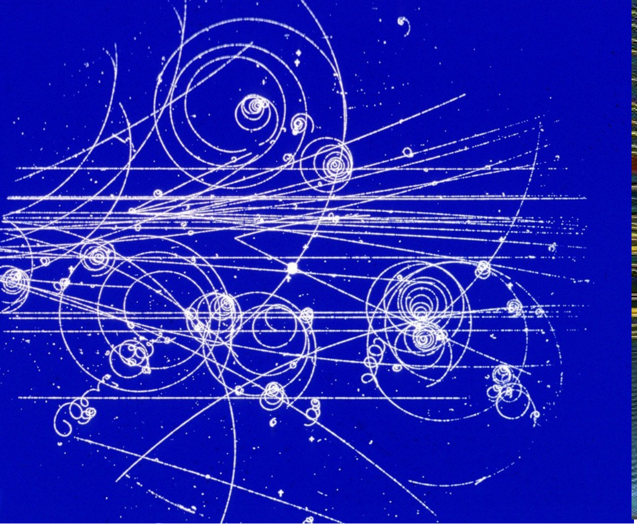
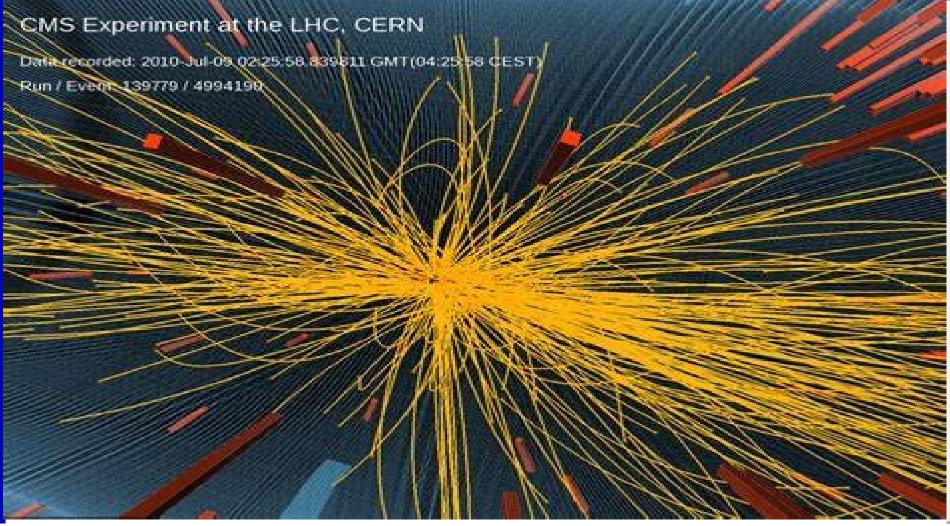
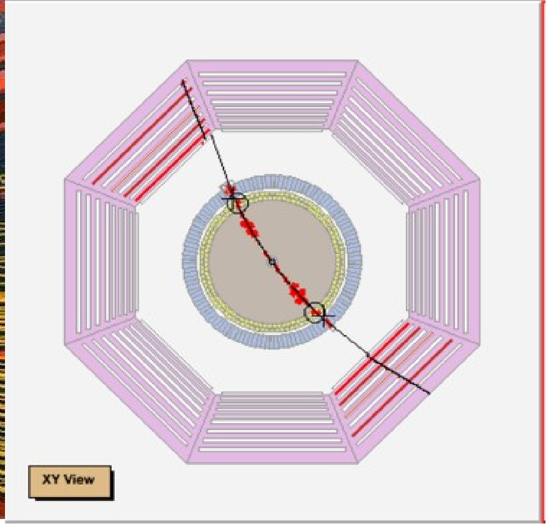






Towards common HEP experiment softwares

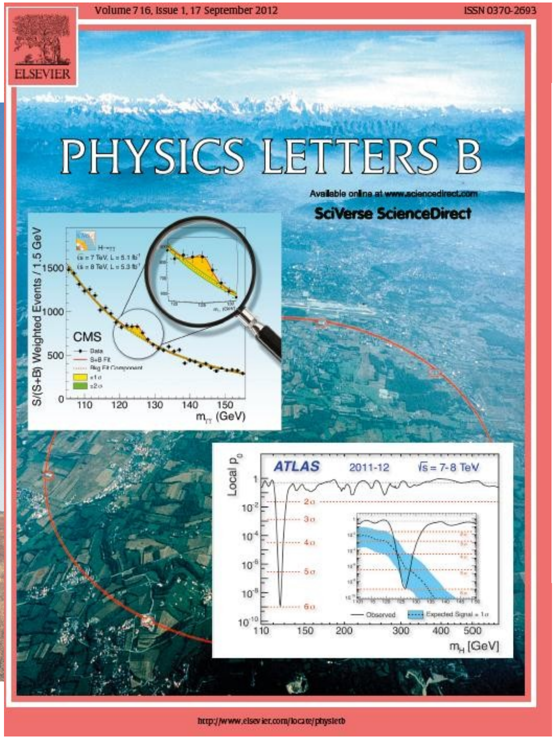
艾小聪 (xiaocongai@zzu.edu.cn), Zhengzhou University

May 05, 2023, IHEP Seminar

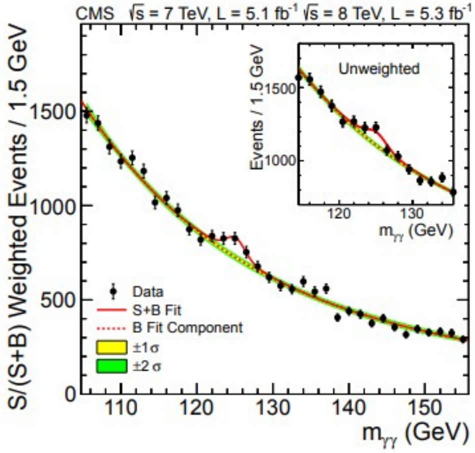
What are HEP experiment softwares about?

A long long road to an exciting paper in particle physics

- O(10) years of efforts of hundreds to thousands of people



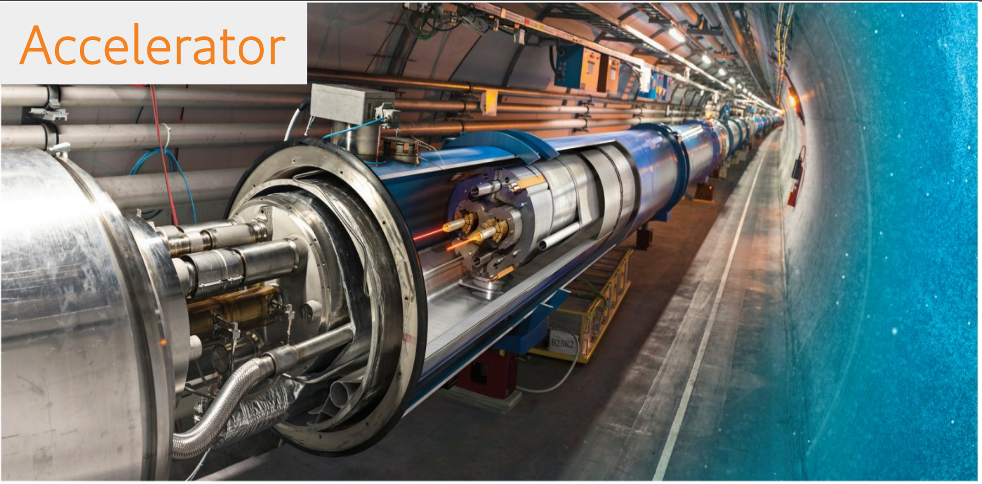
Higgs discovery at CMS



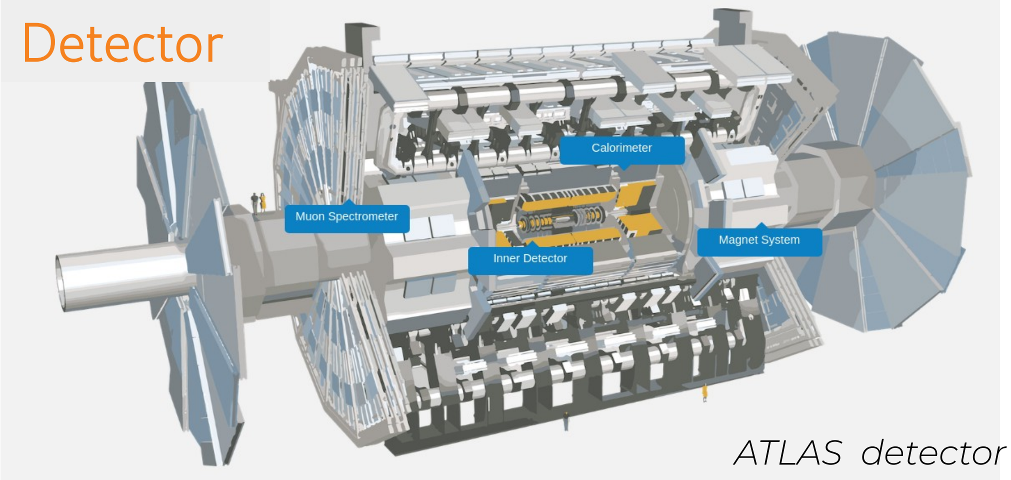
Phys. Lett. B 716 (2012) 30

What makes a particle and nuclear experiment?

Accelerator

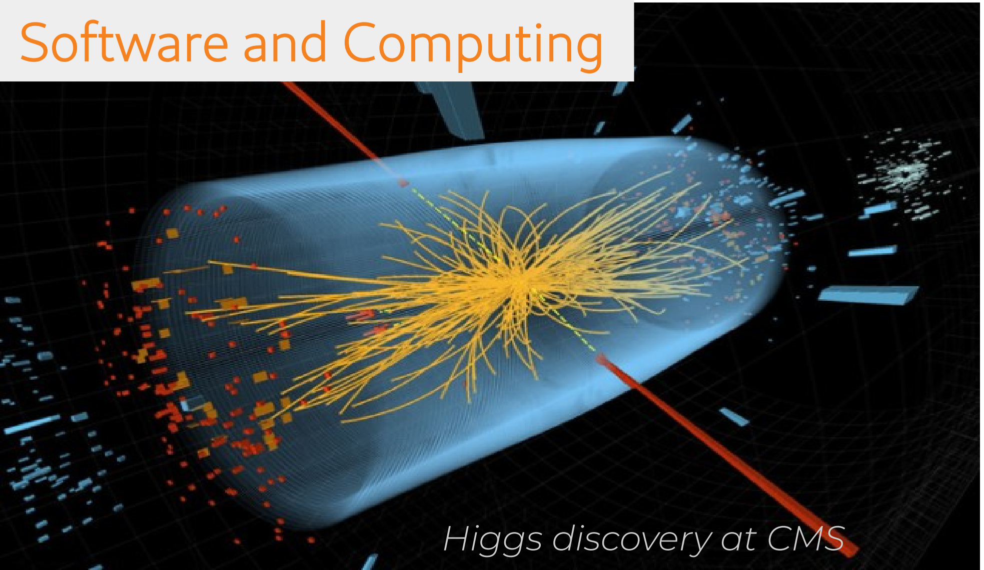


Detector



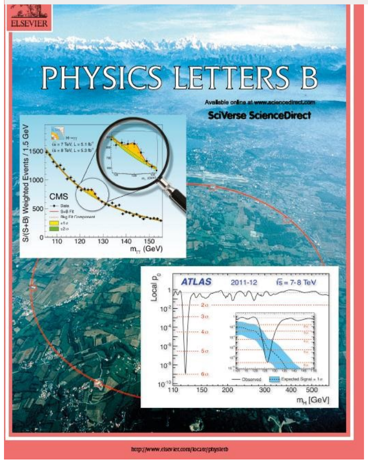
ATLAS detector

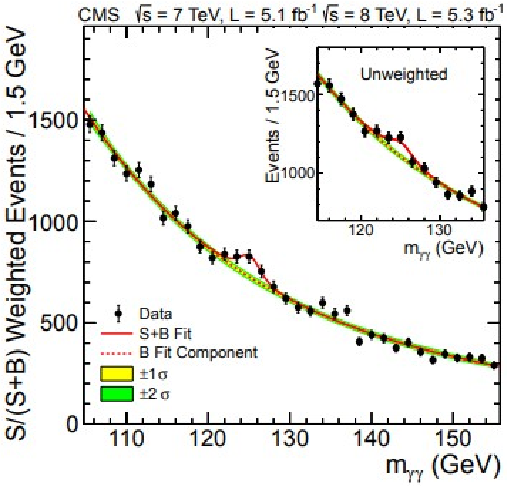
Software and Computing



Higgs discovery at CMS

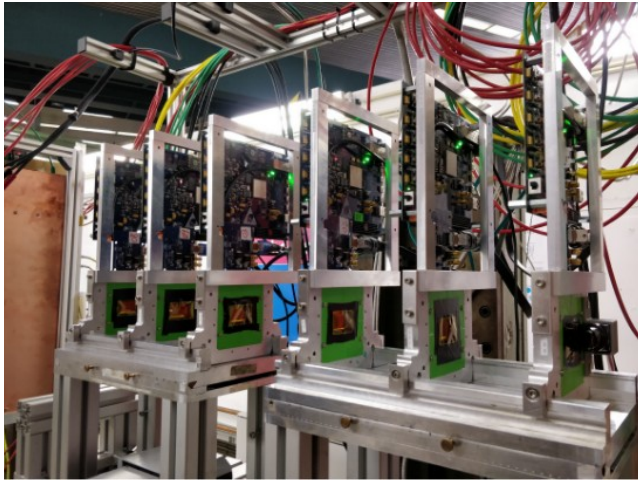
Physics Analysis





What is the role of software and computing in HEP&NP?

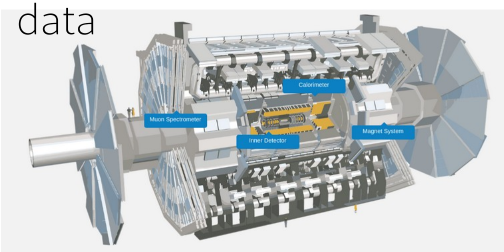
- Guide and support the **design, characterization and construction** of detector through **detector simulation**
- **Turns the recorded data by detectors into physics objects** for ultimate extraction of physics signals
  - Reconstruction, identification, calibration

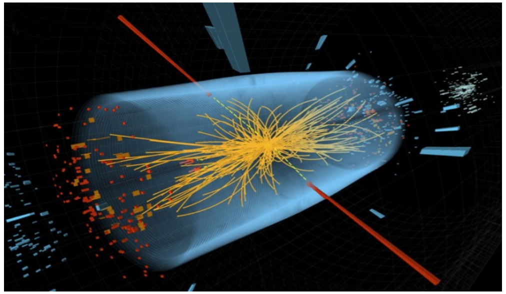


ALPIDE Addenium telescope at DESY

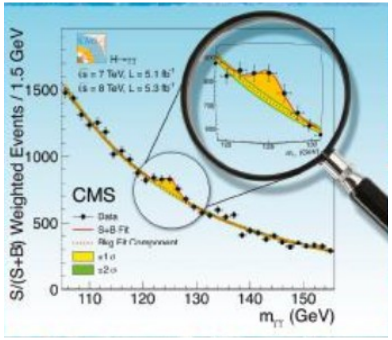
MC: event generation + simulation + digitization

data



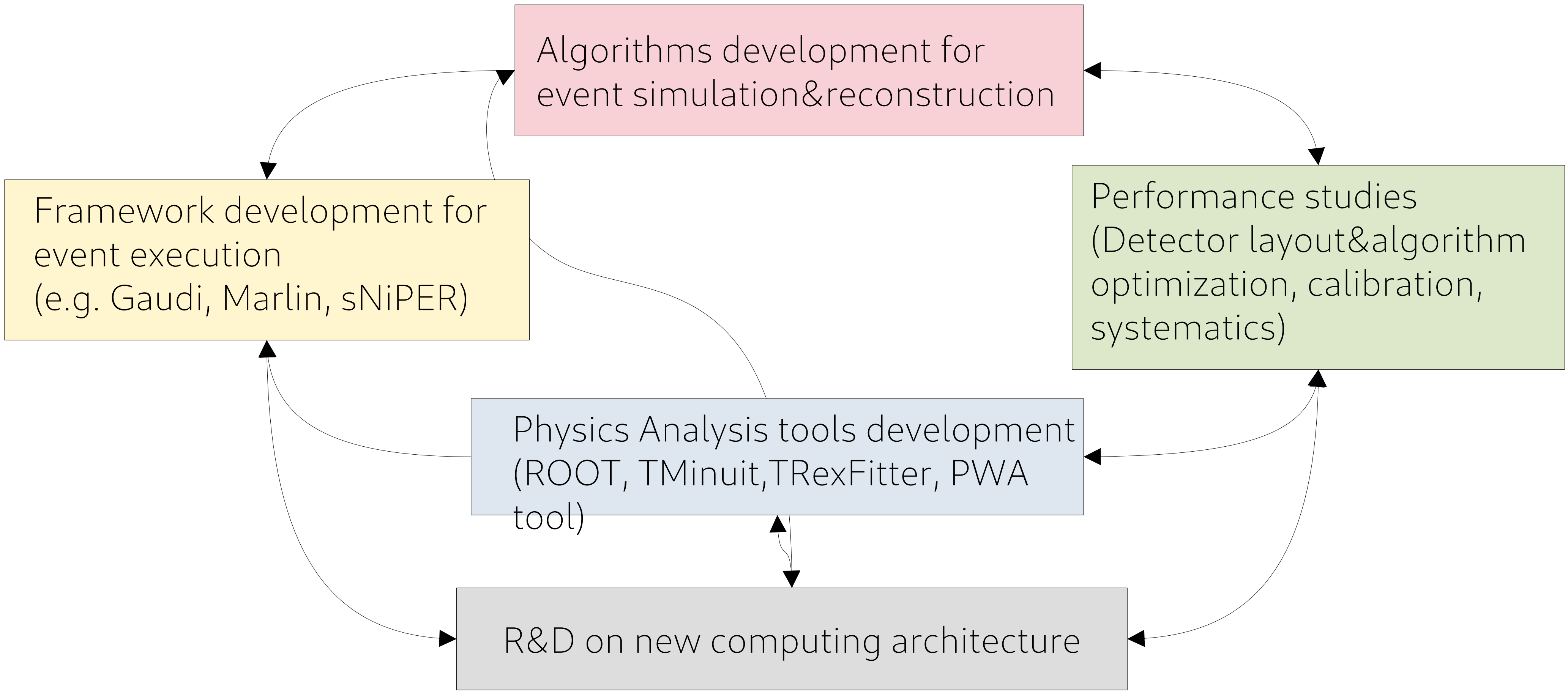


Particle position, drift time, flight time, deposited energy…



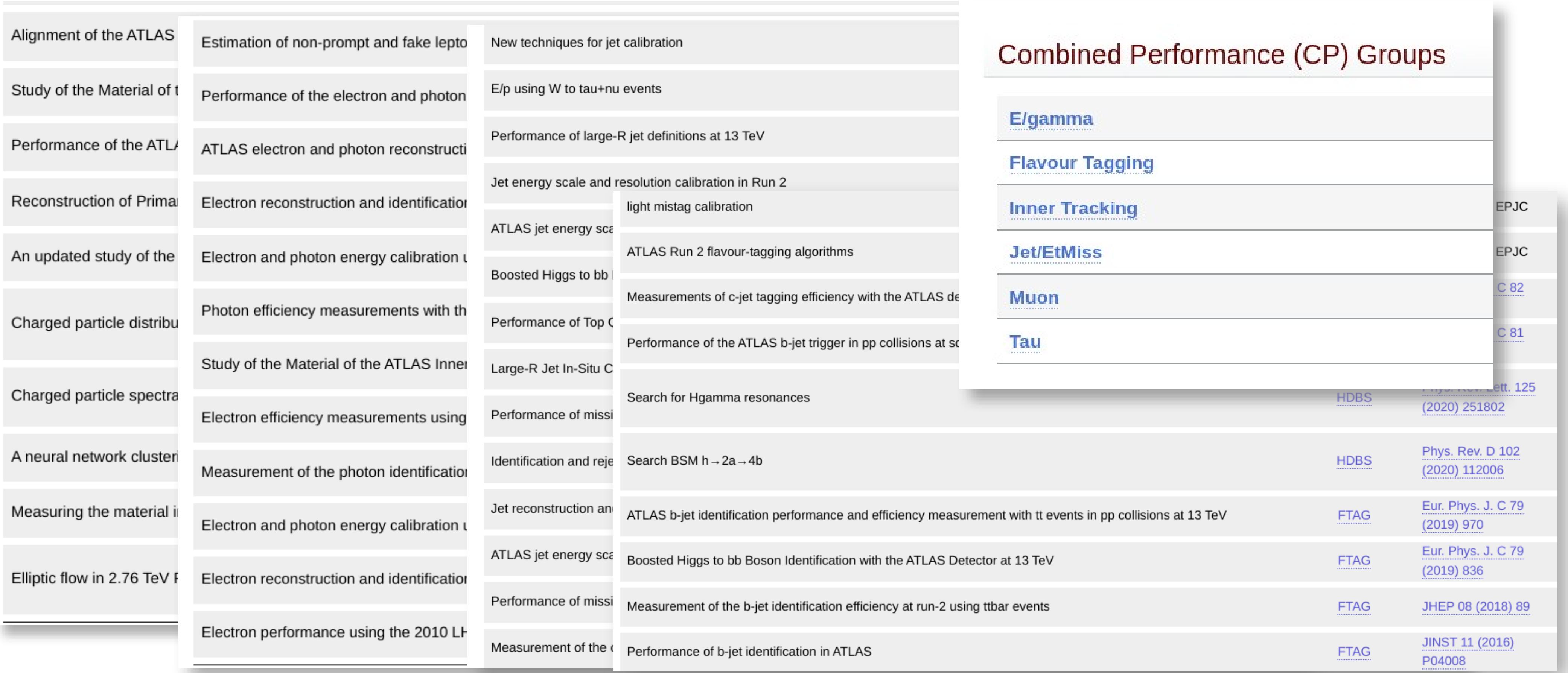
Tracks, vertexs, e, μ, τ, b/c/light-flavor jet, charged hadrons, neutron, K_L, missing energy

HEP software is far more than fixing bugs



HEP software is about physics

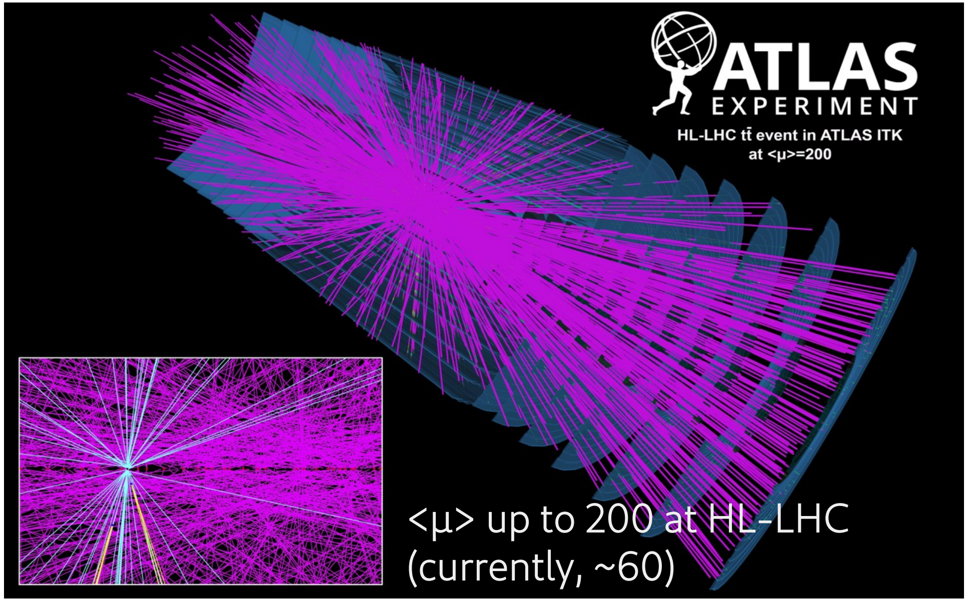
- Efficiency and precision of physics object reconstruction drives the precision of physics measurement

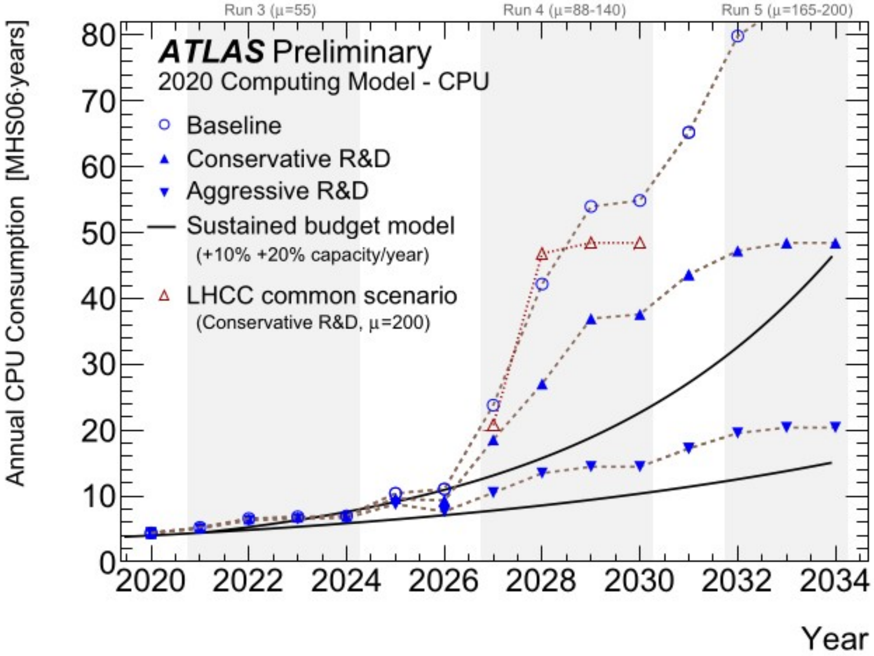


HEP software challenges

Much more dense environment

- Future colliders tend to have much increased luminosity => higher pileup, e.g.
  - <μ> = 200 at HL-LHC, <μ> = 1000 at FCC-hh
- Much **increased combinatorics, data rate and CPU needs**
  - ~7k particles/event at HL-LHC

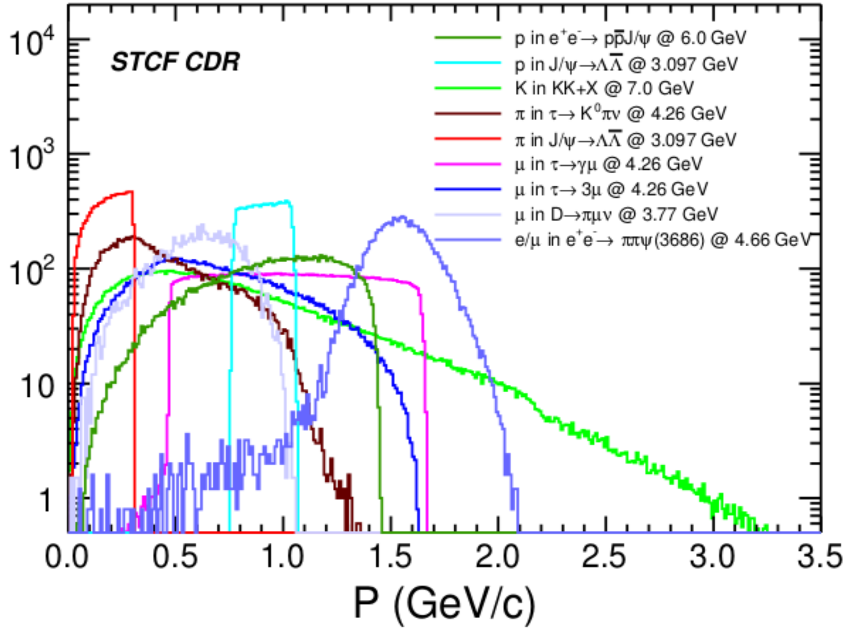




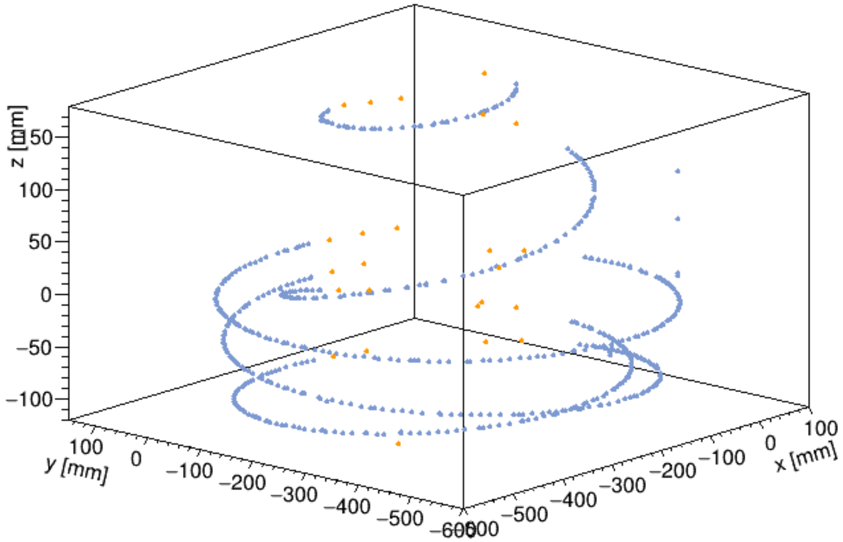
More strict physics requirements

- Tracking of **low p_T tracks** is very important at future flavor factories
  - e.g. tracking eff. > 50/90/99 % with pt > 50/100/300 MeV at STCF (important to probe CPV in $\tau \to K_S \pi \nu_\tau$ and $J/\psi \to \Lambda\bar{\Lambda}$)

Momentum distributions of charged particles

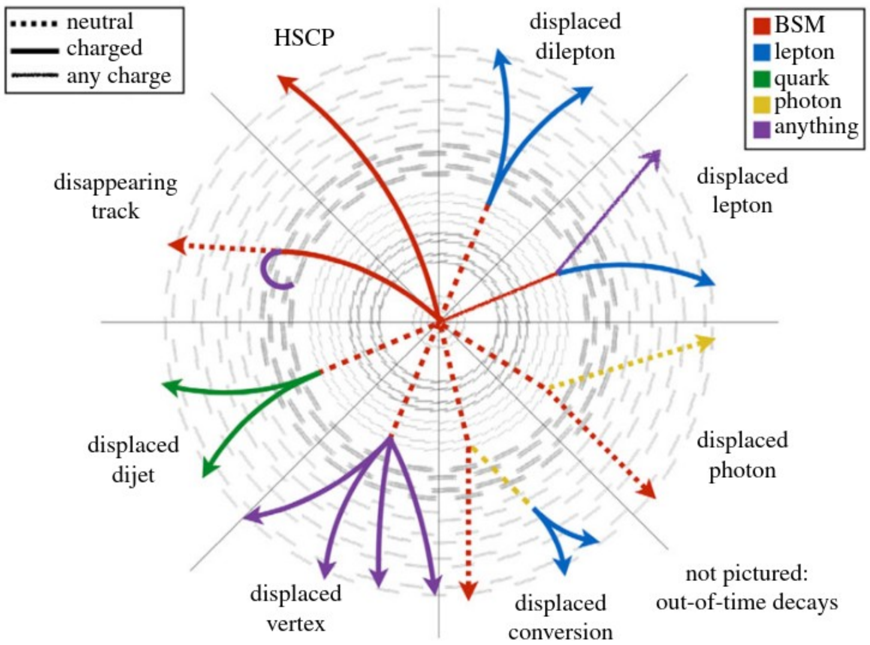


An example of muon trajectory (pT = 100 MeV, theta = 90) at STCF



More complex physics signatures

- Reconstruction of **long-lived particle (LLP) signatures** is import for New Physics search:
  - Displaced tracks
  - Disappearing tracks
  - Anomalous Ionization
  - Magnetic monopole
  - Fractional/multiple Electric Charge



Phil. Trans. R. Soc. A 377, 20190047

More difficult maintenance and extension

- HEP software is typically developed in last century
  - Limited support for new software/detector technology
- Growing manpower for maintenance
  - Software is usually broken with new detectors
  - More and more hacks (often hardcoded)
  - Debug can be painful with hundreds of packages in flight

More difficult maintenance and extension

- HEP software is typically developed in last century
  - Limited support for new software/detector technology
- Growing manpower for maintenance
  - Software is usually broken with new detectors
  - More and more hacks (often hardcoded)
  - Debug can be painful with hundreds of packages in flight

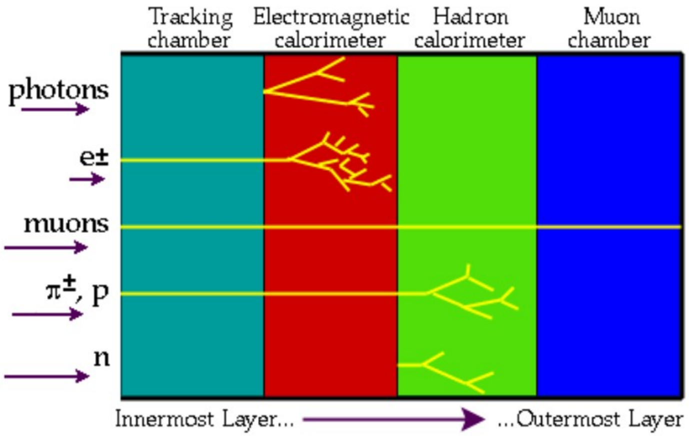
Code that is not maintained will fade away and eventually die…

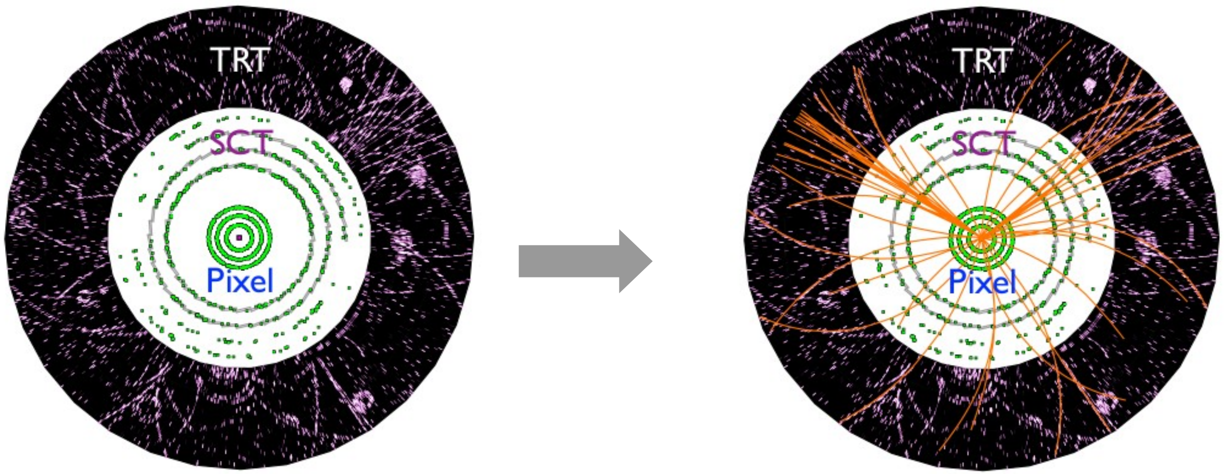
HEP experiments are trending towards maintainable and extendable common softwares, e.g. **common tracking software**!

Why tracking matters?

What is track reconstruction (a.k.a. tracking)?

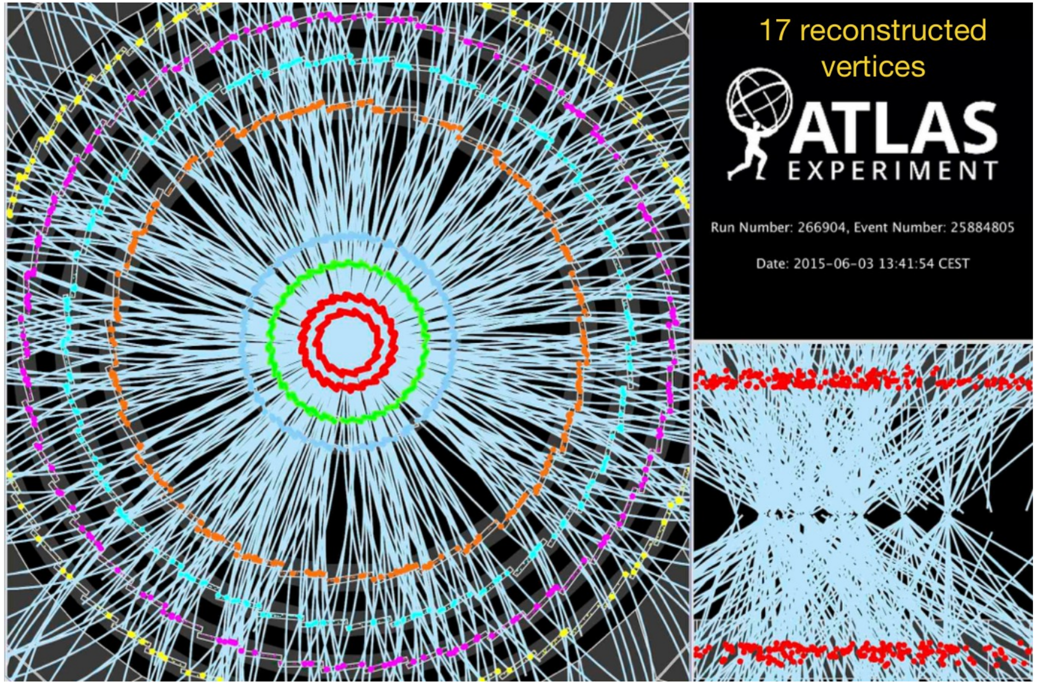
- **Reconstruction** (i.e. track finding) of charged tracks and **measurement** (i.e. track fitting) of their quantities, using the signals of trackers (usually in magnetic field)
  - Position
  - Momentum
  - Charge
  - Vertex
  - Velocity (dE/dx)





Tracking is about vertex reconstruction

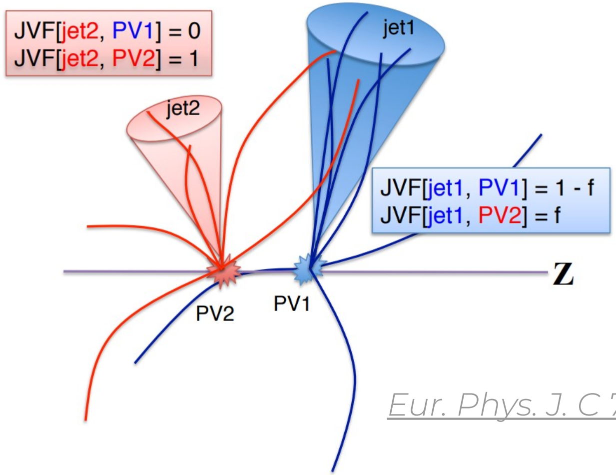
- Primary vertex reconstruction uses estimated track parameters of charged particles as inputs
  - Vertex finding
    - Associate tracks to vertices
  - Vertex fitting
    - Estimate vertex position



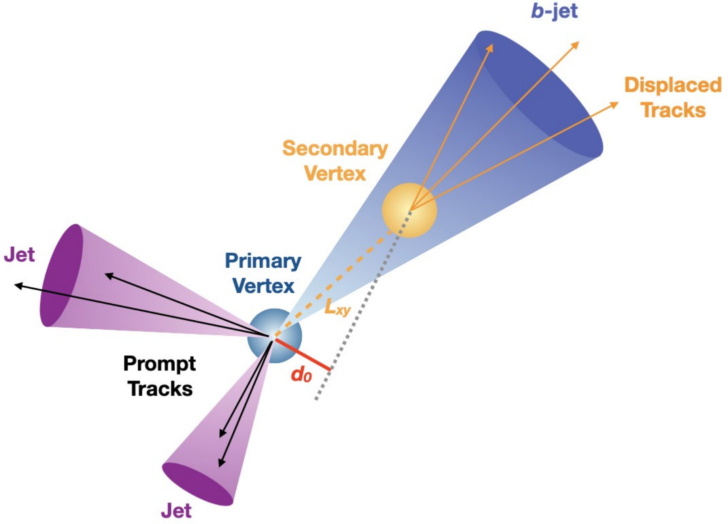
Tens to hundreds of additional proton–proton collisions accompanying the hard-scatter interaction, i.e. pile-up (μ)

Tracks/vertices are not just about charged particles

- Jets and missing energy reconstruction
  - Better p_T resolution for low p_T tracks and angular resolution provided by tracker
  - Tracks/vertices are crucial for pile-up mitigation (needs precise jet-vertex association)
- Jet flavor-tagging (b, c or light-flavor jet)
  - Impact parameters, secondary vertices and length of flight
- Reconstruction of photon conversion vertex
  - Important input for e/γ discrimination



Eur. Phys. J. C 76, 581



Tracking is about detector alignment

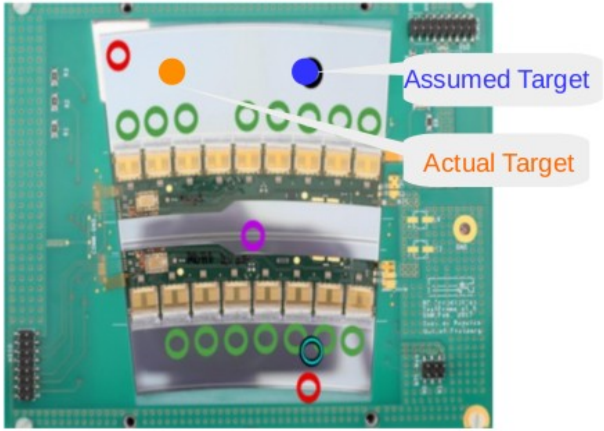
- Misalignment will deteriorate the track resolution, but tracking can notice and correct the misalignment
  - Track-based alignment is a common practice

Local track parameters and global alignment parameters

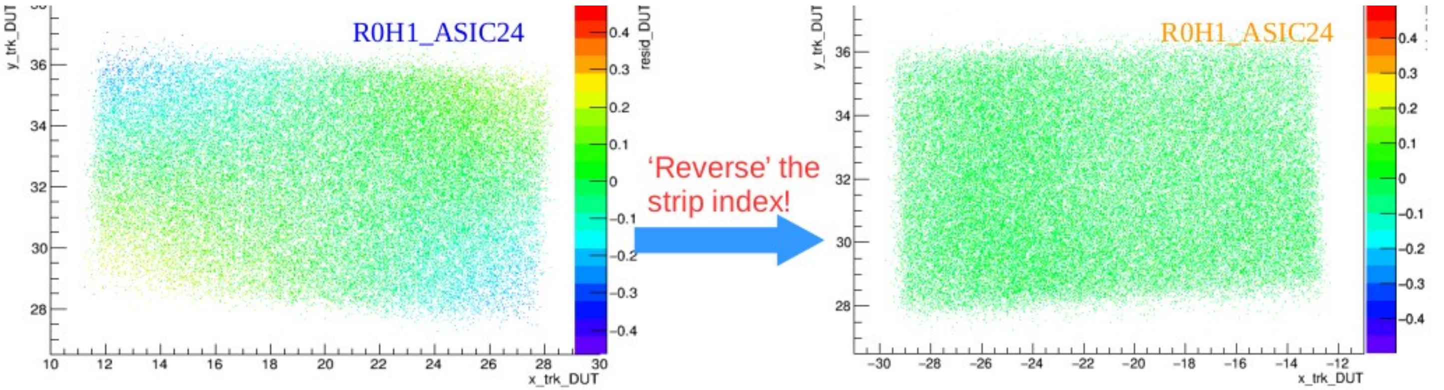
$$\chi^2 = \sum_i {\chi_i}^2 = \sum_i [\vec{m}_i - \vec{h}_i(\vec{x}_i(\vec{\alpha}), \vec{\alpha})]^T V^{-1} [\vec{m}_i - \vec{h}_i(\vec{x}_i(\vec{\alpha}), \vec{\alpha})]$$

$$\frac{d^2\chi^2}{d^2\vec{\alpha}}|_{\vec{\alpha_0}} \Delta\vec{\alpha} = -\frac{d\chi^2}{d\vec{\alpha}}|_{\vec{\alpha_0}}$$

First ATLAS ITk Endcap strip R0 module at testbeam in 2017

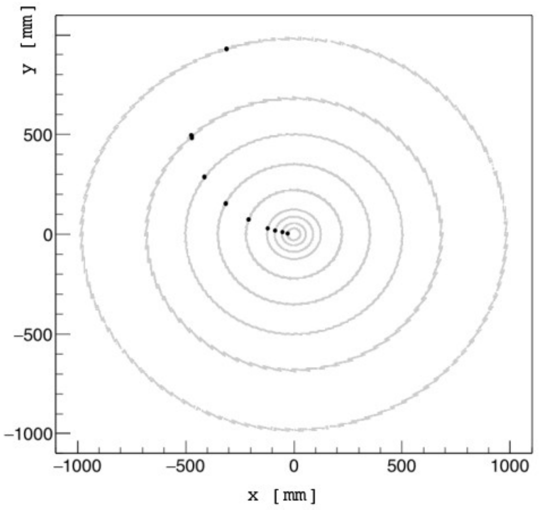


A mistake of moving DUT to wrong target is detected by alignment!

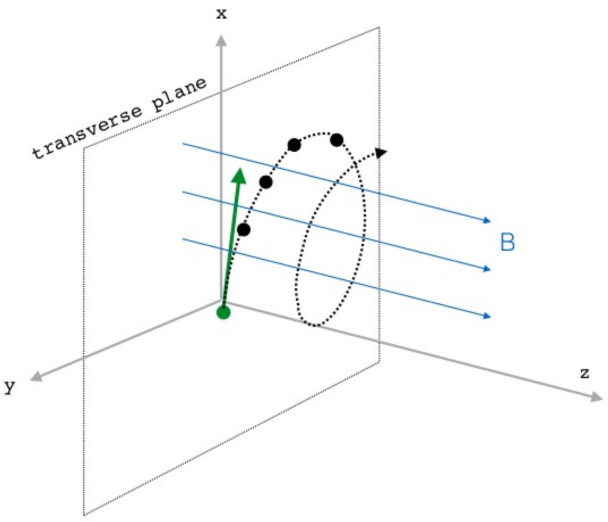


Tracking strategies

A helix trajectory in homogeneous magnetic field



arxiv:1904.06778



$$\frac{d^2\mathbf{r}}{ds^2} = \frac{q}{p}\left[\frac{d\mathbf{r}}{ds} \times \mathbf{B}(\mathbf{r})\right]$$

Track propagation is solved numerically using fourth-order Runge-Kutta-Nyström method

Track parameterization

- Described by five parameters (three for transverse + two for longitudinal)

ATLAS parameterization (no assumption of helix):

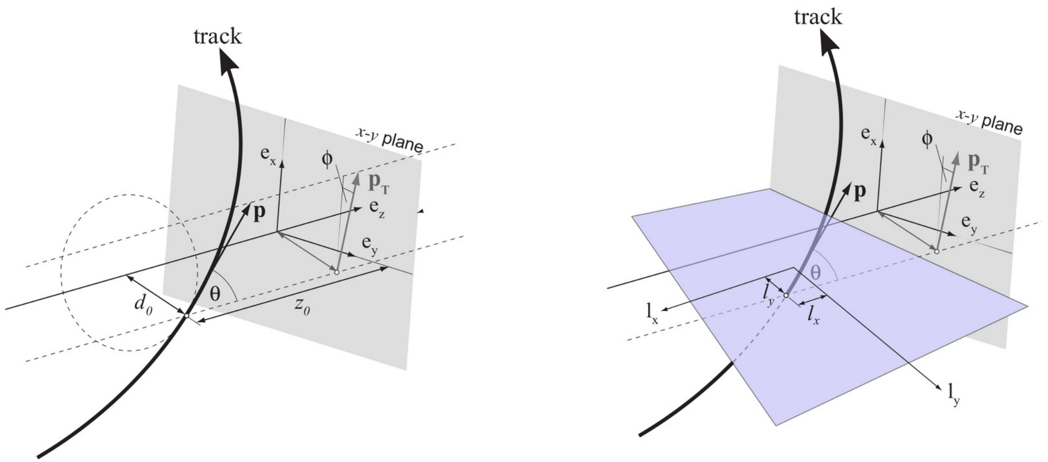
(loc0, loc1, ϕ, θ, q/p)

loc0, loc1: track position

ϕ, θ: track direction

q: charge

p: momentum



From E. Moyse

BESIII, Belle II... parameterization:

(d_0, ϕ, κ, d_z, $\tan\lambda$)

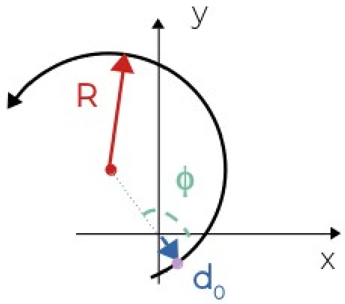
d_0 : distance between reference point to track on xy plane

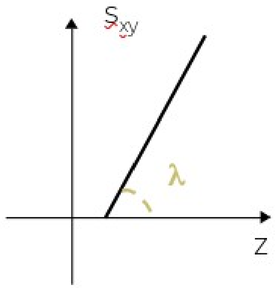
ϕ (ϕ_0) : azimuthal angle of line connecting reference point and circle center on xy plane

$\kappa(\omega)$: (signed) circle curvature

d_z : z coordinate of POA

$\tan\lambda$: ratio of path length on xy (s_{xy}) and along z





How to find & fit tracks ?

Input preparation

↓

Track finding

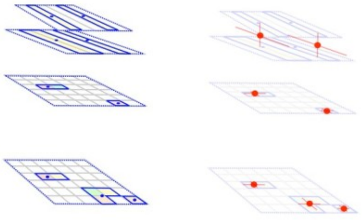
↓

Track fitting

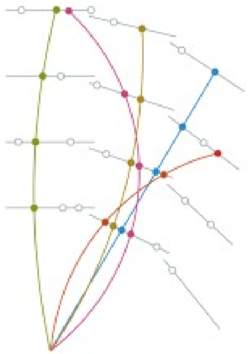
↓

Physics object reconstruction

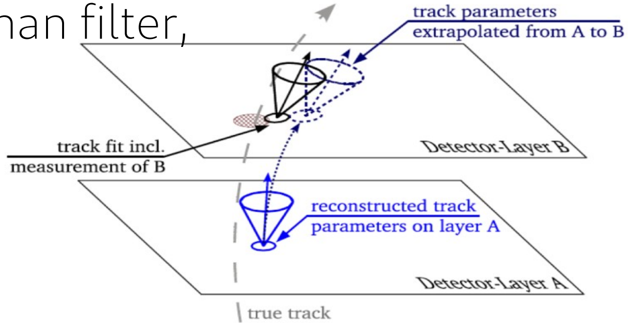
- Raw data converted to cluster/drift circle
- Formation of 3D space point



- Identify measurements to individual tracks
  - Global approach: Hough transform, Graph Neural Networks
  - Local approach: Cellular automaton, Combinatorial Kalman Filter (CKF)

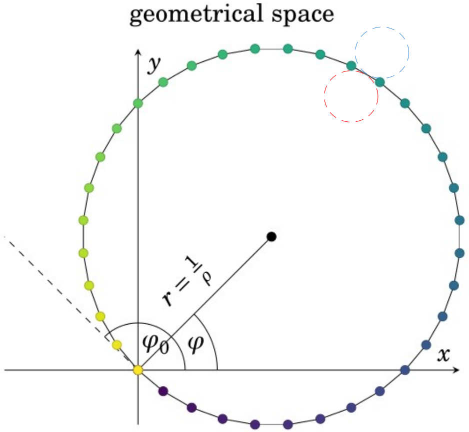


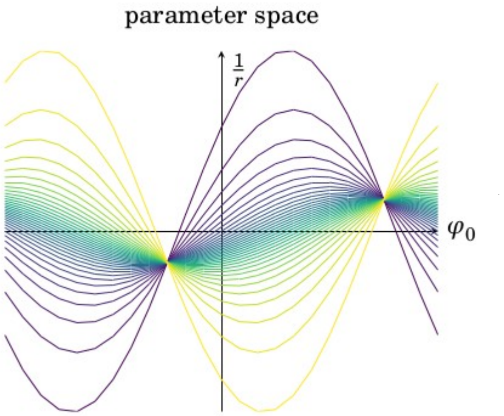
- Estimate the track parameters
  - Least-square fitter (superceded by Kalman filter, can resolve left/right ambiguity)
  - Kalman-filter



Hough Transform

- Each point (x, y) or a drift circle in geometrical space is transformed to a curve (described by two circle parameters) in parameter space
- Track finding becomes finding crossing points of curves in parameter space





e.g. a point (x,y) on the helix circle:

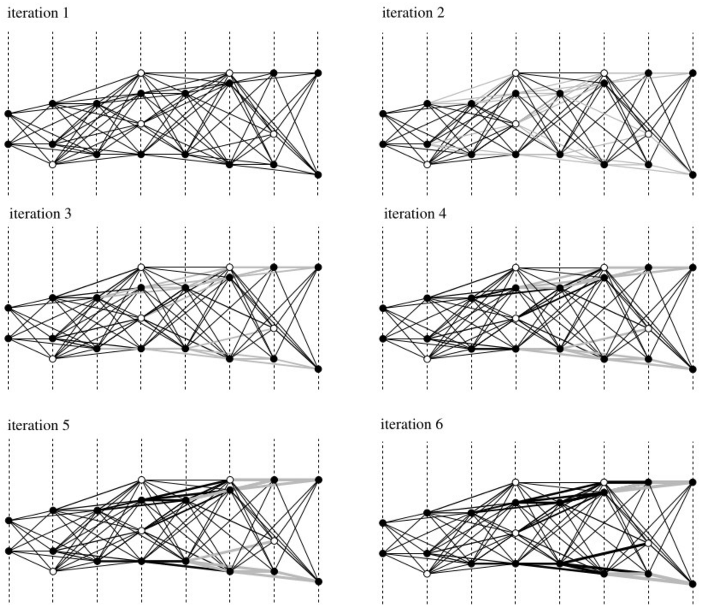
$$0 = x^2 + y^2 - 2r \cdot (x \cdot \cos\varphi + y \cdot \sin\varphi)$$

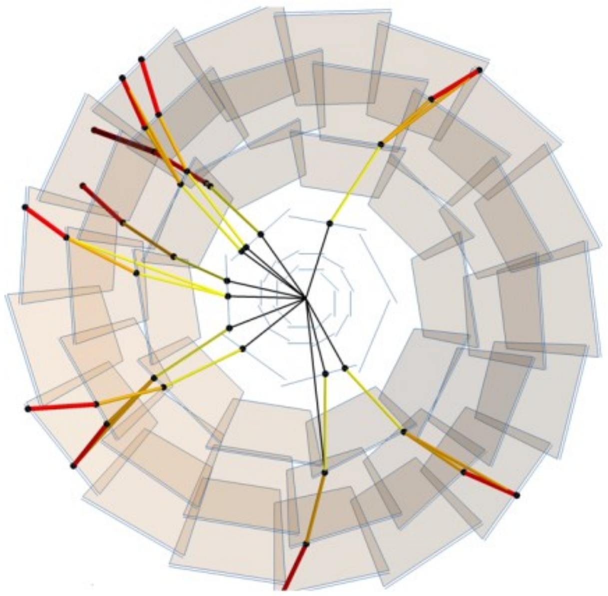
$$\frac{1}{r} = \frac{2}{r_P} sin(\varphi_0 - \varphi_P)$$

$$r_P = \sqrt{X^2 + Y^2}, \alpha_P = arctan(Y/X)$$

Cellular Automaton

- A cell is a duplet (two hits) with state (describing the “depth” of the cell in a track)
- Evolution of depth of cells following certain “game life”
  - In one evolution, simultaneous revisiting of all cells (increment of the cell depth by one if there is compatible left-sided/inner neighbor and its depth is the same as the neighbor)



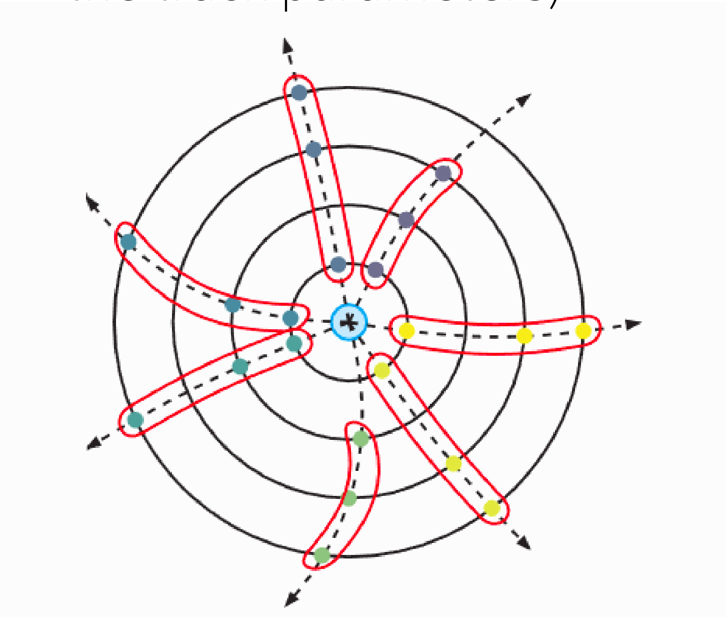


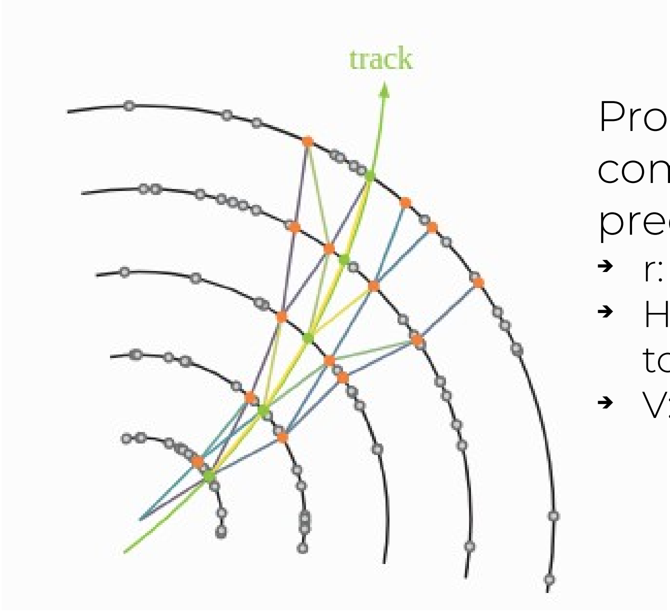
Combinatorial Kalman Filter

Seeding
(provides initial estimate of the track parameters)

→

Combinatorial Kalman Filter
(track finding through KF fitting)





Progressingly associate compatible hits to tracks based on prediction $\chi^2 = r^T (HCH^T + V)^{-1} r$.

- r: residual
- H: projection from track parameters to measurement
- V: measurement covariance

Figures from ACTS readthedocs

Towards a modern, efficient, accurate and fast common tracking software

Why a common tracking software?

- Tracking is a necessity at particle and nuclear physics experiments
- Tracking experience can be shared with different experiments
- Common software can save manpower from duplicated development and facilitate the long-term maintenance
  - e.g. great success of GEANT4, ROOT, DD4hep…!
- Understanding and optimizing old tracking software which is usually >20 years old is never easy!

A Common Tracking Software (ACTS) project

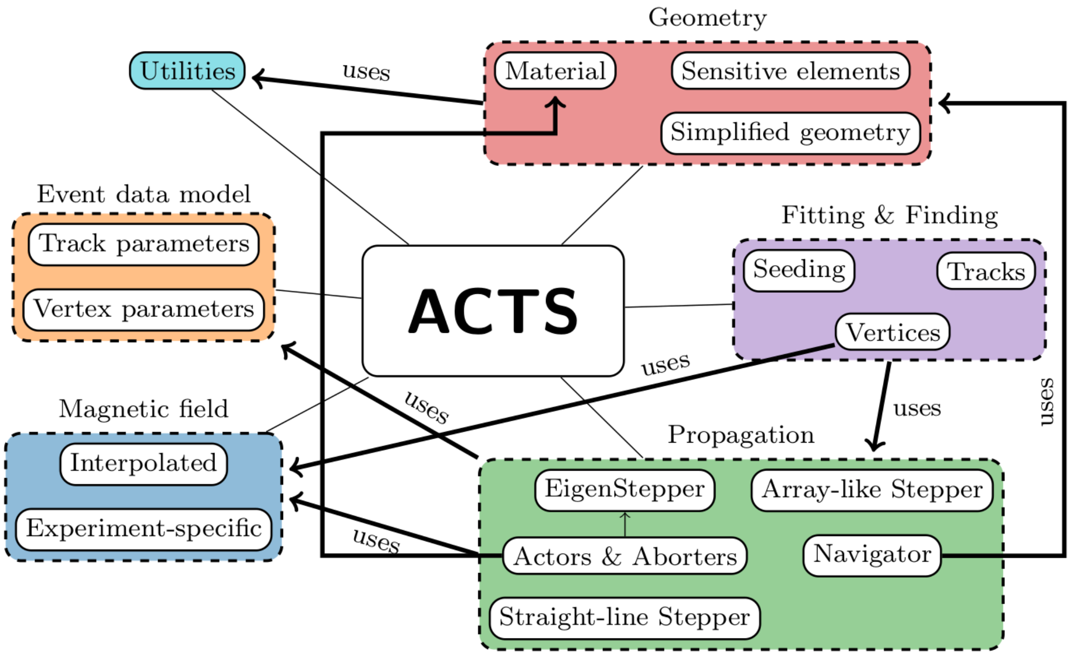
- A modern open-source **detector-independent tracking toolkit** for current&future HEP experiments (ATLAS, ALICE, sPHENIX, FASER, MUC, CEPC, STCF…) based on LHC tracking experience
- A **R&D platform** for innovative tracking techniques (ML) & computing architectures (GPU)

- **Modern C++ 17 (→ 20)** concepts
- **Detector and magnetic field agnostic**
- Strict **thread-safety** to facilitate concurrency
- Supports for **contextual** condition
- Minimal dependency (only Eigen as algebra library)
- Highly configurable for **usability**
- Well documented and maintained
- Flight time in track parameterization (facilitate time measurement):
  $\vec{x} = (l_0, l_1, \phi, \theta, q/p, t)^T$

Github:
https://github.com/acts-project/acts

Readthedocs:
https://acts.readthedocs.io/en/latest/

The core tracking&vertexing&alignment tools in ACTS



https://link.springer.com/article/10.1007/s41781-021-00078-8

- Track fitting:
  - ✔ (Extended) KF well, Gaussian Sum Filter, Non-linear KF
  - ◆ Global chisq fitter in WIP
- Track finding
  - ✔ Seeding, CKF, Graph Neural Networks
  - ◆ Hough Transform in WIP
- Vertex finding&fitting
  - ✔ Primary vertex: AMVF, IVF
- KF-based Alignment in WIP

The third-party plugins

- Autodiff
- Cuda
- DD4hep
- EDM4hep
- ExaTrkX
- Geant4
- Identification
- Json
- Legacy
- Mlpack
- Onnx
- Sycl
- TGeo

- Supports experiment applications, R&D on ML and GPU...
  - Geometry: Geant4, DD4hep, TGeo
  - GPU: Cuda, Sycl
  - ML: Onnx, Mlpack, ExaTrkX
  - Configration: Json
  - EDM: EDM4hep
  - Math tool: Autodiff

World-wide contribution

- 10~15 active developers on Core project

Contributors 54

+ 43 contributors

supported by

EP-R&D Programme on Technologies for Future Experiments

iris hep

Bundesministerium für Bildung und Forschung

cooperations

HSF HEP Software Foundation

intel

CERN openlab

ACTS is one of the four projects in IRIS-HEP (Institute for Research and Innovation in Software for High Energy Physics)

➔ $25M, i.e. ~0.17 B CNY, funded by National Science Foundation

https://iris-hep.org

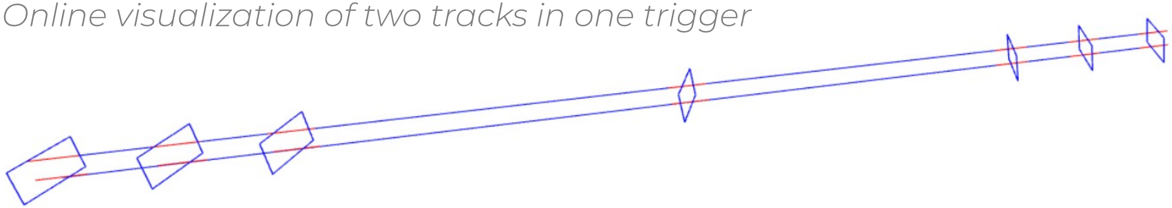
World-wide applications

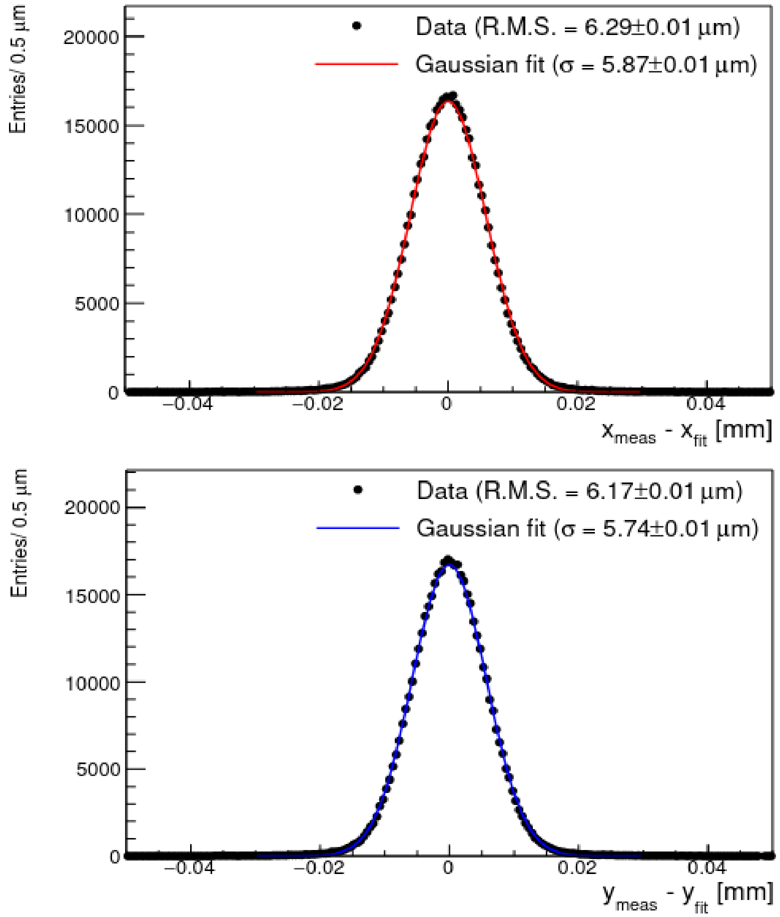
- World-wide users from particle and nuclear physics, collider and non-collider experiments
  - >10 experiments
  - >15 institutes : CERN, LBNL, ORNL, UC Berkeley, Stanford University, DESY, ZZU, …
  - 119 Forks
- Regular/irregular discussion between developers and experiment users
  - ATLAS, FASER, sPHENIX, ALICE, EIC…

ACTS application example: ADENIUM beam telescope

- Beam telescope is a key instrumental tool for particle detector prototyping
- Combined tracking fitting and finding with CKF much ease the tracking process
- Good time performance allows **online track reconstruction and visualization**
  - Event processing rate up to 20 kHz in a single thread!
- First application of ACTS for real data processing!

Online visualization of two tracks in one trigger



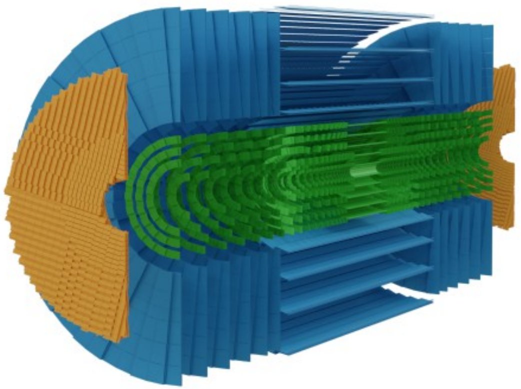


ADENIUM beam telescope

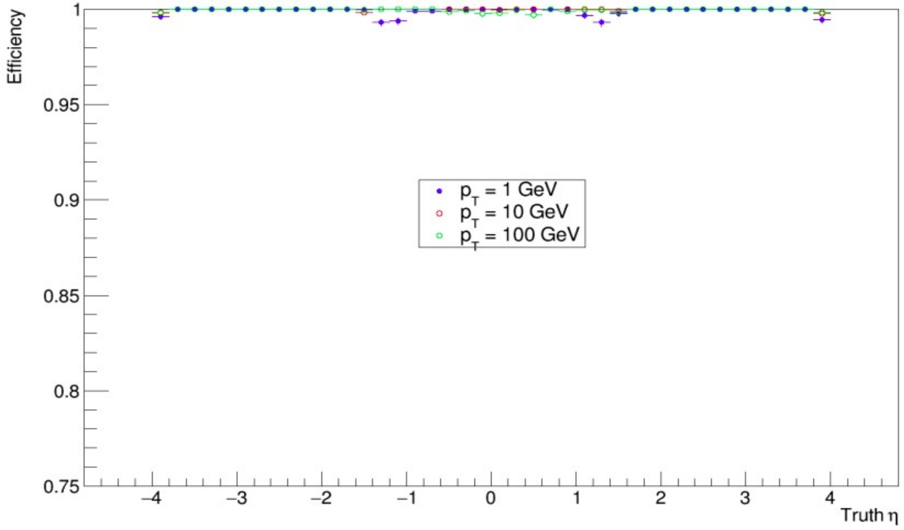
Y. Liu et al. arXiv: 1907.10600

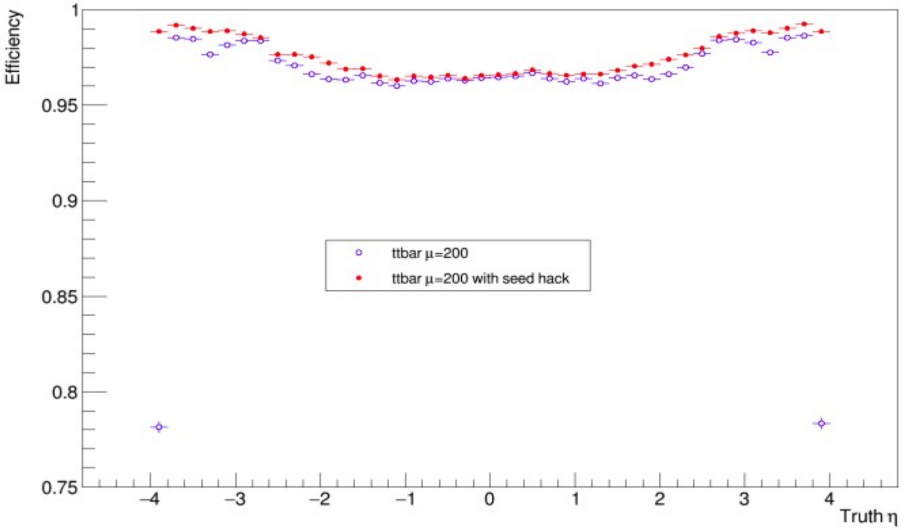
Application for ATLAS ITk

ATLAS ITk (HGTD)



- For example, >95% track finding efficiency for ttbar with μ = 200 for ATLAS ITk

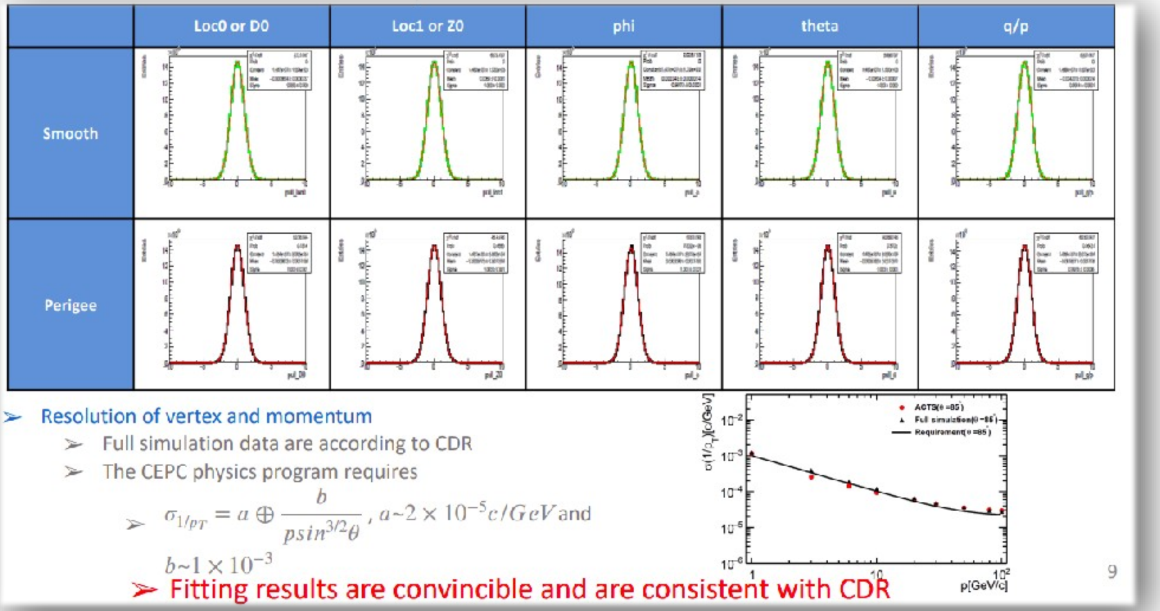


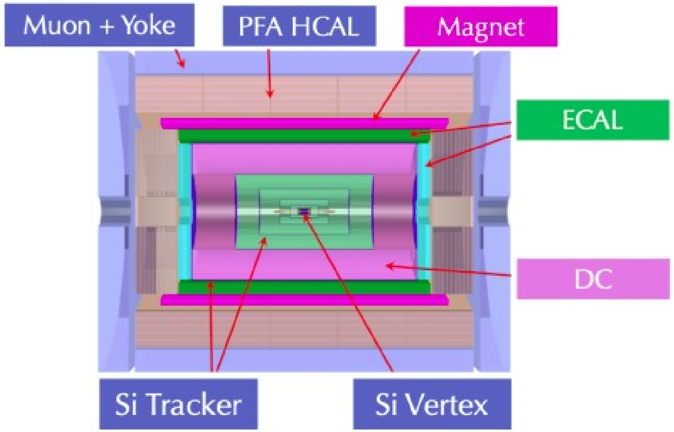


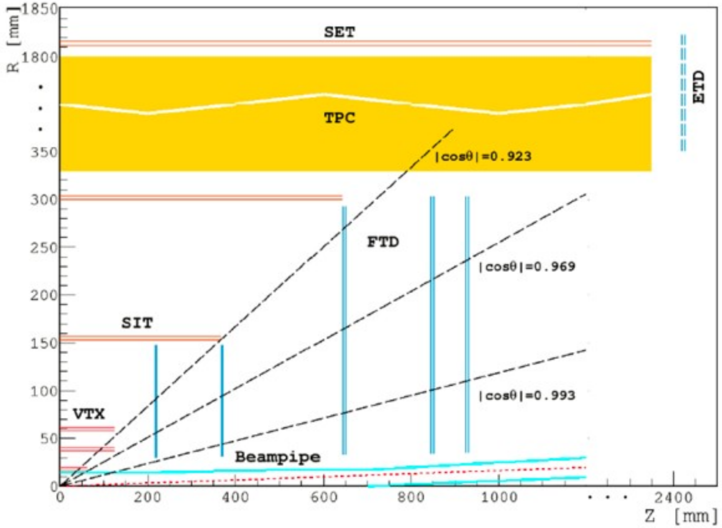
Application for CEPC

- Truth fitting using ACTS KF shows compatible performance with CDR

Figure from J. Zhang's slides at CEPC workshop in 2021

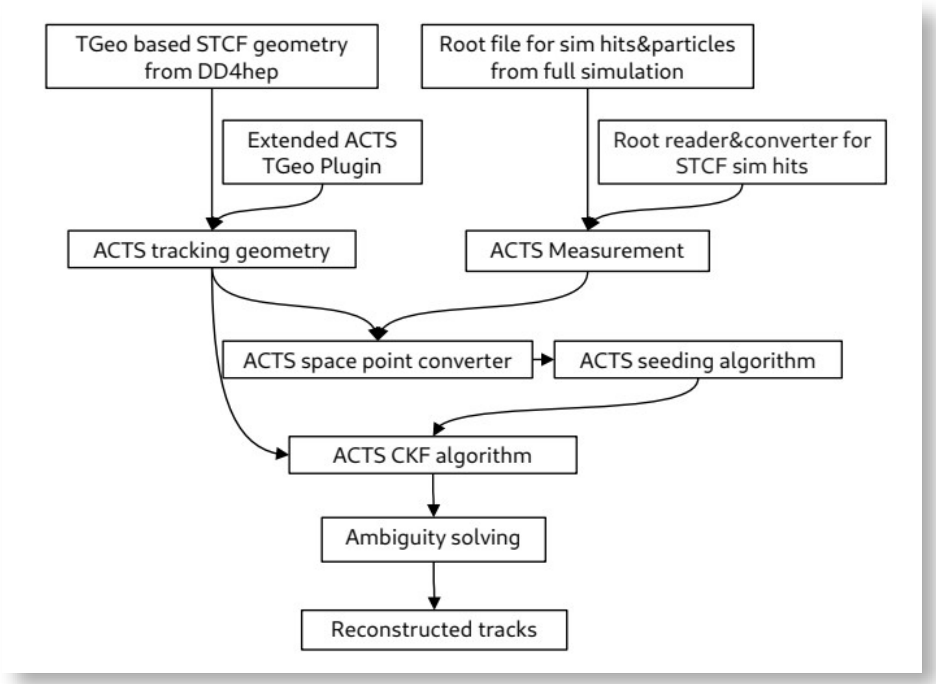


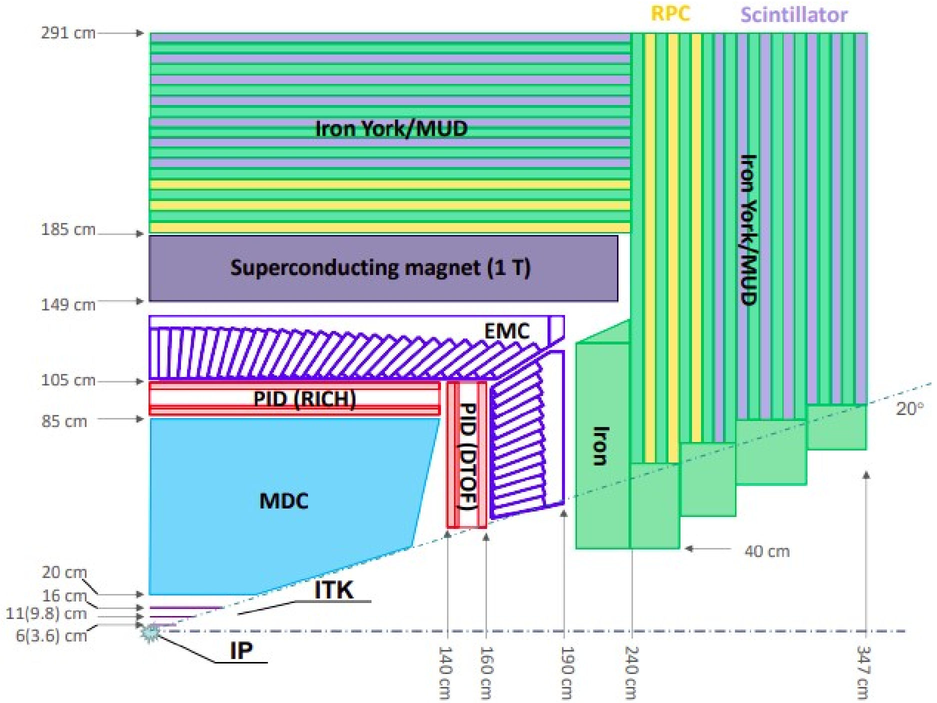




Application for STCF

- First application of ACTS CKF for a drift chamber!





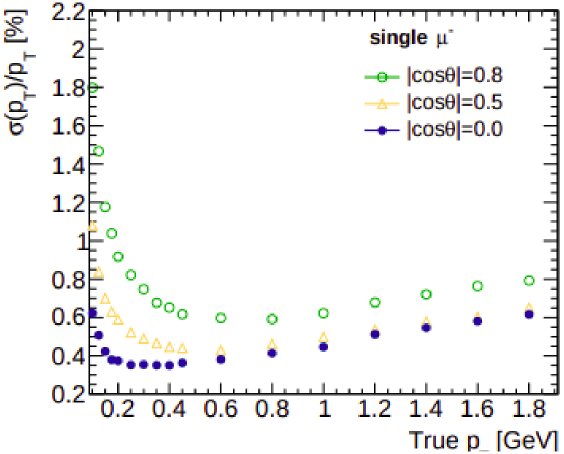
X. Ai, X. Huang, Y. Liu, arXiv:2301.04306

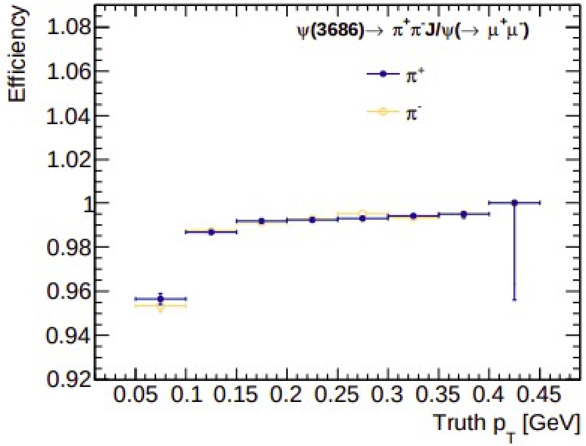
ITK: 3 layers, $\sigma_{r\text{-}\phi} \times \sigma_z \approx$ 100 um x 400 um

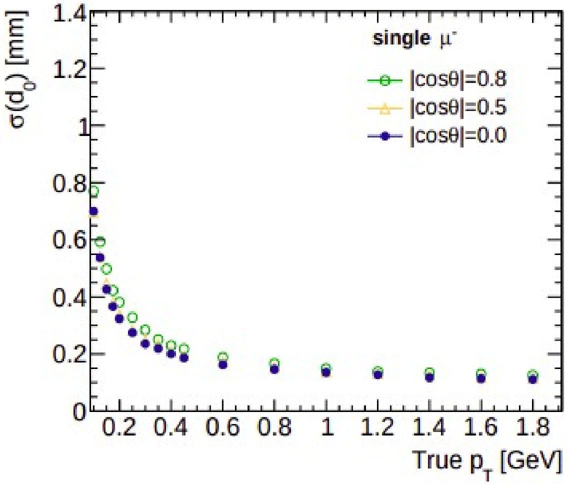
MDC: 48 layers, $\sigma_{\text{drift dist}} \approx$ 120~130 um

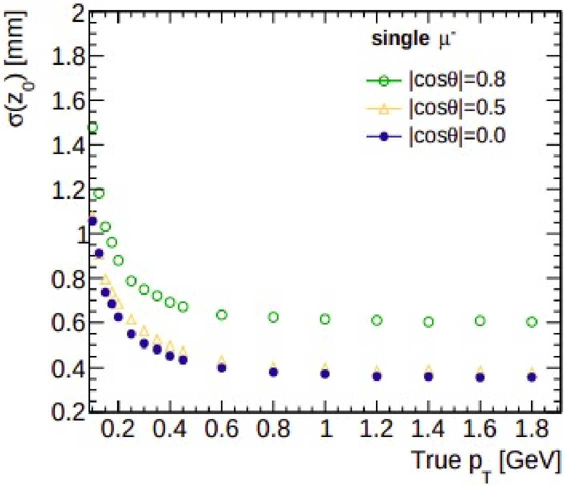
Tracking performance for STCF

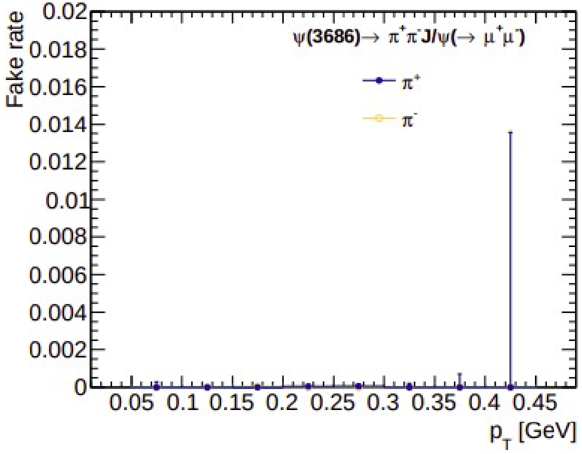
- For p_T = 1 GeV, θ = 90 deg:
  - $\sigma(p_T)/p_T < 0.5\%$, $\sigma(d_0) \sim 150$ μm, $\sigma(z_0) \sim 400$ μm
- >95% track eff. for p_T in [50, 100] MeV in $\pi^+\pi^- J/\psi$











X. Ai, X. Huang, Y. Liu, arXiv:2301.04306

Summary&Outlook

Summary & Outlook

- HEP softwares are about development, performance and physics!
- Event reconstruction is very complicated and will become much more challenging in the future
- Tracking is pivotal to event reconstruction in HEP&NP
- ACTS is an international project to develop an open source and highly performant tracking software
  - Becomes very popular in recent years and interest keeps growing
  - Used by >10 particle and nuclear physics programs
  - Still a lot remain to be developed and optimized

The last remarks

China should and need to play a role in development, optimization and maintenance of common HEP softwares!

What's more pleasant than using tools developed by us hence well known to us?

Q&A

The deployed tracking strategies

Experiments	Track finding algorithms	Track fitting algorithms
ATLAS	CKF	Global chisq fitter
CMS	CKF	KF
ALICE	Cellular automaton	KF
LHCb	Track segment finder	KF
BESIII	Track segment finder, Hough Transform	KF
Belle II	CKF, Cellular automaton, Hough Transform	KF