Fast Simulation for a muon collider

Michele Selvaggi

CERN

Detector Simulation

- Full simulation (GEANT):
  - **simulates** all **particle-detector interaction** (e.m/hadron showers, nuclear interaction, brem, conversions)

10^2-10^3 s/ev

- Experiment Fast Simulation (ATLAS, CMS ..)
  - **simplify** geometry, smear at the level of detector hits, frozen showers

10-10^2 s/ev

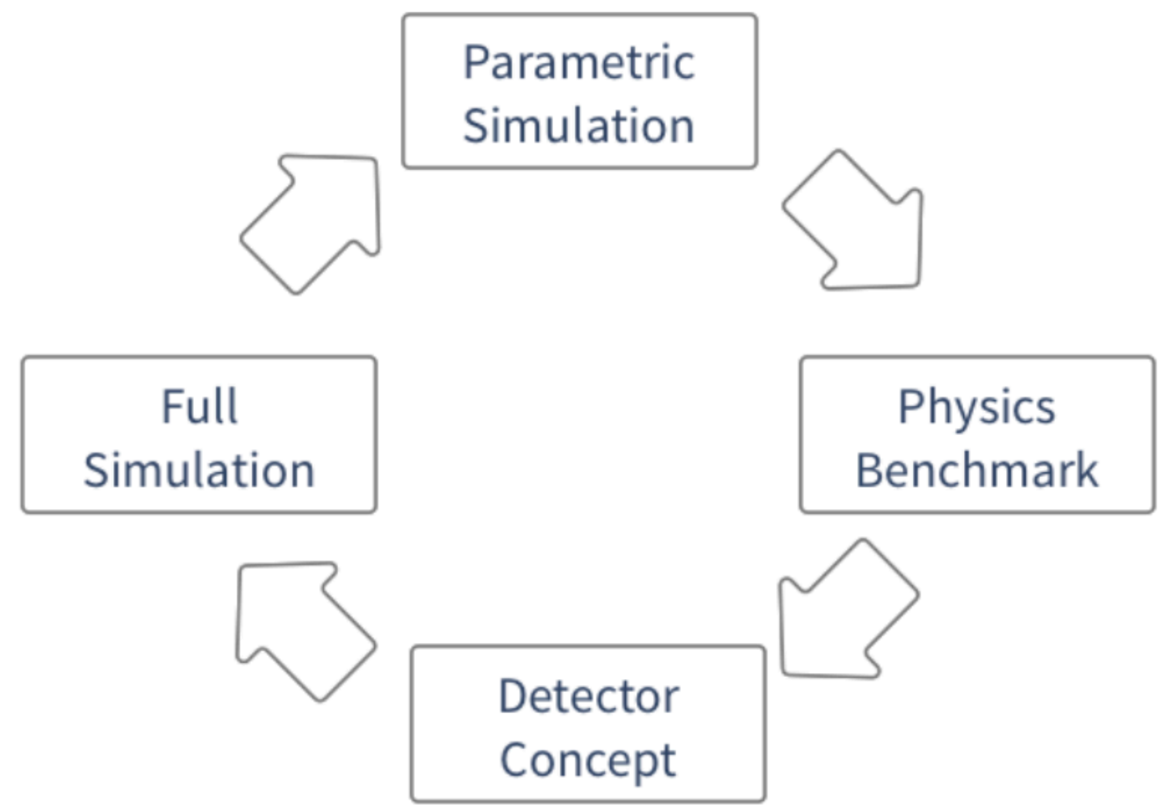
- Parametric simulation (**Delphes**, PGS):
  - **parameterise detector response** at the particle level (efficiency, resolution on tracks, calorimeter objects)
  - reconstruct **complex objects** and observables (use particle-flow, jets, missing ET, pile-up ..)

10^{-2} - 10^{-1} s/ev

Parametric simulation paradigm for future colliders

Why fast parametric detector simulation?

- Easily scan detector parameters
- Reverse engineer detector that maximises performance
- Preliminary sensitivity studies for key physics benchmarks

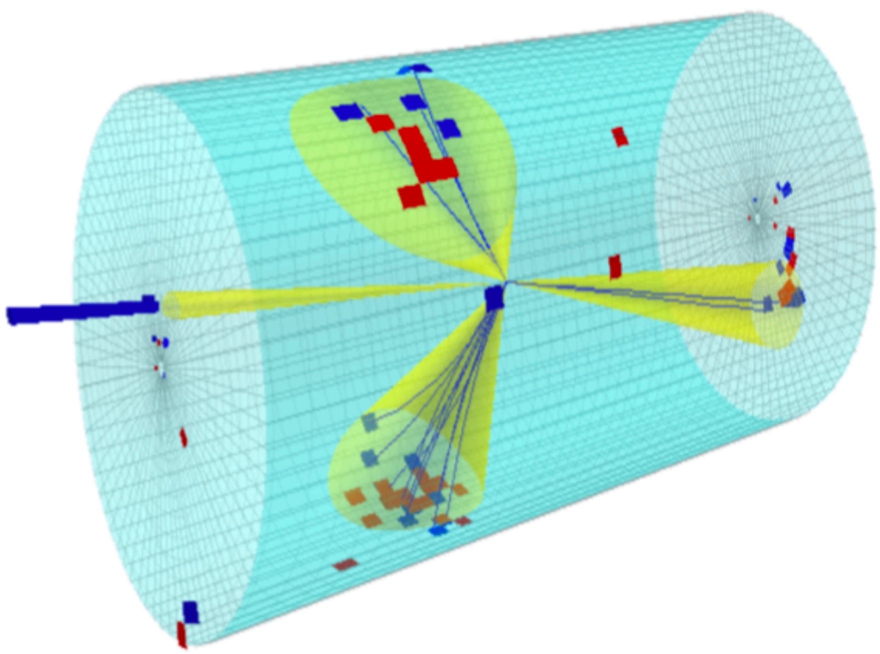


→ (usual) paradigm adopted in the context of FCC studies

Delphes in a nutshell

- **Delphes** is a modular framework that simulates the response of a **multipurpose detector** in a parameterised fashion

- Includes:
  - pile-up
  - charged particle propagation in B field
  - EM/Had calorimeters
  - particle-flow



- Provides:
  - leptons, photons, neutral hadrons
  - jets, missing energy
  - heavy flavour tagging

  - designed to deal with hadronic environment
  - well-suited also for e+e- studies
  - detector cards for: CMS (current/PhaseII) - ATLAS - LHCb - FCC-hh - ILD - CEPC - FCCee (IDEA/CLD) - CLICdet

Philosophy

- The interest in the TH/pheno community is to **assess the physics reach at the highest possible energies** $\sqrt{s}$ = 10, 14, (30, 100) TeV

- Need to be able to reconstruct: muons, electrons, photons, jets, tops, V up to **at the highest possible energies ($\sqrt{s}/2$)**
  - $\mu\,\mu \rightarrow \mu\,\mu$ / e e / j j / t t~ (hadronic) ,V V (hadronic)
  - $\mu\,\mu \rightarrow$ EW-inos, stops $\rightarrow$ SM

With many respects, the **constraints from physics** at **moderate / high p_T** are similar to **CLIC/ FCC-hh**

The boosted regime

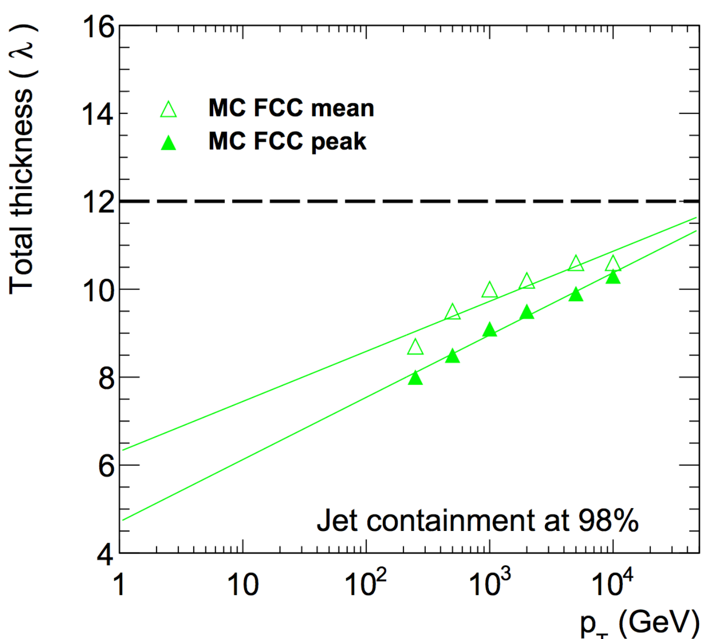
→ measure leptons, jets, photons, muons originating from multi-TeV resonances

Tracking: $\frac{\sigma(p)}{p} \approx \frac{p\sigma_x}{BL^2}$

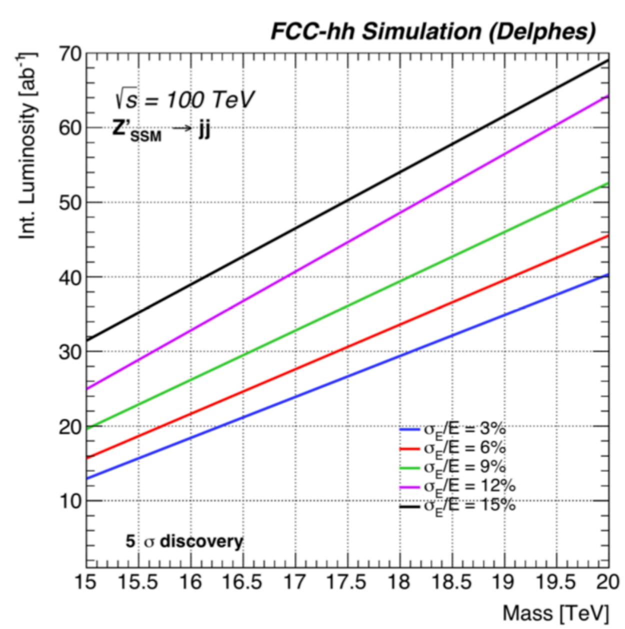
Calorimeters: $\frac{\sigma(E)}{E} \approx \frac{A}{\sqrt{E}} \bigoplus B$

- Tracking target : **σ / p = 20% @10 TeV**
- Muons target: **σ / p = 20% @20 TeV**

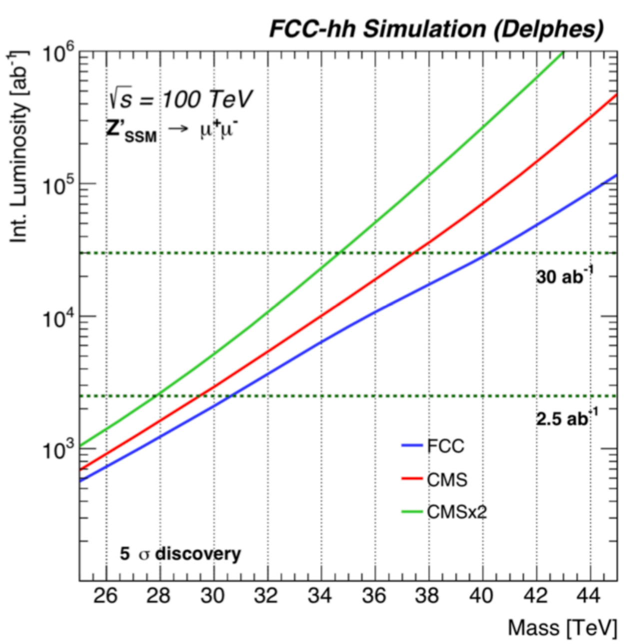
- Calorimeters target: containment of p_T = 20 TeV jets



≳11 λ_I for EM + Had



high p_T jets



high p_T muons

The boosted regime

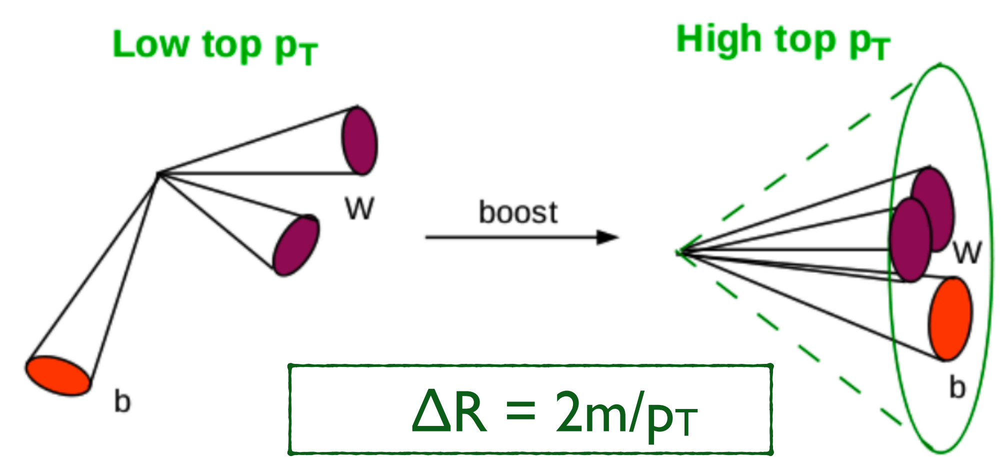
- The boosted regime:
  → measure W, H, top jets from multi-TeV resonances

  - Highly boosted hadronically decaying SM heavy states (W, Z, H or t) will have highly collimated decay products

  - The ability to distinguish such boosted states from vanilla QCD jets is an essential tool in many searches for BSM (such as top partners, Z', etc …)

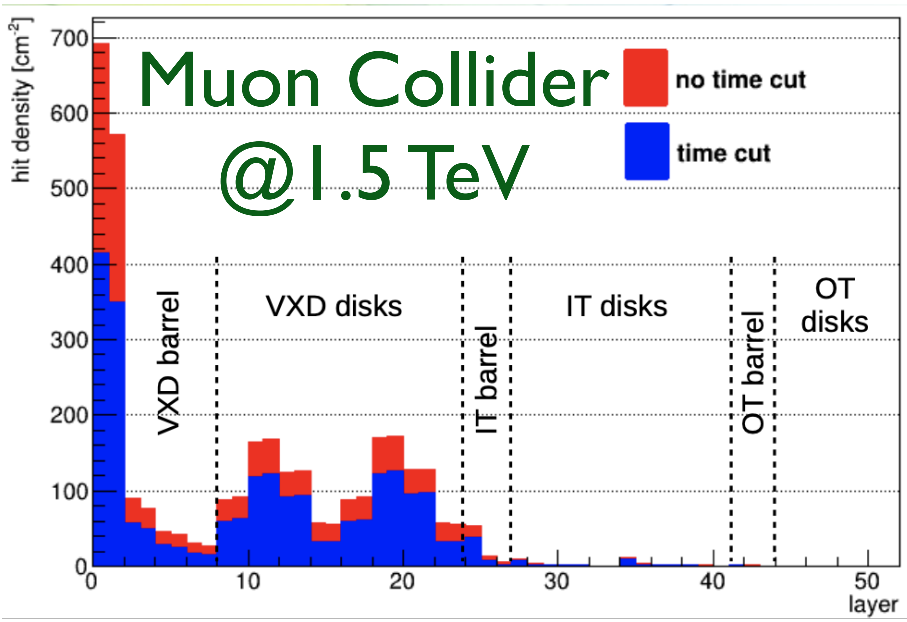
    ex: W(10 TeV) will have decay products separated by DR = 0.01 = 10 mrad

- need highly granular sub-detectors:
  - Tracker - pixel:10 µm @ 2cm → $\sigma_{\eta x \varphi} \approx$ 5 mrad
  - Calorimeters: 2 cm @ 2m → $\sigma_{\eta x \varphi} \approx$ 10 mrad



BIB vs FCC-hh

@first pixel ~ 2 cm from beam-pipe



charged fluence: 400-700 (cm^{-2} / BX)

Barrel layer:	1	2	3	4	5	6
Average radius [mm]	25	60	100	150	260	380
Maximum fluence [cm^{-2}]	328.1	79.7	35.1	16.9	6.8	3.3
Module occupancy [%]	1.63	0.39	0.18	0.10	0.28	0.15
Data size per bunch crossing [Mb]	56.60	37.66	28.51	23.46	10.95	8.72
Data rate [Tb/s]	2263.1	1506.4	1140.3	938.5	438.0	348.6
Data rate density @ 40 MHz [Gb/s/cm^{-2}]	944.0	229.6	107.0	60.2	14.8	8.0
Data rate density @ 1 MHz [Gb/s/cm^{-2}]	23.6	5.7	2.7	1.5	0.4	0.2
	7	8	9	10	11	12
	530	742	937	1132	1327	1540
	1.9	0.83	0.46	0.26	0.16	0.13
	0.09	0.04	3.0	1.9	1.3	0.9
	835.5	537.8	331.3	249.0	192.8	109.5
	20.8875	13.445	8.2825	6.225	4.82	2.7375
	5.1	2.4	1.2	0.7	0.5	0.2
	0.1	…				

FCC-hh

Table 5: Summary of maximum fluence [cm^{-2}], module occupancy, data size per bunch crossing [Mb/s], data rates [Tb/s] and data rate densities [Gb/s/cm^{-2}], as estimated for the nominal FCC-hh pile-up of 1000 events and tracker in flat layout [28].

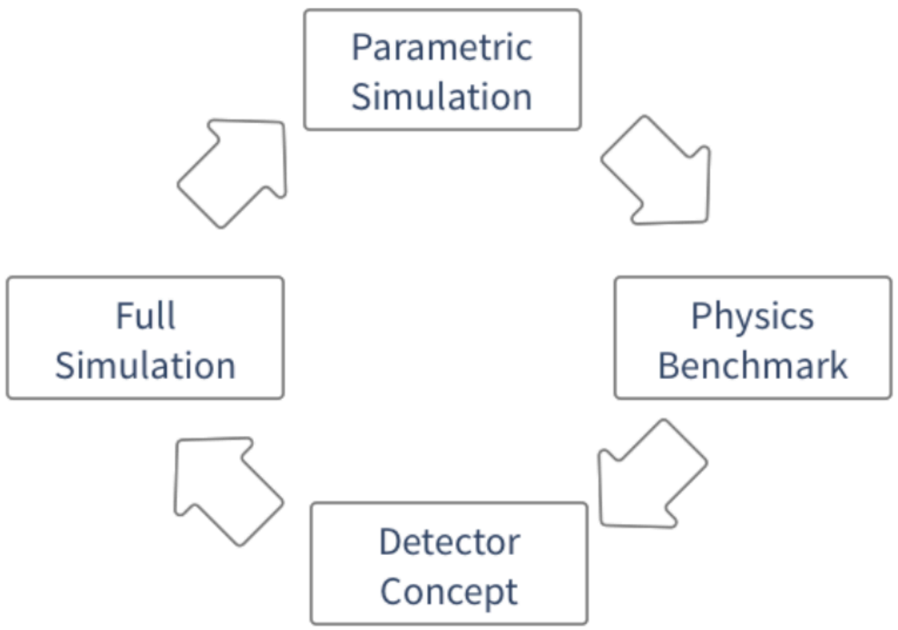
charged fluence: 330 (cm^{-2} / BX)

- At a muon collider (with nozzle shielding) the # hits / event x2 larger than at the FCC-hh @1000 pile-up (700 cm^{-2} vs 350 cm^{-2})
- However, the **collision rate is x1000 smaller** (70 kHz vs 40MHz)
  - can afford low power, low mass, highly granular pixel detector (MAPs)

  → occupancy: 0.6% (700 / (1cm^2 / 30 μm^2)) ~ **2x HL-LHC** or **0.5x FCC-hh**
- **non-pointing** background (more handles)
- situation improves a higher energies (fewer muon decays)

Philosophy

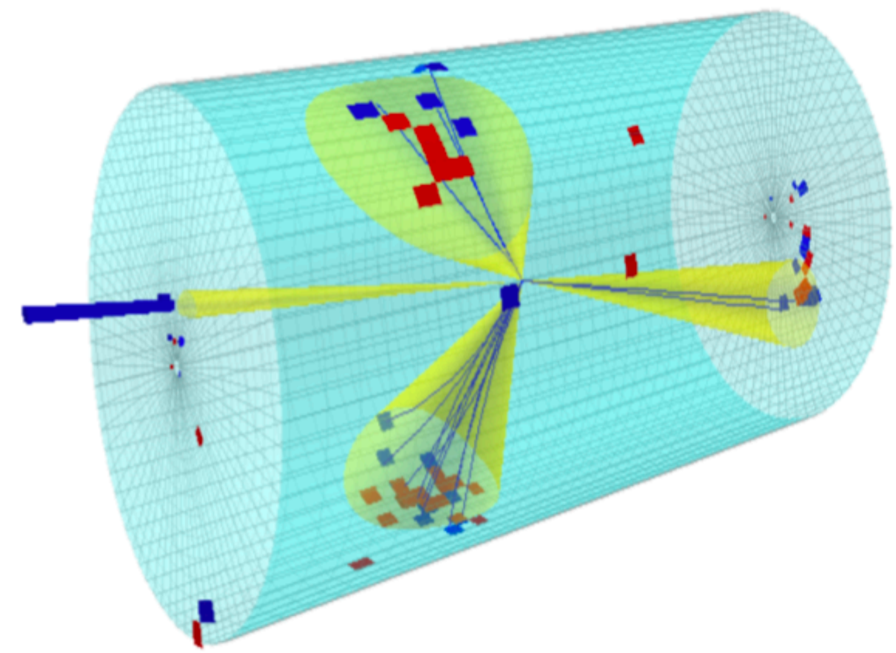
- **Goal** of the Delphes card (and physics studies):
  - **define a target** for the detector performance (free of BIB)
  - **study benchmark physics** channels with **target performance**
  - study impact of **variations of detector performance** around nominal on physics
  - **iterate** on detector design



The nominal MuCol detector card is a **hybrid** between the **FCC-hh** and **CLICdet**

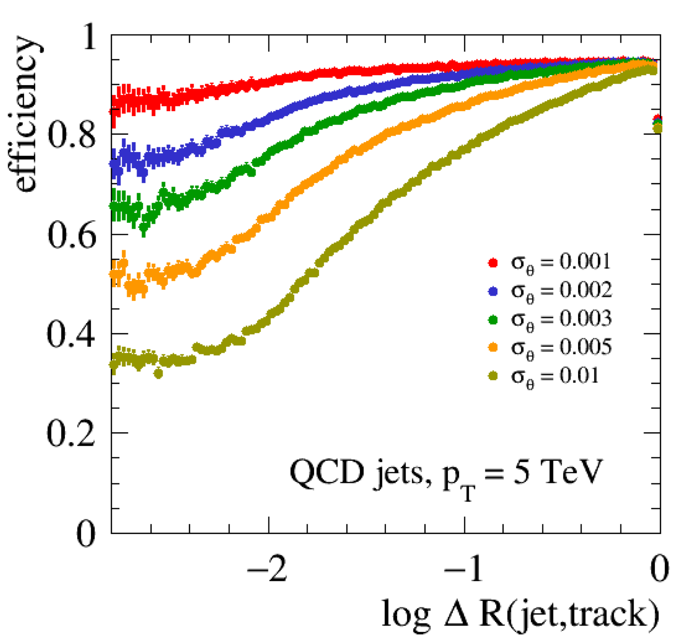
Particle Propagator/DTF

```
#################################
# Propagate particles in cylinder
#################################

module ParticlePropagator ParticlePropagator {
    set InputArray Delphes/stableParticles

    set OutputArray stableParticles
    set ChargedHadronOutputArray chargedHadrons
    set ElectronOutputArray electrons
    set MuonOutputArray muons

    # radius of the magnetic field coverage in the calorimeter, in m
    set Radius 1.5
    # half-length of the magnetic field coverage in the calorimeter, in m
    set HalfLength 2.31

    # magnetic field, in T
    set Bz 4.0
}
```



from CLICdet

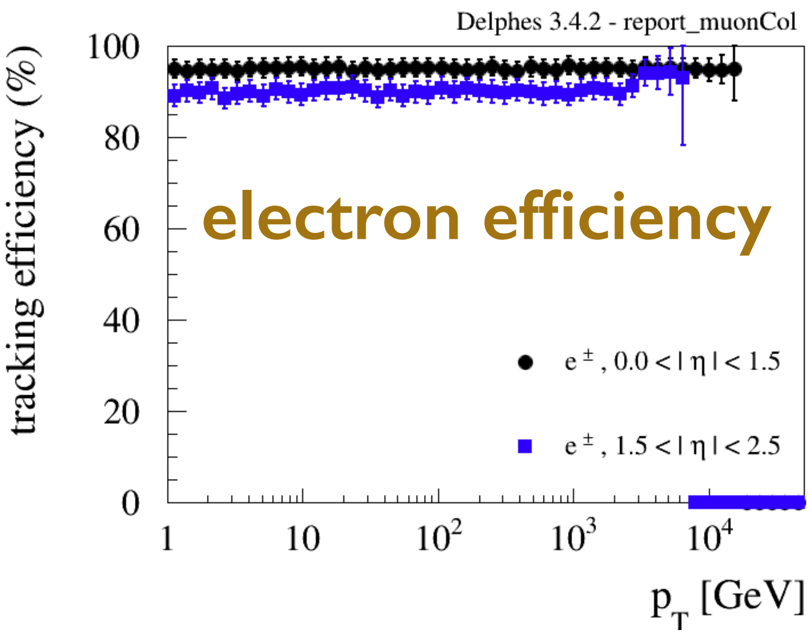
```
####################
#   Dense Track Filter
####################

module DenseTrackFilter DenseTrackFilter {

  set TrackInputArray DenseMergeTracks/tracks

  set TrackOutputArray tracks
  set ChargedHadronOutputArray chargedHadrons
  set ElectronOutputArray electrons
  set MuonOutputArray muons

  set EtaPhiRes 0.003
  set EtaMax 2.5
```

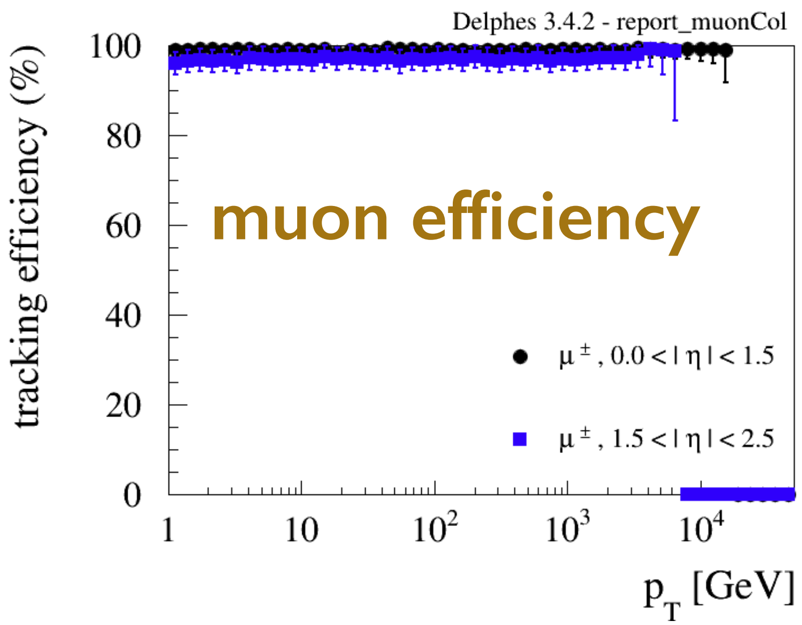


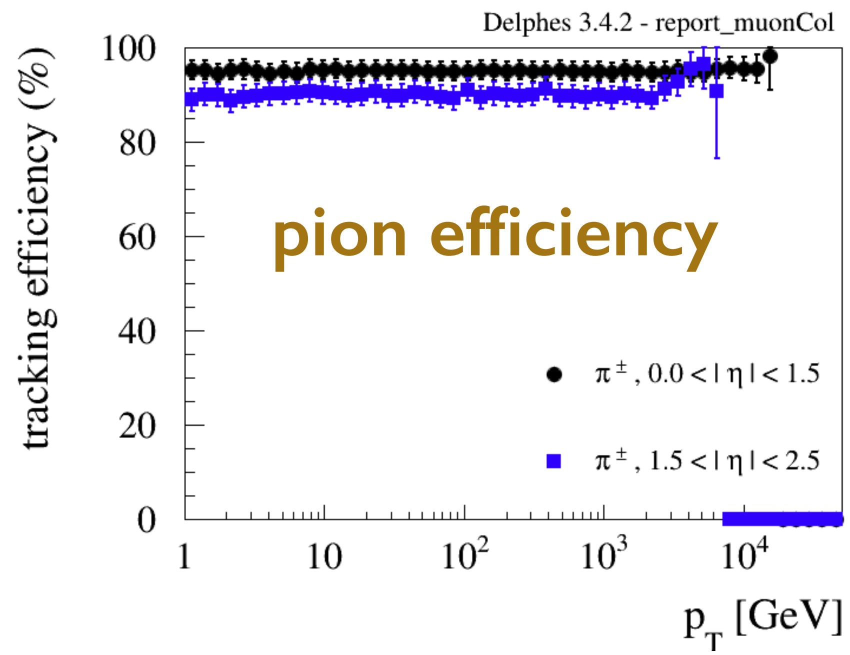
Modelling the track efficiency loss within highly boosted jets

from FCC-hh

Tracking efficiency/resolution

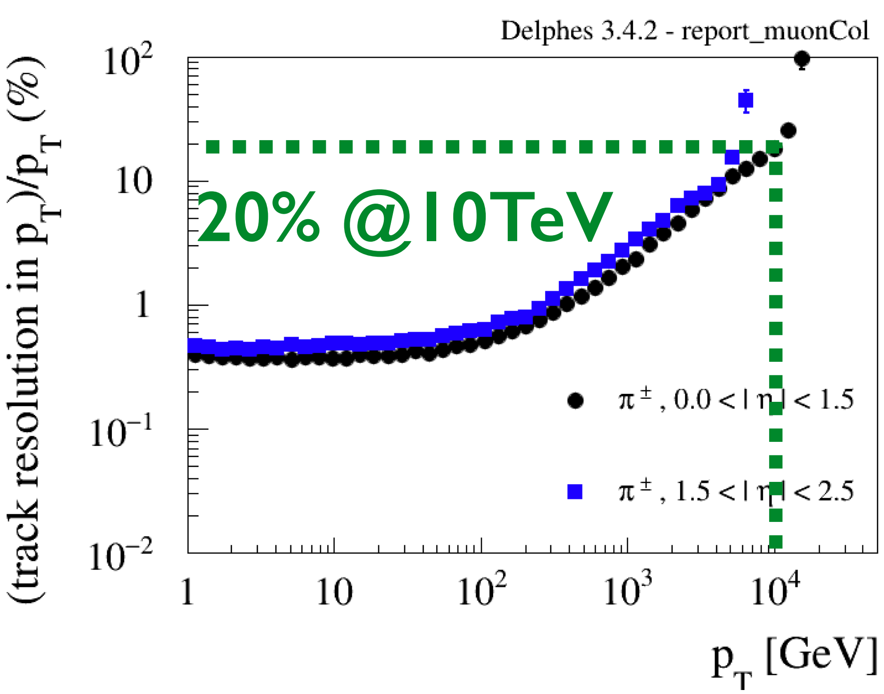




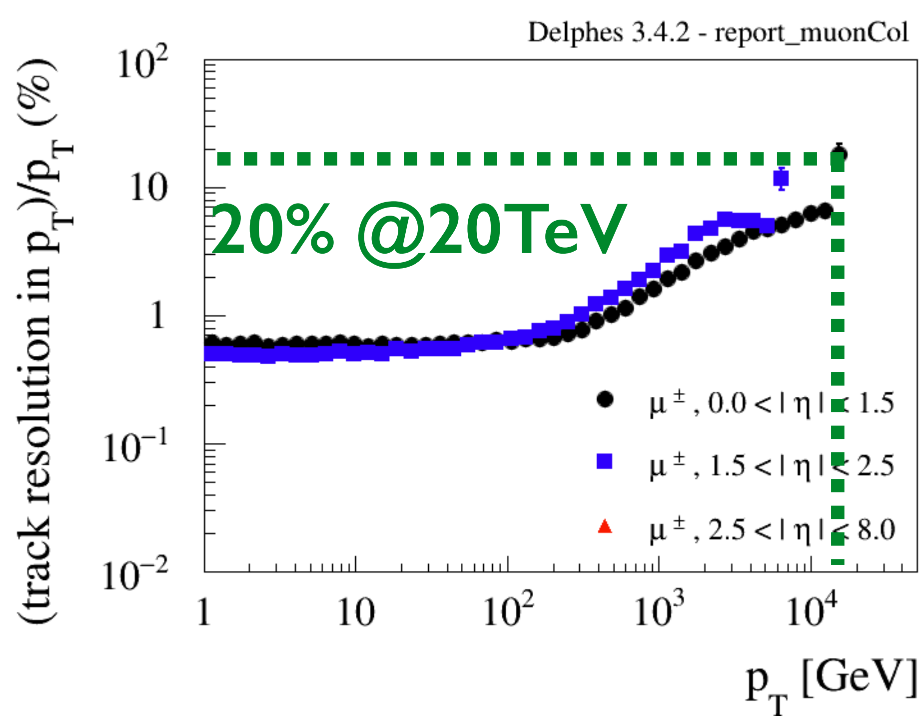


inspired from FCC-hh

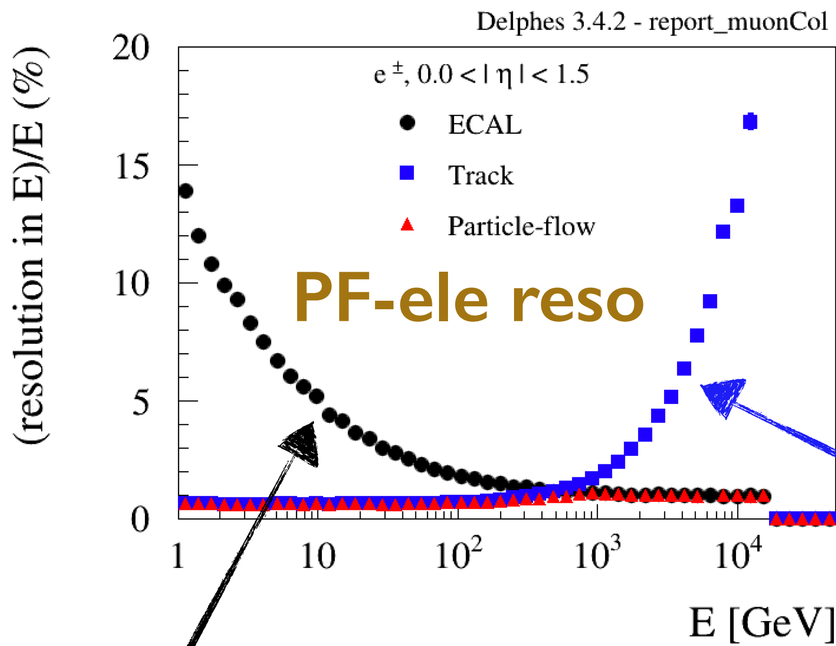
track resolution

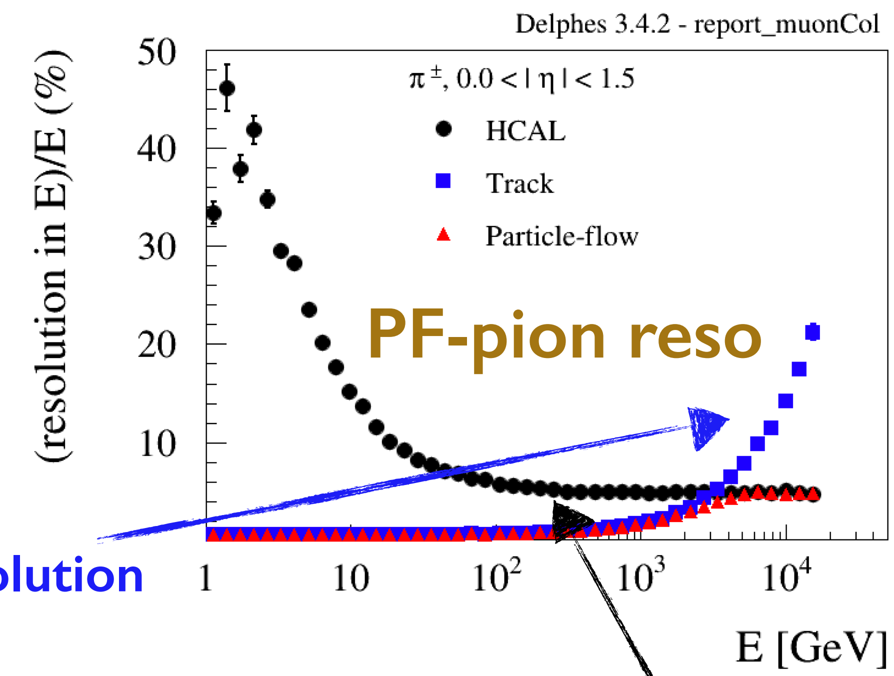


muon resolution



Calorimeters/Particle-Flow reco



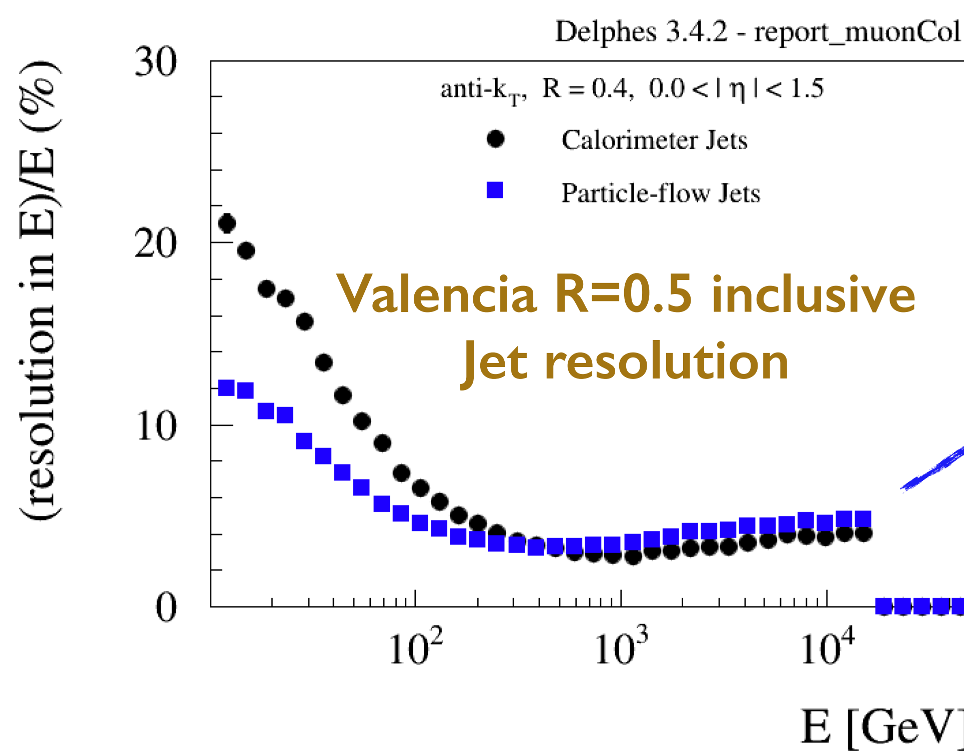


Tracking resolution from FCC-hh

Calorimeters inspired from CLIC

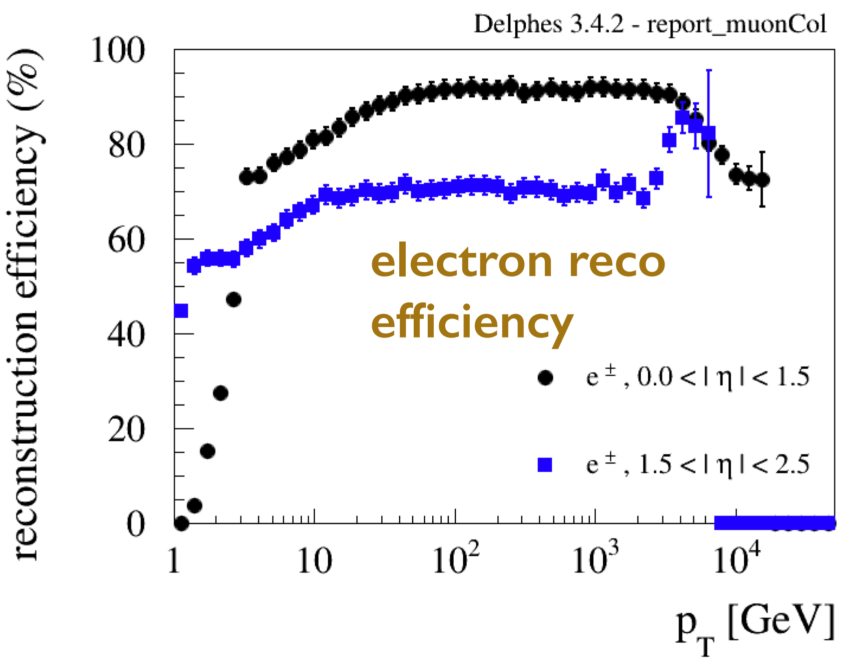
EM resolution from CLICdet

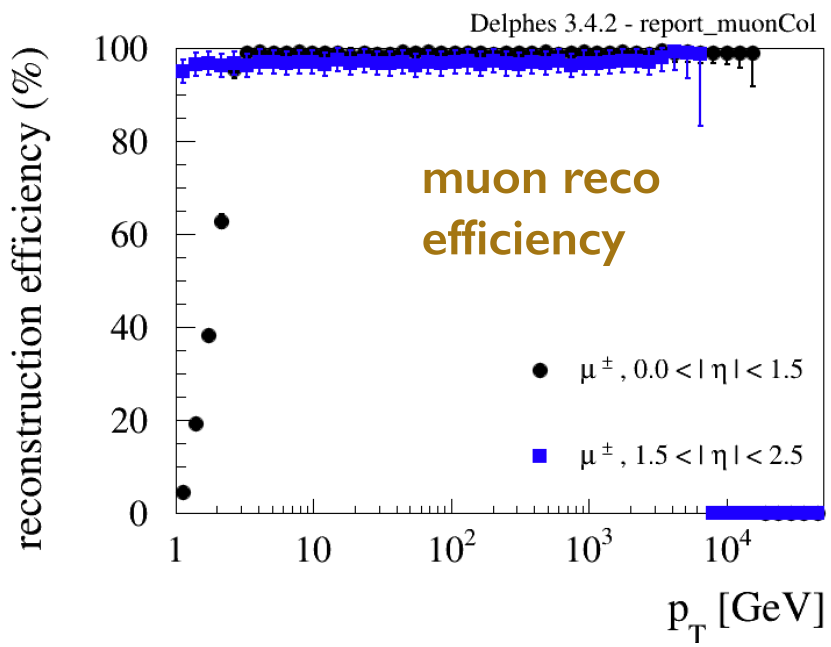
Hadronic resolution from CLICdet

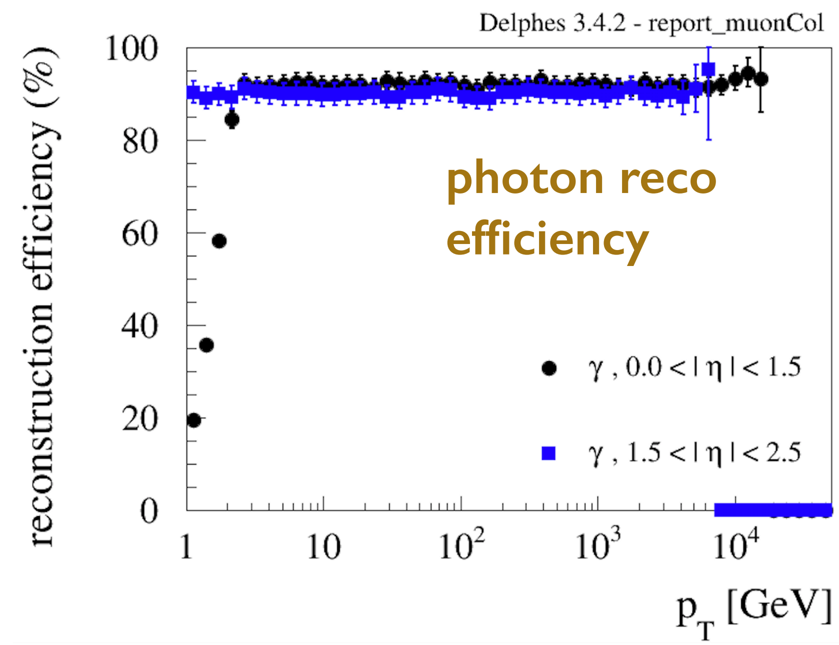


PF jet include BIB smearing from CLIC stage 3

E/mu/gamma efficiency



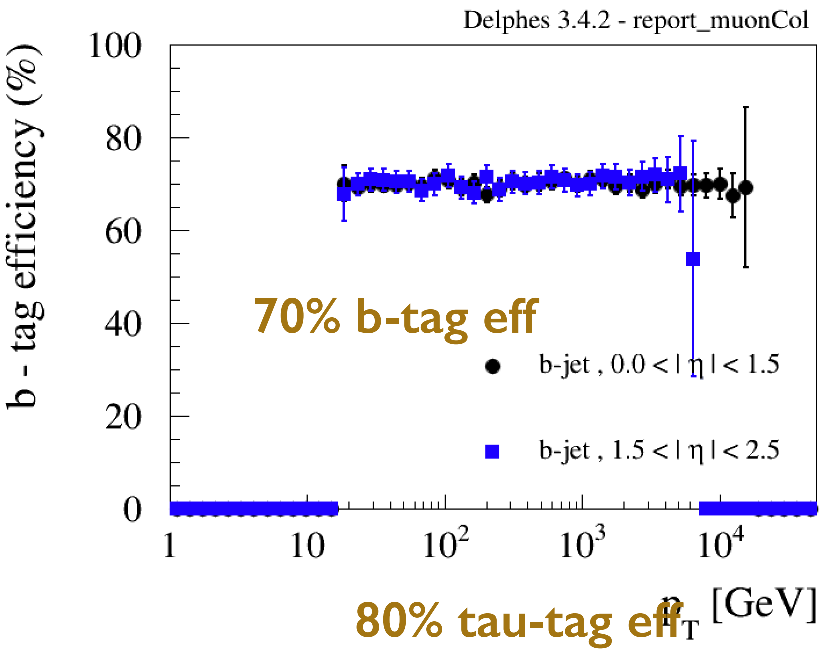


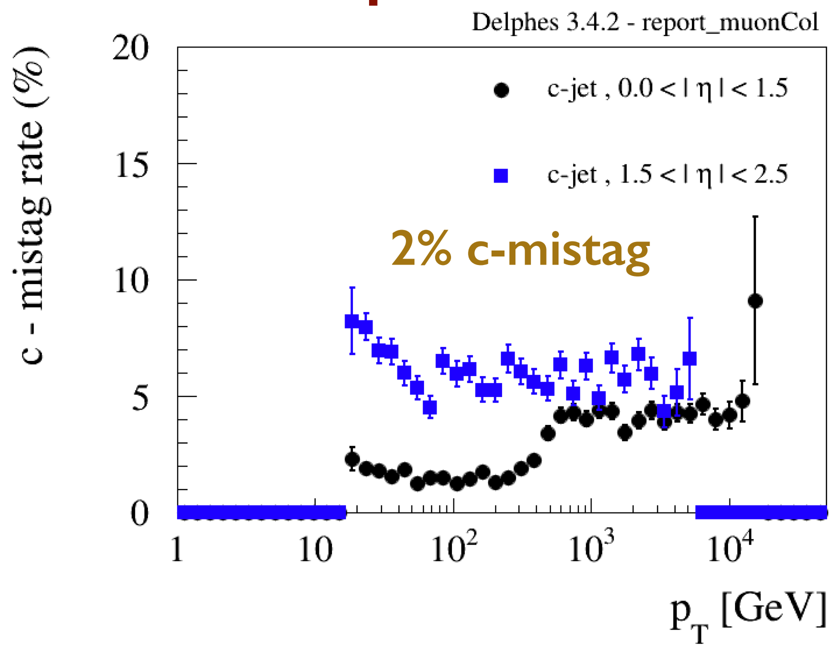


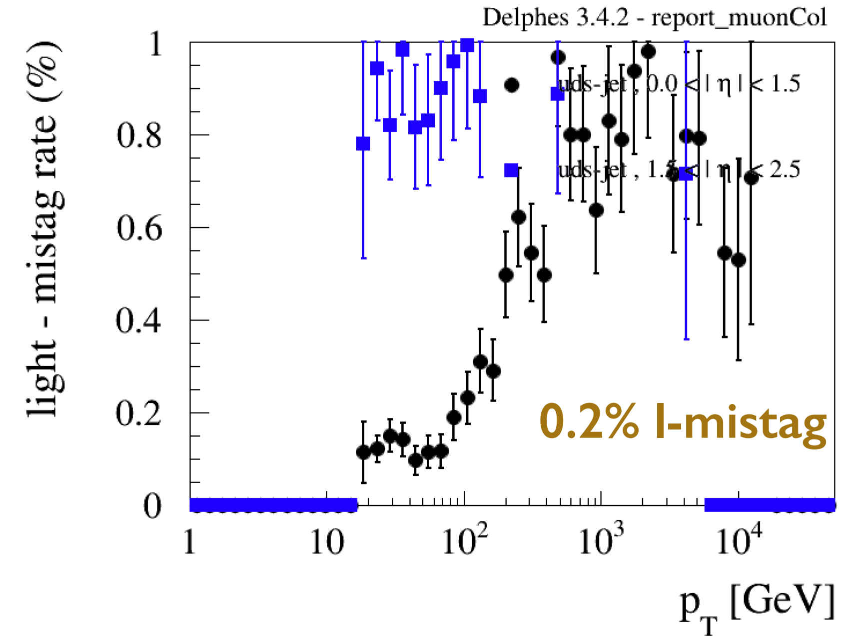
inspired from CLIC det

BTagging and Tau Tagging

inspired from CLIC det

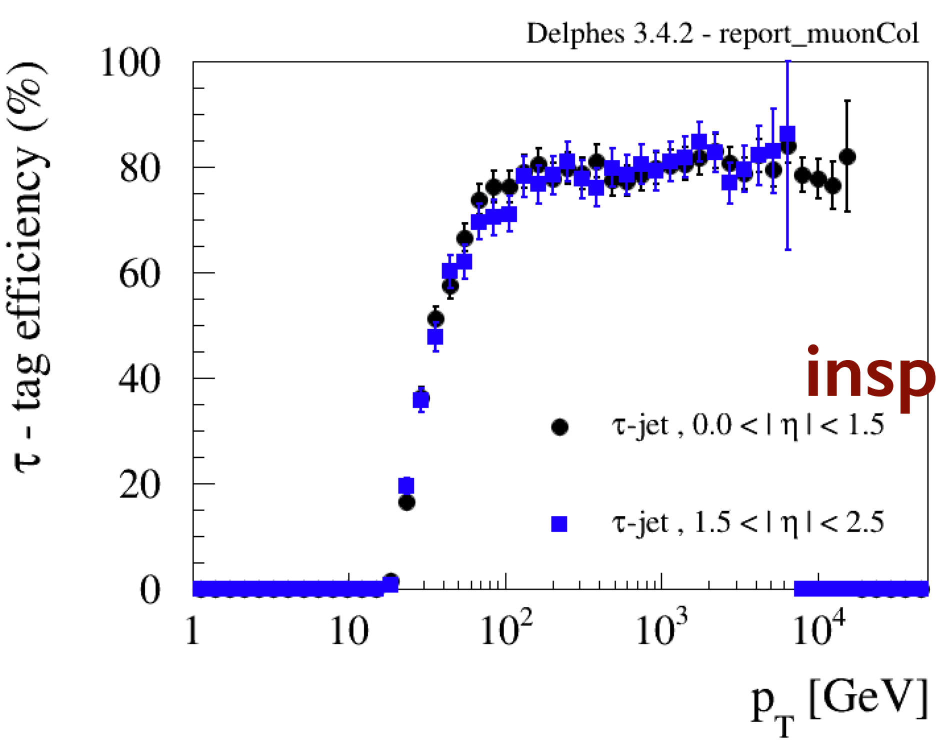




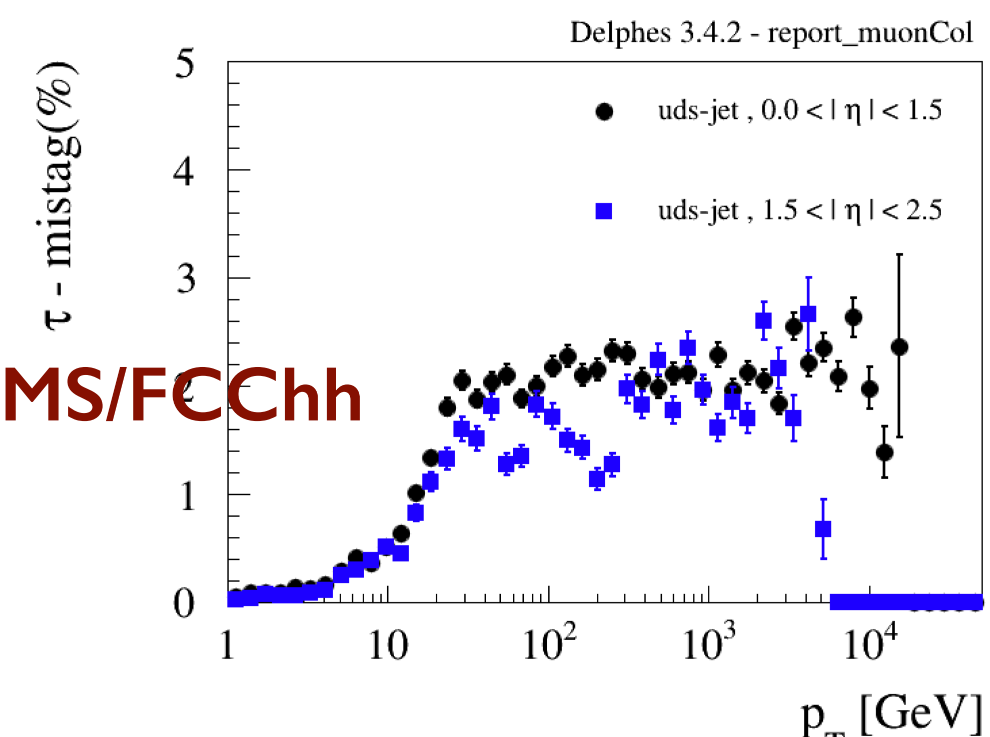


Loose, **Medium** and **Tight** working points are defined (only medium shown here…)

2% l-mistag



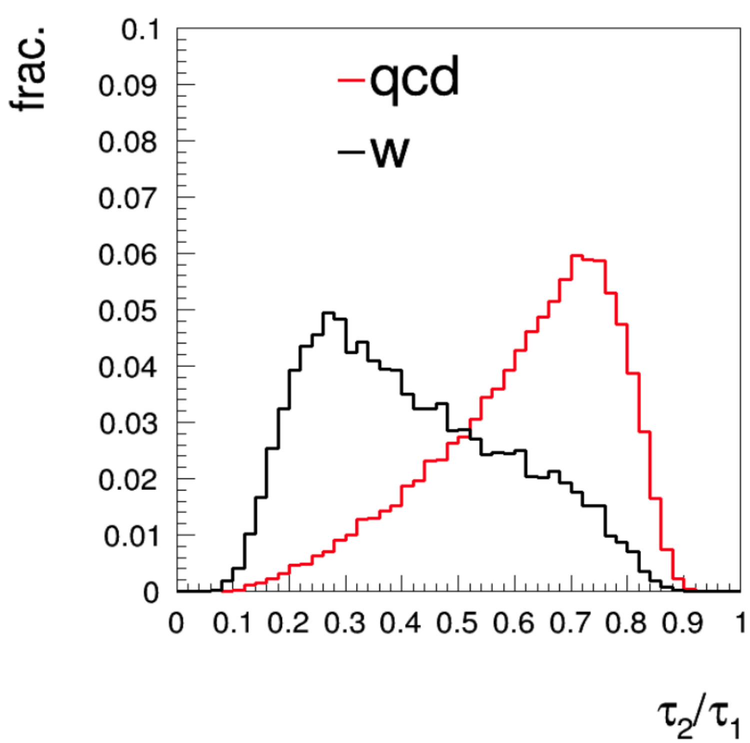
inspired from CMS/FCChh



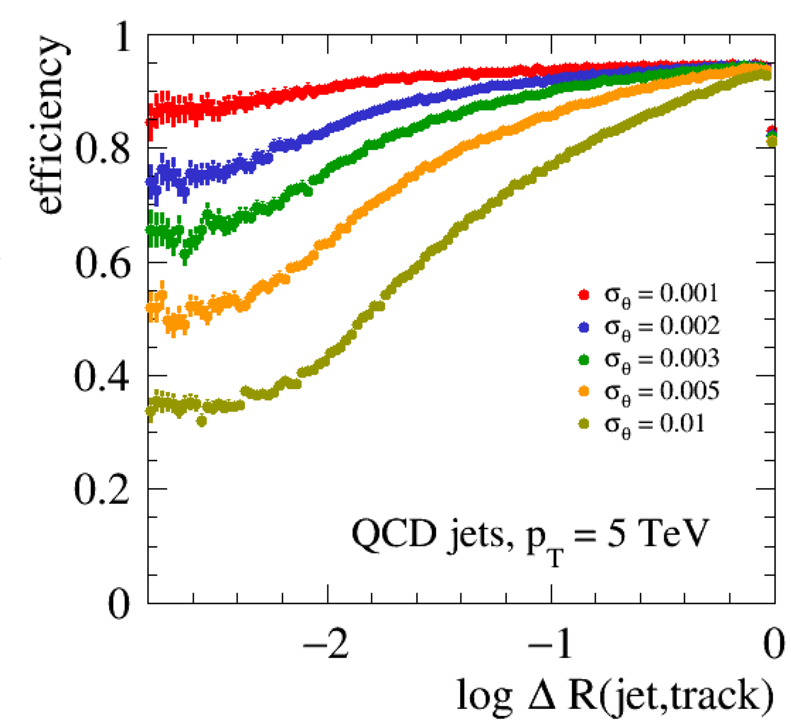
Jets and Substructure

```
############
# Jet finder
############

module FastJetFinder FatJetFinder {
#  set InputArray TowerMerger/towers
  set InputArray EFlowMerger/eflow

  set OutputArray jets

  set JetAlgorithm 5
  set ParameterR 0.8

  set ComputeNsubjettiness 1
  set Beta 1.0
  set AxisMode 4

  set ComputeTrimming 1
  set RTrim 0.2
  set PtFracTrim 0.05

  set ComputePruning 1
  set ZcutPrun 0.1
  set RcutPrun 0.5
  set RPrun 0.8

  set ComputeSoftDrop 1
  set BetaSoftDrop 0.0
  set SymmetryCutSoftDrop 0.1
  set R0SoftDrop 0.8

  set JetPTMin 200.0
}
```

- FastJet performs **jet clustering** via the FastJetFinder module
- Most used Jet substructure algorithms are included (N-subjettiness, SoftDrop, Trimming, Pruning …)

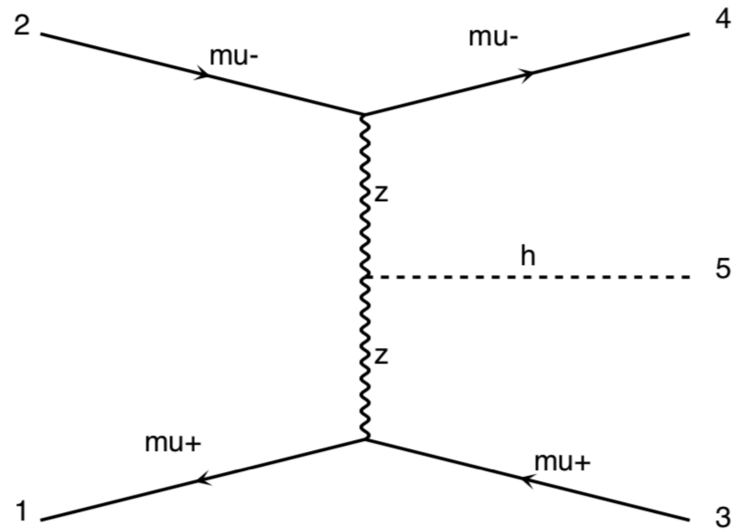


Modelling the track efficiency loss within highly boosted jets

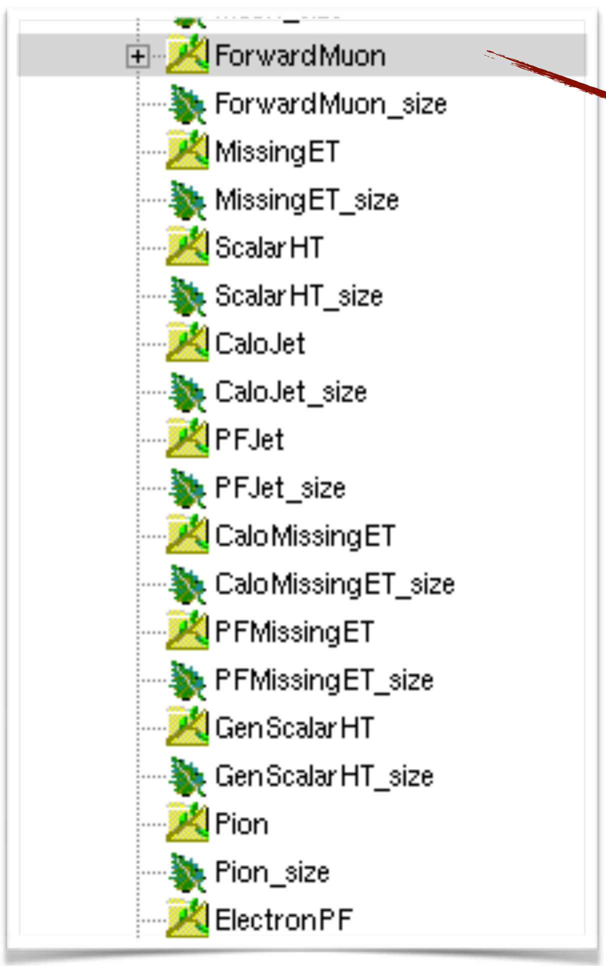


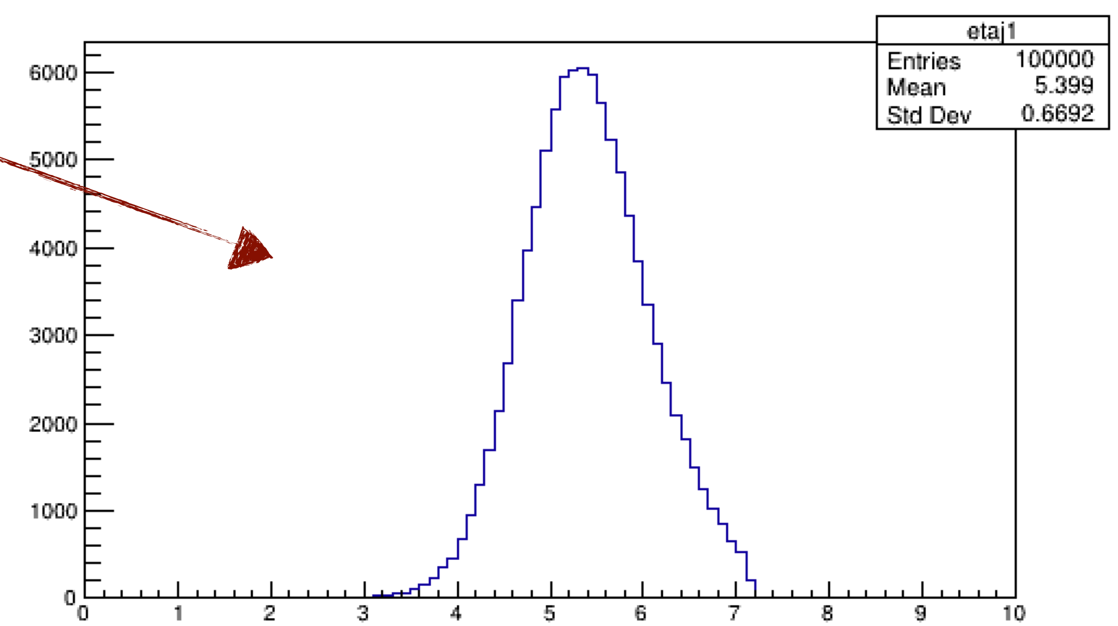
Forward muon collection

- **Neutral vector boson scattering** produces multi-TeV forward muons $|\eta| \approx 5.5$



- Forward **Muon collection:**
  - $2.5 < |\eta| < 8.0$
  - energy resolution: 10%



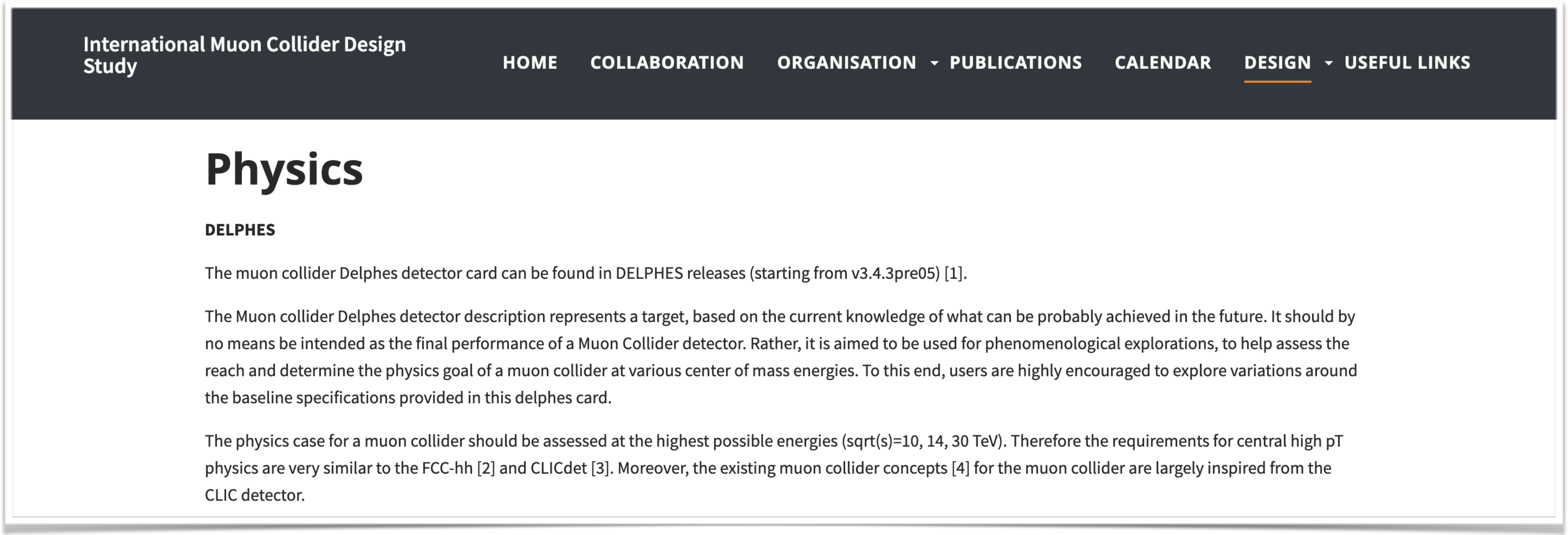


most forward muon $|\eta|$

Possible detector variations

- **p_T acceptance** of final state objects (pt = [10-50]), in particular HH→4b
- **angular detector acceptance:** the baseline detector card assumes a maximum rapidity of eta=2.5. Ranges between [1.5, 3.0] can be studied. This simulates various assumptions on the dead cone introduced by the nozzle shielding.
- **forward muon detector performance:** no detector concept currently exists for reconstructing muons in the challenging BIB environment at small angles. Both the acceptance and the resolution for reconstructing such muons can be explored. This can studied in the context of neutral vector boson scattering.
- **Track and Calorimeter resolutions** can be degraded by factor 2-4 in physics studies that involve resonant signals. Alternatively, the jet energy resolution can also be degraded by similar factors. This can be studied for instance in the context of double and triple Higgs production in fully hadronic final states.
- **Identification efficiencies,** in particular lepton, photons ID, and heavy flavour tagging. This can be also be studied for instance in the context of double and triple Higgs production where b/c/light flavour discrimination can be important.
- **detector volume, timing resolution,** track reconstruction efficiency as a function of **displacement** for **LLP studies** and exotic signatures

Muon Collider baseline detector card


International Muon Collider Design Study

HOME COLLABORATION ORGANISATION PUBLICATIONS CALENDAR DESIGN USEFUL LINKS

Physics

DELPHES

The muon collider Delphes detector card can be found in DELPHES releases (starting from v3.4.3pre05) [1].

The Muon collider Delphes detector description represents a target, based on the current knowledge of what can be probably achieved in the future. It should by no means be intended as the final performance of a Muon Collider detector. Rather, it is aimed to be used for phenomenological explorations, to help assess the reach and determine the physics goal of a muon collider at various center of mass energies. To this end, users are highly encouraged to explore variations around the baseline specifications provided in this delphes card.

The physics case for a muon collider should be assessed at the highest possible energies (sqrt(s)=10, 14, 30 TeV). Therefore the requirements for central high pT physics are very similar to the FCC-hh [2] and CLICdet [3]. Moreover, the existing muon collider concepts [4] for the muon collider are largely inspired from the CLIC detector.

- "Final" v0 can be found here:
  - https://github.com/delphes/delphes/blob/master/cards/delphes_card_MuonColliderDet.tcl
  - https://github.com/delphes/delphes/tree/master/cards/MuonCollider

- A description can be found in:
  - https://muoncollider.web.cern.ch/node/14

Final comments

- The performance that has been encoded in the Delphes muon collider card is to be intended as a **"target" performance** for the highest energies

However:

- Nothing will be written in stone, this detector card only represents a **baseline** → should be intended as a **moving target**
- Users should **explore variations around target performance** to assess **physics reach** as a function of particular **detector choices**, and impact of beam induced background
- Ultimately, desired precision on physics will determine detector design.

Backup

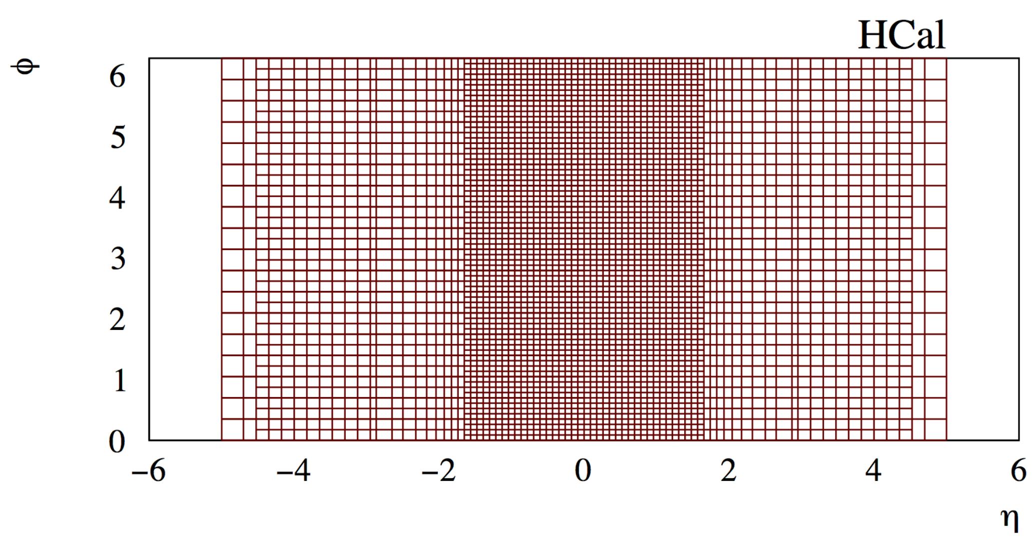
Calorimetry

- ECAL/HCAL **segmentation** specified in (η,ϕ) coordinates
- Particles that reach calorimeters **deposits fixed fraction of energy** in f_{EM} (f_{HAD}) in ECAL(HCAL)

particles	f_{EM}	f_{HAD}
e γ π^0	1	0
Long-lived neutral hadrons (K^0_s , Λ^0)	0.3	0.7
ν μ	0	0
others	0	1

- Particle **energy** and **position** is **smeared** according to the calorimeter it reaches

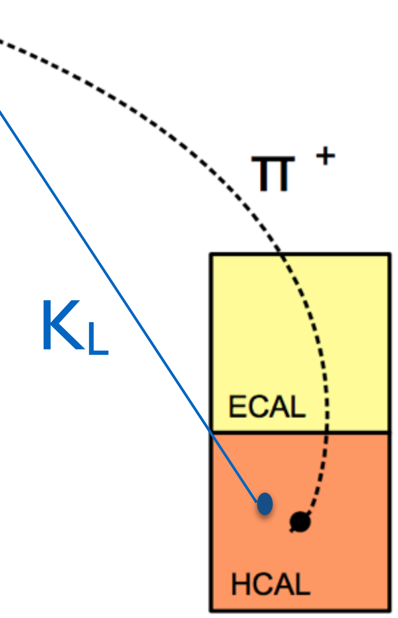
$$\left(\frac{\sigma}{E}\right)^2 = \left(\frac{S(\eta)}{\sqrt{E}}\right)^2 + \left(\frac{N(\eta)}{E}\right)^2 + C(\eta)^2$$



Particle-Flow

- Given **charged track** hitting **calorimeter cell**:
  - is deposit more compatible with **charged only** or **charged + neutral** hypothesis?
  - how to **assign momenta** to resulting components?

- We have two measurements (E_{trk}, σ_{trk}) and (E_{calo}, σ_{calo})
- Define $E_{Neutral} = E_{calo} - E_{trk}$



Algorithm:

- If $E_{neutral}/\sqrt{(\sigma_{calo}^2 + \sigma_{trk}^2)} > S$:
  → create PF-neutral particle + PF-track

- Else:
  create PF-track and **rescale momentum** by combined calo+trk measurement

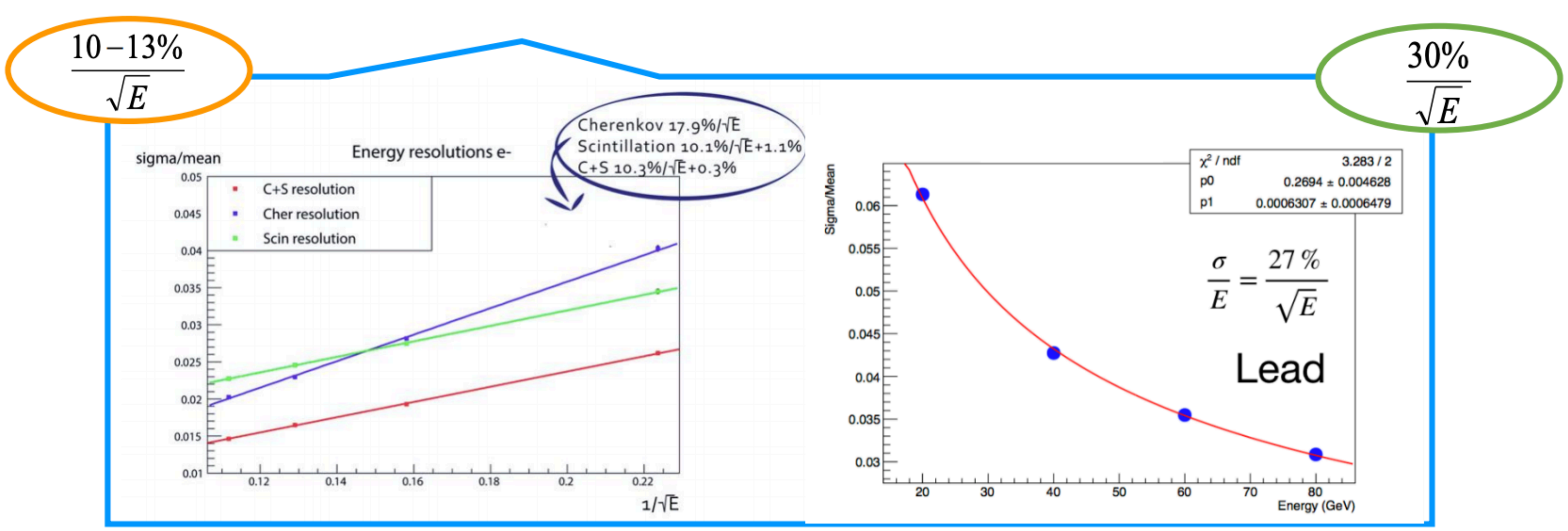
  - EM (had) deposit 100% in ECAL (HCAL)
  - No propagation in calorimeters
  - No clustering (topological) clustering, exploiting pre-defined grid

IDEA - (DR) Calorimetry

```
#############
# Calorimeter
#############
module DualReadoutCalorimeter Calorimeter {
  set ParticleInputArray ParticlePropagator/stableParticles
  set TrackInputArray TrackMerger/tracks
```

```
# Lists of the edges of each tower in eta and phi;
# each list starts with the lower edge of the first tower;
# the list ends with the higher edged of the last tower.
# Barrel:  deta=0.02 towers up to |eta| <= 0.88 ( up to 45°)
# Endcaps: deta=0.02 towers up to |eta| <= 3.0 (8.6° = 100 mrad)
# Cell size: about 6 cm x 6 cm
```

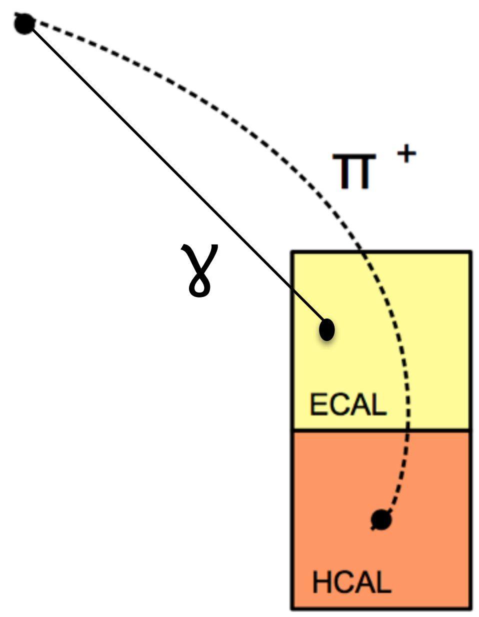
- If EM > 0 and E_{had} =0 → σ(EM)
    - e.g. ɣ
- If E_{had} > 0 → σ(had)
    - e.g. π^+ or (ɣ,π^+)



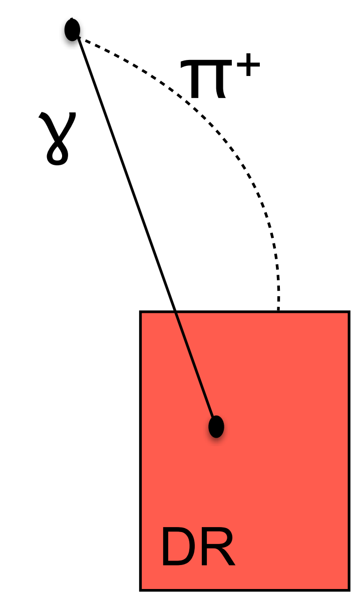
E. Fontanesi
L. Pezzotti
M. Antonello

Dual Readout Particle-Flow

- Given **charged track** hitting **calorimeter cell**:
  - is deposit more compatible with **charged only** or **charged + neutral** hypothesis?
  - how to **assign momenta** to resulting components?
  - If **charged + neutral** how to associate particle ID to **charged** and **neutral** components, e.g **(γ,π^+)** or **($e+$,K_L)** ?

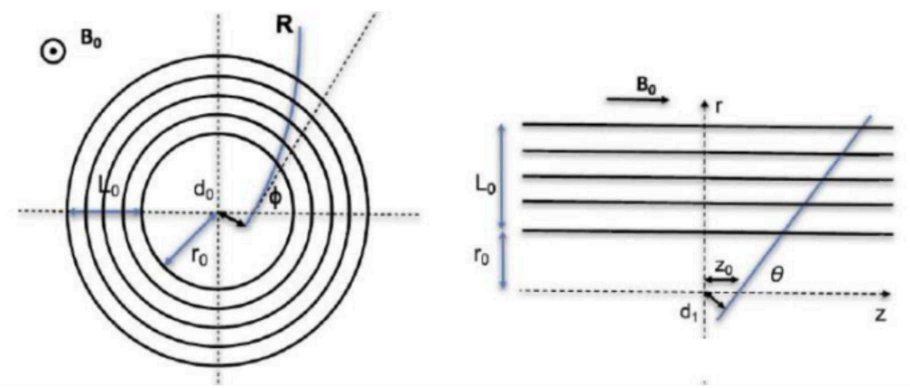


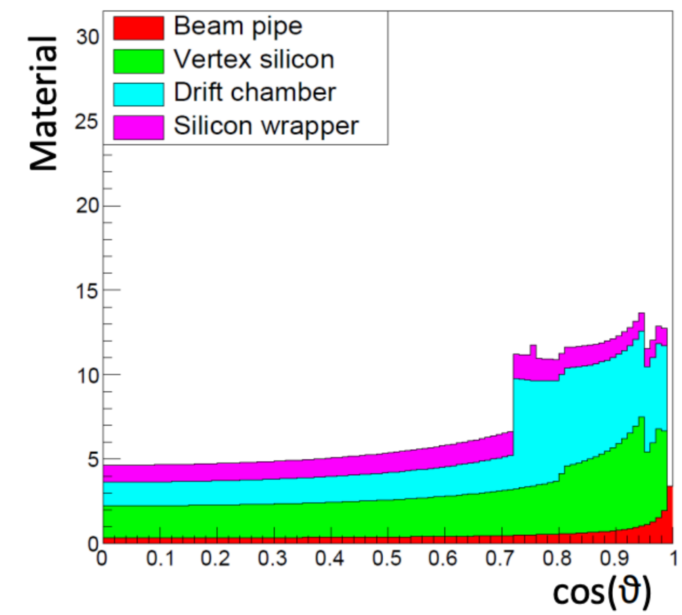
- **DualReadoutCalorimeter** module in Delphes assumes we can always disentangle these two cases
  - Probably ok at FCC-ee since probability of overlap not so large (except for taus?)
  - Impact of granularity on performance was studied extensively by Elisa Fontanesi (see here)

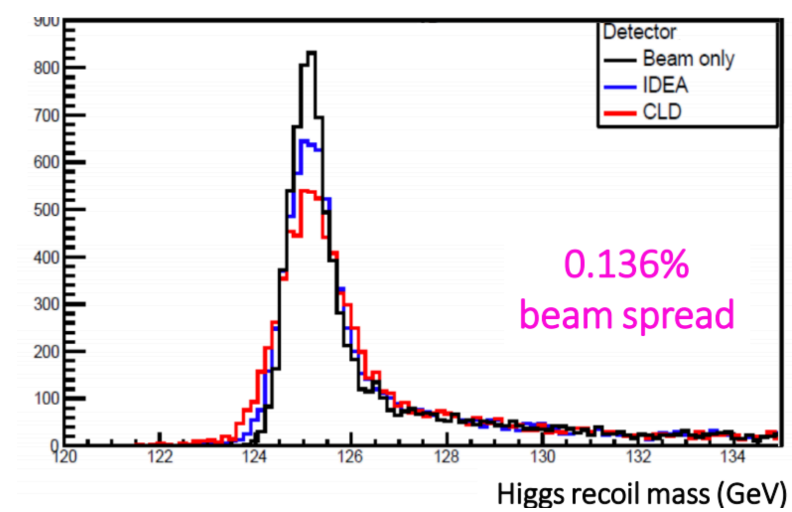


Fast Tracking Simulation

DELPHES
fast simulation







<u>Track Smearing</u>

- Simple tracker geometry implementation, **including material**

- Computes **full covariance matrix** (in present Delphes we have “diagonal” smearing in the 5 tracking parameters)

- Can be used for studying impact of material and realistic **HF tagging** simulation (need a **secondary vertexing** algorithm)

F. Bedeschi

TrackCovariance module

```
void TrackCovariance::Init()
{
  fBz = GetDouble("Bz", 0.0);
  fGeometry->Read(GetString("DetectorGeometry", ""));

  fCovariance->Calc(fGeometry);

  // import input array

  fInputArray = ImportArray(GetString("InputArray", "TrackMerger/tracks"));
  fItInputArray = fInputArray->MakeIterator();

  // create output array

  fOutputArray = ExportArray(GetString("OutputArray", "tracks"));
}
```

```
########################################
# Smearing for charged tracks
########################################

module TrackCovariance TrackSmearing {
    set InputArray TrackMergerPre/tracks
    set OutputArray tracks

    ## uses https://raw.githubusercontent.com/selvaggi/FastTrackCovariance/master/GeoIDEA_BASE.txt
    set DetectorGeometry {

      1 PIPE -100 100 0.015 0.0012 0.35276 0 0 0 0 0 0
      1 VTXLOW -0.12 0.12 0.017 0.00028 0.0937 2 0 1.5708 3e-006 3e-006 1
      1 VTXLOW -0.16 0.16 0.023 0.00028 0.0937 2 0 1.5708 3e-006 3e-006 1
      1 VTXLOW -0.16 0.16 0.031 0.00028 0.0937 2 0 1.5708 3e-006 3e-006 1
      1 VTXHIGH -1 1 0.32 0.00047 0.0937 2 0 1.5708 7e-006 7e-006 1
      1 VTXHIGH -1.05 1.05 0.34 0.00047 0.0937 2 0 1.5708 7e-006 7e-006 1
      1 DCHCANI -2.125 2.125 0.345 0.0002 0.237223 0 0 0 0 0 0
      1 DCH -2 2 0.36 0.0147748 1400 1 0.0203738 0 0.0001 0 1
      1 DCH -2 2 0.374775 0.0147748 1400 1 -0.0212097 0 0.0001 0 1
      1 DCH -2 2 0.38955 0.0147748 1400 1 0.0220456 0 0.0001 0 1
      1 DCH -2 2 0.404324 0.0147748 1400 1 -0.0228814 0 0.0001 0 1
      1 DCH -2 2 0.419099 0.0147748 1400 1 0.0237172 0 0.0001 0 1
      1 DCH -2 2 0.433874 0.0147748 1400 1 -0.024553 0 0.0001 0 1
      1 DCH -2 2 0.448649 0.0147748 1400 1 0.0253888 0 0.0001 0 1
```

TrackCovariance module

- **Requires:**
  - Geometry input
  - Magnetic field

New card ready for testing:

delphes_card_IDEAtrkCov.tcl

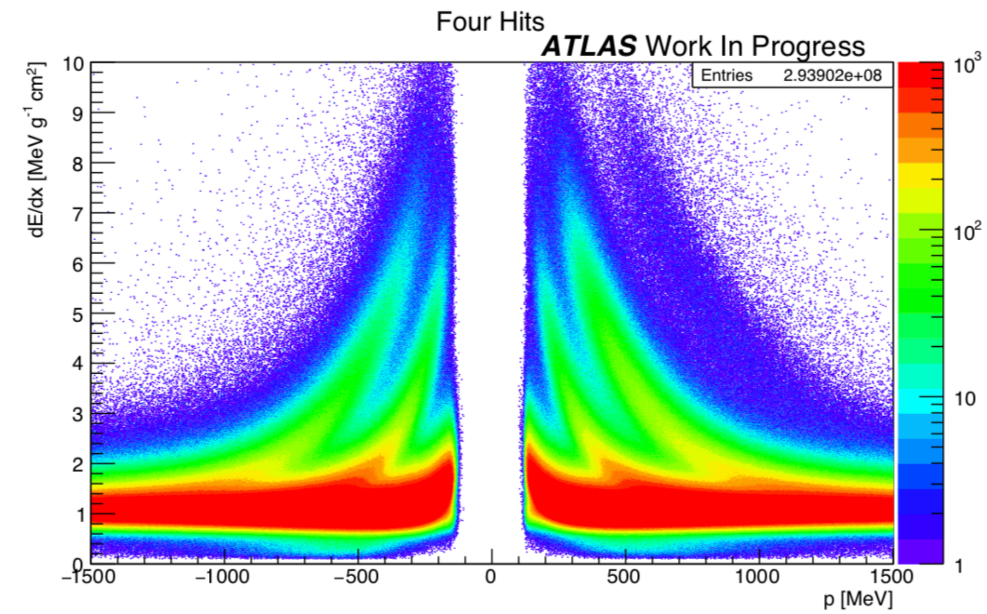
dEdx method (beta)

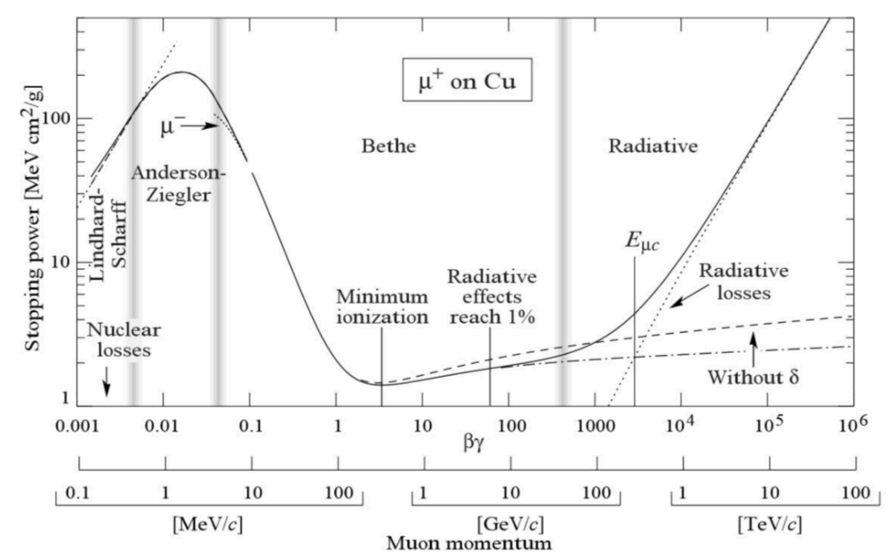
DELPHES
fast simulation

Algorithm:

For each track:

- Compute **path length L** (already done in Delphes):
  - **material properties**: e.g Silicon
  - **sensor thickness**: 100 um of Silicon
  - **fraction of active material** e.g. f = 1mm/1.5m
  - **resolution of charge measurement** in sensor: r in MeV/cm
  - **number of measurements** computed by the module as Nhits = f*L/thickness
  - specify fraction of measurements to be thrown away (truncated mean)
  - assume all charge collected
- Given material composition, particle velocity, etc… compute **ΔE_i from Landau distribution**
- Produce additional smearing with Gaussian resolution R to simulate finite resolution in charge collection
- Compute truncated mean



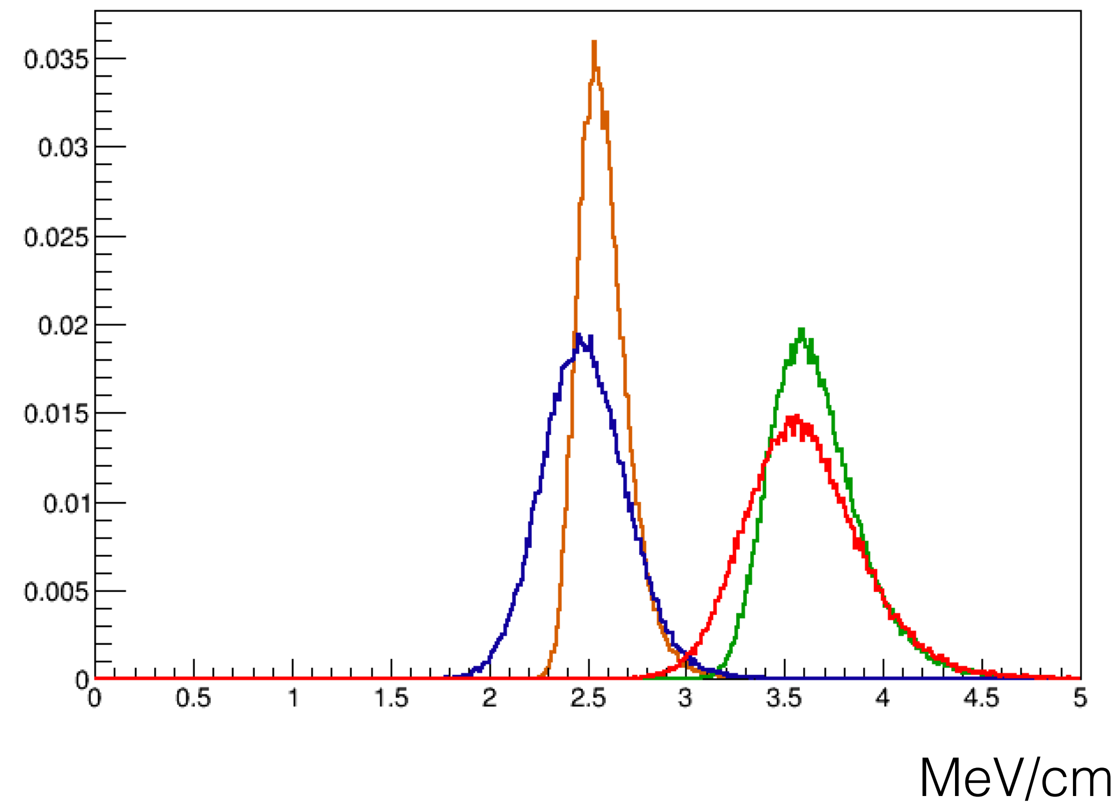


Degrees of freedom:

- **TruncatedMeanFraction**
- **Resolution**

dEdX in Silicon Tracker

```
module EnergyLoss EnergyLoss {
  add InputArray ChargedHadronMomentumSmearing/chargedHadrons
  add InputArray ElectronMomentumSmearing/electrons
  add InputArray MuonMomentumSmearing/muons


  # absolute resolution per measurement (normalized in MeV/cm)
  # CMS pixel detector performance is reproduceable with r = 0.4
  # dedicated dEdX detector can achieve r = 0.0 or below (i.e better than Landau)

  #set Resolution 0.4
  set Resolution 0.2

  # fraction of measurements to ignore when computing truncated mean
  # suggested range [0.4-0.6]

  set TruncatedMeanFraction 0.5

  # detector properties (active fraction = nhits*thickness/L)
  set Thickness 100E-6
  set ActiveFraction 0.0006666

  # Silicon properties, for other materials:
  # cf. http://pdg.lbl.gov/2014/AtomicNuclearProperties/properties8.dat

  set Z 14.
  set A 28.0855
  set rho 2.329

  # material polarisation correction parameters
  set a 0.1492
  set m 3.2546
  set x0 0.2015
  set x1 2.8716
  set I 173.0
  set c0 4.4355
```



10 measurements x100 um silicon

beta = 1 (Landau)

beta = 0.75 (Landau)

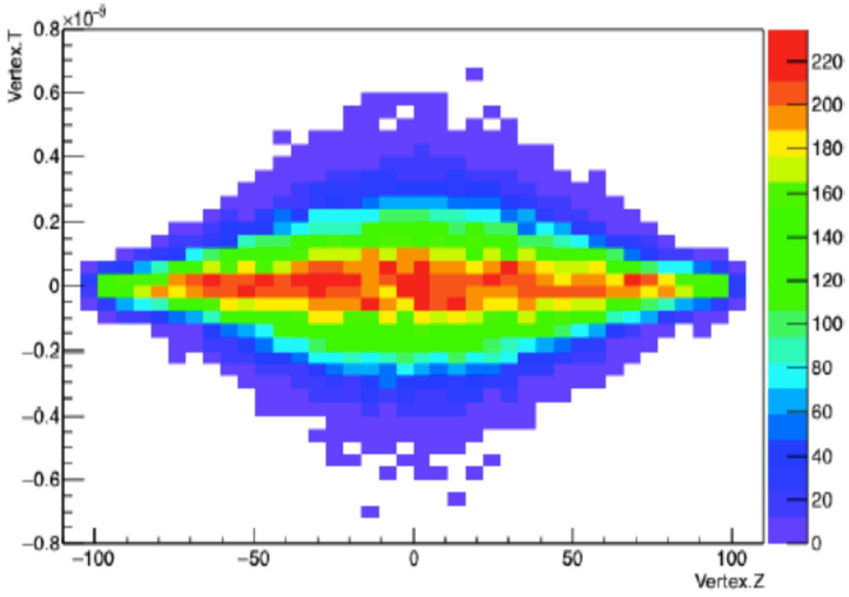
beta = 1 (LanGauss)

beta =0.75 (LanGauss)

Pile-up Simulation and Timing

- Pile-up can be mixed with hard event, with f(z,t) profile

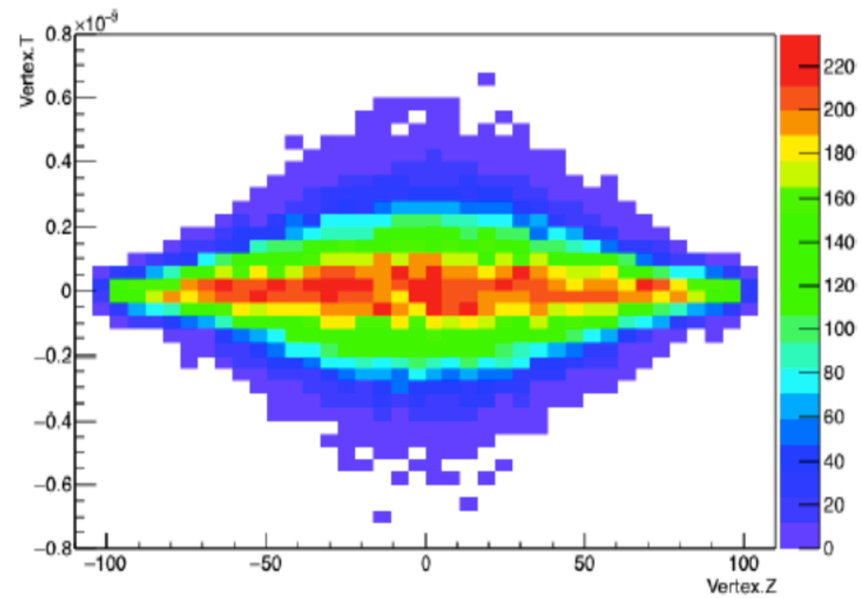
- Time of flight automatically computed in Delphes and propagated in the tracking volume, timing measurement simulated with TimeSmearing module

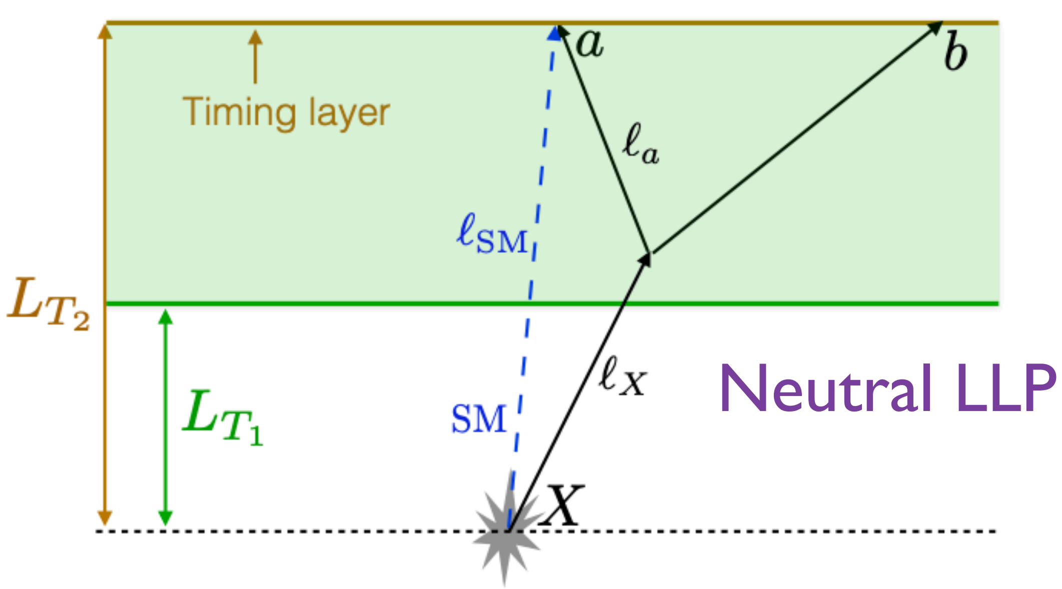


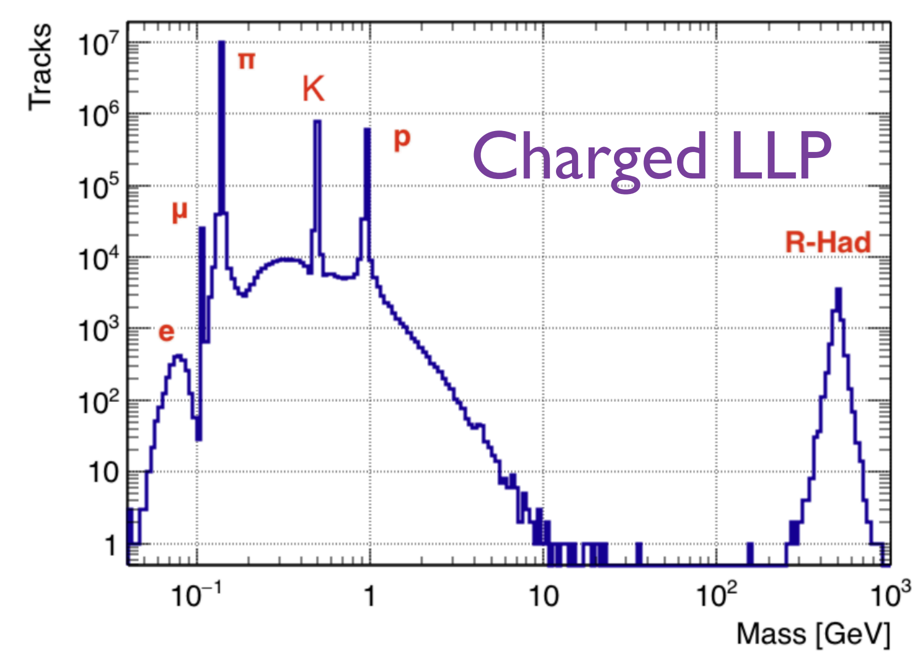

```
########################################
#   Time Smearing Neutral Photons
########################################

module TimeSmearing TimeSmearingPhotons {
  set InputArray ECal/eflowPhotons
  set OutputArray photons
  set TimeResolution {
                        (abs(eta) > 0.0 && abs(eta) <= 6.0) * sqrt(20e-12^2 + 150e-12^2)/energy^2
                     }
}
```

Timing detectors

- At the LHC, timing information can be used to disentangle hard vertex from pile-up, by vertexing in 4D
  - can this be used profitably in any way?
- Timing can be used to measure TOF, and hence for particle ID (either SM or BSM long lived particles)





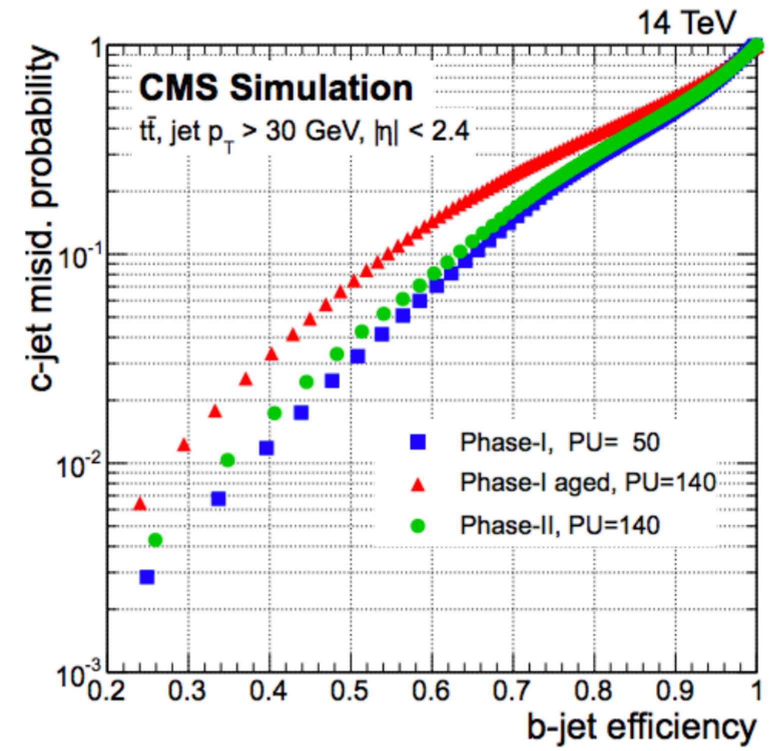


Heavy flavour Flavour Tagging

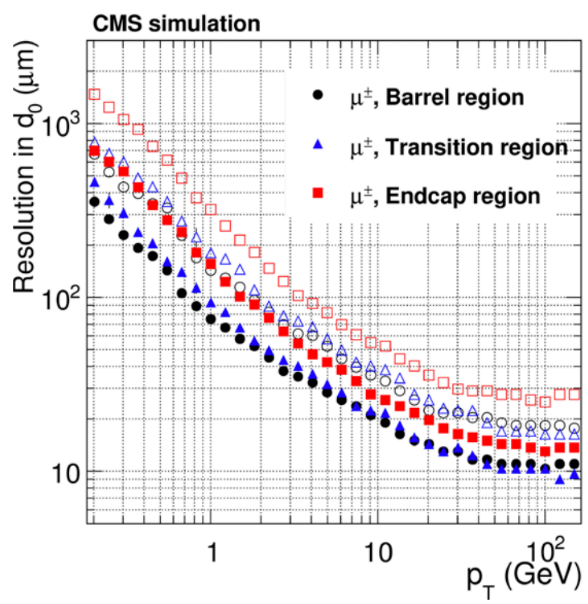
DELPHES fast simulation

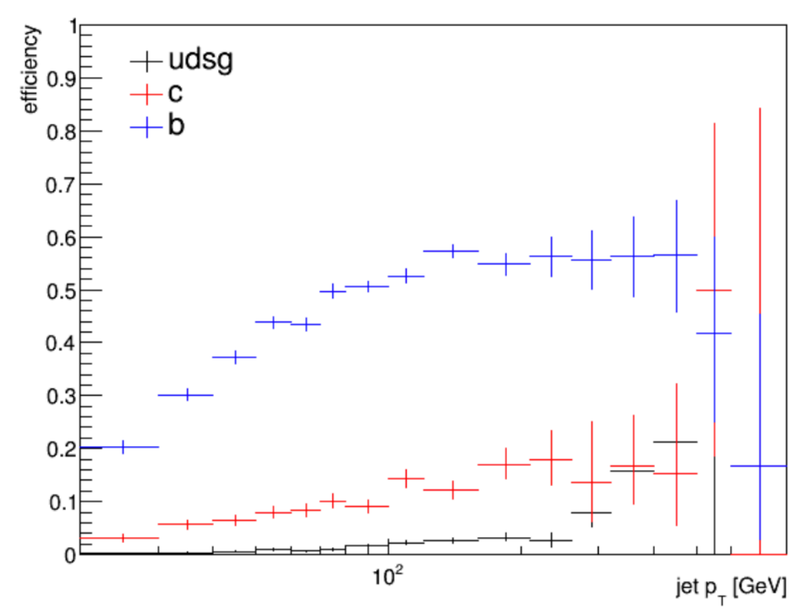
- Track Counting B-Tagging:
  - parameterise **longitudinal** and **transverse impact parameter resolution (see previous slide)**
  - count number of tracks with significant displacement
  - no **secondary vertexing** is performed yet

Can be used in conjunction with TrkCovariance module



Kevin Pedro





Practicalities

Run

- Install ROOT from root.cern.ch
- Clone Delphes from github.com/delphes
- Run Delphes:

```
>    ./configure
>    make
>    ./DelphesHepMC [detector_card] [output] [input(s)]
```

- Input formats: STDHEP, HepMC, ProMC, Pythia8
- Output: ROOT Tree

Configuration file

- Delphes configuration file is based on tcl scripting language
- This is where the detector parameters, the data-flow and the output content delphes root tree content are defined.
- Delphes provides tuned configurations for most existing detectors:
  - ATLAS, CMS, ILD, FCC, CEPC …

The order of execution of the various modules is configured in the execution path (usually defined at the beginning of the card):

```
set ExecutionPath {
    ParticlePropagator
    TrackEfficiency
    ...
    Calorimeter
    ...
    TreeWriter
}
```

Configuration file

```
module FastJetFinder FastJetFinder {

  set InputArray EFlowMerger/eflow
  set OutputArray jets

  # algorithm: 1 CDFJetClu, 2 MidPoint, 3 SIScone, 4 kt, 5 Cambridge/Aachen, 6 antikt
  set JetAlgorithm 5
  set ParameterR 0.8

  set ComputeNsubjettiness 1
  set Beta 1.0
  set AxisMode 4

  set ComputeTrimming 1
  set RTrim 0.2
  set PtFracTrim 0.05

  set ComputePruning 1
  set ZcutPrun 0.1
  set RcutPrun 0.5
  set RPrun 0.8

  set ComputeSoftDrop 1
  set BetaSoftDrop 0.0
  set SymmetryCutSoftDrop 0.1
  set R0SoftDrop 0.8

  set JetPTMin 20.0


}
```

Configuration file

```
module Calorimeter Calorimeter {

  set ParticleInputArray ParticlePropagator/stableParticles
  set TrackInputArray TrackMerger/tracks

  set TowerOutputArray towers
  set PhotonOutputArray photons

  set EFlowTrackOutputArray eflowTracks
  set EFlowPhotonOutputArray eflowPhotons
  set EFlowNeutralHadronOutputArray eflowNeutralHadrons

  ...

  # 10 degrees towers
  set PhiBins {}
  for {set i -18} {$i <= 18} {incr i} {
    add PhiBins [expr {$i * $pi/18.0}]
  }
  foreach eta {-3.2 -2.5 -2.4 -2.3 -2.2 -2.1 -2 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -1 -0.9 -0.8
-0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8
1.9 2 2.1 2.2 2.3 2.4 2.5 2.6 3.3} {
    add EtaPhiBins $eta $PhiBins
  }

  ...
  set ECalResolutionFormula {
            (abs(eta) <= 1.5) * (1+0.64*eta^2) * sqrt(energy^2*0.008^2 + energy*0.11^2 + 0.40^2) +
            (abs(eta) > 1.5 && abs(eta) <= 2.5) * (2.16 + 5.6*(abs(eta)-2)^2) * sqrt(energy^2*0.008^2 +
energy*0.11^2 + 0.40^2) +
            (abs(eta) > 2.5 && abs(eta) <= 5.0) * sqrt(energy^2*0.107^2 + energy*2.08^2)}
```

input(s) candidates

output(s) candidates

13

Configuration file

Output collections are configured in the TreeWriter module

```
module TreeWriter TreeWriter {
# add Branch InputArray BranchName BranchClass
  add Branch Delphes/allParticles Particle GenParticle

  add Branch TrackMerger/tracks Track Track
  add Branch Calorimeter/towers Tower Tower

  add Branch Calorimeter/eflowTracks EFlowTrack Track
  add Branch Calorimeter/eflowPhotons EFlowPhoton Tower
  add Branch Calorimeter/eflowNeutralHadrons EFlowNeutralHadron Tower

  add Branch GenJetFinder/jets GenJet Jet
  add Branch GenMissingET/momentum GenMissingET MissingET

  add Branch UniqueObjectFinder/jets Jet Jet
  add Branch UniqueObjectFinder/electrons Electron Electron
  add Branch UniqueObjectFinder/photons Photon Photon
  add Branch UniqueObjectFinder/muons Muon Muon
  add Branch MissingET/momentum MissingET MissingET
  add Branch ScalarHT/energy ScalarHT ScalarHT
}
```