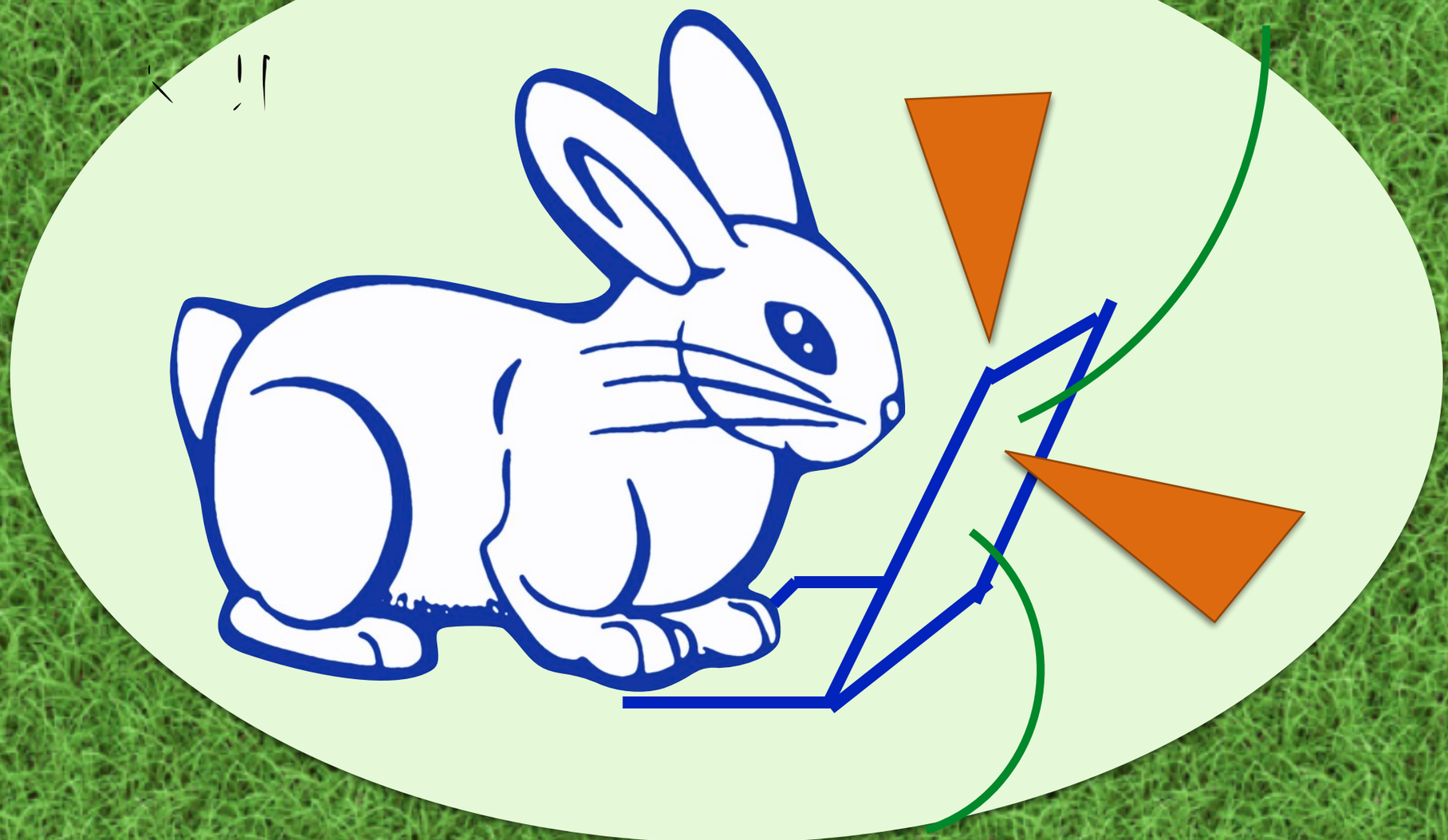


Recent developments in CMS FastSim

Sezen Sekmen / Kyungpook National University
for CMS Collaboration

ICHEP 2016, Chicago, 3-10 Aug 2016

Detector simulation in CMS

CMS FullSim

- detailed geometry
- particles tracked in small steps
- detailed material interaction model (mostly Geant4)
- detailed emulation of detector electronics and trigger
- standard event reconstruction

-O(100s) per ttbar event

CMS FastSim

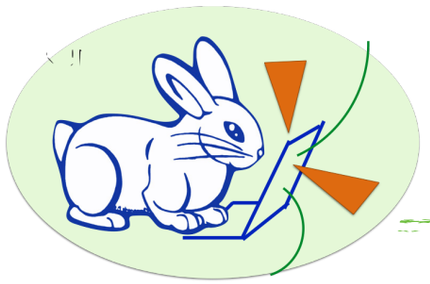
- simplified geometry
- infinitely thin material layers
- simple analytical material interaction models
- detailed emulation of detector electronics and trigger, with exceptions
- standard event reconstruction, with exceptions

-O(5s) per ttbar event

Delphes

- (almost) simple 4-vector smearing

-O(.01s) per ttbar event



Introduction to CMS FastSim

Fast simulation (FastSim) is an integral part of CMS physics studies.

- Speeds up CMS event simulation ~100 times and CMS event simulation +reconstruction ~20 times.
- Actively maintained by ~20 developers working part time on different aspects of the framework.
- FastSim is used for
  - large SUSY signal model scans
  - some signal models for Exotica studies
  - systematic studies for top quark measurements
  - private productions for MSc/PhD theses.
- Regularly validated within the official CMS software release validation framework.
- Mainly validated against FullSim. Reproduces FullSim mostly by about 10%.

CMS FastSim flowchart

GenParticles

All subsystems

Simulate ***energy depositions along particle trajectory***

SimHits

Tracker

ECAL, HCAL, Muons

Gaussian / template smearing

Standard simulation of electronics (digitization, and hit reconstruction)

RecHits

Tracker

ECAL, HCAL, Muons

Track finding and reconstruction using MC truth

Standard reconstruction

Reconstructed particles (same in Fast and Full)

Fast emulation
Standard reconstruction
Objects (same for Fast&Full)

CMS FastSim flowchart

GenParticles

All subsystems

Simulate ***energy depositions along particle trajectory***

SimHits

Tracker

ECAL, HCAL, Muons

Gaussian / template smearing

Standard simulation of electronics (digitization, and hit reconstruction)

RecHits

Fast emulation
Standard reconstruction
Objects (same for Fast&Full)

Tracker

ECAL, HCAL, Muons

Track finding and reconstruction using MC truth

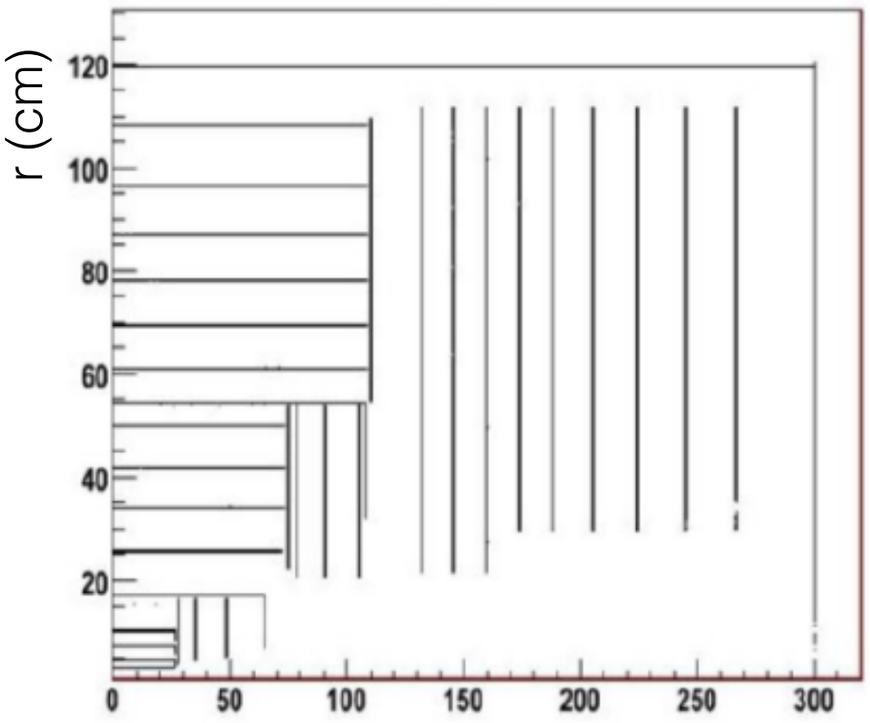
Standard reconstruction

Reconstructed particles (same in Fast and Full)

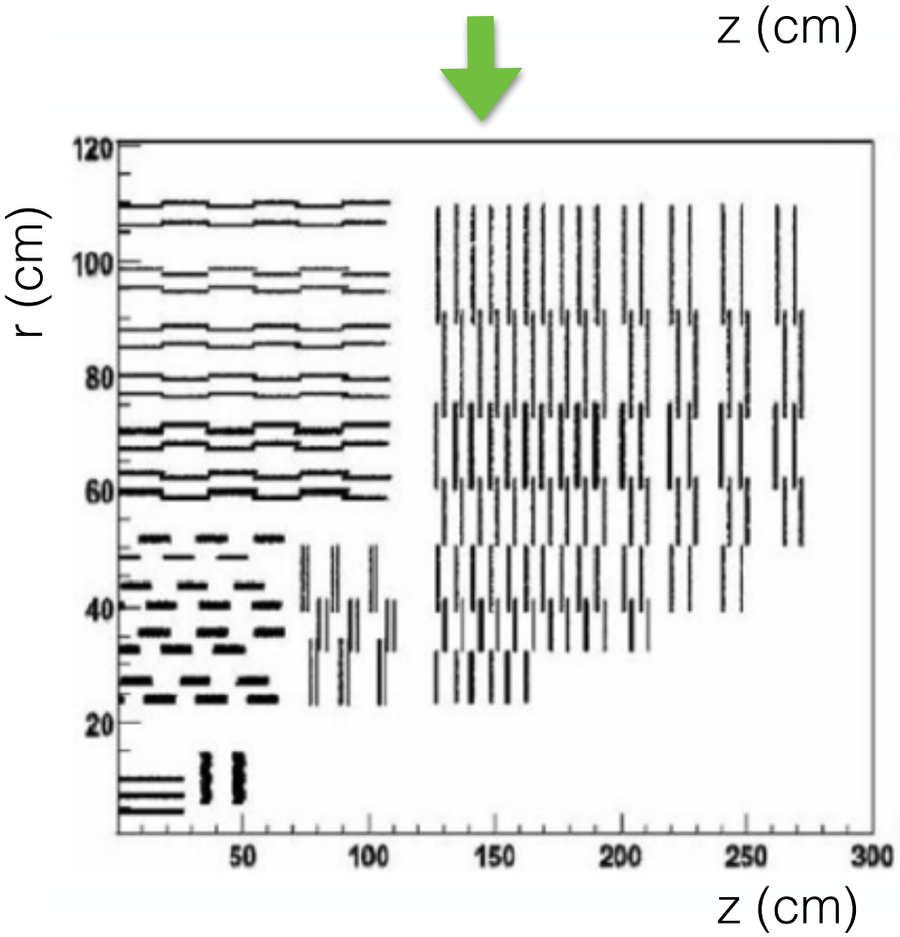
Recent developments in this talk

Tracker: geometry & SimHits production

- CMS tracker geometry is approximated by infinitely thin cylinders and disks.
- Material resides on the surface of layers.
- Material emulation is done by assigning thickness in radiation lengths / interaction lengths to the cylinders.
- Particles propagated with a simplified B field.
- Particle-material interactions occur in each layer.



- Next, consider real geometry and project particle intersections in ideal geometry to it.
- We find crossings between particles and modules
  —> calculate entry exit points
  —> obtain tracker SimHits.



Tracker: emulating RecHits

Pixel RecHits emulation: Pixels measure 2D in x, y

RecHit position: Smear SimHit positions based on templates obtained from a detailed pixel simulation (PIXELAV). Templates depend on:

- dimensions: X / Y (local coordinates)
- pixel type
- incident angle
- number of pixels that are hit

RecHit error: Extract from pixel templates used in standard pixel hit reconstruction.

Strip RecHits emulation: Strips measure 1D in x.

RecHit position:

- X: Gaussian smearing <— width from FullSim
- Y: taken as the strip center

RecHit error:

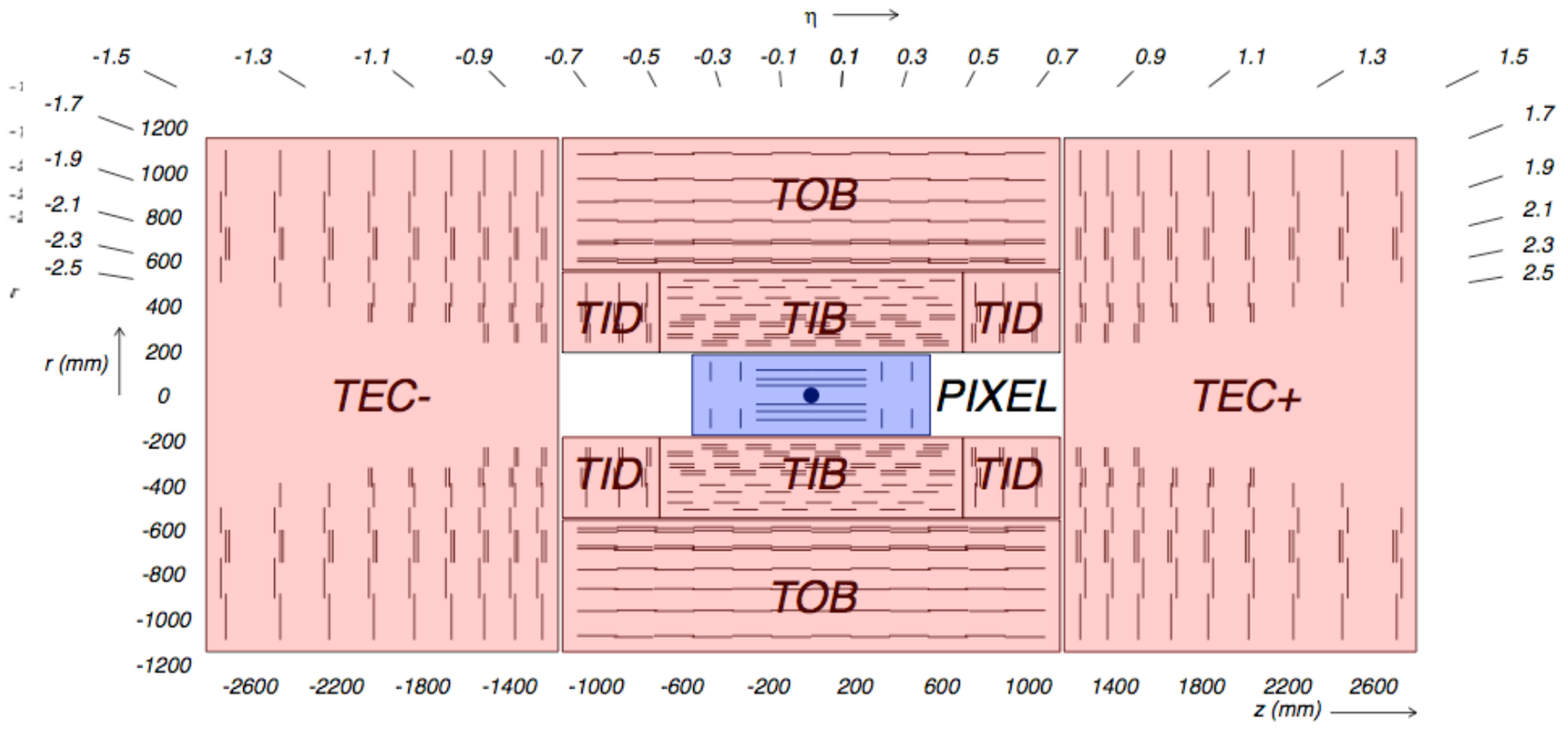
- X: width of the Gaussian
- strip length / sqrt(12)

Tracker: Tracking RecHit Producer

NEW!

Designed a new package for a modular and systematic fast emulation of tracking RecHits.

- It consists of plugins configured per detector ID. Different features are activated when necessary.
- Behaviour is controlled through python configuration, e.g.: same plugin for all strips but with different smearing parameters for each layer.
- Tested and integrated.



detailed simulation with hit merging

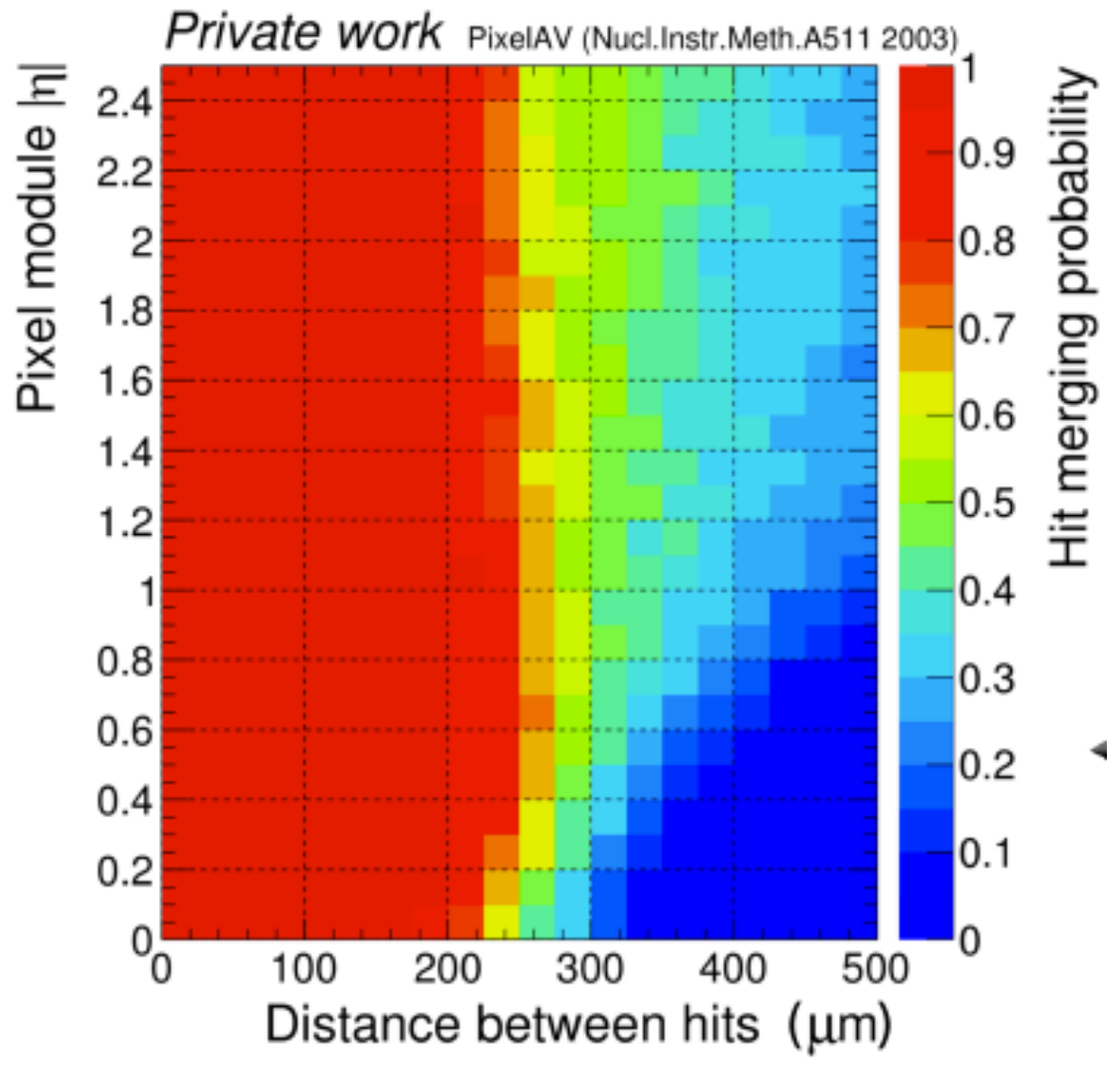
simple Gaussian smearing

Tracker: Pixel hit resolution & merging

NEW!

- Used PIXELAV/SiPixelTemplate to produce resolution histograms for pixel SimHit smearing to obtain RecHits.
- Used PIXELAV to simulate probability of two SimHits merging given local η and distance between them. Define new RecHit from merged SimHits.
- Inefficiency through hit merging is currently being studied.
- Incorporated in the new Tracking RecHit Producer. Tests ongoing.

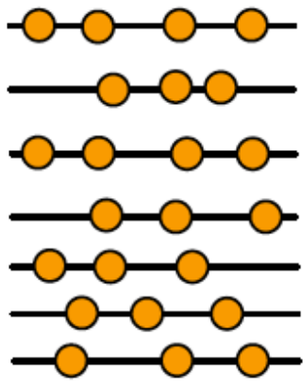
SimHit merging probability



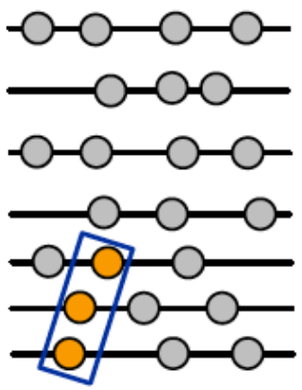
Better pixel RecHit resolution
—> better simulation of isolated tracks
—> improvements in b tagging, top tagging, boosted object tagging.

Track reco: FullSim vs. FastSim

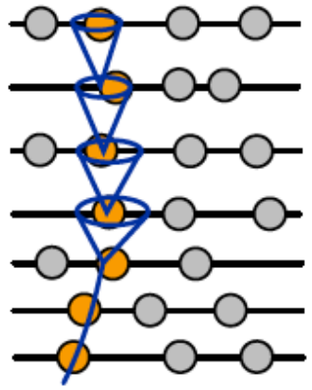
FullSim: Combinations of hits need to identified from a nearly infinite number of hit permutations created by charged particle trajectories, bent by the B field.



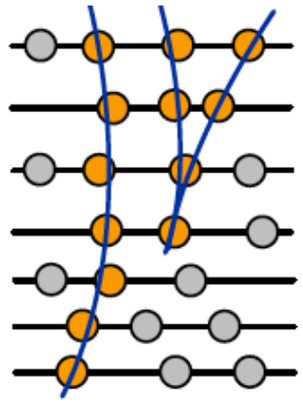
RecHits from charge deposits



seeding: find start of potential trajectories

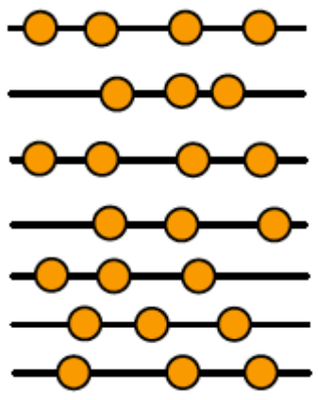


trajectory finding: add hits to the seed that supports trajectory hypothesis

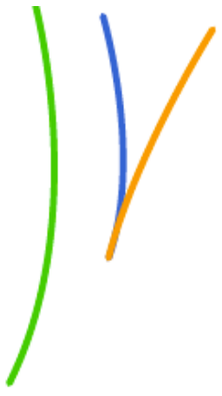


trajectory fit: global fit to estimate track parameters.

FastSim: Restrict seeding and trajectory finding to only a local subset of hits using MC truth information. Large speed up by skipping permutations.



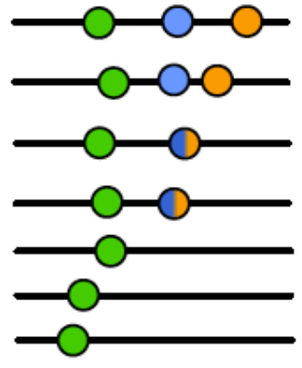
RecHits from charge deposits



Look up particle truth information

- create seeds
- build a track candidate
- fit a track.

Iterative tracking: Hits used in one iteration are hidden from the next.

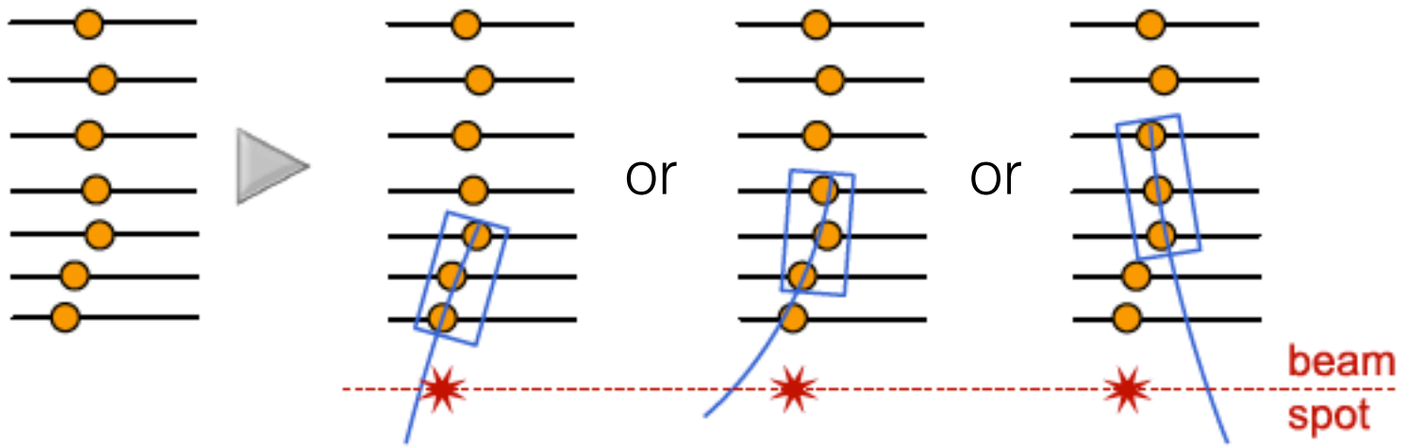


Create subsets of consistent RecHits

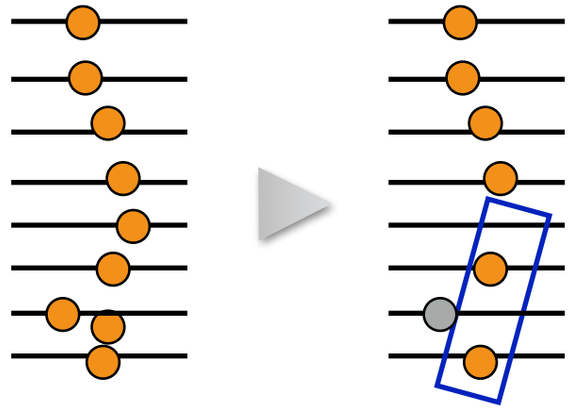
Track reco: Recent fixes to fast tracking

NEW!

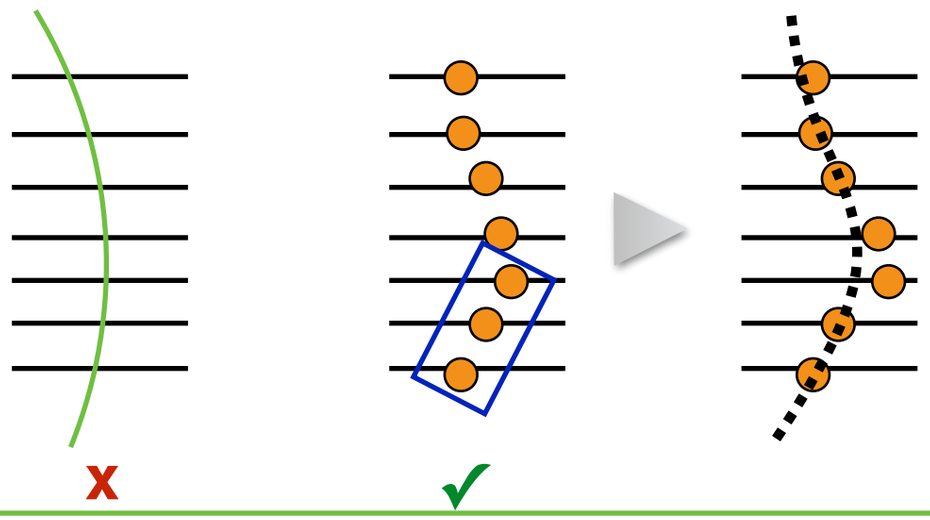
Fix1: Doublet-triplet check in seed finding: Seed doublets or triplets are required to fulfil certain quality criteria (similar to standard track reco), e.g. distance to beam spot.



Fix2: Hit skipping: Seed creation skips hits based on quality cuts. Track reco now also skips these hits.



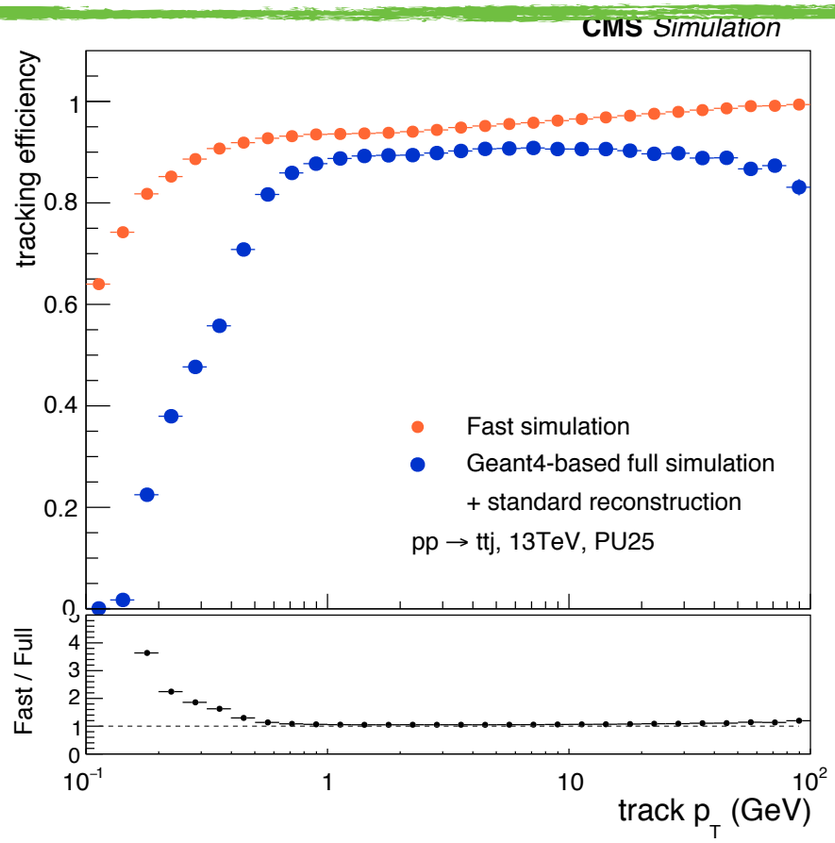
Fix3: Using seed parameters in track fitting: Track fit took the initial parameters from the truth particles. Now, initial parameters are taken from the seed. Track fit can work on any set of hits.

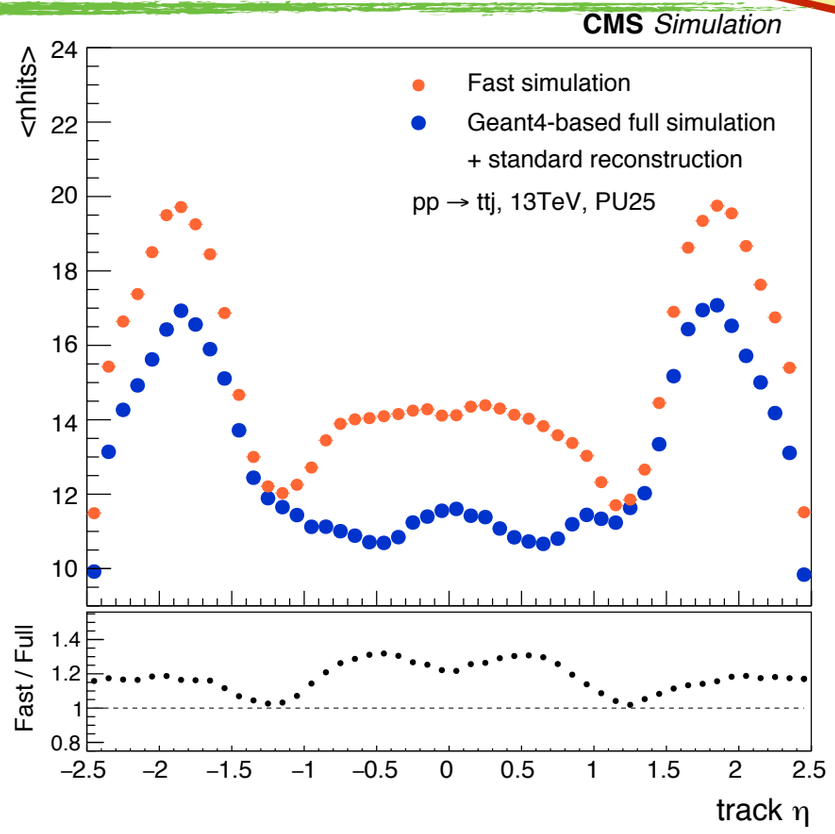


Track reco: Results of the fixes

NEW!

Before the fast tracking fixes:

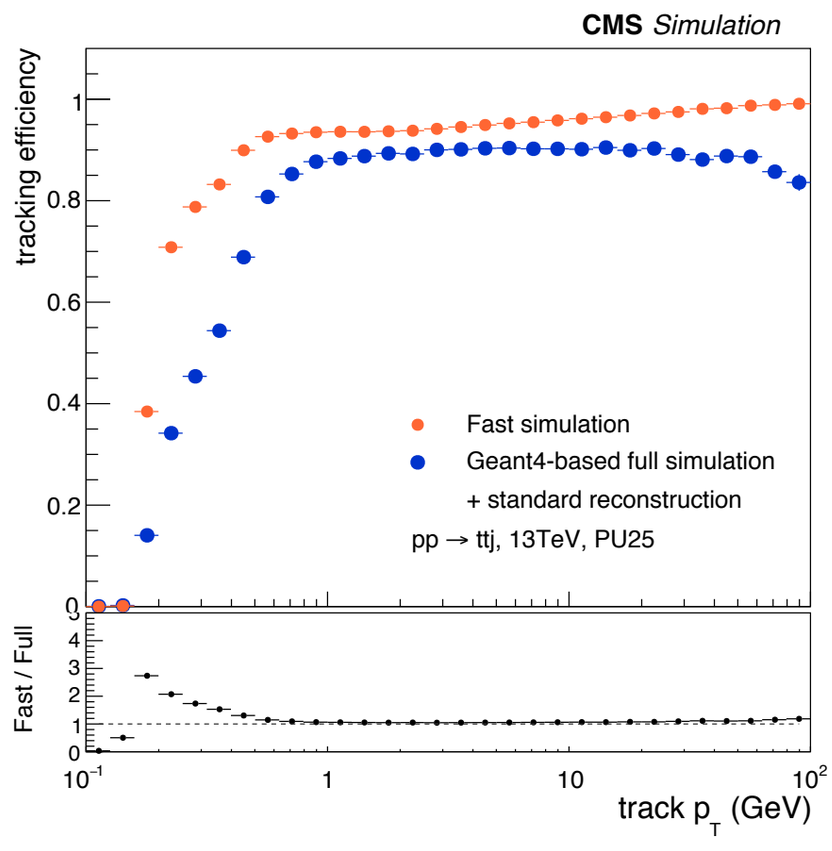


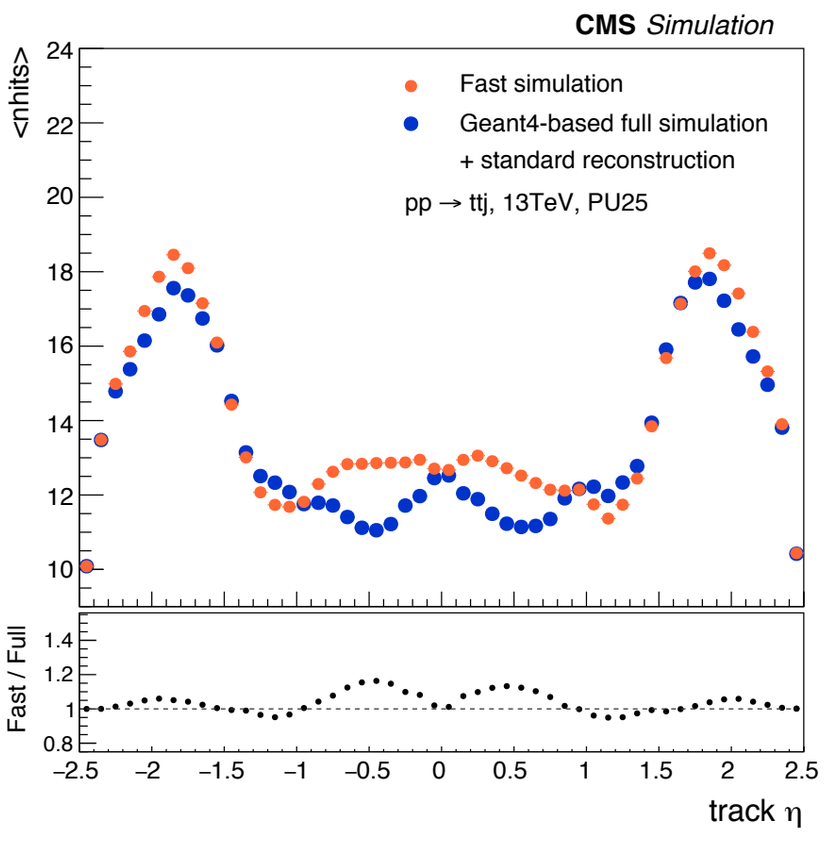


After the fast tracking fixes:

Significant improvement in FullSim-FastSim agreement.

Better agreement w/o PU.





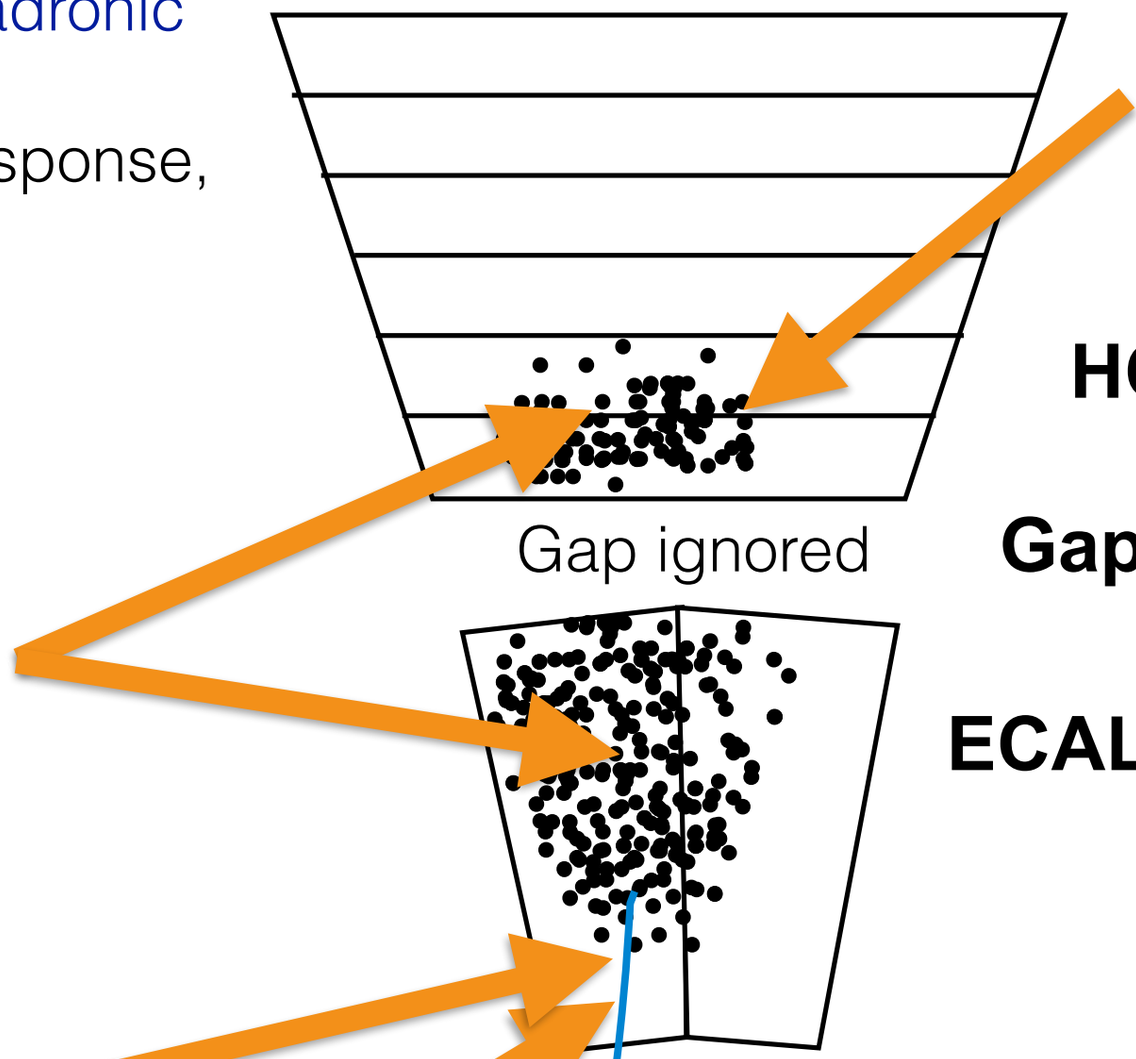
Calorimetry: Simulation Overview

Barrel and endcap: EM and hadronic showers simulated by
—> parametrizing the calor response,
—> fitting to FullSim or data.

Longitudinal and transverse shower shapes modelled with ~GFLASH parametrization.
Shower shapes fluctuate from shower to shower:
—> Energy spots within a shower —> SimHits

Shower starting point modelled based on radiation / interaction length

No energy loss before shower start



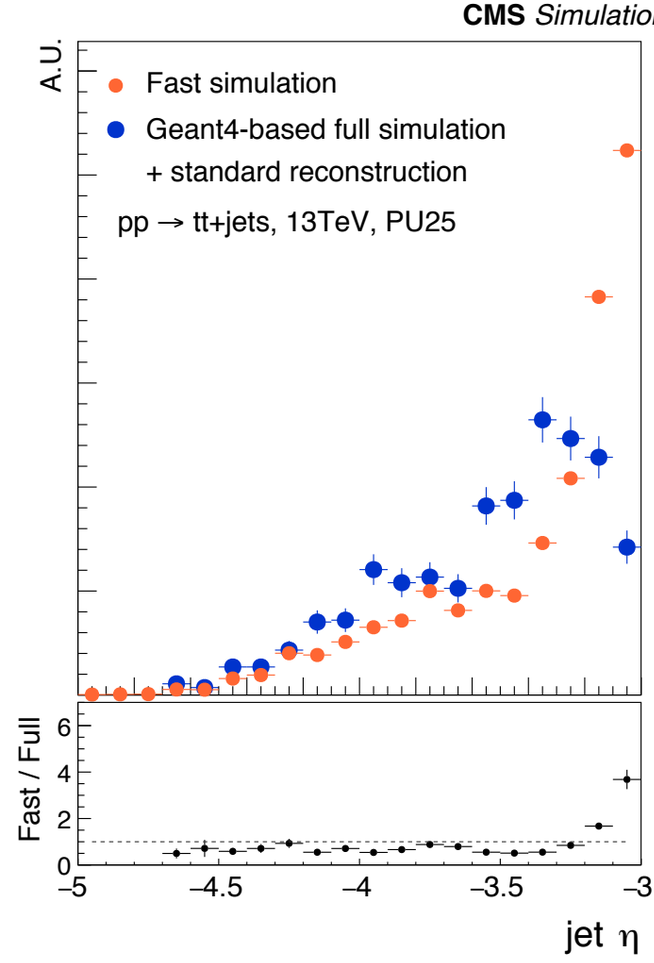
- γs: simulate conversion then simulate showers of e^-e^+
- μs: E loss via ionization.

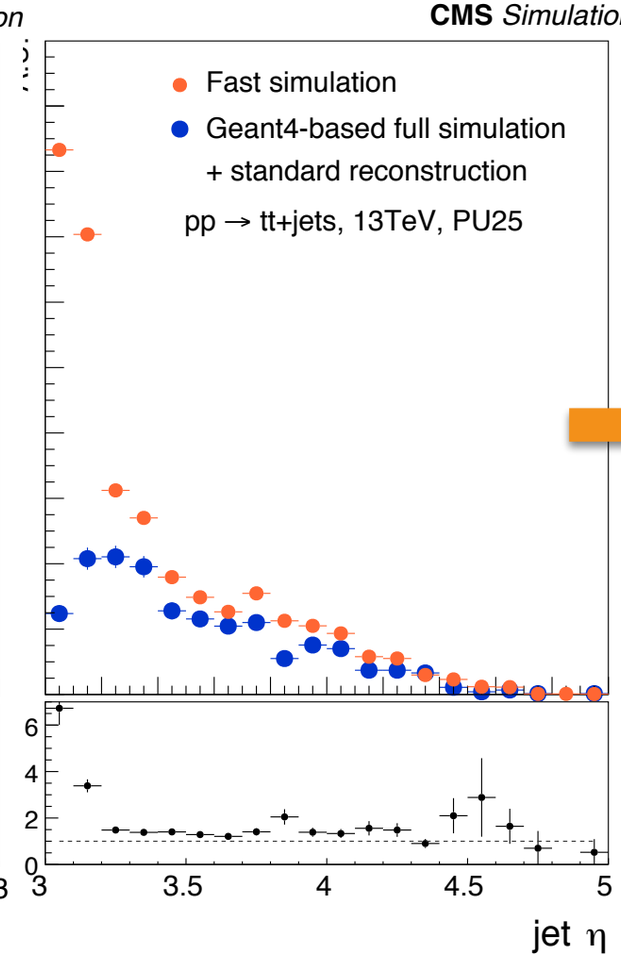
Calorimetry: HF showers

NEW!

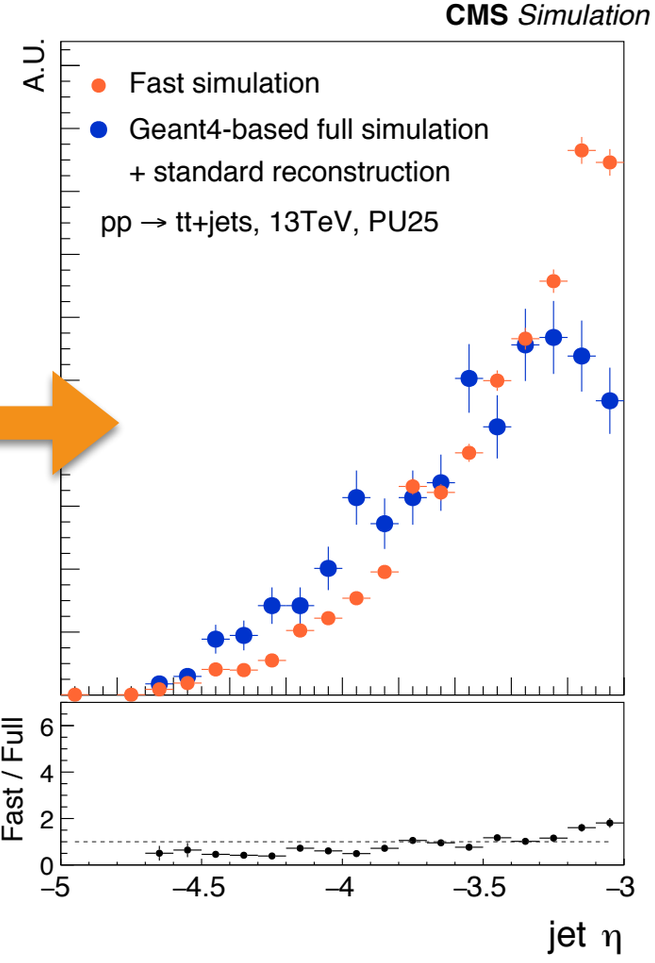
- We use a shower library simulated using GEANT4 to simulate the showers in the hadron forward calorimeter (HF) region.
- Showers are classified according to particle type, η and energy.
- Material in front of the HF is not modelled. We recently applied FastSim-specific correction factors to correct the incomplete modelling.

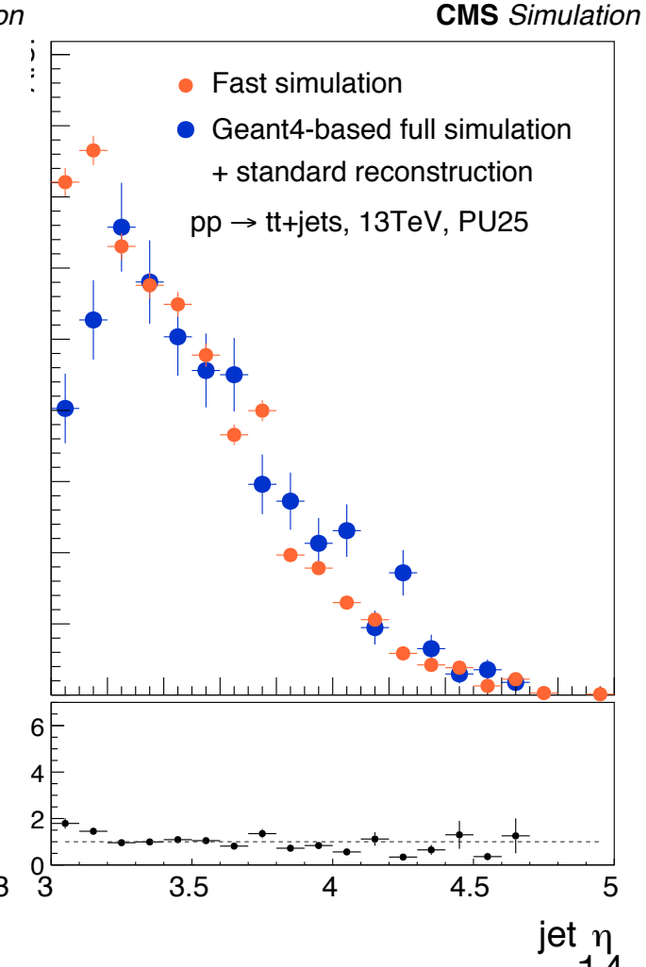
Before HF corrections:





After HF corrections:

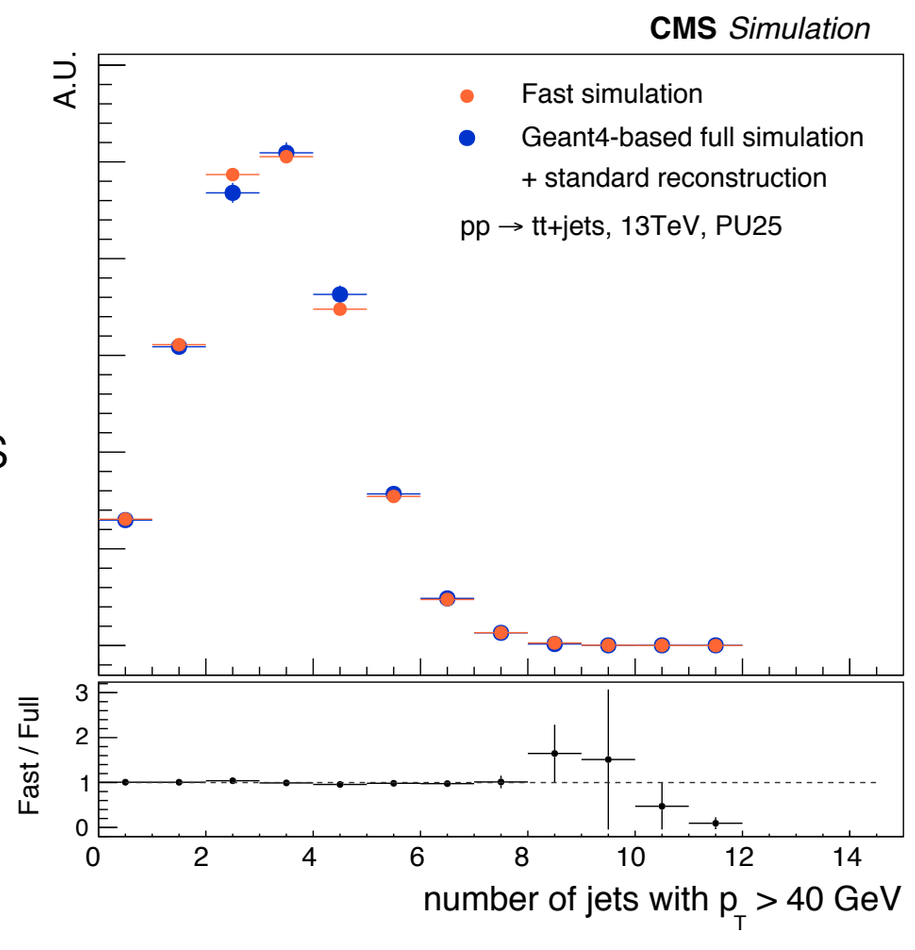


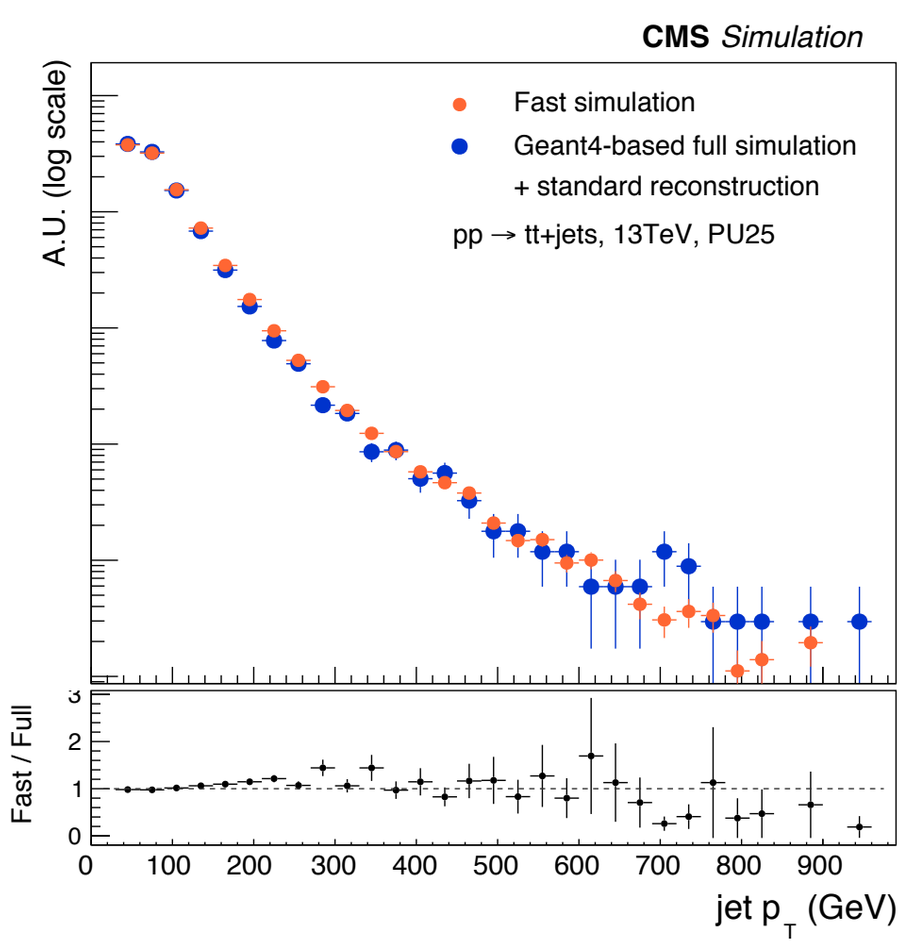


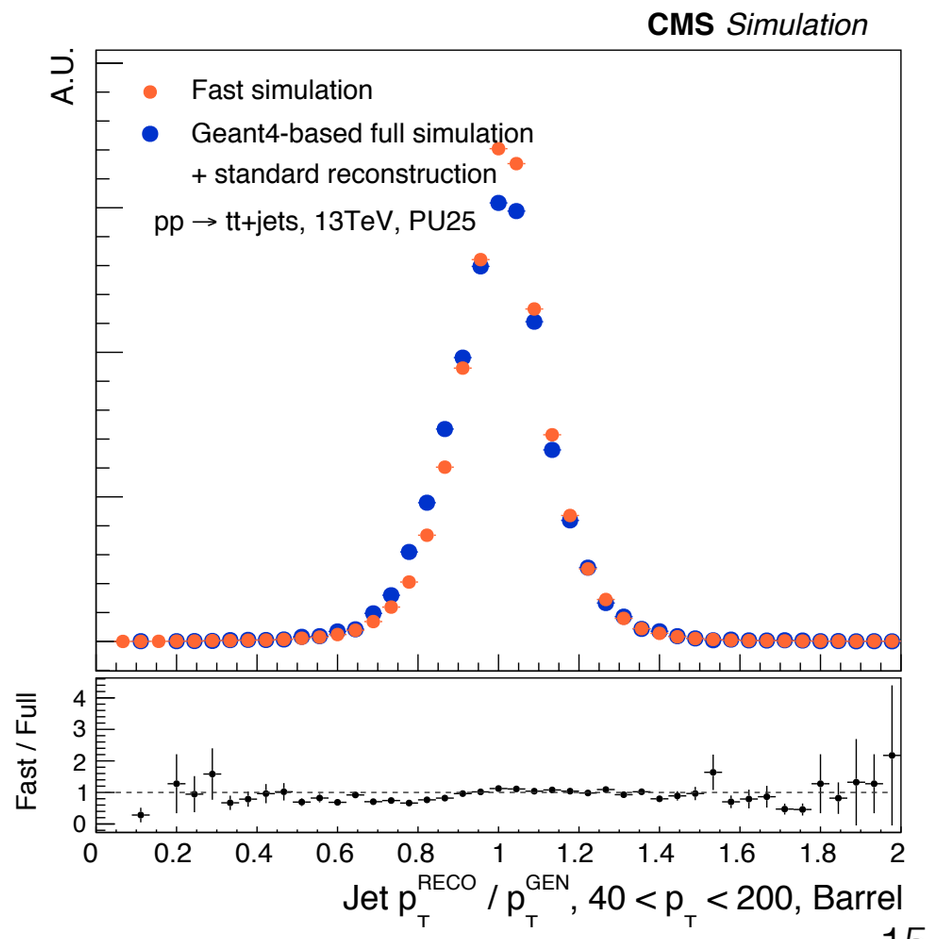
Validation: Jets

Jets are reconstructed with particle flow algorithm using the tracker in $|\eta| < 2.5$ but rely on calorimeters outside - especially HF for $3 < |\eta| < 5$

- Difference wrt FullSim: track simulation and reconstruction, calorimetric shower simulation.
- Excellent performance of basic properties, e.g. jet energy scale, resolution (better in barrel)
- Work ongoing to improve shower modelling —> will result in better jet composition.



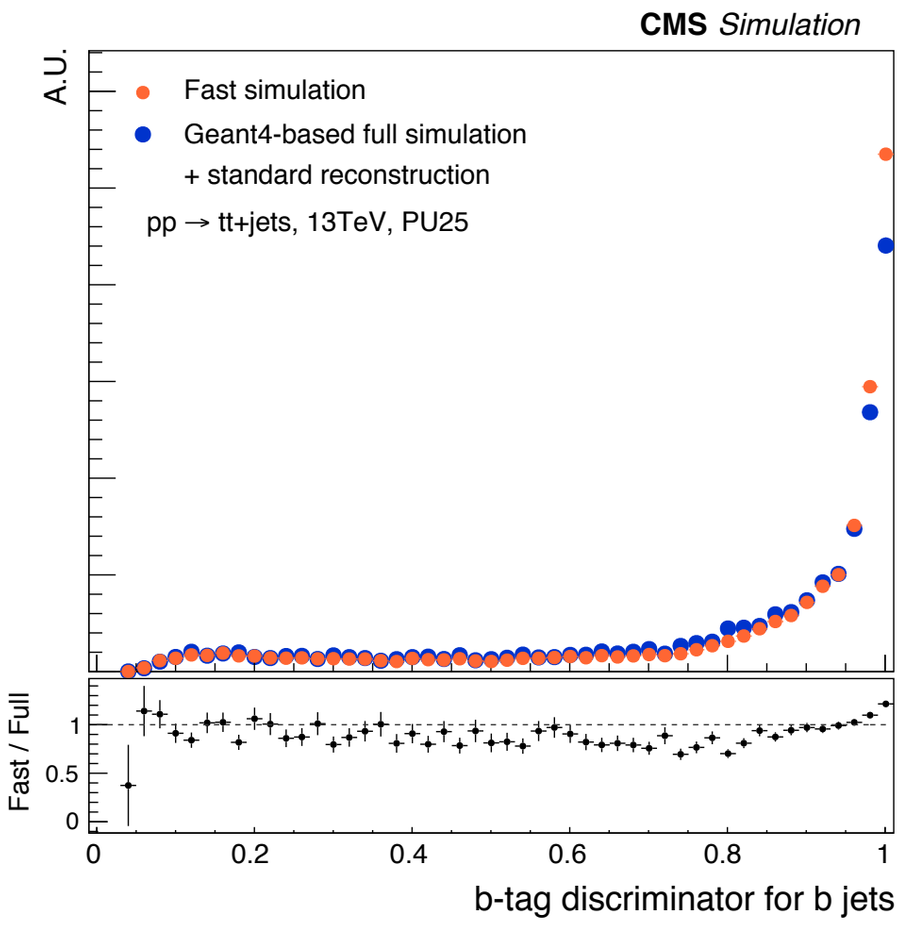


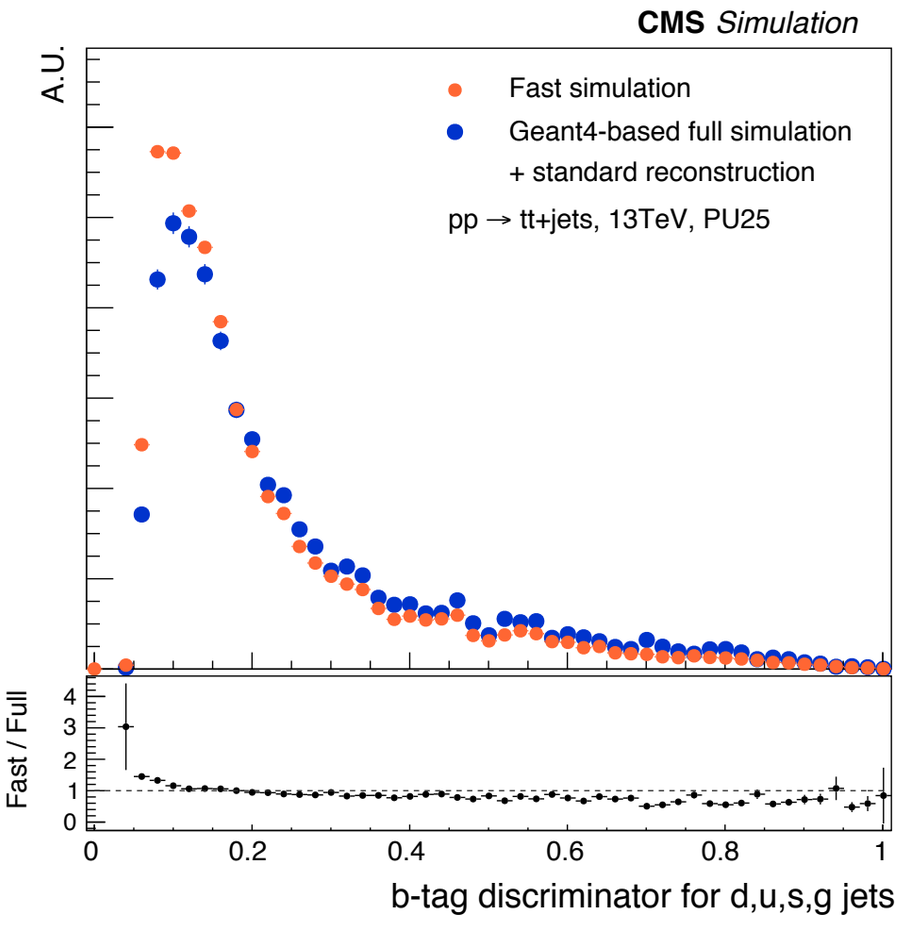


Validation: b tagging

b-tagging is important in new physics, top quark and Higgs boson studies.

- b-tagging algorithms same as FullSim. Main differences arise from low level track and vertex reconstruction.
- Recent tracking developments give an improved, very good agreement.

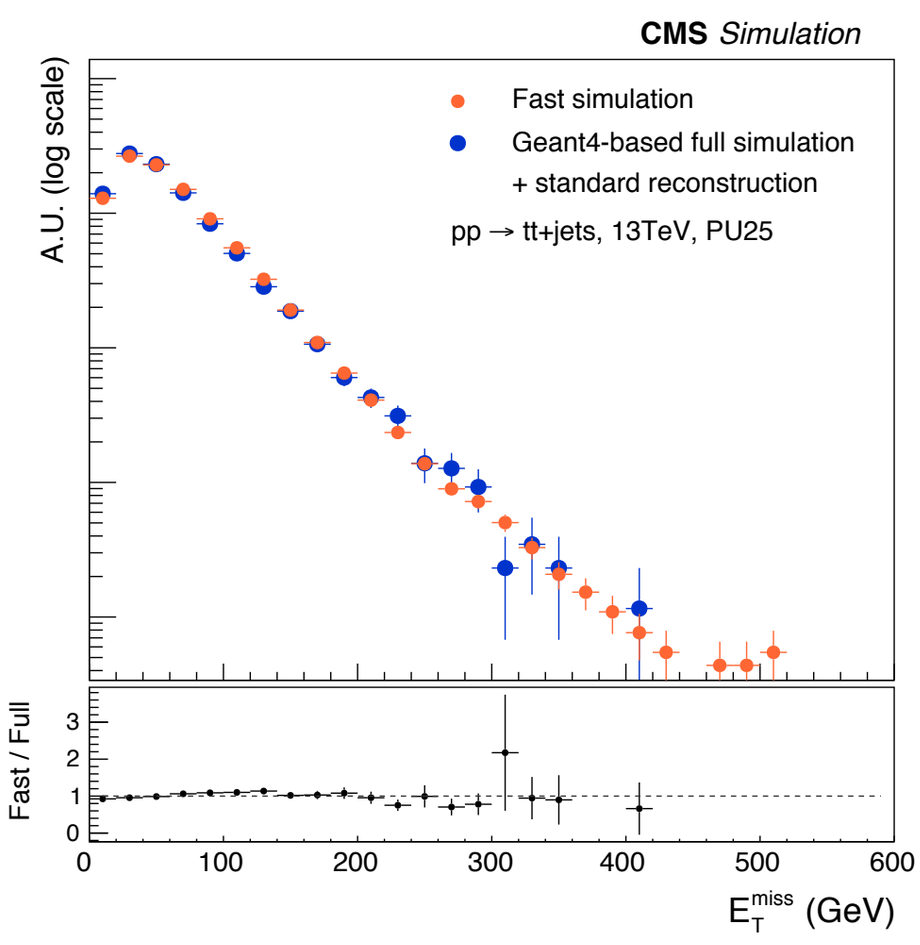


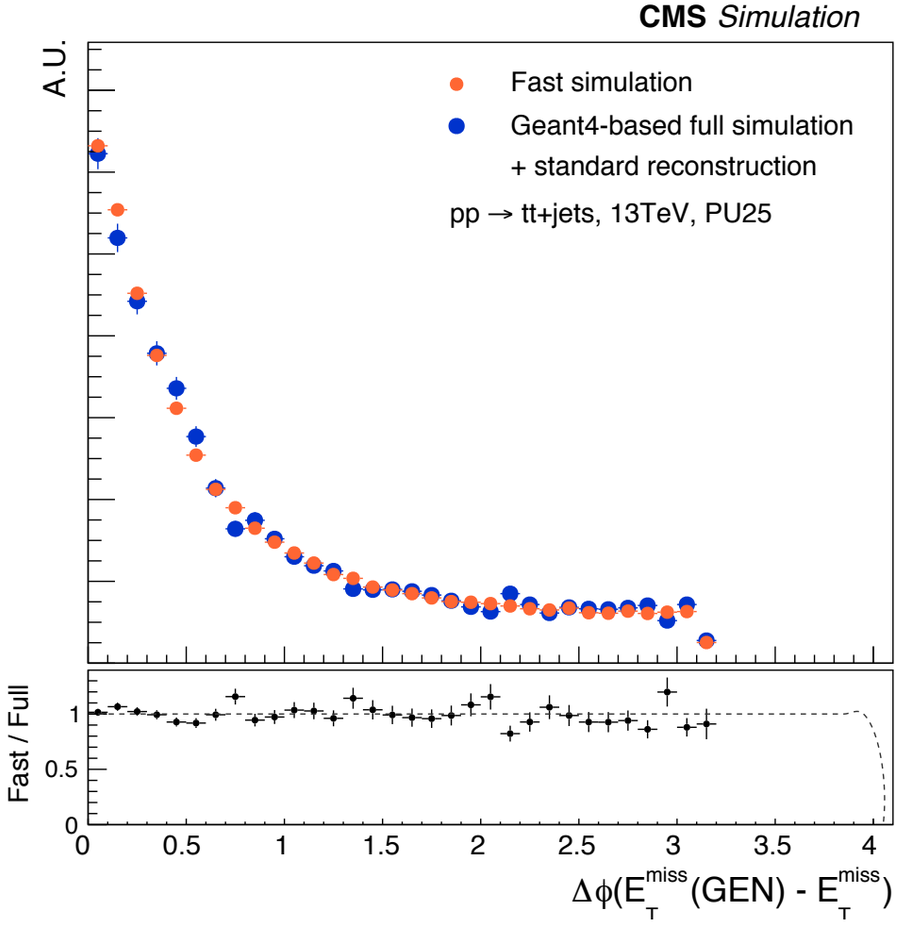


Validation: Missing transverse energy

E_T^{miss} is very important for new physics searches. FastSim is used for simulation of all SUSY signals in CMS. It is also the "golden observable" where any significant mis-modelling of any object will show up.

- Difference wrt FullSim: tracking, calorimetric shower simulation
- Overall, very good agreement with FullSim. E_T^{miss} resolution will be improved by ongoing calorimetry shower developments.

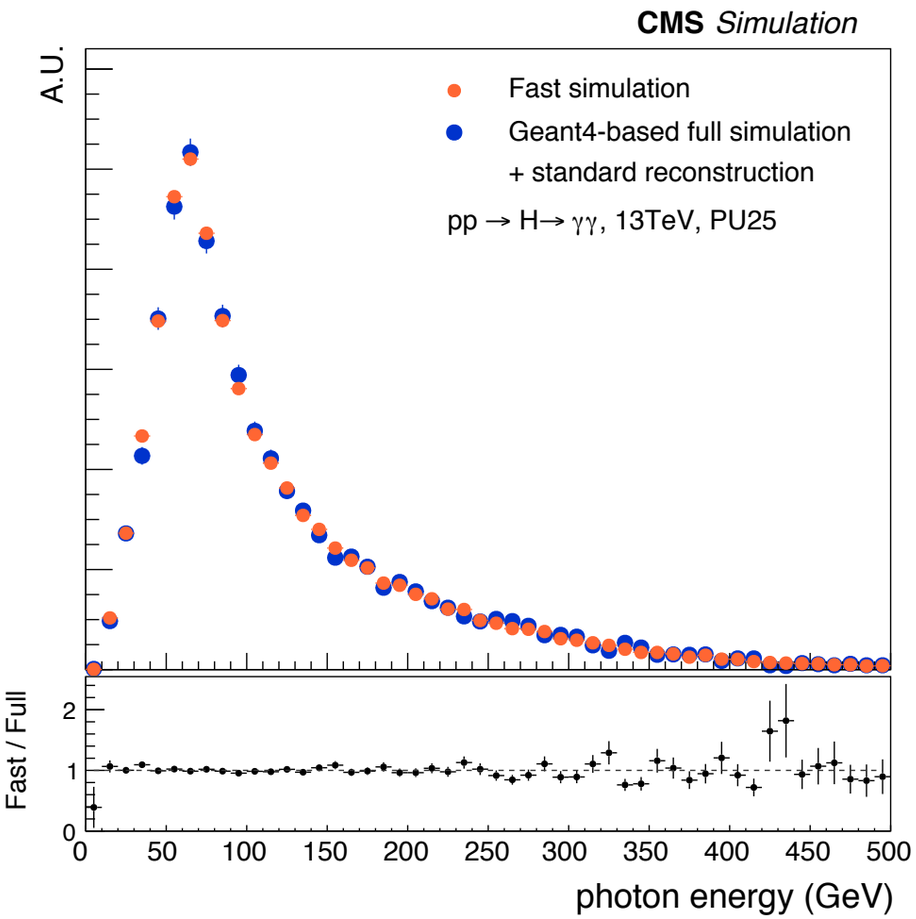


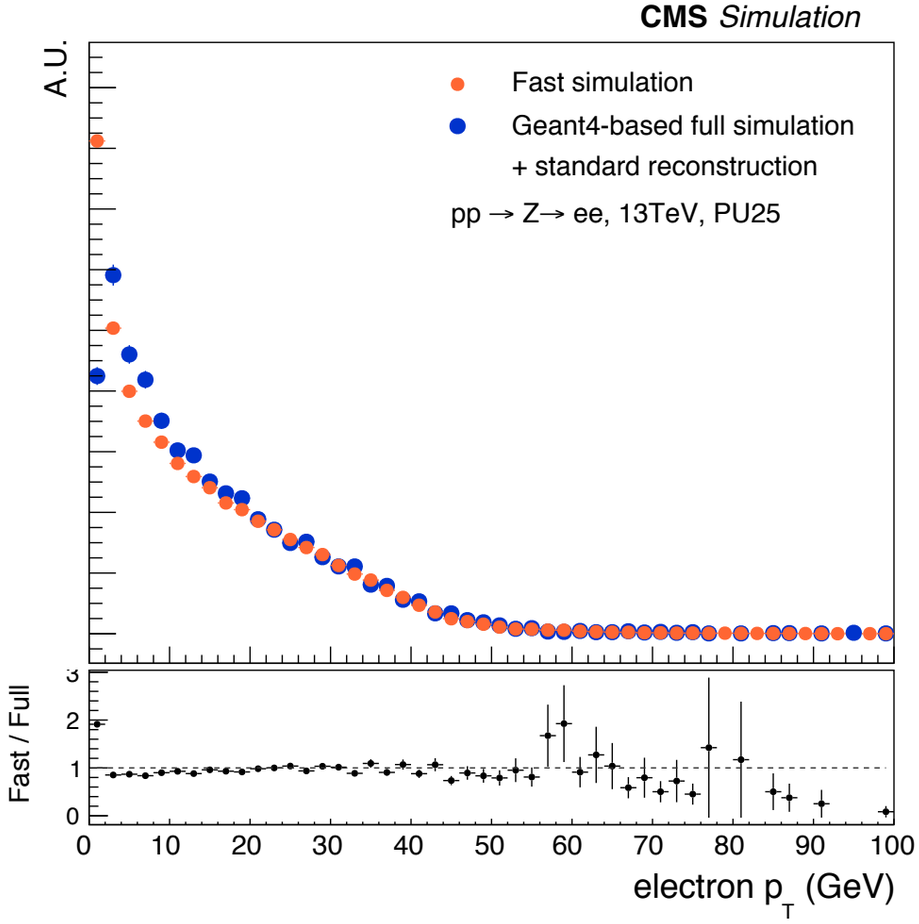


Validation: photons, electrons

Photons: New physics and Higgs searches

- Difference wrt FullSim: EM showers.
- Energy scale, resolution, ID, isolation well-reproduced.
- Cluster shapes can be improved for a better agreement.





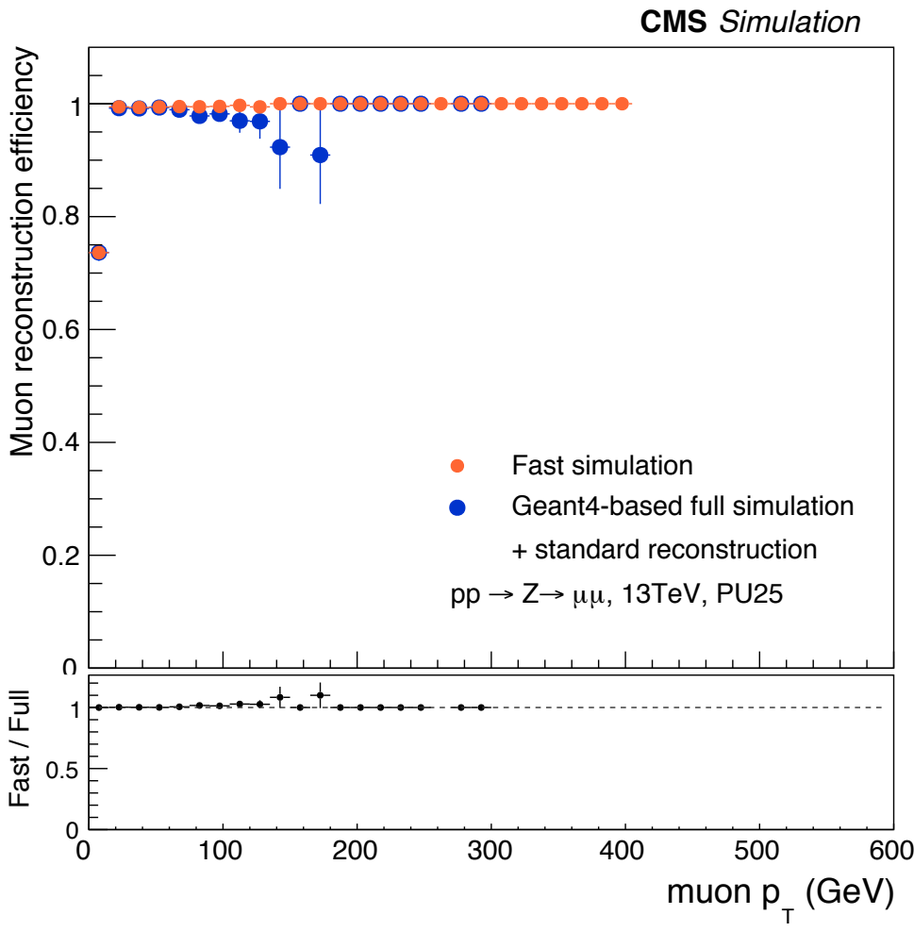
Electrons: New physics, top quark searches.

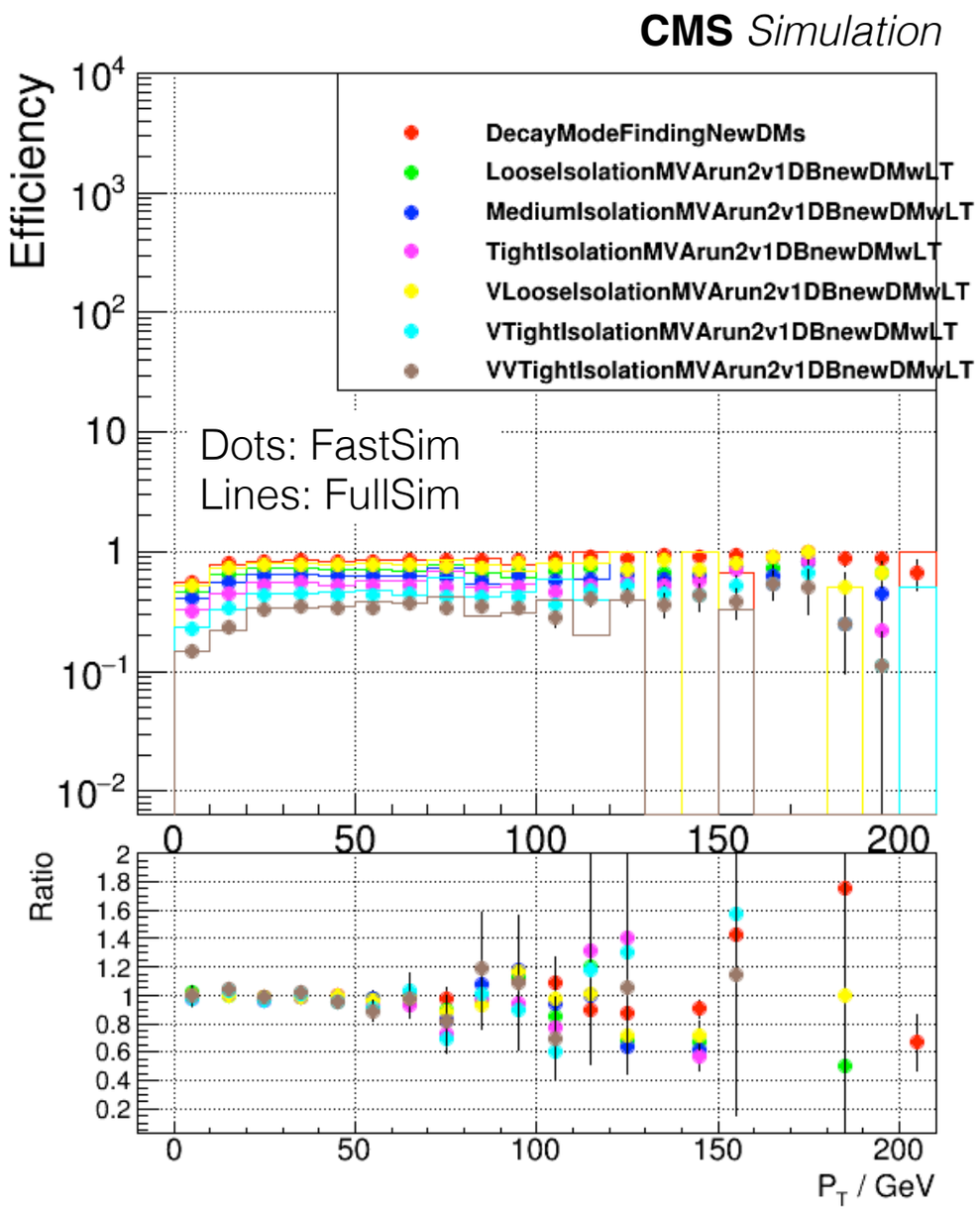
- Difference wrt FullSim: tracking and EM showers.
- Energy scale and resolution, ID, isolation well-reproduced. Better behaviour in central regions.
- Track-cluster matching and cluster shapes can be improved for better agreement.

Validation: Muons, taus

Muons: New physics, top quark searches.

- Difference wrt FullSim: tracking.
- Very good agreement.





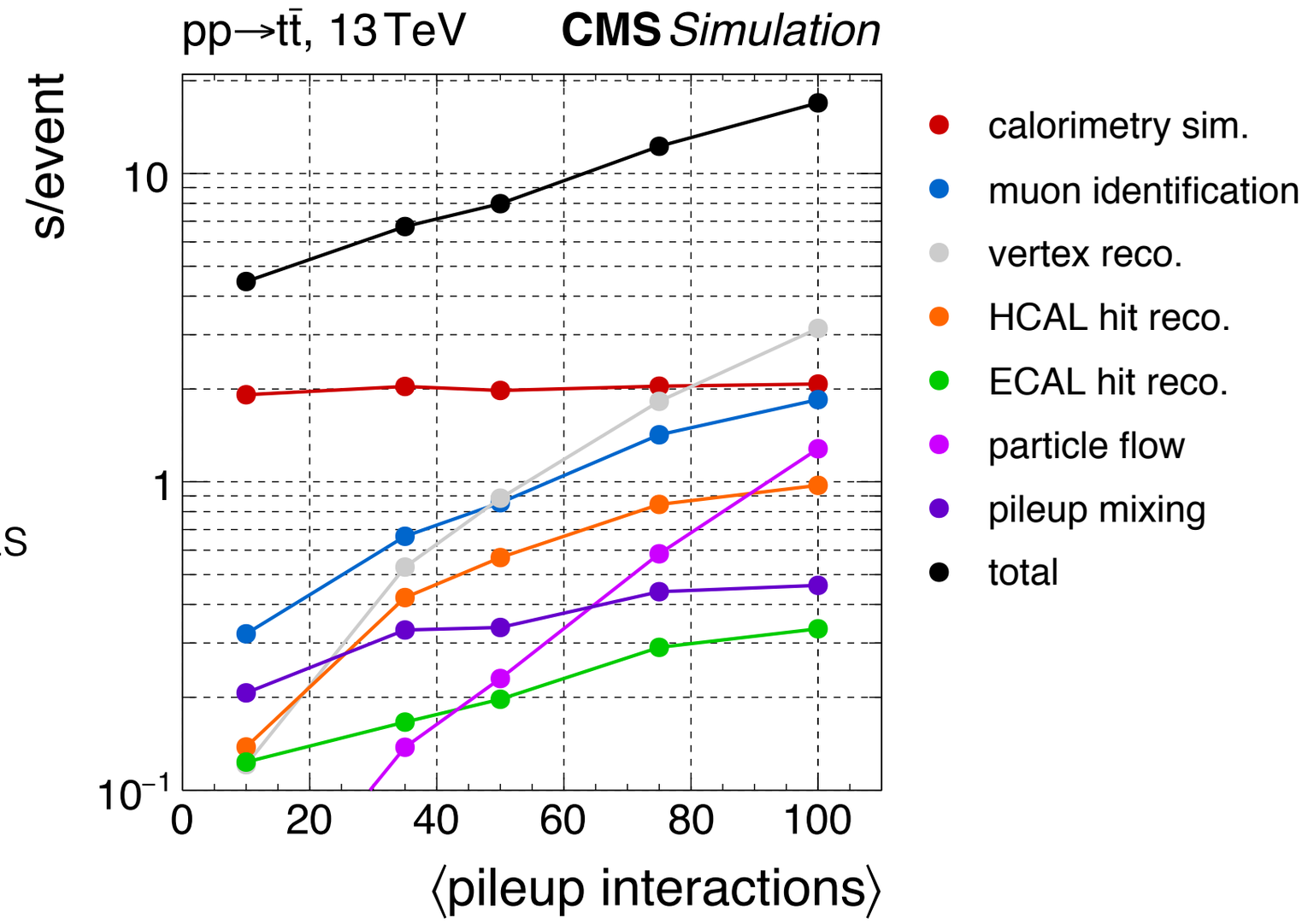
Taus: New physics, top quark searches.

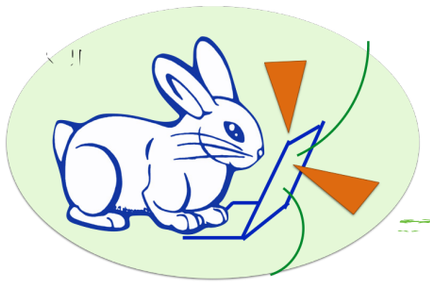
- Difference wrt FullSim: tracking and hadronic showers. Tau reconstruction is same as FullSim.
- Very good agreement.

Using FastSim with pile-up

Without PU, FullSim simulation step for tt+jets takes ~50s/event.

FastSim will become increasingly economical as the number of PU interactions increases.



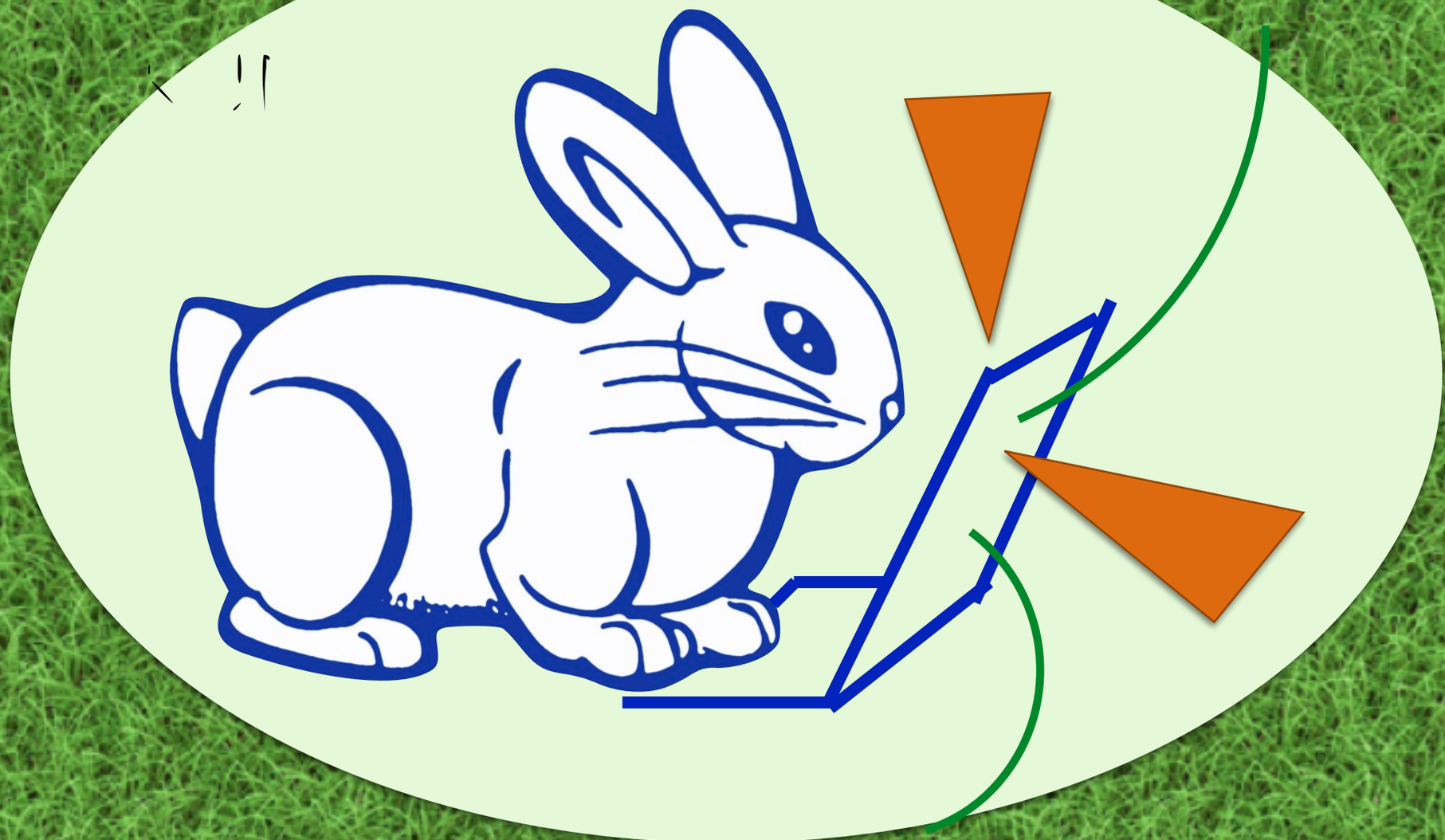


What is next?

CMS FastSim performs very well, and is used efficiently in CMS (see CMS SUSY and top talks at this conference).

The following work is ongoing or foreseen for near future:

- Prepare FastSim for the upcoming pixel upgrade (work ongoing).
- Complete improving and tuning the model for hadronic interactions with the ECAL and HCAL
  - FastSim calorimetry package is being fully restructured for this study. This will achieve a more efficient tuning procedure and more flexibility in calorimetry upgrade implementations.
- Develop a package for validation and tuning of material interactions in the tracker.
- Improve the material description in front of the hadron forward calorimeter.

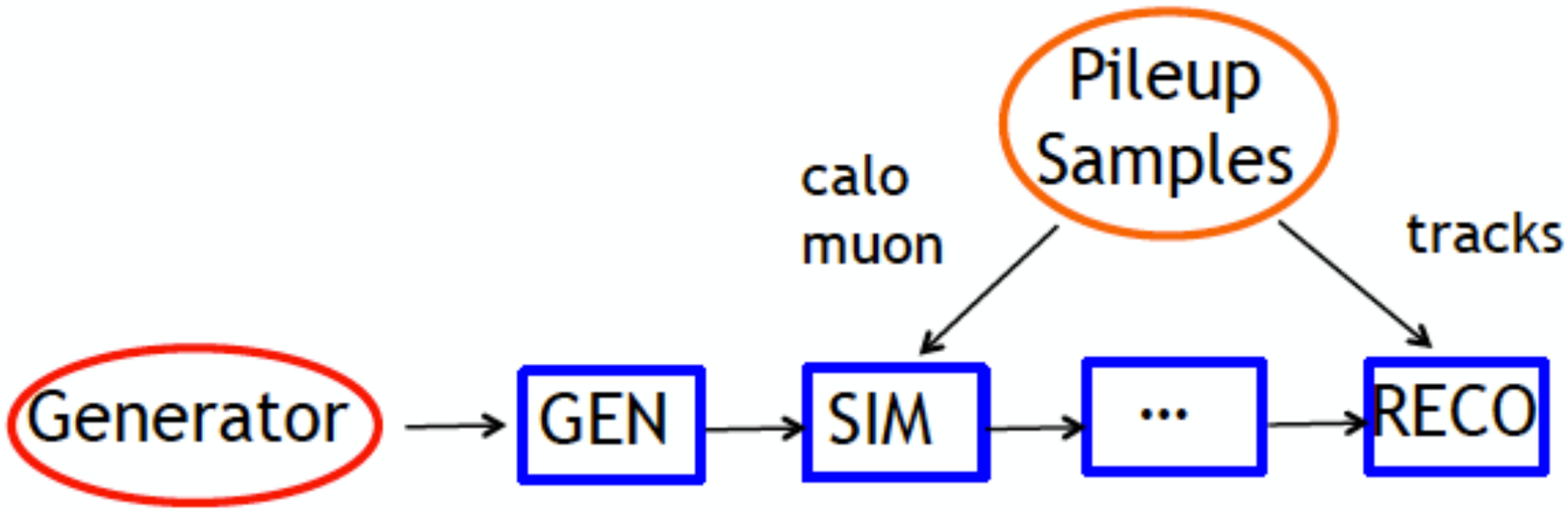


Thanks to all CMS FastSim developers!

Back up

Pile up treatment

Pile up (PU) events come from a minimum bias sample generated separately with FastSim.

Some number of PU events are mixed into each event.

- Calorimeters and muon systems: mix SimHits - in time and out-of-time PU. Exactly like FullSim.
- Tracker: mix reconstructed tracks - in time PU only.