Proceedings of CTD 2020
PROC-CTD2020-10
September 10, 2020

Deep Sets for Flavor Tagging at the ATLAS Experiment

Nicole Hartman, Michael Kagan, Rafael Teixeira de Lima,
On behalf of the ATLAS Collaboration

SLAC National Accelerator Laboratory

ABSTRACT

Flavor tagging is a major client for tracking in particle physics experiments at high energy colliders, such as the ATLAS detector at the LHC, where it is used to identify the experimental signatures of heavy flavor production. Among other features, charm and beauty hadron decays produce jets containing several tracks with large impact parameters. This work introduces a new architecture for flavor tagging, based on Deep Sets, which models the jet as a set of tracks, not relying on any particular ordering or sequence. This approach is an evolution with respect to the Recurrent Neural Network currently adopted by the ATLAS experiment, which treats track collections as a sequence. The Deep Sets algorithm uses track impact parameters and kinematics within a permutation invariant architecture, leading to a significant decrease in training and evaluation time compared to the sequential recurrent algorithms. The Deep Sets algorithm is compared with current ATLAS flavor tagging benchmarks and methods are provided to explore and interpret the information learned by the network in the training process.

PRESENTED AT

Connecting the Dots Workshop (CTD 2020)
April 20-30, 2020

1 Introduction

The process of identifying jets from b-quarks, called b-tagging, is crucial to the physics program of the ATLAS experiment [1] at the Large Hadron Collider. Since the b-hadron decays via the weak force, its characteristic "long" lifetime of ≈1.5 ps leads to an experimental signature of a displaced decay in the detector. ATLAS organizes the task of b-jet classification into two types of algorithms: *impact parameter (IP)-based* which takes as input the collection of tracks and looks for those that are displaced from the interaction point, and *vertex-based* which explicitly reconstructs either a single secondary vertex or the multiple displaced vertices of the cascade decay chain [2]. The final ATLAS b-tagger is a multi-layer perceptron trained on the outputs of these IP-based and vertex-based taggers [2].

ATLAS is a multipurpose detector comprising an inner tracking system surrounded by electromagnetic and hadronic calorimeters which are encased by muon spectrometers [1].* Tracks are reconstructed from hits in the inner tracker, and the identified primary vertex (PV) is the reconstructed vertex with the largest $\sum_i p_{\mathrm{T},i}^2$ of the associated tracks. Jets are formed by clustering particle flow objects [3] with the anti-k_{T} algorithm [4], and tracks are associated to jets using a p_{T} dependent ΔR association [2]. Algorithm training and evaluation is performed with simulated semi-leptonic $t\bar{t}$ events, produced in $\sqrt{s} = 13$ TeV proton-proton collisions. Events are generated using the PowhegBox [5–8] v2 generator interfaced to Pythia v8.230 [9] to model the parton shower, hadronization, and underlying event. The decays of b- and c-hadrons are performed by EvtGen v1.6.0 [10]. Particles are passed through the ATLAS detector simulation [11] based on GEANT4 [12]. More details can be found in [13].

This work presents the use of a permutation invariant Deep Sets architecture [14] and on the application of the Deep Sets formalism within particle physics known as Energy / Particle Flow Networks [17] to replace the currently deployed b-tagging Recurrent Neural Network (RNN). The training and inference speed-up of Deep Sets is characterized and further optimization improvements are shown. A linearized view of the network response is used to understand what the network has learned about the properties of b-jets, and we demonstrate the feasibility of including such a tagger in the current calibration workflow.

2 IP-based b-tagging Algorithms

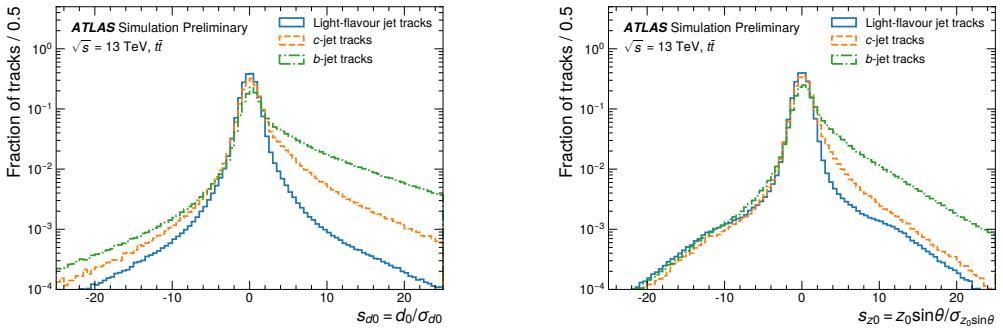
2.1 Previous Work

For the lifetime-based taggers, a positive sign is assigned to the IP of tracks more consistent with being from a long-lived decay [2]. The distributions for the significances (or IP divided by the error) are shown in Figure 1 for tracks from b, c, and light quark or gluon (l) jets [13], with the jet labelling definition from [2].

Figure 1 shows that the jets with true displaced heavy flavor (HF) decays are more likely to have positive IPs, and this asymmetry is the key discriminating feature used by the IP3D algorithm which uses these histograms to define probability density functions (pdfs) for the track significances conditioned on the jet flavor. To aggregate information across the tracks in the jet, the track IPs are assumed to be independent to define a higher-level discriminant by multiplying the ratio of probabilities across the tracks in the jet: $D_{\mathrm{IP3D},\{l,c\}} = \log \prod_{i \in \mathrm{tracks}} p_b^{(i)} / p_{\{l,c\}}^{(i)}$. These pdfs for the tracks are defined for 14 exclusive categories depending on the hit patterns of the tracks to give a measure of the track quality in these discriminants [2].

Since this uncorrelated track assumption limits the expressivity of the IP3D algorithm, ATLAS also employs another lifetime-based tagger, RNNIP [15]. The recurrent architecture can operate on an arbitrary number of tracks in the jet since tracks are sequentially processed as the RNN continuously updates a fixed dimensional hidden state which is used to determine the probabilities for a jet to be classified as b, c, or l. For the RNNIP results included here, the tracks are ordered by decreasing s_{d0}. As neural networks (NNs) avoid the curse-of-dimensionality (sparsification of the entries as the number of dimensions grows) inherent

*ATLAS uses a right-handed coordinate system with its origin at the nominal interaction point in the centre of the detector and the z-axis along the beam pipe. The x-axis points from the interaction point to the centre of the LHC ring, and the y-axis points upwards. Cylindrical coordinates (r, ϕ) are used in the transverse plane, ϕ being the azimuthal angle around the z-axis. The pseudorapidity is defined in terms of the polar angle θ as $\eta = -\ln \tan(\theta/2)$. Angular distance is measured in units of $\Delta R \equiv \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$.



(a) (b)

Figure 1: Lifetime signed track significances in the (a) transverse and (b) longitudinal directions [13].

in histogram-based approaches, the low-level hit information is used instead of the IP3D categories, and variables sensitive to the kinematics of the decay are included. The full set and description of the currently adopted RNNIP inputs are shown in Table 1 [13].

Input	Description
s_{d0}	d_0/σ_{d0}: transverse IP significance
s_{z0}	$z_0 \sin\theta/\sigma_{z0\sin\theta}$: longitudinal IP significance
$\log p_{\mathrm{T}}^{\mathrm{frac}}$	$\log p_{\mathrm{T}}^{\mathrm{track}}/p_{\mathrm{T}}^{\mathrm{jet}}$: logarithm of fraction of the jet p_{T} carried by the track
$\log \Delta R$	Logarithm of opening angle between the track and the jet axis
IBL hits	Number of hits in the IBL: could be { 0, 1, or 2 }
PIX1 hits	Number of hits in the next-to-innermost pixel layer: could be { 0, 1, or 2 }
shared IBL hits	Number of shared hits in the IBL
split IBL hits	Number of split hits in the IBL
nPixHits	Combined number of hits in the pixel layers
shared pixel hits	Number of shared hits in the pixel layers
split pixel hits	Number of split hits in the pixel layers
nSCTHits	Combined number of hits in the SCT layers
shared SCT hits	Number of shared hits in the SCT layers

Table 1: Track feature inputs for RNNIP and DIPS (Deep Impact Parameter Sets) algorithms [13]. IBL (Insertable B-Layer) is the innermost pixel layer, and SCT (SemiConductor Tracker) are the strips layers.

The inclusion of RNNIP in the recommended ATLAS tagger has led to an increase in performance from the incorporation of the track correlations and the additional input features [16], but it's not clear whether using a sequence to model the jet is a necessary assumption for the performance gains. The ordering of the tracks for RNNIP was empirically determined rather than specified by the problem, and processing tracks sequentially is inherently a slow process.

2.2 Deep Sets Algorithm

For the reasons stated above, we were interested in looking at permutation invariant models for this task using a Deep Sets model. Adopting the notation of [17], the model can be expressed by

$$\mathcal{O}\left(\{p_1, \ldots p_n\}\right) = F\left(\sum_{i=1}^{n} \Phi(p_i)\right) \tag{1}$$

where for each track $p_i \in \mathbb{R}^m$ corresponding to the $m = 13$ features listed in Table 1, Φ is a NN that extracts per track features, the sum operation over the n tracks encodes the permutation invariance, and F is another NN which operates on the (permutation invariant) vector to account for the correlations between the tracks. The architecture used for these studies is illustrated in Figure 2, with an additional batch normalization layer [18] before applying the Rectified Linear Unit (ReLU) nonlinearity [19]. This architecture is based off the one in [17], although we additionally did experiments varying the number of layers and hidden units for the F and Φ networks. Most of the settings we considered yielded the same performance, and we show results with this architecture to keep the number of trainable parameters for the RNNIP and DIPS algorithms roughly the same. The implementation of the Deep Sets algorithm for b-tagging will be referred to as DIPS (Deep Impact Parameter Sets) in following sections.

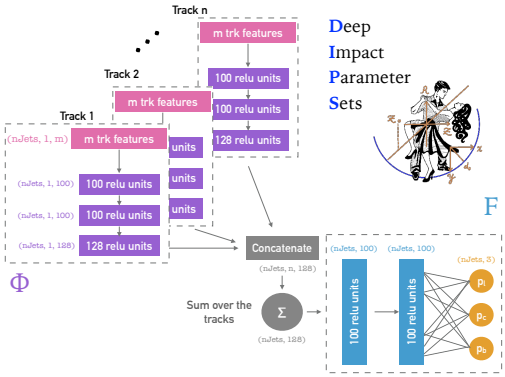


Figure 2: Schematic for the implementation of the Deep Sets architecture [13], where the Φ network extracts 128 features for each of the tracks.

3 Results

The results here show retrainings for both the RNNIP and DIPS architectures with $t\bar{t}$ simulation as described in [13]. Of the 3 million jets in the training set, 20% are held out for a validation set for early stopping, and another independent test sample of 3 million jets is used for evaluating the models. For RNNIP, tracks are sorted by decreasing s_{d0}, and the baseline kinematic track selection considers the tracks with $p_{\mathrm{T}} > 1$ GeV, $|d_0| < 1$ mm, and $|z_0 \sin\theta| < 1.5$ mm[†], a tight selection adopted by both the IP3D and RNNIP algorithms. When quantifying the rejections and times, each model is trained five times and we report the averages across the trained models with errors given by the standard deviations.

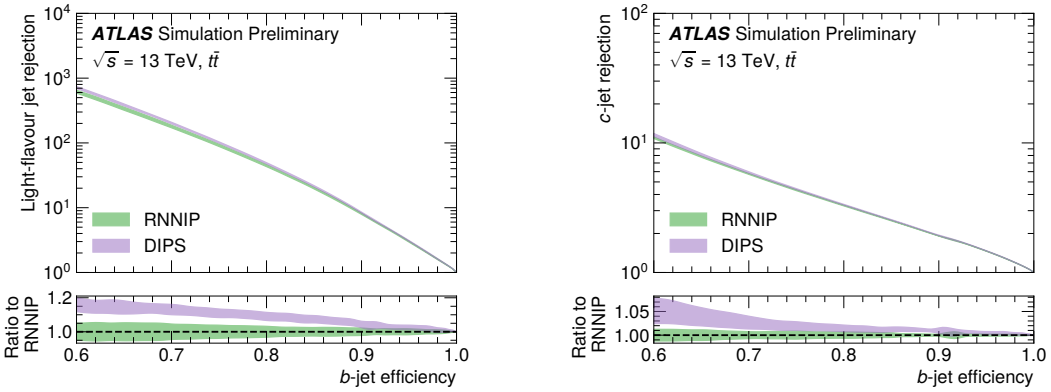
3.1 Timing Studies

Since Deep Sets is not a sequential algorithm, the track feature extraction (with the Φ networks) can be parallelized, resulting in a speed-up of 3-4 as demonstrated by Table 2, with the DIPS hyperparameters chosen to roughly match the number of trainable parameters of RNNIP. Information from the NN's multi-class probabilities p_b, p_c, and p_l are aggregated into a single discriminant $D_b = \frac{p_b}{f_c p_c + (1-f_c) p_l}$, where f_c is a parameter chosen after the training is completed. For these studies we use $f_c = 0.07$, the c-quark fraction in $t\bar{t}$ events. The performance of these models is quantified through Receiver Operator Characteristic (ROC) curves, which are defined as the background rejection (1 / mistagging efficiency) versus b-jet identification efficiency, as shown in Figure 3. The speed improvement does not come at a loss in efficiency, and there is even a slight improvement in the model performance up to 20% for l-rejection and 5% for c-rejection.

[†]IPs are measured with respect to the reconstructed PV.

Model	# of parameters	GPU training time / epoch [s]	GPU eval time [s]	CPU eval time [s]
RNNIP	47k	241 ± 14	170 ± 2	685 ± 84
DIPS	49k	78 ± 4	46 ± 2	206 ± 98

Table 2: Timing metrics for trainings on Nvidia 2080 Ti GPUs, and GPU evaluations on an Nvidia Titan X GPU. The nominal value denotes the mean of five independent trainings (for sample-size of 3 million jets), while the error bar is the standard deviation [13].



(a) (b)

Figure 3: Performance for RNNIP and DIPS models with the same inputs for (a) l-rejection versus b-efficiency and (b) c-rejection versus b-efficiency [13].

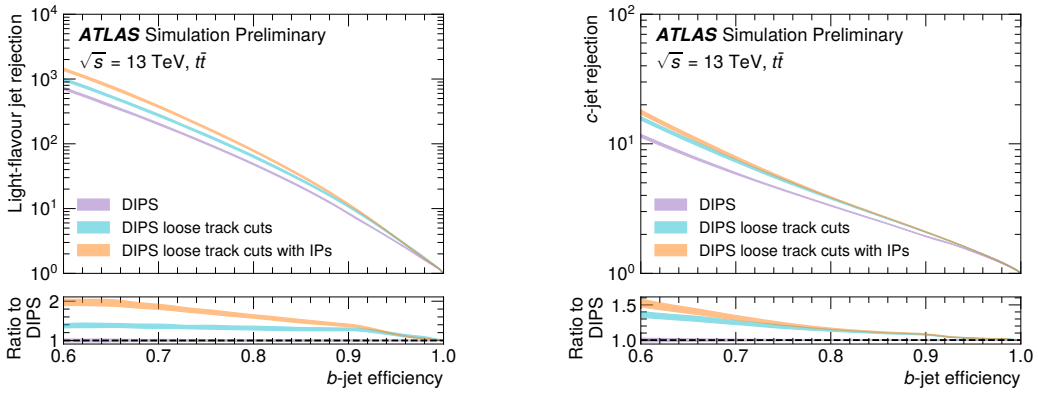
3.2 Performance Optimization

Figure 4 shows the results of two additional optimization studies [13]. Loosening the track selection to $p_{\mathrm{T}} > 500$ MeV, $|d_0| < 3.5$ mm, $|z_0 \sin\theta| < 5$ mm increases both the l- and c-rejection performance by up to 40%. Adding in the IPs (d_0 and $z_0 \sin\theta$) for each track further improves the performance of the l-rejection by up to 200% with respect to the baseline DIPS. While the impact parameter significance carries combined information about both decay length and track quality, the impact parameters focus only on information about the track displacement regardless of the track reconstruction quality. Thus the two features carry information which is not totally correlated, thus giving the model additional handles to learn how track properties are distributed in the different jet flavors.

Physics analyses require a tagger that performs well across a broad kinematic range. As such, Figure 5 shows the l- and c-rejections with a 77% b-jet efficiency working point (WP) which is flat as a function of jet p_{T} and η. Figure 5 compares the RNNIP and DIPS models that use the baseline track selection to the Optimized DIPS model which includes the loosened track selection and IP inputs. The overall performance gains of Optimized DIPS persist across the kinematic regimes of interest.

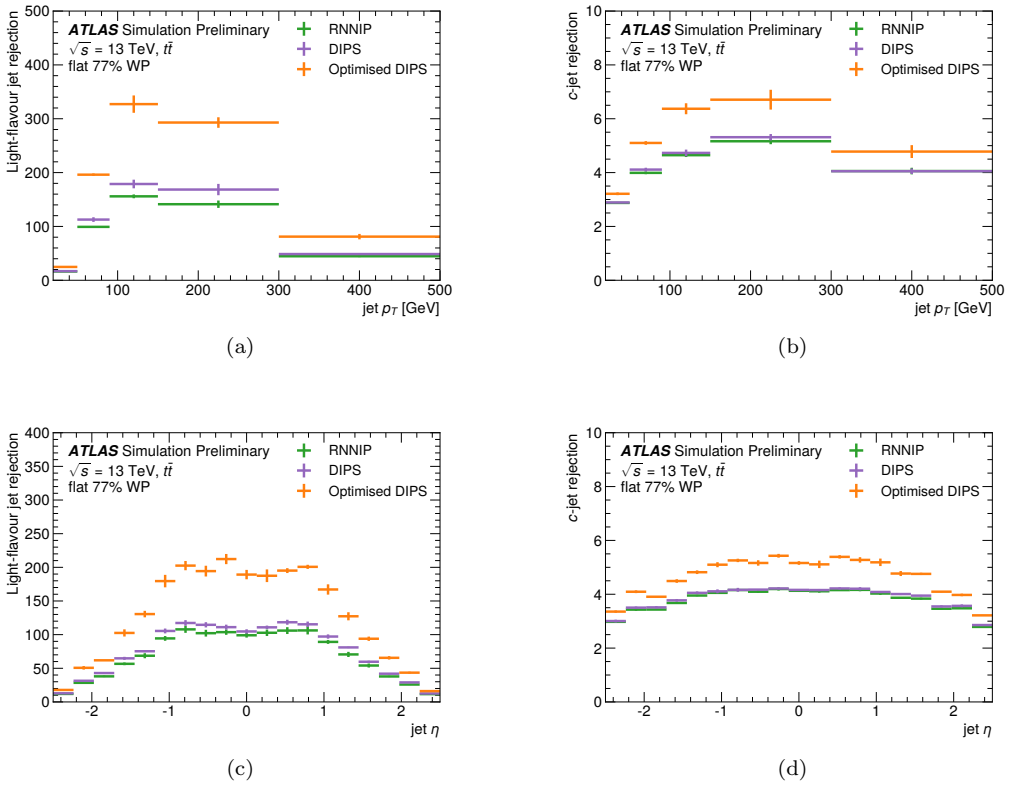
4 Architecture Understanding

For the following studies we show results for DIPS with the baseline set of inputs and nominal track selection.



(a) (b)

Figure 4: Light jet rejection (a) and c-jet rejection (b) versus b-jet efficiency for different track selections and input features [13].



(a) (b)

(c) (d)

Figure 5: Evolution of the background rejections with several algorithms, for a 77% efficiency b-tagging WP that is flat as a function of jet p_{T} shown for (a) l-rejection and (b) c-rejection, and flat as a function of jet η shown for (c) l-rejection and (d) c-rejection [13].

4.1 Saliency Maps

To investigate what the architecture has learned, we employ a method called saliency maps [20], originally used to understand what parts of an image convolutional neural networks focus on for classification. A saliency map represents a linearized version of the network response by considering the gradient of the discriminant with respect to the input features. For a single jet, $\nabla_{inputs} D_b \in \mathbb{R}^{n \times m}$, where n is the number of tracks in the jet and $m = 13$ is the number of features associated with each track. Information is aggregated across a sample of jets to observe trends, and thus a sample of b-jets that *failed* to be identified by the 77% efficiency working point was considered, as mistagged jets are in the regime where the gradients are still informative.[‡] Additionally, to ameliorate the dependence on track multiplicity, jets were required to have exactly $n = 8$ tracks, and in computing the average, the tracks defining the image were ordered by decreasing s_{d0}, as shown in Figure 6. The gradients in this image show how to modify this sample of jets to make them more b-like, illustrating what the model has learned about the characteristics of a b-jet. The significances are shown in the bottom rows, and demonstrate that the network wants at least five high IP tracks in the jet, consistent with the multiplicity expected from b-hadron decays. The leading tracks in s_{d0} are encouraged to be harder, consistent with expectations for the b-quark fragmentation function, and the leading track in s_{d0} is encouraged to have a larger opening angle due to the geometric constraints a highly displaced track has with respect to the jet axis. Finally, in the upper left hand portion of the image, the negative gradients for the tracks' shared and split hits indicate that the high s_{d0} tracks in the jet need to be of good quality when relying on them for a b-tagging decision. These observations give a qualitative assessment that DIPS has learned sensible properties of b-jets.

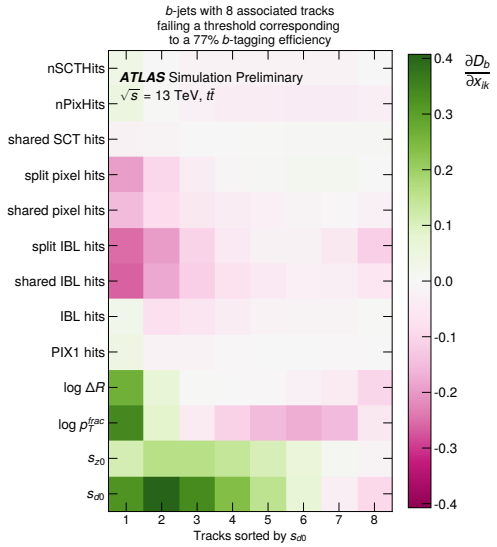


Figure 6: Gradient of the discriminant D_b with respect to the jet inputs for b-jets with 8 tracks failing the 77% efficiency WP [13].

4.2 Flipped Taggers

In training the tagger and quantifying its performance, a simulation with known jet flavor labels is used, but when deploying a tagger in an analysis, *scale factors* (SFs) for each jet flavor are derived to correct for the efficiency modifications induced by the domain shift between simulation and data. As demonstrated in Figure 3, the $\mathcal{O}(1000)$ l-rejections mean that even if a calibration starts with a l-jet dominated sample, applying the tagger will reduce the l-jets to a level where the l-jet mistagging efficiency can no longer be directly extracted. For deriving the l-jet SFs, we calibrate a "flipped" definition of the tagger which

[‡]Considering instead b-jets above a specified threshold starts saturating some of the gradients.

has a similar discriminant distribution for l-jets but a much reduced performance for b-jets to avoid being overwhelmed by HF jets after the tagger application [21]. We do not plan to calibrate DIPS directly, but the b-tagger that gets trained on both DIPS and the other IP-based and vertex-based tagger outputs. But ensuring that DIPS has a flipped tagger definition with the desired properties can significantly aid the ability of the final b-tagger to flip successfully.

The key feature used for the DIPSFlip tagger is the symmetry of the track significances in l-jets, as shown in Figure 1 [13]. DIPSFlip takes the DIPS weights trained with our nominal sample, but at the evaluation time, modifies the inputs by multiplying the track significances by -1. Thus, DIPSFlip tags become largely driven by tracks with large negative IP significances rather than large positive IP significances which dominate nominal DIPS tags. Figure 7 shows 1 - tagging probability for DIPS and DIPSFlip (in the solid and dashed lines, respectively) as the threshold on the b-tagging discriminant is varied. For the l-jets, the flipped and nominal tagger definitions are nearly overlapping, while the desired drop in efficiency for the heavy flavor jets is observed when using the flipped tagger definition, which is promising for using DIPS in defining a high-level flipped tagger.

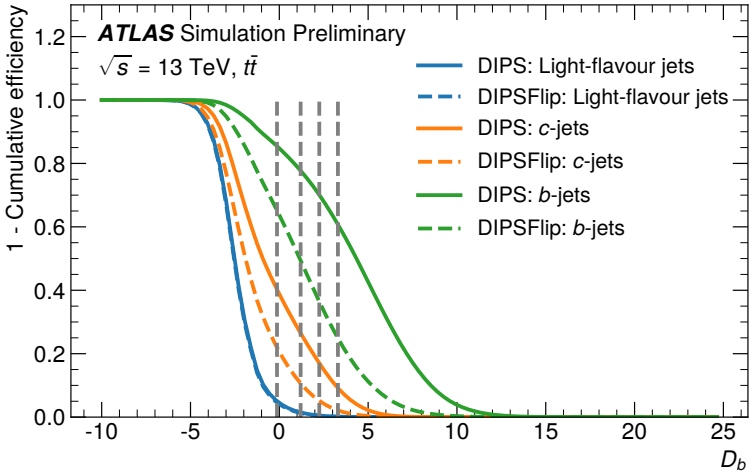


Figure 7: Performance of the DIPS and DIPSFlip taggers. The dashed grey lines show the WPs that ATLAS uses for calibration [13].

5 Conclusions

In this work, we present a new lifetime-based b-tagger, DIPS, based on an architecture which is permutation invariant to the track reorderings. We show a 3-4 speed-up in training and inference with respect to RNNIP, and demonstrate input modifications which continue to boost the performance. These low-latency algorithms are promising not only to help reduce the computational overhead of the ATLAS reconstruction, but also for exploring the potential of implementing b-tagging in hardware triggers.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1656518. MK is supported by the US Department of Energy (DOE) under grant DE-AC02-76SF00515 and by the SLAC Panofsky Fellowship. RTdL is supported by the SLAC Panofsky Fellowship.

References

[1] ATLAS Collaboration, "The ATLAS Experiment at the CERN Large Hadron Collider", JINST **3**, S08003 (2008).

[2] ATLAS Collaboration, "ATLAS b-jet identification performance and efficiency measurement with $t\bar{t}$ events in pp collisions at $\sqrt{s} = 13$ TeV", Eur. Phys. J. C **79**, 970 (2019). [arXiv:1907.05120 [hep-ex]].

[3] ATLAS Collaboration, "Jet reconstruction and performance using particle flow with the ATLAS Detector", Eur. Phys. J. C **77**, 466 (2017). [arXiv:1703.10485 [hep-ex]].

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

[5] S. Frixione, P. Nason, and G. Ridolfi, "A positive-weight next-to-leading-order Monte Carlo for heavy flavour hadroproduction", JHEP **09**, 126 (2007). [arXiv:0707.3088 [hep-ph]].

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

[7] S. Frixione, P. Nason, and C. Oleari, "Matching NLO QCD computations with parton shower simulations: the POWHEG method", JHEP **11**, 070 (2007). [arXiv:0709.2092 [hep-ph]].

[8] 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**, 043 (2010). [arXiv:1002.2581 [hep-ph]]).

[9] T. Sjöstrand et al., "An introduction to PYTHIA 8.2", Comput. Phys. Commun. **191**, 159 (2015). [arXiv:1410.3012 [hep-ph]].

[10] D. J. Lange, "The EvtGen particle decay simulation package", Nucl. Instrum. Meth. A **462**, 152 (2001).

[11] ATLAS Collaboration, "The ATLAS Simulation Infrastructure", Eur. Phys. J. C **70**, 823 (2010). [arXiv:1005.4568 [physics.ins-det]].

[12] S. Agostinelli et al., "GEANT4 – a simulation toolkit", Nucl. Instrum. Meth. A **506**, 250 (2003).

[13] ATLAS Collaboration, "Deep Sets based Neural Networks for Impact Parameter Flavour Tagging in ATLAS", ATL-PHYS-PUB-2020-014 (2020). https://cds.cern.ch/record/2718948.

[14] M. Zaheer et al., "Deep Sets", CoRR (2017). [arXiv:1703.06114 [cs.LG]].

[15] ATLAS Collaboration, "Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment", ATL-PHYS-PUB-2017-003 (2017). https://cds.cern.ch/record/2255226.

[16] ATLAS Collaboration, "Expected performance of the 2019 ATLAS b-taggers", (2019). http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/FTAG-2019-005.

[17] P. T. Komiske, E. M. Metodiev, and J. Thaler, "Energy flow networks: Deep Sets for particle jets", JHEP **01**, 121 (2019). [arXiv:1810.05165 [hep-ph]]. https://link.springer.com/article/10.1007/JHEP01(2019)121

[18] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", CoRR (2015). [arXiv:1502.03167 [cs.LG]].

[19] V. Nair and G. Hinton " Rectified Linear Units Improve Restricted Boltzmann Machines", International Conference on Machine Learning, 2010.

[20] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", (2013). [arXiv:1312.6034 [cs.CV]].

[21] ATLAS Collaboration "Calibration of light-flavour b-jet mistagging rates using ATLAS proton–proton collision data at $\sqrt{s} = 13$ TeV", ATLAS-CONF-2018-006 (2018). https://cds.cern. ch/record/2314418.