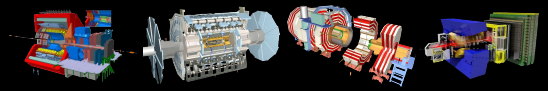
Tracking and vertexing

Mattia Faggin
University and INFN, Trieste (Italy)

On behalf of ALICE, ATLAS, CMS, LHCb
25th May 2023

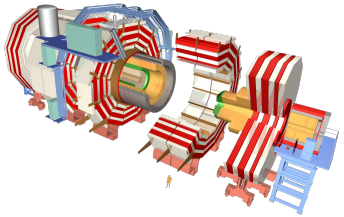
Tracking and vertexing at the LHC



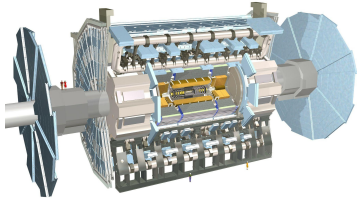
mfaggin@cern.ch

- **Tracking and vertexing**: **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)

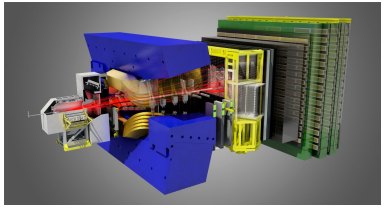


A Toroidal LHC Apparatus (ATLAS)

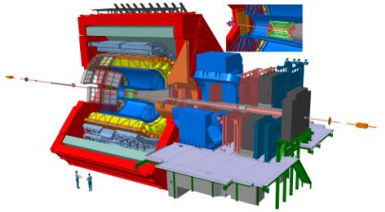


Reconstruction algorithms renewed to handle higher average in-bunch pile-up ($\langle\mu\rangle$) collisions

Large Hadron Collider beauty (LHCb)



A Large Ion Collider Experiment (ALICE)



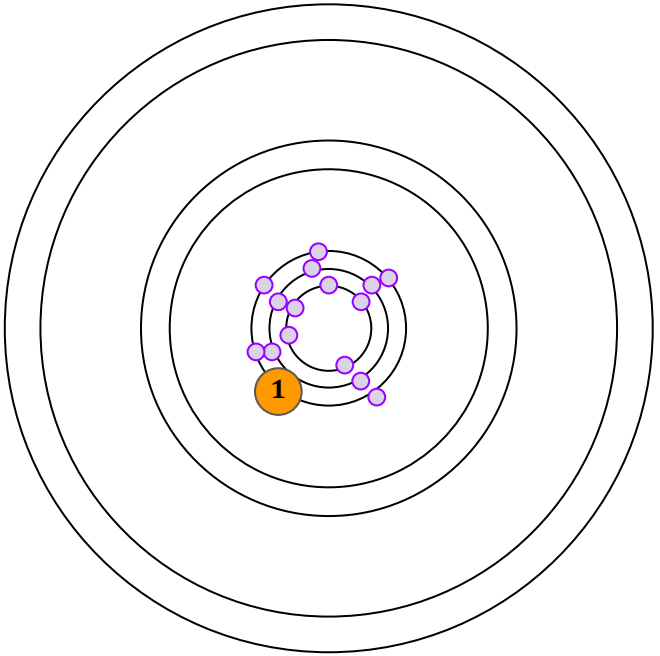
Major detector upgrades and renewed data flow to significantly increase the collected statistics in Run 3, 4

Common basic concepts of tracking

mfaggin@cern.ch

- The tracking can be summarized in these **4 main steps**:

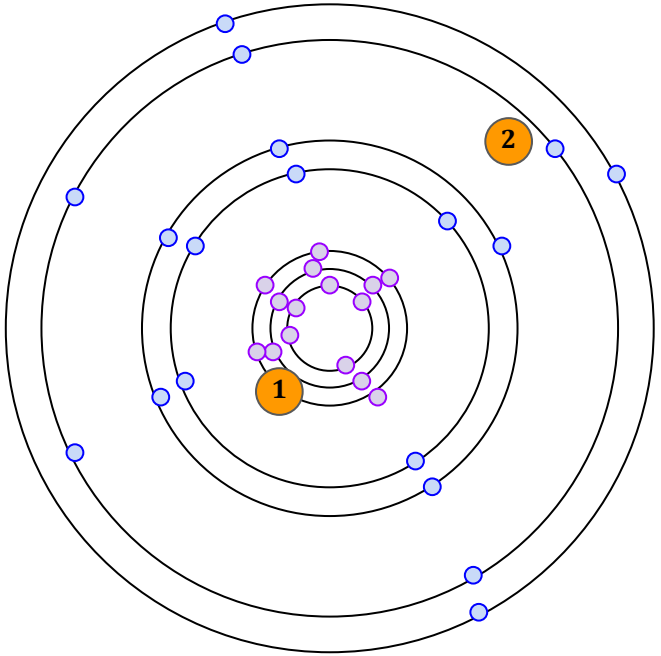
1. **Seeding**: built “short tracks” to be used as seeds for longer tracks



Common basic concepts of tracking

- The tracking can be summarized in these **4 main steps**:

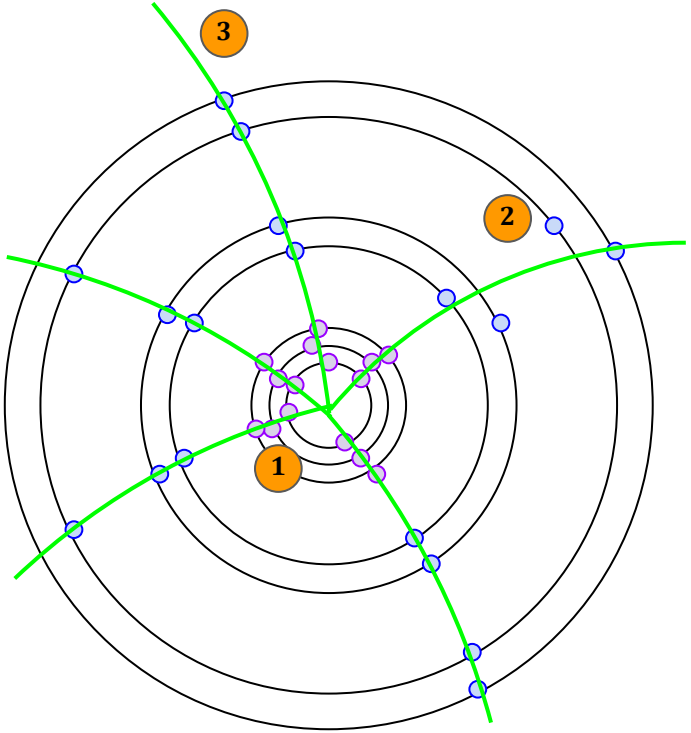
1. **Seeding**: built “short tracks” to be used as seeds for longer tracks
2. **Track finding / pattern recognition**: search for additional hits to prolong track seeds to other tracking layers



Common basic concepts of tracking

- The tracking can be summarized in these **4 main steps**:

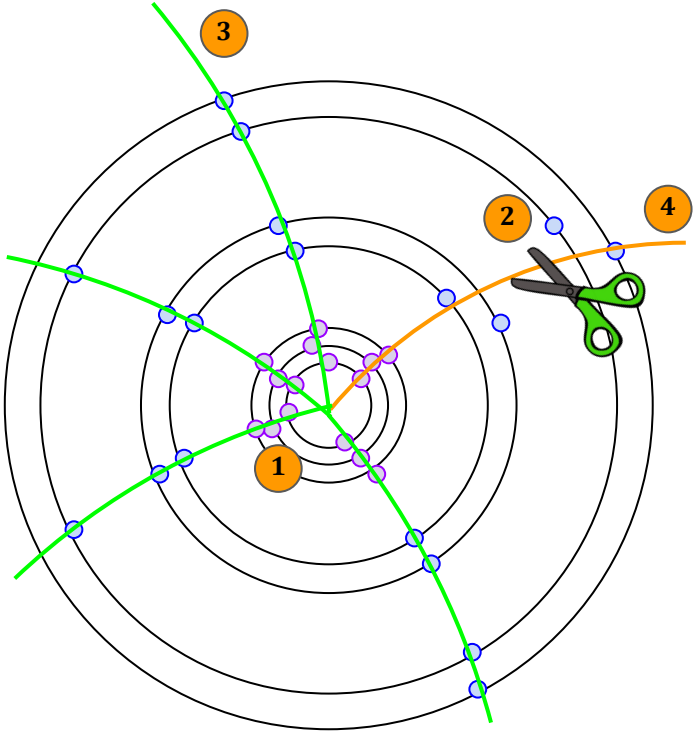
1. **Seeding**: built “short tracks” to be used as seeds for longer tracks
2. **Track finding / pattern recognition**: search for additional hits to prolong track seeds to other tracking layers
3. **Track fitting**: use the points found during the track finding to calculate the track parameters and covariance matrix



Common basic concepts of tracking

- The tracking can be summarized in these **4 main steps**:

1. **Seeding**: built “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 tracks

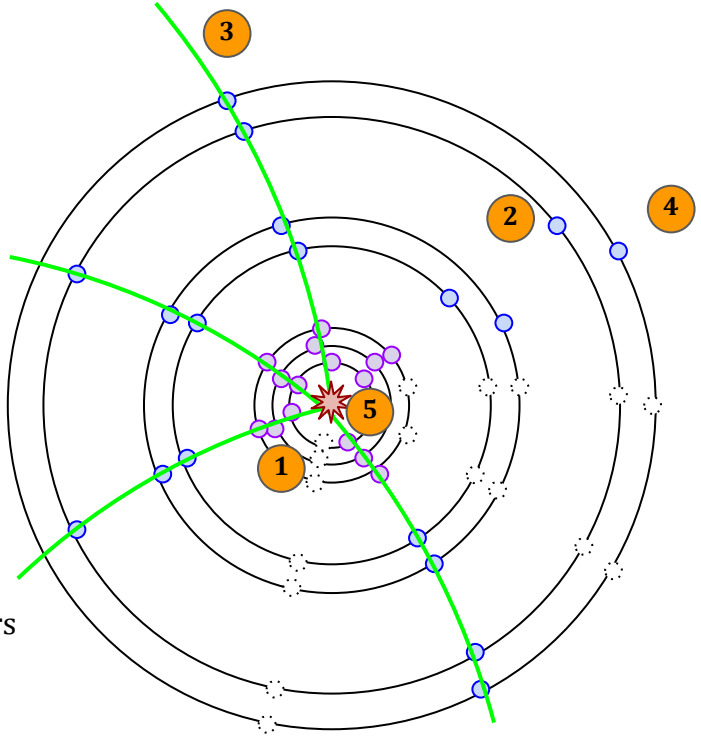


Common basic concepts of tracking

- The tracking can be summarized in these **4 main steps**:

1. **Seeding**: built "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 tracks
5. **Primary vertexing**: find the beam collision point(s) as the minimum-distance point among tracks

Iterative tracking:
- remove assigned clusters
- restart using unused clusters



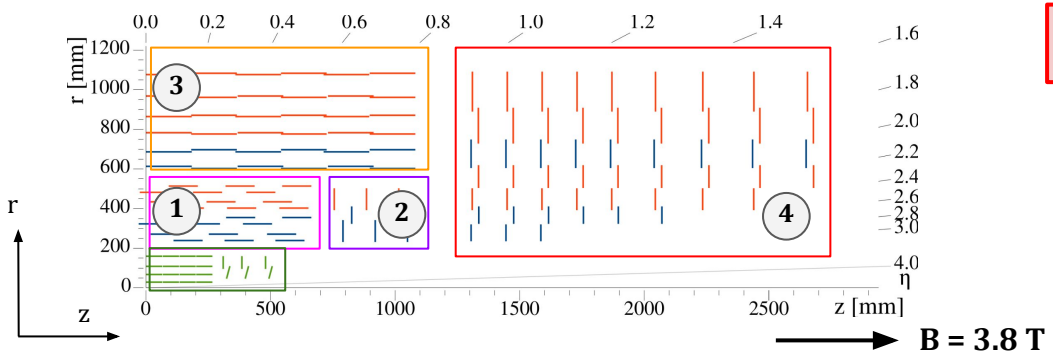
Disclaimers
- Talk focused on charged-particle reconstruction
  → **no** explicit references to **muon-chamber tracking**

- **Different experiments** implement the procedure in a **different way** (different detectors, algorithms, …)
- This presentation: **summary** of tracking and vertexing in **ALICE, ATLAS, CMS and LHCb**

CMS - Silicon Tracker and tracking

2021 JINST 16 P02027
2008 JINST 3 S08004
2014 JINST 9 P10009
arXiv:2304.05853v1, CMS-DP-2022-018

mfaggin@cern.ch



Hermetic tracking system within $|\eta| < 3$

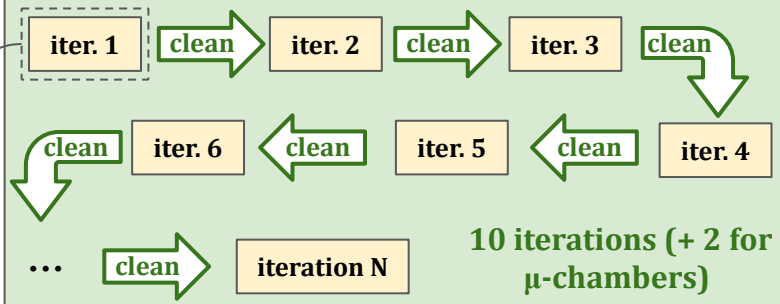
- **CMS Phase-1 pixel detector**
- **Silicon Strip Tracker**
  1. Tracker Inner Barrel (**TIB**)
  2. Tracker Inner Disks (**TID+-**)
  3. Tracker Outer Barrel (**TOB**)
  4. Tracker EndCaps (**TEC+-**)

more in backup
"CMS - Silicon Tracker and tracking"
"CMS - offline tracking"

1. **Seeding**: 3D points from pixels and/or at least two mono-stereo layers in the Silicon Strip Tracker
2. **Track finding / pattern recognition**
   - Outward KF + further inward search of further hits
   - cleaner/filter (in each iteration) using shared hits and quality requirements
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
   - Iteratively repeat the above to reject outlier hits
4. **Track selection**: quality selections to reduce fake tracks
   - DNN-based since Run 3 (CMS-DP-2023-009)

Combinatorial Track Finder (CTF)

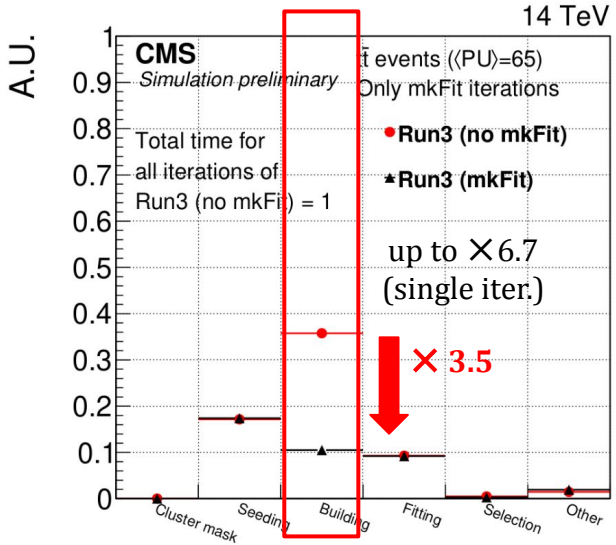
- **Combinatorial Kalman Filter (CKF)**: pattern recognition + track fitting
- **Iterative tracking** →different track categories in each iteration



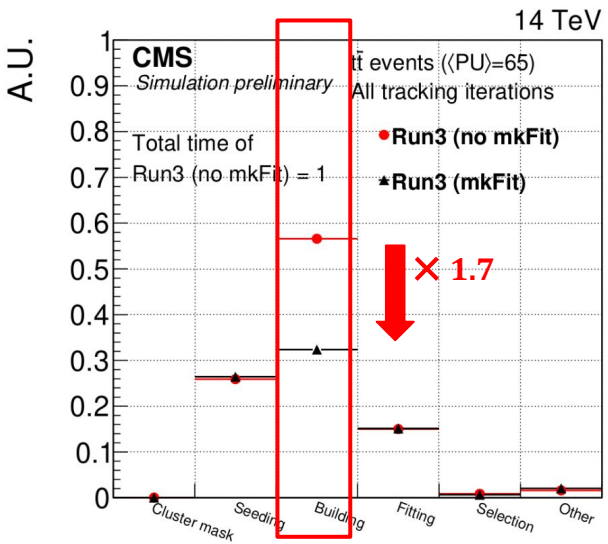
Pattern recognition optimization in Run 3:

MATRIPLEX Kalman-fitter algorithm (mkFit)

- **Parallelized** and **vectorized CKF**
- MATRIPLEX: custom library to optimize memory access to track cov. matrices in KF
- **Similar** physics **performance** as Run 2 CKF
- Significant **speed up** (simplified tracker geometry)
- Used by a subset of tracking iterations reconstructing ~90% hard-scattering event tracks



Only iterations improved by mkFit



All tracking iterations

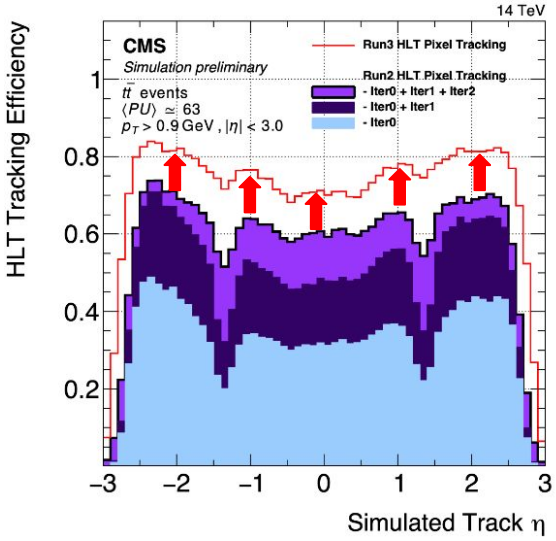
more in backup
"CMS - offline tracking"

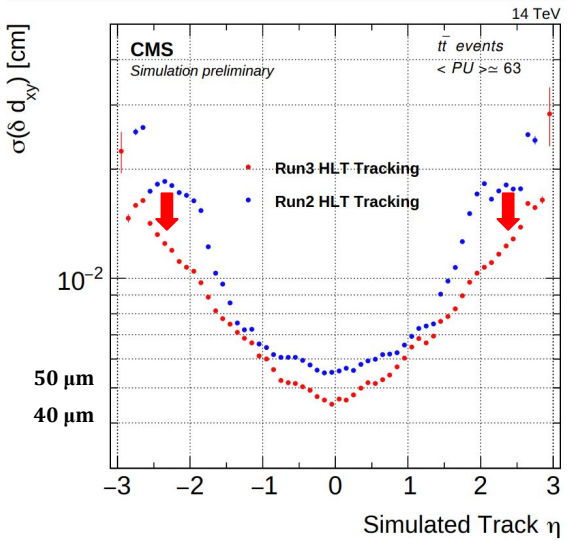
- High-Level Trigger (**HLT**): streamlined version of the offline reconstruction software on a farm for large reduction in data rate
- HLT track seeding and vertexing based on pixel detector only

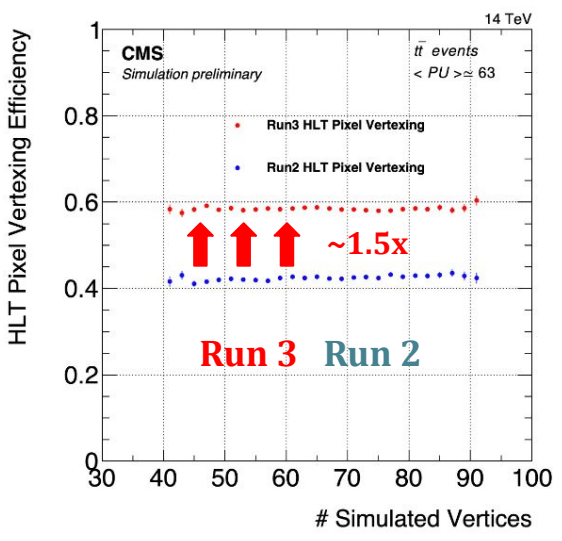
more in backup
"CMS - HLT tracking and vertexing"

PATATRACK

- **HLT pixel tracking** ported to **GPUs** → heterogeneous computing with CUDA (Nickolls et al., 2008)
- Better physics performance and throughput





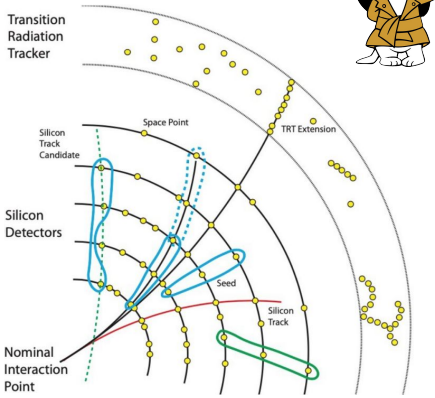


ATLAS - ID and tracking

Eur. Phys. J. C (2017) 77:673
ATL-PHYS-PUB-2021-012

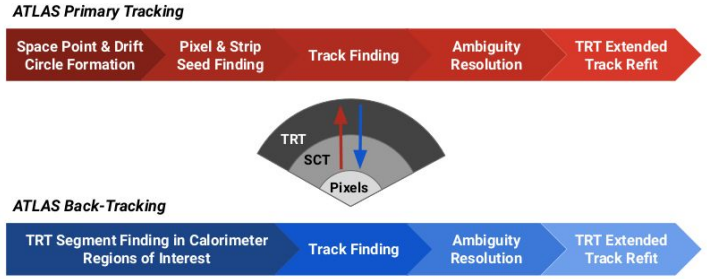
mfaggin@cern.ch

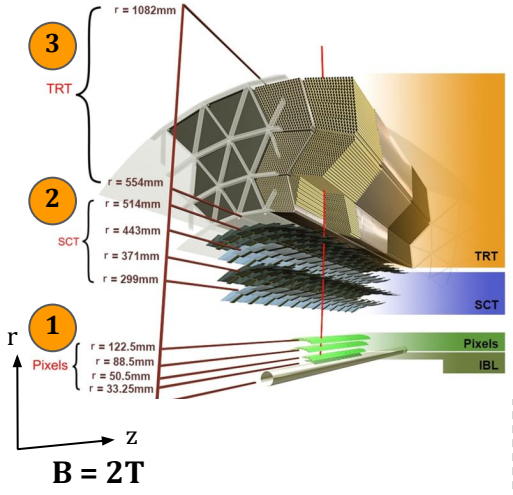
more in backup
"ID and tracking"



Inner Detector (ID) tracker ($|\eta| < 2.5$)

1. **Pixels**
2. **Silicon SemiConductor Tracker (SCT)**
3. **Transition Radiation Tracker (TRT)**





Primary tracking (INSIDE-out) → <u>primaries</u>

- 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, <u>γ-conversions w/o silicon hits</u>

- 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 - Run 3 optimization

mfaggin@cern.ch

Run 2: **μ = 20-40**

Run 3: **μ ~ 50**

Challenge

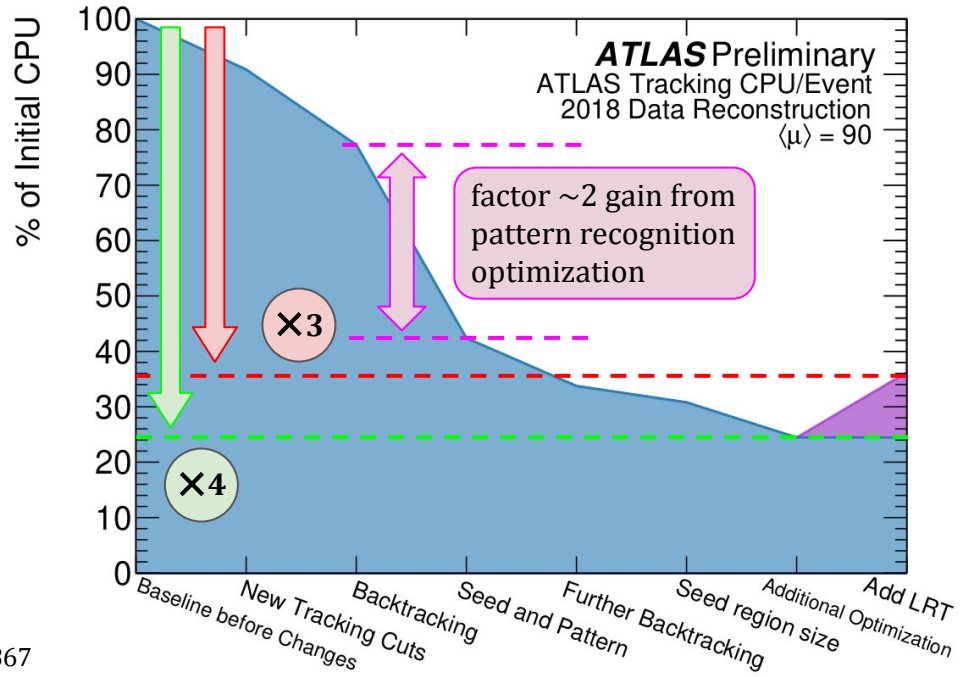
⬇️ resource consumption
≥ track quality

Several improvements for Run 3:

1. **Tighter selections** for the **ambiguity solver**
2. **More stringent** conditions for track **seeding** and **track finding**
3. **New** primary vertex (**PV**) reconstruction **algorithm**: **A**daptive **m**ulti-**v**ertex **f**itter (**AMVF**)

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

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



more in backup
"ATLAS - Run 3 optimization"

arXiv:2304.12867

Run 3

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

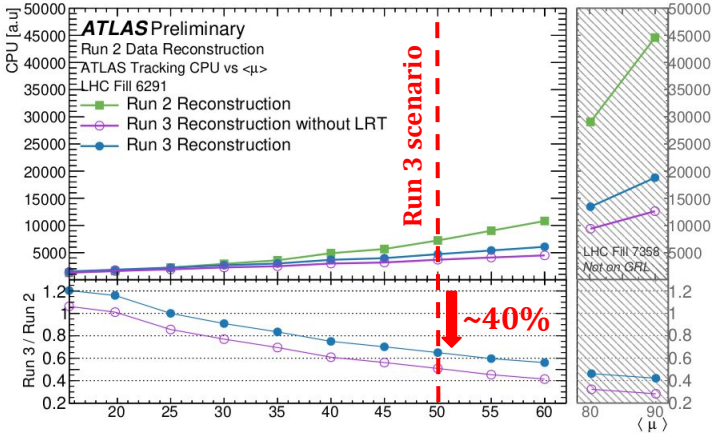
- run only on ~10% Run 2 data; enabled in Run 3 reducing fake-track fraction

ATLAS - Run 3 performance

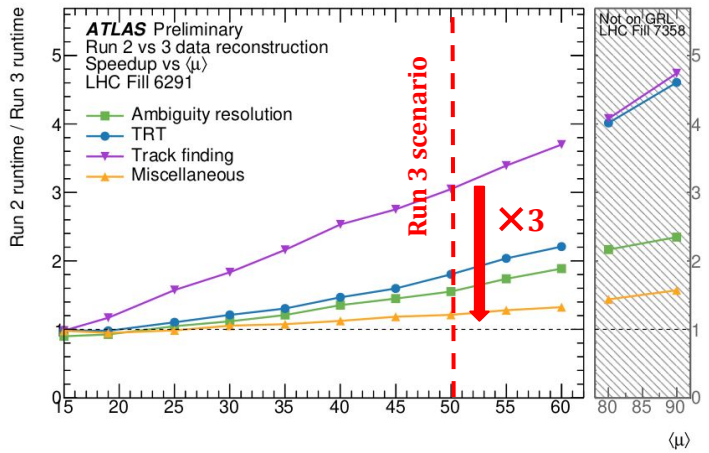
ATL-PHYS-PUB-2021-012
ATL-PHYS-PUB-2019-015

mfaggin@cern.ch

Tracking

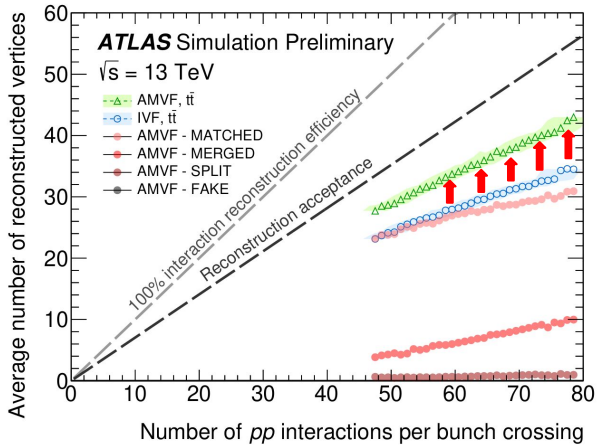


Tracking



- 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 ⟨μ⟩, lost by the Run 2 algorithm (Iterative Vertex Finding)

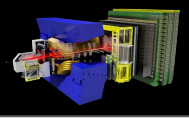
Primary vertexing



more in backup *"ATLAS - Run 3 performance"*

LHCb - upgrades in LS2

2022 JINST 17 C01046
LHCb TDR 015
Comput Softw Big Sci 4, 7 (2020)

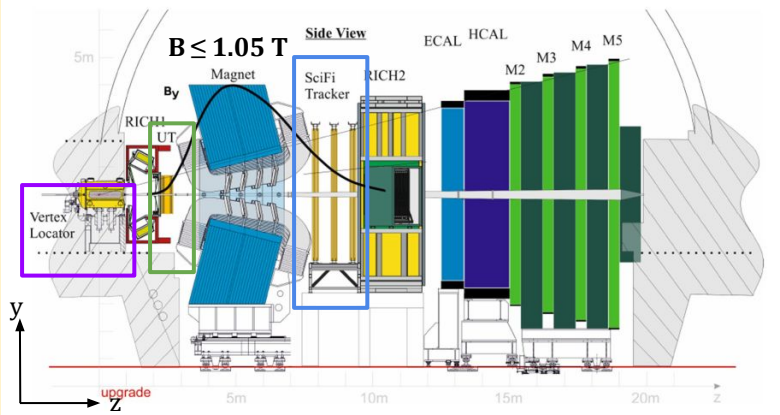


mfaggin@cern.ch

Challenges

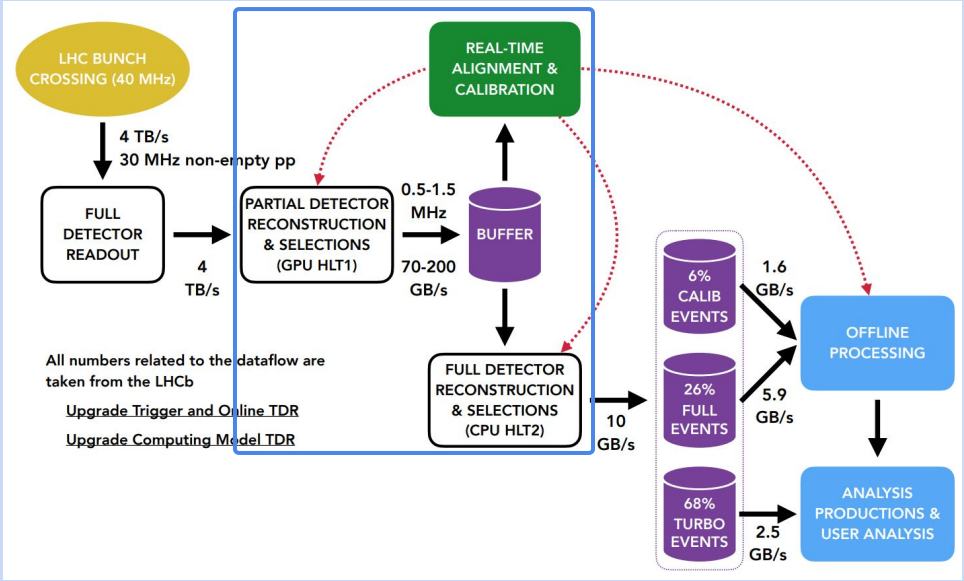
- bunch-crossing (BC) rate up to 40 MHz
- pile-up: <μ> ~1.4 → <μ> ~5

Detector upgrades (tracking only!)



- **Vertex Locator (VELO)** $2 < \eta < 5$
  - [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

Renewed data flow



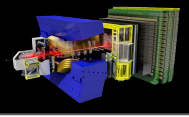
Level-0 hardware trigger (~1 MHz) → software trigger to be (~30 MHz non-empty pp collisions)

1. GPU **High-Level Trigger 1** (**HLT1**)

   — Real-time alignment and calibrations —

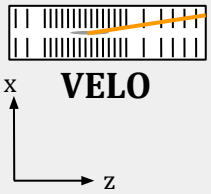
2. CPU **High-Level Trigger 2** (**HLT2**)

LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)



mfaggin@cern.ch

B = 0



1 **VELO**

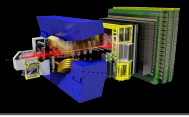
Allen: A High-Level Trigger on GPU's for LHCb

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

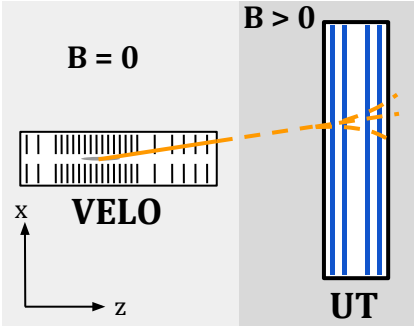
1

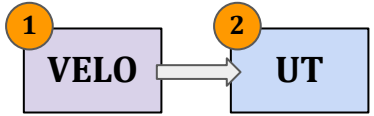
- **Seeds** from three hits on consecutive layers (**triples**)
- **Extension** to other layers with **linear KF**

LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)



mfaggin@cern.ch





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

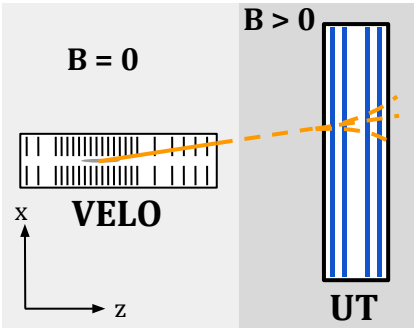
- **Cheaper** and **more scalable** than CPU alternative
- Chosen as baseline of the upgrade
- Implemented with **O(200)** Nvidia RTX A5000 **GPUs**

1

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

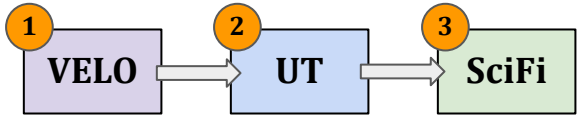
2

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



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

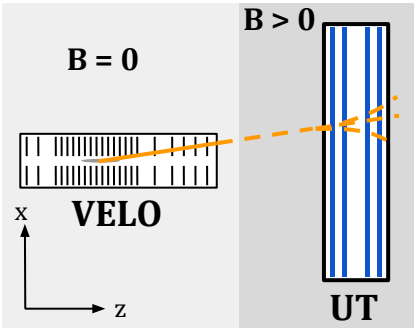
- **Cheaper** and **more scalable** than CPU alternative
- Chosen as baseline of the upgrade
- Implemented with **O(200)** Nvidia RTX A5000 **GPUs**



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

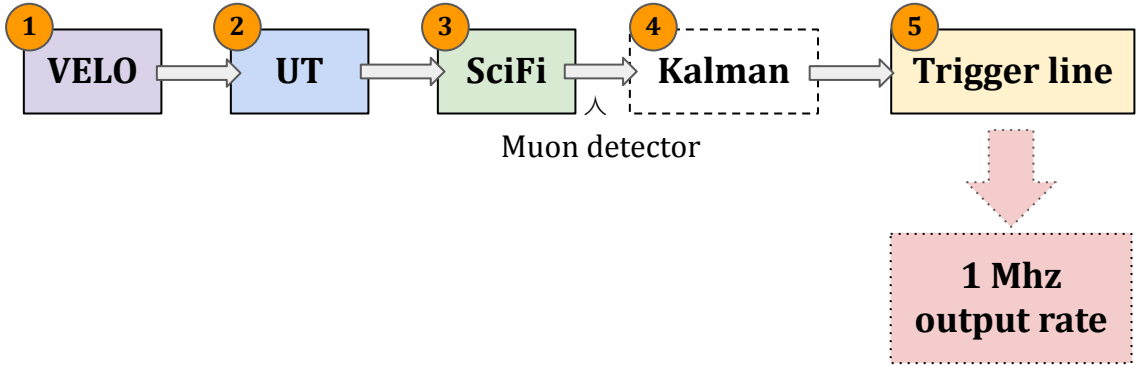
2
- **Extrapolation** of VELO tracks **to UT**
- **Momentum** estimate from **bending**

3
- VELO+UT tracks extrapolation to SciFi tracker



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

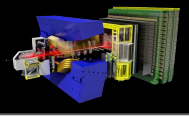
2
- **Extrapolation** of VELO tracks **to UT**
- **Momentum** estimate from **bending**

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-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)

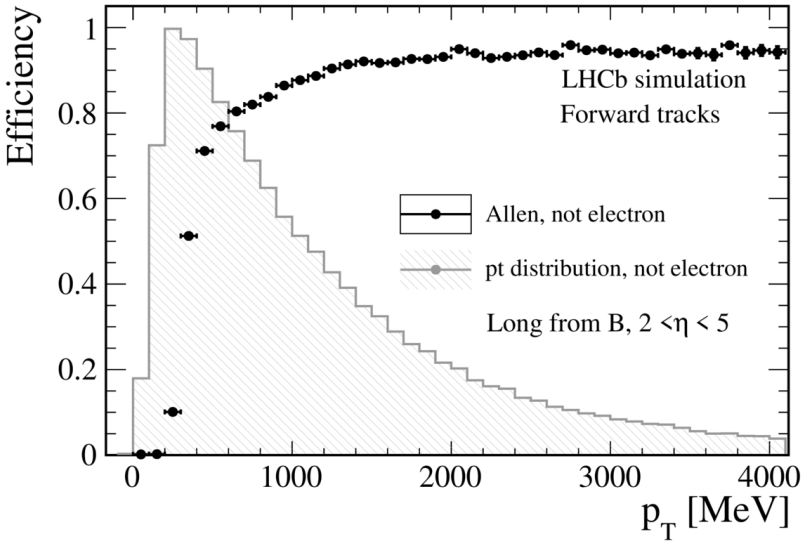


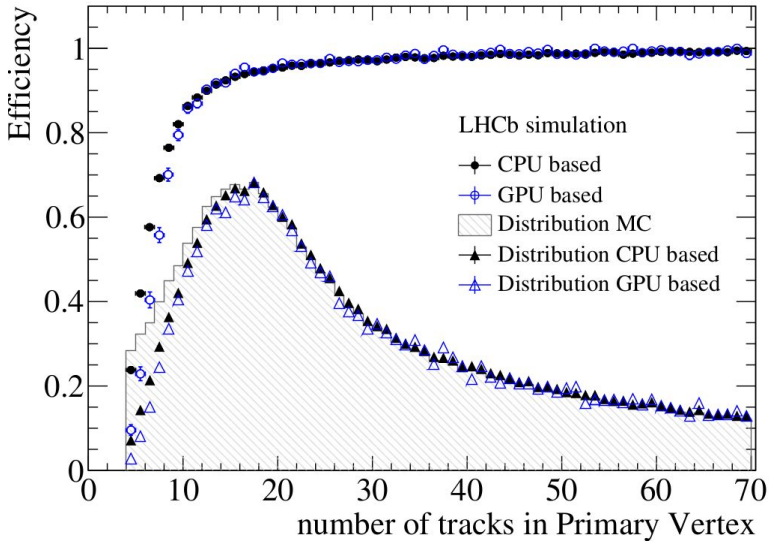
mfaggin@cern.ch

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**

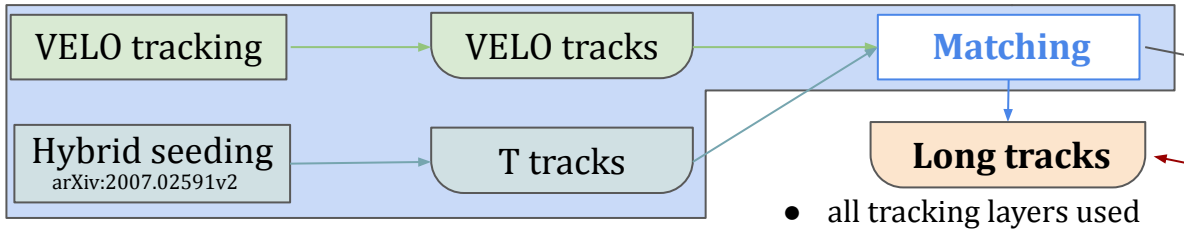
more in backup
"LHCb - HLT1 with Allen"



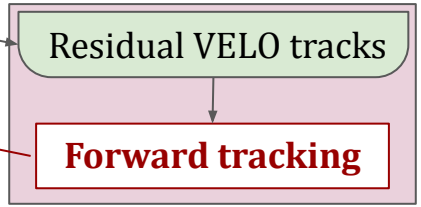


- **Tracking efficiency > 90%** for p_T>1 GeV/c
- **PV efficiency > 90%** (95%) for **VELO tracks > 10** (20)

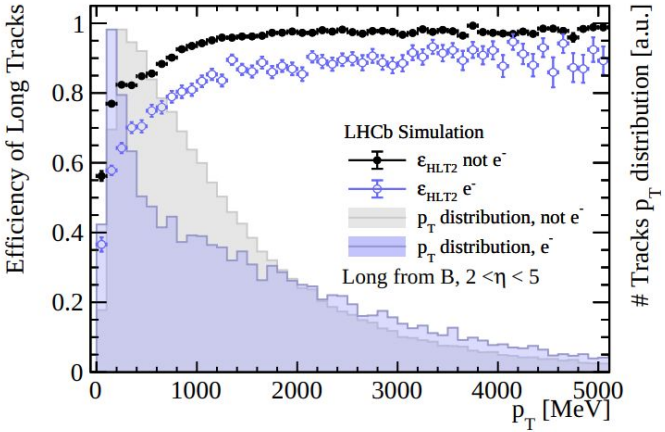
1 **Matching**: neural network trained on MC to match VELO and T tracks

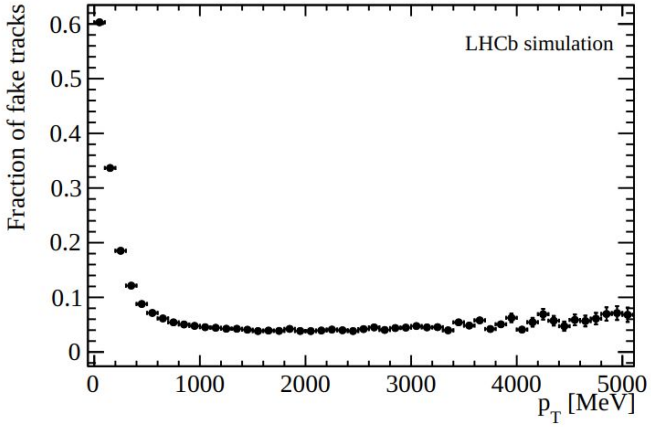


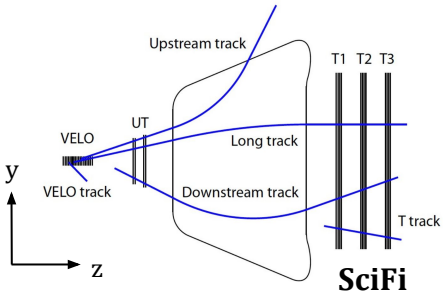
2 **Forward tracking**: VELO+UT-track extension to SciFi (# hits ≥ 10)



- all tracking layers used
- best p resolution → analysis







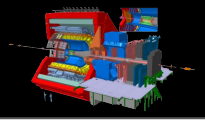
SciFi tracker

- **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)

more in backup
"LHCb - tracking in HLT2"

ALICE - upgrades in LS2

ALICE-TDR-018 ALICE-TDR-019
ALICE-TDR-016
J. Phys. G: Nucl. Part. Phys. 41 (2014) 087002

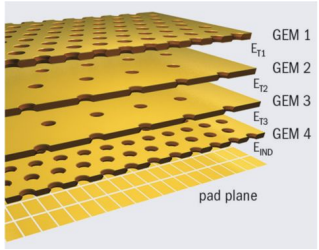


mfaggin@cern.ch

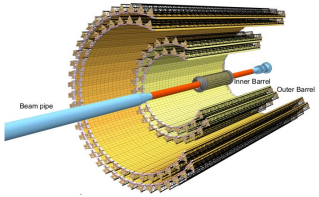
Challenges

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

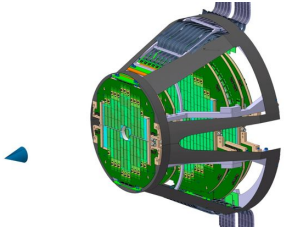
Detector upgrades



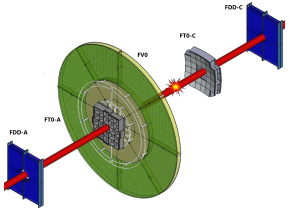
Time Projection Chamber (TPC) upgrade → $|\eta| < 0.9$



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

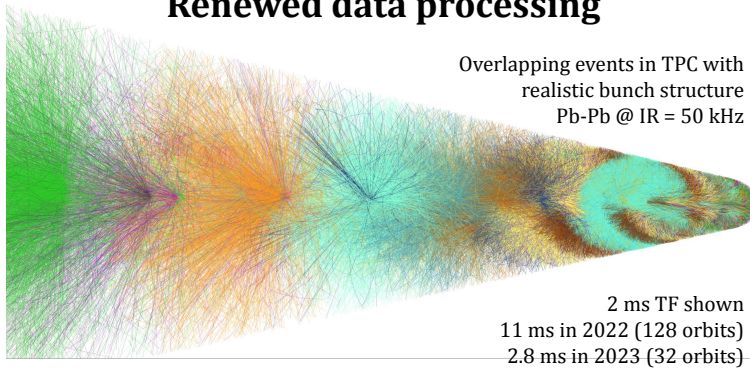


Muon Forward Tracker (MFT) → $2.5 < \eta < 3.6$



Fast Interaction Trigger (FIT) → $2.2 < \eta < 6.3$, $-6.9 < \eta < -2.3$

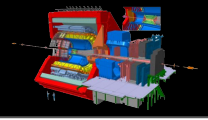
Renewed data processing



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

J. Liu, ***"Run3 performance of new hardware in ALICE"***

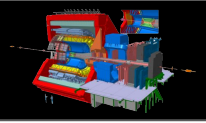
ALICE - mid-*y* tracking in Run 3, 4



- **Continuous readout** of Time Frames (TFs)
- A priori track-to-collision association not possible
- <u>FIT detector</u>
    a. excellent time resolution ($\sigma \leq 18$ ps)
    b. good correlation with PV reconstructed with ***global tracks*** in the central barrel

more in backup *"ALICE - data processing in Run 3, 4"*

ALICE - mid-*y* tracking in Run 3, 4



more in backup *"ALICE - data processing in Run 3, 4"*

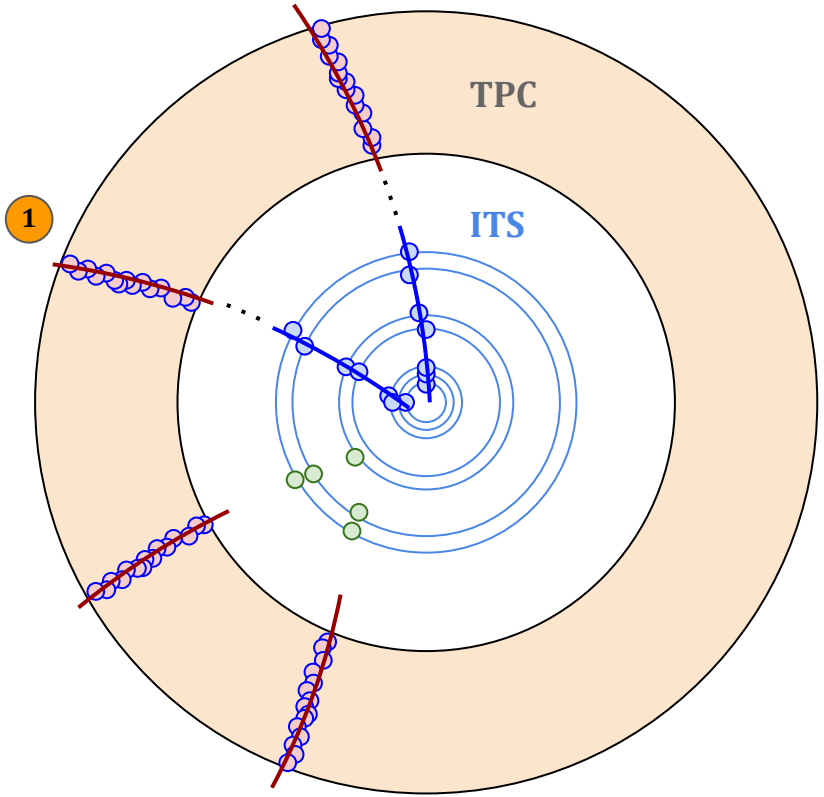
- **Continuous readout** of Time Frames (TFs)
- A priori track-to-collision association not possible
- FIT detector
  a. excellent time resolution ($\sigma \leq 18$ ps)
  b. good correlation with PV reconstructed with ***global tracks*** in the central barrel

1

Time-matching between **ITS** and **TPC**

TPC standalone tracking + **ITS standalone tracking**

- **Main challenge** for 35x **data compression** in synchronous reconstruction
  - TPC: ~3.4 TB/s → ~ 70 GB/s (↓50x)
- Ported to **GPUs**
- **Up to 100x gain** with GPUs compared to 1-core CPU (backup)



ALICE - mid-*y* tracking in Run 3, 4

- **Continuous readout** of Time Frames (TFs)
- A priori track-to-collision association not possible
- FIT detector
  a. excellent time resolution ($\sigma \leq 18$ ps)
  b. good correlation with PV reconstructed with ***global tracks*** in the central barrel

more in backup *"ALICE - data processing in Run 3, 4"*
more in backup *"ALICE - mid-y tracking in Run 3, 4"*

1

Time-matching between **ITS** and **TPC**

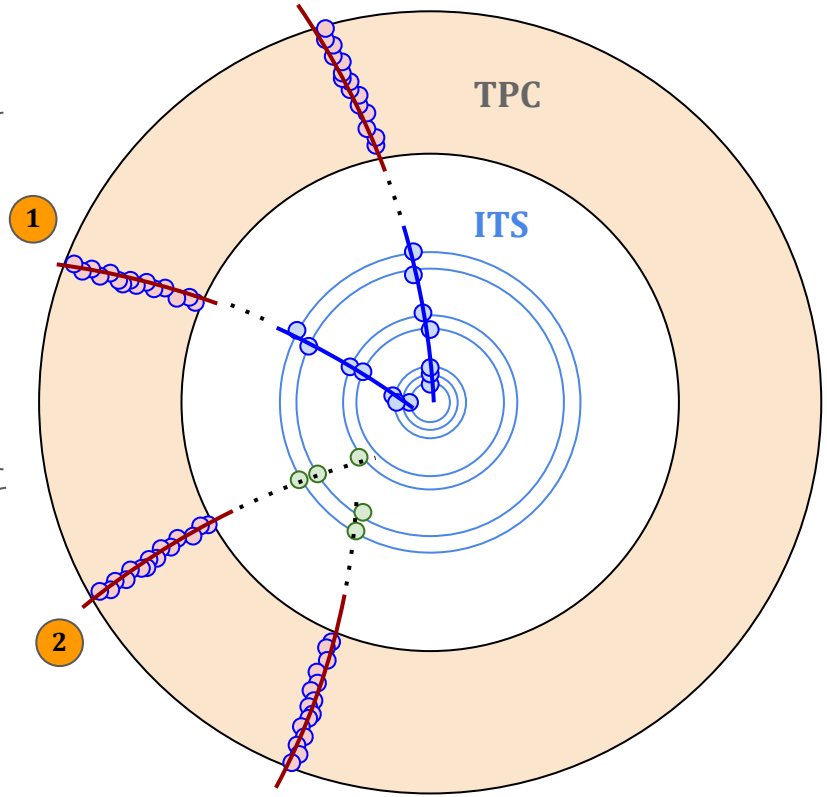
TPC standalone tracking + **ITS standalone tracking**

2

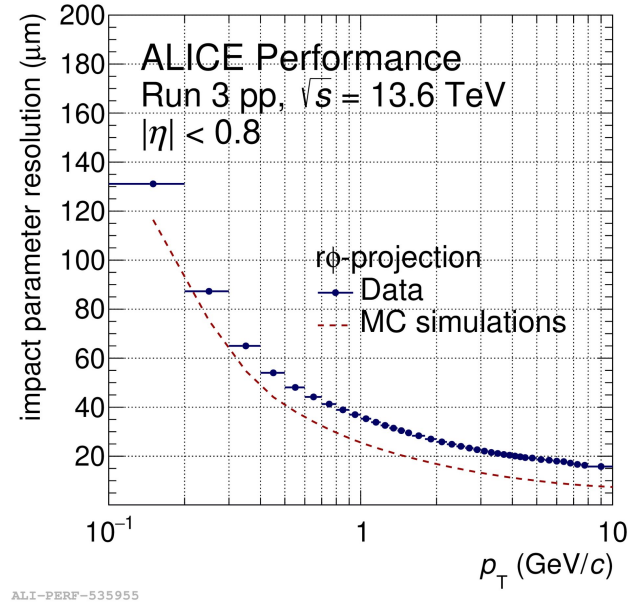
"***Afterburner***" algorithm

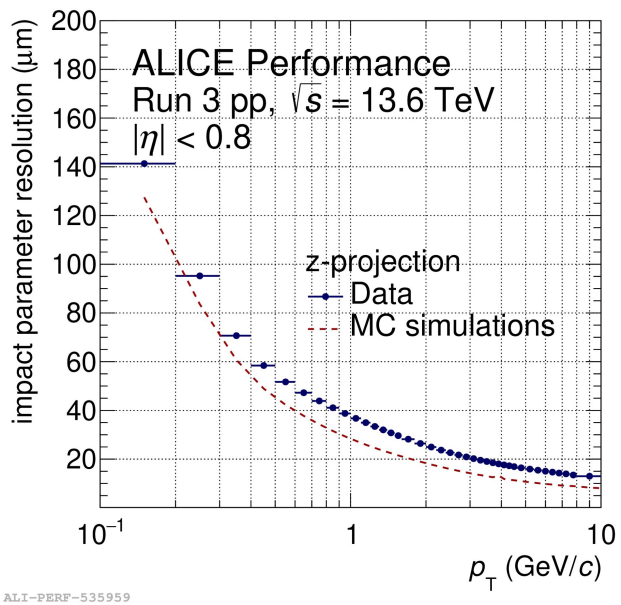
TPC standalone tracking + **unused ITS clusters**

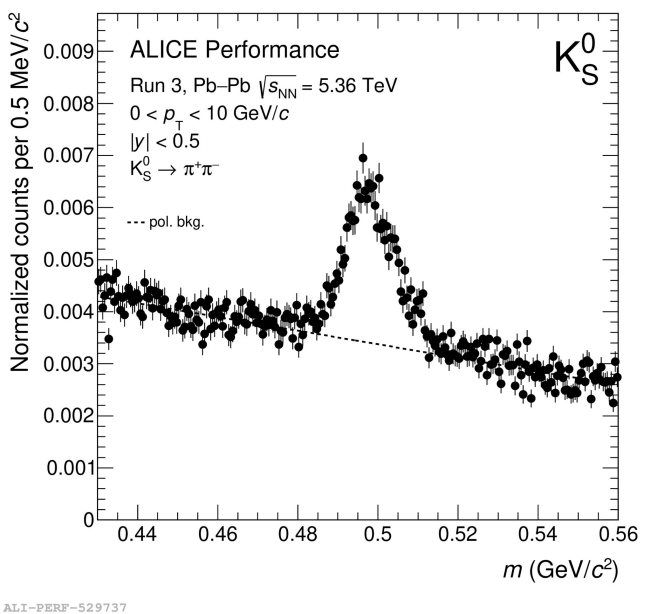
Useful to recover efficiency for V0/cascades decaying within the ITS



ALICE - tracking performance in Run 3, 4







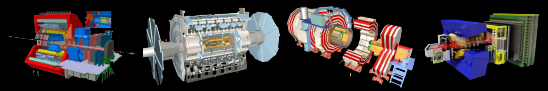
- **Pointing resolution** to the PV of **~35-40 µm @ p_T = 1 GeV/c**
- **2x (4-5x) better** performance in ***rφ*** **(*z*)** compared to Run 2
- Fine-tuning on TPC calibrations/ITS alignment ongoing to fix residual mismatch with MC

- Nice performance for $K^0_S \rightarrow \pi^+\pi^-$ signal reconstruction in 2022 Pb-Pb data

First Run 3 Pb-Pb data!

more in backup
"ALICE - tracking performance in Run 3, 4"

Summary

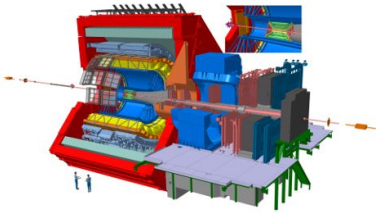


- **Excellent** tracking and vertex **performance** for ALICE, ATLAS, CMS, LHCb experiments
- Several **improvements** in place for Run 3
  - pile-up handling
  - improved tracking in trigger
  - improved tracking in dense environment
  - multi-threading and algorithm optimization
- Experiments ready for fruitful data taking, reconstruction and physics analysis in Run 3!

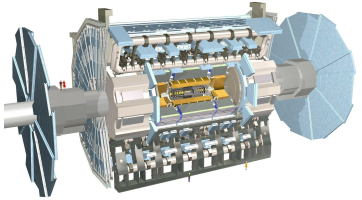
C. Sonnabend, ***"Particle identification"***

J. Liu, ***"Run3 performance of new hardware in ALICE"***

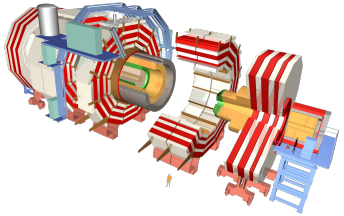
A Large Ion Collider Experiment (ALICE)



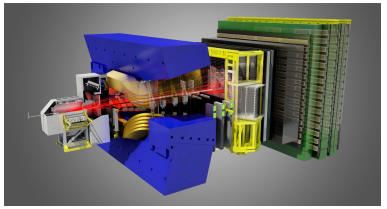
A Toroidal LHC Apparatus (ATLAS)



Compact Muon Solenoid (CMS)



Large Hadron Collider beauty (LHCb)



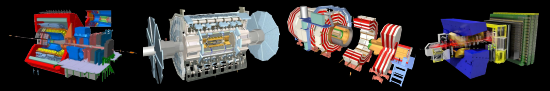
Thanks a lot for the attention

*"This work is **(partially)** supported by ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union – NextGenerationEU".*

Backup

More verbose slides

Tracking and vertexing at the LHC



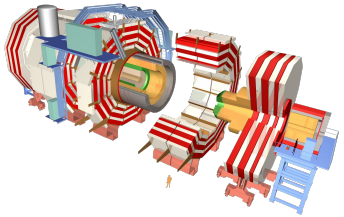
mfaggin@cern.ch

- **<u>Tracking and vertexing</u>**: **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

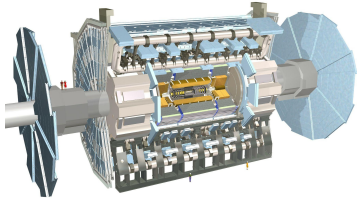
Disclaimers

- This presentation: **summary** of tracking and vertexing in **ALICE, ATLAS, CMS and LHCb**
- Talk focused on charged-particle reconstruction → **no** explicit references to **muon-chamber tracking**

Compact Muon Solenoid (CMS)

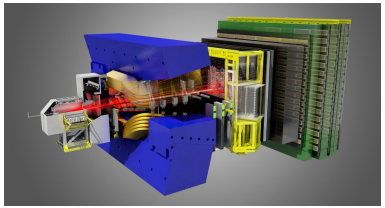


A Toroidal LHC Apparatus (ATLAS)

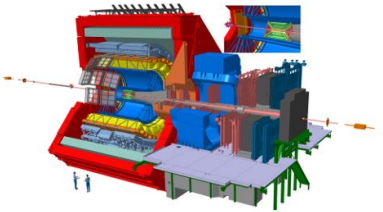


Reconstruction algorithms renewed to handle higher average in-bunch pile-up ($\langle\mu\rangle$) collisions

Large Hadron Collider beauty (LHCb)



A Large Ion Collider Experiment (ALICE)

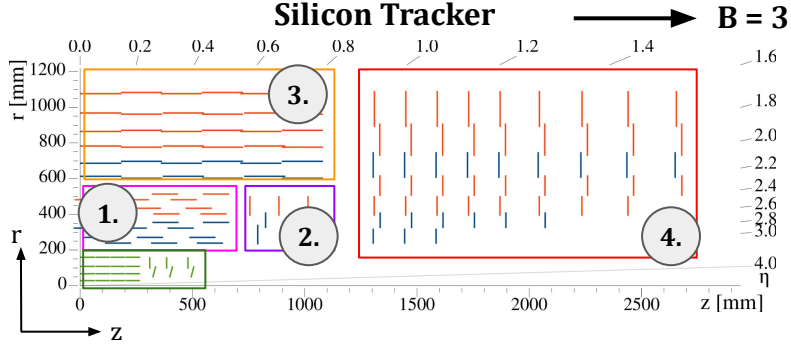


Major detector upgrades and renewed data flow to significantly increase the collected statistics in Run 3, 4

CMS - Silicon Tracker and tracking

2021 JINST 16 P02027
2008 JINST 3 S08004
2014 JINST 9 P10009
arXiv:2304.05853v1, CMS-DP-2022-018

mfaggin@cern.ch



- **CMS Phase-1 pixel detector** ($|\eta| < 3$)
  - mid-η: 4 layers L1-L4 → r = 2.9 - 16.0 cm
  - forward-η: 3 disks (D1-D3) on each end

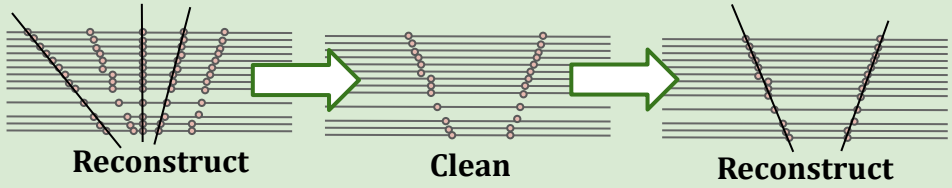
- **Silicon Strip Tracker**
  1. Tracker Inner Barrel (**TIB**)
     - 4 barrel layers (r < 55 cm)
  2. Tracker Inner Disks (**TID+-**)
     - 3 disks with 3 rings on each side ($|z|$ < 118 cm)
  3. Tracker Outer Barrel (**TOB**)
     - 6 barrel layers (r > 55 cm, $|z|$ < 118 cm)
  4. Tracker EndCaps (**TEC+-**)
     - 9 disks with up to 7 rings on each side ($|z|$ > 118 cm)

Hermetic tracking system within $|\eta| < 2.5$

RUN 2

Combinatorial Track Finder (CTF)

- **Combinatorial Kalman Filter (CKF)**: pattern recognition + track fitting
- **Iterative tracking** →easiest topologies first (e.g. high p_T, primaries)
- Different settings in seeding/track finding in each iteration to look for different track categories (next slide)



MATRIPLEX Kalman-fitter algorithm (mkFit)

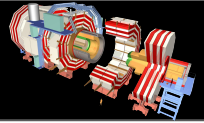
- **Parallelized** and **vectorized CKF**
- **Similar** physics **performance** as the CKF
- Significant **speed up**
- Used by a subset of tracking iterations reconstructing ~90% hard-scattering event tracks

RUN 3

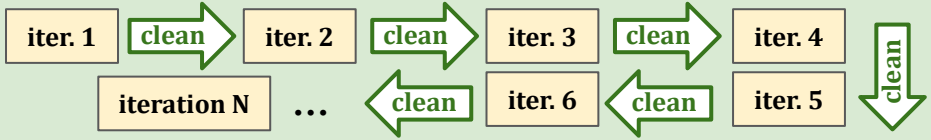
CMS - offline tracking

RUN 2 | **CKF**

2014 JINST 9 P10009
arXiv:2304.05853v1
CMS-DP-2022-018



mfaggin@cern.ch



10 iterations (+ 2 for μ-chambers)

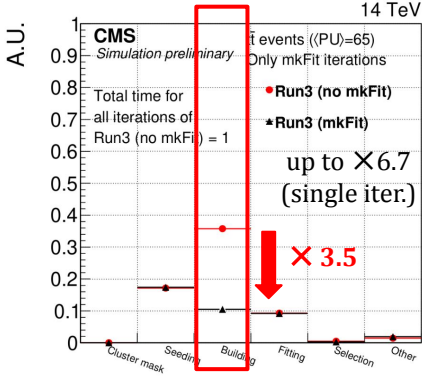
1. **Seeding**: starting from the inner part of tracker, despite the larger track density
   - pixel granularity (66M in $1m^2$)→ 10-100x lower occupancy than outer strip layers
   - 3D points from pixels and/or at least two mono-stereo layers in the Silicon Strip Tracker(double-side strips → *matched hits*)
   - higher efficiency, also to ease low-pt track reconstruction
2. **Track finding / pattern recognition**
   - Outward KF + further inward search (add seeding hits; recover $r\varphi$ 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

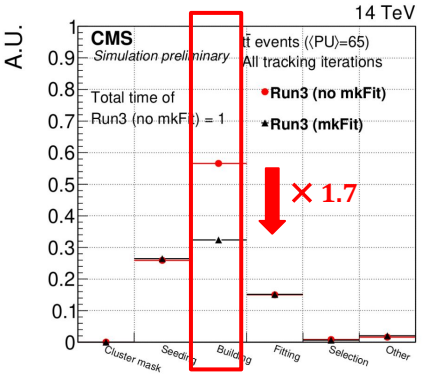
mkFit

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

CMS - HLT tracking and vertexing

- 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-Run2: 3 steps

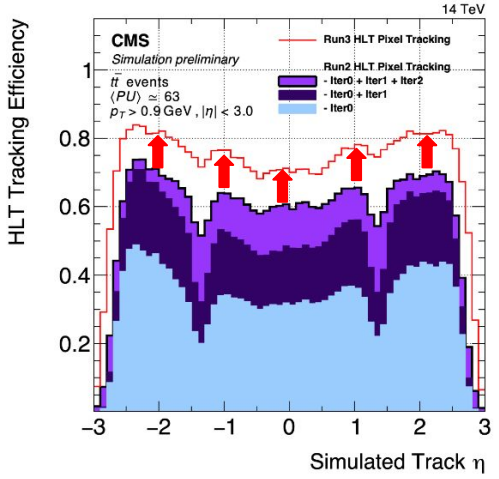
	Selections
iter. 0	4 pixel hits, p_T>0.8 GeV/c
iter. 1	4 pixel hits, p_T>0.4 GeV/c
iter. 2	4 pixel hits, p_T>0.4 GeV/c around jets (calo + iter 0, 1)

PATATRACK

HLT-Run3: 1 step

	Selections
iter. 0	at least 3 pixel hits, p_T>0.8 GeV/c

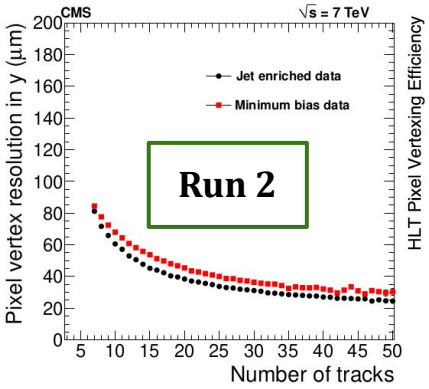
- **HLT pixel tracking** ported to **GPUs**
  → heterogeneous computing with CUDA (Nickolls et al., 2008)
  → seeding via Cellular Automaton algorithm
- Validation with CMS open data from 2018 (backup) and Run3 projections with simulated pp @ 14 TeV with ttbar events
- Better physics performance and throughput

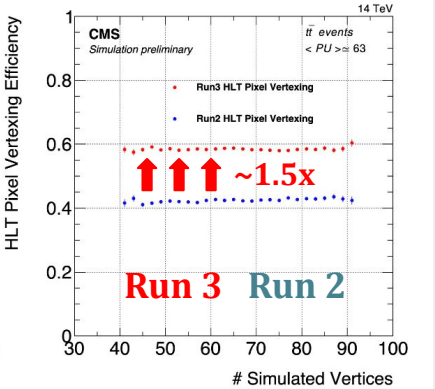


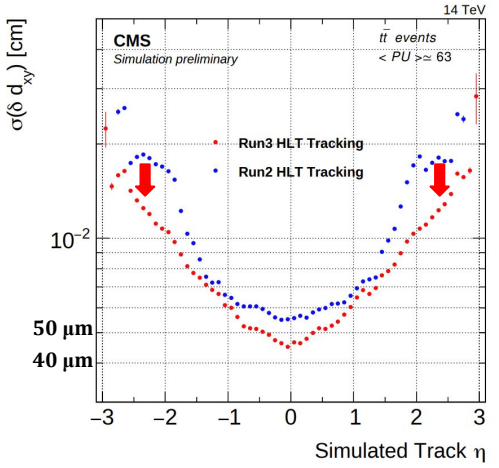
HLT pixel primary vertexing

1. **Track selection** (# hits ≥ 4, p_T > 0.5 GeV/c)
2. **Clustering**
   - track ordering based on z distance from beam-spot
   - "gap clustering": tracks with Δz < 2 mm split into separated vertices
3. **Fitting** via Adaptive Vertex Fitter
   - annealing to avoid local minima

→ Resolution from *"splitting method"*: two track subsets, σ from Gaus fit of the difference of the two fitted vertices



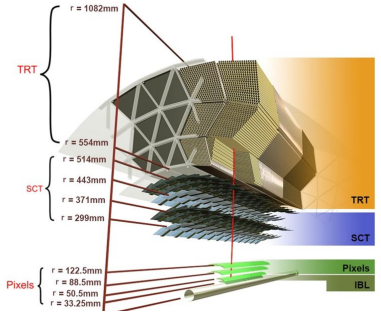




ATLAS - inner detector and tracking

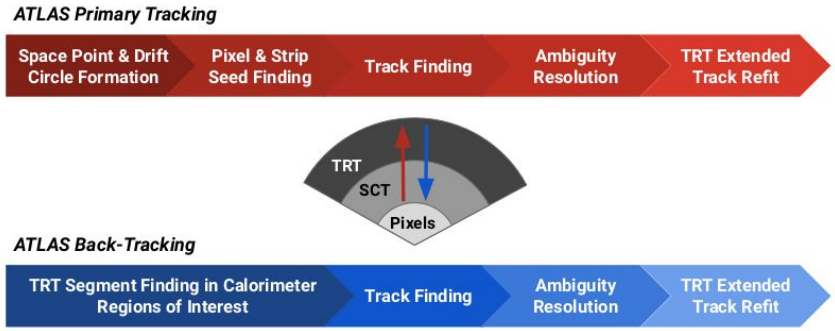
Eur. Phys. J. C (2017) 77:673
ATL-PHYS-PUB-2021-012

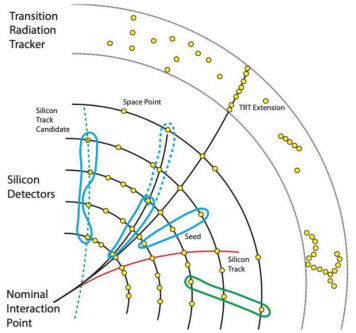
mfaggin@cern.ch



Inner Detector (ID) tracker ($|\eta| < 2.5$)

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 $\perp$ beam line within 0.8 m < $|z|$ < 2.7 m





Primary tracking (INSIDE-out) → primaries

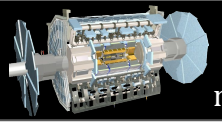
- Seeding: triplets in pixel + SCT
- TRKFIND: CKF to extend tracks outwards up to SCT outer layers
- Track **ambiguity solver**
  - scoring based on hit topology (holes, shared hits) and quality (χ^2, …)
  - track-quality selections (e.g. # hits ≥ 7; shared clusters/track ≤ 2)
  - neural network (NN) 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 - Run 3 optimization

Eur. Phys. J. C (2017) 77:332
ATL-PHYS-PUB-2021-012
ATL-PHYS-PUB-2019-015



mfaggin@cern.ch

Run 2: **μ = 20-40**

Run 3: **μ ~ 50**

Challenge

⬇️resource consumption
≥ track quality

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

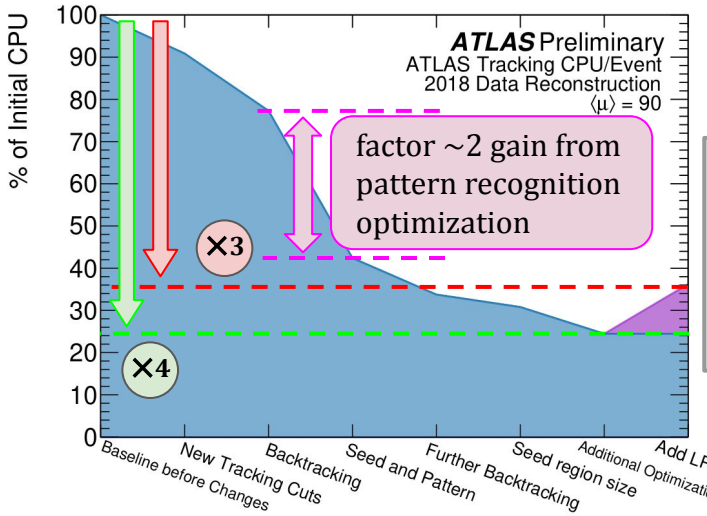
1 **Tighter selections** for the **ambiguity solver**
→pixel + SCT hits ≥ 8 (old: 7); $|dca_{XY}| < 5$ mm (old: 10 mm)
→low-quality tracks reduced

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



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

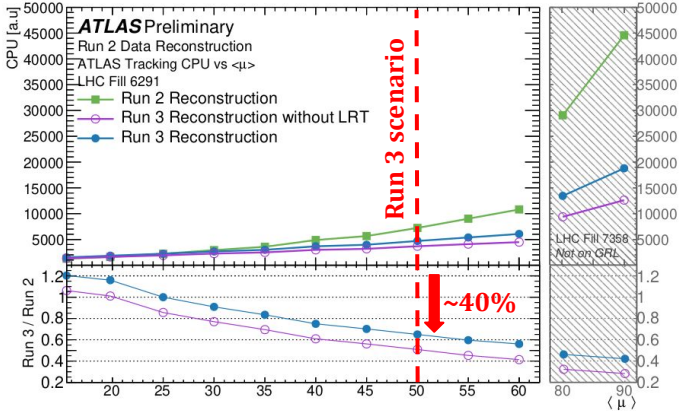
Adaptive multi-vertex fitter (AMVF)
- track weights for more than 1 vertex at a time
- convergence to 1 vertex due to deterministic annealing

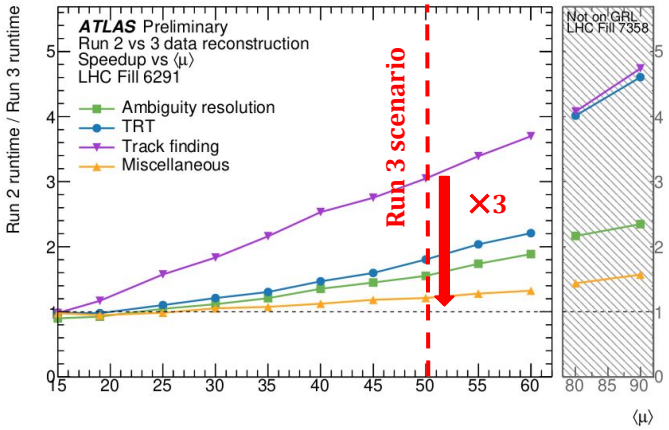
NEW

ATLAS - Run 3 performance

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

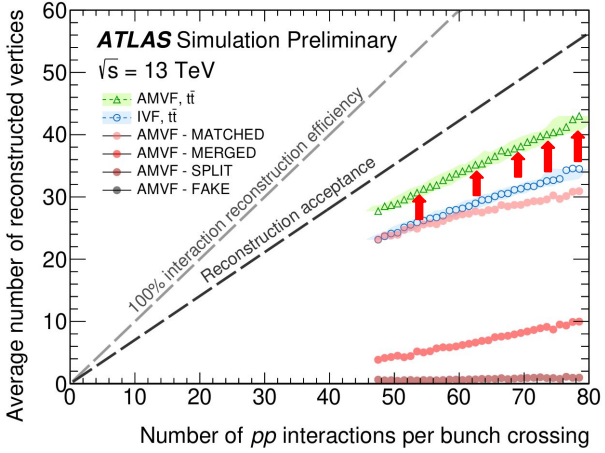
mfaggin@cern.ch



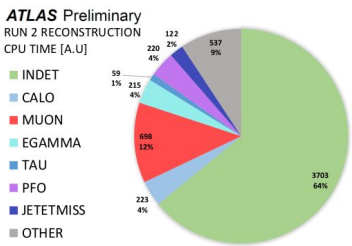


Performance on Run 2 data

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



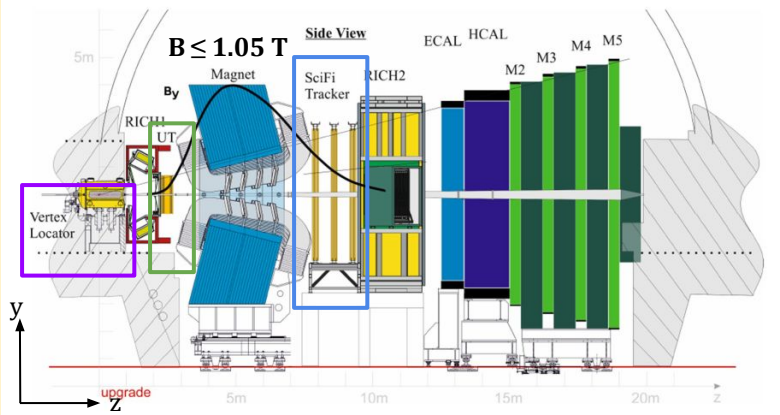
- **Stronger-than-linear scaling** vs. ⟨μ⟩ with **Run 2 reconstruction chain**
- Near **linear scaling** vs. ⟨μ⟩ with **Run 3 reconstruction chain**
  - ⟨μ⟩ ~ 50: CPU usage lower of ~40%
  - ⟨μ⟩ ~ 50: pattern recognition runtime ~3 times lower (1.5-2 others)
- **ID tracking and vertexing** only ~40% total CPU (~64% in Run 2)
- **AMVF** recovers **up to 35%** of the reconstructable primary vertices at high ⟨μ⟩, lost by the IVF



Challenges

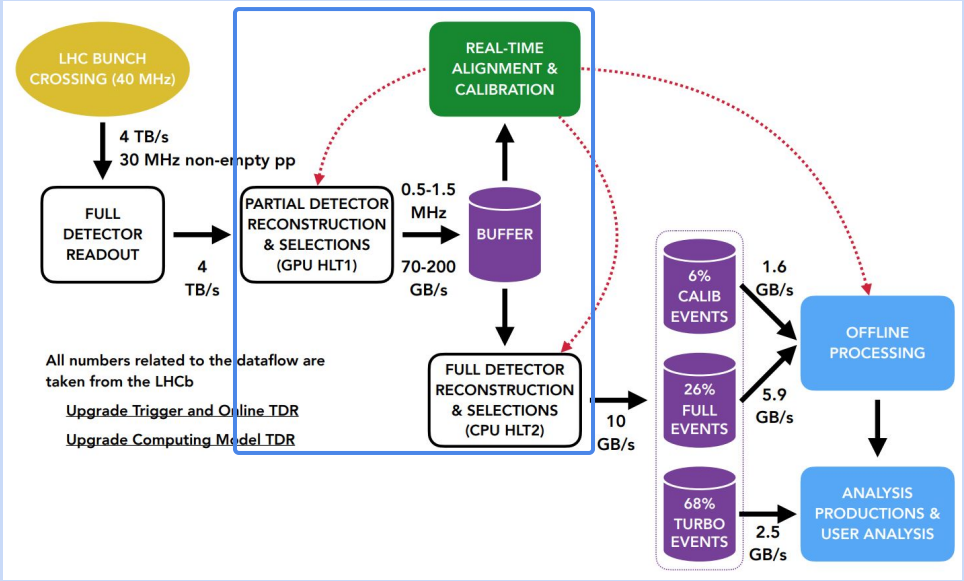
- bunch-crossing (BC) rate up to 40 MHz
- pile-up: $\langle\mu\rangle$ ~1.4 → $\langle\mu\rangle$ ~5
- $\mathcal{L}_{peak}$ ~2×10^{33} cm^{-2}s^{-1}, $\mathcal{L}_{int}$ ~50 fb^{-1} (Run 3,4)

Detector upgrades (tracking only!)



- **Vertex Locator (VELO)** $2 < \eta < 5$
  - [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

Renewed data flow



Level-0 hardware trigger (~1 MHz) → software trigger to be (~30 MHz non-empty pp collisions)

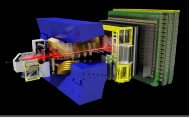
1. GPU **H**igh-**L**evel **T**rigger **1** (**HLT1**)

   — Real-time alignment and calibrations —

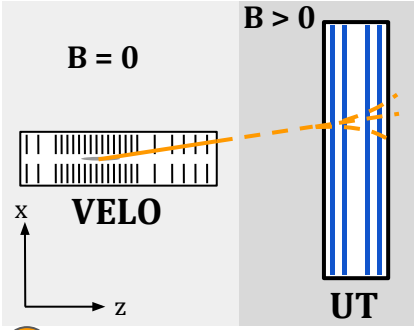
2. CPU **H**igh-**L**evel **T**rigger **2** (**HLT2**)

LHCb - HLT1 with Allen

LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)

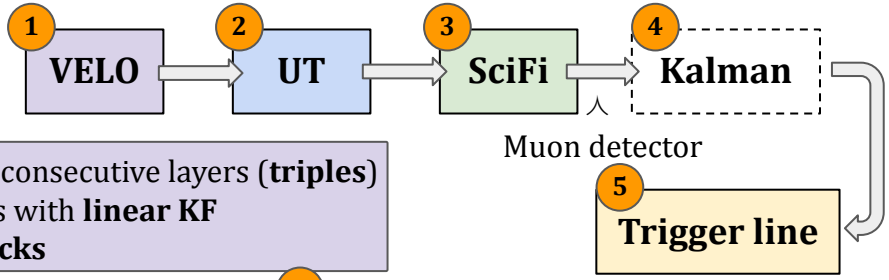


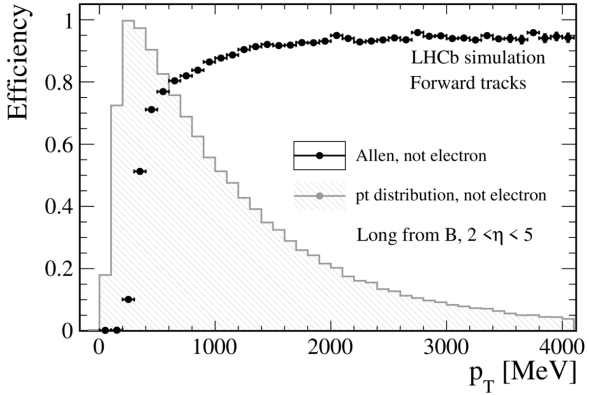
mfaggin@cern.ch

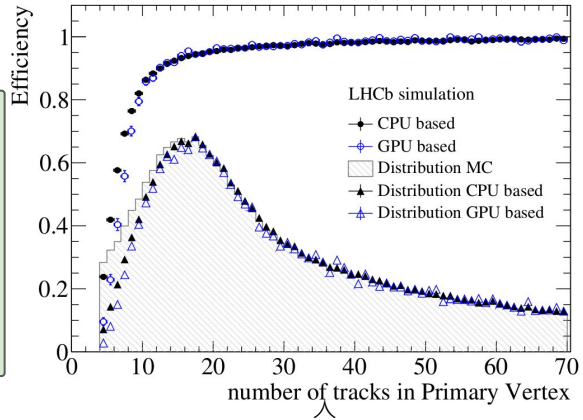


Allen: A High-Level Trigger on GPU's for LHCb

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







1

- **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)

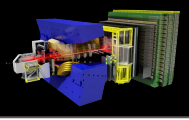
5

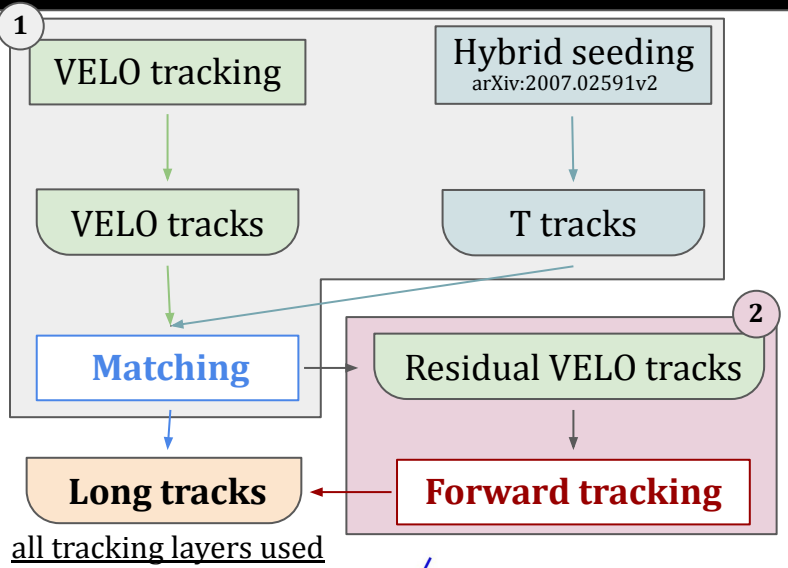
- Parallel fitting of **2-track** secondary vertices (**SV**)
- **Trigger** selections on single **tracks** and/or **SV**
- **Output rate ~ 1 MHz**

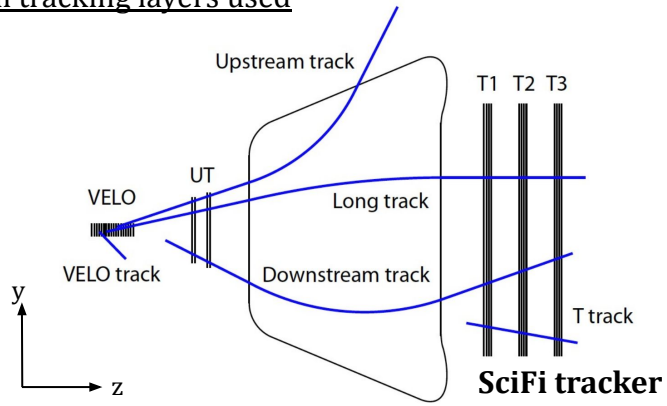
- Tracking efficiency > 90% for p_T>1 GeV/c
- PV efficiency > 90% (95%) for VELO tracks > 10 (20)

VELO

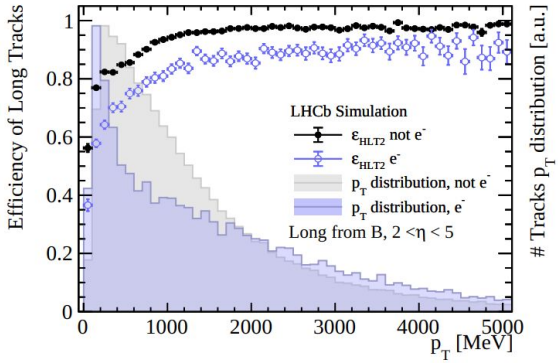
LHCb - tracking in HLT2

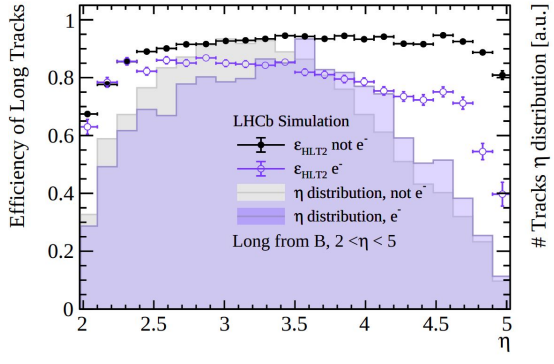






- **Long tracks**: best p resolution → analysis
- Two independent algorithms
  a. **Matching**: neural network trained on MC to match VELO and T tracks
  b. **Forward tracking**: VELO+UT-track extension to SciFi (# hits ≥ 10)
     - [finding] polynomial search window assuming $p > 1.5$ GeV/c
     - [finding] 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

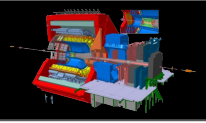




- Tracking efficiency for hadrons and μ←B ~ 90% (> 95% for $p_T > 1$ GeV/c)
- 0.1-0.2 lower efficiency for electrons
  - trajectory deflection due to bremsstrahlung
  - major effect at large η → more material

ALICE - upgrades in LS2

ALICE-TDR-018 ALICE-TDR-019
ALICE-TDR-016
J. Phys. G: Nucl. Part. Phys. 41 (2014) 087002



mfaggin@cern.ch

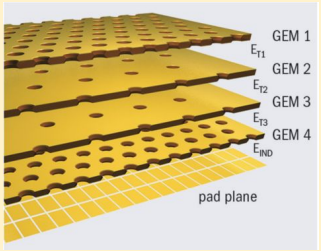
Challenges (pp, $\sqrt{s}$ = 13.6 TeV)

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

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

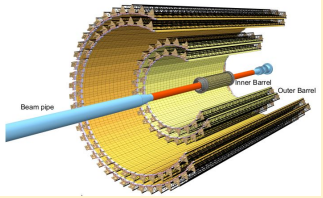
- IR~ 50 kHz
- $\mathcal{L}_{peak}$~6×10^{27} cm^{-2}s^{-1}, $\mathcal{L}_{int}$~13 nb^{-1} (Run 3,4) (×50-100 more than Run 2)

Detector upgrades (tracking only!)



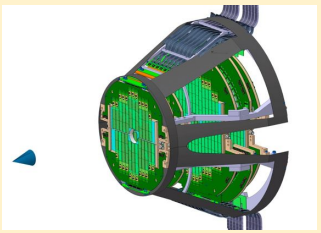
Time Projection Chamber (TPC) upgrade → $|\eta| < 0.9$

- 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 → $|\eta| < 1.3$

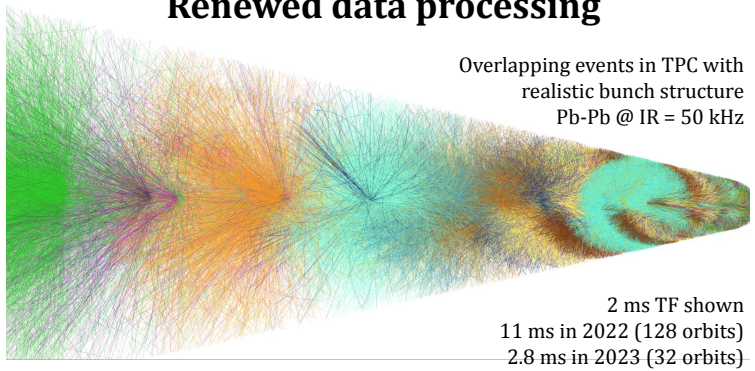
- 7 layers of Alice Pixel Detector (ALPIDE) chips: custom Monolitic Active Pixel Sensors (MAPS)
- Readout rate in Pb-Pb up to 100 kHz → 100x more than Run 2
- Material budget 0.35% X_0 (innermost layer) → 3x lower than Run 2



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

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

Renewed data processing

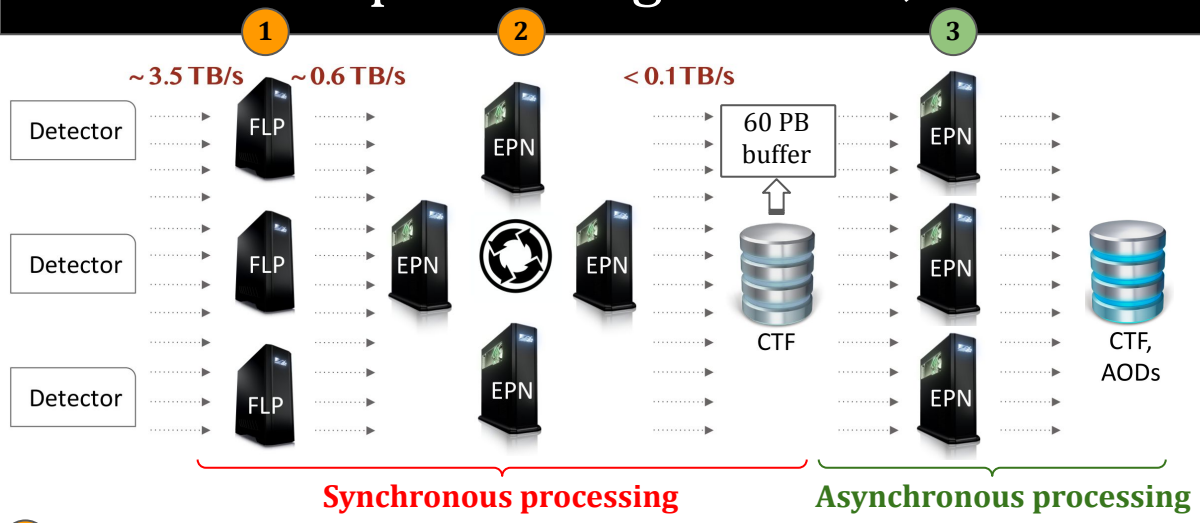


Overlapping events in TPC with realistic bunch structure
Pb-Pb @ IR = 50 kHz

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
- **O^2**: new framework for **online/offline data reconstruction** and **analysis**

ALICE - data processing in Run 3, 4



1 First Level Processors (FLP)

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

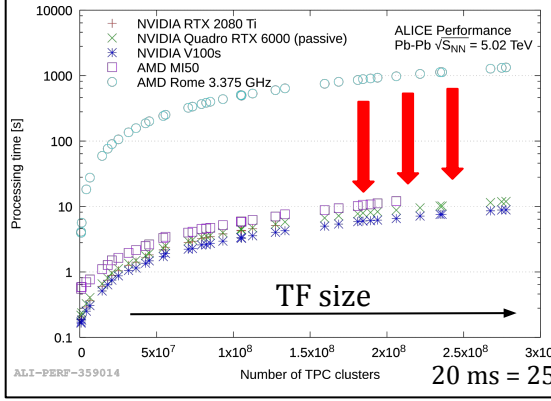
2 Event Processing Nodes (EPN)

- 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

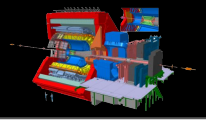
- Goal: total compression factor = **35**
- Main challenge: **TPC**
  - ~3.4 TB/s → ~ 70 GB/s (↓50x)
- **Clusterization** and **tracking** on O(2000) **GPUs** AMD MI50
  - >40x faster, only 4x more expensive



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 in Run 3, 4



TPC tracking

1. Tracking within a φ sector (36)
   a. **C**ellular **A**utomaton (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

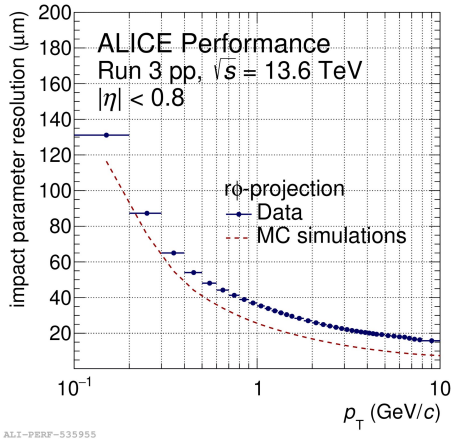
ITS tracking

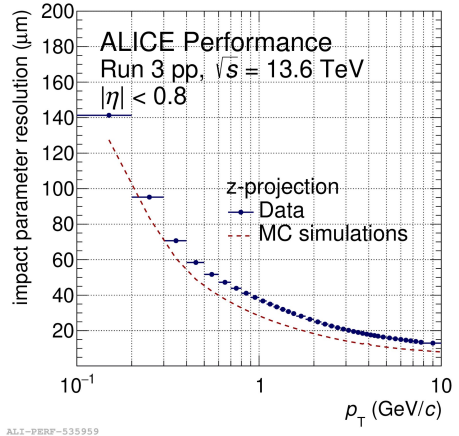
1. PV seeding
   a. tracklets in 3 innermost layers
   b. linear extrapolation of tracklets
   c. clustering to find collision point(s)
2. Track finding and fitting
   a. PV used to reduce combinatorics in matching the hits
   b. CA: track segments (cells) connection into candidate tracks
   c. KF fit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

ITS-TPC tracks

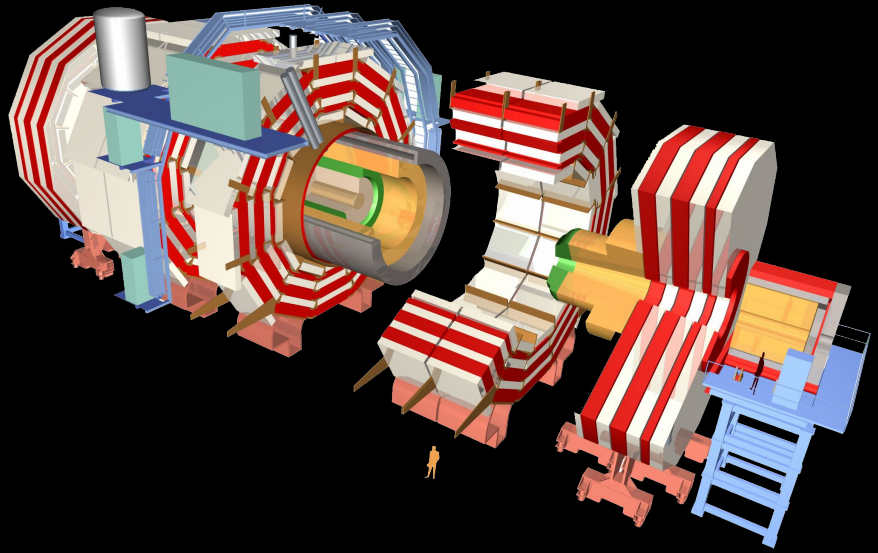
- Time-matching:
  - among ITS and TPC standalone (SA) tracks
  - between a TPC-SA track and left ITS clusters (*afterburner*)
    → Daughters of V0/cascades decaying in ITS
- Prolongation to TRD/TOF
- KF refit outwards and inwards
  → Async. reco.: final calibrations for position-dependent TPC distortions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

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

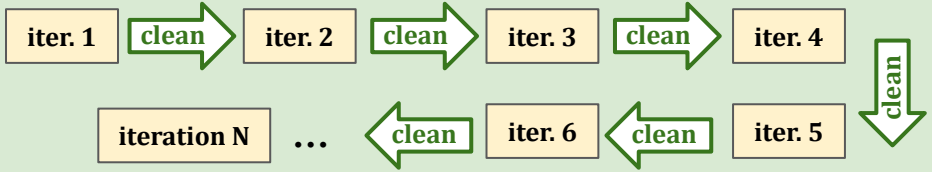




CMS



CMS - tracking

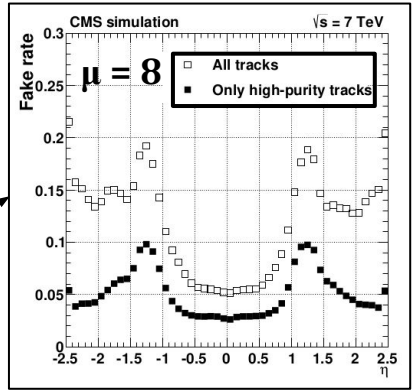


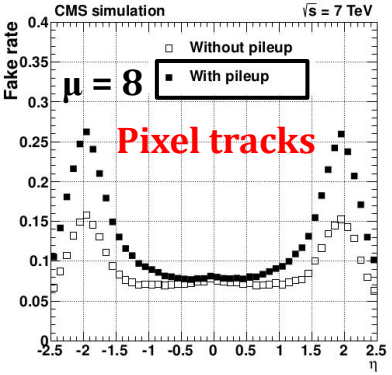
1. **Seeding**: starting from the inner part of tracker, despite the larger track density
   - pixel granularity (66M in $1m^2$)→ lower occupancy (✕10-100)
   - 3D points from pixels and 2 innermost layers of TIB (double-side strips → *matched hits*)
   - higher efficiency, also to ease low-pt track reconstruction
2. **Track finding (TRKFIND)**
   - Outward 4-step KF + further inward search (add seeding hits; matched hits to reduce seeding combinations)
   - Trajectory cleaner (iterative)→ remove tracks with # shared clusters > 19%
3. **Track fitting (TRKFIT)**
   - 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

Offline tracking

High-Level Trigger (HLT)

1. Pixel tracks/vertices → fast (only 3 tracking layers, low occupancy)

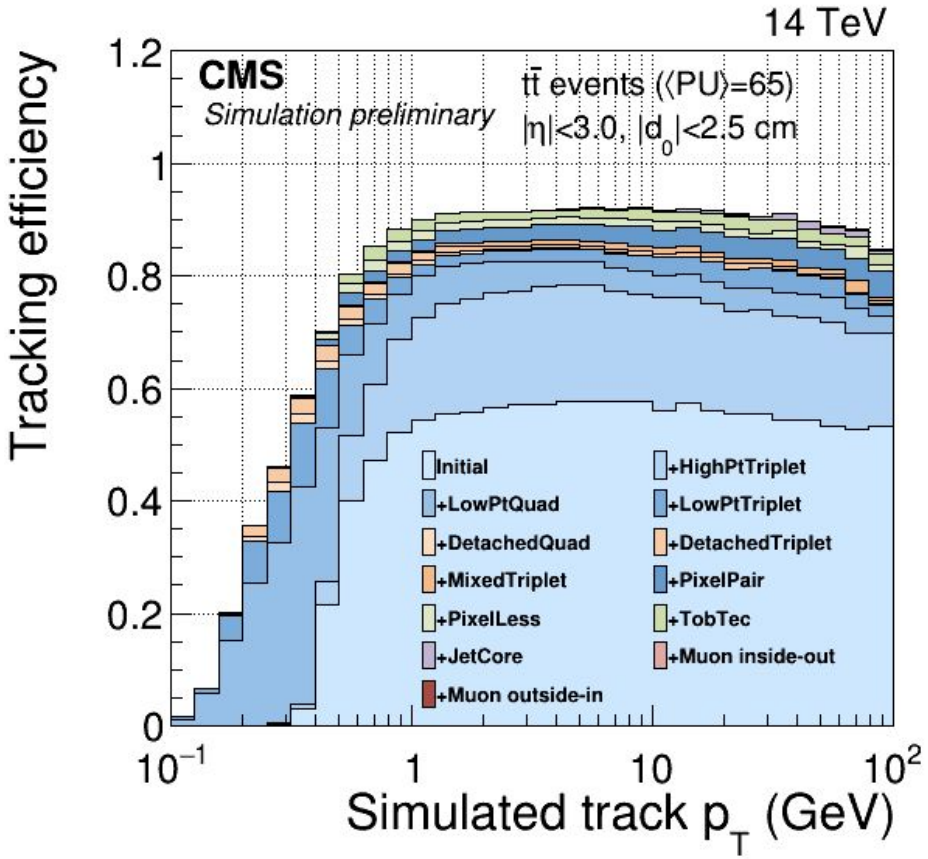


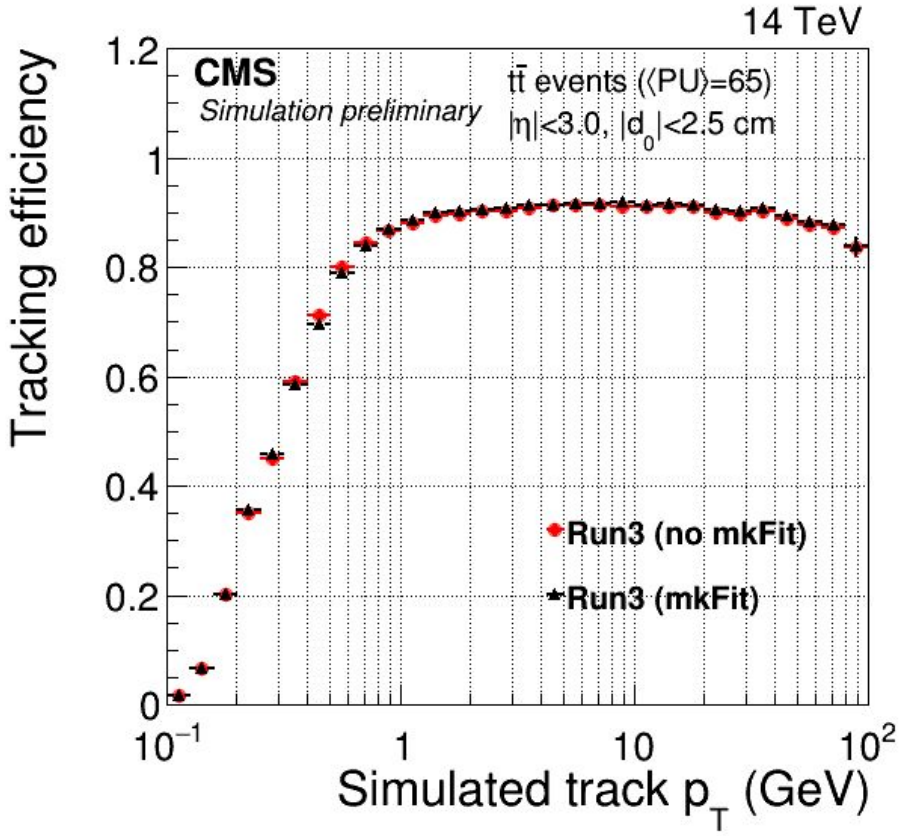


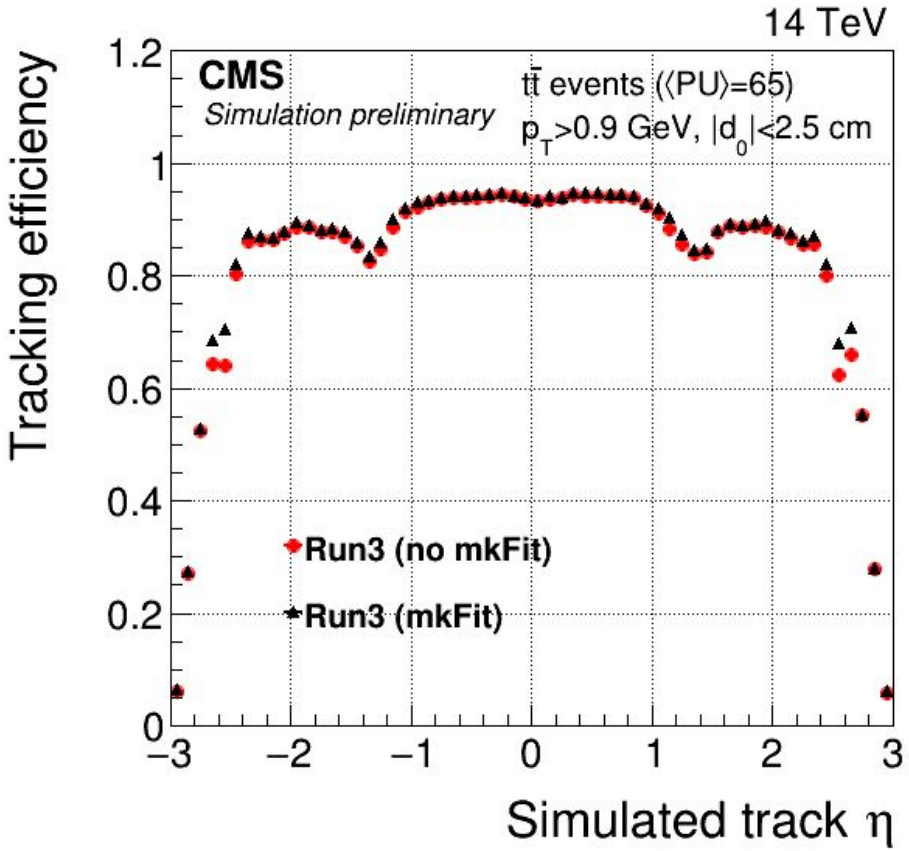
2. Pixel + strips → higher CPU usage
   a. local seeding + track reconstruction
   b. 1 iter. + higher p_T for seeding
   c. partial TRKFIND (~ w/o outer layers)

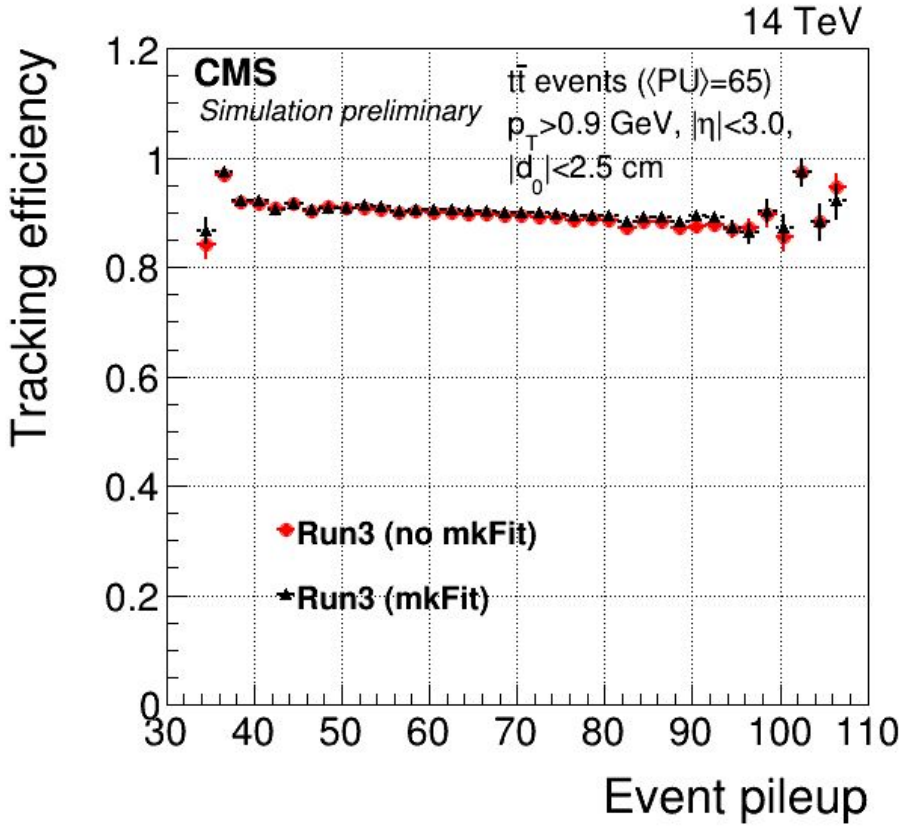
- *InitialPreSplitting*, initial iteration before splitting merged pixel clusters in dense jet environments;
- *Initial*, initial iteration;
- *HighPtTriplet*, high-p_T triplet iteration;
- *DetachedQuad*, detached quadruplet iteration;
- *DetachedTriplet*, detached triplet iteration;
- *PixelLess*, pixel-less iteration.

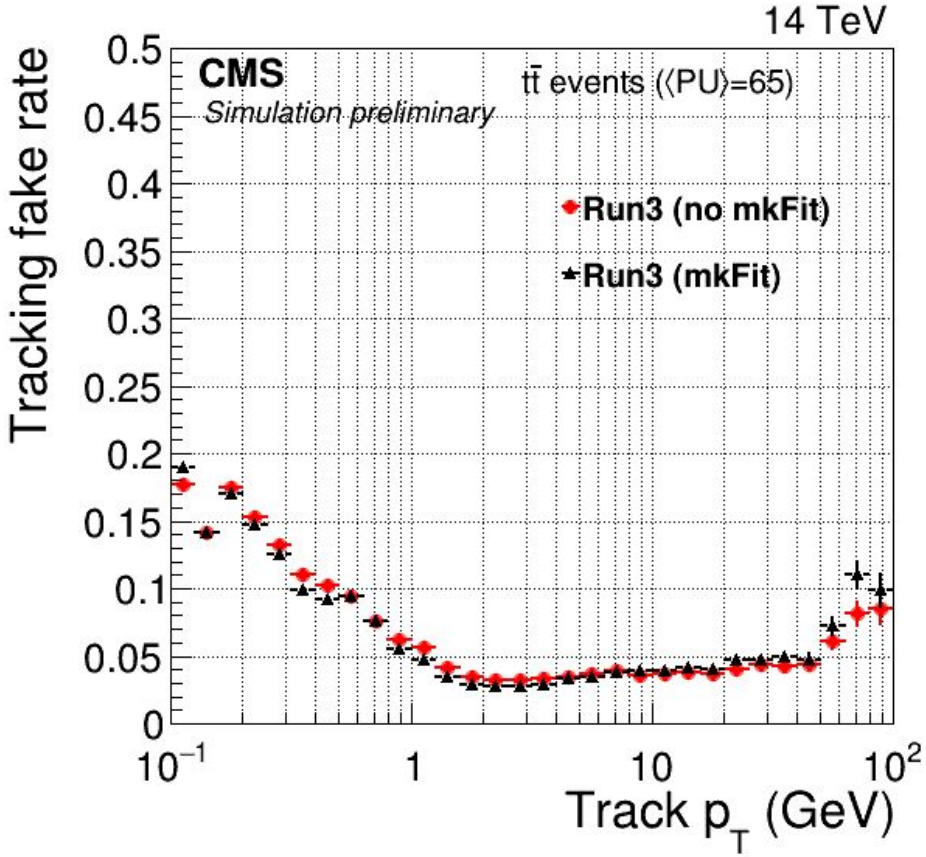
Iteration	Seeding	Target track
Initial	pixel quadruplets	prompt, high p_T
LowPtQuad	pixel quadruplets	prompt, low p_T
HighPtTriplet	pixel triplets	prompt, high p_T recovery
LowPtTriplet	pixel triplets	prompt, low p_T recovery
DetachedQuad	pixel quadruplets	displaced--
DetachedTriplet	pixel triplets	displaced-- 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

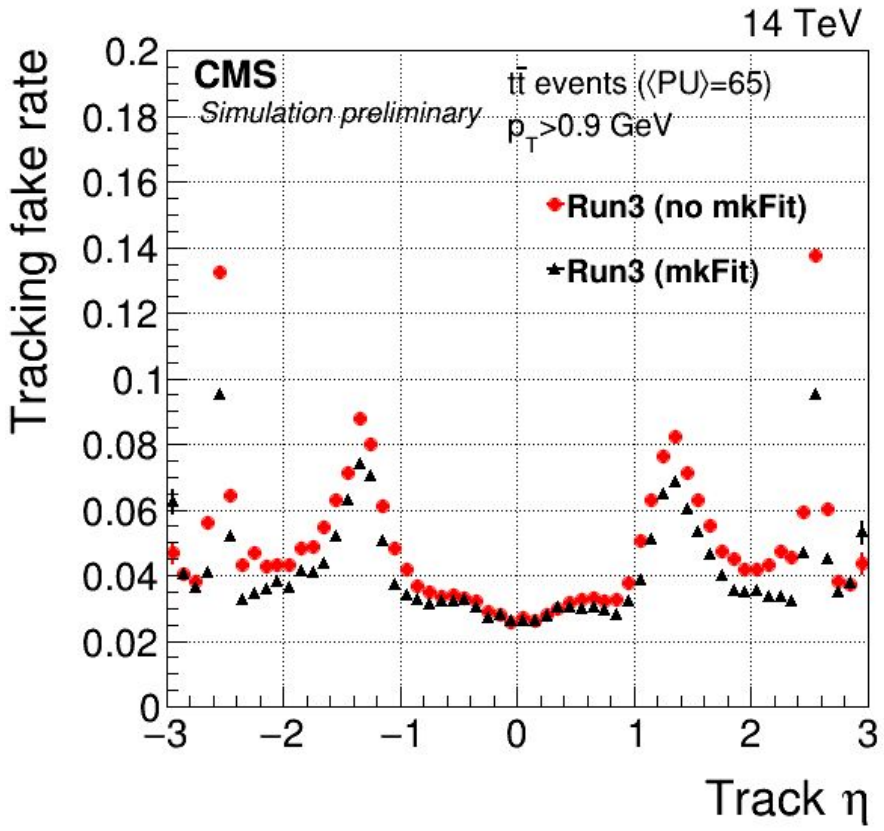

14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
|η|<3.0, |d0|<2.5 cm
Tracking efficiency
Simulated track pT (GeV)
Initial
+LowPtQuad
+DetachedQuad
+MixedTriplet
+PixelLess
+JetCore
+Muon outside-in
+HighPtTriplet
+LowPtTriplet
+DetachedTriplet
+PixelPair
+TobTec
+Muon inside-out

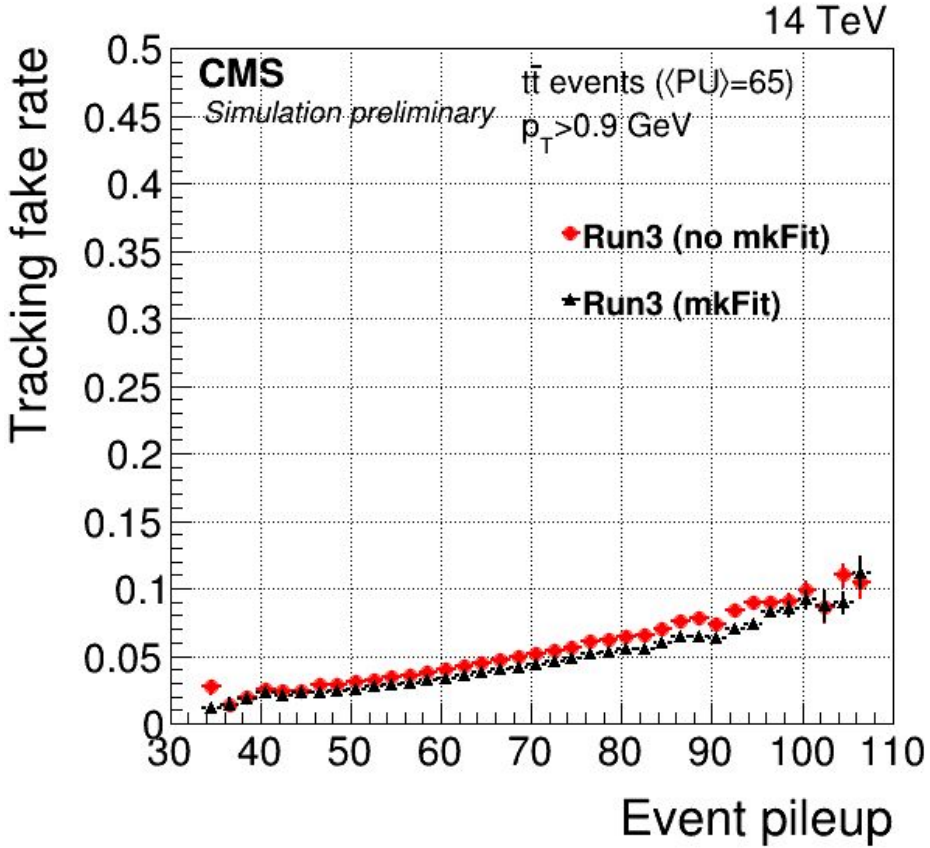

14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
|η|<3.0, |d0|<2.5 cm
Tracking efficiency
Simulated track pT (GeV)
Run3 (no mkFit)
Run3 (mkFit)

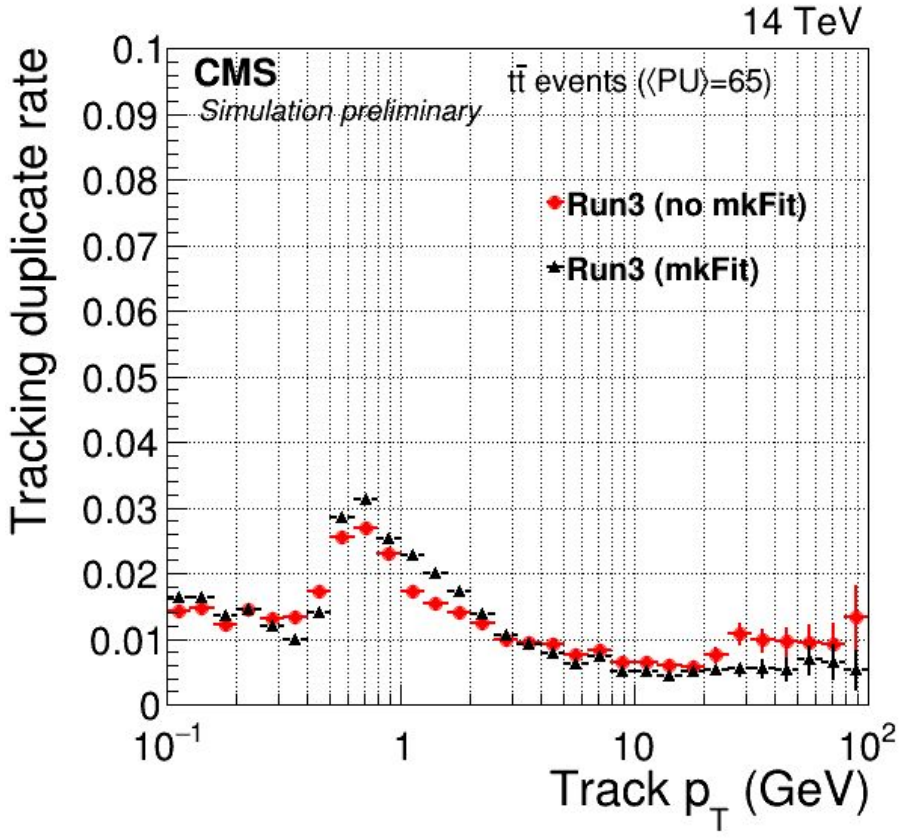

14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
p_T>0.9 GeV, |d_0|<2.5 cm
Run3 (no mkFit)
Run3 (mkFit)
Tracking efficiency
Simulated track η

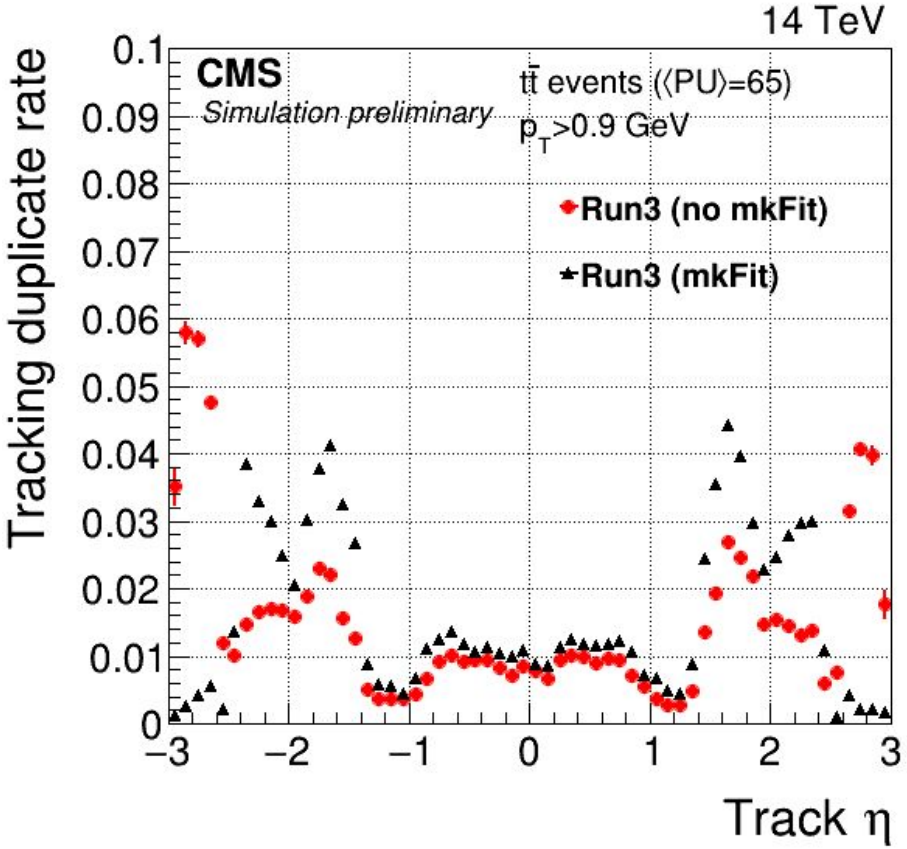

14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
p_T>0.9 GeV, |η|<3.0,
|d_0|<2.5 cm
Run3 (no mkFit)
Run3 (mkFit)
Tracking efficiency
Event pileup

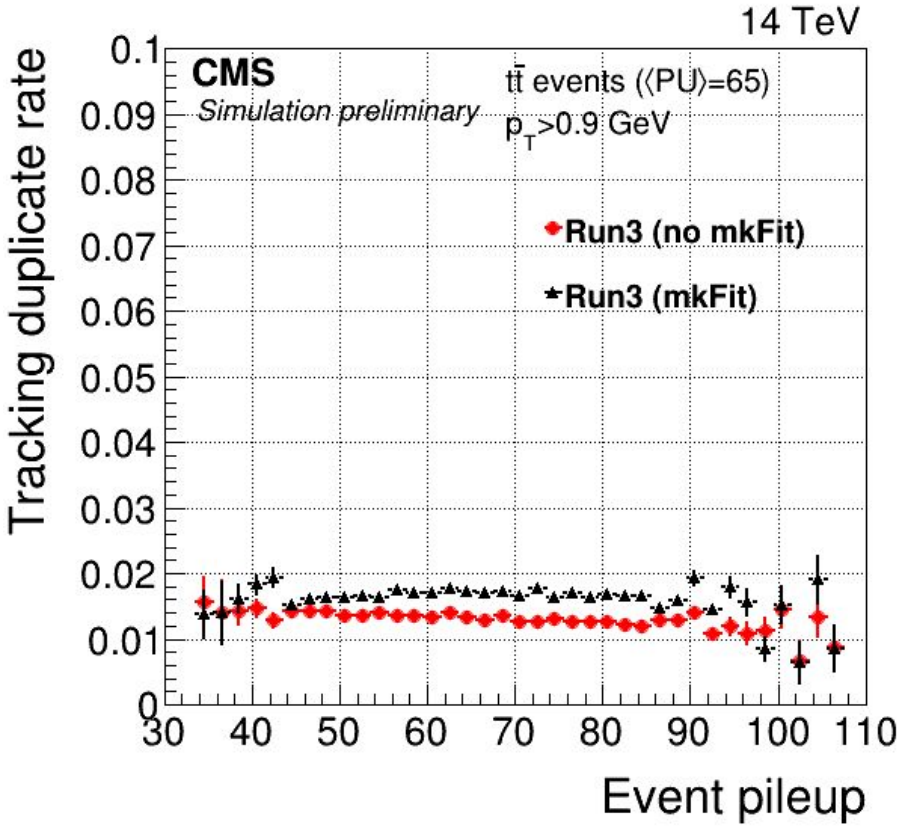

14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
Run3 (no mkFit)
Run3 (mkFit)
Tracking fake rate
Track p_T (GeV)


14 TeV
CMS
Simulation preliminary
tt̄ events (⟨PU⟩=65)
p_T>0.9 GeV
Run3 (no mkFit)
Run3 (mkFit)
Tracking fake rate
Track η

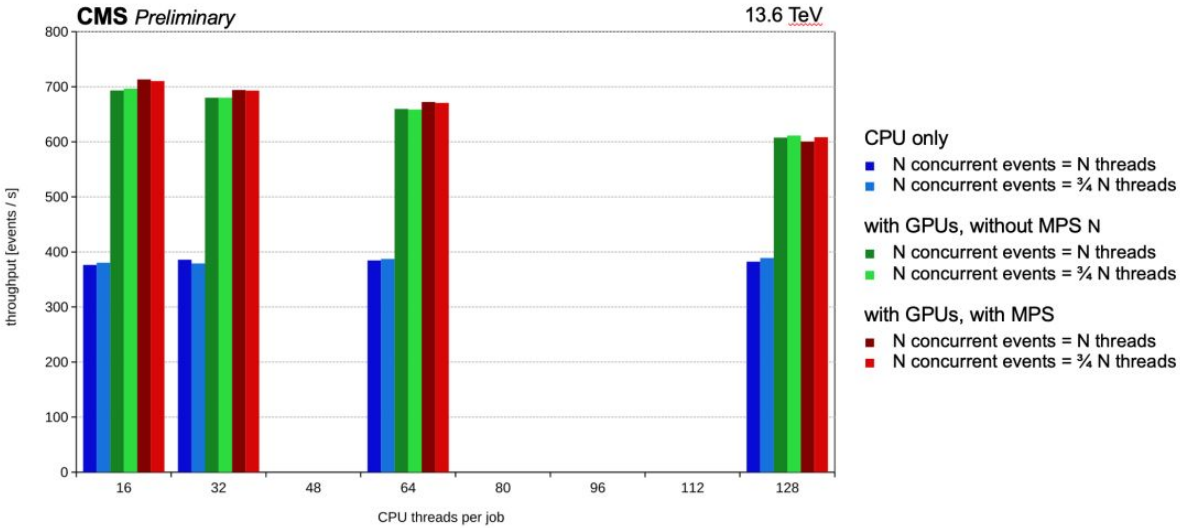

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events ⟨PU⟩=65)
p_T>0.9 GeV
Run3 (no mkFit)
Run3 (mkFit)
Tracking fake rate
Event pileup


14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events ⟨PU⟩=65)
Run3 (no mkFit)
Run3 (mkFit)
Tracking duplicate rate
Track p_T (GeV)


14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events (⟨PU⟩=65)
p_T>0.9 GeV
Run3 (no mkFit)
Run3 (mkFit)
Tracking duplicate rate
Track η

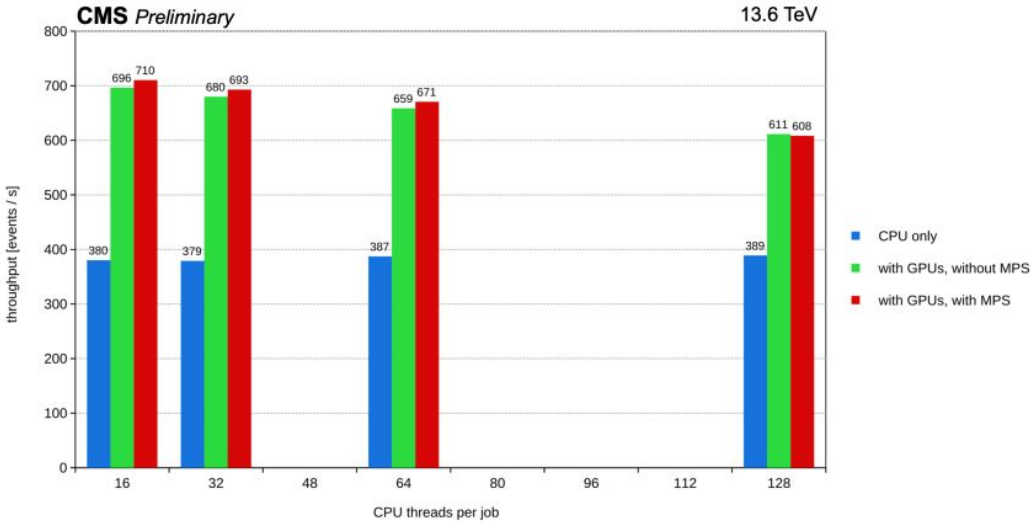

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events (⟨PU⟩=65)
p_T>0.9 GeV
Run3 (no mkFit)
Run3 (mkFit)
Tracking duplicate rate
Event pileup

CMS - HLT tracking and vertexing



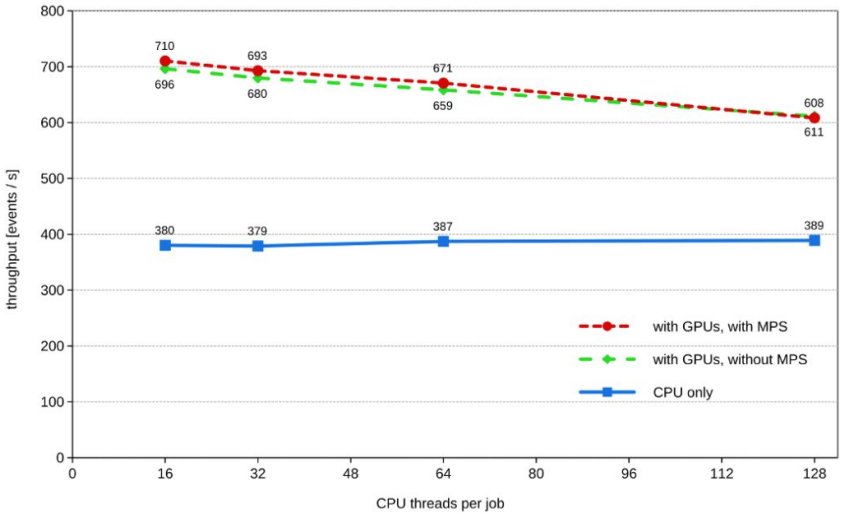
- High Level Trigger throughput, in event per second, measured under different conditions:
  - the different groups of bars correspond to measurements with different number of jobs (16, 8, 4, 2) and threads per job (16, 32, 64, 128), keeping the total number of threads constant at 256;
  - the different colours indicate (blue) jobs running only on the CPU, (green) jobs offloading part of the computation to the GPU, and (red) jobs offloading part of the computation to the GPU and using the NVIDIA Multi-Process Server (MPS) to improve the GPU sharing;
  - the darker shades of colour indicate jobs processing as many concurrent events as CPU threads; the lighter shades indicate jobs processing 3 concurrent events for every 4 CPU threads.
- The configuration used in production has 8 concurrent jobs, each running with 32 CPU threads and 24 concurrent events, and it does not use the NVIDIA MPS.

CMS - HLT tracking and vertexing



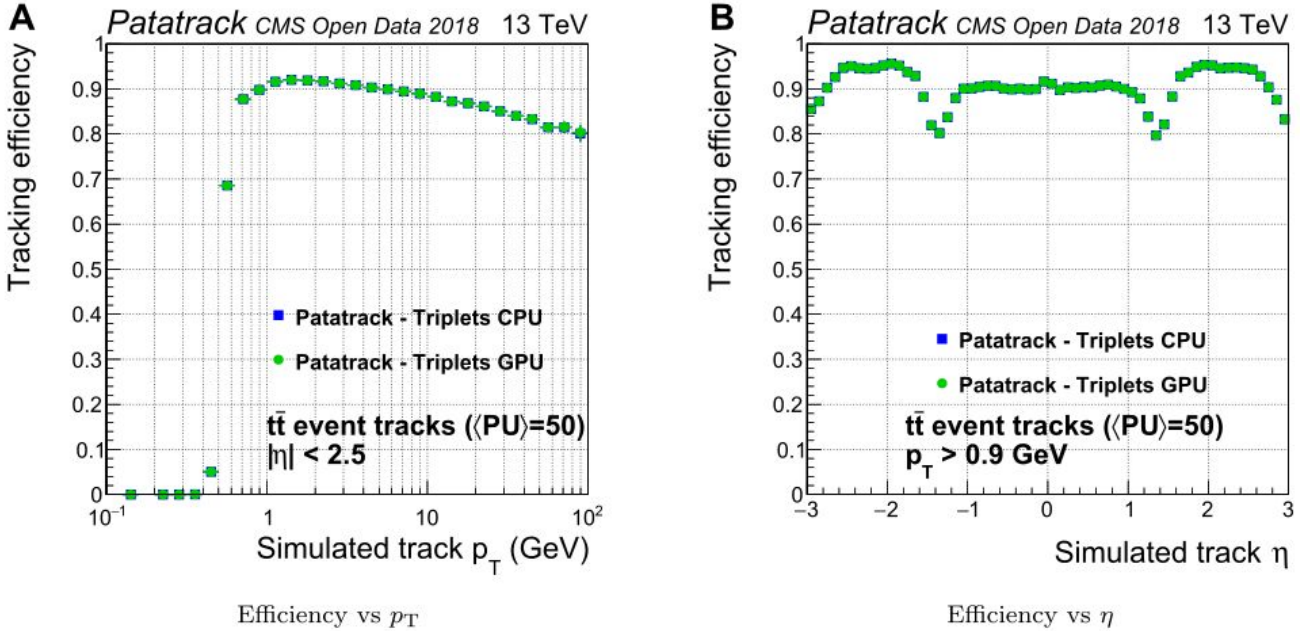
- High Level Trigger throughput, in event per second, measured under different conditions:
  - the different groups of bars correspond to measurements with different number of jobs (16, 8, 4, 2) and threads per job (16, 32, 64, 128), keeping the total number of threads constant at 256;
  - the different colours indicate (blue) jobs running only on the CPU, (green) jobs offloading part of the computation to the GPU, and (red) jobs offloading part of the computation to the GPU and using the NVIDIA Multi-Process Server (MPS) to improve the GPU sharing;
- The configuration used in production has 8 concurrent jobs, each running with 32 CPU threads and 24 concurrent events, and it does not use the NVIDIA MPS.

CMS - HLT tracking and vertexing



- High Level Trigger throughput, in event per second, measured under different conditions:
  - the different points correspond to measurements with different number of jobs (16, 8, 4, 2) and threads per job (16, 32, 64, 128), keeping the total number of threads constant at 256;
  - the different colours indicate (blue) jobs running only on the CPU, (green) jobs offloading part of the computation to the GPU, and (red) jobs offloading part of the computation to the GPU and using the NVIDIA Multi-Process Server (MPS) to improve the GPU sharing;
- The configuration used in production has 8 concurrent jobs, each running with 32 CPU threads and 24 concurrent events, and it does not use the NVIDIA MPS.

PATATRACK

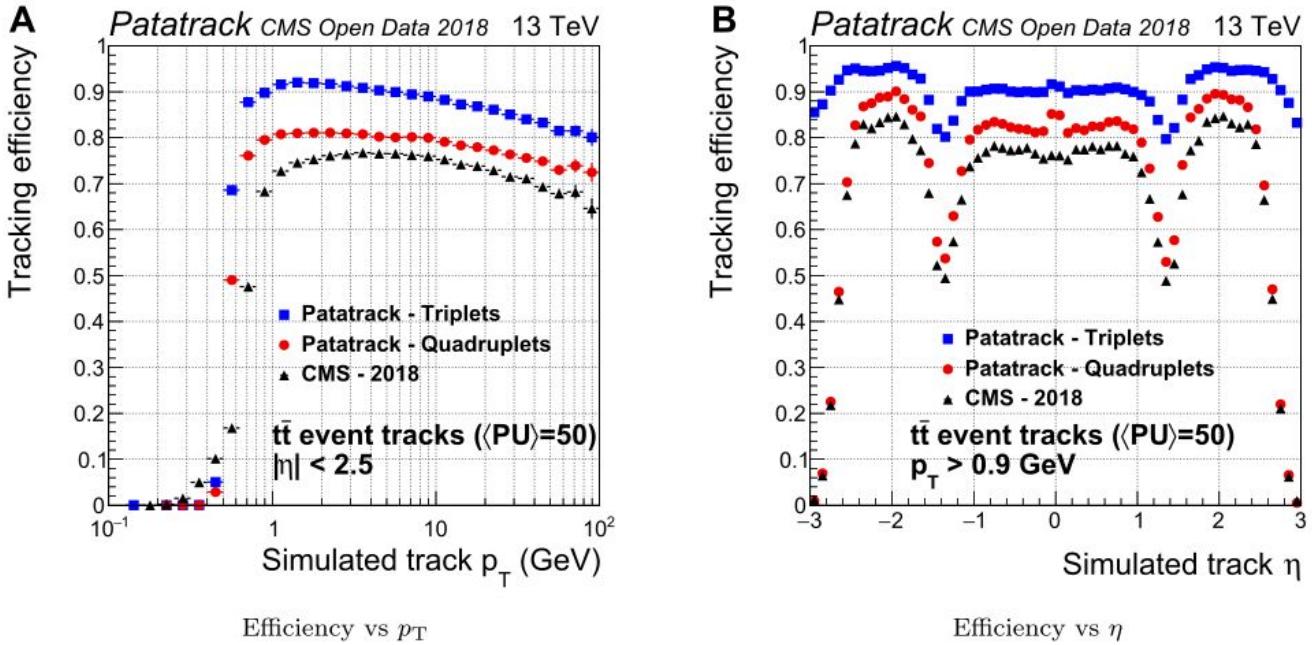


Efficiency vs p_T

Efficiency vs η

FIGURE 6 | Comparison of the pixel tracks reconstruction efficiency of the CPU and GPU versions of the Patatrack pixel reconstruction for simulated $t\bar{t}$ events with an average of 50 superimposed pileup collisions.

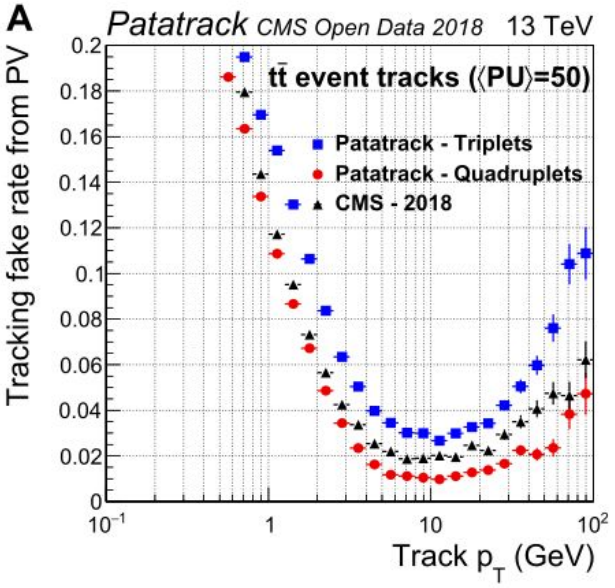
PATATRACK



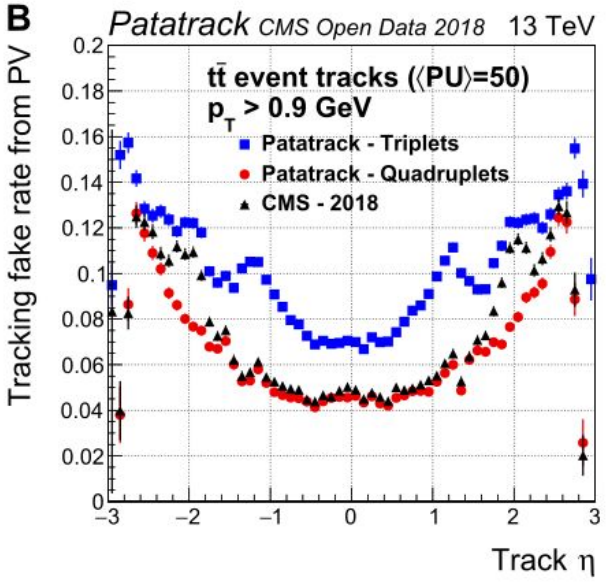
Efficiency vs p_T Efficiency vs η

FIGURE 7 | Pixel tracks reconstruction efficiency for simulated $t\bar{t}$ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from *n*-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



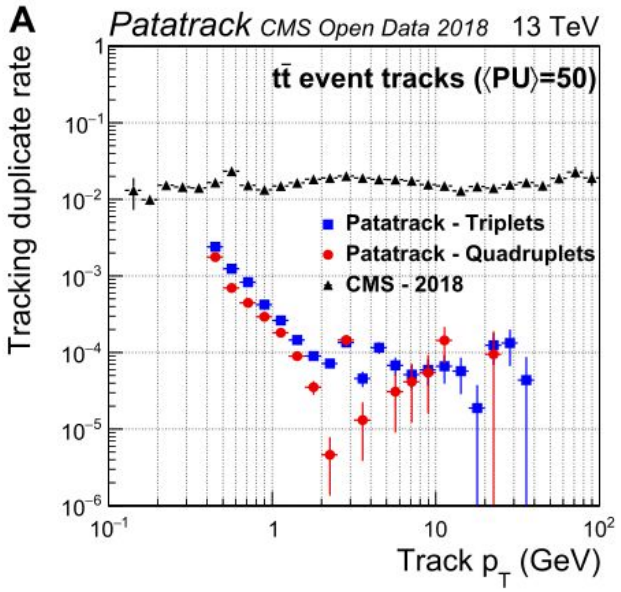
Fake rate vs p_T



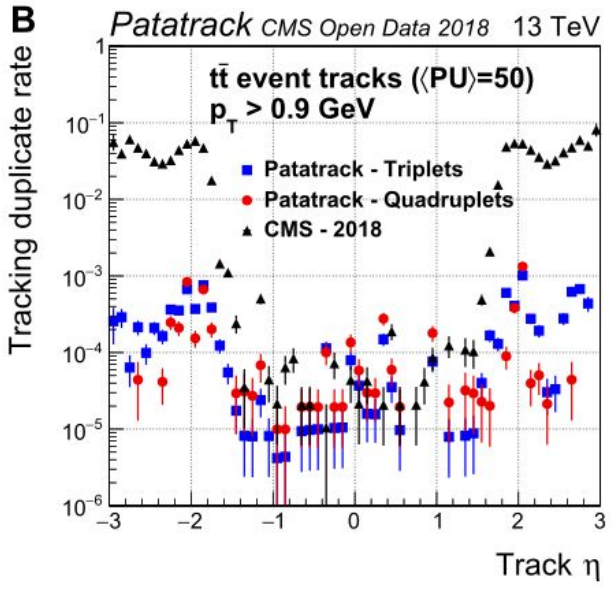
Fake rate vs η

FIGURE 8 | Pixel tracks reconstruction fake rate for simulated $t\bar{t}$ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



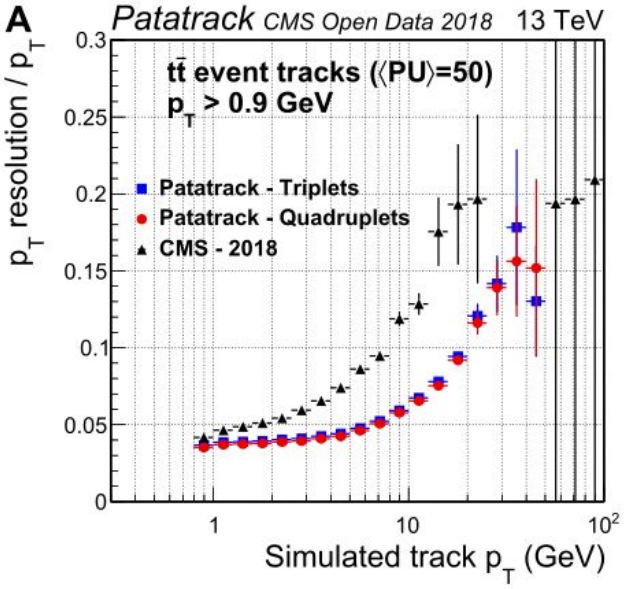
Duplicate rate vs p_T



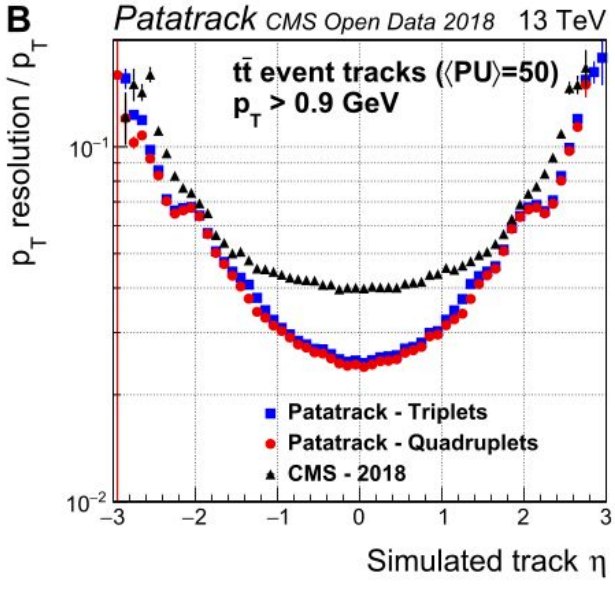
Duplicate rate vs η

FIGURE 9 | Pixel tracks reconstruction duplicate rate for simulated tt̄ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



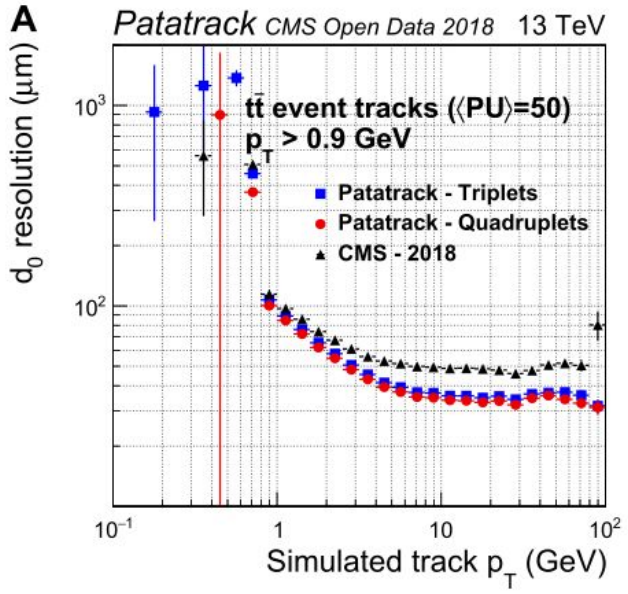
p_T resolution vs p_T



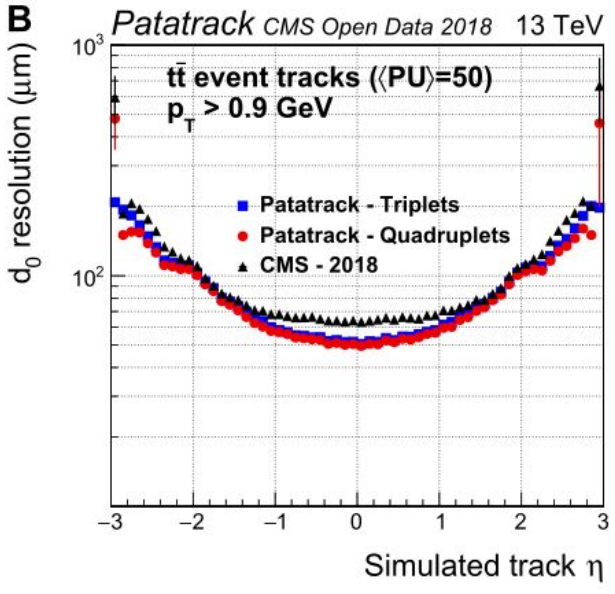
p_T resolution vs η

FIGURE 10 | Pixel tracks p_T resolution for simulated tt̄ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



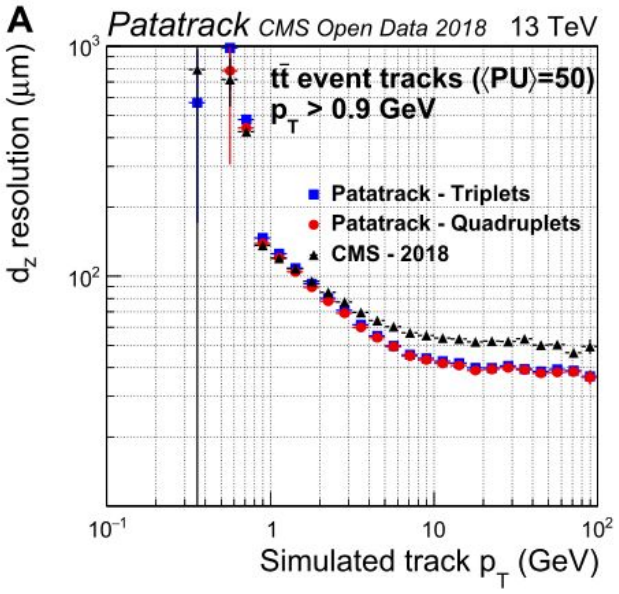
d_{xy} resolution vs p_T



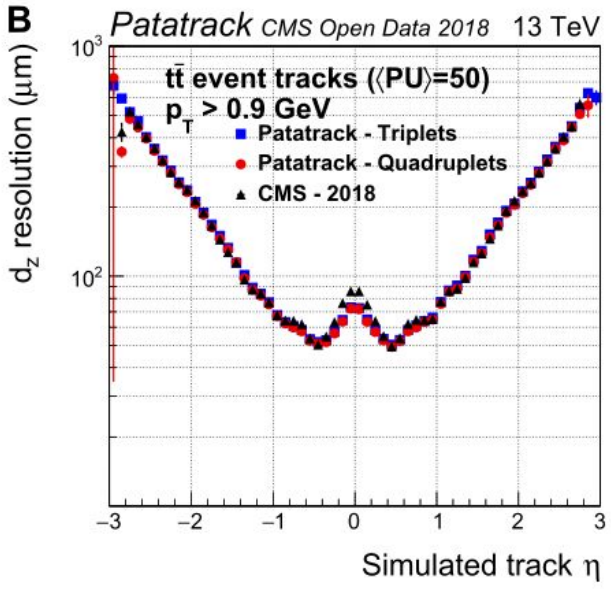
d_{xy} resolution vs η

FIGURE 11 | Pixel tracks transverse impact parameter resolution for simulated $t\bar{t}$ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



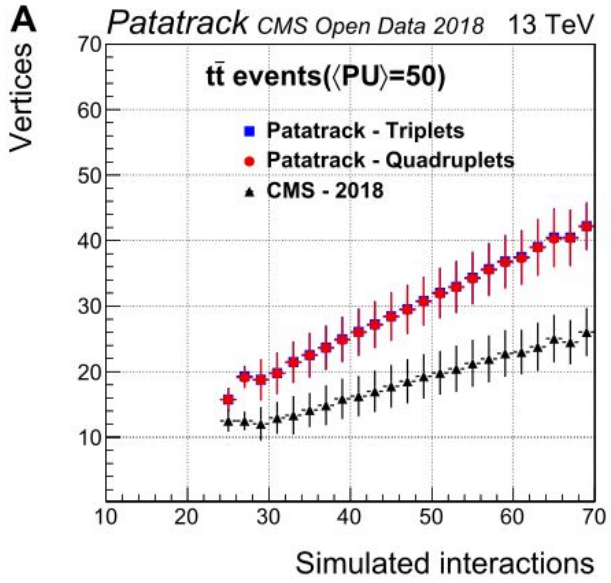
d_z resolution vs p_T



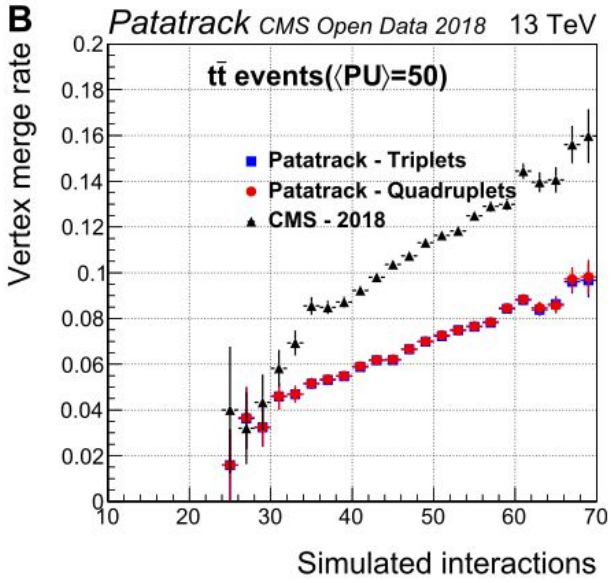
d_z resolution vs η

FIGURE 12 | Pixel tracks longitudinal impact parameter resolution for simulated tt̄ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{\text{hits}} \geq 3$ and $n_{\text{hits}} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

PATATRACK



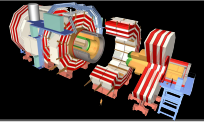
Reconstructed vs simulated vertices



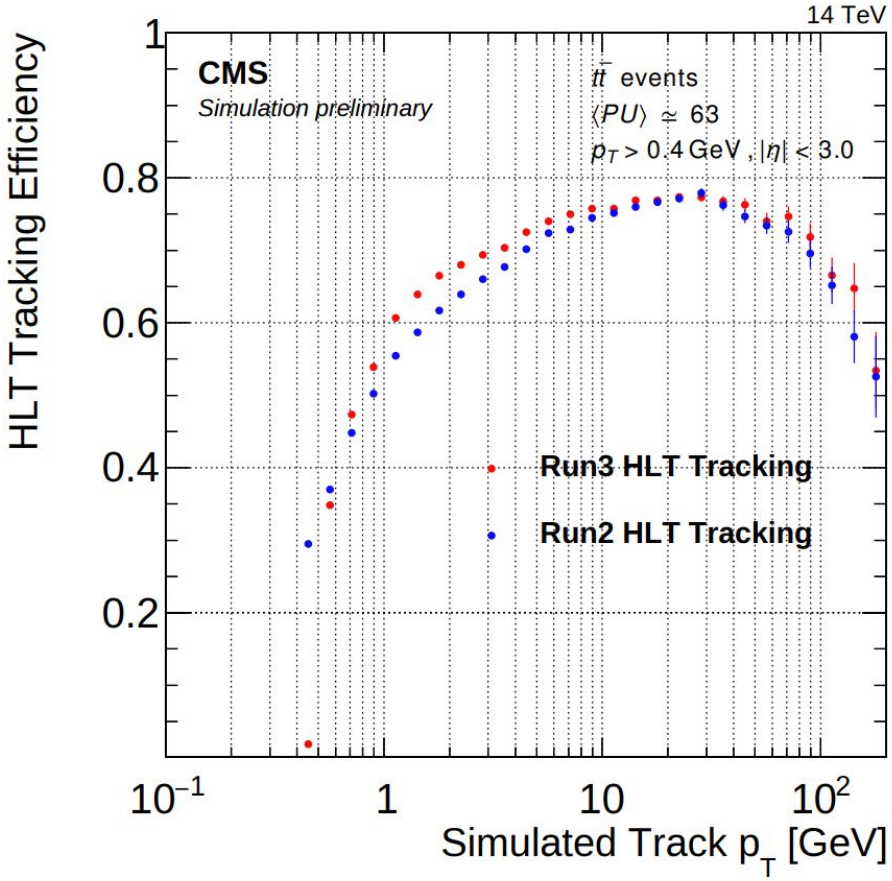
Vertex merge rate vs simulated vertices

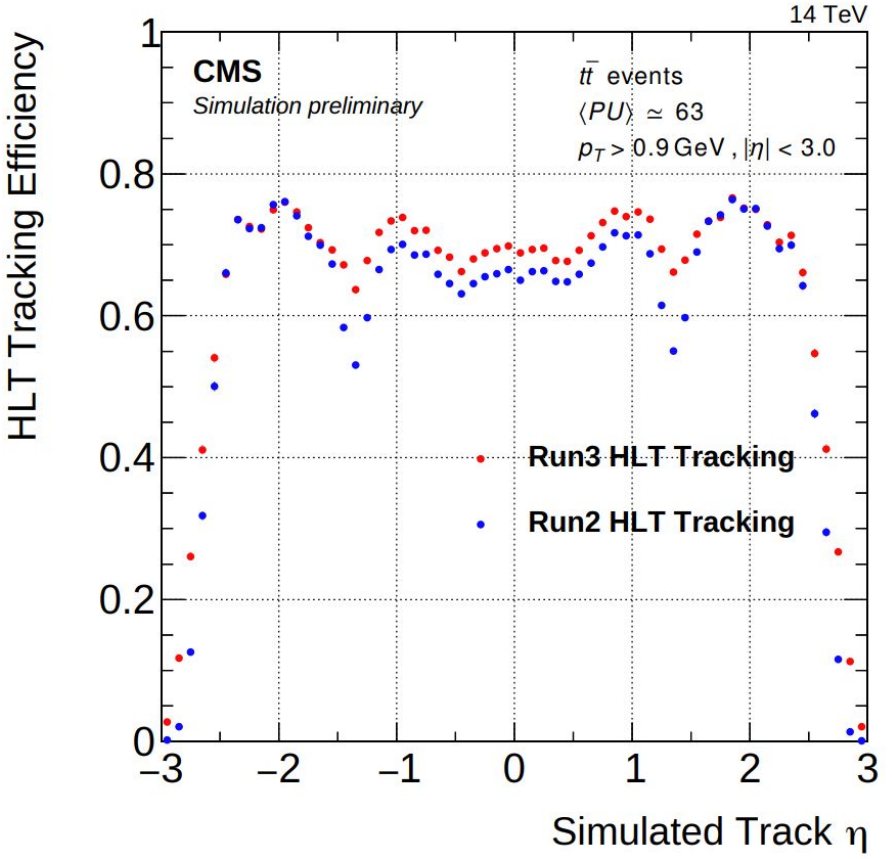
FIGURE 13 | Pixel vertices reconstruction efficiency and merge rate for simulated tt̄ events with an average of 50 superimposed pileup collisions. The performance of the Patatrack reconstruction when producing pixel tracks starting from n-tuplets with $n_{hits} \geq 3$ and $n_{hits} \geq 4$ are represented respectively by blue squares and red circles. The performance of CMS-2018 is represented by black triangles.

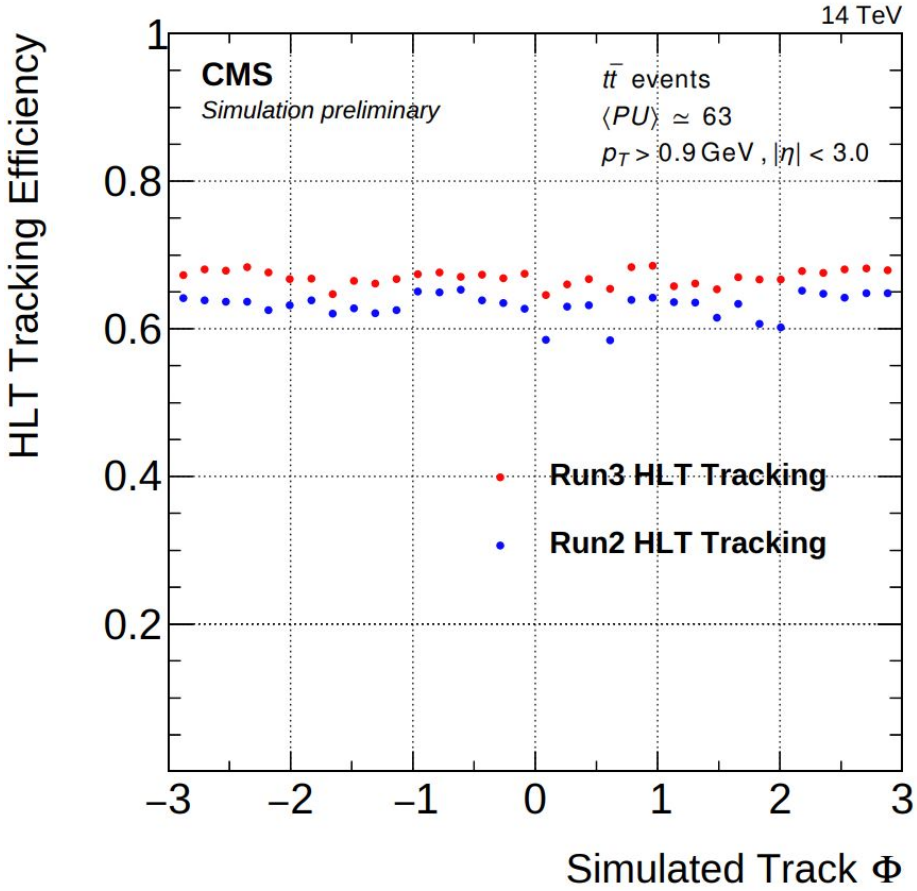
CMS - HLT tracking and vertexing

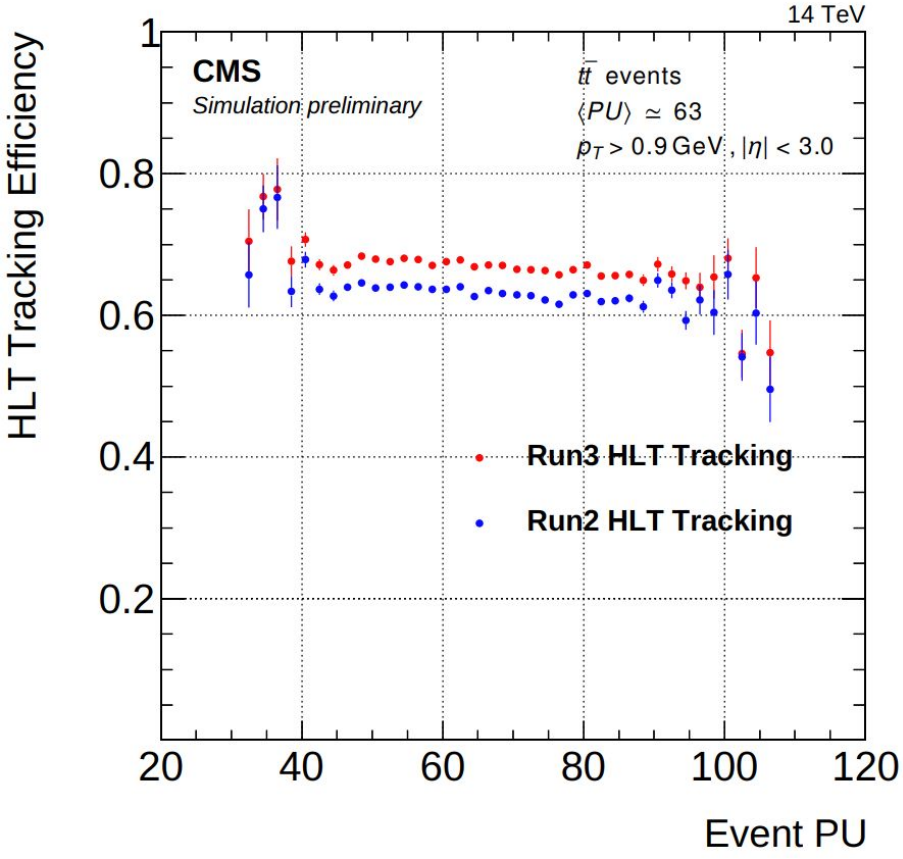


mfaggin@cern.ch



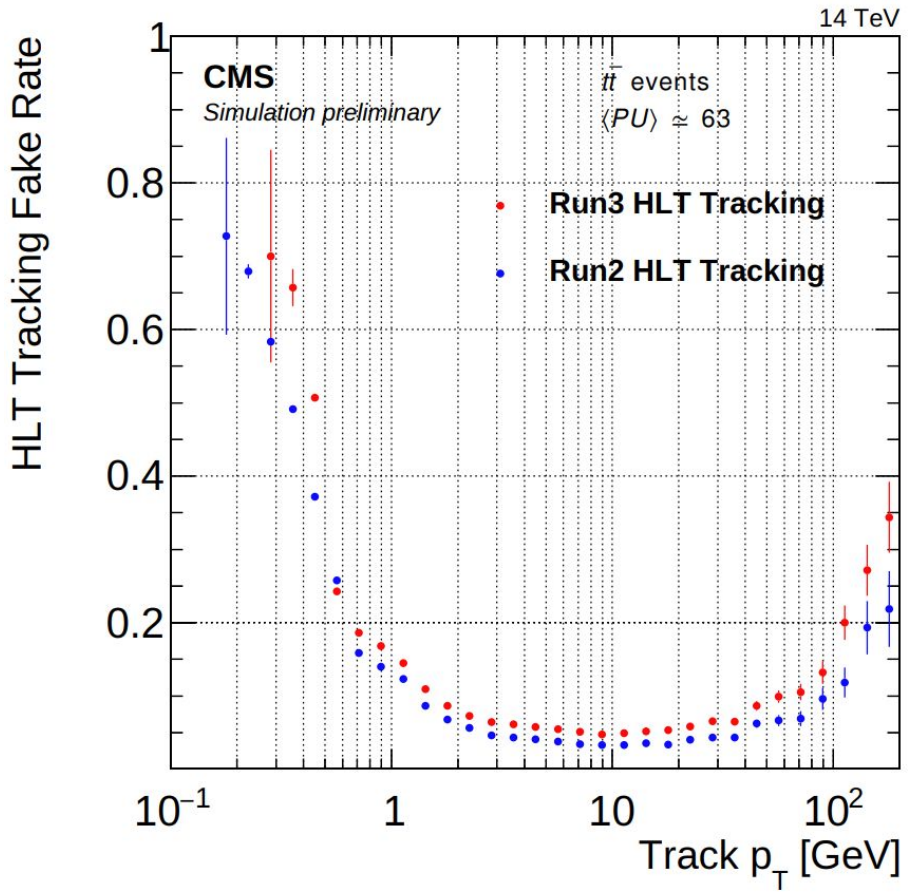


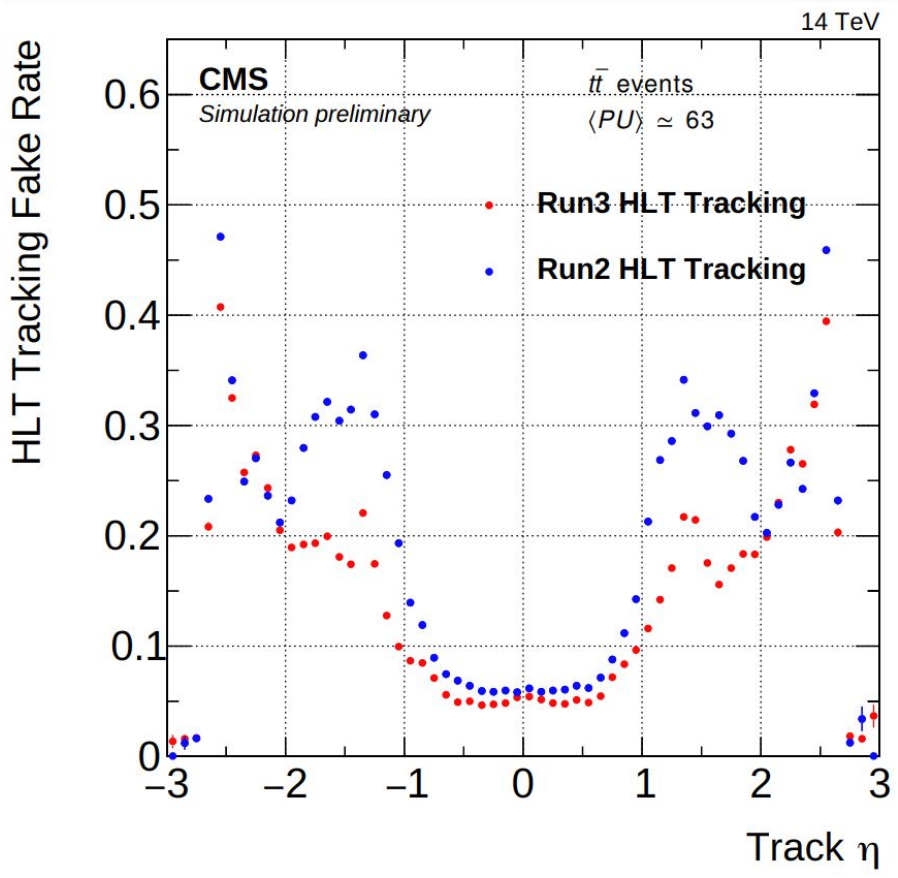

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
$p_T > 0.9\,\text{GeV}$, $|\eta| < 3.0$
Run3 HLT Tracking
Run2 HLT Tracking
HLT Tracking Efficiency
Simulated Track Φ
1
0.8
0.6
0.4
0.2
−3
−2
−1
0
1
2
3

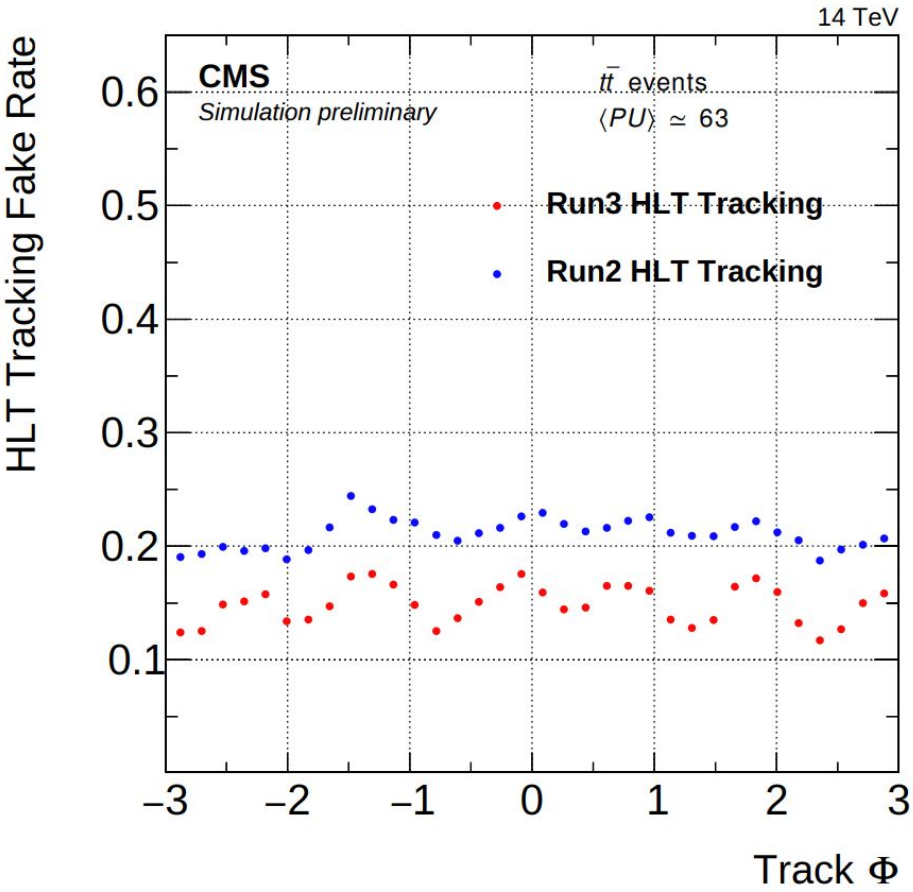

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
$p_T > 0.9\,\text{GeV}$, $|\eta| < 3.0$
Run3 HLT Tracking
Run2 HLT Tracking
HLT Tracking Efficiency
Event PU
1
0.8
0.6
0.4
0.2
20
40
60
80
100
120

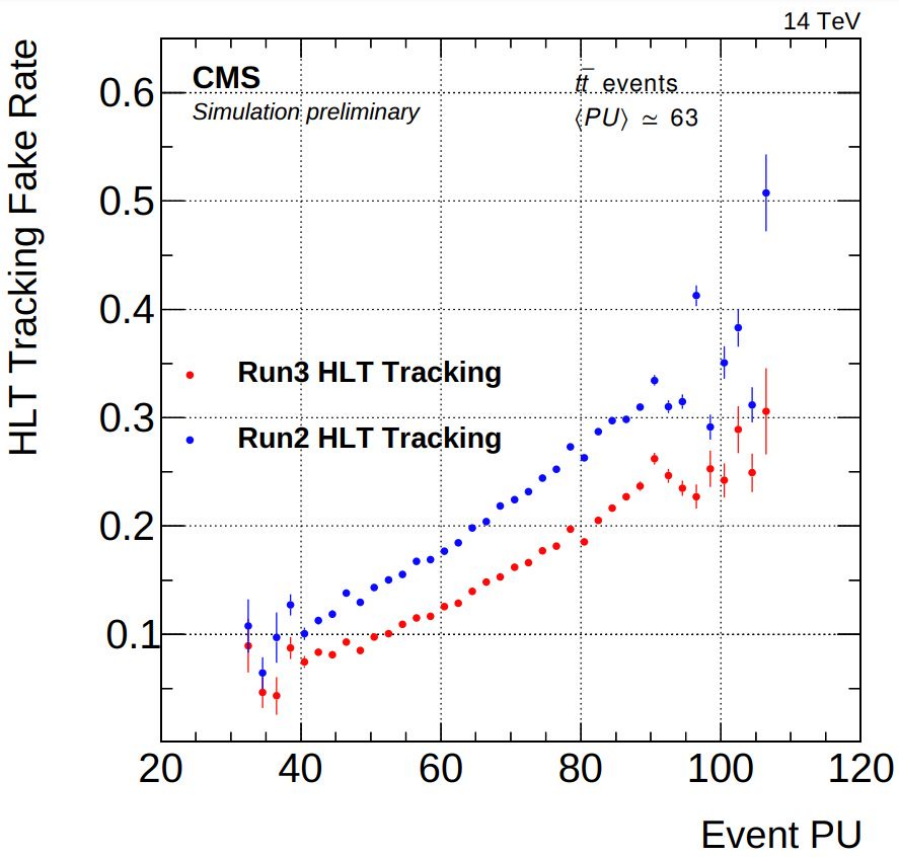
CMS - HLT tracking and vertexing

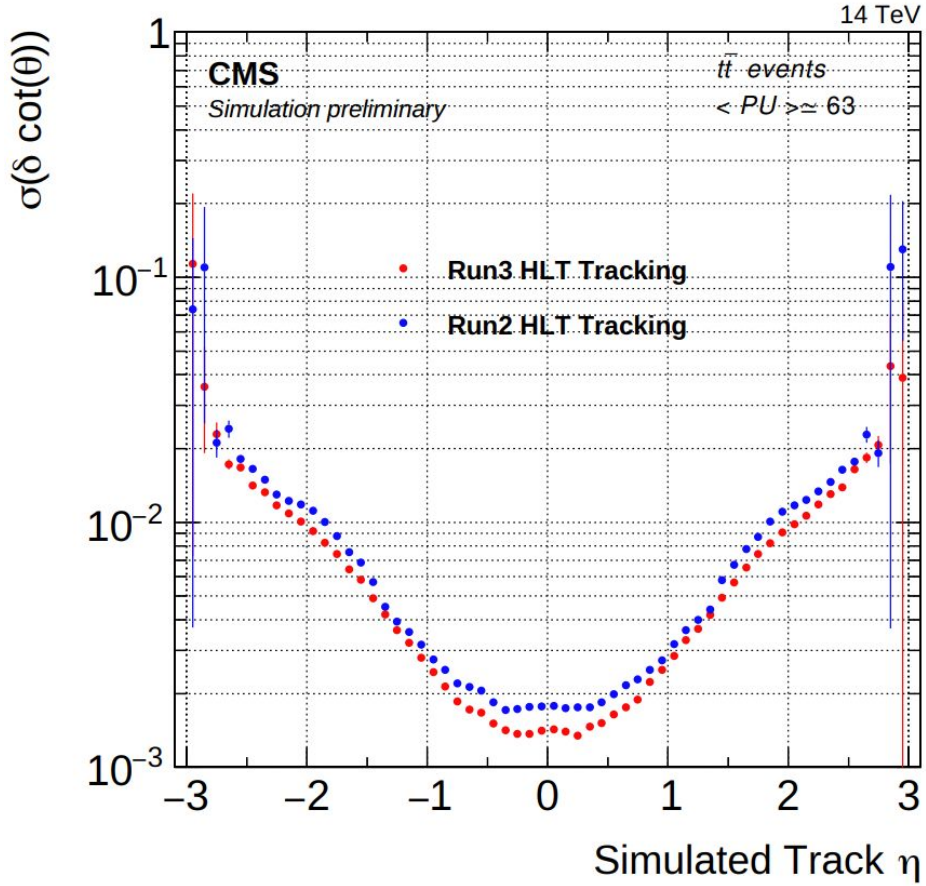
mfaggin@cern.ch

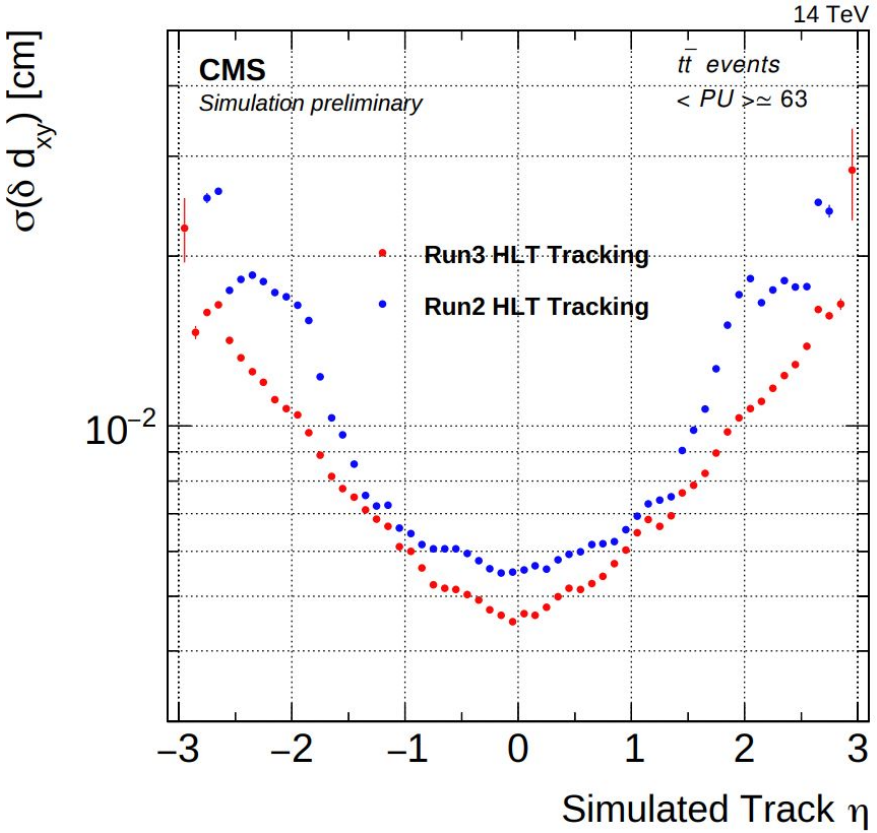




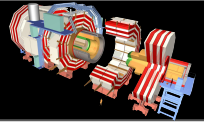

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
Run3 HLT Tracking
Run2 HLT Tracking
HLT Tracking Fake Rate
0.6
0.5
0.4
0.3
0.2
0.1
−3
−2
−1
0
1
2
3
Track Φ


14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
Run3 HLT Tracking
Run2 HLT Tracking
HLT Tracking Fake Rate
0.6
0.5
0.4
0.3
0.2
0.1
20
40
60
80
100
120
Event PU

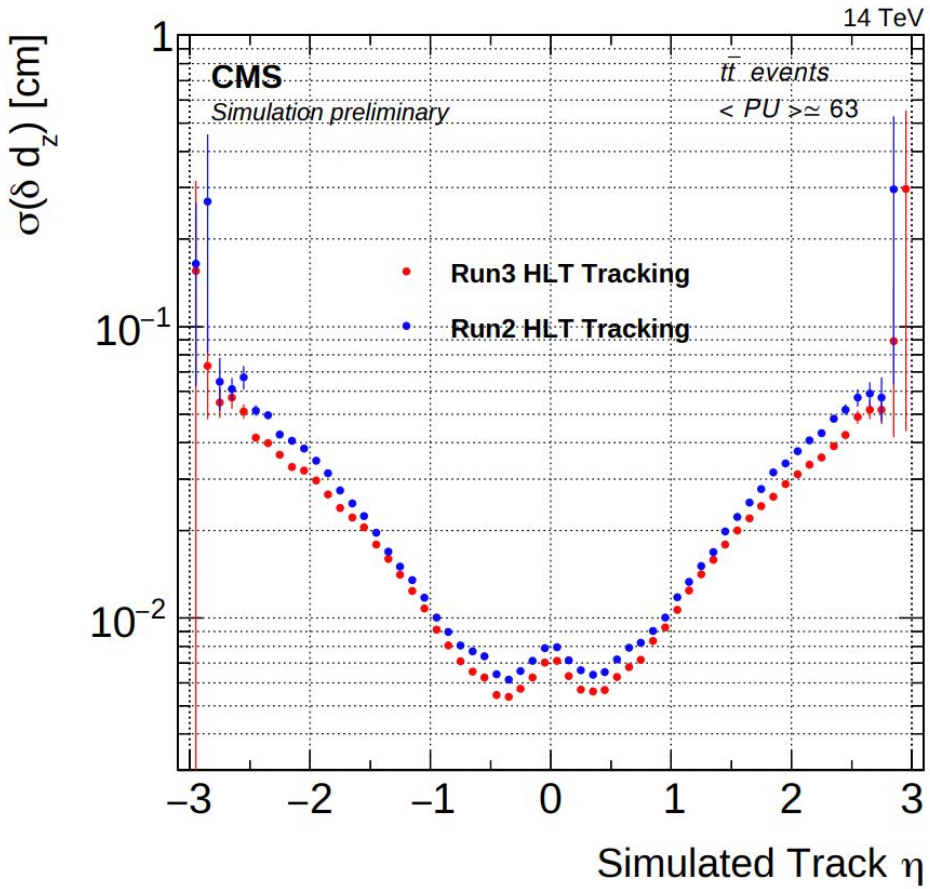

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
< PU >≃ 63
Run3 HLT Tracking
Run2 HLT Tracking
σ(δ cot(θ))
1
10^{-1}
10^{-2}
10^{-3}
−3 −2 −1 0 1 2 3
Simulated Track η

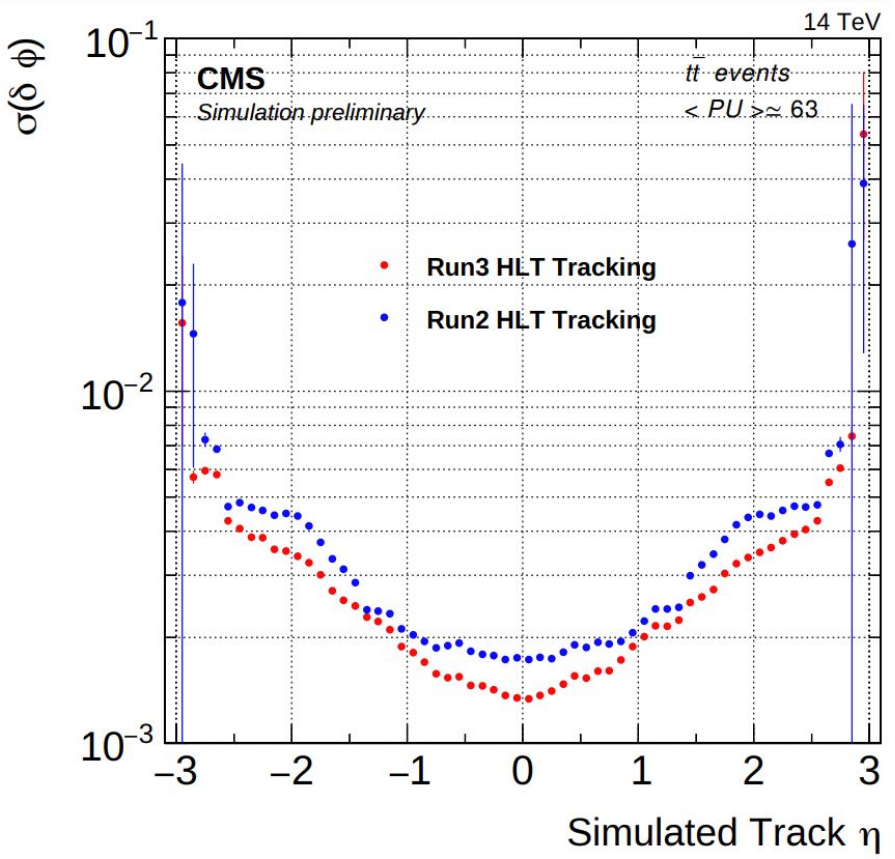

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
< PU >≃ 63
Run3 HLT Tracking
Run2 HLT Tracking
σ(δ d_{xy}) [cm]
10^{-2}
−3 −2 −1 0 1 2 3
Simulated Track η

CMS - HLT tracking and vertexing

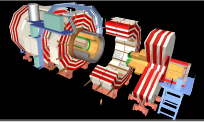


mfaggin@cern.ch

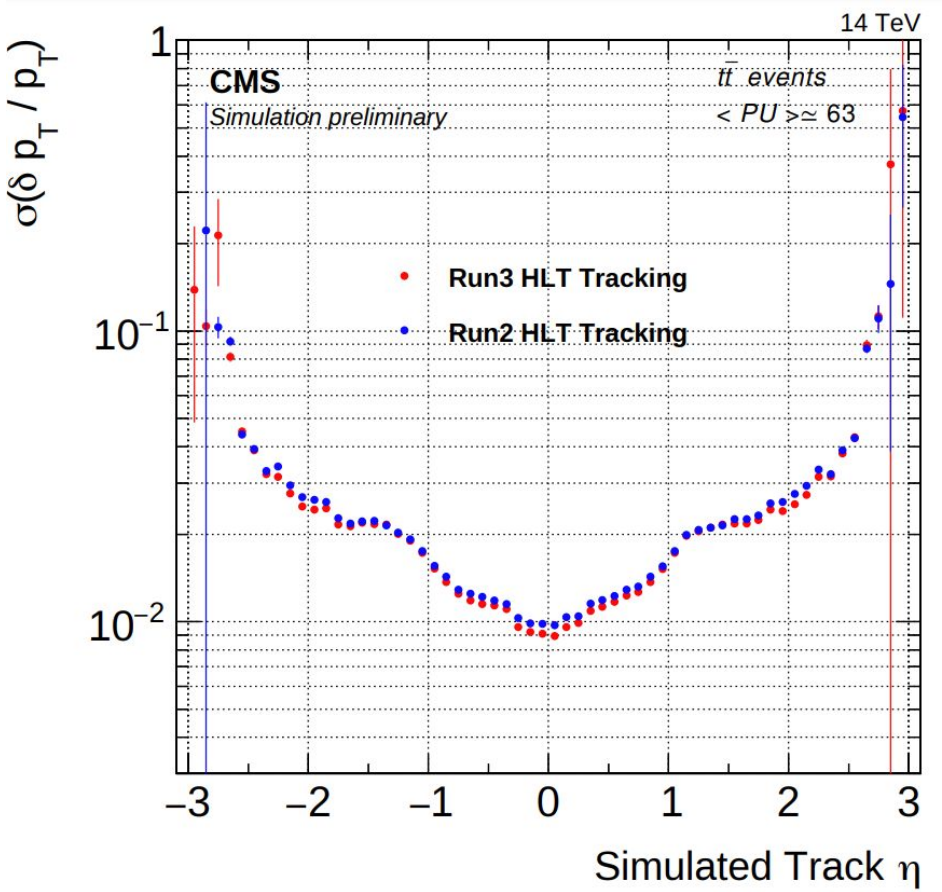


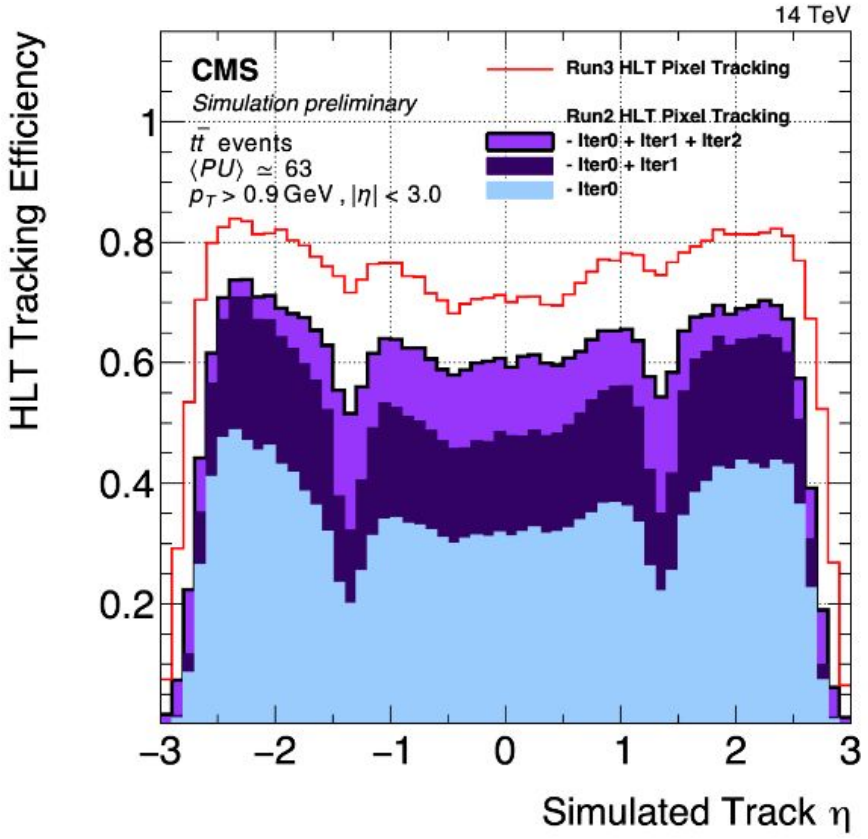


CMS - HLT tracking and vertexing

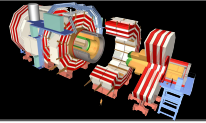


mfaggin@cern.ch

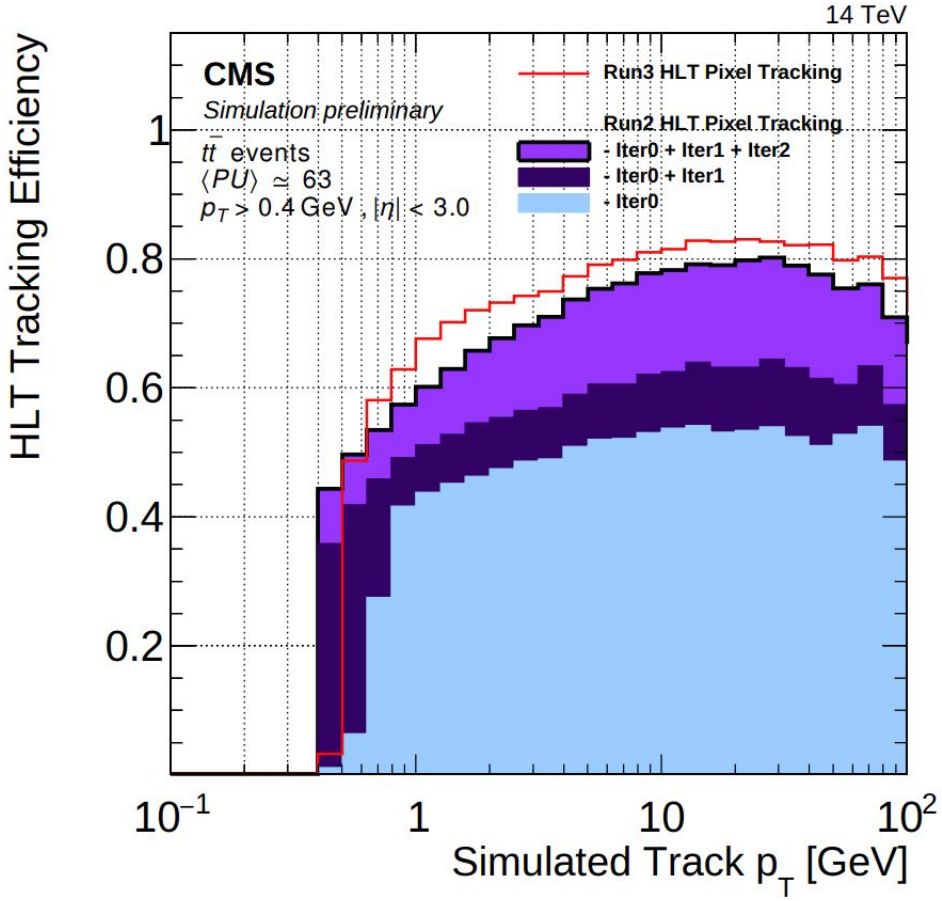


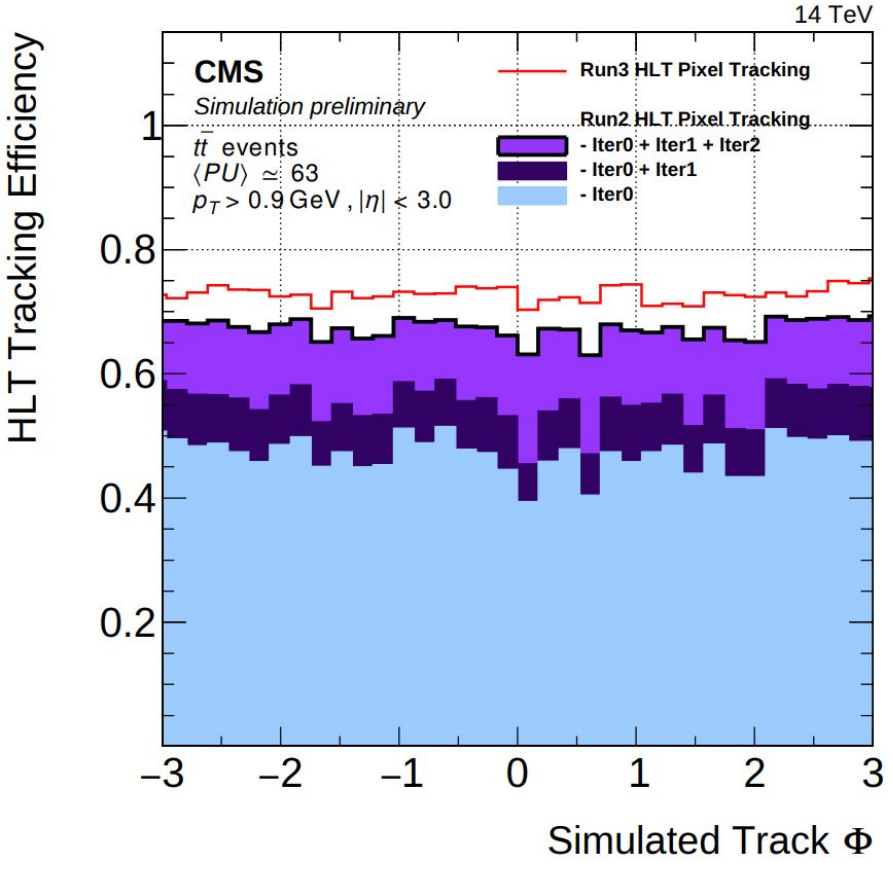


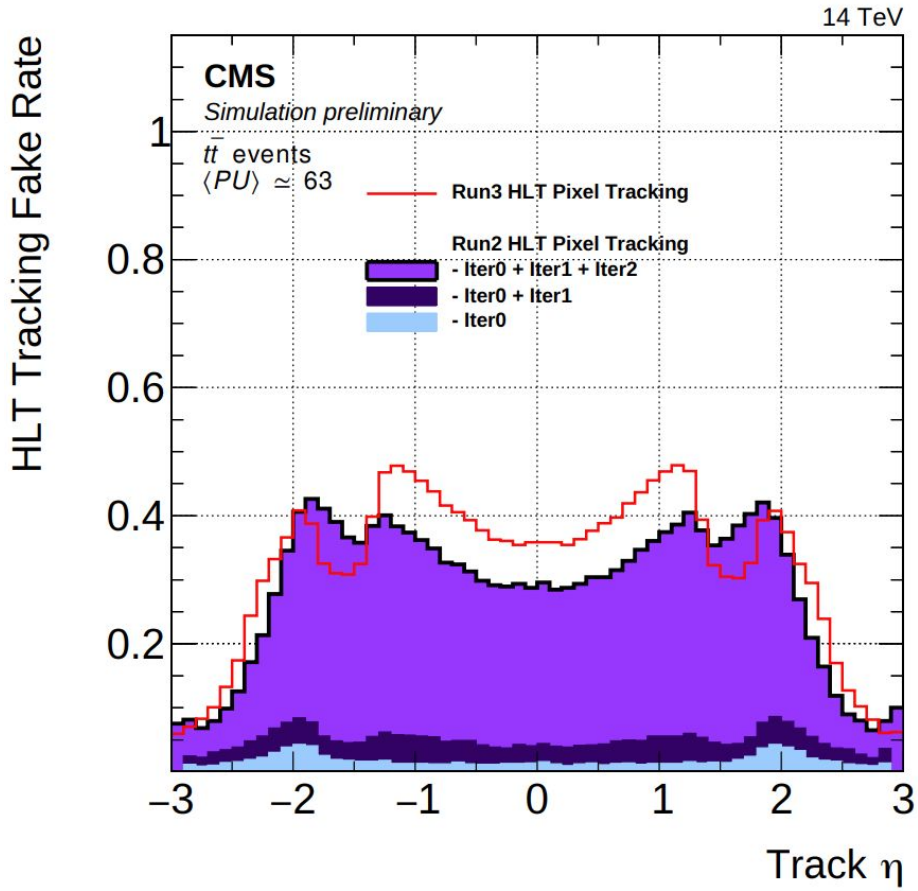
CMS - HLT tracking and vertexing

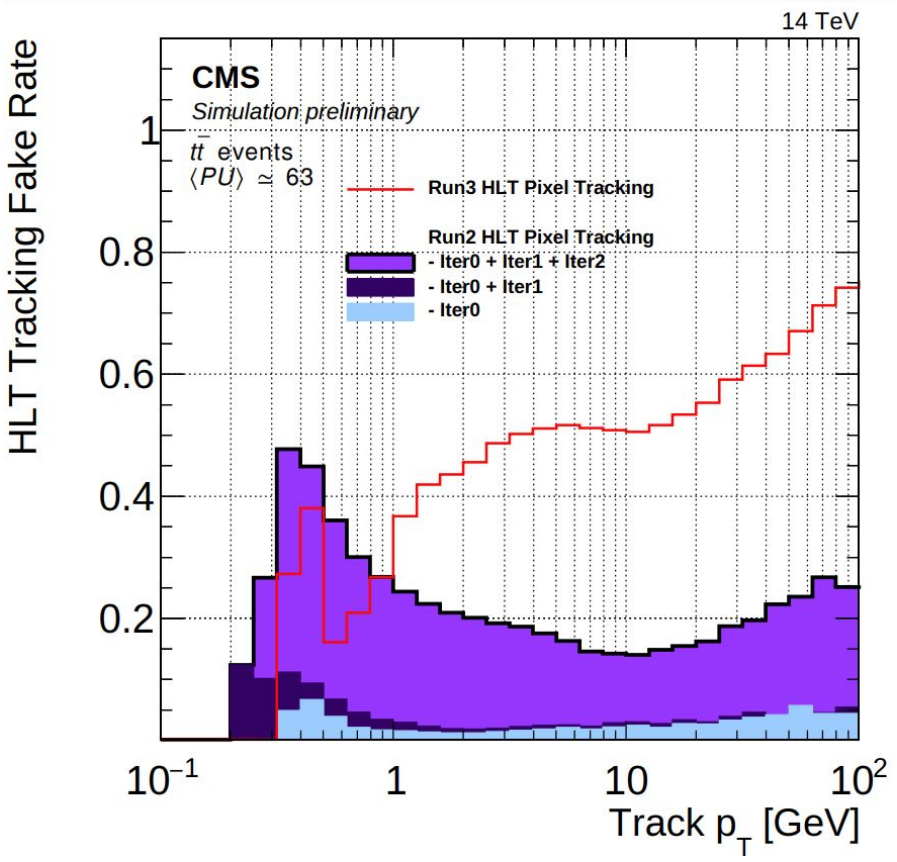


mfaggin@cern.ch

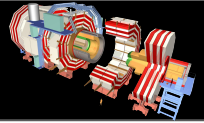




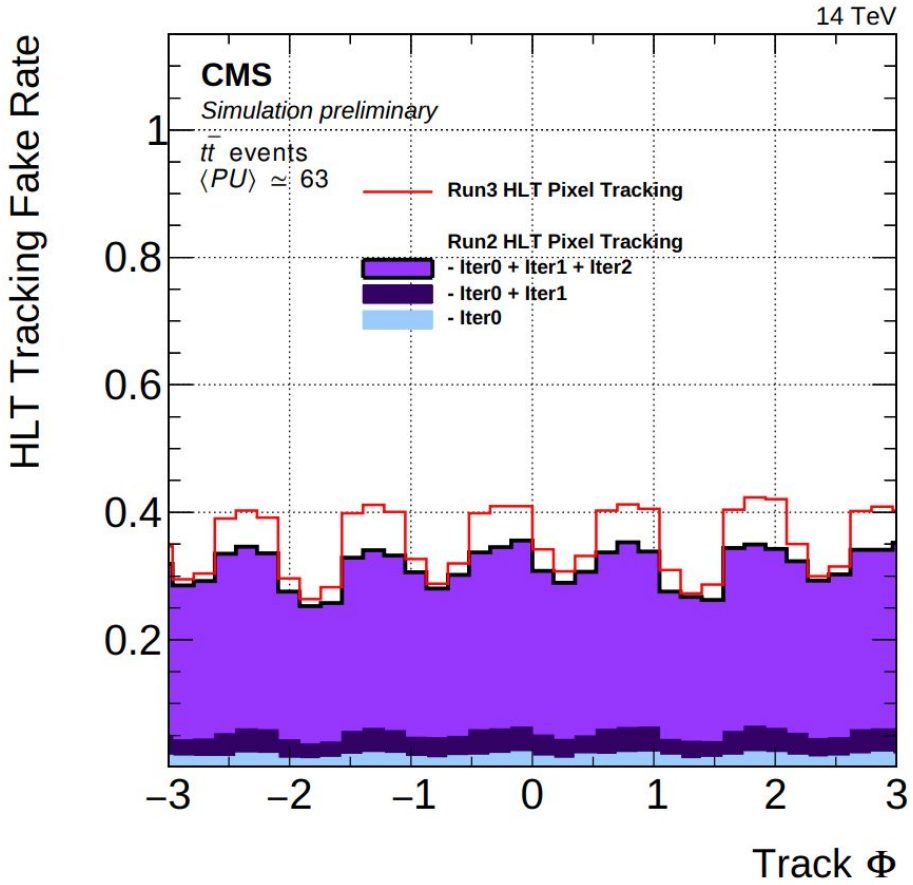

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
Run3 HLT Pixel Tracking
Run2 HLT Pixel Tracking
- Iter0 + Iter1 + Iter2
- Iter0 + Iter1
- Iter0
HLT Tracking Fake Rate
Track η

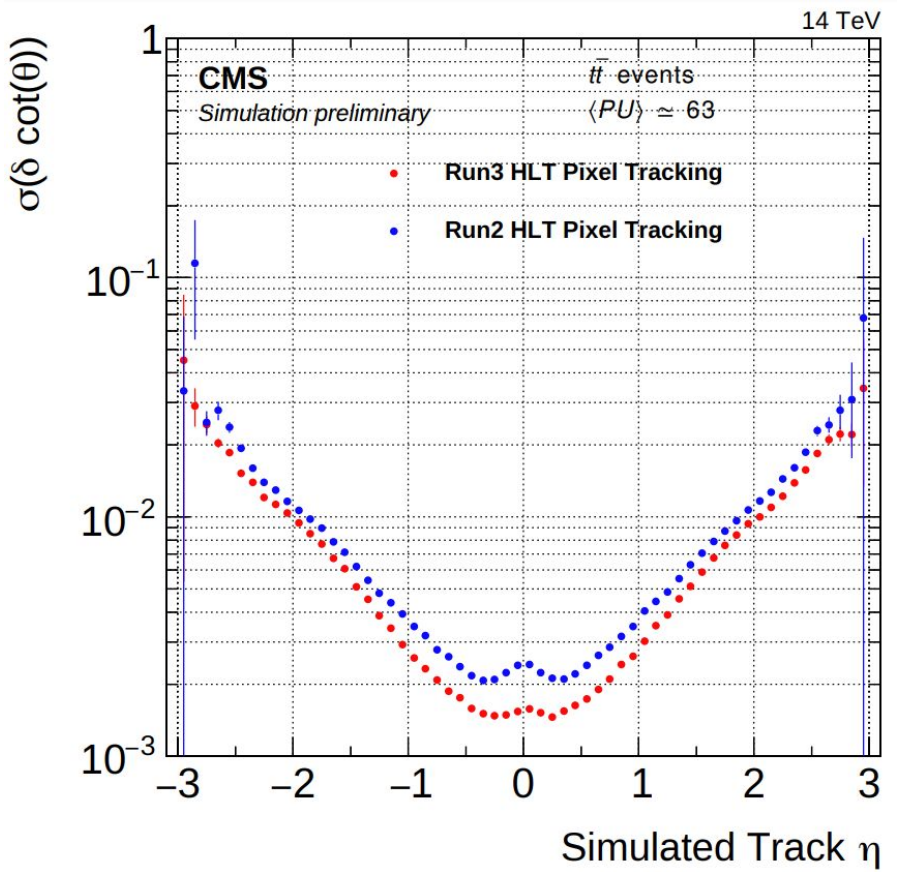

14 TeV
CMS
Simulation preliminary
$t\bar{t}$ events
$\langle PU \rangle \simeq 63$
Run3 HLT Pixel Tracking
Run2 HLT Pixel Tracking
- Iter0 + Iter1 + Iter2
- Iter0 + Iter1
- Iter0
HLT Tracking Fake Rate
Track p_T [GeV]

CMS - HLT tracking and vertexing

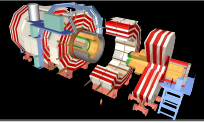


mfaggin@cern.ch

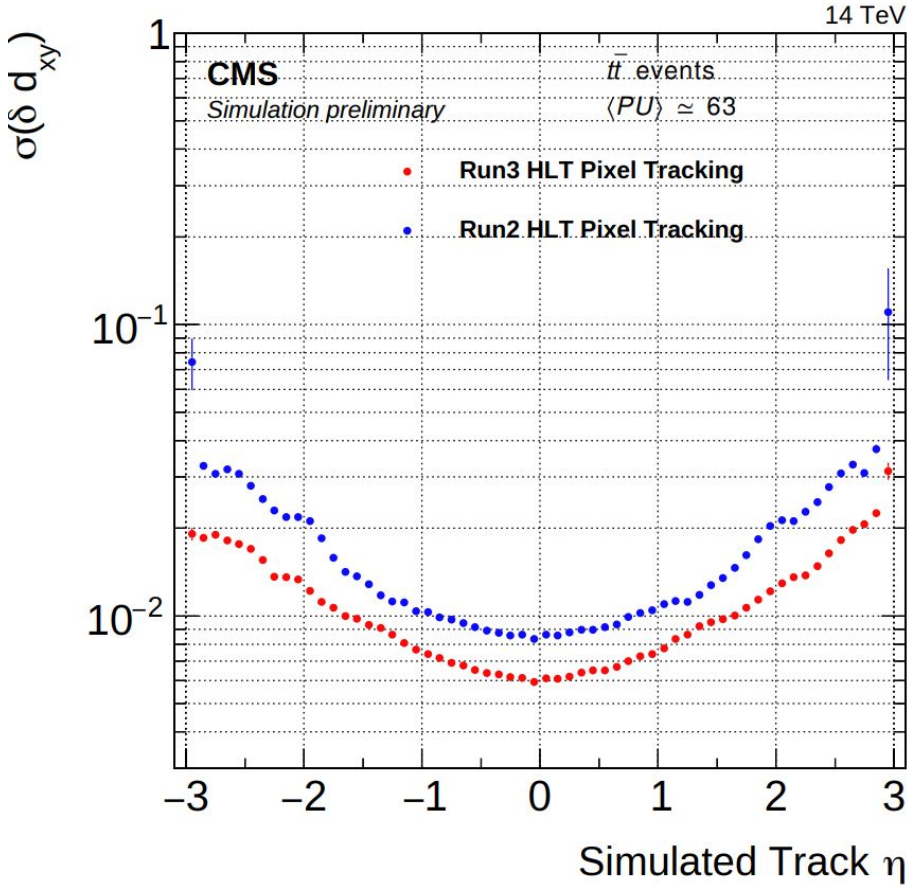


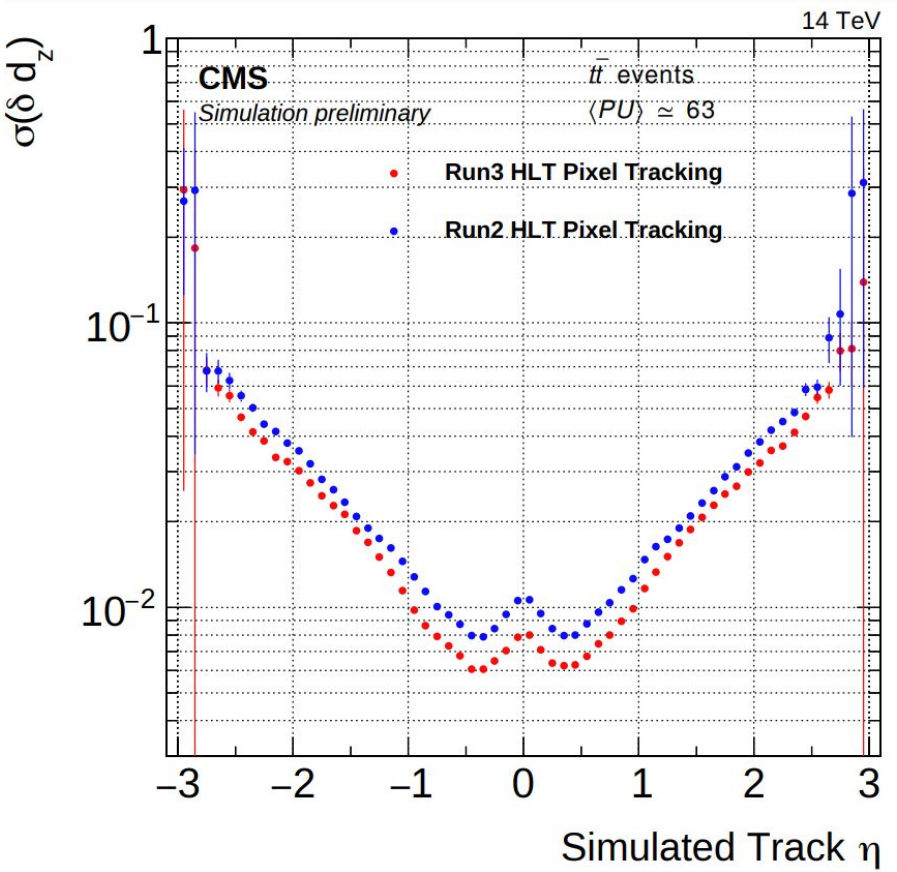


CMS - HLT tracking and vertexing

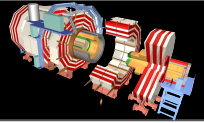


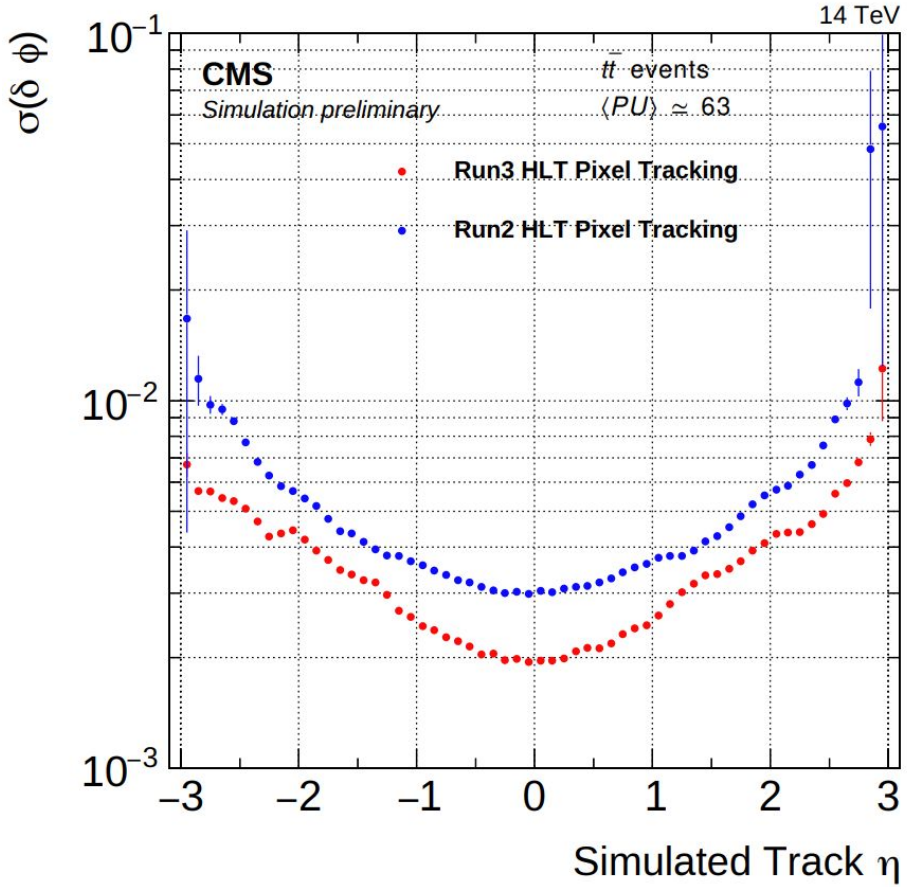
mfaggin@cern.ch

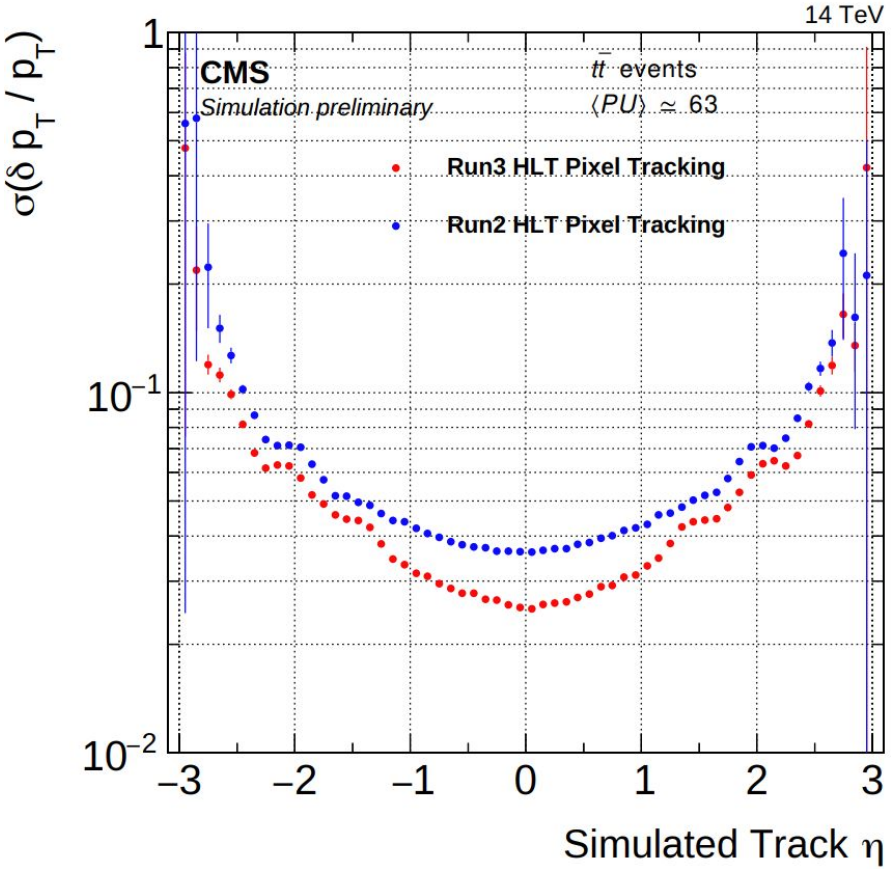




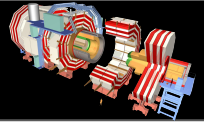
CMS - HLT tracking and vertexing



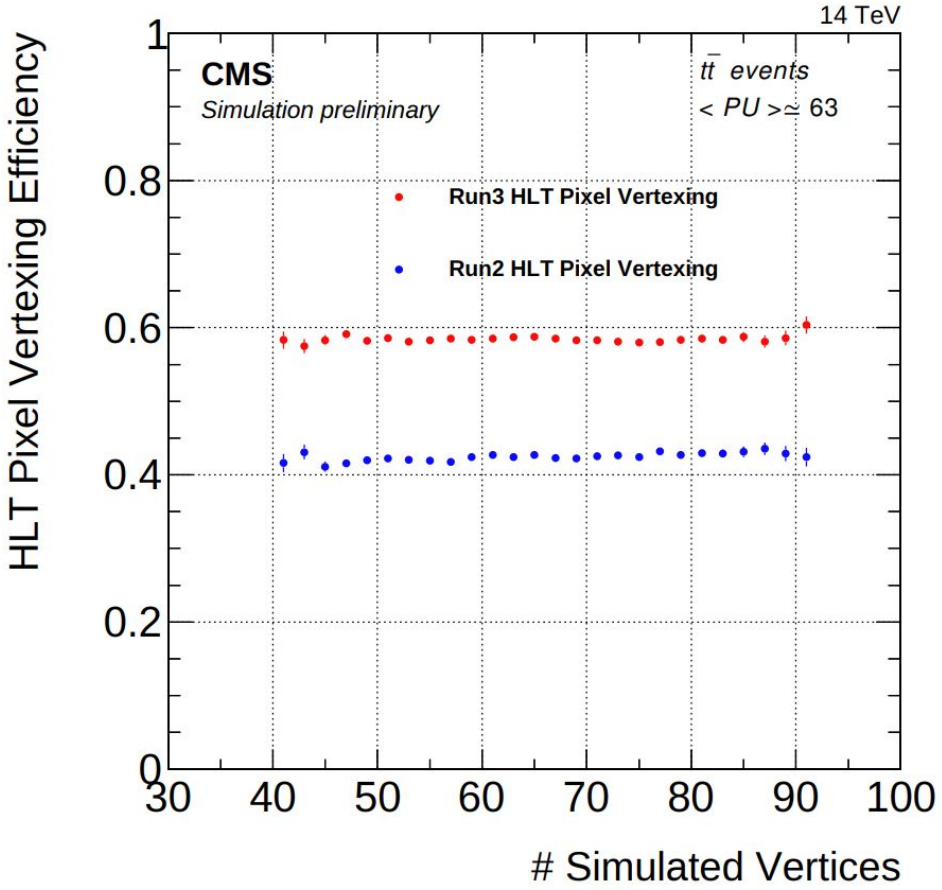


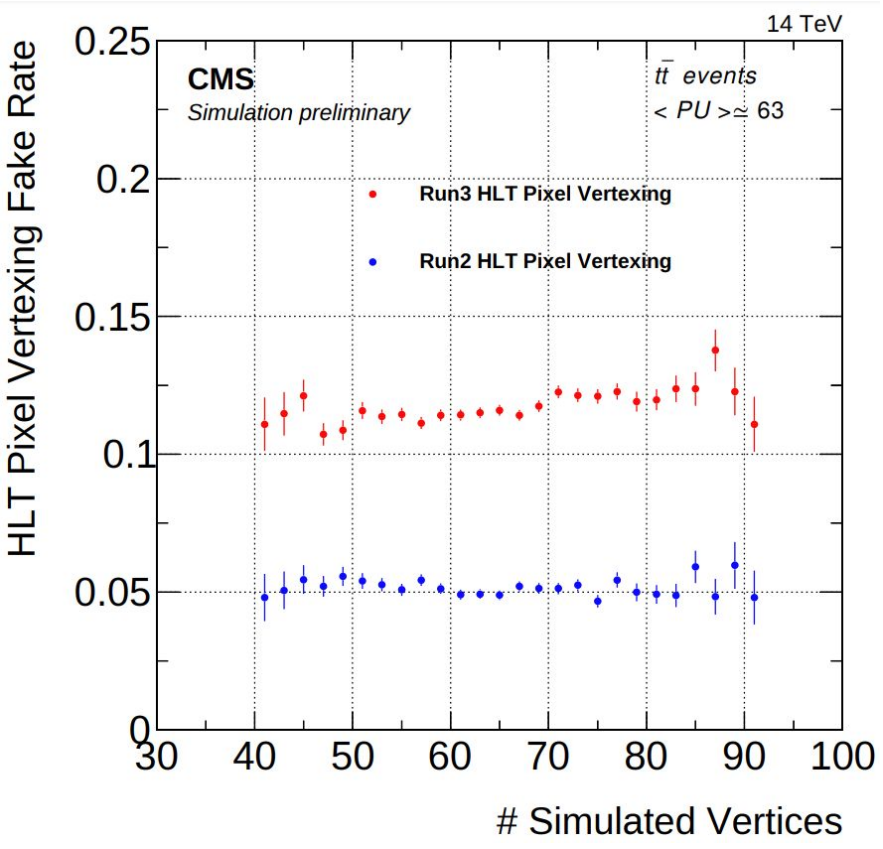


CMS - HLT tracking and vertexing

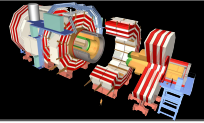


mfaggin@cern.ch

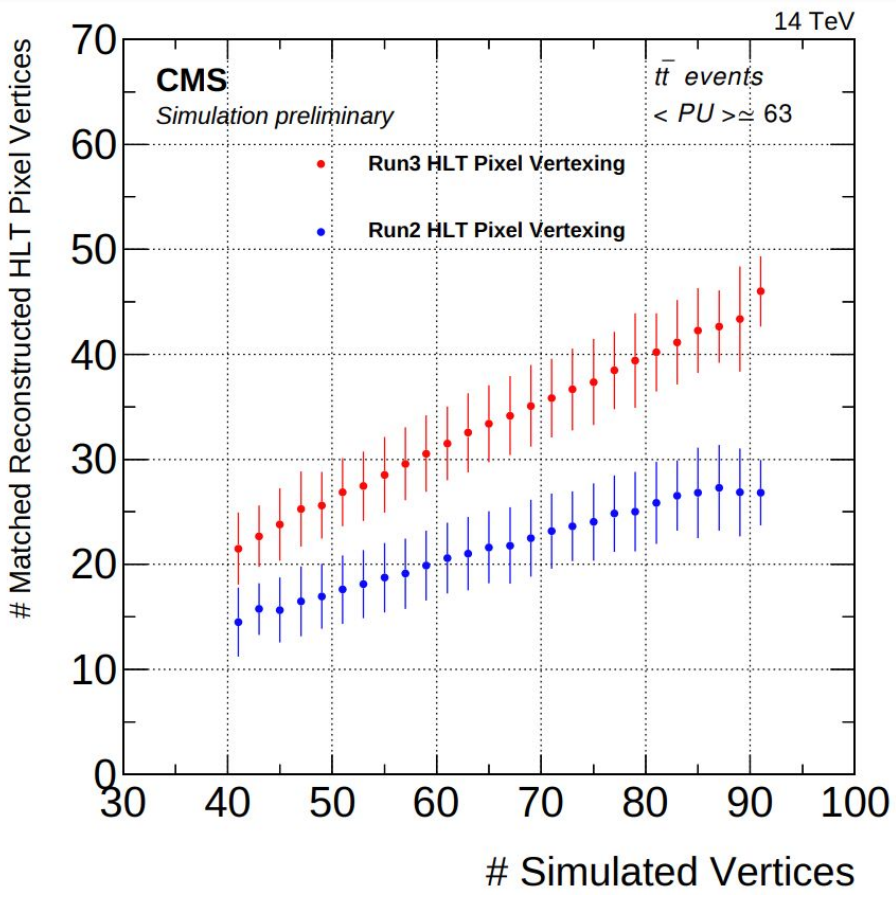




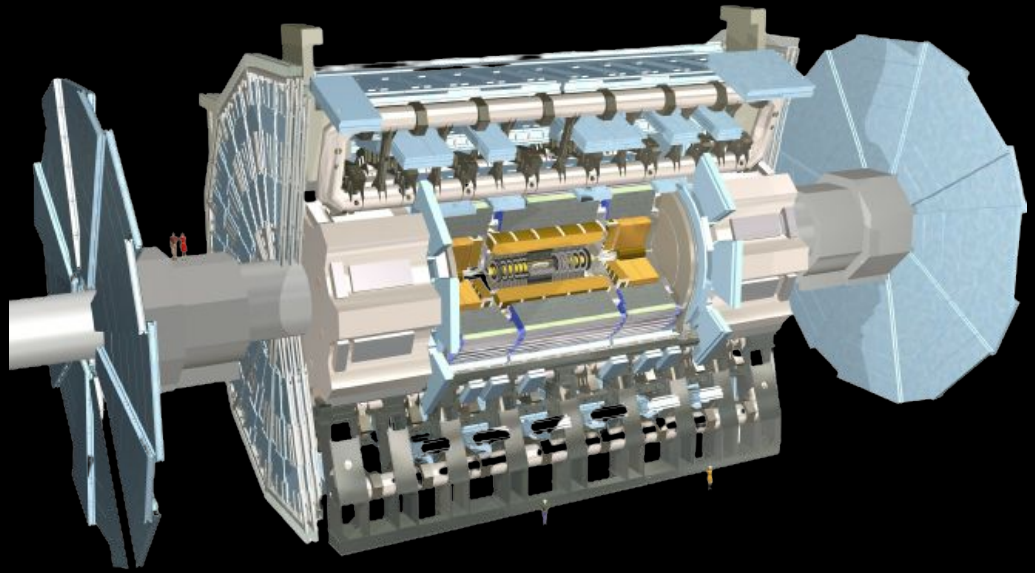
CMS - HLT tracking and vertexing



mfaggin@cern.ch



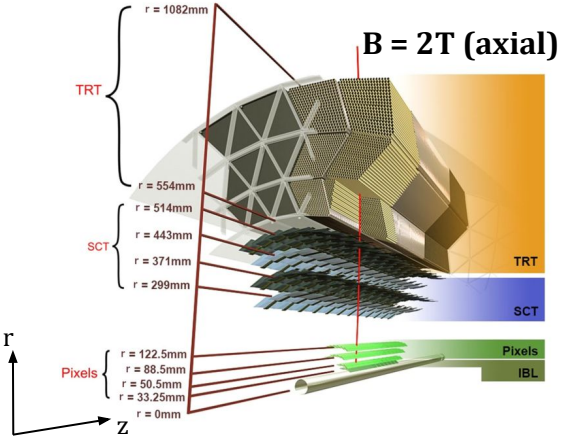
ATLAS

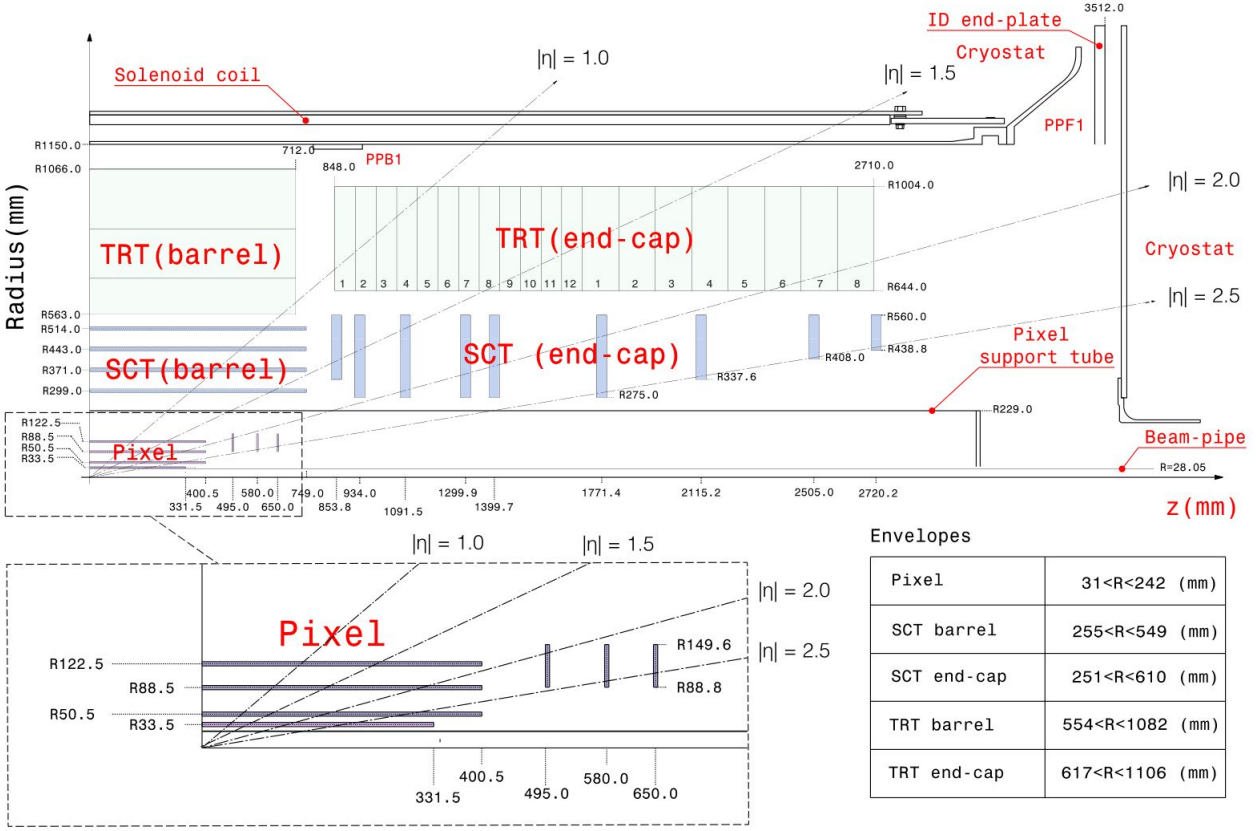


ATLAS - ID tracking system

mfaggin@cern.ch

Inner Detector (ID) tracking system





Envelopes

Pixel	31<R<242 (mm)
SCT barrel	255<R<549 (mm)
SCT end-cap	251<R<610 (mm)
TRT barrel	554<R<1082 (mm)
TRT end-cap	617<R<1106 (mm)

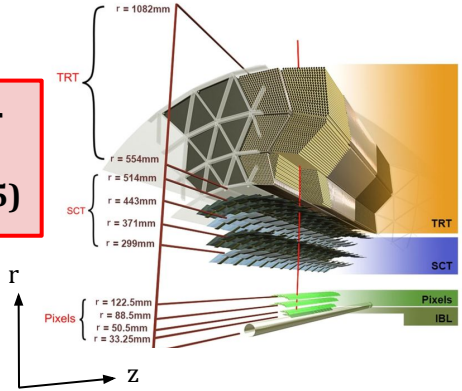
ATLAS - ID and tracking procedure

Eur. Phys. J. C (2017) 77:673
ATL-PHYS-PUB-2021-012

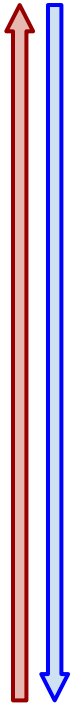
mfaggin@cern.ch

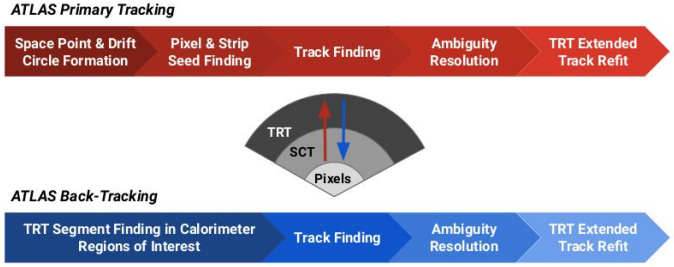
B = 2T (axial)

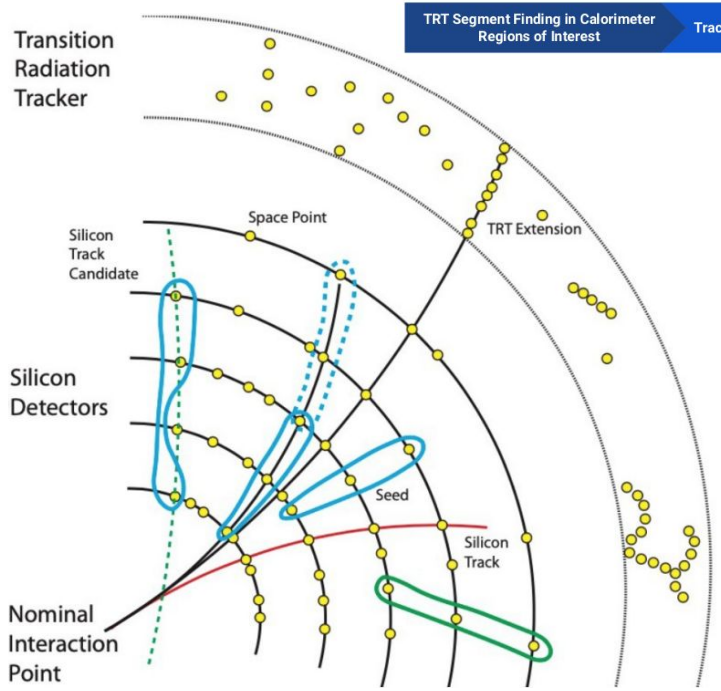
Inner Detector (ID) tracking system ($|\eta| < 2.5$)



1. **Pixel**
   - [barrel] 3 layers + insertable B-layer (IBL)
   - [endcap] 3 disks on each side
2. **Silicon SemiConductor Tracker (SCT)**
   - Strip detector
   - [barrel] 4 double-strip layers
   - [endcap] 9 disks on each side
3. **Transition Radiation Tracker (TRT)**
   - Straw-tube tracker → tubes 4 mm wide
   - [barrel] $0.5 \lesssim r \lesssim 1$
   - [endcap] straw tubes $\perp$ beam line within 0.8 m < $|z|$ < 2.7 m







After the seed has been determined, the iterative primary vertex finding procedure begins. The vertex position is determined using an adaptive vertex fitting algorithm with an annealing procedure [25]. Using the seed position as the starting point and parameters of reconstructed tracks as input measurements, the algorithm performs an iterative χ^2 minimisation, finding the optimal vertex position. Each input track is assigned a weight, reflecting its compatibility with the vertex estimate. The vertex position is recalculated using the weighted tracks, and then the procedure is repeated, recalculating track weights with respect to the new vertex position. The individual track weights are calculated according to the following equation:

$$\omega(\hat{\chi}^2) = \frac{1}{1 + \exp\left(\frac{\hat{\chi}^2 - \chi^2_{cutoff}}{2T}\right)}. \quad (4)$$

Here $\hat{\chi}^2$ is the χ^2 value calculated in three dimensions between the last estimated vertex position and the respective point of the closest approach of the track. Tracks with lower weights are less compatible with the vertex and will have less influence on the position calculation. The constant χ^2_{cutoff} defines the threshold where the weight of an individual track becomes equal to 0.5. Tracks with low weights are not removed, but will have less impact on the calculated vertex position. The value of χ^2_{cutoff} is set to nine, which corresponds to about three standard deviations. The temperature T controls the smoothness of the weighting procedure. For low values of T, $\omega(\hat{\chi}^2)$ approaches a step function, and for large values of T the function flattens, progressively losing its χ^2 dependence. To avoid convergence in local minima, the weighting procedure is applied progressively by decreasing the temperature T during the fit iterations. The temperature is lowered from some high starting value in a pre-defined sequence of steps that converges at $T = 1$. A typical distribution of track weights is shown in Fig. 3. It widens as T decreases, reaching an optimal separation of track outliers for $T = 1$.

- *Seed-finding* Tracks not yet assigned to any vertex candidate are analysed to find the most likely position of a primary vertex using the GS (Section 5).
- *Track to seed assignment* After a seed is found, the set of nearby tracks to fit is chosen. One essential difference from the iterative single-vertex strategy is that *all* tracks passing the quality selection (not only unassigned tracks, but also tracks already assigned to one or more previously accepted vertex candidates) are eligible for assignment to new vertex candidates. Thus, unlike the IVF, each track may, and generally will, be assigned to multiple vertices. The track assignment criteria and their tuning are detailed in Section 6.2.
- *Fitting* The linearized helical parameters of assigned tracks are used to fit the position of the vertex candidate with a weighted adaptive Kalman filter, subject to transverse and longitudinal constraints provided by the beam spot and seed positions, respectively. Another important difference from the iterative strategy is that each time a new vertex candidate is fit, all other candidates linked to it (through a chain of tracks and vertices of any length) are also simultaneously refit.[2] Track weights ω with each vertex i are annealed in six steps based on compatibility (χ^2). After each decrease of the temperature T, weights are recalculated and all connected vertices are refit. The total weight for a track across all assigned vertices is normalized to one, but for purposes of weight normalization (only) tracks are also given a notional three standard deviation (corresponding to $\chi^2_0 = 9$) compatibility with "unassigned":

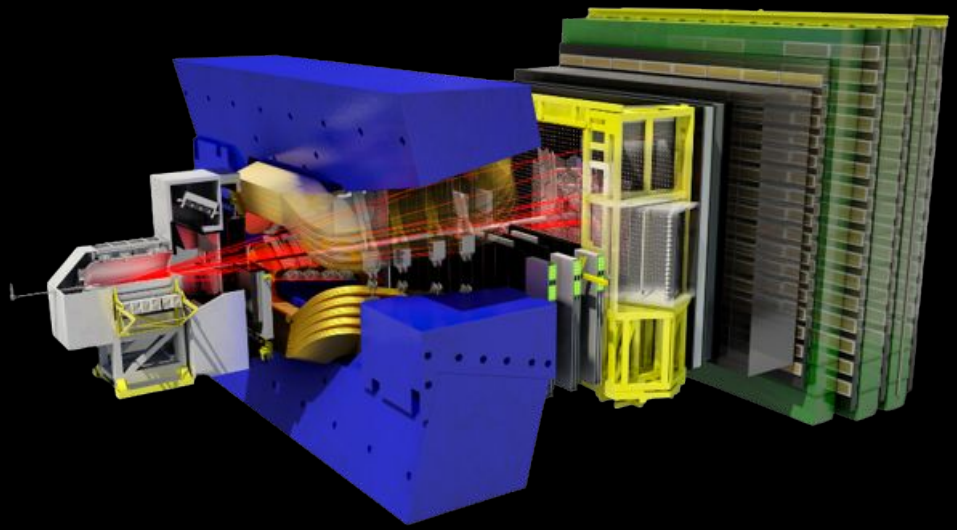
$$\omega_i(\chi^2_i, T) = \frac{e^{-\frac{1}{2}\chi^2_i/T}}{\sum_j e^{-\frac{1}{2}\chi^2_j/T} + e^{-\frac{1}{2}\chi^2_0/T}}$$

- *Acceptance/rejection* The new vertex candidate is accepted if it satisfies three criteria. First, it must have at least two compatible tracks from the seeding track pool; a new vertex cannot be formed from *only* tracks that are compatible with some previously-found vertex. Track compatibility is defined as a χ^2_2 probability greater than 10^{-4}. Second, the new vertex's fit position may not be within 3σ of any other, based on the calculated fit errors of both vertices. Third, the weighted average of track weights in the fit ($\sum \omega^2 / \sum \omega$) must be greater than $\frac{2}{3}$. If the vertex candidate is accepted, all compatible tracks are removed from the seeding pool. If the vertex candidate is not accepted, only the *most* compatible track is removed from the seeding pool. Thus, regardless of outcome, the pool of remaining tracks for seeding is always reduced by each iteration, ensuring eventual termination of vertex-finding.

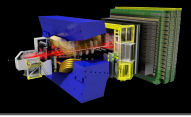
Adaptive multi-vertex finding is complete when

- less than two tracks remain in the pool for seed-finding, or
- the seed finder is unable to return a seed, or
- the maximum allowed number of iterations is exceeded.

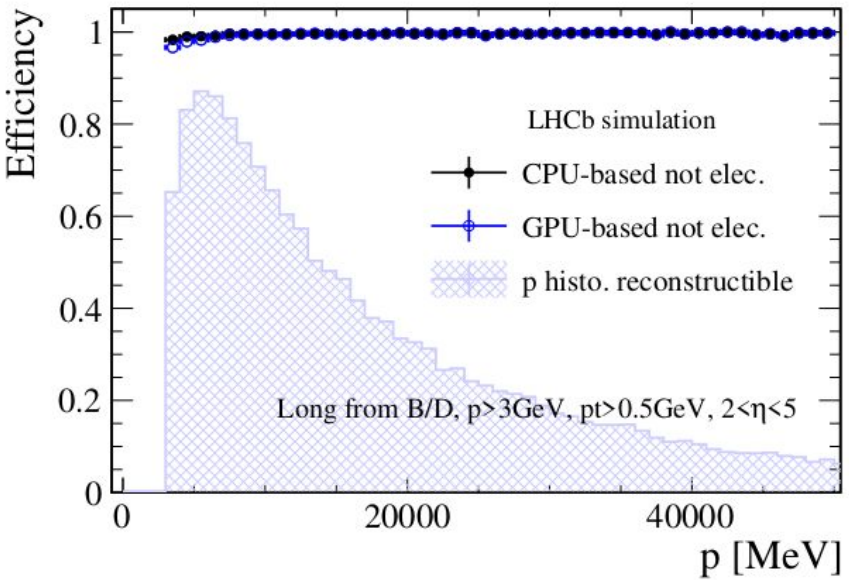
LHCb

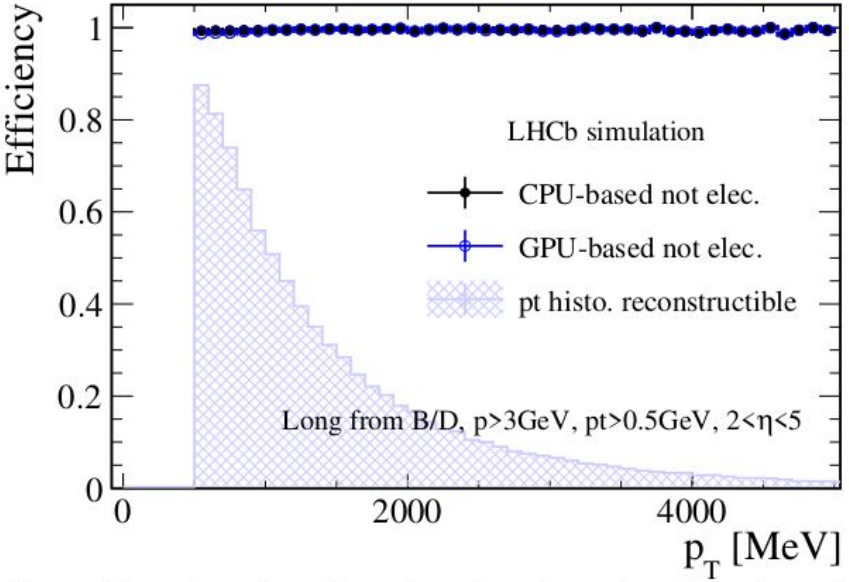


LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)

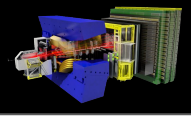


mfaggin@cern.ch

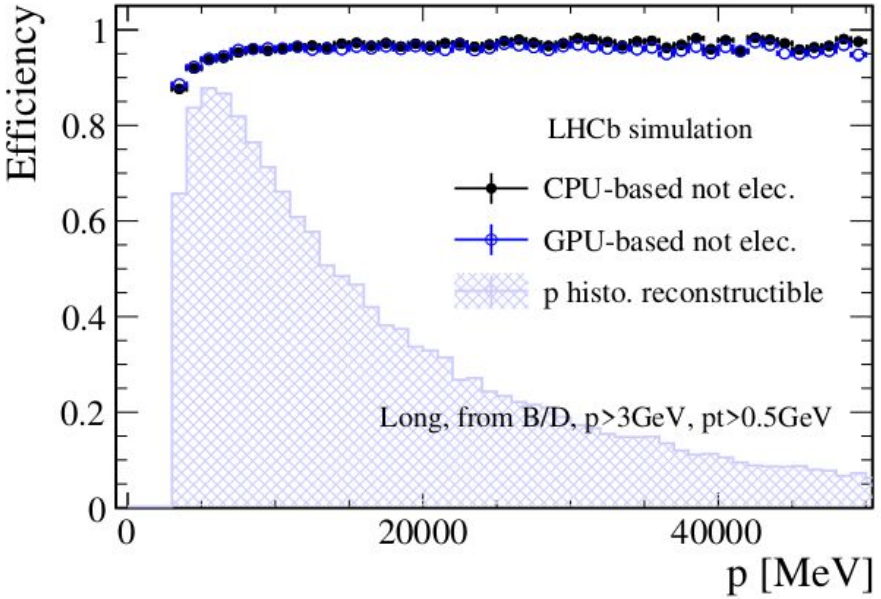


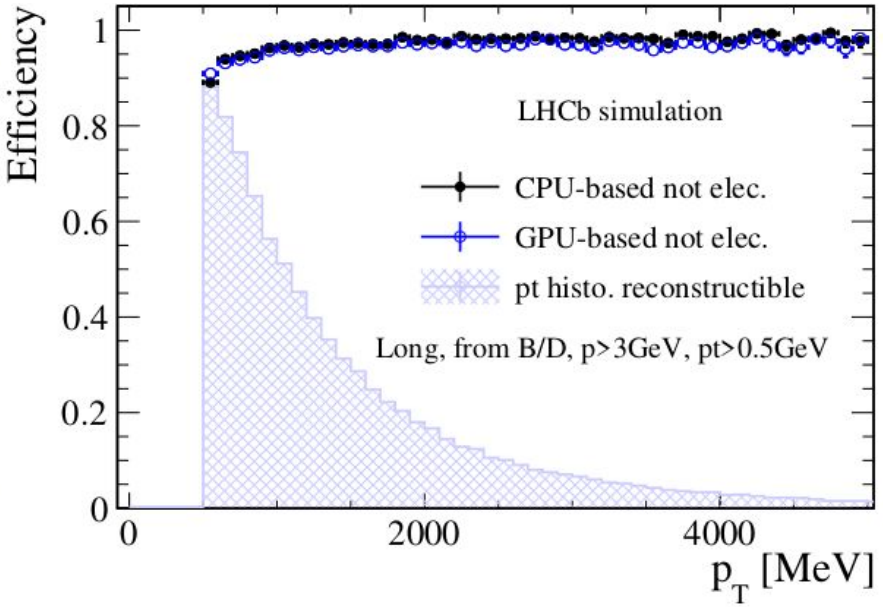


LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)

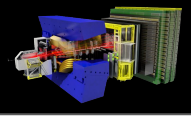


mfaggin@cern.ch

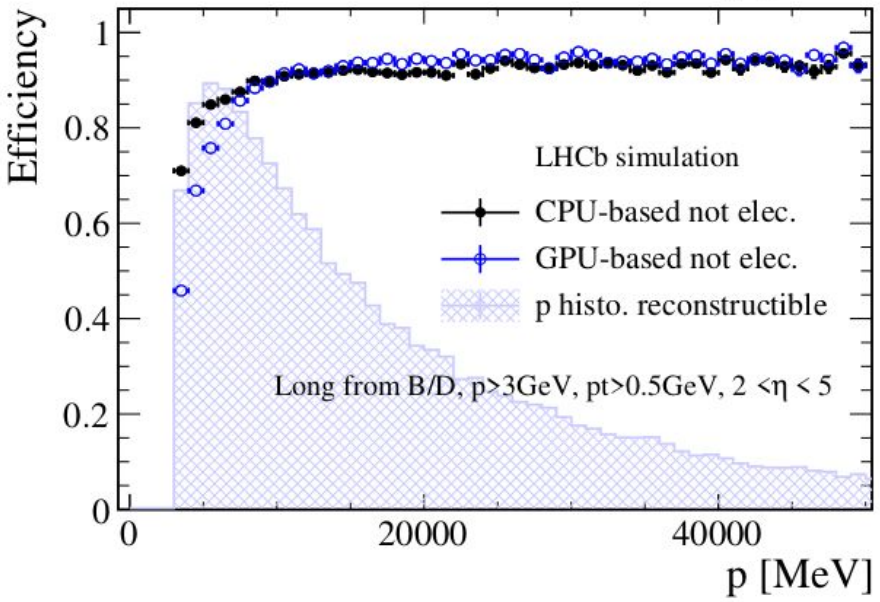


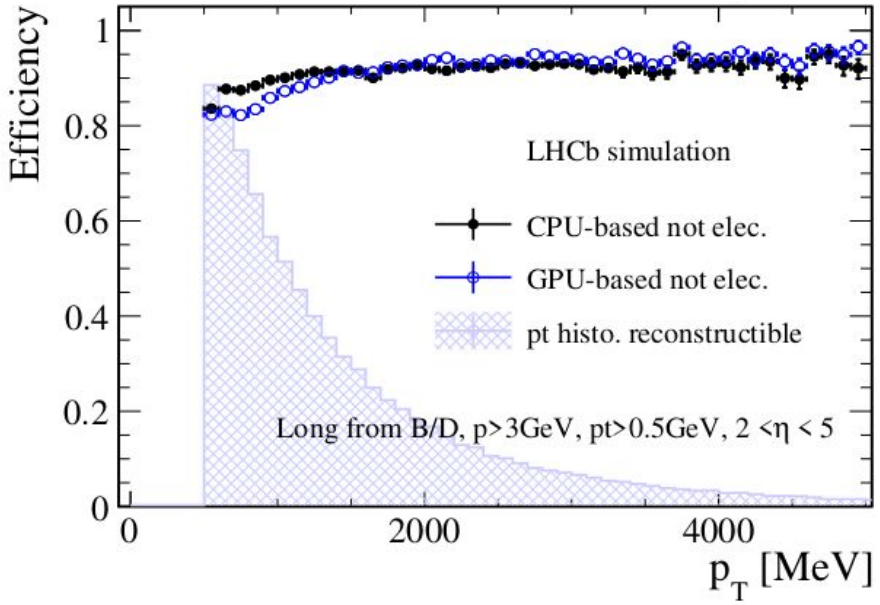


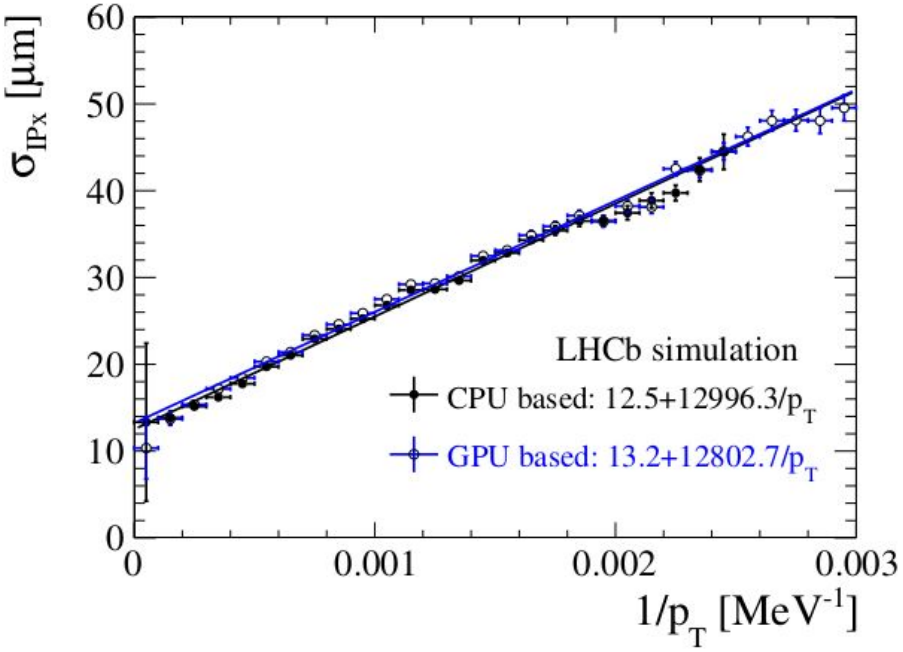
LHCb-DP-2021-003
Comput Softw Big Sci 4, 7 (2020)

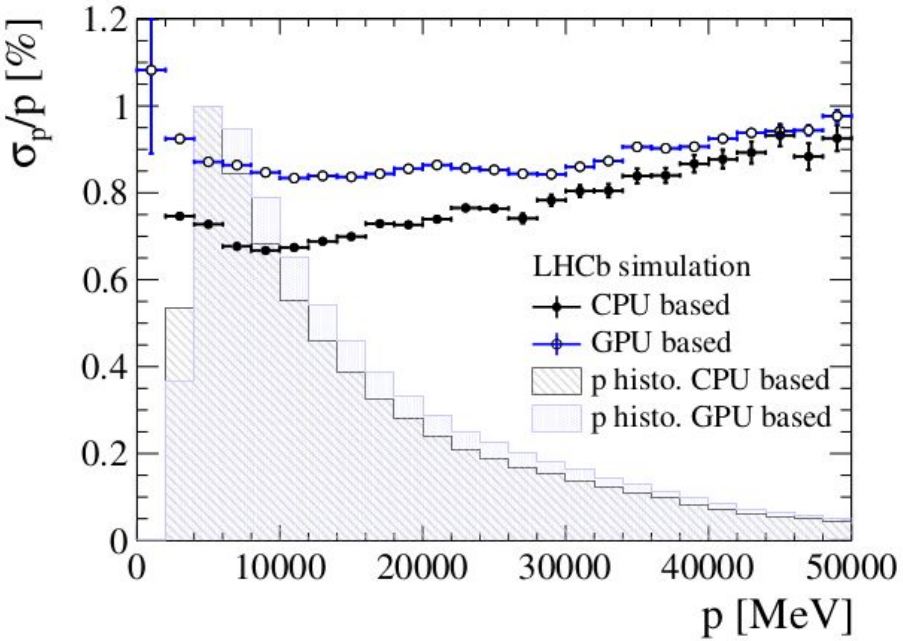


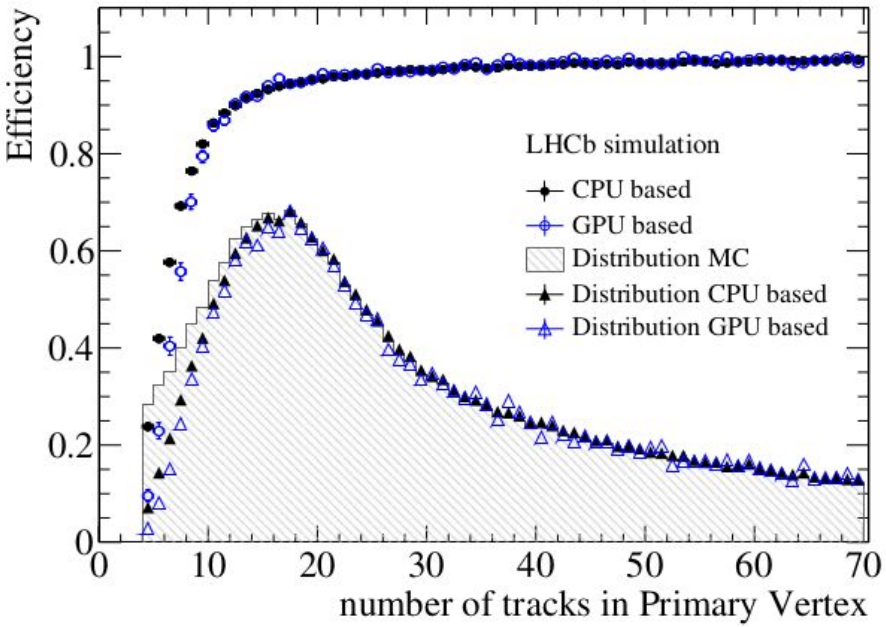
mfaggin@cern.ch

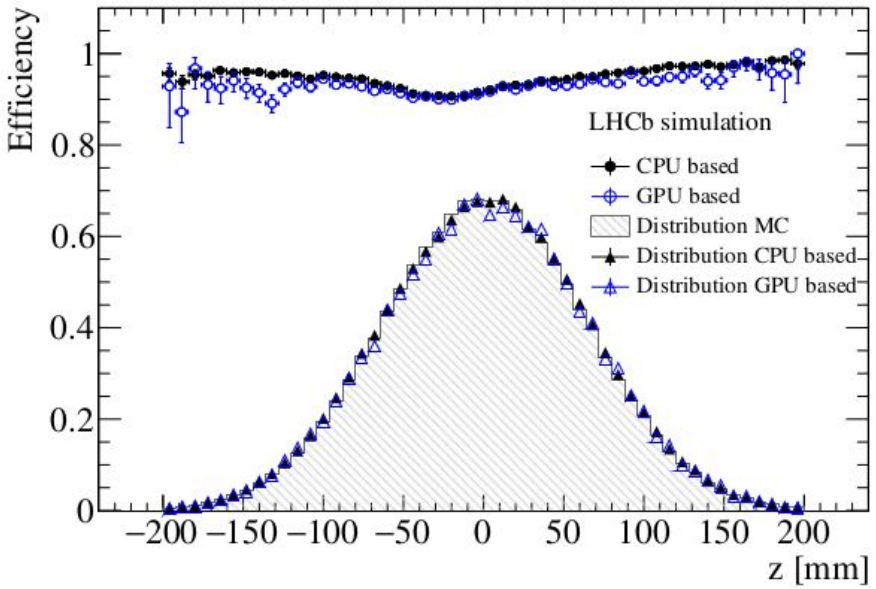





σ_{IPx} [μm]
60
50
40
30
20
10
0
0
0.001
0.002
0.003
$1/p_T$ [MeV^{-1}]
LHCb simulation
CPU based: 12.5+12996.3/p_T
GPU based: 13.2+12802.7/p_T


σ_p/p [%]
1.2
1
0.8
0.6
0.4
0.2
0
0
10000
20000
30000
40000
50000
p [MeV]
LHCb simulation
CPU based
GPU based
p histo. CPU based
p histo. GPU based


Efficiency
LHCb simulation
CPU based
GPU based
Distribution MC
Distribution CPU based
Distribution GPU based
number of tracks in Primary Vertex


Efficiency
LHCb simulation
CPU based
GPU based
Distribution MC
Distribution CPU based
Distribution GPU based
z [mm]

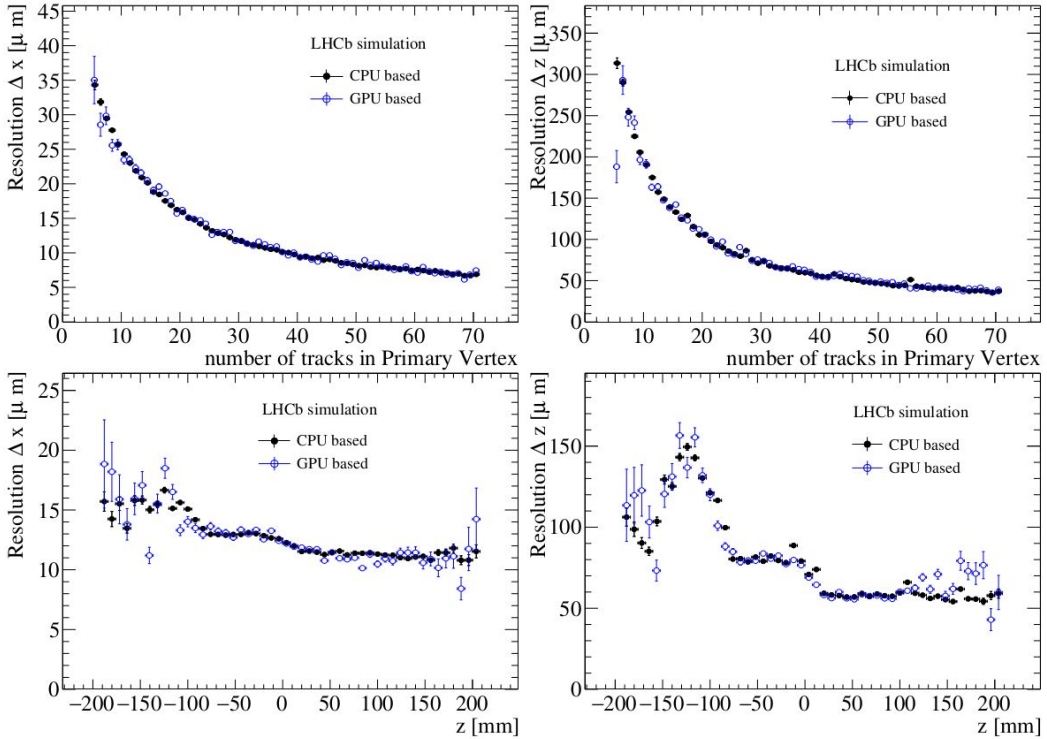
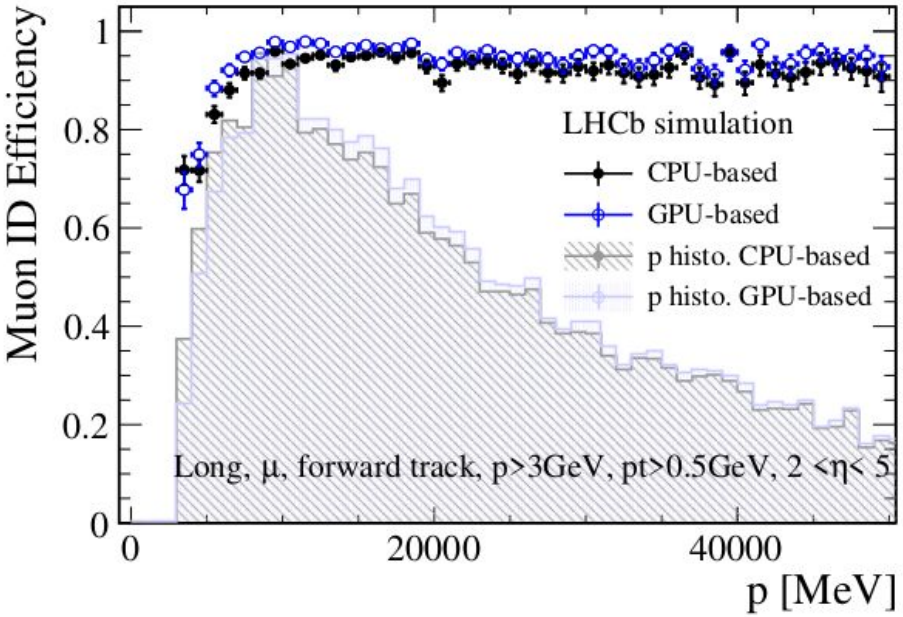
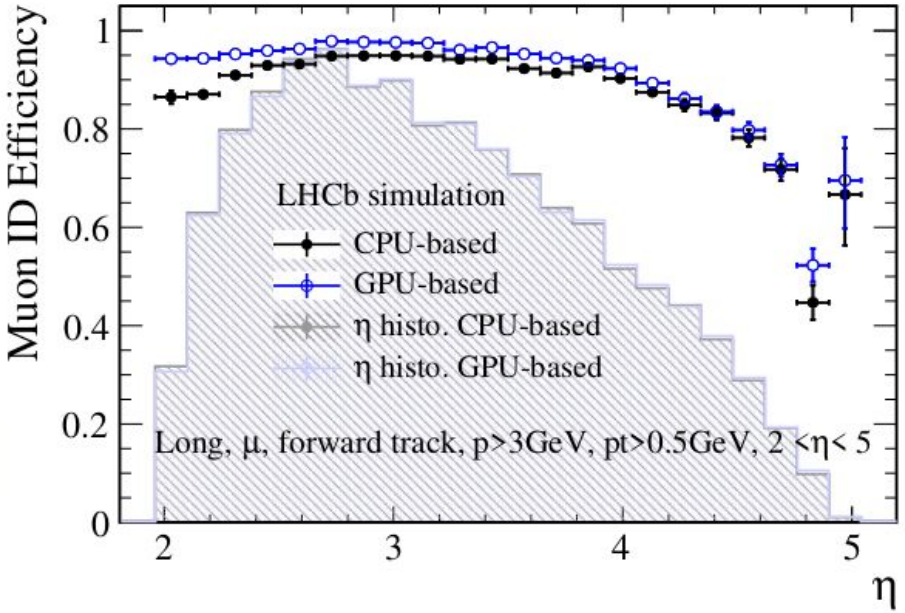


Figure 8: Resolution in x and z of all reconstructed primary vertices as function of the number of reconstructed Velo tracks associated to the simulated primary vertex (top row) and as function of the true vertex z position (bottom row).


Muon ID Efficiency
LHCb simulation
CPU-based
GPU-based
p histo. CPU-based
p histo. GPU-based
Long, μ, forward track, p>3GeV, pt>0.5GeV, 2 <η< 5
p [MeV]


Muon ID Efficiency
LHCb simulation
CPU-based
GPU-based
η histo. CPU-based
η histo. GPU-based
Long, μ, forward track, p>3GeV, pt>0.5GeV, 2 <η< 5
η

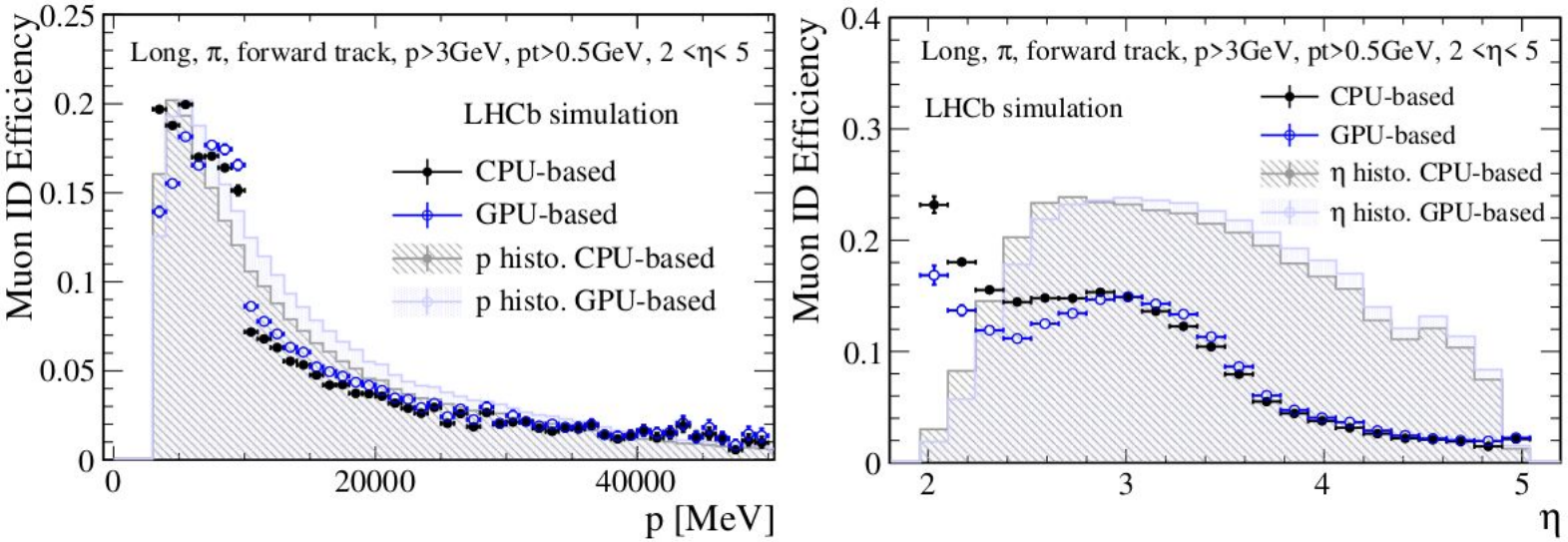


Figure 10: Efficiency to falsely identify true pions reconstructed as long muon tracks as muons as function of the pion momentum and pseudorapidity.

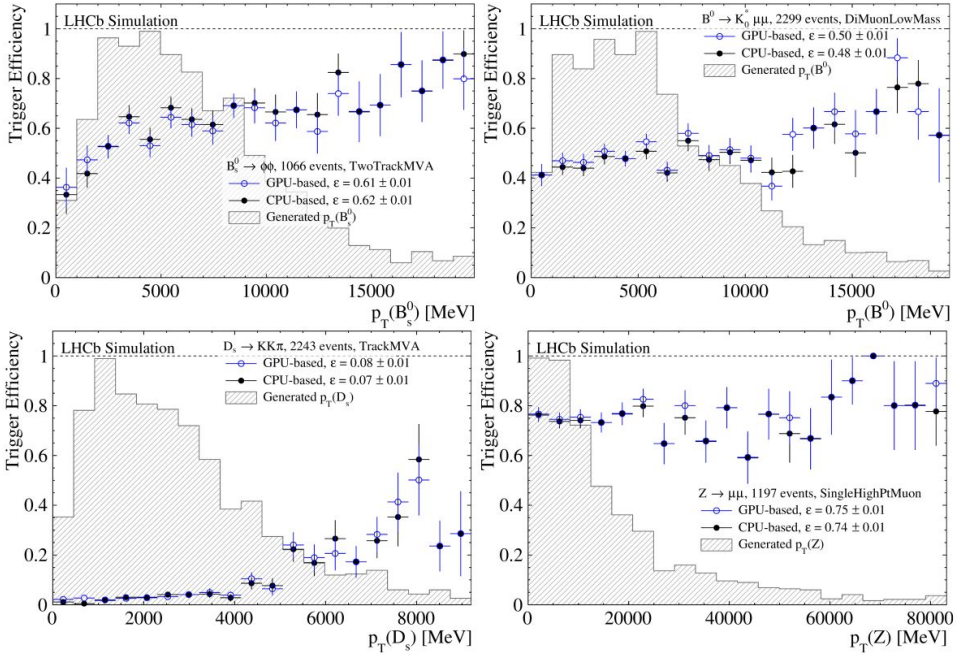


Figure 11: Trigger efficiencies for CPU-based and GPU-based HLT1 as a function of parent transverse momentum. Results are shown for the `TwoTrackMVA` (top left), `DiMuonLowMass` (top right), `TrackMVA` (bottom left) and `SingleHighPtMuon` (bottom right) selections firing on the $B_s^0 \to \phi\phi$, $B^0 \to K^{*0}\mu^+\mu^-$, $D_s \to KK\pi$ and $Z \to \mu^+\mu^-$ signal samples, respectively. The generated parent transverse momentum distribution is also shown for all events passing the denominator requirement.

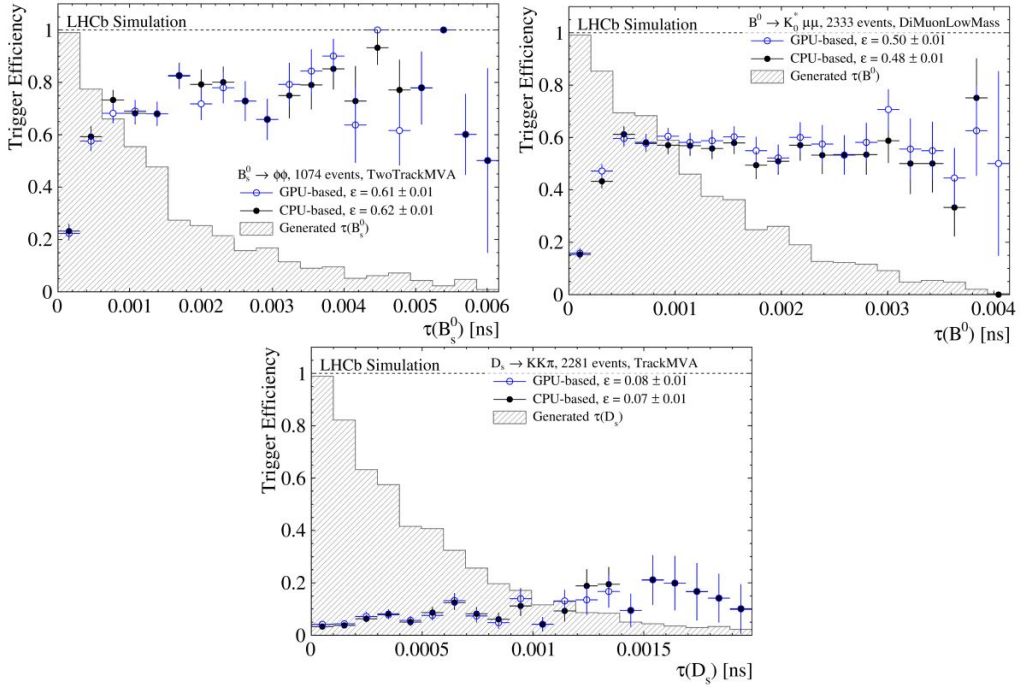
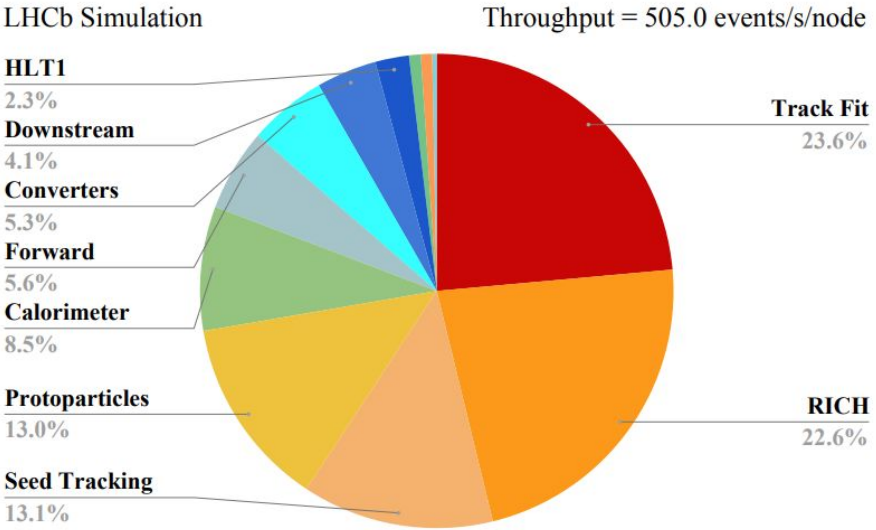
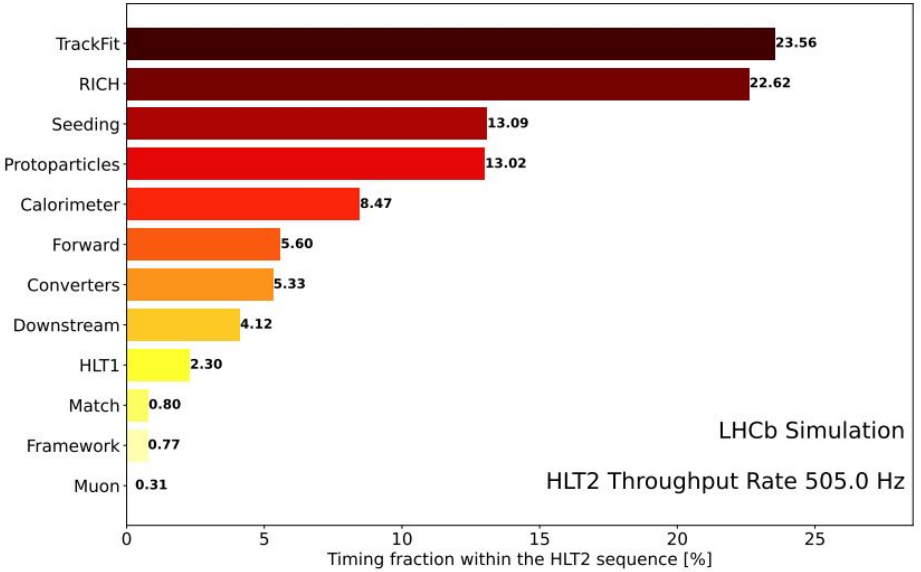
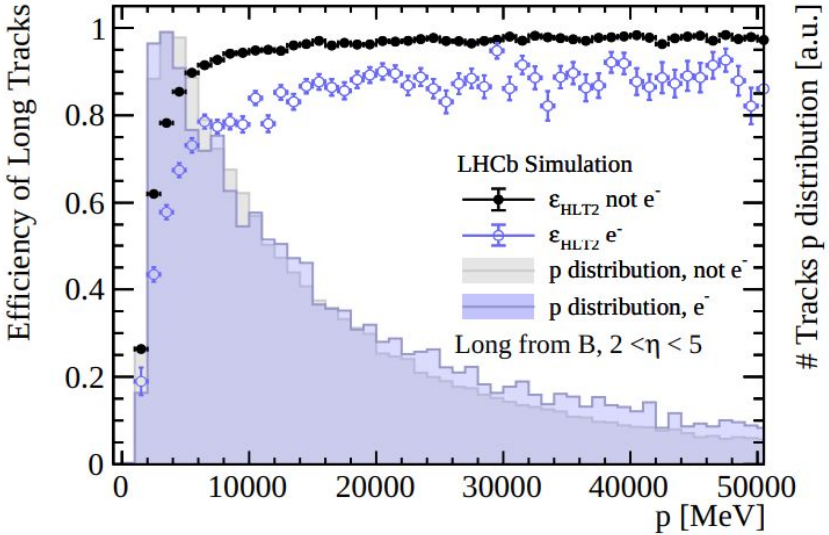
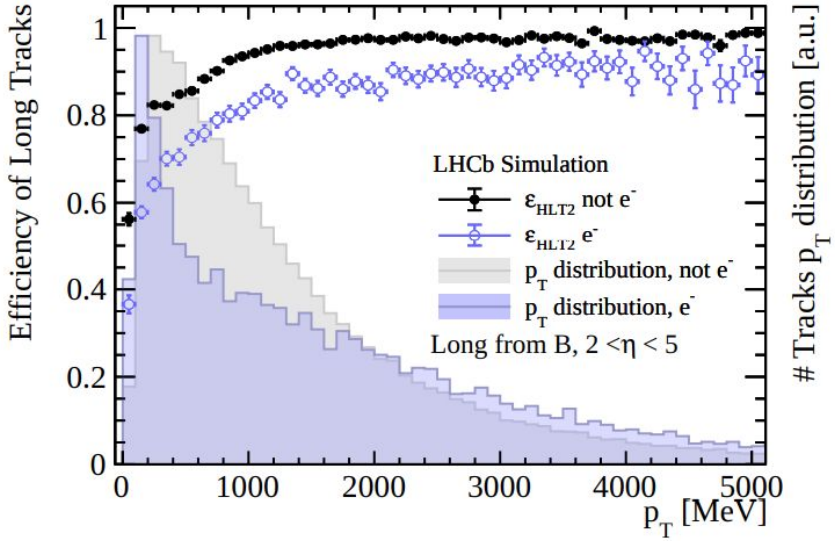


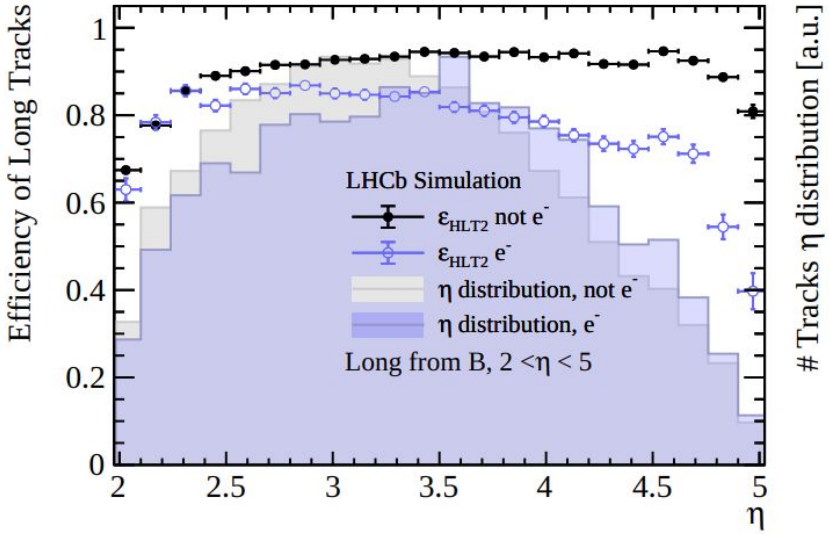
Figure 12: Trigger efficiencies for CPU-based and GPU-based HLT1 as a function of parent decay time. Results are shown for the `TwoTrackMVA` (top left), `DiMuonLowMass` (top right) and `TrackMVA` (bottom) selections firing on the $B_s^0 \to \phi\phi$, $B^0 \to K^{*0}\mu^+\mu^-$ and $D_s \to KK\pi$ signal samples, respectively. The generated parent decay time distribution is also shown for all events passing the denominator requirement.

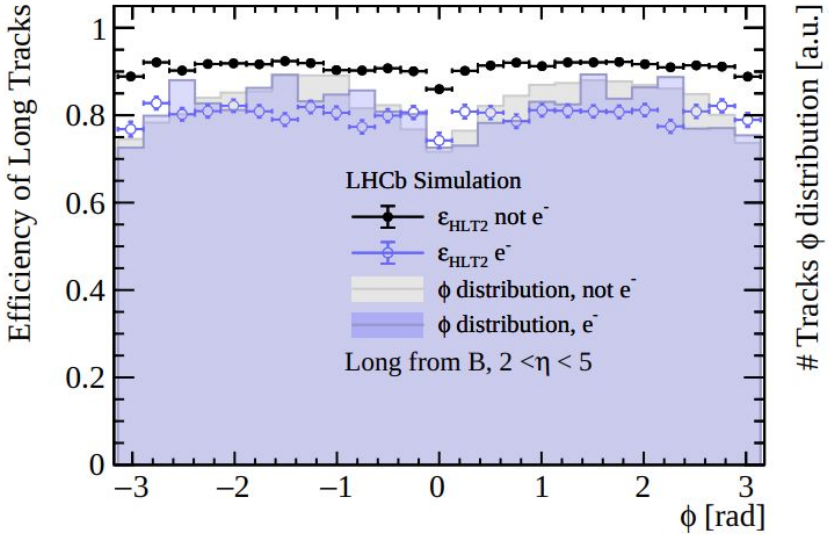

LHCb Simulation
Throughput = 505.0 events/s/node
HLT1
2.3%
Downstream
4.1%
Converters
5.3%
Forward
5.6%
Calorimeter
8.5%
Protoparticles
13.0%
Seed Tracking
13.1%
Track Fit
23.6%
RICH
22.6%

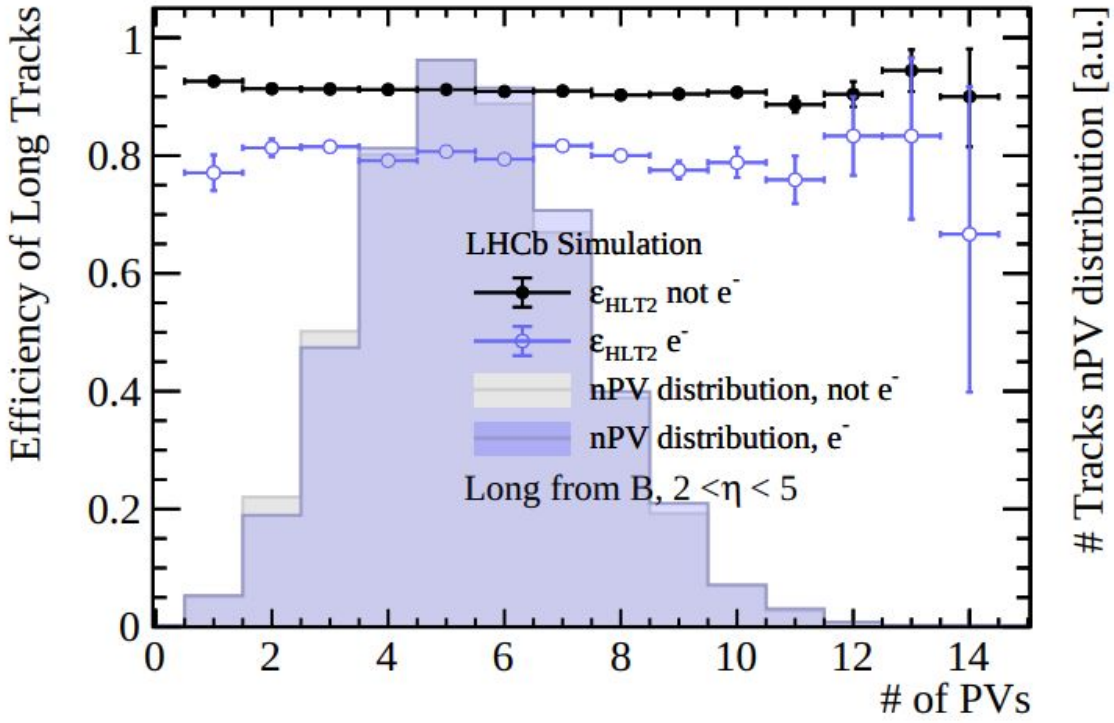

TrackFit 23.56
RICH 22.62
Seeding 13.09
Protoparticles 13.02
Calorimeter 8.47
Forward 5.60
Converters 5.33
Downstream 4.12
HLT1 2.30
Match 0.80
Framework 0.77
Muon 0.31
LHCb Simulation
HLT2 Throughput Rate 505.0 Hz
0
5
10
15
20
25
Timing fraction within the HLT2 sequence [%]


Efficiency of Long Tracks
Tracks p distribution [a.u.]
p [MeV]
LHCb Simulation
ε_{HLT2} not e$^-$
ε_{HLT2} e$^-$
p distribution, not e$^-$
p distribution, e$^-$
Long from B, 2 <η < 5


Efficiency of Long Tracks
Tracks p_T distribution [a.u.]
p_T [MeV]
LHCb Simulation
ε_{HLT2} not e$^-$
ε_{HLT2} e$^-$
p_T distribution, not e$^-$
p_T distribution, e$^-$
Long from B, 2 <η < 5


Efficiency of Long Tracks
Tracks η distribution [a.u.]
η
LHCb Simulation
ε_HLT2 not e⁻
ε_HLT2 e⁻
η distribution, not e⁻
η distribution, e⁻
Long from B, 2 <η < 5


Efficiency of Long Tracks
Tracks ϕ distribution [a.u.]
ϕ [rad]
LHCb Simulation
ε_HLT2 not e⁻
ε_HLT2 e⁻
ϕ distribution, not e⁻
ϕ distribution, e⁻
Long from B, 2 <η < 5


Efficiency of Long Tracks
Tracks nPV distribution [a.u.]
of PVs
LHCb Simulation
ε_{HLT2} not e$^-$
ε_{HLT2} e$^-$
nPV distribution, not e$^-$
nPV distribution, e$^-$
Long from B, 2 <η < 5
0
0.2
0.4
0.6
0.8
1
0
2
4
6
8
10
12
14

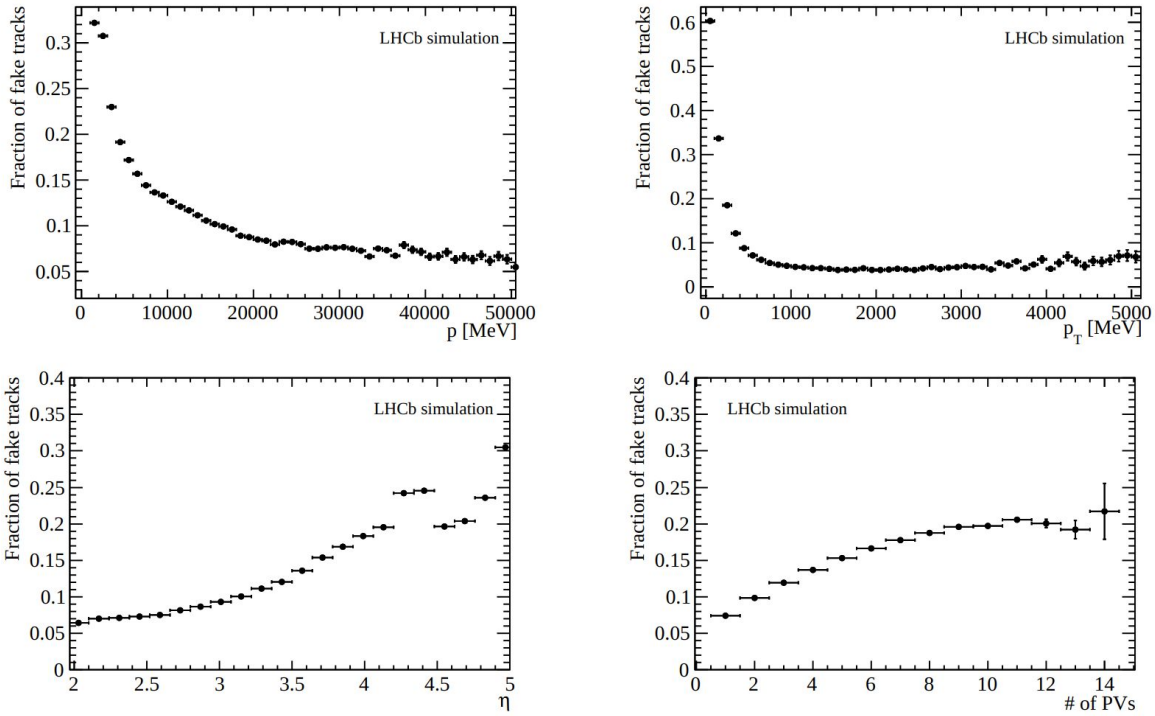
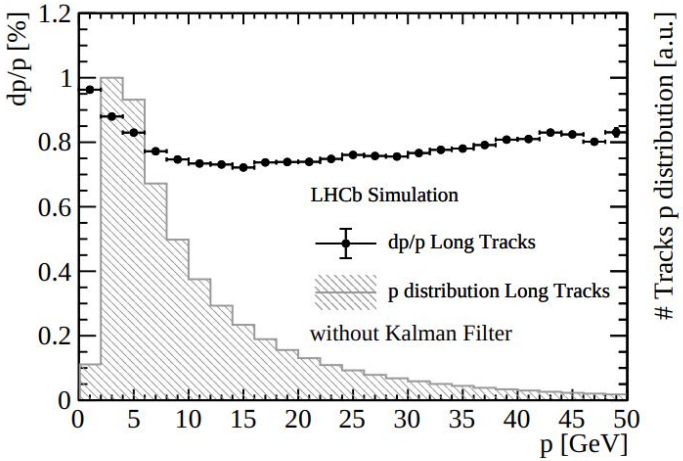


Figure 3: Fake track fraction of long tracks reconstructed by the Forward Tracking algorithm as a function of momentum p, transverse momentum p_T, pseudo-rapidity η and number of primary vertices.

FAKE TRACKS: the high pseudorapidity region exhibits the highest fake track fraction, partly because of increased multiple scattering and hadronic interactions due to the material, but even more because of the higher SciFi hit density in the very forward direction leading to more possible random hit combinations. Fig. 4a shows the fake track fraction in dependence of the fake track momentum. The integrated fake track fraction coming from the Forward Tracking amounts to 15%. It is afterwards reduced by applying a Kalman Filter and evaluation of a fake track classifier.



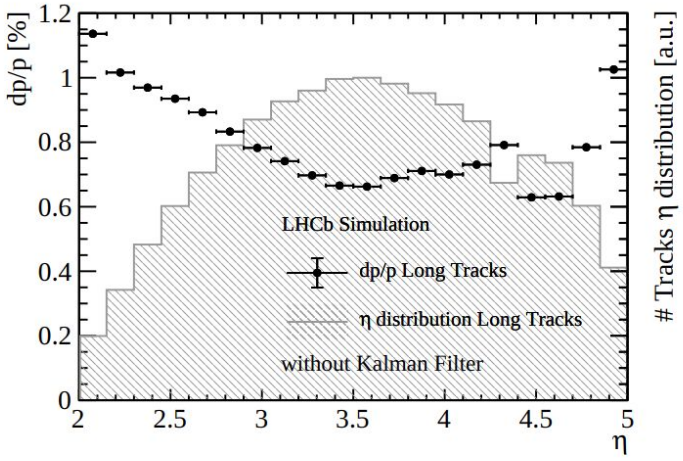


Figure 4: Relative resolution of the momentum of tracks reconstructed by the Forward Tracking algorithm as a function of momentum p and pseudorapidity η.

3.1 SciFi hit selection

Starting from a state vector $(x, y, \frac{\partial x}{\partial z}, \frac{\partial y}{\partial z}, \frac{q}{p})$ containing the position, slopes and so far unknown charge q and momentum p at the end of the VELO, *i.e.* a VELO track, the Forward Tracking defines a polynomial $P(\frac{\partial x}{\partial z}, \frac{\partial y}{\partial z}, p)$ parameterising the propagation of the track in the xz-plane through the magnetic field down to the SciFi layers. As the momentum and charge are not known yet, this parameterisation is used to calculate x_{min} and x_{max} positions for each layer assuming a minimum reconstructible track momentum p_{min}, *e.g.* $p_{\text{min}} = 1.5\,\text{GeV}$, and both possible charges. Only SciFi hits with $x \in [x_{min}, x_{max}]$ are selected for further processing, reducing the complexity of the problem. The determination of this hit search window by a polynomial is computationally cheap and therefore preferred over numerically solving the equations of motion

3.2 Simplified particle trajectory

Similarly to the parameterisation used to define the hit search window (Sec. 3.1), the Forward Tracking defines a simplified particle trajectory by treating the magnet as an optical lens as described in Ref. [18]. Just like the model of light rays refracted by a thin lens, the particle's movement in the xz-plane through the magnetic field is modelled by a straight line that gets a kick at the centre of the magnetic field and propagates further as a straight line with a different slope. Once a single hit (x, z) downstream of the magnet is taken into account, predicting the x coordinate at a given z position is a simple linear extrapolation within the model. Deviations from this model occur because of fringe magnetic fields that reach into the SciFi detector and are corrected for by parameterising the effect using polynomials as described in Ref. [19].

3.3 Hough-like transform

The main component of the Forward Tracking applies a map-reduce pattern inspired by the Hough transform to sieve out sets of SciFi hits that do not form a matching extension to the VELO track.

The x positions of SciFi hits selected by the search window described in Sec. 3.1 are mapped to a reference plane at a fixed z position. All hits' x positions at the reference plane are calculated using the simplified trajectory introduced in Sec. 3.2 and filled into a histogram. The histogram counts the number of unique SciFi detector layers that are present among the hits in one bin. This way, hits that do not qualify as an extension to the VELO track form a flat distribution, while hits that match the VELO track accumulate in a few bins as depicted in Fig. 2. Subsequently, the histogram is scanned for small groups of neighbouring bins exceeding a layer-count threshold, thus reducing the large set of hits from within the search window to none, one or several small sets of hits, which become candidates for the VELO track extension. The found hit sets are then cleaned from outliers, fitted using a third-order polynomial function and further selected according to the fit result. The remaining candidates are promoted to Long tracks and their charge and momentum are estimated.

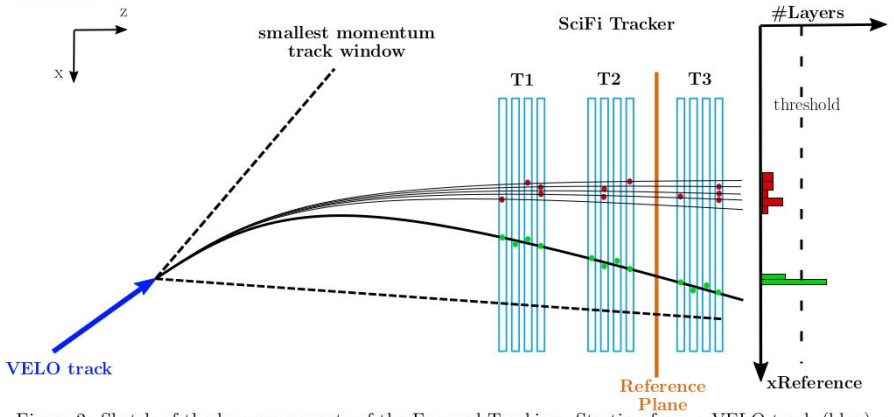
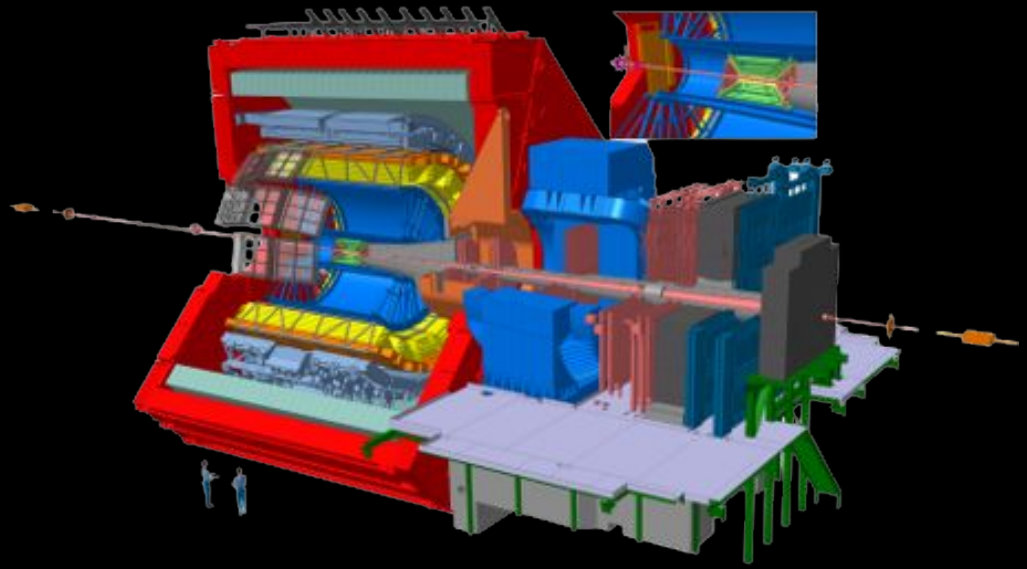
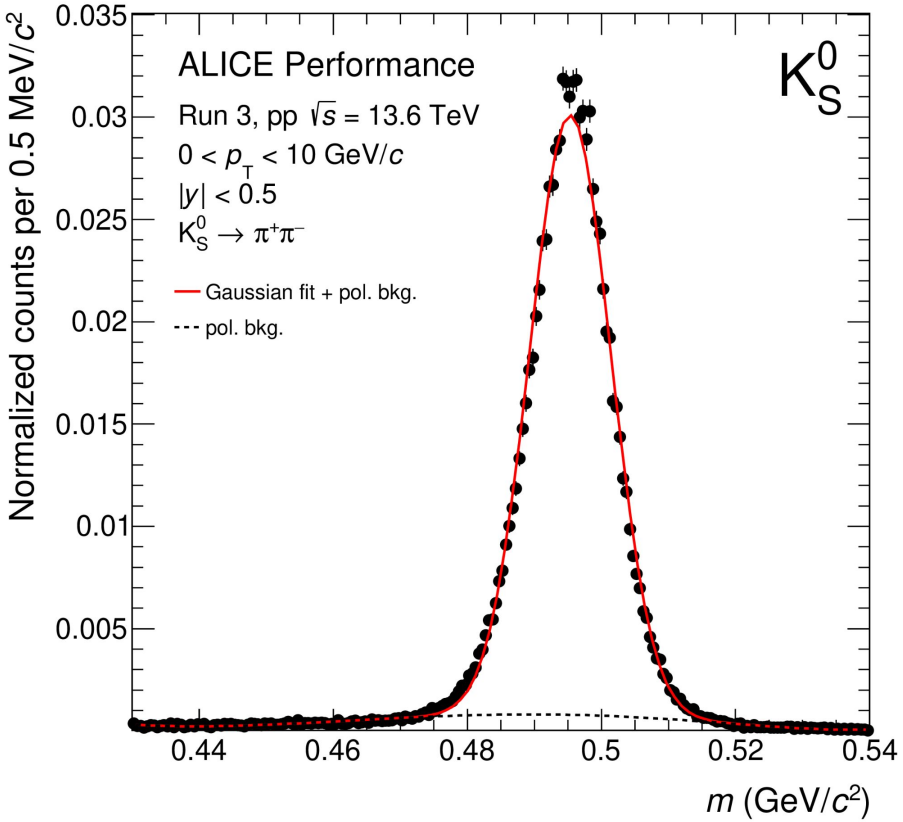


Figure 2: Sketch of the key components of the Forward Tracking. Starting from a VELO track (blue), a smallest-momentum hit search window is calculated (black dashed line). The x positions of hits in the twelve SciFi detector layers (three stations T1, T2 and T3 with four layers each in light blue) are projected to the reference plane (orange) using a simplified track model (not shown here). The rightmost part of the figure shows the histogram counting the number of unique SciFi detector layers depending on the projected x positions. Hits belonging to the VELO track are shown in green, and other hits are in red.

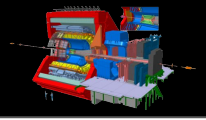
ALICE



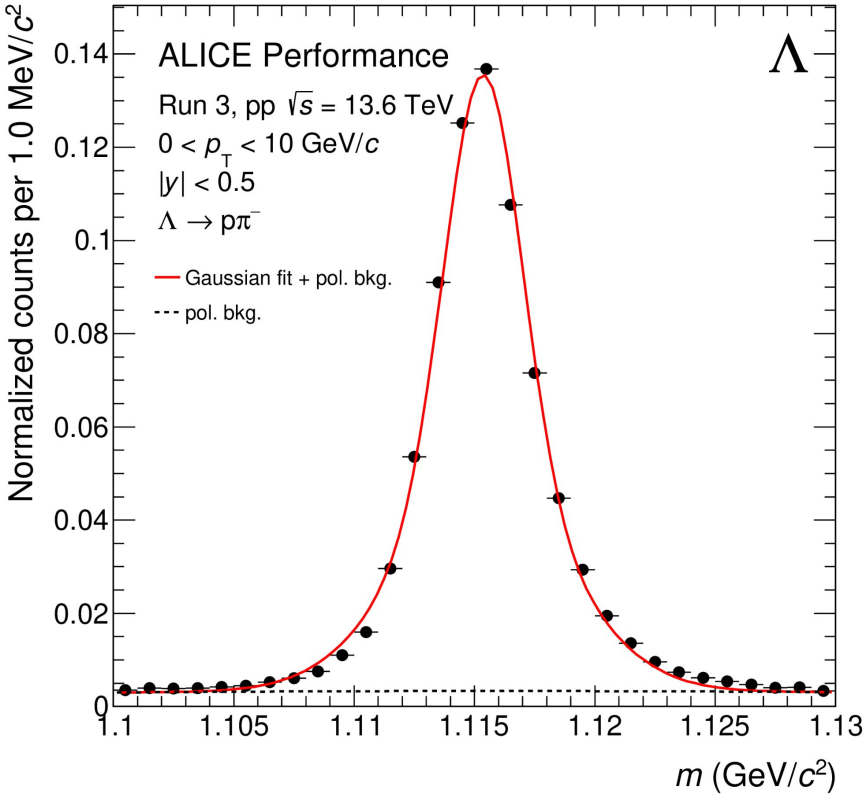


ALI-PERF-528877

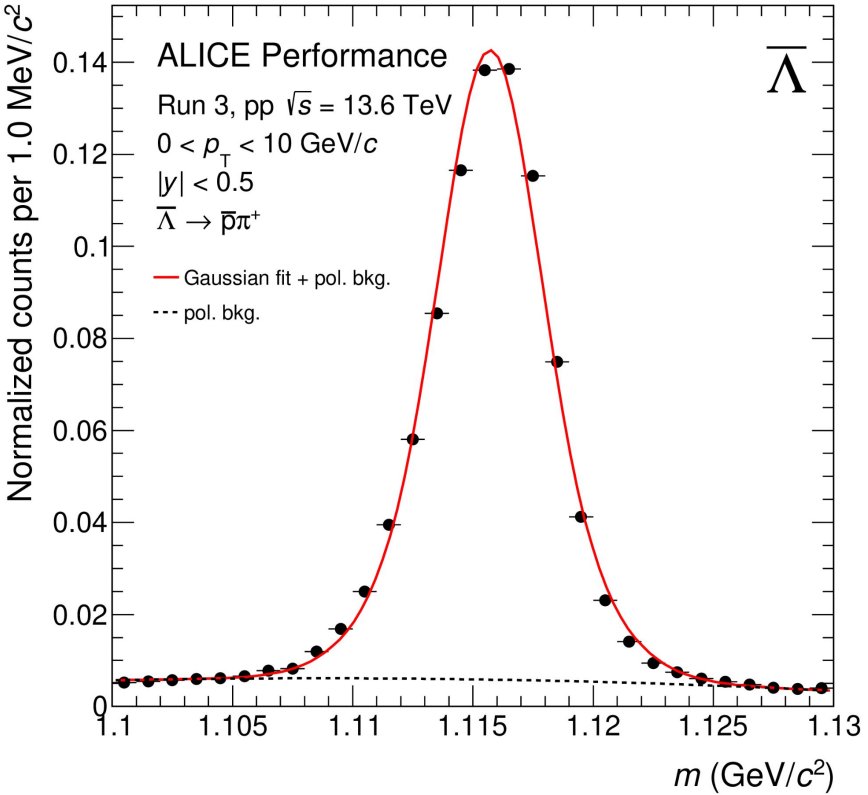
ALICE - tracking performance in Run 3, 4



mfaggin@cern.ch

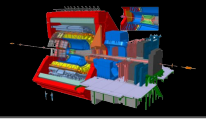


ALI-PERF-529191

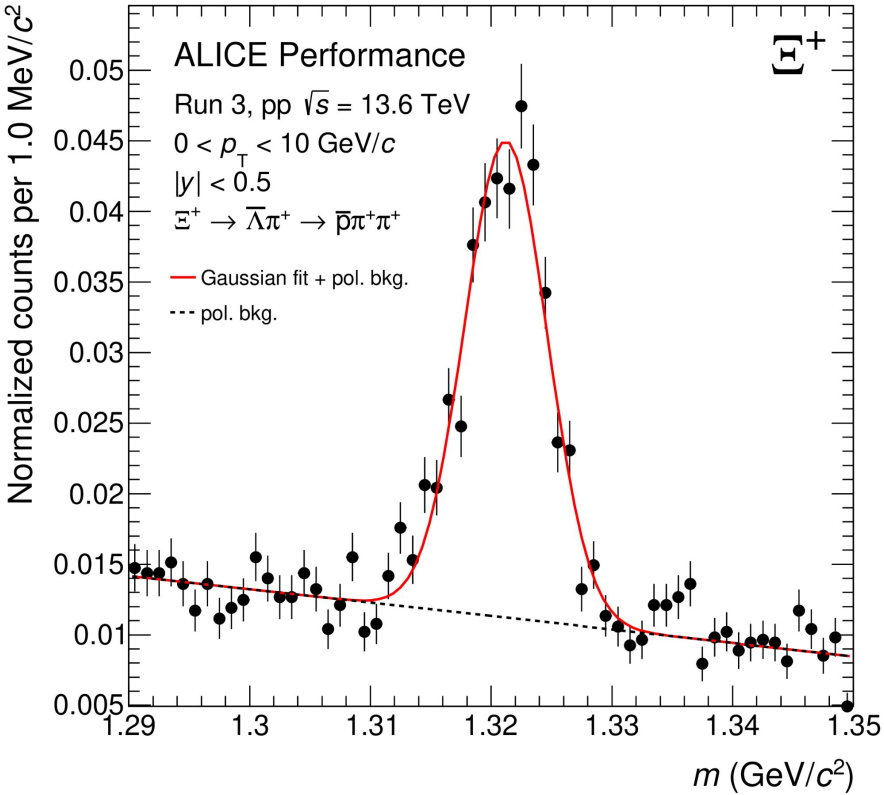


ALI-PERF-529185

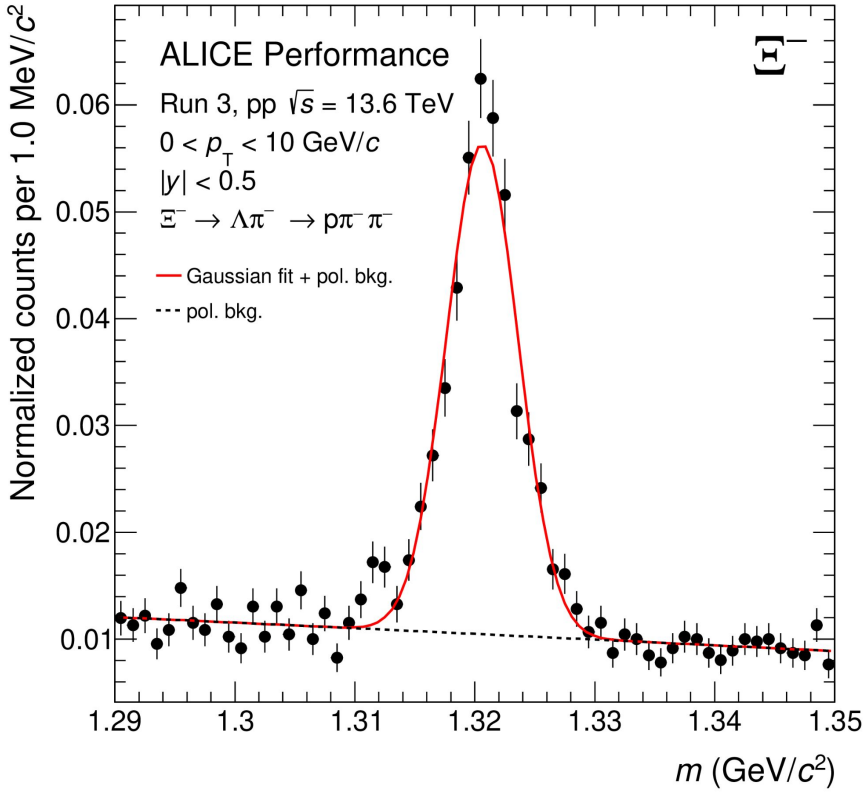
ALICE - tracking performance in Run 3, 4



mfaggin@cern.ch

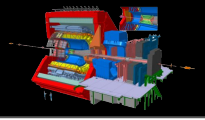


ALI-PERF-528851

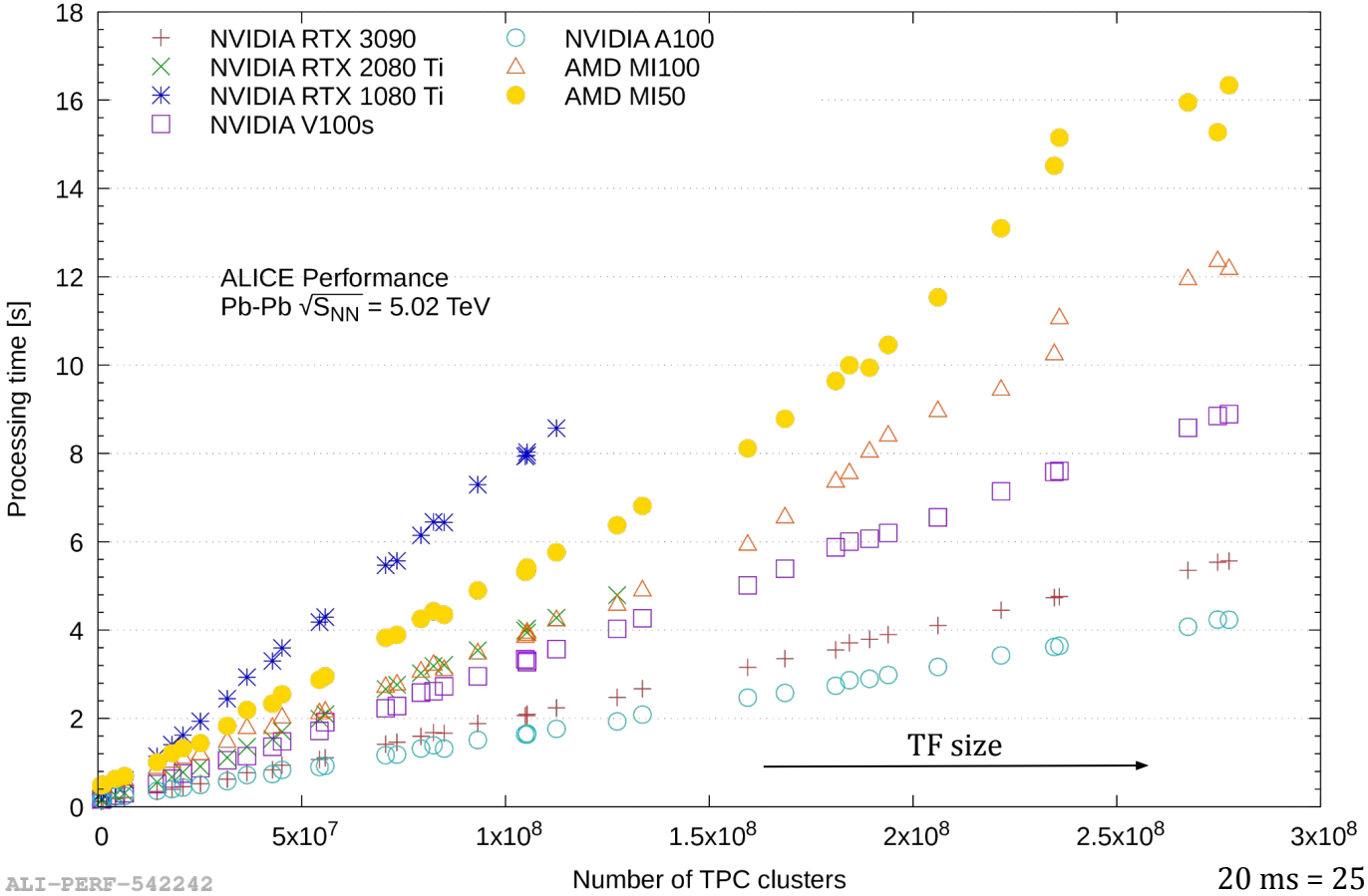


ALI-PERF-528848

ALICE - sync. TPC tracking on GPUs



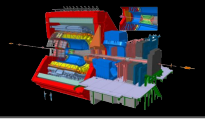
mfaggin@cern.ch



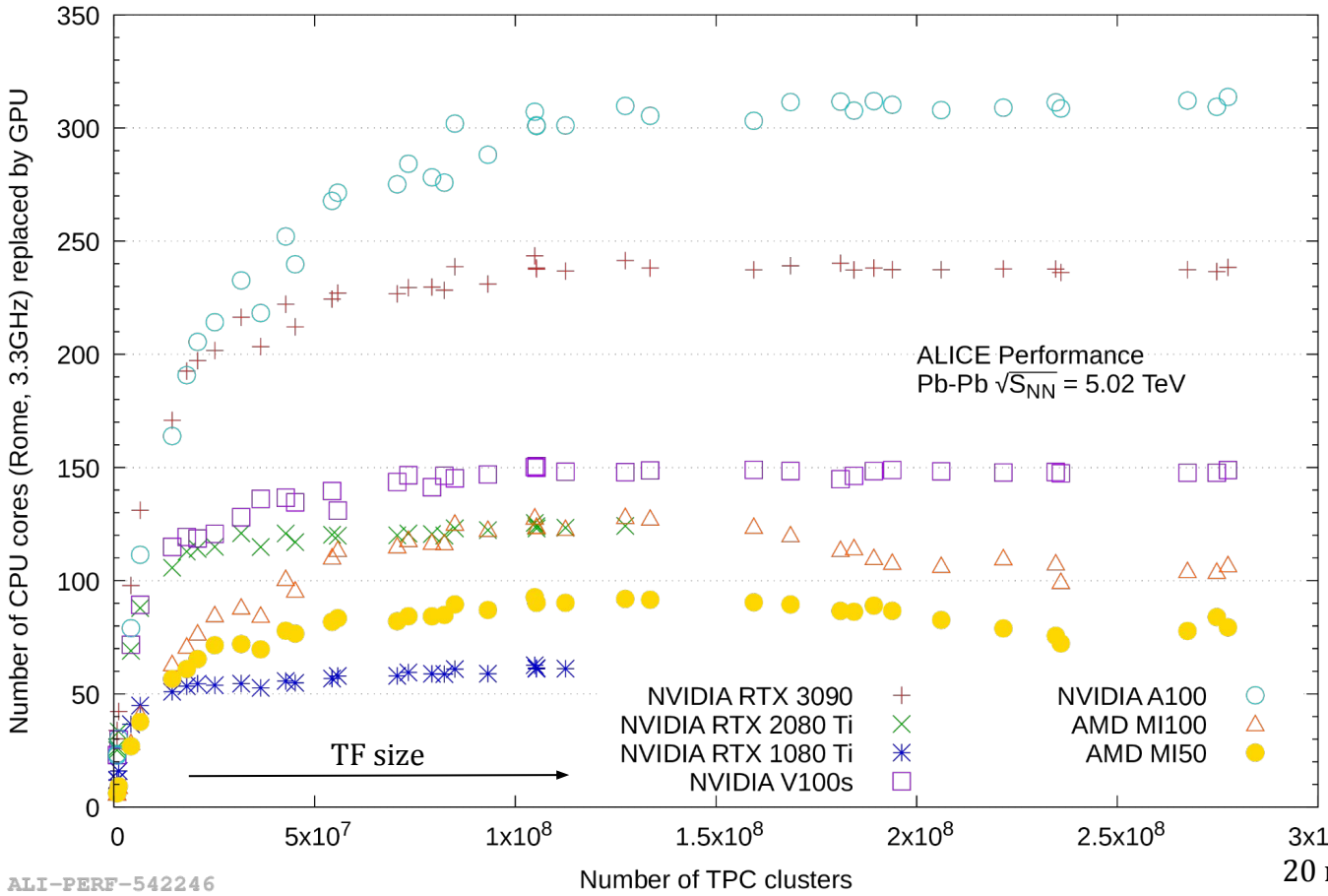
ALI-PERF-542242

20 ms = 256 orbits

ALICE - sync. TPC tracking on GPUs



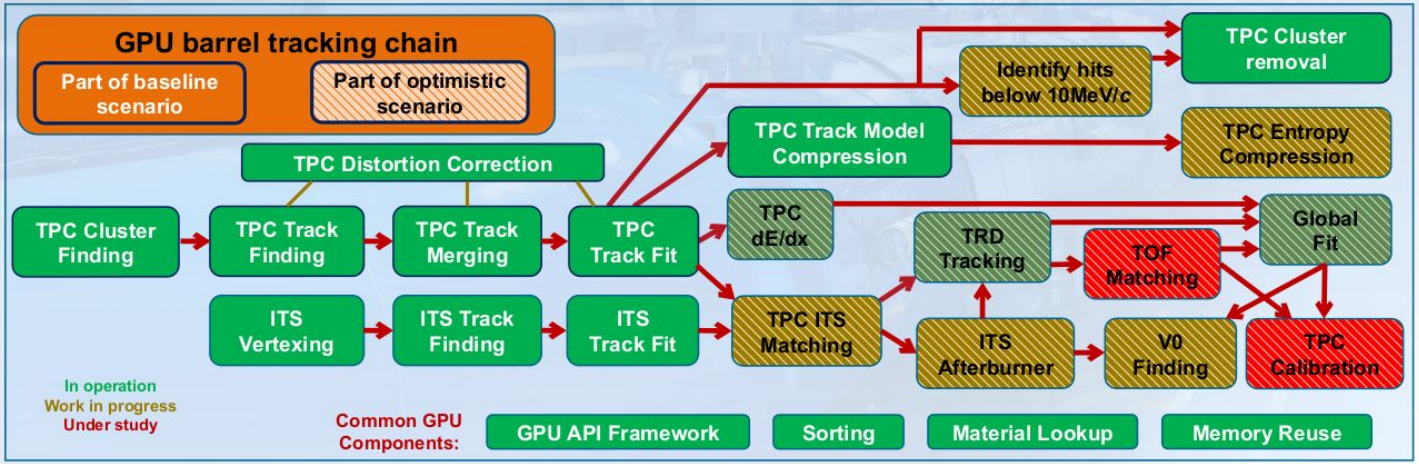
mfaggin@cern.ch



20 ms = 256 orbits

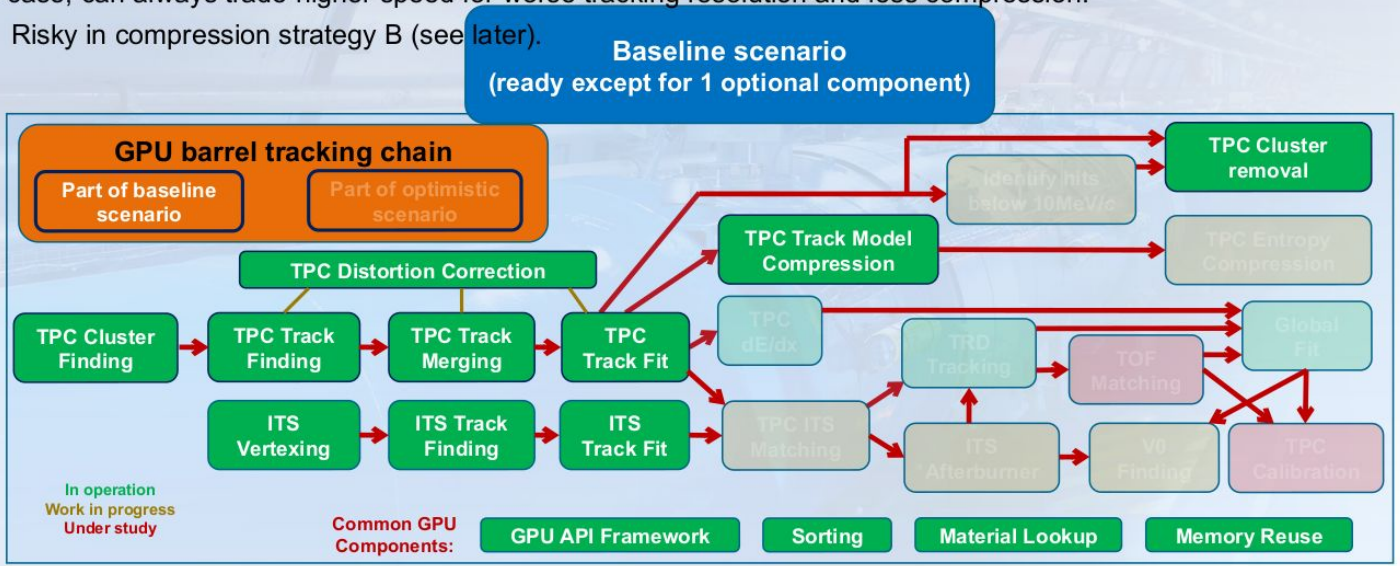
Central Barrel Global Tracking Chain

- Overview of reconstruction steps considered for GPU-offload:
  - Mandatory **baseline scenario** includes everything that must run on the GPU during synchronous reconstruction.
  - **Optimistic scenario** includes everything related to the barrel tracking.



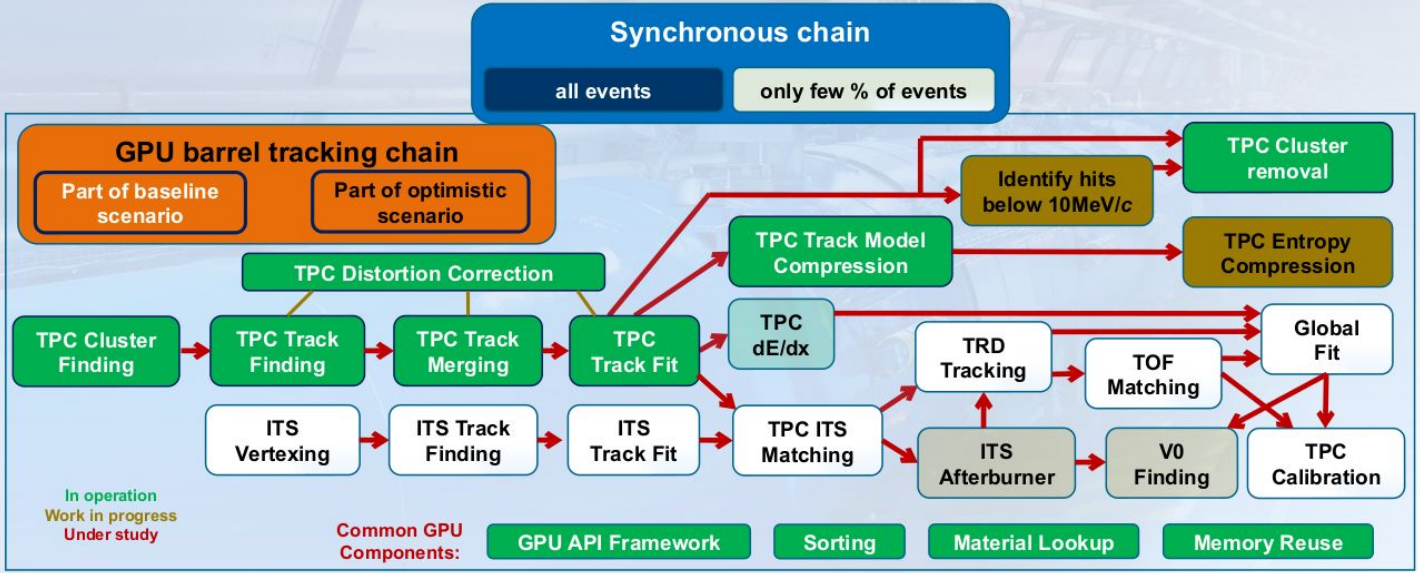
Central Barrel Global Tracking Chain

- Baseline scenario fully implemented (module some improvements e.g. distrotion correction).
  - Not mandatory to speed up the synchronous GPU code further, but we should try nonetheless.
  - If we add / improve reconstruction steps, we have to speed it up accordingly to remain in the 2000 GPU budget.
  - Worst case, can always trade higher speed for worse tracking resolution and less compression.
    - Risky in compression strategy B (see later).



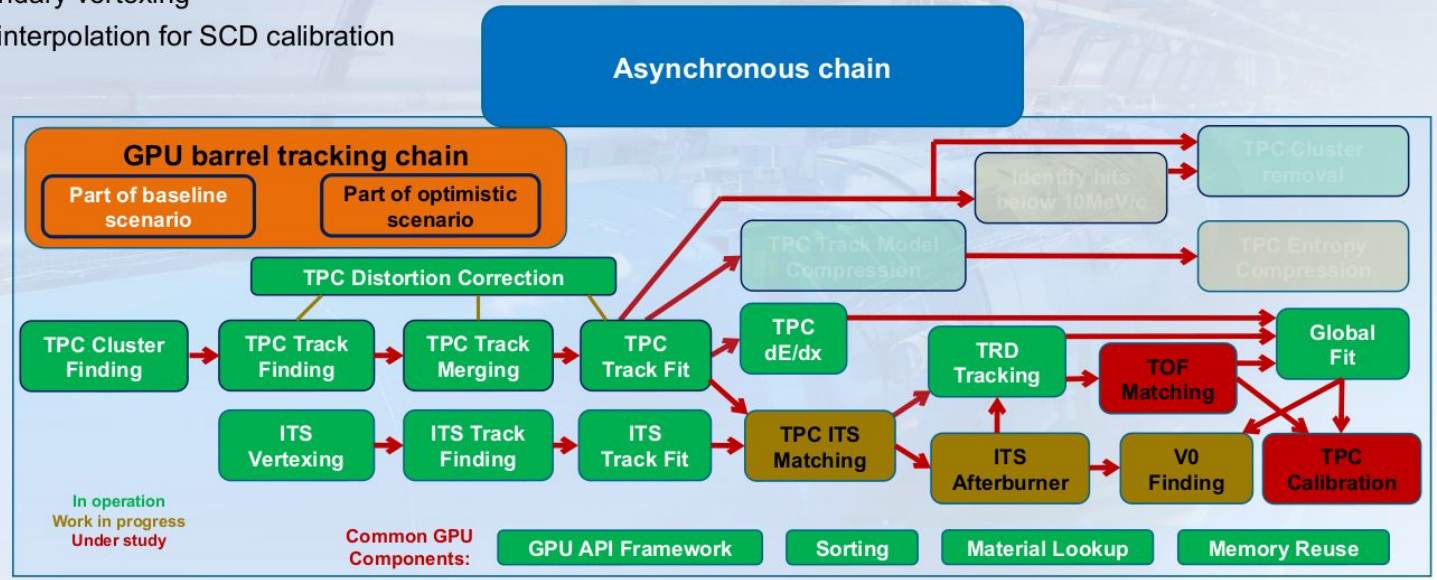
Central Barrel Global Tracking Chain

- Baseline scenario fully implemented (module some improvements e.g. distrotion correction).
  - 2 optional parts still being investigated for sync. reco on GPU: TPC entropy encoding / Looper identification < 10 MeV.



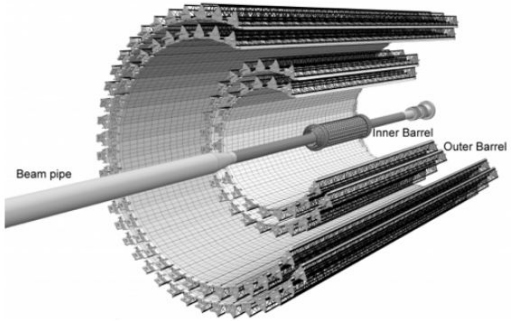
Central Barrel Global Tracking Chain

- Several steps missing in asynchronous reconstruction:
  - Matching to ITS
  - Matching to TOF
  - Secondary vertexing
  - TPC interpolation for SCD calibration



ITS reconstruction in Run 3

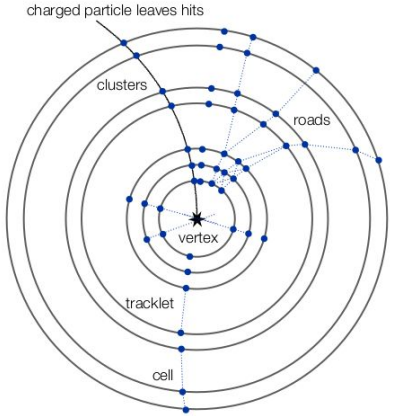
- A new upgraded Inner Tracking System
  - A cylindrical silicon detector, is the innermost detector of the apparatus
  - Thin layers, 12.5 billion pixels and 10 m^2 of sensitive area
  - Provide spatial information in the form of clusters of fired pixels
- Continuous readout, continuous track reconstruction
  - Digital signals from pixels are clusterised and compressed
  - Timeframes are divided into Readout Frames (ROF): ~4μs
  - Collision data can split across multiple adjacent rofs
- New standalone vertexing and tracking algorithm
  - Single implementation steered via configuration (sync/async, collision system, ...)
  - During synchronous reconstruction focuses on primaries (7 clusters-long tracks)
  - During asynchronous reconstruction is sensitive to secondaries and tracks lower p_T
  - In- and cross-bunch pile-up ready



Timeframe			
ROF 0	ROF 1	ROF ...	ROF N
compressed clusters of pixels	compressed clusters of pixels	...	compressed clusters of pixels

ITS vertexing and tracking

- Primary vertex seeding
  - Look for correlations between hits in the three innermost ITS layers
  - Combinatorial matching followed by linear extrapolations of *tracklets*
  - Unsupervised clustering to find the collision point(s)
- Track finding and track fitting
  - Uses vertex position to reduce the combinatorics in matching the hits
  - Connect segments of tracks, the *cells*, into a tree of candidates: *roads*
  - Kalman filter to fit tracks from candidates and apply quality cuts
- The algorithm is decomposable into multiple parallelisable steps
  - ALICE Data Processing Layer manages parallel Timeframe scheduling
  - Each ROF can be processed independently(*)
  - In-frame combinatorics can be processed simultaneously
  - Current CPU implementation can profit from multi-threaded sections

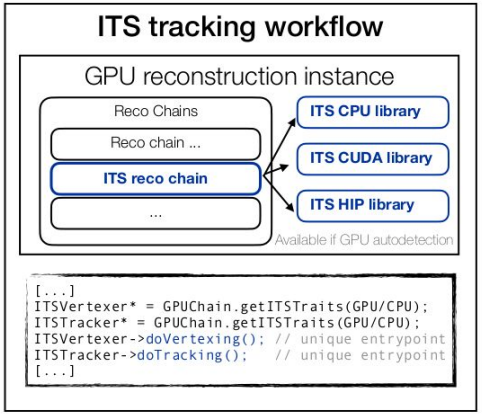


Timeframe			
ROF 0	**ROF 1**	**ROF ...**	**ROF N**
- clusters - vertices - tracklets - cells - roads **- tracks**	- clusters - vertices - tracklets - cells - roads **- tracks**	...	- clusters - vertices - tracklets - cells - roads **- tracks**

(*) *Information from adjacent ROFs can be used to recover from information splitting*

A parallel implementation using GPUs

- Accelerate the processing using massively parallel architectures
  - Promising porting of some routines based on CUDA and OpenCL in the past
- Today: offload the whole vertexing and tracking on GPUs
  - Release the corresponding CPU cycles, improving resource usage efficiency
  - Integrate it into the broader reconstruction GPU chain by extending its coverage
- First phase: load and operate standalone GPU tracking for ITS
  - Mainstream reconstruction framework provides the interface for GPU lib loading
  - The ITS GPU library fully manages graphics card resource
  - Easy-to-contribute: plain C++ and CUDA code, focus on routines development
  - Supports CUDA and HIP with a single code base, no compatibility layer
- Second phase: build a GPU reconstruction chain including ITS
  - Centrally manage GPU memory and kernel scheduling for deeper integration
  - Easier to later add additional steps like the ITS-TPC matching

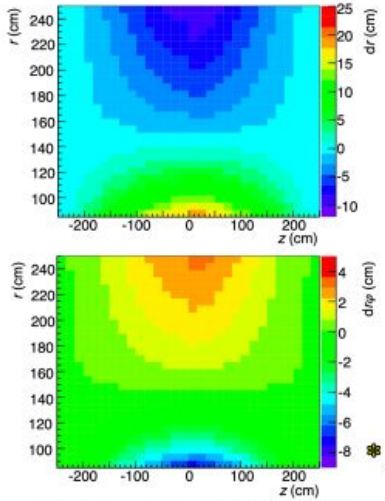


Space-Charge Distortions in the TPC

TPC GEM configuration designed to reduce to the minimum the ion backflow (< 1%)

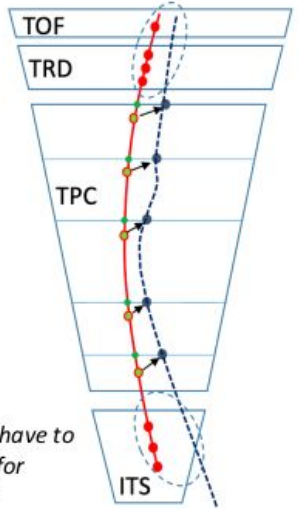
- Still, positive charge accumulating and moving in the TPC → modified E-field → distortions in the TPC

Expectations for Pb-Pb @ 50 kHz:



http://cds.cern.ch/record/1622286/

Run 2 strategy for average (*) distortions:



→ Interpolation of refitted ITS, TRD and TOF **track segments** to the TPC as **reference points** for the **true track position** — sync

→ Collect ΔY, ΔZ between **distorted clusters** and **references** in TPC sub-volumes (voxels) — sync

→ Extract 3D distortion vector in every voxel — offline

→ Use during asynchronous reconstruction — async

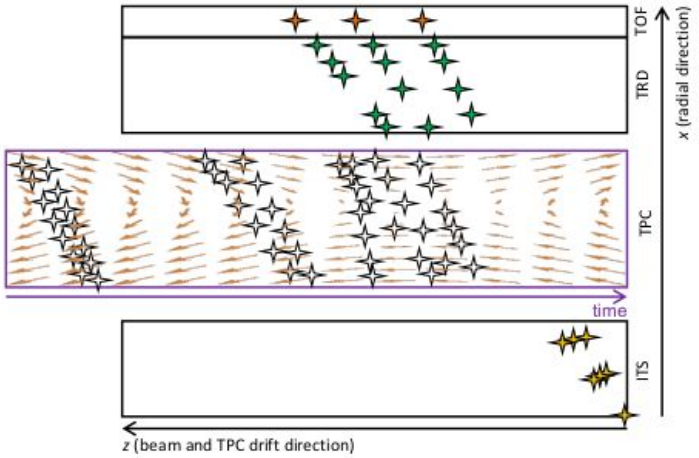
* *Fluctuations will have to be accounted for separately*

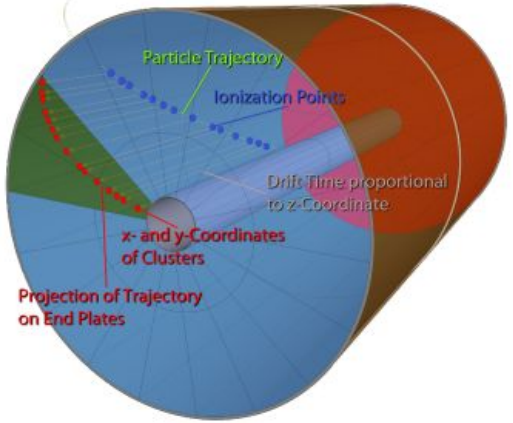
Full ITS-TPC-TRD-TOF reconstruction of a fraction (~0.6%) of the tracks at synchronous stage

Tracking in the central barrel

Challenge in Run 3 + 4!

- **overlap** of multiple collisions (5 collisions in the TPC drift time @50 kHz Pb-Pb)
- with TPC clusters without a well-defined *z* coordinate, but **just a time** (*t*)
- presence of **distortion** corrections that are position dependent



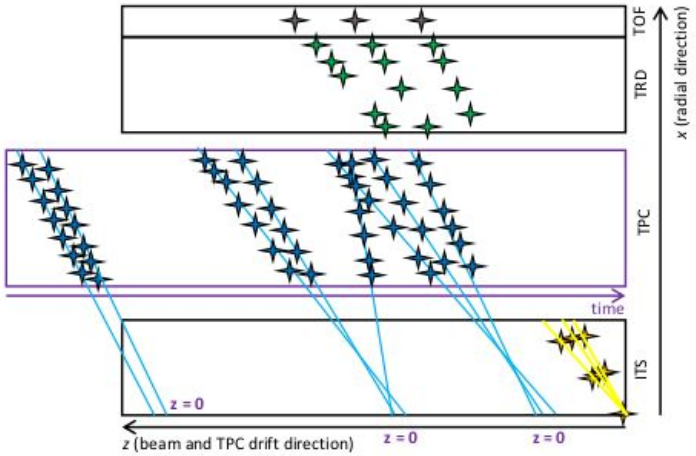


$$z = (t - t_{vertex}) * v_{drift}$$

Tracking in the central barrel

Challenge in Run 3 + 4!

- **overlap** of multiple collisions (5 collisions in the TPC drift time @50 kHz Pb-Pb)
- with TPC clusters without a well-defined *z* coordinate, but **just a time** (*t*)
- presence of **distortion** corrections that are position dependent

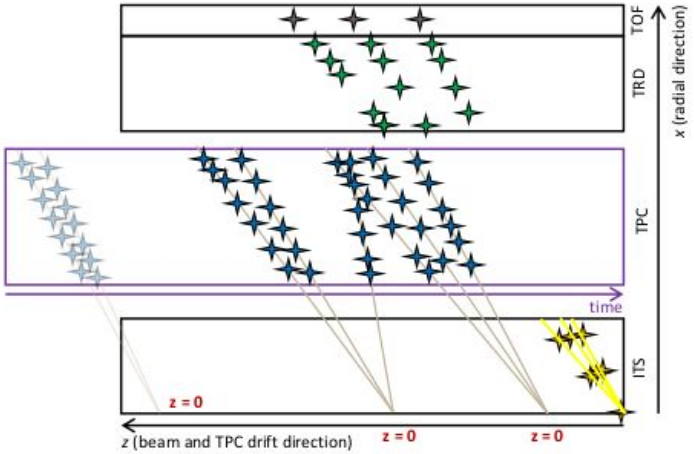


- Standalone ITS tracking
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to $x = 0$, define $z = 0$ as if the track was primary
  → good enough at this stage (sync!)
- Track following to find missing clusters

Tracking in the central barrel

Challenge in Run 3 + 4!

- **overlap** of multiple collisions (5 collisions in the TPC drift time @50 kHz Pb-Pb)
- with TPC clusters without a well-defined *z* coordinate, but **just a time** (*t*)
- presence of **distortion** corrections that are position dependent

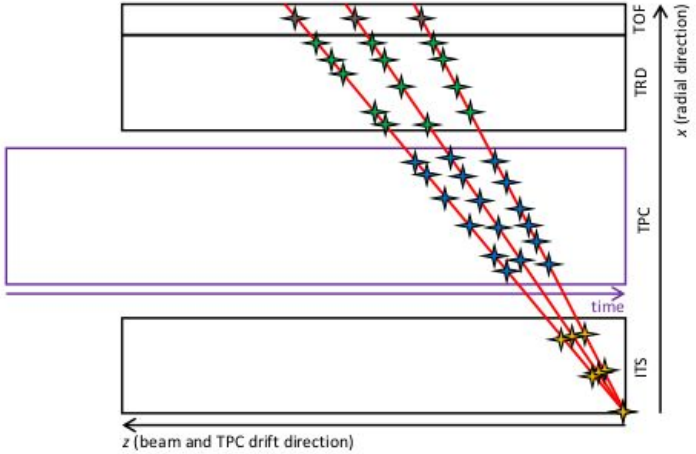


- Standalone ITS tracking
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to $x = 0$, define $z = 0$ as if the track was primary
  → good enough at this stage (sync!)
- Track following to find missing clusters
- Refine $z = 0$ estimate, refit track with best precision
- Find ITS-TPC track compatibility using times

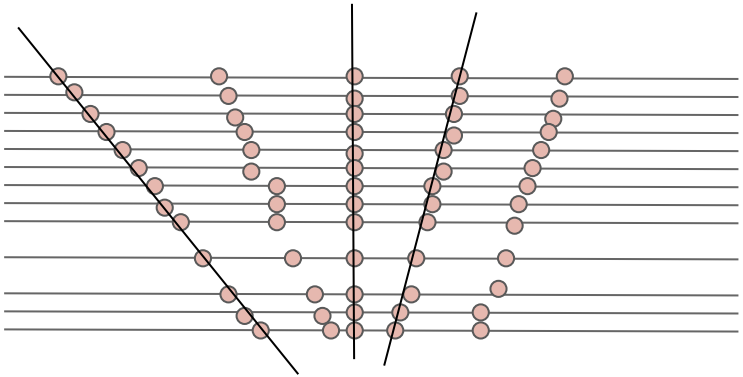
Tracking in the central barrel

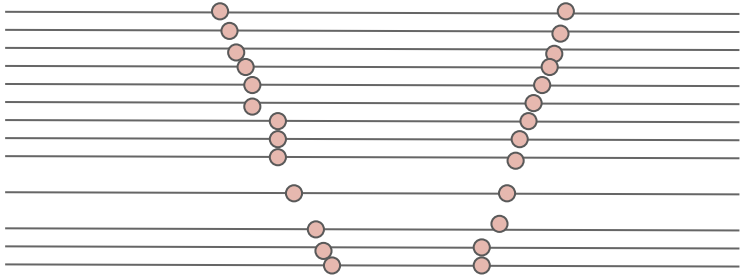
Challenge in Run 3 + 4!

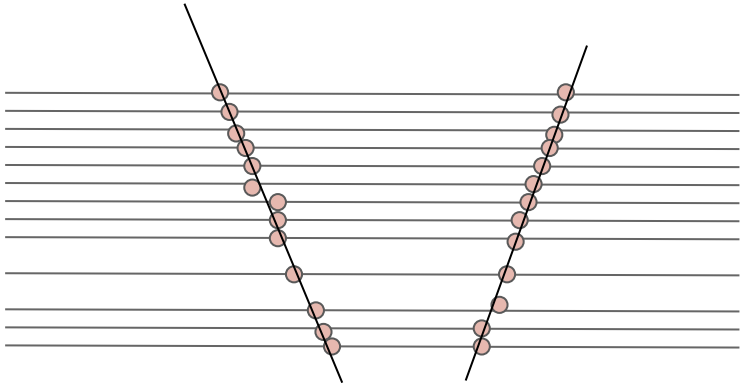
- **overlap** of multiple collisions (5 collisions in the TPC drift time @50 kHz Pb-Pb)
- with TPC clusters without a well-defined *z* coordinate, but **just a time** (*t*)
- presence of **distortion** corrections that are position dependent

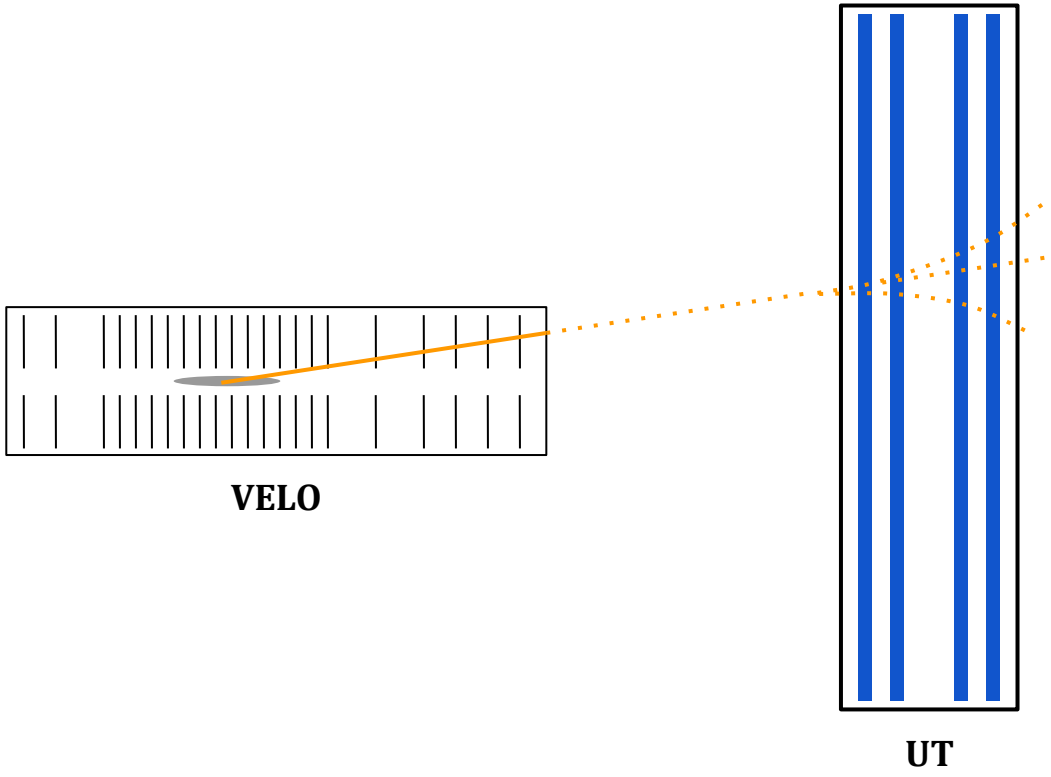


- Standalone ITS tracking
- Standalone TPC tracking, scaling *t* linearly to an arbitrary *z*.
- Extrapolate to $x = 0$, define $z = 0$ as if the track was primary → good enough at this stage (sync!)
- Track following to find missing clusters
- Refine $z = 0$ estimate, refit track with best precision
- Find ITS-TPC track compatibility using times
- Match TPC track to ITS track, fixing *z*-position and *t* of the TPC track
- Refit ITS + TPC track outwards and inwards
- Prolong into TRD / TOF








VELO
UT