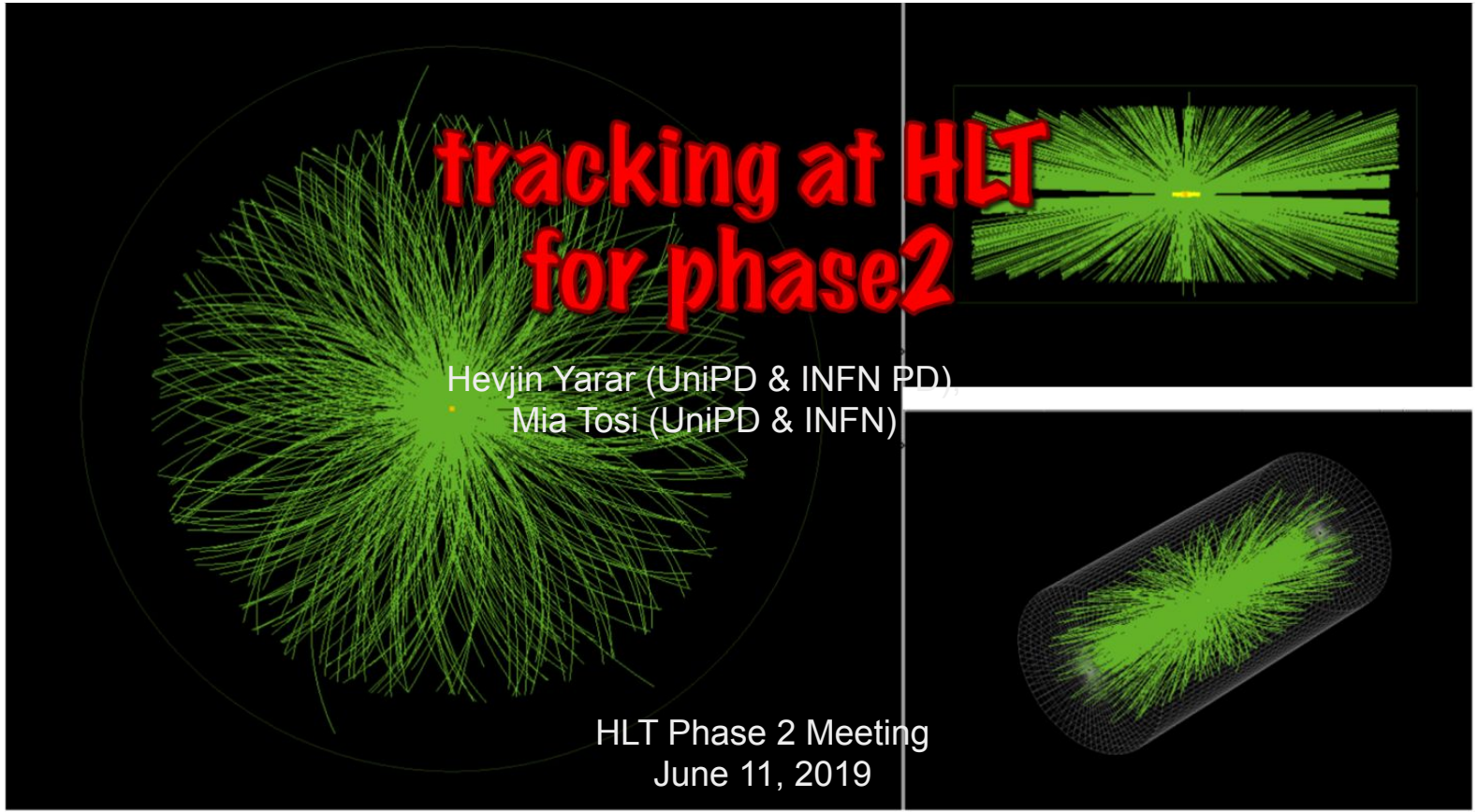
tracking at HLT for phase2

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



HLT Phase 2 Meeting
June 11, 2019

Schedule

- Having a running configuration (DONE)
- Cleaning up the config (ongoing)
- Study of current tracking configuration (on going)
  - Tracking performance (DONE)
  - Timing Study (DONE)
- Optimize track reconstruction (by the end of the Summer)
  - Decrease number of iterations
  - Increase the thresholds
  - Check pixel-only tracks for PV reconstruction
  - Make the steps PV constraint, probably
  - Try seed cleaning
  - Make use of pixelTracks
- Study the impact of using L1 tracks as seeds
- Study the impact of using *Vector Hits*

fishbone, seed cleaner
To be integrated in CMSSW

integrate *Patatrack*
pixel-based developments

instructions for running
the Track Trigger tracks

***Vector Hits* needs still to be**
integrated in CMSSW ….

Current Phase 2 Reconstruction Recipe

> cmsrel CMSSW_10_4_0_mtd5

> cd CMSSW_10_4_0_mtd5/src/ && cmsenv

> runTheMatrix.py -w upgrade -n | grep 2023 | grep trackingOnly

> runTheMatrix.py -w upgrade -l 21224.1 --dryRun

Reconstruction and validation done with:
cmsRun step3_RAW2DIGI_RECO_VALIDATION_DQM.py

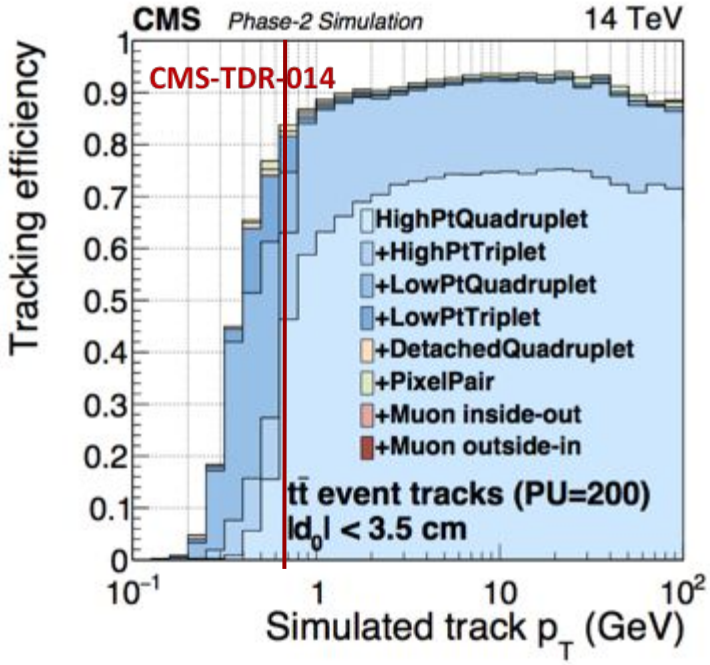
Cleaner version: /afs/cern.ch/user/h/hyarar/public/trking/step3_clean.py

List of TTbar Samples to work with:
PU0:
/store/mc/PhaseIIMTDTDRAutumn18DR/TTbar_TuneCP5_14TeV_pythia8/FEVT/NoPU_103X_upgrade2023_realistic_v2-v1/
PU200:
/store/mc/PhaseIIMTDTDRAutumn18DR/TTbar_14TeV_TuneCP5_Pythia8/FEVT/PU200_103X_upgrade2023_realistic_v2-v1/

In the step3 python script change the process.source from step2.root to the list of the samples:
process=cms.Process(“RECO”) ----> process = cms.Process(“RECOHLT”) #otherwise complains

Current Phase 2 Reconstruction

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

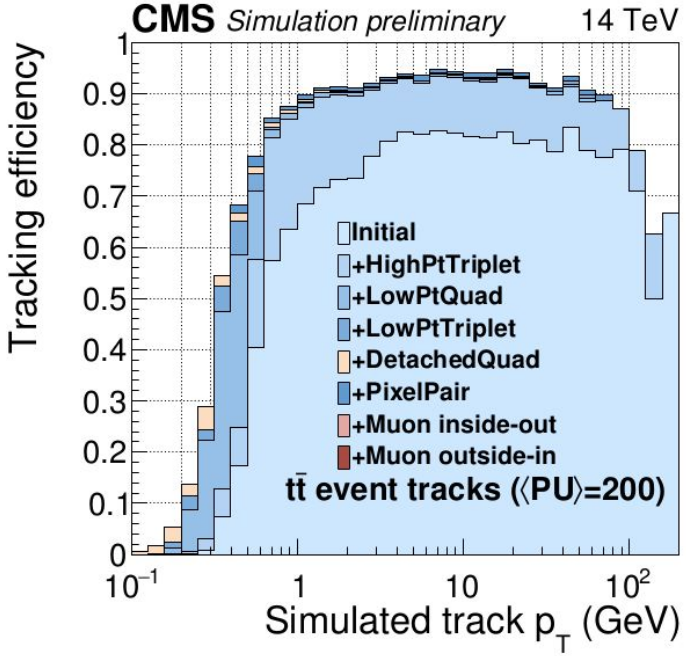


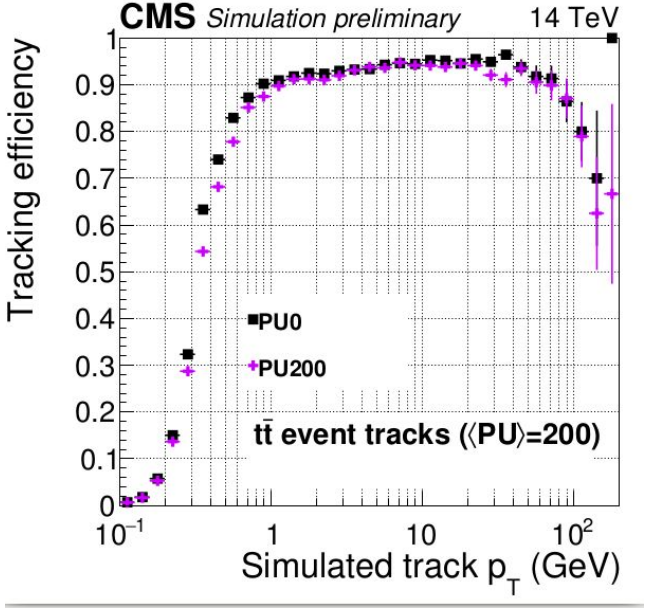
	Step Name	Seeding	Target Tracks
*	HighPtQuad	pixel quadruplets	prompt, high p_T
*	HighPtTriplet	pixel triplets	prompt, high p_T, recovery
*	LowPtQuad	pixel quadruplets	prompt, low p_T
	LowPtTriplet	pixel triplets	prompt, low p_T, recovery
	DetachedQuad	pixel quadruplets	displaced
*	PixelPair	pixel pairs	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

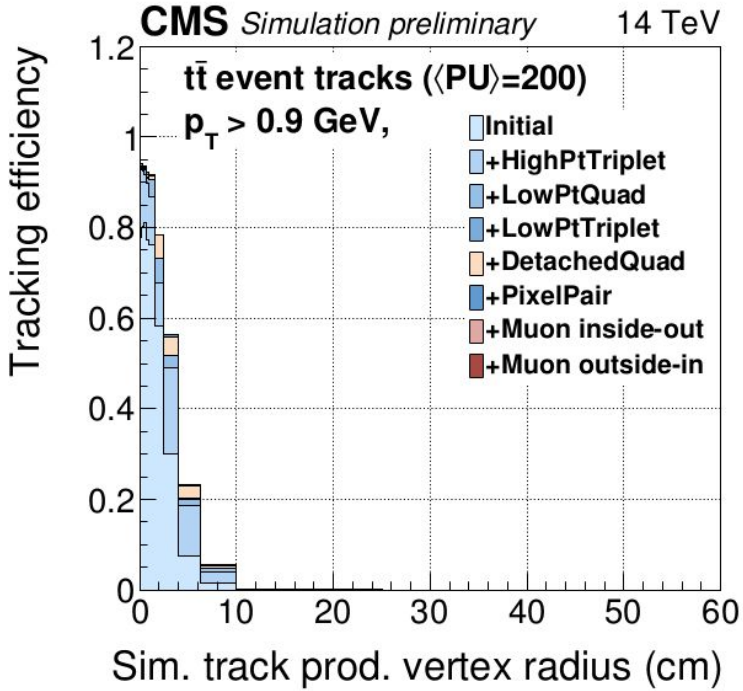
Baseline Performance - Efficiency





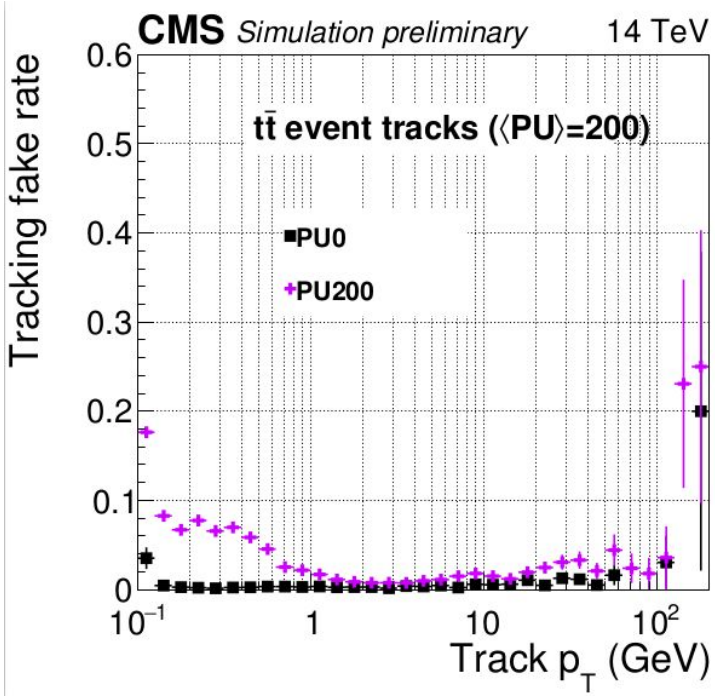
- LowPtTriplet, DetachedQuad, PixelPair, Muon iterations do not add much to the efficiency.
- With increasing PU we lose few % of efficiency in [0.3 - 1] GeV range.

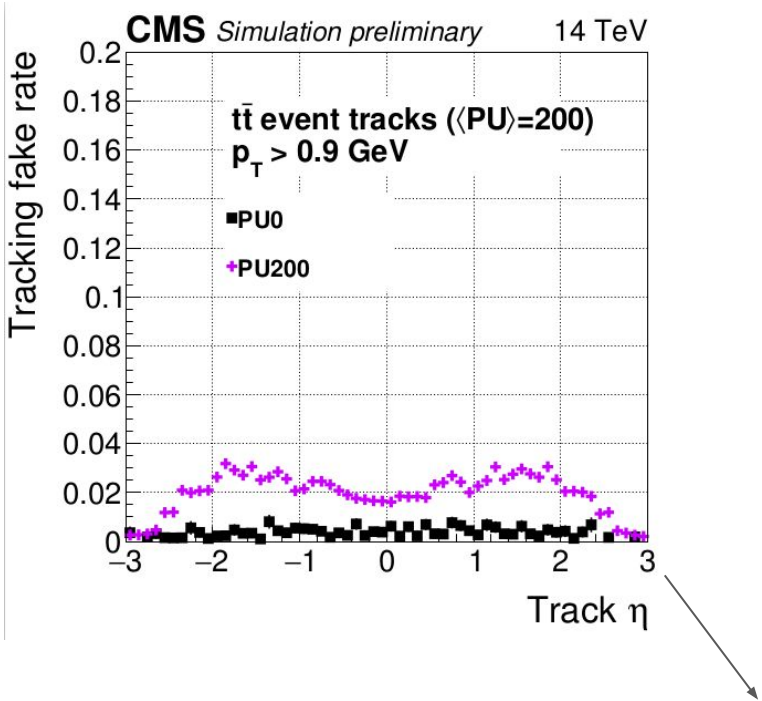
Baseline Performance - Efficiency



- *DetachedQuad* provides few % efficiency for slightly displaced tracks
- displaced tracks are still missing
  → *VectorHits* needs to be integrated in CMSSW, and it will be used for outer tracker seeded steps

Baseline Performance - Fake Rate





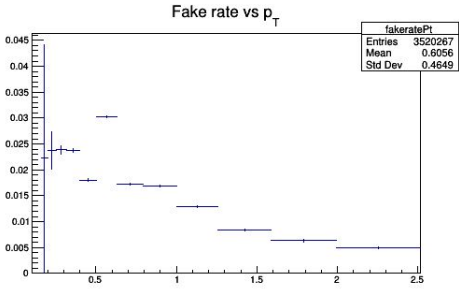
- as expected, with the increase of PU, the fake rate increases
  - above all at very low pT
  - marginally even at high pT

To be updated, eta should go up to 4.

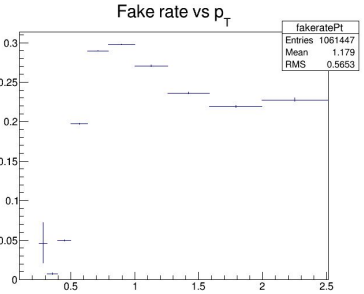
Baseline Performance - Fake Rate per iteration

Tracks (after fitting and selection)

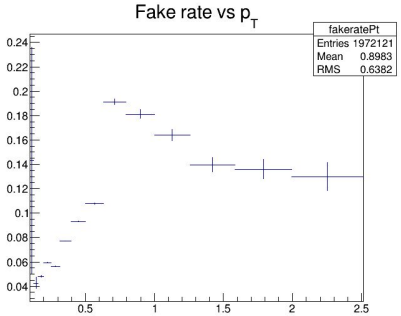
InitialStep



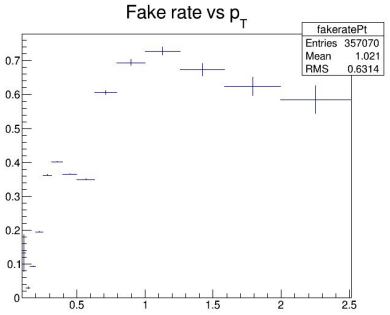
HighPtTriplet



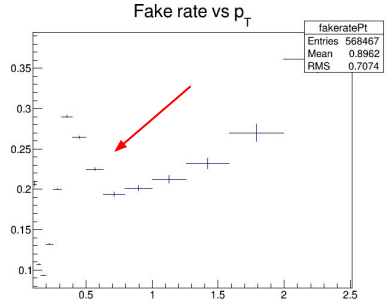
LowPtQuad



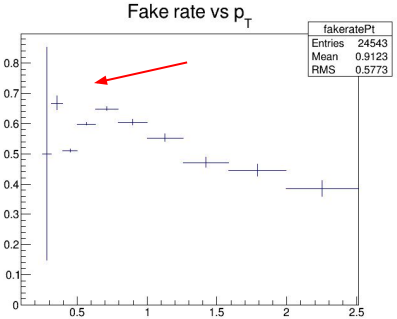
LowPtTriplet



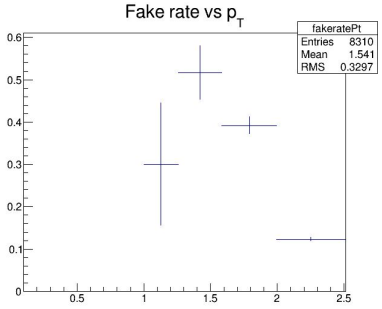
DetachedQuad



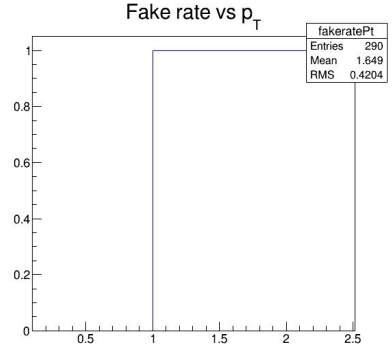
PixelPair



Muon inside-out



Muon outside-in



Timing

Running on

- vocms004 machine
- 1k events locally (in /data/user/tosi/)
- TTbar_14 events (PhaseIIMTDTDRAutumn18DR campaign: CMSSW_10_4_0_mtd5, 103X_upgrade2023_realistic_v2 ,)
- Multithreaded (otherwise jobs take forever !)
  - 16 cores

```
process.options.numberOfStreams = cms.untracked.uint32(16)
process.options.numberOfThreads = cms.untracked.uint32(16)
```

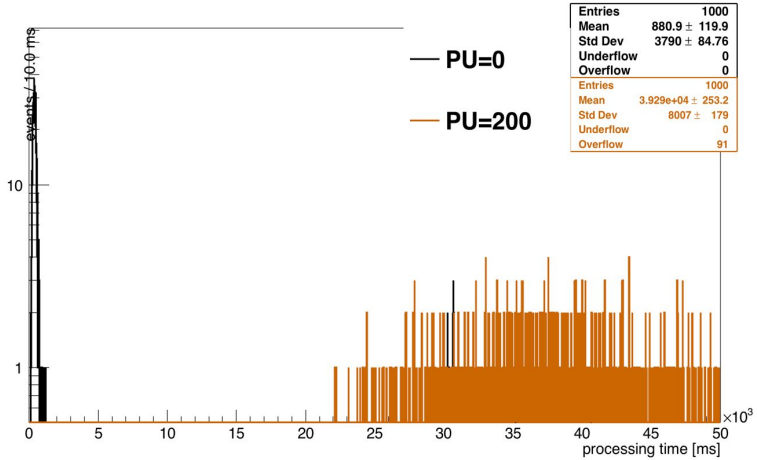
we ran w/ only 8 cores as well, but still to check ...

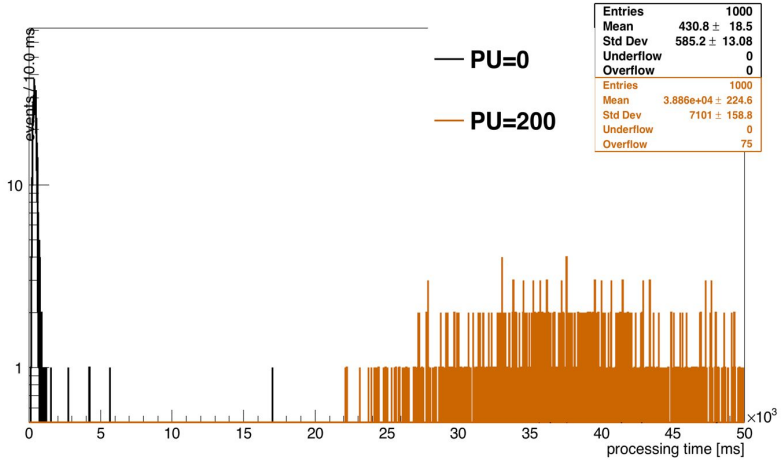
Timing **x44 !**

	process <timing> per event [ms]	tracking path <timing> per event [ms]
<PU>=0	900	450 (50 %)
<PU>=200	39 300	38 900 (99 %)

as expected,
loading conditions and source files is

- independent of PU
- negligible at <PU>=200



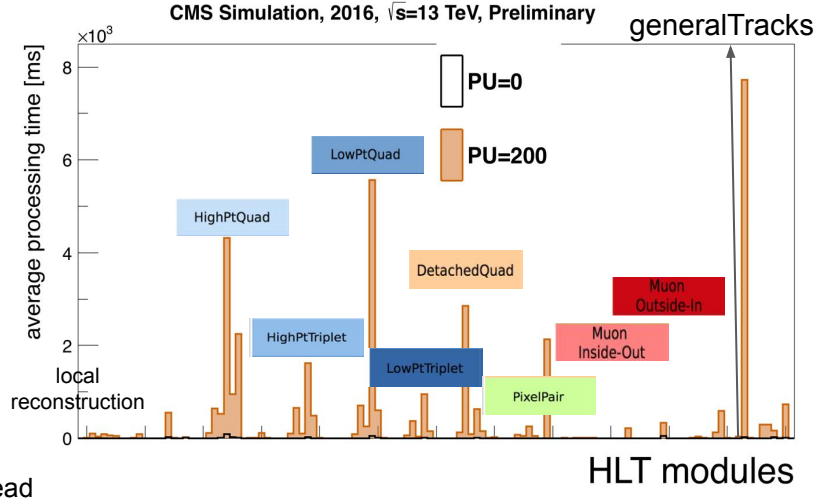


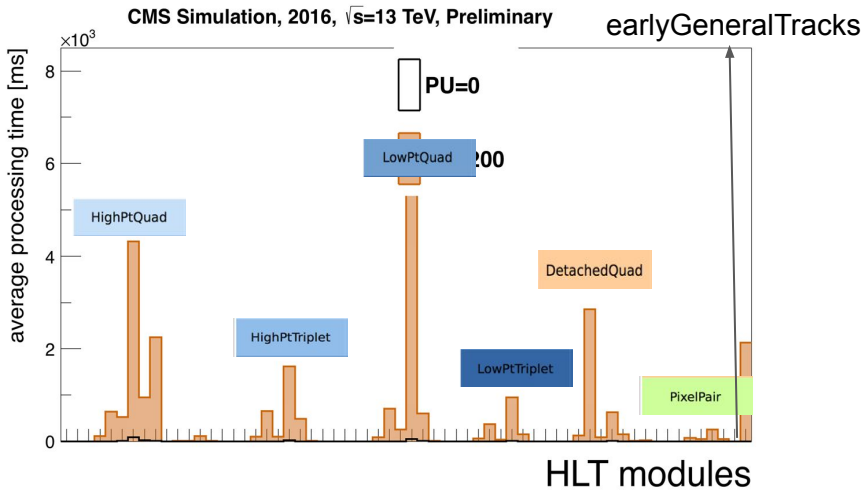
with the new samples at different PU scenarios, we will be able to check the PU dependence

per thread

Timing **x100 !!!**

	process <timing> per event [ms]	tracking path <timing> per event [ms]	tracking-only modules <timing> per event [ms]	inner tracking-only modules <timing> per event [ms]
<PU>=0	900	450 (50 %)	350 (78 %)	260 (58 %)
<PU>=200	39 300	38 900 (99 %)	28 600 (74%)	25 000 (64 %)





per thread

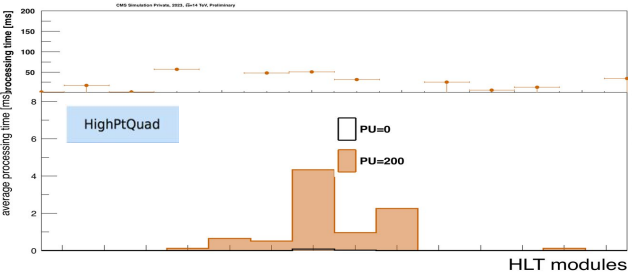
iterations

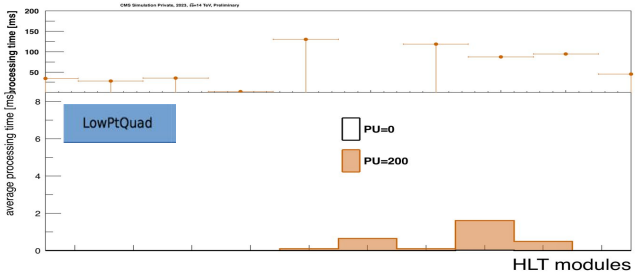
we need to squeeze
as much as possible these numbers !

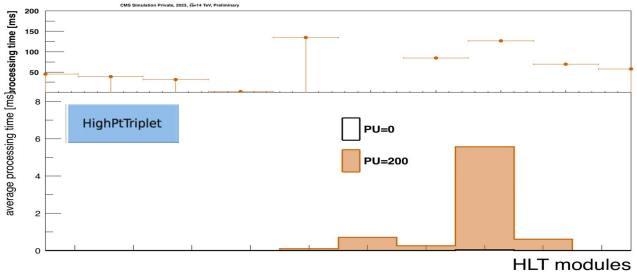
	efficiency [%]		fakerate [%]			<timing> per event [ms]		
	<PU>=0	<PU>=200	<PU>=0	<PU>=200		<PU>=0	<PU>=200	
HighPtQuad	76	74	0.25	1.8	x7	130*	6600*	x50
HighPtTriplet	15	15	0.7	25	x36	26	3000	x115
LowPtQuad	0.8	0.9	0.3	7.8	x26	60	7200	x120
LowPtTriplet	0.05	0.09	1	35	x35	9	1600	x180
DetachedQuad	0.6	0.5	0.7	17	x24	11	3900	x350
PixelPair	1	1	2	55	x27	5	440	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

iterations



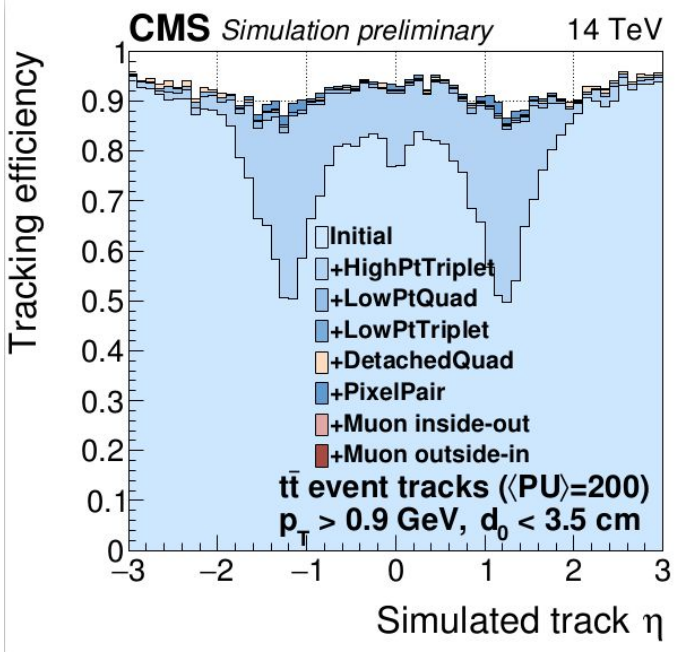


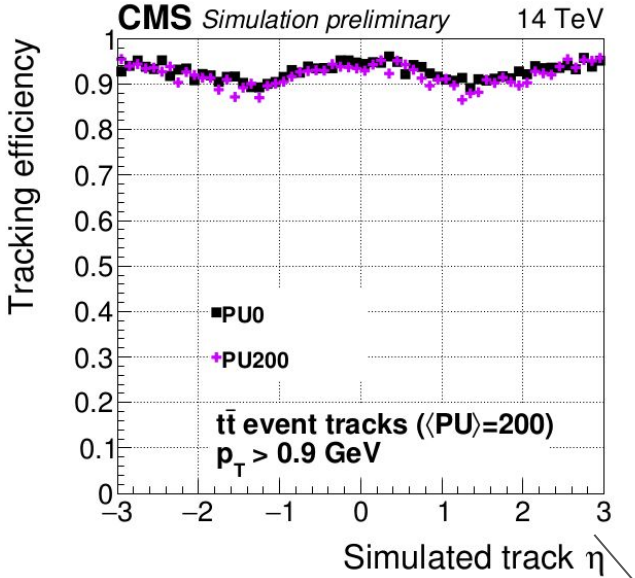


we have to squeeze the timing

- limit the amount of fakes (and duplicates) already at seeding level
- limit the amount of candidates in the pattern recognition step

Backup 1
Baseline Performance - Efficiency

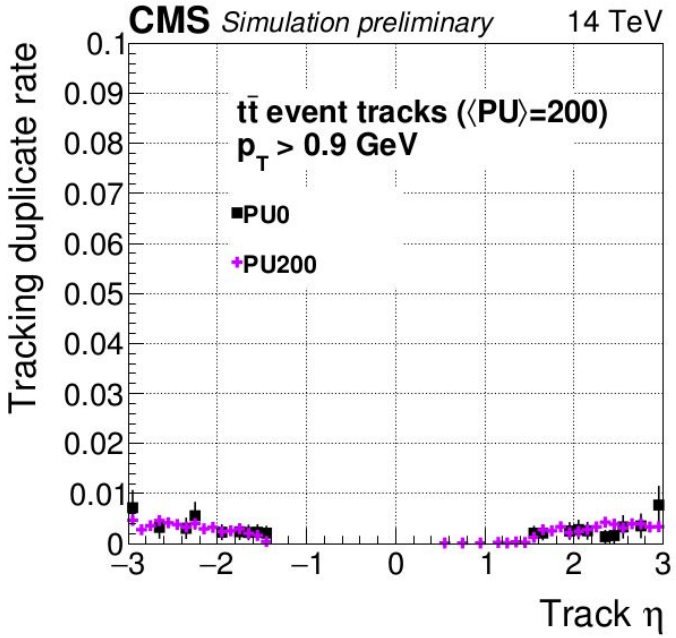




- Efficiency drop with PU at eta ~ 1 (?)

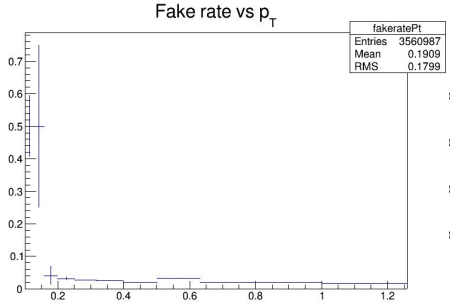
To be updated, eta should go up to 4.

Backup 2
Baseline Performance - Duplicate Rate

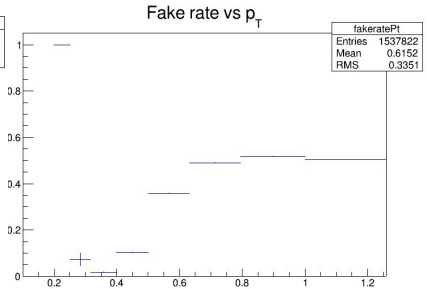


Backup 3
Baseline Performance - Fake Rate per Iteration for Track Building
Track Candidates

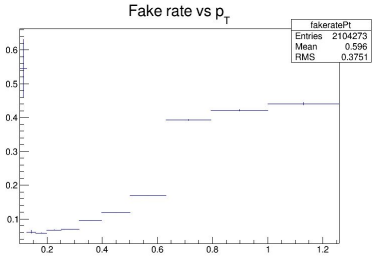
InitialStep



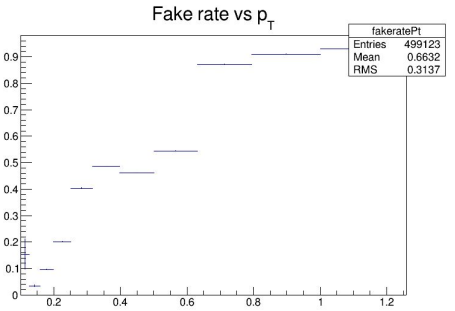
HighPtTriplet



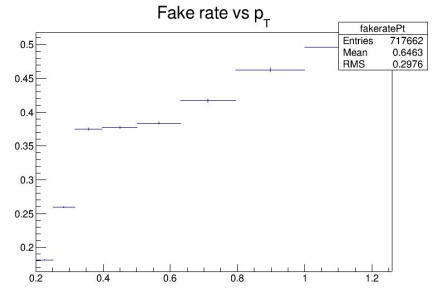
LowPtQuad



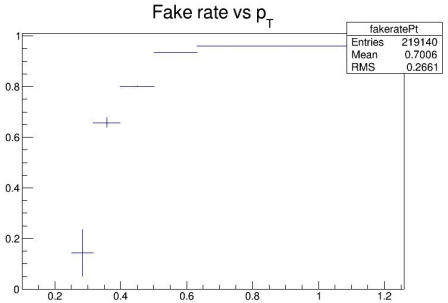
LowPtTriplet



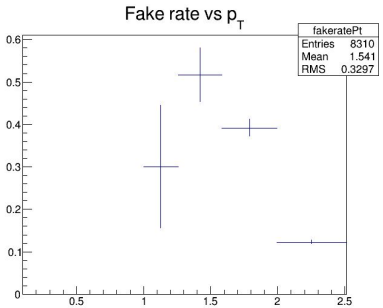
DetachedQuad



PixelPair



Muon inside-out



Muon outside-in