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

The fundamental challenge at the energy frontier of particle physics is identifying subtle signals beneath enormous backgrounds. A monumental triumph was the recent discovery of the Higgs boson - a particle that is produced one in every 10^{10} collisions at the Large Hadron Collider (LHC). Machine learning techniques have played a key role in all aspects of this search. At the most basic level, charged particle tracks are reconstructed using pattern recognition techniques, higher level objects, blah blah.

Techniques built on physically motivated features have succesfully probed the highest energies and smallest scales ever achieved in terestrial experiments. At the same time, there have been significant gains

The high particle multiplicity final states of hadron collider events Something like "we have figured out many cool variables" "machine learning can find variables" "lets use one to inform/improve the other!"

Places where ML is used in HEP: TMVA [1]

b-tagging: MV1 (ATLAS) and CSV (CMS) tau-tagging NP searches rare SM measurements (e.g. single top) Higgs

Jet Images paper: http://arxiv.org/abs/1407.5675

Let's keep a running document with all pub-ready plots in here.

We can even use this later on when we actually write the paper – this hosting allows templates, styles, etc. and we can all edit, which is great.

If you guys have any tips on plotting style, let's keep that here too – as I make graphics now, I want them to be pub ready. You definitely know what reviewers go for more than I do.

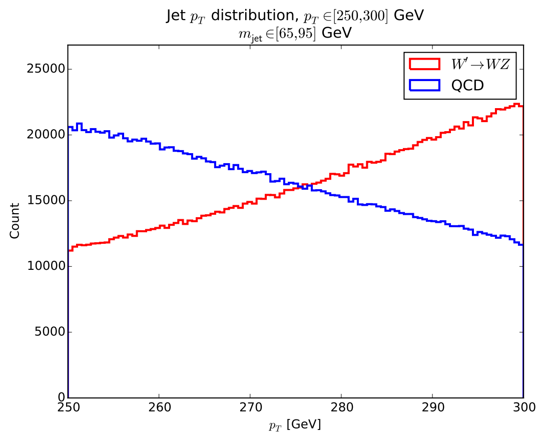
2 Simulation Details

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 task is to distinguish the hadronic decays of high p_T W bosons from generic SM background processes computes of quarks and gluons. To simulate high p_T W bosons, a hypothetical W' boson is generated and forced to decay to a hadronically decaying W boson and a Z boson which decays invisibly.

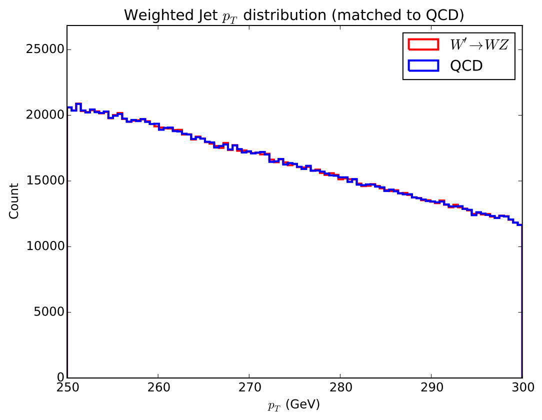
Two physics processes are generated using Pythia 8.170 [2, 3] at $\sqrt{s} = 14$ TeV. To simulate high p_T hadronic W decays, W' bosons are generated to decay exclusively into a $W(\to qq')$ and $Z(\to \nu\nu)$. The p_T scale of the hadronically decaying W is set by the mass of the W' which is tuned to 600 and 800 GeV for this study so that the 200 GeV $\lesssim p_T^W \lesssim$ 400 GeV. In this p_T^W range, the W decay products are expected to merge within a cone of R 1.0 where $\Delta R^2 = \Delta\phi^2 + \Delta\eta^2 \sim 4m_W^2/p_{T,W}^2$. To study the impact on signal versus background, QCD dijets are generated with a range of $\hat{p}_T$ that is approximately in the same range as the relevant signal process. Anti-k_t jets are clustered using FastJet [4] 3.0.3. The signal processes are chosen such that jets with radius parameter $R = 1$ are most appropriate in capturing the decay products of the heavy particles. The anti-k_t jets are trimmed [5] 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 (y, ϕ) extends out to $y = 5.0$. 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

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

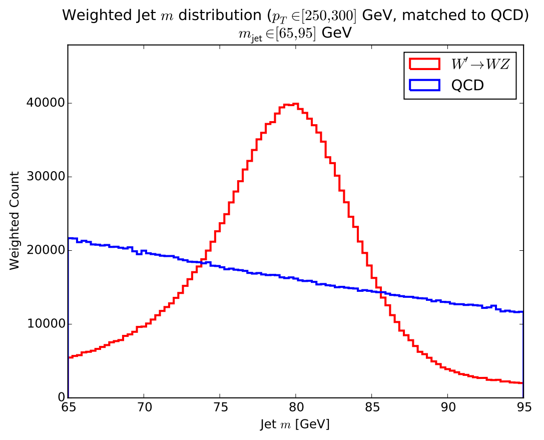


(a) Unweighted p_T distribution

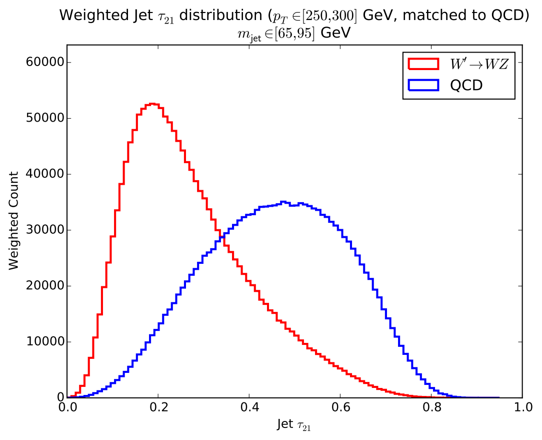


(b) Weighted p_T distribution

Figure 1: Jets originating from the $W' \to WZ$ decay are re-weighted such that their p_T spectrum matches that of QCD jets



(a) Weighted jet mass distribution

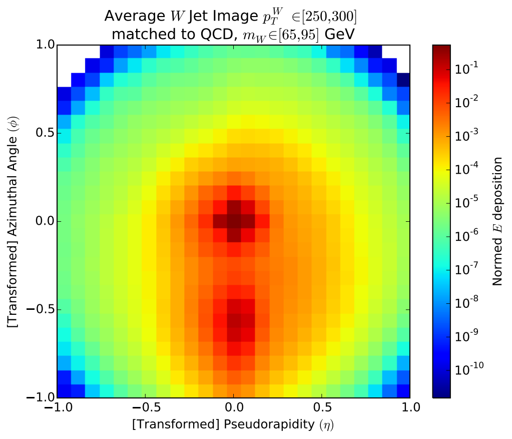


(b) Weighted τ_{21} distribution

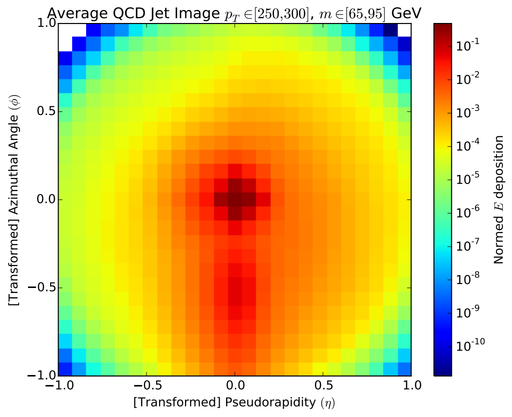
Figure 2: Weighted mass (left) and n-subjettiness (right) of samples, with $W' \to WZ$ decays in red and QCD jets in blue.

3 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



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



(b) Average weighted QCD image

Figure 3: Weighted $W' \to WZ$ (left) and QCD (right) average jet-image

the vector of weights over the entire evaluation sample. We define the *rejection* of such a threshold is defined as

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

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

We define the *efficiency* of c as

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

where $\mathrm{TPR}(c, w)$ 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.

4 Deep Learning

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, do not investigate large network architectures, rather we focus on understanding what information and higher level representations 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.

5 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.

5.0.1 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

$$[\texttt{Dropout} \rightarrow \texttt{Conv} \rightarrow \texttt{ReLU} \rightarrow \texttt{MaxPool}] * 2 \rightarrow [\texttt{Dropout} \rightarrow \texttt{FC} \rightarrow \texttt{ReLU}] * 2 \rightarrow \texttt{Sigmoid}. \tag{5.1}$$

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 3.

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.

5.0.2 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 [6] 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 [7] algorithm and Stochastic Gradient Descent with Nesterov Momentum [8]. We found that SGD+Nesterov outperformed Adam, and thus is used in all following facts and figures.

6 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.

6.1 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 [9] 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 4, 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 4 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?

6.1.1 Understanding what is learned

In Figure 5, 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.

Blah... linear correlations with pixels

6.1.2 Physics in Deep Representations

6.2 Flat Hypercube Studies

Here, we see the ROC blah...

6.3 Small Window Studies

Performance inside the window, use a Fisher Discriminant...

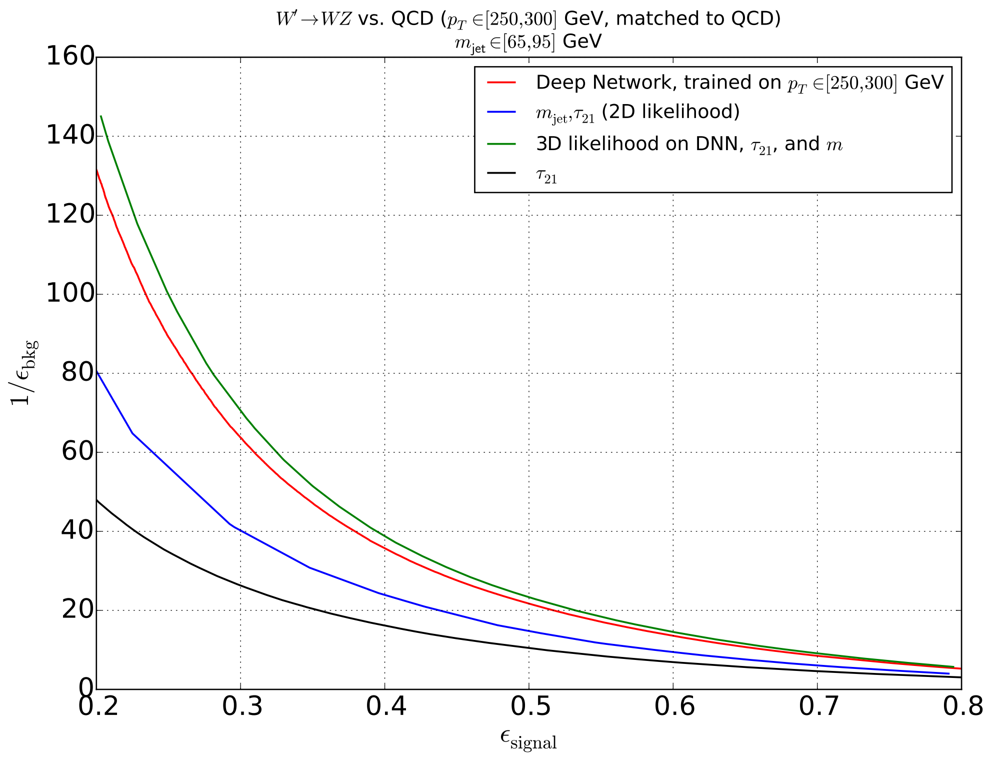


Figure 4: Receiver Operating Characteristic (ROC) over coarse sample

6.3.1 Understanding what we learn

7 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?

References

[1] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne and H. Voss, *TMVA: Toolkit for Multivariate Data Analysis*, *PoS* **ACAT** (2007) 040 [physics/0703039].

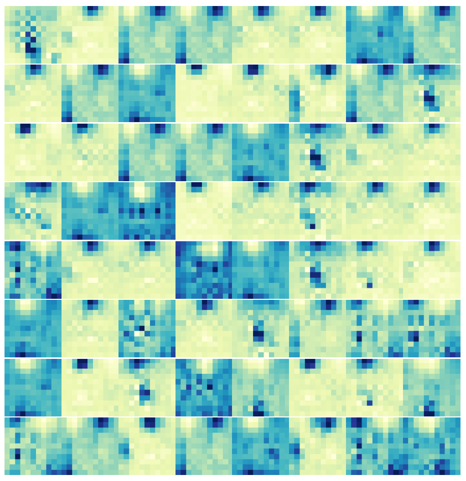
[2] T. Sjostrand, S. Mrenna and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852–867 [0710.3820].

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

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

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

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


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

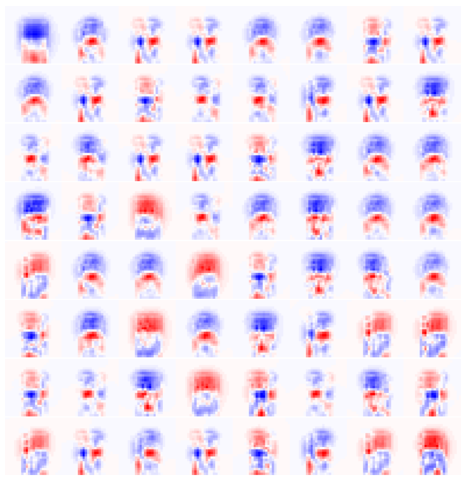

(b) Convolved Jet Image differences

Figure 5: Convolutional Kernels (left), and convolved feature differences in jet images (right)

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

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

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

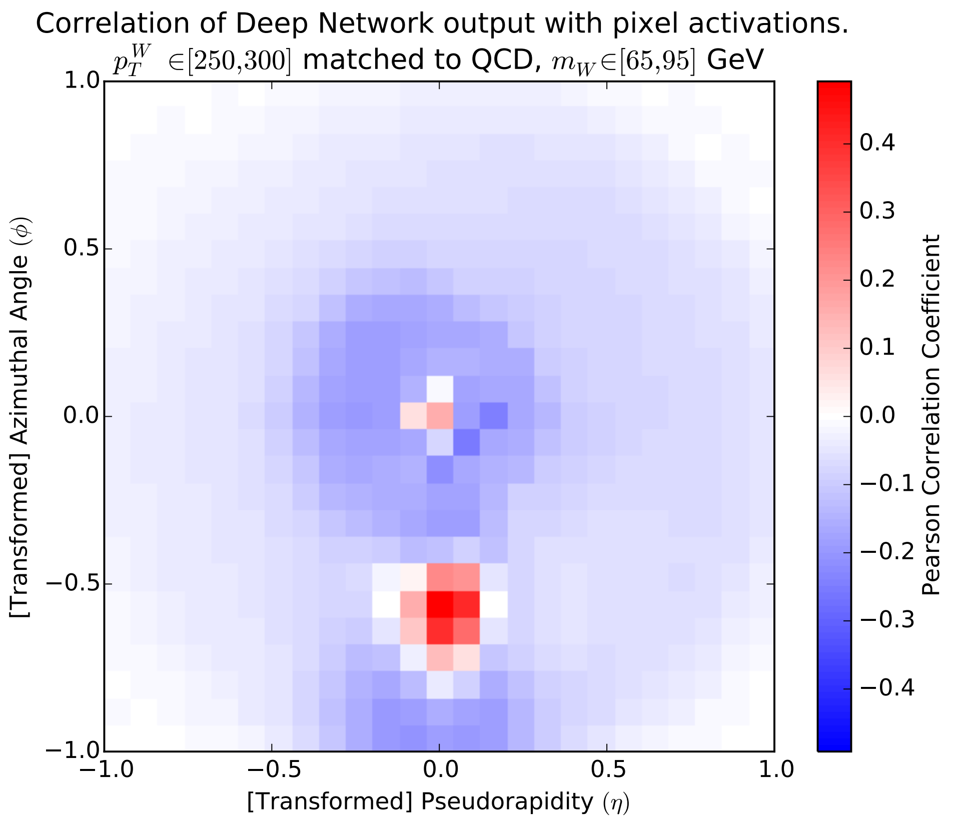
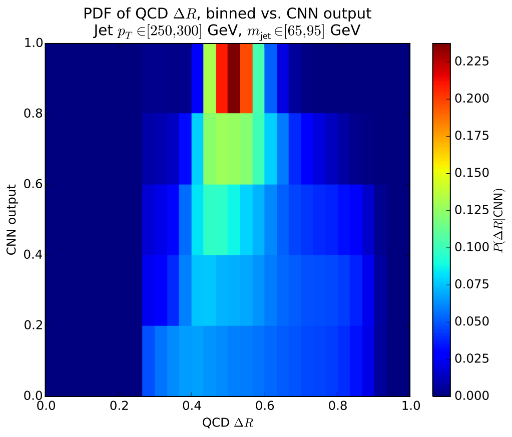
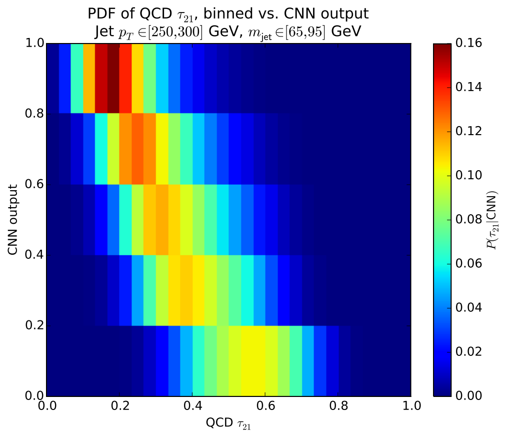


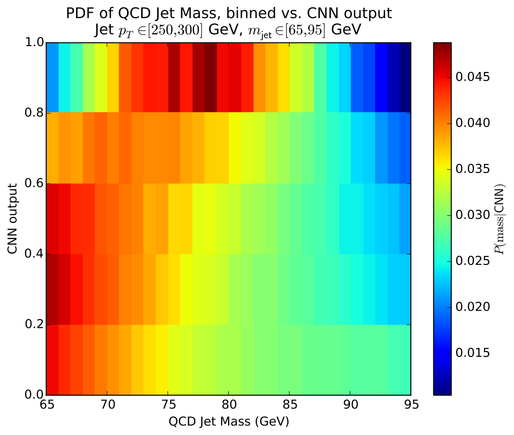
Figure 6: Per-pixel linear correlation with DNN output



(a) Sculpted QCD ΔR distribution



(b) Sculpted QCD τ_{21} distribution



(c) Sculpted QCD mass distribution

Figure 7: Sculpted QCD distributions

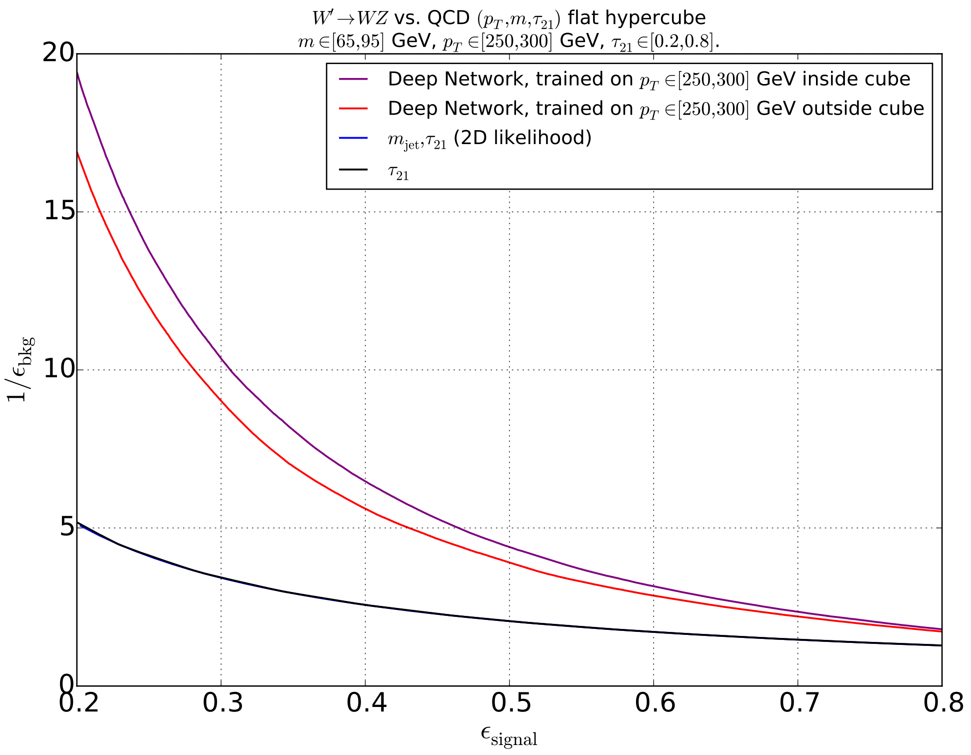
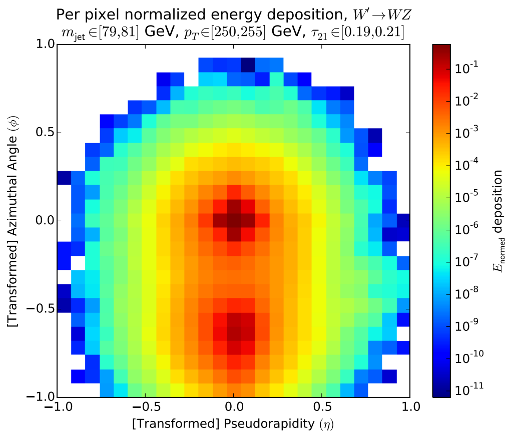
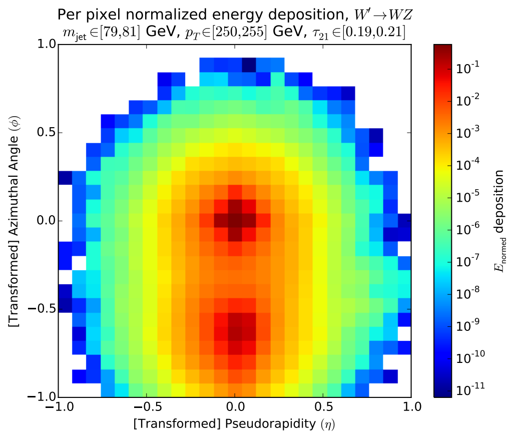


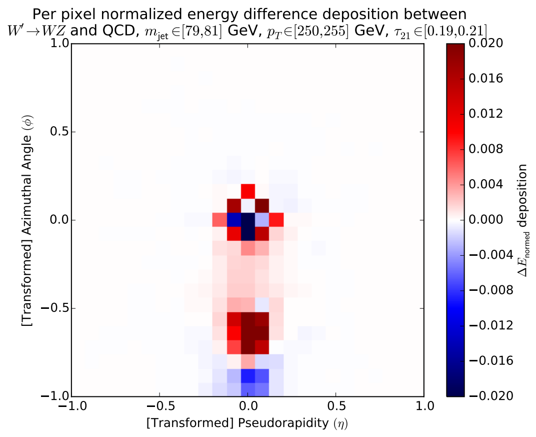
Figure 8: ROC Curve for weigth-flattened hypercube, with $m \in [65, 95]$GeV, $p_T \in [250, 300]$GeV, and $\tau_{21} \in [0.2, 0.8]$



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



(b) Average QCD image



(c) Average image difference

Figure 9: $W' \to WZ$ (left) and QCD (right) average jet-images, and Signal - Background image difference (bottom)

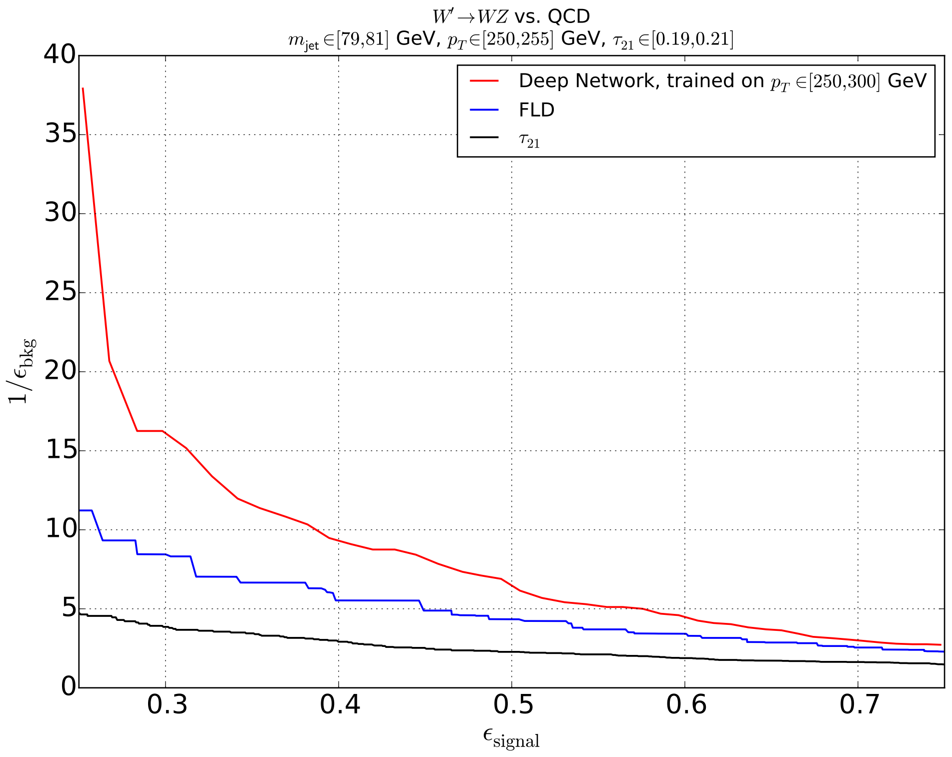


Figure 10: Receiver Operating Characteristic (ROC) over window sample

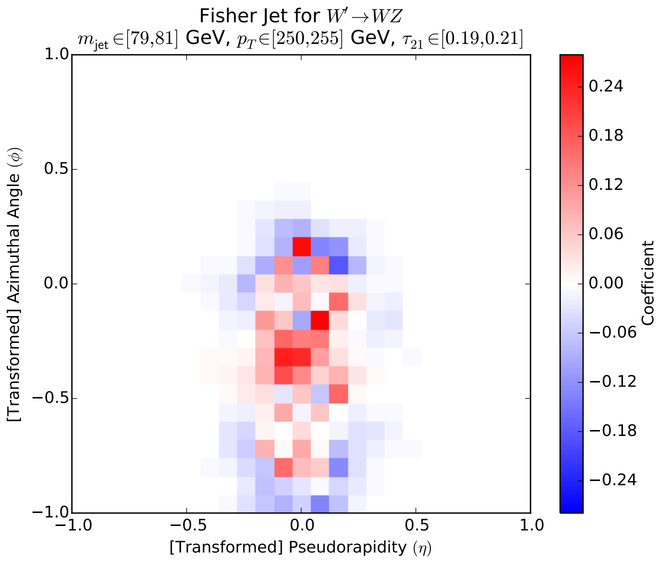


Figure 11: caption

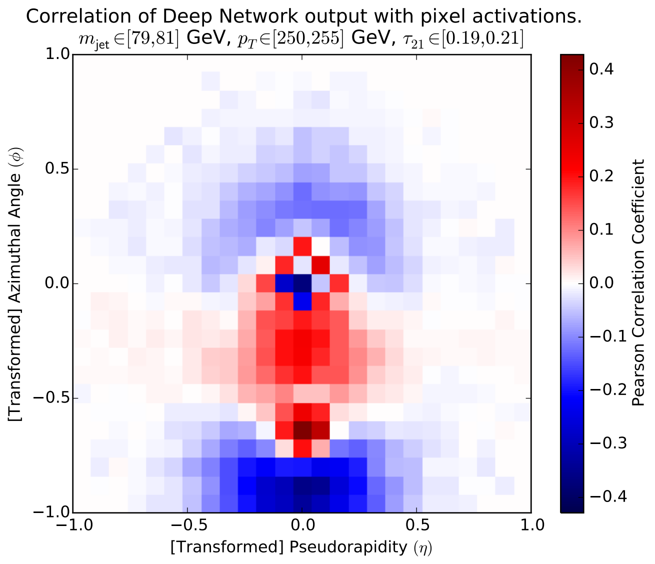


Figure 12: caption

Figure 13: caption