HL-LHC ATLAS 4D tracking
ACTS CKF timing reconsustruction

Rodrigo Estevam de Paula

Marco Aurelio Lisboa Leite

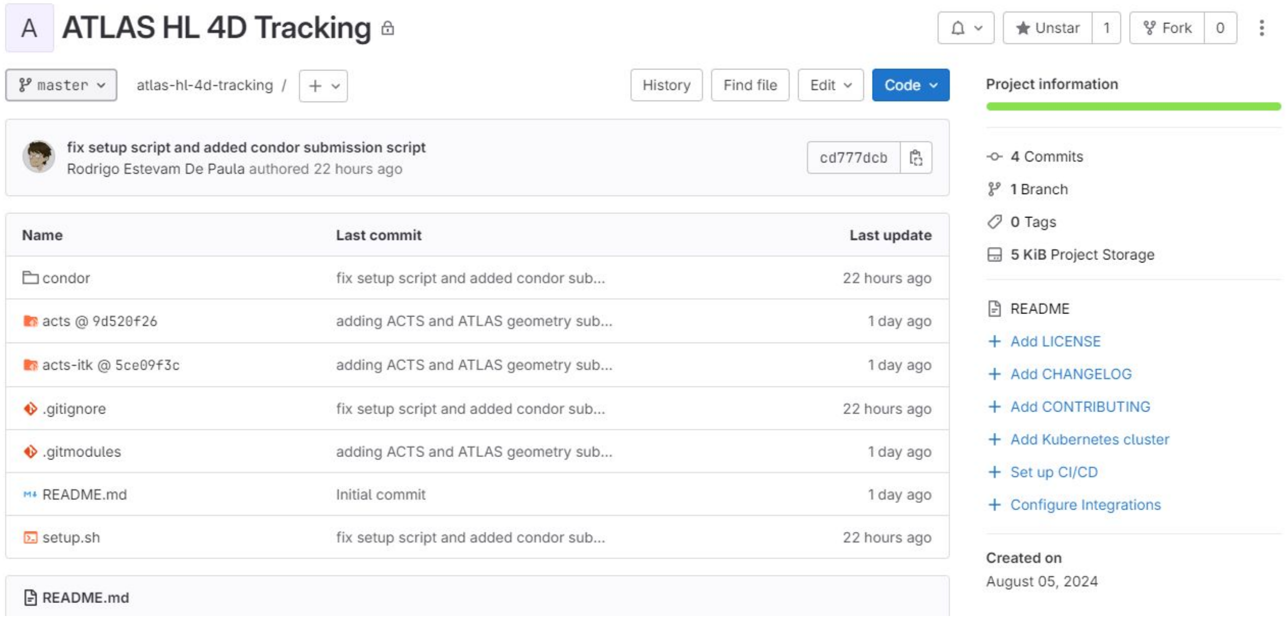
Vitor Heloiz Nascimento

21 August, 2024

Quick Updates

Git Repository

- https://gitlab.cern.ch/USP/hgtd/atlas-hl-4d-tracking
- Contains ACTS and ACTS-ITK submodules for the framework and geometry respectively
- Will contain our reconstruction proposals and performance evaluation scripts



fix setup script and added condor submission script
Rodrigo Estevam De Paula authored 22 hours ago
cd777dcb

Name	Last commit	Last update
condor	fix setup script and added condor sub...	22 hours ago
acts @ 9d520f26	adding ACTS and ATLAS geometry sub...	1 day ago
acts-itk @ 5ce09f3c	adding ACTS and ATLAS geometry sub...	1 day ago
.gitignore	fix setup script and added condor sub...	22 hours ago
.gitmodules	adding ACTS and ATLAS geometry sub...	1 day ago
README.md	Initial commit	1 day ago
setup.sh	fix setup script and added condor sub...	22 hours ago

README.md

Project information
4 Commits
1 Branch
0 Tags
5 KiB Project Storage
README
Add LICENSE
Add CHANGELOG
Add CONTRIBUTING
Add Kubernetes cluster
Set up CI/CD
Configure Integrations
Created on
August 05, 2024

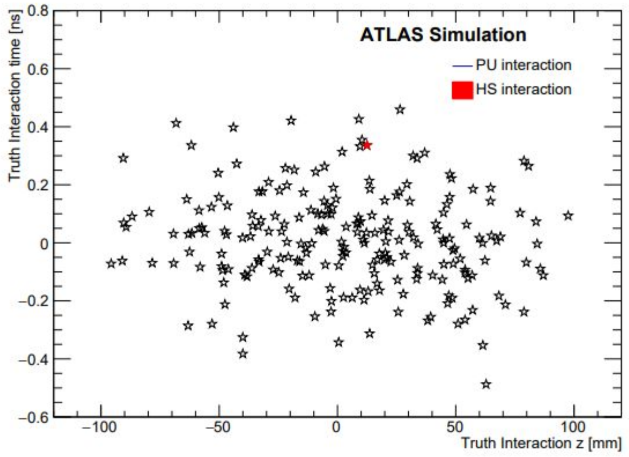
Additional activities

- Sampa cluster updated to ALMA9, so I updated ACTS to release 35.2 (36.0 is broken)
- Learned how to submit ACTS jobs to condor
  - Allows for bigger and more complex simulations
  - Possible to tweak machine specs (e.g. processor threads, allocated RAM, etc)
- Taking part on the ATLAS software training
- Theoretical Studies
  - Reading Alexander thesis on HGTD performance studies (link)
    - Chapter 12 presents time reconstruction performance and t0 vertex reconstruction
  - Pattern Recognition, Tracking and Vertex Reconstruction in Particle Detectors book

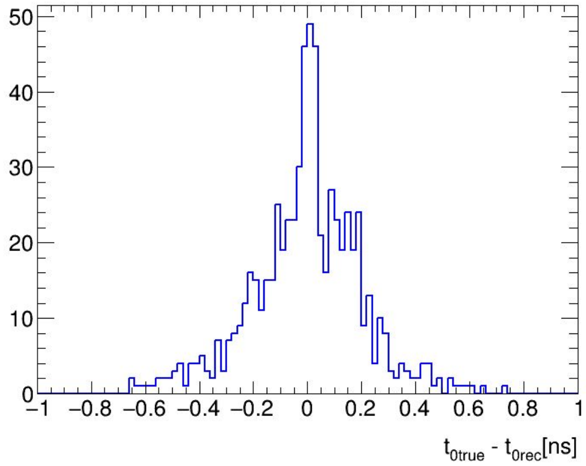
ACTS Time reconstruction

Recap of the previous meeting

- We want to investigate the resolution in which we can reconstruct the time (t0) of the primary vertexes
  - If the resolution is good enough, it can be used to isolate HS from pileup interactions
  - Intend to validate the usage of HGTD for this purpose



Simulation of primary vertexes



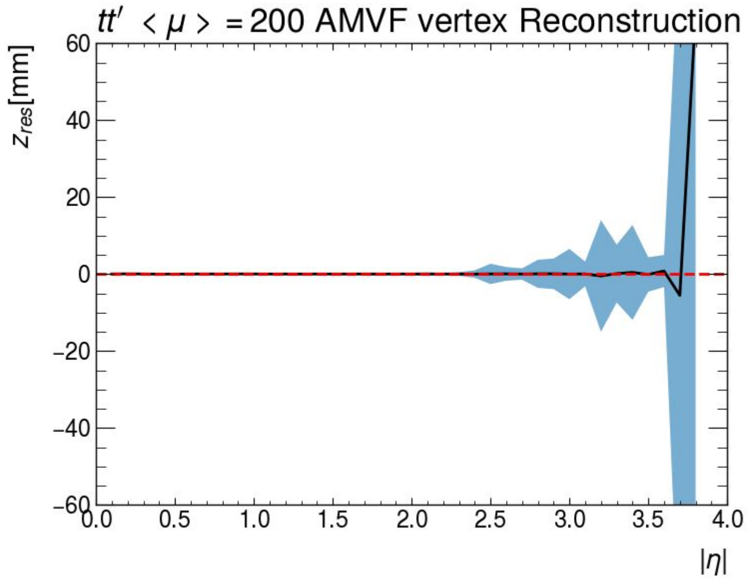
residual plot of reconstructed t0 for ttbar events with <mu> = 200

Evaluating primary vertex resolution

- It is expected worse resolution of z0 for vertexes composed of tracks with high η (go to the endcap regions)
- If, for these cases, t0 has a good resolution, it can be used to complement this measurement and help in pileup rejection

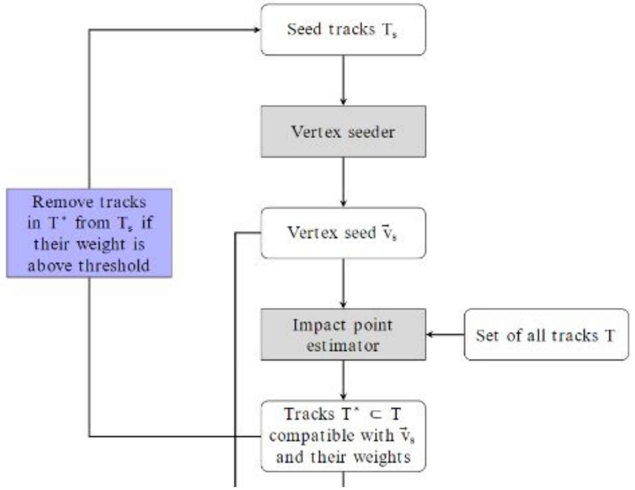
Two main vertex reconstruction methods:

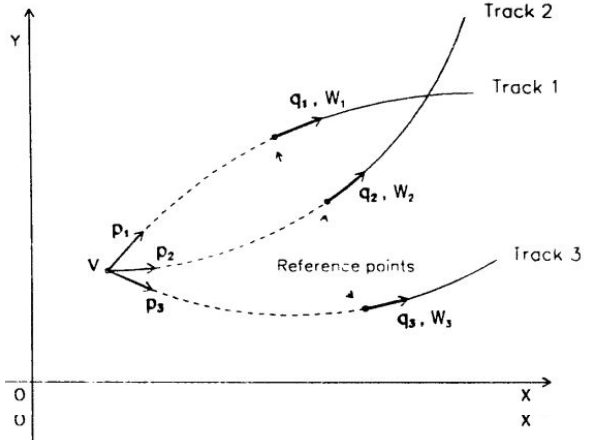
- Iterative Vertex Fitter (IVF)
  - Used for results showed in the previous meeting
- Adaptive Multi Vertex Fitter (AMVF)
  - More robust method adopted by ATLAS



Iterative Vertex Fitter (IVF)

- Chose high pt tracks to serve as seeder for vertexes
-
- Fitter Based on the Full Billoir Vertex Fitter:
  - Fast vertex fitting with a local parametrization of tracks
- The main goal is to find **V** and **pi** that minimize the **X²**


Seed tracks T_s
Vertex seeder
Remove tracks in T^* from T_s if their weight is above threshold
Vertex seed $\vec{v}_s$
Impact point estimator
Set of all tracks T
Tracks $T^* \subset T$ compatible with $\vec{v}_s$ and their weights



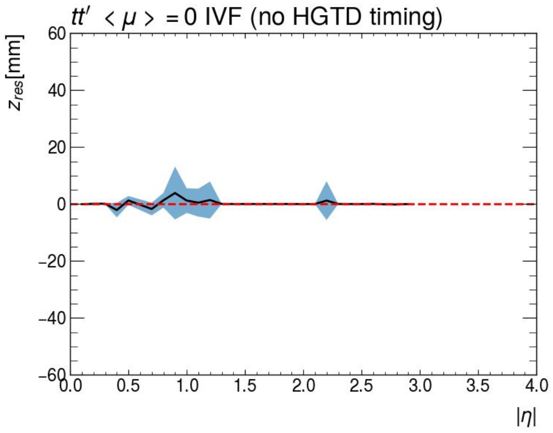
$$\chi^2 = \sum_i \Delta q_i^T W_i \, \Delta q_i,$$

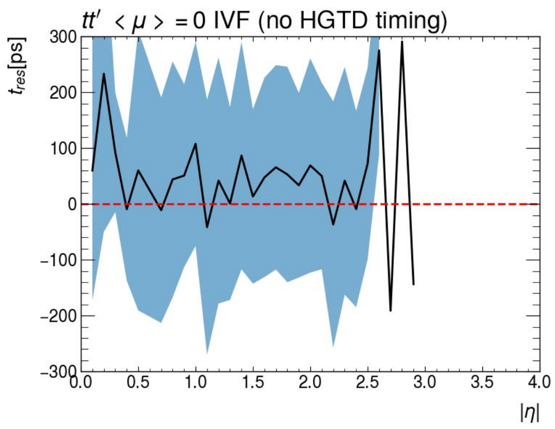
where

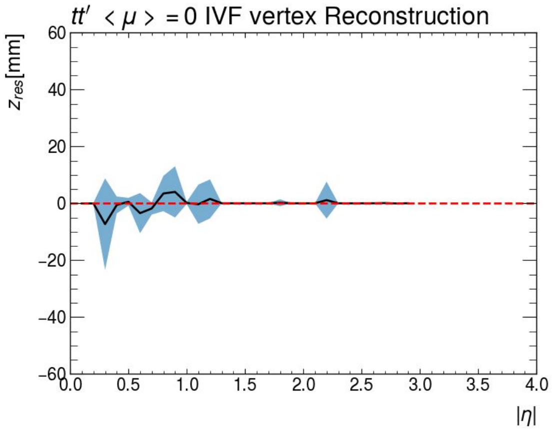
$$\Delta q_i = q_i^{\text{measured}} - F(V, p_i).$$

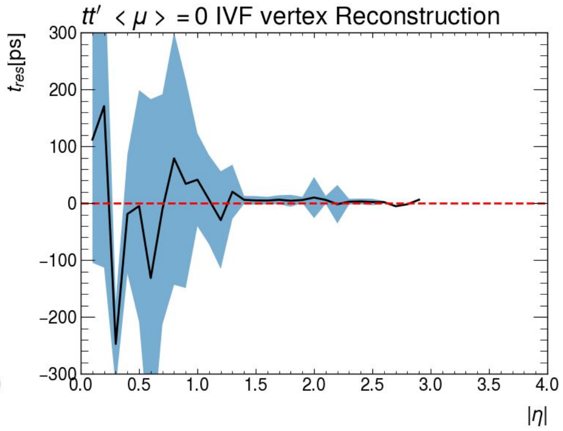
Results of IVF without pileup

- In general, worse resolution for reconstruction of z0
  - Compared to AMVF
- Inclusion of HGTD timing measurement increases significantly the resolution for |η| > 2.5



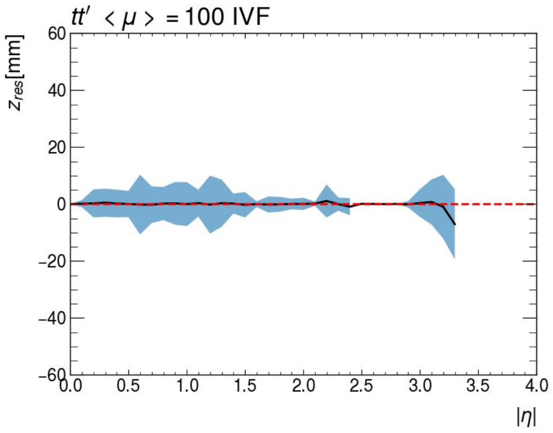


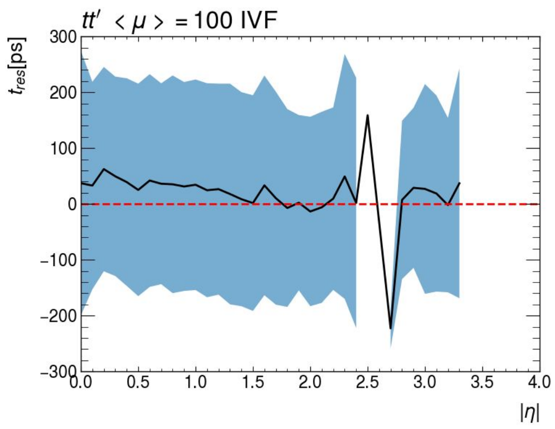


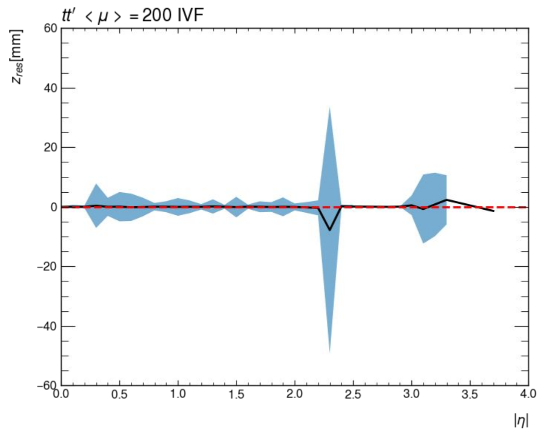


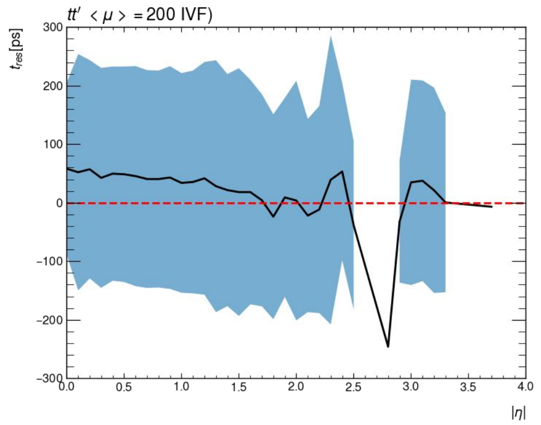
Results of IVF with pileup

- Resolution of z0 worsens with increasing pileup
- Inclusion of HGTD timing measurement doesn't impact the resolution for events with pileup









Adaptive Multi Vertex Fitting (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

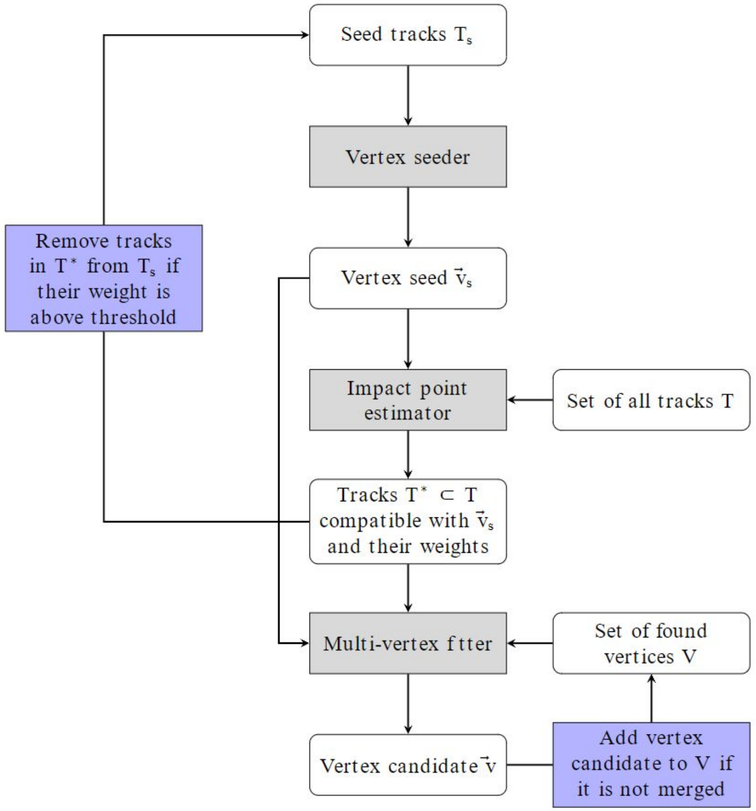
The weight of vertex j to track i is:

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

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

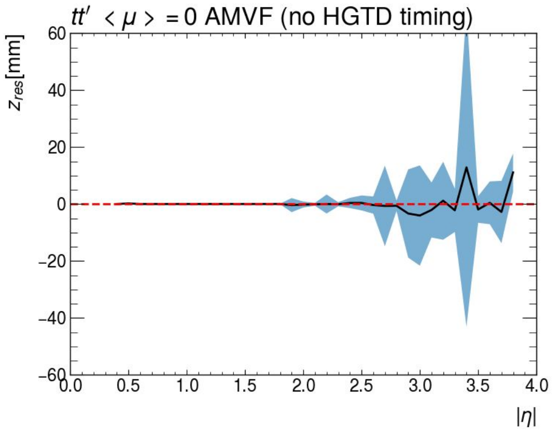
Adaptive Multi Vertex Fitting (AMVF)

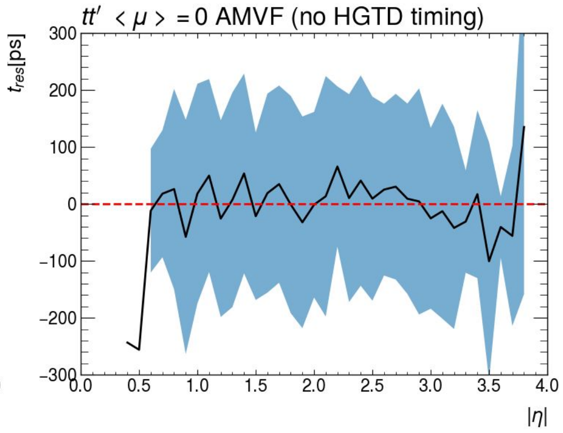
- ACTS implementation of AMVF also uses it to iterate over the vertex seeds
- The diagram aside shows how the integration works

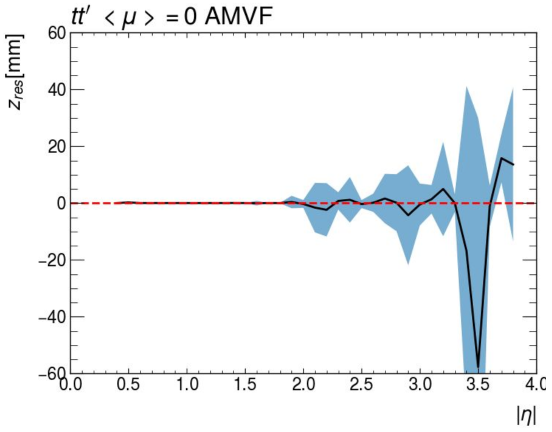

Seed tracks T_s
Vertex seeder
Remove tracks in T^* from T_s if their weight is above threshold
Vertex seed $\vec{v}_s$
Impact point estimator
Set of all tracks T
Tracks $T^* \subset T$ compatible with $\vec{v}_s$ and their weights
Multi-vertex ftter
Set of found vertices V
Vertex candidate $\vec{v}$
Add vertex candidate to V if it is not merged

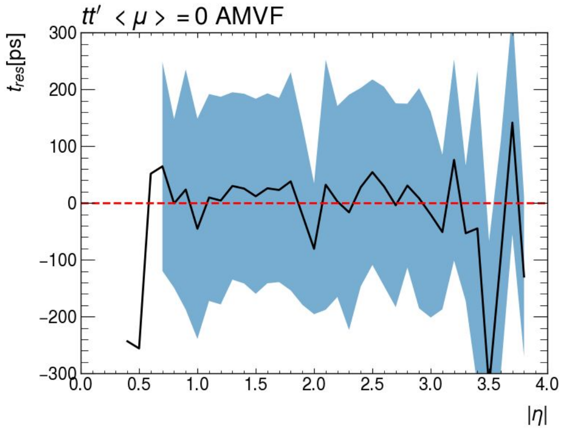
Results of AMVF without pileup

- In general, worse resolution for reconstruction of z0
  - Compared to AMVF
- Inclusion of HGTD timing measurement increases significantly the resolution for |η| > 2.5



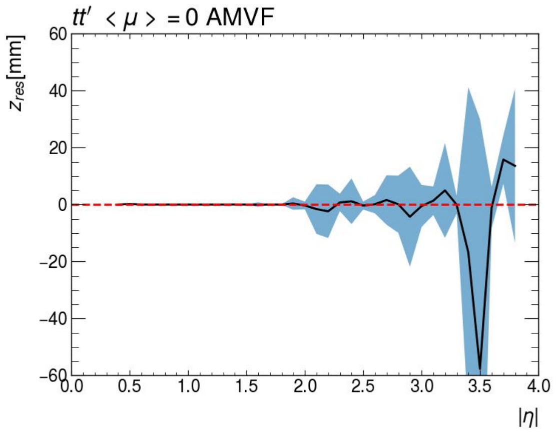


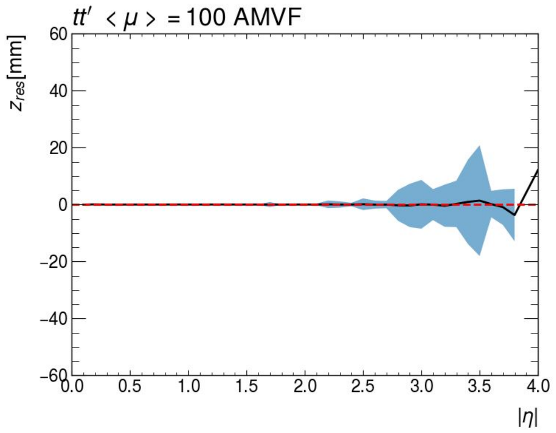


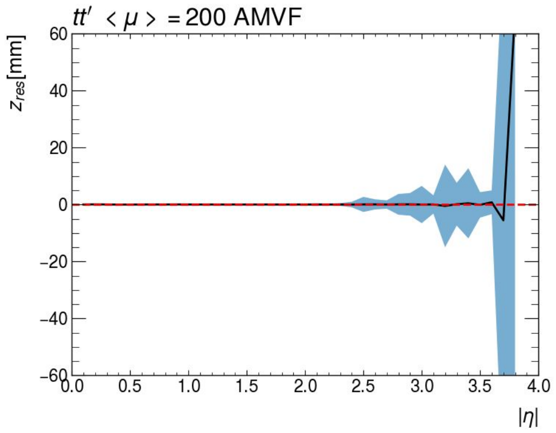


Results of AMVF with pileup

- As HGTD timing has no impact in the reconstruction, presenting here only z0 plots
- The increase is pileup doesn't have a significant impact on the z0 resolution (great!)







Next steps on vertex reconstruction

- Incorporate time coordinate in the AMVF
  - AMVF seems more promising than IVF
  - May take some time until it works…
- If we get any significant result from it, plan to present at the ACTS workshop
  - Deadline to submit the abstract is September 10th
  - If not, document the process to establish our baseline, so no work lost
- Complete the analysis with other metrics
  - Tracking and vertexing efficiency

Midterm next steps

Explore ACTS (on going)

- Continue investigating the CKF until a full understanding
- Study the implementation of the GSF in the core library
- 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

Follow HGTD ACTS integration campaign

- Start AQP

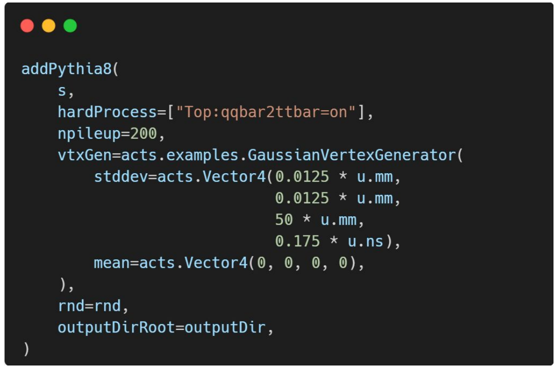
Theoretical Study

- H. Kolanosky, Particle Detectors (2020)
  - Next: Chapters 8-9
- Advance on the study notes
- Read papers of systematic review
  - Search more papers on high dimensional combinatorial optimization problems

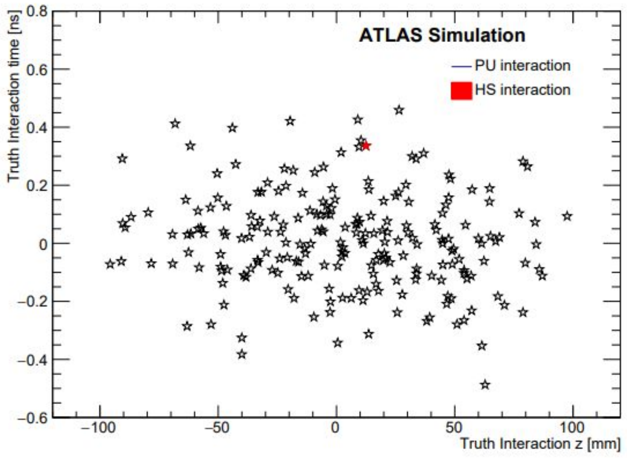
Backup

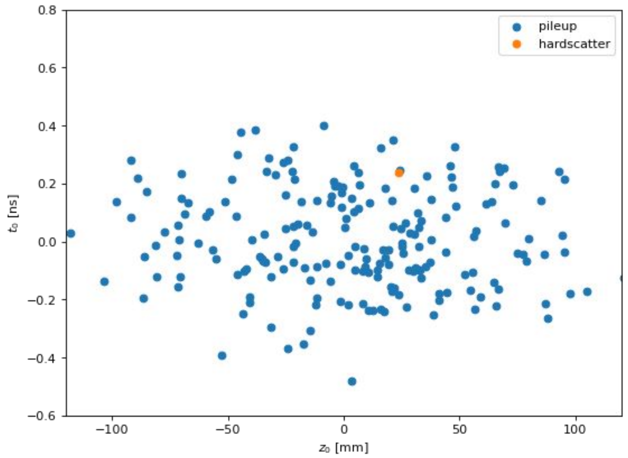
Simulating HL-LHC beam with ACTS Pythia8

- 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



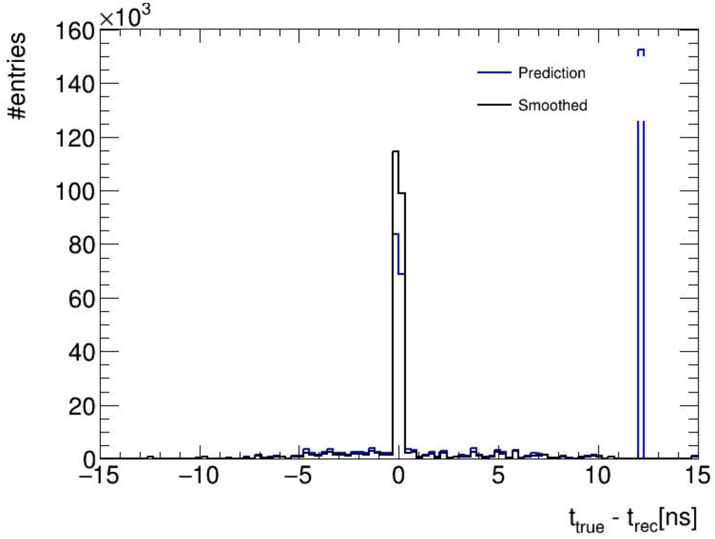
```
addPythia8(
    s,
    hardProcess=["Top:qqbar2ttbar=on"],
    npileup=200,
    vtxGen=acts.examples.GaussianVertexGenerator(
        stddev=acts.Vector4(0.0125 * u.mm,
                            0.0125 * u.mm,
                            50 * u.mm,
                            0.175 * u.ns),
  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

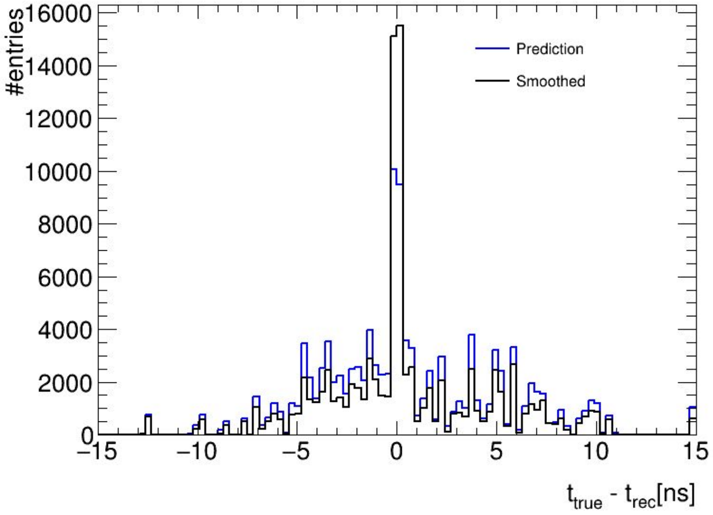




Investigating weird residual bin at residual histogram

- 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.

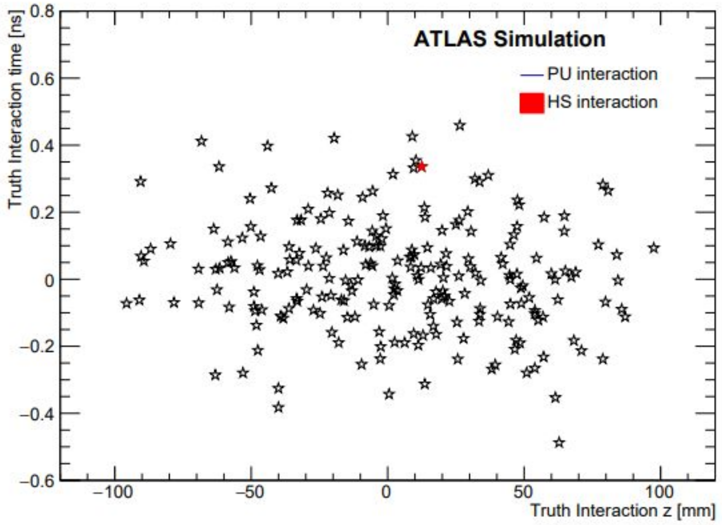
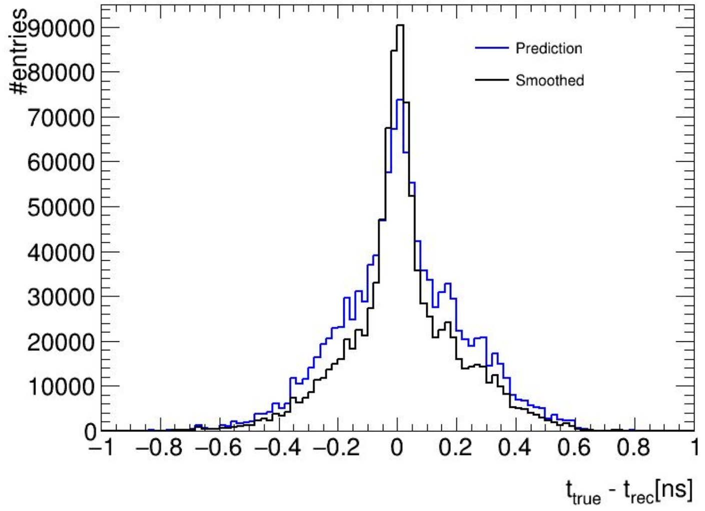


Figure: Visualisation of the truth interactions in a single bunch crossing in the z–t plane, showing the simulated Hard Scatter (HS) ttbar event interaction (red) with pileup interactions superimposed (black) for <μ> = 200

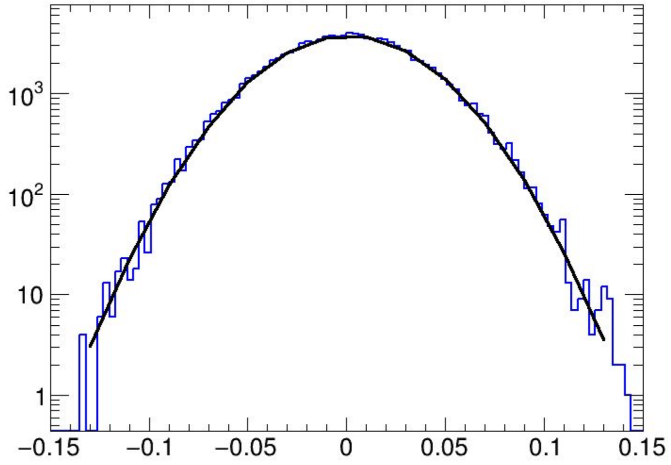
ttbar <u> = 0 residue plots

- 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



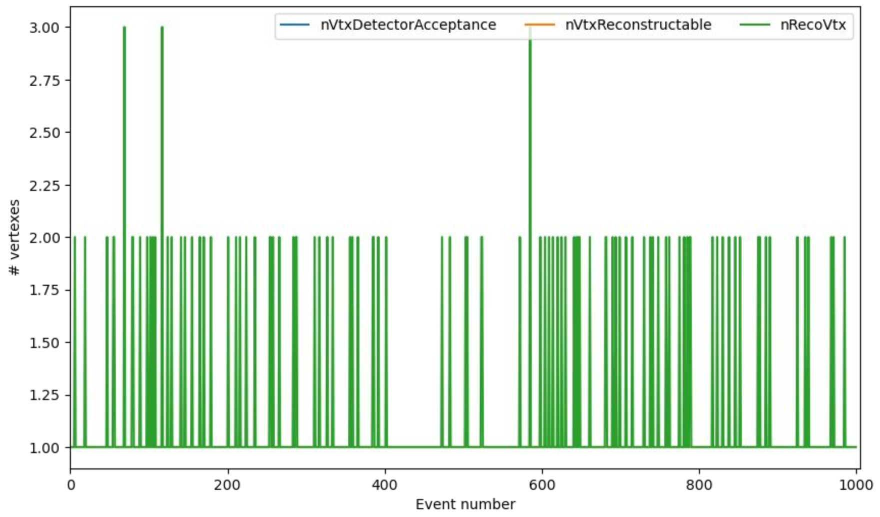
HGTD volumes



smoothed fitted (σ = 35 ps)

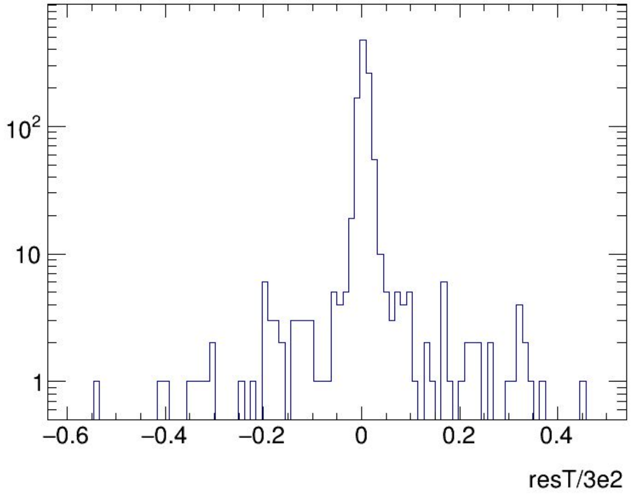
ttbar <u> = 0 vertex reconstruction efficience

- 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

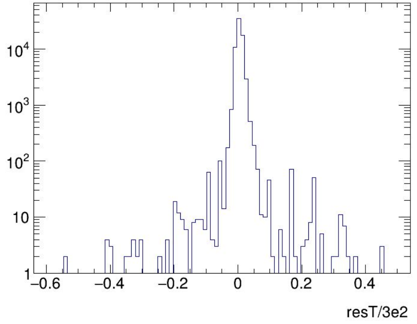


ttbar <u> = 0 vertex reconstruction residue

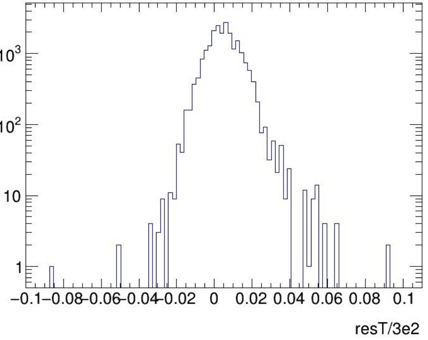
- t0 resolution is slightly better than whole time estimation, spams from -0.6 to 0.4
- If we isolate barrel region tracks ($|\eta| < 2.4$) from endcap region ($|\eta| > 2.4$) we get different distributions
  - Endcap region has better resolution, indicating that HGTD tagging is improving the reconstruction



whole detector



barrel ($|\eta| < 2.4$)

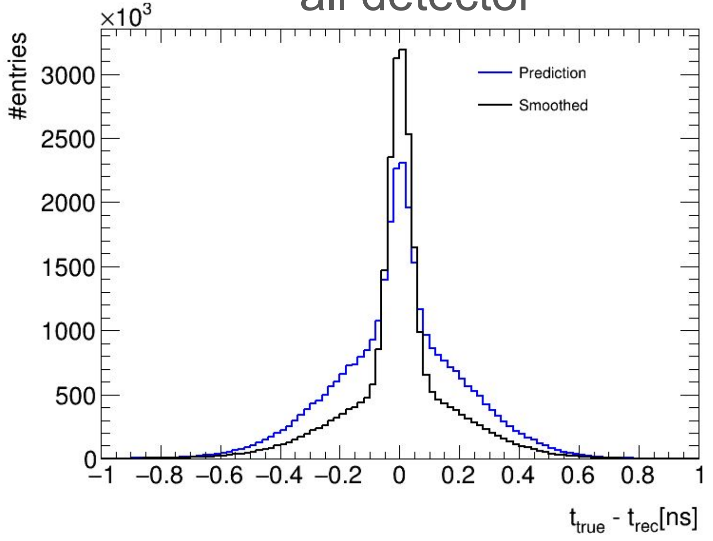


endcaps ($|\eta| > 2.4$)

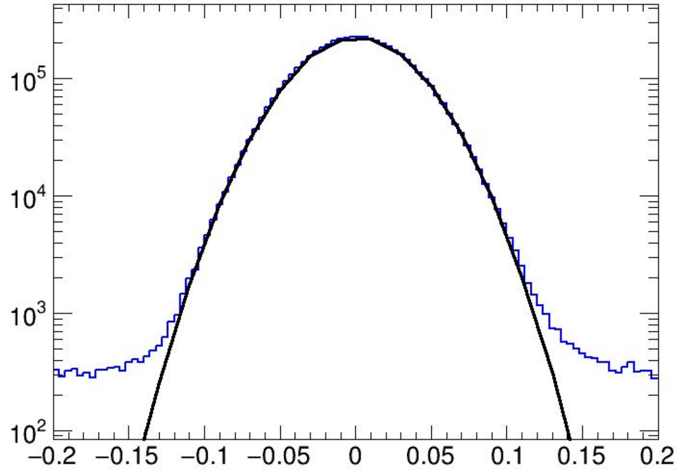
ttbar <u> = 200 residue plots

- 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



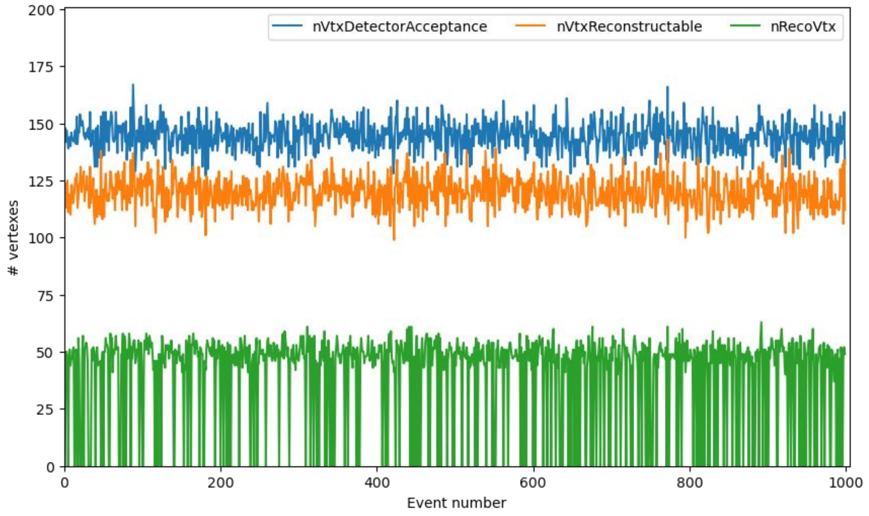
HGTD volumes

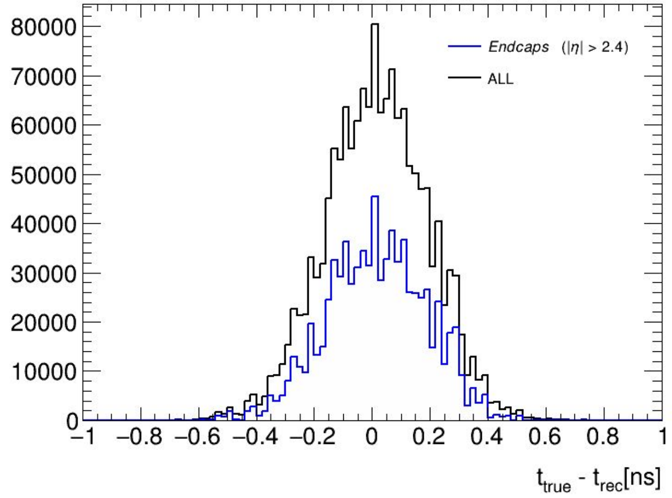


smoothed fitted (σ = 35 ps)

ttbar <u> = 200 vertex reconstruction

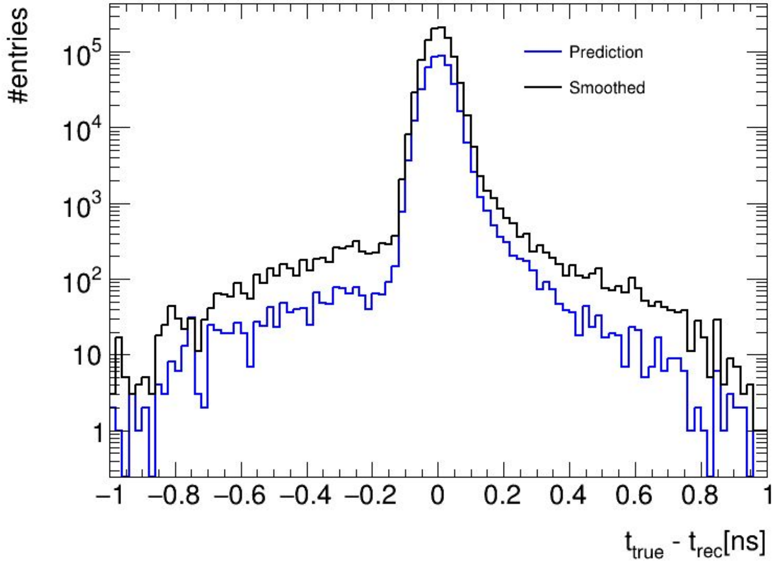
- Not all of the 201 vertex were reconstructed
  - Limited by detector geometry and preselection parameters on the reconstruction
- t0 resolution spams from -0.6 to 0.6
- No significant difference from endcaps to barrel region
  - Probably due to pileup tracks not tagged by HGTD

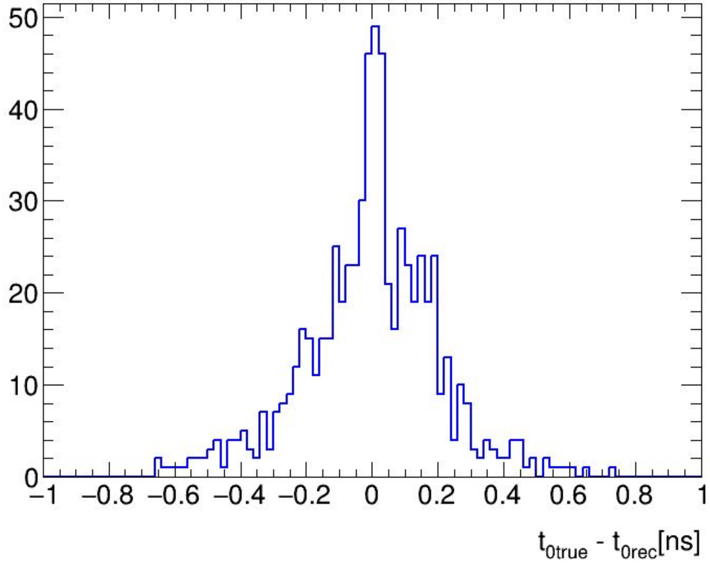




ttbar <u> = 200 for HGTD tagged tracks

- 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

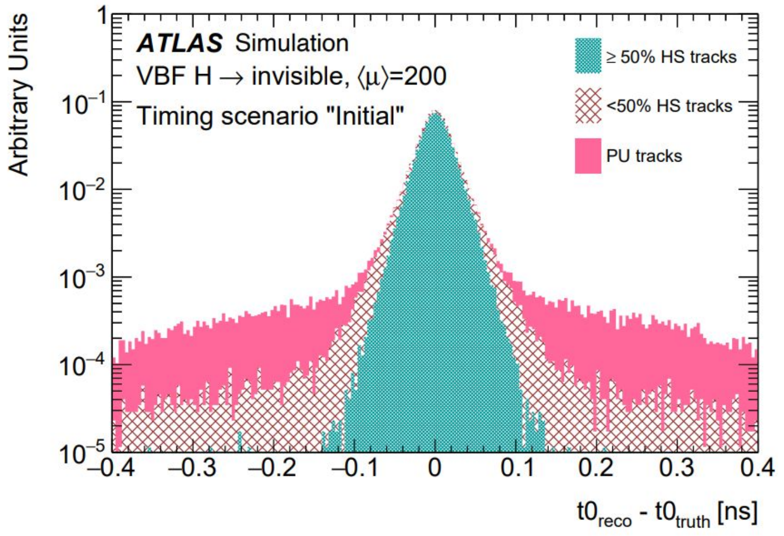




Importance of primary vertex time (t0) determination

- HGTD TDR states that (pag 37):

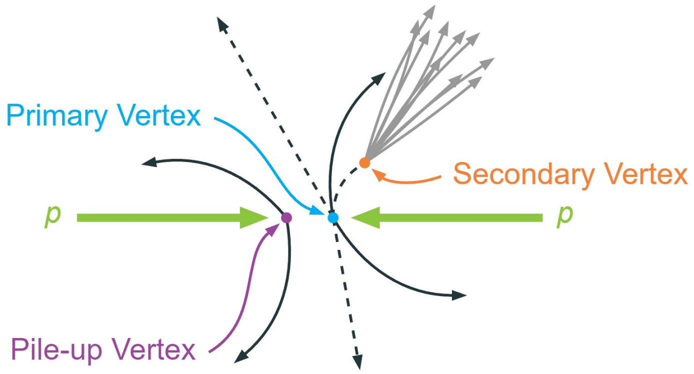
Due to the large uncertainty of the longitudinal impact parameter for tracks in the forward region (Figure 2.6), the association of tracks to nearby vertices purely based on spatial information is ambiguous in high-pileup environments, especially for low transverse momentum tracks. The ability to determine 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.

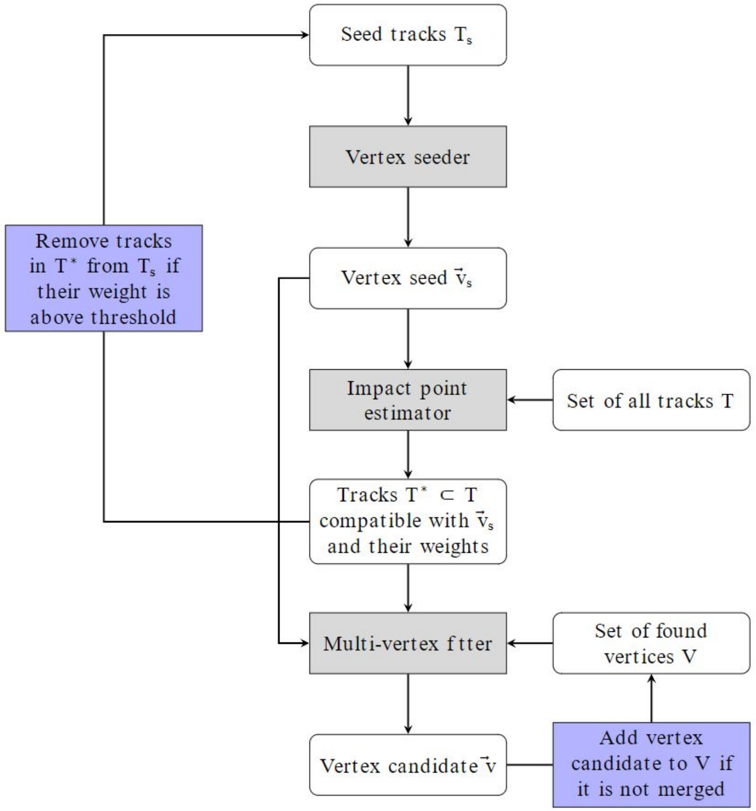


Vertex t0 resolution separately for various cases, where "HS" ("PU") stands for hard scatter (pileup). **Only track clusters tagged by HGTD were evaluated.**

Vertex reconstruction

- **Vertex finding:** cluster together the origin of tracks
- **Vertex fitting:** assume helicoidal (or linear) trajectory to enhance the estimate of the vertex
- Will deepen this explanation in future meetings




Seed tracks T_s
Vertex seeder
Vertex seed $\vec{v}_s$
Impact point estimator
Set of all tracks T
Tracks $T^* \subset T$ compatible with $\vec{v}_s$ and their weights
Multi-vertex ftter
Set of found vertices V
Vertex candidate $\vec{v}$
Add vertex candidate to V if it is not merged
Remove tracks in T^* from T_s if their weight is above threshold

Time extrapolation

- ACTS time propagation is described in this paper: (Klimpel, 2021)

$$\frac{dt}{ds} = \frac{1}{v} = \frac{E}{p} = \frac{\sqrt{m^2 + p^2}}{p} = \sqrt{m^2/p^2 + 1}$$

- In vacuum the extrapolation becomes: $t_{n+1} = t_n + h\sqrt{\frac{m^2}{p^2} + 1}.$

- Necessary to include particle mass in the state vector
- The propagation implemented in ACTS is done in vacuum, but its also possible to use Range Kutta to make this extrapolation with more precision
- This propagation is already included in the CKF but as no estimates or measurements are evaluated, this parameter is not properly reconstructed

Time measurements and smearing

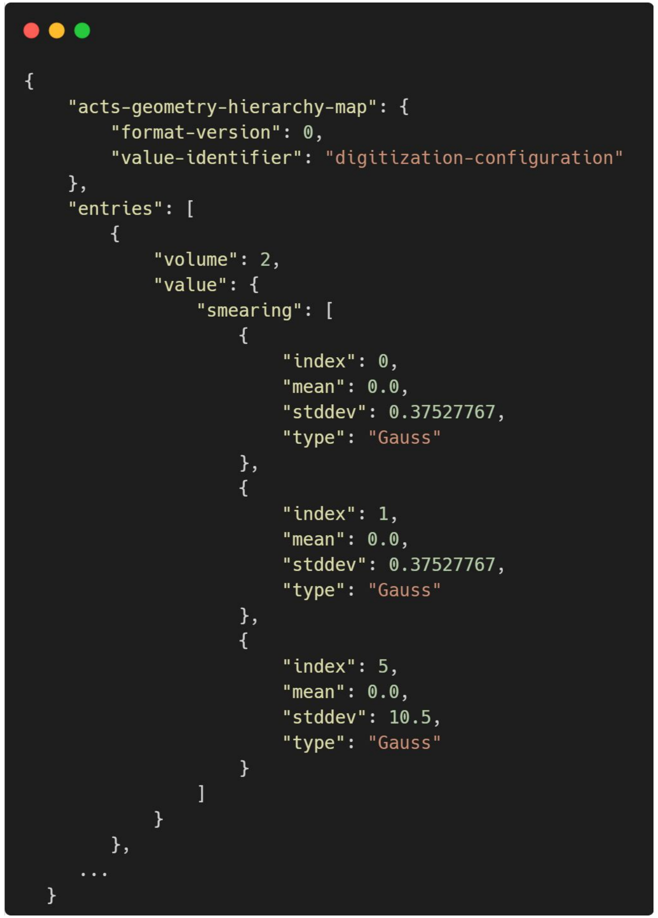
- At the digitization step, ACTS uses a geometry config file to simulate smearing of measurements
- We included the time parameter at volumes 2 and 25, which represent HGTD
  - Other volumes just measure l_0 and l_1

$$\vec{x} = (l_0, l_1, \phi, \theta, q/p, t)^T$$

- Got the smearing time with HGTD group

$$\sigma = 35 \text{ ps} \qquad \sigma_t = s \times \sigma$$

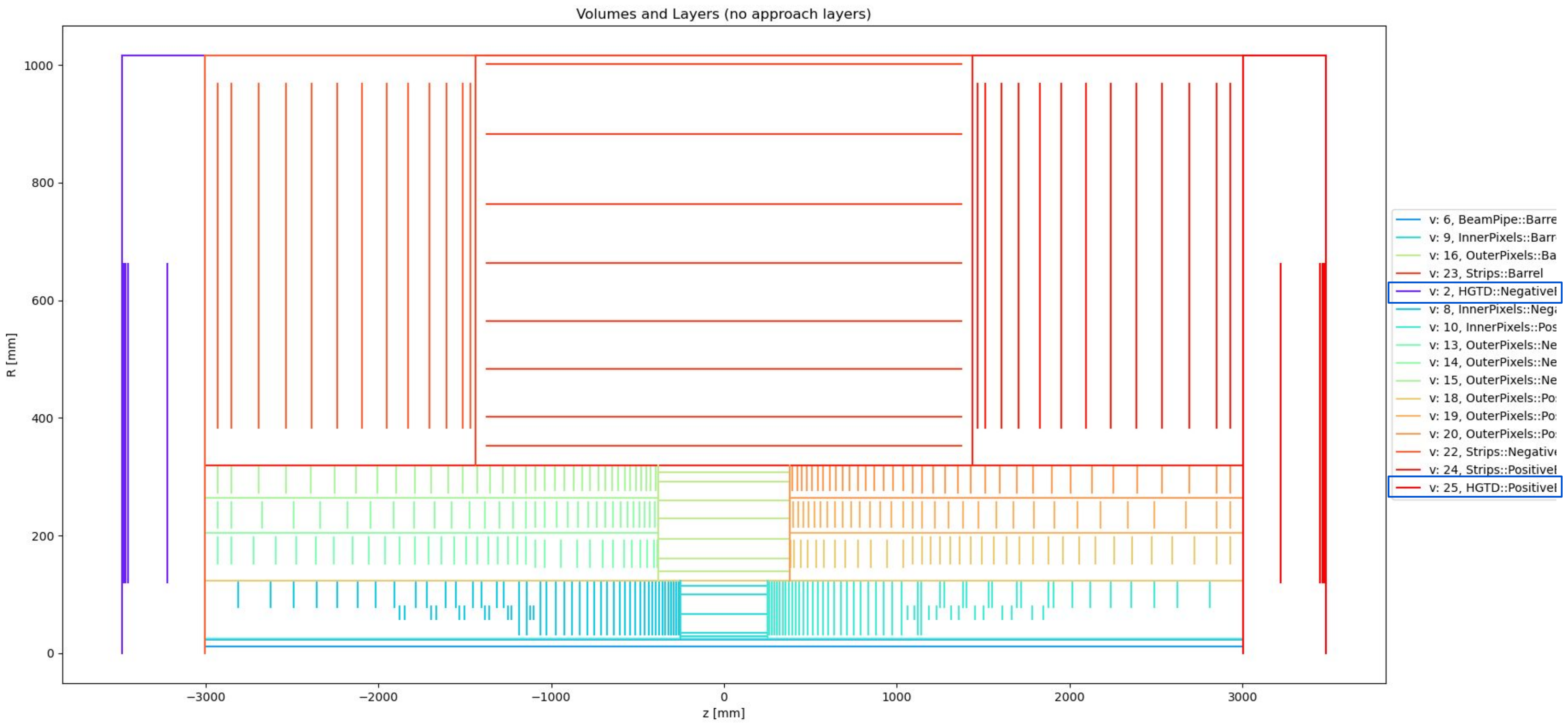
$$s = 299792458 \text{ mm/s} \qquad \sigma_t = 10.5 \text{ mm}$$



```
{
    "acts-geometry-hierarchy-map": {
        "format-version": 0,
        "value-identifier": "digitization-configuration"
    },
    "entries": [
        {
            "volume": 2,
            "value": {
                "smearing": [
                    {
                        "index": 0,
                        "mean": 0.0,
                        "stddev": 0.37527767,
                        "type": "Gauss"
                    },
                    {
                        "index": 1,
                        "mean": 0.0,
                        "stddev": 0.37527767,
                        "type": "Gauss"
                    },
                    {
                        "index": 5,
                        "mean": 0.0,
                        "stddev": 10.5,
                        "type": "Gauss"
                    }
                ]
            }
        },
      ...
  }
```

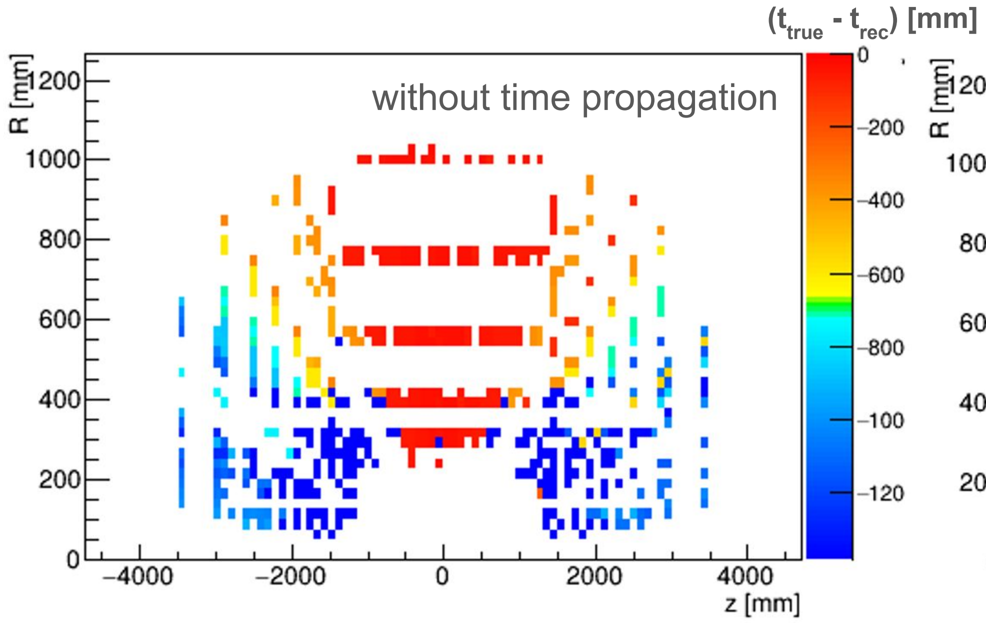
ITK + HGTD geometry at ACTS

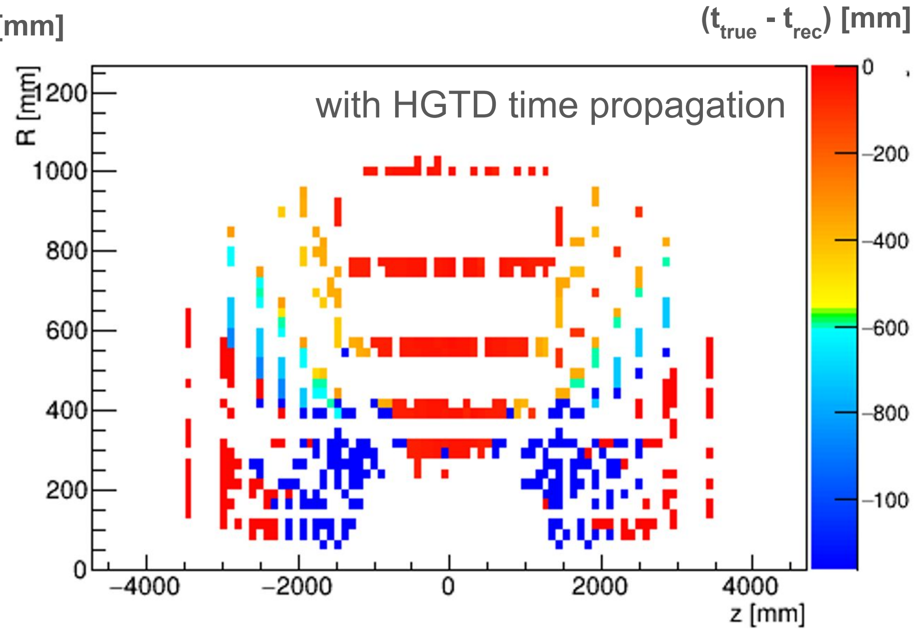
- HGTD endcaps are mapped to volume 2 and 25 of our geometry file



Simulation with particle guns

- Aimed to analyse the time residue (t_{true} - t_{rec}) before and after the smearing inclusion
- 100 events with particle gun of a single muon distributed uniformly between eta -4 and 4
- Regional plots below show the **mean time residue after smoothing** for each region





Changing first time estimate

- The actual CKF implementation uses the first measurement as the initial estimate
- As there's no time reading at the ITk sensors, the first estimate will have the time coordinate being 0.

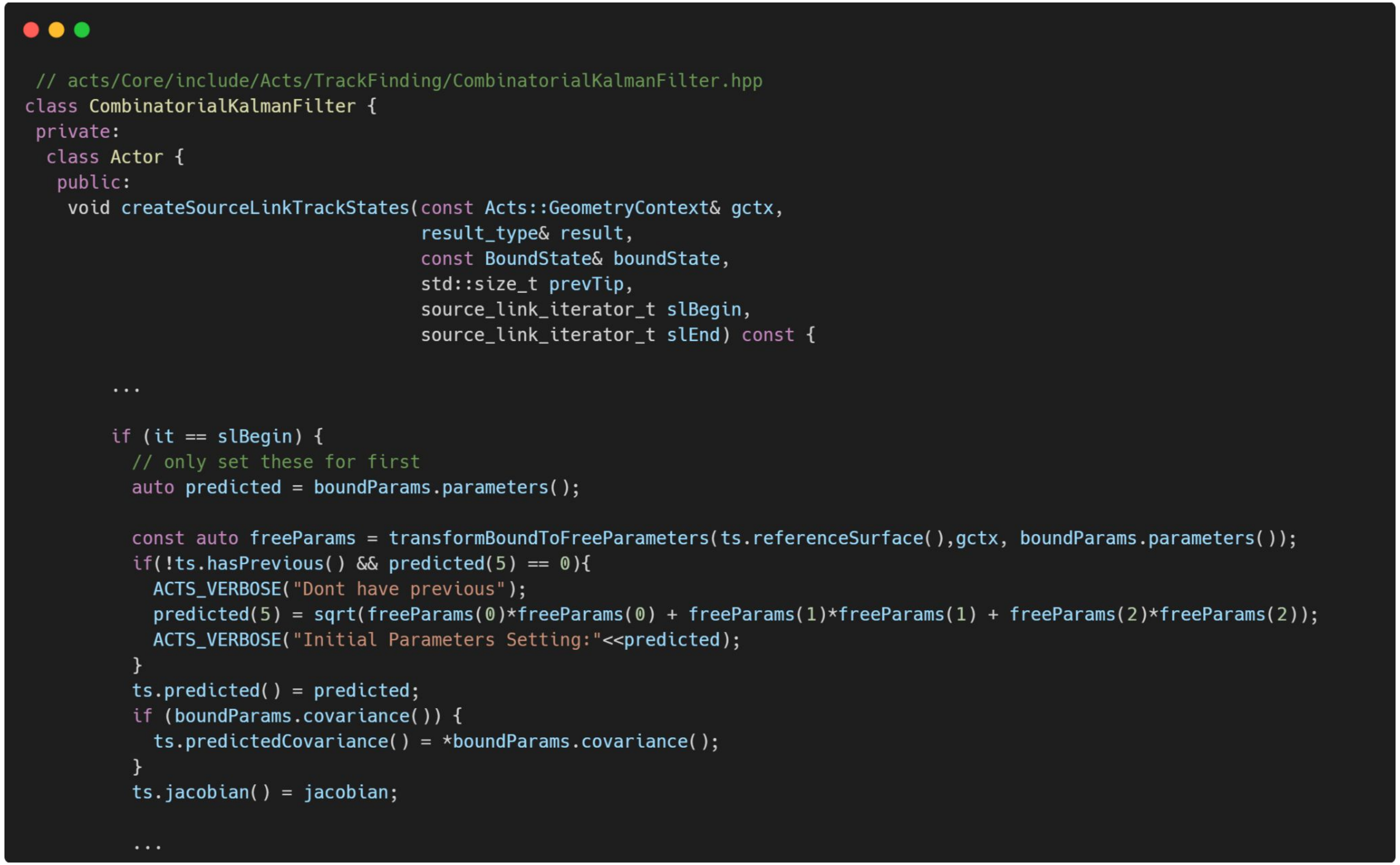
What would be a good first estimate for the time at the first hit?

- First idea: distance between first hit and collision centre

$$t_1 = \sqrt{x_1^2 + y_1^2 + z_1^2}$$

- Works well if p >> m then $t_{n+1} = t_n + h$
- And if the particle is generated at the position and time 0

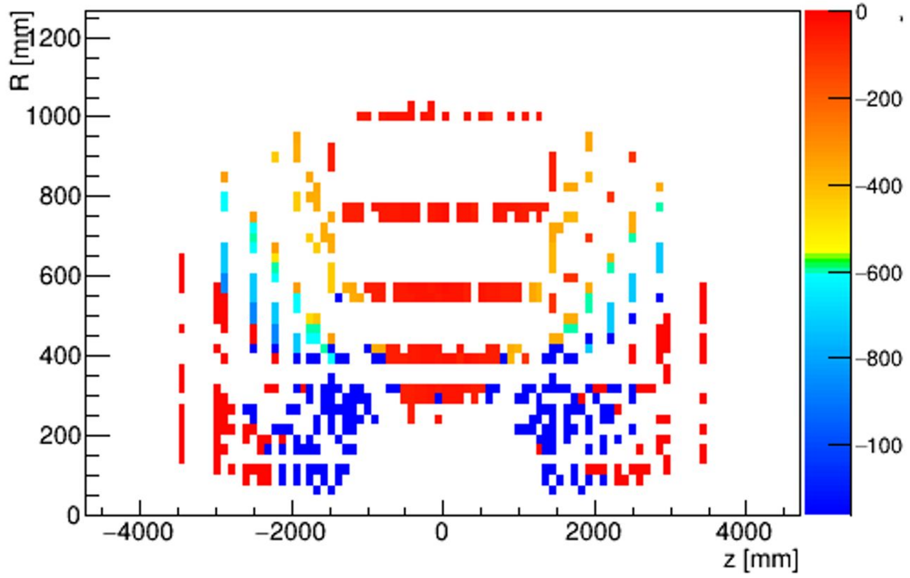
Changing first time estimate (code implementation)

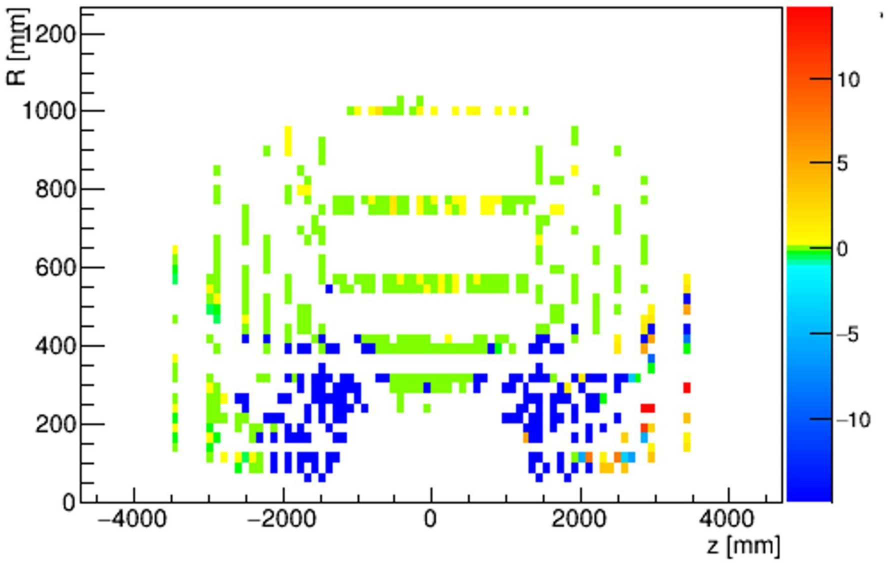


```
 // acts/Core/include/Acts/TrackFinding/CombinatorialKalmanFilter.hpp
class CombinatorialKalmanFilter {
 private:
  class Actor {
   public:
    void createSourceLinkTrackStates(const Acts::GeometryContext& gctx,
                                     result_type& result,
                                     const BoundState& boundState,
                                     std::size_t prevTip,
                                     source_link_iterator_t slBegin,
                                     source_link_iterator_t slEnd) const {

        ...

        if (it == slBegin) {
          // only set these for first
          auto predicted = boundParams.parameters();

          const auto freeParams = transformBoundToFreeParameters(ts.referenceSurface(),gctx, boundParams.parameters());
          if(!ts.hasPrevious() && predicted(5) == 0){
            ACTS_VERBOSE("Dont have previous");
            predicted(5) = sqrt(freeParams(0)*freeParams(0) + freeParams(1)*freeParams(1) + freeParams(2)*freeParams(2));
            ACTS_VERBOSE("Initial Parameters Setting:"<<predicted);
          }
          ts.predicted() = predicted;
          if (boundParams.covariance()) {
            ts.predictedCovariance() = *boundParams.covariance();
          }
          ts.jacobian() = jacobian;

          ...
```

Simulation with particle guns

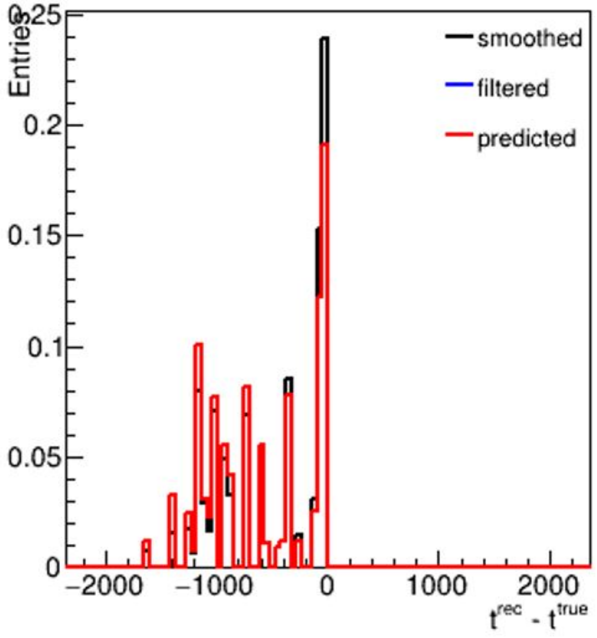
- Aimed to analyse the time residue (t_{true} - t_{rec}) before and after the smearing inclusion
- 100 events with particle gun of a single muon distributed uniformly between eta -4 and 4
- Regional plots below show the **mean time residue after smoothing** for each region

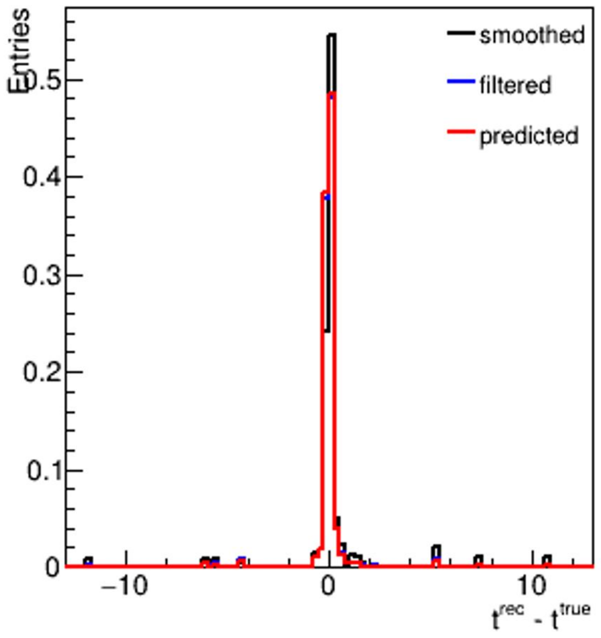




Performance of particle guns reconstruction

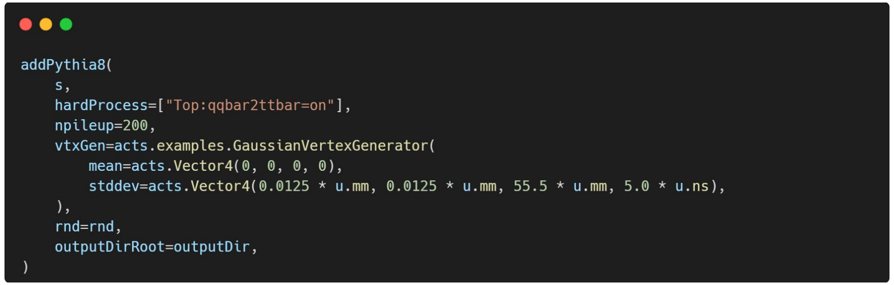
- For this scenario, the residue now is centred at zero with a variance lower than HGTD resolution
- Outliers happen because of HGTD resolution being way bigger than the prediction error





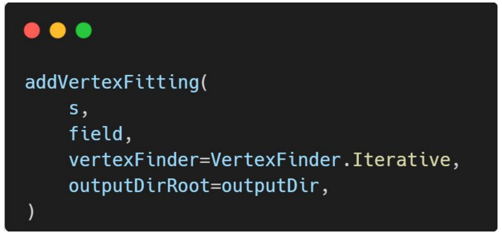
Simulation of ttbar events

- For a more complex scenario, we used ttbar events(Top: qq’->tt’)
  - Jets (particle sprays) to the frontal region



```
addPythia8(
    s,
    hardProcess=["Top:qqbar2ttbar=on"],
    npileup=200,
    vtxGen=acts.examples.GaussianVertexGenerator(
        mean=acts.Vector4(0, 0, 0, 0),
        stddev=acts.Vector4(0.0125 * u.mm, 0.0125 * u.mm, 55.5 * u.mm, 5.0 * u.ns),
    ),
    rnd=rnd,
    outputDirRoot=outputDir,
)
```

- Now particles aren’t only generated at instant 0 (spread of 5 ns), making our first estimate inaccurate
  - HGTD has an important role in fixing the time prediction for tracks generated at t0 != 0

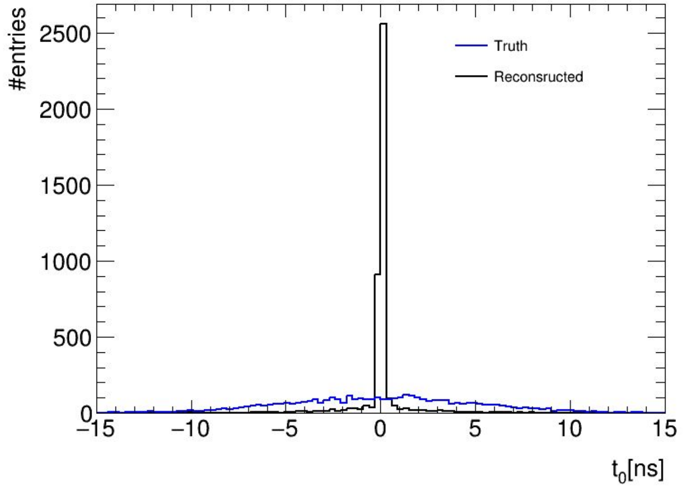


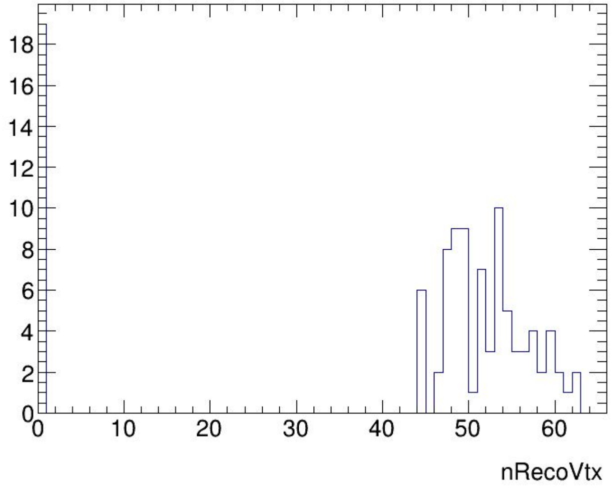
```
addVertexFitting(
    s,
    field,
    vertexFinder=VertexFinder.Iterative,
    outputDirRoot=outputDir,
)
```

Also added Vertex reconstruction

ttbar <u> = 200 vertex reconstruction

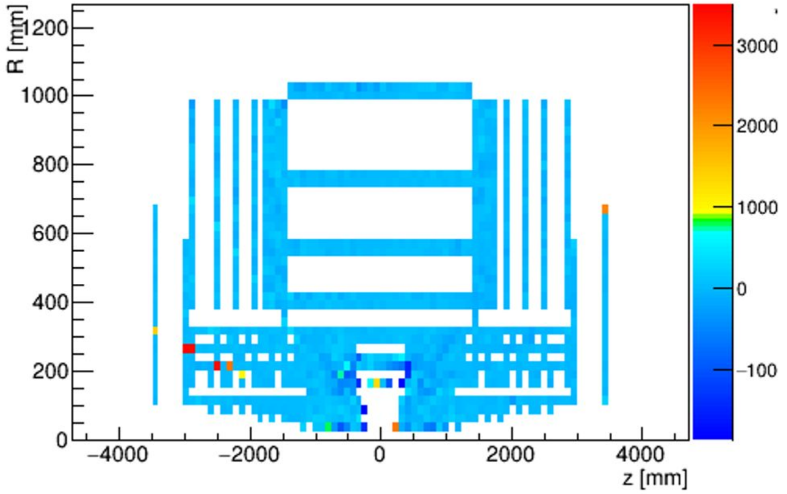
- Reconstruction fails to reconstruct vertexes
    - Error in the scale of ns
    - From 200 vertex only ~60 were reconstructed per event
- Tried to filter only particles with HGTD hits (eta > 2.4 and high pT) but the reconstruction still doesn't work

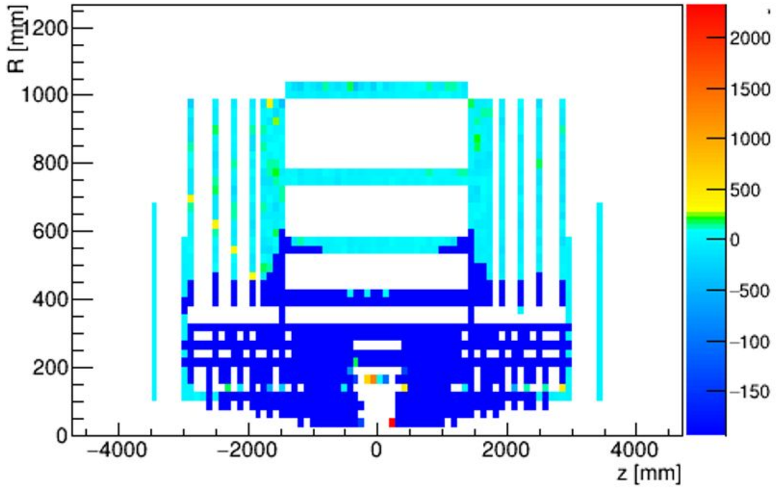




Points of improvement

- Use more accurate event generation time smearing
  - For now, vertexes are generated with $t_0 \sim N(0,5ns)$, but that the stddev is way higher than it should be
- Improve (understand) the smoothing step of the CKF
  - Have to fix weird behaviour where smoothed samples are worse than filtered ones





- Need to understand Vertex reconstruction methods