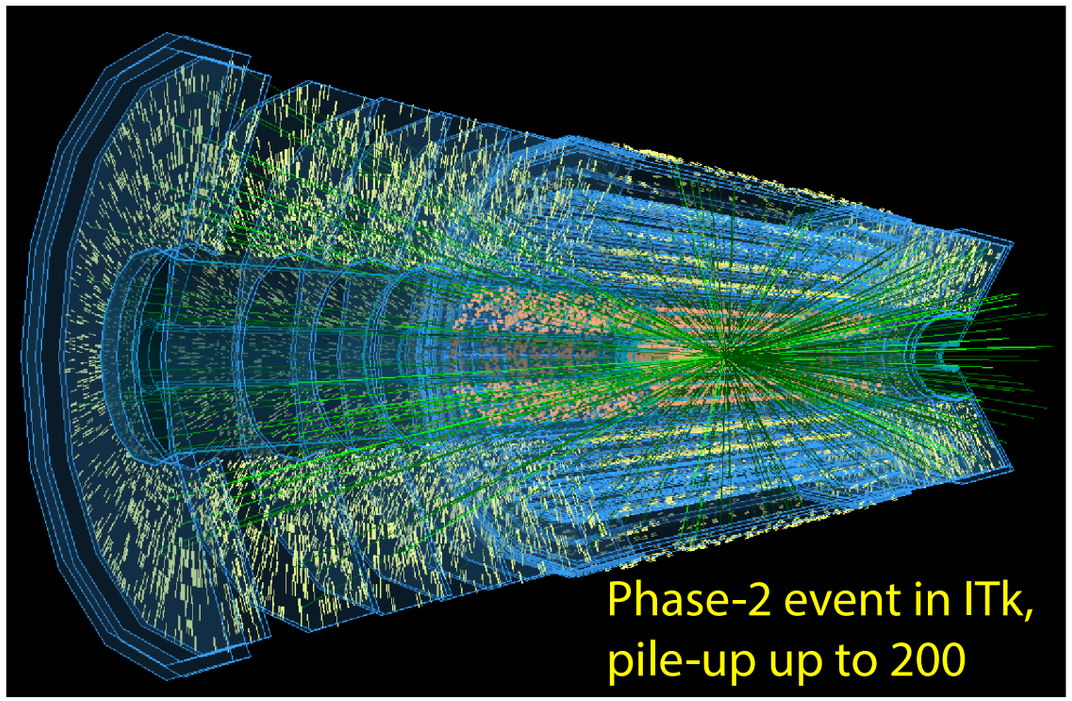
Fast Track Reconstruction for HL-LHC in ATLAS

by Markus Elsing

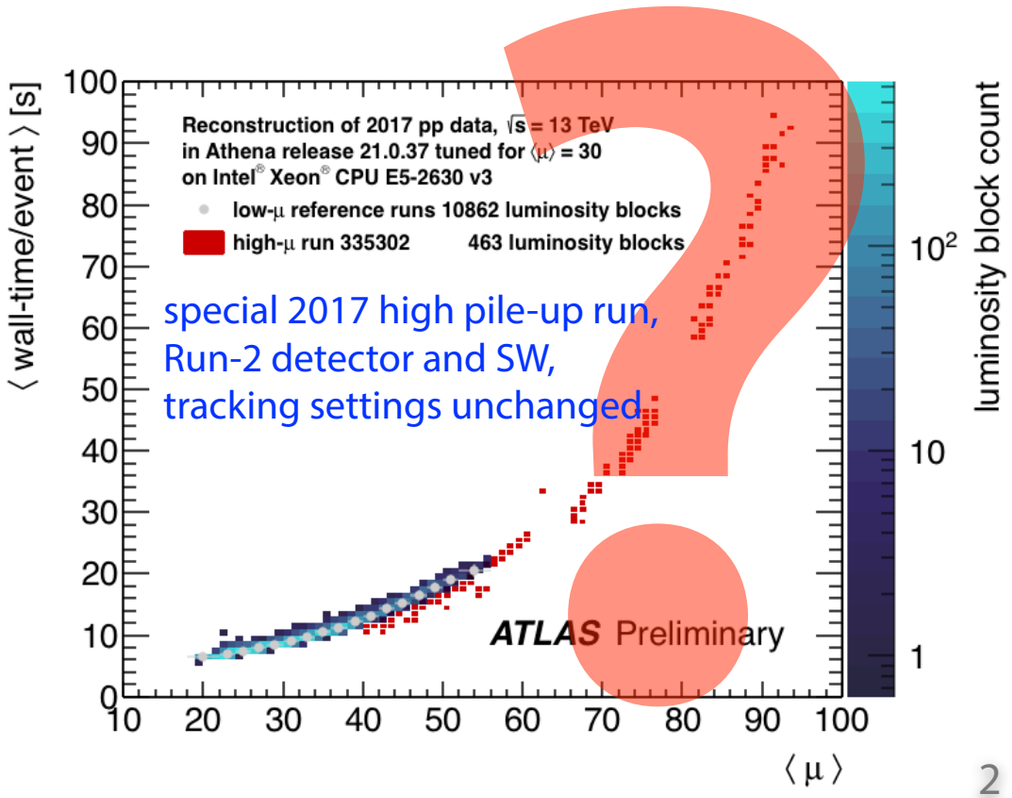
- presentation at HSF reconstruction group meeting

Introduction

- Phase-2 levels of pile-up
  - ➡ a challenge all aspects of the experiment
    - in particular for track reconstruction
  - ➡ famous CPU plot using special Run-2 run with very high pile-up
    - Run-2 detector not made for this !
  - ➡ tracking settings unchanged
    - not adequate for such high pile-up levels
  - ➡ still raised a lot of concerns and interest in R&D on tracking algorithms !



- outline of the following:
  - ➡ brief introduction to the ITK upgrade
  - ➡ performance of default ITk reconstruction and current computing model
  - ➡ R&D for fast track reconstruction
  - ➡ results of fast ITK track reconstruction study
  - ➡ computing model implications and outlook



Run-2 NewTracking (for completeness)

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

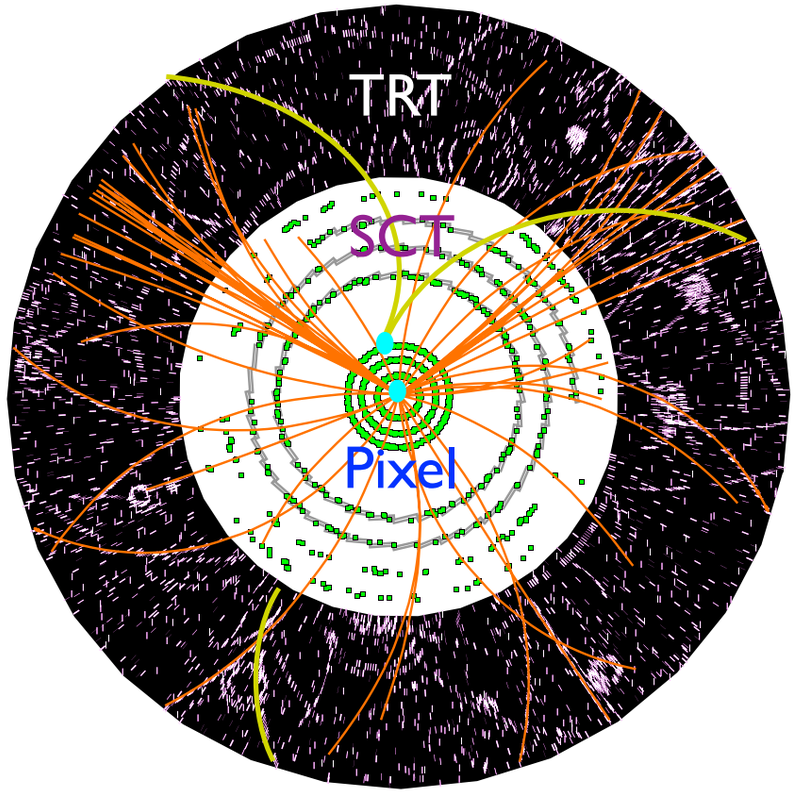
TRT segment finder
- ➡ in EM Regions-of-Interest
- ➡ on remaining drift circles
- ➡ uses Hough transform

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

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

standalone TRT
- ➡ unused TRT segments

vertexing
- ➡ primary vertexing
- ➡ conversion and V0 search

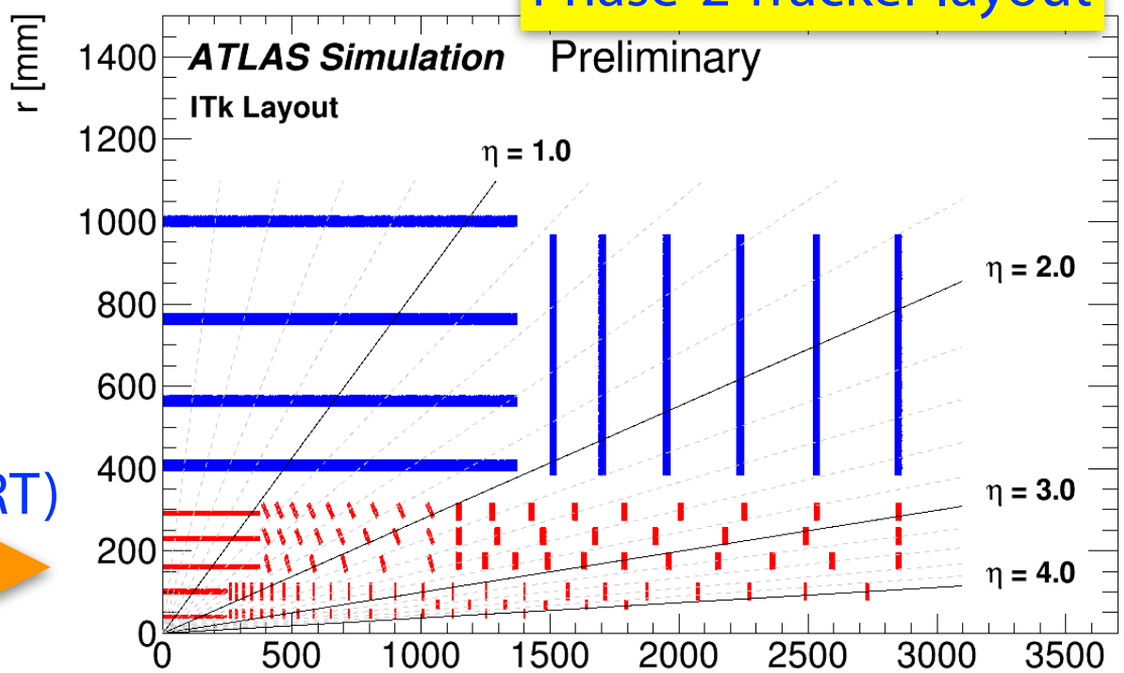


ATLAS Phase-2 Tracker Upgrade ITk

- Run-2 detector designed for $\langle\mu\rangle \simeq 20$
  - ➡ will operate at bandwidth limit already in Run-3
  - ➡ will reach end-of-lifetime
- ITk layout optimised for $\langle\mu\rangle \simeq 200$
  - ➡ higher granularity Pixel and Strip Detector (no TRT)
  - ➡ 5 layer Pixel Detector covering $|\eta| < 4$

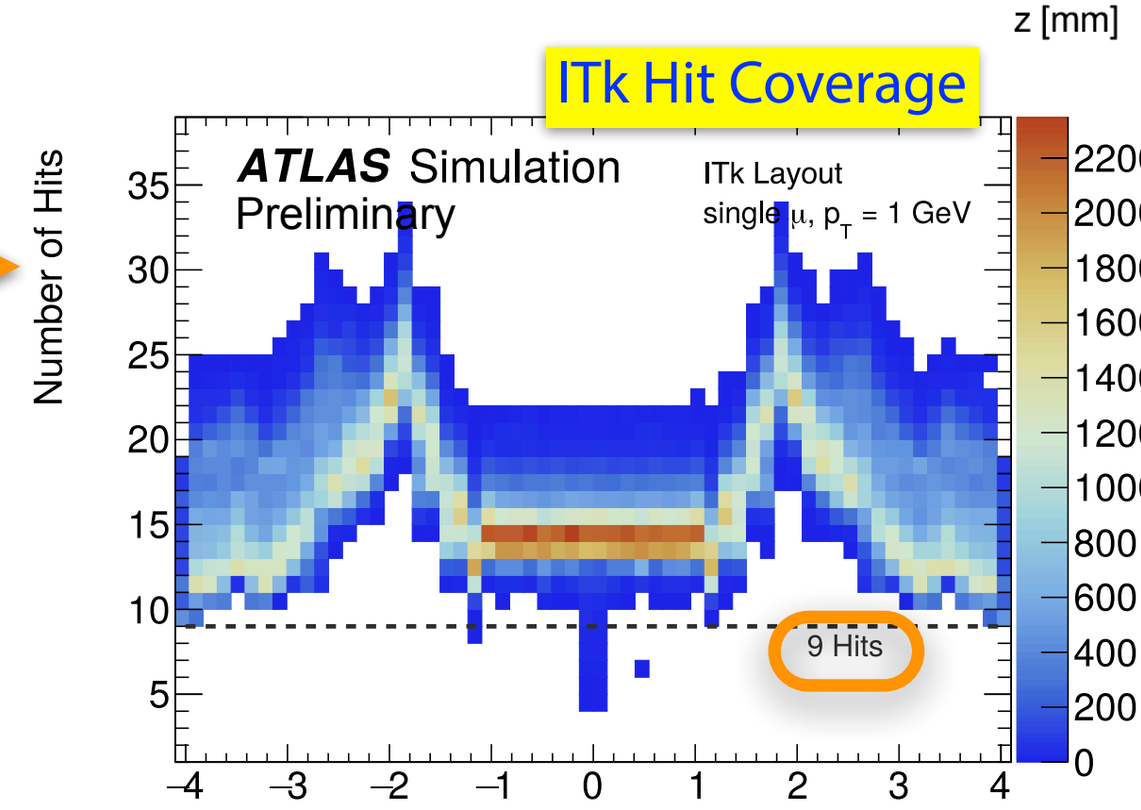
    [50x50 µm² pixels, L0 barrel 25x100 µm², r(L0) = 34 mm]

Phase-2 Tracker layout



- ring design of ITk Pixel Detector
  - ➡ optimised for:
    - tracking performance at all η
    - required hit coverage as function of η
    - minimal pixel surface to achieve this (cost)
    - and CPU performance of tracking (!)
  - ➡ multiple hits per layer in forward region

ITk Hit Coverage



ITk Track Reconstruction using Run-2 Sofware

- Run-2 NewTracking software adapted to ITk:
  - ➡ harder selection cuts compared to Run-2(3):

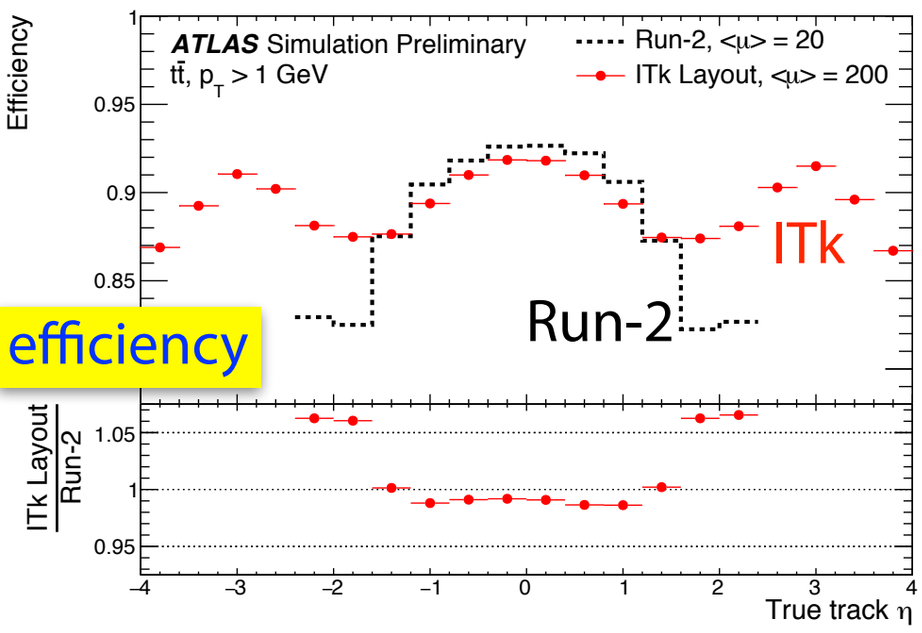
Requirement	Pseudorapidity interval		
	$\|\eta\| < 2.0$	$2.0 < \|\eta\| < 2.6$	$2.6 < \|\eta\| < 4.0$
Pixel+Strip hits	≥ 9	≥ 8	≥ 7
Pixel hits	≥ 1	≥ 1	≥ 1
Holes	≤ 2	≤ 2	≤ 2
p_T [MeV]	> 900	> 400	> 400
$\|d_0\|$ [mm]	≤ 2	≤ 2	≤ 10
$\|z_0\|$ [cm]	≤ 20	≤ 20	≤ 20

  - ➡ high purity working point, used for TDRs and Yellow report

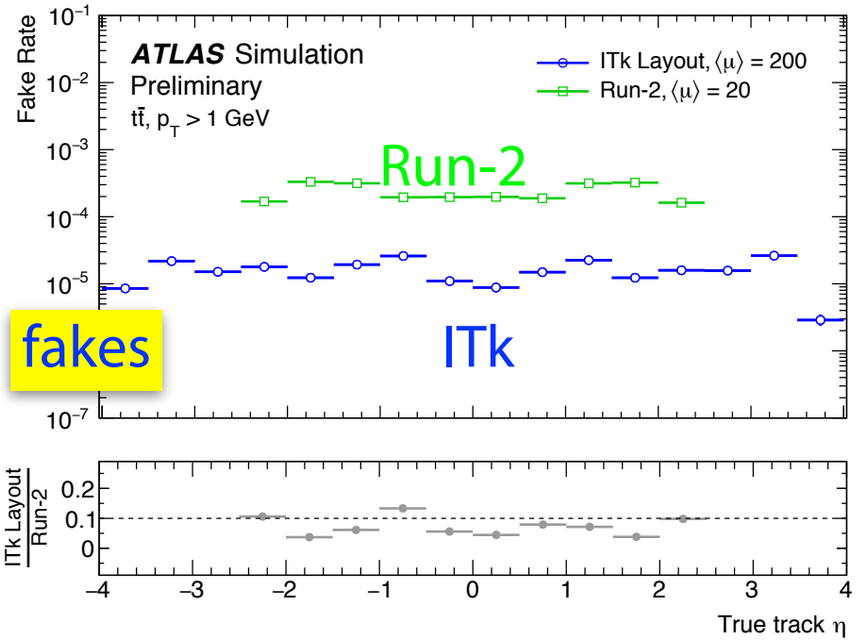
Pre-processing
- ➡ Pixel+SCT clustering
- ➡ Space Points formation

Silicon Track Finding
- ➡ iterative :
  1. Pixel seeds
  2. Strip seeds
- ➡ restricted to roads
- ➡ combinatorial Kalman Filter
- ➡ Brem.recovery in EM Regions-of-Interest

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

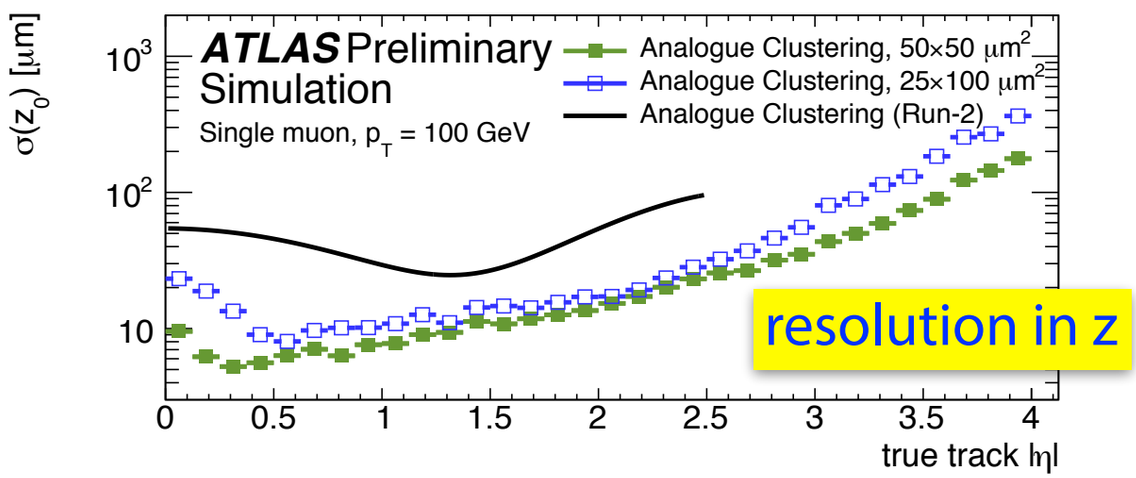


efficiency

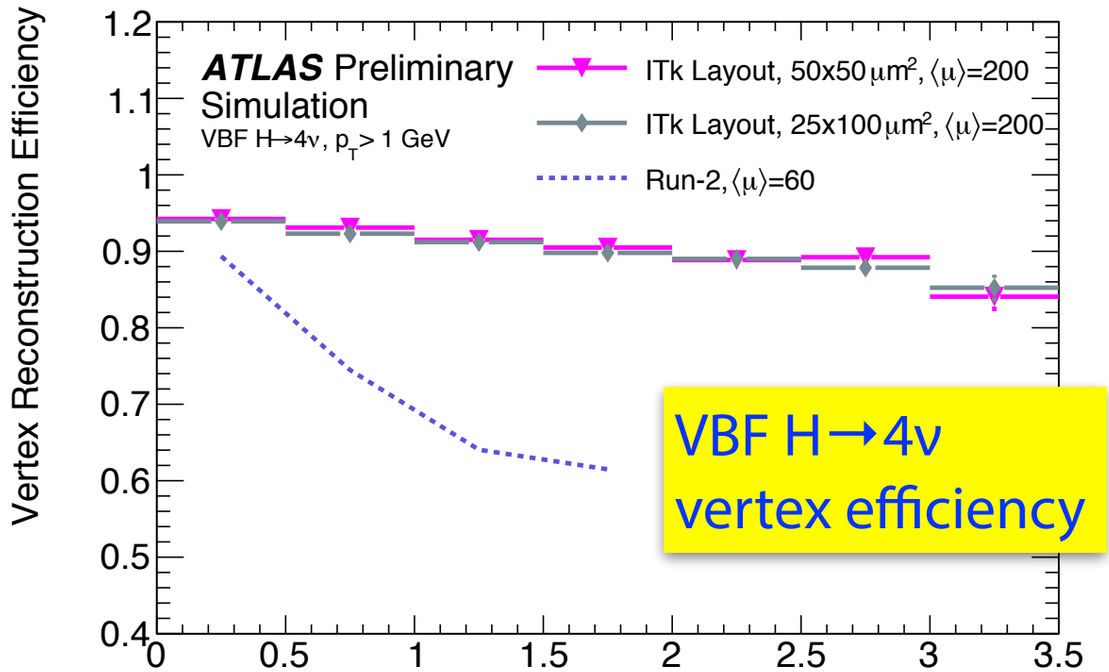


fakes

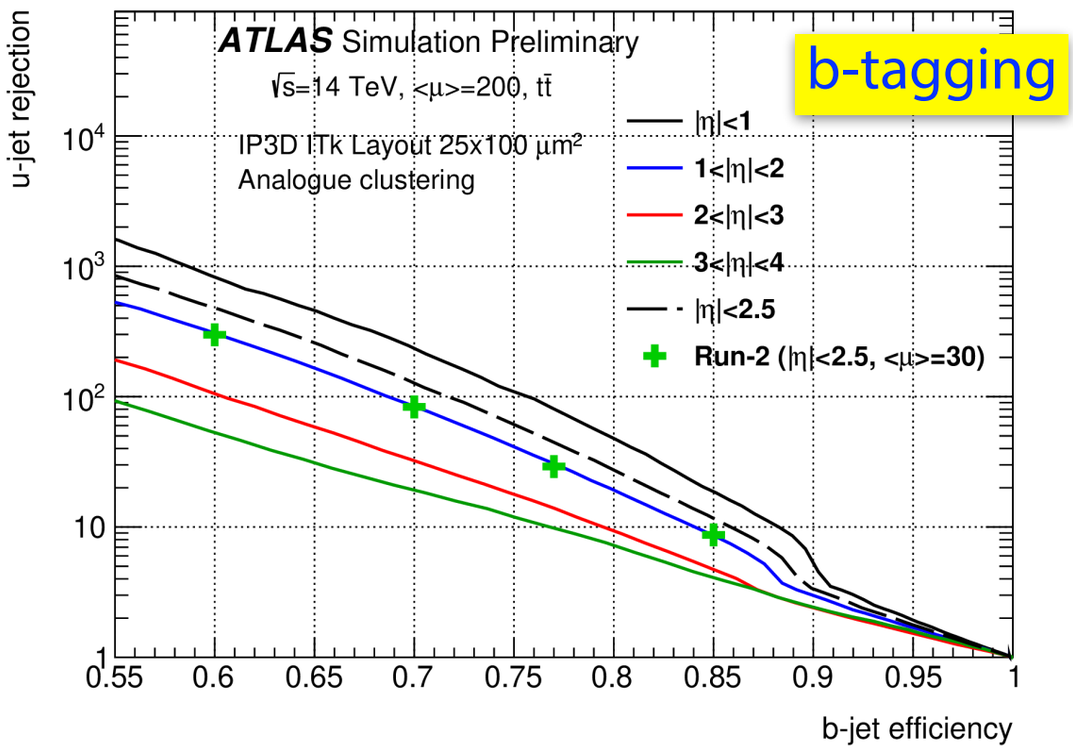
Examples for ITk Physics Performance



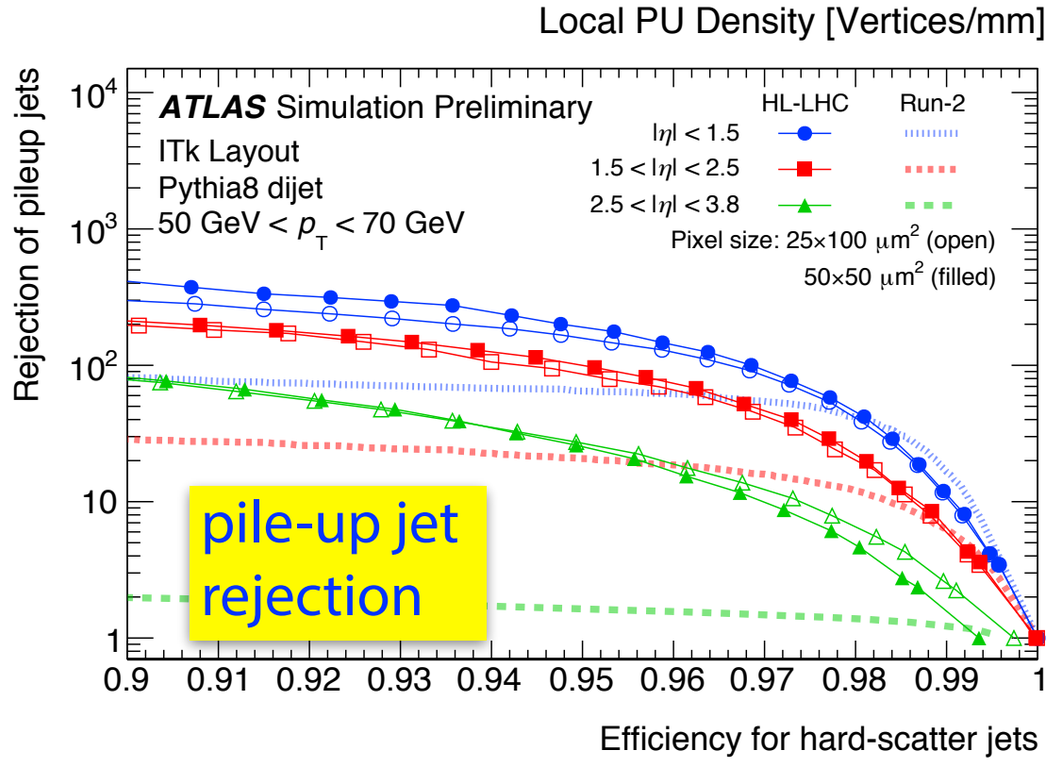
resolution in z



VBF H→4ν vertex efficiency



b-tagging

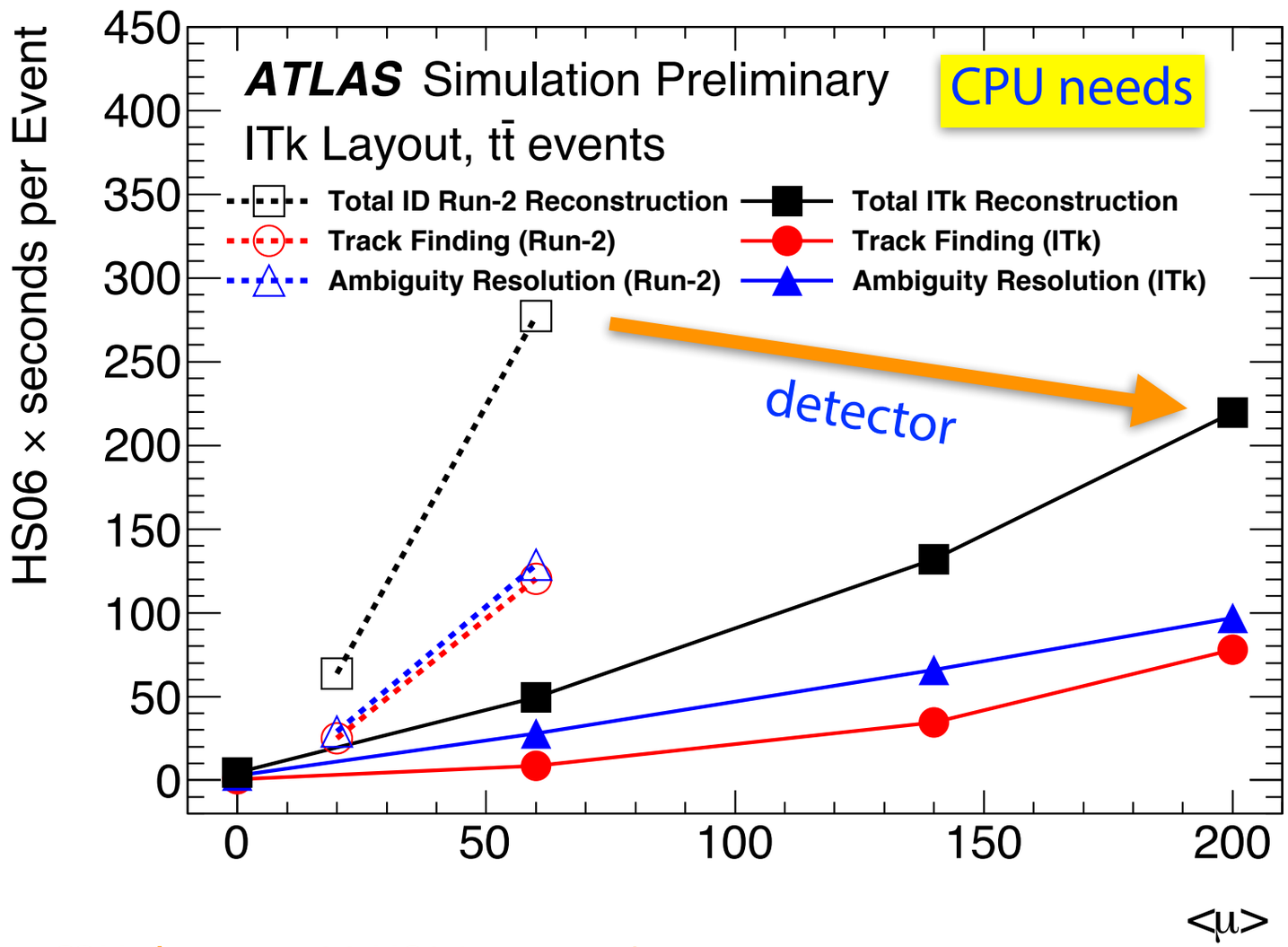


pile-up jet rejection

CPU needs for ITk Track Reconstruction

- large CPU reduction due to detector upgrade
  - ➡ same Run-2 tracking software base
  - ➡ biggest contributions in both cases:
    - Silicon Track Finding
    - Ambiguity Resolution

    (different scaling for both with ITk)
  - ➡ in addition, TRT reconstruction for Run-2 ID (not dominant)
  - ➡ tuning of tracking code as used for all TDRs and Yellow report
    - mainly for physics performance !

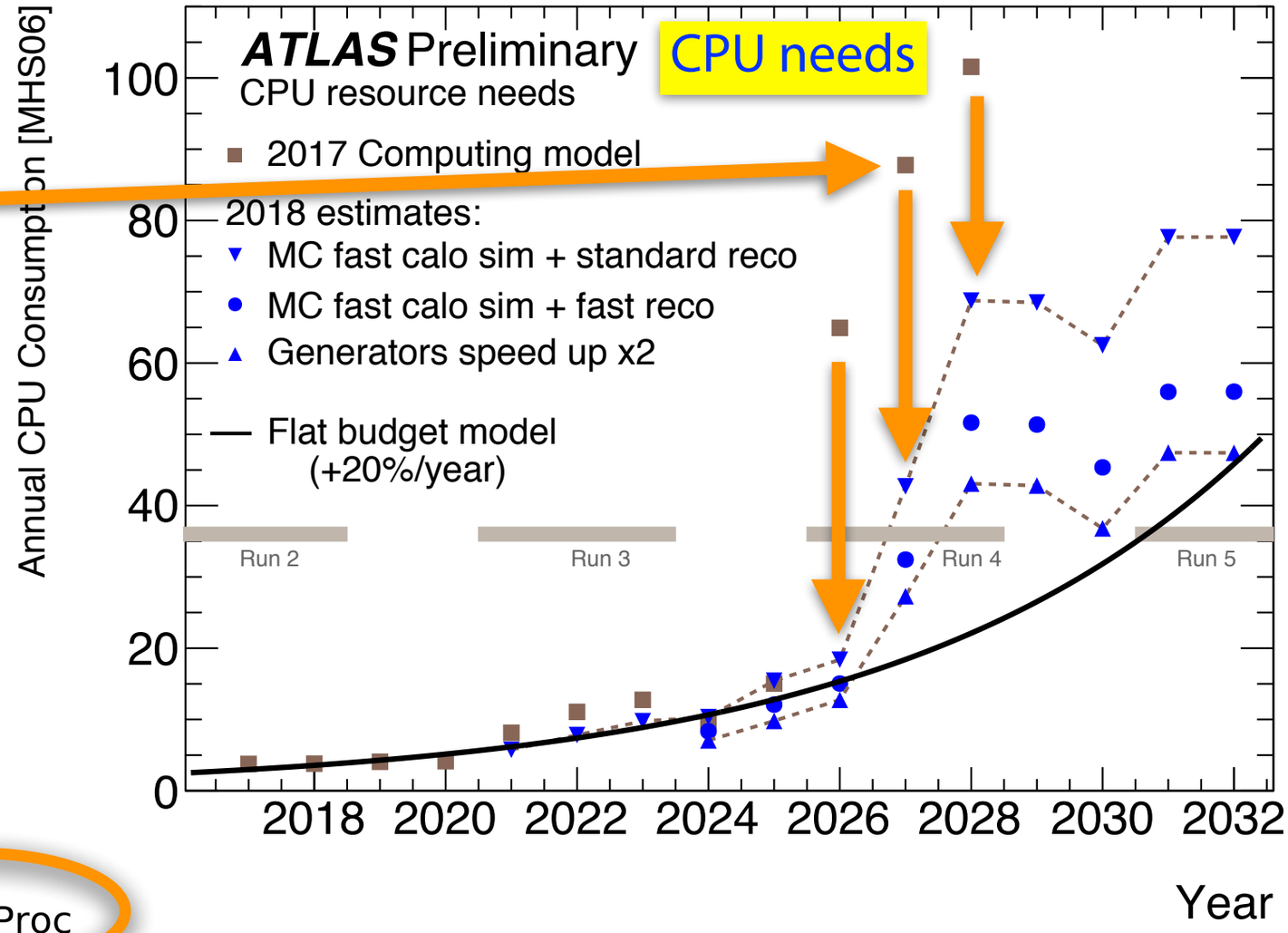


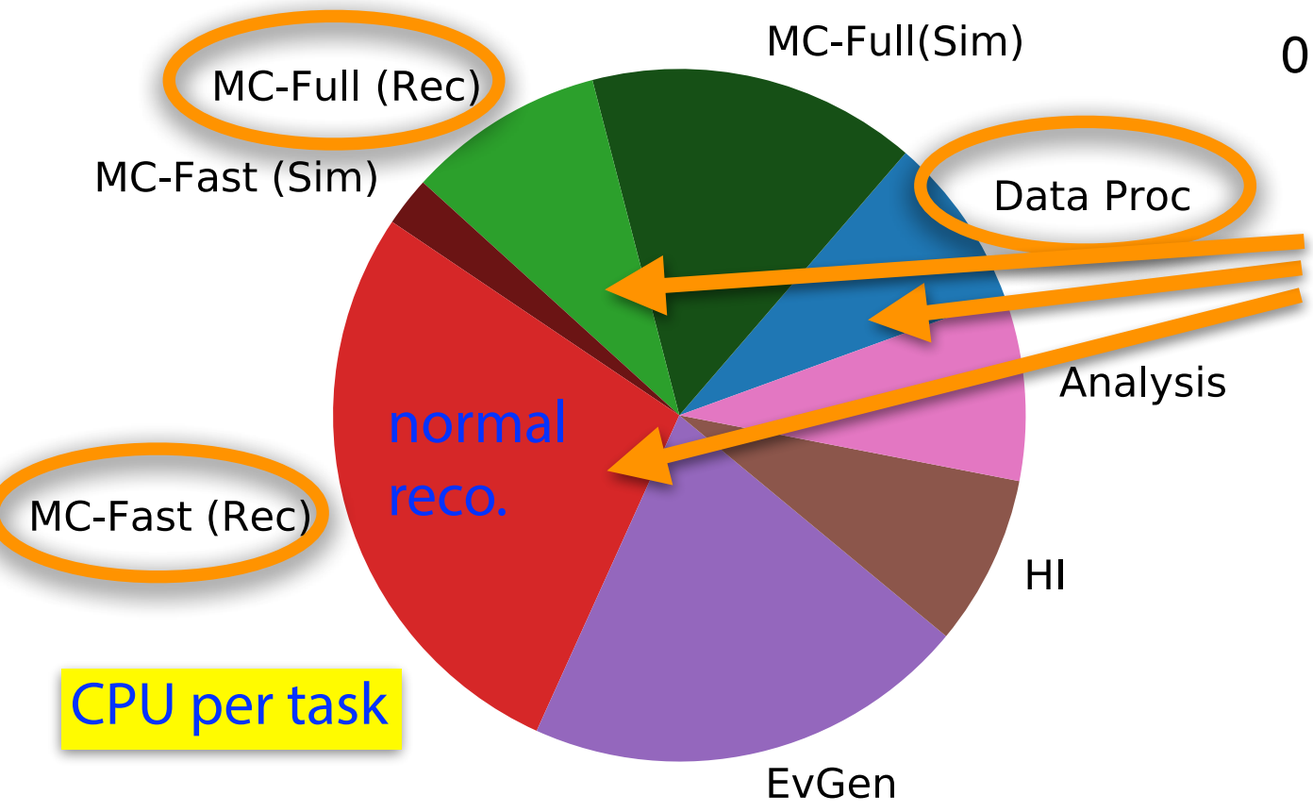
(numbers in HS06*seconds, todays CPUs have 15-20 HS06 per core)

Detector	Pile-up	Cluster Finding	Space Points	Si Track Finding	Ambiguity Resolution	TRT+Back Tracking
ITk	200	22	6.5*	78	97	-
Run-2	20	1.5	0.7	23	15	19

ITk Layout Design and Phase-2 Computing

- Phase-2 computing model
  - ➡ original Phase-2 CPU projections used extrapolations based on Run-2
  - ➡ 2018 update based on ITk and current reconstruction software
  - ➡ resulted in significant reduction !



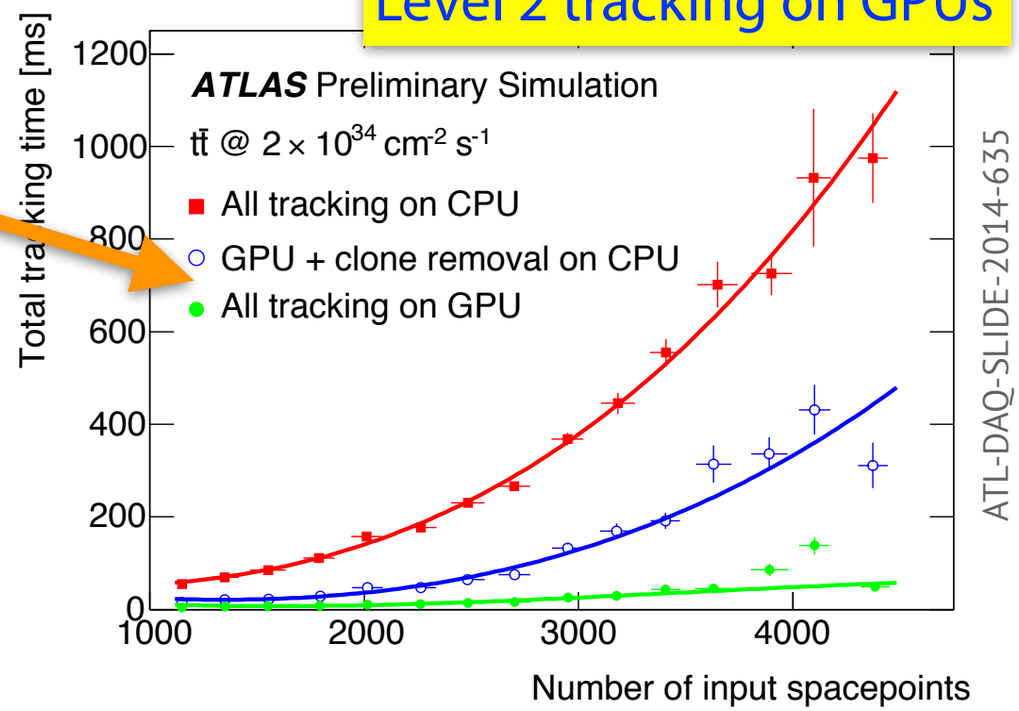


- reconstruction still ~ 45% of projected Phase-2 CPU needs
  - ➡ still above flat computing budget
  - ➡ optimistic projections use **truth based tracking** for fast simulation
  - ➡ can we do better ?

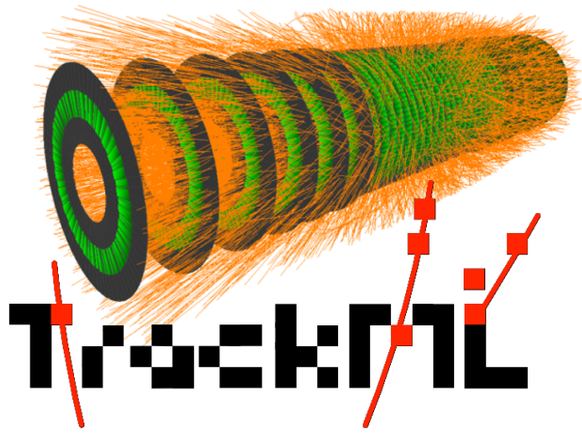
Towards Fast Offline Tracking for Phase-2

- Intensive R&D on tracking software
  - ➡ tracking community workshops (CTD)
  - ➡ ACTS as an open source tracking project, "community" project ATLAS/Belle-II/FCC ...
  - ➡ R&D on using GPUs and other co-processors
    - dedicated hardware for tracking (HTT)
    - R&D on GPU track finding for High Level Trigger
    - R&D on ACTS track seeding prototype on GPUs
  - ➡ TrackingML Challenge
    - reaching out to data science community
    - open data detector based on ACTS software
  - ➡ tracking R&D using unconventional approaches

Level 2 tracking on GPUs



- ATLAS also invests in optimising its classical tracking chain !
  - ➡ rest of this talk....



Modifying the ITk Track Reconstruction

- start from classical software chain
  - ➡ goal is to push CPU limits
  - ➡ some physics performance compromises tolerable for this study

- main steps of ITk track reconstruction
  - ➡ pre-processing
  - ➡ Silicon Tracking Finding
  - ➡ Ambiguity Resolution

Pre-processing
- ➡ Pixel+SCT clustering
- ➡ Space Points formation

Silicon Track Finding
- ➡ iterative :
  1. Pixel seeds
  2. Strip seeds
- ➡ restricted to roads
- ➡ combinatorial Kalman Filter
- ➡ Brem.recovery in EM Regions-of-Interest

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

Modifying the ITk Track Reconstruction Chain

- remove the Ambiguity Resolution step ?
  - ➡ would loose precise track fit, hence resolutions affected
  - ➡ no NN cluster splitting, so loss of performance for b-tagging

- rely on Silicon Track Finding instead:
  - ➡ use its Kalman Filter to estimate track parameters
    - remove (some) approximations in the material estimates
    - make use of full cluster calibrations in track finding
  - ➡ add some improved duplicate removal (shared hit treatment) and apply final selection cuts:

Requirement	Pseudorapidity interval		
	$\|\eta\| < 2.0$	$2.0 < \|\eta\| < 2.6$	$2.6 < \|\eta\| < 4.0$
Pixel+Strip hits	≥ 9 (7)	≥ 8 (7)	≥ 7 (7)
unique hits	≥ 7 (1)	≥ 6 (1)	≥ 5 (1)
shared hits	≤ 2 (no cut)	≤ 2 (no cut)	≤ 2 (no cut)
p_T [MeV]	> 1000 (900)	> 400 (400)	> 400 (400)
$\|z_0\|$ [cm]	≤ 15 (20)	≤ 15 (20)	≤ 15 (20)

(tighter selection cuts also speed-up track finding !)

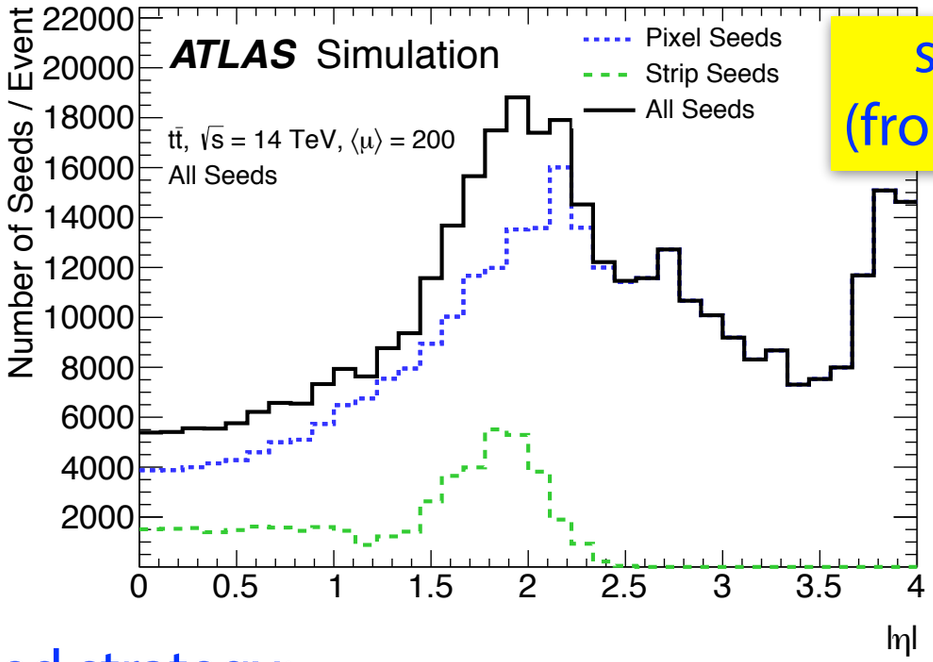
Pre-processing
- ➡ Pixel+SCT clustering
- ➡ Space Points formation

Silicon Track Finding
- ➡ iterative :
  1. Pixel seeds
  2. Strip seeds
- ➡ restricted to roads
- ➡ combinatorial Kalman Filter
- ➡ Brem.recovery in EM Regions-of-Interest

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

Modifying the ITk Track Reconstruction Chain

- simplify strategy for Track Finding:
  - ➡ at pile-up 200 about 50% of CPU time is spend on seeding



seeds vs η (from Pixel TDR)

- ➡ optimised strategy:
  drop strip seed iteration and improve purity of pixel seeds
- ➡ less seeds means also less CPU for combinatorial finder !

  (in addition, temporarily turned off the Brem.recovery)

Pre-processing
- ➡ Pixel+SCT clustering
- ➡ Space Points formation

Silicon Track Finding
- ➡ iterative :
  1. Pixel seeds
  2. ~~Strip seeds~~
- ➡ restricted to roads
- ➡ combinatorial Kalman Filter
- ➡ ~~Brem recovery in EM Regions-of-Interest~~

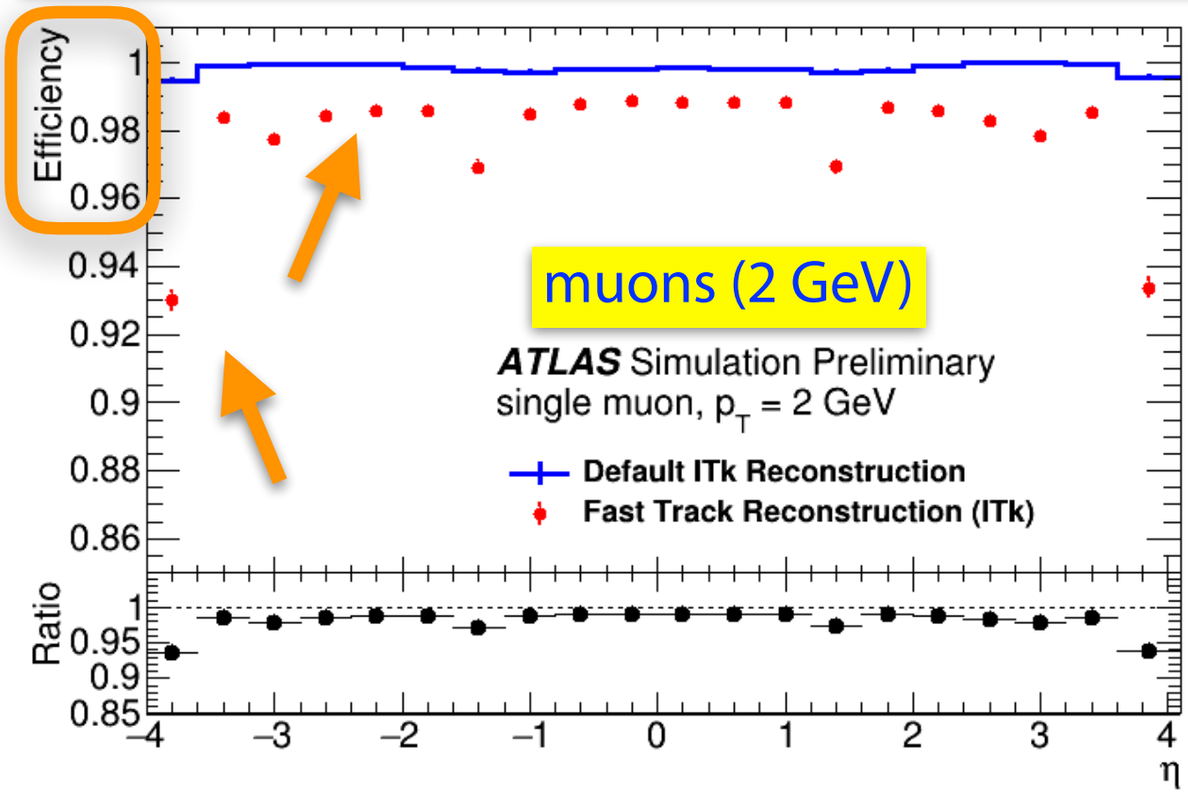
~~Ambiguity Resolution~~
- ~~➡ runs hole search~~
- ~~➡ scores tracks according to quality~~
- ~~➡ NN cluster splitting in jets~~
- ~~➡ precise least square track fit with Brem.recovery~~
- ~~➡ final selection cuts~~

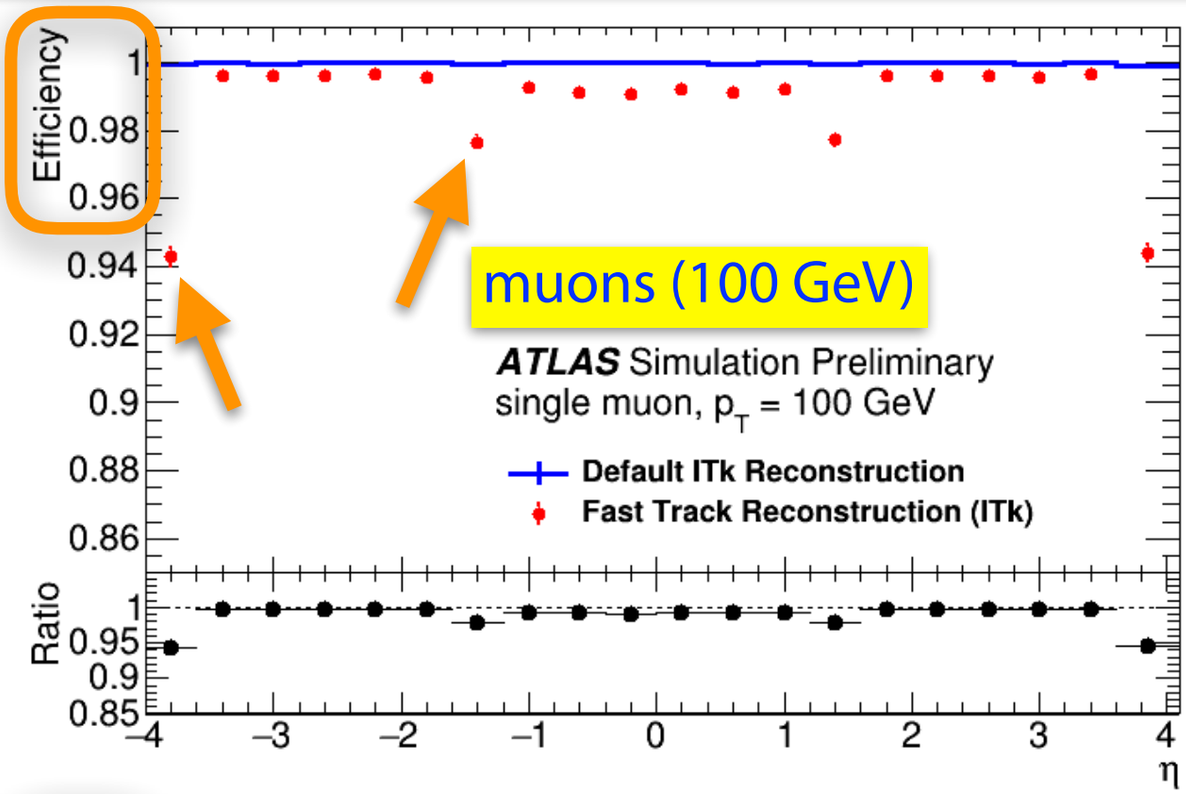
Modifying the ITk Track Reconstruction Chain

- improvements to Pre-processing
  - ➡ technical improvements to speed-up code execution
    - Pixel Space Point formation and Strip clustering code optimisation
    - Pixel clustering needed a bigger rewrite to speed it up
  - ➡ Strip Space Point formation only needed for seed finding
    - can be dropped, because no seeding using Strips anymore

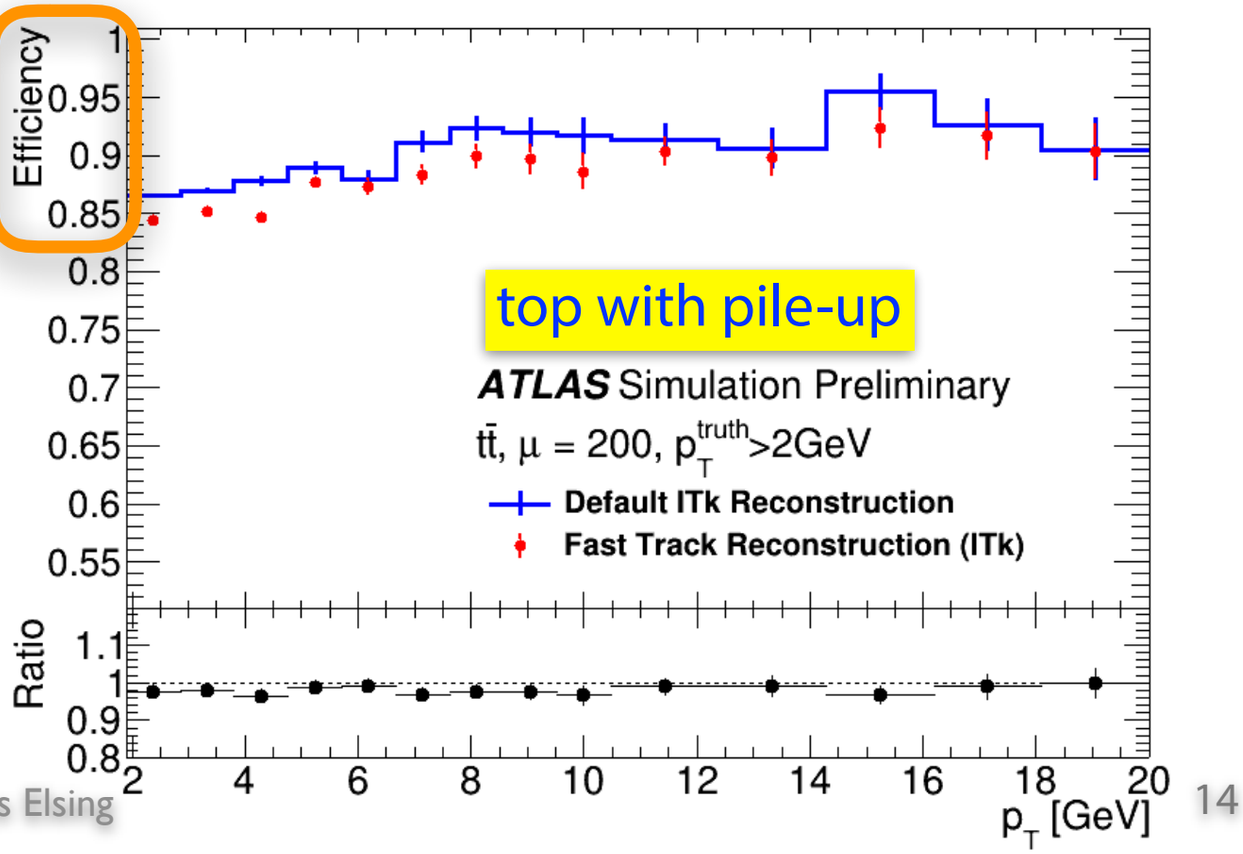
- raw data byte stream decoding
  - ➡ no decoding software for new Pixel frontend chip (RD53B)
    - new chip significantly compresses pixel information
    - event size for 200 pile-up: 580 kB for Pixels, 470 kB for Strips
  - ➡ use Run-2 raw data decoding times scaled to ITk event sizes
    - for MC, this is replaced by ROOT file decoding

Performance of Fast Track Reconstruction



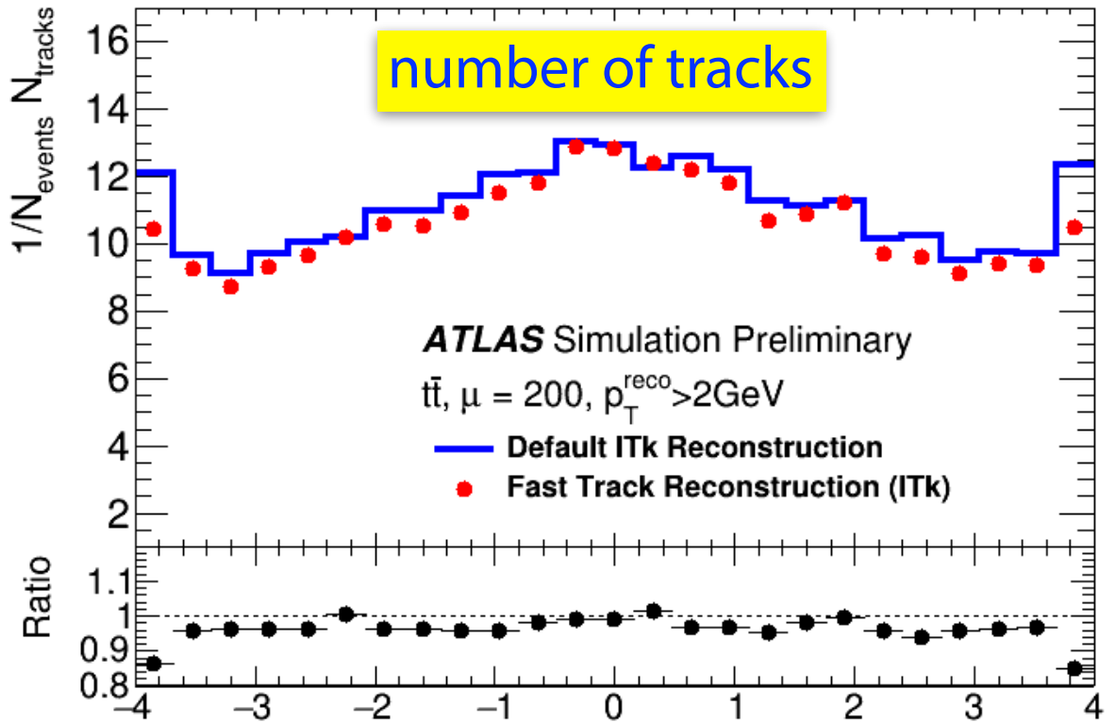


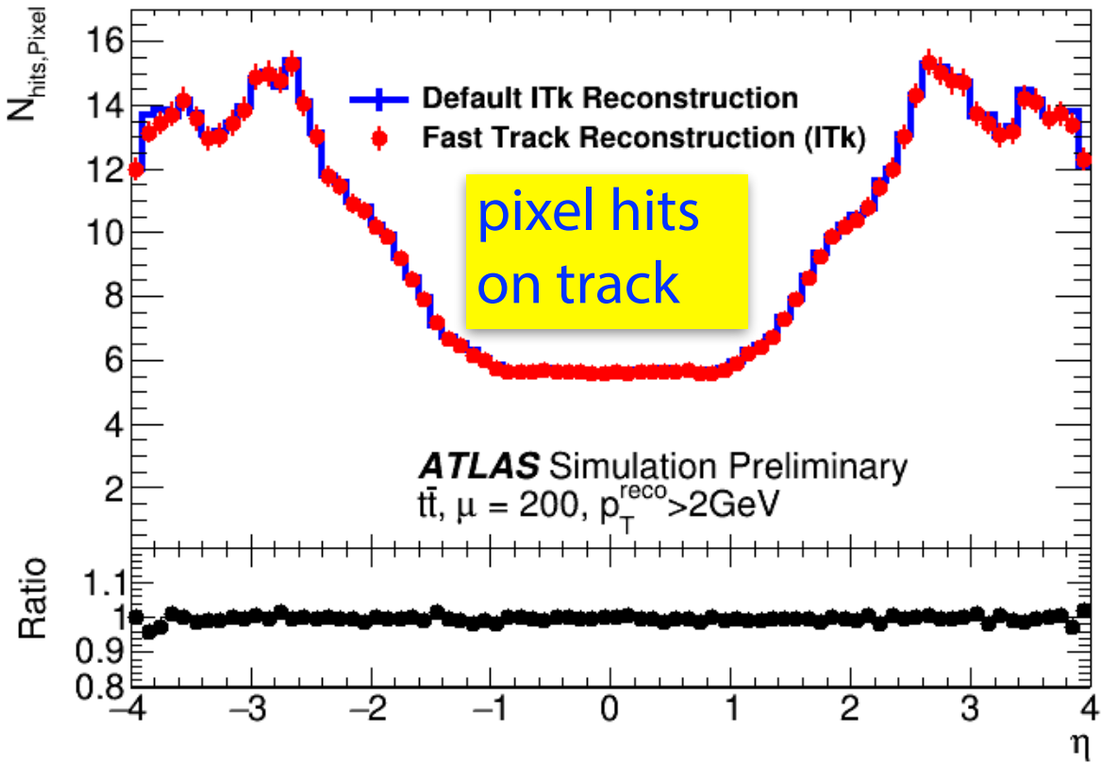
- small efficiency losses
  - ➡ strip barrel/end-cap gap, $|\eta| \sim 4$, low-p_T
  - ➡ stable as function of p_T and in hadronic events, dominated by material interactions
  - ➡ encouraging, but still some work to do !

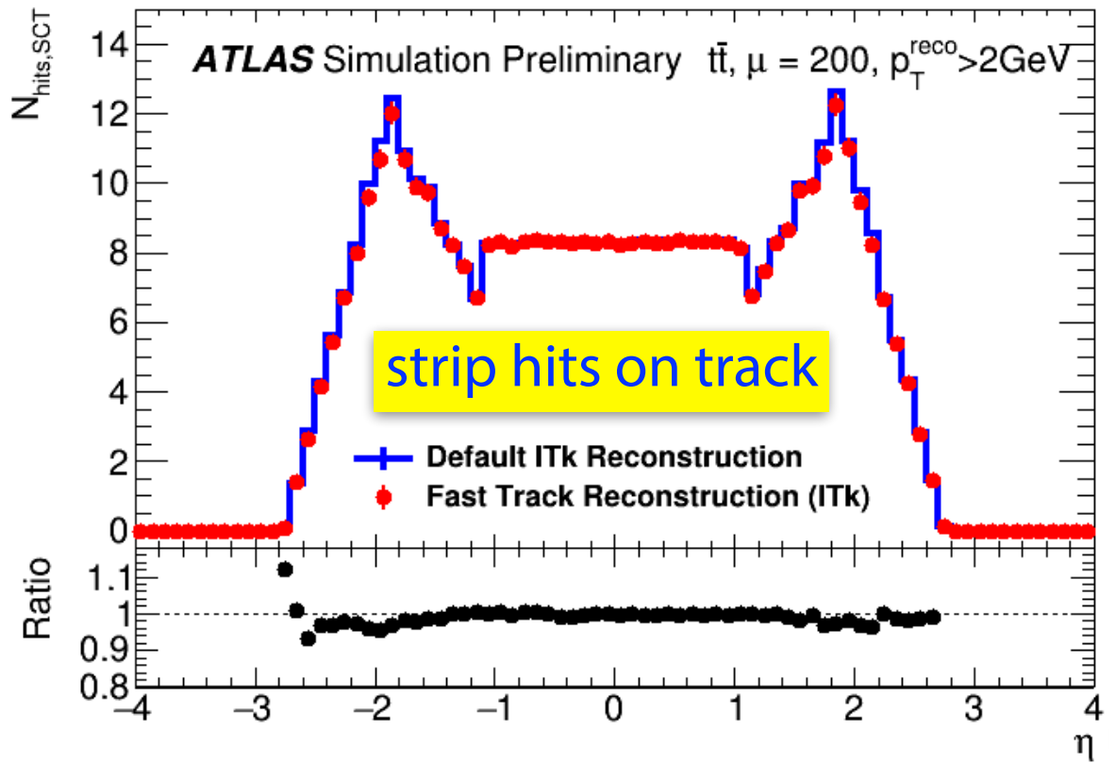


Performance of Fast Track Reconstruction

- slightly less tracks found
  - ➡ reflects lower tracking efficiency
  - ➡ no significant additional rate of fakes
- hit association almost identical
  - ➡ **full detector** used for regional and global tracking ! (all η , all layers)

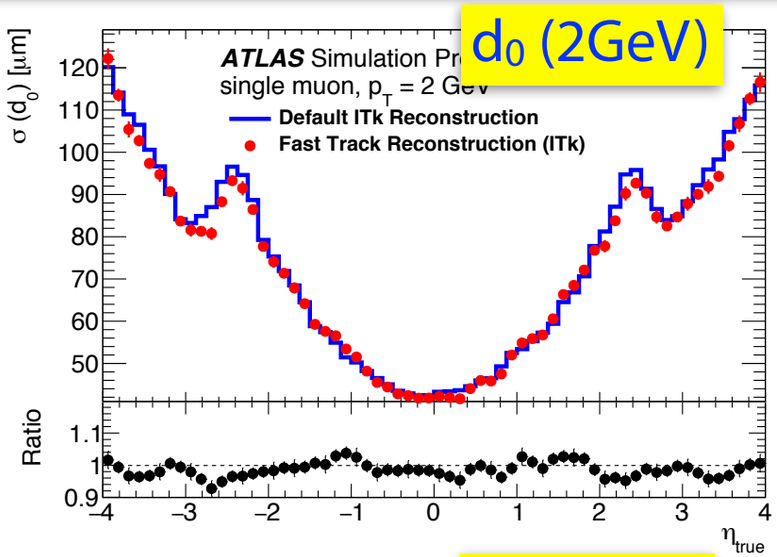


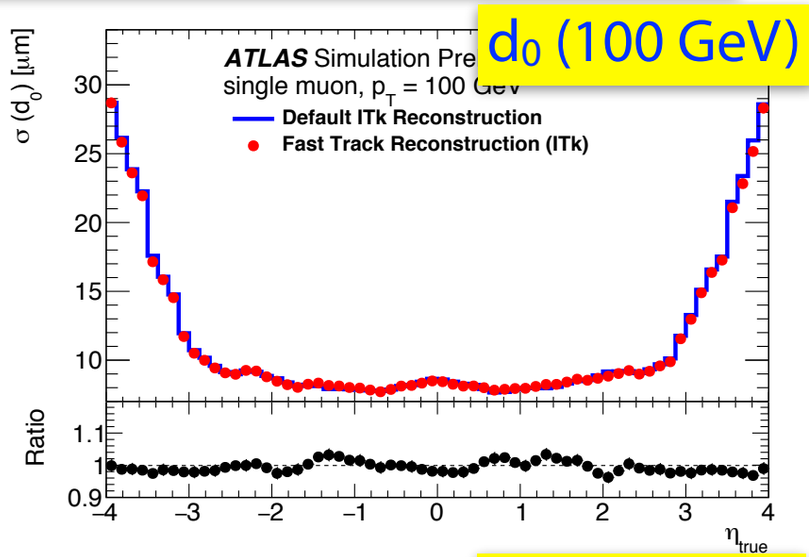


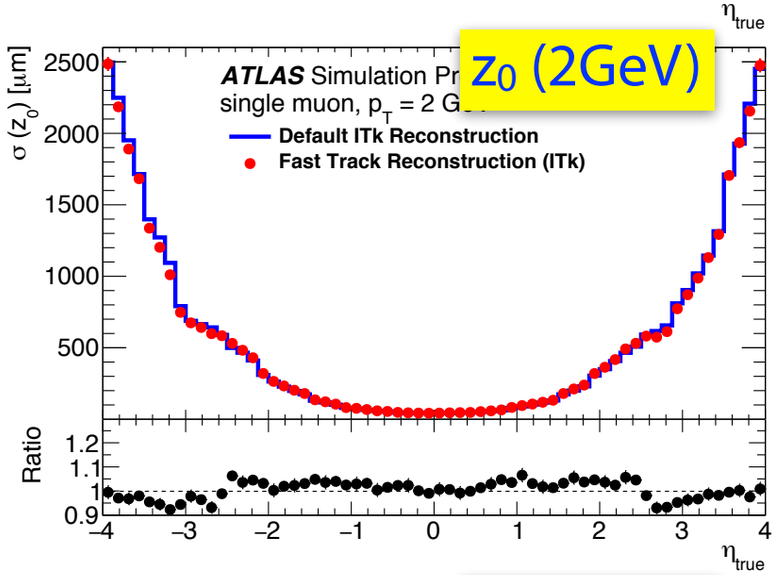


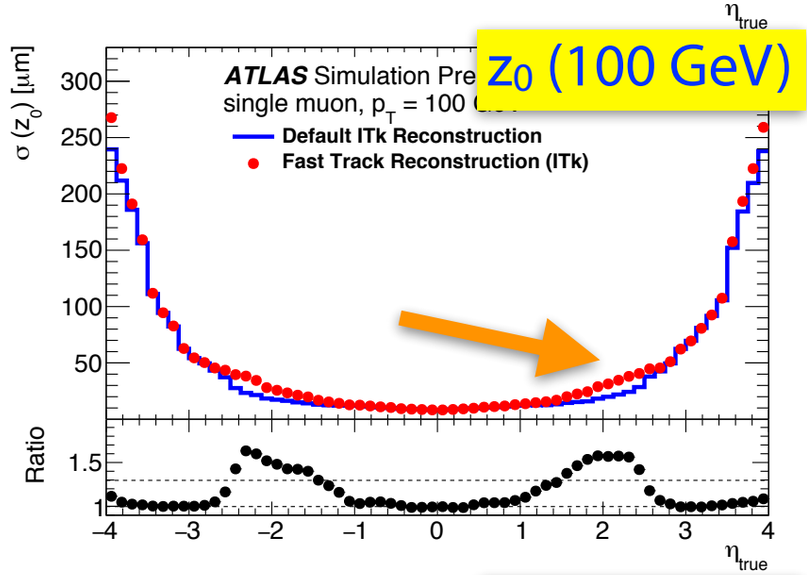
Performance of Fast Track Reconstruction

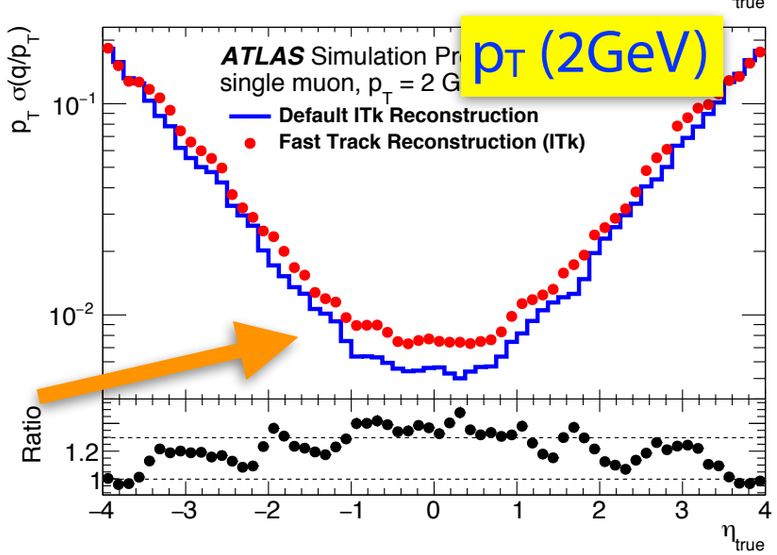
- track resolutions
  - ➡ single muons to illustrate effects
  - ➡ at 2 GeV:
    - 20% loss in p_T resolution
    - imperfect correction of multiple scattering effects
  - ➡ at 100 GeV:
    - 50% loss in z_0 around $|\eta|$~2
    - imperfect cluster correction in track fit
  - ➡ we do understand the sources of the issues !
- tracking performance already good enough for High Level Trigger
  - ➡ for offline, still work to do !

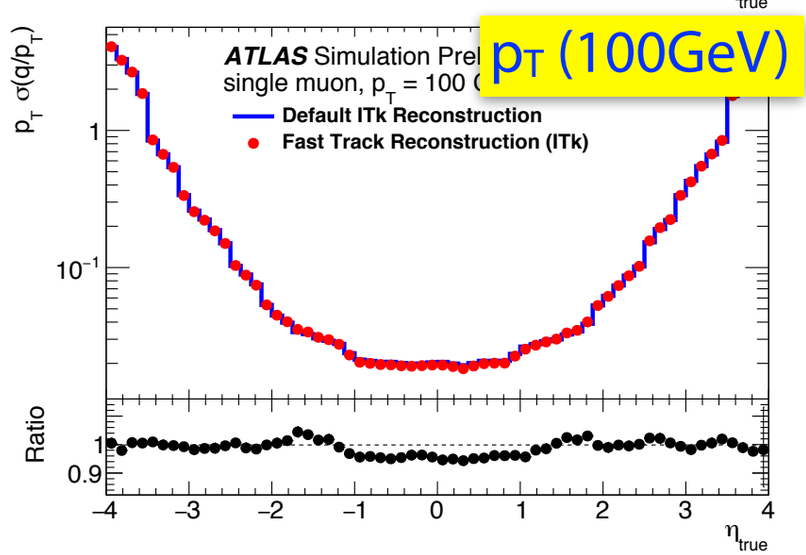






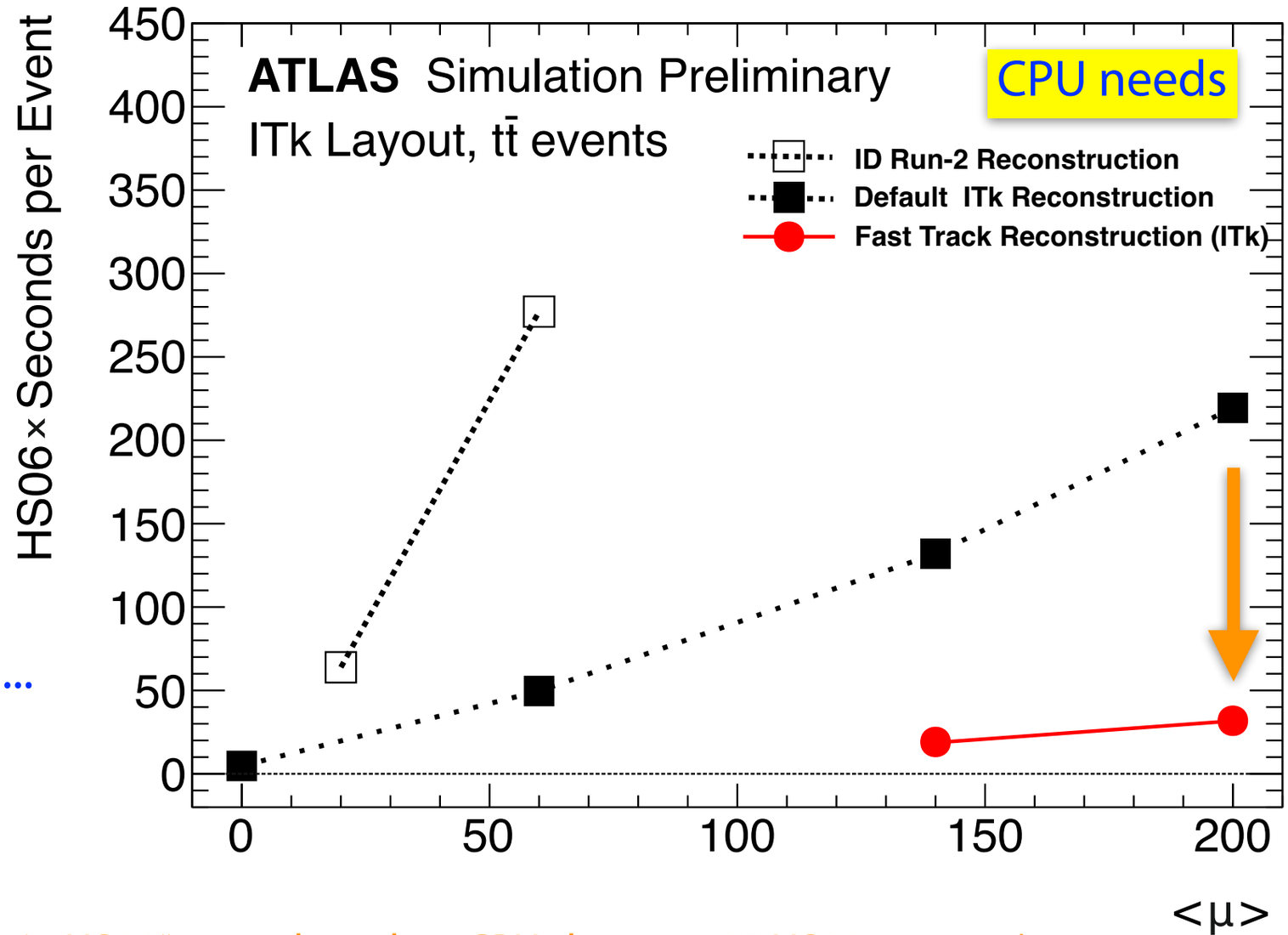






CPU Performance of Fast Track Reconstruction

- dramatic speed-up:
  - ➡ factor 6-7 overall for both, 140 and 200 pile-up
  - ➡ full detector reconstruction ($|\eta| < 4$, all layers)
  - ➡ results for $p_T > 1$ (0.4) GeV (global tracking requirement)
- software based on Run-2
  - ➡ old release, no benefit from ACTS...

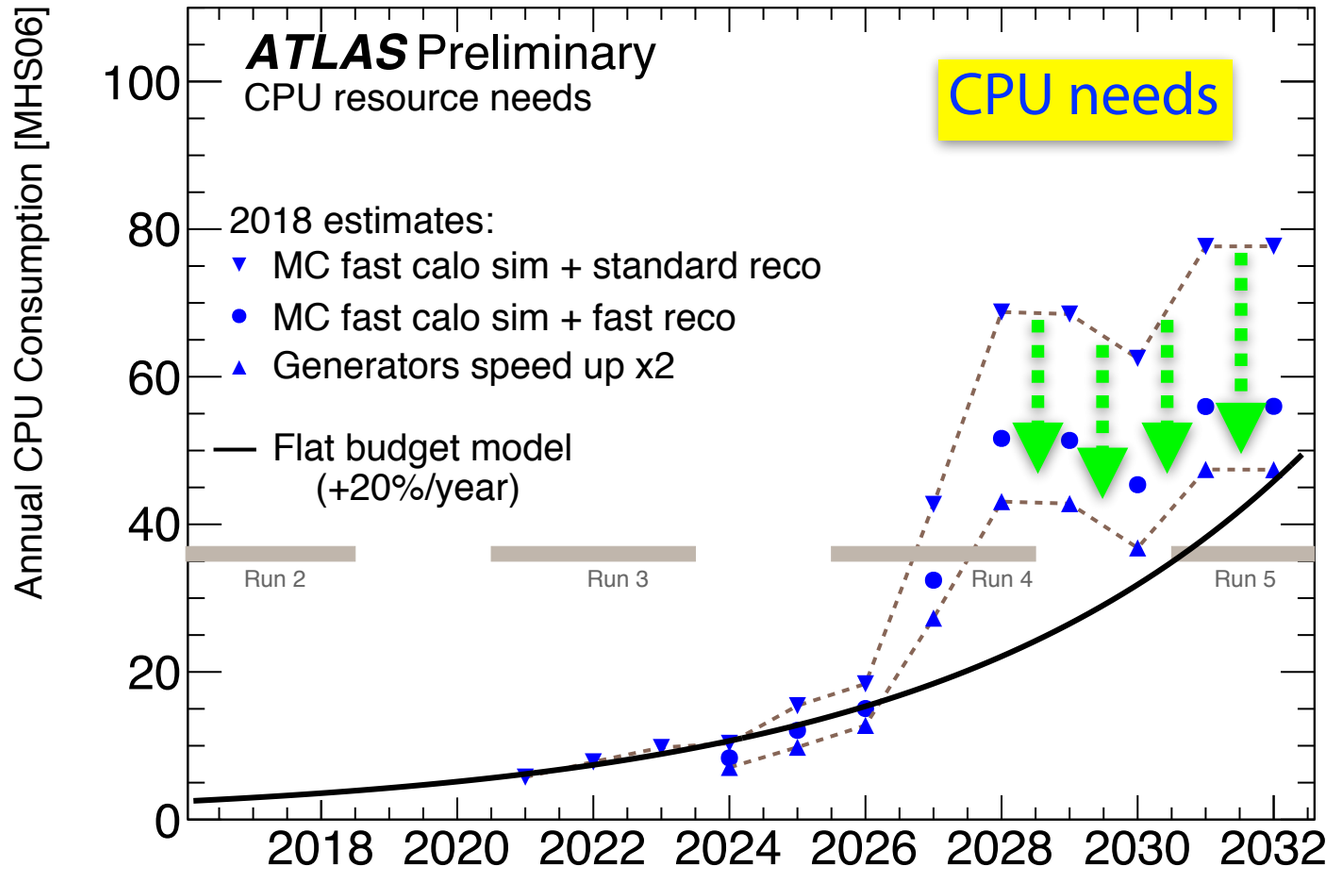


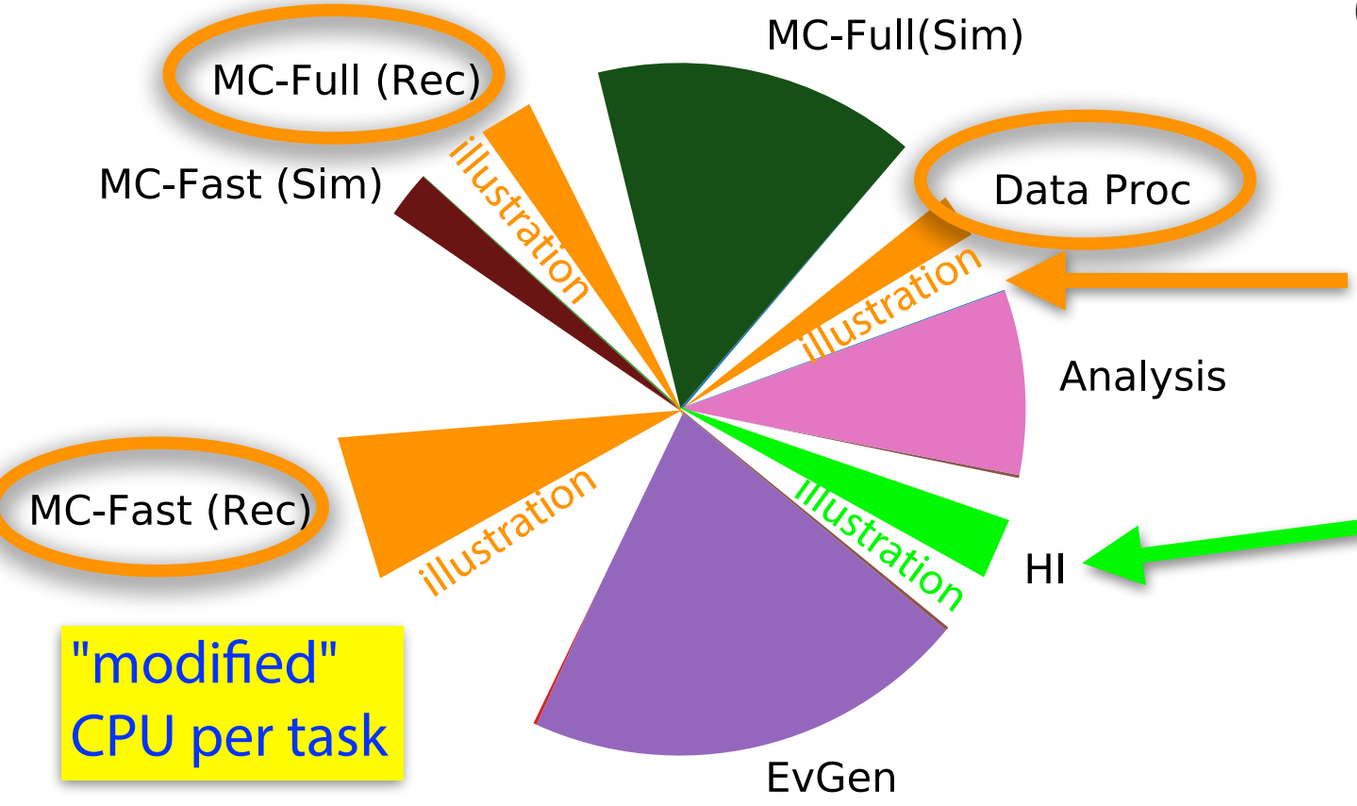
(numbers in HS06*seconds, todays CPUs have 15-20 HS06 per core)

$\langle\mu\rangle$	Tracking	Byte Stream Decoding	Cluster Finding	Space Points	Si Track Finding	Ambiguity Resolution	Total ITk
140	default	$1.2^{(*)}$	17.1	6.0	41.1	58.2	123.6
	fast		4.5	0.9	12.4	-	19.0
200	default	$1.6^{(*)}$	26.3	8.6	85.8	92.0	214.3
	fast		6.3	1.2	22.6	-	31.7

Implications on the Computing Model

- fast ITk track reconstruction matches previous model assumptions for "fast reco" !
  - ➡ "●" model assumes this for "MC-Fast"
  - ➡ will also reduce CPU for "MC-Full" and "Data Proc" !
  - ➡ model update for upcoming ATLAS HL-LHC Computing CDR (not yet publ





- naive illustration of reduction for current computing model
  - ➡ reduction in CPU for reconstruction
    (CPU for Heavy Ion mostly reconstruction as well)
  - ➡ CPU for event generators and G4 will dominate
  - ➡

Summary and Outlook

- fast ITk track reconstruction at ⟨μ⟩=200 is **8x** faster than Run-2 ID track reconstruction at ⟨μ⟩=60
  - ➡ physics performance already almost matches offline, exceeds trigger requirements

- demonstrated potential of the classical tracking software
  - ➡ of course, we don't want to lose physics performance
  - ➡ clear development plan to address problems without losing CPU performance

- ACTS essential to fix performance and to keep the CPU gains
  - ➡ ACTS comes with a fast Kalman Filter with full resolutions, aim is to use this directly in Silicon Track Finding (no refitting)
  - ➡ re-purpose Ambiguity Resolution to just deal with NN cluster splitting in jet cores
  - ➡ and ACTS data model will be faster (less malloc), ...

- in parallel, ATLAS continues intensive R&D on ML inspired tracking algorithms and on using co-processors
  - ➡ single tracking algorithm no longer CPU hot-spot in reconstruction
  - ➡ harder to gain from fine-grained offloading on co-processors...