Status of CLD tracking

Leonhard Reichenbach

CERN / University of Bonn

FCC Software Meeting
June 26, 2023

SPONSORED BY THE

Federal Ministry
of Education
and Research

How am I connected to tracking and CLD?

What do I want to do:

- Tracking performance plots ‘the key4hep way’ (detector independent but CLD first)
- Integrate ACTS for track fitting in key4hep
- Goal: study electron track reconstruction with Gaussian sum filters (GSF)
- Goal 2: be able to compare detector performances by just swapping geometry and reconstruction steering

How am I connected to tracking and CLD?

What do I want to do:

- Tracking performance plots 'the key4hep way' (detector independent but CLD first)
- Integrate ACTS for track fitting in key4hep
- Goal: study electron track reconstruction with Gaussian sum filters (GSF)
- Goal 2: be able to compare detector performances by just swapping geometry and reconstruction steering

I started this in March and I am still at the first step...

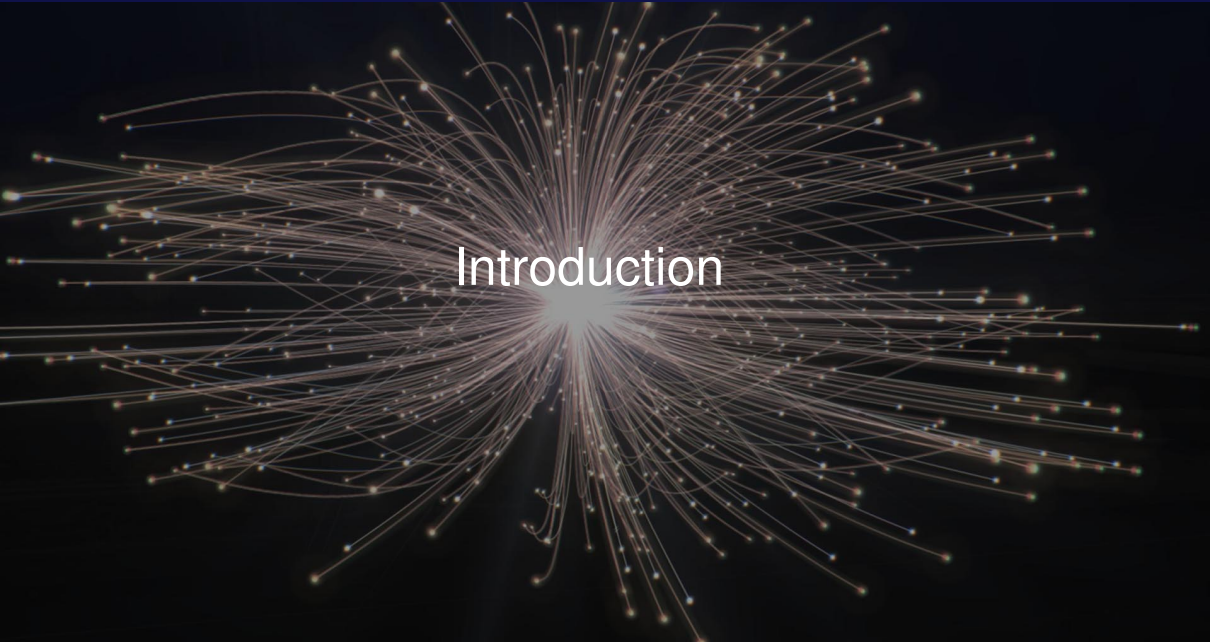
How am I connected to tracking and CLD?

What do I want to do:

- Tracking performance plots ‘the key4hep way’ (detector independent but CLD first)
- Integrate ACTS for track fitting in key4hep
- Goal: study electron track reconstruction with Gaussian sum filters (GSF)
- Goal 2: be able to compare detector performances by just swapping geometry and reconstruction steering

I started this in March and I am still at the first step...

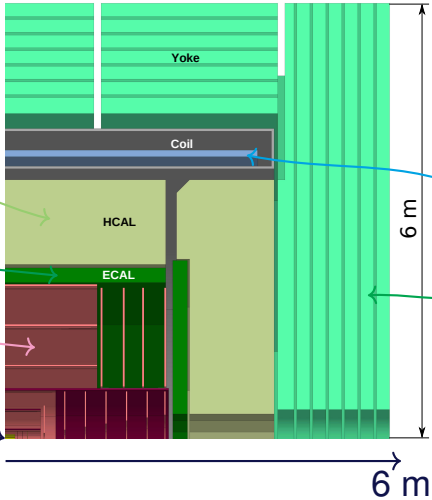
...let me share some of the pitfalls I encountered so you don’t have to



Introduction

CLD

General purpose detector for Particle Flow reconstruction [1]

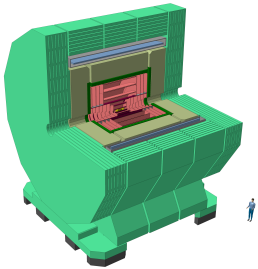


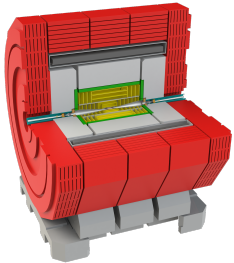
- Steel–Scintillator HCal with 3 cm cell-size
- Silicon–Tungsten ECal with 5 mm cell-size
- Silicon Tracker, mostly 50 μm pitch strips
- Vertex Detector with 25 μm pixels

- Superconducting Solenoid of 2 T
- Iron Yoke with RPCs for Muon ID

Historical context

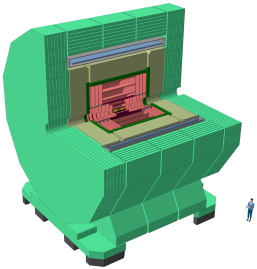
- The CLD (CLIC Like Detector) concept did not appear out of thin air
- It is an adaptation of CLICdet made to conform to FCCee conditions (2T solenoid limit)
- CLICdet was optimized for much higher energies (380GeV–3TeV)
- CLICdet itself already inherited much from ILD and they share the `Marlin` reconstruction framework





Historical context

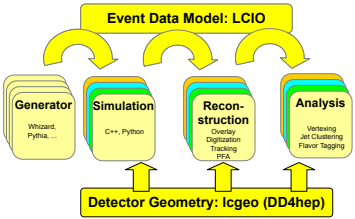
- The CLD (CLIC Like Detector) concept did not appear out of thin air
- It is an adaptation of CLICdet made to conform to FCCee conditions (2T solenoid limit)
- CLICdet was optimized for much higher energies (380GeV–3TeV)
- CLICdet itself already inherited much from ILD and they share the `Marlin` reconstruction framework
- (If you look deep into the source code you can even find traces of the GLD and LDC concepts that fused to form ILD)



Reconstruction and analysis at CLD and CLICdet

The (pre-key4hep) workflow used for existing studies:

- (Generation with Whizard or particle gun)
- Simulation with `ddsim` in LCIO (`.slcio`) format
- Reconstruction with `Marlin`
- Production of ntuples also with `Marlin`
- Analysis with ROOT macros, custom python scripts etc.
  or histograms directly produced with `Marlin`

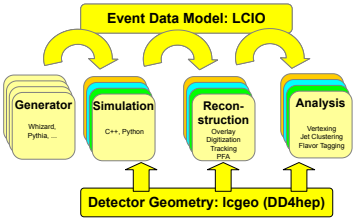


Reconstruction and analysis at CLD and CLICdet

The (pre-key4hep) workflow used for existing studies:

- (Generation with Whizard or particle gun)
- Simulation with `ddsim` in LCIO (`.slcio`) format
- Reconstruction with `Marlin`
- Production of ntuples also with `Marlin`
- Analysis with ROOT macros, custom python scripts etc.
  or histograms directly produced with `Marlin`

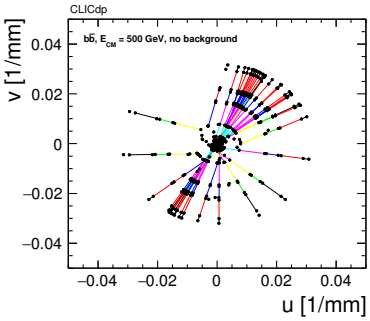
All necessary tools for full sim analysis are already there!



Tracking in Marlin

Used by CLD/CLICdet:

- `MarlinTrkProcessors`
  - A collection of processor for digitisation, track finding and fitting.
  - E.g. `DDPlanarDigiProcessor`, `RefitFinal`, `ClonesAndSplitTracksFinder`
- `MarlinTrk`
  - Provides track factory and interface to different fitters (`DDKalTest`, `aidaTT`)
- `ConformalTracking`
  - Finds and fits tracks using a conformal mapping[2]

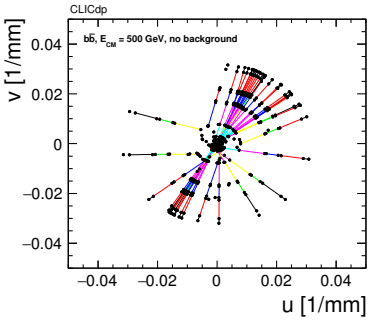


And a lot more not used by CLD like `FullLDCTracking_MarlinTrk`

Tracking in Marlin

Used by CLD/CLICdet:

- `MarlinTrkProcessors`
  - A collection of processor for digitisation, track finding and fitting.
  - E.g. `DDPlanarDigiProcessor`, `RefitFinal`, `ClonesAndSplitTracksFinder`
- `MarlinTrk`
  - Provides track factory and interface to different fitters (`DDKalTest`, `aidaTT`)
- `ConformalTracking`
  - Finds and fits tracks using a conformal mapping[2]



And a lot more not used by CLD like `FullLDCTracking_MarlinTrk`

CLD tracking in Key4hep

What about Key4hep?

- Gaudi (`k4run`) instead of Marlin as reconstruction framework
- EDM4hep instead of LCIO as event data model
- But we don't want to waste the $O(20)$ years worth of knowledge existing in Marlin
- The answer: `k4MarlinWrapper`

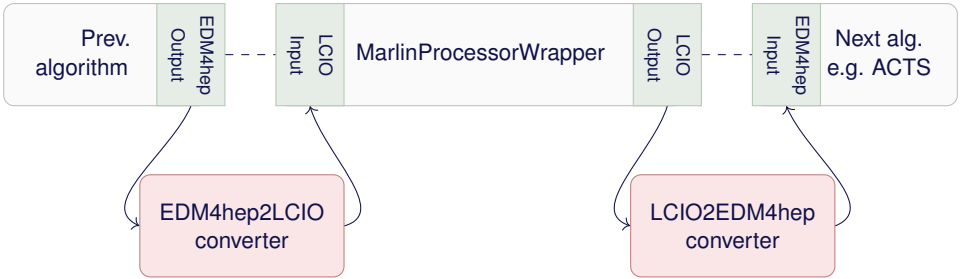
What about Key4hep?

- Gaudi (`k4run`) instead of Marlin as reconstruction framework
- EDM4hep instead of LCIO as event data model
- But we don't want to waste the $O(20)$ years worth of knowledge existing in Marlin
- The answer: `k4MarlinWrapper`

k4MarlinWrapper

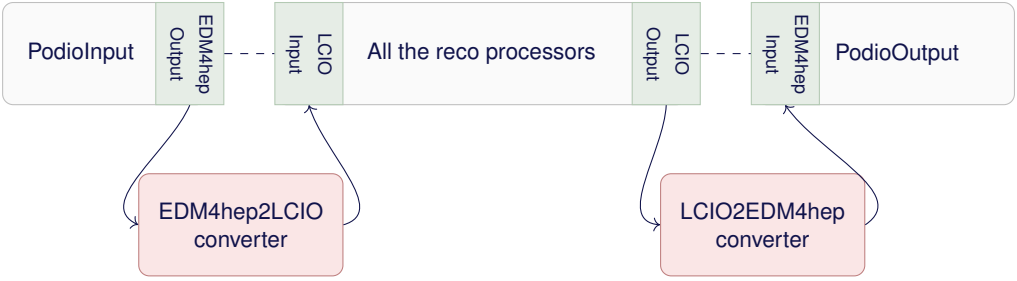
- Wraps Marlin processors to run as Gaudi algorithms
- Also offers translation tools to translate between EDM4hep and LCIO
- Enables us to use the whole CLD Marlin reconstruction chain also in Gaudi

- This works quite well, i.e. everything that worked in Marlin still works!
- Goal: be able to mix Gaudi algorithms and Marlin processors freely and have EDM4hep as in- and output



k4MarlinWrapper: reality

- At the moment our entire reconstruction runs on wrapped Marlin processors
- With EDM4hep input we attach a converter to the first wrapper to convert to LCIO
- The LCIO data exists in memory separate from the EDM4hep data
- MarlinProcessorWrappers further down the chain only touch this separate event store
- We attach another converter to the last wrapper to convert back to EDM4hep
- Unfortunately this does not run completely smooth yet



k4MarlinWrapper: formats

Since the last key4hep release in April some bugs in the conversion were found

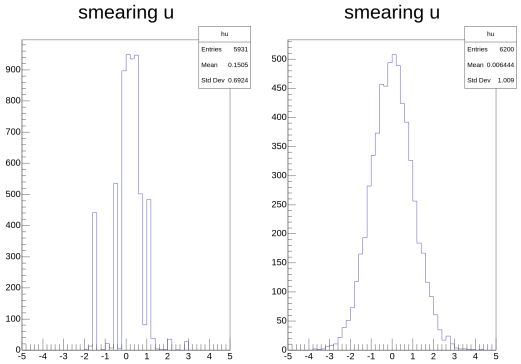
- EDM4hep EventHeader was not converted: k4MarlinWrapper#111
- LCIO to EDM4hep conversion does not properly treat subset collections and relations: k4MarlinWrapper#113

in\out	LCIO	EDM4hep
LCIO	✅	✅
EDM4hep	(✅)	🚧

CLD reconstruction format choices

k4MarlinWrapper#111

- EDM4hep EventHeader was not converted: k4MarlinWrapper#111
- Therefore event and run number were always 0 but they are used for random seeding
- Fixed in the nightlies and will be in the next release



Gaussian smearing of silicon hits before and after fix

`k4MarlinWrapper`: conversion of relations from LCIO to EDM4hep

Here we had a multitude of overlapping bugs related to non-proper treatment of subset collections

- LCIO subset collection particles were cloned and the clones existed independently of each other in multiple collections
- Conversions of LCIO relations to EDM4hep associations then took the particle (clone) from the lexicographically first collection containing it.
- Users found very creative ‘workarounds’ for this but never reported the issue...

Row	Instance	colID
0	0	41
0	1	1
0	2	41
0	3	41
0	4	1
0	5	41

Changing collectionID in EDM4hep associations

`k4MarlinWrapper`: conversion of relations from LCIO to EDM4hep

Aforementioned problem is partly fixed, but...

- Old conversion only uses non-subset collections to build associations now
- New conversion also treats them appropriately
- Independently of this a memory leak was fixed in the old conversion uncovering a new problem that both conversions share

`k4MarlinWrapper`: conversion of relations from LCIO to EDM4hep

Aforementioned problem is partly fixed, but...

- Old conversion only uses non-subset collections to build associations now
- New conversion also treats them appropriately
- Independently of this a memory leak was fixed in the old conversion uncovering a new problem that both conversions share

- There is no synchronisation between which LCIO and EDM4hep objects belong together
- If there is more than one conversion step, relations/associations where at least one of the two sides existed before the first conversion break!
- I.e. our new standard workflow (EDM4hep $\rightarrow$ LCIO $\rightarrow$ EDM4hep) comes with broken associations!

`k4MarlinWrapper`: other issues and changes

- Everything using explicit collection IDs will break in the next release due to switch to hashed collection names (podio#412)
- In the last couple of releases `ddsim` does not attach calorimeter hits to events in LCIO output mode $\rightarrow$ fix is already in the nightlies (key4hep-spack#487)

`k4MarlinWrapper`: status & workarounds

TL; DR:

- All the Marlin stuff works e.g. replication of CLIC/CLD studies
- If you want to have EDM4hep output you can either
  - Use `ddsim` in LCIO output mode
  - Convert your EDM4hep input to LCIO by a standalone conversion before the reconstruction
- If you are only interested in LCIO output but have EDM4hep input files that should also work

in\out	LCIO	EDM4hep
LCIO	✅	✅
EDM4hep	(✅)	🚧

CLD reconstruction format choices

If you encounter any issues **please let us know** by opening a ticket in github.com/key4hep/k4MarlinWrapper/issues

`k4MarlinWrapper`: status & workarounds

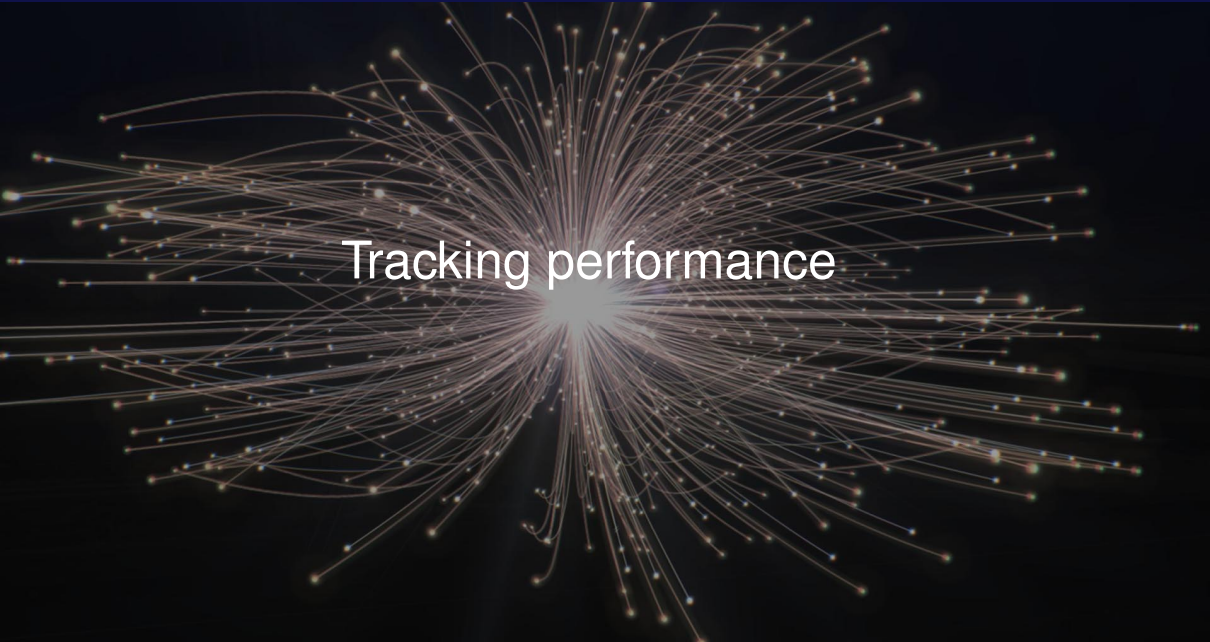
TL; DR:

- All the Marlin stuff works e.g. replication of CLIC/CLD studies
- If you want to have EDM4hep output you can either
  - Use `ddsim` in LCIO output mode
  - Convert your EDM4hep input to LCIO by a standalone conversion before the reconstruction
- If you are only interested in LCIO output but have EDM4hep input files that should also work

in\out	LCIO	EDM4hep
LCIO	✅	✅
EDM4hep	(✅)	🚧

CLD reconstruction format choices

If you encounter any issues **please let us know** by opening a ticket in github.com/key4hep/k4MarlinWrapper/issues

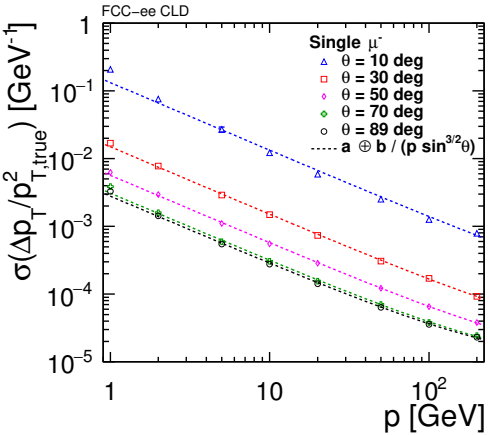


Tracking performance

Tracking performance

All plots still made with the old beam pipe (`FCCee_o1_v04`) and old Marlin workflow[1]

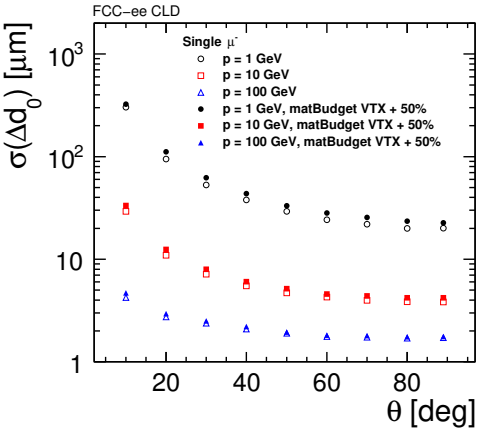
- Momentum resolution
- Impact parameter resolution
  - Also estimated for larger material budget in the vertex detector
- Single particle efficiency w.r.t. transverse momentum
- Single particle efficiency w.r.t. radius



Tracking performance

All plots still made with the old beam pipe (`FCCee_o1_v04`) and old Marlin workflow[1]

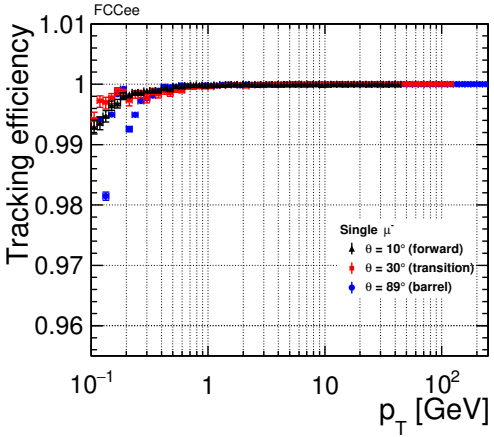
- Momentum resolution
- Impact parameter resolution
  - Also estimated for larger material budget in the vertex detector
- Single particle efficiency w.r.t. transverse momentum
- Single particle efficiency w.r.t. radius



Tracking performance

All plots still made with the old beam pipe (`FCCee_o1_v04`) and old Marlin workflow[1]

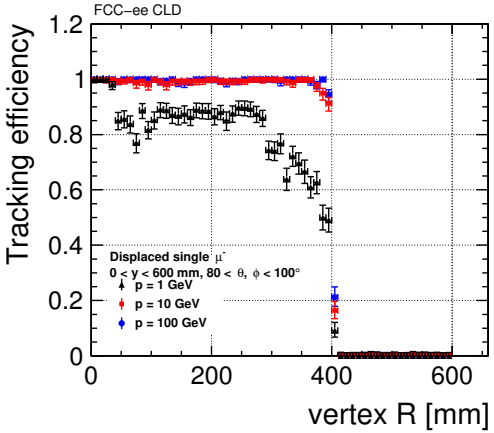
- Momentum resolution
- Impact parameter resolution
  - Also estimated for larger material budget in the vertex detector
- Single particle efficiency w.r.t. transverse momentum
- Single particle efficiency w.r.t. radius

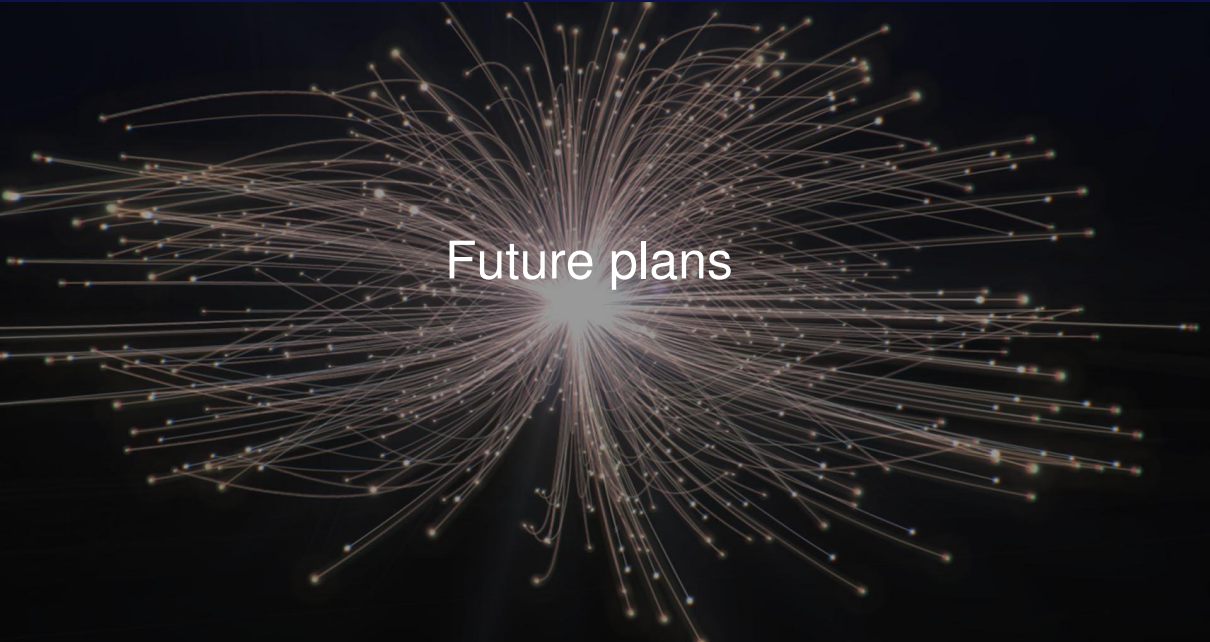


Tracking performance

All plots still made with the old beam pipe (`FCCee_o1_v04`) and old Marlin workflow[1]

- Momentum resolution
- Impact parameter resolution
  - Also estimated for larger material budget in the vertex detector
- Single particle efficiency w.r.t. transverse momentum
- Single particle efficiency w.r.t. radius





Future plans

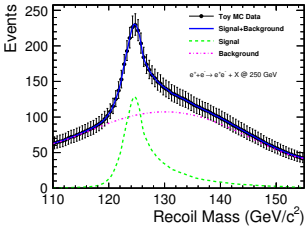
Performance evaluation plans

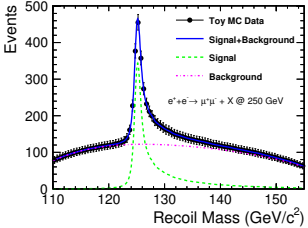
- Re-create CLIC/CLD tracking performance plots in a Key4hep ‘native’ and detector agnostic way
- Also create physics analysis based benchmarks
  - ZH-recoil and $WW/ev W$
- Electron focused to evaluate planned specialised electron tracking

Tracking plans

Improve electron track reconstruction

- ACTS-based tracking in Key4hep
- Additional track fit for electrons with a Gaussian Sum Filter (GSF)
- Matching of bremsstrahlung photons
- $\rightarrow$Improved p_{T} and impact parameter resolution





Taken from 1604.07524 [3]

Summary

Summary

- CLD reconstruction works in Key4hep but format conversion is tricky and not completely solved yet!
- We plan to enable better and easier comparisons between concepts for future detectors.
- Key4hep will get ACTS-enabled tracking with additional GSF fits for electron candidates
- If you encounter any issues **please let us know** by opening a ticket in e.g. github.com/key4hep/k4MarlinWrapper/issues

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 13E18CHA).

SPONSORED BY THE
Federal Ministry of Education and Research

Summary

- CLD reconstruction works in Key4hep but format conversion is tricky and not completely solved yet!
- We plan to enable better and easier comparisons between concepts for future detectors.
- Key4hep will get ACTS-enabled tracking with additional GSF fits for electron candidates
- If you encounter any issues **please let us know** by opening a ticket in e.g. github.com/key4hep/k4MarlinWrapper/issues

This work has been sponsored by the Wolfgang Gentner Programme of the German Federal Ministry of Education and Research (grant no. 13E18CHA).

SPONSORED BY THE
Federal Ministry of Education and Research

References

[1] N. Bacchetta et al. *CLD – A Detector Concept for the FCC-ee*. 2019. arXiv: 1911.12230 [physics.ins-det].

[2] E. Brondolin et al. "Conformal tracking for all-silicon trackers at future electron–positron colliders". In: *Nucl. Instr. Meth.* A956 (2020), p. 163304. DOI: 10.1016/j.nima.2019.163304.

[3] Jacqueline Yan et al. "Measurement of the Higgs boson mass and $e^+e^- \to ZH$ cross section using $Z \to \mu^+\mu^-$ and $Z \to e^+e^-$ at the ILC". In: *Phys. Rev. D* 94.11 (2016). [Erratum: Phys.Rev.D 103, 099903 (2021)], p. 113002. DOI: 10.1103/PhysRevD.94.113002. arXiv: 1604.07524 [hep-ex].