CMS Draft Analysis Note

The content of this note is intended for CMS internal use and distribution only

2017/07/21
Head Id: 417226
Archive Id: -1:417226M
Archive Date: 2017/07/21
Archive Tag: trunk

Deep learning for jet reconstruction

The CMS Collaboration
CERN

Abstract

Deep learning lead to several breakthroughs outside the field of high energy physics, yet in jet reconstruction at CMS it was not used so far. This report shows results of applying deep learning strategies to jet reconstruction at the stage of tagging and calibration. Jets with cone radius 0.4 and fat jets with radius 0.8 where investigated. We introduce deep neural network structures that were not yet proposed in this context and show that in all cases we studies significant gain in performance can be achieved by this approach with respect to the established CMS methods. We systematically also review other recently proposed network structures in context of sub-structure tagging without flavours. The proposed strategy is a multi-classification and for jets with radius of 0.4 and 0,8, as well as also a transverse momentum estimation for slim jets. The classes include heavy objects like, top, H, Z, W for fat jets and heavy flavour, light quark and gluon tagging for the slim jets.

1 Introduction

The reconstruction of jets is a central element in high energy physics collider experiments. Recently several studies using simplified simulation made first studies on using deep neural networks (DNNs) to identify (tag) the particle that caused a jet. Some used the analogy of the calorimeter cells to pixels in photographs to apply convolutional or dense networks that are often used for photo labeling [1–4]. The results were mixed, ranging from some improvement to no improvements with respect to established methods. Also recurrent neural networks were proposed [5, 6]. CMS and ATLAS released public documents [7, 8] on applying DNNs in context of flavour tagging and in CMS the default flavour tagger is derived from a DNN, that, for the first time, showed the gain in performance in real data for a real detector.

In this note we present results of using new DNN structures in the context of jet tagging and regression for jets with radii of 0.4 and 0.8, which are the default jets in CMS. In section 2, we discuss the samples used for to train the different tagger, the input variables (we from now on use the machine learning term: features) used for the tagging (classification), the generator level truth of the different particles ID (in the following we call these labels), e.g. B-hadron, and finally the pre-processing applied to the raw features. Section 3 describes the DNN architectures chosen for AK4 and AK8. Finally, in section 4, we show the results compared to the standard tools in CMS.

2 The setup for DNN training

2.1 Training samples for slim jets

For the training of AK4 jets we use the QCD and $t\bar{t}$ samples listed in table 1. The generator used for $t\bar{t}$ is POWHEGv1.0 [9–13] generators were used. Showering and hadronization is done by the PYTHIA 8.2 package [14] and the detector simulation by the GEANT4 [15] package. QCD is done with PYTHIA only. All samples are with simulated using phase1 detector design. After a pres-election, which reduces the gluon jets we altogether have about 80M jets for training, testing and validation.

The samples used for the training of the AK8 jets are listed in table 2. We are currently using 2016 samples which have much larger statistics than the available PhaseI 2017 MC samples (training a DNN using samples with limited number of simulated events can impact the performance).

Table 1: Simulated phase1 samples for the training for slim jets.

Sample	Full name
QCD	/QCD_Pt_XtoY_TuneCUETP8M1_13TeV_pythia8/PhaseIFall16MiniAOD-PhaseIFall16PUFlat20to50_PhaseIFall16_81X_upgrade2017_realistic_v26-v1/MINIAODSIM
$t\bar{t}$	/TT_TuneCUETP8M2T4_13TeV-powheg-pythia8/PhaseIFall16MiniAOD-PhaseIFall16PUFlat20to50_PhaseIFall16_81X_upgrade2017_realistic_v26-v1/MINIAODSIM

Table 2: Simulated MC samples used for the training of the AK8 jets. The samples are from the Summer16 campaign.

Sample	Full name
t	/TT_Mtt-*_TuneCUETP8M2T4_13TeV-powheg-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToTTJet_M-*_TuneCUETP8M1_13TeV-amcatnlo-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/RSGluonToTT_M-*_TuneCUETP8M1_13TeV-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
W	/BulkGravToWWToWlepWhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/BulkGravToWW_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/RadionToWWToWlepWhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/RadionToWW_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/RadionToWW_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZToWhadZinv_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZToWhadZlep_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZToWhadZlep_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToWWToWlepWhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToWWToWlepWhad_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToWW_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToWW_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
Z	/BulkGravToZZToZhadZhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/BulkGravToZZToZhadZinv_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/BulkGravToZZToZlepZhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/RadionToZZToZlepZhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZToWlepZhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZToWlepZhad_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
H	/BulkGravTohhTohVVhbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/BulkGravTohhTohbbhbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWhToWlephbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToZhToZinvhbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/ZprimeToZhToZlephbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
W/Z	/WprimeToWZToWhadZhad_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
	/WprimeToWZ_width0p*_M-*_TuneCUETP8M1_13TeV-madgraph-pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
W/H	/WprimeToWhToWhadhbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
Z/H	/ZprimeToZhToZhadhbb_narrow_M-*_13TeV-madgraph/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-v1/MINIAODSIM
QCD	/QCD_Pt_*_TuneCUETP8M1_13TeV_pythia8/RunIISummer16MiniAODv2-PUMoriond17_80X_mcRun2_asymptotic_2016_TrancheIV_v6-*/MINIAODSIM

2.2 Labeling

We define the labeling in major and minor labels. The major labels define a broader category of jets, e.g. jets with at least one B hadron. The minor labels are a further sub-division of the major labels, e.g. jets with two or more B hadrons. All major labels with respect to all other major labels and all minor labels with respect to all other minor labels are orthogonal. The major label are the sum of their sub-labels.

2.2.1 Slim jet labels

Heavy flavor hadrons, scaled to negligible transverse momentum in order not to impact the final properties of the jet, are added to the list of stable particles to be clustered by the AK4 jet algorithm. Jets containing one or more heavy flavor objects in its constituents are assigned one of the major heavy flavor labels, which than are further sub-dived to minor labels to separate different decays or the number of heavy flavour hadrons in a jet. Jets not containing any clustered heavy-flavor hadron are labeled according to the flavor of the hardest (maximum transverse momentum) parton with PYTHIA 8 `status = 23`, assigning either the `light quark` or the `gluon` labels. This labeling of `light quarks` or `gluons` is following the "physics definition" as defined in [16]. Jets with no heavy-flavor hadron clustered among the constituents, but with a heavy-flavor quark as hardest parton, are considered undefined and excluded from the training and evaluation procedure. This is a limitation of the miniAOD data format used to extract the dataset. A summary of all major and minor label is shown in table 3

Table 3: The three major flavour label and the sub division of each of these flavour in even more detailed label. The sum of all minor (sub) labels is equivalent to the major label.

Major label	Minor (sub-)label
`B`, $\geq$ 1 B hadron	`bb`, two ore more B hadrons $\mathtt{b}_{lep}$, exactly one B hadron with leptonic decay `b`, exactly one B hadronic hadronic decay
`C`, $\geq$ 1 C hadron and no B hadron	`cc`, two or more C hadrons `c`, exactly one C hadron
`L`, None of the above and parton matched	`uds`, physics definition [16] `g`, physics definition [16]

2.2.2 AK8 labels

The multi-classification (top, H, Z and W tagging) approach followed for the AK8 jets requires mutually exclusive labelling. Priority is given to the hadronically decaying heavy objects (i.e. top, H, Z, W). The generated heavy object, X, and its decay products, X_{decay} are matched to the AK8 jet following the conditions: $\Delta R(X, \mathrm{AK8}) < 0.6$ and $\Delta R(X_{\mathrm{decay}}, \mathrm{AK8}) < 0.6$.Then AK8 jets with heavy flavor content are identified following the BTV-style. The remaining jets are identified as light quarks/gluons. The proposed labelling is summarized in table 4 and is designed to have high granularity. The various labels can be easier combined. We therefore two differnt kind of labels, "major" and "minor", to target the different analysis needs. We consider AK8 jets with transverse momentum, p_{T}, greater than 300 GeV.

The proposal described above is currently pending approval from the relevant CMS sub-groups. For the results shown in this version of the note we follow a simplified approach presented in 5.

Table 4: Summary of the various labels proposed for the classification of the AK8 jets.

Major label	minor label
t	bcq
	bqq
	bc
	bq
W	cq
	qq
Z	bb
	cc
	qq
H	bb
	cc
	qq
	qqqq
QCD	bb
	b
	q
	g

Table 5: Summary of the labelling used for the results presented in the current version of the note for the classification of the AK8 jets. q_X refers to the quarks from the hadronic decay of the boson X.

Major label	Requirement
t	$\Delta R(\mathrm{b}, \mathrm{AK8}) < 0.8$ and $\Delta R(q_\mathrm{W}, \mathrm{AK8}) < 0.8$
W	if W is from t decay: $\Delta R(\mathrm{b}, \mathrm{AK8}) > 0.8$ and $\Delta R(q_\mathrm{W}, \mathrm{AK8}) < 0.8$ else: $\Delta R(q_\mathrm{W}, \mathrm{AK8}) < 0.8$
Z	$\Delta R(q_\mathrm{Z}, \mathrm{AK8}) < 0.8$
H	$\Delta R(q_\mathrm{H}, \mathrm{AK8}) < 0.8$
QCD	anything else

2.3 Preprocessing

Before the input variables are feed into the neural networks for the training or the evaluation of a trained model, they are subject to a preprocessing. Goal of the preprocessing is to avoid inputs to the neural network with significantly different scales or unintended biases as this leads to easier convergence of the minimization.

2.3.1 Slim jet preprocessing

The scales are unified using the mean $< x >$ and the standard deviation σ_x of each feature x which is rescaled to

$$x^I = \frac{x - < x >}{\sigma_x}. \tag{1}$$

The p_T and η of each jet are direct input to the neural network, such that the evaluation of the input features can be adjusted according to the jet kinematics. However, the kinematics of jets originating from a different parton flavours show partially significant deviations from each other. In consequence the neural network could learn to assign a jet flavour purely based on

the p_T and η of the jet. To avoid such biases, jets are removed from the training sample, such that their p_T and η distributions agree for all jet flavours. As reference p_T and η shape we use the shape of the b-jets, i.e. finally the other labels will have all the b-jet shape. In addition we remove 50% of the gluons in order to avoid a gluon dominated training dataset. The probabilities to remove a jet are calculated based on the entire training sample to minimise the impact of statistical fluctuations.
In case a feature is missing we put in default values, that are not far from the normalized scale and are not overlapping with the core distribution.

2.3.2 AK8 preprocessing

The input variables for AK8 tagging are preprocessed in a similar way as in AK4 tagging. Each input variable x is transformed according to Eq. (2.3.2),

$$x^I = \frac{x - p_{50\%}}{p_{84\%} - p_{50\%}}, \quad (2)$$

where $p_{50\%}$ and $p_{84\%}$ are the 50th and 84th percentiles of the variable x. In the case when $p_{84\%} = p_{50\%}$ (which happens for some discrete variables), the denominator is taken to be 1. The use of percentiles instead of the mean and the standard deviation tends to be less sensitive to outliers and distributions with long tails, leading to more unified scales for different variables. The transformed values are further clipped to be in the range of $[-5, 5]$ before feeding into the neural networks, which are found to help improve the stability of neural network training.

To avoid biases from the difference in the jet p_T spectrum, jets in the training sample are reweighted to have a flat distribution in p_T, and the contribution of each source (top, W, Z, Higgs, and QCD) is equalized. However, from our studies we found that applying weights to the neural network training often causes degradation of performance or even failure in convergence. As a result, the reweighting is done "on-the-fly" by randomly sampling the training dataset according to the "flattening" weights, thus effectively achieves the reweighting without losing statistics.

For evaluating the performance, jets in the testing sample are reweighted such that different signal processes (top, W, Z, Higgs) all have the same p_T spectrum as the background process (QCD).

2.4 Input features

The basis for the taggers are the Particle Flow [17] jet constituents (particle candidates), namely charged and neutral PF candidates as well as reconstructed secondary vertices within the jet.

2.4.1 slim jet input features

For the DeepFlavour tagger, several features of the jet constituents and of secondary vertices within a cone of $\Delta R = 0.4$ with respect to the jet axis are used. In some cases, their variation is restricted to a reasonable range to avoid large outliers e.g. due to mis-measurements having a strong effect on the training without providing any discrimination power. In addition, particular inputs are shifted by a constant offset, such that 0 corresponds to a value that is outside of the bulk of the distribution and provides no handle on the flavour separation. For the charged PF candidates, the majority of the input features are calculated following previous b-tagging algorithms [18]. These are in the following indicated as BTV features and their exact definition can be found in the reference [18]. The additional variables are described in the following.

- $p_{\mathrm{T}}(j)$: jet p_{T}
- $\eta(j)$: jet η
- N_{cPF}: number of charged PF candidates within the jet
- N_{nPF}: number of neutral PF candidates within the jet
- N_{SV}: number of secondary vertices within the jet
- N_{PV}: number of primary vertices in the event
- $p_{\mathrm{T}}(cPF)/p_{\mathrm{T}}(j)$: relative p_{T} of a charged jet constituent with respect to the jet p_{T}
- $p_{\mathrm{T}}(nPF)/p_{\mathrm{T}}(j)$: relative p_{T} of a neutral jet constituent with respect to the jet p_{T}
- $\Delta R_m(cPF, SV)$: ΔR of charged candidate and closest secondary vertex within the jet
- $\Delta R_m(nPF, SV)$: ΔR of neutral candidate and closest secondary vertex within the jet
- VTXass: flags indicating whether the charged particle track is used in the primary vertex fit, includes steps from low purity to high purity requirements.
- fromPV: similar to VTXass, but partially including information about the primary vertex fit quality. Can indirectly include lepton information
- $w_p(cPF)$: weight assigned to the charged particle by the PUPPI [19] algorithm
- $w_p(nPF)$: weight assigned to the neutral particle by the PUPPI algorithm
- χ^2: charged PF candidate track χ^2
- quality: flag that indicates the charged particle track reconstruction quality, from passing low purity to high purity requirements
- $\Delta R(cPF)$: ΔR to jet axis of a charged candidate
- $\Delta R(nPF)$: ΔR to jet axis of a neutral candidate
- isGamma: flag whether a neutral candidate passes loose photon identification requirements
- hadFrac: fraction of energy deposits in the hadronic calorimeter, only for neutral candidates
- $p_{\mathrm{T}}(SV)$: secondary vertex p_{T}
- $\Delta R(SV)$: ΔR between jet axis and secondary vertex flight direction
- m_{SV}: invariant mass of reconstructed secondary vertex
- $N_{\mathrm{tracks}}(SV)$: number of tracks associated to the secondary vertex
- $\chi^2(SV)$: secondary vertex χ^2
- $\chi^2_n(SV)$: secondary vertex χ^2 normalised to degrees of freedom
- $d_{xy}(SV)$: transverse impact parameter of secondary vertex
- $S_{xy}(SV)$: transverse impact parameter significance of secondary vertex
- $d_{3D}(SV)$: 3D impact parameter of secondary vertex
- $S_{3D}(SV)$: 3D impact parameter significance of secondary vertex
- $\cos\theta(SV)$: $\cos\theta$ of secondary vertex with respect to primary vertex
- $E_{rel}(SV)$: ratio of secondary vertex energy with respect to the jet

All global features with per-jet values that are considered are summarised in Table 6. No offsets, upper or lower bounds are applied. These are applied to particular properties or charged and neutral PF candidates, and secondary vertices as listed in Tables 7, 8 and 9.

Table 6: List of global input features for the AK4 DeepFlavour tagger

feature	comment
$p_{\mathrm{T}}(j)$	
$\eta(j)$	
N_{cPF}	
N_{nPF}	
N_{SV}	
N_{PV}	
trackSumJetEtRatio	BTV
trackSumJetDeltaR	BTV
vertexCategory	BTV
trackSip2dValAboveCharm	BTV
trackSip2dSigAboveCharm	BTV
trackSip3dValAboveCharm	BTV
trackSip3dSigAboveCharm	BTV
jetNSelectedTracks	BTV
jetNTracksEtaRel	BTV

Table 7: Full list of charged PF candidate features used as input to the DeepFlavour network for AK4 jets

feature	offset	lower bound	upper bound	comment
trackEtaRel	-	-5	15	BTV
trackPtRel	-	-	4	BTV
trackPPar	-	-10^5	10^5	BTV
trackDeltaR	-	-5	5	BTV
trackPParRatio	-10	100	-	BTV
trackSip2dVal	-	-	70	BTV
trackSip2dSig	-	-	$4 \cdot 10^4$	BTV
trackSip3dVal	-	-	10^5	BTV
trackSip3dSig	-	-	$4 \cdot 10^4$	BTV
trackJetDistVal	-	-20	1	BTV
trackJetDistSig	-	-1	10^5	BTV
$p_{\mathrm{T}}(cPF)/p_{\mathrm{T}}(j)$	-1	-1	0	
$\Delta R_m(cPF, SV)$	-5	-5	0	
fromPV	-	-	-	
VTXass	-	-	-	
$w_p(cPF)$	-	-	-	
χ^2	-	-	-	
quality	-	-	-	

Table 8: Full list of neutral PF candidate features used as input to the DeepFlavour network for AK4 jets

feature	offset	lower bound	upper bound
$p_{\mathrm{T}}(nPF)/p_{\mathrm{T}}(j)$	-1	-1	0
$\Delta R_m(nPF, SV)$	-5	-5	0
isGamma	-	-	-
hadFrac	-	-	-
$\Delta R(nPF)$	-0.6	-0.6	0
$w_p(cPF)$	-	-	-

Table 9: Full list of secondary vertex features used as input to the DeepFlavour network for AK4 jets

feature	offset	lower bound	upper bound
$p_{\mathrm{T}}(SV)$			
$\Delta R(SV)$	-0.5	-2	0
$m_{SV})$	-	-	-
$N_{\text{tracks}}(SV)$	-	-	-
$\chi^2(SV)$			
$\chi^2_n(SV)$	0	-1000	1000
$d_{xy}(SV)$	-	-	-
$S_{xy}(SV)$	-	-	800
$d_{3D}(SV)$	-	-	-
$S_{3D}(SV)$	-2	-2	0
$\cos\theta(SV)$	-	-	-
$E_{rel}(SV)$	-	-	-

The particles and vertices are ordered using a hierarchical sorting algorithm. Charged candidates and secondary vertices are sorted by impact parameter significance. If the charged candidate was used in the primary vertex fit, they are appended starting from the lowest $\Delta R_m(cPF, SV)$ value. If no secondary vertex is present within the jet, the particle p_{T} is used instead. The latter two sorting requirements are also applied to neutral PF candidates.

2.4.2 AK8 input features

The input features used by AK8 tagging are similar to those used in the AK4 DeepFlavour tagger. They are organized into three groups: inclusive (charged and neutral) PF candidates, charged PF candidates, and secondary vertices. We take up to 100 inclusive PF candidates, sorted in descending p_{T} order, and up to 60 charged PF candidates and up to 5 secondary vertices, ordered by impact parameter significance. The full lists of variables used in each group are summarized in Table 10 to 12.

Table 10: Full list of charged PF candidate features used as input to the DeepAK8 network

feature	comment
trackEtaRel	BTV
trackPtRatio	BTV
trackPParRatio	BTV
trackSip2dVal	BTV
trackSip2dSig	BTV
trackSip3dVal	BTV
trackSip3dSig	BTV
trackJetDistVal	BTV
$p_{\mathrm{T}}(cPF)/p_{\mathrm{T}}(j)$	
$E_{rel}(cPF)$	
$\Delta\phi(cPF, j)$	
$\Delta\eta(cPF, j)$	
$\Delta R(cPF, j)$	
$\Delta R_m(cPF, SV)$	
$\Delta R(cPF, \text{subjet } 1)$	
$\Delta R(cPF, \text{subjet } 2)$	
χ^2_n	
quality	
d_z	
S_z	
d_{xy}	
S_{xy}	
track_dptdpt	track covariance
track_detadeta	track covariance
track_dphidphi	track covariance
track_dxydxy	track covariance
track_dzdz	track covariance
track_dxydz	track covariance
track_dphidxy	track covariance
track_dlambdadz	track covariance

3 Deep neural network architectures

The neural network structure was designed to be able to make good use of the large input we give to the neural network. In contrast to previous proposals we use more information per particles candidate or vertex. This lead to the special challenge to digest the huge amount of input features. In order to not expose the later layer to such a huge amount of features we build a reduced set features per particle (or per few particles) candidate or vertex by so called

Table 11: Full list of inclusive PF candidate features used as input to the DeepAK8 network

feature
$p_{\mathrm{T}}(PF)/p_{\mathrm{T}}(j)$
$E_{rel}(PF)$
$\Delta\phi(PF, j)$
$\Delta\eta(PF, j)$
$\Delta R(PF, j)$
$\Delta R_m(PF, SV)$
$\Delta R(PF, \text{subjet 1})$
$\Delta R(PF, \text{subjet 2})$
$w_p(PF)$
f_{HCAL}

Table 12: Full list of secondary vertex features used as input to the DeepAK8 network

feature
$p_{\mathrm{T}}(SV)/p_{\mathrm{T}}(j)$
$E_{rel}(SV)$
$\Delta\phi(SV, j)$
$\Delta\eta(SV, j)$
$\Delta R(SV, j)$
$p_{\mathrm{T}}(SV)$
m_{SV}
$N_{\text{tracks}}(SV)$
$\chi^2_n(SV)$
$d_{xy}(SV)$
$S_{xy}(SV)$
$d_{3D}(SV)$
$S_{3D}(SV)$
$\cos\theta(SV)$

convolutional layers. Convolutional layers learn a transformation from a typically higher dimensional representation to a lower representation of features, which in our physics jargon would be similar to building a few variables from a larger input. This is done simultaneously with the overall optimization, i.e. the transformation is trained to be ideal for the classification. Convolution networks are very spread in image recognition, where they effectively summarize small region of the image and build more useful features than the raw pixels, like edges or alike, which than are feed to the following layers. In our case a particle candidate or vertex takes the role of such a small region of an image.
While slim and fat jets share this basic structure in the beginning, we currently use slightly different networks structures in the later layers.

3.1 Slim jet DNN architecture

The first layers are convolutional layers as explained in the previous paragraph. Figure 1 indicates the number of layer and nodes for these convolutional layers. To allow non-linearities we use up to four convolutional layers. The convolution are done 1x1, i.e. they are applied only to individual particle candidates and they only reduce the dimension of the feature per candidate or vertex, but are not a summary of several candidates. We use the rectified linear unit (ReLu) activation function.
From convolutional layer we get sequences of features of particle candidates. The sequence order is still defined from the input particle candidate (or vertex) sorting. They are sorted by displacement significance. The most displaced are the last in the list. In case the particles are not displaced and no secondary is in the jet, they are sorted with increasing p_{T}. Exact sorting details are in 2.4. These sequences are than feed into recurrent neural networks (LSTM) and by that compressed to a single vector per sequence, i.e. charged and neutral candidates and vertices. When using recurrent networks the ordering is important, thus our underlying assumption is that the most displaced (in case of displacement) or the highest p_{T} candidates matter the most.
The output of the recurrent layers is than combined with the global variables, like p_{T} and η. This is put into a fully connected neural network with 8 layers. The first layer has 200 nodes and the latter 100.Again we use ReLu activation.
In between the layer we use a dropout of 0.1 and do batch normalization apart for the input (layet 0) and output (last layer before loss). For the final layer we use the softmax function as activation and cross entropy as the loss to minimize. For the minimization we use the adam [20] optimizer and train for 50 epochs. The workflow was implemented using [21] that relies on [22, 23] for the neural network implementation. To check for over-training we use separate sample that is not used for training and no over-training was found. The final ROCs curves in the results section 4 using another third set of independent samples.

3.2 DNN AK8

The task of tagging heavy objects (top, W, Z, Higgs) with AK8 jets is more challenging than b-tagging in some aspects. With a larger jet radius, a typical AK8 jet has many more constituent particles than AK4 jets. And the interrelationship between theses particles, like the spatial pattern and the energy correlation, is more crucial for heavy objects than for b-tagging. Thus, a more complex DNN architecture is adopted for AK8 tagging.

Similar to the DNN model for AK4 tagging, the DNN model for AK8 tagging, as illustrated in Fig. 2, first processes inclusive PF candidates, charge PF candidates and SVs separately with convolutional neural networks, and then combines outputs from these three networks in a fully-connected layer before yielding the final prediction. The network is trained as a whole to

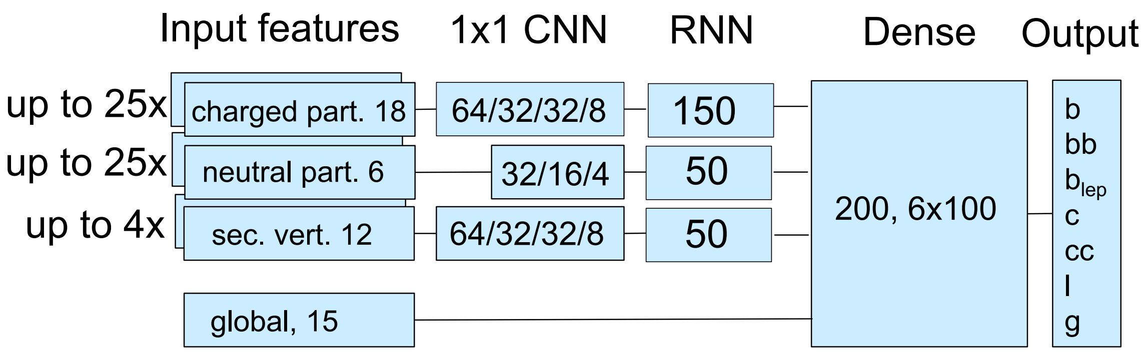

Figure 1: DNN architecture illustration. Dropout and batch normalizations are not indicated. The number in the boxes indicate the number of nodes per layer.

optimize all the components simultaneously.

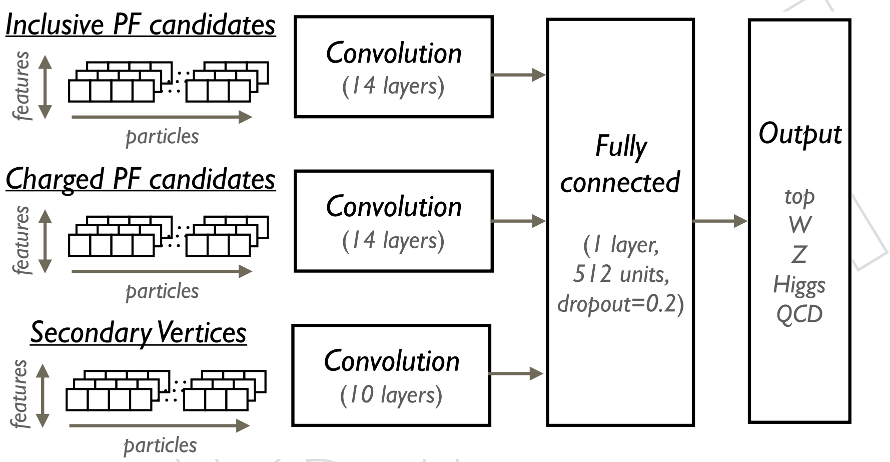

Figure 2: Illustration of the overall DNN architecture used by AK8 tagging.

Different from the convolutional neural networks in AK4 tagging, where the convolution is performed for each individual particle (i.e., "1x1"), the convolution here is performed for each adjacent particle triplet (i.e., "3x1") with overlaps (in CNN jargon, we use a kernel size of 3 and a stride of 1). Such "3x1" convolutions are stacked on top of each other, thus allow the DNN to see the correlation between nearby particles at earlier stages, and to have a more global view of the particle correlations at later stages. The design of the convolutional neural networks model is largely based on the ResNet model [24], which is one of the state-of-the-art model for image recognition. We adapt it to work with one-dimensional particle list instead of two-dimensional pictures, but adopt the main structure and all important ingredients such as residual connection [25], batch normalization [26], and ReLU [27] activation function. The depth of the convolutional network is 14 for inclusive PF candidates and charge PF candidates, and 10 for SVs. The filter sizes (i.e., the number of output features) for each convolutional layer ranges between 32 to 128.

The outputs from the three separate convolutional neural networks are combined in a fully-connected layer with 512 units, followed a ReLU activation and a DropOut layer with a rate of 0.2. We use the softmax function in the final layer to yield the final prediction, and cross entropy as the loss function to minimize. The neural network is implemented with the MXNet package [28] and trained with the Adam [20] optimizer with a learning rate of 0.001.

4 Results in simulation

We compare the results of the classification for AK4 and AK8 jets to references that are used in public analysis in CMS. We use physics sample with label composition and p_T and η shapes as they come naturally from the samples process in question. We reject jets with undefined labels.

4.1 AK4 jet results

For AK4 we use the CVSv2 and DeepCSV b-tagger as reference [18, 29] and for the quark-gluon discrimination we compare to [30] and alternative deep neural network structures. Figure 3 compares by showing the ROC curves the DeepFlavour tagger results to the former default CMS tagger CSVv2 and DeepCSV for different processes and p_T. For both physics processes, $t\bar{t}$ and QCD we see significant gain in all region of the ROC curves. For very high b-jet p_T the b-jet efficiency is increased by 50% with respect to the DeepCSV for a light fake rate of 1%. At higher p_T of jets gluon splitting leads to an increased amount of jets in QCD with two b-hadrons inside the jet. In a high p_T region we thus show in Figure 4 the efficiency for jets with single b-hadron using only the single b and leptonic b labels as discriminator. Identifying single bs is slightly more difficult than double bs and the performance is slightly less good. We also show the tagging performance for the bb label, using also the bb discriminator and it can be seen that separating bb from light jets is easier, as the performance is improved with respect to the single b case. The second curve in the double b ROC is the separation of b and bb, using the probabilities as binary classifier (binary means here that two estimated propabilities (double b and single b) used are renormalized to add up to one, before they are used as discriminator). It is interesting that about 1/3 of the jets can be identified as double b with only a fake-rate of 1% for single bs. Figure 4 also shows the efficiency of leptonic decays of single bs vs the mistag rate for light jets and we see a decent separation, just in between the bb and single b case. The separation of hadronic and leptonic bs, even using the separation as binary classifier, does not lead to good leptonic decay separation. It should be noted that we did not explicitly add lepton information to the tagger.
The c-tagging of DeepJet is compared to DeepCSV in Figure 5. We do see a gain with respect to DeepCSV also for c-tagging.
We defined three working points, which lead to a light jet efficiecy of 10%. 1%, and 0.1% repectively for jets of the QCD sample in a range from 80 to 120 GeV. Using these working points, called loose, medium and tight, we illustrate the dependence of the tagger performance as a function of p_T. η, and number of primary vertices in Figure 6. We see the expected degradation of performance with higher p_T and at low , for large η and large number of primary vertices. Note that the QCD sample has a relatively flat p_T distribution, thus the p_T integrated illustrations are dominated by high p_T.
An overview of the discriminator shapes us given in Figure 7. The discrimunators shown in Figure 7 are according to labels that were present in CMS before and methods to estimate the data simulation agreement re present. The minor labels of the b-hadron jet major label are shown in the appendix Figure A. Especially the double b vs single b separation seems promising and it might motivate a dedicated effort to develop methods to also establish these discriminators in data in the future.

Figure 8 and 9 show the comparison of DeepJet quark gluon separation to the default likelihood method for different p_T in the central and forward region of the detector, respectively. The output of DeepJet was made to a binary classifier to compare to the quark-gluon likelihood

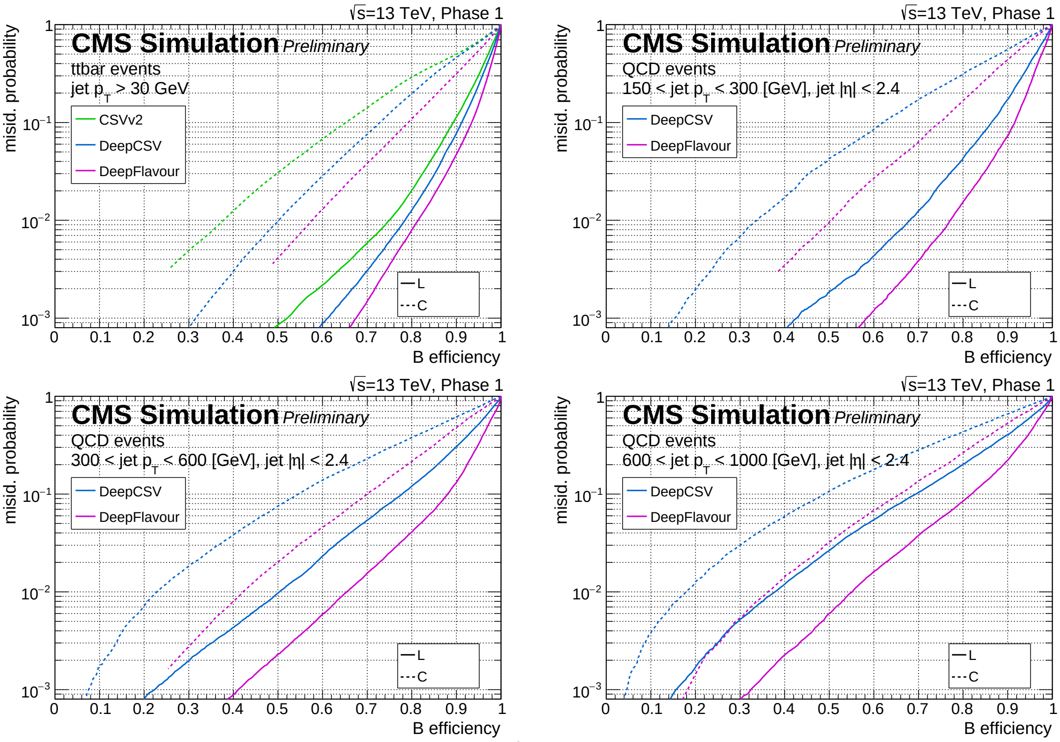

Figure 3: ROC curves for different processes and p_T ranges for b-tagging. Here all three categories, b, b_{lep}, and bb are considered as b-jet. Light includes the gluon and uds-quark categories. The top left plots show the ROC evaluated with $t\bar{t}$ events. The latter show the ROCs for QCD samples with increasing p_T.

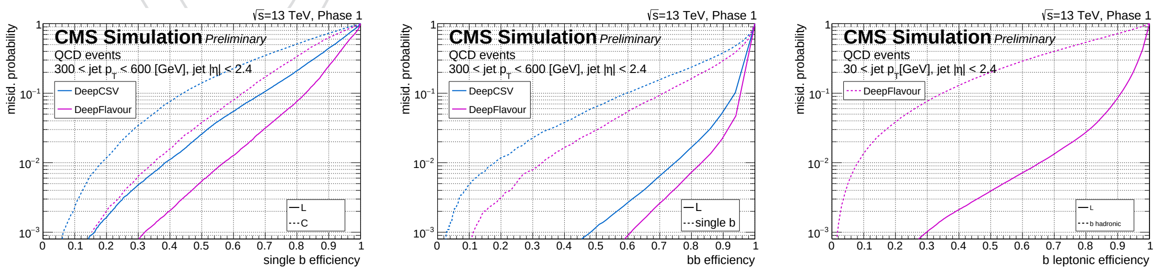

Figure 4: ROC curves for different b cathegories. For the efficiency vs light and c-jet always the unmodified multi-label propability of the b cathegorie is used. For the double b vs b and leptonic b decay vs. hadronic the binary classifyers between these cathegories are used.

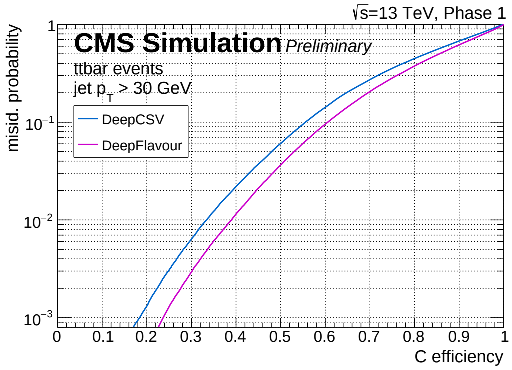

Figure 5: ROC curves for C-tagging in $t\bar{t}$ events using the estimated propabilities as binary classifier for light jet and c-jet separation.

method, which technically was done by scaling the estimated probabilities for light quarks and gluons such that they add up to one. We see constantly a significant improvement of about 5-10% absolute better efficiency for light quarks compared to likelihood method. Note that for this comparison all jets of the QCD sample are taken into account that pass the kinematic criteria and where we did find that the label was well defined, i.e. no balance selection of jets was applied and instead it was only checked that a parton was found. We also conformed that a balance selection as done by other quark gluon taggers did not effect the conclusions, but moved all ROCS by a tiny amount towards better tagging.
We also added as comparison other DNN architectures that should also be able to find jet structures, but would be blind to heavy flavour. For these DNNs we use only p_T of the candidates relative to the jet, relative ϕ, relative η, if the particles are charged or not, and the puppi weight. We try two DNN structures, one according to [3], i.e. an image with 22 bins and a convolutional DNN. We use relative eta from each particle candidate to define the bin which the particle belongs to. For each bin we store the p_T sum of the particles with puppi weight (we tried both, with and without puppi weights) and the multiplicities of charged and neutral particle candidates. Also the exact details of the layer structure are used as in the reference with the only difference that we could remove the regularization layers, as we use larger samples. Alternatively we also took the list of particles candidates (charged and neutral), with the same above mentioned information sorted in descending p_T. As for DeepFlavour we use a recurrent (LSTM) network followed by several dense layer. We compared these different flavour blind neural network structures to DeepFlavour. All three structure give similar results as seen in 8 and 9.
Another study we did is the impact of reduced input, i.e. using a few human made variables. We gave as input the five variables currently used by the BDT quark gluon effort and added p_T, η and rho. This made altogether 8 input variables. Again we used a DNN very similar to the one of the recommended CMS flavour tagger DeepCSV. Only 15 nodes per layer are used, which as for DeepCSV is between 1-2 times the inputs. We used 7 hidden layers. The comparison can be found in 10 and 11. We see a gain by using a full DNN with larger input than only the human made variables. The effect varies depending on the p_T and η and can be sizable at background rejection of around 10%.
To have an illustrative example on how the multi-classification simplifies real-life tasks, we show how one can select light quarks in $t\bar{t}$ events. traditionally this was quite difficult, as we had a tagger to separate gluons and quarks, one to separate b-jets from "light" jets and yet another to separate c-jets from "light" jets. Finally on would apply cuts on all three taggers or even build a meta-tagger based on other taggers output. For DeepJet it is sufficient to just ask

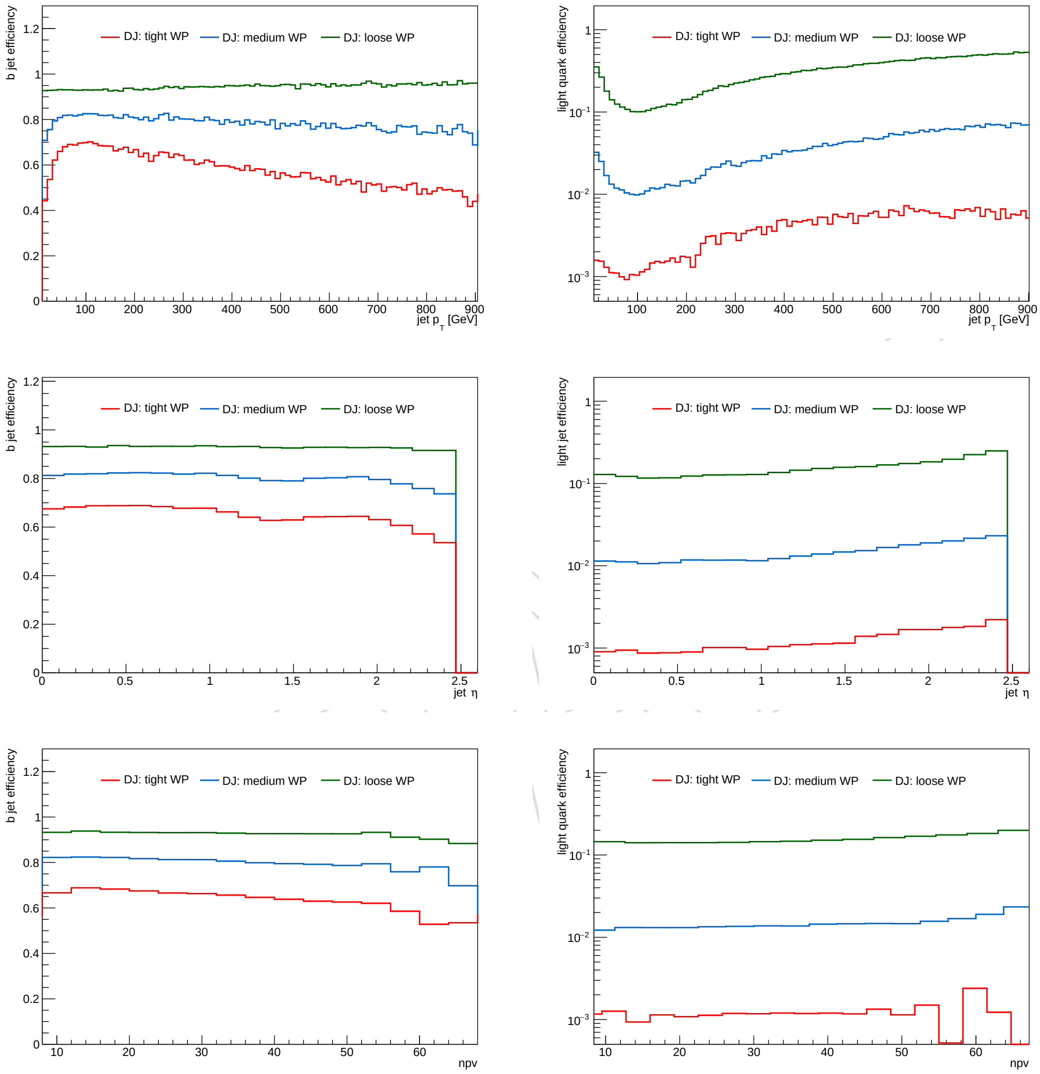

Figure 6: Efficiency for B and L in the QCD sample as functions of p_T, η and number of primary verices (npv). To not be dominated by high p_T jet in QCD, for the η and npv dependencies only jets with $p_T > 30$ and < 150 GeV were used. Efficiencies for three working points: loose, medium and tight are shown.

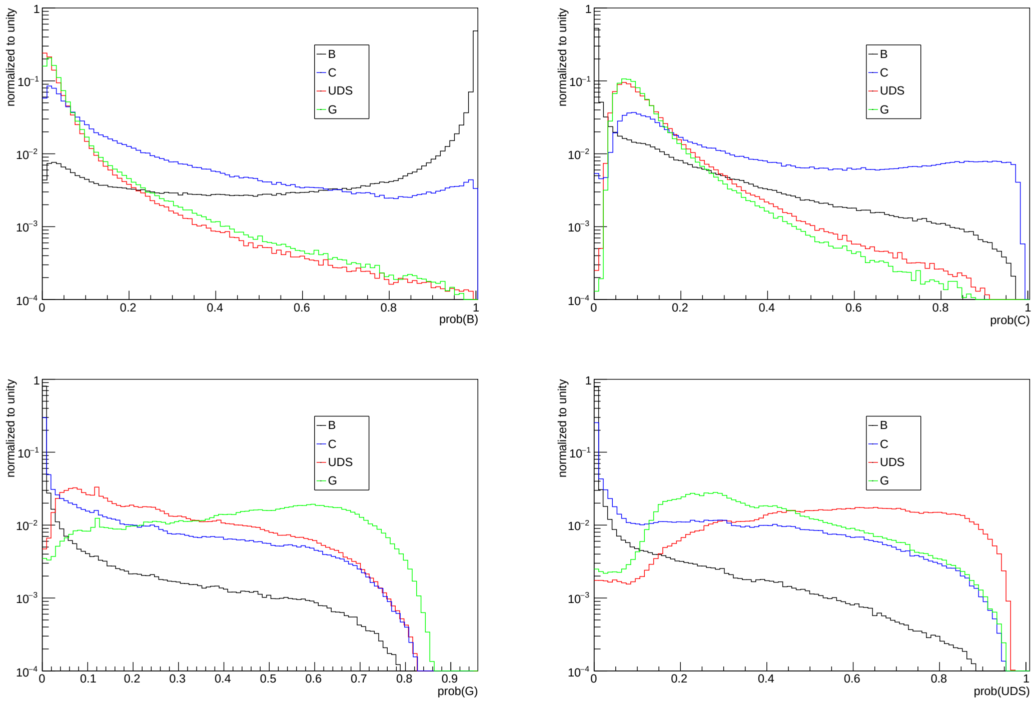

Figure 7: The different estimated probabilities for different jet labels as observed in the $t\bar{t}$ sample.

for the uds probability. A comparison is shown in Figure 13, where we show how the quark-gluon likelihood compares to DeepJet in answering the question if a jet is a uds-jet. While it is difficult and often sub-optimal to try to extract a jet label from different standalone taggers, it is straight forward to reduce a multiclass tagger to a tagger with only two label by rescaling the tagger information as e.g. done in the previous paragraph for the quark-gluon separation.

For the results shown in Figures 8 and 9, the area under the curve is listed in Table 13. In addition, the Table shows the efficiency ϵ to select a light quark for two working points, defined by a misidentification probability of 0.2 (loose), 0.1 (medium), or 0.01 (tight). The reference for these working points is the QCD sample with $\hat{p}_T = 80 - 120\,\text{GeV}$ for central η. The numbers are extracted from the same sample and for the sample with .$\hat{p}_T = 300 - 470\,\text{GeV}$ The conclusion is similar for all $\hat{p}_T$ ranges.

4.2 AK8 jet results

This section discusses the performance of the classification of the AK8 jets as originating from t, H, Z, W, or QCD jet using the DNN structure detailed in Section 3.2. The distributions of the individual probabilities as obtained from the DeepAK8 multi-tagger for different cases of truth-matched jets are displayed in Fig 15.

The performance of the DeepAK8 multi-tagger is compared with the performance a boosted decision tree (BDT) classifier heavily based on the t and W BDT developed in the all-hadronic search for direct stop production [31]. Details about the selection of variables, the training, as well as on the performance in MC and data can be found in Section 3.3 and Appendix B in [32]. Very briefly the input variables exploit jet kinematics, Nsubjettiness ratios, soft-drop

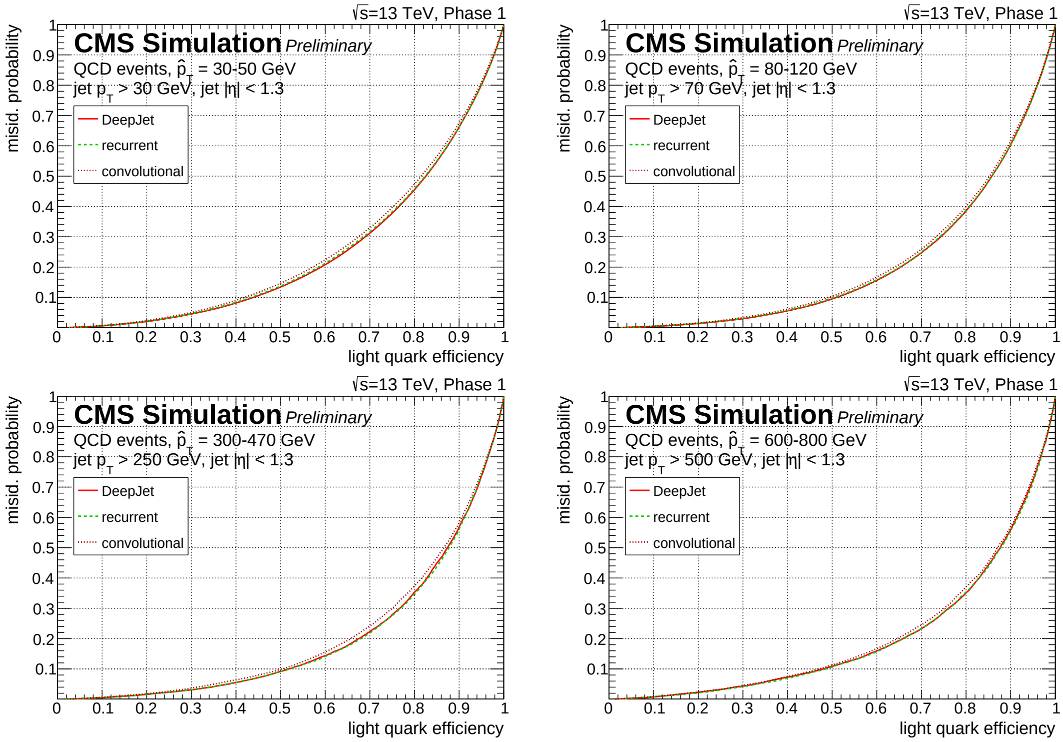

Figure 8: ROC curves in QCD simulation for different p_T ranges in the central region ($|\eta| < 1.3$). Compared are the default DeepJet and the recurrent and convolutional approaches.

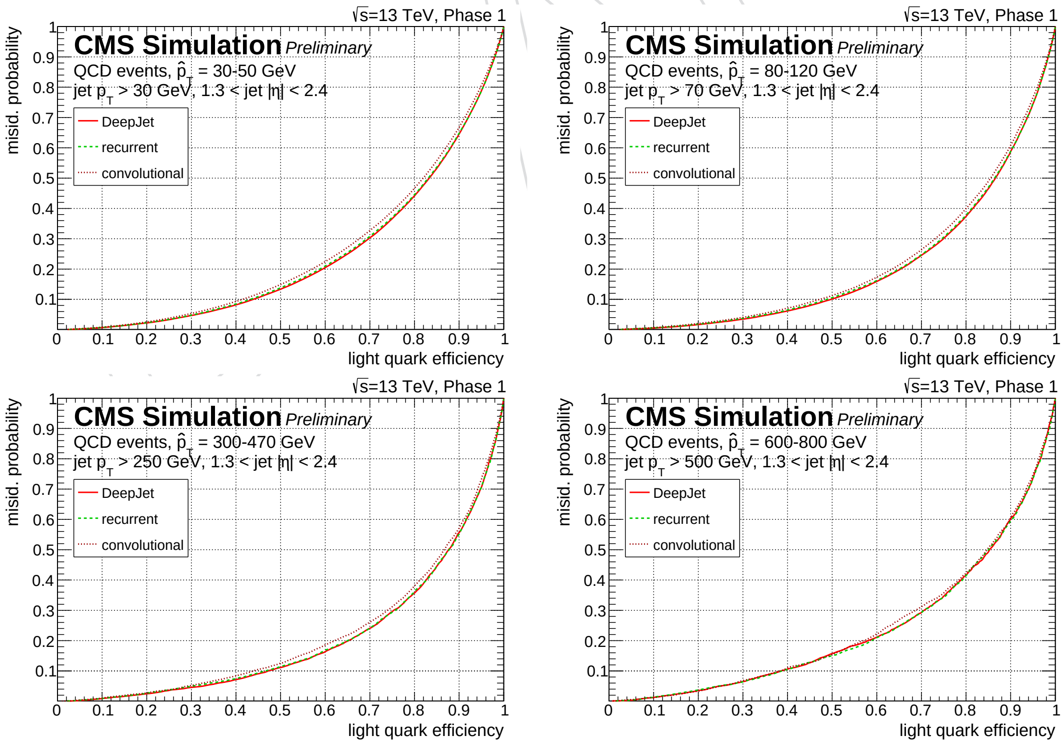

Figure 9: ROC curves in QCD simulation for different p_T ranges in the forward region ($1.3 < |\eta| < 2.4$). Compared are the default DeepJet and the recurrent and convolutional approaches.

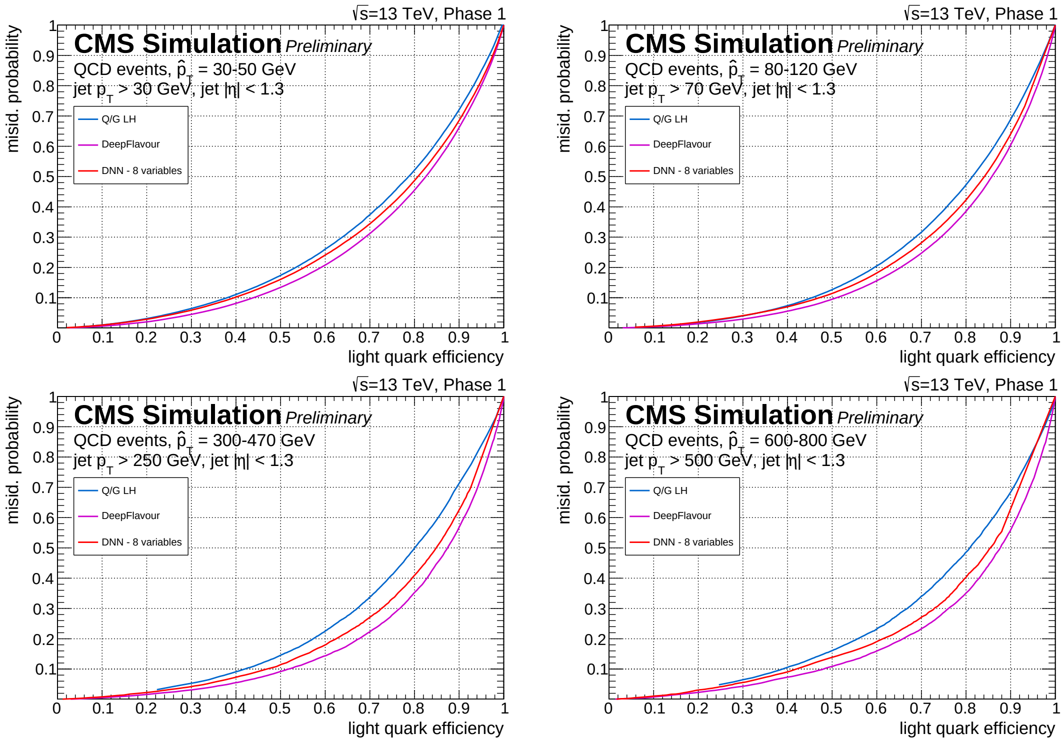

Figure 10: ROC curves in QCD simulation for different p_{T} ranges in the central region ($|\eta| < 1.3$). Compared are the likelihood method, the 8 parameter DNN and the default DeepFlavour.

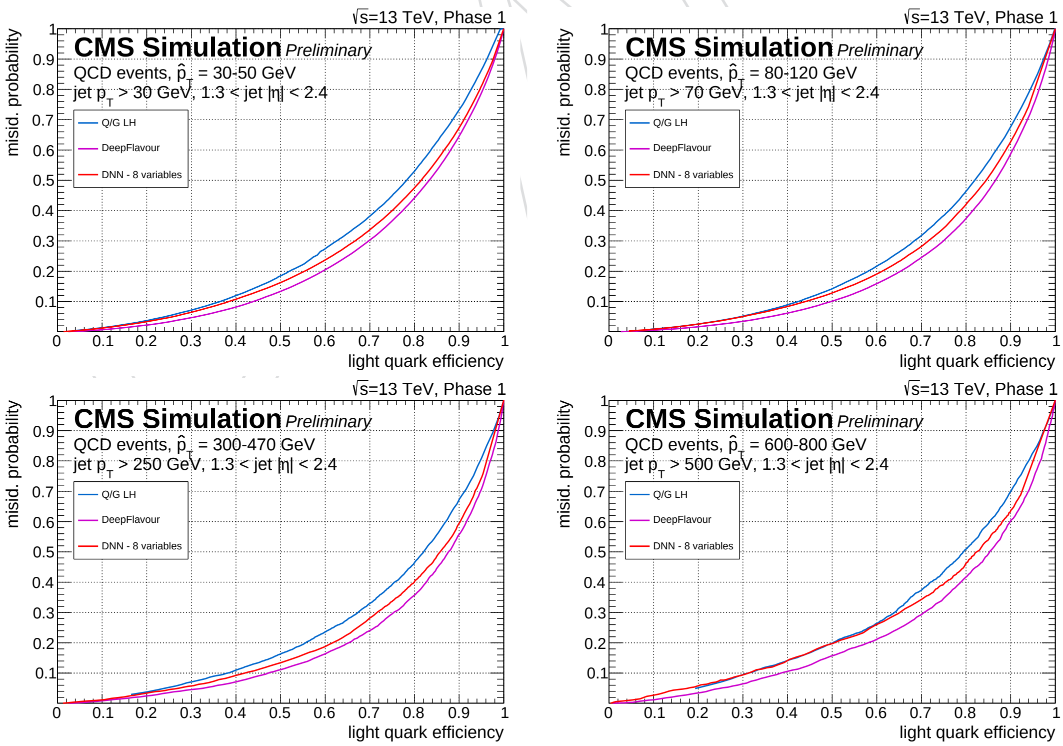

Figure 11: ROC curves in QCD simulation for different p_{T} ranges in the forward region ($1.3 < |\eta| < 2.4$). Compared are the likelihood method, the 8 parameter DNN and the default DeepFlavour.

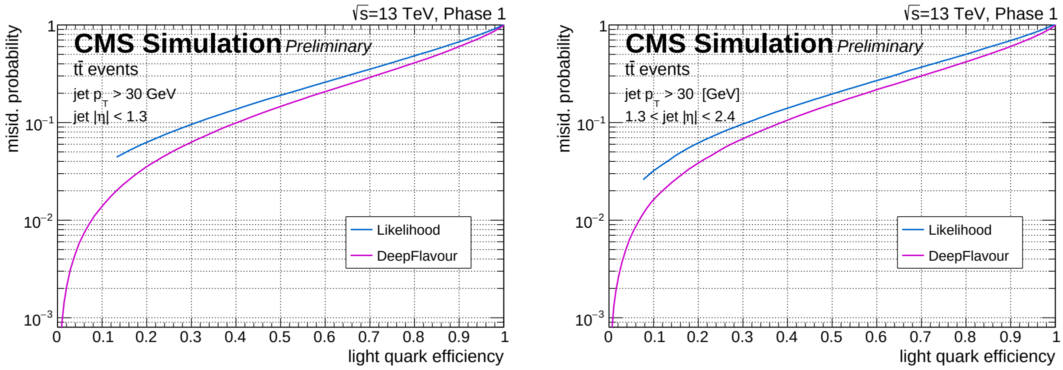

Figure 12: ROC curves in $t\bar{t}$ simulation for $p_T > 30$ GeV in the central ($|\eta| < 1.3$) and forward region ($1.3 < |\eta| < 2.4$) for light-quark and gluon separation.

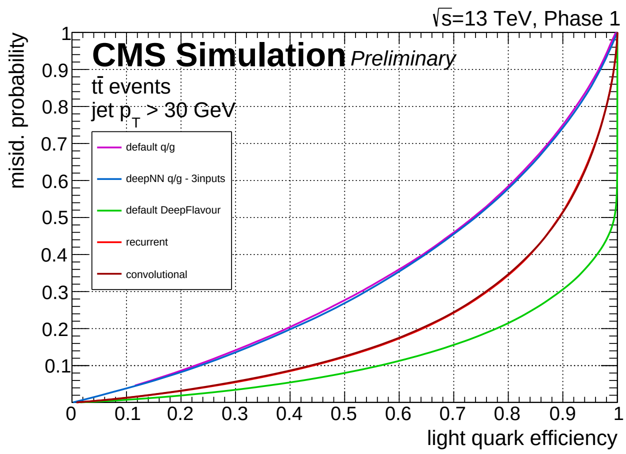

Figure 13: ROC curves in $t\bar{t}$ simulation for $p_T > 30$ GeV for light-quark-jets as signal and any other labeled jet as background.

Table 13: Area under the ROC curve and efficiencies for two selected working points for the different DNN-based approaches for quark-gluon tagging, evaluated for QCD samples with different $\hat{p}_T$ and jet p_T thresholds.

	Area under ROC	ϵ(tight)	ϵ(medium)	ϵ(loose)
QCD $\hat{p}_T = 80 - 120$ GeV, jet $p_T > 70$ GeV				
DeepJet central	0.204	0.17	0.51	0.65
DeepJet forward	0.203	0.15	0.50	0.65
Convolution central	0.211	0.15	0.49	0.64
Convolution forward	0.215	0.13	0.47	0.63
Recurrent central	0.205	0.16	0.51	0.65
Recurrent forward	0.205	0.14	0.49	0.65
QCD $\hat{p}_T = 300 - 470$ GeV, jet $p_T > 250$ GeV				
DeepJet central	0.193	0.15	0.52	0.68
DeepJet forward	0.201	0.11	0.47	0.65
Convolution central	0.203	0.13	0.50	0.66
Convolution forward	0.214	0.10	0.44	0.62
Recurrent central	0.191	0.15	0.52	0.68
Recurrent forward	0.203	0.10	0.47	0.65

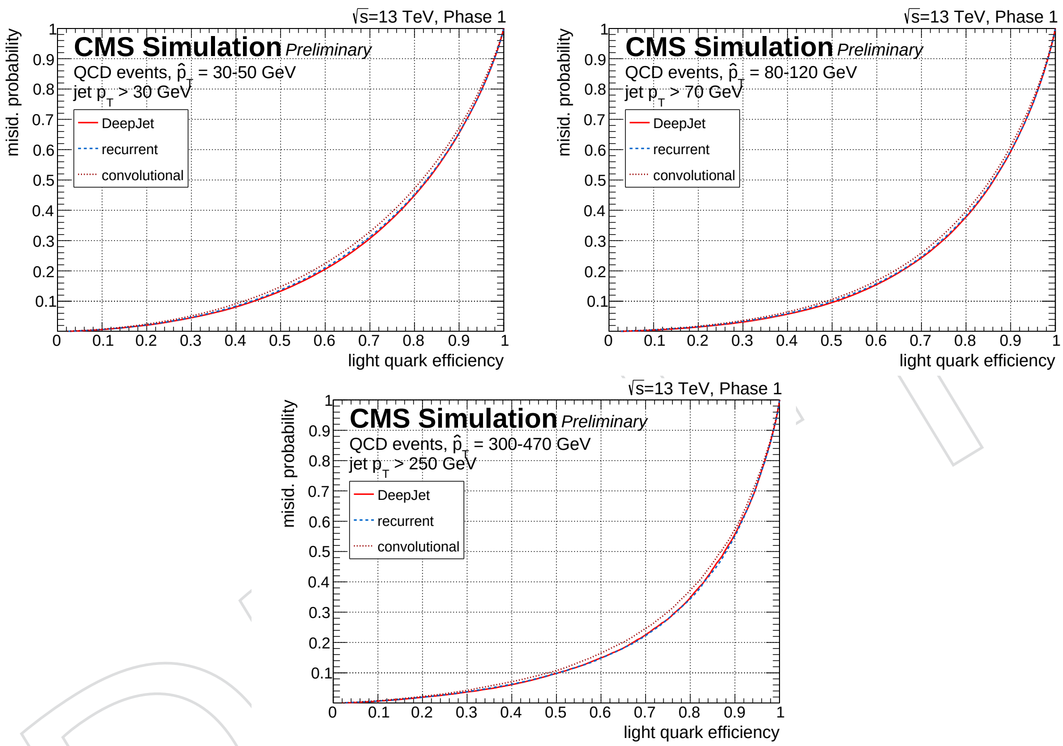

Figure 14: ROC curves in Pythia 8 QCD simulation for different p_{T} ranges in the full $|\eta|$ region. Compared are the default DeepJet, the recurrent and convolutional approaches. Jets matched to uds quarks are considered light quark jets. Gluon jets are defined by a matching to gluons. Jets with heavy-flavour hadrons are excluded from the jets considered. This also applies to gluon splitting to BB and CC.

(SD) mass, Q/G variables and CSV discriminants of the SD subjets. Given that the DeepAK8 multi-tagger targets also hadronic decays of the H and Z bosons, to allow for a fair comparison, the input variables used by the boosted double-b tagger [33] are also included in the t/W BDT. This results to a total of 45 input variables to the BDT. The BDT is retrained using the same samples as the DeepAK8 multi-tagger.

The performance of the DeepAK8 and BDT-based multi-taggers is evaluated in different regions of the p_{T} of the AK8 jet in terms of receiver operating characteristic (ROC) curves. An independent sample (i.e. not the sample used for train or validation) is used to produce the ROC curves. The results are displayed in Fig. 16-19. The efficiency of correctly identifying a t, H, Z, or W (signal efficiency, x-axis) is always compared against the QCD efficiency (background efficiency, y-axis). The DeepAK8 multi-tagger outperforms the BDT multi-tagger in all cases. For example in a working point corresponding to a background efficiency of $\sim$1%, the DeepAK8 multi-tagger yields 10-25% larger signal efficiency in all classes.

One of the advantages of a multi-tagger is that allows separation between different objects. As an example, in Fig. 20 we compare the performance of the DeepAK8 and BDT multi-taggers to separate W and Z jets. This is a very challenging problem given the similar mass of the two bosons. The DeepAK8 multi-tagger shows significantly better performance over a wide range of p_{T}.

Appendix B includes comparison of the performance plots of the DeepAK8 and BDT multi-tagger using the samples and matching definitions described in [34]. In addition to the DeepAK8 and the BDT multi-tagger ROC curves, we include the performance of two "cut-based" working points (high and low purity) for each heavy object as described in [35] and [36].

5 Conclusion

For cases of our muti taggers we see significant gain with repect the the CMS reconstruction defaults taggers in the performance evaluated in simulation by ROC curves. This is true for slim jet taggin for heavy flaavours and quark gluon separation as well as for heavy resonances, H, top, W, and Z tagging for fat jet. The improvements range from a couple of % to even factors of 2 in efficiency gain at some mistag rate. The next step will be study of these gains in real data.

References

[1] J. Cogan, M. Kagan, E. Strauss, and A. Schwarztman, "Jet-Images: Computer Vision Inspired Techniques for Jet Tagging", *JHEP* **02** (2015) 118, `doi:10.1007/JHEP02(2015)118`, `arXiv:1407.5675`.

[2] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, "Deep-learning Top Taggers or The End of QCD?", *JHEP* **05** (2017) 006, `doi:10.1007/JHEP05(2017)006`, `arXiv:1701.08784`.

[3] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, "Deep learning in color: towards automated quark/gluon jet discrimination", *JHEP* **01** (2017) 110, `doi:10.1007/JHEP01(2017)110`, `arXiv:1612.01551`.

[4] P. Baldi et al., "Jet Substructure Classification in High-Energy Physics with Deep Neural Networks", *Phys. Rev.* **D93** (2016), no. 9, 094034, `doi:10.1103/PhysRevD.93.094034`, `arXiv:1603.09349`.

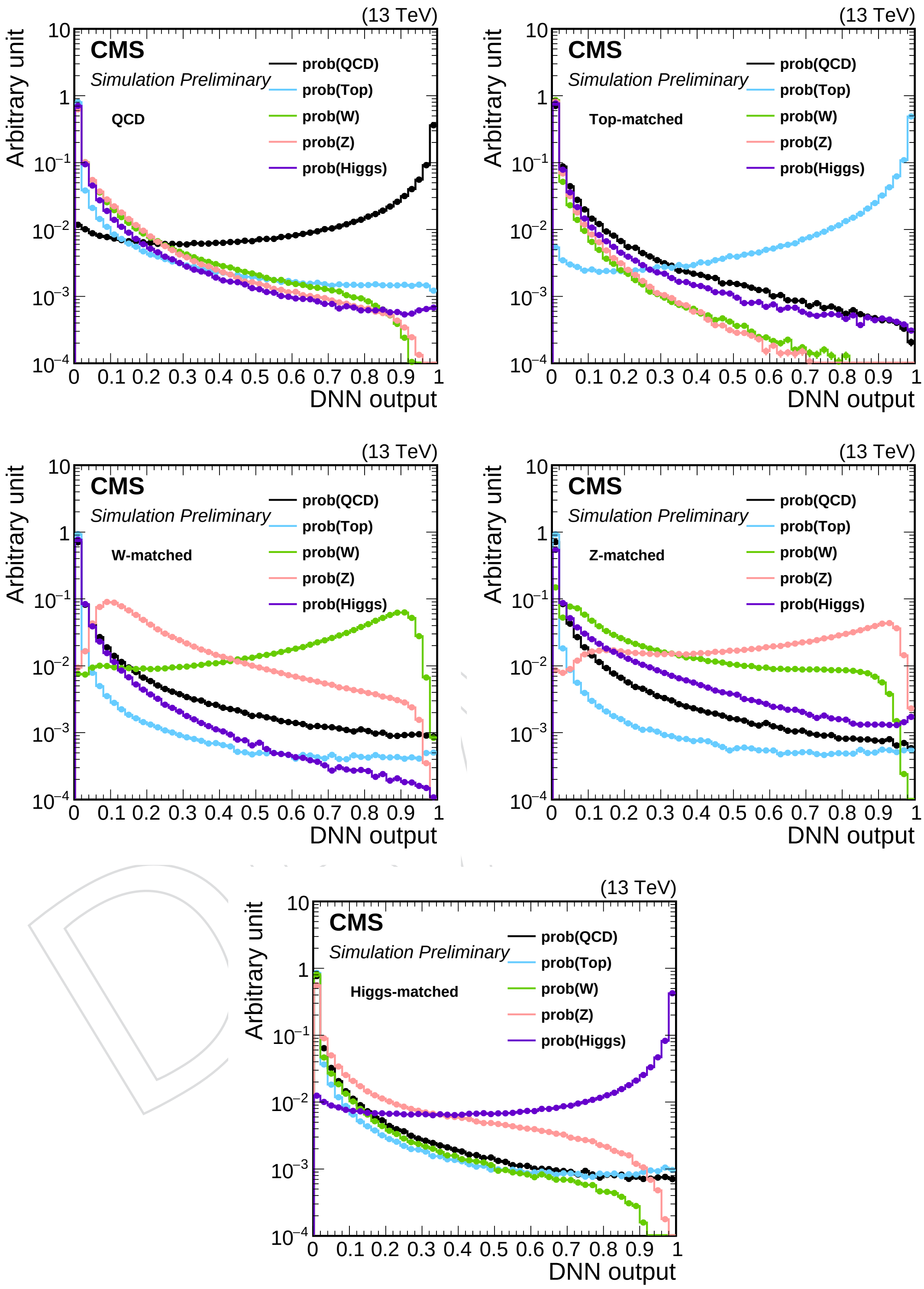

Figure 15: Distribution of the individual probabilities (DNN output) as obtained from the DeepAK8 tagger for different cases of truth-matched jets. Each truth-matched case is indicated on the plot. In this example AK8 jets with $p_{\mathrm{T}} > 400$ GeV are considered.

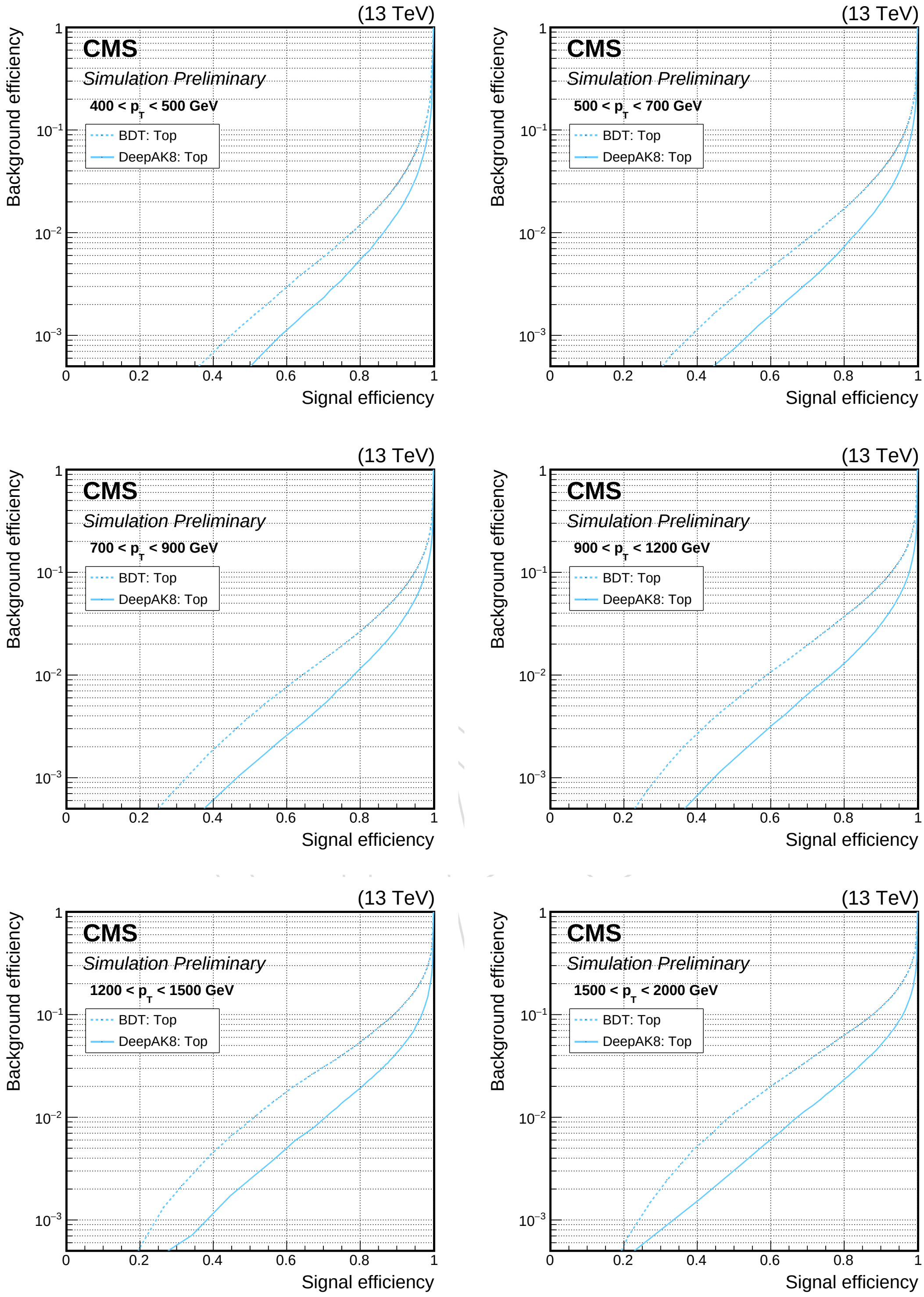

Figure 16: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for t jets as signal and QCD jets as background. The plots correspond to different p_{T} ranges of the AK8 jet. upper: $400 < p_{\mathrm{T}} < 500$ GeV (left) and $500 < p_{\mathrm{T}} < 700$ GeV (right), middle: $700 < p_{\mathrm{T}} < 900$ GeV (left) and $900 < p_{\mathrm{T}} < 1200$ GeV (right), lower: $1200 < p_{\mathrm{T}} < 1500$ GeV (left) and $p_{\mathrm{T}} > 1500$ GeV (right).

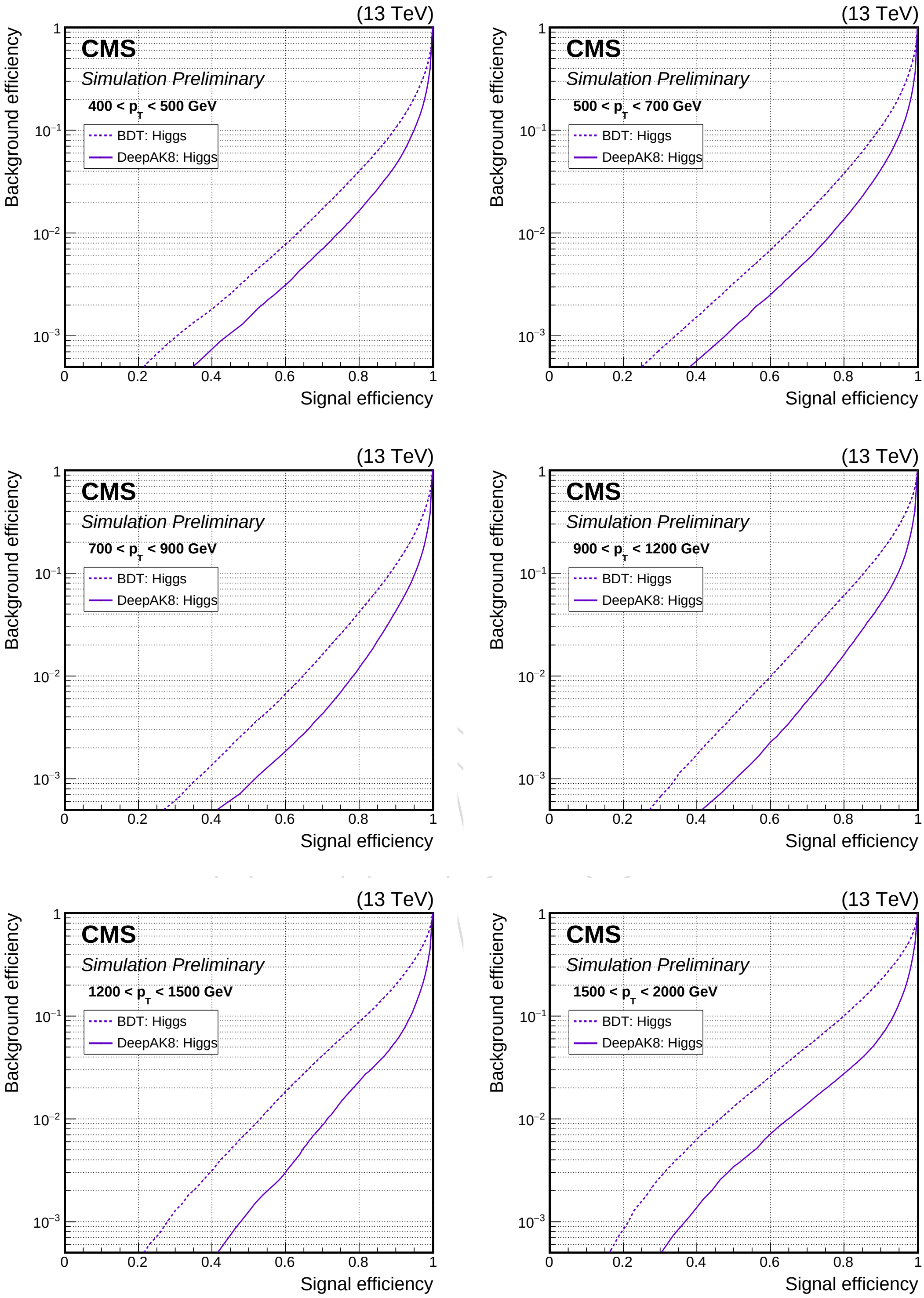

Figure 17: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for H jets as signal and QCD jets as background. The plots correspond to different p_T ranges of the AK8 jet. upper: $400 < p_T < 500$ GeV (left) and $500 < p_T < 700$ GeV (right), middle: $700 < p_T < 900$ GeV (left) and $900 < p_T < 1200$ GeV (right), lower: $1200 < p_T < 1500$ GeV (left) and $p_T > 1500$ GeV (right).

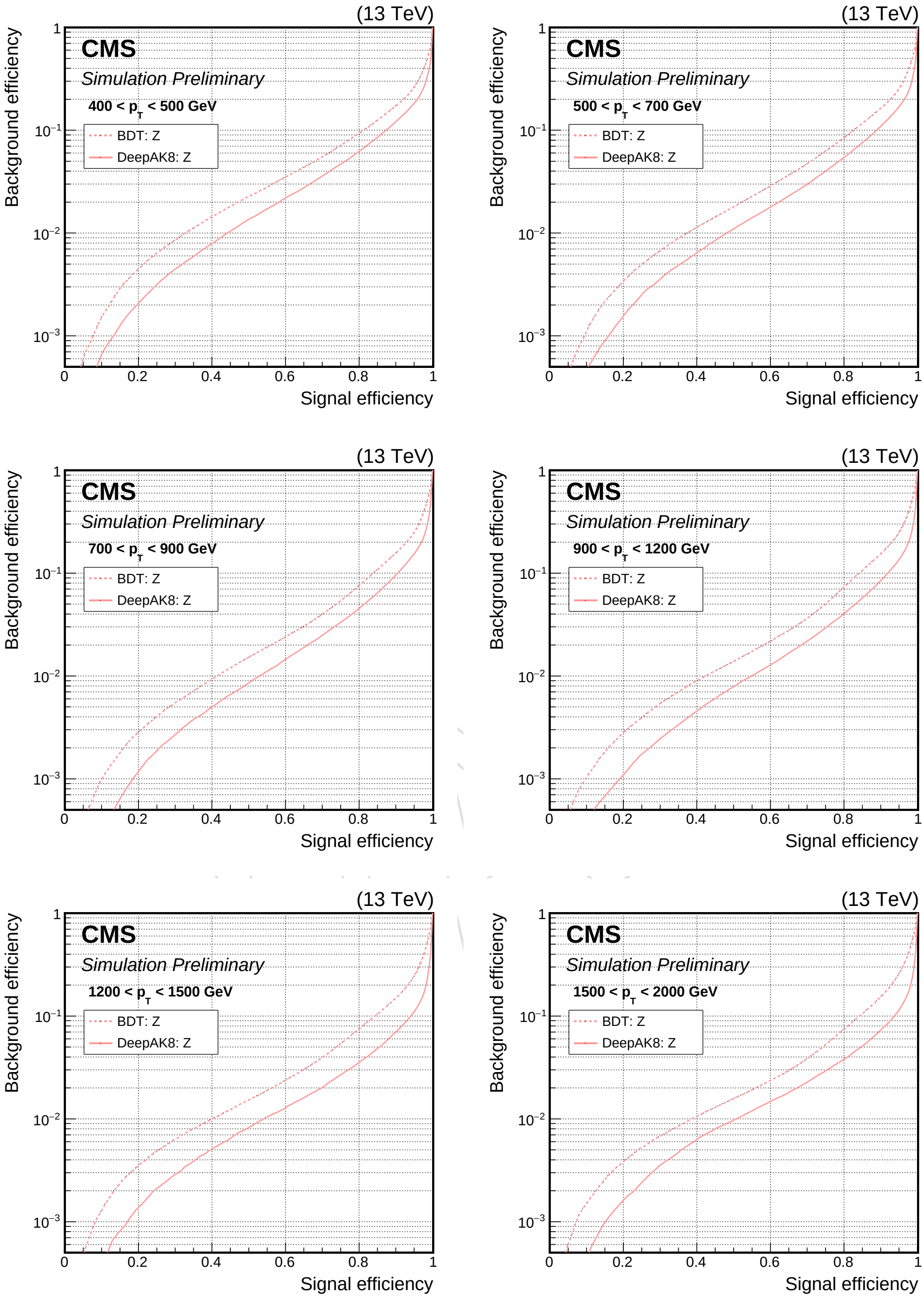

Figure 18: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for Z jets as signal and QCD jets as background. The plots correspond to different p_{T} ranges of the AK8 jet. upper: $400 < p_{\mathrm{T}} < 500$ GeV (left) and $500 < p_{\mathrm{T}} < 700$ GeV (right), middle: $700 < p_{\mathrm{T}} < 900$ GeV (left) and $900 < p_{\mathrm{T}} < 1200$ GeV (right), lower: $1200 < p_{\mathrm{T}} < 1500$ GeV (left) and $p_{\mathrm{T}} > 1500$ GeV (right).

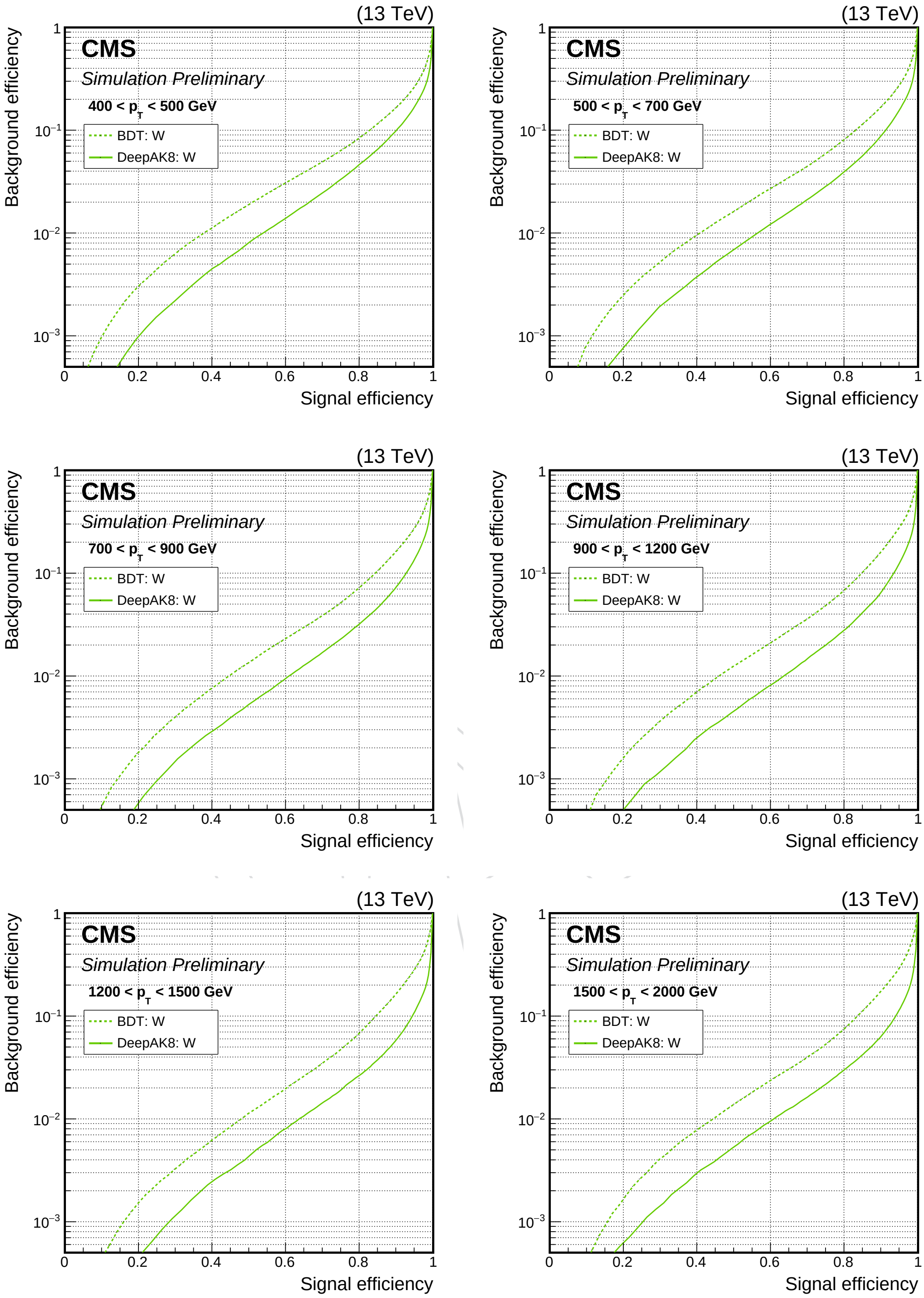

Figure 19: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for W jets as signal and QCD jets as background. The plots correspond to different p_{T} ranges of the AK8 jet. upper: $400 < p_{\mathrm{T}} < 500$ GeV (left) and $500 < p_{\mathrm{T}} < 700$ GeV (right), middle: $700 < p_{\mathrm{T}} < 900$ GeV (left) and $900 < p_{\mathrm{T}} < 1200$ GeV (right), lower: $1200 < p_{\mathrm{T}} < 1500$ GeV (left) and $p_{\mathrm{T}} > 1500$ GeV (right).

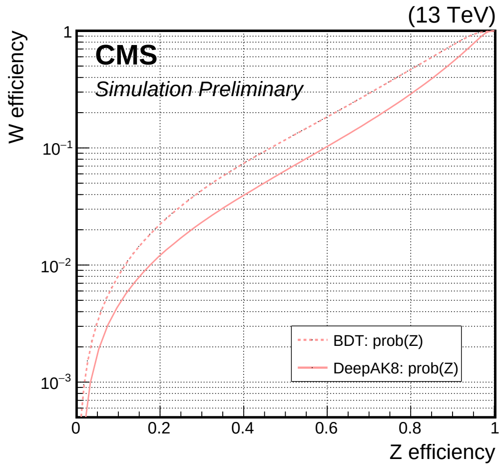

Figure 20: Comparison of the performance of the DeepAK8 (solid line) and BDT (dashed line) multi-taggers to separate W and Z jets in MC simulated events. In this example AK8 jets with $p_{\mathrm{T}} > 400$ GeV are considered.

[5] D. Guest et al., "Jet Flavor Classification in High-Energy Physics with Deep Neural Networks", *Phys. Rev.* **D94** (2016), no. 11, 112002, doi:10.1103/PhysRevD.94.112002, arXiv:1607.08633.

[6] G. Louppe, K. Cho, C. Becot, and K. Cranmer, "QCD-Aware Recursive Neural Networks for Jet Physics", arXiv:1702.00748.

[7] CMS Collaboration, "CMS Phase 1 heavy flavour identification performance and developments", CMS Detector Performance Summary CMS-DPS-17-013, 2017.

[8] ATLAS Collaboration, "Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment", ATLAS note ATL-PHYS-PUB-2017-003, 2017.

[9] P. Nason, "A new method for combining NLO QCD with shower Monte Carlo algorithms", *JHEP* **11** (2004) 040, doi:10.1088/1126-6708/2004/11/040, arXiv:hep-ph/0409146.

[10] S. Frixione, P. Nason, and C. Oleari, "Matching NLO QCD computations with parton shower simulations: the POWHEG method", *JHEP* **11** (2007) 070, doi:10.1088/1126-6708/2007/11/070, arXiv:0709.2092.

[11] S. Alioli, P. Nason, C. Oleari, and E. Re, "A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX", *JHEP* **06** (2010) 043, doi:10.1007/JHEP06(2010)043, arXiv:1002.2581.

[12] S. Alioli, P. Nason, C. Oleari, and E. Re, "NLO single-top production matched with shower in POWHEG: *s*- and *t*-channel contributions", *JHEP* **09** (2009) 111, doi:10.1007/JHEP02(2010)011, 10.1088/1126-6708/2009/09/111, arXiv:0907.4076. [Erratum: JHEP **02** (2010) 011].

[13] E. Re, "Single-top Wt-channel production matched with parton showers using the POWHEG method", *Eur. Phys. J. C* **71** (2011) 1547, doi:10.1140/epjc/s10052-011-1547-z, arXiv:1009.2450.

[14] T. Sjöstrand et al., "An Introduction to PYTHIA 8.2", *Comput. Phys. Commun.* **191** (2015) 159, doi:10.1016/j.cpc.2015.01.024, arXiv:1410.3012.

[15] S. Agostinelli et al., "GEANT4 — a simulation toolkit", *Nucl. Instr. and Meth. A* **506** (2003) 250, doi:10.1016/S0168-9002(03)01368-8.

[16] CMS Collaboration, "TWiki: Jet Flavour Identification (MC Truth), section on physics definition in PAT", cms twiki, 2017.

[17] CMS Collaboration, "ParticleFlow Event Reconstruction in CMS and Performance for Jets, Taus, and E_T^{miss}", CMS Physics Analysis Summary CMS-PAS-PFT-09-001, 2009.

[18] CMS Collaboration, "Identification of b quark jets at the CMS Experiment in the LHC Run 2", CMS Physics Analysis Summary CMS-PAS-BTV-15-001, 2015.

[19] D. Bertolini, P. Harris, M. Low, and N. Tran, "Pileup Per Particle Identification", *JHEP* **10** (2014) 059, doi:10.1007/JHEP10(2014)059, arXiv:1407.6013.

[20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *CoRR* **abs/1412.6980** (2014).

[21] M. V. Jan Kieseler, Markus Stoye, "DeepJet: Framework for the development of deep-neural-network based reconstruction in high-energy-physics", CMS Analysis Note AN 2017/126, 2017.

[22] F. Chollet, "Keras". https://github.com/fchollet/keras,https://keras.io/, 2015.

[23] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems", 2015. Software available from tensorflow.org.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks", *arXiv preprint arXiv:1603.05027* (2016).

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778. 2016.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *International Conference on Machine Learning*, pp. 448–456. 2015.

[27] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines", in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814. 2010.

[28] T. Chen et al., "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems", *arXiv preprint arXiv:1512.01274* (2015).

[29] CMS Collaboration, "Heavy flavor identification at CMS with deep neural networks",.

[30] CMS Collaboration, "Performance of quark/gluon discrimination in 8 TeV pp data", CMS Physics Analysis Summary CMS-PAS-JME-13-002, 2013.

[31] CMS Collaboration, "Search for direct top squark pair production in the all-hadronic final state in proton-proton collisions at sqrt(s) = 13 TeV",. pending submission to JHEP.

[32] SUS-16-049 authors, "Search for direct production of top squark pairs in the fully-hadronic final state with data collected in pp collisions at 13 TeV using the entire 2016 dataset", *CMS-AN-16-437* (2016).

[33] CMS Collaboration, "Identification of double-b quark jets in boosted event topologies",.

[34] CMS Collaboration, "TWiki: Reference for JMAR Heavy Resonance Overview", cms twiki, 2017.

[35] CMS Collaboration, "TWiki: JMAR general page", cms twiki, 2017.

[36] CMS Collaboration, "TWiki: Usage of b/c Tag Objects for 13 TeV Data in 2016 and 80X MC", cms twiki, 2017.

A Slim jet minor labels probabilities

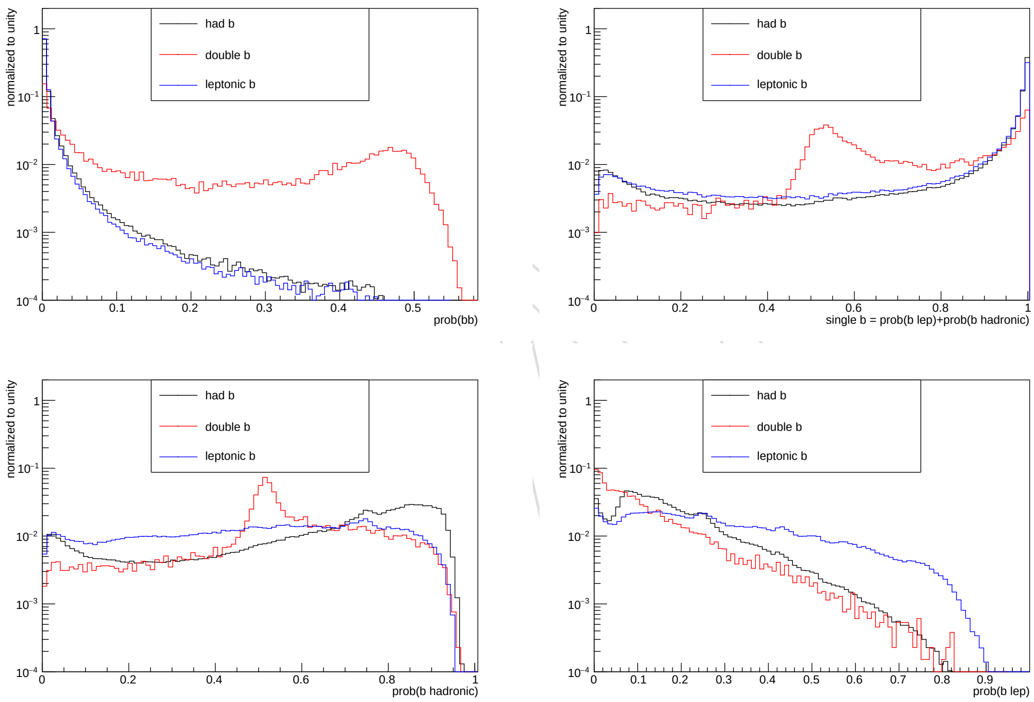

Figure 21: Minor b jet labels separately. Note that no direct lepton information is included in the input. We would not recommend using the separation of leptonic and hadronic single b-hadron jets at this point for analysis purposes.

B Performance plots of DeepAK8 multi-tagger using the JMAR definition

Performance plots of the DeepAK8 and BDT multi-tagger, as well as two “Cut based” working points (low and high purity) for each heavy objects as defined from the corresponding POGs [i.e. JMAR and BTV]. The ROC curves are estimated using the samples and matching definition suggested by JMAR. More details in the main text.

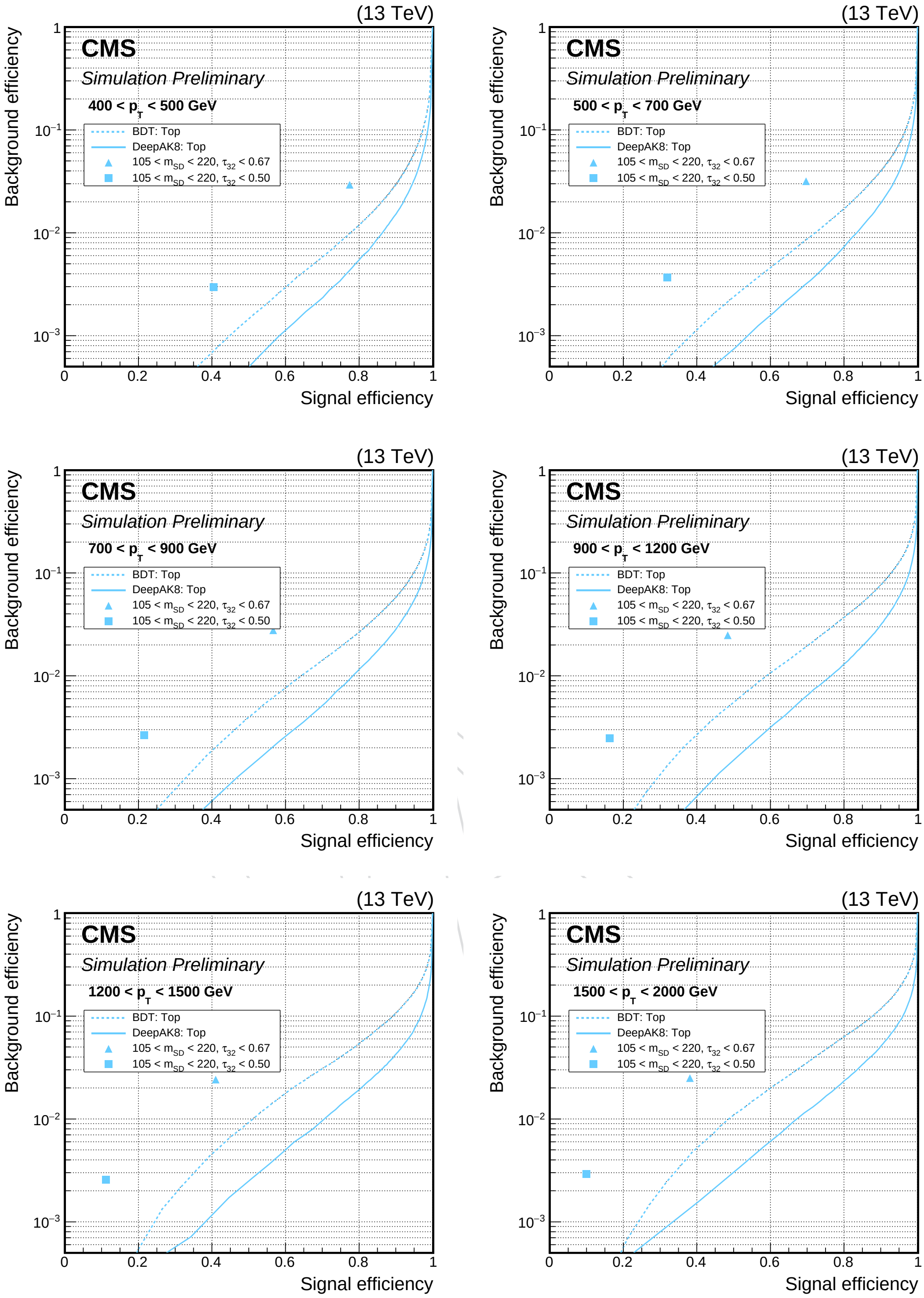

Figure 22: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for t jets as signal and QCD jets as background. Two "cut-based" working points (low and high purity) are included in the plot. The working points are defined by the corresponding POG. The plots correspond to different p_T ranges of the AK8 jet. upper: $400 < p_T < 500$ GeV (left) and $500 < p_T < 700$ GeV (right), middle: $700 < p_T < 900$ GeV (left) and $900 < p_T < 1200$ GeV (right), lower: $1200 < p_T < 1500$ GeV (left) and $p_T > 1500$ GeV (right).

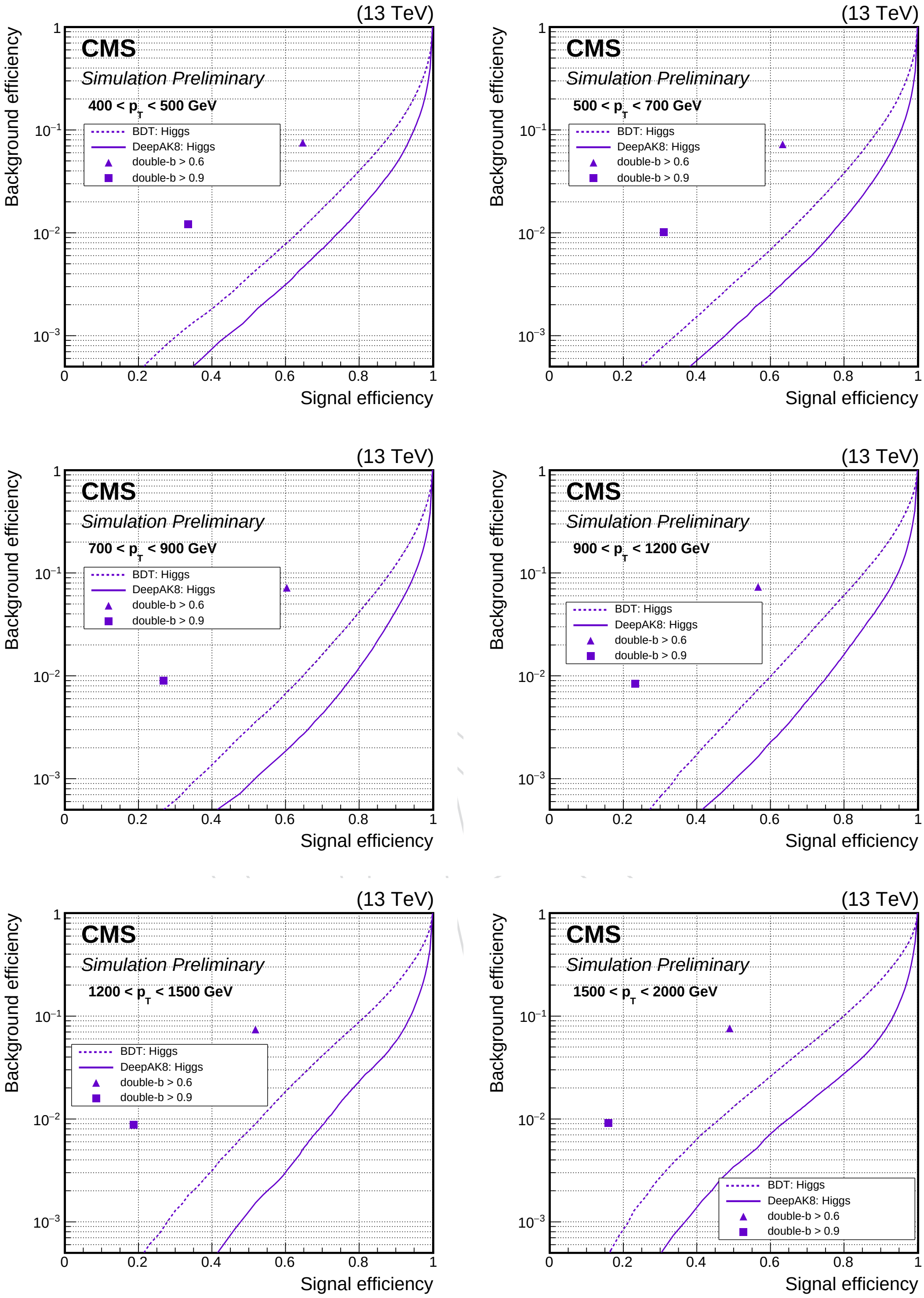

Figure 23: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for H jets as signal and QCD jets as background. Two "cut-based" working points (low and high purity) are included in the plot. The working points are defined by the corresponding POG. The plots correspond to different p_T ranges of the AK8 jet. upper: $400 < p_T < 500$ GeV (left) and $500 < p_T < 700$ GeV (right), middle: $700 < p_T < 900$ GeV (left) and $900 < p_T < 1200$ GeV (right), lower: $1200 < p_T < 1500$ GeV (left) and $p_T > 1500$ GeV (right).

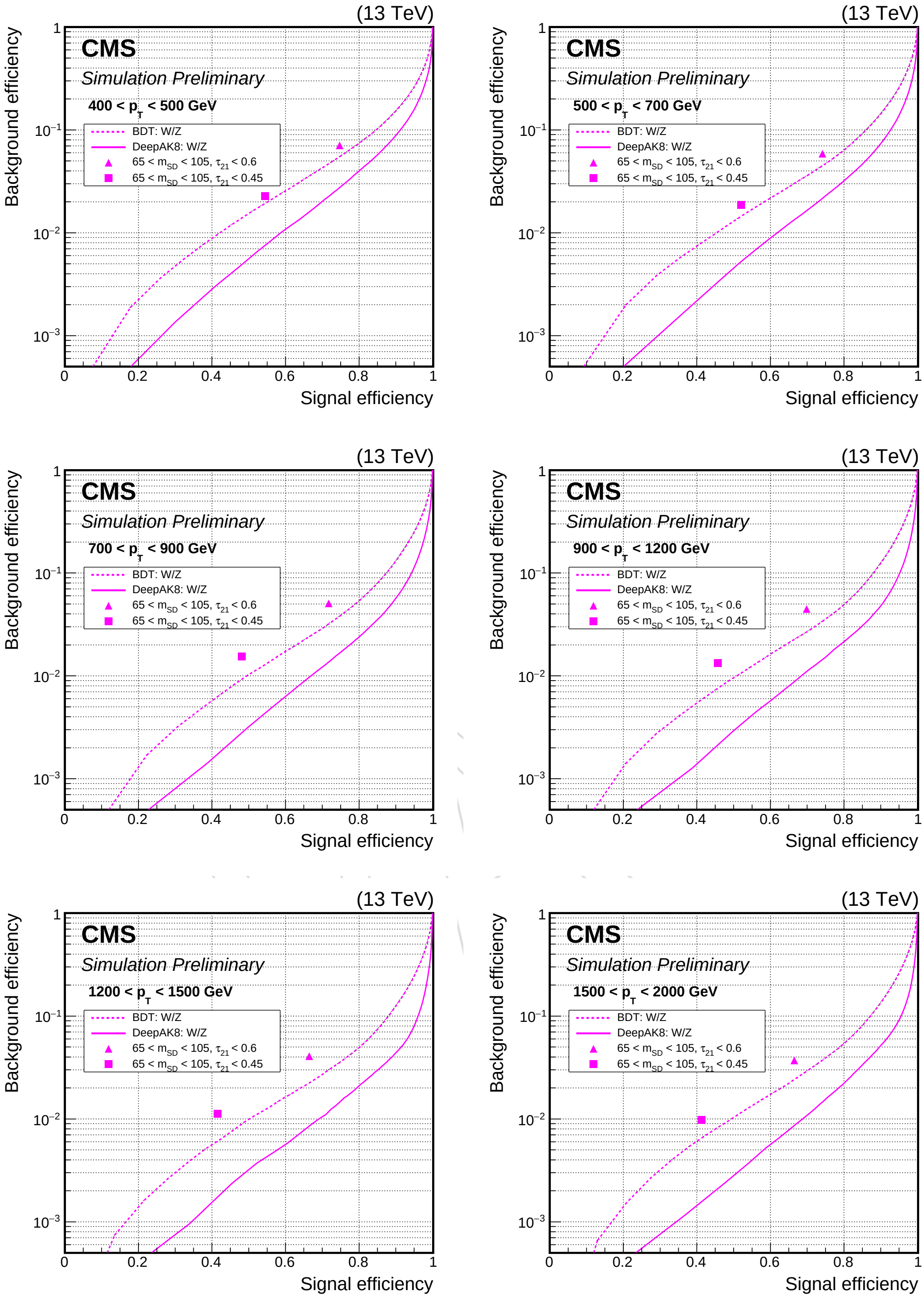

Figure 24: Comparison of the ROC curves obtained with the DeepAK8 multi-tagger (solid lines) and the BDT multi-tagger (dashed lines) in MC simulated events for Z or W jets as signal and QCD jets as background. Two "cut-based" working points (low and high purity) are included in the plot. The working points are defined by the corresponding POG. The plots correspond to different p_{T} ranges of the AK8 jet. upper: $400 < p_{\mathrm{T}} < 500$ GeV (left) and $500 < p_{\mathrm{T}} < 700$ GeV (right), middle: $700 < p_{\mathrm{T}} < 900$ GeV (left) and $900 < p_{\mathrm{T}} < 1200$ GeV (right), lower: $1200 < p_{\mathrm{T}} < 1500$ GeV (left) and $p_{\mathrm{T}} > 1500$ GeV (right).

C Additional plots on the DeepAK8 multi-tagger

C.1 Performance of the DeepAK8 multi-tagger as a function of jet p_T, jet η, and the number of primary vertices

Figure 25 to 28 show the performance of the DeepAK8 multi-tagger as a function of jet p_T, jet η, and the number of primary vertices in the event. We have chosen two working points based on the misidentification rate of 10% (loose) and 1% (tight). We focus on the top, W and QCD classes of the multi-classifier. The matching definition and the samples used to test the performance follow the recommendations from JMAR in [34].

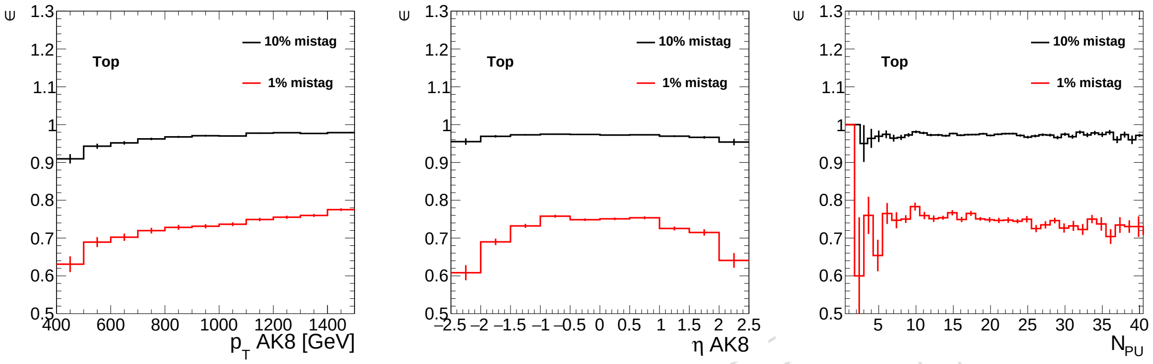

Figure 25: Efficiency of tagging truth-matched top quarks as a function of jet p_T (left), jet η(middle) and the number of primary vertices in the event (right).

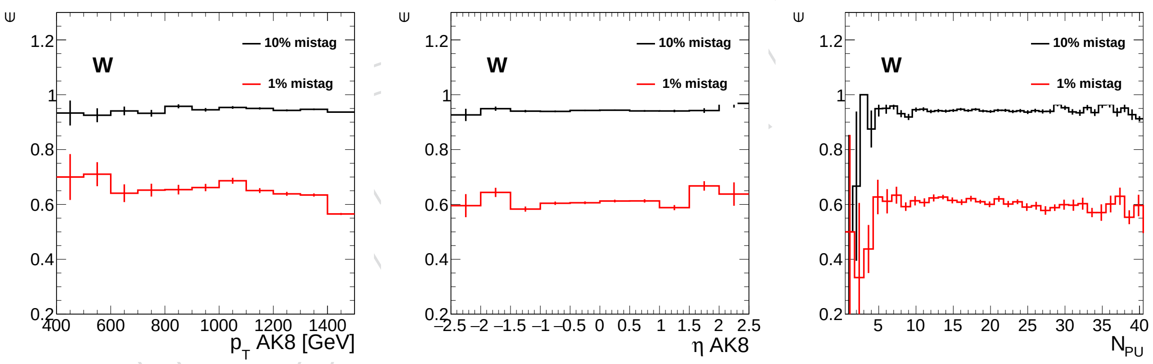

Figure 26: Efficiency of tagging truth-matched W bosons as a function of jet p_T (left), jet η(middle) and the number of primary vertices in the event (right).

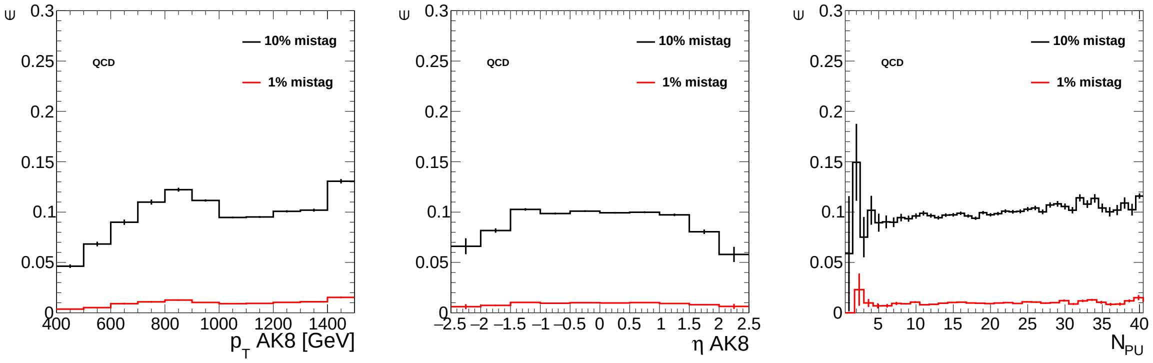

Figure 27: Rate of misidentifying QCD jets as top quarks as a function of jet p_T (left), jet η(middle) and the number of primary vertices in the event (right).

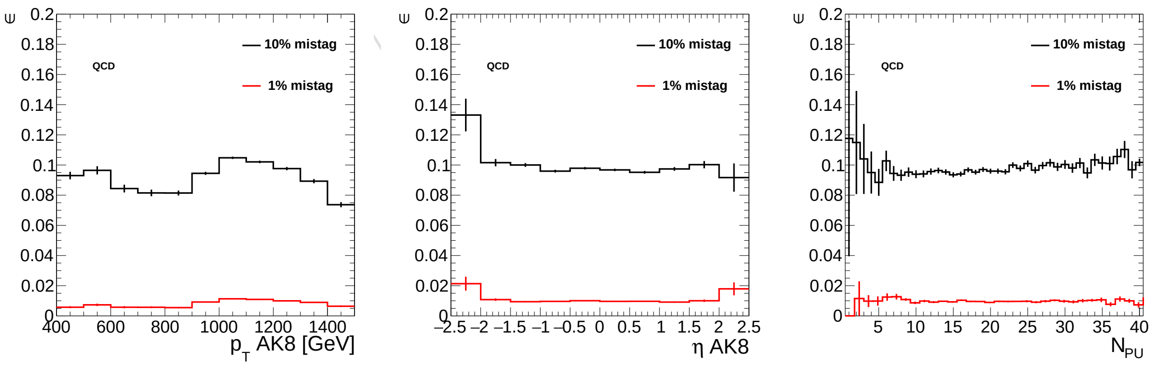

Figure 28: Rate of misidentifying QCD jets as W bosons as a function of jet p_T (left), jet η(middle) and the number of primary vertices in the event (right).

C.2 Correlation between the DeepAK8 multi-tagger and traditional tagging variables

Figure 29 to 32 show the correlation between the DeepAK8 multi-tagger output with traditional jet tagging variables like the N-subjettiness ratios τ_{21}, τ_{32} and the soft-drop mass. We focus on the top, W and QCD classes of the multi-classifier. The matching definition and the samples used to test the performance follow the recommendations from JMAR in [34].

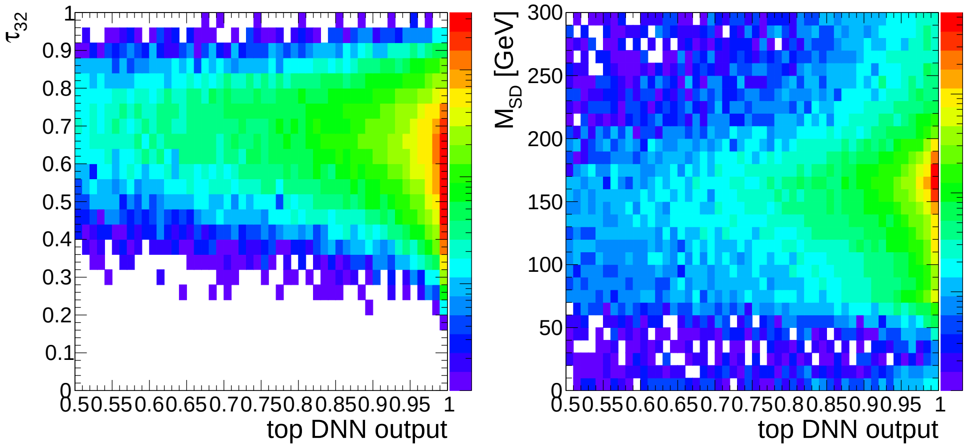

Figure 29: Correlation between the DNN output of the top class and the N-subjettiness ratio τ_{32} (left) and the soft-drop mass M_{SD} (right) in truth-matched top jets.

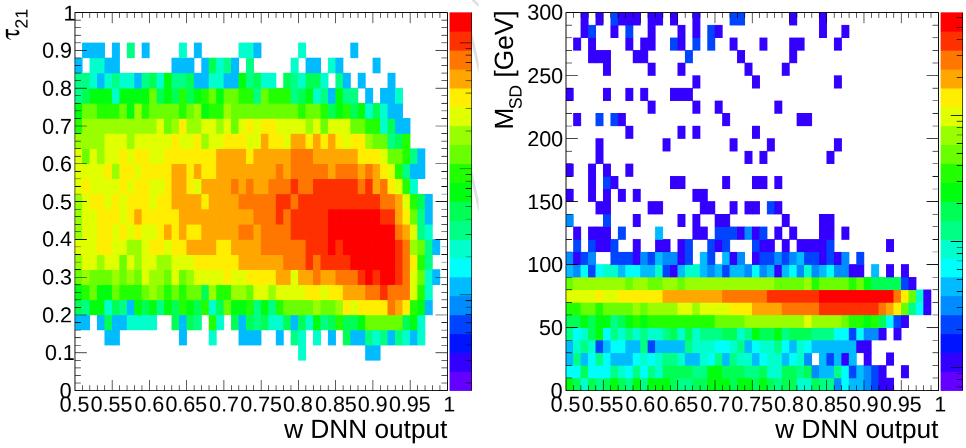

Figure 30: Correlation between the DNN output of the top class and the N-subjettiness ratio τ_{21} (left) and the soft-drop mass M_{SD} (right) in truth-matched W jets.

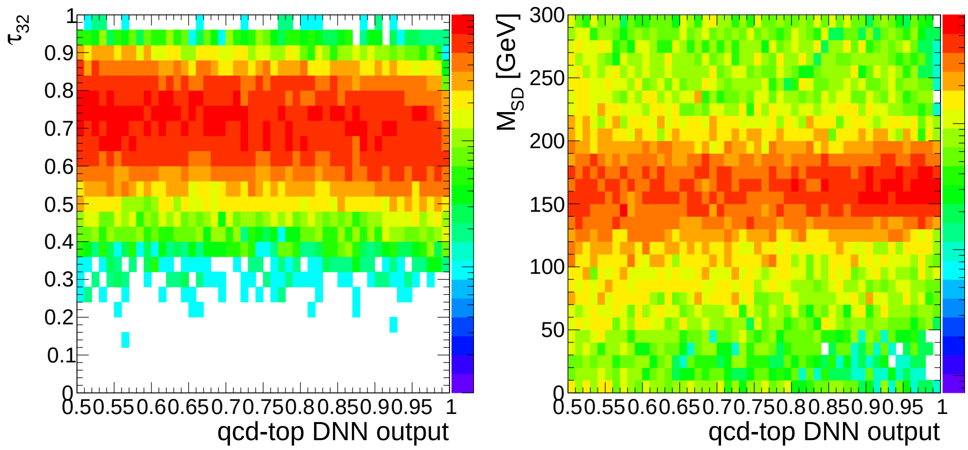

Figure 31: Correlation between the DNN output of the top class and the N-subjettiness ratio τ_{32} (left) and the soft-drop mass M_{SD} (right) in QCD jets.

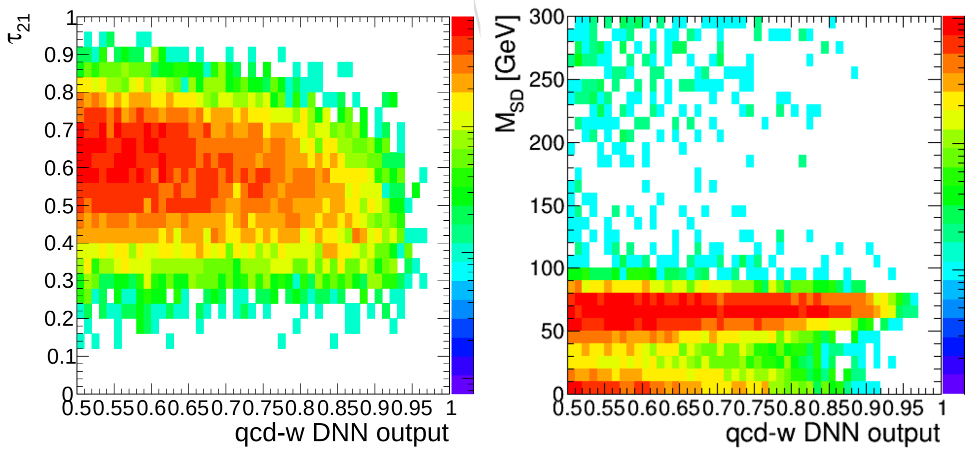

Figure 32: Correlation between the DNN output of the top class and the N-subjettiness ratio τ_{21} (left) and the soft-drop mass M_{SD} (right) in QCD jets.

C.3 Jet mass distribution

Figure 33 and 34 show the soft-drop mass of the jets inclusively and after the loose and tight working points in truth-matched jets and QCD jets, respectively.

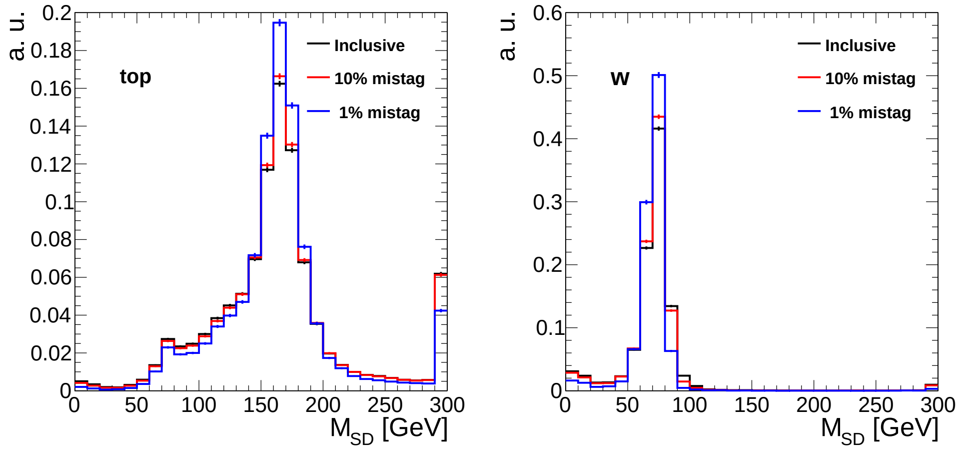

Figure 33: Soft-drop mass of the truth-matched jets inclusively (black line), after passing the loose working point (red line) and after passing the tight working point (blue line). Left: jets matched to top quarks; Right: jets matched to W bosons.

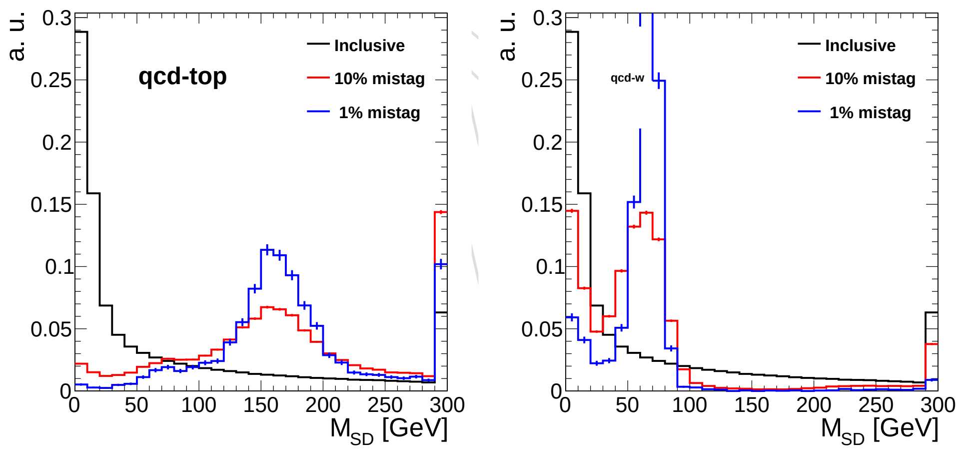

Figure 34: Soft-drop mass of the QCD jets inclusively (black line), after passing the loose working point (red line) and after passing the tight working point (blue line). Left: for top tagging; Right: for W tagging.