CEPC JOI

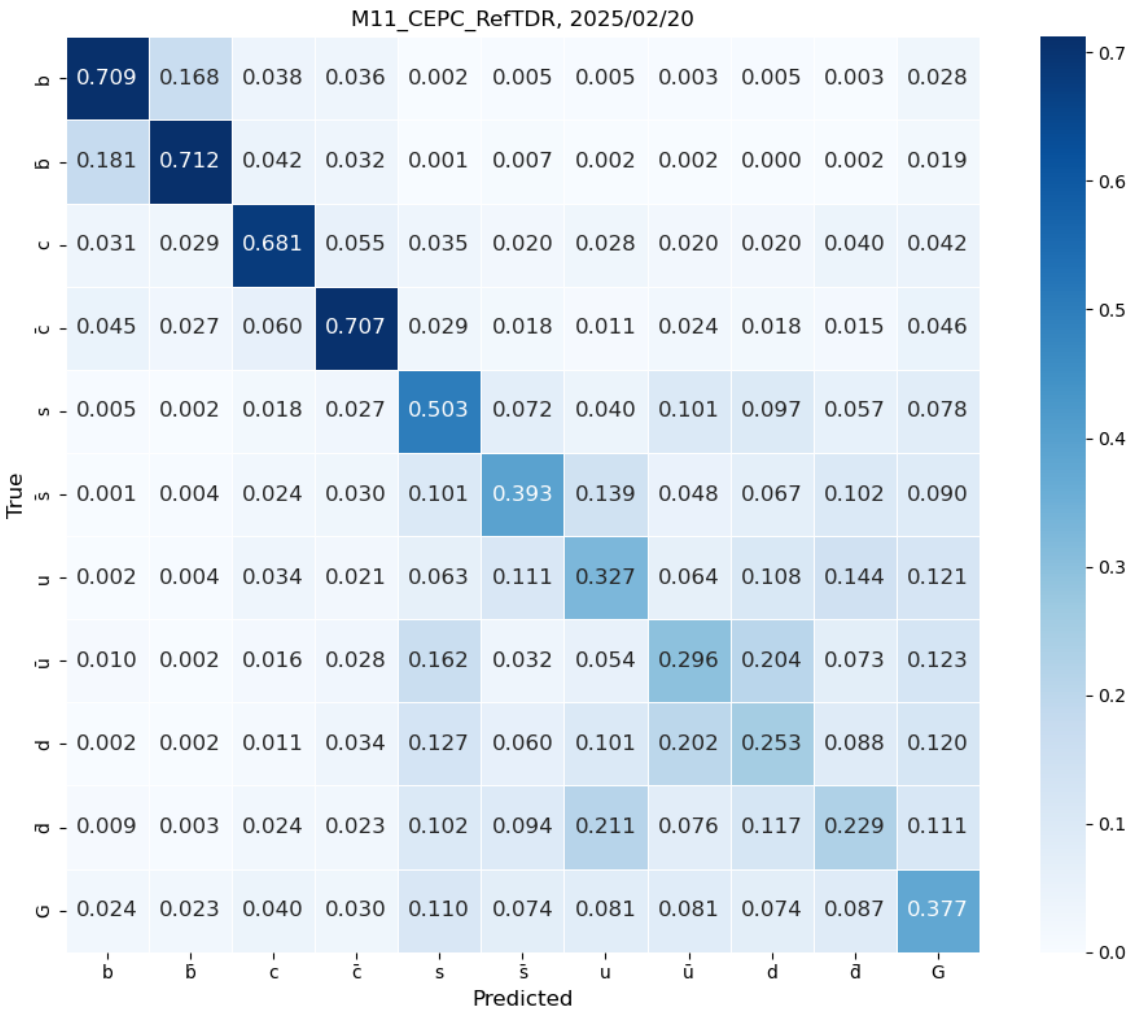
Kaili Zhang

IHEP

zhangkl@ihep.ac.cn

Current Best JOI

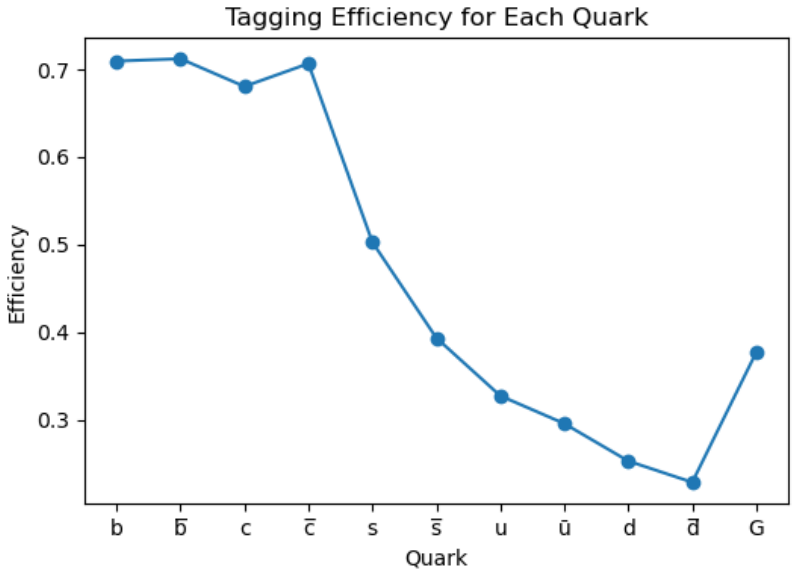
0.47151

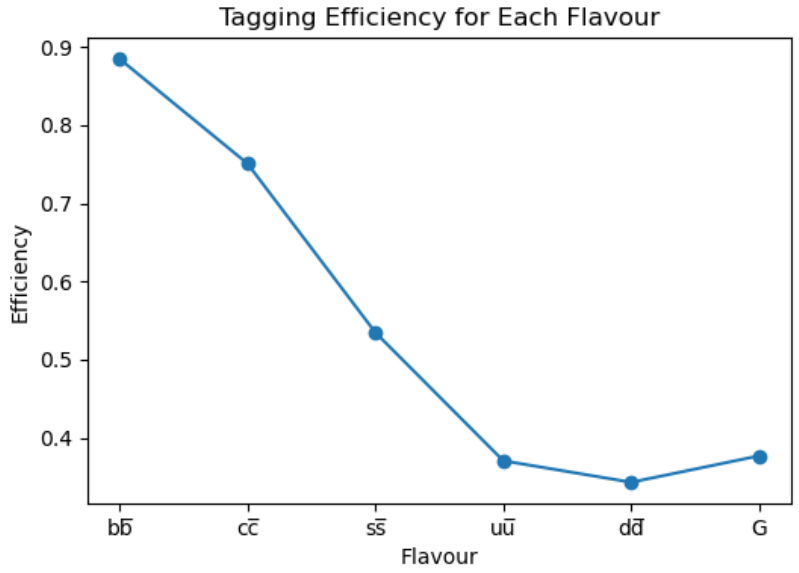


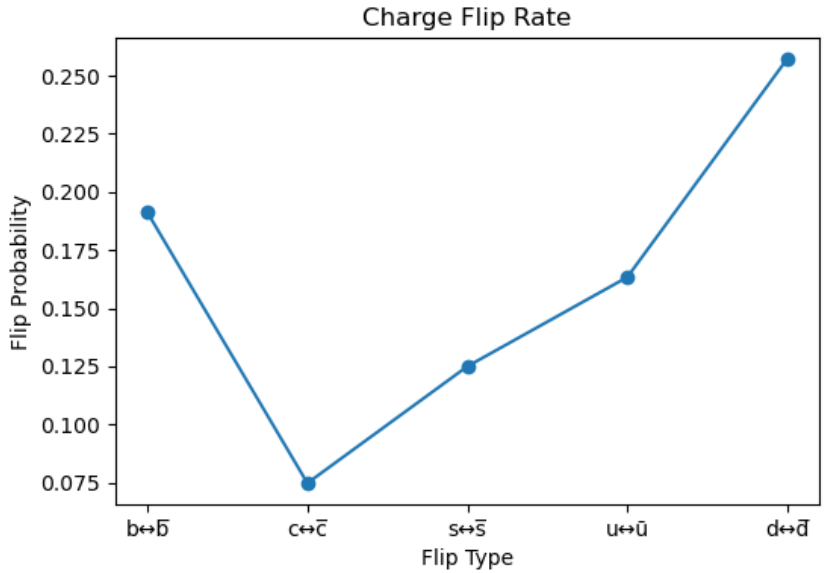
- This result is done in:
- CEPCSW 25.1.1
- Length: 40.
- Total sample size: 980W
- Train: Test: Validation=8:1:1

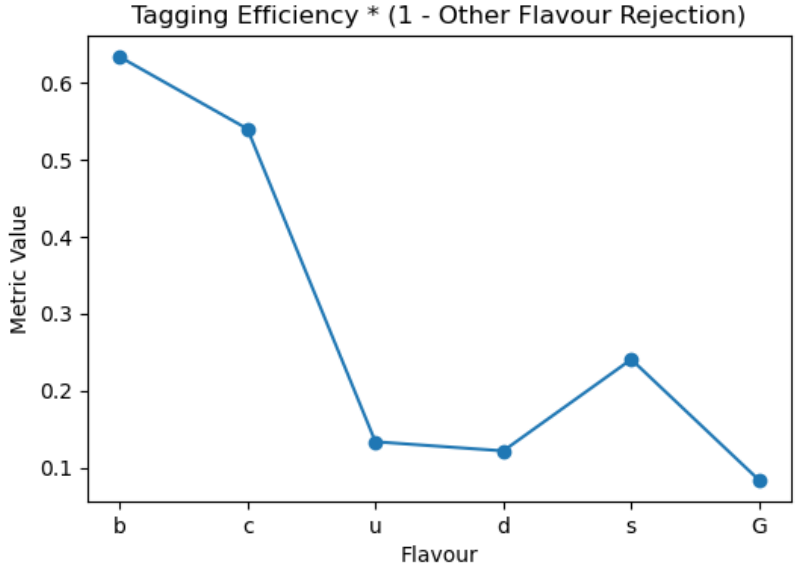
- With overtraining issue. (Epoch #2 gives best result. Following worse;)
- Under tuning. But with ~89% btagging eff.

Current JOI


Tagging Efficiency for Each Quark
Efficiency
0.7
0.6
0.5
0.4
0.3
b
b̄
c
c̄
s
s̄
u
ū
d
d̄
G
Quark


Tagging Efficiency for Each Flavour
Efficiency
0.9
0.8
0.7
0.6
0.5
0.4
bb̄
cc̄
ss̄
uū
dd̄
G
Flavour


Charge Flip Rate
Flip Probability
0.250
0.225
0.200
0.175
0.150
0.125
0.100
0.075
b↔b̄
c↔c̄
s↔s̄
u↔ū
d↔d̄
Flip Type


Tagging Efficiency * (1 - Other Flavour Rejection)
Metric Value
0.6
0.5
0.4
0.3
0.2
0.1
b
c
u
d
s
G
Flavour

Scaling Law

- Trying use new small size sample for training/testing. 420000 : 50000

```
[2025-02-24 02:35:50,058] INFO: Processed 420000 entries in total (avg. speed 634.2 entries/s)
[2025-02-24 02:35:50,058] INFO: Train AvgLoss: 0.57868, AvgAcc: 0.79346
[2025-02-24 02:35:50,059] INFO: Train class distribution:
    [(0, 38122), (1, 38131), (2, 38500), (3, 38496), (4, 38507), (5, 38489), (6, 38312), (7, 38304), (8, 37711), (9, 37819), (10, 37609)]
```

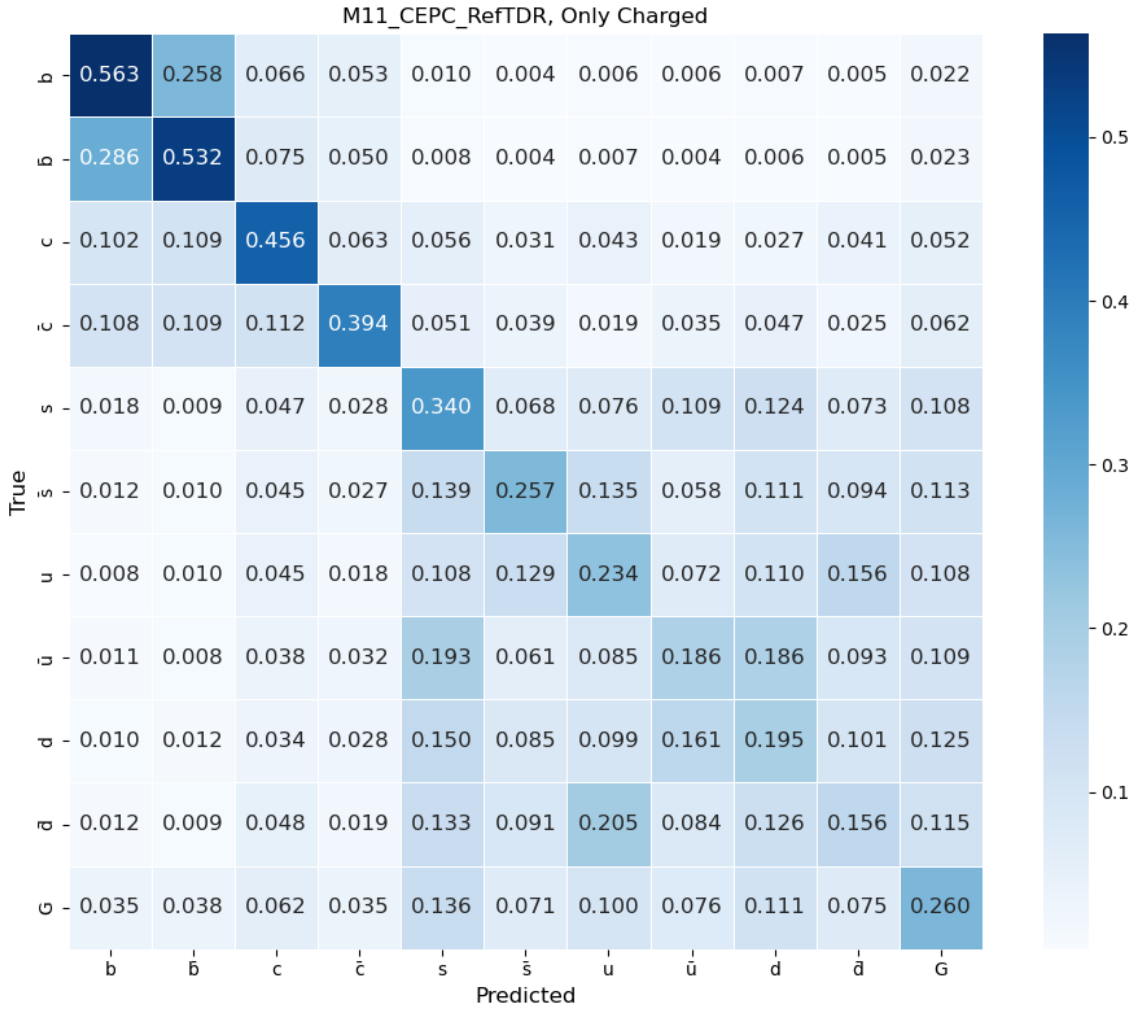
```
[2025-02-24 02:47:02,547] INFO: Processed 50000 entries in total (avg. speed 2342.1 entries/s)
[2025-02-24 02:47:02,547] INFO: Evaluation class distribution:
    [(0, 4452), (1, 4458), (2, 4447), (3, 4472), (4, 4357), (5, 4360), (6, 4451), (7, 4470), (8, 4398), (9, 4410), (10, 5725)]
```

- 1/20 size of full set.
  - Even though small amount size giving enough result.
  - Overtraining issue is studying with small size sample for rapid response. (~10mins for 1 epoch).

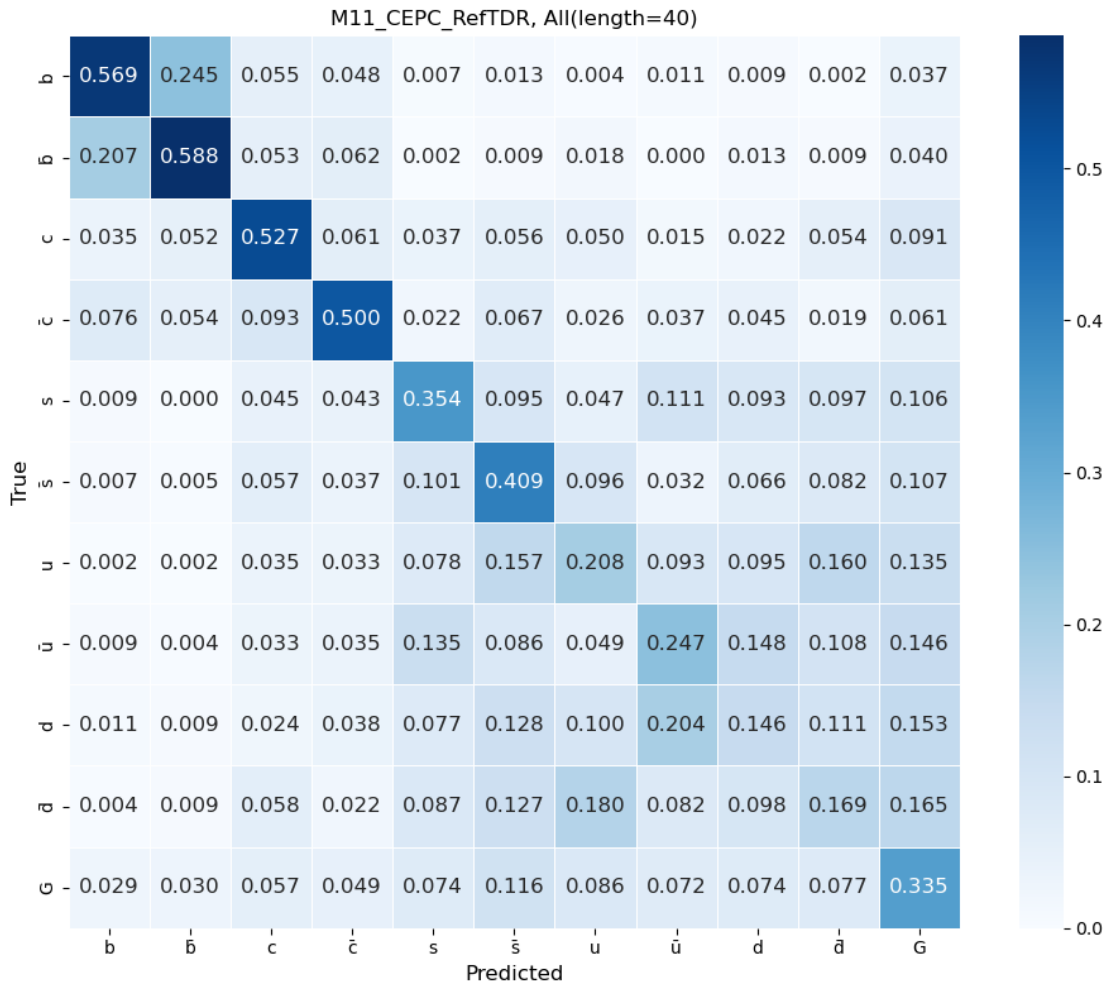
TDR Charged/Neutral

With small 1/20 set.

0.32314



0.36847

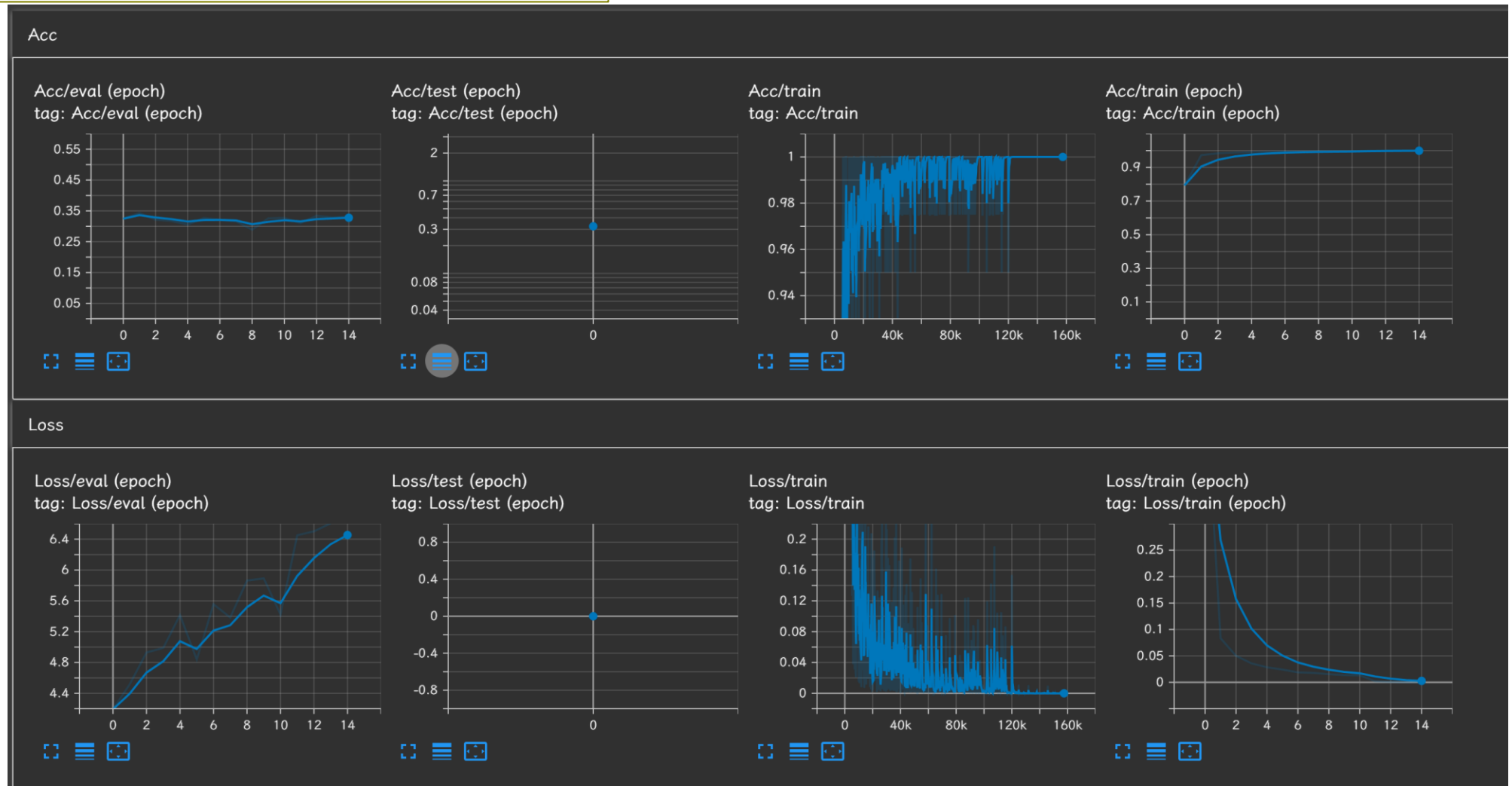


Using mask=pfo_withtrack;

Overtraining test on small set

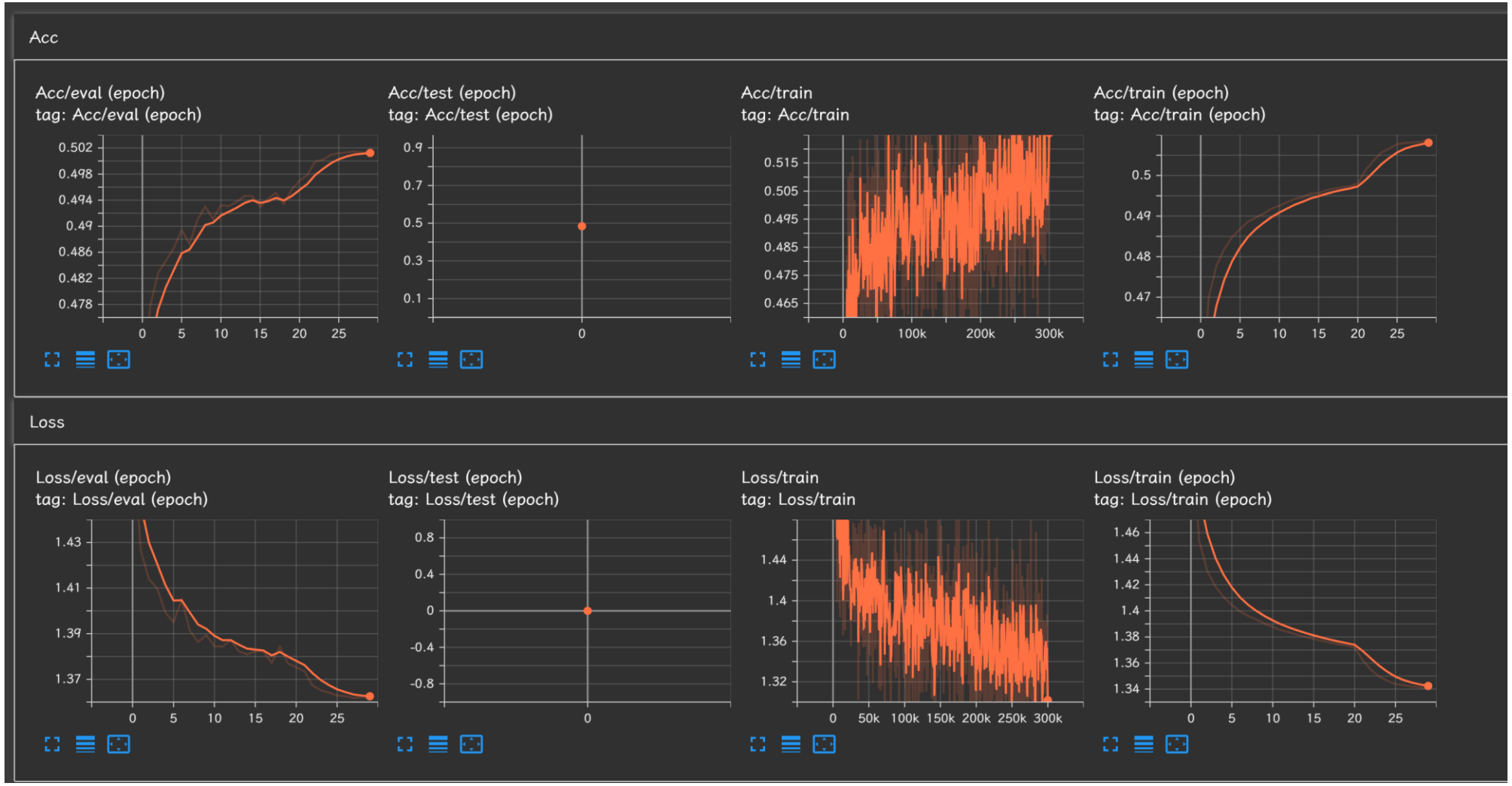
Tensorboard: Corresponds to only charged tracks.
Loss ++, showing overtraining.

Flat accuracy, training accucacy-1. Didn't learn knowledge in following studying.



CDR Tensorboard

One Typical success run like:

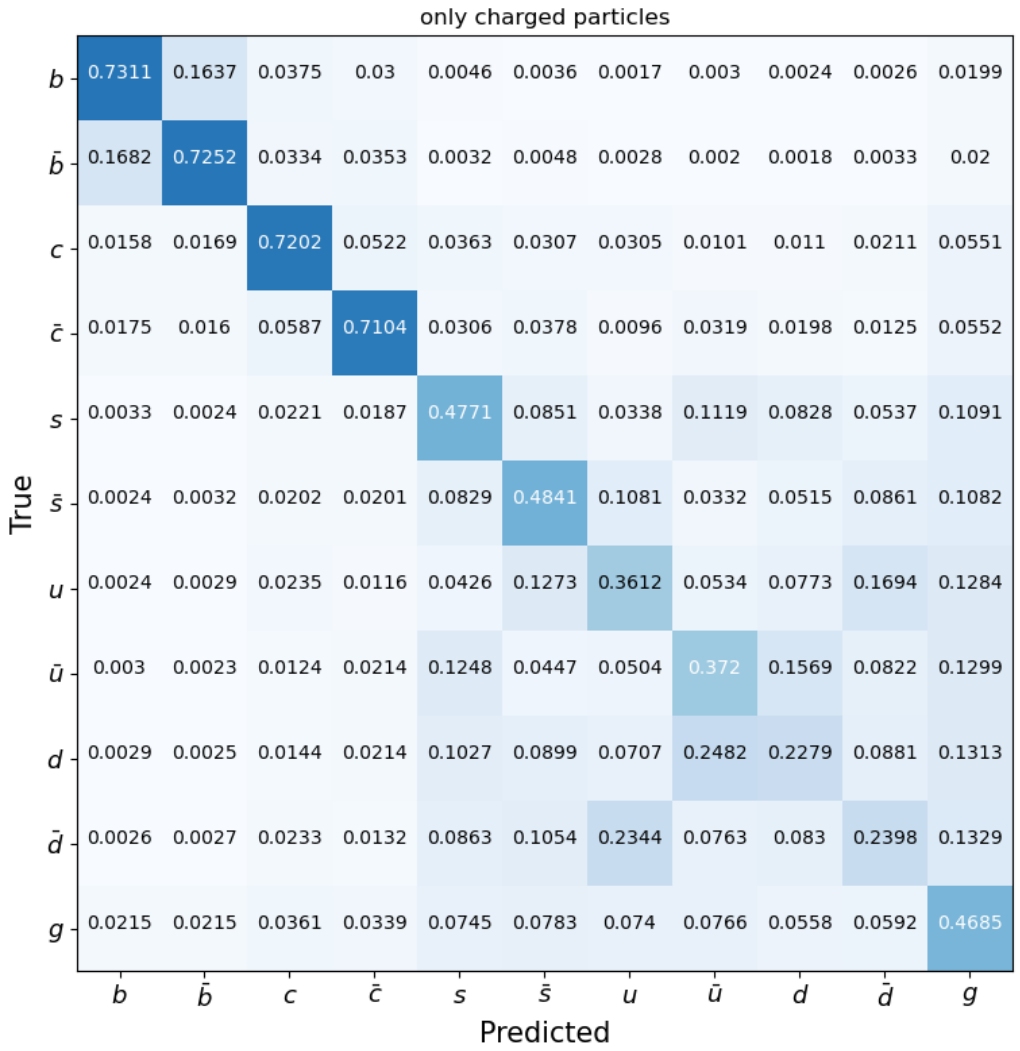


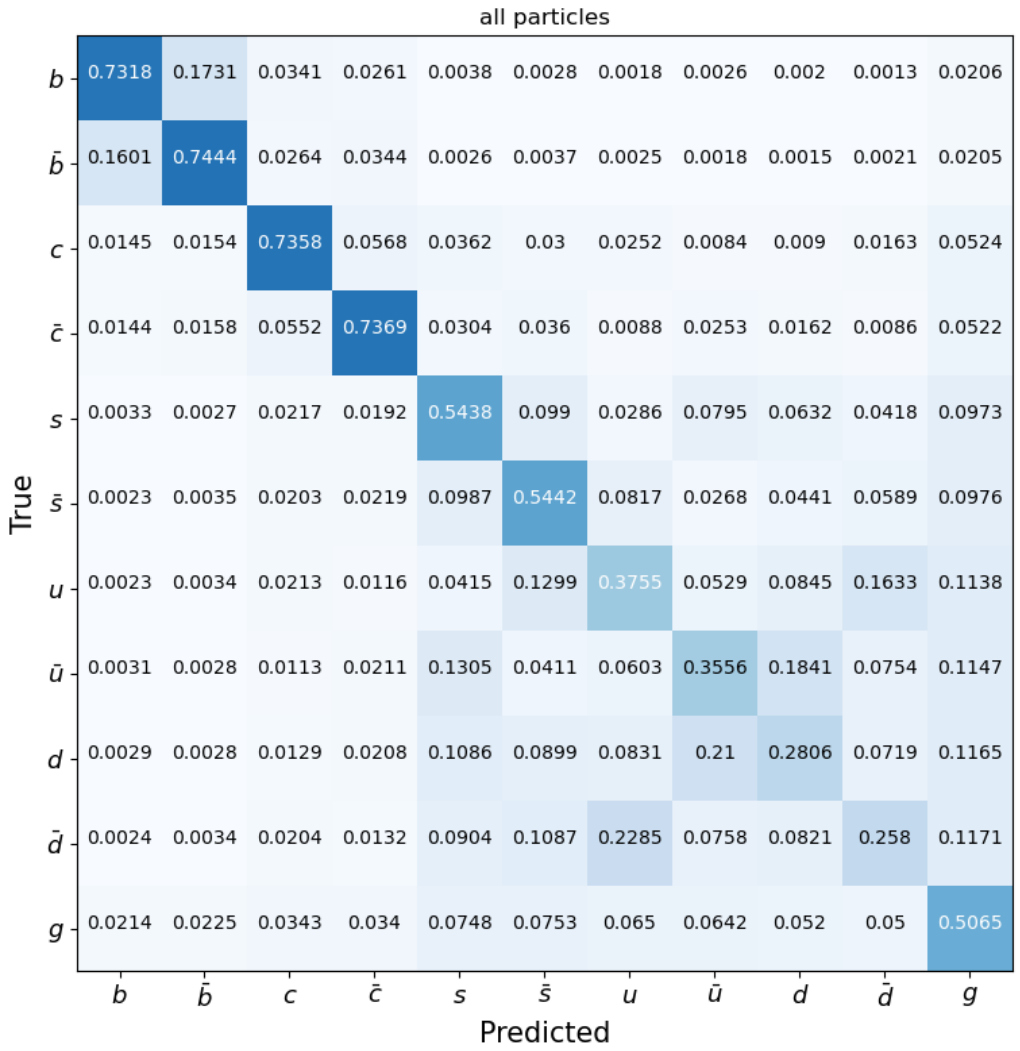
CDR Charged/Neutral

Only Charged:0.50146

5% worse

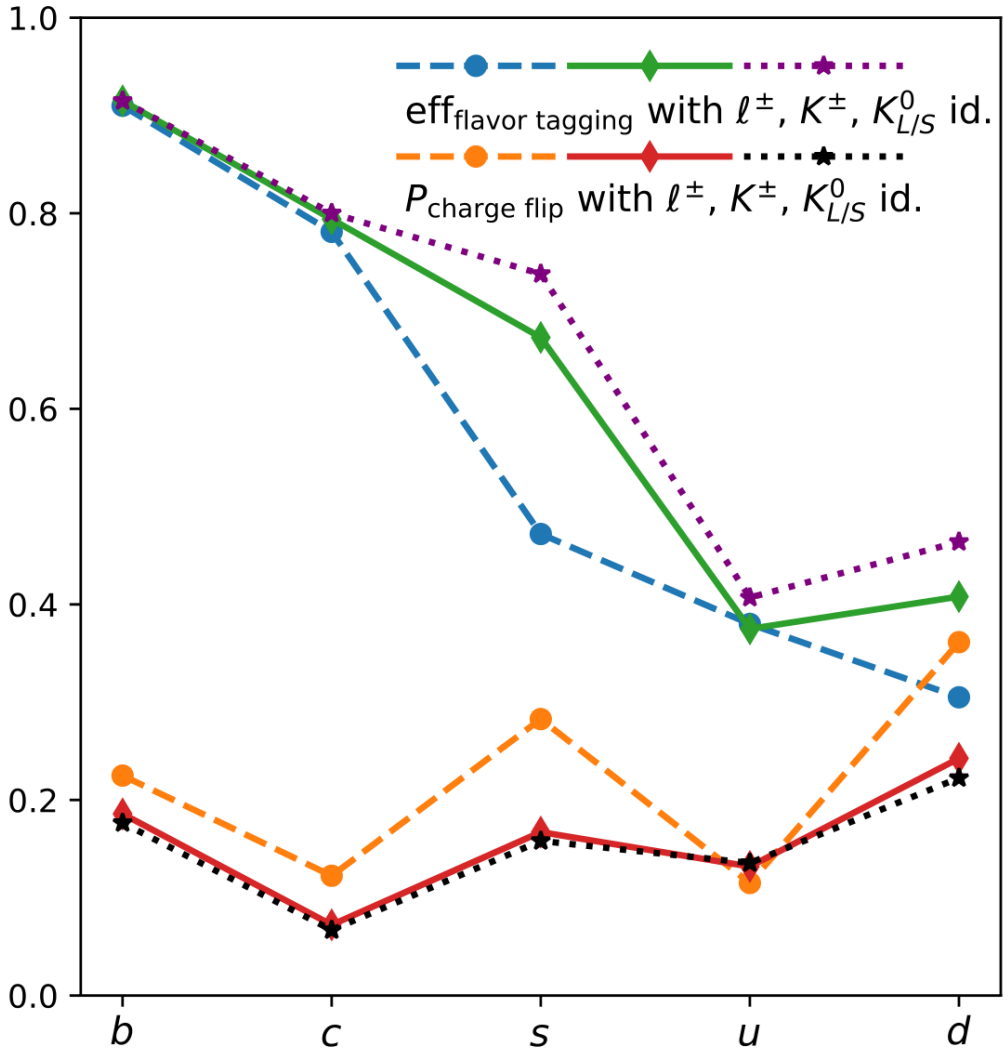
All: 0.52836

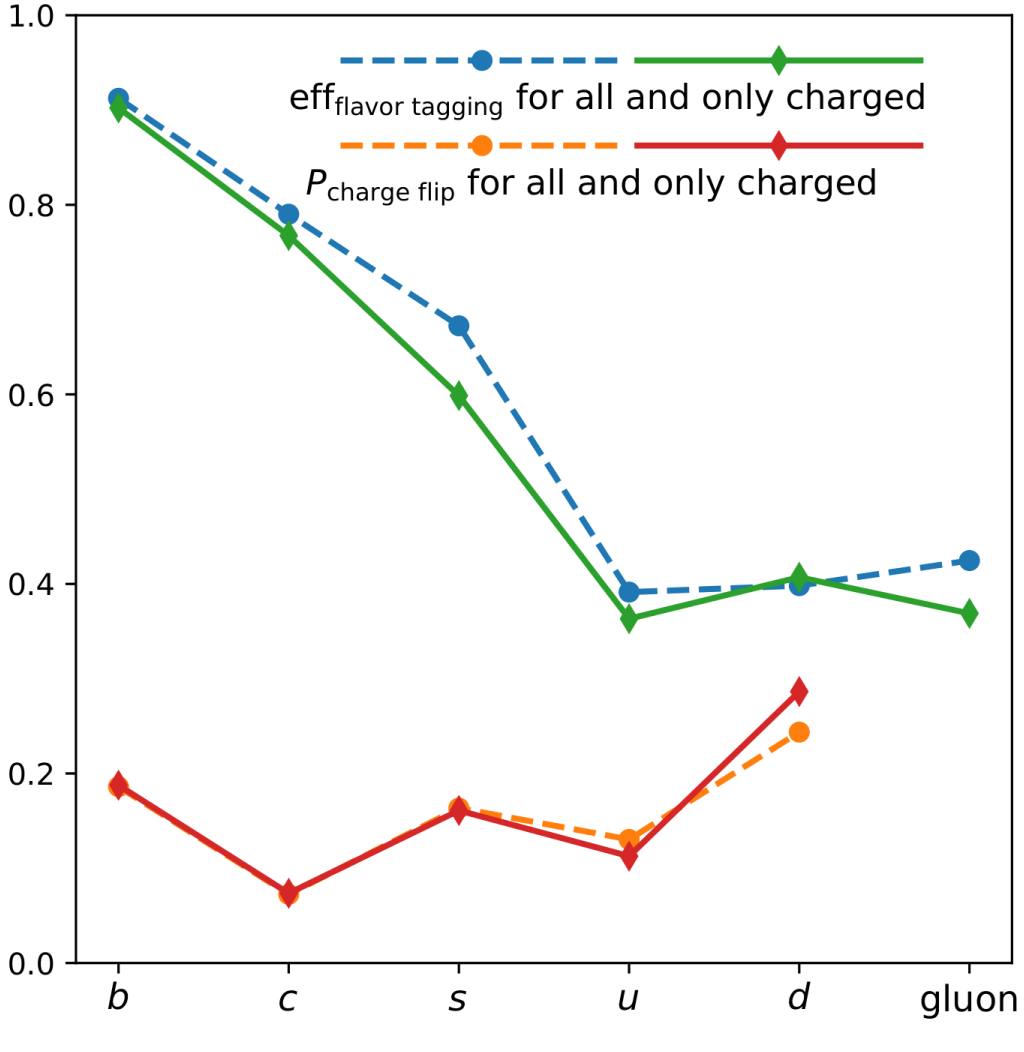




CDR Curve

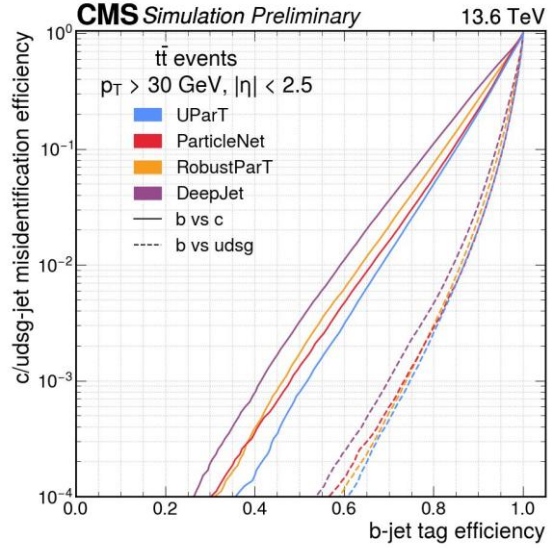
CDR flavor tagging eff use test set max(score_b+score_bbar, score_c+score_cbar……)





WP/CMS Result

UParT: CMS-DP-2024/066



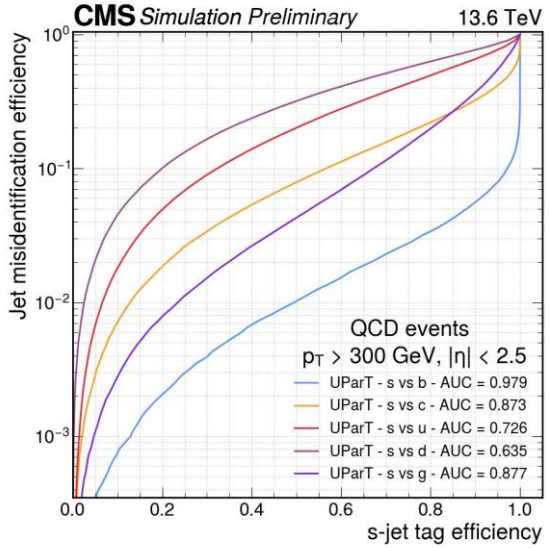
$$\text{B vs All} = \frac{prob(b)}{prob(b) + prob(c) + prob(udsg)}$$

$$\text{B vs L} = \frac{prob(b)}{prob(b) + prob(udsg)}$$

$$\text{B vs C} = \frac{prob(b)}{prob(b) + prob(c)}$$

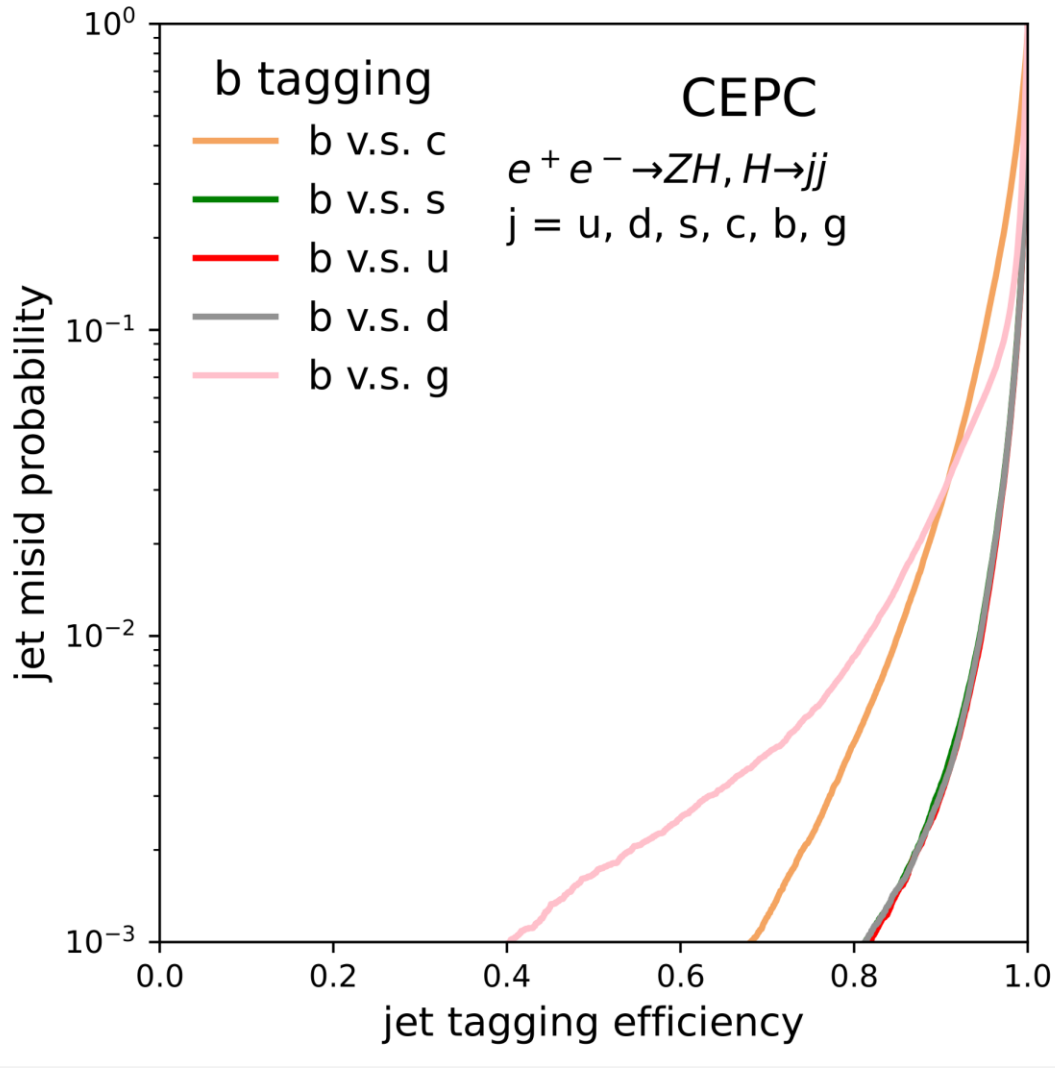
$$\text{B vs All weighted} = \frac{prob(b)}{k_c \cdot prob(c) + (1 - k_c) \cdot prob(udsg)}$$

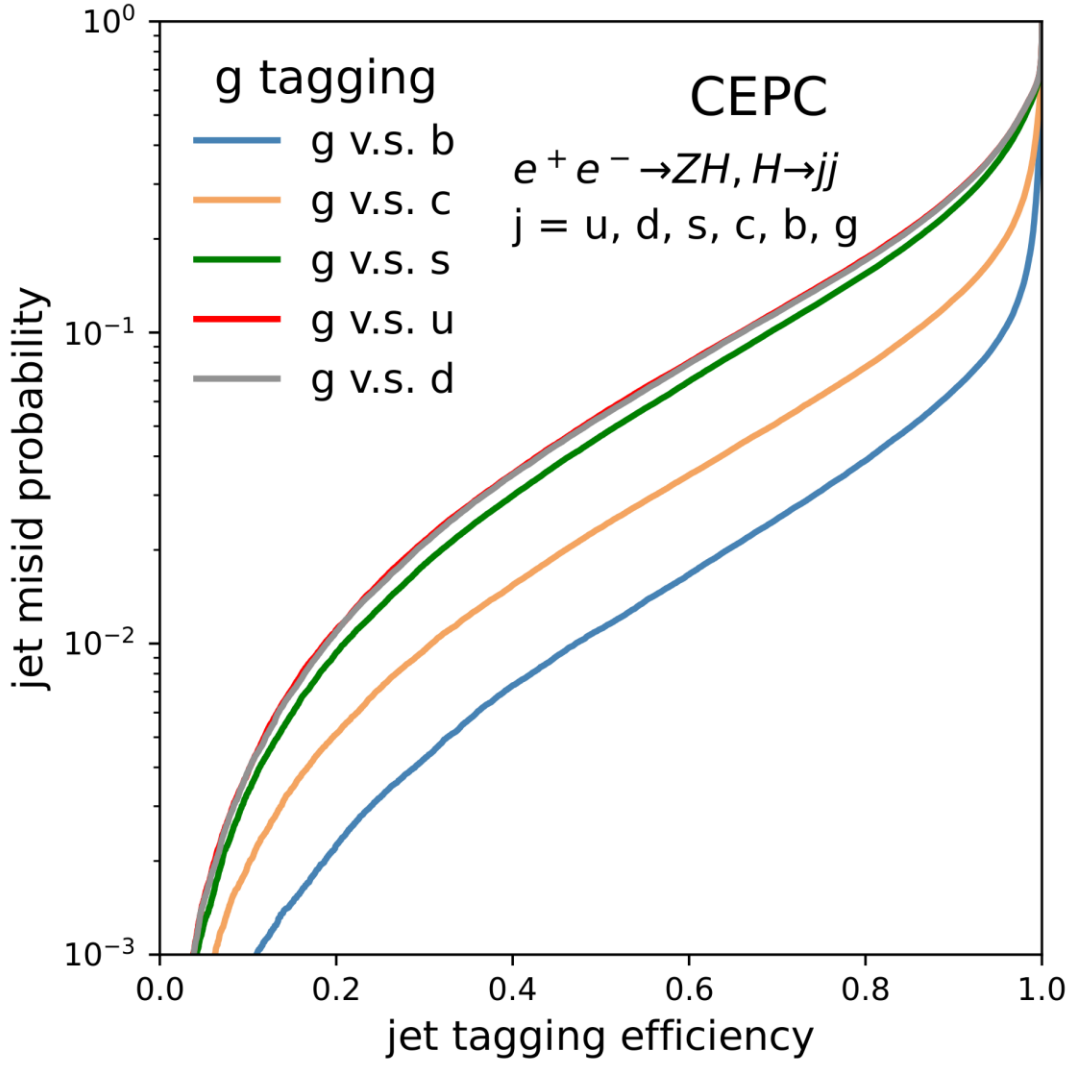
- For different scenario/hypothesis, different score/WP given.
- CMS use ttbar/QCD events with different tagger. (different bcdusg ratio).
- CDR: Higgs only (b dominant)
- TDR: uniform bcdusg? Or different for individual?



CDR “WP” set

This curve is done in Higgs jets only (b dominant).

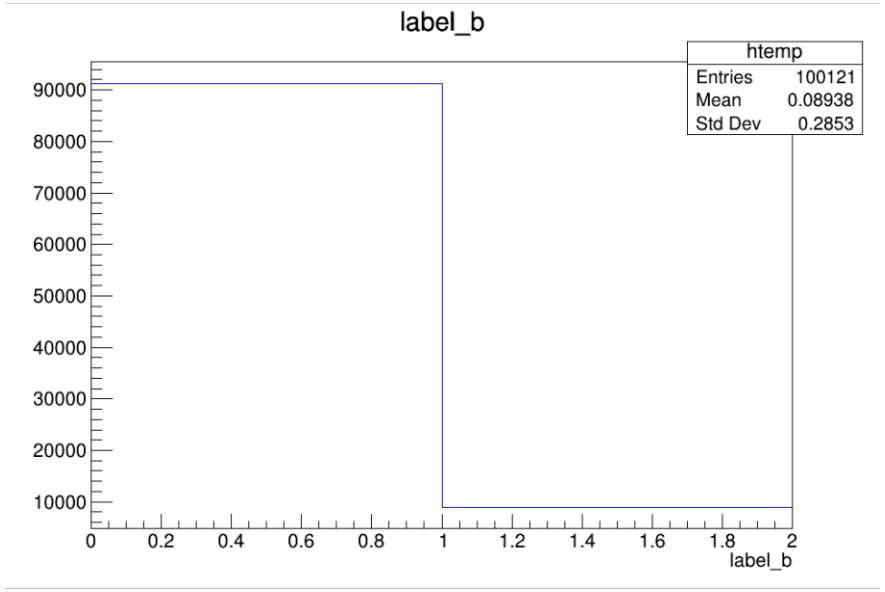


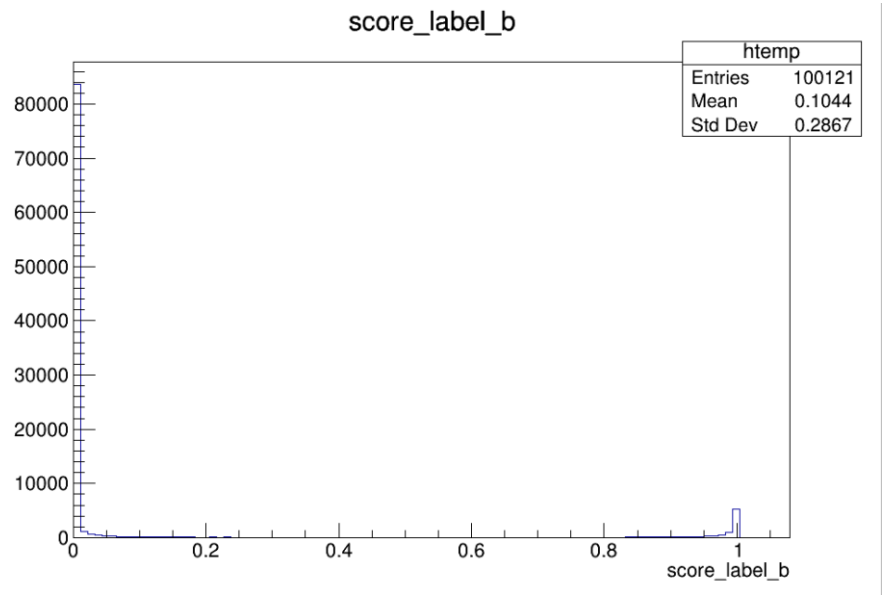


TDR Application

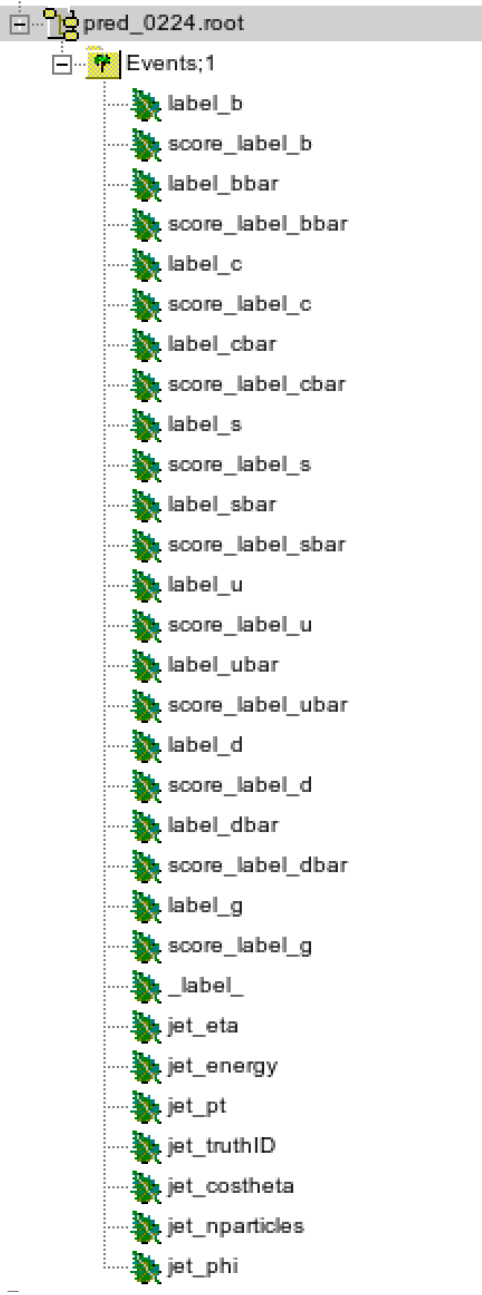
/cefs/higgs/zhangkl/AI/JOI/higgs/pred_0224.root

- For test sample 100k, category and score given.





- With onnx model, for each candidate, score given.
- B tagging score can be : score_b/(score_b+score_c+……11)



Calculation

- As example, taking b_score>0.5 = b jet.

1. Eff_btagging= (N_truthb, with bscore>0.5)/(N_truthb, all)
2. Purity_btagging= (N_truthb, with bscore>0.5)/(N_all, withscore>0.5)
3. Mis_tagging_bvc= (N_truthc, with bscore>0.5)/(N_truthc, all)
4. Mis_tagging_bvall= (N_truth_others, with bscore>0.5)/(N_truth_others, all)

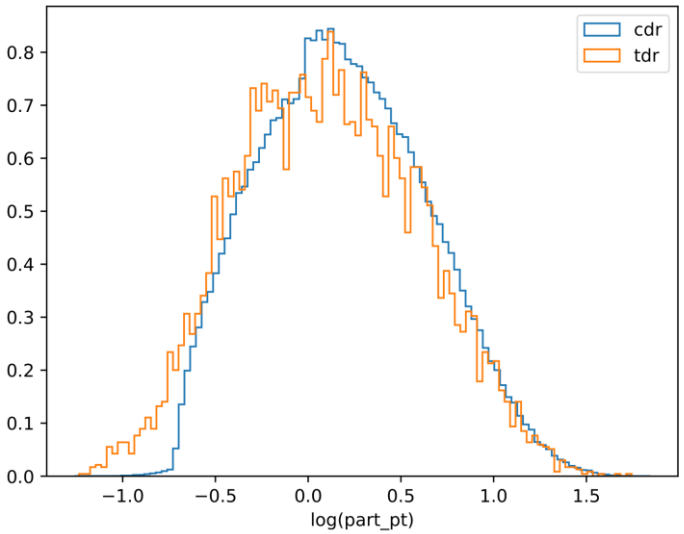
Use 1:3, 1:4 as ROC curve;

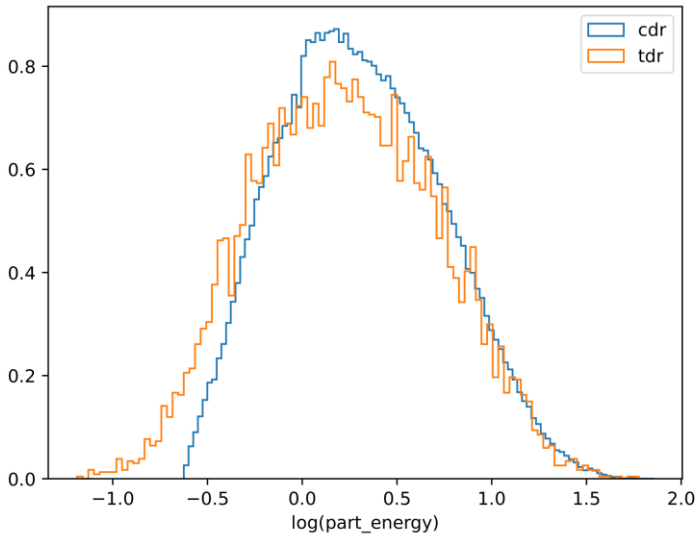
Maximize (1*2) for best cut point.

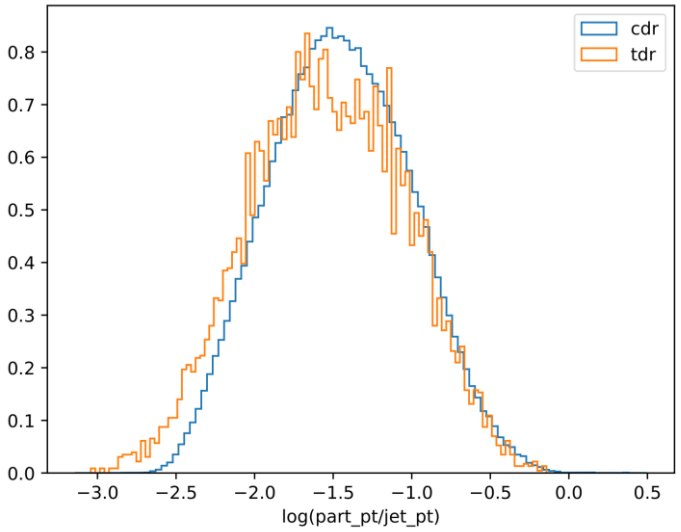
Based on hypothesis of mixing ratio. Need to determine.

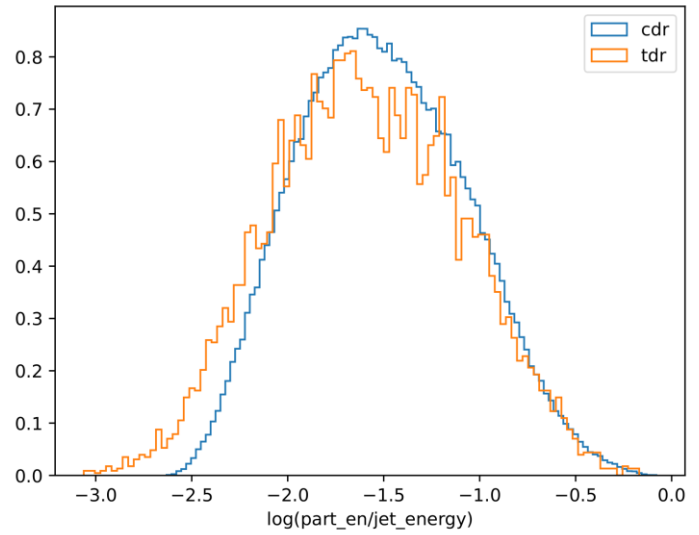
Input variable distribution CDR&TDR

@Yongfeng



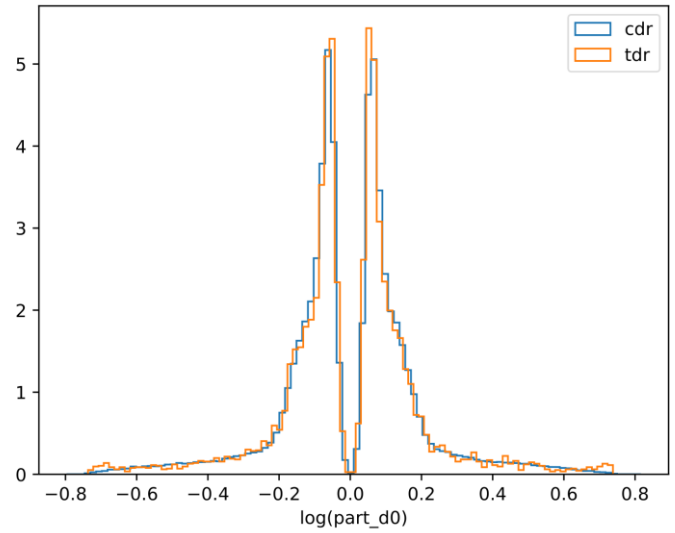


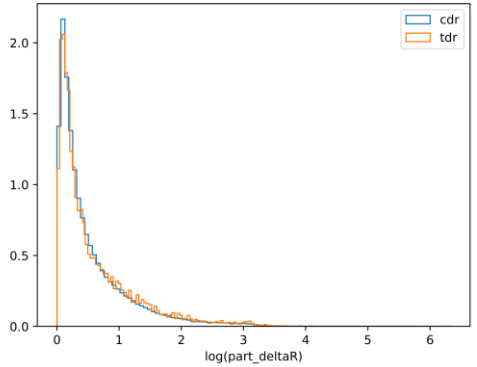


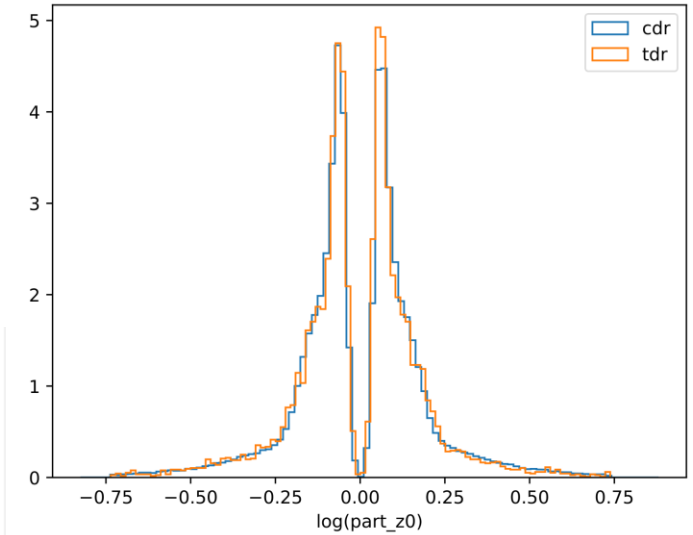


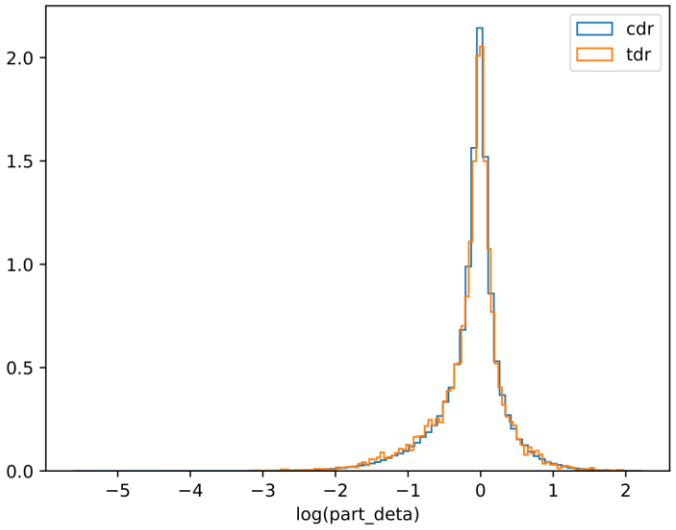
Input variable distribution CDR&TDR

@Yongfeng

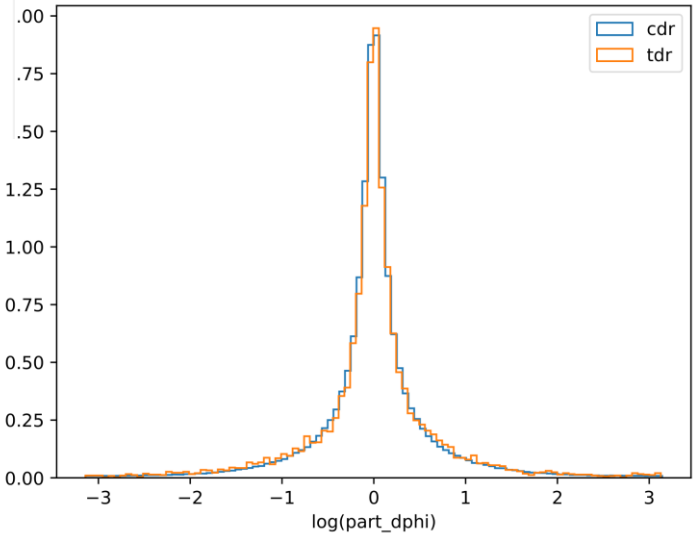




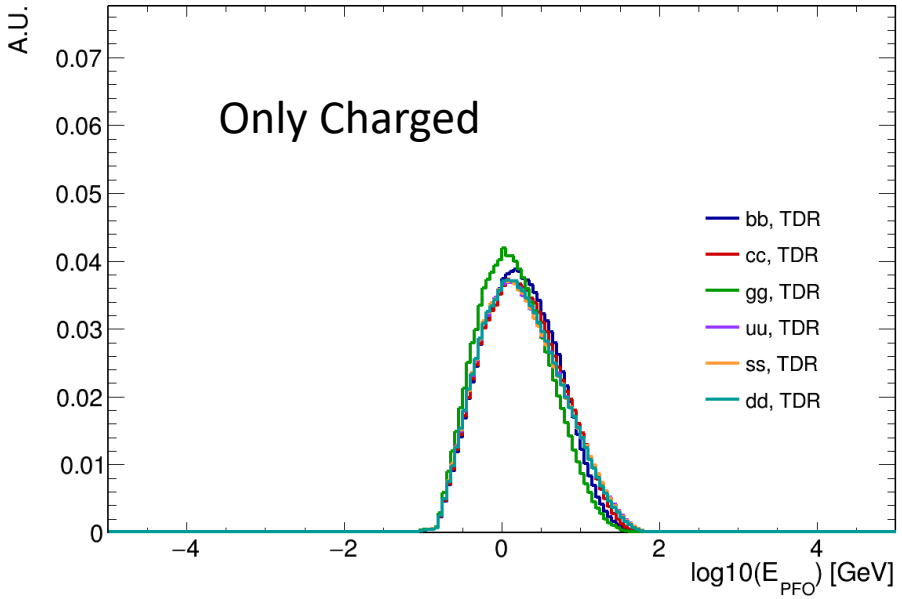


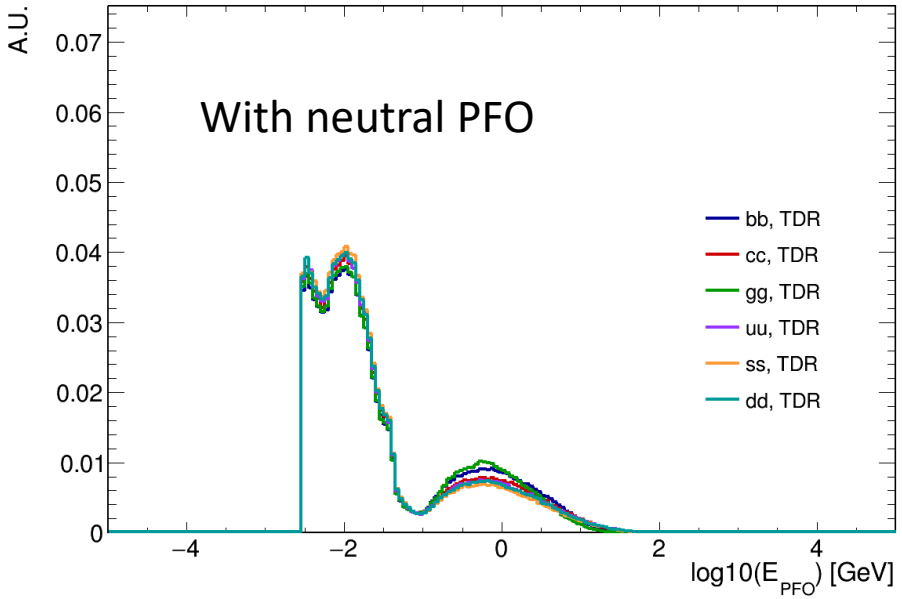


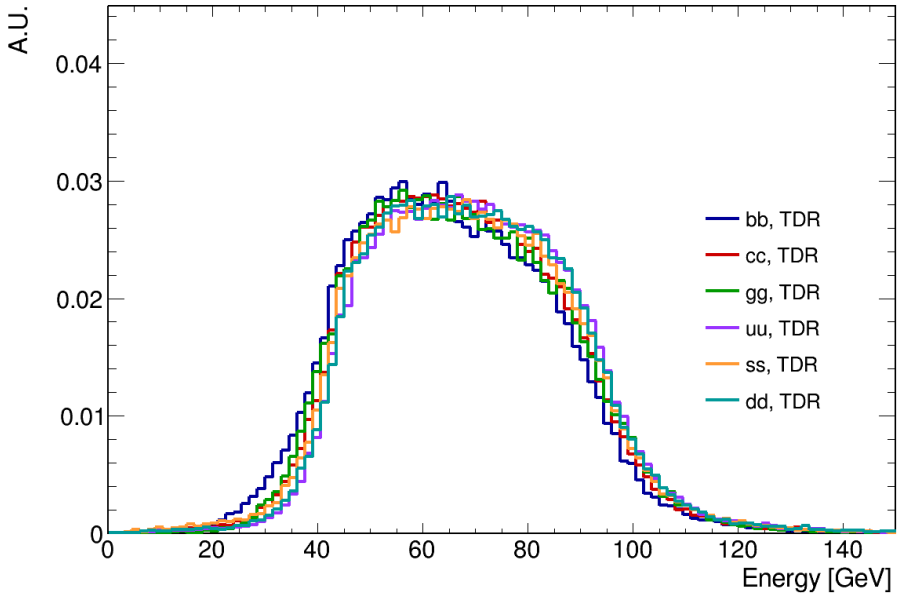
Checked by yongfeng:
for most variables,
Consistent between
CDR&TDR.



Input variable distribution







TDR: /cefs/higgs/zhangkl/JOI/datasets0226
CDR: /cefs/higgs/zhangkl/AI/datasets/zhuyf_higgs125

Truth Matching Eff

/cefs/higgs/zhangkl/CEPCSW/Analysis/JetOrigin/src

- Current CEPCSW do not have reliable truth link between reco PFO and mc truth.
- Use DeltaR and DeltaE to match.
  - Currently, for each charged track, eff >90%
- Fail case:
  - All stable truth particle already matched with additional track;
  - Track with small energy (like, <0.5GeV)
  - Other under tuning issue.

Truth Matching Method

DeltaE<0.2 &&
(DR<0.2 && Dpx+Dpy+Dpz)<0.3||(Dpx+Dpy+Dpz)<0.12 ||(DR<0.12)

```
TLorentzVector T_temp;
T_temp.SetXYZM(T_part.Px(), T_part.Py(), T_part.Pz(), MC_mass );


TLorentzVector T_MC;
T_MC.SetPxPyPzE(Gen.getMomentum()[0], Gen.getMomentum()[1], Gen.getMomentum()[2], Gen.getEnergy());

double PFO_Truth_DR      = T_temp.DeltaR( T_MC );
// double mini_PFO_Truth_DR = 0.2;
// double mini_PFO_Truth_DE = 0.2;

double residualE= abs((T_temp.E() - T_MC.E())/T_MC.E());
double residual=
abs((T_temp.Px() - T_MC.Px())/ T_MC.E())+
abs((T_temp.Py() - T_MC.Py())/ T_MC.E())+
abs((T_temp.Pz() - T_MC.Pz())/ T_MC.E());

if ( residualE<0.2 && ((PFO_Truth_DR <  0.2 && residual < 0.3) || (PFO_Truth_DR <  0.12 ) || (residual < 0.12 )) ) /
```

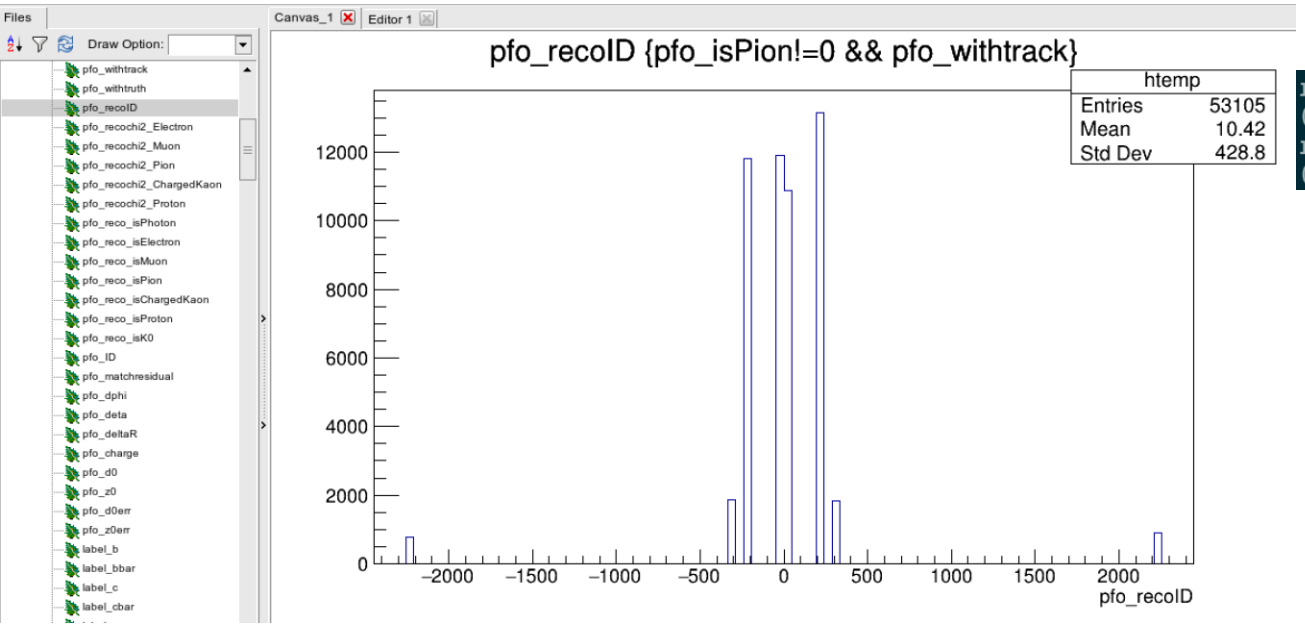
```
JetOrigin           INFO  Matched with truth particle Energy: 0.321898: 0.310204 MCIndex: 107
JetOrigin           INFO  (reco-truth)/truth E: -0.0363294
JetOrigin           INFO  (reco-truth)/truth Pt: -0.0450998
JetOrigin           INFO  (reco-truth)/truth Px: -0.0438147
JetOrigin           INFO  (reco-truth)/truth Py: -0.106302
JetOrigin           INFO  (reco-truth)/truth Pz: -0.0448689
JetOrigin           INFO  (reco-truth)/truth Eta: -0.000167079 (reco-truth)/truth Phi: 0.00943658 (reco-truth)/truth DeltaR: 0.00943806
JetOrigin           INFO  PFO Px: 0.198272 PFO Py: -0.0273029 PFO Pz: -0.191523 PFO E: 0.310204
JetOrigin           INFO  E_Ecal: 0        E_Hcal: 0        ESize: 0        HSize: 0
JetOrigin           INFO  D0:    -0.0924448 Z0:    -0.0011889 D0err: 0.00154263 Z0err: 0.00247036
JetOrigin           INFO  Dphi:  -0.52123 Deta:  -0.0987253 DeltaR: 0.530497
JetOrigin           INFO  PID:   5        Charge: 1
JetOrigin           INFO  MResidual: 0.253956
JetOrigin           INFO  Track: 1        Truth: 1
JetOrigin           INFO  S E_Ecal: -inf     S E_Hcal: -inf     S ESize: 0        S HSize: 0
JetOrigin           INFO  C Dphi: -3.60986 C E:   -1.85753 C Pt:  -3.10872 C relE: -9.76728 C relPt: -10.053
JetOrigin           INFO  C Z0:  -1.03412 C D0:  -2.92486 C D0err: -1.40587 C Z0err: -1.30362
JetOrigin           INFO  id: 620d570 107
PDG : 211
generatorStatus : 1
simulatorStatus : 0
charge : 1
time : 0
mass : 0.13957
vertex : 0 0 0
endpoint : 69.2728 -21.2372 -69.6248
momentum : 0.207358 -0.0305505 -0.20052
momentumAtEndpoint : 0 -0 -0
spin : 0 0 0
colorFlow : 0 0
parents : c68708c168
daughters :
```

For most matchings,
D(pxpypzE) <0.1 is efficient.

Reco id information First look

https://code.ihep.ac.cn/cepc/CEPCSW/-/blob/master/Reconstruction/ParticleID/src/FinalPIDAlg.cpp?ref_type=heads

Reco information available.
TOF+TPC e/mu/K/Pi/P chi2 and PID stored.



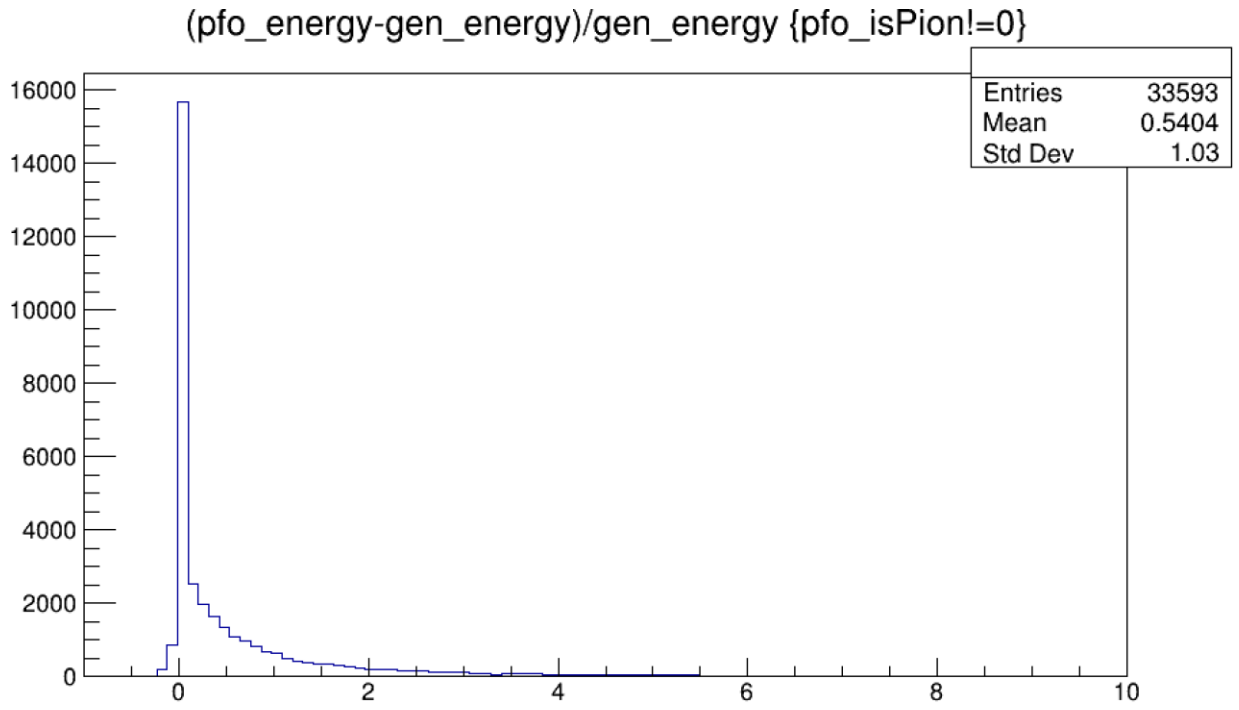

```
root [20] tree→Draw("pfo_recoID","pfo_isPion==1 && pfo_withtrack && pfo_recoID==211")
(long long) 13142
root [21] tree→Draw("pfo_recoID","pfo_isPion==1 && pfo_withtrack")
(long long) 27774
```

```
root [23] tree→Draw("pfo_recoID","pfo_withtrack && pfo_recoID==211")
(long long) 13234
```

In this dd.root,
Eff(Pion+)=47.31%
Purity(Pion+)=99.3%.

Considering many truth Pion with E<1 GeV, this result reasonable.

Kinematic Distribution for reco-truth Pion

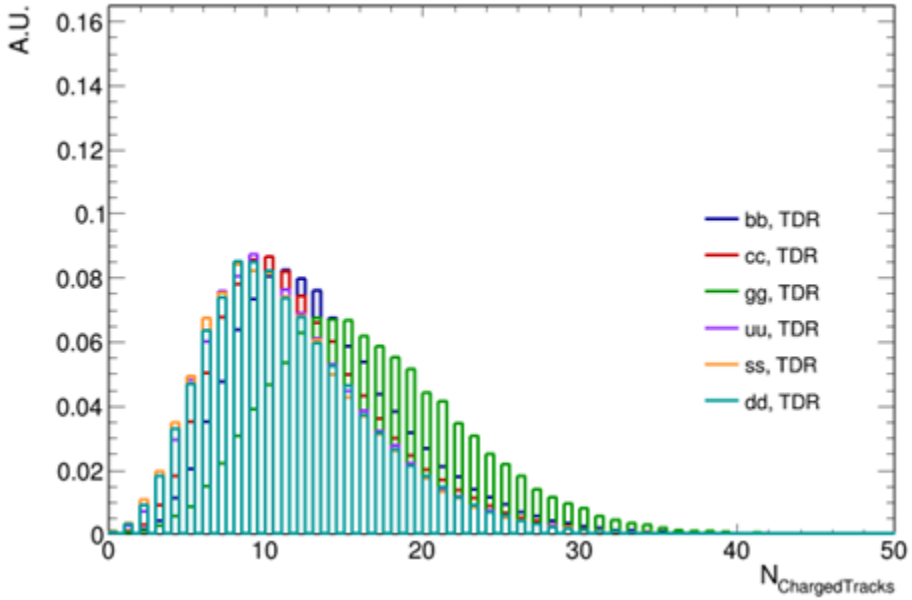


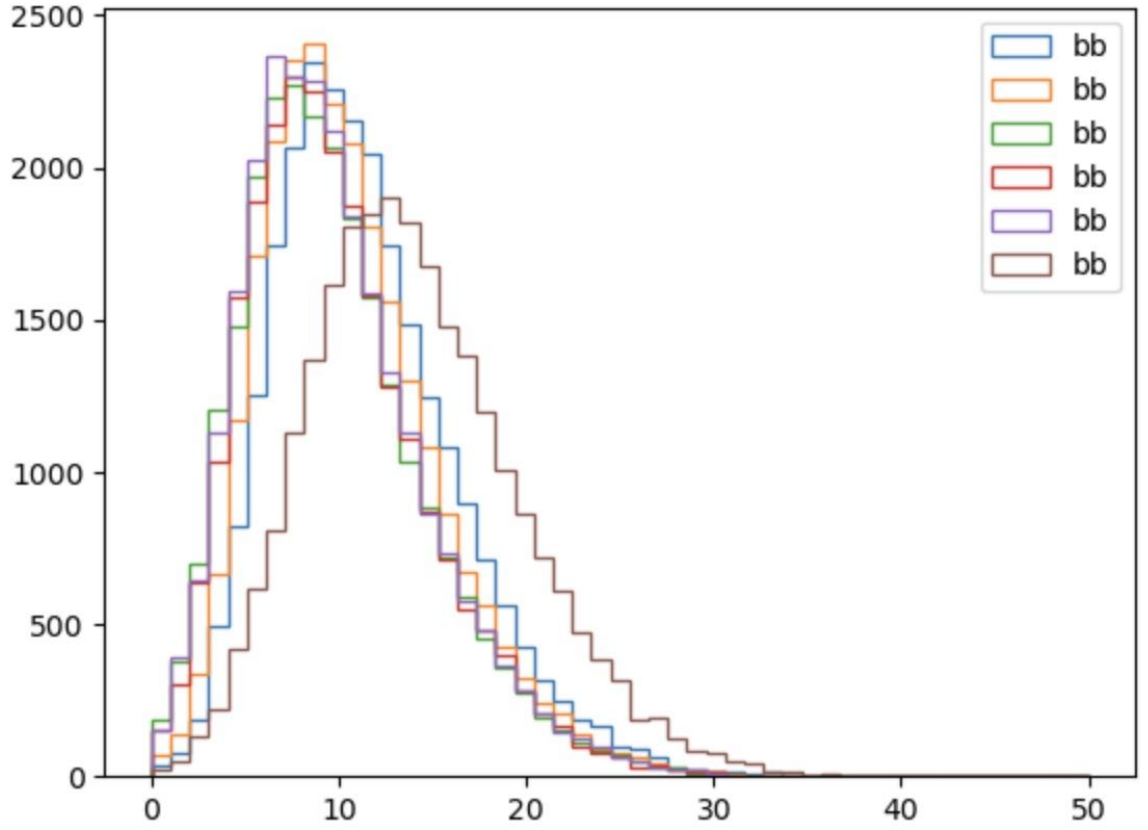
- Energy difference for successful matched Pions.
- PFO energy already set Mass:

```
TLorentzVector T_temp;
T_temp.SetXYZM(T_part.Px(), T_part.Py(), T_part.Pz(), MC_mass );
```

backup

N_tracks CDR&TDR





Machine Learnings on Jets

- P-CNN
  - https://scipost.org/10.21468/SciPostPhys.7.1.014
- Particle Flow Network
  - https://arxiv.org/abs/1810.05165
  - CEPC@Xiaotian : https://arxiv.org/abs/2410.04465v2
- LundNet
  - https://doi.org/10.1007/jhep03(2021)052
- ParticleNet
  - Arxiv:1902.08570
  - https://github.com/hqucms/ParticleNet

ParticleTransformer

- https://arxiv.org/abs/2202.03772
- https://github.com/jet-universe/particle_transformer
- Platforms: https://github.com/hqucms/weaver-core
- Application on CEPC: 2309.13231 , PRL 132, 221802 (2024)
- Tutorial on CEPC: https://github.com/ZHUYFgit/CEPC-Jet-Origin-Identification
- Inputs from CEPCsoft: /cefs/higgs/zhangkl/AI/datasets
- Inputs from LHC, JetClass: /cefs/higgs/zhangkl/AI/jetclass
- Require higgsgpu group. Request on https://ccsinfo.ihep.ac.cn/
- Follow the tutorial, build the env if you are interested.

ParticleTransformer @ CEPC

https://github.com/ZHUYFgit/CEPC-Jet-Origin-Identification

- Variable list in M11origin.cc
  - Under development to CEPCSW
  - Unit as one jet: 4 momentum, M11 id information……
- Train in Weaver: JetClass_full.yaml
- Submit jobs on IHEP: train_JetClass.sh
- Output: Pred.root: Label and score for each jets.
- Application: onnx format

Inputs for JOI

/cefs/higgs/zhangkl/CEPCSW/Analysis/JetOrigin/src

- Jet->Event;
- PFO->Component;
- Length: 200
- Label: M11

- Current training use truth PID information, in application reco PID will be used.

Type	Var	Comment
PFO point distance	$\Delta\phi(pfo, Jet)$	Delta Phi, pfo to jet
	$\Delta\eta(pfo, Jet)$	Delta Eta, pfo to jet
PFO Vector variable	(px, py, pz, E)	4 momentum of PFO
PFO feature variable	P_t^{PFO}, $\log\frac{P_t^{PFO}}{P_t^{jet}}$	Pfo pt and relative pt
	E_t^{PFO}, $\log\frac{E_t^{PFO}}{E_t^{jet}}$	Pfo E and relative E
	$\Delta R(pfo, Jet)$	Delta R, pfo to jet
	N_charge, N_chargeflip	Charge of PFO
	D0, Z0, D0err, Z0err	(if with track) impact parameters
	N_Ecluster, N_Hcluster	
	E_ecal, E_hcal	
	PID	Truth PID type

Variable convention

- Feature variable, Transformer prefer normal distribution with mean ~0, range (-1,1) with cut edge maximum(-5,5).
- (4-momentum vector variable not included)
- Normalization functions like Tanh() used.
-