Fast simulation for CEPC

Gang Li

IHEP, November 3rd, 2017

国际创新海外团队，院前沿重点项目&所创新团队2017年联合年会

Outline

✓ **Introduction**

✓ **Delphes card for CEPC simulation**

- ● **Tracks, photons, and neutral hadrons**
- ● **Jet clustering：exclusive mode**

✓ **Validations**

✓ **New add-ons**

- ● **Exclusive jets**
- ● **Flavor Tagging**

✓ **Analysis examples**

- ✓ e^+e^- → **qqH, H→qq**
- ✓ e^+e^- → **qqH, H→$\mu\mu$**

✓ **Summary**

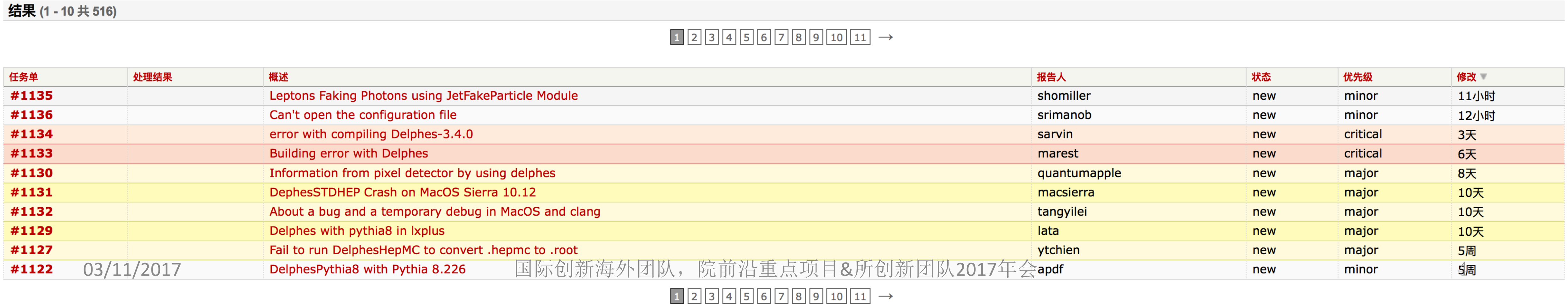
Introduction

Delphes is a C++ framework to simulate detector response. The detector includes a tracking and calorimetric system embedded into a magnetic field, as well as a muon system.

- ✧ **Modular design, interfaces to external package** … **easy to use**
- ✧ **Well documented and maintained, "ticket" scheme for bug-report and trace**

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

Code : "git clone https://github.com/delphes/delphes.git"



结果 (1 - 10 共 516)

1 2 3 4 5 6 7 8 9 10 11 →

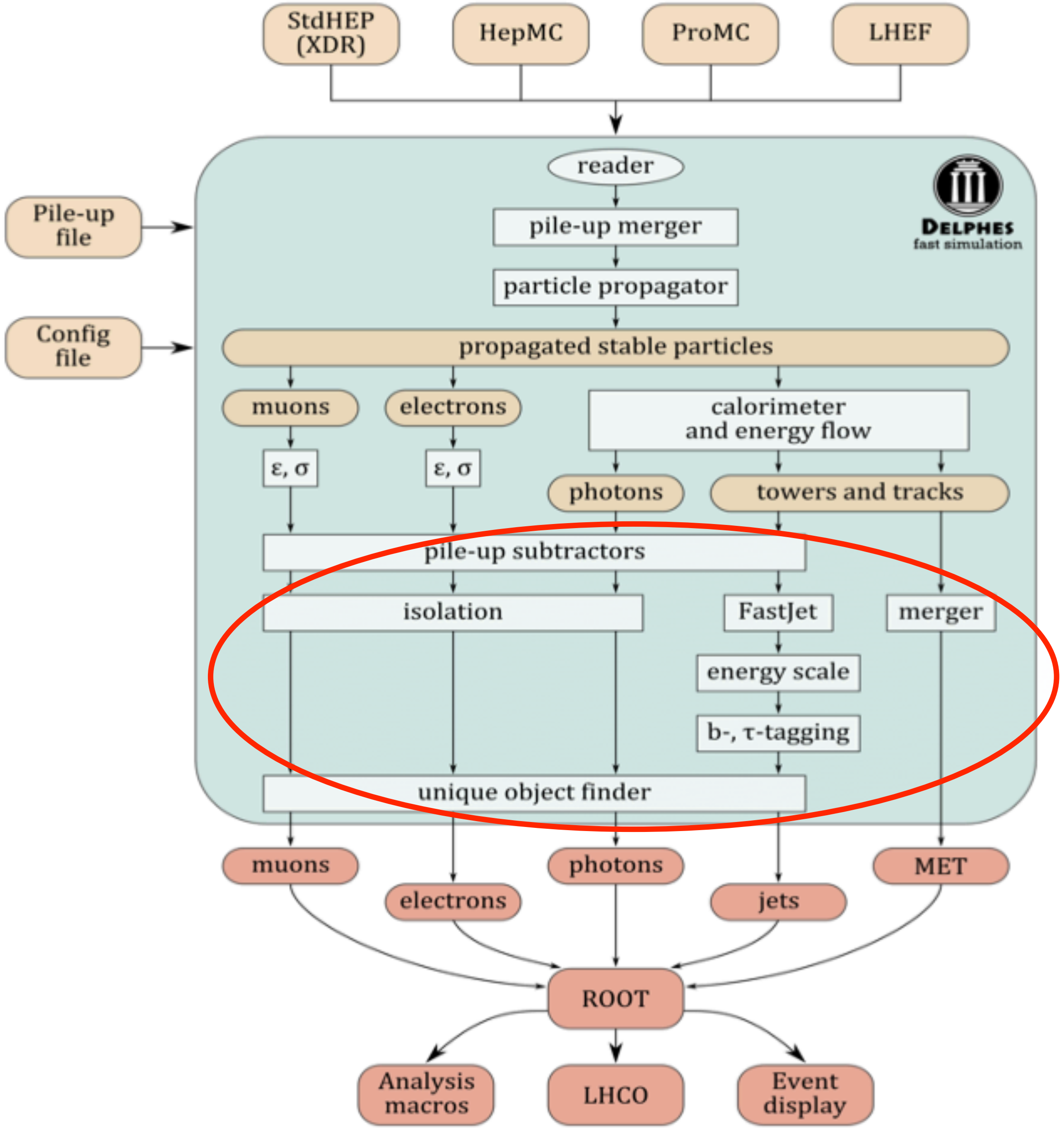
任务单	处理结果	概述	报告人	状态	优先级	修改 ▼
#1135		Leptons Faking Photons using JetFakeParticle Module	shomiller	new	minor	11小时
#1136		Can't open the configuration file	srimanob	new	minor	12小时
#1134		error with compiling Delphes-3.4.0	sarvin	new	critical	3天
#1133		Building error with Delphes	marest	new	critical	6天
#1130		Information from pixel detector by using delphes	quantumapple	new	major	8天
#1131		DephesSTDHEP Crash on MacOS Sierra 10.12	macsierra	new	major	10天
#1132		About a bug and a temporary debug in MacOS and clang	tangyilei	new	major	10天
#1129		Delphes with pythia8 in lxplus	lata	new	major	10天
#1127		Fail to run DelphesHepMC to convert .hepmc to .root	ytchien	new	major	5周
#1122		DelphesPythia8 with Pythia 8.226	apdf	new	minor	5周

1 2 3 4 5 6 7 8 9 10 11 →

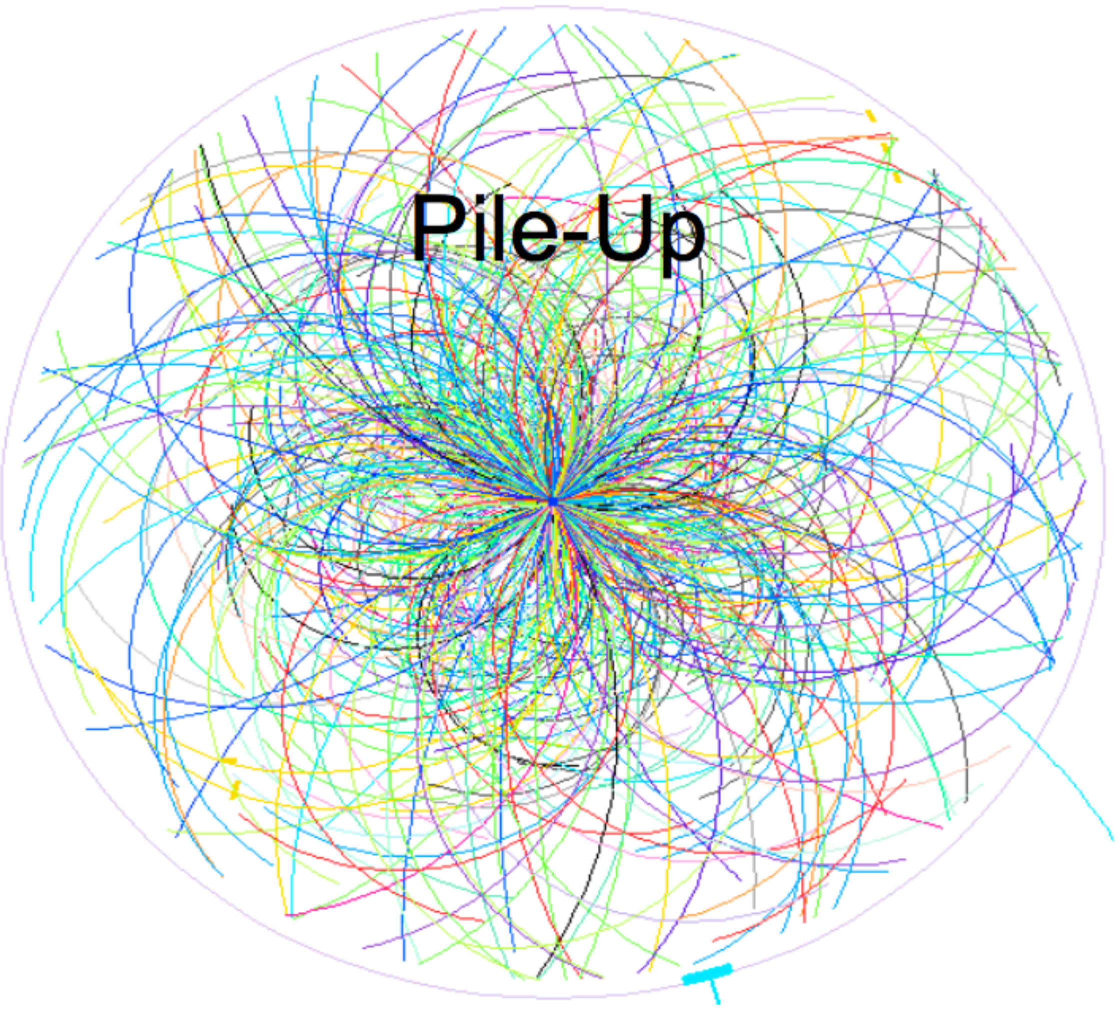
Designed for hadron collider experiments

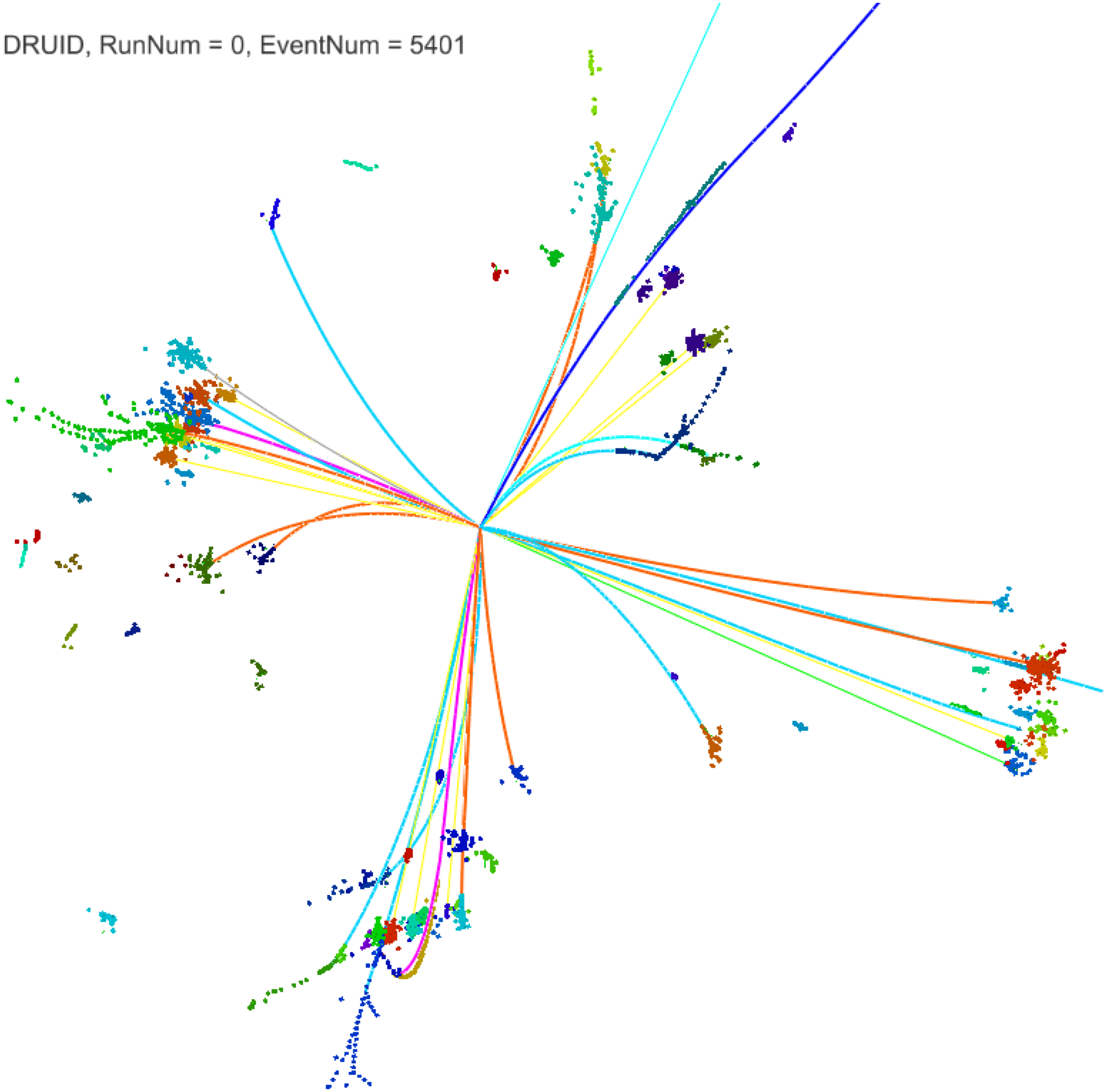
--- Inclusive analysis,

usually not all the final particles will be used



Typical events at hadron collider and lepton collider



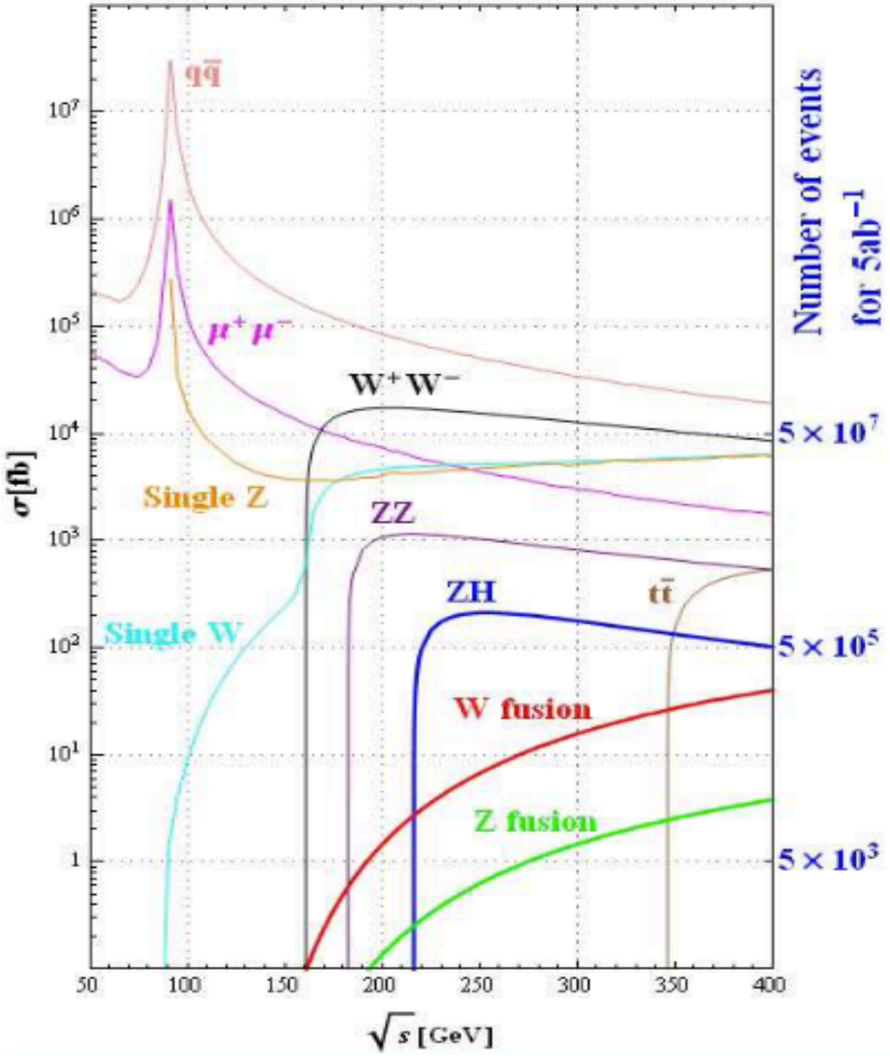


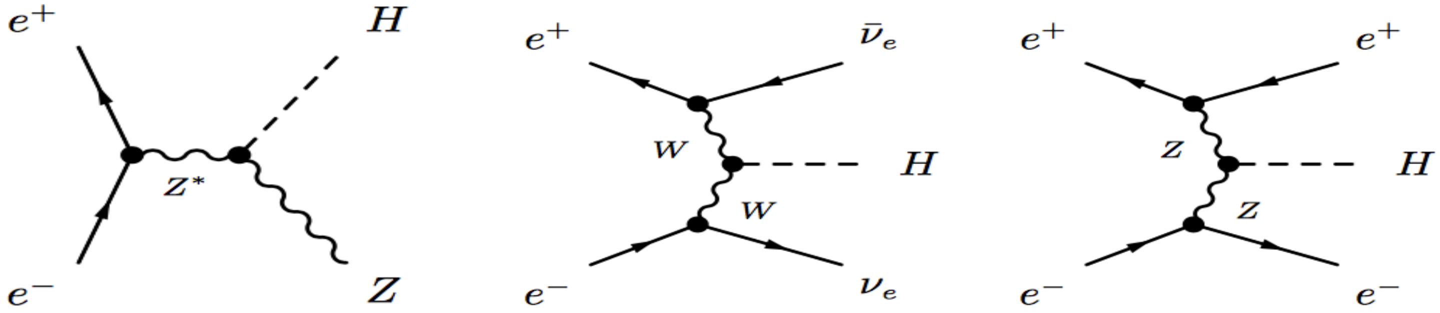
e^+e^- experiments

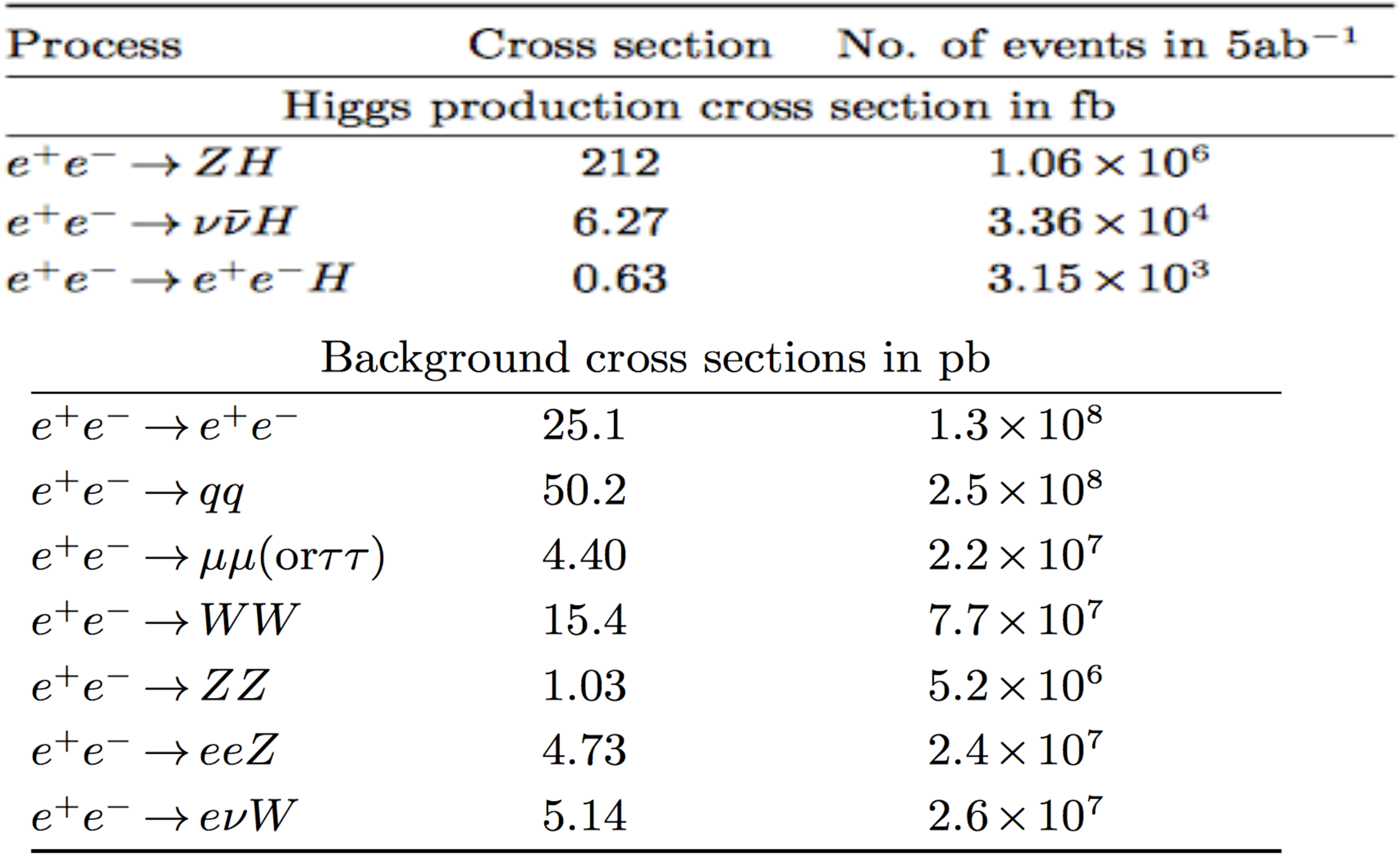
- Precisely defined initial P4,
- Much much less backgrounds,
- Pile-up free,
- Almost all final state particles recorded

--- exclusive analysis

All signal/backgrounds generated by Whizard1.95 in stdhep(5/ab)







Process	Cross section	No. of events in $5ab^{-1}$
Higgs production cross section in fb		
$e^+e^- \to ZH$	212	1.06×10^6
$e^+e^- \to \nu\bar{\nu}H$	6.27	3.36×10^4
$e^+e^- \to e^+e^-H$	0.63	3.15×10^3
Background cross sections in pb		
$e^+e^- \to e^+e^-$	25.1	1.3×10^8
$e^+e^- \to qq$	50.2	2.5×10^8
$e^+e^- \to \mu\mu(\text{or}\tau\tau)$	4.40	2.2×10^7
$e^+e^- \to WW$	15.4	7.7×10^7
$e^+e^- \to ZZ$	1.03	5.2×10^6
$e^+e^- \to eeZ$	4.73	2.4×10^7
$e^+e^- \to e\nu W$	5.14	2.6×10^7

Configuration of CEPC fast simulation

Object	**Resolution (barrel)**	**Resolution (end-cap)**	**Efficiency(|η|<3%)**	**Efficiency (|η|>3，%)**
Electron	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	100	0
Muon	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	100	0
Charged hadrons	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	$\sqrt{(0.001^2+pt^2\times2\times10^{-5})}$	100	0
Neutral hadrons	$\sqrt{(E^2\times0.01^2+E^2\times0.6^2)}$		100	0
Photons	$\sqrt{(E^2\times0.005^2+E^2\times0.2^2)}$		100	0
Jets(PFA)	3-4%		~100%	

P or E cut on particles affects jet energy scale

More details and validations can be found Cheng’s talk in CEPC workshop at Wuhan :
http://indico.ihep.ac.cn/event/6433/session/19/contribution/91/material/slides/1.pdf

Jet-clustering at e^+e^- collider

- Fastjet package used for jet-clustering in Delphes
- Only inclusive modes implemented：such as kt, anti-kt, and etc. reconstructing the jet candidates according to the user’s requirement
- But e^+e^- experiment favors exclusive jet-clustering: clustering the input particles into a fixed number of jets as user requested, usually the input particle collection not including the isolated leptons and photons.
- It was put into Delphes by us, and will appear in the new official release

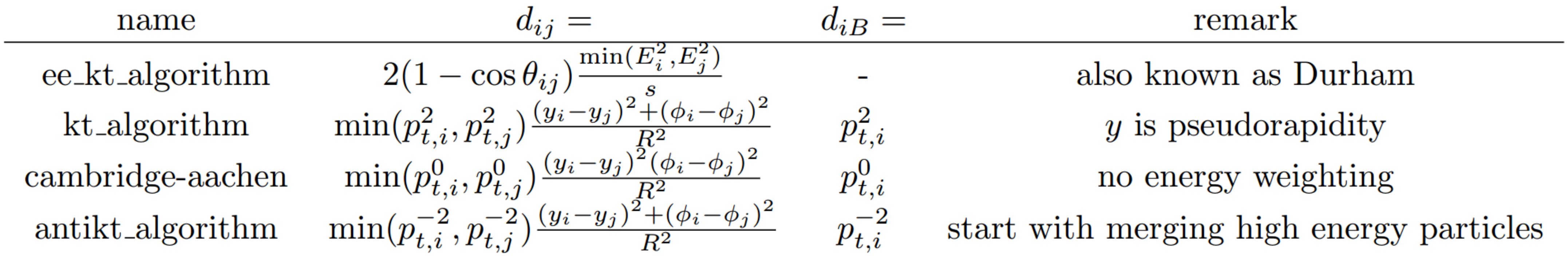
Jet clustering algorithm at hadron collider

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

for CEPC, beam jets negligible

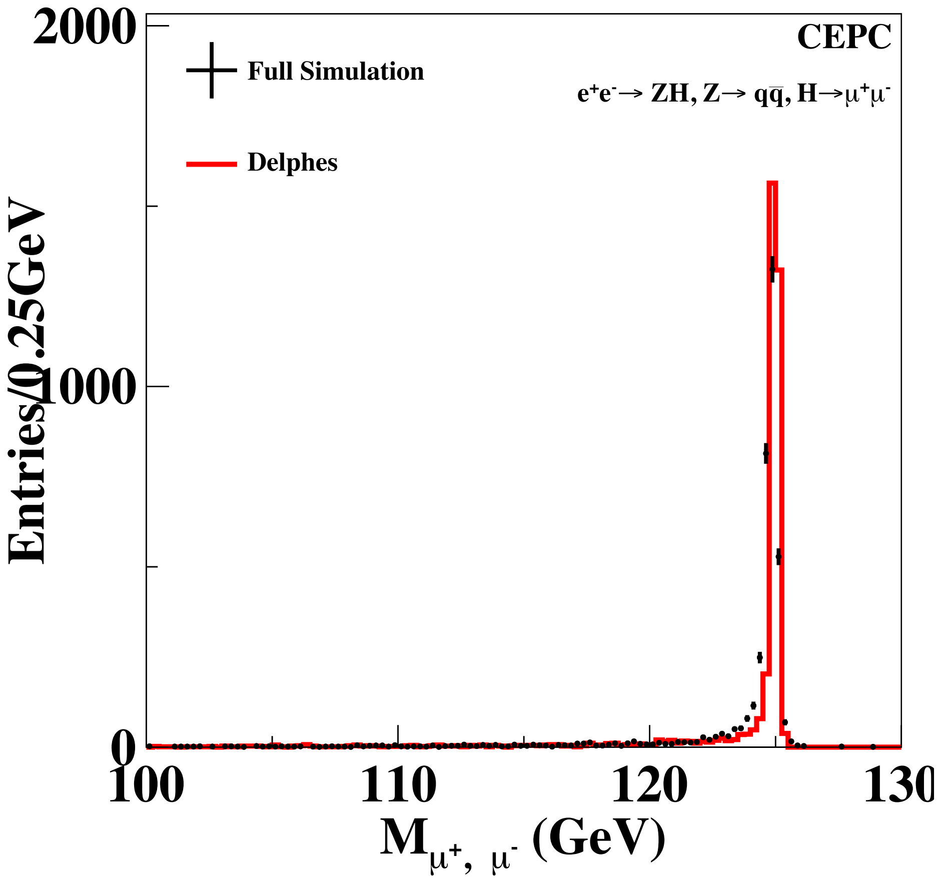
ee_kt_algorithm

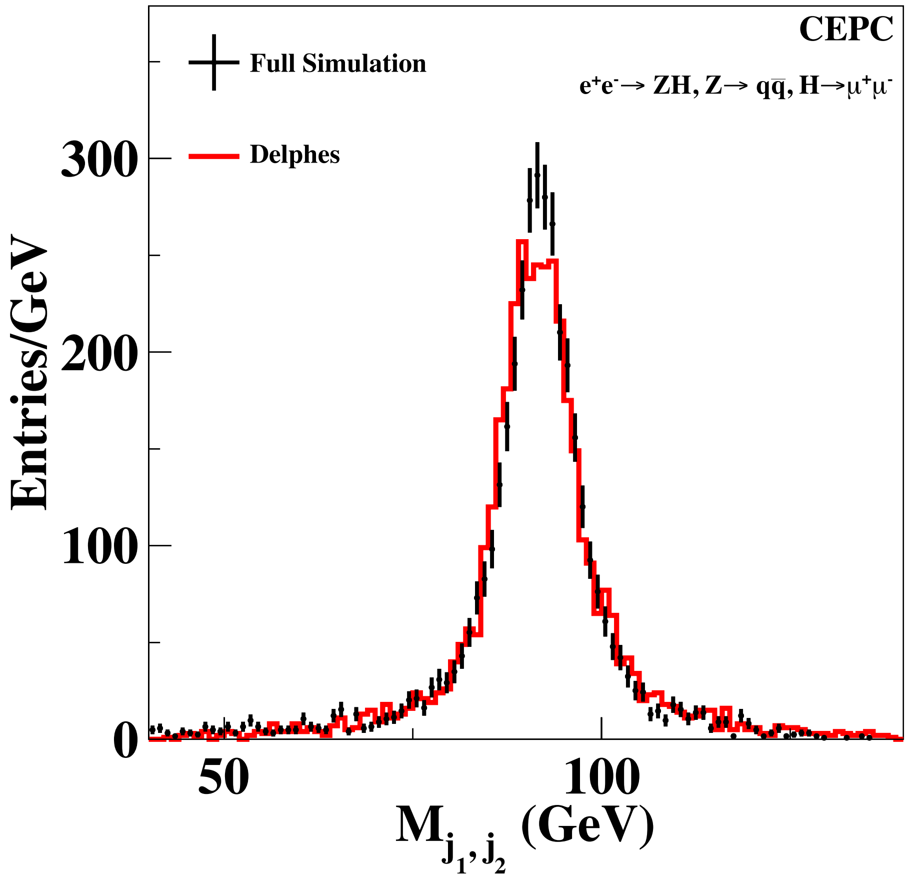
S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, Phys. Lett. B 269, 432 (1991)



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

Delphes vs. CEPC Full simulation Tracking and jet energy resolution

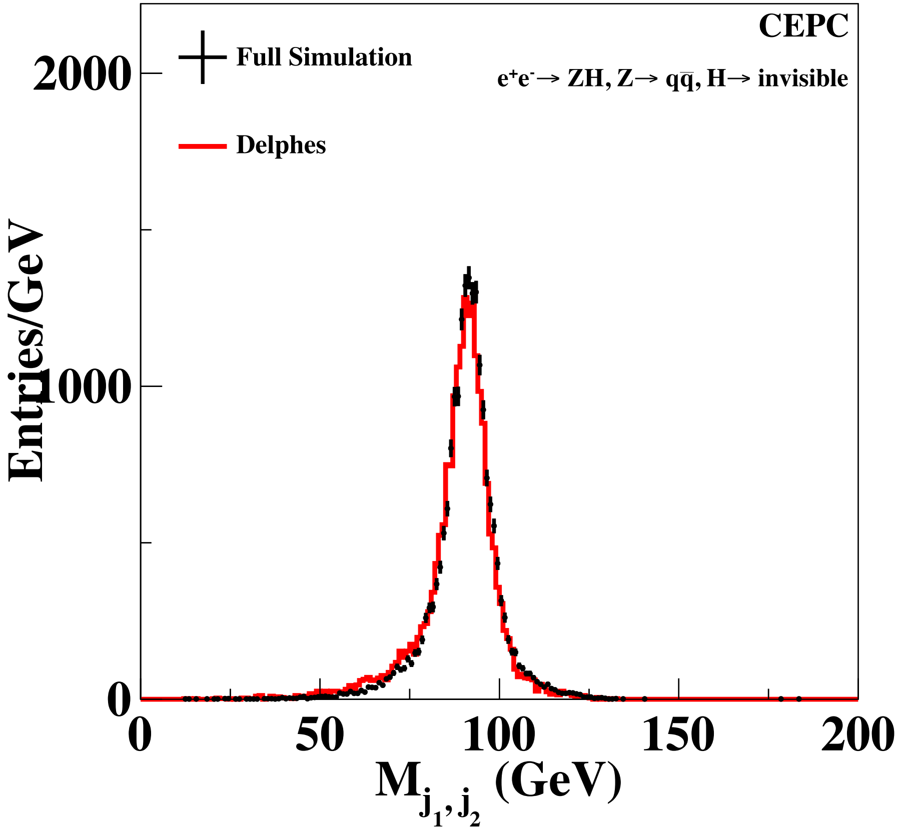


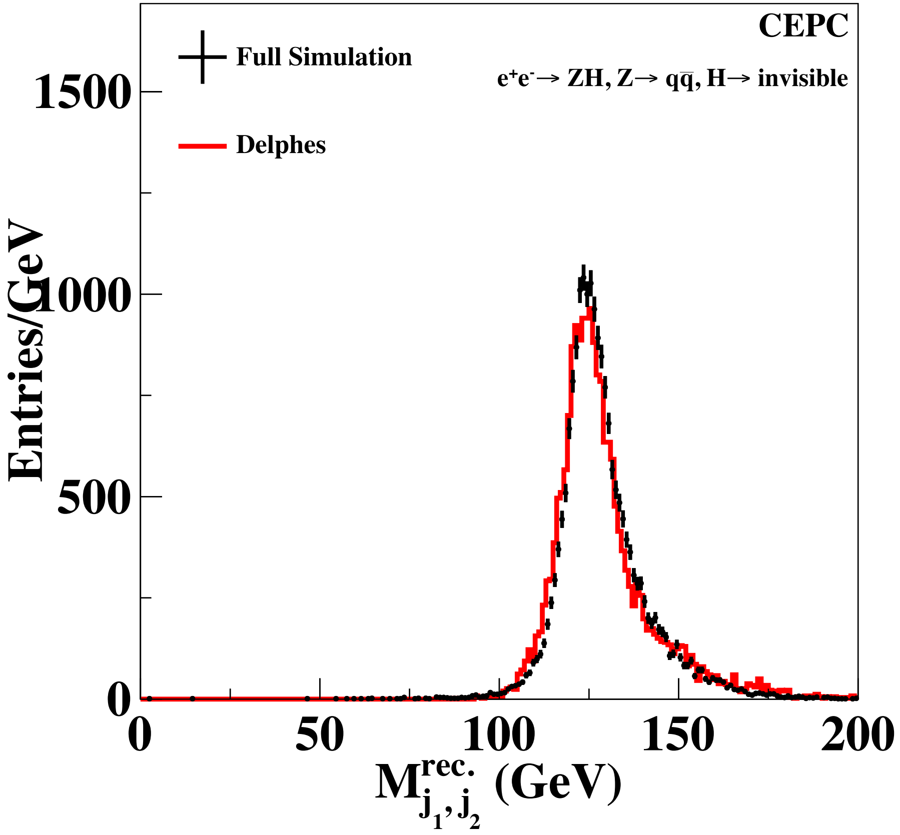


Jet-clustering algorithm:

$e^+e^-\ k_t$ algorithm

Delphes vs. CEPC Full simulation jet energy resolution

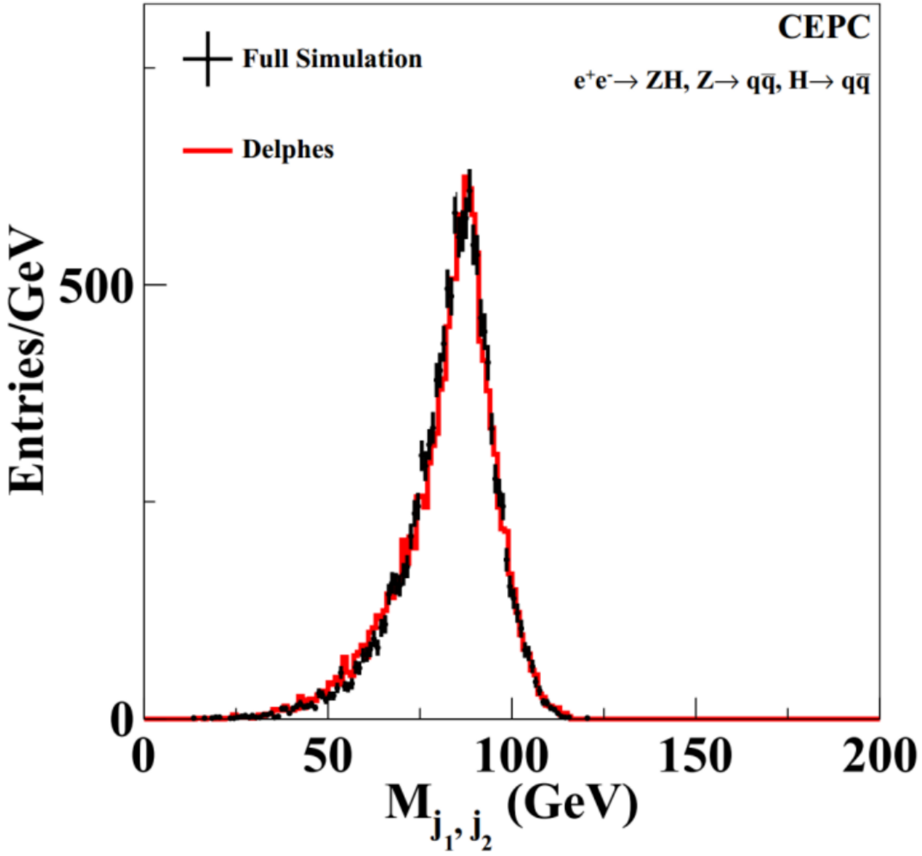


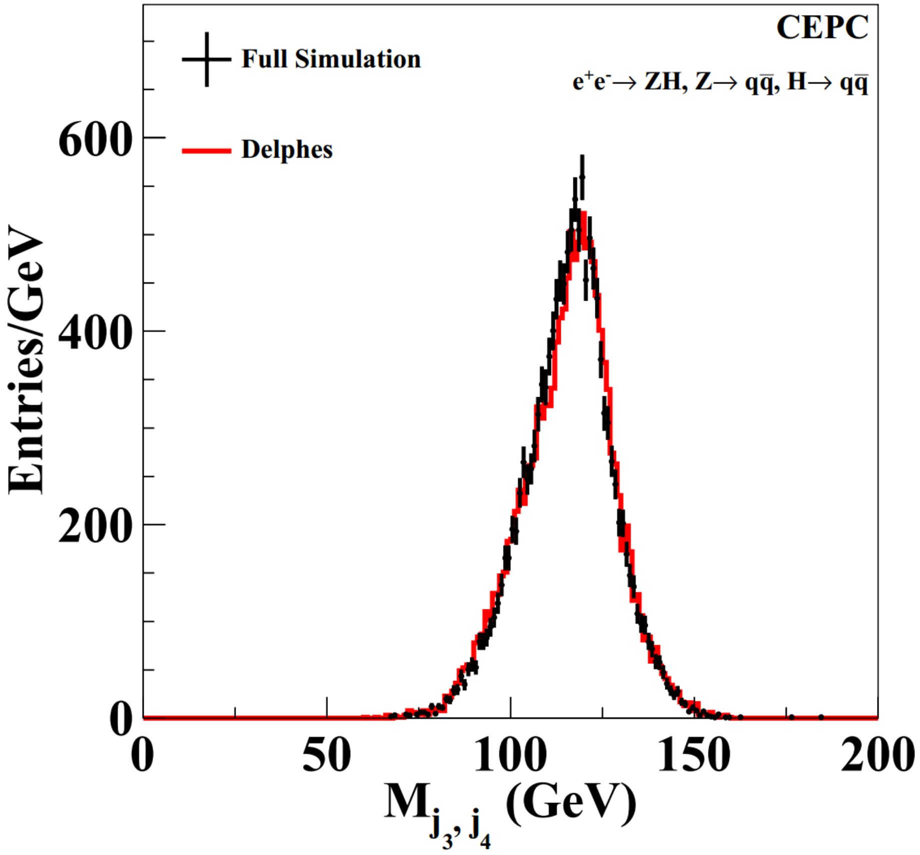


Jet-clustering algorithm:

e^+e^- k_t algorithm

Delphes vs. CEPC Full simulation

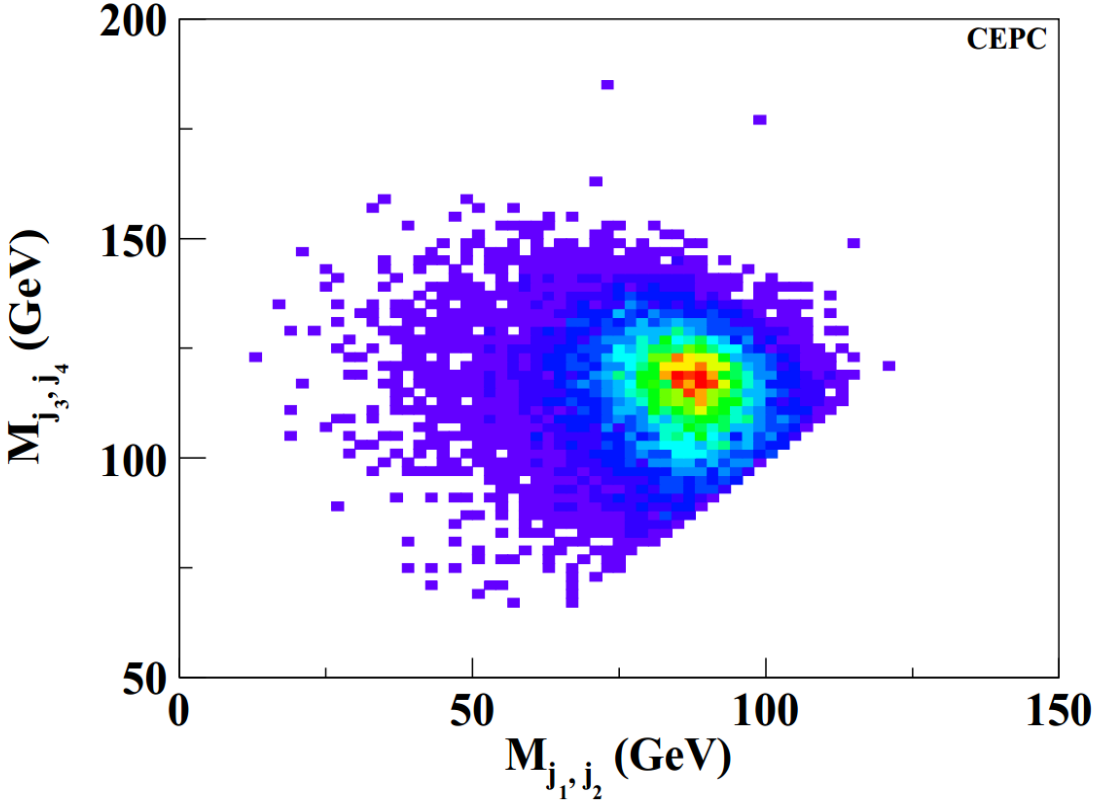


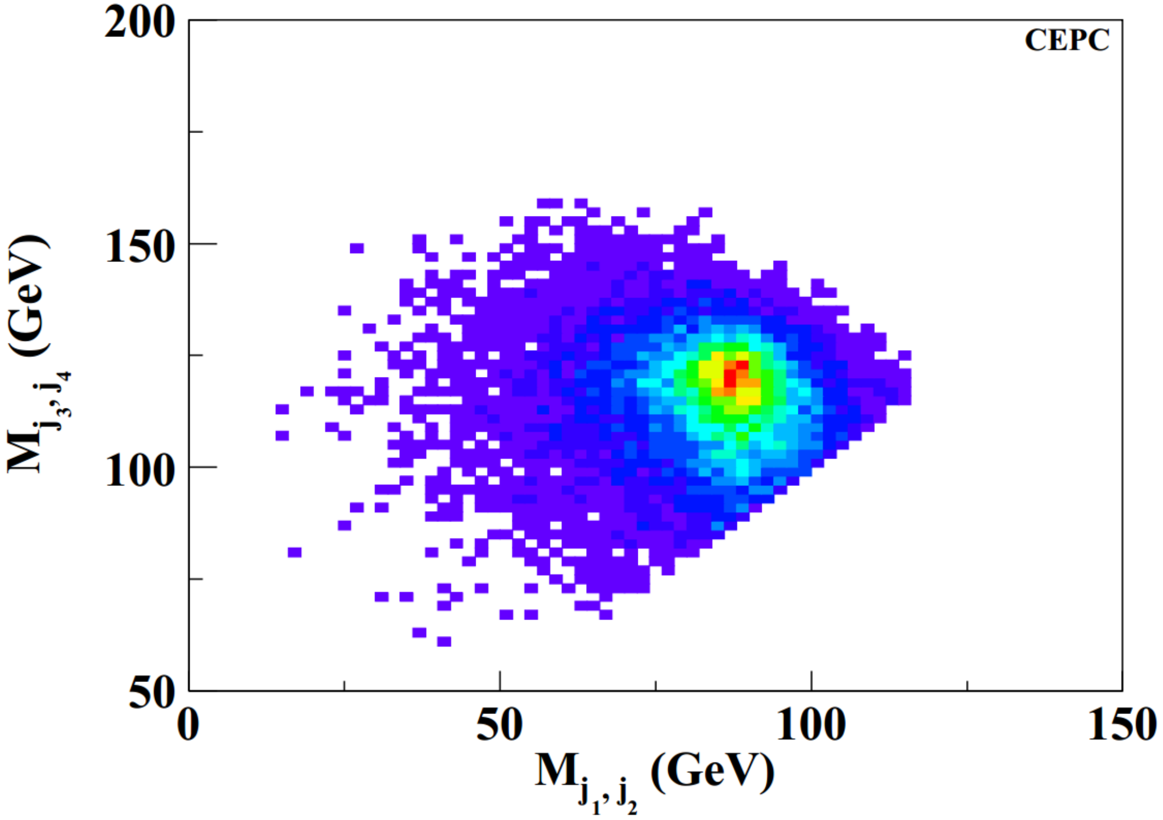


Jet-clustering algorithm:

e^+e^- k_t algorithm

Delphes vs. Full simulation
Jet pairing





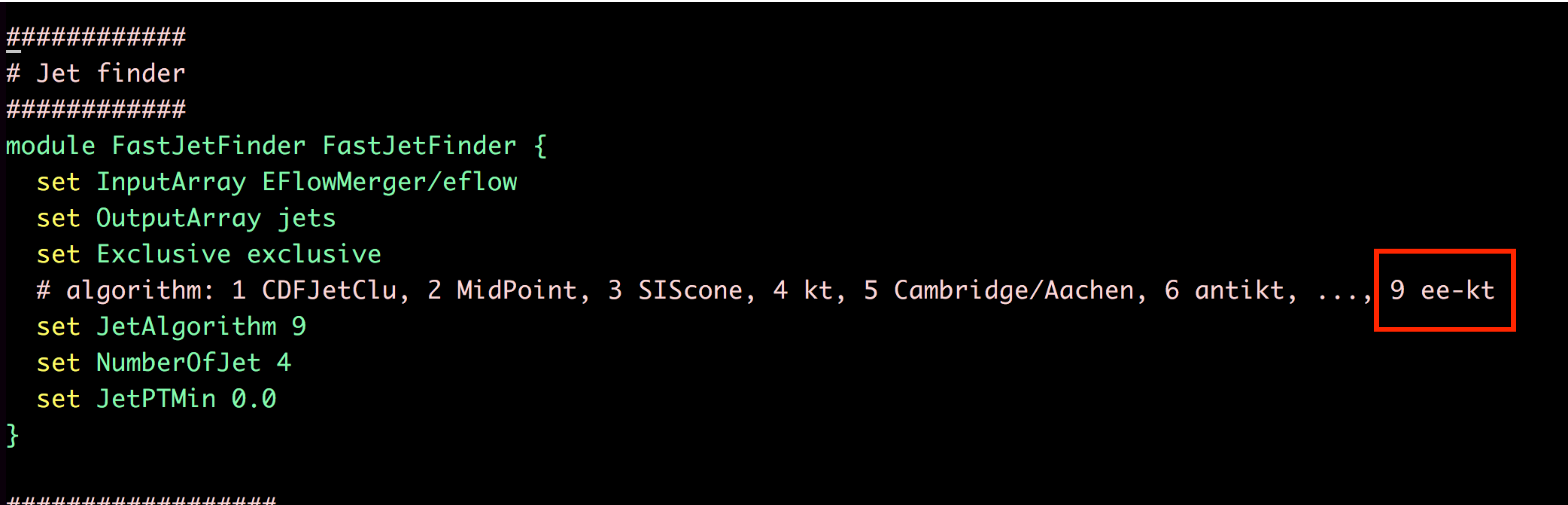
$e^+e^-\rightarrow ZH, Z\rightarrow q\bar{q}, H\rightarrow q\bar{q}$

full simulation

fast simulation

Example 1: e^+e^- → qqH, H→qq

- All final particle are from 4 jets, if neglect ISR photons
- Feed all final objects to ee-kt, force them into 4-jets w/o parameter
- Loose Pt cut
- Mass pairing with 4-jets, take the best pairing scheme which fits Z/H mass
- Suppress background with *yij* parameters, which are from the jet-clustering algorithm



```
############
# Jet finder
############
module FastJetFinder FastJetFinder {
  set InputArray EFlowMerger/eflow
  set OutputArray jets
  set Exclusive exclusive
  # algorithm: 1 CDFJetClu, 2 MidPoint, 3 SIScone, 4 kt, 5 Cambridge/Aachen, 6 antikt, ..., 9 ee-kt
  set JetAlgorithm 9
  set NumberOfJet 4
  set JetPTMin 0.0
}

##################
```

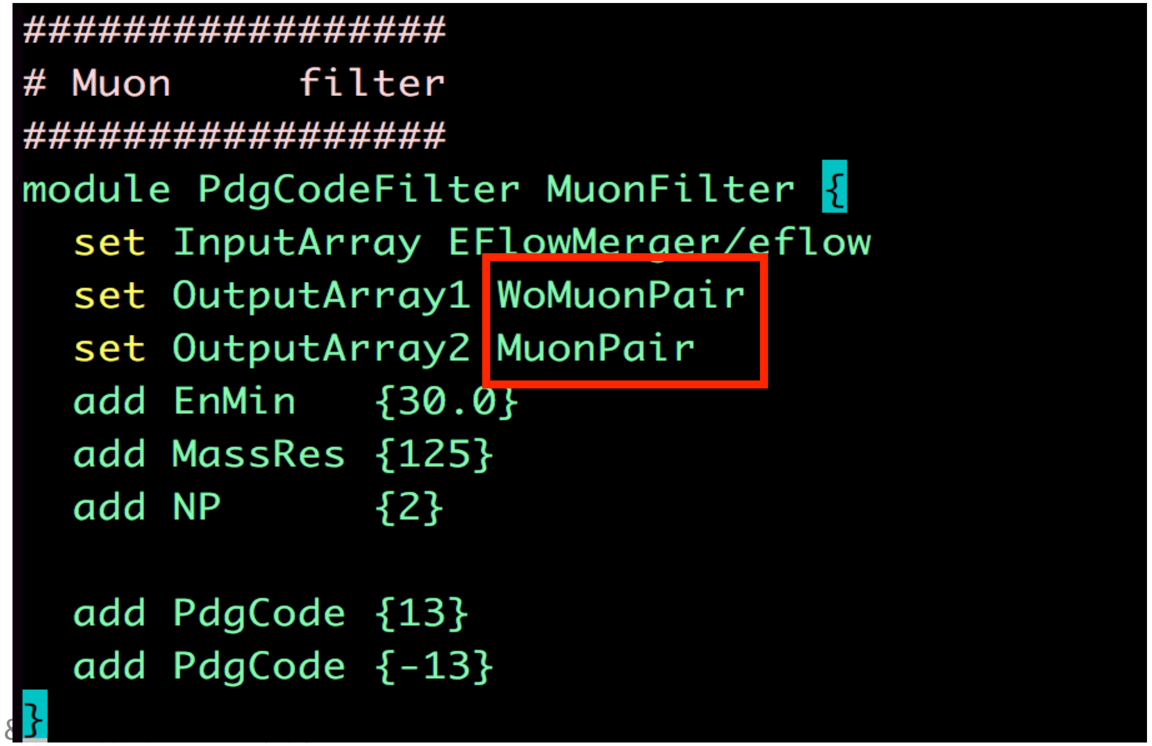
Example 2: **e^+e^- →ZH, Z→qq, H→$\mu^+\mu^-$**

✓First Step： Selecting the muon pair first

- muon: E>30GeV and isolated (not very necessary)
- Take the one with M($\mu^+\mu^-$) ~125GeV when multi-entry occurs

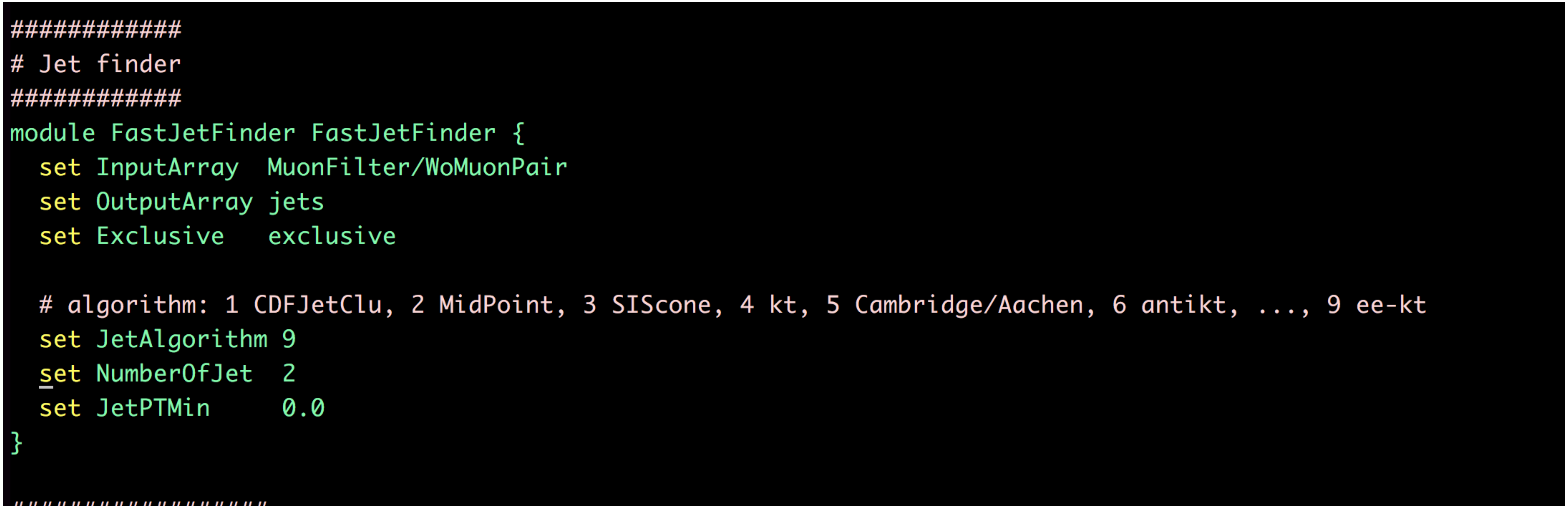
✓The results

- A muon pair, and
- The remain particles → jet-clustering



```
#################
# Muon     filter
#################
module PdgCodeFilter MuonFilter {
  set InputArray EFlowMerger/eflow
  set OutputArray1 WoMuonPair
  set OutputArray2 MuonPair
  add EnMin   {30.0}
  add MassRes {125}
  add NP      {2}

  add PdgCode {13}
  add PdgCode {-13}
}
```

Second step: jet-clustering



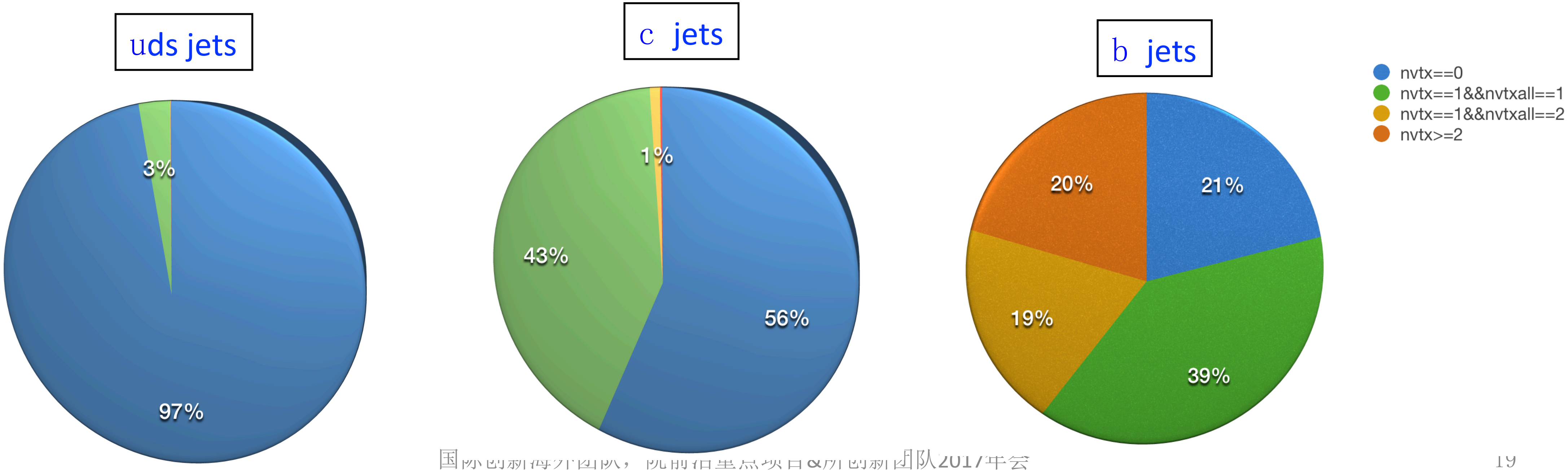
```
############
# Jet finder
############
module FastJetFinder FastJetFinder {
  set InputArray  MuonFilter/WoMuonPair
  set OutputArray jets
  set Exclusive   exclusive

  # algorithm: 1 CDFJetClu, 2 MidPoint, 3 SIScone, 4 kt, 5 Cambridge/Aachen, 6 antikt, ..., 9 ee-kt
  set JetAlgorithm 9
  set NumberOfJet  2
  set JetPTMin     0.0
}
```

Force the remain particle into two output jets with ee-kt in exclusive mode.

You got them: two muons and two jets, no duplication

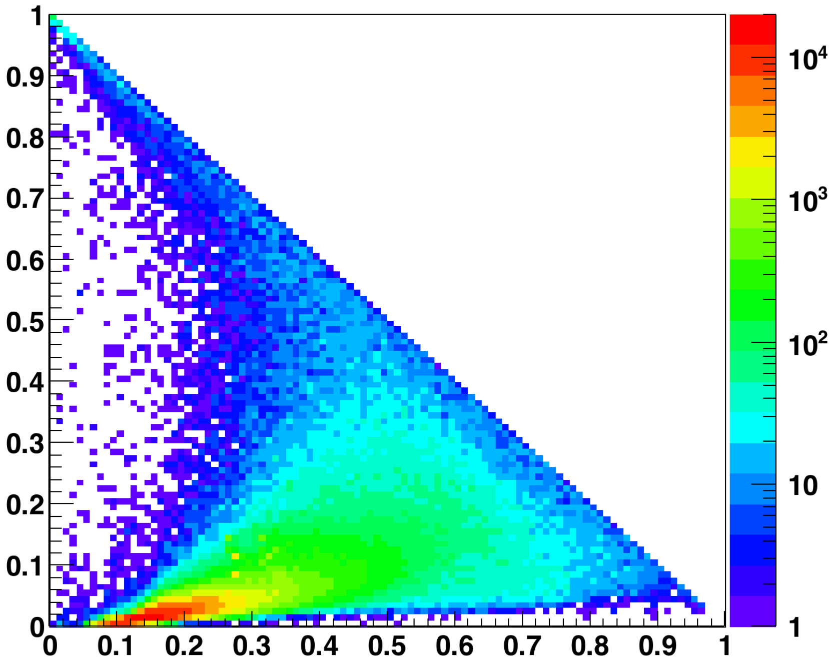
Flavor tagging in fast simulation

- Delphes only attaches two integers to each jets according to efficiency and fake rate from user
- In full simulation and real experiments, flavor tagging benefits from precise vertex measurements and PFA philosophy
- CEPC uses machine learning (BDT) method to provide two numbers for each jets: b-likeness and c-likeness

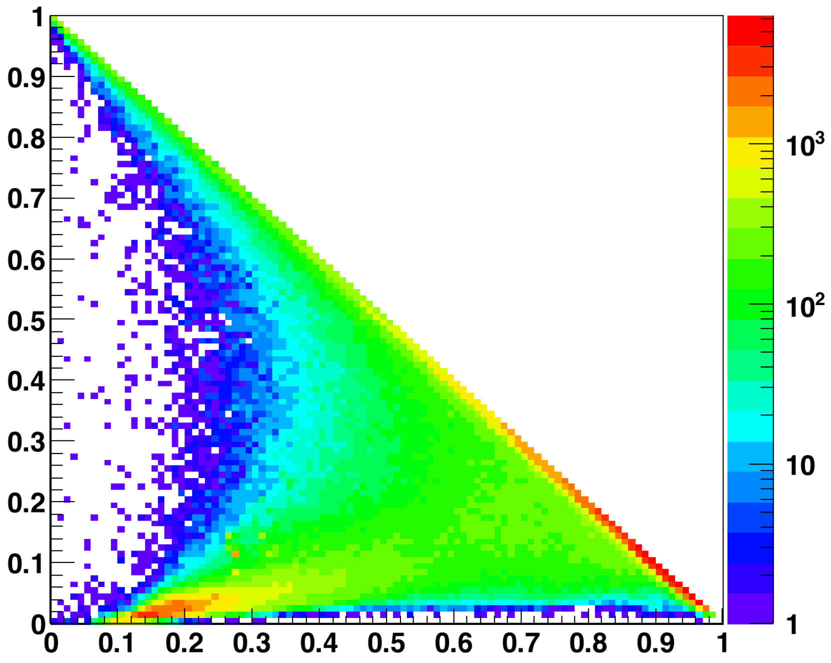


b-likeness vs. c-likeness from full simulation

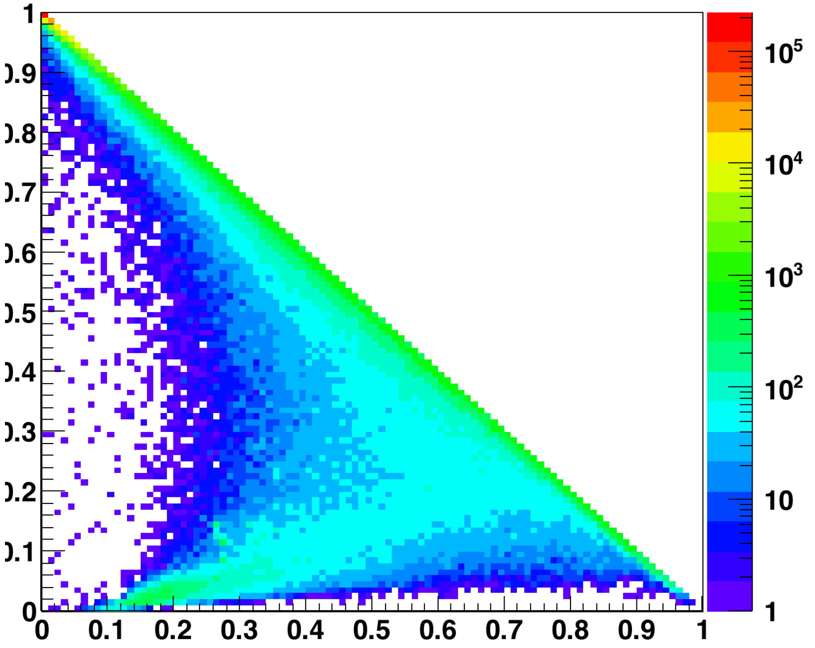
uds jets



c jets

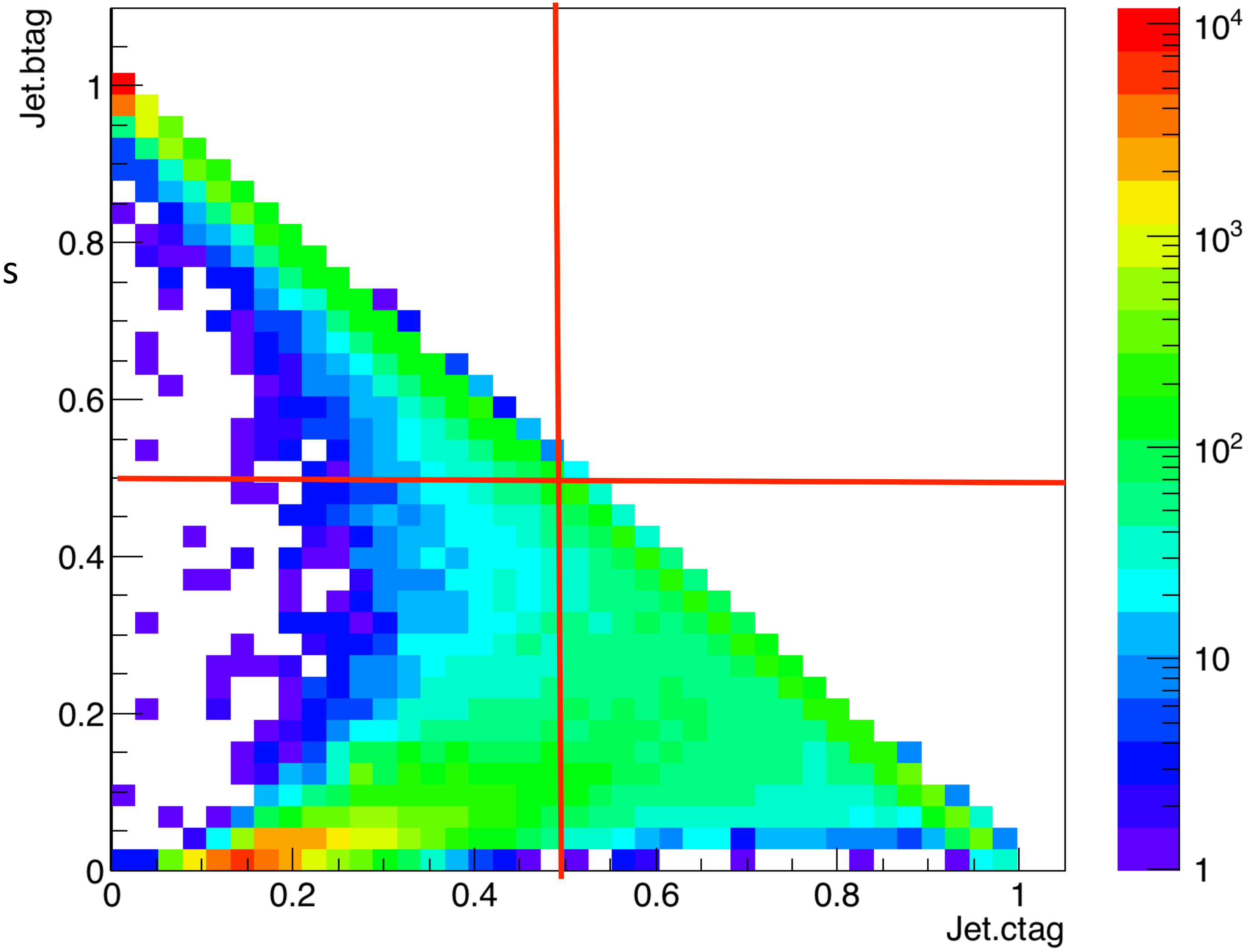


b jets



Fast simulation

Now you can cut on these two variables to improve S/B ratio



Summary

- ✧ CEPC fast simulation with Delphes available after some necessary coding work
- ✧ New add-ons:
  - ✓ Jet-clustering with ee-kt and exclusive jets
  - ✓ Realized exclusive analysis method using Delphes modules
  - ✓ New flavor−tagging: each jet attached probabilities of specific flavor, same as full simulation
- ✧ It was carefully checked and validated

- ✧ CEPC Delphes card and code available: https://github.com/zeusmail/delphes
- ✧ A paper with new implementations and validations coming soon