Jet-Images – Deep Learning Edition

Luke de Oliveira,[a] **Michael Kagan,**[b] **Lester Mackey,**[c] **Benjamin Nachman,**[b] **and Ariel Schwartzman**[b]

[a] *Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA*
[b] *SLAC National Accelerator Laboratory, Stanford University, 2575 Sand Hill Rd, Menlo Park, CA 94025, U.S.A.*
[a] *Department of Statistics, Stanford University, Stanford, CA 94305, USA*
E-mail: lukedeo@stanford.edu, mkatan@cern.ch, lmackey@stanford.edu, bnachman@cern.ch, sch@slac.stanford.edu

Abstract: Building on the notion of a particle physics detector as a camera and the collimated streams of high energy particles it measures as an image, we investigate the potential of machine learning techniques based on deep learning architectures. Modern deep learning algorithms trained on *jet images* can out-perform standard physically-motivated feature driven approaches to jet tagging. We develop techniques for visualizing where these features are learned by the network and what additional information is used to improve performance. This feedback loop between physically-motivated feature driven tools and unsupervised learning algorithms is general and can be used to significantly increase the sensitivity to discover new particles and new forces.

1 Introduction

Jets, or collimated sprays of particles resulting from the production of high energy quarks, provide a unique handle to search for signs of new physics at Large Hadron Collider (LHC). Specifically, new heavy particles are predicted by a wealth of theories to decay to Standard Model (SM) particles, such as $W^{\pm}$, Z, and Higgs bosons, or top quarks, and in doing so impart such SM particles large amounts of energy, or boost. This boost leads to the collimation of the SM particle's decay products, resulting in the formation of boosted heavy particle jets when the SM particle decays to quarks. Boosted jets have a rich internal substructure which is unlike that of the typical jet backgrounds produced from vanilla Quantum Chromo Dynamic (QCD) interactions. The identification of the underlying particle which produced a boosted jet, or jet tagging, is a fundamental challenge to searching for signs of new physics at the LHC. There is a wealth of literature addressing the topic of jet tagging by designing physics-inspired features to exploit the jet substructure [**?**]. However, in this paper we address the challenge of jet tagging though the use of Machine Learning (ML) and Computer Vision (CV) techniques combined with low-level information, rather than using physics inspired features. In doing so, we not only improve discrimination power, but also gain new and deep insight into the underlying physical processes that provide discrimination power by extracting the information learned by such ML algorithms.

The analysis presented here is an extension of the jet-images approach, first studied in [**?**] and then studied with similar approaches by [**?**], whereby jets are represented as images with the energy depositions of the particles within the jet serving as the pixel intensities. When first introduced, jet-image preprocessing techniques, based on the underlying physics symmetries of the jets, were combined with linear Fisher discriminant analysis (FDA) to perform jet tagging and to study the learned discrimination information. Here, we make use of modern deep neural networks (DNN) architectures, which have been found to outperform competing algorithms in CV tasks similar to jet-tagging with jet-images. While such DNN's are significantly more complex than FDA, they also provide the capability to learn rich high-level representations of jet-images and to greatly enhance discrimination. By developing techniques to access this rich information, we can explore and understand what has been learned by the DNN's and subsequently use this to improve our understanding the physics governing jet substructure. We also re-examine the jet pre-processing techniques, to specifically analyze the impact of the pre-processing on the physical information contained within the jet.

Since it's first usage by it's current name [**?**], Deep Learning has taken on many forms and seen success in a variety of fields that have traditionally utilized human-engineered features to create classifiers and apply out-of-the-box machine learning algorithms. In particular, the field of Computer Vision has changed drastically. Since the 2012 ILSVRC winning entry by Alex Krizhevsky and the University of Toronto group [**?**], Deep Learning – in particular Convolutional Neural Networks – have taken over vision-based machine learning, consistently showing human and recently super-human levels performance on key baseline datasets. The increasingly widespread availability of GPUs and associated numerical frameworks has made the time intensive estimation procedures associated with deep neural networks more feasible, and has allowed the size of models for image tasks to grow exponentially. For example, the Google team's contribution to ILSVRC 2014 – the GoogLeNet [**?**] – consisted of 22 layers of convolutional black boxes called "Inception Units", and set the benchmarks both for accuracy and speed of a model on such a large scale.

As it relates to our work, we investigate several deep network architectures, though not as large as some described above, and focus on understanding what information and higher level representations a fully connected and a convolutional neural network will learn in the context of High Energy

Physics. We let our knowledge of physics guide our investigations into visualization, understanding, and demystification of deep representations for physics. We shed light inside the black-box of deep learning in the context of object identification in HEP.

This paper is organized as follows: The details of the simulated data sets and the definition of jet-images are described in Section 2. The pre-processing techniques, included new insights into the relationship with underlying physics information, is discussed in Section 3. We then introduce the deep neural network architectures that we us in Section 4. The discrimination performance and the exploration of the information learned by the DNN's is presented in Section 5.

2 Simulation Details and the Jet Image

In order to study jet images in a realistic scenario, we use Monte Carlo (MC) simulations of high energy particle collisions. One important jet tagging application is the identification of highly Lorentz boosted W bosons decaying into quarks amidst a large background from the generic production of quarks and gluons. This classification task has been thoroughly studied experimentally[1] [1–3] and used in many analyses [4–16].

To simulate highly boosted W bosons, a hypothetical W' boson is generated and forced to decay to a hadronically decaying W boson ($W \to qq'$) and a Z boson which decays invisibly ($Z \to \nu\bar{\nu}$). The mass of the W' boson determines the Lorentz boost of the W boson in the lab frame since the W' is produced nearly at rest and the W boson momentum is approximately $m_{W'}/2$. The invisible decay of the Z boson ensures that the jet in the event with the highest transverse momentum is the W boson jet. Multijet production of quarks and gluons is simulated as a background. Both the W' signal and the multijet background are generated using Pythia 8.170 [17, 18] at $\sqrt{s} = 14$ TeV. The angular separation of the W boson decay products in the plane transverse to the beam direction scales as $2m_W/p_{T,W}$, where $m_W \approx 80$ GeV and $p_{T,W}$ is the component of the W boson momentum in this plane. The tagging strategy and performance depend strongly on $p_{T,W}$, so we focus on a particular range: 250 GeV $< p_{T,W} <$ 300 GeV. This corresponds to an angular spread of about 1 radian. The decay products of the W bosons as well as the background are clustered into jets using the anti-k_t algorithm [19] via FastJet [20] 3.0.3. To mitigate the contribution from the underlying event, jets are are trimmed [21] by re-clustering the constituents into $R = 0.3$ k_t subjets and dropping those which have $p_T^{\text{subjet}} < 0.05 \times p_T^{\text{jet}}$.

To model the discretization and finite acceptance of a real detector, a calorimeter of towers with size 0.1×0.1 in (η, ϕ) extends out to $\eta = 5.0$. The total energy of the simulated particles incident upon a particular cell are added as scalars and the four-vector p_j of any particular tower j is given by

$$p_j = \sum_{i \text{ incident on } j} E_i(\cos\phi_j/\cosh\eta_j, \sin\phi_j/\cosh\eta_j, \sinh\eta_j/\cosh\eta_j, 1), \tag{2.1}$$

where E_i is the energy of particle i and the center of the tower j is (η_j, ϕ_j). Towers are treated as massless.

A *jet image* is formed by taking the constituents of a jet and discretizing its energy into pixels in (η, ϕ), with the intensity of each pixel given by the sum of the energy of all constituents of the jet inside that (η, ϕ) pixel. In our studies, we take the jet image pixelation to match the simulated calorimeter tower granularity. In the next section, we will discuss the nuances of standardizing the coordinates of a jet image as a pre-processing step prior to applying machine learning.

[1] There is also an extensive literature on phenomenological studies - see references within the experimental papers.

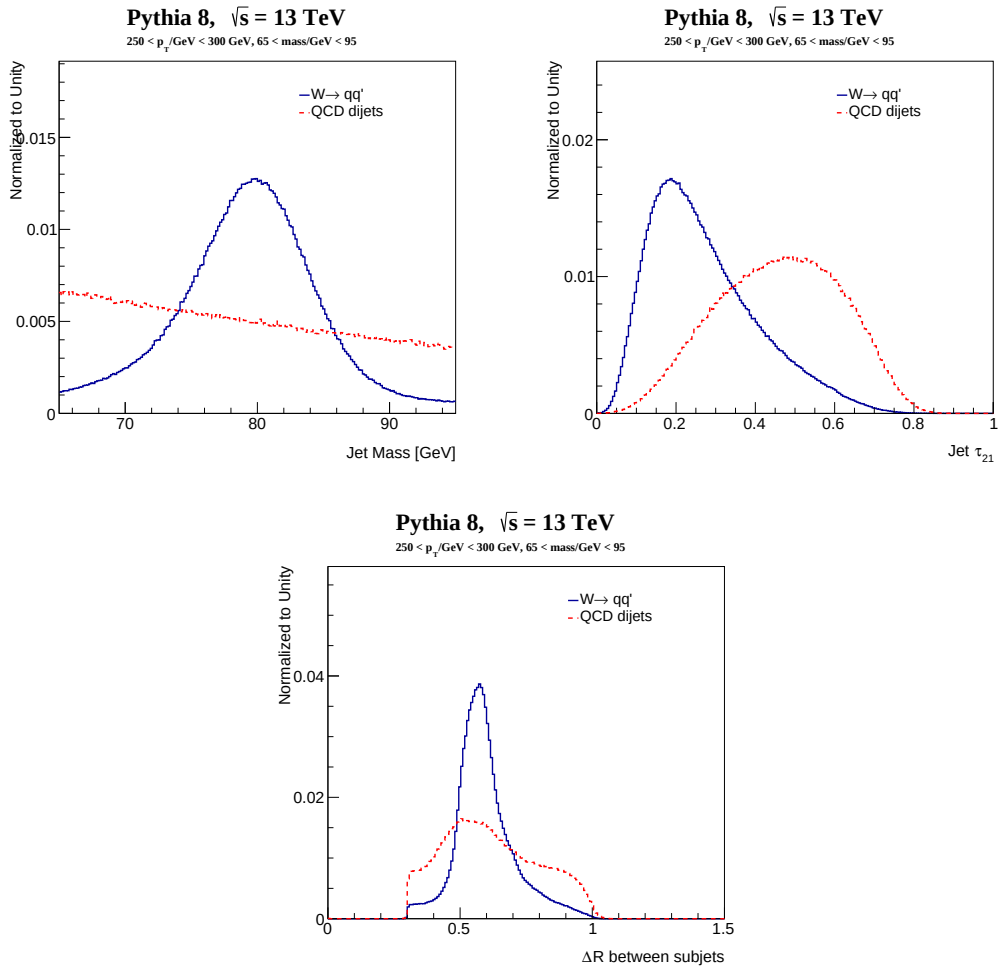


Figure 1

pT weighting

3 Pre-processing and the Symmetries of Space-time

In order for the machine learning algorithms to most efficiency learn discriminating features between signal and background and not learn the symmetries of space-time, the jet images are pre-processed. This procedure can greatly improve performance and reduce the required size of the sample used for testing. Our pre-processing procedure happens in four steps: translation, rotation, re-pixelation, and inversion. To begin, the jet images are translated so that the leading subjet is at $(\eta, phi) = (0, 0)$. Translations in ϕ are rotations around the z-axis and so the pixel intensity is unchanged by this operation. On the other hand, translations in η are *Lorentz boosts* along z, which do not preserve the pixel intensity. Therefore, a proper translation in η would modify the pixel intensity. One simple modification of the jet image to circumvent this change is to replace the pixel intensity E_i with $p_{T,i} = E_i/\cosh(\eta_i)$. This new definition of intensity is invariant under translations in η and is used exclusively for the rest of this paper.

The second step of pre-processing is to rotate the images around the center of the jet. If a jet has a second subjet, then the rotation is performed so that the second subjet is at $-\pi/2$. If no second subjet exists, then the jet image is rotated so that the first principle component of the pixel intensity distribution is at $-\pi/2$. Unless the rotation is by an integer multiple of $\pi/4$, the rotated grid will not line up with the original grid. Therefore, the energy in the rotated grid must be re-distributed amongst the pixels of the original image grid. A cublic spline interpolation is used in this case - see Ref. [**?**] for details. The last step is a parity flip so that the right side of the jet image has the highest sum pixel intensity.

Figure 2 shows the average jet image for W boson jets and QCD jets before and after the rotation, re-pixelation, and inversion steps of the pre-processing. The more pronounced second-subjet is already pronounced in the left plots of Fig. 2, where there is a clear annulus for the signal W jets which is nearly absent for the background QCD jets. However, after the rotation, the second core of energy is well isolated and localized in the images. The spread of energy around the leading subjet is more diffuse for the QCD background which consists largely of gluon jets which have an octet radiation pattern compared to the singlet radiation pattern of the W jets, where the radiation is mostly restricted to the region between the two hard cores.

One standard pre-processing step that is often additionally applied in machine learning is normalization. A common normalization scheme is the L^2 norm such that $\sum I_i^2 = 1$ where I_i is the intensity of pixel i. This is particularly useful for the jet images where pixel intensities can span many orders of magnitude. The jet transverse momenta are all around 250 GeV, but this can be spread amongst many pixels or concentrated in only a few and the L^2 norm helps mitigate the spread and thus makes training easier for the machine learning algorithm. However, normalization can distort information contained within the jet image. Some information, such as the Euclidean distance ΔR between subjets in (η, ϕ) is invariant under all of the pre-processing steps as well as normalization. However, consider the *image mass*,

$$m_I^2 = \sum_{i<j} E_i E_j (1 - \cos(\theta_{ij})), \tag{3.1}$$

where $E_i = I_i/cosh(\eta_i)$ for pixel intensity I_i and θ_{ij} is the angle between massless four-vectors with η and ϕ at the i and j pixel centers. The image mass is not invariant under all pre-processing steps, but does encode useful discrimination information. As discussed earlier, with the proper choice of pixel intensity, translations preserve the image mass since it is a Lorentz invariant quantity. However, the

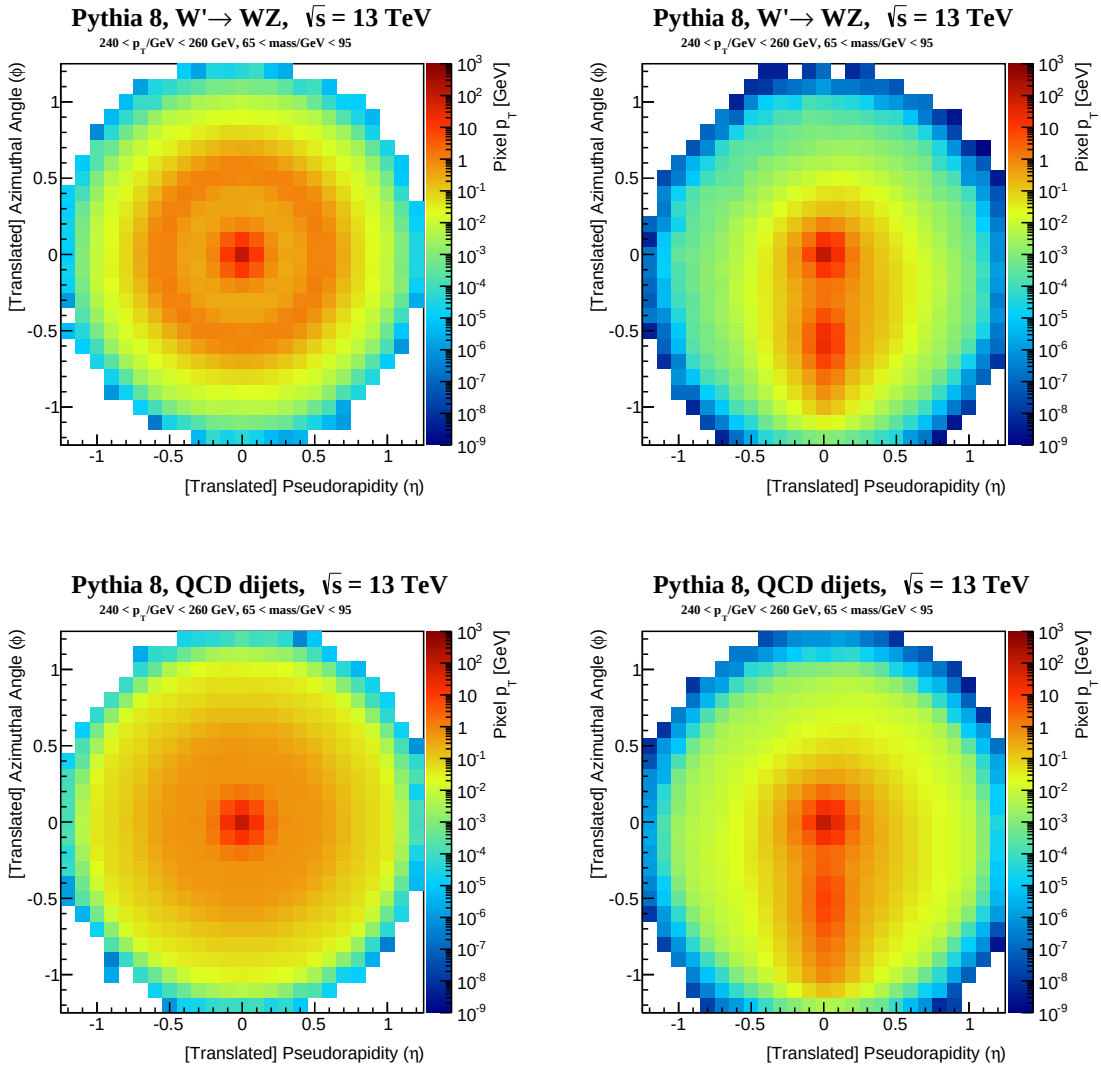


Figure 2: The average jet image for signal W jets (top) and background QCD jets (bottom) before (left) and after (right) applying the rotation, re-pixelation, and inversion steps of the pre-processing. The average is taken over images of jets with 240 GeV $< p_T <$ 260 GeV and 65 GeV $<$ mass $<$ 95 GeV.

rotation pre-processing step does not preserve the image mass. To see this, consider two four-vectors: $p^\mu = (1, 0, 0, 1)$ and $q^\mu = (0, 1, 0, 1)$. The invariant mass of these vectors is $\sqrt{2}$. The vector p^μ is at the center of the jet image coordinates and the vector q^μ is located at $\pi/2$ degrees. If we rotate the image around the jet axis so that the vector q^μ is at 0, akin to rotating the jet image so that the sub-leading subjet goes from $\pi/2$ to 0, then p^μ is unchanged but $q^\mu \to (1, 0, \sinh(1), \cosh(1))$. The new invariant mass of q^μ and p^μ is about 1, which is reduced from its original value of $\sqrt{2}$. The parity inversion

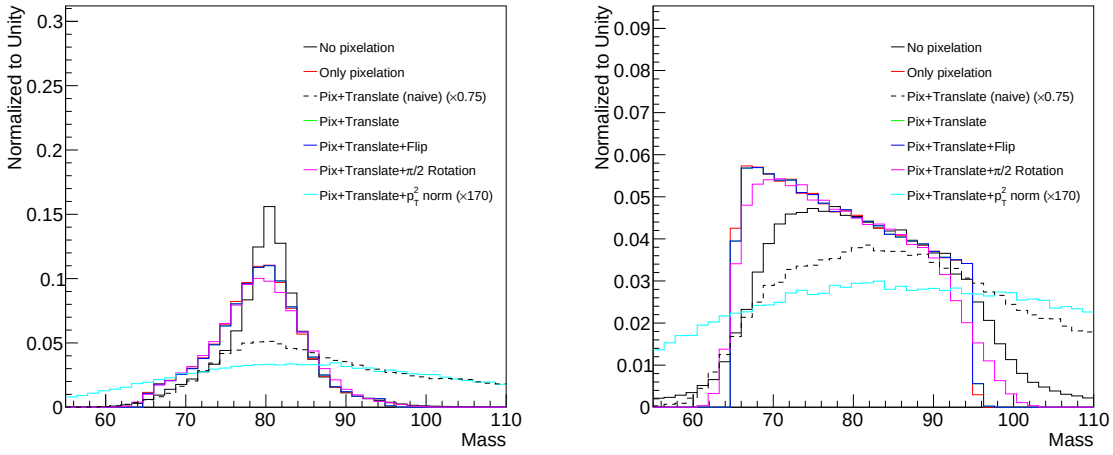


Figure 3: The distribution of the image mass after various states of pre-processing for signal jets (left) and background jets (right). The *No pixelation line* is the jet mass without any detector granularity and without any pre-processing. *Only pixelation* has only detector granularity but no pre-processing and all subsequent lines have this pixelation applied as well as translation to center the image at the origin. The translation is called *naive* when the energy is used as the pixel intensity instead of the tower transverse momentum. *Flip* denotes the parity inversion operation and the p_T^2 norm is a I^2 normalization scheme. The naive translation and the I^2 normalization image masses are both multiplied by constants so that the centers of the distribution are roughly in the same location as for the other distributions.

pre-processing step does not impact the image mass, but a I^2 normalization does modify the image mass. The easiest way to see this is to take a series of images with exactly the same image mass but variable I^2 norm. The map $I_i \mapsto I_i / \sum_j I_j^2$ modifies the mass by $m_I \mapsto m_I / \sum_j I_j^2$ and so the spread in the normalizations induces a spread in the mass distribution.

The impact of the various stages of pre-processing on the image mass are illustrated in Fig. 3. The finite granularity of the simulated detector slightly degrades the resolution, but the translation and parity inversion (flip) have no impact, by construction. The rotation that will have the biggest potential impact on the image mass is by $\pi/2$ (maximally changing η and ϕ), which does lead to a small change in the mass distribution. A naive translation in η that uses the tower energy as the intensity instead of the transverse momentum or an I^2 normalization scheme both significantly broaden the mass distribution. One way to quantify the amount of information in the jet mass that is lost by various pre-processing steps is shown in the Receiver Operator Characteristic (ROC) curve of Fig. 4. Information about the mass is lost when the ability to use the mass to differentiate signal and background is diminished. The naive translation and the I^2 normalization schemes are significantly worse than the other image mass curves which are themselves similar in performance.

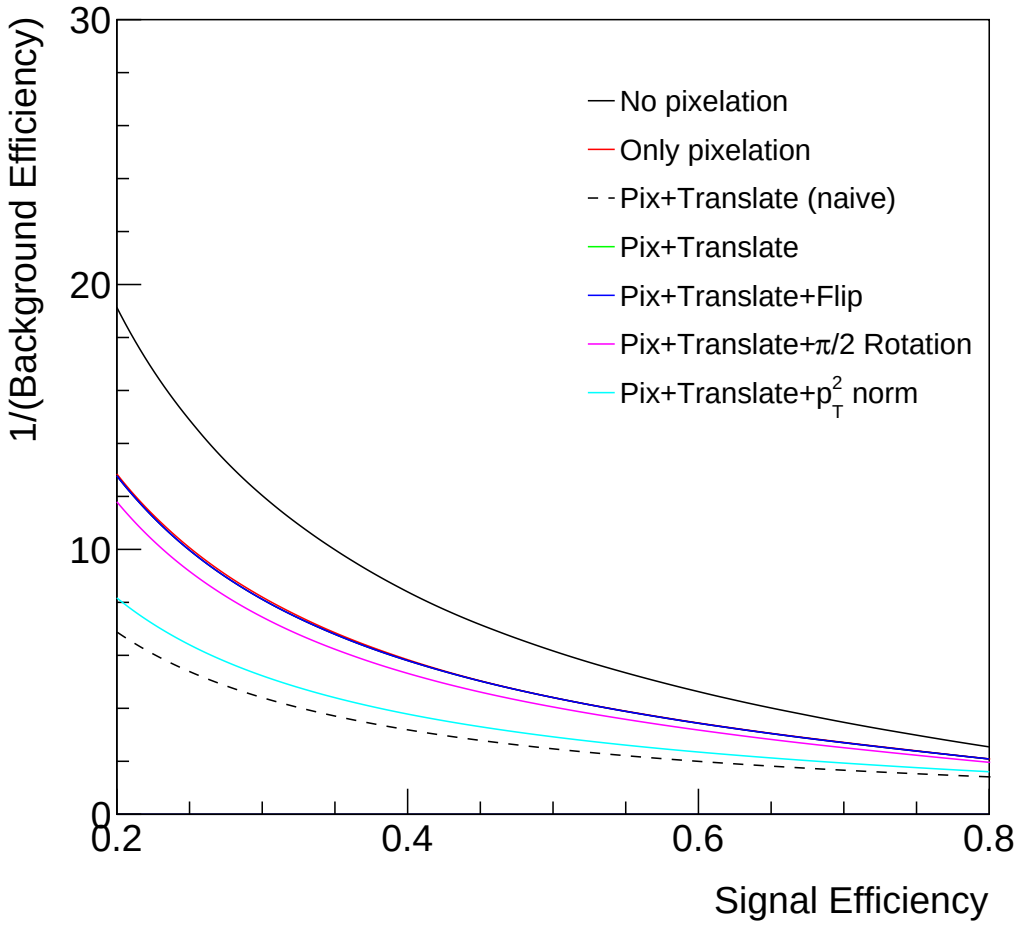


Figure 4: The tradeoff between W boson (signal) jet efficiency and inverse QCD (background) efficiency for various pre-processing algorithms applied to the jet (images). The *No pixelation line* is the jet mass without any detector granularity and without any pre-processing. *Only pixelation* has only detector granularity but no pre-processing and all subsequent lines have this pixelation applied as well as translation to center the image at the origin. The translation is called *naive* when the energy is used as the pixel intensity instead of the tower transverse momentum. *Flip* denotes the parity inversion operation and the p_T^2 norm is a I^2 normalization scheme.

4 Network Architecture

We begin with the notion that the discretization procedure outlined in Section 2 produces 25×25 "energy-scale" images in one channel – a High Energy Physics analogue of a grayscale image. We note that the images we work with are *sparse* – roughly 12% of pixels are active on average. Future work can build on efficient techniques for exploiting the sparse nature of these images – i.e., memoized convolutions. However, since speed is not our driving force in this work, we utilized convolution implementations defined for dense inputs.

4.0.1 Architectural Selection

We utilize a very simple convolutional architecture for our studies, consisting of two sequential `[Conv + Max-Pool + Dropout]` units, followed by two fully connected, dense layers. Our architecture can be succinctly written as

$$[\texttt{Dropout} \to \texttt{Conv} \to \texttt{ReLU} \to \texttt{MaxPool}] * 2 \to [\texttt{Dropout} \to \texttt{FC} \to \texttt{ReLU}] * 2 \to \texttt{Sigmoid}. \tag{4.1}$$

After early experiments with the standard 3×3 kernel size, we discovered no improvement over a more basic MaxOut [**?**] feedforward network. After further investigation into larger convolutional kernel size, we discovered that larger-than-normal kernels work well on our application. Though not common in the Deep Learning community, we hypothesize that this larger kernel size is helpful when dealing with sparse structures in the input images. In Table 1, we show the optimal kernel size of 11×11 while considering the metric outlined in Section 5.1.

Kernel size	**AUC**
(3×3) Conv	14.770
(4×4) Conv	12.452
(5×5) Conv	11.061
(7×7) Conv	13.308
(9×9) Conv	17.291
(11×11) Conv	20.286
(15×15) Conv	18.140

Table 1: First layer convolution size vs. performance

We follow up the first layer of convolutions with Rectified Linear Unit activations, then utilize $(2, 2)$ max-pooling to downsample. We then use (4×4) convolutions in the second convolutional unit + $(2, 2)$ max-pooling, and connect to 64 units then one final output.

4.0.2 Implementation and Training

Event generation and simulation was conducted on the SLAC `atlint` cluster. All Deep Learning experiments were conducted in Python with the Keras [22] Deep Learning library on the Stanford Institute for Computational and Mathematical Engineering GPU cluster, utilizing NVIDIA C2070 graphics cards.

We used 30 million training and 30 million testing samples, and trained networks using both the Adam [23] algorithm and Stochastic Gradient Descent with Nesterov Momentum [24]. We found that SGD+Nesterov outperformed Adam, and thus is used in all following facts and figures.

5 Studies

To begin understanding what a deep network can learn about jet topology, we choose a finite region of phase space, and standardize our comparisons. In an effort to define a standard way that physics object identification using machine learning should be conducted, we exactly define our procedure for comparisons. In particular, we restrict our studies to 250 GeV $\leq p_T \leq$ 300 GeV, and confine ourselves to a 65 GeV $\leq m \leq$ 95 GeV mass window, wholly containing the peak of the W.

We construct a scaffolded and multi-approach series of methodologies for understanding, visualizing, and validating neural networks within HEP.

5.1 Figure of Merit

As is commonly done in High Energy Physics, we eschew the commonly chosen metric of basic accuracy in favor of the Receiver Operating characteristic. This is because we must examine the entire spectrum of trade-off between Type-I and Type-II error, as many applications in physics will choose different points along the trade-off curve. We use a slight modification of the traditional ROC. For any discriminating variable, let c be a threshold on the likelihood ratio on that variable, and let w be the vector of weights over the entire evaluation sample. We define the *rejection* of such a threshold is defined as

$$\rho(c) = \frac{1}{\text{FPR}(c, w)},$$

where $\text{FPR}(c, w)$ is the weighted false positive rate for using c as a threshold.

We define the *efficiency* of c as

$$\varepsilon(c) = \text{TPR}(c, w),$$

where $\text{TPR}(c, w)$ is the weighted false positive rate for using c as a threshold. We then evaluate our algorithms using the area under the line generated by $\{(\varepsilon(c), \rho(c)) : \varepsilon(c) \in [0.2, 0.8]\}$. We say that an classifier is *strictly* more performant if the ROC curve is above a baseline for all efficiencies.

5.2 Coarse Studies

To a first order, the first desirable characteristic is a simple performance improvement over the standard physics-driven variables for discrimination. In particular, we compare our network to n-subjettiness [25] and the jet mass. We henceforth refer to n-subjettiness as τ_{21} for our purposes, as τ_2/τ_1 is relevant for our classification problem.

In Figure 5, we illustrate the performance gains of a deep neural network over both τ_{21} and the 2D likelihood of τ_{21} and jet mass.

We also provide a comparison to a 3D likelihood constructed on τ_{21}, jet mass, and the deep network output itself. We can gain a significant piece of insight from this. Note how in Figure 5 we can see that the DNN represents a large gain on a physics-only likelihood. However, when we explicitly include the physics variable in a 3D likelihood, we see a small but definitively non-zero performance gain. This implies that the performance boost *by definition* is getting its gain from something that is not *fully* encapsulated in τ_{21} and jet mass.

Though important on it's own, this figure of merit does little to help drive understanding in the context of HEP. Such an increase begs further questions – what is this gain, and where does it come from? Why is the DNN able to pick up on this?

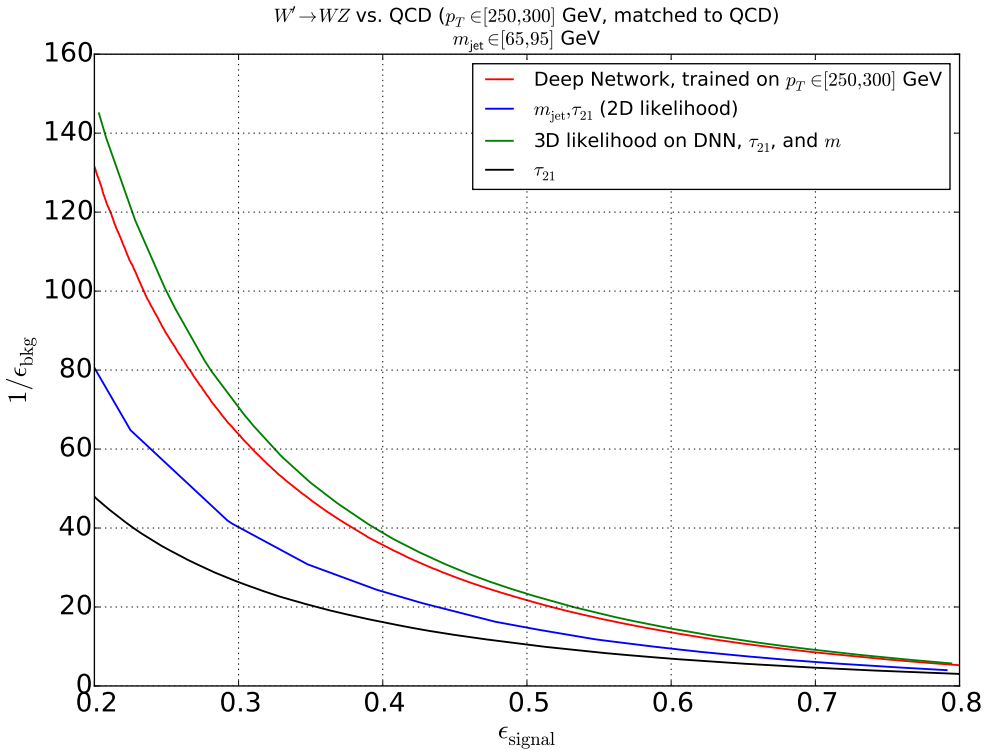


Figure 5: Receiver Operating Characteristic (ROC) over coarse sample

5.2.1 Understanding what is learned

In Figure 6, we first examine the 11×11 convolutional filters in the first layer and look for structure. In

In order to understand what we learn, we first take a look *inside* the deep network, and visualize features learned during training.

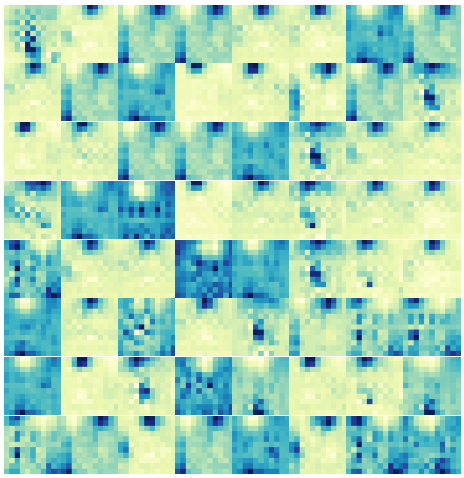
In Figure 6a, we show the (11×11) convolutional filters learned by our network. To mimic the operation in the first layer of the network, we can convolve each filter with an average jet image to get an understanding of what features the network learns at the first layer.

More formally, let $J_s = \frac{1}{n} \sum_{i:i \text{ is signal}} J^{(i)}$ and $J_b = \frac{1}{n} \sum_{i:i \text{ is background}} J^{(i)}$ represent the average signal and background jet over a sample, where $J^{(i)}$ is the ith jet image. In addition, we can select a filter $w_i \in \mathbb{R}^{11 \times 11}$ from the first convolutional layer.

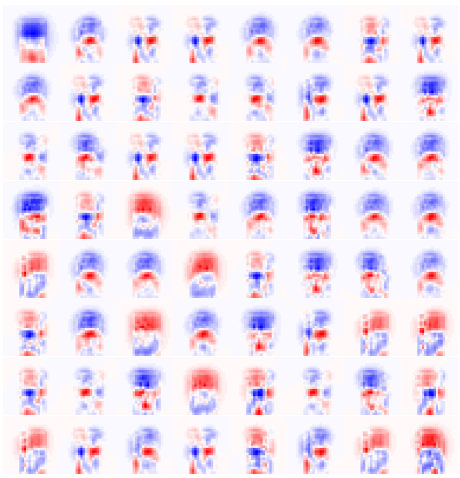
We then examine the differences in the post convolution layer. We take

$$J_s * w_i - J_b * w_i, \forall i, \tag{5.1}$$

where $*$ is the standard convolution operator. We arrange these new "convolved images" in a grid, and show in red regions where signal has a stronger representation, and in blue where background has a stronger representation. In Figure 6b, we show the convolved differences described above, where each (i, j) image is the representation under the (i, j) convolutional filter. We note the existence of interesting patterns around the regions where the leading and subleading subjets are expected to be.



(a) (11×11) convolutional kernels from first layer



(b) Convolved Jet Image differences

Figure 6: Convolutional Kernels (left), and convolved feature differences in jet images (right)

We also draw attention to the fact that there is a large diversity in the the convolved representations, indicating that the DNN is able to learn and pick up on multiple features that are descriptive.

5.2.2 Physics in Deep Representations

To get a tangible and more intuitive understanding of what jet structures a DNN learns, we construct the following. Let y be the DNN output, and consider every pixel $p_i j$ in transformed (η, ϕ) space. We the construct an image where each pixel (i, j) is $\rho_{p_{ij}, y}$, the Pearson Correlation Coefficient of that pixels energy deposition with the final DNN output. In Figure 7, we construct this image, and can see interesting structure in the subleading subjet region as well as asymmetric scattering patterns around the leading subjet.

5.3 Flat Hypercube Studies

Since we outperform physics-derived variables, we would like to know where these performance improvements come from. Our first approach to this is *hypercube reweighting*. In particular, we derive weights such that the joint distributions of mass, n-subjettiness, and p_T are non-discriminative. Specifically, we require

$$f(m, \tau_{21}, p_T | W' \to WZ) \approx f(m, \tau_{21}, p_T | QCD). \tag{5.2}$$

We then take our globally trained neural network (Figure 5) and apply the discriminant under this "flattening" transformation. We also use the training weights inside this window and train an additional CNN. In particular, we look for increases in performance, which would indicate information learned beyond physics variables since we removed the discrimination power using hypercube weighting.

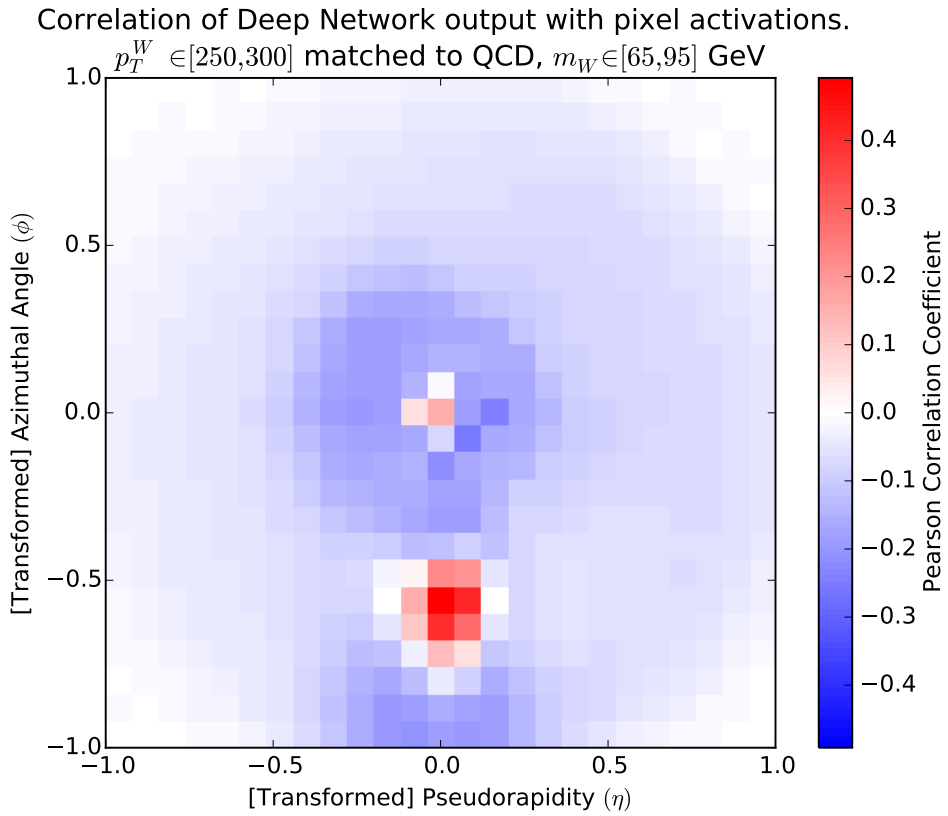


Figure 7: Per-pixel linear correlation with DNN output

In Figure 9 we show that the globally trained NN retains discrimination power even once we "subtract" the discrimination power of primitives from physics.

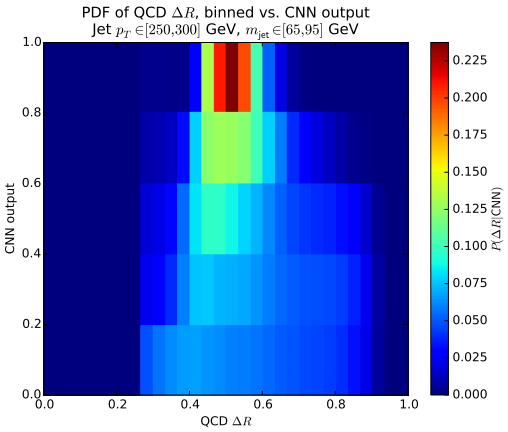
5.4 Small Window Studies

Though we see performance improvements using a deep network under an induced flat hypercube, we want to be sure that these performance improvements are valid and carry over to a less contrived distribution. In particular, we consider a mass window of [79, 81] GeV, a p_T window of [250, 255] GeV, and we require $\tau_2 1$ to be in [0.19, 0.21]. In Figure 10, we see the differences between signal and background for such a window.

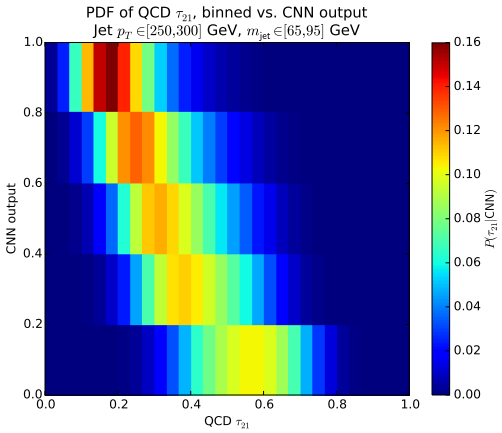
In this window, we compare the DNN trained outside the window to a Fisher Linear Discriminant trained inside the window. In Figure 11 we see this performance comparison, and note that our DNN outperforms the FLD.

5.4.1 Understanding what we learn

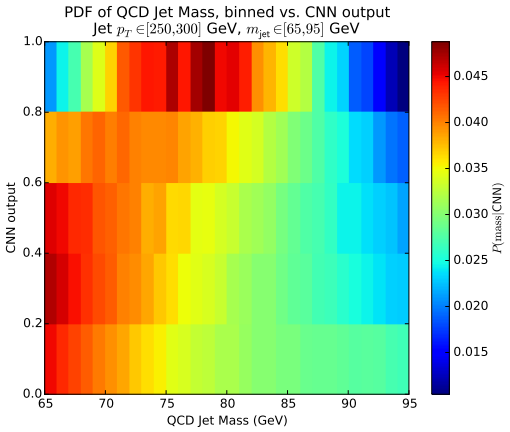
In Figure 5.4.1, we show the same feature representations as in Figure 6b, which show the convolved differences in images over signal and background in the window.



(a) Sculpted QCD ΔR distribution



(b) Sculpted QCD τ_{21} distribution



(c) Sculpted QCD mass distribution

Figure 8: Sculpted QCD distributions

6 Acknowledgements

This work is supported by the US Department of Energy (DOE) Early Career Research Program and grant DE-AC02-76SF00515. BN is supported by the NSF Graduate Research Fellowship under Grant No. DGE-4747 and by the Stanford Graduate Fellowship. SDSI?

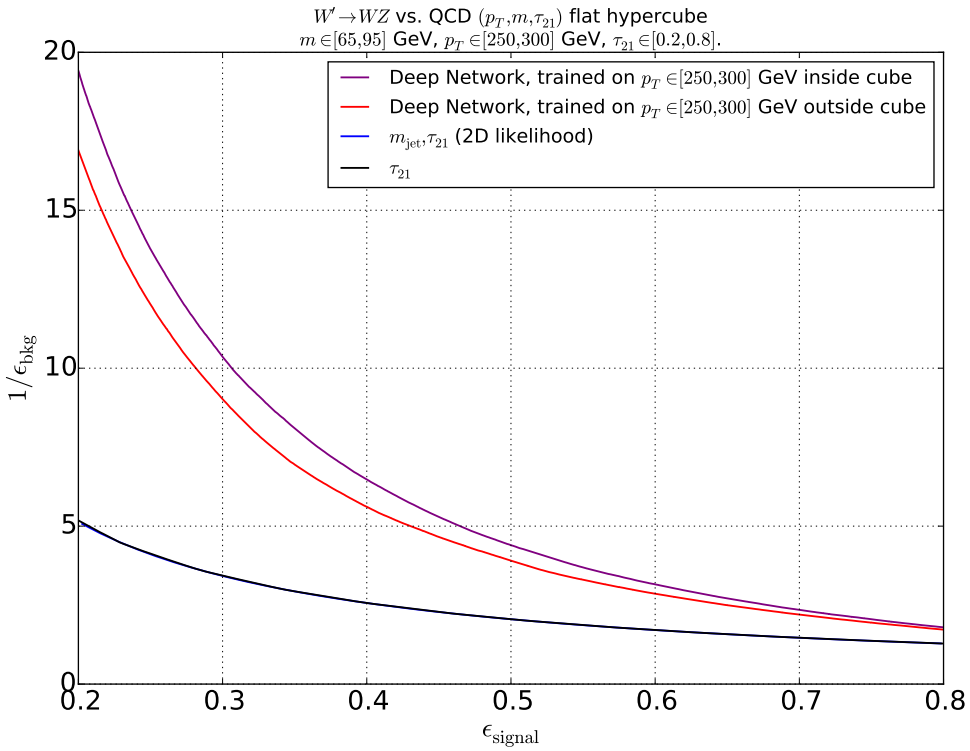
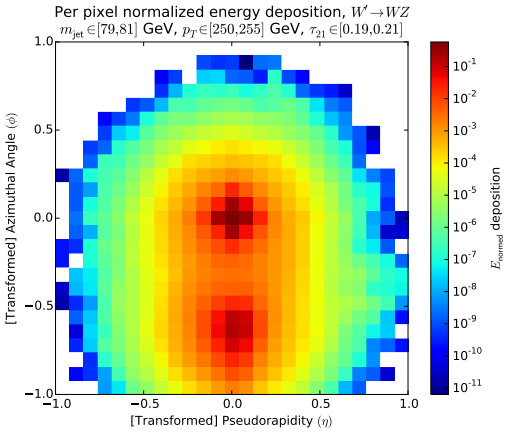
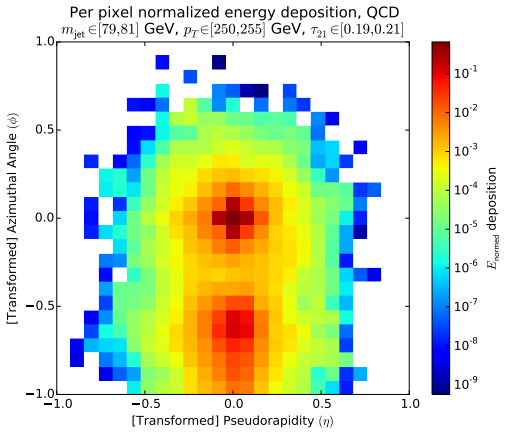


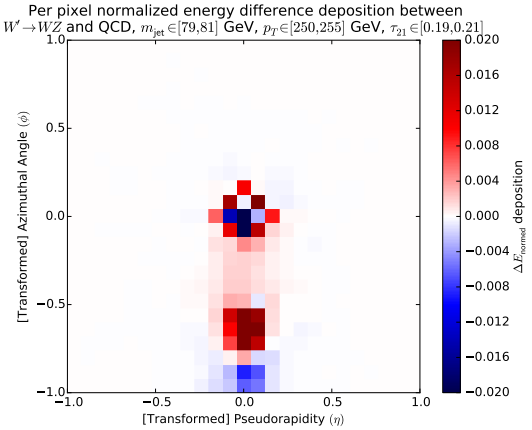
Figure 9: ROC Curve for weigth-flattened hypercube, with $m \in [65, 95]$GeV, $p_T \in [250, 300]$GeV, and $\tau_{21} \in [0.2, 0.8]$



(a) Average $W' \to WZ$ image



(b) Average QCD image



(c) Average image difference

Figure 10: $W' \to WZ$ (left) and QCD (right) average jet-images, and Signal - Background image difference (bottom)

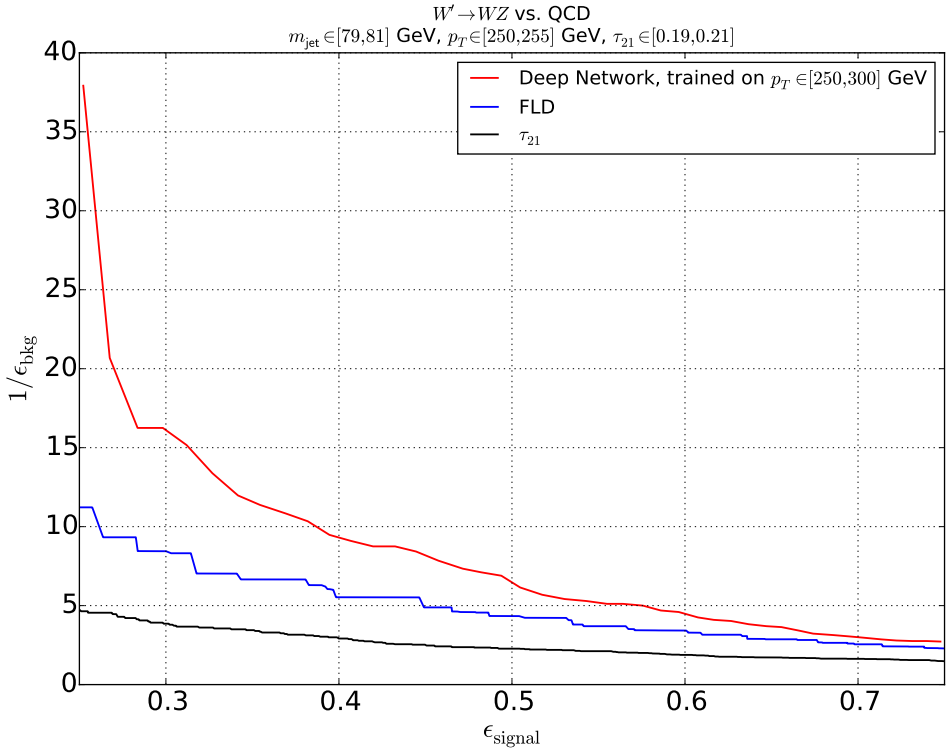


Figure 11: Receiver Operating Characteristic (ROC) over window sample

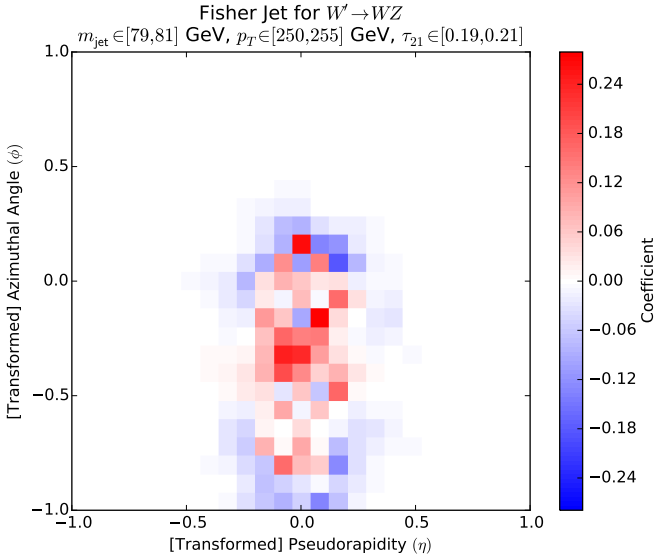


Figure 12: Cell coefficients from Fisher Linear Discriminant in window: $m_{\text{jet}} \in [79, 81]$ GeV, $p_T \in [250, 255]$ GeV, $\tau_{21} \in [0.19, 0.21]$

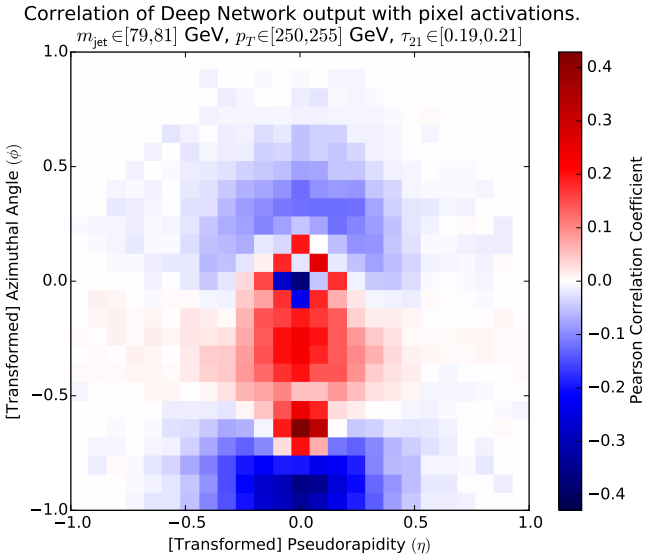


Figure 13: Pearson Correlation Coefficient for pixels vs. DNN output, $m_{\text{jet}} \in [79, 81]$ GeV, $p_T \in [250, 255]$ GeV, $\tau_{21} \in [0.19, 0.21]$

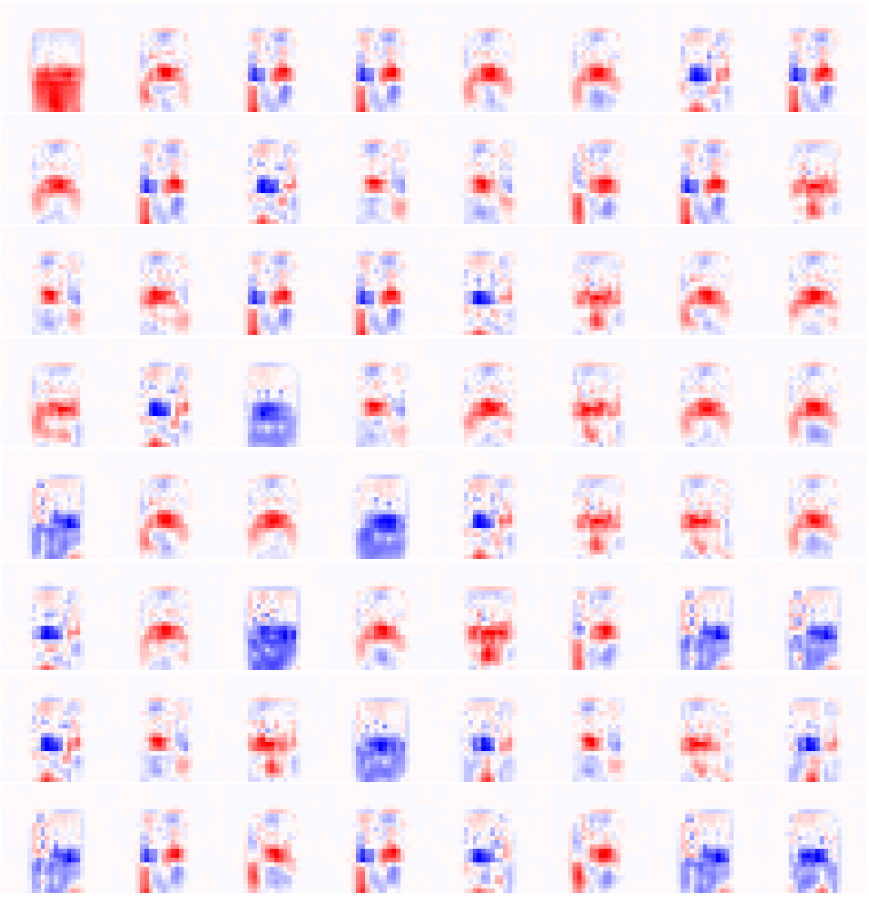


Figure 14: Convolved Feature Differences in jet images, jet $\in [79, 81]$ GeV, $p_T \in [250, 255]$ GeV, $\tau_{21} \in [0.19, 0.21]$

References

[1] **CMS** Collaboration, V. Khachatryan *et. al.*, *Identification techniques for highly boosted W bosons that decay into hadrons*, *JHEP* **12** (2014) 017 [1410.4227].

[2] *Identification of boosted, hadronically-decaying W and Z bosons in $\sqrt{s} = 13$ TeV Monte Carlo Simulations for ATLAS*, Tech. Rep. ATL-PHYS-PUB-2015-033, CERN, Geneva, Aug, 2015.

[3] *Performance of Boosted W Boson Identification with the ATLAS Detector*, Tech. Rep. ATL-PHYS-PUB-2014-004, CERN, Geneva, March, 2014.

[4] **ATLAS** Collaboration, G. Aad *et. al.*, *Search for high-mass diboson resonances with boson-tagged jets in proton-proton collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector*, 1506.00962.

[5] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for massive resonances in dijet systems containing jets tagged as W or Z boson decays in pp collisions at $\sqrt{s} = 8$ TeV*, *JHEP* **08** (2014) 173 [1405.1994].

[6] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for the production of an excited bottom quark decaying to tW in proton-proton collisions at $\sqrt{s} = 8$ TeV*, 1509.08141.

[7] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for vector-like charge 2/3 T quarks in proton-proton collisions at $\sqrt{s} = 8$ TeV*, 1509.04177.

[8] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for pair-produced vector-like B quarks in proton-proton collisions at $\sqrt{s} = 8$ TeV*, 1507.07129.

[9] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for A Massive Resonance Decaying into a Higgs Boson and a W or Z Boson in Hadronic Final States in Proton-Proton Collisions at $\sqrt{s} = 8$ TeV*, 1506.01443.

[10] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for a Higgs Boson in the Mass Range from 145 to 1000 GeV Decaying to a Pair of W or Z Bosons*, 1504.00936.

[11] **CMS** Collaboration, V. Khachatryan *et. al.*, *Search for Narrow High-Mass Resonances in Proton–Proton Collisions at $\sqrt{s} = 8$ TeV Decaying to a Z and a Higgs Boson*, *Phys. Lett.* **B748** (2015) 255–277 [1502.04994].

[12] **ATLAS** Collaboration, G. Aad *et. al.*, *Search for squarks and gluinos with the ATLAS detector in final states with jets and missing transverse momentum using $\sqrt{s} = 8$ TeV proton–proton collision data*, *JHEP* **09** (2014) 176 [1405.7875].

[13] **ATLAS** Collaboration, G. Aad *et. al.*, *Search for a high-mass Higgs boson decaying to a W boson pair in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector*, 1509.00389.

[14] **ATLAS** Collaboration, G. Aad *et. al.*, *Search for an additional, heavy Higgs boson in the $H \to ZZ$ decay channel at $\sqrt{s} = 8$ TeV in pp collision data with the ATLAS detector*, 1507.05930.

[15] **ATLAS** Collaboration, G. Aad *et. al.*, *Search for production of WW/WZ resonances decaying to a lepton, neutrino and jets in pp collisions at $\sqrt{s} = 8$ TeV with the ATLAS detector*, *Eur. Phys. J.* **C75** (2015), no. 5 209 [1503.04677]. [Erratum: Eur. Phys. J.C75,370(2015)].

[16] **ATLAS** Collaboration, G. Aad *et. al.*, *Measurement of the cross-section of high transverse momentum vector bosons reconstructed as single jets and studies of jet substructure in pp collisions at $\sqrt{s} = 7$ TeV with the ATLAS detector*, *New J. Phys.* **16** (2014), no. 11 113013 [1407.0800].

[17] T. Sjostrand, S. Mrenna and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852–867 [0710.3820].

[18] T. Sjostrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **0605** (2006) 026 [hep-ph/0603175].

[19] M. Cacciari, G. P. Salam and G. Soyez, *The Anti-k(t) jet clustering algorithm*, *JHEP* **0804** (2008) 063.

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

[21] D. Krohn, J. Thaler and L.-T. Wang, *Jet Trimming*, *JHEP* **1002** (2010) 084 [0912.1342].

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

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

[24] Y. Nesterov, *A method of solving a convex programming problem with convergence rate O(1/sqr(k))*, *Soviet Mathematics Doklady* **27** (1983) 372–376.

[25] J. Thaler and K. Van Tilburg, *Identifying Boosted Objects with N-subjettiness*, *JHEP* **1103** (2011) 015 [1011.2268].