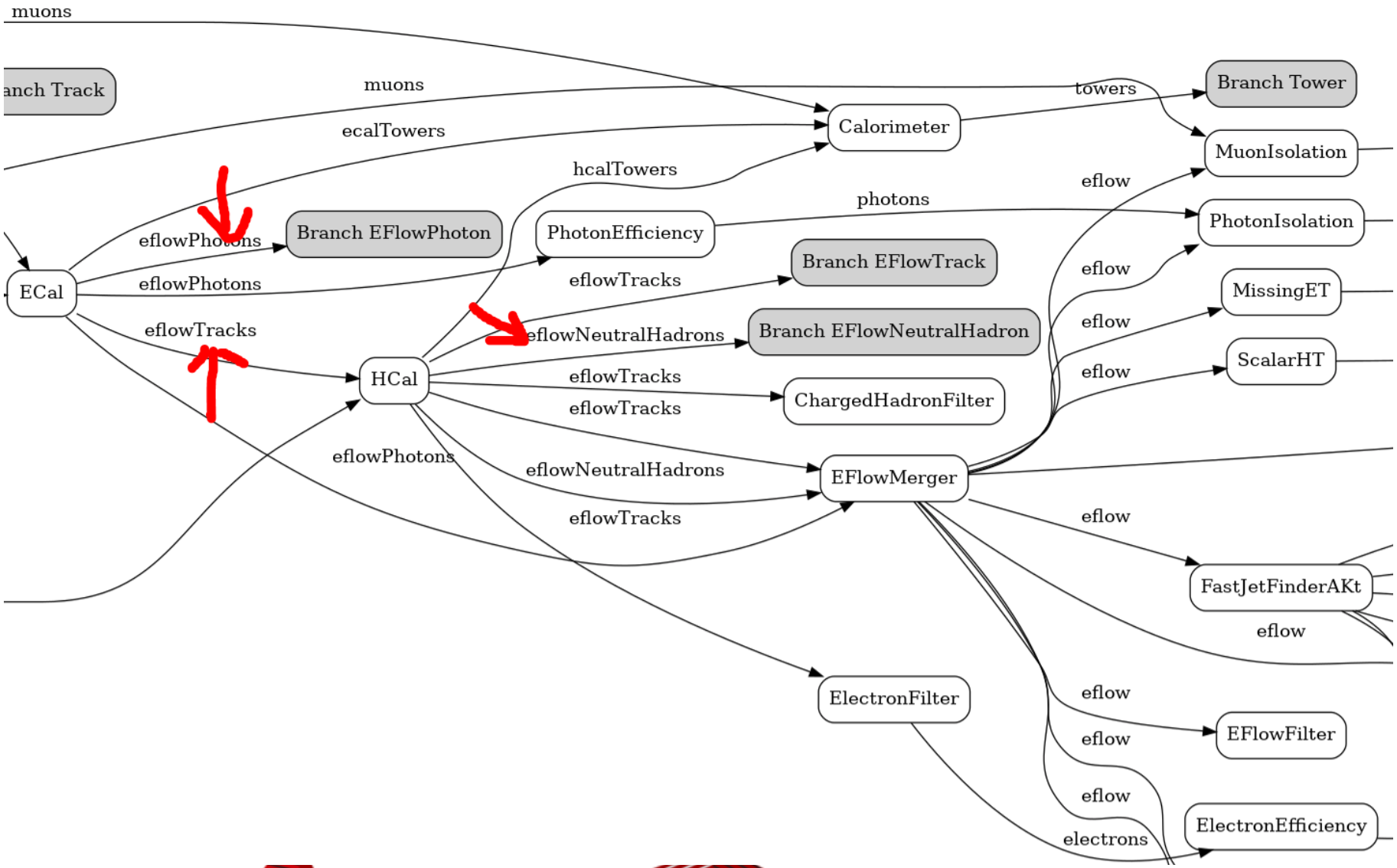
Measuring the Higgs Self-coupling Using the Muon Collider

Kenny Jia

Mar 13th, 2023


muons
anch Track
muons
ecalTowers
towers
Branch Tower
Calorimeter
MuonIsolation
hcalTowers
eflow
photons
eflowPhotons
Branch EFlowPhoton
PhotonEfficiency
PhotonIsolation
ECal
eflowPhotons
eflowTracks
Branch EFlowTrack
eflow
MissingET
eflowTracks
eflowNeutralHadrons
Branch EFlowNeutralHadron
eflow
ScalarHT
HCal
eflowTracks
ChargedHadronFilter
eflow
eflowTracks
eflowPhotons
eflowNeutralHadrons
EFlowMerger
eflowTracks
eflow
FastJetFinderAKt
eflow
ElectronFilter
eflow
EFlowFilter
eflow
eflow
ElectronEfficiency
electrons

```
module BIBECal BIBECal {
    set PhotonInputArray ECal/eflowPhotons
    set TtackInputArray ECal/eflowTracks

    set PhotonOutputArray eflowPhotons
    set TrackOutputArray eflowTracks


    set NumParticles 9000000
        #37759050

    set FileName /nfs_scratch/hjia38/MG5_aMC_v3_3_2/Delphes/cards/MuonCollider/histograms_MCPar_muComb_files1To8_Allevts.root
    set xHistName x
    set PositionHistName z_r
    #set PhiHistName phi
    #set ThetaHistName theta
    set MomentumHistName px_py_pz
    set PdgEnergyHistName pdgid_energy
    set PhotonsDeltaR 0.5
    set TracksDeltaR 0.5

}
```

```
//check if BIB generated in proximity of original eflowPhotons
if (EMFlag) {
    while((original = static_cast<Candidate *>(originalPhotons -> Next()))) {
        original = static_cast<Candidate *>(original -> Clone());
        TLorentzVector *originalMomentum = original -> Momentum;
        if (originalMomentum.DeltaR(bibMomentum) < fPhotonsDeltaR) proximity = true;
    }
    if (proximity) fPhotonsOutputArray -> Add(bib);
}

if (HadFlag) {
    while((original = static_cast<Candidate *>(originalTracks -> Next()))) {
        original = static_cast<Candidate *>(original -> Clone());
        TLorentzVector *originalMomentum = original -> Momentum;
        if (originalMomentum.DeltaR(bibMomentum) < fTracksDeltaR) proximity = true;
    }
    if (proximity) fTracksOutputArray -> Add(bib);
}
```

- Currently, I am just adding BIB particles into eflowTracks, eflowPhotons. But we could not know the kinematics of these eflows with only kinematic info of BIB particles on generator level. How could we know the distribution of tracks?

- Maybe we could write our own ECal and HCal adding BIBs to it and let the Delphes SimpleCalorimeter module finish the classification and calorimeter work? Maybe a simple particle propagation charged particle and calorimeter in just one module to save memory use?

```
#############
#   ECAL
#############

module SimpleCalorimeter ECal {
    set ParticleInputArray ParticlePropagator/stableParticles
    set TrackInputArray TrackMerger/tracks

    set TowerOutputArray ecalTowers
    set EFlowTrackOutputArray eflowTracks
    set EFlowTowerOutputArray eflowPhotons
```

```
#############
#   HCAL
#############

module SimpleCalorimeter HCal {
    set ParticleInputArray ParticlePropagator/stableParticles
    set TrackInputArray ECal/eflowTracks

    set TowerOutputArray hcalTowers
    set EFlowTrackOutputArray eflowTracks
    set EFlowTowerOutputArray eflowNeutralHadrons
```