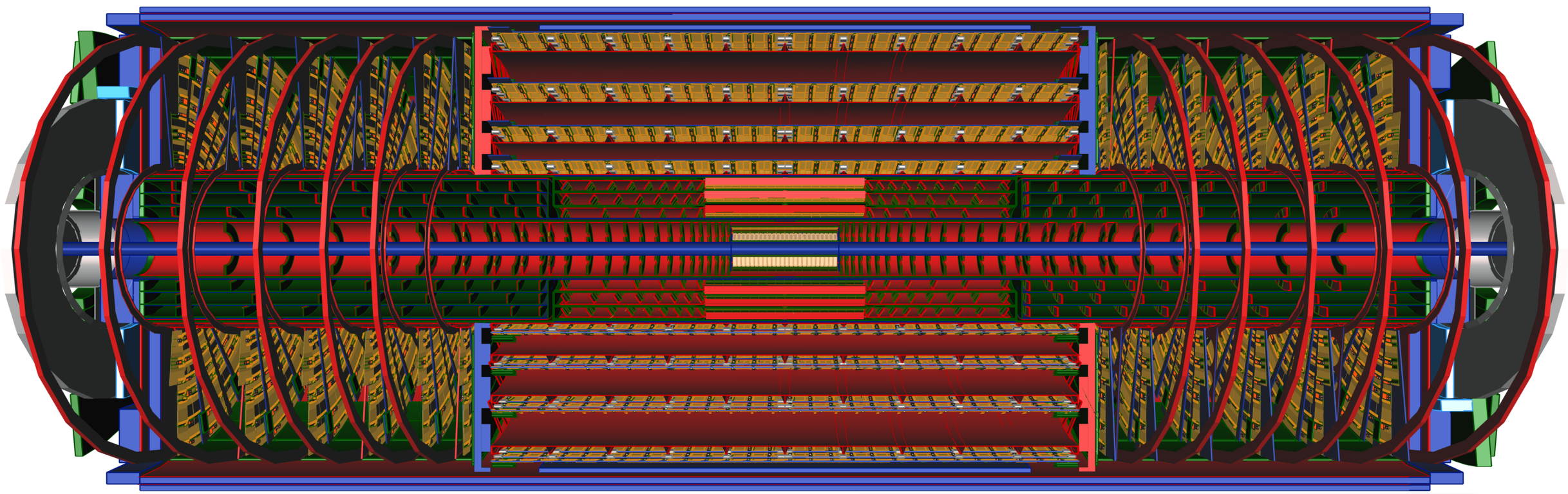
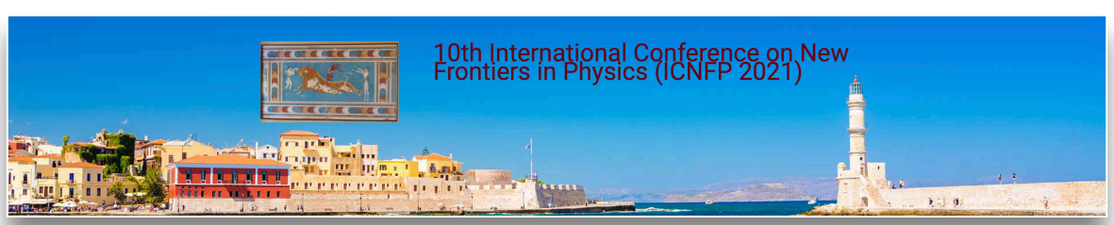
ATLAS Track and Vertex Reconstruction for Run-3 and High-Luminosity LHC

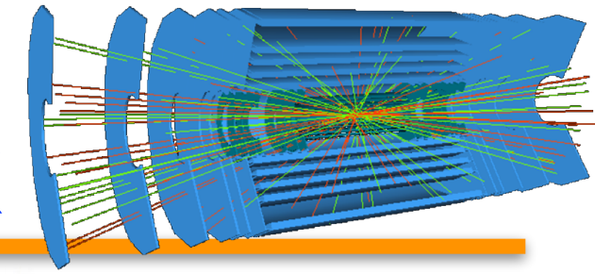
by Markus Elsing on behalf of the ATLAS Collaboration



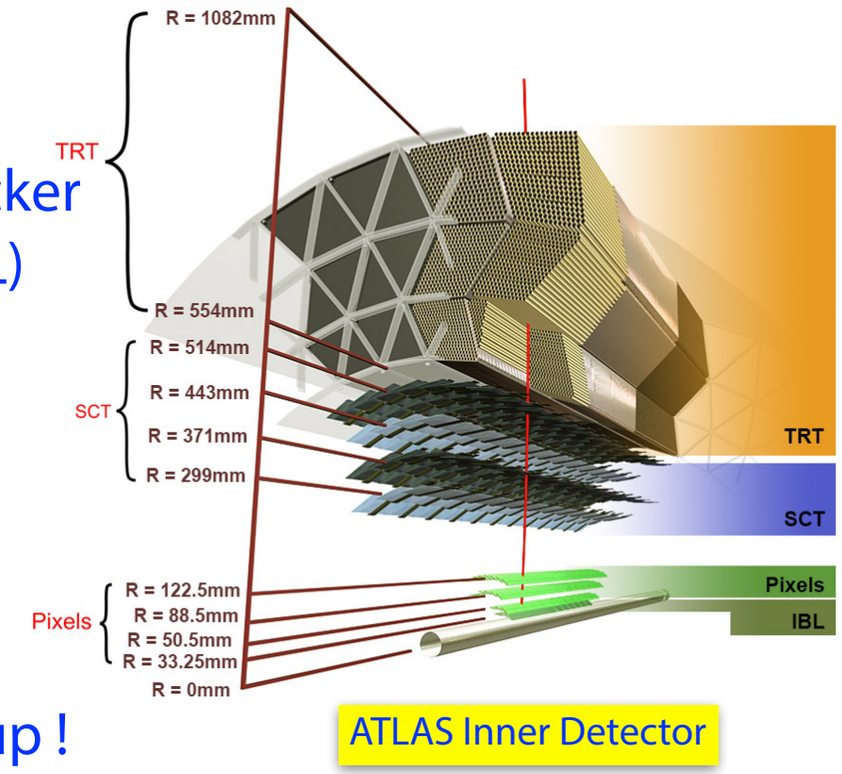
ATLAS ITk : HL-LHC upgrade of the inner tracking system



Experience from Run-2 and Outlook

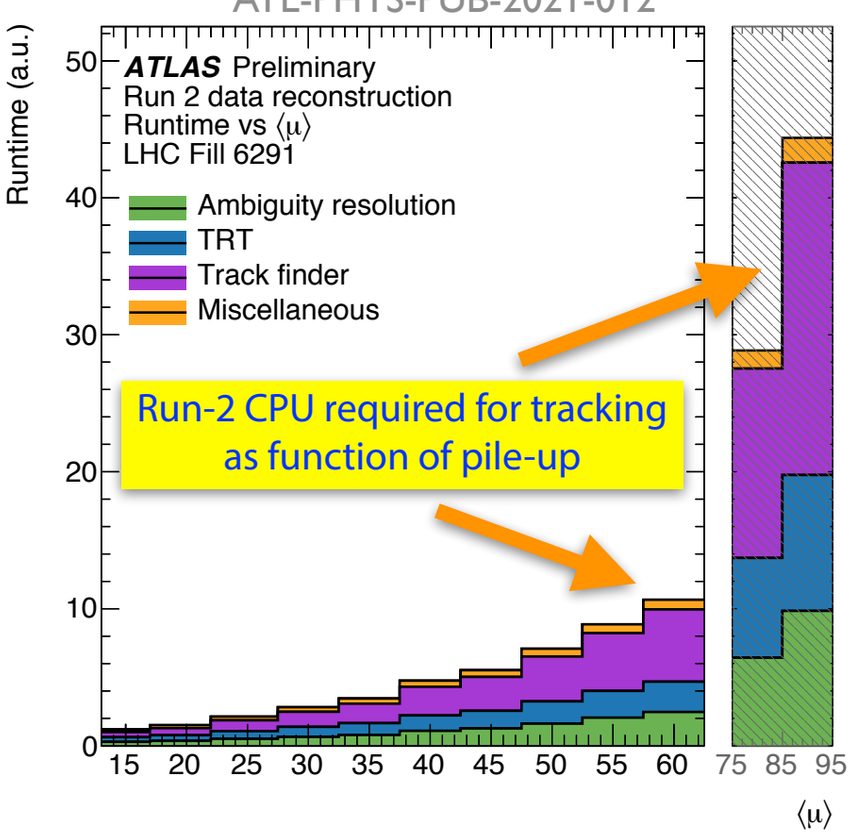


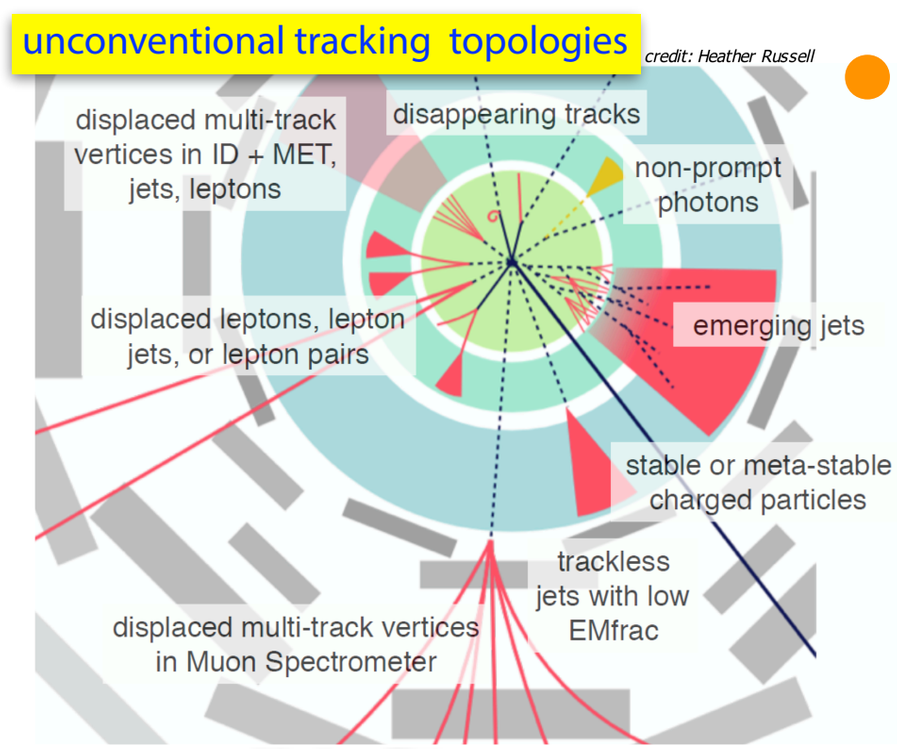
- current ATLAS Inner Detector
  - ➡ 4 Pixel layers, 4 double sided Strips, Transition Radiation Tracker
    - was upgraded in Long Shutdown-1 with 4th Pixel layer (IBL)
  - ➡ designed for excellent performance at pile-up of 23
    - during Run-2 operated routinely at pile-up well above 35



- preserving performance is a challenge !
  - ➡ tracking and vertexing needs to be robust against pile-up
  - ➡ CPU required for reconstruction increases rapidly with pile-up !

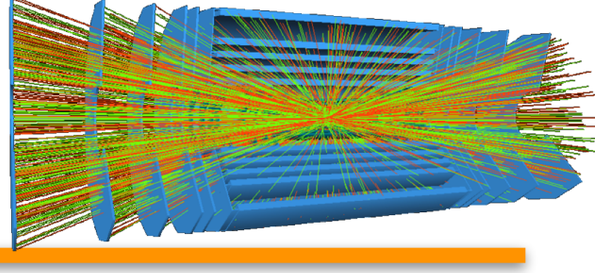
ATL-PHYS-PUB-2021-012



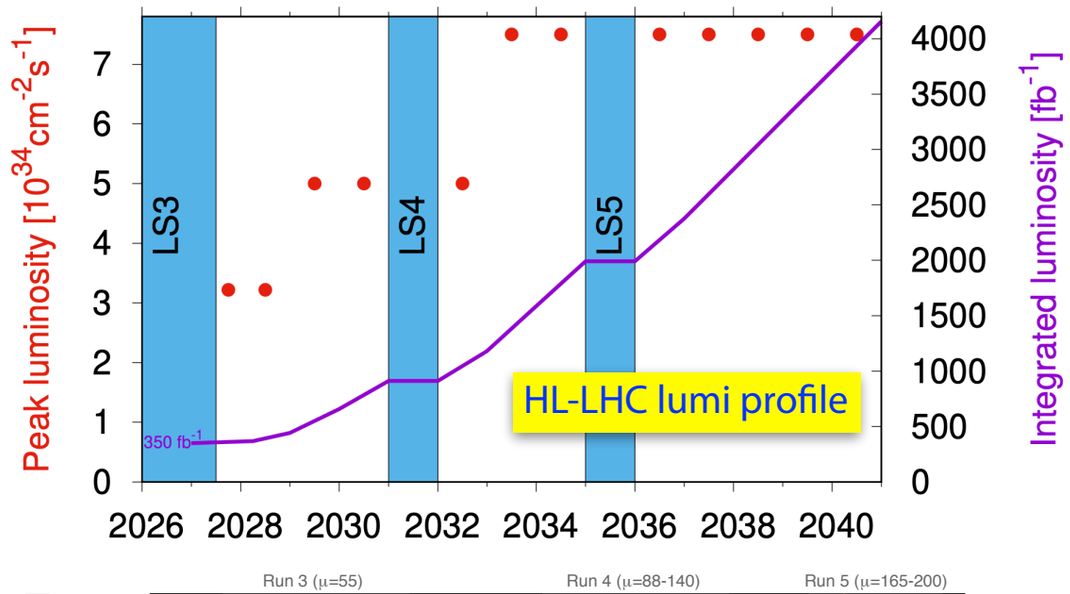


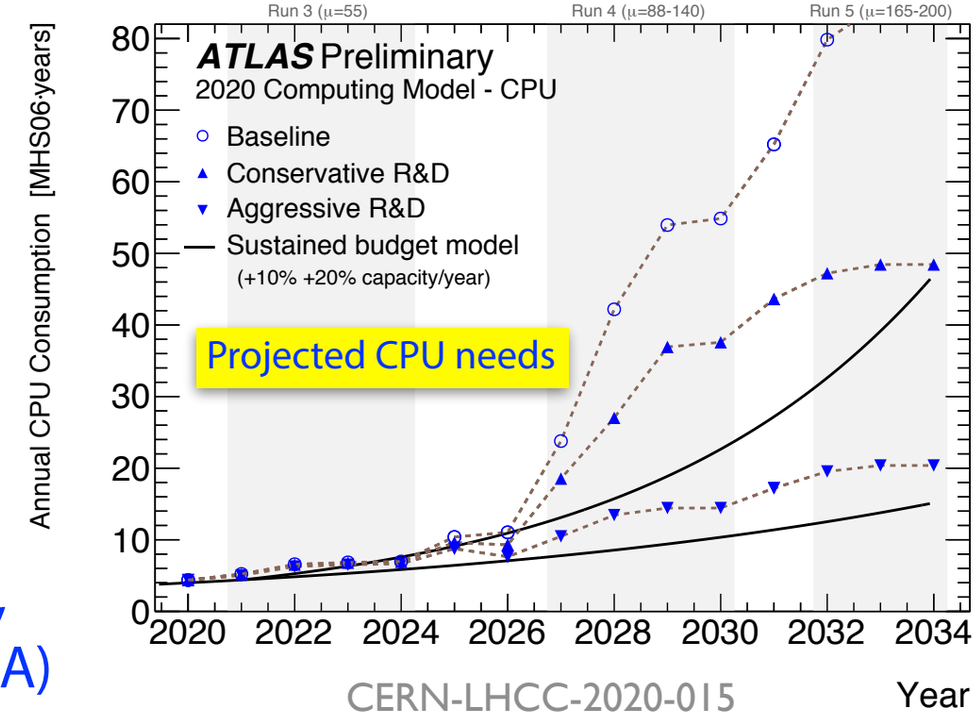
- large radius tracking
  - ➡ searches for long-lived particles !
  - ➡ significant CPU for additional tracking pass
  - ➡ not possible in Tier-0 during Run-2
  - ➡ required dedicated stream, processed on the Grid

Outlook to Run-3 and HL-LHC



- Run-3 expected av. pile-up is ~50
  - ➡ double total integrated luminosity until 2025
  - ➡ end-of-lifetime of current Inner Detector
- goal of LH-LHC is to collect 4 ab^{-1}
  - ➡ average pile-up will rise to 140 to 200 !
  - ➡ radiation levels and data rates unprecedented
  - ➡ ATLAS Inner Tracker (ITk) upgrade (all silicon)
- a challenge !
  - ➡ require excellent tracking performance, despite harsh pile-up environment
  - ➡ CPU for reconstruction and disk space needs will exceed computing budget extrapolations
  - ➡ computing technology becoming heterogenous, with many core CPUs and accelerators (GPU, FPGA)



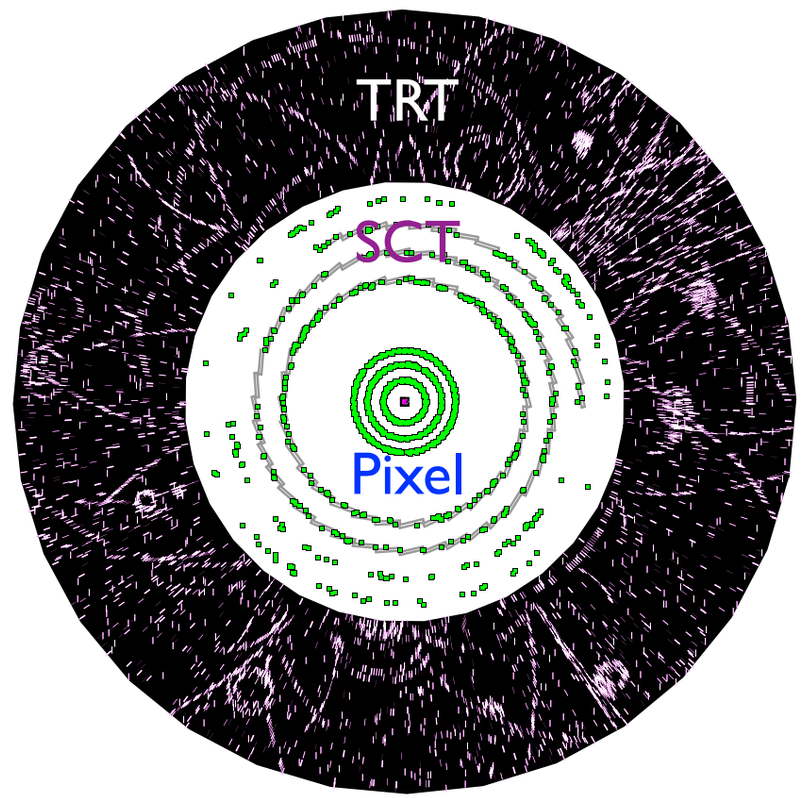


- tracking developments for Run-3 are also a preparation towards HL-LHC

Run-2(3) Tracking and Vertexing Chain

pre-processing

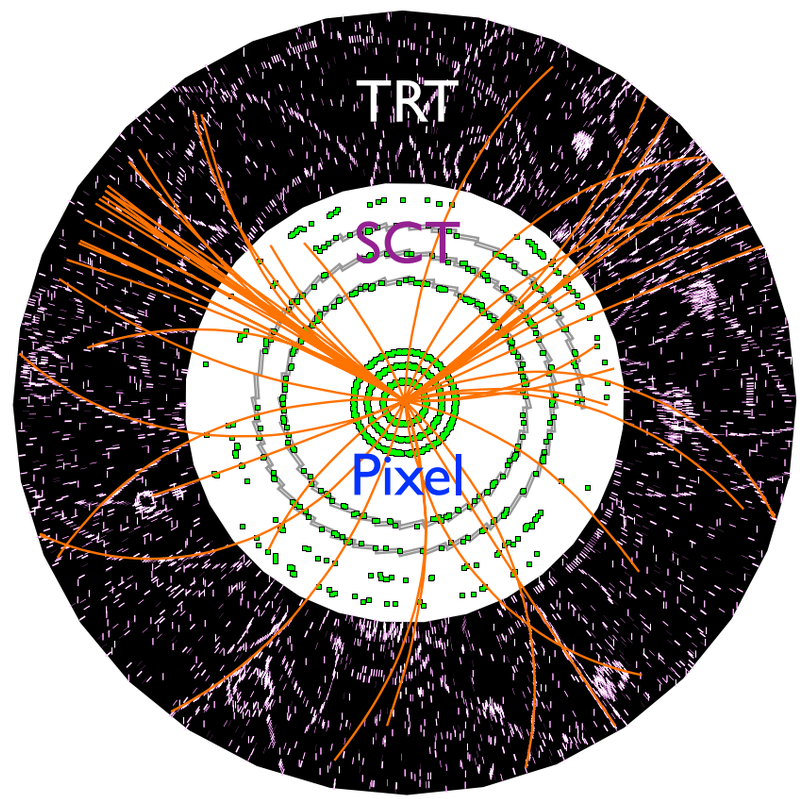
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation



Run-2(3) Tracking and Vertexing Chain

pre-processing

- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation



combinatorial track finder

- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

ambiguity solution

- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

extension into TRT

- ➡ progressive finder
- ➡ refit of track with Brem.
- ➡ scoring and selection cuts

Run-2(3) Tracking and Vertexing Chain

pre-processing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

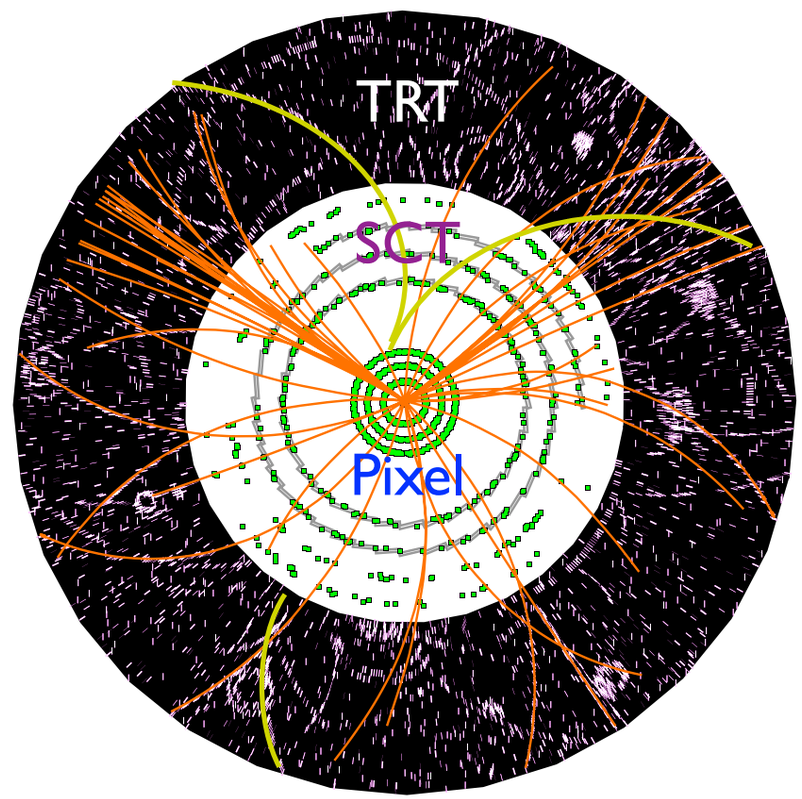
ambiguity solution
- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

extension into TRT
- ➡ progressive finder
- ➡ refit of track with Brem.
- ➡ scoring and selection cuts

TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks



Run-2(3) Tracking and Vertexing Chain

pre-processing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

ambiguity solution
- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

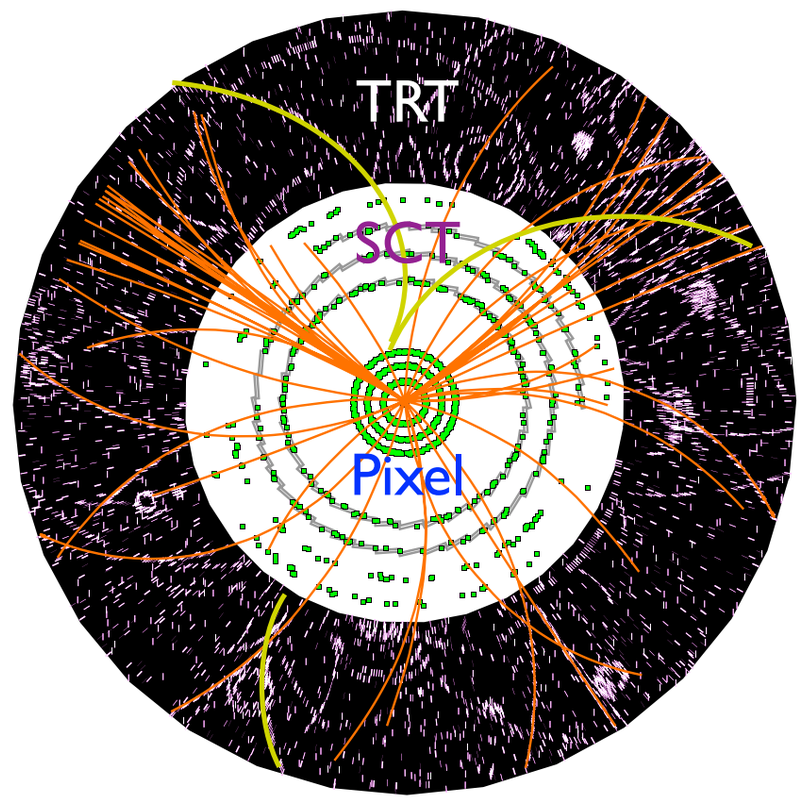
extension into TRT
- ➡ progressive finder
- ➡ refit of track with Brem.
- ➡ scoring and selection cuts

TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

tracklet finder
- ➡ muon candidates |η|>2.5
- ➡ short tracks



Run-2(3) Tracking and Vertexing Chain

pre-processing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

ambiguity solution
- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

extension into TRT
- ➡ progressive finder
- ➡ refit of track with Brem.
- ➡ scoring and selection cuts

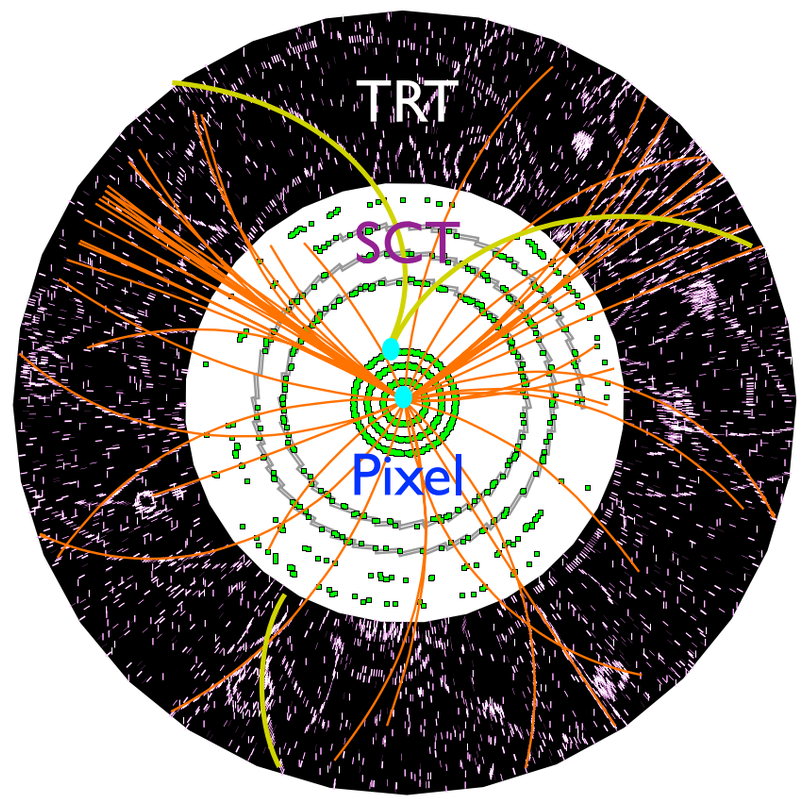
TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

tracklet finder
- ➡ muon candidates |η|>2.5
- ➡ short tracks

vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search



Run-2(3) Tracking and Vertexing Chain

pre-processing
- ➡ Pixel+SCT clustering
- ➡ TRT drift circle formation
- ➡ space points formation

combinatorial track finder
- ➡ iterative :
  1. Pixel seeds
  2. Pixel+SCT seeds
  3. SCT seeds
- ➡ restricted to roads
- ➡ Brem.recovery in EM Regions-of-Interest

ambiguity solution
- ➡ runs hole search
- ➡ scores tracks according to quality
- ➡ NN cluster splitting in jets
- ➡ precise least square fit with Brem.recovery
- ➡ final selection cuts

extension into TRT
- ➡ progressive finder
- ➡ refit of track with Brem.
- ➡ scoring and selection cuts

TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles
- ➡ uses Hough transform

TRT seeded finder
- ➡ from TRT into SCT+Pixels
- ➡ combinatorial finder

ambiguity solution
- ➡ precise fit and selection
- ➡ TRT seeded tracks

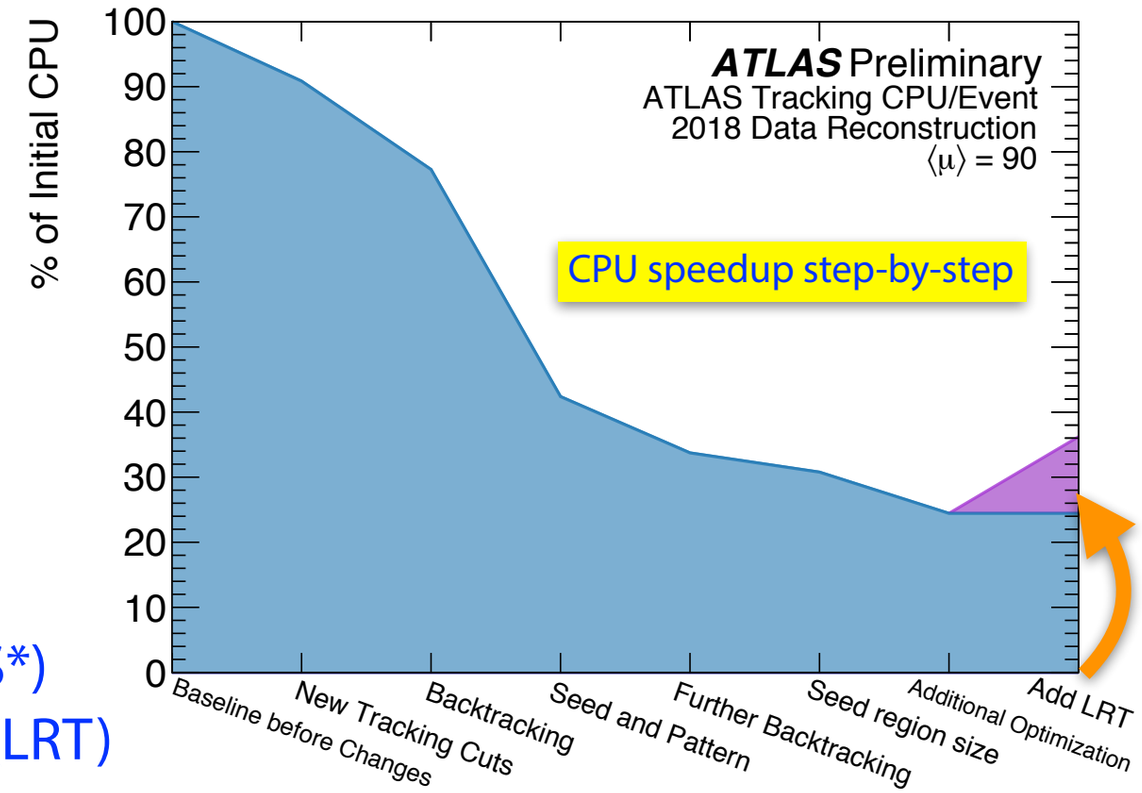
tracklet finder
- ➡ muon candidates |η|>2.5
- ➡ short tracks

vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search

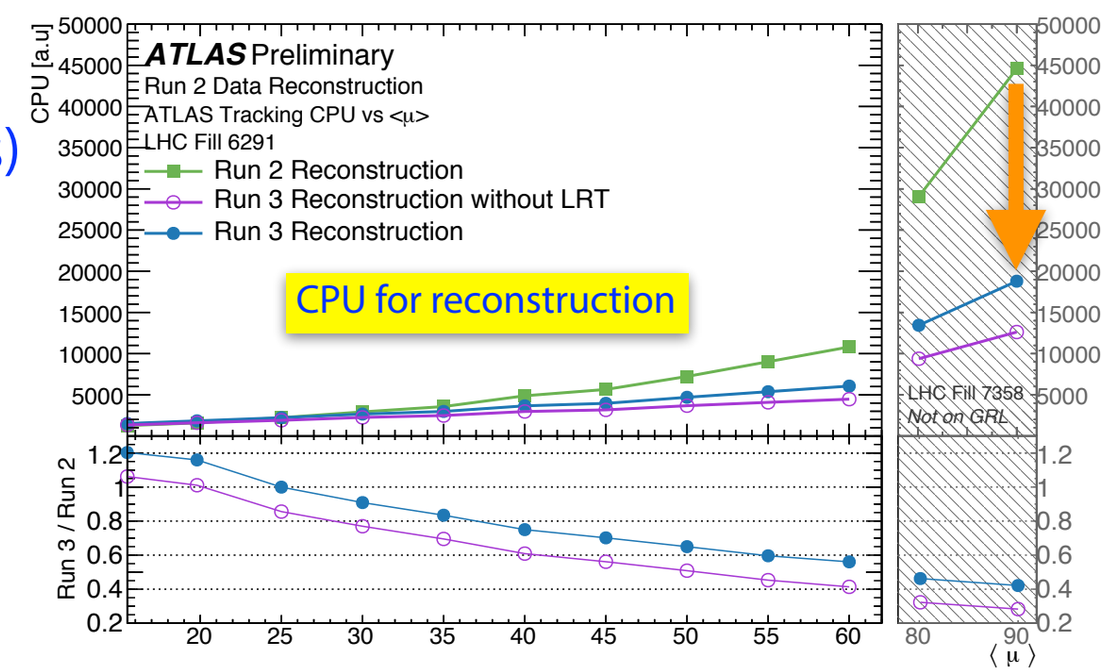
Large Radius Tracking (LRT) as a additional iteration
- ➡ combinatorial track finder, ambiguity solution, extension, secondary vertexing
- ➡ different track selection strategy adopted to event topology

Optimisation of Run-3 Tracking Software

- intensive development programme
  - ➡ software technology changed to multi-threading
  - ➡ optimisation of primary tracking:
    - stricter cuts: at least 8 hits and $|d_0| < 5$ mm
    - Back-Tracking and TRT optimised and seeded using EM calorimeter deposits with $E_T > 6$ GeV
    - seeding and pattern recognition tuning: stricter cuts, narrower roads, seed confirmation
    - novel primary vertex finder (first time using ACTS*)
  - ➡ apply optimisations also to Large Radius Tracking (LRT)



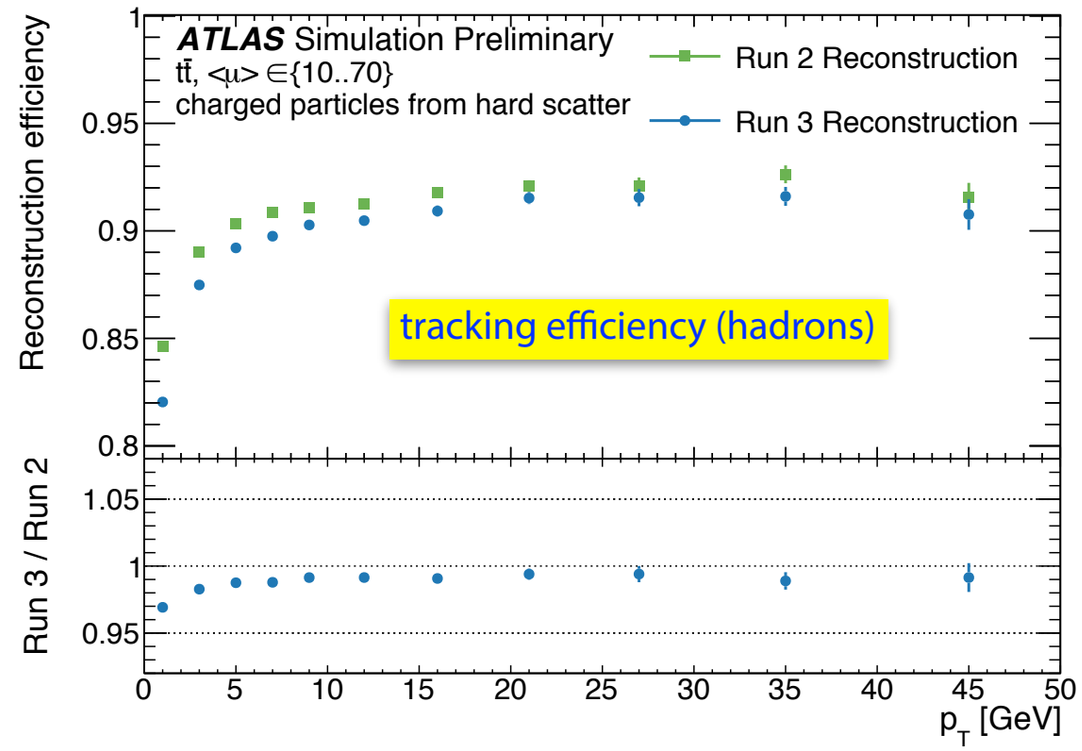
- ×4 speedup for tracking at 90 pile-up
  - ➡ include faster LRT in Tier-0 reconstruction (all events)
    - huge simplification in computing model and reduction in storage
    - exiting prospects for searches !
  - ➡ Run-3 reconstruction overall ×2 faster and improved selection reduces event size by 25%



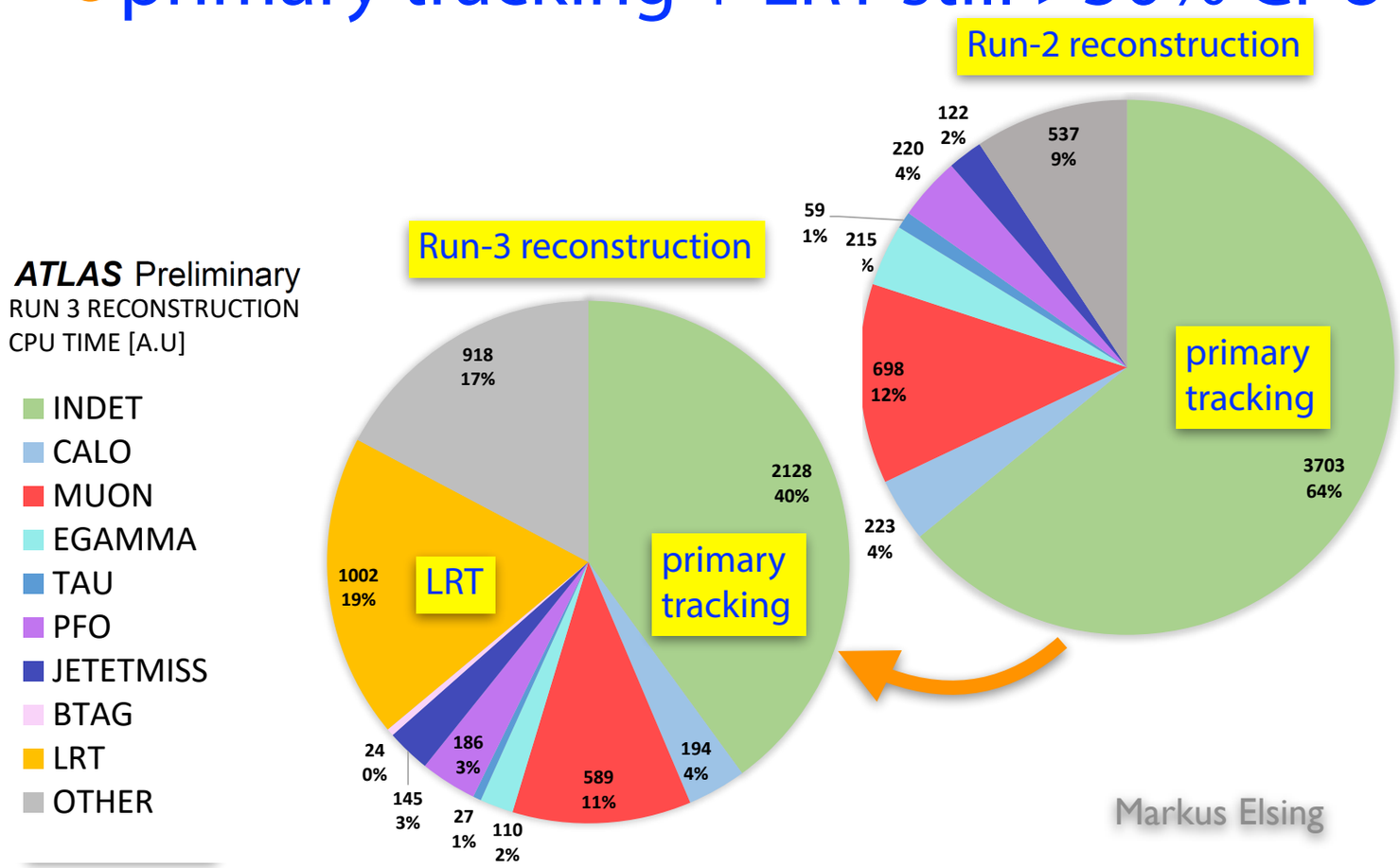
* I'll come to ACTS later

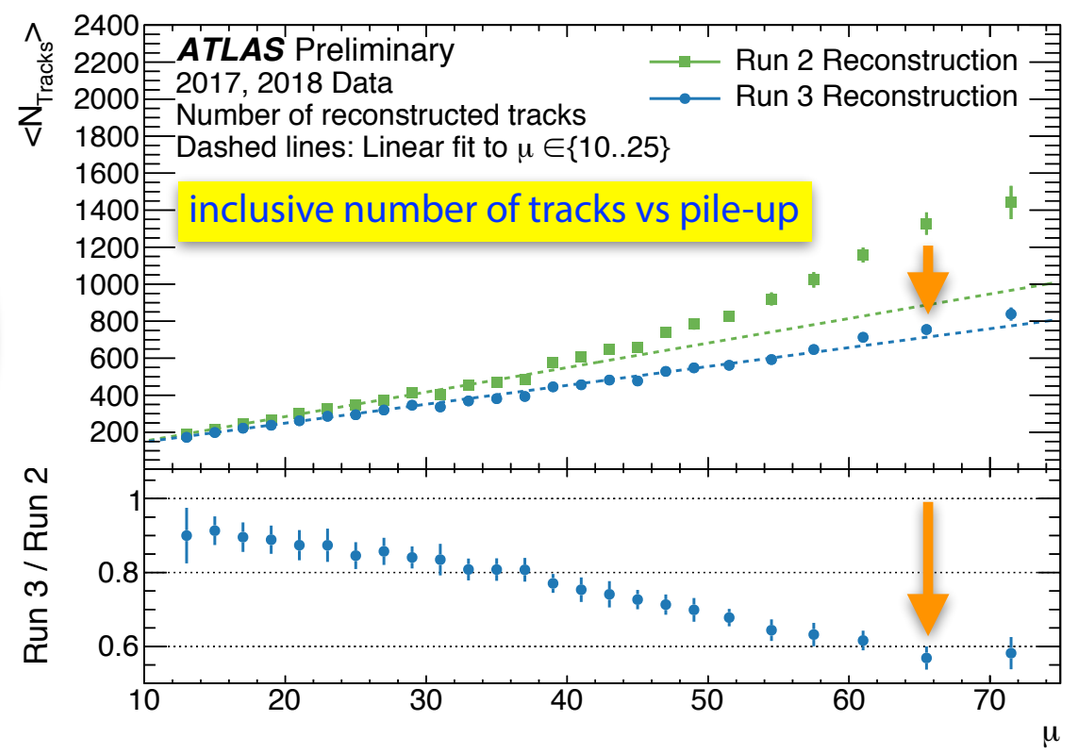
Optimisation of Run-3 Tracking Software

- effect on tracking performance
  - ➡ CPU improvements largely due to early rejection of fakes and secondaries
  - ➡ efficiency loss varies from 1% (4%) at high-(low-)p_T
    [N(hit) ≥ 8 for Run-3, compared to N(hit)≥7 for Run-2]
  - ➡ much more lineare increase in number of tracks vs pile-up indicates improved pile-up robustness



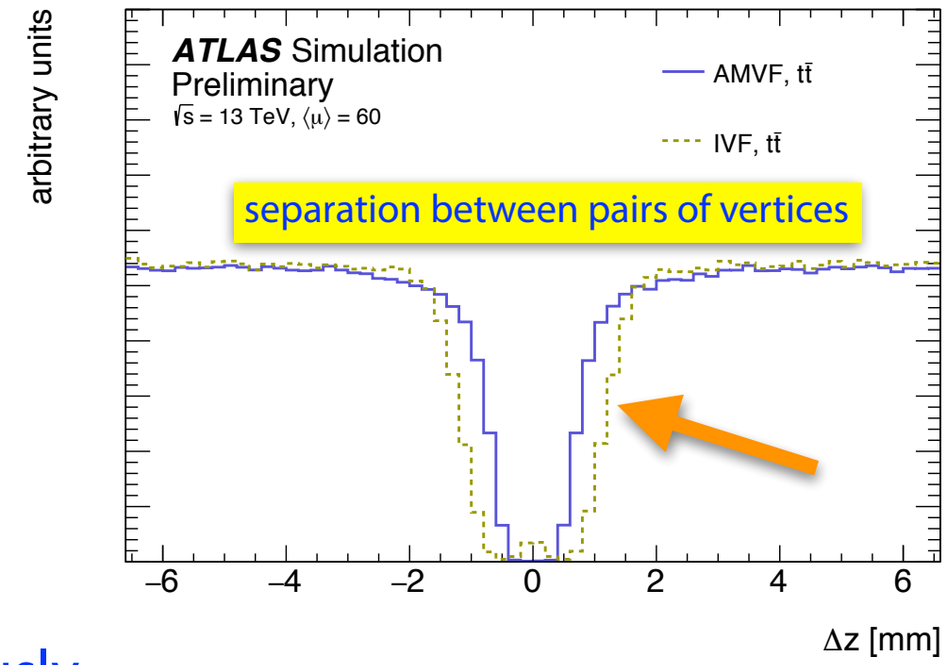
- primary tracking + LRT still >50% CPU





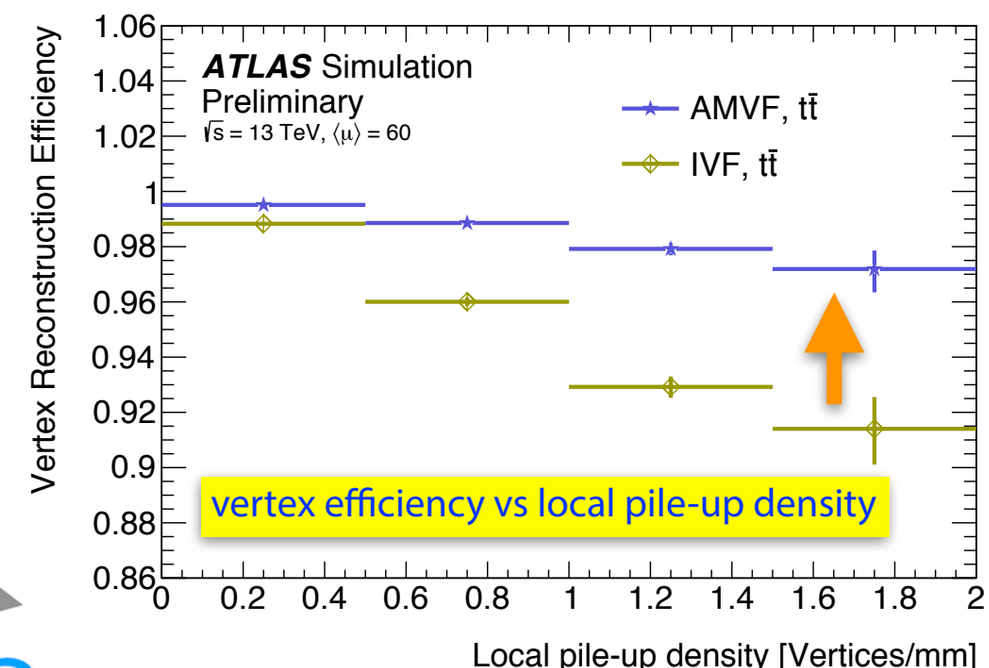
New Run-3 Primary Vertex Reconstruction

- change in vertex finding algorithm:
  - ➡ Run-2: Iterative Primary Vertex Finder (IVF)
  - ➡ Run-3: Adaptive Multi-Vertex Finder (AMVF)



- AMVF reconstruction strategy:
  - ➡ novel gaussian track density vertex seed finder
  - ➡ tracks are associated to vertex candidates with weights according to their distance
  - ➡ vertex candidates share tracks and are fitted simultaneously

- much improved pile-up performance:
  - ➡ improved separation of nearby vertices along z
  - ➡ better efficiency for $t\bar{t}$ and VBF H → 4ν
  - ➡ 20% (10%) better resolution for $t\bar{t}$ (VBF H → 4ν)

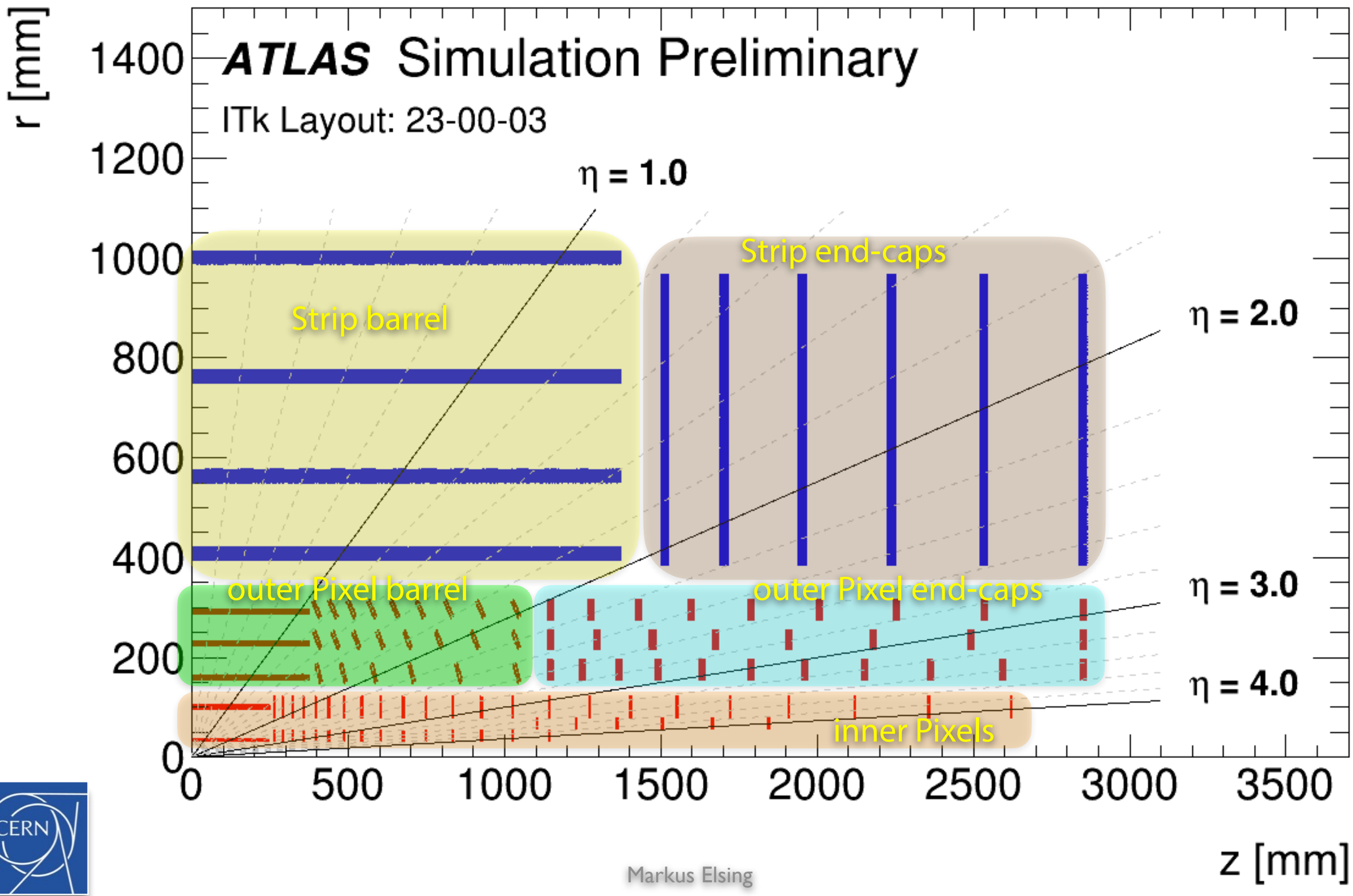


→ number of vertices per unit length within a ±2 mm longitudinal window

- AMVF implemented in ACTS* framework
  - ➡ modern software yields 40% speedup in CPU !

* I'll come to ACTS later

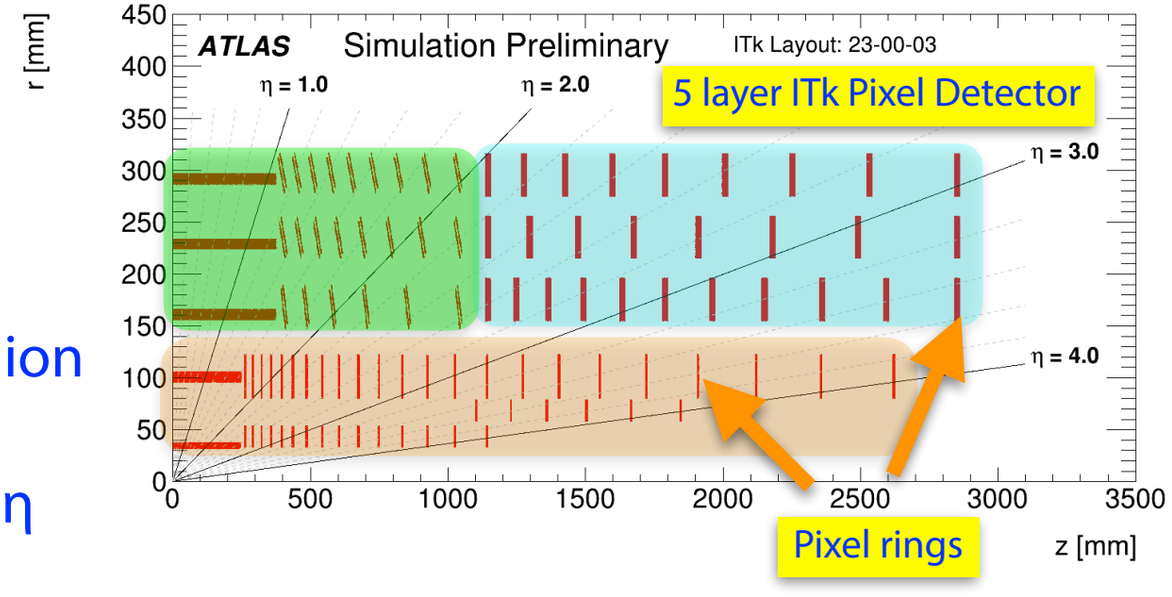
The ATLAS HL-LHC Tracker (ITk) Upgrade



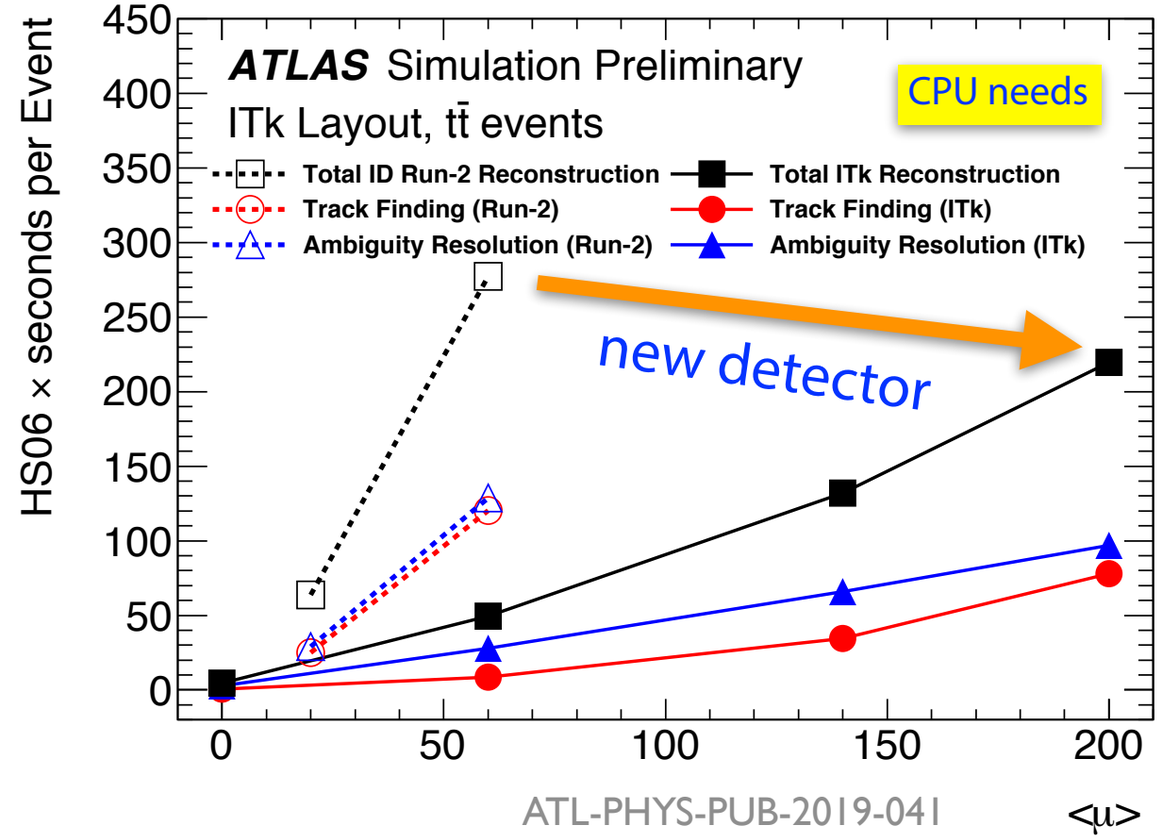
The ATLAS HL-LHC Tracker (ITk) Upgrade

- **designed for fast precision tracking**
  - ➡ 5 Pixel layers, 4 Strip layers (double-sided)
    - extends |η| coverage from 2.5 to 4.0
    - improved granularity and hit redundancy
    - aims to minimise material in active tracking region
  - ➡ 5 layer ITk Pixel Detector with ring design:
    - efficient standalone seed finding in Pixels at all η
  - ➡ flexible Pixel ring placement:
    - keeps hit coverage constant in forward region
    - avoid large gaps between adjacent rings

ATL-PHYS-PUB-2021-024



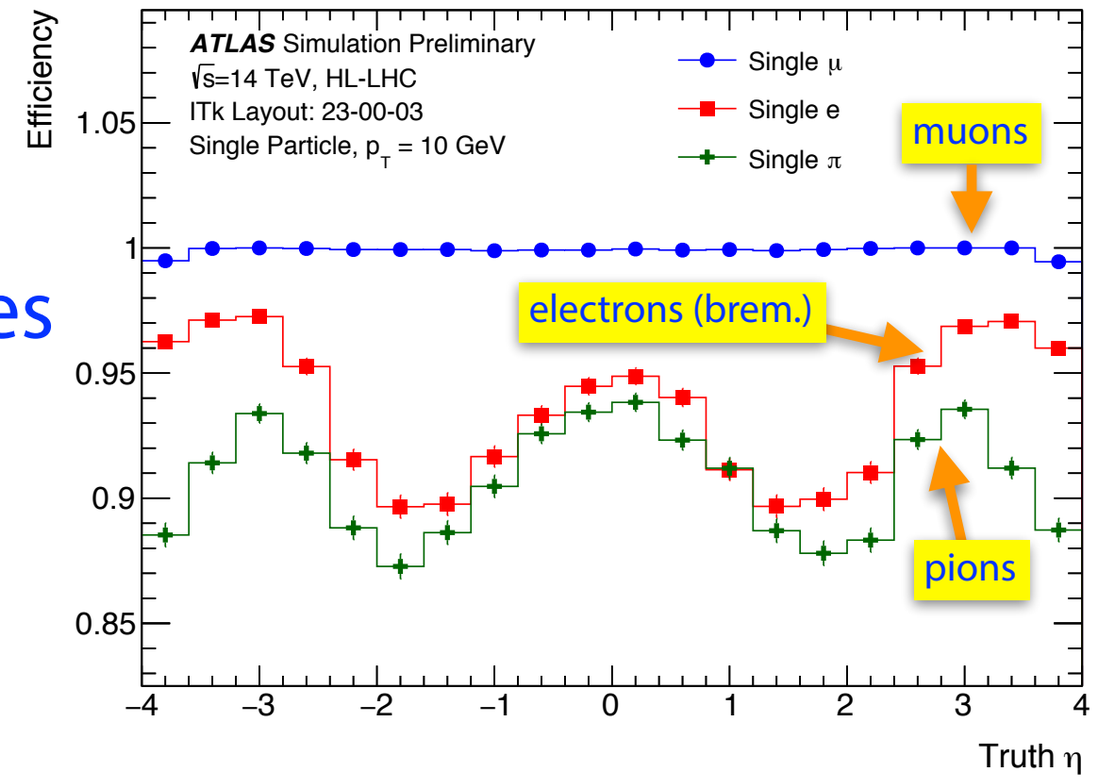
- **default ITk track reconstruction:**
  - ➡ no TRT, hence simplified tracking chain
  - ➡ seed finding only in Pixels or in Strips, followed by track finder and ambiguity solution
  - ➡ significant speed-up of reconstruction at 200 pile-up, with excellent performance

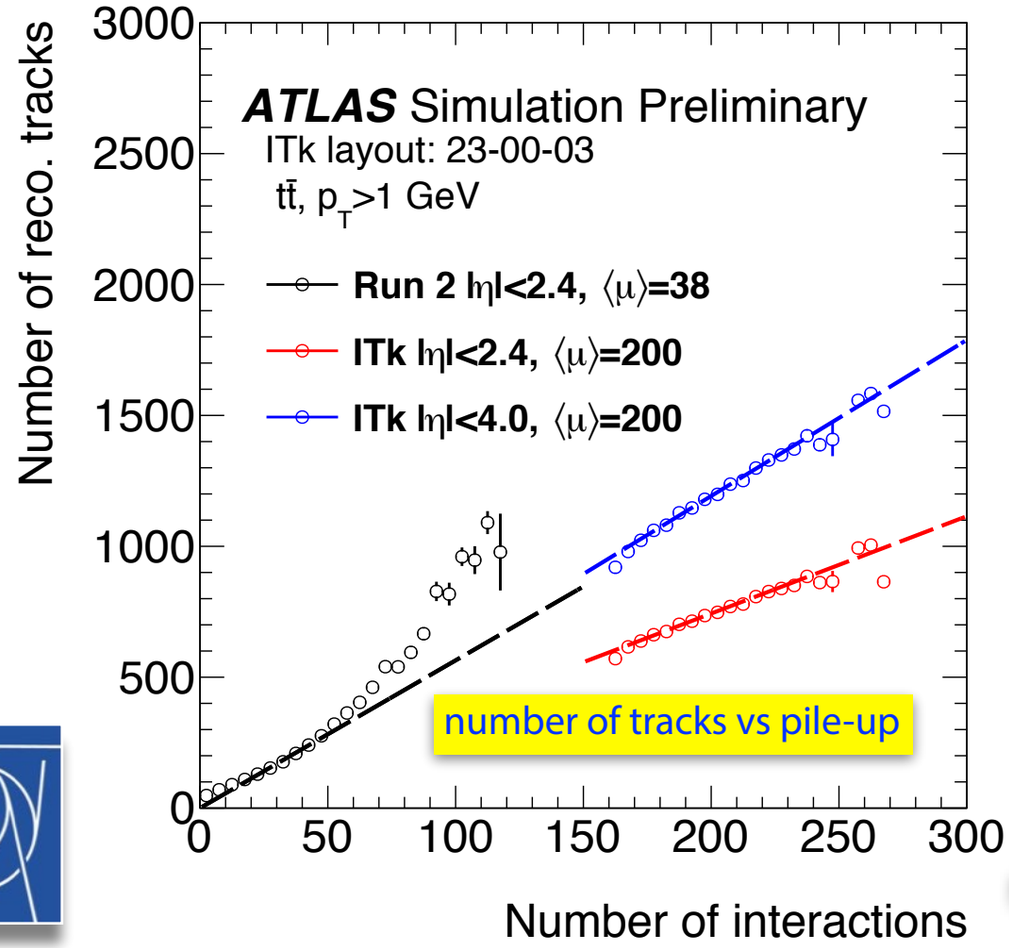


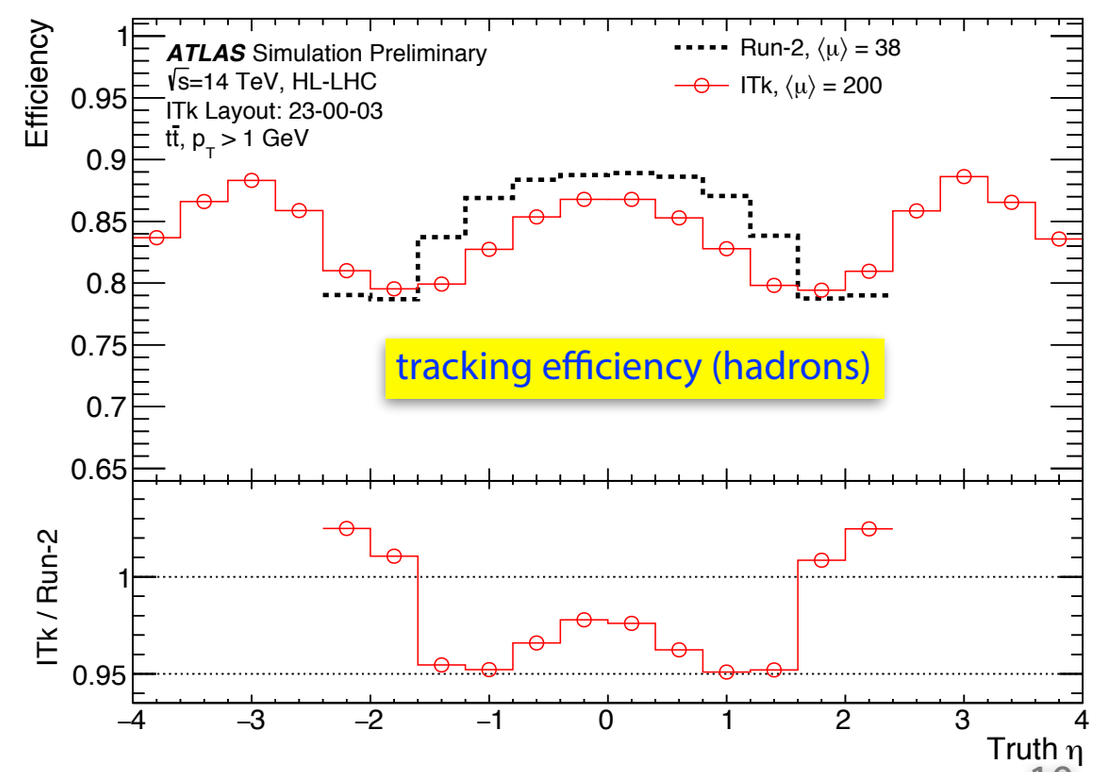
ATL-PHYS-PUB-2019-041

ITk Tracking Performance (Default Tracking)

- high-purity track selection
  - ➡ raise N(hit) cut from 7 (Run-2) to 9 to further reduce fakes
- excellent tracking efficiency and fakes
  - ➡ pion efficiency limited by hadronic interactions in detector ITk material
  - ➡ excellent linearity in number of tracks vs pile-up, up to the highest pile-up

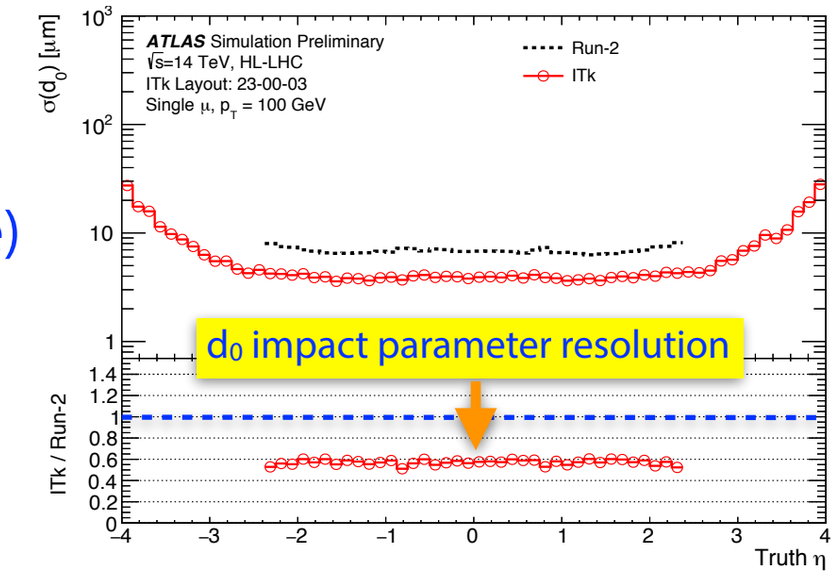




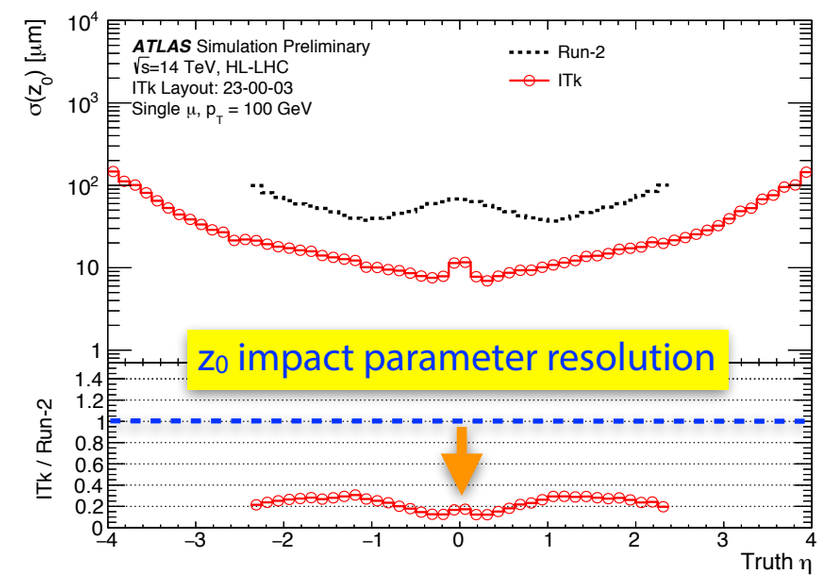


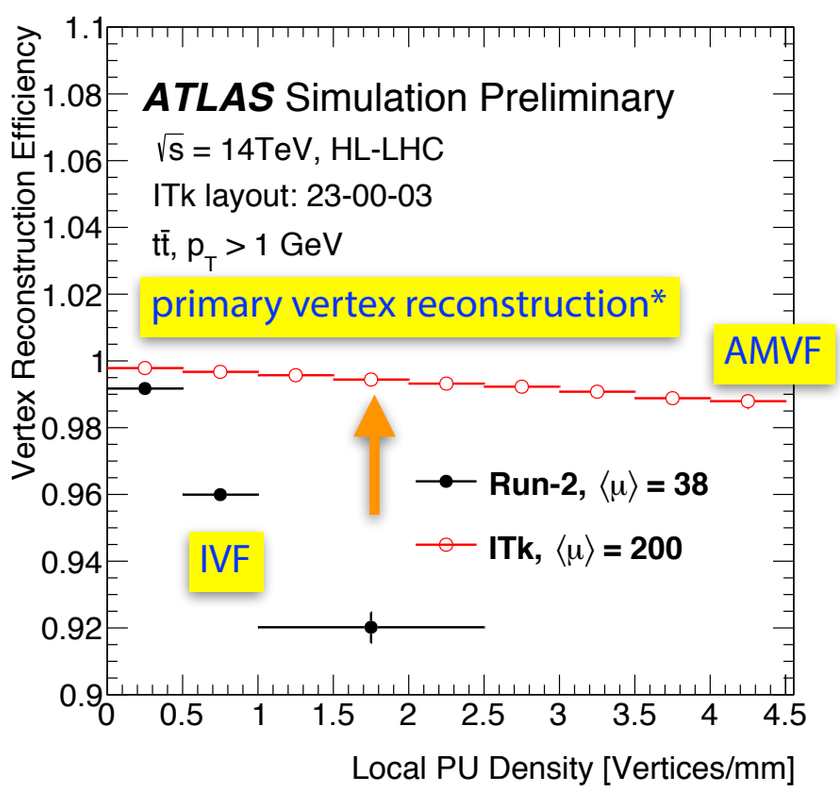
ITk Tracking Performance (Default Tracking)

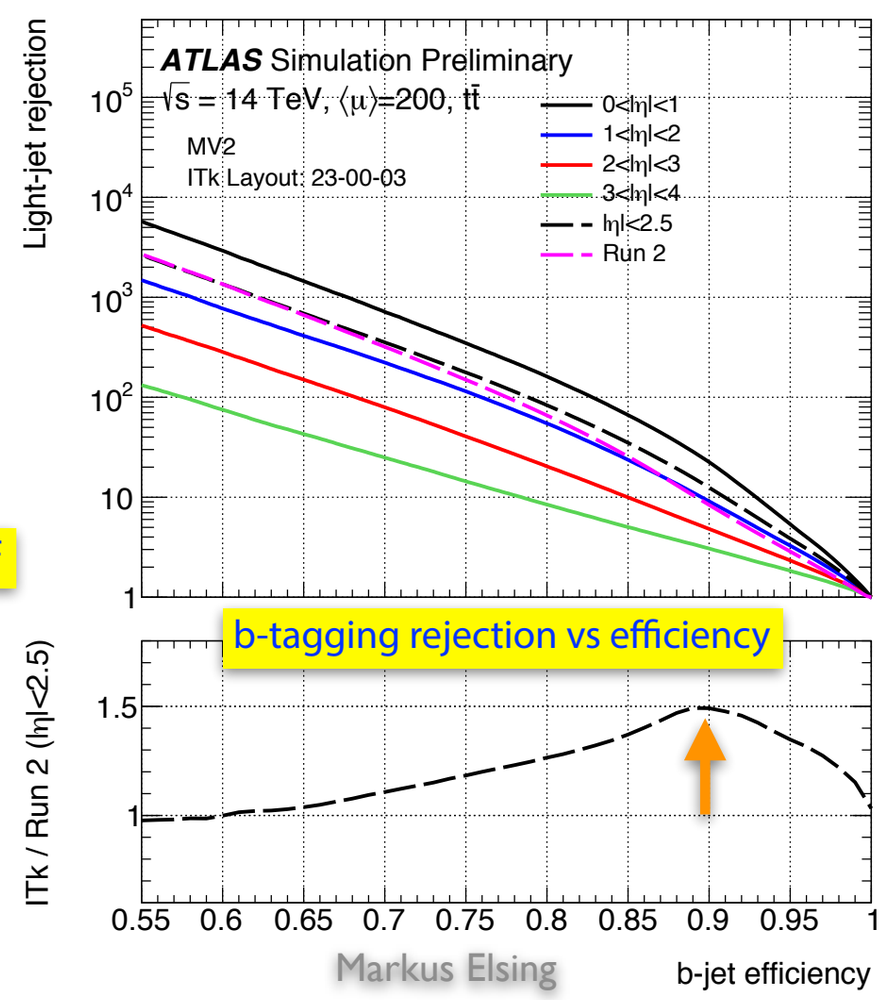
- much better impact parameter resolution
  - ➡ in particular at high-p_T (less affected by multiple scattering)
  - ➡ Pixel pitch: 25x100 μm² (layer-0 barrel), 50x50 μm² (elsewhere)



- results in improved performance for physics
  - ➡ primary vertex reconstruction
  - ➡ b-tagging
  - ➡ pile-up jet rejection
  - ➡ ...



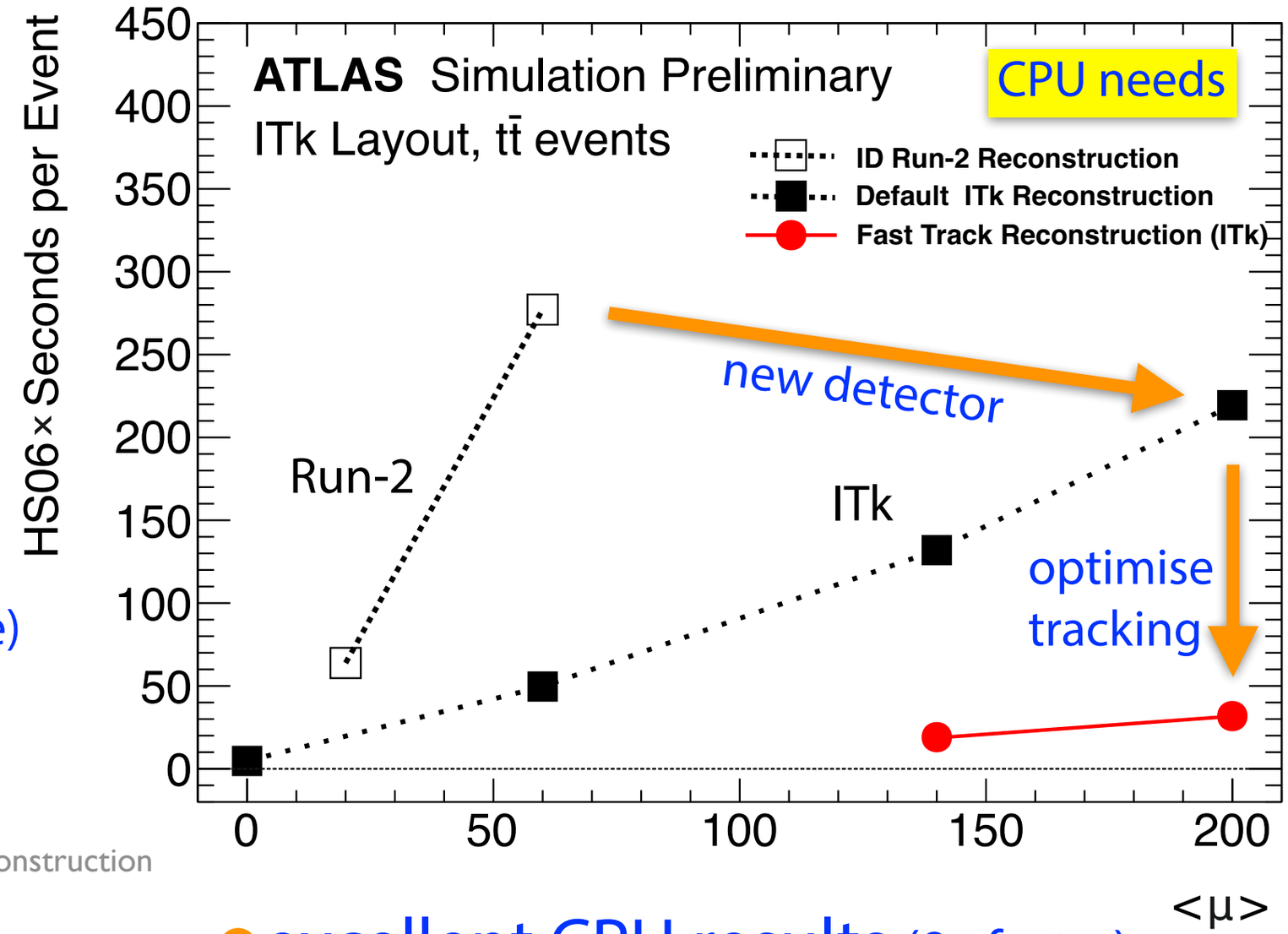


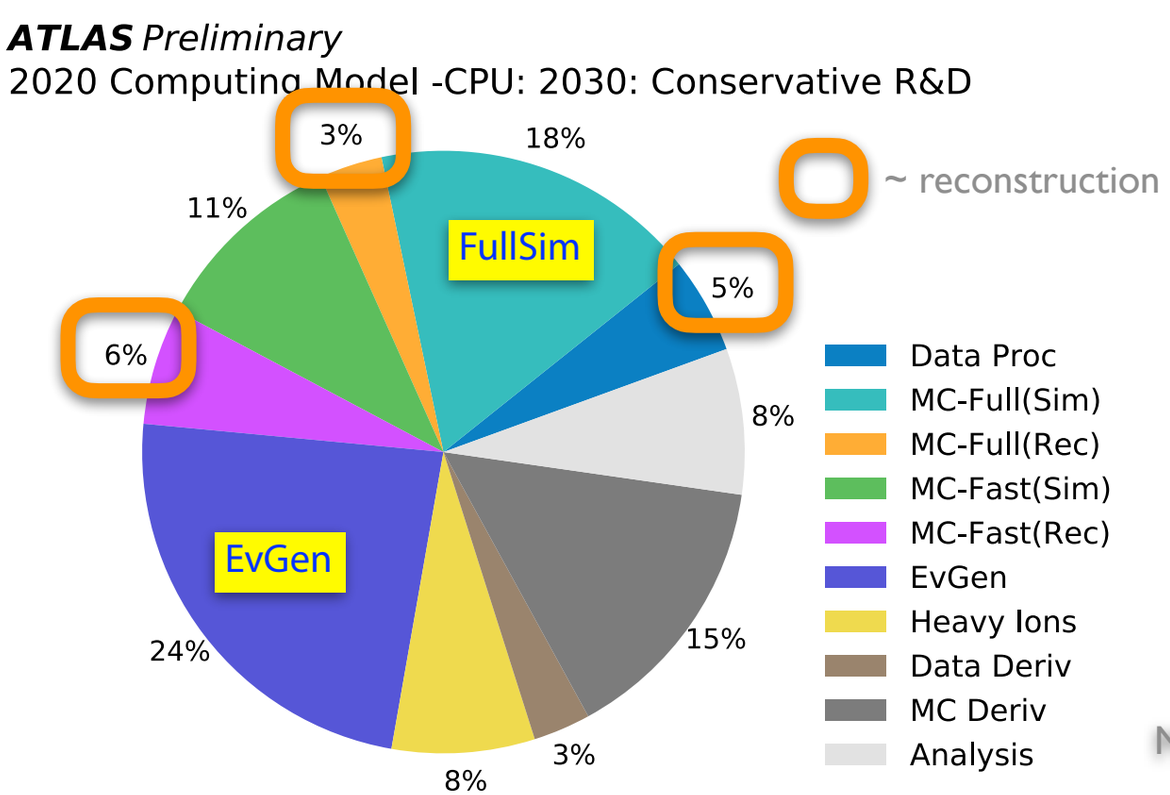


* Run-2 reference is still IVF, ITk uses AMVF with an alternative vertex seed finder

Fast ITk Track Reconstruction Prototype

- fast ITK tracking strategy
  - ➡ drop Strip seeding iteration, rely on 5 layer Pixel seeding (!)
  - ➡ drop Ambiguity Solution, move functionality to Track Finding:
    - use Kalman Filter as final fit
    - precise cluster corrections
    - material model (approximate)
    - final track selection
    - duplicate removal (approximate)

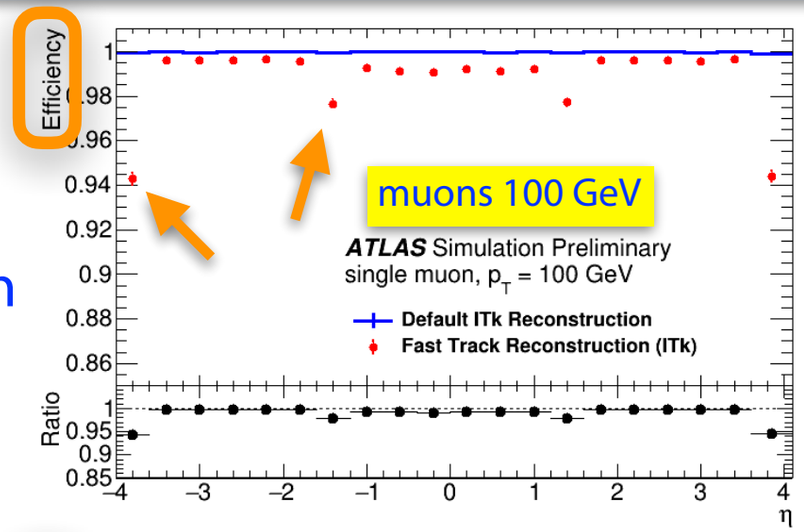




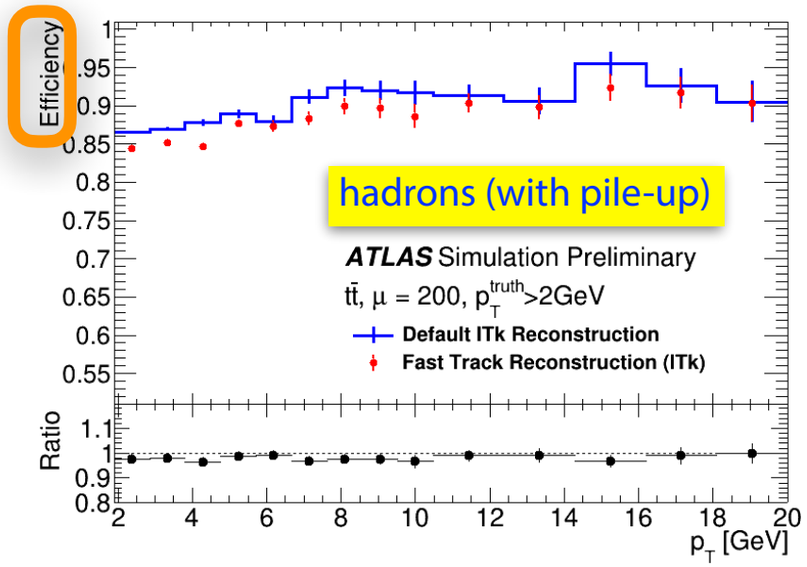
- excellent CPU results (8x faster)
  - ➡ at cost of some performance
  - ➡ CPU for tracking not dominating in event reconstruction anymore
  - ➡ QCD generators and Geant4 dominate in total ATLAS CPU budget for HL-LHC (!)

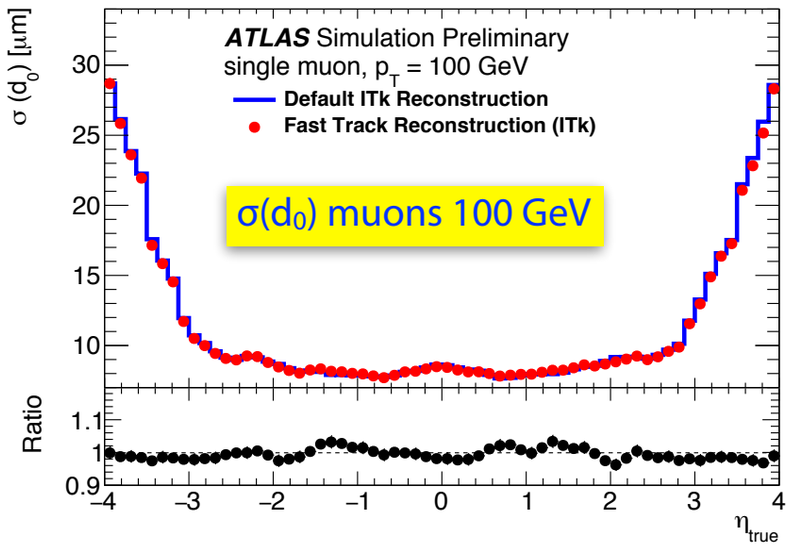
Performance of Fast ITk Reconstruction

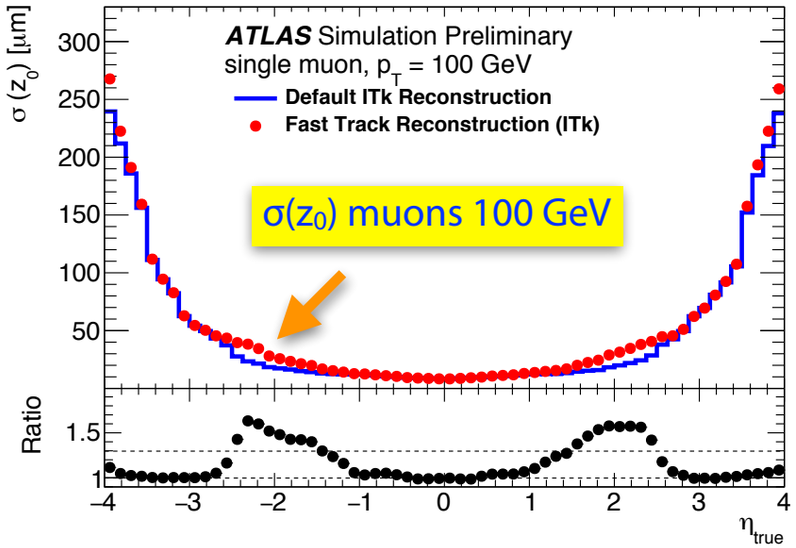
- prototype yields excellent results, despite approximations to achieve CPU savings
  - ➡ some efficiency loss in barrel, transition and forward region
  - ➡ no large rate of additional (fake) tracks
  - ➡ resolutions mostly as default reconstruction, with some exceptions due to approximations in Kalman Filter

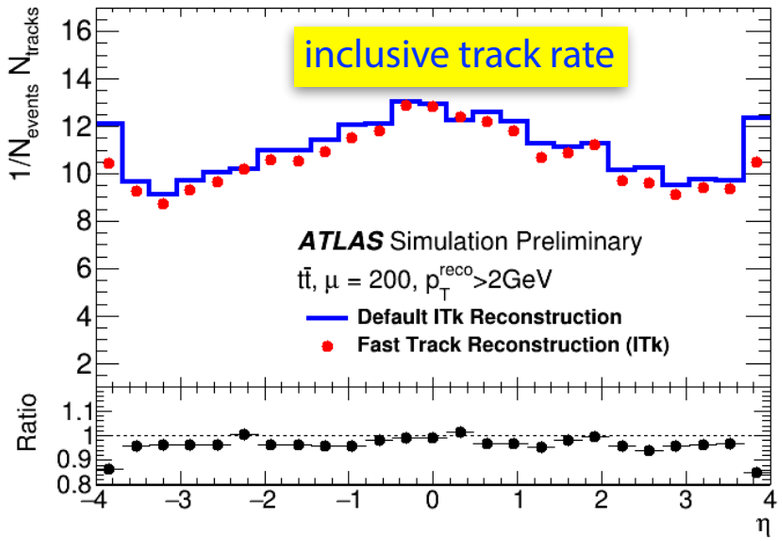


- performance of fast ITk reconstruction already satisfies trigger requirements
  - ➡ relevant for Event Filter reconstruction, possible alternative to hardware based tracking at Point-1

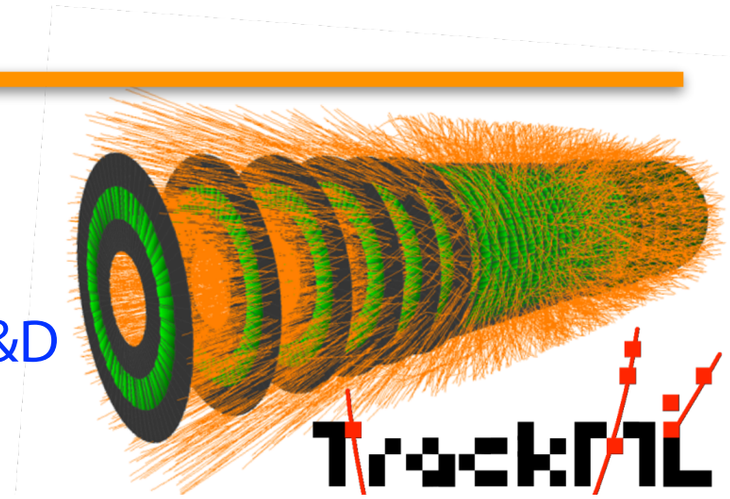




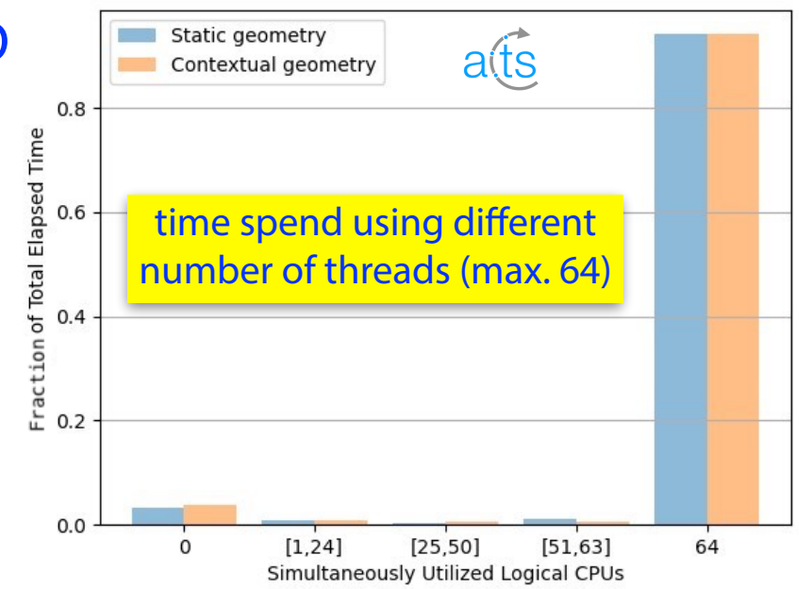




ACTS Tracking Software Project

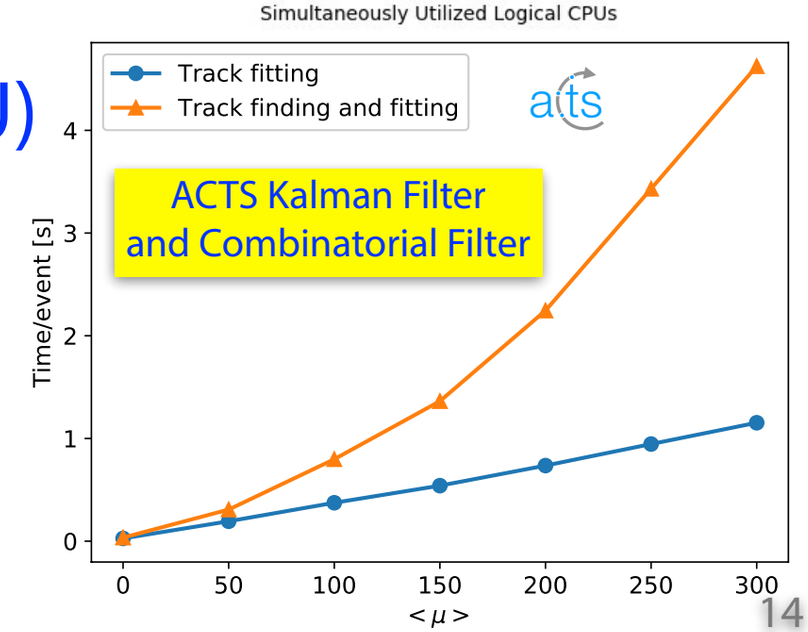


- ACTS (A Common Tracking Software)
  - ➡ open source project, initially started by ATLAS in view of HL-LHC
  - ➡ number of experiments contributing and used as platform for R&D
    - sPHENIX, Belle-II, FASER, LHCb, CEPC, FCC, ...
    - Tracking Machine Learning (TrackingML) challenge
    - Open Data Detector (ODD) allows for generic tracking R&D



- tracking suite for ATLAS Phase-2 software
  - ➡ designed ground-up for multi-threading
  - ➡ modern data model and code design
  - ➡ support for heterogeneous architectures (CPU, GPU, ...)
  - ➡ Run-3 primary vertexing first deployment of ACTS code

- much improved technical performance (CPU)
  - ➡ use fast ACTS (Combinatorial) Kalman Filter to implement a fast ITk reconstruction without approximations
  - ➡ goal is to fully recover physics performance without losing excellent CPU results of current prototype



Summary and Conclusions

- tracking and vertexing software is well prepared for Run-3
  - ➡ based on Run-2 experience, significant improvements have been implemented
    - technical migration to multi-threading
    - reconstruction time reduced by more than a factor ×2
    - large radius tracking will be integrated into prompt reconstruction at Tier-0
    - new Adaptive Multi Vertex Finder deployed (ACTS based)

- tracking software preparation for HL-LHC is advancing well
  - ➡ the ITk upgrade, together with optimised tracking software, will allow to improve on Run-2(3) tracking performance, even in presence of 200 pile-up
  - ➡ fully functional fast ITk reconstruction cuts down CPU time for tracking by factor ×8, a game-changer for ATLAS Phase-2 offline computing and for trigger processing
  - ➡ ATLAS is investing strongly into the ACTS open source tracking software project to modernise its tracking software for the HL-LHC