Prepared for submission to JHEP

Recursive Neural Networks in Quark/Gluon Tagging

Taoli Cheng

School of Physics, University of Chinese Academy of Sciences, 100049 Beijing, P. R. China

E-mail: chengtaoli.1990@gmail.com

Abstract: Since the machine learning techniques are improving rapidly, it has been shown that the image recognition techniques in deep neural networks can be used to detect jet substructure. And it turns out that deep neural networks can match or outperform traditional approach of expert features. However, there are disadvantages such as sparseness of jet images. Based on the natural tree-like structure of jet sequential clustering, the recursive neural networks (RecNNs), which embed jet clustering history recursively as in natural language processing, have a better behavior when confronted with these problems. We thus try to explore the performance of RecNN in quark/gluon discrimination. In order to indicate the realistic potential at the LHC, We include the detector simulation in our data preparation. We attempt to implement particle flow identification in one-hot vectors or using instead a recursively defined pt-weighted charge. The results show that RecNNs work better than the baseline BDT by a few percent in gluon rejection at the working point of 50% quark acceptance. However, extra implementation of particle flow identification only increases the performance slightly. We also experimented on some relevant aspects which might influence the performance of networks. It shows even only particle flow identification as input feature without any extra information on momentum or angular position is already giving a fairly good result, which indicates that the most of the information for q/g discrimination is already included in the tree-structure itself. As a bonus, a rough u/d discrimination is also explored.

arXiv:1711.02633v1 [hep-ph] 7 Nov 2017

Contents

1 Introduction

The study of jet substructure has been the very advanced topic in jet physics at the LHC. The techniques in distinguishing different substructures have been established well [1, 2]. And at this stage, different jet observables have been invented to improve our understanding about jets. As for application, we can use jet substructure to detect new physics such as supersymmetry or Two Higgs Double Models. For theoretical development, demand on theoretical calculation of jet substructure observables actually pushes our precision calculation forward and helps us understand QCD better. And the development of Monte Carlo tools also interacts with jet measurements at the LHC.

When the experimental environment gets more complex and the problem has larger and larger dimension, the number of observables we need is increasing, and sometimes so large that exceed our actual computation capability . Artificial neural networks have already been employed for high dimensional problems. Earlier research using neural networks are carried out in a framework of designing observables by hand at first and then feeding these observables into neural networks to do classification. It strongly relies on the pre-stage of expert-features designing, and thus depends on physicists' understanding on the problem. The expert-feature approach has a very long history in high energy physics, but there are several pros and cons. They generally have clear physical intuition. The observables are designed according to physical understanding and theoretical insights. And their behavior are well understood, and based on theoretical framework. However, they can only deal with problems case by case, and

highly depend on the specific processes worked with. And there are always strong correlation between observables. At last, since we are not guaranteed that all the information can be captured in the observables, the best we can do is just approaching the limit by trial and error. And there is no guide for how much information we have captured.

Along with the new framework in machine learning (ML) getting more and more mature, relevant techniques have been employed in high energy physics. The new machine learning framework augmented our ability to fully utilize experimental data. On one hand, the input data can be taken from the raw detector measurements, which means we don't have to lose information because of data transformation or specific observable designing. On the other hand, this new input formulation also brings new insights on how we organize our observations. Finally, the uniform formulation and general approach might help us out of the busy tasks from too many sub-channel designing. Especially for jets and their structure analysis, we have the opportunity to improve our working culture to adapt to the ML era.

There have already been some work using deep neural networks (DNN) in jet physics. The very first attempt was made in using computer vision [3] to help jet tagging. And later people started to use image recognition in boosted top tagging [4, 5], boosted W tagging [6], heavy flavour classification [7], and the investigation of parton shower uncertainties is also made [8]. A detailed report on image recognition in jet substructure is given in reference [9]. And more interestingly, colored version of image recognition inspired by particle flow has also been proposed [10]. Based on all these recent development, it can be shown that deep neural networks generally match or outperform the conventional expert feature approach. And the colored version can perform better than grey scale in some cases since it employs more information. The basic idea of image recognition in jet physics is mapping a jet onto the (η, ϕ) plane and translating the p_T of every constituents of the jet into intensity of the pixel. Thus the higher p_T, the darker the pixel will be. By feeding these jet images after some preprocessing into deep neural networks, we can discriminate signal and background with the raw data from the detector and in avoid of too much human design on the physics problem. And at experiments, CMS has already carried out heavy flavor identification using DNN recently [11].

Until now, most of the work is done in the regime of image recognition, since it is the most intuitive, simple and general approach. However, there are some disadvantages for image recognition:

- sparseness: in most cases only 5-10% pixels are active in jet images of fixed size [9]. Thus most of the parameter space is actually wasted.
- the pixalisation process causes information loss.
- computation cost: too many model parameters require more computation power. Not very efficient for larger image size.
- it's more complicated to do the event-level analysis. Of course one can channel the output of image classifier into later classifier based on expert features or other network

architecture, but that will anyway bring some complication and detailed investigation is necessary.

Aside from image recognition, another natural approach is recurrent neural networks (RNN) [12], which takes sequences as input and is widely used in natural language processing. RNN adapts to the problems confronted in image recognition better since it can properly deal with sequential input of variable length. And also the parameter sharing makes it very efficient. To be more concrete, a recurrent network is structured as a recurrent chain, in which every step takes in the output the last computation, and all the steps share the same set of parameters. Recursive Neural Networks (here abbreviated as RecNN in order not to be confused with recurrent neural networks), rather, has a tree-like structure, other than the chain-like one of RNN. Respect to RNN, RecNN reduces the computation depth from τ to $\mathcal{O}(\log \tau)$. As an example, RNN is explored in [7] for heavy flavor tagging, by taking low level track and vertex information as input for the neural networks.

Base on the natural analogy of the sequentially clustered jet and the input structure of RecNN, , a group [13] has implemented the RecNN version of jet analysis. In their work, the framework is built for embedding jet recursively into a root jet node, and then feeding it into the sequential classifier. And by a simple extension, event-level analysis can be easily implemented in a structure-in-structure manner, although right now only dealing with jets-only events with very limited application range. The work is done in the regime of jet substructure for boosted gauge boson. And the results show that the RecNN outperforms their expert feature approach and image version in DNN.

Motivated by all these progresses and the prospects, we try to explore the performance of RecNN in another very interesting topic: quark/gluon discrimination. Quark/gluon tagging is being gaining great potential at the LHC. Since gluon ($C_A = 3$) has a larger color factor than quarks ($C_F = 4/3$), gluon jets will generally have more radiation and also broader radiation pattern. The ratio of final state counts for gluon and quark jets approximately scales as $\langle N \rangle_g / \langle N \rangle_q \sim C_A/C_F \sim 2$, and the ratio of the variances is $\sigma_g^2/\sigma_q^2 \sim C_A/C_F$. The measurements at the LHC [14] also support this prediction, where the charged particle multiplicity is measured, and shows the tendency of approaching the limit.

The conventional approach in quark/gluon discrimination is defining some jet observables, such as charge multiplicity, jet mass, jet subjettiness [15] (Fig. 1 shows the distributions of a few observables for 1 TeV quark and gluon jets). These jet observables have turned to be very efficient and well-motivated discriminants. For a multivariable analysis, the general performance is: for a 50% quark acceptance, the gluon rejection can reach 80 % - 90 %, and the corresponding significance can have an increase by a factor of 2 - 3. These results show a great potential in helping new physics search, and thus the exploration in DNN which can make use of all the low level information is worthwhile. The image approach using CNN has been explored [10], showing performance matching or outperforming conventional expert feature approach. And the DNN simulation from ATLAS [16] and CMS [17] using either CNN or RNN also confirm the potential. For a 50% quark acceptance at jet $p_T \sim 200$

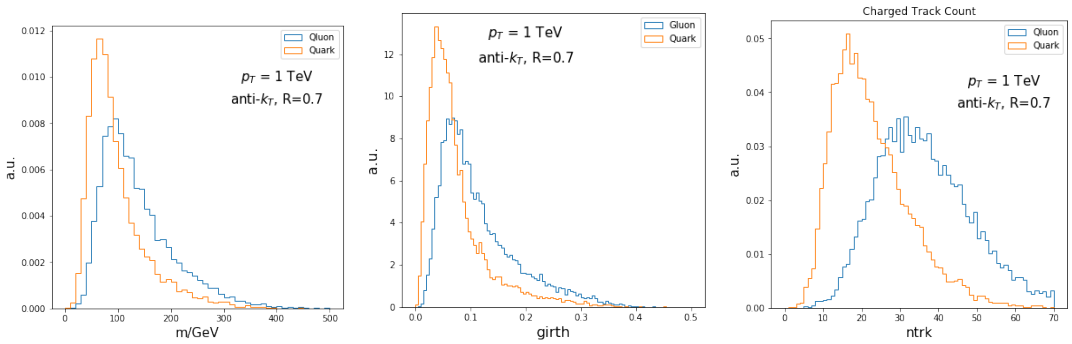

Figure 1. Distributions of jet scaled mass, girth, and track count for quark and gluon jets with p_T = 1 TeV.

GeV, mis-identification rate is approximately 10% (a few percent away from the pythia-level results).

In this work, we aim to address the following questions:

- At first, how does the RecNN generally perform in q/g discrimination at the LHC.
- We include faster detector simulation in our analysis, trying to bring a more realistic picture to the problem.
- How to better implement particle flow identification in RecNN, in order to gain a more containing information set.

The paper is organized as following: in Section 2, we describe in detail the process of jet embedding and the neural network architecture. Then we explore the RecNN performance in q/g discrimination in Section 3. Conclusions are given in Section 4. And finally a short discussion and outlook in Section 5 is presented.

2 Methodology

2.1 Jets at the LHC

There are several different definitions [18] of a "jet". How to connect the theoretical parton and the experimental measurements on a collimated spray of hadrons is not a trivial problem. But in practice, the jet is operationally defined by the jet algorithm used to group particles into clusters.

The basic idea of sequential jet clustering [19] is to recombine the particles into pseudojets recurrently according to some measure and recombination scheme. The "closest" particles will be combined into pseudojets at first. And by this QCD-inspired measure, we can get soft and collinear safe jet definition. And through this recursive clustering process, the global

information including number of particles, their momenta, clustering history, is embedded in the tree structure of the clustering sequence.

Here shows the typical tree-structures of a 1 TeV gluon and quark in Fig. 2. A naive im-

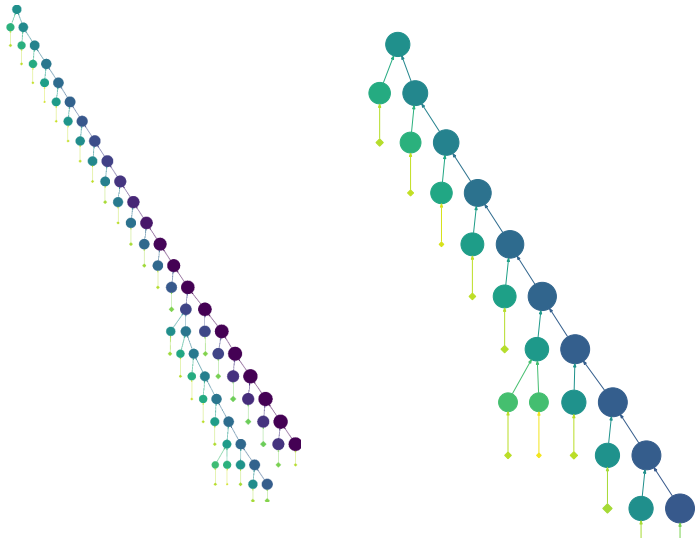

Figure 2. Typical tree structures for 1 TeV gluon jet (left) and quark jet(right).

pression is that the gluon has more constituents and accordingly more complicated clustering structure. Actually the mean content number of a 1 TeV gluon jet is ~ 90, while for a quark jet is approximately 50. And the typical number of charged constituents for a 1 TeV gluon jet is 36, and for a quark jet is 22. Thus if we input all the measured degrees of freedom into analysis, for every final state particle we have 3 d.o.f (p_T, η and ϕ) and one more if particle flow identification is taken into account, thus for a whole jet the total d.o.f can reach $\sim 4 \times 100$. If we want to utilize all the information we've gotten, such high dimensionality naturally brings us to the DNN approach, although there are arguments about how much information actually is included in a jet [20].

2.2 Network Architecture

Here first gives a short introduction to RNN. Recurrent neural networks are initially designed for processing sequential data. The state of the system is recurrently defined. For the t-th step, the state $\mathbf{h}^{(t)}$ is defined by the output of last step and the new feed as:

$$\begin{aligned} \mathbf{h}^{(t)} &= g^{(t)}(x^{(t)}, x^{(t-1)}, ..., x^{(1)}) \\ &= f(h^{(t-1)}, x^{(t)}; \theta) \end{aligned} \tag{2.1}$$

where the same transition function f is applied for every step, thus this parameter sharing makes the model very simple and efficient. And the recursive networks, rather, use a tree-structure instead of a chain-like structure. As shown in the right part of Fig. 3, the matrices

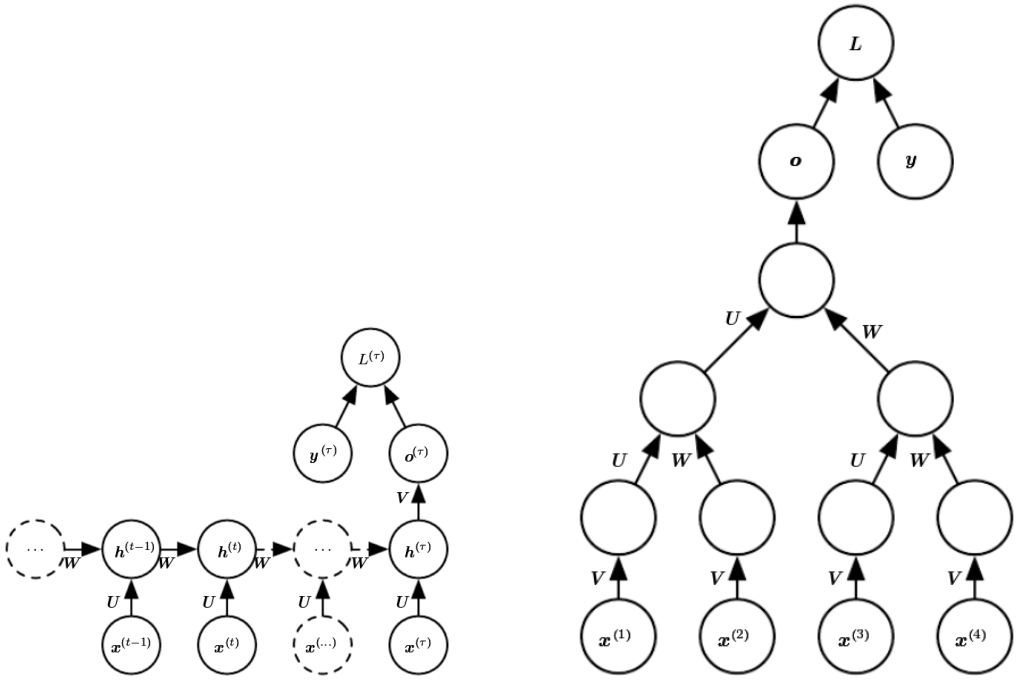

Figure 3. **Left**: Time-unfolded computational graph for a RNN which only takes the output at the end of the sequence. **Right**: Time-unfolded computational graph for a RecNN. Plots are taken from Ref. [12]

W, **U** and **V** are shared parameters by all the steps. In Fig. 3, unfolded computational graphs of a typical RNN and a RecNN are depicted. And there is one hidden-to-output at the end of the sequence. This output can be a final loss function or can be fed into following classifier. In brief, it summarizes a sequence and gives a fixed-length representation.

Based on the similarity between jet clustering and RecNN architecture, the application of RecNN is straightforward. The raw data from detector can be used as direct input to the networks. The measured transverse momentum p_T and the angular location (η, ϕ) give the basic feature set. After a regular jet clustering process, the jet (defined by the clustering structure **t** with its contents $\{v_i, i = 1, .., N_j\}$, where v_i denotes the four-momentum vector of i-th particle within the jet) is embedded recursively into an embedding space of fixed size, then the embedded jet node will be channeled to following classifier (a MLP).

According to Ref. [13], a jet is recursively embedded into a single jet node $\mathbf{h}_1^{\text{jet}}$. And through this recursive embedding, the history of jet clustering can be included in the final jet node. The procedure of embedding is a mapping from the feature space (with dimension f) to the embedding space (with dimension q) $R^f \to R^q$. For every jet with N_j constituents, there are $2N_j - 1$ clustering nodes. Every node is represented by an input vector in the embedding space $\mathbf{u} \in R^q$ by a transformation from feature vector $\mathbf{x}$ (generally can de defined as (p_T, η,

ϕ) or other version with the same information contents) as following:

$$\mathbf{u}_k = \sigma(W_u \mathbf{x}_k) + b_u) \quad \text{for the k-th node} \tag{2.2}$$

where $W_u \in R^{q\times f}$, $b_u \in R^q$ and σ denotes the ReLU activation function.

Then, the embedding of every node is defined by its children and its own input feed (we can also define the embedding by simply using the children. This version is also explored and shows similar performance while reducing the model parameters, see discussion in the next section):

$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \\ \mathbf{u}_k \end{bmatrix} + b_h\right) \tag{2.3}$$

where we have model parameters $W_h \in R^{q\times 3q}$, $b_h \in R^q$. k_L and k_R are left and right children of node k respectively. Down to the leaves (the original jet constituents), we have directly $\mathbf{h}_k = \mathbf{u}_k$. Thus, combining the raw input of detector measurements and jet clustering history, we can embed the information finally into a single root jet node $\mathbf{h}_1$, which is passed to a following classifier for final classification. All the embedding parameters (W_u, b_u, W_h and b_h) are learned using backpropagation jointly with the parameters of the following classifier W_{clf} and b_{clf}, by trying to minimize the loss function.

The procedure is depicted as following:

$$[\{\mathbf{t}, \{v_i, i = 1, .., N_j\}\} \to \mathbf{h}_{\text{jet}} \in R^q] \to \text{ReLU} \to \text{ReLU} \to \text{Sigmoid} \tag{2.4}$$

where the rectified linear unit [21] (ReLU) $= \max\{0, z\}$ is used for the hidden layers in the classifier, and Sigmoid ($\frac{1}{1+e^{-z}}$) activation is used for the output layer. And the log loss function is employed:

$$L = -\frac{1}{N}\sum_i^N (y_i \log(y_i^{pred}) + (1 - y_i) \log(1 - y_i^{pred})) \tag{2.5}$$

where y_i is the label for i-th jet, and y_i^{pred} is the prediction of the model.

Recursively defined jet charge:

Other than the very basic input set (p_T, η, ϕ), more information can be included. At the LHC, the particle flow algorithm [22] combines the information through different parts of the detector thus gives more identification ability. It can match tracks to the energy deposit in the calorimeters, thus we have more accurate knowledge about the final states, and also higher precision on their transvers momenta. Right now, we can identify charged tracks, neutral particles and photons within one jet. But how to implement this information in RecNN deserves some exploration. They can be implemented in a naive way as one-hot vectors (($i_{\text{neutral hadron}}$, i_{photon}, i_+, i_-), $i = 0$ or 1) added to the feature vector. However, since the one-hot implementation doesn't has an additive nature (or, the particle flow identification of

the inner nodes is not well-defined), the recursive embedding can't utilize this information effectively. In Ref. [13], the authors claimed that including particle flow identification wouldn't gain any significant improvement for their attemp in discriminating boosted W jets and QCD jets.

In order to search for a better way to implement the particle flow identification, we ask help from a jet observable: jet electric charge. Jet charge is a very useful observable for identifying jet flavor and the identification W' and Z'. The pt-weight jet charge [23, 24] is defined as following:

$$Q_\kappa^J = \sum_{i \in J} (\frac{p_T^i}{p_T^J})^\kappa q_i \ , \tag{2.6}$$

where q_i is the electric charge of the particle within the jet, and p_T^i is the transverse momentum of the component, which p_T^J denotes the total transverse momentum of the jet. κ is a free parameter, and $\kappa \to 0$ gives the limit of simply adding charges of the components while $\kappa \to \infty$ gives the limit of the charge of the hardest component. The typical value has been used by experiments lies between 0.2 and 1.0.

Trying to carry the particle flow information in the RecNN, we construct the recursively defined pt-weighted charge for the clustering tree:

$$Q_k^{\text{rec}} = \frac{Q_{k_L}^{\text{rec}} (p_T^{k_L})^\kappa + Q_{k_R}^{\text{rec}} (p_T^{k_R})^\kappa}{(p_T^k)^\kappa} \tag{2.7}$$

while for the leaves $Q_k^{\text{rec}} = q_i$. The pt-weighted charge of k-th node Q_k^{rec} is defined by its children in the same manner as in Eqn. 2.6. In this way, we still get the right pt-weighted charge for the jet node at the end of the embedding. This Q_k^{rec} along with the traditional feature set consist a almost-complete set of the information we get from detectors. $\kappa = 0.5$ will be used throughout this work.

Prepared with all the basic settings, we depict the whole architecture in Fig. 4.

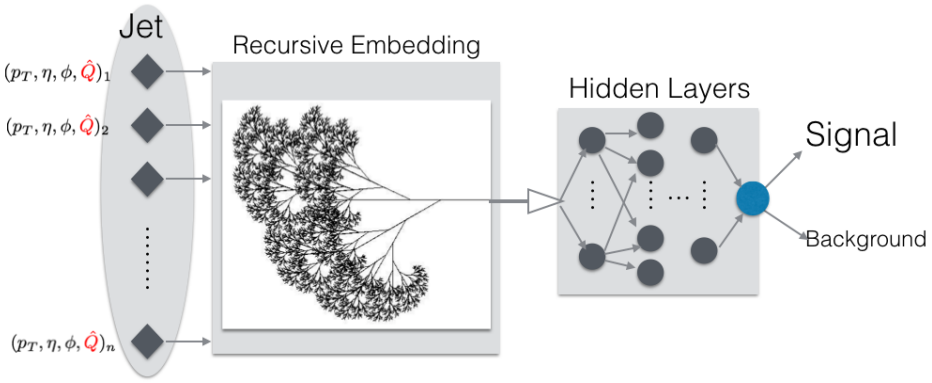

Figure 4. Workflow including jet embedding and the following classification.

2.3 Setup

HEP Data Preparation:

In preparing sample jets, Pythia8 [25] is used to generate events and carry out the parton shower and the following haronization. Delphes [26] is used for fast detector simulation (pile-up effects are left aside for the time being). Jets are clustered using FastJet [27]. And the data is generally formatted in hd5f using interfaces in deepjets [8].

Preprocessing:

Before being fed into the RecNN embedding, some basic preprocessing steps are applied to the samples. We only apply the necessary translation and rotation to extract "pure" jets disconnected to the rest of the event.

- translation: jets are translated to be centered at $(\eta, \phi) = (0, 0)$.
- rotation: the jet is rotated in the (η, ϕ) plane such that the energy deposition axis is the same (here is (1,0)) for all samples in order to eliminate the effects of global influence from color connections with the remnants of the event (but it will be very interesting that we have a solution on event level which takes the color connections seriously). Thus only the intra-jet correlation is used for analysis. The rotation reads as:

$$\eta = \hat{\eta}\cos\alpha + \hat{\phi}\sin\alpha \tag{2.8}$$

$$\phi = -\hat{\eta}\sin\alpha + \hat{\phi}\cos\alpha \tag{2.9}$$

where the rotation angle α is defined by the "principal axis" [4]:

$$\tan\alpha = \frac{\sum_i \frac{\hat{\phi}_i E_i}{\Delta\hat{R}_i}}{\sum_i \frac{\hat{\eta}_i E_i}{\Delta\hat{R}_i}} \tag{2.10}$$

Neural Networks Setup:

The training is carried out with library Scikit-learn [28] and the RecNN framework[1] built by [13]. And the subsequent classifier we are using consists of two hidden layers with rectified linear unit (ReLU) activations, and the final output node is equiped with sigmoid activation. And the Adam [29] algorithm is employed for the minimization.

The input feature vector finally fed into the networks is $\mathbf{x}_i = (p_i, \eta_i, \phi_i, E_i, E_i/E_J, p_{Ti}, \theta_i = 2\arctan(\exp(-\eta_i)))$. The dimension of embedding space is set to be 40 (which has been proven to be large enough for the case we are examining and also not too large to preserve computation time). The training is done through 10 epoches with the batch size of 64. The learning rate is set to be 0.0005 initially and is decayed linearly by a rate of 0.9 for every epoch. The training set consists of 50,000 - 100,000 (the exact number depends on the case) data samples and among which 5,000 is used as validation set to prevent overfitting. And the performance is tested with a dataset of 10,000 - 20,000 samples.

[1] https://github.com/glouppe/recnn

3 Results

In this section we show the performance of RecNN on the task of discriminating q/g jets. And some effects from variants are discussed. As a byproduct, we also explored the first step on (light-)jet flavor identification in Subsection 3.2.

We show here the discrimination power in use of Receiver Operating Characteristic (ROC) curve, which is the standard measure (1/(false positive rate) v.s true positive rate) and sometimes is plotted as the signal efficiency v.s. background rejection rate (1 - f.p.r) for a more intuitive presentation. The Area Under the Curve (AUC) of ROC is the overall metric to measure the performance. Generally, the larger the AUC, the better the performance. And as baseline, we also give the Boosted Decision Tree (BDT) results using expert features as input.

3.1 Quark/Gluon Discrimination

The processes used to sample jets are $qq \to gg$ and $gg \to gg$ for gluon jets, and $gg \to q\bar{q}$, $q\bar{q} \to q\bar{q}, qq \to qq$ for light quarks (u,d,s), at $\sqrt{s} = 13$ TeV. A set of benchmark jet p_Ts are examined. The p_T bins are: [90,110], [180, 220], [480, 520], [950, 1050] (in GeV). Jets are clustered using anti-kt algorithm [30] and cone size is set to $R = 0.7$ for high p_T ($p_T = 1$ TeV) and $R = 0.4$ for other relatively lower p_Ts. For pythia level analysis, we employ the constraints of $|\eta| < 2.5$, and discard neutrinos before clustering.

We first investigated the RecNN performance with the architecture indicated in the previous section, ROCs are shown and comparison with image approach (CNN) and BDT is also carried out. The baseline of BDT is composed of scaled mass (m_J/p_T), charged particle multiplicity, and girth ($g = \sum_{i\in \text{Jet}} \frac{p_T^i}{p_T^J} r_i$). For RecNN we tried three scenarios: 1) without extra particle flow identification; 2) with one-hot vector implementation of particle flow identification; 3) with recursively defined pt-weighted charge Q^{rec} instead. We compared the results for different levels: pythia level, delphes e-flow and delphes towers. Then for delphes e-flow, we studied the jet-p_T dependence. Furthermore, we explored the behavior of several variants: change cone size for jet clustering; modification of the input feature set; modification of the embedding feed; etc.

To show the physical implications more clearly, the acceptance and rejection rate can be translated into a significance improvement (SI) factor for any working point as:

$$\sigma \equiv \frac{S}{\sqrt{B}} \;\to\; \frac{\epsilon_S S}{\sqrt{\epsilon_B B}} = \left(\frac{\epsilon_S}{\sqrt{\epsilon_B}}\right)\sigma \;\to\; \text{SI} = \frac{\epsilon_S}{\sqrt{\epsilon_B}} \tag{3.1}$$

Thus, the ROCs can be mapped into significance improvement curves (SICs) [31]. We also show the corresponding SICs in the following.

In Table 1, we present the gluon efficiency at the working point of 50% quark acceptance, at benchmarks $p_T = 200, 1000$ GeV. We also present the results from Ref. [10] for CNN (pythia level) and [17] for CMS simulation in DNN (CMS Collaboration has done the simulation within three different DNN architectures: DeepJet, LSTM, CNN). We first show

Gluon Jet Efficiency (%) at 50 % Quark Jet Acceptance	200 GeV	1000 GeV
BDT of all jet variables	*5.2**	*5.2**
Deep CNN without Color	*4.8**	*4.0**
Deep CNN with Color	*4.6**	*3.4**
RecNN without pflow	6.4	4.5
DNN@CMS	$\sim$10.0†	–
BDT	9.5	6.2
RecNN without pflow	7.8	4.6
RecNN with categorical pflow	7.1	4.5
RecNN with pt-weighted charge	7.8	4.9

Table 1. Results displayed for different scenarios for comparison. Data with * is CNN results taken from Ref. [10], and data with † is take from [17] (with extrapolation) for quick comparison. In the upper part, we show the particle level (pythia level) results; and in the lower part, detector simulation is included, thus generally reduces the discrimination power a bit.

the results for pythia level in the upper part of the table. And the performance after detector simulation is shown in the lower part. In pythia-level, RecNN is not working much better, but still matches the previous results from BDT and CNN. Interestingly, we observe that for RecNN, the detector effects actually don't change the performance very much as long as the particle flow algorithm is applied. The RecNN is still give excellent performance after fast detector simulation. The numbers show that it's obviously surpassing BDT and there is a potential for full detector simulation. According to our experiments, using only towers can't provide significant discrimination for q/g, thus we don't put the results here. And compared to pythia level, the delphes level got more influce from pt. The decrease in jet p_T reduces the performance much quicker. The detector responses at different p_Ts deserve more careful investigation.

In Fig. 5, we display the ROCs and corresponding SICs for benchmark transverse momentum $p_T = 1$ TeV. On the left panel, we show the ROCs curves for RecNNs (with three scenarios concerning pflow information) and the baseline BDT. For a signal efficiency of 50%, the $\sim$ 95% background can be rejected. The three RecNNs give better performance than BDT. And for a signal efficiency of 80%, the mid-identification rate is about 23%. For CNN within image recognition approach, the particle flow information implemented in colored channels [10] (where the detector effects are not taken into account) gives slight improvement for high p_T jets (500 GeV - 1 TeV) (for $p_T = 1$ TeV, the gluon efficiency drops from 4.0% to 3.4% for working point of 50% quark acceptance, and the best SI increases by a factor 1.2), while leaving performance on low p_T jets not much changed. The CMS simulation results [17] show the colored CNN is not working better than DeepJet architecture or LSTMs. However, the particle flow information implemented here seems not helping the performance much (although there is a slight increase. And in contrast to CNN, here the increase is more visible for relatively lower jet p_Ts. see Table. 2 for detailed numbers), as claimed in [13] for

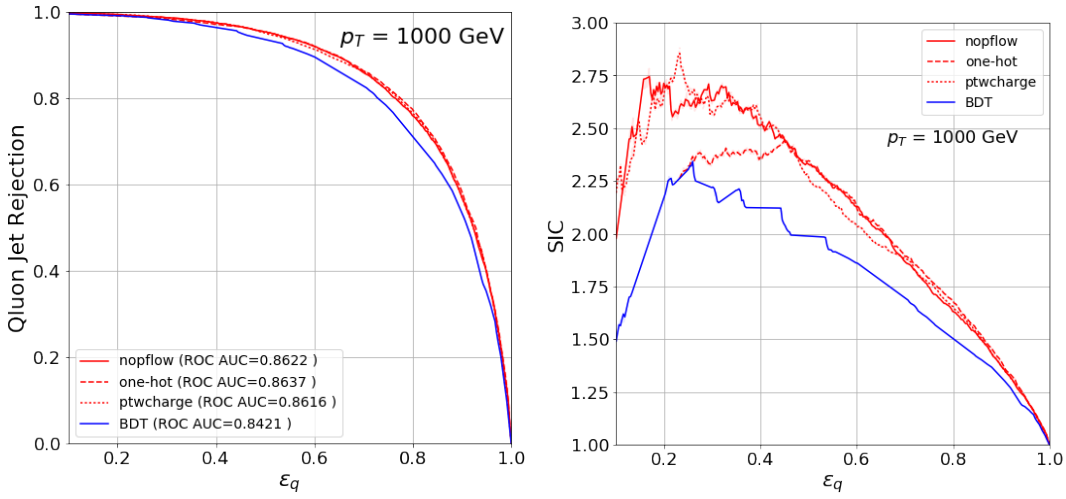

Figure 5. ROCs (left) and SICs (right) for jet p_T = 1 TeV. Baseline BDT and three scenarios concerning particle flow information are displayed."nopflow": no extra particle flow identification is added to RecNN; "one-hot": one-hot implementation of particle flow; "ptwcharge": recursively defined pt-weighted charge implemented in the embedding process.

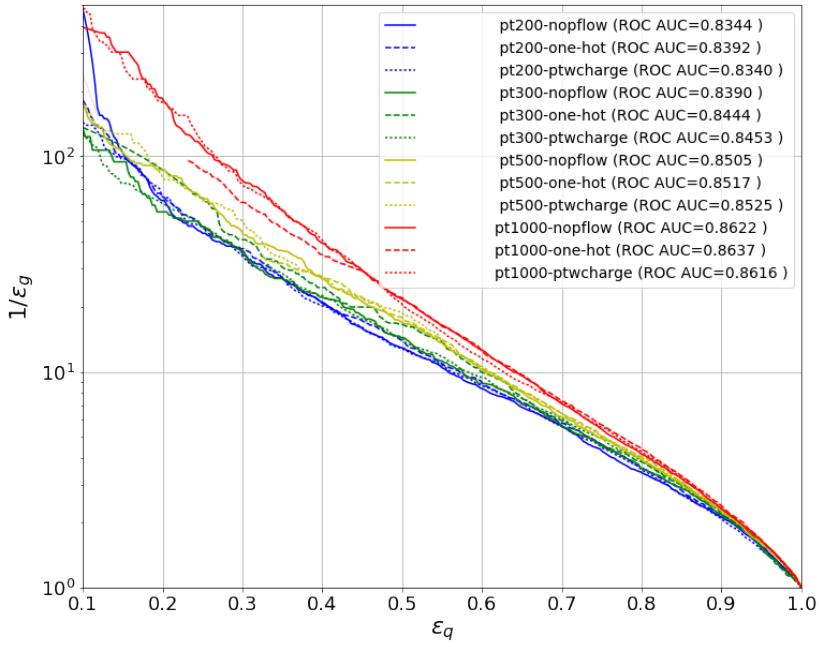

Figure 6. ROCs of RecNNs for various jet p_Ts: 200 GeV, 300 GeV, 500 GeV, and 1 TeV. Notations are the same as in Fig. 5.

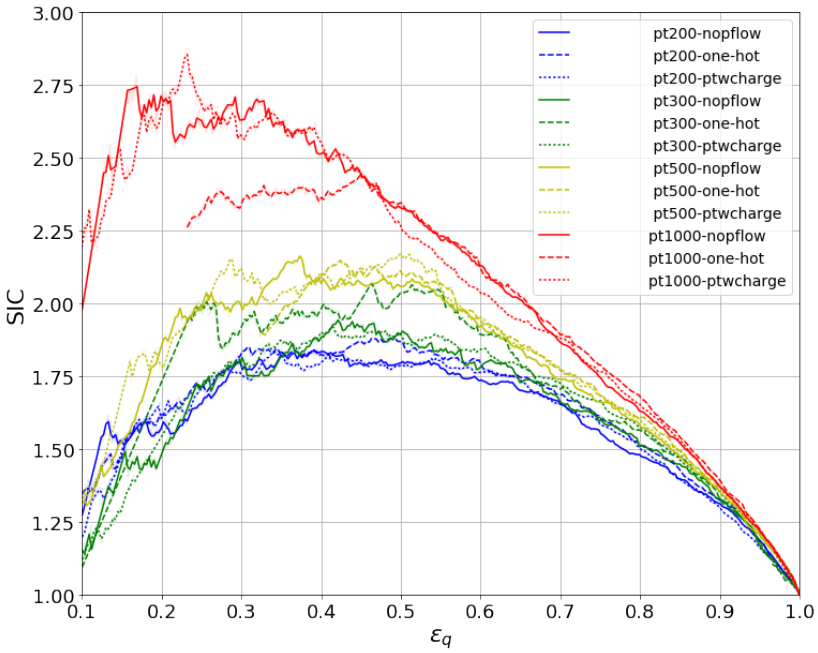

Figure 7. SICs of RecNNs for various jet p_Ts: 200 GeV, 300 GeV, 500 GeV, and 1 TeV. Notations are the same as in Fig. 5.

W tagging. There may be several reasons for this indifference here: the dominance of the particle multiplicity; the saturation of information; or these are not the best way to implement particle flow. On the right hand side of Fig. 5, we show the SICs. The significance can be improved by a factor of ∼2.75 (best SI) for a signal efficiency of 0.2 - 0.3, and by a factor of ∼1.7 for $\epsilon_S = 0.8$.

To see the performance on different p_Ts, we show the ROC curves of RecNN for different bins of jet p_T in Fig. 6. Different from Fig. 5, here we are showing the ($1/\epsilon_g$ v.s ϵ_q) for a clear comparison. The discriminating power increases along with jet p_T, which coincides with the behavior of conventional approach, since for higher p_T, the multiplicity ratio between gluon and quark $\frac{N_g}{N_q} \sim \frac{C_A}{C_F} = \frac{9}{4}$ increases and slowly reaches its asymptotic limit. The SICs are shown in Fig. 7.

For a detailed tabulation of results in numbers, one can find the AUCs and background rejection rates for different p_Ts in Table 2. Here the rejection rate is defined as $R_{\epsilon_S} = 1/\epsilon_B$ @ ϵ_S.

Variants in Network Details and Jet Clustering

Moreover, we examined several variants (minor modifications) of the procedure in order to find out how are they affecting the performance. For simplicity, the experiments are only carried with samples of $p_T = 200$ GeV. The results are shown in Table. 3.

- Preprocessing: it has shown that the preprocessing is necessary for a sane result.

ROC AUC | $R_{\epsilon=80\%}$ | $R_{\epsilon=50\%}$	200 GeV	300 GeV	500 GeV	1000 GeV
BDT	0.8164 | 3.1 | 10.5	0.8443 | 3.8 | 16.5	0.8385 | 3.5 | 14.1	0.8421 | 3.6 | 16.1
RecNN without pflow identification	0.8344 | 3.4 | 12.9	0.8390 | 3.6 | 14.4	0.8505 | 3.9 | 16.9	0.8623 | 4.2 | 21.9
RecNN with categorical pflow	**0.8392** | 3.6 | **14.0**	0.8443 | 3.8 | **16.5**	0.8517 | 4.0 | 17.8	**0.8637** | 4.4 | **22.0**
RecNN with pt-weighted charge	0.8340 | 3.5 | 12.8	**0.8453** | 3.9 | 14.5	**0.8525** | 4.0 | **18.6**	0.8616 | 4.3 | 20.4

Table 2. AUCs and background rejection rates for different jet p_Ts. The baseline BDT and three scenarios concerning particle flow identification are considered. The largest AUCs and $R_{\epsilon=50\%}$s are highlighted in bold.

- Cone size: larger cone size of jet clustering of 0.7 is examined.
- Dismiss the self-representation u_k for embedding (since all the information is already contained by the children) in Eqn. 2.3, i.e. Eqn. 3.2. Thus we can reduce the number of embedding parameters almost by 1/3 with $W_h \in \mathcal{R}^{q*3q} \to W_h \in \mathcal{R}^{q*2q}$. And we found not much difference, which means we can even reduce the size of the model while maintaining the performance.

$$\mathbf{h}_k = \sigma\left(W_h \begin{bmatrix} \mathbf{h}^{\text{jet}}_{k_L} \\ \mathbf{h}^{\text{jet}}_{k_R} \end{bmatrix} + b_h\right) \tag{3.2}$$

- We explored different variants of the input feature set, and found that the choice of input features doesn't really matter here. The default input feature vector is $\mathbf{x}_i = (p_i, \eta_i, \phi_i, E_i, E_i/E_J, p_{Ti}, \theta_i = 2\arctan(\exp(-\eta_i)))$. We experimented other sets: (p_T, η, ϕ); (η, ϕ); (p_T); and only particle flow identification. One can find in Table 3 that for all the sets we have tried, $R_{\epsilon=50\%}$ can at least reach 11.3 (i.e. mis-identification rate is 8.8%). What is intriguing, only particle flow information without any other information from the momentum or angular location still gives us a fairly good result. This again corresponds to the dominance of constituents count. There might be redundant information in the full set.
- Multiparton Interaction (MPI): we examined the effects of MPI, and no significant difference was observed.

These experiments have shown that in the case of q/g discrimination, RecNN is quite robust against the variances in input features. And there is still large space for even simplifying the model. This is partially due to the fact that in q/g tagging the discrete particle count already dominates, thus most of the information is already contained in the structure itself.

3.2 (Light) Quark Jet Flavor

Although the particle flow information doesn't show much help in q/g discrimination, but it might have other use. As a bonus, we also checked the RecNN performance on a more difficult task: light quark flavor identification. Since for all the light quarks, the dominant QCD effects are universal. And also no heavy-flavor final states, which can have long enough life time such

Variants	AUC	$R_{\epsilon=50\%}$
Baseline	0.8344	12.9
R=0.7	0.8210	12.4
$W_h \to R^{q\times 2q}$	0.8268	12.3
$W_h \to R^{q\times 2q}$ with one-hot	0.8313	13.7
$\mathbf{x}=(p_T,\, \eta,\, \phi)$	0.8291	11.8
$\mathbf{x}=(\eta,\, \phi)$	0.8249	11.9
$\mathbf{x}=(p_T)$	0.8264	11.6
only one-hot	0.8255	11.9
$\mathbf{x}=(Q^{\text{rec}}_{\kappa=50\%})$	0.8234	11.3

Table 3. Comparison of variants in the network settings. Here "Baseline" is the original RecNN without pflow in Table. 2.

that leaves secondary vertices at the detector, are present. A possible discriminating element would be remnant electric charge, although the main effects might be washed out by parton show and hadronization. However it's been shown [24] that even at the LHC, there is still a great potential for measuring jet electric charge. In Fig. 8, we show the distributions of pt-weighted jet charges ($\kappa = 0.5$) for u-quark initiating jets and d-quark initiating jets with $p_T = 1$ TeV.

Jet charge as an useful tool to identify initiating light quark flavor has been measured at the LHC [32, 33]. And it's obviously promising if we can proceed further in this direction.

Thus as an intentional bonus, we also explored the RecNN for (light) jet flavor identification. We take u/d discrimination as an attempt. Jets are sampled from the production of scaled-up W and Z bosons, at the 13 TeV LHC: $pp \to W'/Z' \to q\bar{q}$. Reconstructed jet are matched to the initiating partons by the criterion of $\Delta R < 0.5$ between the center of jet and the parton. And the network architecture and the training settings are similar to the ones in Subsection 3.1.

The results are shown in Fig. 9 and Table 4. The RecNN without pflow and the one-hot implementation is not showing any discriminating power. Only the hybrid version with pt-weighted charge carried in the embedding is behaving. We also examined several variants: using only tracks; using tracks and photons. The results showed that using only tracks is enough. A comparison is made between the single observable pt-weighted jet charge and RecNN. This hybrid implementation shows the basic yet not better performance.

This small experiment shows although no new improvements respect to the original observable of jet pt-weighted charge, but it can be useful for a larger framework for multiclassification in jet flavors (gluon, b-tagging, c-tagging, light flavor identification). And it also might be useful for boosted W/Z jet tagging.

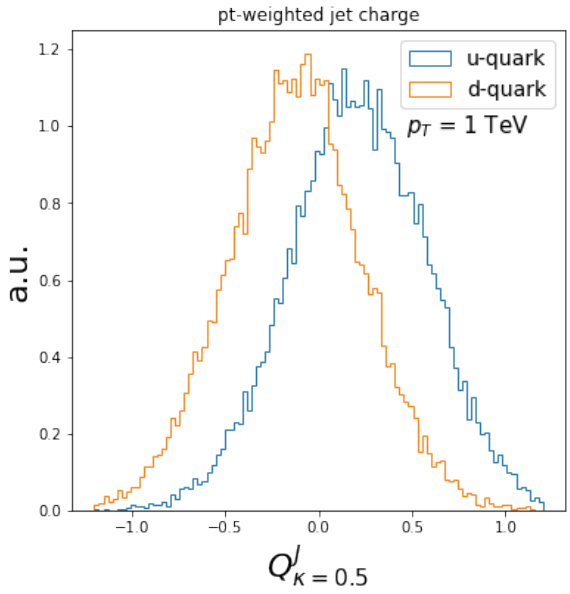

Figure 8. Distributions of pt-weighted jet charge ($\kappa = 0.5$) for jets initiating from u-quark and d-quark with $p_T = 1$ TeV.

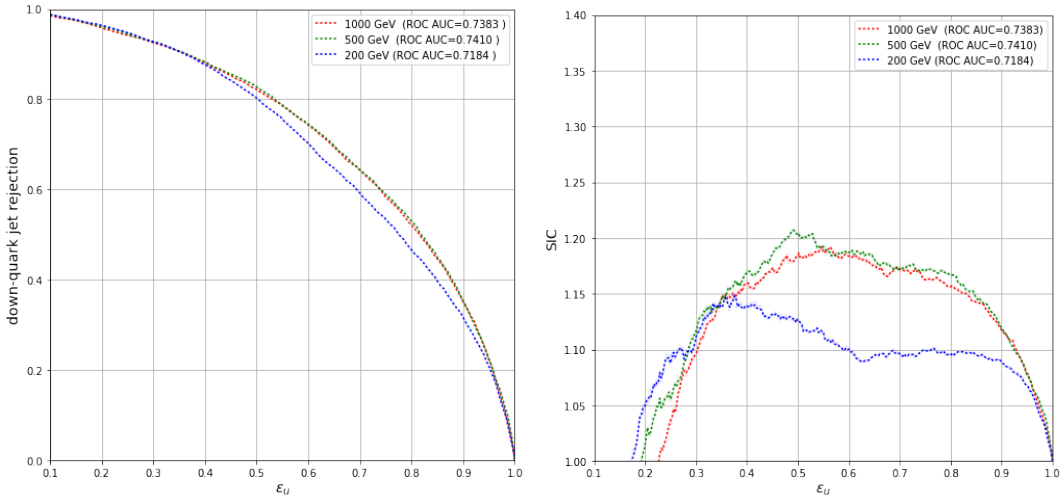

Figure 9. On the left panel, ROCs of RecNNs are shown for u/d discrimination; on the right hand side, corresponding SICs are shown. Benchmark p_Ts inlcude 200 GeV, 500 GeV and 1 TeV.

4 Conclusions

Based on the fact that machine learning has been evolving quickly enough to help with some complex tasks which wouldn't be possible in the conventional approach, the investigation of

ROC AUC | $R_\epsilon = 50\%$	200 GeV	500 GeV	1000 GeV
$Q_{\kappa=0.5}$	0.7317 | 5.3	0.7487 | 5.9	0.7437 | 5.7
RecNN with pt-weighted charge	0.7184 | 5.1	0.7410 | 5.8	0.7383 | 5.6

Table 4. Discriminating u/d quark with pt-weighted charge information implemented in RecNN. The traditional single pt-weighted charge is used as baseline.

its application including formulating data, preprocessing, architecture, and the interpretation of the results should be addressed carefully.

Motivated by the advantages of efficiency and nice structure of RecNN, and the previous findings on it capability, we in this work try to explore its application in an advanced topic for jet physics at the LHC: quark/gluon discrimination. And in order to make sure that we make use of all the information collected at the detectors, we further explored the implementation of particle flow in RecNN. Besides, some variants for the network details are examined.

In the RecNN approach, the raw data taken from detectors are fed into the networks directly without information loss. By embedding the jet data through a tree-like recursive process and subsequently feeding into following classifier, we have a natural and efficient DNN strategy for jet tagging. Along with the naive one-hot implementation, we examined an alternative by recursively defining a pt-weighted charge for every node of the jet tree. We first investigated the performance of RecNN on pythia-level data, i.e. the hadrons without detector effects. Then we employed fast detector simulation and took the detector measurements as input. At this stage, particle flow objects and only towers in the calorimeters are both examined. While input of towers not giving significant results, the particle flow input can give discrimination power of rejecting $\sim$95% at $\epsilon_q = 50\%$ for jets with $p_T = 1$ TeV. With only fast simulation, the numbers show better potential then the full simulation results from CMS Collaboration in three different NN architectures. As for extra particle flow identification implemented in RecNNs, slight increase can be generally observed (especially for lower jet p_Ts), but not significant enough. We examined several variants on the details of the procedure, and interestingly the results showed that even only with particle flow identification, RecNN still has fairly good performance. This might indicate that most of the information for q/g discrimination is already contained in the tree-structure itself. Pile up effects are not taken into account in this work, and jet grooming is also not examined here. These can be left for future work.

As a byproduct, we also apply the recursively defined charge to a more difficult task in jet physics: jet flavor (light quarks) identification. It actually is the simplest extension from the conventional pt-weighted jet charge to its DNN version. And it's showing no better performance, but still gives the discrimination power at the same level. We hope it will helps in further study on multiclassification in jet physics.

Thus as conclusions, we have:

- The results with detector simulation indicate a great potential for RecNN in q/g dis-

crimination.

- RecNN is robust against the variances in input feature sets. The tree structure itself already contains most of the information in q/g discrimination, this is partly because of the fact that the particle multiplicity dominates.

- Extra particle flow identification is not showing significant effects in q/g discrimination.

5 Discussion and Outlook

There are several interesting aspects and extensions which deserve further investigation. Here we briefly discuss on a few of them.

Event Level Analysis

Actually a jet can't be isolated from the remnants of the event, although we can get quite "pure" jets by grooming. Color connections can be very useful in many cases. How to manifest these effects also deserves attention. And it might be potential if we can deal with them better in event-level analysis.

Easy to be extended to event-level analysis is an important motivation for RecNN. It is natural to be augmented into lager hierarchical structure. The event analysis with only jets has been explored in [13]. A simple RNN chain is used there for constructing events from jets. About the implementation in event-level, how to structure the whole event is not trivial. Every event can be seen as a structured data tree, and the whole information of one event just reside within the properties of the nodes and links between these nodes. How to properly represent every object and its connections with other parts of the event would be crucial for designing NN architectures.

Jet Algorithms as Unsupervised Learning Procedure

In the framework of DNN, accordingly adjusting the jet clustering might help to gain better performance. Actually jet finding itself can be treated as a minimization problem [34–37], thus it will be very interesting if we can naturally include the process of jet finding in event-level implementation.

Application: New Physics

Since new physics will have interesting patterns when concerned with their new spectrum and decay modes. Such as SUSY events will generally produce large amounts of final states (more complicated hierachical structure), and some times several soft leptons in electroweakino search. Whether DNNs have better tolerance on this kind of topologies is also worthwhile for investigation.

Acknowledgments

This work is partially supported by Natural Science Foundation of China under grant no. 11475180. And the author would like to thank Lei Wang for discussion on ML application in

condensed matter physics, and giving some nice advice; Zhao Li and Hao Zhang for collaboration which inspires part of this work. More importantly, a big thank-you goes to Gilles Louppe for information on some techincal details of the RecNN package, and Edmund Noel Dawe for updating the python version of the deepjets package. The author would also like to acknowledge the MCnet summer school during which part of this work was carried out.

References

[1] A. J. Larkoski, I. Moult, and B. Nachman, *Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning*, arXiv:1709.04464.

[2] D. Adams et al., *Towards an Understanding of the Correlations in Jet Substructure*, *Eur. Phys. J.* **C75** (2015), no. 9 409, [arXiv:1504.00679].

[3] J. Cogan, M. Kagan, E. Strauss, and A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, *JHEP* **02** (2015) 118, [arXiv:1407.5675].

[4] L. G. Almeida, M. Backovi, M. Cliche, S. J. Lee, and M. Perelstein, *Playing Tag with ANN: Boosted Top Identification with Pattern Recognition*, *JHEP* **07** (2015) 086, [arXiv:1501.05968].

[5] J. Pearkes, W. Fedorko, A. Lister, and C. Gay, *Jet Constituents for Deep Neural Network Based Top Quark Tagging*, arXiv:1704.02124.

[6] P. Baldi, K. Bauer, C. Eng, P. Sadowski, and D. Whiteson, *Jet Substructure Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev.* **D93** (2016), no. 9 094034, [arXiv:1603.09349].

[7] 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.* **D94** (2016), no. 11 112002, [arXiv:1607.08633].

[8] J. Barnard, E. N. Dawe, M. J. Dolan, and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks*, *Phys. Rev.* **D95** (2017), no. 1 014018, [arXiv:1609.00607].

[9] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman, and A. Schwartzman, *Jet-images deep learning edition*, *JHEP* **07** (2016) 069, [arXiv:1511.05190].

[10] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, arXiv:1612.01551.

[11] **CMS Collaboration** Collaboration, *Heavy flavor identification at CMS with deep neural networks*, .

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

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

[14] **ATLAS** Collaboration, G. Aad et al., *Measurement of the charged-particle multiplicity inside jets from $\sqrt{s} = 8$ TeV pp collisions with the ATLAS detector*, *Eur. Phys. J.* **C76** (2016), no. 6 322, [arXiv:1602.00988].

[15] J. Gallicchio and M. D. Schwartz, *Quark and Gluon Jet Substructure*, *JHEP* **04** (2013) 090, [arXiv:1211.7038].

[16] **ATLAS Collaboration** Collaboration, *Quark versus Gluon Jet Tagging Using Jet Images with the ATLAS Detector*, Tech. Rep. ATL-PHYS-PUB-2017-017, CERN, Geneva, Jul, 2017.

[17] **CMS Collaboration** Collaboration, *New Developments for Jet Substructure Reconstruction in CMS*, .

[18] P. Gras, S. Hoeche, D. Kar, A. Larkoski, L. Lnnblad, S. Pltzer, A. Sidmok, P. Skands, G. Soyez, and J. Thaler, *Systematics of quark/gluon tagging*, arXiv:1704.03878.

[19] G. P. Salam, *Towards Jetography*, *Eur. Phys. J.* **C67** (2010) 637–686, [arXiv:0906.1833].

[20] K. Datta and A. Larkoski, *How Much Information is in a Jet?*, *JHEP* **06** (2017) 073, [arXiv:1704.08249].

[21] X. Glorot, A. Bordes, and Y. Bengio, *Deep sparse rectifier neural networks*, in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudk, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 315–323, PMLR, 11–13 Apr, 2011.

[22] **CMS** Collaboration, A. M. Sirunyan et al., *Particle-flow reconstruction and global event description with the CMS detector*, *JINST* **12** (2017), no. 10 P10003, [arXiv:1706.04965].

[23] R. D. Field and R. P. Feynman, *A Parametrization of the Properties of Quark Jets*, *Nucl. Phys.* **B136** (1978) 1.

[24] D. Krohn, M. D. Schwartz, T. Lin, and W. J. Waalewijn, *Jet Charge at the LHC*, *Phys. Rev. Lett.* **110** (2013), no. 21 212001, [arXiv:1209.2421].

[25] T. Sjstrand, 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*, *Comput. Phys. Commun.* **191** (2015) 159–177, [arXiv:1410.3012].

[26] **DELPHES 3** Collaboration, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lematre, A. Mertens, and M. Selvaggi, *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057, [arXiv:1307.6346].

[27] M. Cacciari, G. P. Salam, and G. Soyez, *FastJet User Manual*, *Eur. Phys. J.* **C72** (2012) 1896, [arXiv:1111.6097].

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Scikit-learn: Machine learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[29] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *CoRR* **abs/1412.6980** (2014) [arXiv:1412.6980].

[30] M. Cacciari, G. P. Salam, and G. Soyez, *The Anti-k(t) jet clustering algorithm*, *JHEP* **04** (2008) 063, [arXiv:0802.1189].

[31] J. Gallicchio, J. Huth, M. Kagan, M. D. Schwartz, K. Black, and B. Tweedie, *Multivariate discrimination and the Higgs + W/Z search*, *JHEP* **04** (2011) 069, [arXiv:1010.3698].

[32] **CMS** Collaboration, C. Collaboration, *Measurement of jet charge observables in dijet events at* $\sqrt{s} = 8$ TeV, .

[33] **ATLAS** Collaboration, G. Aad et al., *Measurement of jet charge in dijet events from* $\sqrt{s}$*=8TeV pp collisions with the ATLAS detector*, *Phys. Rev.* **D93** (2016), no. 5 052003, [arXiv:1509.05190].

[34] L. Angelini, G. Nardulli, L. Nitti, M. Pellicoro, D. Perrino, and S. Stramaglia, *Deterministic annealing as a jet clustering algorithm in hadronic collisions*, *Phys. Lett.* **B601** (2004) 56–63, [hep-ph/0407214].

[35] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov, *Towards a standard jet definition*, *Phys. Rev. Lett.* **91** (2003) 061801, [hep-ph/0301185].

[36] D. Yu. Grigoriev, E. Jankowski, and F. V. Tkachov, *Optimal jet finder*, *Comput. Phys. Commun.* **155** (2003) 42–64, [hep-ph/0301226].

[37] I. Volobouev, *FFTJet: A Package for Multiresolution Particle Jet Reconstruction in the Fourier Domain*, arXiv:0907.0270.