Long Short-Term Memory (LSTM) networks for boosted top tagging

Shannon Egan*, Wojtek Fedorko, Alison Lister

Department of Physics and Astronomy, University of British Columbia, Vancouver, Canada
shannon.egan@mail.mcgill.ca

Abstract

Neural networks provide an attractive alternative to traditional top tagging methods, as the learned tuning of network parameters may identify subtle distinguishing patterns that high-level quantum chromodynamics (QCD) motivated variables (such as N-subjettiness, τ_{32} or jet mass) cannot capture at high transverse momentum (p_T). Our previous work investigated the approach of feeding jet information to a Dense Deep Neural Network (DNN) as a flat, ordered list of jet constituent four-momenta. Given how naturally the constituents can be arranged as sequences of various orderings, for example by subjet or by p_T, this problem lends itself well to the use of Long Short-Term Memory (LSTM), a machine learning method useful for treating sequences and learning long range patterns in data. We thus explore neural networks which incorporate Long Short-Term Memory (LSTM) layers. Here we present results for boosted top tagging using various network architectures, ordering of jet constituents, and input properties such as trimming and pileup. Our best performing LSTM network achieves a background rejection of 101 for 50% signal efficiency. This represents more than a factor of two improvement over the Dense DNN, which yields a background rejection of 45 at the same signal efficiency.

I. Introduction

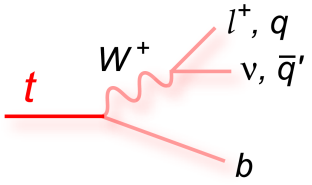

Figure 1: *Feynman diagram of a top quark decay.*

The top quark's exceptionally high mass and strong coupling to the Higgs mechanism have long made it an important subject for probes of the Standard Model (SM) [1]. In addition to this, several theory extensions to the SM predict new particles decaying to or being produced in association with "boosted" top quarks, which carry a transverse momentum (p_T) much greater than their rest mass. Such theories include the existence of additional gauge bosons decaying to top quarks (for example, $Z' \to t\bar{t}$) [2], vector-like quarks (VLQs) which couple strongly to the third-generation [3], and supersymmetric top squark partners [4]. ATLAS and CMS are already searching for such phenomena in a wide variety of channels [5–12], often using top tagging algorithms to distinguish between top jets and those which arise from light quarks and gluons - henceforth referred to as quantum chromodynamics (QCD) background or simply background.

The standard decay channels of the top quark are illustrated as a Feynman diagram in Figure 1. At low p_T this decay takes the shape of 3 distinct prongs, which serves as a useful signature for top tagging. At greater p_T, the jets become highly collimated and merge into a single "fat-jet" of high radius (usually 0.8). Traditional top taggers use quantum chromodynamics (QCD) based

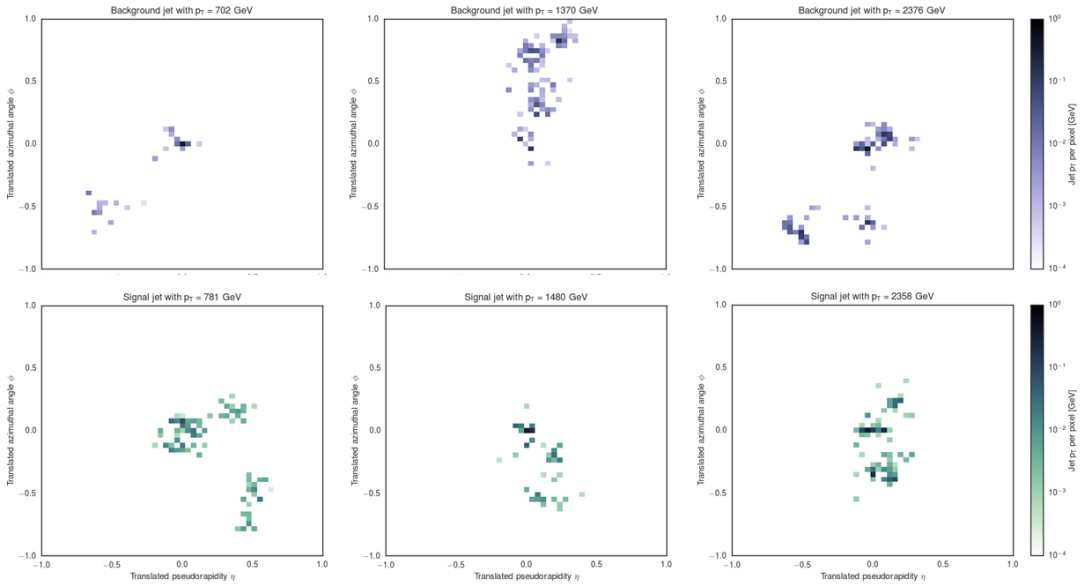

Figure 2: *Histograms of the fraction of jet pT carried by constituents in η - ϕ space for examples of signal and background jets. Jets were preprocessed as described in section 3.*

variables such as τ_{32} and N-subjettiness [13], as well as jet mass and jet reconstruction history to classify events. While well motivated, these methods achieve relatively low performance metrics on boosted top jets.

Tagger performance is usually evaluated based on the shape of response operating characteristic (ROC) curves, of which Figures 5, 8, 9 are examples. These plot a classifier's background (bkg) rejection versus signal (sig) efficiency, where:

$$\begin{aligned} Bkg\,rejection &= \frac{1}{false\,positive\,rate} \\ &= \frac{\#\,true\,bkg}{\#\,true\,bkg\,tagged\,sig} \\ Sig\,efficiency &= true\,positive\,rate \\ &= \frac{\#\,true\,sig\,tagged\,sig}{\#\,true\,sig} \end{aligned}$$

Higher background rejection at a given signal efficiency corresponds to better tagger performance in that regime, and thus the best performing network is characterized by an ROC curve with the maximum area beneath it.

Reviews of jet sub-structure based techniques in ATLAS [14] and CMS [15] find background rejection factors on the order of 15 and 5 in the 50% and 80% signal efficiency range, respectively, on jets with p_T near 1 TeV. As LHC experiments continue to collect and analyze more data from boosted top quarks, it is critical to improve tagger performance in this high p_T regime to increase chances of detecting rare new physics.

Use of neural networks (NNs) for signal/background discrimination has become increasingly prevalent in particle physics analyses [16–19], but is not yet commonly implemented for top tagging. A great deal of experimentation is underway, however, with physicists exploring many input

strategies and architectures. Currently, the most common approach is to translate information from hadronic calorimeter deposits into a 2D "jet image", which is then processed by a Convolutional Neural Network (CNN) [20] or Deep Neural Network (DNN) [21,22]. Figure 2 shows example jet images of both top and background jets that could be input to a CNN. Detector activation for these jets is noticeably sparse and it appears difficult to discern edges or distinguishing structural features of the jets. Thus, despite the power of CNNs as a tool for image recognition, their promise in this application may be limited by the loss of information that occurs during pixelization.

The search for alternative machine learning methods for boosted tagging is therefore in full force. Our recent paper [23] presents an input strategy in which a Dense Deep Neural Network (DNN) discriminator is fed 4-momentum vectors of the particles which comprise the jet (i.e. jet constituents). This approach has the advantages of requiring extremely minimal processing beyond reconstruction of calorimeter deposits and allowing great flexibility in how inputs are ordered, a feature which could be exploited to communicate salient information about jet substructure. Both Louppe et al. [24] and Butter et al. [25] have investigated a similar 4-momentum input strategy for boosted W-tagging and boosted top tagging, respectively. The former use a recursive jet embedding to reconstruct the QCD processes underlying the jet structure, and the latter use a DNN with custom layers based on Lorentz transformations and Minkowski metrics. Louppe et al. were also interested in the effect of constituent ordering on tagger performance. Surprisingly, their study found that a simple descending p_T sorting scheme yielded the best performance, even though initial studies showed that ordering based on jet structure was more collinear and infrared safe.

In the case of our DNN tagger, the 4-momenta are fed to the network in the form of a flat list where each jet constituent is described by its p_T, pseudorapidty (η) and azimuth (ϕ). While performance of this tagger was promising, the lack of demarcation between constituents is likely not ideal for learning jet structure and the DNN may not be as sensitive to information offered by the ordering of constituents, given the flatness of the input list and simplicity with which fully-connected layers treat data. These problems could be addressed by introducing a Recurrent Neural Network (RNN) method called Long Short-Term Memory (LSTM) [26]. LSTM neurons accept sequences of vectors as inputs and use a feature called the cell-state to retain information about previous entries. The neuron then uses information contained in the cell-state to exert influence on both the value of the outputs and what proportion of the incoming signal is retained. This enables us to arrange our four-momentum inputs in a more logical way: as a true sequence of jet constituents, where the list is reshaped as a tensor with a "time" dimension. Each jet constituent then constitutes a "timestep" with three features (p_T, η and ϕ). In the present study, we use this sequence representation of the jet constituents as the input for an LSTM based top tagger. Of particular interest is the effect of different sequence ordering methods on tagger performance.

Part II of this report explains the simulation and preprocessing of the data for network training and testing. Part III presents the network architecture and experiments in optimizing hyper-parameters. Finally, Part IV provides a detailed look at tagger performance under different sorting, pileup, and jet p_T conditions.

II. Dataset

i. Signal and background modelling

A more detailed account of simulation and data selection methods can be found in Reference [23]. The jet samples required for network training were generated using Monte Carlo (MC) simulations. Both signal from hadronic top quark decays and background from gluon and light quark jets were

generated at leading order using PYTHIA v8.219 [27].

The signal samples consist of Sequential Standard Model Z' boson [2] production at the LHC with pole mass ranging from 1400-6360 GeV. Cuts are applied on the Z' centre of mass energy and on top quark p_T to ensure that the pseudorapidity distribution of the top jets is approximately equivalent to that of the background jets. The final signal samples consist of $Z' \to t\bar{t}$ events with all-hadronic top decay. Background events are generated as QCD "dijet" processes, including gluon-gluon, quark-gluon and quark-quark scattering, with p_T of outgoing partons ranging from 470 to 2790 GeV. A large number of "soft" QCD interactions, referred to as minimum bias, are also generated.

Detector response is simulated using the DELPHES v3.4.0 suite [28] running with the default emulation of the CMS detector and particle flow event-reconstruction [29,30] - the latter known as energy flow in DELPHES. Minimum bias is added to signal and background events in order to simulate "pileup" [31], a property of LHC events which describes how many proton-proton (pp) collisions occur at approximately the same time. Given that each of these interactions produce hits in the detector, higher pileup correlates with more noise and higher rates of detector activation (both spatial and temporal). A random number of pp collisions are overlaid, with the exact number being drawn from a Poisson distribution. We investigated two different pileup scenarios. The first uses a Poisson distribution with a mean of 23, corresponding approximately to conditions of LHC data-taking during 2016; while the second has a distribution with mean 50, mimicking the conditions expected at the end of LHC Run 2. We will refer to the first scenario as LHC 2016 pileup, and the second as 50 pileup.

ii. Jet selection

DELPHES energy flow objects resulting from event reconstruction were clustered into high radius jets using the anti-k_T algorithm [32] implemented in FastJet [33]. The jet radius (R), a parameter of the clustering which determines the minimum distance between the centres of two jets, is set to $R = 1$. The distance R between two particles in $\eta - \phi$ space is defined as

$$R = \sqrt{\Delta y^2 + \Delta \eta^2} \tag{1}$$

Where y is the rapidity.

On some samples, a trimming procedure [34] is applied to reduce noise from soft radiation. This involves using the k_T algorithm [35] to re-cluster energy flow objects into "subjets" with $R = 0.2$. Any subjet which carries less than 5% of the initial jet's p_T is eliminated by removing its constituents from the list of jet particles.

Additional cuts were applied to constrain the set to jets with $600\,GeV \leq p_T \leq 2500\,GeV$ and $\eta \leq 2.0$. The remaining jets were sub-sampled to achieve a flat distribution in p_T and ensure that both signal and background have matching distributions in η. This is done in attempt to prevent the network from relying on the underlying distributions to distinguish signal from background, and to mitigate the degradation in performance at high p_T that is typical of jet taggers.

The final sample consists of approximately 7 million jets, split evenly between signal and background. The full sample is divided into 3 subsets which play different roles in network training. The first 80% of jets form the training set, which are passed to the network as labelled examples. Network parameters are updated as a function of the output's agreement with training set labels. 10% are assigned validation samples, which serve as a quick check to ensure that the network returns similar results on different subsets (and thus that the training set was not skewed in some way). The final 10% forms the test set, on which performance metrics such as loss, accuracy and background rejection are evaluated to track progress during training. The jets are

shuffled after each round of training (known as an epoch), such that each set contains a different subsample of jets. In addition to this sample, an orthogonal set of 11 million jets (again comprised of 50% signal, 50% background) is generated for the final evaluation of the trained network.

iii. Input shape and sequence ordering

The input list must be reshaped to a sequence for compatibility with the LSTM. The DNN inputs consisted of a flat list of the jet constituents, where every 3 entries characterize a single energy flow object. Given that Dense layers require a fixed number of inputs, jets with more than 120 constituents were truncated, while jets with fewer were zero padded so that the total length of the list was constant at 360. For input to an LSTM, the jet constituents are instead arranged as a sequence of 3-dimensional vectors (having values in p_T , η, and ϕ), with each sample consisting of 120 timesteps.

We hypothesize that the order of this sequence can provide salient information for signal/background discrimination to the LSTM tagger, and thus develop alternative sorting methods which attempt to represent the underlying QCD and substructure of the jets. In particular, we use a recursive algorithm which utilizes the history of the initial anti-k_T clustering to add constituents to the input list in an order which reflects the jet substructure. Clustering algorithms effectively produce a binary tree from the reconstructed particles, as depicted in Figure 3, where the intermediate jets are referred to as "PseudoJets" and are given by summing the 4-momenta of the particles or PseudoJets with the smallest distance metric [1] at a given iteration. Our jet structure sorting algorithm performs a depth-first traversal of this tree, adding to the input list when it reaches an original particle, such that the final ordering resembles the tree's branching pattern.

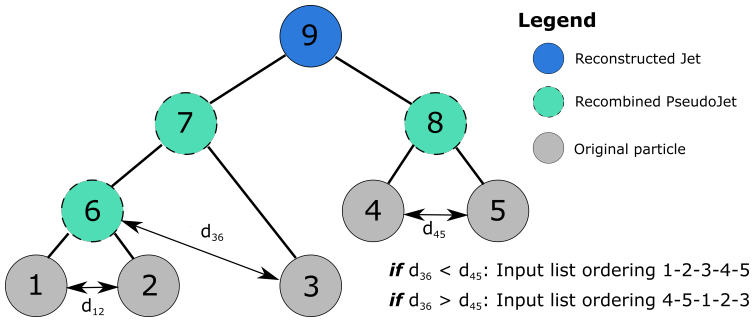

Figure 3: *Representation of the binary tree constructed by jet algorithms during clustering. In this study, an algorithm is developed to produce an list of original particles that provides information about the tree structure. The tree traversal is based on d_{ij} of earlier recombinations.*

This method is compared to sequence ordering schemes that were previously tested on the DNN; namely sorting purely by p_T of jet constituents, and sorting by subjet. The subjet sorting scheme first orders constituents by subjet p_T, and then by p_T of individual constituents. Subjet sorting was found to yield the best performance in the DNN tagger.

III. Network Architecture

All networks were implemented in the KERAS suite [36] using a Theano [37] backend. Network architecture was determined by a largely heuristic approach, using examples from literature to

[1] The distance metric used is referred to as d_{ij}, i and j being indices of particles or PseudoJets in the event list, and is defined as: $d_{ij} = min(k_{ti}^{2p}, k_{ti}^{2p}) \frac{\Delta_{ij}^2}{R^2}$, where k_{ti} is the transverse momentum of particle i, p define the precise algorithm used ($p = 1$ for k_T, $p = -1$ for anti-k_T or $p = 0$ for Cambridge-Aachen), R is a parameter of the clustering, and

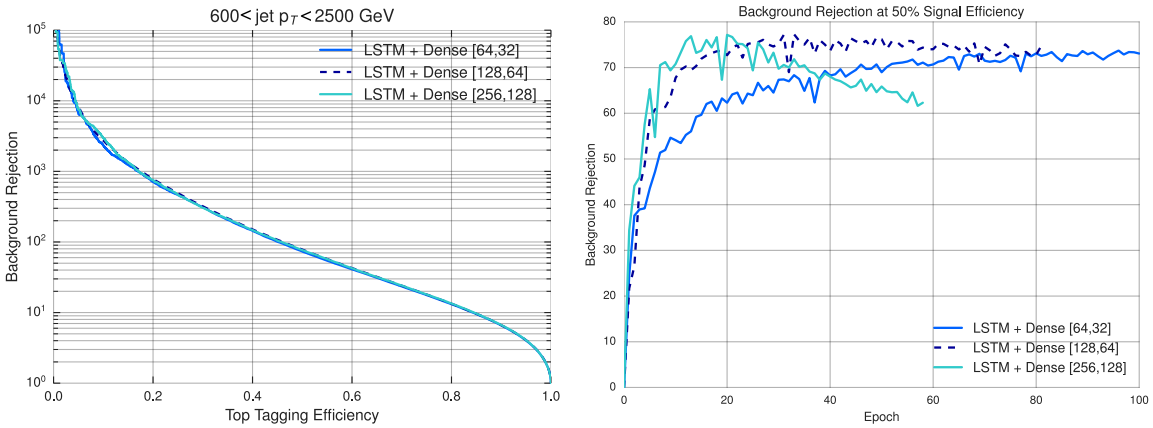

Figure 4: *Response operating characteristic (ROC) curve comparing LSTM + Dense networks of various widths. Differences in performance are not substantial, however the intermediate size gives the highest background rejection values overall.*

guide our experimentation. Architecture studies were performed on data pre-processed according to the best performing approach from our DNN study [23]. The samples mimicked LHC 2016 pileup, and were sorted first according to highest p_T subjet, followed by constituent p_T.

The simplest networks tested consist of a single LSTM layer followed by a fully-connected (or Dense) layer of half the width. These dimensions were set because wider fully-connected layers require greater computing time, and initial studies found that changing the width of this layer had little impact on performance relative to the effect of changing the width of the LSTM layer. Adding this Dense "projection" was inspired by CNNs, which use fully connected layers to extract information from the Convolution and Pooling stages. A procedure called masking was applied to the inputs to facilitate inputting sequences of variable length. All inputs are first fed through a masking layer, whose role is to remove zero-padded timesteps (i.e. jet constituent place-holders where $p_T = \eta = \phi = 0$) which could otherwise skew the LSTM parameter adjustments. Initial studies showed that performance without masking was relatively poor, with classification accuracy of maximum 73% and background rejection at 50% signal efficiency of 10.

Layer sizes were varied by powers of two to constrain the hyper-parameter search and to maximize computing efficiency on the GPU. The result of our LSTM layer size experiments is summarized in Figure 4. We found that a network with 128 LSTM nodes followed by a fully connected layer of 64 nodes converged to the best performance in terms of background rejection at 50% and area under the ROC curve. Networks with wider LSTM layers had a tendency to over-train very quickly and reach lower background rejection factors overall.

Network architectures incorporating a 2 LSTM layers were also tested, however this was found to yield minimal benefit to performance (See Figure A1), while substantially increasing computing time.

Experiments were also performed to determine the ideal optimizing function. We primarily tested variants of RMSprop [38], including Adam (RMSprop with momentum) [39] and Nadam (RMSprop with Nesterov-accelerated momentum) [40], due to its widespread use in sequence learning problems. While networks optimized with Nadam typically achieved higher background rejection values than with RMSprop or Adam, this optimizer also exhibited very unstable performance with respect to the test subsample and training course, especially for larger networks. As

$\Delta_{ij}^2 = (y_i - y_j)^2 + (\phi_i - \phi_j)^2$, y being the rapidity

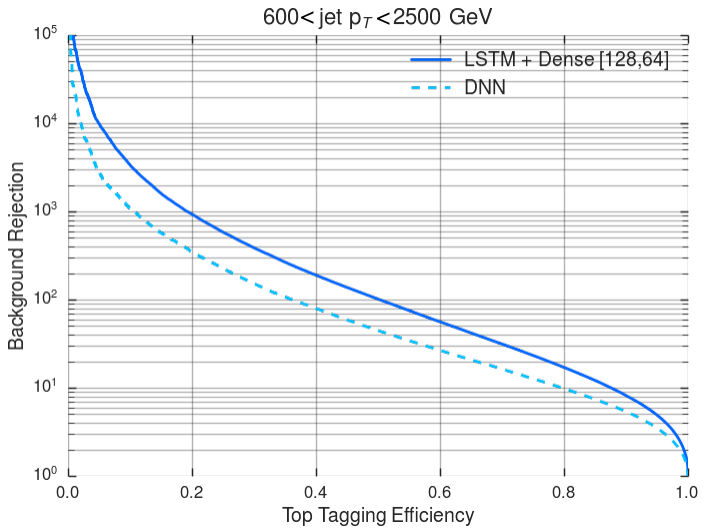

Figure 5: *Response operating characteristic (ROC) curve comparing the best performing DNN tagger to best performing LSTM tagger under LHC 2016 pileup conditions. Inputs to the DNN were trimmed and sorted by subjet, while LSTM inputs were untrimmed and sorted by the jet substructure-based method described in Section I.iii*

seen in Figure A2, the background rejection occasionally crashed to very low values, sometimes not recovering. This behaviour called into question the Nadam-optimized network's robustness to sample variation, and thus we felt it important to find an alternative which converged more uniformly. Using the Adam optimization method greatly stabilized performance during training with minimal loss in performance.

Our final best performing network consists of an LSTM layer with 128 nodes, which is fed variable length sequence inputs by the masking layer, followed by a fully-connected layer of 64 nodes before the binary output. The network is optimized using Adam applied to a binary cross-entropy cost function. This network is used in all further experimentation detailed in Section IV.

IV. Performance

The primary interest of this study was to evaluate how an LSTM network would compare to the previously developed DNN. Figure 5 shows ROC curves for the DNN and LSTM taggers under their respective best performing architectures and input conditions. The LSTM network yields better performance than the DNN across all signal efficiencies, in particular giving greater than a factor of two improvement in background rejection at the 50% signal efficiency point.

Comparing the histograms of each of tagger's outputs helps us to deduce how LSTMs improve the classification. Figure 6 shows prediction histograms for the DNN and LSTM tagger on identical inputs – trimmed jets mimicking LHC 2016 pileup – in order to isolate the effect of the change in network type. Based on this plot, the improvement in background rejection appears to result from enhanced signal classification. Prediction on signal jets is strongly shifted towards the two highest bins ($0.98 - 1.00$), thus reducing signal overlap with background jets given higher predictions.

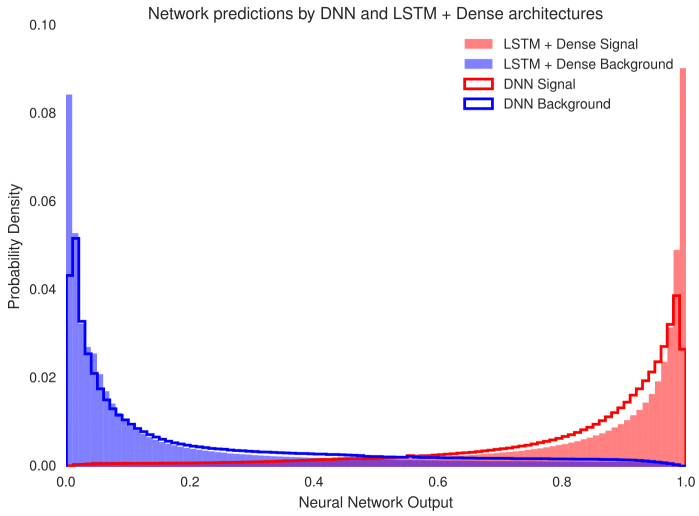

Figure 6: *Superposed network prediction histograms of DNN and LSTM taggers, separated by true signal and true background. The networks are compared on identical inputs - trimmed jets representing the LHC 2016 pileup case. The gain in discriminating power appears to result largely from an upward shift in output value for signal jets.*

i. Sequence ordering and trimming

The hypothesized advantage of LSTM was that memory properties conferred by the cell-state would enable our neural network to learn long-range patterns in data. We sought to take further advantage of this by ordering the jet constituent sequences such that variation in the "time" dimension could communicate useful information about a jet's substructure, and thus the underlying QCD. The algorithm used to accomplish this is described in Subsection I.iii.

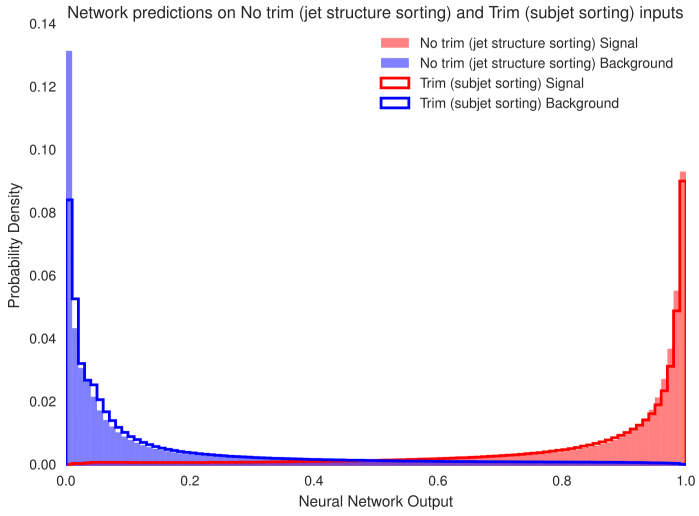

Figure 7: *Superposed prediction histograms of an LSTM network trained on trimmed, subjet-sorted inputs versus untrimmed, jet structure sorted inputs. Jets mimic LHC 2016 pileup in both cases.*

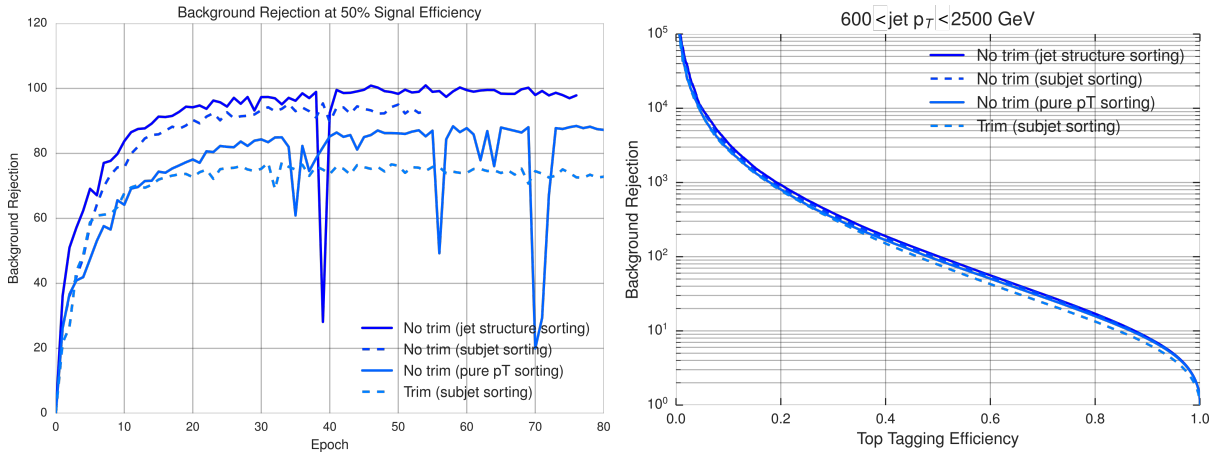

Figure 8: *Training progress (left) and response operating characteristic (ROC) curve (right) of sequence ordering and trimming conditions on the best performing LSTM architecture. Training on untrimmed inputs gives better performance regardless of sorting method, with a sequence ordering scheme based on jet substructure giving the best performance overall. All jets resemble LHC 2016 pileup.*

Different trimming cases were also tested. While inputting untrimmed jets slightly harmed performance in the DNN tagger, disabling trimming markedly increased background rejection given by the LSTM network, with the best background rejection at 50% rising from 78 to 98 on jets resembling LHC 2016 pileup and sorted by subjet. We hypothesize that leaving the jets untrimmed preserved randomness in background jets that was essential for identifying them as such. This is corroborated by the prediction histogram in Figure 7, which shows that LSTM networks trained on untrimmed inputs shift predictions on background jets towards the lowest bin, while producing negligible change in the signal distribution.

Sequence ordering methods were also shown to have an impact on performance, though not to as great an extent as trimming. Figure 8 shows performance of several sorting schemes on untrimmed jets, and compares them to the original subjet sorting on trimmed inputs as a benchmark. All three sequence ordering schemes - jet structure, subjet, and pure p_T - give higher background rejection values than are achieved with trimmed inputs. Furthermore, the sequence ordering methods that relate to jet substructure/QCD (namely, subjet and jet structure sorting) outperformed the pure p_T sorting, suggesting that the tagger can learn about structural features from the ordering of constituents, and that this information is useful for classification.

It is particularly encouraging that our sorting algorithm based on the jet clustering tree yields the best performance overall, and motivates us to seek even more useful representations of the jet structure through sequence ordering. Networks trained on jet structure ordered inputs surpassed 100 background rejection at 50% signal efficiency, which corresponds to only 1% of background jets being falsely identified as signal. Interestingly, the difference in performance between ordering methods is much less pronounced in the high signal efficiency regime. Thinking in terms of prediction histograms, this suggests that the differences arise from shifting the signal/background distributions in the high prediction range, rather than reducing overlap in the intermediate values.

ii. Pileup

Given the anticipated increases in LHC pileup in the coming years, it was important for us to study the effect of higher pileup on tagger performance. The soft interactions of pileup add

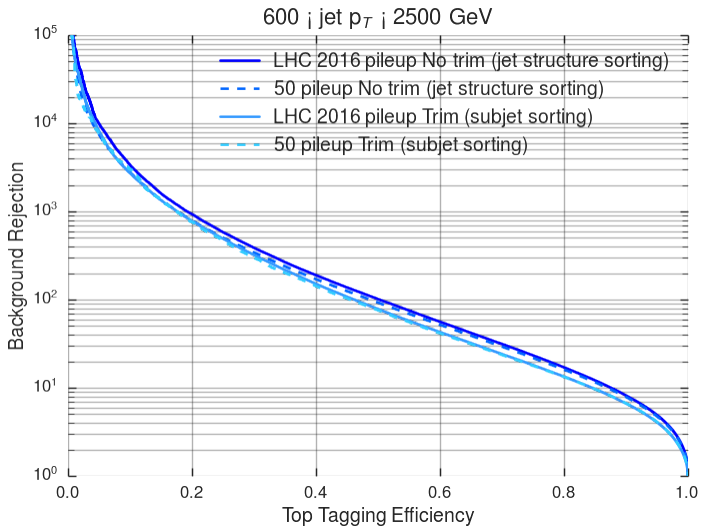

Figure 9: *Response operating characteristic (ROC) curve comparing varying trim and pileup conditions on an LSTM network. Performance is very resilient to higher pileup in the trimmed case, but less so when the jet inputs are untrimmed.*

noise that makes it more difficult to discern the structure of the hard scattering interaction. We looked in particular at the interaction between pileup and trimming conditions, being concerned that the observed performance benefits of disabling trimming could be lost at higher pileup. Figure 9 summarizes the result of this analysis, comparing trimming cases for a network trained on trimmed subjet-sorted inputs, and untrimmed jet structure-sorted inputs. Consistent with the DNN study, tagger performance is remarkably resilient to pileup when inputs are trimmed. Performance slightly degrades at 50 pileup in the untrimmed case, however at intermediate to high signal efficiency training on untrimmed jets still gives higher background rejection. Curiously, at low signal efficiency the performance approaches that of networks trained on trimmed jets.

Another interesting effect of pileup was in its interaction with sorting methods. As mentioned in the previous section, the jet structure sorting algorithm gave the best performance on inputs mimicking LHC 2016 pileup. The values given in Table 1 show that this advantage is lost in the 50 pileup case. This could indicate that the jet clustering is more random and less representative of the underlying QCD when a high degree of minimum bias is present, and thus the more general structure provided by subjet sorting is better for learning to classify jets.

V. Conclusion

The development of jet constituent-based top tagging leads naturally to experimentation with LSTMs, given the possibility to arrange such inputs as sequences of discrete timesteps. We show that using a simple and relatively narrow LSTM network with a fully-connected projection improves greatly on a DNN top tagger using the exact same jet constituent inputs in list form. Our best performing LSTM reaches background rejection at 50% of 101 on jets with $600\ GeV \leq p_T \leq 2500\ GeV$, more than two times greater than that achieved by our previously studied DNN.

Contrary to the DNN tagger, inputting untrimmed jets benefits LSTM performance by up to $\tilde{2}5\%$. Furthermore, using a sequence ordering method based on jet clustering algorithms allows

Input conditions			Background rejection at % signal efficiency		
Pileup	Trim	Sorting	80%	50%	20%
	DNN		9.8	45	365
LHC 2016	Yes	Subjet	13.4	78	779
	No	Jet structure	17.0	101	931
		Subjet	16.7	97.1	854.3
50	Yes	Subjet	13.5	78	779
	No	Jet structure	16.1	93	791
		Subjet	16.6	96	889

Table 1: *Background rejection factors of the best performing LSTM network architecture, as described in section III, given different input types. Sorting methods are described in Section I.iii. Background rejection values given by the best DNN tagger, trained and evaluated on trimmed jets with subjet sorting on LHC 2016 pileup, are also given for comparison.*

the network to reach slightly higher background rejection factors at LHC 2016 pileup, but the advantage of this jet structure sorting disappears at 50 pileup. As the LHC continues to move towards higher luminosity, it is important to thoroughly assess and try to mitigate such pileup effects.

Our ultimate goal is to implement improved top tagging in particle physics analyses, with the hope of increasing our sensitivity to the physics of highly boosted top quarks. This could open to door to new discoveries in BSM physics, as well as improve precision measurements of the Standard Model.

References

[1] P. Bartos, f. t. CDF, and D0 collaborations, "Measurements of top quark production and properties at the Tevatron," Sep. 2014.

[2] P. Langacker, "The Physics of Heavy $Z^{'}$ Gauge Bosons," *Rev. Mod. Phys.*, vol. 81, pp. 1199–1228, 2009.

[3] J. A. Aguilar-Saavedra, R. Benbrik, S. Heinemeyer, and M. PÃl'rez-Victoria, "Handbook of vectorlike quarks: Mixing and single production," *Phys. Rev.*, vol. D88, no. 9, p. 094010, 2013.

[4] H. Baer, V. Barger, N. Nagata, and M. Savoy, "Phenomenological profile of top squarks from natural supersymmetry at the LHC," *Phys. Rev.*, vol. D95, no. 5, p. 055012, 2017.

[5] ATLAS Collaboration, "Search for heavy particles decaying to pairs of highly-boosted top quarks using lepton-plus-jets events in proton–proton collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *ATLAS-CONF-2016-014*, 2016. [Online]. Available: http://cdsweb.cern.ch/record/2141001

[6] ——, "A search for $t\bar{t}$ resonances using lepton-plus-jets events in proton-proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector," *JHEP*, vol. 08, p. 148, 2015.

[7] CMS Collaboration, "Search for resonant $t\bar{t}$ production in proton-proton collisions at $\sqrt{s} = 8$ TeV," *Phys. Rev.*, vol. D93, no. 1, p. 012001, 2016.

[8] ATLAS Collaboration, "Search for production of vector-like top quark pairs and of four top quarks in the lepton-plus-jets final state in pp collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector," *ATLAS-CONF-2016-013*, 2016. [Online]. Available: http://cdsweb.cern.ch/record/2140998

[9] ——, "Search for pair production of heavy vector-like quarks decaying to high-p_T W bosons and b quarks in the lepton-plus-jets final state in pp collisions at $\sqrt{s}$=13 TeV with the ATLAS detector," 2017.

[10] CMS Collaboration, "Search for single production of vector-like quarks decaying to a Z boson and a top or a bottom quark in proton-proton collisions at $\sqrt{s} = 13$ TeV," *JHEP*, vol. 05, p. 029, 2017.

[11] ATLAS Collaboration, "Search for top squarks in final states with one isolated lepton, jets, and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector," *Phys. Rev.*, vol. D94, no. 5, p. 052009, 2016.

[12] CMS Collaboration, "Search for direct production of supersymmetric partners of the top quark in the all-jets final state in proton-proton collisions at sqrt(s) = 13 TeV," 2017.

[13] J. Thaler and K. Van Tilburg, "Identifying Boosted Objects with N-subjettiness," *JHEP*, vol. 03, p. 015, 2011.

[14] ATLAS Collaboration, "Identification of high transverse momentum top quarks in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector," *JHEP*, vol. 06, p. 093, 2016.

[15] CMS Collaboration, "Boosted Top Jet Tagging at CMS," *CMS-PAS-JME-13-007*, 2014. [Online]. Available: https://cds.cern.ch/record/1647419

[16] D. Guest, J. Collado, P. Baldi, S.-C. Hsu, G. Urban, and D. Whiteson, "Jet Flavor Classification in High-Energy Physics with Deep Neural Networks," *Phys. Rev.*, vol. D94, no. 11, p. 112002, 2016.

[17] ATLAS Collaboration, "Identification of Hadronically-Decaying W Bosons and Top Quarks Using High-Level Features as Input to Boosted Decision Trees and Deep Neural Networks in ATLAS at $\sqrt{s} = 13$ TeV," *ATL-PHYS-PUB-2017-004*, 2017. [Online]. Available: https://cds.cern.ch/record/2259646

[18] ——, "Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment," *ATL-PHYS-PUB-2017-003*, 2017. [Online]. Available: https://cds.cern.ch/record/2255226

[19] ——, "Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector," *ATL-PHYS-PUB-2017-017*, 2017. [Online]. Available: https://cds.cern.ch/record/2275641

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017. [Online]. Available: http://doi.acm.org/10.1145/3065386

[21] L. G. Almeida, M. Backovi, M. Cliche, S. J. Lee, and M. Perelstein, "Playing Tag with ANN: Boosted Top Identification with Pattern Recognition," *JHEP*, vol. 07, p. 086, 2015.

[22] G. Kasieczka, T. Plehn, M. Russell, and T. Schell, "Deep-learning Top Taggers or The End of QCD?" *JHEP*, vol. 05, p. 006, 2017.

[23] J. Pearkes, W. Fedorko, A. Lister, and C. Gay, "Jet Constituents for Deep Neural Network Based Top Quark Tagging," 2017.

[24] G. Louppe, K. Cho, C. Becot, and K. Cranmer, "QCD-Aware Recursive Neural Networks for Jet Physics," 2017.

[25] A. Butter, G. Kasieczka, T. Plehn, and M. Russell, "Deep-learned Top Tagging using Lorentz Invariance and Nothing Else," 2017.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov 1997.

[27] T. SjÃűstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, "An introduction to pythia 8.2," *Computer Physics Communications*, vol. 191, pp. 159 – 177, 2015.

[28] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. LemaÃőtre, A. Mertens, and M. Selvaggi, "DELPHES 3, A modular framework for fast simulation of a generic collider experiment," *JHEP*, vol. 02, p. 057, 2014.

[29] CMS Collaboration, "Particle-Flow Event Reconstruction in CMS and Performance for Jets, Taus, and Emiss," *CMS-PAS-PFT-09-001*, 2009. [Online]. Available: https://cds.cern.ch/record/1194487

[30] ——, "Commissioning of the particle-flow event reconstruction with the first LHC collisions recorded in the CMS detector," *CMS-PAS-PFT-10-001*, 2010. [Online]. Available: https://cds.cern.ch/record/1247373

[31] Z. Marshall, "Simulation of Pile-up in the ATLAS Experiment," *J. Phys. Conf. Ser.*, vol. 513, p. 022024, 2014.

[32] M. Cacciari, G. P. Salam, and G. Soyez, "The Anti-k_t jet clustering algorithm," *JHEP*, vol. 04, p. 063, 2008.

[33] ——, "FastJet User Manual," *Eur. Phys. J.*, vol. C72, p. 1896, 2012.

[34] D. Krohn, J. Thaler, and L.-T. Wang, "Jet Trimming," *JHEP*, vol. 02, p. 084, 2010.

[35] S. Catani, Y. L. Dokshitzer, M. H. Seymour, and B. R. Webber, "Longitudinally invariant K_t clustering algorithms for hadron hadron collisions," *Nucl. Phys.*, vol. B406, pp. 187–224, 1993.

[36] F. Chollet, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[37] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," May 2016.

[38] G. Hinton, "Neural networks for machine learning: overview of mini-batch gradient descent." [Online]. Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[39] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv e-prints*, Dec. 2014.

[40] T. Dozat. Incorporating nesterov momentum into adam. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

A. Supplemental Figures

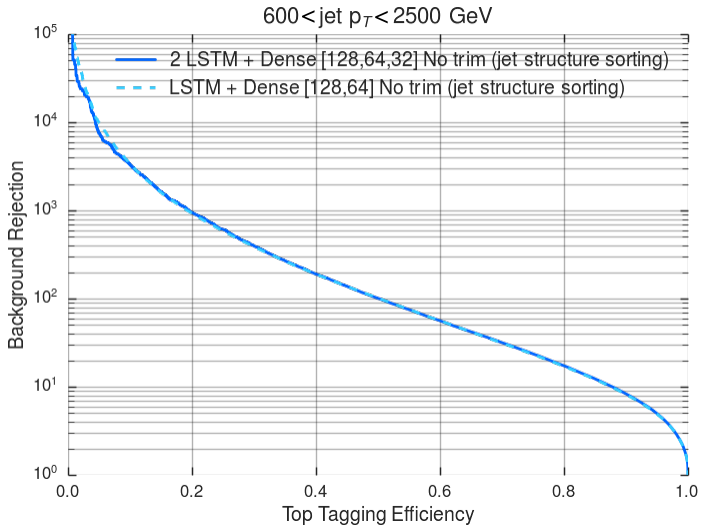

Figure A1: *Response operating characteristic (ROC) curve for networks with one or two LSTM layers. Adding a second LSTM layer gives negligible change in performance.*

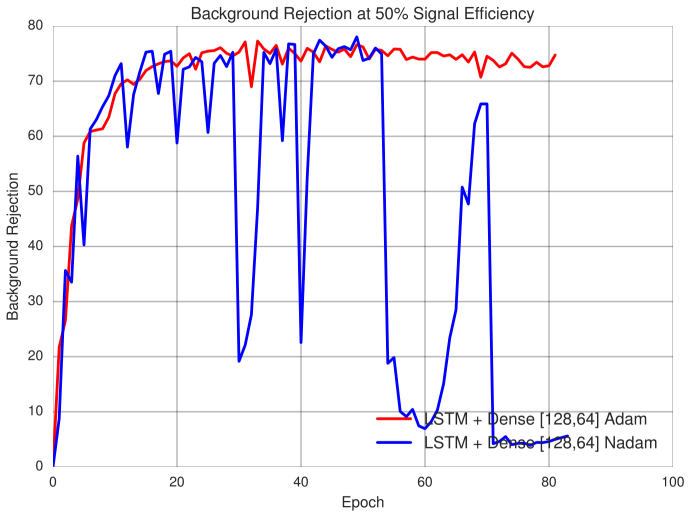

Figure A2: *Training progress of LSTM network demonstrating relative instability of the Nadam optimizer.*