ATLAS NOTE

ATL-PHYS-PUB-2017-003

March 7, 2017

Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment

The ATLAS Collaboration

Abstract

A b-jet identification algorithm is constructed with a Recurrent Neural Network (RNN) at the ATLAS experiment at the CERN Large Hadron Collider. The RNN based b-tagging algorithm processes charged particle tracks associated to jets without reliance on secondary vertex finding, and can augment secondary-vertex based taggers. In contrast to traditional impact-parameter-based b-tagging algorithms which assume that tracks associated to jets are independent from each other, the RNN based b-tagging algorithm can exploit the spatial and kinematic correlations between tracks which are initiated from the same b-hadron. This new approach also accommodates an extended set of input variables. This note presents the expected performance of the RNN based b-tagging algorithm in simulated $t\bar{t}$ events created in proton–proton collisions at $\sqrt{s}$ = 13 TeV.

ATL-PHYS-PUB-2017-003
10 March 2017

1 Introduction

The identification of jets containing a b-hadron, typically referred to as b-tagging, plays a vital role for the ATLAS experiment [1] at the CERN Large Hadron Collider (LHC). It is important for both precise Standard Model measurements, including the Higgs sector, and for exploring new physics scenarios in the $\sqrt{s}$ = 13 TeV proton-proton collisions now being delivered in Run 2 of the LHC. The identification of b-quark jets in ATLAS is based on several low-level b-tagging algorithms which can be grouped into impact-parameter based (IP) and secondary-vertex based algorithms [2]. Since the two classes of algorithms offer complementary information, the outputs from these algorithms are combined in a boosted decision tree which is the default high-level algorithm used by ATLAS physics analyses [3].

IP algorithms have the benefit that discrimination is possible even if no secondary vertex is explicitly reconstructed. They typically assign per-track probabilities that the track originated from a jet of a given flavor and combine these probabilities as a likelihood product. The probabilities are estimated by referencing binned likelihood distributions from simulation which ignore any interdependencies between track parameters of different tracks in a given jet. This simplification is driven by practical limitations in the likelihood algorithm: accounting for the impact parameters and track quality for every track in a jet would quickly drive the likelihood algorithm to a space of unmanagably large dimensionality. To overcome these challenges, a new algorithm is introduced based on track properties and a recurrent neural network (RNN).

RNNs are a subclass of neural network architectures that allow extensions to sequence-based and temporal domains (see Ref. [4] and References therein). They are typically used in natural language processing [5, 6], machine translation [7, 8], and time-series analysis [9, 10]. In the case of b-tagging, the tracks in a b-jet can be treated as a variable-length sequence to recast b-tagging into a domain where RNNs have proven to be useful. This note introduces one such algorithm, and shows how it can improve upon and extend current impact-parameter based taggers.

This note is organized as follows: The simulated event samples and the object and event selection are described in Section 2. The RNN based b-tagging algorithm is described in Section 3. A comparison between key b-tagging quantities in simulated events is presented in Section 4. Conclusions are presented in Section 5.

2 Event Samples, Object Selection, Association and Flavor Labeling

The following performance plots are produced from simulated $t\bar{t}$ production corresponding to $\sqrt{s}$ =13 TeV proton-proton collisions. Events are generated with the next-to-leading order generator Powheg-Box v2 [11] and the CT10 [12] PDF set, interfaced with Pythia 6.428 [13] with the CTEQ6L1 PDF set [14] and the Perugia 2012 tune for the parton shower [15]. EvtGen [16] is used to model the decays of the b and c-hadrons. Minimum bias interactions are generated with Pythia8 [17] and are overlaid on the $t\bar{t}$ events. Particles are passed through the ATLAS detector simulation [18] which is based on GEANT4 [19].

Events are selected by requiring a reconstructed primary vertex. If the event has several candidate vertices, the primary vertex is defined as the vertex with the largest sum of squared transverse momenta of the associated tracks. Jets are reconstructed by clustering energy deposits in the calorimeter with the anti-k_t algorithm [20, 21] and a radius parameter of 0.4, where clusters are calibrated at the electromagnetic energy scale. The resulting jet is calibrated at the hadronic scale through a transverse momentum, p_{T},

and pseudorapidity[1],η, dependent correction factor. A preliminary version of the Run-2 jet energy scale calibration is applied to the jets [22]. In this note, only jets with p_{T} above 20 GeV and $|\eta| < 2.5$ are considered. A jet vertex tagger (JVT) is used to reject jets originating from pileup interactions [23].

The tracks used in the b-tagging algorithms are associated to jets using the angular separation ΔR between the track and the jet axis. The ΔR requirement varies as a function of jet p_{T}, being wide for low p_{T} jets and narrower for high p_{T} jets which tend to be more collimated. For instance, at 20 GeV, it is $\Delta R < 0.45$ while for more energetic jets with a p_{T} of 150 GeV the threshold is $\Delta R < 0.26$ [2]. A similar geometric matching scheme is used to label jets as b-, c-, light-jets, or as hadronically decaying τ-leptons (τ-jets) in simulation [24]. Under this matching scheme the sample is composed of approximately 26% b-jets. The remaining jets are 89% light-flavor jets, 4% τ-jets and 7% c-jets.

ATLAS's baseline high-level b-tagging algorithm is MV2c10 [3], which employs a boosted decision tree based on jet kinematics and properties computed from an impact parameter tagging algorithm (IP3D), a secondary vertex fitting algorithm (SV1), and a multivertex decay chain finding algorithm (JetFitter). MV2c10 is trained with b-jets as signal, and a mix of 7% c and 93% light jets as background. More detailed descriptions of these algorithms can be found in references [2, 3, 24].

For a jet to be considered b-tagged, the output of the high-level b-tagging algorithm is required to be above a fixed threshold value. Several such thresholds, or "working points" (WP), are defined, in such a way as to correspond to a well-defined average efficiency when applied to b-jets from a sample of inclusive $t\bar{t}$ events. This note references two types of working points: cuts that are tuned to an average 70% b-tagging efficiency across the p_{T} spectrum, and "flat efficiency" working points in which the threshold for the tagger discriminant is varied with jet p_{T} in order to achieve a uniform efficiency as a function of p_{T}.

3 Recurrent Neural Network Based b-Tagging

3.1 Motivation for RNN tagging

Within the decay of a b-hadron, several charged particles can emerge from the secondary (or tertiary) decay vertex with large impact parameters, as measured by the distance of closest approach to the primary vertex. These impact parameters are intrinsically correlated: if one track is found with a large impact parameter then finding a second track with large impact parameter is more likely. If no displaced decay is present, as in light-flavor jets, then such a correlation should not exist. The 2D distribution of transverse impact parameter significance[2] (S_{d_0}) for the leading and subleading transverse impact parameter significance tracks can be found in Figure 1 for b-jets, where a correlation can clearly be seen, and light flavour-jets, where no such correlation is observed.

ATLAS's baseline IP based b-tagging algorithm, IP3D, uses 3D likelihood templates in S_{d_0}, S_{z_0}, and a track categorization to compute three per-flavor conditional likelihoods, p_b, p_c, and p_{light}. These likelihood templates are derived from histograms with 35 bins in S_{d_0}, 20 bins in S_{z_0}, and 14 bins in track category,

[1] 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 upward. 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 $\eta = -\ln\tan(\theta/2)$, and the ΔR between two objects is defined in terms of angular separation in η and ϕ, $\Delta R = ((\Delta\eta)^2 + (\Delta\phi)^2)^{1/2}$

[2] In all algorithms discussed here the impact parameter is characterized by the lifetime signed transverse impact parameter significance ($S_{d_0} \equiv d_0/\sigma_{d_0}$) and the lifetime signed longitudinal impact parameter significance ($S_{z_0} \equiv z_0 \sin\theta/\sigma_{z_0 \sin\theta}$).

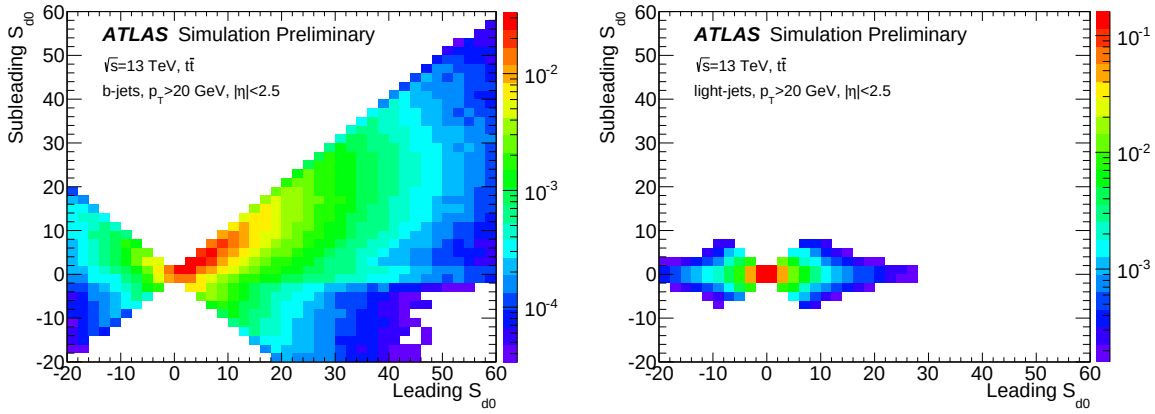

Figure 1: The distribution of the d_0 significance for the leading d_0 significance track and subleading d_0 significance track in b-jets (left) and light jets (right). The plots were produced with 700k b-jets and 1M light jets, and each distribution is normalized to unity.

where each category corresponds to a different track quality [24]. Multiplying by the three flavors, this results in a final bin count of $35 \times 20 \times 14 \times 3 = 29,400$. As the probability is computed per track, the likelihood of a jet being of a given flavour is computed as the product of the per-track likelihoods. The IP3D discriminant is built from the conditional log-likelihood ratio, IP3D $= \ln \prod_{i \in \text{tracks}} p_b^i / p_{\text{light}}^i$.

One of the main assumptions of the IP3D algorithm is that the per-track flavor conditional likelihood can be computed independent of the other tracks in the jet. Such a likelihood model does not account for the effect shown in Figure 1, and the method of building templates to define likelihoods requires large sample sizes. In addition, extending the template to account for additional kinematic variables is computationally expensive, since the number of template bins (and the number of simulated events required to fill them) grows exponentially with the number of variables. Such algorithmic deficiencies can be rectified using machine learning classifiers.

3.2 Recurrent Networks

Recurrent neural networks are used to directly learn sequential dependencies for arbitrary-length sequences [4, 25]. The fundamental unit of an RNN is a cell encapsulating an internal state vector. As the first step of processing any given sequence (in this case the tracks in a jet), the internal state is initialized to zero. At each step in the sequence, the cell is handed a fixed number of inputs (in this case the parameters that describe one track). These parameters are combined with the *current* internal state in order to compute a *new* internal state based on a set of rules which are tuned in the training phase. At the end of the sequence the cell's internal state serves as a fixed-dimensional representation of the entire sequence. In this way a recurrent cell is able to reduce a sequence of arbitrary length to a fixed number of variables, which can then be processed by a traditional feed-forward network.[3]

Much of the recent success of RNNs in various natural language and long-sequence processing applications can be attributed to the advent of Long Short-Term Memory (LSTM) [27] units and later variants such as Gated Recurrent Units (GRUs) [28, 29]. These architectural modifications at the cell level mitigate issues

[3] For a review of terminology such as "fully-connected", and related concepts, and a more pedagogical introduction to deep learning, see for instance References [25, 26].

related to vanishing and exploding gradients [30–32], and improve the knowledge persistence of long-term dependencies. These special kinds of recurrent units employ different internal gating mechanisms to modify the cell state in order to balance and regulate the relative importance of long-term and short-term information.

3.3 Implementation

In addition to the track selection described in Section 2, tracks fed to the RNN are required to pass additional quality requirements identical to those required by IP3D. Specifically, tracks must have $p_T > 1$ GeV, $|d_0| < 1$ mm and $|z_0 \sin\theta| < 1.5$ mm, seven or more silicon hits, at most two silicon holes, and at most one hole in the pixel detector. A hole is defined as a hit expected to be associated with the track but not present. Both IP3D and the RNN are able to accommodate an arbitrary length track sequence, and thus no limit is placed on the number of tracks fed to the algorithms.

For each selected track, the variables provided to the network can be found in Table 1, and the architecture is represented schematically in Figure 2. Initial tests have shown that a network using only S_{d_0}, S_{z_0}, and the track category outperformed IP3D, indicating that the RNN algorithm alone adds discrimination over a likelihood-based approach. Additional discrimination power is gained from variables which exploit the mass of heavy hadrons and p_{T} differences between tracks from fragmentation and those from heavy hadrons. As shown in Figure 4, the fraction of jet energy carried by each track, $p_{\mathrm{T}}^{\mathrm{frac}}$, and the angular separation between the track and jet axis, $\Delta R(\text{track, jet})$, improve the performance of the RNN tagger. The track category used by IP3D requires special attention: as the numerical values of the categories have no relative meaning, the category is embedded into a trainable, unit-normalized, 2D continuous representation.

After the initial selection and category embedding, tracks are ordered by $|S_{d_0}|$ and passed to a LSTM cell which transforms the arbitrary-length track sequence to a 50 dimensional vector. This vector is then fed into a feed-forward fully-connected layer with four outputs corresponding to the b-jet, c-jet, light-jet, and τ-jet probabilities (p_b, p_c, p_{light}, and p_τ). To ensure that these outputs sum to 1, they are then fed through a final softmax layer. The final network includes 11,636 trainable free parameters—a 60% reduction compared to IP3D.

The network was trained with 3.2 million jets[4] and tested with an independent sample of 4 million jets. When evaluating the RNN, all tracks satisfying the quality criteria listed above are used, although for the sake of training the sequence was truncated at 15 tracks. Training with longer sequences showed negligible differences, which is unsurprising given that the chosen track ordering puts tracks from b-hadrons early in the sequence, and that only 0.5% of jets include more than 15 tracks. The entire network was trained for 50 epochs using Keras [33] with the Theano [34] backend and the Adam optimizer [35]. All layers were initialized from a Glorot Uniform distribution [36]. Within the ATLAS event reconstruction software, the network is evaluated using lwtnn [37]. When training, the jet transverse momentum spectra of b-jets and c-jets were separately reweighed to the light-jet spectrum so as to prevent the neural network from learning to discriminate directly from sample and flavor specific momentum distributions.

In addition to the network described above, several related architectures were considered. Ordering tracks by $|S_{d_0}^2 + S_{z_0}^2|$ or p_{T} showed minor differences in performance but no substantial benefit. In another configuration, the 2D embedded track category was removed and replaced with the tracking variables

[4] Network trainings with fewer jets, such as 1 million, only showed minor degradation in performance.

Track Variable	Description
Used in IP3D and RNN tager	
S_{d_0}	Lifetime signed transverse impact parameter significance, d_0/σ_{d_0}, where d_0 is the transverse displacement at the point of closest approach to the primary vertex, σ_{d_0} is the error on d_0, and the sign is defined to be positive (negative) if the point of closest approach to the primary vertex is in front (behind) the primary vertex with respect to the jet direction.
S_{z_0}	Lifetime signed longitudinal impact parameter significance, z_0/σ_{z_0}, where z_0 is the longitudinal displacement at the point of closest approach to the primary vertex, σ_{z_0} is the error on z_0, and the sign is defined to be positive (negative) if the point of closest approach of the track to the primary vertex is in front (behind) the primary vertex with respect to the jet direction.
Category [24]	A categorization of the tracks depending on the number of observed, expected, or missing hits in the different layers of the silicon pixel and strip detectors. The category attempts to organize tracks based on impact parameter resolution.
New to the RNN tagger	
$p_{\mathrm{T}}^{\mathrm{frac}}$	The fraction of transverse momentum carried by the track relative to the jet, $p_T^{\mathrm{track}}/p_T^{\mathrm{jet}}$.
ΔR(track, jet)	The angular distance between the track and the jet axis, $\sqrt{(\phi_{\mathrm{track}} - \phi_{\mathrm{jet}})^2 + (\eta_{\mathrm{track}} - \eta_{\mathrm{jet}})^2}$.

Table 1: Descriptions of track variables used in IP3D and the RNN tagger.

which define track categories. While this variant yielded similar performance, the category embedding architecture was chosen because the modeling of the IP3D categorization scheme has been more carefully scrutinized. Many other versions of the RNN presented above were examined, including different sets of track variables, alternative recurrent units, additional recurrent layers, additional fully-connected layers, and variations in the training parameters such as training epochs and learning rates. The above network was ultimately found to be an ideal compromise when accounting for classification accuracy, training time, and simplicity.

4 Performance Results

To demonstrate the performance of the RNN tagging algorithm, the b-tagging efficiency and background rejection (1 / background efficiency) are examined both inclusively in jet-p_{T} and as a function of the jet p_{T}. As the RNN tagger has a four-class output it provides a much more flexible class of discriminants than traditional single-class algorithms like MV2c10.[5] For the sake of visualization, however, these outputs are combined into the following discriminant function:

$$D_{\mathrm{RNN}} = \ln \frac{p_b}{f_c p_c + f_\tau p_\tau + (1 - f_c - f_\tau) p_{\mathrm{light}}} \tag{1}$$

where f_c and f_τ are parameters representing the c- and τ-jet fractions, respectively, which can be used change the relative importance of c-jet, τ-jet and light-jet rejection by the discriminant. The f_c parameter

[5] Note that a multi-class tagging discriminant can be useful to identify any of the classes, not just b-jets. For a non–b-tagging example see charm tagging [38].

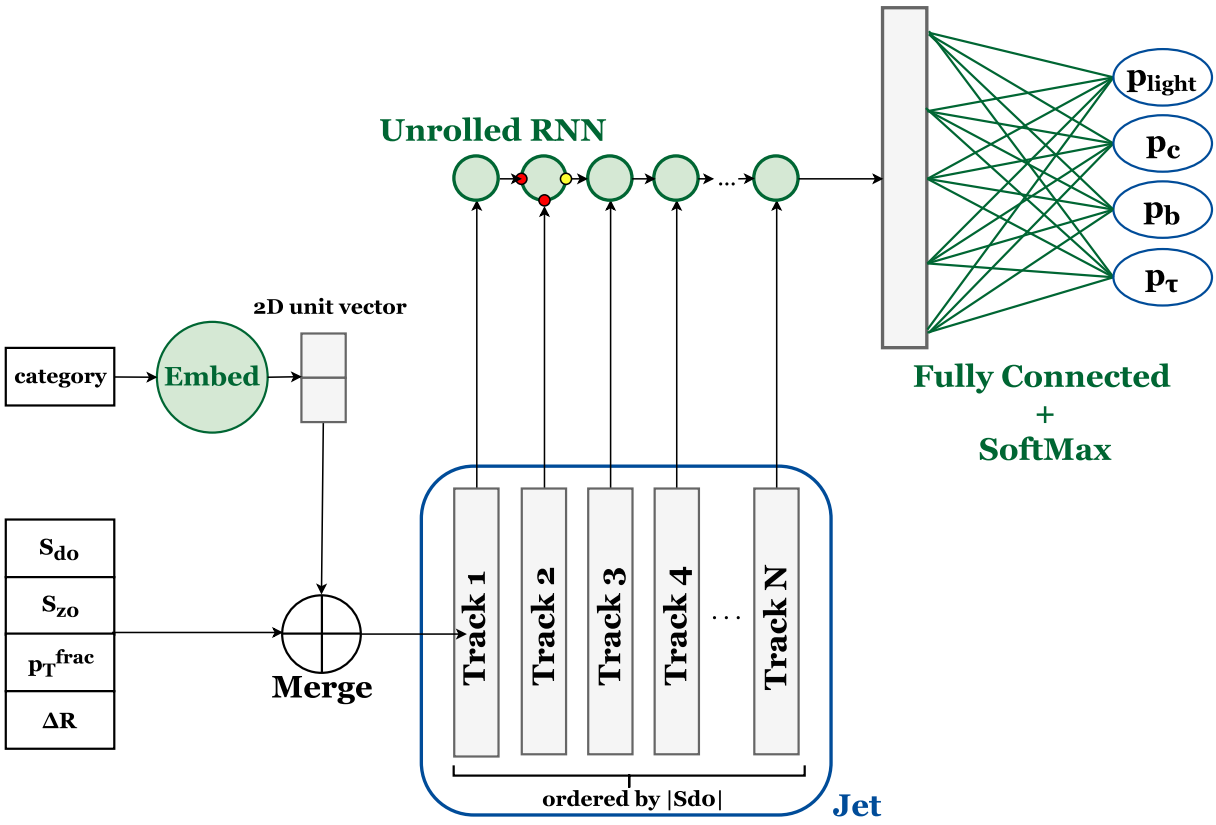

Figure 2: A schematic diagram of the RNN-based flavor-tagger, showing input features, network structure, and the 4-class output of $\{p_b, p_c, p_{\text{light}}, p_\tau\}$. In trainng track sequences are truncated after a maximum of 15 tracks, while in application all tracks are considered.

is fixed at $f_c = 0.07$, which is chosen based on the $t\bar{t}$ training sample[6]. Given that τ discriminants typically combine calorimeter and vertex information [39, 40], any meaningful comparisons with existing τ reconstruction algorithms are beyond the scope of this note. For consistency with existing b-tagging algorithms[7] the f_τ parameter is therefore set to $f_\tau = 0$, effectively ignoring p_τ.

Background rejection versus signal efficiency curves are produced by scanning a minimum threshold on D_{RNN} and computing background rejection and signal efficiency at each threshold. These curves can be found in Figure 3, for a background of light jets and a background of c-jets separately. The RNN outperforms IP3D, which is promising given the similar input variables, and given that neither of these algorithms relies on reconstructing a secondary vertex. For a b-tagging efficiency of 70% the RNN has 2.5 times the light-jet rejection and 1.2 times the c-jet rejection of IP3D. To illustrate the complementarity between IP-based and vertex-based algorithms, the secondary vertex reconstruction algorithm SV1 and the high-level algorithm MV2c10 are also shown. The limitations of secondary vertex reconstruction are clearly illustrated by the maximum efficiency of SV1: in roughly 20% of b-jets no secondary vertex can be

[6] Small changes to this fraction were observed to have little effect on the discriminant performance
[7] Jets labeled as τ jets are removed from the SV1, IP3D, and MV2c10 training.

reconstructed. Although not pictured, JetFitter suffers from a similar maximum efficiency. Despite their limited efficiency, however, the vertex-based algorithms clearly complement the IP-based algorithms as illustrated by the superior performance of MV2c10, which combines JetFitter, SV1, and IP3D in a BDT.

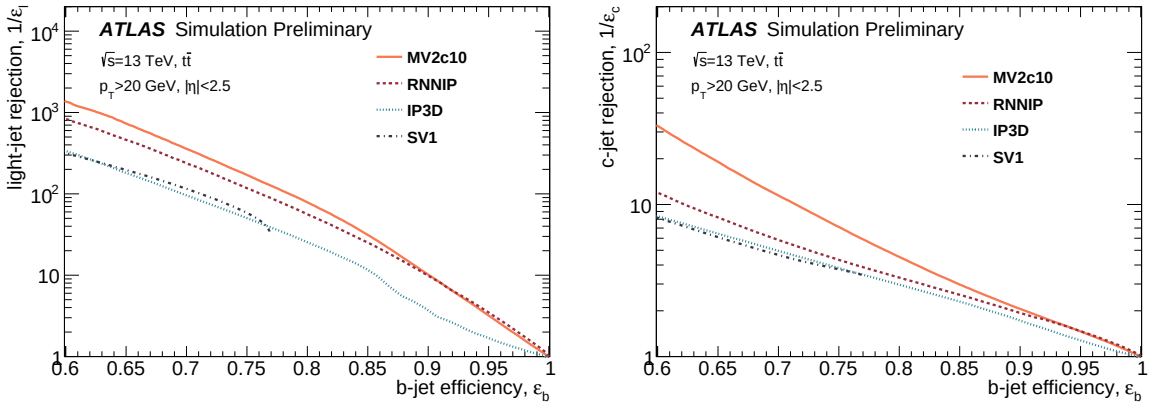

Figure 3: The light-jet (left) and c-jet (right) rejection versus b-tagging efficiency for jets with $p_{\mathrm{T}} > 20$ GeV and $|\eta| < 2.5$. The statistical error on the curve is less than 3%. MV2c10 is a high level BDT tagger which integrates IP3D outputs outputs with additional vertex information from JetFitter and SV1.

To factorize the gains from the recurrent network from those provided by the additional variables, Figure 4 compares the performance of an RNN trained on *only* the IP3D inputs to one which uses the additional $\Delta R(\text{track, jet})$ and $p_{\mathrm{T}}^{\mathrm{frac}}$ inputs. A network using exactly the same inputs as IP3D improves light-jet rejection by a factor of 1.7 and c-jet rejection by a factor 1.05, even in the absence of any additional variables.

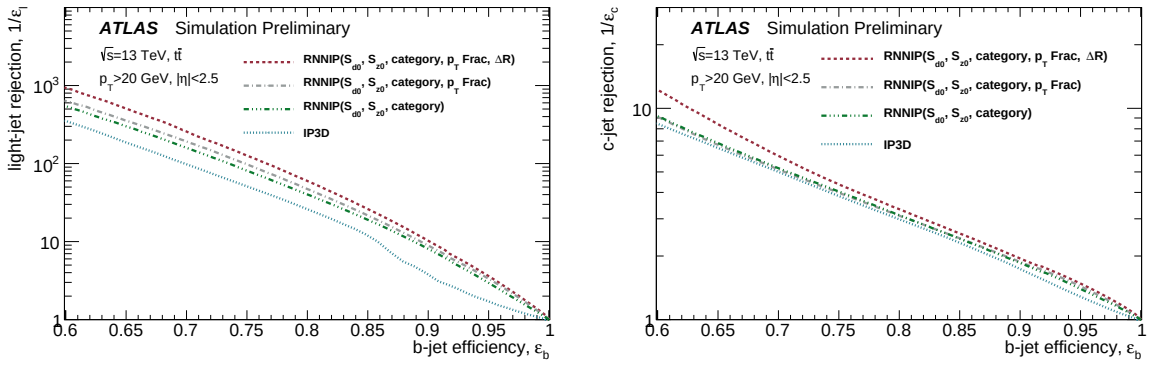

Figure 4: The light-jet (left) and c-jet (right) rejection versus b-tagging efficiency for jets with $p_{\mathrm{T}} > 20$ GeV and $|\eta| < 2.5$, for RNNs trained using various sets of input variables, and for IP3D. The RNN without $p_{\mathrm{T}}^{\mathrm{frac}}$ and $\Delta R(\text{track, jet})$ uses only the inputs available to IP3D.

In order to understand how the tagging performance depends on jet kinematics, the b-tagging efficiency versus jet p_{T} is shown in Figure 5. To isolate the effect of a changing b-tagging efficiency from that of the changing light-jet and c-jet rejection rejection, a flat-efficiency 70% WP is examined. In this case, all taggers have a 70% efficiency across p_{T}, and only the rejection is varying. The light- and c-jet rejection

as a function of jet p_{T} can be seen in Figure 6. The RNN outperforms IP3D in terms of light- and c-jet rejection across the p_{T} range.

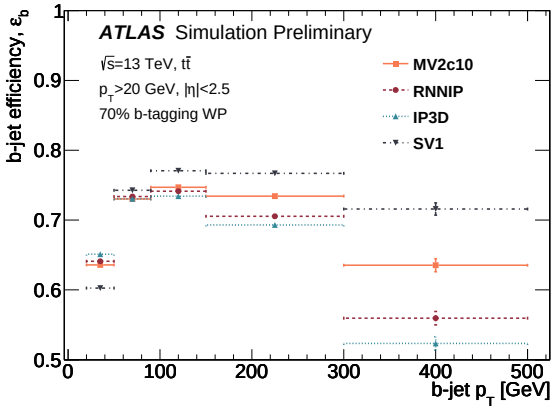

Figure 5: b-tagging efficiency for a 70% WP cut versus jet p_T.

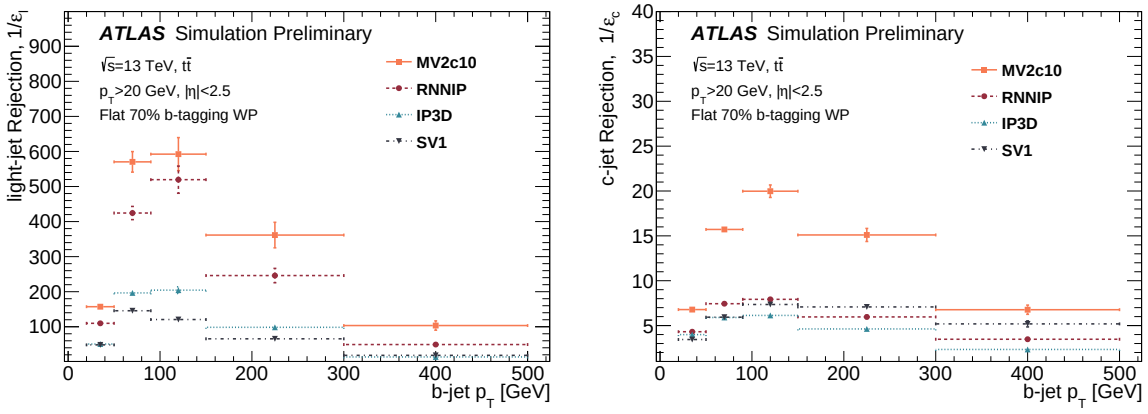

Figure 6: Light jet rejection (left) and c-jet rejection (right) at flat b-tagging efficiency of 70% versus jet p_{T}. Like MV2c10 and SV1, the RNN tagger performs best on jets with a p_{T} of roughly 50–150 GeV. MV2c10 is a high level BDT tagger which integrates IP3D outputs outputs with additional vertex information from JetFitter and SV1.

The power of the RNN approach, which can be inferred from the ROC curves in Figures 3 and 4, is due to the network's ability to build non-linear hierarchical representations in a high-dimensional space. To help illustrate the network's behavior, Figure 7 shows the correlation coefficient, ρ, between the input variables and D_{RNN} for each track in the sequence. The strongest correlation is for S_{d_0}, especially for the first ~ 8 tracks in the sequence. This may be related to the typical charged particle multiplicity expected in a b-jet. S_{z_0} is the second most correlated variable. There is a small negative correlation with the $p_{\mathrm{T}}^{\mathrm{frac}}$, especially for tracks later in the sequence, which may be related to the harder fragmentation of b-quarks compared to lighter quarks, leading to an expectation of only a small number of high $p_{\mathrm{T}}^{\mathrm{frac}}$ tracks. Finally there is a small negative correlation with ΔR, indicating that tracks far from the jet axis are less likely inside of b-jets where most tracks from a high momentum b-hadron are collimated.

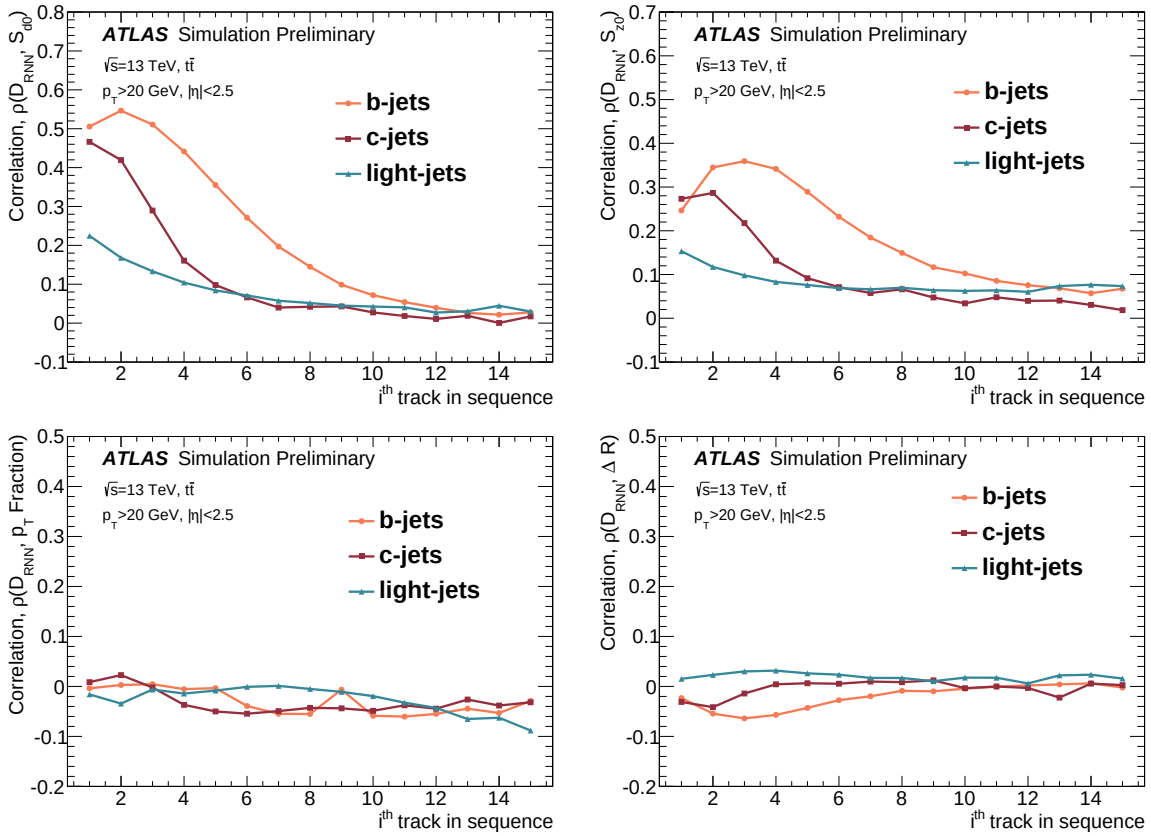

Figure 7: Correlations of per track input variables S_{d_0} (top left), S_{z_0} (top right), p_T^{frac} (bottom left), and ΔR, with the RNN score D_{RNN}.

5 Conclusions

A new low-level *b*-tagging algorithm has been presented which is built from a Recurrent Neural Network with a sequence of track-by-track variables as input. This algorithm is seen to outperform impact parameter taggers, as is expected due to the ability to learn and discriminate on the correlations between tracks in a given jet and the ability to extend the number of input variables well beyond what is feasible with likelihood based impact parameter taggers. Given this flexibility, including more relatively low-level tracking variables as RNN inputs offers a potential avenue for further improvements. While it is difficult to pinpoint what discriminating information the RNN has learned, this is partially illuminated by examining the correlation between the RNN output and the various track inputs to the network.

High-level ATLAS taggers such at MV2 integrate the outputs from an IP-based algorithm with two vertex-based algorithms, each of which relies on the full set of track parameters and covariance matrices. As a potential replacement for the IP-based component, the RNN discriminant should not be compared directly to MV2. Instead the usefulness of the RNN tagger relies on the additional information that it contributes to the high-level algorithm. This additional information may result in a performance boost to the high-level tagger, and indeed quantifying this improvement remains an active area of study within the ATLAS experiment. These studies include both quantifying the performance gains from adding the RNN outputs into a multi-algorithm composition of taggers like MV2 and studies of the correlations of

the various low-level and high-level taggers to more fully elucidate how b-tagging sensitive information is used by different algorithms.

References

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

[2] ATLAS Collaboration, *Performance of b-jet identification in the ATLAS experiment*, JINST **11**.04 (2016) P04008.

[3] ATLAS Collaboration, *Optimisation of the ATLAS b-tagging performance for the 2016 LHC Run*, ATL-PHYS-PUB-2016-012 (2016), URL: http://cds.cern.ch/record/2160731.

[4] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, Studies in Computational Intelligence 385, Springer, 2012.

[5] T. Mikolov et al., "Recurrent neural network based language model," *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, 2010 1045, URL: http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.

[6] A. Graves, A. Mohamed, and G. E. Hinton, *Speech Recognition with Deep Recurrent Neural Networks* (2013), arXiv: 1303.5778.

[7] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation* (2014), arXiv: 1406.1078.

[8] M. Luong, H. Pham, and C. D. Manning, *Effective Approaches to Attention-based Neural Machine Translation* (2015), arXiv: 1508.04025.

[9] Z. Che et al., *Recurrent Neural Networks for Multivariate Time Series with Missing Values* (2016), arXiv: 1606.01865.

[10] J. T. Connor, R. D. Martin, and L. E. Atlas, *Recurrent Neural Networks and Robust Time Series Prediction*, Trans. Neur. Netw. **5**.2 (Mar. 1994) 240, ISSN: 1045-9227.

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

[12] H.-L. Lai et al., *New parton distributions for collider physics*, Phys. Rev. D **82** (2010) 074024, arXiv: 1007.2241.

[13] T. Sjostrand, S. Mrenna, and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, JHEP **05** (2006) 026, arXiv: 0603175 [hep-ph].

[14] J. Pumplin et al., *New generation of parton distributions with uncertainties from global QCD analysis*, JHEP **0207** (2002) 012, arXiv: 0201195 [hep-ph].

[15] P. Z. Skands, *Tuning Monte Carlo Generators: The Perugia Tunes*, Phys.Rev. **D82** (2010) 074018, arXiv: 1005.3457 [hep-ph].

[16] D. Lange, *The EvtGen particle decay simulation package*, Nucl.Instrum.Meth. **A462** (2001) 152, URL: http://www.sciencedirect.com/science/article/pii/S0168900201004533.

[17] T. Sjostrand, S. Mrenna, and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, Comput. Phys. Commun. **178** (2008) 852, arXiv: 0710.3820.

[18] ATLAS Collaboration, *The ATLAS Simulation Infrastructure*, Eur. Phys. J **C70** (2010) 823, arXiv: 1005.4568 [physics.ins-det].

[19] S. Agostinelli et al., *GEANT4: A simulation toolkit*, Nucl. Instrum. Meth. **A506** (2003) 250, URL: http://geant4.web.cern.ch/geant4/.

[20] ATLAS Collaboration, *Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1* (2016), arXiv: 1603.02934 [hep-ex].

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

[22] ATLAS Collaboration, *Jet Calibration and Systematic Uncertainties for Jets Reconstructed in the ATLAS Detector at $\sqrt{s}$=13 TeV*, ATL-PHYS-PUB-2015-015 (2015), URL: http://cds.cern.ch/record/2037613.

[23] ATLAS Collaboration, *Tagging and suppression of pileup jets with the ATLAS detector*, ATLAS-CONF-2014-018 (2014), URL: https://cds.cern.ch/record/1700870.

[24] ATLAS Collaboration, *Expected performance of the ATLAS b-tagging algorithms in Run-2*, ATL-PHYS-PUB-2015-022 (2015), URL: https://cds.cern.ch/record/2037697.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, 2016.

[26] J. Schmidhuber, *Deep Learning in Neural Networks: An Overview*, Neural Networks **61** (2015) 85 , ISSN: 0893-6080, arXiv: 1404.7828.

[27] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Computation, 9 **8** (1997).

[28] J. Chung et al., *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling* (2014), arXiv: 1412.3555.

[29] K. Cho et al., *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation* (2014), arXiv: 1406.1078.

[30] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München," 1991, URL: http://people.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf.

[31] Y. Bengio, P. Simard, and P. Frasconi, *Learning Long-term Dependencies with Gradient Descent is Difficult*, Trans. Neur. Netw. **5**.2 (Mar. 1994) 157, ISSN: 1045-9227.

[32] R. Pascanu, T. Mikolov, and Y. Bengio, *Understanding the exploding gradient problem* (2012), arXiv: 1211.5063.

[33] F. Chollet, *Keras*, GitHub (2015), URL: https://github.com/fchollet/keras.

[34] Theano Development Team, *Theano: A Python framework for fast computation of mathematical expressions* (May 2016), arXiv: 1605.02688.

[35] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization* (2014), arXiv: 1412.6980.

[36] X. Glorot and Y. Bengio,
"Understanding the difficulty of training deep feedforward neural networks,"
Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010, 2010 249,
URL: http://www.jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf.

[37] D. H. Guest et al., "lwtnn/lwtnn: Release for Athena v21," 2017,
URL: https://doi.org/10.5281/zenodo.290682.

[38] ATLAS Collaboration,
Performance and Calibration of the JetFitterCharm Algorithm for c-Jet Identification,
ATL-PHYS-PUB-2015-001 (2015), URL: https://cds.cern.ch/record/1980463.

[39] ATLAS Collaboration, *Identification and energy calibration of hadronically decaying tau leptons with the ATLAS experiment in pp collisions at* $\sqrt{s}$*=8 TeV*, Eur. Phys. J. **C75**.7 (2015) 303,
arXiv: 1412.7086 [hep-ex].

[40] ATLAS Collaboration,
Reconstruction of hadronic decay products of tau leptons with the ATLAS experiment,
Eur. Phys. J. **C76**.5 (2016) 295, arXiv: 1512.05955 [hep-ex].