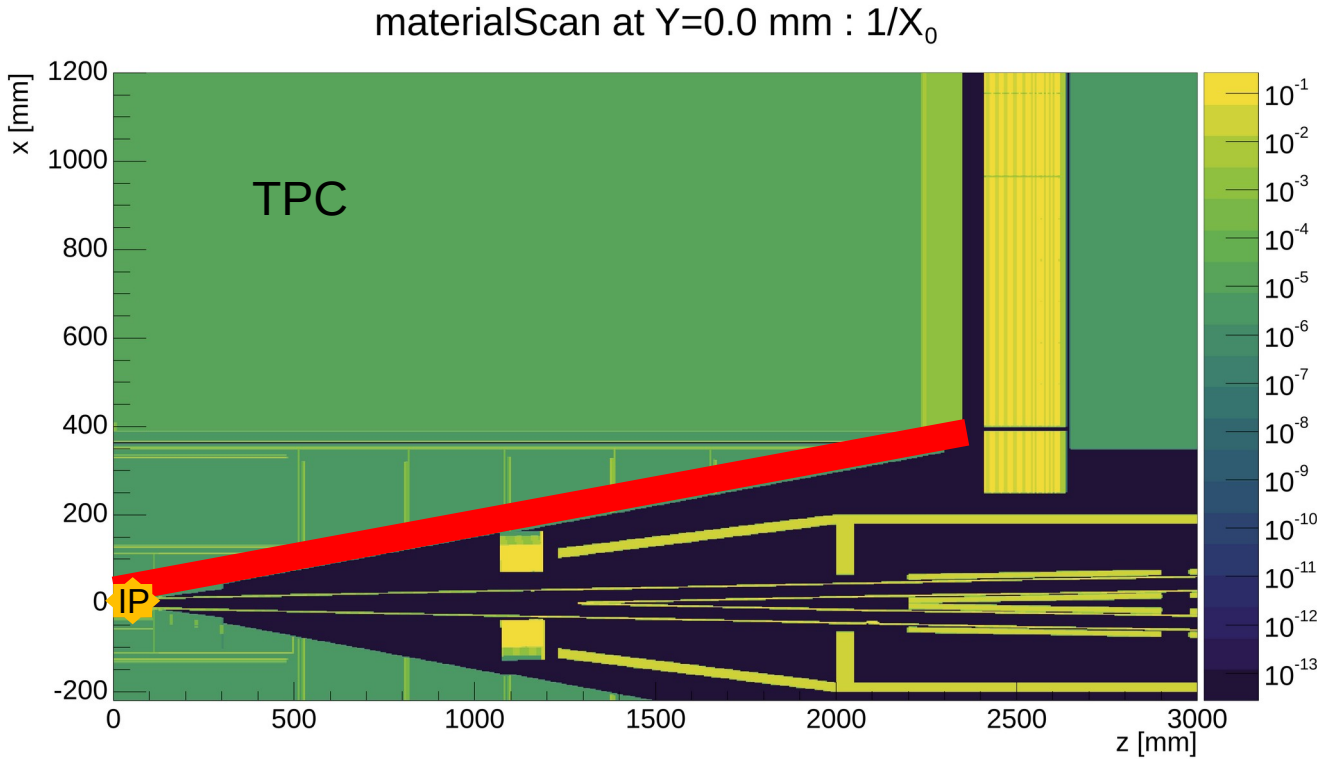


Hybrid Tracking: First Order Solution



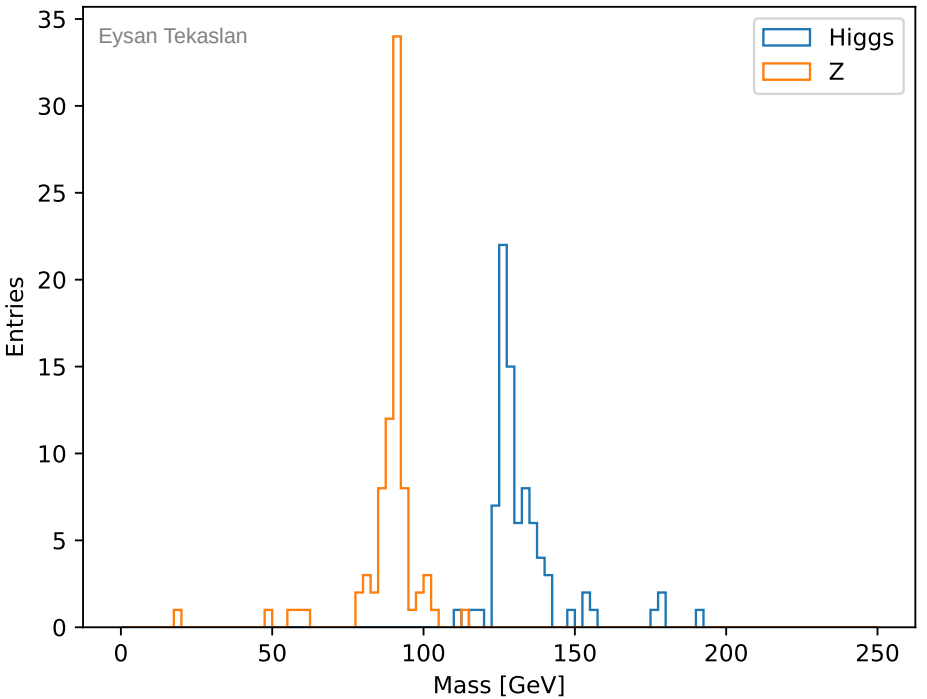
Implications of Current Track Reconstruction

- *Experimental* `Clupatra` option
- Picking up silicon hits relies on known order of silicon layers
  - Given only in barrel region → not trivial for endcaps
- Approach requires at least ~3 hits in the TPC
- Very forward region (opening angle ~9°) has no TPC activity
  - Majority anyways not in geometric acceptance
  - Fundamental limitation of this strategy



Higgs Recoil Mass with First Order Solution for ILD@FCC

- **Full reconstruction pipeline** including `PandoraPFO` is **operational**
  - Analysis code development unblocked
- Caveats: currently tracks have no `subTracks`



Next Steps

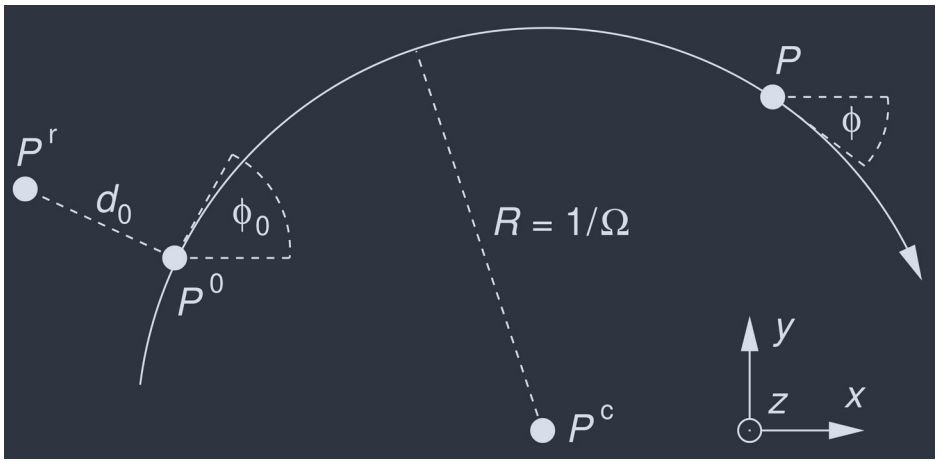
- Plots of efficiencies, resolutions, …
  - Finish adapting `ILDPerformance` to `key4hep/Gaudi` & `ILD@FCC`
    - Requires `subdetectorHits` to be filled
      - Fill by refitting with `RefitProcessor`
        - Adapt `RefitProcessor` to `ILD@FCC`: WIP

Higher Order Solution: Track Parameter-based Merging

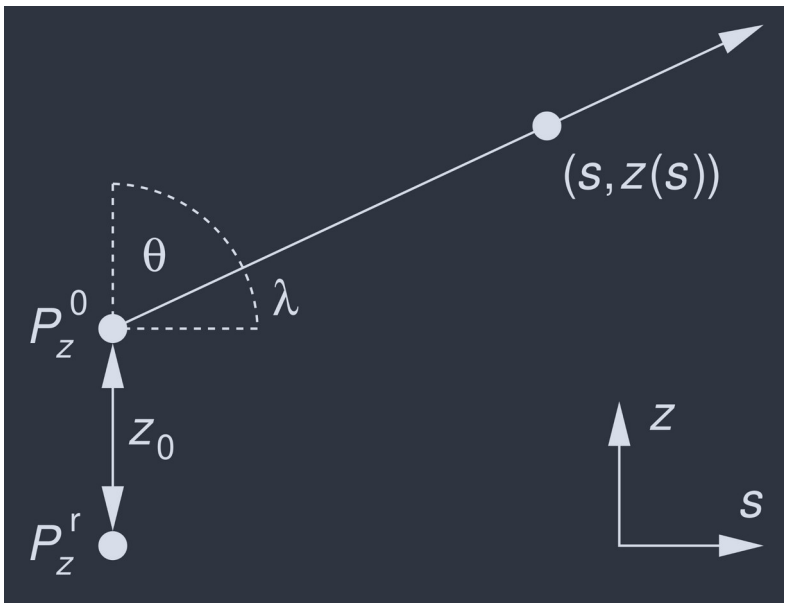
Track Parameters in LCIO/edm4hep

Reference point + five so-called ‘track parameters’: Ω, φ_0, d_0, z_0 and $\tan\lambda$

Projection in xy-plane



Projection in the sz-plane



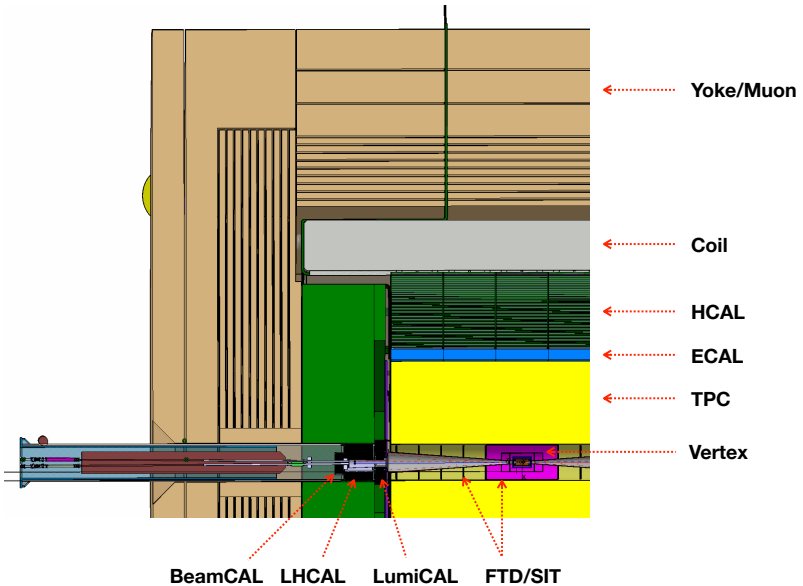
The Track Merger

- Idea: Merge tracks based on compatibility of their params
  - e.g.: (Δd_0 <= .5 && Δz_0 <= 2.5)
- `Gaudi`-native code available in `k4RecTracker` (integrated in `key4hep` nightlies)
- Completely geometry-agnostic!
- `TrackMerger` does only track finding
  - Refit with FCC-adapted `RefitProcessor` - WIP
- TODO:
  - Adapt `ILDPerformance` and benchmark solutions
  - Optimize parameter thresholds

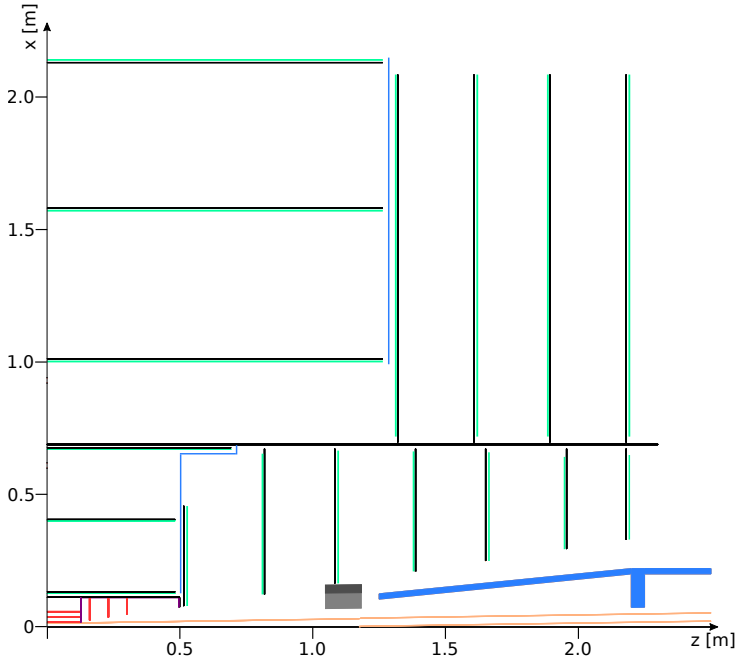
Challenges in Adapting Processor for ILD@FCC

Challenges

- Different silicon subdetector layout `ILD@ILC` vs CLD



- Vertex Detector (barrel-like)
- Silicon Inner Tracker (barrel-like)
- Forward Tracking Det. (endcap-like)
- TPC



- Vertex Detector: Barrel + Endcap
- Inner Tracker: Barrel + Endcap
- Outer Tracker: Barrel + Endcap (not `ILD@FCC`)

Challenges

- Different silicon subdetector layout ILD@ILC vs CLD

- Vertex Detector (barrel-like)
- Silicon Inner Tracker (barrel-like)
- Forward Tracking Det. (endcap-like)
- TPC

- Vertex Detector: Barrel + Endcap
- Inner Tracker: Barrel + Endcap
- Outer Tracker: Barrel + Endcap

Unclear mapping:

{VertexBarrel, VertexEndcap, InnerTrackerEndcap} → {VXD, FTD}

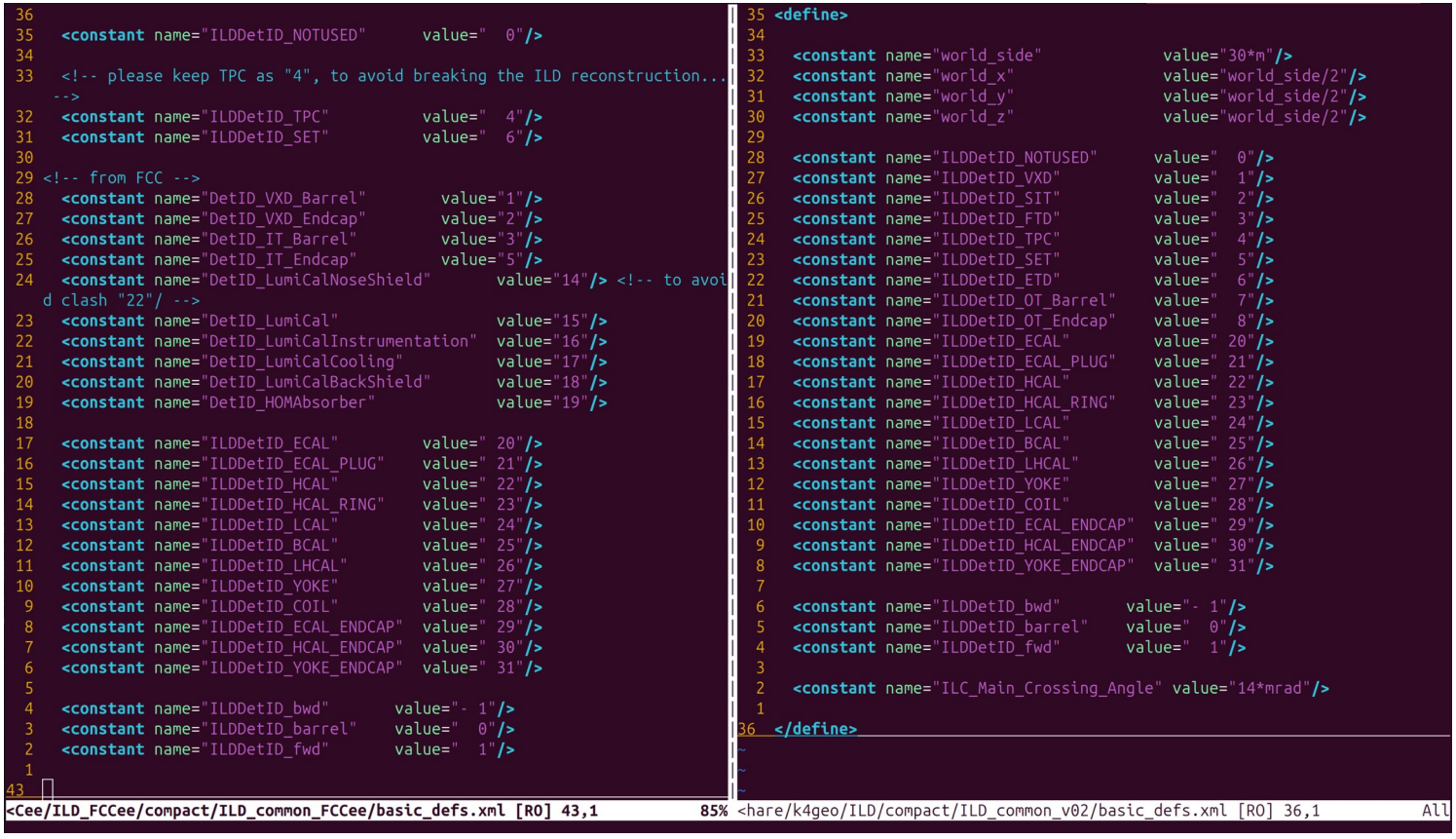
- VXD → VertexBarrel
- FTD → InnerTrackerEndcap

Potential idea:

{VertexEndcap, InnerTrackerEndcap} → {fakeFTD} (shift IT layer ID by nLayerVXD)

Challenges

- Different subdetector arrangement, subdetector names and subdetector IDs



```
36
35   <constant name="ILDDetID_NOTUSED"      value="  0"/>
34
33   <!-- please keep TPC as "4", to avoid breaking the ILD reconstruction...
     -->
32   <constant name="ILDDetID_TPC"          value="  4"/>
31   <constant name="ILDDetID_SET"          value="  6"/>
30
29 <!-- from FCC -->
28   <constant name="DetID_VXD_Barrel"        value="1"/>
27   <constant name="DetID_VXD_Endcap"        value="2"/>
26   <constant name="DetID_IT_Barrel"         value="3"/>
25   <constant name="DetID_IT_Endcap"         value="5"/>
24   <constant name="DetID_LumiCalNoseShield"       value="14"/> <!-- to avoi
     d clash "22"/ -->
23   <constant name="DetID_LumiCal"                 value="15"/>
22   <constant name="DetID_LumiCalInstrumentation"  value="16"/>
21   <constant name="DetID_LumiCalCooling"          value="17"/>
20   <constant name="DetID_LumiCalBackShield"       value="18"/>
19   <constant name="DetID_HOMAbsorber"             value="19"/>
18
17   <constant name="ILDDetID_ECAL"         value=" 20"/>
16   <constant name="ILDDetID_ECAL_PLUG"    value=" 21"/>
15   <constant name="ILDDetID_HCAL"         value=" 22"/>
14   <constant name="ILDDetID_HCAL_RING"    value=" 23"/>
13   <constant name="ILDDetID_LCAL"         value=" 24"/>
12   <constant name="ILDDetID_BCAL"         value=" 25"/>
11   <constant name="ILDDetID_LHCAL"        value=" 26"/>
10   <constant name="ILDDetID_YOKE"         value=" 27"/>
 9   <constant name="ILDDetID_COIL"         value=" 28"/>
 8   <constant name="ILDDetID_ECAL_ENDCAP"  value=" 29"/>
 7   <constant name="ILDDetID_HCAL_ENDCAP"  value=" 30"/>
 6   <constant name="ILDDetID_YOKE_ENDCAP"  value=" 31"/>
 5
 4   <constant name="ILDDetID_bwd"       value="- 1"/>
 3   <constant name="ILDDetID_barrel"    value="  0"/>
 2   <constant name="ILDDetID_fwd"       value="  1"/>
 1
43
<Cee/ILD_FCCee/compact/ILD_common_FCCee/basic_defs.xml [RO] 43,1          85%
```

```
35 <define>
34
33   <constant name="world_side"             value="30*m"/>
32   <constant name="world_x"                value="world_side/2"/>
31   <constant name="world_y"                value="world_side/2"/>
30   <constant name="world_z"                value="world_side/2"/>
29
28   <constant name="ILDDetID_NOTUSED"      value="  0"/>
27   <constant name="ILDDetID_VXD"          value="  1"/>
26   <constant name="ILDDetID_SIT"          value="  2"/>
25   <constant name="ILDDetID_FTD"          value="  3"/>
24   <constant name="ILDDetID_TPC"          value="  4"/>
23   <constant name="ILDDetID_SET"          value="  5"/>
22   <constant name="ILDDetID_ETD"          value="  6"/>
21   <constant name="ILDDetID_OT_Barrel"    value="  7"/>
20   <constant name="ILDDetID_OT_Endcap"    value="  8"/>
19   <constant name="ILDDetID_ECAL"         value=" 20"/>
18   <constant name="ILDDetID_ECAL_PLUG"    value=" 21"/>
17   <constant name="ILDDetID_HCAL"         value=" 22"/>
16   <constant name="ILDDetID_HCAL_RING"    value=" 23"/>
15   <constant name="ILDDetID_LCAL"         value=" 24"/>
14   <constant name="ILDDetID_BCAL"         value=" 25"/>
13   <constant name="ILDDetID_LHCAL"        value=" 26"/>
12   <constant name="ILDDetID_YOKE"         value=" 27"/>
11   <constant name="ILDDetID_COIL"         value=" 28"/>
10   <constant name="ILDDetID_ECAL_ENDCAP"  value=" 29"/>
 9   <constant name="ILDDetID_HCAL_ENDCAP"  value=" 30"/>
 8   <constant name="ILDDetID_YOKE_ENDCAP"  value=" 31"/>
 7
 6   <constant name="ILDDetID_bwd"       value="- 1"/>
 5   <constant name="ILDDetID_barrel"    value="  0"/>
 4   <constant name="ILDDetID_fwd"       value="  1"/>
 3
 2   <constant name="ILC_Main_Crossing_Angle" value="14*mrad"/>
 1
36 </define>
~
~
~
<hare/k4geo/ILD/compact/ILD_common_v02/basic_defs.xml [RO] 36,1           All
```

Challenges

- Different subdetector *geometry*, subdetector *names* and subdetector *IDs*

- Different names and IDs have to be changed in many place
  - Steering files (e.g. `{ILDConfig}/Tracking/TrackingDigi_FCC.py`)
  - Hardcoded in source code
    - e.g. IDs as magic numbers in `subdetectorHits` calls
    - Must be made available as processor parameters
- Silent failures if collections not available

→ If you implement a new processor be it Marlin or Gaudi, keep the different geometries in mind!!!!!

Challenges

- Steering files in `ILDConfig` and corresponding processor source code to be checked:
  - `Tracking/TrackingDigi_FCCeeMDI.py` (mostly done)
  - `Tracking/TrackingReco_FCCeeMDI.py`
  - `HighLevelReco/HighLevelReco_FCCee.py`
  - `ParticleFlow/PandoraPFA.py` (FCC version?)
  - `ParticleFlow/PandoraPFAPerfect.py` (FCC version?)

Summary

- `ILD@FCC` is an ILD-CLD-Frankstein model
- Key challenge is merging the tracks from inner silicon dets and TPC
- First order solution: Extrapolate TPC into barrel-like silicon dets and pick up hits
- Second order solution: Merge tracks based on track-param compatibility (soon!)
- Next steps:
  - Adapt `RefitProcessor`
  - Benchmark tracking approaches with `ILDPerformance`
  - Optimize parameters of both track merging approaches
  - Port analyses from `ILD@ILC` to `ILD@FCC`
    - check specialized tracking procs, particle flow & high-level reconstruction processor by processor

Entry point for starting with `ILD@FCC`: fcc-config

Summary

- `ILD@FCC` is an ILD-CLD-Frankstein model
- Key challenge is merging the tracks from inner silicon dets and TPC
- First order solution: Extrapolate TPC into barrel-like silicon dets and pick up hits
- Second order solution: Merge tracks based on track-param compatibility (soon!)
- Next steps:
  - Adapt `RefitProcessor`
  - Benchmark tracking approaches with `ILDPerformance`
  - Optimize parameters of both track merging approaches
  - Port analyses from `ILD@ILC` to `ILD@FCC` contribute
    - check specialized tracking procs, particle flow & high-level reconstruction processor by processor

Entry point for starting with `ILD@FCC`: fcc-config

Thank you for your attention!

This work was partially supported by the European Union's Horizon Europe Marie Sklodowska-Curie Staff Exchanges programme under grant agreement no. 101086276.

Co-funded by the European Union