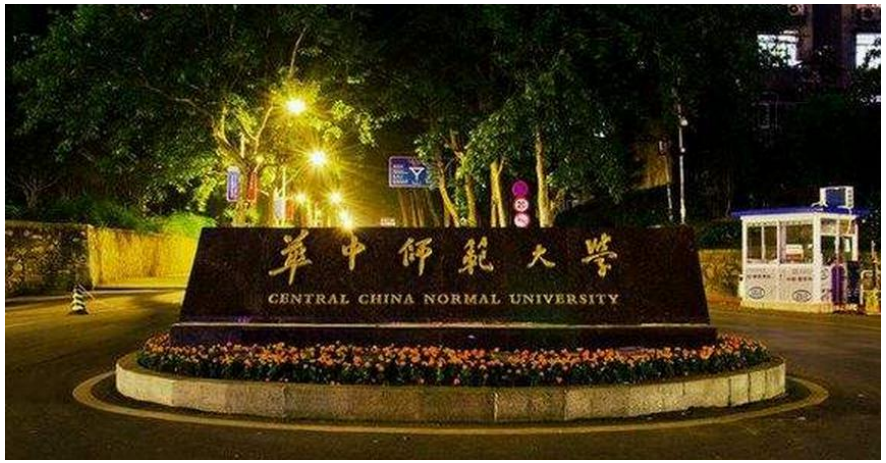
DELPHES3 for fast simulation of a generic collider experiment

Gang Li, Manqi Ruan, Qiang Li,Zhenwei Cui,Lei Wang,**Cheng Chen**

Wuhan CEPC-2017

Central China Normal University

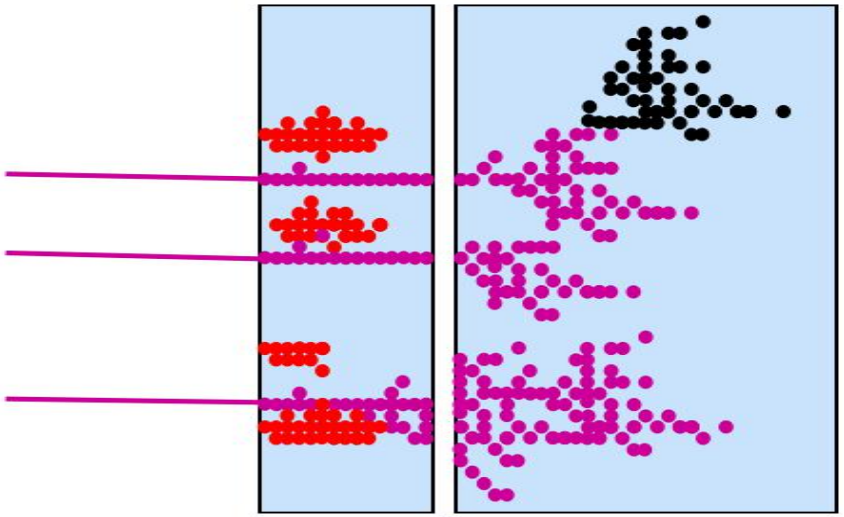


- **Introduction**
  - Delphes brief introduction
  - Delphes Module
- **MadGraph for ISR**
- **Event display**
- **Delphes Validation**
  - Pythia8 Generate Events
  - Efficiency and Resolution Check
- **Delphes for CEPC**
  - Muon Resolution : Dimu Mass Check with Full Simulation
  - Jet Reconstruction : Invariant mass, Recoiling mass Check with Full Simulation
- **Summary**

Introduction

Delphes is a C++ framework. The simulation includes a tracking system, embedded into a magnetic field, calorimeters and a muon system.

- Track
- Calorimeter
- Particle-flow



https://cp3.irmp.ucl.ac.be/projects/delphes

- **Modular C++ code**

- **Uses**
  - ROOT classes
  - FastJet package

- **Input**
  - Pythia/Herwig output (HepMC,STDHEP)
  - LHE (MadGraph/MadEvent)
  - ProMC

- **Configuration file**
  - Define geometry
  - Resolution/reconstruction/selection criteria
  - Output object collections

- **Output**
  - ROOT trees

- **Delphes Card Module**

  -Magnet

  -Track

  -Calorimeters

  -Electrons

  -Muons

  -Photons

  -Jet & MET

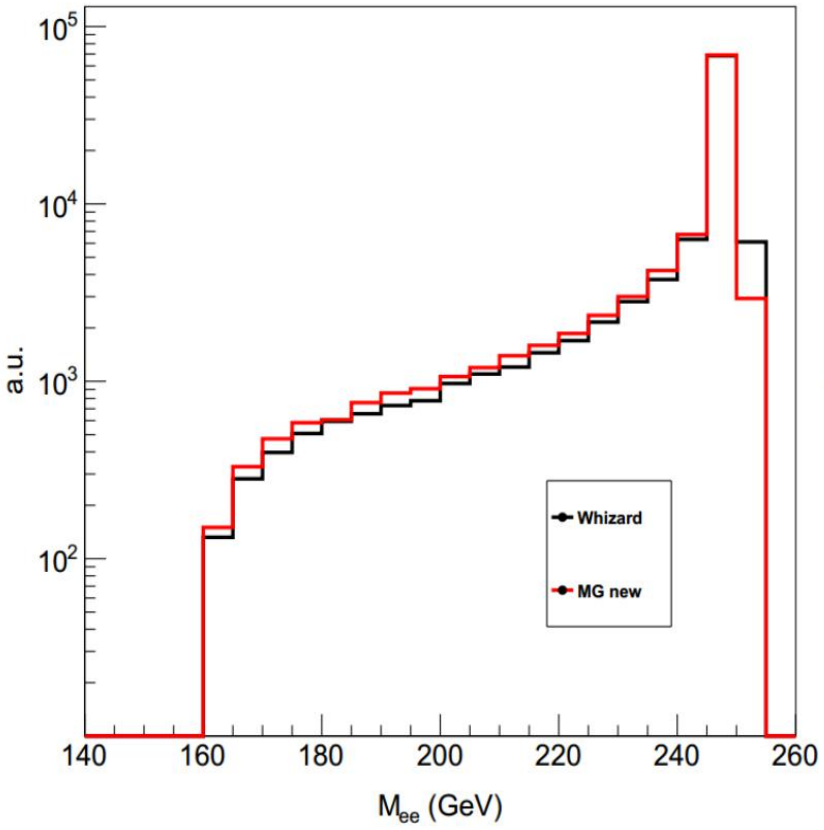
  -B-tagging /C-tagging

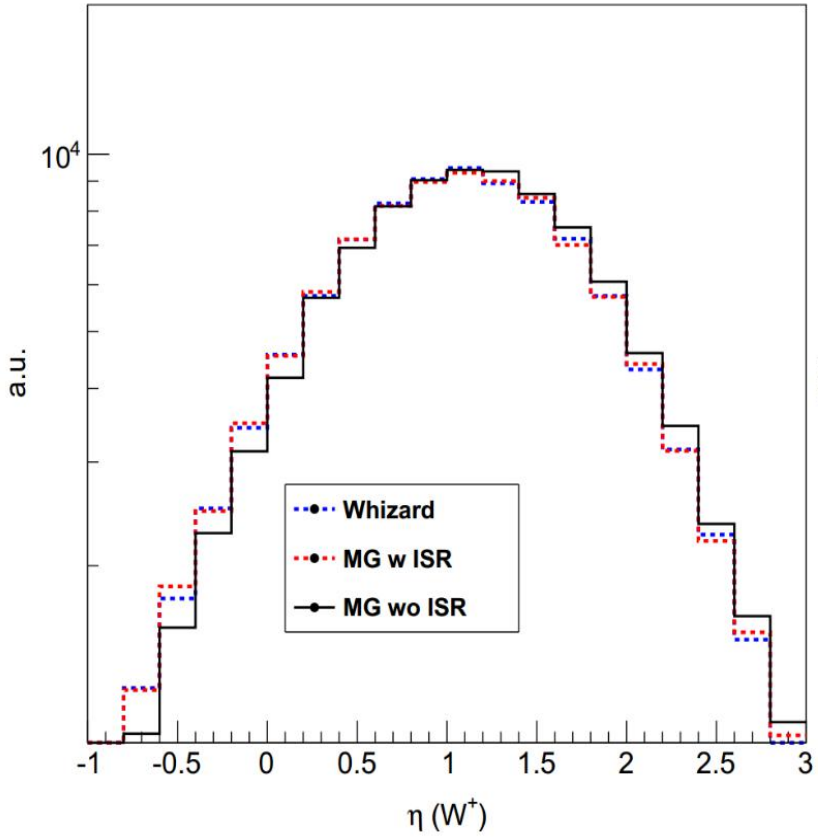
  -Tau-tagging

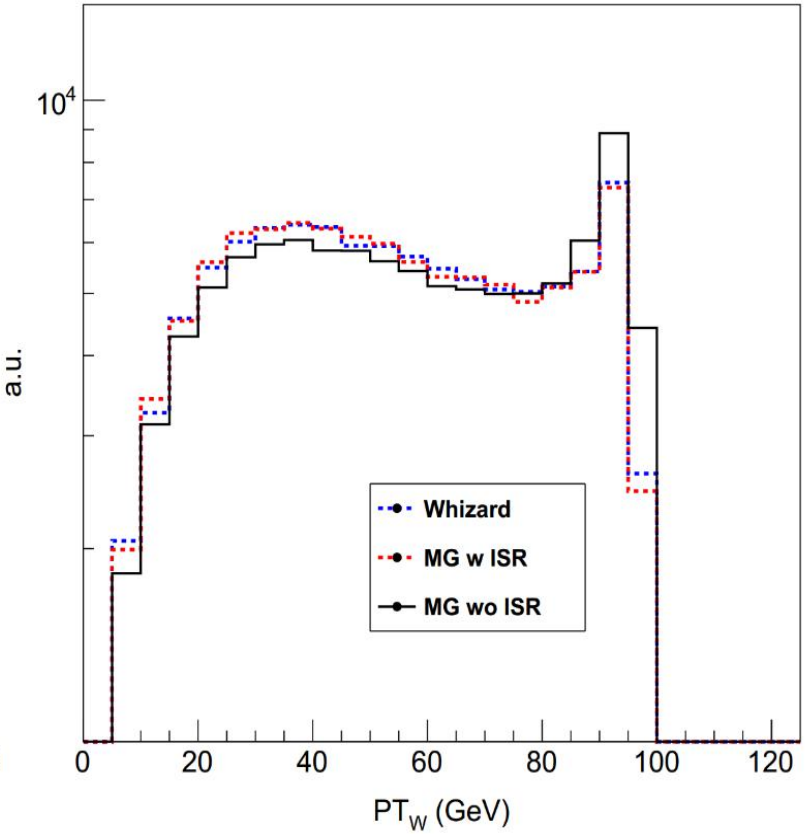
MadGraph for ISR

arXiv: 1505.01270 Beamstruhlung effect at CEPC is small, however, ISR effect is not small

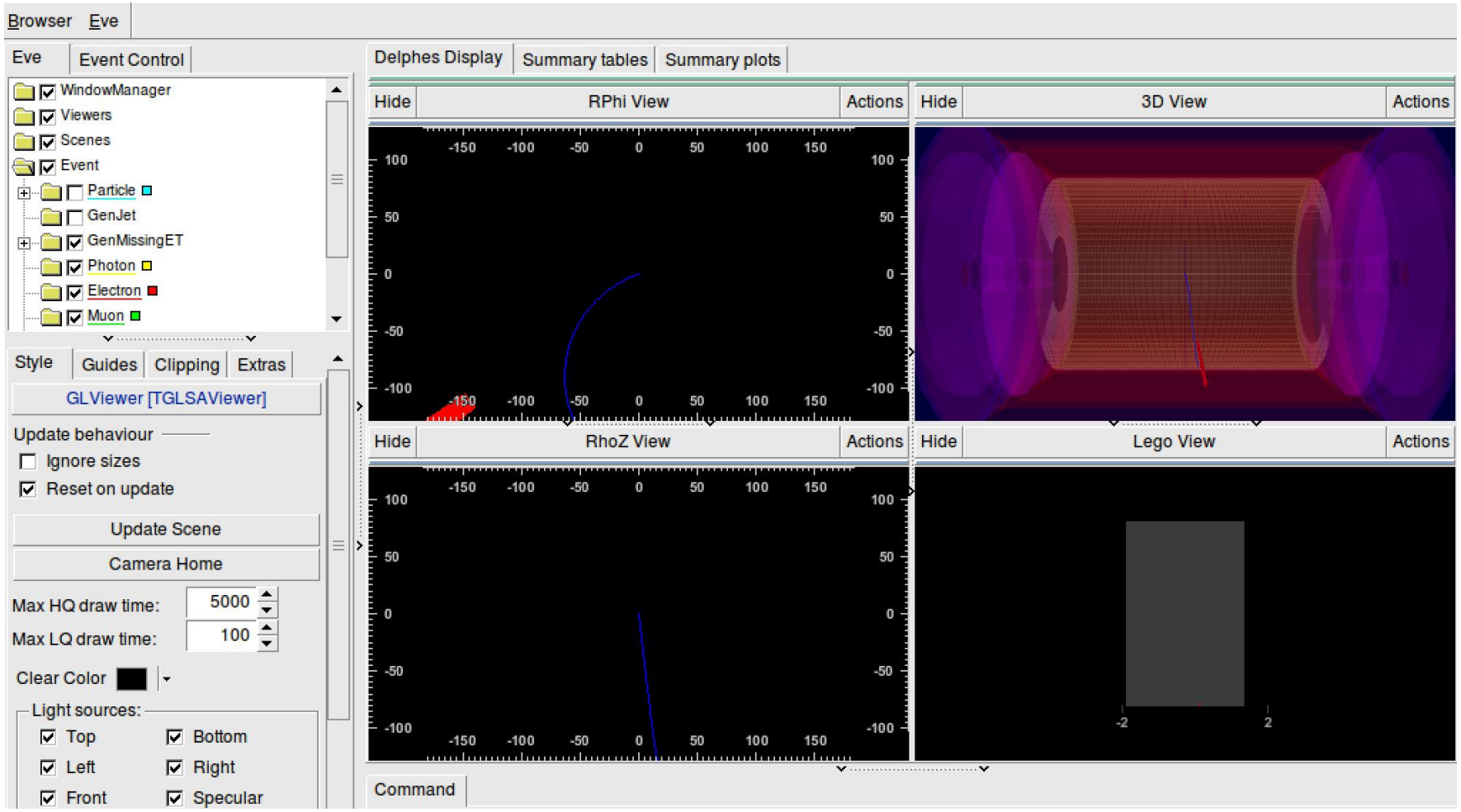
ISR effect has been added to MG successfully, with checks done for e.g. e+e-> w+w- and Z H







Event display



display.C ''Calorimeters''change to''ECal,HCal''

Delphes Validation

- Pythia8 Generate Events

```
void fillParticle(int id, double pt, double thetaIn, double phiIn,
  Event& event, ParticleData& pdt, Rndm& rndm, bool atRest = false) {

  // Reset event record to allow for new event.
  event.reset();

  // Select particle mass; where relevant according to Breit-Wigner.
  double mm = pdt.mSel(id);

  if (atRest) {
   double ee = mm;
  double  pp = 0.;
  }
//cout<<atRest<<endl;
  // Angles as input or uniform in solid angle.
  double cThe, sThe, phi;
  if (thetaIn >= 0.) {
    cThe = cos(thetaIn);
    sThe = sin(thetaIn);
    phi  = phiIn;
  } else {double qq=0.1*(rand()%10)+0.01*(rand()%10)+0.001*(rand()%10)
  +0.0001*(rand()%10)+0.00001*(rand()%10);
    cThe = 1-2*qq;
    sThe = sqrtpos(1. - cThe * cThe);
    phi = 2. * M_PI * qq;
  }
double pp = pt/sThe;
double ee = sqrtpos(pp*pp + mm*mm);
  // Store the particle in the event record.
  event.append( id, 1, 0, 0, pp * sThe * cos(phi),
 pp * sThe * sin(phi), pp * cThe, ee, mm);
```

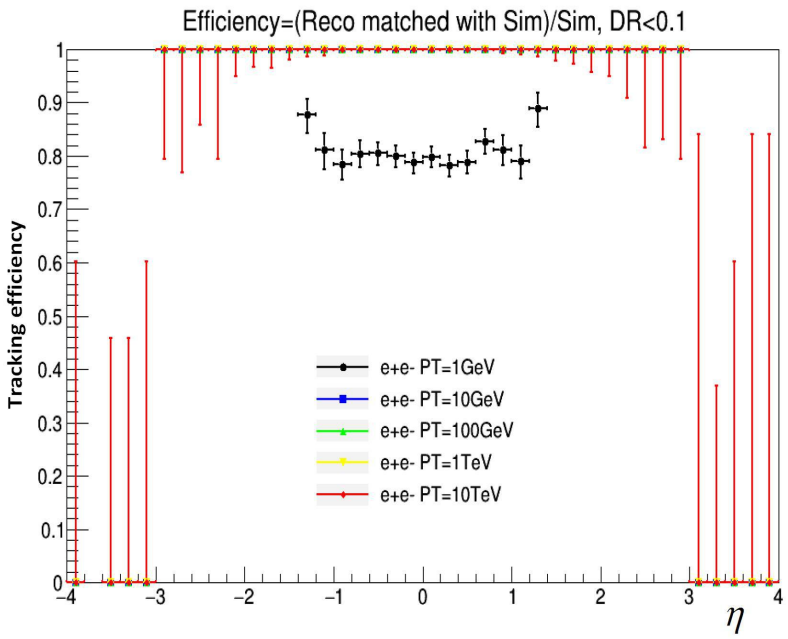
```
Pythia pythia;
Event& event      = pythia.event;
HepMC::Pythia8ToHepMC ToHepMC;
HepMC::GenEvent* hepmcevt = new HepMC::GenEvent();
ToHepMC.fill_next_event( pythia, hepmcevt );
```

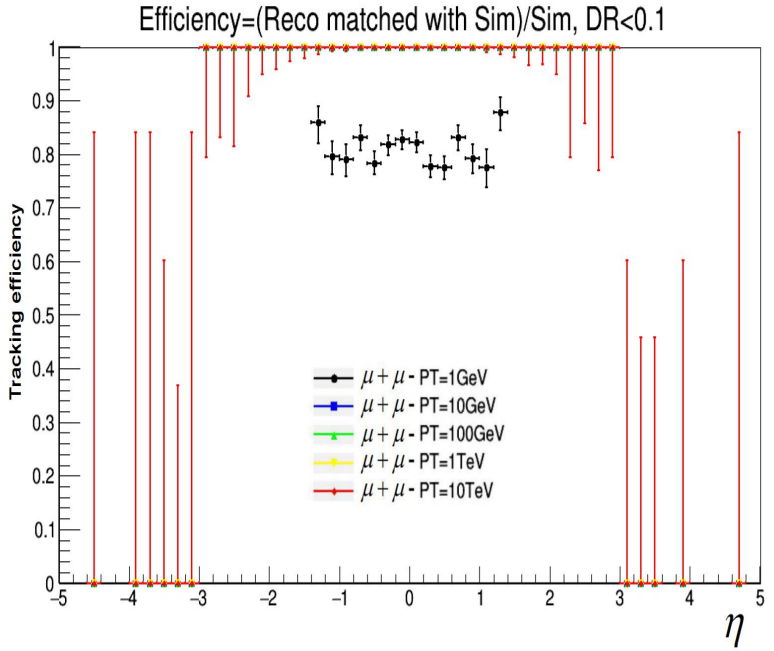
output.hep file

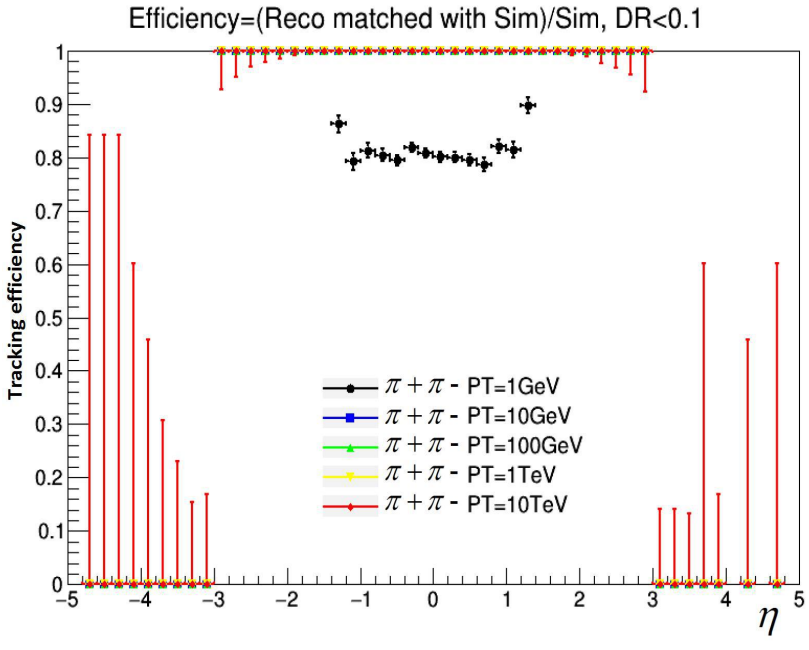
./DelphesHepMC cards/delphes_card_CEPC.tcl output.root output.hep

Efficiency and Resolution Check

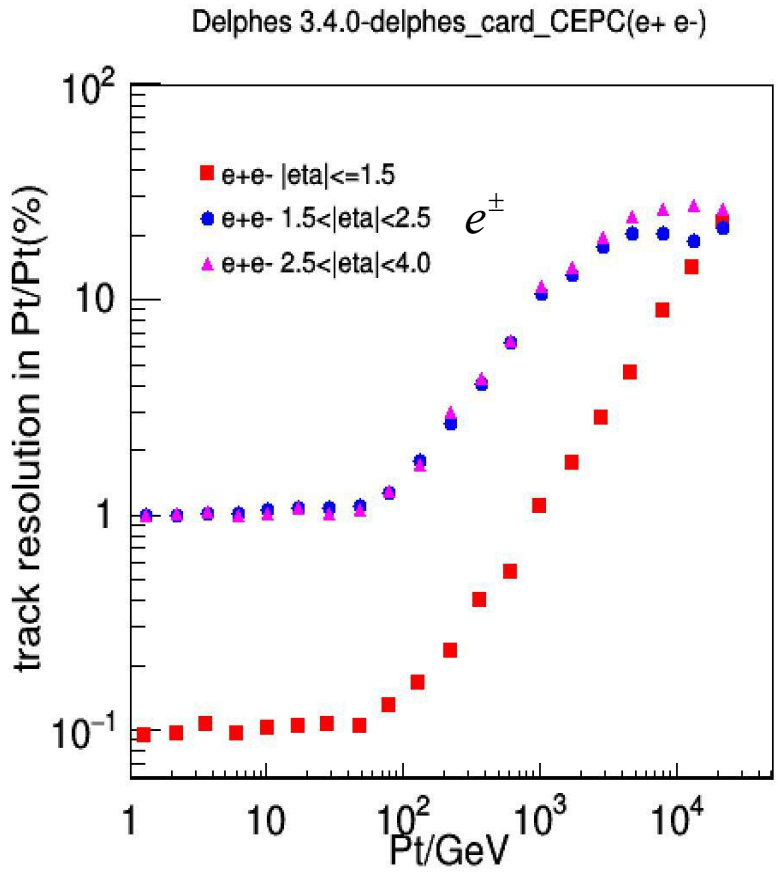
Tracking efficiency for $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$

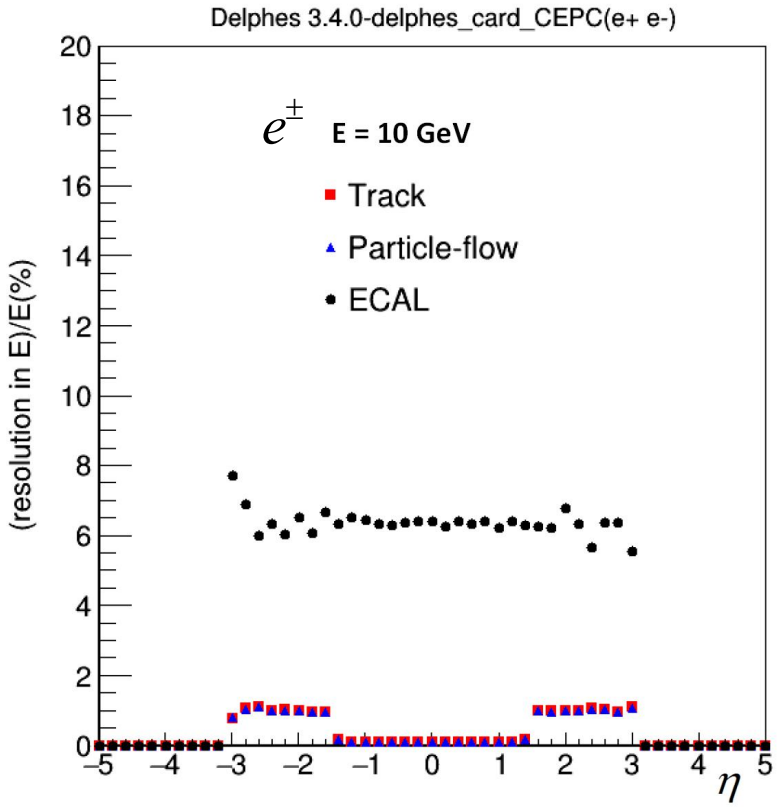


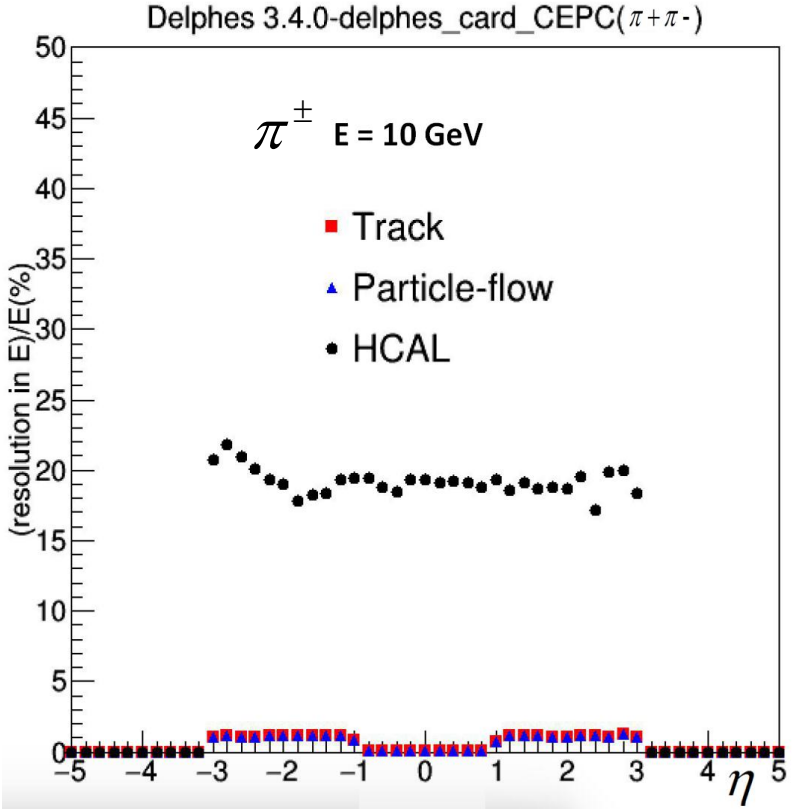





```
# tracking efficiency formula
set EfficiencyFormula {
                                    (energy <= 2.)    * (0.80) +
(abs(eta) <= 3.0)                 * (energy > 2.)     * (1.00) +
(abs(eta) >  3.0)                                     * (0.00)}
```








```
# resolution formula for electrons
set ResolutionFormula {    (abs(eta) <= 1.5)                    * sqrt(0.001^2 + pt^2*1.e-5^2) +
                           (abs(eta) > 1.5 && abs(eta) <= 4.0) * sqrt(0.01^2 + pt^2*1.e-4^2)}
```

Delphes for CEPC

· CEPC Delphes Card

CEPC: magnetic field

```
# radius of the magnetic field coverage, in m
 set Radius 1.8
 # half-length of the magnetic field coverage, in m
 set HalfLength 2.35
 # magnetic field
 set Bz 3.5
```

http://cepc.ihep.ac.cn/preCDR/main_preCDR.pdf

CEPC: calorimeter cell

ECAL:

```
# 1.0 degree towers (3 cm x 3 cm)
set PhiBins {} for {set i -180} {$i <= 180} {incr i} {
 add PhiBins [expr {$i * $pi/180.0}]
 }
# 0.02 unit in eta up to eta = 3.0
for {set i -150} {$i <= 150} {incr i} {
set eta [expr {$i * 0.02}]
add EtaPhiBins $eta $PhiBins
 }
```

HCAL:

```
# 2.0 degree towers (6 cm x 6 cm)
set PhiBins {} for {set i -180} {$i <= 180} {incr i} {
 add PhiBins [expr {$i * $pi/180.0}]
 }
# 0.04 unit in eta up to eta = 3.0
for {set i -75} {$i <= 75} {incr i} {
set eta [expr {$i * 0.04}]
add EtaPhiBins $eta $PhiBins
 }
```

· CEPC Delphes Card

```
 # resolution formula for muons
set ResolutionFormula {(abs(eta) <= 3.0)*pt*sqrt((2.0*
1.e-5)^2+(1.e-3)^2/(pt^2*((2^2)/(exp(eta)+exp(-eta))^2)))}

# tracking efficiency formula for muons
set EfficiencyFormula {(energy <= 10.0)   * (0.00) +
    (abs(eta) <= 3.0)  * (energy > 10.0)  * (0.985) +
    (abs(eta) >  3.0)                     * (0.00)}

 # tracking efficiency formula for electrons
 set EfficiencyFormula {(energy <= 10.0)   * (0.00) +
      (abs(eta) <= 3.0) * (energy > 10.0) * (0.995) +
                          (abs(eta) >  3.0) * (0.00)}
```

resolution formula:

$$\sigma_{1/p_{\mathrm{T}}} = 2 \times 10^{-5} \oplus 1 \times \frac{10^{-3}}{p_{\mathrm{T}} \cdot \sin\theta.}$$

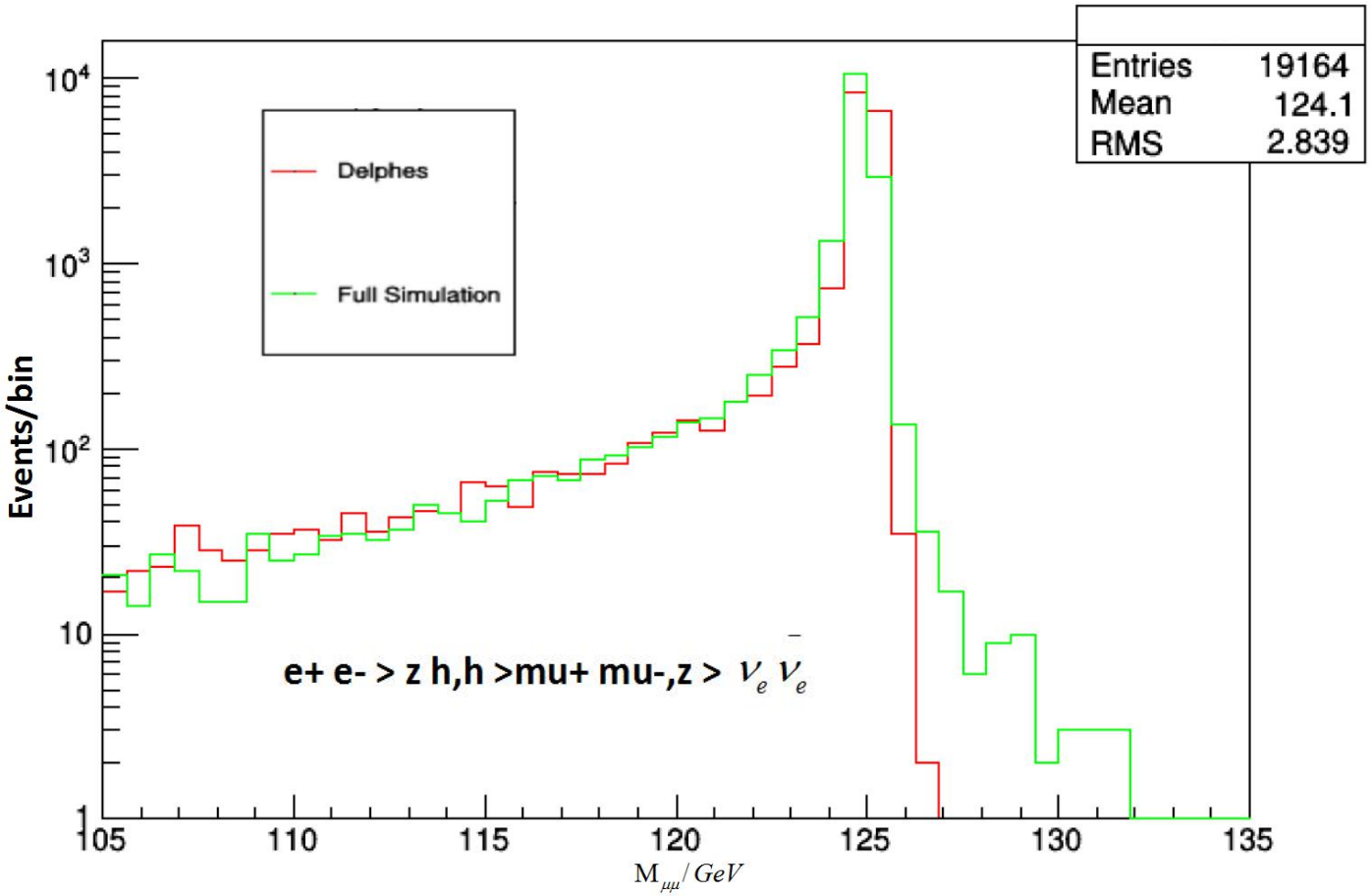
Muon identification efficiency ($E > 10$ GeV)

98.5%

http://cepc.ihep.ac.cn/preCDR/main_preCDR.pdf

Dimu Mass Check with Full Simulation

Delphes compare with CEPC Full simulation



Madgraph Generate Events

resolution formula:

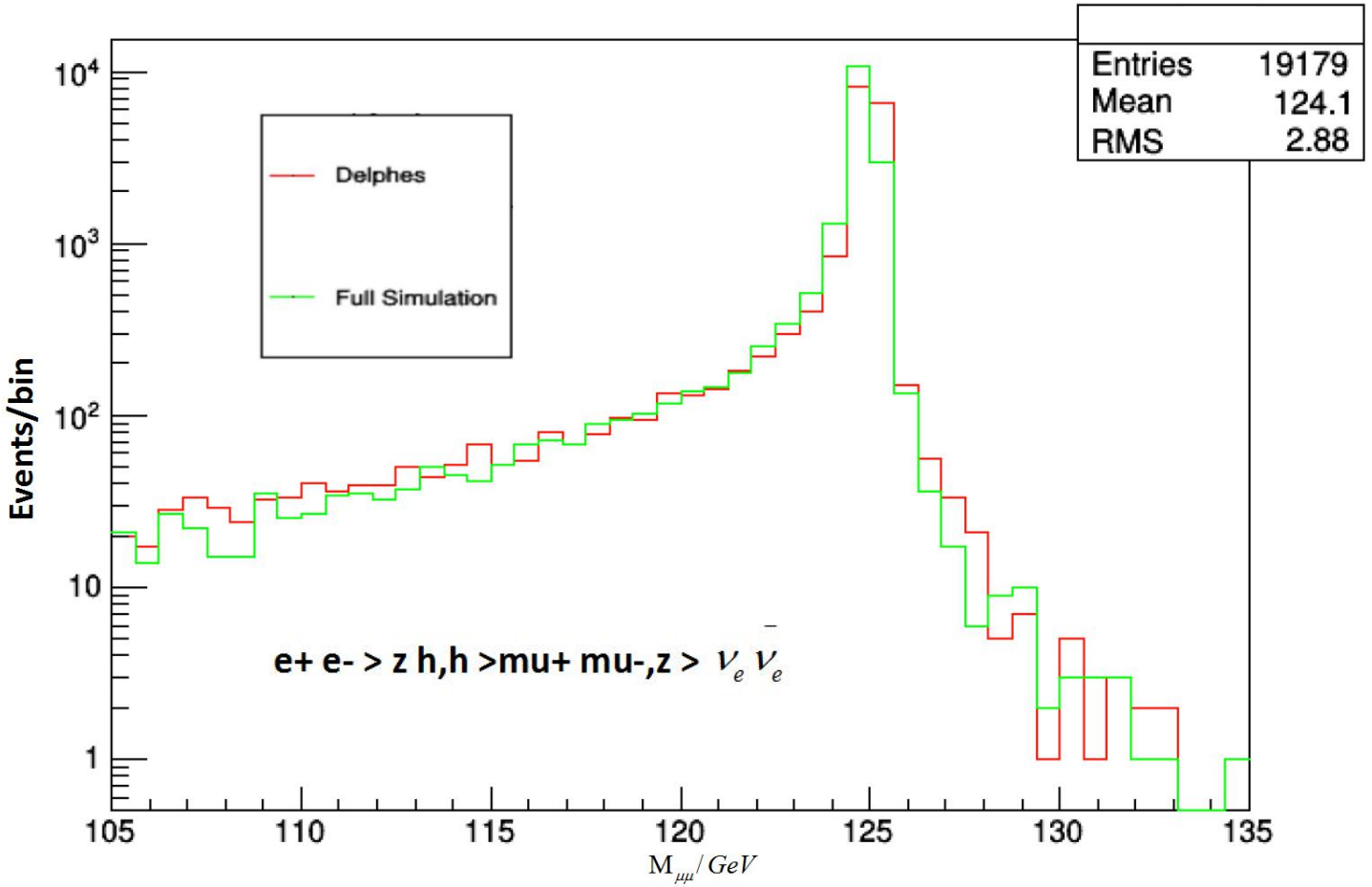
$$\sigma_{1/p_T} = 2 \times 10^{-5} \oplus 1 \times \frac{10^{-3}}{p_T \cdot \sin\theta.}$$

Muon identification efficiency ($E > 10$ GeV)

98.5%

```
 # resolution formula for muons
set ResolutionFormula {(abs(eta) <= 3.0)*pt*sqrt((2.0*
1.e-5)^2+(1.e-3)^2/(pt^2*((2^2)/(exp(eta)+exp(-eta))^2)))}
```

Delphes compare with CEPC Full simulation



The latest resolution formula for muon

```
float RefEnergy[7] = {5, 10, 20, 40, 60, 80, 100};
float kCoeff[7] = {0.000211212, 0.000107858, 6.01923e-05,
      3.69668e-05, 2.8476e-05, 2.44015e-05, 2.24783e-05};
        if(e < 5)
        {
                ObjkCoeff = 0.000267917;
        }
        else if(e > 100)
        {
                ObjkCoeff = 3.61945e-05;
        }
        else
        {
                for(int i = 0; i < 6; i++)
        if(e < RefEnergy[i+1] && e > RefEnergy[i])
        {
                 kCoeffLow = kCoeff[i];
                 kCoeffHigh = kCoeff[i+1];
                 EnDisToLow = e - RefEnergy[i];
                 EnDisToHigh = RefEnergy[i+1] - e;
                        ObjkCoeff = (kCoeffLow*EnDisToHigh +
      kCoeffHigh*EnDisToLow)/(RefEnergy[i+1] - RefEnergy[i]);
                }
        }
}
if(cosTheta > 0.86)     //Scale as effective R^2
{
        Scalefactor = (1.0/(cosTheta*cosTheta) - 1)*2.96;
             //2.96 = (Half_Z/Radius)**2
        ObjkCoeff = ObjkCoeff*1.0/Scalefactor;
}
res = ObjkCoeff*e;      // To be extended
```

Jet Reconstruction

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta y^2 + \Delta\phi^2}{R^2} \qquad d_{iB} = k_{ti}^{2p}$$

for CEPC, No ee beam effect

ee_kt_algorithm

S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, Phys. Lett. B 269, 432 (1991)

name	$d_{ij} =$	$d_{iB} =$	remark
ee_kt_algorithm	$2(1-\cos\theta_{ij})\frac{\min(E_i^2, E_j^2)}{s}$	-	also known as Durham
kt_algorithm	$\min(p_{t,i}^2, p_{t,j}^2)\frac{(y_i-y_j)^2+(\phi_i-\phi_j)^2}{R^2}$	$p_{t,i}^2$	y is pseudorapidity
cambridge-aachen	$\min(p_{t,i}^0, p_{t,j}^0)\frac{(y_i-y_j)^2(\phi_i-\phi_j)^2}{R^2}$	$p_{t,i}^0$	no energy weighting
antikt_algorithm	$\min(p_{t,i}^{-2}, p_{t,j}^{-2})\frac{(y_i-y_j)^2+(\phi_i-\phi_j)^2}{R^2}$	$p_{t,i}^{-2}$	start with merging high energy particles

ee_kt_algorithm for Jet Reconstruction:

No defined ee_kt

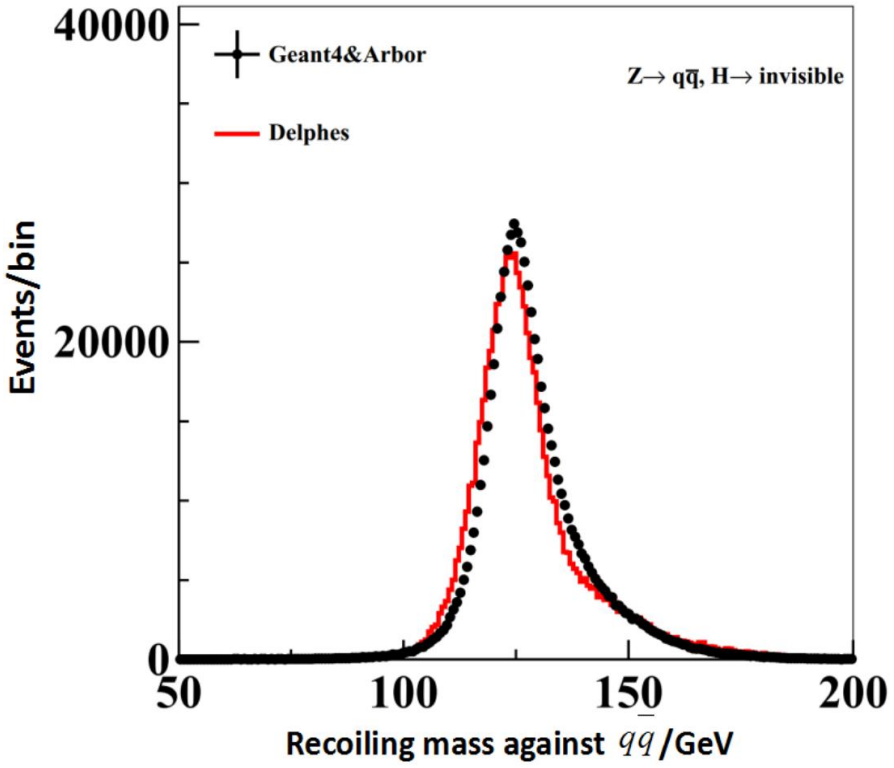
```
module FastJetFinder FastJetFinder {
#  set InputArray Calorimeter/towers
  set InputArray EFlowMerger/eflow

  set OutputArray jets

  # algorithm: 1 CDFJetClu,
  # 2 MidPoint,
  # 3 SIScone, 4 kt,
  # 5 Cambridge/Aachen, 6 antikt
  set JetAlgorithm 9
  set ParameterR 0.5
  set JetPTMin 20.0
```

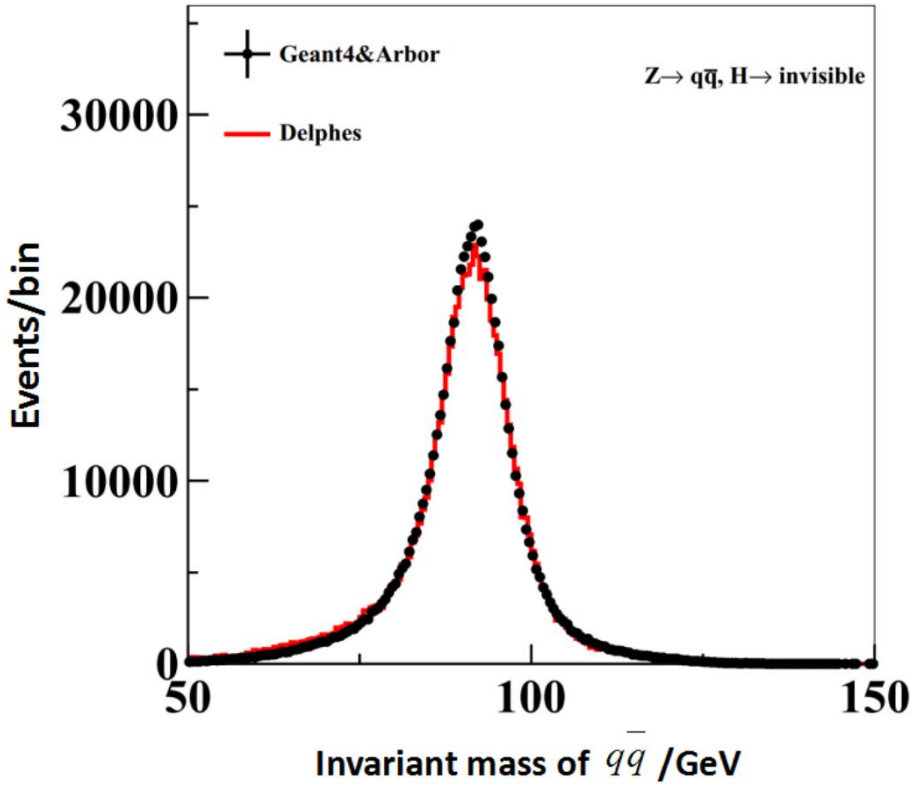
What to add

```
      case 9:
        fDefinition = new JetDefinition(ee_kt_algorithm);
        break;
```

```
if (fJetAlgorithm == 9){
   outputList = sorted_by_pt(sequence->exclusive_jets(4));
}
else {
outputList = sorted_by_pt(sequence->inclusive_jets(fJetPTMin));
}
```

Delphes compare with CEPC Full simulation





Z→ qq̄, H→ invisible

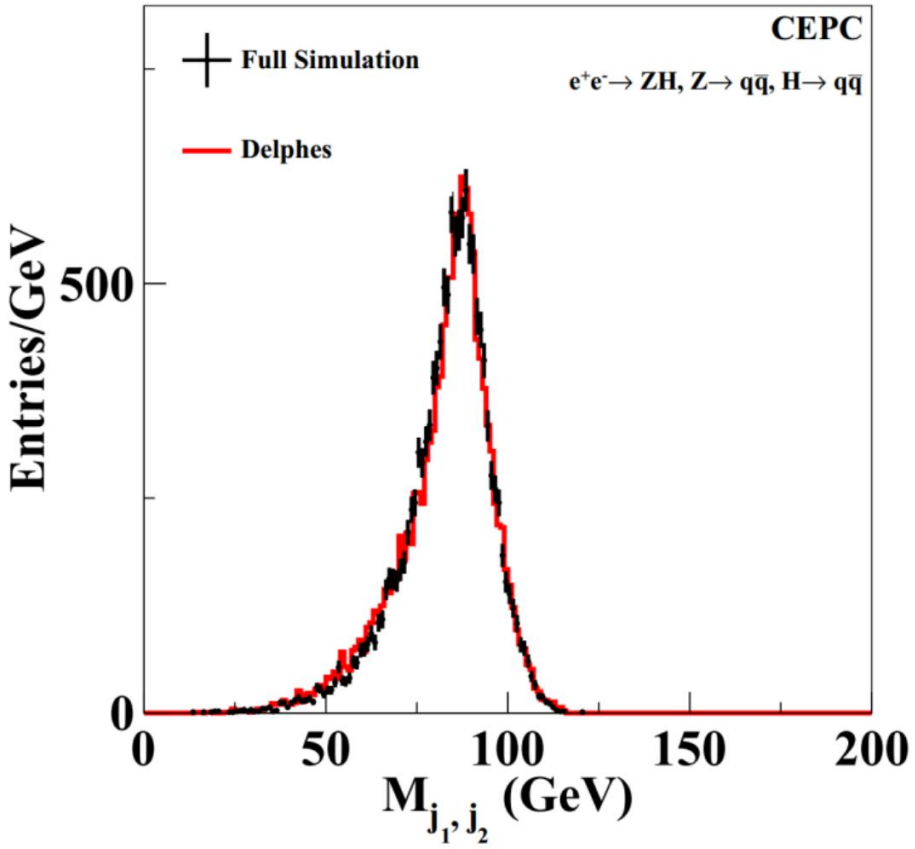
Reconstruction algorithm:

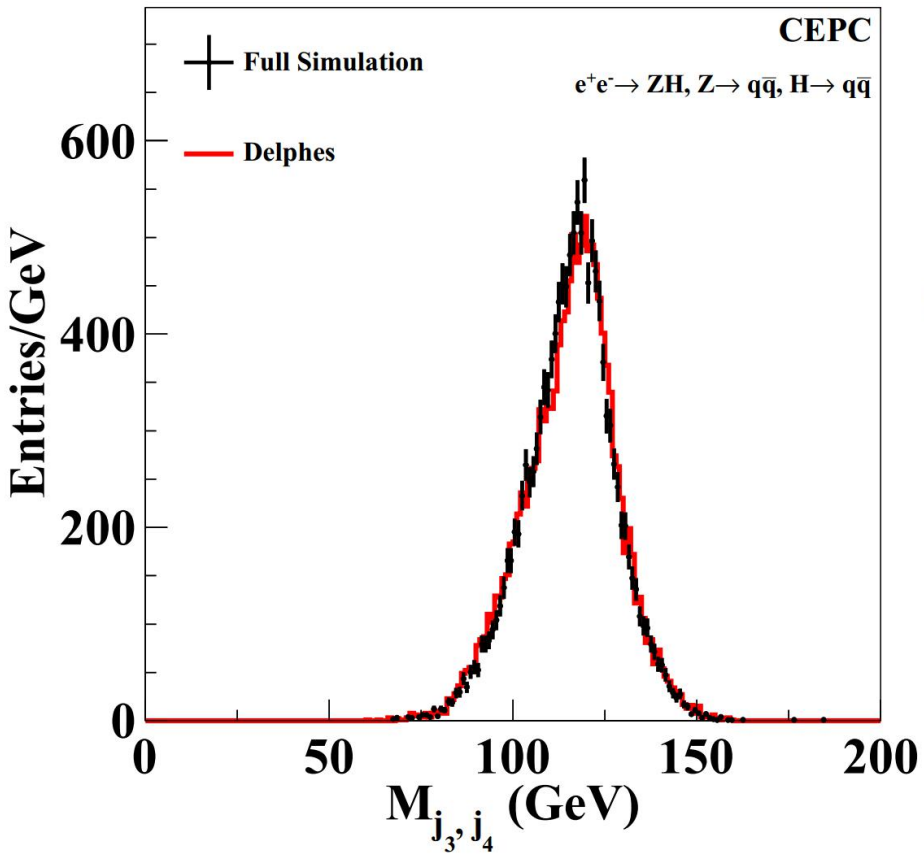
e^+e^- k_t algorithm

Resolution formula:

$$\sigma = \sqrt{0.001^2 + (10^{-5} p_t)^2}$$

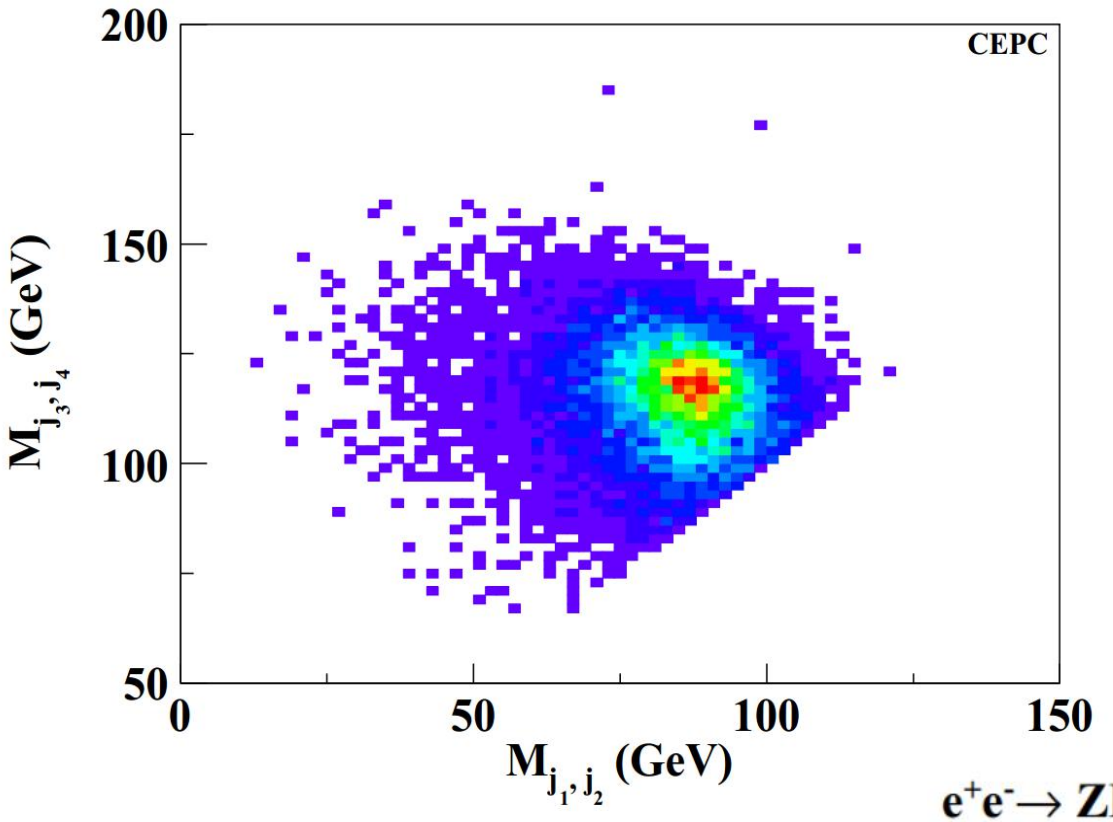
Delphes compare with CEPC Full simulation





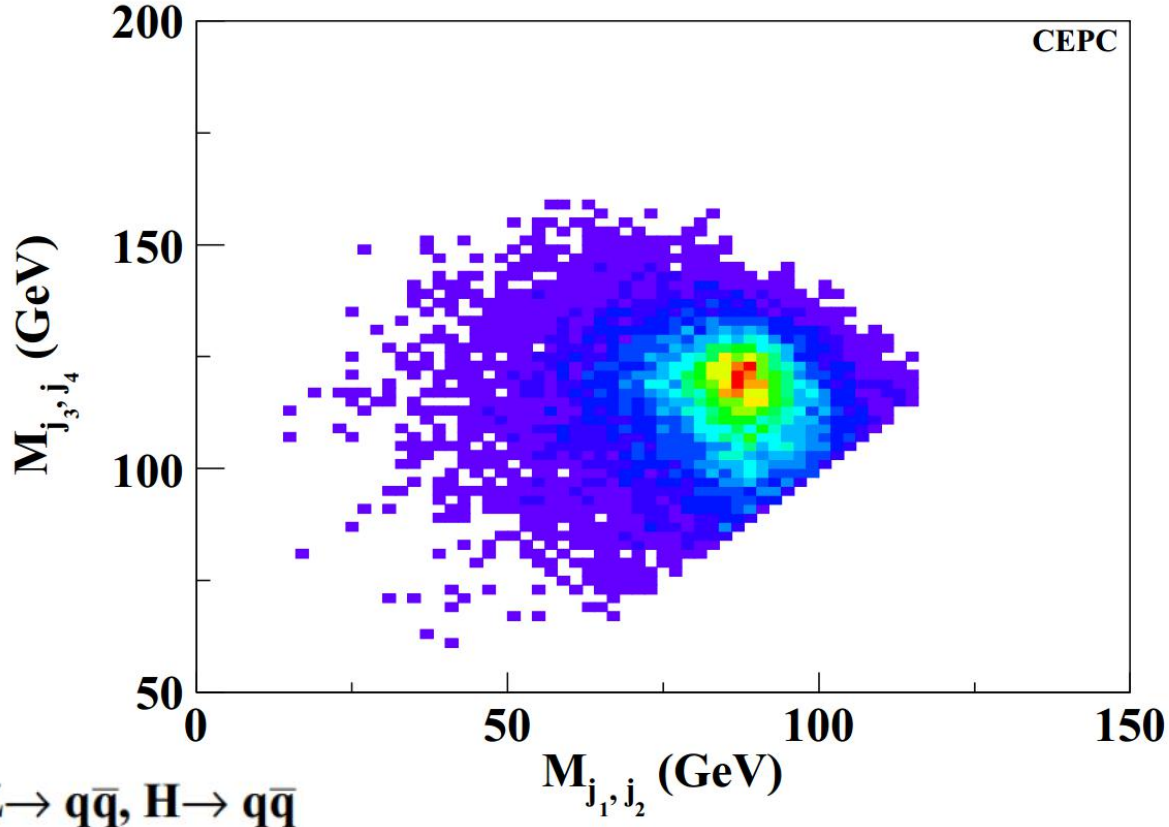
$e^+e^- \to ZH, Z \to q\bar{q}, H \to q\bar{q}$

Delphes compare with CEPC Full simulation



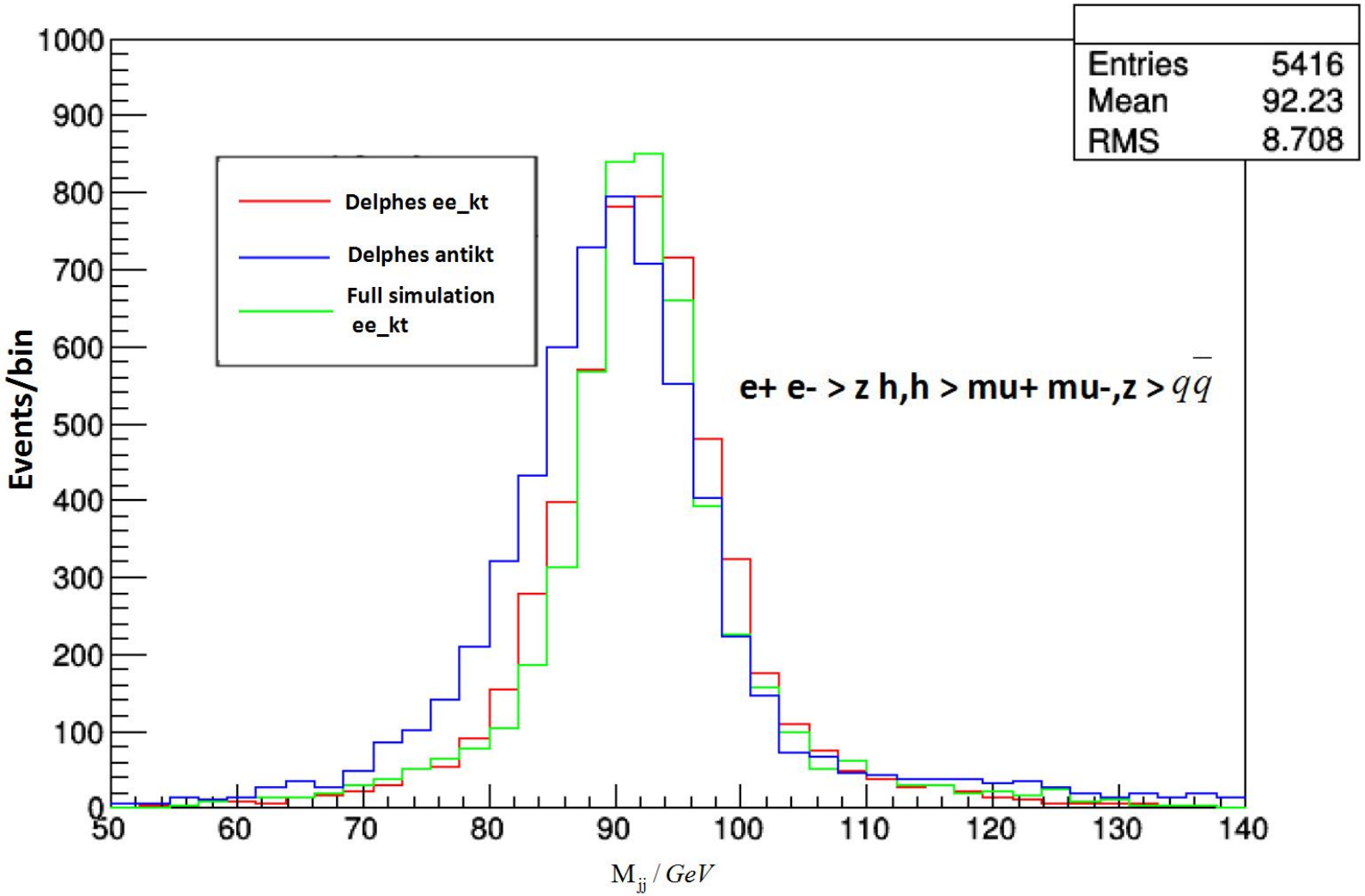
full simulation

$e^+e^- \rightarrow ZH, Z \rightarrow q\bar{q}, H \rightarrow q\bar{q}$



fast simulation

Delphes compare with CEPC Full simulation (Under study)



Summary

- Delphes Module
- MadGraph for ISR
- Event display
- Pythia8 Generate Events
- Efficiency and Resolution Check
- Dimu Mass Check with Full Simulation
- Jet Reconstruction Check with Full Simulation
- Next to do
  - More delphes check with CEPC Full simulation
  - Tau: temporary from FCC
  - B-Tagging and fake rate: from CEPC pre-study