Outline

- Jet Tasks Last Wednesday slides
- Dimuon check
- Computing

Jet tasks to do

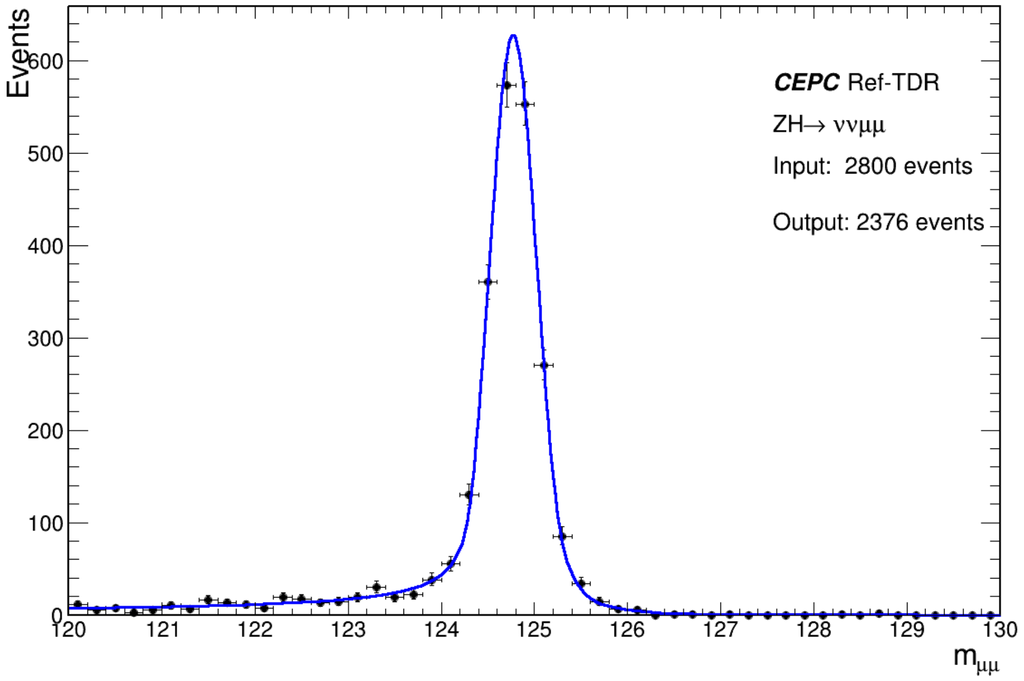
- Jet Sample Production
- Jet Gen Match
- JE related plots
- JA related plots
- BMR plots
- Neutral jet superclusters
- Particle gun one-type particle response

Tasks sub priority

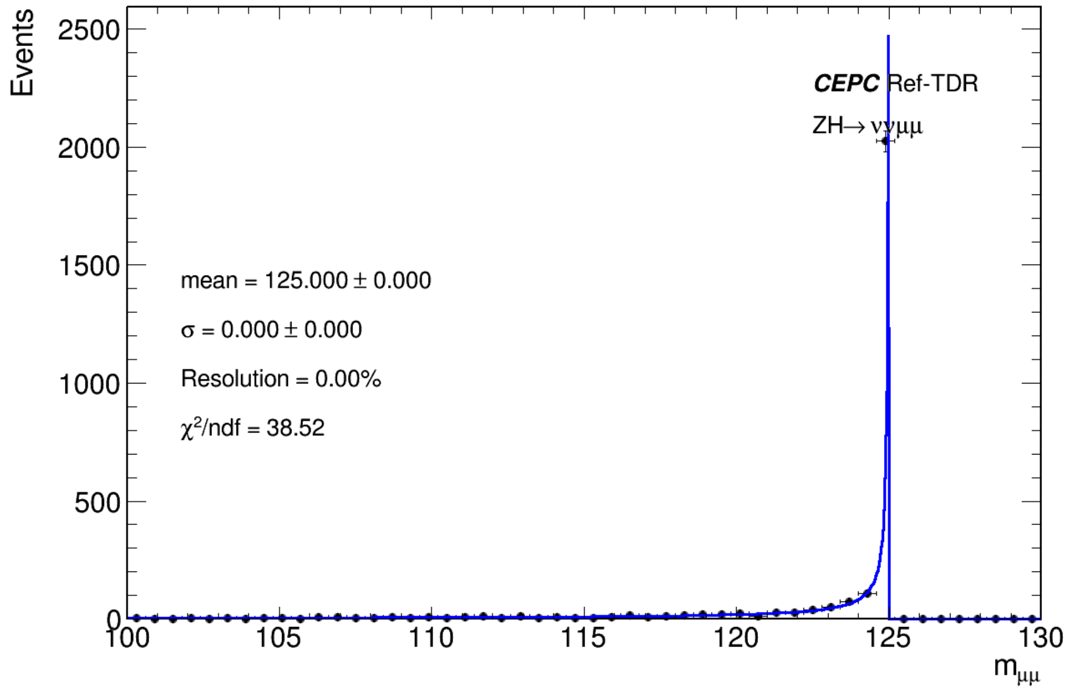
- Remove isolated lepton/photon in PFO then jet clustering.
  - Need a quick PID
- Jet Event display
- Validation ee-kt algorithm with others.
- Validation generator Whizard with others.
- Flavors/JOI
- Endcap jet performance
- Repeat Ecal/Hcal performance

dimuons

Reco



Truth

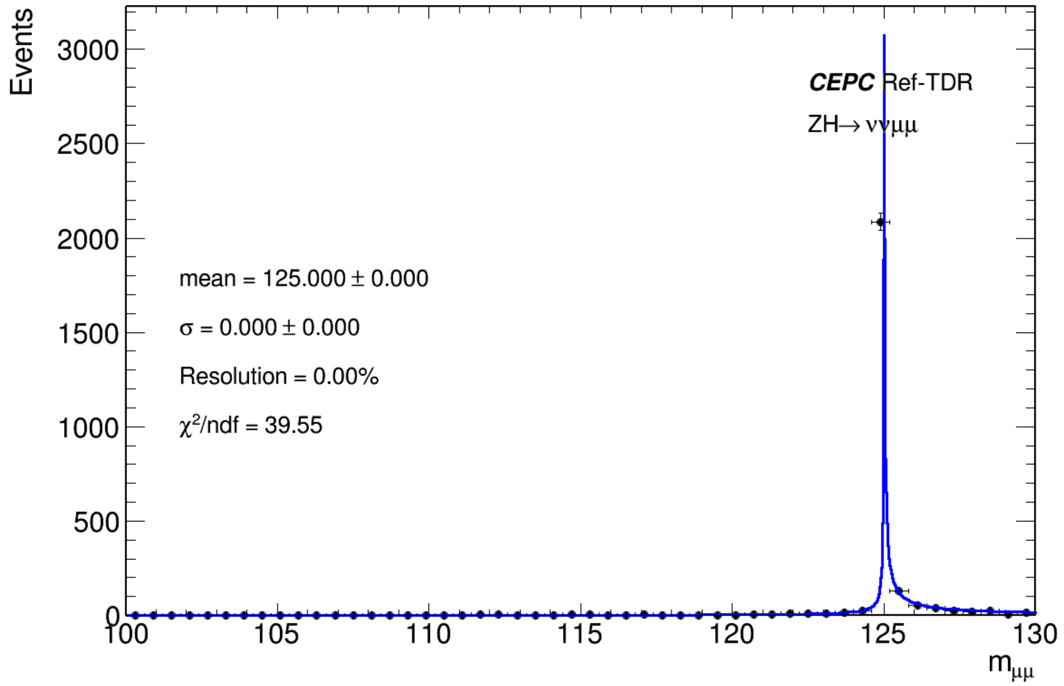


In reco, 13% are <120. In truth, 8.7% <120%.
Occasionally, FSR photons energy > 10GeV.

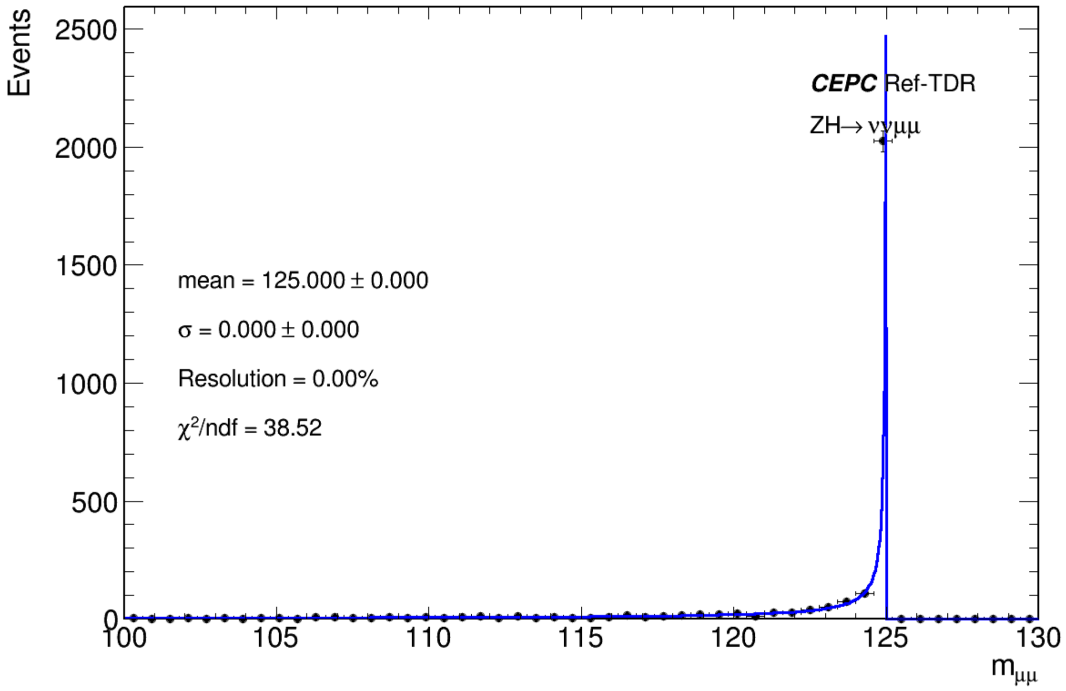
dimuons

By selecting 2 muons in status=2 (intermediate) and add photons back (in Dr=0.4 close to muon).

Truth + Photons



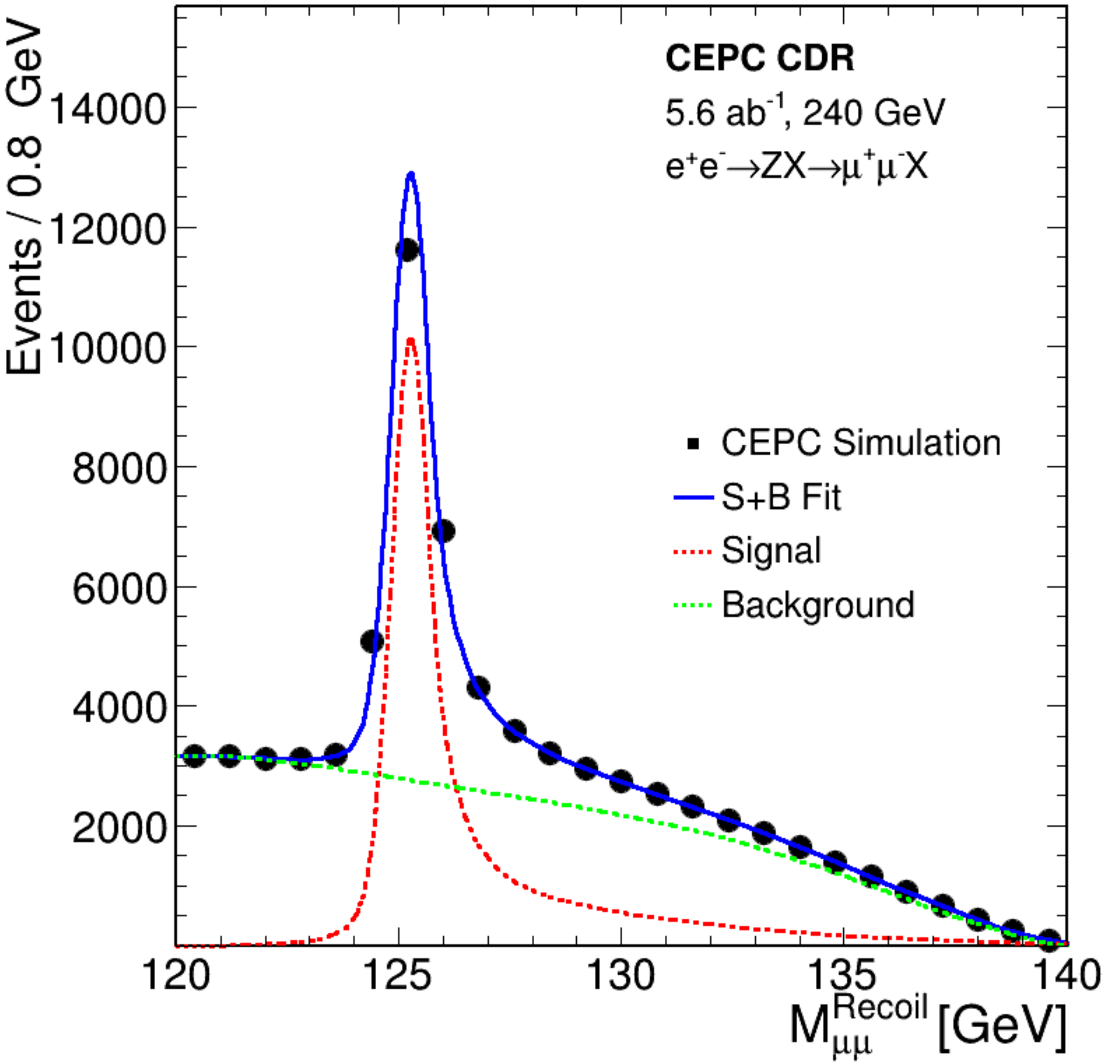
Truth

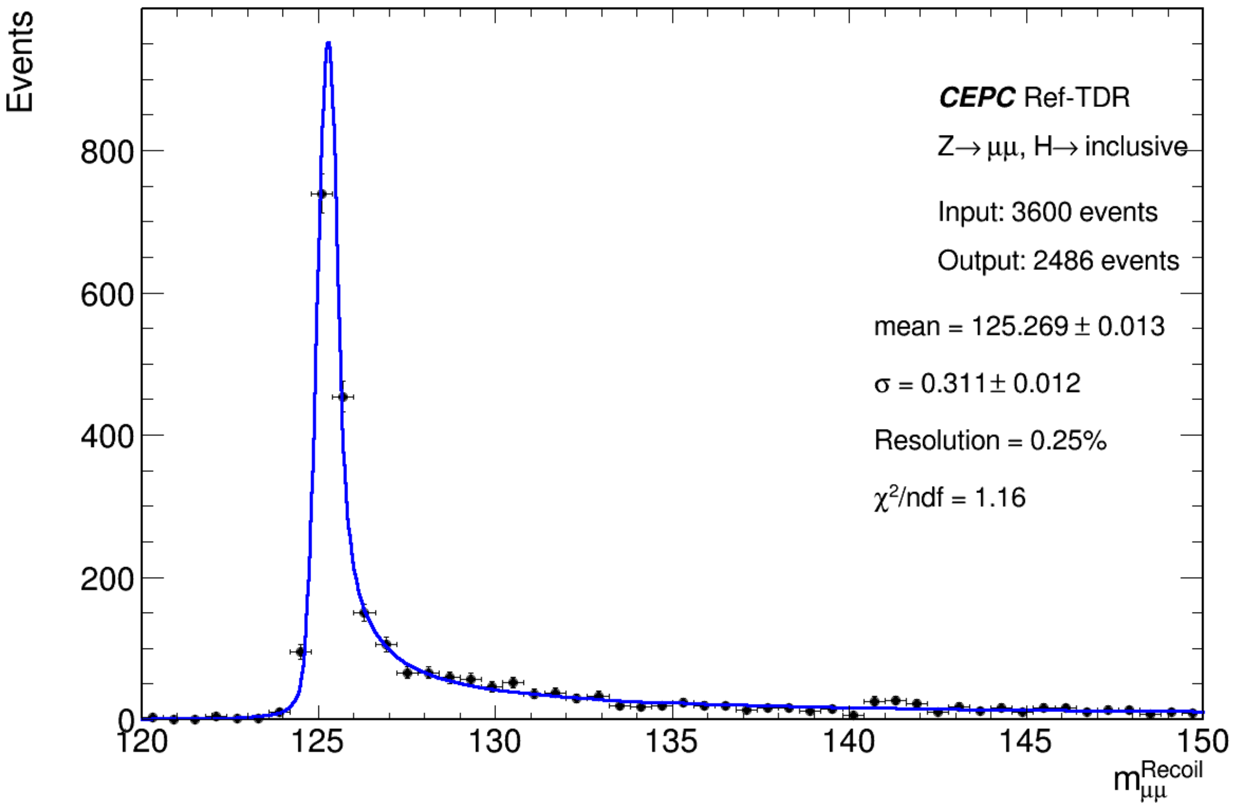


Conclusion: due to FSR photons, dimuon lose ~10% in truth. This reduce the yields but impacts on resolution are small.

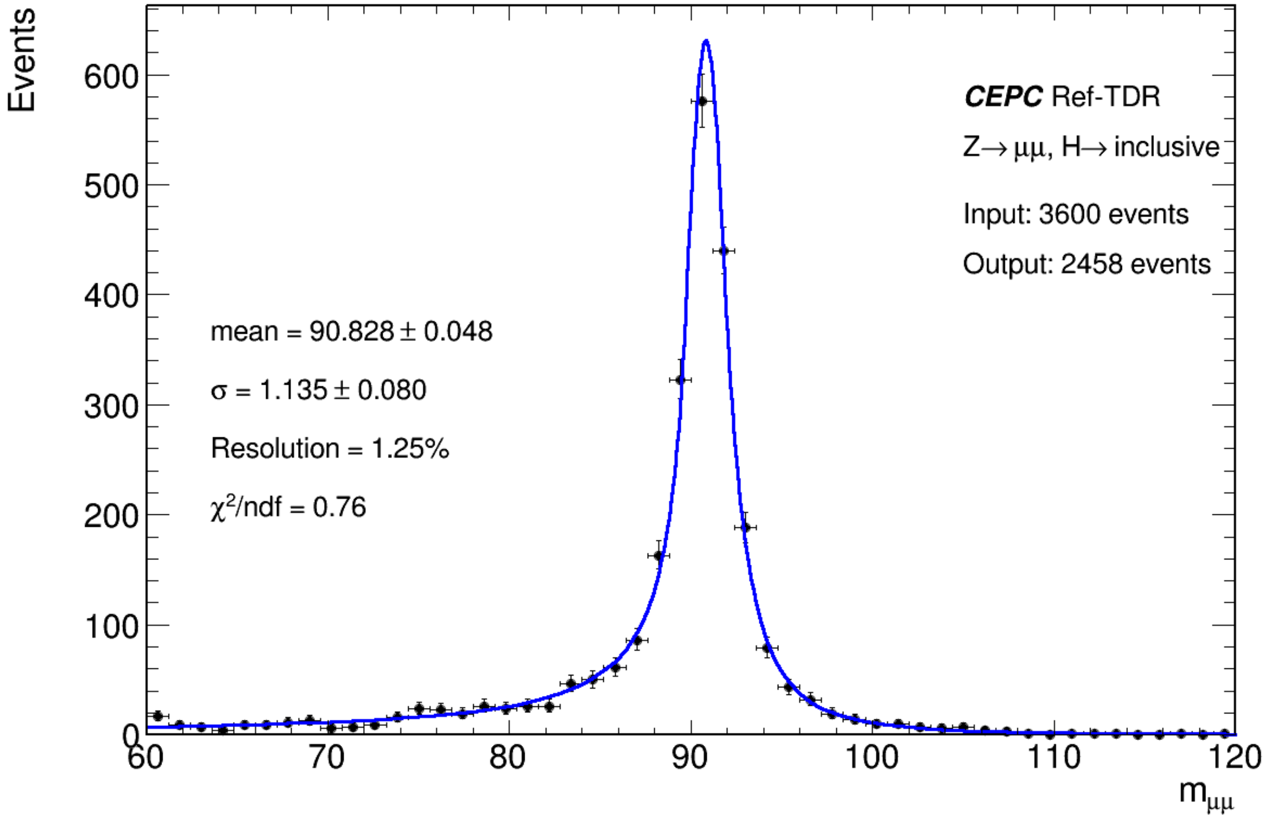
Z->mm H->inclusive

0.25% resolution on M_recoil_mm
Consistent with Gang's 0.26% results.
Better than CDR.
In this fit $\Delta m = 13\text{MeV}$.





Z->mm H->inclusive



- Without PID, currently use truth muon match to find reco muons.
- Can be done preliminarily but matching eff not so good (70%);
- Need real pid for inclusive study.

Job Time/Memory consuming

Use Z->vv H->bb 100k events for testing. In TDR24.10.0 but muon chamber removed;

- Z->vv H->bb 100events per job, can be run under 4g memory quota.
- Sim: 32.5min Digi: 15min
- Tracking: 4min Rec: 7min (+Jet clustering)
- So 100 hadronic events -> 1h. 1 event -> 36s.

- 10 events ->1000s. 1 event ->100s. The init require ~10 mins per job.

Time ratio:

- Sim: Digi: Tracking: Rec = 4:3:1:1. Or 45:31:13:11.

Job Disk quota consuming

- Z->vv H->bb 100events
- Sim: 1.4GB; 14M per event;
- Generator+Digi+Trk+Rec+Jet+…… 100M.
- Total 1.5GB; 15M per event.

Job CPU/Quene consuming

- In the same time, my account ~200 CPU can be used.
- Not all computers can use 4G or even larger memory.

- May meet problems when big sample production.

Duplicated tracks

```
 CyberPFO.energy = 76.903656, 76.838547, 26.500
78323, 1.213580, 0.989452
 CyberPFO.momentum.x = -17.793270, -17.776035,
88513, -0.416533, -0.157947, 0.361082
 CyberPFO.momentum.y = 18.909502, 18.893721, 0.
7, 0.922086, 0.054292, 0.508451
 CyberPFO.momentum.z = -72.387863, -72.327049,
.077820, 1.202032, 0.768187
 CyberPFO.referencePoint.x = 0.000000, 0.000000
0.000000, 0.000000, 0.000000
 CyberPFO.referencePoint.y = 0.000000, 0.000000
0.000000, 0.000000, 0.000000
 CyberPFO.referencePoint.z = 0.000000, 0.000000
0.000000, 0.000000, 0.000000
 CyberPFO.charge = -1.000000, -1.000000, 1.0000
```

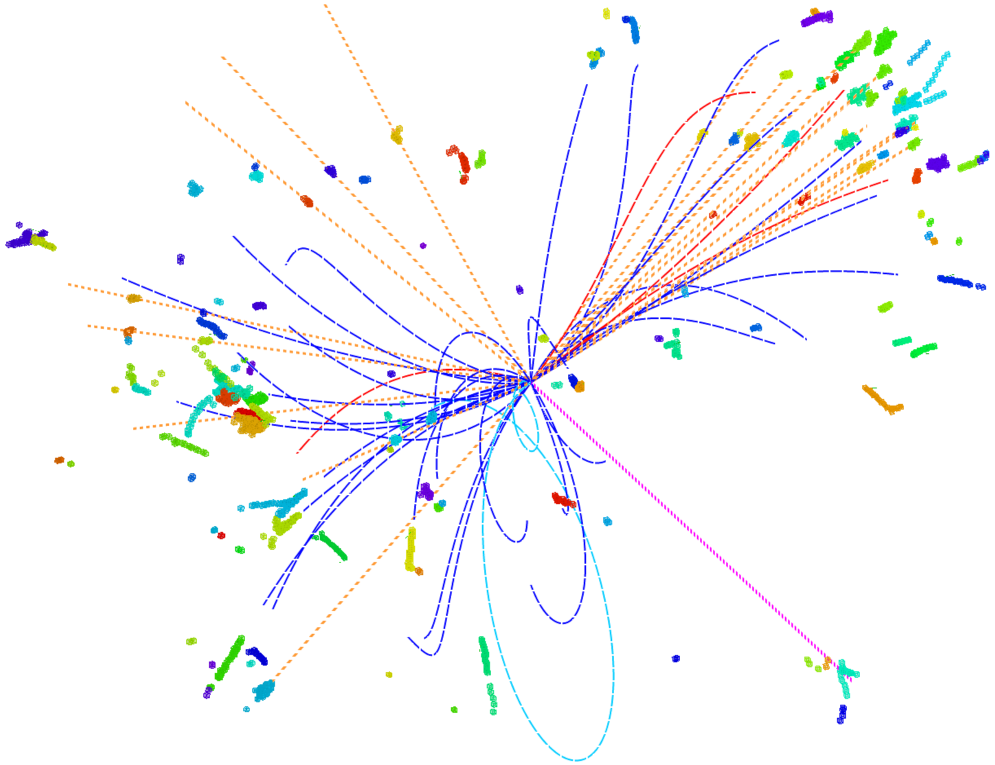
18	-17.77603	18.893722	-72.32704	76.838546	71
19	-0.036795	-0.326339	0.3206066	0.4589554	41
20	0.0086232	0.3858807	-0.115507	0.4028900	240
21	0.4124406	-0.192677	-0.173640	0.4872195	38
22	-0.121198	-0.844869	0.1654589	0.8694078	228
23	0.3610824	0.5084511	0.7681869	0.9894514	170
24	-17.79327	18.909502	-72.38786	76.903656	11

- In Z->mm H->inclusive
- Found 2 PFOs with very close PxPyPzE.
  - Difference in 0.1GeV. Not realistic.
- The track indicates that there are one track but splitting to 2. -> Got 2 PFOs.
- 0.7% in dimuon.
- More often in low energy cases.

- Should only pick 1.
  - Duplication removal in Reco level.
- @Fangyi and track experts.

Backup

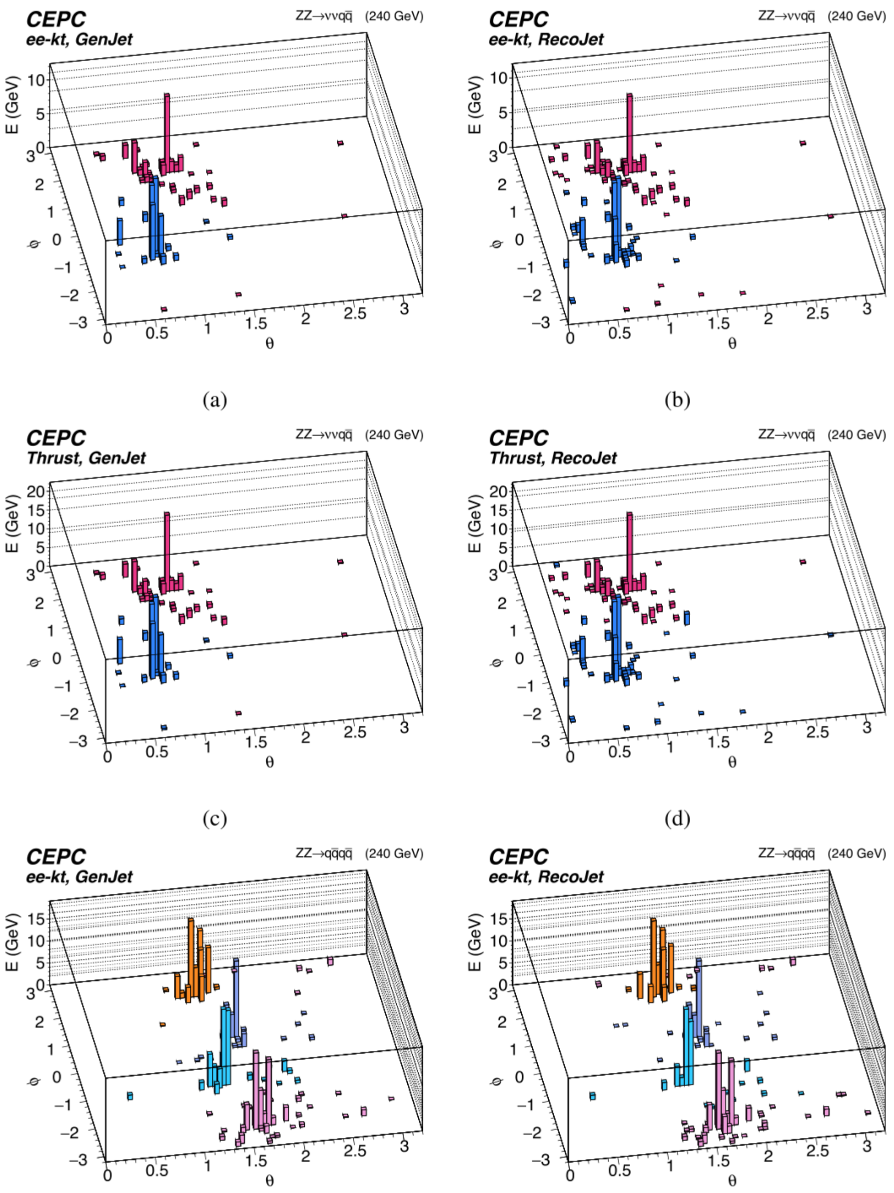
Jets 喷注



- Including varied components
- CEPC uses FastJet package to do jet clustering.
- Now, ee-kt/Durham algorithm used.
  - You need and only need to specify N_jets for Fastjet.
  - Generally, for all kt algos, 2 parameter: R and P can be adjusted.
  - ee-kt no R setting, so all clusters will be clustered.

$$d_{ij} = \min(E_i^{2p}, E_j^{2p}) \frac{(1 - \cos\theta_{ij})}{(1 - \cos R)},$$
$$d_{iB} = E_i^{2p},$$

Jet event display



(a) (b)

(c) (d)

(e) (f)

Jet @ CEPCSW

```
JetDefinition jet_def(ee_kt_algorithm);

ClusterSequence clust_seq(input_particles, jet_def);

vector<PseudoJet> jets = sorted_by_pt(clust_seq.exclusive_jets(nJets));
```

- Specify PFO container and njets.
- Not stored in final ntuple.
- In Zebing’s Genmatch
  - Many variable stored.
  - Find jet_ntuple to extract informations

```
StatusCode GenMatch::initialize(){

  // Create a new TTree
  _file = TFile::Open(m_outputFile.value().c_str(), "RECREATE");
  _tree = new TTree("jets", "jets");
  _tree->Branch("jet1_px", &jet1_px, "jet1_px/D");
  _tree->Branch("jet1_py", &jet1_py, "jet1_py/D");
  _tree->Branch("jet1_pz", &jet1_pz, "jet1_pz/D");
  _tree->Branch("jet1_E", &jet1_E, "jet1_E/D");
  _tree->Branch("jet1_costheta", &jet1_costheta, "jet1_costheta/D");
  _tree->Branch("jet1_phi", &jet1_phi, "jet1_phi/D");
  _tree->Branch("jet1_pt", &jet1_pt, "jet1_pt/D");
  _tree->Branch("jet1_nconstituents", &jet1_nconstituents, "jet1_nconstituents/I");
  _tree->Branch("jet2_px", &jet2_px, "jet2_px/D");
  _tree->Branch("jet2_py", &jet2_py, "jet2_py/D");
  _tree->Branch("jet2_pz", &jet2_pz, "jet2_pz/D");
  _tree->Branch("jet2_E", &jet2_E, "jet2_E/D");
  _tree->Branch("jet2_costheta", &jet2_costheta, "jet2_costheta/D");
  _tree->Branch("jet2_phi", &jet2_phi, "jet2_phi/D");
  _tree->Branch("jet2_pt", &jet2_pt, "jet2_pt/D");
  _tree->Branch("jet2_nconstituents", &jet2_nconstituents, "jet2_nconstituents/I");
  _tree->Branch("constituents_E1tot", &constituents_E1tot, "constituents_E1tot/D");
  _tree->Branch("constituents_E2tot", &constituents_E2tot, "constituents_E2tot/D");
  _tree->Branch("mass", &mass, "mass/D");
  _tree->Branch("ymerge", ymerge, "ymerge[6]/D");
  _tree->Branch("nparticles", &nparticles, "nparticles/I");
  _tree->Branch("jet1_GENMatch_id", &jet1_GENMatch_id, "jet1_GENMatch_id/I");
  _tree->Branch("jet2_GENMatch_id", &jet2_GENMatch_id, "jet2_GENMatch_id/I");
  _tree->Branch("jet1_GENMatch_mindR", &jet1_GENMatch_mindR, "jet1_GENMatch_mindR/D");
  _tree->Branch("jet2_GENMatch_mindR", &jet2_GENMatch_mindR, "jet2_GENMatch_mindR/D");

  _tree->Branch("PFO_Energy_muon", &PFO_Energy_muon);
  _tree->Branch("PFO_Energy_muon_GENMatch_dR", &PFO_Energy_muon_GENMatch_dR);
  _tree->Branch("PFO_Energy_muon_GENMatch_ID", &PFO_Energy_muon_GENMatch_ID);
  _tree->Branch("PFO_Energy_muon_GENMatch_E", &PFO_Energy_muon_GENMatch_E);
  _tree->Branch("PFO_Energy_Charge", &PFO_Energy_Charge);
  _tree->Branch("PFO_Energy_Charge_Ecal", &PFO_Energy_Charge_Ecal);
  _tree->Branch("PFO_Energy_Charge_Hcal", &PFO_Energy_Charge_Hcal);
  _tree->Branch("PFO_Energy_Charge_GENMatch_dR", &PFO_Energy_Charge_GENMatch_dR);
  _tree->Branch("PFO_Energy_Charge_GENMatch_ID", &PFO_Energy_Charge_GENMatch_ID);
  _tree->Branch("PFO_Energy_Charge_GENMatch_E", &PFO_Energy_Charge_GENMatch_E);
  _tree->Branch("PFO_Hits_Charge_E", &PFO_Hits_Charge_E);
  _tree->Branch("PFO_Hits_Charge_R", &PFO_Hits_Charge_R);
  _tree->Branch("PFO_Hits_Charge_theta", &PFO_Hits_Charge_theta);
  _tree->Branch("PFO_Hits_Charge_phi", &PFO_Hits_Charge_phi);

  _tree->Branch("PFO_Energy_Neutral", &PFO_Energy_Neutral);
  _tree->Branch("PFO_Energy_Neutral_singleCluster", &PFO_Energy_Neutral_singleCluster);
  _tree->Branch("PFO_Energy_Neutral_singleCluster_R", &PFO_Energy_Neutral_singleCluster_R);
  _tree->Branch("PFO_Hits_Neutral_E", &PFO_Hits_Neutral_E);
  _tree->Branch("PFO_Hits_Neutral_R", &PFO_Hits_Neutral_R);
  _tree->Branch("PFO_Hits_Neutral_theta", &PFO_Hits_Neutral_theta);
  _tree->Branch("PFO_Hits_Neutral_phi", &PFO_Hits_Neutral_phi);
```

Jet performance parameters

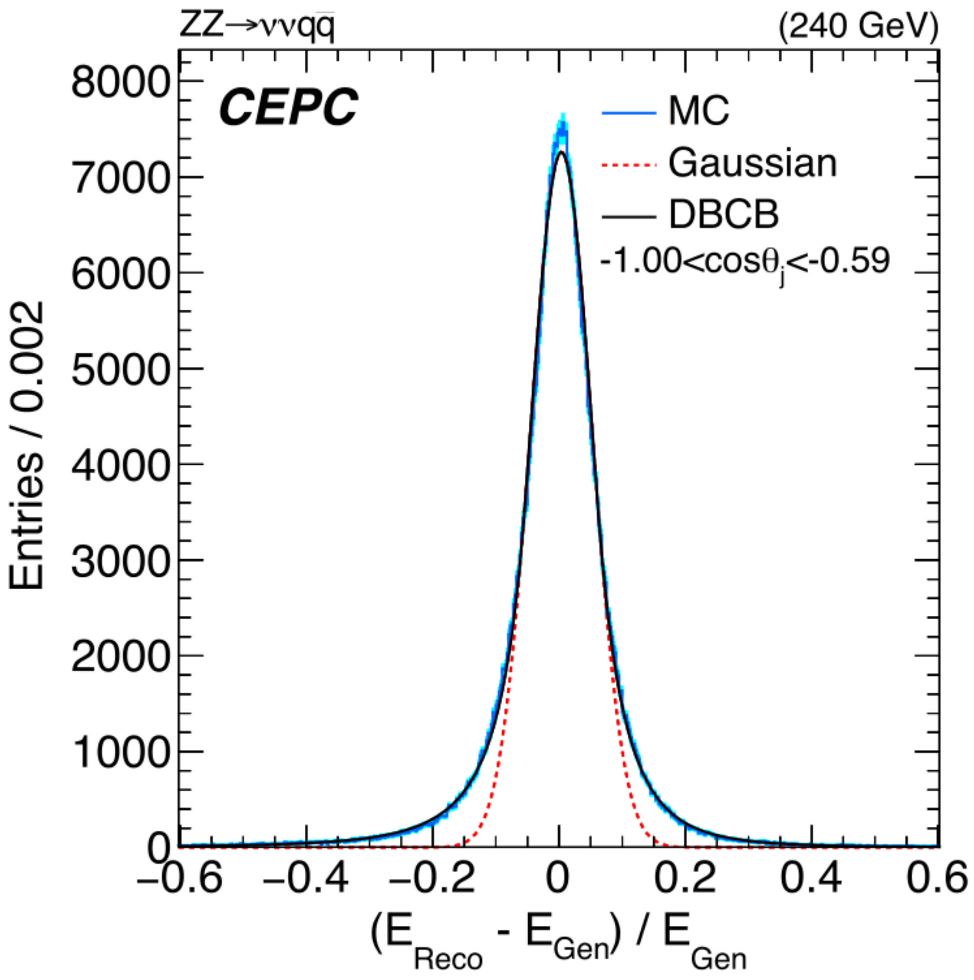
Peizhu, JINST

$$R_{R-G} = \frac{E_{\text{RecoJet}} - E_{\text{GenJet}}}{E_{\text{GenJet}}}$$

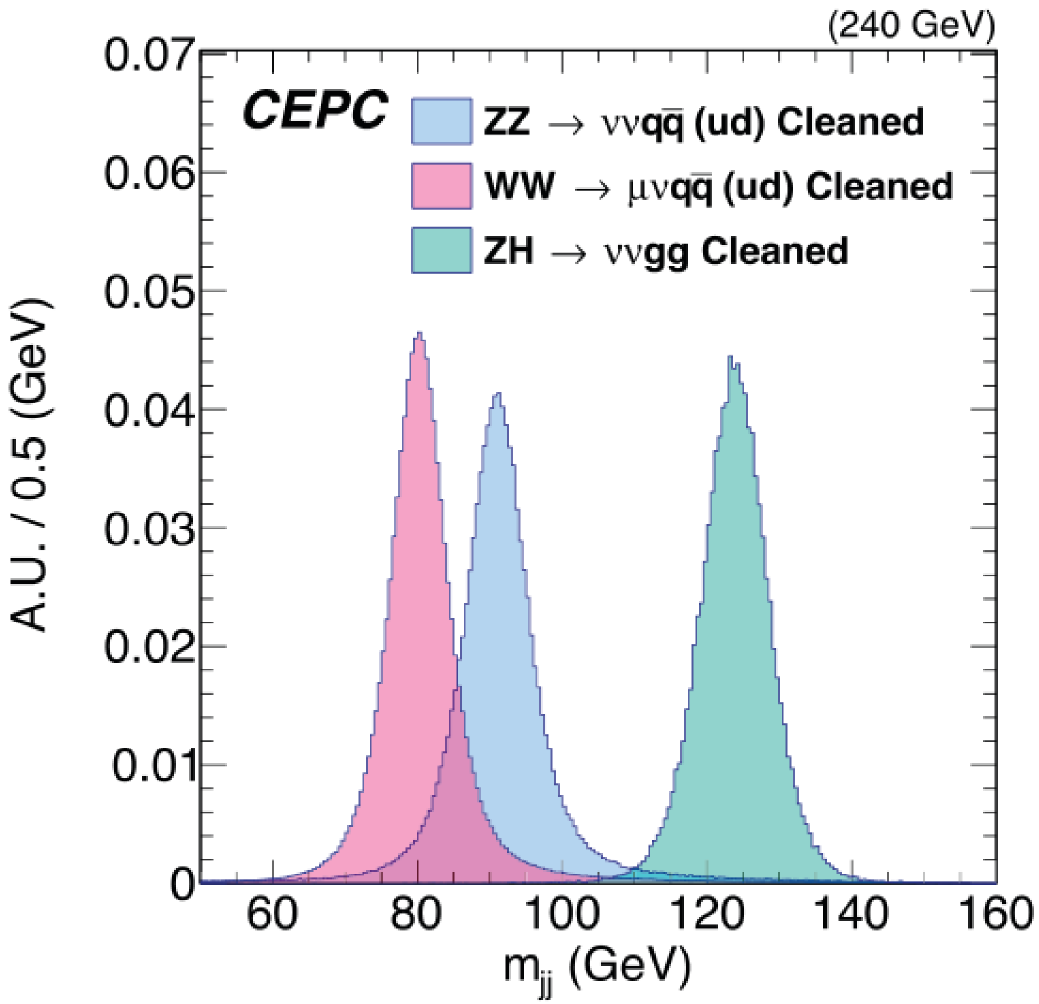
- After Reco-Gen matching,

In this difference plot (DSCB fit)

- JES Jet energy scale
  - Mean value shifted ($\bar{x}$)
- JER Jet energy resolution
  - Standard deviation (σ)



Jet performance parameters



- BMR(Boson mass resolution).
- ~Jet energy resolution.
- In CDR, when calculating BMR:
  - Veto total ISR components Pt > 1GeV;
  - Veto total neutrino Pt > 1GeV;
    - ISR and neutrino from single jet from Higgs.
  - Require Costheta_Jet;
- Current CEPCSW no endcap calo;
  - Require Costheta_Jet<0.65 (under tuning)

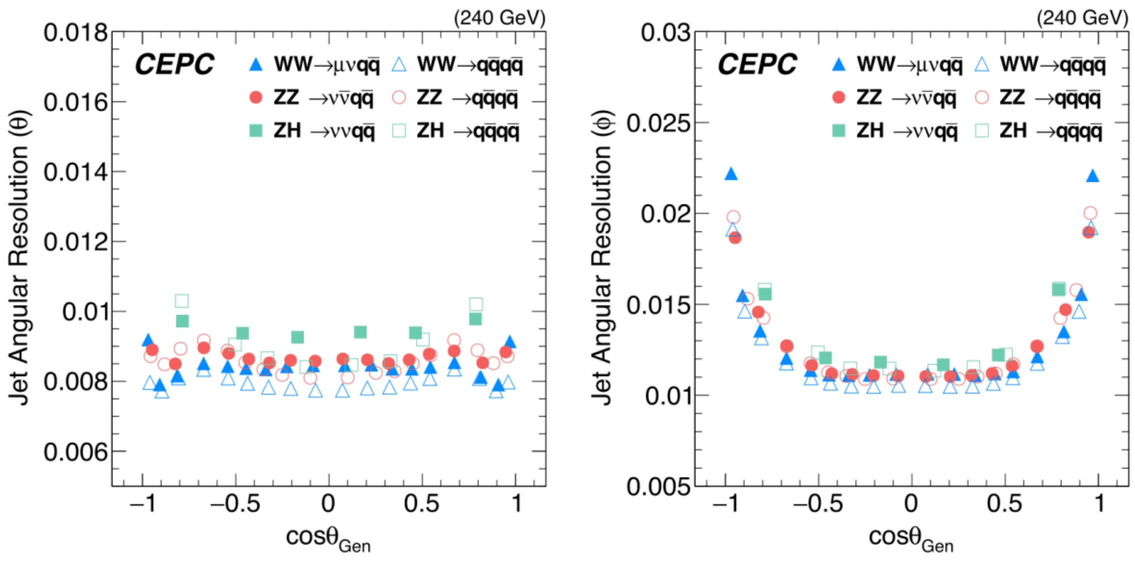
Table 1. Event cumulative efficiency for Higgs boson exclusive decay at the CEPC with $\sqrt{s} = 240$ GeV.

	gg(%)	bb(%)	cc(%)	WW*(%)	ZZ* (%)
Pt_ISR < 1 GeV	95.15	95.37	95.30	95.16	95.24
Pt_neutrino < 1 GeV	89.33	39.04	66.36	37.46	41.39
|Cos(Theta_Jet)| < 0.85	67.30	28.65	49.31	–	–

Jet angular performance

$$D_{R-G} = \theta_{\mathrm{RecoJet}} - \theta_{\mathrm{GenJet}} \quad \text{or} \quad \phi_{\mathrm{RecoJet}} - \phi_{\mathrm{GenJet}}$$

- JAR(θ, ϕ): Standard deviation (σ)
- JAS(θ, ϕ): Mean value shifted ($\bar{x}$)

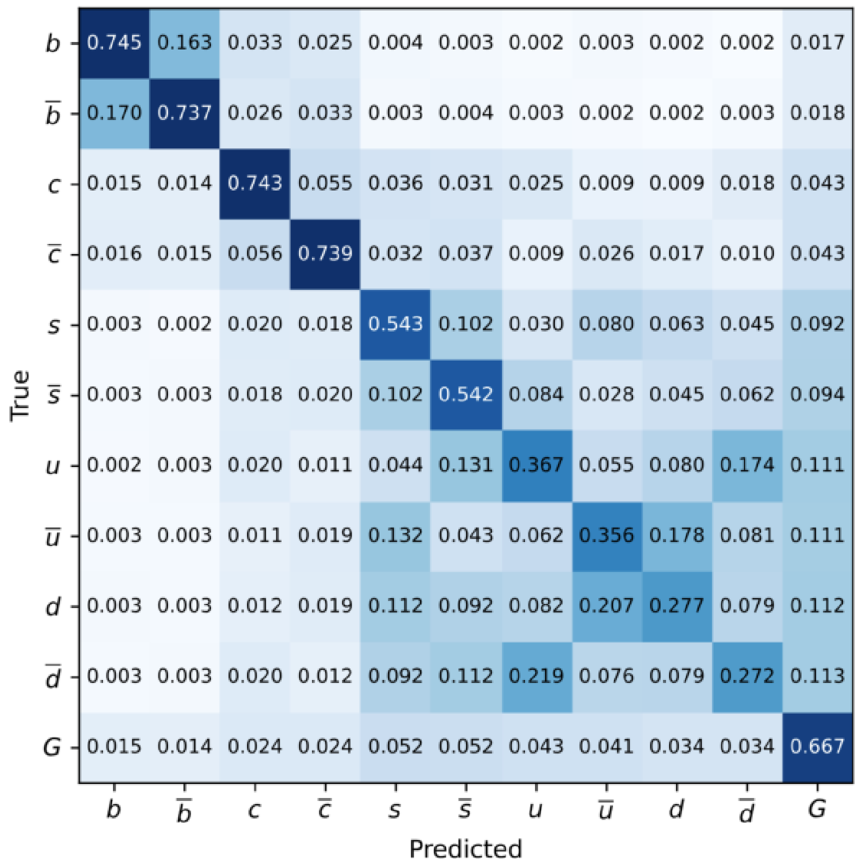


(a) (b)

Most of the plots and performance need re-check under current CEPC ref-TDR.
Both for performance study and sanity check.

Flavor information

- Use traditional LCFIplus package, or ML training like ParticleTransformer, jet flavor information can be tagged.
- ML shows better performance
- Need migration.



Sample preparing

Dijets:

- ZH, Z->vv, H->uu, dd, ss, bb, cc, gg each 100k;
- ZZ, ZZ->vvqq; 100k;
- WW, WW->mvqq 100k;
    - Need muon removal.

4jets: each 100k?

- ZH, Z->vv, H->ZZ->qqqq
- ZH, Z->vv, H->WW->qqqq
- ZZ, ZZ->qqqq
- WW, WW->qqqq

6jets(?)

- ZH->ZZZ(ZWW)->qqqqqq.

Need to verify after binning if the stats are enough.