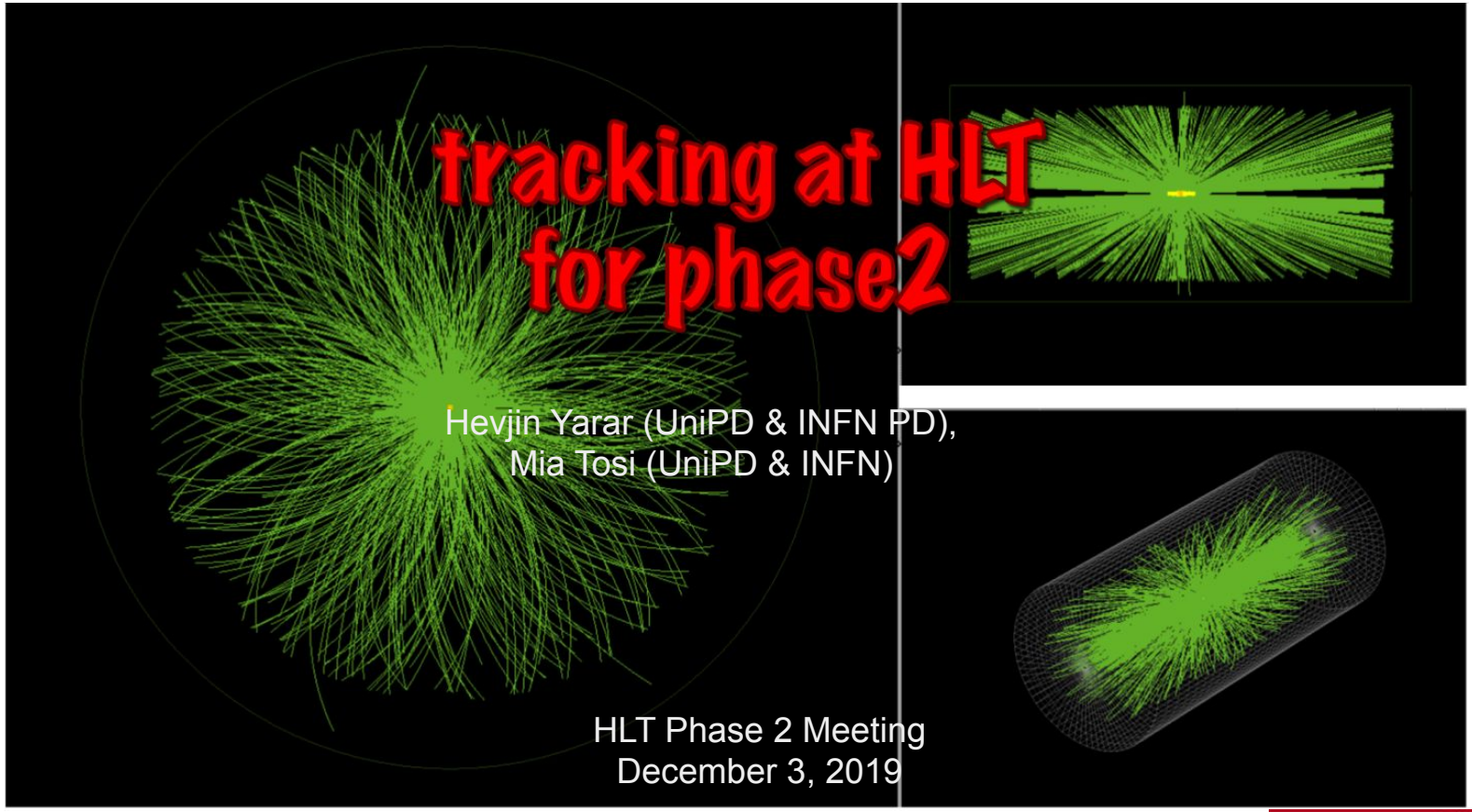
# tracking at HLT for phase2

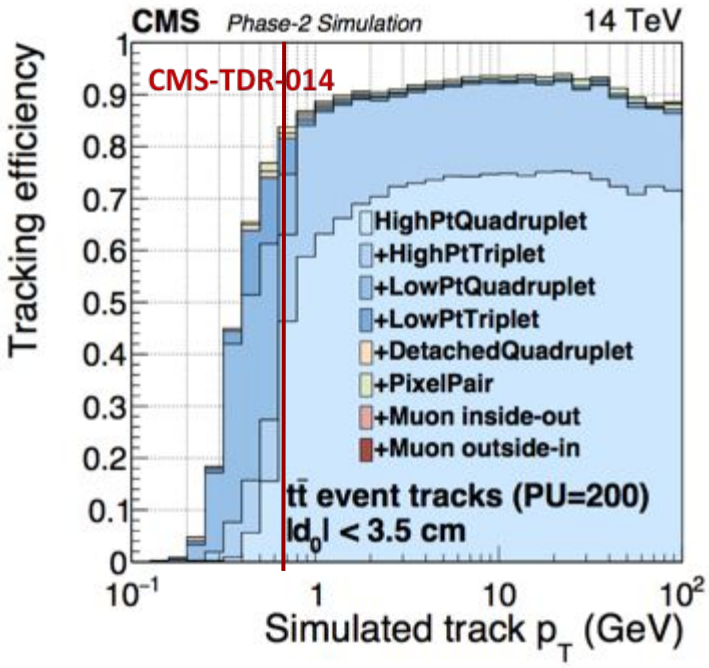


Hevjin Yarar (UniPD & INFN PD),
Mia Tosi (UniPD & INFN)

HLT Phase 2 Meeting
December 3, 2019

# Current Phase 2 Reconstruction - Performance

- the offline tracking for Phase2 is based on the **Iterative Tracking**




| | Step Name | Seeding | Target Tracks |
|---|---|---|---|
| * | HighPtQuad | pixel quadruplets | prompt, high $p_T$ |
| * | HighPtTriplet | pixel triplets | prompt, high $p_T$, recovery |
| * | LowPtQuad | pixel quadruplets | prompt, low $p_T$ |
| | LowPtTriplet | pixel triplets | prompt, low $p_T$, recovery |
| | DetachedQuad | pixel quadruplets | displaced |
| * | PixelPair | pixel pairs | high $p_T$, recovery |
| | Muon Inside-Out | muon-tagged tracks | muon |
| | Muon Outside-In | muon-tagged tracks | muon |

only for the pixel recovery)

* iterations as we have in the current HLT configuration

# Current Phase 2 Reconstruction - Timing

From our previous slides:
(previous release)

| | Efficiency [%] | | Fakerate [%] | | | <timing> per event [ms] | | |
|---|---|---|---|---|---|---|---|---|
| | <PU>=0 | <PU>=200 | <PU>=0 | <PU>=200 | | <PU>=0 | <PU>=200 | |
| HighPtQuad | 77 | 74 | 0.25 | 1.8 | x7 | 140 | 6600 | x50 |
| HighPtTriplet | 15 | 15 | 0.7 | 25 | x36 | 28 | 3000 | x115 |
| LowPtQuad | 0.8 | 0.9 | 0.3 | 7.8 | x26 | 60 | 7300 | x120 |
| LowPtTriplet | 0.05 | 0.09 | 1 | 35 | x35 | 10 | 1600 | x180 |
| DetachedQuad | 0.6 | 0.5 | 0.7 | 17 | x24 | 12 | 3900 | x350 |
| PixelPair | 1 | 1 | 2 | 55 | x27 | 6 | 450 | x90 |

* w/in the HighPtQuad <timing> we are not taking into account the currently needed PV reconstruction, which costs ***8*** and ***2400 ms*** at <PU>=0 and <PU>=200 respectively , as now

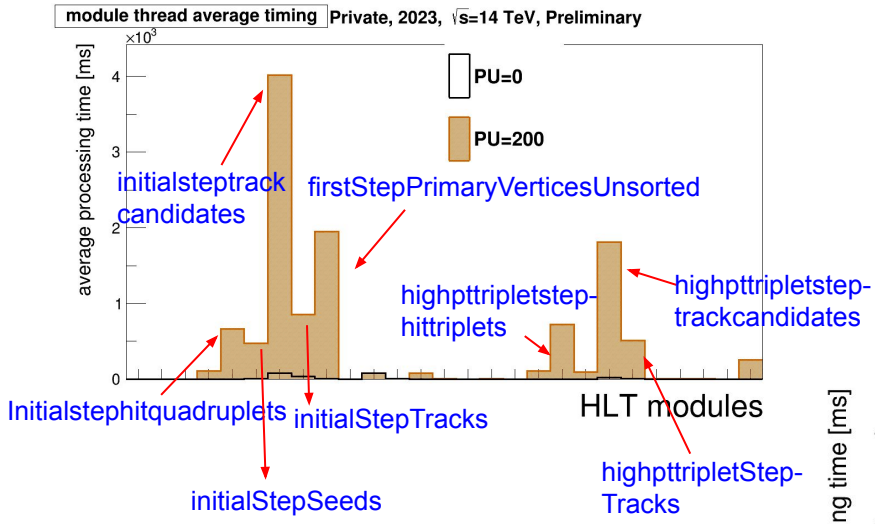
per thread

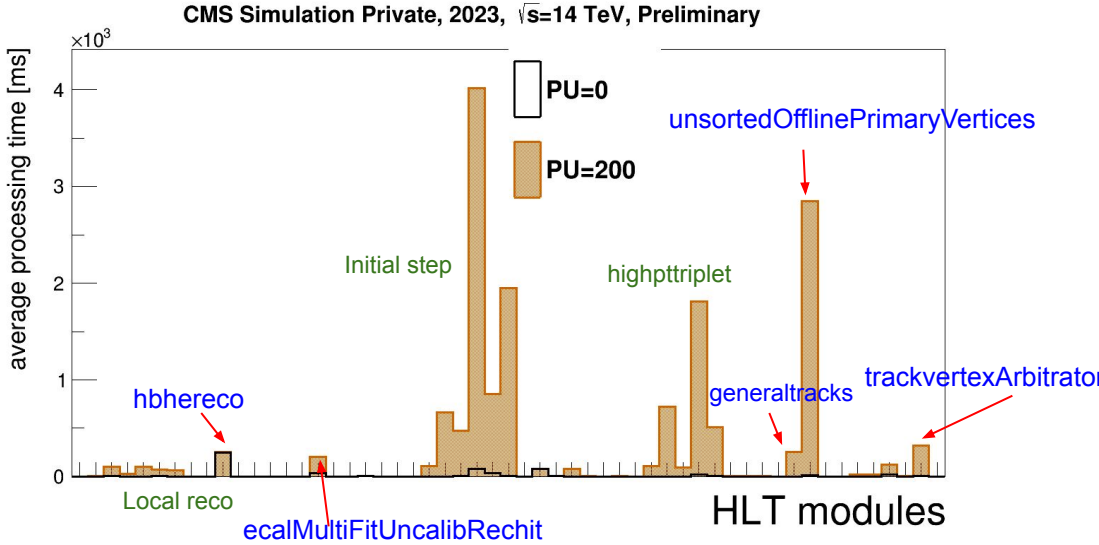
# To Do

- reduce the number of iterations from 6(8) to 2 iterations ✓
- tuning the **pattern recognition** in order to minimize the timing (while keeping the physics performance) (✓)
    - i.e. limit the number of candidates in the pattern recognition ✓
- update: using the **beamspot** instead of the PV in the track selection (~✓)
- Increase the thresholds (pT > 0.9 GeV) (ongoing)
- Limit pseudorapidity (|eta| < 3) (ongoing)
- Check usage of pixel PV
- Check usage of pixelTracks
- Try seed cleaning
- Check effect of PV constraint

- MC_Tracking_v0 : it represents the minimum set of modules for the track reconstruction as it is done in the offline reconstruction ✓
- MC_Tracking_v2 : path with 2 iterations ✓
    - Performance and Timing Plots with this version ✓
    - Working with CMSSW_11_0_0_pre6 + Run on new samples (106X, thanks !) ✓

# CMSSW_11 Baseline Timing



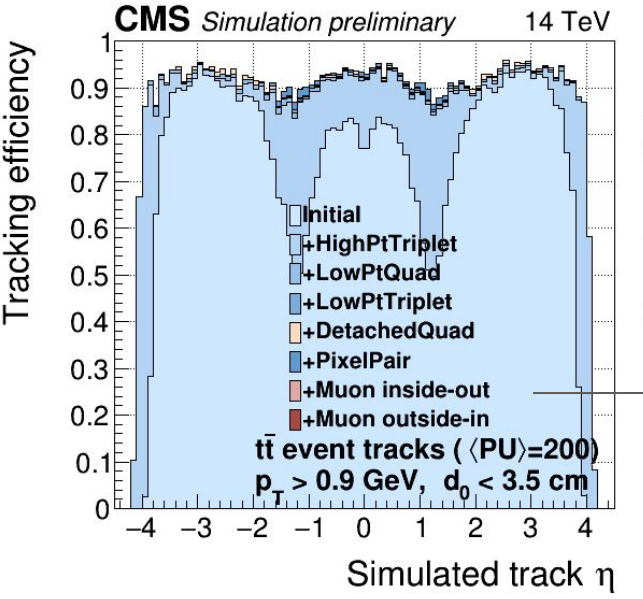



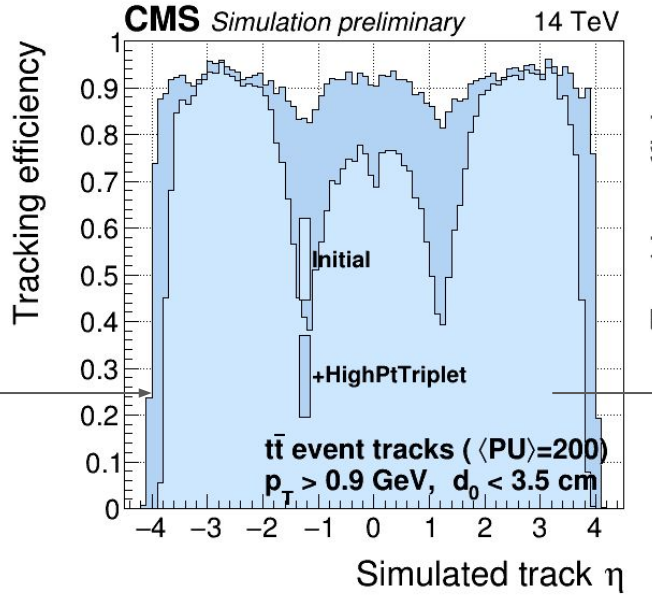

Inner tracking / total

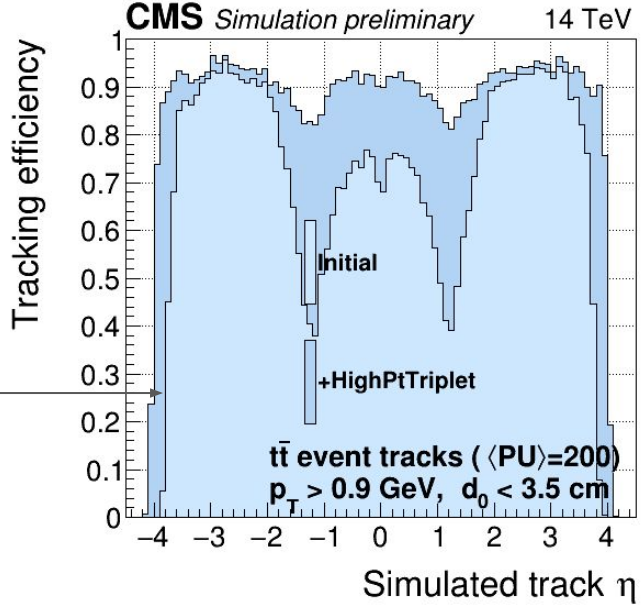
PU=0: 255 / 600 ms [ 42% ]

PU=200 : 11537 / 15985 ms [ 72% ]

# Adapting MC_Tracking_v2 to HLT - Performance Plots



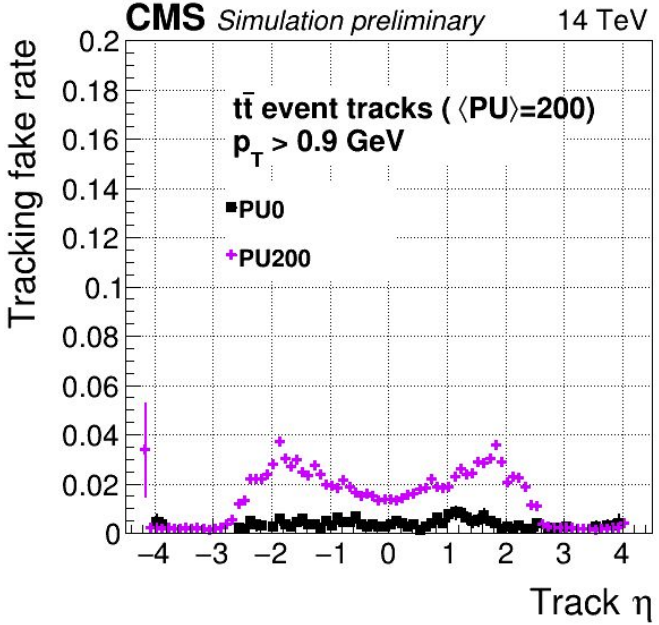




Baseline CMSSW_11
Efficiency vs Eta - PU200

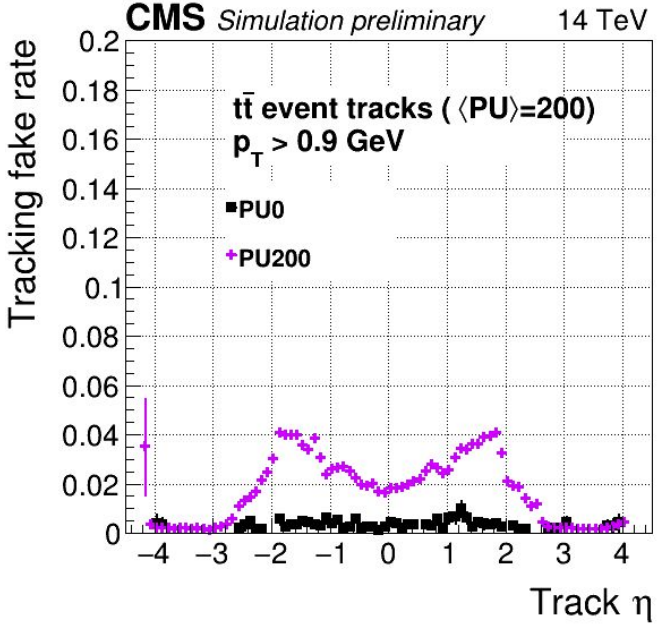
Efficiency vs Eta - PU200

# Adapting MC_Tracking_v2 to HLT - Performance Plots



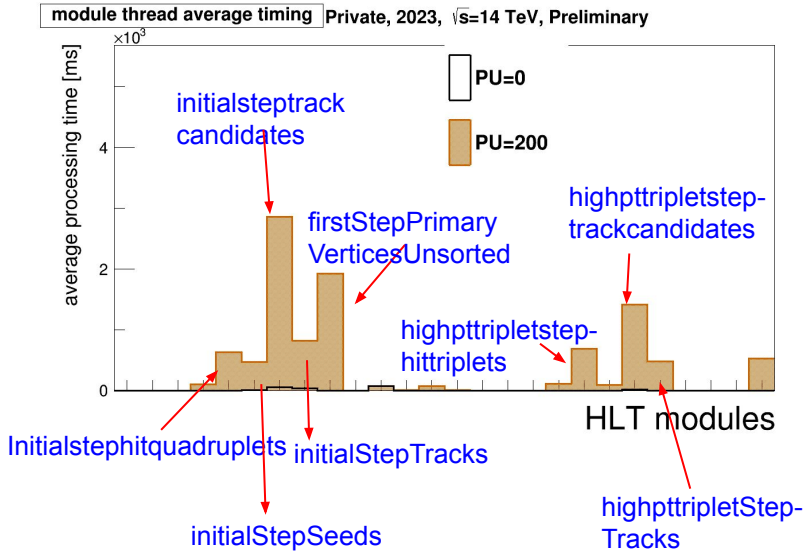

Baseline

FakeRate vs PU

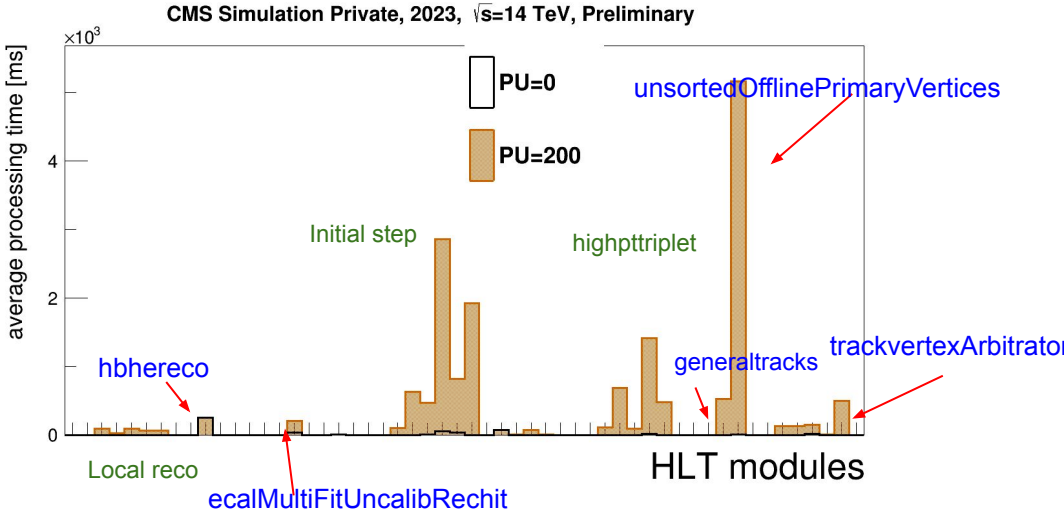



FakeRate vs PU

# Timing




- Time for “initialsteptrackcandidates” and “highpttripletsteptrackcandidates” decreased
- Time for “unsortedOfflinePrimaryVertices” increased (slightly more fakes)




inner tracking / total

PU=0: 235 / 590 ms [ 40% ]

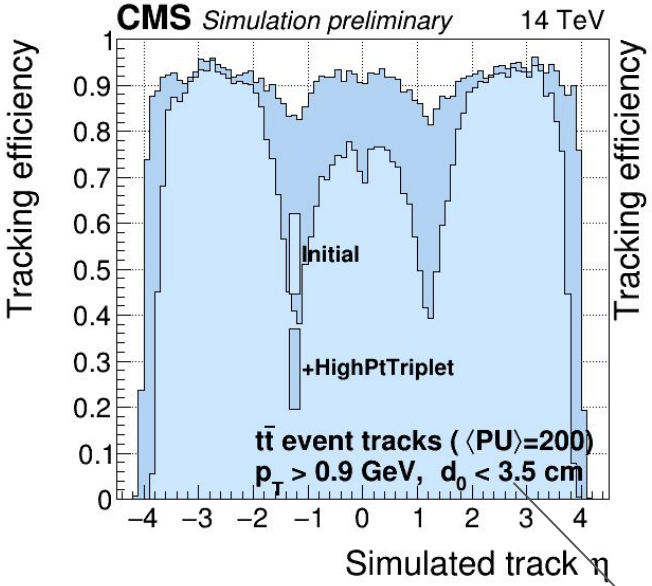
PU=200: 9820 / 17271 ms [ 57% ]

**timing gain : 2.5 s (15%)**

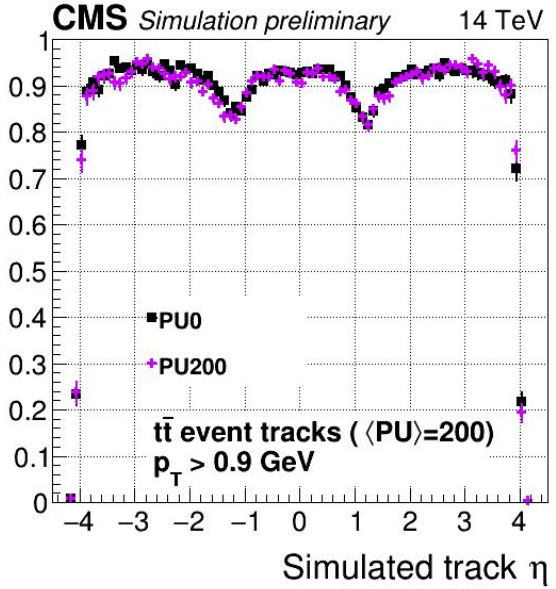
# BACKUP

Backup 0

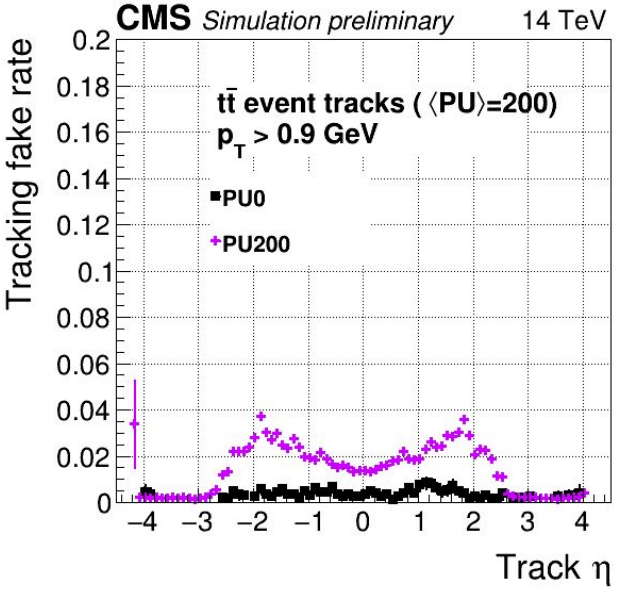
# CMSSW_11 Baseline Performance Plots




Efficiency vs Eta - PU200



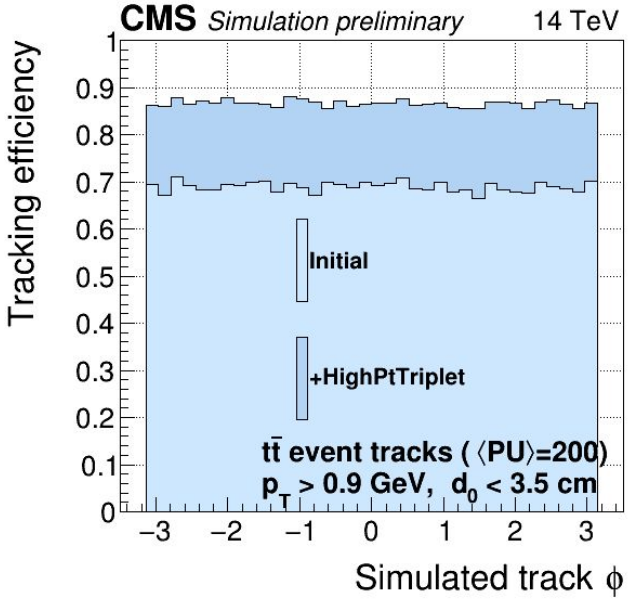

Efficiency vs PU



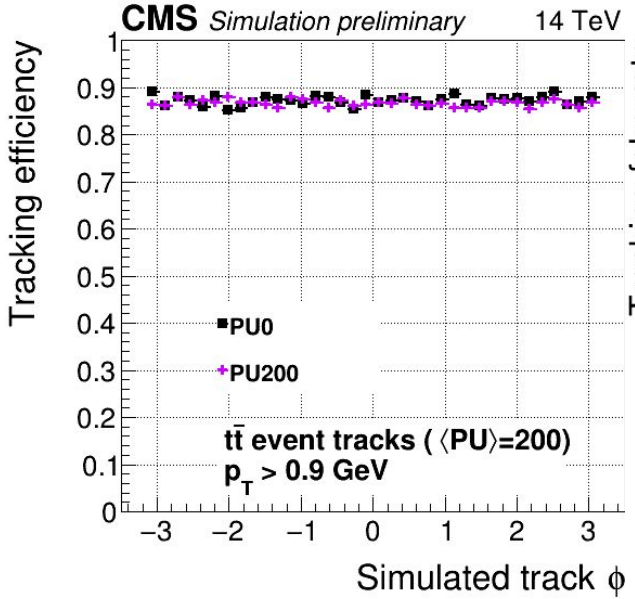

Fake Rate vs PU

d0 < 2.5 cm

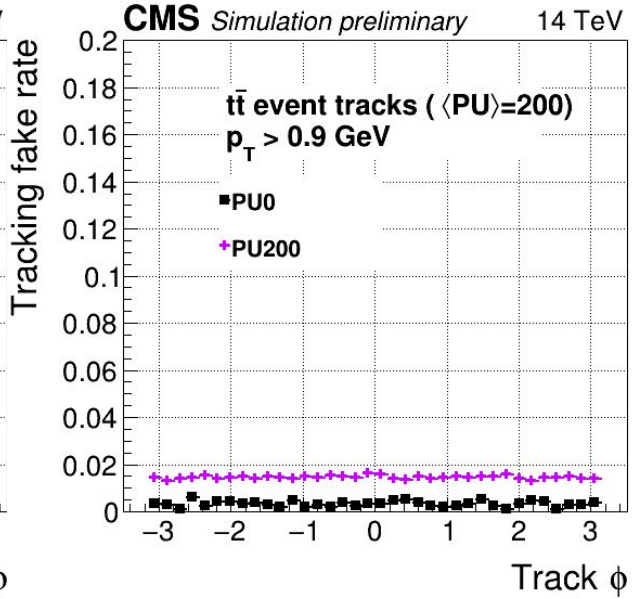
# CMSSW_11 Baseline Performance Plots




Efficiency vs Phi - PU200




Efficiency vs PU

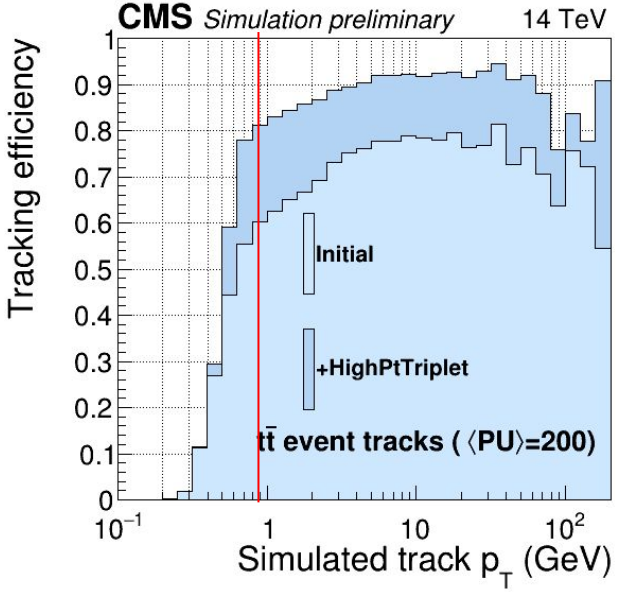



Fake Rate vs PU

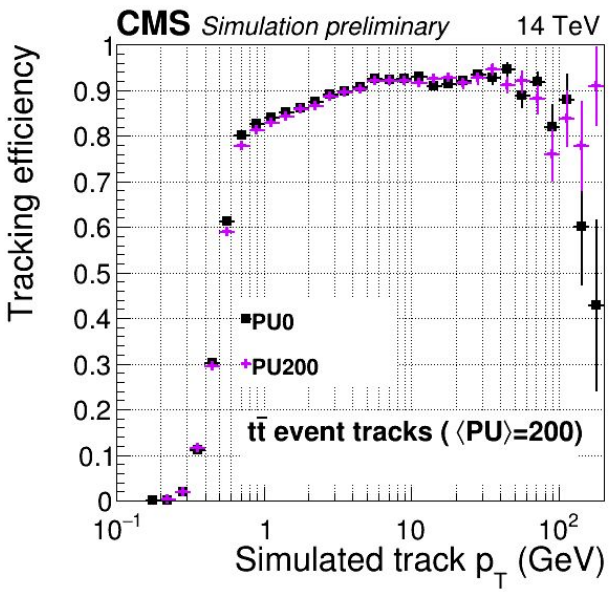
See Backup2 for the rest of the plots

Backup 0

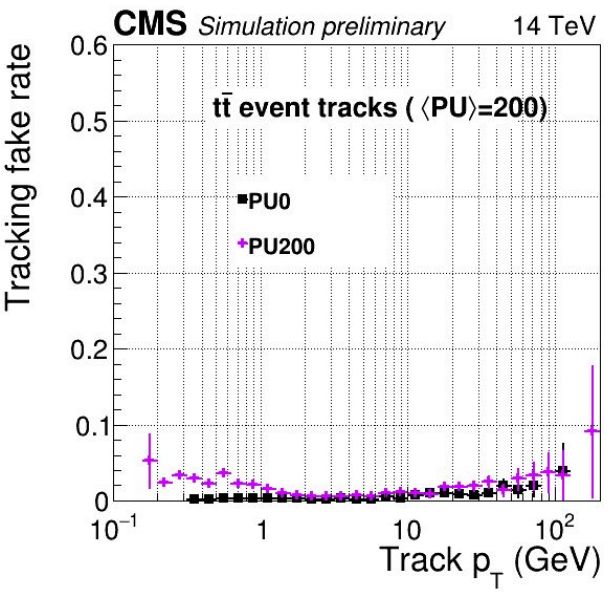
# CMSSW_11 Baseline Performance Plots




Efficiency vs Pt - PU200



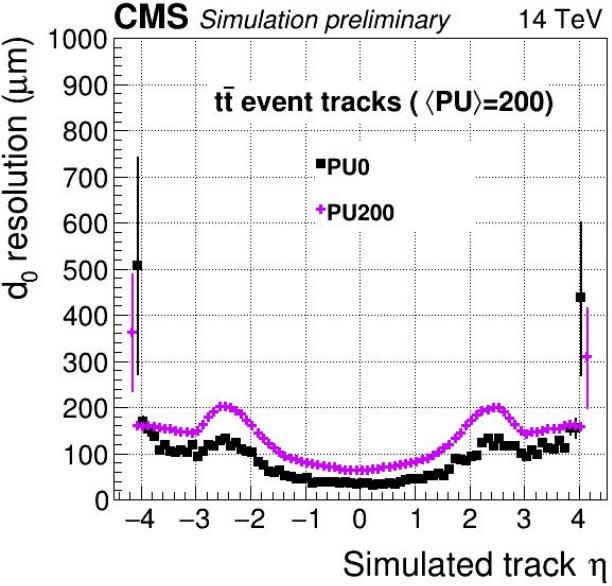

Efficiency vs PU



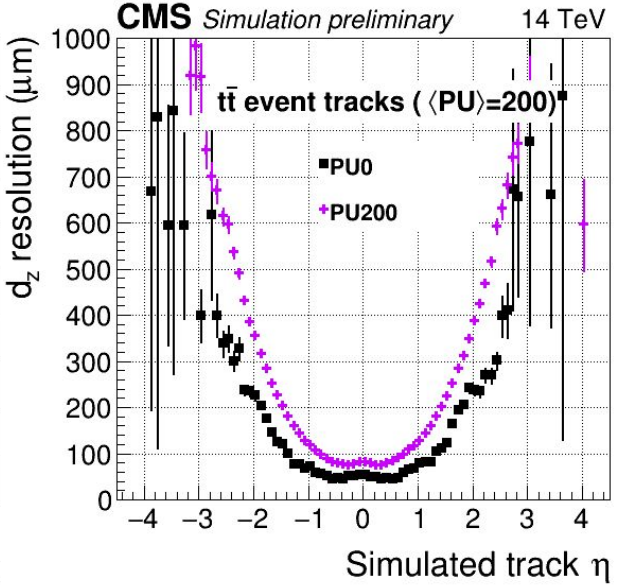

Fake Rate vs PU

Backup 0

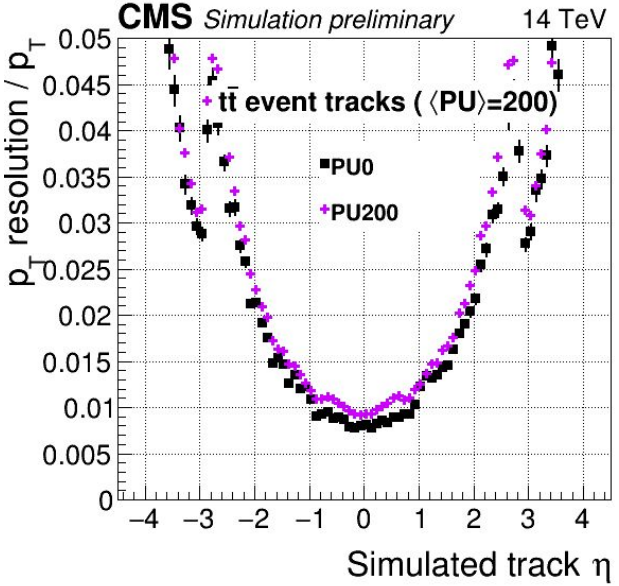
# CMSSW_11 Baseline Performance Plots




dxy res vs eta



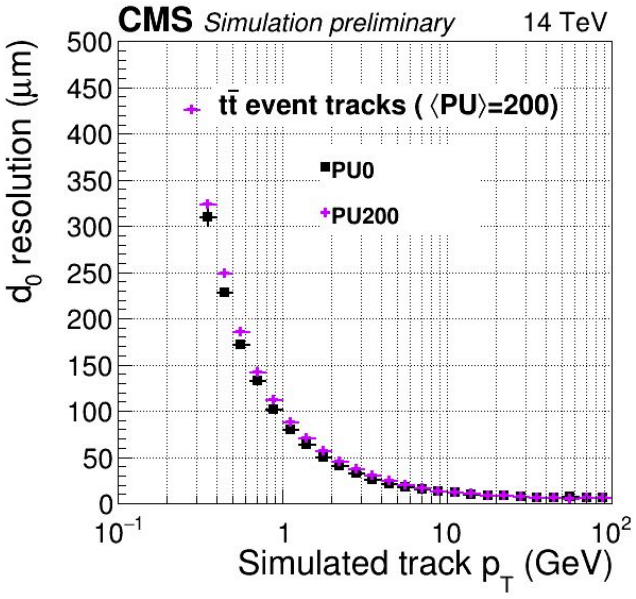

dz res vs eta



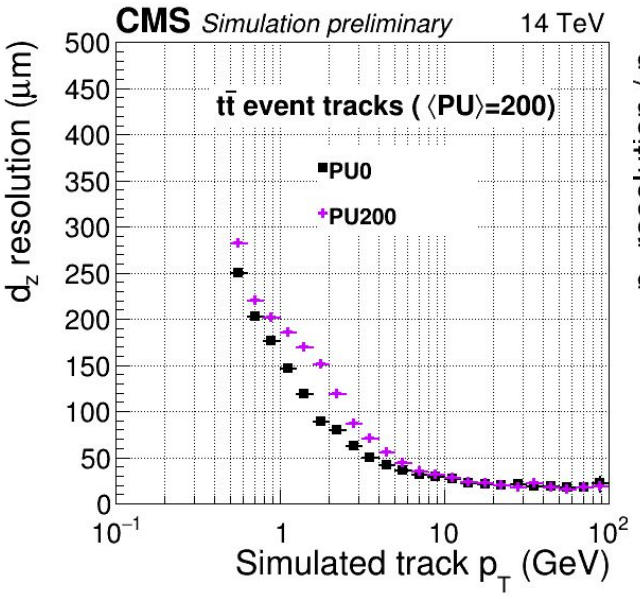

pt res vs eta

Backup 0

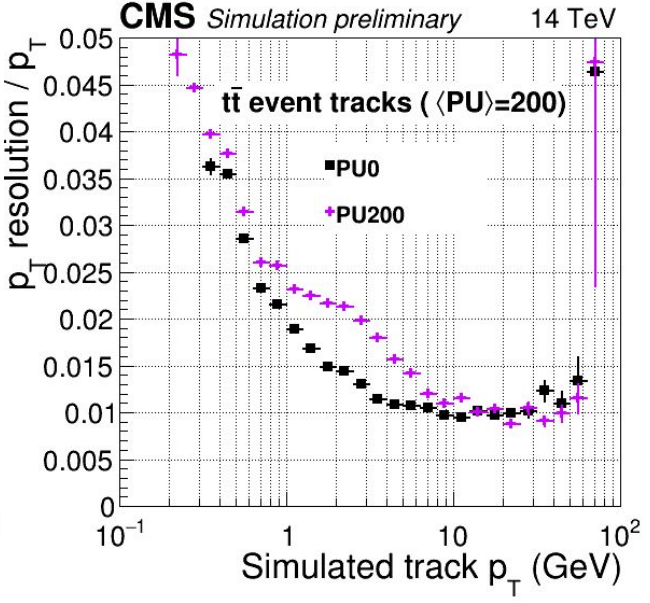
# CMSSW_11 Baseline Performance Plots




dxy res vs pt




dz res vs pt




pt res vs pt

Backup 1

Switch to CMSSW_11_0_0_pre6 - Part 1

## Step3 from the release (trackingOnly):

> cmsrel CMSSW_11_0_0_pre6

> cd CMSSW_11_0_0_pre6/src/ && cmsenv

> runTheMatrix.py -w upgrade -n | grep 2026 | grep trackingOnly | grep 14TeV

> runTheMatrix.py -w upgrade -l 20434.1 > 20434.1.log &

## Reconstruction and validation done with:

cmsRun step3_RAW2DIGI_RECO_VALIDATION_DQM.py (MC samples? For now running on CMSSW_10_6 samples)

Backup 1

Switch to CMSSW_11_0_0_pre6 - Part 2

→ /afs/cern.ch/work/h/hyarar/public/Phase2/MC_Tracking/MC_Tracking_CMSSW_11_0_pre6

## Release generated script with cleaned up cff files & paths

Version 2) Release generated script with new paths plugged in
############
step3_RAW2DIGI_RECO_VALIDATION_DQM.py

needs:
- step2.root
- extras_cmssw_11_0_cff.py (extra modules needed)
- raw2digi_step_cff.py
- MC_Tracking_v0_cmssw_11_0_cff.py / MC_Tracking_v1_cmssw_11_0_cff.py / MC_Tracking_v2_cmssw_11_0_cff.py
- MC_prevalidation_v0_cff.py / MC_prevalidation_v1_cff.py / MC_prevalidation_v2_cff.py
- MC_Dqmoffline_step_v0_cff.py / MC_Dqmoffline_step_v1_cff.py / MC_Dqmoffline_step_v2_cff.py

example

```
process.load('raw2digi_step_cff')
process.load('MC_Tracking_v2_cmssw_11_0_cff')
process.load('MC_prevalidation_v2_cff')
process.load('MC_Dqmoffline_v2_cff')
```

slight changes in the new release, like a parameter called seedAs5DHit is needed for TrajectoryBuilders (all changes can be found with a comment search #cmssw_11_0)

Extra needed modules

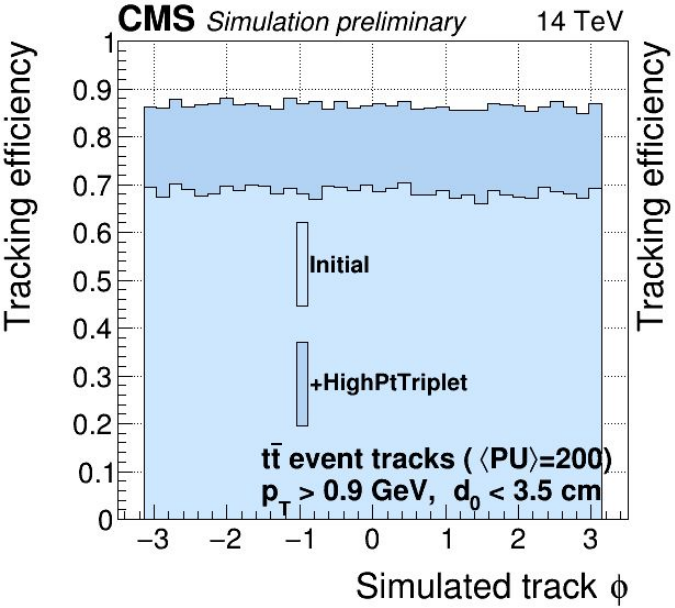
```
process.load("extras_cmssw_11_0_cff")
process.load('Configuration.Geometry.GeometryExtended2026D41Reco_cff')
process.load('Configuration.StandardSequences.MagneticField_cff')
#process.load('Configuration.StandardSequences.RawToDigi_cff')
#process.load('Configuration.StandardSequences.Reconstruction_cff')
#process.load('Configuration.StandardSequences.Validation_cff')
#process.load('DQMServices.Core.DQMStoreNonLegacy_cff')
#process.load('DQMOffline.Configuration.DQMOfflineMC_cff')
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cff')
```

commented out

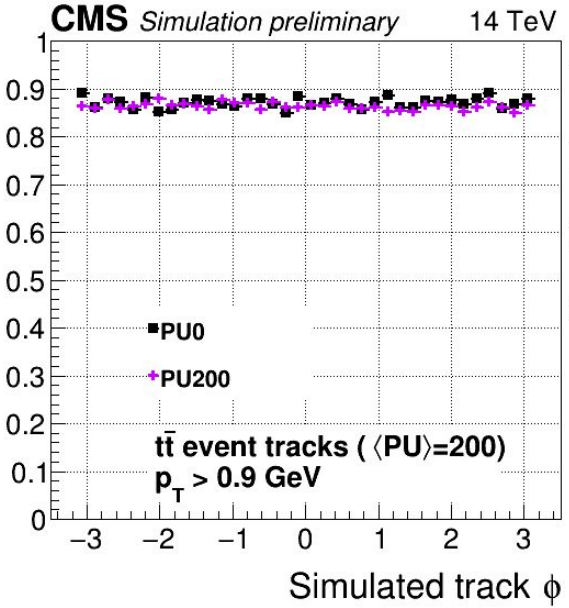
```
process.schedule = cms.Schedule(
*[process.raw2digi_step,process.MC_Tracking_v2,
process.MC_prevalidation_v2,process.MC_validation_v2,
process.MC_Dqmoffline_v2, process.DQMoutput_step])
```

Backup 3

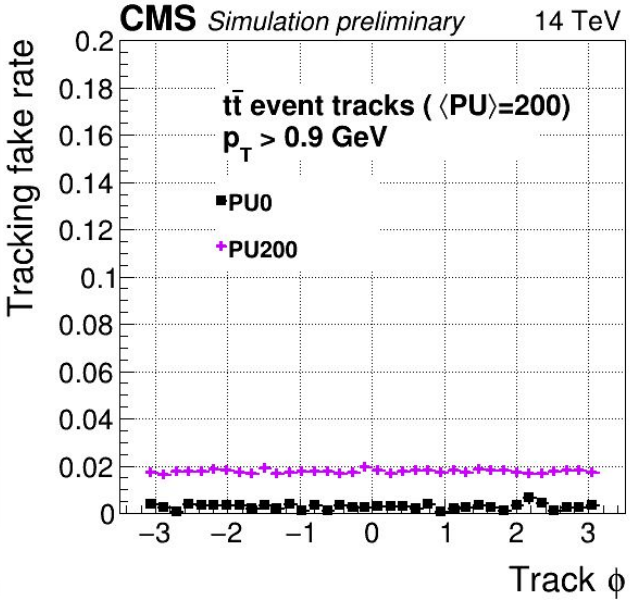
# Adapting MC_Tracking_v2 to HLT - Performance Plots




Efficiency vs Phi - PU200




Efficiency vs PU




Fake Rate vs PU

- Performance seems fine

Backup 4

# Current Phase 2 Reconstruction - Timing

→ For timing studies

Cmssw Timing Job with multithreading
(https://twiki.cern.ch/twiki/bin/viewauth/CMS/TriggerStudiesTiming )

- Running on vocms003/4 machine
- Multithreaded
  - 4 cores

## 4. Executing the CMSSW Timing Job

As with all other hlt jobs you just have to run with cmsRun. There are a few tips for doing a timing study however:

- If running witough multithreading, just execute the cmsRun job as always.

```
cmsRun hlt.py >& full.log&
```

- If you want to compare your results with the timing obtained online, use 4 threads like the online processes do. In the HTL menu configuration scripts set the number of threads to 4

```
# enable TrigReport, TimeReport and MultiThreading
process.options = cms.untracked.PSet(
    wantSummary = cms.untracked.bool( True ),
    numberOfThreads = cms.untracked.uint32( 4 ),
    numberOfStreams = cms.untracked.uint32( 0 ),
)
```

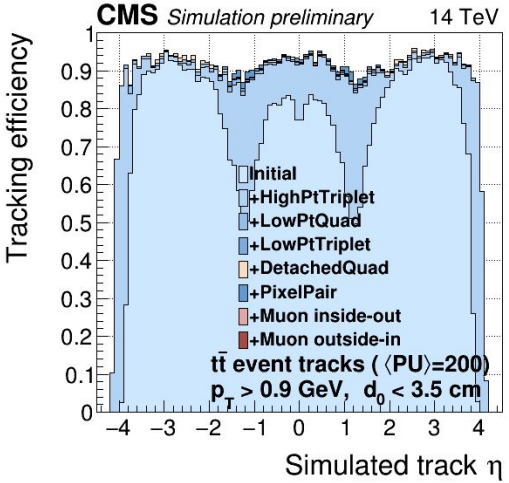
Then start the script and assign the threads to 4 cores. This will assure you run as online (i.e., without hyperthreading).

```
#example assign to run on cores 0,1,2,3
nohup taskset -c 0-3 cmsRun hlt.py >& full.log&
```

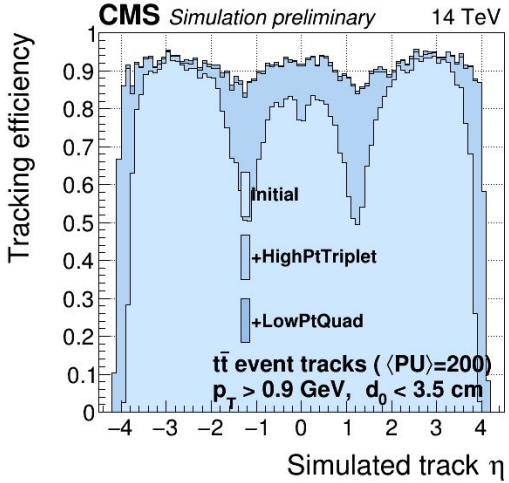
Backup 5

# 3 iterations: Performance - Efficiency vs eta

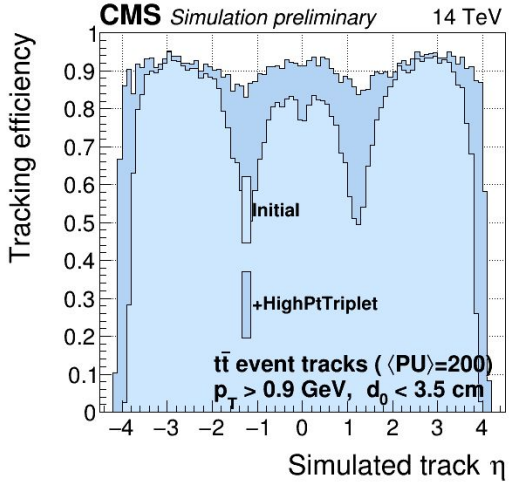
All Iterations




3 Iterations




2 Iterations

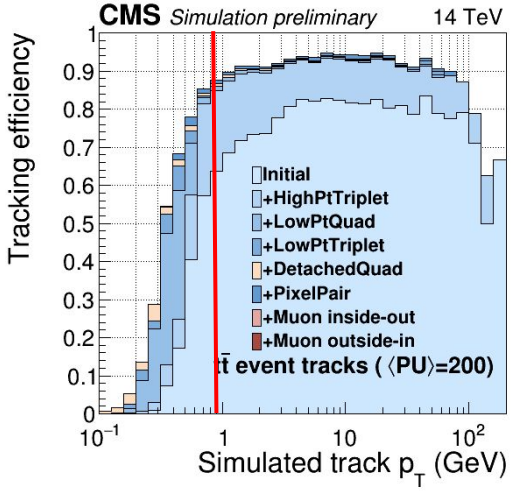



- The efficiency is recovered by the HighPtTriplet step, LowPtQuad does not add much.

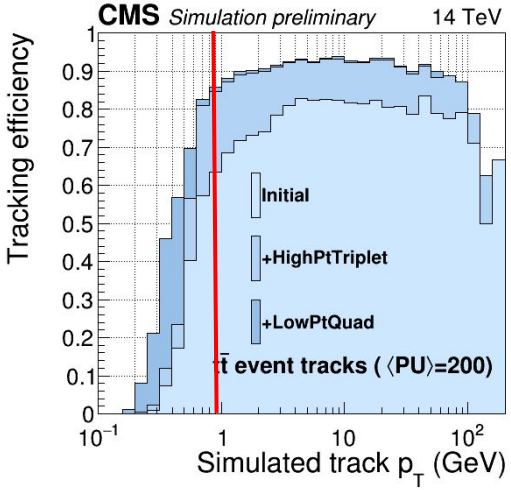
Backup 5

# 3 iterations: Performance - Efficiency vs pt

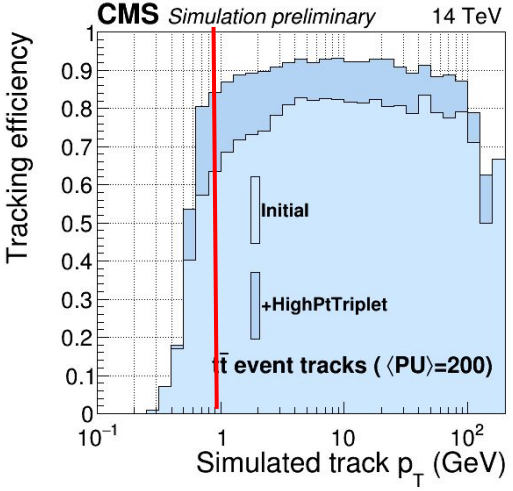
All Iterations




3 Iterations




2 Iterations




- LowPtTriplet, DetachedQuad, PixelPair, and Muon iterations
  - are meant for recovering the efficiency in phase-space which are probably not that relevant at HLT