Journal of Instrumentation

Jet flavour classification using DeepJet

Published by IOP Publishing for Sissa Medialab

Received: *August 24, 2020*
Revised: *October 5, 2020*
Accepted: *October 27, 2020*
Published: *December 9, 2020*

Jet flavour classification using DeepJet

E. Bols,[a,1] **J. Kieseler,**[b] **M. Verzetti,**[b] **M. Stoye**[c,d] **and A. Stakia**[e]

[a] *Vrije Universiteit Brussels,*
Pleinlaan 2, 1050 Brussels, Belgium
[b] *CERN,*
Espl. des Particules 1, 1211 Meyrin, Switzerland
[c] *Imperial College London,*
South Kensington Campus, London SW7 2AZ, U.K.
[d] *Reexen Technology*
[e] *National Centre for Scientific Research 'Demokritos',*
Patr. Gregoriou E & 27 Neapoleos Str, Athens, Greece

E-mail: emil.bols@cern.ch

Abstract: Jet flavour classification is of paramount importance for a broad range of applications in modern-day high-energy-physics experiments, particularly at the LHC. In this paper we propose a novel architecture for this task that exploits modern deep learning techniques. This new model, called DeepJet, overcomes the limitations in input size that affected previous approaches. As a result, the heavy flavour classification performance improves, and the model is extended to also perform quark-gluon tagging.

Keywords: Analysis and statistical methods; Pattern recognition, cluster finding, calibration and fitting methods; Data processing methods

ArXiv ePrint: 2008.10519

[1] Corresponding author.

https://doi.org/10.1088/1748-0221/15/12/P12012

2020 JINST 15 P12012

Contents

2020 JINST 15 P12012

1 Introduction

The Standard Model of particle physics (SM) [1, 2] is a remarkably effective theory, able to describe the experimental observations made thus far in high energy physics with unprecedented precision and completeness. Despite its success however, this model fails to explain several observations like the baryon asymmetry and the presence of dark matter, which inspires searches for extensions to the SM. The study of the recently discovered [3–5] Higgs boson [6–11], and the search for extensions of the electroweak sector are two of the most active research sectors in the field.

Because of the flavour asymmetry associated to production and decay processes in each case, the ability to classify jets originating from heavy-flavour (bottom and charm) quarks is important. Heavy-flavour (HF) jets contain an open-bottom or open-charm hadron as a result of the fragmentation process. This hadron carries a large fraction of the initial parton momentum. HF hadrons also have a sizeable lifetime, with a $c\tau$ of ~ 0.5 mm and ~ 0.3 mm for bottom and charm, respectively. Most of the HF hadrons produced in the fragmentation process undergo a decay process far enough

from the primary interaction vertex (PV) to result in displaced tracks and their decay products can be clustered in resolved secondary vertices (SV).

Traditionally, jet flavour classification has leveraged both fragmentation and lifetime properties of HF hadrons. The discrimination power provided by single features is not adequate for a both efficient and pure classification, but the collective behaviour of the tracks in the jet, as well as the presence of a reconstructed secondary vertex, allows for a good performance of a combined algorithm. For this reason, machine learning has long been used in the field of jet flavour classification, starting from simple naive bayes classifiers [12, 13], to more complex and modern models like shallow neural networks [14, 15], or tree ensembles [14].

More recently, simple dense deep neural networks [16] have been successfully employed for this classification task alongside recurrent neural networks (LSTM) [17], outperforming previous classifiers. The ATLAS *RNN* algorithm [18], which is based on a recurrent neural network, processes as a sequence a small number of selected high purity charged particle tracks associated with the jet, sorted by their impact parameter. It exceeds the performance of the IP3D algorithm [15], which considers the same inputs, but does not fully take into account the correlations between the tracks.

Within CMS, the DNN based b-tagger DeepCSV [19] has been used as the baseline reference for b-tagging performance for quite a while. DeepCSV is a fully connected neural network consisting of 5 layers with 100 nodes each using high level features of selected tracks and vertices as input. In contrast with the ATLAS RNN tagger, DeepCSV combines properties from selected tracks and secondary vertices directly, but does not employ sequence processing.

The common feature between these two approaches, however, is that both models use only a small subset of the charged jet constituents that pass stringent quality criteria in order to provide the cleanest and simplest environment possible for the classifier to perform its task.

A stringent selection comes with information loss and therefore potential performance degradation. It has recently been shown that architectures compressing and converting individual track features to a latent space can overcome these limitations. The ATLAS Deep Sets b-tagging algorithm [20] relying on the idea of Deep Sets [21], where the features of the input tracks are transformed into a latent space representation before being combined by a simple per-feature sum, has been shown to have similar performance as the RNN approach with reduced training and inference time given the same inputs. Studies on relaxing the input selection criteria on the tracks show a significant gain.

In this paper, we describe DeepJet, a new network architecture that does not rely on a selection of the jet constituents and thus overcomes the previous limitations on the purity and number of inputs. In addition, DeepJet considers the full information of all jet constituents, charged and neutral particles, secondary vertices, and global event variables simultaneously.

2 Setup

2.1 Training Samples

The DeepJet model is trained and tested using a sample of simulated anti k_T [22] jets (with $R = 0.4$) drawn from a mix of QCD multijet events and fully hadronic top-quark pair ($t\bar{t}$) events, generated with PYTHIA8 [23] and POWHEGv2 [24–27], respectively. In both samples the hadronization and showering are performed using PYTHIA8. In order to apply the approach to a realistic reconstruction

2020 JINST 15 P12012

problem, GEANT4 [28] is used to simulate the response of the CMS Phase 1 detector [29, 30] to the generated particles. The jet constituents are reconstructed using the Particle Flow (PF) algorithm [31] within the CMS Phase 1 detector reconstruction algorithms [32]. The entire training sample consists of roughly 130 million jets in total. This is then split into a training, a validation and a testing sample in the ratio 0.765 : 0.135 : 0.1.

2.2 Labeling

The jets are labelled by *ghost association* [19]. In this approach the last instance of each generated b and c hadron before the decay have their momentum scaled to a very small value, in order to carry only directional information. These "ghosts" are then included among the particles to be clustered by the jet algorithm. If a jet contains at least one b hadron, it is labeled a b jet. If the jet contains at least one c hadron, and no b hadrons, it is labelled a c jet. Everything else is then labelled as a light jet. The jets labelled as b jets are further sub-labelled into three categories: bb for jets containing two b hadrons, b_{lept} for jets containing b hadrons decaying leptonically, and b for jets with b hadrons decaying hadronically. The light jets are also sub-labelled into quark and gluon jets using the CMS definition [33].

2.3 Input features

DeepJet uses approximately 650 input variables, divided into four categories: global variables, charged PF candidate features, neutral PF candidate features, and SV features associated with the jet. The global variables include jet-level information such as the jet kinematics, the number of tracks in the jet, as well as the number of secondary vertices in the jets. Additionally, the global variables also include the number of reconstructed primary vertices in the event, in order to help the network learn the effect of multiple interactions per bunch crossing (pileup). The information related to the charged PF candidates is summarised in 16 features of the first 25 candidates with respect to their displacement significance. These features contain information on the track kinematics, track fit quality, displacement, and displacement uncertainty with respect to the PV. Similarly, 6 features of the leading 25 neutral candidates are provided to the network. Finally, up to 4 secondary vertices are considered, each with 12 variables, such as the flight distance significance. A full list of the variables used can be found in the appendix.

2.4 Preprocessing of features

Since the b-tagger is meant to generalise beyond the specific kinematic domain used for the training, it is important to remove any flavour dependence on transverse momentum (p_T), or pseudorapidity (η) to avoid the classifier leveraging those differences. During the data pre-processing, we downsample each of the flavour classes in p_T and η such that they have the same shape as the b jet class. Several variables are also bounded within a physically well defined range. In case features of an object are either not available or infinite, they are replaced with an appropriately chosen value for the specific feature. The particle objects are ordered based on their assumed importance. Charged particle candidates are ordered by impact parameter significance, while neutral candidates are ordered by the shortest angular distance to a secondary vertex. If there is no secondary vertex in the jet, p_T ordering is used instead. Secondary vertices are sorted by flight distance significance.

2020 JINST 15 P12012

3 DeepJet

3.1 Architecture

The concept of DeepJet is to use low-level features from as many jet constituents as possible as opposed to selecting a few that are well identified and reconstructed and exploiting higher level features. The exact number of features and constituents can vary with experiment and use case, but the all-inclusive concept remains the same. To process this large input space, a suitable architecture is needed. An illustration of the DeepJet architecture can be seen in figure 1. In the first step, an automatic feature engineering is performed for each constituent using convolutional branches comprising 1x1 convolutional layers [34]. The filter size of 1x1 is chosen as inherently all candidates should undergo the same feature transformation without taking into account the other constituents in the jets, and a priori a specific class of physics objects should be treated in a consistent manner independent of input ordering.[1] Separate convolutional branches are used for vertices, charged PF candidates and neutral PF candidates. The convolutional branches use a sequences of layers with a decreasing number of filters, as this projects and compresses the features of each constituent into a lower dimensional space. Each of the outputs are then fed into a dedicated recurrent layer of the LSTM type. This treats the constituents as a sequence, following the order described in section 2.4. In particular the LSTM nodes are well suited for dealing with the variable length sequences of inputs, which occurs in jet physics. The charged candidate LSTM uses 150 nodes, whereas the neutral LSTM and the secondary vertex LSTM each use 50 nodes. The three LSTM outputs are concatenated with the global features of the jet and then fed into a fully connected layer with 200 nodes, followed by 7 fully connected layers with 100 nodes each. Finally 6 outputs nodes are integrated into the multi-classifier, allowing it to perform b-tagging, c-tagging and quark/gluon tagging. Throughout the network ReLU [35] activation functions are used, except for the output layer, where the softmax activation function is applied.

At the beginning of the network, and in between each layer, batch normalization [36] is performed, and dropout [37] with a rate of 0.1 is applied for regularisation.

3.2 Training procedure

The neural network is trained using the Adam optimizer [38] with a learning rate of $3 \cdot 10^{-4}$ for 65 epochs and a categorical cross entropy loss. The learning rate is also halved if the validation sample loss stagnates for more than 10 epochs. After the first training epoch, the input batch normalization layer is frozen. Possible over-training is monitored through the validation loss, which is mostly dominated by medium p_T jets, and additionally through ROC curves in different p_T ranges. The training is performed on an Nvidia GEFORCE GTX 1080 Ti GPU. With a training sample size of roughly 130 million examples, the training convergence time is around 3 days.

4 Performance

4.1 Comparison with other b-tagging algorithms

The overall performance of DeepJet is compared to the CMS b-tagger DeepCSV, which is a multi-classifier embedded in the CMS reconstruction framework, and which uses similar, but strongly

[1] This concept is closely related to the idea of Deep Sets [21].

2020 JINST 15 P12012

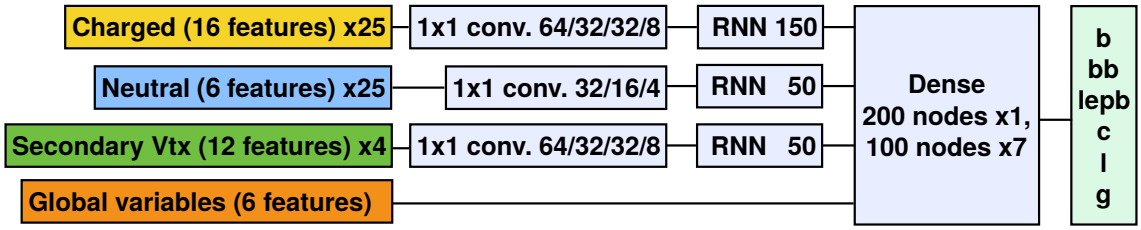

Figure 1. An illustration of the DeepJet architecture. Three seperate branches are used to process charged candidates, neutral candidates and secondary vertices. The algorithm makes use of 1x1 convolutional layers to perform automatic feature engineering for each class of jet constituents. The three RNN (LSTM) layers combine the information for each sequence of constituents. Finally the full jet information is combined using fully connected layers.

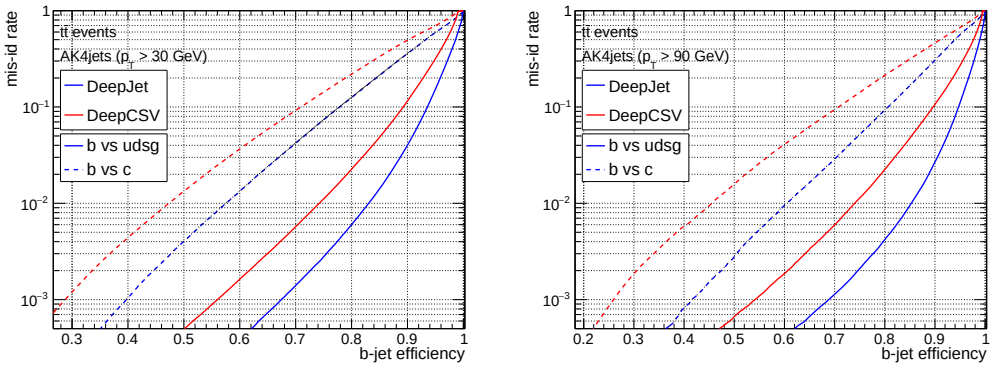

Figure 2. Performance of the DeepJet and DeepCSV b-tagging algorithms on $t\bar{t}$ events with both top quarks decaying hadronically. The jets are required to have $p_T > 30\,GeV$ (left) and $p_T > 90\,GeV$ (right). The performance is shown for both b vs. c classification (dashed lines), and b vs. light (solid lines).

preselected, inputs along with additional high-level variables. The comparison is made on a fully hadronic $t\bar{t}$ sample, as shown in figure 2. DeepJet performs significantly better than DeepCSV with an efficiency[2] increase of almost 20% at 10^{-3} misidentification probability[3] for jets with $p_T > 90\,GeV$.

The performance of the classifiers can also be compared on a set of multi-jet QCD events, which allow accessing higher jet momenta. The inclusive results are shown in figure 3. A comparison for different jet p_T can be found in figure 4 for fixed light jet misidentification probabilities. In both cases DeepJet shows an increasing performance gain, in particular for higher jet p_T, presumably exploiting the information contained in all constituents.

The multi-classifier nature of DeepJet and DeepCSV allows the models to also efficiently perform c jet identification. Figure 5 shows that DeepJet outperforms DeepCSV also in this task. In this case, both DeepCSV and DeepJet use a binary discriminator defined as $\frac{P(c)}{P(c)+P(uds)+P(g)}$, where $P(c)$, $P(uds)$, and $P(g)$ describe the classifier output for c, light quark (u, d and s quark), and gluon jets, respectively.

[2]True Positive Rate.
[3]False Positive Rate.

2020 JINST 15 P12012

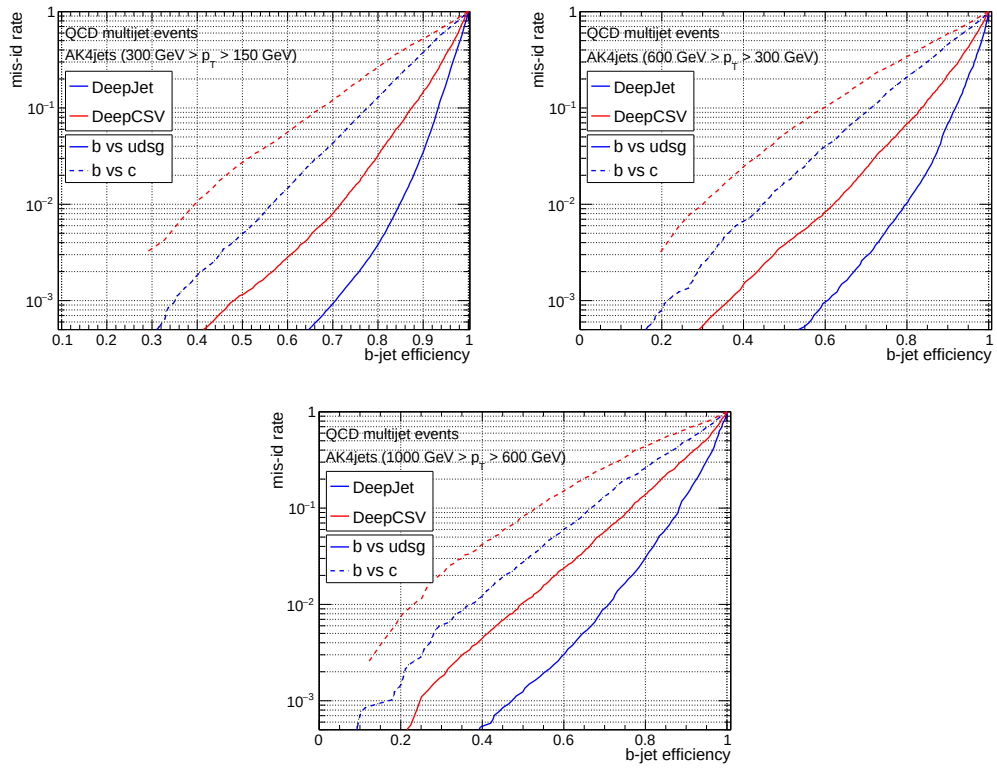

Figure 3. The performance of DeepJet and DeepCSV in three different jet p_T ranges in QCD multijet events. The performance is shown for both b vs. c (dashed lines), and b vs. light (solid lines).

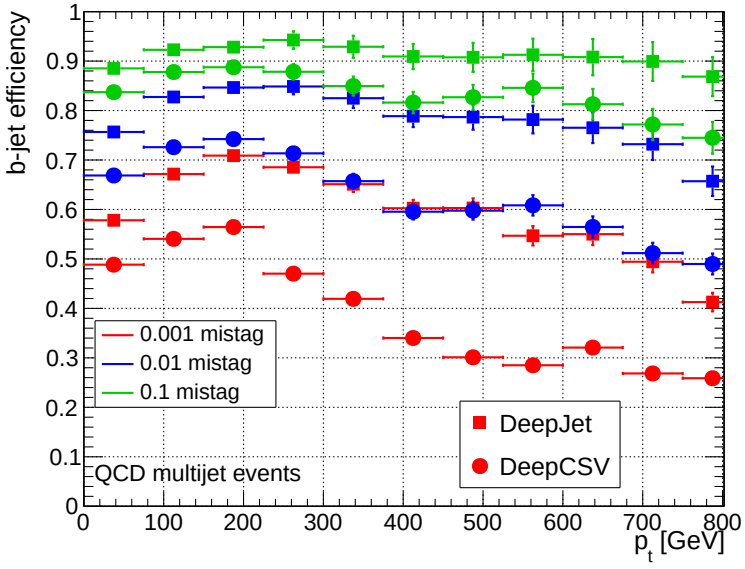

Figure 4. Performance of DeepJet and DeepCSV as a function of jet p_T. The efficiency for b jets is shown for three different fixed light jet misidentification probabilities (mistag rates) using QCD multijet events.

2020 JINST 15 P12012

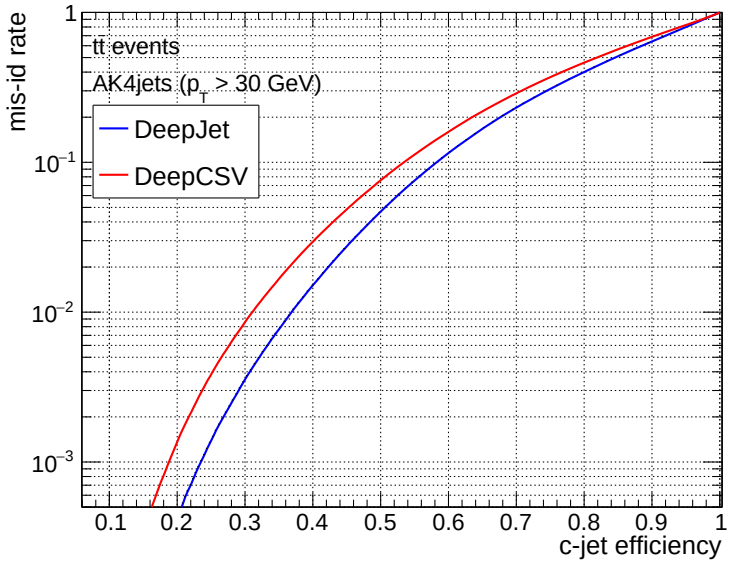

Figure 5. Light vs. c jet classification performance of DeepJet and DeepCSV in fully hadronic $t\bar{t}$ events.

Finally, DeepJet can be used for quark/gluon discrimination. DeepJet is compared to the "quark/gluon likelihood" [33], a binary quark/gluon classifier included in the CMS reconstruction framework. For this purpose, the DeepJet output is also projected to a binary classifier, $\frac{P(\text{uds})}{P(\text{uds})+P(\text{g})}$. Also here, a performance increase can be observed when employing the DeepJet model on a sample consisting purely of light quark and gluon jets, as shown in figure 6. Moreover, given that a standard quark/gluon classifier uses a binary approach, we expect a larger performance gain in scenarios where gluon jets and light quark jets also need to be separated from heavy flavour jets.

The effects of multiple interactions per bunch crossing (pileup) can further degrade the b-tagging performance. In figure 7 the b-tagging performance is shown as a function of number of reconstructed primary vertices for both DeepJet and DeepCSV. Despite the use of more low purity input in DeepJet, a similar level of performance degradation is observed in both algorithms when increasing pileup.

4.2 Qualitative assessment of the performance gain source

Given that the architecture and the network inputs are changed simultaneously between DeepCSV and DeepJet, two new network configurations are trained to assess the relative impact of these changes to the performance. The first model, referred to as DeepCSV RNN, contains the same inputs as DeepCSV, but its architecture is changed to mimic that of DeepJet, with a series of convolutional layers feeding into a LSTM processing the list of inputs belonging to different tracks. The second model uses the architecture of DeepCSV but uses the full input of DeepJet. The performance of these models are compared with DeepCSV and DeepJet in figure 8 for inclusive $t\bar{t}$ events. Increasing the number of inputs does not help much, when it is not complemented by a network structure that allows efficient processing of the larger and less pure track set. However adopting the new architecture without increasing the input and removing the track selection also has limited gain. Similar

2020 JINST 15 P12012

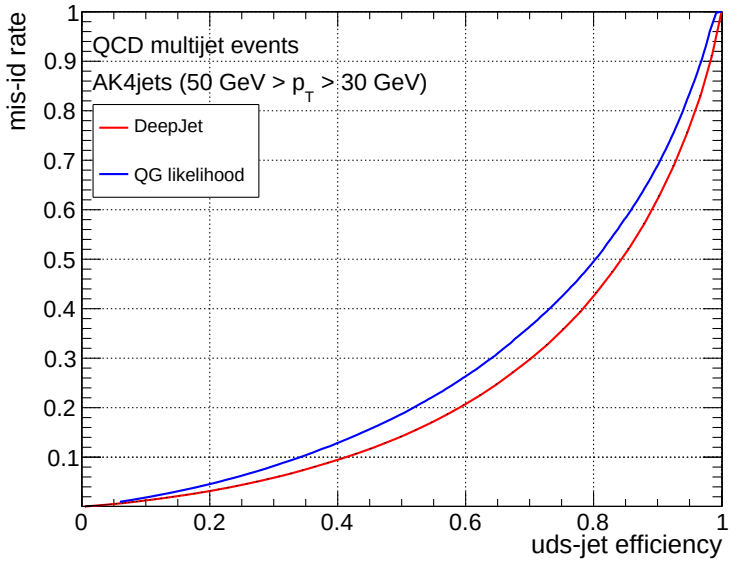

Figure 6. Quark gluon discrimination performance of DeepJet compared to the CMS "quark-gluon likelihood" method in a sample of pure light quark (uds) and gluon jets.

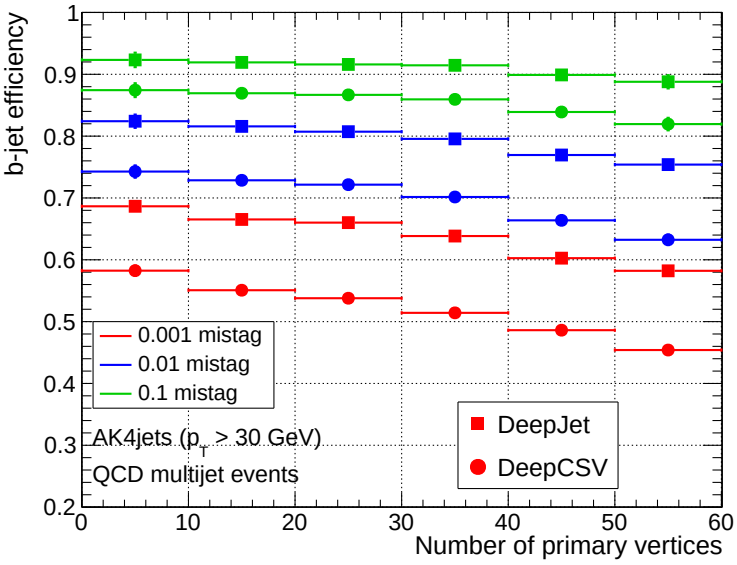

Figure 7. Performance of DeepJet and DeepCSV as a function of reconstructed primary vertices. The efficiency for b jets is shown for three different fixed light jet misidentification probabilities (mistag rates) using QCD multijet events.

2020 JINST 15 P12012

improvements coming with similar preprocessing of unselected jet constituent inputs could be observed in the Deep Set based tagger [20]. This qualitative statement also explains the additional gain observed in DeepJet at high jet p_T as the track selection was historically optimised for mild jet boosts.

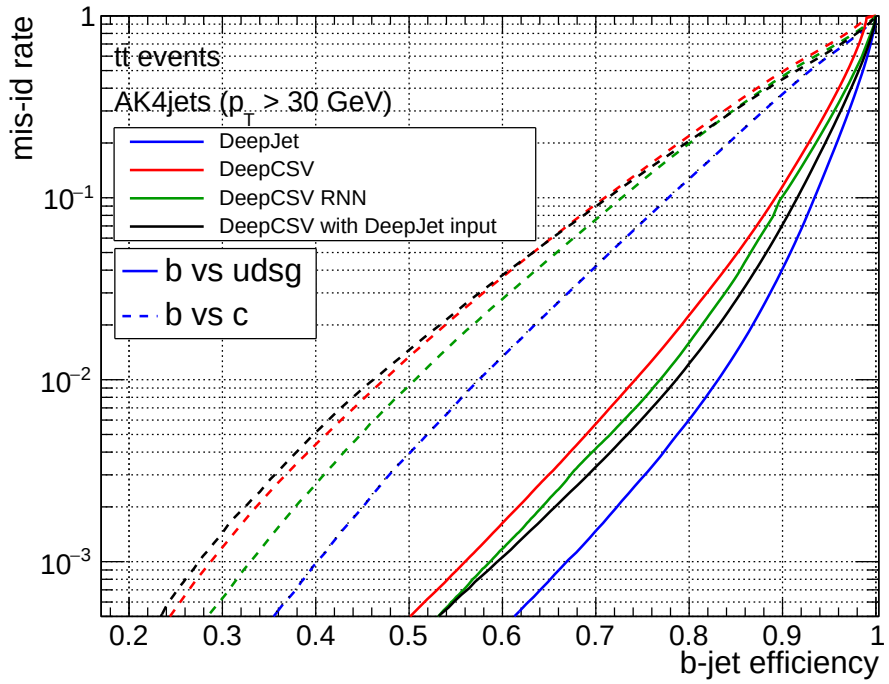

Figure 8. An interpolation between the DeepCSV tagger [19] and DeepJet.

The default DeepJet architecture is also compared to an adapted architecture employing the Deep Set idea. In the DeepSet approach it is required to project the single objects into a high dimensional feature space, whereas the DeepJet model compresses the features. To mimic DeepSet, the DeepJet architecture is adapted as follows: the 1×1 convolutional branches are extended to contain 100 filters, each, increasing the number of free parameters. The final layer of each branch is enlarged further, with the charged branch being set to 256 filters, whereas the vertex and the neutral branch are set to 128 filters. At the same time, the LSTM layers are replaced by a simple sum that is evaluated independently for each convolutional branch, after which these sums are concatenated and fed to a series of fully connected layers. Another comparison model is added that shows a DeepJet model with the same enlarged convolutional structure as the DeepSet model, but keeping the LSTM for aggregating the tracks.

As shown in figure 9, the performance of the Deep Set based architecture shows a slight gain for b to c jet identification, but shows no significant difference with respect to the b to light jet identification or the quark/gluon discrimination performance compared to the default DeepJet architecture. The DeepJet model using enlarged convectional filters performs slightly better than DeepSet, but like DeepSet, it projects the input objects into large features spaces, leading to increased resource requirements for the subsequent LSTMs.

2020 JINST 15 P12012

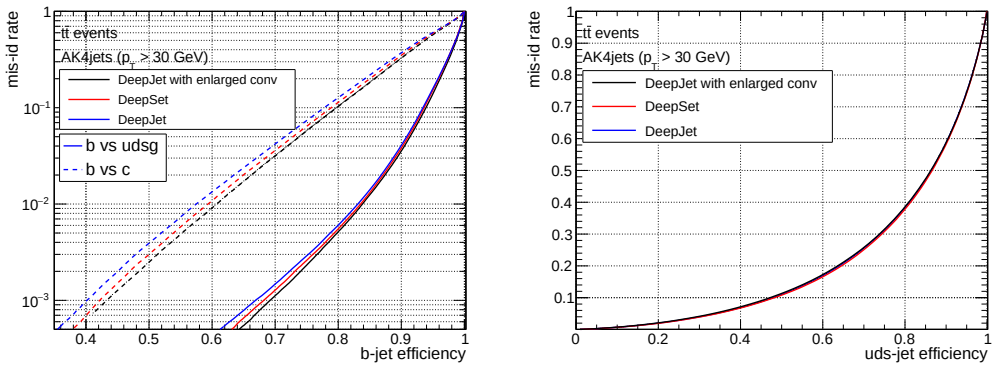

Figure 9. Comparison of the DeepJet architecture, an adapted DeepJet architecture based on the Deep Sets approach, and a DeepJet model with enlarged number of convolutional filters. Left: b jet identification performance. Right: quark/gluon jet separation performance.

5 Conclusion

A multiclass flavour tagging algorithm, DeepJet, has been developed to exploit the full information in a jet. The model was tested using CMS simulation. The model relies on low-level variables and loose selection of the inputs, and it uses an architecture that can process these inputs in an efficient manner. When compared with fully connected models using a smaller set of engineered features a gain in performance is observed in all topologies of flavour tagging, in some cases exceeding a two-fold efficiency gain for the same misidentification rate. The model goes beyond heavy flavour tagging and performs quark-gluon discrimination as well. The applicability of the DeepJet approach has been successfully extended in different directions, such as the construction of tagging algorithms for exotic long lived particles [39].

DeepJet has been trained using the homonymous github package [40, 41], with a streamlined training process and multi-threaded data access. The training set has been scraped from the simulation data using [42], which contains an algorithmic description of all the variables used in the training.

Acknowledgments

We thank our colleagues within the CMS collaboration for the ability to use the CMS simulation, and especially the CMS b-tagging and vertexing community for their continuous support to integrate DeepJet in the CMS framework. The training of the model was performed on the GPU clusters of the CERN EP/CMG group.

A Appendices: Inputs

A.1 List of global variables

- Jet p_t
- Jet η

2020 JINST 15 P12012

- The number of charged particle flow candidates in the jet
- The number of neutral particle flow candidates in the jet
- The number of secondary vertices in the jet
- The number of primary vertices in the event

A.2 List of charged candidate variables

- Charged track η relative to the jet axis
- Charged track p_t relative to the jet axis
- Dot product of the jet and track momentum
- Dot product of the jet and track momentum divided by the magnitude of the jet momentum
- ΔR between the jet axis and the track
- The track 2D impact parameter value
- The track 2D impact parameter significance
- The track 3D impact parameter value
- The track 3D impact parameter significance
- The track distance to the jet axis
- Fraction of the jet momentum carried by the track.
- ΔR between the track and the closest secondary vertex
- An integer flag that indicate whether the track was used in the primary vertex fit.
- The charged candidates PUPPI weight
- χ^2 of the charged track fit.
- A integer flag which indicate the quality of the fitted track, based on number of detector hits used for the reconstruction as well as the overall χ^2 of the charged track fit.

A.3 List of neutral candidate variables

- Fraction of the jet momentum carried by the neutral candidate
- ΔR between the jet axis and the neutral candidate
- A integer flag indicating whether the neutral candidate is a photon.
- Fraction of the neutral candidate energy deposited in the hadronic calorimeter.
- ΔR between the neutral candidate and the closest secondary vertex
- The neutral candidates PUPPI weight

2020 JINST 15 P12012

A.4 List of secondary vertex variables

- Secondary vertex p_t
- ΔR between the jet axis and the secondary vertex
- Secondary vertex mass
- Number of tracks in the secondary vertex
- χ^2 of the secondary vertex fit
- Reduced χ^2 of the secondary vertex fit
- The secondary vertex 2D impact parameter value
- The secondary vertex 2D impact parameter significance
- The secondary vertex 3D impact parameter value
- The secondary vertex 3D impact parameter significance
- Cosine of the angle between the secondary vertex flight direction and the direction of the secondary vertex momentum.
- Ratio of the secondary vertex energy to the jet energy

References

[1] S. Weinberg, *A Model of Leptons*, *Phys. Rev. Lett.* **19** (1967) 1264.

[2] A. Salam, *Weak and Electromagnetic Interactions*, *Conf. Proc. C* **680519** (1968) 367.

[3] CMS collaboration, *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, *Phys. Lett. B* **716** (2012) 30 [arXiv:1207.7235].

[4] CMS collaboration, *Observation of a New Boson with Mass Near 125 GeV in pp Collisions at* $\sqrt{s} = 7$ *and 8 TeV*, *JHEP* **06** (2013) 081 [arXiv:1303.4571].

[5] ATLAS collaboration, *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett. B* **716** (2012) 1 [arXiv:1207.7214].

[6] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, *Phys. Rev. Lett.* **13** (1964) 321.

[7] P.W. Higgs, *Broken symmetries, massless particles and gauge fields*, *Phys. Lett.* **12** (1964) 132.

[8] P.W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, *Phys. Rev. Lett.* **13** (1964) 508.

[9] G.S. Guralnik, C.R. Hagen and T.W.B. Kibble, *Global Conservation Laws and Massless Particles*, *Phys. Rev. Lett.* **13** (1964) 585.

[10] P.W. Higgs, *Spontaneous Symmetry Breakdown without Massless Bosons*, *Phys. Rev.* **145** (1966) 1156.

[11] T.W.B. Kibble, *Symmetry breaking in nonAbelian gauge theories*, *Phys. Rev.* **155** (1967) 1554.

[12] CMS collaboration, *Algorithms for b Jet identification in CMS*, Tech. Rep., CMS-PAS-BTV-09-001, CERN, Geneva (2009).

2020 JINST 15 P12012

[13] CMS collaboration, *Identification of b-Quark Jets with the CMS Experiment*, 2013 *JINST* **8** P04013 [arXiv:1211.4462].

[14] CMS collaboration, *Identification of b quark jets at the CMS Experiment in the LHC Run 2*, Tech. Rep., CMS-PAS-BTV-15-001, CERN, Geneva (2016).

[15] ATLAS collaboration, *Performance of b-Jet Identification in the ATLAS Experiment*, 2016 *JINST* **11** P04008 [arXiv:1512.01094].

[16] CMS collaboration, *Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV*, 2018 *JINST* **13** P05011 [arXiv:1712.07158].

[17] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural Comput.* **9** (1997) 1735.

[18] ATLAS collaboration, *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment*, Tech. Rep., ATL-PHYS-PUB-2017-003, CERN, Geneva (2017).

[19] CMS collaboration, *Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV*, 2018 *JINST* **13** P05011 [arXiv:1712.07158].

[20] ATLAS collaboration, *Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS*, Tech. Rep., ATL-PHYS-PUB-2020-014, CERN, Geneva (2020).

[21] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov and A.J. Smola, *Deep sets* in *Proceedings of Advances in Neural Information Processing Systems 30*, Long Beach, CA, U.S.A., 4–9 December 2017, pp. 3391–3401 [arXiv:1703.06114].

[22] M. Cacciari, G.P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, *JHEP* **04** (2008) 063 [arXiv:0802.1189].

[23] T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159 [arXiv:1410.3012].

[24] P. Nason, *A New method for combining NLO QCD with shower Monte Carlo algorithms*, *JHEP* **11** (2004) 040 [hep-ph/0409146].

[25] 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 [arXiv:1002.2581].

[26] S. Frixione, P. Nason and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, *JHEP* **11** (2007) 070 [arXiv:0709.2092].

[27] J.M. Campbell, R.K. Ellis, P. Nason and E. Re, *Top-Pair Production and Decay at NLO Matched with Parton Showers*, *JHEP* **04** (2015) 114 [arXiv:1412.1828].

[28] GEANT4 collaboration, *GEANT4–a simulation toolkit*, *Nucl. Instrum. Meth. A* **506** (2003) 250.

[29] CMS collaboration, *The CMS Experiment at the CERN LHC*, 2008 *JINST* **3** S08004.

[30] CMS collaboration, D.A. Matzner Dominguez et al., eds., *CMS Technical Design Report for the Pixel Detector Upgrade*,.

[31] CMS collaboration, *Particle-flow reconstruction and global event description with the CMS detector*, 2017 *JINST* **12** P10003 [arXiv:1706.04965].

[32] CMS collaboration, *CMS Software (CMSSW)*, https://github.com/cms-sw/cmssw.

[33] CMS collaboration, *Jet algorithms performance in 13 TeV data*, Tech. Rep., CMS-PAS-JME-16-003, CERN, Geneva (2017).

[34] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, in *Intelligent Signal Processing*, pp. 306–351, IEEE Press (2001).

2020 JINST 15 P12012

[35] V. Nair and G. Hinton, *Rectified linear units improve restricted boltzmann machines*, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Haifa, Israel, 21–24 June 2010,pp. 807–814, 2010.

[36] S. Ioffe and C. Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167.

[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Dropout: A simple way to prevent neural networks from overfitting*, *J. Mach. Learn. Res.* **15** (2014) 1929.

[38] D.P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980.

[39] CMS collaboration, *A deep neural network to search for new long-lived particles decaying to jets*, *Mach. Learn. Sci. Tech.* **1** (2020) 035012 [arXiv:1912.12238].

[40] J. Kieseler, M. Stoye, M. Verzetti, E. Bols, A. Stakia, H. Kirschenmann et al., *Deepjet*, *Zenodo* (2020).

[41] J. Kieseler, M. Stoye, M. Verzetti, P. Silva, S.S. Mehta, A. Stakia et al., *Deepjetcore*, *Zenodo* (2020).

[42] J. Kieseler, E. Bols, M. Verzetti, C. Vernieri, D. Majumder, L. Gouskos et al., *Deepntuples*, *Zenodo* (2020).

2020 JINST 15 P12012