# DELPHES
## fast simulation


Michele Selvaggi, for the Delphes Team

*Université Catholique de Louvain (UCL)*
*Center for Particle Physics and Phenomenology (CP3)*


**2nd Fast Monte Carlo Workshop in HEP**
**15 January 2014**

# Detector simulation

- Full simulation (GEANT):

  - **simulates** particle-matter interaction (including e.m. showering, nuclear int., brehmstrahlung, photon conversions, etc ...) $\rightarrow$ 10-100 s /ev

- Experiment Fast simulation (ATLAS, CMS ...):

  - **simplifies** and makes faster simulation and reconstruction
  - mixes G4, parametric, libraries $\rightarrow$ 1 s /ev

- Parametric simulation:

  **Delphes**, PGS**:**

  - **parameterize** detector response, reconstruct complex objects $\rightarrow$ 10 ms /ev

  TurboSim

  - **no detector**, 4-momentum smearing, look-up table (parton $\leftrightarrow$ reco)

# DELPHES

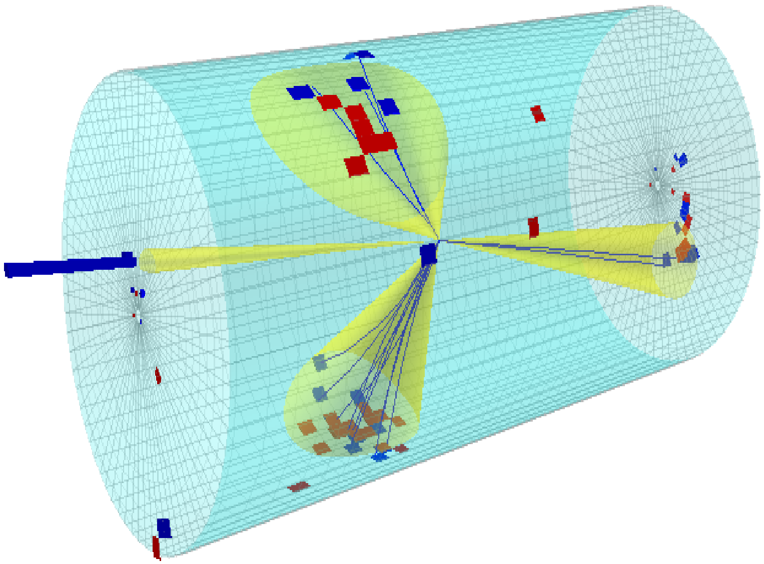
# Development

- Delphes project started back in 2007 at UCL
- Since 2009, its development is **community-based**
  - **ticketing system** for improvement and bug-fixes
    → user proposed patches
  - **Quality control** and **core development** is done at the UCL
- In 2013, **DELPHES 3** was released:
  - modular software
  - new features
  - included in MG/ME suite
- **Widely** tested and used by the community (mainly pheno)
- Website and manual: https://cp3.irmp.ucl.ac.be/projects/delphes
- Paper: arXiv:1307.6346

# DELPHES in a nutshell

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

- **Simulates**:
  - pile-up
  - charged particle propagation in magnetic field: **tracking**
  - electromagnetic and hadronic **calorimeters**
  - **muon** system



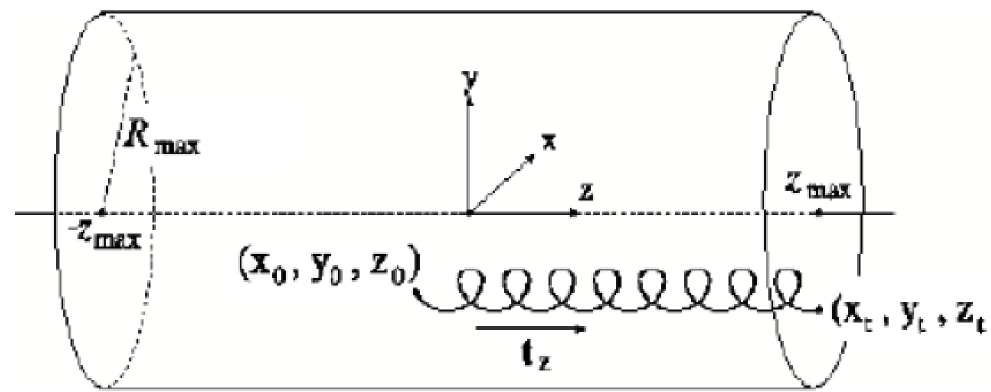
- **Reconstructs**:
  - leptons (electrons and muons)
  - photons
  - jets and missing transverse energy (particle-flow)
  - taus and b's



Remark : Hadron collider specific, but easily extendible to $e^+ e^-$

# *Particle Propagation*

- **Charged** and **neutral** particles are propagated in the magnetic field until they reach the calorimeters




- Propagation parameters:
  - magnetic field **B**
  - **radius** and **half-length** ($R_{max}$, $z_{max}$)

- Efficiency/resolution depends on:
  - particle ID
  - transverse momentum
  - pseudorapidity

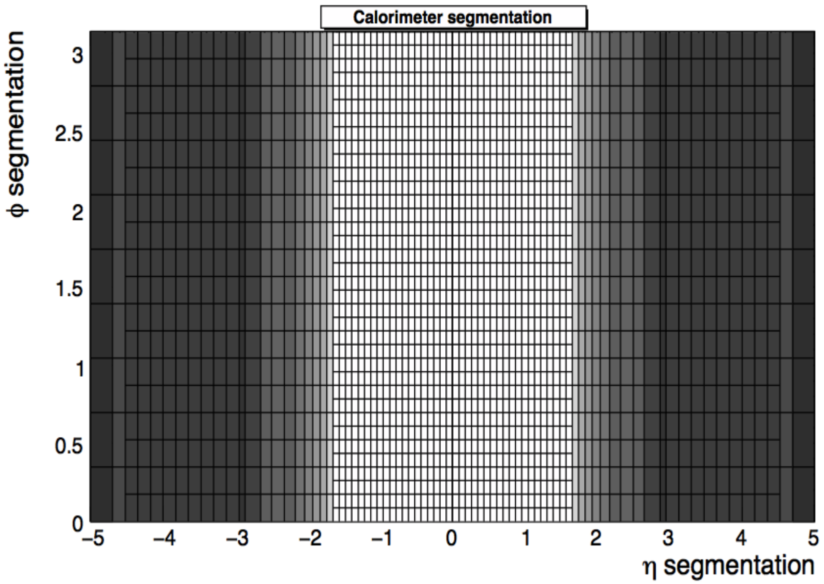
```
# efficiency formula for muons
add EfficiencyFormula {13} {                                          (pt <= 0.1)   * (0.000) + \
                                            (abs(eta) <= 1.5) * (pt > 0.1   && pt <= 1.0)   * (0.750) + \
                                            (abs(eta) <= 1.5) * (pt > 1.0)                  * (1.000) + \
                        (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 0.1   && pt <= 1.0)   * (0.700) + \
                        (abs(eta) > 1.5 && abs(eta) <= 2.5) * (pt > 1.0)                  * (0.975) + \
                        (abs(eta) > 2.5)                                                   * (0.000)}
```

Not real tracking/vertexing !!
→ no fake tracks/ conversions (but can be easily implemented)
→ no dE/dx measurements

# Calorimetry

- em/had calorimeters have same **segmentation** in eta/phi
- Each particle that reaches the calorimeters **deposits a fraction of its energy** in one ECAL cell ($f_{EM}$) and HCAL cell ($f_{HAD}$), depending on its type:

| particles | $f_{EM}$ | $f_{HAD}$ |
| --- | --- | --- |
| e γ $\pi^0$ | 1 | 0 |
| Long-lived neutral hadrons ($K^0_S$, $\Lambda^0$) | 0.3 | 0.7 |
| ν μ | 0 | 0 |
| others | 0 | 1 |




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

$$E_{Tower} = \sum_{particles} \ln\mathcal{N}\left(f_{ECAL} \cdot E, \sigma_{ECAL}(E,\eta)\right) + \ln\mathcal{N}\left(f_{HCAL} \cdot E, \sigma_{HCAL}(E,\eta)\right)$$

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

# *Particle-Flow reconstruction*

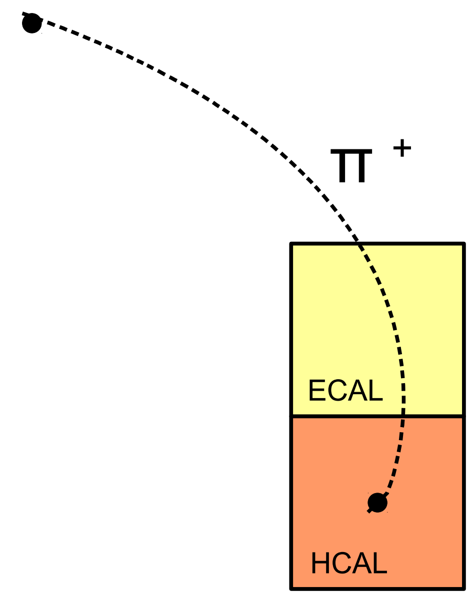
- Idea: optimally combine **all sub-detectors** information
- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)

→ assume **σ(trk) < σ(calo)**

# *Particle-Flow reconstruction*

- Idea: optimally combine **all sub-detectors** information
- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)

→ assume **σ(trk) < σ(calo)**



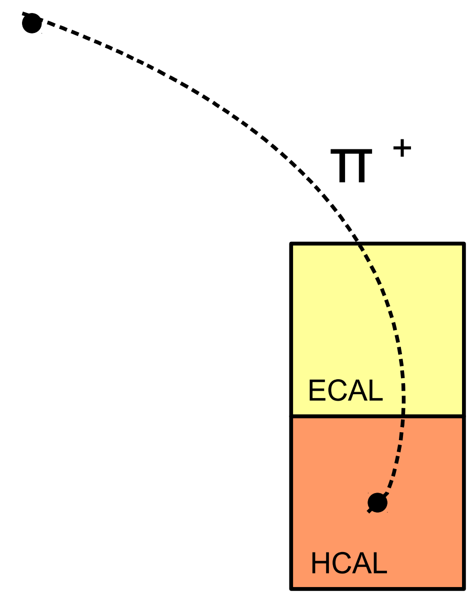

$E^{MC}(\pi^+) = 10$ GeV → $E^{HCAL}(\pi^+) = 9$ GeV
$E^{TRK}(\pi^+) = 11$ GeV

**Particle-Flow** algorithm creates:

→ PF-track, with energy $E^{PF\text{-}trk} = 11$ GeV

# *Particle-Flow reconstruction*

- Idea: optimally combine **all sub-detectors** information
- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)




→ assume **σ(trk) < σ(calo)**

$E^{MC}(\pi^+) = 10$ GeV → $E^{HCAL}(\pi^+) = 9$ GeV

$E^{TRK}(\pi^+) = 11$ GeV

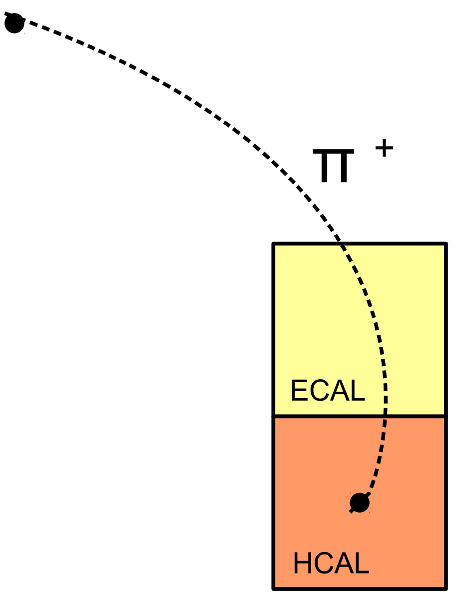
**Particle-Flow** algorithm creates:

→ PF-track, with energy $E^{PF\text{-}trk} = 11$ GeV

# *Particle-Flow reconstruction*

- Idea: optimally combine **all sub-detectors** information
- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)

→ assume **σ(trk) < σ(calo)**




$E^{MC}(\pi^+) = 10$ GeV → $E^{HCAL}(\pi^+) = 15$ GeV

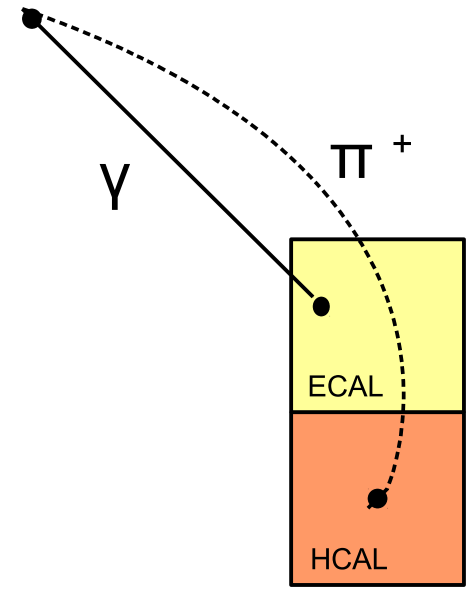
$E^{TRK}(\pi^+) = 11$ GeV

**Particle-Flow** algorithm creates:

→ PF-track, with energy $E^{PF\text{-}trk} = 11$ GeV

→ PF-tower, with energy $E^{PF\text{-}tower} = 4$ GeV

# Particle-Flow reconstruction




- Idea: optimally combine **all sub-detectors** information
- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)

→ assume **σ(trk) < σ(calo)**

$E^{MC}(\gamma) = 20$ GeV → $E^{ECAL}(\gamma) = 18$ GeV

$E^{MC}(\pi^+) = 10$ GeV → $E^{HCAL}(\pi^+) = 15$ GeV

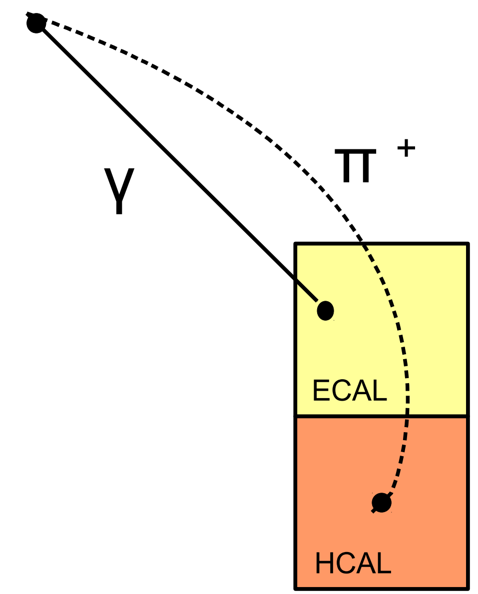
$E^{TRK}(\pi^+) = 11$ GeV

**Particle-Flow** algorithm creates:

→ PF-track, with energy $E^{PF\text{-}trk} = 11$ GeV

→ PF-tower, with energy $E^{PF\text{-}tower} = 4 + 18$ GeV

# *Particle-Flow reconstruction*




- Idea: optimally combine **all sub-detectors** information

- In practice, in DELPHES use **tracking and calo** info to reconstruct high reso. input objects for later use (jets, $E_T^{miss}$, $H_T$)

→ assume **σ(trk) < σ(calo)**

$E^{MC}(\gamma) = 20$ GeV → $E^{ECAL}(\gamma) = 18$ GeV
$E^{MC}(\pi^+) = 10$ GeV → $E^{HCAL}(\pi^+) = 15$ GeV
$E^{TRK}(\pi^+) = 11$ GeV

**Particle-Flow** algorithm creates:

→ PF-track, with energy $E^{PF\text{-}trk} = 11$ GeV
→ PF-tower, with energy $E^{PF\text{-}tower} = 4 + 18$ GeV

Separate neutral and charged calo deposits has crucial implications for pile-up subtraction

# Jets / $E_T^{miss}$ / $H_T$

- Delphes uses **FastJet** libraries for jet clustering
- Inputs can be formed from:
  - **calorimeter towers**
  - “**particle-flow**” tracks and towers

# Leptons, photons

- Muons/photons/electrons
  - **identified** via their PDG id
  - muons do not deposit energy in calo (independent smearing parameterized in $p_T$ and η)
  - electrons and photons smeared according to electromagnetic calorimeter resolution
- Isolation:

$$I(P) = \frac{\sum_{i \neq P}^{\Delta R < R,\ p_T(i) > p_T^{min}} p_T(i)}{p_T(P)}$$

→ modular structure allows to easily define different isolation

If **I(P) < Imin**, the lepton is **isolated**

User can specify parameters $I_{min}$, ΔR, $p_T^{min}$

- Not taken into account:
  - fakes, punch-through, brehmstrahlung, conversions

# b and tau jets

- b-jets

  - if **b** parton is found in a cone ΔR w.r.t jet direction
    → apply **efficiency**
  - if **c** parton is found in a cone ΔR w.r.t jet direction
    → apply **c-mistag rate**
  - if **u,d,s,g** parton is found in a cone ΔR w.r.t jet direction
    → apply **light-mistag rate**

b-tag **flag** is then stored in the jet collection

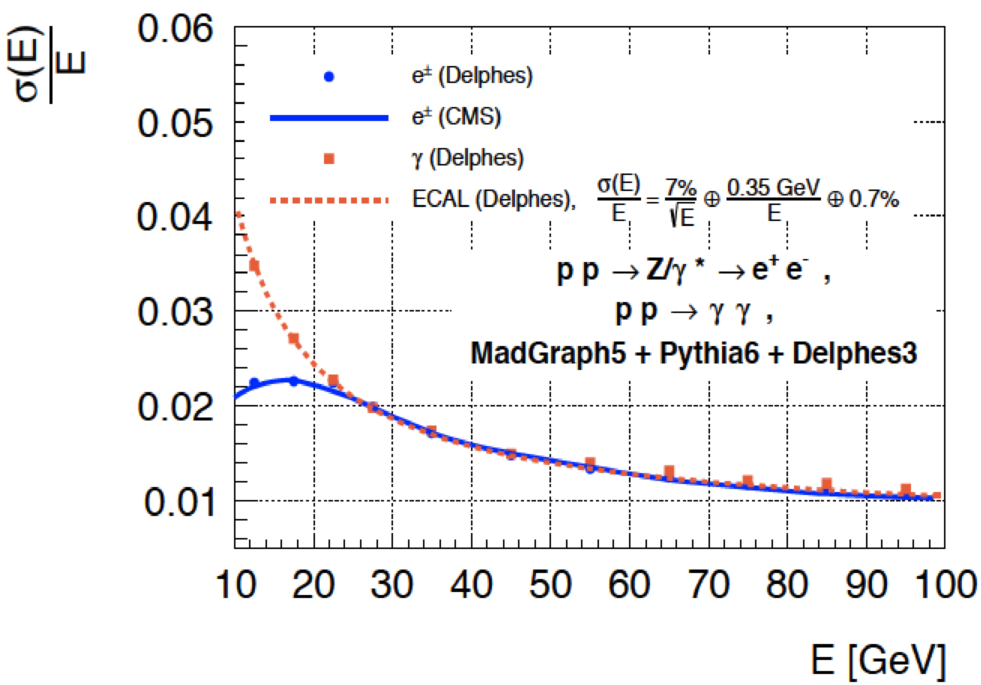
- tau-jets

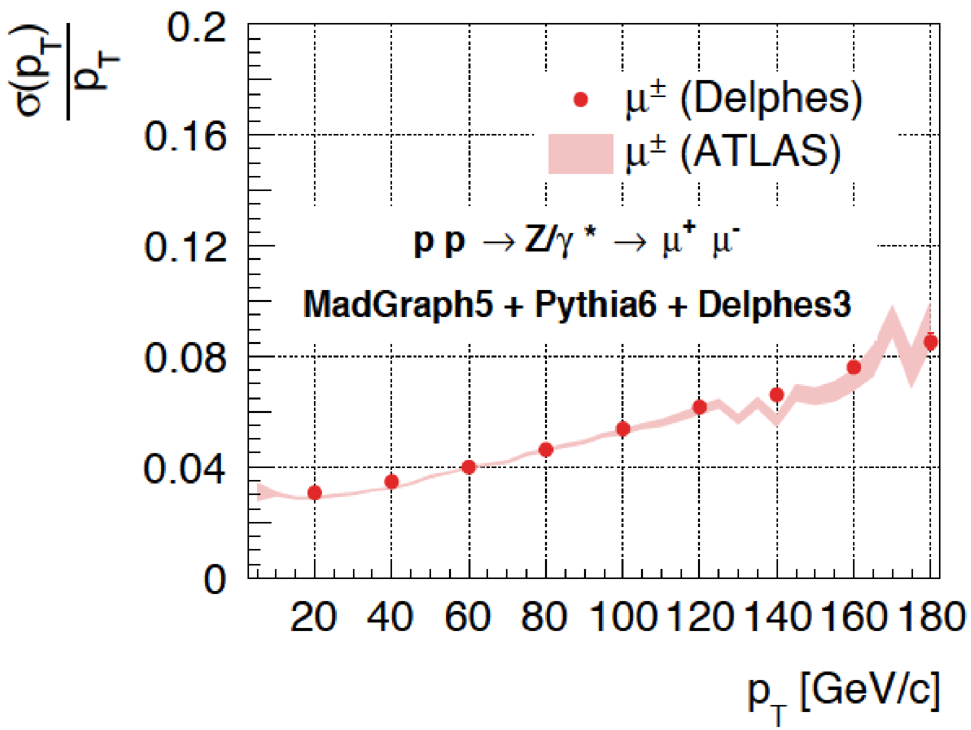
  - if tau lepton is found in a cone ΔR w.r.t jet direction
    → apply **efficiency**
  - else
    → apply **tau-mistag rate**

**$p_T$ and η dependent** efficiency and mistag rate

# VALIDATION

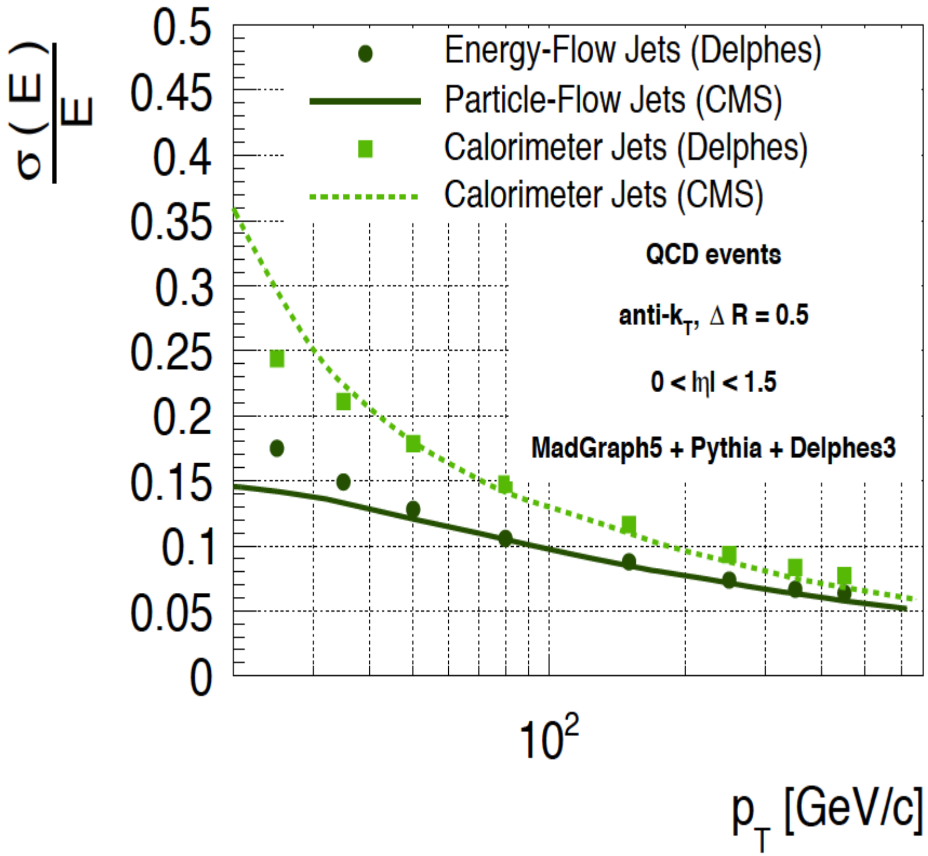
# Validation: electrons and muons

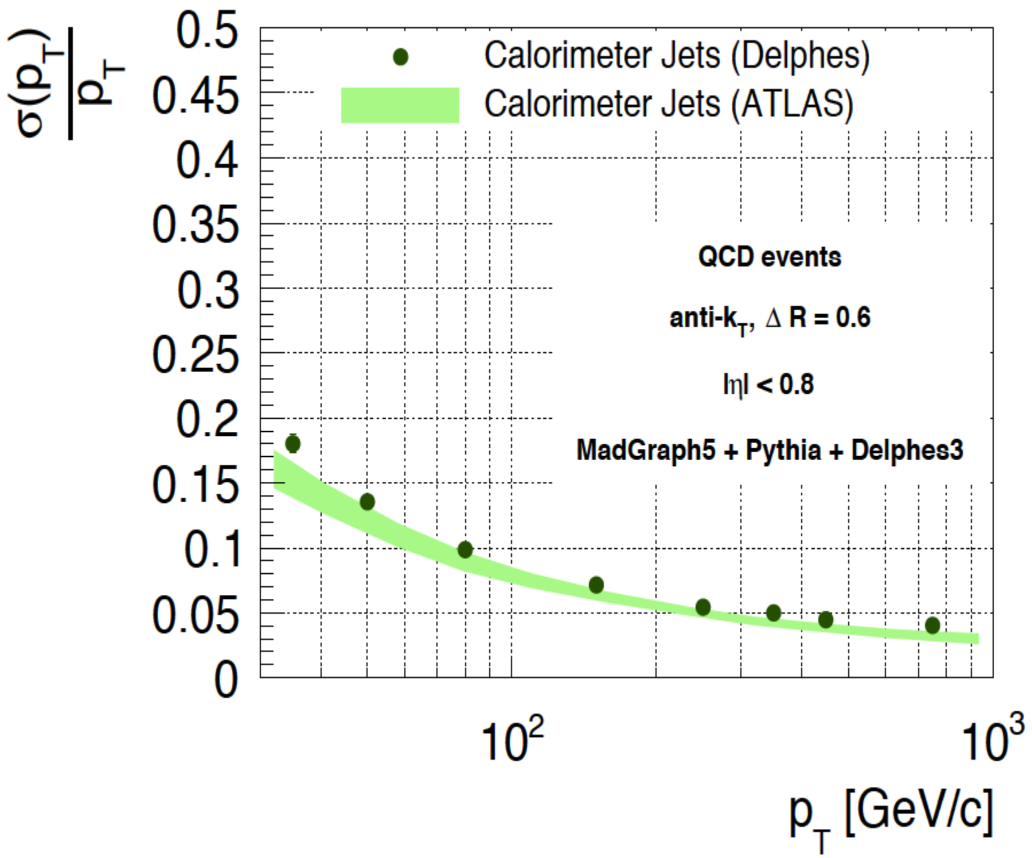



**→ excellent agreement**

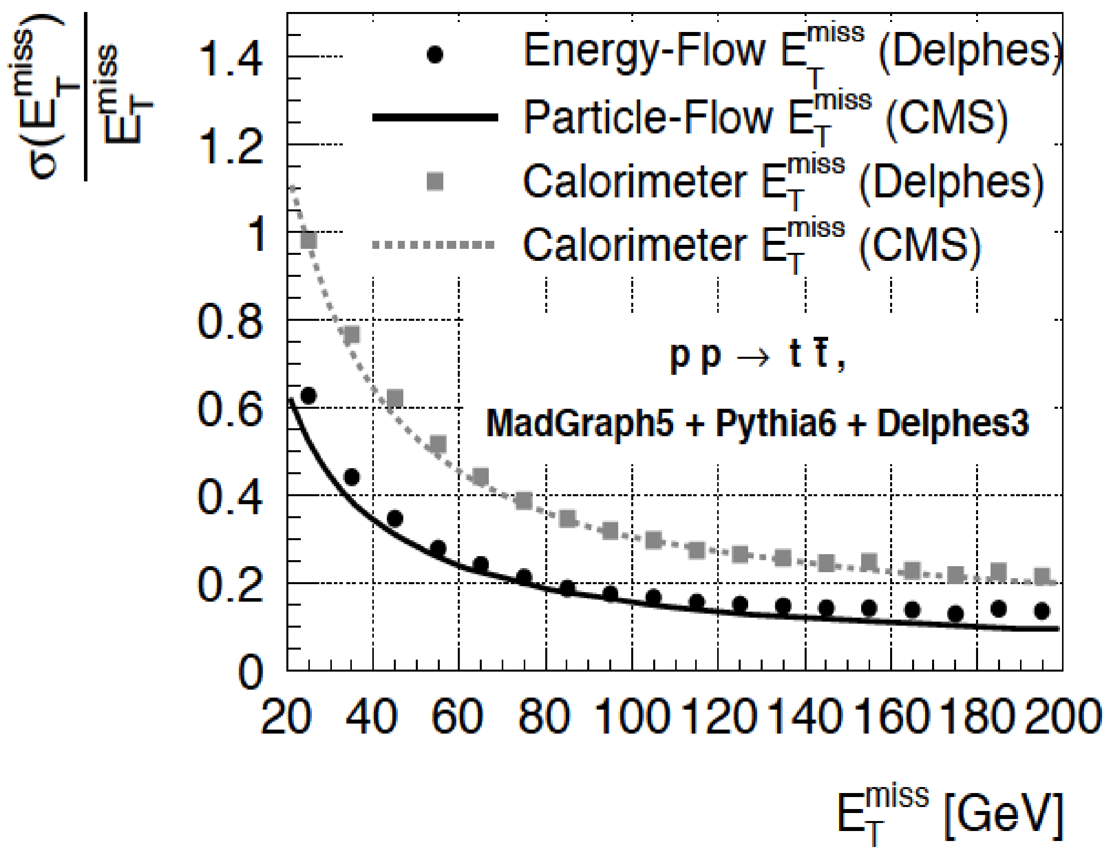
# Validation: jets




→ **good agreement**

# *Validation: $E_T^{miss}$*




→ **excellent agreement**

# Physics example

- Reproduce part of **top mass measurement** in semi-leptonic decay (arXiv:1209:2319)

- **Signal** produced with MG5+Pythia+Delphes3

- Selection criteria:

→ = 1 lepton $p_T$ > 30 GeV, $|\eta|$ < 2.1
→ ≥ 4 jets $p_T$ > 30 GeV, $|\eta|$ < 2.4
→ ≥ 2 b-tagged jets, ≥ 2 light jets

eff(Delphes) = 2.8% vs. eff(CMS) = 2.3%

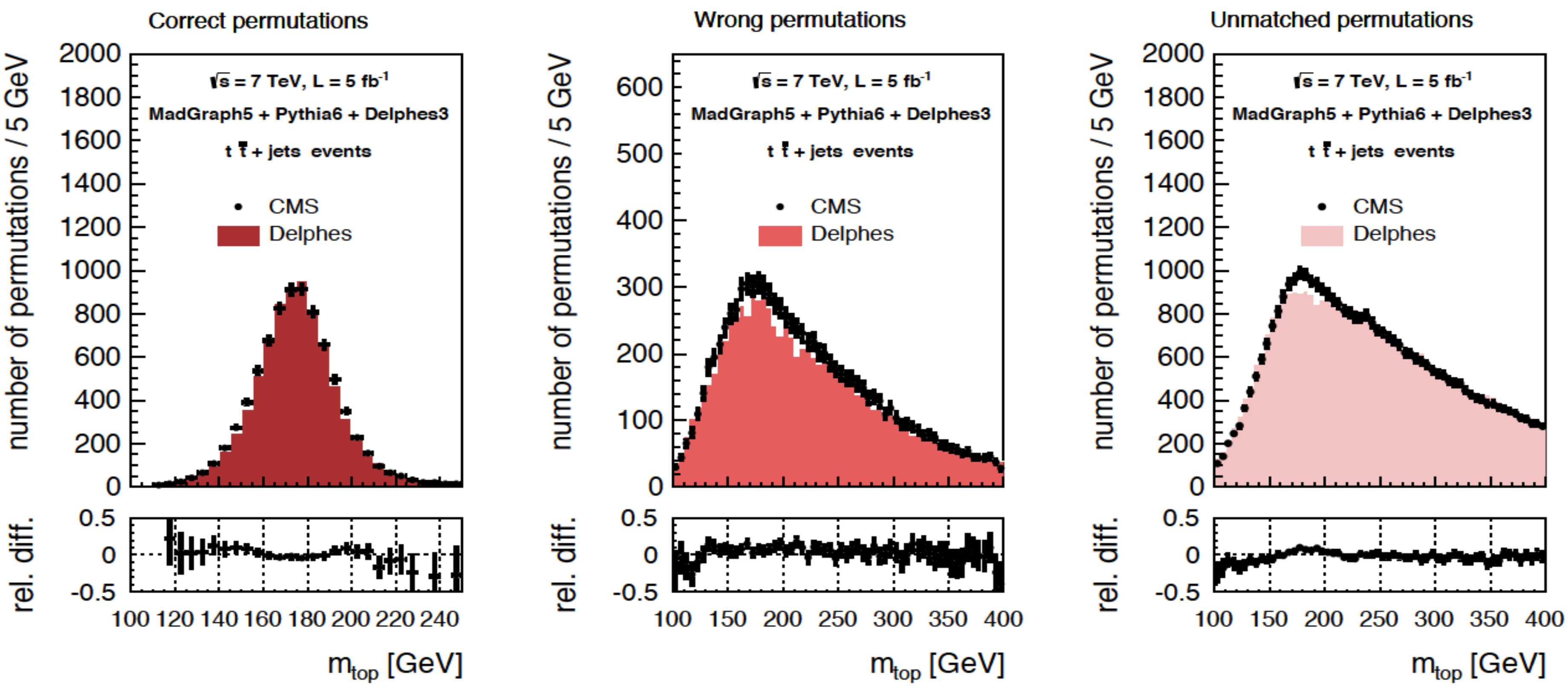
→ **good agreement**

# Physics example

Look at **hardest** 2 b-tagged and 2 light jets (à la CMS):

- correct : 4 jets are good, match right b with lights
- wrong : 4 jets are good, match wrong b with lights
- unmatched : at least one of the jets don't match

| | CMS | DELPHES |
|---|---|---|
| correct | 15.5 % | 15.8 % |
| wrong | 17.4 % | 16.5 % |
| unmatched | 67.1 % | 67.7 % |

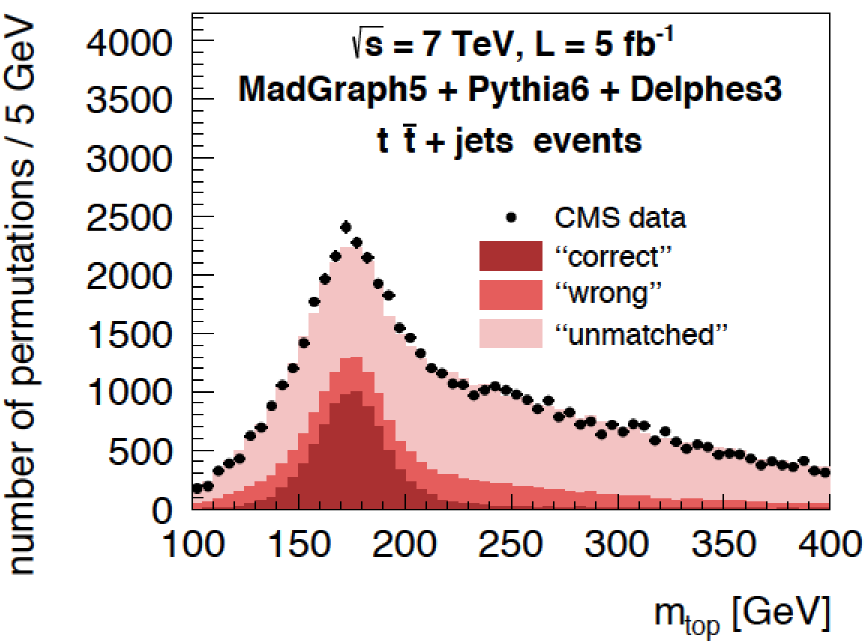


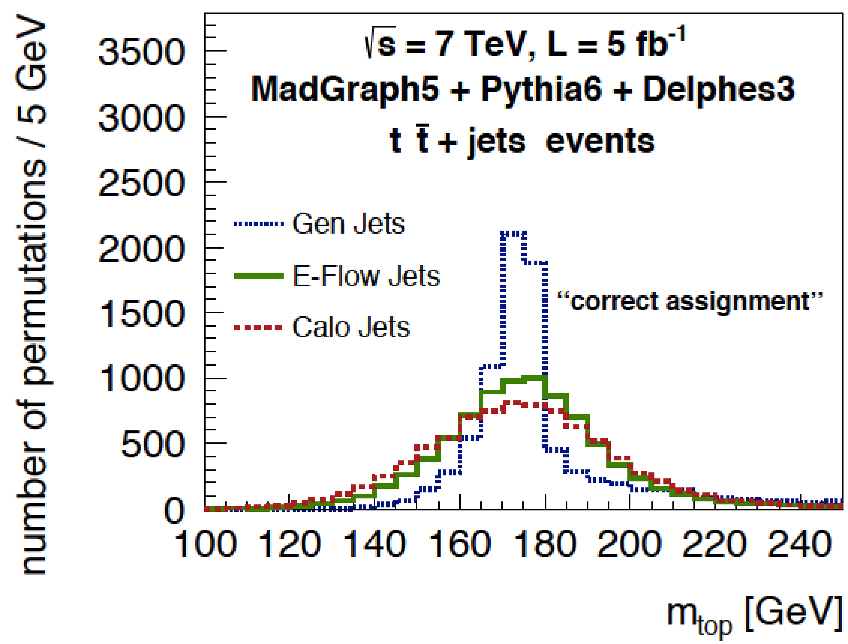
# Physics example

Look at **hardest** 2 b-tagged and 2 light jets (à la CMS):

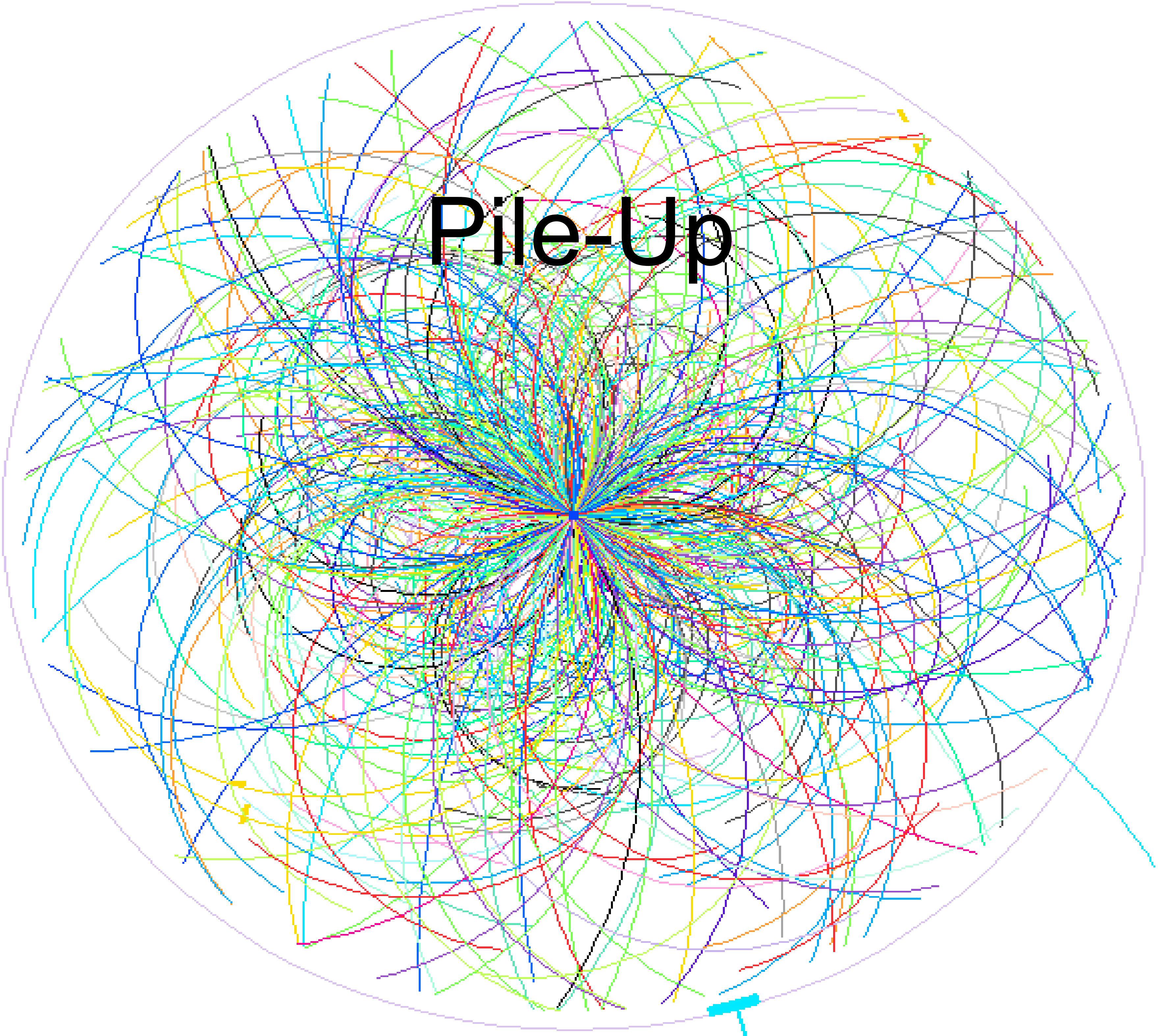
- correct : 4 jets are good, match right b with lights
- wrong : 4 jets are good, match wrong b with lights
- unmatched : at least one of the jets don't match

| | CMS | Delphes |
|---|---|---|
| correct | 15.5 % | 15.8 % |
| wrong | 17.4 % | 16.5 % |
| unmatched | 67.1 % | 67.7 % |



# Pile-Up



# *Pile-up motivations*

- **Pile-up** becomes an issue at **high luminosity** LHC
  - worsened **resolution** (jets, $E_T^{miss}$)
  - degraded **isolation**
  - **fake** tracks, jets
- **Efficiencies** and **resolutions** can be **tuned by hand** to mimic pile-up
- Fake objects (jets) need to be simulated. Also, we want to have some predictive power:

  → We therefore introduced: tunable **simulation** of pile-up
  pile-up **substraction** procedure.

# *Pile-up implementation*

- **Pile-up** is implemented in Delphes **since version 3.0.4**
  - **mixes** N minimum bias events with hard event sample
  - spreads **poisson(N)** events along z-axis with configurable spread
  - rotate event by random angle $\varphi$ wrt z-axis

- **Charged** Pile-up subtraction (most effective if used with PF jets)

  - if **$z < |Zres|$** keep all **charged and neutrals** (→ ch. particles too close to hard scattering to be rejected)

  - if **$z > |Zres|$** keep only **neutrals** (perfect charged subtraction)

  - allows user to tune amount of charged particle subtraction by **adjusting Z spread/resolution**

- **Residual** pile-up substraction is needed for jets and isolation.
  - Use the FastJet Area approach (Cacciari, Salam, Soyez)
    - compute $\rho$ = event pile-up density
    - jet correction : $pT \rightarrow pT - \rho A$ (JetPileUpSubtractor)
    - isolation : $\sum pT \rightarrow \sum pT - \rho \pi R^2$ (Isolation module itself)
  - Subtraction can be $|\eta|$ dependent

# Pile – Up Validation

# Validation: Pile-Up

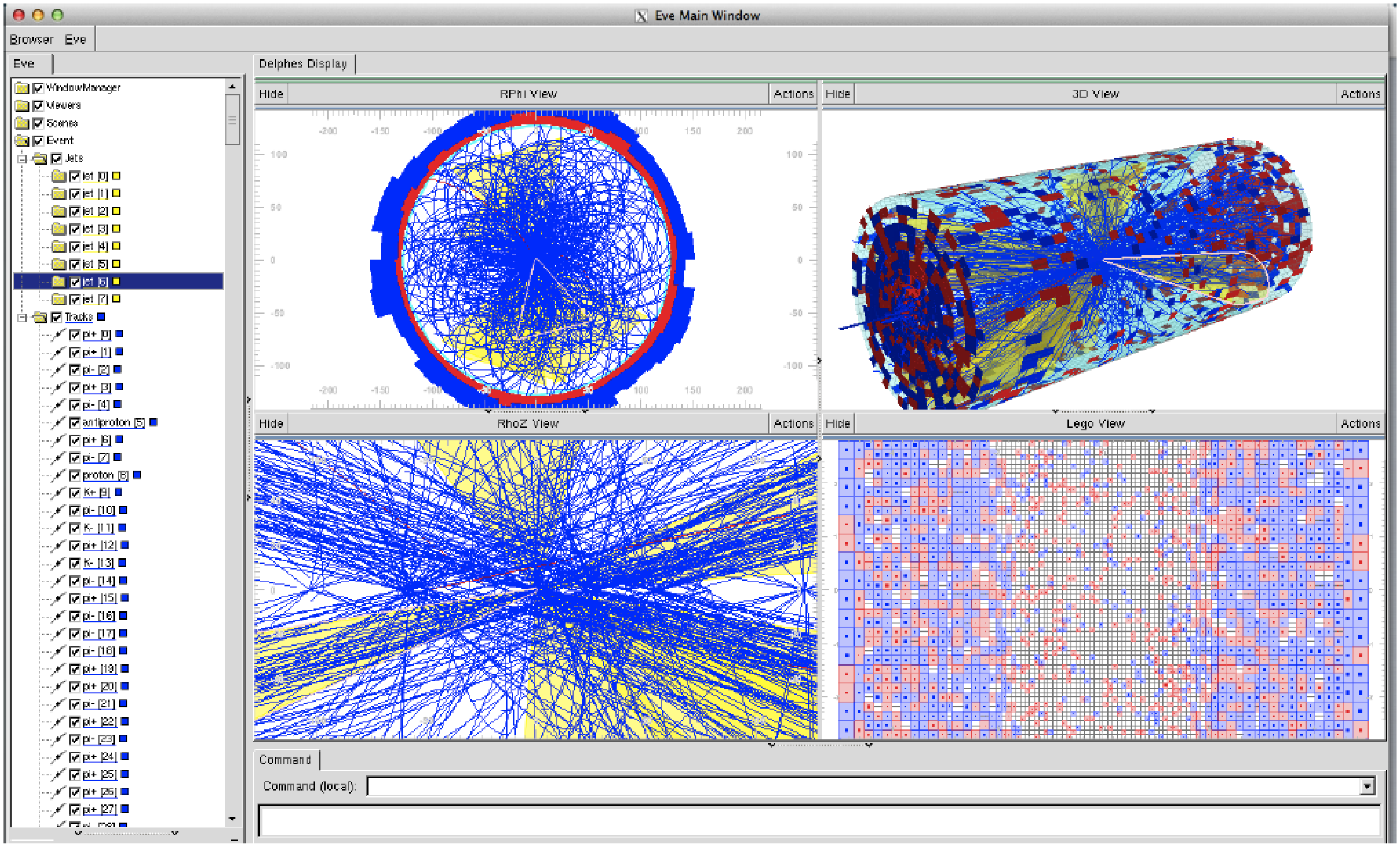
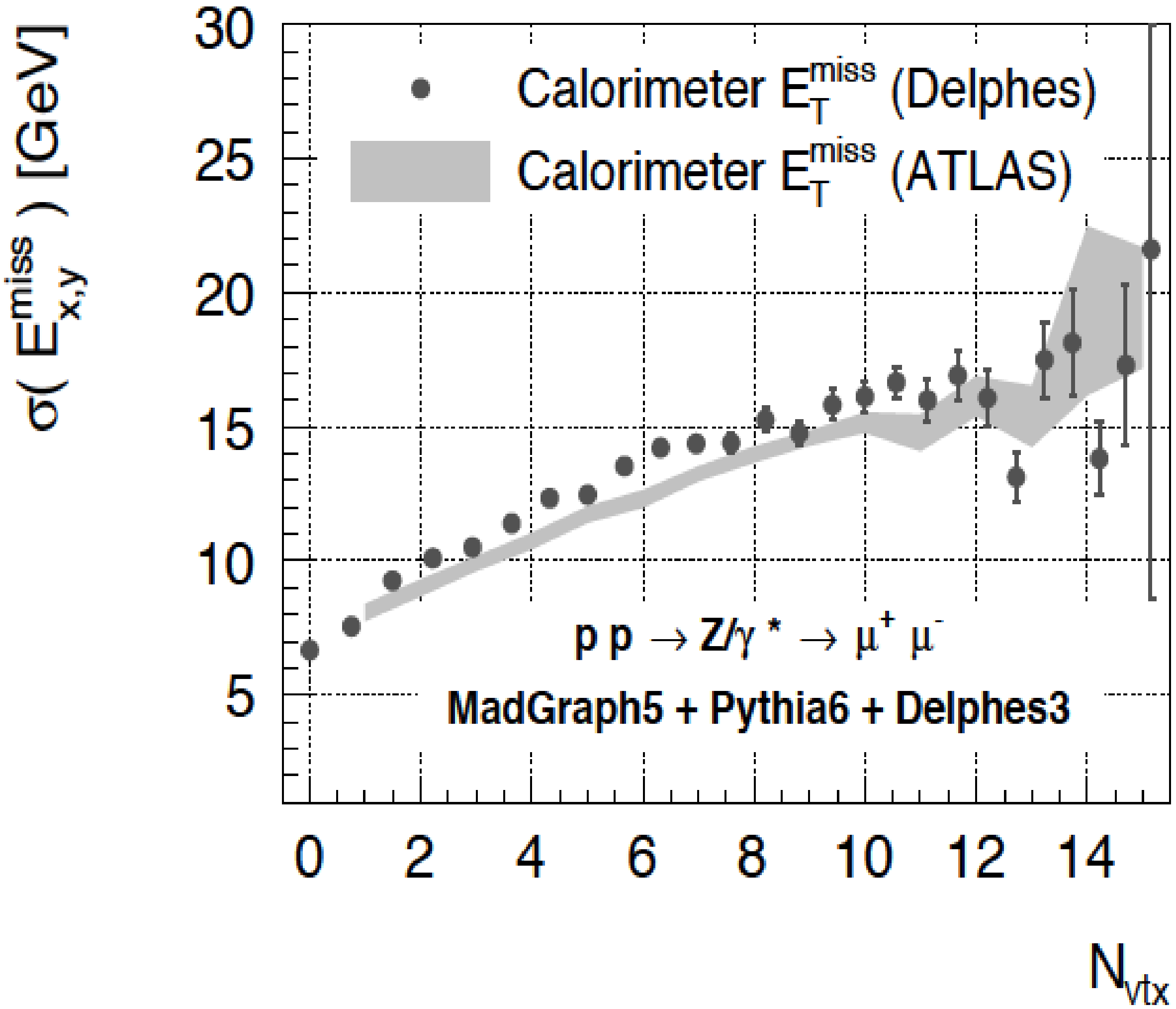



**Figure 3.** QCD event with 50 pile-up interactions shown with the DELPHES event display based on the ROOTEVE libraries [12]. Transverse view (top left), longitudinal view (bottom left), 3D view (top right), $(\eta,\phi)$ view (bottom right).

# Validation: Pile-Up



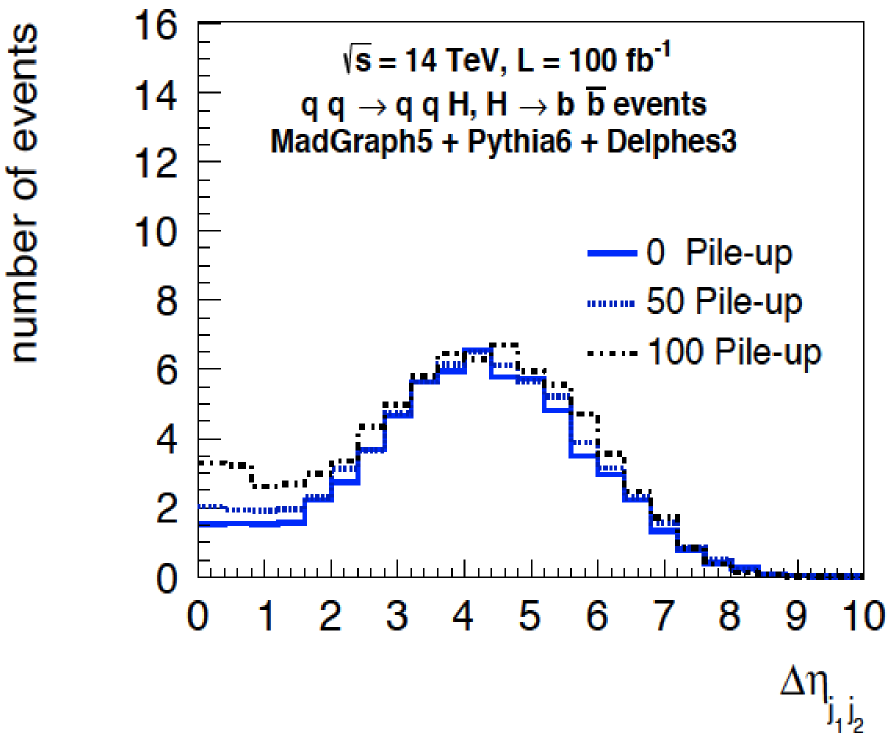

→ **good agreement**

# Validation: Pile-Up

- H → bb in **VBF channel** expected to be highly affected by pile-up
- Irreducible background **bb+jets**
- Select >4 jets with pT > 80, 60, 40, 40 (at least 2 b-tagged, at least 2 light)

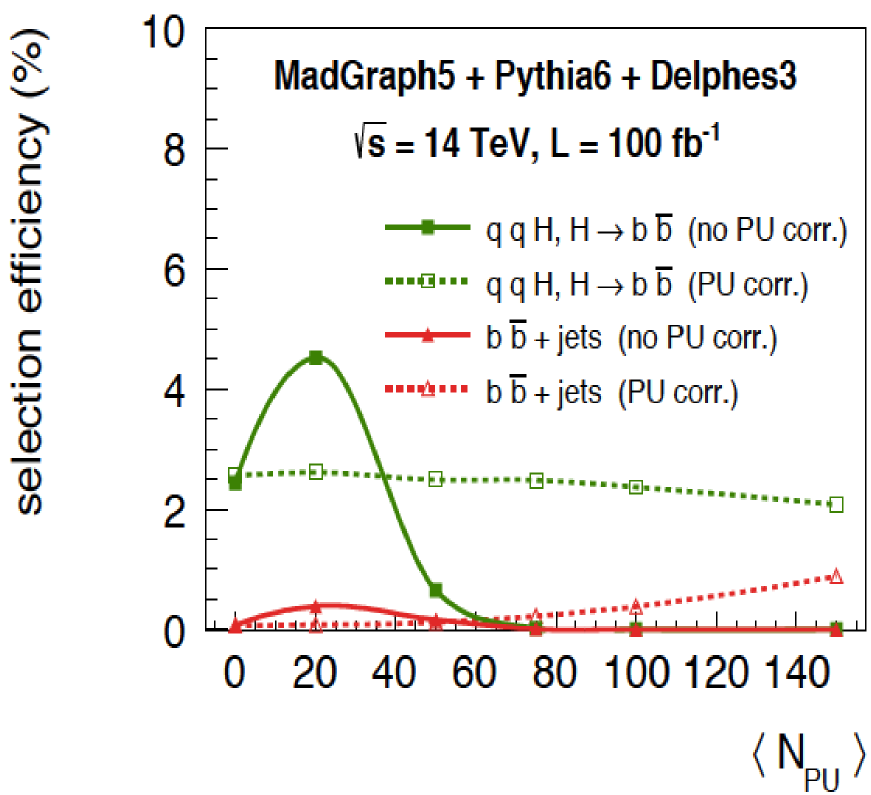
Emergence of pile-up jets in the central region:

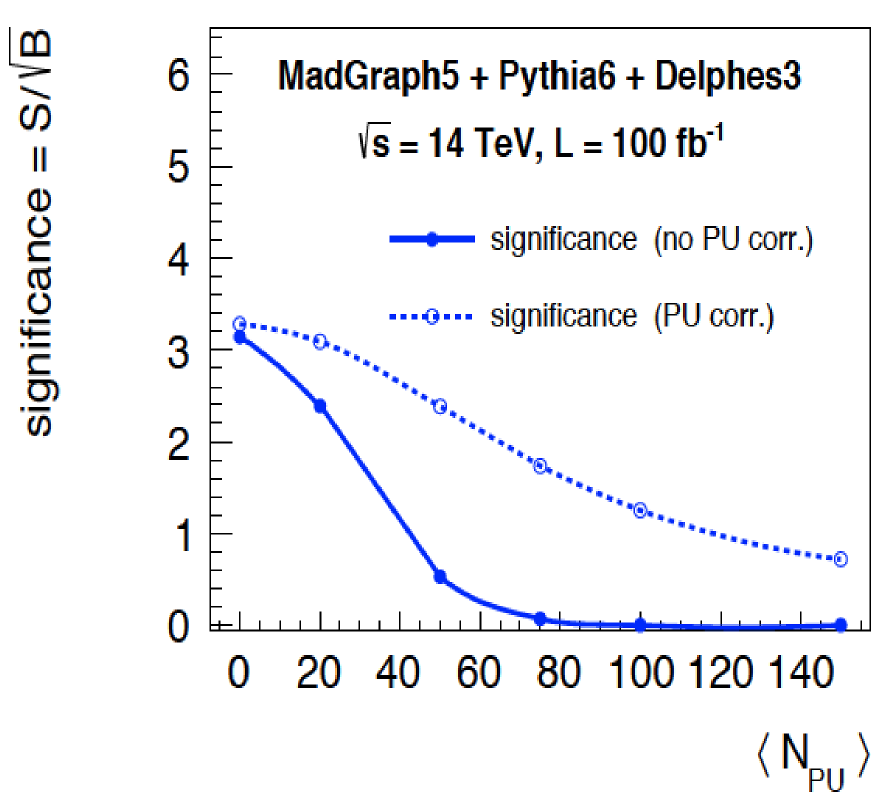
→ **depletion of rapidity gap**



# Validation: Pile-Up

- Require **large rapidity gap** between light jets, **no hadronic activity** in between
- 100 < m(bb) < 200 GeV

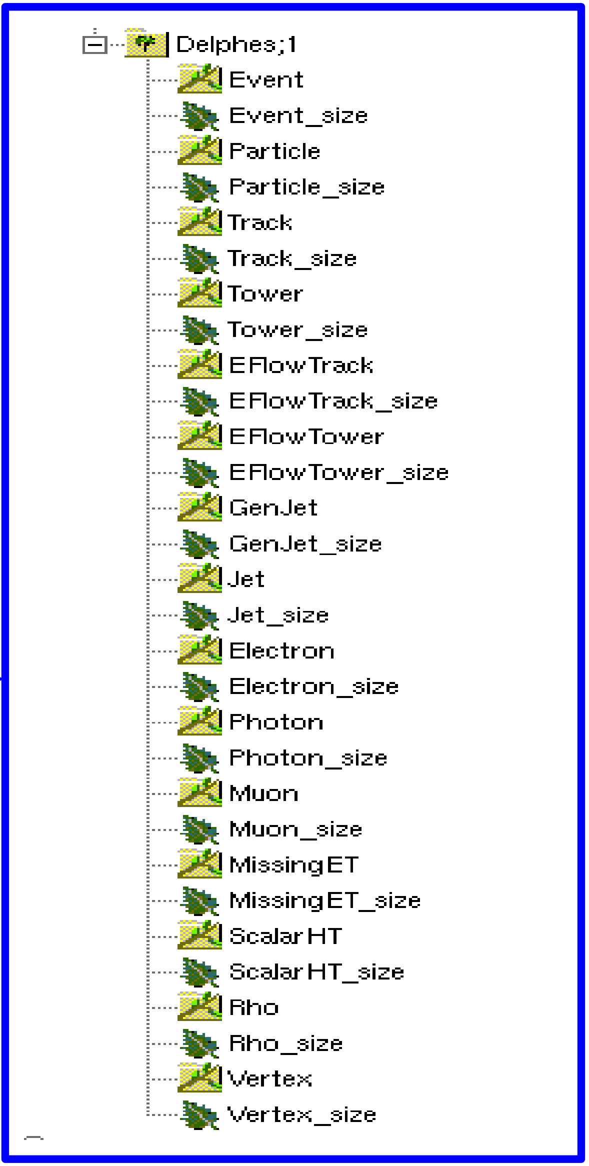


# Technical features

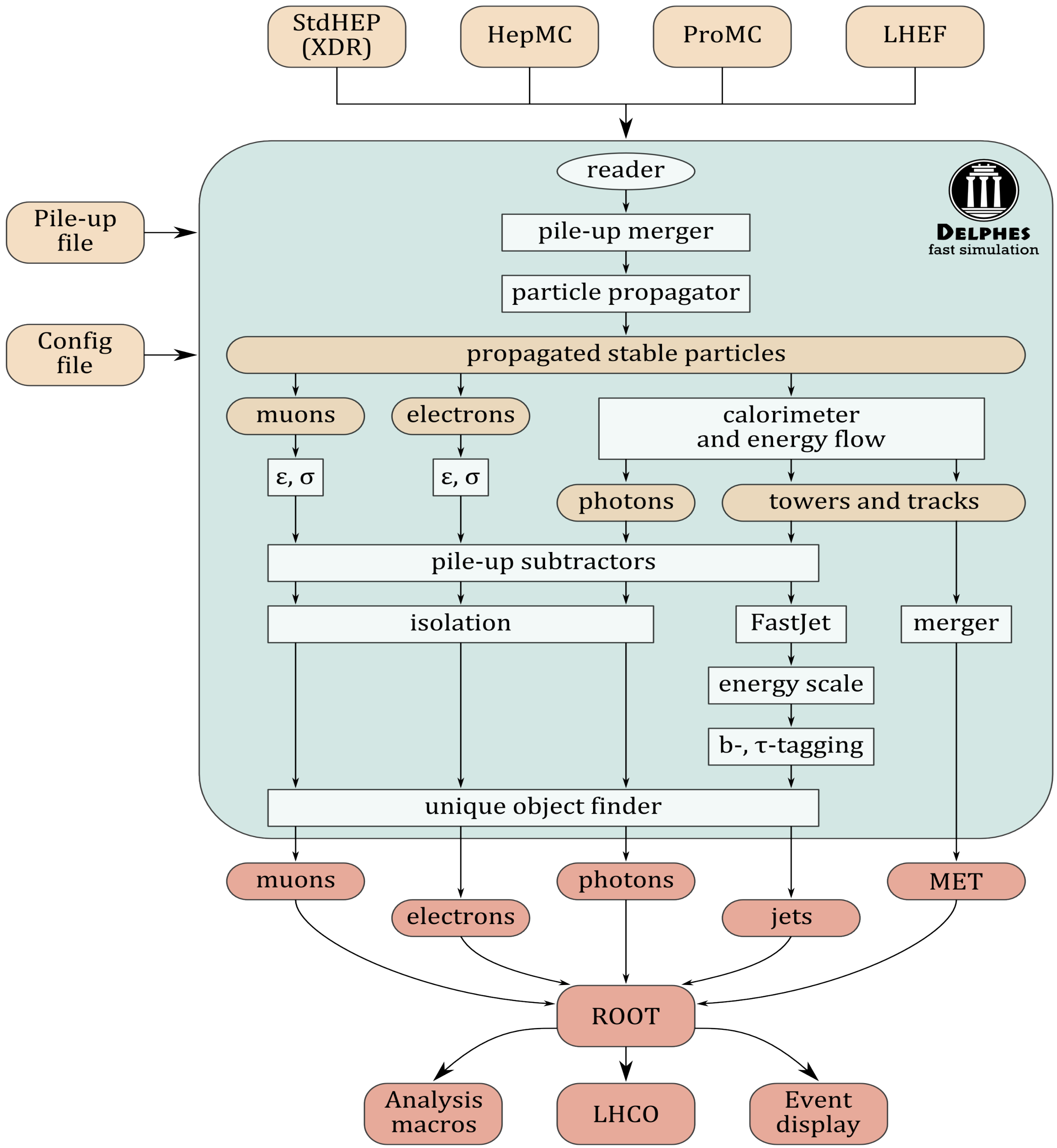
# Technical features

- **modular** C++ code, uses ROOT classes
- Input
  - Pythia/Herwig output (HepMC,STDHEP)
  - LHE (MadGraph/MadEvent)
  - ProMC
- Output
  - ROOT trees
- Configuration file
  - define geometry
  - resolution/reconstruction/selection criteria
  - output object collections



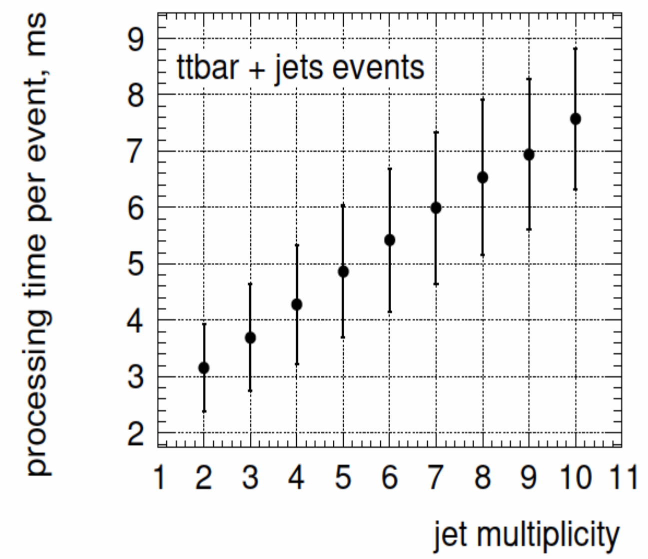

default **CMS** and **ATLAS** configurations are included in any Delphes release

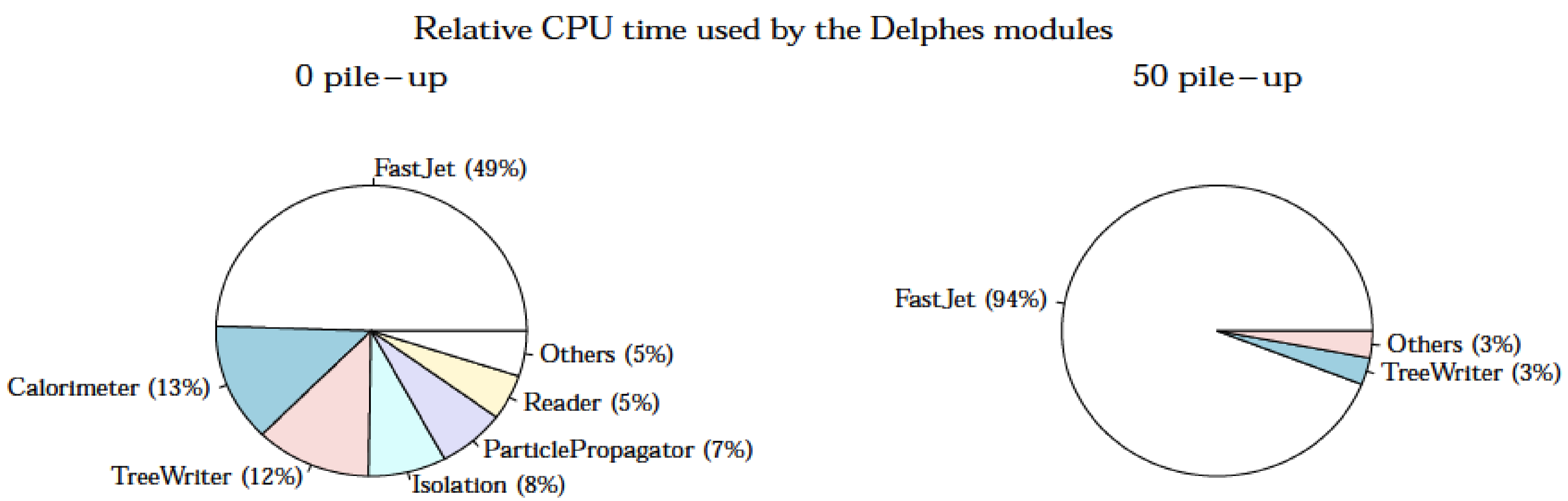
# Modularity in action



# CPU time

- event generation 1ms – 10s
- reconstruction 1ms (0 PU) – 1s (150 PU)
- ME calculation 1s – 100s



# When and when not DELPHES?

- When do you need Delphes?

→ more advanced than parton-level studies
→ testing analysis methods (multivariate/Matrix Element)
→ test your model (CheckMATE)
→ scan big parameter space (SUSY-like)
→ preliminary tests of new geometries/resolutions (upgrades, Snowmass)
→ educational purpose (bachelor/master thesis)

- When not to use Delphes?

→ high precision studies
→ very exotic topologies (heavy stable charged particles)
→ study is sensitive to tails

# Conclusions

- **Delphes 3** has been out for one year now, with **major improvements**:

  - modularity
  - pile-up implementation
  - revamped particle flow algorithm
  - new visualization tool based on ROOT EVE
  - default cards giving results on par with published performance from LHC experiments
  - now fully integrated within MadGraph5

- To-do list:

  - PileUpJetID (see Seth Senz next talk)
  - Timing Detector
  - Neighboring cells energy sharing in calorimeters

- Delphes 2 is no longer supported!!

- Test it, and give us feedback!

# People

Severine Ovyn
Xavier Rouby
Jerome de Favereau
Christophe Delaere
Pavel Demin
Andrea Giammanco
Vincent Lemaitre
Alexandre Mertens
M.S.

the community ...