

# Jet-Images – Deep Learning Edition

---

Luke de Oliveira,<sup>a</sup> Michael Kagan,<sup>b</sup> Lester Mackey,<sup>c</sup> Benjamin Nachman,<sup>b</sup> and Ariel Schwartzman<sup>b</sup>

<sup>a</sup> Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305, USA

<sup>b</sup> SLAC National Accelerator Laboratory, Stanford University, 2575 Sand Hill Rd, Menlo Park, CA 94025, U.S.A.

<sup>c</sup> Department of Statistics, Stanford University, Stanford, CA 94305, USA

E-mail: [lukedeo@stanford.edu](mailto:lukedeo@stanford.edu), [mkagan@cern.ch](mailto:mkagan@cern.ch), [lmackey@stanford.edu](mailto:lmackey@stanford.edu), [bnachman@cern.ch](mailto:bnachman@cern.ch), [sch@slac.stanford.edu](mailto: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, or jets, it measures as an image, we investigate the potential of machine learning techniques based on deep learning architectures to identify highly boosted  $W$  bosons. 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 how these features are learned by the network and what additional information is used to improve performance. This interplay between physically-motivated feature driven tools and supervised learning algorithms is general and can be used to significantly increase the sensitivity to discover new particles and new forces, and gain a deeper understanding of the physics within jets.

## 1 Introduction

Collimated sprays of particles, called *jets*, resulting from the production of high energy quarks and gluons provide an important handle to search for signs of physics beyond the Standard Model (SM) at the Large Hadron Collider (LHC). In many extensions of the SM, there are new, heavy particles that decay to heavy SM particles such as  $W$ ,  $Z$ , and Higgs bosons as well as top quarks. As is often the case, the mass of the SM particles is much smaller than the mass of the new particles and so they are imparted with a large Lorentz boost. As a result, the SM particles from the boosted boson and top quark decays are highly collimated in the lab frame and may be captured by a single jet. Classifying the origin of these jets and differentiating them from the overwhelming Quantum Chromodynamic (QCD) multijet background is a fundamental challenge for searches with jets at the LHC. Jets from boosted bosons and top quarks have a rich internal substructure. There is a wealth of literature addressing the topic of jet tagging by designing physics-inspired features to exploit the jet substructure (see e.g. Ref. [2–4]). However, in this paper we address the challenge of jet tagging through the use of Machine Learning (ML) and Computer Vision (CV) techniques combined with low-level information, rather than directly using physics inspired features. In doing so, we not only improve discrimination power, but also gain new insight into the underlying physical processes that provide discrimination power by extracting information learned by such ML algorithms.

The analysis presented here is an extension of the jet-images approach, first introduced in Ref. [5] and then also studied with similar approaches by Ref. [6], 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 pre-processing techniques based on the underlying physics symmetries of the jets were combined with a linear Fisher discriminant 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 DNNs are significantly more complex than Fisher discriminants, they also provide the capability to learn rich high-level representations of jet images and to greatly enhance discrimination power. By developing techniques to access this rich information, we can explore and understand what has been learned by the DNN and subsequently improve our understanding of 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.

Automatic feature extraction and high-level learned feature representations via deep learning have led to state-of-the-art performance in Computer Vision [7–9]. The focus of this work is on robust networks architectures to investigate what information and higher level representations a fully-connected multi-layer network and a convolutional neural network learn about jets. There will be a focus on connecting the gains in performance with the underlying physical properties of jets through visualization. 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, including new insights into the relationship with underlying physics information, is discussed in Section 3. We then introduce the deep neural network architectures that we use in Section 4. The discrimination performance and the exploration of the information learned by the DNNs 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<sup>1</sup> [10–12] and used in many analyses [13–25].

To simulate highly boosted  $W$  bosons, a hypothetical  $W'$  boson is generated and forced to decay to a hadronically decaying  $W$  boson ( $W \rightarrow qq'$ ) and a  $Z$  boson which decays invisibly ( $Z \rightarrow \nu\bar{\nu}$ ). 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 [26, 27] at  $\sqrt{s} = 14$  TeV. The minimum 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:  $250 \text{ GeV} < p_{T,W} < 300 \text{ GeV}$ . This corresponds to an angular spread of about  $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2} \sim 0.6$ , where  $\Delta\eta$  and  $\Delta\phi$  are the distances between  $W$  boson decay products in  $(\eta, \phi)$  coordinates. The decay products of the  $W$  bosons as well as the background are clustered into jets using the anti- $k_t$  algorithm [28] via FASTJET [29] 3.0.3. To mitigate the contribution from the underlying event, jets are trimmed [30] by re-clustering the constituents into  $R = 0.3$   $k_t$  subjets and dropping those which have  $p_T^{\text{subjet}} < 0.05 \times p_T^{\text{jet}}$ . Trimming also reduces the impact of multiple proton-proton collisions occurring in the same event as the hard-scatter process (pileup). We leave investigation of the robustness of the neural network performance to pileup for future studies.

Three key jet features for distinguishing between  $W$  jets and QCD jets are the *jet mass*, *nsubjettiness* [31] and the distance in  $(\eta, \phi)$  space between subjets of the trimmed jet ( $\Delta R$ ). The distributions of these three discriminating variables are shown in Fig. 1. The jet mass is defined as  $m_{\text{jet}}^2 = \sum_{i,j} p_i p_j$ , with jet constituent four-vectors  $p_i$ , and is a proxy for the boson mass in the case of  $W$  boson events. In the case of QCD background jets, the jet mass scales with the transverse momentum and the size of the jet.  $N$ -subjettiness, in the form of  $\tau_{21}$ , is a measure of the likelihood that the jet has two hard prongs instead of one hard prong. In this application, the winner-takes-all axis [32] is used to define the axis in the  $\tau_{21}$  calculation. One other useful feature is the jet transverse momentum. However, since many of the other features have a strong dependence on the jet transverse momentum, we re-weight the signal so have the same  $p_T$  distribution as the background.

To model the discretization and finite acceptance of a real detector, a calorimeter of towers with size  $0.1 \times 0.1$  in  $(\eta, \phi)$  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

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

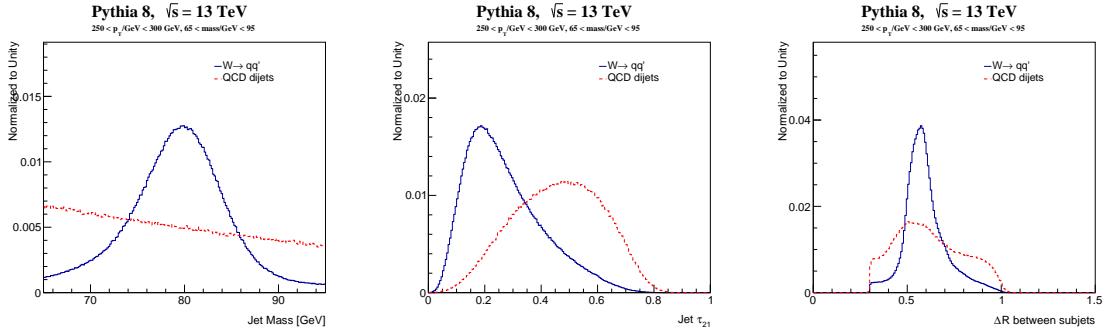
where  $E_i$  is the energy of particle  $i$  and the center of the tower  $j$  is  $(\eta_j, \phi_j)$ . Towers are treated as massless.

A *jet image* is formed by taking the constituents of a jet and discretizing its energy into pixels in  $(\eta, \phi)$ , with the intensity of each pixel given by the sum of the energy of all constituents of the jet inside that  $(\eta, \phi)$  pixel. We also investigate the use of the transverse projection of the energy in each tower as the pixel intensity. In our studies, we take the jet image pixelation to match the simulated

---

<sup>1</sup>There is also an extensive literature on phenomenological studies - see references within the experimental papers.

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.



**Figure 1:** The distributions of the jet mass (top left),  $\tau_{21}$  (top right) and the  $\Delta R$  between subjets (bottom) for signal (blue) and background (red) jets.

### 3 Pre-processing and the Symmetries of Space-time

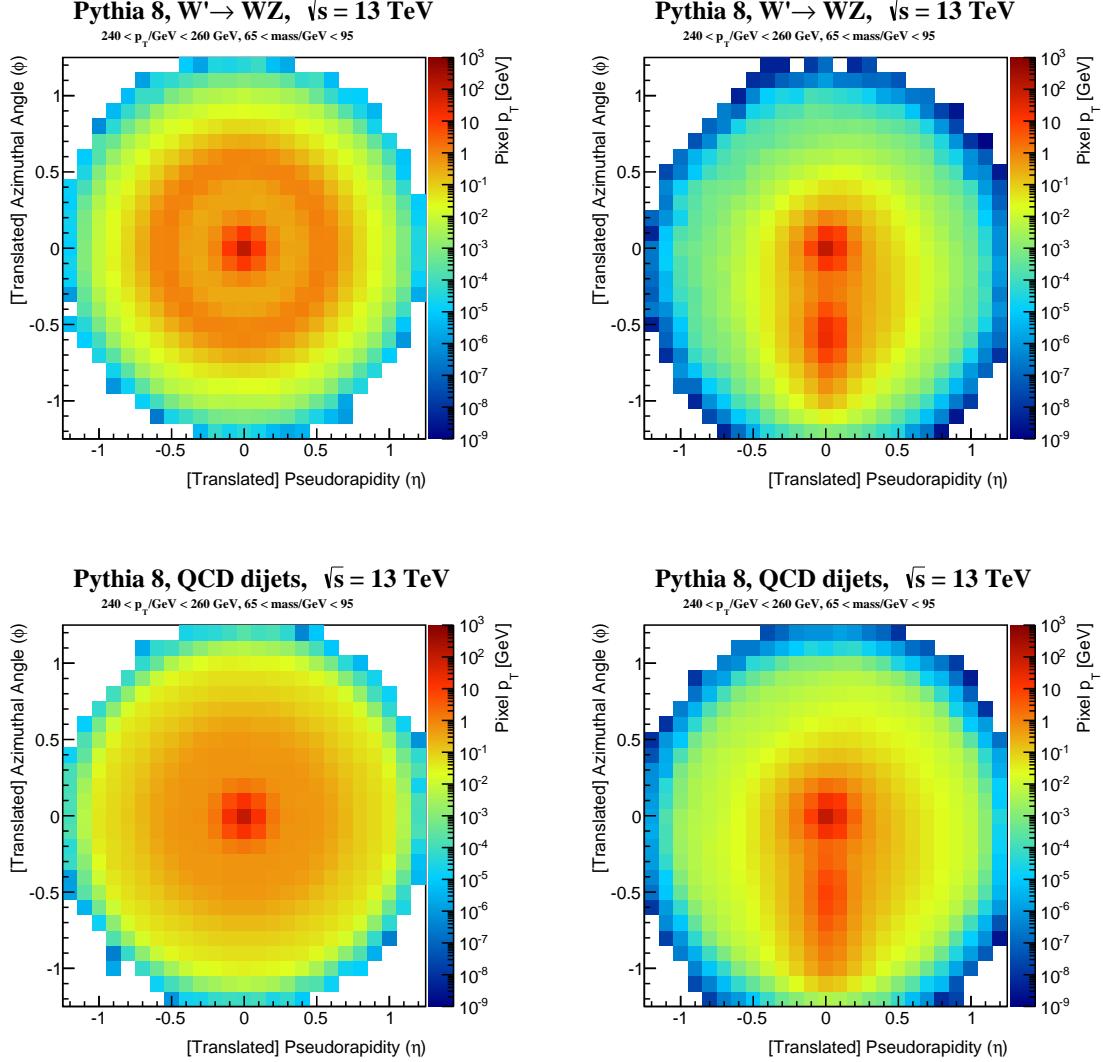
In order for the machine learning algorithms to most efficiently learn discriminating features between signal and background and to not learn the symmetries of space-time, the jet images are pre-processed. This procedure can greatly improve performance and reduce the required size of the sample used 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  $\phi$  are rotations around the  $z$ -axis and so the pixel intensity is unchanged by this operation. On the other hand, translations in  $\eta$  are *Lorentz boosts* along the  $z$ -axis, which do not preserve the pixel intensity. Therefore, a proper translation in  $\eta$  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 the transverse energy  $p_{T,i} = E_i / \cosh(\eta_i)$ . This new definition of intensity is invariant under translations in  $\eta$  and is used exclusively for the rest of this paper<sup>2</sup>.

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 aligned along the vertical axis. 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 cubic spline interpolation is used in this case - see Ref. [5] 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 parity flip steps of the pre-processing. The more pronounced second-subjet can already be observed 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

<sup>2</sup>Transverse energy based pixel intensity was used in the original Jet-Images paper [5]

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.



**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  $240 \text{ GeV} < p_T < 260 \text{ GeV}$  and  $65 \text{ GeV} < \text{mass} < 95 \text{ GeV}$ .

One standard pre-processing step that is often additionally applied in Computer Vision tasks 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, and when there is large pixel intensity variations between images. In this study, the jet transverse momenta are all around 250 GeV, but this can be spread amongst many pixels or concentrated in only a few. 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  $\Delta R$  between subjets in  $(\eta, \phi)$  is invariant under all of the pre-processing steps as well as normalization. However, consider the *image mass*,

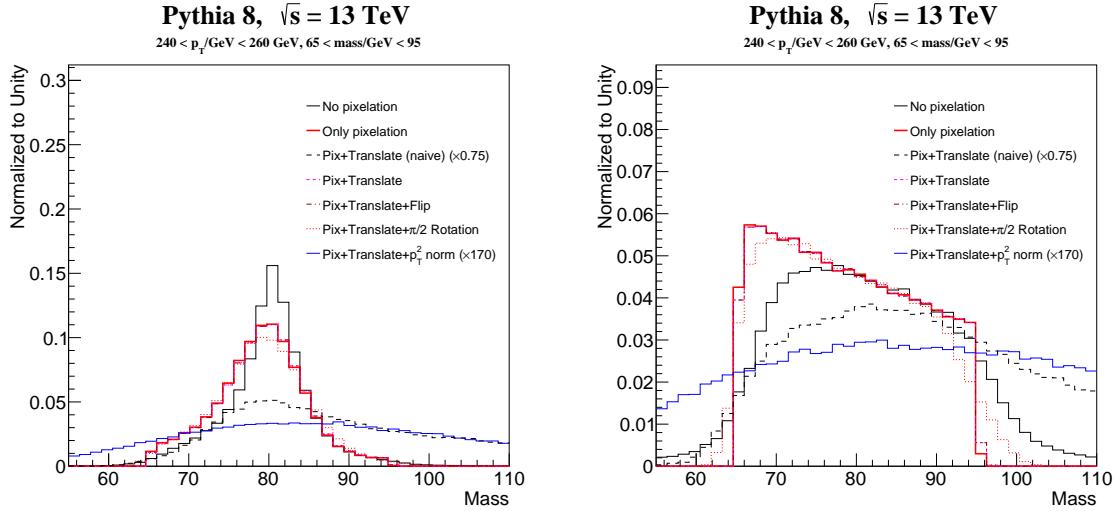
$$m_I^2 = \sum_{i < j} E_i E_j (1 - \cos(\theta_{ij})), \quad (3.1)$$

where  $E_i = I_i / \cosh(\eta_i)$  for pixel intensity  $I_i$  and  $\theta_{ij}$  is the angle between massless four-vectors with  $\eta$  and  $\phi$  at the  $i$  and  $j$  pixel centers. The image mass is not invariant under all pre-processing steps but does encode key information to identify highly boosted bosons that would ideally be preserved by the pre-processing step. As discussed earlier, with the proper choice of pixel intensity, translations preserve the image mass since it is a Lorentz invariant quantity. However, the rotation pre-processing step does not preserve the image mass. To understand this effect, 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  $p^\mu$  is at the center of the jet image coordinates and the vector  $q^\mu$  is located at  $\pi/2$  degrees. If we rotate the image around the jet axis so that the vector  $q^\mu$  is at 0 degrees, akin to rotating the jet image so that the sub-leading subjet goes from  $\pi/2$  to 0, then  $p^\mu$  is unchanged but  $q^\mu \rightarrow (1, 0, \sinh(1), \cosh(1))$ . The new invariant mass of  $q^\mu$  and  $p^\mu$  is about 1, which is reduced from its original value of  $\sqrt{2}$ . The parity inversion 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 variation in the normalizations induces a smearing in the jet-image mass distribution.

The impact of the various stages of pre-processing on the image mass are illustrated in Fig. 3. The finite segmentation of the simulated detector slightly degrades the jet mass resolution, but the translation and parity inversion (flip) have no impact, by construction, on the jet mass. The rotation that will have the biggest potential impact on the image mass is when the rotation angle is  $\pi/2$  (maximally changing  $\eta$  and  $\phi$ ), which does lead to a small change in the mass distribution. A translation in  $\eta$  that uses the pixel energy as the intensity instead of the transverse momentum, which we refer to as a *naive translation*, or an  $L^2$  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, which shows the inverse of the background efficiency versus the signal efficiency for passing a threshold on the signal-to-background likelihood ratio of the mass distribution (as described in Section 5). 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.

## 4 Network Architecture

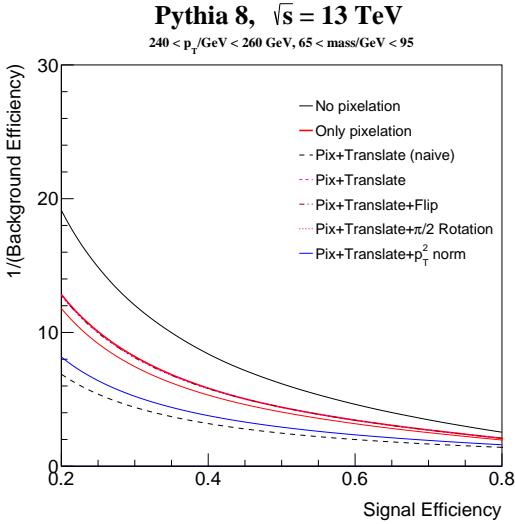
We begin with the notion that the discretization procedure outlined in Section 2 produces  $25 \times 25$  “transverse-energy-scale” images in one channel – a High Energy Physics analogue of a grayscale image. We note that the images we work with are *sparse* – roughly 5-10% of pixels are active on average (see Appendix A for details). Future work can build on efficient techniques for exploiting



**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 pixel transverse momentum. *Flip* denotes the parity inversion operation and the  $p_T^2$  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.

the sparse nature of these images. However, since speed is not our driving force in this work, we used convolution implementations defined for dense inputs. We also study fully connected MaxOut networks [8]. Other architectures were also studied, such as Stack Denoising Autoencoders [33], and multi-layer fully connected networks with various activation functions, but found that convolution and MaxOut networks were the most performant.

As a brief aside, we discuss some of the key neural network concepts which are used in the following section to describe our network architectures. Fully connected (FC) layers take all features as input. Convolution networks utilize convolution filters (or kernels) which are a set of weights  $W$  that operate linearly on a small  $n \times n$  (horizontal  $\times$  vertical) patch of the input image. For instance, a  $3 \times 3$  filter takes as input a  $3 \times 3$  patch of pixels and outputs  $z = \sum_{i,j=1}^3 x_{ij} W_{ij}$ , where  $x_{ij}$  is the input image patch. The filter output can be considered as centered on that patch. Each filter is convolved with the input image, in that the filter is applied to a given input patch and then moved horizontally and/or vertically to a new input patch on which the filter is applied. By scanning over the entire image in this way, a the filter is convolved with the input, producing a convolved output. An important consideration when using convolutional networks is how one handles borders of images. Two main options exist – one can consider only  $n \times n$  patches that are fully contained within the input images, or one can consider every convolution that has at least one pixel from the image, *zero-padding* as

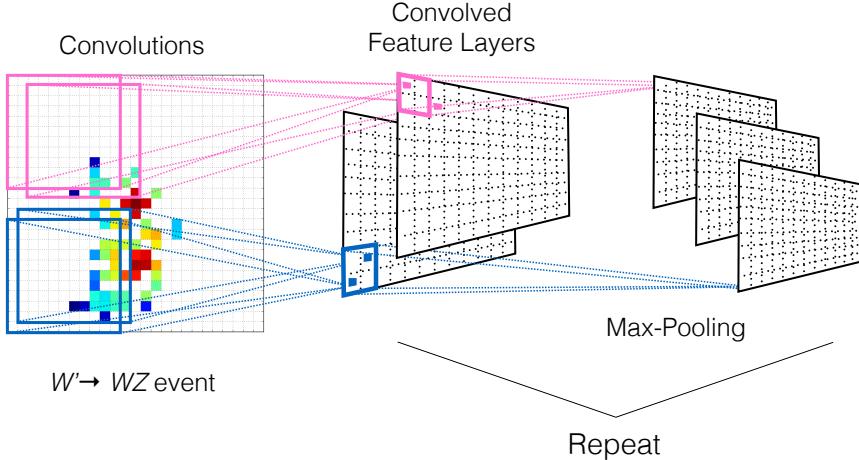


**Figure 4:** The tradeoff between  $W$  boson (signal) jet efficiency and inverse QCD (background) efficiency for various pre-processing algorithms applied to the jet (images). The *No pixelation* line is the jet mass without any detector granularity and without any pre-processing. *Only pixelation* 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 pixel transverse momentum. *Flip* denotes the parity inversion operation and the  $p_T^2$  norm is a  $L^2$  normalization scheme.

necessary to create valid convolutions. We use the latter, as we found better performance and better, more physics-driven filters.

A non-linear activation function is typically applied to these convolution outputs, for which we use the Rectified Linear Unit (ReLU) [34] that takes an input  $z$  and outputs  $\max\{0, z\}$ . ReLU's have been found to improve network training time, whilst having enough non-linear behavior to not degrade network performance. In addition, Rectified Linear Units do not suffer from a vanishing gradient, and speed up computation time while allowing for sparse networks by having true zero-valued activations. After convolution(+activation) layers, a non-linear down-sampling is frequently performed using Max-pooling [35] which takes non-overlapping patches of convolution outputs as input, and outputs the maximum value for each patch. A conceptual visualization of the convolution + Max-pooling network architecture that we employ can be seen in Figure 5.

Finally, the MaxOut network makes use of the dense (Fully Connected) MaxOut activation unit, which takes an input vector  $x$  and computes  $k$  linear weightings  $z_{j \in [1, k]} = \sum_i x_i W_{ij} + b_j$  and outputs  $\max_{j \in [1, k]} z_j$ . Natural extensions of MaxOut layers to convolutional units exist, but were not examined. Conceptually, one can view the Rectified Linear Unit as a special case of the MaxOut with  $k = 2$  and with one of the weightings forced to output only zero. Though MaxOut units do not force sparsity of activation outputs in the same way as ReLU units, MaxOut networks provide the desirable attribute that they pair nicely with the model averaging effects of dropout in a natural way [8].



**Figure 5:** The convolution neural network concept as applied to jet-images.

#### 4.1 Architectural Selection

For the MaxOut architecture, we utilize two FC layers with MaxOut activation (the first with 256 units, the second with 128 units, both of which have 5 piecewise components in the MaxOut-operation), followed by two FC layers with ReLU activations (the first with 64 units, the second with 25 units), followed by a FC sigmoid layer for classification. We found that the He-uniform initialization [36] for the initial MaxOut layer weights was needed in order to train the network, which we suspect is due to the sparsity of the jet-image input. In cases where other initialization schemes were used, the networks often converged to very sub optimal solutions. This network is trained (and evaluated) on un-normalized jet-images using the transverse energy for the pixel intensities

For the deep convolution networks, we use a convolutional architecture consisting of three sequential [Conv + Max-Pool + Dropout] units, followed by a local response normalization (LRN) layer [9], followed by two fully connected, dense layers. We note that the convolutional layers used are so called “full” convolutions – i.e., zero padding is added to the input pre-convolution. Our architecture can be succinctly written as:

$$[\text{Dropout} \rightarrow \text{Conv} \rightarrow \text{ReLU} \rightarrow \text{MaxPool}] * 3 \rightarrow \text{LRN} \rightarrow [\text{Dropout} \rightarrow \text{FC} \rightarrow \text{ReLU}] \rightarrow \text{Dropout} \rightarrow \text{Sigmoid}. \quad (4.1)$$

The convolution layers each utilize 32 feature maps, or filters, with filter sizes of  $11 \times 11$ ,  $3 \times 3$ , and  $3 \times 3$  respectively. All convolution layers are regularized with the  $L^2$  weight matrix norm. A down-sampling of  $(2, 2)$ ,  $(3, 3)$ , and  $(3, 3)$  is performed by the three max pooling layers, respectively. A dropout [9] of 20% is used before the first FC layer, and a dropout 10% is used before the output layer. The FC hidden layer consists of 64 units.

After early experiments with the standard  $3 \times 3$  filter size, we discovered significantly worse performance over a more basic MaxOut [8] feedforward network. After further investigation into larger convolutional filter size, we discovered that larger-than-normal filters work well on our application. Though not common in the Deep Learning community, we hypothesize that this larger filter size is helpful when dealing with sparse structures in the input images. In Table 1, we compare different filter sizes, finding the optimal filter size of  $11 \times 11$ , when considering the Area Under the ROC Curve (AUC) metric, based on the ROC curve outlined in Sections 3 and 5.

Kernel size	$(3 \times 3)$	$(4 \times 4)$	$(5 \times 5)$	$(7 \times 7)$	$(9 \times 9)$	$(11 \times 11)$	$(15 \times 15)$
AUC	14.770	12.452	11.061	13.308	17.291	20.286	18.140

**Table 1:** First layer convolution size vs. performance

Two convolution networks, which differ in their pre-processing, are studied in this paper. The first, which we refer to as the ConvNet, is trained (and evaluated) on un-normalized jet-images using the transverse energy for the pixel intensities. The second, which we refer to as ConvNet-Norm, is trained (and evaluated) on  $L^2$  normalized jet-images using the transverse-energy for the pixel intensities. Examining the performance of both networks allows us to study the possible effects of normalization in the pre-processing.

#### 4.2 Implementation and Training

All Deep Learning experiments were conducted in Python with the Keras [37] Deep Learning library, utilizing NVIDIA C2070 graphics cards. One GPU was used per training, but several architectures were trained in parallel on different GPU's to optimize the performance of networks with different hyper-parameters.

We used 8 million training examples, with an additional 2 million validation samples for tuning the hyper-parameters, and 3 million testing samples. Signal examples are weighted such that the total sum of weights is the same as the total number of background examples (as explained in Section 2). These weights are used by the cost function in the training and in the ROC curve computations of the test samples. The networks were trained with the Adam [38] algorithm (Stochastic Gradient Descent with Nesterov Momentum [39] was also examined, but did not provide performance gains). The training consisted of 100 epochs, with a 10 epoch patience parameter on the increase in AUC between 0.2 and 0.8 on a validation set. Batch sizes of 32 were used for the MaxOut network, while batch sizes of 96 were used for the convolution networks.

### 5 Analysis and Visualization

In this section, we examine the performance of the MaxOut and Convolution deep neural networks, described in Section 4, in classifying boosted  $W^\pm \rightarrow qq'$  from QCD jets. As one of our primary goals is to understand what these NN's can learn about jet topology for discrimination, we focus on a restricted phase space of the mass and transverse momentum of the jets. In particular, we restrict our studies to  $250 \text{ GeV} \leq p_T \leq 300 \text{ GeV}$ , and confine ourselves to a  $65 \text{ GeV} \leq m \leq 95 \text{ GeV}$  mass window that contains the peak of the  $W$ . We also perform studies in which the discrimination power of the most discriminating physics variables has been removed, either through sample weighting or highly restrictive phase space selections, which allows us to focus on information learned by the networks beyond such known physics variables. In this way, we construct a scaffolded and multi-approach methodology for understanding, visualizing, and validating neural networks within this jet-physics study, though these approaches could be used broadly.

The primary figure of merit used to compare the performance of different classifiers is the ROC curve. The ROC curves allow us to examine the entire spectrum of trade-off between Type-I and Type-II errors<sup>3</sup>, as many applications of such classifiers will choose different points along the trade-off curve.

---

<sup>3</sup>In this context, Type-I errors refer to incorrectly rejecting the signal, while Type-II errors refer to incorrectly accepting the background.

Since the classifier output distributions are not necessarily monotonic in the signal-to-background ratio, for each classifier we compute the signal-to-background likelihood ratio<sup>4</sup>. The ROC curves are computed by applying a threshold to the classifier output likelihood ratio, and plotting the inverse of the fraction of background jet passing the threshold (the background rejection) versus the fraction of signal events passing the threshold (the signal efficiency). We say that a classifier is *strictly* more performant if the ROC curve is above a baseline for all efficiencies. In decision theory, this is often referred to as domination (i.e. one classifier dominates another). It should be noted that any weights used to modify the distributions of jets (e.g. the  $p_T$  weighting described in Section 2) are also used when computing the ROC curves.

For information exploration, several techniques were used:

- **ROC Curve Comparisons to Multi-Dimensional Likelihood Ratios:** By combining several physics-inspired variables and computing their joint likelihood ratio, we can explore the difference between such multi-dimensional likelihood ratios and the neural networks' performance. We also compute the joint likelihood ratio of the neural network output and physics-inspired variables. If such joint classifiers improve upon the neural network performance, then we can consider the information in the physics-inspired variable (conditioned on the neural network output) as having been learned by the neural network. If the joint classifier shows improved performance over the neural network, then the neural network has not completely learned the information contained in the physics-inspired variable.
- **Convolution Filters:** For convolution neural networks, we display the weights of the 11x11 filters as images. These filters show how discrimination information is distributed throughout patches of the jets and give a view of the higher level representations learned by the network. However, such filters are not always easy to interpret, and thus we also convolve each filter with a set of signal and background jet-images. We then examine the difference between the convolution output on the average signal jet-images and average background jet-images. These differences give deeper insight into how the filters act on the jets to accentuate discriminating information.
- **Joint and Conditional Distributions:** We examine the joint and conditional distributions of various physics inspired features and the neural network outputs. If the conditional distribution of the physics variable  $v$  given the neural network output  $O$  is not independent of the neural network output, i.e.  $P(v|O) \neq P(v) \forall O$ , then we consider the network to have learned information about this physics feature.
- **Average, Difference, and Fisher Jet-Images:** We examine average images for signal and background and their differences, as well as the Fisher Jets. This is particularly illuminating when we select jets with specific values of highly discriminating physics-inspired variables. This allows us to explore discriminating information contained in the jet images beyond the physics inspired variables.
- **Neural Network Correlations per Pixel:** We compute the linear correlations (i.e. Pearson correlation coefficient) between the neural network output and the distributions of intensity in each pixel. This allows for a visualization of how the discriminating information learned by the

---

<sup>4</sup>Practically, this is done by binning the distribution using variable width bins such that each bin has a fixed number of background events. This number of background events is used to regulate the approximation and we check that the results are not sensitive to this choice.

neural network is distributed throughout the jet. These visualizations are an approximation to the neural network discriminator and can be used to aid the development of new physics inspired variables (much like the Fisher Jet visualization).

The performance evaluation and information exploration techniques are examined in three settings, all of which require the aforementioned mass and transverse momentum selection.

1. **General Phase Space:** No alterations are made to the phase space. This gives an overview of the performance and information learned by the networks
2. **Uniform Phase Space:** The weight of each jet is altered such that the joint distributions of mass,  $n$ -subjettiness, and  $p_T$  are non-discriminative. Specifically, we derive weights such that:

$$f(m, \tau_{21}, p_T | W' \rightarrow WZ) \approx f(m, \tau_{21}, p_T | QCD). \quad (5.1)$$

Both the weighting and network evaluation are performed in a slightly more restricted phase space requiring  $\tau_{21} \in [0.2, 0.8]$ . While  $p_T$  is weighted in all phase space setting, mass and  $n$ -subjettiness are also weighted in this setting as they are amongst the most discriminating physics-inspired variables. This weighting ensures that mass,  $n$ -subjettiness, and  $p_T$  do not contribute to differences between signal and background, and thus this information is essentially removed from the discrimination power of the samples. This allows us to examine what information beyond these variables has been learned and to understand where the neural network performance improvements beyond these physics derived variables comes from. Neural networks that are trained in the General Phase Space are applied as the discriminant under this “flattening” transformation. We also use the training weights inside this window to train an additional convolution network. We look for increases in performance that would indicate information learned beyond the information contained in the weighted physics variables.

3. **Highly Restricted Phase Space:** The phase space of mass,  $n$ -subjettiness, and  $p_T$  are restricted to very small windows of size:  $m \in [79, 81]$  GeV,  $p_T \in [250, 255]$  GeV, and  $\tau_{21} \in [0.19, 0.21]$ . No weighting (beyond the  $p_T$  weighted described in Section 2) is performed, and the networks trained in the General Phase Space are used for discrimination and evaluation. This highly restricted window provides a different method to effectively remove the discrimination power of mass,  $n$ -subjettiness, and  $p_T$  as there is little to no variation of the variables in this phase space for either signal or background. Thus, any discrimination improvements of the neural networks over the physics-inspired variables would be coming from information learned beyond these variables. While the weighting in the Uniform Phase Space is designed also to remove such discrimination, it produces a non-physical phase space. The Highly Restricted Phase Space allows us to ensure that the neural network performance improvements are valid and transferrable to a less contrived phase space.

By examining the performance of the neural networks in these different phase spaces, we aim to systematically remove known discriminative information from the networks’ performance and thereby probe the information learned beyond what is already known by physics inspired variables.

### 5.1 Studies in the General Phase Space

In order to evaluate the overall discrimination performance of the DNNs to that of the physics-driven variables, we examine the ROC curves in Figure 6. In particular, we compare the DNNs to

$n$ -subjettiness [31]  $\tau_{21} = \tau_2/\tau_1$ , the jet mass, and the distance  $\Delta R$  between the two leading  $p_T$  subjets. In Figure 6a, we can see that the three DNNs have similar performance, but the MaxOut networks outperforms the ConvNet networks. We suspect that the MaxOut outperforms the ConvNets due to sparsity of the jet-images, whereby the MaxOut network views the full jet-image from the initial hidden layer while the sparsity tends to make it difficult for the ConvNets to learn meaningful convolution filters. We also see that the ConvNet-Norm outperforms the ConvNet trained on the un-normalized jet-images. We observe that the classification performance of the ConvNet discriminant is highest when jet images are normalized, despite the fact that image normalization destroys jet mass information from the images. As we will see soon, it is difficult for these networks to fully learn the jet mass, so the lack of mass information from pre-processing does not necessarily lead to worse discrimination performance. On the other hand, normalization is having an impact on the ability to effectively train the ConvNet network on jet images. Finally, we see that the DNNs significantly improve the discrimination power relative to the Fisher-Jet discriminant<sup>5</sup>, as described in reference [5]. In addition, in Figure 6b we see that the DNNs also outperform the two-variable combinations of the physics inspired variables (computed using the 2D likelihood ratio<sup>6</sup>). It is interesting to note that combining mass and  $\tau_{21}$ , or  $\tau_{21}$  and  $\Delta R$ , achieve much higher performance than the individual variables and are significantly closer to the performance of the DNNs. However, the large difference in performance between the DNNs and the physics-variable combinations implies the DNNs are learning information beyond these physics variables.

While we can see in Figure 6 that the DNNs outperform the individual and two-variable physics inspired discriminators, we want to understand if these physics variables have been learned by the networks. As such, we compute the combination of the DNNs with each of the physics inspired variables (using the 2D likelihood), as seen for the ConvNet in Figure 7a and for the MaxOut network in Figure 7b. In both cases, we see that the discriminators combining  $\Delta R$  or  $\tau_{21}$  with the DNNs does not improve performance. This indicates that the discriminating information in these variables relevant for the classification task has already been fully learned by the networks<sup>7</sup>. However, adding mass in combination with the DNNs shows a noticeable improvement in performance over the DNNs alone. This indicates that not all of the discriminating information relevant for jet tagging contained in the mass variable has been learned by the DNNs. While it is not shown, similar patterns are found for the Convnet-Norm network.

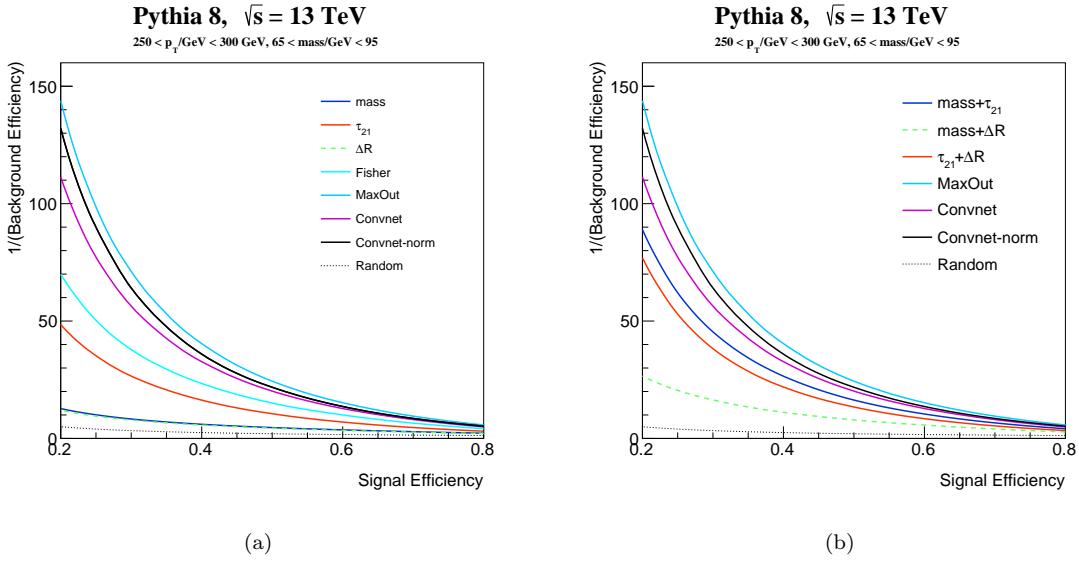
The conditional distributions between the DNN output and the physics-variables are shown in Figure 8a for the ConvNet, and in Figure 8b for the MaxOut network, against the jet mass,  $\Delta R$ , and  $\tau_{21}$ . These distributions are normalized in bins of the DNN output, and thus the  $z$ -axis shows a discretized estimate of the conditional probability density of a physics variable value given the network output (i.e.  $\text{Pr}(\text{variable}|\text{network output})$ ). Normalizing the distributions in this way allows us to see the most probable values of the physics variables at each point of the network output, without being affected by the overall distribution of jets in this 2D space. We can see that both networks show a strong non-linear relationship with  $\tau_{21}$  and  $\Delta R$ , giving further evidence that this information has been

---

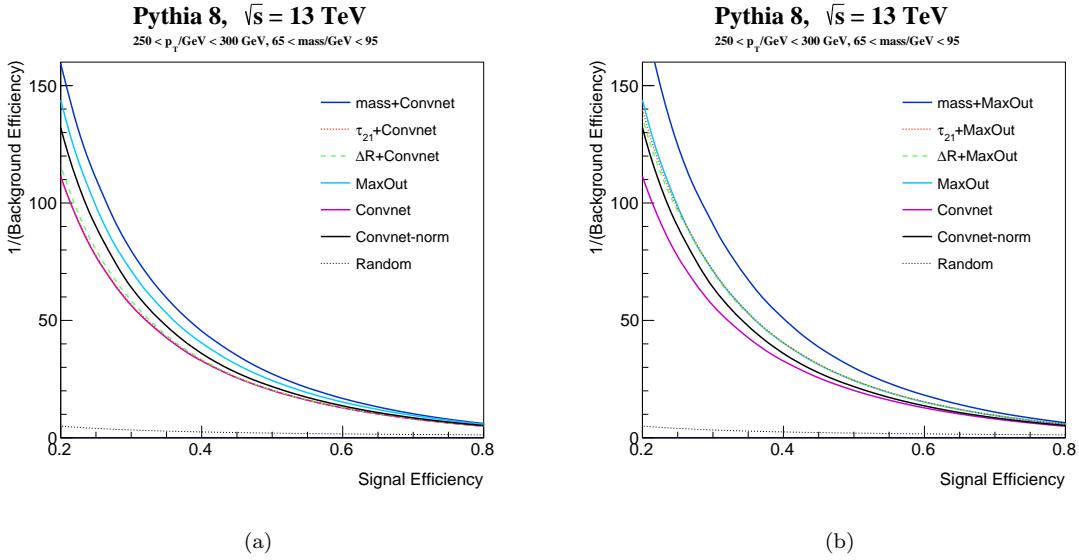
<sup>5</sup>The Fisher discriminant is trained in three partitions of  $\Delta R$  ( $\Delta R \in [0.25, 0.5]$ ,  $[0.5, 0.75]$ ,  $[> 0.75]$ ), in order to account for the non-linear variation in jet-images from the differing positions of the two subjets. Also note that unlike in the original implementation, here we do not normalize the jet images when computing the Fisher Jet. This leads to slightly better performance.

<sup>6</sup>This is computed using the same regulated binning scheme as the 1D likelihoods described earlier.

<sup>7</sup>This is not strictly speaking true, since there may be other variables that are needed in order to fully capture the full information of a given variable. For example, consider independent random variables  $X_i$  that are  $\pm 1$  with probability 1/2. If  $Y = X_1 X_2$ , then  $X_1$  is independent of  $Y$  but the joint distribution of  $(X_1, X_2)$  is not independent of  $Y$ . The statement is true in the absence of interactions with other variables.

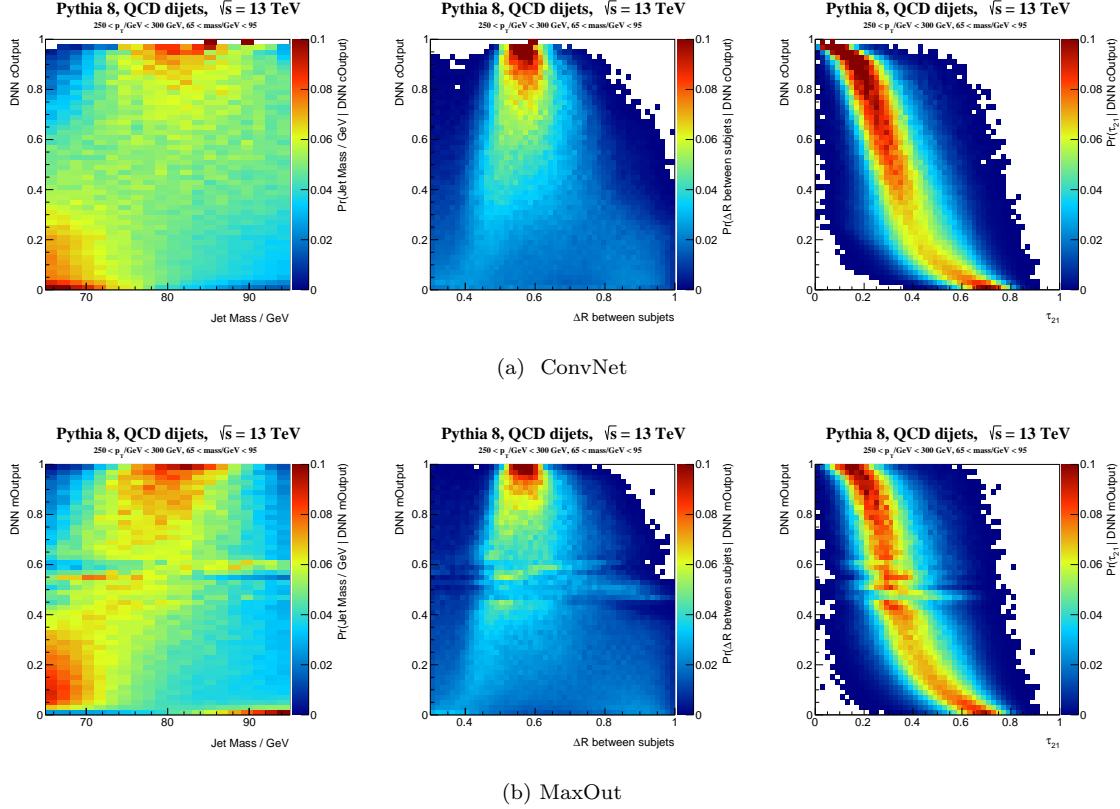


**Figure 6:** Left: ROC curves for individual physics-motivated features as well as three deep neural network discriminants. Right: the DNNs are compared with pairwise combinations of the physics-motivated features.



**Figure 7:** ROC curves that combined the DNN outputs with physics motivated features for the Convnet (left) and MaxOut (right) architectures.

learned by the networks. However, the correlations are much weaker with the jet mass variable, being slightly more strongly correlated with the MaxOut network output. While it is not shown, similar patterns are found for the Conv-Norm network. For reference, the full joint distributions can be found in Appendix B.

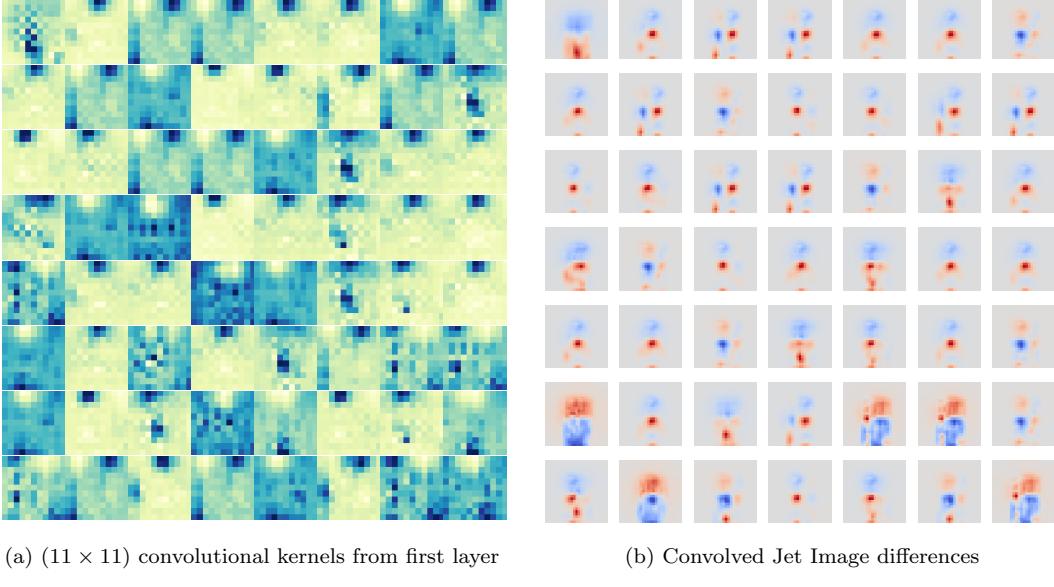


**Figure 8:** Network output versus mass (left),  $\Delta R$  (middle), and  $\tau_{21}$  (right) for the ConvNet (top) and MaxOut (bottom) networks. Each row is normalized and represents the probability distribution of the variable shown on the x-axis given the network output.

## 5.2 Understanding what is learned

In order to gain a deeper understanding of the physics leaned by the DNNs, in this section we examine how the internal structure of the network relates to the substructure and properties of W bosons versus QCD jets.

In Figure 9a, we show the first layer  $11 \times 11$  convolutional filters learned by the Conv-Norm network. Each filter is visualized by showing the learned weight in each position of the filter  $W_{ij}$  from Section 4. We can see that there is variation between filters, indicating that they are learning different features of the jet-images, but this variation is not as large as seen in many CV problems due to the sparsity of the jet-images. We also see that they tend to learn representations of the subjets and distances between subjets, as seen by the circular features found in many of the filters.



**Figure 9:** Convolutional Kernels (left), and convolved feature differences in jet images (right)

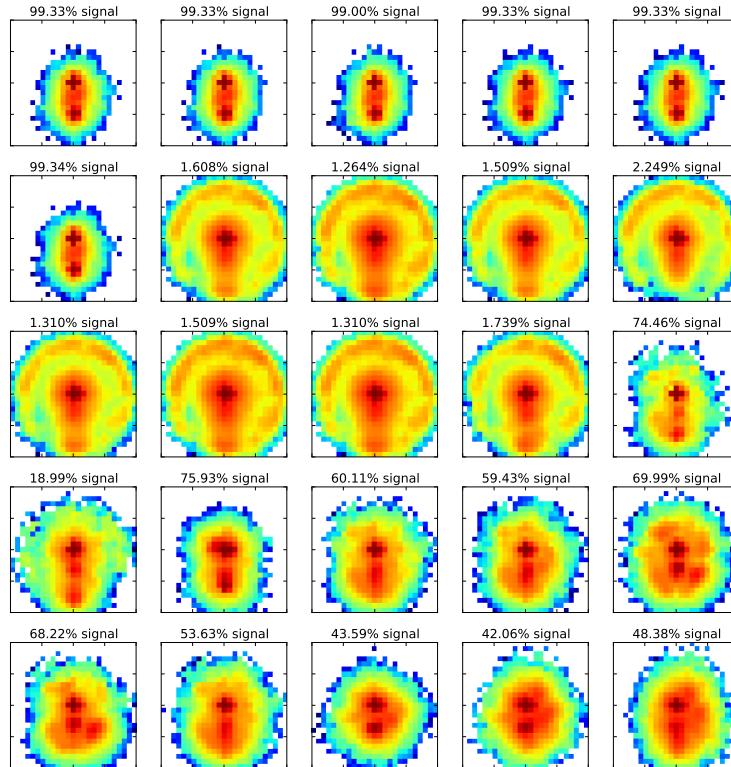
To get a better understanding of how these filters provide discrimination, we mimic the operation in the first layer of the network by convolving each filter with average of large samples of signal and background jet images. The difference between the convolved average signal and background jet-images helps to provide an understanding of what difference in features the network learns at the first layer in order to help discriminate.

More formally, let  $J_s = \frac{1}{n} \sum_{i:i} J^{(i)}$  is signal  $J^{(i)}$  and  $J_b = \frac{1}{n} \sum_{i:i} J^{(i)}$  is background  $J^{(i)}$  represent the average signal and background jet over a sample, where  $J^{(i)}$  is the  $i$ th 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 by computing:

$$J_s * w_i - J_b * w_i, \forall i, \quad (5.2)$$

where  $*$  is the convolution operator. We arrange these new ‘‘convolved jet-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 9b, 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. We also draw attention to the fact that there is a large diversity in the convolved representations, indicating that the DNN is able to learn and pick up on multiple features that are descriptive.

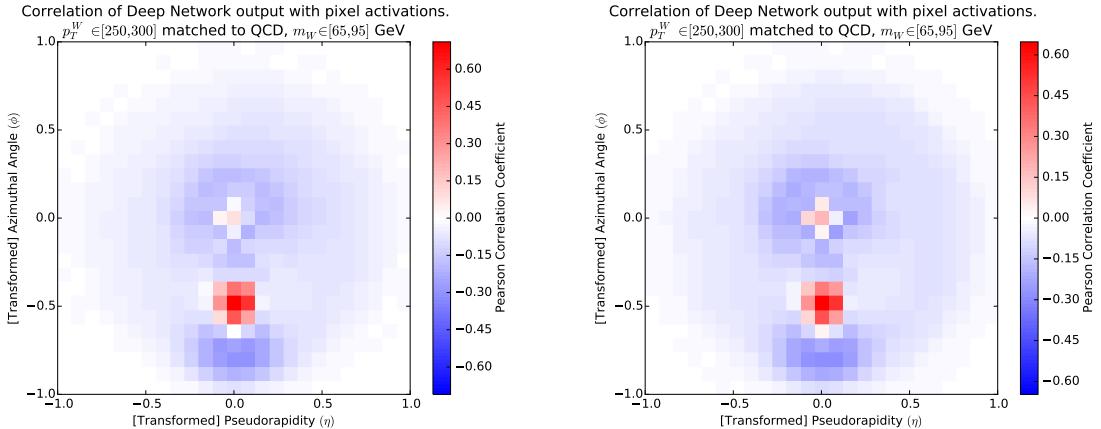
A related way to visualize the information learned by various nodes in the network is to consider the jet images which most activate a given node. Fig. 10 shows the average of the 500 jet images with the highest node activation for the last hidden layer of the MaxOut network (the layer before the classification layer). The first row of images in Fig. 10 show clear two-prong signal-like structure whereas the second and third rows show one-prong diffuse radiation patterns that are more background-like. The remaining rows have a variety of  $\Delta R$  distances between subjets and have a mix of background and signal-like features.



**Figure 10:** The average of the 500 jet images with the highest node activation for the last hidden layer of the MaxOut network. The nodes are ordered from top left to bottom right by increasing sparsity. The top left is the most commonly activated node whereas the bottom right node is least activated and frequently zero.

### 5.3 Physics in Deep Representations

To get a tangible and more intuitive understanding of what jet structures a DNN learns, we compute the correlation of the DNN output with each pixel of the jet-images. Specifically, let  $y$  be the DNN output, and consider the intensity of each pixel  $I_{ij}$  in transformed  $(\eta, \phi)$  space. We construct an image, which we denote the *deep correlation jet-image*, where each pixel  $(i, j)$  is  $\rho_{I_{ij}, y}$ , the Pearson Correlation Coefficient of the pixels intensity with the final DNN output, across images. While this image does not give a direct view of the discriminating information learned within the network, it does provide a guide to how such information may be contained within the network. In Figure 11, we construct this deep correlation jet-image for both the ConvNet and the MaxOut networks. We can see that the location and energy of the subleading subjet, found at the bottom of the image, is highly correlated with the DNN output and important for identifying signal jet-images. In contrast, the information contained in the leading subjet, seen at  $(x, y) \sim (0, 0)$  in the image, is not particularly correlated with the network output owing to the fact that both signal and background jets have high energy leading subjets. We also see asymmetric regions around both subjets that are correlated with the DNN output and indicate the presence of additional radiation expected in the QCD background jets. Finally, a small negative correlation with the rest of the jet area is seen, indicating that radiation from the background jets is more likely to be observed in these regions. The exact function form of these distributions are not known, nor does it seem to describe exactly any known physics inspired variable.



**Figure 11:** Per-pixel linear correlation with DNN output for the Convnet (left) and the MaxOut network (right). Signal and background jets are combined.

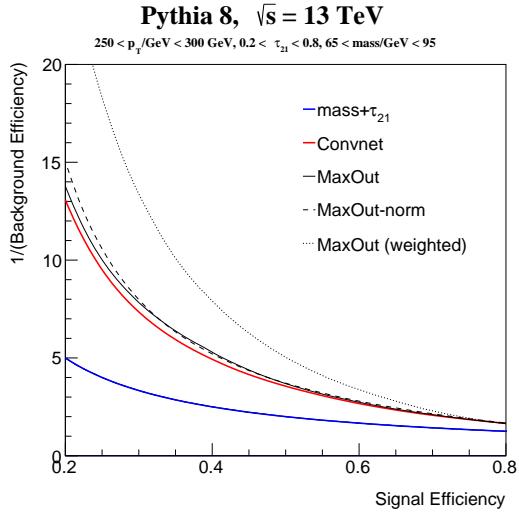
## 5.4 Studies in the Uniform Phase Space

An important part of the investigation into what the neural networks are learning beyond the standard physics features is to quantify the performance when these features are removed. This represents the *unique information* learned by the network. One way to remove the discrimination power from a given feature is to apply a transformation such that the marginal likelihood ratio is constant at unity. In other words, we derive event-by-event weights such that

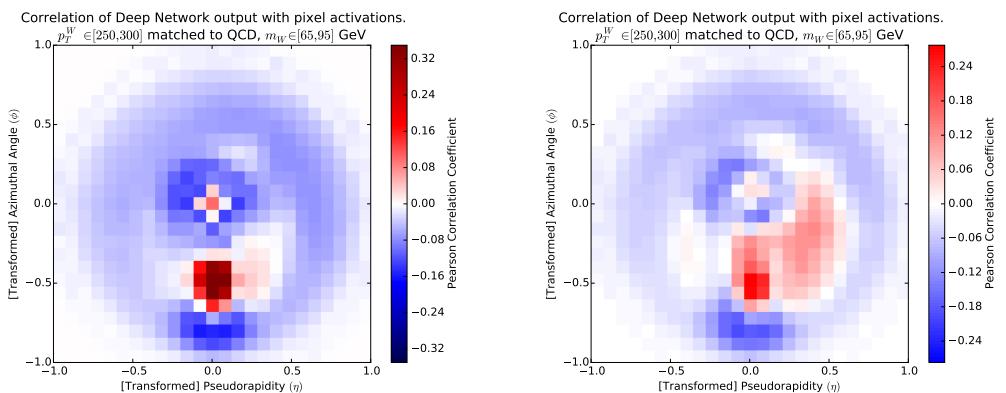
$$f(m, \tau_{21}, p_T | W' \rightarrow WZ) \approx f(m, \tau_{21}, p_T | QCD), \quad (5.3)$$

where  $f(X|Y)$  is the probability density function of  $X$  given  $Y$ . This is done practically by binning the mass and  $\tau_{21}$  distributions and then assigning to each event a weight given by the inverse bin content corresponding to the jet mass and  $\tau_{21}$  of that particular event. Figure 12 shows the ROC curve for various features with this weighting scheme applied. By construction,  $\tau_{21}$  and the jet mass do not have any discrimination power between signal and background, evident by the fact that  $\epsilon_{\text{bkg}} = \epsilon_{\text{signal}} =$  the random guess line. However, the convolutional network that is trained inclusively (without the weights from Equation 5.3) does have some discrimination power when the weights from Equation 5.3 are applied. For a fixed signal efficiency, the overall performance is significantly degraded with respect to the un-weighted ROC curve in Figure 6, but the improvement over a random guess is significant. Interestingly, the network performance is significantly better in this re-weighted setting when the same weighting is applied during training (effort by the network is not needed to learn  $\tau_{21}$ , for instance). The ConvNet and MaxOut procedures training inclusively have similar performance.

One can gain intuition about the unique information learned by the network by studying the correlation of the network output and the pixel intensities with the Equation 5.3 weights applied. This is shown in Figure 13 with and without the weights applied during training. The two correlation plots are qualitatively similar, but the region to the right of the subjets is more enhanced when the weights are applied during the training. This suggests that information about radiation surrounding the subjets contains important discrimination power contributing to the network's unique information.



**Figure 12:** Various ROC curves with event weights that enforce Eq. 5.3 inside  $m \in [65, 95]$  GeV,  $p_T \in [250, 300]$  GeV, and  $\tau_{21} \in [0.2, 0.8]$ . By construction, the  $\tau_{21}$  and likelihood combination of  $\tau_{21}$  and mass are non-discriminating (and are thus equal to a random guess). The ConvNet, MaxOut, and MaxOut-Norm networks are trained without the weights applied and the MaxOut (weighted) line was trained with the weights applied during training.



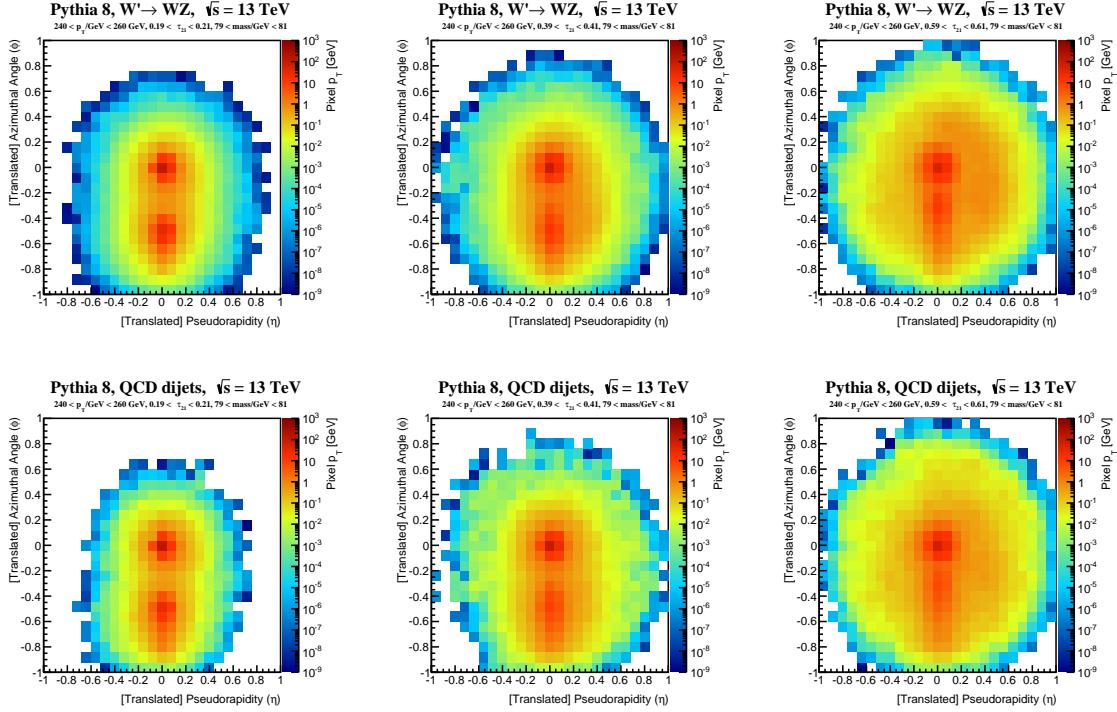
**Figure 13:** Pearson Correlation Coefficient for pixel intensity and the convolutional neural network output for  $W' \rightarrow WZ$  and QCD (combined) for the MaxOut network training inclusively and then weighted (left) and for the MaxOut network training with the weights from Equation 5.3 applied also during the training.

## 5.5 Studies in the Highly Restricted Phase Space

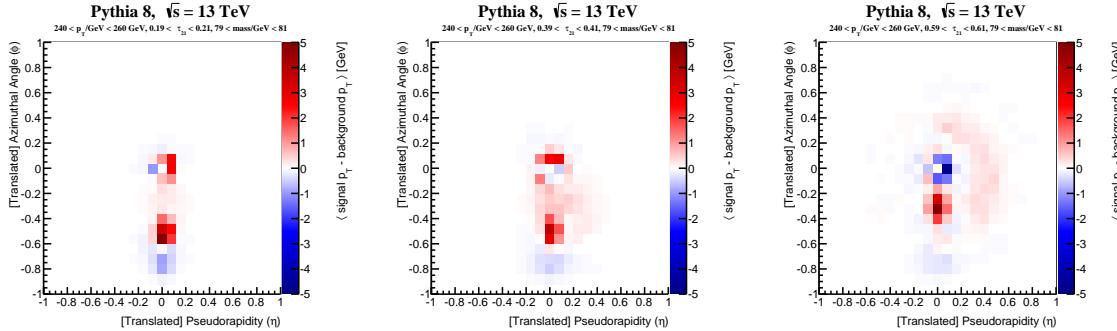
Another way to quantify the unique information learned by the network that also provides useful information about physical information learned by the network is to restrict the considered phase space such that  $\tau_{21}$  and the jet mass distributions do not vary appreciably over the reduced space. Figure 14 shows the average signal and background jet image in three small windows of  $\tau_{21}$ , jet mass, and jet  $p_T$ . In all three windows, the jet mass is restricted to be between 79 GeV and 81 GeV and the jet  $p_T$  is required to be in the interval [250,260] GeV. The three windows are then defined by their value of  $\tau_{21}$ : [0.19,0.21] in the most two-prong-like case, [0.39,0.41] in a region with likelihood ratio near unity and [0.59,0.61] in a mostly one-prong-like case. The key physics features of the jets falling in these windows are easily visualized from the average jet images. The most striking observation is that in these three windows, signal jets look very similar to background jets. When  $\tau_{21} \in [0.19, 0.21]$ , both signal and background jets have a second subjet that is distinct from the leading subjet, which becomes less prominent as the value of  $\tau_{21}$  increases.

The differences between images in these small windows tells us about what information *could be learned* by the networks beyond  $\tau_{21}$  and the jet mass. Since the differences are subtle, the average difference is explicitly computed and plotted in Figure 15 for the three narrow windows of  $\tau_{21}$ . In the window with  $\tau_{21} \in [0.19, 0.21]$ , there are five features: a localized blue patch in the bottom center, a localized red patch just above that, a red diffuse region between the red patch and the center and then a blue dot just left of center surrounded by a red shell to the right. Each of these have a physics meaning: the lower two localized patches give information about the orientation of the second subjet ( $\Delta R$ ) which is slightly wider for the QCD jets which need a slightly wider angle to satisfy the mass requirement. The red diffuse region just above the localized patches is likely an indication of colorflow: the  $W$  bosons are color singlets compared to the color octet gluon jet background, and thus we expect the radiation pattern to be mostly between the two subjets for the  $W$ . One can draw similar conclusions for all the features in each of the plots in Figure 15.

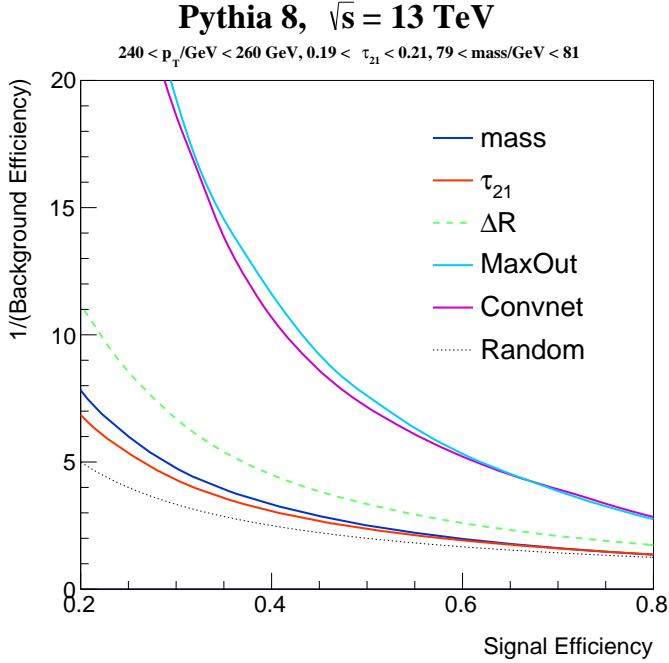
Now, we turn back to the neural network and their performance in these small windows of jet mass and  $\tau_{21}$ . Figure 16 shows three ROC curves in the window  $\tau_{21} \in [0.19, 0.21]$ . By construction, the  $\tau_{21}$  and jet mass curves are not much better than a random guess, since these variables do not significantly vary over the small window. The other curves show the performance of  $\Delta R$  and the ConvNet and MaxOut neural networks trained inclusively, which have similar performance to each other. As in the previous section, this allows us to quantify the unique information in the neural network. One way to *visualize* the unique information is to look at the per-pixel correlation between the intensity and neural network output (Figure 17). The physical interpretation of the red and blue areas in Figure 17 are related to the colorflow of  $W$  and background jets. The area in-between the subjets should have more radiation than the area around and outside of the subjets for  $W$  jets and vice-versa for QCD jets. While Figure 17 is not directly the discriminant used in the network and only represents linear correlations with the network output, it does show non-linear spatial information and gives a sense of where in the image the network is looking for discriminating features.



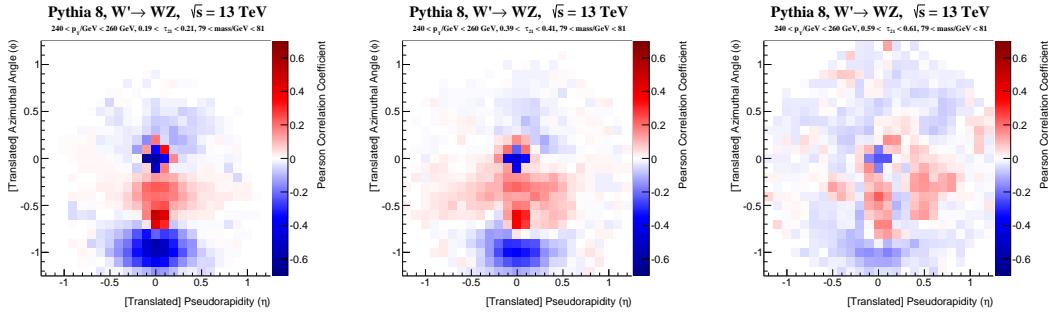
**Figure 14:**  $W' \rightarrow WZ$  (top) and QCD (bottom) average jet-images in three small windows of  $\tau_{21}$ :  $[0.19, 0.21]$  (left),  $[0.39, 0.41]$  (middle), and  $[0.59, 0.61]$  (right). In all cases, jet mass is restricted to be between 79 GeV and 81 GeV and the jet  $p_T$  is required to be in the interval [250,260] GeV.



**Figure 15:** The average difference between  $W' \rightarrow WZ$  jet-images in three small windows of  $\tau_{21}$ :  $[0.19, 0.21]$  (left),  $[0.39, 0.41]$  (middle), and  $[0.59, 0.61]$  (right). In all cases, jet mass is restricted to be between 79 GeV and 81 GeV and the jet  $p_T$  is required to be in the interval [250,260] GeV. The red colors are more signal-like and the blue is more background-like.



**Figure 16:** ROC curves for  $m_{\text{jet}} \in [79, 81] \text{ GeV}$ ,  $p_T \in [250, 255] \text{ GeV}$ ,  $\tau_{21} \in [0.19, 0.21]$ . By construction,  $\tau_{21}$  is no better than a random guess in this small window. The neural networks are trained inclusively (but still within the stated mass and  $p_T$  windows).



**Figure 17:** Pearson Correlation Coefficient for pixel intensity and the convolutional neural network output for  $W' \rightarrow WZ$  and QCD (combined) in three small windows of  $\tau_{21}$ :  $[0.19, 0.21]$  (left),  $[0.39, 0.41]$  (middle), and  $[0.59, 0.61]$  (right). In all cases, jet mass is restricted to be between 79 GeV and 81 GeV and the jet  $p_T$  is required to be in the interval  $[250, 260]$  GeV.

## 6 Outlook and Conclusions

Jet Images are a powerful paradigm for visualizing and classifying jets. We have shown that when applied directly to jet images, deep neural networks are a powerful tool for identifying boosted hadronically decaying  $W$  bosons from QCD multijet processes. These advanced Computer Vision algorithms outperform several known and highly discriminating engineered physics-inspired features such as the jet mass and  $n$ -subjettiness,  $\tau_{21}$ . Through a variety of studies, we have shown that some of these features are *learned* by the network. However, despite detailed studies to preserve the jet mass, this important variable seems to not be fully captured by the neural networks studied in this article. Understanding how to fully learn the jet mass is a goal of our future work.

In this paper, we propose several techniques for quantifying and visualizing the information learned by the DNNs, and connect these visualizations with physics properties. This is studied by removing the information from jet mass and  $\tau_{21}$  through a re-weighting or redaction of the phase space. In this way, we can evaluate the performance of the network beyond these features to quantify the unique information learned by the network. In addition to quantifying the amount of additional discrimination achieved by the network, we also show how the new information can be visualized through the deep correlation jet image which displays the network output correlation with each input pixel. These visualizations are a powerful tool for understanding what the network is learning - in this case, colorflow patterns suggest that at least part of the unique information comes from the octet versus singlet nature of  $W$  bosons and gluon jets. These visualizations may even be useful in the future for engineering other simple variables which may be able to match the performance of the neural network.

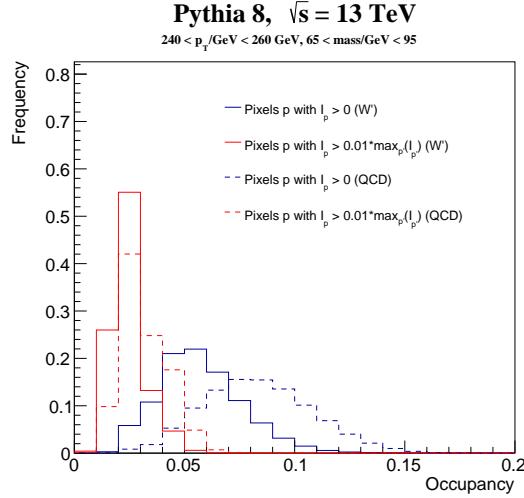
This edition of the study of jet images has built a new link between particle physics and computer vision by using state of the art deep neural networks for classifying high-dimensional high energy physics data. By processing the raw jet image pixels with these advanced techniques, we have shown that there is a great potential for jet classification. Many analyses at the LHC use boosted hadronically decaying bosons as probes of physics beyond the Standard Model and the methods presented in this paper have important implications for improving the sensitivity of these analyses. In addition to improving tagging capabilities, further studies with deep neural networks will help us discover new features to improve our understanding and improve upon existing features to fully capture the wealth of information inside jets.

## 7 Acknowledgements

We would like to thank Andrew Larkoski for useful conversations about the physics observed in the jet images. This work is supported by the Stanford Data Science Initiative and by the US Department of Energy (DOE) under grant DE-AC02-76SF00515. BN is supported by the NSF Graduate Research Fellowship under Grant No. DGE-4747 and by the Stanford Graduate Fellowship.

## A Image Sparsity

Figure 18 quantifies the sparsity of the jet images by showing the distribution of the pixel occupancy: the fraction of pixels that have a non-zero entry. Also plotted is the fraction of pixels that have at least 1% of the intensity of the scalar sum of the pixel intensities from all pixels. In general, the background has a more diffuse radiation pattern and thus the corresponding jet images have a higher average occupancy.

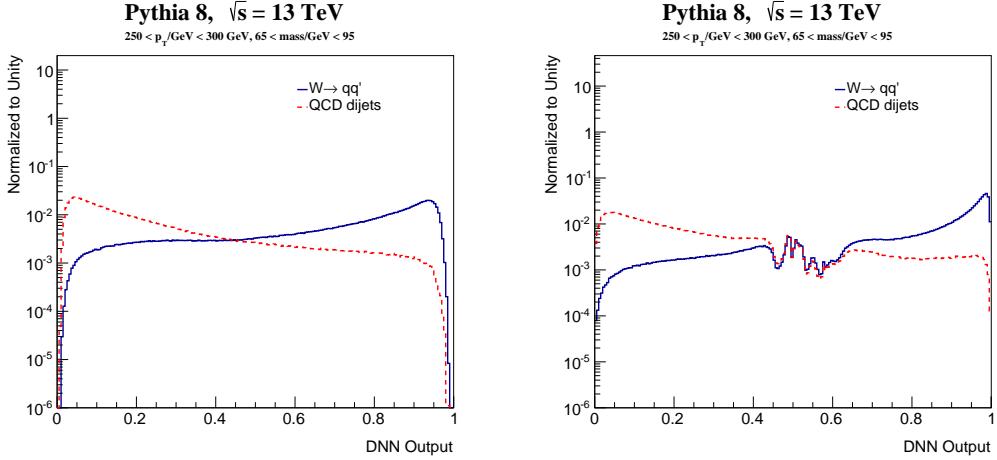


**Figure 18:** The distribution of the fraction of pixels (occupancy) that have a nonzero entry (blue) or at least 1% of the scalar sum of the pixel intensities from all pixels (red).

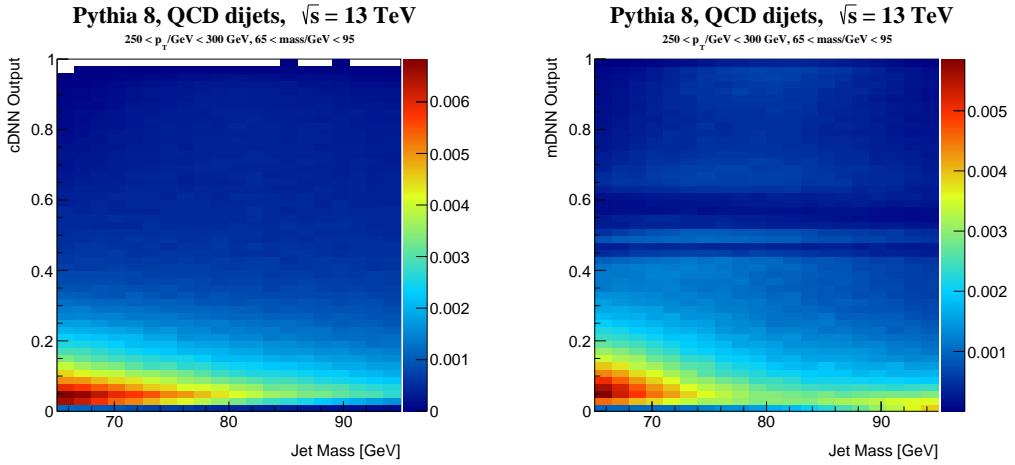
## B Joint and Marginal Distributions

Figure 19 shows the marginal distributions of the network outputs for signal and background jets. The MaxOut network has a wavy feature in the distribution near 0.5 where the likelihood ratio is unity. In that regime, the network cannot differentiate between signal and background and in this particular case results in a non-smooth distribution at the fixed likelihood ratio value.

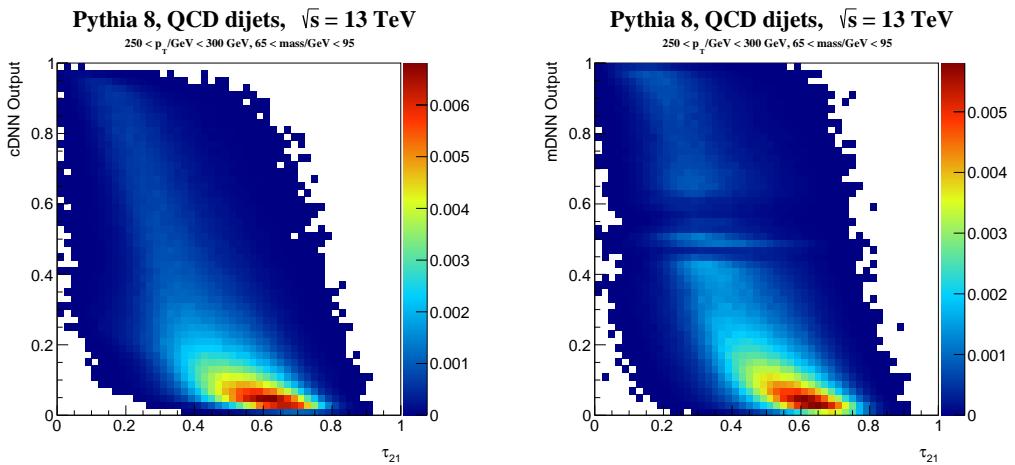
The joint distributions of the network with the jet mass,  $\tau_{21}$ , and the  $\Delta R$  between subjets are shown in Fig. 20, Fig. 21, and Fig. 22, respectively. The joint distributions between the various combinations of the physics features are shown in Fig. 23 and Fig. 24.



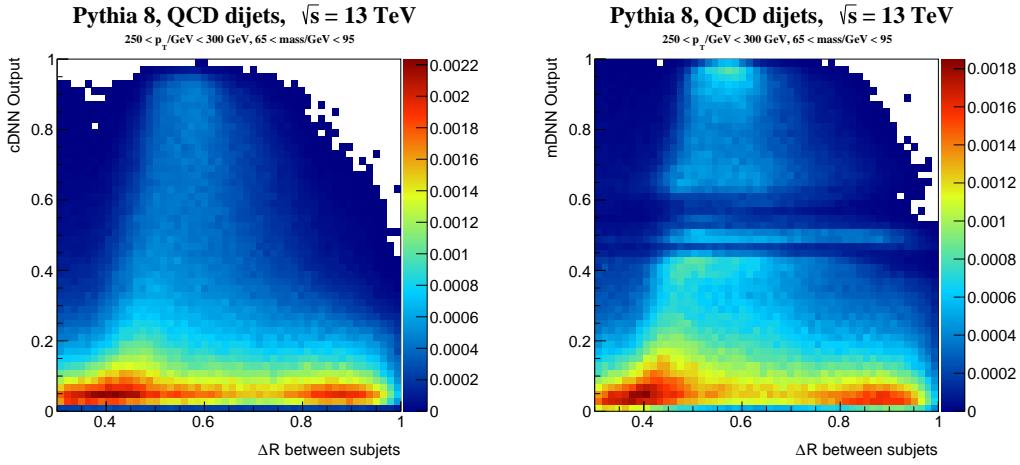
**Figure 19:** The marginal distributions of the ConvNet (left) and MaxOut (right) network outputs for signal and background jet images.



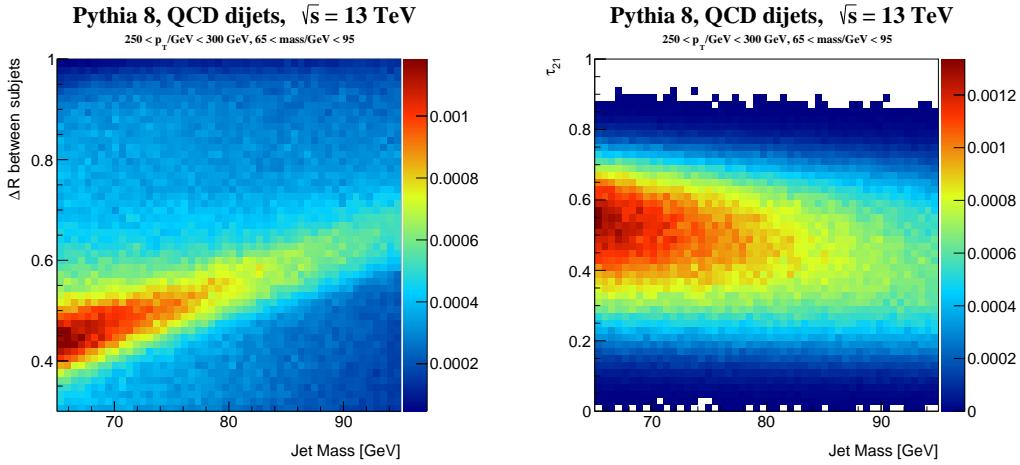
**Figure 20:** The joint probability distribution the jet mass and the ConvNet (left) and MaxOut (right) network outputs for the background.



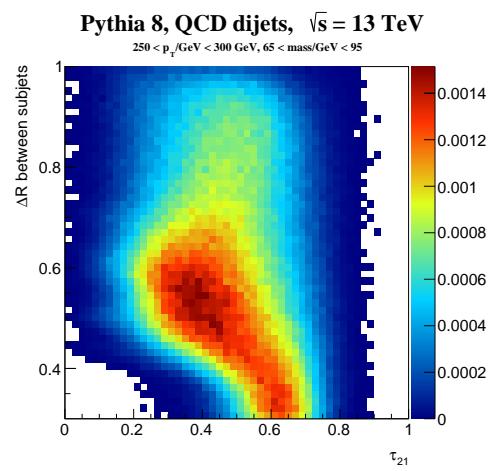
**Figure 21:** The joint probability distribution between  $\tau_{21}$  and the ConvNet (left) and MaxOut (right) network outputs for the background.



**Figure 22:** The joint probability distribution between the  $\Delta R$  between subjets and the ConvNet (left) and MaxOut (right) network outputs for the background.



**Figure 23:** The joint probability distribution between jet mass and the  $\Delta R$  between subjets (left) and  $\tau_{21}$  (right) for the background.



**Figure 24:** The joint probability distribution between the  $\Delta R$  between subjets and  $\tau_{21}$  for the background.

## References

- [1] L. Evans and P. Bryant, *LHC Machine*, *JINST* **3** (2008) S08001.
- [2] A. Altheimer *et. al.*, *Jet Substructure at the Tevatron and LHC: New results, new tools, new benchmarks*, *J. Phys.* **G39** (2012) 063001 [[1201.0008](#)].
- [3] A. Altheimer *et. al.*, *Boosted objects and jet substructure at the LHC. Report of BOOST2012, held at IFIC Valencia, 23rd-27th of July 2012*, *Eur. Phys. J.* **C74** (2014), no. 3 2792 [[1311.2708](#)].
- [4] D. Adams *et. al.*, *Towards an Understanding of the Correlations in Jet Substructure*, *Eur. Phys. J.* **C75** (2015), no. 9 409 [[1504.00679](#)].
- [5] J. Cogan, M. Kagan, E. Strauss and A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, *JHEP* **02** (2015) 118 [[1407.5675](#)].
- [6] L. G. Almeida, M. Backović, M. Cliche, S. J. Lee and M. Perelstein, *Playing Tag with ANN: Boosted Top Identification with Pattern Recognition*, *JHEP* **07** (2015) 086 [[1501.05968](#)].
- [7] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, *CoRR* **abs/1409.1556** (2014).
- [8] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville and Y. Bengio, *Maxout Networks*, *ArXiv e-prints* (Feb., 2013) [[1302.4389](#)].
- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, *CoRR* **abs/1207.0580** (2012).
- [10] CMS Collaboration, V. Khachatryan *et. al.*, *Identification techniques for highly boosted W bosons that decay into hadrons*, *JHEP* **12** (2014) 017 [[1410.4227](#)].
- [11] *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.
- [12] *Performance of Boosted W Boson Identification with the ATLAS Detector*, Tech. Rep. ATL-PHYS-PUB-2014-004, CERN, Geneva, March, 2014.
- [13] 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\*\*](#).
- [14] 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](#)].
- [15] 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\*\*](#).
- [16] 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\*\*](#).
- [17] 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\*\*](#).
- [18] 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\*\*](#).
- [19] 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\*\*](#).

- [20] 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. B* **748** (2015) 255–277 [[1502.04994](#)].
- [21] 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](#)].
- [22] 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](#).
- [23] ATLAS Collaboration, G. Aad *et. al.*, *Search for an additional, heavy Higgs boson in the  $H \rightarrow ZZ$  decay channel at  $\sqrt{s} = 8$  TeV in pp collision data with the ATLAS detector*, [1507.05930](#).
- [24] 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. C* **75** (2015), no. 5 209 [[1503.04677](#)]. [Erratum: *Eur. Phys. J.C*75,370(2015)].
- [25] 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](#)].
- [26] T. Sjostrand, S. Mrenna and P. Z. Skands, *A Brief Introduction to PYTHIA 8.1*, *Comput. Phys. Commun.* **178** (2008) 852–867 [[0710.3820](#)].
- [27] T. Sjostrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **0605** (2006) 026 [[hep-ph/0603175](#)].
- [28] M. Cacciari, G. P. Salam and G. Soyez, *The Anti- $k(t)$  jet clustering algorithm*, *JHEP* **0804** (2008) 063.
- [29] M. Cacciari, G. P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896 [[1111.6097](#)].
- [30] D. Krohn, J. Thaler and L.-T. Wang, *Jet Trimming*, *JHEP* **1002** (2010) 084 [[0912.1342](#)].
- [31] J. Thaler and K. Van Tilburg, *Identifying Boosted Objects with N-subjettiness*, *JHEP* **1103** (2011) 015 [[1011.2268](#)].
- [32] A. J. Larkoski, D. Neill and J. Thaler, *Jet Shapes with the Broadening Axis*, *JHEP* **04** (2014) 017 [[1401.2158](#)].
- [33] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio and P.-A. Manzagol, *Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion*, *Journal of Machine Learning Research* **11** (2010).
- [34] X. Glorot, A. Bordes and Y. Bengio, *Deep sparse rectifier neural networks*, *Journal of Machine Learning Research* **15** (2011).
- [35] D. Scherer, A. Muller and S. Behnke, *Evaluation of pooling operations in convolutional architectures for object recognition*, *Proc. of the Intl. Conf. on Artificial Neural Networks* (2010).
- [36] K. He, X. Zhang, S. Ren and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, *CoRR* **abs/1502.01852** (2015).
- [37] F. Chollet, “Keras.” <https://github.com/fchollet/keras>, 2015.
- [38] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *CoRR* **abs/1412.6980** (2014).
- [39] Y. Nesterov, *A method of solving a convex programming problem with convergence rate  $O(1/\text{sqr}(k))$* , *Soviet Mathematics Doklady* **27** (1983) 372–376.