Augmented Recursive Neural Networks for Jet Physics

Yurui Mu, Sudhir Nallam, Xiaoyu Wang, Wenqing Zhu
New York University, Center for Data Science
{ym1495, psn240, xw1435, wz1070}@nyu.edu

Abstract

Applying deep neural networks for classification of Jet substructure has seen much progress in recent years. Ref. [1] proposed to use Recursive Neural Networks to embed the sequential recombination of jet algorithms for detecting new particles, like W boson, to known particles. In this paper we propose two extensions to the Recursive Neural Network model. Firstly to apply Network in Network[2] strategy and use 1×1 convolution as activation function replacing ReLU activation, which has shown an accuracy increase on k_t structure dataset. Secondly, we propose to combine pivotal adversarial network[3] with Recursive Neural Net, to build robust models for data with different pileup levels.

1 Introduction - Wenqing

1.1 Background

Jet physics has been studied for decades. Jet is a structure can be seen at Large Hadron Collider (LHC) and it is caused by the collision of energetic hadrons, which are made up of quarks and gluons. Quantum chromodynamics (QCD) is a quantum field theory that describes the strong interaction between quarks and gluons, and it is a basic part of standard model in particle physics. In QCD processes, there will be different jets produced, and they can be treated as two parts, one is the most common seen mundane QCD processes, which interferes the process of source determination; the other one is a certain set of jets which is “highly boosted standard model particles decaying hadronically”, and the probability to find it is low. When this certain type of highly boosted hadronically decaying particles called W boson is found, its decay products will make up a fat jet with a characteristic substructure. The study of jet structure has been a very advanced topic and attracted lots of interests in recent years.

Machine learning is a field of computer science. After decades of development, now it has become a multidisciplinary subject that can be applied in several fields, especially in scientific disciplines. In recent days, apply machine learning techniques for study of jet substructure has been the very advanced topic in jet physics. Many methods use the “calorimeters and images” [4][5] analogy to solve the jet classification problem, but these image-based networks encounter the trouble of dealing with multiple free parameters and training large amounts of data. So in Louppe et al’s paper[1], it presents an approach based on the application of structure of sequential recombination jet algorithms and deep neural networks. In their work, they study jet physics built upon an analogy between QCD and natural languages,and use Recursive Neural Network to leverage techniques of sentences’ semantic structure to represent the jet substructure as a "jet embedding", which maps a set of 4-momenta (vector representing the particle) into a root jet node, and then feed a sequence of jet embeddings into the sequential classifier. This method shows that the recursive architectures will obtain a more accurate result and more data efficient than previous image-based networks.

In our project, based on the work of Ref.[1], we apply a few techniques to make the model proposed in Louppe at el's paper more robust to systematic uncertainties. In this work, we propose two augmented recursive neural network models. The first one is using $1 * 1$ convolution as activation function instead of ReLu function. The other one is combining pivotal adversarial network[3] with Recursive Neural Network. We train these two models on different pileup levels data.

1.2 Problem Statement

Same as the approach in Louppe et al's paper [1], our goal in this project is to explore and find W boson, the highly boosted hadronically decaying particles from the fat jet of their decaying product. We describe a collision event as being composed of a varying number of particles, and each particle is represented by its 4-momentum vector. The 4-momenta in each event are then clustered into jets with a recombination jet algorithm. More specifically we are using two optimal jet recombination methods: k_t among jet clustering and desc-p_T among sequential sorting algorithm to train models with.

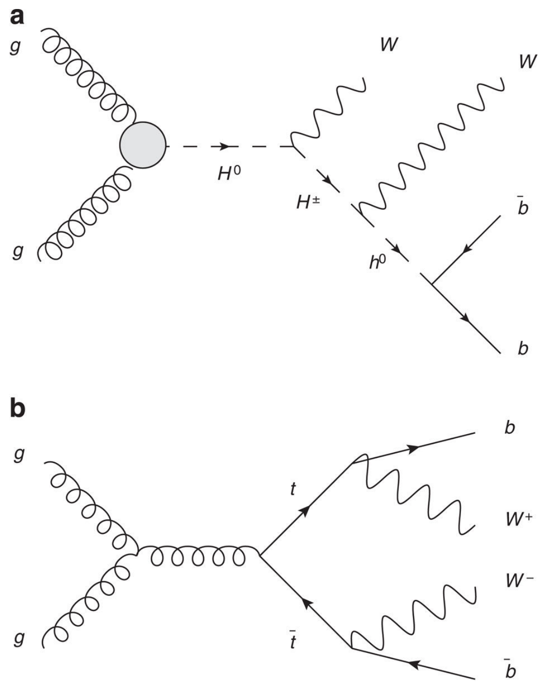


Figure 1: Signal Process and Background Process in Diagrams: Diagram (a) describes the signal process involving new exotic Higgs bosons H^0 and $H_\pm$. While Diagram (b) describes the background process involving top quarks t. In both cases, the resulting particles are two W bosons and two b-quarks.[6]

In this project, we aim to perform a jet-level binary classification to distinguish signal process with label $y = 1$ corresponding to a hadronically decaying W boson with $200 < p_T < 500GeV$ from the background processes on the basis of previous QCD-Aware Recursive Neural Net[1].
Our goal is to experiment on different tricks adding to original model and expect to improve the model performance in the aspect of accuracy and robustness. First, We would like to find a model which might improve prediction accuracy by adding a twist, 1×1 Convolution in Simple RNN.

Secondly, as in realistic settings, proton-proton collisions usually occur simultaneously during high illuminosity runs[7], thus generating pileup data. We would like to apply inference method and utilize pivotal adversarial loss in Simple RNN to build a more robust model for a steady performance on data with different pileup levels.

2 Related Works

Image recognition in deep neural networks is a common seen technique using to detect jet substructure. As shown in reference[8], this group creates 'jet-image' to represent data by using image processing for jet tagging at the LHC. Jet images is defined by a fixed size grid in $(\eta * \phi)$ space, and a variable-length set of 4-momenta projected into it. By applying this method, the original jet classification problem will be changed into an image classification problem, and train two models including Convolutional Neural Networks (CNN), and Fully Connected (FC) MaxOut networks in order to understand LHC events better and design a new classification method. Although most works is done using image recognition technique, and as the reference[8] says, this method has good performance on high energy physics, we cannot ignore its drawbacks, including:

- sparseness of jet images
- the projection process of jets causes information loss
- the training process requires large amounts of data
- too much free parameters result in high computation cost

By recognizing the disadvantages of image recognition techniques, another natural approach is shown up, which is recurrent neural networks (RNN)[9]. The recurrent network is structured as a recurrent chain, it can deal with the variable length of each sequence. And in RNN, all the steps share the same parameters, which saves lots of computation cost. So RNN can adapt the above drawbacks of image recognition better. Recursive Neural Networks is like an updated version of RNN by changing the recurrent chain into a tree-like structure. By applying this change, the computation depth is shorten thus reduce the cost effectively.

Unlike the group implement the recursive neural network for jet substructure problems in paper Ref.[1], Cheng explores the performance of recursive neural network in quark/gluon discrimination[10]. In Cheng's work, based on the conventional approach in quark/gluon discrimination which defines some jet observables[11], Cheng explores how the Recursive Neural Network performs in q/g discrimination at the LHC, which is much better than baseline; and he implements particle flow identification, which only increases the performance slightly.

3 Data

3.1 Jets Level Data and Architectures

In orgininal paper from Professor Cranmer[1], Louppe and Cranmer thoroughly studied substructures of jet physics and explored Recursive Neural Network performances on jet-level classification task under multiple conditions: both particle-level inputs and towers input from DELPHES simulation, multiple types of architectures, and with or without image preprocessing.

According to the experiment results, there shows a significant increase of Area Under the ROC Curve(AUC) score and Inverse of False Positive Rate(1/FPR) at 50% signal efficiency($R_{\epsilon=50\%}$) when not using jet images, comparing to the results when using jet image preprocessing[1], suggesting that image projection may lead to loss of informations. Moreover, according to previous experiment result in the paper, RNN works better with particles input than towers input with higher AUC score and $R_{\epsilon=50\%}$. Thus in this project we are using jet level data without image processing.

Moreover, from previous works, Professor Cranmer experimented with multiple recombination algorithms recombining 4-momenta events into jets, including sequential recombination jet clustering algorithms such as anti-k_t[12], k_t[13], Cambridge-Aachen[14], and simple sequential sorting algorithms based on p_T, such as desc-p_T and asc-p_T.

In mentioned jet clustering algorithms, the distance measures to cluster particles are defined as

$$d_{ii'}^{\alpha} = min(p_{ti}^{2\alpha}, p_{ti'}^{2\alpha}) \frac{\Delta R_{ii'}^2}{R^2} \tag{1}$$

where $\alpha = 1, 0, -1$ each corresponds to k_t, Cambridge-Aachen and anti-k_t algorithms[1]. As for sequential sorting algorithm, it builds binary tree topology by simply sorting particles by p_T values.

Desc-p_T has the highest p_T value at the root of the tree while asc-p_T has the lowest p_T at the root.[1]

According to previous experiments, k_T and Cambridge-Aachen methods outperform anti-k_T method among jet clustering algorithms, and desc-p_T outperforms asc-p_T methods[1]. Thus we choose to use desc-p_T and k_t structured data to compare RNN performance with our proposed Convo-RNN model performance.

In each of our training and testing datasets, we have $100,000$ data entries(jets), each contains 4-momenta vectors of all the particles in the jets, and features of jets like mass, η, p_T, energy and constructed binary tree structures. The label $(y = 1)$ corresponds to a hadronically decaying W boson with $200 < p_T < 500GeV$, while $(y = 0)$ corresponds to QCD jet with the same range of p_T.[1]

3.2 Pileup Data

Pileup data is a more practical data source with additional proton-proton interactions which generated from Large Hadron Colliders(LHC).[15] And it has been a lasting task in jet physics to produce precise classification from hadronic final states while in presence of additional proton-proton collisions occurring at the same time during high illuminiosity runs.[7]

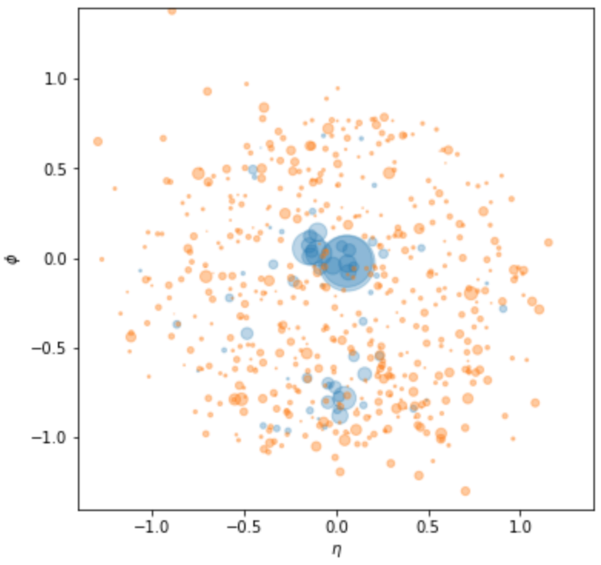


Figure 2: One example from the pileup 50 sample (blue = "interesting" particles from the high-energy interaction, orange = pileup particles, area of circles = energy associated to them).

Since high illuminosity is essential in many practical experimental setting, such pileup jets often occur in reality complicating the process of analyses. Traditionally such pileup data are treated with Monte Carlo simulations with high model dependency.[16] There are other approaches dealing with average correction procedures. In 2007, Cacciari also proposes an eventbyevent, jetbyjet pileup subtraction approach[7]. And in 2014 a SoftKiller, a particle level pileup removal method was also introduced

Since our second task is to apply pivotal adversarial model in Recursive Neural Net to gain robustness when training with pileup data, we are also using pileup data with four pileup levels: 25, 40, 50, 60 under k_t architecture kindly provided by Professor Cranmer and Johann Brehmer. The pileup level 25 represents LHC realistic conditions for LHC Run I and pileup level 40,50,60 represents realistic conditions for LHC Run II.[17]

3.3 Data Preprocessing

In our dataset, we have 100,000 data entries in training set, and 100,000 other data entries in test set, which is independent of the training data. Among the 100,000 data entries of training set, we have both signal and background events with equal prior.

In many Machine learning tasks, researchers and programmers need to rescale dataset. Most frequently, people rescale data by removing mean and rescale data by standard deviation. However this method sometimes could not perform as well when dealing with outliers. Thus in this project, we import Robust Scaler from Scikit-learn package in Python, which scales features using statistics that are robust to outliers. Instead of removing mean and rescale by standard deviation, Robust Scaler removes data median and rescale dataset according to quantile range.

After receiving rescaled training set, we also dump to save the transformation fitting function, so that we could rescale the test set the same way we did for training set, to keep scaling consistency for the two data sets.

Other than data rescaling, we also used data cropping method for our pileup datasets to use the data with desirable p_T and jet mass that fit in our proposed range. The original crop default for non-pileup dataset was set to be $150 < p_T < 220$ and $50 < m < 110$. However given different characteristics on pileup 40, 50, 60 datasets, we then follow advices from Johann Brehner and apply three different levels of cropping methods:

1. For pileup 40, we crop the data by $360 < p_T < 435$ and $220 < m < 300$
2. For pileup 50, we crop the data by $390 < p_T < 460$ and $245 < m < 350$
3. For pileup 60, we crop the data by $420 < p_T < 490$ and $270 < m < 360$

as a compromise between not including edge effects in the distributions and getting as close as possible to the efficiencies of the original samples.

4 Model

4.1 Recursive Neural Network based Model

4.1.1 Recursive embedding

In order to correctly classify jets label on individual jets level, it is crucial that we provide an effective and efficient jets embedding to recover a structure dictated by QCD and organized via the clustering history of a sequential recombination jet algorithm from particle level data. A central idea of the our modeling approach is recursive embedding. This enables a more natural simulation of the jets recombination as well as the minimal amount of information loss.

In the case of individual jets, particles are topologically structured as a binary tree t_j. The embedding of h_k^{jets} of node k is recursively defined as following:

$$h_k^{jet} = \begin{cases} u_k & if\ k\ is\ a\ leaf \\ \sigma\left(W_h \left\{ \begin{array}{c} h_{KL}^{jet} \\ h_{KR}^{jet} \\ u_k \end{array} \right\} + b_h \right) & otherwise \end{cases}$$

$$o_k = \begin{cases} v_{i(k)} & if\ k\ is\ a\ leaf \\ o_{k_L} + o_{k_R} & otherwise \end{cases}$$

$$u_k = \sigma(W_u g(o_k) + b_u) \tag{2}$$

where k from 1,.....$2N_j - 1$ denotes the nodes of the binary tree t_j, the left and right children of node k is denoted by k_L and k_R. W_h, b_h, W_u, and b_u are the shared parameters to be learned. σ is the activation function and g is a function that extract the kinematic features, p, η, θ, ϕ, E and p_T from the 4-momentum o_k.

Starting with the root node and apply h_k^{jet} recursively down to the leaf nodes of the binary tree, we received an embedding as a single vector of fixed size. Moreover, the structural information of particles forming jets is effectively summarized/preserved in this embedding.

The jets embeddings $h_1^{jet}(t_j)$ are then passed to the network structure f for a classification task. There are two set of parameters in this RNN based model, the first set of parameters controls the recursive embedding process, the second set of parameters controls the classifier parameters. These two sets of parameters are both learned using backpropagation to minimize the $Loss^{jets}$

4.1.2 1 by 1 Convolution

In the previous work of [1], the activation function used in the recursive embedding is ReLU function. This is applied at the formation of all internal nodes of the binary structure as well as the root node. However, jets recombination from a physics point of view is a far more complex process. In order to maintain a high level approximation of this process, we further proposed adding 1 by 1 convolution layers before applying the ReLU activation to increase the level of complexity of nonlinearity.

The idea of using 1x1 convolution is sometimes called Network In Network which is derived in the work of [2]. A convolution layer with 1x1 convolution kernel is equivalent to the cross channel parametric pooling layer. The cross channel pooled feature map pooled features across channel again and again, which allows for complex and learnable interactions of across channel information.

In our model implementation, we added three layers of 1x1 convolution layer. This further increases level of nonlinearity and enables the formulation of internal nodes to simulate a more complex function and thus advances the expressiveness of recursive embedding structure.

4.2 The Intuition of Finding Pivot

A common challenge for the machine learning community the existence of systematic uncertainty in the data generating process, under which the training data collected not being representative of the real world data.

Based on Adversarial networks, a novel idea was introduced in the work of [citation pivot paper]. They proposed a flexible learning structure for enforcing the decision function f(X) is pivot with respect to nuisance parameters Z and derived theoretical results for proving this procedure converges towards a model that is both optimal and statistically independent of the nuisance parameters.

In this framework, a function f with parameters θ_f is pit against an adversarial model r := $p\theta_r(z|f(X;\theta_f) = s)$ with parameters θ_r and associated loss $L_r(\theta_f, \theta_r)$. The adversarial model takes predictions s of $f(X;\theta_f)$ as input and produces a function modeling the posterior probability density $p\theta_r(z|f(X;\theta_f) = s)$.

The intuition behind this model structure is that if $p(f(X;\theta_f) = s|z)$ varies with z, then the corresponding correlation can be captured by r. On the other hand, if $p(f(X;\theta_f) = s|z)$ is invariant with z, then the performance of r should be poor and as it is predicting z no better than random guessing.

Thus maximizing the performance of f while in the same time minimizing performance of r can be viewed as a regularization on f towards:

$$p(f(X;\theta_f) = s|z) = p(f(X;\theta_f) = s|z') \tag{3}$$

for all z, and all values s of $f(X;\theta_f)$.

4.3 Augmented Model

In reality, the presence of pileups made our classification problem more difficult. As the pileup interactions produce additional particles, they contribute significant energies to jets and further interferes the classification correctness.Thus it is essential that we can find a classification model that is robust to pileup interactions.

Inspired by the [3], we augmented the initial objective of learning a RNN based classification model $f : X \to S$ with parameters θ_f that minimizes a loss $L_f(\theta_f)$ with an adversarial loss, and through the co-training process to find a classifier f as a pivotal quantity with respect to the level of pileup interactions.

We pit the Recurrent Neural Network based model against the adversarial model so that inference based on $f(X;\theta_f)$ will be robust to the pileup interaction level z. This is realized by training f and r simultaneously, thus the new objective function can be formulated as:

$$E(\theta_f, \theta_r) = L_f(\theta_f) - \lambda \cdot L_r(\theta_f, \theta_r) \tag{4}$$

The loss function are formulated as:

$$L_f(\theta_f) = \mathbb{E}_{x\sim X}\mathbb{E}_{y\sim Y}[-\log p_{\theta_f}(y|x)] \tag{5}$$
$$L_r(\theta_f, \theta_r) = \mathbb{E}_{s\sim f(X;\theta_f)}\mathbb{E}_{z\sim Z|s}[-\log p_{\theta_r}(z|s)] \tag{6}$$

As the assumption of existence of an optimal and pivotal classifier may not hold in practice, the introduction of a hyper parameter λ in equation (5) is to control the trade-off between decision function optimality and pivotality. In order to find the best balance, we further experimented on different λ values and evaluate the effects on model performance.

Through taking stochastic gradient descent steps alternatively, we can solve Equation(5) and reach the minmax solution:

$$\hat{\theta_f}, \hat{\theta_r} = arg \min \max E(\theta_f, \theta_r) \tag{7}$$

The algorithm can be summarized as below:

Algorithm 1 Adversarial training of a classifier f against an adversary r.

```
Inputs: training data $\{x_i, y_i, z_i\}_{i=1}^N$; Outputs: $\hat{\theta}_f, \hat{\theta}_r$.
1: for $t = 1$ to $T$ do
2:     for $k = 1$ to $K$ do
3:         Sample minibatch $\{x_m, z_m, s_m = f(x_m;\theta_f)\}_{m=1}^M$ of size $M$;
4:         With $\theta_f$ fixed, update $r$ by ascending its stochastic gradient $\nabla_{\theta_r} E(\theta_f, \theta_r) :=$
               $\nabla_{\theta_r} \sum_{m=1}^{M} \log p_{\theta_r}(z_m|s_m);$
5:     end for
6:     Sample minibatch $\{x_m, y_m, z_m, s_m = f(x_m;\theta_f)\}_{m=1}^M$ of size $M$;
7:     With $\theta_r$ fixed, update $f$ by descending its stochastic gradient $\nabla_{\theta_f} E(\theta_f, \theta_r) :=$
           $\nabla_{\theta_f} \sum_{m=1}^{M} [-\log p_{\theta_f}(y_m|x_m) + \log p_{\theta_r}(z_m|s_m)],$
       where $p_{\theta_f}(y_m|x_m)$ denotes $\mathbf{1}(y_m = 0)(1 - s_m) + \mathbf{1}(y_m = 1)s_m$;
8: end for
```

In this augmented RNN based model, we have three set of parameters, the recursive embedding parameters and classifier parameters are both learned through backpropogation on $L_f(\theta_f)$, while the adversarial model parameters are learned through backpropagation on $L_r(\theta_f, \theta_r)$. Thus, final embeddings of jets are also reshaped by the effect of adversarial loss.

5 Empirical Studies

As stated earlier this work is extension of [1], so we will give some background of the experimental setup used in the published paper[1]. Experiments at high energy colliders consist of searching for new particles beyond those described by the Standard Model(SM) of particle physics. The paper explored classifications on two levels: jet-level classification and event-level classification. Series of jets form an event e. In this work we will concentrate on Jet-classification.

In original paper[1], input for the Jet-level classification is the 4-momenta v_i from both the particle-level data and the DELPHES tower. As we are using particle-level input for jet-level classification problem, data is clustered using initial $\alpha = 1$, k_t jet clustering algorithm as in Equation (1), which identifies the constituents of the highest p_T jet.

Further the jets are subjected to translation, rotation and reflection preprocessing steps. Apart from the mentioned binary tree topology based on k_T algorithm, we also considered replicating the results of the desc-p_T binary trees, since desc-p_T architecture performs the best and achieves the highest background rejection rate at 50% signal efficiency.

5.1 Recursive Neural Net with Convolution Layer

Our addition to the Ref. [1] is in two-fold, first we have replaced the ReLU activation in the RNN architecture with Network-in-Network Ref. [2]. We have used convolution with kernel size (1,1) and stride of 1. We have used a 3 layer setup of this network. Figure 3 shows the 1x1 convolution setup. The additional architecture has increased the number of parameters form by 10. Which is far less than the parameters used for the MaxOut architecture in Ref. [18].

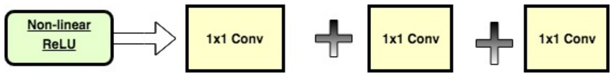


Figure 3: ReLU to Network-to-Network conversion

5.2 Recursive Neural Net with Adversarial Loss

Secondly for handling the pileup/nuisance parameters that effect the performance of the model, we have trained the Recurrent neural network along with the adversarial loss. Jet streams comes with additional noisy data called pileup, or multiple proton-to-proton interactions occurring simultaneously with the primary interactions. These pileup interactions produce unrelated energies confusing the classification model. For making the classification model robust to these pileup data, we use Adversarial network.

Adversarial network is fully connected 3-layered Neural network with ReLU activations, which takes input the outcome of the RNN model and produces the pileup variable. Figure 4 shows the architecture of the classification model with adversarial loss. We used artificially generate pileup data for three cases, Z = 40,50 and 60. Adversarial network predicts has softmax on these three z values.

We have pre-trained the classification model for 10 epochs, before initializing the adversarial training. We varied the hyper-parameter λ to get a balance between accuracy of the classification model and the robustness of the model to pileup data. We have taken λ = 1, 10, 50, 100 and 500 and plotted the variation in the conditional probability distribution of the decision scores for different z-values.

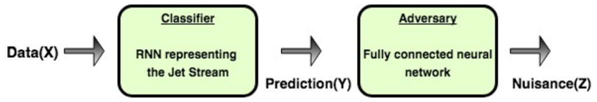

Data(X)
Classifier
RNN representing the Jet Stream
Prediction(Y)
Adversary
Fully connected neural network
Nuisance(Z)

Figure 4: Classification model with adversarial loss

5.3 Experiment Setup

Training was carried out on a dataset of 100,000 jet stream signals with equal priors and evaluation on a independent test set of 100,000 signals. While training Recursive Neural Network with 1×1 convolution layer as activation function, we set number of training epochs as 50, batch size as 100, step size as 0.001, hidden size as 40, parameter of decay as 0.94 and crop the dataset as we described in Section 3.3. As for training Recursive Neural Nets with Adversarial Loss with pileup dataset, we have seen a huge growth of the size of data. Because in each jet, there includes other particles from proton-proton collisions, and in each jet data, the number of particle 4-momenta vectors almost double up and so does number of tree edges. Thus given the huge size of pileup data and limited computation resource, we set number of training epochs as 10, and z as each pileup level. Other than that, we keep all other parameters the same as Simple RNN.

5.4 Evaluation Metrics

We have used two evaluation metrics for measuring the performance of the model. 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. And we used $R_{\epsilon=50\%}$, which is the value of inverse of false positive rate at true positive rate = 0.5. In Jet Physics setting, $1/FPR$ at $R_{\epsilon=50\%}$ represents background rejection at 50% signal efficiency. For measuring the robustness of the adversarial model we have plotted the conditional probability densities of the classification performance for the three different Z values.

6 Results and Analysis

6.1 Non-pileup data

In this section we will show the performance of Recursive neural network for classifying the jet streams with out any pileup. We are augmenting the results presented in [1]. So firstly we want to replicate the results generated by the [1], we have taken *kt* and *desc-pt* datasets for replicating the results. We also show the improvement in performance of classification model by introducing the Network-in-Network model. Table 1 shows the ROC AUC and $R_{\epsilon=50\%}$ values for different settings on both the datasets.

Table 1: Summary of Jet classification Performance

Input	Architectures	Model	ROC AUC	$R_{\epsilon=50\%}$
particles	kt	RNN	0.9009	58.75
particles	kt	RNN + 1x1 Convolution	0.9156	56.45
particles	desc-pt	RNN	0.9202	70.71
particles	desc-pt	RNN + 1x1 Convolution	0.9026	41.06

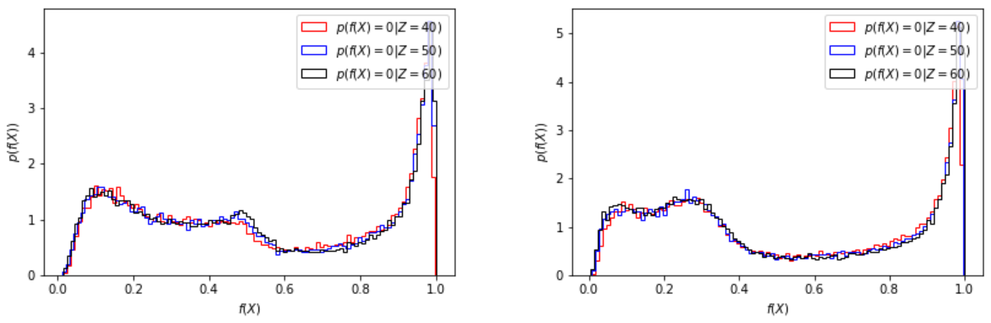
From the table ablove, we could see that, by adding three layers of 1×1 Convolution in place of using single ReLU as activation layer on k_t architecture, the newly proposed model shows a better performance on both ROC AUC score and background rejection at signal efficiency level at 50%.

However, When applying the same strategy on desc-p_T dataset, the proposed model did not perform as good as applying Recursive Neural Net alone. As desc-p_T was the best binary tree architecture when performed on particle level input in jet-level classification task when applying Simple Recursive Neural Net model, it has an AUC score of 0.9202 and an Inverse FPR at 50%: 70.71. Meanwhile Table1 shows that by adding 1×1 convolution layer, model performs far less well than simple RNN with AUC 0.9026 and Inverse FPR ar 50% 41.06.

One way to explain such situation might be the internal feature of desc-p_T architecture. Desc-p_T is a binary tree topology which builds jets(trees) by sorting p_T of particles with the largest p_T at the root of tree. Given its recursive way of building the tree, it is possible that this architecture fully amplifies the advantage of recursive neural net and keeps the sequential information inside the structure. However using convolution as activation layer, it is possible that convolution layer mixed up the sequential tree structure and leading to contamination of sequential informations. While k_t architecture is built by clustering algorithms, where convolution layer might be able to perform better with.

6.2 Pileup data

In this section we show the robustness of the classification model with pileup data. We have taken adversarial model on *kt* data with non pileup and pileup level 25. And we further experimented on the model performance using generated data on pileup levels of 40,50 an 60. As the generated data distribution using these three values are close to each other, these three pileup interaction level reflects a situation more close to reality settings.



(a) Without Adversarial Training

(b) With Adversarial Training

Figure 5: Comparison between Non-Adversarial Training and Adversarial Training($p(f(x)|z)$)

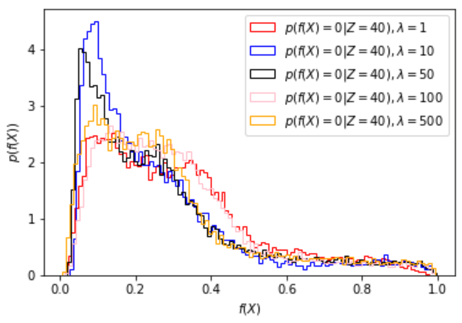
As shown in Figure 5, before adversarial training, the conditional probability density of the decision scores are discrepant, when $f(x) = 0, 1$. We can see there is an explicit dependency on level of pileup interactions, indicating our classifier is not pivotal. When the classifier is co-trained with adversarial loss, we can see that the resulting densities are almost indistinguishable, indicating only a small dependency on level of pileup interactions.

To examine the effects of λ on balancing the trade-off between optimality and pivotality, we varied this hyper parameter value from 1, 10, to 50, 100, and 500. Table 2 shows the values of the ROC and $R_{\epsilon=50\%}$.

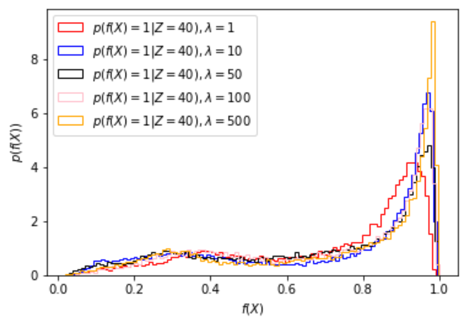
Table 2: Jet Classification Performance with pileup data

λ	Pileup level	ROC AUC	$R_{\epsilon=50\%}$
1	40	0.8777	32.50
1	50	0.8757	32.98
1	60	0.8697	33.14
10	40	0.8768	32.62
10	50	0.8751	34.35
10	60	0.8689	30.47
50	40	0.8677	22.39
50	50	0.8690	27.81
50	60	0.8666	29.39
100	40	0.8747	30.88
100	50	0.8732	33.31
100	60	0.8684	32.46
500	40	0.8795	33.09
500	50	0.8789	33.82
500	60	0.8736	34.99

Table 2 presents how the different values of λ affect the model performance. It is hard to tell the differences by just looking at the ROC AUC values. Thus we further calculated the variance of ROC AUC values among three pileup levels(40, 50, 60). The variance for $\lambda = 1$ is 0.1156×10^{-5}, the variance for $\lambda = 10$ is 1.1552×10^{-5}. As λ gets larger, this variance decreases, indicating that the ROC AUC values fluctuate less. When λ get even larger at 100, the variance is 0.7220×10^{-6} and when when λ is set at 500, the variance shrinks to 0.7220×10^{-6}. The further indicates that the classifier we have received after co-training is indeed more stable.



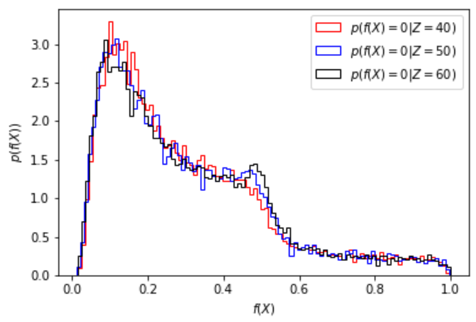
(a) Y=0



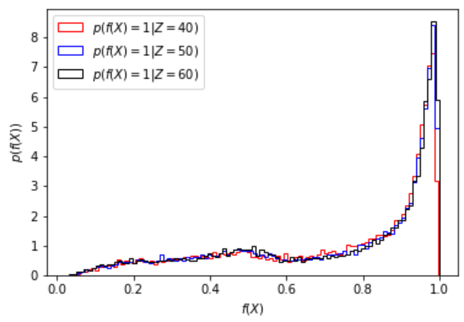
(b) Y=1

Figure 6: Variation of decision function of classification model with Adversarial loss w.r.t hyper parameter

Figure 6 show the variation of the decision function of the classification model for Y= 0 and Y= 1 with respect to the hyper parameter. From Figure 2 we can see that, when pileup interaction level is fixating at 40, as we increased λ values, the conditional probability density are getting flatter and flatter. This indicates that decision function we have produced is more towards pivotal than optimal.



(a) Y=0



(b) Y=1

Figure 7: Variation of decision function of classification model without Adversarial loss

Figure 7 show the variation of the decision function of the classifier model with out the adversarial network. As the figure shows there is lot of variation in the decision function with the variation of the nuisance parameter. However one observation is that large variations across λ values is more apparent when Y=0. When Y=1, the conditional probability density is seldom affected by the variant λ values.

7 Conclusions and Future Works

7.1 Conclusions

In this paper, we have closely studied Recursive Neural Nets applied in Jet Physics field of study. The application of Machine Learning domain knowledge has provided an accuracy of 0.92 in jet-level classification task.

Through our experiments, the RNN model with 1×1 Convolution layer we proposed displays different performances on jets clustering architecture k_t and jets sequential sorting recombination architecture desc-p_T, with which we could conclude that Convolution layer works well to improve model performances with non-sequential data structure, while contaminating sequential information in sequential data structure, which has a significant impact to prediction results.

In second half of our project, we further develop a Recursive Neural Net combining adversarial loss to act against our classifier aiming for a robust classifier independent of pivotal value z, that is pileup level in our problem. Our experiments show that when λ value as large as 500, we could see a decrease of ROC variance, showing a sign of robustness.

7.2 Future Works

We have seen improvement in classification model prediction using 3 layer 1×1 convolution over k_t dataset, we could apply same non-linearity to other datasets, but with parameterized 1×1 convolution layers. Root of the collision in the Jet stream is more complex than the rest of the sequence so instead of including that in the recurrent neural network, we could train it with different weights. And finally we could train the Adversarial network along with gated RNN's would further increase the accuracy and robustness of the model.

References

[1] G. Louppe, K. Cho, C Becot, and K. Cranmer. QCD-Aware Recursive Neural Networks for Jet Physics. *ArXiv e-prints*, February 2017.

[2] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[3] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to Pivot with Adversarial Networks. 2016.

[4] James Barnard, Edmund Noel Dawe, Matthew J. Dolan, and Nina Rajcic. Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks. *Phys. Rev.*, D95(1):014018, 2017.

[5] Luke de Oliveira, Michael Kagan, Lester Mackey, Benjamin Nachman, and Ariel Schwartzman. Jet-images — deep learning edition. *JHEP*, 07:069, 2016.

[6] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5, jul 2014.

[7] Matteo Cacciari and Gavin P. Salam. Pileup subtraction using jet areas. *Phys. Lett.*, B659:119–126, 2008.

[8] J Cogan.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[10] Taoli Cheng. Recursive Neural Networks in Quark/Gluon Tagging. 2017.

[11] Jason Gallicchio and Matthew D. Schwartz. Quark and gluon jet substructure. *Journal of High Energy Physics*, 2013(4):90, Apr 2013.

[12] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. The Anti-k(t) jet clustering algorithm. *JHEP*, 04:063, 2008.

[13] Stephen D. Ellis and Davison E. Soper. Successive combination jet algorithm for hadron collisions. *Phys. Rev.*, D48:3160–3166, 1993.

[14] M. Wobisch and T. Wengler. Hadronization corrections to jet cross-sections in deep inelastic scattering. In *Monte Carlo generators for HERA physics. Proceedings, Workshop, Hamburg, Germany, 1998-1999*, pages 270–279, 1998.

[15] Morad Aaboud et al. Identification and rejection of pile-up jets at high pseudorapidity with the ATLAS detector. *Eur. Phys. J.*, C77(9):580, 2017. [Erratum: Eur. Phys. J.C77,no.10,712(2017)].

[16] F. Hautmann, H. Jung, and H. Van Haevermaet. Treating jet correlations in high pile-up at hadron colliders. *Physics Letters B*, 754(Supplement C):260 – 263, 2016.

[17] Peter Berta, Martin Spousta, David W. Miller, and Rupert Leitner. Particle-level pileup subtraction for jets and jet shapes. *JHEP*, 06:092, 2014.

[18] James Barnard, Edmund Noel Dawe, Matthew J. Dolan, and Nina Rajcic. Parton shower uncertainties in jet substructure analyses with deep neural networks. *Phys. Rev. D*, 95:014018, Jan 2017.