Physics analysis at CEPC
--- view points of an analyzer

LI Gang

ligang@ihep.ac.cn

IH

Physics at CEPC

- Precision and systematical study of Higgs, W, and Z boson, as well as flavor physics with huge Z decay sample

  - WHZ physics (exclusive dominant)

  - Flavor physics (inclusive analysis)

Physics analysis at e+e- colliders: 1st dimension

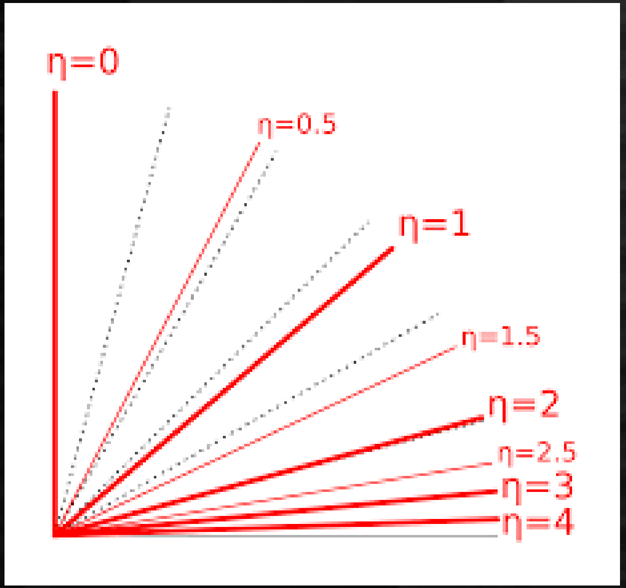
- Exclusive and inclusive
  - Exclusive analysis dominant
  - More constrain (information) available
  - Avoid information loss

Physics analysis at e+e- colliders: 2nd dimension

- 3D information, not only transverse
  - P_t
  - E_t
  - Missing X_t
- Use $\cos(\theta)$ instead of η

Forget about them !

$$r^2 dr d\cos\theta d\phi$$

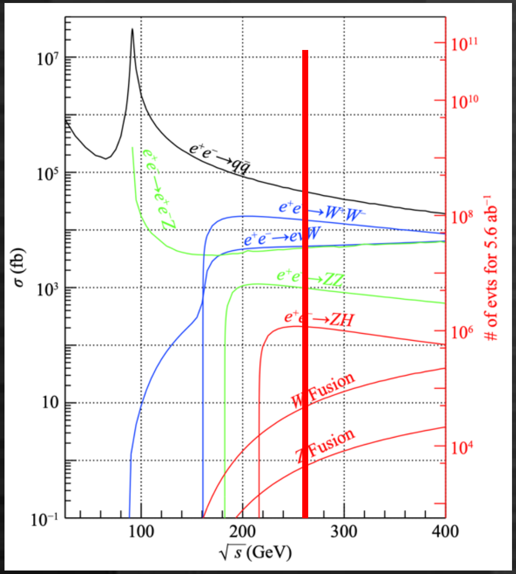


Physics analysis at e+e- colliders: 3rd dimension

- Reconstruct and analysis simultaneously
  - Why?
    - efficiency
    - Jet too soft
  - For example : $\mu^{+}\mu^{-}$ + di-jet
    - ~50 particles in an signal event: fast to process
    - 4 GeV/particle: every particle need to be used
    - The best way: selection muon pair first, then perform the jet clustering in exclusive mode on the rest particles
    - Testing all combinations is identical with reconstructed results

Feature 1: clean

- High S/N ratio: O(0.001)
- But less ~4 M Higgs
- Avoid eff. loss



Higgs signal
(~0.2pb)

2fermion
backgrounds
(~ 50pb)

4fermion
backgrounds
(~ 30 pb)

Feature 2: high (trigger) efficiency

- Even triggerless
  - Pile-up free(240 GeV and 360 GeV)
  - low background
  - High hermeticity : an complete event without fake particles

Feature 3: kinematic constrain

- Very powerful to improve (mass) resolution
- Provide an extra feature for suppress backgrounds

- Package: MarlinKinFit

Feature 4: use ee-kt exclusively

- Exclusive: each particle is useful in an event
- Analysis specific jet-clustering
- Or you will lose efficiency

```
vector<PseudoJet> particles;
// an event with three particles:   px    py  pz      E
particles.push_back( PseudoJet(   99.0,  0.1,  0, 100.0) );
particles.push_back( PseudoJet(    4.0, -4.1,  0,   5.0) );
particles.push_back( PseudoJet(    2.0, -8.1,  0,   5.0) );
particles.push_back( PseudoJet(   -2.0, -1.1,  0,   5.0) );
particles.push_back( PseudoJet(   -2.0,  9.1,  0,   5.0) );
particles.push_back( PseudoJet(  -99.0,  6.2,  0,  99.0) );

// choose a jet definition
int nJets  = 2;
JetDefinition jet_def(ee_kt_algorithm);

// run the clustering, extract the jets
ClusterSequence cs(particles, jet_def);
vector<PseudoJet> jets = sorted_by_pt(cs.exclusive_jets(nJets));

// print out some infos
cout << "Clustering with " << jet_def.description() << endl;

// print the jets
printf("                     p           costheta  phi\n");
for (unsigned i = 0; i < jets.size(); i++) {
   printf(" jet %2d             %9.4f %9.4f %9.4f\n", i, jets[i].pt(), jets[i].cos_theta(),  jets[i].phi() );
   vector<PseudoJet> constituents = jets[i].constituents();
   for (unsigned j = 0; j < constituents.size(); j++) {
      printf("    constituent  %2d %9.4f %9.4f %9.4f\n", i, constituents[j].pt(), constituents[j].cos_theta(),  constituents[j].phi() )
   }
}

printf("\n");
double _ymin[20];
for(int i=1; i<6;i++){
   _ymin[i-1] = cs.exclusive_ymerge (i);
   printf(" -log10(y%1d%1d) = %12.6f\n", i, i+1, -log10(_ymin[i-1]) );
}
```

An example

- e+e- → μ+μ-H
- Model independent measurement
- Only the muons from Z decay used

$$M_{\mathrm{recoil}} = \sqrt{s + M^2_{\mu^+\mu^-} - 2(E_{\mu^+} + E_{\mu^-})\sqrt{s}} \ ,$$

Muon candidates

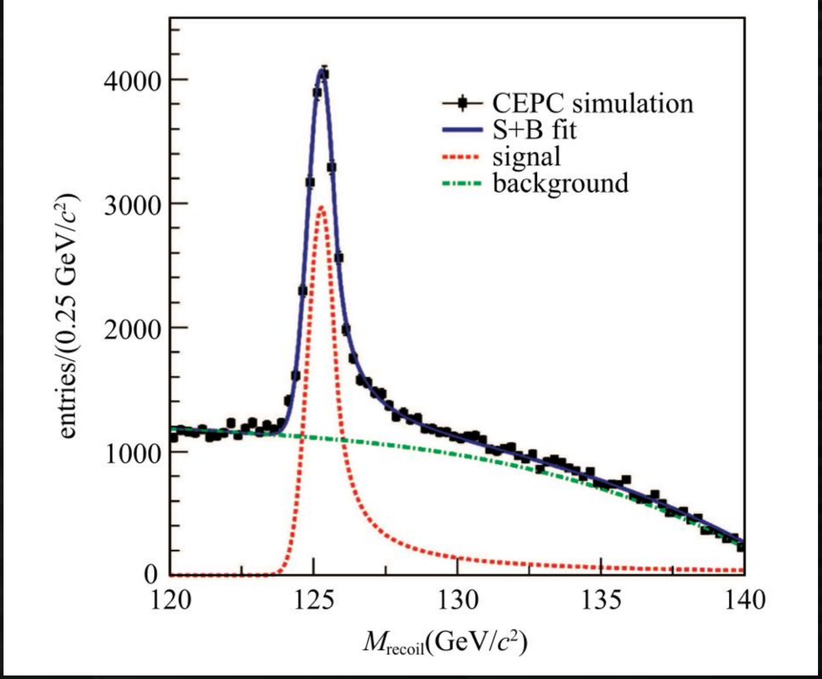
```
for(int j = 0; j < _nRecoP; j++)
{
        ReconstructedParticle *a_RecoP = dynamic_cast<EVENT::ReconstructedParticle *>(col_RecoP->getElementAt(j));
        if(a_RecoP->getCharge()==0) continue;
        RecoPID = a_RecoP->getType();
        RecoE = a_RecoP->getEnergy();
        RecoP[0] = a_RecoP->getMomentum()[0];
        RecoP[1] = a_RecoP->getMomentum()[1];
        RecoP[2] = a_RecoP->getMomentum()[2];

        TLorentzVector currP(RecoP[0], RecoP[1], RecoP[2], RecoE);

        if(RecoE>2.0) _NCh++;


        for(int s = 0; s < 4; s++)
        {
                _P_allCharged[s] += currP[s];
        }

        if( RecoE > 10 && RecoE < 100 ) //0.4*sqrt(s)
        {
                if(abs(RecoPID) == _leptonID )  //Put by hand... guess enough
                {
                        if(RecoPID == _leptonID )       //Got swapped...gosh!
                        {
                                FourMom_MuonM.push_back(currP);
                        }
                        else
                        {
                                FourMom_MuonP.push_back(currP);
                        }
                }
                else if( a_RecoP->getCharge() > 0.5 )
                {
                        P_ChP.push_back(currP);
                }
                else if( a_RecoP->getCharge() < -0.5 )
                {
                        P_ChM.push_back(currP);

                }
        }
}
```

Muon pair

```
if( NCandiP > 0 && NCandiM > 0 )
{
        for(int p = 0; p < NCandiP; p++)
        {
                P_P = CandiP[p];

                for(int m = 0; m < NCandiM; m++)
                {
                        P_M = CandiM[m];

                        currInvMass = (P_P + P_M).M();

                        if(fabs(currInvMass - 91.2) < MinZThrDis)
                        {
                                MinZThrDis = fabs(currInvMass - 91.2);
                                _InvMass = currInvMass;
                                for(int i=0; i<11; i++)
                                {
                                    currRecoilMass = (P_T[i] - P_P - P_M).M();
                                    _RecoilMass[i] = currRecoilMass;
                                }
                                for(int s = 0; s < 4; s++)
                                {
                                        _P_MuP[s] = P_P[s];
                                        _P_MuM[s] = P_M[s];
                                        _P_DL[s] = _P_MuP[s] + _P_MuM[s];
                                }
                                _acop = fabs(P_P.Phi()-P_M.Phi());
                                TLorentzVector miss = ecms - _P_DL;
                                _cosmis = miss.CosTheta();
                                _acol = P_P.Angle(P_M.Vect())*180./3.1415926;
                                _Pt_Z = sqrt(_P_DL[0]*_P_DL[0]+_P_DL[1]*_P_DL[1]);
                                _DeltaPt = _Pt_Z - _Pt_photon;
                                _cosZ = _P_DL[2]/sqrt(_P_DL[0]*_P_DL[0]+_P_DL[1]*_P_DL[1]+_P_DL[2]*_P_DL[2]);
                                float phi_p_tmp = atan2(_P_MuP[1],_P_MuP[0])*180./3.14159265;
                                float phi_m_tmp = atan2(_P_MuM[1],_P_MuM[0])*180./3.14159265;
                                if(_P_MuP[1] < 0) phi_p_tmp = phi_p_tmp + 360.;
                                if(_P_MuM[1] < 0) phi_m_tmp = phi_m_tmp + 360.;
                                _D_phi = fabs(phi_p_tmp - phi_m_tmp);
                                if (_D_phi > 180) _D_phi = 360. - _D_phi;
                        }
                }
        }
}
```

Recoil mass

```
for(int i2  = 0; i2 < 4; i2++)
{
        _P_allReco[i2] = _P_allCharged[i2] + _P_allNeutral[i2];
        _P_Higgs[i2] = _P_allReco[i2] - _P_DL[i2];
}

_Hmass = sqrt( _P_Higgs[3]* _P_Higgs[3] - _P_Higgs[0]* _P_Higgs[0] - _P_Higgs[1]* _P_Higgs[1] - _P_Higgs[2]* _P_Higgs[2] );
```



Another example

- e+e- → μ+μ-H, Higgs →di-jet

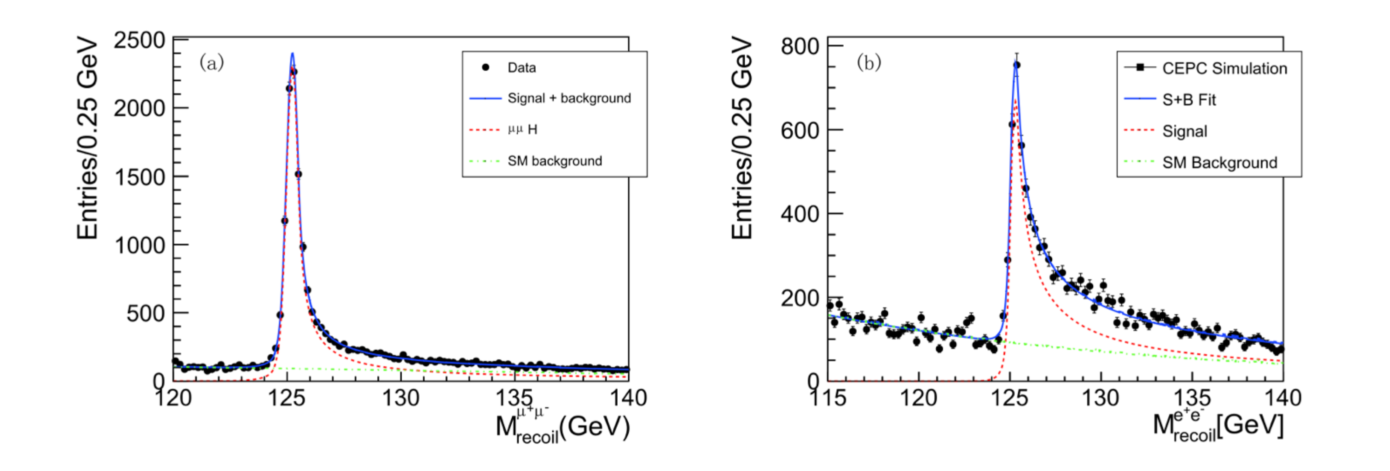
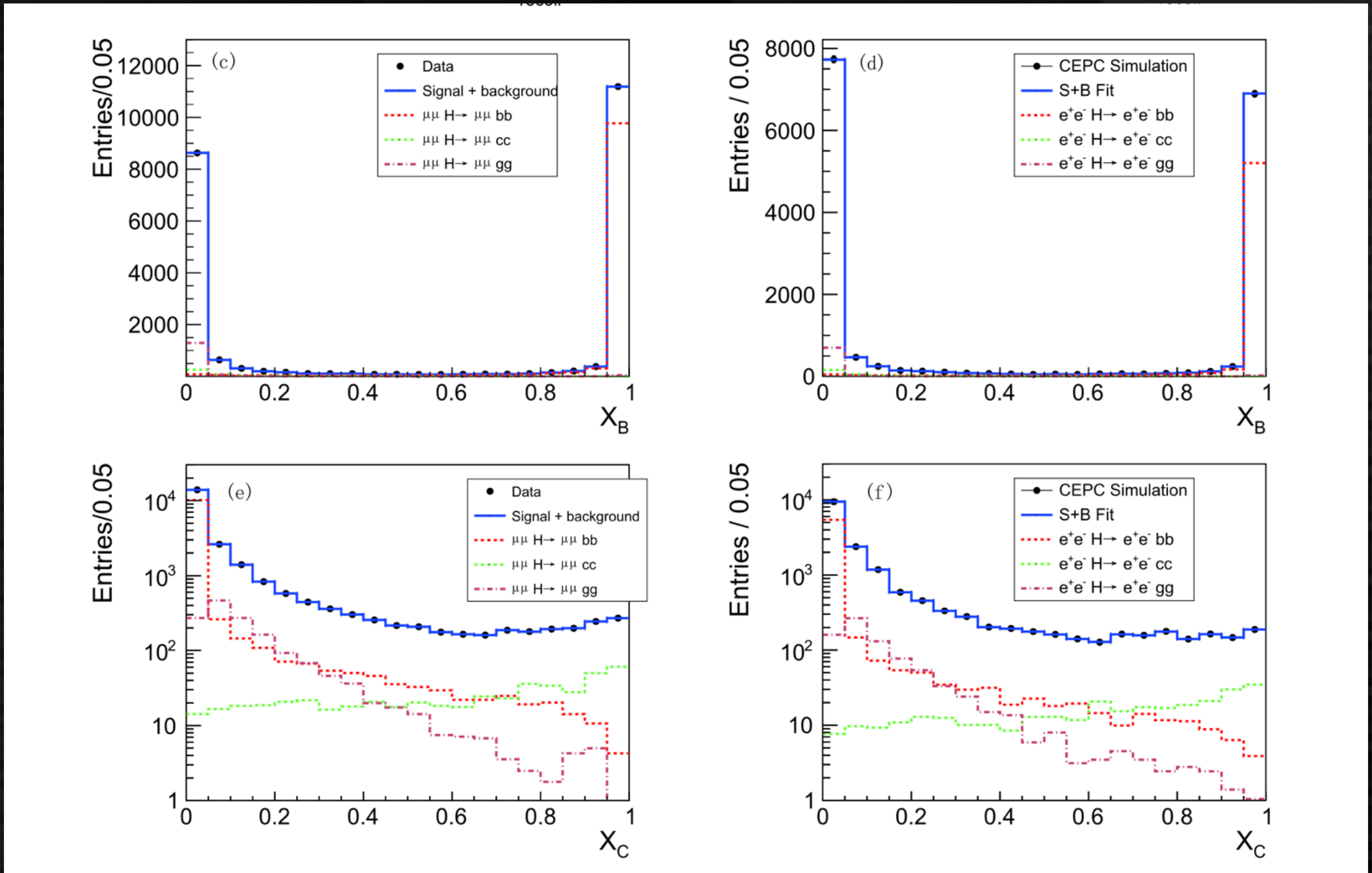


Table 2. Uncertainties on $\sigma^{b\bar{b}}_{l^+l^-H}$, $\sigma^{c\bar{c}}_{l^+l^-H}$ and $\sigma^{gg}_{l^+l^-H}$.

Higgs boson production	$\mu^+\mu^-H$			e^+e^-H		
Higgs boson decay	$H\to b\bar{b}$	$H\to c\bar{c}$	$H\to gg$	$H\to b\bar{b}$	$H\to c\bar{c}$	$H\to gg$
statistic uncertainty	1.1%	10.5%	5.4%	1.6%	14.7%	10.5%
fixed background	−0.2% +0.1%	+4.1% −4.2%	7.6%	−0.2% +0.1%	+4.1% −4.2%	7.6%
event selection	+0.7% −0.2%	+0.4% −1.1%	+0.7% −1.7%	+0.7% −0.2%	+0.4% −1.1%	+0.7% −1.7%
flavor tagging	−0.4% +0.2%	+3.7% −5.0%	+0.2% −0.7%	−0.4% +0.2%	+3.7% −5.0%	+0.2% −0.7%
combined systematic uncertainty	+0.7% −0.5%	+5.5% −6.6%	+7.6% −7.8%	+0.7% −0.5%	+5.5% −6.6%	+7.6% −7.8%

Another example

- e+e- → μ+μ-H, Higgs →di-jet



Another example

- e+e- → μ+μ-H, Higgs →di-jet

Table 2. Uncertainties on $\sigma^{b\bar{b}}_{l^+l^-H}$, $\sigma^{c\bar{c}}_{l^+l^-H}$ and $\sigma^{gg}_{l^+l^-H}$.

Higgs boson production	$\mu^+\mu^-H$			e^+e^-H		
Higgs boson decay	$H\to b\bar{b}$	$H\to c\bar{c}$	$H\to gg$	$H\to b\bar{b}$	$H\to c\bar{c}$	$H\to gg$
statistic uncertainty	1.1%	10.5%	5.4%	1.6%	14.7%	10.5%
fixed background	−0.2% +0.1%	+4.1% −4.2%	7.6%	−0.2% +0.1%	+4.1% −4.2%	7.6%
event selection	+0.7% −0.2%	+0.4% −1.1%	+0.7% −1.7%	+0.7% −0.2%	+0.4% −1.1%	+0.7% −1.7%
flavor tagging	−0.4% +0.2%	+3.7% −5.0%	+0.2% −0.7%	−0.4% +0.2%	+3.7% −5.0%	+0.2% −0.7%
combined systematic uncertainty	+0.7% −0.5%	+5.5% −6.6%	+7.6% −7.8%	+0.7% −0.5%	+5.5% −6.6%	+7.6% −7.8%

Job options

```
<processor name="FSClasserProcessor" type="FSClasserProcessor">
  <!--Name of the MCParticle collection-->
  <parameter name="InputMCParticlesCollection" type="string" lcioInType="MCParticle"> MCParticle </parameter>
  <parameter name="InputMCTruthLinkCollection" type="string" lcioInType="LCRelation"> RecoMCTruthLink </parameter>
  <parameter name="InputIsoLepsCollection"     type="string" lcioInType="ReconstructedParticle"> ArborPFOs </parameter>
  <parameter name="InputPandoraPFOsCollection" type="string" lcioInType="ReconstructedParticle"> ArborPFOs </parameter>
  <parameter name="InputJetsCollection" type="string" lcioInType="ReconstructedParticle"> RefinedJets </parameter>
  <!-- -->
  <!-- -->
  <parameter name="FS130"    type="string">     INC2_0000000 </parameter>
  <parameter name="FS131"    type="string">     INC0_0001100 </parameter>
  <parameter name="FS132"    type="string">     EXC2_0001100 </parameter>
  <parameter name="FS133"    type="string">     EXC0_2001100 </parameter>
  <!-- -->
  <parameter name="FastOrFull"    type="int"   >  0       </parameter>
  <parameter name="ShowMC"        type="int"   >  0       </parameter>
  <!-- -->
  <parameter name="Verbosity"     type="string">  4       </parameter>
  <parameter name="DEBUG"         type="string">  1       </parameter>
  <parameter name="Luxury"        type="string">  1       </parameter>
  <parameter name="MatchMC"       type="string">  1       </parameter>
  <parameter name="TagFlavor"     type="string">  0       </parameter>
  <parameter name="kmfit"         type="string">  1       </parameter>
  <parameter name="Kappa"         type="string">  1.0     </parameter>
  <parameter name="ECM"           type="string">  250.0   </parameter>
</processor>
```

Information on screen

- Marlin FS_example.xml

```
[ VERBOSE "FSClasserProcessor"] FSClasser:  Initializing Final State INC2_0000000
[ VERBOSE "FSClasserProcessor"] FSClasser:  Checking the Final State INC2_0000000
FSClasser:       jet:  normal
FSClasser:       jet:  normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:  Initializing Final State INC0_0001100
[ VERBOSE "FSClasserProcessor"] FSClasser:  Checking the Final State INC0_0001100
FSClasser:       mu+:  normal
FSClasser:       mu-:  normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:  Initializing Final State EXC2_0001100
[ VERBOSE "FSClasserProcessor"] FSClasser:  Checking the Final State EXC2_0001100
FSClasser:       jet:  normal
FSClasser:       jet:  normal
FSClasser:       mu+:  normal
FSClasser:       mu-:  normal
[ VERBOSE "FSClasserProcessor"]
[ VERBOSE "FSClasserProcessor"] FSClasser:  Initializing Final State EXC0_2001100
[ VERBOSE "FSClasserProcessor"] FSClasser:  Checking the Final State EXC0_2001100
FSClasser:     gamma:  normal
FSClasser:     gamma:  normal
FSClasser:       mu+:  normal
FSClasser:       mu-:  normal
Channel    0: INC2_0000000
```

Summary

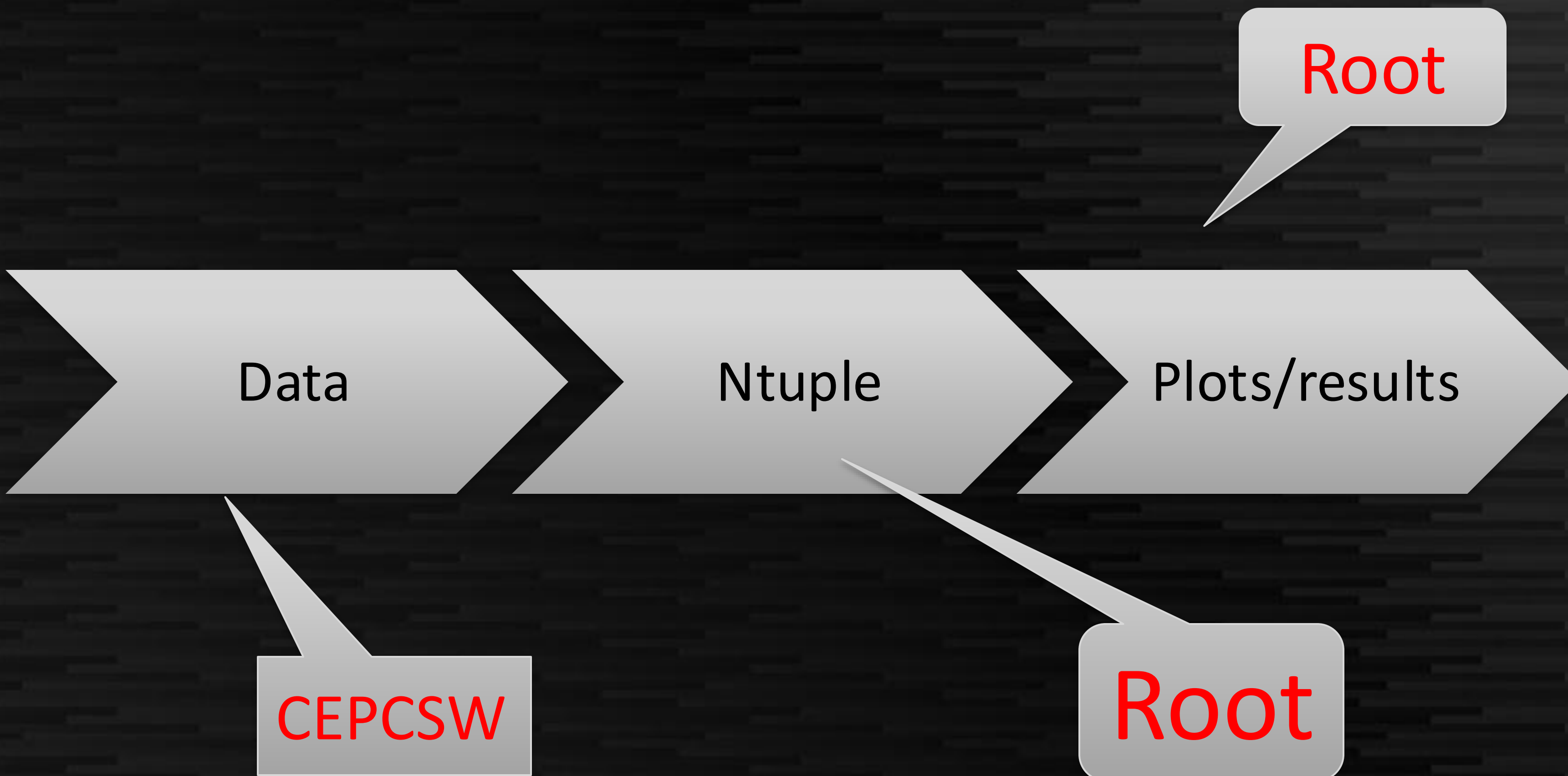
- Analysis at e+e- colliders
  - Get used to do (some) reconstruction by yourself, tuning it according to your analysis
  - Try to use all particles in an event
  - Try to use kinematic fit
  - Using p and cos(θ) instead of pt and eta respectively

Backups

Overview of data-analysis

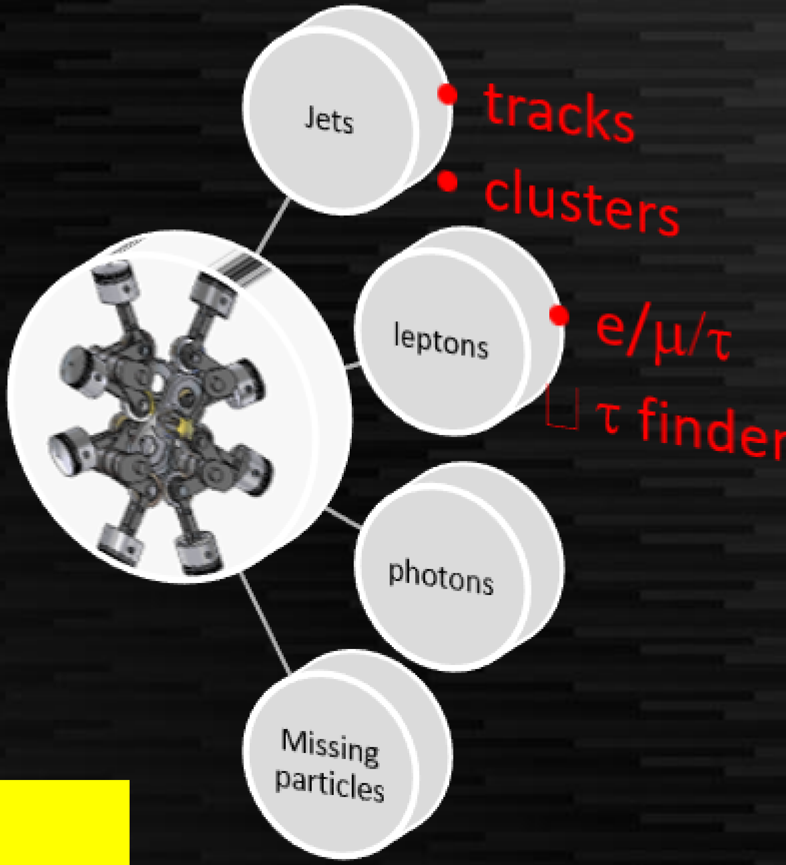
- Two stages:
  - Pre-selection and Ntuple production
  - Root script – plots and numerical results
- First state
  - Particle Objects
    - MC particles – used for comparison
    - Reconstructed particles (tracks, clusters, jets) → event
  - Combination of objects → candidate events
  - Fill ntuples for the next stage in root …

Overview of data-analysis (cont'd)



Overview of data-analysis (cont'd)

Feed all types of particle object to the combination engine for further processing



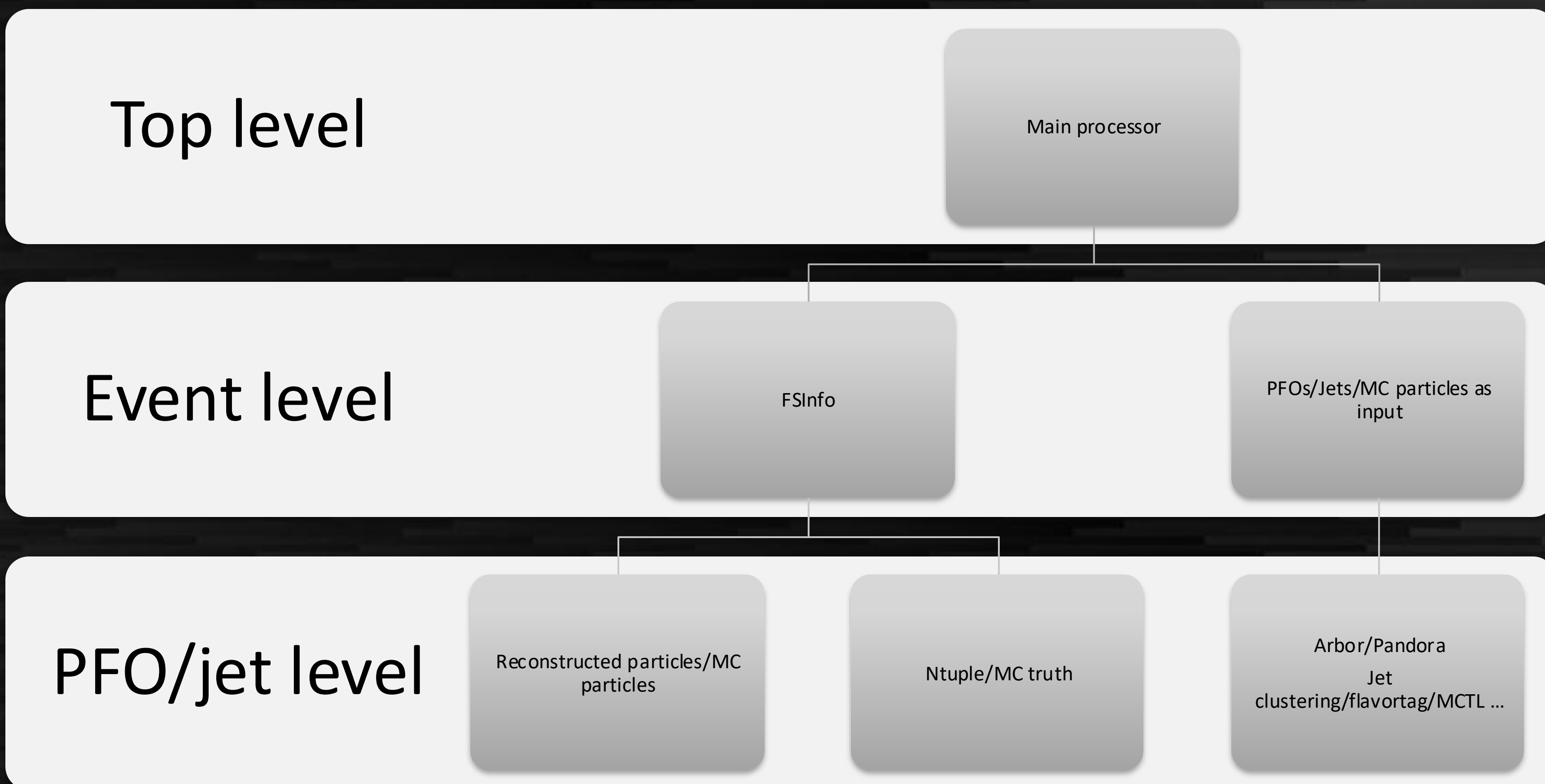
ee+X, μμ+X, jj+ee, jj+μμ …

Abstract of tasks

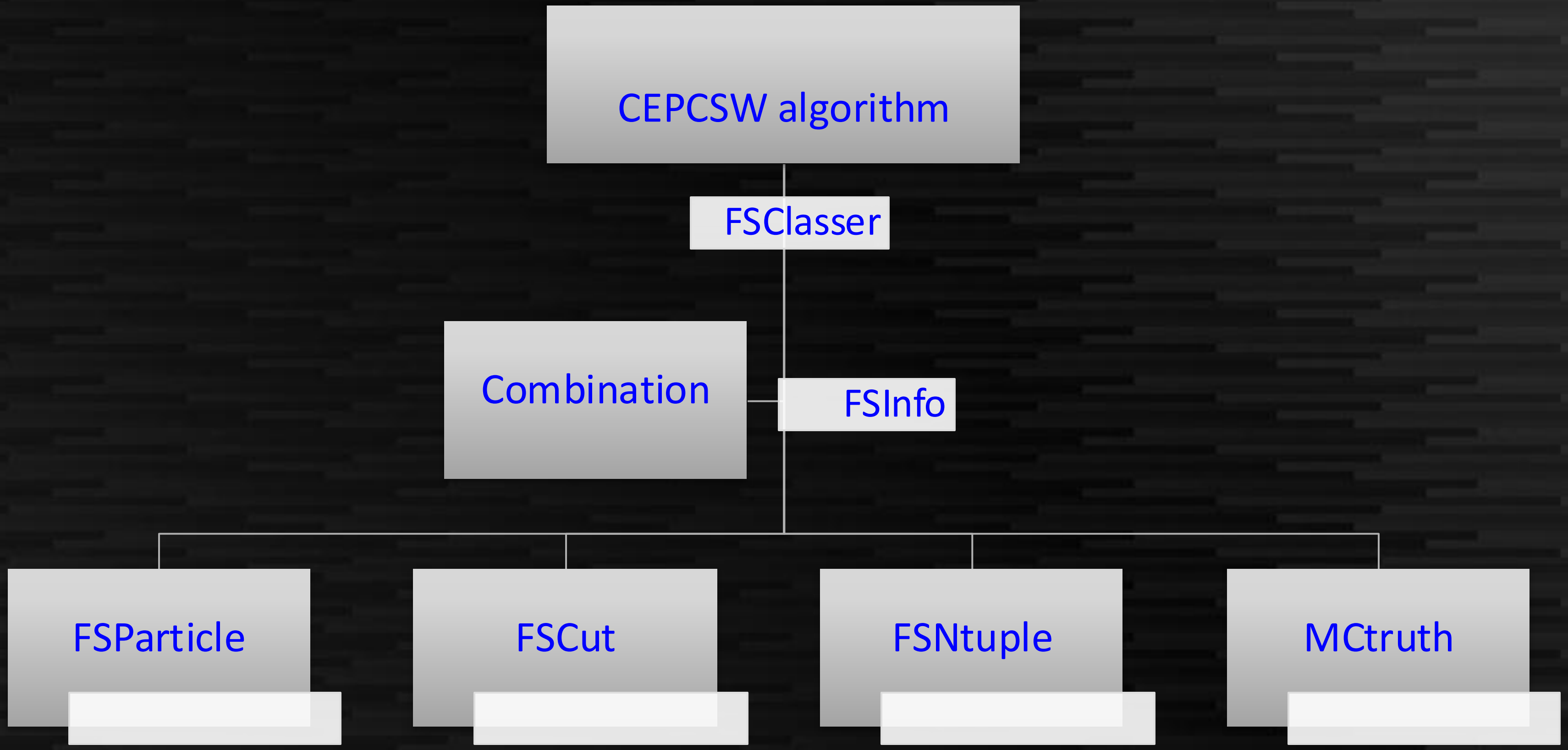
- Class FSParticle → all types of (reconstructed) particles
- Class FSinfo → all kinds of combination
- Class NTupleHelper → service of ntuple
- Class MCTruthHelper → service of MC truth
- Class FSCut → preliminary cuts

CEPC analysis

Final state classification



Structure of code



Structure of main program

Prepare particle lists	
e,μ, γ	Jet, τ

↓

Combination the particle list as the request	
Events, Ntuple,	MC truth

↓

Loop over all combinations: calculate variables and fill ntuple	
Simple cuts	Kinematic fit

↓

Fill ntuples

Class FSPaticle

- Data
  - PID/Mass/charge/ 4-momentum, p, pT …
  - Flavor/vertex
  - Matched MC object

```
ReconstructedParticle *        m_pfo;
MCParticle *                   m_mcp;
//JetFitObject*                  m_JetFitObject;


string   m_name;
int      m_type;
int      m_pdgid;
bool     m_missed;
bool     m_fast;
double   m_mass;
double   m_recmass;
double   m_charge;
double   m_pT;
double   m_pZ;
double   m_Energy;
double   m_Rapidity;
double   m_CosTheta;
double   m_btag;
double   m_ctag;
double   m_bctag;
double   m_flavor;

TLorentzVector      m_rawFourMomentum;
TLorentzVector      m_fitFourMomentum;

vector<int>         m_trackId;
vector<int>         m_showerId;
```

FSInfo

- Data
  - Combination of a list of particles/jets
  - the associated MC truth/ Ntuple
  - Cuts
  - Steers

```
private:
   string m_FSName;
   vector<string> m_particleNames;
   vector<int>    m_particleStatus;
   int            m_nChargedParticles;
   int            m_nMissingParticles;
   NTupleHelper*  m_NT;
   NTupleHelper*  m_NTGen;

   int m_decayCode1;
   int m_decayCode2;

   bool   m_fast;

   bool   m_Constrain4Mom ;
   bool   m_missingMassFit;
   double m_missingMassValue;
   string m_missedParticle;

  vector< vector<unsigned int> >& submodeIndices(const string& submodeN

   vector<FSCut*>                                  m_FSCuts;
   vector<vector<FSParticle*> >                    m_particleCombinations;
   map<string, vector< vector <unsigned int> > > m_submodeIndices;
```