



PrimeTagSvc

Jets, samples and Wednesday working meeting Kaili Zhang <u>zhangkl@ihep.ac.cn</u>

Latest: v2.2



Interface for output;

std::vector<float> GetProb() const override; int get_type_M11() const override; int get_type_M6() const override;

// M11 Boolean functions (11 total)
bool is_b_quark() const override;
bool is_b_bar_quark() const override;
bool is_c_quark() const override;
bool is_s_quark() const override;
bool is_s_bar_quark() const override;
bool is_u_quark() const override;
bool is_u_bar_quark() const override;
bool is_d_quark() const override;
bool is_d_bar_quark() const override;
bool is_gluon() const override;

// M6 Boolean functions (6 total)
bool is_b_jet() const override;
bool is_c_jet() const override;
bool is_s_jet() const override;
bool is_u_jet() const override;
bool is_d_jet() const override;
bool is_gluon_jet() const override;

- <u>https://code.ihep.ac.cn/zhangkl/PrimeTagSvc</u>
- Code reconstructed;
 - Ensure PFO sorted by Energy, length up to 50.
- Follow the tutorial JetDump
 - Mode: Zjet(M10) and Hjet(M11).

Default: M11/M10 model



In weaver environment we got results like this;

		СЕРС	Ref-	TDR				$ZH \rightarrow vvqq, \sqrt{s} = 240 \text{ GeV}$						
	b	0.811	0.132	0.019	0.016	0.002	0.001	0.001	0.002	0.002	0.001	0.013		
	Đ	0.124	0.819	0.017	0.018	0.001	0.002	0.002	0.001	0.001	0.002	0.014		
	с	0.009	0.012	0.798	0.042	0.019	0.027	0.027	0.006	0.007	0.017	0.035		
	ō	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		
Truth	ŝ	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		
	ū	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		
	ā	0.001	0.002	0.020	0.009	0.078	0.132	0.239	0.059	0.076	0.289	0.095		
		0.011	0.012	0.029	0.029	0.074	0.077	0.072	0.066	0.057	0.057	0.514		
		Ø	Ń	C	ć	Pre	edict	ed	ن	6	Έ			

	CEPC	Ret-		∠→qq, 91.2 Gev						
q	0.823	0.127	0.021	0.017	0.004	0.002	0.001	0.002	0.002	0.001
ġ	0.124	0.826	0.017	0.021	0.002	0.003	0.002	0.001	0.001	0.002
J	0.009	0.009	0.815	0.047	0.031	0.034	0.025	0.006	0.008	0.016
υ	0.009	0.009	0.045	0.816	0.035	0.031	0.006	0.025	0.016	0.007
s	0.001	0.001	0.022	0.023	0.601	0.107	0.028	0.093	0.081	0.043
s ILL	0.001	0.002	0.023	0.021	0.108	0.601	0.091	0.028	0.044	0.082
п	0.001	0.001	0.026	0.011	0.056	0.172	0.412	0.049	0.080	0.192
ū	0.002	0.001	0.011	0.028	0.170	0.055	0.047	0.417	0.188	0.081
q	0.002	0.001	0.014	0.025	0.155	0.103	0.078	0.213	0.336	0.073
ġ	0.001	0.002	0.024	0.014	0.105	0.156	0.208	0.080	0.071	0.339
	b	b	С	Ē	<i>s</i> Pred	ة icted	и	ū	d	đ

Current output in onnx, test in 10k.



- 1. b and bbar charge flip error;
- 2. incorrect response like s jet;

..... Current b eff ~83%.





Onnx consistence



_						
-	Final	Laye	er Bias	Values	('mod.fc.0.bias') -	
•						
	b	:	-0.3516	605		
	bbar	:	-0.3768	952		
	С	:	0.0803	361		
	cbar	:	-0.0410	993		
	S	:	0.0307	752		
	sbar	:	0.1226	531		
	u	:	0.0608	356		
	ubar	:	-0.0100	973		
	d	:	0.0978	329		
	dbar	:	0.0654	122		

• By checking the bias value and

input/output structure, we can

confirm that the pt in pytorch and

onnx in C++ is exactly same.

Possible reason for errors

For low eff:

ill-behavior jets.

With small energy/PFOs and far away from quark direction. (DR>1.2)

This jet has only 3 tracks in final. Model didn't see weird jets before.

PrimeTagSvc		;	DEBUG F1	avor	Tagging	Scores							
b)		0.001500										•
b	_bar		0.025265										
c	;		0.000655										
c	_bar		0.003506										
s	;		0.387091										
s	_bar		0.321368										
ι	I		0.009419										
ι	ı_bar		0.021320										
c	i		0.030352										
c	l_bar		0.199525										
Jet	DumpAlg		DEBUG Jet 1,	E: 2	8.82, PF	Os: 15,	M11_type:	3, M6_type:	3, is_b:	No , is_c: No	o , is_light:	Yes	

For model energy transition: Hjet and Zjet model application difference;

- For charge flip and wrong response like s jets:
- Same Onnx model -> Inputs must be different.
- Float/Double, decimal changes between weaver/C++...
- Plan to retrain one new model to test.



To do



- Shap
- Variable Validation
- Delphes Validation
- Event multihead

Current Input variables





- Impact parameter d0 and log(d0):
 - To test.

Argument on feature engineering d0 and log(d0): CEP



- The goal is to make the learning task easier for the optimizer;
- By pre-computing a non-linear basis, the model now learns an easy linear combination instead of a hard logarithmic function.
- Possibly: linear d0 information lost in tanh() normalization (tanh(9)=1 in float), while log(d0) complete it;
- Standard Practice in CS:
 - ResNet, core idea: F(x)-x magic;
 - Positional Encodings: The "sin(pos)" trick give the information model already knows; ٠
- Would try a new training to test its impact.

D0 distribution



Both TDR/CDR full simulation has similar D0/Z0 performance. And showing patterns in both D0 and log(D0) distribution.



Log10(d0) pattern



In D0 and Z0 plot, 3 pattern can be seen:

IP(Primary Vertex), while track in jet E~1GeV, the d0/z0 precision ~5um. (-2.5~3um)

Secondary/Thirdary Vertex(From b decay. Length~100um.)

and Long-Live Decay Vertex(From Kshort, Lambda....., length~cm.)



Fast/Full discrepancy



TDR truthID, fast(left)/full(right) JOI. Fast overall worse, esp. for b eff.

Hqq_ParT_TruthID													
b -	0.832	0.113	0.019	0.016	0.002	0.001	0.001	0.002	0.001	0.001	0.012		
bbar -	0.113	0.831	0.016	0.018	0.001	0.002	0.002	0.001	0.001	0.002	0.012		
c -	0.009	0.009	0.822	0.034	0.024	0.024	0.023	0.005	0.006	0.014	0.030		
cbar -	0.009	0.009	0.035	0.820	0.024	0.024	0.005	0.023	0.016	0.006	0.030		
s -	0.002	0.001	0.015	0.017	0.584	0.081	0.023	0.085	0.081	0.042	0.069		
- sbar -	0.001	0.001	0.015	0.016	0.079	0.586	0.087	0.023	0.046	0.076	0.070		
u -	0.001	0.002	0.018	0.006	0.029	0.114	0.440	0.044	0.079	0.183	0.085		
ubar -	0.002	0.001	0.007	0.018	0.115	0.031	0.048	0.422	0.202	0.071	0.084		
d -	0.001	0.001	0.009	0.016	0.096	0.074	0.072	0.212	0.362	0.071	0.086		
dbar -	0.001	0.001	0.016	0.009	0.074	0.099	0.233	0.067	0.079	0.334	0.087		
g -	0.012	0.012	0.028	0.027	0.075	0.075	0.076	0.071	0.067	0.060	0.498		
	لم الم الم الم الم الم الم الم الم الم ا												



PN/ParT difference



By PN to ParT we have 2% upgrade in Fast, 5% in Full.

						Pr	redicte	ed				
		b	$\frac{1}{b}$	ċ	$\frac{1}{c}$	s	5	ù	ū	d	d	Ġ
	G -	0.019	0.018	0.029	0.029	0.072	0.072	0.063	0.066	0.056	0.058	0.518
	d -	0.004	0.005	0.019	0.014	0.080	0.086	0.202	0.078	0.076	0.313	0.121
	d -	0.005	0.005	0.015	0.019	0.083	0.081	0.081	0.206	0.304	0.081	0.121
	u -	0.004	0.005	0.015	0.020	0.108	0.040	0.058	0.378	0.171	0.083	0.119
	u -	0.004	0.005	0.019	0.014	0.038	0.110	0.375	0.056	0.080	0.180	0.119
True	<u>s</u> -	0.004	0.005	0.020	0.021	0.099	0.558	0.072	0.024	0.041	0.054	0.102
	s -	0.004	0.005	0.020	0.020	0.555	0.103	0.024	0.073	0.053	0.043	0.099
	. -	0.014	0.014	0.057	0.751	0.028	0.032	0.009	0.021	0.014	0.008	0.052
	с-	0.014	0.014	0.748	0.059	0.032	0.029	0.021	0.009	0.007	0.014	0.052
	b -	0.151	0.739	0.025	0.033	0.004	0.006	0.003	0.003	0.002	0.003	0.031
	b -	0.743	0.147	0.032	0.025	0.005	0.005	0.003	0.004	0.002	0.002	0.031

						Pr	edicte	ed				
		b	$\frac{1}{b}$	ċ	$\frac{1}{c}$	s	5	ù	$\frac{1}{u}$	d	d	Ġ
	G -	0.018	0.017	0.029	0.028	0.072	0.068	0.065	0.067	0.060	0.059	0.517
	<u>d</u> -	0.004	0.005	0.020	0.014	0.077	0.076	0.204	0.086	0.076	0.324	0.115
	d -	0.005	0.004	0.016	0.018	0.079	0.073	0.087	0.199	0.330	0.074	0.114
	u -	0.005	0.004	0.015	0.019	0.102	0.035	0.064	0.384	0.181	0.078	0.113
	u -	0.004	0.005	0.020	0.014	0.037	0.099	0.389	0.062	0.082	0.175	0.113
Irue	<u>s</u> -	0.004	0.005	0.020	0.021	0.100	0.556	0.071	0.027	0.046	0.057	0.093
	s -	0.005	0.004	0.023	0.019	0.566	0.093	0.027	0.074	0.056	0.044	0.090
	c -	0.014	0.015	0.056	0.757	0.026	0.029	0.009	0.021	0.015	0.008	0.049
	с-	0.015	0.013	0.760	0.055	0.030	0.026	0.021	0.009	0.008	0.015	0.049
	b -	0.145	0.753	0.021	0.029	0.004	0.005	0.004	0.004	0.002	0.003	0.030
	b -	0.762	0.136	0.029	0.021	0.005	0.004	0.003	0.004	0.003	0.002	0.030

Fast simulation:



Current fast simulation with overall worse performance including BMR, diphoton resolution and so on. Under check.

