



CEPC

Jets, samples and Wednesday working meeting

Kaili Zhang

zhangkl@ihep.ac.cn

- /hpcfs/cepc/higgsgpu/zhangkl/CEPCSW/Service/PrimeTagSvc
- Under developing: Structure following FinalPID as one service.
 - LoadPFOs; (For 1 jet)
 - Prepare Input;
 - Get Predictions from onnxmodel. Return likelihood, no WP yet;
 - Need 1 week to provide one demo package.

```
bool LoadPF0s(std::vector<edm4hep::MutableReconstructedParticle>& PF0s, int n_jet)
{
    //Currently for only 1 jet first.
    size_t n_pfo = PF0s->size();
    if (n_pfo == 0) {
        debug() << "No PF0s found in the collection." << endmsg;
        return false;
    }
    if (n_pfo < n_jet) {
        debug() << "Number of PF0s is less than the number of jets requested: " << n_pfo << " < " << n_jet << endmsg;
        return false;
    }
    if (n_jet < 1) {
        debug() << "Number of jets requested is less than 1: " << n_jet << endmsg;
        return false;
    }

    TLorentzVector T_Jet;
    for (size_t i=0; i<n_pfo; i++)
    {
        edm4hep::MutableReconstructedParticle pfo = PF0s->at(i);
        if (isnan(pfo.getMomentum()[0]) || isnan(pfo.getMomentum()[1]) || isnan(pfo.getMomentum()[2]) || pfo.getEnergy() == 0) {
            debug() << "Invalid PF0 found at index " << i << ": NaN momentum or zero energy." << endmsg;
            continue;
        }

        TLorentzVector p_pfo = TLorentzVector(pfo.getMomentum()[0], pfo.getMomentum()[1], pfo.getMomentum()[2], pfo.getEnergy());
        T_Jet+=p_pfo;
    }
    for (size_t i=0; i<n_pfo; i++)
    {
        TLorentzVector p_pfo = TLorentzVector(pfo.getMomentum()[0], pfo.getMomentum()[1], pfo.getMomentum()[2], pfo.getEnergy());
        auto pidpfo = pfo.clone();
        bool load = m_pid_svc->LoadPF0(pidpfo);
        if (!load) debug() << "Load PF0 failed." << endmsg;

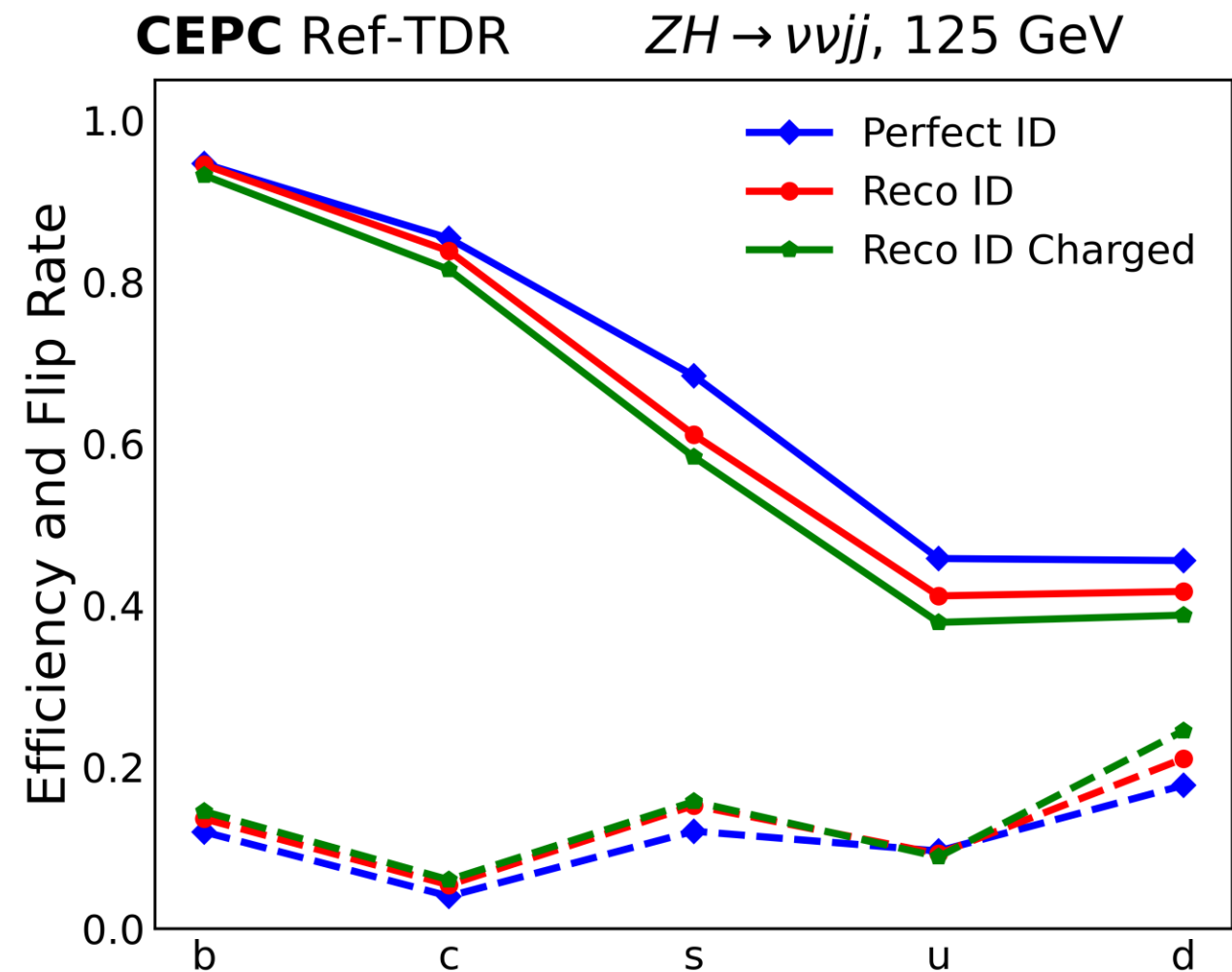
        bool Trackmatched = false;
        if (pfo.tracks_size() >= 1) {
            Trackmatched = true;
        }
    }
}
```

```
// Input all ready. Preparing Reader input.
std::map<std::string, std::vector<double> > map_input; map_input.clear();
std::vector<std::string> names = {"pf_points", "pf_features", "pf_vectors", "pf_mask"};
std::vector<std::string> pf_points_name = {"dphi", "deta"};
std::vector<std::string> pf_vectors_name = {"px", "py", "pz", "R_E"};
std::vector<std::string> pf_features_name = {
    "c_pt", "c_energy", "c_rel_energy",
    "c_rel_pt", "c_deltaR", "charge",
    "c_Ecal", "c_EclusterSize", "c_HclusterSize",
    "c_EcalR90", "c_HcalR90", "c_EcalL", "c_HcalL",
    "c_EcalWeta2", "c_HcalWeta2", "c_EcalWphi2", "c_HcalWphi2",
    "c_MindR_last", "c_dd_last", "c_Nmuon",
    "c_TOF", "c_Dndx", "dphi", "deta",
    "c_ChiTPC_e", "c_ChiTPC_mu", "c_ChiTPC_pi", "c_ChiTPC_kaon", "c_ChiTPC_p",
    "c_ChiTOF_e", "c_ChiTOF_mu", "c_ChiTOF_pi", "c_ChiTOF_kaon", "c_ChiTOF_p",
    "R_PElectron", "R_PMuon", "R_PPion", "R_PKaon", "R_Proton", "R_PPhoton",
    "d0", "z0", "d0err", "z0err",
    "px", "py", "pz", "R_E"}; //2+4+42(2)

map_input["c_pt"] = Preprocessing(part_c_p);
map_input["c_energy"] = Preprocessing(part_c_E);
map_input["c_rel_energy"] = Preprocessing(part_c_rel_E);
map_input["c_rel_pt"] = Preprocessing(part_c_rel_p);
map_input["c_deltaR"] = Preprocessing(part_c_deltaR);
map_input["charge"] = Preprocessing(part_charge);
map_input["R_PKaon"] = Preprocessing(part_Prob_kaon);
map_input["R_PPion"] = Preprocessing(part_Prob_pi);
map_input["R_PProton"] = Preprocessing(part_Prob_p);
map_input["R_PElectron"] = Preprocessing(part_Prob_e);
map_input["R_PMuon"] = Preprocessing(part_Prob_mu);
map_input["R_PPhoton"] = Preprocessing(part_Prob_photon);
map_input["c_Ecal"] = Preprocessing(part_c_Ecal);
map_input["c_EclusterSize"] = Preprocessing(part_c_EclusterSize);
map_input["c_HclusterSize"] = Preprocessing(part_c_HclusterSize);
map_input["c_EcalR90"] = Preprocessing(part_c_EcalR90);
map_input["c_HcalR90"] = Preprocessing(part_c_HcalR90);
map_input["c_EcalL"] = Preprocessing(part_c_EcalL);
map_input["c_HcalL"] = Preprocessing(part_c_HcalL);
map_input["c_EcalWeta2"] = Preprocessing(part_c_EcalWeta2);
map_input["c_HcalWeta2"] = Preprocessing(part_c_HcalWeta2);
```

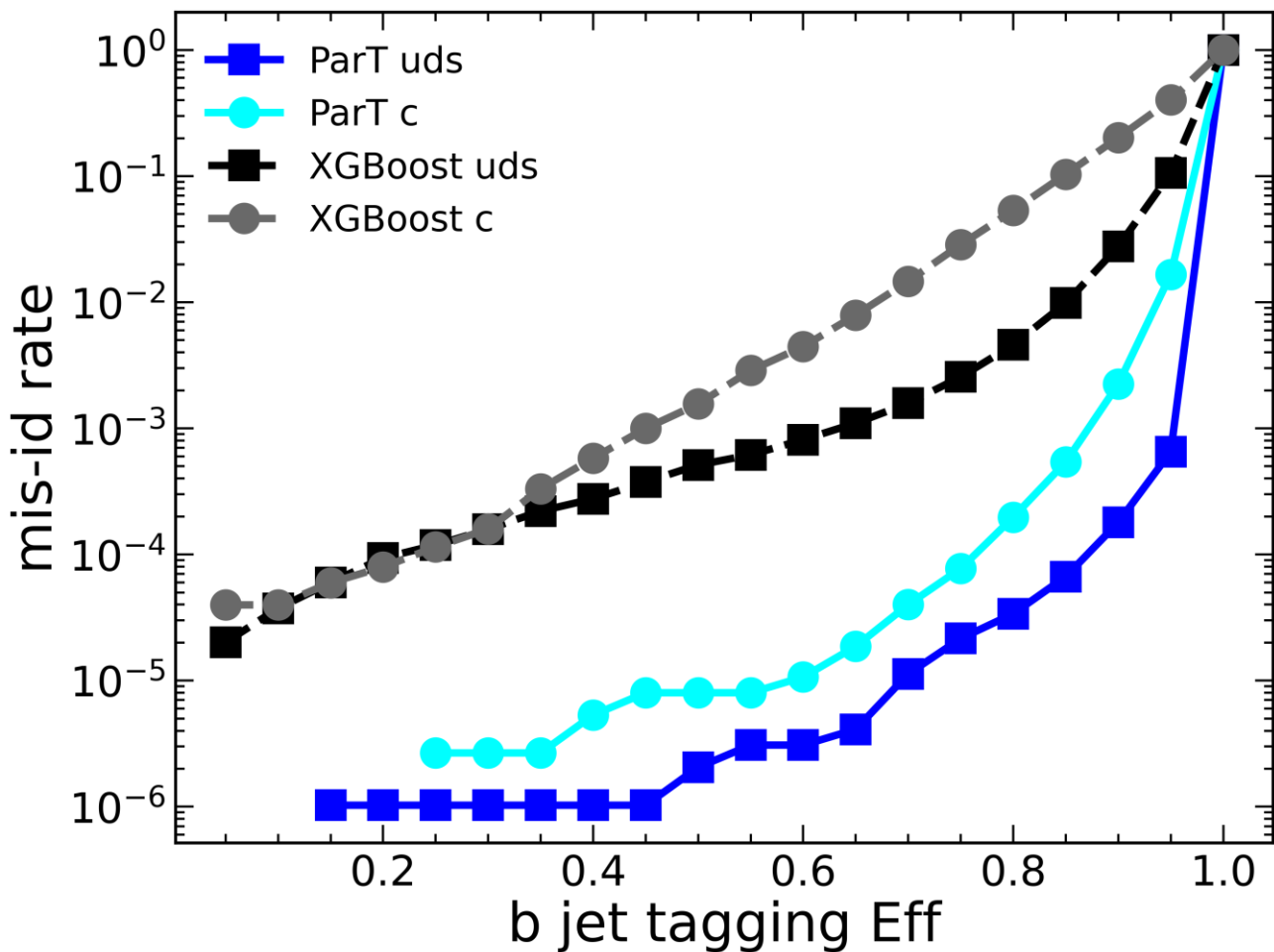
CEPC Ref-TDR $H \rightarrow qq$, 125 GeV

Truth \ Predicted	b	$\bar{b}$	c	$\bar{c}$	s	$\bar{s}$	u	$\bar{u}$	d	$\bar{d}$	g
b	0.811	0.132	0.019	0.016	0.002	0.001	0.001	0.002	0.002	0.001	0.013
$\bar{b}$	0.124	0.819	0.017	0.018	0.001	0.002	0.002	0.001	0.001	0.002	0.014
c	0.009	0.012	0.798	0.042	0.019	0.027	0.027	0.006	0.007	0.017	0.035
$\bar{c}$	0.013	0.011	0.049	0.790	0.027	0.022	0.006	0.026	0.016	0.007	0.033
s	0.002	0.001	0.016	0.019	0.488	0.095	0.028	0.119	0.093	0.053	0.084
$\bar{s}$	0.001	0.002	0.020	0.015	0.084	0.508	0.124	0.024	0.049	0.091	0.082
u	0.001	0.002	0.021	0.008	0.035	0.146	0.413	0.037	0.068	0.178	0.092
$\bar{u}$	0.002	0.001	0.008	0.021	0.139	0.040	0.045	0.391	0.189	0.070	0.093
d	0.002	0.001	0.011	0.019	0.124	0.088	0.066	0.218	0.296	0.080	0.096
$\bar{d}$	0.001	0.002	0.020	0.009	0.078	0.132	0.239	0.059	0.076	0.289	0.095
g	0.011	0.012	0.029	0.029	0.074	0.077	0.072	0.066	0.057	0.057	0.514



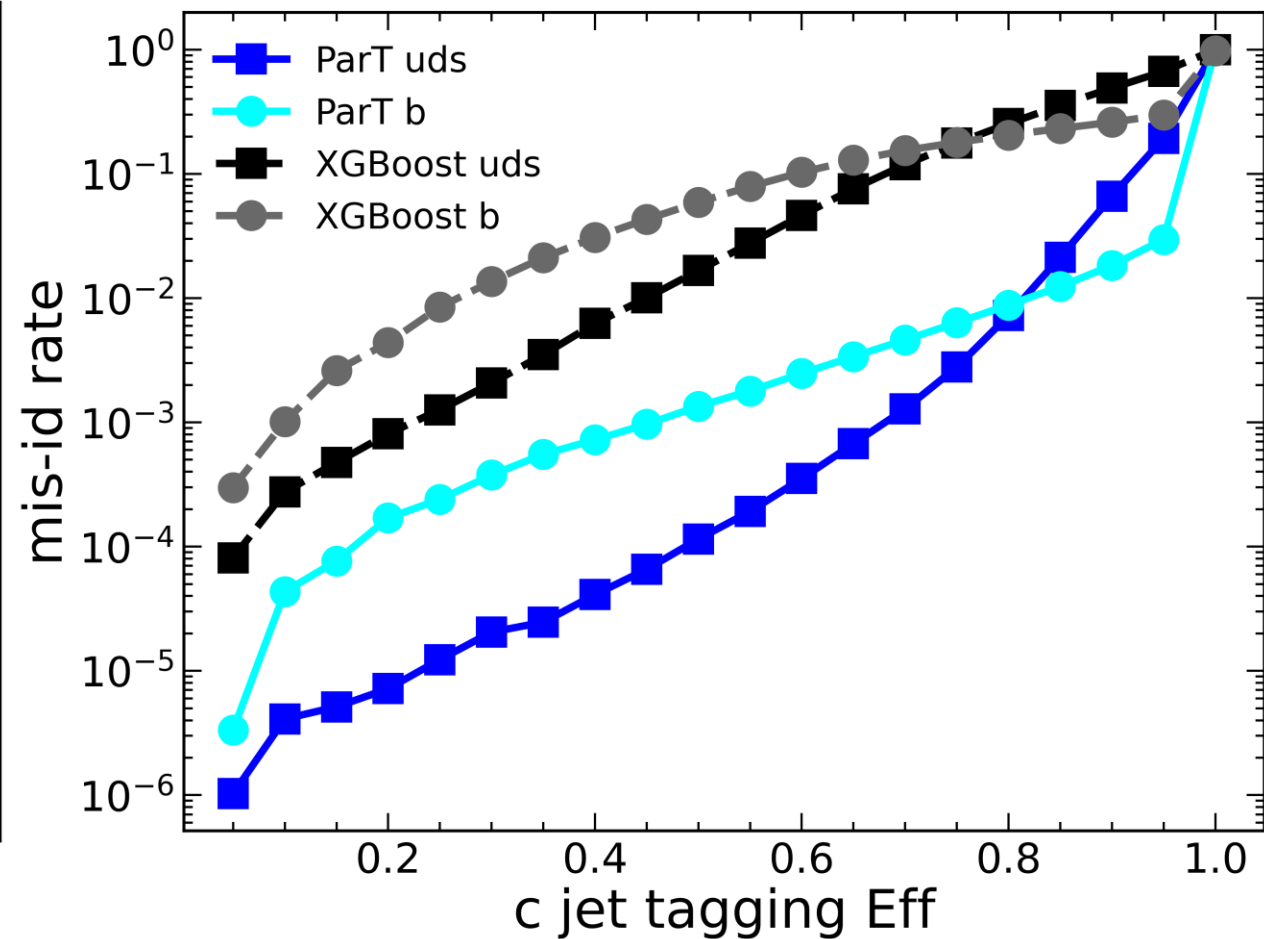
CEPC Ref-TDR

$Z \rightarrow qq$, 91.2 GeV

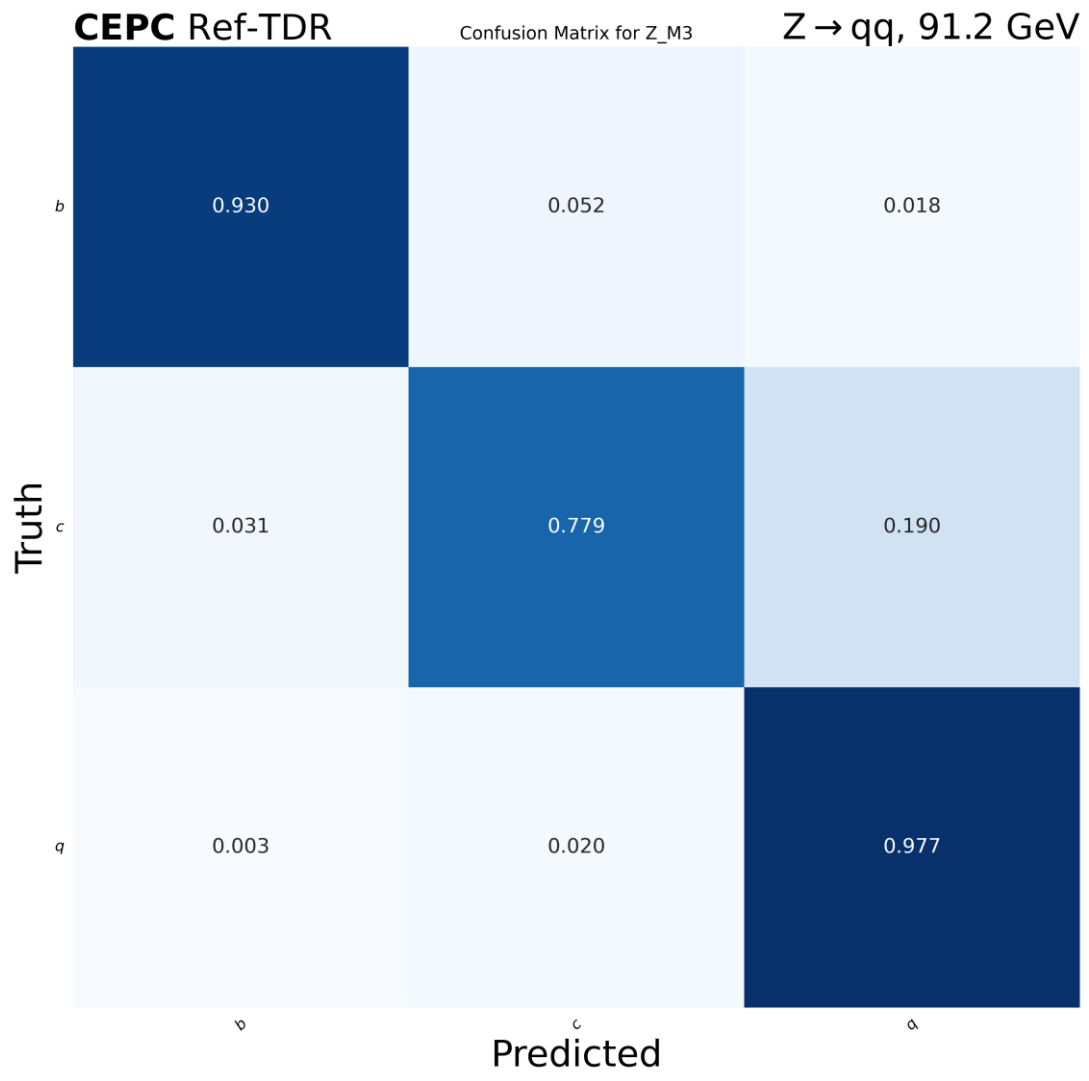
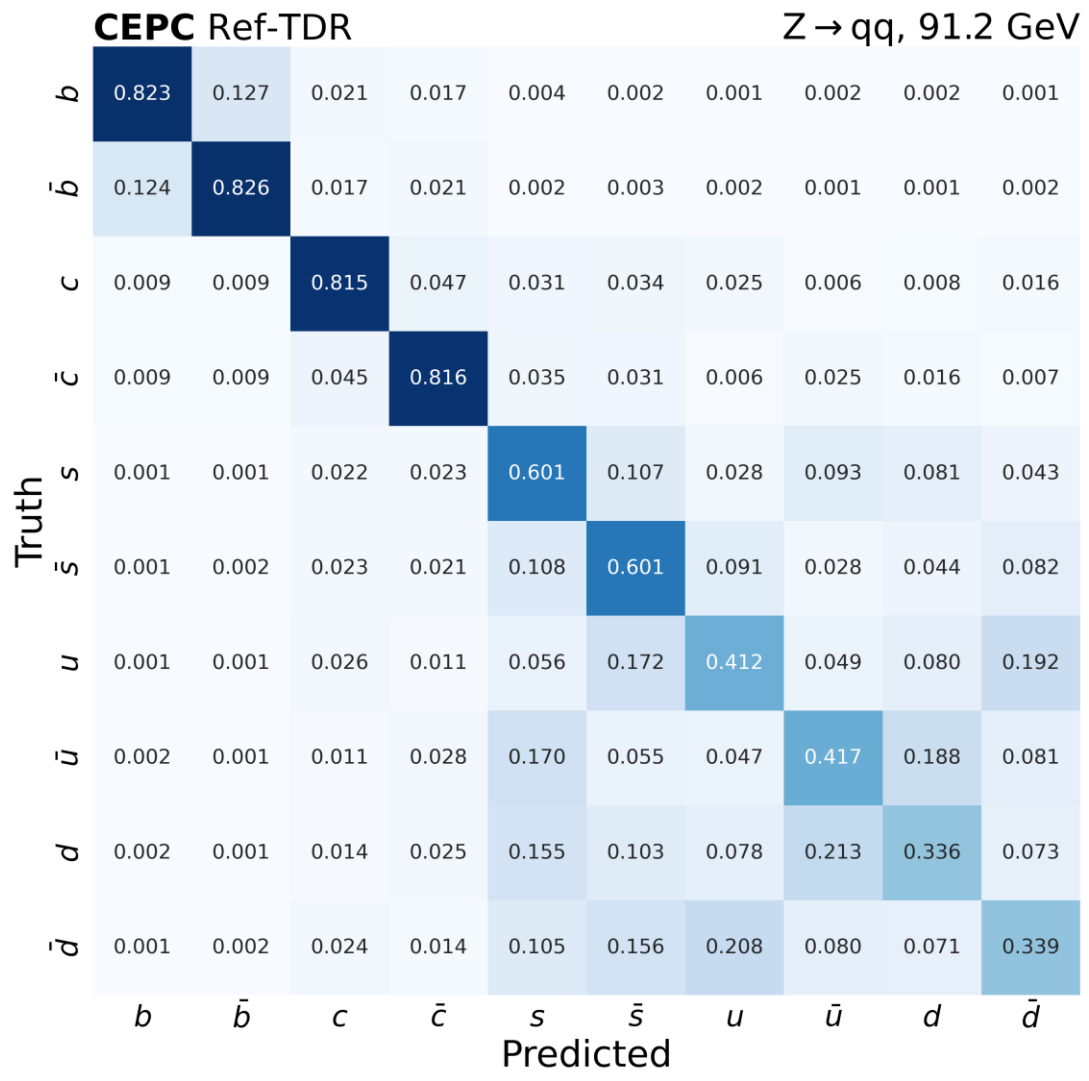


CEPC Ref-TDR

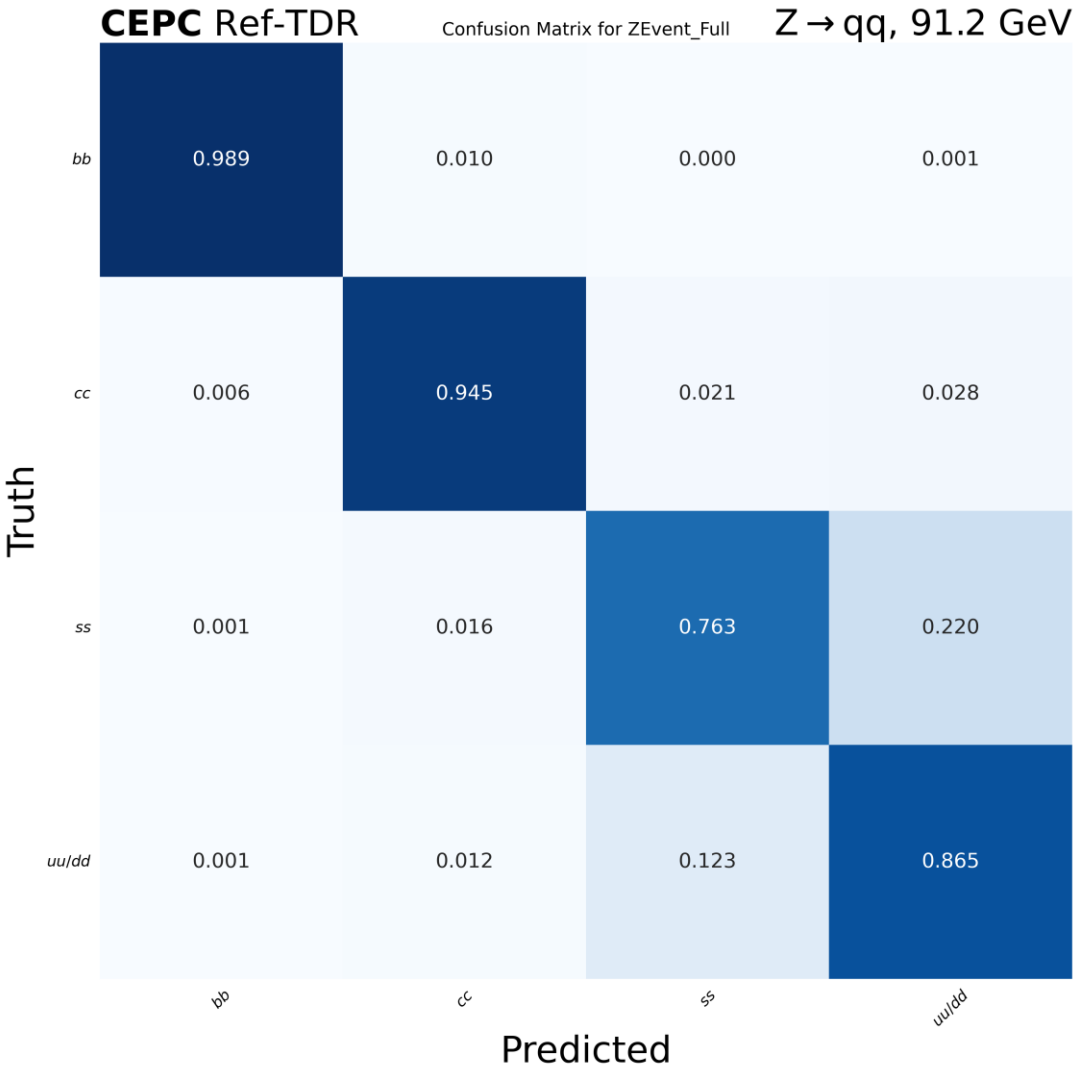
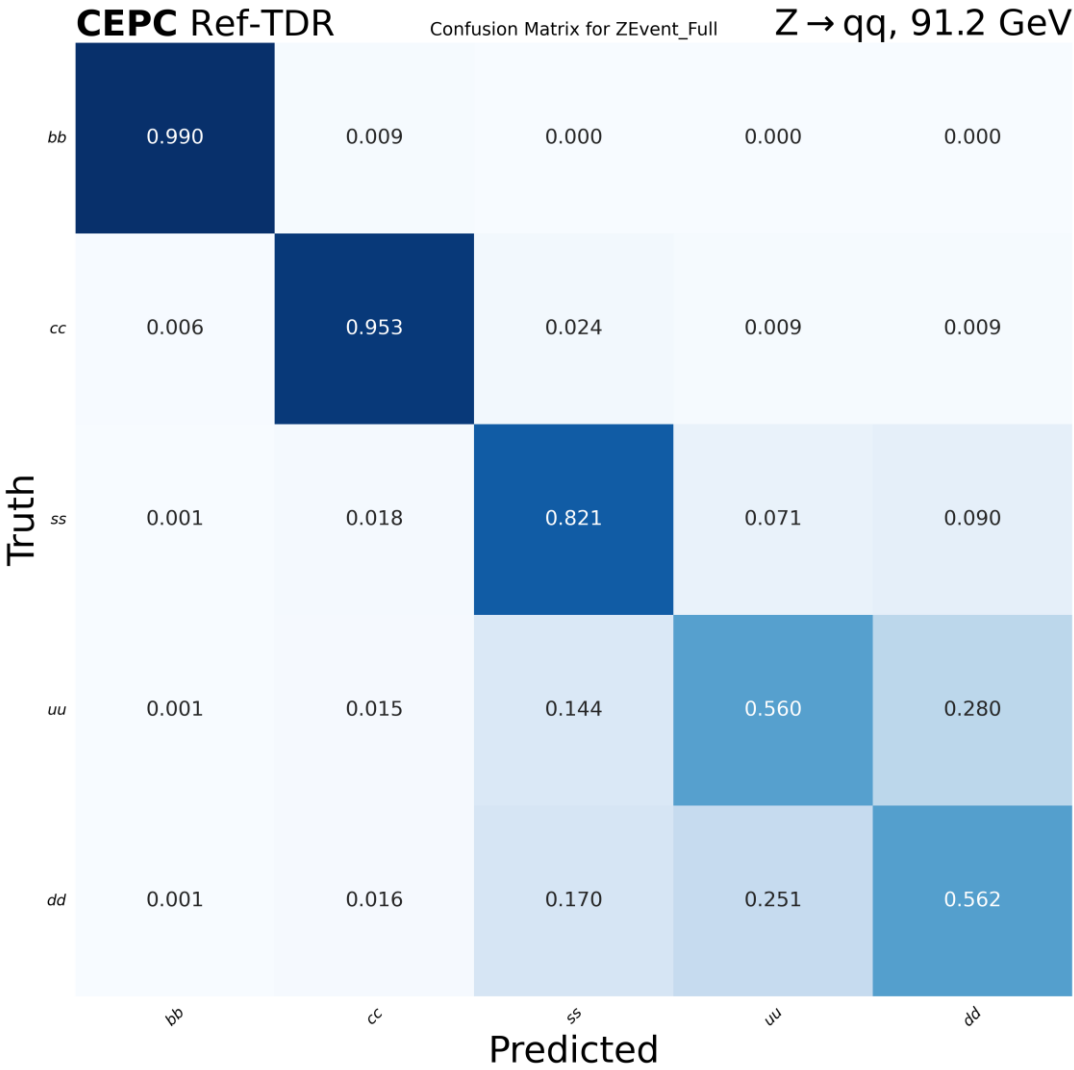
$Z \rightarrow qq$, 91.2 GeV

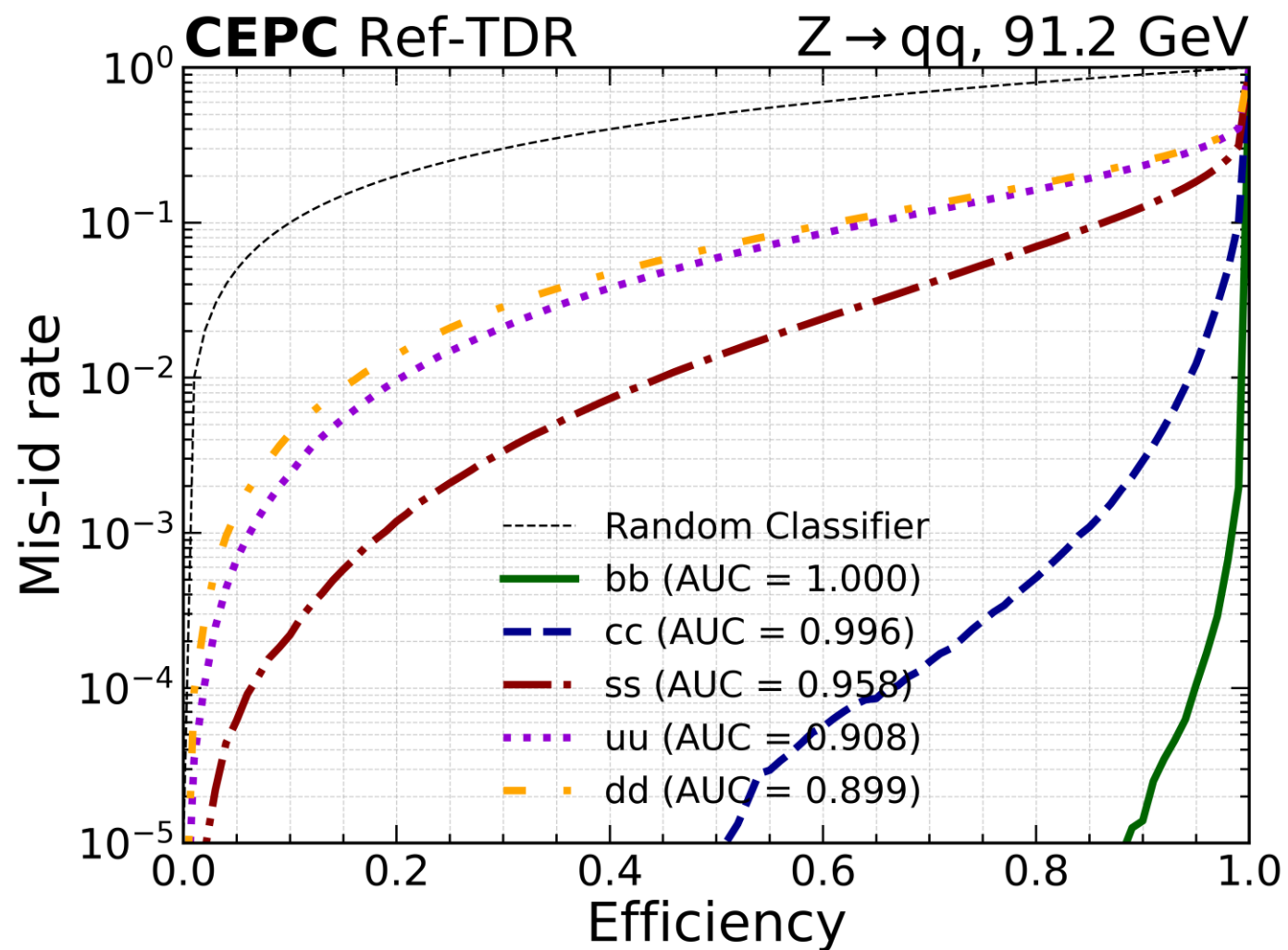


M11 and M3



Event category: M5, M4





H125, M16

