Tracking, Vertexing and b-tagging at the LHC

M. Musich

Università di Pisa & INFN

on behalf of the ALICE, ATLAS, CMS, LHCb collaborations

VERTEX 2023

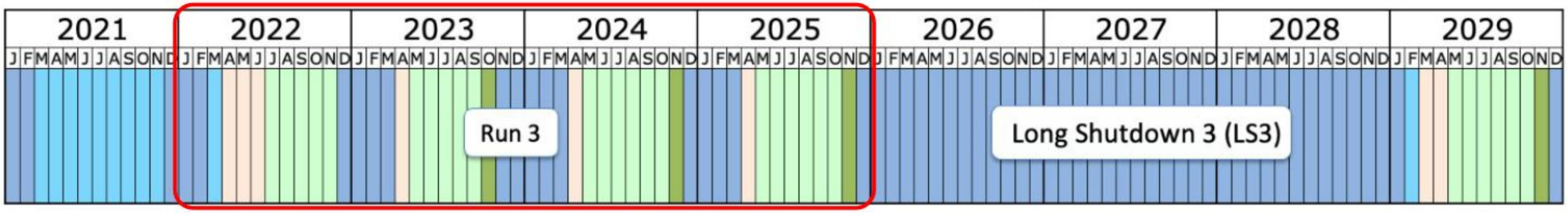
VERTEX 2023 - 32nd International Workshop on Vertex Detectors

Sestri Levante (IT) 16-20th October 2023

Introduction

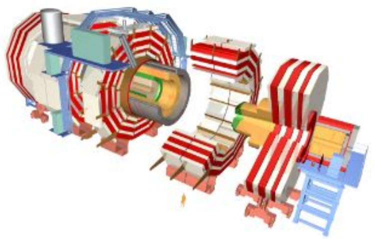
- Tracking and Vertexing at the LHC:
    - The tracking challenge at the LHC
- Common basic concepts of Tracking
- CMS Silicon Tracker and Tracking:
    - Online & Offline performance
- ATLAS ID and Tracking:
    - Run 3 optimization and performance
- LHCb upgrades during LS2:
    - HLT1 with `Allen` and tracking in HLT2
- ALICE - upgrades in LS2:
    - Mid-y tracking in Run 3, 4 and performance
- Few words on flavour tagging:
    - ATLAS & CMS results
- Conclusions & Outlook

Tracking & Vertexing at LHC



- Tracking and vertexing: are key ingredients to reconstruct collisions at the LHC;
- Reconstruction needs to be efficient, precise, pure and quick;
- Complex combinatorial problem in high pile-up and/or high interaction rates scenarios as in Run 3 at the LHC;

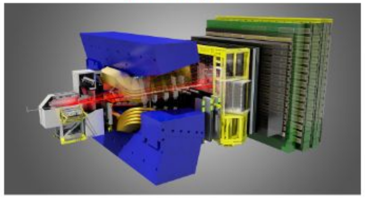
Compact Muon Solenoid (CMS)



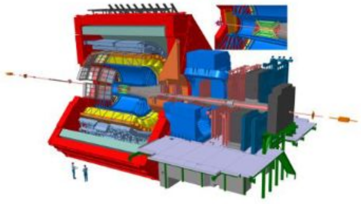
A Toroidal LHC Apparatus (ATLAS)

Reconstruction algorithms renewed to handle higher average in-bunch pile-up collisions (⟨μ⟩).

Large Hadron Collider beauty (LHCb)



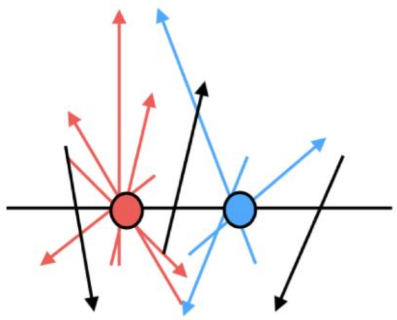
A Large Ion Collider Experiment (ALICE)

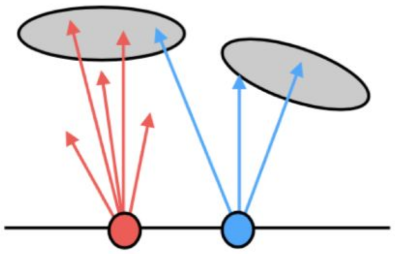


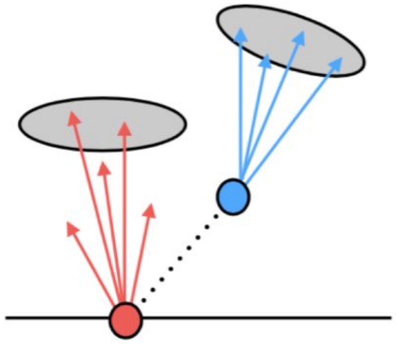
Major detector upgrades and renewed data flow to significantly increase the collected statistics in Run 3, 4

Tracking challenge at LHC

- The tracking challenge at the LHC:
    - typically 30 charged particles within the tracking volume acceptance per proton-proton collision
    - and 50-60 collisions per event: O(1500) charged particles per event;
- These need to be reconstructed:
    - with **very high efficiency** (>90% for ~GeV pions)
    - **precise** track parameters
    - very **low fake rate**: O(~ few %)
    - **quickly** (stringent CPU limits)
- Very strong requirements on track reconstruction algorithms
- Track reconstruction is not just about reconstructing charged particles:
    - used in almost every element of reconstruction


vertex reconstruction

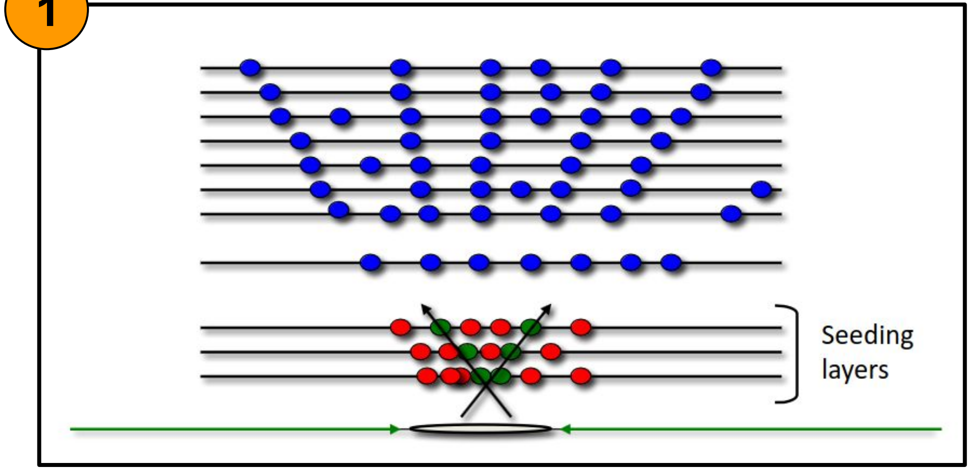

pile up removal


jet flavour tagging

Common basic concepts

- Tracking can be summarized in 4 main steps
  1. **Seeding**: build “short tracks” to be used as seeds for longer tracks;

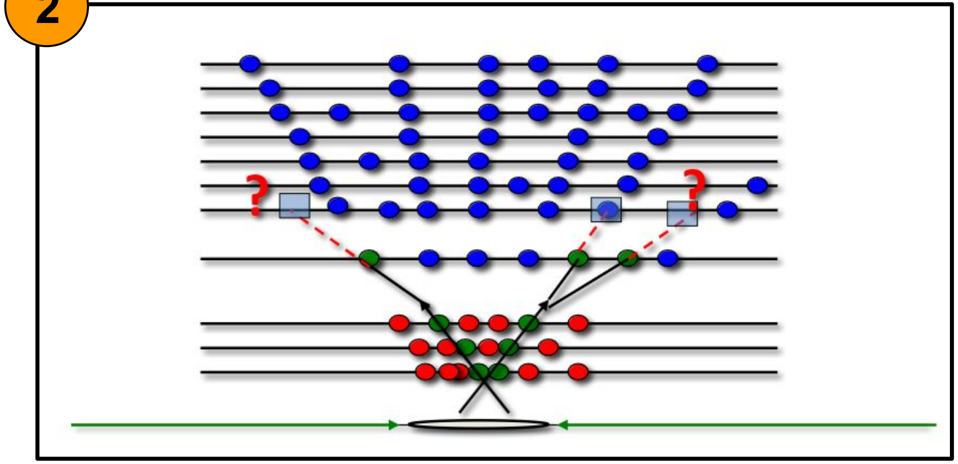
1



Common basic concepts

- Tracking can be summarized in 4 main steps
  1. **Seeding**: build “short tracks” to be used as seeds for longer tracks;
  2. **Track finding / pattern recognition**: search for additional hits to prolong track seeds to other tracking layers;

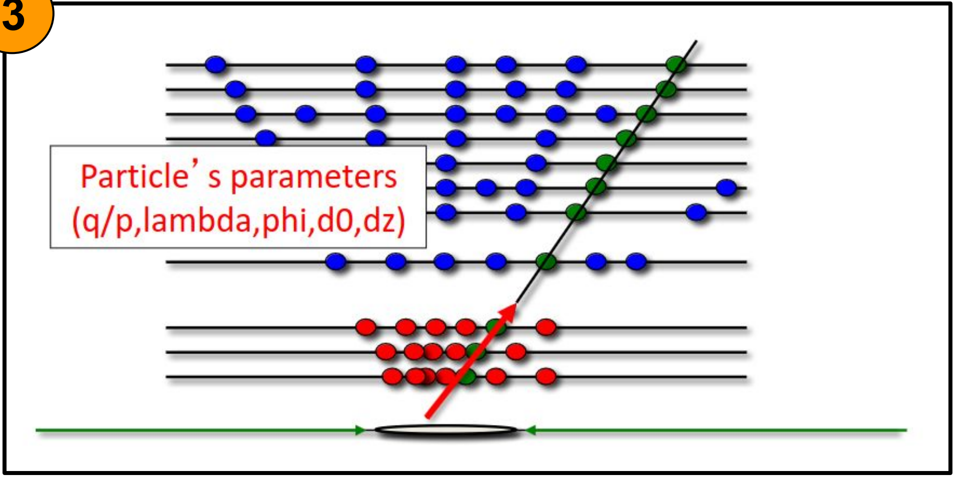
2



Common basic concepts

- Tracking can be summarized in 4 main steps
  - ① **Seeding**: build “short tracks” to be used as seeds for longer tracks;
  - ② **Track finding / pattern recognition**: search for additional hits to prolong track seeds to other tracking layers;
  - ③ **Track fitting**: use the points found during the track finding to calculate the track parameters and covariance matrix;

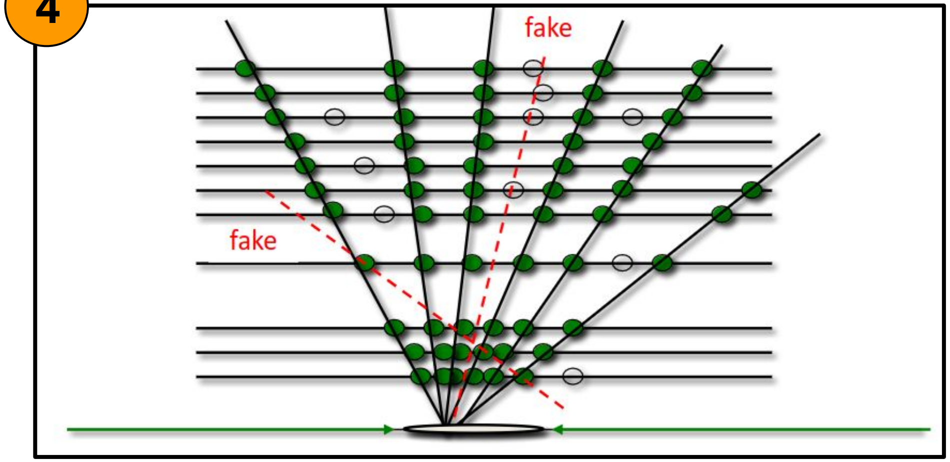
③



Common basic concepts

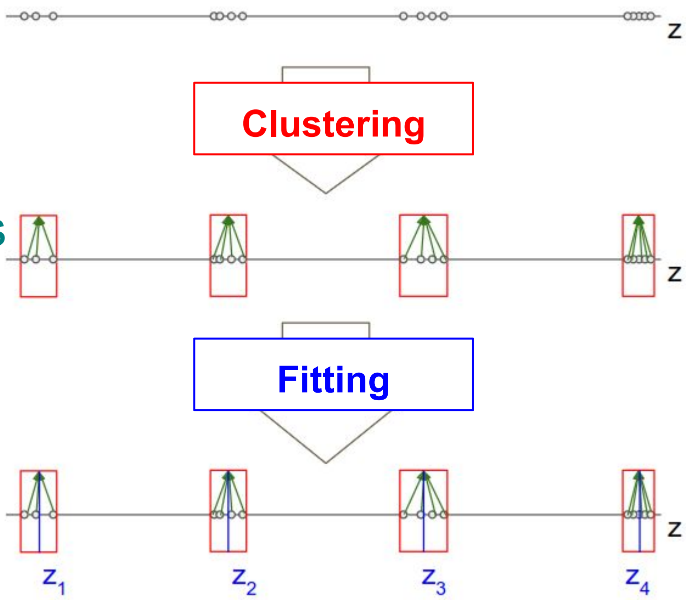
- Tracking can be summarized in 4 main steps
  1. **Seeding**: build “short tracks” to be used as seeds for longer tracks;
  2. **Track finding / pattern recognition**: search for additional hits to prolong track seeds to other tracking layers;
  3. **Track fitting**: use the points found during the track finding to calculate the track parameters and covariance matrix;
  4. **Track selection**: apply quality criteria to reduce the fraction of bad-quality and fake track.

4



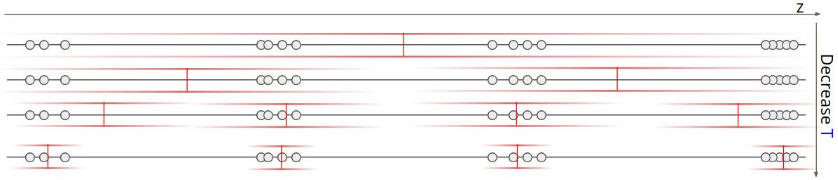
Vetexing at the LHC

- Vertexing starts from a set of tracks.
- Then proceeds into two steps:
  - ○ **Clustering**: group together close-by tracks in cluster candidates. The algorithm used is deterministic annealing;
  - ○ **Fitting**: fit vertex properties of those clusters from those of the tracks. The algorithm used is Adaptive vertex fitting algorithm;

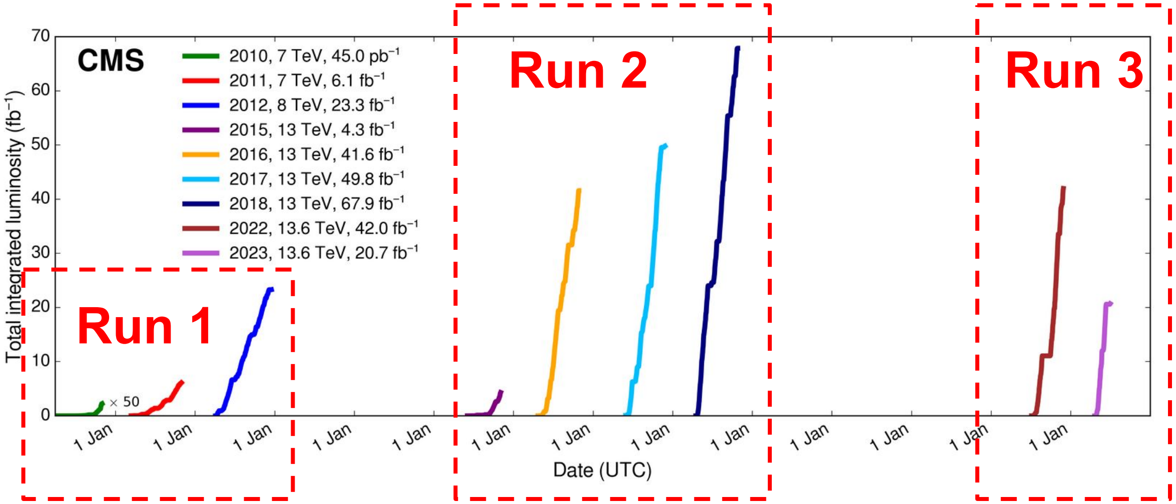


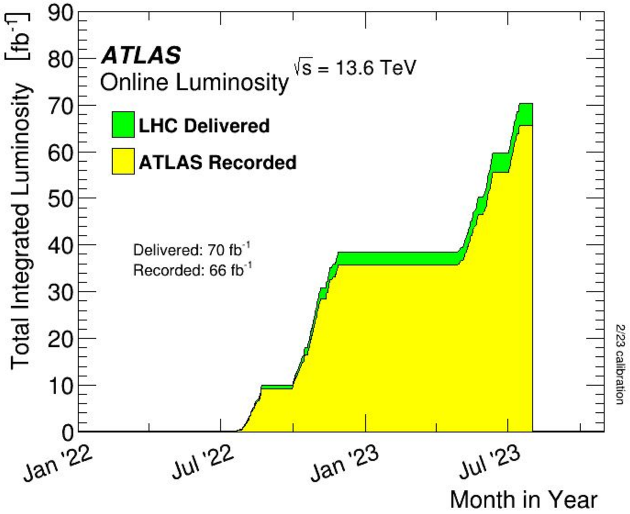
- The Deterministic Annealing (DA) for clustering is quite common at the LHC and is based on optimizing an energy (assignment) function with a penalization entropy term:
  - ○ Starting at very high temperature (T) all tracks are assigned to one single cluster;
  - ○ As we lower T, splitting the cluster into several becomes beneficial;
  - ○ Iteratively update assignment probabilities P_{ik} while lowering T provides a final robust estimation of the clusters.

$$E - TS = \sum_{i=1}^{I}\sum_{k=1}^{K} P_{ik}\frac{(z_i - z_k)^2}{\sigma_i^2} + T\sum_{i=1}^{I}\sum_{k=1}^{K} P_{ik}\log(P_{ik})$$

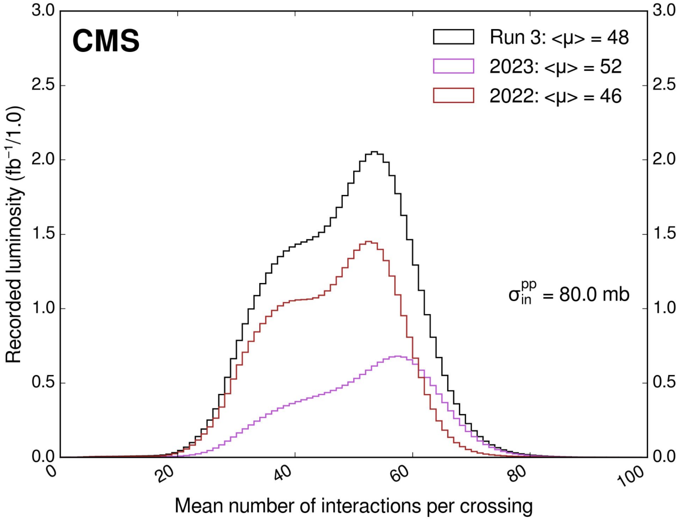


Run-3: Data Taking so far





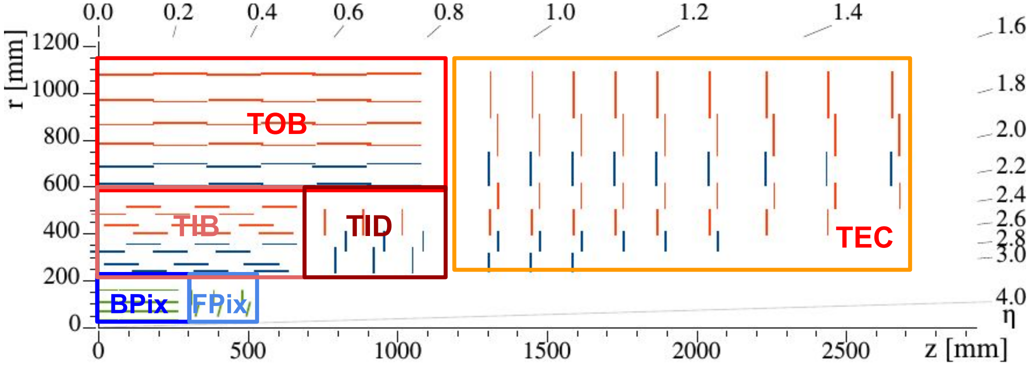
- Luminosity delivered to CMS/ATLAS by the end of Run 2 is >190 fb^{-1}.
- Luminosity delivered in Run 3 as of today during Run 3 is ~70fb^{-1}.
  - LHC is expected to deliver around 250fb^{-1}
- Average number of pp interactions per crossing in Run 3 is 48, 52 considering only 2023:
  - Highly irradiated environment, challenging conditions for the tracking detectors.



CMS Tracker & Tracking

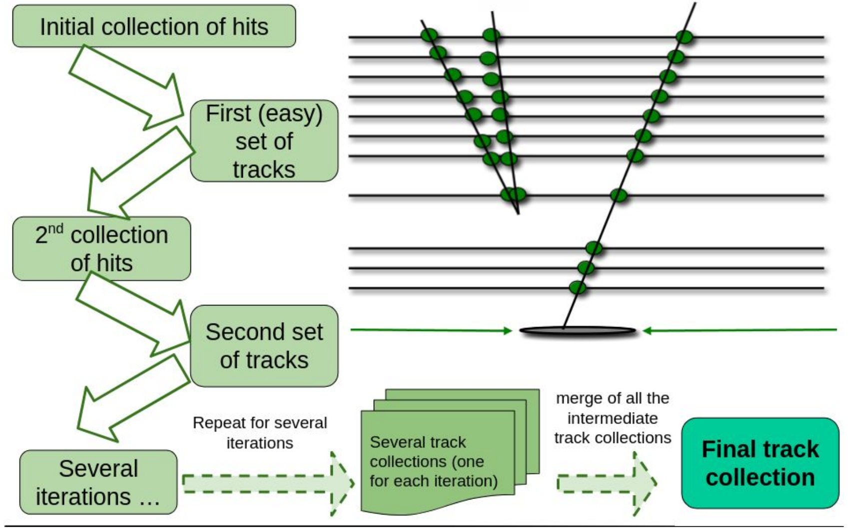
- **Seeding**: 3D points from pixels and/or at least two mono-stereo layers in the Silicon Strip Tracker
- **Track finding** / pattern recognition:
  - Outward KF + further inward search of further hits;
  - cleaner/filter (in each iteration) using shared hits and quality requirements;
- **Track fitting**:
  - Outward KF initialized at the innermost hit.
  - Smoother: second filter initialized to the result of the first one;
  - Final track parameters: weighted average;
  - Iteratively repeat the above to reject outlier hits;
- **Track selection**: quality selections to reduce fake tracks
  - DNN-based since Run 3 ([CMS DP-2023/009](CMS DP-2023/009))

Hermetic tracking system within |η|< 3



- Combinatorial Kalman Filter (CKF): pattern recognition + track fitting:
- Iterative tracking →different track categories in each iteration

[2014 *JINST* **9** P10009](2014 JINST 9 P10009)


Initial collection of hits

First (easy) set of tracks

2^{nd} collection of hits

Second set of tracks

Repeat for several iterations

Several iterations …

Several track collections (one for each iteration)

merge of all the intermediate track collections

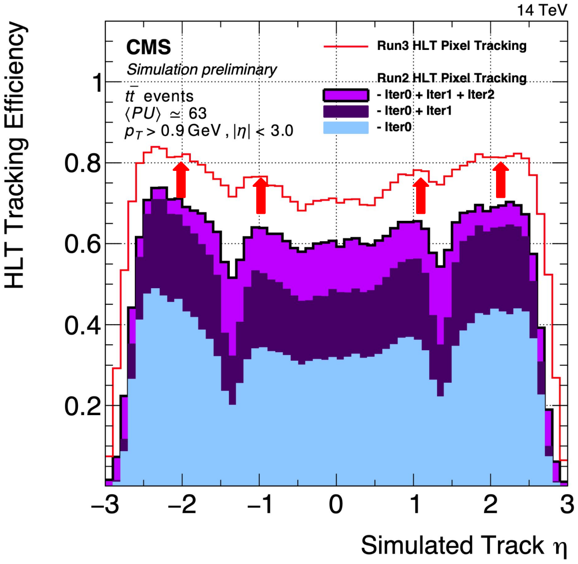
Final track collection

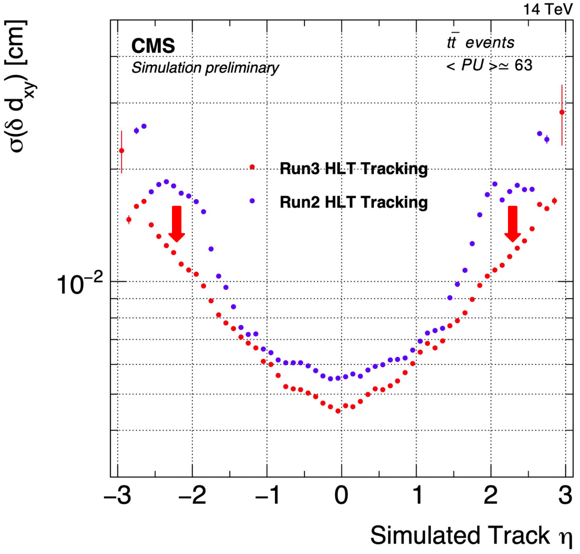
CMS Online Tracking in Run3

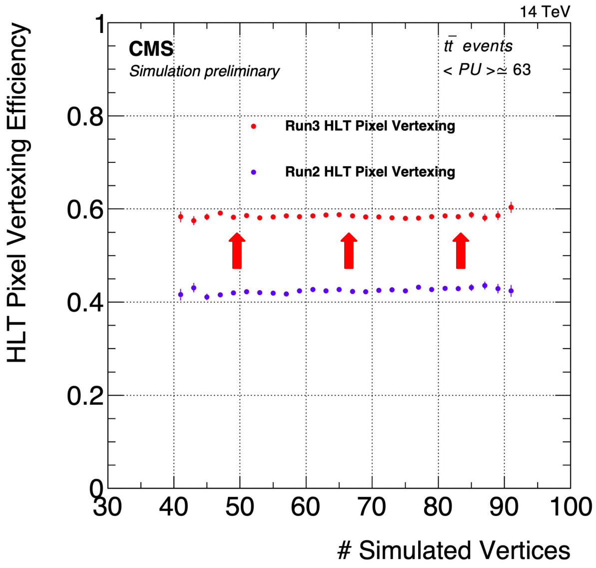
- High-Level Trigger (HLT): streamlined version of the offline reconstruction software on a farm for large reduction in data rate;

- HLT track seeding and vertexing based on pixel detector only
    - HLT pixel tracking ported to GPUs → heterogeneous computing with CUDA (“**Patatrack**” Front.Big Data 3 (2020), 601728)
- Better physics performance and throughput;
    - With respect to the Run 2 HLT tracking, better fake rate rejection and improved impact parameters resolutions.

CERN-CMS-DP-2022-014

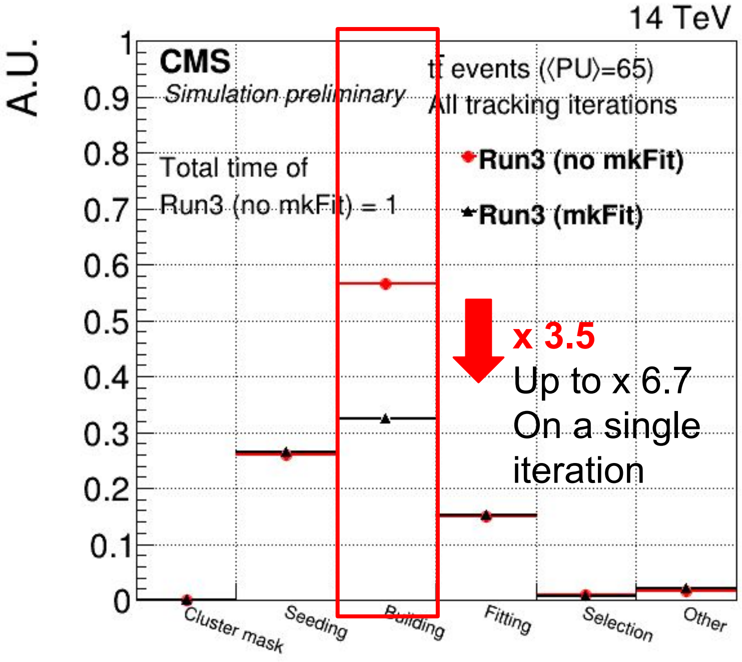




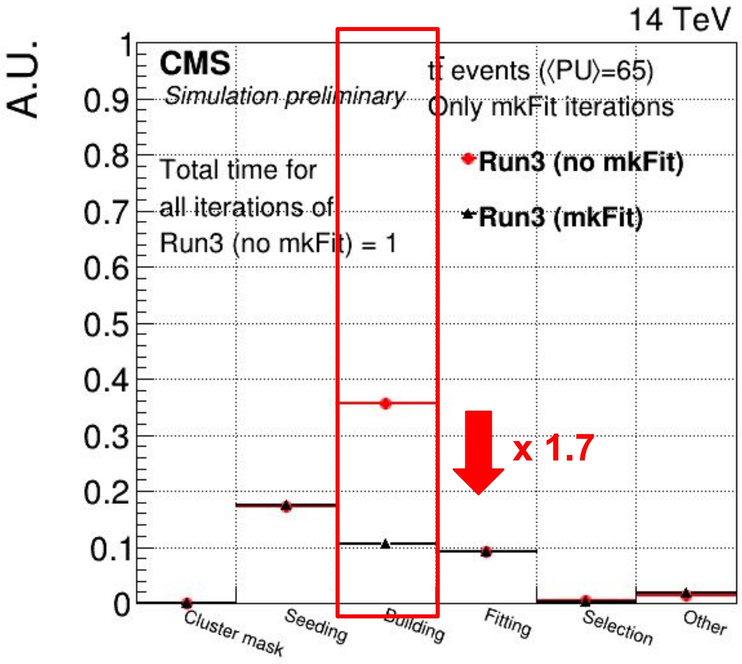


CMS Offline Tracking in Run3

- Pattern recognition optimization in Run 3:
  - MATRIPLEX Kalman-filter algo (**mkFit,** S. Lantz *et al* 2020 *JINST* **15** P09030)
- Parallelized and vectorized CKF:
  - MATRIPLEX: custom library to optimize memory access to track covariance matrices in CKF;
  - Similar physics performance as Run 2 CKF;
  - Significant speed up (also through simplified tracker geometry);
  - Used by a subset of tracking iterations reconstructing ~90% hard-scattering event



Only iterations improved by mkFit



All tracking iterations

CMS-DP-2022-018

CMS: speeding up vertexing

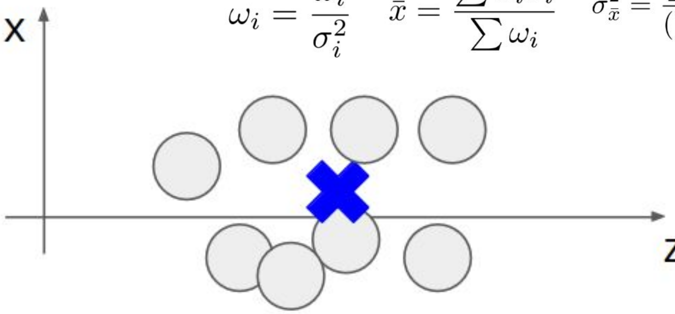
- In the Phase-2 environment both steps (clustering and fitting) in vertexing will involve computations across ~1000s of tracks and ~100s of vertices.
- The legacy algorithms scale baldy.
  - Proposal to redesign them in order to fit better in a heterogeneous computing environment.
- The new clustering procedure sorts the tracks in the z coordinate, splits them in blocks of same size (set by default to 512) with a fixed overlap fraction between blocks (set by default to 0.5) and performs independently the DA along all the blocks.

CMS-DP-2022-052

- The new estimator iteratively estimates the vertex 3D coordinates and errors using the weighted mean of tracks impact point at the beamspot position and uncertainty. The iterations include an outlier rejection to improve the performance.

$$t^2_{i,x} = \frac{(x_i - \overline{x})^2}{(\sigma^2_{i,x} + \sigma^2_{\bar{x}})} \quad \text{If } \forall j \ \ t_{i,j} < 3 \rightarrow \tilde{\omega}_i = 1 \quad \text{Else } \tilde{\omega}_i = 0$$

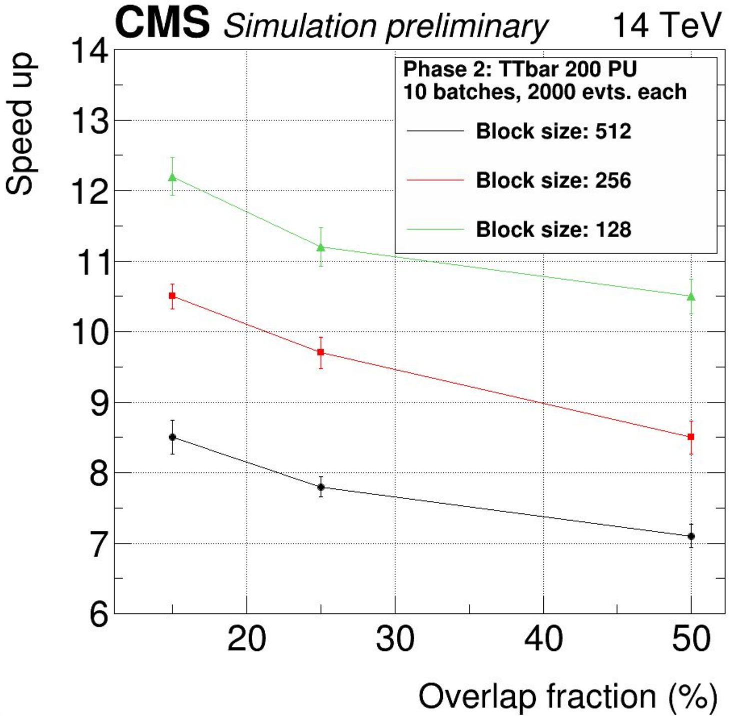
$$\omega_i = \frac{\tilde{\omega}_i}{\sigma_i^2} \qquad \bar{x} = \frac{\sum \omega_i x_i}{\sum \omega_i} \qquad \sigma^2_{\bar{x}} = \frac{\sum \omega_i \tilde{\omega}_i}{(\sum \omega_i)^2}$$



CMS: speeding up vertexing

- Performance increases already in the CPU due to the decrease in the complexity of the algorithm as we dramatically decrease the number of track-vertex association needed:

$$p_{ij}^{(n)} = \frac{e^{-\beta\left(\frac{z_i - z_k^{(n-1)}}{\sigma_i^2}\right)^2}}{\sum_{l=1}^{K} e^{-\beta\left(\frac{z_i - z_l^{(n-1)}}{\sigma_i^2}\right)^2}}$$



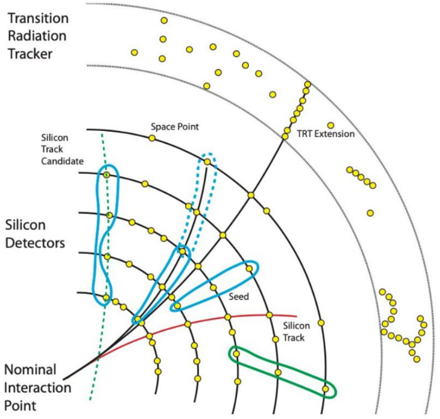
CMS-DP-2022-052

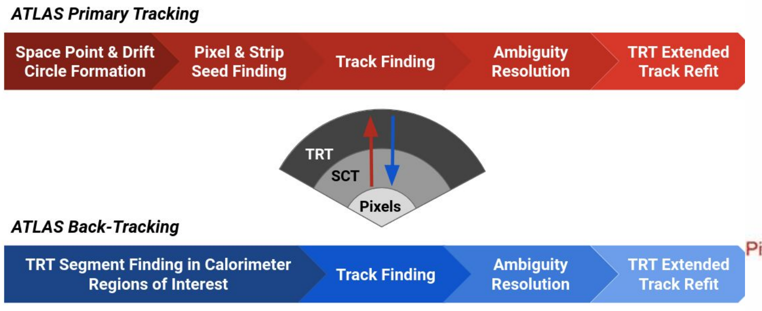
- Single block:
  - ~200 vertices x ~10000 tracks => 2×10^6 P_{ii} values.

- 512 track block:
  - split the 10000 tracks in 40 overlapping blocks. 40 blocks x 10 vertices x 500 tracks => 2×10^5 P_{ij} values;
    - Effectively we are transforming the problem of clustering at <PU> ~ 200 into 40 overlapping problems of clustering at <PU> ~ 10.

ATLAS: ID & tracking

- Inner Detector (ID) tracker (|η| < 2.5)
    - Pixel Tracker;
    - Silicon SemiConductor Tracker (SCT);
    - Transition Radiation Tracker (TRT);



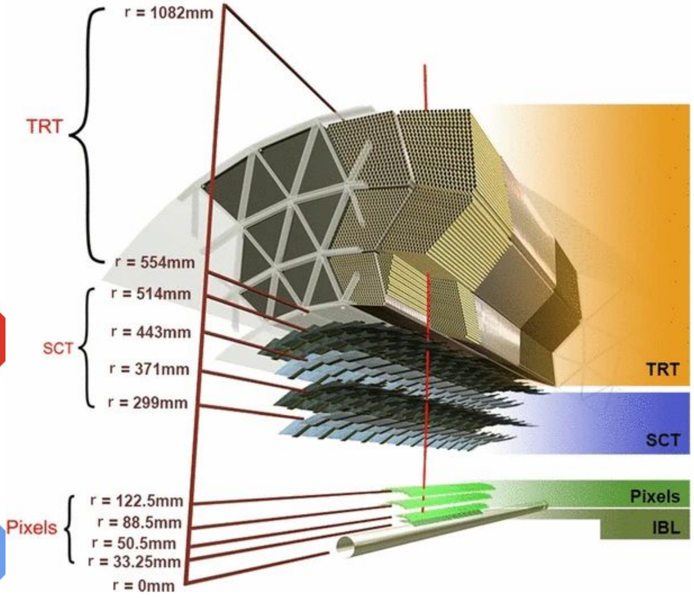


ATLAS Primary Tracking

Space Point & Drift Circle Formation → Pixel & Strip Seed Finding → Track Finding → Ambiguity Resolution → TRT Extended Track Refit

ATLAS Back-Tracking

TRT Segment Finding in Calorimeter Regions of Interest → Track Finding → Ambiguity Resolution → TRT Extended Track Refit



- **Primary tracking (INSIDE-out)** → primaries
    - Seeding: triplets in pixel + SCT
    - Track finding: CKF to extend tracks outwards up to SCT outer layers;
    - Track ambiguity solver:
        - track scoring based on hit topology (holes, shared hits) and quality (χ^2 , …)
        - neural network (NN) to minimize inefficiency due to merged clusters
    - Global fitting + extension to TRT (+ re-fit)

- **Back-tracking (OUTSIDE-in)** → secondaries, (γ-conversions w/o silicon hits)
    - Seeding and pattern recognition starting from TRT
    - Inward tracking → include silicon segments missed by primary tracking
    - Hits assigned to tracks by INSIDE-out not considered

📚 ATL-PHYS-PUB-2021-012

📚 Eur. Phys. J. C (2017) 77:673

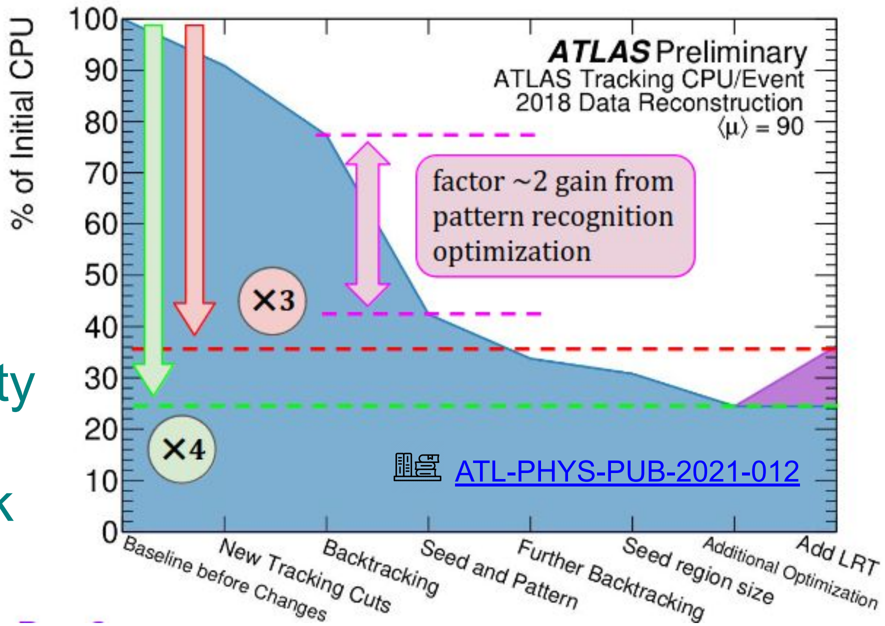
ATLAS: Run 3 optimization

- The challenge:
  - Run 2: μ = 20-40→ Run 3: μ ~ 50
  - lower the resource consumption, while retaining unvaried track quality;
- Several improvements for Run 3:
  - Tighter selections for the ambiguity solver;
  - More stringent conditions for track seeding and track finding;
  - New primary vertex (PV) reconstruction algorithm: Adaptive multi-vertex fitter (**AMVF**).

ATL-PHYS-PUB-2019-015

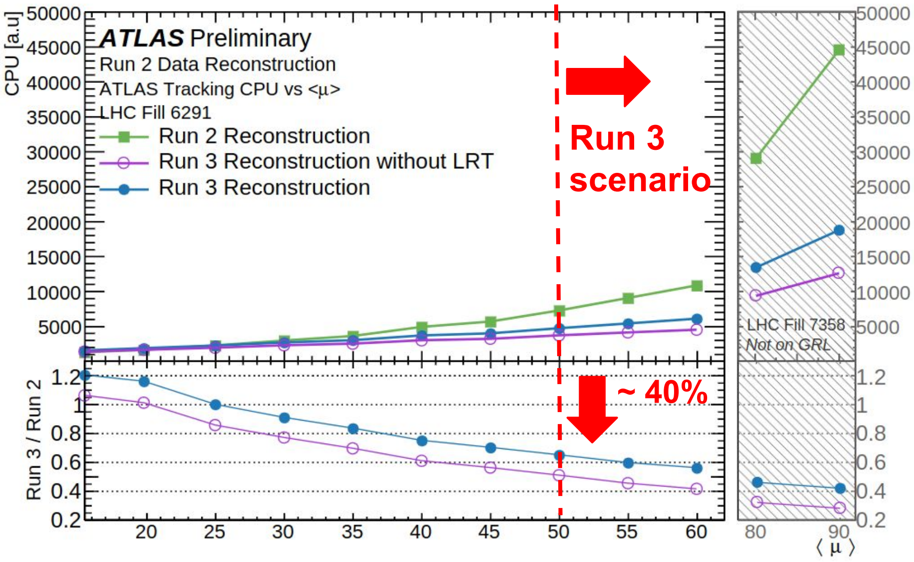
Eur. Phys. J. C (2017) 77:332

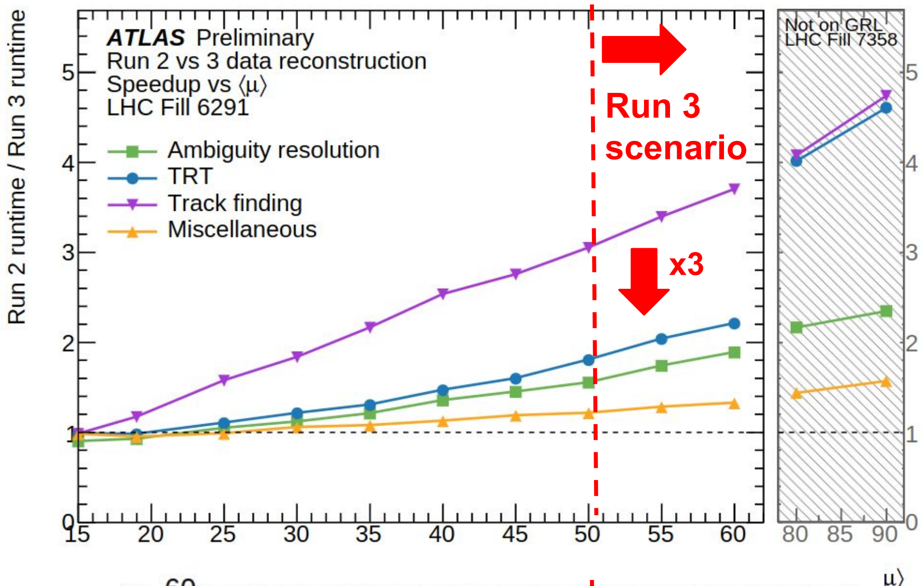
  - Reduced fractions of low-quality and fake tracks.
  - Improved PV reconstruction efficiency.



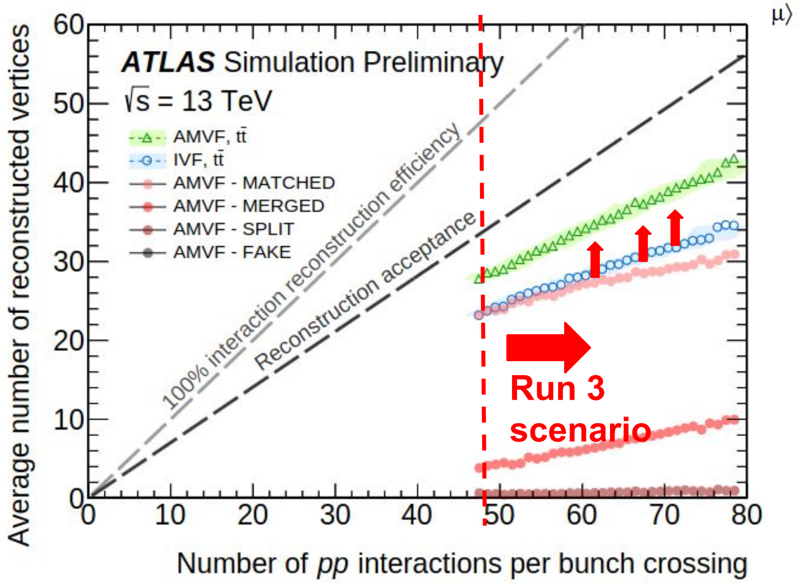
- Reduction of single-thread CPU timing for tracking per bunch-crossing;
- In Run 3: Large Radius Tracking (LRT): further reconstruction pass to recover non-pointing tracks from displaced decays (strangeness)

ATLAS: Run 3 performance





- Near linear scaling vs.⟨μ⟩with Run 3 reconstruction chain
  - **⟨μ⟩~ 50:** CPU usage lower of ~40% than Run 2
  - **⟨μ⟩~ 50:** pattern recognition runtime ~3 times lower (1.5-2 others)
- AMVF recovers up to 35% of the reconstructable primary vertices at high value of⟨μ⟩, lost by the Run 2 algorithm (Iterative Vertex Finding).

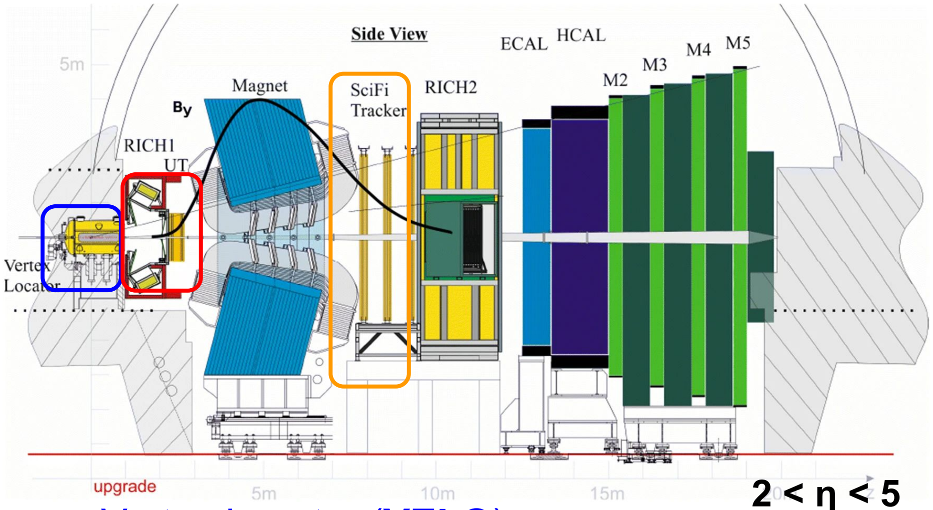


ATL-PHYS-PUB-2019-015 ATL-PHYS-PUB-2021-012

LHCb: upgrades in LS2

- **Challenges for Run 3**:
  - bunch-crossing rate up to 40 MHz
  - pile-up: $\langle\mu\rangle \sim 1.4 \rightarrow \langle\mu\rangle \sim 5$

Detector upgrades (tracking only!)



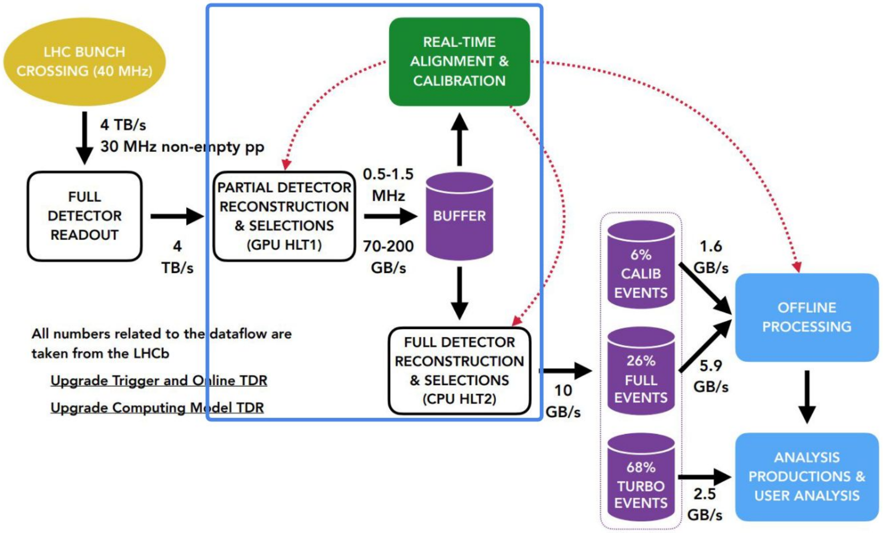
2 < η < 5

- Vertex Locator (**VELO**)
  - [old] Si strips → [new] 26 Si-pixel layers
- Upstream Tracker (**UT**)
  - 4 layers of high-granularity Si micro-strips
- Scintillating Fiber Tracker (**SciFi**) + Si photo-multipliers (SiPMs)
  - 3 stations × 4 SciFi layers

[2022 JINST 17 C01046] [LHCb TDR 015]

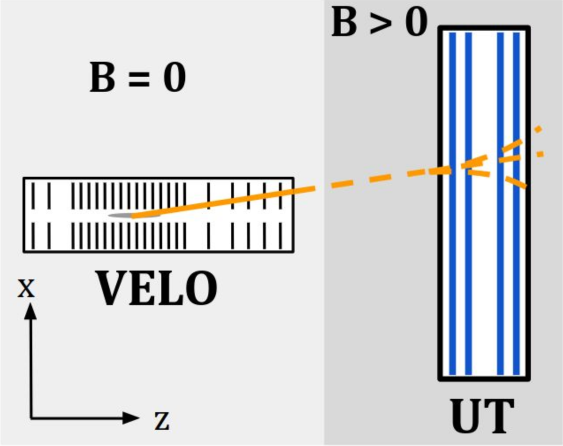
Renewed data flow

[Comput Softw Big Sci 4, 7 (2020)]



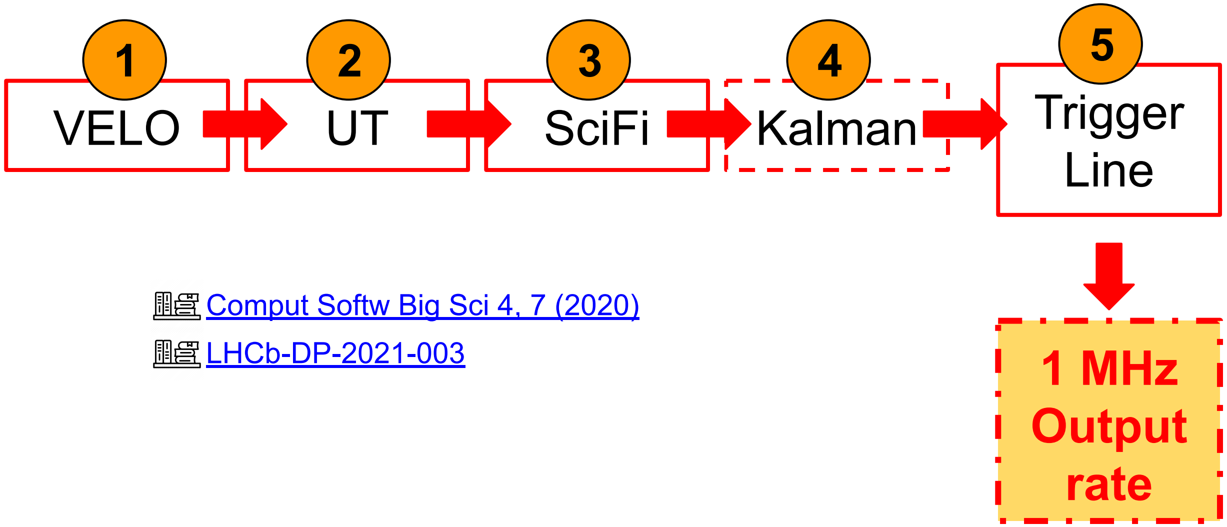
- Level-0 hardware trigger (~1 MHz) → software trigger to be (~30 MHz non-empty pp collisions)
- GPU High-Level Trigger 1 (**HLT1**)
  - Real-time alignment and calibrations
- CPU High-Level Trigger 2 (**HLT2**)

LHCb - HLT1 with Allen



Allen: A High-Level Trigger on GPU's for LHCb:

- Cheaper and more scalable than CPU alternative;
- Chosen as baseline of the upgrade;
- Implemented with O(200) Nvidia RTX A5000 GPUs;



Comput Softw Big Sci 4, 7 (2020)

LHCb-DP-2021-003

1
- Seeds from three hits on consecutive layers (triples)
- Extension to other layers with linear KF

2
- Extrapolation of VELO tracks to UT
- Momentum estimate from bending

3
- VELO+UT tracks extrapolation to SciFi tracker

4
- KF to improve dca resolution
- VELO-only KF in HLT1 (speedup)

5
- Parallel fitting of 2-track secondary vertices (SV)
- Trigger selections (tracks and/or SV)

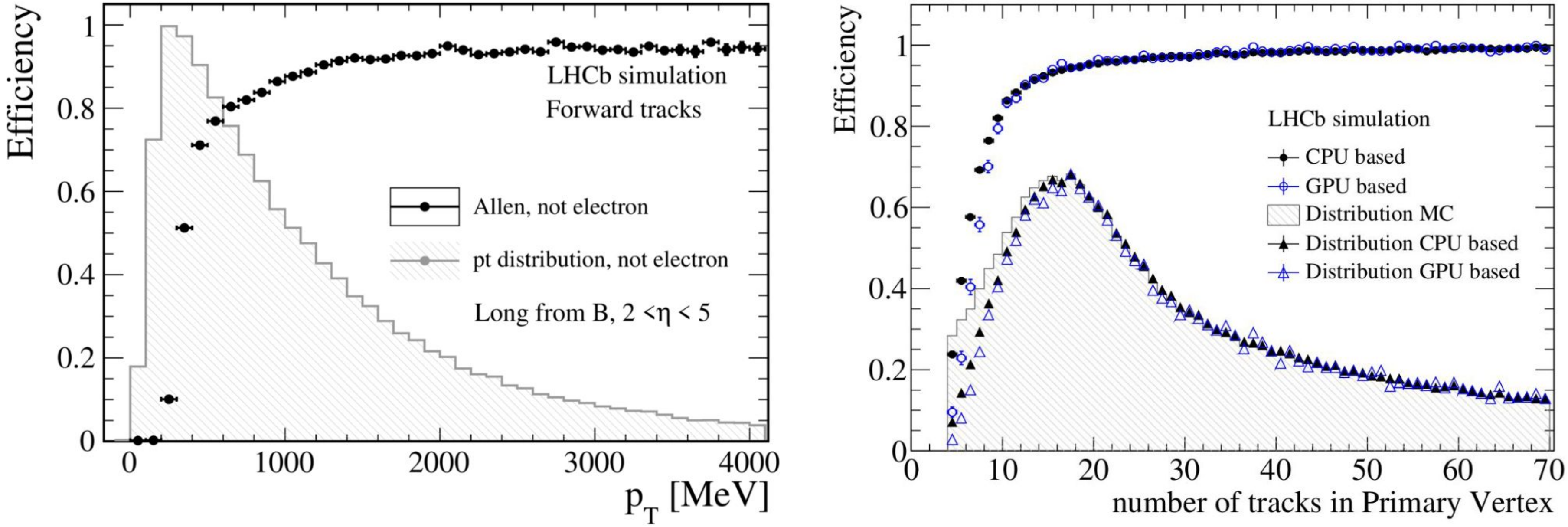
LHCb - HLT1 performance

📚 Comput Softw Big Sci 4, 7 (2020)

📚 LHCb-DP-2021-003

Allen: A High-Level Trigger on GPU's for LHCb:

- Cheaper and more scalable than CPU alternative;
- Chosen as baseline of the upgrade;
- Implemented with O(200) Nvidia RTX A5000 GPUs;

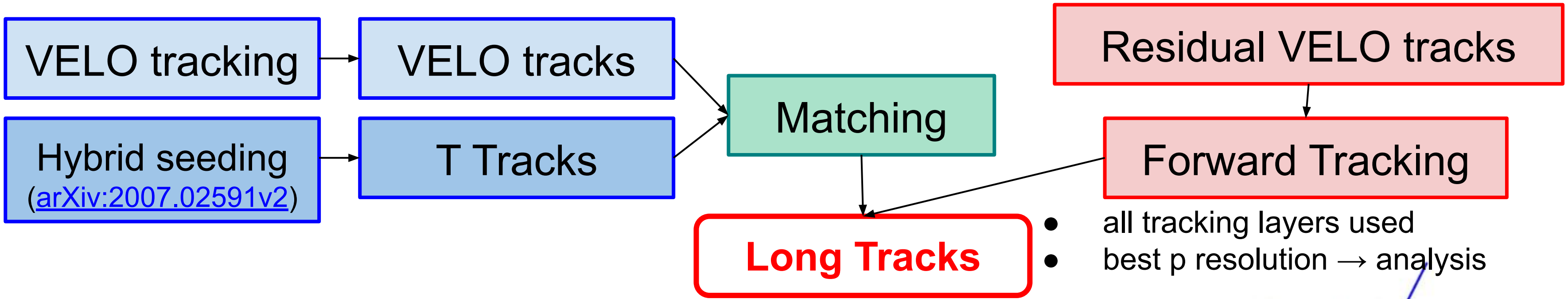


- **Tracking efficiency > 90%** for p_T >1 GeV/c
- **PV efficiency > 90% (95%)** for VELO tracks > 10 (20)

LHCb - tracking in HLT2

Matching: neural network trained on MC to match VELO and T- tracks

Forward tracking: VELO+UT-track extension to SciFi (# hits ≥ 10)



VELO tracking

VELO tracks

Hybrid seeding (arXiv:2007.02591v2)

T Tracks

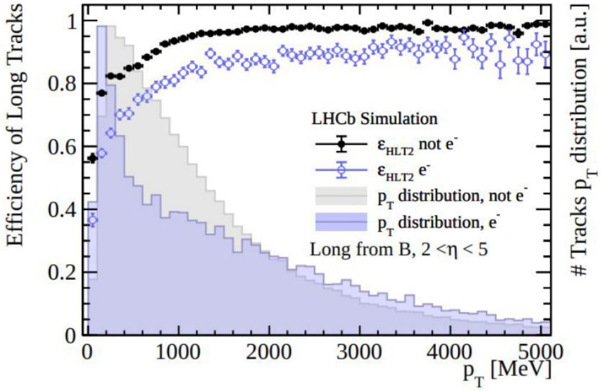
Matching

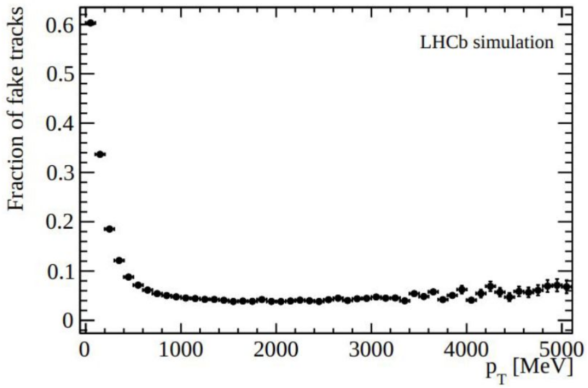
Residual VELO tracks

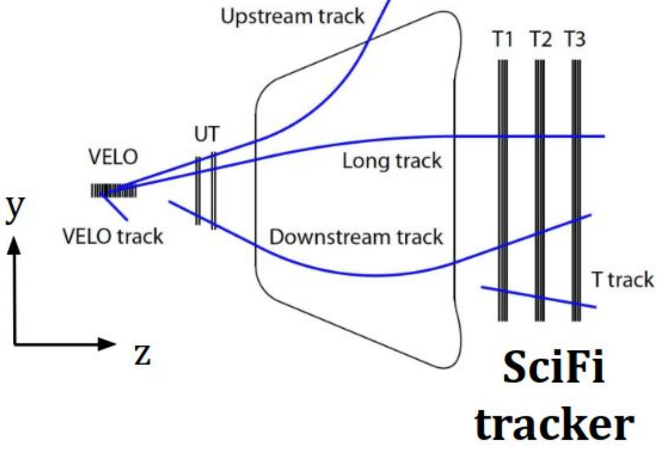
Forward Tracking

Long Tracks

- all tracking layers used
- best p resolution → analysis







- **Tracking efficiency** for hadrons and μ←B ~ 90% (**> 95%** for **p_T > 1 GeV/c**)
- Fraction of **fake-tracks ~ 6%** for **p_T > 1 GeV/c**
- Larger at low p_T (multiple scattering)

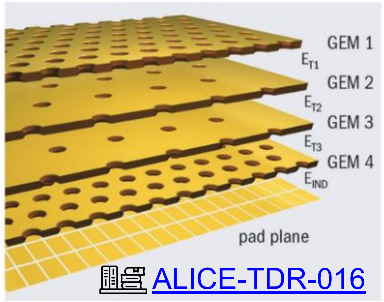
LHCb-FIGURE-2022-005

LHCb-PROC-2022-009

ALICE - upgrades in LS2

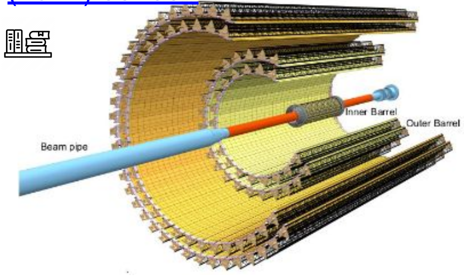
Challenges For Run 3:
Interaction rate up to 1MHz (pp, $\sqrt{s}$ = 13.6 TeV)
Interaction rate ~ 50 kHz (Pb-Pb, $\sqrt{s_{NN}}$ = 5.44 TeV)

Detector Upgrades



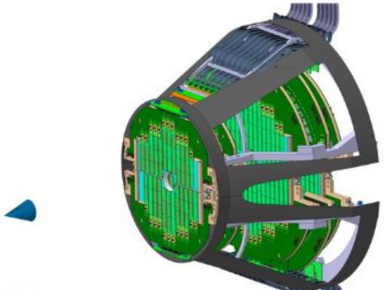
ALICE-TDR-016

J. Phys. G: Nucl. Part. Phys. 41 (2014) 087002

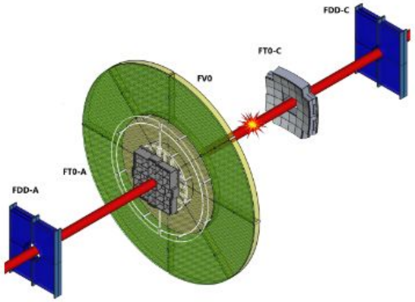


Time **P**rojection **C**hamber (**TPC**) upgrade → |η| < 0.9

Inner **T**racking **S**ystem (**ITS**) upgrade → |η| < 1.3



ALICE-TDR-018

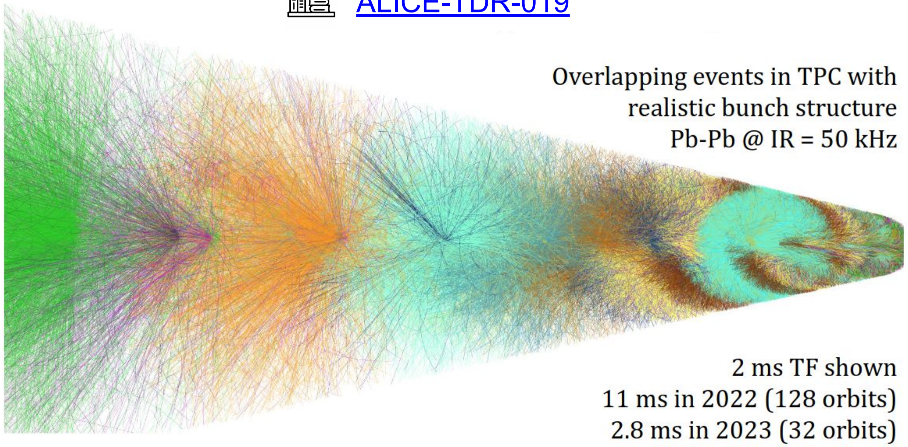


Muon **F**orward **T**racker (**MFT**) → 2.5 < η < 3.6

Fast **I**nteraction **T**rigger (**FIT**) → 2.2 < η < 6.3, -6.9 < η < -2.3

Renewed data processing

ALICE-TDR-019



Overlapping events in TPC with
realistic bunch structure
Pb-Pb @ IR = 50 kHz

2 ms TF shown
11 ms in 2022 (128 orbits)
2.8 ms in 2023 (32 orbits)

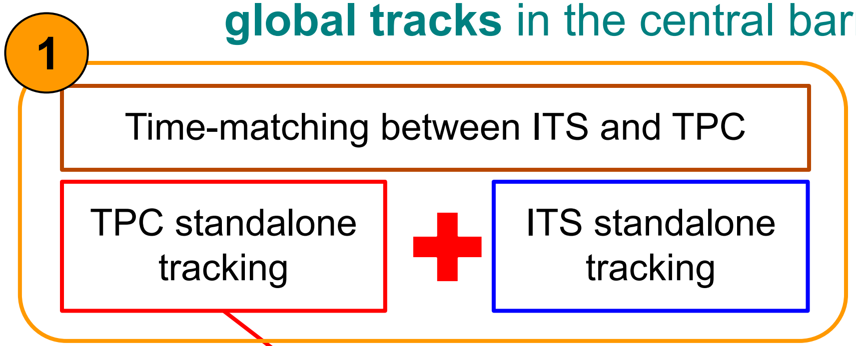
- **O2**: new framework for online/offline data reconstruction and analysis
- Continuous readout of Time Frames (TFs)
- Data reconstruction developed in synchronous + asynchronous phases

ALICE: mid-y tracking

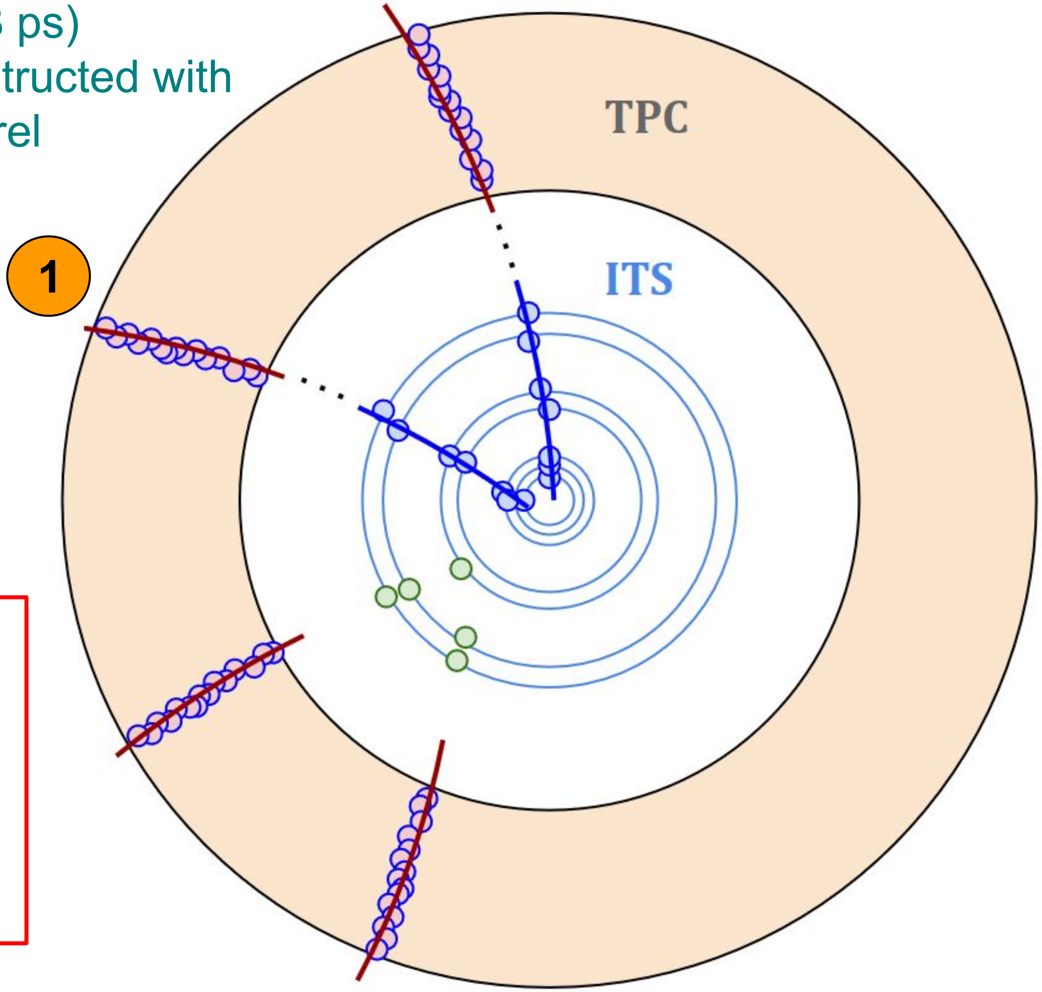
M. Concas for the ALICE collaboration

C. Zampolli for the ALICE collaboration

- Continuous readout of Time Frames (TFs)
- A priori track-to-collision association not possible
- FIT detector:
  - excellent time resolution (σ ≤ 18 ps)
  - good correlation with PV reconstructed with **global tracks** in the central barrel



- Main challenge: data compression in asynchronous reconstruction:
  - TPC: ~3.4 TB/s → ~ 70 GB/s (↓50x)
- Ported to GPUs:
  - Up to 100x gain with GPUs compared to 1-core CPU



ALICE: mid-y tracking

M. Concas for the ALICE collaboration

C. Zampolli for the ALICE collaboration

- Continuous readout of Time Frames (TFs)
- A priori track-to-collision association not possible
- FIT detector:
    - excellent time resolution (σ ≤ 18 ps)
    - good correlation with PV reconstructed with **global tracks** in the central barrel

1

Time-matching between ITS and TPC

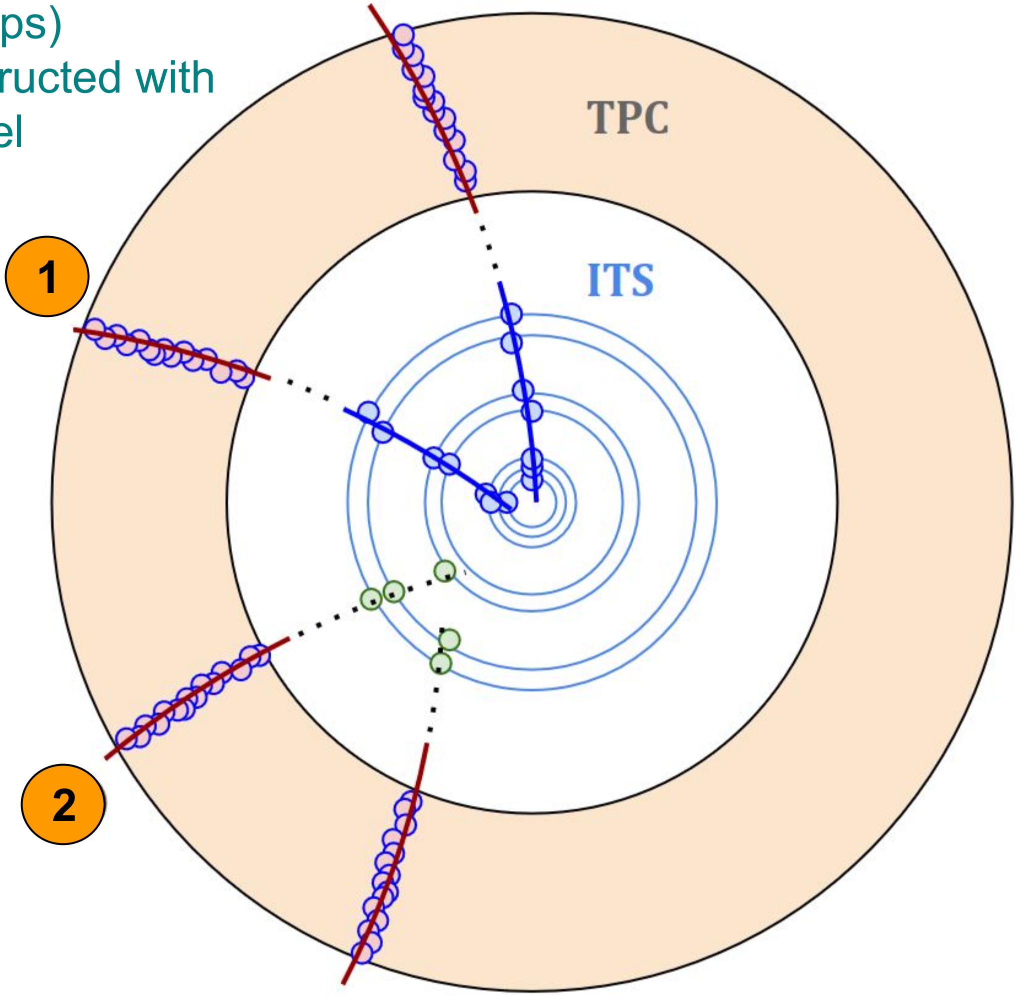
TPC standalone tracking + ITS standalone tracking

2

“Afterburner” algorithm

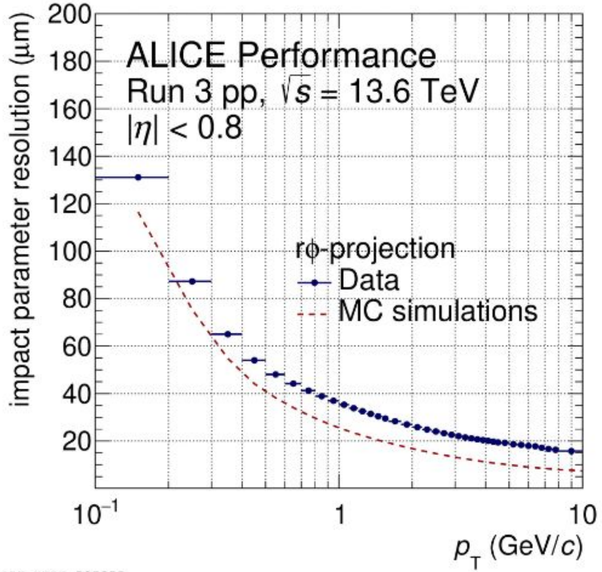
TPC standalone tracking + Unused IT clusters

Useful to recover efficiency for V0/cascades decaying within the ITS

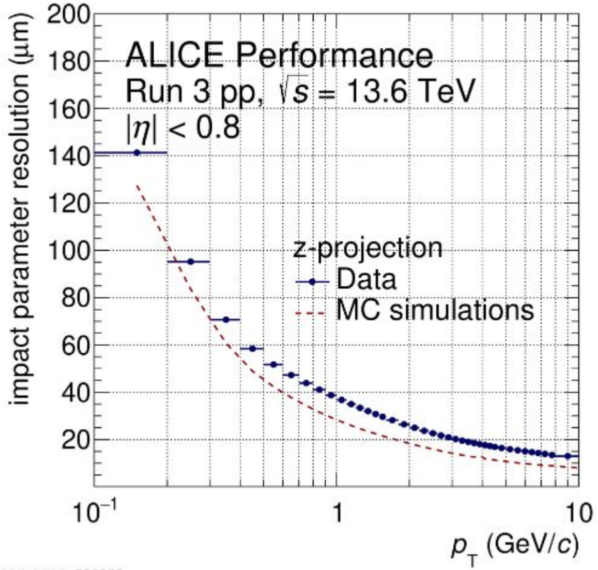


ALICE - tracking performance

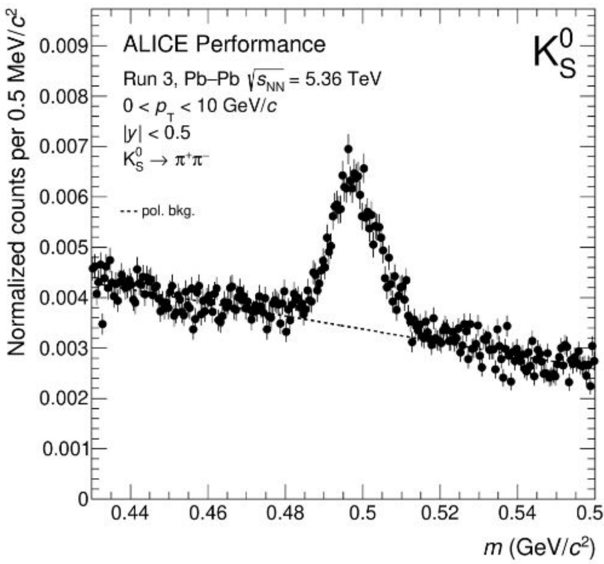
ALICE performance figures repository



ALI-PERF-535955



ALI-PERF-535959



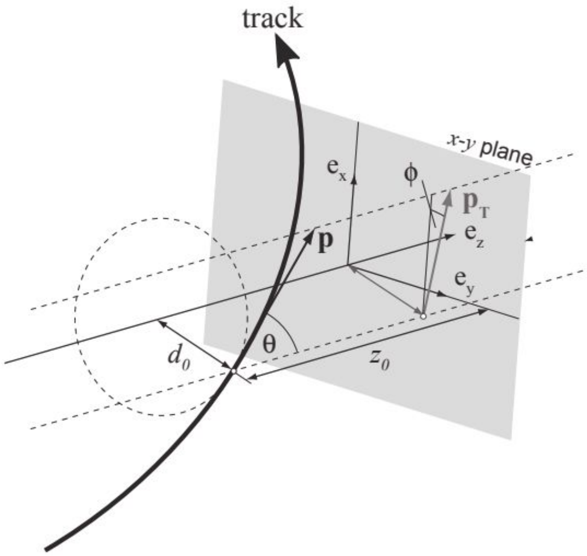
ALI-PERF-529737

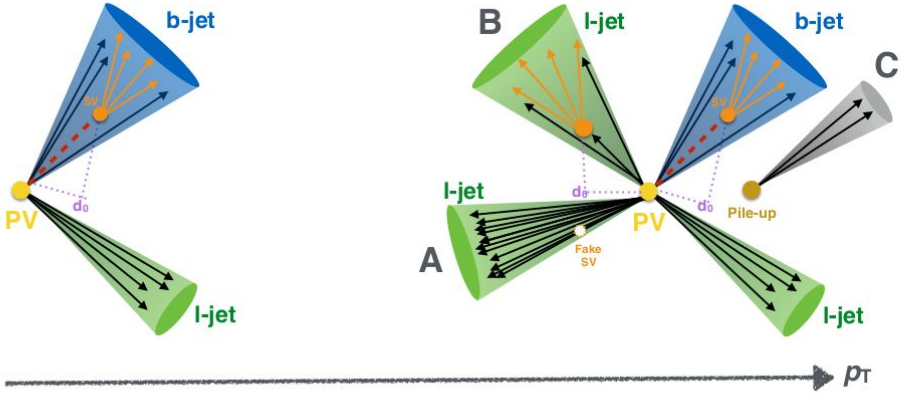
- **Pointing resolution** to the PV of **~35-40 μm @ p_T = 1 GeV/c**
- **2x (4-5x) better** performance in **r-φ (z)** compared to Run 2
- Fine-tuning on TPC calibrations/ITS alignment ongoing to fix residual mismatch with MC

- Nice performance for **$K^0_s \to \pi^+\pi^-$ signal** reconstruction in 2022 Pb-Pb data

Flavor Tagging 101

- Identify jets originating from heavy flavour (b, c) quarks and separate from other sources (e.g. light quarks)
  - Mainly b-tagging;
- Using the topology of heavy-flavour jets;
- Lifetime of the b-hadrons (1.5ps) gives unique properties to the jets:
  - Hard fragmentation;
  - Displaced secondary and tertiary vertices;
  - Large impact parameters (d_0);
- Using different jet- and track variables to distinguish the jet flavour
  - Jet p_T, η
  - Relative track p_T, impact parameter etc.



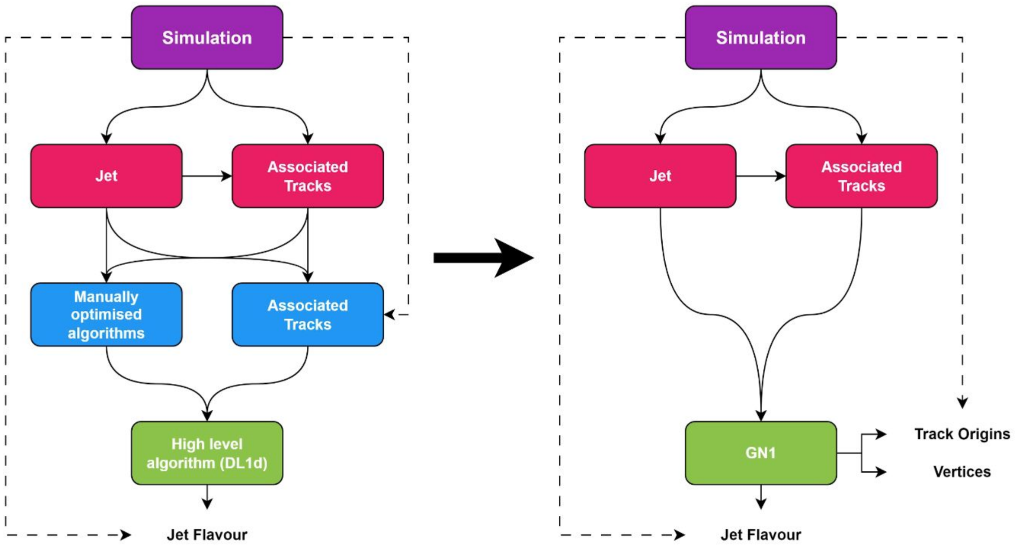


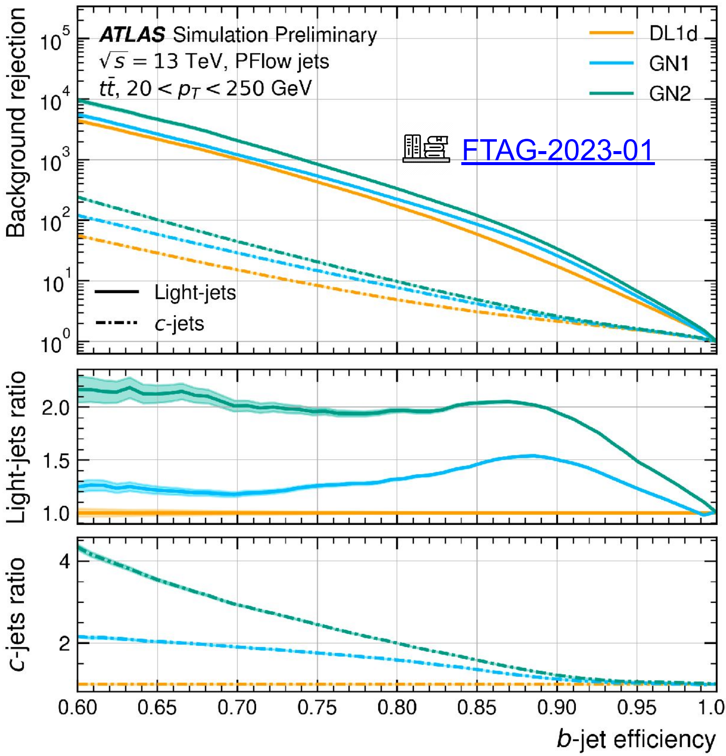
Becomes more complicated at high p_T

ATLAS: GN1/GN2 algorithms

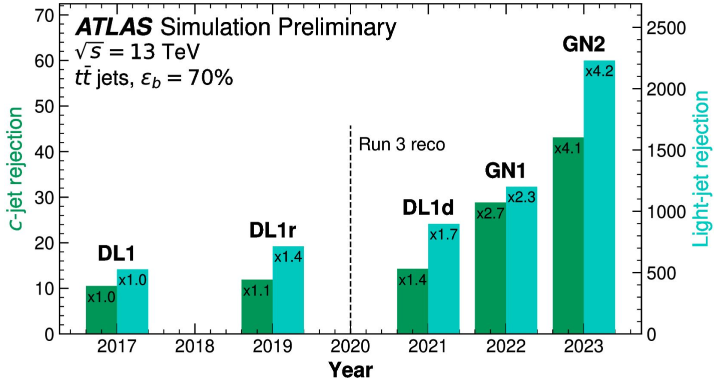
- Previous taggers used two-stage approach;
- Manually optimised algorithms → Low level;
- Final neural network which uses low level algorithms as input → High level;
- New taggers (GN1/GN2) uses-one stage approach:
- Easier to handle → Less manual optimisation!
    - Improved performance

A. Froch for the ATLAS collaboration





FTAG-2023-01

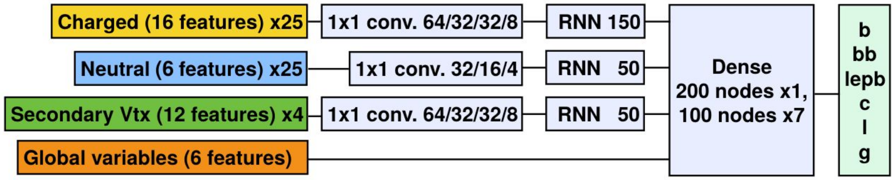


DeepJet tagger at CMS

Using a Deep Neural Network (DNN)

- Using charged and neutral constituents, secondary vertices and global variables of the jet;
- Charged and neutral constituents and secondary vertices variables are automatic feature engineered using 1x1 convolutional layers (CNNs);
- Using Recurrent Neural Networks (RNNs) to further process the information;
- Concatenating RNN outputs and global features and feed it into a multi-classifier DNN;
- Outputs probabilities for jet originating from a certain source;

[arXiv:2008.10519](https://arxiv.org/abs/2008.10519)



Charged (16 features) x25 — 1x1 conv. 64/32/32/8 — RNN 150

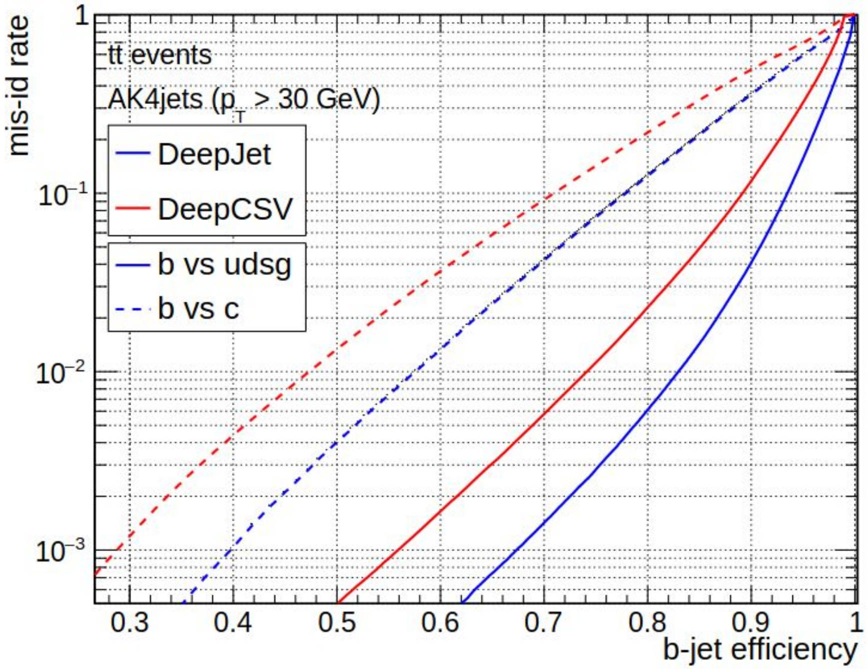
Neutral (6 features) x25 — 1x1 conv. 32/16/4 — RNN 50

Secondary Vtx (12 features) x4 — 1x1 conv. 64/32/32/8 — RNN 50

Global variables (6 features)

Dense
200 nodes x1,
100 nodes x7

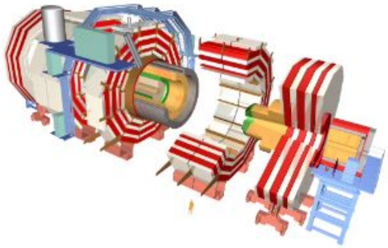
b
bb
lepb
c
l
g



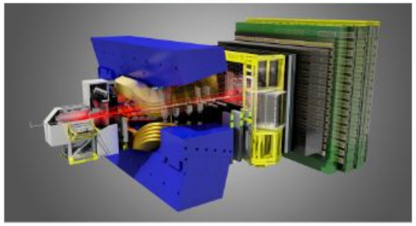
- Large performance gains for `DeepJet` over older `DeepCSV` algorithm for light- and gluon- jet rejection;
- Also: Large performance gains for c-jet rejection!
- Performance gains in higher p_T regions also significant;

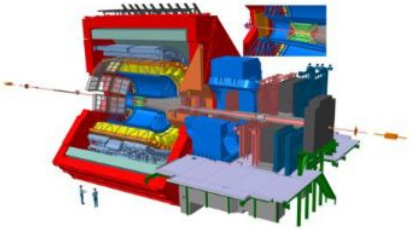
Summary & Outlook

- Tracking algorithms need to provide high-quality tracks efficiently and with an efficient usage of resources:
  - high tracking and vertexing performance in Run 3 (despite challenging conditions at the LHC);
- In order to provide more precise and accurate track reconstruction sophisticated algorithms, techniques and calibrations have been developed.
- Run 3 developments include:
  - pile-up handling;
  - improved tracking in trigger;
  - improved tracking in dense environment;
  - multi-threading and algorithm optimization;
- Experiments ready for fruitful data taking, reconstruction and physics analysis in Run 3!


Compact Muon Solenoid (CMS)


A Toroidal LHC Apparatus (ATLAS)


Large Hadron Collider beauty (LHCb)

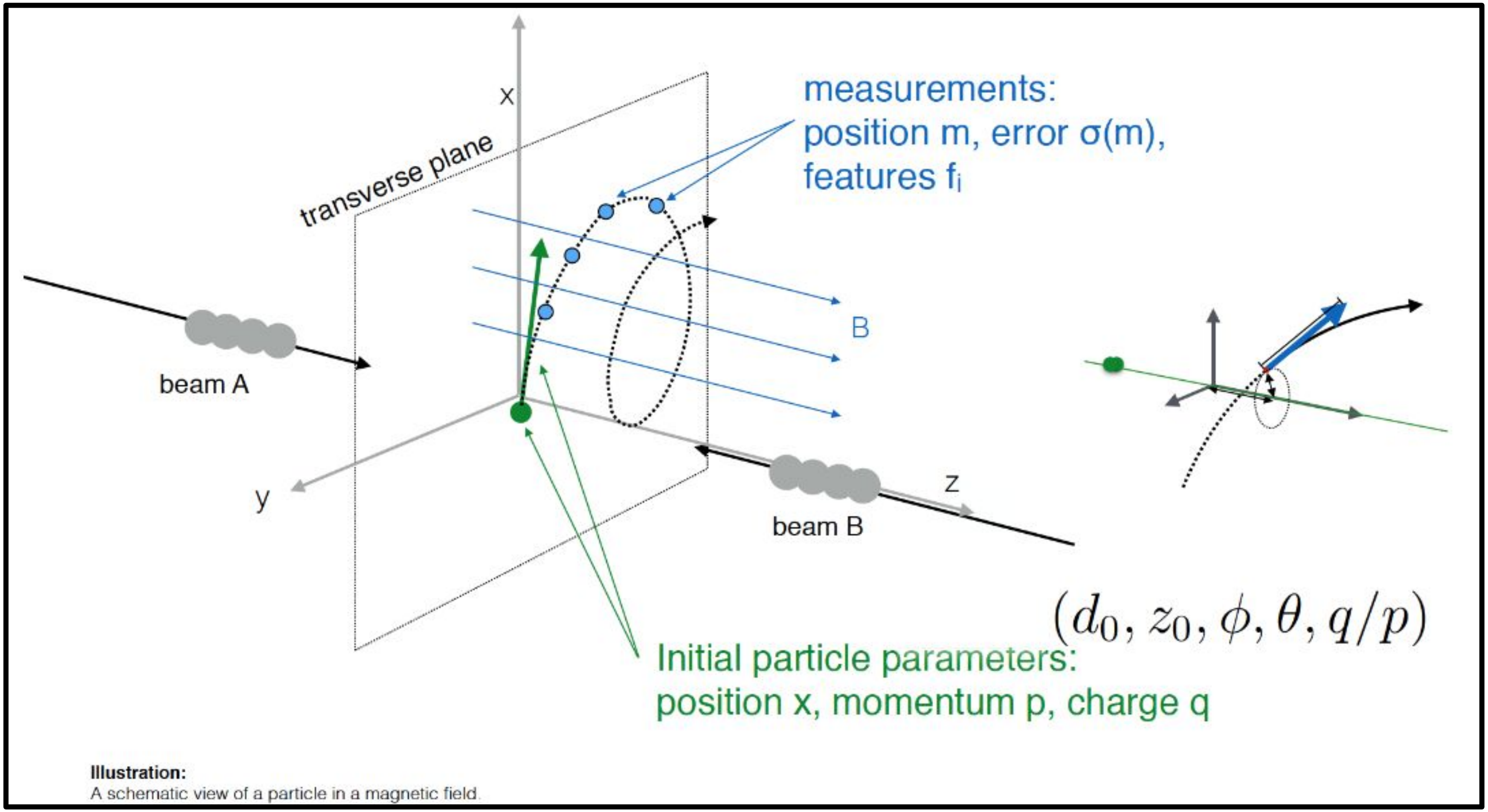

A Large Ion Collider Experimen (ALICE)

Thanks for the attention!

BACKUP

Tracking at the collider: basics

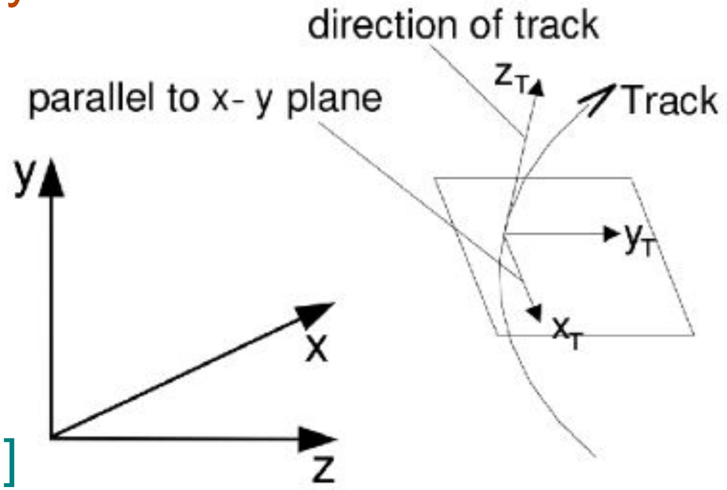
Track reconstruction = Pattern Recognition + Track fitting



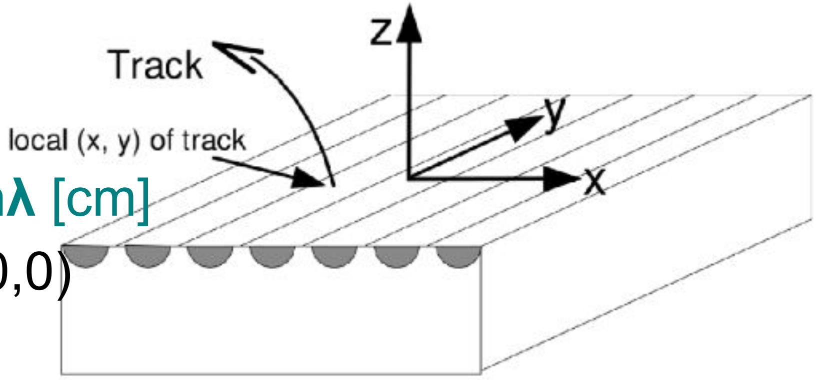
$$(d_0, z_0, \phi, \theta, q/p)$$

Tracking at the collider: formalism

- It turns out there are many "convenient" ways to define a track or its state relative to some reference
    - **Cartesian** (x, y, z, p_x, p_y, p_z)
    - **Curvilinear** ($q/p, \lambda, \varphi, x_T, y_T$)
    - **Local** ($q/p, dx/dz, dy/dz, x, y$)



- Curvilinear actually used in CMS:
    - **q/|p|** signed inverse momentum [GeV^{-1}]
    - **$\lambda = \pi/2 - \theta$** (θ is the polar angle)
    - **φ**, the azimuthal angle
    - **$x_T = - v_x \sin\varphi + v_y \cos\varphi$** [cm]
    - **$y_T = v_z \cos\lambda - (v_x \cos\varphi + v_y \sin\varphi) \sin\lambda$** [cm]
        - (v_x, v_y, v_z) is the track PCA to (0,0,0)



- Local is also used sometimes in tracking

Tracking at colliders: the circle

- Circle: most relevant part of tracking in xy plane
  - common shape for track trajectories
- Radius of the trajectory:

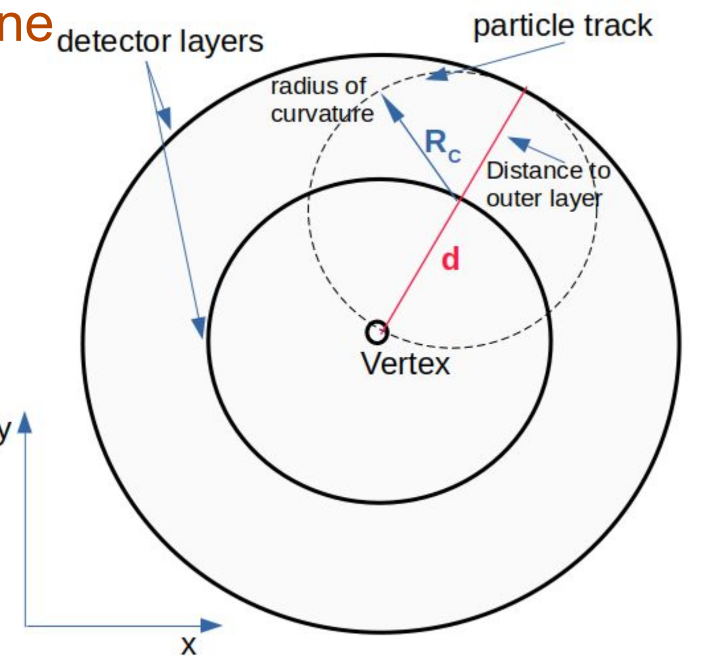
$$\boxed{R_c \approx \frac{p_T}{0.3B_z}}$$

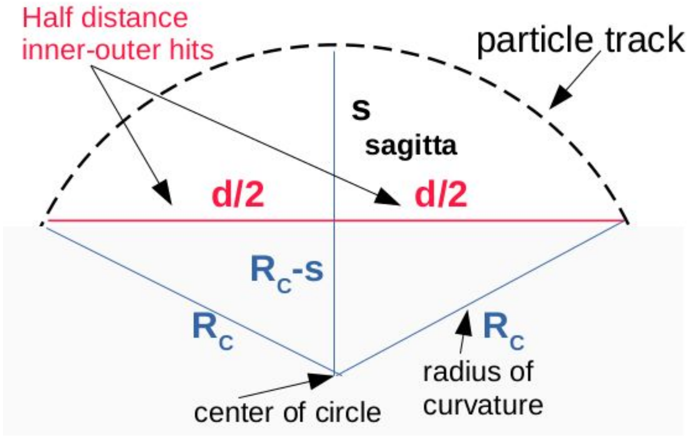
- max p_T for trajectory that loops inside tracker:
  - for CMS d ~ 1 m

$$\boxed{p_{T,d} \approx 0.3\frac{d}{2}B_z[CMS :\approx 0.6\mathrm{GeV}]}$$

- sagitta, *s*, relates to track p_T in a simple way:

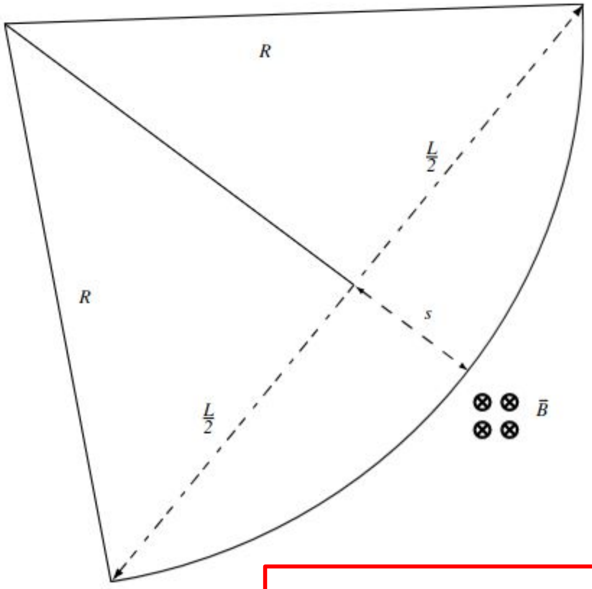
$$\boxed{p_T \approx p_{T,d}\frac{d}{4s}}$$





Tracking performance: p_T

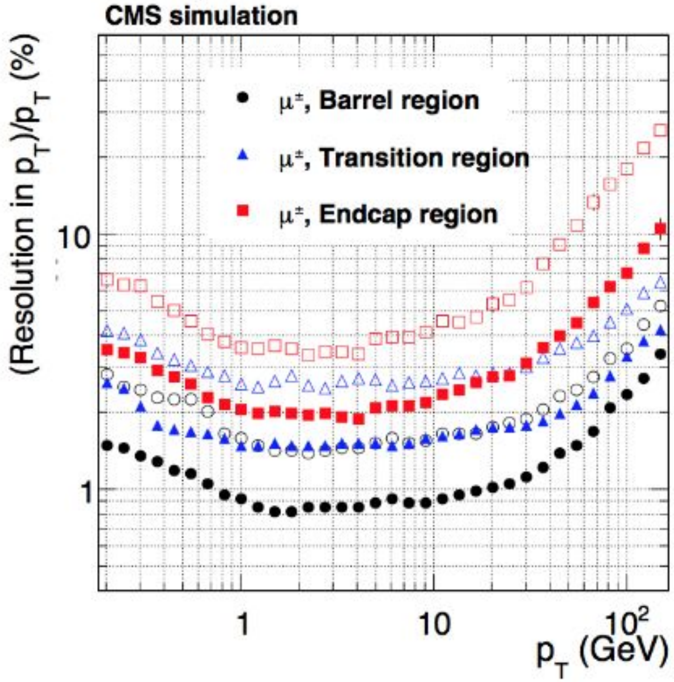
$$\boxed{p_T[\mathrm{GeV}] = 0.3\,B[T] \times R[m]}$$



- s: sagitta
- R: radius of curvature
- sagitta — bigger B, L is better
  - large detector with strong B field
- p_T resolution peters out at higher momentum
  - CMS: 2-3% resolution at pT ~ 100 GeV

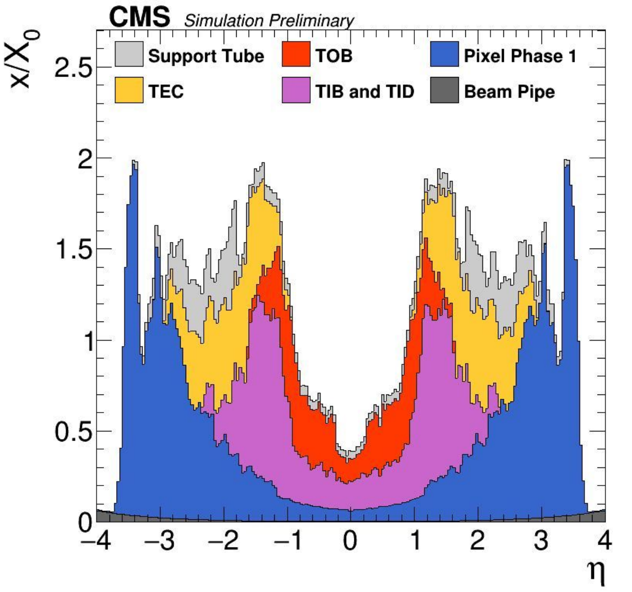
$$\boxed{s \simeq \frac{L^2}{8R} \propto \frac{BL^2}{p_T}}$$

$$\boxed{\delta p_T / p_T \propto c \times p_T}$$



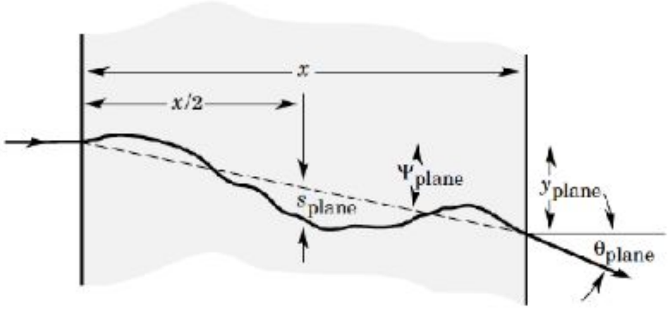
- Momentum resolution degrades with p_T

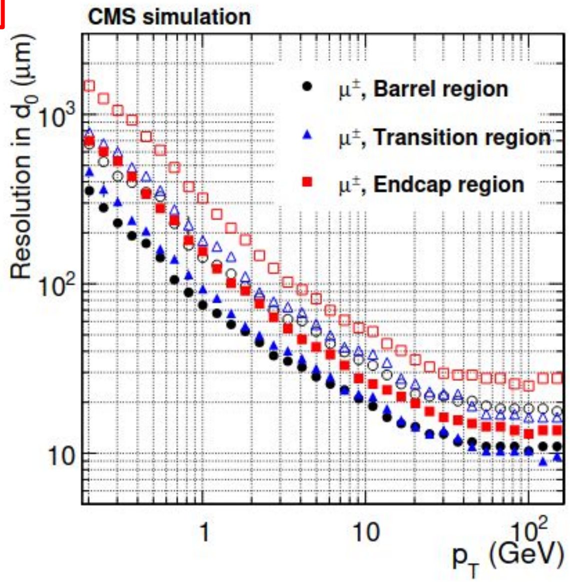
Tracking performance: IP

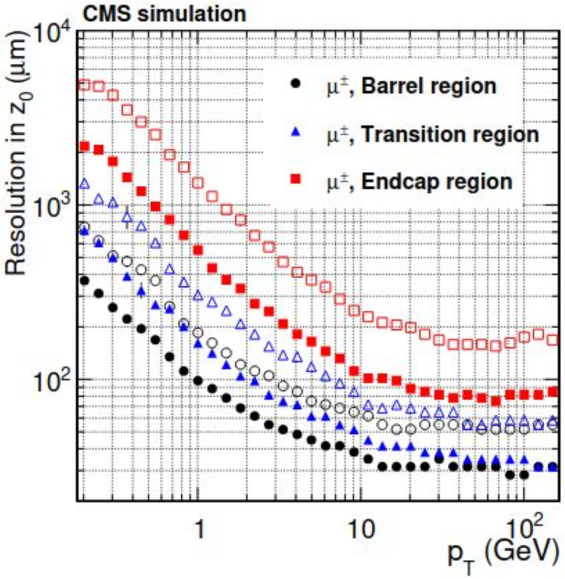


- Impact parameter resolution increases with decreasing pT
    - Limited by hit resolution and alignment at high end;
    - Limited by multiple scattering at low end;
- Recall: multiple scattering

$$\theta_{\text{plane}} \approx \frac{14\text{MeV}}{p}\sqrt{x/X_0}$$

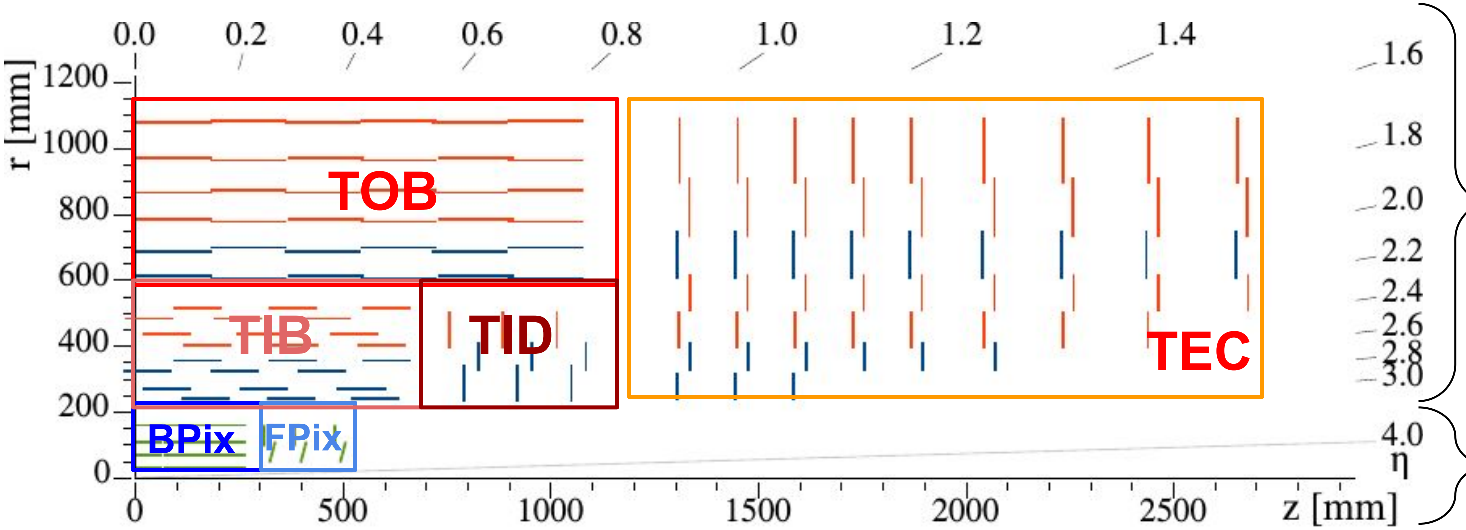






CMS Silicon Tracker

- All-silicon design:
  - Allows for high-precision charged particle tracking up to $|\eta|<3$;
  - Essential in particle identification, heavy-flavour tagging, trigger decisions, vertex reconstruction;
  - Largest Si tracker in the world: ~200 m^2 area, ~135M electronic channels
- Comprised of the Pixel (innermost parts)
  - 4 layers in the barrel (BPix) and 3 disk (FPix) in the forward regions:
    - 1,856 Pixel modules.
- and the Strips sub-detectors (outer parts)
  - 10 layers in the barrel (TIB, TOB) and 12 forward disks (TID, TEC):
    - 15,148 Strips modules.



- Micro-strip sensors
- Stereo modules (two components with a 100 mrad stereo angle)
- Analog readout

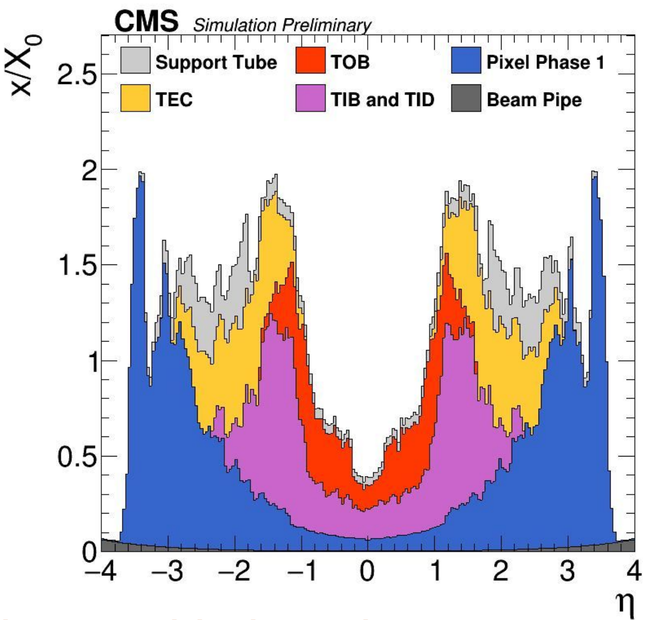
- pixel sensors
- ~124M channels
- digital readout

Track reconstruction in CMS

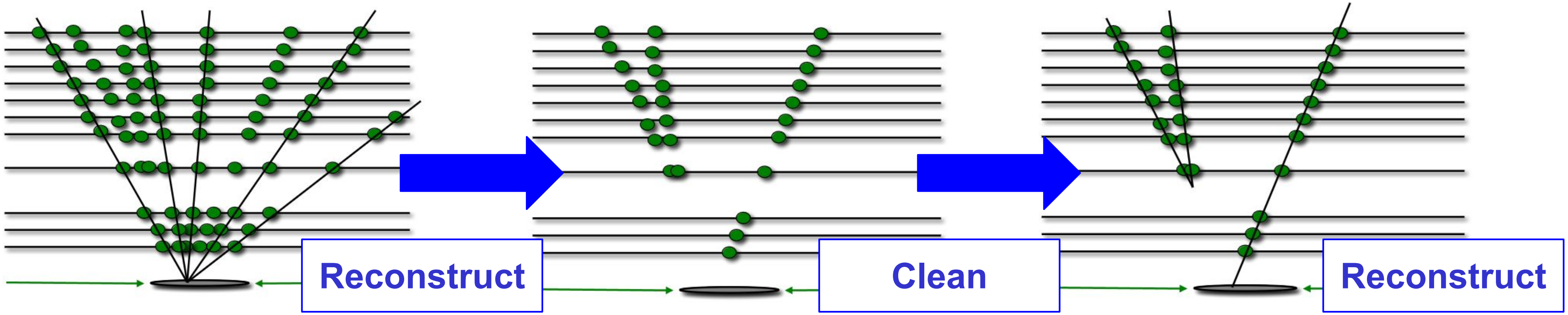
- Few, but precise measurements;
- Non negligible amount of dead material inside the tracker volume

R_{inner}[cm]	R_{outer} [m]	|η| coverage	B field [T]
3	1.1	3.0	3.8

X_0 @ |η|=0	p_T resolution @1 (100) GeV, |η|=0	d_0 resolution @1 (100) GeV, |η|=0 [μm]
0.4	0.7 (1.5)%	90 (20)



- Main tracking algorithm: Combinatorial Track Finder used in iterative steps:
  - limits the number of combinatorics in pattern recognition
  - tracking reach guarantee, w/o degrading computing performance

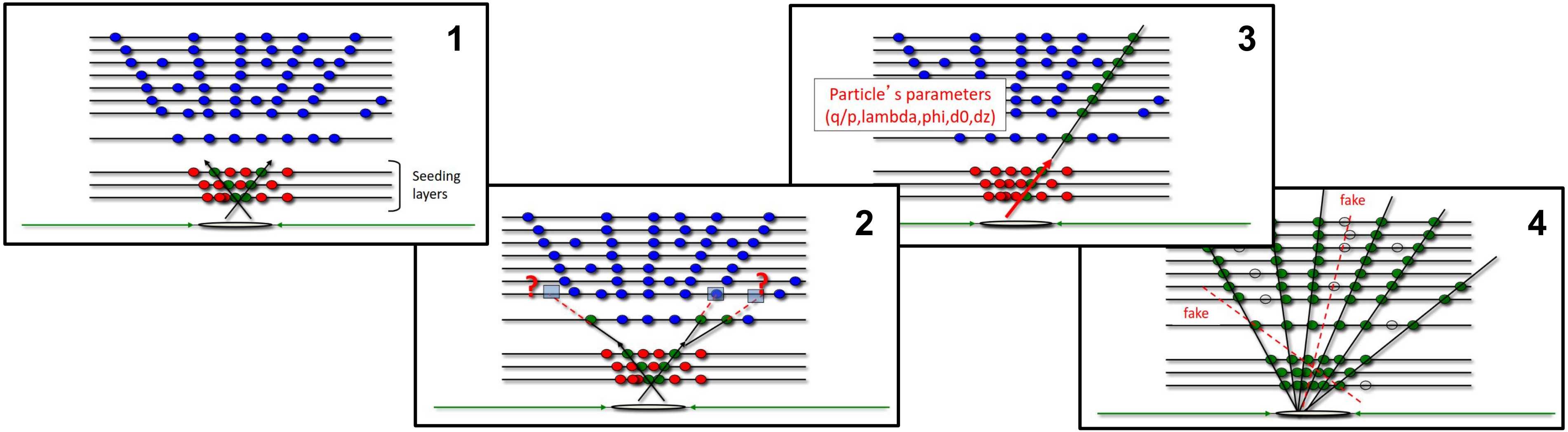

Reconstruct

Clean

Reconstruct

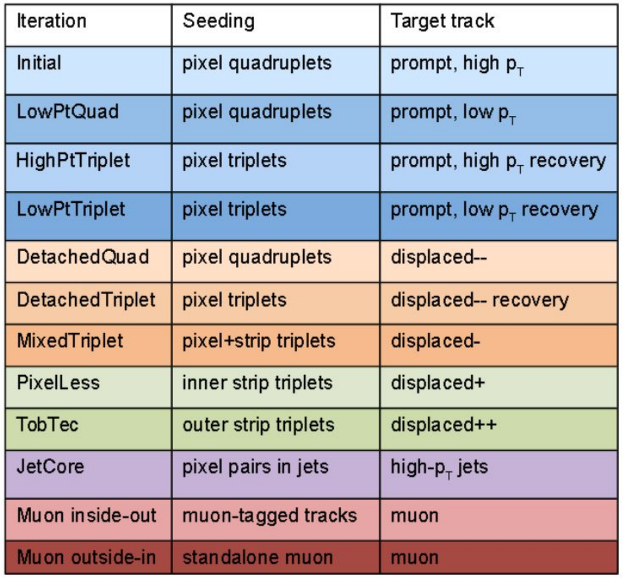
Track reconstruction in CMS

- In each iteration, tracks are reconstructed in four steps:
- **1. Seeding**:
    - provides track candidates, with an initial estimate of the trajectory parameters and their uncertainties (use combination of pixel, strip or mixed hits);
- **2. Pattern recognition**:
    - hits compatible with the predicted track position are added (Kalman update) to the trajectory and track parameters are updated;
- **3. Final fit**:
    - taking into account the B-field non uniformity and a detailed description of the material budget;
    - provides the best estimate of the parameters of each smooth trajectory after combining all associated hits (outlier hits are rejected);
- **4. Selection**:
    - sets quality flags based on a ML-based MVA with more than 20 inputs;
    - aims to reject fake tracks; tracks sharing too many hits are also cleaned as duplicates;

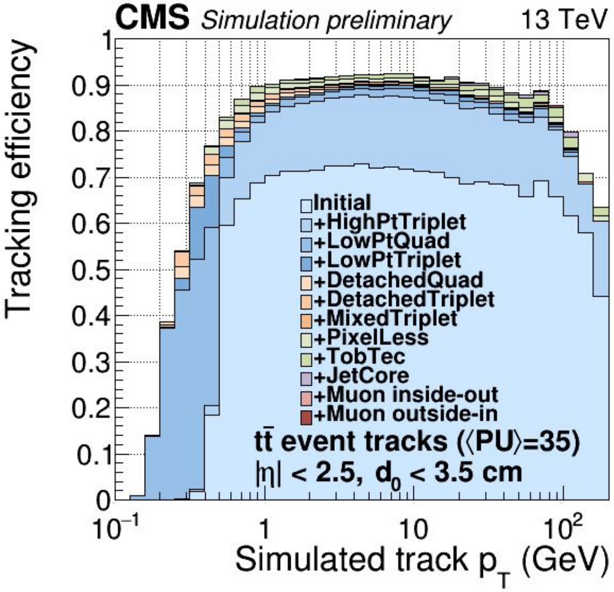


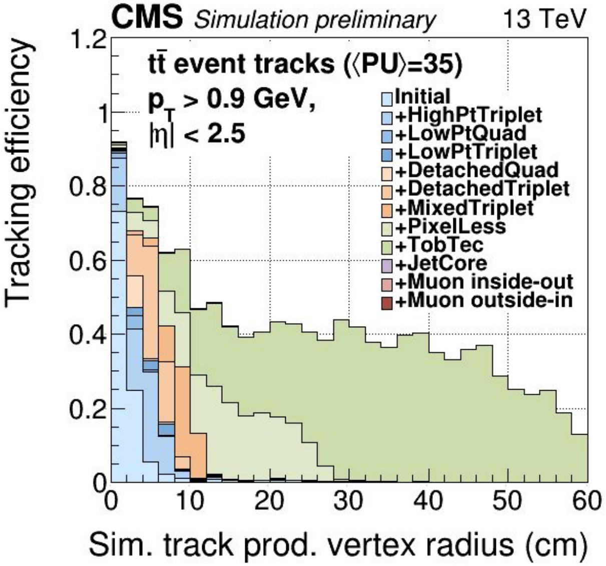
Iterative tracking at CMS

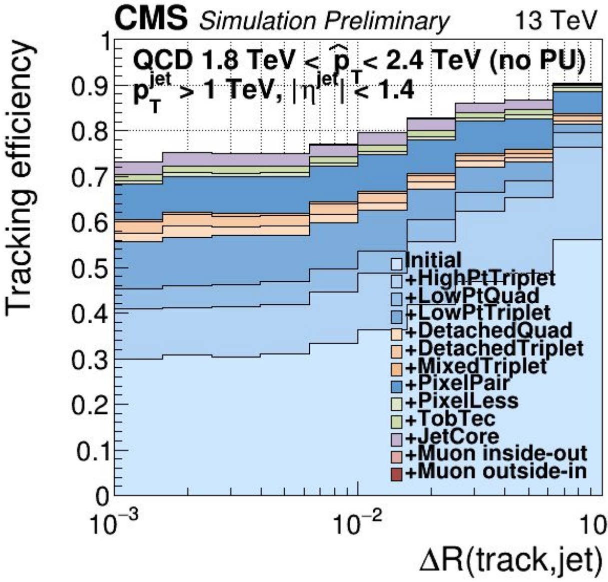
- Tracks reconstruction is an iterative procedure:
  - the InitialStep makes use of high-pT quadruplets coming from the beam spot region
  - Subsequent steps use triplets, or improve the acceptance either in pT or in displacement
  - the later steps use seeds w/ hits from the strip detector to find detached tracks,
  - final steps are dedicated to special phase-space
    - highly dense environment (i.e. within jets)
    - clean environment (i.e. muons)



Iteration	Seeding	Target track
Initial	pixel quadruplets	prompt, high p_T
LowPtQuad	pixel quadruplets	prompt, low p_T
HighPtTriplet	pixel triplets	prompt, high p_T recovery
LowPtTriplet	pixel triplets	prompt, low p_T recovery
DetachedQuad	pixel quadruplets	displaced--
DetachedTriplet	pixel triplets	displaced-- recovery
MixedTriplet	pixel+strip triplets	displaced-
PixelLess	inner strip triplets	displaced+
TobTec	outer strip triplets	displaced++
JetCore	pixel pairs in jets	high-p_T jets
Muon inside-out	muon-tagged tracks	muon
Muon outside-in	standalone muon	muon



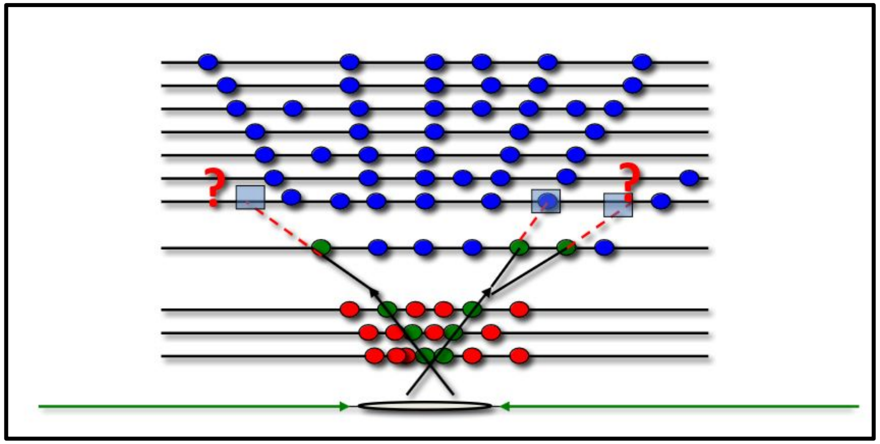




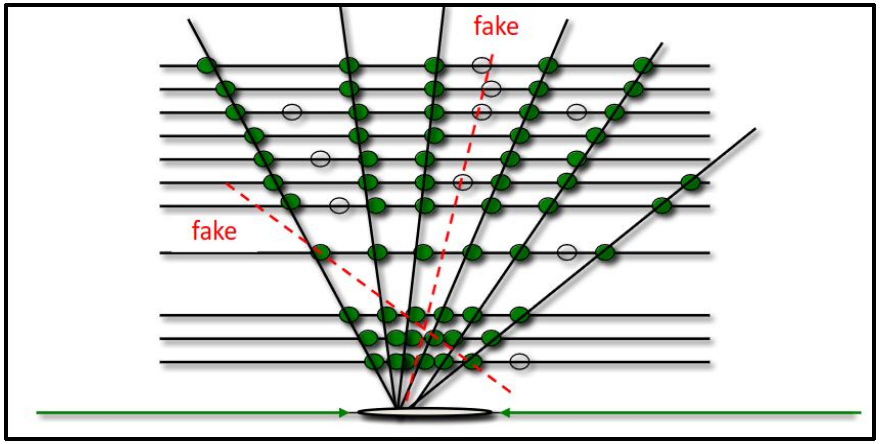
CMS Algorithms for Run 3

- Developments during the LHC Long Shutdown 2 focused on the tracking algorithmic improvements targeted to reconstruction timing and tracking fake rate:
  - Parallelization and vectorization at multiple levels using Kalman Filter, since including the mkFit algorithm (CMS-DP-2022-018)
  - After final fit, track quality is assessed with track classifier: from a Boosted Decision Tree to a Deep Neural Network (CMS-DP-2023-009)

Track Building step: from CKF to mkFit

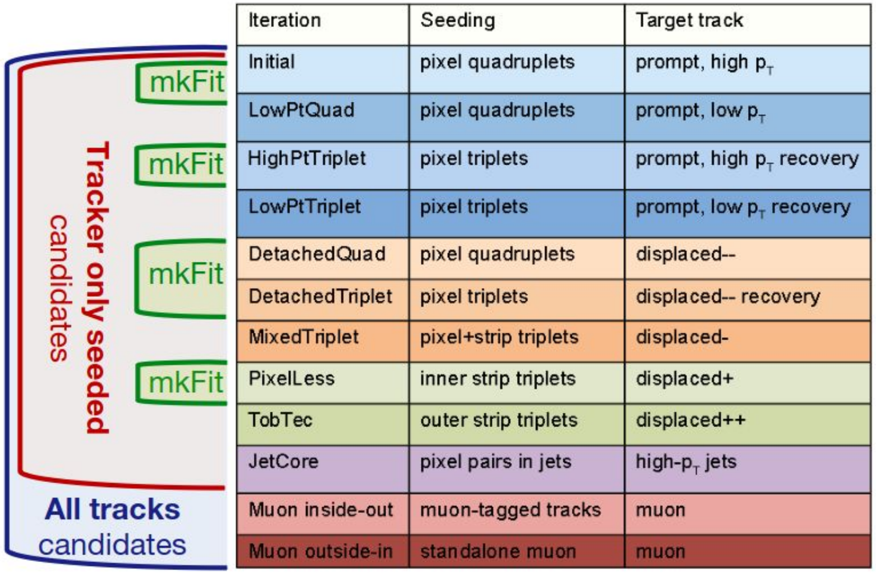


Track Selection step: from BDT to DNN



mkFit algorithm

- In Run 2, the CMS track reconstruction algorithm used an iterative approach based on combinatorial Kalman Filter (CKF), consisting of twelve main iterations targeting different track topologies and seeded with different seed tracks.
- For Run 3, a new algorithm has been developed for track pattern recognition (or track building), named mkFit, that maximally exploits parallelization and vectorization in multi-core CPU architectures. This algorithm has been deployed in the CMS software for a subset of tracking iterations:
    - **InitialPreSplitting**:
        - initial iteration before splitting merged pixel clusters in dense jet environments;
    - **Initial**:
        - initial iteration;
    - **HighPtTriplet**:
        - high-pT triplet iteration;
    - **DetachedQuad**:
        - detached quadruplet iteration;
    - **DetachedTriplet**:
        - detached triplet iteration;

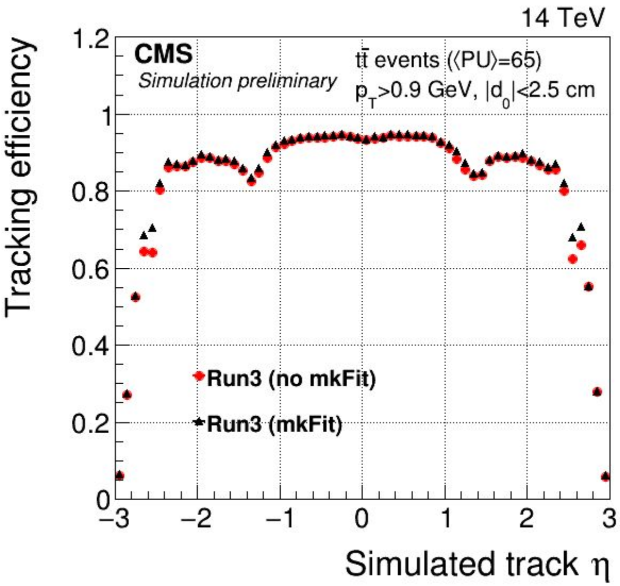


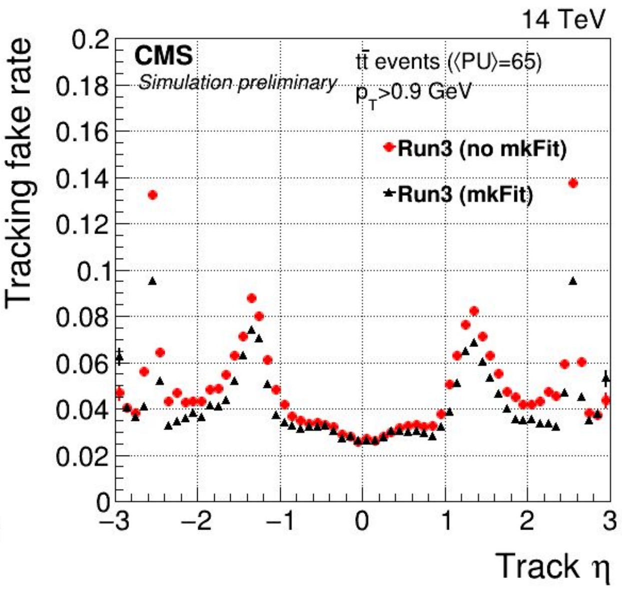
Iteration	Seeding	Target track
Initial	pixel quadruplets	prompt, high p_T
LowPtQuad	pixel quadruplets	prompt, low p_T
HighPtTriplet	pixel triplets	prompt, high p_T recovery
LowPtTriplet	pixel triplets	prompt, low p_T recovery
DetachedQuad	pixel quadruplets	displaced--
DetachedTriplet	pixel triplets	displaced-- recovery
MixedTriplet	pixel+strip triplets	displaced-
PixelLess	inner strip triplets	displaced+
TobTec	outer strip triplets	displaced++
JetCore	pixel pairs in jets	high-p_T jets
Muon inside-out	muon-tagged tracks	muon
Muon outside-in	standalone muon	muon

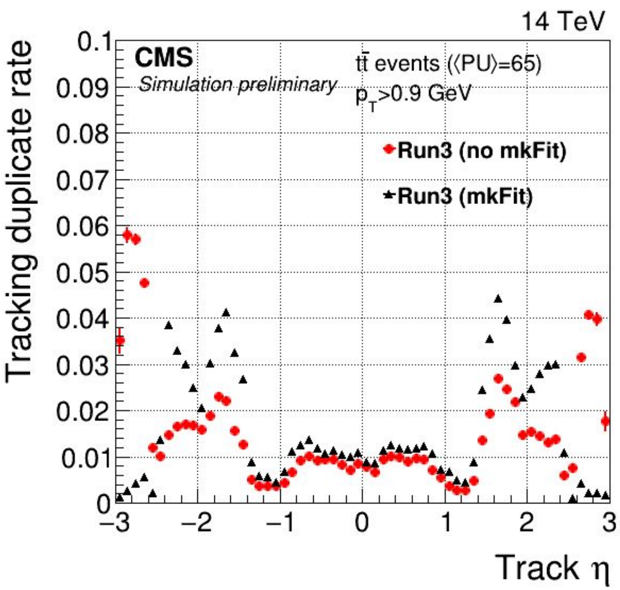
- The mkFit algorithm allows to retain a similar physics performance with respect to the traditional CKF-based pattern recognition, while substantially improving the computational performance of the CMS track reconstruction

`mkFit` physics performance

- The performance has been measured in a simulated $t\bar{t}$ sample with superimposed pileup events 55 to 75 (flast). The detector conditions account for the residual radiation damage due to Run 2 operations.
- When mkFit is used for track building in a subset of iterations:
  - The **tracking efficiency** is consistent with the one obtained with the traditional CKF tracking algorithm;
  - The **tracking fake rate** is on average lower than the one obtained with the traditional CKF tracking algorithm;
  - The **tracking duplicate rate** is higher than the one obtained with the traditional CKF tracking algorithm especially at 1.45<|η|<2.5, while it's lower at |η|>2.5.

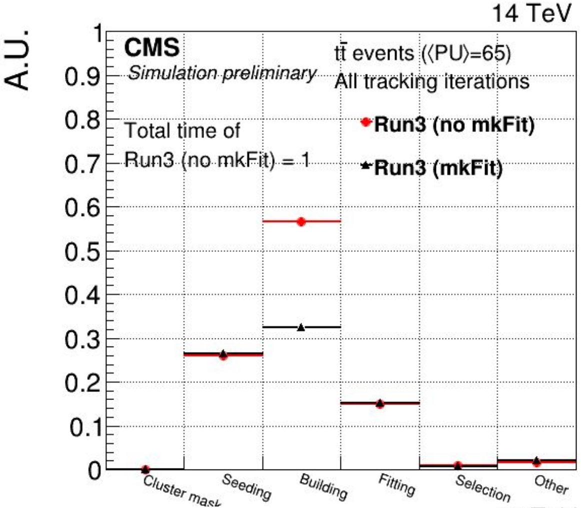




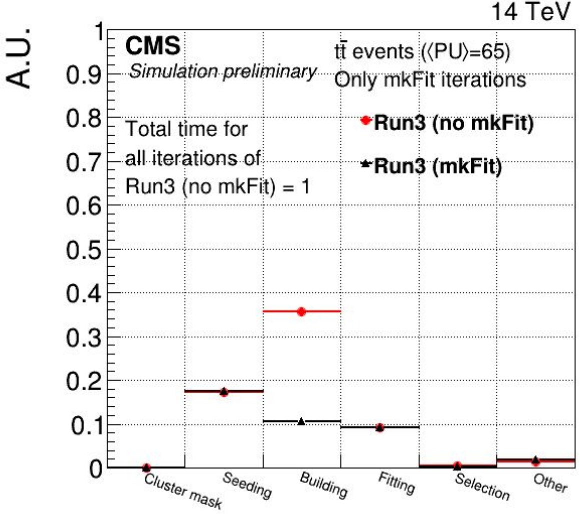


`mkFit` timing performance

- The tracking time performance has been measured in the same simulated $t\bar{t}$ sample with superimposed pileup (PU) events as for the physics performance
- Single-threaded measurements are performed with local access to the input



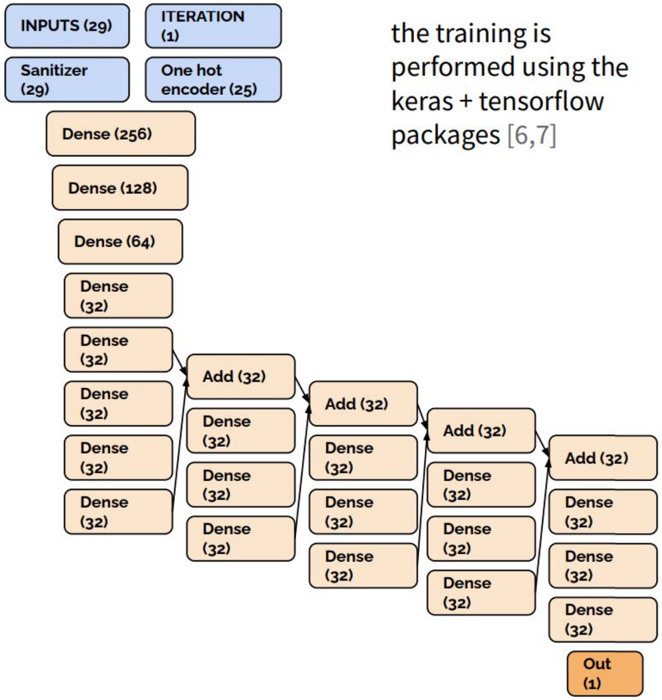
- Overall, using `mkFit` in a subset of tracking iterations allows to **reduce the track building time by a factor of about 1.7**, corresponding to a reduction of the total tracking time by about 25%. In Run 3, tracking has been measured to make about half of the total offline reconstruction time.
  - Thus, this translates to a reduction of the total offline CMS reconstruction time or conversely to **an increase of the event throughput by 10-15%**.



- Using `mkFit` allows to **reduce the track building time by a factor of about 3.5** considering the sum of iterations where `mkFit` is employed.
  - In individual iterations where `mkFit` is employed, this factor varies from about 2.7 to about 6.7.

CMS: Track Selection DNN

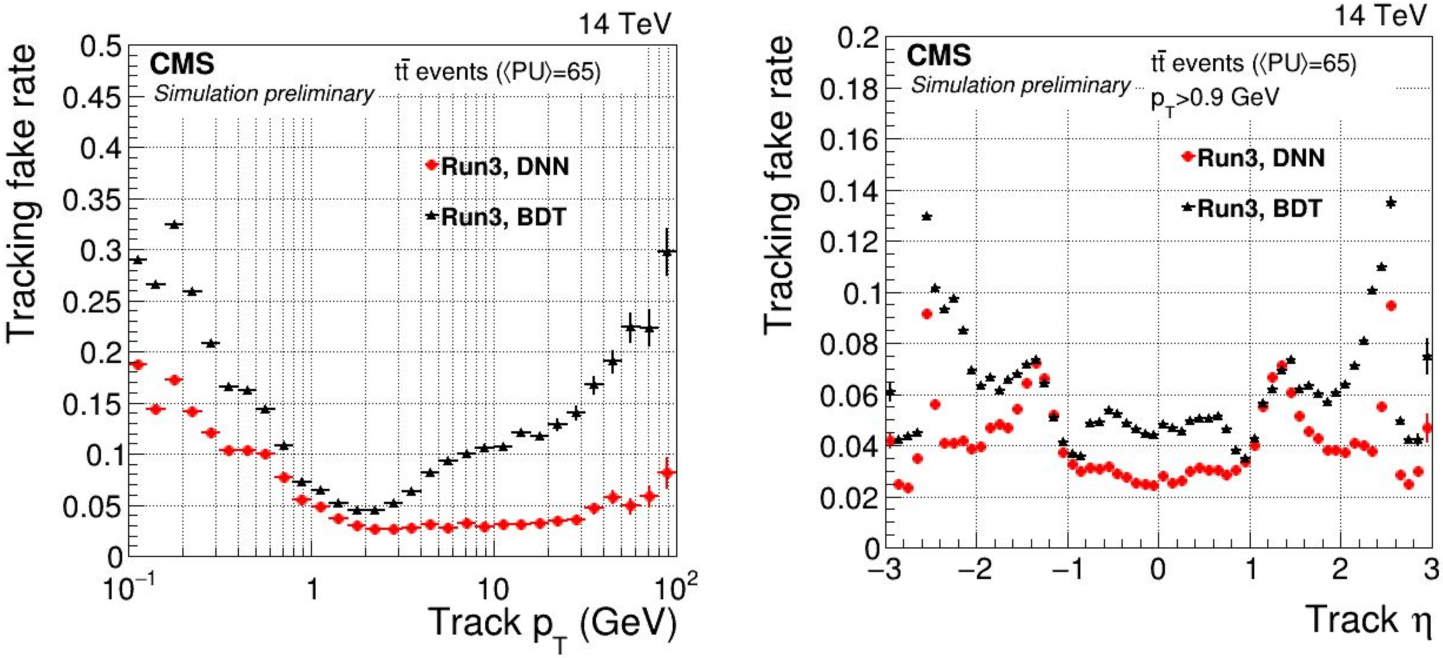
- After the pattern recognition and the fit, based on Kalman Filter techniques, high purity tracks are selected and the hits belonging to those tracks are not used in the following iterations, thus keeping the complexity of the pattern recognition under control for later iterations.
  - The track selection was gradually improved: starting with a parametric selection in Run 1, moving to a BDT in Run 2, and to a DNN in Run 3.
- DNN Architecture:
  - Relatively simple feed-forward network, with 5 iteration of “skip connection” and sum of the layer outputs in the downstream layers;
  - The “sanitizer” layer applies log/absolute value transformations to some of the inputs, while the “one hot encoder” converts the iteration flag into a boolean vector by category;
  - **Activations**: ELU in hidden layers, sigmoid for output;
  - **Loss function**: binary cross-entropy;



the training is performed using the keras + tensorflow packages [6,7]

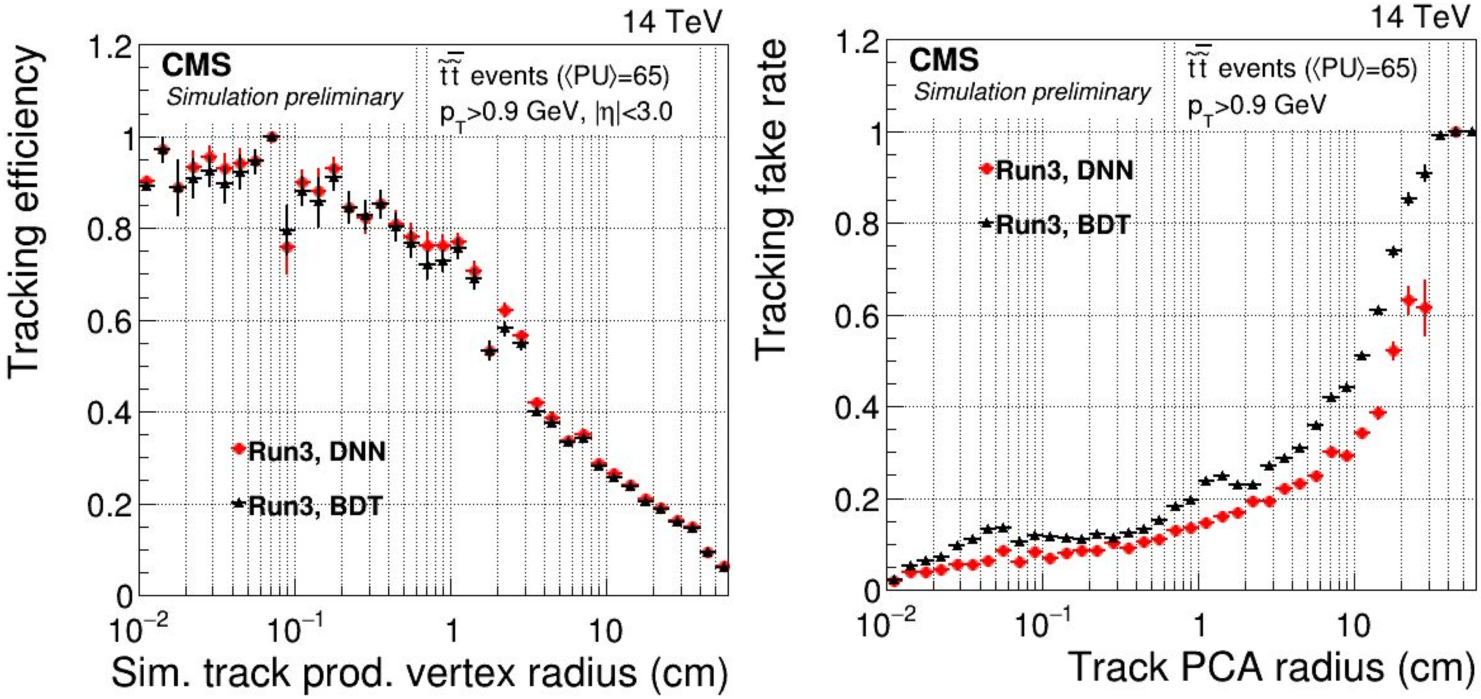
CMS DNN performance

- The performance has been measured in a simulated $t\bar{t}$ sample.
  - The physics results are shown after applying the high purity BDT or DNN selection to each iteration and after merging all the tracks from the iterations into one collection.
- The tracking fake rate when the DNN is used is notably lower than the one obtained using the BDT:
  - especially for very low and very high p_T values. Overall the fake rate is reduced by about 40%.
  - the largest fake rate reductions are in the tracker endcaps (|η|>2) and in the barrel (|η|<1). The discontinuities follow the tracker regions.



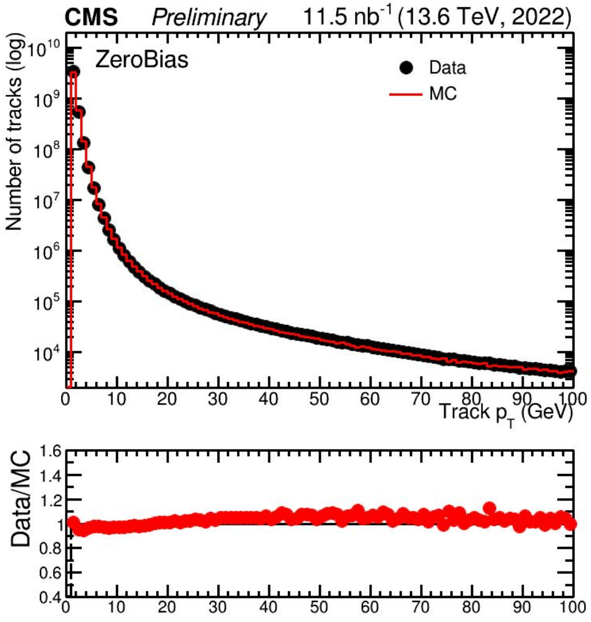
CMS: DNN track selection

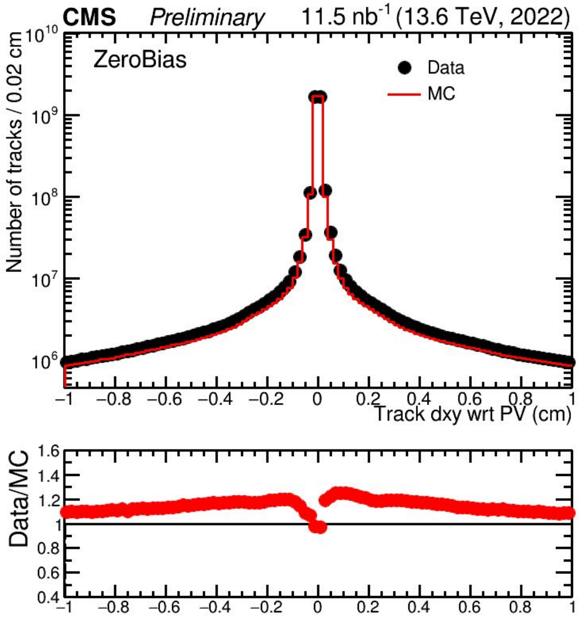
- The performance has been measured in a sample with stop-antistop production in RPV SUSY, where the stops have a significant decay length and produce displaced tracks,.
- The physics results are shown after applying the high purity BDT or DNN selection to each iteration and after merging all the tracks from the iterations into one collection.
    - The tracking efficiency when the DNN is used is consistent or slightly higher than the one obtained using the BDT at all radii.
    - The tracking fake rate when the DNN is used is lower than the one obtained using the BDT across all the radii values, with a reduction of about 30%.

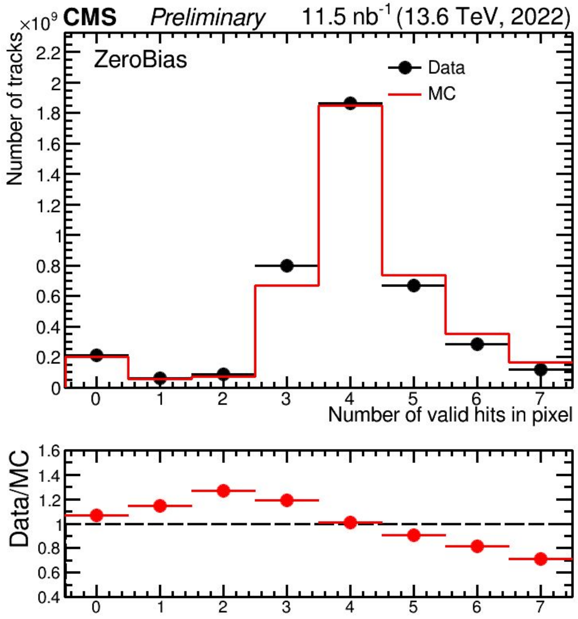


CMS Tracking Performance

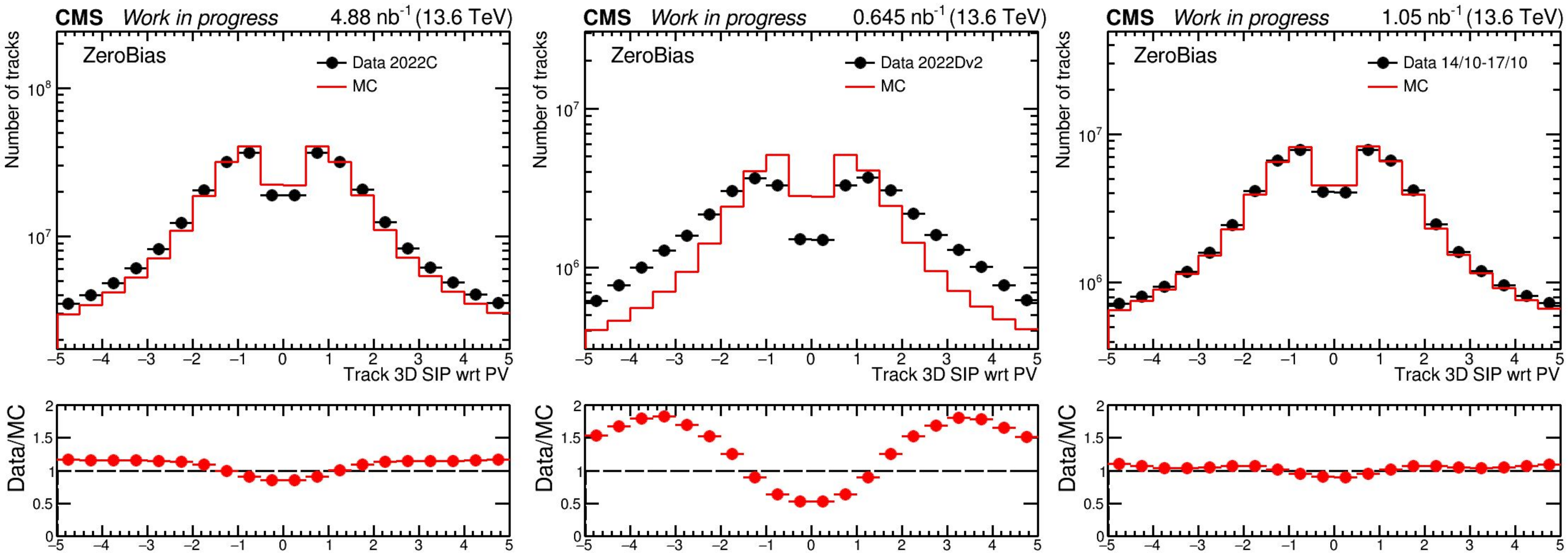
- The figures in the following show a comparison between 2022 CMS data and MC of the reconstructed track properties (documented in CMS-DP-2022-064).
    - Events used are selected with minimal trigger bias, using only the information on the beam-beam coincidence, and were collected from July 19th, 2022 to October 17th, 2022 (with the exception of the period from August 23rd to September 27th). The trigger which is used collects only a fraction of delivered events.
    - the tracks which are considered are tracks which pass the `highPurity` selection (see previous slides), with p_T>1GeV.
    - MC distributions are normalized to the number of vertices in data.
- Overall and without further corrections a **reasonable agreement** is found between data and simulation.





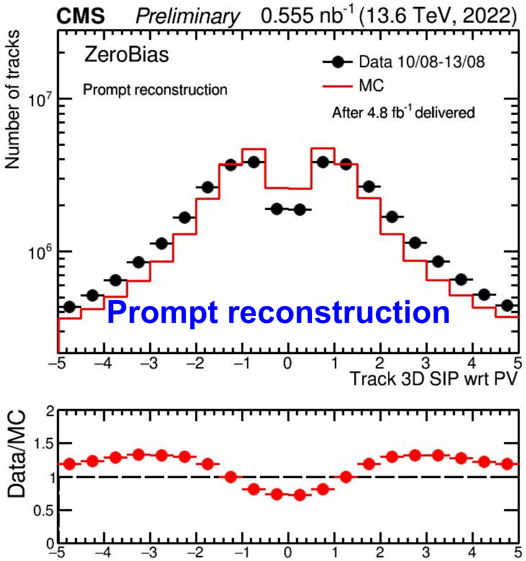


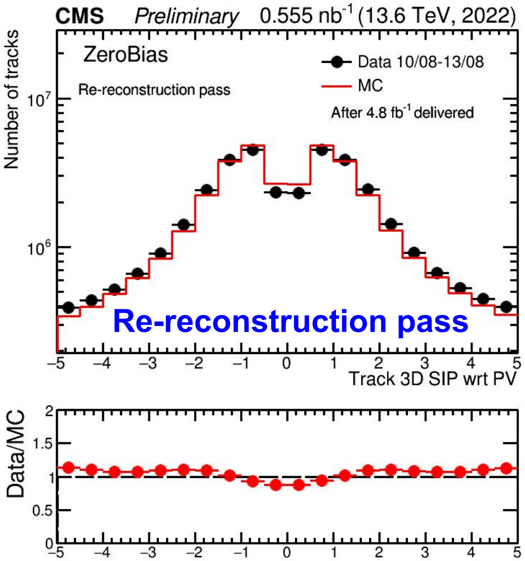
CMS Tracking Performance

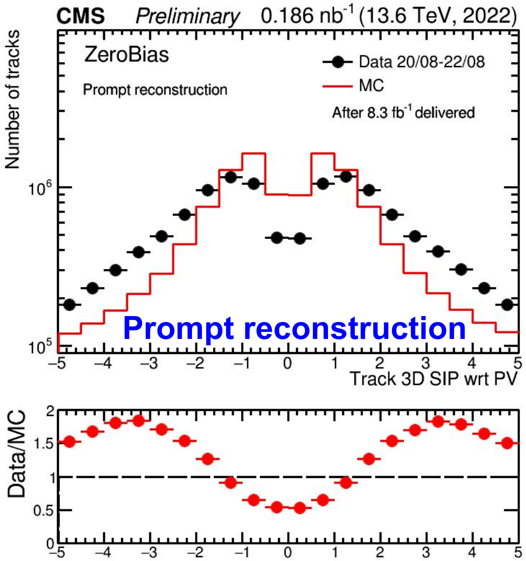


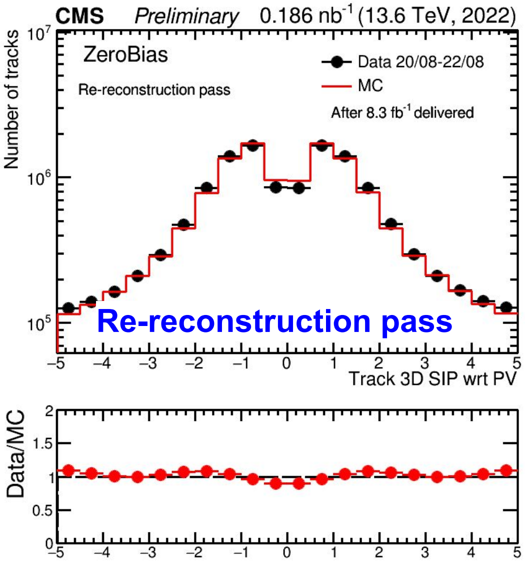
- The figures show the distributions of the significance of 3D impact parameters with respect to the Primary Vertex of tracks from events passing the selection described above.
- Comparisons are shown for the different periods of time shown in the figures, after the indicated luminosity was delivered since the installation of the new BPix layer 1.
  - Agreement between data and MC gets worse over time, indicating aging of BPix layer 1 due to accumulated irradiation.
  - Improvement in agreement in the latter data taking period due to an update in the high-voltages and in the alignment which has been implemented later in the data-taking.

CMS: Tracking Performance





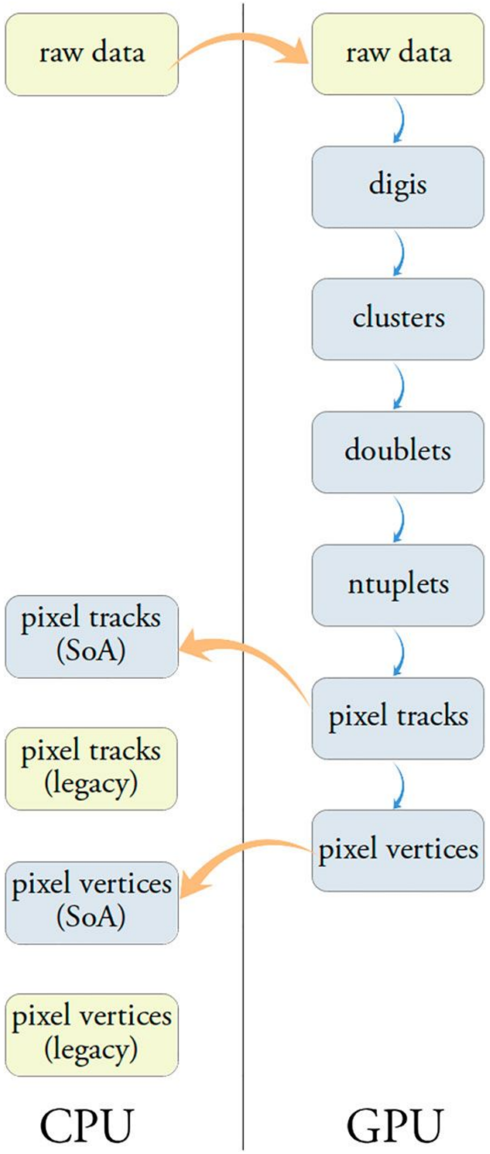




- The figures show the distributions of the significance of 3D impact parameters with respect to the Primary Vertex of tracks from events, passing the selection described above.
- In this case only those events which have been re-reconstructed are considered here.
- Re-reconstruction includes updates to pixel local reconstruction and the alignment of the tracker, leading to better performance.
  - The figures on the left shows the prompt reconstruction, the figures on the right shows the re-reconstruction pass, for the period indicated and after the indicated luminosity was delivered since the installation of the new BPix layer 1.
  - Variables connected to impact parameters (hence used for b/tau tagging, etc.) are the ones most improved by the re-reconstruction conditions, as expected from the updates previously indicated.
- The agreement between data and MC is much better for re-reconstructed data.

CMS: HLT Tracking

- The CMS High-Level Trigger (HLT) runs a version of the full event reconstruction optimized for fast processing.
    - Since the start of Run 3, the HLT makes use of a heterogeneous computing farm.
- In Run 3, HLT tracking is based on a single iteration of the Combinational Kalman Filter, seeded by pixel tracks reconstructed by the Patatrack algorithm ([DOI:10.3389/fdata.2020.601728](https://doi.org/10.3389/fdata.2020.601728)), which can be offloaded to GPUs.
- To be used as seeds, Patatrack pixel tracks are required to:
    - Be built with at least three pixel hits;
    - Have transverse momentum $p_T > 0.3$ GeV;
    - Be consistent with a leading pixel vertex;
- Pixel vertices from primary interactions are reconstructed at the HLT from pixel tracks with at least four hits and $p_T > 0.5$ GeV.
- The vertex with largest summed $\sum p^2_T$ of constituent tracks, is the primary vertex (PV).



raw data

raw data

digis

clusters

doublets

ntuplets

pixel tracks (SoA)

pixel tracks

pixel tracks (legacy)

pixel vertices

pixel vertices (SoA)

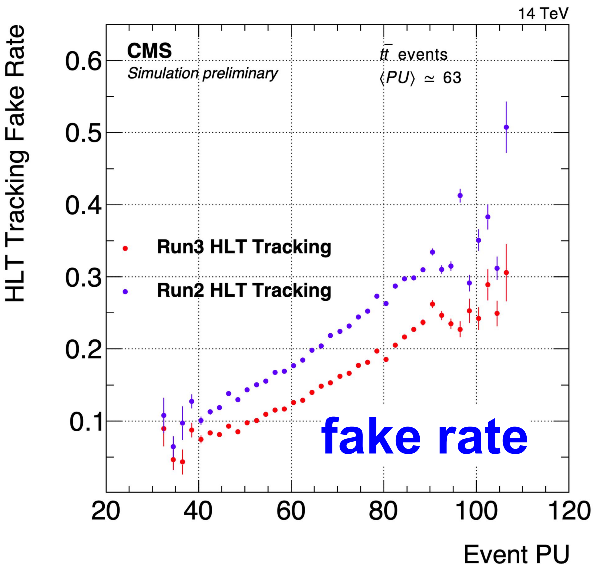
pixel vertices (legacy)

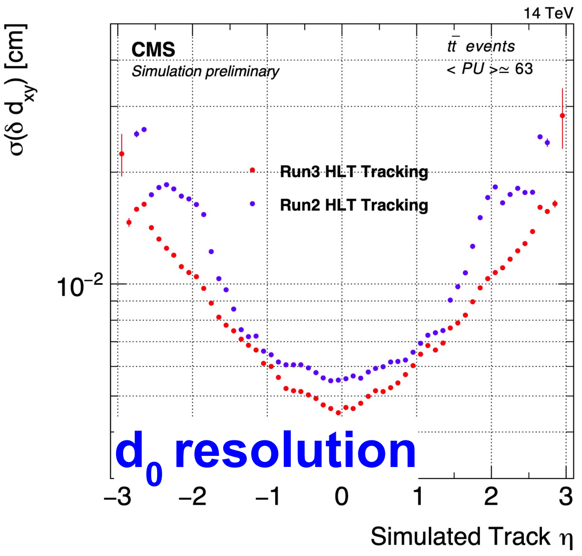
CPU

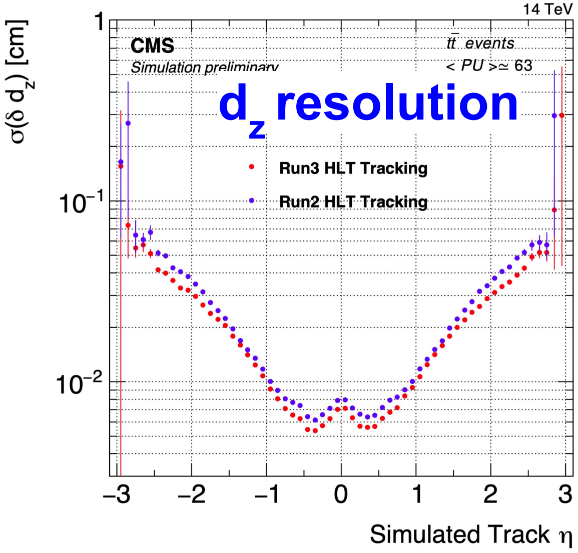
GPU

CMS TRK at HLT Performance

- Performance in simulation is documented in CMS-DP-2022-014
- The performance has been measured in a simulated ttbar sample with superimposed pileup (PU) events.
  - The number of PU events generated follows a uniform distribution from 55 to 75. The detector conditions are simulated with no module failure and taking into account the residual radiation damage due to Run-2 operations
- Some highlights below:
  - With respect to the Run 2 HLT tracking, better fake rate rejection and improved impact parameters resolutions.

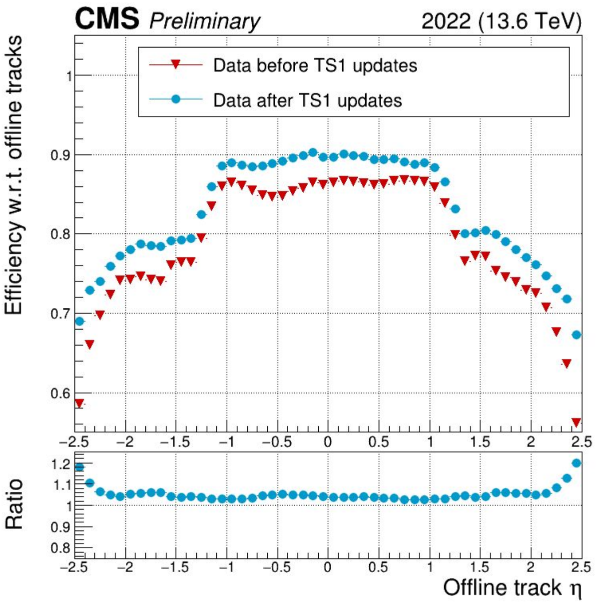


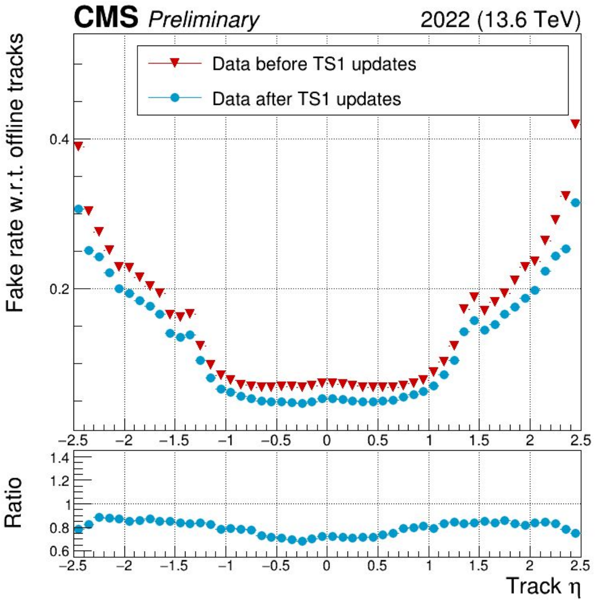


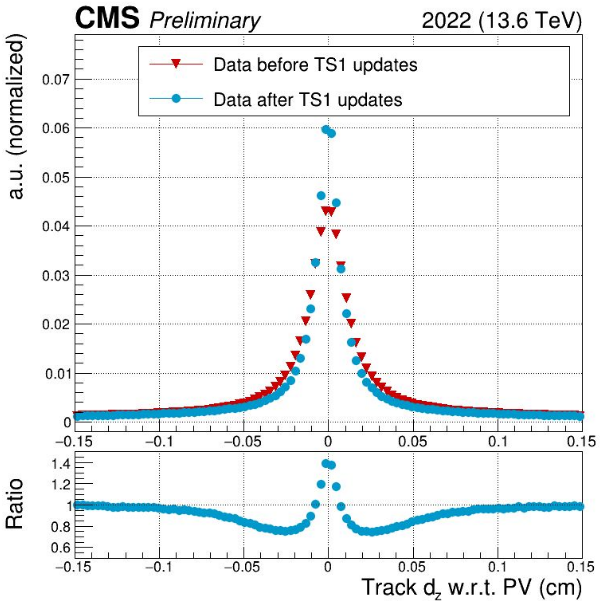


CMS TRK at HLT performance

- The performance is measured ([CMS DP-2023/028](CMS DP-2023/028)) in data recorded at √s=13.6 TeV in 2022, using runs taken shortly before and shortly after the first Technical Stop (TS1) of the LHC, when several updates in detector conditions took place:
  - ○ Increase in BPix L1 reverse bias high voltage (HV) from 150 V to 300 V, with a corresponding;
  - ○ update of the pixel cluster position estimator (CPE), as well as a new pixel detector gain calibration and a new tracker alignment.
- The HLT tracking efficiency and fake rate measured in data are defined with respect to offline tracks, i.e. tracks produced by the full offline event reconstruction, which satisfy high-purity track quality criteria.



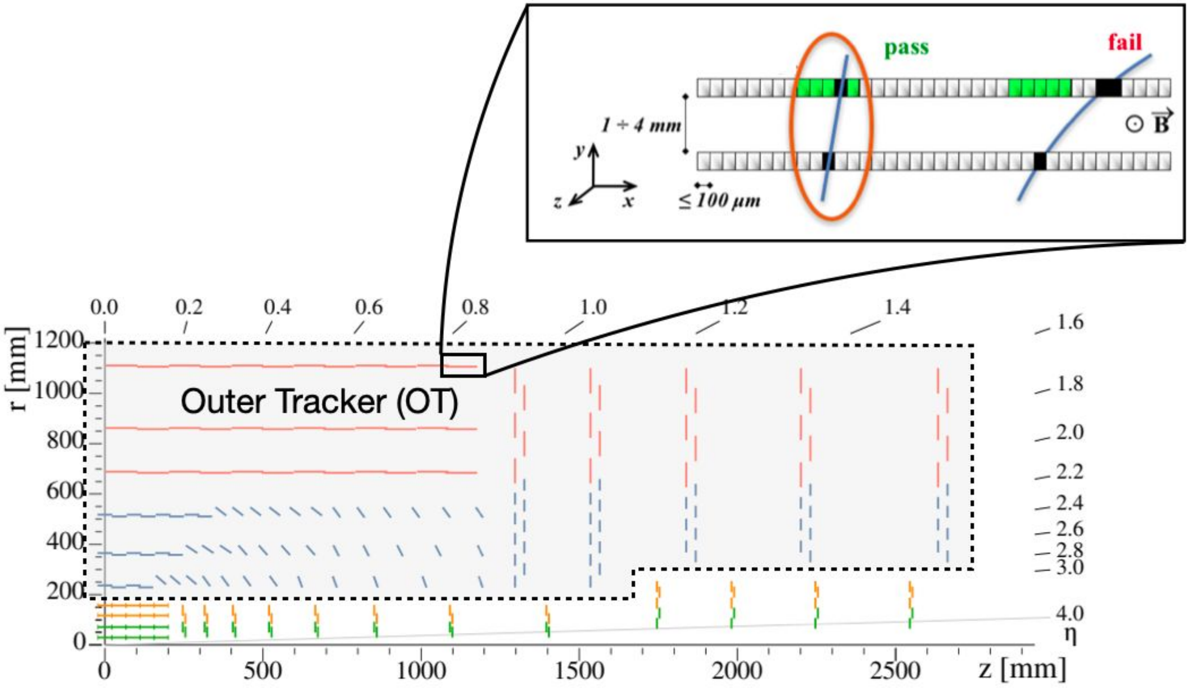




CMS Tracker Phase2 upgrade

- The CMS Phase-2 Tracker will be composed of:
    - Inner Tracker: 4 barrel layer, 12 disks
        - Extended up to η = 4.0
    - Outer Tracker: 6 barrel layer, 5 disks
        - Each module consists of two closely spaced sensors allowing for an L1 track trigger (“pT modules”)

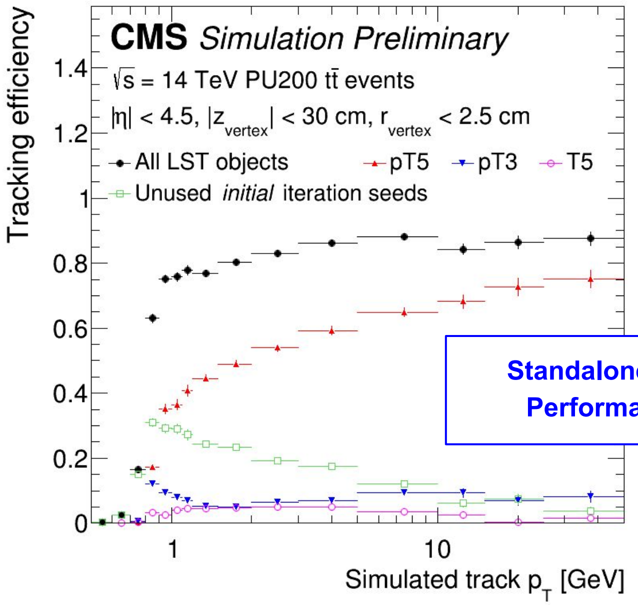
Parameters	Run 3	HL-LHC
Years	2022-2025	≥2029
√s	13.6 TeV	14 TeV
Inst. lumi	2.5 10^{34} Hz/cm^2	7.5 10^{34} Hz/cm^2
Bx rate	40 MHz	40 MHz



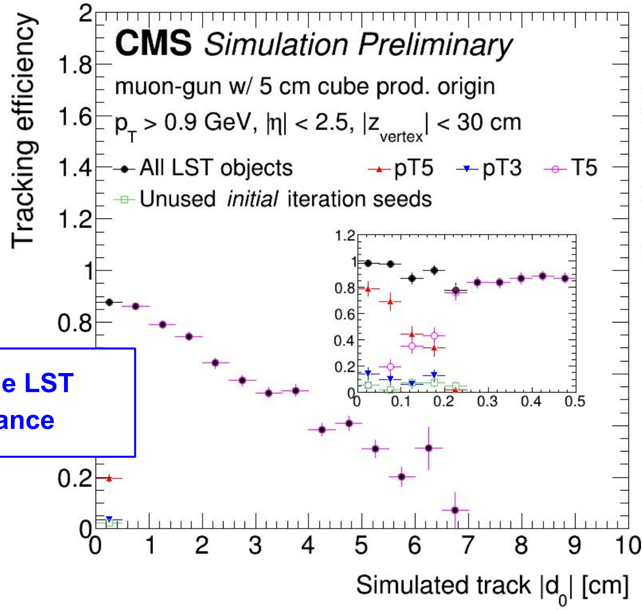
- The modules are arranged in a tilted geometry for the Barrel Layers 1,2,3.

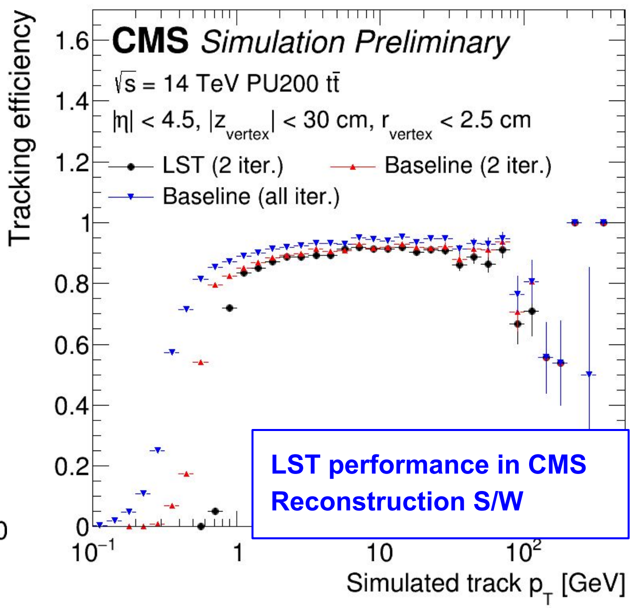
CMS: Line Segment Tracking

- The LST algorithm (CMS-DP-2023-019) creates following objects in OT through linking of objects:
    - MiniDoublet (**MD**): linked pair of hits in individual pT modules
    - Line Segments (**LS**): linked pair of MDs in neighboring layers
    - Triplet (**T3**): linked pair of LSs with a common MD
    - Quintuplet (**T5**): linked pair of T3s with a common MD
- Using a subset of inner tracker (IT) pixel seed iterations, (i.e. initial iteration seed, and `highPtTriplet` iteration seed), the LST algorithm creates following objects through linking of OT objects with IT seeds:
    - pixel + Quintuplet (**pT5**): linked pair of a pixel seed and a T5
    - pixel + Triplet (**pT3**): linked pair of a pixel seed and a T3 (both not in a pT5)



Standalone LST Performance





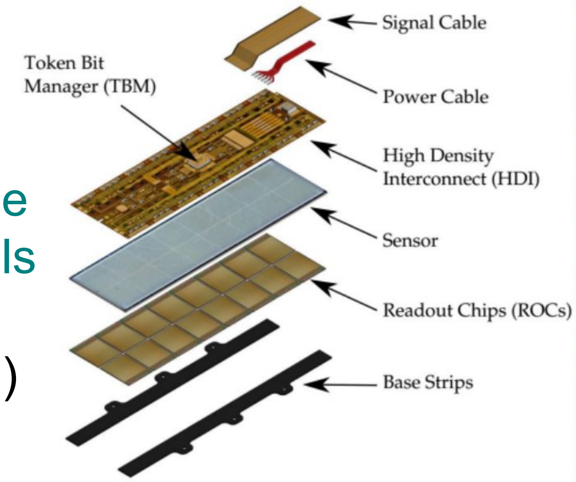
LST performance in CMS Reconstruction S/W

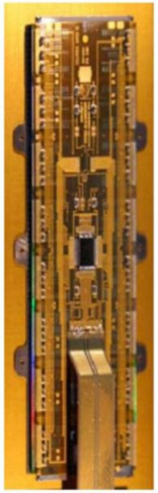
CMS Outlook in Run3/Phase2

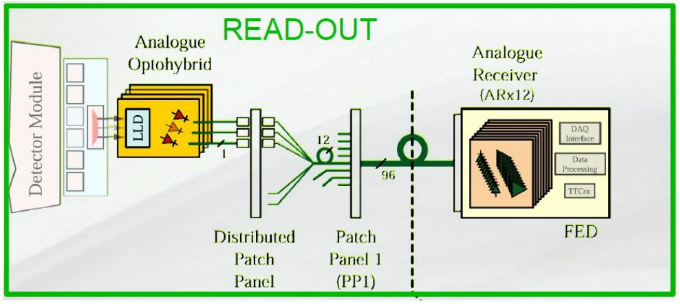
- Tracking algorithms need to provide high-quality tracks efficiently and with an efficient usage of resources:
  - high tracking and vertexing performance in Run 3 (despite challenging conditions at the LHC);
- In order to provide more precise and accurate track reconstruction sophisticated algorithms, techniques and calibrations have been developed.
- Run 3 developments include:
  - Speed-up in the track building (`mkFit`);
  - Improve the track selection algorithm (DNN);
  - improvements at tracking at trigger level (on GPUs);
  - Monitoring of Data performance vs MC as well as online reconstruction vs offline reconstruction.
- The HL-LHC will provide unprecedented challenges in terms of track and vertex reconstruction
    - this open up a rich playground for future developments in both hardware and machine learning based tracking.
    - Two promising developments have been shown

CMS Silicon Tracker

- Silicon Pixel modules (Phase-1 detector):
  - 100x150x280 μm^3 n-in-n pixel cells used everywhere in the detector;
  - Readout Chip (ROC): 250nm CMS ASIC pulse height read-out, reads matrices of 52x80 pixels
  - Two chips employed:
    - **PSI46dig** (same architecture as Phase 0) digital readout and double column drain;
    - **PROC600** (dedicated for BPix Layer 1) dynamic cluster drain;
- Silicon Strip modules:
  - 320 μm Si in inner layers (TIB, TID and inner TEC rings 1-4);
  - 500 μm Si in outer layers (TOB, TEC ring 5-7) → two silicon wafers daisy-chained.
  - Analog readout with **APV25** chip.
    - Each chip reads out 128 channels.
    - Tracker module have 4 or 6 APV chips.
    - Signal from 2 chips multiplexed to a Laser Driver.







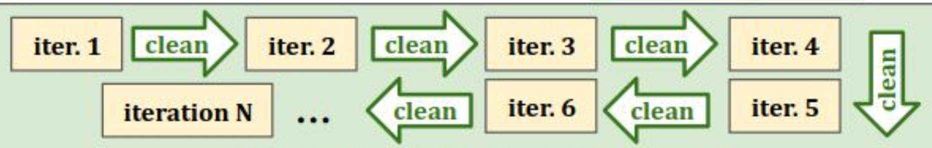
CMS: Prepartion for Run 3

- New Pixel Layer 1 installed in already in 2021:
  - Able to be operated up to 800 V compared to 600 V during Run 2
  - Enhanced front-end ASICs to improve efficiency and increase resistance against single-event upsets;
- Degradation of performance due to irradiation is expected nonetheless:
  - Especially in BPix Layer 1 due to its proximity to the LHC luminous region (29mm from the beam line);
  - Degradation visible in Pixel Hit Efficiency and Strip Signal-to-Noise ratio;
  - Effects of radiation are closely monitored, and measures are taken to mitigate the degradation;
- Routine bias voltage scans and increase of bias voltage when needed, along with routine calibrations for Pixel:
  - Adjusting temperature and bias voltage of the Strips to mitigate leakage currents;
  - Beneficial annealing during no-beam periods help improve performance;
- Improvements in online automated alignment procedure from 36 (low granularity) to ~5k parameters (high granularity) prompt calibration loop.

CMS Resources

- Performance of Run-3 HLT Track Reconstruction (CMS DP-2022/014)
- Performance of Run 3 track reconstruction with the mkFit algorithm (CMS DP-2022/018)
- Primary Vertex Reconstruction for Heterogeneous Architecture at CMS (CMS DP-2022/052)
- Early Run-3 data/MC comparison to study CMS Tracking Performance (CMS DP-2022/064)
- Performance of the track selection DNN in Run 3 (CMS DP-2023/009)
- Performance of Line Segment Tracking algorithm at HL-LHC (CMS DP-2023/019)
- Performance of Track Reconstruction at the CMS High-Level Trigger in 2022 data (CMS DP-2023/028)
- CMS Pixel Detector Performance in 2022: CMS-DP-2022-067
- CMS Silicon Strip Tracker Performance Results in 2022: CMS-DP-2023-030
- CMS Tracker Alignment Performance in 2022: CMS DP-2022/044, CMS-DP-2022/070
- CMS Pixel Detector Performance in 2023: CMS DP-2023/041
- CMS Silicon Strip Tracker Performance Results in early 2023:
- CMS Tracker Alignment Performance in 2023: CMS DP-2023/039

CMS Offline Tracking



10 iterations (+ 2 for μ-chambers)

① **Seeding**: starting from the inner part of tracker, despite the larger track density
- pixel granularity (66M in 1m²)→ 10-100x lower occupancy than outer strip layers
- 3D points from pixels and/or at least two mono-stereo layers in the Silicon Strip Tracker(double-side strips → *matched hits*)
- higher efficiency, also to ease low-pt track reconstruction

② **Track finding / pattern recognition**
- Outward KF + further inward search (add seeding hits; recover *rφ* regions excluded using matched hits to reduce seeding combinations)
- cleaner/filter (in each iteration) using shared hits and quality requirements (it.1, 2: remove tracks with # sh. clusters > 19%)

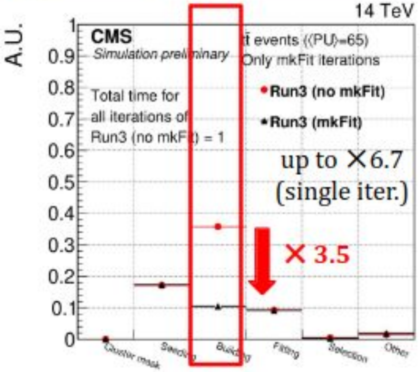
③ **Track fitting**
- Outward KF initialized at the innermost hit
- **Smoother**: second filter initialized to the result of the 1st one
- Final track parameters: weighted average

④ **Track selection**: quality selections to reduce fake tracks

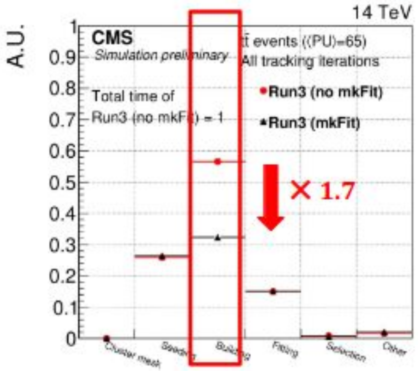
RUN 3

mkFit

Parallelized/vectorized seeding and track finding
- minimized branching points
  - TRKFIND parallelized in multiple levels (different events, η, z-/r-/φ- sorted seeds)
- distributed workload
  - Intel® Threading Building Blocks (TBB)
- memory accesses minimized and optimized
  - MATRIPLEX: custom matrix library to optimize memory access to track candidate cov. matrices in KF
  - simplified tracker geometry → tracker details stored in 2D (r or z, φ) map

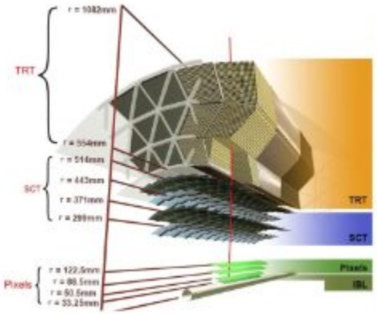


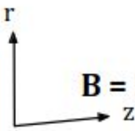
Only iterations improved by mkFit



All tracking iterations

ATLAS ID & Tracking





Inner Detector (ID) tracker (|η| < 2.5)

1. **Pixel**

Silicon

   - [barrel] 3 layers + insertable B-layer (IBL)
   - [endcap] 3 disks on each side

2. **Silicon SemiConductor Tracker (SCT)**

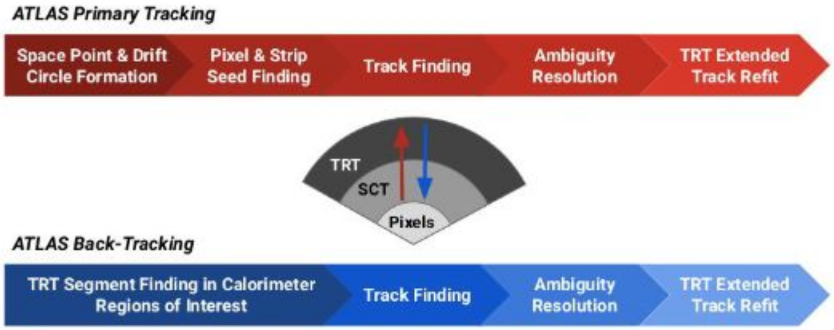
Silicon

   - Strip detector
   - [barrel] 4 double-strip layers
   - [endcap] 9 disks on each side

3. **Transition Radiation Tracker (TRT)**

Gas

   - Straw-tube tracker → tubes 4 mm wide
   - [barrel] $0.5 \lesssim r \lesssim 1$
   - [endcap] straw tubes ⊥ beam line within $0.8\text{ m} < |z| < 2.7\text{ m}$

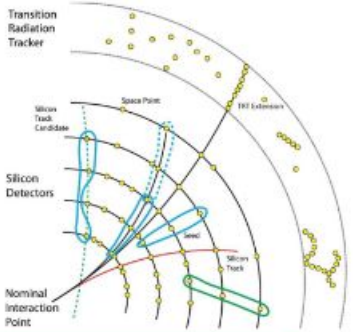


ATLAS Primary Tracking

Space Point & Drift Circle Formation → Pixel & Strip Seed Finding → Track Finding → Ambiguity Resolution → TRT Extended Track Refit

ATLAS Back-Tracking

TRT Segment Finding in Calorimeter Regions of Interest → Track Finding → Ambiguity Resolution → TRT Extended Track Refit



Primary tracking (INSIDE-out) → primaries

- Seeding: triplets in pixel + SCT
- TRKFIND: CKF to extend tracks outwards up to SCT outer layers
- Track **ambiguity solver**
  - scoring based on hit topology (holes, shared hits) and quality (χ^2, …)
  - track-quality selections (e.g. # hits ≥ 7; shared clusters/track ≤ 2)
  - neural network (NN) to minimize inefficiency due to merged clusters
- Global fitting + extension to TRT (+ re-fit)

Back-tracking (OUTSIDE-in) → secondaries, *γ*-conversions w/o silicon hits

- Seeding and pattern recognition starting from TRT
- Inward tracking → include silicon segments missed by primary tracking
- Hits assigned to tracks by INSIDE-out not considered

ATLAS Run3 optimization

Run 2: **μ = 20-40**

Run 3: **μ ~ 50**

Challenge

⬇resource consumption

≥ track quality

① **Tighter selections** for the **ambiguity solver**
→pixel + SCT hits ≥ 8 (old: 7); $|dca_{XY}|$< 5 mm (old: 10 mm)
→low-quality tracks reduced

② **Back-tracking** only for **regions** of interest with E deposit in **EM calorimeter** (E_T > 6 GeV)
→fake-tracks from TRT inward seeding 20x reduced

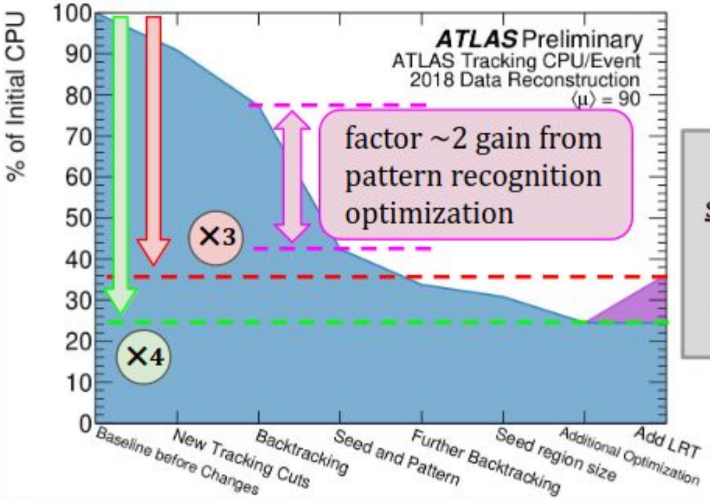
③ **INSIDE-out seeding** improved using **IBL**
→ fake-tracks reduced

④ **Restrict angular window** for **seeding** based on the lowest p_T to be reconstructed
→ combinatorial reduced ⇒ speed increased

⑤ **Additional optimizations**

- Early interruption of TRT extension w/o enough compatible hits
  → TRT extension faster (~30%), same efficiency
- New algorithm for primary vertex (PV) reconstruction

⑥ **Large Radius Tracking (LRT)**: further reconstruction pass to recover non-pointing tracks from displaced decays (strangeness)



Reduction of single-thread CPU timing for tracking per bunch-crossing

Iterative PV finding (IVF)

- iterative χ^2 minimization
- tracks weighting based on 3D χ^2 between current PV position and track DCA
- track association to one vertex at a time

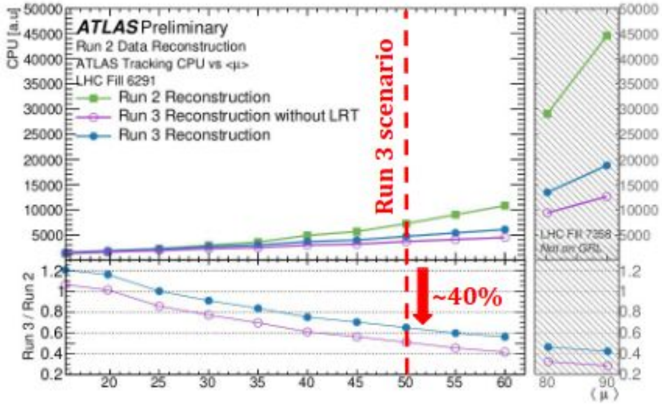
OLD

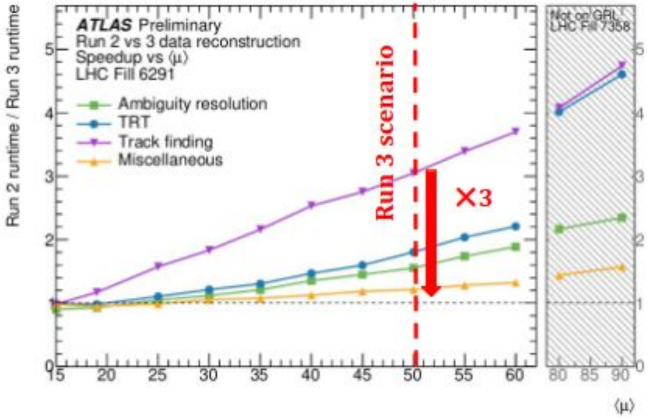
Adaptive multi-vertex fitter (AMVF)

- track weights for more than 1 vertex at a time
- convergence to 1 vertex due to deterministic annealing

NEW

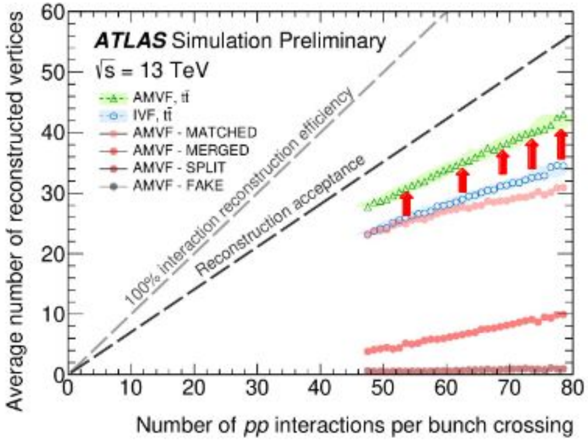
ATLAS Run3 performance





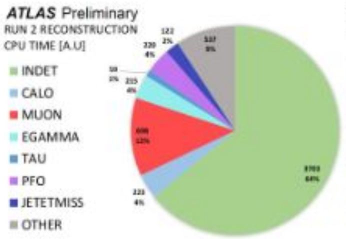
Performance on Run 2 data

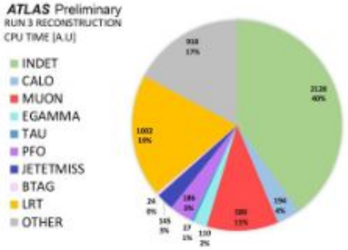
- fill 6291: 2017 data in good runlist (GRL)
  → standard data quality
- fill 7358: 2018 data not in GRL



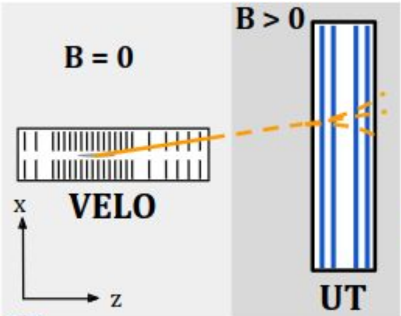
- **Stronger-than-linear scaling** vs. ⟨μ⟩ with **Run 2 reconstruction chain**
- Near **linear scaling** vs. ⟨μ⟩ with **Run 3 reconstruction chain**
  - ⟨μ⟩ ~ 50: CPU usage lower of ~40%
  - ⟨μ⟩ ~ 50: pattern recognition runtime ~3 times lower (1.5-2 others)
- **ID tracking and vertexing** only ~40% total CPU (~64% in Run 2)

- **AMVF** recovers **up to 35%** of the reconstructable primary vertices at high ⟨μ⟩, lost by the IVF





LHCb - HLT1 with Allen



Allen: A High-Level Trigger on GPU's for LHCb
- **Cheaper** and **more scalable** than CPU alternative
- Chosen as baseline of the upgrade
- Implemented with **O(200)** Nvidia RTX A5000 **GPUs**

① **VELO** → ② **UT** → ③ **SciFi** → ④ **Kalman** → ⑤ **Trigger line**

Muon detector

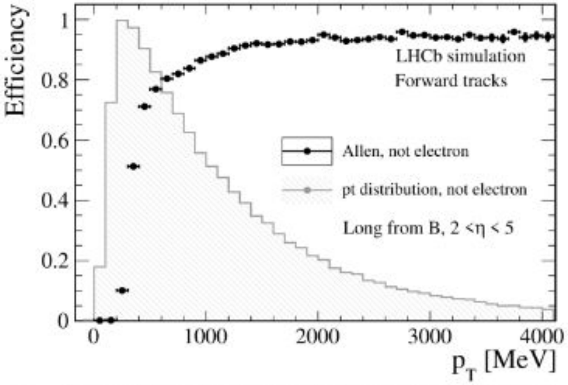
①
- **Seeds** from three hits on consecutive layers (**triples**)
- **Extension** to other layers with **linear KF**
- **PV** search with **VELO tracks**

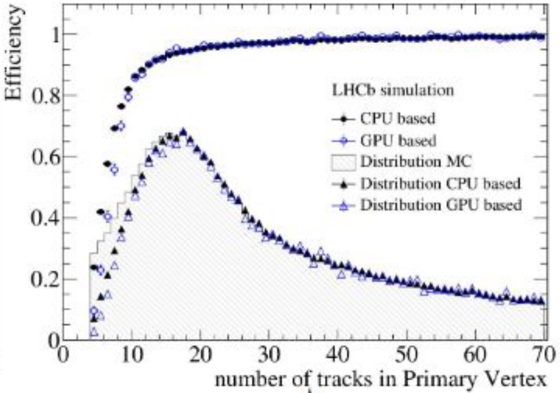
②
- **Extrapolation** of VELO tracks **to UT**
- **Parallelized tracklet finding** (# UT hits ≥ 3)
- **Momentum** estimate from **bending**

③
- VELO+UT tracks extrapolation with last parametrization
- **Parallelized Forward algorithm**
  - Extension with triplets using all the 3 stations
  - Add layers within the search window

④
- KF to improve impact parameter resolution
- VELO-only KF in HLT1 (speedup)

⑤
- Parallel fitting of **2-track** secondary vertices (**SV**)
- **Trigger** selections on single **tracks** and/or **SV**
- **Output rate ~ 1 MHz**

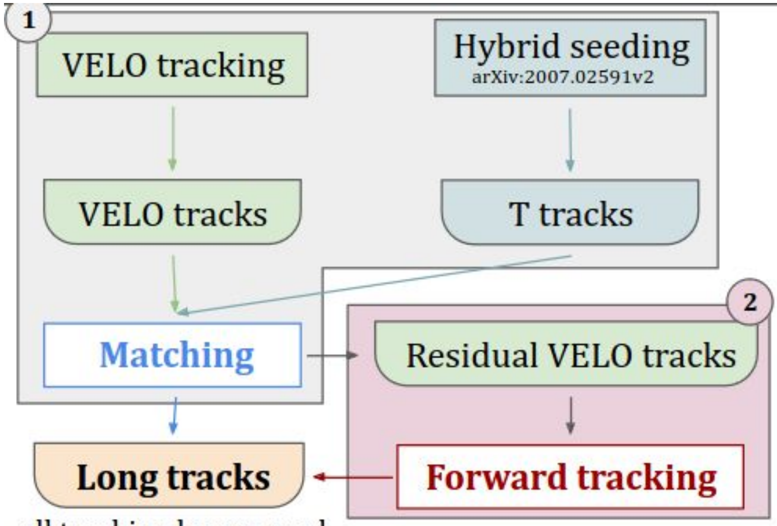




VELO

- Tracking efficiency > 90% for p_T>1 GeV/c
- PV efficiency > 90% (95%) for VELO tracks > 10 (20)

LHCb - tracking in HLT2



①

VELO tracking

Hybrid seeding
arXiv:2007.02591v2

VELO tracks

T tracks

Matching

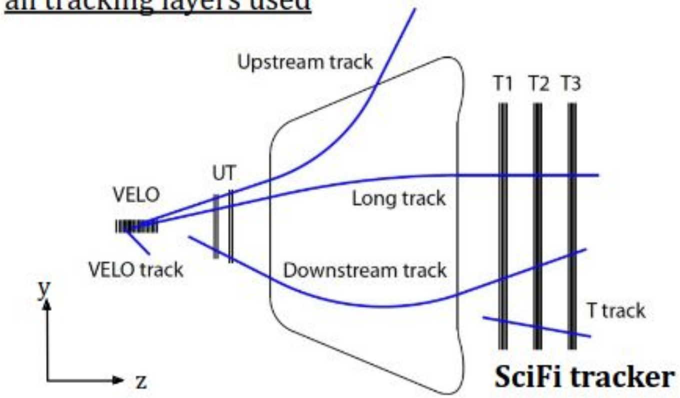
②

Residual VELO tracks

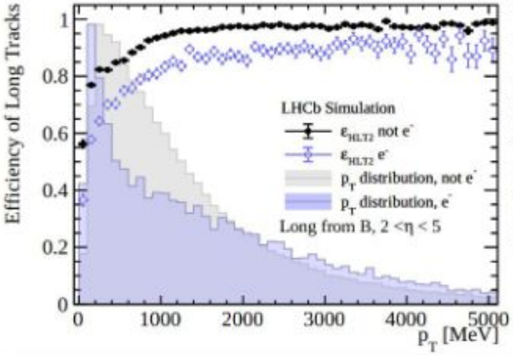
Forward tracking

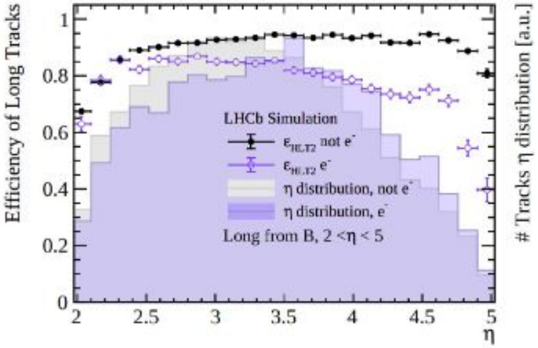
Long tracks

all tracking layers used



- **Long tracks**: best p resolution → analysis
- Two independent algorithms
  a. **Matching**: neural network trained on MC to match VELO and T tracks
  b. **Forward tracking**: VELO+UT-track extension to SciFi (# hits ≥ 10)
    - [finding] polynomial search window assuming $p > 1.5$ GeV/c
    - [finding] simplified trajectory treating the magnet as an optical lens
    - [finding] Hough-like transform to find correct SciFi hits
    - [fitting] Global KF to estimate track parameters





- Tracking efficiency for hadrons and μ←B ~ 90% (> 95% for p_T > 1 GeV/c)
- 0.1-0.2 lower efficiency for electrons
  - trajectory deflection due to bremsstrahlung
  - major effect at large η → more material

ALICE - upgrades in LS2

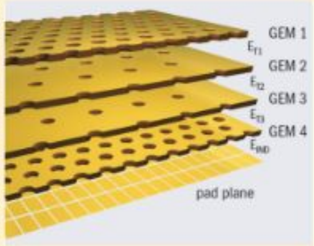
Challenges (pp, $\sqrt{s}$ = 13.6 TeV)

- Interaction rate (IR) up to 1MHz
- $\mathscr{L}_{\mathrm{int}} \sim 200\ \mathrm{pb}^{-1}$ trigger ($\sim 3\ \mathrm{pb}^{-1}$ MB)

Challenges (Pb-Pb, $\sqrt{s_{\mathrm{NN}}}$ = 5.44 TeV)

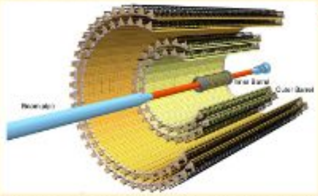
- IR~ 50 kHz
- $\mathscr{L}_{\mathrm{peak}} \sim 6 \times 10^{27}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$, $\mathscr{L}_{\mathrm{int}} \sim 13\ \mathrm{nb}^{-1}$ (Run 3,4) (×50-100 more than Run 2)

Detector upgrades (tracking only!)



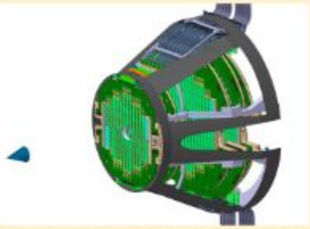
Time Projection Chamber (TPC) upgrade → $|\eta| < 0.9$

- Gas Electron Multiplier (GEM) → 152 pad rows
- No more gating grid (IR limitation ~ 3 kHz) and continuous readout
- Preserve dE/dx performance of Run 2



Inner Tracking System (ITS) upgrade → $|\eta| < 1.3$

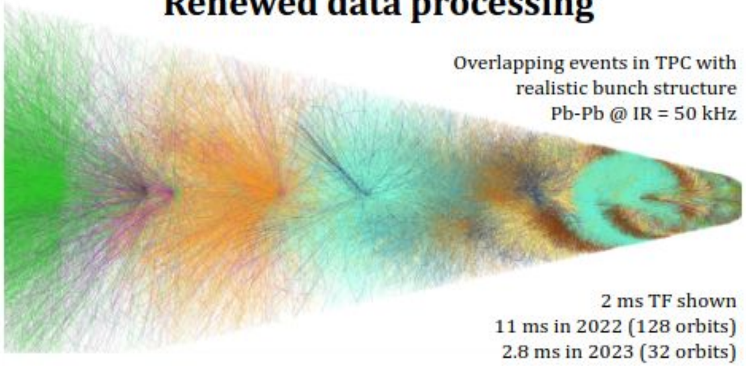
- 7 layers of Alice Pixel Detector (ALPIDE) chips: custom Monolitic Active Pixel Sensors (MAPS)
- Readout rate in Pb-Pb up to 100 kHz → 100x more than Run 2
- Material budget 0.35% X_0 (innermost layer) → 3x lower than Run 2



Muon Forward Tracker (MFT) → $2.5 < \eta < 3.6$

- 5 disks of ALPIDE chips
- Secondary vertex reconstruction at forward-η

Renewed data processing

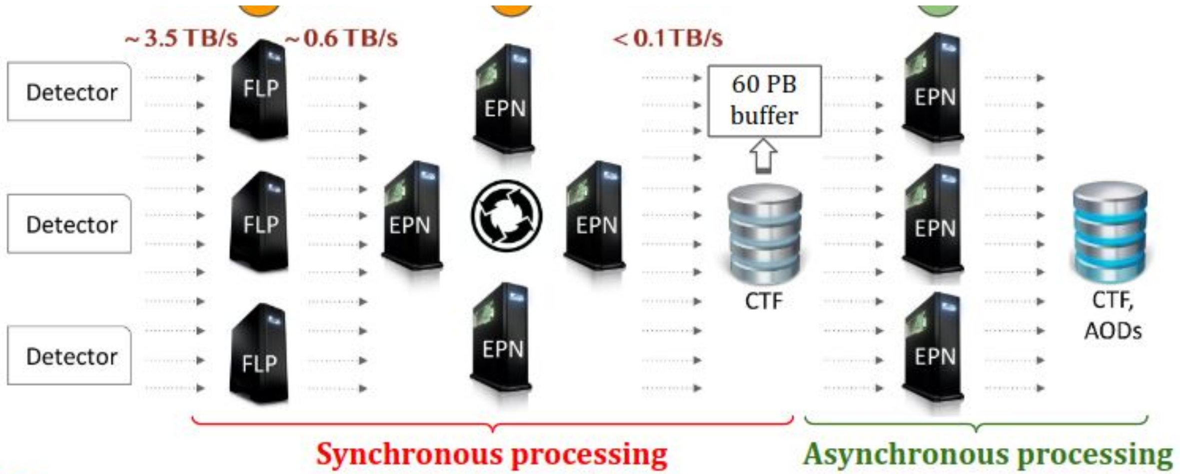


Overlapping events in TPC with realistic bunch structure
Pb-Pb @ IR = 50 kHz

2 ms TF shown
11 ms in 2022 (128 orbits)
2.8 ms in 2023 (32 orbits)

- **Continuous readout** of Time Frames (TFs)
- **Data reconstruction** developed in **synchronous + asynchronous** phases
- Software trigger infrastructure for data skimming
- **O²**: new framework for **online/offline data reconstruction** and **analysis**

ALICE: Run3 data processing



1 **First Level Processors (FLP)**

- First compression (*zero suppression*) of data from detector readout links
- Data division in sub-TFs on each FLP

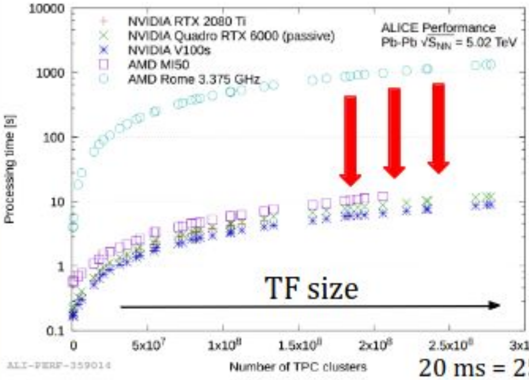
2 **Event Processing Nodes (EPN)**

- Sub-TF merge in complete TFs
    - 1 TF = 11 ms in 2022 (128 orbs), 2.8 ms in 2023 (32 orbs)
- Synchronous reconstruction, calibration, data compression
- Compressed TFs (CTFs) buffer

3 **Asynchronous processing**

- New reconstruction with final calibrations on EPN, T0 and T1
- Final Analysis Object Data (AO2Ds) produced and stored
- CTFs cancelled to free disk space

- Goal: total compression factor = **35**
- Main challenge: **TPC**
    - ~3.4 TB/s → ~ 70 GB/s (↓50x)
- **Clusterization** and **tracking** on O(2000) **GPUs** AMD MI50
    - >40x faster, only 4x more expensive



20 ms = 256 orbits

- **Up to 100x gain** with **GPUs** compared to 1-core CPU
- **Linearity** of **GPU** processing **time** vs # TPC clusters up to 256 orbits TF (backup)
- No impact of TF length on number of needed GPUs

ALICE mid-y tracking

TPC tracking

1. Tracking within a φ sector (36)
   - a. **Cellular Automaton** (CA) track **seeding**
   - b. First KF within a sector
2. Track merging among sectors
   - a. Prolongation to segments in adjacent sectors
   - b. Pick-up of further clusters
   - c. Final KF fit

ITS tracking

1. PV seeding
   - a. tracklets in 3 innermost layers
   - b. linear extrapolation of tracklets
   - c. clustering to find collision point(s)
2. Track finding and fitting
   - a. PV used to reduce combinatorics in matching the hits
   - b. CA: track segments (cells) connection into candidate tracks
   - c. KF fit of candidates ($\geq$ 4 consecutive hits)

ITS-TPC tracks

- Time-matching:
  - among ITS and TPC standalone (SA) tracks
  - between a TPC-SA track and left ITS clusters (*afterburner*) → Daughters of V0/cascades decaying in ITS
- Prolongation to TRD/TOF
- KF refit outwards and inwards
  → Async. reco.: final calibrations for position-dependent TPC distortions applied

- **Pointing resolution** to the PV of **~35-40 μm @ p_T = 1 GeV/c**
- **2x** (4-5x) **better** performance in $r\varphi$ (z) compared to Run 2
- Fine-tuning on TPC calibrations/ITS alignment ongoing to fix residual mismatch with MC