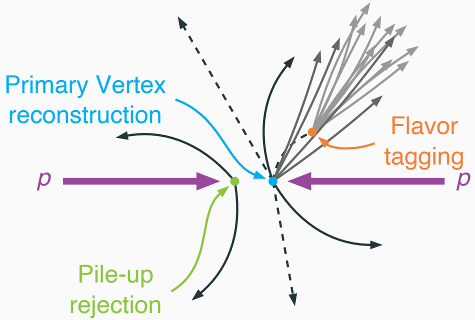
# Expected tracking performance of the ATLAS Inner Tracker Upgrade for Phase-II

Paul Gessinger — on behalf of the ATLAS Collaboration
CERN
2023-10-10 — Connecting The Dots 2023, Toulouse

# Tracking challenges at HL-LHC

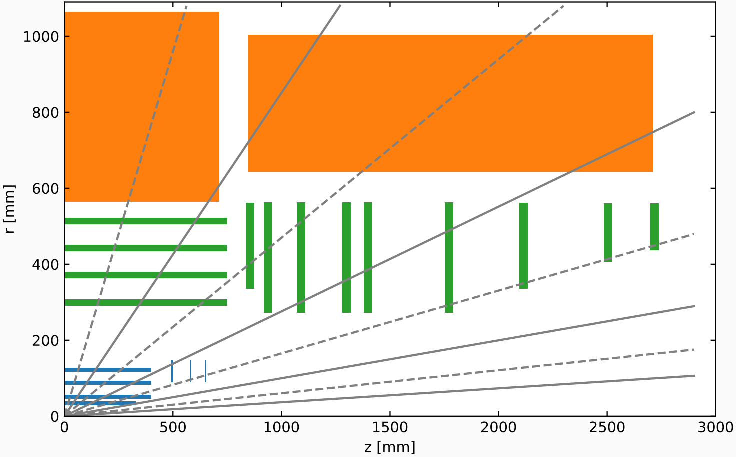
- Tracking is **fundamental for event reconstruction**:
  **almost everything** uses some form of **tracking information**
- Require tracking to
  - Be **highly efficient**
  - Have low **fake rate**
  - Produce **high-precision** tracks
  - Be **fast**!
- Tracking complexity scales with **number of interactions** ($\mu$):
  It's a combinatorial problem!
- Will push luminosity for HL-LHC
- Detector + reconstruction need to keep up!

| Run 2 | Run 3 | HL-LHC |
|---|---|---|
| $\langle\mu\rangle \sim 25$ | $\langle\mu\rangle \sim 50$ | $\langle\mu\rangle \sim 200$ |



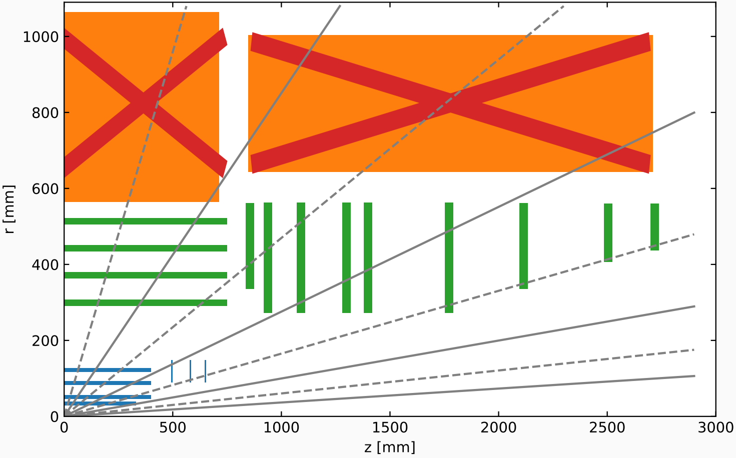
# ATLAS InnerTracker Phase-II Upgrade

# InnerDetector



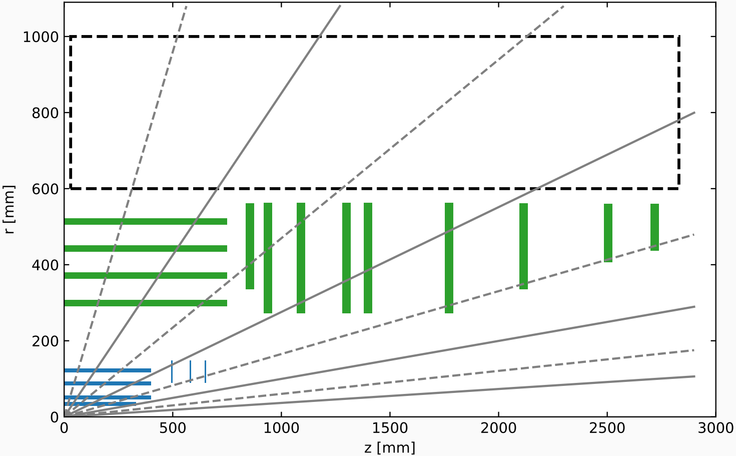

- Start from the InnerDetector

# InnerTetector



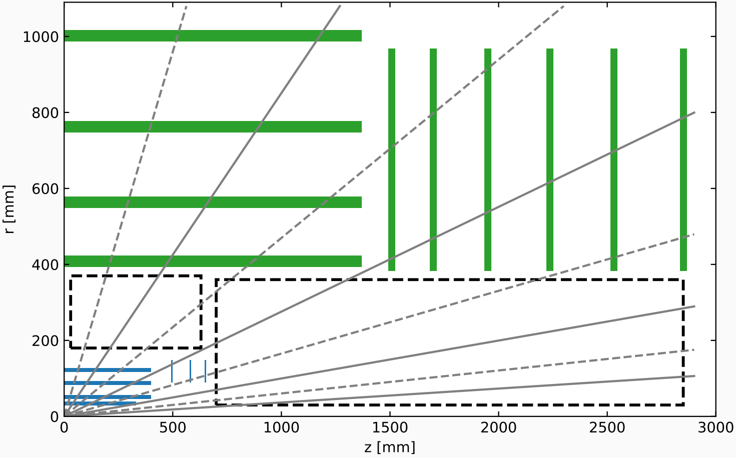

- Start from the InnerDetector
- **Drop** the TRT

# InnerTrtector



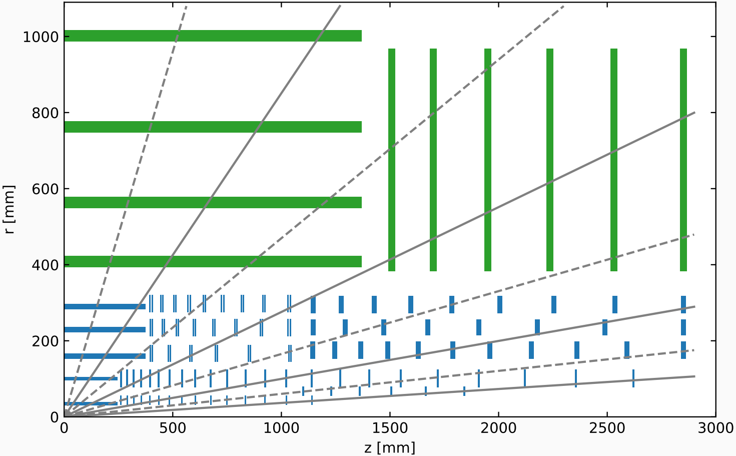

- Start from the InnerDetector
- **Drop** the TRT

# InnerTraector



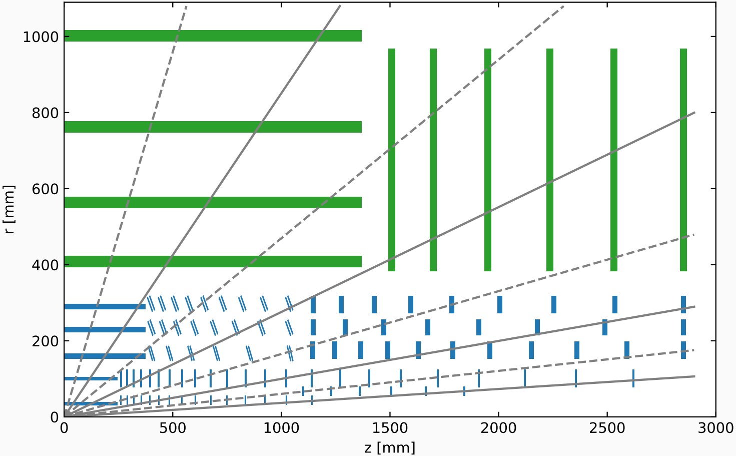

- Start from the InnerDetector
- **Drop** the TRT
- Get a **new Strip detector** with **larger radius**

# InnerTractor



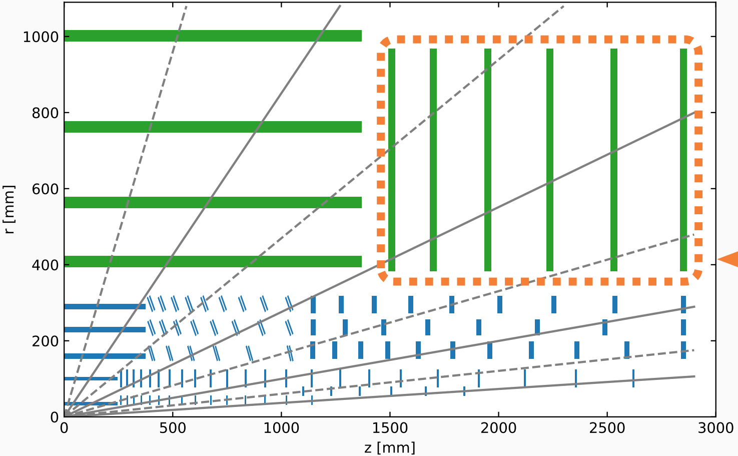

- Start from the InnerDetector
- **Drop** the TRT
- Get a **new Strip detector** with **larger radius**
- Swap out the Pixel detector, fill gap with **extra layers**

# InnerTracker

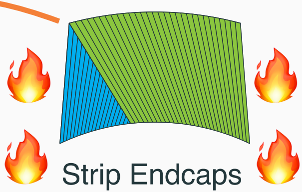



- Start from the InnerDetector
- **Drop** the TRT
- Get a **new Strip detector** with **larger radius**
- Swap out the Pixel detector, fill gap with **extra layers**
- Don’t forget to **incline your modules**!

# InnerTracker

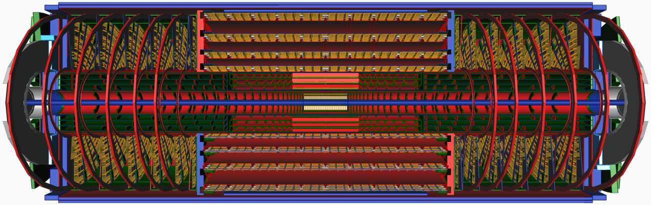


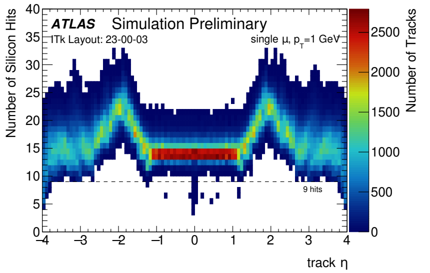

- Start from the InnerDetector
- **Drop** the TRT
- Get a **new Strip detector** with **larger radius**
- Swap out the Pixel detector, fill gap with **extra layers**
- Don't forget to **incline your modules**!




**Hottest sensor shape in the business!**

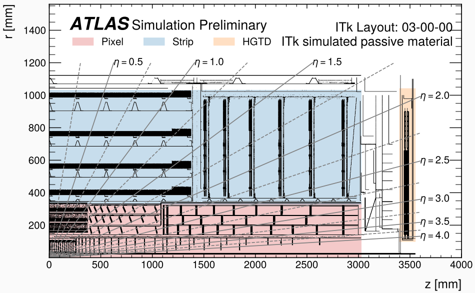
# ATLAS InnerTracker Phase-II Upgrade



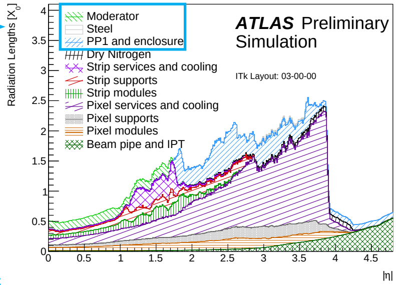


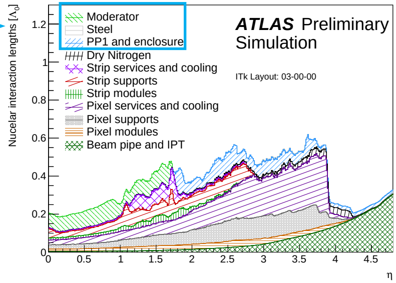

- **All-silicon** detector!
- Strip detector: **4** barrel layers + **6** endcap disks
- Pixel detector: **5** barrel layers + **inclined and vertical** rings
  - Lowest-radius measurement moves from 33.25 mm (IBL) to 34 mm
  - Pitch of innermost layer: $50 \times 250\ \mu m^2$ (IBL) $\Rightarrow 25 \times 100\ \mu m^2$
- Shown here: **best knowledge** of detector **at this time** ( 03-00-00 )!
  - Some results shown with 23-00-03 from 2021
  - Working on **comprehensive update as a paper**

# ITk structure & passive material




Largely behind sensitive layers

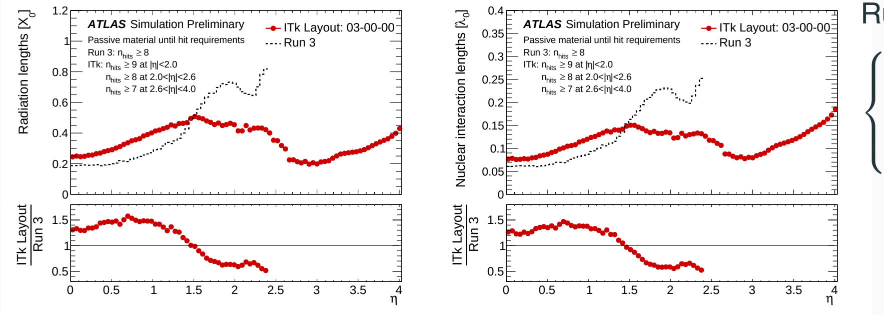



- **Material budget** is key consideration for ITk
- Simulation is evolving: best knowledge at this time!
  - +~20% material at $1.0 < |\eta| < 2.0$ w.r.t. ATL-PHYS-PUB-2021-024
- Full ITk now uses unified geometry toolkit ( GeoModelXML )

# ITk structure & passive material

Run 3: $n_{\text{hit}} \geq 8$




Run 4: $n_{\text{hit}} \geq \begin{cases} 9 & |\eta| < 2.0 \\ 8 & 2.0 \leq |\eta| < 2.6 \\ 7 & 2.6 \geq |\eta| \end{cases}$

- Key quantity of interest: **material until hit-requirement**
- Quantity has evolved as understanding of ITk construction improved
- Increase material until hit requirement in ITk at central $\eta$
- Crossover point at $|\eta| \sim 1.5$ due to inclined Pixel layers
  - Less material in ITk from about $|\eta| > 1.5$, to the end of ID coverage

# Software environment

`20.20` Dawn of Time → ∼2021

`21.9` 2021 → 2023

`main` 2023 → ??

`ACTS` for tracking

- Historically: ITk used **separate branch** for development
  - Herculean effort to **bring changes back** into modern toolchain (`21.9`, still decoupled, `ATL-PHYS-PUB-2021-024`)
  - Follow-up effort to merge with `main`, **reconcile with ongoing Run 3 developments**
  - Migration of tracking to **ACTS** for **performance** and long-term **maintainability**
- These are the **first public results on ITk performance** with the `main` **ATLAS software** + first public results of **ACTS in Athena**!

# Software environment

More on that later!

`20.20`

Dawn of Time → ∼2021

`21.9`

2021 → 2023

`main`

2023 → ??

`ACTS`

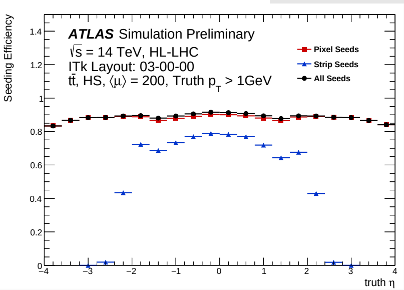
for tracking

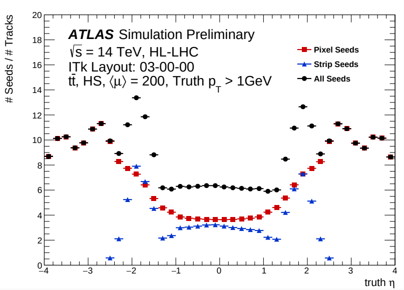
- Historically: ITk used **separate branch** for development
  - Herculean effort to **bring changes back** into modern toolchain ( `21.9` , still decoupled, `ATL-PHYS-PUB-2021-024` )
  - Follow-up effort to merge with `main` , **reconcile with ongoing Run 3 developments**
  - Migration of tracking to **ACTS** for **performance** and long-term **maintainability**
- These are the **first public results on ITk performance** with the `main` **ATLAS software** + first public results of **ACTS in Athena**!

# ITk tracking with the current software

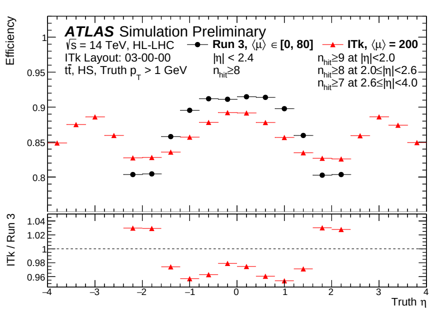
# Pattern recognition: Triplet seeding

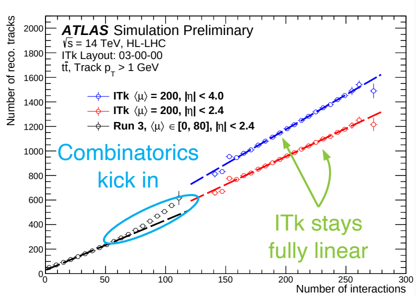
- First stage of ATLAS track finding: triplet seeding
- **Seed efficiency**: fraction of particles with a seed with >50% hits shared with the particle
- Independent efficiencies for Pixel- and Strip-only seeds
- Reco chain: **Strip seeds processed first**, Pixel seeds only used **if no track found**
  - High barrel efficiency even for Pixel-only due to 5 layers
  - Also enables early rejection by requiring 4th compatible hit
- **Seed redundancy**: number of seeds over number of tracks vs truth $\eta$
  - Well above one, illustrates robustness of pattern recognition



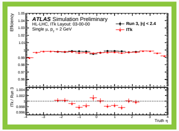
# Tracking efficiency in $t\bar{t}$

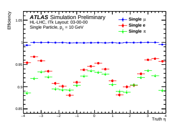


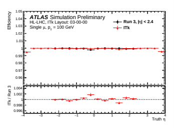



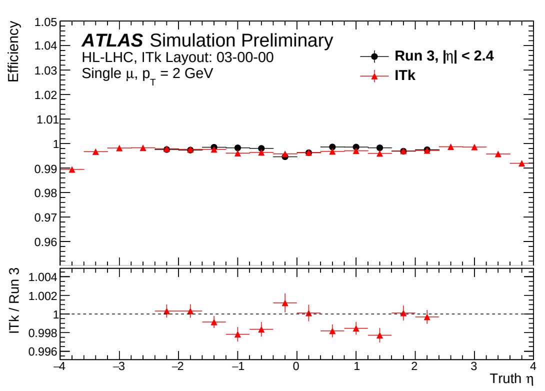

- **Tracking efficiency** at $\langle\mu\rangle = 200$ within 5% of Run 3 at $\langle\mu\rangle \in [0, 80]$ (particles $p_{\mathrm{T}} > 1$ GeV)
- **Increased hit requirements** for ITk largely **compensated** by optimized layout
- **SW improvement** for Run 3. But: still **becomes non-linear** at $\langle\mu\rangle \sim 80$
- Hardened hit req. and improved seed purity results in **linear** scaling of number of tracks with $\mu \rightsquigarrow$ **negligible fake-rate**

# Tracking efficiency in single-particles



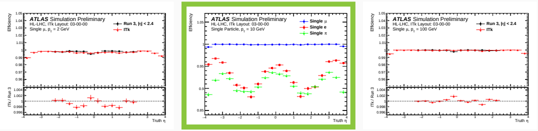



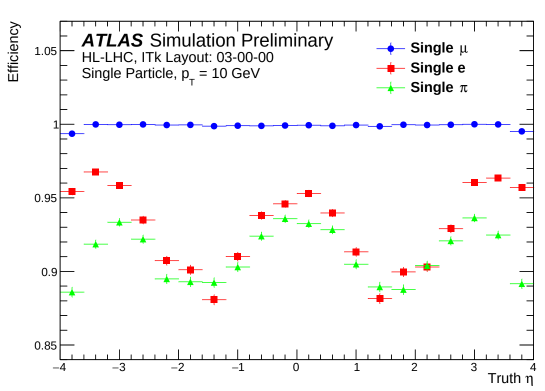




- Single-particle indicative of performance isolated from event activity
- Characterizes detector material, precision, etc.
- $\mu$ efficiency is **very high** as expected at all momenta
- *e* and $\pi$ efficiencies modulated with detector geometry / material
- High-$p_{\mathrm{T}}$ $\mu$ loss-rate of $\sim$140 ppm due to beam-spot requirements & pattern recognition edge effects

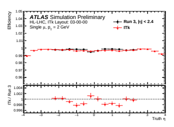
# Tracking efficiency in single-particles

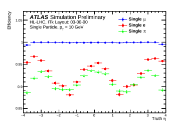


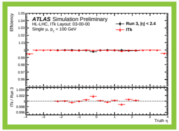


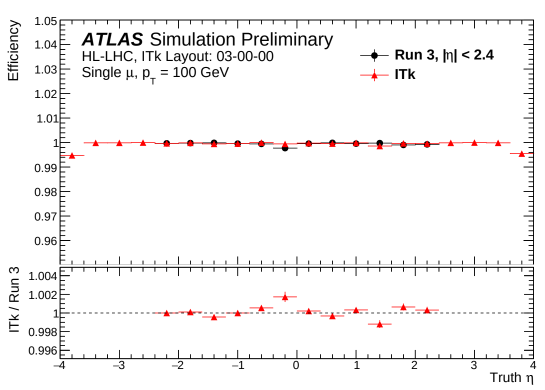

- Single-particle indicative of performance isolated from event activity
- Characterizes detector material, precision, etc.
- $\mu$ efficiency is **very high** as expected at all momenta
- *e* and $\pi$ efficiencies modulated with detector geometry / material
- High-$p_{\mathrm{T}}$ $\mu$ loss-rate of $\sim$140 ppm due to beam-spot requirements & pattern recognition edge effects

# Tracking efficiency in single-particles



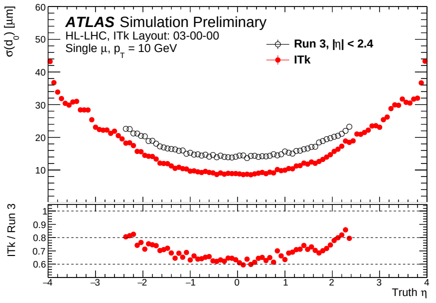




- Single-particle indicative of performance isolated from event activity
- Characterizes detector material, precision, etc.
- $\mu$ efficiency is **very high** as expected at all momenta
- *e* and $\pi$ efficiencies modulated with detector geometry / material
- High-$p_\mathrm{T}$ $\mu$ loss-rate of $\sim$140 ppm due to beam-spot requirements & pattern recognition edge effects

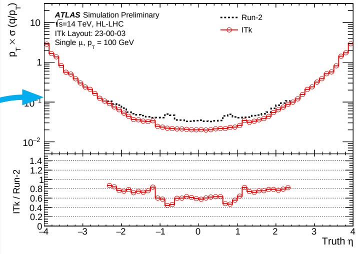
# Impact parameter resolution

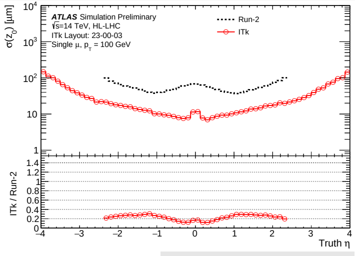
PLOTS-IDTR-2023-05




- ITk **improves** IP resolution w.r.t. the Run 3 detector across its coverage
  - Shown here: transverse IP $d_0$ at 10 GeV
  - Same picture at other momenta & $z_0$
- Critical for **pileup rejection** & **flavor tagging**
- Smaller Pixel pitch of innermost layer helps a lot! ($50\times250\ \mu m^2 \rightarrow 25\times100\ \mu m^2$)

- Silicon **strips outperform TRT** in bending plane: improved momentum resolution

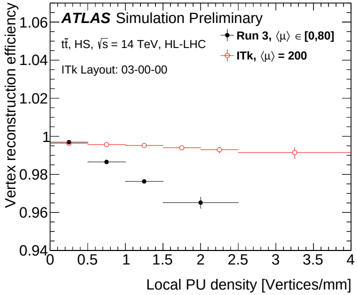


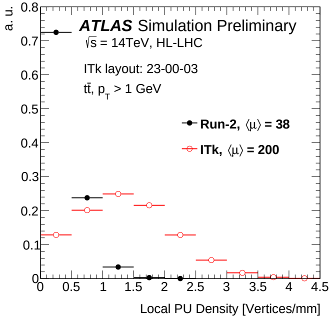




ATL-PHYS-PUB-2021-024

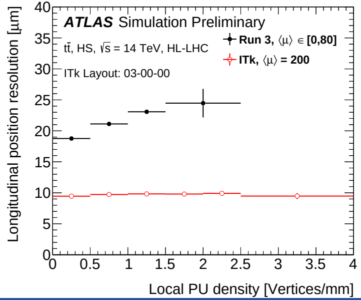
# Vertexing performance

- High pileup vertex density at $\langle \mu \rangle = 200$: primary vertexing challenging in presence of pileup tracks
  - Studied in $t\bar{t}$ events with pileup vertices
- Vertex efficiency **scales favorably with density**
- Longitudinal resolution at $\sim 10\ \mu m$: enables **robust pileup vertex rejection!**
- ITk reconstruction now (also) uses the **ACTS** *Adaptive Multi-Vertex Finder* algorithm
  - Deployed as default for Run 3




ATL-PHYS-PUB-2021-024



# ITk tracking with ACTS

# ACTS in ATLAS

## What is ACTS? (Repository)

- Experiment-independent toolkit for tracking
  - In-use or in evaluation by a number of experiments!
- Modern software, unit tested, continuous integration
- Minimal external dependencies
- Ready for multi-threading by design

## ACTS and ATLAS

- ATLAS will **heavily use ACTS** for reconstruction in HL-LHC
- **Most advanced** deployment: **ACTS for ITk** reconstruction
  - Projects in other domains ongoing!
- Development fully integrated into ATLAS software environment

# Why?

## ATLAS stepper transcribed to ACTS

```
double PC = pVector[4] * C[0]
  + pVector[5] * C[1] + pVector[6] * C[2];
double Bn = 1. / PC;

double Bx2 = -A[2] * pVector[29];
double Bx3 = A[1] * pVector[38]
  - A[2] * pVector[37];

double By2 = A[2] * pVector[28];
double By3 = A[2] * pVector[36]
  - A[0] * pVector[38];

double Bz2 = A[0] * pVector[29]
  - A[1] * pVector[28];
double Bz3 = A[0] * pVector[37]
  - A[1] * pVector[36];

double B2 = B[0] * Bx2 + B[1]
  * By2 + B[2] * Bz2;
double B3 = B[0] * Bx3 + B[1]
  * By3 + B[2] * Bz3;
```

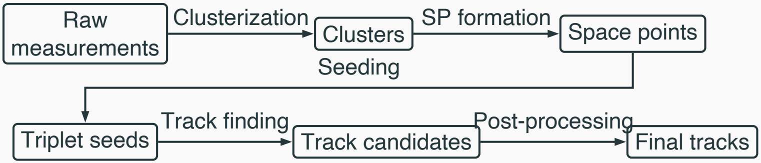
## ACTS reimplemented EigenStepper

```
boundToCurvilinearJacobian(direction,
                           boundToFreeJacobian,
                           freeTransportJacobian,
                           freeToPathDerivatives,
                           fullTransportJacobian);

boundCovariance = fullTransportJacobian
                  * boundCovariance
                  * fullTransportJacobian.transpose();

reinitializeJacobians(freeTransportJacobian,
                      freeToPathDerivatives,
                      boundToFreeJacobian,
                      direction);
```

🚨 **Software written 30+ years before with no one still around who wrote it is not maintainable!**

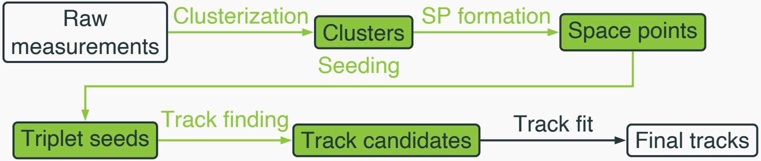
*not *exactly* identical code

# ACTS for ITk deployment




- Adding implementations using ACTS tools
- Add option allowing running a tracking chain with pieces using ACTS
- At the same time: enable ACTS output Event Data Model (EDM) to use ATLAS IO infrastructure
  - Converters for validation to allow reusing robust tooling in place
  - No conversions foreseen for final configuration
- At this time: **full tracking chain using ACTS available!**
  - Post-processing currently using non-ACTS components

# ACTS for ITk deployment




**Implemented with ACTS!**

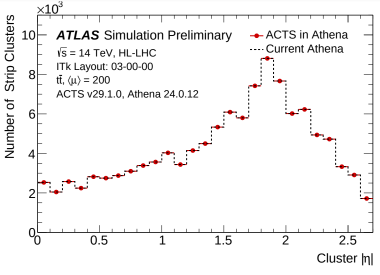
- Adding implementations using ACTS tools
- Add option allowing running a tracking chain with pieces using ACTS
- At the same time: enable ACTS output Event Data Model (EDM) to use ATLAS IO infrastructure
  - Converters for validation to allow reusing robust tooling in place
  - No conversions foreseen for final configuration
- At this time: **full tracking chain using ACTS available!**
  - Post-processing currently using non-ACTS components

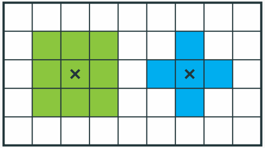
# ACTS Clusterization

PLOTS-IDTR-2023-04

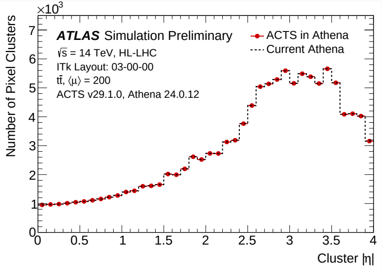
- Reimplementation of pixel and strip clustering
  - Based on prior ATLAS implementation, with some modifications
- Number of clusters and cluster sizes agree with current ATLAS SW
- Slightly favorable timing compared to current ATLAS SW

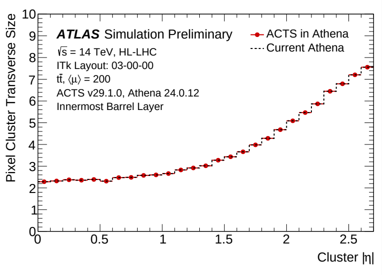
## Strips



## Pixel



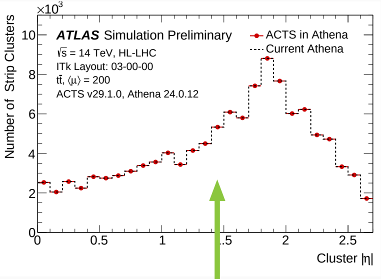



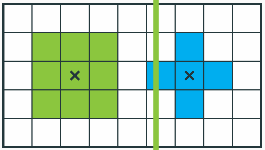
# ACTS Clusterization



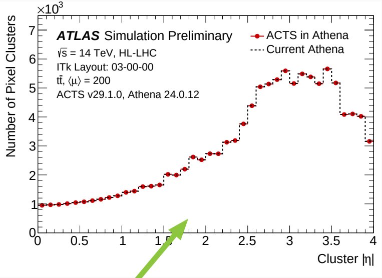
- Reimplementation of pixel and strip clustering
  - Based on prior ATLAS implementation, with some modifications
- Number of clusters and cluster sizes agree with current ATLAS SW
- Slightly favorable timing compared to current ATLAS SW

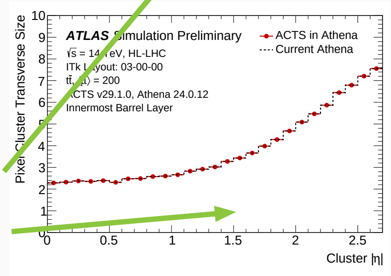
Strips



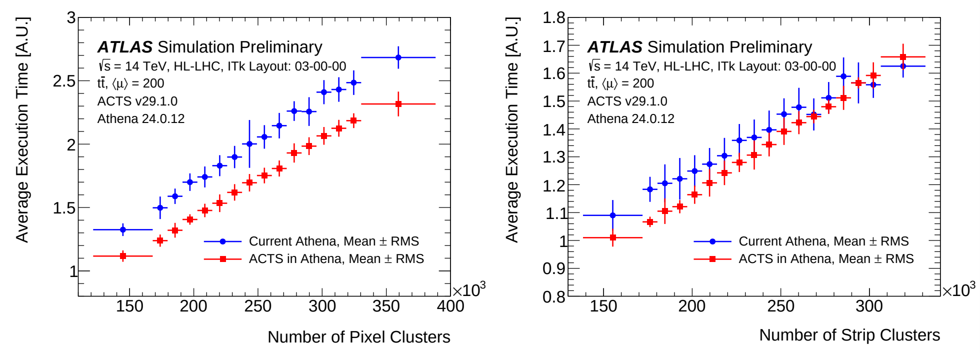
Pixel




**Clustering: exact problem!**
**Exactly 1 right solution.**
**100% agreement**

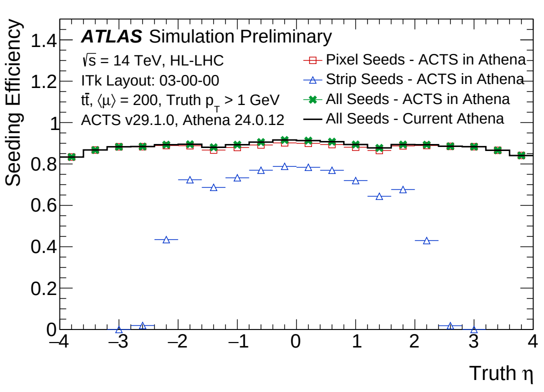
# ACTS Clusterization timing




- ACTS Clusterization faster than previous Athena implementation
- Pixel: timing differences constant vs. event complexity
- Strips: ACTS implementation has larger speedup at lower complexity

# ACTS Seeding performance

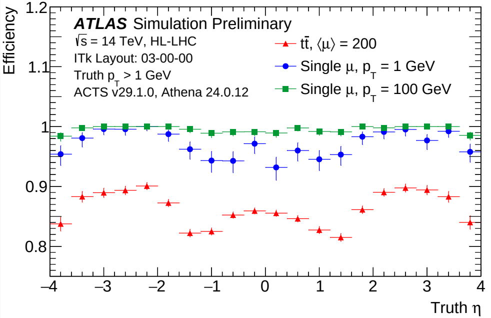
- Reimplementation of ATLAS seeding in ACTS [1]
- Validated to reproduce 1:1 identical seeds to non-ACTS seeding
- Trade-off of ultimate CPU performance vs. maintainability:
  - No magic numbers!
  - Can tune without changing hard-coded values, less risk of inconsistencies
  - Currently about 10% slower than Athena implementation
  - Optimization ongoing




[1] See L. Coelho's talk tomorrow

# ACTS tracking efficiency

- Simulation of $t\bar{t}$ events with pileup with standard Athena workflow
- Configured ACTS tracking chain: clusterization, space-point formation, seeding, combinatorial track finding using Combinatorial Track Finder (CKF)
- CKF is a complete reimplementation!
- Outputs converted to standard ATLAS tracks
  - Ambiguity resolution (without refitting) using non-ACTS tools
  - Standard performance validation toolchain

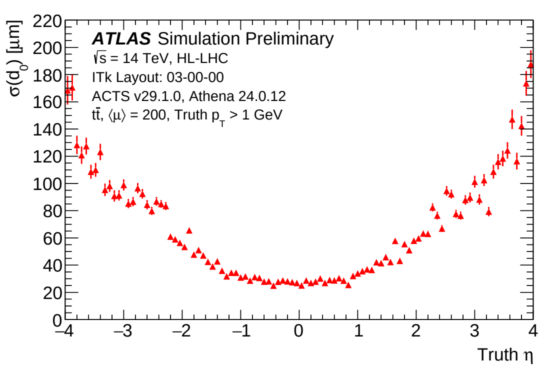



- Single $\mu$ efficiencies reasonably high (note: reconstruction requires $p_{\mathrm{T}} > 900$ MeV at central $\eta$!)
- $t\bar{t}$ efficiency within striking distance target performance

**All of this is pending thorough optimization & tuning!**

# ACTS resolution

- Transverse impact parameter resolution directly from the ACTS CKF (no refitting)
- Produces results compatible with the standalone Kalman Filter
  - Caveats: no smoothing, no in-fit measurement calibration
  - KF independently validated against the ATLAS workhorse: the global $\chi^2$ fitter
- Resolution here mainly due to composition of track population under study



# Conclusion

## Current Athena

- ITk reconstruction runs on main ATLAS software branch **for the first time!**
  - No more disconnected branch!
- Brought forward **all ITk specific developments**
- **Most up-to-date description** of ITk available
- Results show **excellent performance** of ITk reconstruction
- Will serve as **baseline for developments** going forward!

## ACTS for ITk

- To ensure long-term maintainability of tracking SW:
  migration to ACTS for tracking!
- Deployment is progressing
- Full ACTS-based tracking chain now available!
- Reimplementations of existing algorithms + new algorithms
- **Performance is promising** but still needs optimization

# Backup

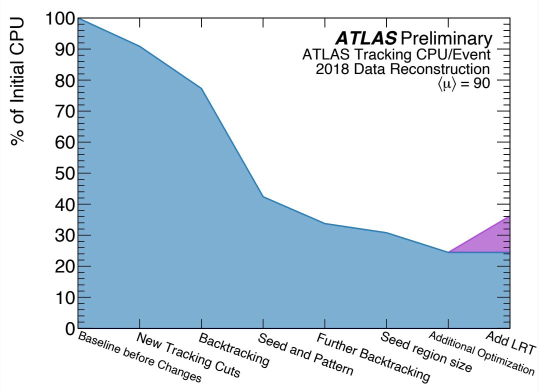
# Truth matching

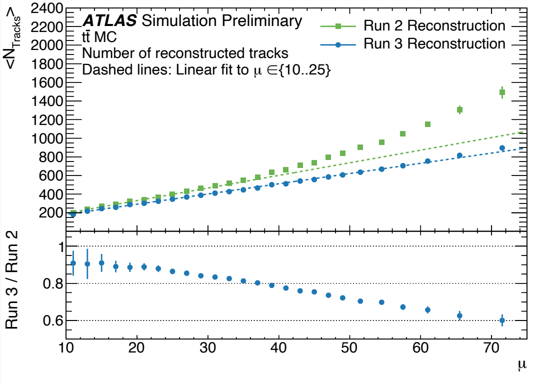
- Efficiencies and resolutions use matched truth particles
- Not trivial since tracks are assembled from individual hits
- Truth matching uses a **truth-match probability**:

$$P_{\text{match}} = \frac{10 \cdot N_{\text{Pixel}}^{\text{common}} + 5 \cdot N_{\text{Strip}}^{\text{common}}}{10 \cdot N_{\text{Pixel}}^{\text{track}} + 5 \cdot N_{\text{Strip}}^{\text{track}}}$$

- Matched particle: $P_{\text{match}} > 0.5$

# Software improvements for Run 3




ATL-PHYS-PUB-2021-012