HL-LHC ATLAS 4D tracking
ACTS CKF timing reconsustruction

Rodrigo Estevam de Paula

Marco Aurelio Lisboa Leite

Vitor Heloiz Nascimento

21 August, 2024

Quick Updates

Updates:

- ACTS Workshop submission deadline is September 15th, we need to decide if we are going to submit something
    - I plan to participate even without a submission
- Ramping up with Qualification Project activities
    - bi-monthly meetings on Monday mornings
    - I will also attend Software integration meetings every Wednesday morning

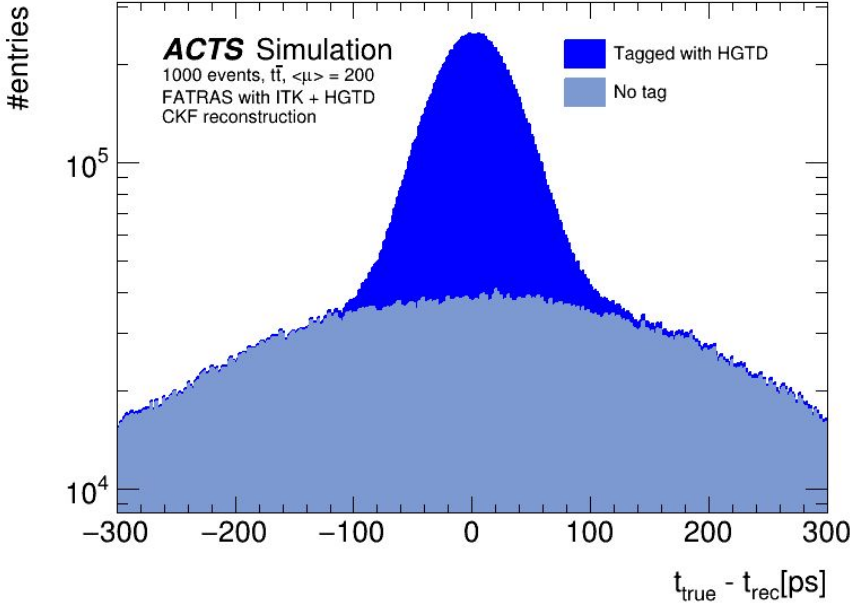
What to discuss today:

- Summary of our work with tracking and vertexing using time information
- Evaluating the performance of the reconstruction considering sensor degradation
- Next steps of the project

CKF Time reconstruction

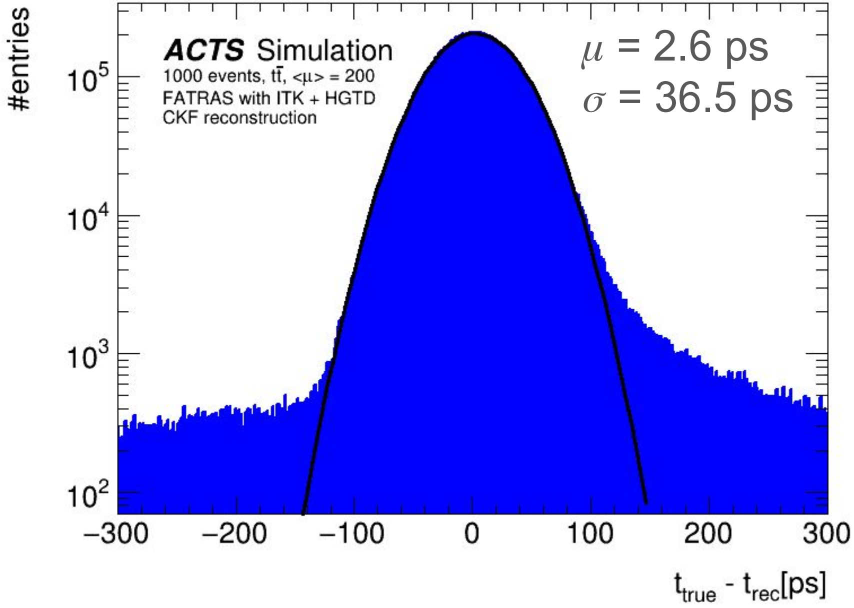
Update on performance plots: Residue plot

- It is possible to isolate tracks tagged by HGTD with the reconstruction covariance matrix
- Tracks without HGTD tagging rely on first estimate for time reconstruct, which has a high error associated with.
- When a track reaches HGTD, the error drops significantly because of the low error attributed to the measure
  - In the plot aside, the covariance threshold to separate the tracks (err_eT) was set to 1000



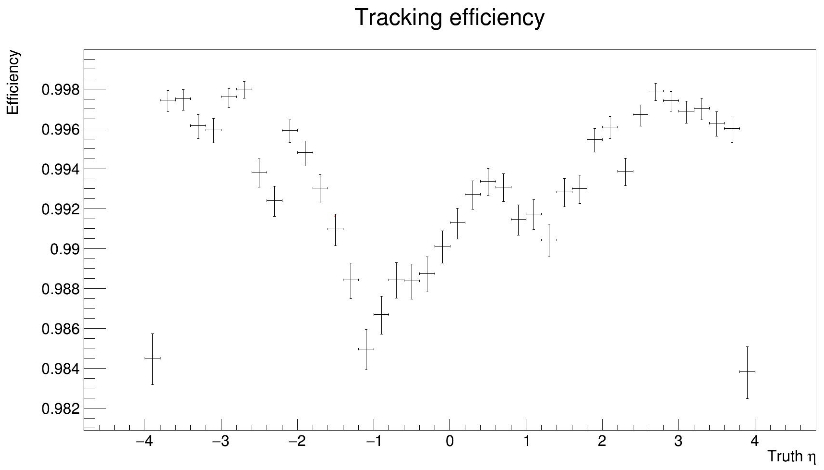
Update on performance plots: Residue plot

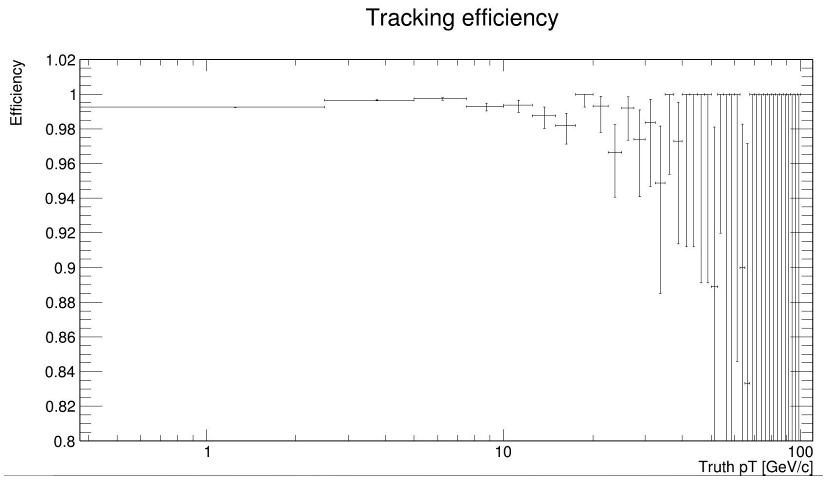
- Isolating only hits from tagged tracks is possible to measure resolution of reconstruction
- The curve asymmetry can be explained by:
  - approximation of particle speed to the speed of light
  - linearization of the particle path between hits



Efficiency plots

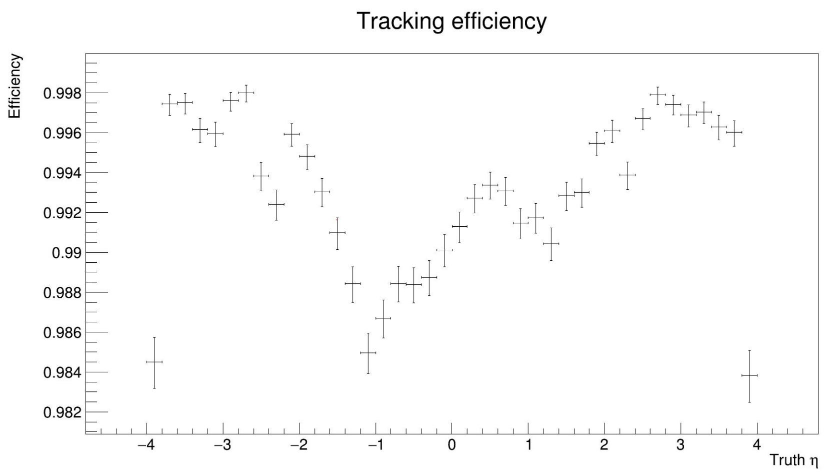
- We can measure efficiency by $|\eta|$, pT of the track and other geometric and kinematic parameters.
- The following plots were generated by an automatic performance output from ACTS
    - 1000 ttbar events with 200 of pileup

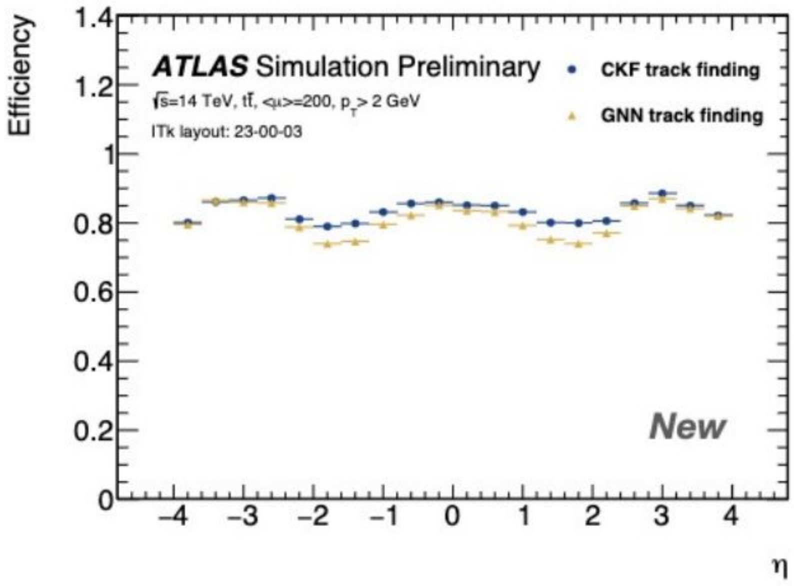




Understanding efficiency plots - eff vs eta

- We still need to understand the difference between our plots and the ones ATLAS produced
  - Could it be explained by truth seeding options?

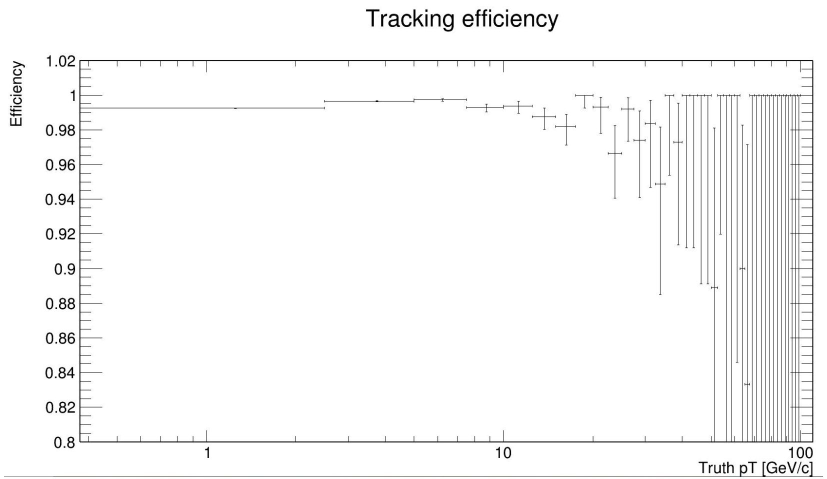


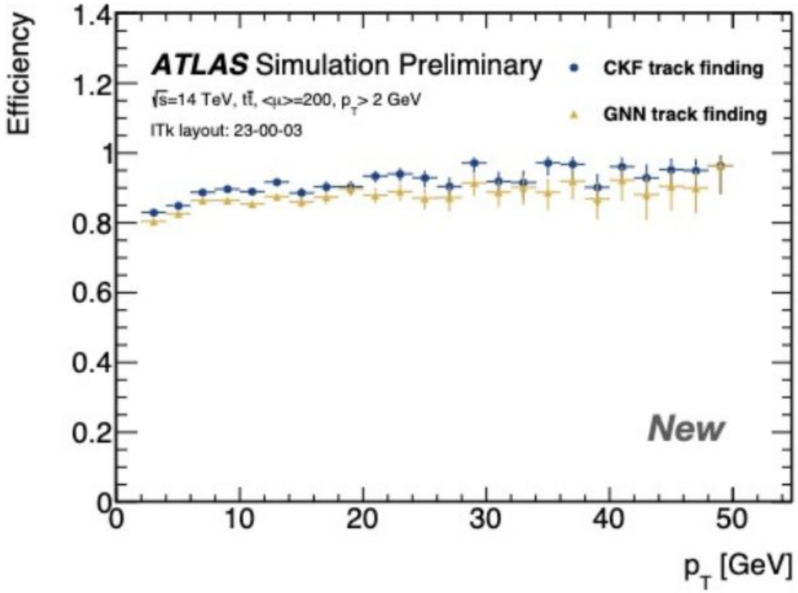


ATLAS plots taken from [here]

Understanding efficiency plots - vs pT

- We still need to understand the difference between our plots and the ones ATLAS
  - Could it be explained by truth seeding options?
  - Is it another metric being used?

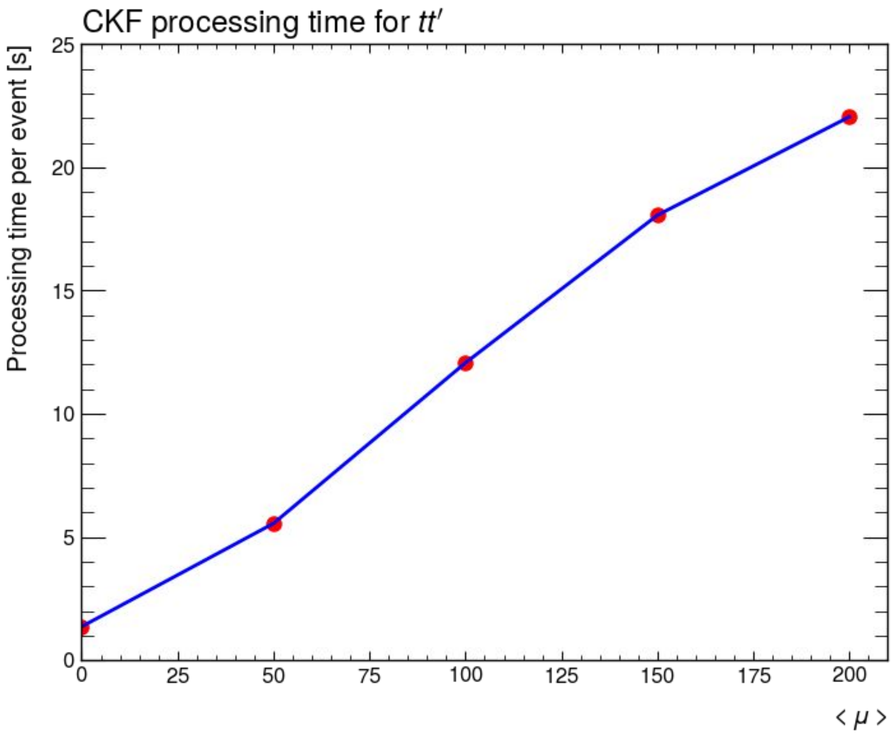




ATLAS plots taken from [here]

Update on performance plots: Processing time

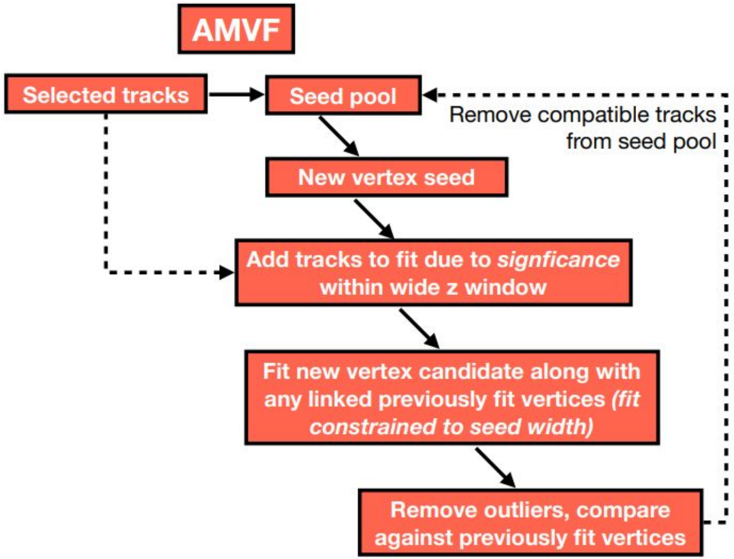
- Measured the processing time of CKF for different levels of pileup
- Simulated 500 events with multi-threading enabled
  - Events are distributed evenly across the threads
- Estimate made summing all time spent on the CKF step and divided by number of events
  - Can vary for different machines



AMVF Time reconstruction

AMVF overview

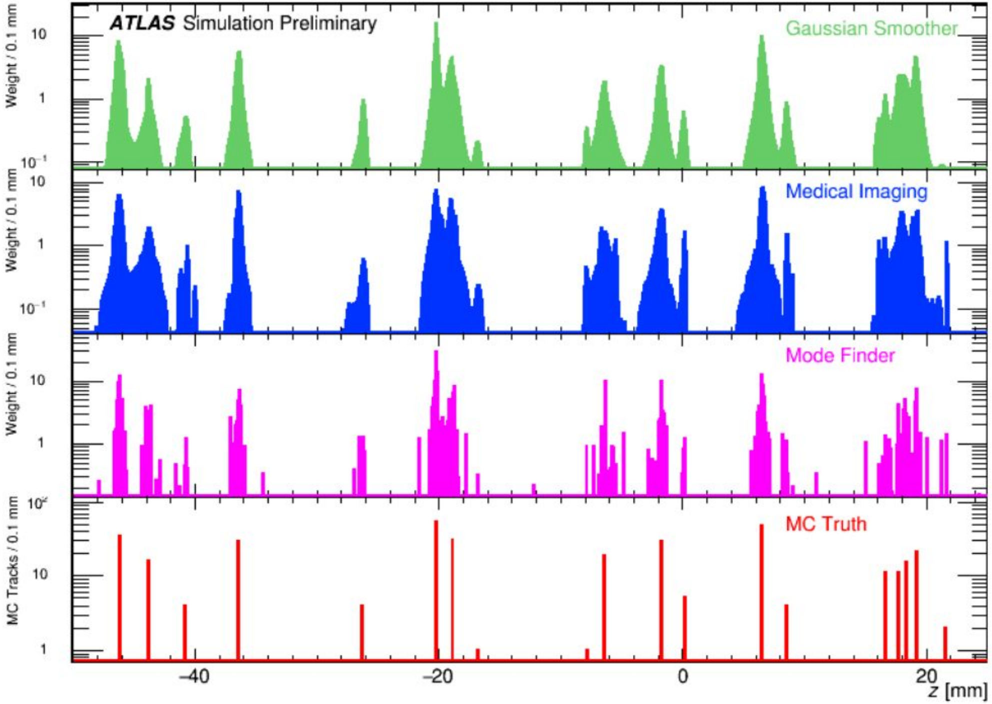
- I managed to include time information in the AMVF reconstruction
- The following slides will be an overview of the method in order to understand its performance
- The process can be divided into three steps:
    - Vertex seeding: Gaussian Track Density
    - Vertex finding: AMVF
    - Vertex fitting: Kalman Filter updater
- The steps are looped until all (valid) tracks are assigned to a vertex



- Reference for this section:
    - [ATLAS Collaboration (2019). Development of ATLAS Primary Vertex Reconstruction for LHC Run 3 [White paper]. CERN.]

Vertex seeder

- The seeding step establishes first estimates for primary vertices positions



- Most of the vertex seeders project tracks to origin and evaluate density distribution
    - The peaks of the distribution will be the seeds for vertices
-

Gaussian Track Density

- The density of track origin can be represented by a multi-variate Gaussian distribution:

$$P(r,z) = \frac{1}{2\pi\sqrt{|\Sigma|}} e^{-\frac{1}{2}((r-d_0),(z-z_0))^T \Sigma^{-1}((r-d_0),(z-z_0))}.$$

$$\Sigma = \begin{pmatrix} \sigma^2(d_0) & \sigma(d_0, z_0) \\ \sigma(d_0, z_0) & \sigma^2(z_0) \end{pmatrix}.$$

- If we project it to the z-axis, we have the density at that axis. Furthermore, If we consider the distribution to be locally Gaussian, it's possible to do a peak search with steps of size:

$$W(z) = \sum_{i \in \text{tracks}} P_i(0, z).$$

$$\Delta z = \frac{W(z)W'(z)}{W'^2(z) - W''(z)W(z)}.$$

- The output will be the position of the peak and the width of the distribution around it

$$z_{max} = \max_z W(z)$$

$$\sigma(z) = \sqrt{-\frac{W(z_{\max})}{W''(z_{\max})}}$$

Adaptive Multi Vertex Finding (AMVF)

- Given a collection of reconstructed tracks and estimates of vertexes, establishes a “compatibility” value for each track-vertex
- The algorithm is adaptive in a sense that vertexes compete for the same track (multiple vertex-tracks weights)
- Iterate over association weights until convergence
- Short paper explaining

Fitting procedure

- Fit all vertexes using the assignment probability as track weights
- Recompute the assignment probabilities using the most recent vertex positions

Weight function

- Having n tracks to be fitted to m vertexes

The weight of vertex j to track i is:

$$w_{ij} = \frac{\exp(-\chi^2_{ij}/2\,T)}{\exp(-\chi^2_{\text{cut}}/2\,T) + \sum_{k=1}^{m} \exp(-\chi^2_{ik}/2\,T)}$$

T is a temperature parameter
X^2cut is a cut-off to suppress not assigned tracks
X^2ij is the chi2 distance between track and vertex

Kalman Filter Updater

- From the collection of tracks assigned to a vertex originated from the previous step, a Kalman Filter is used to fit the vertex position
- The position of the first deposition (measurement) is used to evaluate the vertex position and momentum of the track (vector state)
- The measurement equation would be:

$$\boldsymbol{q}_i = \boldsymbol{h}_i(\boldsymbol{v}, \boldsymbol{p}_i),\ i = 1 \ldots, n.$$

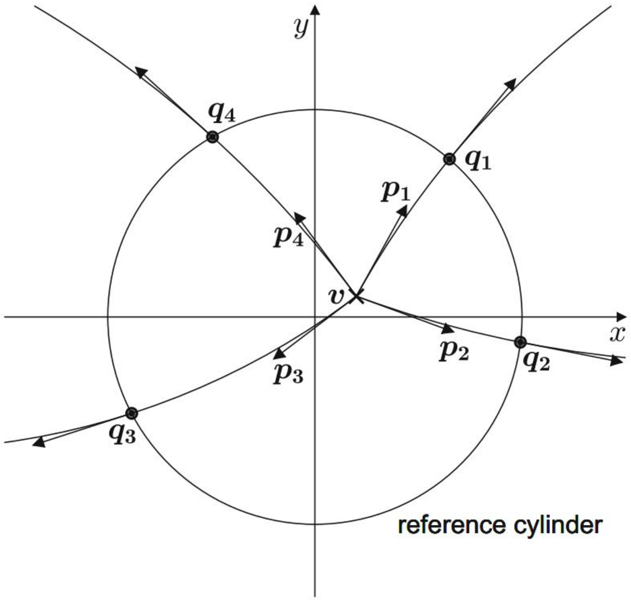


Fig. 8.1 A vertex fit with four tracks. The parameters of the fit are the vertex $\boldsymbol{v}$ and the momentum vectors $\boldsymbol{p}_i$; the observations are the estimated track parameters $\boldsymbol{q}_i$

Kalman Filter updater equations

- The measurement equation is linearized so the Kalman Filter can be used
- Check on the book for detailed explanation
- The AMVF weights multiply the inverse of the covariance matrix as to imitate a “significance” of the measurement

$$\boldsymbol{v}_i = \boldsymbol{C}_i \left[\boldsymbol{C}_{i-1}^{-1} \boldsymbol{v}_{i-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} (\boldsymbol{q}_i - \boldsymbol{c}_i) \right],$$

$$\boldsymbol{p}_i = \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i (\boldsymbol{q}_i - \boldsymbol{c}_i - \boldsymbol{A}_i \boldsymbol{v}_i),$$

update equation for both vertex position and particle momentum

$$\mathrm{Var}[\boldsymbol{v}_i] = \boldsymbol{C}_i = \left(\boldsymbol{C}_{i-1}^{-1} + \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i^{\mathrm{B}} \boldsymbol{A}_i \right)^{-1},$$

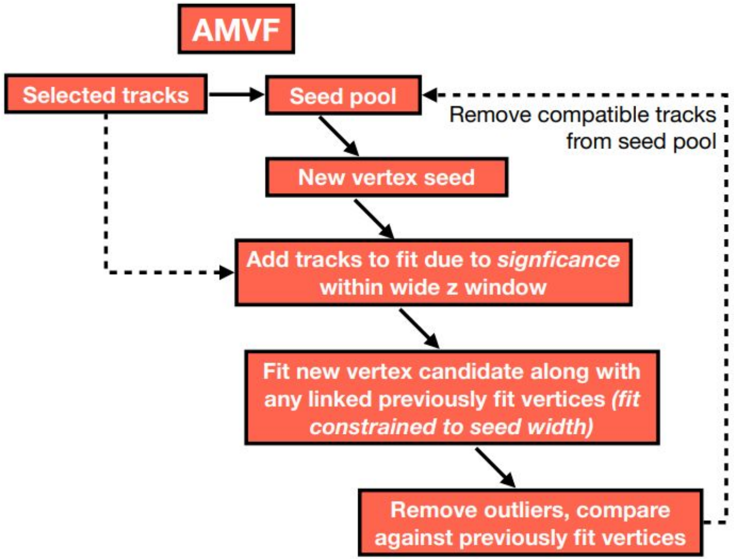
$$\mathrm{Var}[\boldsymbol{p}_i] = \boldsymbol{W}_i + \boldsymbol{W}_i \boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i,$$

$$\mathrm{Cov}[\boldsymbol{v}_i, \boldsymbol{p}_i] = -\boldsymbol{C}_i \boldsymbol{A}_i^{\mathsf{T}} \boldsymbol{G}_i \boldsymbol{B}_i \boldsymbol{W}_i.$$

update equation for both vertex position and particle momentum

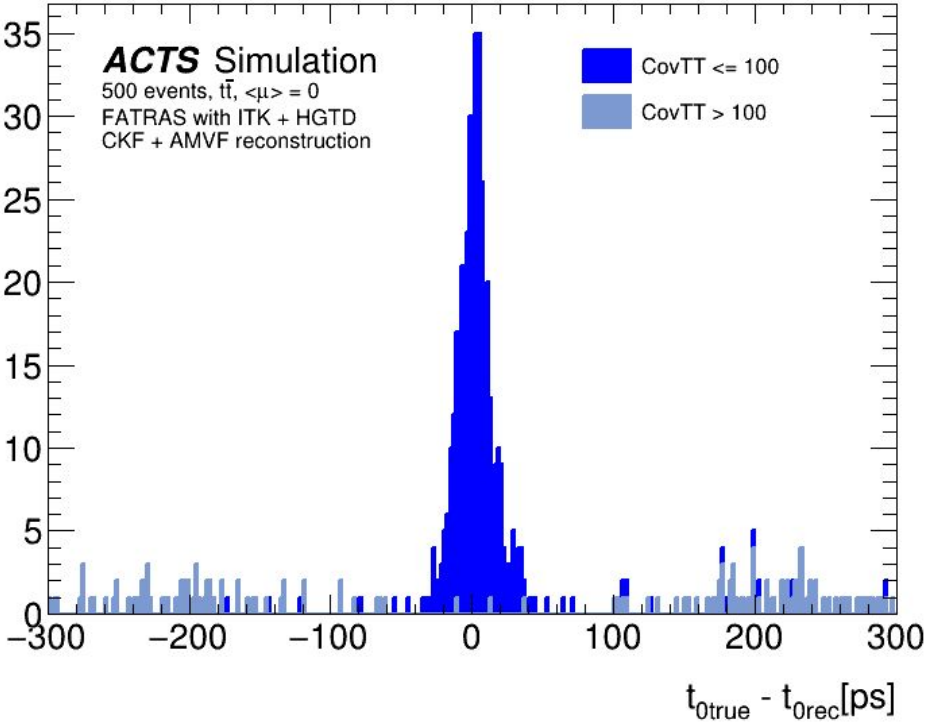
AMVF overview (again)

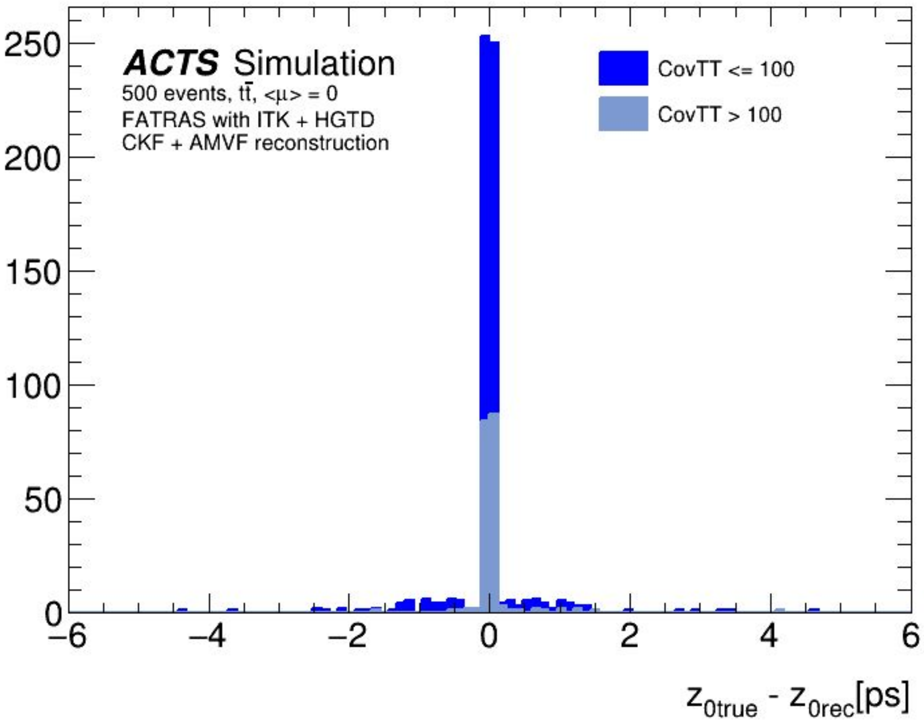
- The process can be divided into three steps:
  - Vertex seeding: Gaussian Track Density
  - Vertex finding: AMVF
  - Vertex fitting: Kalman Filter updater
- The steps are looped until all (valid) tracks are assigned to a vertex



AMVF Vertex Time reconstruction

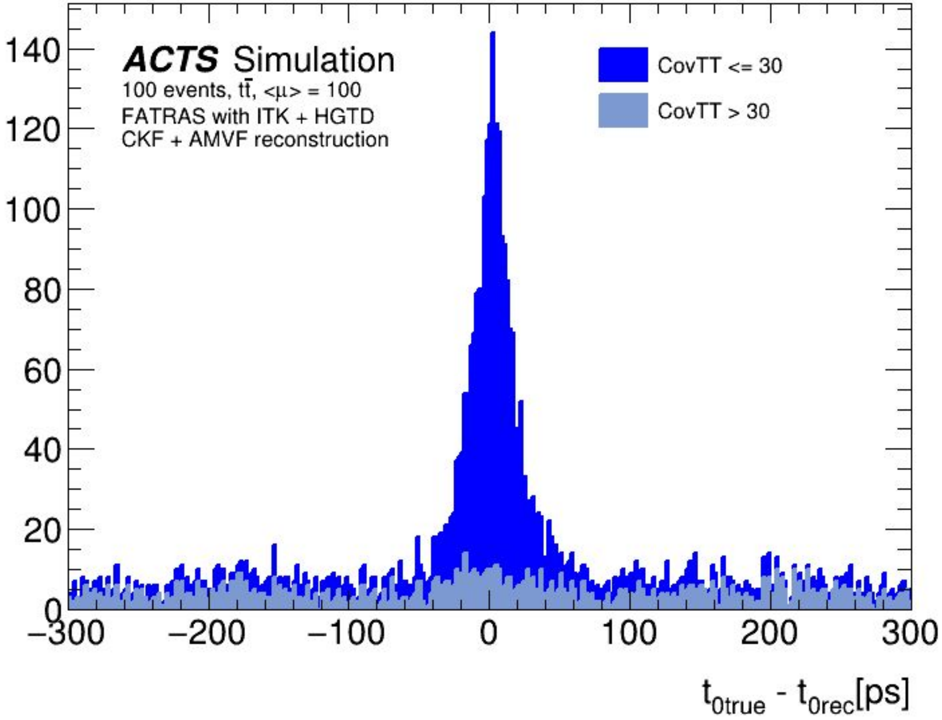
AMVF with time inclusion (mu = 0)

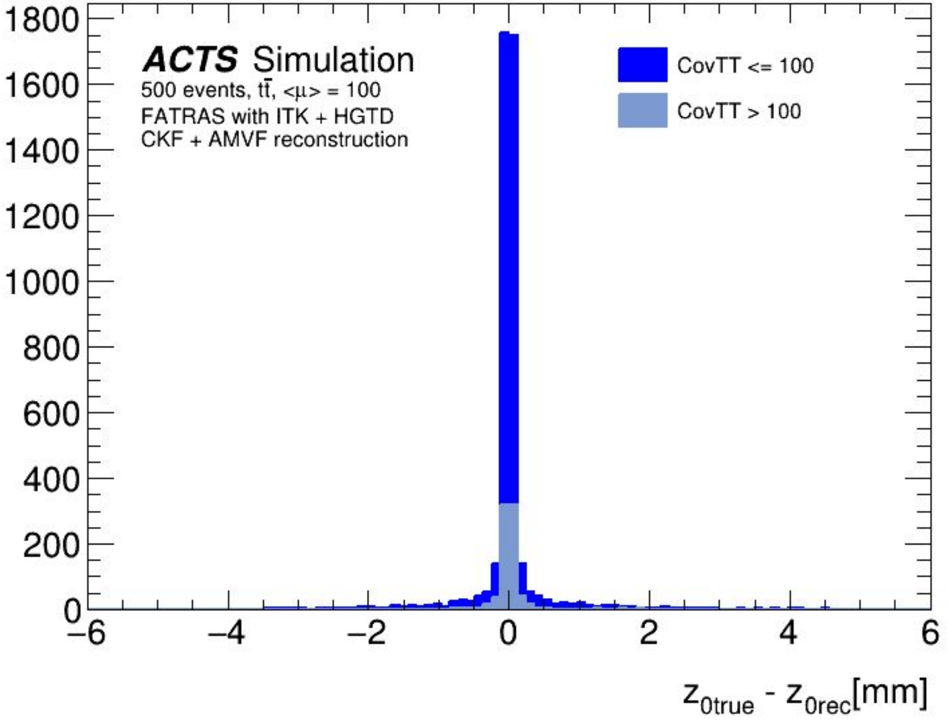




σ = 12 ps

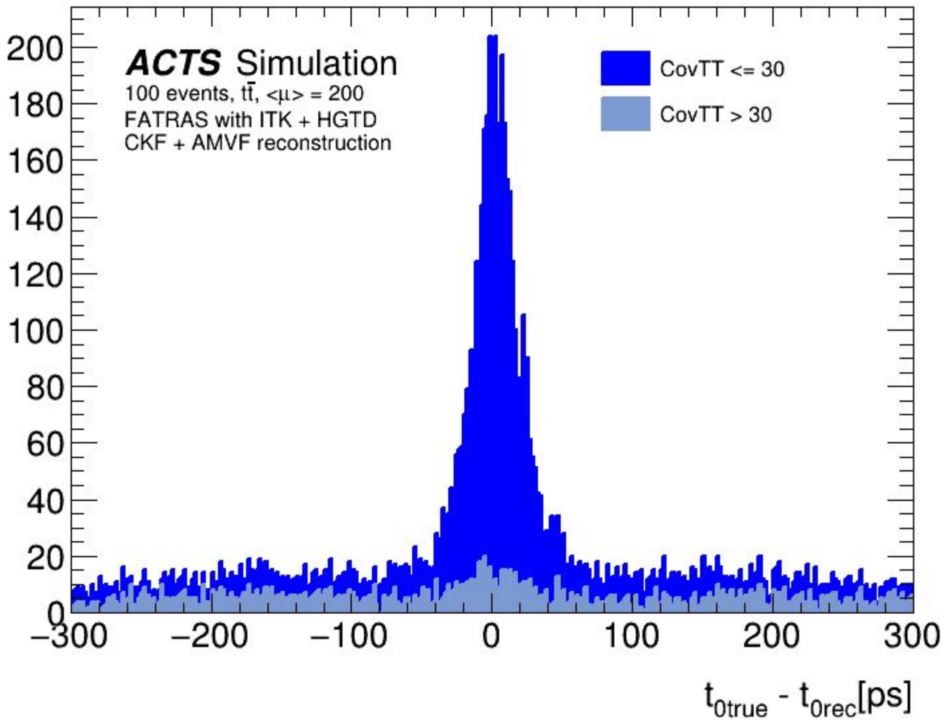
AMVF with time inclusion (mu = 100)

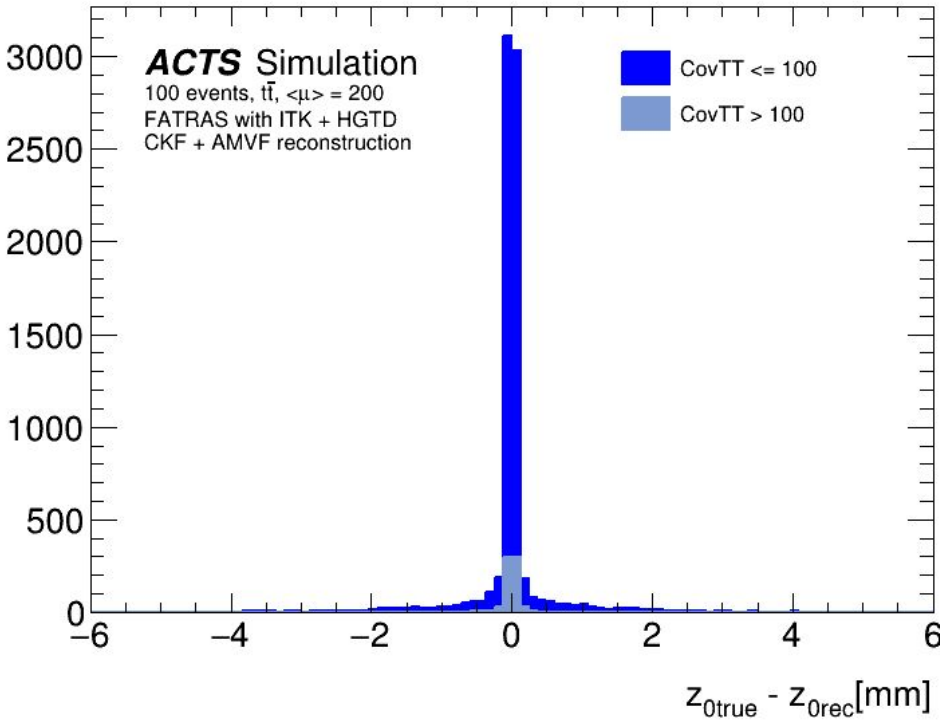




σ = 17 ps

AMVF with time inclusion (mu = 200)



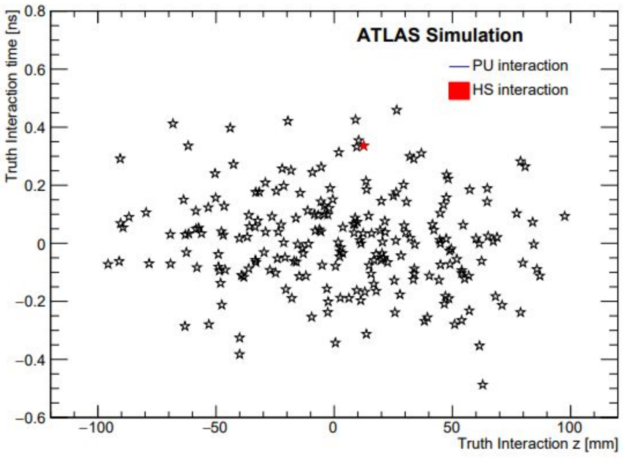


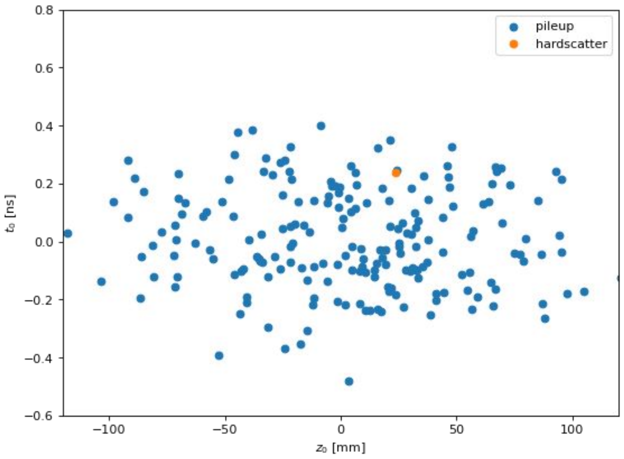
σ = 20 ps

Simulating HL-LHC beam with ACTS Pythia8

- Managed to simulate the same scenario as the TDR
    - z0 ~ N(0,50 mm), t0 ~ N(0,175ps)
- Simulated ttbar events with 1000 events
- Using Fatras
    - Got some propagation errors which needs to be addressed, but its not a blocking problem

```
addPythia8(
    s,
    hardProcess=["Top:qqbar2ttbar=on"],
    npileup=200,
    vtxGen=acts.examples.GaussianVertexGenerator(
        stddev=acts.Vector4(0.0125 * u.mm,
                            0.0125 * u.mm,
                            50 * u.mm,
                            0.175 * u.ns),
        mean=acts.Vector4(0, 0, 0, 0),
    ),
    rnd=rnd,
    outputDirRoot=outputDir,
)
```

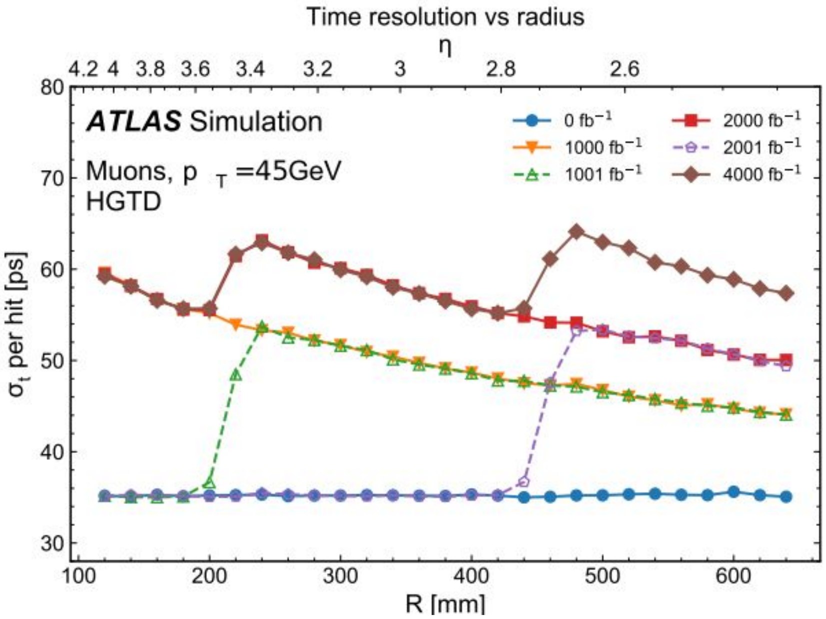


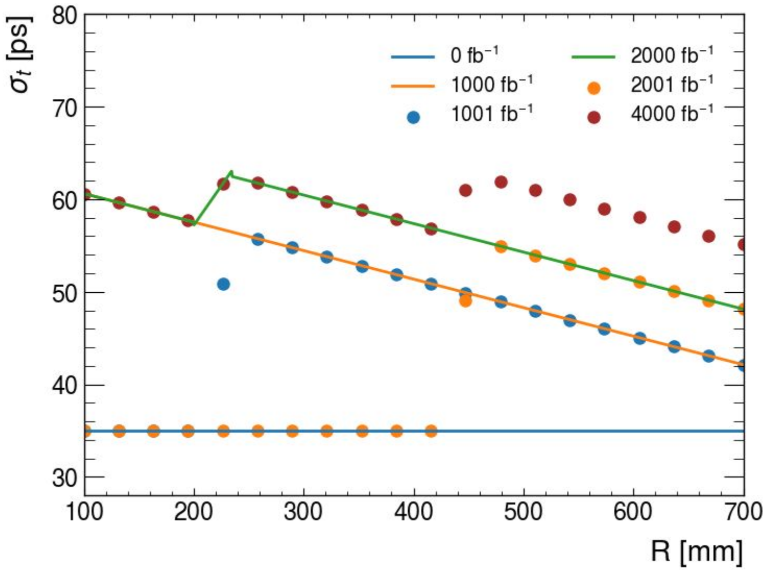


Evaluating impact of sensor degradation

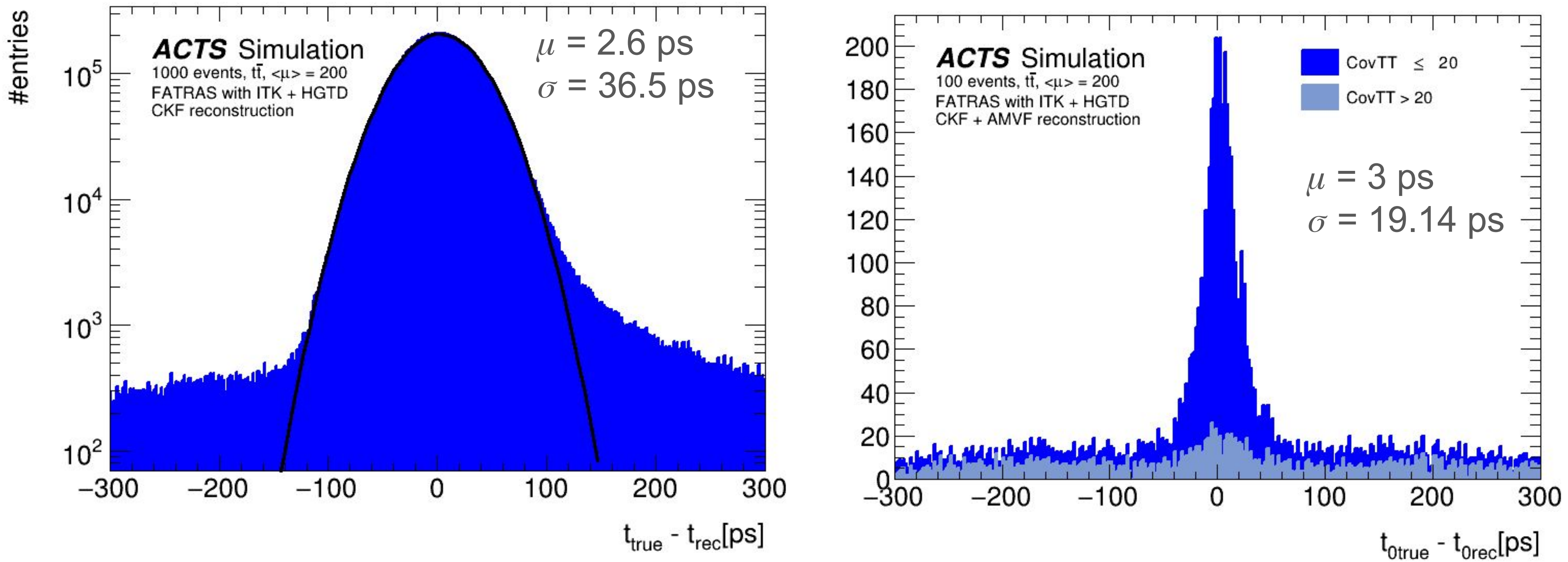
Simulation of sensor deterioration

- Simulated the sensor deterioration with increasing integrated luminosity
- The change in resolution was implemented in the digitization smearing as shown in the plots

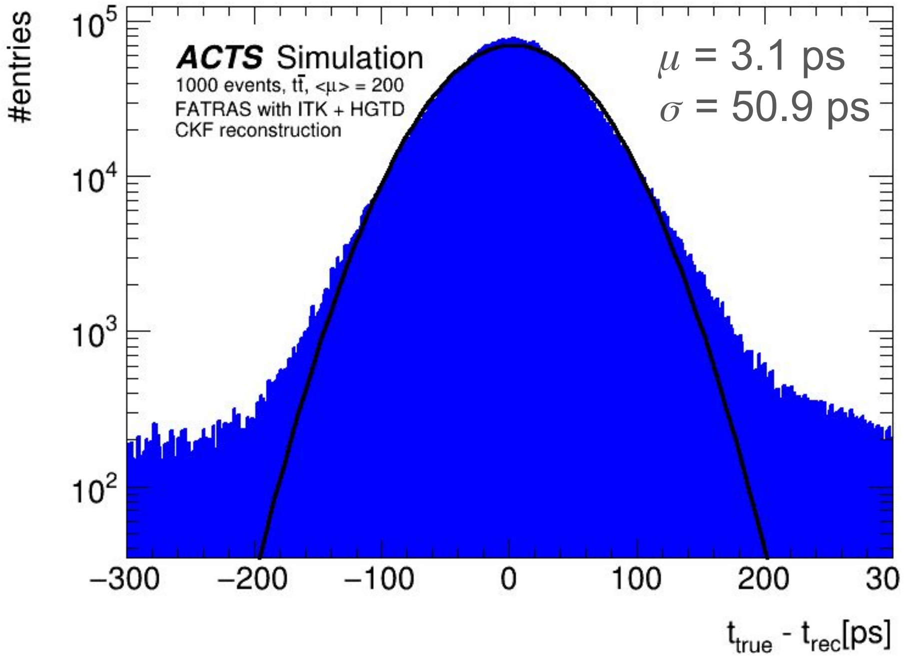


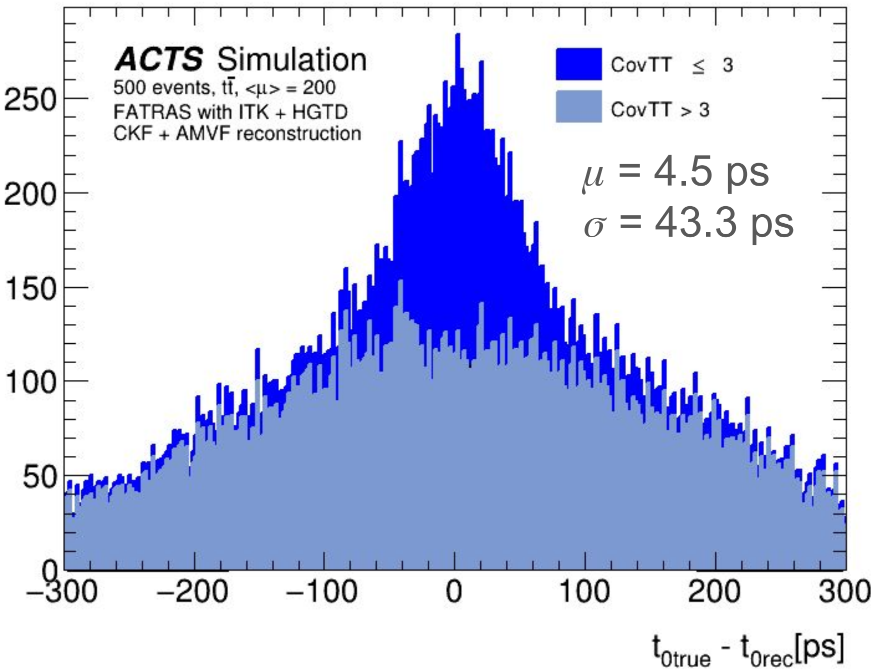


Reconstruction performance at 0 fb^{-1}

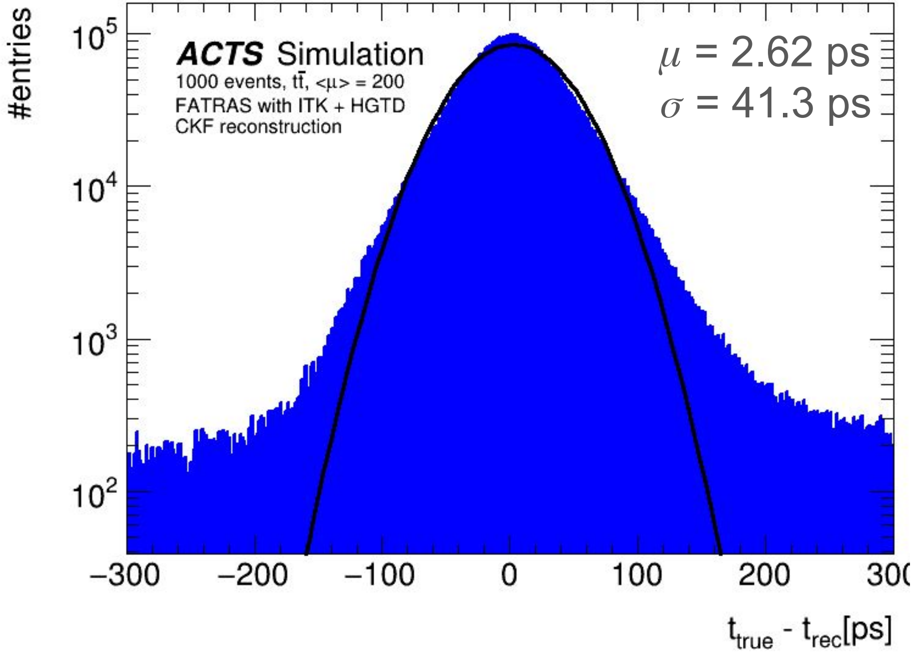


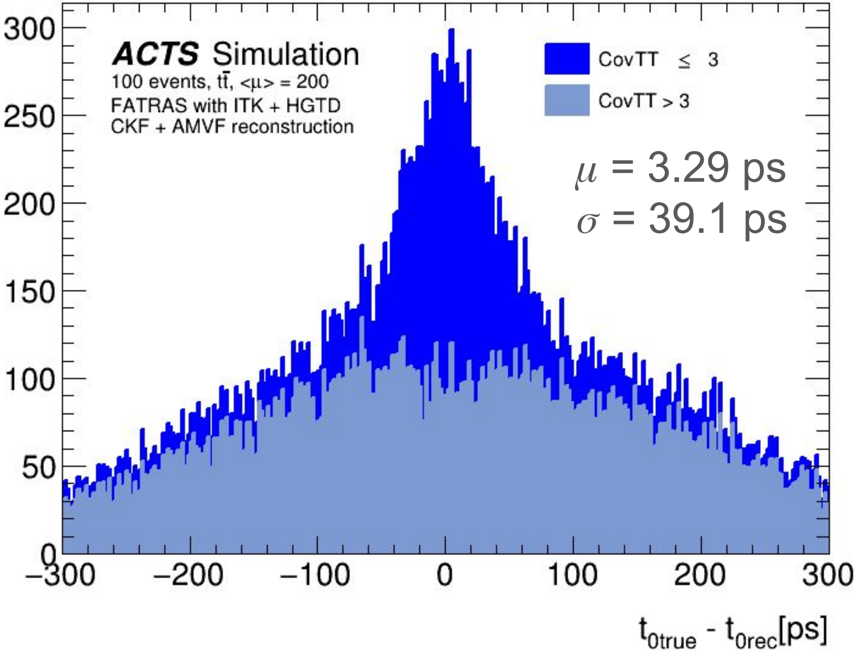
Reconstruction performance at 1000 fb^{-1}



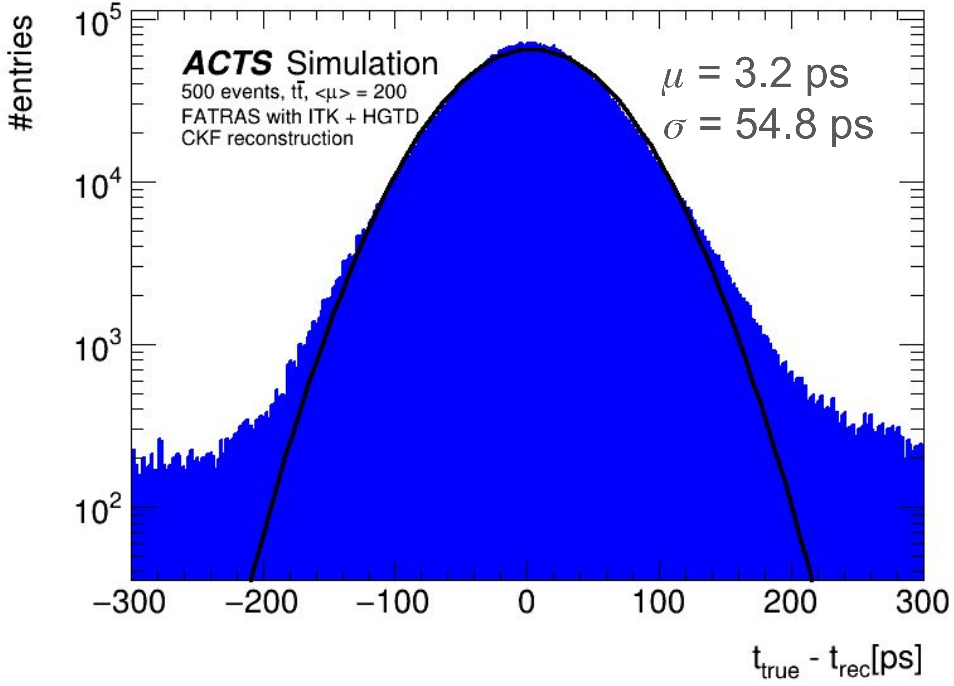


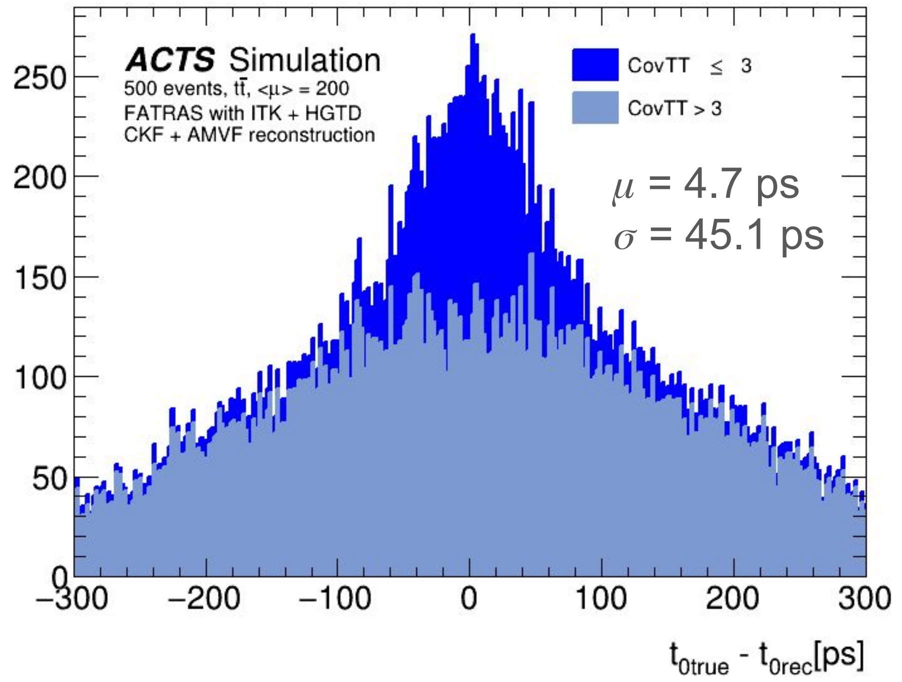
Reconstruction performance at 1001 fb^{-1}



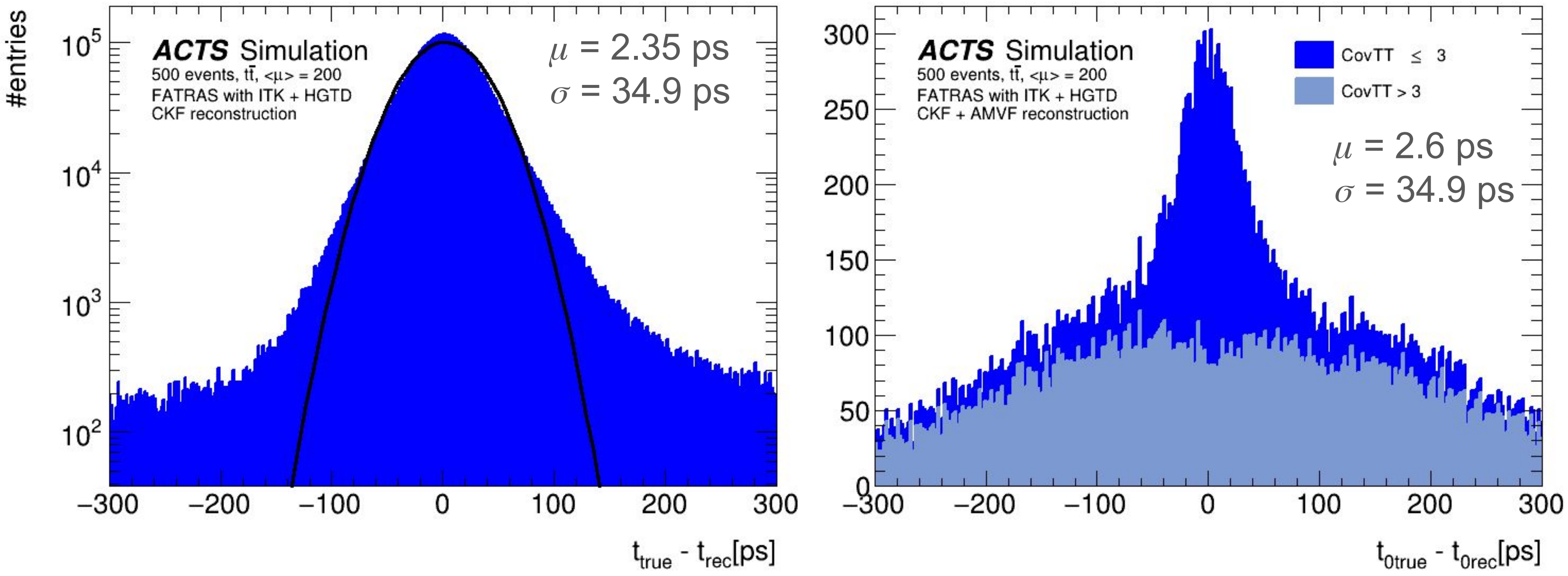


Reconstruction performance at 2000 fb^{-1}

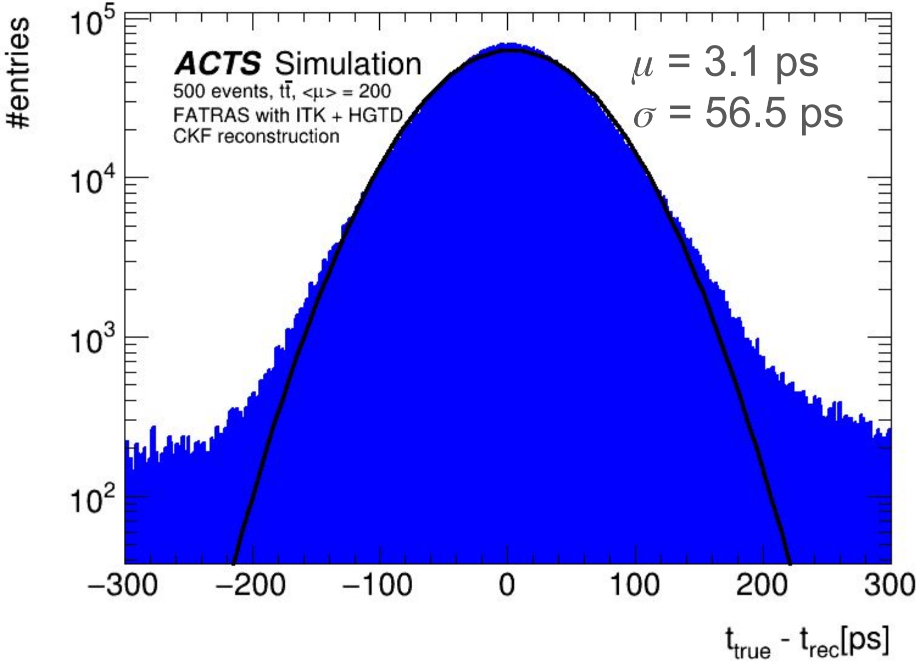


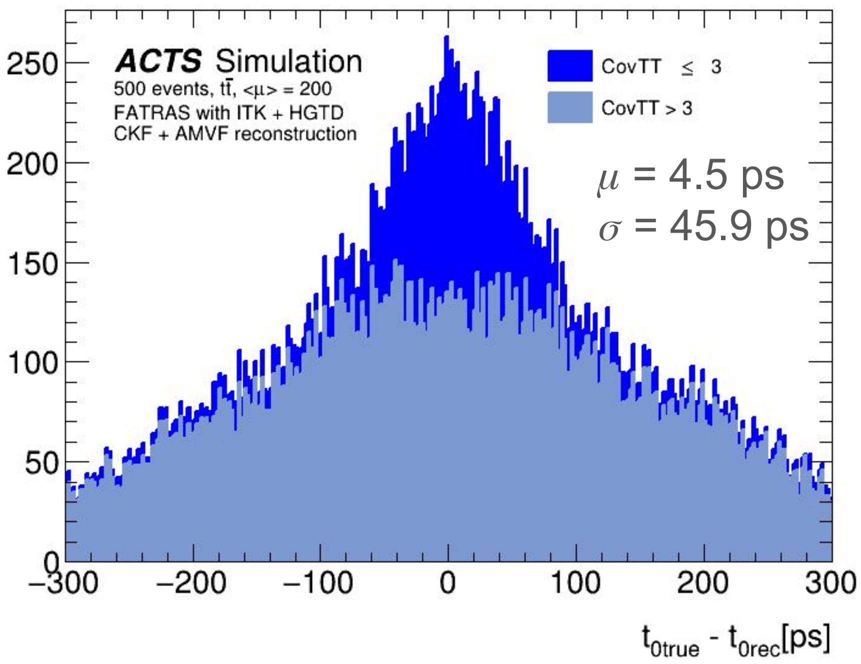


Reconstruction performance at 2001 fb^{-1}



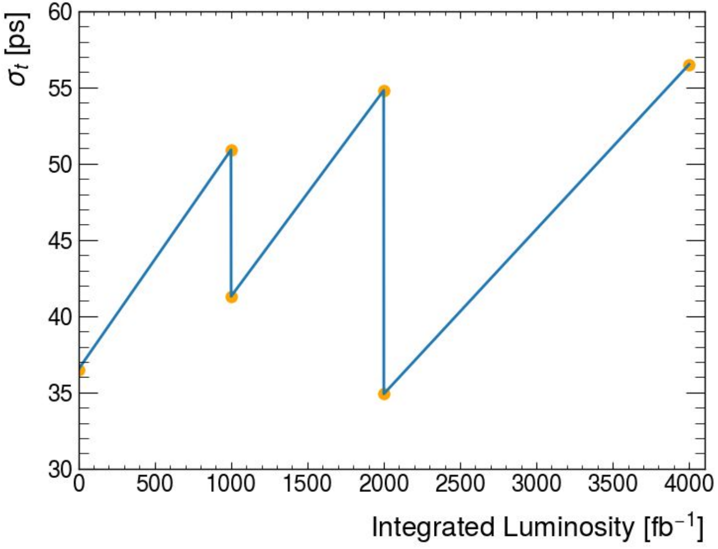
Reconstruction performance at 4000 fb^{-1}



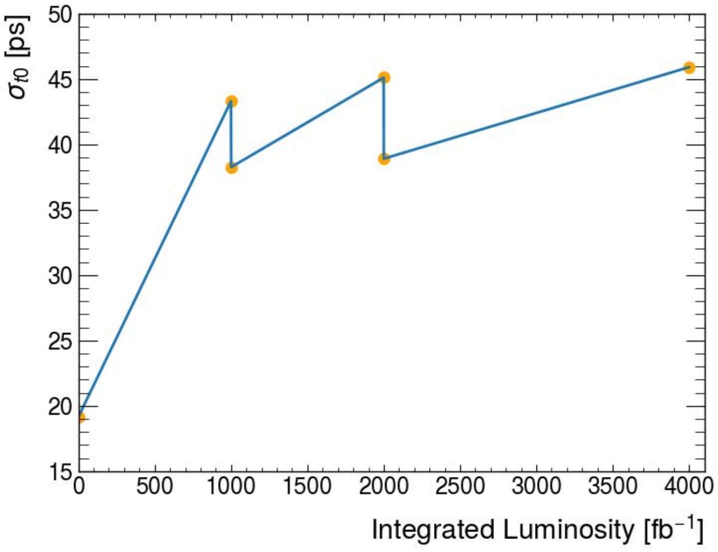


Time reconstruction x luminosity

- ATLAS is measuring performance in a slightly different way, so we dont have a metric to compare
- Errors when using the Gaussian fit to estimate the standard deviation can explain the “weird” results, but in general, it seems to make sense



hit time resolution



t0 resolution

Next steps

Explore ACTS (on going)

- Continue investigating the CKF until a full understanding
- Start investigate ExaTrk plugin to test ML based reconstruction methods
  - Get a general understanding of these methods but not to jump to it right away

ACTS Workshop

- Should we submit to the workshop?
  - Need to prepare an abstract for it in the next 5 days

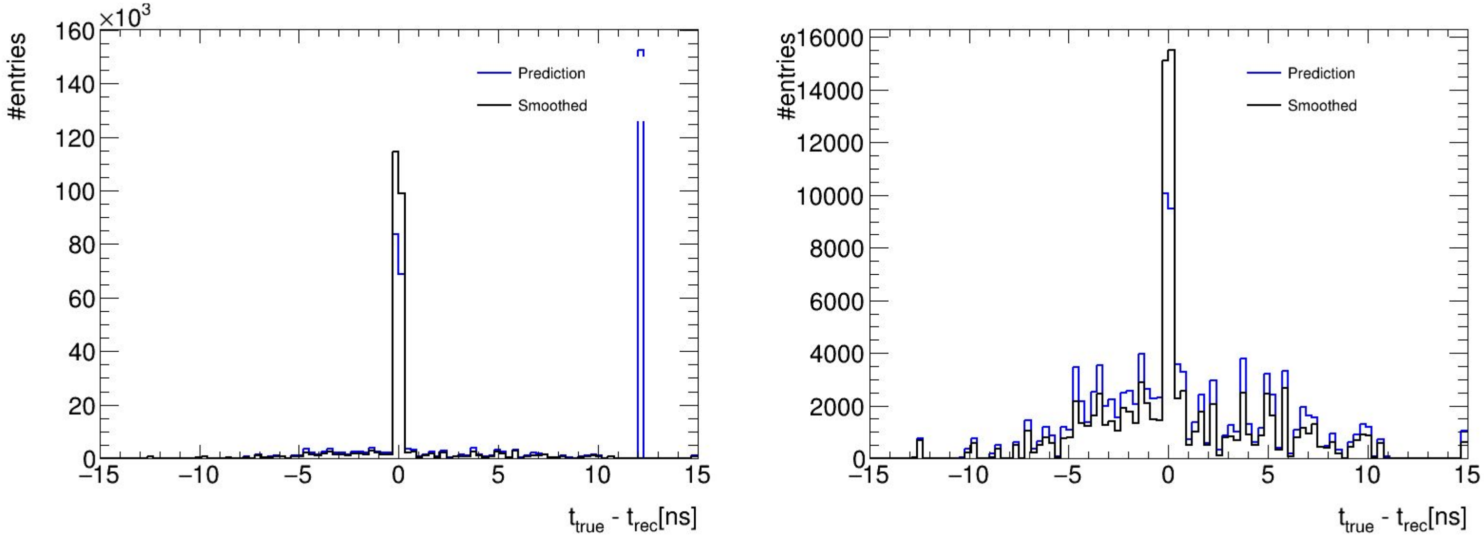
Theoretical Study

- H. Kolanosky, Particle Detectors (2020)
  - Next: Chapters 8-9
- Advance on the study notes
- Read papers of systematic review

Backup

Investigating weird residual bin at residual histogram

- In the previous meeting there was a weird residual bin for predicted samples
- This bin is composed of a singe value of ~12,25 ns repeated multiple times and all comes from event 96 (1 event out of 100)
- Filtering the event solves it out
    - Will investigate further later



Beam conditions at the HL-LHC

- HGTD TDR states (pag 5):

A major challenge for the ITk is the pileup suppression in the primary vertex and in the object reconstruction in this high pileup environment, especially in the end-cap region. The luminous region will have an estimated Gaussian spread of 30 to 60 mm along the beam axis (z direction1 .) The width in time could range from 175 to 260 ps. The case considered in this report is the “nominal” scenario, with Gaussian standard deviation of approximately 50 mm along the beam axis and spreads of 175 ps in time.

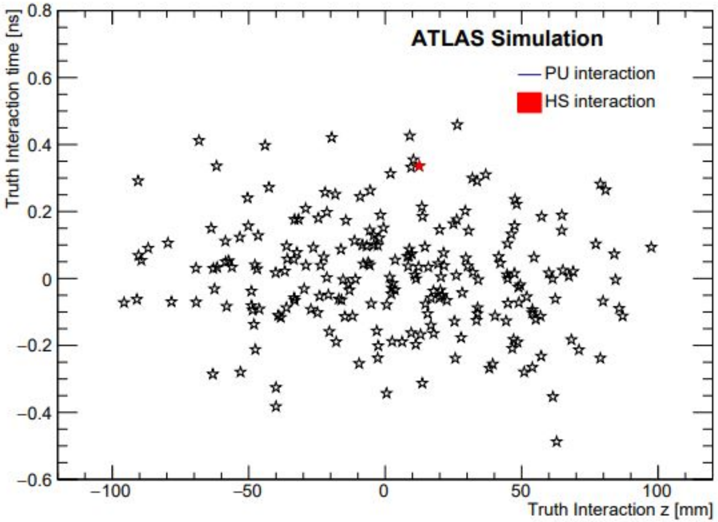
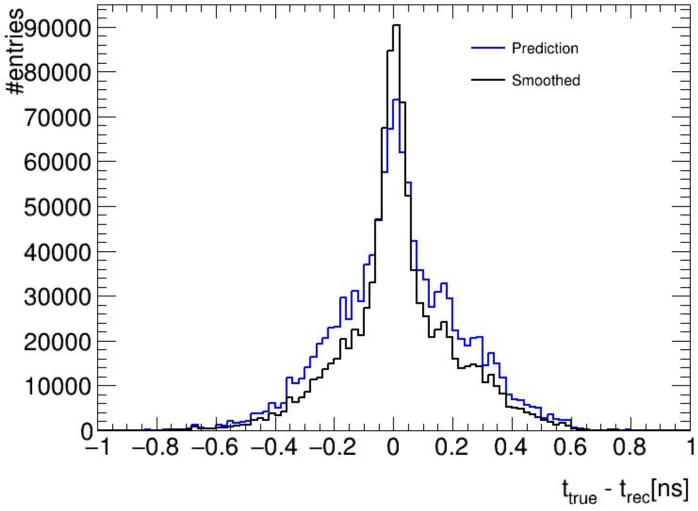


Figure: Visualisation of the truth interactions in a single bunch crossing in the z–t plane, showing the simulated Hard Scatter (HS) ttbar event interaction (red) with pileup interactions superimposed (black) for <μ> = 200

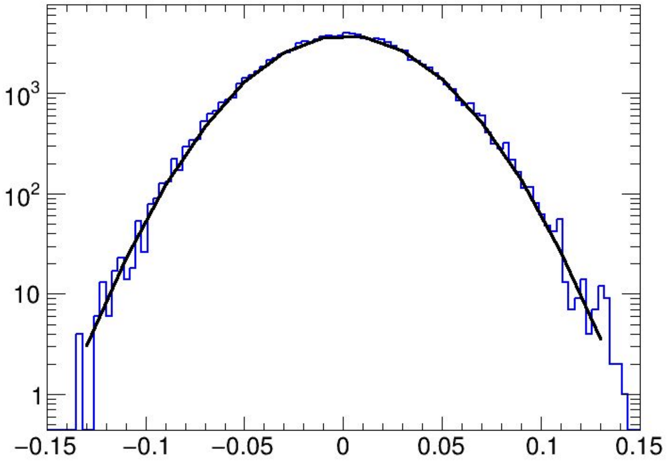
ttbar <u> = 0 residue plots

- Needed to filter 3 events, for the same reason as slide 6
- Residue spams from -0.8 to 0.8 ns
  - This is the whole range of our vertex generation, so it's no much of a positive result
- HGTD volumes presents the resolution expected by specification (35 ps)

whole detector



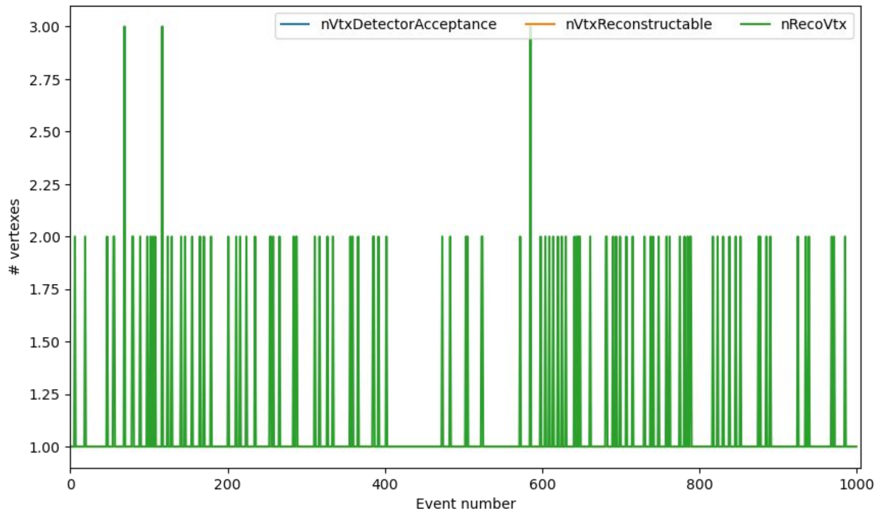
HGTD volumes



smoothed fitted (σ = 35 ps)

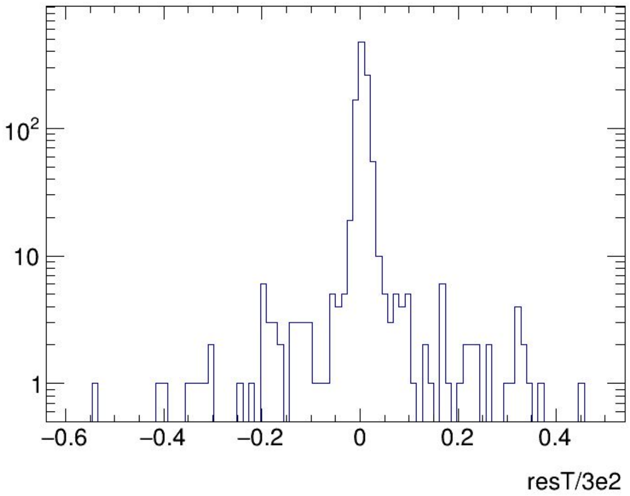
ttbar <u> = 0 vertex reconstruction efficience

- Without pileup there’s only one vertex to be reconstructed
- Sometimes the reconstruction splits the vertexes and reconstructs 2 or 3
  - Need further analysis to see if the recovered vertexes are close in space and time

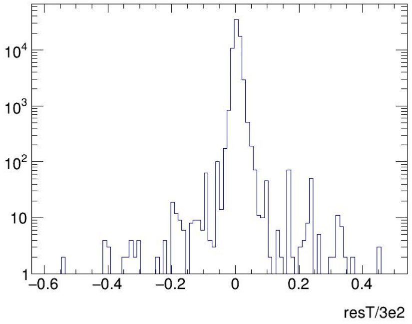


ttbar <u> = 0 vertex reconstruction residue

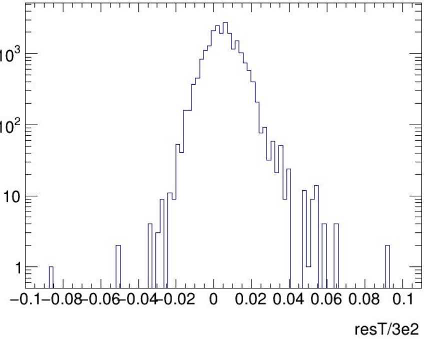
- t0 resolution is slightly better than whole time estimation, spams from -0.6 to 0.4
- If we isolate barrel region tracks ($|\eta| < 2.4$) from endcap region ($|\eta| > 2.4$) we get different distributions
  - Endcap region has better resolution, indicating that HGTD tagging is improving the reconstruction



whole detector



barrel ($|\eta| < 2.4$)

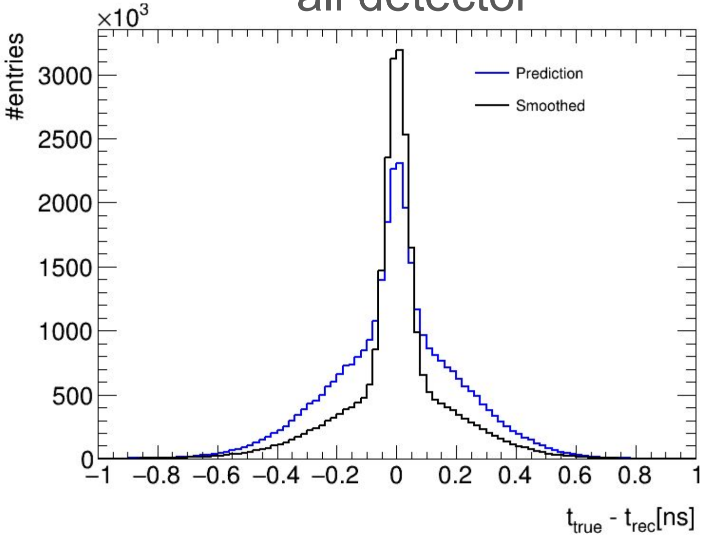


endcaps ($|\eta| > 2.4$)

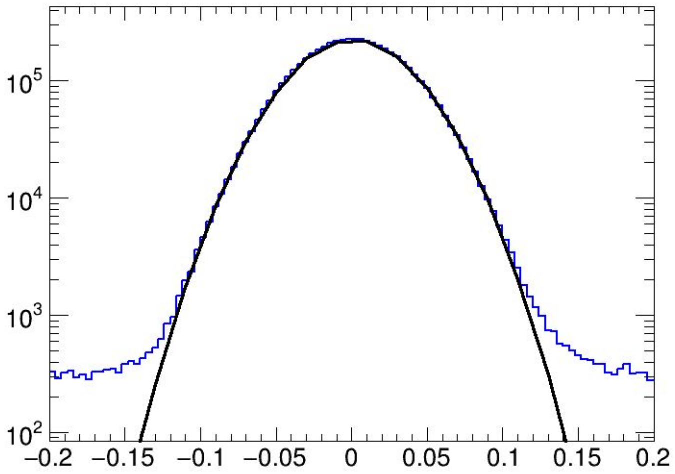
ttbar <u> = 200 residue plots

- Better separation between smoothed and predicted samples
  - Indicating the smoothing is important to mitigate pileup
- Same resolution as the simulation without pileup
- HGTD volumes show expected resolution but with pronounced tails

all detector



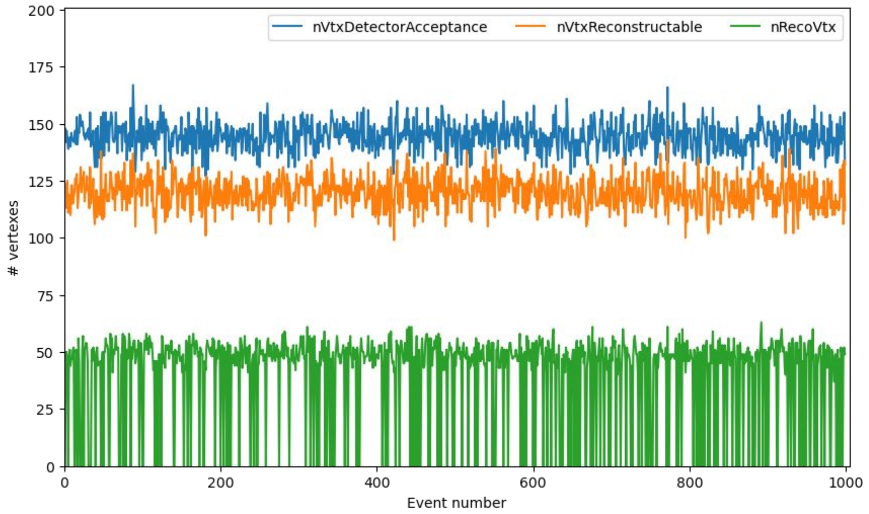
HGTD volumes

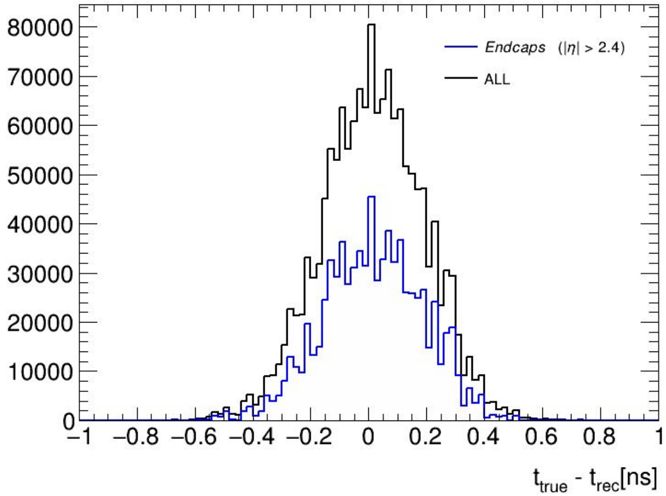


smoothed fitted (σ = 35 ps)

ttbar <u> = 200 vertex reconstruction

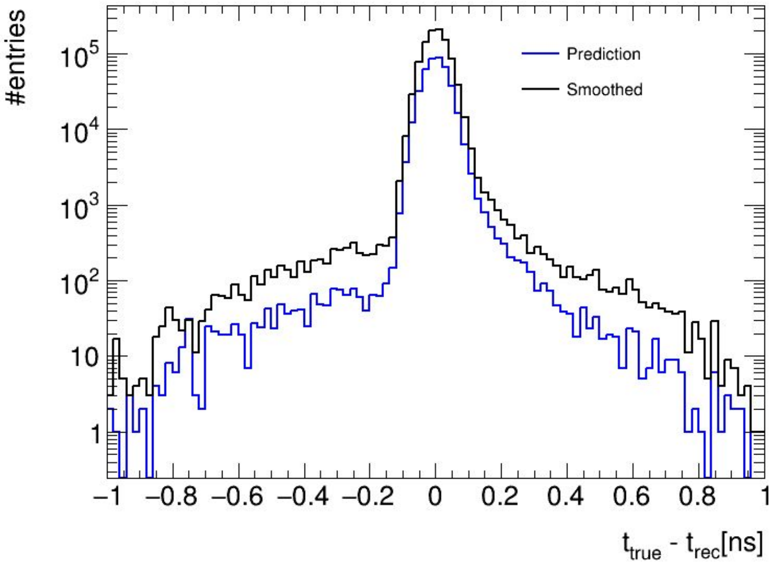
- Not all of the 201 vertex were reconstructed
  - Limited by detector geometry and preselection parameters on the reconstruction
- t0 resolution spams from -0.6 to 0.6
- No significant difference from endcaps to barrel region
  - Probably due to pileup tracks not tagged by HGTD

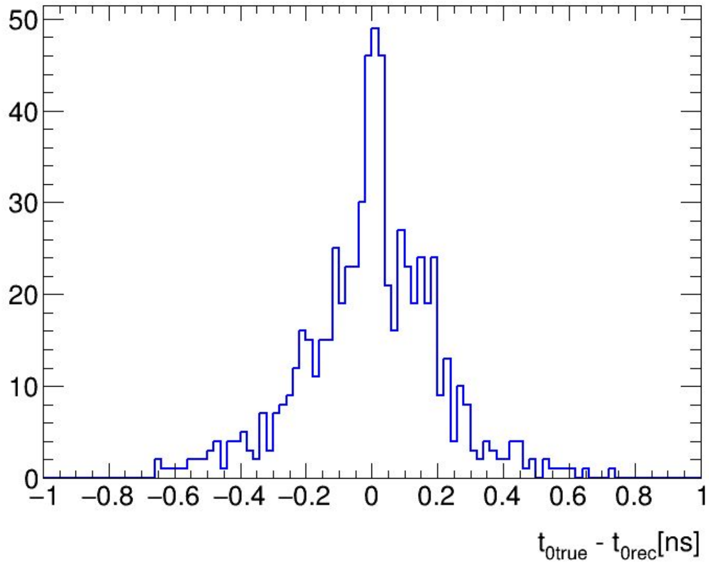




ttbar <u> = 200 for HGTD tagged tracks

- Made a experiment by putting a high value for initial time predictions
- This way only tracks with HGTD hits will be centered at zero
- Tried now to evaluate the performance of this subset, but it was not much better than before

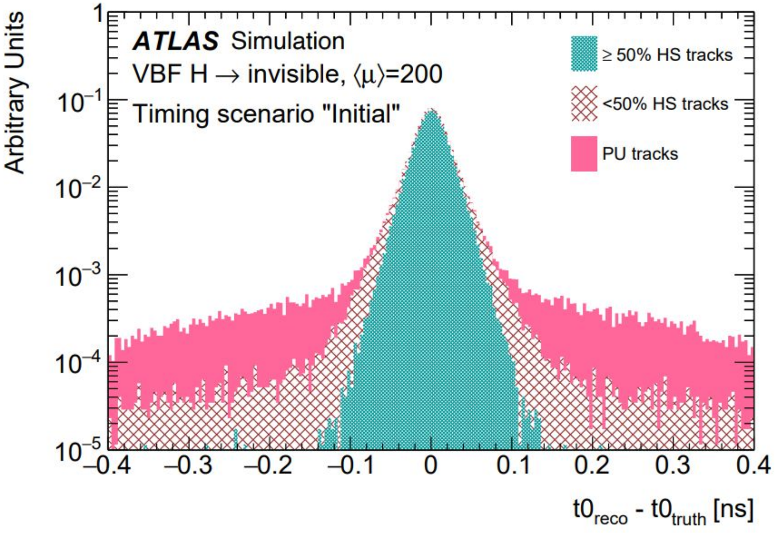




Importance of primary vertex time (t0) determination

- HGTD TDR states that (pag 37):

Due to the large uncertainty of the longitudinal impact parameter for tracks in the forward region (Figure 2.6), the association of tracks to nearby vertices purely based on spatial information is ambiguous in high-pileup environments, especially for low transverse momentum tracks. The ability to determine the time of the primary vertex of the hard-scatter process, here denoted as t0, provides a new handle to enhance the capability of the ATLAS detector to remove pileup tracks contaminating physics objects originating from the hard-scatter vertex.



Vertex t0 resolution separately for various cases, where "HS" ("PU") stands for hard scatter (pileup). **Only track clusters tagged by HGTD were evaluated.**