ATLAS EXPERIMENT

25 August 2021

Further Improvements to AngryTops: A machine learning package for the reconstruction of top-quark pair decay kinematics

Darren Zeming Chan & Maggie Fen Wang, University of Toronto

May 2021 - August 2021

Summary

Studies of the top quark provide unique insights into the Standard Model due to its large mass. However, the kinematics of $t\bar{t}$ decays is difficult to reconstruct due to the complexity of these events and limited detector resolution. Neural networks are thought to perform as well as state-of-the-art statistical algorithms for reconstruction purposes.

Our group has developed a machine learning package called AngryTops, a BLSTM neural network that reconstructs $t\bar{t}$ decay pair kinematics resulting from 13 TeV pp collisions. Although the package successfully reconstructs the kinematic variable distributions, we lack a systematic way to evaluate the network's performance on individual events. We implement improvements to better characterize the network's performance. We also introduce an algorithm that matches the observed leptons and jets with particles arising from the $t\bar{t}$ decay (truth particles). The variables used for matching are then used to filter the training dataset, retaining only events that the network should be able to reconstruct well.

We train the network on the filtered dataset and evaluate how the network performs compared to when it is trained on the original sample. We also develop a χ^2 metric and corresponding p-value test to assess predictions for each event. Further developments including data augmentation and fine-tuning parameters will be investigated using the χ^2 test and matching algorithm as performance metrics.

Contents

1 Introduction and Background Theory

The goal of this project is to improve a machine learning package known as AngryTops, which reconstructs $t\bar{t}$ decay pair kinematics. We begin with a brief discussion of $t\bar{t}$ decay kinematics, previous work done on AngryTops, and the role of machine learning in modelling such kinematics.

1.1 Top Quark Pair Kinematics

The top quark is the most massive fundamental particle. With a mass of 172.5 GeV, it is useful a as probe to search for new massive particles [8]. Currently, the Large Hadron Collider (LHC) produces $t\bar{t}$ pairs via pp collisions at 13 TeV. The top quarks decay into two b-quarks and two W-bosons, which further decay into and are detected as leptons and jets, which are collimated sprays of hadrons to which we can assign a momentum and direction [5]. The kinematic reconstruction of such a decay involves identifying the daughter jets and leptons of the W-bosons and b-quarks. However, the number of decay products and distortions due to limited detector resolution make this process challenging to study. Our overall goal is to improve the ability of a neural network to reconstruct the kinematics of the top quarks, b-quarks, and W-bosons.

This project focuses on only one decay channel. We assume that each event consists of one fully hadronic and one semi-leptonic top quark decay. In this decay process, both top quarks decay into a b-quark and a W-boson. Each b-quark decays into a single jet. The W-boson in the hadronic decay produces two jets, while the semi-leptonically decaying W-boson produces a muon and a neutrino. The process that we model is shown in the Feynman diagram in Figure 1.

We refer to the b-quark, W-boson, and t-quark in the hadronic decay as the hadronic b, hadronic W, and hadronic t respectively. Similarly, the particles in the semi-leptonic decay are referred to as the semi-leptonic b, leptonic W, and leptonic t.

During subsequent analysis, each event is characterized by three types of data:

1. "Truth": Truth data consists of the momenta of the six final state decay products. It is what the model tries to predict based on the input. To create truth data, the MadGraph5 Monte Carlo event generator [1] was used to calculate the process that produces top-quark pairs in pp collisions at 13 TeV.

2. "Observed": Since the input to the network represents how events are observed through the LHC detectors, it is referred to as "observed data". It was created by using PYTHIA8 [6] and DELPHES3 [2] to model parton showering and detector distortions on top of the Monte Carlo events.

3. "Predicted/Fitted": The neural network's output, consisting of its predictions of

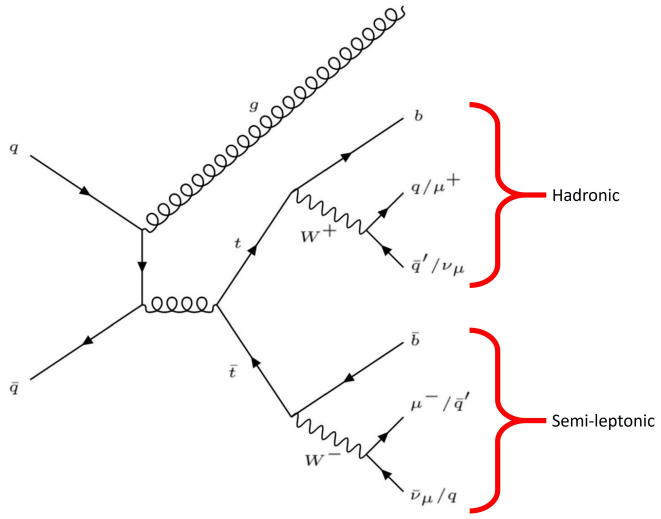


Figure 1: Hadronic and semi-leptonic decays of $t\bar{t}$ pairs from 13 TeV pp collisions [4]. This is only one of several diagrams that contribute to $t\bar{t}$ production.

the decay products' momenta. It is compared against truth data to determine how well the network performs.

Figure 2a depicts the observed (input) data for each event. Figure 2b shows the truth and predicted (output) matrix for each event.

$$\begin{bmatrix} p_x^{\mu} & p_x^{j,1} & p_x^{j,2} & p_x^{j,3} & p_x^{j,4} & p_x^{j,5} \\ p_y^{\mu} & p_y^{j,1} & p_y^{j,2} & p_y^{j,3} & p_y^{j,4} & p_y^{j,5} \\ p_z^{\mu} & p_y^{j,1} & p_z^{j,2} & p_z^{j,3} & p_z^{j,4} & p_z^{j,5} \\ T_0^{\mu} & E^{j,1} & E^{j,2} & E^{j,3} & E^{j,4} & E^{j,5} \\ E_T^{miss} & M^{j,1} & M^{j,2} & M^{j,3} & M^{j,4} & M^{j,5} \\ E_{\phi}^{miss} & B^{j,1} & B^{j,2} & B^{j,3} & B^{j,4} & B^{j,5} \end{bmatrix} \quad \begin{bmatrix} p_x^{b,had} & p_y^{b,had} & p_z^{b,had} \\ p_x^{b,lep} & p_y^{b,lep} & p_z^{b,lep} \\ p_x^{W,had} & p_y^{W,had} & p_z^{W,had} \\ p_x^{W,lep} & p_y^{W,lep} & p_z^{W,lep} \\ p_x^{t,had} & p_y^{t,had} & p_z^{t,had} \\ p_x^{t,lep} & p_y^{t,lep} & p_z^{t,lep} \end{bmatrix}$$

(a) Input matrix (b) Output matrix

Figure 2: Input and output matrices to the network for each event.

For the input matrix (Figure 2a):

- Elements (1,1), (2,1), and (3,1) are the observed muon momentum components in Cartesian space. Used for leptonic W calculations.
- Element (4,1) is the muon time-of-flight, and is not used in our analysis.
- Elements (5,1) and (6,1) are the magnitude and azimuthal angle, respectively, of the missing transverse energy vector. The missing transverse energy is the energy

that is carried away by a neutrino, and so is not directly measured by the detector. These variables are used to calculate properties of the leptonic W.

- Block (1-5, 2-6) represents variables associated with the observed jets, namely their three momentum components, their energy, and their mass. For this project, we consider only the four or five jets with the highest energies. If we are dealing with a four-jet event, the fifth column is filled with zeros and later removed from the analysis. These are used for b-quark and hadronic W calculations.
- Block (6, 2-6) represents the b-tagging state of each jet. The b-tagging state of a jet refers to whether or not it has already been pre-assigned to a b-quark. This process has an efficiency of 80%. The b-tagging state takes on values of 1 or 0, where 1 indicates that the jet likely originates from a b-quark. The b-tagging state is used to identify b-jets, and in subsequent sections, for hadronic W calculations.

For the output matrix (Figure 2b):

- Each row contains the p_x, p_y, and p_z of one of the six final state particles: hadronic and leptonic ts, bs, and Ws.

We use the network's output to calculate eight kinematic variables for each particle:

1. Energy (E)
2. Mass (m)
3. Momentum in the x-direction (p_x)
4. Momentum in the y-direction (p_y)
5. Momentum in the z-direction along beam axis (p_z)
6. Transverse momentum ($p_T = \sqrt{p_x + p_y}$)
7. Azimuthal angle ($\phi = \arctan(\frac{p_y}{p_x})$)
8. Rapidity ($y = \frac{1}{2} ln \frac{E+p_z c}{E-p_z c}$)

1.2 Previous Work

There have been previous attempts to develop and improve a neural network's ability to reconstruct $t\bar{t}$ decay kinematics. Syed et al. [7] introduced AngryTops, a machine learning package designed for this reconstruction problem. During preliminary comparisons, Syed et al. [7] also found that AngryTops is competitive with standard statistical reconstruction

algorithms such as χ^2-FIT, while offering improved flexibility in terms of input. Across CNNs, LSTMs, FFNNs, and BLSTMs networks, the BLSTMs performed the best [7].

The investigation by Tan et al. [3] into a new RNN architecture and other training parameters found that the RNN performed worst than existing architectures and needed further optimization. Between different data representations, scaling functions, and epochs, some variables showed improvements while predictions for others were worse. As an additional measure of network performance, Tan et al. [3] also introduced a matching algorithm which identifies the closest daughter jets to each b-quark and W-boson.

Finally, the authors of [4] continued the comparison of different architectures, finding that the CNN performs best and trains faster. [4] also modified the jet matching algorithm to categorize events as "fully reconstructable", "partially reconstructable", and "unreconstructable". For the Predicted vs. Truth comparison, the modified algorithm assigned these criteria to events based on whether η-ϕ distances are within tolerances determined from the correlation plots [4]. For the Truth vs. Observed comparison, the algorithm assigned the criteria based on the number of jets (out of four) that are within other tolerances of the Truth momenta [4].

1.3 Machine Learning Techniques

The neural network architecture used for this project was a Bi-Directional Long Short-Term Memory (BLSTM) network. Data sent through a BLSTM network passes through two Long Short-Term Memory (LSTM) layers, followed by an activation layer [9] which translates the network weight to the final outputs. The main advantage of a BLSTM is that it has access to information from both the past and the future inputs by using both LSTMs [4]. For this reason, it is most commonly used for language applications, where the context of input words is relevant. Previous work found that the BLSTM architecture performs the best [7], albeit requiring a significantly longer training time compared to the CNN [4].

Our sample consists of 9.5 million events, of which 90% were used for training, and 10% were reserved for testing. During training, a further 10% of the training dataset is randomly used as validation.

The loss function is a measure of how how well the network predicts events over the entire dataset, and is calculated for each training period (epoch). Subsequent investigations use the mean squared error (MSE) loss function, calculated as

$$M.S.E. = \frac{1}{N}\sum_{n=1}^{N}\sum_{W,b,t}(p^t_{x,n} - p^p_{x,n})^2 + (p^t_{y,n} - p^p_{y,n})^2 + (p^t_{z,n} - p^p_{z,n})^2 \tag{1}$$

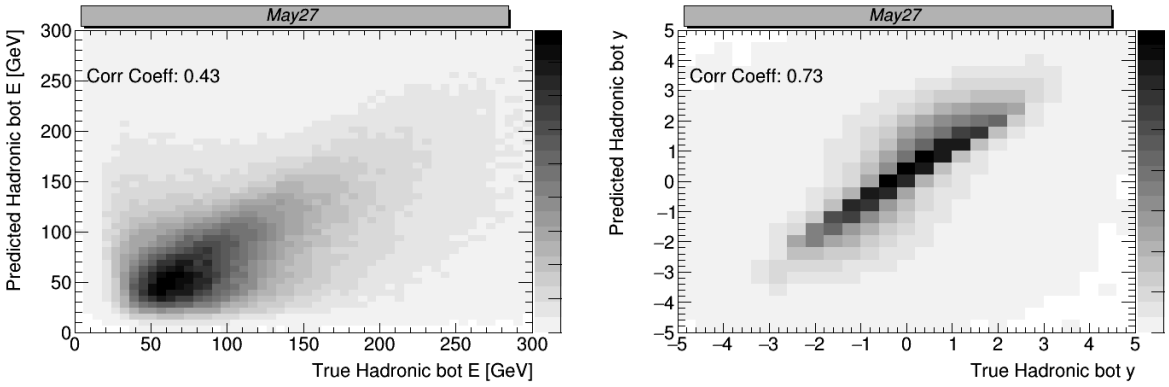
where N is the number of events in the dataset. $p^t_{x,n}$, $p^t_{y,n}$, and $p^t_{z,n}$ are the n$^{\text{th}}$ truth momenta values for the hadronic and semi-leptonic Ws, bs, and ts, and $p^p_{x,n}$, $p^p_{y,n}$, and $p^p_{z,n}$ are the n$^{\text{th}}$ predicted momenta values.

2 Characterization of and Improvements to the Network Itself

We evaluate the performance of the network by comparing its output ("predicted") against the MC-generated truth data. There are multiple metrics that we use to quantify the "distance" between these datasets.

2.1 Correlation Plots

The first metric that we use is correlation plots. These consist of heat maps with both axes divided into bins. Each bin is coloured based on the number of events inside it, i.e. a 2D-histogram. We generate a set of correlation plots for all kinematic variables of the six particles (Section 8) but mass. We generate this set for all six particles. The Pearson correlation coefficient is calculated on a bin-by-bin basis. The correlation coefficient takes on values from -1 through 1. A higher correlation indicates a better linear match between the predicted and true distributions. For example, Figure 3a has a lower correlation, and therefore appears to have a less linear relationship compared to Figure 3b.



(a) Energy (b) η

Figure 3: Sample correlation plots between true and predicted values for the hadronic b-quark after the network was trained for 50 epochs.

2.2 Optimal Training Length

During training, the number of times the model iterates through the entire dataset and the resulting length of training is dictated by the number of epochs. Usually, the model's performance improves as it is trained for longer (i.e. for a higher number of epochs). However, as the length of training increases, the model may become biased to details in the training data and be unable to generalize to new data, a phenomenon known as

overfitting. Graphically, overfitting is seen when the validation loss plateaus or begins to increase while the training loss continues to decrease.

The network requires two hours to train per epoch. To avoid overfitting and unnecessarily long training times, early stopping methods are used. In AngryTops, a patience value is specified, whereby the training ends if the network's loss value does not increase for a specified number of epochs.

Previously, the patience of the network was set to 3, causing the network to stop before the specified number of epochs. Our goal was to determine the optimal number of epochs to train the network for, and the point at which overfitting occurs, irrespective of the patience value. To do so, we set the patience to a large number of 100 so that early stopping does not occur and compare the network's performance after training for 9, 15, 36, and 50 epochs.

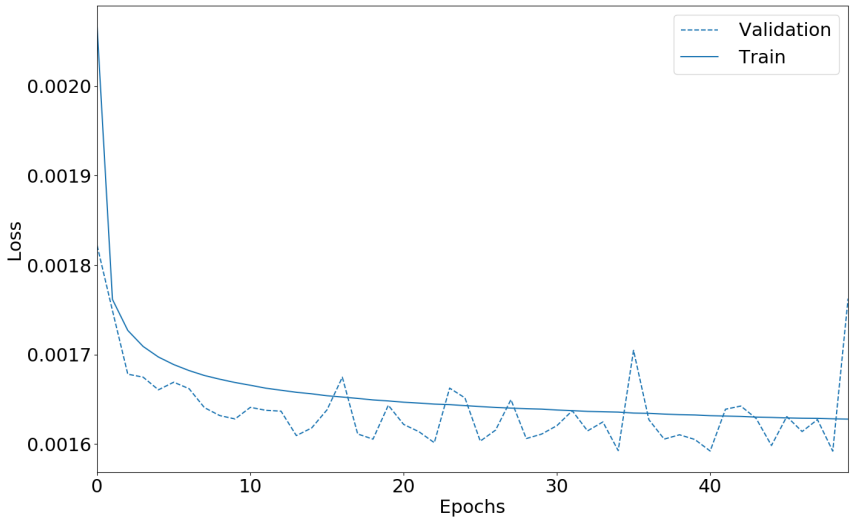


Figure 4: History of the loss function after training the network for 50 epochs.

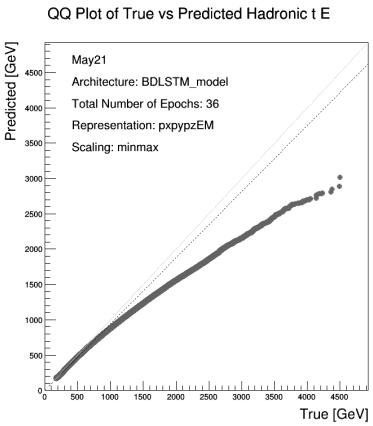
Correlation coefficients as described in Section 2.1 were used to examine improvements in specific variables. We found that between 9 and 36 epochs, the energy of each decay product showed some improvement, while most other variables did not improve significantly. Looking at the training history for the 50 epoch run in Figure 4, overfitting occurred at around 40-45 epochs. Based on these observations, we determined the optimal training length of the network to be around 36 epochs.

2.3 Quantile-Quantile Plots

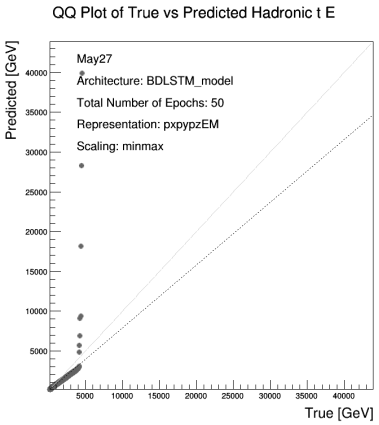
Quantile-quantile (qq) plots are a visual way of comparing two distributions, where the quantiles of the sample distribution are plotted against quantiles of the theoretical dis-

tribution. If the distributions are the same, we expect the qq plot to follow the $y = x$ line.

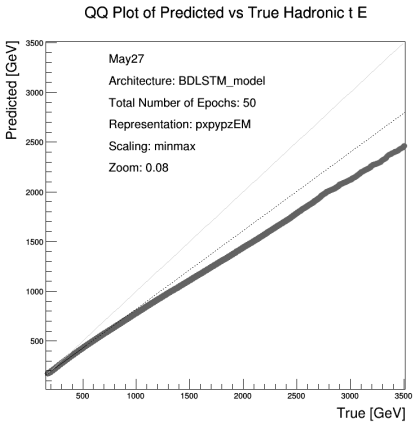
For the plots in Figure 5, we order the predicted and truth data in increasing order, and plot the predicted events against the truth. For reference, the dotted line passes through the 25th and 75th quantiles, while the light gray line is $y = x$. Most events have low energies, as the dotted line deviates from the events at higher energies. Visually, the line of best fit is much closer to $y = x$ for events lower than the 75th quantile. This suggests that high-energy events are more susceptible to the network's biases and to overfitting, likely because there is less data for the network to learn from. The effect of overfitting is especially noticeable when comparing the 36 epoch to the 50 epoch run, where the network's grossly over-predicts the energies of the events around 4500 GeV.



(a) Hadronic t, 36 epochs



(b) Hadronic t, 50 epochs



(c) Zoom of hadronic t, 50 epochs

Figure 5: Quantile-quantile plots for hadronic t energy after training for 36 and 50 epochs.

2.4 Checks on the Network Prediction of Kinematics

In both decay channels studied (hadronic and semi-leptonic in Figure 1), four-momentum should be conserved between the final state products since the t-quarks decay into a W-boson and b-quark. Although the network predictions are not constrained by existing kinematic laws, we expect to be able to perform some form of Lorentz computations using the network's output of the W-boson and b-quark four-momenta to get that of the t-quark.

To evaluate if the network can learn that energy and momentum should be conserved, we summed the momentum vectors of the appropriate b-quark and W boson for each event. We also summed the energies separately. The summed momenta and energies were compared to the corresponding top quark variables in three ways:

1. The W+b energy was compared to the t energy. The t energy has a lower bound of the t quark invariant mass.
2. The angle between $\vec{p}_W + \vec{p}_b$ and $\vec{p}_t$. This is calculated using the following equation:

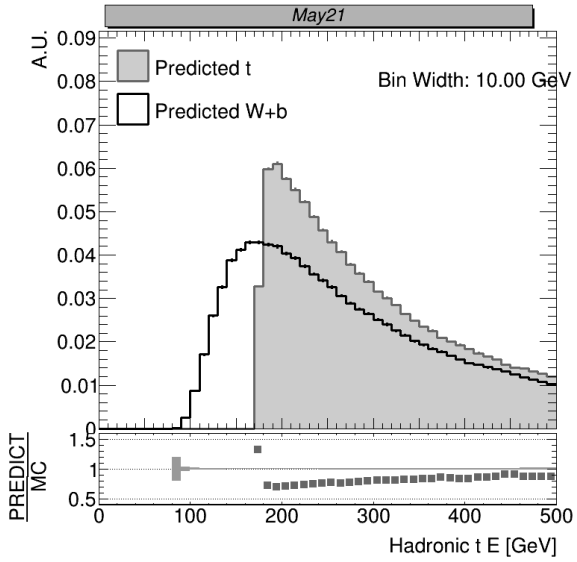
$$\theta = \arccos\left(\frac{(\vec{p}_W + \vec{p}_b) \cdot \vec{p}_{top}}{\|(\vec{p}_W + \vec{p}_b)\|^2 \|\vec{p}_{top}\|^2}\right) \tag{2}$$

   This angle should be close to 0 if the two momentum vectors overlap.
3. The scalar projection of $\vec{p}_W + \vec{p}_b$ onto $\vec{p}_t$. This is calculated using the following equation:

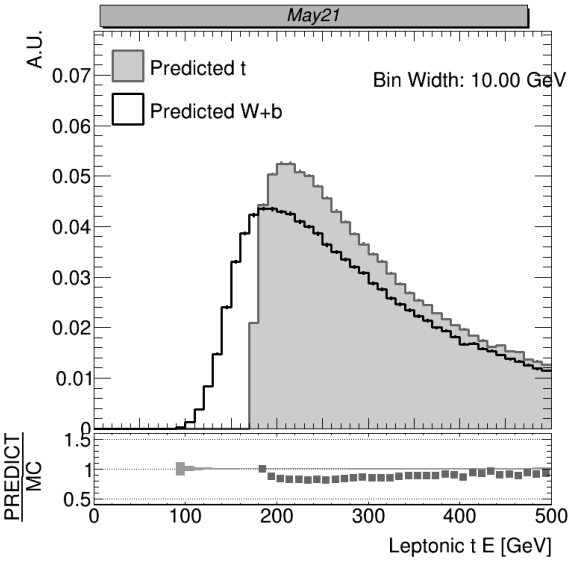
$$projection = \frac{(\vec{p}_W + \vec{p}_b) \cdot \vec{p}_{top}}{\|\vec{p}_{top}\|^2} \tag{3}$$

The distributions of the three quantities are shown in Figures 6, 7, and 9.

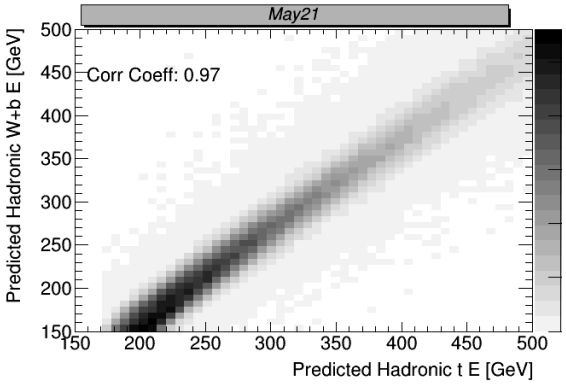
Across all variables, the semi-leptonic decay channel was better predicted. In the histograms, the predicted black outline more closely matches the shaded grey region, and the difference plots have smaller full-width half maxima (FWHM). This is likely because reconstructing the semi-leptonic decay channel is not a combinatorial problem and is relatively straightforward, as examined in section 3. Nevertheless, the hadronic decay channel was still well-predicted, with high correlation coefficients between the energy variables, and difference plots of the angle (Figure 8) and scalar projection (Figure 10) having low FWHM.



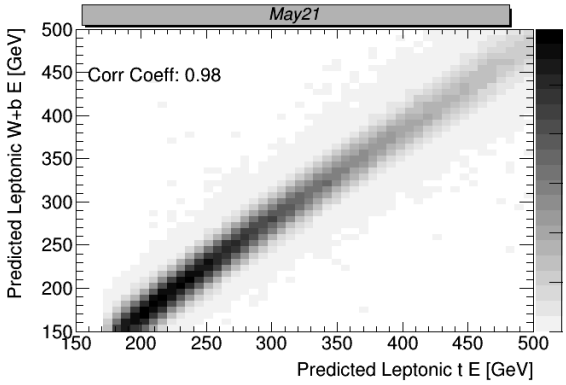
(a) Histogram of hadronic $E_W + E_b$ and E_t



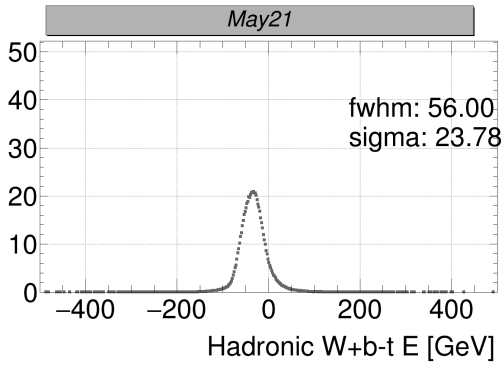
(b) Histogram of leptonic $E_W + E_b$ and E_t



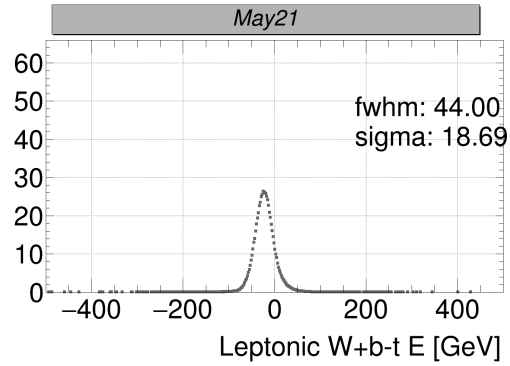
(c) Corresponding correlation plot for Figure 6a.



(d) Corresponding correlation plot for Figure 6b.

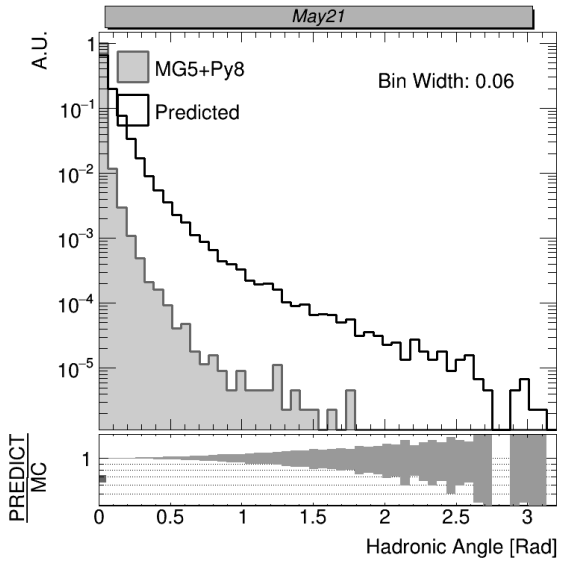


(e) Corresponding difference plot for Figure 6a of $(E_W + E_b) - E_t$.

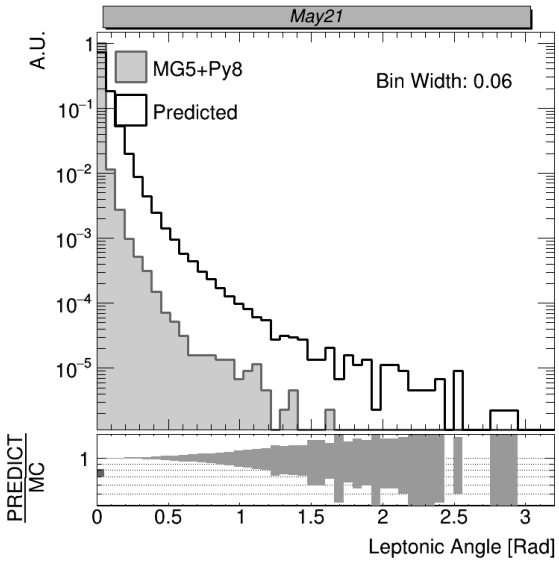


(f) Corresponding difference plot for Figure 6b of $(E_W + E_b) - E_t$.

Figure 6: $E_W + E_b$ and E_t for the hadronic and leptonic decays. In the histograms, the grey area represents E_t, while the black outline shows $E_W + E_b$.

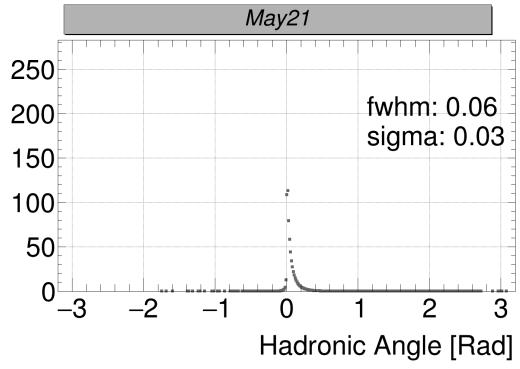


(a) Hadronic

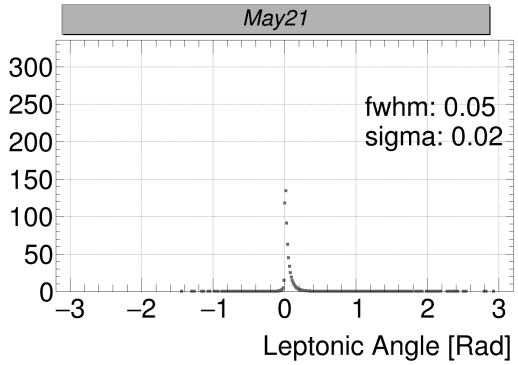


(b) Semi-leptonic

Figure 7: Angle Between $\vec{p}_W + \vec{p}_b$ and $\vec{p}_t$. The grey area shows the angle calculated based on the truth momenta, and the black outline shows calculations based on the predicted values.

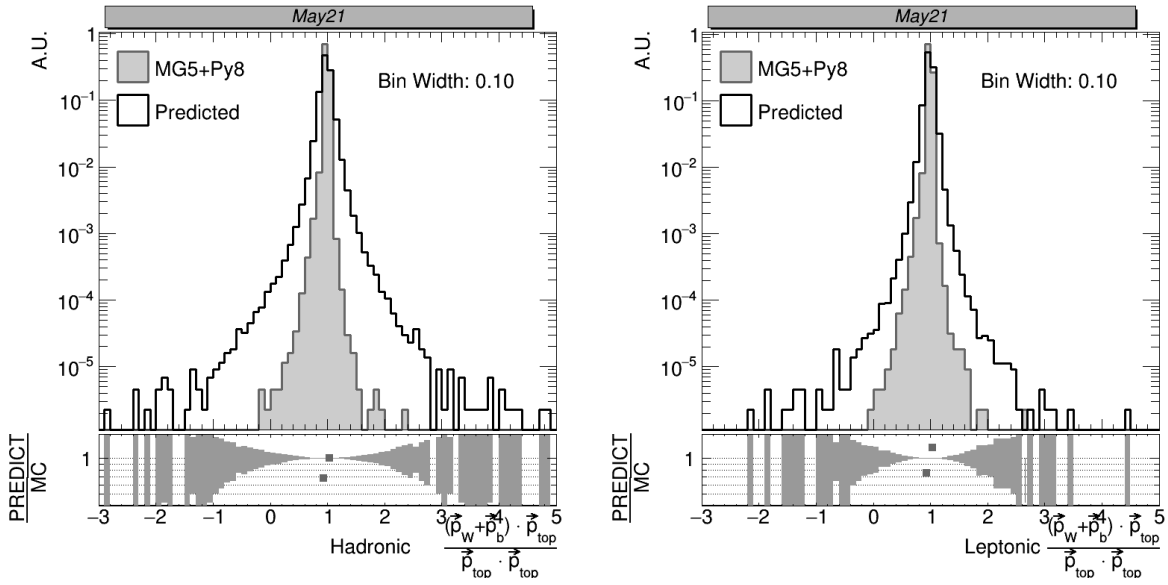


(a) Hadronic



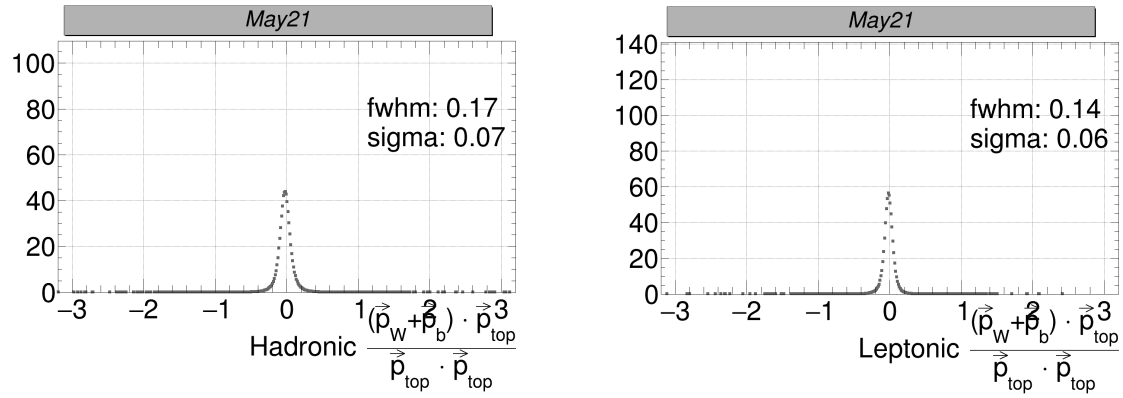
(b) Semi-leptonic

Figure 8: Difference in angle between $\vec{p}_W + \vec{p}_b$ and $\vec{p}_t$, based on fitted angle - truth angle.



(a) Hadronic (b) Semi-leptonic

Figure 9: Scalar projection of $\vec{p}_W + \vec{p}_b$ onto $\vec{p}_t$. The grey area shows the angle calculated based on the truth momenta, and the black outline shows calculations based on the predicted values.



(a) Hadronic (b) Semi-leptonic

Figure 10: Difference plots of the scalar projection of $\vec{p}_W + \vec{p}_b$ onto $\vec{p}_t$, based on fitted projection - truth projection.

3 Jet Matching

Our initial sample consists of around 9.5 million events. The observed data for each event consists of the momenta of four or five jets, the muon momentum, and the missing transverse energy. Due to detector distortions, not all jets can be easily matched to a particle, and not all jets originating from a decay product may be included as part of the event information, making it difficult to reconstruct the kinematics of some events. We expect the network to better predict events whose jets' kinematic variables are more closely matched to truth momenta.

Previous work matched **observed** jets to **truth** momenta for the b-quarks and the hadronic W by finding the minimum $\eta - \phi$ among all sums of two jets' momenta [3]. The work was then extended to consider $\eta - \phi$ distances between the **truth** momenta and the network's **predicted** momenta, and classify how easily an event can be reconstructed based on the number of matched jets in a given event [4].

We build on the previous jet matching script to introduce invariant mass and p_T difference as additional matching criteria and improve the hadronic W matching algorithm. We also introduce a separate algorithm that calculates the observed leptonic W momentum *without* matching jets. Finally, we investigate the network's performance when trained on a subset of our initial sample consisting of only high-quality events.

The invariant mass, $\eta - \phi$ distance, and p_T difference are calculated as:

$$m_{obs} = \sqrt{E_{obs}^2 - ||\vec{p}_{obs}||^2} \tag{4}$$

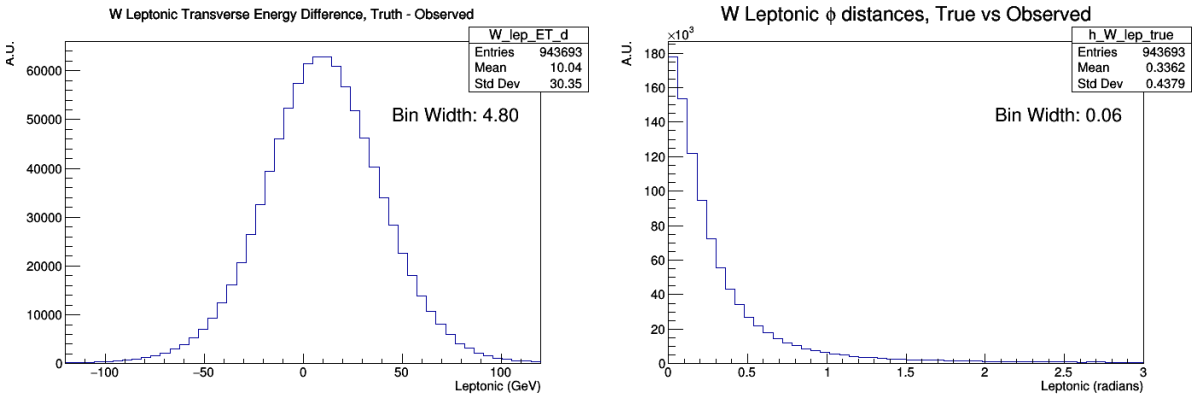
$$R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{5}$$

$$\Delta p_T = p_{T,true} - p_{T,obs} \tag{6}$$

where $\Delta\eta = \eta_{true} - \eta_{obs}$ and $\Delta\phi = \phi_{true} - \phi_{obs}$.

3.1 Semi-Leptonic W Bosons

The W-boson from the semi-leptonic top quark decays into a muon and muon anti-neutrino, which are not observed as jets. To reconstruct the observed leptonic W p_T, we first assume that the muon and neutrino are massless. Then, by conservation of four-momentum, we add the muon p_T and missing E_T. Since the z-component of the missing energy is not known, the invariant mass cannot be calculated. As a result, only the $\eta - \phi$ distance and transverse E_T difference are shown in Figure 11. We then use these distributions as a benchmark against which we can compare the distributions of the more challenging hadronic W matching.



(a) E_T difference

(b) $\eta - \phi$ distance

Figure 11: Leptonic W distributions for the E_T difference and $\eta - \phi$ difference.

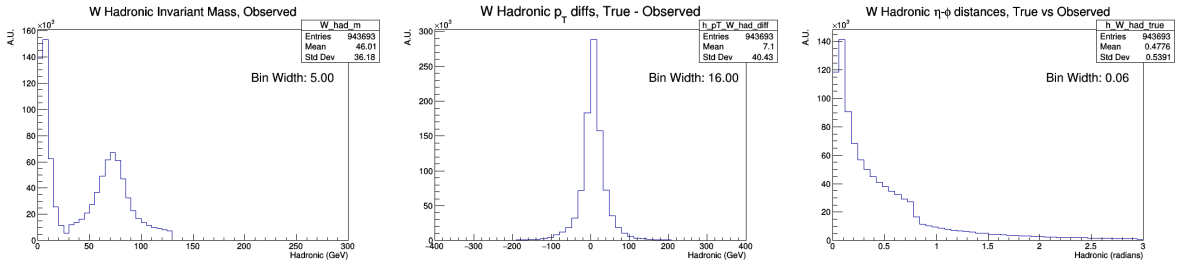
3.2 Hadronic W Bosons

Although the hadronically decaying W-boson typically results in two jets, its decay products may also be observed as three jets or even a single jet. To account for the detector response, we consider matches between the hadronic W-boson and combinations of one, two, and three jets. We also remove jets that have been b-tagged from consideration as they should be matched to bs and not Ws.

The matching algorithm first iterates through every possible sum of two jets' momenta. It identifies the two jet combination whose total momentum vector results in the minimum $\eta - \phi$ distance to the particle's truth momentum. If the best two jet match is not within some specified $\eta - \phi$, invariant mass, and p_T difference tolerances with the true hadronic W, then the match is discarded, and the process is repeated with all three jet combinations. If there are no satisfactory two or three jet matches, a match is made with the closest single jet.

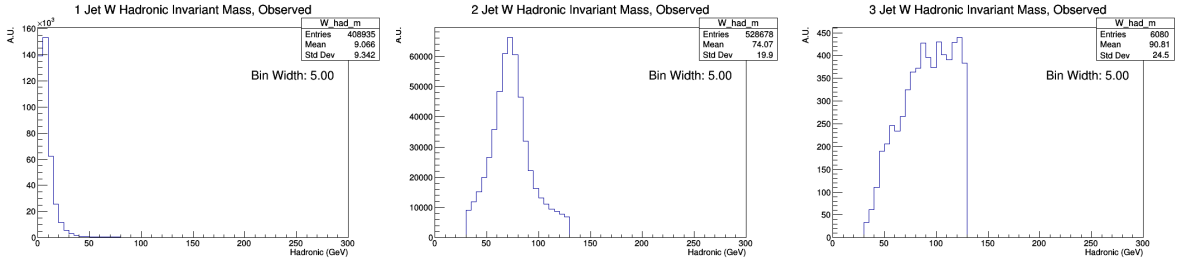
This process ensures that the maximum number of satisfactory two and three jet matches are made by prioritizing those jet combinations, resulting in the distributions in Figure 12.

As seen in Figure 12a, a large number of events are matched to jets with very low mass, near 0 GeV. Upon further examination of each type of jet combination, it was found that the low mass events were almost all matched to single jets, as Figure 13a shows. Consequently, 43% of events were matched to single jets, a much larger percentage than what we expect to observe. The mass distribution of events matched to two jets (56% of all events) has a peak at 80 GeV, the mass of the W-boson. 0.6% of events are comprised of three jet matches, whose masses seem to mainly consist of background noise with a small increase near the W-boson mass.



(a) Invariant mass (b) p_T difference (c) $\eta - \phi$ distance

Figure 12: Hadronic W distributions for the invariant mass, p_T-difference, and $\eta - \phi$ distance. Note the large peak at masses close to 0 in Figure 12a.

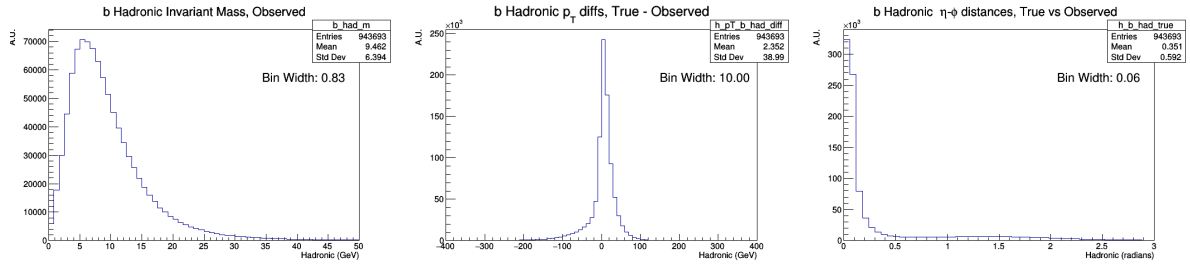


(a) 1-jet matches' invariant mass (b) 2-jet matches' invariant mass (c) 3-jet matches' invariant mass

Figure 13: Hadronic W invariant mass distributions for closest matches consisting of single, double, and triple jets.

3.3 Hadronic and Semi-Leptonic b-Quarks

The hadronic and semi-leptonic b-quarks each produce one jet. Our algorithm searches among all four or five jets, including b-tagged jets. Matches are made between the single jet with the smallest $\eta - \phi$ distance and the truth b-quark. It is possible for the same jet to be matched to a b-quark and the hadronic W.



(a) Invariant mass (b) p_T difference (c) $\eta - \phi$ distance

Figure 14: Hadronic b distributions for the invariant mass, p_T difference, and $\eta - \phi$ distance.

The distributions for all three variables (Figure 14) look more realistic than the corres-

ponding distributions for the hadronic W. As only one jet needs to be matched to the truth, the network is able to reconstruct the b quarks' momenta more accurately.

3.4 Filtered Training Dataset

We next use the three variables (Equations 4, 5, and 6) to design a set of cuts for all particles in each event. An event where all three variables for all Ws and bs pass the cuts is deemed reconstructable. We hope that training the network on a subset of the initial sample containing only reconstructable events will improve the accuracy of the predictions. These results are shown in Section 3.5.

Cuts on the $\eta - \phi$ distance are made for all particles. Further requirements are made on the p_T difference of the b-quarks and hadronic W-boson, as well as the E_T difference of the semi-leptonic W-boson. Finally, mass is used as a criteria for the hadronic W-boson only because its distribution for the b-quarks is already realistic (Figure 14a). Approximately 34% of events are remaining after applying all cuts, or 3171365 events out of 9438633 when run on the entire dataset.

The requirements made on each particle are summarized in Table 1. The last row indicates the percentage of events remaining when cuts for each particle are made separately. The mass requirements made on the hadronic W-boson are the strictest, accounting for the removal of 43% of events.

	Hadronic W	Semi-Leptonic W	Hadronic b	Semi-Leptonic b
Mass (GeV)	(30, 130)			
p_T difference (GeV)	(-100, 100)	(-100, 120)	(-80, 100)	(-80, 100)
$\eta - \phi$ difference (unitless)	(0, 0.8)	(0, 1.0)	(0, 0.8)	(0, 0.8)
% events remaining	57.3%	92.5%	82.1%	82.6%

Table 1: Requirements on variables for all decay particles in reconstructable events. Variables that pass the cuts must lie within the given intervals.

3.5 Training Results

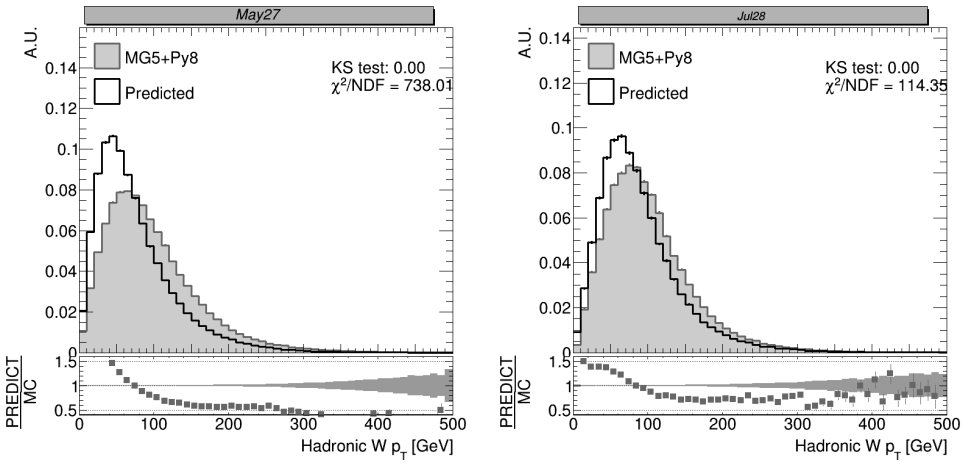
Training and evaluating the network on the subset with only reconstructable events provided a noticeable improvement compared to the full dataset. The largest improvement was seen for the hadronic W-boson, while the network showed the least improvement for the leptonic W-boson. This was expected as the network could already successfully reconstruct the leptonic W momentum for the original sample. Significant improvements were also observed for both b-quarks, whose reconstruction is also a jet matching problem (albeit a simpler one than for the hadronic W). Comparisons of the kinematic variable

distributions are shown in Figures 15, 16, and 17. The distributions for the semi-leptonic b are similar to those for hadronic b. Although all kinematic variables show improvements in their distributions, only the plots for p_T are shown here to avoid an overabundance of plots.

Looking at the correlation coefficients, the network shows some improvement for all particles. Like for the distributions themselves, the hadronic W coefficients show the largest improvement, while the leptonic W coefficients show the smallest. Figure 18 plots the original correlation coefficients for the p_T of each variable when trained on the full dataset, and the increase when trained and tested on the subset.

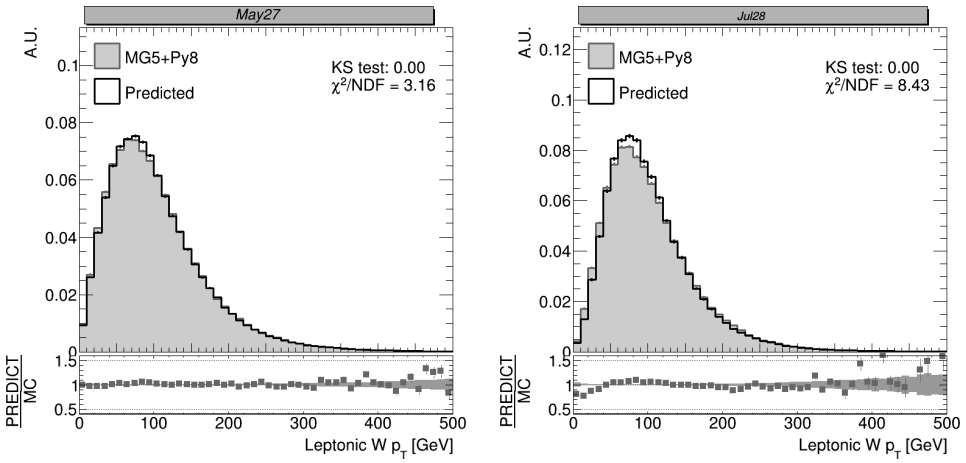
The optimum length of training was also investigated for the reduced dataset. The optimum number of epochs was determined to be around 25 epochs, based on the correlation coefficients for each variable. Unlike the full dataset, there are no significant improvements in any variable beyond 15 epochs. Looking at the training history across 50 epochs in Figure 19, both training and validation losses start to plateau after 20-25 epochs.

Although the final loss value while training on the reduced dataset is higher than for the full dataset, the physical interpretation of the loss value is unclear and the histograms and correlation plots both suggest that the network performs better on the reduced dataset.



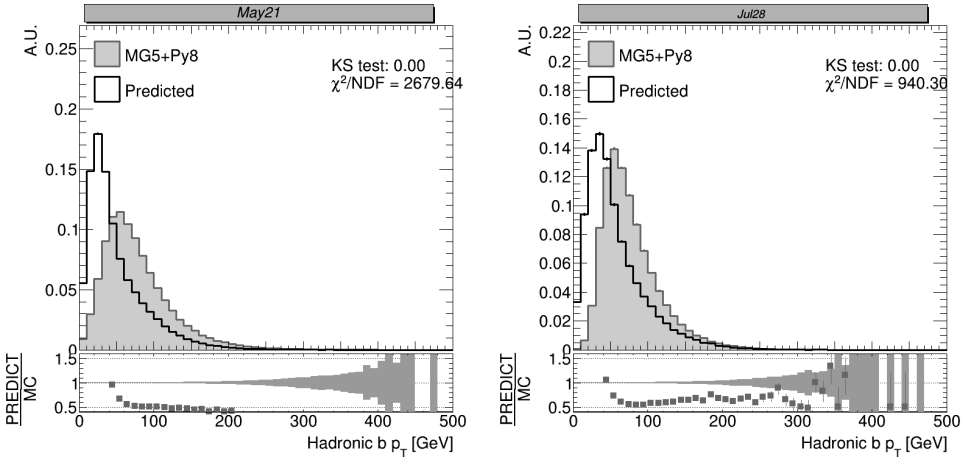
(a) Original sample (b) Reconstructable subset

Figure 15: Sample p_T distributions for hadronic W after training for 36 epochs.



(a) Original sample (b) Reconstructable subset

Figure 16: Sample p_T distributions for leptonic W after training for 36 epochs.



(a) Original sample (b) Reconstructable subset

Figure 17: Sample p_T distributions for hadronic b after training for 36 epochs.

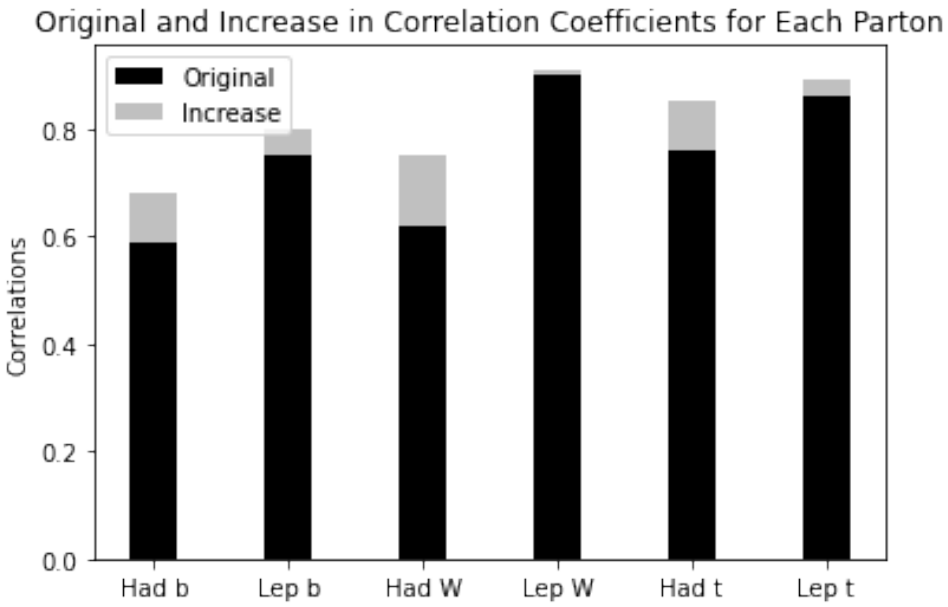


Figure 18: Correlation coefficients of p_T for each particle. The original correlations when trained on the full dataset is in black, and the increase after training on the reduced subset is in grey.

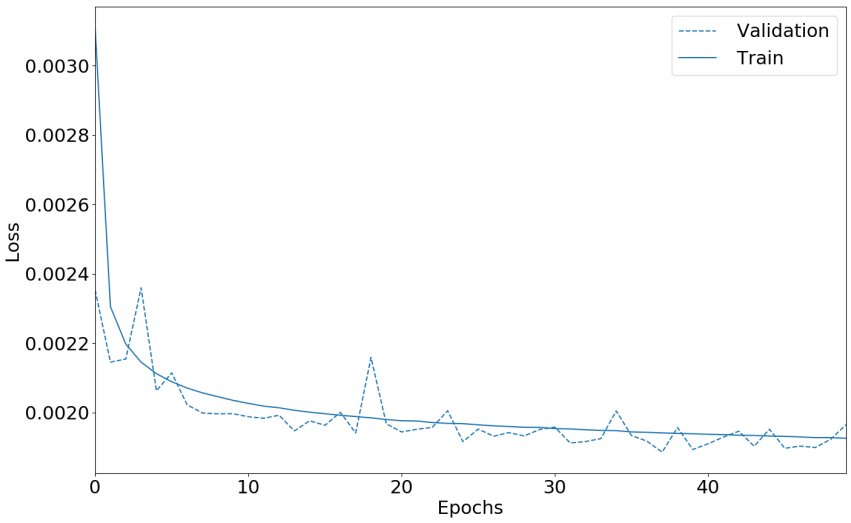


Figure 19: History of the loss function after training the network for 50 epochs on the reduced dataset.

4 Statistical Comparisons of Truth vs. Predicted

Current metrics used in Sections 2.1, 2.3, and 2.4 to evaluate how well the network trains do so by comparing predicted variable distributions against truth variable distributions (eg: correlation coefficients). However, these metrics do not directly measure individual events' predicted and truth variables against each other. We would like to quantify the network's ability to match momenta event-by-event.

To this end, we developed a new χ^2 statistic to compare the network performance under different training conditions, such as training length and using different datasets. The χ^2 for a given event is defined as

$$\chi^2 = \sum_{W,b} \left[\frac{(\Delta\phi_i)^2}{\sigma^2_{\phi,i}} + \frac{(\Delta\eta_i)^2}{\sigma^2_{\eta,i}} + \frac{(\Delta p_{T,i})^2}{\sigma^2_{p_{T,i}}} \right] \tag{7}$$

summed over each W-boson and b-quark. $(\Delta\phi_i)^2$, $(\Delta\eta_i)^2$, and $(\Delta p_{T,i})^2$ are the squared predicted-truth differences for each variable. The σs are calculated based on the FWHM of the residual histograms using $\sigma = FWHM/2.3548$. We use the FWHM to calculate σ because the FWHM is more resistant to long-tailed distributions than ROOT's Gaussian fit function. For each variable, the FWHM is approximated using the histogram's bin centres from plots of $\phi_{predicted} - \phi_{truth}$, $\eta_{predicted} - \eta_{truth}$, and $p_{T,predicted} - p_{T,truth}$, such as Figure 20.

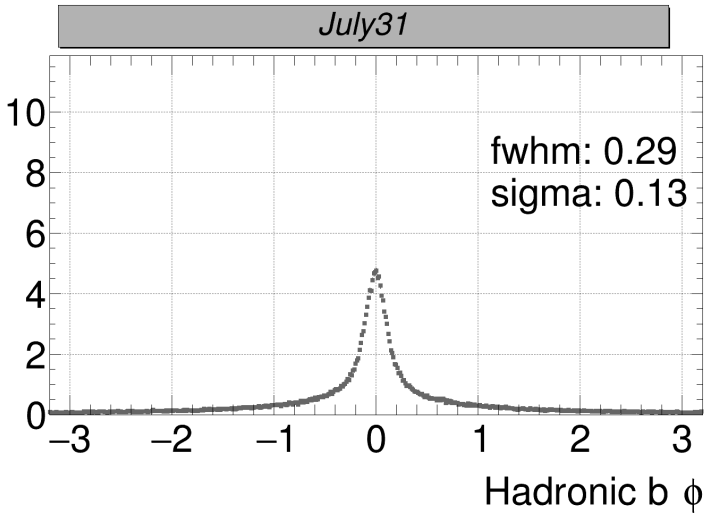
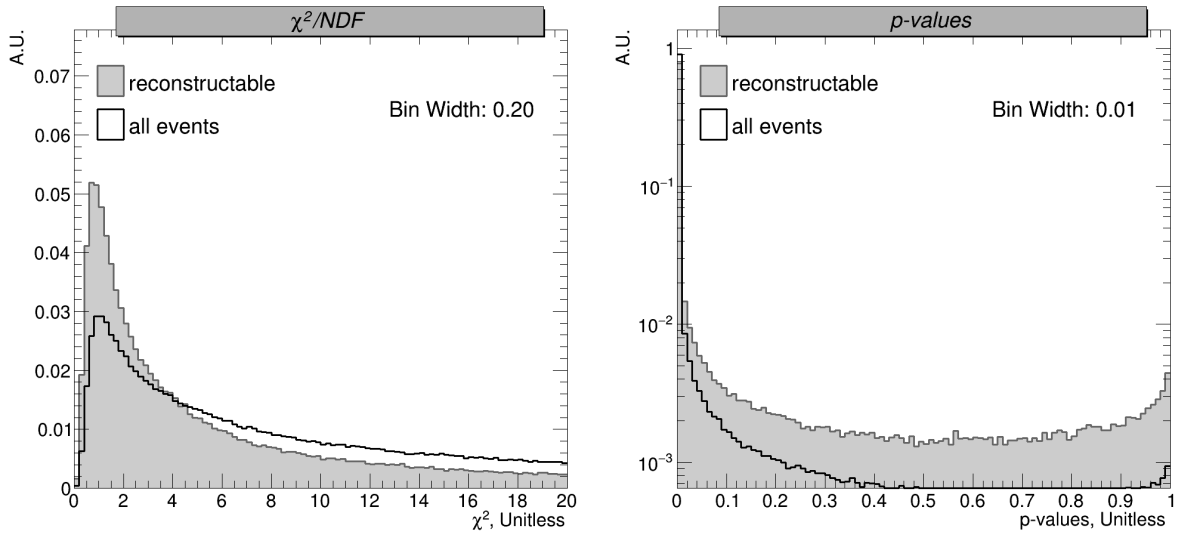


Figure 20: Residual/difference plot of the hadronic b ϕ with 500 bins.

The resulting χ^2s have 12 degrees of freedom, as we sum three variables for both bs and Ws. We divide χ^2 by 12 to get the reduced χ^2, which are plotted for the network's predictions when trained on all events and when trained on the reconstructable sample only. Both distributions have a peak around 1, indicating that most events are well predicted. These plots are shown in Figure 21a, and further affirms that the network



(a) Reduced χ^2 distributions

(b) p-values of χ^2 per event

Figure 21: Reduced χ^2 and p-values of all events after training the network 36 epochs. The network is trained and tested on the entire sample in black, and the reduced subset in grey.

performs better on the reduced sample, which has a narrower tail and more events with a reduced χ^2 around 1.

The χ^2s for each event are then used to calculate p-values using the following equation:

$$p - value = 1 - CDF(\chi^2), \tag{8}$$

where CDF is the cumulative distribution function of a chi-squared distribution with 12 degrees of freedom. Our null hypothesis assumes that there is no difference between the network's predictions and the truth distributions. We therefore reject events with small p-values, as those are events that the network has predicted poorly. As expected, in Figure 21b, the reconstructable sample has more events with higher p-values, and slightly fewer events with p-values near 0.

An interesting observation is that the plots in Figure 21 look the same whether the network is trained on the full sample or only on the unreconstructable sample. This is probably because two-thirds of events in the full sample are unreconstructable to begin with.

Lastly, we would like to explore to explore the use of a statistical metric (that does not require any truth information) to evaluate the quality of a given event. The reason for doing this is that we plan to eventually feed AngryTops real lepton+jets data from the LHC. Unlike our current simulated data, there is no corresponding truth momentum to compare against for real events.

One strategy is to replace the truth momenta with momenta reconstructed using KLFitter.

As a step in this direction, we began by determining if p-values could be used as a proxy for reconstructable events. Recall that large p-values support the null hypothesis that there is no difference between the predicted and truth variables.

Beginning with distributions such as those in Figure 21b, we applied a series of cuts to both the reconstructable and original testing datasets. The percentages of events remaining with p-values higher than the cutoff are shown in Table 2.

p-value	% of reconstructable events	% of total events	# of reconstructable events / # of total events	% of reconstructable events / % of total events
0	100%	100%	34%	1.00
0.01	22%	9.7%	77%	2.27
0.05	18%	7.6%	81%	2.37
0.1	16%	6.5%	84%	2.46

Table 2: Effect of p-value cuts on the size of the sample. The first column shows the value of the cut below which all p-values are rejected. The second column shows the fraction of reconstructable events that pass the cut. The third column shows the fraction of all events, both reconstructable and unreconstructable, that pass the cuts. The fourth column shows the ratio of the raw number of reconstructable events that pass to the raw number of total events that pass. The fifth column shows the result obtained by dividing the values in the second column by the values in the third column.

Indeed, the fourth column of Table 2 shows that as we implement stricter and stricter p-value cuts (thus retaining fewer and fewer events), a larger proportion of the events that pass the cuts are reconstructable events. The fifth column can be defined/interpreted by Equation 9:

$$\begin{aligned}\frac{\text{Signal}}{\text{Background}} &\approx \text{Efficiency} * \text{Rejection factor} \\ &= \frac{\#\text{ of reconstructable that pass}}{\text{Total }\#\text{ of reconstructable}} * \left(1 \div \frac{\text{Total }\#\text{ of both types that pass}}{\text{Total }\#\text{ of both types}}\right)\end{aligned} \tag{9}$$

Thus, we conclude that it is possible to use a simple p-value cut instead of a complicated matching algorithm to identify reconstructable events. All that remains to be done is to find a suitable substitute for the truth values in the χ^2 test statistic.

We also see from the fifth column of Table 2 that there is not much improvement when the p-value cutoffs are increased from 0.01 through 0.1. This is because the majority of events have p-values very close to 0 and so are already rejected by the 0.01 p-value cut (see Figure 21b). This suggests that if a χ^2 metric not involving truth information is implemented, a low p-value cut will be sufficient to maximize the reconstructable signal-to-background ratio.

5 Conclusions and Future Work

We investigate several ways to characterize and improve the performance of the AngryTops machine learning reconstruction package. We found that the optimal number of epochs to train the network for was around 36 epochs, based on correlation coefficients for each variable. We also create quantile-quantile plots as a means to evaluate the network's performance on individual events. Furthermore, kinematic distributions confirm that the network performs better on the semi-leptonic decay channel. We also create a dataset consisting of easily reconstructable events using an updated jet matching algorithm. The network performs better when trained and tested on the filtered dataset.

A remaining improvement that has not yet been addressed is the incorporation of different cuts for each type of best-matched hadronic W combination (1-jet, 2-jets, and 3-jets, see Section 3.2). At the moment, we apply the same set of cuts for the hadronic W irrespective of the number of jets it is matched to. Allowing different sets of cuts will improve the percent of events that pass the cuts while retaining the correct physics.

Future points of investigation include exploring additional hyperparameters and training on different network architectures such as RNNs and CNNs. Data augmentation might also be required to prevent overfitting when the network trains on only the reconstructable subsample. Additionally, the network's performance may be compared with state-of-the-art statistical reconstruction algorithms such as KLFitter.

Acknowledgements

We would like to thank Professor Pekka Sinervo of the University of Toronto ATLAS Group for his invaluable mentorship and for sharing with us the joys of particle physics. We would also like to acknowledge funding from the University of Toronto Department of Physics via its NSERC USRA Program. Finally, DC would like to thank CERN and the Institute of Particle Physics for organizing a virtual, yet engaging CERN Summer Student Programme.

Authors' Contact Information

- Darren Zeming Chan: dzchan@uwaterloo.ca
- Maggie Fen Wang: maggiefen.wang@mail.utoronto.ca

References

[1] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H.-S. Shao, T. Stelzer, P. Torrielli, and M. Zaro. The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. *Journal of High Energy Physics*, 2014(7), Jul 2014.

[2] J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens, and M. Selvaggi. Delphes 3: a modular framework for fast simulation of a generic collider experiment. *Journal of High Energy Physics*, 2014(2), Feb 2014.

[3] Brandon Tan Chin Hian and Haolin Liu. Improvements to angrytops: a machine learning approach to top quark kinematics reconstruction. *University of Toronto*, 2020.

[4] Kuunal Mahtani. Reconstructing top anti-top quark pair kinematics using machine learning. *University of Toronto*, 2021.

[5] Gavin P. Salam. Towards jetography. *The European Physics Journal C*, 67:637–686, 2010.

[6] Torbjörn Sjöstrand, Stefan Ask, Jesper R. Christiansen, Richard Corke, Nishita Desai, Philip Ilten, Stephen Mrenna, Stefan Prestel, Christine O. Rasmussen, and Peter Z. Skands. An introduction to pythia 8.2. *Computer Physics Communications*, 191:159–177, Jun 2015.

[7] Fardin Syed, Riccardo Di Sipio, and Pekka K. Sinervo. Bidirectional long short-term memory (blstm) neural networks for reconstruction of top-quark pair decay kinematics. *arXiv*, 2019.

[8] P.A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020.

[9] Özal Yildirim. A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification. *Computers in Biology and Medicine*, 96:189–202, 2018.

A Appendices: Updates to ReadMe, a.k.a. How to Run the Relevant Scripts

These appendices should take precedence over any READMEs in the folders because this report is based on the most recent work.

A.1 Downloading and Setting Up AngryTops

Follow these instructions to set up the AngryTops package in your home directory:

1. Log onto the Huron computer cluster.
2. Clone the AngryTops project in the appropriate branch from https://github.com/IMFardz/AngryTops
3. As an alternative to Step 2, copy either the directory `~dchan/AngryTops` or `~mwang/AngryTops` into your home directory.
4. Add the following two lines to the .bashrc file in your home directory:
   - `source /usr/local/packages/root/bin/thisroot.sh`
   - `export PYTHONPATH=$PYTHONPATH:'/home/YOURNAME/AngryTops/'`
5. Run the above two commands or log out of Huron and log back in.
6. You're good to go! Welcome to AngryTops; we hope you have a great time improving it.

A.2 Training Script

Training - we updated the training script to output a cleaner log file. For future users, might be easier to put items 5-6 in a python file in the `AngryTops/AngryTops` folder and run directly, where each parameter can be changed via a command line argument. To train the network:

1. Create a new folder in `CheckPoints` with today's date in the following format: `mkdir ~/AngryTops/CheckPoints/Jul30`
2. Open screen as training takes many hours: `screen`
3. Run `ipython |& tee ~/AngryTops/CheckPoints/DATE/log.txt`

4. Run `cd ~/AngryTops/AngryTops`
5. Run `from ModelTraining import train_simple_model`
6. Run `train_simple_model.train_model("BDLSTM_model", "Jul30", "Feb9.csv", scaling='minmax', rep='pxpypzEM', EPOCHES=25, sort_jets=False, load_model=False, log_training=True)`
7. Replace `"Feb9.csv"` with `"Jul26_good.csv"` if you want to train on the reconstructable subset or `"Jul26_bad.csv"` for the unreconstructable subset.

A.3 Plotting Scripts

After the training is complete, first generate plots of the kinematic variables in the `img` subfolder:

1. Run `cd ~/AngryTops/AngryTops`
2. Run `bash make_plots.sh ~/AngryTops/CheckPoints/DATE pxpypzEM DATE` where `DATE` is the training date.

Next, the following scripts must be run after `make_plots.sh` as they use the tree generated in `Plotting/fit.py`. They can be run directly through python2:

1. Run `cd ~/AngryTops/AngryTops`
2. Run `python2 SCRIPT.py TRAINING_DIR REP CAPTION ADD_ARGS`

`TRAINING_DIR` is formatted as `/home/YOURNAME/AngryTops/CheckPoints/SUBDIR`. The representation should be the same as the representation used during training, and caption is used as the titles of the output plots. Not all of these arguments are used, but are included for consistency. Any additional arguments required must be added after the first three, and are specified in Table 3.

Purpose	**File Name**	**Other Arguments**
qq plots	`Plotting/qq.py`	`ZOOM_FACTOR`
energy of t and $b+W$	`Plotting/kinematics_energy.py`	
angle between t and $b+W$	`Plotting/kinematics_Pangle.py`	`LOG_AXIS`

Table 3: Plotting Scripts

`ZOOM_FACTOR` multiplies the x and y axes of the qq plots by a factor between 0 and 1 to create a cropped version of the plots. `LOG_AXIS` creates a semi-log plot if True, and regular histogram if False or unspecified.

A.4 Jet Matching Scripts

For the testing dataset - `bW_identification.py` matches jets to truth particles' momenta, applies cuts, and outputs the resulting p_T, $\eta-\phi$, and mass histograms for the bs and Ws. The script also generates plots of the $\eta-\phi$ distance between Predicted and Truth, as well as Predicted and Observed. Note: The folder `~/AngryTops/CheckPoints/DATE/closejets_img_c` must be created in `CheckPoints` before running this scripts, if the output text file is to be made in the specified location.

1. Run `cd ~/AngryTops/AngryTops`
2. Run `python2 Plotting/bW_identification.py ~/AngryTops/CheckPoints/DATE/ pxpypzEM > ~/AngryTops/CheckPoints/DATE/closejets_img_cuts/closejets.txt`

For the training dataset - The script below reads in "Feb9.csv", matches jets to truth momenta, applies the same cut parameters as `bW_identification.py`, and saves the resulting array in a csv of the same format as the input csv. `PLOT_CUTS` plots only events that passes the cuts if True, and all events if False. DATE specifies the name of the output csv.

1. Run `cd ~/AngryTops/AngryTops`
2. Run `python2 Plotting/bW_id_csv_filtering.py ~/AngryTops/csv/training_closejets_cuts pxpypzEM PLOT_CUTS DATE > ~/AngryTops/csv/training_closejets_cuts/closejets.txt`

A.5 Chi-Squared Statistical Comparison Script

The script in this section performs all statistical analysis described in Section 4. This includes generating distributions of χ^2 and p-values. This script also has the ability to compare different distributions corresponding to samples containing all events, only reconstructable events, or only unreconstructable events. To run this script:

1. Run `cd ~/AngryTops/AngryTops`
2. Run `python2 Plotting/chi.py DATE1 DATE2 DATE3 'unreconstructable' > ~/AngryTops/CheckPoints/DATE1/img_chi_pval/chi_pval.txt`

`DATE1` is the date of the folder containing the difference plots used to calculate the standard deviations for each variable. `DATE2` is the date of the folder containing the training output

from the reconstructable subsample. `DATE3` is the date of the folder containing the training output from the unreconstructable subsample. 'unreconstructable' should be replaced by 'all events' depending on the type of event in the `DATE3` folder. The output folder of `chi.py` is the `DATE1` folder.