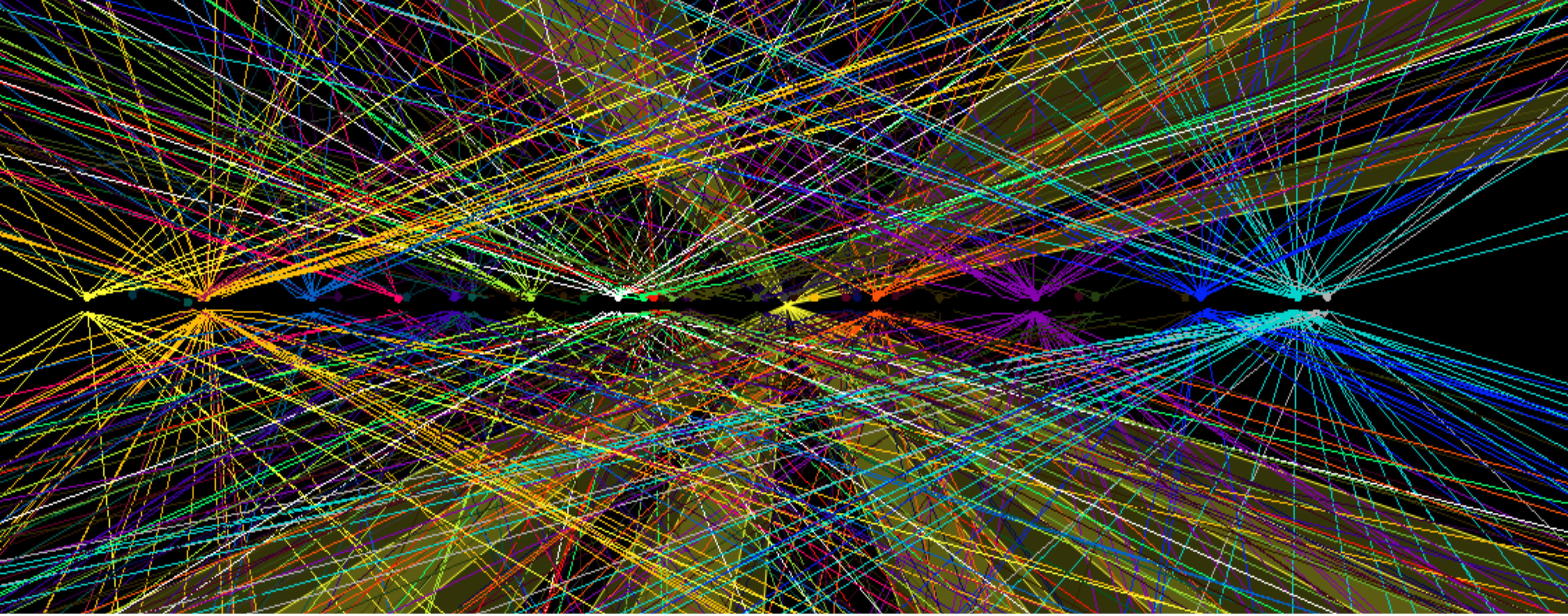


TRK-POG
ready for 2017 (?)

Operation Tasks: development(?)

- The new pixel should be completely transparent to any task using Tracks and Vertices (PixelTracks included)
  - At most more stringent quality criteria can be set requiring for instance one more pixel-hit
    - Reminder: also no dyn-ineff anymore!
  - The solution of the VFP issue should also allow to be more strict in the strip as well, if needed

Operation Tasks: who

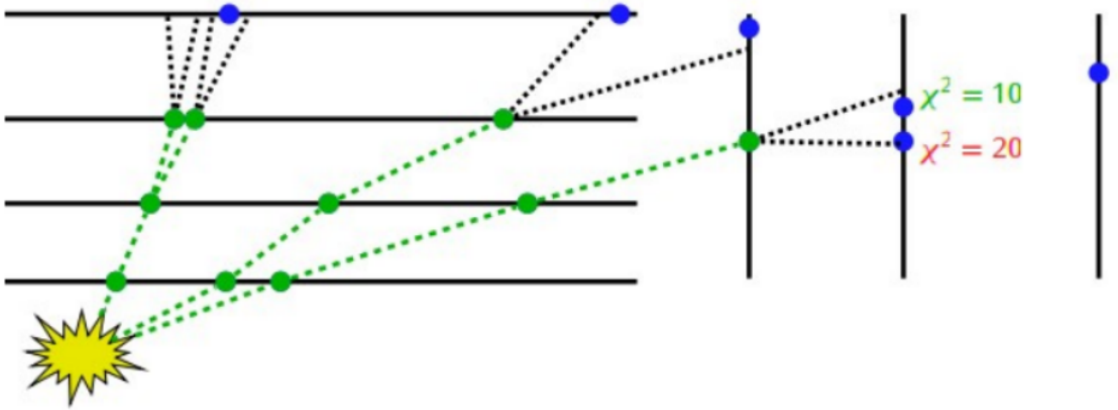
- Online Beamspot:
  - Vacant, asked collaboration to help
- Offline Beamspot:
  - Covered
- Validation (data/MC)
  - Covered
- Efficiency/Resolution
  - Covered
- DQM
  - Covered

What new in tracking?

- Essentially a completely new seeding "step"
  - New algorithms
  - New framework
- Still discovering issues inherited from the far past
  - Clean solutions will not appear before the summer

Triplet Propagation

Propagate 1-2-3 triplet to 4th layer and search for compatible hits using a fast algorithm



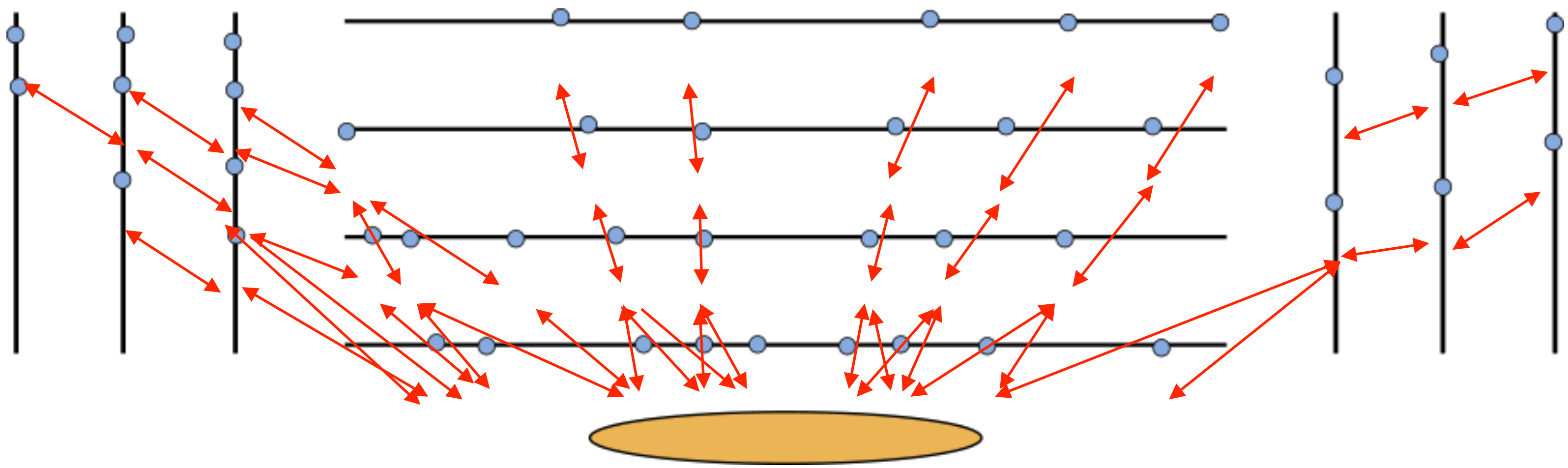
Natural continuation of the current approach from pairs to triplets

Variant: "Pixel seed extension"

- 0 code development (0 innovation, likely 0 "regression w/r/t phase0)
- Seeded by triplets from layers 1-2-3
- Use Kalman filter for the propagation
- In pattern recognition, stop trajectory propagation if no 4th pixel hit right after the seed

Cellular Automaton (CA)

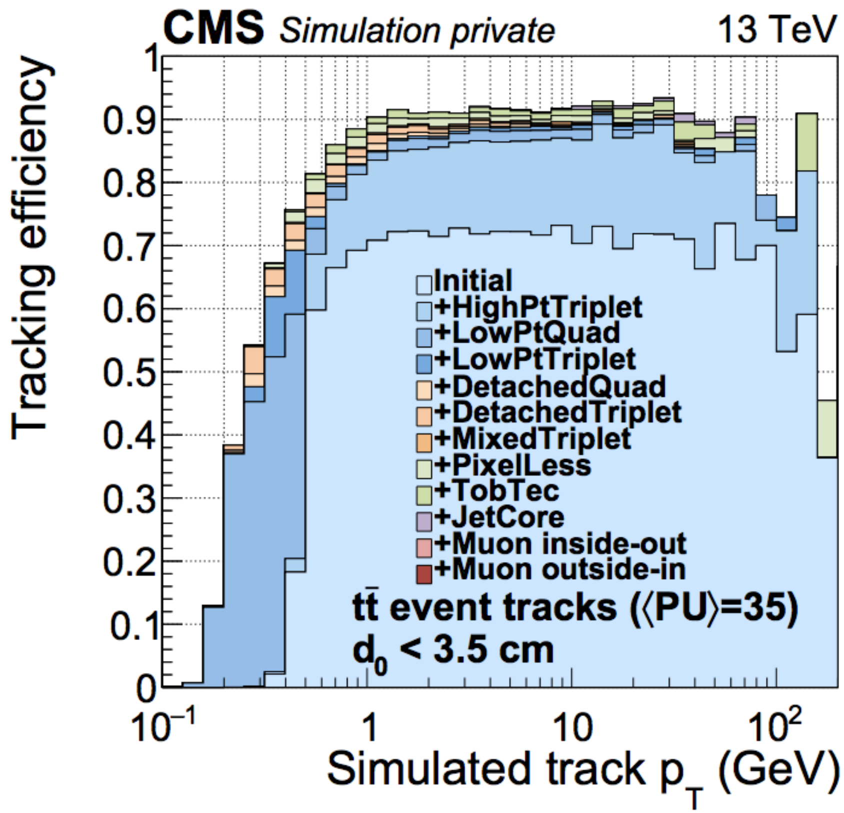
- The CA is a track seeding algorithm designed for parallel architectures
- It requires a list of layers and their pairings
  - A graph of all the possible connections between layers is created
  - Doublets aka Cells are created for each pair of layers (compatible with a region hypothesis)
  - Fast computation of the compatibility between two connected cells
  - No knowledge of the world outside adjacent neighboring cells required, making it easy to parallelize



Tracking for Phase1 pixel

M. Kortelainen

step name	seeding	target track
Initial	pixel quadruplets[3]	prompt, high p_T
LowPtQuad	pixel quadruplets[2]	prompt, low p_T
HighPtTriplet	pixel triplets	prompt, high p_T recovery
LowPtTriplet	pixel triplets	prompt, low p_T recovery
DetachedQuad	pixel quadruplets[2]	displaced−−
DetachedTriplet	pixel triplets	displaced−− recovery
MixedTriplet	pixel+strip triplets	displaced−
PixelLess	inner strip triplets	displaced+
TobTec	outer strip triplets	displaced++
JetCore	pixel pairs in jets	high p_T jet
Muon inside-out	muon-tagged tracks	muon
Muon outside-in	standalone muon	muon



2) Triplet propagation

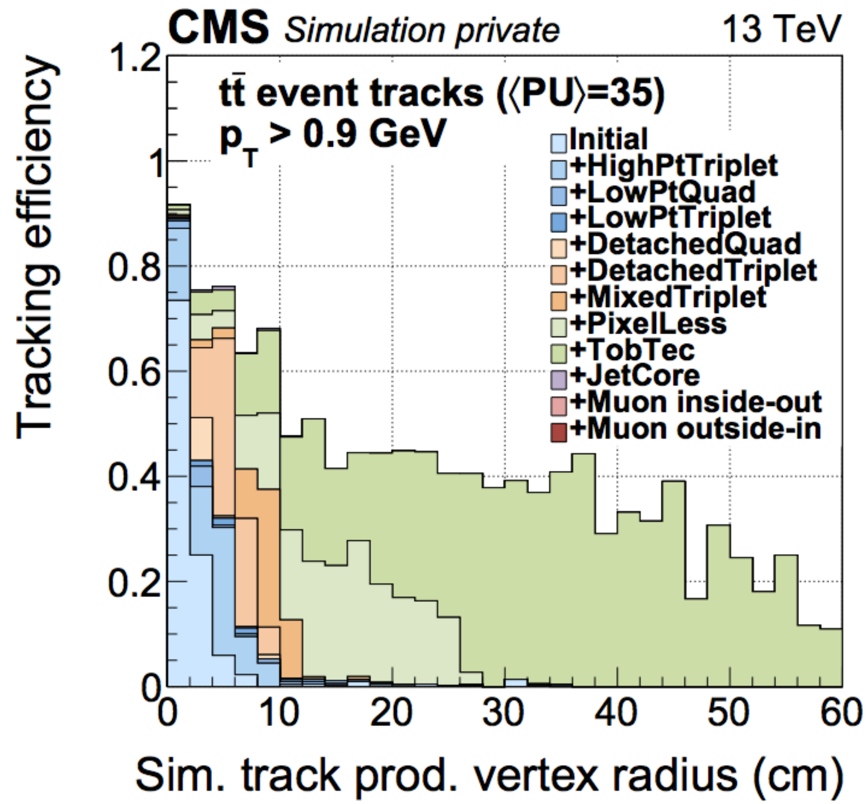
3) Pixel seed extension

- Currently using 2016 track selection MVA out of the box
  - With cut-based selection for HighPtTriplet and LowPtTriplet
- MVA retraining for 2017 almost finished
  - Expected to be finalized by first week of February

Tracking for Phase1 pixel

M. Kortelainen

step name	seeding	target track
Initial	pixel quadruplets 3)	prompt, high p_T
LowPtQuad	pixel quadruplets 2)	prompt, low p_T
HighPtTriplet	pixel triplets	prompt, high p_T recovery
LowPtTriplet	pixel triplets	prompt, low p_T recovery
DetachedQuad	pixel quadruplets 2)	displaced−−
DetachedTriplet	pixel triplets	displaced−− recovery
MixedTriplet	pixel+strip triplets	displaced−
PixelLess	inner strip triplets	displaced+
TobTec	outer strip triplets	displaced++
JetCore	pixel pairs in jets	high p_T jet
Muon inside-out	muon-tagged tracks	muon
Muon outside-in	standalone muon	muon



2) Triplet propagation
3) Pixel seed extension

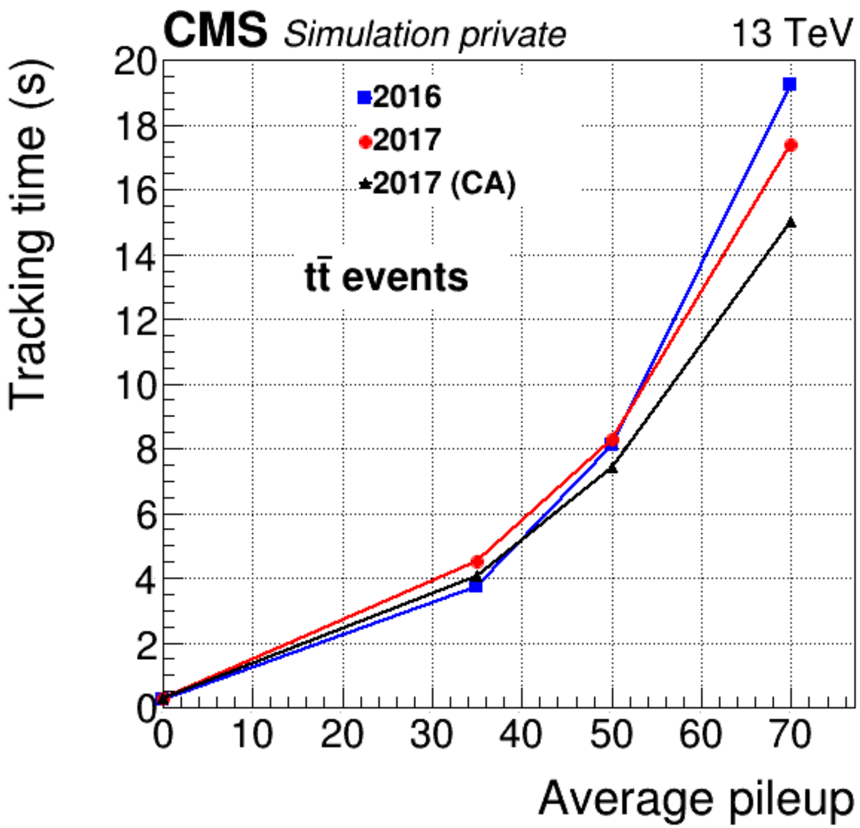
- Currently using 2016 track selection MVA out of the box
  - With cut-based selection for HighPtTriplet and LowPtTriplet
- MVA retraining for 2017 almost finished
  - Expected to be finalized by first week of February

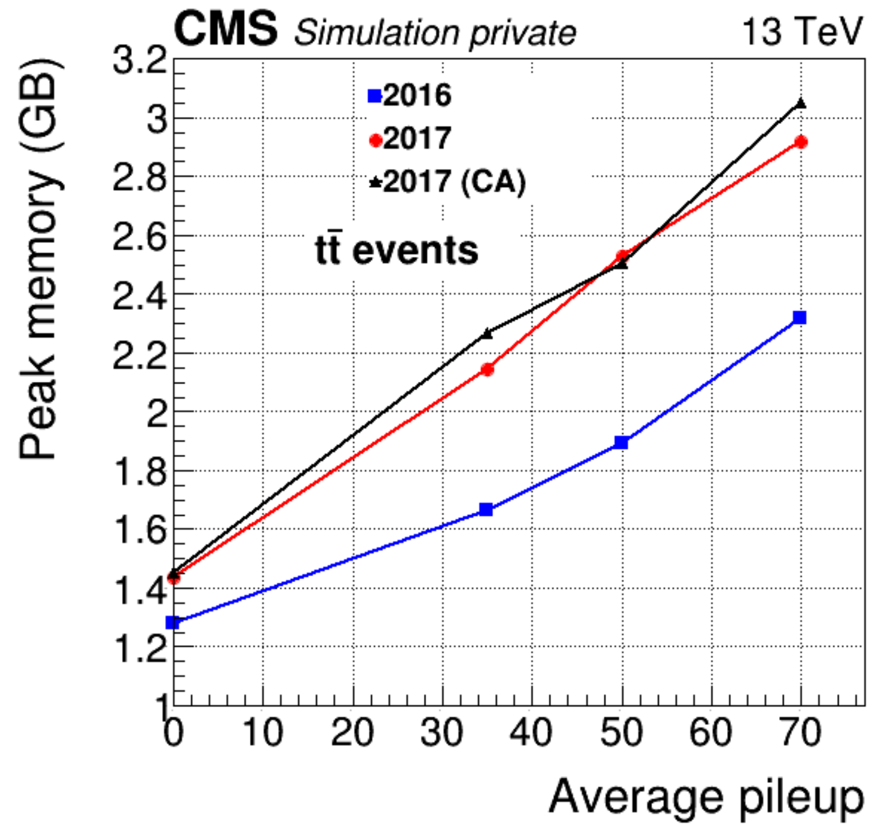
Mostly for reference

PERFORMANCE

Time & Memory

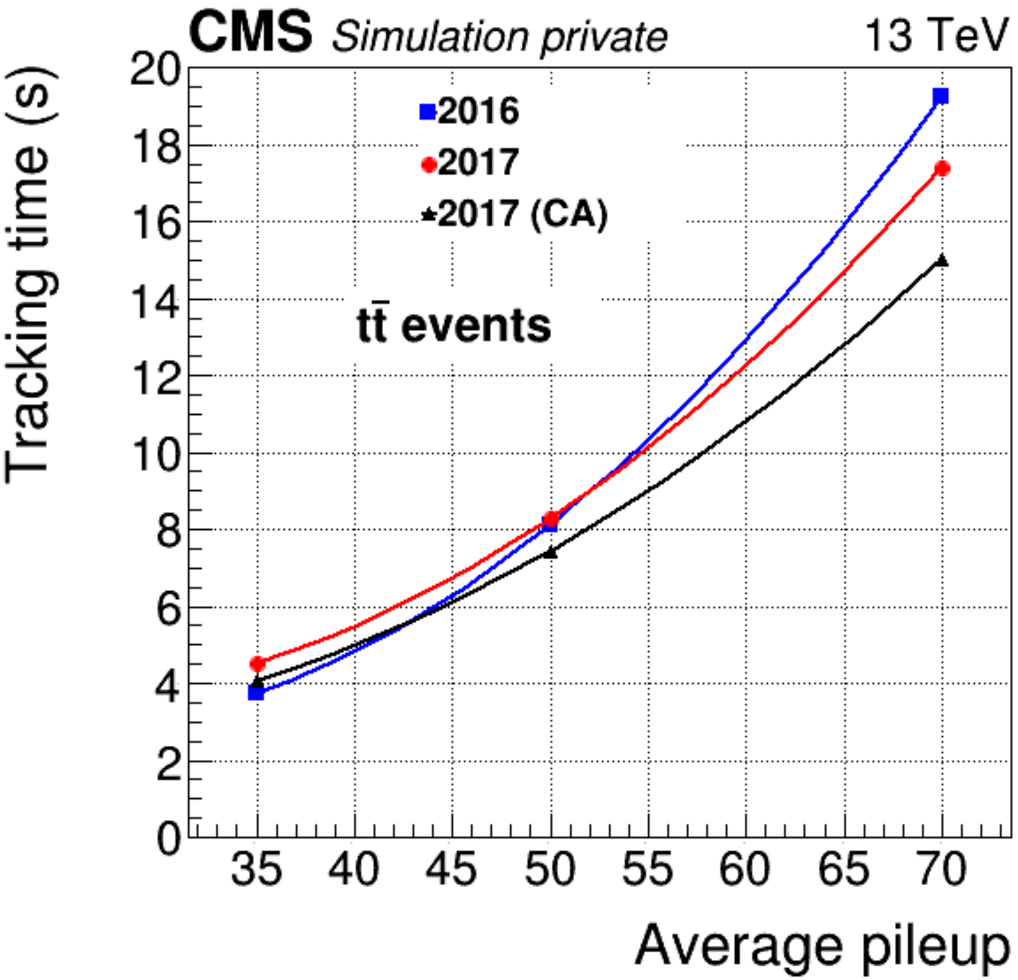
4 Threads, tracking only,No Validation,DQM,Output

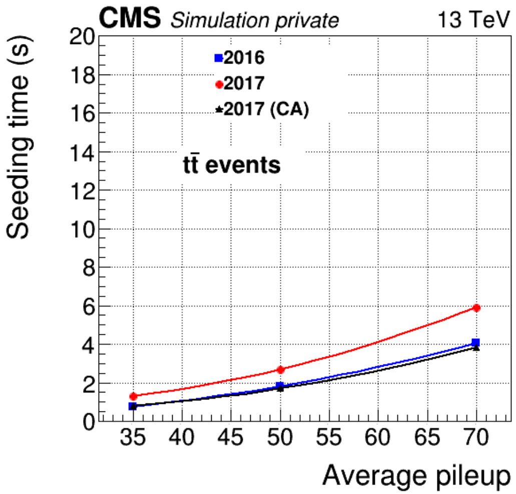


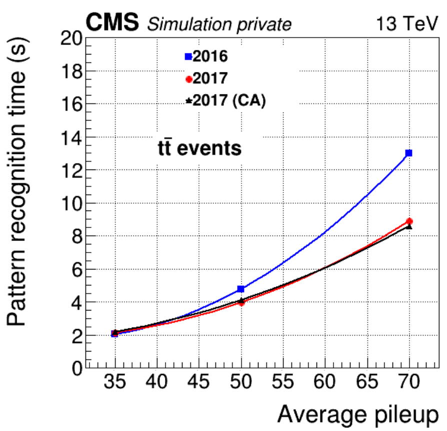


Despite an increase in seeding time, the higher accuracy of quadruplets eventually reduces combinatorics in pattern-recognition producing a significant gain in timing. No free lunch though: memory increase is sizable and action to reduce the memory footprint shall be welcome

Timing

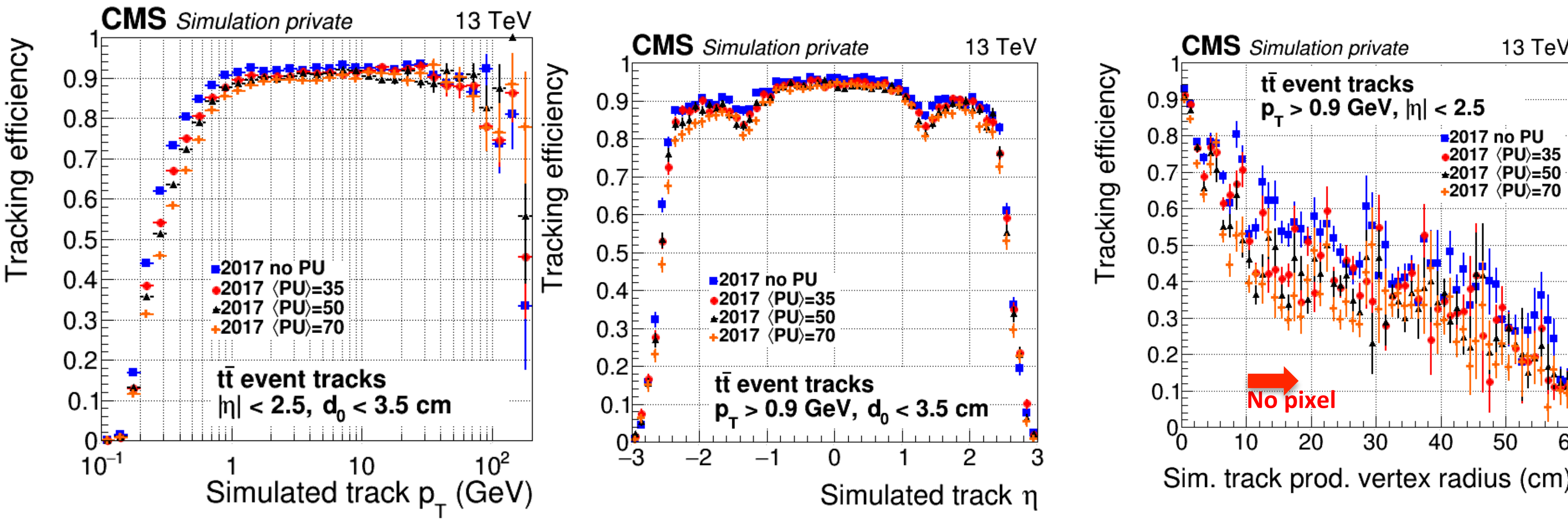




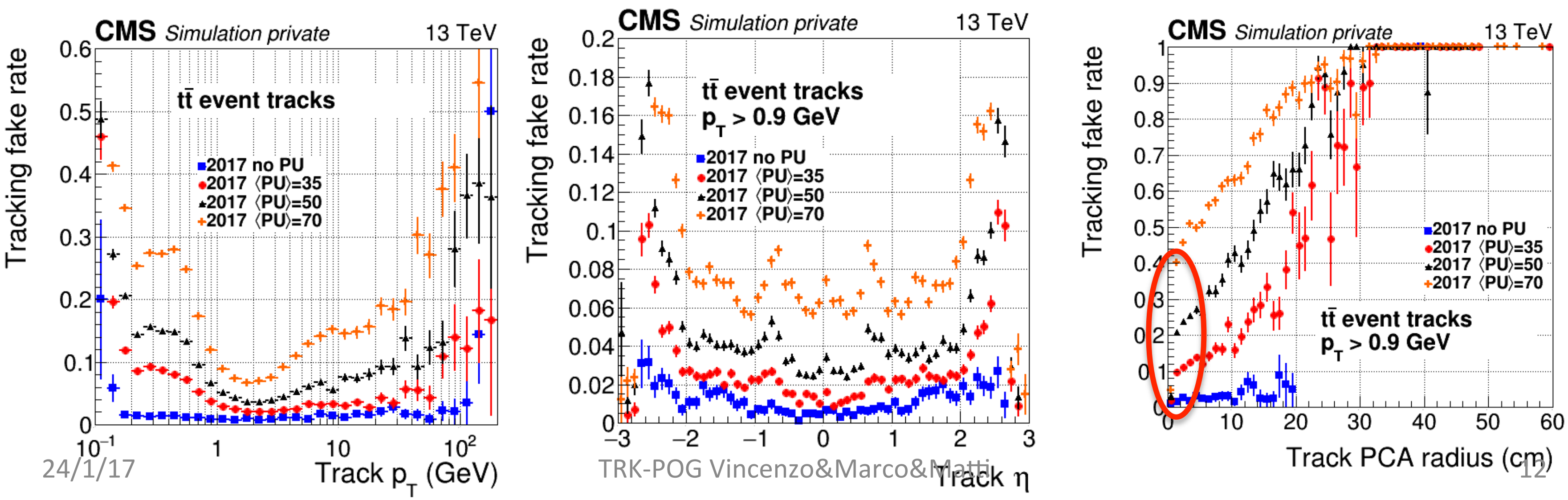


As PU increase CA become advantageous

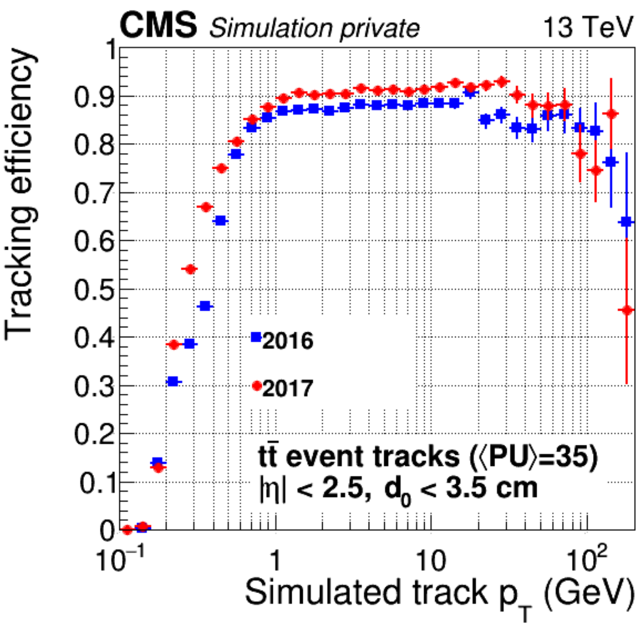
Performance : vs PU

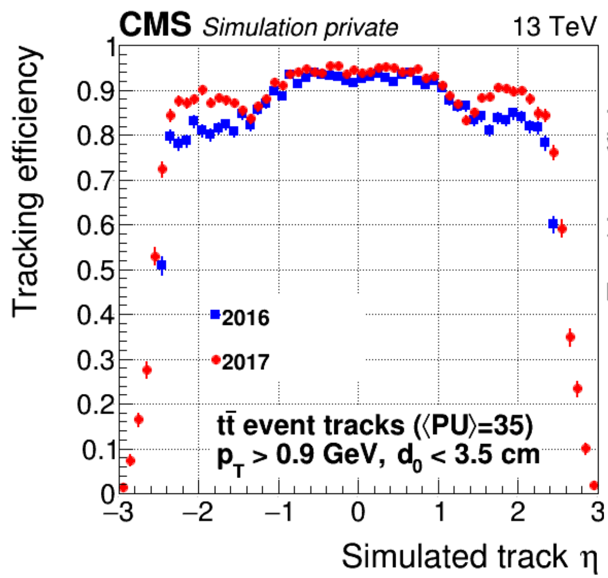


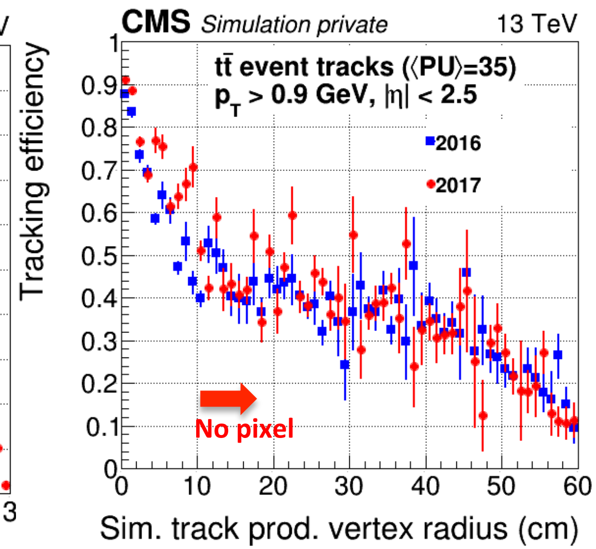
Efficiency ok, fake rate starts to explode for PU>50, in particular for R>1cm

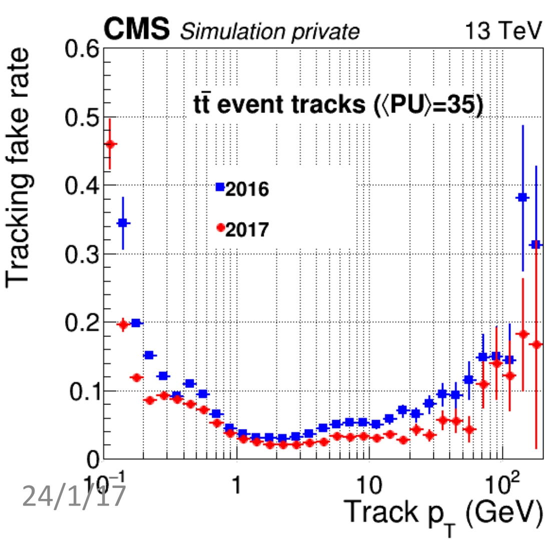


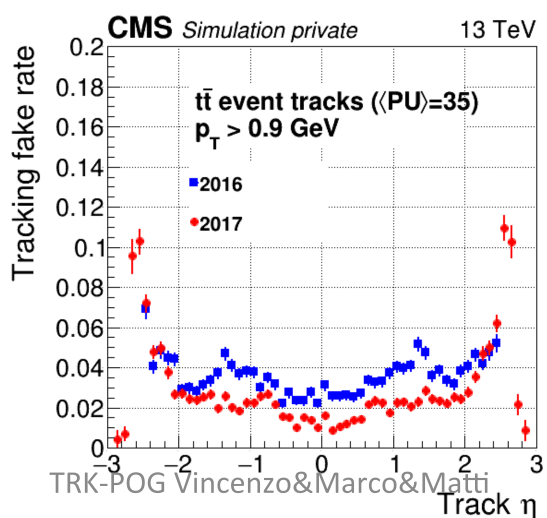
vs 2016

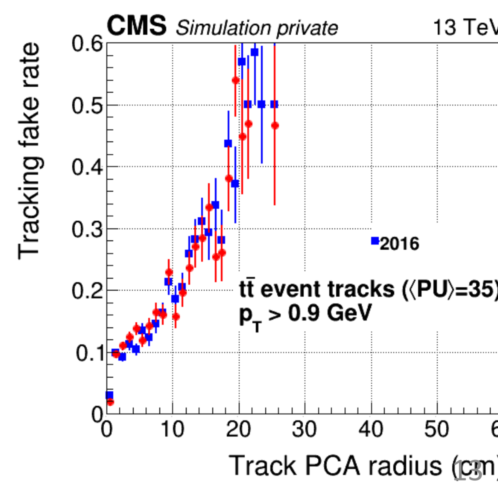

CMS Simulation private 13 TeV
Tracking efficiency
2016
2017
tt̄ event tracks (⟨PU⟩=35)
|η| < 2.5, d_0 < 3.5 cm
Simulated track p_T (GeV)


CMS Simulation private 13 TeV
Tracking efficiency
2016
2017
tt̄ event tracks (⟨PU⟩=35)
p_T > 0.9 GeV, d_0 < 3.5 cm
Simulated track η

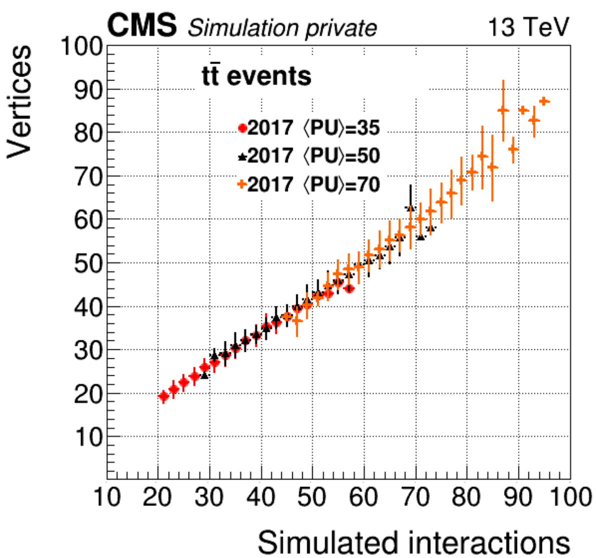

CMS Simulation private 13 TeV
Tracking efficiency
tt̄ event tracks (⟨PU⟩=35)
p_T > 0.9 GeV, |η| < 2.5
2016
2017
No pixel
Sim. track prod. vertex radius (cm)

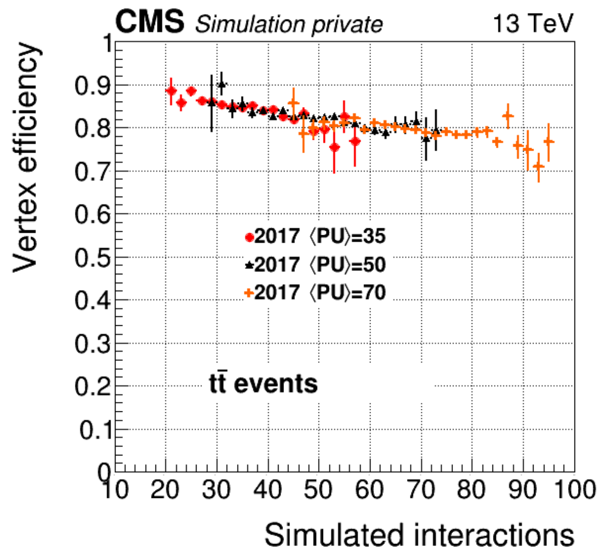

CMS Simulation private 13 TeV
Tracking fake rate
tt̄ event tracks (⟨PU⟩=35)
2016
2017
Track p_T (GeV)

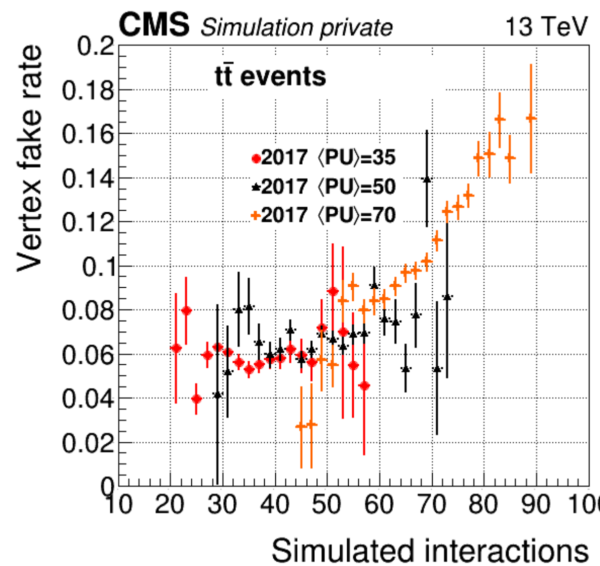

CMS Simulation private 13 TeV
Tracking fake rate
tt̄ event tracks (⟨PU⟩=35)
p_T > 0.9 GeV
2016
2017
Track η

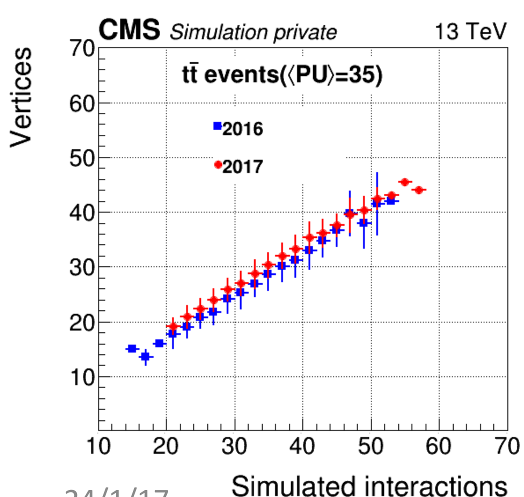

CMS Simulation private 13 TeV
Tracking fake rate
2016
tt̄ event tracks (⟨PU⟩=35)
p_T > 0.9 GeV
Track PCA radius (cm)

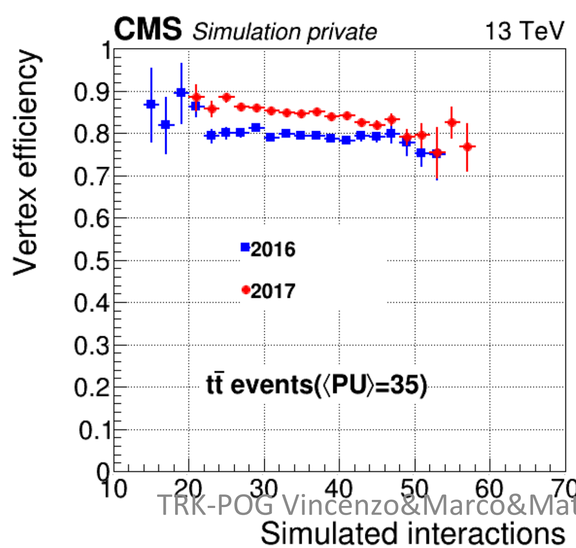
Vertex

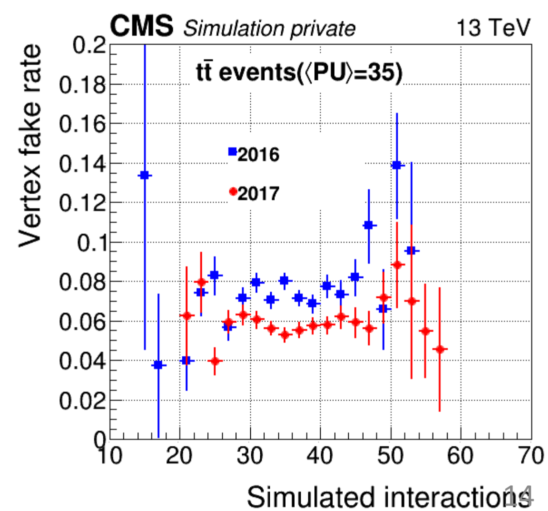

CMS Simulation private 13 TeV
tt̄ events
2017 ⟨PU⟩=35
2017 ⟨PU⟩=50
2017 ⟨PU⟩=70
Vertices
Simulated interactions


CMS Simulation private 13 TeV
2017 ⟨PU⟩=35
2017 ⟨PU⟩=50
2017 ⟨PU⟩=70
tt̄ events
Vertex efficiency
Simulated interactions

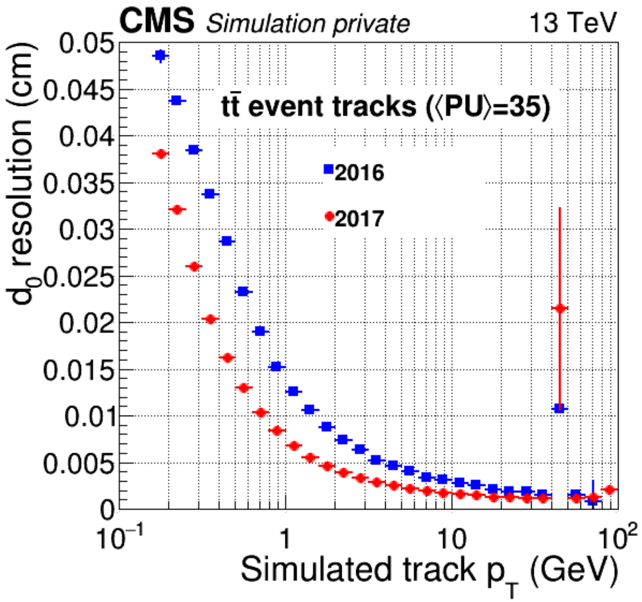

CMS Simulation private 13 TeV
tt̄ events
2017 ⟨PU⟩=35
2017 ⟨PU⟩=50
2017 ⟨PU⟩=70
Vertex fake rate
Simulated interactions

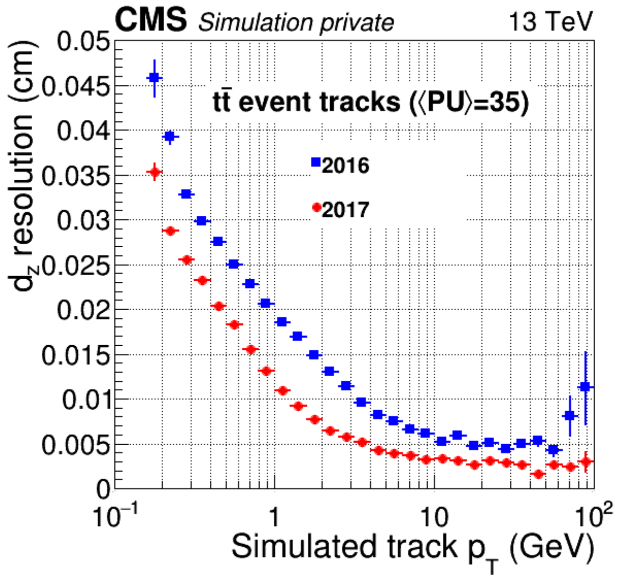

CMS Simulation private 13 TeV
tt̄ events(⟨PU⟩=35)
2016
2017
Vertices
Simulated interactions

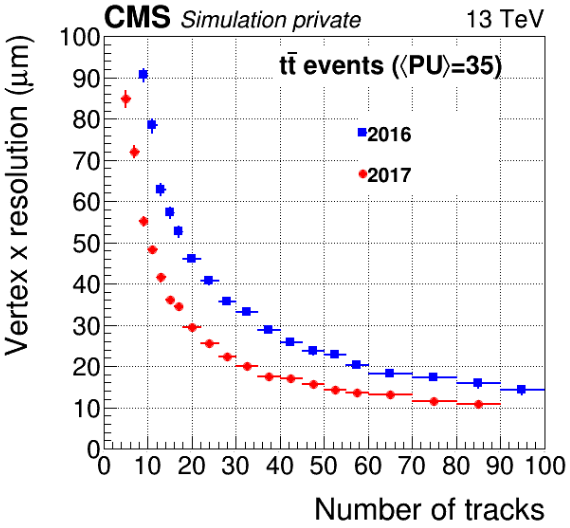

CMS Simulation private 13 TeV
2016
2017
tt̄ events(⟨PU⟩=35)
Vertex efficiency
Simulated interactions

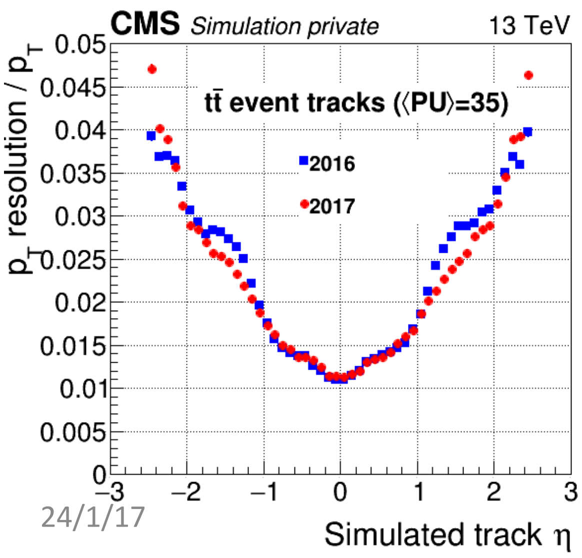

CMS Simulation private 13 TeV
tt̄ events(⟨PU⟩=35)
2016
2017
Vertex fake rate
Simulated interactions

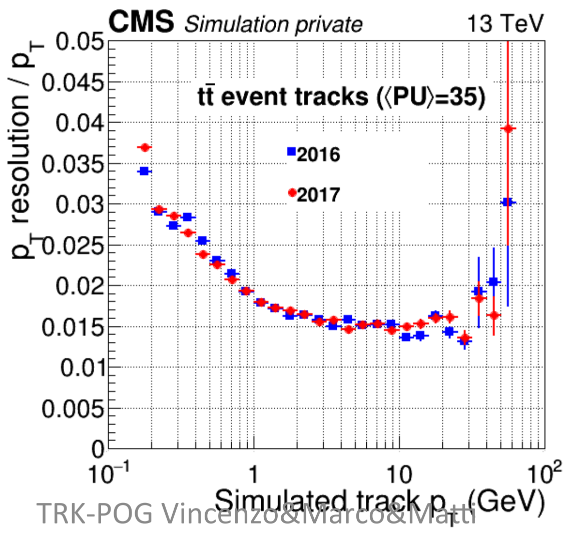
Resolution
Tracks & Vertices

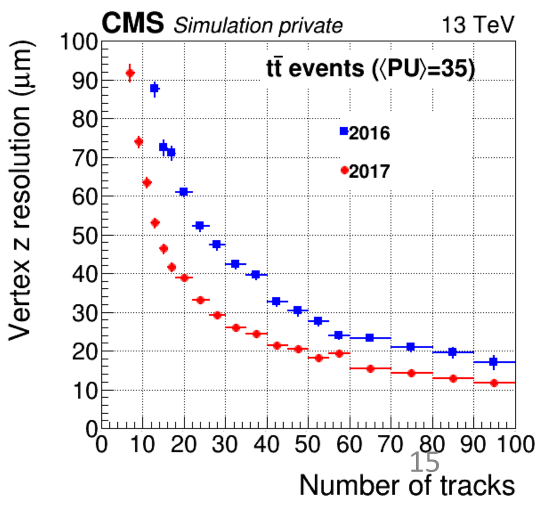






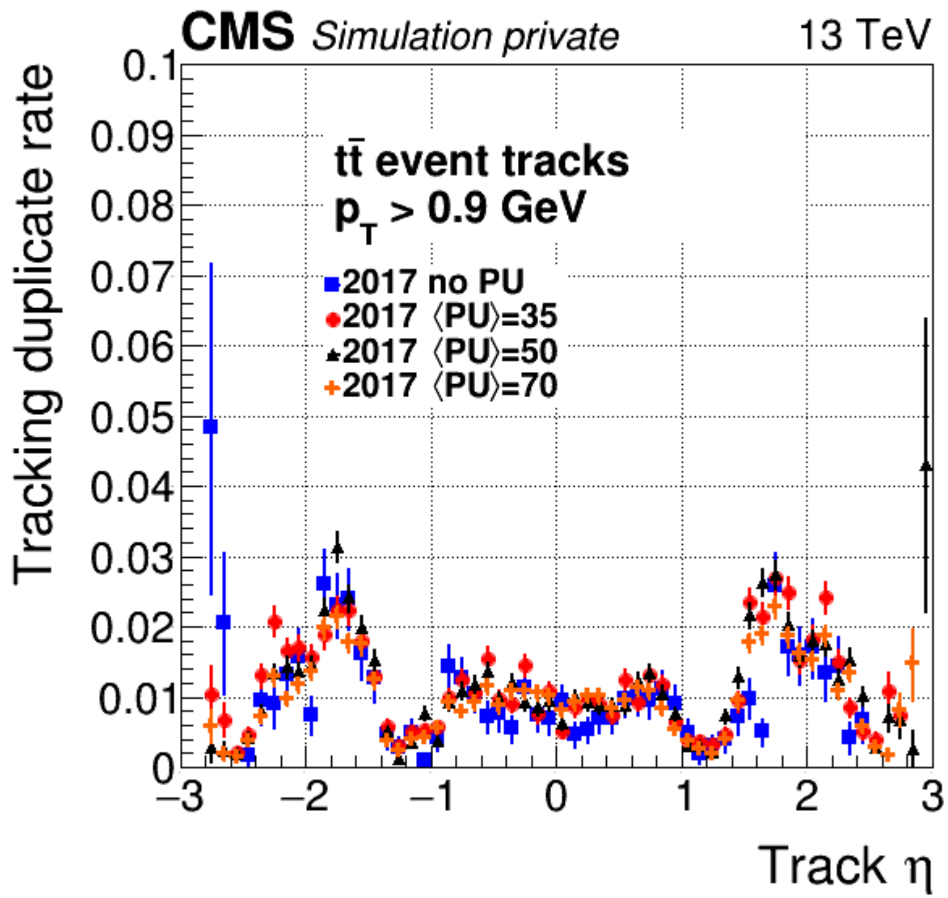






OPEN ISSUES

Duplicates

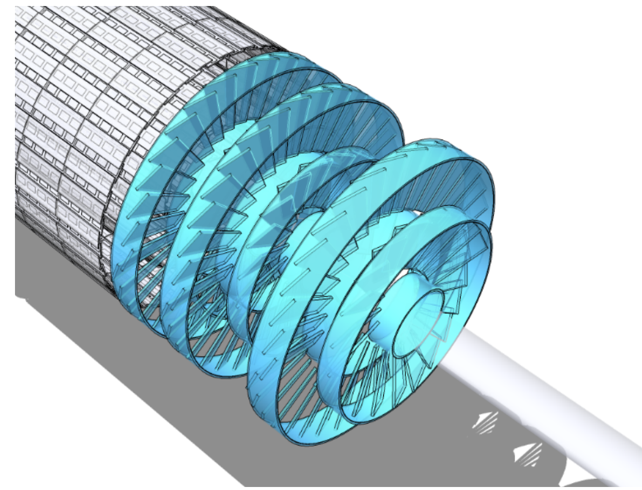


CMS Simulation private
13 TeV
Tracking duplicate rate
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
tt̄ event tracks
p_T > 0.9 GeV
2017 no PU
2017 ⟨PU⟩=35
2017 ⟨PU⟩=50
2017 ⟨PU⟩=70
−3
−2
−1
0
1
2
3
Track η

Duplicates

Quadruplets seeding brings in duplicate tracks

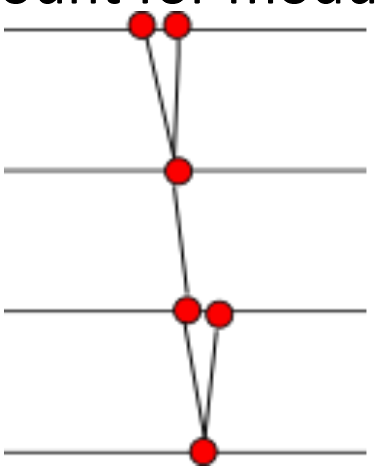
- Type I: Mainly due to the module overlaps (mostly in the FPix)
  - "with great hermeticity comes great duplicate rate"
  - Does not depend on which quadruplet seeding algorithm is used



- Type II: due to track shortening
  - A quadruplet is found in the pixel which fails to propagate in the strips
  - The remaining hits in the strips could lead to a track in the pixelLess step (or not)

Type I: Solution and mitigation

- Long-term solution:
  - Instead of using the CA for producing quadruplets, “fishbone” seeds can be produced to account for module overlaps



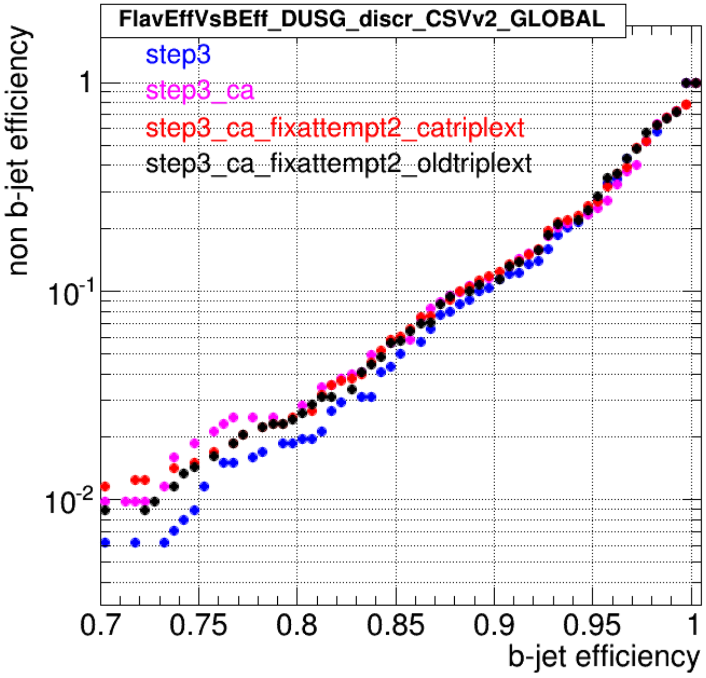
- Short-term mitigation:
  - Enhance the capability of the duplicate merger (ongoing by Matti)

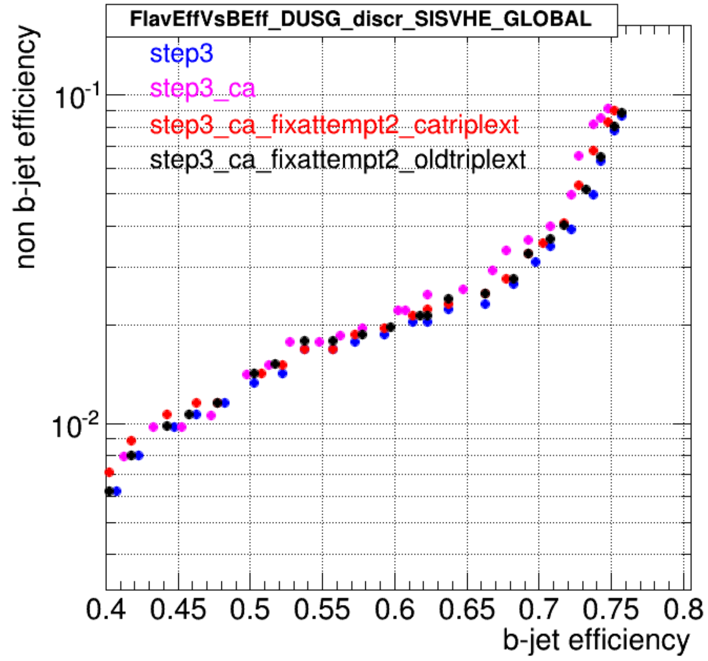
Type II: Solution and mitigation

- A type II duplicate/inefficiency appears in the building process
  - **It is a symptom of a deeper (more disturbing) issue!**
- The same quadruplet can lead to different built tracks depending on the TrackingRegion of the iteration
  - The evaluation of the error in the initialKinematics depends on the region's originRadius and on a *"multiplier"*
  - The Cellular Automaton is more efficient in finding displaced tracks even in "prompt" iterations (thanks to the x-y compatibility)
  - Building quadruplets does not really need to take into account the beam-spot, as four pixel hits can already provide a good-enough fit
- Type II inefficiency is one of the causes of inefficiency in tracks from B-hadrons
- Long-term solution: decouple the computation of the initialKinematics from the region
  - Use a proper fit of the seed
- Mitigation: modify the *multiplier* in quadruplet iterations to take into account the CA x-y plane extra tolerance
- In contact with BTV

B-Tag inefficiency

- Still under investigation
  - little understanding of the driving forces in b-tag estimators
- Exacerbated by CA
- Possible cause: one “wrong” hit in seed?
  - Global track params mostly ok BUT impact-param





Failure scenarios

- Still in the doing
- No plan to use PixelPair in seeding
  - Unless major permanent failure
- Experience shows that relaxing cuts does not help
  - Just increase fake rate
- We need to implement the use of "Inactive detector" information at seeding time and propagate that information to Ckf

"Commissioning scenarios"

- Pixel-less tracking (strip-Only)
  - Not at High PU
  - Forget beamspot
- Strip-less tracking (Pixel-Only)
  - Ok
- Large misalignments
  - More fakes
- Consecutive inactive (largish-section of) layers
  - Inefficiency & fakes (may affect MET as well)
  - In pixel or TIB1/2 : bad data at HighPU

Short term plan (February)

- Finalize MVA
- Mitigate Duplicates
- Finalize Seeding Strategy
  - Move to CA (still in tuning)
- Retune Vertexing
- Reduce Memory footprint

Medium term Plan (summer?)

- Implement a proper fit of Seeds
  - Use it as "initial kinematics" in Ckf
- Detect inactive/missing layers at seeding time and propagate the info to Ckf
- Fishbone seeding
- New MVA for high PU (Deep Learning?)

In reality

Cope with reality

Summary

- TRK-POG is essentially ready for 2017 commissioning
- Still missing responsible of Online Beamspot
- Baseline reconstruction with good performance up to PU 70 in the release
- Innovation in Seeding is uncovering (possibly old) issues
  - Aiming at 0 regression is not necessary progress….
- Improvements in tracking are expected for the summer and eventually later on
- Ready to face reality

BACKUP

Time & Memory

One Thread
Including Validation and DQM