Physics Inspired Deep Neural Networks for Top Quark Reconstruction

Kevin Greif[1,*] and *Kevin* Lannon[1,**]

[1]University of Notre Dame

Abstract. Deep neural networks (DNNs) have been applied to the fields of computer vision and natural language processing with great success in recent years. The success of these applications has hinged on the development of specialized DNN architectures that take advantage of specific characteristics of the problem to be solved, namely convolutional neural networks for computer vision and recurrent neural networks for natural language processing. This research explores whether a neural network architecture specific to the task of identifying t → Wb decays in particle collision data yields better performance than a generic, fully-connected DNN. Although applied here to resolved top quark decays, this approach is inspired by an DNN technique for tagging boosted top quarks, which consists of defining custom neural network layers known as the combination and Lorentz layers. These layers encode knowledge of relativistic kinematics applied to combinations of particles, and the output of these specialized layers can then be fed into a fully connected neural network to learn tasks such as classification. This research compares the performance of these physics inspired networks to that of a generic, fully-connected DNN, to see if there is any advantage in terms of classification performance, size of the network, or ease of training.

1 Introduction

The process of reconstructing top quarks from collision data is an important first step in studying rare top quark processes such as the $t\bar{t}H$ process. This task can grow difficult when looking to study a process that can produce as many as ten jets in a single event. Current techniques are able to achieve high accuracy when reconstructing a top anti-top quark pair by itself, but accuracy drops below 40 percent for events with extra jets [1]. Increasing the performance of top reconstruction methods would improve ability to measure rare processes at the LHC that produce many jets. This study aims to improve reconstruction techniques for resolved top quarks rather than focusing on the boosted regime, since much effort has already gone into reconstructing boosted top quarks. Instead, we focus on improving results for resolved top quark reconstruction, which could be useful in making measurements of the indirect effects of off-shell particles. A diagram of a top quark display is shown in Figure 1, where a top quark decays into a b quark and a W boson, which further decays into either a pair of leptons or a pair of quarks. This work focuses on reconstructing the latter decay

[*] e-mail: kgreif@nd.edu
[**] e-mail: klannon@nd.edu

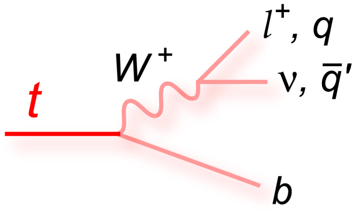

Figure 1. Decay of a top quark to a W boson and a b quark. The W boson can then decay to two other quarks, making a jet triplet [2].

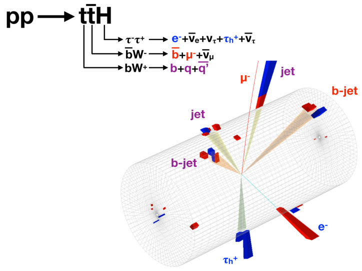
$pp \rightarrow t\bar{t}H$
$\tau^-\tau^+ \rightarrow e^-+\bar{\nu}_e+\nu_\tau+\tau_h^++\bar{\nu}_\tau$
$\bar{b}W^- \rightarrow \bar{b}+\mu^-+\bar{\nu}_\mu$
$bW^+ \rightarrow b+q+\bar{q}'$
μ^-
jet
b-jet
jet
b-jet
e^-
τ_h^+

Figure 2. Candidate event for the $t\bar{t}H$ process, in which one W boson decays to two quarks producing 4 total jets [3].

mode, in which a top quark decays to three other quarks and produces a jet triplet signature in a particle detector.

The top quark reconstruction method of interest uses "physics inspired" neural networks. A standard feed-forward deep neural network (DNN) takes a vector of inputs, in this case containing the momentum space 4-vectors of the three jets, and then feeds the inputs through several hidden layers until the information reaches an output layer. Often the network is configured to perform a classification task, meaning the output layer produces numbers between 0 and 1 which can be interpreted as a consistency score that rates the network's confidence that the inputs constitute an event of a certain type. "Physics inspired" neural networks perform these classification tasks using DNN's, except they prepend the feed-forward network with specialized layers that perform physics calculations on the raw collision data. The results of these calculations then serve as the feed-forward network's inputs. The intuition behind these specialized layers is that certain results of relativistic kinematics are well known to humans, but need to be learned by a DNN starting from random weights. Physics inspired layers allow existing knowledge of physics to be encoded into the structure of the network, bootstrapping the network's learning and hopefully increasing the network's performance.

Such networks were first developed for a different, but related task of tagging boosted top quarks by analyzing jet substructure [4]. This work produced a physics inspired network that could tag top quarks with performance comparable to the current state of the art image based top tagging DNN's [5], but had a flexible and simple design that could be easily adapted to other physics problems. In this paper, the physics inspired networks were adapted to approach the problem of top quark reconstruction from clustered jet triplets in the non-boosted regime. Figure 2 shows a candidate $t\bar{t}H$ event in which the pair of top quarks decays into two W bosons and two bottom quarks as in Figure 1. One of the W bosons then decays into a pair of quarks, making for four jets in the collision data. Correctly identifying the particles produced in this event requires recognizing the jet triplet pattern of a top quark decay against the background of the three other jet triplet combinations that do not represent a top decay. This classification task is a natural extension of previous work done with physics inspired neural networks.

2 Network Design

The physics inspired neural network used in this project, known as the Lorentz neural network (LNN), consists of three specialized layers for performing physics calculations. First, the combination layer takes sums of momentum space 4-vectors in order to assemble candidate 4-vectors for the W boson and top quark. The Lorentz layer takes this expanded set of 4-vectors and calculates relevant physics quantities, and the standardization layer transforms the Lorentz layer outputs to be on the same scale, making them appropriate inputs for the feed-forward network. To create a standard against which to judge the performance of the physics inspired network, a standard fully connected feed-forward network known as the Vanilla neural network (VNN) was also developed. This network had only the standardization layer of the LNN, since it was supposed to illustrate the performance of the physics calculations contained in the combination and Lorentz layers. This means that raw data was standardized and then fed into the fully connected layers. All networks used in this project were implemented, trained, and evaluated using the PyTorch machine learning library [6].

2.1 Combination Layer

Taking sums of momentum space 4-vectors involves simple matrix multiplication given by

$$p'_{jk} = p_{ji}C_{ik} = \begin{pmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \\ p_{41} & p_{42} & p_{43} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \tag{1}$$

where the x matrix is the momentum space jet triplet 4-vectors, and the x' matrix is the expanded set of three original jets, three possible jet doublets, and the total jet triplet. Each of these vectors is of interest because any single jet could represent a b quark, any pair of jets could represent a W boson, and the triplet of jets could represent the parent top quark.

2.2 Lorentz Layer

The Lorentz layer performs four types of calculations on each of the 4-vectors produced in the Combination layer. The first two are routine calculations of the 4-vector's invariant mass and transverse momentum, given by the equations

$$m^2 = p_\mu g^{\mu\nu} p_\nu \tag{2}$$

$$p_T = \sqrt{p_1^2 + p_2^2}. \tag{3}$$

In this notation, the 1 and 2 components of the four momentum are transverse, while the 3 component is along the beam axis. The invariant mass is a Lorentz invariant quantity, while the transverse momentum is invariant to boosts along the z-axis. This means both of these quantities are invariant for the boosts produced in collider experiments. Beyond the properties of single momentum space 4-vectors, it is also possible to define a Lorentz invariant distance between vectors given by the Minkowski distance. Since the combination layer produced a set of seven vectors, there are 21 possible Minkowski distances to compute for each event. Given that number, it proved useful to first multiply each Minkowski distance by a trainable weight that could be optimized during network training, then either take the sum or the minimum of the Minkowski distances produced for each particle. Given the signature of the metric, d^2 tends to be a negative quantity, so the minimization operation selects the largest distance between vectors. These two calculations are given by

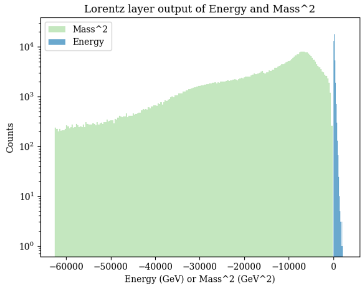

Figure 3. The raw output of the Lorentz layer calculations. The invariant calculation produces negative quantities orders of magnitude larger than the energy output.

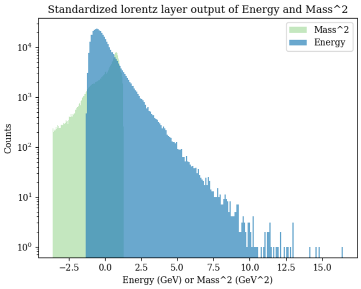

Figure 4. Output of the Lorentz layer calculations after standardization. The invariant mass and energy distributions are now on comparable scales.

$$d^2_{sum} = \sum_m w_{jm}(p_j - p_m)_\mu g^{\mu\nu}(p_j - p_m)_\nu \tag{4}$$

$$d^2_{min} = \min w_{jm}(p_j - p_m)_\mu g^{\mu\nu}(p_j - p_m)_\nu, \tag{5}$$

where w_{jm} are the elements of a matrix of trainable weights. Since the weights are drawn from a uniform distribution at the beginning of training, multiple instances of the d^2 calculations can produce different results and thus be useful to the network. Two of each such outputs were included in the final design for the Lorentz layer. This made for six total calculations performed in the Lorentz layer, in addition to the raw 4-vectors which were passed through the Lorentz layer to supply the network with full kinematic information.

2.3 Standardization Layer

Initial attempts to train the LNN with only the Combination and Lorentz layers failed for reasons that became obvious when examining the outputs of the Lorentz layer, shown in Figure 3. The invariant mass calculation produced negative values up to -62 TeV^2, orders of magnitude larger than the energy scale of the individual 4-vectors. Such large Lorentz layer outputs caused the softmax output layer described in Section 2.4 to saturate, producing network outputs that were extreme values of nearly 0 or 1. Since network training relies on computing gradients, such extreme outputs made the gradients vanish and no training progress was made. The solution to this problem was to standardize the Lorentz layer outputs so that they were on equivalent and reasonable scales. This involved including a third specialized layer in the network that would perform a simple transformation on each of the Lorentz layer outputs, given by

$$p' = \frac{p - \mu_p}{\sigma_p}, \tag{6}$$

where μ_p is the mean of the raw output and σ_p is the standard deviation. The transformed invariant mass and energy distributions are shown in Figure 4. With the transformed Lorentz layer outputs, the network was able to train reliably.

2.4 Feed-Forward Network

The feed forward portion of the network takes a total of 70 inputs from the physics inspired layers. It consists of 4 hidden layers, each with a ReLU activation and 100, 75, 50, and 25 nodes respectively. The decreasing size of the hidden layers is intended to force the network to extract higher and higher level information from the inputs as they are passed through the layers. The output layer consists of 2 nodes with a softmax activation function, which requires that the sum of the two outputs be equal to 1. This allows the network output to be interpreted as a score that rates the network's confidence the jet triplet is matched to a top quark decay. A cutoff can then be imposed that classifies the triplets into signal and background.

The Vanilla neural network consisted of only the standardization layer and the feed forward pieces of the Lorentz neural network, but to conduct a fair comparison between the Vanilla and Lorentz neural networks, extra trainable weights were added to the VNN's fully connected layers to make up for the trainable weights included in the LNN's Lorentz layer and the information encoded in the physics calculations. However, the number of hidden layers and the softmax output layer of the VNN was identical to the LNN. Each hidden layer again used a ReLU activation function and had 120, 90, 60, and 30 nodes respectively.

3 Network Training

All data used to train and evaluate the network were drawn from the top pair production dataset in the semi-leptonic decay mode. All top quarks were decayed and hadronized to final state particles, which were then clustered into jets using the anti-k_T jet clustering algorithm. This was done with a radius parameter of 0.4. The data also had no applied detector simulation. The data was partitioned into training, testing, and validation sets, where the training set consisted of 2.1 million events and the testing and validation sets each contained 264 thousand events. All data sets contained equal numbers of jet triplets matched to top quarks and unmatched jet triplets.

All networks were trained using a batch size of 200 jet triplets and the Adam optimizer with an initial learning rate of 1×10^{-3}. Network training was allowed to continue until the network saw no decrease in loss averaged over the entire testing set for 5 consecutive epochs. In practice, this limit was usually reached after 20 to 30 epochs. When the training was run on GPU machines provided by the Notre Dame Center for Research computing, this training took on average 66 minutes for the Lorentz neural network and 59 minutes for the Vanilla network. This difference in training time is a result of the extra calculations included in the LNN, but such differences in training time are negligible for practical purposes.

4 Results

Performance results for the Lorentz and Vanilla networks are shown in Table 1, with the accompanying ROC curves for the AUC measurement shown in Figure 5. These results are averaged over 10 training runs for each of the networks, and the error bars are the sample standard deviation of the 10 values. Both the area under the receiving operator characteristic curve (AUC) and the accuracy (ACC) performance measurements show a small but significant increase when including the physics inspired layers. This is clear evidence that the layers provide some benefit to the network that is not being learned by the trainable weights in the Vanilla network.

Histograms of a sample network's output for the Lorentz and Vanilla structure are shown in Figure 6 and 7. The most prominent indicator of the increased performance of the Lorentz network is the green background curve in the total output histograms, which is much more

Network	AUC	ACC	Epochs	Training Time (min)
Lorentz NN	0.8540 ± 0.0005	0.7714 ± 0.0024	29	66
Vanilla NN	0.8263 ± 0.0006	0.7477 ± 0.0006	22	59

Table 1. Summary of Lorentz and Vanilla network performances. All values are averaged across 10 training runs, and error bars are the sample standard deviation of the 10 values. AUC is area under the receiving operator characteristic curve, and ACC is the accuracy measurement with a threshold of 0.5.

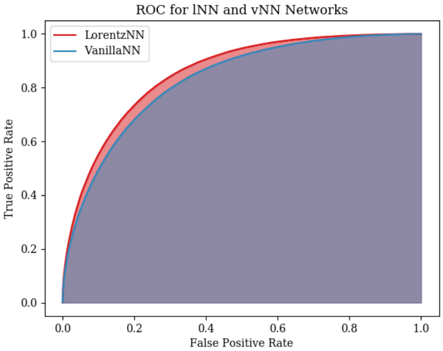

Figure 5. ROC curves for the Vanilla and Lorentz networks. The red curve shows a small but clear performance gain for the Lorentz network.

sharply peaked near low output probability in the Lorentz network histogram. This suggests that the physics inspired layers boost the network's ability to discriminate background. This is especially apparent when plotting only the events outside of the top mass peak, meaning the invariant mass of the sum of the jet vectors is more than 25 GeV from the top mass. Plots of network output for events inside and outside the top peak are shown in the lower histograms in Figure 6 and 7. Most of the Lorentz network's gains over the Vanilla network come from more properly identifying background outside of the top mass peak. The other feature of the LNN output is the spike in the number of events with output probability around 0.68. The reasons for this spike are not entirely clear, but some experimentation showed that

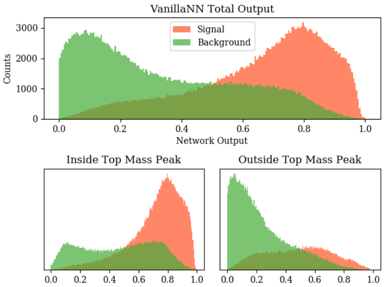

Figure 6. Output histograms for a sample Vanilla network. Lower histograms display events inside and outside of the top mass peak.

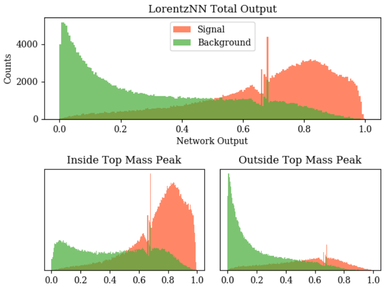

Figure 7. Output histograms for a sample Lorentz network. The green background curve is more sharply peaked at low output compared to the Vanilla network output.

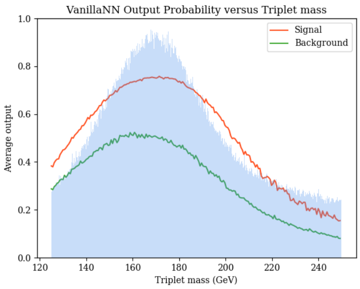

Figure 8. Average VNN output plotted against triplet mass. Blue histogram shows raw triplet mass distribution.

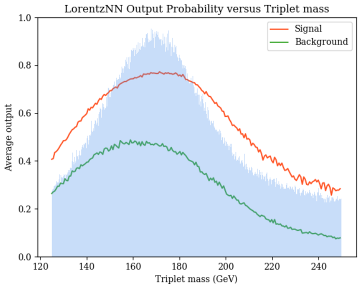

Figure 9. Average LNN output plotted against triplet mass. Performance is most increased in high energy tail of mass distribution.

the spikes disappeared when the network used sigmoid activation functions. This suggests that the spikes are a result of the ReLU activation functions acting on some subset of the Lorentz layer outputs. More experimentation would be required to pin down the source of these spikes, but network performance remained the same when sigmoid activations were used and the spikes were eliminated, so this feature of the output seems to have little effect on network performance.

In addition to examining raw network output, it also proved useful to examine network output plotted against some physics variable being calculated in the Lorentz layer. This provided insight into how the Lorentz network was drawing out additional information through its hardwired physics calculations. Figure 8 and Figure 9 plot the average output of a network against the invariant mass of the sum of the jet vectors. These plots are two dimensional histograms (sorting events by the invariant mass of the jet triplet and the network output) that has been averaged over the network output axis in order to show how network output varies with the triplet mass. The blue histogram behind the curves is just the triplet mass distribution plotted to show the location of the top mass peak. The signal and background scores assigned by the Vanilla network have a clear dependence on the triplet mass, which shows that the standard fully connected network architecture was able to learn to use the invariant mass distribution to distinguish top quarks. However, the network struggled to distinguish signal and background for events both inside of the top mass peak and in the high energy tail of the distribution. The Lorentz network, which was given the triplet invariant mass without any training, learned to distinguish signal and background slightly better than the Vanilla network. This difference is particularly striking in the high energy tail of the distribution, where the orange signal score in Figure 9 is significantly higher than the corresponding curve in Figure 8.

5 Conclusion

The physics inspired Lorentz neural network was shown to provide a small performance boost on a top quark reconstruction task compared to a standard fully connected, feed-forward deep neural network. While this performance gain was modest, the differences in network output plotted against physics variables shows that there are important differences in the ways that networks learn when encoded with existing knowledge of the problem. Both the Lorentz and Vanilla networks learned to distinguish events using the triplet mass distribution, but the Lorentz network was able to extract information that the Vanilla network was not, since it was forced down the correct training path by the physics inspired layers. There is no reason all of the calculations encoded in the physics inspired layers could not be learned by a standard fully-connected network, but these layers fix calculations that are known to be of use, placing the network in the right ballpark of the correct categorization of events and letting the training explore further improvements within this more limited solution space. Although the final improvement to network performance is modest, encoding existing physics knowledge into a neural network did cause improvement, and further optimization of physics inspired layers could provide more robust performance gains. There are a number of possible further optimizations to the physics inspired layers. A natural extension is to add physical information to the data set to provide richer information for the network to analyze. Example extensions include the jet charge or b-quark identification information [7]. Additionally, further physics knowledge could be encoded into the physics inspired layers. For example a deep set network could be used to enforce permutation invariance. All of these ideas offer a possibility of turning the modest performance gain of this work into a new and more powerful top quark reconstruction technique.

References

[1] A.M. Sirunyan et al. (CMS), Phys. Rev. **D97**, 112003 (2018), 1803.08856

[2] A. Heinson, *Useful diagrams of top signals and backgrounds*, https://www-d0.fnal.gov/

[3] CMS Collaboration (2018), CMS Collection, https://cds.cern.ch/record/2621446

[4] A. Butter, G. Kasieczka, T. Plehn, M. Russell, SciPost Physics **5** (2018)

[5] A. Butter et al., SciPost Phys. **7**, 014 (2019), 1902.09914

[6] A. Paszke et al., in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc, 2019), pp. 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[7] S. Chatrchyan et al. (CMS), JINST **8**, P04013 (2013), 1211.4462