

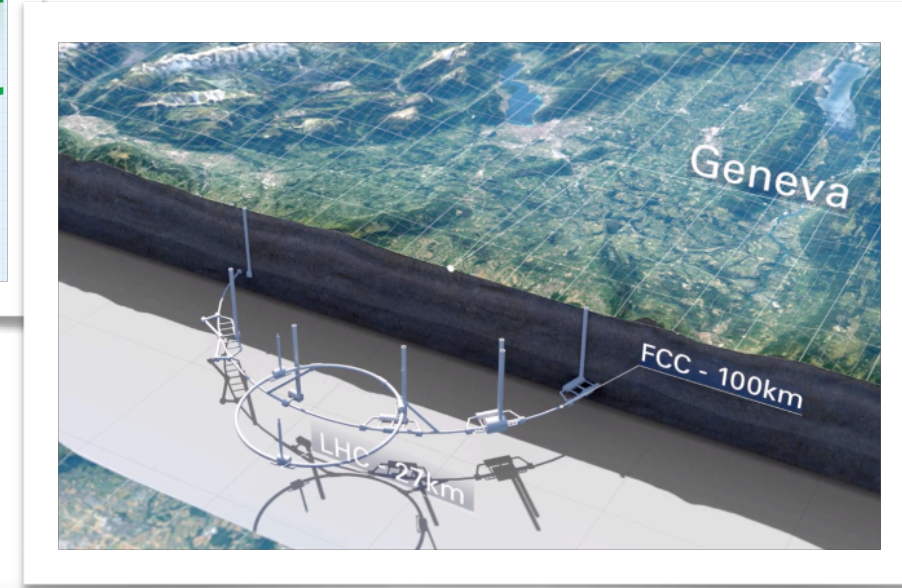
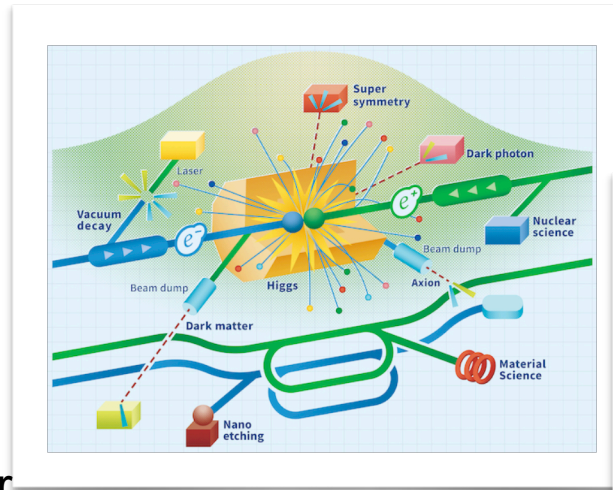
Reconstruction for e^+e^- Higgs Top & Electroweak Factory Experiments

3rd ECFA workshop on Higgs, Top & Electroweak Factories
9-11 Oct, Paris

Frank Gaede, DESY

Outline

- Introduction
 - Key4hep, DD4hep and MarlinWrapper
- Existing reconstruction code
 - MarlinTrk, PandoraPFA,...
- Highlights of more recent developments:
 - Tracking, PFA, HF-Tagging, PID,...
 - with and without AI/ML
- Summary



ECFA Higgs Factories: 1st Topical Meeting on Reconstruction

4 May 2022, 09:00 → 5 May 2022, 18:00 Europe/Zurich

DESY

ECFA Higgs Factories: 2nd Topical Meeting on Reconstruction

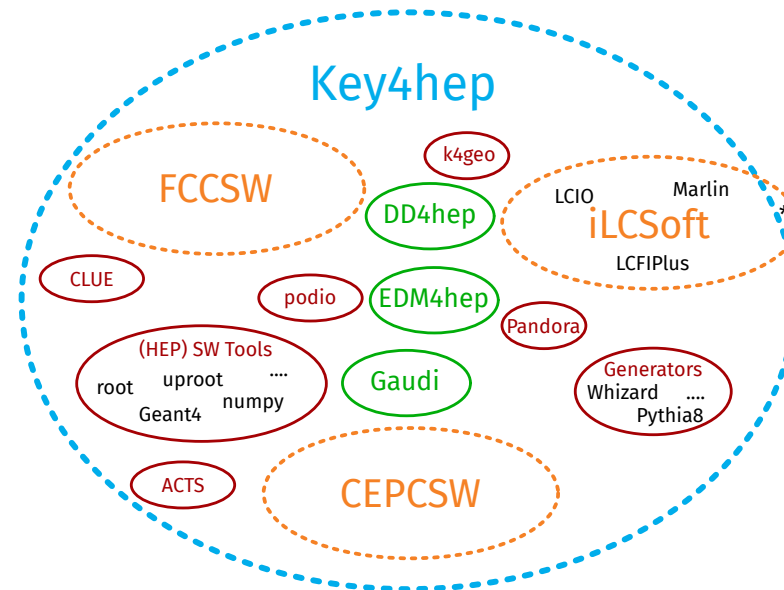
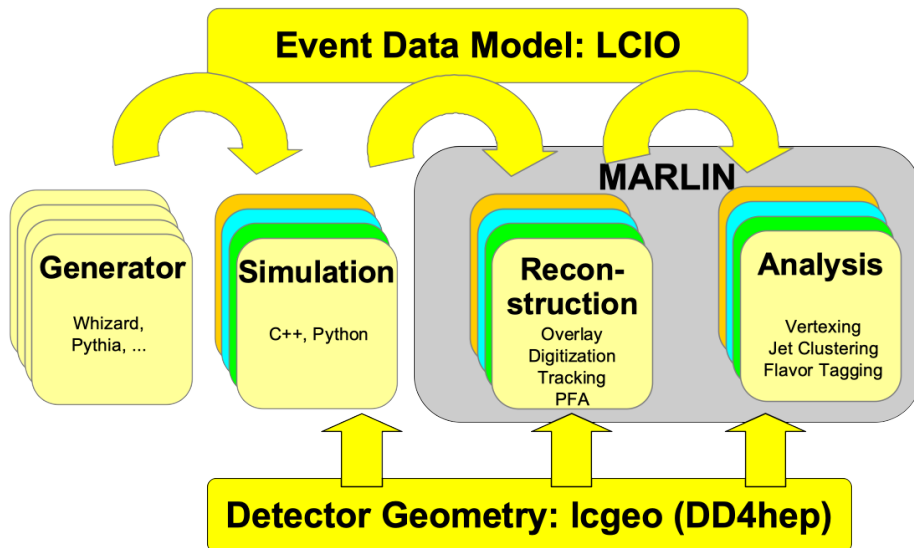
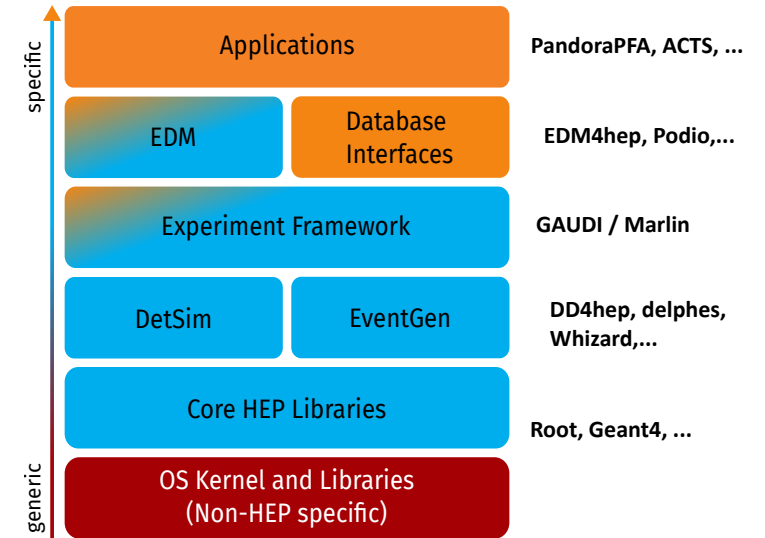
11 Jul 2023, 09:00 → 12 Jul 2023, 18:00 Europe/Zurich

40/S2-C01 - Salle Curie (CERN)

Key4hep

the turnkey software stack for ALL future colliders

- HEP community decided 5 years ago to develop a **common turnkey software stack** – for future collider studies
- create a software ecosystem integrating in an **optimal way the best software components** to provide a **ready-to-use full-fledged solution** for data processing of (future collider) **HEP** experiments
- involved communities/contributors: CEPC, CLIC, EIC, FCCee, FCChh, ILC, LUXE, Muon Collider ...

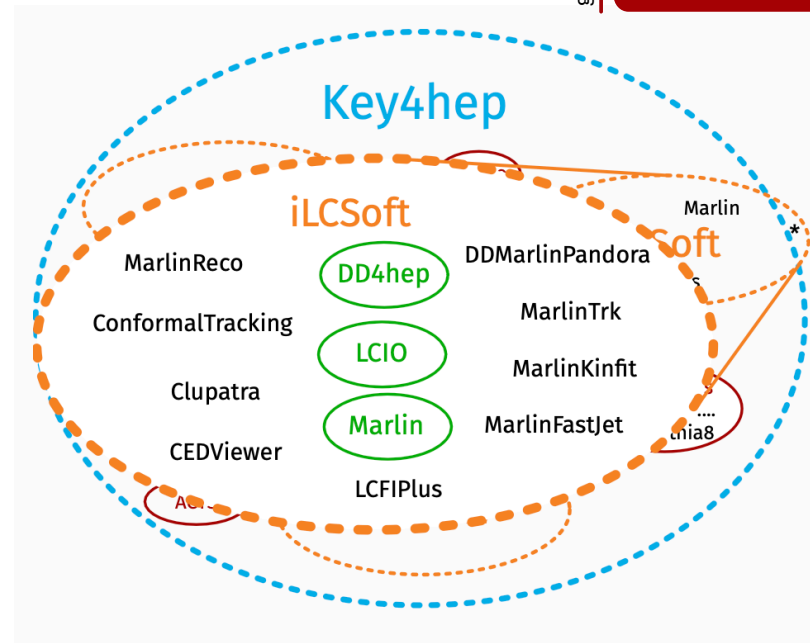
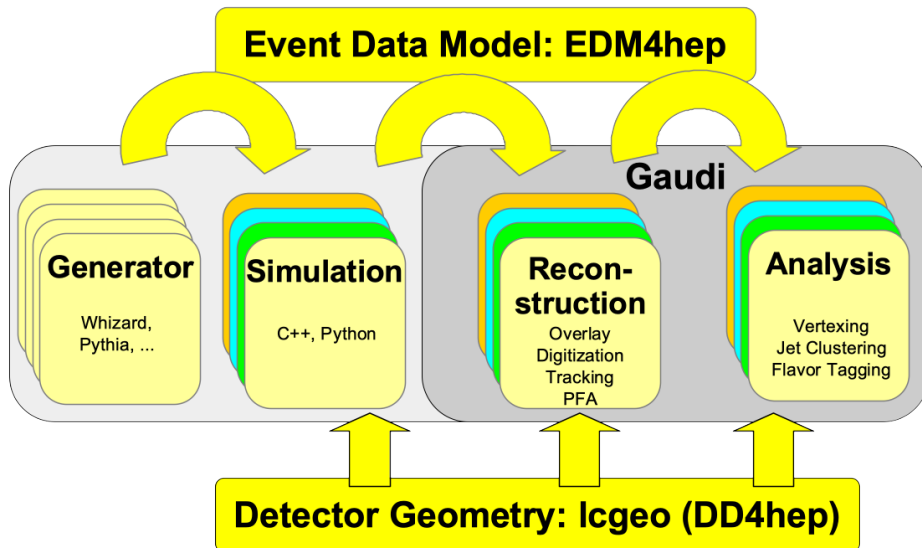
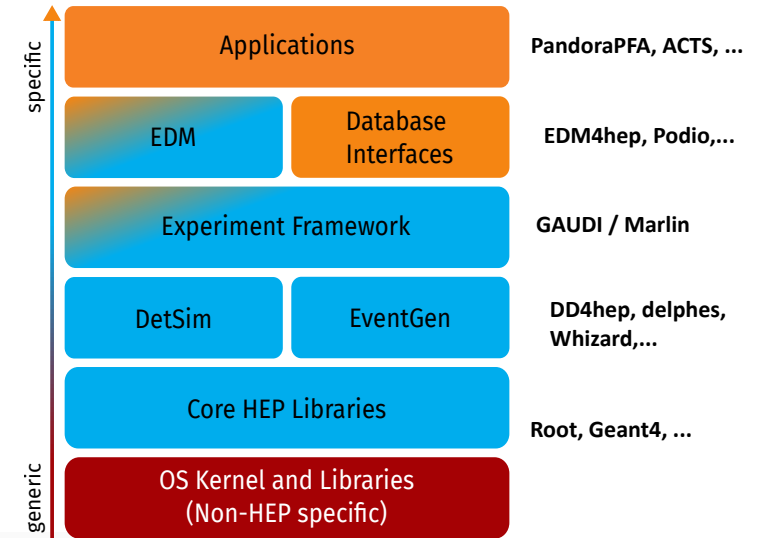


see talk by J.Smiesko, Wed.

Key4hep

the turnkey software stack for ALL future colliders

- HEP community decided 5 years ago to develop a **common turnkey software stack** – for future collider studies
- create a software ecosystem integrating in an **optimal way the best software components** to provide a **ready-to-use full-fledged solution** for data processing of (future collider) **HEP** experiments
- involved communities/contributors: CEPC, CLIC, EIC, FCCee, FCChh, ILC, LUXE, Muon Collider ...

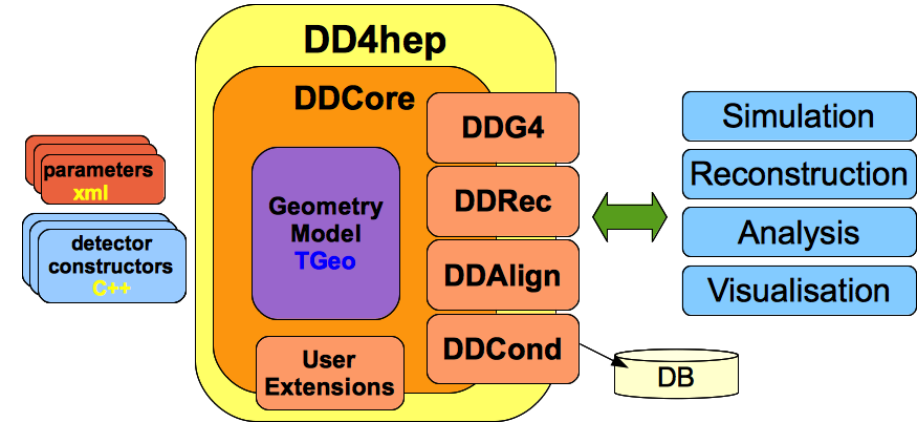


see talk by J.Smiesko, Wed.

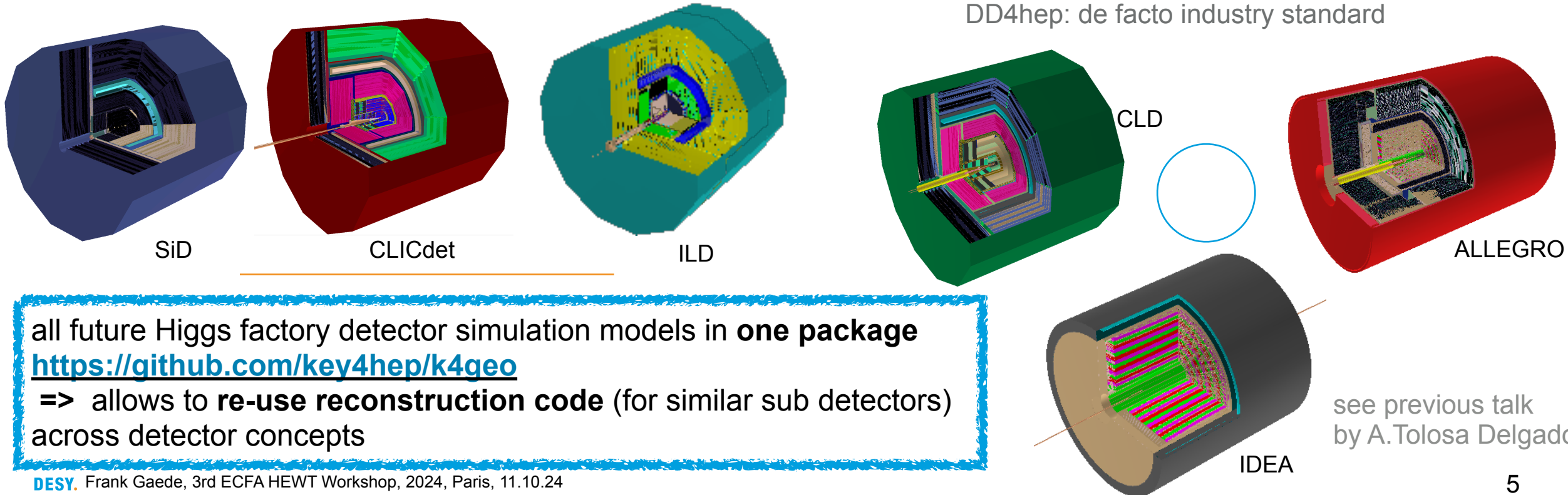
DD4hep geometry toolkit

defining the detector geometry and different views on it

- supporting the full life cycle of the experiment
- **single source** of information for full **simulation**, **reconstruction**, **conditions**, **alignment**, **visualisation** and **analysis**
 - used by CEPC, CLIC, CMS, EIC, FCC, ILC, LHCb, ...



DD4hep: de facto industry standard



all future Higgs factory detector simulation models in **one package**

<https://github.com/key4hep/k4geo>

=> allows to **re-use reconstruction code** (for similar sub detectors) across detector concepts

see previous talk by A.Tolosa Delgado

large reconstruction code base in iLCSoft

Developed over >15 years for (linear) lepton colliders

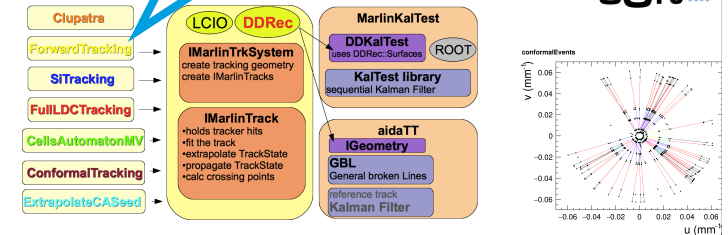
- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
 - generic API for fitting algorithms
 - large number of pattern recognition algorithms

Hopfield Model inside

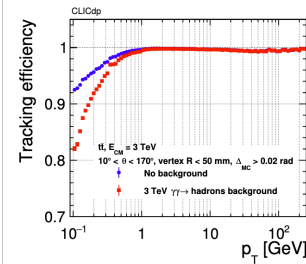
Tracking in iLCSoft

pattern recognition and Kalman-Filter

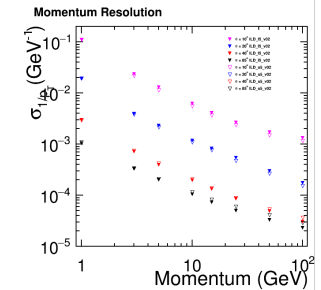
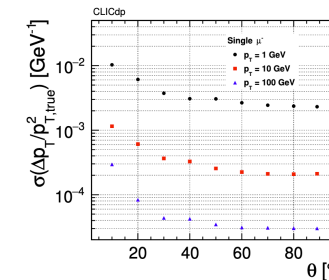
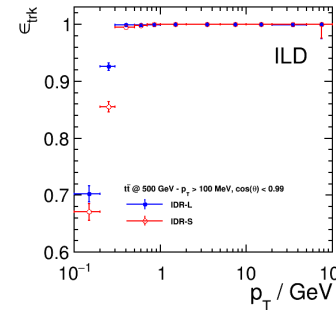
- generic tracking API MarlinTrk based on DDRec material surfaces
- many pattern recognition algorithms exist, e.g.
- **ConformalTracking**:
 - generic algorithm that works for all Si-Trackers
 - used by CLICdet and SiD (also works for ILD inner)



achieve excellent tracking efficiencies and resolution w/ realistic tracking codes



DESY. Frank Gaede, LCWS 2021, 17.03.21



6

large reconstruction code base in iLCSoft

Developed over >15 years for (linear) lepton colliders

- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
 - generic API for fitting algorithms
 - large number of pattern recognition algorithms
- particle flow algorithms
 - PandoraPFA ans Arbor, AprilPFA

Hopfield Model inside

Tracking in iLCSoft

pattern recognition and Kalman-Filter

Clupetra

LCIO

DDRec

MarlinKalTest

DDKalTest

Particle Flow Algorithms

highly granular calorimeter reconstruction

- all current detector concepts for LC are based on highly granular calorimeters
 - optimised for the Particle Flow Algorithm
- **PandoraPFA** is the **de facto standard** used by ILD, SiD and CLICdP
- alternative PFA algorithms exist and provide possibility to cross check
 - Arbor (CEPC), April (SDHCAL prototype)

Pandora Algorithms

slide: J.Marshall

Cone-based forward projective method

Projected track position

Clustering Algorithm

Topological Association Algorithms

Photon Recovery Algorithm

Fragment Removal Algorithms

Track-cluster Association Algorithms

PFO Construction Algorithm

Cone associations

Back-scattered tracks

Looping tracks

9 GeV

6 GeV

9 GeV

6 GeV

Fraction of energy in cone

ArborPFA

AprilPFA

DESY. Frank Gaede, LCWS 2021, 17.03.21

large reconstruction code base in iLCSoft

Developed over >15 years for (linear) lepton colliders

- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
 - generic API for fitting algorithms
 - large number of pattern recognition algorithms
- particle flow algorithms
 - PandoraPFA and Arbor, AprilPFA
- high level reconstruction
 - jet finding, flavor tagging, PID, TOF, ...

Hopfield Model inside

Tracking in iLCSoft

pattern recognition and Kalman-Filter

Clupetra, LCIO, DDRc, MarlinKalTest, DDKalTest

Particle Flow Algorithms

High Level Reconstruction

analysing the Particle Flow Objects

- **High-Level reconstruction** algorithms are crucial to achieve the ultimate physics reach of detectors
- vertex finding and flavor tagging: **LCFIPlus**
- PID tools: dE/dx, TOF, shower shapes, ...
- Jet clustering: Durham, Valencia, ...

- very active field of development
 - already good set of tools available
 - further improvement in HLR tools often directly impacts the final physics performance

$\delta\lambda_{HHH}$ improves by 40% w/ perfect jet clustering

DESY. Frank Gaede, LCWS 2021, 17.03.21

large reconstruction code base in iLCSoft

Developed over >15 years for (linear) lepton colliders

- realistic detector models for incl. tracking/reconstruction geometry
- track reconstruction
 - generic API for f
 - large number of recognition algo
- particle flow algorithm
 - PandoraPFA ar
- high level reconstruction
 - jet finding, flavor tagging, PID, TOF,...

• vital for the FC community to preserve this code
 • a lot of this also very useful for FCC(ee) studies
 • new algorithms should - and are - developed in Gaudi
 • developed a migration scenario that allows a smooth transition from LCIO/Marlin to EDM4hep/Gaudi

Hopfield Model inside

Tracking in iLCSoft

pattern recognition and Kalman-Filter

Clupetra

LCIO

DDRec

MarlinKalTest

DDKalTest

scatter plot of two Higgs masses

perfect jet-clustering

Legend: ■ vvhH channel, ■ vvbb channel, ■ vvbb channel

real jet-clustering

Legend: ■ vvhH channel, ■ vvbb channel, ■ vvbb channel

(without beam overlay)

$\delta\lambda_{HHH}$ improves by 40% w/ perfect jet clustering

DESY. Frank Gaede, LCWS 2021, 17.03.21

very active field of development

- already good set of tools available
- further improvement in HLR tools often directly impacts the final physics performance

algorithms are crucial to achieve the ultimate physics reach of detectors

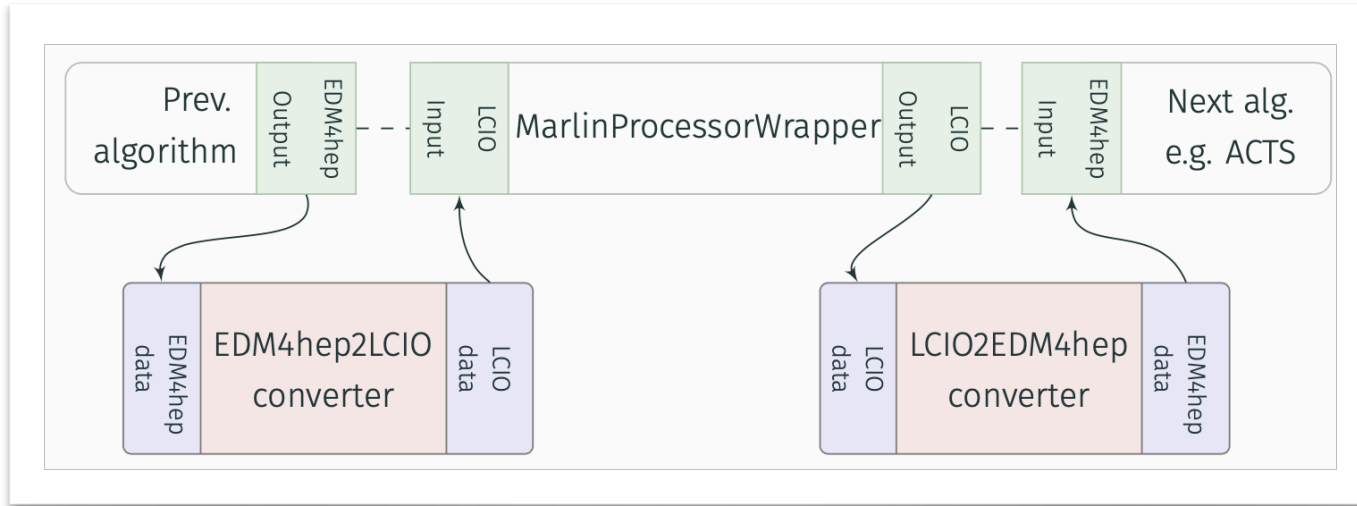
- vertex finding and flavor tagging: **LCFIPlus**
- PID tools: dE/dx, TOF, shower shapes,...
- Jet clustering: Durham, Valencia, ...

dE/dx (GeV/mm) vs Momentum (GeV)

Legend: ■ ILD, ■ IORL

K4MarlinWrapppper

the vision: mix and match Marlin and Gaudi algorithms



- in a transition phase algorithms **developed in the new EDM4hep/Gaudi** world can gradually replace older algorithms
 - e.g. eventually one might want to replace track fitting with **ACTS** also for LC detectors
- some technicalities are address *under-the-hood* via **k4EDM4hep2LcioConv**



all existing (high level) reconstruction algorithms as Marlin processors fully available in Key4hep !

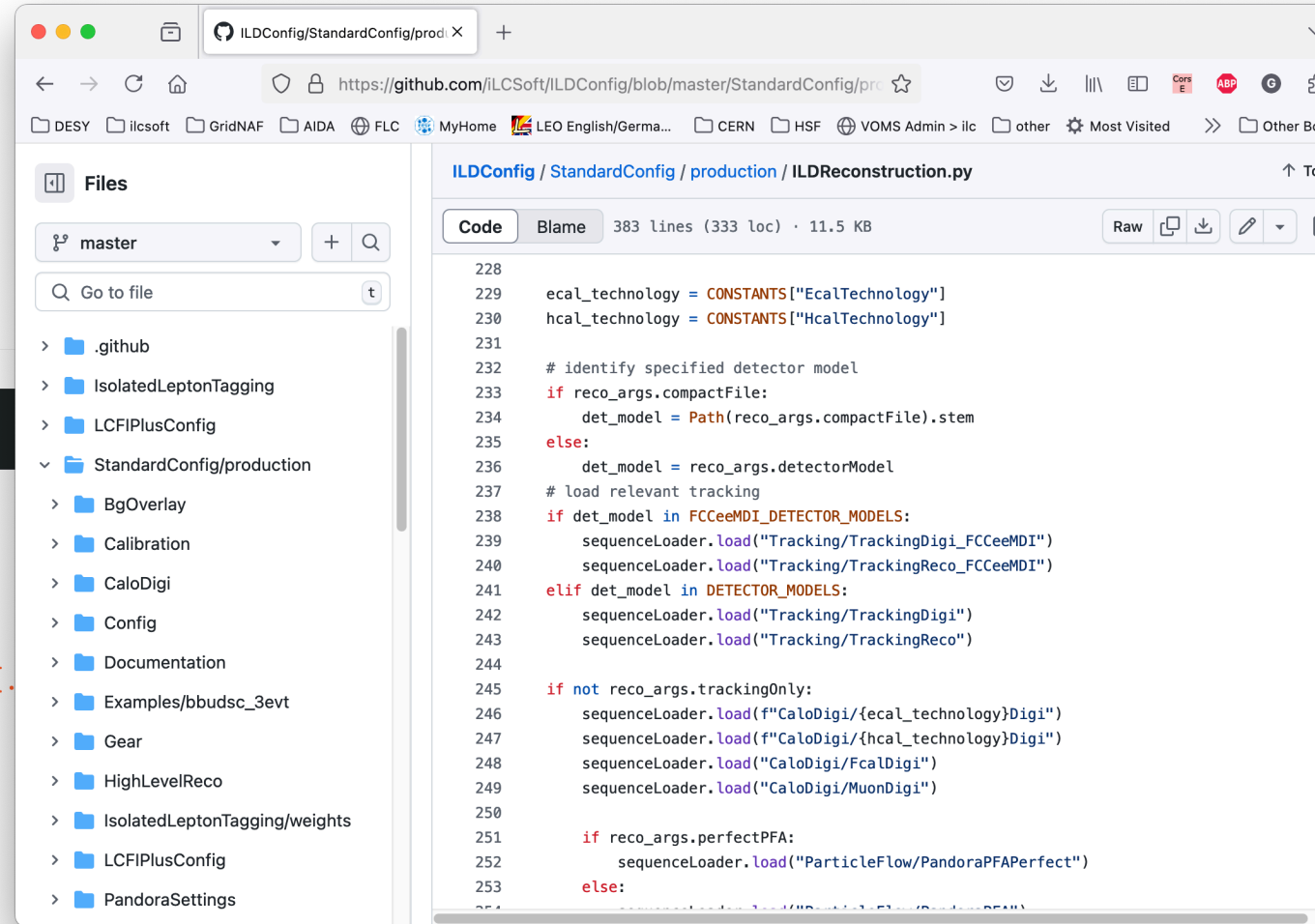
ILD Standard Reconstruction

now smoothly runs in in Key4hep

- translation of *MarlinReco.xml* to GAUDI python steering file *ILDReconstruction.py*
- plan to use this as new standard for ILD soon

ILD standard reconstruction in Key4hep

- All configuration available from [iLCSoft/ILDConfig](https://github.com/iLCSoft/ILDConfig)
 - Everything that works in iLCSoft also works in Key4hep!
- ```
Marlin MarlinStdReco.xml --global.LCIOInputFiles=<input-file> [...]
```
- Now also with Gaudi
    - `k4run ILDReconstruction.py --inputFiles=<input-file> [...]`
      - Works with EDM4hep and LCIO inputs
      - EDM4hep output by default, LCIO output via `--lcioOutput=[true|only]`
  - Facilitates collaboration with other projects, e.g. CLD
  - Full migration of all workflows will take some time but process started
    - Some new developments already done exclusively in Gaudi configuration



The screenshot shows a web browser displaying the GitHub repository for `iLCSoft/ILDConfig`. The file browser on the left shows the directory structure, with `StandardConfig/production` expanded to show subdirectories like `BgOverlay`, `Calibration`, `CaloDigi`, `Config`, `Documentation`, `Examples/bbudsc_3evt`, `Gear`, `HighLevelReco`, `IsolatedLeptonTagging/weights`, `LCFIPlusConfig`, and `PandoraSettings`.

The main content area shows the code for `ILDReconstruction.py` (383 lines, 333 loc, 11.5 KB). The code includes configuration for detector models and sequence loading. Key lines include:

```

228
229 ecal_technology = CONSTANTS["EcalTechnology"]
230 hcal_technology = CONSTANTS["HcalTechnology"]
231
232 # identify specified detector model
233 if reco_args.compactFile:
234 det_model = Path(reco_args.compactFile).stem
235 else:
236 det_model = reco_args.detectorModel
237 # load relevant tracking
238 if det_model in FCCeeMDI_DETECTOR_MODELS:
239 sequenceLoader.load("Tracking/TrackingDigi_FCCeeMDI")
240 sequenceLoader.load("Tracking/TrackingReco_FCCeeMDI")
241 elif det_model in DETECTOR_MODELS:
242 sequenceLoader.load("Tracking/TrackingDigi")
243 sequenceLoader.load("Tracking/TrackingReco")
244
245 if not reco_args.trackingOnly:
246 sequenceLoader.load(f"CaloDigi/{ecal_technology}Digi")
247 sequenceLoader.load(f"CaloDigi/{hcal_technology}Digi")
248 sequenceLoader.load("CaloDigi/FcalDigi")
249 sequenceLoader.load("CaloDigi/MuonDigi")
250
251 if reco_args.perfectPFA:
252 sequenceLoader.load("ParticleFlow/PandoraPFAPerfect")
253 else:

```

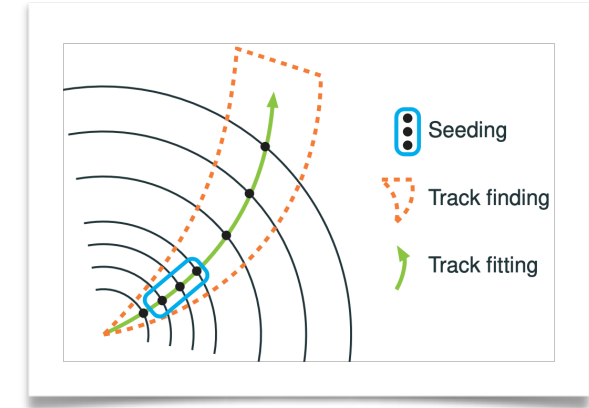
T.Madlener

# ACTS - A Common Tracking Toolkit

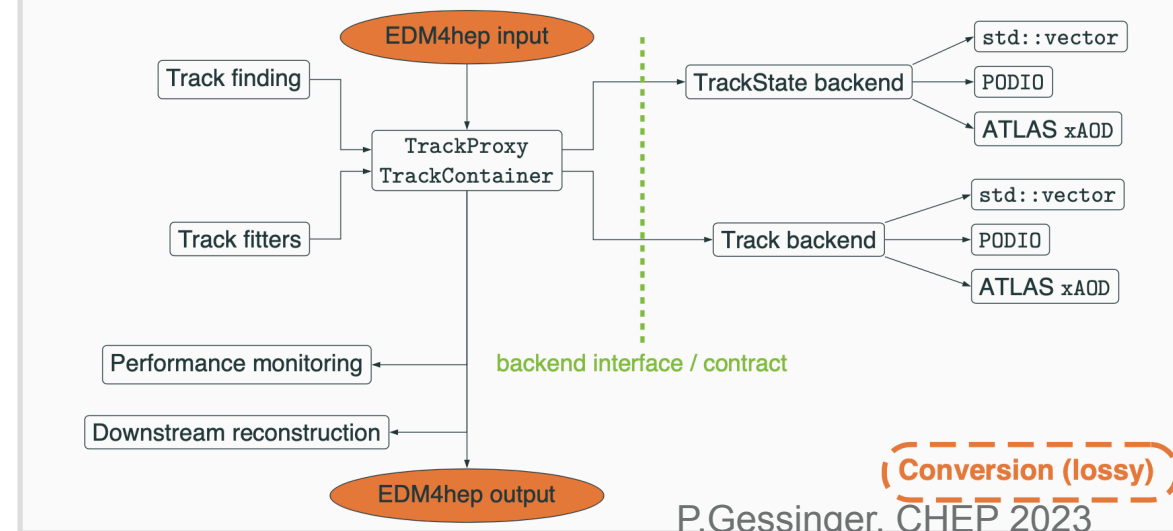
used in Key4hep

- **ACTS** tracking toolkit is the the current choice for track fitting (and finding) in Key4hep
- new package **k4ACTSTracking** provides ACTS fitter
  - implemented as GAUDI algorithm
  - uses DD4hep geometry
- first implementation of TrueTrackfinder for CERN OpenDetector
- successfully used for MuonCollider w/ handcrafted tracking geometry
- ongoing work on **automatic and transparent** construction of **tracking geometry** for all detectors in k4geo (Key4hep)
- not clear how soon this will be resolved...

L.Reichenbach



## Architecture

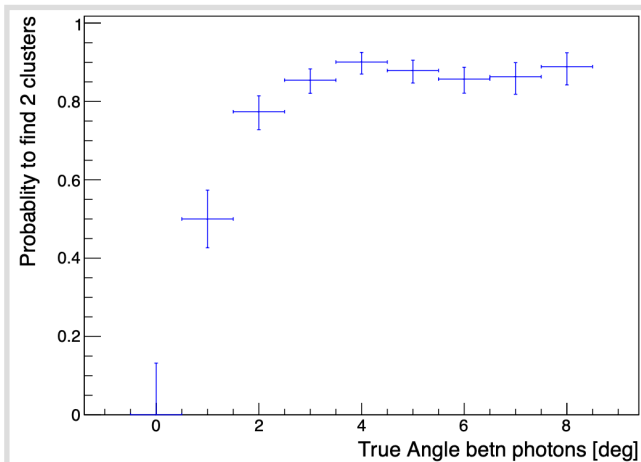
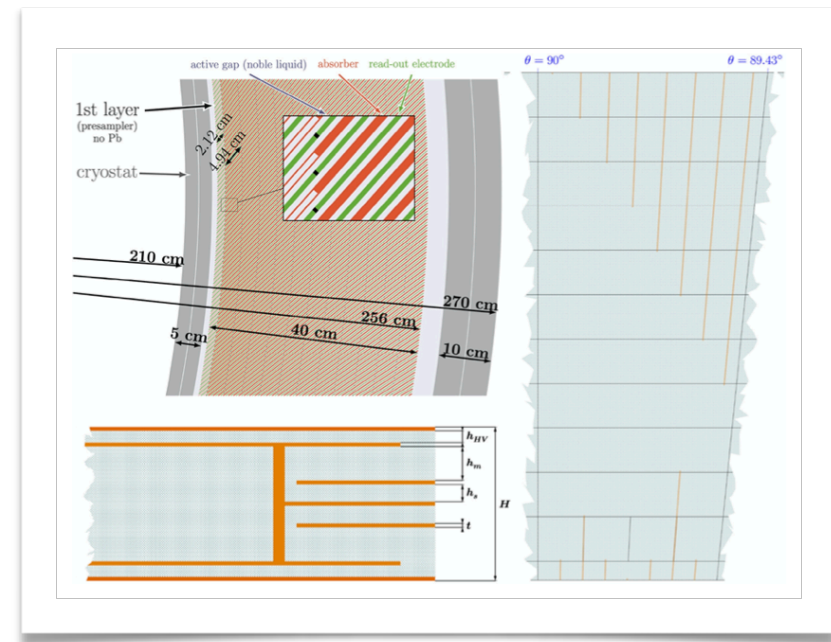


P.Gessinger, CHEP 2023

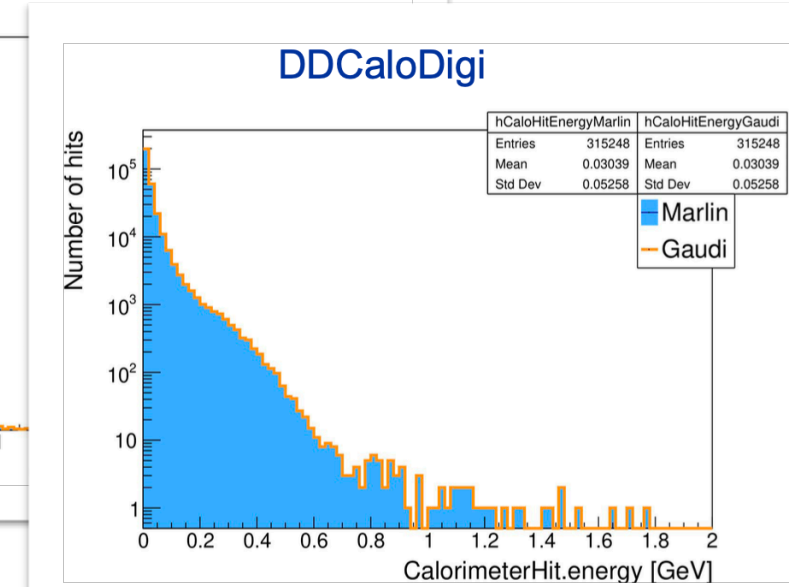
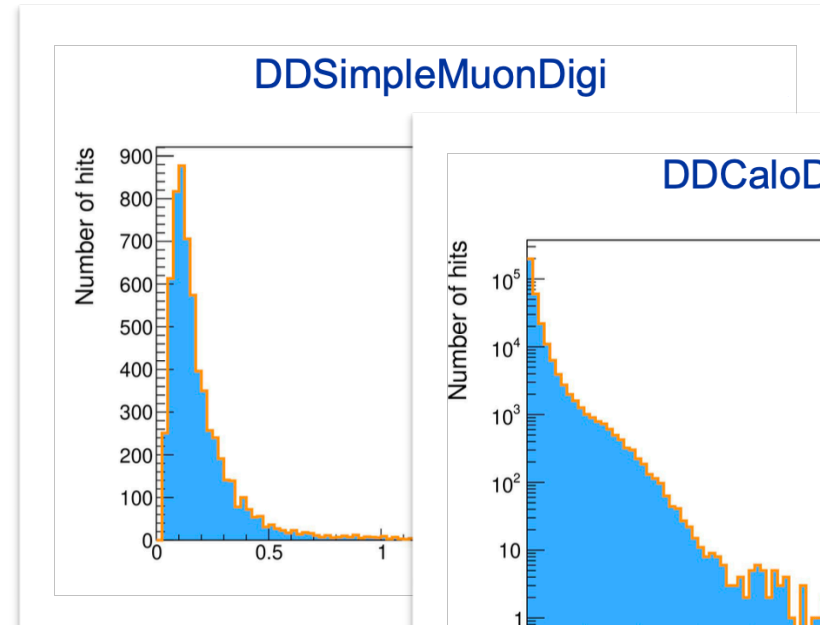
# PFA for LAr Calorimeters

## k4GaudiPandora: interfacing PandoraPFA to Key4hep

- goal to have general interface to PandoraPFA for all future collider detectors that have a DD4hep geometry model
- first implementation adapt use of **PandoraPFA for LAr calorimeter reconstruction**
  - needed ‘re-interpretation’ of calorimeter layers as used in highly granular calorimeters
  - adressed in DD4hep::MaterialManager
- used special version of CLD detector with a LAr calorimeter
- started porting DDMarlinPandora to **k4GaudiPandora**



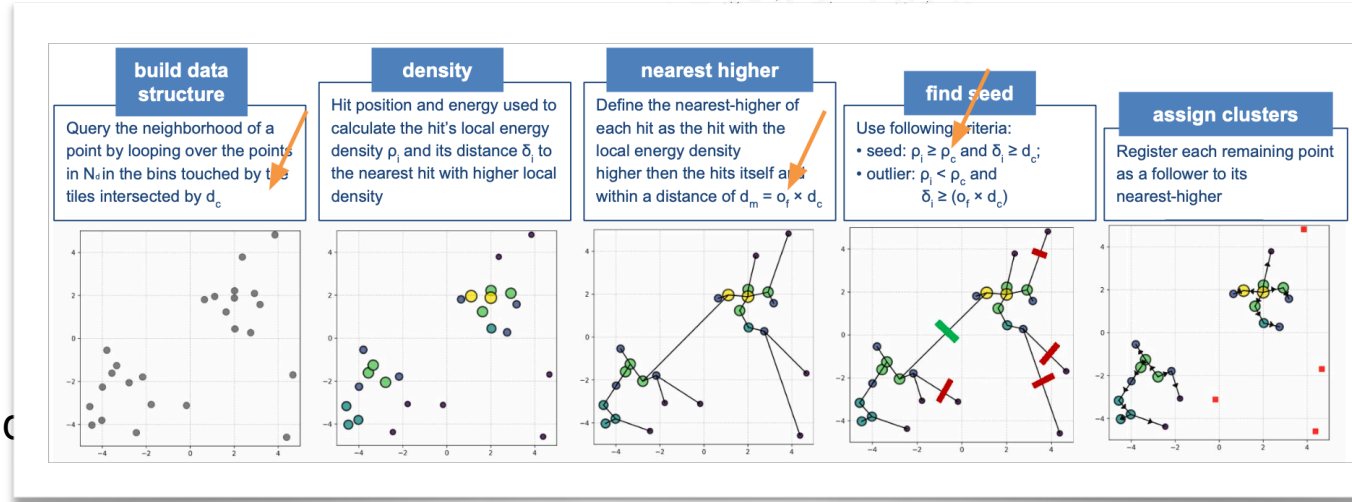
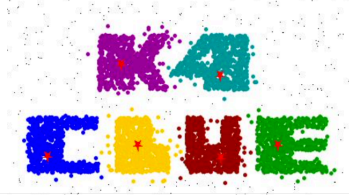
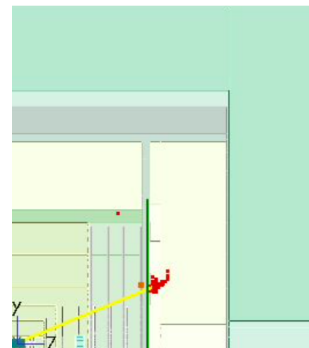
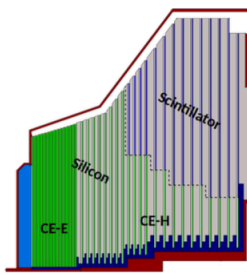
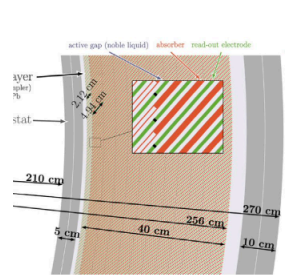
S.Sassikumar



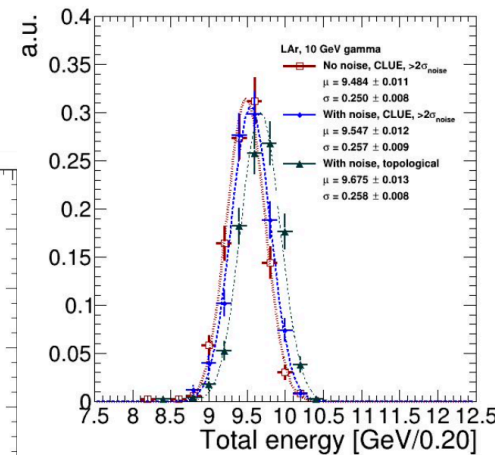
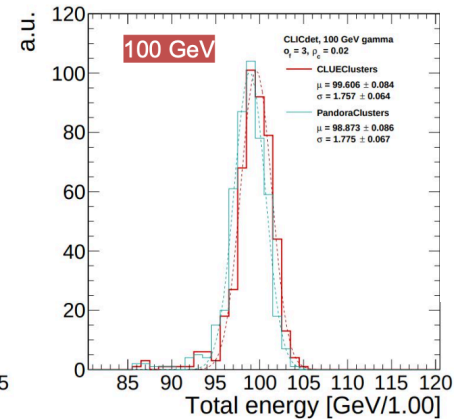
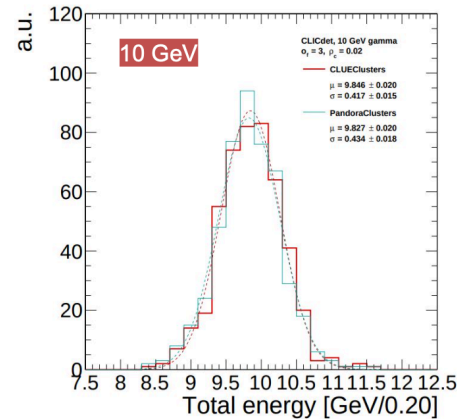
# k4Clue

## CLUstering Energy

- originally developed for CMS HGCal:
  - fast clustering for high granular calorimeters - based on local energy density
- ported to Key4hep and extended to 4pi geometry
- dedicated tuning of clustering parameters for CLD and LAr ECal
  - shows performance comparable to pre-existing dedicated algorithms
- versatile clustering algorithm for a variety of different calorimeter technologies



E.Brondolin

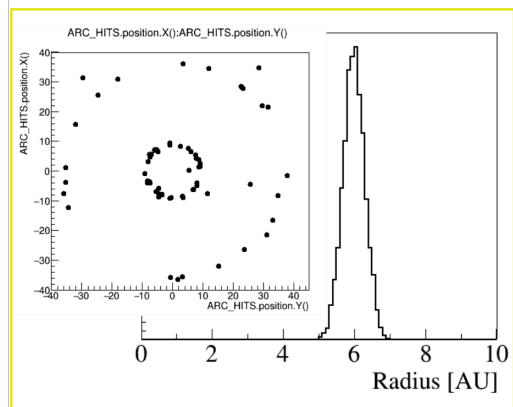
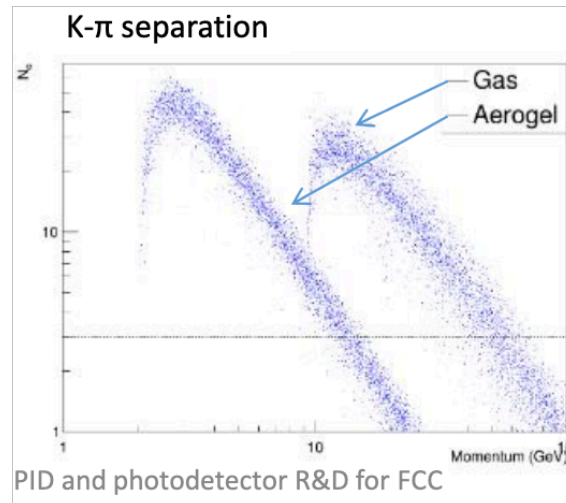
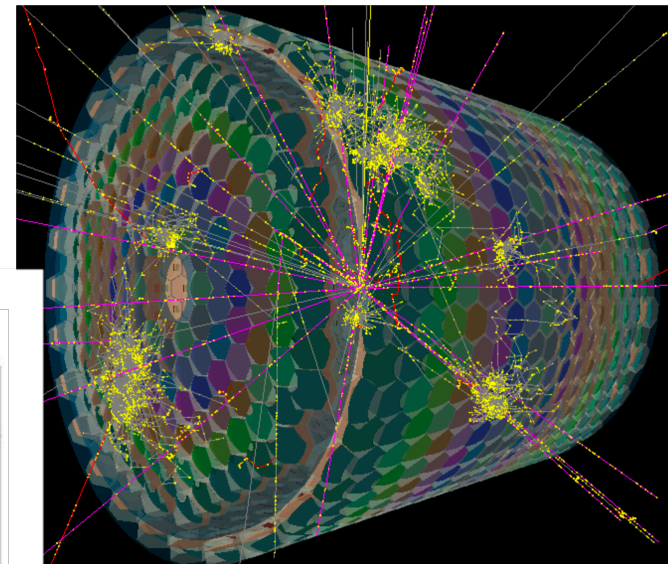


# ParticleID performance with the ARC

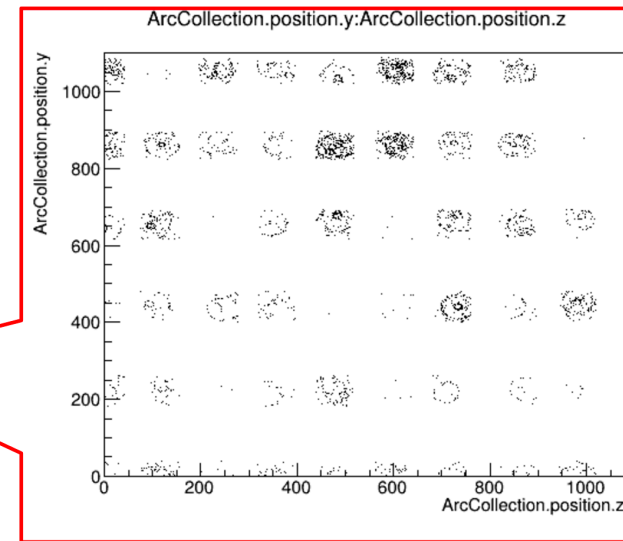
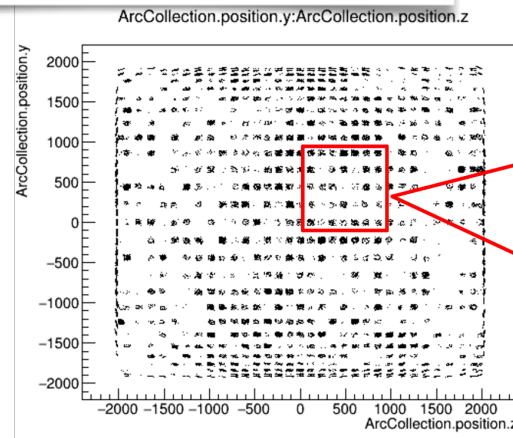
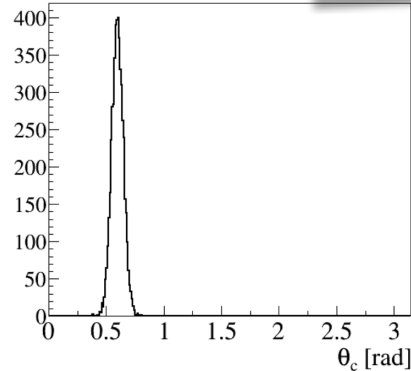
a novel GAUDI algorithm in Key4hep

A. Tolosa Delgado

- can simulate full events in **CLD w/ ARC** with dddim (DD4hep)
- standalone reconstruction w/ inverse ray-tracing exists for single cell
  - should provide excellent K-pi separation from 2-50 GeV
- ongoing work: full ARC reconstruction in Gaudi - aim for end of summer



Inverse ray-tracing

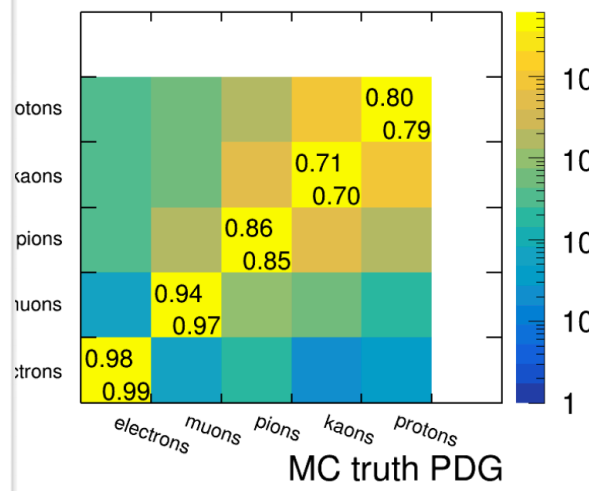
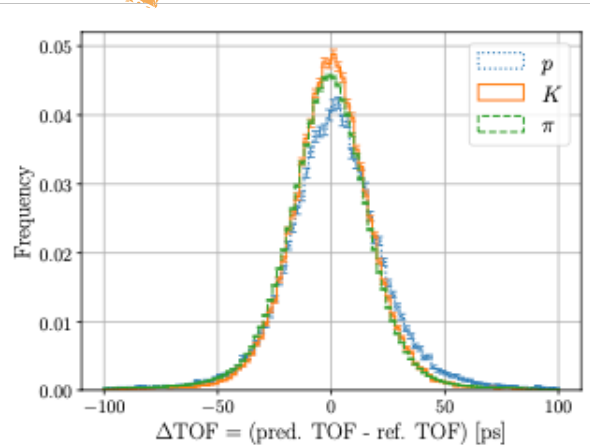
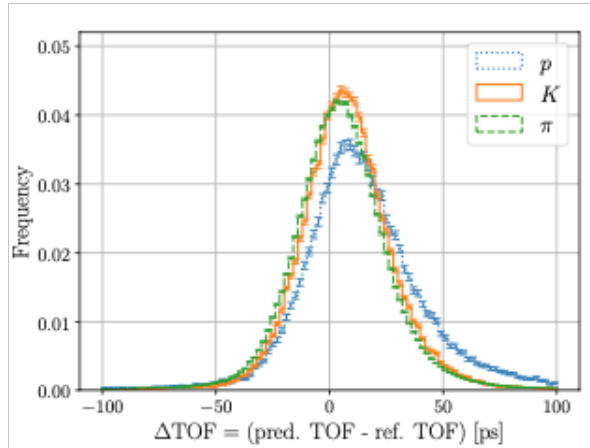
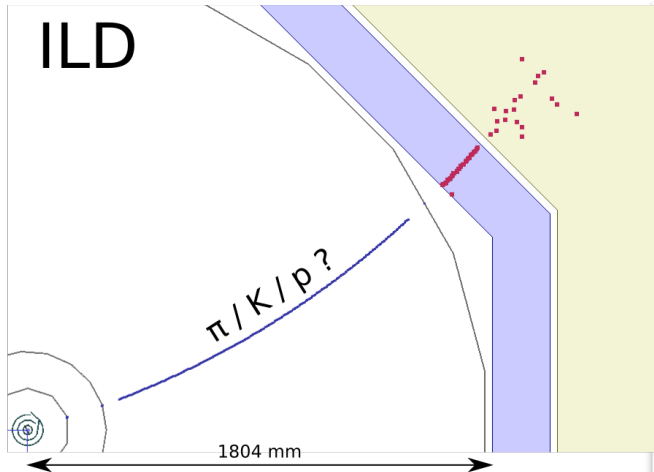
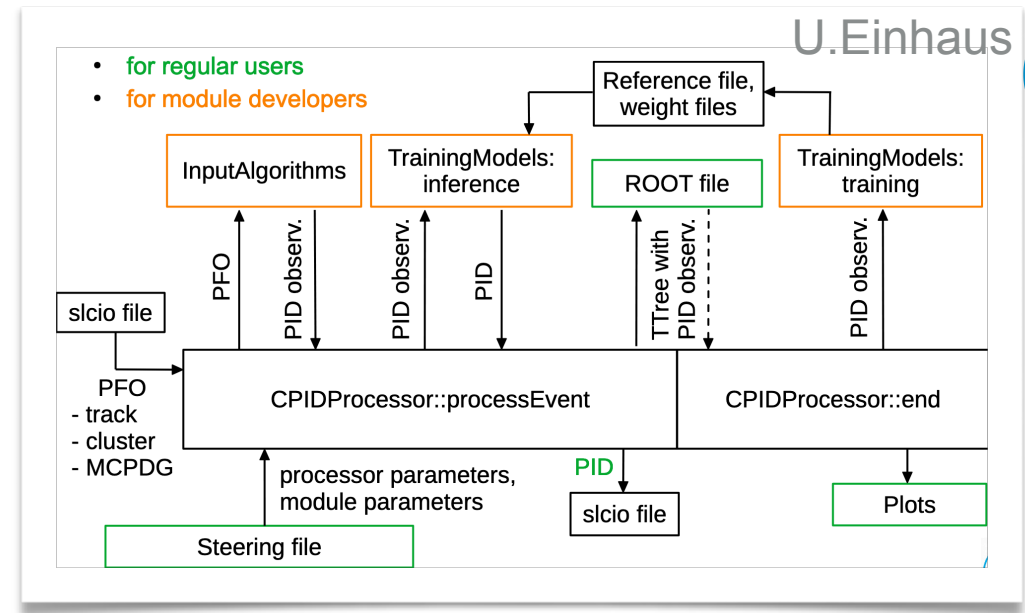


Sagittal cut of the detector (events on the endcaps are compressed in Z, not visible in this projection)

# Particle identification

## Using dE/dx, TOF and AI/ML

- comprehensive particle identification **CPID**:
  - toolkit for ML based PID (in ILD) using: dE/dx, **TOF**, PandoraPFAID, dedicated leptonID
  - applicable to other (all) HET detectors**
- dedicated study of extracting **TOF from the calorimeter in ILD**
  - w/ classical and AI/ML estimator



$$L = \sum_i^{N_{\text{hits}}} L_i = \sum_i^{N_{\text{hits}}} \frac{|z_{i+1} - z_i|}{|\tan \lambda_i|} \sqrt{1 + \tan^2 \lambda_i}$$

B.Dudar, K.Helms



# flavour tagging with deep learning methods

AI/ML in Key4hep

T.Suehara et al

- implemented ParticleTransformer flavour tagging for ILD
  - originally developed for CMS and adapted to FCCee
  - achieve **dramatically better** results than LCFIPlus
- massive improvement w/ more training data at FCC -> to be studied !
- framework (Marlin/Gaudi?) inference work in progress
- started to look into **strange tagging**

## Improvements wrt. LCFIPlus

- Factor (3-9) improvement at ParT from LCFIPlus without any tuning
- Another factor (max 3) improvement by tuning
  - Optimizing input variables
  - Separate embedding for tracks/neutrals

|                   | b-tag 80% eff. |          | c-tag 50% eff. |          |
|-------------------|----------------|----------|----------------|----------|
| background        | c jets         | uds jets | b jets         | uds jets |
| +LCFIPlus (BDT)   | 6.3%           | 0.79%    | 7.4%           | 1.2%     |
| *ParT (initial)   | 1.3%           | 0.25%    | 1.0%           | 0.43%    |
| **ParT (improved) | 0.48%          | 0.14%    | 0.86%          | 0.34%    |

Tracks embed

Neutrals embed

Pairs embed

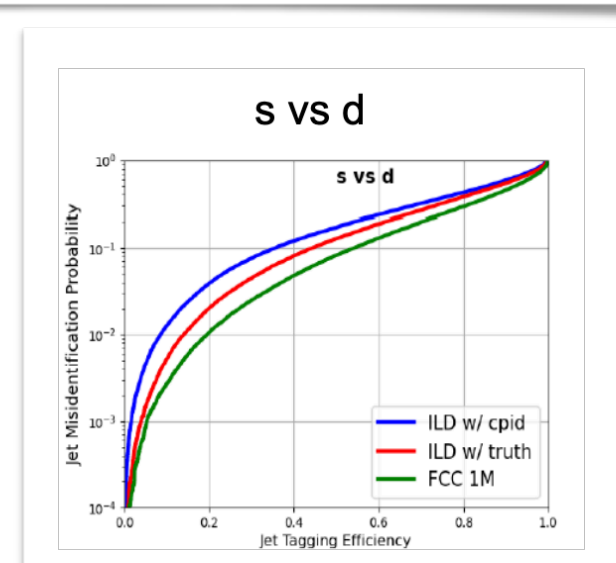
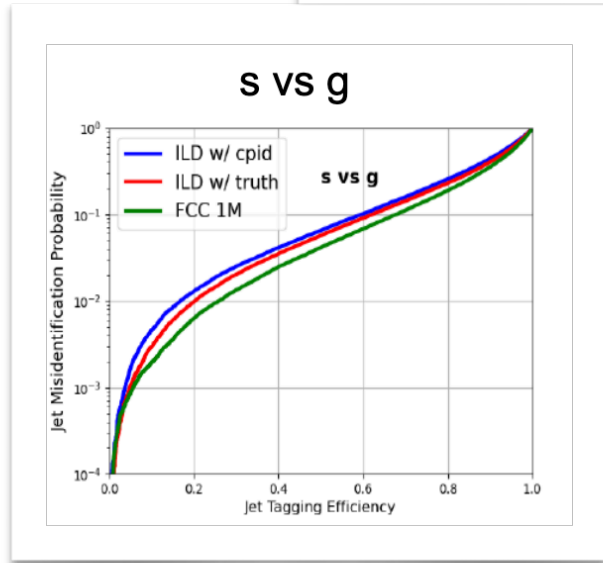
ILC Simulation - unsorted sample - 20 Epochs

ILC Simulation - unsorted sample - 20 Epochs

Suehara et al., ILD Software and Analysis meeting, 2 Oct. 2024, page 7

| Sample / sample size        | b-tag 80% eff. |          | c-tag 50% eff. |          |
|-----------------------------|----------------|----------|----------------|----------|
| background                  | c jets         | uds jets | b jets         | uds jets |
| ILD full-sim 1M (optimized) | 0.48%          | 0.14%    | 0.86%          | 0.34%    |
| FCCee Delphes 1M (reduced)  | 0.47%          | 0.12%    | 0.64%          | 0.10%    |
| FCCee Delphes 1M (full)     | 0.21%          | 0.054%   | 0.36%          | 0.059%   |
| FCCee Delphes 4M            | 0.045%         | 0.025%   | 0.20%          | 0.033%   |
| FCCee Delphes 6M            | 0.014%         | 0.010%   | 0.13%          | 0.022%   |
| FCCee Delphes 8M            | 0.007%         | 0.006%   | 0.076%         | 0.021%   |

We see mild consistency between ILD and FCC!

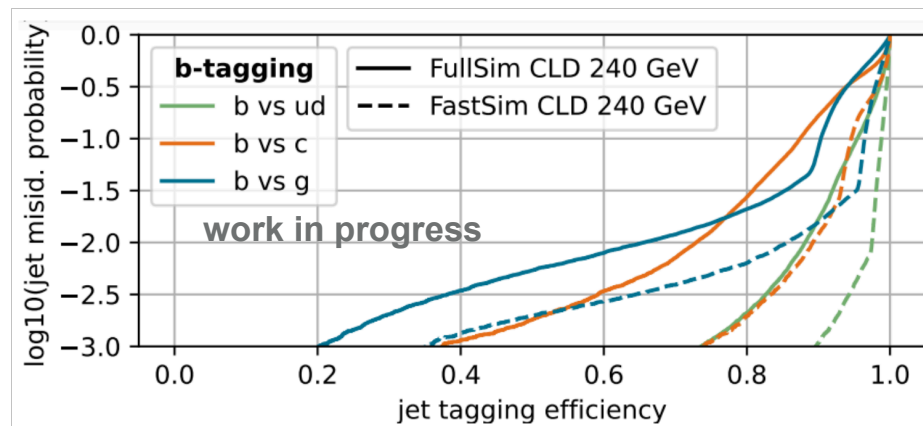
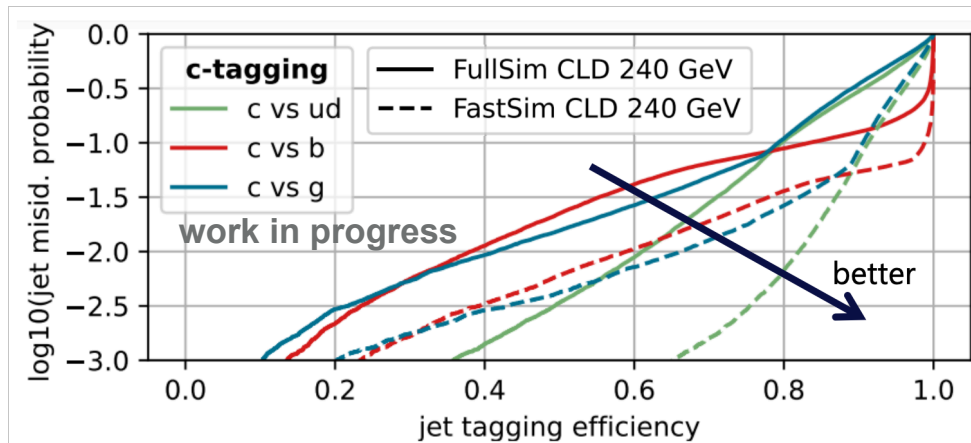
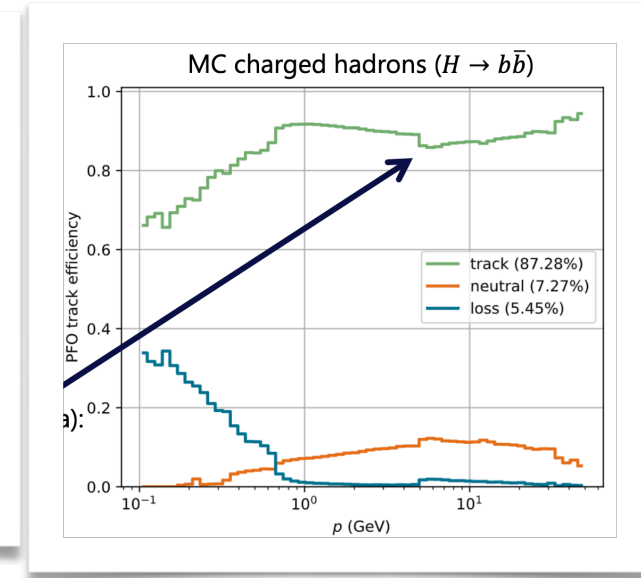
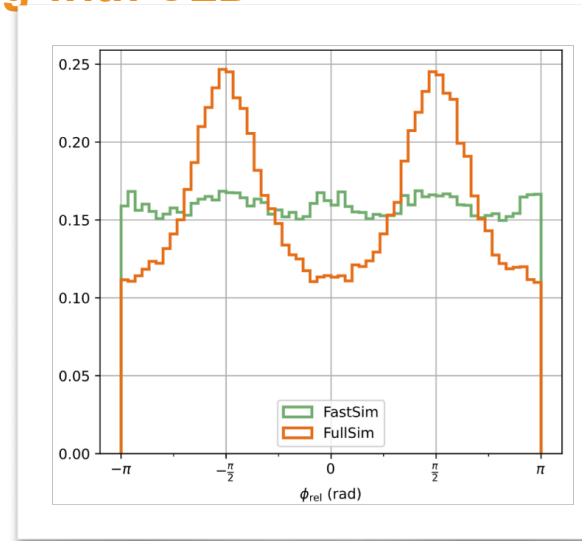


# flavour tagging with deep learning methods

## comparing fast sim and full sim based flavour tagging with CLD

S.Aumiller

- training a PartTransformer for CLD w/
  - Delphes and Fullsim samples
- observe differences in extra neutrals due to
  - split clusters in real algorithm and due to tracks intentionally dropped as no cluster found
- leads to significant differences in tagging performance

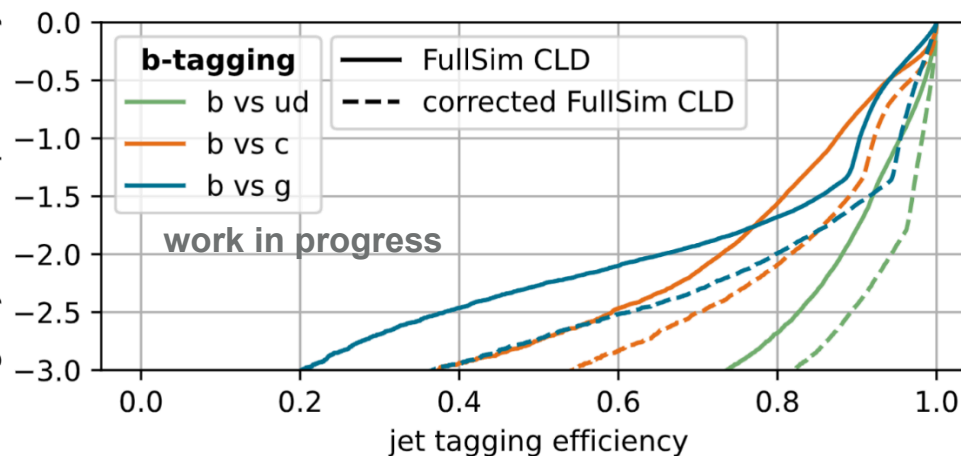
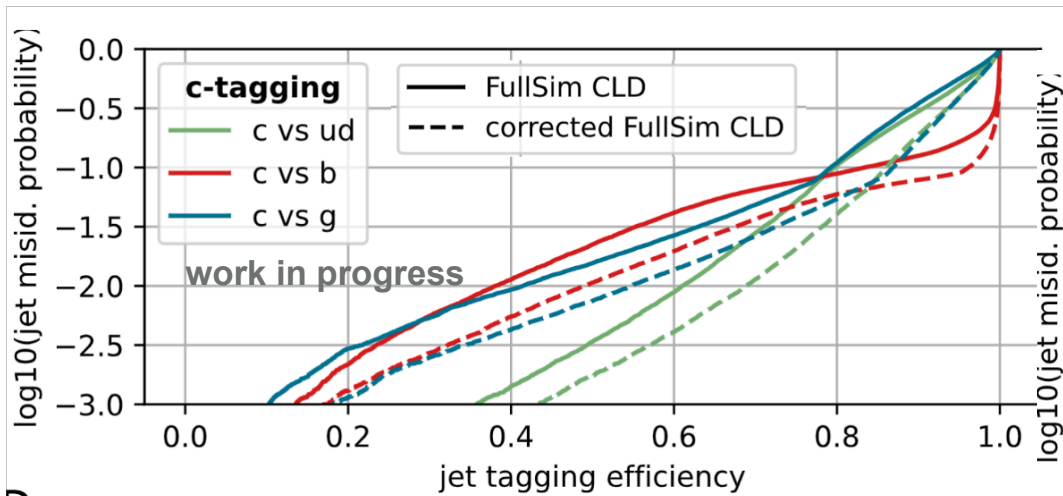
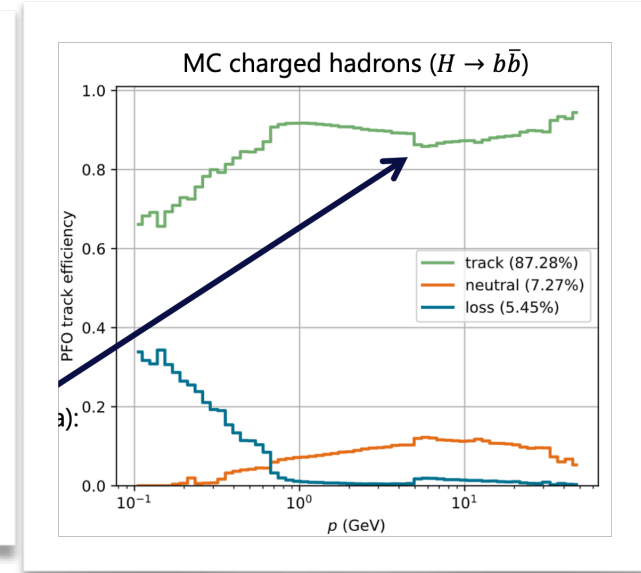
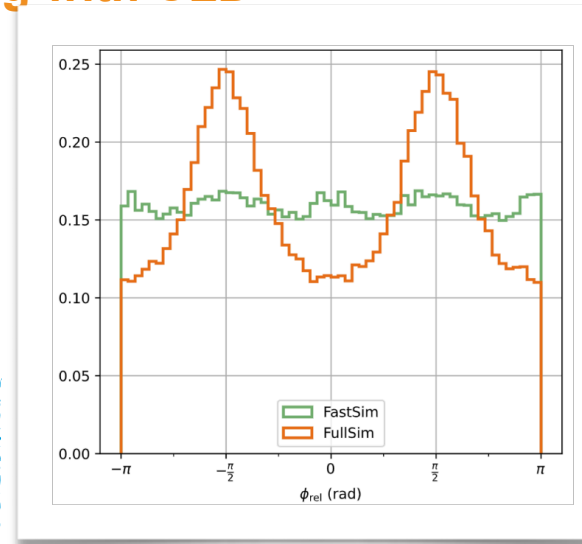


# flavour tagging with deep learning methods

S.Aumiller

## comparing fast sim and full sim based flavour tagging with CLD

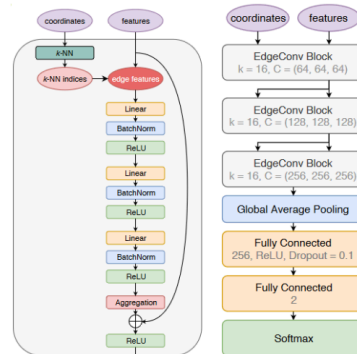
- training a PartTransformer for CLD w/
  - Delphes and Fullsim samples
- observe differences in extra neutrals due to
  - can recuperate some of the flavour tag performance with better reconstruction
    - here cheating w/ MCTruth for demonstration



# flavour tagging with deep learning methods

## Flavour Taggers

- Long-time (> 10 a) standard tool: LCFIPlus, now out-of-date with BDT and b/c/light tag
- A number of new flavour taggers have been developed, making use of neural networks and advanced machine learning
  - ParticleNet (ILD/CLD) [Meyer]
  - ParticleNet (IDEA) [Garcia e.a.]
  - ParticleNet (CEPC baseline) [Ruan e.a.]
  - Particle Transformer (ILD, IDEA) [Suehara]
  - Particle Transformer (CLD, Si-IDEA) [Aumiller e.a.]
- Adding strange-tag via PID (kaonID), possibly individual u- and d-tag via FSR
- Systematic uncertainties on flavour taggers?
  - calibrate at Z-pole with double-tag method and  $10^{12}$  events (FCC-ee) or some  $10^9$  (LC)
  - limits to detector adaptations to center-of-mass energy?



### Jet Flavour Tagging at FCC-ee with a Transformer-based Neural Network: DeepJet Transformer

Freya Blekman <sup>1,2,4</sup>, Florencia Canelli <sup>3</sup>, Alexandre De Moor <sup>1</sup>, Kunal Gautam <sup>1,3</sup>, Armin Ilg <sup>3</sup>, Anna Macchiolo <sup>3</sup>, Eduardo Ploerer <sup>1,3</sup>

fast simulation (Delphes) of IDEA concept

D.Jeans

- large number of ML flavour taggers exist for HET detectors
- need to systematically benchmark and understand the performance of each algorithm (on each detector)

# Particle Flow with Machine Learning

T.Suehara



## AI/ML in Key4hep

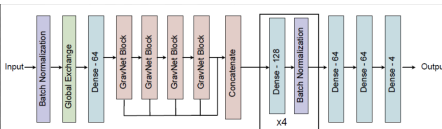
- started to develop deep neural networks for particle flow in the ILD detector
- using GravNet and Object Condensation
- after early, promising results **made significant progress**
- better than PandoraPFA** in one selected metric

### Particle flow with DNN: introduction

- Separation of cluster at calorimeter
  - Charged or neutral cluster
- Essential for jet energy resolution
- Current algorithm: PandoraPFA
  - Combination of various process
  - Not easy to optimize or adding more info
- CMS HGCal clustering
  - Similar to ILD calo
  - Good for starting point

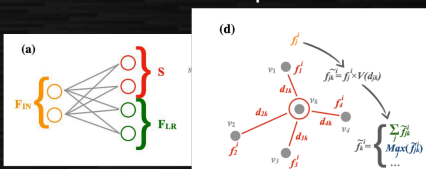
### PFA: clustering algorithm

- Input: position/energy/timing of each hit
- Output: virtual coordinate and  $\beta$  for each hit



#### GravNet [arXiv:1902.07987](https://arxiv.org/abs/1902.07987)

- The virtual coordinate (S) is derived from input variables with simple MLP
- Convolution using "distance" at S (bigger convolution with nearer hits)
- Concatenate the output with MLP



#### Object Condensation (loss function) [arXiv:2002.03605](https://arxiv.org/abs/2002.03605)

$$L = L_p + s_c(L_\beta + L_V)$$

- Condensation point: The hit with largest  $\beta$  at each (MC) cluster
  - $L_V$ : Attractive potential to the condensation point of the same cluster and repulsive potential to the condensation point of different clusters
  - $L_\beta$ : Pulling up  $\beta$  of the condensation point
  - $L_p$ : Regression to output features
- 

### Results on efficiency and purity

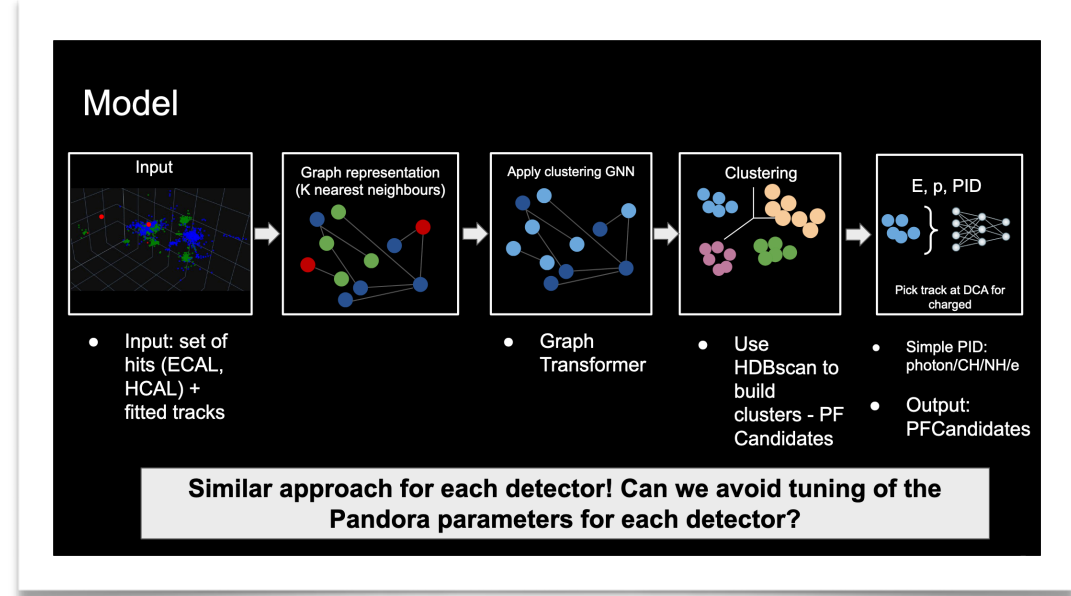
| Algorithm train/test           | Electron eff. | Pion eff. | Photon eff. | Electron pur. | Pion pur. | Photon pur. |
|--------------------------------|---------------|-----------|-------------|---------------|-----------|-------------|
| GravNet 10 taus/10 taus        | 99.1%         | 96.5%     | 99.0%       | 91.8%         | 98.9%     | 97.1%       |
| PandoraPFA 10 taus             | 99.3%         | 94.0%     | 99.1%       | 91.8%         | 94.6%     | 97.2%       |
| GravNet jets/jets              | 94.5%         | 93.1%     | 95.2%       | 94.6%         | 93.2%     | 92.4%       |
| PandoraPFA jets                | 80.2%         | 90.4%     | 79.0%       | 75.0%         | 90.6%     | 77.7%       |
| PandoraPFA jets (ILCSof truth) | 96.7%         | 95.5%     | 96.4%       | 97.1%         | 90.4%     | 97.7%       |

At least in our measure, performance of GravNet-based algorithm exceeds PandoraPFA → Promising as full PFA (but energy regression to be done)  
 Definition of MC truth clusters needs to be tuned (see ILCSof truth)

# Particle Flow with Machine Learning

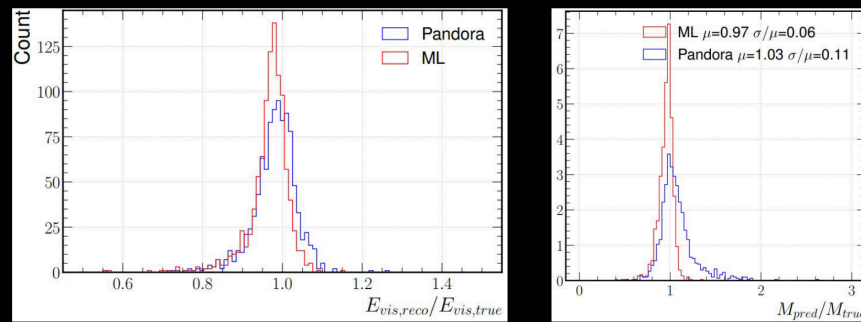
## AI/ML in Key4hep

- alternative approach ML-PFA w/ GNN using full simulation for CLD
- also observe very **promising results** - **better** than the well established **PandoraPFA** on simplified events
  - would be interesting to see in on more realistic collision events (ee->uds ?)



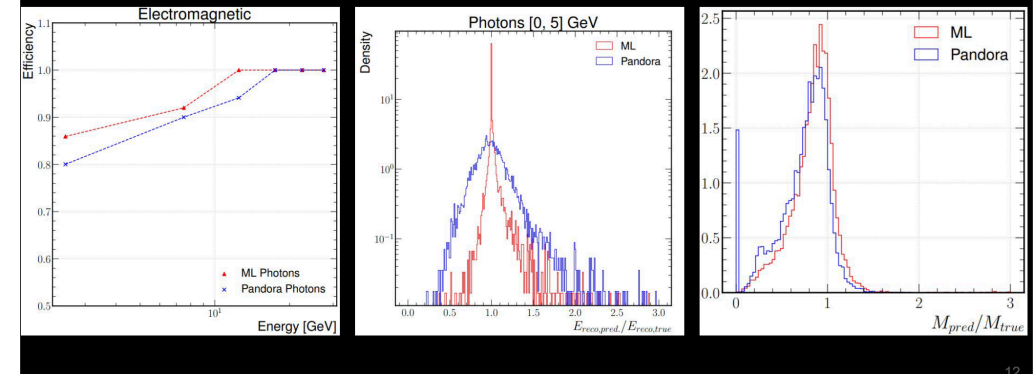
### Results - 10-15 particles dataset

- Better clustering efficiency, energy correction for neutrals leads to better invariant mass resolution (here, particles species are present equally in the same quantity - not the case for physics events!)



### Results: $K_S \rightarrow \pi^0 \pi^0$

- Not feasible to do direction regression - weighted average of the hits from (0,0,0) is a better solution - similar to Pandora

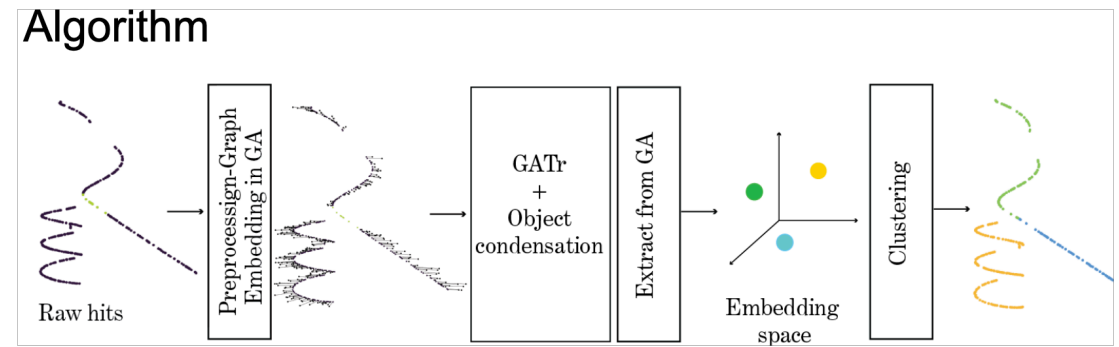
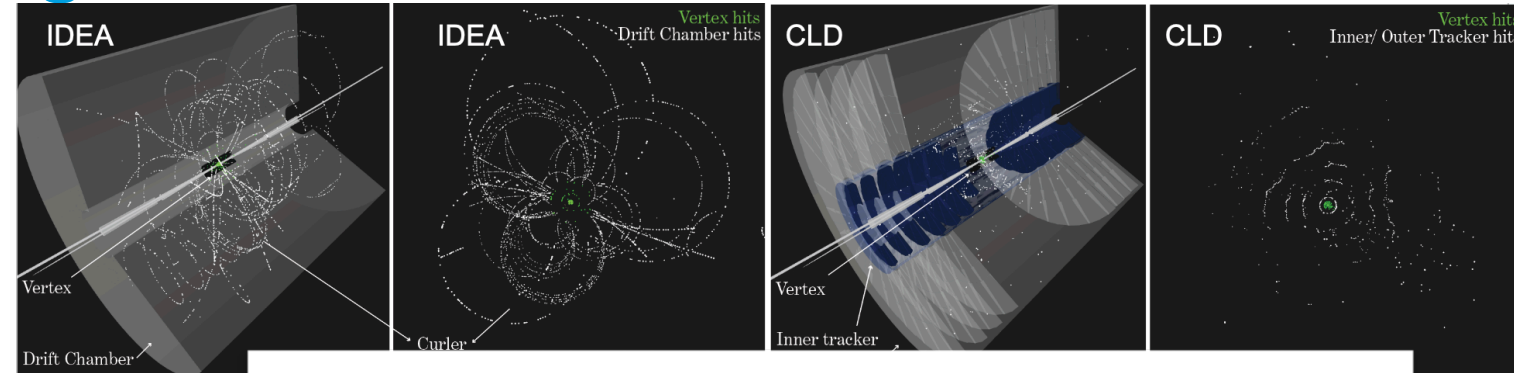


# AI/ML based Pattern Recognition

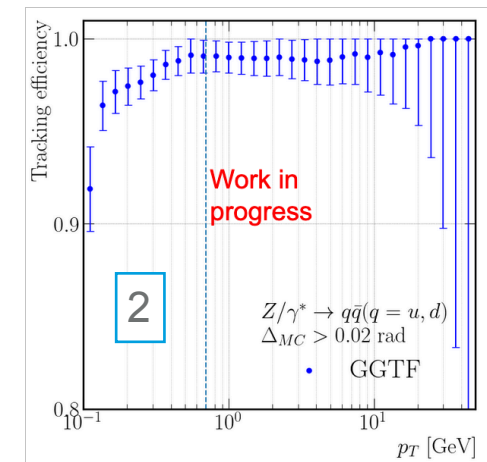
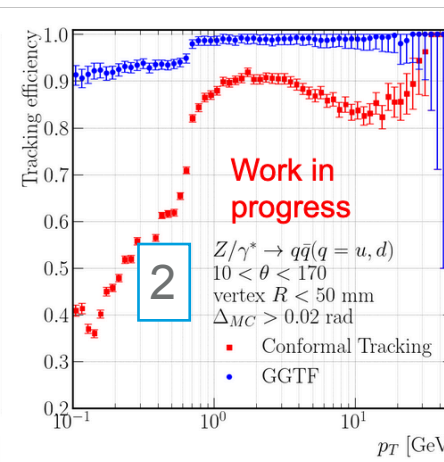
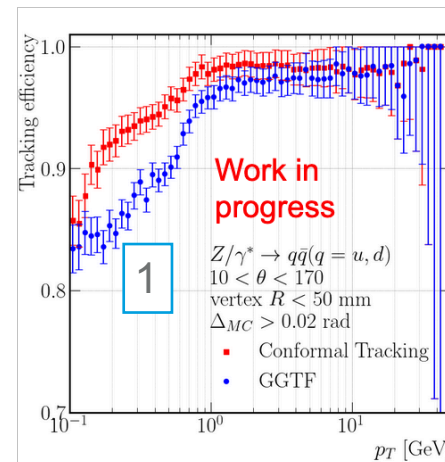
D.Garcia

## for CLD and IDEA

- goal: ML based track finding algorithms that is largely geometry and detector independent
- developed GGTF for IDEA and CLD !
- for CLD observe higher efficiency but lower purity compared to ConformalTracking
- works also nicely for IDEA w/ drift chamber
- implemented in Key4hep-GAUDI



1: hit purity > 75%  
2: hit purity & eff. > 50%



# Benchmarking

## Algorithms and eventually detector concepts

- ultimate physics performance is a combination of detector and reconstruction performance
- for developing optimised detectors benchmarking is absolutely indispensable
- started a new package *k4DetPerformance* w/
  - tracking performance benchmarks
    - residuals, pulls, resolutions, efficiencies
    - ...
- in Key4hep-Gaudi that works for all HET detectors
- should be extended to PFA, favour tagging, PID ...

G.Sadowski

### k4DetPerformance: A Framework for Tracking Performance Studies in Full Simulation Environments

Gaëlle Sadowski<sup>1</sup>, Victor Schwan<sup>2</sup>, Leonhard Reichenbach<sup>3</sup>  
<sup>1</sup>CNRS, University of Strasbourg, <sup>2</sup>DESY, University of Hamburg, <sup>3</sup>CERN, University of Bonn

#### Abstract

etPerformance is a framework designed to study tracking performance within full simulation environments. Initially developed for the CLD detector CC-ee, K4DetPerformance has now been integrated into the Key4hep software stack. Current efforts are focused on extending its applicability to other detectors. The framework requires a complete simulation and reconstruction setup. It employs Condor for running simulations and reconstructions, FCCAnalyses for handling RDataFrame, and matches reconstructed tracks to simulated particles. The framework supports plotting options, including the ability to superimpose plots and ratios for comparative analysis. K4DetPerformance provides a robust solution for tracking performance evaluation and has become an essential tool for detector performance studies across different detector models within the Key4hep software ecosystem.

#### Full Simulation

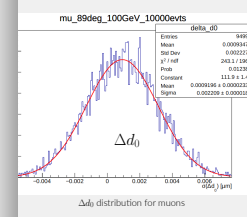
Full simulation refers to a detailed and comprehensive process using Geant4. The key aspects of full simulation include:

- **Realistic modeling** of particle interactions and movement.
- Detailed detector specifications, such as dimensions, materials, and layout.
- Simulation of the entire sequence from particle generation to their interactions with the detectors and the data capture by sensors.

Reconstruction involves interpreting the signals left in the detectors to recreate the original properties of the particles, such as their trajectory, energy, and momentum.

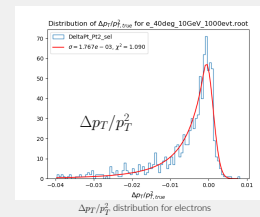
- Simulation with **particle gun** events
  - Single particle events with fixed momentum and  $\theta$  and flat  $\phi$
  - With muons, electrons and pions
  - Simulation and reconstruction are performed using **Condor**

#### Tracking resolution



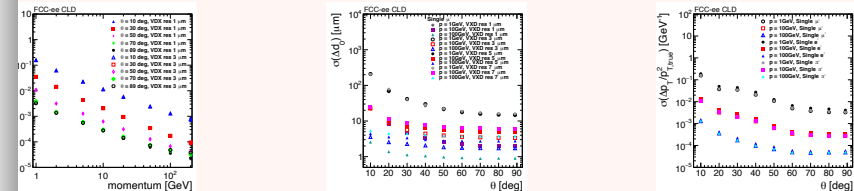
The tracking resolution is determined by comparing reconstructed tracks to their corresponding simulated particles. This is achieved through the following steps using FCCAnalyses for handling RDataFrame:

- Matching reconstructed tracks – simulated particle
- Calculation of resolution:  $\sigma(\Delta = \text{reco} - \text{true})$ 
  - For p and pT, resolution is:  $\sigma(\Delta = \text{reco} - \text{true}) / \text{true}^2$
  - Resolution is the width of the gaussian fit, or crystal ball fit for electron momentum



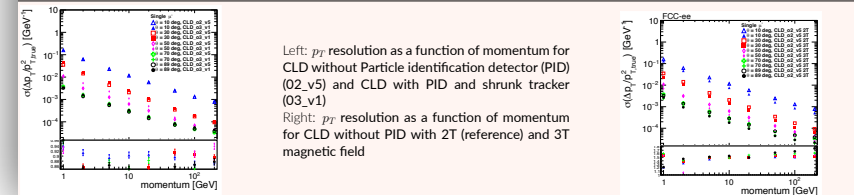
#### Some results for CLD detector

##### Superimposed plots



from left to right:  $p_T$  resolution as a function of momentum for different vertex spatial resolution,  $d_0$  resolution as a function of polar angle  $\theta$  for different vertex spatial resolution,  $p_T$  resolution as a function of polar angle  $\theta$  for  $\mu^-$ ,  $e^-$  and  $\pi^-$

##### Ratio plots



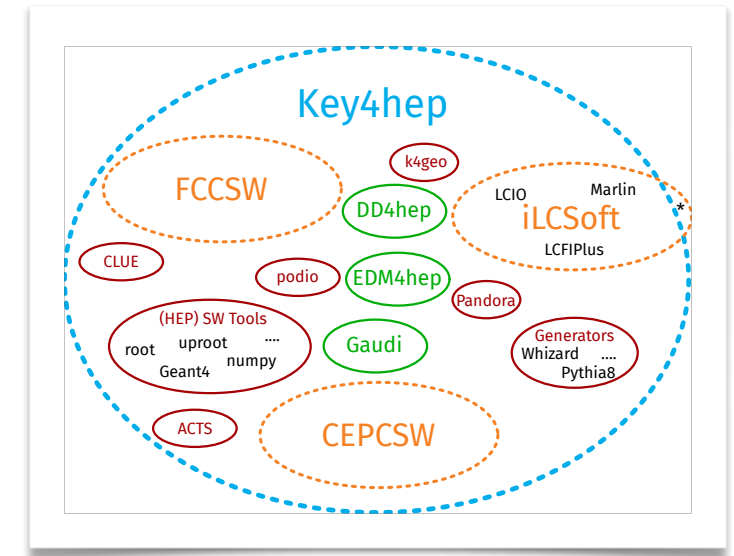
Left:  $p_T$  resolution as a function of momentum for CLD without Particle identification detector (PID) (02\_v5) and CLD with PID and shrunk tracker (03\_v1)

Right:  $p_T$  resolution as a function of momentum for CLD without PID with 2T (reference) and 3T magnetic field



# Summary and Outlook

- **Key4hep** started as a new future collider community wide effort in 2020 to put together a modern turnkey software stack for all future colliders
  - contributors: CEPC, CLIC, FCC, EIC, ILC, LUXE, Muon Collider ...
- existing **standard reconstruction** (linear colliders) can be run in Key4hep w/ **MarlinWrapper** as before - or w/ EDM4hep output
- first **genuine Key4hep/EDM4hep/Gaudi reconstruction algorithms** start to become available (k4Clue, k4ACTS, k4GaudiPandora,...)
- many new developments in (high level) reconstruction tools
  - more and more **ML/AI** based !
- **full simulation and reconstruction** is important for detector optimisation - as well as systematic **benchmarking**



ECFA Report: Reconstruction editors Ulrich Einhaus, Loukas Gouskos, Taikan Suehara

- development of (reconstruction) tools is mostly manpower limited
- vital for future collider community that Key4hep continues to receive funding